Hewlett Packard
Enterprise

# HPE Cray EX System Admin Toolkit (SAT) Guide (2.1.17) (S-8031)

Part Number: S-8031
Published: November 2021

# HPE Cray EX System Admin Toolkit (SAT) Guide

## Contents

# 1 Copyright and Version

© Copyright 2021 Hewlett Packard Enterprise Development LP. All third-party marks are the property of their respective owners.

SAT: 2.1.17-63; Wed Nov 17 2021

Doc git hash: 4742630cb1ad4c23697159fc736fc55acbef135e

# 2 Introduction to SAT

## 2.1 About System Admin Toolkit (SAT)

The System Admin Toolkit (SAT) is designed to assist administrators with common tasks, such as troubleshooting and querying information about the HPE Cray EX System and its components, system boot and shutdown, and replacing hardware components.

SAT offers a command line utility which uses subcommands. There are similarities between SAT commands and xt commands used on the Cray XC platform. For more information on SAT commands, see System Admin Toolkit Command Overview.

Six Kibana Dashboards are included with SAT. They provide organized output for system health information.

- AER Kibana Dashboard
- ATOM Kibana Dashboard
- Heartbeat Kibana Dashboard
- Kernel Kibana Dashboard
- MCE Kibana Dashboard
- Rasdaemon Kibana Dashboard

Four Grafana Dashboards are included with SAT. They display messages that are generated by the HSN (High Speed Network) and are reported through Redfish.

- Grafana Fabric Congestion Dashboard
- Grafana Fabric Errors Dashboard
- Grafana Fabric Port State Dashboard
- Grafana Fabric RFC3635 Dashboard

SAT is installed as a separate product as part of the HPE Cray EX System base installation.

## 2.2 System Admin Toolkit Command Overview

Describes the SAT Command Line Utility, lists the key commands found in the System Admin Toolkit man pages, and provides instruction on the SAT Container Environment.

### 2.2.1 SAT Command Line Utility

The primary component of the System Admin Toolkit (SAT) is a command-line utility run from Kubernetes manager nodes (`ncn-m` nodes).

It is designed to assist administrators with common tasks, such as troubleshooting and querying information about the HPE Cray EX System and its components, system boot and shutdown, and replacing hardware components. There are similarities between SAT commands and xt commands used on the Cray XC platform.

### 2.2.2 SAT Commands

The top-level SAT man page describes the toolkit, documents the global options affecting all subcommands, documents configuration file options, and references the man page for each subcommand. SAT consists of many subcommands that each have their own set of options.

### 2.2.3 SAT Container Environment

The `sat` command-line utility runs in a container using podman, a daemonless container runtime. SAT runs on Kubernetes manager nodes. A few important points about the SAT container environment include the following:

- Using either `sat` or `sat bash` always launches a container.
- The SAT container does not have access to the NCN file system.

There are two ways to run sat.

- **Interactive**: Launching a container using `sat bash`, followed by a `sat` command.
- **Non-interactive**: Running a `sat` command directly on a Kubernetes manager node.

In both of these cases, a container is launched in the background to execute the command. The first option, running `sat bash` first, gives an interactive shell, at which point `sat` commands can be run. In the second option, the container is launched, executes the command, and upon the command's completion the container exits. The following two examples show the same action, checking the system status, using interactive and non-interactive modes.

#### 2.2.3.1 Interactive

```
ncn-m001# sat bash
(CONTAINER-ID)sat-container# sat status
```

#### 2.2.3.2 Non-interactive

```
ncn-m001# sat status
```

#### 2.2.3.3 Interactive Advantages

Running `sat` using the interactive command prompt gives the ability to read and write local files on ephemeral container storage. If multiple `sat` commands are being run in succession, then use sat bash to launch the container beforehand. This will save time because the container does not need to be launched for each `sat` command.

#### 2.2.3.4 Non-interactive Advantages

The non-interactive mode is useful if calling `sat` with a script, or when running a single `sat` command as a part of several steps that need to be executed from a management NCN.

#### 2.2.3.5 Man Pages - Interactive and Non-interactive Modes

To view a `sat` man page from a Kubernetes manager node, use `sat-man` on the manager node as shown in the following example.

```
ncn-m001# sat-man status
```

A man page describing the SAT container environment is available on the Kubernetes manager nodes, which can be viewed either with `man sat` or man `sat-podman` from the manager node.

```
ncn-m001# man sat
```

```
ncn-m001# man sat-podman
```

## 2.3 Command Prompt Conventions in SAT

The host name in a command prompt indicates where the command must be run. The account that must run the command is also indicated in the prompt.

- The `root` or super-user account always has the # character at the end of the prompt and has the host name of the host in the prompt.
- Any non-`root` account is indicated with account@hostname>. A user account that is neither `root` nor `crayadm` is referred to as `user`.
- The command prompt inside the SAT container environment is indicated with the string as follows. It also has the "#" character at the end of the prompt.

| Command Prompt | Meaning |
|---|---|
| `ncn-m001#` | Run on one of the Kubernetes Manager servers. (**Non-interactive**) |
| `(CONTAINER_ID) sat-container#` | Run the command inside the SAT container environment by first running `sat bash`. (**Interactive**) |

Examples of the `sat status` command used by an administrator:

```
ncn-m001# sat status
```

```
ncn-m001# sat bash
(CONTAINER_ID) sat-container# sat status
```

## 2.4 SAT Dependencies

SAT has dependencies on other products for some of its functionality. The following table shows the dependencies that each `sat` subcommand has on other products in the HPE Cray EX (Shasta) software stack. It shows the products as well as the specific services or components of those products on which the given `sat` command depends.

