Engineering
Simplicity

Address Pool Manager Installation Guide

Published
2026-02-09

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Address Pool Manager Installation Guide

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at https://support.juniper.net/support/eula/. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

About This Guide

Use this guide to install and configure Address Pool Manager (APM) on a Kubernetes orchestration platform.

1

CHAPTER

Juniper APM Installation

IN THIS CHAPTER

APM Installation

IN THIS SECTION

APM Installation Overview

Juniper Address Pool Manager (APM) is an automated, centralized, container-based cloud-native application that network operators and administrators use to manage IP prefix resources. APM works with managed broadband network gateways (BNGs) to monitor address pools on BNGs. When the number of free addresses drops below a set threshold, the BNG raises an alarm. The alarm triggers APM to allocate unused prefixes from its global list of prefixes and provision a subset of the prefixes to the BNG as new pools.

APM can be installed on a single Kubernetes cluster or on a multiple geography, multiple cluster setup. The installation requirements and installation process for these two types of setups are different. See the followings sections for the requirements for your APM setup:

- "APM Requirements for a Single Geography Setup" on page 3
- "APM Requirements for a Multiple Geography Setup" on page 7

NOTE: The term *BNG* in this document also applies to the BNG CUPS Controller.

You can deploy APM on any hardware that meets the requirements. The following sections describe:

- APM installation requirements
- How to install APM
- How to adjust APM setup parameters

APM Installation Requirements

IN THIS SECTION

To install APM, you need the following hardware and software requirements listed in this section.

APM Requirements for a Single Geography Setup

APM installs on a Kubernetes cluster comprised of physical or virtual machines (VMs). For availability, you must have at least three nodes hosting the control plane's etcd function and three nodes hosting the worker function in the cluster.

APM has been qualified against the single geography cluster described in Table 1 on page 4.

Table 1: Single Kubernetes Cluster Setup Requirements

Category	Details
Kubernetes cluster	The Kubernetes cluster requires the following: • Node specification: • Use one of the following for Kubernetes distribution: • RedHat OpenShift Container Platform 4.16.50 or 4.18.31 (required for production clusters). Configured with Longhorn (v1.9.1) CSI, and compatible MetalLB, NmState Operator, OVN CNI, and OpenShift Container Image Registry software. • BBE Cloudsetup utility— Used for proof of concept (POC) or demonstration clusters only. The BBE Cloudsetup utility builds a cluster with RKE1 distribution with versions of Longhorn CSI, MetalLB, Flannel CNI, and Docker Container Registry software. • CPU: 14 cores or 28 cores. Use a 28 core node if you plan on running other applications on the cluster (such as the BNG CUPS Controller application). • Memory: 256 gibibytes (GiB) • Storage: 512 GiB partitioned as 256 GiB root, 256 GiB /var/lib/longhorn • Network Interfaces: 2x10 GE • Cluster width—The cluster must have a minimum of 3 combined nodes with control plane, etcd, and worker functions. The 3 combined nodes supports APM and BNG CUPS Controller with a single control plane instance. If you are adding additional BNG CUPS Controller control plane instances, you

Table 1: Single Kubernetes Cluster Setup Requirements ***(Continued)***

Category	Details
	must add another worker node for each additional pair of BNG CUPS Controller control plane instance that you add.
Jump host	The jump host requires the following: • Operating system: Ubuntu version 22.04 LTS or 24.04 LTS • CPU: 2 core • Memory: 8 gibibytes (GiB) • Storage: 128 (GiB)

Table 1: Single Kubernetes Cluster Setup Requirements *(Continued)*

Category	Details
Jump host software	The jump host requires the following software: • POC systems: • Ubuntu 22.04 LTS • Python 3.10-venv • The BBE Cloudsetup utility will install compatible versions of the Kubernetes CLI, Helm, docker-ce, and other components necessary to orchestrate the application. • Production systems (RHOCP): • Ubuntu 22.04 LTS or Ubuntu 24.04 LTS. • Python 3.10-venv (Ubuntu 22.04) or 3.12-venv (Ubuntu 24.04) • Docker version 28.3.2. • Helm version 3.18.4. package manager. • Container version 1.7.27. • Compatible OpenShift and Kubernetes CLI (depends on the version of RHOCP that is installed on the cluster). You can use either of the following: • 4.16.50—https://mirror.openshift.com/pub/openshift-v4/x86_64/clients/ocp/4.16.50/openshift-client-linux.tar.gz • 4.18.31—https://mirror.openshift.com/pub/openshift-v4/x86_64/clients/ocp/4.18.31/openshift-client-linux.tar.gz
Storage	A storage class named jnpr-bbe-storage.
Network load balancer address	One for APMi.

