VOSS Automate Platform Guide

Release 25.2

August 21, 2025

Legal Information

DOCUMENT ID: 20250821114602

Contents

1 What's New **1**
1.1 Platform Guide: Release 25.2 . . . 1

2 Overview **2**
2.1 High-level Functions . . . 2

3 The Command Line Interface **3**
3.1 Overview . . . 3
3.2 CLI Commands . . . 3
3.3 Clustering commands . . . 6
3.4 Using the `tmux` command . . . 7
3.5 System Specific Commands . . . 9
3.6 System Metrics . . . 27
3.7 Data Export Commands . . . 33

4 Cluster Deployment Commands and Steps **37**
4.1 Deployment Commands . . . 37
4.2 Migrate a Unified Node Cluster to a Modular Cluster . . . 38
4.3 Determine the primary node role in a cluster . . . 42

5 Provisioning **44**
5.1 Provisioning . . . 44

6 Networking **45**
6.1 Network interfaces . . . 45
6.2 Switching Network Addresses from IPv4 to IPv6 . . . 47
6.3 Network services . . . 50
6.4 Network URI specification . . . 50
6.5 Network Docker container range . . . 51

7 Application Control **53**
7.1 Application Control and Status . . . 53
7.2 Starting and Stopping . . . 54
7.3 Installing Applications . . . 55
7.4 Updating Applications . . . 58
7.5 Summary Attribute Migration . . . 58
7.6 Remote Execution in Clusters . . . 59
7.7 List of Unused Cluster Commands . . . 59
7.8 Self-service Localization Management . . . 61
7.9 Web Services . . . 61
7.10 Insights Analytics . . . 62

8 System Control **70**

8.1 System Commands Overview . . . 70
8.2 System restart . . . 70
8.3 Passwords . . . 70
8.4 System Boot Passwords . . . 71
8.5 Federal Information Processing Standards (FIPS) . . . 72
8.6 File Management . . . 76
8.7 Drive control . . . 78
8.8 Transaction Prioritization . . . 80
8.9 Banner . . . 80
8.10 Maintenance Mode . . . 81
8.11 Checksum . . . 82
8.12 Terminal Reset . . . 82

9 Diagnostics **83**
9.1 Login Report . . . 83
9.2 Cluster Check . . . 85
9.3 Enable Health Monitoring . . . 88
9.4 Enable Database Scheduling . . . 89
9.5 Command History . . . 90
9.6 Logs . . . 90
9.7 Viewing Logs . . . 92
9.8 Sending and Collecting Logs . . . 93
9.9 Log Types . . . 97
9.10 Audit Log Rule Sets . . . 99
9.11 Log Type Commands . . . 102
9.12 Audit Log Format and Details . . . 104
9.13 Event Log Format and Details . . . 111
9.14 Log Streaming . . . 114
9.15 Remote Log Type Encryption . . . 116
9.16 The Mail Command . . . 117
9.17 Diagnostic Tools . . . 118
9.18 Diagnostic Troubleshooting . . . 119

10 Notifications **123**
10.1 Warnings and Notifications . . . 123
10.2 Events and SNMP Messages . . . 125
10.3 SNMP Configuration and Queries . . . 127

11 SNMP **130**
11.1 Introduction to SNMP and MIB . . . 130
11.2 SNMP Traps . . . 131
11.3 Management Information Bases . . . 132
11.4 Trap Details . . . 136
11.5 VOSS Automate System Monitoring Traps . . . 163

12 Scheduling **170**
12.1 Scheduling . . . 170
12.2 Internal Report Schedules . . . 171

13 Backups **172**
13.1 Backups . . . 172
13.2 Backup and Restore in a Modular Architecture . . . 173
13.3 Backup Destinations . . . 173
13.4 Backup Passphrase . . . 174
13.5 Backup size considerations . . . 175

13.6 Create a Backup . . . 176
13.7 Restore a Backup in a Clustered Environment . . . 178
13.8 Create Space for a Backup or Restore . . . 178
13.9 Maintaining Backups . . . 178
13.10 Backup and Import to a New Environment . . . 179
13.11 VMware Snapshot Maintenance . . . 181
13.12 Backup on Azure . . . 182
13.13 Restoring a Backup on a New Environment . . . 182

14 System Security **186**
14.1 Introduction to system security . . . 186
14.2 Configuration Encrypted . . . 186
14.3 Backup Encrypted . . . 187
14.4 Application Install Files Encrypted . . . 187
14.5 File Integrity . . . 187
14.6 Protected Application Environments (Jails) . . . 188
14.7 Restricted User Shell . . . 188
14.8 User Security and Security Policy Management . . . 188
14.9 Creating Additional Users . . . 191
14.10 Creating and Managing SFTP Users . . . 192
14.11 Granting and revoking user rights . . . 193
14.12 Password Strength Rules . . . 194
14.13 SSH Login Fail Limit . . . 194
14.14 SSH Session Limit . . . 196
14.15 SSH key management . . . 197
14.16 SSH Algorithm Management . . . 198
14.17 Adding a Key for Automatic User Login . . . 199
14.18 Prevention of DOS Attacks . . . 199
14.19 Memory Dumps and Security . . . 200
14.20 Manage Read-Only Database Users . . . 200
14.21 System Intrusion Detection . . . 201

15 Network Security **204**
15.1 Network communications between nodes within the cluster . . . 204
15.2 Network communications external to the cluster . . . 206
15.3 Dynamic Firewall . . . 207
15.4 Web Certificate Setup Options . . . 208
15.5 Supported SSL Ciphers . . . 209
15.6 VOSS Automate Setup a Web Certificate . . . 214
15.7 Own Web Certificate Setup . . . 216
15.8 Web Certificate Expiration Notice . . . 217
15.9 Convert Web Certificates from P7B to PEM Format . . . 218
15.10 Web Certificate Commands . . . 218
15.11 Web Hosts Commands . . . 219
15.12 Web TLS Protocol Configuration . . . 221
15.13 Web TLS Cipher Management . . . 223
15.14 Network URI specification . . . 224
15.15 Web External Proxies . . . 224

16 High Availability and Disaster Recovery (DR) **226**
16.1 High Availability Overview . . . 226
16.2 Default HA and DR scenario . . . 226
16.3 HA and DR scenario with Cisco VMDC geo-redundancy architecture . . . 227
16.4 Configuring a HA System Platform on VMware . . . 227

16.5 DR Failover and Recovery Overview . . . 228
16.6 DR Failover and Recovery in a Unified Node Cluster Topology . . . 239
16.7 DR Failover and Recovery in a Modular Cluster Topology . . . 266

17 Troubleshooting **287**
17.1 Platform User Password Recovery Procedure . . . 287
17.2 'No Space Left on Device' Error . . . 288
17.3 Loss of the whole cluster and redeploying new servers . . . 289
17.4 Error Messages . . . 291

18 Appendices **310**
18.1 MIBs . . . 310
18.2 Data Export Types . . . 320
18.3 Command Examples . . . 361
18.4 SNMP Trap Examples . . . 368

Index **418**

1. What's New

1.1. Platform Guide: Release 25.2

- EKB-24140: Deprecate `screen` command and replace with `tmux`. See: *Using the tmux command*

  Added details on the "tmux" command replacing the "screen" command.
- EKB-24245: GUI unresponsive during security scan. See: *System Intrusion Detection*

  Added details on the new service available to manage SYN flood attacks

2. Overview

2.1. High-level Functions

The VOSS Automate platform is an Infrastructure As A Service layer (IAAS) built on top of Ubuntu Linux.

This platform layer supports the following high-level functions:

- Installation, upgrades
- Application and process manipulation
- Clustering of multiple nodes with High Availability (HA) and Disaster Recovery (DR) capabilities
- Backup creation and restore
- Scheduling of tasks
- Security implementation
- System diagnostics

Both the platform and application are designed as a loose collection of processes which can be deployed in a wide range of topologies. Individual nodes can be clustered and provisioned together to provide High Availability and Disaster Recovery.

2.1.1. Database Specification

- **Database Type**: MongoDB
- **Database Name**: VOSS
- **System Storage Type**: xfs (depends on OS)
- **System Storage Version**: v5 Filesystem (Ubuntu 20.04)
- **Storage Protocol**: WiredTiger
- **TNS**: N/A
- **Storage IP/URL**: host IP. e.g. 192.168.100.3 **
- **Storage Port**: 27020

3. The Command Line Interface

3.1. Overview

Maintenance is carried out from a platform user login application command line, either by SSH or from the VM console command line. The password is configured during installation and can be changed using **system password**. On initial login, the system displays a banner indicating the general system health.

A local home directory is available to the user and must be managed by the user with standard Unix commands:

- **ls**
- **cp**
- **mv**
- **rm**
- **less**
- **grep**

The user is not permitted to view directories or run commands outside the home directory.

During system maintenance, a specially configured rbash shell enables a set of commands to be executed.

The exact list of commands users can run is determined by the user's specific privileges and the specific setup of the machine. Different installed applications can add their own additional commands. The list of commands are displayed on login and can be redisplayed by typing the **help** command.

3.2. CLI Commands

Enter **help** (or **health**) to display the following screen::

```
System information as of Mon Mar 31 11:30:20 AM UTC 2025

 System load:  1.93               Processes:               453
 Usage of /:   52.5% of 19.20GB   Users logged in:         2
 Memory usage: 25%                IPv4 address for ens160: 192.168.100.3
 Swap usage:   0%
 You have mail.
 Last login: Mon Mar 31 11:30:21 UTC 2025 from 172.29.93.50 on pts/2
```

(continues on next page)

(continued from previous page)

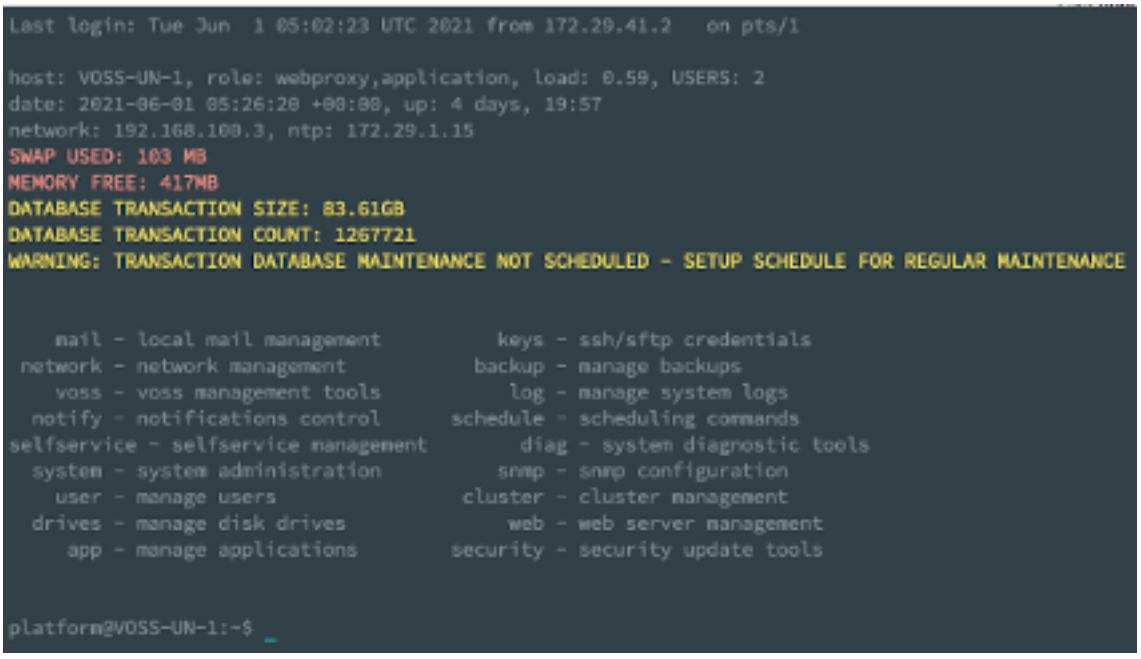
```
host: VOSS, role: webproxy,application,database
date: 2025-03-31 11:32:53 +00:00, up: 3:59
network: 192.168.100.3/27, ntp: 172.29.1.11
WARNING: TRANSACTION DATABASE MAINTENANCE NOT SCHEDULED - SETUP SCHEDULE FOR REGULAR␣
↪MAINTENANCE
WARNING: CURRENT LICENSE FOR 67ea4f0656ad41abb24a7b22 (Automate) IS SET TO EXPIRE ON␣
↪2025-04-07


        user - manage users                database - database management
        diag - system diagnostic tools       drives - manage disk drives
        mail - local mail management        network - network management
         app - manage applications              log - manage system logs
      system - system administration         cluster - cluster management
      backup - manage backups               insights - insights management
         web - web server management            keys - ssh/sftp credentials
      notify - notifications control            voss - voss management tools
 selfservice - selfservice management        schedule - scheduling commands
        snmp - snmp configuration             license - license management
```

Note: From release 21.1 onwards, **cluster** commands should be used in standalone ("cluster-of-one") deployments.

On web proxy nodes, the only cluster command you can run is **cluster prepnode**. Database commands are also not available on web proxy nodes.

Notices and critical warnings are shown in colors upon login.



This includes system license errors and warnings, for example:

```
WARNING: CURRENT LICENSE FOR <platform ID> IS SET TO EXPIRE ON <date>
```

For more details, refer to *Login Report*.

Entering any valid command name displays the usage parameters of that command. The **system** command help display is shown below::

```
platform@development:~$ system
USAGE:
------
system date                 - Display the system date and time
system download <url>       - Download a specific URL to media directory
system history              - Display a history of all executed UI commands
system keyboard <kbd-type> - Change the keyboard type (e.g. dvorak, us)
system mount                - Mount all removable media
system password             - Change the platform password
system provision            - Provision all the applications
system reboot               - Reboot the system
system root                 - Support administration via one-time-password
system shutdown             - Halt the system
system unmount              - Unmount all removable media
```

When commands are run on a cluster, a number of options are available to specify the nodes on which the commands can be run. In other words, there is a *<where>* clause: **cluster run <where>**. The clause can take:

- *role* - the role of the node: `application`, `database`, `webproxy`
- `all` - the entire cluster
- `notme` - all nodes except the one the command is run on

For example, **cluster run notme system shutdown** would issue the command to shut down all nodes except the one the command is run on.

Note: In a cluster, reboot and shutdown of the entire cluster should be done on each node and not with the **cluster run all** command - see: *Remote Execution in Clusters*.

Tab completion is available from the CLI for commands, parameters and partial filenames, for example:

```
$ log <Tab>
audit      collect    follow     list       merge      purge      send       sendnewer ␣
↪view

$ log audit <Tab>
locallog   remotelog  ssl        status

$ log view process/ngin <Tab>
$ log view process/nginx.proxy.log
```

See also *Using the tmux command*.

3.3. Clustering commands

Note: From release 21.1 onwards, a *standalone* topology is considered a *single node cluster (cluster-of-one)*. This means that commands such as the following should be run on a standalone topology:

- `cluster provision`
- `cluster list`
- `cluster status`
- `cluster maintenance-mode`

The following Command Line Interface console display shows the available commands for clustering.

```
cluster add <ip>                    - add a new node to join the existing cluster
cluster check <verbose>             - Display pre-upgrade readiness.
                                      For each node, test if ports
                                      are available, the time taken to connect, drive space
                                      percentage and lastly checks if NTP is running
cluster del <ip>                    - remove a node from the existing cluster
cluster job kill <pid>              - Kill a detached job <pid>
cluster job list                    - List detached jobs in the cluster
cluster job reconnect <pid>         - Reconnect to a detached job <pid>
cluster list                        - display the list of nodes associated with the cluster
cluster maintenance-mode            - Display the status, start or stop maintenance
         <start|stop|status>          mode across the cluster
cluster prepnode                    - Prepares the system so that it can be joined to a␣
↪cluster
cluster primary                     - Check if the system is considered the primary by␣
↪patch
                                      or delta install scripts
cluster primary role                - Check if the system is considered the
         <application|database>       'primary' in the given role
cluster provision                   - perform cluster-wide provisioning
         [datacentre <location>]
         [role <role>]
cluster run <where> <command>       - run the command on a particular host.
                                      <where> can either be a name prefix, ip, role, or
↪'all'
cluster status                      - display the status of the cluster
cluster upgrade <iso/url>           - upgrade all applications from iso image <iso-name>.
         [datacentre <location>]      <iso-name> can be a URL for upgrading
         [backup <location>]          from a remote server.
cluster where <application>         - determine on which nodes the application is installed
```

3.4. Using the tmux command

The tmux command is available to execute long-running commands (for example, when upgrading) in the background.

Note: From release 25.2, the screen command is no longer available on the Automate CLI - typing in screen on the CLI will return a message to use tmux instead.

The following commands require the running of tmux:

- **backup create <location-name>**
- **cluster provision**
- **cluster upgrade**
- **app template**
- **voss export type <args>**
- **voss export group <args>**
- **voss subscriber_data_export**

A message is displayed to indicate that tmux should be run first:

```
This is a potentially long-running command and should be executed in a tmux session
Run tmux and then execute the command again
```

The use of tmux is *not affected* by the use of the --force parameter with any of these commands.

The commands then run in a tmux session that can be reconnected. The standard tmux command parameters are available, in particular:

- List sessions:

```
tmux list-sessions
0: 1 windows (created Thu Apr 10 22:32:29 2025)
1: 1 windows (created Thu Apr 10 22:32:41 2025) (attached)
```

- Detach

```
Ctrl+b + d
```

- Exit (kill session)

```
Ctrl+d
```

- Attach to a session

```
tmux attach -t 0
```

- Scroll

```
Ctrl+b + [
```

- Scroll off

```
Esc
```

- Execute a command (example)

```
$ tmux -c 'voss export type license_audit'
Starting license_audit export, please wait...
Completed license_audit export, created vlf_[...]
```

To create a tmux log file in the /var/log/platform/tmux/ directory:

```
tmux -Log
```

For details, refer to the on-line help - also referenced below.

If the tmux session times out, you can obtain console output from the log file, for example:

$ sftp platform@<host>:media/<tmux-logfilename>.log

3.4.1. On-line help

The platform CLI offers an enhanced online help to the tmux command that also includes details and examples, as well as custom parameters to output the session to a log file.

The enhanced, full on-line help information available from the CLI is shown below:

```
platform@VOSS:~$ tmux -h
usage: tmux [-2CDlNuvV] [-c shell-command] [-f file] [-L socket-name]
            [-S socket-path] [-T features] [command [flags]]

COMMON COMMANDS
 attach-session (alias: attach) [-t target-session]    target-session is the number
                                                        from 'tmux ls'.
 list-commands  (alias: lscm) [command]                List the syntax of command or
                                                        - if omitted - of all commands
                                                        supported by tmux.
 list-sessions  (alias: ls)                            List all sessions managed by
                                                        the server.
 man                                                   Displays tmux manual.

COMMON KEY BINDINGS
 tmux may be controlled from an attached client by using a key combination of a
 prefix key, 'Ctrl-b', followed by a command key

 d           Detach the current client.
 :           Enter the tmux command prompt.

VOSS LOGGING ENHANCEMENTS:

 -Log                    Turn on output logging. Log folder /var/log/platform/tmux/
 -LogPrefix [prefix]     Prefix added to the log filename.
```

(continues on next page)

(continued from previous page)

```
LOG NAME PATTERN:

<[PREFIX]->tmuxlog-[TIMESTAMP]-[USER_ID]-S[SESSION_NAME]-W[WINDOW_ID]-P[PROCESS_ID].log
 I.e.
   tmuxlog-20250527120013-1401-0-W0-P1329199.log
   my-prefix-tmuxlog-20250527120013-1401-S0-W0-P1329199.log

Examples:
  1. Start a logged session with default name pattern:
  tmux -Log
      I.e. tmuxlog-20250527120013-1401-S0-W0-P1329199.log
  2. Run specific command with custom log prefix:
  tmux -LogPrefix my-log -c "my-long-running-command"
      I.e. my-log-tmuxlog-20250527120013-1401-S0-W0-P1329199.log
```

The `list-commands` parameter offers details on available commands, such as shown in the snippet below:

```
platform@VOSS:~$ tmux list-commands
attach-session (attach) [-dErx] [-c working-directory] [-f flags] [-t target-session]
bind-key (bind) [-nr] [-T key-table] [-N note] key [command [arguments]]
break-pane (breakp) [-abdP] [-F format] [-n window-name] [-s src-pane] [-t dst-window]
capture-pane (capturep) [-aCeJNpPq] [-b buffer-name] [-E end-line] [-S start-line] [-t␣
↪target-pane]
choose-buffer [-NrZ] [-F format] [-f filter] [-K key-format] [-O sort-order] [-t target-
↪pane] [template]
choose-client [-NrZ] [-F format] [-f filter] [-K key-format] [-O sort-order] [-t target-
↪pane] [template]
choose-tree [-GNrswZ] [-F format] [-f filter] [-K key-format] [-O sort-order] [-t target-
↪pane] [template]
clear-history (clearhist) [-t target-pane]
clock-mode [-t target-pane]
command-prompt [-1kiNTW] [-I inputs] [-p prompts] [-t target-client] [template]
[...]
```

3.5. System Specific Commands

3.5.1. VOSS Management Tools

The CLI (Command Line Interface) menu provides access to a number of commands specifically related to VOSS Automate.

In addition to the description of the commands available from the CLI **voss - voss management tools** menu, further details are provided for a selection of the commands. Note that some of the commands are used by developers only.

Important: On a Modular Architecture deployment, `voss` system specific commands can only be run on

application nodes and for Unified and Single Node architectures, *not* on database or web proxy type nodes.

The commands have been arranged into functional categories:

- Install commands: commands typically used during the install process.
- Database commands: commands that directly manage the database.
- Performance commands: commands to manage the system performance.
- System specific commands: general commands not specifically related to the categories above.

3.5.2. Install Commands

Important: On a Modular Architecture deployment, `voss` system specific commands can only be run on application nodes and for Unified and Single Node architectures, *not* on database or web proxy type nodes.

- **voss cleardown** - the command reinitialises the VOSS Automate database. It is usually run on a fresh installation and care should be taken with its use, as it deletes all system data.

  Note that this step may take some time. You can follow the process by running **log follow upgrade_db.log** or **log follow voss-deviceapi/app.log**.
- **voss migrate_summary_attributes <model_type>** - Migrates the summary attribute schema for instances of the specified model.

3.5.3. Database Commands

Important: On a Modular Architecture deployment, `voss` system specific commands can only be run on application nodes and for Unified and Single Node architectures, *not* on database or web proxy type nodes.

- **voss cleardown** - the command re-initializes the VOSS Automate database. It is usually run on a fresh installation and care should be taken with its use, as it deletes all system data.

  Note that this step may take some time. You can follow the process by running **log follow upgrade_db.log** or **log follow voss-deviceapi/app.log**.
- **voss db_collection_stats [all | <collection,collection,...>]** - Display detailed statistics of all the VOSS Automate databases, or of only a list of collections. Mongo database collections are similar to tables in relational databases.

  Refer to the example snippets below.

```
$ voss db_collection_stats


+--------------------+-----+--------+---------+-------------+------+----------+
| Collection         | Cnt | % Size | DB Size | AvgObj Size | Idx  | Idx Size |
+--------------------+-----+--------+---------+-------------+------+----------+
| VOSS.DATA_ACCESS.. |   8 |   0.0% |  92.88K |      11.61K |    2 |   15.97K |
| VOSS.DATA_APIVER.. |   1 |   0.0% | 496.00b |     496.00b |    2 |   15.97K |
| VOSS.DATA_APPLIC.. |   1 |   0.0% |   1.98K |       1.98K |    2 |   15.97K |
```

(continues on next page)

(continued from previous page)

```
+--------------------+-----+--------+---------+-------------+------+----------+

...

| VOSS.WORKER_QUEUE  |9508 |   9.4% | 117.76M |      12.68K |    9 |    4.34M |
+--------------------+-----+--------+---------+-------------+------+----------+
Total Documents: 144624
Total Data Size: 1.22G
Total Index Size: 73.31M
```

```
$ voss db_collection_stats DATA_RULEMODELHIERARCHYTYPE,SCHEMA

+--------------------+-----+--------+---------+--------------+------+------------+
| Collection         | Cnt | % Size | DB Size | AvOb Size    | Idx  | Idx Size   |
+--------------------+-----+--------+---------+--------------+------+------------+
| VOSS.DATA_RULEMO.. | 318 | 100.0% |  97.61K |   314.00b    |   2  |   96.00K   |
+--------------------+-----+--------+---------+--------------+------+------------+
| VOSS.SCHEMA        |  12 |  95.2% |   1.89M |      160.92K |    2 |     72.00K |
+--------------------+-----+--------+---------+--------------+------+------------+
Total Documents: 330
Total Data Size: 1.98M
Total Index Size: 168.00K
```

- **voss db_index_stats** - Display detailed statistics of the VOSS Automate database indices, including the five largest.

  Refer to the example below.

```
$ voss db_index_stats

Index Overview
+----------------------+--------------------------------+--------+------------+
| Collection           | Index                          | % Size | Index Size |
+----------------------+--------------------------------+--------+------------+
| VOSS.DATA_USER       | _id_                           |   0.1% |     44.00K |
| VOSS.DATA_USER       | username_1___hierarchy_1       |   0.1% |     52.00K |
| VOSS.RESOURCE        | _id_                           |   0.6% |    460.00K |
| VOSS.RESOURCE        | _search._i.v_1                 |  10.8% |      7.83M |
| VOSS.RESOURCE        | lock_1                         |   0.3% |    196.00K |

...


| VOSS.WORKER_QUEUE    | parent_transaction_id_1_..     |   0.0% |     12.00K |
+----------------------+--------------------------------+--------+------------+


Top 5 Largest Indexes
+----------------------+--------------------------------+--------+------------+
| Collection           | Index                          | % Size | Index Size |
+----------------------+--------------------------------+--------+------------+
```

(continues on next page)

(continued from previous page)

```
| VOSS.RESOURCE        | meta.model_type_1__search._i.. |  16.5% |     11.97M |
| VOSS.TRANSACTION_LOG | _id_                           |  10.8% |      7.85M |
| VOSS.RESOURCE        | _search._i.v_1                 |  10.8% |      7.83M |
| VOSS.TRANSACTION_LOG | _id.t_id_1_time_1              |   8.6% |      6.26M |
| VOSS.TRANSACTION     | submitted_time_-1_parent_1_a.. |   6.3% |      4.59M |
+----------------------+--------------------------------+--------+------------+

Total Documents: 743332
Total Data Size: 2.86G
Total Index Size: 72.56M
```

- **voss db_collection_cap_check TRANSACTION_LOG** - Returns the `TRANSACTION_LOG` database collection cap size in GB.
- **voss db_collection_cap TRANSACTION_LOG <10-50GB>** - Updates the `TRANSACTION_LOG` database collection to a cap size in GB provided as parameter.

  Important:

  1. The resize operation will impact the usage on the disk size allocated for the database (typically, 250GB is reserved upon installation). Consider a larger initial disk size allocation if a larger cap size is set.
  2. When resizing a collection, the existing collection is dropped and a new collection is created. Transactions in the existing collection can therefore not be re-run.

     Therefore, if you resize the transaction log, all existing entries will be flushed and are unrecoverable.
  3. The resize operation must be carried out during a maintenance window, since the `voss-deviceapi` service is restarted during this process.

  – cap size in GB parameter values: from 10GB up to 50GB

  This command to change the cap size is to enable larger providers to do this for operational reasons, for example for diagnostics and a longer transaction replay window.

3.5.4. Database Commands for Transaction Management

Important: On a Modular Architecture deployment, `voss` system specific commands can only be run on application nodes and for Unified and Single Node architectures, *not* on database or web proxy type nodes.

Commands are available to count, delete and export transactions from the database. All the commands take a `<days>` parameter that indicates transactions tasks are for transactions *older than* this number of days, counting from the current time.

For transaction **archive** and **delete** commands, the user is prompted on the command line to proceed or not.

If the transaction commands fail or are aborted:

- For *all* commands, a notification message is sent.
- For transaction **archive** and **delete** commands, the number of successful transaction deletes are applied.

- For transaction **export** commands, no export file is created.

If transactions are exported, the exported archive file will be in the `media/txn_archive/` directory. Available partition space is checked before any transaction export carried out and reported on. Estimated export sizes are based on average transaction sizes. If this directory contains files older than 30 days, an error notification is sent, with a message to remove these files.

The exported file is of the format *transaction_archive-YYYYMMDD_HHMMSS.gz*, where the UTC date stamp is the *current time*, for example:

```
media/txn_archive/transaction_archive-20190130_110122.gz
```

The exported `.gz` archive file contains a text file with lines of JSON formatted strings of the transactions.

- For suggestions on transaction archiving best practices, also refer to *Transaction archiving*.
- For more details on scheduling the transaction archiving to happen automatically, see *Enable Database Scheduling*.
- **voss transaction count <days>** - Count the number of transaction entries in the database that are older than the number of days specified. When running the commands interactively, this command is typically used before deleting or exporting.

  Example:

  ```
  $ voss transaction count 11
  167,582
  ```

- **voss transaction delete <days> [limit <number>]** - Delete transaction entries from the database that are older than the number of days specified, optionally limiting the number of *oldest* transactions to delete.

  The optional **[limit <number>]** parameter limits the number of transactions deleted and is typically used when a large number of transactions are older than the specified number of days (using **voss transaction count <days>**), which would impact the time to delete transactions. The parameter can then be used to manage the delete transaction time.

  The user is prompted to continue or not.

  Example:

  ```
  $ voss transaction delete 11
  You are about to delete transactions from the system. Do you wish to continue?y
  Available space:    27,219,492 KB
  Estimated space:        25,737 KB
  Deleting 167,582 transactions [################################################] 100%
  ```

- **voss transaction export <days>** - Create an archive file of the transaction entries in the database that are older than the number of days specified. No entries are deleted from the database. The export file has the format indicated above.

  Example:

  ```
  $ voss transaction export 11
  Available space:    27,219,492 KB
  Estimated space:        25,737 KB
  Exporting 167,582 transactions [################################################] 100%
  ```

- **voss transaction archive <days>** - First create an archive file of and then delete transaction entries in the database that are older than the number of days specified. This command therefore combines two commands: **voss transaction export <days>** and **voss transaction delete <days>**. The export file has the format indicated above.

  The user is prompted to continue or not.

  Example:

```
$ voss transaction archive 11
You are about to delete transactions from the system. Do you wish to continue?y
Available space:    27,219,492 KB
Estimated space:        25,737 KB
Exporting 167,582 transactions [################################################] 100
↪%
     Used space:       24,767 KB
Archive saved to media/txn_archive/transaction_archive-20190130_110122.gz
Deleting 167,582 transactions [################################################] 100
↪%
```

3.5.5. Performance Commands

Important: On a Modular Architecture deployment, `voss` system specific commands can only be run on application nodes and for Unified and Single Node architectures, *not* on database or web proxy type nodes.

- **voss throttle-rates** - A command with parameters to set, show and disable the API request rate for an interface or for any user. The command may be used to manage API request overload.
  - Throttle rates apply to each unified node in a cluster.
  - Use **voss throttle-rates help** to see command parameters and options.
  - Please contact support before changing any settings for throttle rates. Great care should be taken when adjusting throttle rates, as a change can have a significant impact on system performance and behavior

  By default, the following interface throttle rates apply:

  - administration: disabled
  - selfservice: 300 req/min
  - per user: 20 req/sec

  If the throttle rates are exceeded, the API returns the HTTP status code and message:

  `Error 429:  Too Many Requests`

  - To set throttling:

    voss throttle-rates type <administration|selfservice|user> requests <number of requests> unit <time unit>

    * The requests parameter is defined as an integer which is the number of requests per unit.
    * The time unit can be second (`sec`, `s`) or minute (`min`, `m`).
    * Command output will show the interface and configuration change and prompt for a service restart.

Examples:

```
$ voss throttle-rates type administration requests 10 unit min
    Administration:
        Current Configuration: Disabled
        New Configuration: 10/min
    Self Service:
        Current Configuration: Disabled
    User:
        Current Configuration: Disabled

  An application restart is required for this change to take effect, e.g.:
  $ cluster run application app start voss-deviceapi:voss-wsgi


$ voss throttle-rates type selfservice requests 20 unit sec
    Administration:
        Current Configuration: 10/min
        Current Rates: 0/min
    Self Service:
        Current Configuration: Disabled
        New Configuration: 20/sec
    User:
        Current Configuration: Disabled

  An application restart is required for this change to take effect, e.g.:
  $ cluster run application app start voss-deviceapi:voss-wsgi


$ voss throttle-rates type user requests 30 unit min
    Administration:
        Current Configuration: 10/min
        Current Rates: 0/min
    Self Service:
        Current Configuration: 20/sec
        Current Rates: 0/sec
    User:
        Current Configuration: Disabled
        New Configuration: 30/min

  An application restart is required for this change to take effect, e.g.:
  $ cluster run application app start voss-deviceapi:voss-wsgi
```

Note:

* If the command is used *without* parameters, the user will be prompted to enter them. Press **Ctrl-C** to exit this interactive mode.
* The user throttle rate can be limited by an interface throttle rate.

– To show current throttling continuously:

voss throttle-rates list-refresh

The current request values are updated until the command is canceled with **Ctrl-C**.

For example:

```
$ voss throttle-rates list-refresh
    Administration:
        Current Configuration: 10/min
        Current Rates: 0/min
    Self Service:
        Current Configuration: 20/sec
        Current Rates: 0/sec
    User:
        Current Configuration: 30/min
Refreshing Ctrl-C to exit..
```

– To show current throttling and exit:

voss throttle-rates list

The current request values are updated and shown. The command then exits.

For example, to list when enabled:

```
$ voss throttle-rates list
    Administration:
        Current Configuration: 10/min
        Current Rates: 0/min
    Self Service:
        Current Configuration: 20/sec
        Current Rates: 0/sec
    User:
        Current Configuration: 30/min
```

For example, to list when throttling is disabled:

```
$ voss throttle-rates list
    Administration:
        Current Configuration: Disabled
    Self Service:
        Current Configuration: Disabled
    User:
        Current Configuration: Disabled
```

– To disable throttling:

voss throttle-rates disable

For example:

```
$ voss throttle-rates disable
    Administration:
        Current Configuration: 10/min
        Current Rates: 0/min
        New Configuration: Disabled
    Self Service:
        Current Configuration: 20/sec
        Current Rates: 0/sec
        New Configuration: Disabled
```

(continues on next page)

(continued from previous page)

```
  User:
      Current Configuration: 30/min
      New Configuration: Disabled

An application restart is required for this change to take effect, e.g.:
$ cluster run application app start voss-deviceapi:voss-wsgi
```

Throttle rates are disabled by default. To restore any rates that were disabled, the throttle rates need to be set again.

- **voss session-limits** - A command with parameters to set, show and disable the number of sessions - based on interface and customer. In other words, for each interface: Administration or Self-service, a Global and a Per Customer Hierarchy session limit can be set. The number of concurrent login sessions per user is determined by the user's Credential Policy. Highest level (above provider administrators) administrator logins do not affect and are not affected by session limits.

Note: For administrators configured with multiple user roles, i.e. users that have a user type "Admin + End User":

  - Each SSO login will consume a session counter from the global/customer administration session limits.
  - Each non-SSO admin login will consume a session counter from the global/customer administration session limits.
  - Even if a SSO/non-SSO login admin switches to the Self-service interface, the session counter will remain consumed on the global/customer administration session limits.

Use **voss session-limits help** to see command parameters and options. Please contact support before changing any settings for session limits.

The command line output of the command to adjust the limits show the current and new values and prompt to restart the `voss-deviceapi:voss-wsgi` service before the limit is changed. For clusters, the session limit is set per cluster, requiring a cluster wide service restart.

By default, the following limits apply:

  - global administration: 200
  - global selfservice: 20000
  - per customer administration : 10
  - per customer selfservice : 1000

The global session limit would always be set to a larger value than a customer hierarchy limit.

A session is active until it expires or the user logs out. If the session limits are exceeded, the API returns the HTTP status code and message:

```
Error 503:  "Login is currently disabled due to a temporary overload. Please try
again later.
```

  - To set session limits:

    voss session-limits type <customer|global> interface <administration|selfservice> limit <number>

* If the command is used *without* parameters, the user will be prompted to enter them. Press **Ctrl-C** to exit this interactive mode.
* The type is the hierarchy to which the limit applies: `global` or `customer`
* The interface is the user interface to which the limit applies: `Administration` or `Self-service`.
* The limit number is an integer value for the number of sessions.

Examples:

```
$ voss session-limits type customer interface administration limit 100
  Administration:
        Global:
            Current Limit: 200
            Current Sessions: 1
        Per Customer Hierarchy:
            Current Limit: 10
            New Limit: 100
    Self Service:
        Global:
            Current Limit: 20000
            Current Sessions: 0
        Per Customer Hierarchy:
            Current Limit: 1000

  An application restart is required for this change to take effect, e.g.:
  $ cluster run application app start voss-deviceapi:voss-wsgi
```

– To list session limits:

voss session-limits <list|list-refresh>

The current session values are shown. if the `list-refresh` option is used, the values are updated every second until the command is canceled with **Ctrl-C**.

Examples for two customers “GenCorp” and “VS-Corp”:

```
$ voss session-limits list-refresh
    Administration:
        Global:
            Current Limit: 200
            Current Sessions: 2
        Per Customer Hierarchy:
            Current Limit: 10
            GenCorp Current Sessions: 0
            VS-Corp Current Sessions: 1
    Self Service:
        Global:
            Current Limit: 20000
            Current Sessions: 0
        Per Customer Hierarchy:
            Current Limit: 1000
            GenCorp Current Sessions: 0
            VS-Corp Current Sessions: 0
```

(continues on next page)

(continued from previous page)

```
    Refreshing, Ctrl-C to exit...
```

– To disable session limits:

voss session-limits disable

* This setting *removes all* session limit settings.
* Recall that for the command to take effect, *all* nodes in the cluster need the service to be restarted.
* To restore any settings that were disabled, they need to be set again.

Example:

```
$ voss session-limits disable
  Administration:
      Global:
          Current Limit: 200
          Current Sessions: 1
          New Limit: Disabled
      Per Customer Hierarchy:
          Current Limit: 10
          New Limit: Disabled
  Self Service:
      Global:
          Current Limit: 20000
          Current Sessions: 0
          New Limit: Disabled
      Per Customer Hierarchy:
          Current Limit: 1000
          New Limit: Disabled

  An application restart is required for this change to take effect, e.g.:
  $ cluster run application app start voss-deviceapi:voss-wsgi
```

- **voss workers** - By default, 30 is the maximum number of parent transactions per unified node from the queue that will be processed at once. Use the command **voss workers <number>** to modify this value.

  This command restarts the `voss-queue` services.

- **voss queues <number>** - Set the number of queue processes

  Increasing the number of queues that can run in parallel. This command restarts the `voss-queue` services.

  voss queues - Get the number of queue processes

  Important: It is strongly advised that VOSS Support is consulted before making changes to the number of queue processes.

  For details, see the Configurable Number of Queue Processes topic in the Best Practices Guide.

Transactions resulting from a number of system components are by default carried out with a low priority that is currently limited to 50% of the maximum allowed parent transactions to be processed at once (the default is 30 per unified node). The system components are:

- Bulk loaders
- Data Sync (CUCM, Unity, LDAP)

Since these transactions may place too much load on the system during business hours when users are using the system for other activities, commands are available to change the percentage of the maximum allowed number of transactions these low priority tasks can use at peak and off-peak times.

- Percentage options are: 20%, 50% and 80%
- Defaults are as follows:
  - Off-peak percentage is 50% of the maximum number of parent transactions per unified node from the queue that will be processed at once. This means if no off-peak percentage is set manually, this value will be the default.
  - Off-peak start and end times on weekdays (Mon - Fri) are 00:00 - 23:59
  - Off-peak start and end times on weekends (Sat - Sun) are 00:00 - 23:59

  In other words, the default is 50% for all hours of the week.

The command that is available to change peak and off-peak times and percentages, is **voss worker low_priority_schedule** and takes a number of parameters.

Without parameters, the command shows the current times and percentages, for example:

```
$ voss worker low_priority_schedule

       Off-peak Time:
           Weekday:
               Start Time: 00:00
               End Time: 23:59
           Weekend:
               Start Time: 00:00
               End Time: 23:59
       Off-peak Percentage: 50
       Peak Percentage: 50
```

The following list shows command parameters to set times and percentages for the low priority schedule. Note that if an end time is set to a value that falls into another schedule, the start time of the other schedule will apply.

- **voss worker low_priority_schedule off_peak_time weekday HH:MM HH:MM**
  - Set the weekday schedule off-peak start and end time. The command prompts for a queue service restart.

    The command supports nodes in different time zones. When times are set on a node in a cluster, the equivalent schedule times are set on nodes in other time zones.
- **voss worker low_priority_schedule off_peak_time weekend HH:MM HH:MM**
  - Set the weekend schedule off-peak start and end time. The command prompts for a queue service restart.

    The command supports nodes in different time zones: when times are set on a node in a cluster, the equivalent schedule times are set on nodes in other time zones.

- **voss worker low_priority_schedule peak_percentage [20|50|80]**
  - Set the percentage of maximum allowed workers for low priority tasks to either 20%, 50%, or 80% during the peak period - the period outside any defined off-peak period. The command prompts for a queue service restart.
- **voss worker low_priority_schedule off_peak_percentage [20|50|80]**
  - Set the percentage of maximum allowed workers for low priority tasks to either 20%, 50%, or 80% at the defined off-peak period. The command prompts for a queue service restart.

For example, the sequence of commands:

```
voss worker low_priority_schedule off_peak_time weekday 23:00 04:00
voss worker low_priority_schedule off_peak_time weekend 13:00 10:00
voss worker low_priority_schedule off_peak_percentage 80
voss worker low_priority_schedule peak_percentage 20
```

will result in a low priority schedules that uses a higher percentage (80%) of the maximum workers on weekdays between 23:00 and 04:00 and on weekends between 13:00 and 10:00, and only 20% during the rest of the time - the peak time.

3.5.6. System Specific Commands

Important: The `voss` system specific commands can *not* be run on:

- web proxy nodes

The `voss` system specific commands can *only* be run on:

- application nodes on a Modular Architecture deployment,
- unified nodes on Unified and Single Node architectures

- **voss finalize_transaction <id>[,<id> . . .]** - If a transaction status incorrectly shows as processing *after a service restart* - for example where a service has stopped - use this command to mark it as finalized.

  Example:

  ```
  platform@VOSS:~$ voss finalize_transaction 50276
  This will mark the given transaction(s) as failed. Do you wish to continue? y
  Transaction state changed to failed for [50276].
  ```

  More than one transaction can be marked to be finalized - enter these as a comma-separated list of transaction sequence IDs.

  If a transaction in a 'Processing' state has child transactions in a 'Processing' state, the tool will attempt to finalize these as well. The output will indicate any transaction IDs that could not be finalized and that require further user action.

  Important:

  - The **voss finalize_transaction** command should never be run for transactions that are still processing. It should *only* be used for transactions that are still in a 'Processing' state after a service restart or other event influencing the `voss-deviceapi` queue and `wsgi` processes.

  - The **voss finalize_transaction** command does *not* cancel or stop a running or queued transaction, but only sets the **Status** flag to 'Failed'.

- **voss migrate_summary_attributes <model_type>** - Migrates the summary attribute schema for instances of the specified model.

  For example:

```
voss migrate_summary_attributes data/InternalNumberInventory
```

- **voss reset_device_concurrency [all | device <device_name> | –show_locks]** - Call Manager can handle eight concurrent connections. VOSS Automate monitors the number of connections it continuously makes and removes or adds to the number as connections are made and closed. For debugging purposes, this "tracking number" can be reset.

  If the concurrency remains at the maximum for more than 10 minutes, it is automatically reset to zero and a log error message `Resetting stale Device Concurrency..` (containing details) is displayed. The parameters are as follows:

  - **all** - this is the default parameter if no parameter is provided. Concurrent connections on *all* devices are reset.
  - **device <device_name>** - reset the concurrent connections on a specific device with `<device_name>`, for example **voss reset_device_concurrency device 192.168.100.15**
  - **–show_locks** - For all devices, show device type, device name , concurrent connections and max current concurrent connections. For example:

```
$ voss reset_device_concurrency --show_locks

INFO:__main__:[CUCM - 192.168.100.15:8443] [0/8]
INFO:__main__:[CUC - 192.168.100.12:8443] [0/8]
```

- **voss clear_device_pending_changes <device_name | all>** - Since Unified CM version 10.0, a Change Notification Feature is available that stores changes to device objects in a cache. The VOSS Automate application service called `voss-deviceapi::voss-cnf_collector` collects these changes from *all* the Unified CM devices as they are added, manages and stores the changes in a data collection.

  This collection can then be used to update the system by means of a Data Sync option on the GUI. An additional tool on the GUI displays the status and manages the collection on a specified device and also allows for the polling interval between collections to be configured. Each device keeps the last time that a collection process ran on it, so that a new collection will only be run on it once its interval has expired.

  The **voss clear_device_pending_changes** command clears all pending changes by the Change Notification Collector for a particular device or for all devices.

  To clear the collection of pending changes from a single device, the command and output is for example:

```
$ voss reset_device_pending_changes 10.120.10.190
This will clear all pending changes.
Do you wish to continue? yes
```

  The status of changes collected from a device can be checked from the GUI.

- **voss set_debug <level[0/1]>** - By default, the level is “level0”, so no debug information is present in the logs.

  Setting the value to “level1” is not supported and will for example result in performance degradation.

- **voss unlock_sysadmin_account** - Unlocks and forces a password reset on the system administrator account. The system administrator is prompted to enter a new password and is then prompted to verify the new password.

- **voss update_device_schemas <schema models>** - Regenerate device model schemas in the database and in the device schemas fixtures file. No data is lost, so this can be done on a production system, although the system should not be used while this is happening. Parameters can be passed through (indicated as “<schema models>”). The available parameters are shown by using the “–help” parameter. Used by developers.

- **voss post-upgrade-migrations** - Schedule a transaction to execute long running data migrations after an upgrade.

  Data migrations that are not critical to system operation can have significant execution time at scale. These need to be performed after the primary upgrade, allowing the migration to proceed whilst the system is in use - thereby limiting upgrade windows.

  – This command is a mandatory step after an upgrade.

  – The command only needs to be run on a single node of a cluster.

  – The command will display progress information of the migration transaction until the transaction concludes, or until the user exits the command using Ctrl-c.

  – Transaction progress can also be followed on the Transaction GUI.

  – If the transaction is cancelled on the GUI or interrupted by a system or queue restart, the command can be run again to re-queue a migration transaction, which will resume the migration process.

  – Console examples are shown below:

```
$ voss post-upgrade-migrations
Queueing post upgrade migrations, please wait...
Post upgrade migrations transaction queued, ID 6.
Transaction progress(ctrl-c to exit, transaction will continue to execute):
-Post upgrade migrations:   0%|              | 0/1 [00:00<?]
--Migrating objects in TRANSACTION colle...:  27%|#######   | 53814/200000␣
↪[00:30<01:21]
^C
Aborting progress monitoring, transaction ID 6 is continuing execution, exiting..
↪.
```

  Subsequent re-execution with the same previously queued transaction still executing - progress display is resumed and displayed until the transaction concludes:

```
$ voss post-upgrade-migrations
Existing post upgrade migrations transaction found, not requeuing.
Transaction progress(ctrl-c to exit, transaction will continue to execute):
-Post upgrade migrations: 100%|####################| 1/1 [01:40<00:00]
--Migrating objects in TRANSACTION colle...: 100%|##########| 200000/200000␣
↪[01:40<00:00]

Post upgrade migrations complete, exiting...
```

3.5.7. Transaction archiving

The following are considerations when determining the frequency of the transaction archiving schedule to set up on the system. If a schedule is not set up for transaction archiving, system Alerts will be raised as well as a warning on the platform CLI login:

`TRANSACTION DATABASE MAINTENANCE NOT SCHEDULED`

- Run `voss transaction count <days>` on your system to inspect the number of transactions during a given period to determine your usage metrics.

  Refer to the *Database Commands for Transaction Management* topic in the Platform Guide for details on transaction archive command use and scheduling:

  - **voss transaction delete <days>**
  - **voss transaction export <days>**
  - **voss transaction archive <days>**

- Business policies - company policies may drive your choices: the immediate access to transaction logs for a period of time, security policy on data/audit retention, and so on.

  Note: The transaction archive process does mean the logs are not lost, just that they are not immediately accessible in the administrator graphical interface for searching.

- You can also set up system monitoring thresholds so that you receive alerts via the GUI and SNMP if the threshold is exceeded - which might indicate you need to review the archive schedule to increase how frequently is runs.

  See the *SNMP* and *Automate System Monitoring Traps* topics in the Platform Guide.

3.5.8. Product Licensing

As part of the VOSS Automate 21.4 release, license enforcement will be enabled on all Automate deployments, including production and lab environments. This will result in the requirement for license keys to be obtained and installed on all platforms (within 7 days from the date of upgrade) to ensure no service disruptions are experienced.

Note:

- For an overview of licensing and functionality available in the GUI, refer to the Product License Management topic in the Core Feature Guide.
- When a license has expired, GUI login is not available. Log in on the platform CLI and update a license as indicated below.
- Customers can use an upgraded or installed platform for a grace period of 7 days in order to license the system.

The Licensing Service

A license service is installed during installation/upgrade from release 21.4 onwards.

Run `app status` to verify the presence and status of the `license-service` application, for example:

```
platform@VOSS:~$ app status
selfservice v21.4.0 (2022-12-03 15:16)
   |-node                  running
snmp v21.4.0 (2022-12-03 15:23)
   |-daemon                running (completed)
   |-traps                 running (completed)
license-service v21.4.0 (2022-12-03 15:11)
   |-license-service       running
cluster v21.4.0 (2022-12-03 15:21)
template_runner v21.4.0 (2022-12-03 15:23)
[...]
```

Note:

- An hourly schedule is run for VOSS Automate to process license service information.
- The `license-service` cannot be stopped by running a command from the command line.

The CLI login screen also shows license information from 120 days before expiry.

The message levels are according to remaining number of days on the license, as follows:

- INFO: 90 < remaining days <= 120
- WARNING: 0 < remaining days <= 90
- ERROR: 0 >= remaining days

Refer to the Product License Management topic in the Core Feature Guide for GUI Alerts intervals.

Otherwise, no information is displayed on login. The `license list` command can however be used to inspect the license status. See: *Login Report*.

Note:

- License commands and status messages are only available on:
  - unified nodes in a Unified Node Topology
  - application nodes in a Modular Cluster Topology

Loading a License on the CLI

1. Obtain the license token from VOSS.
2. Log in on the platform CLI.
3. If required, the token can be copied to a file and uploaded to the `/media` directory.

   Otherwise, have the token available in the copy buffer.

   a. If copied to a file, use the command below to add the token file - using the `file` parameter:

   ```
   license add file media/<token_file>
   ```

   b. If copied to the copy buffer, use the command below to paste the token - using the `token` parameter:

   ```
   license add token [<token string>]
   ```

   If a license token file is successfully added, the `license check` command is automatically run to process any changed licensing status or updated license token. Output example:

   ```
   Current license status "Licensed" remains unchanged.
   Status was set on 2024-05-22T10:20:07
   ```

4. To process any changed licensing status or updated license token manually, use the command:

   ```
   license check
   ```

   (the equivalent command: `voss check-license` is also still available to ensure legacy support)

5. To view the currently loaded license token information, use the command:

   ```
   license list
   ```

Check License Details

At any stage, the details of license tokens can be viewed.

- Use the `license list` command, for example:

```
platform@VOSS:~$ license list
    6009914029c384002787ef2e:
        Audience: Automate
        Expired: false
        Expiry: 2022-11-07T10:20:09Z
        Platform Description: N/A
        Token ID: d2ba2d70-fc84-47af-a57e-5cb9c39878d8
```

Refresh License

A command is available to verify a VOSS Automate license status, for example, after updating the license token.

The command: `voss check-license` uses the running `license-service` application information to refresh the license information of the VOSS Automate deployment.

For example, if the command is run from the CLI:

```
platform@VOSS:~$ voss check-license
Current license status "Licensed" remains unchanged.
Status was set on 2022-11-01T13:46:07
```

then the time of that command will update the **License Last Checked** time stamp on the **Manage License** GUI form (accessible by the `sysadmin` user) to that time. In this example, the system was already licensed and this check reflects this license status as "unchanged".

Renewing a License

When a license has expired, contact VOSS to renew the license. Provide the platform ID (UUID) which is shown when running `license list`.

3.6. System Metrics

3.6.1. Report Transaction Commands

Important: On a Modular Architecture deployment, `voss` system specific commands can only be run on application nodes and for Unified and Single Node architectures, *not* on database or web proxy type nodes.

VOSS Automate provides a set of utilities available to provide transaction metrics. The commands are typically used to inspect and monitor transaction performance, for example in the case where transaction performance issues are encountered.

Note: For clusters, the number of workers are set to be the total default for 4 unified nodes (4*30 =120).

Profile

A command is available that aggregates the transactions performed during the given date-time range by grouping them by model type and sorting them by duration.

A **Pct** column is also available to indicate the percentage of the interval time used by the transaction.

The command and parameters are:

- **voss report transaction profile [OPTIONS] START_TIME END_TIME**
- `START_TIME` - The date-time value from which to start the sample collection.

- END_TIME - The date-time value from which to start the sample collection. Valid formats: *'%Y-%m-%d_%H:%M:%S', '%Y-%m-%d_%H:%M', '%Y-%m-%d'*

Example:

```
platform@VOSS:~$ voss report transaction profile 2019-05-06_16:39 2019-05-06_23:00
                       Entity  Operation      Calls   RespTime      QTime    SvcTime  ␣
↪    Pct
              tool/DataImport    execute      11.00    3128.24       1.11    3127.13  ␣
↪  99.95
                    data/User     update       2.00       0.86       0.09       0.77  ␣
↪   0.03
                data/Schedule     update       1.00       0.28       0.18       0.10  ␣
↪   0.01
            data/PasswordReset     create       2.00       0.17       0.10       0.08  ␣
↪   0.01
                data/Schedule    Execute       1.00       0.15       0.00       0.14  ␣
↪   0.00
```

Throughput

Given a number of output rows to display and duration in seconds between samples, a command is available to display the information about transaction throughput during the selected sample periods.

Throughput is the projected count per minute based on the sample interval and transaction count.

The command and parameters are:

- **voss report transaction throughput [OPTIONS] START_TIME INTERVALS DURATION**
- START_TIME - The date-time value from which to start the sample collection. Valid formats: *'%Y-%m-%d_%H:%M:%S', '%Y-%m-%d_%H:%M', '%Y-%m-%d'*
- INTERVALS - Specifies the number of samples to display.
- DURATION - The duration of the samples in seconds, for example every 30 seconds or every 60 seconds.

The command output shows parent and child transaction data in columns:

- **Count**: Transaction count
- **Utilisation**: ratio of total workers (120) used: ((**Throughput** * **RespTime**) / 120) * 100

  Example of parent transactions:

  formula: `( (Throughput * RespTime) / 120 ) * 100`

  calculation: `(2.00 * 4.86 / 120) * 100 = 9.71`
- **Qtime**: ratio: (total queue time for transactions at time stamp / transaction count at the time stamp). Else, zero if no transactions.
- **SvcTime**: ratio: (total service time for transactions at time stamp / transaction count at the time stamp). Else, zero if no transactions.
- **RespTime**: calculated as total transaction time (**Qtime** + **SvcTime**). Else, zero if no transactions.
- **Throughput**: ratio: (transaction count at the time stamp / minutes (end_time - start_time)). Else, zero if no transactions.

  Example of child transactions where DURATION is 30 seconds (0.5 minutes):

21 child transactions / 0.5 minutes = 42.00

- **Workers**: calculated as (**Throughput** * **RespTime**)

Example command output:

```
platform@VOSS:~$ voss report transaction throughput 2019-05-06_16:39 2 30
                                    Parent Transactions                      ␣
↪              Child Transactions
TS                    Count   Utilisation    Workers      Qtime    SvcTime  ␣
↪RespTime Throughput  Count      Qtime    SvcTime   RespTime Throughput

2019-05-06_16:39:00       0       0.00       0.00       0.00       0.00       0.
↪00       0.00    21       0.07       0.86       0.93      42.00
2019-05-06_16:39:30       0       0.00       0.00       0.00       0.00       0.
↪00       0.00    80       0.07       0.14       0.22     160.00
```

```
platform@VOSS:~$ voss report transaction throughput 2019-05-06_16:42 2 30
                                    Parent Transactions                      ␣
↪              Child Transactions
TS                    Count   Utilisation    Workers      Qtime    SvcTime  ␣
↪RespTime Throughput  Count      Qtime    SvcTime   RespTime Throughput

2019-05-06_16:42:00       1       9.71       8.10       0.07       4.79       4.
↪86       2.00    98       0.07       0.05       0.13     196.00
2019-05-06_16:42:30       2    3421.50    2851.25       0.12     855.26     855.
↪37       4.00    34       0.07       0.21       0.28      68.00
```

Worker-Usage

A command is available to show the active and queued transactions at particular intervals from a given start time and then to provide a *utilisation* value for the transaction, relative to a worker value.

The command and parameters are:

- **voss report transaction worker-usage [OPTIONS] START_TIME INTERVALS DURATION**
- `START_TIME` - The date-time value from which to start the sample collection. Valid formats: *'%Y-%m-%d_%H:%M:%S', '%Y-%m-%d_%H:%M', '%Y-%m-%d'*
- `INTERVALS` - Number of report entries to list.
- `DURATION` - Number of seconds between each report entry.

The command output shows parent and child transaction data in columns:

- **Utilisation**: ratio: (active parents / total workers (=120)) * 100
- **Zombie**: number of transactions that have not been updated (changed from Queued to Active) for 60 minutes

Example output

```
platform@VOSS:~$ voss report transaction worker-usage 2019-05-06_16:39 50 30
REPORT:
Timestamp           Utilisation     QueuedParents    ActiveParents    QueuedChildren  ␣
```

(continues on next page)

(continued from previous page)

```
↪ActiveChildren   Zombies
2019-05-06_16:39:00        0.83                0                1                0      ␣
↪          1          0
2019-05-06_16:39:30        0.83                0                1                0      ␣
↪          1          0
2019-05-06_16:40:00        0.83                0                1                0      ␣
↪          0          0
2019-05-06_16:40:30        0.83                0                1                0      ␣
↪          0          0
2019-05-06_16:41:00        0.83                0                1                0      ␣
↪          1          0
2019-05-06_16:41:30        0.83                0                1                1      ␣
↪          0          0
2019-05-06_16:42:00        0.83                0                1                0      ␣
↪          0          0
2019-05-06_16:42:30        0.00                0                0                0      ␣
↪          0          0
2019-05-06_16:43:00        0.83                0                1                0      ␣
↪          1          0
2019-05-06_16:43:30        0.83                0                1                0      ␣
↪          0          0
2019-05-06_16:44:00        0.83                0                1                0      ␣
↪          0          0
```

Current-Usage

A command is available to show a table of transactions grouped by:

- Processing and Queued transactions
- Parent and child transactions
- Node name on which the transaction is running

  Queued transactions will be shown with a node name of `default` because it is not known on which node they will be executed.
- Transaction priority (High, Medium, Low)

The report is useful to verify that all unified nodes are correctly processing transactions.

The command is:

- **voss report transaction current-usage**

Example output from a single node:

```
platform@VOSS:~$ voss report transaction current-usage

REPORT:
parent                     |   Processing transactions    |      Queued transactions    ␣
↪|
                           |          Priority            |          Priority           ␣
↪|
                      Node|     High    Medium       Low |      High    Medium      ␣
```

(continues on next page)

(continued from previous page)

```
↪Low|
          node1-voss-queue|        1         0         0 |         0         0        ␣
↪0|

child                     |   Processing transactions    |      Queued transactions     ␣
↪|
                          |          Priority            |          Priority            ␣
↪|
                     Node|     High    Medium       Low |      High    Medium       ␣
↪Low|
          node1-voss-queue|        0         0         0 |         0         0        ␣
↪0|
```

3.6.2. Report API Commands

Important: On a Modular Architecture deployment, `voss` system specific commands can only be run on application nodes and for Unified and Single Node architectures, *not* on database or web proxy type nodes.

VOSS Automate provides a set of utilities available to provide API request metrics. The commands are typically used to inspect and monitor API request performance over a defined period.

Profile

A command is available that aggregates the API requests performed during a given date-time range by grouping them by model type and sorting them by duration.

The command and parameters are:

- **voss report api profile [OPTIONS] START_TIME END_TIME**
- `START_TIME` - The date-time value from which to start the sample collection.
- `END_TIME` - The date-time value from which to start the sample collection. Valid formats: *'%Y-%m-%d_%H:%M:%S', '%Y-%m-%d_%H:%M', '%Y-%m-%d'*
- Options:
  - `--path TEXT` API log file path (default is `nginx/access.log`)
  - `--limit INTEGER` Limits the number of profile lines to display. If limited, the last row is listed as `Other`

The command output shows Request_URI count data in columns.

Column headers:

- **90th-PCTL**: 90th percentile time value of the count
- **CV**: Coefficient of variation of the count

Example:

```
platform@VOSS:~$ voss report api profile --limit 6 2019-05-06_15:49 2019-05-06_16:00
     Count       TotalTime   AverageTime      Percentage      90th-PCTL              CV ␣
↪             Method  Request-URI
       435          199.97          0.46          72.11%           0.69          34.66% ␣
↪               PUT  /api/data/DataModel
        69           64.92          0.94          23.41%           1.32          29.76% ␣
↪             LIST  /api/data/DataModel
        44           10.94          0.25           3.95%           0.33          38.26% ␣
↪             LIST  /api/data/Migration
         1            0.12          0.12           0.04%           0.00           0.00% ␣
↪             LIST  /api/data/Cuccx
         1            0.09          0.09           0.03%           0.00           0.00% ␣
↪             LIST  /api/data/Smtp
         1            0.07          0.07           0.02%           0.00           0.00% ␣
↪             LIST  /api/data/Package
        26            1.19          0.00           0.43%           0.00           0.00% ␣
↪                    Other
```

Throughput

A command is available to show the number of API calls at particular intervals from a given start time and then to provide a *throughput* value for the transaction, in other words number of requests per time interval.

The command and parameters are:

- **voss report api throughput [OPTIONS] START_TIME INTERVALS DURATION**
- `START_TIME` - The date-time value from which to start the API call collection. Valid formats: *'%Y-%m-%d_%H:%M:%S', '%Y-%m-%d_%H:%M', '%Y-%m-%d'*
- `INTERVALS` - Number of report entries to list.
- `DURATION` - Number of seconds between each report entry.
- Options:
  - `--path TEXT` API log file path (default is `nginx/access.log`)

The command output shows API request count data and throughput in columns.

Column headers:

- **Throughput**: ratio: (Count / minutes (end_time - start_time))
- **SvcTime**, **RespTime**: time per request over timestamp period
- **Qtime**: defaults to 0 - not used
- Example **Throughput** of API requests:

  `DURATION` is 30 seconds (0.5 minutes)

  1 API request / 0.5 minutes = 2.00

```
platform@VOSS:~$ voss report api throughput 2019-05-06_15:49 10 30
Timestamp               Count       Qtime        SvcTime       RespTime     Throughput
2019-05-06_15:49:00         1        0.00           1.47           1.47           2.00
2019-05-06_15:49:30        27        0.00           0.56           0.56          54.00
```

(continues on next page)

(continued from previous page)

```
2019-05-06_15:50:00         28        0.00          0.49          0.49          56.00
2019-05-06_15:50:30         32        0.00          0.42          0.42          64.00
2019-05-06_15:51:00         25        0.00          0.49          0.49          50.00
2019-05-06_15:51:30         21        0.00          0.59          0.59          42.00
2019-05-06_15:52:00         20        0.00          0.59          0.59          40.00
2019-05-06_15:52:30         28        0.00          0.51          0.51          56.00
2019-05-06_15:53:00         29        0.00          0.44          0.44          58.00
2019-05-06_15:53:30         21        0.00          0.63          0.63          42.00
```

3.7. Data Export Commands

3.7.1. Introduction

The Subscriber Data Extract is a capability to provide an adhoc or scheduled basic data feed via the generation of a set of files.

The content of the files is settings around Subscribers and other key services in order to support billing/expense management operations and is not intended as a general data feed.

Additional fields or files may be added to the collection over time via the roadmap, so any planned file consumption should take that into account to minimize impact.

Data Export Overview

The **voss export** command is used to carry out a bulk data export from the VOSS Automate system database. The exported data can for example be imported into a warehouse.

Important: Since a data export can take time, the **voss export** command can only be run in a `tmux` session. First run `tmux` and then **voss export** and its parameters.

Type **voss export help** for details.

The data extract schedule can be managed with the **schedule** command. For details on the use of the command, see: *Scheduling*. Since bulk data exports can typically take more than an hour on a scale system, it is recommended to schedule this task instead of running it manually from the console.

The export file format is JSON as per RFC 7159. For details on the filename, format and contents of the export files, refer to the Data Export Types topic in the Appendices.

The **voss export** command takes a `type` or `group` parameter to indicate the type of data to export.

The following are values of the `group` parameter:

- `subscriber`
- `license`

For example:

voss export group subscriber

```
platform@VOSS:~$ voss export group subscriber
Starting subscriber group export consisting of analogue_line_mgcp,
 analogue_line_sccp, call_pickup_group, contact_center_enterprise,
 contact_center_express, customer, extension_mobility, fmc,
 hunt_group, line, phones, site, subscriber, webex_teams, please wait...
Starting analogue_line_mgcp export, please wait...
Completed analogue_line_mgcp export,
 created 2019-09-30_0859_analogue_line_mgcp.json.gz.
[...]
```

3.7.2. Subscriber data export command

Note: The command `voss subscriber_data_export` is equivalent to `voss export group subscriber`.

Important:

- To optimize performance:
  - On a unified node topology, run and schedule the data export command from the *secondary* database server if possible.
  - On a modular topology, run and schedule the data export command from any *application* server.
- Since a data export can take time, the `voss subscriber_data_export` and `voss export` commands can only be run in a `tmux` session. First run `tmux` and then `voss export` and its parameters. See also:

  Using the tmux command in the Installation Guide
- Since the data export command runs database queries, it is recommended that the data exports be scheduled. Refer to the topic on scheduling for details and syntax.

  for example:

  schedule add subscriber_export voss export group subscriber

  schedule time subscriber_export weekly 1

  Best practices for scheduling to consider, are:
  - Individual report exports should be scheduled in a serial manner so that they do not overlap and result in a high database load.
  - For resilience:
    - Stagger the schedule based on how long it is expected to run - in accordance with the number of subscribers in the database.
    - For better failover support, schedules can be created on all active Unified Nodes. This requires a more complex schedule staggering and collection management.
    - For simplified schedule staggering and the export collection management, schedules can be created and staggered on a single Unified Node. This option but requires a manual re-schedule in the case of node failover.

More than one parameter can be specified for the command by prefixing them with the `type` parameter. For example: `voss export type line site`

The `type` parameter values by `subscriber` group are listed below, as well as a reference to the content details:

- `analogue_line_mgcp` (*Analogue line MGCP Data Export*)
- `analogue_line_sccp` (*Analogue Line SCCP Data Export*)
- `call_pickup_group` (*Call Pickup Group Data Export*)
- `contact_center_enterprise` (*Contact Center Enterprise Data Export*)
- `contact_center_express` (*Contact Center Express Data Export*)
- `customer` (*Customer Data Export*)
- `extension_mobility` (*Extension Mobility Data Export*)
- `fmc` (*FMC Data Export*)
- `hunt_group` (*Hunt Group Data Export*)
- `hybrid` (*Hybrid Data Export*)
- `line` (*Line Data Export*)
- `ms_o365` (*MS Office 365 Data Export*)
- `ms_teams` (*MS Teams Data Export*)
- `ms_exchange` (*MS Exchange Data Export*)
- `pexip_conference` (*Pexip Data Export*)
- `phones` (*Phones Data Export*)
- `site` (*Site Data Export*)
- `subscriber` (*Subscriber Data Export*)
- `voss_phone_servers` (*VOSS phone servers data export*)
- `webex_teams` (*Webex Teams Data Export*)

The export file directory and file format of the `subscriber` group is:

- directory: `media/data_export/<YYYY-MM-DD>`
- file naming format: `<YYYY-MM-DD_HHMM>_<type>.json.gz`

For `subscriber` group files:

- A retention policy of 30 days is in place. After each successful extraction of the data, any extract files 31 days old or older will be removed.
- If an export contains no data, a JSON file will contain an empty JSON list: `[]`.

Example:

`media/data_export/2018-10-11/2018-10-11_1236_analogue_line_sccp.json.gz`

Command examples:

- Single type

```
$ voss export type line
Starting line export, please wait...
Completed line export, created 2018-10-11_1236_line.json.gz.
```

- Multiple types

```
$ voss export type line site
Starting line export, please wait...
Completed line export, created 2024-05-07_0705_line.json.gz.
Starting site export, please wait...
Completed site export, created 2024-05-07_0706_site.json.gz.
```

- Group

  All types in a group are exported.

```
$ voss export group subscriber
Starting subscriber group export consisting of analogue_line_mgcp, analogue_line_sccp, [..
↪.]
Starting analogue_line_mgcp export, please wait...
Completed analogue_line_mgcp export, created 2018-10-11_1236_analogue_line_mgcp.json.gz.
Starting analogue_line_sccp export, please wait...
Completed analogue_line_sccp export, created 2018-10-11_1236_analogue_line_sccp.json.gz.
[...]
Completed subscriber group export.
```

The export files can then be copied to a remote system. For example, from the Automate system, list out the data export files:

```
$ ls media/data_export/2018-10-11
2018-10-11_1236_analogue_line_sccp.json.gz
```

The exported files can be copied to a remote system using SCP or SFTP on port 22. For example:

```
remote_system:~$ scp <platform_user>@<voss_system>:media/data_export/2018-10-11/2018-10-
↪11_1236_analogue_line_sccp.json.gz .
```

Note: Contact your VOSS Account team for details regarding the reports obtained from the following commands:

- `voss export type nbi-subscriber` (internal)

---

4. Cluster Deployment Commands and Steps

4.1. Deployment Commands

Architecturally, VOSS Automate offers two main deployment topologies:

- Unified Node Cluster Topology
- Modular Cluster Deployment

For details:

see the VOSS Automate Deployment Topologies chapter in the Architecture and Hardware Specification Guide.

Also refer to:

- the Install Guide
- *Clustering commands*

Note: From release 21.1 onwards, a standalone topology is considered a cluster-of-one. This means that commands such as **cluster list** and **cluster status** should be run on a standalone topology.

In a clustered topology, a number of nodes with different roles are clustered together and provisioned to form a networked system. When nodes are clustered together, High Availability and Disaster Recovery can be achieved by ensuring that there are redundant services. Nodes can be deployed in any order.

Important:

- On a Unified node topology, the commands below should be run on the primary database node.
- On a Modular topology, the commands below should be run on the database node.

Once two or more nodes have been deployed, the nodes can be grouped into a cluster by executing **cluster add <ip>**. Note that a node already in one cluster cannot be added to another cluster. Likewise, nodes can be removed from a cluster with the **cluster del <ip>** command. The nodes in the cluster can be displayed using **cluster list**.

Cluster roles cannot be changed after installation because it is dependent on installed software and other configuration at time of deployment.

The status of the cluster can be viewed using **cluster status**.

If the node topology needs to be changed, the following procedure can be followed:

- A node can be removed from the cluster with **cluster del <ip>**

- The node can be redeployed with the correct parameters
- Add the new node to the cluster with **cluster add <ip>**
- Provision the cluster with **cluster provision**. This command should only be run on one node in the cluster, usually an application server.

4.2. Migrate a Unified Node Cluster to a Modular Cluster

From release 21.1 onwards, a new Modular Architecture deployment is available.

Important: Since individual requirements and topologies differ, contact VOSS for guidance and assistance *before* migrating.

Note:

- A topology migration from a unified node cluster topology to a modular cluster is recommended under certain circumstances to improve the system behaviour at scale and under load.
- Since the *minimum* number of nodes in a modular architecture is 8 (2 web proxy and 6 application/database), *only* a migration of a unified node deployment of 8 nodes is supported.
- The migration to a modular architecture deployment is a one-way process. A migrated system cannot be migrated back to a unified node cluster topology.
- The migration process requires a maintenance window to complete, since a VM power down is needed for the migration of some nodes.
- For details on *installing* with this topology, see the Architecture and Hardware Specification Guide and Installation Guide:
  - Multinode Modular Cluster with Application and Database Nodes
  - Multinode Modular Cluster Hardware Specification
  - Modular Architecture Multinode Installation

Commands are available to migrate unified nodes from the unified node cluster topology. Since unified nodes combine database and application roles, the commands remove one of these roles. The commands are:

- **app remove-role db**: remove the database role from a unified node, leaving the application role.
- **app remove-role app** remove the application role from a unified node, leaving the database role.

4.2.1. Considerations and Prerequisites

- VOSS has been contacted and is assisting.
- The migration requires a maintenance window.
- The new topology should be decided, in other words, which nodes take the db and app roles in the database centers.

- Backups before restarting:

  As part of the rollback procedure, ensure that a suitable restore point is obtained prior to the start of the activity, as per the guidelines for the infrastructure on which the VOSS Automate platform is deployed.

  Database backups from the Primary Unified node cannot be used to restore the database on the new primary database node after migration.

  Note: If you need to roll back from the migration using the restore points, first revert *all* the cluster nodes to the correct restore point, then power the nodes on. Do not revert the restore point and power on each node in sequence.

- Hardware impact:
  - Conversion from unified to app node - has disk space impact (db space unused and can be removed in the VM)
  - Conversion from unified to database node - unified node requirement on VM of 16GB RAM needs to be changed to 32GB RAM for database node.
- Schedules:
  - system
  - admin created
  - application related: NBI

4.2.2. Steps

The example and steps below shows a migration from a unified node cluster topology with 6 unified nodes in 2 data centers to modular topology with 3 database nodes and 3 application nodes:

Data Center 1		
Unified Node Topology	**Migration Command**	**Modular Topology**
UN1	**app remove-role application**	db node (primary db)
UN2	**app remove-role database**	app node
UN3	**app remove-role application**	db node
UN4	**app remove-role database**	app node
WP1	N/A	WP1

Data Center 2 (Data Recovery site)		
Unified Node Topology	**Migration Command**	**Modular Topology**
UN5	**app remove-role application**	db node
UN6	**app remove-role database**	app node
WP2	N/A	WP2

Before you start, verify that the nodes on which you run the migrate commands are *unified* nodes.

1. Check the memory of the nodes that will have application roles removed (the nodes: UN1, UN3 and UN5) to identify those with memory *less* than 32GB.

   You can run **diag mem** on the nodes to verify this and add `Mb used` and `Mb free`.

2. Shut down the system:

   a. From the primary database node: **cluster run notme system shutdown**

   b. Then **system shutdown**

3. Make a restore point as per the guidelines for the infrastructure on which the VOSS Automate platform is deployed.

4. For the VMs of unified nodes that:

   - will become db nodes and
   - have less than 32GB memory (from the nodes: UN1, UN3 and UN5),

   resize the VM memory to 32GB as follows:

   a. Log into VMWare GUI.

   b. Adjust the **Memory** assigned to the system to 32GB.

5. Power up the VMs.

6. Run **app remove-role database** on UN2, UN4 and UN6.

7. Run **app remove-role application** on UN1, UN3 and UN5.

8. Do not modify the web proxy nodes WP1, WP2.

9. Run **cluster provision** on UN1.

10. Run **cluster status** to ensure the cluster roles are as expected.

Note: When the deployment is migrated to a modular system, the **cluster provision** command will run a check to verify that all nodes have been migrated, in other words, that no "hybrid" system is present.

An error message is displayed otherwise:

```
Cluster provisioning not allowed on cluster with mixed unified and db or app nodes
```

---

11. Run **database config** on the first new *database* node and ensure the database nodes are in the correct state. In this example it would be UN1.

12. Verify the primary application node (UN2) with the **cluster primary role application** command run on the node. The output should be *true*, for example:

```
platform@UN2:~$ cluster primary role application
is_primary: true
```

    See: *Determine the primary node role in a cluster*.

13. For the new *application* nodes (nodes where the database role has been removed, in other words UN2, UN4 and UN6)

    a. Check the number of queues using **voss queues** and if the number is *less than 2*, set the queues to 2 with **voss queues 2**.

Note: Applications are reconfigured and the `voss-queue` process is restarted.

b. Remove the database drives and volumes. Use the following commands to check and carry out the tasks. See the output example: *Removing Database Drives and Volumes*.

- `drives list`
- `drives remove_logical voss dbroot`
- `drives remove_volume voss`

Important: Ensure the nodes are application nodes: you can run **app status** and verify no `mongodb` service is running on them.

14. You can remove disks from VMs of nodes with *application* roles (UN2, UN4 and UN6).

Important: Ensure the nodes you remove VM disks are application nodes: run **app status** and verify no `mongodb` service is running on them.

If disks are hot-mounted, a system restart may be needed.

a. Run **drive list** and confirm that LVM volume is shown as `available`. Make note of VG size.
b. Log into VMWare GUI.
c. Look at disks attached to system, and specifically ones that are the size noted above.
d. Remove that disk from the system.
e. On the CLI, rerun **drive list** to confirm that LVM and VG are removed.

4.2.3. Removing Database Drives and Volumes

Example of commands and output when removing database drives and volumes.

```
platform@UN2:~$ drives list
Used disks and mountpoints:
    sdc1 -  services:backups

Unused disks:
dm-0

Unused mountpoints:
    services:SWAPSPACE

Volume Groups
    voss - 25.0 GB free, 250.0 GB total
      Physical volumes:
        sdd1
      Logical volumes:
        dbroot/dm-0 - 225.0 GB
platform@UN2:~$
```

```
platform@UN2:~$ drives remove_logical voss dbroot
Logical volume dbroot has been removed from voss
platform@UN2:~$ drives list
Used disks and mountpoints:
    sdc1 -  services:backups

Unused disks:
    none - if disks have been hot-mounted, it may be necessary to reboot the system

Unused mountpoints:
    services:SWAPSPACE

Volume Groups
    voss - 250.0 GB free, 250.0 GB total
      Physical volumes:
        sdd1
      Logical volumes:
platform@UN2:~$
```

```
platform@UN2:~$ drives remove_volume voss
Volume group voss has been removed
platform@UN2:~$ drives list
Used disks and mountpoints:
    sdc1 -  services:backups

Unused disks:
sdd

Unused mountpoints:
    services:SWAPSPACE

platform@UN2:~$
```

4.3. Determine the primary node role in a cluster

To determine if the node is the primary database node or the application node, run the `cluster primary` command (with additional parameters) from the CLI on a node:

Note: This command should be used to establish and confirm the primary application node during patch, patch bundle, or delta bundle installation.

```
platform@VOSS-UN-1:~$ cluster primary
is_primary: true
platform@VOSS-UN-1:~$ cluster primary role application
is_primary: true
platform@VOSS-UN-1:~$ cluster primary role database
is_primary: false
```

(continues on next page)

(continued from previous page)

```
platform@VOSS-UN-1:~$
```

5. Provisioning

5.1. Provisioning

The system is installed as a loosely bundled set of applications. In order for the applications to be coupled, a process called ‘provision’ must take place.

Standalone systems are single-node (“cluster-of-one”) clusters and are provisioned automatically since there is only one node in the system. This can be performed manually afterward with **cluster provision**.

When the topology of the cluster changes, e.g. additional nodes or applications are added; or to reprovision the system to bypass a faulty node, the cluster must be reprovisioned using **cluster provision**.

Note that for multi-node clusters, the cluster provisioning needs to reconfigure and restart services across the cluster in a complex arrangement and the provisioning duration is dependent on the number of nodes - it can take a number of hours for large installations.

Provision the cluster from the primary node with **cluster provision** For backwards compatibility, this command is the same as for example **cluster provision fast**.

Use the **cluster provision serial** command if the VMware host is under load.

The provisioning step may take some time, because all applications must be cross-configured to work with one another and the database is also provisioned. If the system discovers that no primary database server exists (or multiple database servers exist), the **cluster provision** command prompts the user to select a primary server manually.

See also: *Using the tmux command*.

6. Networking

6.1. Network interfaces

Important: From Automate release 24.1 onwards, network addresses are in CIDR (Classless Inter-Domain Routing) format, for example: `192.168.100.3/27` or `e00d::fafe:23/112`. The use of a netmask in the `255.255.255.0` format is no longer supported.

The command **network interfaces** will display the available network interfaces and their configuration.

The hostname can be set or changed with **network name <hostname>**.

Note: The maximum character length for the hostname is 56.

A network interface can be configured or changed as follows:

network <interface-name> <cidrIP> <netmask> <gateway>

For example:

- IPv4:

  network eth0 172.29.89.100/24 172.29.89.1

- IPV6:

  network ens160 a00d::eafe:3/112 a00d::eafe:25

The gateway can be configured using:

network <interface-name> gateway <gateway>

The network interface with IP, netmask and gateway can be configured with:

network <interface-name> <cidrAddress> <gateway>

The IP address can be changed without affecting the netmask and gateway using:

network <interface-name> ip <ip>

or

network <interface-name> ip <ip>/<netmask>

For example IPv4: **network eth0 ip 172.29.89.100**.

Deleting a IPv4 interface: **network -4 del ens160** Deleting a IPv6 interface: **network -6 del ens160**

How to change a Network Interface

The steps below apply to the network interface change on both single node and multi-node topologies.

Note:

- During this process, the application or database primary might be migrated automatically to a new node.
- Ensure you are working on the correct node when running the below commands by checking with the *cluster primary* command.
- A single node system will have all commands run on it unless otherwise stated.

---

1. If the node to be changed is a database or unified node, then record the currently assigned database weight *database weight list*.
2. On the current application primary run: *cluster del <IP>* where IP is the IP address of the node being changed.
3. On the node being changed, run *network list* and note the interface name that will be changed.
4. On the node being changed, run the *network <interface> ip <newIPAddress>* command to change the IP.
5. Reboot the node being changed.
6. If not a single node (standalone):

   Log in as the platform user to the node with the new IP address and run *cluster prepnode*.
7. On the current application primary run *cluster add <newIPAddress>*
8. On the current primary database or unified node (not standalone / single node), run *database weight add <NewIPAddress> <weight>* to restore the database weight previously assigned to the changed node.
9. On the current primary application node, run *cluster provision*.
10. On the current application primary node, ensure services are restored by running: *cluster check* and *cluster run all app status*.
11. If any services are not running, try and restart them using: *app start <service>*.