| SAT Subcommand | Product Dependencies |
| --- | --- |
| `sat auth` | **CSM**<br>• Keycloak |
| `sat bootsys` | **CSM**<br>• Boot Orchestration Service (BOS)<br>• Cray Advanced Platform Monitoring and Control (CAPMC)<br>• Ceph<br>• Compute Rolling Upgrade Service (CRUS)<br>• Etcd<br>• Firmware Action Service (FAS)<br>• Hardware State Manager (HSM)<br>• Kubernetes<br>• S3<br>**COS**<br>• Node Memory Dump (NMD) |
| `sat diag` | **CSM-Diag**<br>• Fox |
| `sat firmware` | **CSM**<br>• Firmware Action Service (FAS) |
| `sat hwinv` | **CSM**<br>• Hardware State Manager (HSM) |
| `sat hwmatch` | **CSM**<br>• Hardware State Manager (HSM) |
| `sat init` | None |
| `sat k8s` | **CSM**<br>• Kubernetes |
| `sat nid2xname` | **CSM**<br>• Hardware State Manager (HSM) |
| `sat sensors` | **CSM**<br>• Hardware State Manager (HSM)<br>• HM Collector<br>**SMA**<br>• Telemetry API |
| `sat setrev` | **CSM**<br>• S3 |
| `sat showrev` | **CSM**<br>• Hardware State Manager (HSM)<br>• Kubernetes<br>• S3 |
| `sat status` | **CSM**<br>• Hardware State Manager (HSM) |
| `sat swap` | **Slingshot**<br>• Fabric Manager |
| `sat switch` | **Deprecated**: See `sat swap` |

| SAT Subcommand | Product Dependencies |
| --- | --- |
| `sat xname2nid` | **CSM**<br>• Hardware State Manager (HSM) |

# 3 SAT Installation and Upgrade

## 3.1 Install the System Admin Toolkit Product Stream

Describes how to install the System Admin Toolkit (SAT) product stream.

### 3.1.1 Prerequisites

- CSM is installed and verified.
- cray-product-catalog is running.
- There must be at least 2 gigabytes of free space on the manager NCN on which the procedure is run.

### 3.1.2 Notes on the Procedures

- Ellipses (`...`) in shell output indicate omitted lines.
- In the examples below, replace `2.1.x` with the version of the SAT product stream being installed.
- 'manager' and 'master' are used interchangeably in the steps below.
- To upgrade SAT, execute the pre-installation, installation, and post-installation procedures for a newer distribution. The newly installed version will become the default.

### 3.1.3 Pre-Installation Procedure

1. Start a typescript.

   The typescript will record the commands and the output from this installation.

   ```
   ncn-m001# script -af product-sat.$(date +%Y-%m-%d).txt
   ncn-m001# export PS1='\u@\H \D{%Y-%m-%d} \t \w # '
   ```

### 3.1.4 Installation Procedure

1. Copy the release distribution gzipped tar file to `ncn-m001`.
2. Unzip and extract the release distribution, `2.1.x`.

   ```
   ncn-m001# tar -xvzf sat-2.1.x.tar.gz
   ```

3. Change directory to the extracted release distribution directory.

   ```
   ncn-m001# cd sat-2.1.x
   ```

4. Run the installer: **install.sh**.

   The script produces a lot of output. The last several lines are included below for reference.

   ```
   ncn-m001# ./install.sh
   ...
   ConfigMap data updates exist; Exiting.
   + clean-install-deps
   + for image in "${vendor_images[@]}"
   + podman rmi -f docker.io/library/cray-nexus-setup:sat-2.1.x-20210804163905-8dbb87d
   Untagged: docker.io/library/cray-nexus-setup:sat-2.1.x-20210804163905-8dbb87d
   Deleted: 2c196c0c6364d9a1699d83dc98550880dc491cc3433a015d35f6cab1987dd6da
   + for image in "${vendor_images[@]}"
   + podman rmi -f docker.io/library/skopeo:sat-2.1.x-20210804163905-8dbb87d
   Untagged: docker.io/library/skopeo:sat-2.1.x-20210804163905-8dbb87d
   Deleted: 1b38b7600f146503e246e753cd9df801e18409a176b3dbb07b0564e6bc27144c
   ```

5. Check the return code of the installer. Zero indicates a successful installation.

   ```
   ncn-m001# echo $?
   0
   ```

6. Check the progress of the SAT configuration import Kubernetes job, which is initiated by `install.sh`.

   If the "Pods Statuses" appear as "Succeeded", the job has completed successfully. The job usually takes between 30 seconds and 2 minutes.

   ```
   ncn-m001# kubectl describe job sat-config-import-2.1.x -n services
   ...
   Pods Statuses:  0 Running / 1 Succeeded / 0 Failed
   ...
   ```

   The job's progress may be monitored using `kubectl logs`. The example below includes the final log lines from a successful configuration import Kubernetes job.

   ```
   ncn-m001# kubectl logs -f -n services --selector \
       job-name=sat-config-import-2.1.x --all-containers
   ...
   ConfigMap update attempt=1
   Resting 1s before reading ConfigMap
   ConfigMap data updates exist; Exiting.
   2021-08-04T21:50:10.275886Z  info    Agent has successfully terminated
   2021-08-04T21:50:10.276118Z  warning envoy main  caught SIGTERM
   # Completed on Wed Aug  4 21:49:44 2021
   ```

   The following error may appear in this log, but it can be ignored.

   ```
   error accept tcp [::]:15020: use of closed network connection
   ```

### 3.1.5 Post-Installation Procedure

1. **Optional:** Remove the SAT release distribution tar file and extracted directory.

   ```
   ncn-m001# rm sat-2.2.x.tar.gz
   ncn-m001# rm -rf sat-2.2.x/
   ```

2. **Upgrade only**: Ensure that the environment variable `SAT_TAG` is not set in the `~/.bashrc` file on any of the management NCNs.

   **NOTE**: This step should only be required when updating from Shasta 1.4.1 or Shasta 1.4.2.

   The following example assumes three manager NCNs: `ncn-m001`, `ncn-m002`, and `ncn-m003`, and shows output from a system in which no further action is needed.

   ```
   ncn-m001# pdsh -w ncn-m00[1-3] cat ~/.bashrc
   ncn-m001: source <(kubectl completion bash)
   ncn-m003: source <(kubectl completion bash)
   ncn-m002: source <(kubectl completion bash)
   ```

   The following example shows that `SAT_TAG` is set in `~/.bashrc` on `ncn-m002`. Remove that line from the `~/.bashrc` file on `ncn-m002`.

   ```
   ncn-m001# pdsh -w ncn-m00[1-3] cat ~/.bashrc
   ncn-m001: source <(kubectl completion bash)
   ncn-m002: source <(kubectl completion bash)
   ncn-m002: export SAT_TAG=3.5.0
   ncn-m003: source <(kubectl completion bash)
   ```

3. Stop the typescript.

   **NOTE**: This step can be skipped if you wish to use the same typescript for the remainder of the SAT install. See Next Steps.

   ```
   ncn-m001# exit
   ```

SAT version `2.1.x` is now installed/upgraded, meaning the SAT `2.1.x` release has been loaded into the system software repository.