Network Routes

Network routes can be displayed with **network routes**.

- To add a route to a subnet (using `<netmask>`):
  - via a specified gateway: **network routes <network-ip>/<netmask> <gateway>**
  - via a specific gateway and interface: **network routes <network-ip>/<netmask> <gateway> dev <interface-name>**
- To a host without specifying the gateway: **network routes <host-ip> dev <interface-name>**.
- Network routes can be deleted with **network routes del <network-address>**.

Network Static Hosts

Static hosts can be maintained in `/etc/hosts` by using the following commands:

- Add: **network static_host add <host> <ip>**
- Delete: **network static_host del <host>**
- List: **network static_host list**

Example:

```
$ network static_host list
    static_hosts:
        lab.solutions.com: 10.121.1.42
```

6.2. Switching Network Addresses from IPv4 to IPv6

From release 24.1, the VOSS Automate platform supports IPv4 as well as IPv6 network addressing. Options available are:

- IPv4 only
- IPv4 and IPv6
  - with IPv4 as primary network
  - with IPv6 as primary network
- IPv6 only

For network address options during installation, refer to the Installation Guide:

Create a New VM Using the Platform-Install OVA.

The steps below provide options for a system using IPv4 addresses to use:

- IPv4 and IPv6
  - with IPv4 as primary network
  - with IPv6 as primary network
- IPv6 only

1. Add the relevant IPv6 address to the node.

```
platform@VOSS:~$ network ens160 f10d::eafe:3/112 f10d::eafe:25
This change might require a reprovision. Do you wish to continue [y/n]? y

Reprovision NOT needed. Primary address is ipv4.

   ipv6:
       gateway: f10d::eafe:25
       ip: f10d::eafe:3/112

Restart your system to reflect networking changes ipv6

platform@VOSS:~$
```

2. Reboot the node if told to do so in the output.
3. Repeat the above for steps all nodes.

Important:

- Stop here if you want to use both IPv4 and IPv6 with IPv4 as the primary network.
- Continue if you want to use both IPv4 and IPv6 with IPv6 as the primary network, or if you only want to use IPv6.

4. Removing IPv4 addresses: on each node, remove the IPv4 address from the cluster with: `cluster del <IPv4Address>`

```
platform@VOSS:~$ cluster del 192.168.100.3
You are about to delete a host from the cluster. Do you wish to continue? y
Cluster successfully deleted node 192.168.100.3

Please run 'cluster provision' to reprovision the services in the cluster

Please note that the remote host may still be part of the database clustering and␣
↪should either be shut down
or reprovisioned as a single node BEFORE this cluster is reprovisioned
platform@VOSS:~$
```

5. Reboot the node: `system reboot`
6. Run `cluster prepnode` on each node except the primary application node

```
platform@VOSS:~$ cluster prepnode
You are about to add this node to a cluster.
Do you wish to continue? y
platform@VOSS:~$
```

7. From the primary application node, add each node to the cluster using the respective IPv6 address: `cluster add <IPv6Address>`

```
platform@VOSS:~$ cluster add a00d::eafe:3
-> Re-using existing platform/seed as I am the only node
Cluster successfully invited node [a00d::eafe:3]

Please run 'cluster provision' to provision the services in the cluster
platform@VOSS:~$
```

8. On the primary application node run: `cluster provision`

```
platform@VOSS:~$ cluster provision
You are about to provision the cluster, which will require restarting services at a␣
↪data center.
Please ensure for a cluster, the nodes are communicating correctly with a test␣
↪command such as 'cluster run all network
list' or a 'cluster run all app status' before you proceed.
Do you wish to continue? y
```

(continues on next page)

(continued from previous page)

```
---------- VOSS, ip=[a00d::eafe:3], role=webproxy,application,database, loc=cpt
```

Important:

- Stop here if you want to use both IPv4 and IPv6 with IPv6 as the primary network.
- Continue if you only want to use IPv6.

9. To remove IPv4 completely, delete the IPv4 addresses from each node: `network -4 del ens160`

```
platform@VOSS:~$ network -4 del ens160
This change will require a reprovision. Do you wish to continue [y/n]?y
Primary address is ipv6, setting ipv4

   ipv4: value not set



platform@VOSS:~$
```

10. Restart the network (you will lose the SSH connection): `network restart`
11. Post change checks include the following:

```
platform@VOSS:~$ cluster list
Cluster has 1 nodes:

       webproxy :  [a00d::eafe:3]
    application :  [a00d::eafe:3]
       database :  [a00d::eafe:3]


platform@VOSS:~$ cluster status

Data Centre: cpt
       webproxy : VOSS[a00d::eafe:3]

    application : VOSS[a00d::eafe:3]

       database : VOSS[a00d::eafe:3]



platform@VOSS:~$ cluster check
No issues found with host checks
platform@VOSS:~$
```

6.3. Network services

Network security is described in detail under the *Network Security* section, including detail regarding firewall ports.

NTP servers can be configured using the following commands:

- `network ntp` will display the list of configured NTP servers
- `network ntp set <ntp-server1> [<ntp-server2> ...]` will set up one or more NTP servers.

  Note: This command will overwrite any existing list of configured NTP servers.

- `network ntp add <ntp-server1> [<ntp-server2> ...]` will add one or more NTP servers to the existing list
- `network ntp del <ntp-server>` will delete a NTP server

DNS servers can be configured using the following commands:

Note: Any network dns add/del/set command results in a service restart. All services will be restarted.

- `network dns` will display the configured DNS servers
- `network dns set <dns-server1> [<dns-server2> ...]` set up one or more DNS servers

  Note: This command will overwrite any existing list of configured DNS servers.

- `network dns add <dns-server1> [<dns-server2> ...]` will add one or more DNS servers to the existing list
- `network dns del <dns-server>` will delete a DNS server
- `network domain <domain-name>` sets the default DNS domain
- Alternate DNS search domains can be configured with `network search add <domain>` and `network search del <domain>`

6.4. Network URI specification

All network locations are specified as a URI, for example download locations, backup destinations, notification destinations, and so on.

The following list shows the URI syntax:

- ftp: `ftp://user[:password]@host[:port][/path]`
- http: `http(s)://user[:password]@host[:port]/path`
- file: `file://{/path}+[/filename]`
- sftp:
  - `sftp://user[:password]@host[:port][/path]`

  - Azure: `sftp://<storage_acct_name>.<sftp_local_user>[:passwd]@<storage_acct_name>/path`
    * `<storage_acct_name>` is set up on Azure. The name of the Storage Account.
    * `<sftp_local_user>` is set up on Azure if you enable SFTP and add a local user within the Storage Account settings.
- scp: `scp://[user@]host[:port]:[/path]`
- email: `mailto:user@host`
- snmpv2: `snmp://community@host[:port]`
- snmpv3: `snmp://user:auth:password]@host[:port] ... minimum auth/password`

The `[password]` in the URI is optional when authentication is set up. Refer to *SSH key management*.

Note:

- If a password contains special characters, it should not be added to the URI, but typed in at the password prompt.
- If necessary, the URI should be URL-encoded: reserved characters should be encoded with percent-encoding.

6.5. Network Docker container range

Important: When troubleshooting network issues, verify that the address range is not in use.

If it is in use, use the following command to modify the Private Address Space, as described below: `network container range add <private IP>`

RFC-1918 states that the following three blocks of the IP address space are reserved for private internets:

```
10.0.0.0        -   10.255.255.255  (10/8 prefix)
172.16.0.0      -   172.31.255.255  (172.16/12 prefix)
192.168.0.0     -   192.168.255.255 (192.168/16 prefix)
```

This subnet block address range can be modified to another Private Address Space if needed.

View the current Private Address Space

Use the following command to display the current Private Address Space: `network container range list`

For example:

```
$ network container range list
    range: 10.1.2.1/24
```

Modify the Private Address Space

Use the following command to modify the Private Address Space: `network container range add <private IP>`

Important: A valid Private Address IP is required as input.

The range `/24` is appended to the IP. For example, if `192.168.0.6` is used, the Private Address range `192.168.0.0/24` is used.

In a clustered environment, you could use the following command: `cluster run all network container range add <private IP>`

Setting a different Private Address Space on each node

If required, you can set the Private Address Space to be different on each node by running the `add` command on each individual node. For example:

```
$ network container range add 192.168.2.3
You are about to restart all services. Do you wish to continue?y
Application processes stopped.  (note this line changes dynamically)

Reconfiguring applications....
Application  processes started. (note this line changes dynamically)
```

7. Application Control

7.1. Application Control and Status

The functioning system is comprised of applications. Each application has a name and a version number. An application may have multiple processes running within and each process has its own state.

7.1.1. Application Status

The command **app status** is used to display the status of the system. When the command is executed, it requests an up-to-date status of every application and process, and hence may take a few seconds to return.

A typical app status screen from the command line interface:

```
platform@VOSS:~$ app status
cluster v24.2.0 (2024-11-25 15:34)
insights-api v24.2.0 (2024-11-25 15:35)
   |-api                   running
insights-voss-sync v24.2.0 (2024-11-25 15:38)
   |-full-sync             disabled
   |-real-time             running (completed)
   |-database              running (healthy)
   |-delta-sync            scheduled
   |-full-transaction      scheduled
   |-multivendor           scheduled
license-service v24.2.0 (2024-11-25 15:25)
   |-license-service       running (healthy)
mongo-upgrade v0.0.0
mongodb v24.2.0 (2024-11-25 15:36)
   |-arbiter               running
   |-database              running
nginx v24.2.0 (2024-11-25 15:36)
   |-proxy                 running
platform v24.2.0 (2024-11-25 15:37)
security v24.2.0 (2024-11-25 15:37)
selenium v24.2.0 (2024-11-25 15:37)
   |-gui_orchestration     running
selfservice v24.2.0 (2024-11-25 15:26)
   |-node                  running
services v24.2.0 (2024-11-25 15:37)
```

(continues on next page)

(continued from previous page)

```
   |-logs                  running
   |-scheduler             running
   |-mount                 running
   |-wsgi                  running
   |-syslog                running (completed)
   |-time                  running (completed)
   |-firewall              running (completed)
snmp v24.2.0 (2024-11-25 15:37)
   |-daemon                running (completed)
   |-traps                 running (completed)
support v24.2.0 (2024-11-25 15:37)
template_runner v24.2.0 (2024-11-25 15:37)
vmware v24.2.0 (2024-11-25 15:37)
voss-deviceapi v24.2.0 (2024-11-25 15:40)
   |-voss-cnf_collector    running
   |-voss-risapi_collector running
   |-voss-monitoring       running
   |-voss-queue            running
   |-voss-wsgi             running
voss-portal v24.2.0 (2024-11-25 15:30)
   |-gui                   running
```

Applications can have multiple processes running within it - each with its state displayed. The following states are defined:

- `running` indicates that the process is running correctly.
- `completed` indicates that the process ran to completion successfully.
- `suspended` indicates that the process is suspended while waiting for another process.
- `stopped` indicates that the process is not running. An error message indicates that the process stopped for an unexpected reason.
- `disabled` indicates that the application is by default disabled or is not licensed.
- `scheduled` indicates that the application is scheduled to run.

7.2. Starting and Stopping

The system application may be stopped with **app stop** and restarted with **app start**.

Important: While services can be started across a cluster, they should not be stopped using the **cluster** command.

In other words, do not run **cluster run <where> app stop <no arg>**

For details on the cluster command and the `<where>` clause, see: *CLI Commands*.

By default this is a non-blocking command, which means that the console prompt will be available after running this command while processes that are a part of it are running.

It is possible to start or stop individual applications and/or processes by appending the `<application-name>[:<process-name>]`. The list of applications can be seen by using the command **app status**.

For example, to start the process `voss-queue`:

app start voss-queue

or

app start voss-deviceapi:voss-queue

It is possible to perform a blocking start by including `blocking` after start but before the `<application-name>[:<process-name>]`. For example:

app start blocking

app start blocking voss-queue

This will ensure that all background processes that are started by **app start** will be completed before the console prompt is available.

7.3. Installing Applications

In general, it is not necessary to install single applications on a running system. Instead, the system is upgraded using **cluster upgrade** as described in the Upgrading Applications section.

The system collects all visible application versions for display using **app list**.

A screen for the app list command from the command line interface:

```
platform@cscluster1:~$ app list
selfservice - selfservice install script
   latest version 1.5.0 (2015-11-23 17:28) is installed

voss-deviceapi - voss-deviceapi install script
   latest version 1.5.0 (2015-11-23 17:26) is installed

cluster - cluster management
   latest version 1.5.0 (2015-11-23 17:25) is installed

template_runner - Template Runner
   not installed
   version available: 1.5.0 (2015-11-23 17:36)

mongodb - no-sql database server
   latest version 1.5.0 (2015-11-23 17:24) is installed

support - Diagnostic tools for tech support
   latest version 1.5.0 (2015-11-23 17:27) is installed

snmp - snmp management client and server
   latest version 1.5.0 (2015-11-23 17:31) is installed

nginx - lightweight web server
```

(continues on next page)

(continued from previous page)

```
   latest version 1.5.0 (2015-11-23 17:24) is installed

services - Platform core services
   latest version 1.5.0 (2015-11-23 17:26) is installed

platform - Platform install/upgrade
   latest version 1.5.0 (2015-11-23 17:26) is installed

nrs - NRS install script
   not installed
   version available: 1.2.0 (2015-11-23 17:36)

security - Latest system security updates
   latest version 1.5.0 (2015-11-23 17:34) is installed

vmware - VMware tools
   not installed
   version available: 1.5.0 (2015-11-23 17:24)
```

Each application will display a short description of the application, the version installed, and whether other versions are available.

Additional applications can be downloaded using the instructions detailed under the System Control:Download section.

7.3.1. Single Application Installation

An application can be installed with **app install <application-name>[version:<version>] [delete-on-success <[yes|y]|[no|n]>]**

For example,

- **app install snmp**
- **app install platform version 0.8.1 2014-01-09 00:46**

For release 19.1.2 and later; when using the **app install** command to install patch scripts that have been added to the `media/` directory, the **delete-on-success** parameter and `yes|no` value can be added to remove or keep the the patch file in the directory after successful installation.

Example: **app install media/patch.script delete-on-success yes**

By default, in other words without this parameter, the user is prompted whether to delete or keep the patch script:

```
Do you want to remove patch after successful completion?
```

If the `--force` parameter is appended to the parameter, no prompt is shown.

Note: This script file removal using the `delete-on-success` parameter only applies to *patch installation* and not to the installation of delta bundles or other bundles.

The application will automatically start its processes on install. In isolated rare cases, it may be necessary to manually provision the system afterward with **cluster provision** so that other applications are configured to work with the new application. This is taken care of automatically during the upgrade process described below.

7.3.2. Multiple Application Installation

More than one application can be installed by adding them as `multi_install` parameters:

A file `media/<manifest_file>.manifest` can be created to list the applications per line or space-separated, in the install sequence.

The `delete-on-success` parameter is optional.

- If the applications are listed on the command line, the sequence of the application parameters in the command is the sequence of the installation.

  app multi_install [<manifest_file.manifest>|<app-name_1> <app-name_2> ...] [delete-on-success <[yes|y]|[no|n]>]

  Note:

  - It is advisable to make a system backup prior to installation so that a backup restore can be carried out in the case of a failure. Contact VOSS support if needed.
  - If an application URI is specified as a parameter, it is downloaded.
  - If an application is already installed, it is skipped with a message.
  - If an application with more than one version is found in `media/`, the latest version is installed.
  - If an application in the list fails to install, the installation is terminated. Refer to the install logs.
  - If the `delete-on-success` parameter is `yes`, the manifest file and application files are removed after successful installation.

- The applications and manifest file can also be bundled into a single archive with extension: `.tar.gz`, `.tar`.

  The command will then be of the format:

  app multi_install <archive_file> [delete-on-success <[yes|y]|[no|n]>]

  The archive is extracted to the `media/<archive_basename>` directory.

  Note:

  - If no manifest file is found, the installation exists with an error message
  - If a file in the manifest is not available, an error message for the entry is displayed.
  - The same considerations apply as noted with a multiple installation that lists the files on the command line.
  - If the `delete-on-success` parameter is `yes`, the `media/<archive_basename>` directory is removed after successful installation.

7.4. Updating Applications

The entire cluster can be upgraded from a single node with the command:

cluster upgrade <ISO>|<URL>

Some issues to note:

- By default, the cluster upgrade is carried in parallel on all nodes and without any backup in order to provide a fast upgrade. For backwards compatibility, this command is the same as for example **cluster upgrade <ISO> backup none fast**.
- Use the **cluster upgrade <ISO>|<URL> serial** command if the VMware host is under load.
- If a backup is required, use the **backup <location>** parameter with a location as it was added with the **backup add** command. The command parameter can for example be:

  cluster upgrade <ISO>|<URL> backup <location>

  A downloadable URL can be specified using **cluster upgrade <URL>** which will first be downloaded before upgrading. For example:

  cluster upgrade http://myserver/path/myfile.iso

  If a downloadable URL is not available, use the instructions detailed under the Download section to copy the application upgrade package to the local server. Once complete, use **cluster upgrade <ISO path>** to upgrade the application, or for example: **cluster upgrade myfile.iso**.

A single, cluster-of-one node can be upgraded using the local variant:

cluster upgrade <ISO>|<URL>

By default, the **upgrade** will upgrade the system using the latest ISO upgrade package in the `platform` user's directory.

The system will automatically reprovision itself after upgrading if necessary.

Note that the system will automatically perform a full backup before the upgrade so that the system can be rolled back if necessary. The destination for this backup can be specified using the following syntax:

cluster upgrade <URL>|<path> backup <backup-destination>

Valid backup destinations can be listed with **backup list**. If necessary, it is also possible to instruct the system not to perform a backup by specifying the backup destination as `None`.

See the backup restore section on how to restore a system to a former snapshot in order to revert to the snapshot prior to upgrade.

7.5. Summary Attribute Migration

If template updates that modify the summary attributes of existing models are installed, then the summary attributes in the data of existing instances of the models need to be migrated.

If the summary attributes are not migrated, the list view representation of these model types will not contain the correct columns or values for display.

The command to carry out this migration for a specific model is:

voss migrate_summary_attributes <model_type>

For example:

```
voss migrate_summary_attributes data/InternalNumberInventory
```

7.6. Remote Execution in Clusters

When commands are run on a cluster, a number of options are available to specify the nodes on which the commands can be run.

There is a *<where>* clause: **cluster run <where>**.

The clause can take:

- *role* - the role of the node: `application`, `database`, `webproxy`
- `all` - all the roles, in other words, the entire cluster
- `notme` - all nodes except the one the command is run on
- `<data center name>` - all nodes in the data center
- `<nodename|IP>` - the hostname or IP address of the node.

For example:

- **cluster run database app start mongodb** will restart the mongodb service on all database nodes.
- **cluster run all app status** will display the app status of all nodes on the cluster.
- **cluster run notme system shutdown** would issue the command to shut down all nodes except the one the command is run on.

Important: In a cluster, reboot and shutdown of the entire cluster should be done on each node and not with the **cluster run all** command.

Sometimes there are long-running processes running on a server. To display such jobs, use the **cluster job list** command.

Note: The **cluster job list** command is not available on a web proxy node.

It is also possible to re-attach to those jobs to see the output, using **cluster job reconnect <pid>**.

7.7. List of Unused Cluster Commands

The following table shows **cluster** commands that should not used:

command
cluster run all system shutdown
cluster run all system reboot
cluster run <any> backup create XXX
cluster run <any> cluster upgrade
cluster run <any> cluster run XXX
cluster run <any> cluster provision
cluster run all cluster add <ip>
cluster run all cluster del <ip>
cluster run all cluster job kill <pid>
cluster run all cluster job list
cluster run all cluster job reconnect <pid>
cluster run all cluster list
cluster run all cluster status

Only one **cluster** command can be run on web proxy nodes. A message is shown if running the other **cluster** commands, for example running the **cluster list** and **cluster status** commands *on a web proxy node*:

```
$ cluster list
Invalid command syntax - refer to help descriptions

USAGE:
------
cluster prepnode  - Prepares the system so that it can be joined to a cluster




$ cluster status
cluster status not available on webproxy nodes.
```

To exclude these **cluster** commands from running on web proxy nodes, use the `<where>` parameter to specify the node types, for example, with the `database` node type as specified below:

```
$ cluster run database cluster list
You are about to a run a command across the cluster, which could affect service␣
↪availability.
Do you wish to continue? y


---------- VOSS-UN-1, ip=164.168.151.3, role=webproxy,application,database, loc=cpt


Cluster has 8 nodes:

     application :  164.168.151.3, 164.168.151.5, 164.168.151.4, 164.168.151.7, 164.168.
↪151.6, 164.168.151.8
```

(continues on next page)

(continued from previous page)

```
        webproxy :  164.168.151.3, 164.168.151.5, 164.168.151.4, 164.168.151.7, 164.168.
↪151.6, 164.168.151.9, 164.168.151.8, 164.168.151.10
        database :  164.168.151.3, 164.168.151.5, 164.168.151.4, 164.168.151.7, 164.168.
↪151.6, 164.168.151.8



---------- VOSS-UN-3, ip=164.168.151.5, role=webproxy,application,database, loc=cpt

[...]
```

7.8. Self-service Localization Management

Translation template files for the Self-service application can be exported for translation and can be added or imported.

To export the Self-service translation template:

selfservice get_translation_template

To add or import a Self-service Translation template:

scp <local.json file> platform@<host>:media/

Log into system as platform user:

selfservice import_translation media/<local.json file>

Translation template files need to follow the naming convention:

`locale-<languagecode>.json`

For example:

- `locale-en-us.json`
- `locale-es-es.json`
- `locale-de-de.json`

7.9. Web Services

On a web proxy node only, Self-service and admin Web services can be disabled, re-enabled and listed if required. The task should be carried out after provisioning and if the Admin Portal or Self-service GUI for example needs to be disabled for security purposes.

Note: It is strongly recommended *not* to allow customer end-users the same level of administrator access as the restricted groups of provider- and customer administrators. This is why Self-service web proxies as well as Administrator web proxies should be used.

Systems with Self-service only web proxies are *only* recommended where the system is customer facing, but where the customer does not administer the system themselves.

The commands should be run on the relevant web proxy node. It is not recommended that the commands be run on a standalone single-node cluster system, but only on a multi-node cluster.

In particular, the commands to disable or enable web services will automatically reconfigure and restart the nginx process, so some downtime will result. Request URLs to a disabled service will redirect the user to the active service.

Note: SSO for end-user Self-service is supported when using a shared VOSS web proxy for Admin and Self-service, when using the Admin URL in the SSO setup. Once authenticated in the IdP via that URL, the user is dropped into the end-user Self-service interface (if they are an end user) and access via their role. SSO is not supported when using a dedicated Self-service proxy.

- To disable admin or Self-service web services on a web proxy node, run the command on the relevant node:

  web service disable <selfservice|admin>

  When running `web service disable admin`, all admin portals are disabled.

- To enable admin or Self-service web services on a web proxy node, run the command on the relevant node:

  web service enable <selfservice|admin>

- To list disabled web services on an Admin or Self-service web services web proxy node:

  web service list

  For example:

```
platform@cscluster1:~$ web service list
disable: admin
```

7.10. Insights Analytics

From release 24.1 onwards:

- Two new apps are available to:
  - Manage Insights Analytics that is used for VOSS Wingman and dashboard widgets in the VOSS Automate GUI. See the topic on *VOSS Wingman* and *Dashboards* in the Core Feature Guide.
  - Manage data sync between Automate and Insights databases.
- The Insights database is installed on each database node in the VOSS system topology.

The new installed apps (run `app status` to see these) are:

- `insights-api`: the API service allowing the communication between VOSS Automate and Insights.

  Note: The communication is associated with a specific VOSS Automate session for authentication, so that session timeout and logout will also terminate this specific communication.

- `insights-voss-sync`: the sync app, consisting of processes:
  - `real-time` and `database` components, ensuring initial creation of database tables and indexes required for the sync between Automate and Insights databases and changes to the Resource collection.

    Note: The `real-time` app runs and completes after the creation of database tables and indexes.

    This is done in order to ensure the accurate representation of data in widgets on VOSS Automate GUI dashboards. Refer to the *Synced Automate Models* section below.
  - `full-sync`, `full-transaction`, `multivendor`, `delta-sync`: sync processes that allow for additional control.

    Note:

    * Default schedules are added for data sync between Automate and Insights databases.
    * Upon upgrade to Automate 25.1, the default `full-transaction` sync schedule is hourly.

    * `full-sync`: (disabled by default, but can be run manually - refer to the topic on Manual Sync below.)
    * `full-transaction`: (default: every hour every day) syncs transaction data since previous run of this sync (the first sync defaults to the last 90 days or 10 million transactions). Only transactions that have completed will be synchronized, i.e not queued and processing transactions.

      A cleanup script runs daily to ensure the number of stored transactions remain under 10 million.
    * `multivendor`: (default: every hour every day) syncs multivendor subscriber details for the multivendor overview dashboard.
    * `delta-sync`: (default: every 30 minutes) initially carries out a full sync and thereafter, an incremental resource sync. Allows for the synchronization of specific Automate objects defined by the `data/ReporterModels` instance which maps Automate and Insights databases. See the topic *Synced Automate Models* below.

    If the initial management of dashboards on the GUI is required prior to the first scheduled sync, the `full-sync`, `full-transaction`, `multivendor` and `delta-sync` sync processes can be run manually - see the topic *Manual Sync* below.

The log files corresponding to `insights-voss-sync` components can be inspected and followed, for example:

```
insights-voss-sync/real-time.log
insights-voss-sync/full-transaction.log
insights-voss-sync/full-sync.log
insights-voss-sync/delta-sync.log
insights-voss-sync/multivendor.log
insights-voss-sync/database.log
```

Note: Since these log files can grow to be large files, they are by default rotated on an hourly basis to conserve disk space.

For details on viewing logs, see: *Logs*.

7.10.1. Sync and service management commands

Scheduling

While default syncs are configured (see output below), Insights sync of data can also be scheduled manually by using the **insights sync schedule time** command to specify daily and weekly schedules.

The default syncs and `disabled` status can be seen:

```
platform@VOSS:~$ insights sync schedule list
    schedule:
        delta-sync:
            hour: every
            minute: every/30
            weekday: every
        full-sync:
            disabled: true
            hour: 1
            minute: 0
            weekday: 6
        full-transaction:
            hour: every/1
            minute: 0
            weekday: 0
        multivendor:
            hour: every
            minute: 0
            weekday: every
```

Use this `schedule list` command to inspect the current state of schedules.

Schedule Commands

The schedule command takes three options: sync type name, time and weekday.

Sync type (`{sync-type}`) can be:

- `full-sync`
- `full-transaction`
- `multivendor`

The `delta-sync` type runs every 30 minutes and is only available as a manual sync using **insights sync run**. Refer to the topic on Manual Sync below.

Time and weekday are set up separately.

Note that time can also be an interval (`every`):

`insights sync schedule time {sync-type} every <{n} hours|{n} minutes>`

or the format: `every/{n}`:

`insights sync schedule time {sync-type} <every/<{n} hours|every/{n} minutes>`

With `{n}`:

- hours: 0 - 23

- minutes: 0 - 59

Setting up the schedule by a specific time:

```
insights sync schedule time {sync-type} <{hour} {minute}>
```

For example:

```
insights sync schedule time {sync-type} 02 15
```

for a 2:15AM sync on the weekday of the setup.

Setting up weekly:

```
insights sync schedule time {sync-type} weekly {weekday}
```

Examples:

- `insights sync schedule time {sync-type} weekly 5`

  to change the time setup to be a weekly Friday sync.

- `insights sync schedule time {sync-type} weekly every`

  to change the time setup to be a weekly every day.

The value convention for `{weekday}` is:

- `0` is Sunday
- `1` is Monday
- `2` is Tuesday
- `3` is Wednesday
- `4` is Thursday
- `5` is Friday
- `6` is Saturday
- `every` is every day

Manual Sync

The scheduled syncs as well as the real-time sync will keep the widget data in sync. A sync can also be run manually at any time in order to ensure the Automate database and Insights reporter database are in sync so that dashboards reflect current data.

Note:

- Since a full sync can be a long process, the manual run of this command should be carried out at a suitable time. The duration of a sync would be dependant on the amount of data in the database and for full-transaction syncs, the amount of activity against that data since the last sync, or 90 days / 10 million transactions.

  A cleanup script runs daily to ensure the number of stored transactions remain under 10 million.

- If a sync is in progress, another sync of the same type will not run.
- Run this command in a terminal opened with the `tmux` command.

  For details, see:

  Using the tmux command in the Platform Guide.

Use the **insights sync run** command with options:

- **insights sync run {sync-type}**

where `{sync-type}` can be:

- `delta-sync`: Fetches changed data incrementally, in other words, data changed since previous fetch. Stale data is also deleted. Note that `multivendor` sync is *not* included.

  This sync type is by default scheduled to run every 30 minutes. The command can also be run using the parameter `--full-refresh` to carry out a full sync of insights data.

- `full-sync`. This command:
  - Carries out the equivalent of the command and parameter `delta-sync --full-refresh`, which fetches all data from the beginning of time (Jan 1, 1970).
  - Concludes with a `multivendor` sync - refreshing multivendor database tables.

  Note that no output is shown in the `tmux` session while the sync is in progress.

```
$ insights sync run full-sync
Syncing can be a long process. Do you wish to continue?yes
```

- `full-transaction`: transaction data sync.

  Only transactions that have completed will be synchronized, i.e not queued and processing transactions.

  In addition, no system and data sync transactions are synchronized.

```
$ insights sync run full-transaction
Syncing can be a long process. Do you wish to continue?yes
```

- `multivendor`: syncs multivendor subscriber details.

If upgrading to release 24.2, no initial manual sync is required for dashboard management, since `delta-sync` runs every 30 minutes and will then carry out this first sync.

Note: The sync process initially takes some time.

Follow the progress of the sync by inspecting the related log files:

```
$ log follow process/insights-voss-sync.full-sync.log
$ log follow process/insights-voss-sync.delta-sync.log
$ log follow process/insights-voss-sync.full-transaction.log
$ log follow process/insights-voss-sync.multivendor.log
```

Note: The `delta-sync` sync process does not sync transaction data.

To sync transaction data, either:

- Run `insights sync run full-transaction` manually
- Use the Platform system scheduling feature - see: *Scheduling*.

A scheduled insights sync can be also disabled and enabled using the command:

```
insights sync schedule <enable|disable> {sync-type}
```

7.10.2. Synced Automate Models

The list of Automate database models (first column) that are synced to the Insights database (third column) are listed below.

These models are also serve as query sources for VOSS Wingman in the VOSS Automate GUI and by some VOSS Automate dashboard widgets. See the topic on *VOSS Wingman* and *Dashboards* in the Core Feature Guide.

If any of these models are updated, the `delta-sync` sync process will create the matching INSERT, UPDATE or DELETE database operation in the Insights database, so that any widgets on VOSS Automate GUI dashboards that reference corresponding data, remains synced.

Table 1: Insights Sync Table

model_type	friendly_name	friendly_model_type
data/MonitoringCluster	Monitoring Platform Cluster	
data/MonitoringQueue	Monitoring Transaction Queue	
data/MonitoringSessions	Monitoring Sessions	
data/MetricDatabaseCollectionStats	Monitoring Database Stats	
data/LicenseAuditCounts	Automate License Counts	
data/BaseSiteDAT	Site	relation/HcsSiteREL
data/CountLimit	Subscriber Count Limit	
data/HcsDpDNE164AssociateDAT	E164 Number Association	
data/HcsDpE164InventoryDAT	E164 Number Inventory	data/HcsDpE164InventoryDAT
data/HcsEntitlementProfileDAT	Entitlement Profile	data/HcsEntitlementProfileDAT
data/HierarchyNode	Hierarchy	
data/InternalNumberInventory	Number Inventory	relation/NumberInventoryREL
data/MicrosoftSubscriberQasStaging	Microsoft Subscriber Staging	data/MicrosoftSubscriberQasStaging
data/User	User	relation/User
device/cuc/User	Cisco CUC User	relation/Voicemail
device/cuc/UserLicense	Cisco CUC User License	
device/cuc/Callhandler	Cisco CUC Call Handler	device/cuc/Callhandler
device/cucm/CallPickupGroup	Cisco UCM Call Pickup Group	device/cucm/CallPickupGroup
device/cucm/DeviceProfile	Cisco UCM Device Profile	relation/SubscriberDeviceProfile
device/cucm/Gateway	Cisco UCM Gateway	
device/cucm/GatewaySccpEndpoints	Cisco UCM Gateway SCCP Endpoints	

continues on next page

Table 1 – continued from previous page

model_type	friendly_name	friendly_model_type
device/cucm/HcsLicense	Cisco UCM License	
device/cucm/HuntList	Cisco UCM Hunt List	
device/cucm/HuntPilot	Cisco UCM Hunt Pilot	relation/HuntGroupRelation
device/cucm/LicensingResourceUsage	Cisco UCM Licensing Resource Usage	
device/cucm/Line	Cisco UCM Line	relation/LineRelation
device/cucm/LineGroup	Cisco UCM Line Group	
device/cucm/Phone	Cisco UCM Phone	relation/SubscriberPhone
device/cucm/PhoneType	Cisco UCM Phone Type	
device/cucm/RemoteDestination	Cisco UCM Remote Destination	
device/cucm/RemoteDestinationProfile	Cisco UCM Remote Destination Profile	relation/SingleNumberReachREL
device/cucm/User	Cisco UCM User	relation/Subscriber
device/pexip/Conference	Pexip Conference	
device/pexip/ConferenceAlias	Pexip Conference Alias	
device/msexchangeonline/UserMailbox	Microsoft Exchange Online User Mailbox	device/msexchangeonline/UserMailbox
device/msgraph/MsolAccountSku	Microsoft O365 User SKU	
device/msgraph/MsolUser	Microsoft O365 User	relation/MicrosoftSubscriber
device/msteamsonline/CsAutoAttendant	Microsoft Teams Auto Attendant	device/msteamsonline/CsAutoAttendant
device/msteamsonline/CsCallQueue	Microsoft Teams Call Queue	device/msteamsonline/CsCallQueue
device/msteamsonline/CsOnlineUser	Microsoft Teams User	device/msteamsonline/CsOnlineUser
device/msteamsonline/CsTeamsClientCo	Microsoft Teams Client Configuration	
device/spark/Announcements	Webex Calling Announcements	device/spark/Announcements
device/spark/AutoAttendants	Webex Calling Auto Attendants	relation/WebexAutoAttendants
device/spark/CallParkExtensions	Webex Calling Call Park Extensions	relation/WebexCallParkExtensions
device/spark/CallParkGroup	Webex Calling Call Park Group	relation/WebexCallParkGroup
device/spark/CallPickup	Webex Calling Call Pickup	relation/WebexCallPickup
device/spark/Device	Webex Calling Device	relation/WebexDevice
device/spark/HuntGroup	Webex Calling Hunt Group	relation/WebexHuntGroup
device/spark/Group	Webex Calling Group	device/spark/Group

continues on next page

Table 1 – continued from previous page

model_type	friendly_name	friendly_model_type
device/spark/License	Webex Calling License	device/spark/License
device/spark/Place	Webex Calling Place	relation/WebexTeamsPlace
device/spark/Number	Webex Calling Number	device/spark/Number
device/spark/Schedules	Webex Calling Schedule	relation/WebexSchedules
device/spark/Team	Webex Calling Team	device/spark/Team
device/spark/User	Webex Calling User	relation/SparkUser
device/uccx/Agent	Cisco UCCX Agent	relation/UccxAgent
device/webex/User	Cisco Webex User	device/webex/User

8. System Control

8.1. System Commands Overview

This section covers commands available from the CLI that are started with the **system** command prefix. The commands are generic, operating system type commands. System commands that start with the prefix **voss**, are VOSS Automate specific system commands.

Note that:

- The system commands that start with **system ssh_session_limit** are covered in the security topic on SSH session limits.
- The **system intrusion-detection** command is covered in the System Security section. (*System Intrusion Detection*)

8.2. System restart

The system can be restarted with **system reboot** and shutdown with **system shutdown**.

Note: In a cluster, reboot and shutdown of the entire cluster should be done on each node and not with the **cluster run all** command - see: *Remote Execution in Clusters*.

8.3. Passwords

The password for the platform user is chosen at install time, but can be changed using **system password** which will then prompt for the old password, the new password and confirmation.

Passwords be least 8 characters long and must contain:

- at least one upper case letter
- at least one lower case letter
- at least one number
- at least one symbol

User passwords must be changed at least every 60 days.

Additional users can be created with **user add <username>**.

Note: The <username> text needs to follow the rules below:

- starts with letter (a-z/A-Z)
- followed by one or more of:
  - letters (a-z/A-Z)
  - digits (0-9)
  - full-stop (.)
  - underscore (_)
  - dash (-)

Refer to the System Security : Creating additional users section.

Each user can be granted access to specific commands offering role-based access control - Refer to System Security : Granting and revoking user rights.

8.4. System Boot Passwords

Password protection can be enabled on the VOSS Automate boot loader configuration from the CLI. Commands are available to check, enable and disable the bootloader password.

- **system boot password** - Check if a bootloader password is enabled or disabled.
- **system boot password enable** - Prompts for and sets the platform user boot loader password. Refer to the topic on Passwords for password text requirements.
- **system boot password disable** - Disable the bootloader password if it is enabled.
- **system boot password reset** - If a password has been set, reset the bootloader password and enter a new password. If the system boot password is disabled when the command is run, a message will show this.

  For example:

```
$ system boot password
System boot password disabled.

$ system boot password reset
You are about to reset the boot password. Do you wish to continue? y
System boot password is disabled. Enable the system boot password first.

$ system boot password enable
You are about to enable the boot password. Do you wish to continue? y
Valid passwords must contain:
   at least one lower- and one upper-case letter,
   at least one numeric digit
   and a special character eg. !#@$%&^*
Please enter platform user boot password:
Password:
Please re-enter password
```

(continues on next page)

(continued from previous page)

```
Password:
System boot password enabled.

$ system boot password reset
You are about to reset the boot password. Do you wish to continue? y
status true
Valid passwords must contain:
   at least one lower- and one upper-case letter,
   at least one numeric digit
   and a special character eg. !#@$%&^*
Please enter platform user boot password:
Password:
Please re-enter password
Password:
System boot password enabled.
```

System boot passwords can also be enabled and set upon installation. Refer to the topic on Installation Details in the Installation Guide.

8.5. Federal Information Processing Standards (FIPS)

An administrator can check and enable the system for adherence to Federal Information Processing Standards (FIPS).

Note: FIPS is not supported in Automate release 25.1.

Important: The use of FIPS on the system requires a subscription to the *Ubuntu Pro* service package from Canonical in order to obtain the necessary cryptographic modules.

See https://ubuntu.com/pro/

A valid subscription to the Ubuntu UA service is required for *each individual node*. Commands also need to be run on *each node*.

Internet access will be required from your system - either directly, or via a proxy - to the necessary Ubuntu Pro service package URLs.

- All system passwords are stored using FIPS 140-2 complaint encryption algorithms, when FIPS mode is enabled or not.
- If FIPS is enabled on a system, all install scripts and templates are encrypted and decrypted using FIPS 140-2 complaint encryption algorithms.

To check the system FIPS status, use **system ua**.

```
platform@VOSS:~$ system ua status
SERVICE       AVAILABLE  DESCRIPTION
fips          yes        NIST-certified core packages
fips-updates  yes        NIST-certified core packages with priority security updates
```

(continues on next page)

(continued from previous page)

```
This machine is not attached to a UA subscription.
See https://ubuntu.com/advantage
```

The output above shows that services are available, but are not attached to the current node.

8.5.1. FIPS Enablement Steps

The step by step process to enable FIPS is as follows. Carry out the commands *on each node*:

1. *Configure the proxy access*
2. *Attach the node to the FIPS subscription*
3. *Enable FIPS Service*
4. Reboot the node
5. Repeat the above steps for all the nodes in the cluster

8.5.2. Configure the proxy access

Configure the proxy access if required, if the node is not set up to allowed to access the internet directly - for FIPS package retrieval.

Display the current proxy configuration:

```
platform@VOSS:~$ system ua config show
http_proxy              None
https_proxy             None
ua_apt_http_proxy       None
ua_apt_https_proxy      None
global_apt_http_proxy   None
global_apt_https_proxy  None
metering_timer          11000
```

Set a proxy:

```
platform@VOSS:~$ system ua config set http_proxy http://192.168.100.25:3128

http_proxy       http://192.168.100.25:3128
https_proxy      None
ua_apt_http_proxy   None
ua_apt_https_proxy  None
global_apt_http_proxy   None
global_apt_https_proxy  None
```

Unset a proxy:

```
platform@VOSS:~$ system ua config unset http_proxy

http_proxy       None
https_proxy      None
ua_apt_http_proxy   None
ua_apt_https_proxy  None
global_apt_http_proxy   None
global_apt_https_proxy  None
```

8.5.3. Attach the node to the FIPS subscription

Attach a node to the FIPS subscription with the command: **system ua attach**.

```
platform@VOSS:~$ system ua attach
You are about to attach this node to a UA account. Do you wish to continue? y
Please enter the UA account key
Key:
This machine is now attached to 'UA Infrastructure - Essential (Virtual)'

SERVICE       ENTITLED  STATUS    DESCRIPTION
fips          yes       disabled  NIST-certified core packages
fips-updates  yes       disabled  NIST-certified core packages with priority security␣
↪updates

NOTICES
Operation in progress: ua attach

Enable services with: ua enable <service>

                Account: My Account Name
           Subscription: UA Infrastructure - Essential (Virtual)
            Valid until: YYYY-MM-DD 00:00:00+00:00
Technical support level: essential

platform@VOSS:~$
```

Note:

- The entered value of `Key:` is not displayed.
- The heading now shows as `ENTITLED STATUS`.

To detach the UA subscription from a node, thus rendering the node disconnected from further updates, use the **system ua detach** command on the node.

```
platform@VOSS:~$ system ua detach
WARNING: Continuing with this command will render this node destroyed
```

(continues on next page)

(continued from previous page)

```
Do you want to continue? y
Detach will disable the following service:
    fips
Updating package lists
A reboot is required to complete disable operation.
This machine is now detached.

You have new mail in /var/mail/platform
platform@VOSS:~$
```

Important: After a node has been detached from the subscription, critical services will no longer be working on that node.

This command should only be used when the node is no longer in service. Should the node be removed by accident, the fail-over recovery process must be followed to replace that node. The previous instance will have to be detached by removing it on the Ubuntu Pro customer page.

8.5.4. Enable FIPS Service

After the FIPS subscription has been attached to a node, enable the selected `<service>` on the node: either `fips` or `fips-updates`.

Important: After running the **system ua enable <fips|fips-updates>** command, a node reboot is required.

- The enable process will take approximately 15 minutes for enabling `fips` per node.
- The enable process will take approximately 30 minutes for enabling `fips-updates` per node.

Only one of `fips` or `fips-updates` can be enabled. Once enabled, the selection cannot be changed.

The required security and versions of packages for FIPS are obtained and installed on the system.

The **STATUS** column shows the service status.

```
platform@VOSS:~$ system ua status
SERVICE       ENTITLED  STATUS    DESCRIPTION
fips          yes       enabled   NIST-certified core packages
fips-updates  yes       disabled  NIST-certified core packages with priority security␣
↪updates

NOTICES
FIPS support requires system reboot to complete configuration.

Enable services with: ua enable <service>

               Account: My Account Name
          Subscription: UA Infrastructure - Essential (Virtual)
           Valid until: YYYY-MM-DD 00:00:00+00:00
Technical support level: essential
```

(continues on next page)

(continued from previous page)

```
platform@VOSS:~$
```

8.5.5. Upgrading from Release 19.3.x with FIPS enabled

If FIPS was enabled a your system (release 19.3.x) *prior* to upgrade, note the following:

- Obtain and run `EKB-11024-19.3.4_patch.script`.
  1. On the Customer Portal, go to **Downloads > VOSS Automate > 19.3.4 > Patches > EKB-11024-19.3.4_patch**.
  2. Download `EKB-11024-19.3.4_patch.script` and follow installation instructions in `MOP-EKB-11024-19.3.4_patch.pdf`.
- After system upgrade, any existing FIPS setup is removed and FIPS needs to be re-enabled. No **system fips** commands are available - FIPS commands are replaced with **system ua** commands.
- After system upgrade and before re-enabling FIPS, the **voss upgrade_db** command cannot be used. A message shows:

```
This system was FIPS enabled previously. To proceed, please enable the Ubuntu Pro
program first before proceeding with the rest of the upgrade
To do this, run 'system ua attach' and 'system ua enable <fips|fips-upgrade>'
```

- Prior to FIPS re-enablement on an upgraded system, obtain the `UA account key` values for the nodes. These will be used when running **system ua attach**.

  System logs do not show entered key values - these are displayed as `XXXXXXX`.
- During upgrade from release 19.3.x, after the **cluster upgrade** and **cluster check** steps, run the *FIPS Enablement Steps*. Also refer to the Upgrade Guide for general upgrade steps.

8.6. File Management

Each user has a unique home directory in which local files can be stored. It is the user's responsibility to manage the disk space used by these files.

The command **diag disk** displays the disk usage. Files in the user's directory are displayed using the standard **ls** command, and deleted with **rm**.

New applications or upgrade packages are uploaded to the platform user using **scp** or **sftp**, for example **scp <filename> platform@192.168.0.1:** on the remote Unix file server. Refer to Network URI Specification for usage. If **sftp** or **ftp** is used, the following FTP servers are supported:

- OpenSSH for Unix or Linux based systems
- Titan SFTP and Cygwin (which is OpenSSH based) for Windows based systems.

A **sftp** or **scp** of files to VOSS Automate must be done in the `media` directory (`/opt/platform/admin/home/media`), which is a writable directory.

Alternatively a downloadable URL can be downloaded directly on the VOSS Automate system using **system download <URL>** and the downloaded file is placed in the platform user's directory, For example: **system download http://myserver/path/myfile.iso**

Individual applications are installed using **app install <filename>.script**. For details, see *Installing Applications*. A list of available applications and versions is displayed using the command **app list**.

ISO packages include all the individual packages required for upgrading. Upgrade the system using **cluster upgrade <filename>.iso**. Alternatively, the ISO package file system can be mounted with the system **mount** command, and the individual applications are visible under the media directory, and visible via the **app list** command.

On a running system, the **app list** command shows versions and details of applications and patches installed or not installed on the node where the command is run, as for example in this output snippet on a web proxy node:

```
...

mongodb - NoSQL database server
   not installed
   version available: 11.5.2 (2017-09-14 16:16)


nginx - Nginx web server
   latest version 21.1.0 (2021-07-11 16:01) is installed
   other version available: 11.5.2 (2017-09-14 16:15)


services - Platform core services
   latest version 21.1.0 (2021-07-11 16:07) is installed
   other version available: 11.5.2 (2017-09-14 16:15)


selfservice - selfservice install script
   not installed
   version available: 11.5.2 (2017-09-14 16:16)


voss-deviceapi - voss-deviceapi install script
   not installed
   version available: 11.5.2 (2017-09-14 16:16)

...
```

To remove an empty mount directory `media/<iso_file basename>` on nodes that may have remained after for example an upgrade, run:

cluster run all app cleanup

This command will also remove any application `.script` files in `media/` that have already been back up.

8.7. Drive control

In order to reduce the risk of *disk full* errors, the platform divides the file system over several disks keeping areas liable to grow outside the main root filesystem. The areas with the highest growth such as logs and database storage are kept on their own private file systems.

Note: The database mount point is stored in a logical volume.

These disk mounts can be migrated onto new, larger disks and some other locations can optionally be moved onto their own disks. This is managed through the **drives** command.

The current mounted filesystems and mount points can be displayed using **drives list mounted** and **drives list mountpoints** respectively.

A screen showing drives list mounted and drives list mountpoints:

```
platform@development:~$ drives list mountpoints
Used disks and mountpoints:
    sdc1 -  services:backups
    dm-0 -  mongodb:dbroot

Unused disks:
sde

Unused mountpoints:
    services:SWAPSPACE

Volume Groups
    voss - 25.0 GB free, 250.0 GB total
      Physical volumes:
        sdd1
      Logical volumes:
        dbroot/dm-0 - 225.0 GB

platform@development:~$ drives list mounted
Used disks and mountpoints:
    sdc1 -  services:backups
    dm-0 -  mongodb:dbroot
platform@development:~$
```

The mount points are as follows:

- `mongodb:dbroot` is the volume used for database storage
- `services:backups` is used for default backup storage
- `services:appdata` is the main system volume used for application data in the users account
- `services:SWAPSPACE` is the swap volume used by the system

Note: While the system is carrying out a backup, additional *Unused disks*, for example `dm-1`, `dm-2`, may show when the **drives list** command is run. These disks are used for snapshots and will not display once the backup is completed.

In order to add or extend an existing disk volume, follow the following steps:

- Under VMware, add an additional disk volume to the VM
- **drives list** displays any unused available volumes
- A free mount point can be linked to a new disk using **drives add <disk> <mountpoint>**.

Note: The **drive add** command on a Generic NBI node (for Billing Data Extract) is not in use.

- An existing used mountpoint (i.e. currently linked to a disk volume) can be linked to a new disk volume of greater size using **drives reassign <disk> <mountpoint> <new disk size in GB>**.

  The *<new disk size in GB>* specification means the volume need not be the size of the entire disk. However, a specified size must *not* be more than 90% of the disk size (or more than <disk size less 10GB> if the disk size is 100GB or smaller).

  Existing data on the current disk will be copied to the new disk volume, and once successful, the new disk volume will be linked.

For example, the following steps can be followed to add a 250GB hard disk to the system:

1. Log into the VMware console and select Server.
2. Right-click and select Edit settings.
3. Click **Add...** and select Hard Disk.
4. Step through the rest of the wizard and edit parameters - in this case 250GB, thick provisioned.
5. Once done, log into system as the platform user.
6. Carry out a disk listing with the command **drives list**.
7. Reassign the disks with the command:
   - For the *database mount point* (*mongodb:dbroot*), a Volume Group must be reassigned:
     a. **drives create_volume <volume_name> <new disk name>**.
     b. Carry out a disk listing to check the Volume Group with the command **drives list**.
     c. **drives reassign <volume_name> mongodb:dbroot <size in GB>**. The *<size in GB>* specification means the volume need not be the size of the entire disk. However, a specified size must *not* be more than 90% of the disk size (or more than <disk size less 10GB> if the disk size is 100GB or smaller).
     d. Old volumes can be removed with:
        – **drives remove_logical <volume_name> <logical_volume_name>**
        – **drives remove_volume <volume_name>**
   - For *other mount points*, a disk must be reassigned: **drives reassign <disk> <mountpoint>**
8. Start the application with **app start**.
9. Verify the new reassignment with the command **drives list**.

Note: Volume Groups for database mount points reserve a 10% or 10GB space - whichever is the largest - which is used for and then released during database backups.

SAN alignment is implemented using the offset value in **drives offset**. This value can be changed if necessary; however the default should be sufficient for most SAN hardware.

8.8. Transaction Prioritization

There are three buckets for transactions in the VOSS Automate system Priority Queue, namely high, medium and low priority:

1. High Priority:
    - Self-service transactions: carried out by end users on the Self-service interface
2. Medium Priority:
    - MACD operations on the VOSS Automate Admin Portal by Administrator users
    - API-based provisioning (HIL)
    - Any other transaction not in the Low Priority bucket
3. Low Priority:
    - Data Sync (LDAP, any device import transaction)
    - Bulk load transactions
    - Data import (import of data in JSON format)

From the Command Line Interface (CLI), a command is available to modify the default number of queue workers:

- Use **voss workers** to show the current number of queue workers. The default is 30.
- Use **voss workers <number>** set the number of queue workers.

An adjustment of the number of queue workers will impact on the number of parallel transactions that will run, which is a factor of this value as well the queue priority bucket to which the transaction belongs.

For example, Data Sync transactions may execute asynchronous workflows which are executed in parallel, or a Bulk Load transaction may have the Parallel flag set to True.

8.9. Banner

An administrator can manage a custom banner on the system from the CLI to display before login. The banner needs to be configured on a per node basis, in other words on each node in a cluster.

Banners are maintained on system update.

Banner text:

- Must be ASCII text in a UTF-8 file
- Can be up to 1600 characters. This includes spaces, tabs and other non-printing characters.
- Displays before login (SSH, SFTP, SCP)

An error message will display if the banner text is longer than the required length.

The following is an example:

```
$ ssh username@host

 ##########################################################################
 # WARNING: Unauthorized access to this system is forbidden and will be #
 # prosecuted by law. By accessing this system, you agree that your     #
 # actions may be monitored if unauthorized usage is suspected.         #
 ##########################################################################

 username@host's password:
```

The banner can be created in a file, uploaded to the system and then enabled.

1. Create the banner in a file, for example `banner`.
2. Upload the banner file, for example:

   ```
   scp banner username@host:media/
   ```

3. When logged in on the system, remove any previous banner and add the uploaded banner:

   ```
   system banner remove
   system banner add media/banner
   ```

4. The banner will then be shown as in the example above.

- To show the current banner, use: **system banner read** If no banner is available, a message will show.
- To remove the current banner, use: **system banner remove**

8.10. Maintenance Mode

CLI commands are available to also place the VOSS Automate system in a *maintenance mode* during system upgrade, to suspend any scheduled transactions. Scheduled transactions that are in progress, will be allowed to complete.

See also: *Clustering commands*.

8.10.1. Maintenance Mode Commands

- Put node or cluster into maintenance mode
  - For all nodes in a cluster: **cluster maintenance-mode start**

  Running this command prompts for confirmation and then shows:

  ```
  System is now in maintenance mode
  ```

  When the node or cluster is in maintenance mode:

  - Upon login on the CLI or when obtaining a health report, e.g. **diag health**, the message is then shown:

```
SYSTEM IN MAINTENANCE MODE
```

  - No scheduled Platform tasks will be started while in maintenance mode.
  - No scheduled Automate transactions will be started while in maintenance mode.

- Check the status of the maintenance mode
  - For all nodes in a cluster: **cluster maintenance-mode status**

  The command returns a message, for example

```
System is in maintenance mode
```

- Take node or cluster out of maintenance mode
  - For all nodes in a cluster: **cluster maintenance-mode stop**

  Running the command requests confirmation and then shows:

```
System is now out of maintenance mode
```

  - Scheduled tasks will resume once the system is out of maintenance mode.

8.11. Checksum

An administrator can generate the SHA256 checksum of a file such as an `.iso` image by using the **system checksum <path-to-file>** command.

The checksum of the file can then be compared to the one originally provided with the file, to verify its integrity.

8.12. Terminal Reset

The VOSS Automate platform provides a command to reset the terminal, in other words return the terminal session to sanity in for example the case where no characters show up on the terminal or where “gibberish” characters such as `^?` and `^H` are displayed.

The command is available from the **system** menu off the main CLI menu.

Run:

system tty sane

9. Diagnostics

9.1. Login Report

On login, the system displays a report indicating the status of the system before displaying the CLI user prompt. This report shows the following:

```
Last login: Mon Oct 31 13:45:17 UTC 2022 from 172.29.99.43 on pts/2

host: VOSS, role: webproxy,application,database, load: 0.58, USERS: 3
date: 2022-10-31 13:53:46 +00:00, up: 4:24
network: 192.168.160.3, ntp: 172.29.68.56
database: 3.5GB
WARNING: TRANSACTION DATABASE MAINTENANCE NOT SCHEDULED - SETUP SCHEDULE FOR REGULAR␣
↪MAINTENANCE
WARNING: CURRENT LICENSE FOR 6049914029c384002787ef2e (Automate) IS SET TO EXPIRE ON␣
↪2022-11-07


    mail - local mail management          keys - ssh/sftp credentials
 network - network management           backup - manage backups
    voss - voss management tools           log - manage system logs
database - database management          notify - notifications control
schedule - scheduling commands        selfservice - selfservice management
    diag - system diagnostic tools      system - system administration
    snmp - snmp configuration             user - manage users
 cluster - cluster management           drives - manage disk drives
     web - web server management             app - manage applications
 license - license management
```

The report explanation is shown below:

Name	Description
host	hostname
role	The role of the node: any of webproxy, application, database.
Last login	Last console login and IP address source. This is only shown if there has been a previous login.
load	The load average of the system.
USERS	The number of CLI users currently logged in. This is only shown if more than one user is logged in.
up	The system uptime.
services	The status of the system services.
HEALTH	A Health notification, for example a scheduled mail message, is set up or not.
database	Current database size.
DATABASE TRANSACTION SIZE	If the size of the TRANSACTION database collection exceeds 20GB, the current size is reported.
DATABASE TRANSACTION COUNT	If the number of entries in the TRANSACTION database collection exceeds 500,000, the current number is reported.
CURRENT LICENSE FOR <platform ID> (<product>)	INFO, WARNING and ERROR notifications on license status, raised according to number of days before expiry - see below.
TRANSACTION DATABASE MAINTENANCE NOT SCHEDULED	A schedule with either the command `voss transaction archive` or `voss transaction delete` is not set up.
Failed logins	If the user failed to log in prior to a successful login, the count, date and origin of the attempts are shown. A successful login resets this login count.
Please note: some checks took too long to complete and were terminated	If the health checks do not complete within 100 seconds, these are terminated with a message.

- disk, CPU and memory warnings are shown if applicable
- warnings are displayed in upper-case to draw attention
- DATABASE TRANSACTION warnings are shown for the following thresholds:
  - Transaction collection exceeds 500,000 documents
  - Transaction collection exceeds 20GB
- <INFO|WARNING|ERROR>: CURRENT LICENSE FOR <platform ID> (<product>) <IS SET TO EXPIRE|EXPIRED> ON <date>

  Note: The <platform ID> value is shown as platform ID on the Grace period license. After licensing, it is changed to the *platform description* as in the obtained license.

  The message levels are according to remaining number of days on the license, as follows:
  - INFO: 90 < remaining days <= 120
  - WARNING: 0 < remaining days <= 90
  - ERROR: 0 >= remaining days

Note:

- No messages are displayed while licenses have more than 120 days remaining.
- License status messages are ony displayed on
  - unified nodes in a Unified Node Topology
  - application nodes in a Modular Cluster Topology

For details on license management, refer to:

- *Product Licensing*
- The topic on Product License management in the Core Feature Guide.

A list of diagnostic tools is available in the topic on Diagnostic Tools.

- For schedule setups if the `HEALTH: NOT MONITORED` message is shown, see *Enable Health Monitoring*.
- For schedule setups if the `TRANSACTION DATABASE MAINTENANCE NOT SCHEDULED` message shows, see *Enable Database Scheduling*.
- See `diag health` at *Diagnostic Troubleshooting*.

9.2. Cluster Check

On a cluster, the **cluster check [verbose]** command is available to check:

- `network`: test and validate connectivity from each node to every other node, for each port required, as well as the time taken to connect to each node.
  - Checks for access to port `27020` on database hosts is not required from web proxy nodes.
  - Checks for access to port `443` is only required from web proxy nodes to unified nodes.
- `database`: carry out a check of database configuration
  - `info`: displays database weights and whether the node state is primary, secondary or arbiter
  - `error`:
    - if there is no connection to the database IP on a port
    - if the current database weight does not match the configured weight
    - if a node is marked as an arbiter but is not in the list of arbiters
  - `warn`: if the primary database node does not have the highest weight
- `disk`: carry out a drive space percentage check
- `ntp`: at NTP is functioning
- `packages`: Check status of packages installed by the system package manager. If an error occurs for a package, a message next to the package name shows: `package in an undesired state`.
- `security`: Check for security updates. Error status:
  - `info`: zero or one security update missed
  - `error`: more than one security update missed
- `cluster status`: also check the cluster status and

- `info`: show status as `OK`
- `error`: display a message to run **cluster status** for details
- `warn`: It is advisable that these be resolved prior to upgrading where possible. Some warnings may be resolved by upgrading.

Note: If *only* node versions mismatch or some nodes are missing components, a `warning` status is displayed. This status will allow for an upgrade of a node during failover recovery.

This caters for scenarios during repair/recovery of nodes. The **cluster check** will warn about version mismatches and not prevent upgrade commands. The cluster check cannot distinguish between whether a recovery process is ongoing or a general fault exists. When no node recovery process is ongoing, then the warning should be treated as an error and resolved before upgrade commences.

This command should also be run *before* carrying out a system upgrade.

Note: Without the `verbose` parameter, the **cluster check** command will *only show warnings and errors*. Otherwise it would only show the message `No issues found with host checks`.

Use the `verbose` parameter to see detailed output.

Example output (abbreviated):

```
$ cluster check
warn
   192.168.322.3:
       drives
          /: 47 % utilised
   192.168.322.5:
       drives
          /: 47 % utilised
   192.168.322.6:
       drives
          /: 47 % utilised

error
   192.168.322.3
       network
          => 192.168.322.4:27020: Failed
   192.168.322.4: Failed to connect to host
   192.168.322.5
       network
          => 192.168.322.4:27020: Failed
   192.168.322.6
          database
              arbiter: not configured
              weight: mismatched

[...]
   cluster
       status
           Error, please run `cluster status` for more information
```

Using the `verbose` parameter to see detailed output Any warnings and errors are then shown at the end of the verbose output.

Abbreviated example, `info` only; no issues:

```
$ cluster check verbose
info
    192.168.100.3
        database
            arbiter: ok
            state: ok
            weight: ok
        disk
            /: 28%
            /opt/platform: 27%
            /opt/platform/apps/mongodb/dbroot: 1%
            /tmp: 1%
            /var/log: 3%
        network
            => 192.168.100.4:8443: 0.223ms
            => 192.168.100.4:27020: 0.205ms
            => 192.168.100.5:8443: 0.246ms
            => 192.168.100.5:27020: 0.405ms
            => 192.168.100.6:8443: 0.169ms
            => 192.168.100.6:27020: 0.218ms
            => 192.168.100.7:8443: 0.225ms
            => 192.168.100.8:8443: 0.208ms
        ntp
            172.29.88.56: 18.313ms
        packages
            package database: ok
        security
            updates: 0 missed
    192.168.100.4
        database
            arbiter: ok
            state: ok
            weight: ok
        disk
            /: 28%
            /opt/platform: 27%
            /opt/platform/apps/mongodb/dbroot: 1%
            /tmp: 1%
            /var/log: 2%
        avx
          enabled
        network

    [...]
    cluster
        status
            OK
```

9.3. Enable Health Monitoring

Note: If this configuration is carried out, it should be set on each node.

Typically, each node will be configured with its own sender email address to identify the source node (for example `unified_node_1@customer.domain`).

Additionally,

- nodes can be configured with different emailrelay servers, for example, according to data center
- nodes can be configured with different SNMP servers and error levels

The steps below are to enable health monitoring if the system status displays `HEALTH: NOT MONITORED` upon login or when typing **help**.

1. Add an email relay address for outgoing email: add the SMTP IP address and optionally an allowed port (default port is 25):

   notify emailrelay <smtp ip address>[:port]

   For example:

   ```
   platform@host:~$ notify emailrelay 192.29.42.30
   emailrelay: 192.29.42.30
   ```

   The email relay can be verified with:

   notify emailrelay

   To disable or remove an external email relay or set the email relay back to default, use the following command:

   notify emailrelay 127.0.0.1

   The default `from` address is `platform@<hostname>`.

   A `from` outgoing address for the email relay can also be configured:

   notify emailfrom <from: email address>

   For example:

   ```
   notify emailfrom person@valid.domain
   ```

   Use the command **notify emailfrom** to list the currently configured `from` outgoing address.

2. Add a schedule instance for health reporting, for example with a schedule name `reports`:

   schedule add reports log send output mailto:user@server.com diag health

   For example:

   ```
   $ schedule add reports log send output mailto:user@server.com diag health

     Automatically setting time to midnight and enabling

     reports:
     active: true
   ```

(continues on next page)

(continued from previous page)

```
command: log send output mailto:user@server.com diag health --force
hour: 0
min: 0
```

Typing **help** at the command line will now *not* show the `HEALTH: NOT MONITORED` message.

The schedule instance can be modified, for example weekly on Sunday:

schedule time reports weekly 0

9.4. Enable Database Scheduling

If the following message is shown upon login or when typing **health login**, the steps below should be carried out:

- `TRANSACTION DATABASE MAINTENANCE NOT SCHEDULED`

For general information on scheduling, see: *Scheduling*.

For considerations and guidance on frequency of the schedule to set up for your system. see the System Maintenance section of the Best Practices Guide.

Important: Schedules must be set up on *all* unified nodes. This ensures that they still run in the event of a failover and DR scenario.

- For *transaction* database maintenance:

  With for example a schedule name `dbtxn`:

  schedule add dbtxn voss transaction archive 7

  or

  schedule add dbtxn voss transaction delete 7

  Note that transaction archiving also deletes transactions. For further details, see *Database Commands for Transaction Management*

  Typing **health login** at the command line will now *not* show the `TRANSACTION DATABASE MAINTENANCE NOT SCHEDULED` message.

  The schedule instance can be modified, for example weekly on Sunday:

  schedule time dbtxn weekly 0

9.5. Command History

A history of CLI commands issued can be displayed using **system history**. This command is the same as **log view platform/ui.log**, for example:

```
platform@development:~$ system history
Aug 15 10:55:44 node00 ui[6348]: app install support_install.script
Aug 15 10:56:26 node00 ui[11345]: diag
Aug 15 10:56:27 node00 ui[11351]: app install security_install.script
Aug 15 10:59:57 node00 ui[11609]: security
Aug 15 11:00:23 node00 ui[11704]: notify list
Aug 15 11:00:28 node00 ui[11892]: backup list
Aug 15 11:01:23 node00 ui[12483]: ssl
Aug 15 11:03:12 node00 ui[13562]: drives add sdd mongodb:dbroot
Aug 15 11:07:21 node00 ui[16568]: voss
Aug 15 11:09:34 node00 ui[20740]: snmp
/var/log/platform/ui.log
```

Press **q** to exit the file.

9.6. Logs

The system maintains a comprehensive list of logs under `/var/log`:

- The `platform/` directory has logs pertaining the general platform or to specific log types.
  - `apps.log` contains application and process control logging
  - `audit.log` a log type available if enabled with **log audit locallog on** - contains audit log information
  - `database.log` contains database logs spawned by VOSS Automate transactions
  - `dockerd.log` contains logs spawned by the Docker container management daemon
  - `backup.log` contains all logging pertaining to backups - available after the first backup
  - `cluster.log` contains all control level management of the cluster
  - `cluster_check.log` contains output of **cluster check** command. Run **cluster check** to have the latest cluster check information logged to `cluster_check.log` - see: *Cluster Check*.
  - `config.log` contains information relating to the platform-level configuration
  - `event.log` a log type available if enabled with **log event locallog on** - contains event log information
  - `execute.log` contains low-level information about command execution
  - `notifications.log` contains information relating to SNMP notifications
  - `reports.log` contains information relating to system reports. Refer to the Scheduling section on how reports can be created.
  - `security.log` contains information relating to user security
  - `security_install_GUI.log` contains information relating to user graphical interface security

  - `security_update.log` contains information relating to security updates
  - `ui.log` contains higher-level information relating to UI commands being executed.
  - `wsgi.log` contains information relating to API-level commands via the WSGI server
- The `provision/` directory contains logs relating to provisioning. Every module provision is logged to component log files.
- The `health/` directory contains health logs. These are stored automatically every half hour, or whenever health is run, and are of the format `health/summary_report-<date>-<time>`.
- The `process/` directory contains process logs instrumental in debugging particular processes. All of the output from each process is logged to an individual file `process/<application>.<process>.log`
- The `install/` directory contains logs detailing the install process.
- The `mongodb/` directory contains logs relating to the Database function.
- The `nginx/` directory contains logs relating to the WebProxy function.
- The `voss-deviceapi/` directory contains logs relating to the Application function. For example, `voss-deviceapi/cnf_collector.log` is the Change Notification Collector log.
- The `billing-data-extract/` directory contains service logs relating to the Billing Data Extract module, if installed. For example, `callback.log`, `extractor.log`, `notifier.log`, `scheduler.log`.
- The `nbi-api/` directory contains logs relating to the Billing Data Extract module API, if installed.
- The `voss-portal/` directory contains nginx logs relating to the portal interface.
- The `insights-voss-sync/` directory contains real time and transaction logs relating to the sync of data for insights
- The `insights-api/` directory contains logs for the insights API.

log list [<search_string>] is used to display a list of logs, optionally matching `search_string`. For example:

```
platform@clusternode:~$ log list alternatives.log
selfservice/alternatives.log
voss-deviceapi/alternatives.log
nginx/alternatives.log
mongodb/alternatives.log
alternatives.log
```

The main log rotation scripts will rotate log files only when files exceed 100M or if the disk containing `/var/log/` is over 80% full. This is checked once per hour. The system will attempt to keep 5 historic zipped files of each log. If the disk containing `/var/log` is over 90% full, files will be purged to ensure that the system continues to function.

- 100M size logs:
  - `mongodb/*.log`
  - `nginx/*.log`
  - `selfservice/*.log`
  - `voss-deviceapi/*.log`
  - `insights-voss-sync/*.log`
- 10M size logs, with 7-day log rotation:
  - logs in `/var/log/process/*log` and `/var/log/platform/*log`

- 35-day log rotation:
  - `/var/log/syslog`
- 1-hour log rotation:
  - `insights-voss-sync/*.log`

All rotated log files and log files exceeding 1GB can be purged manually with **log purge**.

9.7. Viewing Logs

Once a file name is known, the particular log can be viewed with **log view <logfile>**. For example: **log view process/mongodb.router.log**.

When the log file is viewed, it can be searched for a particular regular expression using the forward slash /, as when using the **less** command.

A log file can also be watched or followed with **log follow <logfile>** The Unix command equivalent is: **tail –f**, so quit with **Ctrl-C**.

Log entries in the `voss-deviceapi/` directory have key-value pairs. The keys are as follows:

- `hostname` - the hostname of the server
- `level` - debug level
- `message` - the actual log message
- `name` - module where log occurred
- `parent process id` - Linux process parent id
- `process id` - Linux process id
- `request uuid` - identifier to group all logs generated in a request.
- `user` - user that generated the log
- `user hierarchy` - user that generated the log's hierarchy.
- `txn_id` - in logs that generate transactions (e.g. not in `request.log`) - the transaction uuid
- `txn_seq_id` - in logs that generate transactions (e.g. not in `request.log`) - transaction ID as seen in the Admin Portal
- `toplevel_txn_seq_id` - in logs that generate transactions (e.g. not in `request.log`) - toplevel transaction ID

Note that the system will attempt to auto-complete the prefix if it uniquely identifies a file, for example:

log view process/nginx

9.8. Sending and Collecting Logs

- **log send**

  Single or multiple log files can be sent to a destination using **log send <destination> <logfile>** and **log send <destination> <prefix>** respectively.

  The <destination> must match one of:

  - scp
  - sftp
  - mailto

  as in the URI specification detailed under *Network URI specification*.

  An example of an email URI is mailto:user@server.com.

  All email communication requires **notify <smtp ip address>[:port]** to be configured with the IP address of your mail relay and optionally an allowed port (default port is 25).

  The default from address is platform@<hostname>. A from outgoing address for the email relay can also be configured with **notify emailfrom <from: email address>** - see: *Enable Health Monitoring*.

  When using **log send** to a sftp destination, no port should be specified.

  For example:

```
$ log send sftp://usr:pass@172.21.21.122/ install/voss-deviceapi_install.script-
↪20150819.log
```

- **log send output**

  The output of a VOSS Automate CLI command can be sent to a destination using

  log send output <destination> <CLI command>

  The <destination> must match one of:

  - scp
  - sftp
  - mailto

  as in the URI specification detailed under *Network URI specification*.

  For example:

```
$ log send output sftp://usr:pass@172.21.21.122/ app status
File transfer successful - 172.21.21.122:None/VOSS_1558945096-combined_logs.tar.gz
```

  The transferred file (archive file format example: var/log/users/platform/<command>.<yyyymmdd>) then contains the output of the CLI command.

  Note that *only VOSS-4UC CLI commands* will generate a file with command output. For example, while the command was **ls media** can be run from the VOSS Automate CLI prompt, it is not a VOSS Automate command and the contents of the transferred lsmedia.<yyyymmdd> file will be command not found: ls.

- **log sendnewer**

  Log files newer than a certain date can be sent using **log sendnewer <yyyy-mm-dd> <destination>**.

  The `<destination>` must match one of:

  - `scp`
  - `sftp`
  - `mailto`

  as in the URI specification detailed under *Network URI specification*.

  If the remote URI destination requires a password, it will prompt for the password.

  A passwordless **scp** session can be enabled by generating keys locally with **keys create** and then sending the local keyset to the remote destination with **keys send user@<hostname>**.

Note: For `log send` and `log sendnewer` commands where the destination is an email address, log files that are too large for email attachments will be excluded and indicated in a message, for example:

```
[...]

Attempting to send log file/s.
Sending VOSS_1747219685-voss-deviceapi.voss-risapi_collector.log.gz [OK]
Attempting to send log file/s.
Sending VOSS_1747219686-services.firewall.log.gz [OK]
The following files were too large to send via email. They have been moved to the home␣
↪directory:
/var/log/journal/04c865b1128/system@ba414bfce.journal
/var/log/journal/04c865b1128/system@b476447c5.journal
/var/log/journal/04c865b1128/system@bf2e26fcd.journal
Please use the 'scp' function to send the files to a remote destination.
```

- **log collect**

  Use **log collect** to collect logs into an archive file. Both system and transaction logs can be collected. Mandatory and optional parameters are available to refine the log collection.

  The syntax is:

  log collect start <start-time> [include <logs|db|all>] [end <end-time>] [limit]

  Note:

  - Run **cluster check** before **log collect** to have the latest cluster check information logged to `cluster_check.log`.
  - If the command is run from a web proxy node, only the system logs can be collected.
  - The `start` and `end` parameters do not affect date range of system logs - they only apply to the date range of transaction collection logs.

  - The `start` parameter is mandatory.
  - The `end` parameter is optional.

    If omitted, the transaction collection logs are collected up to the current time.

- The <start_time> and <end_time> date value format can be:
  - +%Y-%m-%d_%H:%M:%S, for example 2016-01-10_00:00:00
  - +%Y-%m-%d, for example 2016-01-10
    - For a <start_time>, if the value is entered as +%Y-%m-%d, then the value used will be the start of that day, in other words: +%Y-%m-%d_00:00:00.
    - For an <end_time>, if the value is entered as +%Y-%m-%d, then the value used will be the end of that day, in other words: +%Y-%m-%d_23:59:59.
  - Year-Month-Day format without leading zeroes, for example: 2016-1-10
- The include parameter is optional. If *not* used, both transaction collection and system logs are collected.

  If used, valid options are:
  - logs: collect database server, system, application and install log files:
    - db_server_status.log - database server status: the output of mongo db.serverStatus()
    - disk_usage.log - disk usage: output of command df -h
    - top.log - current CPU consumption per process: output of command top -b -n 1 -o +%CPU
    - sar<DD> - today's system activity by the system activity reporter sar: output of sa2 -A
    - db_collection_stats.log - database collection stats: output of CLI command **voss db_collection_stats all**
    - db_index_stats.log - database index stats: output of CLI command **voss db_index_stats**
    - from the /var/log directory:

      Log files with the following file wild cards are collected:

      ```
      /var/log/syslog*
      /var/log/dmesg*
      /var/log/psmem*
      ```

      Log files in the following directories are collected:

      ```
      /var/log/nginx
      /var/log/sysstat
      /var/log/platform
      /var/log/process
      /var/log/voss-deviceapi
      /var/log/provision
      /var/log/install
      /var/log/mongodb
      /var/log/voss-portal
      /var/log/billing-data-extract (if available)
      /var/log/nbi-api (if available)
      /var/log/m2uc-nx (if available)
      ```

You can inspect the list of collected files with the **log list** command and a search parameter, for example **log list install** to see all the install log files, including those for patches, for example:

```
platform@VOSS:~$ log list install
install/voss-deviceapi_install.script-20210709.log
install/disable-platform-password-expiry_patch.script-20210709.log
install/selenium_install.script-20210709.log
install/support_install.script-20210709.log
install/mongodb_install.script-20210709.log
install/nginx_install.script-20210709.log
install/snmp_install.script-20210709.log
...
```

This option *excludes* the transaction collection. The `start` and `end` parameters do not affect this collection.

* `db`: collect transaction collection logs and exclude the system logs. By default, this includes:
  - transaction activity log records (`TRANSACTION.json.gz`)
  - the detailed content of transactions as seen on the GUI in the Log transaction list (`TRANSACTION_LOG.json.gz`).
  - worker transaction queue activity log records (`WORKER_QUEUE.json.gz`)

  This option can only be added when the command is run on a unified or database node.
* `all`: *both* transaction collection (`db`) and system logs (`logs`) are collected

– The `limit` option *only* affects the transaction collection logs. It specifies that the detailed transaction collection logs (`TRANSACTION_LOG.json.gz`) are *not* included.

  This parameter is usually used if the logs are required for a task such as performance analysis, but not for debugging. Typically, all collection logs are needed for debugging.

Important:

– The *log collect* command with default parameters should not be run in parallel on all nodes.

– Transaction database data should not be collected from secondary nodes. So the commands issued on a primary would be different from that on secondaries, e.g.:

  Primary

  `log collect start 2020-04-10 end 2020-04-14`

  Secondaries

  `log collect start 2020-04-10 include logs end 2020-04-14`

An example of the console input and output of the command is shown below:

```
$ log collect start 2019-08-27 include db limit
2019-08-27T08:16:02.140+0000        connected to: localhost:27020
2019-08-27T08:16:02.148+0000        exported 3 records
Output saved to media/logs.VOSS.2019-08-27_08-16-01.tar.gz
```

The log file archive is of the format: `logs.<hostname>.<timestamp>.tar.gz`, where `<timestamp>` is the time that the collection was requested, in the format: `%Y-%m-%d_%H-%M-%S`. This file is created in the `media/` directory.

The log file archive can then for example be fetched with **scp**, for example:

scp platform@VOSS:media/logs.VOSS.2019-08-27_08-16-01.tar.gz

9.9. Log Types

The VOSS Automate system can log records of certain types, that can be logged locally or remotely. The log types contain events or transactions that originate from the:

- User interface
- API
- Command Line Interface (CLI)
- System activity (for example database connections)

The minimum specifications of the remote system for audit and event logs are:

- 2 VCPU's
- 80 GB HDD
- 2GB RAM

Log types:

1. Audit

Important: The available types of audit logs are determined by the audit log rule set that is active - see: *Audit Log Rule Sets*

The details below show details on the contents of types of audit logs.

- On the Admin Portal and Self-service GUI and API:
  - login and logout attempts (successful and unsuccessful) and session login time, logout time and expire events using any of the authentication methods:
    - SSO
    - LDAP
    - VOSS Automate

    Expired sessions will only be logged at 5 minute intervals.
  - User account creations, modifications, disabling, and termination events. This means all create, update, delete operations on the `data/User` data model.

    User modifications include user move operations from one hierarchy to another.

    In particular, operations on the list of VOSS Automate models or attributes below, for: add, modify and delete.
    - `data/Role`
    - `data/AccessProfile`

        * `data/User.role`
        * `data/CredentialPolicy`

      Note that these operations on any created models that refer to these core models are not logged.

  - On the Command Line Interface (CLI):
    - login and logout attempts (successful and unsuccessful) and session login time, logout time and expire events; and also including:
      * root shell login and logout using the nrs script
      * ssh
      * scp
      * sftp
    - All root shell CLI commands are logged.
    - All CLI commands are logged. The audit log will show “CLI” or “Cluster” depending on how command was run.

      For the creation of schedules (using **schedule**), these are logged, but the scheduled commands are not logged when they execute.

      This includes for example user account creations, modifications, disabling, and termination events commands from the CLI:
      * **user add**
      * **user del**
      * **user grant**
      * **user revoke**
      * **user list**
      * **voss unlock_sysadmin_account**
      * See: *Audit Log Format and Details*

2. Event
   - All transactions, sub-transactions as well as their details as seen when viewing the Transaction Log in the GUI.

     Note that the detailed logs are not recorded. In other words, the rows of entries under the Logs table of a transaction as seen in the GUI under Administration Tools > Transaction are not shown in the event log, since the primary purpose of the log is auditing: “who did what”.
   - See *Event Log Format and Details*
3. Stream
   - Refers to the method of distribution of selected log files to a syslog server.
   - See: *Log Streaming*

9.10. Audit Log Rule Sets

Audit log rule sets are available to manage the level of detail in audit logs. Types of logs can be added to or removed from rule sets by means of **log audit ruleset** command line parameters.

The following table shows rule sets and their default state:

Table 1: Rule Sets

Option	Name	Enabled
1	Default Rules	true
2	CLI Commands	true
3	Users and Groups	false
4	Network Events	false
5	Security	false
6	Software Management	false
7	Root Commands	false
8	File Access	false

For details on the logs associated with the rules, see:

- *Types of command and change logs in audit rules*
- the audit log description under *Log Types*

This means that by default, the audit log only shows logs associated with the default audit rules (1) and any VOSS Automate platform CLI commands (2).

The following parameters are available for the command **log audit ruleset**:

- **log audit ruleset list**

  Show the current ruleset, in other words the enabled and disabled rules.

  For example, consider the following (non-default option 7 has been enabled):

```
$ log audit ruleset list


              Option    Name
              ------    ----

 Rules Enabled
                   1    Default Rules
                   2    CLI Commands
                   7    Root Commands

Rules Disabled

                   3    Users and Groups
                   4    Network Events
                   5    Security
```

(continues on next page)

(continued from previous page)

```
                  6    Software Management
                  8    File Access
```

- **log audit ruleset disable 1,2**

  Disable rules 1 and 2 from the rule set.

  Note: The parameter syntax is a comma separated list of option numbers *without* spaces.

  For example:

```
$ log audit ruleset disable 1,2

$ log audit ruleset list


              Option    Name
              ------    ----

 Rules Enabled
                  7    Root Commands

Rules Disabled

                  1    Default Rules
                  2    CLI Commands
                  3    Users and Groups
                  4    Network Events
                  5    Security
                  6    Software Management
                  8    File Access
```

- **log audit ruleset enable 2**

  Enable rule 2 from the rule set.

  For example:

```
$ log audit ruleset enable 2

$ log audit ruleset list


              Option    Name
              ------    ----

 Rules Enabled

                  2    CLI Commands
                  7    Root Commands

Rules Disabled
```

(continues on next page)

(continued from previous page)

```
                  1    Default Rules
                  3    Users and Groups
                  4    Network Events
                  5    Security
                  6    Software Management
                  8    File Access
```

- **log audit ruleset enable all**

  Enable all the rules.

  For example:

```
$ log audit ruleset enable all

$ log audit ruleset list


              Option    Name
              ------    ----

 Rules Enabled

                   1    Default Rules
                   2    CLI Commands
                   3    Users and Groups
                   4    Network Events
                   5    Security
                   6    Software Management
                   8    File Access
                   7    Root Commands

Rules Disabled
```

- **log audit ruleset default**

  Reset the rules to the default set.

  For example:

```
$ log audit ruleset default

$ log audit ruleset list


              Option    Name
              ------    ----

 Rules Enabled

                   1    Default Rules
                   2    CLI Commands
```

(continues on next page)

(continued from previous page)

```
Rules Disabled

                    3    Users and Groups
                    4    Network Events
                    5    Security
                    6    Software Management
                    8    File Access
                    7    Root Commands
```

9.10.1. Types of command and change logs in audit rules

Option #	Name	Purpose
1	Default Rules	Audit mgt tool, kernel , mount, swap, stunnel, cron events
2	CLI Commands	All Voss CLI commands logged in a clear text format
3	Users and Groups	User, group, sudo, password, login/logout events
4	Network Events	Hostname, pam, ssh, systemd, access failures, power state, session initiation, access control, etc
5	Security	Suspicious activity, reconnaissance, code injection, and privilege abuse
6	Software Management	Package management (dpkg, apt, aptitude)
7	Root Commands	Commands executed as root (high volume)
8	File Access	File access failures and deletion

9.11. Log Type Commands

The **log** command takes a log type parameter, as can be seen from the command syntax `[audit|event|stream]`:

```
$ log
USAGE:
------
log [audit|event] locallog on|off                - Enable or disable audit/event logging
log [audit|event|stream] remotelog               - Get the config for remote system␣
↪logging
log [audit|event|stream] remotelog <IP:port>|off - Configure a remote system for sending␣
↪logs
log [audit|event|stream|ssl] status              - Get the status for audit/event/stream/
↪ssl logging
```

For an overview of the log types and formats, see:

- *Log Types*
- *Audit Log Format and Details*

- *Event Log Format and Details*
- *Log Streaming*

Note: Audit log details are determined by the audit log ruleset - see: *Audit Log Rule Sets*.

To enable or disable local audit and event logging, use the command and its respective option:

- **log audit locallog on|off**
- **log event locallog on|off**

Important: In a clustered environment, logging should be enabled or disabled on all application nodes in order to generate or stop logs completely, since a single transaction queue is utilized in the cluster and transactions can run on all application nodes. For commands on a cluster, see the **cluster run** command: *Remote Execution in Clusters*.

If local logs are enabled, local log files of the type are available:

- Audit log files can be viewed as with all logs: **log view platform/audit.log**
- Event logs: **log view platform/event.log**

To enable remote logs of a type requires a remote system IP address and port as input parameters. The location and format of the logged data on the remote system would depend on the syslog application being used and the configuration of that application.

For remote system requirements, see: *Log Types*.

Note:

- When audit or event logging is enabled or disabled locally or remotely, the syslog service restarts.
- When stream logging is enabled or disabled, the syslog service restarts.

Remote log type disable CLI output example:

```
$ log audit remotelog off
You are about to restart syslog. Do you wish to continue? yes
You have new mail in /var/mail/<username>
```

The log type status for *both* local and remote logging can be checked with: **log audit status** or **log event status**, for example:

```
$ log audit status
     audit:
         ip: 112.19.42.249:10514
         locallog: true
```

To check *only the remote* logging status of a log type: **log audit remotelog** or **log event remotelog**, for example:

```
$ log audit remotelog
      ip: 112.19.42.249:10514
```

Note:

- The internal rsyslog statistics are checked every 60 seconds to detect failed actions. If a failure is detected, the failure notification is retransmitted every 10 minutes.
- If the remote syslog server stops receiving logs, an email message or SNMP trap is generated, with the email message:

```
Subject: Log processing failure

Message: System unable to send <event type> messages to <IP>
```

  In the case of an SNMP trap:

```
mteHotTrigger: Log processing failure

mteHotContextName: System unable to send <event type> messages to <IP>
```

- If the remote syslog server stops receiving logs, the local disk space of the queue of logs can grow to a maximum of 1GB before logs are not queued and log messages are discarded.

See *Warnings and Notifications* to set up the notification.

9.12. Audit Log Format and Details

The following is the format of an audit log entry. Line breaks have been added here for readability.

```
%b %d %Y %H:%M:%S.%f %Z|
UserID : %s
ClientAddress : %s
Severity : %s
EventType : %s
ResourceAccessed: %s
EventStatus : %s
CompulsoryEvent : No
AuditCategory : %s
ComponentID : VOSS Automate
AuditDetails : %s
App ID: %s
```

The first entry is the string format of the timestamp, while the `%s` is a variable for a value.

An example of the timestamp would be:

```
Oct 23 2015 10:54:28.615377 UTC
```

- Audit logs include logs for `auditd` and `audispd` which include system events. If system events are not required, they must be filtered by the client.
- All remote syslog streaming from VOSS Automate is via TCP. UDP is not supported.

The tables below show key and example descriptions in the audit log.

UserID	Username
johnB	Username on CLI or database
johnB prov1.cust1	GUI username and hierarchy
ProviderUser@Provider.com	User email address from GUI login
hidden	Invalid username

ClientAddress	IP address / pseudo terminal
102.29.232.50:/dev/pts/1	From IP: 102.29.232.50 and pseudo terminal /dev/pts/1
127.0.0.1	Internal API user
102.29.232.50	IP of GUI or API. Also Bulk Load, JSON import.

Severity	0-2. Higher is more severe
0	Basic log activity on the CLI. All log activity on the GUI or API.
1	All Rootshell activity
2	CLI: AuditCategory : Priviliged, AuditDetails : user list and App ID: CLI - user may not run **user list** command

EventType	Type of event
UserLogging	Login, logout, expiry activity
FileDetection	File checksum activity
<AuditCategory>	GUI or API event type is the AuditCategory

ResourceAccessed	Resource accessed
CLI	CLI transaction
DB	Database logging
Application REST API	GUI or API resource

EventStatus	Status of the event
Success	Successful transaction
Failed	Failed transaction
Unknown	Note: Mongo successful login has this status

CompulsoryEvent	Not in use
No	Currently always No

AuditCategory	Activity category
AdministrativeEvent	non-privileged CLI command
Privileged	CLI transactions as root user, and commands by any user from the list below.
SecurityEvent	Login or logout to CLI, database,
PrivilegedDataModelAdd	e.g. GUI or API system user, including the type and operation. Type can also be Mod and Del. Details in AuditDetails.
DataModelAdd	e.g. GUI or API ordinary user, including the type and operation. Type can also be Mod and Del. Details in AuditDetails.
UserRoleChange	Transactions on the GUI, API flagged as privileged, including the type and operation. Details in AuditDetails.
UserLogin	Login on the GUI, API.
UserLogout	Logout on the GUI, API.
MultipleSourceLogin	Simultaneous login on GUI, API. Multiple sources in AuditDetails.

The CLI commands that are flagged as Privileged, are:

- **user** (and any parameters, such as **user del**)
- **voss unlock_sysadmin_account**
- **voss cleardown**
- **system password**
- **system reboot**
- **system shutdown**

The GUI and API commands flagged as privileged, are:

- carried out by a system user
- operations on the models:
  - data/AccessProfile
  - data/CredentialPolicy
  - data/HierarchyDefault
  - data/Role
  - data/User
  - data/Settings
  - data/Application
  - data/UnityConnection
  - data/CallManager

- `data/AuthorizedAdminHierarchy`

Audit Category for GUI and API transaction on a data model can be: *[Privileged]DataModel(Add|Delete|Update)*

ComponentID	Identifier
`VOSS Automate`	The value is always `VOSS Automate`

App ID	Application
`VOSS Automate`	The application GUI and API interface
`CLI`	CLI command
`VOSS Automate CLI`	Rootshell login
`VOSS Automate SSH`	SSH login
`VOSS Automate DB`	Database, for example Mongo connect, login, logout

Audit Details	Details of transaction
Login	CLI or database login
Login from 172.29.232.88	GUI or API login also shows IP address
Logout	CLI or database logout
Login Invalid User	CLI or database login
Login Invalid Password	CLI or database login
User account locked - {} / {}	CLI or database login. Account locked after failed_login_attempts / allowed_attempts
User account expired	CLI or database login. Account expired
RootShell login	Root shell login
RootShell logout	Root shell logout
File checksum initialized	File checksum process initialized. The EventType is FileDetection.
<CLI command>	The CLI command that is run
Resource type data/User named User Name: Joe	Example of a create transaction on the data/User model.
User Joe role updated to admin	Example of a role update on a user.
Login failed with Unknown from 172.29.232.88	
[Basic|NonInteractive|SSO|LDAP] Authentication on Log [in|out]	Login or log out by a user using the indicated credentials (Basic, NonInteractive, SSO, LDAP). The log entry includes Client Address for source of the login.
Session Expired	Session timeout
Permission Error	Access control error: the user has no permission for an operation on a resource type from a hierarchy.
Invalid Request	If the request URL is not found (HTTP response is 400, 404)
Password retry limit reached. Locking account with username ..	When an account is locked due to failed password attempts
Unlocking account with username ..	When an account is unlocked
Locking account with username ..	When an account is locked

9.12.1. Example Syslog Messages

The following are example audit log entries.

Note: Line breaks have been added for readability.

```
API,Login,2019-10-29T21:11:20+00:00 VOSS audit: Oct 29 2019 21:11:20.042962 UTC|
UserID : CS-PAdmin
ClientAddress : 172.29.90.25
Severity : 0
EventType : UserLogin
ResourceAccessed : Application REST API
EventStatus : Success
CompulsoryEvent : No
AuditCategory : UserLogin
ComponentID : VOSS Automate
AuditDetails : Login with Mongo from 172.29.90.25 using interface None
App ID: VOSS Automate

API,Logout,2019-10-29T21:11:11+00:00 VOSS audit: Oct 29 2019 21:11:11.449544 UTC|
UserID : CS-PAdmin
ClientAddress : 172.29.90.25
Severity : 0
EventType : AuthLogout
ResourceAccessed : Application REST API
EventStatus : Success
CompulsoryEvent : No
AuditCategory : AuthLogout
ComponentID : VOSS Automate
AuditDetails : Logged out from 172.29.90.25
App ID: VOSS Automate

API,Access Control Bypass,2019-10-29T21:14:36+00:00 VOSS audit: Oct 29 2019 21:14:36.
↪016777 UTC|
UserID : CS-PAdmin sys.hcs.CS-P
ClientAddress : 172.29.90.25
Severity : 0
EventType : PermissionError
ResourceAccessed : Application REST API
EventStatus : Failed
CompulsoryEvent : No
AuditCategory : PermissionError
ComponentID : VOSS Automate
AuditDetails : Read operation on model type data/Countries
App ID: VOSS Automate

API,Data Model Add,2019-10-29T21:31:33+00:00 VOSS audit: Oct 29 2019 21:31:33.872904 UTC|
UserID : CS-PAdmin sys.hcs.CS-P
ClientAddress : 172.31.252.1
Severity : 0
EventType : DataModelAdd
ResourceAccessed : Application REST API
EventStatus : Success
CompulsoryEvent : No
AuditCategory : DataModelAdd
ComponentID : VOSS Automate
   AuditDetails : Resource type data/Role named
Name: Test
```

(continues on next page)

(continued from previous page)

```
App ID: VOSS Automate

CLI,User Add,
"2019-10-29T21:45:42+00:00
VOSS audispd:
  node=VOSS
  type=ADD_GROUP
msg=audit(1572385542.608:242353):
  pid=421859
  uid=0
  auid=1401
  ses=4
  msg='op=adding group acct=""testuser"" exe=""/usr/sbin/useradd"" hostname=? addr=?␣
↪terminal=pts/0 res=success'

2019-10-29T21:45:42+00:00
VOSS audispd:
  node=VOSS
  type=USER_CHAUTHTOK
msg=audit(1572385542.736:242401):
  pid=421872
  uid=0
  auid=1401
  ses=4
  msg='op=PAM:chauthtok acct=""testuser"" exe=""/usr/sbin/chpasswd"" hostname=? addr=?␣
↪terminal=? res=success'

2019-10-29T21:45:42+00:00
VOSS audispd:
  node=VOSS
  type=PATH
msg=audit(1572385542.764:242413):
  item=0
  name=""/opt/platform/users/testuser""
  inode=1654786
  dev=08:12
  mode=040700
  ouid=0
  ogid=0
  rdev=00:00
  nametype=NORMAL

2019-10-29T21:45:42+00:00
VOSS audispd:
  node=VOSS
  type=PATH
  msg=audit(1572385542.768:242417):
    item=0
    name=""/opt/platform/users/testuser/media""
    inode=1654788
    dev=08:12
```

(continues on next page)

(continued from previous page)

```
    mode=040500
    ouid=0
    ogid=0
    rdev=00:00
    nametype=NORMAL


2021-05-26T15:27:33.715215+00:00 VOSS audit: May 26 2021 15:27:33.714993 UTC|
UserID : system
ClientAddress : 172.29.90.57
Severity : 0
EventType : SecurityEvent
ResourceAccessed : Application REST API
EventStatus : Failed
CompulsoryEvent : No
AuditCategory : SecurityEvent
ComponentID : VOSS Automate
AuditDetails : Password retry limit reached. Locking account with username john_smith.
App ID: VOSS Automate

...
```

9.13. Event Log Format and Details

Event log streaming sends all transaction data via `syslog`. It should be noted that the data is simply raw transaction data with no hierarchical grouping of parent and associated child transactions. If required, the remote `syslog` server must recreate a transaction tree hierarchy as part of log processing.

The following describes the format of an event log entry. The event log file contains single lines of data in JSON format, with meta data and data elements.

- Meta data has `event_` - attributes, and describes the type of event log.

  For example, for `"event_type": "transaction.finalise"`: when a transaction is finalized in the system:

  ```
  {
    "event_level": "INFO",
    "event_type": "transaction.finalise",
    "event_source": "voss-un1",
    "event_id": "abc08383-5adb-48cb-8181-ef6adc546791",
    "event_timestamp": "2017-12-04T12:18:07.025595Z",
    "event_client_ip_address":"172.29.90.35",
    "event_message": "Transaction 1267 finalised.",
    "event_data": {
    [...]
  ```

  - `event_id`: Unique ID associated with the log entry.
  - `event_message`: Message specified during log entry creation.
  - `event_level`: Log level with which log entry was created.

  - `event_timestamp`: Datetime string of timestamp when the log entry was created.
  - `event_type`: Unique type associated with event described by log entry.
  - `event_source`: Hostname of host from which log was created.
  - `event_data`: Additional data associated with log entry, containing a `transaction` object.
  - `event_client_ip_address`: IP address from where the transaction originated. This value can then be searched for and matched in `access.log` in order to record/audit the source IP address from where access to the platform has been made when transactions are carried out.

- Data is recorded in the "`event_data`" element of the event type, with each event type determining its own `event_data` JSON structure.

  For example, for "`event_type`": "`transaction.finalise`", the `event_data` is "`transaction`", the start of the structure is for example:

```
[...]
 "event_data": {
   "transaction": {
     "status": "Success",
     "username": "sysadmin",
     "rolled_back": false,
     "resource": {
       "hierarchy": "1c0nfmo2c0deab90da595101",
       [...]
```

  - `transaction`: Transaction specific data.
    - `action`: Transaction's `action` field, which is displayed by the Admin Portal in its Action field.
    - `completed_time`: Datetime string of the transaction's `completed_time` field, which is displayed by the Admin Portal in its Completed Time field.
    - `detail`: Transaction's detail field, which is displayed by the Admin Portal in its Detail field.
    - `duration`: Transaction's `duration` field (in seconds), which is displayed by the Admin Portal in its Duration field.
    - `hierarchy`: Transaction's `execution_hierarchy` field.
    - `message`: Transaction's `exception_message` field (if any), which is displayed by the Admin Portal in its Message field.
    - `operation`: Transaction's `operation` field.
    - `parent_pkid`: Transaction's `parent` field (if present, can be used to identify a parent transaction `pkid` value and any sub-transactions).
    - `pkid`: Transaction's `_id` field (this value will be the `parent_pkid` of sub-transactions if there are any).
    - `priority`: Transaction's `config['priority']` field, which is displayed by the Admin Portal in its Priority field.
    - `processor_host_name`: Transaction's `processor_host_name` field, which is displayed by the Admin Portal in its Processor Host Name field.
    - `resource`: resource object associated with transaction
      - `hierarchy`: Transaction's resource `hierarchy` field.

        - `model_type`: Transaction's resource `model_type` field, which is displayed by the Admin Portal in its Model Type field.
        - `pkid`: Transaction's resource `pkid` field.
    * `rolled_back`: Transaction's `rollback` field, which is displayed by the Admin Portal in its Rolled Back field.
    * `started_time`: Datetime string of the transaction's `started_time` field, which is displayed by the Admin Portal in its Started Time field.
    * `status`: Transaction's `status` field, which is displayed by the Admin Portal in its Status field.
    * `submitted_time`: Datetime string of the transaction's `submitted_time` field, which is displayed by the Admin Portal in its Submitted Time field.
    * `submitter_host_name`: Transaction's `submitter_host_name` field, which is displayed by the Admin Portal in its Submitter Host Name field.
    * `txn_seq_id`: Transaction's `txn_seq_id` field, which is displayed by the Admin Portal in its Transaction ID field.
    * `username`: Transaction's `username` field, which is displayed by the Admin Portal in its Username field.

A full example of a `transaction.finalise` type event log entry is shown below (formatted multiline):


```
{
  "event_level": "INFO",
  "event_type": "transaction.finalise",
  "event_source": "voss-un1",
  "event_id": "abc08383-5adb-48cb-8181-ef6adc546791",
  "event_timestamp": "2017-12-04T12:18:07.025595Z",
  "event_client_ip_address":"172.29.90.35",
  "event_message": "Transaction 1267 finalised.",
  "event_data": {
    "transaction": {
      "status": "Success",
      "username": "sysadmin",
      "rolled_back": false,
      "resource": {
        "hierarchy": "1c0nfmo2c0deab90da595101",
        "model_type": "data\\/Countries",
        "pkid": "5a203da004222e1c67f93c83"
      },
      "processor_host_name": "voss-un1",
      "pkid": "233cd3b1-8acc-4702-bd64-90653c02cd81",
      "hierarchy": "1c0nfmo2c0deab90da595101",
      "started_time": "2017-12-04T12:18:04.946000Z",
      "detail": "Australia",
      "completed_time": "2017-12-04T12:18:07.022000Z",
      "priority": "Normal",
      "duration": 2.076404,
      "submitted_time": "2017-12-04T12:18:04.461000Z",
      "submitter_host_name": "voss-un2",
      "txn_seq_id": "1267",
      "parent_pkid": null,
```


(continues on next page)

(continued from previous page)

```
      "action": "Update Countries",
      "message": null,
      "operation": "update"
    }
  }
}
```

9.14. Log Streaming

VOSS Automate platform CLI commands are available to enable and manage the streaming of a configurable list of log files to a remote syslog server that can then for example generate alarms based on log messages. The VOSS Assurance product can for example be configured to be the remote server that monitors the system.

The following parameters are available:

- Remote IP
- Remote port
- Protocol
- Files to stream

The **log stream** command parameters can therefore be used to:

- set up and remove remote log streaming on a server
- add and remove files to stream to a server

Note: The **log stream** commands that make changes will restart the `syslog` service.

Refer to the command examples for parameter usage and CLI output.

Command Examples

- **log stream remotelog**

```
platform@VOSS:~$ log stream remotelog
ip: Unset
```

- **log stream remotelog <IP:port>|off**

```
platform@VOSS:~$ log stream remotelog 192.168.100.25:514
You are about to restart syslog. Do you wish to continue? y
System updated to use 192.168.100.25 as remote syslog server
You have new mail in /var/mail/platform
```

- **log stream status**

```
platform@VOSS:~$ log stream status
    stream:
        ip: 192.168.100.25
        port: 514
        protocol: udp
```

- **log stream remotelog status**

```
platform@VOSS:~$ log stream remotelog status
ip: 192.168.100.25:514
```

- **log stream enable <filename>[,<filename>...]**

```
platform@VOSS:~$ log stream enable syslog,auth.log
You are about to restart syslog. Do you wish to continue? y
syslog already in list of files to send to remote
Updating list of files to stream to remote
You have new mail in /var/mail/platform

platform@VOSS:~$ log stream status
    stream:
        filenames:
            syslog
            auth.log
        ip: 192.168.100.25
        port: 514
        protocol: udp

platform@VOSS:~$ log stream enable dmesg,dpkg.log
You are about to restart syslog. Do you wish to continue? y
Updating list of files to stream to remote
platform@VOSS:~$ log stream status
    stream:
        filenames:
            syslog
            auth.log
            dmesg
            dpkg.log
        ip: 192.168.100.25
        port: 514
        protocol: udp
```

If a file is already in the list, a message shows: “already in list of files to send to remote”.

Note: If the `platform/event.log` and `platform/audit.log` files are to be streamed, ensure that local logging has been enabled for these, using the `log event locallog on` and `log audit locallog on` commands. See: *Log Type Commands*.

- **log stream disable <filename>**

```
platform@VOSS:~$ log stream disable syslog
You are about to restart syslog. Do you wish to continue? y
syslog not currently in list of files to send to remote
You have mail in /var/mail/platform
```

- **log stream protocol <tcp|udp>**: `udp` is default if not specified.

See also:

- *Log Types*
- *Log Type Commands*

9.15. Remote Log Type Encryption

The VOSS Automate system can encrypt remote log types: `audit`, `event` or `stream`.

The steps and commands to follow for remote log type encryption are set out below:

1. Edit SSL details on the system. (The user is prompted for C,ST,O,OU,FQDN):

   log cert details edit

   Inspect the edited SSL details:

   log cert details

2. Generate a Certificate Signing Request (CSR) file and submit it to the certificate authority (CA).

   log cert gen_csr

   The CSR file can also be printed out:

   log cert print_csr

3. Receive the signed certificate. Then upload it to the system (using for example **scp**) and add your signed certificate with:

   log cert add <filename>

   For example:

   ```
   $ log cert add media/cert.pem
   ```

   Add the CA certificate to the system with:

   log cert addca <filename>

   For example:

   ```
   $ log cert addca media/ca-chain.cert.pem
   ```

   Inspecting the SSL details at this stage, using **log cert details**, shows the SSL details for:

   - Issuer data
   - Key data
   - User set details

4. Enable remote logging of the log type. This will restart the syslog server.

   log [audit|event|stream] remotelog <IP:port>

5. Enable SSL on log type logging. This will restart the syslog server.

   log ssl enable

SSL logging of log type can be disabled by the command **log ssl disable**. This will restart the syslog server.

To see SSL logging details and if it is enabled or not, run **log ssl status**.

For example, the output below shows `enabled: false`:

```
user@host:~$ log ssl status
    ssl:
        C: ZA
        CN: VOSS.visionoss.int
        L: Cape Town
        O: Voss-Solutions
        OU: Platform
        ST: WP
        email: user@host.com
        enabled: false
```

9.16. The Mail Command

The system monitors a number of events – these are described in more detail in the topic on Warnings and Notifications. The events can be signalled externally using email and snmp. However, a local copy of all events is maintained in the platform user's mailbox.

Command	Description
mail list	Display a list of events stored in the mailbox.
mail read all	Read all mail.
mail read <number>	Read a specific mail message.
mail del <number>	Delete a specific mail message.
mail del <from> <to>	Delete a range of mail messages.
mail del all	Delete all mail messages.

Mail events may accumulate over time. The system will purge old events automatically if the mailbox becomes too full (more than 500 messages).

9.17. Diagnostic Tools

There is an extensive list of diagnostic tools available under the **diag** menu.

For example output of **diag** commands, see: *diag Command Examples*.

```
platform@VOSS:~$ diag
USAGE:
------
diag config [<config-item>]                    - Display configuration, optionally␣
↪specifying the configuration prefix, diag
                                                 config /platform/network/name
diag config app <app-config> [<config-item>] - Display application configuration,␣
↪optionally specifying the configuration
                                                 prefix, eg diag config app snmp /community
diag date <epoch timestamp>                    - Convert epoch timestamp to datetime
diag disk                                      - Display diagnostics for disk usage
diag epoch <yyyy-mm-dd hh:mm>                  - Convert datetime to epoch timestamp
diag fileguard                                 - Run a file integrity check using AIDE␣
↪(Advanced Intrusion Detection Environment)
                                                 to detect unauthorized or unexpected␣
↪changes to critical system files.
diag free                                      - Display diagnostics relating to free␣
↪memory
diag health                                    - Display a health report
diag health report                             - Save a health report as a logfile
diag iostat                                    - IO subsystem statistics
diag largefiles                                - Find the largest files on your system no␣
↪more than the top 10 items are display
diag mem                                       - Display memory diagnostics
diag monitor                                   - Display the system resource analysis
diag monitor list                              - Display the long-term average and maximum␣
↪system resource usage
diag nicstat                                   - Network interface statistics
diag ntp                                       - Display ntp status
diag ping <host>                               - Ping a remote host to test network␣
↪reachability
diag proc                                      - Display a list of system processes
diag resolve <host>                            - Resolve a hostname to IP address
diag stats                                     - Display the container statistics
diag tasks                                     - Display constant task listing
diag test_connection <host> <port>             - Test if system can open a connection to a␣
↪remote port
diag top                                       - Process resource statistics
diag traceroute <host>                         - Discover the network path to <host>
diag vmstat                                    - Virtual memory subsystem statistics
```

Command Descriptions

Some of the commands are provided with details below:

Command	Description
diag ping <host>	Test network reachability to a network host.
diag resolve <hostname>	Test DNS resolution of a hostname.
diag test_connection <host> <port>	Given a host IP and port number, return a message: "Successfully connected to <host>:<port>" or "Failed to connect to <host>:<port>".
diag free	Display the memory usage.
diag disk	Display the disk usage. Logical volumes for the database have the format `/dev/mapper/voss-dbroot`.
diag largefiles	Display up to 10 largest files on the system over 1GB in size, else no files are shown.
diag mem	Display a more detailed memory usage by process.
diag ntp	Display details of ntp server, including peers and host addresses. Equivalent to the `ntp -pn` command.
diag health	Display a comprehensive health summary. This includes status on the following: CPU, Memory, Disk, Security Update, Application, Cluster, Cluster Failover and Health email scheduling. Logical volumes used by the database have the format: `/dev/mapper/voss-dbroot`.
diag top	Display a single Unix top summary.
diag unittests	Utility for developers only. Note that services will be restarted by this utility.
diag fileguard	Although a checksum of system and configuration files is carried out regularly, a manual check for changes since the previous check can be carried out. If any files have changed, these will be listed in the command output. File tamper detection and integrity monitoring is also carried out. If any files have changed, these will be listed in the command output.

For **diag health**, see also *Diagnostic Troubleshooting*.

9.18. Diagnostic Troubleshooting

The health displayed on login will normally include sufficient information to determine that the system is either working, or experiencing a fault. More detailed health reports can be be displayed with **diag health**.

Important: Since the **diag health** command output is paged on the console, you can scroll up or down to see all the output.

Type `q` at the `:` prompt to quit the console pager and output (*not Ctrl-C*).

```
platform@atlantic:~$ diag health

Health summary report for date:
     Mon Aug 16 12:56:51 UTC 2021
CPU Status:     12:56:51 up 12:13, 1 user, load average: 1.75, 1.72, 1.75
Platform version:
     platform v21.1.0 (2021-08-15 13:37) Network Status:    System name:  VOSS
    Device:  ens160  Ip:   Netmask:   Gateway: 192.168.100.1
Memory Status:
      total used free shared buff/cache available
     Mem: 8152816 5820696 392336 164500 1939784 1872452
     Swap: 2096124 112640 1983484
Disk Status:
     Filesystem Size Used Avail Use% Mounted on
     /dev/sda1 18G 4.7G 13G 28% /
     /dev/sdb1 9.9G 154M 9.2G 2% /var/log
     /dev/sdb2 40G 8.4G 30G 23% /opt/platform
     /dev/sdc1 49G 53M 47G 1% /backups
     /dev/mapper/voss-dbroot 225G 5.1G 220G 3% /opt/platform/apps/mongodb/dbroot
Security Update Status:
     There are 0 security updates available for the base system.
Application Status:
     selfservice v21.1.0 (2021-08-15 13:36)
      |-node running
     voss-deviceapi v21.1.0 (2021-08-15 13:36)
      |-voss-cnf_collector running
      |-voss-queue running
      |-voss-wsgi running
      |-voss-risapi_collector running
      |-voss-monitoring running
     cluster v21.1.0 (2021-08-15 13:36)
     template_runner v21.1.0 (2021-08-15 13:43)
     mongodb v21.1.0 (2021-08-15 13:36)
      |-arbiter running
      |-database running
     support v21.1.0 (2021-08-15 13:43)
     selenium v21.1.0 (2021-08-15 13:42)
...
```

A rich set of SNMP and SMTP traps are described in the Notifications section which can be used to automate fault discovery.

Determine if all processes are running using **app status**. If a process is not running, investigate its log file with:

log view process/<application>.<process>

For example, checking processes:

```
platform@development:~$ app status
development v0.8.0 (2013-08-12 12:41)
voss-deviceapi v0.6.0 (2013-11-19 07:37)
   |-voss-celerycam             running
   |-voss-queue_high_priority   running
```

(continues on next page)

(continued from previous page)

```
   ...
core_services v0.8.0 (2013-08-27 10:46)
   |-wsgi                       running
   |-logsizemon                 running
   |-firewall                   running
   |-mountall                   running
   |-syslog                     running (completed)
   |-timesync                   stopped (failed with error 1)
nginx v0.8.0 (2013-08-27 10:53)
   |-nginx                      running
security v0.8.0 (2013-08-27 11:02)
```

Followed by a log investigation for a stopped process:

```
platform@development:~$ log view process/core_services.timesync
2013-08-15 10:55:20.234932 is stopping from basic_stop
2013-08-15 10:55:20:    core_services:timesync killed
  successfully
2013-08-15 10:55:20: Apps.StatusGenerator core_services:timesync
  returned 1 after 1 loops
App core_services:timesync is not running with status stopped

...

+ /usr/sbin/ntpdate 172.29.1.15
2014-02-04 09:27:31: Apps.StatusGenerator core_services:timesync
  returned 0 after 1 loops
2014-02-04 09:27:31: WaitRunning core_services:timesync is reporting
  return code 0
core_services:timesync:/opt/platform/apps/core_services/timesync
  started
4 Feb 09:27:38 ntpdate[2766]: no server suitable for
  synchronization found
+ echo 'Failed to contact server: 172.29.1.15 - retrying'
Failed to contact server: 172.29.1.15 - retrying
+ COUNTER=2
+ sleep 1
+ test 2 -lt 3
+ /usr/sbin/ntpdate 172.29.1.15
4 Feb 09:27:48 ntpdate[3197]: no server suitable for
  synchronization found
+ echo 'Failed to contact server: 172.29.1.15 - retrying'
Failed to contact server: 172.29.1.15 - retrying
+ COUNTER=3
+ sleep 1
+ test 3 -lt 3
+ test 3 -eq 3
+ echo 'Timesync  - could not contact server 172.29.1.15 after
    three tries. Giving up'
Timesync  - could not contact server 172.29.1.15 after
```

(continues on next page)

(continued from previous page)

```
   three tries. Giving up
+ exit 1
```

The error message and return code being displayed in the browser is also invaluable in determining the cause of the problem.

The system resources can be inspected as follows:

- **diag disk** will display the disk status
- **diag free** and **diag mem** will display the memory status
- **diag top** will display the CPU status

10. Notifications

10.1. Warnings and Notifications

On console sign-in, a health report indicates the system status. This health report shows this data:

```
Last login: Tue Sep 3 10:19:07 2013 from 172.29.232.68
host: alan, role: standalone, load: 0.35, USERS: 3
date: 2013-09-03 10:20:02 +00:00, up: 2:05
network: 172.29.89.182, ntp: 172.29.1.15
SECURITY UPDATES: 136 security updates available
database: 8.0Gb
services: ok
```

The report values mean:

- last console sign-in and IP address source
- the load average of the system
- the number of users currently signed in
- the system uptime
- the status of the system services
- whether security updates are available
- disk, CPU, and memory warnings if applicable
- warnings are displayed in uppercase to draw attention

The report can be redisplayed by typing the command:

health

The system can be configured to forward warnings and notifications to various destinations, including:

- local email
- remote email addresses
- remote SNMP destinations

Local email allows the administrator to view a list of warnings, and delete them as necessary.

Note: If this configuration is carried out, it should be set on each node.

Typically, each node will be configured with its own sender email address to identify the source node (for example `unified_node_1@customer.domain`).

Additionally,

- nodes can be configured with different emailrelay servers, for example, according to data center
- nodes can be configured with different SNMP servers and error levels

The notification destinations can be displayed with **notify list**. The destinations for each event level can be set with **notify add info|warn|error <destination-URI>** Refer to the Network URI Specification topic for a detailed description of URIs.

Note that email notifications require the mail relay to be set with **notify emailrelay <smtp ip address>[:port]**. Add the SMTP IP address and optionally an allowed port (default port is 25).

The default `from` address is `platform@<hostname>`. A `from` outgoing address for the email relay can also be configured with **notify emailfrom <from: email address>** - see: *Enable Health Monitoring*. A test event can be generated with **notify test info|warn|error** to test the notification delivery mechanism.

Examples:

- **notify add info mailto:sysadmin@mycompany.com**
- **notify add error snmp://public@mysnmpserver.com**

```
$ notify add error snmp://public@mysnmpserver.com
   notifications:
     emailrelay: 172.1.1.1
     level:
         error:
             snmp://public@mysnmpserver.com
             mailto:platform@localhost
         info:
             mailto:platform@localhost
         warn:
             mailto:platform@localhost
```

In addition to external email and SNMP alerts, the system also records various events to a local mailbox. Refer to the Mail Command section for details.

SNMP CPU load notifications are set using:

snmp load <1min load> <5min load> <15min load>

This results in notifications being sent should the threshold be exceeded. For a server with 2 CPUs, it is recommended that this setting be:

snmp load 8 4 2

This means that notifications are sent if the 2-CPU system load averages over the last 1, 5, and 15 minutes reach these values.

10.2. Events and SNMP Messages

The following conditions are monitored for which SNMP traps can be sent. The trap levels and message strings are shown for the condition:

- Script install failures
  - ‘error’,’upgrade failed’,’upgrade failed as other activity is in progress
- Backup Success/Failure
  - ‘error’, ‘Backup failed’,
  - ‘info’, ‘Backup completed’
  - ‘error’, ‘ERROR: The last backup was more than 2 days ago’, ‘Backup list:
  - ‘info’, ‘INFO: Backups now runs regularly’, ‘Backup list:
- Restore Success/Failure
  - ‘info’, ‘Backup restored
  - ‘error’, ‘Backup restoration failed
- Nginx reconfiguration - If a webproxy is unable to contact one of the upstream systems
  - “error”, “nginx upstream failure”, “upstream %s server %s failed: %s”
  - “info”, “nginx upstreams OK”, “nginx upstream servers returned to normal”
- Disk full/cleared (if a monitored disk is above 80%, it will send a trap, and also when this is cleared)
  - level, ‘WARNING: DISK ALMOST FULL: Disk <disk name> is more than 80 percent full’
  - level, ‘INFO: DISK STATUS: Disk <disk name> is now running below 80 percent’
  - ‘error’, ‘WARNING: DISK ALMOST FULL: Disk /var/log is more than 80 percent full’,’Use log purge to purge all rotated logsnnCurrent disk status:
- Email failure (if a monitoring email was set up, and system cannot reach it)
  - ‘error’, ‘ERROR: Trouble sending health email’, ‘Trouble sending health email’
  - ‘info’, ‘INFO: Health emails is now being sent’, ‘Health emails is now being sent’
  - ‘info’, ‘INFO: Messages for <username> auto archived as it reached more than 500’ % user, ‘Use the following command to view archived messages:nnlog view <username>’ %
  - ‘info’, ‘INFO: The total local messages for <username> is now under 200’
  - ‘warn’, ‘WARNING: Not all notify levels is configured with an external email address ‘
  - ‘info’, ‘INFO: All notify levels is now configured with an external email address’
- Database usage
  - ‘warn’, reason=’WARN: Database <database name> exceeded threshold’
  - ‘info’, reason=’INFO: Database <database name> returned to normal’
- High disk latency
  - ‘error’, ‘ERROR: Disk slow ‘, ‘Disk latency info:
  - ‘info’, ‘INFO: The disk latency returned to normal’, ‘The disk latency returned to normal.’
- Database failover (if the DB fails over from one node to another twice in a 5 minute period)

  - ‘error’, ‘ERROR: The db is failing over constantly within 5 min’, ‘Cluster failover status:
  - ‘info’, ‘INFO: The db failover status returned to normal’, ‘Cluster failover status:
- Large log file warning
  - ‘error’, ‘ERROR: Log files larger than 1Gig found in /var/log ‘, ‘Logrotation was executed to rotate the following logs: <log filename>’
  - ‘info’, ‘INFO: /var/log rotated’, ‘/var/log rotated’
- Network
  - ‘error’, ‘ERROR: Network Failures’, ‘The following network failures occured: <network errors>’
  - ‘info’, ‘INFO: Network failures resolved’, ‘Network failures resolved’,
- Service failures
  - ‘error’, ‘ERROR: Service Failures’
  - ‘info’, ‘INFO: Services started successfully’
- Security updates available
  - ‘warn’, ‘WARNING: Security Updates available’, ‘<number> updates available’
  - ‘info’, ‘INFO: Security Updates applied’, ‘There are 0 security updates available
- High memory and CPU usage
  - ‘error’, ‘ERROR: High memory usage’, ‘Memory activity:
  - ‘info’, ‘INFO: Memory usage returned to normal’, ‘Memory more than 1024MB’
  - ‘warn’, ‘WARNING: High CPU usage’, ‘CPU activity:
  - ‘error’, ‘ERROR: Extremely high CPU usage’, ‘CPU activity:
  - ‘info’, ‘INFO: CPU usage returned to normal’
- High swap usage
- NTP configuration issues
  - ‘warn’, ‘WARNING: The ntp daemon has stopped on <server name>’, ‘Run 'app start services:time' to restart ntpd’,
  - ‘warn’, ‘WARNING: The ntp offset exceeds 1 second on %s’ % system_info, ‘ntp offset exceeds 1 secondnCurrent ntp offset value: <ntp offset>’
  - ‘info’, ‘INFO: ntp is now configured for <server name>’, ‘NTP cleared’, value=0
  - ‘info’, ‘INFO: The ntp offset restored to normal on <server name>’, ‘ntp offset clearednCurrent ntp offset value: <ntp offset>’
- DNS configuration issues
  - ‘warn’, ‘WARNING: No dns configured for <server name>’, ‘It is recommended that the dns is configured.nnTo configure dns use the following command:nnetwork dns <server1> <server2>’
  - ‘info’, ‘INFO: dns is now configured for <server name>’, ‘DNS cleared’, value=0
- Domain configuration issues
  - ‘info’, ‘INFO: domain is now configured for <server name>, ‘Domain cleared’, value=0

  - ‘warn’, ‘WARNING: No domain configured for <server name>, ‘It is recommended that the domain is configured.nnTo configure the domain use the following command:nnetwork domain <server1> <server2>’,

- Log processing failure
  - ‘error’, ‘System unable to send <event type> messages to <IP>’
- Monthly report export failure: (example at *Internal Report Schedules*)
  - ERROR: License file generation failed. The license audit report scheduled for <month> <year> was not successful. Please contact your VOSS account manager.

For details, refer to the topic on the individual SNMP trap.

10.3. SNMP Configuration and Queries

This topic covers configurations for various versions.

10.3.1. SNMP Configuration

SNMP must be configured under the SNMP menu and the SNMP URI needs to be configured for all the notify severity levels(info|warn|error]).

Note:

- Characters used in the SNMP URI can include uppercase and lowercase letters, decimal digits, hyphen, period, underscore, and tilde:

  ` ` ALPHA / DIGIT / “-” / “.” / “_” / “~” ` `
- The SNMP host should be an IP address, *not* a host name.

SNMP URI usage:

- snmpv2: **snmp://community@host[:port]**
- snmpv3: **snmp://user:auth:password]@host[:port]** . . . minimum auth/password length is 8 characters.

For example:

- snmpv2: **notify add info snmp://public@1.2.3.4**
- snmpv3: **notify add error snmp://public:publicauth:password@1.2.3.4**

The following options can be configured under the SNMP menu in the CLI.

- Enabled -Enable or disable SNMP Queries
- Community- SNMP v2c Community String used to query this server
- Authorized Username - SNMP v3 Username to query this server
- Password - SNMP v3 Password to query this server
- Query - IP address that is allowed to query this server
- Sysname - Name of this server, as it will appear when queried via SNMP

- Syslocation - Location of this server
- Syscontact - Contact person(s) for this server (email address)
- Load1 - 1 Minute load average alarm value
- Load5 - 5 Minute load average alarm value
- Load15 - 15 Minute load average alarm value

The following options can be configured in the CLI:

- Hostname - Server name to send SNMP traps to.
- Version - Version of SNMP to use for sending trap, version 2c or 3.
- Community - refer to the SNMP-URI command usage.

10.3.2. SNMP Queries

The VOSS Automate server permits multiple remote query sources to perform SNMP queries against.

The following commands are available to set SNMP details:

- **snmp contact <system contact>**
- **snmp name <system name>**
- **snmp location <system location>**

SNMP query sources can be added with

snmp query add <uri>

SNMP v2 can be set with:

snmp query add snmp://<community string>@<ip>

SNMP v3 username and password can be set with:

snmp query add snmp://user:auth:password@<ip>:<port>

Where:

- `user`: the username for the SNMPv3 server
- `auth`: the SNMPv3 authKey, with a minimum length of 8 characters (SHA authentication)
- `password`: the SNMPv3 privKey, with a minimum length of 8 characters (AES encryption)
- To see the list of added query sources, run **snmp query list**.
- To remove a query source, run **snmp query del <uri>**.

The screen console output below are example of the use of **add**, **list**, and **del** parameters with SNMPv2:

```
platform@host:~$ snmp query add snmp://private@192.29.21.3
You are about to restart the SNMP service. Do you wish to continue? y
Please update notify to reflect your latest changes.
You have new mail in /var/mail/platform
platform@host:~$

platform@host:~$ snmp list
    load1: 4
```

(continues on next page)

(continued from previous page)

```
    load15: 1
    load5: 2
    query:
        snmp://public@192.29.21.2
        snmp://private@192.29.21.3
    syscontact: Robert Frame
    syslocation: Dublin
    sysname: host

platform@host:~$ snmp query del snmp://private@192.29.21.3
You are about to restart the SNMP service. Do you wish to continue? y

Application services:firewall processes stopped.
Application snmp processes stopped.
Application snmp processes started.
Please update notify to reflect your latest changes.
You have new mail in /var/mail/platform
platform@host:~$
```

SNMP CPU load notifications are set using:

snmp load <1min load> <5min load> <15min load>

This results in notifications being sent should the threshold be exceeded. For a server with 2 CPUs, it is recommended that this setting be:

snmp load 8 4 2

This means that notifications are sent if the 2-CPU system load averages over the last 1, 5, and 15 minutes reach these values.

11. SNMP

11.1. Introduction to SNMP and MIB

Simple Network Management Protocol(SNMP) is a UDP-based network protocol used mostly in network management systems to monitor network-attached devices. SNMP is a component of the Internet Protocol Suite as defined by the Internet Engineering Task Force(IETF) and consists of a set of standards for network management, including an application layer protocol, a database schema and a set of data objects.

SNMP exposes management data in the form of variables on the managed systems that describe the system configuration. These variables can be queried using SNMP management applications.

SNMP allows a Network Management Station to do the following:

- Poll a device for info or to trend data i.e. VOSS Automate server load graph via HOST-SYSTEMS-MIB
- Receive notifications in the form of traps or informs in response to events, threshold violations, whatever the trap definitions in the loaded MIBs are. We enable process monitoring and disk space checks - when triggered, these send out a trap.

A management information base (MIB) is a form of virtual database used for managing the entities in a communications network. Working closely with SNMP, the hierarchical data structure describes all of the objects that a device can report the status of.

The MIB is structured based on the RFC 1155 standard. This standard defines how the MIB information is organized, what data types are allowed and how resources within the MIB are named. Each MIB contains the name, object identifier (a numeral), data type and the permissions relating to whether the value can be read or written to. The top hierarchies of the MIB are fixed; however, certain sub trees can be defined by product vendors and other organizations.

The variables within MIB are named using the Abstract Syntax Notation 1 (ASN.1). This is an international standard for representing data.

SNMP Terminology:

- MIB: The term MIB is used to refer to the complete collection of management information available on an entity, while MIB subsets are referred to as MIB-modules.
- NMS: ANetwork Management System is a combination of hardware and software used to monitor and administer a network and the devices associated with that network.

SNMP on VOSS Automate Platform is configured after initial system setup. The following SNMP parameters can be configured. Refer to *SNMP Configuration and Queries* and the index for commands.

- The SNMP system name

  Identifies the system being monitored on the NMS (Network Management System). Defaults to nodename.domainname.

- The SNMP system location

  Describes the location of the system. Defaults to Unknown.

- The SNMP system contact

  Defines the email address of administrator responsible for the system. Defaults to None.

- The SNMP query source

  URI from which the system accepts SNMP queries. Formatted as `snmp://community@host[:port]` for version 2 and `snmp://user:auth:password@host[:port]` for version 3

  CIDR-style IP (e.g. 196.0.0.0/8) network allowed to query SNMP from this host. This is used to limit the hosts allowed to manage the system via SNMP. Defaults to all hosts.

- The SNMP load triggers

  The 1, 5 and 15 minute load averages that will trigger warnings via SNMP. Defaults to values dynamically calculated from the number of CPUs in the system. This should be formatted as 8n/4n/2n (where n represents the number of processors available) when entered into the configuration wizard during setup.

- The SNMP trap destination

  Formatting identical to query source.

11.2. SNMP Traps

When the managed system generates certain events, it will forward a SNMP trap. The reason for the event trap is contained in the SNMP MIB string.

Important: It is strongly recommended that you monitor SNMP traps that can have a **error** severity level. Refer to the examples in this document.

For a list of notification messages, see: *Notification Messages*.

Note that if the corresponding SNMP MIB is not loaded on the NMS, a numerical representation of the SNMP entry is provided.

The list of monitored events is described in the SNMP Trap section below. A detailed breakdown of each SNMP trap type is provided in the appendix.

The SNMP will send traps to the trap destination configured. If the trap destination is incorrect or not configured, the NMS will not receive the traps.

SNMP configuration settings can be managed from the CLI. Refer to the CLI **notify** command:

```
platform@development:~$ notify
USAGE:
------
notify add [info|warn|error]             - Add the email or snmp URI to a
   <email/snmp-uri> ...                      specified notification level.
```

The following system parameters are monitored by default

- Disk Space: warnings are issued if the file system breaches the following thresholds:

```
disk  / 30% free
disk  /opt/platform 30% free
disk  /var/log 10% free
```

- System Load Monitoring: warnings are issued if the system load is excessive (the system load parameters can be defined during configuration)
- SNMP: standard SNMP System Events, for example, Cold Start
- Process state changes: Informative messages are sent to the NMS indicating that processes have been restarted.

In general, the originator of the SNMP traps is determined by originating hostname / IP address. Many Network Management Systems provide trap management and escalation per system being managed, including identification based on system name, location and contact details.

Those events monitored directly by VOSS Automate (e.g. disk space, system load and process warnings) include the system name as part of the variable bindings to assist identification of the originating system.

The state of the VOSS Automate system can be monitored either on the NMS or via the command line interface using the **diag** command.

11.3. Management Information Bases

Important:

- The VOSS Automate system uses standard MIBs that are usually deployed as part of a Network Management System (NMS).

  No VOSS Automate specific MIBs are available.

  The standard MIBs can for example be inspected from on-line resources, such as http://www.oidview.com/mibs/detail.html.
- String values in the trap descriptions and examples shown here is illustrative purposes only.
- Multi-line display in the trap examples shown here is done for formatting purposes only.

SNMP information is grouped together in Management Information Bases (MIBs). The MIBs loaded on the VOSS Automate system represent all the configuration/data items that can be queried or be used to generate traps (notifications) when certain events occur. A list of all MIBs loaded on the system is provided below.

In order to manage the system, a Network Management System (NMS) should be installed at the customer site (e.g. HP OpenView, iReasoningMib Browser). The NMS should be loaded with the same set of MIBs as those installed on the system. The NMS should be configured to send SNMP queries to the managed host (i.e. correct IP address, port number (default 161), community string (default public), and version (default version 2c).

Further, the NMS should be configured to receive traps from the managed host - the correct IP, port number (default 162), version (default version 2), and community strings (default public) should be provided).

SNMP items can be selected in the MIBs and the item queried on the remote managed system. The remote system will return a response to the MIB entry being queried.

For example, if the following entry is queried

```
.1.3.6.1.2.1.1.5.0 alias '.iso.org.dod.internet.mgmt.mib-2.system.sysName.0'
```

the system will return the system name that was assigned during setup (e.g. sysName.0 'Voss Node00').

Note that if any of the configured details on the NMS are incorrect, it is likely that the query will never reach the managed host and no response will be received.

Please ensure that version 2 is selected with the correct community string (default public).

When the managed system generates certain events, it will forward a SNMP trap. The reason for the event trap is contained in the SNMP MIB string.

Note that if the corresponding SNMP MIB is not loaded on the NMS, a numerical representation of the SNMP entry is provided.

The list of monitored events is described in the SNMP Trap section below.

Refer to the MIB List at the end of this document for the list of net-SNMP packages that ship with VOSS Automate.

11.3.1. SNMPv2-MIB - RFC 3418 - Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)

Basic information about SNMP on the entity. Includes:

- sysDescr: A text description of the entity
- sysObjectID: The vendor's authoritative identification of the network management subsystem contained in the entity.

Note: sysUpTime indicates how long the SNMP software has been running on the box, and not how long the box itself has been up (this is a common misconception).

- sysUpTime: The time since the network management portion of the system was last re-initialised.
- Counters for SNMP requests and responses

11.3.2. IF-MIB - RFC 2863 - The Interfaces Group MIB

Describes the network interfaces on the entity. For each interface the following information is given:

- ifType: The type of interface
- ifMtu: Size of the largest packet which can be sent/received on the interface
- ifSpeed: An estimate of the interface's current bandwidth
- ifPhysAddress: The interface's address at its protocol sub-layer. For 802.x interfaces, this is the MAC address
- The administrative and operational state of the interface
- The number of octets and packets sent and received on the interface

11.3.3. MIB-II - RFC 1213 - Management Information Base for Network Management of TCP/IP- based internets

TCP/IP network information not covered by the other MIBs, split into a number of groups:

- Address translation group:
- atPhysAddress: The media-dependent physical address
- atNetAddress: The network address (IP address) corresponding to the physical address
- IP group:
- ipRouteTable: IP routing table, contains an entry for each route presently known to this entity

11.3.4. IP-MIB - RFC 4293 - Management Information Base for the Internet Protocol (IP)

Internet Protocol information:

- Counters for IP packets sent and received
- For each IP address:
- The IP address
- Index of the physical interface (in the IF-MIB)
- Netmask
- ICMP counters

11.3.5. TCP-MIB - RFC 4022 - Management Information Base for the Transmission Control Protocol (TCP)

TCP information:

- Retransmission timeout information
- Overall counters for number of inbound and outbound connections
- For each current connection:
- Connection state
- Local and remote IP addresses and TCP port numbers

11.3.6. UDP-MIB - RFC 4113 - Management Information Base for the User Datagram Protocol (UDP)

UDP information:

- Counters for datagrams sent and received
- Local IP addresses and UDP port numbers

11.3.7. HOST-RESOURCES-MIB - RFC 2790 - Management Information Base for Host Resources

Objects useful for the management of host computers. These are split into a number of groups:

- System Group
  - hrSystemUptime: Amount of time since the host was last initialised (note this is different from sysUpTime).
  - hrSystemDate: The host's notion of the local date and time of day
  - hrSystemProcesses: The number of process contexts currently loaded or running on this system
- Storage Group
  - hrMemorySize: The amount of physical read-write main memory, typically RAM, contained by the host
- For each storage device:
  - hrStorageType: The type of storage (RAM, fixed disk etc.)
  - hrStorageDescr: A description of the storage (Swap Space, mount point etc.)
- Size of storage units, number available and number used
- Device Group
  - For each device:
    - Type (processor, network, disk, printer etc.)
    - Description
- For each disk storage device:
  - Access (read-write, read-only)
  - Fixed/removable
  - Capacity
- For each disk partition:
  - Label
- For each file system:
  - Mount point
  - Type
  - Access (read-write, read-only)
  - Bootable
- Running Software Group
  - For each running process:
    - Name
    - Path
    - Parameters
    - Status

* Running Software Performance Group for each running process:
* CPU resources consumed by this process
* Amount of real system memory allocated to this process

11.4. Trap Details

11.4.1. Notification Messages

There are 3 levels of notifications:

- info - these are informational messages intended to communicate a positive event.
- warn - these are non-critical notifications indicating that there is a temporary error
- error - these are critical messages indicating a system fault that is affecting system functionality

The sections in this chapter provide examples of messages and their output format.

Note: The output strings in the messages of these examples may differ from current messages. Also refer to the CSV below.

For a full list of all messages, refer to the CSV below.

To download all notification messages as a CSV file, refer to the HTML documentation.

Backups

Category	Level	Message
Backup monitoring	error	ERROR: The last backup was more than 2 days ago . . .
Backup monitoring	info	INFO: Backups now runs regularly . . .
Backups	info	INFO: Backup completed. . .
Backups	info	INFO: Restore successful. . .
Backups	error	ERROR: Backup failed .

Examples

```
Oct 10 21:39:17 robot-slave snmptrapd[480]: 2022-10-10 21:39:17 <UNKNOWN> [UDP: [192.168.
↪100.3]:59765->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (306489) 0:51:04.89
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "Backup completed"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Backup was successfully created at localbackup"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:39:27 robot-slave snmptrapd[480]: 2022-10-10 21:39:27 <UNKNOWN> [UDP: [192.168.
↪100.3]:25962->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (307540) 0:51:15.40
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: Backups now runs regularly"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Backup list:
#012
#012    localbackup:
#012        backups:
#012            1 backups have been created - most recently 2022-10-10 19:38
#012        uri: file:///backups
#012
#012
#012
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:39:30 robot-slave snmptrapd[480]: 2022-10-10 21:39:30 <UNKNOWN> [UDP: [192.168.
↪100.3]:14347->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (307808) 0:51:18.08
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: The last backup was more than 2 days ago"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Backup list:
#012
#012    localbackup:
#012        backups:
#012            1 backups have been created - most recently 2022-10-10 19:38
#012        uri: file:///backups
#012
#012
#012
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

Resolution:

Investigate if the scheduler is set correctly and that the backup is set up correctly. Run a manual backup to test if it is working as it should. Otherwise, call Support to investigate.

```
Oct 10 21:40:06 robot-slave snmptrapd[480]: 2022-10-10 21:40:06 <UNKNOWN> [UDP: [192.168.
↪100.3]:53579->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (311359) 0:51:53.59
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "backup.py : num_sockets exeeded warning␣
↪threshold of 6 with 7"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Command backup.py attribute num_sockets
#012Current average value 7
#012Error threshold 10
#012Warning threshold 6
#012"
```

(continues on next page)

(continued from previous page)

```
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:42:01 robot-slave snmptrapd[480]: 2022-10-10 21:42:01 <UNKNOWN> [UDP: [192.168.
↪100.3]:7560->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (322901) 0:53:49.01
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "Backup completed"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Backup was successfully created at sftpbackup"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:17:09 robot-slave snmptrapd[480]: 2022-10-10 22:17:09 <UNKNOWN> [UDP: [192.168.
↪100.3]:36106->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (533696) 1:28:56.96
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Backup failed"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "ERROR: Backup failed - error running pre-backup␣
↪method for mongodb"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:22:14 robot-slave snmptrapd[480]: 2022-10-10 22:22:14 <UNKNOWN> [UDP: [192.168.
↪100.3]:21378->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (564173) 1:34:01.73
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Backup failed"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Backup to localbackup failed to verify"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

Resolution:

There may have been some form of corruption during the time that the backup was run. Run backup manually and see whether the same failure is detected. If it is the case, investigate. Otherwise, contact L2 Support to investigate.

Database in Cluster Failover

Category	Level	Message
Cluster Failover	error	ERROR: The db is failing over constantly within 5 min . . .
Cluster Failover	info	INFO: The db failover status returned to normal . . .

Database Maintenance and Usage

Category	Level	Message
Database Maintenance	error	ERROR: Database maintenance not scheduled. . .
Database Maintenance	info	INFO: Database maintenance is scheduled . . .
Database Usage	warn	WARNING: Database %s exceeded threshold . . .
Database Usage	info	INFO: Database %s returned to normal . . .

Resolution if no schedule

Create schedule and time:

With for example a schedule name `dbtxn`:

schedule add dbtxn voss transaction archive or **schedule add dbtxn voss transaction delete**

schedule time dbtxn weekly 0

See: *Enable Database Scheduling*.

Examples

```
Mar 18 08:00:50 platform-agent snmptrapd[515]: 2024-03-18 08:00:50 <UNKNOWN> [UDP: [192.
↪168.100.3]:8397->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (289519) 0:48:15.19
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "WARNING: Database Maintenance"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "ID: TRANSACTION_DATABASE_MAINTENANCE-VOSS, Code:␣
↪-1, Occurences: 5, Latest Occurence: 2024-03-18T06:00:49.368Z"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.1.5.0 = STRING: "VOSS"

Mar 18 08:00:51 platform-agent snmptrapd[515]: 2024-03-18 08:00:51 <UNKNOWN> [UDP: [192.
↪168.100.3]:29600->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (289617) 0:48:16.17
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Database maintenance not scheduled"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "TRANSACTION DATABASE MAINTENANCE NOT SCHEDULED -␣
↪SETUP SCHEDULE FOR REGULAR MAINTENANCE"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.1.5.0 = STRING: "VOSS"


Oct 10 22:43:20 robot-slave snmptrapd[480]: 2022-10-10 22:43:19 <UNKNOWN> [UDP: [192.168.
↪100.3]:41903->[192.168.100.25]:162]:
```

(continues on next page)

(continued from previous page)

```
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (690787) 1:55:07.87
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "WARN: Database transaction count exceeded␣
↪threshold"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "count: 13332"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"

Oct 10 22:43:24 robot-slave snmptrapd[480]: 2022-10-10 22:43:24 <UNKNOWN> [UDP: [192.168.
↪100.3]:25484->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (691198) 1:55:11.98
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: Database transaction count returned to␣
↪normal"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "The transaction count returned to normal."
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"

Oct 10 22:43:28 robot-slave snmptrapd[480]: 2022-10-10 22:43:28 <UNKNOWN> [UDP: [192.168.
↪100.3]:11973->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (691609) 1:55:16.09
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "WARN: Database transaction size exceeded␣
↪threshold"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "size: 32264767"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

Resolution for transaction size

The following commands can be used to resolve this:

`voss transaction delete <days> [limit <number>]`

`voss transaction archive <days>`

Disk Latency and Status

Category	Level	Message
Disk Latency	error	ERROR: Disk slow ‘, ‘Disk latency info:. . .
Disk Latency	info	INFO: The disk latency returned to normal
Disk Status	error	WARNING: DISK ALMOST FULL: Disk {} is more than {} percent full
Disk Status	info	INFO: DISK STATUS: Disk {} is now running below {} percent. Disk %s cleared Disk status after it was cleared:
Disk Status	warn	WARNING: DISK ALMOST FULL: Disk /var/log is more than {} percent full

Examples

```
Oct 11 00:02:46 robot-slave snmptrapd[480]: 2022-10-11 00:02:46 <UNKNOWN> [UDP: [192.168.
↪100.3]:63595->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1167361) 3:14:33.61
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "DISK ALMOST FULL: Disk /var/log is more than 80␣
↪percent full"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Use log purge to purge all rotated logs
#012
#012Current disk status:
#012Filesystem: /dev/sdb1
#012      Size: 9.9G
#012      Used: 9.5G
#012     Avail: 0
#012      Use%: 100%
#012Mounted on: /var/log"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:03:46 robot-slave snmptrapd[480]: 2022-10-11 00:03:46 <UNKNOWN> [UDP: [192.168.
↪100.3]:4569->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1173418) 3:15:34.18
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "DISK STATUS:  Disk /var/log is now running below␣
↪80 percent"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Disk /var/log cleared
#012Disk status after it was cleared:
#012Filesystem: /dev/sdb1
#012      Size: 9.9G
#012      Used: 31M
#012     Avail: 9.4G
#012      Use%: 1%
#012Mounted on: /var/log"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:05:19 robot-slave snmptrapd[480]: 2022-10-11 00:05:19 <UNKNOWN> [UDP: [192.168.
↪100.3]:36811->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1182726) 3:17:07.26
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Disk slow "
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Disk latency info:
#012
#012Disk latency:34.44ms
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"

Oct 26 10:59:57 robot-slave snmptrapd[546]: 2022-10-26 10:59:57 <UNKNOWN> [UDP: [192.168.
↪100.3]:33587->[192.168.100.25]:162]:
```

(continues on next page)

(continued from previous page)

```
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (135921) 0:22:39.21
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "DISK ALMOST FULL: Disk /opt/platform/apps/
↪mongodb/dbroot is more than 80 percent full"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Contact support to free space
#012
#012Current disk status:
#012      Use%: 92%
#012      Used: 207G
#012     Avail: 19G
#012 use_limit: 92
#012Filesystem: /dev/mapper/voss-dbroot
#012Mounted on: /opt/platform/apps/mongodb/dbroot
#012    Message is truncated"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"

Oct 26 10:59:58 robot-slave snmptrapd[546]: 2022-10-26 10:59:58 <UNKNOWN> [UDP: [192.168.
↪100.3]:48675->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (136014) 0:22:40.14
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "DISK STATUS:  Disk /opt/platform/apps/mongodb/
↪dbroot is now running below 80 percent"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Disk /opt/platform/apps/mongodb/dbroot cleared
#012Disk status after it was cleared:
#012      Use%: 1%
#012      Used: 1.3G
#012     Avail: 224G
#012 use_limit: 1
#012Filesystem: /dev/mapper/voss-dbroot
#012Mounted on: /opt/pla  Message is truncated"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

Resolution:

This trap indicates that there is heavy disk activity. This will be normal in many situations and should not always require immediate action. The one reason for this is that there are many transactions spawned at the same time. The disk utilization can be relieved by cancelling a few current transactions or by rescheduling some others.

However, this should be monitored closely and should it persist over several hours then Support should be contacted to investigate.

```
Oct 11 00:01:27 robot-slave snmptrapd[480]: 2022-10-11 00:01:27 <UNKNOWN> [UDP: [192.168.
↪100.3]:22430->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1159464) 3:13:14.64
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Disk full"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.9.1.100.1
```

(continues on next page)

(continued from previous page)

```
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.9.1.2.1 = STRING: "/"
#011iso.3.6.1.4.1.2021.9.1.101.1 = STRING: "/: less than 30% free (= 0%)"
```

```
Oct 11 00:02:28 robot-slave snmptrapd[480]: 2022-10-11 00:02:28 <UNKNOWN> [UDP: [192.168.
↪100.3]:25418->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1165577) 3:14:15.77
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Disk full"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.9.1.100.3
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.9.1.2.3 = STRING: "/var/log"
#011iso.3.6.1.4.1.2021.9.1.101.3 = STRING: "/var/log: less than 10% free (= 0%)"
```

```
Oct 11 00:02:46 robot-slave snmptrapd[480]: 2022-10-11 00:02:46 <UNKNOWN> [UDP: [192.168.
↪100.3]:63595->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1167361) 3:14:33.61
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "DISK ALMOST FULL: Disk /var/log is more than 80␣
↪percent full"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Use log purge to purge all rotated logs
#012
#012Current disk status:
#012Filesystem: /dev/sdb1
#012      Size: 9.9G
#012      Used: 9.5G
#012     Avail: 0
#012      Use%: 100%
#012Mounted on: /var/log"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

Disk Full Resolution

It depends on which disk is full:

- If it is the media or backup disks, then clean up the disk space.
- If it is any other disk then contact Support immediately.

DNS and Domain

Category	Level	Message
DNS	warn	WARNING: No dns configured for %s. . .
DNS	info	INFO: dns is now configured for %s . . .
Domain	warn	WARNING: No domain configured for %s. . .
Domain	info	INFO: domain is now configured for %s. . .

Examples

```
Oct 10 21:31:32 robot-slave snmptrapd[480]: 2022-10-10 21:31:32 <UNKNOWN> [UDP: [192.168.
↪100.3]:21656->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (260016) 0:43:20.16
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "WARNING: No dns configured for VOSS-192.168.100.3
↪"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "It is recommended that the dns is configured.
#012
#012To configure dns use the following command:
#012network dns <server1> <server2>"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

Resolution:

Configure a DNS for the system with `network dns add <server>`.

```
Oct 10 21:32:51 robot-slave snmptrapd[480]: 2022-10-10 21:32:51 <UNKNOWN> [UDP: [192.168.
↪100.3]:48247->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (267891) 0:44:38.91
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: dns is now configured for VOSS-192.168.100.
↪3"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "DNS cleared"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:33:56 robot-slave snmptrapd[480]: 2022-10-10 21:33:56 <UNKNOWN> [UDP: [192.168.
↪100.3]:36367->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (274443) 0:45:44.43
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: domain is now configured for VOSS-192.168.
↪100.3"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Domain cleared"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:34:00 robot-slave snmptrapd[480]: 2022-10-10 21:34:00 <UNKNOWN> [UDP: [192.168.
↪100.3]:50982->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (274843) 0:45:48.43
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "WARNING: No domain configured for VOSS-192.168.
↪100.3"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "It is recommended that the domain is configured.
#012
#012To configure the domain use the following command:
#012network domain <domain-name>"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

Resolution:

Configure domain with `network domain <name>`.

Mail

Category	Level	Message
External Mail Monitoring	warn	WARNING: Not all notify levels is configured with an external email address . . .
External Mail Monitoring	info	INFO: All notify levels is now configured with an external email address ..
Health email	error	ERROR: Trouble sending health email
Health email	info	INFO: Health emails is now being sent
Local Mail Monitoring	warn	WARNING: The total messages in the local mailbox for %s has reached in excess of 200..
Local Mail Monitoring	info	INFO: Messages for %s auto archived as it reached more than 500. . .
Local Mail Monitoring	info	INFO: The total local messages for %s is now under 200. . .

Examples

```
Oct 10 21:05:39 robot-slave snmptrapd[480]: 2022-10-10 21:05:39 <UNKNOWN> [UDP: [192.168.
↪100.3]:8550->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (104723) 0:17:27.23
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: All notify levels is now configured with␣
↪an external email address"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "    notifications:
#012        emailrelay: 127.0.0.1
#012        level:
#012            error:
#012                mailto:platform@localhost
```

(continues on next page)

(continued from previous page)

```
#012                snmp://public@192.168.100.25
#012             info:
#012          Message is truncated"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:27:29 robot-slave snmptrapd[480]: 2022-10-10 21:27:29 <UNKNOWN> [UDP: [192.168.
↪100.3]:5434->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (235697) 0:39:16.97
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "WARNING: The total messages in the local mailbox␣
↪for platform has reached in excess of 200"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Use the following mail commands to manage the␣
↪local mailbox:
#012
#012mail del <number>      - delete the selected mail
#012mail del <from
#> <to
#> - deletes the selected range of mail message
#012mail del all          Message is truncated"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:27:46 robot-slave snmptrapd[480]: 2022-10-10 21:27:46 <UNKNOWN> [UDP: [192.168.
↪100.3]:32395->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (237369) 0:39:33.69
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: Messages for platform auto archived as it␣
↪reached more than 500"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Use the following command to view archived␣
↪messages:
#012
#012log view /var/log/platform/archived_mail.log"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Notification message from (1, 3, 6, 1, 6, 1, 1):('192.29.22.122', 25067):
Var-binds:
1.3.6.1.2.1.1.3.0 = 7420086
1.3.6.1.6.3.1.1.4.1.0 = 1.3.6.1.2.1.88.2.0.1
1.3.6.1.2.1.88.2.1.1.0 = WARNING:  Some notify levels are configured with a local email␣
↪address
1.3.6.1.2.1.88.2.1.3.0 = Notify list:
    notifications:
    emailrelay: 172.29.42.30
    level:
    audit:
    snmp://rrako@192.29.21.225
    error:
    mailto:xlatform@loc  Message is truncated
```

(continues on next page)

(continued from previous page)

```
1.3.6.1.2.1.88.2.1.5.0 = 1
1.3.6.1.2.1.1.5.0 = UN1-192.29.22.122
```

Resolution:

Ensure that all notifications are non-local.

License

Category	Level	Message
License	error	ERROR: License file generation failed The license audit report scheduled for %s was not successful . . .
License	info	INFO: License file generated The license audit report scheduled for %s was successful. . .

Examples

```
Mar 18 07:57:38 platform-agent snmptrapd[515]: 2024-03-18 07:57:38 <UNKNOWN> [UDP: [192.
↪168.100.3]:46971->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (270390) 0:45:03.90
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "INFO: Application was killed license-
↪service:license-service"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"


Oct 10 22:46:06 robot-slave snmptrapd[480]: 2022-10-10 22:46:06 <UNKNOWN> [UDP: [192.168.
↪100.3]:33539->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (707420) 1:57:54.20
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: License file generation failed"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "The license audit report scheduled for October␣
↪2022 was not successful.
#012Please contact your VOSS account manager. "
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"

Oct 10 23:00:41 robot-slave snmptrapd[480]: 2022-10-10 23:00:41 <UNKNOWN> [UDP: [192.168.
↪100.3]:52074->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (794896) 2:12:28.96
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: License file generated"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "The license audit report scheduled for October␣
↪2022 was successful."
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
```

(continues on next page)

(continued from previous page)

```
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"

Nov  9 10:31:46 robot-slave snmptrapd[391]: 2022-11-09 10:31:46 <UNKNOWN> [UDP: [192.168.
↪100.3]:56185->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (63460) 0:10:34.60
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "Automate License Service Error"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "ID: AUTOMATE_LICENSE, Code: 12003, Occurences: 1,
↪ Latest Occurence: 2022-11-09T08:31:43.994Z"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"

Nov  9 10:32:09 robot-slave snmptrapd[391]: 2022-11-09 10:32:09 <UNKNOWN> [UDP: [192.168.
↪100.3]:4445->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (65783) 0:10:57.83
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application start command completed license-
↪service:license-service"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"

Nov  9 10:32:51 robot-slave snmptrapd[391]: 2022-11-09 10:32:51 <UNKNOWN> [UDP: [192.168.
↪100.3]:24239->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (69973) 0:11:39.73
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1#011iso.3.6.1.2.1.88.2.1.1.0 =␣
↪STRING: "Automate License Expiry"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "ID: AUTOMATE_LICENSE, Code: 12002, Occurences: 1,
↪ Latest Occurence: 2022-11-09T08:32:48.826Z"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

Resolution

See: *Product Licensing*.

Logs

Category	Level	Message
Log Processing	error	ERROR: System unable to send {} messages to {}. . .
Logging	error	ERROR: Log files larger than 1Gig found in /var/log . . .
Logging	info	INFO: /var/log rotated . . .
Logs Rotate	warn	WARNING: Forcing log rotation as /var/log is at $USAGE usage: $message
Logs Rotate	warn	WARNING: Autopurging /var/log due to extremely high disk usage. Usage $USAGE: $message

Examples

```
Oct 11 00:01:42 robot-slave snmptrapd[480]: 2022-10-11 00:01:42 <UNKNOWN> [UDP: [192.168.
↪100.3]:47625->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1160957) 3:13:29.57
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Log files larger than 1Gig found in /var/
↪log "
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Logrotation was executed to rotate the following␣
↪logs:
#012/var/log/big-log.log: 1.5G"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:01:43 robot-slave snmptrapd[480]: 2022-10-11 00:01:43 <UNKNOWN> [UDP: [192.168.
↪100.3]:6397->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1161096) 3:13:30.96
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: /var/log rotated"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "/var/log rotated"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:02:45 robot-slave snmptrapd[480]: 2022-10-11 00:02:45 <UNKNOWN> [UDP: [192.168.
↪100.3]:45283->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1167292) 3:14:32.92
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "Logs"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "WARNING: Forcing log rotation as /var/log is at␣
↪100 usage. [platform_mon.py] reading config file /etc/logrotate.conf
#012including /etc/logrotate.d
#012reading config file alternatives
#012reading config file apt
#012reading   Message is truncated"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:02:45 robot-slave snmptrapd[480]: 2022-10-11 00:02:45 <UNKNOWN> [UDP: [192.168.
↪100.3]:45283->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1167292) 3:14:32.92
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "Logs"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "WARNING: Forcing log rotation as /var/log is at␣
↪100 usage. [platform_mon.py] reading config file /etc/logrotate.conf
#012including /etc/logrotate.d
#012reading config file alternatives
#012reading config file apt
#012reading   Message is truncated"
```

(continues on next page)

(continued from previous page)

```
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:02:45 robot-slave snmptrapd[480]: 2022-10-11 00:02:45 <UNKNOWN> [UDP: [192.168.
↪100.3]:43304->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1167346) 3:14:33.46
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "Logs"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "WARNING: Autopurging /var/log due to extremely␣
↪high disk usage. [platform_mon.py] Usage 100: Deleting:
#0124.0K
#011/var/log/alternatives.log.2.gz
#012104K
#011/var/log/apt/term.log.3.gz
#0128.0K
#011/var/log/apt/history.log.1.gz
#01212K  Message is truncated"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

Network Monitoring

Category	Level	Message
Network Monitoring	error	ERROR: Network Failures. The following network failures occurred: ...
Network Monitoring	info	INFO: Network failures resolved ...

Examples

```
Mar 22 14:07:58 robot-sl snmptrapd[1214]:
2019-03-22 14:07:58 <UNKNOWN>
[UDP: [192.168.100.3]:18751->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1155411) 3:12:34.11
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Network Failures"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "The following network failures occured: netntp:␣
↪172.29.1.15"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Mar 29 13:57:11 robot-sl snmptrapd
[1234]: 2019-03-29 13:57:11 <UNKNOWN>
[UDP: [192.168.100.3]:32794->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (165816) 0:27:38.16
```

(continues on next page)

(continued from previous page)

```
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: Network failures resolved"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Network failures resolved"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

Memory Monitoring

Category	Level	Message
Memory Monitoring	error	ERROR: High memory usage Memory activity: . . .
Memory Monitoring	info	INFO: Memory usage returned to normal Memory more than 1024MB

Examples

```
Oct 11 00:05:02 robot-slave snmptrapd[480]: 2022-10-11 00:05:02 <UNKNOWN> [UDP: [192.168.
↪100.3]:40305->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1180986) 3:16:49.86
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: High memory usage"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Memory activity:
#012
#012223Mb"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:05:16 robot-slave snmptrapd[480]: 2022-10-11 00:05:16 <UNKNOWN> [UDP: [192.168.
↪100.3]:35079->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1182434) 3:17:04.34
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: Memory usage returned to normal"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Memory more than 1024MB"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:18:13 robot-slave snmptrapd[480]: 2022-10-11 00:18:13 <UNKNOWN> [UDP: [192.168.
↪100.3]:34120->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1260111) 3:30:01.11
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Memory swap error"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.4.100.0
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
```

(continues on next page)

(continued from previous page)

```
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.4.2.0 = STRING: "swap"
#011iso.3.6.1.4.1.2021.4.101.0 = STRING: "Running out of swap space (0)"
```

NTP

Category	Level	Message
NTP	error	ERROR: No ntp configured for %s . . .
NTP	info	INFO: ntp is now configured for %s . . .
NTP Daemon	warn	WARNING: The ntp daemon has stopped on %s . . .
NTP Daemon	info	INFO: The ntp daemon is now running on %s . . .
NTP Offset	warn	WARNING: The ntp offset exceeds 1 second on %s ..
NTP Offset	info	INFO: The ntp offset restored to normal on %s . . .

Examples

```
Oct 10 21:28:15 robot-slave snmptrapd[480]: 2022-10-10 21:28:15 <UNKNOWN> [UDP: [192.168.
↪100.3]:62399->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (240260) 0:40:02.60
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: No ntp configured for VOSS-192.168.100.3"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "It is mandatory that the ntp is configured.
#012
#012To configure ntp use the following command:
#012network ntp <server1> <server2>"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:28:23 robot-slave snmptrapd[480]: 2022-10-10 21:28:23 <UNKNOWN> [UDP: [192.168.
↪100.3]:40886->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (241068) 0:40:10.68
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: ntp is now configured for VOSS-192.168.100.
↪3"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "NTP cleared"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:29:12 robot-slave snmptrapd[480]: 2022-10-10 21:29:12 <UNKNOWN> [UDP: [192.168.
↪100.3]:41584->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (246013) 0:41:00.13
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "WARNING: The ntp daemon has stopped on VOSS-192.
```

(continues on next page)

(continued from previous page)

```
↪168.100.3"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Run 'app start services:time' to restart ntpd"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:29:44 robot-slave snmptrapd[480]: 2022-10-10 21:29:44 <UNKNOWN> [UDP: [192.168.
↪100.3]:64806->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (249163) 0:41:31.63
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: The ntp daemon is now running on VOSS-192.
↪168.100.3"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "ntp daemon running"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

Security Update

Category	Level	Message
Security Update	warn	WARNING: Security Updates available . . .
Security Update	info	INFO: Security Updates applied . . .

Service Monitoring

Category	Level	Message
Service Monitoring	error	ERROR: Service Failures . . .
Service Monitoring	info	INFO: Services started successfully . . .

Examples

```
Oct 10 23:00:35 robot-slave snmptrapd[480]: 2022-10-10 23:00:35 <UNKNOWN> [UDP: [192.168.
↪100.3]:15622->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (794329) 2:12:23.29
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: Services started successfully"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "selfservice v22.1.0 (2022-10-10 13:57)
#012   |-node                  running
#012voss-deviceapi v22.1.0 (2022-10-10 13:57)
#012   |-voss-cnf_collector    running
#012   |-voss-queue            running
#012   |-voss-wsgi  Message is truncated"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:03:03 robot-slave snmptrapd[480]: 2022-10-11 00:03:03 <UNKNOWN> [UDP: [192.168.
↪100.3]:41278->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1169102) 3:14:51.02
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: Services started successfully"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "0"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

Service Startup Changes Made

Severity levels, messages and resolution:

- Info : `ProcessRestart`, `ProcessStart`

  Resolution: If this is an unexpected event, call Support should be called for further investigation. This trap can also be triggered as expected, when **app start** or **system reboot** is run.

- Warning: `ProcessWarning`

  Resolution: This trap should be seen when a process or service is being restarted or stopped. If this is an unexpected event, call Support should be called for further investigation.

- Error : `ProcessStop`, `ProcessError`

  Resolution: If this is an unexpected event, call Support should be called for further investigation. This trap can also be triggered as expected, when **app stop** or **system reboot** is run.

Examples

```
Mar 18 07:58:18 platform-agent snmptrapd[515]: 2024-03-18 07:58:18 <UNKNOWN> [UDP: [192.
↪168.100.3]:58315->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (274360) 0:45:43.60
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "INFO: Application has changed its state from␣
↪starting to running - services:mount"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"

Mar 18 07:58:19 platform-agent snmptrapd[515]: 2024-03-18 07:58:19 <UNKNOWN> [UDP: [192.
↪168.100.3]:43944->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (274442) 0:45:44.42
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "INFO: Application has changed its state from␣
↪starting to running - services:wsgi"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"

Mar 18 07:58:20 platform-agent snmptrapd[515]: 2024-03-18 07:58:20 <UNKNOWN> [UDP: [192.
↪168.100.3]:39595->[192.168.100.25]:162]:
```

(continues on next page)

(continued from previous page)

```
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (274580) 0:45:45.80
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "INFO: Application has changed its state from␣
↪starting to running - services:syslog"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"

Oct 10 22:23:43 robot-slave snmptrapd[480]: 2022-10-10 22:23:43 <UNKNOWN> [UDP: [192.168.
↪100.3]:26646->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (573107) 1:35:31.07
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessRestart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Applications are restarting services:firewall"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"

Oct 10 22:23:54 robot-slave snmptrapd[480]: 2022-10-10 22:23:54 <UNKNOWN> [UDP: [192.168.
↪100.3]:54975->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (574250) 1:35:42.50
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application start command completed␣
↪services:firewall"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"

Oct 10 22:23:45 robot-slave snmptrapd[480]: 2022-10-10 22:23:45 <UNKNOWN> [UDP: [192.168.
↪100.3]:61828->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (573306) 1:35:33.06
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed services:firewall"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"

Oct 10 22:28:06 robot-slave snmptrapd[480]: 2022-10-10 22:28:06 <UNKNOWN> [UDP: [192.168.
↪100.3]:34679->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (599418) 1:39:54.18
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStop"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Applications are stopping mongodb"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"

Oct 11 00:00:34 robot-slave snmptrapd[480]: 2022-10-11 00:00:34 <UNKNOWN> [UDP: [192.168.
↪100.3]:48544->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1154165) 3:12:21.65
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessError"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application failed with error 1 nginx:proxy"
```

(continues on next page)

(continued from previous page)

```
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

Excessive Load

Load average interval	<LoadIdx>	<LoadError>	<LoadMessage>
1 minute	1	Load-1	1 min Load Average too high (= 2.52)
5 minute	2	Load-5	5 min Load Average too high (= 1.27)
15 minute	3	Load-15	15 min Load Average too high (= 1.27)

Examples

```
Oct 11 00:13:48 robot-slave snmptrapd[480]: 2022-10-11 00:13:48 <UNKNOWN> [UDP: [192.168.
↪100.3]:10511->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1233585) 3:25:35.85
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Excessive load"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.10.1.100.1
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.10.1.2.1 = STRING: "Load-1"
#011iso.3.6.1.4.1.2021.10.1.101.1 = STRING: "1 min Load Average too high (= 5.59)"

Oct 11 00:13:48 robot-slave snmptrapd[480]: 2022-10-11 00:13:48 <UNKNOWN> [UDP: [192.168.
↪100.3]:16438->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1233630) 3:25:36.30
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Excessive load"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.10.1.100.2
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.10.1.2.2 = STRING: "Load-5"
#011iso.3.6.1.4.1.2021.10.1.101.2 = STRING: "5 min Load Average too high (= 2.43)"

Oct 11 00:15:48 robot-slave snmptrapd[480]: 2022-10-11 00:15:48 <UNKNOWN> [UDP: [192.168.
↪100.3]:43788->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1245577) 3:27:35.77
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Excessive load"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.10.1.100.3
```

(continues on next page)

(continued from previous page)

```
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.10.1.2.3 = STRING: "Load-15"
#011iso.3.6.1.4.1.2021.10.1.101.3 = STRING: "15 min Load Average too high (= 3.14)"
```

Transaction Archive

Category	Level	Message
Transaction Archive	error	ERROR: Transaction archive files older than 30 days...
Transaction Archive	info	INFO: Transaction archive files OK...

Upgrade

Category	Level	Message
Upgrade	error	ERROR: upgrade failed
Upgrade	error	ERROR: upgrade failed upgrade failed as other activity is in progress ...

Examples

```
Oct 10 22:30:17 robot-slave snmptrapd[480]: 2022-10-10 22:30:17 <UNKNOWN> [UDP: [192.168.
↪100.3]:32373->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (612524) 1:42:05.24
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "upgrade failed"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "upgrade failed as other activity is in progress"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

NGINX Status

Level:

- Error: nginx upstream failure, upstream <node> server <server> failed
- Info: nginx upstreams OK

Examples

```
Mar 25 13:30:12 robot-sl snmptrapd[1214]:
2019-03-25 13:30:12 <UNKNOWN>
[UDP: [172.29.21.129]:63573->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (293333) 0:48:53.33
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "nginx upstream failure"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "upstream selfservice server 192.29.22.122:443␣
↪failed: <urlopen error timed out>"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "WP2-172.29.21.129"
```

```
Mar 25 13:34:02 robot-sl snmptrapd[1214]:
2019-03-25 13:34:02 <UNKNOWN>
[UDP: [172.29.21.129]:31265->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (316311) 0:52:43.11
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "nginx upstreams OK"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "nginx upstream servers returned to normal"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "WP2-172.29.21.129"
```

Web Certificate

A trap is generated when the nginx web certificate is about to expire.