- SAT configuration content for this release has been uploaded to VCS.
- SAT content for this release has been uploaded to the CSM product catalog.
- SAT content for this release has been uploaded to Nexus repositories.
- The `sat` command won't be available until the NCN Personalization procedure has been executed.

### 3.1.6 Next Steps

If other HPE Cray EX software products are being installed or upgraded in conjunction with SAT, refer to the *HPE Cray EX System Software Getting Started Guide* to determine which step to execute next.

If no other HPE Cray EX software products are being installed or upgraded at this time, proceed to the sections listed below.

**NOTE:** The **NCN Personalization** procedure **is required when upgrading SAT**. The setup procedures in **SAT Setup**, however, are **not required when upgrading SAT**. They should have been executed during the first installation of SAT.

Execute the **NCN Personalization** procedure:

- Perform NCN Personalization

If performing a fresh install, execute the **SAT Setup** procedures:

- SAT Authentication
- Generate SAT S3 Credentials
- Run Sat Setrev to Set System Information

If performing an upgrade, execute the **upgrade** procedures:

- SAT Post-Upgrade

## 3.2 Perform NCN Personalization

Describes how to perform NCN personalization using CFS. This personalization process will configure the System Admin Toolkit (SAT) product stream.

### 3.2.1 Prerequisites

- The Install the System Admin Toolkit Product Stream procedure has been successfully completed.

### 3.2.2 Notes on the Procedure

- Ellipses (`...`) in shell output indicate omitted lines.
- In the examples below, replace `2.1.x` with the version of the SAT product stream being installed.
- 'manager' and 'master' are used interchangeably in the steps below.
- If upgrading SAT, the existing configuration will likely include other Cray EX product entries. Update the SAT entry as described in this procedure. The *HPE Cray EX System Software Getting Started Guide* provides guidance on how and when to update the entries for the other products.

### 3.2.3 Procedure

1. Start a typescript if not already using one.

   The typescript will capture the commands and the output from this installation procedure.

   ```
   ncn-m001# script -af product-sat.$(date +%Y-%m-%d).txt
   ncn-m001# export PS1='\u@\H \D{%Y-%m-%d} \t \w # '
   ```

2. Get the git commit ID for the branch with a version number matching the version of SAT.

   This represents a revision of Ansible configuration content stored in VCS.

   Get and store the VCS password (required to access the remote VCS repo).

   ```
   ncn-m001# VCS_PASS=$(kubectl get secret -n services vcs-user-credentials \
       --template={{.data.vcs_password}} | base64 --decode)
   ```

   In this example, the git commit ID is `82537e59c24dd5607d5f5d6f92cdff971bd9c615`, and the version number is `2.1.x`.

```
ncn-m001# git ls-remote \
    https://crayvcs:$VCS_PASS@api-gw-service-nmn.local/vcs/cray/sat-config-management.git \
    refs/heads/cray/sat/*
...
82537e59c24dd5607d5f5d6f92cdff971bd9c615 refs/heads/cray/sat/2.1.x
```

3. Add a `sat` layer to the CFS configuration(s) associated with the manager NCNs.

    1. Get the name(s) of the CFS configuration(s).

        **NOTE:** Each manager NCN uses a single CFS configuration. An individual CFS configuration may be used by any number of manage NCNs, i.e., three manager NCNs might use one, two, or three CFS configurations.

        In the following example, all three manager NCNs use the same CFS configuration – `ncn-personalization`.

        ```
        ncn-m001:~ # for component in $(cray hsm state components list \
            --role Management --subrole Master --format json | jq -r \
            '.Components | .[].ID'); do cray cfs components describe $component \
            --format json | jq -r '.desiredConfig'; done
        ncn-personalization
        ncn-personalization
        ncn-personalization
        ```

        In the following example, the three manager NCNs all use different configurations, each with a unique name.

        ```
        ncn-personalization-m001
        ncn-personalization-m002
        ncn-personalization-m003
        ```

        Execute the following sub-steps (3.2 through 3.5) once for each unique CFS configuration name.

        **NOTE:** Examples in the following sub-steps assume that all manager NCNs use the CFS configuration `ncn-personalization`.

    2. Get the current configuration layers for each CFS configuration, and save the data to a local JSON file.

        The JSON file created in this sub-step will serve as a template for updating an existing CFS configuration, or creating a new one.

        ```
        ncn-m001# cray cfs configurations describe ncn-personalization --format \
            json | jq '{ layers }' > ncn-personalization.json
        ```

        If the configuration does not exist yet, you may see the following error. In this case, create a new JSON file for that CFS configuration, e.g., `ncn-personalization.json`.

        ```
        Error: Configuration could not found.: Configuration ncn-personalization could not be found
        ```

        **NOTE:** For more on CFS configuration management, refer to “Manage a Configuration with CFS” in the CSM product documentation.

    3. Append a `sat` layer to the end of the JSON file’s list of layers.

        If the file already contains a `sat` layer entry, update it.

        If the configuration data could not be found in the previous sub-step, the JSON file will be empty. In this case, copy the `ncn-personalization.json` example below, paste it into the JSON file, delete the ellipsis, and make appropriate changes to the `sat` layer entry.

        Use the git commit ID from step 8, e.g. `82537e59c24dd5607d5f5d6f92cdff971bd9c615`.

        **NOTE:** The `name` value in the example below may be changed, but the installation procedure uses the example value, `sat-ncn`. If an alternate value is used, some of the following examples must be updated accordingly before they are executed.

        ```
        ncn-m001# vim ncn-personalization.json
        ...
        ncn-m001# cat ncn-personalization.json
        {
            "layers": [
                ...
                {
        ```

```
            "cloneUrl": "https://api-gw-service-nmn.local/vcs/cray/sat-config-management.git",
            "commit": "82537e59c24dd5607d5f5d6f92cdff971bd9c615",
            "name": "sat-ncn",
            "playbook": "sat-ncn.yml"
        }
    ]
}
```

4. Update the existing CFS configuration, or create a new one.

   The command should output a JSON-formatted representation of the CFS configuration, which will look like the JSON file, but with `lastUpdated` and `name` fields.

   ```
   ncn-m001# cray cfs configurations update ncn-personalization --file \
       ncn-personalization.json --format json
   {
       "lastUpdated": "2021-08-05T16:38:53Z",
       "layers": {
           ...
       },
       "name": "ncn-personalization"
   }
   ```

5. **Optional:** Delete the JSON file.

   **NOTE:** There is no reason to keep the file. If you keep it, verify that it is up-to-date with the actual CFS configuration before using it again.

   ```
   ncn-m001# rm ncn-personalization.json
   ```

4. Invoke the CFS configurations that you created or updated in the previous step.

   This step will create a CFS session based on the given configuration and install SAT on the associated manager NCNs.

   The `--configuration-limit` option causes only the `sat-ncn` layer of the configuration, `ncn-personalization`, to run.

   **CAUTION:** In this example, the session `--name` is `sat-session`. That value is only an example. Declare a unique name for each configuration session.

   You should see a representation of the CFS session in the output.

   ```
   ncn-m001# cray cfs sessions create --name sat-session --configuration-name \
       ncn-personalization --configuration-limit sat-ncn
   name="sat-session"

   [ansible]
   ...
   ```

   Execute this step once for each unique CFS configuration that you created or updated in the previous step.

5. Monitor the progress of each CFS session.

   First, list all containers associated with the CFS session:

   ```
   ncn-m001# kubectl get pod -n services --selector=cfsession=sat-session \
       -o json | jq '.items[0].spec.containers[] | .name'
   "inventory"
   "ansible-1"
   "istio-proxy"
   ```

   Next, get the logs for the `ansible-1` container.

   **NOTE:** the trailing digit might differ from "1". It is the zero-based index of the `sat-ncn` layer within the configuration's layers.

   ```
   ncn-m001# kubectl logs -c ansible-1 --tail 100 -f -n services \
       --selector=cfsession=sat-session
   ```

Ansible plays, which are run by the CFS session, will install SAT on all the manager NCNs on the system. Successful results for all of the manager NCN xnames can be found at the end of the container log. For example:

```
...
PLAY RECAP *********************************************************************
x3000c0s1b0n0              : ok=3    changed=3    unreachable=0    failed=0    skipped=0    rescued=0
x3000c0s3b0n0              : ok=3    changed=3    unreachable=0    failed=0    skipped=0    rescued=0
x3000c0s5b0n0              : ok=3    changed=3    unreachable=0    failed=0    skipped=0    rescued=0
```

Execute this step for each unique CFS configuration.

**NOTE:** Ensure that the PLAY RECAPs for each session show successes for all manager NCNs before proceeding.

6. Verify that SAT was successfully configured.

   If `sat` is configured, the `--version` command will indicate which version is installed. If `sat` is not properly configured, the command will fail.

   **NOTE:** This version number will differ from the version number of the SAT release distribution. This is the semantic version of the `sat` Python package, which is different from the version number of the overall SAT release distribution.

   ```
   ncn-m001# sat --version
   sat 3.7.0
   ```

   **NOTE**: Upon first running `sat`, you may see additional output while the `sat` container image is downloaded. This will occur the first time `sat` is run on each manager NCN. For example, if you run `sat` for the first time on `ncn-m001` and then for the first time on `ncn-m002`, you will see this additional output both times.

   ```
   Trying to pull registry.local/cray/cray-sat:3.7.0-20210514024359_9fed037...
   Getting image source signatures
   Copying blob da64e8df3afc done
   Copying blob 0f36fd81d583 done
   Copying blob 12527cf455ba done
   ...
   sat 3.7.0
   ```

7. Stop the typescript.

   ```
   ncn-m001# exit
   ```

SAT version `2.1.x` is now configured:

- The SAT RPM package is installed on the associated NCNs.

### 3.2.4 Next Steps

If other HPE Cray EX software products are being installed or upgraded in conjunction with SAT, refer to the *HPE Cray EX System Software Getting Started Guide* to determine which step to execute next.

If no other HPE Cray EX software products are being installed or upgraded at this time, proceed to the remaining **SAT Setup** or **SAT Post-Upgrade** procedures.

If performing a fresh install, execute the **SAT Setup** procedures:

- SAT Authentication
- Generate SAT S3 Credentials
- Run Sat Setrev to Set System Information

If performing an upgrade, execute the **SAT Post-Upgrade** procedures:

- SAT Post-Upgrade

# 4 SAT Setup

## 4.1 SAT Authentication

Initially, as part of the installation and configuration, SAT authentication is set up so sat commands can be used in later steps of the install process. The admin account used to authenticate with `sat auth` must be enabled in Keycloak and must have its ***assigned role*** set to ***admin***. For instructions on editing ***Role Mappings*** see ***Create Internal User Accounts in the Keycloak Shasta Realm*** in the CSM product documentation. For additional information on SAT authentication, see ***System Security and Authentication*** in the CSM documentation.

**NOTE:** This procedure is only required after initially installing SAT. It is not required after upgrading SAT.

### 4.1.1 Description of SAT Command Authentication Types

Some SAT subcommands make requests to the Shasta services through the API gateway and thus require authentication to the API gateway in order to function. Other SAT subcommands use the Kubernetes API. Some `sat` commands require S3 to be configured (see: Generate SAT S3 Credentials). In order to use the SAT S3 bucket, the System Administrator must generate the S3 access key and secret keys and write them to a local file. This must be done on every Kubernetes manager node where SAT commands are run.

Below is a table describing SAT commands and the types of authentication they require.

| SAT Subcommand | Authentication/Credentials Required | Man Page | Description |
|---|---|---|---|
| `sat auth` | Responsible for authenticating to the API gateway and storing a token. | `sat-auth` | Authenticate to the API gateway and save the token. |
| `sat bootsys` | Requires authentication to the API gateway. Requires kubernetes configuration and authentication, which is configured on `ncn-m001` during the install. Some stages require passwordless SSH to be configured to all other NCNs. Requires S3 to be configured for some stages. | `sat-bootsys` | Boot or shutdown the system, including compute nodes, application nodes, and non-compute nodes (NCNs) running the management software. |
| `sat diag` | Requires authentication to the API gateway. | `sat-diag` | Launch diagnostics on the HSN switches and generate a report. |
| `sat firmware` | Requires authentication to the API gateway. | `sat-firmware` | Report firmware version. |
| `sat hwinv` | Requires authentication to the API gateway. | `sat-hwinv` | Give a listing of the hardware of the HPE Cray EX system. |
| `sat hwmatch` | Requires authentication to the API gateway. | `sat-hwmatch` | Report hardware mismatches. |
| `sat init` | None | `sat-init` | Create a default SAT configuration file. |
| `sat k8s` | Requires kubernetes configuration and authentication, which is automatically configured on ncn-w001 during the install. | `sat-k8s` | Report on kubernetes replicasets that have co-located replicas (i.e. replicas on the same node). |
| `sat linkhealth` | | | **This command has been deprecated.** |