Three intervals are checked which will return a result:

- 30 days prior to expiry
- 14 days prior to expiry
- 1 day prior to expiry

After expiry, an alert is sent every day.

Category	Level	Message
Web Certificate Maintenance	error	ERROR: Web certificate about to expire..
Web Certificate Maintenance	info	INFO: Web certificate has been renewed . . .

Examples

```
Oct 10 22:58:45 robot-slave snmptrapd[480]: 2022-10-10 22:58:42 <UNKNOWN> [UDP: [192.168.
↪100.3]:39688->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (782988) 2:10:29.88
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "Certificate_Maintenance"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "ID: WEB_CERTIFICATE_VOSS, Code: -1, Occurences:␣
↪1, Latest Occurence: 2022-10-10T20:58:41.312Z"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:58:45 robot-slave snmptrapd[480]: 2022-10-10 22:58:45 <UNKNOWN> [UDP: [192.168.
↪100.3]:39098->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (783366) 2:10:33.66
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Web certificate about to expire"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Cert will expire in less than 30 day(s)"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:58:48 robot-slave snmptrapd[480]: 2022-10-10 22:58:48 <UNKNOWN> [UDP: [192.168.
↪100.3]:15074->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (783622) 2:10:36.22
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: Web certificate has been renewed"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Web certificate has been renewed"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

Example GUI alert shown in the `data/Alert` instance in the Database

- AlertCode: “25005”
- Category: “Certificate_Maintenance”
- AlertId: “WEB_CERTIFICATE_<certificate-name>”
- Message: “The certificate will expire in less than {} day(s).” (number of days - alert will result)
- Severity: WARNING
- alert_timestamp: <timestamp>

Example error raised through platform monitoring

```
Notification message from (1, 3, 6, 1, 6, 1, 1):('10.120.1.203', 5194):
Var-binds:
1.3.6.1.2.1.1.3.0 = 16128774
1.3.6.1.6.3.1.1.4.1.0 = 1.3.6.1.2.1.88.2.0.1
1.3.6.1.2.1.88.2.1.1.0 = ERROR: Web certificate about to expire
1.3.6.1.2.1.88.2.1.3.0 = Cert will expire in less than {} day(s)
1.3.6.1.2.1.88.2.1.5.0 = 1
1.3.6.1.2.1.1.1.5.0 = VOSS-UN-1
```

Resolution

Renew the web certificate - see: *Web Certificate Setup Options*.

Thresholds

Examples

```
Oct 10 21:30:04 robot-slave snmptrapd[480]: 2022-10-10 21:30:04 <UNKNOWN> [UDP: [192.168.
↪100.3]:63587->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (251207) 0:41:52.07
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "rsyslogd : num_files exceeded maximum value of␣
↪30 with 78"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Command rsyslogd attribute num_files
#012Current maximum value 78
#012Error threshold 50
#012Warning threshold 33
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.1.5.0 = STRING: "VOSS"

Oct 10 21:30:04 robot-slave snmptrapd[480]: 2022-10-10 21:30:04 <UNKNOWN> [UDP: [192.168.
↪100.3]:63524->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (251245) 0:41:52.45
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "sysmon : num_sockets exceeded maximum value of␣
↪10 with 11"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Command sysmon attribute num_sockets
#012Current maximum value 11
#012Error threshold 10
#012Warning threshold 6
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.1.5.0 = STRING: "VOSS"


Oct 10 22:40:03 robot-slave snmptrapd[480]: 2022-10-10 22:40:03 <UNKNOWN> [UDP: [192.168.
↪100.3]:62002->[192.168.100.25]:162]:
```

(continues on next page)

(continued from previous page)

```
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (671171) 1:51:51.71
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "containerd : num_sockets exceeded maximum value␣
↪of 16 with 17"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Command containerd attribute num_sockets
#012Current maximum value 17
#012Error threshold 10
#012Warning threshold 6
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"


Oct 10 22:50:03 robot-slave snmptrapd[480]: 2022-10-10 22:50:03 <UNKNOWN> [UDP: [192.168.
↪100.3]:34433->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (731163) 2:01:51.63
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "execute : num_sockets exeeded warning threshold␣
↪of 6 with 6"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Command execute attribute num_sockets
#012Current average value 6
#012Error threshold 10
#012Warning threshold 6
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"

Oct 10 23:50:04 robot-slave snmptrapd[480]: 2022-10-10 23:50:04 <UNKNOWN> [UDP: [192.168.
↪100.3]:15565->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1091211) 3:01:52.11
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "containerd-shim : num_files exceeded maximum␣
↪value of 19 with 29"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Command containerd-shim attribute num_files
#012Current maximum value 29
#012Error threshold 20
#012Warning threshold 13
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"

Oct 10 23:50:05 robot-slave snmptrapd[480]: 2022-10-10 23:50:05 <UNKNOWN> [UDP: [192.168.
↪100.3]:2699->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1091258) 3:01:52.58
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "checkmongoalive : num_files exeeded warning␣
↪threshold of 13 with 20"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Command checkmongoalive attribute num_files
#012Current average value 20
#012Error threshold 20
#012Warning threshold 13
```

(continues on next page)

(continued from previous page)

```
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

Resolution:

Check if `mongodb` is up and running: Use the command `app status` to see if `mongodb` is running. If not, run `app start mongodb`. Else contact VOSS support.

```
Oct 10 21:40:06 robot-slave snmptrapd[480]: 2022-10-10 21:40:06 <UNKNOWN> [UDP: [192.168.
↪100.3]:53579->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (311359) 0:51:53.59
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "backup.py : num_sockets exeeded warning␣
↪threshold of 6 with 7"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Command backup.py attribute num_sockets
#012Current average value 7
#012Error threshold 10
#012Warning threshold 6
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"

Oct 10 22:00:16 robot-slave snmptrapd[480]: 2022-10-10 22:00:16 <UNKNOWN> [UDP: [192.168.
↪100.3]:40320->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (432428) 1:12:04.28
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "docker-proxy : num_sockets exceeded maximum␣
↪value of 7 with 27"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Command docker-proxy attribute num_sockets
#012Current maximum value 27
#012Error threshold 20
#012Warning threshold 13
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"

Oct 10 22:00:17 robot-slave snmptrapd[480]: 2022-10-10 22:00:17 <UNKNOWN> [UDP: [192.168.
↪100.3]:39273->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (432500) 1:12:05.00
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "docker-proxy : num_files exceeded maximum value␣
↪of 17 with 57"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Command docker-proxy attribute num_files
#012Current maximum value 57
#012Error threshold 20
#012Warning threshold 13
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"

Oct 11 00:00:15 robot-slave snmptrapd[480]: 2022-10-11 00:00:15 <UNKNOWN> [UDP: [192.168.
```

(continues on next page)

(continued from previous page)

```
↪100.3]:55586->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1152330) 3:12:03.30
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "dockerd : num_files exceeded maximum value of 78␣
↪with 84"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Command dockerd attribute num_files
#012Current maximum value 84
#012Error threshold 80
#012Warning threshold 53
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

11.4.2. SNMP Traps: Number Inventory Alerting

SNMP traps are triggered when INI availability threshold is reached.

Example alert

```
2022-03-02 08:40:18 <UNKNOWN>
[UDP: [172.17.9.1]:59991->[172.17.9.2]:162]:
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (3878204) 10:46:22.04
SNMPv2-MIB::snmpTrapOID.0 = OID: DISMAN-EVENT-MIB::mteTriggerFired
DISMAN-EVENT-MIB::mteHotTrigger.0 = STRING: Number Inventory
DISMAN-EVENT-MIB::mteHotContextName.0 = STRING:
 ID: Number Inventory Threshold of 80% Exceeded for Provider [CS-P].,
 Code: 110000,
 Occurences: 20,
 Latest Occurence: 2022-03-02T08:40:17.188Z
DISMAN-EVENT-MIB::mteHotValue.0 = INTEGER: 1
SNMPv2-MIB::sysName.0 = STRING: VOSS
```

11.5. VOSS Automate System Monitoring Traps

11.5.1. SNMP Traps: System Monitoring

Administrators at `sysadmin` level can configure additional SNMP traps for alerts from the **System Monitoring > Configuration** menu on the GUI (menu model: `data/SystemMonitoringConfig`). Note: some traps are not configurable.

Refer to the topic on System Monitoring Configuration in the Advanced Configuration Guide.

The following alerts are configured:

Notification	Interval	Level	Configurable
Txn Queue Size	Hourly	warn	Yes
Failed Txn	Immediate	warn	Yes
Stuck in Queued	Hourly	error	Yes
Stuck in Processing	Hourly	error	Yes
data/Alert (CNF atm)	Immediate	Alert defined	No
Session Exceeded	Immediate	warn	Yes (via platform CLI command)
API Request Throttled	Immediate	warn	Yes (via platform CLI command)
Total DB Index Size	Daily	warn	Yes
Total DB Size	Daily	warn	Yes
Device Comms. Concurrency Limit	Immediate	warn	No

For platform CLI commands for session limits and throttle rates, see: *Performance Commands*.

Transaction Queue Size

In accordance with the configurable threshold (default 500)

Identifying strings and example context:

```
DISMAN-EVENT-MIB::mteHotTrigger.0 = STRING: Transaction Queue Size Exceeded Threshold
DISMAN-EVENT-MIB::mteHotContextName.0 = STRING: Current Size: 520 Threshold: 500
```

Transactions: maximum time in Queued and Processing state

Alerts on transactions exceeding maximum configured queued and processing time.

Identifying strings:

```
DISMAN-EVENT-MIB::mteHotTrigger.0 = STRING:

Stuck in Queued: <n> transaction(s) 'Queued' for too long.
Stuck in Processing: <n> transaction(s) 'Processing' for too long.
```

Example: Queued

```
2021-11-19 15:31:38 <UNKNOWN>
[UDP: [192.168.100.3]:13177->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (17398363) 2 days, 0:19:43.63
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "Stuck in Queued"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING:
 "ID: Transactions,
 Code: 72054,
 Occurences: 44,
 Latest Occurence: 2021-11-19T13:31:36.948Z"
```

(continues on next page)

(continued from previous page)

```
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

Example: Processing

```
2021-11-19 18:31:40 <UNKNOWN>
[UDP: [192.168.100.3]:47295->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (18478492) 2 days, 3:19:44.92
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "Stuck in Processing"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING:
 "ID: Transactions,
 Code: 72055,
 Occurences: 39,
 Latest Occurence: 2021-11-19T16:31:38.655Z"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

Transactions: Model Operations Alerts

- Alerts on transactions failure
  - per model (wild cards allowed, default is `data/*`)
  - model operations (default is **Import**)

Identifying string:

```
DISMAN-EVENT-MIB::mteHotTrigger.0 = STRING: Transaction Completed with Fail
```

Transaction trap context information (200 chars):

- `ID`: transaction ID (same as on GUI - further transaction details available on GUI)
- `Action`: transaction message (same as on GUI)
- `Detail`: source of resource (source host for import)
- `Hierarchy`: friendly path of the resource, else the execution hierarchy of transaction

Example: Import Fail

```
2019-03-28 10:54:46 <UNKNOWN>
[UDP: [192.168.100.3]:31384->[192.168.100.25]:162]:
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (170158257) 19 days, 16:39:42.57
SNMPv2-MIB::snmpTrapOID.0 = OID: DISMAN-EVENT-MIB::mteTriggerFired
DISMAN-EVENT-MIB::mteHotTrigger.0 = STRING:
 Transaction Completed with Fail
DISMAN-EVENT-MIB::mteHotContextName.0 = STRING:
 ID: 44967,
 Action: Import Call Manager,
 Detail: 192.168.100.15,
 Hierarchy: sys
```

(continues on next page)

(continued from previous page)

```
DISMAN-EVENT-MIB::mteHotValue.0 = INTEGER: 1
SNMPv2-MIB::sysName.0 = STRING: VOSS
```

Change Notification Feature (CNF)

CNF traps are triggered when Change Notification Sync transactions add or update instances on the `data/Alerts` model.

The identifying alert string is:

```
DISMAN-EVENT-MIB::mteHotTrigger.0 = STRING: Device Change Notification
```

The `data/Alerts` attribute values of the model are provided in the traps details:

```
alert_severity
alert_category
alert_timestamp
alert_count
alert_id
alert_message
alert_code
```

For example , the trap Context information (200 chars) is:

- `ID`: Device Host business key (`alert_id`)
- `Code`: CNF Alert code (`alert_code`)
- `Occurrences`: number of occurrences
- `Latest Occurrence:`: time stamp (`alert_timestamp`)

Warning and Error Alert Codes

The following table shows alert codes and details

Code	Details	Resolution
72051	ERROR. Device connectivity failure	For connectivity checks, see *UC Apps Reachability* in the Advanced Configuration Guide and also the Platform Guide and Health Checks for Cluster Installations Guide.
72052	WARNING. Slow device connection: the roundtrip time (RTT) is greater than 400ms.	For latency checks, see *UC Apps Reachability* in the Advanced Configuration Guide and the Health Checks for Cluster Installations Guide.
72053	ERROR. Utilization approaching limit: if the maximum number of sessions during the interval exceeds 80% of the configured limit.	The threshold can be reached if many users are using the system at the same time, or by not logging out. For **Utilization %**, see *Login Sessions* in the Advanced Configuration Guide.
72054	ERROR. Stuck in Queued: <n> transaction(s) 'Queued' for too long.	The error is raised if transactions are in *queued* state longer than configured maximum time. See *System Monitoring Configuration* in the Advanced Configuration Guide.
72055	ERROR. Stuck in Processing: <n> transaction(s) 'Processing' for too long.	The error is raised if transactions are in *processing* state longer than configured maximum time. See *System Monitoring Configuration* in the Advanced Configuration Guide.

Example: CNF alert

```
2019-03-28 10:54:46 <UNKNOWN>
[UDP: [192.168.100.3]:31384->[192.168.100.25]:162]:
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (170158257) 19 days, 16:39:42.57
SNMPv2-MIB::snmpTrapOID.0 = OID: DISMAN-EVENT-MIB::mteTriggerFired
DISMAN-EVENT-MIB::mteHotTrigger.0 = STRING: Device Change Notification
DISMAN-EVENT-MIB::mteHotContextName.0 = STRING:
 ID: 44967,
 Code: 100034,
 Occurrences: 1,
 Latest Occurrence: 2019-03-28 10:54:44Z
DISMAN-EVENT-MIB::mteHotValue.0 = INTEGER: 1
SNMPv2-MIB::sysName.0 = STRING: VOSS
```

Session Limits

SNMP traps are triggered when session limits are reached.

Example:

For example, the customer administrator session limit default is 10 and a trap is triggered if it is exceeded. (The default can be configured with the **voss session-limits** command).

Note: Global session limits do not show a `Hierarchy` value in the message string.

```
2019-03-28 10:54:46 <UNKNOWN>
[UDP: [192.168.100.3]:31384->[192.168.100.25]:162]:
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (170158257) 19 days, 16:39:42.57
SNMPv2-MIB::snmpTrapOID.0 = OID: DISMAN-EVENT-MIB::mteTriggerFired
DISMAN-EVENT-MIB::mteHotTrigger.0 = STRING: Customer Administration Session Limit␣
↪Exceeded
DISMAN-EVENT-MIB::mteHotContextName.0 = STRING:
 Limit: 10,
 Hierarchy: sys.hcs.Varidion.GSCorp
DISMAN-EVENT-MIB::mteHotValue.0 = INTEGER: 1
SNMPv2-MIB::sysName.0 = STRING: VOSS
```

API Request Throttle

SNMP traps are triggered when throttle rates are reached.

Throttle rates are configured with:

> **voss throttle-rates type <administration|selfservice|user> requests <number of requests> unit <min|sec>**

In other words, the SNMP trap would be triggered for request limits for any of:

- Administration
- Self-service
- User-specific

Identifying strings and Self-service as example:

```
DISMAN-EVENT-MIB::mteHotTrigger.0 = STRING: Selfservice Api Request Limit Exceeded
DISMAN-EVENT-MIB::mteHotContextName.0 = STRING: Rate 20/min
```

Total DB Index Size

In accordance with the configurable threshold (default 50)

Identifying strings and example:

```
DISMAN-EVENT-MIB::mteHotTrigger.0 = STRING: DB Index Size Exceeded Threshold
DISMAN-EVENT-MIB::mteHotContextName.0 = STRING: DB Index Size (60.00GB) exceeded␣
↪threshold (50GB)
```

Total DB Size

In accordance with the configurable threshold (default 200)

Identifying strings and example:

```
DISMAN-EVENT-MIB::mteHotTrigger.0 = STRING: DB Size Exceeded Threshold
DISMAN-EVENT-MIB::mteHotContextName.0 = STRING: DB Size (210.30GB) exceeded threshold␣
↪(200GB)
```

Device Communications Concurrency Limit

SNMP traps are sent if there is a timeout failure while connecting to a device and waiting for the concurrency limit.

Current concurrency support:

- 8 concurrent requests to Unified CM
- 8 concurrent requests to Unity Connection

12. Scheduling

12.1. Scheduling

Any CLI command can be scheduled to run automatically, including but not restricted to backups and security upgrades.

By default there is no backup maintenance scheduled. Backup maintenance can be scheduled with the number of copies to be kept - refer to the backup maintenance topic.

Note: Scheduled commands will not run while the system is in maintenance mode. See System Maintenance Mode in the Platform Guide.

The automated job schedule format is as follows:

- **schedule add <job-name> <user-command>**
- **schedule time <job-name> <hour> <minute>**
- **schedule time <job-name> every <N> hours**
- Alternatively the job can be scheduled to run every week on Monday with **schedule time <job-name> weekly 1**; where 0 is Sunday, 1 is Monday, 2 is Tuesday, 3 is Wednesday, 4 is Thursday, 5 is Friday and 6 is Saturday
- **schedule enable <job-name>**

Example:

schedule add mybackups backup create localbackup

schedule time mybackups 2 0

schedule time mybackups weekly 0

schedule enable mybackups

Among the tasks that can be scheduled are:

- Backup creation, e.g. **schedule add backupme backup create localbackup**
- Backup maintenance, e.g. **schedule add backupclean backup clean localbackup keep 5**
- Health reports, e.g. **schedule add reports diag report**

Note: If a schedule is in a state where the last executed and next execution time are equal, then the next execution time will be recalculated to ensure its execution.

12.2. Internal Report Schedules

The system runs an internal schedule to generate monthly license reports. For details on license reports and how to generate these manually, refer to the Licensing Guide.

This internal schedule cannot be disabled. The schedule is configured to run at 3AM UTC on the first day of the month. The date cannot be changed, but the time can. Please contact your VOSS account manager if a schedule time change is required.

After the monthly schedule is run, a check is carried out for the generated report. If the report was generated successfully, no messages are sent and no notifications are generated. If the report was not generated successfully, a message shows on the CLI console when logging in or when typing the **health** command:

```
LICENSE REPORT: FAILED - Please run 'voss export type license_initial_audit'
```

This message will continue to show until the report is generated successfully by running the command shown in the message.

An e-mail notification is also sent after the check fails:

```
ERROR: License file generation failed
The license audit report scheduled for <month> <year> was not successful.
Please contact your VOSS account manager.
```

An example SNMP trap that is generated when the report fails to run is show below - <month> <year> are variables in the example:

```
May 23 02:01:00 robot-slave snmptrapd[18891]: 2018-05-23 02:01:00 <UNKNOWN>
[UDP: [192.168.100.3]:11814->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (207758) 0:34:37.58
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING:
  "ERROR: License file generation failed"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING:
 "The license audit report scheduled for <month> <year> was not successful.
#012Please contact your VOSS account manager. "
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

13. Backups

13.1. Backups

Backups represent a snapshot of the system, including database, configuration and system applications. Backups can be created manually, scheduled automatically, or created automatically when the system is upgraded.

Note:

- In a multinode environment, database backups are created from the highest priority secondary database node, thereby reducing the system load on the primary database node.

  Use the **database config** command to check the secondary node priority.

- The **backup** command on a Generic NBI node (for Billing Data Extract) is not in use.

These backups are encrypted and can be stored on the local file system, or to a remote network location. The encryption key is needed to delete, export, restore and create a backup.

Off-site (non-local) backups are recommended, because this reduces the risk of sabotage or disk failures causing a loss of information.

As part of the rollback procedure, ensure that a suitable restore point is obtained prior to the start of the activity, as per the guidelines for the infrastructure on which the VOSS Automate platform is deployed.

There is no direct requirement for VMware snapshots. If VMware snapshots are used, also refer to the topic on *VMware Snapshot Maintenance*.

For examples of backup maintenance commands and output, refer to the topics on Scheduling and Create a Backup.

In the case of a single node cluster, if the VOSS Automate node is not recoverable, due to for example a hardware failure, a new node can be deployed and an existing backup restored to restore the node to service.

When restoring a backup on a cluster, the latest backup from the highest weighted secondary database server must be used to ensure the entire cluster (excluding the web certificates) including the database is restored and the cluster is automatically provisioned as part of the restore.

13.2. Backup and Restore in a Modular Architecture

Backups that were made *before* topology migration to a modular architecture can not be restored to the topology *after* migration. Manual intervention is required to restore.

A new modular architecture backup must be created immediately after topology migration. This backup will be on the highest priority secondary db node.

Important:

- Backups for an Application and Web Proxy type node are not required. A backup created on the secondary database node will include all the relevant data, excluding the web certificates.

  After a restore is performed by following the documented process: *Restore the Backup*, the web certificates need to be reinstalled on each of the Application and Web Proxy nodes. For commands, see: *Web Certificate Commands*. These certificates should already be stored in a safe location after initially installed, therefore do not require inclusion in the backups.

- When restoring a backup on a cluster, the latest backup from the highest weighted secondary database server must be used to ensure the entire cluster (excluding the web certificates) including the database is restored and the cluster is automatically provisioned as part of the restore.

13.3. Backup Destinations

Backups can be made to the local file system or a remote destination. Off-site (non-local) backups are recommended to reduce the risk of the loss of information.

- Display available backup destinations with **backup list**.
- Add a new backup destination with **backup add <location-name> <URI>**.

  See *Network URI specification*.

Local backups are stored on a separate backup volume and the `localbackup` destination is pre-configured.

- Display the list of localbackups with **backup list localbackup**, for example:

```
$ backup list localbackup
   localbackup:
      URI: file:///backups
      Backups:
         2016-06-21 13:33
         2016-06-21 13:16
```

If the backup volume is too small, it can be increased - see: *Backup size considerations*. If the `localbackup` destination is removed or renamed, an ISO file upgrade will no longer function. Therefore, it is imperative that this destination is not removed.

Example:

```
backup add myserverbackup sftp://user:password@server/path
```

Azure example:

```
backup add myserverbackup sftp://<storage_account_name>.<sftp_local_user>[:password]@
↪<storage_account_name>/path
```

- <storage_acct_name> is set up on Azure. The name of the Storage Account.
- <sftp_local_user> is set up on Azure if you enable SFTP and add a local user within the Storage Account settings.

Note: During Azure host key rotation, in order to allow for a smooth transition when the rotation occurs, both the existing and new key should be available as trusted hosts. This can be carried out and verified with the `keys add <host>`, `keys send <user>@<host>` and `keys show` commands. See: *SSH key management*.

Backups to sftp require ssh key-based authentication to be setup. Refer to *SSH key management* for further details. If you select ssh key-based authentication that was set up without a password, you will be prompted to input the password.

```
platform@VOSS-UN-6:~$ backup add remote sftp://dan@182.59.31.201/sftp
No password found. Do you want to use sshkeys? [y/N]: y
```

Important: If the username or password contains any of the following characters:

```
;|\\$&`!
```

then use the backup setup command interactively by choosing “No” at the prompts if SSH Keys are being used and then add the username and password directly.

If a common remote backup point is to be used by all nodes in the cluster, the backup destination needs to be added to each node. This can be automated by using cluster remote execution, for example:

```
cluster run all backup add myserverbackup sftp://user:password@server/path
```

The creation of scheduled backups of all nodes is done for failover reasons.

Note: In a multinode configuration, while only the highest priority secondary node backup contains data, in a failover scenario, the new primary node will contain the restored backup data.

13.4. Backup Passphrase

System backups are encrypted. The encryption key is initially set as the platform user’s password as set in the installation wizard.

An encryption key is required to delete, export, restore and create a backup. If the backups are on an external system, they can be deleted manually.

It is recommended that this be changed after installation.This can be done by running **backup passphrase**.

Important: Changing the passphrase on a node affects only the backups created on that node. The passphrase should be changed on *all* nodes to keep the passphrase in sync.

The passphrase is subject the same rules as a password. For example, if the minimum password length is set using **user password length**, this also applies to the passphrase. See also Password Strength Rules.

The following example shows the console output:

```
platform@masternode:~$ backup passphrase
Please enter current backup passphrase
Password:
Please enter new backup passphrase
Password:
Please re-enter new backup passphrase
Password:

Backup passphrase successfully changed
```

Keep this password, because restoring the backup to a new system requires this password.

To restore on the new system, run **backup passphrase** and enter the password used to create the backup.

13.5. Backup size considerations

13.5.1. Default backup partition size

The default backup partition size is 50GB for the default 250GB database partition size. These are the default partition installation sizes.

Important: Backups should be created and restored in a `tmux` session. For details, refer to *Using the tmux command*.

13.5.2. Backup space requirements

Consider the following:

- A local backup requires free space of at least twice the size of the database. Preferably add an additional 30%.

  At any given time, it is recommended that available backup space should be approximately twice the database size plus 30% of the database size.

  For remote backups, the backup requires free space of at least the size of the database plus an additional 30%.
- Database growth over time needs to be considered and allowed for in the backup partition size.
- Space for multiple local backups also needs to be considered and added to the calculated backup partition size.

13.5.3. Run a size check

To determine the required space needed to perform a backup for a specific backup partition, from the CLI, run the following command:

`backup create <location-name>`

The command output indicates the required space needed to do the backup.

If the current backup partition size is too small, the command will fail and suggest the size of the partition required.

13.5.4. Increase size

To increase the size of the backup partition, use the following command: `drives add`

Refer to the Drive Control topic in the Platform Guide.

If there is sufficient space but only a size check is required, the backup command can be canceled (**Ctrl-C**), if needed.

13.5.5. Display size of current database

To display the size of the current database, run: `voss db_collection_stats all`

This command validates the size of the database. This total will be smaller than the suggested backup size.

13.6. Create a Backup

Note:

- On a multi-node system, to reduce the system load on the primary database node, backups are processed by the available secondary unified node with the highest database priority, i.e. the secondary node that is not for example in a recovery state. (To check priorities: **database config**.)

  Only if no secondary unified node is available or on a standalone single-node cluster system, is the backup processed on the primary unified node.
- For best performance, it is recommended that remote backups be sent to a SFTP server at the *same data center* as the node processing the backup.

Important: Backups should be created and restored in a `tmux` session - see: *Using the tmux command*.

Backups can be created using **backup create <location-name>**, for example:

backup create localbackup or **backup create myserverbackup**.

An example of the console output is shown below:

```
platform@myhost:~$ backup create localbackup

... collecting data (step 1/3)
   ... preparing mongo data backup
   ... space available: 232 GB
   ... space required: 93 MB
... creating backup (step 2/3)
... verifying backup (step 3/3)
Backup was successfully created at localbackup:::\
  058bccead2588a6f11f1dd86678bab68de48691d

WARNING:  Backup maintenance of this location is not scheduled
    schedule add localbackup -maintain backup clean localbackup keep 5

You have new mail in /var/mail/platform

platform@myhost:~$ backup list localbackup
localbackup:
   URI: file:///backups
   Backups:
      2019-08-13 13:08
```

- Backups contain application data.
- Details of the backup can be seen in the log: **log view platform/backup.log**

Backups can be scheduled to run automatically - refer to the **schedule** command to automated backups.

For example:

- **schedule add mybackups backup create myserverbackup**
- **schedule time mybackups 2 0**
- **schedule enable mybackups**

The creation of scheduled backups of all nodes is therefore done for failover reasons. While only the available secondary unified node with the highest database priority contains data, in a failover scenario, the new secondary unified node with the highest database priority will contain the backup data.

If a common network URI is used as backup destination across the cluster, each node's backup will be uniquely identified by its UID in the remote backup directory.

Note: Off-site backups are recommended. In other words, export a local backup to a remote sftp server. Follow the process as described in the topic called Backup and Import to a New Environment.

13.7. Restore a Backup in a Clustered Environment

In a clustered environment, servers can allow for failures and can keep data intact, because when a server fails, an automatic failover occurs.

If all services are kept running and data remains accessible, a **backup restore** would only be necessary in very specific scenarios.

Restoring a backup in a cluster would only be necessary in the following cases:

- Data Corruption (Bad Data)
- Losing the whole cluster - requiring a redeploy of new servers

When restoring a backup on a cluster, the latest backup from the highest weighted secondary database server must be used to ensure the entire cluster (excluding the web certificates) including the database is restored and the cluster is automatically provisioned as part of the restore.

13.8. Create Space for a Backup or Restore

If a `No space left on device` message is received during a backup or a restore, carry out the following steps:

1. In VMware, add a disk to the system:
    a. Click on **VM > Edit Settings...**.
    b. Click **Add...**.
    c. Select **Hard Disk**, then **Create a new virtual disk**.
    d. Set the size to be the same as the DB disk - 250GB.
    e. Click **Finish**
2. Log into platform account, and run **drives list**. Make note of the disk under `Unused disks:`.
3. Run `drives reassign <disk from step 2> services:backups`

Once done, all current data would have been moved to new disk and the old one can be removed from VMware. The **restore** command can now be rerun.

13.9. Maintaining Backups

A complete list of backups on a location can be displayed using **backup list <location>**.

Backups can be deleted using the following commands:

- **backup clean <location> keep <N>** will delete older copies so that only N copies are kept. Note: `<N>` must be larger than 0.
- **backup clean <location> before <yyyy-mm-dd [HH:MM]>** will delete copies older than the specified date.

By default, there is no regular maintenance of backups, and a scheduled job should be created to perform this maintenance, for example:

- **schedule add backuprotate backup clean localbackup keep 5**
- **schedule time backuprotate 3 0**
- **schedule enable backuprotate**

13.10. Backup and Import to a New Environment

The steps below show how to backup and import to a new environment.

1. Export:
   a. On the source system, create a remote backup location, for example location name `sftpbackup`:

      backup add sftpbackup <URI>

      For example:

      ```
      backup add sftpbackup sftp://sftpusr:sftpw@172.29.41.107/home/sftp
      ```

      If a directory is specified in the <URI>, this will be created during the backup. Backups to sftp require ssh key-based authentication to be set up. Refer to *SSH key management* for further details.

      Alternatively, enter the password at the prompt, for example:

      ```
      $ backup add sftpbackup sftp://sftpusr:sftpw@172.29.41.107/home/sftp

       No password found. Do you want to use sshkeys? no
       What is the host ssh password?

       <type password here>

       Location has been added
      ```

   b. Create a local backup:

      Important: Backups should be created and restored in a `tmux` session - see: *Using the tmux command*.

      backup create localbackup

      In a multi-node configuration, the database backup will be created on the secondary node with the highest priority. Use **database config** to check the priority.

   c. List backups to get the date:

      backup list

      For example:

      ```
      $ backup list
       localbackup:
          URI: file:///backups
          Backups:
      ```

(continues on next page)

(continued from previous page)

```
      1 backups have been created - most recently 2020-03-19 08:21
sftpbackup:
   URI: sftp://sftpusr:********@172.29.41.107:home/sftp
   Backups:
      No backups created yet
```

d. Export the local backup to the remote destination created by **backup add <remote_name>**.

- The system ID is appended as a directory to the backup <URI> destination path. This can be checked locally by running **system id**.
- The backup file is called <hostname>_<timestamp>.tar.gz.

Example output:

```
platform@VOSS:~$ backup export localbackup sftpbackup 2020-04-02 11:34
This operation could take a while if the backup is sizeable. Do you wish to␣
↪continue? y
Compressing backup files for date 2020-04-02 11:34
Backup files successfully compressed to 202004021134.tar.gz
Backup files successfully compressed to /backups/
↪49940d3feaa39a6a9f36cb5ff533202157c3b77a/VOSS_202004021348.tar.gz
platform@10.120.1.246's password:
platform@10.120.1.246's password:
Export successfully created at platform@10.120.1.246:media/
↪49940d3feaa39a6a9f36cb5ff533202157c3b77a/VOSS_202004021348.tar.gz
```

2. Import:

Note: For large backup files that cannot be imported due to space limitations, see: *Restoring backup files too large for import*.

---

a. From the SFTP server, **scp** the VOSS_202004021348.tar.gz file to the new box (for example, platform@172.29.21.97). If the file on the SFTP server is in the directory /backups/49940d3feaa39a6a9f36cb5ff533202157c3b77a, change to the directory, then:

```
$ ls

 VOSS_202004021348.tar.gz

$ scp VOSS_202004021348.tar.gz platform@172.29.21.97:/opt/platform/admin/home/
↪media/
```

b. Import the file as a local backup, for example:

```
$ backup import localbackup media/VOSS_202004021348.tar.gz
```

c. Get the file timestamp of the imported backup with **backup list** and restore the backup, for example:

Important: Backups should be created and restored in a tmux session - see: *Using the tmux*

command.

```
$  backup restore localbackup 2020-04-02 15:41
```

13.10.1. Restoring backup files too large for import

If a backup file is available and is too large to copy into the local system `/home/media/` directory, the following procedure can be followed. (Available space on `home/media` can be checked using **diag disk**.)

1. On the system where the backup is to be restored, add the remote SFTP site, for example:

```
backup add sftpbackup sftp://sftpusr:sftpw@172.29.41.107/home/sftp
```

2. Obtain the system UUID by running **system id**. It should look like this:

```
49940d3feaa39a6a9f36cb5ff533202157c3b77a
```

3. Given an available backup file, e.g. `VOSS_202004021348.tar.gz`, (with timestamp reference: `2020-04-02 15:41`) this file can be extracted to the *remote* SFTP site configured above as follows:
   a. On the remote SFTP site, locate the sub-directory with the name of the UUID, for example

```
/home/sftp/49940d3feaa39a6a9f36cb5ff533202157c3b77a
```

   b. Extract the backup file in this directory. Ensure that the extracted backup files are listed in the root of this directory (not all in a subdirectory).
4. On the system where the backup is to be restored, run the remote restore, for example:

```
$  backup restore sftpbackup 2020-04-02 15:41
```

13.11. VMware Snapshot Maintenance

It is not recommend to keep more than two snapshots at any one time as this can negatively affect performance. Refer to the VMware Knowledge Base topic on Best practices for virtual machine snapshots in the VMware environment.

Unused or deprecated snapshots on VMware are caused by multiple snapshot creation and deletion. These can be removed to save space. Note however that these snapshots and disk images may not show in the vSphere GUI admin tool (Snapshot Manager).

Follow the steps recommended by VMware to remove deprecated, orphaned, unused and old snapshots via your VMware administrator. Pay special attention to the `.vmx` configuration file to avoid removing live disks.

Always perform a backup and export of your data. We recommend shutting down your VOSS Automate instance(s) during a maintenance window in order to remove all unused images and to create a fresh snapshot.

13.12. Backup on Azure

To enable, trigger and roll back a snapshot on Azure, follow the steps outlined by Microsoft in Quickstart: Back up a virtual machine in Azure.

The VOSS database configuration ensures that VOSS Automate supports Application Consistency during snapshot restoration.

13.13. Restoring a Backup on a New Environment

13.13.1. Introduction to Restoring Backups on a New Environment

Before restoring a backup on a new environment, take note of the following:

- Every backup made on the VOSS Automate platform is encrypted using a passphrase.
- To restore a backup, you need to set the passphrase where the restore will be done.
- The passphrase is initially set on deployment of the environment and uses the platform user's password as the passphrase for backup encryption.
- The passphrase can be set manually using the **backup passphrase** command.
- Note that if a new passphrase has been set on the system, all backups made with the previous passphrase cannot be restored unless the passphrase is set back to the passphrase used to create the backup.
- Currently, two disks can be impacted: the backup drive and the dbroot drive. The backup drive size is initially 50GB and the dbroot size is initially 250GB (60GB on a standalone single-node cluster deployment).
- If the data size you restore is bigger than the size of these drives, you need to reassign these drives to add more space for the restore.
- When restoring a backup on a cluster, the latest backup from the highest weighted secondary database server must be used to ensure the entire cluster (excluding the web certificates) including the database is restored and the cluster is automatically provisioned as part of the restore.

13.13.2. Setting up the Backup Passphrase on a New Environment

To set the backup passphrase to restore on a different environment:

1. Log into the new environment. If this is a cluster deployment, log in on the DB Primary.
2. Run the **backup passphrase** command.
3. Specify the current passphrase. This is normally the password of the platform user set during the deploy of the system.
4. Enter the new passphrase twice.

13.13.3. Adding More Hard Disk Space

In a clustered environment, this procedure needs to be performed on all of the DB nodes.

1. Right click on the VM in the VMware Client and click **Edit Settings**.
2. On the Hardware tab, click **Add**.
3. Follow the wizard to add a new hard disk to the VM with the correct size.
4. This step is required for example to accommodate a large database restore. If the restore size exceeds both the backup and dbroot drives size, ensure you add two hard disks to the VM.

13.13.4. Reassign Current Drives (Backup and DBroot)

1. Once the hard disks are added, reassign the drives using the **drives reassign <disk> <mountpoint name> <size in GB>** command.
2. Use the **drives list** command to list the new drives added through VMware. For example, if the new drive is listed as `sdf`, use the reassign command as follows: **drives reassign sdf services:backups <size in GB>**.
3. Similarly, to reassign the dbroot, use the reassign command as follows: **drives reassign sde mongodb:dbroot <size in GB>**.

Note:

- The *<size in GB>* specification means the volume need not be the size of the entire disk. However, a specified size must *not* be more than 90% of the disk size (or more than <disk size less 10GB> if the disk size is 100GB or smaller).

13.13.5. Restore the Backup

A complete list of backups on a location can be displayed using **backup list <location>**. To restore on a new system, run the **backup passphrase** command and enter password used to create the backup.

1. Copy the backup to the environment with **scp**. It will be located in the `media/` folder.
2. Once the file is successfully copied, use the **backup import** command to import the backup to a location that was set up, or the default `localbackup`.
3. Once the import is complete, run the **backup list** command as for example:

```
platform@Restore:~$ backup list
localbackup:
    URI: file:///backups
    Backups:
       2019-08-13 13:08
```

4. Run the **backup restore** command as for example:

Important:

- Backups should be created and restored in a `tmux` session - see: *Using the tmux command*.
- When restoring a backup on a cluster, the latest backup from the highest weighted secondary database server must be used to ensure the entire cluster (excluding the web certificates) including the database is restored and the cluster is automatically provisioned as part of the restore.

```
platform@Restore:~$  backup restore localbackup 2019-08-13 13:08
Services will be restarted during the restore. Do you wish to continue? y
Check if restore can continue (step 1/4)
Enough space on /opt/platform/apps/mongodb/dbroot: total 93MB / 224GB
Enough space on /backups/appdata: free 934MB / 232GB
Running pre-restore scripts (step 2/4)
Stopping Application while performing database restore
Running backup restore (step 3/4)
System restore starting from \
  file:///backups/058bccead2588a6f11f1dd86678bab68de48691d (1565701713)
Local and Remote metadata are synchronized, no sync needed.
Last full backup date: Tue Aug 13 13:08:33 2019
/backups/appdata/
completed
Running post-restore scripts (step 4/4)
Starting Application after performing database restore
Restarting services

Application processes stopped.

Reconfiguring applications....

If this includes a database restore, it may take some time to sync
Please run 'database config' to check when all nodes are done

Restored successfully
You have new mail in /var/mail/platform
```

13.13.6. Example of a Successful Restore

```
platform@Restore:~$ backup restore localbackup 2015-02-26 00:22
Services will be restarted during the restore. Do you wish to continue? y
Application voss-deviceapi processes stopped.
Stopping Application while performing database restore


--- Restore, ip=172.29.41.240, role=webproxy,application,database, loc=jhb


Application nginx processes stopped.
System restore starting from
 file:///backups/93d19980b574ed743d9b000a7595e42cad6a6d6b (1424910132)
Local and Remote metadata are synchronized, no sync needed.
Last full backup date: Thu Feb 26 00:22:12 2015
```

(continues on next page)

(continued from previous page)

```
Successfully restored to /backups/appdata/restore_temp_1427441507,
 moving to /backups/appdata
Removing temporary files in /backups/appdata/restore_temp_1427441507
local\|admin
Dropping database PLATFORM before restoring
MongoDB shell version: 2.6.1
connecting to: 127.0.0.1:27020/PLATFORM
[object Object]
Repairing database PLATFORM before restoring
MongoDB shell version: 2.6.1
connecting to: 127.0.0.1:27020/PLATFORM
[object Object]
Dropping database VOSS_FILES before restoring
MongoDB shell version: 2.6.1
connecting to: 127.0.0.1:27020/VOSS_FILES
[object Object]
Repairing database VOSS_FILES before restoring
MongoDB shell version: 2.6.1
connecting to: 127.0.0.1:27020/VOSS_FILES
[object Object]
Dropping database VOSS before restoring
MongoDB shell version: 2.6.1
connecting to: 127.0.0.1:27020/VOSS
[object Object]
Repairing database VOSS before restoring
MongoDB shell version: 2.6.1
connecting to: 127.0.0.1:27020/VOSS
[object Object]
Trying with oplogReplay
restore successfull
{'172.29.41.240': (200, '\n')}
Starting Application after performing database restore


--- Restore, ip=172.29.41.240, role=webproxy,application,database, loc=jhb


Application services:firewall processes stopped.
Application nginx processes started.
Restarting services

Application processes stopped.

Application  processes started.

System settings have changed, please reboot using 'system reboot'

Restored successfully
You have new mail in /var/mail/platform
```

14. System Security

14.1. Introduction to system security

The platform is installed without antivirus software or an index of whitelisted applications. The functionality provided by these anti-malware measures is implemented by means of an extensive number of measures to lock down and harden the operating system, platform system and network.

Security covers areas such as application and operating system security updates, operating system hardening, file and application encryption, jailed environments for applications, firewalls, and user security. This locked down system ensures that platform users can't install their own packages, binaries, or applications on the system to perform any malicious actions or allow the exploitation of vulnerabilities (such as Meltdown/Spectre).

Note: Since the application runs in a virtual environment it is important that the underlying VM infrastructure stays up to date to be protected against any vulnerabilities that may compromise the Virtual Machine on the infrastructure.

14.2. Configuration Encrypted

In order to help protect customer data and service stability the system configuration files are frequently recreated by the platform. This means that even malicious tampering to the platform will generally be undone by a simple restart. The configuration data is stored in the platform's internal files. These files are encrypted using a strong AES encryption layer to make them tamper-proof. They are never decrypted on disk, instead the applications which manage them will decrypt them in memory, read and make modifications as needed and then re-encrypt the data before writing them back to disk.

In this way the risk of tampering or data theft through the configuration system is greatly minimized.

14.3. Backup Encrypted

System backups include copies of the full system configuration as well as the full contents of the database. Thus theft of a backup would effectively constitute theft of all customer data stored on the platform. To mitigate this risk backups are encrypted using a strong 2048-bit in-line GPG encryption.

The encryption key for this is auto-generated by the platform based on a unique machine UUID. While it's possible for support to recover backups from a different machine this process is deliberately hard and only available to official technical support representatives. Backups on shared locations are separated on a per-source-machine basis to prevent conflicts.

14.4. Application Install Files Encrypted

In order to protect the trustworthiness of applications shipped for the platform, all application installers are encrypted files. The strong 2048-bit key needed to decrypt these are shipped with the platform and is different from the per-machine unique keys used for other encryption tasks. This key will only decrypt applications encrypted specifically with the unique key owned by VOSS. The system will refuse to install any application that is not encrypted or encrypted with a different key.

This ensures that only valid, untampered copies of genuine VOSS-released applications can be installed on the system.

14.5. File Integrity

System installation and upgrade binaries, as well as configuration files, are regularly checked for file integrity against a file hash. The types of files and directories to check, is configured.

A scheduled task is configured to initialize and to carry out the regular validation. If audit logging is enabled on a system, this initialization will show in the audit logs as the EventType `FileDetection` and Audit Details as `File checksum initialized`.

The Command Line Interface (CLI) diagnostic command **diag fileguard** is also available to carry out a manual check for changes to these files of since the previous check. File tamper detection and integrity monitoring is also carried out. Note that the file check validates all system files and is a time consuming task.

If any files have been changed, removed or added to the configured types and directories, these will be listed in the command output, together with the type of changes.

Also refer to the topics on Diagnostic Tools and Audit Log Format and Details.

14.6. Protected Application Environments (Jails)

VOSS Automate runs the service providing applications in secured jail environments. This has significant value for the security and reliability of the system. It prevents applications from cross-interfering which makes the system more stable and reliable. In terms of security it effectively confines all services to dedicated and separate mini file systems with predictable content. In the event that an attacker were to gain access to the system through a vulnerability in a service he would therefore not gain access to the platform but only to the small confined jail in which the service was running. In that environment only the jail itself is vulnerable and this can be very easily restored if damaged. The underlying system cannot be accessed from the jailed environment.

The VOSS system does not allow direct root access over ssh. If root access is required for debugging purposes, there is a tool called NRS. This tool requires the user to log in as a user with install privileges, who has to run **app install nrs**. The tool generates a key, which can only be deciphered by VOSS. VOSS uses this key to then gain root access in order to proceed with debugging.

14.7. Restricted User Shell

The platform attempts to reduce the risk of unintentional harm to the operation of the software by restricting the actions users can take. This is done using a specially configured setup of the well-known and actively maintained rbash shell.

The shell actively prevents the following:

- Users cannot set environment variables or alter their command path.
- Users cannot change the current directory.
- Users cannot specify a path to a command to run.

The commands users thus are able to run is only what is allowed by the platform setup. The vast majority of these commands use a common execution interface designed to allow only enough privileges to perform the system administration tasks they are created for. The exact list of commands a user can run is determined by his specific privileges and the specific setup of the machine on which he is working (different applications can add their own additional commands). This list is displayed on login and can be redisplayed with the **help** command.

14.8. User Security and Security Policy Management

Upon installation, user passwords are restricted as follows:

- Password length : 8
- Minimum number of days between password change : 1
- Maximum number of days between password change : 60
- Number of days of warning before password expires : 14
- Number number of days between password change: 10

User password and account security settings and policy details can also be configured. Commands are available to manage:

- password length
- automatic account locking after inactivity
- number of days between password change - valid values are from 5 to 20

The following commands are available to show the current length and set the default minimum password length:

- **user password length**
- **user password length <min_length>**

The value of <min_length> can be set from 8 to 127 characters. By default, it is 8 characters. For other password rules, refer to Password Strength Rules. The setting also applies to backup passphrases.

By default, any account that is created has the inactive lock set to 35 days.

To set the number of days between user password expiration:

user password expiry <username> [60-365,never]

Valid values for days is from 60 to 365. If never is typed in, the password does not expire and when typing **user passwordinfo <username>**, the Maximum number of days between password change value shows as -1.

The password re-use frequency default is 10 passwords, which means that the last 10 passwords cannot be re-used, unless this is set, using:

user password history <number-of-passwords>

where 5 <= <number-of-passwords> <= 20.

To see the current <number-of-passwords> re-use frequency:

user password history

For example:

```
platform@VOSS:~$ user password history
The default password history is set to 10.
```

The commands below are available to carry out these tasks and to manage users.

- **user passwordinfo <username>**

  Show details such as password expiry in days for a user, for example:

```
$ user passwordinfo joebrown
Last password change                                  : Nov 30, 2015
Password expires                                      : Feb 28, 2016
Password inactive                                     : Apr 03, 2016
Account expires                                       : never
Minimum number of days between password change        : 1
Maximum number of days between password change        : 60
Number of days of warning before password expires     : 14
```

- **user inactivelock <days> <user>**

  Set the number of days of inactivity before a user account is locked, for example:

```
$ user inactivelock 35 joebrown
A 35 day inactive logon policy has been set for user: joebrown
```

- **user lock <user>**

  Manually lock a user account, for example:

```
$ user lock joebrown
passwd: password expiry information changed.
```

- **user unlock <user>**

  Manually unlock a user account, for example:

```
$ user unlock joebrown
passwd: password expiry information changed.
```

  To unlock users who exceeded allowed number of failed login attempts:

```
$ system ssh fail_limit reset joebrown
$ user unlock joebrown
passwd: password expiry information changed.
```

- **user password view_lock <user>**

  The command output is different in accordance with the event that locked the user account:

  Not a manual user lock:

```
$ user password view_lock joebrown
There is no password lock applied for user joebrown.
 Please run 'system ssh fail_limit view joebrown' to
 ensure the account is not locked because the user has
 reached the maximum number of failed attempts .
```

  Manual user lock:

```
$ user password view_lock joebrown
The password for user: joebrown has been locked.
 Please run 'user unlock joebrown' and
 'system ssh fail_limit reset joebrown' to ensure
 you unlock and reset lock limits for this user account
```

- **user lastlogon <username>**

  Show details of the last logon for:

  – a user who has logged in before:

```
$ user lastlogon joebrown
joebrown 172.29.90.74 Thu Dec 3 11:04:54.
```

  – a user who has not logged in before:

```
$ user lastlogon joebrown
joebrown logged in***
```

Use the **user help** command to see the general user management options such as user list, add, grant or revoke rights and remove users.

The command **user list** provides rights and security policy details of *all* users, while **user list <username>** provides details for a single user. For example:

```
$ user list
  user:
      joebrown:
          rights:
              mail
              app
      janedoe:
          rights: value not set
      billsmith:
          rights: value not set


  security_policy:
      user:
          platform:
              auto_inactive_account_lockout: 35
          joebrown:
              account_locked: No
              auto_inactive_account_lockout: 35
          janedoe:
              auto_inactive_account_lockout: 35
          billsmith:
              account_locked: No
```

In addition, a system wide account security setting can be configured and displayed. The setting will then apply to all *new* users and override the default inactive lock setting of 35 days.

The following commands are available:

- **system inactivelock**: show the current system wide inactive lock default:

```
$ system inactivelock
Newly added users will have their inactivity lock set to 35 days.
```

- **system inactivelock <num of days>**: set the system wide inactive lock default for all new user accounts, in other words, for users created *after* the setting of the system wide inactive lock:

```
$ system inactivelock 35
Newly added users will have their inactivity lock set to 35 days.
```

14.9. Creating Additional Users

During installation a user called ‘platform’ is created which has full access to all allowed commands within the restricted environment. This user (and others with the appropriate rights) can then create additional users who are further restricted to only be able to run certain commands. For example a user could be created who can only run diagnostic and logging commands - able to monitor the health of a system but required to escalate any actions.

Users are created, managed and deleted through the user command. To create a new user use:

user add <username>

Note: The `<username>` text needs to follow the rules below:

- starts with letter (a-z/A-Z)
- followed by one or more of:
  - letters (a-z/A-Z)
  - digits (0-9)
  - full-stop (.)
  - underscore (_)
  - dash (-)

The system will create a Unix user with the name specified and set up to use a restricted shell identical to the platform user. Initially this user's password is set to match the username but it must be changed on first login. New users start out with no rights and can only run the very basic system commands provided to all users (such as **ls**).

For SFTP only users, see: *Creating and Managing SFTP Users*.

14.10. Creating and Managing SFTP Users

Administrators can add and manage users who have SFTP only access. For platform user management, see: *Creating Additional Users*.

To create a new SFTP only user, use the command:

user sftp add <username>

Note: The `<username>` text needs to follow the rules below:

- starts with letter (a-z/A-Z)
- followed by one or more of:
  - letters (a-z/A-Z)
  - digits (0-9)
  - full-stop (.)
  - underscore (_)
  - dash (-)

Add a username and password. See *Password Strength Rules*.

The system will create a user with the provided name and password provided, with the following restrictions:

- the system can *only* be accessed by SFTP
- user access is restricted to the platform `home/` directory only
- the SFTP user will have a SSH key attached.

- only the administrator can change the SFTP only user password

To attach a SSH key to the SFTP user:

1. Copy the SSH public key for the user onto the system
2. Run **user addkey <username> <keyfile>** to attach the key to the user

See also: *Adding a Key for Automatic User Login*.

To change the SFTP user password:

user sftp password <username>

To remove the SFTP user:

user del <username>

SFTP users are listed under the `sftp-only-users` group when running the **user list** command.

An SFTP only user can also be added without a password. This option can be used for an SFTP implementation that uses a key only (no password):

user sftp add_nopass

14.11. Granting and revoking user rights

Once a user is added the user needs to be granted access to run commands. The user's command menu will only display those commands to which access have been granted.

To grant access to a command use the 'user grant' command as follows:

user grant <username> <command> [options]

Only one command can be granted at a time, however these can be complex. The more detailed the command, the more fine-grained the privilege becomes. This is best explained by example.

Running the following command:

user grant peter app

Will allow the user peter to execute any command within the 'app' series of commands. However it could be restricted further by instead running a command like:

user grant peter app list

With this version peter will see the **app** command on his menu, but its help will only display 'list' as a sub-command - peter can thus see the list of apps but cannot perform more potentially risky tasks such as installing or restarting applications.

This can be expanded to other subsets by simply running additional grants:

user grant peter app start

Would now allow peter to both see the list of applications or restart applications that failed, however he will not be able to do other app related tasks such as installations. The **grant** command effectively verifies that the start of a command by a user matches one of the privileges granted to that user - so peter will be able to add options to any command he is granted access to.

In order to restrict commands - be sure to determine whether any options should be allowed and if not, only grant access to the specific parameters you wish peter to be able to execute. For example if peter is your database administrator for example you may wish to use:

user grant peter app start mongodb

Instead of giving access to all **app start** commands.

Should you wish to revoke a command privilege from a user you can do this using the following command:

user revoke <username> <full command>

The command being revoked must match exactly one of the commands previously granted to a user. To review the current privileges of a user use:

user list <username>

Which will display the user's entire list of granted commands in full. You can also just run

user list

Without an option to list all users created on your system and their privileges.

14.12. Password Strength Rules

The platform user and any CLI users created are held to strong password rules to help reduce the risk of system penetration. These rules are enforced whenever passwords are changed or set. In order to meet system password strength rules a user's password must:

- By default be at least 8 characters long. This can be modified with the **user password length <min_length>** command. See *User Security and Security Policy Management*.
- Contain at least one capital and one non-capital letter.
- Contain at least one number.
- Contain at least one special character.
- A password change should differ by at least 8 characters from the old password. In other words, if an old password was 8 characters, then *all* new password characters should differ.

14.13. SSH Login Fail Limit

An administrator can view and modify the number of login attempts for a user.

- The default number of failed login attempts for a user is 10 before the account is locked.
- The default duration that an account will be locked, is 15 minutes (900 seconds).
- **system ssh fail_limit set <number>**

  Set the number of failed login attempts for all user accounts on this system before account lockout occurs. For example:

```
$ system ssh fail_limit set 3
You are about to set a limit for failed login attempts.
   This limit will apply to all user accounts on this system.
   Do you wish to continue? Y
```

- **system ssh fail_limit view**

  Show the current number of failed logins allowed

```
$ system ssh fail_limit view
SSH session fail_limit is set to 10
```

- **system ssh fail_limit view <username>**

  View the current status of a user's failed login attempts. Examples:

```
$ system ssh fail_limit view joebrown
Login     Failures  Latest failure     From
joebrown     0

$ system ssh fail_limit view joebrown
Login     Failures  Latest failure     From
joebrown      1    12/04/15 10:38:00   192.168.0.90
```

  If no users are defined, the message `No users created` is shown.

- **system ssh fail_limit status**

  View the current status of all users failed login attempts

  Examples:

```
$ system ssh fail_limit status
Login     Failures  Latest failure     From
joebrown     0
```

- **system ssh fail_limit reset <username>**

  Reset the limit back to 0 on a locked out account. This will allow a user to log back in to the system without resetting a password after a lockout occurs. For example:

```
$ system ssh fail_limit view joebrown
Login     Failures  Latest failure     From
joebrown      3    12/04/15 10:38:00   192.168.0.90

$ system ssh fail_limit reset joebrown
You are about to reset the account lockout information for
  user: joebrown. This will allow this user to log back in to
  the system.  Do you wish to continue? y

$ system ssh fail_limit view joebrown
Login     Failures  Latest failure     From
joebrown     0
```

- **system ssh fail_limit unlock_time <seconds>**

  Enable the unlock time and set the duration in seconds that an account will be locked for after it has been locked.

  To disable the unlock time setting, use the command with the parameter value as zero:

  system ssh fail_limit unlock_time 0

  The example output below shows the command response for parameter values:

```
$ system ssh fail_limit unlock_time 60
SSH session unlock time has been set.

$ system ssh fail_limit unlock_time 0
SSH session unlock time has been disabled.

$ system ssh fail_limit unlock_time -1
Please enter a valid number for unlock time.
```

- **system ssh fail_limit unlock_time**

  Display the status of the unlock time setting.

  For example:

```
$ system ssh fail_limit unlock_time
SSH session unlock time is not set.

$ system ssh fail_limit unlock_time 60
SSH session unlock time has been set.

$ system ssh fail_limit unlock_time
SSH session unlock time has been set to 60 seconds.
```

14.14. SSH Session Limit

An administrator can set and modify the number of SSH sessions allowed:

- system-wide (default is 10 if not set)
- for a user (default set to the system-wide setting)

Note: The default number of SSH sessions allowed *per IP source* is limited to 10. This means that if a user SSH session limit is higher than this limit, the user session origin needs to be from a different IP source.

Best practice is to set the system-wide SSH session limit first as this will be the default for any new users created on the system. Also note that the per user SSH session limit cannot be set higher than the system-wide SSH session limit.

To see the current system-wide SSH limit, use:

system ssh_session_limit

To set the system-wide SSH limit:

system ssh_session_limit set <number>

This system wide value will restrict the *per user* limit that can be set.

When a user is added and no session limit is added, the user's number of SSH sessions is set to the default system wide default limit of 10. It is recommended to also set the user's session limit.

To set the SSH session limit for a user:

user credential_policy session_limit <username> <number>

where <number> cannot be larger than the system wide session limit, if it has been set.

The current SSH session limit for users can be seen by using the **user list** command, for example:

```
platform@drp32:~$ user credential_policy session_limit joebrown 5
platform@drp32:~$ user list
   user:
     joebrown:
       rights: value not set

   security_policy:
     joebrown:
       account_locked: No
       auto_inactive_account_lockout: 35
       ssh_connection_limit: 5

platform@drp32:~$
```

If a user has sessions open while the session limit is set, the limit in affect when new sessions are opened.

14.15. SSH key management

SSH authentication requires maintaining the system SSH keys. This can be done as follows:

- **keys create** creates a local SSH keyset
- **keys add <host>** adds the remote host to the known hosts list allowing outgoing connections
- **keys send <user>@<host>** will send the public key from the local SSH keyset to the remote server, thereby enabling remote SSH authentication.

For example, if you wish to perform a backup to a remote host, first create a local key if necessary with **keys create**. Allow communication with the host using **keys add <host>**. Send the key to the remote host with **keys send <user>@<host>**.

If you select ssh key-based authentication that was set up without a password, you will be prompted to input the password when adding a sftp backup host:

```
platform@VOSS-UN-6:~$ backup add remote sftp://dan@182.59.31.201/sftp
No password found. Do you want to use sshkeys? [y/N]: y
```

Important: If the username or password contains any of the following characters:

```
;|\\$&`!
```

then use the backup setup command interactively by choosing “No” at the prompts if SSH Keys are being used and then add the username and password directly.

The certificates are independent of web servers/proxies.

For more details on SSH key-based authentication, refer to OpenSSH documentation.

14.16. SSH Algorithm Management

SSH algorithms on the VOSS Automate platform can be viewed, enabled, disabled or reset to the default list.

The available commands are:

- **system ssh algorithm default** - Reset enabled and disabled algorithms to their default.
- **system ssh algorithm list < all | algorithm-type >** - Display *all* or any of *<cipher|mac|kex|key>* SSH algorithms. The list will also show `enabled` and `disabled` algorithm types.
- **system ssh algorithm disable <algorithm-type> <algorithm-name>** - Disable algorithms of a specific type *<cipher|mac|kex|key>*, by specifying a space separated list of algorithm names.

  Note that not all algorithms can be disabled.

  For example, to disable two of the `cipher` algorithms, the command would be:

  system ssh algorithm disable cipher aes128-ctr aes192-ctr
- **system ssh algorithm enable <algorithm-type> <algorithm-name>** - Enable algorithms of a specific type *<cipher|mac|kex|key>*, by specifying a space separated list of algorithm names.

Command example to view status of *all* algorithms types:

```
platform@VOSS:~$ system ssh algorithm list all
    cipher:
        enabled:
            aes128-ctr
            aes192-ctr
            aes256-ctr
    kex:
        enabled:
            diffie-hellman-group1-sha1
            diffie-hellman-group14-sha1
            diffie-hellman-group-exchange-sha1
            diffie-hellman-group-exchange-sha256
            curve25519-sha256@libssh.org
    key:
        enabled:
            ssh-ed25519
            ssh-ed25519-cert-v01@openssh.com
            ssh-rsa
            ssh-dss
            ecdsa-sha2-nistp256
            ecdsa-sha2-nistp384
            ecdsa-sha2-nistp521
            ssh-rsa-cert-v01@openssh.com
            ssh-dss-cert-v01@openssh.com
            ecdsa-sha2-nistp256-cert-v01@openssh.com
            ecdsa-sha2-nistp384-cert-v01@openssh.com
            ecdsa-sha2-nistp521-cert-v01@openssh.com
    mac:
        enabled:
            hmac-sha1
            hmac-sha2-256
```

(continues on next page)

(continued from previous page)

```
            hmac-sha2-512
            hmac-ripemd160
            hmac-ripemd160@openssh.com
            umac-128@openssh.com
            hmac-sha1-etm@openssh.com
            hmac-sha2-256-etm@openssh.com
            hmac-sha2-512-etm@openssh.com
            hmac-ripemd160-etm@openssh.com
            umac-128-etm@openssh.com
```

14.17. Adding a Key for Automatic User Login

To automate tasks such as backups from remote hosts, it may be necessary to allow for the SSH login on VOSS Automate by a user without a password.

This login requires the addition of a public SSH key for the user. The **user addkey** command is available to add a keyfile for a user. The command to run, is of the format:

user addkey <username> <keyfile>

Note that:

- The user who runs this command, should have the `<keyfile>` available on their local directory. The public keyfile should therefore be copied to VOSS Automate.
- The user (`<username>`) for whom the key is to be added, should exist. If the user does not exist on VOSS Automate, a message shows to indicate this.

If the command is successful, the following message is shown:

```
User key added. You should try to ssh now
```

14.18. Prevention of DOS Attacks

The following list shows measures implemented in VOSS Automate to protect the system against Denial of Service (DOS) attacks:

- Firewall protection:
  - TCP flood protection against:
    - the SSH port
    - web server ports
  - SYN flood protection
- Configurable session limits for the VOSS Automate platform SSH access is **Sessions per user** and **Sessions per application**. An administrator can set and modify the number of SSH sessions allowed:
  - system-wide (default is 10 if not set)
  - for a user (default is 10 if not set)

See SSH Session Limit for detailed information.

- The usage of ports, protocols, and services are registered with the DoD PPS Database
- An automated, continuous on-line monitoring of the system is implemented, with:
  - Audit trail creation capability in a format that a log viewing application can immediately alert personnel of any unusual or inappropriate activity with potential Information Assurance (IA) implications.
  - A command line command that a user can automatically disable the system if serious IA violations are detected.
- Applications are monitored and notifications sent when resource conditions reach a predefined threshold indicating there may be attack occurring, for example through SNMP traps and triggers.
- High disk utilization is managed due to error notifications. For log files, disk utilization is managed by:
  - daily log rotation
  - 4 weeks of backlogs
  - the creation of new (empty) log files after rotating old ones
  - log file compression
  - a logging restriction of 20 messages per minute
- A continuous cycle of updating packages during releases is in place with notifications during updates. Commands to carry out a security check or update can be run at any time.

14.19. Memory Dumps and Security

Memory dumps in VOSS Automate are restricted to Cisco Administrators. Attackers will therefore not be able to gain access to sensitive data which may appear unencrypted in memory.

14.20. Manage Read-Only Database Users

Remote read-only database access for users can be managed. A username, source IP address and password are required as parameters.

Important: Since the system firewall service is restarted when adding or removing database users, this may affect system operation.

It is therefore strongly recommended that the task be carried out during a maintenance window.

Note:

- When adding a user, a prompt for a password is given. The password rules are:
  - 8 or more characters minimum
  - contains at least 1 uppercase
  - contains at least 1 lower case
  - contains a least 1 digit

  - contains at least 1 punctuation character
- Adding and deleting users require a service restart.

The following commands and parameters are available:

- **database user add <ip> <username>**
- **database user del <ip> <username>**
- **database user list**

Example console output:

```
platform@dev:~$ database user list
No users defined
platform@dev:~$ database user add 192.79.22.52 alex
You are about to restart services. Do you wish to continue? yes
New user password:
Please verify password:
platform@dev:~$ database user list
IP  users
192.79.22.52  alex
platform@dev:~$ database user del 192.79.22.52 alex
You are about to restart services. Do you wish to continue? yes
platform@dev:~$ database user list
No users defined
```

14.21. System Intrusion Detection

The VOSS Automate platform provides an optional service that can be enabled in order to detect and prevent SYN flood denial-of-service network attacks.

When the service is active, source IP addresses of such attacks are detected and communication from these addresses is blocked for *10 minutes*. However, if the attacks persist after this period, the addresses are re-blocked.

Log files are also created to provide service details

Important: The service commands only apply to the node on which the commands are executed. In order to run commands across a cluster, the `cluster` command prefix is required, for example, to enable the service cluster-wide:

```
cluster run all system intrusion-detection profile enable synflood
```

See: *Remote Execution in Clusters*

14.21.1. Start the service

To check the logging level and if the service is running:

```
platform@VOSS:~$ system intrusion-detection logging get-level
    level: NOTICE

platform@VOSS:~$ system intrusion-detection profile show
    synflood:
        enabled: false
```

When running `app status`, the `intrusion-detection` service will show as `running`:

```
services v25.2 (2025-07-01 14:11)
   |-logs                running
   |-scheduler           running
   |-intrusion-detection running
```

A profile must be enabled:

`system intrusion-detection profile enable synflood`

Refer to the command details below.

14.21.2. Manage the service

The following commands are available to manage this service:

- `system intrusion-detection logging get-level`

  Get the log level for the intrusion detection system. Refer to the log level values below.

- `system intrusion-detection logging set-level <level>`

  Set the log level for the intrusion detection system. Changing the level requires a service restart. A message shows when the log level has been set. Permitted values are (case-insensitive):

  DEBUG, INFO, NOTICE, WARNING, ERROR, CRITICAL

Level	Description
DEBUG	Verbose debug messages for troubleshooting.
INFO	Standard informational messages about regular activity and significant events.
NOTICE	Normal but notable messages. This is the *default level*.
WARNING	Warning messages about potential issues or warnings.
ERROR	Only error events get logged.
CRITICAL	Only critical messages get logged.

- `system intrusion-detection profile disable <profile>`

  Disable intrusion detection for a profile, for example, `synflood`.

- `system intrusion-detection profile enable <profile>`

  Enable intrusion detection for a profile e.g. `synflood`.

  Note: Currently, only the `synflood` profile is available.

- `system intrusion-detection profile show`

  Display the configuration of intrusion detection profiles

14.21.3. Stop the service

Note: Blocked IP addresses become available after service restart, but will be blocked if SYN flood attacks presist.

14.21.4. Log files

Log files can be inspected using the `log view <log_file>` command, for example: *log view platform/fail2ban.log`*.

- `process/services.intrusion-detection.log`

  Shows service logs with process information, for example: start/stop/status/log-level updates

- `platform/fail2ban.log`

  Internal fail2ban service logs, providing details on each logging event with: a timestamp, log level, process ID, component, and log message.

  Example output:

```
2025-07-01 13:50:01,133 fail2ban.actions        [1239]: NOTICE  [ssh] Unban␣
↪172.90.123.113
2025-07-01 14:15:00,952 fail2ban.filter         [1239]: INFO    [ssh] Found␣
↪67.89.123.45
2025-07-01 17:00:13,171 fail2ban.actions        [1239]: NOTICE  [ssh] Ban 90.
↪123.171.90
```

15. Network Security

15.1. Network communications between nodes within the cluster

The cluster contains multiple nodes, which can be contained in separate, firewalled networks. Network ports must be opened on firewalls to allow inter-node communication.

All communication between nodes is encrypted.

The following details are all based on the default settings. These can vary depending on the application setup and network design (such as NAT) of the solution, so may need adjustment accordingly. Where a dependant is noted, this is fully dependant on the configuration with no default.

These communications are all related to communications between application nodes within the cluster. There are a few different deployment models so the details below cover the different models and relevant ports. Review and implement according to the deployment model in use.

Note: *Standalone* is only a single node so this section is not relevant for that deployment model.

Proxy to proxy node

This is relevant if the proxy node is present in the system.

Communication	Protocol	Port
Cluster Communications	HTTPS	TCP 8443 bi-directional

Proxy to unified/application node

This is relevant if the proxy node is present in the system.

Communication	Protocol	Port
User access	HTTPS	TCP 443
Cluster Communications	HTTPS	TCP 8443 bi-directional

Unified node to unified node

This is relevant to the communications between the unified nodes (application and database combined). If the application and database nodes are split, refer to the relevant application and database node details below. Database arbiters run on port 27030.

Communication	Protocol	Port
Database access	database	TCP 27020 and 27030 bi-directional
Cluster Communications	HTTPS	TCP 8443

Application node to application node

This is relevant to the communications between application nodes in the system, only where the database node is separate from the application node (that is, not unified node).

Communication	Protocol	Port
Cluster communications	HTTPS	TCP 8443 bi-directional

Application node to database node

This is relevant to the communications between the application node and the database node, if the database node is separate from the application node. Database arbiters run on port 27030.

Communication	Protocol	Port
Database access	database	TCP 27020, 27030, 5432 bi-directional
Cluster Communications	HTTPS	TCP 8443

Database node to database node

This is relevant to the communications between the application node and the database node, if the database node is separate from the application node. Database arbiters run on port 27030.

Communication	Protocol	Port
Database access	database	TCP 27020 and 27030 bi-directional
Cluster Communications	HTTPS	TCP 8443

15.2. Network communications external to the cluster

Details in this section are all based on the default settings, which can vary depending on the application setup and network design (such as NAT) of the solution. Adjust accordingly. Where a dependant is noted, this is fully dependant on the configuration with no default.

These communications are all related to communications with devices external to the cluster. Details are provided for the following:

- Outbound communications to devices from the application/unified nodes
- Outbound to external systems from the proxy node
- Outbound to external systems from all nodes
- Inbound communications from external systems to the proxy node
- Inbound communications to all nodes
- On-line help links to external documentation

Outbound communications to devices from the application/unified nodes

Communication	Protocol	Port
Cisco Unified Communications Manager (UCM)	HTTPS	TCP 8443
Cisco Unity Connection (CUXN)	HTTPS	TCP 443
Webex	HTTPS	TCP 443
LDAP directory	LDAP	TCP/UDP 389 and/or 636(TLS/SSL)
MS PowerShell Proxy Node	HTTPS	TCP 5986
Microsoft 365 (Graph API)	HTTPS	TCP 443
Zoom	HTTPS	TCP 443

Outbound to external systems from the proxy node

Communication	Protocol	Network Protocol and Port
API Sync and Async responses	HTTPS	TCP 443
Northbound Notification messages	HTTPS	dependant
Microsoft Teams / Microsoft Exchange	HTTPS	443
VOSS Cloud Licensing Service	HTTP HTTPS	80 443

Outbound to external systems from all nodes

Communication	Protocol	Port
SNMP	SNMP	TCP/UDP 162
SFTP as required for backup destinations	SFTP	TCP 22
NTP	NTP	UDP 123

Inbound communications from external systems to the proxy node

Communication	Protocol	Port
Web Access	HTTPS	TCP 443
API Request	HTTPS	TCP 443

Inbound communications to all nodes

Communication	Protocol	Port
SSH and SFTP for management and files transfers	SFTP/SSH	TCP/UDP 22

On-line Help links to external documentation

To have access to the online help website URL, you may need to request that your network administrator provides access to the website.

15.3. Dynamic Firewall

The most important part of the network security model is the system firewall.

The platform uses a dynamic firewall which does not open a fixed set of ports but adapts to the applications installed, only allowing such traffic as the specific set of running services require.

If an application is stopped, it's ports are automatically closed. This creates a default block list firewall which pinholes only those ports required for the operation of the specific setup in use.

The firewall is one of the very first services the platform brings up and among the very last it shuts down in order maximize the network security.

Where possible, the firewall will also ratelimit connections to services to prevent abuse (see the section: Prevention of DOS attacks for more details).

15.4. Web Certificate Setup Options

The platform installs a self-signed certificate for the web-frontend by default. This provides encryption of the web-traffic but does not provide users with valid authentication that the server is correct or protect against man-in-the-middle attacks.

Two types of certificate setups are supported:

- VOSS Automate certificate setup

  We strongly advise customers to obtain a trusted CA-signed certificate and install it on the server. A 4096 bit RSA certificate is generated on VOSS Automate systems.

  Once a signed, trusted certificate is obtained from the CA, copy it to the platform using **scp** and then install the file into the server using:

  web cert add <filename>

  For details, see: *VOSS Automate Setup a Web Certificate*

- Own private certificate and generated Subject Alternative Name (SAN) certificate setup

  Customers can upload their own private certificate and generated SAN certificates, in other words it is not necessary to run **web cert gen_csr** on the platform CLI. One certificate can therefore be uploaded on all nodes. Note that customers are then responsible for the security of their private keys.

  For details, see: *Own Web Certificate Setup*.

  The file to upload should be in a PEM format. PEM certificates typically have extensions like `.pem`, `.crt`, `.cer` and `.key`.

  The PEM file must have the correct form of line termination: a single "Line Feed" character. If your PEM file was saved on MS Windows, be sure to remove the ^M characters from the file, for example in a Linux console with:

```
$ tr -d '\r' < original.pem > fixed.pem
```

  In the file, the SAN certificate composition has the private key first and then the certificate and the private key should be *unencrypted* (i.e. the key header text would then not show "`BEGIN ENCRYPTED PRIVATE KEY`").

  For example:

```
-----BEGIN PRIVATE KEY-----
MIIEvAIBADANBgkqhkiG9w0BAQEFAASCBKYwggSiAgEAAoIBAQDNV1pXvjIiiWuJIABW
[...]
IeJnlBPwDJX6Yo9Q==
-----END PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
MIIEbTCCAlUCAgPoMA0GCSqGSIb3DQEBCwUAMIGbMQswCQYDVQQGEwJaQTELM
[...]
ulfj0D54fozATLIdMZSrmImk8CfkDPkmWbIKRce729DTQwHrMG/OolZC2
-----END CERTIFICATE-----
```

  Copy the certificate file to the platform `media/` directory using **scp** and then install the file using:

  web cert add_san <filename>

  For example:

```
platform@host:~$ web cert add_san media/cert.pem
Updating the certificate requires the web server to be restarted.
Do you wish to continue? yes
Restarting nginx
platform@host:~$
```

Note:

- SSO certificate management is carried out on the GUI. Refer to the GUI documentation for details.
- VOSS Automate supports wildcards for Common names (CN) in the web browser certificate.
- Only one certificate file can be installed on the platform. For more details on NGINX compatible certificates see the relevant nginx documentation here: [http://nginx.org/en/docs/http/ngx_http_ssl_module.html]
- Please note the importance of ensuring that SSL certificates generated match the assigned network name of the platform.

15.5. Supported SSL Ciphers

The list of supported SSL ciphers are as follows. This list may change as ciphers are added or found to be insecure:

- ADH-AES128-GCM-SHA256
- ADH-AES128-SHA
- ADH-AES128-SHA256
- ADH-AES256-GCM-SHA384
- ADH-AES256-SHA
- ADH-AES256-SHA256
- ADH-CAMELLIA128-SHA
- ADH-CAMELLIA128-SHA256
- ADH-CAMELLIA256-SHA
- ADH-CAMELLIA256-SHA256
- ADH-SEED-SHA
- AECDH-AES128-SHA
- AECDH-AES256-SHA
- AECDH-NULL-SHA
- AES128-CCM
- AES128-CCM8
- AES128-GCM-SHA256
- AES128-SHA

- AES128-SHA256
- AES256-CCM
- AES256-CCM8
- AES256-GCM-SHA384
- AES256-SHA
- AES256-SHA256
- ARIA128-GCM-SHA256
- ARIA256-GCM-SHA384
- CAMELLIA128-SHA
- CAMELLIA128-SHA256
- CAMELLIA256-SHA
- CAMELLIA256-SHA256
- DHE-DSS-AES128-GCM-SHA256
- DHE-DSS-AES128-SHA
- DHE-DSS-AES128-SHA256
- DHE-DSS-AES256-GCM-SHA384
- DHE-DSS-AES256-SHA
- DHE-DSS-AES256-SHA256
- DHE-DSS-ARIA128-GCM-SHA256
- DHE-DSS-ARIA256-GCM-SHA384
- DHE-DSS-CAMELLIA128-SHA
- DHE-DSS-CAMELLIA128-SHA256
- DHE-DSS-CAMELLIA256-SHA
- DHE-DSS-CAMELLIA256-SHA256
- DHE-DSS-SEED-SHA
- DHE-PSK-AES128-CBC-SHA
- DHE-PSK-AES128-CBC-SHA256
- DHE-PSK-AES128-CCM
- DHE-PSK-AES128-CCM8
- DHE-PSK-AES128-GCM-SHA256
- DHE-PSK-AES256-CBC-SHA
- DHE-PSK-AES256-CBC-SHA384
- DHE-PSK-AES256-CCM
- DHE-PSK-AES256-CCM8
- DHE-PSK-AES256-GCM-SHA384
- DHE-PSK-ARIA128-GCM-SHA256

- DHE-PSK-ARIA256-GCM-SHA384
- DHE-PSK-CAMELLIA128-SHA256
- DHE-PSK-CAMELLIA256-SHA384
- DHE-PSK-CHACHA20-POLY1305
- DHE-PSK-NULL-SHA
- DHE-PSK-NULL-SHA256
- DHE-PSK-NULL-SHA384
- DHE-RSA-AES128-CCM
- DHE-RSA-AES128-CCM8
- DHE-RSA-AES128-GCM-SHA256
- DHE-RSA-AES128-SHA
- DHE-RSA-AES128-SHA256
- DHE-RSA-AES256-CCM
- DHE-RSA-AES256-CCM8
- DHE-RSA-AES256-GCM-SHA384
- DHE-RSA-AES256-SHA
- DHE-RSA-AES256-SHA256
- DHE-RSA-ARIA128-GCM-SHA256
- DHE-RSA-ARIA256-GCM-SHA384
- DHE-RSA-CAMELLIA128-SHA
- DHE-RSA-CAMELLIA128-SHA256
- DHE-RSA-CAMELLIA256-SHA
- DHE-RSA-CAMELLIA256-SHA256
- DHE-RSA-CHACHA20-POLY1305
- DHE-RSA-SEED-SHA
- ECDHE-ARIA128-GCM-SHA256
- ECDHE-ARIA256-GCM-SHA384
- ECDHE-ECDSA-AES128-CCM
- ECDHE-ECDSA-AES128-CCM8
- ECDHE-ECDSA-AES128-GCM-SHA256
- ECDHE-ECDSA-AES128-SHA
- ECDHE-ECDSA-AES128-SHA256
- ECDHE-ECDSA-AES256-CCM
- ECDHE-ECDSA-AES256-CCM8
- ECDHE-ECDSA-AES256-GCM-SHA384
- ECDHE-ECDSA-AES256-SHA

- ECDHE-ECDSA-AES256-SHA384
- ECDHE-ECDSA-ARIA128-GCM-SHA256
- ECDHE-ECDSA-ARIA256-GCM-SHA384
- ECDHE-ECDSA-CAMELLIA128-SHA256
- ECDHE-ECDSA-CAMELLIA256-SHA384
- ECDHE-ECDSA-CHACHA20-POLY1305
- ECDHE-ECDSA-NULL-SHA
- ECDHE-PSK-AES128-CBC-SHA
- ECDHE-PSK-AES128-CBC-SHA256
- ECDHE-PSK-AES256-CBC-SHA
- ECDHE-PSK-AES256-CBC-SHA384
- ECDHE-PSK-CAMELLIA128-SHA256
- ECDHE-PSK-CAMELLIA256-SHA384
- ECDHE-PSK-CHACHA20-POLY1305
- ECDHE-PSK-NULL-SHA
- ECDHE-PSK-NULL-SHA256
- ECDHE-PSK-NULL-SHA384
- ECDHE-RSA-AES128-GCM-SHA256
- ECDHE-RSA-AES128-SHA256
- ECDHE-RSA-AES256-GCM-SHA384
- ECDHE-RSA-AES256-SHA384
- ECDHE-RSA-CAMELLIA128-SHA256
- ECDHE-RSA-CAMELLIA256-SHA384
- ECDHE-RSA-CHACHA20-POLY1305
- ECDHE-RSA-NULL-SHA
- NULL-MD5
- NULL-SHA
- NULL-SHA256
- PSK-AES128-CBC-SHA
- PSK-AES128-CBC-SHA256
- PSK-AES128-CCM
- PSK-AES128-CCM8
- PSK-AES128-GCM-SHA256
- PSK-AES256-CBC-SHA
- PSK-AES256-CBC-SHA384
- PSK-AES256-CCM

- PSK-AES256-CCM8
- PSK-AES256-GCM-SHA384
- PSK-ARIA128-GCM-SHA256
- PSK-ARIA256-GCM-SHA384
- PSK-CAMELLIA128-SHA256
- PSK-CAMELLIA256-SHA384
- PSK-CHACHA20-POLY1305
- PSK-NULL-SHA
- PSK-NULL-SHA256
- PSK-NULL-SHA384
- RSA-PSK-AES128-CBC-SHA
- RSA-PSK-AES128-CBC-SHA256
- RSA-PSK-AES128-GCM-SHA256
- RSA-PSK-AES256-CBC-SHA
- RSA-PSK-AES256-CBC-SHA384
- RSA-PSK-AES256-GCM-SHA384
- RSA-PSK-ARIA128-GCM-SHA256
- RSA-PSK-ARIA256-GCM-SHA384
- RSA-PSK-CAMELLIA128-SHA256
- RSA-PSK-CAMELLIA256-SHA384
- RSA-PSK-CHACHA20-POLY1305
- RSA-PSK-NULL-SHA
- RSA-PSK-NULL-SHA256
- RSA-PSK-NULL-SHA384
- SEED-SHA
- SRP-AES-128-CBC-SHA
- SRP-AES-256-CBC-SHA
- SRP-DSS-AES-128-CBC-SHA
- SRP-DSS-AES-256-CBC-SHA
- SRP-RSA-AES-128-CBC-SHA
- SRP-RSA-AES-256-CBC-SHA
- TLS_AES_128_GCM_SHA256
- TLS_AES_256_GCM_SHA384
- TLS_CHACHA20_POLY1305_SHA256

15.6. VOSS Automate Setup a Web Certificate

The VOSS Automate platform generates a 4096 bit RSA private key file, using the details stored when using the **web cert details edit** command, along with a Certificate Signing Request (.csr) file.

Repeat the steps below for each proxy that requires signed SSL certificates:

1. Check the current certificate details with **web cert details**. Initially, the `User set details` is `Unset`. For example:

```
platform@host:~$ web cert details
    Issuer data:
        C: SA
        CN: 11.120.11.100
        L: DeviceAPI
        O: Platform
        ST: WP
    Key data:
        C: SA
        CN: 11.120.11.100
        L: DeviceAPI
        O: Platform
        ST: WP
    User set details: Unset
```

2. Run **web cert details edit** if needed to edit the details displayed from the server. For example:

```
platform@host:~$ web cert details edit
Country Name (2 letter code): C:IE
State or Province Name (full name): ST:Dublin
Locality Name (eg, city): L:Dublin
Organization Name (eg, company): O:DublinSolutions Ltd.
Organizational Unit Name (eg, section): OU:R&D
Common Name (e.g. server FQDN or IP): CN:dublinsolutions.com
Email Address: platform@dublinsolutions.com
details stored
platform@host:~$
```

   Verify the edits by running **web cert details** after editing. For changes, the Issuer details will then not match the User set details.

3. Run **web cert gen_csr** to generate the Certification Signing Request (.csr) file `media/cert_sign_req.csr` for signing.

   For example:

```
platform@host:~$ web cert gen_csr
-----BEGIN CERTIFICATE REQUEST-----
M88E8TCCAtrCAQAwgasxCzAJBgNVBAYTAlpBMQswCQYDVQQIDAJXUDERMA8GA1UE
[...]
IIDr1vrepZkFQr+XDah2L5g5v8bI
-----END CERTIFICATE REQUEST-----
```

(continues on next page)

(continued from previous page)

```
==================================================
Please send the above or the actual file /opt/platform/admin/home/media/cert_sign_
↪req.csr to a CA to be signed


platform@host:~$ ls -la media/cert_sign_req.csr
-rw-rw-rw- 1 root platform 1789 Jan 18 11:20 media/cert_sign_req.csr
```

4. Use **scp** on a remote workstation to copy the file off the VOSS Automate platform `media/` directory and send it to a Certificate Authority (CA). Request a PEM format file to be returned.

   The returned file received from the CA should be a PEM certificate file. PEM certificates typically have extensions like `.pem`, `.crt`, `.cer` and `.key`.

   - If you did not receive a combined certificate from the CA, concatenate the reply signed cert and the reply intermediate CA cert into a file.

     The signed certificate must be first in the concatenated file.

     The PEM must have the correct form of line termination: a single "Line Feed" character. If your PEM file was saved on MS Windows, be sure to remove the ^M characters from the file, for example in a Linux console with:

     ```
     $ tr -d '\r' < original.pem > fixed.pem
     ```

   - If the received file is a `.p7b` file, it should be converted to a PEM format - refer to the topic: *Convert Web Certificates from P7B to PEM Format*.
   - If the received file is in another format, carry out the required conversion. For example, when a received `.crt` file is opened and is not in the correct format in MS Windows, it may show a message on MS Windows Certificate panel: "Windows does not have enough information to verify the certificate". Choose the Details tab of the panel, select Copy to File... to open the Export Wizard. Choose Base-64 encoded as export format.

5. Upload the PEM file to the proxy using **sftp** or **scp**. The file will be added to the `media/` directory, for example: `media/cert.pem`.
6. Once the file is uploaded, run **web cert add <filename of uploaded file>**. This command will combine the key and PEM file, and present it to nginx to use for secure (SSL) web communication. For example:

```
platform@host:~$ web cert add media/cert.pem
Updating the certificate requires the web server to be restarted.
Do you wish to continue? yes
Restarting nginx
platform@host:~$
```

15.7. Own Web Certificate Setup

The steps below provide and example of own private certificate and generated Subject Alternative Name (SAN) certificate setup as summarized in *Web Certificate Setup Options*.

See also:

- *VOSS Automate Setup a Web Certificate*
- *Convert Web Certificates from P7B to PEM Format*

Note: The Subject Alternative Name (`alt_names`) field lets you specify additional host names (sites, IP addresses, common names, etc.) to be protected by a single SAN Certificate.

---

1. Log into a system that has the **openssl** command set up.
2. Create a bash script file with contents as for example below.

   Note: This is an example, and not necessarily comprehensive. Refer to OpenSSL documentation for details.

```
openssl req -new -sha256 -nodes -out cert.csr -newkey rsa:4096 -keyout private.key -
↪config <(
cat <<-EOF
[req]
default_bits = 2048
prompt = no
default_md = sha256
req_extensions = req_ext
distinguished_name = dn

[ dn ]
C=<Country code>
ST=<County/State>
L=<City>
O=<Orginization>
OU=<Org Unit>
emailAddress=<admin email address>
CN = <Main DNS Name>

[ req_ext ]
subjectAltName = @alt_names

[ alt_names ]
DNS.1 = <Alternate name 1>
DNS.2 = <Alternate name 2>
IP.1 = <Alternate IP 1>

... you can add more below ...
EOF
)
```

3. Edit the sections in < > brackets.
4. Run **bash <scriptfile from above>**
5. Send the file called `cert.csr` to your CA, requesting them to make sure to sign it as a SAN certificate.
6. Take the file that they send back, save it as `signed.crt`
7. Combine the `private.key file` with `signed.crt`:

   Run **cat private.key signed.crt > complete.cert**
8. Upload the `complete.cert` file to the VOSS Automate system using **sftp** or **scp**. The file will be added to the `media/` directory, for example: `media/complete.cert`
9. On the VOSS Automate system, run **web cert add_san media/complete.cert**

15.8. Web Certificate Expiration Notice

If a Web Certificate is due to expire, a notice will display on the status display 30 days before expiration:

```
platform@development:~$ health


host: AS01, role: webproxy,application,database, LOAD: 3.85
date: 2014-08-28 11:24:22 +00:00, up: 6 days, 3:03
network: 172.29.42.100, ntp: 196.26.5.10
HEALTH: NOT MONITORED
database: 20Gb
application: up
WEB CERT EXPIRES AT: 2014-09-26 11:30:02


    mail - local mail management          keys - ssh/sftp credentials
 network - network management           backup - manage backups
    voss - voss management tools           log - manage system logs
database - database management          notify - notifications control
schedule - scheduling commands            diag - system diagnostic tools
  system - system administration          snmp - snmp configuration
    user - manage users                 cluster - cluster management
  drives - manage disk drives               web - web server management
     app - manage applications         template - template pack creator
```

Web certificate expiration can also be monitored if scheduled health monitoring is enabled - see:`enable_health_monitoring`. The health email will then show this message.

If a Web Certificate has expired, the notice on the status displays:

```
WEB CERT EXPIRED AT: 2014-09-26 11:30:02
```

Once the certificate is expired, the system can be used as normal, but the certificate will be expired and for non self-signed certificates (like a Godaddy or Thawte certificates), the data will no longer be properly encrypted.

15.8.1. Renewing Expired Certificates

According to the certificate type in use, refer to the setup steps to manage certificates:

- *Own Web Certificate Setup*
- *VOSS Automate Setup a Web Certificate*

15.9. Convert Web Certificates from P7B to PEM Format

VOSS Automate uses web certificates in Privacy Enhanced Mail (PEM) format. PEM certificates typically have extensions like `.pem`, `.crt`, `.cer` and `.key`.

If a P7B format certificate is received from a Certificate Authority (CA):

1. Copy the files to a workstation with Linux console available (Not the VOSS Automate system).
2. Run the following command for each `<filename>.p7b`, for example:

   ```
   sudo openssl pkcs7 -in <filename>.p7b -inform DER -print_certs -out <filename>.pem
   ```

3. Open the PEM file in a text editor. You will see formatting like the example below in the file:

   ```
   subject=/C=GB/ST=West Midlands/L=Coventry/O=Service Coventry/OU=Network Services/
      CN=ccs-cp-v4uc.svcoventry.gov.uk/emailAddress=network@svcoventry.co.uk
   issuer=/C=GB/O=Coventry/OU=PKI/CN=BCC Intermediate CA
   -----BEGIN CERTIFICATE-----
   MIIFxTLmBK2gAwIBAgITXgAAAMBXLQb0/ImKBwALmQAAwDANBgkqhkiG9w0BAQsF
   ADBOMQswLmYDVLmQEwJHQjETMBEGA1UEQhMKQmlybWluZ2hhbTEMMAoGA1UEQxMD
   UEtJMRwwGgYDVLmQExNLm0MgSW50ZXJtZWRpYXRlIENBMB4LmQE2MTAwNTEyNTIx
   ```

4. Delete all text and blank lines outside the lines:

   `-----BEGIN CERTIFICATE-----` and `-----END CERTIFICATE-----`

5. Save the file and make sure that the file is not saved in a DOS format.

15.10. Web Certificate Commands

The following Command Line Interface console display shows the available commands for web certificates.

```
web cert add <filename>                 - Install the certificate from <filename>
                                          into the web server
web cert add_san <filename>             - Install a SAN certificate from <filename>
                                          into the web server
web cert del                            - Revert to a self-signed certificate
web cert details                        - Print the certicate details in config
                                          system
web cert details edit                   - Update the certicate details in config
                                          system
web cert gen_csr                        - Create a CSR file in
                                          /opt/platform/admin/home/media
```

(continues on next page)

(continued from previous page)

```
web cert gen_selfsigned                - Generate a self-signed certificate
web cert print_csr                     - Create a CSR file in
                                         /opt/platform/admin/home/media
web cluster prepnode                   - Prepares the system so that it can be
                                         joined to a cluster as a web proxy
web ssl list                           - Shows a list of the supported SSL protocols, and␣
↪their current state
web ssl enable <protocol>              - Enable an SSL protocol
web ssl disable <protocol>             - Disable an SSL protocol
web weight add <server:port> <weight> - Modify the weights of an upstream
                                         service. Higher weights will serve
                                         more requests, while 0 will only be
                                         used if no other servers are available
web weight del <server:port>           - Delete the user-defined service weight
                                         and use the system default.
web weight list                        - Display the weights of upstream services
```

15.11. Web Hosts Commands

In order to manage the location HTTP header in HTTP redirect responses to only include safe hosts, the **web hosts** command can be used with the required parameters.

This feature protects against host header injection during the `http` -> `https` redirect upon login to VOSS Automate. The value supplied in the Host header is contained in the HTTP redirect response in the `Location` HTTP header.

The following Command Line Interface console display shows the available commands for web hosts.

```
web hosts add <hostname>  - Add the hostname to the allowed hosts
web hosts del <hostname>  - Delete the hostname from the allowed hosts
web hosts disable         - Disables the allowed hosts feature
web hosts enable          - Enables the allowed hosts feature, which blocks requests␣
↪with unrecognised
                            HOST headers
web hosts list            - Displays the additional allowed hosts
```

15.11.1. Enable the feature

```
$ web hosts enable
Enabling the allowed hosts feature requires the web server.
Do you wish to continue? y
Allowed hosts
    enabled: true
    hosts: value not set


Restarting nginx for settings to take effect
```

(continues on next page)

(continued from previous page)

```
Application nginx processes stopped.

Application services:firewall processes stopped.
Reconfiguring applications...
Application nginx processes started.
```

After the feature is enabled and no hosts specified, the web server closes the connection.

15.11.2. Add and Delete a <hostname>

In this example, the hostname `atlantic.net` is added.

```
$ web hosts add atlantic.net
Adding a new allowed host requires the web server to be restarted.
Do you wish to continue? y
Allowed hosts
    enabled: true
    hosts:
        atlantic.net
Restarting nginx for settings to take effect
Application nginx processes stopped.
Application services:firewall processes stopped.
Reconfiguring applications...
Application nginx processes started.
```

Note:

- For hostname format, refer to for example: RFC 1035, RFC 2181 and RFC 4343.

To remove a hostname from the list (example is `atlantic.net`):

```
$ web hosts del atlantic.net
```

15.11.3. Listing host names

Use the **web hosts list** command to show status and list all safe hosts that can be in the `Location` HTTP header.

```
$ web hosts list
Allowed hosts
    enabled: true
    hosts:
        atlantic.net
```

15.11.4. Disabling the feature

The feature can be disabled with the **web hosts disable** command. This will disable port 80 on the web server completely.

```
$ web hosts disable
Disabling the allowed hosts feature requires the web server to be restarted.
Do you wish to continue? y
Allowed hosts
    enabled: false
    hosts: value not set
Restarting nginx for settings to take effect
Application nginx processes stopped.
Application services:firewall processes stopped.
Reconfiguring applications...
Application nginx processes started.
```

15.12. Web TLS Protocol Configuration

Commands are available to list Transport Layer Security (TLS) protocol versions and also to enable or disable TLS versions.

Note:

- The command should be run on all nodes in a cluster.
- When enabling or disabling a TLS protocol version, the web server needs to be restarted. Running the command will show a message and carry out this task.

The following protocols are available in VOSS Automate:

- TLSv1.2
- TLSv1.3

Important:

- TLSv1.2 is enabled by default upon installation. Upon upgrade, your current protocol is retained.
- TLSv1.2 can only be disabled by enabling TLSv1.3.

- **web ssl list**

  Example:

  ```
  $ web ssl list
  TLSv1.3: Disabled
  TLSv1.2: Enabled
  ```

  - Enabling or disabling a protocol that is already in that state, will raise an error message.
- **web ssl disable <TLS version>**

  - Enabling or disabling a protocol that is already in that state, will raise an error message.

Example:

```
$ web ssl disable TLSv1.2
Disabling the TLSv1.2 protocol requires the web server to be restarted.
Do you wish to continue? yes
TLSv1.2: Disabled
TLSv1.3: Enabled

Restarting nginx for settings to take effect

Application nginx processes stopped.


Application services: firewall processes stopped.
Application nginx processes started.
```

- **web ssl enable <TLS version>**

  Note:

  - When running **web ssl enable TLSv1.3**, it will disable TLSv1.2. Users will not be able to alter web ciphers.
  - When running **web ssl enable TLSv1.2**, it will disable TLSv1.3. Users can change the web ciphers.

  - Enabling or disabling a protocol that is already in that state, will raise an error message.

Example:

```
$ web ssl enable TLSv1.3
Enabling the TLSv1.3 protocol requires the web server to be restarted.
Do you wish to continue? yes
TLSv1.3: Enabled
TLSv1.2: Enabled


Restarting nginx for settings to take effect

Application nginx processes stopped.


Application services: firewall processes stopped.
Application nginx processes started.
```

The table below shows the result of running **web ssl enable** or **web ssl disable** given a specific state (from **web ssl list**).

State		Command	Result	
1.2	1.3	on/off	1.2	1.3
on	off	1.2 on	on	off
off	on	1.3 on	off	on
on	off	1.2 off	off	on
off	on	1.3 off	on	off

15.13. Web TLS Cipher Management

Web TLS ciphers on the VOSS Automate platform can be listed and managed. This can be done as follows:

- **web ssl cipher list** will list nginx ciphers grouped by status: `disabled`, `enabled`.
- **web ssl cipher default** will set the default nginx ciphers. This command requires the web server to be restarted.
- **web ssl cipher enable <space separated cipher(s)>** will enable the listed nginx ciphers. This command requires the web server to be restarted.
- **web ssl cipher disable <space separated cipher(s)>** will disable the listed nginx ciphers. This command requires the web server to be restarted.

Note: The enabled ciphers cannot *all* be disabled.

Command examples:

- List:

```
platform@VOSS:~$ web ssl cipher list
    enabled:
        ECDHE-RSA-AES256-SHA
        ECDHE-ECDSA-AES256-SHA
        SRP-DSS-AES-256-CBC-SHA
        SRP-RSA-AES-256-CBC-SHA
        SRP-AES-256-CBC-SHA
        DHE-RSA-AES256-SHA
        DHE-DSS-AES256-SHA
        DH-RSA-AES256-SHA
        DH-DSS-AES256-SHA
        DHE-RSA-CAMELLIA256-SHA
        DHE-DSS-CAMELLIA256-SHA
    ...
```

- Disable:

```
platform@VOSS:~$ web ssl cipher disable CAMELLIA256-SHA
Disabling nginx ciphers requires the web server to be restarted.
Do you wish to continue? y
```

(continues on next page)

(continued from previous page)

```
Application services:firewall processes stopped.
Application nginx processes stopped.
Reconfiguring applications...
Application nginx processes started.
    disabled:
        CAMELLIA256-SHA
    enabled:
        ECDHE-RSA-AES256-GCM-SHA384
        ECDHE-ECDSA-AES256-GCM-SHA384
        ECDHE-RSA-AES256-SHA384
    ...
```

15.14. Network URI specification

All network locations are specified as a URI, for example download locations, backup destinations, notification destinations, and so on.

The following are examples:

`mailto:user@host`

`sftp://user:password@host/directory`

`ssh://user:password@host/directory`

`snmp://community@host` for SNMP v2

`snmp://community:password@host` for SNMP v3

15.15. Web External Proxies

On a web proxy node only, remote server proxies can be managed.

The commands should be run on the relevant web proxy node.

The commands to add or remove remote server proxies will automatically reconfigure and restart the nginx process, so some downtime will result.

- To *add* remote server proxies and ports, run the command on the relevant node:

  web external add <URL> <remote IP:port>

  Note the `<URL>` can include a path, for example:

  `http://172.49.90.53:8888/myurl/set`

  so that when the proxy URL is then entered as `https://<vossIP>/myproxy/`, this will then resolve to `http://172.49.90.53:8888/myurl/set`.

- To *remove* remote server proxies, run the command on the relevant node:

  web external del <URL>

- To *list* configured remote proxies on a web proxy node, run the command on the relevant node:

  web external list

For example, adding, listing and removing remote servers:

```
platform@VOSS:~$ web external add myproxy http://172.49.90.53:8888
Updating will restart Nginx.
Do you wish to continue? y
You have new mail in /var/mail/platform

platform@VOSS:~$ web external list
    external:
        myproxy:
            upstream: http://172.49.90.53:8888

platform@VOSS:~$ web external del myproxy
Updating will restart Nginx.
Do you wish to continue? y
You have new mail in /var/mail/platform

platform@VOSS:~$ web external list
    external: value not set
```

16. High Availability and Disaster Recovery (DR)

16.1. High Availability Overview

High Availability (HA) is an approach to IT system design and configuration that ensures VOSS Automate is operational and accessible during a specified time frame. This is achieved using redundant hardware and resources. If there is a failure, an automatic failover will occur to the secondary database node.

This section outlines the configuration steps to deploy a HA enabled VOSS Automate Platform on VMware. It presupposes familiarity with the Installation Guide and Platform Guide - for the latter guide, in particular the topics on DR Failover and Recovery.

16.2. Default HA and DR scenario

VOSS Automate supports using off-the-shelf VMware tools.

High Availability is implemented using VMware HA clusters, with data accessed via a central storage facility (SAN). VMware monitors the primary server, and should it fail, another instance of the VM is automatically started on a different hardware instance. Since data is shared on the SAN, the new HA instance will have access to the full dataset.

Disaster Recovery is implemented by streaming data updates to a separate DR instance that remains powered on. If the primary server fails, the DR instance can take over operation. The switch-over to DR instance is scripted, but must be invoked manually.

During a HA failover, the HA instance assumes the primary IP address, and no reconfiguration of other UC elements is required. However, in the case of a DR failover, interaction with other UC elements should be considered.

- DNS can be used effectively to provide hostname abstraction of underlying IP addresses. In such a case, a DNS update will allow existing UC elements to seamlessly interact with the new DR instance.
- If DNS is not available, and the UC elements cannot be configured with the IP address of the DR instance, it is necessary for the DR instance to assume the primary IP address. In such a case, the DR and the primary IP addresses can be swapped using the CLI interface. Standard networking practices should be employed to ensure that the IP address is correctly routed, e.g. Stretched layer-2 vLAN, and ensuring that the Primary and DR instances are not operated with the same IP address.

The following failure points should be considered:

- Since the HA instance is started automatically if the primary instance fails, a slight interruption in service is expected, including VMware polling latency in determining that the primary server has failed, and the startup delay of the HA instance. This delay is around 3 minutes

- If data is corrupted on the SAN, the HA instance will start with the same corrupt code and data instances
- Since VMware is checking only for VM liveness, it is not able to check that the primary instance is functionally active.
- Data updates are transported to the DR instance. If data updates cannot be shipped by the primary instance, SNMP traps are generated informing administration of the problem. However, if this is not fixed timeously, it is possible for the DR instance to become out of sync. These delays could result in data loss between the primary and DR instances. Database updates are scheduled every 3 minutes and/or 16MB.
- There are certain manual steps that are required to bring the DR instance online. These steps are documented in the Platform Guide.

16.3. HA and DR scenario with Cisco VMDC geo-redundancy architecture

HA and DR instances can be geo-relocated at will within the capabilities of the underlying network architecture.

For example, it is feasible to extend a VMware HA cluster geographically using high speed data links and layer-2 stretched vLANs.

DR as implemented by the VOSS Automate system lends itself to geographical separation with streaming data replication to a second powered-on instance.

Interaction with other UC elements must be considered within the capabilities of the network, using either DNS for seamless transition, or IP reconfiguration either within the UC elements or the VOSS Automate system.

16.4. Configuring a HA System Platform on VMware

This is an optional step, however, for production servers it is highly recommended that they are run in a HA deployment configuration. This can be done by the client, but should be checked by a system representative

1. Log into VMware VSphere, then select **File > New >Cluster...**.
2. Enter the Name, and select the Turn on VMware HA check box.
3. Make sure that the Enable Host Monitoring check box and Enable: Do not power on VMs that violate availability constraints radio buttons are selected.
4. Select the required default restart priority.
5. Select the VM Monitoring Only option from the VM Monitoring drop-down list, and set the Default Cluster Settings/Monitoring sensitivity to High.
6. Select the Disable EVC radio button, unless you know the exact version of CPU technologies that are enabled on your system.
7. Select the Store the swapfile in the same directory as the virtual machine (recommended) radio button.
8. Ensure the settings are all correct and click the **Finish** button.
9. Drag all of the machines that will be used into the newly created cluster.

10. Once done, they will be listed below the new cluster, with any VM's that were moved into the root of the cluster.
11. Select each of the Machines in the cluster then select the Configuration tab.
12. If Time Configuration is displayed in red, select Properties, then click the **Options** button.
13. Select NTP Settings, and then click the **Add** button.
14. Select the Restart NTP service to apply changes check box, and then click the **OK** button.
15. Select the relevant Cluster, and then select the Summary tab. There should be no configuration issues listed.

The production OVA is deployed as in the hardware specifications of the deployment topologies and installation steps and considerations indicated in the Installation Guide.

16.5. DR Failover and Recovery Overview

16.5.1. DR Failover

The VOSS Automate system makes use of database replication facilities during normal operation. During a failover, if 50% or more of the service resources are lost, the system will no longer function without manual intervention.

In this case, the following high level process should be followed.

1. Display the current cluster topology using **cluster status**.
2. Remove the dead nodes using **cluster del <ip>**. Power off the deleted node, or disable its Network Interface Card.
3. Once the cluster topology is adjusted, the cluster must be reprovisioned using **cluster provision**.
4. Afterward, the cluster status can be rechecked with **cluster status**.

See also: *Using the tmux command*.

Refer to the appropriate detailed DR scenarios for the complete sequence of DR steps.

16.5.2. Cluster Failure Scenarios

The status of the cluster can be displayed from the command-line on any node using the command:

cluster status

Note: In the case where a node is down, the command output will show `unknown`, for example:

```
platform@VOSS-UN-1:~$ cluster status

Data Centre: unknown
     application : unknown_192.168.100.4[192.168.100.4] (not responding)
     webproxy : unknown_192.168.100.4[192.168.100.4] (not responding)
     database : unknown_192.168.100.4[192.168.100.4] (not responding)
```

(continues on next page)

(continued from previous page)

```
Data Centre: jhb
     application : VOSS-UN-5[192.168.100.9]
                   VOSS-UN-6[192.168.100.10]
     webproxy : VOSS-UN-5[192.168.100.9]
                   VOSS-WP-3[192.168.100.11]
                   VOSS-UN-6[192.168.100.10]
     database : VOSS-UN-5[192.168.100.9]
                   VOSS-UN-6[192.168.100.10]

...
```

The system can automatically signal email and/or SNMP events in the event that a node is found to be down.

Refer to the diagrams in the Installation Guide section on deployments.

Loss of an Application role
The Web Proxy will keep directing traffic to alternate Application role servers. There is no downtime.

Loss of a Web Proxy
Communication via the lost Web Proxy will fail, unless some another load balancing infrastructure is in place (DNS, external load balancer, VIP technology). The node can be installed as a HA pair so that the VMware infrastructure will restore the node if it fails. Downtime takes place while updating the DNS entry or returning the Web Proxy to service. For continued service, traffic can be directed to an alternate Web Proxy or directly to an Application node if available. Traffic can be directed manually (i.e. network elements must be configured to forward traffic to the alternate Web Proxy).

Loss of a Database role
If the primary database service is lost, the system will automatically revert to the secondary database. The primary and secondary database nodes can be configured via the Command Line Interface (CLI) using **database weight <ip> <weight>**. For example, the primary can be configured with a weight of 40, and the secondary with a weight of 20. If both the primary and the secondary Database servers are lost, the remaining Database servers will vote to elect a new primary Database server. There is downtime (usually no more than a few seconds) during election and failover, with a possible loss of data in transit (a single transaction). The GUI web-frontend transaction status can be queried to determine if any transactions failed. The downtime for a Primary to Secondary failover is significantly less and the risk of data loss likewise reduced. A full election (with higher downtime and risk) is therefore limited only to cases of severe outages where it is unavoidable.

Although any values can be used, for 4 database nodes the weights: 40/30/20/10 is recommended and for 6 database nodes, 60/50/40/30/20/10. These numbers ensure that if a reprovision happens (when the primary data center goes offline for an indeterminate time), the remaining systems have weights that will allow a new primary to be chosen.

Loss of a site
Unified and Database nodes have database roles. The status of the roles can be displayed using **cluster status**. If 50% or more of the database roles are down, then there is insufficient availability for the cluster to function as is. Either additional role servers must be added, or the nodes with down roles must be removed from the cluster and the cluster needs to be reprovisioned. If there is insufficient (less than 50% means the system is down) Database role availability, manual intervention is required to reprovision the system – downtime is dependent on the size of the cluster. Refer to the Platform Guide for details on DR Failover. Database role availability can be increased by adding Database roles, providing greater probability of automatic failover. To delete a failed node and replace it with a new one if database primary is for example lost: The node can be deleted using **cluster del <ip>**. Additional nodes can be deployed and added to the cluster with **cluster add <ip>**. The database weights can be

adjusted using **database weight <ip> <weight>**. Finally, the cluster can be reprovisioned with **cluster provision** (it is recommended that this step is run in a terminal opened with the `tmux` command). This command is the same as **cluster provision fast**. The `fast` parameter is available for backwards compatibility and is the default behavior, which is to run the provisioning on all nodes in parallel. Use the command **cluster provision serial** on systems where the VMware host is under load.

The console output below shows examples of these commands.

The cluster status:

```
platform@cpt-bld2-cluster-01:~$ cluster status


Data Centre: jhb
     application : cpt-bld2-cluster-04[172.29.21.243]
                   cpt-bld2-cluster-03[172.29.21.242]

        webproxy : cpt-bld2-cluster-06[172.29.21.245]
                   cpt-bld2-cluster-04[172.29.21.243]
                   cpt-bld2-cluster-03[172.29.21.242]

        database : cpt-bld2-cluster-04[172.29.21.243]
                   cpt-bld2-cluster-03[172.29.21.242]


Data Centre: cpt
     application : cpt-bld2-cluster-02[172.29.21.241]
                   cpt-bld2-cluster-01[172.29.21.240] (services down)

        webproxy : cpt-bld2-cluster-05[172.29.21.244]
                   cpt-bld2-cluster-02[172.29.21.241]
                   cpt-bld2-cluster-01[172.29.21.240] (services down)

        database : cpt-bld2-cluster-02[172.29.21.241]
                   cpt-bld2-cluster-01[172.29.21.240] (services down)
```

Deleting a node:

```
platform@cpt-bld2-cluster-01:~$ cluster del 172.29.21.245
You are about to delete a host from the cluster. Do you wish to continue? y
Cluster successfully deleted node 172.29.21.245

Please run 'cluster provision' to reprovision the services in the cluster

Please note that the remote host may still be part of the database clustering
and should either be shut down or reprovisioned as a single node BEFORE this
cluster is reprovisioned
You have new mail in /var/mail/platform
```

Adding a node:

```
platform@cpt-bld2-cluster-01:~$ cluster add 172.29.21.245

Cluster successfully invited node 172.29.21.245
```

(continues on next page)

(continued from previous page)

```
Please run 'cluster provision' to provision the services in the cluster
```

Database weights: listing and adding

```
platform@cpt-bld2-cluster-01:~$ database weight list
    172.29.21.240:
        weight: 5
    172.29.21.241:
        weight: 3
    172.29.21.243:
        weight: 2
    172.29.21.244:
        weight: 1

platform@cpt-bld2-cluster-01:~$ database weight 172.29.21.240 10
    172.29.21.240:
        weight: 10
    172.29.21.241:
        weight: 3
    172.29.21.243:
        weight: 2
    172.29.21.244:
        weight: 1
```

16.5.3. Platform install OVA on a VM

Overview

This topic describes the steps for a fresh install, using the latest OVA file.

Note: If an OVA file is not available for your current release, you need to obtain the most recent release OVA for which there is an upgrade path to your release. Refer to *What to do if the OVA file is not available for your current version*

The steps for creating a VM and running the wizard should be completed for *each* node in your topology. Therefore, these tasks will need to be performed either once or multiple times during installation, depending on the topology you're installing.

The install tasks (creating the VM and running the install wizard) describe the steps for a common setup of a *single node* from the OVA file for each of the following fresh install scenarios:

- Standalone installation
- A node install during multi-node installation

  Note: For the *node install during multi-node installation* option, refer to *Role of each VM installation for multi-node installation*.

- Failover recovery

Step 1: Download the OVA file

Download the **OVA** file for your release from the **New Installation** folder on the client portal.

The downloaded OVA file is imported into VMware vCenter Server (when creating the VM). Only one OVA file is used to deploy all the functional roles. Each time you run the install wizard on the VM, you will select the relevant role for the node type you're installing on.

Step 2: Create the VM

Prerequisites:

- *Step 1: Download the OVA file*

To create the VM:

1. Log in to vSphere to access the ESXi Host.
2. Choose **File > Deploy OVF Template**.
3. Choose Source, browse to the location of the **.ova** file, and click **Next**.
4. On the Name and Location page, fill out a name for this server.
5. On the **Deployment Configuration** page, select the appropriate node type.

   Note: Refer to the description for **role** in the install wizard step, below.

6. Choose the resource pool in which to locate the VM.
7. Choose the data store you want to use to deploy the new VM.
8. Choose the disk format to use when deploying the new VM.
   - For non-SSD-based drives in production environments, "thick provisioning" is mandatory. Thick Provision Eager Zeroed is recommended.
   - For SSD-based drives, "thin provisioning" is supported.
9. On the **Network Mapping** page, choose your network on which this VM will reside.
10. Do not select Power on after deployment.
11. On the **Ready to Complete** page, click **Finish** to start the deployment.
12. After the VM is created, select the CD ROM configuration and verify the **Connect at Power On** checkbox is enabled. Also, verify the memory, CPU, and disk settings against the requirements shown in either the Single-node cluster (cluster-of-one) System Hardware Specification or Multi-node Cluster Hardware Specification section in the Install Guide.
13. Next steps: *Step 3: Run the installation wizard on the VM*

Step 3: Run the installation wizard on the VM

Prerequisites:

- *Step 2: Create the VM*

To run the install wizard on the VM:

1. Power on the VM.
2. Configure the options in the installation wizard:

Step	Description
1. network device	The network device name.
2. IP	The IP address of the server. The required format is with Classless Inter-Domain Routing (CIDR): `ip/netmask`.[1]
3. gateway	The IP address of the network gateway.Page 234, 1
4. DNS	The DNS server is optional. Ensure that the DNS server is capable of looking up all hostnames referred to, including NTP server and remote backup locations.[1]
5. NTP	The NTP server is mandatory to ensure that time keeping is accurate and synchronized among nodes in the same cluster.[1]
6. boot password	Enable boot loader configuration password. See the example below.
7. hostname	The hostname, not the fully qualified domain name (FQDN). The maximum character length for the hostname is 56.
8. role	Choose a role for the node you're installing on. Note: only WebProxy, Application and Database nodes are used for a modular architecture installation. • A WebProxy role installs only the front-end web server together with ability to distribute load among multiple middleware nodes. • An Application node is the main transaction processing engine and includes a web server which can operate by itself, or route transactions from a web node. • A Database node provides persistent storage of data. • A Standalone node consists of the Web, Application, and Database roles on one node. For Single-node cluster (cluster-of-one). • A Unified node consists of the Web, Application, and Database roles on one node. On installation, the system needs to be clustered with other nodes and the cluster provisioned. • A General node used for M2UC, NBI.
9. data center	The system's geographic location (data center name, city, country that a customer can use to identify the system location). You cannot change this setting once set.
10. platform password	Platform password must be at least eight characters long and must contain both uppercase and lowercase letters and at least one numeric or special character.

[1] VOSS Automate supports IPv4 or IPv6
IPv6 allows the following input formats to be used:
- IPv6 Compressed, e.g.: `::ffff:c0a8:6403/64`
- IPv6 Expanded (Shortened), e.g.: `0:0:0:0:0:ffff:c0a8:6403/64`

3. Once all details are entered, installation proceeds. Monitor install progress.

   When the installation completes, the system reboots. Since all services will be stopped, this takes some time.

Related topics

- *Boot password and security*

Step 4: Finalize the installation

1. Once the OVA installation completes, a sign-in prompt for the platform user displays. This confirms that the system is ready for use.
2. Connect to newly deployed server CLI as the platform user.

   A login message such as the following displays:

```
Last login: Wed Nov  2 11:12:45 UTC 2016 from thwh on pts/6
Last failed login: Wed Nov  2 11:19:53 UTC 2016 from iza on ssh:notty
There were 2 failed login attempts since the last successful login.

host: dev-test, role: webproxy,application,database, load: 0.21, USERS: 3
date: 2016-11-02 11:19:57 +00:00, up: 14:19
network: 172.29.253.14, ntp: 172.29.1.15
HEALTH: NOT MONITORED
database: 31Gb
Failed logins: 2 since Wed Nov 02 11:19:53 2016 from iza

    mail - local mail management          keys - ssh/sftp credentials
 network - network management           backup - manage backups
    voss - voss management tools           log - manage system logs
database - database management          notify - notifications control
schedule - scheduling commands        selfservice - selfservice management
    diag - system diagnostic tools      system - system administration
    snmp - snmp configuration             user - manage users
 cluster - cluster management           drives - manage disk drives
     web - web server management           app - manage applications
```

   If the user failed to log in prior to a successful login, the count, date, and origin of the attempts are shown as *Failed logins*. A successful login resets this login count.

3. Run `app status` on all application nodes and ensure the services are all running and reporting the correct version before continuing.
4. Return to Multi-node Installation, Standalone Installation or Failover step to complete the overall installation or failover recovery procedure.

• IPV6 Expanded, e.g.: `0000:0000:0000:0000:0000:ffff:c0a8:6403/64`

From Automate release 24.1 onwards, network addresses are in CIDR (Classless Inter-Domain Routing) format, for example: `192.168.100.3/27` or `e00d::fafe:23/112`. The use of a netmask in the `255.255.255.0` format is no longer supported.

On a fresh install, if you run the install on a network with a DHCP server and encounter the following error, you can enter a valid DNS server address to continue the installation:

"Error: DNS server <DNS server> is either invalid or cannot be reached on the network"

Boot password and security

The default security protocol for the web server is TLSv1.2.

Password protection can be enabled on the VOSS Automate boot loader configuration from the install wizard upon first install and also from the CLI - see the topic on System Boot Passwords in the Platform Guide for commands to enable, disable or reset the boot password.

Important: The boot password is non-recoverable.

The console example below shows the `boot password` configuration output:

```
(1)              ip    (199.29.21.89)
(2)         netmask    (255.255.255.0)
(3)         gateway    (199.29.21.1)
(4)             dns    (199.29.88.56)
(5)             ntp    (199.29.88.56)
(6)   boot password    (disabled)
(7)        hostname    (atlantic)
(8)            role    (UNDEFINED)
(9)     data centre    (earth)
(10) platform password (UNDEFINED)
Select option ? 6
Valid passwords must contain:
   at least one lower- and one upper-case letter,
   at least one numeric digit
   and a special character eg. !#@$%&^*
Password: Please enter platform user password:
 Please re-enter password
Password:
NOTE: The system boot password is now set for user platform.
```

When the boot password is set, the wizard will show:

```
(6)   boot password    (*****)
```

Role of each VM installation for multi-node installation

According to the multi-node deployment topology and specification, the *role* of each VM installation is as indicated below.

- **For each web proxy instance**:
  - Create a new VM using the platform-install OVA.
  - For **role**, select **(3) WebProxy**.
  - Specify the appropriate data center (Primary/DR site) for each web proxy instance.
- **For each unified instance** (*Standard Topology only*):
  - Create a new VM using the platform-install OVA.
  - For **role**, select **(2) Unified**.
  - Specify the appropriate data center (Primary/DR Site) for each unified instance.

The following unified nodes are required in the cluster:

- One unified node as the Primary node at the Primary site
- One unified node as the Secondary node at the Primary site
- Two unified nodes as the Secondary nodes at the Disaster Recovery (DR) site

---

Note: For a six node multi cluster deployment there are:

- Two unified nodes (one Primary and one Secondary)
- One web proxy node at the Primary site
- Two unified nodes (both Secondary)
- One web proxy node at the DR site

For an eight node multi cluster deployment, there are:

- Four unified nodes (one Primary and three Secondary)
- One web proxy node at the Primary site
- Two unified nodes (both Secondary)
- One web proxy node at the DR site

- *Modular Architecture Topology*

The following nodes are required in a typical Modular Architecture cluster:

- One Application node as the Primary node at the Primary site
- One additional Application node at the Primary site
- One Database node as the Primary Database node at the Primary site
- One additional Database node at the Primary site
- One Application node at the Disaster Recovery (DR) site
- One Database node at the Disaster Recovery (DR) site

Note: For a typical Modular Architecture cluster there is one web proxy at the Primary site and one WebProxy node at the DR site.

For each Database instance:

- Create a new VM using the platform-install OVA.
- For **role**, select **(2) Database**.
- Specify the appropriate data center (Primary/DR Site) for each database instance.

For each Application instance:

- Create a new VM using the platform-install OVA.
- For **role**, select **(2) Application**.
- Specify the appropriate data center (Primary/DR Site) for each Application instance.

Also refer to Multi-node Installation section in the Install Guide.

Detailed configuration can be applied from the Command Line Interface (CLI). Use the following commands for details: `network help` or `network`

For example, domain can be configured using `network domain add <domain-name>`.

For a geo-redundant deployment, the **data center** information entered in the wizard is equivalent to the location information.

What to do if the OVA file is not available for your current version

If the OVA file is not available for your current release, you will need to obtain and install the most recent release OVA for which there is an upgrade path to your current release. In this case however, you will be performing an upgrade of your system.

The table describes the steps for obtaining the relevant OVA file and applying the upgrade:

Scenario	If the OVA file is not available for your current release . . .
Standalone installation	1. Obtain and install the most recent release OVA for which there is an upgrade path to your release. 2. Apply the Delta Bundle Upgrade steps for the current release to the OVA to upgrade it.
A node install during multi-node installation	1. Obtain and install the most recent release OVA for which there is an upgrade path to your release. 2. Apply the Delta Bundle Upgrade steps for the current release *to the cluster* to upgrade it. Refer to the *Upgrade Guide with Delta Bundle*.
Failover recovery	1. Obtain and install the most recent release OVA for which there is an upgrade path to your release. 2. Add it to your cluster. Use the same configure options in the table below as were applied to the lost node. **Note:** The node version mismatch in the cluster can be ignored, since the next upgrade step aligns the versions. 3. Apply the Delta Bundle Upgrade steps for the current release *to the cluster* to upgrade it. For details, refer to the *Upgrade Guide with Delta Bundle* and to the specific scenario Disaster Recovery steps in the *Platform Guide*.

16.6. DR Failover and Recovery in a Unified Node Cluster Topology

16.6.1. Election of a New Primary and Failover

In the case where unified nodes fail, the system follows a failover procedure. For details on the failover and DR process, refer to the topics in the Platform Guide.

If the primary database is lost, the failover process involves the election of a new primary database by the remaining database nodes. Each node in a cluster is allocated a number of votes that are used in the failover election of a new primary database - the election of a running node with the highest database weight.

The database weights for a node can be seen as the `priority` value when running the **database config** command. Note that database weight of a node does not necessarily match its number of votes.

```
$ database config
    date: 2016-04-25T09:50:34Z
    members:
        172.29.21.101:27020:
            priority: 16
            stateStr: PRIMARY
        172.29.21.101:27030:
            stateStr: ARBITER
        172.29.21.102:27020:
            priority: 8
            stateStr: SECONDARY
        172.29.21.102:27030:
            stateStr: ARBITER
        172.29.21.103:27020:
            priority: 4
            stateStr: SECONDARY
        172.29.21.103:27030:
            stateStr: ARBITER
        172.29.21.104:27020:
            priority: 2
            stateStr: SECONDARY
    myState: 1
    ok: 1
    set: DEVICEAPI
```

The maximum number of votes in a cluster should not exceed 7 and arbiter votes are added to nodes to provide a total of 7 votes.

The tables below show the system status and failover for a selection of scenarios for 6 node and 8 node clusters. Also refer to the topics on the specific DR scenarios. The abbreviations used are as follows:

- Pri : Primary site
- DR : DR site
- N : node. Primary node is N1, secondary node is N2.
- w : database weight
- v : vote
- a : arbiter vote

Not all scenarions are listed for 8 node clusters and example weights have been allocated.

- For a 6 node cluster with 4 database nodes and 2 sites, initial votes are as follows:

  Primary database node, nodes 2-3: 2 (1 + 1 arbiter) Secondary database nodes 4: 1 (no arbiter)

Pri N1 w:40 v:1 a:1	**Pri N2 w:30 v:1 a:1**	**DR N3 w:20 v:1 a:1**	**DR N4 w:10 v:1**	**Votes**	**System Status under scenario**
Up	Up	Up	Up	7	System is functioning normally.
Up	Up	Up	Down	6	Scenario: Loss of a Non-primary Server in the DR Site. System continues functioning normally.
Up	Up	Down	Up	6	Scenario: Loss of a Non-primary Server in the DR Site. System continues functioning normally.
Up	Down	Up	Up	6	Scenario: Loss of a Non-primary Node in the Primary Site. System continues functioning normally.
Down	Up	Up	Up	5	Scenario: Loss of the Primary Database Server. Some downtime occurs. System automatically fails over to N2.
Down	Down	Up	Up	3	Scenario: Loss of a Primary Site. Manual recovery required
Up	Up	Down	Down	4	System continues functioning normally.
Up	Down	Down	Up	3	Manual recovery required
Up	Down	Up	Down	4	System continues functioning normally.

- For an 8 node cluster with 6 database nodes and 2 sites, initial votes are as follows:

  Primary database node: 2 (1 + 1 arbiter voting member) Secondary database nodes total: 5 (no arbiter votes)

  The table here shows a representative selection of scenarios.

Pri N1 w:60 v:1 a:1	Pri N2 w:50 v:1	Pri N3 w:40 v:1	Pri N4 w:30 v:1	DR N5 w:20 v:1	DR N6 w:10 v:1	Votes	System Status under scenario
Up	Up	Up	Up	Up	Up	7	System is functioning normally.
Up	Up	Up	Down	Down	Down	4	Scenarios: Loss of a Non-primary Node in the Primary and Secondary Site. System continues functioning normally.
Up	Up	Up	Up	Down	Up	6	Scenario: Loss of a Non-primary Server in the DR Site. System continues functioning normally.
Up	Down	Up	Up	Up	Up	6	Scenario: Loss of a Non-primary Node in the Primary Site. System continues functioning normally.
Up	Down	Down	Up	Up	Up	6	Scenario: Loss of a Non-primary Node in the Primary Site. System continues functioning normally.
Down	Up	Up	Up	Up	Up	6	Scenario: Loss of the Primary Database Server. Some downtime occurs. System automatically fails over to N2.
Down	Down	Up	Up	Up	Up	4	Some downtime occurs. System automatically fails over to N3.
Down	Down	Down	Up	Up	Up	3	Manual recovery required
Down	Down	Down	Down	Up	Up	2	Scenario: Loss of a Primary Site. Manual recovery required
Up	Up	Down	Up	Up	Up	6	Scenario: Loss of a Non-primary Node in the Primary Site. System continues functioning normally.
Up	Up	Down	Down	Up	Up	5	Scenario: Loss of a Non-primary Node in the Primary Site. System continues functioning normally.
Up	Up	Down	Down	Down	Up	4	Scenarios: Loss of a Non-primary Node in the Primary and Secondary Site. System continues functioning normally.
Up	Up	Down	Down	Down	Down	3	Manual recovery required
Up	Down	Up	Down	Down	Down	3	Manual recovery required

As the represenative table above shows, the 8 node status and scenarios are similar for a number of permutations of nodes. For example, the failure of a single node N2, N3 or N4 results in the same scenario:

- Scenario: Loss of a Non-primary Node in the Primary Site. System continues functioning normally.

The list below shows individual nodes (N1 to N6) and groups of nodes that will result in the same failover scenario.

Upon recovery, there is typically a delay of 10-20 minutes in the continuance of transaction processing.

- N2, N3, N4
- N5, N6

- N2+N3, N2+N4, N3+N4
- N1+N2+N3, N1+N2+N4, N1+N3+N4
- N1+N5, N1+N6
- N2+N5, N2+N6, N3+N5, N3+N6, N4+N5, N4+N6
- N2+N3+N4
- N2+N3+N5, N2+N3+N6, N2+N4+N5, N2+N4+N6, N3+N4+N5, N3+N4+N6
- N5+N6

A failure in other groupings will require a manual recovery, for example, in such groups as:

- N1+N2+N3, N1+N2+N4, N1+N2+N5, N1+N2+N6, N1+N3+N4, N1+N3+N5, N1+N3+N6, N1+N4+N5, N1+N4+N6, N1+N5+N6
- N2+N3+N4+N5, N2+N3+N4+N6, N3+N4+N5+N6
- N1+N2+N3+N4, N1+N2+N3+N5, N1+N2+N3+N6, N1+N3+N4+N5, N1+N3+N4+N6, N1+N4+N5+N6
- N1+N2+N3+N4+N5, N1+N2+N3+N4+N6

16.6.2. DR Failover and Recovery Scenarios

A number of failover scenarios and recovery steps are shown. In each case, a node topology is assumed: 6 or 8 node clusters in 2 sites - primary and Disaster Recovery (DR). A node failure scenario is indicated and a set of recovery steps are provided.

The following scenarios that are covered:

- Power off of a node
- Loss of a non-primary node in the Primary site
- Loss of a non-primary server in the DR site
- Loss of the Primary Database Server
- Loss of a Primary Site
- Loss of a DR Site

For the scenarios below, the following procedures and definitions apply:

- In the event of a network failure or a temporary network outage affecting a single a node, the node will be inaccessible and the cluster will respond in the same way as if the node had failed. If network connectivity is then restored, no action is required, because the node will again start communicating with the other nodes in the cluster, provided no changes were made to that node during the outage window.
- In a clustered deployment, the datacentre would typically be two different datacentres, for example “Virginia” and “Seattle”. These can be thought of as a primary site and a DR (Disaster Recovery) site in case of a failure in the primary site. These two datacentres can exist on the same physical hardware, so the separation of the cluster is into two sets of three nodes.

  When datacentres are defined during installation, the nodes of a cluster may or may not be in the same physical location. The cluster is designed to communicate across all nodes, regardless of their physical location.

- During recovery, the command **cluster provision** must be run every time a node is deleted from or added to a cluster, even if it is a replacement node. It is recommended that this step is run in a terminal opened with the `tmux` command. See: *Using the tmux command*.
- During recovery and installation, the command **cluster prepnode** must be run on every node.
- During recovery of 8 node clusters, database weights should be deleted and added again.

16.6.3. Scenario: Power Off and On of a Node

The scenario and recovery steps apply to Unified and Proxy nodes.

Node powered off

- Secondary nodes assume primary
- There is no cluster downtime and normal operations continue where the cluster is processing requests and transactions are committed successfully up to the point where a node is powered off.
- At this point, *all* transactions that are currently in flight at the node are lost and will not recover. The lost transactions have to be rerun.
- The lost transactions have to be replayed or rerun.

  Bulk load transactions cannot be replayed and have to be rerun. Before resubmitting a failed Bulk load job, carry out the following command on the primary node CLI in order to manually clear each failure transaction that still has a Processing status *after a service restart*. Use the command:

  voss finalize_transaction <Trans ID>

  The failed transaction status then changes from Processing to Fail. With the node still powered off, replaying the failed transactions is successful

Recovery steps if the node is powered off:

1. Power up the node. The node re-syncs. Run the **database config** command to verify the state of the database members. A typical output of the command would be:

```
$ database config
    date: 2017-04-25T09:50:34Z
    heartbeatIntervalMillis: 2000
    members:
        172.29.21.41:27020:
            priority: 60.0
            stateStr: PRIMARY
            storageEngine: WiredTiger
        172.29.21.41:27030:
            priority: 1.0
            stateStr: ARBITER
            storageEngine: WiredTiger
        172.29.21.42:27020:
            priority: 50.0
            stateStr: SECONDARY
            storageEngine: WiredTiger
        172.29.21.43:27020:
            priority: 40.0
            stateStr: SECONDARY
```

(continues on next page)

(continued from previous page)

```
            storageEngine: WiredTiger
        172.29.21.44:27020:
            priority: 30.0
            stateStr: SECONDARY
            storageEngine: WiredTiger
        172.29.21.45:27020:
            priority: 20.0
            stateStr: SECONDARY
            storageEngine: WiredTiger
        172.29.21.46:27020:
            priority: 10.0
            stateStr: SECONDARY
            storageEngine: WiredTiger
    myState: 1
    ok: 1.0
    set: DEVICEAPI
    term: 38
```

Note that `storageEngine` will show as `WiredTiger` after the database engine upgrade to Wired Tiger when upgrading to VOSS-4-UC v17.4. Otherwise, the value is `MMAPv1`.

In other words, the database should not for example be any of: `STARTUP`, `STARTUP2` or `RECOVERING`. Note however that is is sometimes expected that nodes are recovering or in startup, but then should change to a normal state after a period of time (depending on how far out of sync those members are).

A file system check may take place.

2. If a replacement node is not on standby, rebuild steps such as boot up, adding to cluster, setting database weight and re-provisioning may take 200-300 minutes, depending on hardware specifications.

   It is recommended that standby nodes are available to be used for faster recovery.

Note: Upon cluster provision failure at any of the proxy nodes during provisioning, the following steps illustrate the cluster provisioning:

1. Run **database config** and check if nodes are either in STARTUP2 or SECONDARY or PRIMARY states with correct arbiter placement.
2. Login to web proxy on both primary and secondary site and add a web weight using **web weight add <ip>:443 1** for all those nodes that you want to provide a web weight of 1 on the respective proxies.
3. Run **cluster provision** to mitigate the failure (it is recommended that this step is run in a terminal opened with the `tmux` command). See: *Using the tmux command*.
4. Run **cluster run all app status** to check if all the services are up and running after cluster provisioning completes.

Note: If the existing nodes in the cluster do not see the new incoming cluster after **cluster add**, try the following steps:

1. Run **cluster del <ip>** from the primary node, <ip> being the IP of the new incoming node.
2. Run **database weight del <ip>** from the primary node, <ip> being the IP of the new incoming node.
3. Log into any secondary node (non primary unified node) and run **cluster add <ip>** ,<ip> being the IP of the new incoming node.

4. Run **database weight add <ip> <weight>** from the same session, <ip> being the IP of the new incoming node.
5. Use **cluster run database cluster list** to check if all nodes see the new incoming nodes inside the cluster.

16.6.4. Scenario: Loss of a Non-primary Node in the Primary Site

- The administrator deployed the cluster into a Primary and DR site.
- The cluster is deployed following the Installation Guide.
- The example here is a typical cluster deployment of 6 nodes, where 4 nodes are database servers and 2 nodes are proxy servers.

  However, this scenario also applies to a cluster deployment of 8 nodes: 6 database servers and 2 proxy servers. In the case where more than one non-primary node is lost on the Primary site, the relevant recovery steps are repeated.

  The design is preferably split over 2 physical data centers.

```
Data Centre: jhb
             application : AS01[172.29.42.100]
                           AS02[172.29.42.101]

             webproxy :    PS01[172.29.42.102]
                           AS01[172.29.42.100]
                           AS02[172.29.42.101]

             database :    AS01[172.29.42.100]
                           AS02[172.29.42.101]

Data Centre: cpt
             application : AS03[172.29.21.100]
                           AS04[172.29.21.101]

             webproxy :   PS02[172.29.21.102]
                          AS03[172.29.21.100]
                          AS04[172.29.21.101]

             database :   AS03[172.29.21.100]
                          AS04[172.29.21.101]
```

Node Failure

- Normal operations continue where the cluster is processing requests and transactions are committed successfully up to the point where a loss of a non-primary node is experienced. In this 6-node example, `AS02[172.29.42.101]` failed while transactions were running.
- Examine the cluster status running **cluster status** to determine the failed state:

```
platform@AS01:~$ cluster status
```

(continues on next page)

(continued from previous page)

```
Data Centre: unknown
application : unknown_172.29.42.101[172.29.42.101] (not responding)

webproxy : unknown_172.29.42.101[172.29.42.101] (not responding)

database : unknown_172.29.42.101[172.29.42.101] (not responding)


Data Centre: jhb
application : AS01[172.29.42.100]

webproxy : PS01[172.29.42.102]
           AS01[172.29.42.100]

database : AS01[172.29.42.100]


Data Centre: cpt
application : AS03[172.29.21.100]
              AS04[172.29.21.101]

webproxy : PS02[172.29.21.102]
           AS03[172.29.21.100]
           AS04[172.29.21.101]

database : AS03[172.29.21.100]
           AS04[172.29.21.101]
```

- At this point, *all* transactions that are currently in flight are lost and will not recover.
- The lost transactions have to be replayed or rerun.

  Bulk load transactions cannot be replayed and have to be rerun. Before resubmitting a failed Bulk load job, carry out the following command on the primary node CLI in order to manually clear each failure transaction that still has a Processing status *after a service restart*. Use the command:

  voss finalize_transaction <Trans ID>

  The failed transaction status then changes from Processing to Fail.
- With the database server `AS02[172.29.42.101]` still down, replaying the failed transactions are successful.

Recovery Steps if the server that is lost, is unrecoverable:

1. A new unified node needs to be deployed. Ensure the server name, IP information and data centre name is the same as on the server that was lost.
2. Delete the failed node database weight (**database weight del <ip>**), for example **database weight del 172.29.42.101**
3. Run **cluster del 172.29.42.101**, because this server no longer exists. Power off the deleted node, or disable its Network Interface Card.
4. Run **cluster provision** on the cluster *without* the node to be added and then create the new unified node - see: *Platform install OVA on a VM*.
5. Switch on the newly installed server.

6. If the node will be a unified or web proxy node, run **cluster prepnode** on it.
7. From the primary unified node, run **cluster add <ip>**, with the IP address of the new unified server to add it to the existing cluster.
8. Add database weights so that the weights distributed throughout the cluster
   - Delete all database weights in the cluster. On a selected unified node, *for each unified node IP*, run **database weight del <IP>**.
   - Re-add all database weights in the cluster. *On each unified node*, for each unified node IP, run **database weight add <IP> <weight>**
   - Check weights - either individually for each node, or for the cluster by using the command:

     cluster run application database weight list

     Make sure all application nodes show correct weights.
9. Run **cluster provision primary <ip of current primary>** to join the new unified node to the cluster communications. It is recommended that this step is run in a terminal opened with the `tmux` command.
10. If an OVA file was not available for your current release and you used the most recent release OVA for which there is an upgrade path to your release to create the new unified node, *re-apply* the Delta Bundle upgrade to the cluster. See the upgrade document for your release.

    Note that the new node version mismatch in the cluster can be ignored, since this upgrade step aligns the versions.

Note: Upon cluster provision failure at any of the proxy nodes during provisioning, the following steps illustrate the cluster provisioning:

1. Run **database config** and check if nodes are either in STARTUP2 or SECONDARY or PRIMARY states with correct arbiter placement.
2. Login to web proxy on both primary and secondary site and add a web weight using **web weight add <ip>:443 1** for all those nodes that you want to provide a web weight of 1 on the respective proxies.
3. Run **cluster provision** to mitigate the failure. It is recommended that this step is run in a terminal opened with the `tmux` command.
4. Run **cluster run all app status** to check if all the services are up and running after cluster provisioning completes.

Note: If the existing nodes in the cluster do not see the new incoming cluster after **cluster add**, try the following steps:

1. Run **cluster del <ip>** from the primary node, <ip> being the IP of the new incoming node.
2. Run **database weight del <ip>** from the primary node, <ip> being the IP of the new incoming node.
3. Log into any secondary node (non primary unified node) and run **cluster add <ip>** ,<ip> being the IP of the new incoming node.
4. Run **database weight add <ip> <weight>** from the same session, <ip> being the IP of the new incoming node.
5. Use **cluster run database cluster list** to check if all nodes see the new incoming nodes inside the cluster.

16.6.5. Scenario: Loss of a Non-primary Server in the DR Site

- The administrator deployed the cluster into a Primary and DR site.
- The cluster is deployed following the Installation Guide.
- The example is a cluster deployment: 6 nodes, where 4 nodes are database servers and 2 nodes are proxy servers.

  However, this scenario also applies to a cluster deployment of 8 nodes: 6 database servers and 2 proxy servers.

  The design is preferably split over 2 physical data centers.

Node Failure

- Normal operations continue where the cluster is processing requests and transactions are committed successfully up to the point where a loss of a non-primary node is experienced.

  In this 6-node example, `AS04[172.29.21.101]` failed while transactions were running.
- Examine the cluster status running **cluster status** to determine the failed state:

```
Data Centre: unknown
             application : unknown_172.29.21.101[172.29.21.101] (not responding)

             webproxy : unknown_172.29.21.101[172.29.21.101] (not responding)

             database : unknown_172.29.21.101[172.29.21.101] (not responding)


Data Centre: jhb
               Application : AS01[172.29.42.100]
                             AS02[172.29.42.101]

               webproxy :    PS01[172.29.42.102]
                             AS01[172.29.42.100]
                             AS02[172.29.42.101]

               database :    AS01[172.29.42.100]
                             AS02[172.29.42.101]

Data Centre: cpt
               application : AS03[172.29.21.100]


               webproxy :   PS02[172.29.21.102]
                            AS03[172.29.21.100]


               database :   AS03[172.29.21.100]
```

- At this point, *all* transactions that are currently in flight are lost and will not recover.
- The lost transactions have to be replayed or rerun.

  Bulk load transactions cannot be replayed and have to be rerun. Before resubmitting a failed Bulk load job, carry out the following command on the primary node CLI in order to manually clear each failure

transaction that still has a Processing status *after a service restart*. Use the command:

voss finalize_transaction <Trans ID>

The failed transaction status then changes from Processing to Fail.

- With the database server `AS04[172.29.21.101]` still down, replaying the failed transactions are successful.

Recovery Steps if the server that is lost, is unrecoverable:

1. A new unified node needs to be deployed. Ensure the server name, IP information and datacentre name is the same as on the server that was lost.
2. Delete the failed node database weight (**database weight del <ip>**), for example **database weight del 172.29.21.101**
3. Run **cluster del 172.29.21.101**, because this server no longer exists. Power off the deleted node, or disable its Network Interface Card.
4. Run **cluster provision** on the cluster *without* the node to be added and then create the new unified node - see: *Platform install OVA on a VM* and switch on the newly installed node.
5. If the node will be a unified or web proxy node, run **cluster prepnode** on it.
6. From the primary unified node, run **cluster add <ip>**, with the IP address of the new unified node to add it to the existing cluster.
7. Add database weights so that the weights distributed throughout the cluster
   - Delete all database weights in the cluster. On a selected unified node, *for each unified node IP*, run **database weight del <IP>**.
   - Re-add all database weights in the cluster. *On each unified node*, for each unified node IP, run **database weight add <IP> <weight>**
   - Check weights - either individually for each node, or for the cluster by using the command:

     cluster run application database weight list

     Make sure all application nodes show correct weights.
8. Run **cluster provision primary <IP of current primary>** to join the new unified node to the cluster communications. It is recommended that this step is run in a terminal opened with the `tmux` command.
9. If an OVA file was not available for your current release and you used the most recent release OVA for which there is an upgrade path to your release to create the new unified node, *re-apply* the Delta Bundle upgrade to the cluster. See the upgrade document for your release.

   Note that the new node version mismatch in the cluster can be ignored, since this upgrade step aligns the versions.
10. If an Active/Passive configuration was enabled prior to failover, this should be reconfigured by logging in on the nodes on the DR site and running the command **voss workers 0**.

Note: Upon cluster provision failure at any of the proxy nodes during provisioning, the following steps illustrate the cluster provisioning:

1. Run **database config** and check if nodes are either in STARTUP2 or SECONDARY or PRIMARY states with correct arbiter placement.
2. Login to web proxy on both primary and secondary site and add a web weight using **web weight add <ip>:443 1** for all those nodes that you want to provide a web weight of 1 on the respective proxies.

3. Run **cluster provision** to mitigate the failure. It is recommended that this step is run in a terminal opened with the `tmux` command.
4. Run **cluster run all app status** to check if all the services are up and running after cluster provisioning completes.

Note: If the existing nodes in the cluster do not see the new incoming cluster after **cluster add**, try the following steps:

1. Run **cluster del <ip>** from the primary node, <ip> being the IP of the new incoming node.
2. Run **database weight del <ip>** from the primary node, <ip> being the IP of the new incoming node.
3. Log into any secondary node (non primary unified node) and run **cluster add <ip>** ,<ip> being the IP of the new incoming node.
4. Run **database weight add <ip> <weight>** from the same session, <ip> being the IP of the new incoming node.
5. Use **cluster run database cluster list** to check if all nodes see the new incoming nodes inside the cluster.

16.6.6. Scenario: Loss of the Primary Database Server

- The administrator deployed the cluster into a Primary and DR site.
- The cluster is deployed following the Installation Guide.
- The example is a typical cluster deployment: 6 nodes, where 4 nodes are database servers and 2 nodes are proxy servers.

  However, this scenario also applies to a cluster deployment of 8 nodes: 6 database servers and 2 proxy servers. If non-primary database servers are also lost on the primary or DR site, then also follow the recovery steps for these nodes.

  The design is preferably split over 2 physical data centers.

Node Failure

- Normal operations continue where the cluster is processing requests and transactions are committed successfully up to the point where a loss of a primary database server is experienced. In this scenario `AS01[172.29.42.100]` failed while transactions were running.
- Examine the cluster status running **cluster status** to determine the failed state:

```
Data Centre: unknown
        application : unknown_172.29.42.100[172.29.42.100] (not responding)

        webproxy : unknown_172.29.42.100[172.29.42.100] (not responding)

        database : unknown_172.29.42.100[172.29.42.100] (not responding)


Data Centre: jhb
        application : AS02[172.29.42.101]
```

(continues on next page)

(continued from previous page)

```
        webproxy :    PS01[172.29.42.102]
                      AS02[172.29.42.101]

        database :    AS02[172.29.42.101]

Data Centre: cpt
        application : AS03[172.29.21.100]
                      AS04[172.29.21.101]

         webproxy :   PS02[172.29.21.102]
                      AS03[172.29.21.100]
                      AS04[172.29.21.101]

         database :   AS03[172.29.21.100]
                      AS04[172.29.21.101]
```

- Some downtime occurs. This can be take up to 15 minutes. To speed up recovery, restart the services: **cluster run all app start**.
- The loss of the Primary database server will cause an election and the node with the highest weighting still running will become primary.
- Check the weights set in the cluster configuration: **database weight list**

```
platform@AS01:~$ database weight list
    172.29.21.100:
        weight: 10
    172.29.21.101:
        weight: 20
    172.29.42.100:
        weight: 50
    172.29.42.101:
        weight: 40
```

- The primary node 172.29.42.100 failed and therefore node 172.29.42.101 will become the primary node after election.
- To find the primary database, run **database primary**.

```
platform@AS02:~$ database primary
172.29.42.101
```

- At this point *all* transactions that are currently in flight are lost and will not recover.
- The lost transactions have to be replayed or rerun.

  Bulk load transactions cannot be replayed and have to be rerun. Before resubmitting a failed Bulk load job, carry out the following command on the primary node CLI in order to manually clear each failure transaction that still has a Processing status *after a service restart*. Use the command:

  voss finalize_transaction <Trans ID>

  The failed transaction status then changes from Processing to Fail.

- With the database server `AS01[172.29.42.100]` still down, replaying the failed transactions is successful.

Recovery Steps if the server that is lost, is unrecoverable:

Generally, **cluster provision** must be run every time a node is deleted or added, even if it is a replacement node. It is recommended that this step is run in a terminal opened with the `tmux` command.

1. Delete its database weight (**database weight del <ip>**), in other words **database weight del 172.29.42.100**
2. Run **cluster del 172.29.42.100**, because this server no longer exists. Power off the deleted node, or disable its Network Interface Card.
3. Run **cluster provision primary 172.29.42.101** from the current primary node. It is recommended that this step is run in a terminal opened with the `tmux` command.

   This server should already have the highest weight, and its database weight can be checked with **database weight list**

   If all the database weights are deleted and provisioning is run again with **cluster provision**, the CLI message is:

   'Please select which of the database should be used as the remaining primary by running "database config", selecting a node to sync from (any node that says primary or secondary and is in a good state, i.e. not in a 'RECOVERING' or 'STARTUP' state) and rerun provisioning with "cluster provision primary <db server ip from commmand above>"'
4. A new unified node needs to be deployed. Ensure the server name, IP information and data centre name is the same as on the server that was lost.
5. Run **cluster provision** on the cluster *without* the node to be added and then create the new unified node - see: *Platform install OVA on a VM*.
6. Run **cluster prepnode** on *all* servers.
7. Run **cluster add <ip>** from the primary unified node (current), with the IP address of the new unified server to add it to the existing cluster.
8. Check the output of the commands: **cluster list** and **cluster status** from the existing node. If the new node does not show up:
   a. Run **cluster del <new node>**
   b. Rerun the add of the node on *another* existing unified node, until the node shows up in **cluster list** and **cluster status**.
   c. Verify that the node shows up from all existing nodes. The recovery process may be time consuming.
9. Delete all database weights in the cluster. On a selected unified node, *for each unified node IP*, run **database weight del <IP>**.
10. Re-add all database weights in the cluster. *On each unified node*, for each unified node IP, run **database weight add <IP> <weight>**, considering the following:
    - *For the new unified node*, add a database weight lower than that of the weight of the current primary if this will be a secondary, or higher if this will be the new primary.
    - If the lost primary unified node release version is 18.1-V4UC-Patch-Bundle-03b and if it will be the new primary, first set its weight lower than the current primary and re-apply the patch on it:

      app install media/18.1-V4UC-Patch-Bundle-03b.script –force

When done, check the database weights - either individually for each node, or for the cluster by using the command:

cluster run application database weight list

Make sure all application nodes show correct weights.

11. Make sure the new node is part of the cluster (run **cluster list**) and run **cluster provision primary 172.29.42.101** *from the current primary*. It is recommended that this step is run in a terminal opened with the `tmux` command.

    During the provision process, the role of primary will then be transferred from the current primary to the node with the highest weight. The role transfer may take a significant amount of time, depending on the database size.

    During the process, typing **app status** from the new primary node will still show the database as `not provisioned`:

```
mongodb v11.5.3 (2018-07-01 14:35)
   |-arbiter            running
   |-database           running (not provisioned)
```

    To check the progress of the transfer, the database log can be checked. Type **log follow mongodb/mongodb/mongodb.log**. When the transfer is complete, an entry will show `sync done` as in the example below:

```
2018-07-09T14:09:48.639986+00:00 un1 mongod.27020[129593]: [initial sync-0] initial␣
↪sync done; took 5821s.
```

    While the primary role transfer is in progress, the system can be used, but bulk database operations should not be carried out, because the sync may fall too far behind to complete.

12. If an OVA file was not available for your current release and you used the most recent release OVA for which there is an upgrade path to your release to create the new unified node, *re-apply* the Delta Bundle upgrade to the cluster. See the upgrade document for your release.

    Note that the new node version mismatch in the cluster can be ignored, since this upgrade step aligns the versions.

Note: Upon cluster provision failure at any of the proxy nodes during provisioning, the following steps illustrate the cluster provisioning:

1. Run **database config** and check if nodes are either in STARTUP2 or SECONDARY or PRIMARY states with correct arbiter placement.
2. Login to web proxy on both primary and secondary site and add a web weight using **web weight add <ip>:443 1** for all those nodes that you want to provide a web weight of 1 on the respective proxies.
3. Run **cluster provision** to mitigate the failure.
4. Run **cluster run all app status** to check if all the services are up and running after cluster provisioning completes.

Note: If the existing nodes in the cluster do not see the new incoming cluster after **cluster add**, try the following steps:

1. Run **cluster del <ip>** from the primary node, <ip> being the IP of the new incoming node.

2. Delete all database weights. Run **database weight del <ip>** from the primary node, <ip> being the IP of the nodes, including the new incoming node.
3. Log into any secondary node (non primary unified node) and run **cluster add <ip>** ,<ip> being the IP of the new incoming node.
4. Re-add all database weights. Run **database weight add <ip> <weight>** from the same session, <ip> being the IP of the nodes, including the new incoming node.
5. Use **cluster run database cluster list** to check if all nodes see the new incoming nodes inside the cluster.

16.6.7. Scenario: Loss of a Primary Site

- The administrator deployed the cluster into a Primary and DR site.
- The cluster is deployed following the Installation Guide.
- The example is a typical cluster deployment: 6 nodes, where 4 nodes are database servers and 2 nodes are proxy servers.

  However, this scenario also applies to a cluster deployment of 8 nodes: 6 database servers and 2 proxy servers.

  The design is preferably split over 2 physical data centers.
- The cluster might also be in two geographically dispersed areas. The cluster has to be installed in two different site names or data center names. In this scenario, a portion of the cluster is in Johannesburg and the other is in Cape Town, South Africa:

```
Data Centre: jhb
        application : AS01[172.29.42.100]
                      AS02[172.29.42.101]

        webproxy :    AS01[172.29.42.100]
                      AS02[172.29.42.101]
                      PS01[172.29.42.102]

        database :    AS01[172.29.42.100]
                      AS02[172.29.42.101]

Data Centre: cpt
        application : AS03[172.29.21.100]
                      AS04[172.29.21.101]

         webproxy :   PS02[172.29.21.102]
                      AS03[172.29.21.100]
                      AS04[172.29.21.101]

         database :   AS03[172.29.21.100]
                      AS04[172.29.21.101]
```

Primary site failure

- Normal operations continue where the cluster is processing requests and transactions are committed successfully up to the point where a loss of a Primary site is experienced. In this scenario, `AS01[172.29.42.100]`, `AS02[172.29.42.101]` and `PS01[172.29.42.102]` failed while transactions were running.
- At this point, *all* transactions that are currently in flight are lost and will not recover.
- The lost transactions have to be replayed or rerun.

  Bulk load transactions cannot be replayed and have to be rerun. Before resubmitting a failed Bulk load job, carry out the following command on the primary node CLI in order to manually clear each failure transaction that still has a Processing status *after a service restart*. Use the command:

  voss finalize_transaction <Trans ID>

  The failed transaction status then changes from Processing to Fail.
- Examine the cluster status by running **cluster status** to determine the failed state:

```
Data Centre: unknown
        application : unknown_172.29.42.100[172.29.42.100] (not responding)
                      unknown_172.29.42.101[172.29.42.101] (not responding)

        webproxy :    unknown_172.29.42.100[172.29.42.100] (not responding)
                      unknown_172.29.42.101[172.29.42.101] (not responding)
                      unknown_172.29.42.102[172.29.42.102] (not responding)

        database :    unknown_172.29.42.100[172.29.42.100] (not responding)
                      unknown_172.29.42.101[172.29.42.101] (not responding)


Data Centre: jhb
             application :


             webproxy :


             database :


Data Centre: cpt
           application :   AS03[172.29.21.100]
                           AS04[172.29.21.101]

             webproxy :    PS02[172.29.21.102]
                           AS03[172.29.21.100]
                           AS04[172.29.21.101]

             database :    AS03[172.29.21.100]
                           AS04[172.29.21.101]
```

- The cluster will be not be operational and manual intervention is needed to recover if a continued flow of transactions is required with a minimum of downtime.
- If it was possible to recover the lost nodes within a reasonable time frame, the cluster will recover automatically if the nodes that were down were brought back into the cluster array successfully.
- To recover the lost nodes and if they are unrecoverable, carry out the following recovery steps.

Recovery Steps (two options):

Commands should be run on an operational unified node from the DR site. During the recovery of clusters, database weights should be deleted and added again.

1. Delete the failed node database weights from the cluster: **database weight del <ip>**
2. Run **cluster del <ip>** to remove the nodes at the failed primary site. Power off the deleted node, or disable its Network Interface Card.
3. At this point, you have two options:
    a. Option A: provision half the cluster for a faster uptime of your DR site. Only the DR site will then be operational after the provision. You can also optionally add unified nodes to this cluster.
    b. Option B: bring the full cluster back up at both the DR site and Primary site. You need to redeploy the Primary site nodes.
4. Option A: provision half the cluster or optionally adding 2 more unified nodes to it.
    a. If you choose to add 2 more unified nodes to optionally create a cluster with 4 unified nodes, deploy the new nodes as follows.
        i. Run **cluster provision** on the cluster *without* the node to be added and then create the new unified node - see: *Platform install OVA on a VM*.
        ii. Run **cluster prepnode** on all new nodes.
        iii. From a running unified node, run **cluster add <ip>**, with the IP address of the new unified node to add it to the existing cluster.
        iv. Add the database weights nodes in the cluster at the DR site.
            - Delete all database weights in the cluster of the DR site. On a selected unified node, *for each unified node IP*, run **database weight del <IP>**.
            - Re-add all database weights in the cluster of the DR site. *On each unified node*, for each unified node IP, run **database weight add <IP> <weight>**, considering the following:

              For the new unified node, add a database weight lower than that of the weight of the current primary if this will be a secondary, or higher if this will be the new primary.
    b. Run **cluster provision primary <ip>** (current primary IP) It is recommended that this step is run in a terminal opened with the `tmux` command.
    c. If an OVA file was not available for your current release and you used the most recent release OVA for which there is an upgrade path to your release to create the new unified node, *re-apply* the Delta Bundle upgrade to the cluster. See the upgrade document for your release.

       Note that the new node version mismatch in the cluster can be ignored, since this upgrade step aligns the versions.
    d. Check all services, nodes and weights - either individually for each node, or for the cluster by using the commands:
        - **cluster run all app status** (make sure no services are stopped/broken - the message 'suspended waiting for mongo' is normal on the fresh unifieds)
        - **cluster run application cluster list** (make sure all application nodes show 3 or 5 nodes)
        - **cluster run application database weight list** (make sure all application nodes show correct weights)
5. Option B: bring the full cluster back up at both the DR site and Primary site. You need to redeploy the Primary site nodes.

a. Deploy 3 nodes: 2 as unified nodes and 1 as a proxy node. For an 8-node topology, deploy the number of Primary site unified nodes and the web proxy node that were lost.

   i. Run **cluster provision** on the cluster *without* the node to be added and then create the new unified node - see: *Platform install OVA on a VM*.

   ii. Run **cluster prepnode** on all new nodes.

   iii. Run **cluster add <ip>** from the current primary unified node, with the IP address of the new unified node to add it to the existing cluster.

   iv. Ensure the database weights are added back:

   - Delete all database weights in the cluster. On a selected unified node, *for each unified node IP*, run **database weight del <IP>**.
   - Re-add all database weights in the cluster. *On each unified node*, for each unified node IP, run **database weight add <IP> <weight>**, considering the following:

     For a new unified node, add a database weight lower than that of the weight of the current primary if this will be a secondary, or higher if this will be the new primary.

   v. Run **cluster provision primary <ip>** (current primary IP), It is recommended that this step is run in a terminal opened with the `tmux` command.

   After provisioning, the node with the largest database weight will be the primary server.

   vi. If an OVA file was not available for your current release and you used the most recent release OVA for which there is an upgrade path to your release to create the new unified node, *re-apply* the Delta Bundle upgrade to the cluster.

   Note that the new node version mismatch in the cluster can be ignored, since this upgrade step aligns the versions.

   See the upgrade document for your release.

b. Check all services, nodes and weights - either individually for each node, or for the cluster by using the commands:

   - **cluster run all app status** (make sure no services are stopped/broken - the message 'suspended waiting for mongo' is normal on the fresh unifieds)
   - **cluster run application cluster list** (make sure all application nodes show 6 nodes - or 8 nodes for an 8-node topology).
   - **cluster run application database weight list** (make sure all application nodes show correct weights)

c. Run **cluster provision primary <ip>**, where `<ip>` is *the current primary in the DR site*. It is recommended that this step is run in a terminal opened with the `tmux` command. The six node (or eight node) cluster then pulls the data from this `<ip>` into the new primary server at the Primary site.

   After provisioning, the database configuration can then be checked with **database config** to verify the primary node in the Primary site.

16.6.8. Scenario: Loss of a DR Site

- The administrator deployed the cluster into a Primary and DR site.
- The cluster is deployed following the Installation Guide.
- The example here is a cluster deployment of 6 nodes, where 4 nodes are database servers and 2 nodes are proxy servers.

  However, this scenario also applies to a cluster deployment of 8 nodes: 6 database servers and 2 proxy servers.

  The design is preferably split over 2 physical data centers.
- The cluster might also be in two geographically dispersed areas. The cluster has to be installed in two different site names or data center names. In this scenario, a portion of the cluster is in Johannesburg and the other is in Cape Town, South Africa:

```
Data Centre: jhb
        application : AS02[172.29.42.101]

        webproxy :    PS01[172.29.42.102]
                      AS02[172.29.42.101]

        database :    AS02[172.29.42.101]

Data Centre: cpt
        application : AS03[172.29.21.100]
                      AS04[172.29.21.101]

         webproxy :   PS02[172.29.21.102]
                      AS03[172.29.21.100]
                      AS04[172.29.21.101]

         database :   AS03[172.29.21.100]
                      AS04[172.29.21.101]
```

DR site failure

- Normal operations continue where the cluster is processing requests and transactions are committed successfully up to the point where a loss of a DR site is experienced. In this scenario, `AS03[172.29.21.100]`, `AS04[172.29.21.101]` and `PS02[172.29.21.100]` failed while transactions were running.
- At this point, *all* transactions that are currently in flight are lost and will not recover. The lost transactions have to be rerun.
- The lost transactions have to be replayed or rerun.