| SAT Subcommand | Authentication/Credentials Required | Man Page | Description |
|---|---|---|---|
| `sat nid2xname` | Requires authentication to the API gateway. | `sat-nid2xname` | Translate node IDs to node xnames. |
| `sat sensors` | Requires authentication to the API gateway. | `sat-sensors` | Report current sensor data. |
| `sat setrev` | Requires S3 to be configured for site information such as system name, serial number, install date, and site name. | `sat-setrev` | Set HPE Cray EX system revision information. |
| `sat showrev` | Requires API gateway authentication in order to query the Interconnect from HSM. Requires S3 to be configured for site information such as system name, serial number, install date, and site name. | `sat-showrev` | Print revision information for the HPE Cray EX system. |
| `sat status` | Requires authentication to the API gateway. | `sat-status` | Report node status across the HPE Cray EX system. |
| `sat swap` | Requires authentication to the API gateway. | `sat-swap` | Prepare HSN switch or cable for replacement and bring HSN switch or cable into service. |
| `sat xname2nid` | Requires authentication to the API gateway. | `sat-xname2nid` | Translate node and node BMC xnames to node IDs. |
| `sat switch` | **This command has been deprecated.** It has been replaced by `sat swap`. | | |

In order to authenticate to the API gateway, you must run the `sat auth` command. This command will prompt for a password on the command line. The username value is obtained from the following locations, in order of higher precedence to lower precedence:

- The `--username` global command-line option.
- The `username` option in the `api_gateway` section of the config file at `~/.config/sat/sat.toml`.
- The name of currently logged in user running the `sat` command.

If credentials are entered correctly when prompted by `sat auth`, a token file will be obtained and saved to `~/.config/sat/tokens`. Subsequent sat commands will determine the username the same way as `sat auth` described above, and will use the token for that username if it has been obtained and saved by `sat auth`.

### 4.1.2 Prerequisites

- The `sat` CLI has been installed following Install The System Admin Toolkit Product Stream.

### 4.1.3 Procedure

The following is the procedure to globally configure the username used by SAT and authenticate to the API gateway:

1. Generate a default SAT configuration file, if one does not exist.

   ```
   ncn-m001# sat init
   Configuration file "/root/.config/sat/sat.toml" generated.
   ```

   **Note:** If the config file already exists, it will print out an error:

   ```
   ERROR: Configuration file "/root/.config/sat/sat.toml" already exists.
   Not generating configuration file.
   ```

2. Edit `~/.config/sat/sat.toml` and set the username option in the `api_gateway` section of the config file. E.g.:

   ```
   username = "crayadmin"
   ```

3. Run `sat auth`. Enter your password when prompted. E.g.:

```
ncn-m001# sat auth
Password for crayadmin:
Succeeded!
```

4. Other `sat` commands are now authenticated to make requests to the API gateway. E.g.:

   ```
   ncn-m001# sat status
   ```

## 4.2 Generate SAT S3 Credentials

Generate S3 credentials and write them to a local file so the SAT user can access S3 storage. In order to use the SAT S3 bucket, the System Administrator must generate the S3 access key and secret keys and write them to a local file. This must be done on every Kubernetes master node where SAT commands are run.

SAT uses S3 storage for several purposes, most importantly to store the site-specific information set with `sat setrev` (see: Run Sat Setrev to Set System Information).

**NOTE:** This procedure is only required after initially installing SAT. It is not required after upgrading SAT.

### 4.2.1 Prerequisites

- The `sat` CLI has been installed following Install The System Admin Toolkit Product Stream.
- The `sat` configuration file has been created (See SAT Authentication).
- CSM has been installed and verified.

### 4.2.2 Procedure

1. Ensure the files are readable only by `root`.

   ```
   ncn-m001# touch /root/.config/sat/s3_access_key \
       /root/.config/sat/s3_secret_key
   ```

   ```
   ncn-m001# chmod 600 /root/.config/sat/s3_access_key \
       /root/.config/sat/s3_secret_key
   ```

2. Write the credentials to local files using `kubectl`.

   ```
   ncn-m001# kubectl get secret sat-s3-credentials -o json -o \
       jsonpath='{.data.access_key}' | base64 -d > \
       /root/.config/sat/s3_access_key
   ```

   ```
   ncn-m001# kubectl get secret sat-s3-credentials -o json -o \
       jsonpath='{.data.secret_key}' | base64 -d > \
       /root/.config/sat/s3_secret_key
   ```

3. Verify the S3 endpoint specified in the SAT configuration file is correct.

   1. Get the SAT configuration file's endpoint valie.

      **NOTE:** If the command's output is commented out, indicated by an initial # character, the SAT configuration will take the default value – `"https://rgw-vip.nmn"`.

      ```
      ncn-m001# grep endpoint ~/.config/sat/sat.toml
      # endpoint = "https://rgw-vip.nmn"
      ```

   2. Get the `sat-s3-credentials` secret's endpoint value.

      ```
      ncn-m001# kubectl get secret sat-s3-credentials -o json -o \
          jsonpath='{.data.s3_endpoint}' | base64 -d | xargs
      https://rgw-vip.nmn
      ```

   3. Compare the two endpoint values.

      If the values differ, modify the SAT configuration file's endpoint value to match the secret's.

4. Copy SAT configurations to every manager node on the system.

```
ncn-m001# for i in ncn-m002 ncn-m003; do echo $i; ssh ${i} \
    mkdir -p /root/.config/sat; \
    scp -pr /root/.config/sat ${i}:/root/.config; done
```

**NOTE**: Depending on how many manager nodes are on the system, the list of manager nodes may be different. This example assumes three manager nodes, where the configuration files must be copied from ncn-m001 to ncn-m002 and ncn-m003. Therefore, the list of hosts above is ncn-m002 and ncn-m003.

## 4.3 Run sat setrev to Set System Information

**NOTE:** This procedure is only required after initially installing SAT. It is not required after upgrading SAT.

### 4.3.1 Prerequisites

- S3 credentials have been generated. See Generate SAT S3 Credentials.
- SAT authentication has been set up. See SAT Authentication.