  Bulk load transactions cannot be replayed and have to be rerun. Before resubmitting a failed Bulk load job, carry out the following command on the primary node CLI in order to manually clear each failure transaction that still has a Processing status *after a service restart*. Use the command:

  voss finalize_transaction <Trans ID>

  The failed transaction status then changes from Processing to Fail.
- With the DR site still down, replaying the failed transactions is successful
- Examine the cluster status by running **cluster status** to determine the failed state:

```
Data Centre: unknown
      application : unknown_172.29.21.100[172.29.21.100] (not responding)
                          unknown_172.29.21.101[172.29.21.101] (not responding)

       webproxy : unknown_172.29.21.100[172.29.21.100] (not responding)
                  unknown_172.29.21.101[172.29.21.101] (not responding)
                  unknown_172.29.21.102[172.29.21.102] (not responding)

       database : unknown_172.29.21.100[172.29.21.100] (not responding)
                  unknown_172.29.21.101[172.29.21.101] (not responding)


Data Centre: jhb
              application : AS01[172.29.42.100]
                            AS02[172.29.42.101]

              webproxy :    PS01[172.29.42.102]
                            AS01[172.29.42.100]
                            AS02[172.29.42.101]

              database :    AS01[172.29.42.100]
                            AS02[172.29.42.101]

Data Centre: cpt
              application :


              webproxy :


              database :
```

- The cluster will be operational, but only on the Primary Site.
- You need to recover the lost nodes and if they are unrecoverable. Follow the recovery steps below.

Recovery Steps

1. Remove the database weights of the failed nodes from the cluster: **database weight del <ip>**
2. Run **cluster del <ip>** to remove the failed nodes from the existing half of the cluster. Power off the deleted node, or disable its Network Interface Card.
3. Run **cluster provision primary <ip>** before a new server is added. It is recommended that this step is run in a terminal opened with the `tmux` command.
4. Redeploy the failed DR site nodes if the nodes are unrecoverable. Deploy 3 nodes: 2 as unified nodes and 1 as a proxy node. This applies to the DR site of a 6 node deployment or 8 node deployment.
5. Run **cluster provision primary** on the cluster *without* the node to be added and then create the new unified node - see: *Platform install OVA on a VM*.
6. If a node will be a unified or web proxy node, run **cluster prepnode** on it.
7. From the primary unified node, after the redeployment, run **cluster add <ip>** with the IP address of the new unified node to add it to the existing cluster. Run **cluster list** to make sure the nodes added in cluster.

8. Add the database weights nodes in the cluster.
    - Delete all database weights in the cluster. On a selected unified node, *for each unified node IP*, run **database weight del <IP>**.
    - Re-add all database weights in the cluster. *On each unified node*, for each unified node IP, run **database weight add <IP> <weight>**
9. Check all services, nodes and weights - either individually for each node, or for the cluster by using the commands:
    - **cluster run all app status** (make sure no services are stopped/broken - the message 'suspended waiting for mongo' is normal on the fresh unifieds)
    - **cluster run application cluster list** (make sure all application nodes show 6 nodes - or 8 nodes for an 8-node topology)
    - **cluster run application database weight list** (make sure all application nodes show correct weights)
10. Run **cluster provision primary <ip>** to ensure that a primary is selected for the provisioning stage. It is recommended that this step is run in a terminal opened with the `tmux` command.

    After provisioning, the database configuration can then be checked with the command **database config**.
11. If an OVA file was not available for your current release and you used the most recent release OVA for which there is an upgrade path to your release to create the new unified node, *re-apply* the Delta Bundle upgrade to the cluster. See the upgrade document for your release.

    Note that the new node version mismatch in the cluster can be ignored, since this upgrade step aligns the versions.
12. If an Active/Passive configuration was enabled prior to failover, this should be reconfigured by logging in on the nodes on the DR site and running the command **voss workers 0**.

16.6.9. DR Failover and Recovery in a 2 Node Cluster

Important: A 2 node cluster will *not* fail over automatically.

With only two Unified nodes, with or without Web proxies, there is no High Availability. The database on the primary node is read/write, while the database on the secondary is read only.

Only redundancy is available.

- If the primary node fails, a manual delete of the primary node on the secondary and a cluster provision will be needed.
- If the secondary node fails, it needs to be replaced.

Scenario: Loss of Primary Node

- The administrator deployed the 2-node cluster.

```
$ cluster status

Data Centre: jhb
             application : AS01[172.29.42.100]
                           AS02[172.29.42.101]

             webproxy :    AS01[172.29.42.100]
                           AS02[172.29.42.101]

             database :    AS01[172.29.42.100]
                           AS02[172.29.42.101]
```

Example database weights:

```
$ database weight list
     172.29.42.100:
         weight: 20
     172.29.42.101:
         weight: 10
```

- Node Failure: in the case where the primary node is lost on the Primary site:

```
$ cluster status

Data Centre: unknown
             application : unknown_172.29.248.100[172.29.248.100] (not responding)

             webproxy : unknown_172.29.248.100[172.29.248.100] (not responding)

             database : unknown_172.29.248.100[172.29.248.100] (not responding)


Data Centre: jhb
             application : AS02[172.29.248.101]

             webproxy : AS02[172.29.248.101]

             database : AS02[172.29.248.101]
```

Recovery Steps

The primary node server is lost.

A. It is decided to fail over to the secondary node:

1. *On the secondary node*, remove the lost server from the cluster:

   cluster del 172.29.248.100

2. *On the secondary node*, run **cluster provision** (it is recommended that this step is run in a terminal opened with the `tmux` command). See: *Using the tmux command*.

   On the secondary node, check:

```
$ cluster status

Data Centre: jhb
             application : AS02[172.29.248.101]

             webproxy : AS02[172.29.248.101]

             database : AS02[172.29.248.101]
```

B. It is decided to recover the primary node:

1. *On the secondary node*, remove the lost server from the cluster:

   cluster del 172.29.248.100

2. *On the secondary node*, run **cluster provision** (it is recommended that this step is run in a terminal opened with the `tmux` command).

   On the secondary node, check:

```
$ cluster status

Data Centre: jhb
             application : AS02[172.29.248.101]

             webproxy : AS02[172.29.248.101]

             database : AS02[172.29.248.101]
```

3. Switch on the newly installed server.

   On the secondary node, add the server. Run **cluster add 172.29.42.100**.

   On either node, check:

```
$ cluster status

Data Centre: jhb
             application : AS01[172.29.42.100]
                           AS02[172.29.42.101]

             webproxy :    AS01[172.29.42.100]
                           AS02[172.29.42.101]
```

(continues on next page)

(continued from previous page)

```
            database :    AS01[172.29.42.100]
                          AS02[172.29.42.101]
```

4. Configure the primary database. *On the newly installed server*, run **cluster provision primary 172.29.42.100** (it is recommended that this step is run in a terminal opened with the `tmux` command).

   Check database configuration on both nodes, for example:

```
$ database config
    date:
        $date: 1549450382862
    heartbeatIntervalMillis: 2000
    members:
        172.29.42.100:27020:
            priority: 20.0
            stateStr: PRIMARY
            storageEngine: WiredTiger
        172.29.42.100:27030:
            priority: 1.0
            stateStr: ARBITER
            storageEngine: Unknown
        172.29.42.101:27020:
            priority: 10.0
            stateStr: SECONDARY
            storageEngine: WiredTiger
    myState: 1
    ok: 1.0
    set: DEVICEAPI
    term: 8
```

Scenario: Loss of Secondary Node - Replace

1. Remove the secondary node:

```
cluster del <secondary node IP>
```

2. Re-provision the cluster without the removed node:

```
cluster provision
```

3. Create a new secondary node: see *Platform install OVA on a VM*
4. On the newly added node, run:

```
cluster prepnode
```

5. From the primary unified node, run the command below - with the IP address of the new unified server to add it to the existing cluster.

```
cluster add <secondary node IP>
```

6. Re-provision the cluster:

```
cluster provision primary <IP of current primary>
```

16.6.10. Scenario: Loss of Full Cluster

Background

- The administrator deployed a single-node cluster or deployed the cluster into a Primary and DR site.
- The cluster is deployed following the Installation Guide.
- The example is a typical cluster deployment: 6 nodes, where 4 nodes are database servers and 2 nodes are proxy servers.

  However, this scenario also applies to a cluster deployment of 8 nodes: 6 database servers and 2 proxy servers.

  The design is preferably split over 2 physical data centers.
- The cluster might also be in two geographically dispersed areas. The cluster has to be installed in two different site names or data center names.

Full cluster failure

- In this scenario, *all* nodes failed while transactions were running.
- At this point, *all* transactions that were in flight are lost and will not recover.
- The lost transactions have to be rerun.
- The cluster will not be operational and manual intervention is needed to recover.
- To recover the cluster, carry out the Recovery Steps.

Recovery Steps

Important:

- Prerequisite: a system backup exported to a remote backup location. The backup file on the remote location would typically have a format *<timestamp>.tar.gz*. This recovery procedure will *only* succeed if you have a valid recent backup to restore.
- For details, considerations and specific commands at each step below, refer to the “Standalone (single-node cluster) Installation” or “Multinode Installation” topic in the *Installation Guide*.

1. Ensure all traces of the previous nodes have been removed from the VMware environment.
2. Deploy fresh nodes as per the original topology.
   - Check deployment topologies and hardware requirements for Multinode Cluster with Unified Nodes, in the *Architecture and Hardware Specification Guide*.

- For new node deployment, see: *Platform install OVA on a VM*.
- For the steps below, follow either the “Standalone (single-node cluster) Installation” or “Multinode Installation” topic in the *Installation Guide*:

3. Add each non-primary node to the cluster by running **cluster prepnode**.
4. From the primary node, add each node to the cluster using the **cluster add <IP address of node>** command.
5. For multi-node clusters:
   a. On the primary node, set the database weights for each database node using the **database weight add <IP address of node> <weight>** command.
   b. Restore a backup made from the highest weighted secondary database node in the original cluster.

      Follow the Import steps here: *Backup and Import to a New Environment*.

      Note: It is not necessary to run **cluster provision** again on the primary node. This action is included in the backup restore process.
6. Ensure all services are up and running:

   Run **cluster run all app status** to check if all the services are up and running after the restore completes.

Note: For multi-node clusters:

Upon cluster provision failure at any of the proxy nodes during provisioning, the following steps illustrate the cluster provisioning:

1. Run **database config** and check if nodes are either in STARTUP2 or SECONDARY or PRIMARY states with correct arbiter placement.
2. Login to web proxy on both primary and secondary site and add a web weight using **web weight add <ip>:443 1** for all those nodes that you want to provide a web weight of 1 on the respective proxies.
3. Run **cluster provision** to mitigate the failure.
4. Run **cluster run all app status** to check if all the services are up and running after cluster provisioning completes.

Note: For multi-node clusters:

If the existing nodes in the cluster do not see the new incoming cluster after **cluster add**, try the following steps:

1. Run **cluster del <ip>** from the primary node, <ip> being the IP of the new incoming node.
2. Delete all database weights. Run **database weight del <ip>** from the primary node, <ip> being the IP of the nodes, including the new incoming node.
3. Log into any secondary node (non primary unified node) and run **cluster add <ip>** ,<ip> being the IP of the new incoming node.
4. Re-add all database weights. Run **database weight add <ip> <weight>** from the same session, <ip> being the IP of the nodes, including the new incoming node.

5. Use **cluster run database cluster list** to check if all nodes see the new incoming nodes inside the cluster.

16.7. DR Failover and Recovery in a Modular Cluster Topology

16.7.1. Modular Cluster: Election of a New Primary and Failover

In the case where nodes fail, the system follows a failover procedure. For details on the failover and DR process, refer to the topics in the Platform Guide.

If the primary database is lost, the failover process involves the election of a new primary database by the remaining database nodes. Each node in a cluster is allocated a number of votes that are used in the failover election of a new primary database - the election of a running node with the highest database weight.

The database weights for a node can be seen as the `priority` value when running the **database config** command.

Note: The database weight of a node does not necessarily match its number of votes.

For voting on a modular system, arbiters are added to the database nodes (N1 and N2) on the primary site, but the secondary site database node (N3) arbiter will be unused.

```
members:
    192.168.100.4:27020:
        priority: 30.0
        stateStr: SECONDARY
        storageEngine: WiredTiger
    192.168.100.4:27030:
        priority: 30.0
        stateStr: ARBITER
        storageEngine: WiredTiger
    192.168.100.6:27020:
        priority: 40.0
        stateStr: PRIMARY
        storageEngine: WiredTiger
    192.168.100.6:27030:
        priority: 40.0
        stateStr: ARBITER
        storageEngine: WiredTiger
    192.168.100.8:27020:
        priority: 10.0
        stateStr: SECONDARY
        storageEngine: WiredTiger
```

The maximum number of votes in a cluster should not exceed 5 and arbiter votes are added to nodes to provide a total of 5 votes.

The tables below show the system status and failover for a selection of scenarios for a 6 node cluster. Also refer to the topics on the specific DR scenarios. The abbreviations used are as follows:

- Pri : Primary site
- DR : DR site
- N : node. Primary node is N1, secondary node is N2.
- w : database weight
- v : vote
- a : arbiter vote
- For example, for a 6 node cluster with 3 database nodes and 2 sites, initial votes per node are as follows:

  Primary database nodes (N1, N2): 2 (each 1 + 1 arbiter) Secondary database node (N3): 1 (no arbiter vote)

Pri N1 w:40 v:1 a:1	Pri N2 w:30 v:1 a:1	DR N3 w:10 v:1	Votes	System Status under scenario
Up	Up	Up	5	System is functioning normally.
Up	Up	Down	4	Scenario: Loss of a Non-primary Server in the DR Site. System continues functioning normally.
Up	Up	Up	5	Scenario: Loss of an Application Server. System continues functioning normally. Some transactions may hang.
Up	Down	Up	3	Scenario: Loss of a Non-primary Node in the Primary Site. System continues functioning normally.

Pri N1 w:40 v:1 a:1	Pri N2 w:30 v:1 a:1	DR N3 w:10 v:1	Votes	System Status under scenario
Down	Up	Up	3	Scenario: Loss of the Primary Database Server. Some downtime occurs. System automatically fails over to N2.
Down	Down	Up	1	Scenario: Loss of a Primary Site. Manual recovery required.
Up	Down	Down	2	Scenario: Loss of all secondary nodes. Manual recovery required.
Down	Up	Down	2	Scenario: Loss of all Primary and DR nodes. Manual recovery required.

16.7.2. DR Failover and Recovery Scenarios in a Modular Cluster

A number of failover scenarios and recovery steps are shown. In each case, a node topology is assumed: 8 node clusters in 2 sites - primary and Disaster Recovery (DR).

- The example is a typical cluster deployment: 8 nodes, where 3 nodes are database servers, 3 nodes are application nodes and 2 nodes are proxy servers.

A node failure scenario is indicated and a set of recovery steps are provided.

The following scenarios that are covered:

- Power Off and On of a Node in a Modular Cluster
- Loss of an app node: Modular Cluster
- Loss of the Primary Database Server in a Modular Cluster
- Loss of a non-primary database: Modular Cluster
- Loss of a Primary Site in a Modular Cluster
- Loss of a DR Site in a Modular Cluster
- Loss of Full Cluster in a Modular Cluster

Background

For the scenarios below, the following procedures and definitions apply:

- In the event of a network failure or a temporary network outage affecting a single a node, the node will be inaccessible and the cluster will respond in the same way as if the node had failed. If network connectivity is then restored, no action is required, because the node will again start communicating with the other nodes in the cluster, provided no changes were made to that node during the outage window.
- In a clustered deployment, the data center would typically be two different data centers, for example “Virginia” and “Seattle”. These can be thought of as a primary site and a DR (Disaster Recovery) site in case of a failure in the primary site. These two data centers can exist on the same physical hardware, so the separation of the cluster is into two sets of three nodes.

  When data centers are defined during installation, the nodes of a cluster may or may not be in the same physical location. The cluster is designed to communicate across all nodes, regardless of their physical location.
- During recovery, the command **cluster provision** must be run every time a node is deleted from or added to a cluster, even if it is a replacement node. It is recommended that this step is run in a terminal opened with the `tmux` command. See: *Using the tmux command*.
- During recovery and installation, the command **cluster prepnode** must be run on every node.
- During recovery of 8 node clusters, database weights should be deleted and added again.

16.7.3. Scenario: Power Off and On of a Node in a Modular Cluster

The scenario and recovery steps apply to database, application and Proxy nodes.

Node powered off

- Secondary database node assumes primary
- There is no cluster downtime and normal operations continue where the cluster is processing requests and transactions are committed successfully up to the point where a node is powered off.
- At this point, *all* transactions that are currently in flight at the node are lost and will not recover. The lost transactions have to be rerun.
- The lost transactions have to be replayed or rerun.

  Bulk load transactions cannot be replayed and have to be rerun. Before resubmitting a failed Bulk load job, carry out the following command on an application node CLI in order to manually clear each failure transaction that still has a Processing status *after a service restart*. Use the command:

  voss finalize_transaction <Trans ID>

  The failed transaction status then changes from Processing to Fail. With the node still powered off, replaying the failed transactions is successful

Recovery steps if the node is powered off:

1. Power up the node. The node re-syncs.

   For a database node, run the **database config** command to verify the state of the database members. A typical output of the command would be:

```
$ database config
    date: 2017-04-25T09:50:34Z
    heartbeatIntervalMillis: 2000
    members:
        172.29.21.41:27020:
            priority: 60.0
            stateStr: PRIMARY
            storageEngine: WiredTiger
        172.29.21.41:27030:
            priority: 1.0
            stateStr: ARBITER
            storageEngine: WiredTiger
        172.29.21.42:27020:
            priority: 50.0
            stateStr: SECONDARY
            storageEngine: WiredTiger
        172.29.21.43:27020:
            priority: 40.0
            stateStr: SECONDARY
            storageEngine: WiredTiger
        172.29.21.44:27020:
            priority: 30.0
            stateStr: SECONDARY
            storageEngine: WiredTiger
        172.29.21.45:27020:
            priority: 20.0
```

(continues on next page)

(continued from previous page)

```
            stateStr: SECONDARY
            storageEngine: WiredTiger
        172.29.21.46:27020:
            priority: 10.0
            stateStr: SECONDARY
            storageEngine: WiredTiger
    myState: 1
    ok: 1.0
    set: DEVICEAPI
    term: 38
```

Note that `storageEngine` will show as `WiredTiger` after the database engine upgrade to Wired Tiger when upgrading VOSS-4-UC 17.4. Otherwise, the value is `MMAPv1`.

In other words, the database should not for example be any of: `STARTUP`, `STARTUP2` or `RECOVERING`. Note however that is is sometimes expected that nodes are recovering or in startup, but then should change to a normal state after a period of time (depending on how far out of sync those members are).

A file system check may take place.

2. If a replacement node is not on standby, rebuild steps such as boot up, adding to cluster, setting database weight and re-provisioning may take 200-300 minutes, depending on hardware specifications.

   It is recommended that standby nodes are available to be used for faster recovery.

Note: Upon cluster provision failure at any of the proxy nodes during provisioning, the following steps illustrate the cluster provisioning:

1. Run **database config** and check if nodes are either in STARTUP2 or SECONDARY or PRIMARY states with correct arbiter placement.
2. Login to web proxy on both primary and secondary site and add a web weight using **web weight add <ip>:443 1** for all those nodes that you want to provide a web weight of 1 on the respective proxies.
3. Run **cluster provision** to mitigate the failure (it is recommended that this step is run in a terminal opened with the `tmux` command). See: *Using the tmux command*.
4. Run **cluster run all app status** to check if all the services are up and running after cluster provisioning completes.

Note: If the existing nodes in the cluster do not see the new incoming cluster after **cluster add**, try the following steps:

1. Run **cluster del <ip>** from the primary database node, <ip> being the IP of the new incoming node.
2. For database nodes, run **database weight del <ip>** from the primary database node, <ip> being the IP of the new incoming node.
3. Log into primary database node and run **cluster add <ip>** ,<ip> being the IP of the new incoming node.
4. For database nodes, run **database weight add <ip> <weight>** from the same session, <ip> being the IP of the new incoming node.
5. Use **cluster run database cluster list** to check if all nodes see the new incoming nodes inside the cluster.

16.7.4. Scenario: Loss of an app node: Modular Cluster

- The administrator deployed the cluster into a Primary and DR site.
- The cluster is deployed following the Installation Guide.
- The example is a typical cluster deployment: 8 nodes, where 3 nodes are database servers, 3 nodes are application nodes and 2 nodes are proxy servers.

  The design is preferably split over 2 physical data centers.

Application Node Failure

- Normal operations continue where the cluster is processing requests and transactions are committed successfully up to the point where a loss of a app node is experienced. In this 8-node example, `AS02[172.29.42.101]` failed while transactions were running.
- Examine the cluster status running **cluster status** to determine the failed state:

```
Data Centre: unknown

            application : unknown_172.29.42.101[172.29.42.101] (not responding)


Data Centre: jhb
              application : AS01[172.29.42.100]

              webproxy :    PS01[172.29.42.102]

              database :    DB01[172.29.42.103]
                            DB02[172.29.42.104]

Data Centre: cpt
              application : AS03[172.29.21.100]

              webproxy :   PS02[172.29.21.102]

              database :   DB03[172.29.21.101]
```

- At this point, *all* transactions that are currently in flight are lost and will not recover.
- The lost transactions have to be replayed or rerun.

  Bulk load transactions cannot be replayed and have to be rerun. Before resubmitting a failed Bulk load job, carry out the following command on an application node in order to manually clear each failure transaction that still has a Processing status *after a service restart*. Use the command:

  voss finalize_transaction <Trans ID>

  The failed transaction status then changes from Processing to Fail.

Recovery steps

If the server that is lost, is unrecoverable:

1. A new app node needs to be deployed. Ensure the server name, IP information and data centre name is the same as on the server that was lost.
2. Run **cluster del <IP of lost app node>**, because this server no longer exists. Power off the deleted node, or disable its Network Interface Card.
3. Run **cluster provision** on the cluster *without* the node to be added and then create the *new app node* at the *required data center* - see: *Platform install OVA on a VM*.
4. Switch on the newly installed server.
5. Run **cluster prepnode** on the new app node.
6. From the primary database node, run **cluster add <ip>**, with the IP address of the new app node to add it to the existing cluster.
7. From the primary database node, run **cluster provision** to join the new app node to the cluster communications. It is recommended that this step is run in a terminal opened with the `tmux` command.
8. If an OVA file was not available for your current release and you used the most recent release OVA for which there is an upgrade path to your release to create the new unified node, *re-apply* the Delta Bundle upgrade to the cluster.

   Important: Re-apply any patches and services (for example, Phone Based Registration) to this node that were added after the initial Delta Bundle upgrade.

   Note that the new node version mismatch in the cluster can be ignored, since this upgrade step aligns the versions.
9. On the new app node, check the number of queues using **voss queues** and if the number is *less than 2*, set the queues to 2 with **voss queues 2**.

   Note: Applications are reconfigured and the `voss-queue` process is restarted.
10. If the app node was replaced on the DR site and an Active/Passive configuration was enabled prior to failover, this should be reconfigured by logging in on the nodes on the DR site and running the command **voss workers 0**.

    See the upgrade document for your release.

Note: Upon cluster provision failure at any of the proxy nodes during provisioning, the following steps illustrate the cluster provisioning:

1. Run **database config** and check if nodes are either in STARTUP2 or SECONDARY or PRIMARY states with correct arbiter placement.
2. Login to web proxy on both primary and secondary site and add a web weight using **web weight add <ip>:443 1** for all those nodes that you want to provide a web weight of 1 on the respective proxies.
3. Run **cluster provision** to mitigate the failure. It is recommended that this step is run in a terminal opened with the `tmux` command.
4. Run **cluster run all app status** to check if all the services are up and running after cluster provisioning completes.

Note: If the existing nodes in the cluster do not see the new incoming cluster after **cluster add**, try the following steps:

1. Run **cluster del <ip>** from the primary database node, <ip> being the IP of the new incoming node.
2. Log into any other node (not the new node) and run **cluster add <ip>** ,<ip> being the IP of the new incoming node.

16.7.5. Scenario: Loss of the Primary Database Server in a Modular Cluster

- The administrator deployed the cluster into a Primary and DR site.
- The cluster is deployed following the Installation Guide.
- The example is a typical cluster deployment: 8 nodes, where 3 nodes are database servers, 3 nodes are application nodes and 2 nodes are proxy servers.

  The design is preferably split over 2 physical data centers.

Database Node Failure

- Normal operations continue where the cluster is processing requests and transactions are committed successfully up to the point where a loss of a primary database server is experienced. In this scenario `DB01[172.29.42.103]` failed while transactions were running.
- Examine the cluster status running **cluster status** to determine the failed state:

```
Data Centre: unknown

            database : unknown_172.29.42.103[172.29.42.103] (not responding)


Data Centre: jhb
              application : AS01[172.29.42.100]
                            AS02[172.29.42.101]

              webproxy :    PS01[172.29.42.102]

              database :    DB02[172.29.42.104]

Data Centre: cpt
              application : AS03[172.29.21.100]

              webproxy :   PS02[172.29.21.102]

              database :   DB03[172.29.21.101]
```

- Some downtime occurs. This can be take up to 15 minutes. To speed up recovery, restart the services: **cluster run all app start**.
- The loss of the primary database server will cause an election and the database node with the highest weighting still running will become primary.

- Check the weights set in the cluster configuration: **database weight list**

```
[ new example for modular needed ]
  platform@AS01:~$ database weight list
      172.29.21.101:
          weight: 20
      172.29.42.103:
          weight: 50
      172.29.42.104:
          weight: 40
```

- The primary database node `172.29.42.103` failed and therefore node `172.29.42.104` will become the primary database node after election.
- To find the primary database, run **database primary**.

```
platform@AS02:~$ database primary
172.29.42.104
```

- At this point *all* transactions that are currently in flight are lost and will not recover.
- The lost transactions have to be replayed or rerun.

  Bulk load transactions cannot be replayed and have to be rerun. Before resubmitting a failed Bulk load job, carry out the following command on the primary node CLI in order to manually clear each failure transaction that still has a Processing status *after a service restart*. Use the command:

  voss finalize_transaction <Trans ID>

  The failed transaction status then changes from Processing to Fail.
- With the database server `DB01[172.29.42.103]` still down, replaying the failed transactions is successful.

Recovery Steps

If the server that is lost, is unrecoverable:

Generally, **cluster provision** must be run every time a node is deleted or added, even if it is a replacement node. It is recommended that this step is run in a terminal opened with the `tmux` command.

1. Delete its database weight (**database weight del <ip>**), in other words **database weight del 172.29.42.103**
2. Run **cluster del 172.29.42.103**, because this server no longer exists. Power off the deleted node, or disable its Network Interface Card.
3. *Only* run **cluster provision primary 172.29.42.104** from the current primary database node if **database config** shows *no* primary. Else only run **cluster provision** from the current primary node. It is recommended that this step is run in a terminal opened with the `tmux` command.

   This server should already have the highest weight, and its database weight can be checked with **database weight list**

   If all the database weights are deleted and provisioning is run again with **cluster provision**, the CLI message is:

   'Please select which of the database should be used as the remaining primary by running "database config", selecting a node to sync from (any node that says primary or secondary and is in a good state,

i.e. not in a 'RECOVERING' or 'STARTUP' state) and rerun provisioning with "cluster provision primary <db server ip from commmand above>"'

4. A new database node needs to be deployed. Ensure the server name, IP information and data centre name is the same as on the server that was lost.
5. Run **cluster provision** on the cluster *without* the node to be added.

   Create the *new database node* at the *required data center* - see: *Platform install OVA on a VM*.
6. Run **cluster prepnode** on *all* servers.
7. Run **cluster add <ip>** from an existing node, with the IP address of the new database server to add it to the existing cluster.
8. Check the output of the commands: **cluster list** and **cluster status** from the existing node. If the new node does not show up:
   a. Run **cluster del <new node>**
   b. Rerun the add of the node on *another* node, until the node shows up in **cluster list** and **cluster status**.
   c. Verify that the node shows up from all existing nodes. The recovery process may be time consuming.
9. Delete all database weights in the cluster. On a selected database node, *for each database node IP*, run **database weight del <IP>**.
10. Re-add all database weights in the cluster. *On each database node*, for each database node IP, run **database weight add <IP> <weight>**, considering the following:
    - *For the new database node*, add a database weight lower than that of the weight of the current primary if this will be a secondary, or higher if this will be the new primary.

    When done, check the database weights - either individually for each node, or for the cluster by using the command:

    cluster run application database weight list

    Make sure all database nodes show correct weights.
11. Make sure the new node is part of the cluster (run **cluster list**) and run **cluster provision primary 172.29.42.104** *from the current primary* (where `172.29.42.104` is an example). It is recommended that this step is run in a terminal opened with the `tmux` command.

    During the provision process, the role of primary will then be transferred from the current primary to the node with the highest weight. The role transfer may take a significant amount of time, depending on the database size.

    During the process, typing **app status** from the new primary node will still show the database as `not provisioned`:

    ```
    mongodb v21.1.1 (2021-05-09 13:36)
       |-arbiter             running
       |-database            running (not provisioned)
    ```

    To check the progress of the transfer, the database log can be checked. Type **log follow mongodb/mongodb/mongodb.log**. When the transfer is complete, an entry will show `sync done` as in the example below:

```
2021-05-10T14:09:48.639986+00:00 un1 mongod.27020[129593]: [initial sync-0] initial␣
↪sync done; took 5821s.
```

While the primary role transfer is in progress, the system can be used, but bulk database operations should not be carried out, because the sync may fall too far behind to complete.

12. If an OVA file was not available for your current release and you used the most recent release OVA for which there is an upgrade path to your release to create the new unified node, *re-apply* the Delta Bundle upgrade to the cluster.

    Note that the new node version mismatch in the cluster can be ignored, since this upgrade step aligns the versions.

    See the upgrade document for your release.

Note: Upon cluster provision failure at any of the proxy nodes during provisioning, the following steps illustrate the cluster provisioning:

1. Run **database config** and check if nodes are either in STARTUP2 or SECONDARY or PRIMARY states with correct arbiter placement.
2. Login to web proxy on both primary and secondary site and add a web weight using **web weight add <ip>:443 1** for all those nodes that you want to provide a web weight of 1 on the respective proxies.
3. Run **cluster provision** to mitigate the failure.
4. Run **cluster run all app status** to check if all the services are up and running after cluster provisioning completes.

Note: If the existing nodes in the cluster do not see the new incoming cluster after **cluster add**, try the following steps:

1. Run **cluster del <ip>** from the primary node, <ip> being the IP of the new incoming node.
2. Delete all database weights. Run **database weight del <ip>** from the primary database node, <ip> being the IP of the nodes, including the new incoming node.
3. Log into a non primary database node and run **cluster add <ip>**, <ip> being the IP of the new incoming node.
4. Re-add all database weights. Run **database weight add <ip> <weight>** from the same session, <ip> being the IP of the nodes, including the new incoming node.
5. Use **cluster run database cluster list** to check if all nodes see the new incoming nodes inside the cluster.

16.7.6. Scenario: Loss of a non-primary database: Modular Cluster

- The administrator deployed the cluster into a Primary and DR site.
- The cluster is deployed following the Installation Guide.
- The example is a typical cluster deployment: 8 nodes, where 3 nodes are database servers, 3 nodes are application nodes and 2 nodes are proxy servers.

  The design is preferably split over 2 physical data centers.

Database Node Failure

- Normal operations continue where the cluster is processing requests and transactions are committed successfully up to the point where a loss of a non-primary node is experienced.

  In this example, `DB02[172.29.42.104]` failed while transactions were running.
- Examine the cluster status running **cluster status** to determine the failed state:

```
Data Centre: unknown

            database : unknown_172.29.42.104[172.29.42.104] (not responding)


Data Centre: jhb
              application : AS01[172.29.42.100]
                            AS02[172.29.42.101]

              webproxy :    PS01[172.29.42.102]

              database :    DB01[172.29.42.103]

Data Centre: cpt
              application : AS03[172.29.21.100]

              webproxy :   PS02[172.29.21.102]

              database :   DB03[172.29.21.101]
```

- At this point, *all* transactions that are currently in flight are lost and will not recover.
- The lost transactions have to be replayed or rerun.

  Bulk load transactions cannot be replayed and have to be rerun. Before resubmitting a failed Bulk load job, carry out the following command on an application node in order to manually clear each failure transaction that still has a Processing status *after a service restart*. Use the command:

  voss finalize_transaction <Trans ID>

  The failed transaction status then changes from Processing to Fail.
- With the database server `DB02[172.29.42.104]` still down, replaying the failed transactions are successful.

Recovery steps

If the server that is lost, is unrecoverable:

1. A new database node needs to be deployed. Ensure the server name, IP information and data center name is the same as on the server that was lost.
2. Delete the failed node database weight (**database weight del <ip>**), for example **database weight del 172.29.42.104**
3. Run **cluster del 172.29.42.104**, because this server no longer exists. Power off the deleted node, or disable its Network Interface Card.
4. Run **cluster provision** on the cluster *without* the node to be added.

   Then create the *new database node* at the *required data center* - see: *Platform install OVA on a VM* and switch on the newly installed node.
5. From the primary database node, run **cluster add <ip>**, with the IP address of the new database node to add it to the existing cluster.
6. Add database weights so that the weights distributed throughout the cluster
   - Delete all database weights in the cluster. On the primary database node, *for each database node IP*, run **database weight del <IP>**.
   - Re-add all database weights in the cluster. *On each database node*, for each database node IP, run **database weight add <IP> <weight>**
   - Check weights - either individually for each node, or for the cluster by using the command:

     cluster run application database weight list

     Make sure all database nodes show correct weights.
7. Run **cluster provision** to join the new database node to the cluster communications. It is recommended that this step is run in a terminal opened with the `tmux` command.
8. If an OVA file was not available for your current release and you used the most recent release OVA for which there is an upgrade path to your release to create the new database node, *re-apply* the Delta Bundle upgrade to the cluster.

   Note that the new node version mismatch in the cluster can be ignored, since this upgrade step aligns the versions.

   See the upgrade document for your release.

Note: Upon cluster provision failure at any of the proxy nodes during provisioning, the following steps illustrate the cluster provisioning:

1. Run **database config** and check if nodes are either in STARTUP2 or SECONDARY or PRIMARY states with correct arbiter placement.
2. Login to web proxy on both primary and secondary site and add a web weight using **web weight add <ip>:443 1** for all those nodes that you want to provide a web weight of 1 on the respective proxies.
3. Run **cluster provision** to mitigate the failure. It is recommended that this step is run in a terminal opened with the `tmux` command.
4. Run **cluster run all app status** to check if all the services are up and running after cluster provisioning completes.

Note: If the existing nodes in the cluster do not see the new incoming cluster after **cluster add**, try the following steps:

1. Run **cluster del <ip>** from the primary database node, <ip> being the IP of the new incoming node.
2. Run **database weight del <ip>** from the primary database node, <ip> being the IP of the new incoming node.
3. Log into any non primary database node and run **cluster add <ip>**, <ip> being the IP of the new incoming node.
4. Run **database weight add <ip> <weight>** from the same session, <ip> being the IP of the new incoming node.
5. Use **cluster run database cluster list** to check if all nodes see the new incoming nodes inside the cluster.

16.7.7. Scenario: Loss of a Primary Site in a Modular Cluster

- The administrator deployed the cluster into a primary and DR site.
- The cluster is deployed following the Installation Guide.
- The example is a typical cluster deployment: 8 nodes, where 3 nodes are database servers, 3 nodes are application nodes and 2 nodes are proxy servers.

  The design is preferably split over 2 physical data centers.
- The cluster might also be in two geographically dispersed areas. The cluster has to be installed in two different site names or data center names. In this scenario, a portion of the cluster is in Johannesburg and the other is in Cape Town, South Africa.

Primary site failure

- Normal operations continue where the cluster is processing requests and transactions are committed successfully up to the point where a loss of a primary site is experienced. In this scenario, the following nodes failed while transactions were running:
  - *AS01[172.29.42.100]*
  - *AS02[172.29.42.101]*
  - *PS01[172.29.42.102]*
  - *DB01[172.29.42.103]*
  - *DB02[172.29.42.104]*
- At this point, *all* transactions that are currently in flight are lost and will not recover.
- The lost transactions have to be replayed or rerun.

  Bulk load transactions cannot be replayed and have to be rerun. Before resubmitting a failed Bulk load job, carry out the following command on the primary node CLI in order to manually clear each failure transaction that still has a Processing status *after a service restart*. Use the command:

  voss finalize_transaction <Trans ID>

The failed transaction status then changes from Processing to Fail.

- Examine the cluster status by running **cluster status** to determine the failed state:

```
Data Centre: unknown
               application : unknown_172.29.42.100[172.29.42.100] (not responding)
                             unknown_172.29.42.101[172.29.42.101] (not responding)

               webproxy :    unknown_172.29.42.102[172.29.42.102] (not responding)

               database :    unknown_172.29.42.103[172.29.42.104] (not responding)
                             unknown_172.29.42.103[172.29.42.103] (not responding)


Data Centre: jhb
               application :

               webproxy :

               database :

Data Centre: cpt
               application : AS03[172.29.21.100]

               webproxy :   PS02[172.29.21.102]

               database :   DB03[172.29.21.101]
```

- The cluster will be not be operational and manual intervention is needed to recover if a continued flow of transactions is required with a minimum of downtime.
- If it was possible to recover the lost nodes within a reasonable time frame, the cluster will recover automatically if the nodes that were down were brought back into the cluster array successfully.
- To recover the lost nodes and if they are unrecoverable, carry out the following recovery steps.

Recovery Steps (two options):

Commands should be run on an operational unified node from the DR site. During the recovery of clusters, database weights should be deleted and added again.

1. Delete the failed node database weights from the cluster: **database weight del <ip>**
2. Run **cluster del <ip>** to remove the nodes at the failed primary site. Power off the deleted node, or disable its Network Interface Card.
3. At this point, you have two options:
    a. Option A: provision half the cluster for a faster uptime of your DR site. Only the DR site will then be operational after the provision. You can also optionally add nodes to this cluster.
    b. Option B: bring the full cluster back up at both the DR site and primary site. You need to redeploy the primary site nodes.
4. Option A: provision half the cluster or optionally adding 2 more nodes to it.
    a. If you choose to add 2 more nodes to optionally create a cluster with 2 application and 2 database nodes, deploy the new nodes as follows.

i. Run **cluster provision** on the cluster *without* the node to be added and then create the *new application and database nodes* at the *required data center* - see: *Platform install OVA on a VM*.

ii. Run **cluster prepnode** on all new nodes.

iii. From a running database node, run **cluster add <ip>**, with the IP address of each new node to add it to the existing cluster.

iv. Add the database weights nodes in the cluster at the DR site.

- Delete all database weights in the cluster of the DR site. On a selected database node, *for each database node IP*, run **database weight del <IP>**.
- Re-add all database weights in the cluster of the DR site. *On each database node*, for each database node IP, run **database weight add <IP> <weight>**, considering the following:

  For the new database node, add a database weight lower than that of the weight of the current primary if this will be a secondary, or higher if this will be the new primary.

b. Run **database config** to determine if you have a primary database. If not, run **cluster provision primary <ip>** (current primary IP) It is recommended that this step is run in a terminal opened with the `tmux` command. If you do have a primary database, only run **cluster providsion**.

c. If an OVA file was not available for your current release and you used the most recent release OVA for which there is an upgrade path to your release to create the new unified node, *re-apply* the Delta Bundle upgrade to the cluster.

Note that the new node version mismatch in the cluster can be ignored, since this upgrade step aligns the versions.

See the upgrade document for your release.

d. Check all services, nodes and weights - either individually for each node, or for the cluster by using the commands:

- **cluster run all app status** (make sure no services are stopped/broken - the message 'suspended waiting for mongo' is normal on the fresh database nodes)
- **cluster run application cluster list** (make sure all nodes show)
- **cluster run application database weight list** (make sure all database nodes show correct weights)

5. Option B: bring the full cluster back up at both the DR site and primary site. You need to redeploy the primary site nodes.

a. Deploy 5 nodes: 2 database nodes, 2 application nodes and 1 proxy node.

i. Run **cluster provision** on the cluster *without* the node to be added and then create the new application, proxy and database nodes at the *required data center* - see: *Platform install OVA on a VM*.

ii. Run **cluster prepnode** on all new nodes.

iii. Run **cluster add <ip>** from the current primary database node, with the IP address of each new node to add it to the existing cluster.

iv. Ensure the database weights are added back:

- Delete all database weights in the cluster. On a selected database node, *for each database node IP*, run **database weight del <IP>**.

- Re-add all database weights in the cluster. *On each database node*, for each database node IP, run **database weight add <IP> <weight>**, considering the following:

  For a new database node, add a database weight lower than that of the weight of the current primary if this will be a secondary, or higher if this will be the new primary.

v. Since the primary database node is newly added, run **cluster provision primary <ip>** (current primary IP), It is recommended that this step is run in a terminal opened with the `tmux` command.

After provisioning, the node with the largest database weight will be the primary server.

vi. If an OVA file was not available for your current release and you used the most recent release OVA for which there is an upgrade path to your release to create the new unified node, *re-apply* the Delta Bundle upgrade to the cluster.

Note that the new node version mismatch in the cluster can be ignored, since this upgrade step aligns the versions.

See the upgrade document for your release.

b. Check all services, nodes and weights - either individually for each node, or for the cluster by using the commands:

- **cluster run all app status** (make sure no services are stopped/broken - the message 'suspended waiting for mongo' is normal on the fresh database nodes)
- **cluster run application cluster list** (make sure all nodes show)
- **cluster run application database weight list** (make sure all database nodes show correct weights)

c. Run **cluster provision primary <ip>**, where `<ip>` is *the current primary database in the DR site*. It is recommended that this step is run in a terminal opened with the `tmux` command. The six node (or eight node) cluster then pulls the data from this `<ip>` into the new primary database server at the primary site.

After provisioning, the database configuration can then be checked with **database config** to verify the primary database node in the primary site.

d. On the new app nodes, check the number of queues using **voss queues** and if the number is *less than 2*, set the queues to 2 with **voss queues 2**.

Note: Applications are reconfigured and the `voss-queue` process is restarted.

16.7.8. Scenario: Loss of a DR Site in a Modular Cluster

- The administrator deployed the cluster into a Primary and DR site.
- The cluster is deployed following the Installation Guide.
- The example is a typical cluster deployment: 8 nodes, where 3 nodes are database servers, 3 nodes are application nodes and 2 nodes are proxy servers.

  The design is preferably split over 2 physical data centers.
- The cluster might also be in two geographically dispersed areas. The cluster has to be installed in two different site names or data center names. In this scenario, a portion of the cluster is in `jhb` and the other is in `cpt`.

DR site failure

- Normal operations continue where the cluster is processing requests and transactions are committed successfully up to the point where a loss of a DR site is experienced. In this scenario, the following nodes failed while transactions were running:
  - AS03[172.29.21.100]
  - PS02[172.29.21.101]
  - DB03[172.29.21.102]
- At this point, *all* transactions that are currently in flight are lost and will not recover. The lost transactions have to be rerun.
- The lost transactions have to be replayed or rerun.

  Bulk load transactions cannot be replayed and have to be rerun. Before resubmitting a failed Bulk load job, carry out the following command on the primary node CLI in order to manually clear each failure transaction that still has a Processing status *after a service restart*. Use the command:

  voss finalize_transaction <Trans ID>

  The failed transaction status then changes from Processing to Fail.
- With the DR site still down, replaying the failed transactions is successful
- Examine the cluster status by running **cluster status** to determine the failed state:

```
Data Centre: unknown
      application : unknown_172.29.21.100[172.29.21.100] (not responding)

      webproxy :    unknown_172.29.21.101[172.29.21.101] (not responding)

      database :    unknown_172.29.21.102[172.29.21.102] (not responding)


Data Centre: jhb
             application : AS01[172.29.42.100]
                           AS02[172.29.42.101]

             webproxy :    PS01[172.29.42.102]

             database :    DB01[172.29.42.103]
                           DB02[172.29.42.104]

Data Centre: cpt
             application :

             webproxy :

             database :
```

- The cluster will be operational, but only on the primary site.
- You need to recover the lost nodes and if they are unrecoverable. Follow the recovery steps below.

Recovery Steps

1. Remove the database weights of the failed database nodes from the cluster: **database weight del <ip>**
2. Run **cluster del <ip>** to remove the failed nodes from the existing half of the cluster. Power off the deleted node, or disable its Network Interface Card.
3. Run **cluster provision** from the primary database node before a new server is added. It is recommended that this step is run in a terminal opened with the `tmux` command.
4. Redeploy the failed DR site nodes if the nodes are unrecoverable. Deploy 3 nodes: 1 application, 1 database and 1 proxy node.
5. Run **cluster provision** from the primary database node on the cluster *without* the node to be added and then create the new node at the *required data center* - see: *Platform install OVA on a VM*.
6. Run **cluster prepnode** on all nodes.
7. From the primary database node, after the redeployment, run **cluster add <ip>** with the IP address of the new nodes to add them to the existing cluster. Run **cluster list** to make sure the nodes added in cluster.
8. Add the database weights nodes in the cluster.
   - Delete all database weights in the cluster. On a selected database node, *for each database node IP*, run **database weight del <IP>**.
   - Re-add all database weights in the cluster. *On each database node*, for each database node IP, run **database weight add <IP> <weight>**
9. Check all services, nodes and weights - either individually for each node, or for the cluster by using the commands:
   - **cluster run all app status** (make sure no services are stopped/broken - the message 'suspended waiting for mongo' is normal on the fresh database nodes)
   - **cluster run application cluster list** (make sure all nodes show)
   - **cluster run application database weight list** (make sure all database nodes show correct weights)
10. Run **cluster provision** on the primary database node to ensure that a primary is selected for the provisioning stage. It is recommended that this step is run in a terminal opened with the `tmux` command.

    After provisioning, the database configuration can then be checked with the command **database config**.
11. If an OVA file was not available for your current release and you used the most recent release OVA for which there is an upgrade path to your release to create the new unified node, *re-apply* the Delta Bundle upgrade to the cluster.

    Note that the new node version mismatch in the cluster can be ignored, since this upgrade step aligns the versions.

    See the upgrade document for your release.
12. If an Active/Passive configuration was enabled prior to failover, this should be reconfigured by logging in on the *application* nodes on the DR site and running the command **voss workers 0**.
13. On the new app nodes, check the number of queues using **voss queues** and if the number is *less than 2*, set the queues to 2 with **voss queues 2**.

Note: Applications are reconfigured and the `voss-queue` process is restarted.

16.7.9. Scenario: Loss of Full Cluster in a Modular Cluster

Background

- The administrator deployed the cluster into a primary and DR site.
- The cluster is deployed following the Installation Guide.
- The example is a typical cluster deployment: 8 nodes, where 3 nodes are database servers, 3 nodes are application nodes and 2 nodes are proxy servers.

  The design is preferably split over 2 physical data centers.
- The cluster might also be in two geographically dispersed areas. The cluster has to be installed in two different site names or data center names.

Full cluster failure

- In this scenario, *all* nodes failed while transactions were running.
- At this point, *all* transactions that were in flight are lost and will not recover.
- The lost transactions have to be rerun.
- The cluster will not be operational and manual intervention is needed to recover.
- To recover the cluster, carry out the Recovery Steps.

Recovery Steps

Important:

- Prerequisite: a system backup exported to a remote backup location. The backup file on the remote location would typically have a format *<timestamp>.tar.gz*. This recovery procedure will *only* succeed if you have a valid recent backup to restore.
- For details, considerations and specific commands at each step below, refer to the “Modular Cluster Multinode Installation” topic in the *Installation Guide*.

1. Ensure all traces of the previous nodes have been removed from the VMware environment.
2. Deploy fresh nodes as per the original topology.
   - Check topologies and hardware requirements in the *Architecture and Hardware Specification Guide*.
   - For new *node type* deployment at the *required data center*, see: *Platform install OVA on a VM*.
   - For the steps below, follow the “Modular Cluster Installation” topics in the *Installation Guide*:
3. Add each node to the cluster by running **cluster prepnode**.
4. From the primary database node, add each node to the cluster using the **cluster add <IP address of node>** command.

5. On the primary database node, set the database weights for each database node using the **database weight add <IP address of node> <weight>** command.
6. Restore a backup made from the highest weighted secondary database node in the original cluster.

   Follow the Import steps here: *Backup and Import to a New Environment*.

   Note: It is not necessary to run **cluster provision** again on the primary node. This action is included in the backup restore process.
7. On the new app nodes, check the number of queues using **voss queues** and if the number is *less than 2*, set the queues to 2 with **voss queues 2**.

   Note: Applications are reconfigured and the `voss-queue` process is restarted.

8. Ensure all services are up and running:

   Run **cluster run all app status** to check if all the services are up and running after the restore completes.

Note:

- Upon cluster provision failure at any of the proxy nodes during provisioning, the following steps illustrate the cluster provisioning:
  1. Run **database config** and check if nodes are either in STARTUP2 or SECONDARY or PRIMARY states with correct arbiter placement.
  2. Login to web proxy on both primary and secondary site and add a web weight using **web weight add <ip>:443 1** for all those nodes that you want to provide a web weight of 1 on the respective proxies.
  3. Run **cluster provision** to mitigate the failure.
  4. Run **cluster run all app status** to check if all the services are up and running after cluster provisioning completes.
- If the existing nodes in the cluster do not see the new incoming cluster after **cluster add**, try the following steps:
  1. Run **cluster del <ip>** from the primary node, <ip> being the IP of the new incoming node.
  2. Delete all database weights. Run **database weight del <ip>** from the primary node, <ip> being the IP of the nodes, including the new incoming node.
  3. Log into any secondary node (non primary unified node) and run **cluster add <ip>** ,<ip> being the IP of the new incoming node.
  4. Re-add all database weights. Run **database weight add <ip> <weight>** from the same session, <ip> being the IP of the nodes, including the new incoming node.
  5. Use **cluster run database cluster list** to check if all nodes see the new incoming nodes inside the cluster.

17. Troubleshooting

17.1. Platform User Password Recovery Procedure

The steps below describe how to reset the VOSS Automate platform user password if you forget the password and you are not able to access the CLI via platform user.

1. Log in to VMWare and choose the VOSS Automate Virtual Machine (VM).
2. Right-click the VM and choose **Edit Settings**.
3. Disconnect the network adapter by un-checking the **Connected** check box. This ensures that transactions are not lost.
4. Click the **VM Options** tab, **During the next boot, force entry into the BIOS setup screen** check-box is checked.
5. Click the **OK** button to apply the settings.
6. Open the VOSS Automate display (**Launch Virtual Machine Console**).
7. Under the **Power** menu option, click the reboot button (**Restart Guest**).
8. In the VM console, press F10 and **YES** to exit the BIOS (Do not make any changes in the BIOS). The next step needs to be performed quickly before the VOSS Automate system boots.
9. While the cursor highlights the first GRUB console entry (Ubuntu), press e. For username, enter: `platform` and password = boot password configured during setup.
10. Navigate to the second to the last line which starts with `linux` and ends with `fsck.repair=yes`.
11. Navigate to the end of the line after `=yes`, add a space and add `init=/bin/bash`.
12. Press Ctrl-x in order to boot the system.
13. When the system has booted, on the console at the `root@(none):/#` prompt, enter commands as follows:

`mount -o remount,rw /`	
`passwd platform`	Type in and confirm a new password. Check for the success message.
`mount -o rw /var/log`	Allow counters to be reset.
`/sbin/pam_tally2 --reset --user platform`	Reset the failed password attempt counter.
`sync`	Force a file system sync.
`exit`	Exit the console.

14. Reconnect the VM network adapter under the **Edit Settings** option.
15. Power off the VM and power on the VM again.
16. When the system boots, choose the default highlighted GRUB entry (not recovery mode).
17. Allow the disk checks to complete if they do run.
18. You can now log in as platform user with the password set above.

17.2. ‘No Space Left on Device’ Error

You receive the following error message while backing up or restoring VOSS Automate on a virtual machine: ‘No Space Left on Device.’ You can create a new virtual disk on the node with the primary database and then reassign the VOSS Automate data to the new disk. The new disk has enough space for you to perform the backup or restore operation.

Important: If you wish to revert to a *smaller* disk size or back to your original disk size after following the steps below, contact VOSS support.

1. In VMware, add a disk on the node that contains the primary database:
   a. From the VM menu, click **Edit Settings**.
   b. Click **Add**. The Add Hardware Wizard opens.
   c. Select Hard Disk and then click Next.
   d. Select Create a new virtual disk and then click Next.
   e. Set the capacity to be the same as the database disk: 250 GB.
   f. Accept the default file name and location, or click **Browse** to select a different location.
   g. Click **Finish**.

   Your guest operating system recognizes the new virtual disk as a new, blank hard disk.

   Note: If the steps above do not succeed on your version of VMWare, first turn off the virtual machine that contains the primary database, carry out the steps and then turn on the virtual machine.

2. Log in to the platform account on the virtual machine and run the **drives list** command.
3. In the command output, note the following information, which you will use in the next step:
   - The name of the new disk in the ‘Unused disks’ section
   - The identifier of the current disk, ‘services:backups,’ in the ‘Used disks and mountpoints’ section
4. Run the following command: **drives reassign <new disk name> services:backups**

All current data is moved to the new disk. You can continue with your backup or restore operation.

17.3. Loss of the whole cluster and redeploying new servers

Important: Backups should be created and restored in a `tmux` session - see: *Using the tmux command*.

The high level redeploy and backup restore steps are as follows:

- Redeploy the cluster.
- Store the backup you want to restore in a different location.
- Recreate the remote backups on the primary node using **backup create <loc-name> <URI>**.
- Copy the saved backup under the new UID folder on the remote backup server.
- Do a **backup list**

For example:

```
pxetest:
   URI: sftp://sftpusr:********@172.29.42.249/AS03
   Backups:
      1 backups have been created - most recently 2014-08-21 10:24
```

A **backup restore** can now be run on the primary.

The example console output below shows the steps and process:

Identifying the database primary:

```
platform@AS01:~$ database primary
172.29.42.100
```

Listing the backups:

```
platform@AS01:~$ backup list
 localbackup:
    URI: file:///backups
    Backups:
       2 backups have been created - most recently 2014-08-21 17:59
 pxetest:
    URI: sftp://sftpusr:********@172.29.42.249/AS01
    Backups:
       2 backups have been created - most recently 2014-08-21 12:54

You have new mail in /var/mail/platform
```

Restoring the backup:

```
platform@AS01:~$ backup restore pxetest 2014-08-21 12:54
Services will be restarted during the restore. Do you wish to continue? y
Application <name>-deviceapi processes stopped.
Stopping Application while performing database restore
```

(continues on next page)

(continued from previous page)

```
---------- AS02, ip=172.29.42.101, role=webproxy,application,database, loc=cpt


Stopping nginx:proxy

---------- AS01, ip=172.29.42.100, role=webproxy,application,database, loc=cpt


Application nginx processes stopped.


---------- AS02, ip=172.29.42.101, role=webproxy,application,database, loc=cpt


Application nginx processes stopped.


---------- AS04, ip=172.29.21.191, role=webproxy,application,database, loc=jhb


Application nginx processes stopped.


---------- AS03, ip=172.29.21.190, role=webproxy,application,database, loc=jhb


Application nginx processes stopped.
System restore starting from
  sftp://sftpusr:sftpusr@172.29.42.249/AS01/ba1e37deff1309edcc2595bf46c6bfc2a99ca164
   (1408625665)
Local and Remote metadata are synchronized, no sync needed.
Last full backup date: Thu Aug 21 12:54:25 2014
Successfully restored to
  /backups/appdata/restore_temp_1408699183, moving to /backups/appdata
Removing temporary files in /backups/appdata/restore_temp_1408699183
local
Dropping database <name>_FILES before restoring
MongoDB shell version: 2.6.1
connecting to: 127.0.0.1:27020/<name>_FILES
[object Object]
Repairing database <name>_FILES before restoring
MongoDB shell version: 2.6.1
connecting to: 127.0.0.1:27020/<name>_FILES
[object Object]
Dropping database PLATFORM before restoring
MongoDB shell version: 2.6.1
connecting to: 127.0.0.1:27020/PLATFORM
[object Object]
Repairing database PLATFORM before restoring
MongoDB shell version: 2.6.1
connecting to: 127.0.0.1:27020/PLATFORM
```

(continues on next page)

(continued from previous page)

```
[object Object]
Dropping database <name> before restoring
MongoDB shell version: 2.6.1
connecting to: 127.0.0.1:27020/<name>
[object Object]
Repairing database <name> before restoring
MongoDB shell version: 2.6.1
connecting to: 127.0.0.1:27020/<name>
[object Object]
Dropping database <name>_LOCKING before restoring
MongoDB shell version: 2.6.1
connecting to: 127.0.0.1:27020/<name>_LOCKING
[object Object]
Repairing database <name>_LOCKING before restoring
MongoDB shell version: 2.6.1
connecting to: 127.0.0.1:27020/<name>_LOCKING
[object Object]
Dropping database admin before restoring
MongoDB shell version: 2.6.1
connecting to: 127.0.0.1:27020/admin
[object Object]
Repairing database admin before restoring
MongoDB shell version: 2.6.1
connecting to: 127.0.0.1:27020/admin
[object Object]
Trying with oplogReplay
Trying without oplogReplay
restore successfull
Restarting services

Application  processes stopped.

Application  processes started.

System settings have changed, please reboot using 'system reboot'
```

17.4. Error Messages

The tables below provide:

- an error code range reference
- message details of the error codes

To inspect application log messages from the command line, set the debug level on and view the app log. Refer to the Platform Guide for more details.

```
voss set_debug 1
log view voss-deviceapi/app.log
```

The message strings are shown in their template format: references to specific properties are shown as placeholders that are represented by {} .

Note: For AuthError codes, the following rules apply:

- For API version 11.5.3 and below, only the **AuthError_11_5_3** table messages apply.
- For API greater than 11.5.3, **AuthError** table messages override the corresponding **AuthError_11_5_3** table messages, while the unchanged **AuthError_11_5_3** table messages still apply.

RuleError	Message	HTTP Code
15000	Invalid hierarchy for this operation. Please select new hierarchy.	449
15001	Multiple devices found at this Hierarchy level. Please select device.	449
15002	Multiple network device lists (NDL) found at this Hierarchy. Please select a NDL.	449
15003	Network device list reference (NDLR) not found at this Hierarchy.	449
15004	Network device list (NDL) with pkid [{}] not found in available list. Please check NDL rule at the Hierarchy	400
15005	No network device lists (NDL) found at this Hierarchy.	449
15999	Error, (UNHANDLED_ERROR)	400

TransactionError	Message	HTTP Code
23000	Unable to determine Transaction ID.	400
23001	Transaction must be registered with valid user details.	400
23002	Transaction not found.	404
23003	Transaction must be viewed with valid user details.	400
23004	{} (MAX_INSTANCES_EXCEEDED)	400
23005	Invalid Transaction State: {}	400
23006	Transaction canceled.	400
23007	Transaction must be registered with the hierarchy in which it is executing.	400
23008	Transaction must be registered with model_type if pkid is provided.	400
23010	The current filter caused a long running request. Please add more filter fields, use Case Sensitive or change the criteria types to one of {}.	400
23011	Invalid choices field [{}].	400
23012	The [{0}] condition on field [{1}], is not allowed.	400
23013	Invalid start and end date range provided in filter.	400
23014	Invalid start and end ID range provided in filter.	400
23015	Invalid ID value in filter	400
23999	Error, {} (UNHANDLED_ERROR)	400

ListUtilError	Message	HTTP Code
20000	Invalid query dictionary, expected 1 key!	400
20999	Error, (UNHANDLED_ERROR)	400

AllError	Message	HTTP Code
999999	All Error	400

ForeignKeyError	Message	HTTP Code
24000	Could not resolve foreign key to {model_type} with {attr_name}: {attr_value}.	400
24999	Error, {} (UNHANDLED_ERROR)	400

ChoicesError	Message	HTTP Code
26000	Instance context for choices not valid, instance: {instance}	400
26999	Error, {} (UNHANDLED_ERROR)	400

CnfError	Message	HTTP Code
40000	Device change notifications are not supported for device {}.	400
40001	Device change notification data for device {} has been lost. Tracking data has been repaired and collector process will continue. Some changes may have been lost, please run a full sync on the device.	400
40002	Device change notification tracking data for device {} has become corrupted. Tracking data has been repaired and collector process will continue. Some changes may have been lost, please run a full sync on the device.	400
40003	Device change notification tracking DB write for device {} failed. The collector process will continue to attempt DB writes. Please investigate the database write failure. {}	400
40004	Device change notification data DB write for device {} failed. The collector process will continue to attempt DB writes. Please investigate the database write failure. {}	400
40005	Unable to repair device change notification tracking data for device {}. {}	400
40006	Too many unprocessed changes recorded for device {}. No new changes will be recorded until at least {} changes are processed. Please configure and run the necessary data syncs.	400
40008	Could not update pending changes data for device {}. {}.	400
40010	Unable to clear device change notifications for device {}. {}.	400

PackageError	Message	HTTP Code
17000	Unable to load package. Package ({}) depends on ({}) but it does not exist.	400
17001	Unable to load package. Package ({}) requires ({} {}) but {} is currently loaded.	400
17999	Unable to load package. {}	400

CascadeDeleteError	Message	HTTP Code
13000	Hierarchy path or pkid required	400
13001	Could not delete {} out of {} resources.	400
13002	Could not move the following resources that failed to delete: {}.	400
13999	Error, (UNHANDLED_ERROR)	400

WebExError	Message	HTTP Code
31000	[{}] Site Name or Site ID must at least be specified	400

CertificateError	Message	HTTP Code
25001	Certificate request cannot be exported while 'Generate Certificate Signing Request' is not set.	400
25002	Certificate can only be imported when 'Generate Certificate Signing Request' is set.	400
25003	Certificate upload failed.	400
25004	Uploaded file is not a certificate in .pem format.	400
25005	The SSL certificate expired.	400
25006	Public key cannot be exported while 'Generate Certificate Signing Request' is set.	400
25999	Error, {} (UNHANDLED_ERROR)	400

FileUploadError	Message	HTTP Code
39000	Can not determine supported file extensions.	400
39001	'{}' does not have a valid file extension.	400
39002	File is too large. Maximum permitted file size is {} bytes.	400

BulkLoadError	Message	HTTP Code
10000	File Upload Error for File Name : ({})	400
10001	File Encoding Error : ({})	400
10002	Only valid Excel xlsx files are accepted	400
10003	General Error; ({})	400
10004	{success} out of {total} items loaded successfully.	400
10005	Resource data was not found in worksheet '{worksheet}'.	400
10006	Both parallel and serial are not allowed in '{worksheet}'.	400
10007	Differing parallel_transaction_limit values are not allowed in '{worksheet}'.	400
10008	Invalid value of '{limit}' for parallel_transaction_limit header in '{worksheet}', should be left blank or a number between 1 and 100(inclusive).	400
10010	Data does not conform to schema; ({})	400
10011	Hierarchy not specified for row with data; ({})	400
10012	'{user}' is not permitted access to resources at '{hierarchy}'.	403
10020	Hierarchy '{hierarchy}' was not found.	400
10021	Action '{action}' not allowed.	400
10022	Action '{action}' not allowed for model '{model}'.	400
10030	User '{username}' is not allowed to {operation} {model_type}.	403

continues on next page

Table 1 – continued from previous page

BulkLoadError	Message	HTTP Code
10040	Fields do not exist in {model}: {fields}.	400
10041	No search fields specified in row.	400
10042	More than one resource found. Search fields '{search}'.	400
10043	Resource not found. Search fields '{search}'.	400
10044	Malformed search fields: {fields}.	400
10045	Malformed fields{message}: {fields}.	400
10046	Can not find meta actions for specified resource instance.	400
10047	Malformed entity header '{header}' in cell '{cell}' worksheet '{sheet}'.	400
10050	Can not enforce data type '{data_type}' on '{data}'. Row data: {row_data}	400
10051	An internal error occurred while processing workbook '{filename}'{note}	400
10052	The specified meta_prefix '{meta_prefix}' in sheet '{sheet_name}' is invalid.	400
10053	The specified meta_prefix '{meta_prefix}' in sheet '{sheet_name}' was not found in base headers.	400
10054	The following base headers '{headers}' in '{sheet_name}' are prefixed, but meta_prefix is not specified.	400
10061	No match for device '{device}'.	400
10062	XLSX File Error: ({})	400

CnfWarning	Message	HTTP Code
45000	Unprocessed changes at 75%% of limit for device {}. Please configure and run the necessary data syncs.	400

DataSyncError	Message	HTTP Code
29000	Could not find user executing data sync operation.	500
29001	User [{}] does not have {} {} permissions.	403
29002	Could not establish a test connection to the device. Verify that your device connection details are correct.	400
29003	Aborting operation. Reason: {}	400
29004	{} (CRITICAL_SUBTRANSCATION_ERROR)	400
29005	Auth Error while testing connection to device	400
29999	Error, {} (UNHANDLED_ERROR)	500

WorkflowError	Message	HTTP Code
7000	Workflow not found	400
7001	Maximum workflow recursion depth exceeded	400
7002	Invalid workflow script identifier {}	400
7003	Specified workflow script name {} not found	400
7004	Error looking up workflow script names against API	400
7005	Invalid workflow action	400
7006	{} (FAILED)	400
7007	Advanced Find Options invalid - Resource not found with options {}	400
7008	{} (CONDITION_CONSTRAINT)	400
7009	Advanced Find Options invalid - More than one resource found with options {}	400
7010	Network Device List {} does not contain an entry for type {}	400
7011	Workflow operation Sync not supported for type {}	400
7012	No target device found for Workflow Sync operation	400
7999	Unexpected error occurred.	400

ExpectError	Message	HTTP Code
35000	The expect binary is not present in the path on the server	500
35001	There was an error executing the expect script : {}	500

ResourceError	Message	HTTP Code
4000	Error, Cannot delete Hierarchy until all resources under it are removed	400
4001	Error, Duplicate Resource Found. {}	400
4002	Resource Not Found {}	404
4003	Failed to save {}. {}	400
4004	Failed to save {}. {}	400
4005	Model Type cannot be None when adding a new Resource	400
4006	Resource Parent {} not found	400
4007	Resource Meta structure corrupt for {}	400
4008	Cannot create a Resource without a Parent Hierarchy	400
4009	Failed to save {}. {}	400
4010	Cannot find Resource relation {}	400
4011	Cannot find target device for model type {} in current hierarchy context	400
4012	Cannot find summary attr [{}] in schema root	400

continues on next page

Table 2 – continued from previous page

ResourceError	Message	HTTP Code
4013	Cannot perform operation, model {} already has one or more instances	400
4014	Cannot perform operation, resource is part of domain model {}	400
4015	Resource Meta structure corrupt. {}	400
4016	Badly-formed schema; 'properties' missing for data type 'object'	400
4017	Cannot perform operation, model {} is already referenced by one or more resources: {}	400
4018	Failed to execute {}. {}	400
4019	One or more errors occurred during import	400
4020	Transaction resource failed with errors {}	400
4021	Resources are not of the same type	400
4022	Model type for Resources not found	400
4023	Cannot move Hierarchy Node {} to {}	400
4024	Resource move failed with error {}	400
4025	Invalid business key {}, expected {}	400
4026	Cascade delete failed with error {}	400
4027	Invalid business key for import. Did not expect path, found {}.	400
4028	Resource move failed, Device at source hierarchy [{}] is different from the target hierarchy [{}]	400
4029	Resource [{}] cannot be accessed by user [{}]	403
4030	Cannot perform operation. Hierarchy Node Type [{}] is reserved.	400
4031	Search index is not up to date. Please notify your administrator before proceeding	400
4032	Attempting to create hierarchy node '{}' is not permitted.	403
4033	Could not update reference cache, from: {}, reference: {}, error: {}	403
4034	Resource move failed, hierarchy [{}] of type [{}] does not contain an NDLR	400
4035	CCM User Group [{}] not allowed.	400
4999	Unhandled Resource Error	400

MacroError	Message	HTTP Code
6000	Template must be a dictionary - got {}	400
6001	No hierarchy supplied	400
6002	Invalid macro specified: {}	400
6003	Macro lookup of {} failed at hierarchy {}	400
6004	Macro lookup of {} returned multiple values {} at hierarchy {}	400
6005	Macro lookup of {} failed when fetching from {} at hierarchy {}	400
6006	Macro lookup failed for field {} in context {}	400
6007	Macro lookup failed for field {} in context {}, type str or int expected not type dict {}	400
6008	Macro function {} not found	400
6009	Macro function arguments error - {}	400
6010	Macro function error - {}	400
6011	Unexpected business key format - {}	400
6012	Conditional Logic error occurred - {}	400
6013	Custom Macro function {} not found	400
6014	Custom Macro function {} not secure or contains invalid strings	400
6015	Could not parse the WhereClause Error:{} WhereClause:{} Please check quotation	400
6016	Lookup field {} not supported/permitted.	400
6017	Filter field: {} not in fields: {}.	400
6018	Incorrect hierarchy direction, {}. Allowed: {}.	400
6019	Error in macro function '{}' - {}	400
6999	Error, (UNHANDLED_ERROR)	400

InternalError	Message	HTTP Code
1000	Cannot import Python model name {}	404
1001	Python Type error	400
1002	{} must be an integer	400
1003	Improperly configured settings, {}	400

GraphLookupError	Message	HTTP Code
37000	Cannot perform operation, Resource with pkid [{}] cannot be accessed.	403

AuthError	Message	HTTP Code
27000	{} (INCORRECT_PASSWORD_ERROR)	401
27001	{} (PASSWORD_VERIFICATION_ERROR)	401
27009	Please enter a valid username and password.	401
27013	External (SSO or LDAP) authentication is required.	401
27014	Please enter valid answers to security questions.	401
27024	Login not allowed currently. Please contact your administrator.	403

ModelError	Message	HTTP Code
5000	[{}] Child model exists; ({})	400
5001	[{}] Model already exists; ({})	400
5002	One or more data sync errors occurred; ({})	400
5003	[{}] The helper cannot instantiate a model it does not recognize; ({})	400
5004	[{}] The specified resource could not be found; ({})	404
5005	[{}] A single model instance was expected but more than one was found; ({})	404
5006	[{}] Attempt to modify a read-only model failed; ({})	400
5007	[{}] Attempt to modify a read-only model field failed; ({})	400
5008	[{}] Data does not conform to schema; {}	400
5009	[{}] Validation failed; {}	400
5010	[{}] Error manipulating schema; ({})	400
5011	[{}] Error generating schema; ({})	400
5012	[{}] Invalid foreign key to {} for business keys {}	400
5013	[{}] Badly-formed schema; ({})	400
5014	[{}] Error deriving field value; {}	400
5015	Singleton constraint violated: Only one instance of [{}] is allowed per {}.	400
5016	The existing device in [{}] model cannot be modified, it is referenced by other resources.	400
5017	[{}] Invalid foreign key to {} for value {}	400
5018	[{}] Operation not supported for model instance; ({})	405
5019	[{}] Operation not supported; ({})	405
5020	Unable to determine workflow for operation '{}'	400
5021	Workflow '{}' not found	400
5022	Workflow operation '{}' clashes with an existing model attribute/method	400
5023	Unable to execute {} workflow. {}	400
5024	Unable to compile data for provisioning workflow for {}, error {}	400
5025	[{}] Connection timeout error after ({}) seconds	400

continues on next page

Table 3 – continued from previous page

ModelError	Message	HTTP Code
5026	[{}] Connection error; ({})	400
5027	[{}] API retry error; ({})	503
5028	[{}] Authentication error; ({})	400
5029	[{}] Attempt to add a contradicting rule; ({})	400
5030	[{}] Phones of this type must be added as gateway endpoints	400
5031	[{}] Unable to add NDLR to hierarchy node containing device models belonging to devices not referenced by NDLR	400
5032	[{}] Unable to query API with available data [{}]	400
5033	Retries exhausted; ({})	400
5050	Password cannot be reused.	400
5051	New password must have {} characters different from old password.	400
5052	User cannot change their password more than once within {} day(s). Please contact your administrator.	400
5053	Password does not meet minimum length required.	400
5054	Password {}.	400
5200	Invalid connection parameters for {}. Username and Password must specified for BASIC authentication method.	400
5201	Invalid connection parameters for {}. Token must specified for OAUTH authentication method.	400
5202	[{} {}] Unable to render model template [{}]. TEMPLATE: {} CONTEXT: {}	400
5203	[{} {}] Unable to parse API response. RESPONSE: {}	400
5204	Invalid connection parameters for {}. Hierarchy must be specified.	400
5205	[{}] Invalid paging parameters: page_size {} page_offset {}	400
5206	[{}] Paging required: page_size {} page_offset {}	400
5207	[{}] External response exceeded memory limit [{}] [{} {}]	400
5208	[{}] Template output exceeded memory limit [{}] [{}]	400
5209	[{}] Bad override for [{}]	400
5210	[{}] Session expired. The session cache has been cleared and the next request will go through successfully.	400
5211	[{}] Unable to authenticate using session based auth. {}	400
5212	[{}] Cannot add device {}	400
5215	[{}] Disallowed input [{}]	400
5270	[{}] Request start over required: {}	400
5280	Request start over attempts exhausted {}	400
5290	AXL request pagination error	400
5998	[{0}] {1}	400
5999	Error, {}. (UNHANDLED_ERROR)	400

ApiError	Message	HTTP Code
3000	Hierarchy context may not be None, please select Hierarchy	400
3001	Error, Incorrect request format	400
3002	Error, Unhandled method for URL	400
3003	Invalid import file specified. {}	400
3004	Invalid export URL specified. {}	400
3005	Error, Invalid list view sort key [{}]. Valid options are {}	400
3006	Error, Invalid list direction [{}]. Valid options are {}	400
3007	Error, No schema available during list view	400
3008	Provisioning Workflow error [{}]	400
3009	Nothing to export	400
3010	List delete failed, error [{}]	400
3011	List size not allowed, requested [{}], maximum [{}]	400
3012	List sort by hierarchy path not allowed	400
3013	Function not implemented	400
3014	Attribute field name required	400
3015	Hierarchy path [{}] not found.	400
3016	Model type list [{}] not found at or above the current hierarchy.	400
3017	Bulk update failed, error [{}].	400
3018	Bulk operation {} failed, error [{}].	400
3019	Schemas of data being imported have cyclic foreign keys {}.	400
3020	Imported {} out of {} items successfully.	400
3021	{} is a required GET parameter.	400
3022	Invalid Range HTTP header: {}	400
3023	{} is an invalid GET parameter.	400
3024	Resource pkid(s) must be specified	400
3025	Request was throttled.	429
3026	Invalid UTC date format given: {0}, requires: {1} or {2}	400
3027	The current filter caused a long running request. Please add more filter fields, use Case Sensitive or change the criteria types to one of {}.	400
3028	Model Instance Filter [{}] not found at or above the current hierarchy.	400
3029	Purge failed, error [{}]	400
3030	Model Type List of [{}] type not valid for [{}] sync.	400
3031	Model Instance Filter of [{}] type not valid for [{}] sync.	400
3032	{} GET parameter has an invalid value.	400
3999	Unhandled API Error	400

AuthError_11_5_3	Message	HTTP Code
27000	{} (INCORRECT_PASSWORD_ERROR)	403
27001	{} (PASSWORD_VERIFICATION_ERROR)	403
27002	{} (USER_NOT_FOUND_ERROR)	404
27003	{} (LOGIN_NOT_ALLOWED_ERROR)	403
27004	Account locked. Please contact your administrator.	403
27005	Too many failed login attempts for this user account. Try again later.	403
27006	Too many failed login attempts from this computer. Try again later.	403
27007	Your Web browser doesn't appear to have cookies enabled. Cookies are required for logging in.	400
27008	User is not allowed to log in.	403
27009	Please enter a valid username and password.	403
27010	This account is inactive.	403
27011	User account password must be changed before any API requests are authorized.	403
27012	{} (ACCOUNT_DISABLED)	403
27013	External (SSO or LDAP) authentication is required.	403
27014	Please enter valid answers to security questions.	403
27015	Password reset is not available for user.	403
27016	Security questions and answers not set up.	403
27017	User can not log in to this interface.	403
27018	User is disabled due to inactivity	403
27019	User is not allowed to login. Please contact your administrator.	403
27020	Login is currently disabled due to a temporary overload. Please try again later.	503
27021	User is not allowed to log in. Maximum user login sessions has been reached.	403

DatabaseError	Message	HTTP Code
2000	Cannot setup Mongo DB collection {}	400
2001	Find failed with spec={}, fields={}, skip={}, limit={}, sort_by={}, err={}	400
2002	Find one failed with spec={}, fields={}, err={}	400
2003	Get archive history failed with spec={}, fields={}, skip={}, limit={}, err={}	400
2004	Remove failed with spec={}, err={}	400
2005	Find and modify failed with spec={}, modify={}, err={}	400
2006	Save failed with spec={}, modify={}, err={}	400
2007	Count failed for {}	400
2008	Find failed with spec={}, fields={}, err={}	400
2009	Duplicate error with spec={}, modify={}, err={}	400
2010	Found more than one record with spec={}	400
2100	Error, Cannot connect to RESOURCE database collection	400
2101	Error, Cannot connect to DATA database collection	400
2102	Error, Cannot connect to ARCHIVE database collection	400
2103	Aggregate failed with group_by={}, match={}, aggregations={}, sort={}, err={}	400
2104	Bulk insert failed, err={}	400
2106	Bulk write failed, err={}	400
2107	Distinct failed with key={}, spec={}, err={}	400
2108	Explain not implemented for {}	400
2999	Unhandled Database Error	400

Authentication-ProxyError	Message	HTTP Code
32000	Cannot decode target user from authentication proxy. Error: {}	400
32001	Insufficient target user details specified by authentication proxy. Target user details must be contained in a JSON-formatted object with an email attribute.	400
32002	User [{}] is not a valid authentication proxy.	400
32003	Proxy user must be at a hierarchy above that of the target user.	400
32004	Error, {} (UNHANDLED_ERROR)	500

LibSchemaError	Message	HTTP Code
9000	Unhandled schema property error: [{}]	400
9001	Unhandled schema and data processing error: [{}]	400
9002	Data type incorrect, property: {}, not of type: {}	400
9999	Error, (UNHANDLED_ERROR)	400

RbacError	Message	HTTP Code
16000	Permission denied: {}.	400
16001	User not found.	400
16002	Role not specified; User [{}]	400
16003	Access profile not specified; User [{}], Role [{}]	400
16004	Role not found; User [{}], Role [{}]	400
16005	Access profile not found; User [{}], Role [{}], Access Profile [{}]	400
16006	User [{username}] is not allowed to {operation} attribute(s) of {model_type} resource [{pkid}]. Attribute(s) in breach: {breach_attrs}. This operation must be performed by the user's administrator.	403
16007	User [{username}] is not allowed to {operation} {model_type} resource [{pkid}]. This operation must be performed by the user's administrator.	403
16008	Invalid authorization token detected.	403
16009	Role not found; Hierarchy [{}], Role [{}]	400
16010	Access profile [{}] not found for Role [{}] in or above Hierarchy [{}]	400
16011	Access profile of role [{}] is not a subset of the request user's.	400
16012	SelfService Access Profile [{}] for Role [{}] at Hierarchy [{}] must not be created outside 'sys' hierarchy.	400
16999	Error, (UNHANDLED_ERROR)	400

SsoSettingsError	Message	HTTP Code
30000	Invalid certificate file found.	400
30001	Invalid key file found.	400
30002	Validity must not be negative or larger than {} hours ({} years).	400

ApiVersionError	Message	HTTP Code
38000	Invalid API header version specified: {}.	400
38001	No API version mapping defined.	400
38002	API header version: {} and API parameter version: {} mismatch	400

ExportError	Message	HTTP Code
36000	The export format is not specified in request.	400
36001	The specified export format is not supported.	415
36002	The worksheet was not initialized and can not be exported.	500
36100	License audit file transfer failed.	400
36101	tool/DataExtract failed for '{}'.	400
36102	A malformed record with pkid: '{}' and model_type: '{}' has been encountered.	400

DataImportError	Message	HTTP Code
11000	Multiple json files {} found in zip archive root; only 1 expected	400
11001	Import file validation failed with: {}	400
11999	Error, (UNHANDLED_ERROR)	400

InterfaceError	Message	HTTP Code
50000	Invalid interface value [{}] for header 'X_INTERFACE'	403
50001	No access profile associated with Interface [{}]	403

BulkLoad-MacroError	Message	HTTP Code
60000	Data type must be {}	400
60001	Invalid bulk load macro format {}. Supported format: {}	400

MigrationError	Message	HTTP Code
21000	Post condition failed. {}	400
21999	Error, {} (UNHANDLED_ERROR)	400

CryptoError	Message	HTTP Code
19000	Cryptography validation failed; {}.	400
19999	Error, (UNHANDLED_ERROR)	400

Saml2SsoError	Message	HTTP Code
14000	Could not find SSO settings; Hierarchy: {}.	400
14001	Found multiple SSO settings, only one expected; Hierarchy: {}.	400
14002	Could not find SSO Identity Provider; Hierarchy: {}, IDP uri: {}.	400
14003	Could not resolve SSO Identity Provider; Hierarchy: {}, IDP uri: {}.	400
14004	System generated certificate expected but not specified in data/SsoSettings.	400
14005	System generated certificate has an invalid private key.	400
14006	System generated certificate has an invalid certificate.	400
14007	Unknown principal: {}.	400
14008	Unsupported binding: {}.	400
14009	Verification error: {}.	400
14010	SubjectConfirmation is used but there is no NotOnOrAfter attribute	400
14012	NotBefore and NotOnOrAfter should be present when using either in Condition	400
14013	OneTimeUse element should be present when neither NotBefore nor NotOnOrAfter attributes in Condition	400
14014	Only one OneTimeUse element should be present in Condition	400
14015	Unencrypted assertions are not allowed	400
14016	The session cannot be used yet	400
14999	Error: {}. (UNHANDLED_ERROR)	400

ScriptError	Message	HTTP Code
8000	Script not found	400
8002	Syntax error on line {}	400
8003	Could not connect to {}	400
8004	Authentication failed {}	400
8999	Error, (UNHANDLED_ERROR)	400

Hierarchy-BasedAccessError	Message	HTTP Code
22000	Invalid traversal argument: '{}'; Traversal must be one of {}.	400
22001	{model_type} with {attr_name} {attr_value} is only permitted at the following hierarchy type(s): {hierarchy_types}.	403
22999	Error, {} (UNHANDLED_ERROR)	400

TestConnection-Error	Message	HTTP Code
12000	Please specify the model type of the device connection parameters	400
12999	Error, (UNHANDLED_ERROR)	400

SysError	Message	HTTP Code
0	Error, Mongo service not started	400
1	Error, Server too busy	400
2	Error, Celery service not started	400

PlatformError	Message	HTTP Code
28000	Could not execute platform command; Exit code: {}	500
28999	Error, {} (UNHANDLED_ERROR)	500

InternalApiUser-Error	Message	HTTP Code
18000	Authorization user [{}] not found.	400
18999	Error, (UNHANDLED_ERROR)	400

SystemMoni-toringError	Message	HTTP Code
70000	Aggregate {} is not supported by {}	400
72051	Connectivity failure	400
72052	Slow connection	400
72053	Utilization approaching limit	400
72054	Transactions queued for too long	400
72055	Transactions processing for too long	400

RisApiError	Message	HTTP Code
80000	RIS API data collection failed for {}	400

ThemeError	Message	HTTP Code
90000	Theme name {} is reserved for system use. Please choose another name. RIS API data collection failed for {}	400

ClientShapeError	Message	HTTP Code
100000	File operation error: '{}'	400

ClientShapeWarning	Message	HTTP Code
105000	System has not yet been linked to an account in ClientShape	400
105001	System is already linked to an account in ClientShape	400
105002	System has already been registered in ClientShape	400

NumberInventoryError	Message	HTTP Code
110000	Number inventory threshold reached: '{}'	400
111000	Failed to write the User Number Inventory CSV file to the configured NFS destination: '{}'	400

License Checking	Message	HTTP Code
120000	<LicenseCheckError>	
120001	<LicenseCheckExpiredError>	
120002	<LicenseCheckExpiryNotice>	
120003	<LicenseCheckServiceError>	
120004	<LicenseCheckAppNotReadyError>	
120000-120999		

License Audit	Message	HTTP Code
140000	The software version of the platform could not be determined.	400
140000 - 140099		

18. Appendices

18.1. MIBs

18.1.1. MIB List

Important: The VOSS Automate system uses standard MIBs that are usually deployed as part of a Network Management System (NMS). No VOSS Automate specific MIBs are added. The standard MIBs can for example be inspected from on-line resources, such as http://www.mibdepot.com.