### 4.3.2 Procedure

1. Run `sat setrev` to set System Revision Information. Follow the on-screen prompts.

```
ncn-m001# sat setrev
--------------------------------------------------------------------------------
Setting:        Serial number
Purpose:        System identification. This will affect how snapshots are
                identified in the HPE backend services.
Description:    This is the top-level serial number which uniquely identifies
                the system. It can be requested from an HPE representative.
Valid values:   Alpha-numeric string, 4 - 20 characters.
Type:           <class 'str'>
Default:        None
Current value:  None
--------------------------------------------------------------------------------
Please do one of the following to set the value of the above setting:
    - Input a new value
    - Press CTRL-C to exit
...
```

2. Run `sat showrev` to verify System Revision Information. The following tables contain example information.

```
ncn-m001# sat showrev
################################################################################
System Revision Information
################################################################################
+---------------------+---------------+
| component           | data          |
+---------------------+---------------+
| Company name        | HPE           |
| Country code        | US            |
| Interconnect        | Sling         |
| Product number      | R4K98A        |
| Serial number       | 12345         |
| Site name           | HPE           |
| Slurm version       | slurm 20.02.5 |
| System description  | Test System   |
| System install date | 2021-01-29    |
| System name         | eniac         |
| System type         | Shasta        |
+---------------------+---------------+
################################################################################
```

```
Product Revision Information
################################################################################
+--------------+-----------------+------------------------------+------------------------------+
| product_name | product_version | images                       | image_recipes                |
+--------------+-----------------+------------------------------+------------------------------+
| csm          | 0.8.14          | cray-shasta-csm-sles15sp1... | cray-shasta-csm-sles15sp1... |
| sat          | 2.0.1           | -                            | -                            |
| sdu          | 1.0.8           | -                            | -                            |
| slingshot    | 0.8.0           | -                            | -                            |
| sma          | 1.4.12          | -                            | -                            |
+--------------+-----------------+------------------------------+------------------------------+
################################################################################
Local Host Operating System
################################################################################
+-----------+----------------------+
| component | version              |
+-----------+----------------------+
| Kernel    | 5.3.18-24.15-default |
| SLES      | SLES 15-SP2          |
+-----------+----------------------+
```

# 5 SAT Post-Upgrade

## 5.1 Optional: Remove old versions after an upgrade

### 5.1.1 Prerequisites

- The Install the System Admin Toolkit Product Stream procedure has been successfully completed.
- The Perform NCN Personalization procedure has been successfully completed.

### 5.1.2 Procedure

After upgrading from a previous version of SAT, the old version of the `cray/cray-sat` container image will remain in the registry on the system. It is **not** removed automatically, but it will **not** be the default version.

The admin can remove the older version of the `cray/cray-sat` container image.

The `cray-product-catalog` Kubernetes configuration map will also show all versions of SAT that are installed. The command `sat showrev --products` will display these versions. See the example:

```
ncn-m001# sat showrev --products
################################################################################
Product Revision Information
################################################################################
+--------------+-----------------+--------------------+------------------------+
| product_name | product_version | images             | image_recipes          |
+--------------+-----------------+--------------------+------------------------+
...
| sat          | 2.1.3           | -                  | -                      |
| sat          | 2.0.4           | -                  | -                      |
...
+--------------+-----------------+--------------------+------------------------+
```

## 5.2 Remove obsolete configuration file sections

### 5.2.1 Prerequisites

- The Install the System Admin Toolkit Product Stream procedure has been successfully completed.
- The Perform NCN Personalization procedure has been successfully completed.

### 5.2.2 Procedure

After upgrading SAT, if using the configuration file from a previous version, there may be configuration file sections no longer used in the new version. For example, when upgrading from Shasta 1.4 to Shasta 1.5, the `[redfish]` configuration file section is no longer used. In that case, the following warning may appear upon running `sat` commands.

```
WARNING: Ignoring unknown section 'redfish' in config file.
```

Remove the `[redfish]` section from `/root/.config/sat/sat.toml` to resolve the warning.

```
[redfish]
username = "admin"
password = "adminpass"
```

Repeat this process for any configuration file sections for which there are “unknown section” warnings.

# 6 SAT Dashboards

## 6.1 SAT Kibana Dashboards

Kibana is an open source analytics and visualization platform designed to search, view, and interact with data stored in Elasticsearch indices. Kibana runs as a web service and has a browser-based interface. It offers visual output of node data in the forms of charts, tables and maps that display real-time Elasticsearch queries. Viewing system data in this way breaks down the complexity of large data volumes into easily understood information.

Kibana can be accessed via web browser at the following URL:

- `https://sma-kibana.<system_name>.<system_domain>`

For additional details about how to access the Kibana Dashboards refer to ***View Logs Via Kibana*** in the SMA product documentation.

Additional details about the AER, ATOM, Heartbeat, Kernel, MCE, and Rasdaemon Kibana Dashboards are included in this table.

| Dashboard | Short Description | Long Description | Kibana Visualization and Search Name |
|---|---|---|---|
| sat-aer | AER corrected | Corrected Advanced Error Reporting messages from PCI Express devices on each node. | Visualization: aer-corrected Search: sat-aer-corrected |
| sat-aer | AER fatal | Fatal Advanced Error Reporting messages from PCI Express devices on each node. | Visualization: aer-fatal Search: sat-aer-fatal |
| sat-atom | ATOM failures | Application Task Orchestration and Management tests are run on a node when a job finishes. Test failures are logged. | sat-atom-failed |
| sat-atom | ATOM admindown | Application Task Orchestration and Management test failures can result in nodes being marked admindown. An admindown node is not available for job launch. | sat-atom-admindown |

| Dashboard | Short Description | Long Description | Kibana Visualization and Search Name |
|---|---|---|---|
| sat-heartbeat | Heartbeat loss events | Heartbeat loss event messages reported by the hbtd pods that monitor for heartbeats across nodes in the system. | sat-heartbeat |
| sat-kernel | Kernel assertions | The kernel software performs a failed assertion when some condition represents a serious fault. The node goes down. | sat-kassertions |
| sat-kernel | Kernel panics | The kernel panics when something is seriously wrong. The node goes down. | sat-kernel-panic |
| sat-kernel | Lustre bugs (LBUGs) | The Lustre software in the kernel stack performs a failed assertion when some condition related to file system logic represents a serious fault. The node goes down. | sat-lbug |
| sat-kernel | CPU stalls | CPU stalls are serous conditions that can reduce node performance, and sometimes cause a node to go down. Technically these are Read-Copy-Update stalls where software in the kernel stack holds onto memory for too long. Read-Copy-Update is a vital aspect of kernel performance and rather esoteric. | sat-cpu-stall |
| sat-kernel | Out of memory | An Out Of Memory (OOM) condition has occurred. The kernel must kill a process to continue. The kernel will select an expendable process when possible. If there is no expendable process the node usually goes down in some manner. Even if there are expendable processes the job is likely to be impacted. OOM conditions are best avoided. | sat-oom |

| Dashboard | Short Description | Long Description | Kibana Visualization and Search Name |
|---|---|---|---|
| sat-mce | MCE | Machine Check Exceptions (MCE) are errors detected at the processor level. | sat-mce |
| sat-rasdaemon | rasdaemon errors | Errors from the `rasdaemon` service on nodes. The `rasdaemon` service is the Reliability, Availability, and Serviceability Daemon, and it is intended to collect all hardware error events reported by the linux kernel, including PCI and MCE errors. This may include certain HSN errors in the future. | sat-rasdaemon-error |
| sat-rasdaemon | rasdaemon messages | All messages from the `rasdaemon` service on nodes. | sat-rasdaemon |

### 6.1.1 Disable Search Highlighting in Kibana Dashboard

By default, search highlighting is enabled. This procedure instructs how to disable search highlighting.

The Kibana Dashboard should be open on your system.

1. Navigate to **Management**
2. Navigate to **Advanced Settings** in the Kibana section, below the Elastic search section
3. Scroll down to the **Discover** section
4. Change **Highlight results** from ***on*** to ***off***
5. Click **Save** to save changes

### 6.1.2 AER Kibana Dashboard

The AER Dashboard displays errors that come from the PCI Express Advanced Error Reporting (AER) driver. These errors are split up into separate visualizations depending on whether they are fatal or corrected errors.

#### 6.1.2.1 View the AER Kibana Dashboard

1. Go to the dashboard section.
2. Select **sat-aer** dashboard.
3. Choose the time range of interest.
4. View the Corrected and Fatal Advanced Error Reporting messages from PCI Express devices on each node. View the matching log messages in the panel(s) on the right, and view the counts of each message per NID in the panel(s) on the left. If desired, results can be filtered by NID by clicking the icon showing a **+** inside a magnifying glass next to each NID.

### 6.1.3 ATOM Kibana Dashboard

The ATOM (Application Task Orchestration and Management) Dashboard displays node failures that occur during health checks and application test failures. Some test failures are of ***possible*** interest even though a node is not marked **admindown** or otherwise fails. They are of ***clear*** interest if a node is marked **admindown**, and might provide clues if a node otherwise fails. They might also show application problems.

#### 6.1.3.1 View the ATOM Kibana Dashboard

HPE Cray EX is installed on the system along with the System Admin Toolkit, which contains the ATOM Kibana Dashboard.

1. Go to the dashboard section.
2. Select **sat-atom** dashboard.
3. Choose the time range of interest.
4. View any nodes marked **admindown** and any ATOM test failures. These failures occur during health checks and application test failures. Test failures marked **admindown** are important to note. View the matching log messages in the panel(s) on the right, and view the counts of each message per NID in the panel(s) on the left. If desired, results can be filtered by NID by clicking the icon showing a **+** inside a magnifying glass next to each NID.

### 6.1.4 Heartbeat Kibana Dashboard

The Heartbeat Dashboard displays heartbeat loss messages that are logged by the hbtd pods in the system. The hbtd pods are responsible for monitoring nodes in the system for heartbeat loss.

#### 6.1.4.1 View the Heartbeat Kibana Dashboard

1. Go to the dashboard section.
2. Select **sat-heartbeat** dashboard.
3. Choose the time range of interest.
4. View the heartbeat loss messages that are logged by the hbtd pods in the system. The hbtd pods are responsible for monitoring nodes in the system for heartbeat loss.View the matching log messages in the panel.

### 6.1.5 Kernel Kibana Dashboard

The Kernel Dashboard displays compute node failures such as kernel assertions, kernel panics, and Lustre LBUG messages. The messages reveal if Lustre has experienced a fatal error on any compute nodes in the system. A CPU stall is a serious problem that might result in a node failure. Out-of-memory conditions can be due to applications or system problems and may require expert analysis. They provide useful clues for some node failures and may reveal if an application is using too much memory.

#### 6.1.5.1 View the Kernel Kibana Dashboard

1. Go to the dashboard section.
2. Select **sat-kernel** dashboard.
3. Choose the time range of interest.
4. View the compute node failures such as kernel assertions, kernel panics, and Lustre LBUG messages. View the matching log messages in the panel(s) on the right, and view the counts of each message per NID in the panel(s) on the left. If desired, results can be filtered by NID by clicking the icon showing a **+** inside a magnifying glass next to each NID.

### 6.1.6 MCE Kibana Dashboard

The MCE Dashboard displays CPU detected processor-level hardware errors.

#### 6.1.6.1 View the MCE Kibana Dashboard

1. Go to the dashboard section.
2. Select **sat-mce** dashboard.
3. Choose the time range of interest.
4. View the Machine Check Exceptions (MCEs) listed including the counts per NID (node). For an MCE, the CPU number and DIMM number can be found in the message, if applicable. View the matching log messages in the panel(s) on the right, and view the counts of each message per NID in the panel(s) on the left. If desired, results can be filtered by NID by clicking the icon showing a **+** inside a magnifying glass next to each NID.

### 6.1.7 Rasdaemon Kibana Dashboard

The Rasdaemon Dashboard displays errors that come from the Reliability, Availability, and Serviceability (RAS) daemon service on nodes in the system. This service collects all hardware error events reported by the linux kernel, including PCI and MCE errors. As a result there may be some duplication between the messages presented here and the messages presented in the MCE and AER dashboards. This dashboard splits up the messages into two separate visualizations, one for only messages of severity "emerg" or "err" and another for all messages from `rasdaemon`.

#### 6.1.7.1 View the Rasdaemon Kibana Dashboard

1. Go to the dashboard section.
2. Select **sat-rasdaemon** dashboard.
3. Choose the time range of interest.
4. View the errors that come from the Reliability, Availability, and Serviceability (RAS) daemon service on nodes in the system. View the matching log messages in the panel(s) on the right, and view the counts of each message per NID in the panel(s) on the left. If desired, results can be filtered by NID by clicking the icon showing a **+** inside a magnifying glass next to each NID.

## 6.2 SAT Grafana Dashboards

The SAT Grafana Dashboards display messages that are generated by the HSN (High Speed Network) and reported through Redfish. The messages are displayed based on severity.