The default net-SNMP packages that ship with VOSS Automate include:

- ACCOUNTING-CONTROL-MIB
- ADSL-LINE-EXT-MIB
- ADSL-LINE-MIB
- ADSL-TC-MIB
- ADSL2-LINE-MIB
- ADSL2-LINE-TC-MIB
- AGENTX-MIB
- AGGREGATE-MIB
- ALARM-MIB
- APM-MIB
- APPC-MIB
- APPLETALK-MIB
- APPLICATION-MIB
- APPN-DLUR-MIB
- APPN-MIB
- APPN-TRAP-MIB
- APS-MIB
- ARC-MIB
- ATM-ACCOUNTING-INFORMATION-MIB

- ATM-MIB
- ATM-TC-MIB
- ATM2-MIB
- BGP4-MIB
- BRIDGE-MIB
- CAPWAP-BASE-MIB
- CAPWAP-DOT11-MIB
- CHARACTER-MIB
- CIRCUIT-IF-MIB
- CLNS-MIB
- COPS-CLIENT-MIB
- DECNET-PHIV-MIB
- DIAL-CONTROL-MIB
- DIFFSERV-CONFIG-MIB
- DIFFSERV-DSCP-TC
- DIFFSERV-MIB
- DIRECTORY-SERVER-MIB
- DISMAN-EVENT-MIB
- DISMAN-EXPRESSION-MIB
- DISMAN-NSLOOKUP-MIB
- DISMAN-PING-MIB
- DISMAN-SCHEDULE-MIB
- DISMAN-SCRIPT-MIB
- DISMAN-TRACEROUTE-MIB
- DLSW-MIB
- DNS-RESOLVER-MIB
- DNS-SERVER-MIB
- DOCS-BPI-MIB
- DOCS-CABLE-DEVICE-MIB
- DOCS-IETF-BPI2-MIB
- DOCS-IETF-CABLE-DEVICE-NOTIFICATION-MIB
- DOCS-IETF-QOS-MIB
- DOCS-IETF-SUBMGT-MIB
- DOCS-IF-MIB
- DOT12-IF-MIB
- DOT12-RPTR-MIB

- DOT3-EPON-MIB
- DOT3-OAM-MIB
- DPI20-MIB
- DS0-MIB
- DS0BUNDLE-MIB
- DS1-MIB
- DS3-MIB
- DSA-MIB
- DSMON-MIB
- DVB-RCS-MIB
- EBN-MIB
- EFM-CU-MIB
- ENTITY-MIB
- ENTITY-SENSOR-MIB
- ENTITY-STATE-MIB
- ENTITY-STATE-TC-MIB
- ETHER-CHIPSET-MIB
- EtherLike-MIB
- FC-MGMT-MIB
- FCIP-MGMT-MIB
- FDDI-SMT73-MIB
- FIBRE-CHANNEL-FE-MIB
- FLOW-METER-MIB
- FORCES-MIB
- FR-ATM-PVC-SERVICE-IWF-MIB
- FR-MFR-MIB
- FRAME-RELAY-DTE-MIB
- FRNETSERV-MIB
- FRSLD-MIB
- Finisher-MIB
- GMPLS-LABEL-STD-MIB
- GMPLS-LSR-STD-MIB
- GMPLS-TC-STD-MIB
- GMPLS-TE-STD-MIB
- GSMP-MIB
- HC-ALARM-MIB

- HC-PerfHist-TC-MIB
- HC-RMON-MIB
- HCNUM-TC
- HDSL2-SHDSL-LINE-MIB
- HOST-RESOURCES-MIB
- HOST-RESOURCES-TYPES
- HPR-IP-MIB
- HPR-MIB
- IBM-6611-APPN-MIB
- IF-CAP-STACK-MIB
- IF-INVERTED-STACK-MIB
- IF-MIB
- IFCP-MGMT-MIB
- IGMP-STD-MIB
- INET-ADDRESS-MIB
- INTEGRATED-SERVICES-GUARANTEED-MIB
- INTEGRATED-SERVICES-MIB
- INTERFACETOPN-MIB
- IP-FORWARD-MIB
- IP-MIB
- IPATM-IPMC-MIB
- IPFIX-MIB
- IPMCAST-MIB
- IPMROUTE-STD-MIB
- IPOA-MIB
- IPS-AUTH-MIB
- IPSEC-SPD-MIB
- IPV6-FLOW-LABEL-MIB
- IPV6-ICMP-MIB
- IPV6-MIB
- IPV6-MLD-MIB
- IPV6-TC
- IPV6-TCP-MIB
- IPV6-UDP-MIB
- ISCSI-MIB
- ISDN-MIB

- ISIS-MIB
- ISNS-MIB
- ITU-ALARM-MIB
- ITU-ALARM-TC-MIB
- Job-Monitoring-MIB
- L2TP-MIB
- LANGTAG-TC-MIB
- LM-SENSORS-MIB
- LMP-MIB
- MALLOC-MIB
- MAU-MIB
- MGMD-STD-MIB
- MIDCOM-MIB
- MIOX25-MIB
- MIP-MIB
- MOBILEIPV6-MIB
- MPLS-FTN-STD-MIB
- MPLS-L3VPN-STD-MIB
- MPLS-LC-ATM-STD-MIB
- MPLS-LC-FR-STD-MIB
- MPLS-LDP-ATM-STD-MIB
- MPLS-LDP-FRAME-RELAY-STD-MIB
- MPLS-LDP-GENERIC-STD-MIB
- MPLS-LDP-STD-MIB
- MPLS-LSR-STD-MIB
- MPLS-TC-STD-MIB
- MPLS-TE-STD-MIB
- MSDP-MIB
- MTA-MIB
- Modem-MIB
- NAT-MIB
- NEMO-MIB
- NET-SNMP-AGENT-MIB
- NET-SNMP-EXAMPLES-MIB
- NET-SNMP-EXTEND-MIB
- NET-SNMP-MIB

- NET-SNMP-MONITOR-MIB
- NET-SNMP-PASS-MIB
- NET-SNMP-SYSTEM-MIB
- NET-SNMP-TC
- NET-SNMP-VACM-MIB
- NETWORK-SERVICES-MIB
- NHRP-MIB
- NOTIFICATION-LOG-MIB
- OPT-IF-MIB
- OSPF-MIB
- OSPF-TRAP-MIB
- OSPFV3-MIB
- P-BRIDGE-MIB
- PARALLEL-MIB
- PIM-BSR-MIB
- PIM-MIB
- PIM-STD-MIB
- PINT-MIB
- PKTC-IETF-EVENT-MIB
- PKTC-IETF-MTA-MIB
- PKTC-IETF-SIG-MIB
- POLICY-BASED-MANAGEMENT-MIB
- POWER-ETHERNET-MIB
- PPP-BRIDGE-NCP-MIB
- PPP-IP-NCP-MIB
- PPP-LCP-MIB
- PPP-SEC-MIB
- PTOPO-MIB
- PW-ATM-MIB
- PW-ENET-STD-MIB
- PW-MPLS-STD-MIB
- PW-STD-MIB
- PW-TC-STD-MIB
- PW-TDM-MIB
- PerfHist-TC-MIB
- Printer-MIB

- Q-BRIDGE-MIB
- RADIUS-ACC-CLIENT-MIB
- RADIUS-ACC-SERVER-MIB
- RADIUS-AUTH-CLIENT-MIB
- RADIUS-AUTH-SERVER-MIB
- RADIUS-DYNAUTH-CLIENT-MIB
- RADIUS-DYNAUTH-SERVER-MIB
- RAQMON-MIB
- RAQMON-RDS-MIB
- RDBMS-MIB
- RFC1155-SMI
- RFC1213-MIB
- RFC1381-MIB
- RFC1382-MIB
- RFC1414-MIB
- RIPv2-MIB
- RMON-MIB
- RMON2-MIB
- ROHC-MIB
- ROHC-RTP-MIB
- ROHC-UNCOMPRESSED-MIB
- RS-232-MIB
- RSERPOOL-MIB
- RSTP-MIB
- RSVP-MIB
- RTP-MIB
- SCSI-MIB
- SCTP-MIB
- SFLOW-MIB
- SIP-COMMON-MIB
- SIP-MIB
- SIP-SERVER-MIB
- SIP-TC-MIB
- SIP-UA-MIB
- SLAPM-MIB
- SMON-MIB

- SMUX-MIB
- SNA-NAU-MIB
- SNA-SDLC-MIB
- SNMP-COMMUNITY-MIB
- SNMP-FRAMEWORK-MIB
- SNMP-IEEE802-TM-MIB
- SNMP-MPD-MIB
- SNMP-NOTIFICATION-MIB
- SNMP-PROXY-MIB
- SNMP-REPEATER-MIB
- SNMP-SSH-TM-MIB
- SNMP-TARGET-MIB
- SNMP-TSM-MIB
- SNMP-USER-BASED-SM-MIB
- SNMP-USM-AES-MIB
- SNMP-USM-DH-OBJECTS-MIB
- SNMP-VIEW-BASED-ACM-MIB
- SNMPv2-CONF
- SNMPv2-M2M-MIB
- SNMPv2-MIB
- SNMPv2-PARTY-MIB
- SNMPv2-PDU
- SNMPv2-SMI
- SNMPv2-TC
- SNMPv2-TM
- SNMPv2-USEC-MIB
- SONET-MIB
- SOURCE-ROUTING-MIB
- SSPM-MIB
- SYSAPPL-MIB
- SYSLOG-MSG-MIB
- SYSLOG-TC-MIB
- T11-FC-FABRIC-ADDR-MGR-MIB
- T11-FC-FABRIC-CONFIG-SERVER-MIB
- T11-FC-FABRIC-LOCK-MIB
- T11-FC-FSPF-MIB

- T11-FC-NAME-SERVER-MIB
- T11-FC-ROUTE-MIB
- T11-FC-RSCN-MIB
- T11-FC-SP-AUTHENTICATION-MIB
- T11-FC-SP-POLICY-MIB
- T11-FC-SP-SA-MIB
- T11-FC-SP-TC-MIB
- T11-FC-SP-ZONING-MIB
- T11-FC-VIRTUAL-FABRIC-MIB
- T11-FC-ZONE-SERVER-MIB
- T11-TC-MIB
- TCP-ESTATS-MIB
- TCP-MIB
- TCPIPX-MIB
- TE-LINK-STD-MIB
- TE-MIB
- TIME-AGGREGATE-MIB
- TN3270E-MIB
- TN3270E-RT-MIB
- TOKEN-RING-RMON-MIB
- TOKENRING-MIB
- TOKENRING-STATION-SR-MIB
- TPM-MIB
- TRANSPORT-ADDRESS-MIB
- TRIP-MIB
- TRIP-TC-MIB
- TUNNEL-MIB
- UCD-DEMO-MIB.inc
- UCD-DEMO-MIB
- UCD-DISKIO-MIB.inc
- UCD-DISKIO-MIB
- UCD-DLMOD-MIB.inc
- UCD-DLMOD-MIB
- UCD-IPFILTER-MIB.inc
- UCD-IPFILTER-MIB
- UCD-IPFWACC-MIB.inc

- UCD-IPFWACC-MIB
- UCD-SNMP-MIB-OLD
- UCD-SNMP-MIB.inc
- UCD-SNMP-MIB
- UDP-MIB
- UDPLITE-MIB
- UPS-MIB
- URI-TC-MIB
- VDSL-LINE-EXT-MCM-MIB
- VDSL-LINE-EXT-SCM-MIB
- VDSL-LINE-MIB
- VDSL2-LINE-MIB
- VDSL2-LINE-TC-MIB
- VPN-TC-STD-MIB
- VRRP-MIB
- WWW-MIB
- IANA-ADDRESS-FAMILY-NUMBERS-MIB
- IANA-CHARSET-MIB
- IANA-FINISHER-MIB
- IANA-GMPLS-TC-MIB
- IANA-IPPM-METRICS-REGISTRY-MIB
- IANA-ITU-ALARM-TC-MIB
- IANA-LANGUAGE-MIB
- IANA-MALLOC-MIB
- IANA-MAU-MIB
- IANA-PRINTER-MIB
- IANA-PWE3-MIB
- IANA-RTPROTO-MIB
- IANATn3270eTC-MIB
- IANAifType-MIB
- IPFIX-SELECTOR-MIB

For further information on how to add a MIB, see:

http://www.net-snmp.org/wiki/index.php/TUT:Using_and_loading_MIBS

18.2. Data Export Types

18.2.1. Analogue line MGCP Data Export

Filename: `<YYYY-MM-DD_HHMM>_analogue_line_mgcp.json.gz`

Layout:

ELEMENT	DESCRIPTION AND SOURCE FIELD	DATA TYPE	VERSION
provider_name	Name of the Provider	string	v2
reseller_name	Name of the Reseller	string	v2
customer_name	name of the customer	string	v1
division_name	Intermediate Node (e.g Division or other node)	string	v1
location_name	Site Name	string	v1
hierarchy	The full hierarchy path for the item being exported	string	v1

- ELEMENT: usernames
  - DESCRIPTION: List of users assigned to the analog port
  - SOURCE FIELD: device/cucm/User.userid
  - DATA TYPE: Array of strings
  - VERSION: v1
- ELEMENT: gateway
  - DESCRIPTION: name of the gateway that the port is on
  - SOURCE FIELD: device/cucm/GatewayEndpointAnalogAccess.domainName
  - DATA TYPE: string
  - VERSION: v1
- ELEMENT: port_number
  - DESCRIPTION: gateway port for this configuration
  - SOURCE FIELD: device/cucm/GatewayEndpointAnalogAccess.endpoint.port.portNumber
  - DATA TYPE: string
  - VERSION: v1
- ELEMENT: port_type
  - DESCRIPTION: the type of port for this gateway (typically FXS for analog)
  - SOURCE FIELD: device/cucm/GatewayEndpointAnalogAccess.endpoint.product
  - DATA TYPE: string
  - VERSION: v1
- ELEMENT: description

  - DESCRIPTION: description of the gateway
  - SOURCE FIELD: device/cucm/GatewayEndpointAnalogAccess.endpoint.description
  - DATA TYPE: string
  - VERSION: v1
- ELEMENT: cucm_dn
  - DESCRIPTION: Internal Number assigned to the device profile (as configured in the PBX)
  - SOURCE FIELD: device/cucm/GatewayEndpointAnalogAccess.endpoint.port.lines.line.0.dirn.pattern
  - DATA TYPE: string
  - VERSION: v1
- ELEMENT: E164Members
  - DESCRIPTION: Array of E164 numbers and ranges assigned to `pilot_number` in the case of N-1 mapped lines
  - SOURCE FIELD: device/cucm/GatewayEndpointAnalogAccess.endpoint.port.lines.line.0.dirn.pattern
  - DATA TYPE: string
  - VERSION: v4
- ELEMENT: E164Members.e164_number
  - DESCRIPTION: E164 number in the case of N-1 mapped lines
  - DATA TYPE: string
  - VERSION: v4
- ELEMENT: E164Members.e164_range
  - DESCRIPTION: E164 range of `E164Members.e164_number` in the case of N-1 mapped lines
  - DATA TYPE: string
  - VERSION: v4
- ELEMENT: E164
  - DESCRIPTION: External Number (E164 number) assigned to the device profile
  - SOURCE FIELD: device/cucm/GatewayEndpointAnalogAccess.endpoint.port.lines.line.0.dirn.pattern
  - DATA TYPE: string
  - VERSION: v1

Example


```
[
{
  "division_name": "Intermed1",
  "usernames": [],
  "location_name": "Site1",
  "description": "",
  "port_number": 0,
  "hierarchy": "sys.171FDD8C03A6.Prov1.Resel1.Cust1.Intermed1.Intermed1_1.Site1",
  "gateway": "site_1_endpoint_1_gateway_name",
```


(continues on next page)

(continued from previous page)


```
  "E164": "s1e1_e164_value",
  "port_type": "Cisco MGCP FXS Port",
  "reseller_name": "Resel1",
  "provider_name": "Prov1",
  "cucm_dn": "11111",
  "customer_name": "Cust1"
},
{
  "division_name": "Intermed1",
  "usernames": [],
  "location_name": "Site1",
  "description": "",
  "port_number": 1,
  "hierarchy": "sys.171FDD8C03A6.Prov1.Resel1.Cust1.Intermed1.Intermed1_1.Site1",
  "gateway": "site_1_endpoint_2_gateway_name",
  "E164": "",
  "port_type": "Cisco MGCP FXS Port",
  "reseller_name": "Resel1",
  "provider_name": "Prov1",
  "cucm_dn": "",
  "customer_name": "Cust1"
},
{
  "division_name": "Intermed2",
  "usernames": [
    "fred",
    "bob"
  ],
  "location_name": "Site2",
  "description": "",
  "port_number": 1,
  "hierarchy": "sys.171FDD8C03A6.Prov2.Resel2.Cust2.Intermed2.Site2",
  "gateway": "site_2_endpoint_1_gateway_name",
  "E164": "",
  "port_type": "Cisco MGCP FXS Port",
  "reseller_name": "Resel2",
  "provider_name": "Prov2",
  "cucm_dn": "333333",
  "customer_name": "Cust2"
}
]
```

18.2.2. Analogue Line SCCP Data Export

Filename: `<YYYY-MM-DD_HHMM>_analogue_line_sccp.json.gz`

Layout:

ELEMENT	DESCRIPTION AND SOURCE FIELD	DATA TYPE	VERSION
provider_name	Name of the Provider	string	v2
reseller_name	Name of the Reseller	string	v2
customer_name	name of the customer	string	v1
division_name	Intermediate Node (e.g Division or other node)	string	v1
location_name	Site Name	string	v1
hierarchy	The full hierarchy path for the item being exported	string	v1
usernames	List of users assigned to the analog port device/cucm/User.userid	Array of strings	v1
gateway	name of the gateway that the port is on device/cucm/GatewaySccpEndpoints.domainName	string	v1
port_number	gateway port for this configuration device/cucm/GatewaySccpEndpoints.endpoint.index	string	v1
port_type	the type of port for this gateway (typically FXS for analog) device/cucm/GatewaySccpEndpoints.endpoint.product	string	v1
description	description of the gateway device/cucm/GatewaySccpEndpoints.endpoint.description	string	v1
E164Members	Array of E164 numbers and ranges assigned to `pilot_number` in the case of N-1 mapped lines	string	v4

- ELEMENT: cucm_dn
  - DESCRIPTION: Internal Number assigned to the device profile (as configured in the PBX)
  - SOURCE FIELD: device/cucm/GatewaySccpEndpoints.endpoint.lines.line.0.dirn.pattern
  - DATA TYPE: string
  - VERSION: v1
- ELEMENT: E164
  - DESCRIPTION: External Number (E164 number) assigned to the device profile
  - SOURCE FIELD: device/cucm/GatewaySccpEndpoints.endpoint.lines.line.0.dirn.pattern
  - DATA TYPE: string
  - VERSION: v1
- ELEMENT: E164Members.e164_number
  - DESCRIPTION: E164 number in the case of N-1 mapped lines
  - DATA TYPE: string
  - VERSION: v4
- ELEMENT: E164Members.e164_range

- DESCRIPTION: E164 range of `E164Members.e164_number` in the case of N-1 mapped lines
- DATA TYPE: string
- VERSION: v4

Example


```
[
{
  "division_name": "Intermed1",
  "usernames": [
    "test_userid"
  ],
  "location_name": "Site1",
  "description": "AN202AAAA202000",
  "port_number": 0,
  "hierarchy": "sys.48D13080D77F.Prov1.Resel1.Cust1.Intermed1.Intermed1_1.Site1",
  "gateway": "SKIGW202AAAA202",
  "E164": "test_e164",
  "port_type": "Analog Phone",
  "reseller_name": "Resel1",
  "provider_name": "Prov1",
  "cucm_dn": "\\+155545",
  "customer_name": "Cust1"
}
]
```


18.2.3. Call Pickup Group Data Export

(New report in version 2)

Filename: `<YYYY-MM-DD_HHMM>_call_pickup_group.json.gz`

Layout:

ELEMENT	DESCRIPTION AND SOURCE FIELD	DATA TYPE	VERSION
provider_name	Name of the Provider	string	v2
reseller_name	Name of the Reseller	string	v2
customer_name	Name of the Customer	string	v2
division_name	Intermediate Node (e.g Division or other node)	string	v2
location_name	Name of the Site	string	v2
hierarchy	The full hierarchy path for the item being exported	string	v2
pickup_group_name	The name of the Call Pickup Group device/cucm/CallPickupGroup.name	string	v2
pickup_group_number	The DN for the Call Pickup Group device/cucm/CallPickupGroup.pattern	string	v2
pickup_group_partition	The route partition for the Call Pickup Group DN device/cucm/CallPickupGroup.routePartitionName	string	v2
member	Array of member lines	array	v2
member.cucm_dn	Description of the directory number and partition device/cucm/Line.pattern	string	v2
member.parition	Route partition associated with the member directory number device/cucm/Line.routePartitionName	string	v2

Example


```
[

    {
        "provider_name": "CS-P",
        "reseller_name": "CS-NB",
        "customer_name": "CustomerName",
        "division_name": "",
        "location_name": "AAA-Boston",
        "hierarchy": "sys.hcs.CS-P.CS-NB.CustomerName.AAA-Boston",
        "pickup_group_name": "Support",
        "pickup_group_number": "80000",
        "pickup_group_partitiion": "Cu1-AllowVm-PT",
        "member": [
            {
                "cucm_dn": "50409",
                "partition": "Cu1-AllowVm-PT"
            }
        ]
    }
]
```

18.2.4. Contact Center Enterprise Data Export

Filename: `<YYYY-MM-DD_HHMM>_contact_center_enterprise.json.gz`

Layout:

ELEMENT	DESCRIPTION AND SOURCE FIELD	DATA TYPE	VERSION
provider_name	Name of the Provider	string	v2.2
reseller_name	Name of the Reseller	string	v2.2
customer_name	name of the customer	string	v2.2
division_name	Intermediate Node (e.g Division or other node)	string	v2.2
location_name	Site Name	string	v2.2
hierarchy	The full hierarchy path for the item being exported	string	v2.2
Name	Contact Center Username device/ccdm/Agent.Name	string	v2.2
PeripheralNumber	Skill group peripheral number device/ccdm/Agent.PeripheralNumber	integer	v2.2
Supervisor	User type device/ccdm/Agent.Supervisor	boolean	v2.2

Example


```
[
 {
   "division_name":"",
   "Supervisor":false,
   "Name":"standalone_ccdm_user_2",
   "hierarchy":"sys.hcs.Provider_01.Reseller_01.Customer_01.Site_01",
   "reseller_name":"Reseller_01",
   "location_name":"Site_01",
   "provider_name":"Provider_01",
   "PeripheralNumber":2,
   "customer_name":"Customer_01"
 }
]
```


18.2.5. Contact Center Express Data Export

Filename: `<YYYY-MM-DD_HHMM>_contact_center_express.json.gz`

Layout:

ELEMENT	DESCRIPTION AND SOURCE FIELD	DATA TYPE	VERSION
provider_name	Name of the Provider	string	v2.2
reseller_name	Name of the Reseller	string	v2.2
customer_name	name of the customer	string	v2.2
division_name	Intermediate Node (e.g Division or other node)	string	v2.2
location_name	Site Name	string	v2.2
hierarchy	The full hierarchy path for the item being exported	string	v2.2
username	Contact Center Express username	string	v2.2
userID	CUCM user ID device/uccx/Agent.userID	string	v2.2
firstName	Agent first name device/uccx/Agent.firstName	string	v2.2
lastName	Agent last name device/uccx/Agent.lastName	string	v2.2
extension	Agent extension device/uccx/Agent.extension	string	v2.2
teamName	Contact Center Express team name device/uccx/Agent.teamName	string	v2.2
type	Contact Center Express user type device/uccx/Agent.type	string	v2.2
autoAvailable	Availability status of the user device/uccx/Agent.autoAvailable	boolean	v2.2

Example


```
[
 {
   "division_name":"",
   "location_name":"Site_01",
   "firstName":"user_46",
   "extension":2,
   "hierarchy":"sys.hcs.Provider_01.Reseller_01.Customer_01.Site_01",
   "lastName":"Latame",
   "userID":"user_46",
   "teamName":"Default",
   "reseller_name":"Reseller_01",
   "provider_name":"Provider_01",
   "customer_name":"Customer_01",
   "type":"Agent",
   "autoAvailable":false
 }

]
```

18.2.6. Customer Data Export

(New report in version 2)

Filename: `<YYYY-MM-DD_HHMM>_customer.json.gz`

Layout:

ELEMENT	DESCRIPTION AND SOURCE FIELD	DATA TYPE	VERSION
provider_name	Name of the Provider	string	v2
reseller_name	Name of the Reseller	string	v2
customer_name	Name of the Customer	string	v2
hierarchy	The full hierarchy path for the item being exported	string	v2
account_id	The customer's account identifier	string	v2
external_id	An externally defined identifier for the customer	string	v2

Example

```
[
  {
      "provider_name": "CS-P",
      "reseller_name": "CS-NB",
      "customer_name": "Customer1",
      "hierarchy": "sys.hcs.CS-P.CS-NB.Customer1",
      "account_id": "ABCXYZ",
      "external_id": ""
  }
]
```

18.2.7. Extension Mobility Data Export

Filename: `<YYYY-MM-DD_HHMM>_extension_mobility.json.gz`

Layout:

ELEMENT	DESCRIPTION AND FIELD SOURCE	DATA TYPE	VERSION
provider_name	Name of the Provider	string	v2
reseller_name	Name of the Reseller	string	v2
customer_name	name of the customer	string	v1
division_name	Intermediate Node (e.g Division or other node)	string	v1
location_name	Site Name	string	v1
hierarchy	The full hierarchy path for the item being exported	string	v1
username	the username of the owner of device profile device/cucm/User.userid	string	v1
device_type	Model the extension mobility profile is setup as device/cucm/DeviceProfile.product	string	v1
device_profile_name	Name of the extension mobility profile device/cucm/DeviceProfile.name	string	v2
lines	Array of objects containing line information	array	v1
lines.cucm_dn	Internal Number assigned to the device profile (as configured in the PBX) device/cucm/DeviceProfile.lines.line.dirn.pattern	string	v1
lines.line_order	Line index. device/cucm/DeviceProfile.lines.line.index	integer	v2
lines.E164	External Number (E164 number) assigned to the device profile device/cucm/DeviceProfile.lines.line.dirn.pattern	string	v1
E164Members	Array of E164 numbers and ranges assigned to `cucm_dn` in the case of N-1 mapped lines	string	v4
E164Members.e164_number	E164 number in the case of N-1 mapped lines	string	v4
E164Members.e164_range	E164 range of `E164Members.e164_number` in the case of N-1 mapped lines	string	v4

Example:


```
[
{
  "division_name": "Intermed1",
  "username": "ba_user2",
  "location_name": "Site1",
  "hierarchy": "sys.822AF46F8FD3.Prov1.Resel1.Cust1.Intermed1.Intermed1_1.Site1",
  "lines": [
    {
      "line_order": 1,
      "cucm_dn": "50407",
      "E164": "91107"
    },
    {
      "line_order": 2,
      "cucm_dn": "50408",
```


(continues on next page)

(continued from previous page)


```
      "E164": "91108"
    }
  ],
  "device_type": "Cisco 9971",
  "device_profile_name": "ba_user2-UDP",
  "reseller_name": "Resel1",
  "provider_name": "Prov1",
  "customer_name": "Cust1"
},
{
  "division_name": "Intermed2",
  "username": "",
  "location_name": "Site2",
  "hierarchy": "sys.822AF46F8FD3.Prov2.Resel2.Cust2.Intermed2.Site2",
  "lines": [],
  "device_type": "Cisco 9971",
  "device_profile_name": "ba_user3-UDP",
  "reseller_name": "Resel2",
  "provider_name": "Prov2",
  "customer_name": "Cust2"
}
]
```


18.2.8. FMC Data Export

(New report in version 2)

This report includes users who have the FMC feature configured. The report includes the destination configured and an indication of whether the service is currently enabled or disabled (based on v2 FMC with CIM-based FMC). Any users without the FMC feature configured will not appear in the file. This report is only populated if the FMC adaptation is installed on the system - the file will be blank on systems without any users configured or if the adaptation is not installed.

Filename: `<YYYY-MM-DD_HHMM>_fmc.json.gz`

Layout:

ELEMENT	DESCRIPTION AND SOURCE FIELD	DATA TYPE	VERSION
provider_name	Name of the Provider	string	v2
reseller_name	Name of the Reseller	string	v2
customer_name	Name of the Customer	string	v2
division_name	Intermediate Node (e.g Division or other node)	string	v2
location_name	Name of the Site	string	v2
hierarchy	The full hierarchy path for the item being exported	string	v2
username	The userid of the remote destination profile data/GS_FMC_UserExtended_DAT.username	string	v2
destination	The mobile number associated with CIM device data/GS_FMC_UserExtended_DAT.fmc.mobile	string	v2
fmc_enabled	An indication of whether fixed mobile convergence is enabled for the destination data/GS_FMC_UserExtended_DAT.fmc.enabled	boolean	v2

Example

```
[
   {
       "provider_name": "CS-P",
       "reseller_name": "CS-NB",
       "customer_name": "AAAGlobal",
       "division_name": "",
       "location_name": "AAA-Boston",
       "hierarchy": "sys.hcs.CS-P.CS-NB.AAAGlobal.AAA-Boston"
       "username": "ba_user4",
       "destination": "08212345678",
       "fmc_enabled": true
   }
]
```

18.2.9. Hunt Group Data Export

Filename: `<YYYY-MM-DD_HHMM>_hunt_group.json.gz`

Layout:

ELEMENT	DESCRIPTION	DATA TYPE	VERSION
provider_name	Name of the Provider	string	v2
reseller_name	Name of the Reseller	string	v2
customer_name	name of the customer	string	v1
division_name	Intermediate Node (e.g Division or other node)	string	v1
hierarchy	The full hierarchy path for the item being exported	string	v1
location_name	Site Name	string	v1
hunt_group_name	Name assigned to the hunt group	string	v1
pilot_number	the internal number assigned as the pilot for the hunt group (as configured in the PBX) device/cucm/HuntPilot.pattern	string	v1
E164	the external number (Full E164 format) assigned as the pilot for the hunt group (as configured in the PBX) device/cucm/HuntPilot.pattern	string	v1
lines	Array of objects containing line information device/cucm/LineGroup	array	v1
lines.cucm_dn	Internal Number assigned to the device profile (as configured in the PBX) device/cucm/LineGroup.members.member.directoryNumber.	string	v1
lines.line_group_name	Name of the line group device/cucm/LineGroup.members.member.name	string	v2
E164Members	Array of E164 numbers and ranges assigned to `pilot_number` in the case of N-1 mapped lines	string	v4
E164Members.e164_number	E164 number in the case of N-1 mapped lines	string	v4
E164Members.e164_range	E164 range of `E164Members.e164_number` in the case of N-1 mapped lines	string	v4
partition	The route partition to which the Hunt Pilot number belongs device/cucm/HuntPilot.routePartitionName	string	v2

Example


```
[
{
  "division_name": "Intermed1",
  "location_name": "Site1",
  "hierarchy": "sys.57C1130EED66.Prov1.Resel1.Cust1.Intermed1.Intermed1_1.Site1",
  "lines": [
    {
      "cucm_dn": "HuntList1LineGroup1DirectoryNumber1Pattern",
      "line_group_name": "HuntList1LineGroup1"
    },
    {
      "cucm_dn": "HuntList1LineGroup1DirectoryNumber2Pattern",
      "line_group_name": "HuntList1LineGroup1"
    },
```


(continues on next page)

(continued from previous page)


```
    {
      "cucm_dn": "HuntList1LineGroup2DirectoryNumber1Pattern",
      "line_group_name": "HuntList1LineGroup2"
    },
    {
      "cucm_dn": "HuntList1LineGroup2DirectoryNumber2Pattern",
      "line_group_name": "HuntList1LineGroup2"
    }
  ],
  "partition": "RoutePartition1",
  "hunt_group_name": "HuntList1",
  "E164": "E164AssocDAT1",
  "reseller_name": "Resel1",
  "pilot_number": "PTCHuntPilot1",
  "provider_name": "Prov1",
  "customer_name": "Cust1"
},
{
  "division_name": "",
  "location_name": "",
  "hierarchy": "sys.57C1130EED66.Prov2.Resel2.Cust2",
  "lines": [],
  "partition": "",
  "hunt_group_name": "HuntList2",
  "E164": "E164AssocDAT2",
  "reseller_name": "Resel2",
  "pilot_number": "2222",
  "provider_name": "Prov2",
  "customer_name": "Cust2"
},
{
  "division_name": "Intermed1",
  "location_name": "Site1",
  "hierarchy": "sys.57C1130EED66.Prov1.Resel1.Cust1.Intermed1.Intermed1_1.Site1",
  "lines": [],
  "E164Members": [
    {
      "e164_number": "\\+495557000",
      "e164_range": "10"
    }
  ],
  "partition": "",
  "hunt_group_name": "HuntList3",
  "E164": "\\+495557000",
  "reseller_name": "Resel1",
  "pilot_number": "8217500",
  "provider_name": "Prov1",
  "customer_name": "Cust1"
}
]
```

18.2.10. Hybrid Data Export

Filename: `<YYYY-MM-DD_HHMM>_hybrid.json.gz`

Layout:

ELEMENT	DESCRIPTION AND SOURCE FIELD	DATA TYPE	VERSION
provider_name	Name of the Provider	string	v4
reseller_name	Name of the Reseller	string	v4
customer_name	name of the customer	string	v4
hierarchy	The full hierarchy path for the item being exported	string	v4
username	First name of user	string	v4
service_type	Hybrid type	string	v4
lines	array of lines: extension, e164, cos	array	v4
lines.extension	*release 19.3.4* data/MultiVendorServiceData_DAT.line1Dn (and line2Dn) *release >= 21.1* data/User.mvs_extensions.0.line	string	v4
lines.e164	*release 19.3.4* data/MultiVendorServiceData_DAT.line1E164 (and line2E164) *release >= 21.1* data/User.mvs_extensions.0.line_e164	string	v4
lines.cos	*release 19.3.4* data/MultiVendorServiceData_DAT.line1CoS (and line2CoS) *release >= 21.1* data/User.mvs_extensions.0.line_cos	string	v4

Example (19.3.4)

```
[
  {
    "username": "user_1",
    "hierarchy": "sys.9F73F4303A93.Provider1Hierarchy.Reseller1Hierarchy.
↪Customer1Hierarchy",
    "lines": [
      {
        "e164": "\\+441184025574",
        "class_of_service": "International-24Hrs-Enhanced",
        "extension": "8445574"
      },
      {
        "e164": "\\+441184025576",
        "class_of_service": "International-24Hrs-Enhanced",
        "extension": "8445576"
```

(continues on next page)

(continued from previous page)

```
      }
    ],
    "service_type": "Cisco-MS",
    "reseller_name": "Reseller1Hierarchy",
    "provider_name": "Provider1Hierarchy",
    "customer_name": "Customer1Hierarchy"
  }
]
```

18.2.11. Line Data Export

(New report in version 2)

Filename: `<YYYY-MM-DD_HHMM>_line.json.gz`

Layout:

ELEMENT	DESCRIPTION AND SOURCE FIELD	DATA TYPE	VERSION
provider_name	Name of the Provider	string	v2
reseller_name	Name of the Reseller	string	v2
customer_name	Name of the Customer	string	v2
division_name	Intermediate Node (e.g Division or other node)	string	v2
location_name	Name of the Site	string	v2
hierarchy	The full hierarchy path for the item being exported	string	v2
cucm_dn	Internal Number of this line device/cucm/Line.pattern	string	v2
partition	The route partition to which the number belongs device/cucm/Line.routePartitionName	string	v2
description	Description of the directory number and partition device/cucm/Line.description	string	v2
calling_search_space	This is mapped to the shareLineAppearanceCssName of the line device/cucm/Line.shareLineAppearanceCssName	string	v2

Example

```
[

    {
        "provider_name": "CS-P",
        "reseller_name": "CS-NB",
        "customer_name": "CustomerName",
        "division_name": "",
        "location_name": "AAA-Boston",
        "hierarchy": "sys.hcs.CS-P.CS-NB.CustomerName.AAA-Boston",
```

(continues on next page)

(continued from previous page)

```
        "cucm_dn": "50409",
        "partition": "Cu1-AllowVm-PT",
        "description": "Front Desk",
        "calling_search_space": "Cu1-ANumAnaly-CSS"
    }
]
```

18.2.12. Phones Data Export

Filename: `<YYYY-MM-DD_HHMM>_phones.json.gz`

Layout:

ELEMENT	DESCRIPTION AND SOURCE FIELD	DATA TYPE	VERSION
provider_name	Name of the Provider	string	v2
reseller_name	Name of the Reseller	string	v2
customer_name	name of the customer	string	v1
division_name	Intermediate Node (e.g Division or other node)	string	v1
location_name	Site Name	string	v1
hierarchy	The full hierarchy path for the item being exported	string	v1
usernames	list of usernames associated to the phones via Unified CM user, associated devices device/cucm/User.userid	array	v1
device_name	the name of the device (includes mac address if hardphone, softclients no mac) device/cucm/Phone.name	string	v1
description	Text field attached to the device device/cucm/Phone.description	string	v3
device_type	the model of the phone device/cucm/Phone.product	string	v1
device_css	Calling search space of the phone device/cucm/Phone.callingSearchSpaceName	string	v2
lines	Array of objects containing line information device/cucm/Phone.lines.line	array	v1
lines.line_order	Line index. device/cucm/Phone.lines.line.index	integer	v2
lines.cucm_dn	Internal Number assigned to the device profile (as configured in the PBX) device/cucm/Phone.lines.line.dirn.pattern	string	v1
lines.E164	External Number (E164 number) assigned to the device profile device/cucm/Phone.lines.line.dirn.pattern	string	v1
lines.recordingFlag	Either Automatic Call Recording Enabled or Automatic Call Recording Enabled or Selective Call Recording Enabled device/cucm/Phone.lines.line.recording_flag	string	v2
lines.recordingProfileName	recordingProfileName - string or None device/cucm/Phone.lines.line.profile	string	v2

ELEMENT	DESCRIPTION AND SOURCE FIELD	DATA TYPE	VERSION
E164Members	Array of E164 numbers and ranges assigned to `cucm_dn` in the case of N-1 mapped lines	string	v4
E164Members.e164_number	E164 number in the case of N-1 mapped lines	string	v4
E164Members.e164_range	E164 range of `E164Members.e164_number` in the case of N-1 mapped lines	string	v4
owner_username	User ID of the assigned phone user *Only in release >= 21.1* device/cucm/Phone.ownerUserName	string	v4
add_on_modules	Array of phone addon modules, incl. name, model, position *Only in release >= 21.1* cache.addOnModules.addOnModule	array	v4

- ELEMENT: add_on_modules.name
  - DESCRIPTION: phone addon module name
  - SOURCE FIELD: *Only in release >= 21.1*

    device/cucm/Phone.addOnModules.addOnModule.loadInformation
  - DATA TYPE: string
  - VERSION: v4
- ELEMENT: add_on_modules.model
  - DESCRIPTION: phone addon module model
  - SOURCE FIELD: *Only in release >= 21.1*

    device/cucm/Phone.addOnModules.addOnModule.model
  - DATA TYPE: string
  - VERSION: v4
- ELEMENT: add_on_modules.position
  - DESCRIPTION: phone addon module model
  - SOURCE FIELD: *Only in release >= 21.1*

    device/cucm/Phone.addOnModules.addOnModule.index
  - DATA TYPE: integer
  - VERSION: v4

Example:

```
[
{
  "division_name": "Intermed1",
  "usernames": [
    "s1p1_user1"
  ],
  "location_name": "Site1",
```

(continues on next page)

(continued from previous page)


```
  "description": "s1p1_desc",
  "hierarchy": "sys.AB707E3E6FC2.Prov1.Resel1.Cust1.Intermed1.Intermed1_1.Site1",
  "lines": [
    {
      "line_order": 0,
      "cucm_dn": "11111",
      "E164": "s1p1l1_e164",
      "recordingFlag": "Automatic Call Recording Enabled",
      "recordingProfileName": "Recording_Profile"
    }
  ],
  "add_on_modules": [
    {
      "position": 0,
      "model": "add_on_module_model-01",
      "name": "add_on_module_info-01"
    }
  ],
  "device_type": "s1p1_product_value",
  "reseller_name": "Resel1",
  "provider_name": "Prov1",
  "device_name": "s1p1_name",
  "device_css": "s1p1_css",
  "customer_name": "Cust1"
},
{
  "division_name": "Intermed1",
  "usernames": [],
  "location_name": "Site1",
  "description": "",
  "hierarchy": "sys.AB707E3E6FC2.Prov1.Resel1.Cust1.Intermed1.Intermed1_1.Site1",
  "lines": [],
  "add_on_modules": [],
  "device_type": "s1p2_product_value",
  "reseller_name": "Resel1",
  "provider_name": "Prov1",
  "device_name": "s1p2_name",
  "device_css": "s1p2_css",
  "customer_name": "Cust1"
},
{
  "division_name": "Intermed2",
  "usernames": [
    "s2p1_user1",
    "s2p1_user2"
  ],
  "location_name": "Site2",
  "description": "",
  "hierarchy": "sys.AB707E3E6FC2.Prov2.Resel2.Cust2.Intermed2.Site2",
  "lines": [
    {
```


(continues on next page)

(continued from previous page)


```
      "line_order": 1,
      "cucm_dn": "33333",
      "E164": "s2p1l1_e164",
      "recordingFlag": "Automatic Call Recording Enabled",
      "recordingProfileName": "Recording_Profile"
    },
    {
      "line_order": 2,
      "cucm_dn": "44444",
      "E164": "",
      "recordingFlag": "Call Recording Disabled",
      "recordingProfileName": ""
    }
  ],
  "add_on_modules": [],
  "device_type": "s2p1_product_value",
  "reseller_name": "Resel2",
  "provider_name": "Prov2",
  "device_name": "s2p1_name",
  "device_css": "s2p1_css",
  "customer_name": "Cust2"
}
]
```


18.2.13. Site Data Export

Filename: `<YYYY-MM-DD_HHMM>_site.json.gz`

Layout:

ELEMENT	DESCRIPTION AND SOURCE FIELD	DATA TYPE	VER.
provider_name	Name of the Provider	string	v2
reseller_name	Name of the Reseller	string	v2
customer_name	Name of the customer	string	v1
division_name	Intermediate Node (e.g Division or other node)	string	v1
location_name	Site Name	string	v1
hierarchy	The full hierarchy path for the item being exported	string	v1
customer_address1	Address string 1 for the customer	string	v1
customer_address2	Address string 2 for the customer	string	v1
customer_address3	Address string 3 for the customer	string	v1
location_address1	Address string 1 for the site data/BaseSiteDAT.Address1	string	v1
location_address2	Address string 2 for the site data/BaseSiteDAT.Address2	string	v1
location_address3	Address string 3 for the site data/BaseSiteDAT.Address3	string	v1
emergency_number	External emergency callback number assigned to the site data/DpSite.emerNumber	string	v1
ndl	The NDL name that the site uses data/Ndl.ndl.name	string	v1
inter_site_prefix	Digit dialled to prefix intersite calls (if the dial plan is setup that way) data/DpCustomer.isp	string	v1
external_access_prefix	Digit dialled to make external calls (if the dial plan is setup that way) data/DpSite.ext	string	v1
site_code	Dial Plan site code assigned to the site (if the dial plan is setup that way) data/DpSite.slc	string	v1
published_number	External published callback number assigned to the site data/DpSite.pubNumber	string	v1
country_code	Country code identifying the site data/Countries.international_dial_code	string	v1

ELEMENT	DESCRIPTION AND SOURCE FIELD	DATA TYPE	VER.
external_id	An externally defined ID for the site data/GS_LinkedSiteData_DAT.externalID	string	v2
extended_name	An expanded name for the site data/GS_LinkedSiteData_DAT.extendedName	string	v2

- ELEMENT: voice_bandwidth

  - DESCRIPTION: voice bandwidth allocation for the site
  - SOURCE FIELD: device/cucm/Location.betweenLocations.betweenLocation.audioBandwidth
  - DATA TYPE: string
  - VERSION: v1
- ELEMENT: video_bandwidth
  - DESCRIPTION: video bandwidth allocation for the site
  - SOURCE FIELD: device/cucm/Location.betweenLocations.betweenLocation.videoBandwidth
  - DATA TYPE: string
  - VERSION: v1

Example:

(* marked fields are new in version 2)


```
[
   {
    *  "provider_name": "CS-P",
    *  "reseller_name": "CS-NB",
       "customer_name": "Varidion",
      "division_name": "",
      "location_name": "Varidion-Reading",
      "hierarchy": "sys.hcs.CS-P.CS-NB.Varidion.Varidion-Reading",
      "customer_address1": "Varidion New York (Head Office)",
      "customer_address2": "L23, 33 Central Square",
      "customer_address3": "Dallas,TX, USA",
      "ndl": "GS-R3-VDN-CL1-NDL",
      "inter_site_prefix": "",
      "site_code": "",
      "video_bandwith": "",
      "emergency_number": "",
      "voice_bandwith": "",
      "country_code": "44",
      "external_access_prefix": "",
      "location_address1": "Varidion Reading",
      "location_address3": "Reading, Berkshire",
      "location_address2": "Atlantic House, Imperial Way",
      "published_number": "",
    * "external_id": "ABCXYZ",
    * "extended_name": "UK IT"
    }
]
```

18.2.14. Subscriber Data Export

Filename: `<YYYY-MM-DD_HHMM>_subscriber.json.gz`

Layout:

ELEMENT	DESCRIPTION AND SOURCE FIELD	DATA TYPE	VERSION
provider_name	Name of the Provider	string	v2
reseller_name	Name of the Reseller	string	v2
customer_name	name of the customer	string	v1
division_name	Intermediate Node (e.g Division or other node)	string	v1
location_name	Site Name	string	v1
hierarchy	The full hierarchy path for the item being exported	string	v1
username	username of the user device/cucm/User.userid	string	v1
first_name	First name of the user device/cucm/User.firstName	string	v1
middle_name	Middle name of the user device/cucm/User.middleName	string	v3
last_name	Last name of the user device/cucm/User.lastName	string	v1
email	email address of the user device/cucm/User.mailid	string	v1

ELEMENT	DESCRIPTION AND SOURCE FIELD	DATA TYPE	VERSION
entitlement_profile	the profile assigned to the user that defines the features they are enabled to have configured *release 19.3.4* data/HcsUserProvisioningStatusDAT.entitlement_profile *release >= 21.1* data/User.entitlement_profile or data/HcsUserProvisioningStatusDAT.entitlement_profile	string	v1
role	The role assigned to the user - defines privileges in the portal data/User.role	string	v1
credential_policy	The security profile assigned to the user - defined credential and other security rules for portal access data/User.account_information.credential_policy	string	v1
snr	Does the user have the SNR service configured device/cucm/RemoteDestinationProfile.userId	boolean	v1
voicemail	Does the user have a voicemail box configured device/cuc/User.Alias	boolean	v1
title	Subscriber's title *release 19.3.4* data/NormalizedUser.title *>= release 21.1* data/User.title or data/NormalizedUser.title	string	v2

ELEMENT	DESCRIPTION AND SOURCE FIELD	DATA TYPE	VERSION
department	Subscriber's department device/cucm/User.department	string	v2
telephone_number	Subscriber's telephone number as configured in the CUCM user record device/cucm/User.telephoneNumber	string	v2
pager_number	Subscriber's pager number device/cucm/User.pagerNumber	string	v3
imp_enabled	User enabled for Unified CM IM and Presence device/cucm/User.imAndPresenceEnable	boolean	v4

Example

(Fields marked * are new in version 2, fields marked ** are new in version 3, and fields marked *** are for v4)

```
[
   {
   *   "provider_name": "CS-P",
   *   "reseller_name": "CS-NB",
       "customer_name": "AAAGlobal",
       "division_name": "",
       "location_name": "AAA-Boston",
       "hierarchy": "sys.hcs.CS-P.CS-NB.AAAGlobal.AAA-Boston",
       "username": "ba_user4",
       "first_name": "Dean",
   **  "middle_name": "John",
       "last_name": "Daniels",
       "voicemail": false,
       "entitlement_profile": "AAAGlobal-Foundation-EP",
       "snr": false,
       "credential_policy": "HcsCredentialPolicy",
       "role": "AAA-BostonSelfService",
       "email": "email@theinternet.com",
   *   "title": "Dr.",
   *   "department": "R&D",
   *   "telephone_number": "0215252020",
   **  "pager_number": "5551234545",
  ***  "imp_enabled": False
    }
]
```


18.2.15. Webex Teams Data Export

Filename: `<YYYY-MM-DD_HHMM>_webex_teams.json.gz`

Layout:

ELEMENT	DESCRIPTION	DATA TYPE	VERSION
provider_name	Name of the Provider	string	v2.2
reseller_name	Name of the Reseller	string	v2.2
customer_name	name of the customer	string	v2.2
division_name	Intermediate Node (e.g. Division or other node)	string	v2.2
location_name	Site Name	string	v2.2
hierarchy	The full hierarchy path for the item being exported	string	v2.2
firstName	First name of user device/spark/User.firstName	string	v2.2
lastName	Last name of user device/spark/User.lastName	string	v2.2
email	User email address device/spark/User.email	string	v2.2
type	device/spark/User.type	string	
loginEnabled	device/spark/User.loginEnabled	boolean	
phoneNumbers	device/spark/User.phoneNumbers	array (objects)	
	• type • value Example: `[{"type": "work", "value": "\\+13125557007"}]`	• string • string	
extension	User extension device/spark/User.extension	string	v2.2
status	Webex App user status device/spark/User.status	string	v2.2
licenses	Webex licenses:	array (objects)	
	• license device/spark/User.licenses	• string	

Example

```
[
  {
   "provider_name": "Provider_01",
   "reseller_name": "Reseller_01",
   "customer_name": "Customer_01",
   "division_name": "Intermediate_Node_01",
   "location_name": "Site_02",
   "hierarchy": "sys.hcs.Provider_01.Reseller_01.Customer_01.Intermediate_Node_01.Site_02
↪",
   "firstName": "Randall",
```

(continues on next page)

(continued from previous page)


```
  "lastName": "Stephens",
  "email": "randall.stephens@shawshank.gov",
  "type": "person",
  "loginEnabled": true,
  "phoneNumbers": [
    {
      "type": "work",
      "value": "\\+13125557007"
    }
  ],
  "extension": "7007",
  "licenses": [
    {
    "license" : "Call on Webex"
    },
    {
    "license" : "Free message"
    },
    {
    "license" : "Free screen share"
    }
  ],
  "status": "The user has never logged in; a status cannot be determined."
 }
]
```


Note: Services reported on are dynamically included. The reference material and JSON snippet here are examples.

ELEMENT	DESCRIPTION	DATA TYPE	VERSION
hybrid_call_services:			v2.2
connect	Users' incoming calls will ring their work phones and the Cisco Webex App app. Users can call their colleagues from either their phones or the app, too. Aware must be enabled before the user can be enabled for Connect.	boolean	v2.2
aware	Users can share content from the Cisco Webex App app during a call from their work phones and view their call history in the app.	boolean	v2.2

ELEMENT	DESCRIPTION	DATA TYPE	VERSION
hybrid_calendar_services:			v2.2
google	Google Calendar	boolean	v2.2
microsoft_exchange	Microsoft Exchange/Office 365	boolean	v2.2

ELEMENT	DESCRIPTION	DATA TYPE	VERSION
meeting	Named User Licence. Each Named User license allows 1 user to be entitled as a meeting host. Named users can hold unlimited meetings.	N/A	N/A
webex_enterprise_200		boolean	v2.2
webex_support_center		boolean	v2.2
webex_meeting_center		boolean	v2.2
webex_cmr		boolean	v2.2
webex_event_center		boolean	v2.2
webex_training_center		boolean	v2.2
meeting		boolean	v2.2

Example


```
[
 {
   "division_name":"",
   "status":"",
   "location_name":"Site_03",
   "firstName":"",
   "hierarchy":"sys.hcs.Provider_01.Reseller_01.Customer_02.Site_03",
   "lastName":"",
   "provider_name":"Provider_01",
   "services":{
     "hybrid_call_services":{
       "connect":false,
       "aware":false
     },
     "message":{
       "messaging":false
     },
     "meeting":{
       "webex_enterprise_200":false,
       "webex_support_center":false,
       "webex_meeting_center":false,
       "webex_cmr":false,
       "webex_event_center":false,
```


(continues on next page)

(continued from previous page)


```
      "webex_training_center":false,
      "meeting":false
    },
    "hybrid_calendar_services":{
      "google":false,
      "microsoft_exchange":false
    },
    "hybrid_message_services":{
      "message":false
    }
  },
  "reseller_name":"Reseller_01",
  "line":"",
  "email":"spark_user_36@emailaccount.com",
  "customer_name":"Customer_02"
 },
]
```


18.2.16. Webex Devices Data Export

Filename: `<YYYY-MM-DD_HHMM>_webex_devices.json.gz`

Layout:

ELEMENT	DESCRIPTION	DATA TYPE	VERSION
provider_name	Name of the Provider	string	
reseller_name	Name of the Reseller	string	
customer_name	name of the customer	string	
division_name	Intermediate Node (e.g. Division or other node)	string	
location_name	Site Name	string	
hierarchy	The full hierarchy path for the item being exported	string	
id	Device ID E.g. : *ID_<device_name>*	string	
displayName	Device displayName E.g.: *DN_<device_name>*	string	
workspace	WorkspaceName E.g.: Boardroom	string	
username	Device username E.g.: *<device_name>_PersonID*	string	
product	product name, e.g. “Cisco Webex DX80”	string	

Example

```
{
  "provider_name": "Provider_01",
  "reseller_name": "Reseller_01",
  "customer_name": "Customer_01",
  "division_name": "Intermediate_Node_01",
  "location_name": "Site_02",
  "hierarchy": "sys.hcs.Provider_01.Reseller_01.Customer_01.Intermediate_Node_01.Site_02
↪",
  "id": "ID_spark_device_1",
  "displayName": "DN_spark_device_1",
  "workspace": "Boardroom",
  "username": "spark_device_1_PersonID",
  "product": "Cisco Webex DX80"
}
```


18.2.17. Webex Workspaces Data Export

Filename: `<YYYY-MM-DD_HHMM>_webex_workspaces.json.gz`

Layout:

ELEMENT	DESCRIPTION	DATA TYPE	VERSION
provider_name	Name of the Provider	string	
reseller_name	Name of the Reseller	string	
customer_name	name of the customer	string	
division_name	Intermediate Node (e.g. Division or other node)	string	
location_name	Site Name	string	
hierarchy	The full hierarchy path for the item being exported	string	
id	Device ID	string	
displayName	Workspace display name `device/spark/Location`	string	
type	Workspace type E.g. “huddle”,”focus”,”meetingRoom”,”open”,...	string	
callingType	Workspace calling type E.g. “freeCalling”	string	

Example


```
{
  "provider_name": "Provider_01",
  "reseller_name": "Reseller_01",
  "customer_name": "Customer_01",
  "division_name": "Intermediate_Node_01",
  "location_name": "Site_02",
```


(continues on next page)

(continued from previous page)

```
  "hierarchy": "sys.hcs.Provider_01.Reseller_01.Customer_01.Intermediate_Node_01.Site_02
↪",
  "id": "ID",
  "displayName": "DN_device",
  "type": "huddle",
  "callingType": "freeCalling"
}
```

18.2.18. MS Office 365 Data Export

Filename: `<YYYY-MM-DD_HHMM>_ms_o365.json.gz`

Layout:

ELEMENT	DESCRIPTION AND SOURCE FIELD	DATA TYPE	VERSION
username	VOSS Username of the user tied to this O365 service instance. For release 21.3 and later, the `data/User.username` field mapping is: `data/User.username_ms_365 = device/graph/MSOLUser.UserPrincipalName` If this field is blank, it means there is not a corresponding `data/User` for this MSOLUser.	string	v4
first_name	First name of the user device/msgraph/MsolUser.FirstName	string	v4
last_name	Last name of the user device/msgraph/MsolUser.LastName	string	v4
entitlement_profile	Entitlement profile of the user data.User.entitlement_profile of matching data/User	string	v4
o365_username	user name on O365 device/msgraph/MsolUser.UserPrincipalName	string	v4

ELEMENT	DESCRIPTION AND SOURCE FIELD	DATA TYPE	VERSION
display_name	displayed name of user device/msgraph/MsolUser.DisplayName	string	v4
is_licensed	User licensing status device/msgraph/MsolUser.IsLicensed	string	v4
licenses	List of licenses device/msgraph/MsolUser.Licenses	array	v4
licenses.account_sku_id	license SKUId licenses.AccountSkuId	string	v4
licenses.disabled_serviceplans	List of disabled service plans Licenses.X.ServicePlans	array	v4
licenses.disabled_serviceplans.$	Disabled service plan name Licenses.X.ServicePlans.X.ServiceName where Enabled is False	string	v4

Example


```
[
{
  "username": "user_1",
  "first_name": "FirstName_1",
  "last_name": "LastName_1",
  "display_name": "DisplayName_user_1",
  "hierarchy": "sys.A242BC6E4F94.Prov1.Resel1.Cust1",
  "is_licensed": "N",
  "entitlement_profile": "Default-EP",
  "o365_username": "user_1@emailaccount.com",
  "licenses": [
    {
      "account_sku_id": "DEVELOPERPACK_E5"
      "disabled_serviceplans": [
       {
        "Service_Name": "AAD_PREMIUM"
       }
      ]
    }
  ],
  "reseller_name": "Resel1",
  "provider_name": "Prov1",
  "customer_name": "Cust1"
},
{
  "username": "user_2",
  "first_name": "FirstName_2",
  "last_name": "LastName_2",
  "display_name": "DisplayName_user_2",
  "hierarchy": "sys.A242BC6E4F94.Prov1.Resel1.Cust1",
  "is_licensed": "N",
```


(continues on next page)

(continued from previous page)


```
  "entitlement_profile": "Default-EP",
  "o365_username": "user_2@emailaccount.com",
  "licenses": [
    {
      "account_sku_id": "DEVELOPERPACK_E5"
    }
  ],
  "reseller_name": "Resel1",
  "provider_name": "Prov1",
  "customer_name": "Cust1"
},
{
  "username": "user_3",
  "first_name": "FirstName_3",
  "last_name": "LastName_3",
  "display_name": "DisplayName_user_3",
  "hierarchy": "sys.A242BC6E4F94.Prov1.Resel1.Cust1",
  "is_licensed": "N",
  "entitlement_profile": "Default-EP",
  "o365_username": "user_3@emailaccount.com",
  "licenses": [
    {
      "account_sku_id": "DEVELOPERPACK_E5"
    }
  ],
  "reseller_name": "Resel1",
  "provider_name": "Prov1",
  "customer_name": "Cust1"
},
{
  "username": "user_4",
  "first_name": "FirstName_4",
  "last_name": "LastName_4",
  "display_name": "DisplayName_user_4",
  "hierarchy": "sys.A242BC6E4F94.Prov1.Resel1.Cust1",
  "is_licensed": "N",
  "entitlement_profile": "Default-EP",
  "o365_username": "user_4@emailaccount.com",
  "licenses": [
    {
      "account_sku_id": "DEVELOPERPACK_E5"
    }
  ],
  "reseller_name": "Resel1",
  "provider_name": "Prov1",
  "customer_name": "Cust1"
},
{
  "username": "user_5",
  "first_name": "FirstName_5",
  "last_name": "LastName_5",
```


(continues on next page)

(continued from previous page)


```
  "display_name": "DisplayName_user_5",
  "hierarchy": "sys.A242BC6E4F94.Prov1.Resel1.Cust1",
  "is_licensed": "N",
  "entitlement_profile": "Default-EP",
  "o365_username": "user_5@emailaccount.com",
  "licenses": [
    {
      "account_sku_id": "DEVELOPERPACK_E5"
    }
  ],
  "reseller_name": "Resel1",
  "provider_name": "Prov1",
  "customer_name": "Cust1"
}
]
```


18.2.19. MS Teams Data Export

Filename: `<YYYY-MM-DD_HHMM>_ms_teams.json.gz`

Layout:

ELEMENT	DESCRIPTION AND SOURCE FIELD	DATA TYPE	VERSION
ms_teams_username	UserPrincipalName device/msteamsonline/CsOnlineUser.UserPrincipalName	string	v4
alias	user alias device/msteamsonline/CsOnlineUser.Alias	string	v4
username	VOSS Username of the user tied to this MS Teams service instance. For release 21.3 and later, the `data/User.username` field mapping is: `data/User.username_ms_teams = device/msteamsonline/CsOnlineUser.UserPrincipalName` If this field is blank, it means there is not a corresponding `data/User` for this CsOnlineUser.	string	v4
entitlement_profile	Entitlement profile of the user data.User.entitlement_profile of matching data/User	string	v4
department	User department device/msteamsonline/CsOnlineUser.Department	string	v4
account_enabled	AccountEnabled device/msteamsonline/CsOnlineUser.AccountEnabled	boolean	v4
enterprise_voice	EnterpriseVoiceEnabled (not used) device/msteamsonline/CsOnlineUser.EnterpriseVoiceEnabled	string	v4
feature_types	FeatureTypes, e.g. Teams, Phone System, etc device/msteamsonline/CsOnlineUser.FeatureTypes	array	v4
line	OnPremLineURI device/msteamsonline/CsOnlineUser.LineURI	string	v4
line_type	LineURIType, e.g. OperatorConnect, DirectRouting, CallingPlan device/msteamsonline/CCsOnlineUser.LineURIType	string	v4
voice_routing_policy	OnlineVoiceRoutingPolicy device/msteamsonline/CsOnlineUser.OnlineVoiceRoutingPolicy	string	v4
tenant_dialplan	TenantDialPlan device/msteamsonline/CsOnlineUser.TenantDialPlan	string	v4
voicemail_policy	HostedVoicemailPolicy device/msteamsonline/CsOnlineUser.HostedVoicemailPolicy	string	v4
teams_upgrade_mod	TeamsUpgradeEffectiveMode device/msteamsonline/CsOnlineUser.TeamsUpgradeEffectiveMode	string	v4

Note: From release 21.4-PB1 onwards, the `first_name` and `last_name` fields have been removed as they are no longer used.

Example

```
[
{
  "username": "user_2",
  "entitlement_profile": "MsTeamsUser",
```

(continues on next page)

(continued from previous page)


```
  "hierarchy": "sys.02676185F05F.Prov1.Resel1.Cust1",
  "ms_teams_username": "user_2@emailaccount.com",
  "voicemail_policy": "BusinessVoice",
  "teams_upgrade_mode":"TeamsOnly",
  "voice_routing_policy": "Global",
  "alias": "user_2",
  "account_enabled": True,
  "tenant_dialplan": "",
  "department": "R&D",
  "reseller_name": "Resel1",
  "provider_name": "Prov1",
  "line": "18694400002",
  "line_type": "OperatorConnect",
  "feature_type": ["Teams"],
  "customer_name": "Cust1"
},
{
  "username": "user_3",
  "entitlement_profile": "MsTeamsUser",
  "hierarchy": "sys.02676185F05F.Prov1.Resel1.Cust1",
  "ms_teams_username": "user_3@emailaccount.com",
  "voicemail_policy": "BusinessVoice",
  "teams_upgrade_mode":"TeamsOnly",
  "voice_routing_policy": "Global",
  "alias": "user_3",
  "account_enabled": True,
  "tenant_dialplan": "",
  "department": "R&D",
  "reseller_name": "Resel1",
  "provider_name": "Prov1",
  "line": "18694400002",
  "line_type": "OperatorConnect",
  "feature_type": ["Teams"],
  "customer_name": "Cust1"
},
{
  "username": "user_1",
  "entitlement_profile": "MsTeamsUser",
  "hierarchy": "sys.02676185F05F.Prov1.Resel1.Cust1",
  "ms_teams_username": "user_1@emailaccount.com",
  "voicemail_policy": "BusinessVoice",
  "teams_upgrade_mode":"TeamsOnly",
  "voice_routing_policy": "Global",
  "alias": "user_1",
  "account_enabled": True,
  "tenant_dialplan": "",
  "department": "R&D",
  "reseller_name": "Resel1",
  "provider_name": "Prov1",
  "line": "18694400002",
  "line_type": "OperatorConnect",
```


(continues on next page)

(continued from previous page)

```
  "feature_type": ["Teams"],
  "customer_name": "Cust1"
}
]
```

18.2.20. MS Exchange Data Export

Filename: `<YYYY-MM-DD_HHMM>_ms_exchange.json.gz`

Layout:

ELEMENT	DESCRIPTION	DATA TYPE	VERSION
provider_name	Name of the Provider	string	v4
reseller_name	Name of the Reseller	string	v4
customer_name	name of the customer	string	v4
hierarchy	The full hierarchy path for the item being exported	string	v4
username	MS Exchange User Name	string	v4
exchange_mailbox_details	dictionary of permission types	dictionary	v4
exchange_mailbox_details.calendar_permissions	List of calendar permissions	array	v4
calendar_permissions.user	Calendar username	string	v4
calendar_permissions.access_rights	list of calendar permission rights	array	v4
exchange_mailbox_details.permissions	List of permissions	array	v4
permissions.user	Permissions username	string	v4
permissions.access_rights	list of permission rights	array	v4

Example

```
[
{
  "username": "ms_exchange_user_1@emailaccount.com",
  "hierarchy": "sys.003954111679.Prov1.Resel1.Cust1",
  "exchange_mailbox_details": {
    "calendar_permissions": [
      {
        "user": "ms_exchange_user_1",
        "access_rights": [
          "Editor",
          "Publishing Author",
          "Author"
        ]
      }
    ],
```

(continues on next page)

(continued from previous page)


```
    "permissions": [
      {
        "user": "ms_exchange_user_1",
        "access_rights": [
          "FullAccess",
          "SendAs"
        ]
      }
    ]
  },
  "reseller_name": "Resel1",
  "provider_name": "Prov1",
  "customer_name": "Cust1"
},
{
  "username": "ms_exchange_user_2@emailaccount.com",
  "hierarchy": "sys.003954111679.Prov1.Resel1.Cust1",
  "exchange_mailbox_details": {
    "calendar_permissions": [
      {
        "user": "ms_exchange_user_2",
        "access_rights": [
          "Editor",
          "Publishing Author",
          "Author"
        ]
      }
    ],
    "permissions": [
      {
        "user": "ms_exchange_user_2",
        "access_rights": [
          "FullAccess",
          "SendAs"
        ]
      }
    ]
  },
  "reseller_name": "Resel1",
  "provider_name": "Prov1",
  "customer_name": "Cust1"
}
]
```

18.2.21. Pexip Data Export

Filename: `<YYYY-MM-DD_HHMM>_pexip_conference.json.gz`

Layout:

ELEMENT	DESCRIPTION	DATA TYPE	VERSION
provider_name	Name of the Provider	string	v4
reseller_name	Name of the Reseller	string	v4
customer_name	name of the customer	string	v4
hierarchy	The full hierarchy path for the item being exported	string	v4
name	First name of user	string	v4
description	Last name of user	string	v4
owner_email	User email address	string	v4
type	service type	string	v4

Example

```
[
 {
   "name": "user_1",
   "hierarchy": "sys.0ECD98831FCF.Provider1Hierarchy.Reseller1Hierarchy.
↪Customer1Hierarchy",
   "description": "Description_PexIp_1",
   "owner_email": "user_1@dummy-emailaccount.com",
   "reseller_name": "Reseller1Hierarchy",
   "provider_name": "Provider1Hierarchy",
   "type": "conference",
   "customer_name": "Customer1Hierarchy"
 }
]
```

18.2.22. VOSS phone servers data export

Filename: <YYYY-MM-DD_HHMM>_voss_phone_servers.json.gz

Layout:

ELEMENT	DESCRIPTION AND SOURCE FIELD	DATA TYPE	VERSION
mac	MAC address of phone server data/PRS_MultiVendorPhone_DAT.mac	string	v4
phone_vendor	Vendor name data/PRS_MultiVendorPhone_DAT.phoneVendor	string	v4
phone_model	Model Name data/PRS_MultiVendorPhone_DAT.phoneModel	string	v4
lines	Lines names and CoS	array	v4
lines.name	Lines names data/PRS_MultiVendorPhone_DAT.line1Name data/PRS_MultiVendorPhone_DAT.line2Name	array	v4
lines.class_of_service	Lines CoS's data/PRS_MultiVendorPhone_DAT.line1Cos data/PRS_MultiVendorPhone_DAT.line2Cos	array	v4

Example


```
[
{
  "phone_model": "Cisco 6921",
  "hierarchy": "sys.F28DA5B756D7.Prov1.Resel1.Cust1",
  "lines": [
    {
      "class_of_service": "International-24Hrs-Enhanced",
      "name": "8445574"
    },
    {
      "class_of_service": "International-24Hrs-Enhanced",
      "name": "8445576"
    }
  ],
  "mac": "2C:54:91:88:C9:02",
  "phone_vendor": "Cisco",
  "reseller_name": "Resel1",
  "provider_name": "Prov1",
  "customer_name": "Cust1"
},
{
  "phone_model": "Cisco 6921",
  "hierarchy": "sys.F28DA5B756D7.Prov1.Resel1.Cust1",
  "lines": [
    {
```


(continues on next page)

(continued from previous page)


```
      "class_of_service": "International-24Hrs-Enhanced",
      "name": "8445574"
    },
    {
      "class_of_service": "International-24Hrs-Enhanced",
      "name": "8445576"
    }
  ],
  "mac": "2C:54:91:88:C9:01",
  "phone_vendor": "Cisco",
  "reseller_name": "Resel1",
  "provider_name": "Prov1",
  "customer_name": "Cust1"
}
]
```


18.3. Command Examples

18.3.1. `diag` Command Examples

For details, see: *Diagnostic Tools*.

`diag disk`

```
platform@VOSS:~$ diag disk
Filesystem               Size  Used Avail Use% Mounted on
udev                     3.9G     0  3.9G   0% /dev
tmpfs                    1.6G   80M  1.5G   5% /run
/dev/sda1                 18G  7.9G  8.9G  47% /
tmpfs                    3.9G  3.8M  3.9G   1% /dev/shm
tmpfs                    5.0M     0  5.0M   0% /run/lock
tmpfs                    3.9G     0  3.9G   0% /sys/fs/cgroup
none                     3.9G  8.0K  3.9G   1% /tmp
none                     3.9G     0  3.9G   0% /run/shm
/dev/sdb1                9.9G  846M  8.6G   9% /var/log
/dev/sdb2                 40G   15G   23G  39% /opt/platform
cgmfs                    100K     0  100K   0% /run/cgmanager/fs
/dev/sdc1                 50G  9.2G   38G  20% /backups
[...]
```

diag monitor

```
platform@VOSS:~$ diag monitor
2021-02-15 12:59:32.325225
  4810.5 Mb used, 2252.8 Mb free, 291 processes
 Private  +   Shared  =  RAM used    Files   Sockets CPU usage       Program

    -0.3 +      0.8 =      0.5              4       0       0       launch.true (20)
    -0.0 +      0.5 =      0.5              10      3       0       qmgr
    -0.0 +      0.6 =      0.6              10      3       0       pickup
     0.0 +      0.2 =      0.2              3       10      0       dbus-daemon
     0.0 +      0.1 =      0.1              7       7       0       systemd-udevd
     0.1 +      0.2 =      0.3              4       0       0       rbash
     0.1 +      0.2 =      0.3              3       21      0       ntpd
     0.7 +      0.7 =      1.4              4       3       0       (sd-pam)
[...]
```

diag monitor list

```
platform@VOSS:~$ diag monitor list
The following monitor findings were recorded:
---------------------------------------------
launch.true:
   max num_threads = 21 at 2019-10-23 07:10:03.2

systemd:
   avg num_sockets = 31
   max num_sockets = 33 at 2020-10-02 17:30:03.1
   avg num_files = 22
   max num_files = 27 at 2020-10-02 17:30:03.2

metric-collecto:
   max num_files = 27 at 2021-02-04 11:00:03.8

dockerd-ce:
   max mem_private = 94 at 2021-01-27 17:10:03.2
   avg num_sockets = 58
   max num_sockets = 65 at 2019-10-23 08:30:04.8
   avg num_files = 67
   max num_files = 78 at 2019-10-23 07:20:03.5
   max mem_total = 94 at 2021-01-27 17:10:03.2

scripts.py:
   max mem_private = 55 at 2019-10-23 06:20:03.4
   max mem_total = 58 at 2019-10-23 06:20:03.4

nginx:
[...]
```

diag nicstat

```
platform@VOSS:~$ diag nicstat
    Time      Int   rKB/s   wKB/s   rPk/s   wPk/s    rAvs    wAvs %Util    Sat
13:01:28 veth64e996f    1.23    6.86    4.29    2.98   292.7  2357.4  0.00   0.00
13:01:28    ens32    0.06    0.60    0.22    0.38   300.5  1621.6  0.00   0.00
13:01:28 veth7b06047    0.00    0.00    0.00    0.00    0.00   72.52  0.00   0.00
13:01:28 veth143e619    0.45    4.08    0.74    0.68   621.0  6172.7  0.00   0.00
13:01:28       lo    0.01    0.01    0.24    0.24   61.94   61.94  0.00   0.00
13:01:28 veth8b143f0   29.02    8.53   28.12   50.87  1056.9   171.8  0.02   0.00
13:01:28 veth6774425    0.04    0.00    0.00    0.00 15564.9   98.22  0.00   0.00
13:01:28 veth7f61555    0.00    0.00    0.00    0.00    0.00   72.52  0.00   0.00
13:01:28 veth1694cf5    0.00    0.00    0.00    0.00    0.00   72.11  0.00   0.00
13:01:28 veth655de40    0.03    0.05    0.28    0.15   123.8   324.7  0.00   0.00
13:01:28 veth76b16c9    0.04    0.09    0.05    0.06   855.2  1523.7  0.00   0.00
13:01:28  docker0   36.91   37.69   78.75   78.69   480.0   490.4  0.00   0.00
13:01:28 veth6f85a4b    0.35    3.94    1.63    1.17   218.6  3460.7  0.00   0.00
13:01:28 veth41ddd78    6.82   14.12   43.64   22.78   160.0   634.7  0.01   0.00
```

diag ntp

```
platform@VOSS:~$ diag ntp
       Server: 91.189.91.157 (za.pool.ntp.org)
Poll interval: 1min 4s (min: 32s; max 34min 8s)
         Leap: normal
      Version: 4
      Stratum: 2
    Reference: 8CCBCC4D
    Precision: 1us (-24)
Root distance: 64.781ms (max: 5s)
       Offset: -88.040ms
        Delay: 754.084ms
       Jitter: 78.200ms
  Packet count: 8
    Frequency: -187.812ppm
```

diag free

```
platform@VOSS:~$ diag free
              total        used        free      shared  buff/cache   available
Mem:        8174732     5024144      127472       59840     3023116     2714864
Swap:       2096124      270168     1825956
```

diag iostat

```
platform@VOSS:~$ diag iostat
Linux 4.4.0-151-generic (VOSS)       02/15/2021      _x86_64_        (4 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           1.86    0.00    1.08    0.09    0.00   96.97

Device:            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
sdb               2.31        37.31         6.40   47828330    8197356
sdc               0.90       114.40         0.01  146633661      14176
sdd               2.10         0.42        18.24     533207   23383024
sda               1.40        26.37         6.98   33802387    8949616
dm-0              2.53         0.41        18.24     530915   23383024
```

diag top

```
platform@VOSS:~$ diag top
top - 12:48:53 up 14 days, 20:03,  1 user,  load average: 0.53, 0.48, 0.30
Tasks: 297 total,   2 running, 295 sleeping,   0 stopped,   0 zombie
%Cpu(s):  1.9 us,  1.0 sy,  0.0 ni, 97.0 id,  0.1 wa,  0.0 hi,  0.1 si,  0.0 st
KiB Mem :  8174732 total,   149012 free,  5032732 used,  2992988 buff/cache
KiB Swap:  2096124 total,  1825764 free,   270360 used.  2706380 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
3165010 root      20   0   40656   3868   3180 R  11.1  0.0   0:00.04 top
   3783 snmp      20   0   59296   5968   2960 S   5.6  0.1  27:33.32 snmpd
   4915 root      20   0   15944   3184   1352 S   5.6  0.0 134:38.96 docker-p+
   8213 www-data  20   0  649084 332456   9660 S   5.6  4.1 445:08.94 python
      1 root      20   0   38036   4952   3108 S   0.0  0.1   1:07.13 systemd
      2 root      20   0       0      0      0 S   0.0  0.0   0:00.41 kthreadd
```

diag config platform

```
platform@VOSS:~$ diag config platform
    platform:
        backup:
            default: localbackup
            locations:
                localbackup:
                    uri: file:///backups
            passphrase: sftp://nobody:********@nowhere/
```

(continues on next page)

(continued from previous page)

```
        credstore: ********
        disa:
            inactive_lock_days: 35
            max_password_age: 60
            max_simultaneous_logins: 10
            min_password_age: 1
            system_default_umask: 077
            user_password_history: 10
            user_password_length: 8
        drivemounts:
            mongodb:
                dbroot:
                    disk: 3a4085ec-b88d-411d-a6c6-e2ea6bb2fefa
                    format: xfs
                    mountpoint: /opt/platform/apps/mongodb/dbroot
                    reserve: 10
            services:
                SWAPSPACE:
                    mountpoint: /SWAPSPACE
                backups:
                    disk: a2a3c270-5541-4cfe-87ba-7d61d0f6ccde
                    mountpoint: /backups
                    onassign: /opt/platform/apps/services/backup.py --name=localbackup --
↪register-location=file:///backups --force
                    permission: 1777
        location: cpt
[...]
```

diag config app snmp

```
platform@VOSS:~$ diag config app snmp
    load1: 4
    load15: 2
    load5: 1
    query: value not set
    syscontact: localhost
    syslocation: None
    sysname: None
```

diag proc

```
platform@VOSS:~$ diag proc
systemd-+-VGAuthService
        |-accounts-daemon-+-{gdbus}
        |                 `-{gmain}
        |-agetty
        |-atd
        |-auditd-+-audispd---{audispd}
```

(continues on next page)

(continued from previous page)

```
        |        `-{auditd}
        |-cgmanager
        |-containerd-+-containerd-shim-+-nginx---nginx---4*[nginx]
        |            |                 `-11*[{containerd-shim}]
        |            |-containerd-shim-+-mongodb---sudo---mongod-+-{ApplyBa.Journal}
        |            |                 |                         |-{Backgro.kSource}
[...]
        |-ntpd
        |-rsyslogd-+-syslog_alert.py
        |          |-{in:imklog}
        |          |-{in:impstats}
        |          |-{in:imuxsock}
        |          `-{rs:main Q:Reg}
        |-snmpd
        |-snmptrapd
        |-sshd---sshd---sshd---rbash---diag---sh---sudo---ui-real.py---execute---pstree
        |-stunnel4
        |-systemd---(sd-pam)
        |-systemd-journal
        |-systemd-logind
        |-systemd-udevd
        `-vmtoolsd---{gmain}
```

diag stats

```
platform@VOSS:~$ diag stats
CONTAINER ID   NAME                  CPU %  MEM USAGE / LIMIT    MEM %   NET I/O         ␣
↪ BLOCK I/O       PIDS
selenium       selenium              0.18%  58.8MiB / 7.796GiB   0.74%   32.1kB / 0B     ␣
↪ 78.9MB / 0B     30
selfservice_   selfservice_node      0.00%  101.9MiB / 7.796GiB  1.28%   32.1kB / 0B     ␣
↪ 32.6MB / 0B     31
voss-portal    voss-portal           0.00%  14.81MiB / 7.796GiB  0.19%   349kB / 58.9MB  ␣
↪ 31MB / 0B       5
voss-wsgi      voss-wsgi             0.90%  570.2MiB / 7.796GiB  7.14%   5.17GB / 456MB  ␣
↪ 186MB / 0B      10
voss-monitor   voss-monitoring       1.29%  189MiB / 7.796GiB    2.37%   62.5MB / 43.8MB␣
↪ 34.8MB / 0B     23
voss-risapi_   voss-risapi_collector 0.10%  106MiB / 7.796GiB    1.33%   5.35GB / 589MB  ␣
↪ 4.71MB / 0B     3
voss-queue     voss-queue            3.57%  315.3MiB / 7.796GiB  3.95%   18.5GB / 8.96GB␣
↪ 5.89MB / 0B     4
voss-cnf_col   voss-cnf_collector    0.15%  105.1MiB / 7.796GiB  1.32%   9.02GB / 1.61GB␣
↪ 31MB / 0B       3
mongoarbiter   mongoarbiter          1.45%  31.75MiB / 7.796GiB  0.40%   32.3kB / 0B     ␣
↪ 151MB / 520kB   27
mongodb        mongodb               3.22%  2.98GiB / 7.796GiB   38.23%  11.2GB / 38.1GB␣
↪ 1.2GB / 37.4GB  109
nginx          nginx                 0.03%  21.03MiB / 7.796GiB  0.26%   118MB / 56.1MB  ␣
↪ 28.4MB / 0B     6
```

diag tasks

```
platform@VOSS:~$ diag tasks
top - 13:04:13 up 14 days, 20:18,  1 user,  load average: 0.07, 0.30, 0.27
Tasks: 294 total,   1 running, 293 sleeping,   0 stopped,   0 zombie
%Cpu(s):  1.9 us,  1.0 sy,  0.0 ni, 97.0 id,  0.1 wa,  0.0 hi,  0.1 si,  0.0 st
KiB Mem :  8174732 total,   390904 free,  5046048 used,  2737780 buff/cache
KiB Swap:  2096124 total,  1828992 free,   267132 used.  2655756 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
3168836 root      20   0   40680   3916   3200 R  22.2  0.0   0:00.07 /usr/bin/top -b -n␣
↪1 -c -w 230
   6680 www-data  20   0  578060   8168   5536 S   5.6  0.1 136:09.89 /opt/voss-
↪deviceapi/go/bin/sysmon
   8213 www-data  20   0  649084 332120   9332 S   5.6  4.1 445:41.70 /usr/bin/python /
↪opt/voss-deviceapi/bin/python /opt/voss-deviceapi/src/deviceapi/background/queue.py 30
      1 root      20   0   38036   4896   3108 S   0.0  0.1   1:07.29 /sbin/init
      2 root      20   0       0      0      0 S   0.0  0.0   0:00.41 [kthreadd]
      3 root      20   0       0      0      0 S   0.0  0.0   0:35.52 [ksoftirqd/0]
      7 root      20   0       0      0      0 S   0.0  0.0   6:54.66 [rcu_sched]
      8 root      20   0       0      0      0 S   0.0  0.0   0:00.00 [rcu_bh]
      9 root      rt   0       0      0      0 S   0.0  0.0   0:08.10 [migration/0]
     10 root      rt   0       0      0      0 S   0.0  0.0   0:04.14 [watchdog/0]
     11 root      rt   0       0      0      0 S   0.0  0.0   0:04.03 [watchdog/1]
     12 root      rt   0       0      0      0 S   0.0  0.0   0:07.98 [migration/1]
     13 root      20   0       0      0      0 S   0.0  0.0   0:34.08 [ksoftirqd/1]
     15 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 [kworker/1:0H]
[...]
```

diag test_connection <host> <port>

```
platform@VOSS:~$ diag test_connection www.voss.com 80
Successfully connected to www.voss.com:80

platform@VOSS:~$ diag test_connection www.voss.com 81
Failed to connect to www.voss.com:81
```

diag top

```
platform@VOSS:~$ diag top
top - 13:04:43 up 14 days, 20:19,  1 user,  load average: 0.04, 0.27, 0.26
Tasks: 295 total,   1 running, 294 sleeping,   0 stopped,   0 zombie
%Cpu(s):  1.9 us,  1.0 sy,  0.0 ni, 97.0 id,  0.1 wa,  0.0 hi,  0.1 si,  0.0 st
KiB Mem :  8174732 total,   379256 free,  5053288 used,  2742188 buff/cache
KiB Swap:  2096124 total,  1835344 free,   260780 used.  2644160 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
3168863 root      20   0   40656   3860   3176 R  23.5  0.0   0:00.06 top
      1 root      20   0   38036   4896   3108 S   0.0  0.1   1:07.29 systemd
```

(continues on next page)

(continued from previous page)

```
      2 root      20   0       0      0      0 S   0.0  0.0   0:00.41 kthreadd
      3 root      20   0       0      0      0 S   0.0  0.0   0:35.52 ksoftirq+
      7 root      20   0       0      0      0 S   0.0  0.0   6:54.69 rcu_sched
      8 root      20   0       0      0      0 S   0.0  0.0   0:00.00 rcu_bh
      9 root      rt   0       0      0      0 S   0.0  0.0   0:08.10 migratio+
     10 root      rt   0       0      0      0 S   0.0  0.0   0:04.14 watchdog+
     11 root      rt   0       0      0      0 S   0.0  0.0   0:04.03 watchdog+
     12 root      rt   0       0      0      0 S   0.0  0.0   0:07.98 migratio+
     13 root      20   0       0      0      0 S   0.0  0.0   0:34.08 ksoftirq+
[...]
```

diag vmstat

```
platform@VOSS:~$ diag vmstat
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 1  0 260780 378512 273412 2468844    0    0    45     8    4    6  2  1 97  0  0
```

18.4. SNMP Trap Examples

18.4.1. SNMP Traps: Applications examples

- ProcessError
- ProcessRestart
- ProcessStart
- ProcessStop
- ProcessWarning

```
Oct 10 21:01:28 robot-slave snmptrapd[480]: 2022-10-10 21:01:28 <UNKNOWN> [UDP: [192.168.
↪100.3]:52036->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (79611) 0:13:16.11
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application start command completed snmp"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:07:36 robot-slave snmptrapd[480]: 2022-10-10 21:07:36 <UNKNOWN> [UDP: [192.168.
↪100.3]:52437->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (116410) 0:19:24.10
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application start command completed␣
```

(continues on next page)

(continued from previous page)

```
↪services:syslog"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:29:27 robot-slave snmptrapd[480]: 2022-10-10 21:29:27 <UNKNOWN> [UDP: [192.168.
↪100.3]:49190->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (247462) 0:41:14.62
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application start command completed services:time
↪"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:44:33 robot-slave snmptrapd[480]: 2022-10-10 21:44:33 <UNKNOWN> [UDP: [192.168.
↪100.3]:22489->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (338098) 0:56:20.98
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application start command completed mongodb"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:44:52 robot-slave snmptrapd[480]: 2022-10-10 21:44:52 <UNKNOWN> [UDP: [192.168.
↪100.3]:55617->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (340005) 0:56:40.05
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application start command completed nginx"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:46:59 robot-slave snmptrapd[480]: 2022-10-10 21:46:59 <UNKNOWN> [UDP: [192.168.
↪100.3]:19054->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (352745) 0:58:47.45
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application start command completed␣
↪services:firewall"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:48:37 robot-slave snmptrapd[480]: 2022-10-10 21:48:37 <UNKNOWN> [UDP: [192.168.
↪100.3]:58314->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (362467) 1:00:24.67
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application start command completed␣
```

(continues on next page)

(continued from previous page)

```
↪mongodb:database"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:52:51 robot-slave snmptrapd[480]: 2022-10-10 21:52:51 <UNKNOWN> [UDP: [192.168.
↪100.3]:20207->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (387871) 1:04:38.71
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application start command completed voss-
↪deviceapi"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:56:20 robot-slave snmptrapd[480]: 2022-10-10 21:56:20 <UNKNOWN> [UDP: [192.168.
↪100.3]:52129->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (408764) 1:08:07.64
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application start command completed selfservice"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:23:43 robot-slave snmptrapd[480]: 2022-10-10 22:23:43 <UNKNOWN> [UDP: [192.168.
↪100.3]:26646->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (573107) 1:35:31.07
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessRestart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Applications are restarting services:firewall"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:23:45 robot-slave snmptrapd[480]: 2022-10-10 22:23:45 <UNKNOWN> [UDP: [192.168.
↪100.3]:61828->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (573306) 1:35:33.06
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed services:firewall"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:28:06 robot-slave snmptrapd[480]: 2022-10-10 22:28:06 <UNKNOWN> [UDP: [192.168.
↪100.3]:34679->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (599418) 1:39:54.18
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStop"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Applications are stopping mongodb"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:28:08 robot-slave snmptrapd[480]: 2022-10-10 22:28:08 <UNKNOWN> [UDP: [192.168.
↪100.3]:40866->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (599581) 1:39:55.81
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed mongodb:arbiter"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:28:08 robot-slave snmptrapd[480]: 2022-10-10 22:28:08 <UNKNOWN> [UDP: [192.168.
↪100.3]:11041->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (599586) 1:39:55.86
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed mongodb:database"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:28:36 robot-slave snmptrapd[480]: 2022-10-10 22:28:36 <UNKNOWN> [UDP: [192.168.
↪100.3]:25020->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (602427) 1:40:24.27
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessRestart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Applications are restarting mongodb"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:32:17 robot-slave snmptrapd[480]: 2022-10-10 22:32:17 <UNKNOWN> [UDP: [192.168.
↪100.3]:43021->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (624584) 1:44:05.84
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application start command completed phone-based-
↪registration"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:50:10 robot-slave snmptrapd[480]: 2022-10-10 22:50:10 <UNKNOWN> [UDP: [192.168.
↪100.3]:6672->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (731774) 2:01:57.74
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application start command completed voss-queue"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:00:16 robot-slave snmptrapd[480]: 2022-10-11 00:00:16 <UNKNOWN> [UDP: [192.168.
↪100.3]:62561->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1152364) 3:12:03.64
```

(continues on next page)

(continued from previous page)

```
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed nginx:proxy"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:00:34 robot-slave snmptrapd[480]: 2022-10-11 00:00:34 <UNKNOWN> [UDP: [192.168.
↪100.3]:48544->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1154165) 3:12:21.65
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessError"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application failed with error 1 nginx:proxy"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:12:33 robot-slave snmptrapd[480]: 2022-10-11 00:12:33 <UNKNOWN> [UDP: [192.168.
↪100.3]:29021->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1226146) 3:24:21.46
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessRestart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Applications are restarting snmp"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:12:38 robot-slave snmptrapd[480]: 2022-10-11 00:12:38 <UNKNOWN> [UDP: [192.168.
↪100.3]:10991->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1226585) 3:24:25.85
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed snmp:daemon"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:12:38 robot-slave snmptrapd[480]: 2022-10-11 00:12:38 <UNKNOWN> [UDP: [192.168.
↪100.3]:21681->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1226610) 3:24:26.10
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed snmp:traps"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:12:39 robot-slave snmptrapd[480]: 2022-10-11 00:12:39 <UNKNOWN> [UDP: [192.168.
↪100.3]:7756->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1226681) 3:24:26.81
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application has changed its state from stopped␣
```

(continues on next page)

(continued from previous page)

```
↪(killed) to unknown (missing launcher,killed) - snmp:traps"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:20:20 robot-slave snmptrapd[480]: 2022-10-11 00:20:20 <UNKNOWN> [UDP: [192.168.
↪100.3]:22009->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1272824) 3:32:08.24
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessRestart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Applications are restarting voss-portal"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:20:28 robot-slave snmptrapd[480]: 2022-10-11 00:20:28 <UNKNOWN> [UDP: [192.168.
↪100.3]:44656->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1273568) 3:32:15.68
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStop"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Applications are stopping voss-portal"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:20:29 robot-slave snmptrapd[480]: 2022-10-11 00:20:29 <UNKNOWN> [UDP: [192.168.
↪100.3]:62367->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1273684) 3:32:16.84
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed voss-portal:gui"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:20:38 robot-slave snmptrapd[480]: 2022-10-11 00:20:38 <UNKNOWN> [UDP: [192.168.
↪100.3]:3228->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1274591) 3:32:25.91
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessError"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application failed with error 200 voss-portal:gui
↪"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:20:39 robot-slave snmptrapd[480]: 2022-10-11 00:20:39 <UNKNOWN> [UDP: [192.168.
↪100.3]:6835->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1274668) 3:32:26.68
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application start command completed voss-portal"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:22:01 robot-slave snmptrapd[480]: 2022-10-11 00:22:01 <UNKNOWN> [UDP: [192.168.
↪100.3]:56285->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1282854) 3:33:48.54
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessRestart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Applications are restarting "
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:22:06 robot-slave snmptrapd[480]: 2022-10-11 00:22:06 <UNKNOWN> [UDP: [192.168.
↪100.3]:22442->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1283425) 3:33:54.25
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed selenium:gui_orchestration
↪"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:22:12 robot-slave snmptrapd[480]: 2022-10-11 00:22:12 <UNKNOWN> [UDP: [192.168.
↪100.3]:12754->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1283985) 3:33:59.85
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed selfservice:node"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:22:19 robot-slave snmptrapd[480]: 2022-10-11 00:22:19 <UNKNOWN> [UDP: [192.168.
↪100.3]:53711->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1284725) 3:34:07.25
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed services:logs"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:22:20 robot-slave snmptrapd[480]: 2022-10-11 00:22:20 <UNKNOWN> [UDP: [192.168.
↪100.3]:19232->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1284769) 3:34:07.69
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed voss-deviceapi:voss-
↪risapi_collector"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:22:20 robot-slave snmptrapd[480]: 2022-10-11 00:22:20 <UNKNOWN> [UDP: [192.168.
↪100.3]:61419->[192.168.100.25]:162]:
```

(continues on next page)

(continued from previous page)

```
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1284805) 3:34:08.05
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed voss-deviceapi:voss-wsgi"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:22:20 robot-slave snmptrapd[480]: 2022-10-11 00:22:20 <UNKNOWN> [UDP: [192.168.
↪100.3]:56372->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1284812) 3:34:08.12
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed voss-deviceapi:voss-queue"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:22:20 robot-slave snmptrapd[480]: 2022-10-11 00:22:20 <UNKNOWN> [UDP: [192.168.
↪100.3]:59666->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1284830) 3:34:08.30
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed voss-deviceapi:voss-
↪monitoring"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:22:20 robot-slave snmptrapd[480]: 2022-10-11 00:22:20 <UNKNOWN> [UDP: [192.168.
↪100.3]:35828->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1284829) 3:34:08.29
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed voss-deviceapi:voss-cnf_
↪collector"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:22:21 robot-slave snmptrapd[480]: 2022-10-11 00:22:21 <UNKNOWN> [UDP: [192.168.
↪100.3]:11890->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1284895) 3:34:08.95
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed services:time"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:22:23 robot-slave snmptrapd[480]: 2022-10-11 00:22:23 <UNKNOWN> [UDP: [192.168.
↪100.3]:24541->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1285134) 3:34:11.34
```

(continues on next page)

(continued from previous page)

```
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application has changed its state from stopped␣
↪(killed) to unknown (missing launcher,missing pid,killed) - services:time"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:22:28 robot-slave snmptrapd[480]: 2022-10-11 00:22:28 <UNKNOWN> [UDP: [192.168.
↪100.3]:4536->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1285611) 3:34:16.11
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed phone-based-
↪registration:nodeservice"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:22:43 robot-slave snmptrapd[480]: 2022-10-11 00:22:43 <UNKNOWN> [UDP: [192.168.
↪100.3]:24351->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1287104) 3:34:31.04
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed services:syslog"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:22:43 robot-slave snmptrapd[480]: 2022-10-11 00:22:43 <UNKNOWN> [UDP: [192.168.
↪100.3]:6262->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1287112) 3:34:31.12
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed services:mount"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:22:43 robot-slave snmptrapd[480]: 2022-10-11 00:22:43 <UNKNOWN> [UDP: [192.168.
↪100.3]:52368->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1287125) 3:34:31.25
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed services:scheduler"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:22:45 robot-slave snmptrapd[480]: 2022-10-11 00:22:45 <UNKNOWN> [UDP: [192.168.
↪100.3]:20331->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1287320) 3:34:33.20
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
```

(continues on next page)

(continued from previous page)

```
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application has changed its state from stopped␣
↪(killed) to unknown (missing launcher,killed) - services:mount"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:22:45 robot-slave snmptrapd[480]: 2022-10-11 00:22:45 <UNKNOWN> [UDP: [192.168.
↪100.3]:50365->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1287322) 3:34:33.22
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application has changed its state from stopped␣
↪(killed) to unknown (missing launcher,killed) - services:syslog"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:23:02 robot-slave snmptrapd[480]: 2022-10-11 00:23:02 <UNKNOWN> [UDP: [192.168.
↪100.3]:52355->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1288998) 3:34:49.98
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application has changed its state from stopped␣
↪(secure tunnel died) to running (secure tunnel died) - services:wsgi"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:23:13 robot-slave snmptrapd[480]: 2022-10-11 00:23:13 <UNKNOWN> [UDP: [192.168.
↪100.3]:26791->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1290061) 3:35:00.61
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application has changed its state from starting␣
↪to running - snmp:daemon"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:23:13 robot-slave snmptrapd[480]: 2022-10-11 00:23:13 <UNKNOWN> [UDP: [192.168.
↪100.3]:47805->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1290070) 3:35:00.70
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application has changed its state from starting␣
↪to running - voss-deviceapi:voss-risapi_collector"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:23:14 robot-slave snmptrapd[480]: 2022-10-11 00:23:14 <UNKNOWN> [UDP: [192.168.
↪100.3]:60807->[192.168.100.25]:162]:
```

(continues on next page)

(continued from previous page)

```
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1290203) 3:35:02.03
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application has changed its state from starting␣
↪to unknown (missing launcher) - snmp:traps"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:23:14 robot-slave snmptrapd[480]: 2022-10-11 00:23:14 <UNKNOWN> [UDP: [192.168.
↪100.3]:25702->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1290221) 3:35:02.21
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application has changed its state from starting␣
↪to unknown (missing launcher) - services:logs"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:23:14 robot-slave snmptrapd[480]: 2022-10-11 00:23:14 <UNKNOWN> [UDP: [192.168.
↪100.3]:60314->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1290239) 3:35:02.39
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application has changed its state from starting␣
↪to running - voss-deviceapi:voss-queue"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:23:16 robot-slave snmptrapd[480]: 2022-10-11 00:23:16 <UNKNOWN> [UDP: [192.168.
↪100.3]:33659->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1290445) 3:35:04.45
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application has changed its state from starting␣
↪to unknown (missing launcher,missing pid) - services:time"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:23:33 robot-slave snmptrapd[480]: 2022-10-11 00:23:33 <UNKNOWN> [UDP: [192.168.
↪100.3]:29625->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1292068) 3:35:20.68
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application start command completed "
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:23:57 robot-slave snmptrapd[480]: 2022-10-11 00:23:57 <UNKNOWN> [UDP: [192.168.
```

(continues on next page)

(continued from previous page)

```
↪100.3]:47870->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1294518) 3:35:45.18
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStop"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Applications are stopping snmp"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:24:19 robot-slave snmptrapd[480]: 2022-10-11 00:24:19 <UNKNOWN> [UDP: [192.168.
↪100.3]:4081->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1296686) 3:36:06.86
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application has changed its state from stopped␣
↪(killed) to unknown (missing launcher,killed) - snmp:daemon"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

18.4.2. SNMP Traps: Load examples

```
Oct 10 21:53:21 robot-slave snmptrapd[480]: 2022-10-10 21:53:21 <UNKNOWN> [UDP: [192.168.
↪100.3]:11885->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (390954) 1:05:09.54
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Excessive load"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.10.1.100.2
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.10.1.2.2 = STRING: "Load-5"
#011iso.3.6.1.4.1.2021.10.1.101.2 = STRING: "5 min Load Average too high (= 2.20)"
```

```
Oct 10 22:00:34 robot-slave snmptrapd[480]: 2022-10-10 22:00:34 <UNKNOWN> [UDP: [192.168.
↪100.3]:22092->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (434192) 1:12:21.92
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Excessive load"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.10.1.100.1
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.10.1.2.1 = STRING: "Load-1"
#011iso.3.6.1.4.1.2021.10.1.101.1 = STRING: "1 min Load Average too high (= 4.27)"
```

```
Oct 10 22:10:18 robot-slave snmptrapd[480]: 2022-10-10 22:10:18 <UNKNOWN> [UDP: [192.168.
↪100.3]:4006->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (492563) 1:22:05.63
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Excessive load"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.10.1.100.2
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.10.1.2.2 = STRING: "Load-5"
#011iso.3.6.1.4.1.2021.10.1.101.2 = STRING: "5 min Load Average too high (= 2.27)"
```

```
Oct 10 22:30:14 robot-slave snmptrapd[480]: 2022-10-10 22:30:14 <UNKNOWN> [UDP: [192.168.
↪100.3]:48416->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (612257) 1:42:02.57
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Excessive load"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.10.1.100.2
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.10.1.2.2 = STRING: "Load-5"
#011iso.3.6.1.4.1.2021.10.1.101.2 = STRING: "5 min Load Average too high (= 2.14)"
```

```
Oct 10 22:31:19 robot-slave snmptrapd[480]: 2022-10-10 22:31:14 <UNKNOWN> [UDP: [192.168.
↪100.3]:64638->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (618257) 1:43:02.57
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Excessive load"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.10.1.100.1
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.10.1.2.1 = STRING: "Load-1"
#011iso.3.6.1.4.1.2021.10.1.101.1 = STRING: "1 min Load Average too high (= 4.52)"
```

```
Oct 10 22:53:14 robot-slave snmptrapd[480]: 2022-10-10 22:53:14 <UNKNOWN> [UDP: [192.168.
↪100.3]:55096->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (750263) 2:05:02.63
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Excessive load"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.10.1.100.2
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.10.1.2.2 = STRING: "Load-5"
#011iso.3.6.1.4.1.2021.10.1.101.2 = STRING: "5 min Load Average too high (= 2.06)"
```

```
Oct 10 22:54:14 robot-slave snmptrapd[480]: 2022-10-10 22:54:14 <UNKNOWN> [UDP: [192.168.
↪100.3]:19700->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (756261) 2:06:02.61
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Excessive load"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.10.1.100.1
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.10.1.2.1 = STRING: "Load-1"
#011iso.3.6.1.4.1.2021.10.1.101.1 = STRING: "1 min Load Average too high (= 5.15)"
```

```
Oct 10 23:48:15 robot-slave snmptrapd[480]: 2022-10-10 23:48:15 <UNKNOWN> [UDP: [192.168.
↪100.3]:39067->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1080266) 3:00:02.66
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Excessive load"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.10.1.100.1
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.10.1.2.1 = STRING: "Load-1"
#011iso.3.6.1.4.1.2021.10.1.101.1 = STRING: "1 min Load Average too high (= 5.83)"
```

```
Oct 10 23:48:15 robot-slave snmptrapd[480]: 2022-10-10 23:48:15 <UNKNOWN> [UDP: [192.168.
↪100.3]:61814->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1080302) 3:00:03.02
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Excessive load"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.10.1.100.2
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.10.1.2.2 = STRING: "Load-5"
#011iso.3.6.1.4.1.2021.10.1.101.2 = STRING: "5 min Load Average too high (= 2.66)"
```

```
Oct 10 23:50:15 robot-slave snmptrapd[480]: 2022-10-10 23:50:15 <UNKNOWN> [UDP: [192.168.
↪100.3]:48240->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1092279) 3:02:02.79
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Excessive load"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.10.1.100.1
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.10.1.2.1 = STRING: "Load-1"
#011iso.3.6.1.4.1.2021.10.1.101.1 = STRING: "1 min Load Average too high (= 7.25)"
```

```
Oct 11 00:00:27 robot-slave snmptrapd[480]: 2022-10-11 00:00:27 <UNKNOWN> [UDP: [192.168.
↪100.3]:38716->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1153470) 3:12:14.70
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Excessive load"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.10.1.100.1
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.10.1.2.1 = STRING: "Load-1"
#011iso.3.6.1.4.1.2021.10.1.101.1 = STRING: "1 min Load Average too high (= 4.54)"
```

```
Oct 11 00:13:48 robot-slave snmptrapd[480]: 2022-10-11 00:13:48 <UNKNOWN> [UDP: [192.168.
↪100.3]:10511->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1233585) 3:25:35.85
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Excessive load"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.10.1.100.1
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.10.1.2.1 = STRING: "Load-1"
#011iso.3.6.1.4.1.2021.10.1.101.1 = STRING: "1 min Load Average too high (= 5.59)"
```

```
Oct 11 00:13:48 robot-slave snmptrapd[480]: 2022-10-11 00:13:48 <UNKNOWN> [UDP: [192.168.
↪100.3]:16438->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1233630) 3:25:36.30
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Excessive load"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.10.1.100.2
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.10.1.2.2 = STRING: "Load-5"
#011iso.3.6.1.4.1.2021.10.1.101.2 = STRING: "5 min Load Average too high (= 2.43)"
```

```
Oct 11 00:15:48 robot-slave snmptrapd[480]: 2022-10-11 00:15:48 <UNKNOWN> [UDP: [192.168.
↪100.3]:43788->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1245577) 3:27:35.77
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Excessive load"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.10.1.100.3
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.10.1.2.3 = STRING: "Load-15"
#011iso.3.6.1.4.1.2021.10.1.101.3 = STRING: "15 min Load Average too high (= 3.14)"
```

18.4.3. SNMP Examples: Backup

```
Oct 10 21:39:17 robot-slave snmptrapd[480]: 2022-10-10 21:39:17 <UNKNOWN> [UDP: [192.168.
↪100.3]:59765->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (306489) 0:51:04.89
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: Backup completed"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Backup was successfully created at localbackup"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:39:27 robot-slave snmptrapd[480]: 2022-10-10 21:39:27 <UNKNOWN> [UDP: [192.168.
↪100.3]:25962->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (307540) 0:51:15.40
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: Backups now runs regularly"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Backup list:
#012
#012    localbackup:
#012        backups:
#012            1 backups have been created - most recently 2022-10-10 19:38
#012        uri: file:///backups
#012
#012
#012
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:39:30 robot-slave snmptrapd[480]: 2022-10-10 21:39:30 <UNKNOWN> [UDP: [192.168.
↪100.3]:14347->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (307808) 0:51:18.08
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: The last backup was more than 2 days ago"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Backup list:
#012
#012    localbackup:
#012        backups:
#012            1 backups have been created - most recently 2022-10-10 19:38
#012        uri: file:///backups
#012
#012
#012
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:40:06 robot-slave snmptrapd[480]: 2022-10-10 21:40:06 <UNKNOWN> [UDP: [192.168.
↪100.3]:53579->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (311359) 0:51:53.59
```

(continues on next page)

(continued from previous page)

```
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "backup.py : num_sockets exeeded warning␣
↪threshold of 6 with 7"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Command backup.py attribute num_sockets
#012Current average value 7
#012Error threshold 10
#012Warning threshold 6
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:42:01 robot-slave snmptrapd[480]: 2022-10-10 21:42:01 <UNKNOWN> [UDP: [192.168.
↪100.3]:7560->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (322901) 0:53:49.01
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: Backup completed"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Backup was successfully created at sftpbackup"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:17:09 robot-slave snmptrapd[480]: 2022-10-10 22:17:09 <UNKNOWN> [UDP: [192.168.
↪100.3]:36106->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (533696) 1:28:56.96
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "Backup failed"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Backup failed - error running pre-backup method␣
↪for mongodb"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:22:14 robot-slave snmptrapd[480]: 2022-10-10 22:22:14 <UNKNOWN> [UDP: [192.168.
↪100.3]:21378->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (564173) 1:34:01.73
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "Backup failed"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Backup to localbackup failed to verify"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

18.4.4. SNMP Trap Examples

```
Oct 10 21:00:58 robot-slave snmptrapd[480]: 2022-10-10 21:00:58 <UNKNOWN> [UDP: [192.168.
↪100.3]:3250->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (76610) 0:12:46.10
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "Test notifications"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "This is a notification test email."
```

(continues on next page)

(continued from previous page)

```
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:01:28 robot-slave snmptrapd[480]: 2022-10-10 21:01:28 <UNKNOWN> [UDP: [192.168.
↪100.3]:52036->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (79611) 0:13:16.11
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application start command completed snmp"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:05:39 robot-slave snmptrapd[480]: 2022-10-10 21:05:39 <UNKNOWN> [UDP: [192.168.
↪100.3]:8550->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (104723) 0:17:27.23
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: All notify levels is now configured with␣
↪an external email address"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "    notifications:
#012        emailrelay: 127.0.0.1
#012        level:
#012            error:
#012                mailto:platform@localhost
#012                snmp://public@192.168.100.25
#012            info:
#012          Message is truncated"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:07:36 robot-slave snmptrapd[480]: 2022-10-10 21:07:36 <UNKNOWN> [UDP: [192.168.
↪100.3]:52437->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (116410) 0:19:24.10
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application start command completed␣
↪services:syslog"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:17:21 robot-slave snmptrapd[480]: 2022-10-10 21:17:21 <UNKNOWN> [UDP: [192.168.
↪100.3]:6743->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (174854) 0:29:08.54
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "Log processing failure"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "ERROR: System unable to send event messages to␣
↪192.168.100.30"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:27:19 robot-slave snmptrapd[480]: 2022-10-10 21:27:19 <UNKNOWN> [UDP: [192.168.
↪100.3]:64657->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (234733) 0:39:07.33
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: The total local messages for platform is␣
↪now under 200"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "The total local messages is now under 200."
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:27:29 robot-slave snmptrapd[480]: 2022-10-10 21:27:29 <UNKNOWN> [UDP: [192.168.
↪100.3]:5434->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (235697) 0:39:16.97
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "WARNING: The total messages in the local mailbox␣
↪for platform has reached in excess of 200"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Use the following mail commands to manage the␣
↪local mailbox:
#012
#012mail del <number>      - delete the selected mail
#012mail del <from
#> <to
#> - deletes the selected range of mail message
#012mail del all          Message is truncated"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:27:46 robot-slave snmptrapd[480]: 2022-10-10 21:27:46 <UNKNOWN> [UDP: [192.168.
↪100.3]:32395->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (237369) 0:39:33.69
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: Messages for platform auto archived as it␣
↪reached more than 500"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Use the following command to view archived␣
↪messages:
#012
#012log view /var/log/platform/archived_mail.log"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:28:15 robot-slave snmptrapd[480]: 2022-10-10 21:28:15 <UNKNOWN> [UDP: [192.168.
↪100.3]:62399->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (240260) 0:40:02.60
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: No ntp configured for VOSS-192.168.100.3"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "It is mandatory that the ntp is configured.
#012
#012To configure ntp use the following command:
#012network ntp <server1> <server2>"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:28:23 robot-slave snmptrapd[480]: 2022-10-10 21:28:23 <UNKNOWN> [UDP: [192.168.
↪100.3]:40886->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (241068) 0:40:10.68
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: ntp is now configured for VOSS-192.168.100.
↪3"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "NTP cleared"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:29:12 robot-slave snmptrapd[480]: 2022-10-10 21:29:12 <UNKNOWN> [UDP: [192.168.
↪100.3]:41584->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (246013) 0:41:00.13
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "WARNING: The ntp daemon has stopped on VOSS-192.
↪168.100.3"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Run 'app start services:time' to restart ntpd"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:29:27 robot-slave snmptrapd[480]: 2022-10-10 21:29:27 <UNKNOWN> [UDP: [192.168.
↪100.3]:49190->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (247462) 0:41:14.62
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application start command completed services:time
↪"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:29:44 robot-slave snmptrapd[480]: 2022-10-10 21:29:44 <UNKNOWN> [UDP: [192.168.
↪100.3]:64806->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (249163) 0:41:31.63
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: The ntp daemon is now running on VOSS-192.
↪168.100.3"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "ntp daemon running"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:30:04 robot-slave snmptrapd[480]: 2022-10-10 21:30:04 <UNKNOWN> [UDP: [192.168.
↪100.3]:63587->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (251207) 0:41:52.07
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "rsyslogd : num_files exceeded maximum value of␣
↪30 with 78"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Command rsyslogd attribute num_files
#012Current maximum value 78
#012Error threshold 50
#012Warning threshold 33
```

(continues on next page)

(continued from previous page)

```
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:30:04 robot-slave snmptrapd[480]: 2022-10-10 21:30:04 <UNKNOWN> [UDP: [192.168.
↪100.3]:63524->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (251245) 0:41:52.45
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "sysmon : num_sockets exceeded maximum value of␣
↪10 with 11"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Command sysmon attribute num_sockets
#012Current maximum value 11
#012Error threshold 10
#012Warning threshold 6
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:31:32 robot-slave snmptrapd[480]: 2022-10-10 21:31:32 <UNKNOWN> [UDP: [192.168.
↪100.3]:21656->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (260016) 0:43:20.16
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "WARNING: No dns configured for VOSS-192.168.100.3
↪"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "It is recommended that the dns is configured.
#012
#012To configure dns use the following command:
#012network dns <server1> <server2>"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:32:51 robot-slave snmptrapd[480]: 2022-10-10 21:32:51 <UNKNOWN> [UDP: [192.168.
↪100.3]:48247->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (267891) 0:44:38.91
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: dns is now configured for VOSS-192.168.100.
↪3"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "DNS cleared"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:33:56 robot-slave snmptrapd[480]: 2022-10-10 21:33:56 <UNKNOWN> [UDP: [192.168.
↪100.3]:36367->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (274443) 0:45:44.43
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: domain is now configured for VOSS-192.168.
↪100.3"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Domain cleared"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
```

(continues on next page)

(continued from previous page)

```
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:34:00 robot-slave snmptrapd[480]: 2022-10-10 21:34:00 <UNKNOWN> [UDP: [192.168.
↪100.3]:50982->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (274843) 0:45:48.43
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "WARNING: No domain configured for VOSS-192.168.
↪100.3"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "It is recommended that the domain is configured.
#012
#012To configure the domain use the following command:
#012network domain <domain-name>"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:39:17 robot-slave snmptrapd[480]: 2022-10-10 21:39:17 <UNKNOWN> [UDP: [192.168.
↪100.3]:59765->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (306489) 0:51:04.89
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: Backup completed"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Backup was successfully created at localbackup"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:39:27 robot-slave snmptrapd[480]: 2022-10-10 21:39:27 <UNKNOWN> [UDP: [192.168.
↪100.3]:25962->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (307540) 0:51:15.40
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: Backups now runs regularly"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Backup list:
#012
#012    localbackup:
#012        backups:
#012            1 backups have been created - most recently 2022-10-10 19:38
#012        uri: file:///backups
#012
#012
#012
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:39:30 robot-slave snmptrapd[480]: 2022-10-10 21:39:30 <UNKNOWN> [UDP: [192.168.
↪100.3]:14347->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (307808) 0:51:18.08
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: The last backup was more than 2 days ago"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Backup list:
#012
```

(continues on next page)

(continued from previous page)

```
#012    localbackup:
#012        backups:
#012            1 backups have been created - most recently 2022-10-10 19:38
#012        uri: file:///backups
#012
#012
#012
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:40:05 robot-slave snmptrapd[480]: 2022-10-10 21:40:05 <UNKNOWN> [UDP: [192.168.
↪100.3]:2870->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (311324) 0:51:53.24
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "rsyslogd : num_files exceeded maximum value of␣
↪78 with 87"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Command rsyslogd attribute num_files
#012Current maximum value 87
#012Error threshold 50
#012Warning threshold 33
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:40:06 robot-slave snmptrapd[480]: 2022-10-10 21:40:06 <UNKNOWN> [UDP: [192.168.
↪100.3]:53579->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (311359) 0:51:53.59
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "backup.py : num_sockets exeeded warning␣
↪threshold of 6 with 7"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Command backup.py attribute num_sockets
#012Current average value 7
#012Error threshold 10
#012Warning threshold 6
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:42:01 robot-slave snmptrapd[480]: 2022-10-10 21:42:01 <UNKNOWN> [UDP: [192.168.
↪100.3]:7560->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (322901) 0:53:49.01
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: Backup completed"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Backup was successfully created at sftpbackup"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:44:33 robot-slave snmptrapd[480]: 2022-10-10 21:44:33 <UNKNOWN> [UDP: [192.168.
```

(continues on next page)

(continued from previous page)

```
↪100.3]:22489->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (338098) 0:56:20.98
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application start command completed mongodb"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:44:52 robot-slave snmptrapd[480]: 2022-10-10 21:44:52 <UNKNOWN> [UDP: [192.168.
↪100.3]:55617->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (340005) 0:56:40.05
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application start command completed nginx"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:46:59 robot-slave snmptrapd[480]: 2022-10-10 21:46:59 <UNKNOWN> [UDP: [192.168.
↪100.3]:19054->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (352745) 0:58:47.45
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application start command completed␣
↪services:firewall"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:48:37 robot-slave snmptrapd[480]: 2022-10-10 21:48:37 <UNKNOWN> [UDP: [192.168.
↪100.3]:58314->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (362467) 1:00:24.67
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application start command completed␣
↪mongodb:database"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:52:51 robot-slave snmptrapd[480]: 2022-10-10 21:52:51 <UNKNOWN> [UDP: [192.168.
↪100.3]:20207->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (387871) 1:04:38.71
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application start command completed voss-
↪deviceapi"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:53:21 robot-slave snmptrapd[480]: 2022-10-10 21:53:21 <UNKNOWN> [UDP: [192.168.
```

(continues on next page)

(continued from previous page)

```
↪100.3]:11885->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (390954) 1:05:09.54
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Excessive load"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.10.1.100.2
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.10.1.2.2 = STRING: "Load-5"
#011iso.3.6.1.4.1.2021.10.1.101.2 = STRING: "5 min Load Average too high (= 2.20)"
```

```
Oct 10 21:56:20 robot-slave snmptrapd[480]: 2022-10-10 21:56:20 <UNKNOWN> [UDP: [192.168.
↪100.3]:52129->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (408764) 1:08:07.64
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application start command completed selfservice"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 21:58:02 robot-slave snmptrapd[480]: 2022-10-10 21:58:02 <UNKNOWN> [UDP: [192.168.
↪100.3]:21964->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (419022) 1:09:50.22
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: Restore successful"
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:00:16 robot-slave snmptrapd[480]: 2022-10-10 22:00:16 <UNKNOWN> [UDP: [192.168.
↪100.3]:40320->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (432428) 1:12:04.28
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "docker-proxy : num_sockets exceeded maximum␣
↪value of 7 with 27"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Command docker-proxy attribute num_sockets
#012Current maximum value 27
#012Error threshold 20
#012Warning threshold 13
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:00:17 robot-slave snmptrapd[480]: 2022-10-10 22:00:17 <UNKNOWN> [UDP: [192.168.
↪100.3]:39273->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (432500) 1:12:05.00
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "docker-proxy : num_files exceeded maximum value␣
```

(continues on next page)

(continued from previous page)

```
↪of 17 with 57"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Command docker-proxy attribute num_files
#012Current maximum value 57
#012Error threshold 20
#012Warning threshold 13
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:00:34 robot-slave snmptrapd[480]: 2022-10-10 22:00:34 <UNKNOWN> [UDP: [192.168.
↪100.3]:22092->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (434192) 1:12:21.92
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Excessive load"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.10.1.100.1
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.10.1.2.1 = STRING: "Load-1"
#011iso.3.6.1.4.1.2021.10.1.101.1 = STRING: "1 min Load Average too high (= 4.27)"
```

```
Oct 10 22:01:33 robot-slave snmptrapd[480]: 2022-10-10 22:01:33 <UNKNOWN> [UDP: [192.168.
↪100.3]:17055->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (440135) 1:13:21.35
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Service Failures"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "selfservice v22.1.0 (2022-10-10 13:57)
#012   |-node                  running
#012voss-deviceapi v22.1.0 (2022-10-10 13:57)
#012   |-voss-cnf_collector    running
#012   |-voss-queue            running
#012   |-voss-risa  Message is truncated"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:10:18 robot-slave snmptrapd[480]: 2022-10-10 22:10:18 <UNKNOWN> [UDP: [192.168.
↪100.3]:4006->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (492563) 1:22:05.63
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Excessive load"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.10.1.100.2
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.10.1.2.2 = STRING: "Load-5"
#011iso.3.6.1.4.1.2021.10.1.101.2 = STRING: "5 min Load Average too high (= 2.27)"
```

```
Oct 10 22:14:25 robot-slave snmptrapd[480]: 2022-10-10 22:14:25 <UNKNOWN> [UDP: [192.168.
↪100.3]:27277->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (517261) 1:26:12.61
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: Restore successful"
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:17:09 robot-slave snmptrapd[480]: 2022-10-10 22:17:09 <UNKNOWN> [UDP: [192.168.
↪100.3]:36106->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (533696) 1:28:56.96
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "Backup failed"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Backup failed - error running pre-backup method␣
↪for mongodb"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:22:14 robot-slave snmptrapd[480]: 2022-10-10 22:22:14 <UNKNOWN> [UDP: [192.168.
↪100.3]:21378->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (564173) 1:34:01.73
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "Backup failed"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Backup to localbackup failed to verify"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:23:43 robot-slave snmptrapd[480]: 2022-10-10 22:23:43 <UNKNOWN> [UDP: [192.168.
↪100.3]:26646->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (573107) 1:35:31.07
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessRestart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Applications are restarting services:firewall"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:23:45 robot-slave snmptrapd[480]: 2022-10-10 22:23:45 <UNKNOWN> [UDP: [192.168.
↪100.3]:61828->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (573306) 1:35:33.06
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed services:firewall"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:28:06 robot-slave snmptrapd[480]: 2022-10-10 22:28:06 <UNKNOWN> [UDP: [192.168.
↪100.3]:34679->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (599418) 1:39:54.18
```

(continues on next page)

(continued from previous page)

```
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStop"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Applications are stopping mongodb"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:28:08 robot-slave snmptrapd[480]: 2022-10-10 22:28:08 <UNKNOWN> [UDP: [192.168.
↪100.3]:40866->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (599581) 1:39:55.81
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed mongodb:arbiter"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:28:08 robot-slave snmptrapd[480]: 2022-10-10 22:28:08 <UNKNOWN> [UDP: [192.168.
↪100.3]:11041->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (599586) 1:39:55.86
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed mongodb:database"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:28:36 robot-slave snmptrapd[480]: 2022-10-10 22:28:36 <UNKNOWN> [UDP: [192.168.
↪100.3]:25020->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (602427) 1:40:24.27
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessRestart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Applications are restarting mongodb"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:30:04 robot-slave snmptrapd[480]: 2022-10-10 22:30:04 <UNKNOWN> [UDP: [192.168.
↪100.3]:3488->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (611240) 1:41:52.40
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "rsyslogd : num_files exceeded maximum value of␣
↪87 with 88"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Command rsyslogd attribute num_files
#012Current maximum value 88
#012Error threshold 50
#012Warning threshold 33
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:30:04 robot-slave snmptrapd[480]: 2022-10-10 22:30:04 <UNKNOWN> [UDP: [192.168.
```

(continues on next page)

(continued from previous page)

```
↪100.3]:24967->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (611283) 1:41:52.83
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "docker-proxy : num_sockets exceeded maximum␣
↪value of 27 with 29"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Command docker-proxy attribute num_sockets
#012Current maximum value 29
#012Error threshold 20
#012Warning threshold 13
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:30:05 robot-slave snmptrapd[480]: 2022-10-10 22:30:05 <UNKNOWN> [UDP: [192.168.
↪100.3]:16104->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (611318) 1:41:53.18
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "docker-proxy : num_files exceeded maximum value␣
↪of 57 with 61"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Command docker-proxy attribute num_files
#012Current maximum value 61
#012Error threshold 20
#012Warning threshold 13
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:30:14 robot-slave snmptrapd[480]: 2022-10-10 22:30:14 <UNKNOWN> [UDP: [192.168.
↪100.3]:48416->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (612257) 1:42:02.57
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Excessive load"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.10.1.100.2
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.10.1.2.2 = STRING: "Load-5"
#011iso.3.6.1.4.1.2021.10.1.101.2 = STRING: "5 min Load Average too high (= 2.14)"
```

```
Oct 10 22:30:17 robot-slave snmptrapd[480]: 2022-10-10 22:30:17 <UNKNOWN> [UDP: [192.168.
↪100.3]:32373->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (612524) 1:42:05.24
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "upgrade failed"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "upgrade failed as other activity is in progress"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:31:19 robot-slave snmptrapd[480]: 2022-10-10 22:31:14 <UNKNOWN> [UDP: [192.168.
↪100.3]:64638->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (618257) 1:43:02.57
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Excessive load"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.10.1.100.1
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.10.1.2.1 = STRING: "Load-1"
#011iso.3.6.1.4.1.2021.10.1.101.1 = STRING: "1 min Load Average too high (= 4.52)"
```

```
Oct 10 22:32:17 robot-slave snmptrapd[480]: 2022-10-10 22:32:17 <UNKNOWN> [UDP: [192.168.
↪100.3]:43021->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (624584) 1:44:05.84
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application start command completed phone-based-
↪registration"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:40:03 robot-slave snmptrapd[480]: 2022-10-10 22:40:03 <UNKNOWN> [UDP: [192.168.
↪100.3]:62002->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (671171) 1:51:51.71
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "containerd : num_sockets exceeded maximum value␣
↪of 16 with 17"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Command containerd attribute num_sockets
#012Current maximum value 17
#012Error threshold 10
#012Warning threshold 6
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:43:20 robot-slave snmptrapd[480]: 2022-10-10 22:43:19 <UNKNOWN> [UDP: [192.168.
↪100.3]:41903->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (690787) 1:55:07.87
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "WARN: Database transaction count exceeded␣
↪threshold"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "count: 13332"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:43:24 robot-slave snmptrapd[480]: 2022-10-10 22:43:24 <UNKNOWN> [UDP: [192.168.
↪100.3]:25484->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (691198) 1:55:11.98
```

(continues on next page)

(continued from previous page)

```
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: Database transaction count returned to␣
↪normal"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "The transaction count returned to normal."
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:43:28 robot-slave snmptrapd[480]: 2022-10-10 22:43:28 <UNKNOWN> [UDP: [192.168.
↪100.3]:11973->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (691609) 1:55:16.09
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "WARN: Database transaction size exceeded␣
↪threshold"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "size: 32264767"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:43:32 robot-slave snmptrapd[480]: 2022-10-10 22:43:32 <UNKNOWN> [UDP: [192.168.
↪100.3]:9468->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (692019) 1:55:20.19
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: Database transaction size returned to␣
↪normal"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "The transaction size returned to normal."
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:44:14 robot-slave snmptrapd[480]: 2022-10-10 22:44:14 <UNKNOWN> [UDP: [192.168.
↪100.3]:19797->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (696202) 1:56:02.02
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "TransactionExport"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Transaction export failed"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:46:06 robot-slave snmptrapd[480]: 2022-10-10 22:46:06 <UNKNOWN> [UDP: [192.168.
↪100.3]:33539->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (707420) 1:57:54.20
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: License file generation failed"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "The license audit report scheduled for October␣
↪2022 was not successful.
#012Please contact your VOSS account manager. "
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:50:03 robot-slave snmptrapd[480]: 2022-10-10 22:50:03 <UNKNOWN> [UDP: [192.168.
```

(continues on next page)

(continued from previous page)

```
↪100.3]:34433->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (731163) 2:01:51.63
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "execute : num_sockets exeeded warning threshold␣
↪of 6 with 6"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Command execute attribute num_sockets
#012Current average value 6
#012Error threshold 10
#012Warning threshold 6
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:50:10 robot-slave snmptrapd[480]: 2022-10-10 22:50:10 <UNKNOWN> [UDP: [192.168.
↪100.3]:6672->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (731774) 2:01:57.74
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application start command completed voss-queue"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:53:14 robot-slave snmptrapd[480]: 2022-10-10 22:53:14 <UNKNOWN> [UDP: [192.168.
↪100.3]:55096->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (750263) 2:05:02.63
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Excessive load"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.10.1.100.2
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.10.1.2.2 = STRING: "Load-5"
#011iso.3.6.1.4.1.2021.10.1.101.2 = STRING: "5 min Load Average too high (= 2.06)"
```

```
Oct 10 22:54:14 robot-slave snmptrapd[480]: 2022-10-10 22:54:14 <UNKNOWN> [UDP: [192.168.
↪100.3]:19700->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (756261) 2:06:02.61
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Excessive load"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.10.1.100.1
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.10.1.2.1 = STRING: "Load-1"
#011iso.3.6.1.4.1.2021.10.1.101.1 = STRING: "1 min Load Average too high (= 5.15)"
```

```
Oct 10 22:58:45 robot-slave snmptrapd[480]: 2022-10-10 22:58:42 <UNKNOWN> [UDP: [192.168.
```

(continues on next page)

(continued from previous page)

```
↪100.3]:39688->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (782988) 2:10:29.88
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "Certificate_Maintenance"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "ID: WEB_CERTIFICATE_VOSS, Code: -1, Occurences:␣
↪1, Latest Occurence: 2022-10-10T20:58:41.312Z"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:58:45 robot-slave snmptrapd[480]: 2022-10-10 22:58:45 <UNKNOWN> [UDP: [192.168.
↪100.3]:39098->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (783366) 2:10:33.66
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Web certificate about to expire"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Cert will expire in less than 30 day(s)"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 22:58:48 robot-slave snmptrapd[480]: 2022-10-10 22:58:48 <UNKNOWN> [UDP: [192.168.
↪100.3]:15074->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (783622) 2:10:36.22
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: Web certificate has been renewed"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Web certificate has been renewed"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 23:00:35 robot-slave snmptrapd[480]: 2022-10-10 23:00:35 <UNKNOWN> [UDP: [192.168.
↪100.3]:15622->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (794329) 2:12:23.29
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: Services started successfully"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "selfservice v22.1.0 (2022-10-10 13:57)
#012   |-node                  running
#012voss-deviceapi v22.1.0 (2022-10-10 13:57)
#012   |-voss-cnf_collector    running
#012   |-voss-queue            running
#012   |-voss-wsgi  Message is truncated"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 23:00:41 robot-slave snmptrapd[480]: 2022-10-10 23:00:41 <UNKNOWN> [UDP: [192.168.
↪100.3]:52074->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (794896) 2:12:28.96
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: License file generated"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "The license audit report scheduled for October␣
↪2022 was successful."
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
```

(continues on next page)

(continued from previous page)

```
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 23:48:15 robot-slave snmptrapd[480]: 2022-10-10 23:48:15 <UNKNOWN> [UDP: [192.168.
↪100.3]:39067->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1080266) 3:00:02.66
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Excessive load"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.10.1.100.1
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.10.1.2.1 = STRING: "Load-1"
#011iso.3.6.1.4.1.2021.10.1.101.1 = STRING: "1 min Load Average too high (= 5.83)"
```

```
Oct 10 23:48:15 robot-slave snmptrapd[480]: 2022-10-10 23:48:15 <UNKNOWN> [UDP: [192.168.
↪100.3]:61814->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1080302) 3:00:03.02
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Excessive load"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.10.1.100.2
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.10.1.2.2 = STRING: "Load-5"
#011iso.3.6.1.4.1.2021.10.1.101.2 = STRING: "5 min Load Average too high (= 2.66)"
```

```
Oct 10 23:50:04 robot-slave snmptrapd[480]: 2022-10-10 23:50:04 <UNKNOWN> [UDP: [192.168.
↪100.3]:15565->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1091211) 3:01:52.11
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "containerd-shim : num_files exceeded maximum␣
↪value of 19 with 29"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Command containerd-shim attribute num_files
#012Current maximum value 29
#012Error threshold 20
#012Warning threshold 13
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 23:50:05 robot-slave snmptrapd[480]: 2022-10-10 23:50:05 <UNKNOWN> [UDP: [192.168.
↪100.3]:2699->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1091258) 3:01:52.58
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "checkmongoalive : num_files exeeded warning␣
↪threshold of 13 with 20"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Command checkmongoalive attribute num_files
```

(continues on next page)

(continued from previous page)

```
#012Current average value 20
#012Error threshold 20
#012Warning threshold 13
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 10 23:50:15 robot-slave snmptrapd[480]: 2022-10-10 23:50:15 <UNKNOWN> [UDP: [192.168.
↪100.3]:48240->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1092279) 3:02:02.79
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Excessive load"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.10.1.100.1
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.10.1.2.1 = STRING: "Load-1"
#011iso.3.6.1.4.1.2021.10.1.101.1 = STRING: "1 min Load Average too high (= 7.25)"
```

```
Oct 11 00:00:14 robot-slave snmptrapd[480]: 2022-10-11 00:00:14 <UNKNOWN> [UDP: [192.168.
↪100.3]:57043->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1152194) 3:12:01.94
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "docker-proxy : num_sockets exceeded maximum␣
↪value of 29 with 43"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Command docker-proxy attribute num_sockets
#012Current maximum value 43
#012Error threshold 20
#012Warning threshold 13
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:00:15 robot-slave snmptrapd[480]: 2022-10-11 00:00:15 <UNKNOWN> [UDP: [192.168.
↪100.3]:40239->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1152282) 3:12:02.82
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "docker-proxy : num_files exceeded maximum value␣
↪of 61 with 89"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Command docker-proxy attribute num_files
#012Current maximum value 89
#012Error threshold 20
#012Warning threshold 13
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:00:15 robot-slave snmptrapd[480]: 2022-10-11 00:00:15 <UNKNOWN> [UDP: [192.168.
```

(continues on next page)

(continued from previous page)

```
↪100.3]:55586->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1152330) 3:12:03.30
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "dockerd : num_files exceeded maximum value of 78␣
↪with 84"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Command dockerd attribute num_files
#012Current maximum value 84
#012Error threshold 80
#012Warning threshold 53
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:00:16 robot-slave snmptrapd[480]: 2022-10-11 00:00:16 <UNKNOWN> [UDP: [192.168.
↪100.3]:62561->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1152364) 3:12:03.64
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed nginx:proxy"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:00:27 robot-slave snmptrapd[480]: 2022-10-11 00:00:27 <UNKNOWN> [UDP: [192.168.
↪100.3]:38716->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1153470) 3:12:14.70
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Excessive load"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.10.1.100.1
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.10.1.2.1 = STRING: "Load-1"
#011iso.3.6.1.4.1.2021.10.1.101.1 = STRING: "1 min Load Average too high (= 4.54)"
```

```
Oct 11 00:00:34 robot-slave snmptrapd[480]: 2022-10-11 00:00:34 <UNKNOWN> [UDP: [192.168.
↪100.3]:48544->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1154165) 3:12:21.65
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessError"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application failed with error 1 nginx:proxy"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:01:27 robot-slave snmptrapd[480]: 2022-10-11 00:01:27 <UNKNOWN> [UDP: [192.168.
↪100.3]:22430->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1159464) 3:13:14.64
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Disk full"
```

(continues on next page)

(continued from previous page)

```
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.9.1.100.1
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.9.1.2.1 = STRING: "/"
#011iso.3.6.1.4.1.2021.9.1.101.1 = STRING: "/: less than 30% free (= 0%)"
```

```
Oct 11 00:01:42 robot-slave snmptrapd[480]: 2022-10-11 00:01:42 <UNKNOWN> [UDP: [192.168.
↪100.3]:47625->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1160957) 3:13:29.57
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Log files larger than 1Gig found in /var/
↪log "
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Logrotation was executed to rotate the following␣
↪logs:
#012/var/log/big-log.log: 1.5G"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:01:43 robot-slave snmptrapd[480]: 2022-10-11 00:01:43 <UNKNOWN> [UDP: [192.168.
↪100.3]:6397->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1161096) 3:13:30.96
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: /var/log rotated"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "/var/log rotated"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:02:28 robot-slave snmptrapd[480]: 2022-10-11 00:02:28 <UNKNOWN> [UDP: [192.168.
↪100.3]:25418->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1165577) 3:14:15.77
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Disk full"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.9.1.100.3
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.9.1.2.3 = STRING: "/var/log"
#011iso.3.6.1.4.1.2021.9.1.101.3 = STRING: "/var/log: less than 10% free (= 0%)"
```

```
Oct 11 00:02:45 robot-slave snmptrapd[480]: 2022-10-11 00:02:45 <UNKNOWN> [UDP: [192.168.
↪100.3]:45283->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1167292) 3:14:32.92
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "Logs"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "WARNING: Forcing log rotation as /var/log is at␣
↪100 usage. [platform_mon.py] reading config file /etc/logrotate.conf
```

(continues on next page)

(continued from previous page)

```
#012including /etc/logrotate.d
#012reading config file alternatives
#012reading config file apt
#012reading   Message is truncated"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:02:45 robot-slave snmptrapd[480]: 2022-10-11 00:02:45 <UNKNOWN> [UDP: [192.168.
↪100.3]:43304->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1167346) 3:14:33.46
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "Logs"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "WARNING: Autopurging /var/log due to extremely␣
↪high disk usage. [platform_mon.py] Usage 100: Deleting:
#0124.0K
#011/var/log/alternatives.log.2.gz
#012104K
#011/var/log/apt/term.log.3.gz
#0128.0K
#011/var/log/apt/history.log.1.gz
#01212K  Message is truncated"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:02:46 robot-slave snmptrapd[480]: 2022-10-11 00:02:46 <UNKNOWN> [UDP: [192.168.
↪100.3]:63595->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1167361) 3:14:33.61
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "DISK ALMOST FULL: Disk /var/log is more than 80␣
↪percent full"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Use log purge to purge all rotated logs
#012
#012Current disk status:
#012Filesystem: /dev/sdb1
#012      Size: 9.9G
#012      Used: 9.5G
#012     Avail: 0
#012      Use%: 100%
#012Mounted on: /var/log"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:03:03 robot-slave snmptrapd[480]: 2022-10-11 00:03:03 <UNKNOWN> [UDP: [192.168.
↪100.3]:41278->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1169102) 3:14:51.02
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: Services started successfully"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "0"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:03:46 robot-slave snmptrapd[480]: 2022-10-11 00:03:46 <UNKNOWN> [UDP: [192.168.
↪100.3]:4569->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1173418) 3:15:34.18
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "DISK STATUS:  Disk /var/log is now running below␣
↪80 percent"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Disk /var/log cleared
#012Disk status after it was cleared:
#012Filesystem: /dev/sdb1
#012      Size: 9.9G
#012      Used: 31M
#012     Avail: 9.4G
#012      Use%: 1%
#012Mounted on: /var/log"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:05:02 robot-slave snmptrapd[480]: 2022-10-11 00:05:02 <UNKNOWN> [UDP: [192.168.
↪100.3]:40305->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1180986) 3:16:49.86
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: High memory usage"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Memory activity:
#012
#012223Mb"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:05:16 robot-slave snmptrapd[480]: 2022-10-11 00:05:16 <UNKNOWN> [UDP: [192.168.
↪100.3]:35079->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1182434) 3:17:04.34
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: Memory usage returned to normal"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Memory more than 1024MB"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:05:19 robot-slave snmptrapd[480]: 2022-10-11 00:05:19 <UNKNOWN> [UDP: [192.168.
↪100.3]:36811->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1182726) 3:17:07.26
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Disk slow "
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Disk latency info:
#012
#012Disk latency:34.44ms
#012"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:06:30 robot-slave snmptrapd[480]: 2022-10-11 00:06:30 <UNKNOWN> [UDP: [192.168.
```

(continues on next page)

(continued from previous page)

```
↪100.3]:40621->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1189781) 3:18:17.81
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: The disk latency returned to normal"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "The disk latency returned to normal."
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:12:33 robot-slave snmptrapd[480]: 2022-10-11 00:12:33 <UNKNOWN> [UDP: [192.168.
↪100.3]:29021->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1226146) 3:24:21.46
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessRestart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Applications are restarting snmp"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:12:37 robot-slave snmptrapd[480]: 2022-10-11 00:12:37 <UNKNOWN> [UDP: [192.168.
↪100.3]:39992->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1226526) 3:24:25.26
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.4.1.8072.4.0.2
#011iso.3.6.1.6.3.1.1.4.3.0 = OID: iso.3.6.1.4.1.8072.4
```

```
Oct 11 00:12:38 robot-slave snmptrapd[480]: 2022-10-11 00:12:38 <UNKNOWN> [UDP: [192.168.
↪100.3]:10991->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1226585) 3:24:25.85
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed snmp:daemon"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:12:38 robot-slave snmptrapd[480]: 2022-10-11 00:12:38 <UNKNOWN> [UDP: [192.168.
↪100.3]:21681->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1226610) 3:24:26.10
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed snmp:traps"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:12:39 robot-slave snmptrapd[480]: 2022-10-11 00:12:39 <UNKNOWN> [UDP: [192.168.
↪100.3]:7756->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1226681) 3:24:26.81
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application has changed its state from stopped␣
↪(killed) to unknown (missing launcher,killed) - snmp:traps"
```

(continues on next page)

(continued from previous page)

```
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:13:48 robot-slave snmptrapd[480]: 2022-10-11 00:13:48 <UNKNOWN> [UDP: [192.168.
↪100.3]:10511->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1233585) 3:25:35.85
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Excessive load"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.10.1.100.1
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.10.1.2.1 = STRING: "Load-1"
#011iso.3.6.1.4.1.2021.10.1.101.1 = STRING: "1 min Load Average too high (= 5.59)"
```

```
Oct 11 00:13:48 robot-slave snmptrapd[480]: 2022-10-11 00:13:48 <UNKNOWN> [UDP: [192.168.
↪100.3]:16438->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1233630) 3:25:36.30
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Excessive load"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.10.1.100.2
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.10.1.2.2 = STRING: "Load-5"
#011iso.3.6.1.4.1.2021.10.1.101.2 = STRING: "5 min Load Average too high (= 2.43)"
```

```
Oct 11 00:15:48 robot-slave snmptrapd[480]: 2022-10-11 00:15:48 <UNKNOWN> [UDP: [192.168.
↪100.3]:43788->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1245577) 3:27:35.77
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Excessive load"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.10.1.100.3
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.10.1.2.3 = STRING: "Load-15"
#011iso.3.6.1.4.1.2021.10.1.101.3 = STRING: "15 min Load Average too high (= 3.14)"
```

```
Oct 11 00:16:13 robot-slave snmptrapd[480]: 2022-10-11 00:16:13 <UNKNOWN> [UDP: [192.168.
↪100.3]:40452->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1248110) 3:28:01.10
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.6.3.1.1.5.1
#011iso.3.6.1.6.3.1.1.4.3.0 = OID: iso.3.6.1.4.1.8072.3.2.10
```

```
Oct 11 00:18:13 robot-slave snmptrapd[480]: 2022-10-11 00:18:13 <UNKNOWN> [UDP: [192.168.
↪100.3]:34120->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1260111) 3:30:01.11
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Memory swap error"
#011iso.3.6.1.2.1.88.2.1.2.0 = ""
#011iso.3.6.1.2.1.88.2.1.3.0 = ""
#011iso.3.6.1.2.1.88.2.1.4.0 = OID: iso.3.6.1.4.1.2021.4.100.0
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
#011iso.3.6.1.4.1.2021.4.2.0 = STRING: "swap"
#011iso.3.6.1.4.1.2021.4.101.0 = STRING: "Running out of swap space (0)"
```

```
Oct 11 00:20:20 robot-slave snmptrapd[480]: 2022-10-11 00:20:20 <UNKNOWN> [UDP: [192.168.
↪100.3]:22009->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1272824) 3:32:08.24
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessRestart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Applications are restarting voss-portal"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:20:28 robot-slave snmptrapd[480]: 2022-10-11 00:20:28 <UNKNOWN> [UDP: [192.168.
↪100.3]:44656->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1273568) 3:32:15.68
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStop"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Applications are stopping voss-portal"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:20:29 robot-slave snmptrapd[480]: 2022-10-11 00:20:29 <UNKNOWN> [UDP: [192.168.
↪100.3]:62367->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1273684) 3:32:16.84
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed voss-portal:gui"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:20:38 robot-slave snmptrapd[480]: 2022-10-11 00:20:38 <UNKNOWN> [UDP: [192.168.
↪100.3]:3228->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1274591) 3:32:25.91
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessError"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application failed with error 200 voss-portal:gui
↪"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:20:39 robot-slave snmptrapd[480]: 2022-10-11 00:20:39 <UNKNOWN> [UDP: [192.168.
↪100.3]:6835->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1274668) 3:32:26.68
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application start command completed voss-portal"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:21:28 robot-slave snmptrapd[480]: 2022-10-11 00:21:28 <UNKNOWN> [UDP: [192.168.
↪100.3]:9139->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1279583) 3:33:15.83
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "Database Maintenance"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "ID: TRANSACTION_DATABASE_MAINTENANCE-VOSS, Code:␣
↪-1, Occurences: 1, Latest Occurence: 2022-10-10T22:21:27.259Z"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:21:29 robot-slave snmptrapd[480]: 2022-10-11 00:21:29 <UNKNOWN> [UDP: [192.168.
↪100.3]:17486->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1279678) 3:33:16.78
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ERROR: Database maintenance not scheduled"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "TRANSACTION DATABASE MAINTENANCE NOT SCHEDULED -␣
↪SETUP SCHEDULE FOR REGULAR MAINTENANCE"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:21:38 robot-slave snmptrapd[480]: 2022-10-11 00:21:38 <UNKNOWN> [UDP: [192.168.
↪100.3]:30600->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1280638) 3:33:26.38
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "INFO: Database maintenance is scheduled"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Database maintenance has been scheduled"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:21:51 robot-slave snmptrapd[480]: 2022-10-11 00:21:51 <UNKNOWN> [UDP: [192.168.
↪100.3]:47088->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1281884) 3:33:38.84
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "Device Change Notification Collector data/
↪CallManager"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "ID: [10.120.9.245,8443,hcs.CS-P.CS-NB.Overton],␣
↪Code: 40006, Occurences: 16, Latest Occurence: 2019-10-07T09:42:59.899Z"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:21:52 robot-slave snmptrapd[480]: 2022-10-11 00:21:52 <UNKNOWN> [UDP: [192.168.
↪100.3]:53437->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1282006) 3:33:40.06
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "Transaction Completed with Fail"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "ID: 33289, Action: Create Cucm User, Detail:␣
↪someone-4, Hierarchy: sys.hcs.Verizon.AReseller_1.ACustomer_1"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:21:53 robot-slave snmptrapd[480]: 2022-10-11 00:21:53 <UNKNOWN> [UDP: [192.168.
↪100.3]:10089->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1282122) 3:33:41.22
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "Transaction Queue Size Exceeded Threshold"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Current Size: 3, Threshold: 2"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 3
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:21:54 robot-slave snmptrapd[480]: 2022-10-11 00:21:54 <UNKNOWN> [UDP: [192.168.
↪100.3]:60956->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1282238) 3:33:42.38
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "DB Index Size Exceeded Threshold"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "DB Index Size (0.00GB) exceeded threshold (1GB)"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 0
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:21:56 robot-slave snmptrapd[480]: 2022-10-11 00:21:56 <UNKNOWN> [UDP: [192.168.
↪100.3]:5918->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1282356) 3:33:43.56
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "DB Size Exceeded Threshold"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "DB Size (1.00GB) exceeded threshold (1GB)"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:21:57 robot-slave snmptrapd[480]: 2022-10-11 00:21:57 <UNKNOWN> [UDP: [192.168.
↪100.3]:58971->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1282473) 3:33:44.73
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "Global Administration Session Limit Exceeded"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Limit: 1"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:21:58 robot-slave snmptrapd[480]: 2022-10-11 00:21:58 <UNKNOWN> [UDP: [192.168.
↪100.3]:20414->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1282595) 3:33:45.95
```

(continues on next page)

(continued from previous page)

```
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "Administration Api Request Limit Exceeded"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Rate: 20/min"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:21:59 robot-slave snmptrapd[480]: 2022-10-11 00:21:59 <UNKNOWN> [UDP: [192.168.
↪100.3]:18233->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1282716) 3:33:47.16
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "User Api Request Limit Exceeded"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Rate: 20/sec"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:22:01 robot-slave snmptrapd[480]: 2022-10-11 00:22:01 <UNKNOWN> [UDP: [192.168.
↪100.3]:56285->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1282854) 3:33:48.54
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessRestart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Applications are restarting "
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:22:06 robot-slave snmptrapd[480]: 2022-10-11 00:22:06 <UNKNOWN> [UDP: [192.168.
↪100.3]:22442->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1283425) 3:33:54.25
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed selenium:gui_orchestration
↪"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:22:12 robot-slave snmptrapd[480]: 2022-10-11 00:22:12 <UNKNOWN> [UDP: [192.168.
↪100.3]:12754->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1283985) 3:33:59.85
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed selfservice:node"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:22:19 robot-slave snmptrapd[480]: 2022-10-11 00:22:19 <UNKNOWN> [UDP: [192.168.
↪100.3]:53711->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1284725) 3:34:07.25
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
```

(continues on next page)

(continued from previous page)

```
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed services:logs"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:22:20 robot-slave snmptrapd[480]: 2022-10-11 00:22:20 <UNKNOWN> [UDP: [192.168.
↪100.3]:19232->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1284769) 3:34:07.69
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed voss-deviceapi:voss-
↪risapi_collector"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:22:20 robot-slave snmptrapd[480]: 2022-10-11 00:22:20 <UNKNOWN> [UDP: [192.168.
↪100.3]:61419->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1284805) 3:34:08.05
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed voss-deviceapi:voss-wsgi"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:22:20 robot-slave snmptrapd[480]: 2022-10-11 00:22:20 <UNKNOWN> [UDP: [192.168.
↪100.3]:56372->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1284812) 3:34:08.12
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed voss-deviceapi:voss-queue"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:22:20 robot-slave snmptrapd[480]: 2022-10-11 00:22:20 <UNKNOWN> [UDP: [192.168.
↪100.3]:59666->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1284830) 3:34:08.30
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed voss-deviceapi:voss-
↪monitoring"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:22:20 robot-slave snmptrapd[480]: 2022-10-11 00:22:20 <UNKNOWN> [UDP: [192.168.
↪100.3]:35828->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1284829) 3:34:08.29
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed voss-deviceapi:voss-cnf_
```

(continues on next page)

(continued from previous page)

```
↪collector"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:22:21 robot-slave snmptrapd[480]: 2022-10-11 00:22:21 <UNKNOWN> [UDP: [192.168.
↪100.3]:11890->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1284895) 3:34:08.95
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed services:time"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:22:23 robot-slave snmptrapd[480]: 2022-10-11 00:22:23 <UNKNOWN> [UDP: [192.168.
↪100.3]:24541->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1285134) 3:34:11.34
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application has changed its state from stopped␣
↪(killed) to unknown (missing launcher,missing pid,killed) - services:time"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:22:28 robot-slave snmptrapd[480]: 2022-10-11 00:22:28 <UNKNOWN> [UDP: [192.168.
↪100.3]:4536->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1285611) 3:34:16.11
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed phone-based-
↪registration:nodeservice"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:22:43 robot-slave snmptrapd[480]: 2022-10-11 00:22:43 <UNKNOWN> [UDP: [192.168.
↪100.3]:24351->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1287104) 3:34:31.04
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed services:syslog"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:22:43 robot-slave snmptrapd[480]: 2022-10-11 00:22:43 <UNKNOWN> [UDP: [192.168.
↪100.3]:6262->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1287112) 3:34:31.12
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed services:mount"
```

(continues on next page)

(continued from previous page)

```
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:22:43 robot-slave snmptrapd[480]: 2022-10-11 00:22:43 <UNKNOWN> [UDP: [192.168.
↪100.3]:52368->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1287125) 3:34:31.25
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application was killed services:scheduler"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:22:45 robot-slave snmptrapd[480]: 2022-10-11 00:22:45 <UNKNOWN> [UDP: [192.168.
↪100.3]:20331->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1287320) 3:34:33.20
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application has changed its state from stopped␣
↪(killed) to unknown (missing launcher,killed) - services:mount"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:22:45 robot-slave snmptrapd[480]: 2022-10-11 00:22:45 <UNKNOWN> [UDP: [192.168.
↪100.3]:50365->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1287322) 3:34:33.22
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application has changed its state from stopped␣
↪(killed) to unknown (missing launcher,killed) - services:syslog"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:23:02 robot-slave snmptrapd[480]: 2022-10-11 00:23:02 <UNKNOWN> [UDP: [192.168.
↪100.3]:52355->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1288998) 3:34:49.98
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application has changed its state from stopped␣
↪(secure tunnel died) to running (secure tunnel died) - services:wsgi"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:23:13 robot-slave snmptrapd[480]: 2022-10-11 00:23:13 <UNKNOWN> [UDP: [192.168.
↪100.3]:26791->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1290061) 3:35:00.61
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application has changed its state from starting␣
```

(continues on next page)

(continued from previous page)

```
↪to running - snmp:daemon"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:23:13 robot-slave snmptrapd[480]: 2022-10-11 00:23:13 <UNKNOWN> [UDP: [192.168.
↪100.3]:47805->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1290070) 3:35:00.70
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application has changed its state from starting␣
↪to running - voss-deviceapi:voss-risapi_collector"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:23:14 robot-slave snmptrapd[480]: 2022-10-11 00:23:14 <UNKNOWN> [UDP: [192.168.
↪100.3]:60807->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1290203) 3:35:02.03
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application has changed its state from starting␣
↪to unknown (missing launcher) - snmp:traps"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:23:14 robot-slave snmptrapd[480]: 2022-10-11 00:23:14 <UNKNOWN> [UDP: [192.168.
↪100.3]:25702->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1290221) 3:35:02.21
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application has changed its state from starting␣
↪to unknown (missing launcher) - services:logs"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:23:14 robot-slave snmptrapd[480]: 2022-10-11 00:23:14 <UNKNOWN> [UDP: [192.168.
↪100.3]:60314->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1290239) 3:35:02.39
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application has changed its state from starting␣
↪to running - voss-deviceapi:voss-queue"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:23:16 robot-slave snmptrapd[480]: 2022-10-11 00:23:16 <UNKNOWN> [UDP: [192.168.
↪100.3]:33659->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1290445) 3:35:04.45
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
```

(continues on next page)

(continued from previous page)

```
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application has changed its state from starting␣
↪to unknown (missing launcher,missing pid) - services:time"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:23:20 robot-slave snmptrapd[480]: 2022-10-11 00:23:20 <UNKNOWN> [UDP: [192.168.
↪100.3]:42353->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1290822) 3:35:08.22
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.6.3.1.1.5.1
#011iso.3.6.1.6.3.1.1.4.3.0 = OID: iso.3.6.1.4.1.8072.3.2.10
```

```
Oct 11 00:23:33 robot-slave snmptrapd[480]: 2022-10-11 00:23:33 <UNKNOWN> [UDP: [192.168.
↪100.3]:29625->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1292068) 3:35:20.68
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStart"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application start command completed "
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:23:57 robot-slave snmptrapd[480]: 2022-10-11 00:23:57 <UNKNOWN> [UDP: [192.168.
↪100.3]:47870->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1294518) 3:35:45.18
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessStop"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Applications are stopping snmp"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:24:19 robot-slave snmptrapd[480]: 2022-10-11 00:24:19 <UNKNOWN> [UDP: [192.168.
↪100.3]:4081->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1296686) 3:36:06.86
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.2.1.88.2.0.1
#011iso.3.6.1.2.1.88.2.1.1.0 = STRING: "ProcessWarning"
#011iso.3.6.1.2.1.88.2.1.3.0 = STRING: "Application has changed its state from stopped␣
↪(killed) to unknown (missing launcher,killed) - snmp:daemon"
#011iso.3.6.1.2.1.88.2.1.5.0 = INTEGER: 1
#011iso.3.6.1.2.1.1.5.0 = STRING: "VOSS"
```

```
Oct 11 00:24:27 robot-slave snmptrapd[480]: 2022-10-11 00:24:27 <UNKNOWN> [UDP: [192.168.
↪100.3]:31239->[192.168.100.25]:162]:
#012iso.3.6.1.2.1.1.3.0 = Timeticks: (1297489) 3:36:14.89
#011iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.6.3.1.1.5.1
#011iso.3.6.1.6.3.1.1.4.3.0 = OID: iso.3.6.1.4.1.8072.3.2.10
```

.

Index

A

app
    app cleanup, 76
    app install, 56, 76
    app install nrs, 188
    app list, 55, 76
    app multi_install, 57
    app remove-role, 38
    app start, 54
    app status, 53, 54, 119
    app stop, 54
    app template, 7

B

backup
    backup add, 58, 173, 179
    backup clean, 178
    backup create, 175, 176, 179, 289
    backup default, 173
    backup import, 183
    backup list, 173, 183, 289
    backup passphrase, 174, 182, 183
    backup restore, 183

C

cluster, 6, 59
    cluster add, 37, 260
    cluster check, 85
    cluster del, 37, 45, 243, 245, 248, 250, 260, 269, 271, 273, 277
    cluster job, 59
    cluster list, 37, 250, 273
    cluster maintenance-mode, 81
    cluster prepnode, 3, 245, 248, 250, 258, 271, 282
    cluster primary, 42
    cluster primary role, 38
    cluster primary role application, 42
    cluster primary role database, 42
    cluster provision, 7, 37, 44, 243, 245, 248, 250, 254, 258, 260, 264, 269, 271, 273, 277, 279, 282, 285
    cluster run, 3, 51, 54, 59, 173, 243, 245, 248, 250, 254, 258, 264, 269, 271, 273, 277, 279, 282, 285
    cluster status, 37, 228, 260
    cluster upgrade, 7, 58, 76

D

database
    database config, 239, 243, 245, 248, 250, 254, 258, 260, 264, 266, 269, 273, 277, 279, 282, 285
    database primary, 250, 273
    database user, 200
    database weight, 228, 243, 245, 248, 250, 254, 258, 260, 264, 269, 273, 277, 279, 282, 285
diag, 361
    diag config app snmp, 365
    diag config platform, 364
    diag disk, 76, 118, 119, 361
    diag fileguard, 118, 187
    diag free, 118, 119, 363
    diag health, 118, 119
    diag iostat, 364
    diag mem, 118
    diag monitor, 362
    diag monitor list, 362
    diag nicstat, 363
    diag ntp, 363
    diag ping, 118
    diag proc, 365
    diag resolve, 118
    diag stats, 366
    diag tasks, 367
    diag test_connection, 118, 367
    diag top, 118, 119, 364, 367
    diag unittests, 118
    diag vmstat, 368
drives
    drives add, 78, 175
    drives create_volume, 78
    drives list, 78, 178, 288
    drives offset, 78
    drives reassign, 78, 183, 288
    drives remove_logical, 38, 78
    drives remove_volume, 38, 78

H

health, 3, 123, 171, 217
help, 3

I

insights sync
    insights sync schedule time;insights sync schedule list;insights sync run;insights sync enable;insights sync disable, 62

K

keys
    keys add, 197
    keys create, 93, 197
    keys send, 93, 197

L

license
    license add file, 24
    license add token, 24
    license check, 24
    license list, 24
log
    log audit, 90, 102
    log audit;log audit ruleset, 99
    log cert, 116
    log collect, 93
    log event, 90, 102
    log follow, 92, 102, 231
    log list, 90, 93
    log purge, 90
    log send, 93
    log send output, 93
    log sendnewer, 93
    log ssl, 116
    log stream, 102, 114
    log view, 92, 102, 119

M

mail
    mail del, 117
    mail list, 117
    mail read, 117

N

network
    network -4 del, 45, 47
    network -6 del, 45
    network <interface-name>, 45
    network container range, 51
    network dns, 50
    network domain, 50
    network interfaces, 45
    network name, 45
    network ntp, 50
    network routes, 45
    network search, 50
    network static_host, 45
notify
    notify add, 123, 127
    notify emailfrom, 88, 93, 123
    notify emailrelay, 88, 93, 123
    notify test, 123

S

schedule
    schedule add, 88, 170, 176, 178
    schedule enable, 170, 176
    schedule time, 88, 170, 176
selfservice
    selfservice get_translation_template, 61
    selfservice import_translation, 61
snmp
    snmp contact, 127
    snmp list, 127
    snmp load, 123, 127
    snmp location, 127
    snmp name, 127
    snmp query, 127
system, 3
    system banner, 80
    system boot password, 71
    system checksum, 82
    system download, 76
    system history, 90
    system id, 179
    system inactivelock, 188
    system intrusion-detection, 201
    system password, 3, 70
    system reboot, 70, 72
    system shutdown, 70
    system ssh, 196
    system ssh algorithm, 198
    system ssh fail_limit, 194
    system ssh_session_limit, 70
    system tty sane, 82
    system ua, 72

T

tmux, 7, 62

U

user
    user add, 70, 97, 191
    user addkey, 192, 199
    user credential_policy, 196

user del, 97, 192
user grant, 97, 193
user inactivelock, 188
user lastlogon, 188
user list, 97, 188, 192, 193, 196
user lock, 188
user password, 174, 188, 194
user password expiry, 188
user password history, 188
user passwordinfo, 188
user revoke, 97, 193
user sftp add, 192
user sftp add_nopass, 192
user sftp password, 192
user unlock, 188

V

voss, 9
voss check-license, 24
voss clear_device_pending_changes, 22
voss cleardown, 10, 231
voss db_collection_cap, 12
voss db_collection_cap_check, 12
voss db_collection_stats, 10, 175
voss db_index_stats, 11
voss export, 34
voss finalize_transaction, 21, 243, 245, 248, 250, 254, 258, 264, 269, 271, 273, 277, 279, 282, 285
voss migrate_summary_attributes, 10, 22, 58
voss post-upgrade-migrations, 23
voss queues, 19, 271, 279, 282, 285
voss report api, 31
voss report transaction, 27
voss reset_device_concurrency, 22
voss session-limits, 17, 163
voss set_debug, 22, 291
voss throttle-rates, 14, 163
voss transaction archive, 13, 24
voss transaction count, 13, 24
voss transaction delete, 13, 24
voss transaction export, 13, 24
voss unlock_sysadmin_account, 23, 97
voss update_device_schemas, 23
voss upgrade_db, 231
voss worker, 20
voss workers, 19, 80, 248, 258, 271, 277, 282
voss export
voss export group, 7
voss export type, 7
voss subscriber_data_export, 7

W

web
web cert, 208, 214, 218
web external, 224
web hosts, 219
web service, 61, 231
web ssl, 221, 223
web weight, 243, 245, 248, 250, 269, 271, 273, 277
web external
web external add; web external del; web external list, 224
web ssl
web ssl cipher, 223