Grafana can be accessed via web browser at the following URL:

- `https://sma-grafana.<system_name>.<system_domain>`

For additional details about how to access the Grafana Dashboards refer to ***Access the Grafana Monitoring UI*** in the SMA product documentation.

For more information about the interpretation of metrics for the SAT Grafana Dashboards refer to ***Fabric Telemetry Kafka Topics*** in the SMA product documentation.

### 6.2.1 Navigate SAT Grafana Dashboards

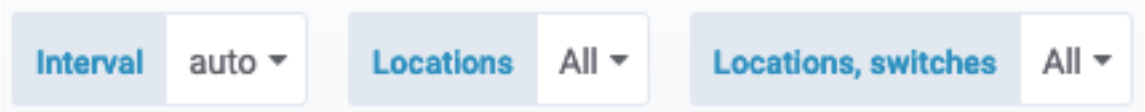
There are four Fabric Telemetry dashboards used in SAT that report on the HSN. Two contain chart panels and two display telemetry in a tabular format.

| Dashboard Name | Display Type |
|---|---|
| Fabric Congestion | Chart Panels |
| Fabric RFC3635 | Chart Panels |
| Fabric Errors | Tabular Format |
| Fabric Port State | Tabular Format |

The tabular format presents a single point of telemetry for a given location and metric, either because the telemetry is not numerical or that it changes infrequently. The value shown is the most recently reported value for that location during the time range selected, if any. The interval setting is not used for tabular dashboards.

### 6.2.2 SAT Grafana Interval and Locations Options

Shows the Interval and Locations Options for the available telemetry.




The value of the **Interval** option sets the time resolution of the received telemetry. This works a bit like a histogram, with the available telemetry in an interval of time going into a "bucket" and averaging out to a single point on the chart or table. The special value ***auto*** will choose an interval based on the time range selected.

For additional information, refer to Grafana Templates and Variables.

The **Locations** option allows restriction of the telemetry shown by locations, either individual links or all links in a switch. The selection presented updates dynamically according to time range, except for the errors dashboard, which always has entries for all links and switches, although the errors shown are restricted to the selected time range.

The chart panels for the RFC3635 and Congestion dashboards allow selection of a single location from the chart's legend or the trace on the chart.

### 6.2.3 Grafana Fabric Congestion Dashboard

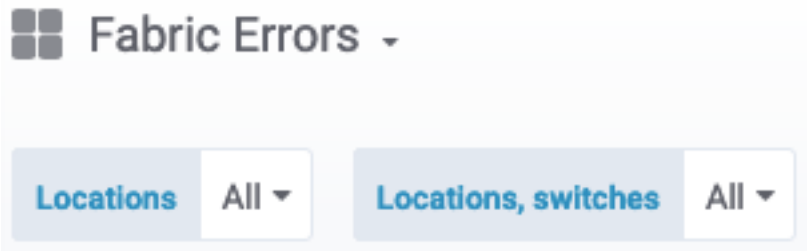

SAT Grafana Dashboards provide system administrators a way to view fabric telemetry data across all Rosetta switches in the system and assess the past and present health of the high-speed network. It also allows the ability to drill down to view data for specific ports on specific switches.

This dashboard contains the variable, **Port Type** not found in the other dashboards. The possible values are ***edge***, ***local***, and ***global*** and correspond to the link's relationship to the network topology. The locations presented in the panels are restricted to the values (any combination, defaults to "all") selected.

The metric values for links of a given port type are similar in value to each other but very distinct from the values of other types. If the values for different port types are all plotted together, the values for links with lower values are indistinguishable from zero when plotted.

The port type of a link is reported as a port state "subtype" event when defined at port initialization.

### 6.2.4 Grafana Fabric Errors Dashboard

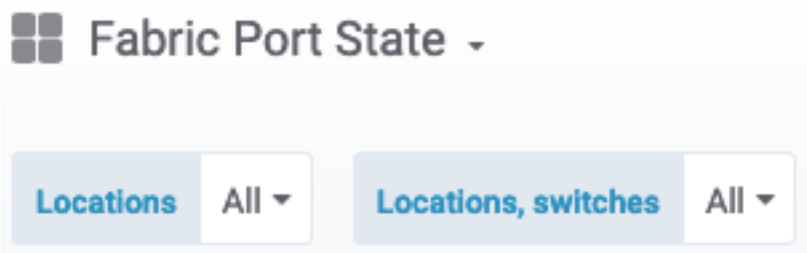



This dashboard reports error counters in a tabular format in three panels.

There is no **Interval** option because this parameter is not used to set a coarseness of the data. Only a single value is presented that displays the most recent value in the time range.

Unlike other dashboards, the locations presented are all locations in the system rather than having telemetry within the time range selected. However, the values are taken from telemetry within the time range.

### 6.2.5 Grafana Fabric Port State Dashboard

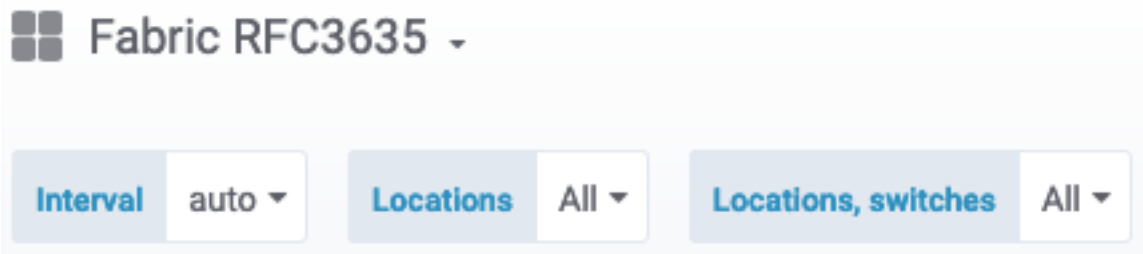


There is no **Interval** option because this parameter is not used to set a coarseness of the data. Only a single value is presented that displays the most recent value in the time range.

The Fabric Port State telemetry is distinct because it typically is not numeric. It also updates infrequently, so a long time range may be necessary to obtain any values. Port State is refreshed daily, so a time range of 24 hours results in all states for all links in the system being shown.

The three columns named, ***group***, ***switch***, and ***port*** are not port state events, but extra information included with all port state events.

### 6.2.6 Grafana Fabric RFC3635 Dashboard




For additional information on performance counters, refer to Definitions of Managed Objects for the Ethernet-like Interface Types, an Internet standards document.

Because these metrics are counters that only increase over time, the values plotted are the change in the counter's value over the interval setting.