Table 1: Single Kubernetes Cluster Setup Requirements *(Continued)*

Category	Details
Registry storage	Each APM release requires approximately 3GiB of container images.

APM Requirements for a Multiple Geography Setup

A geographically redundant, multiple cluster APM setup consists of three separate Kubernetes clusters. Each of the three clusters is geographically separated, so that service impacting events affecting one cluster do not affect the other clusters. The clusters which comprise the multiple cluster setup take on specific roles.

One cluster takes on the role of the management cluster and the other two clusters take on the role of workload clusters. The workload clusters provide a redundant platform where most of the APM application runs. The management cluster hosts execution of the Karmada multiple cluster orchestration software. Karmada manages the propagation of APM workloads across the workload clusters.

Each Kubernetes cluster node in a multiple cluster can be constructed from a physical or virtual machine (VM).

For POC installations, you cannot use the BBE cloudsetup utility to build the clusters used in a multiple geography multiple cluster setup. A separate procedure is available through support to build a POC multiple geography multiple cluster setup.

APM has been qualified against the multiple geography cluster described in Table 2 on page 7.

Table 2: Multiple Geography Kubernetes Cluster Setup Requirements

Category	Details
Cluster	The multiple cluster consists of three clusters with each cluster consisting of 3 hybrid nodes. The three clusters must consist of one management cluster and two workload clusters. **NOTE**: Make sure that the cluster and service CIDRs for each workload cluster do not overlap. The cluster internal networks of each workload cluster are connected by a Submariner IP tunnel. The internal CIDRS must be distinct.

Table 2: Multiple Geography Kubernetes Cluster Setup Requirements *(Continued)*

Category	Details
Management Cluster	The management cluster requires the following: - Node specification: - Kubernetes distribution—RedHat OpenShift Container Platform 4.16.50 or 4.18.31. Configured with compatible Longhorn CSI, MetalLB, NmState Operator, OVN CNI, and OpenShift Container Image Registry software. - Multiple cluster software: - Karmada 1.13.1—Multiple cluster orchestration. - Submariner 0.20.0—Inter-cluster network tunnel. - CPU: 8 cores - Memory: 24 GiB - Storage: 256 GiB partitioned as 160 GiB root and 96 GiB /var/lib/longhorn - Cluster width—The cluster must have a minimum of 3 combined nodes with control-plane, etcd, and worker functions.

Table 2: Multiple Geography Kubernetes Cluster Setup Requirements *(Continued)*

Category	Details
Workload cluster	Each workload cluster requires the following: • Node specification: • Kubernetes distribution—RedHat OpenShift Container Platform 4.16.50 or 4.18.31. Configured with Longhorn (v1.9.1) CSI, and compatible MetalLB, NmState Operator, OVN CNI, and OpenShift Container Image Registry software. Each workload cluster must use non-overlapping pod and service CIDRs in order to be connected through the Submariner tunnel. • Multiple cluster software—Submariner 0.20.0, for inter-cluster network tunnel. • CPU: 14 cores or 28 cores. Use a 28 core node if you plan on running other applications on the cluster (such as the BNG CUPS Controller application). • Memory: 256 GiB • Storage: 512 GiB partitioned as 256 GiB root and 256 GiB /var/lib/longhorn • Network interface: 2x10 GE. • Cluster width—The cluster must have a minimum of 3 combined nodes with control-plane, etcd, and worker functions. Cluster width—The cluster must have a minimum of 3 combined nodes with control plane, etcd, and worker functions. The 3 combined nodes supports APM and BNG CUPS Controller with a single control plane instance. If you are adding additional BNG CUPS Controller control plane instances, you must add another worker node for each additional pair of BNG CUPS Controller control plane instance that you add.

Table 2: Multiple Geography Kubernetes Cluster Setup Requirements *(Continued)*

Category	Details
Jump host	The jump host requires the following: • Operating system: Ubuntu version 22.04 LTS or 24.04 LTS • CPU: 2 core • Memory: 8 gibibytes (GiB) • Storage: 128 (GiB)
Jump host software	The jump host requires the following software for production systems: • Ubuntu 22.04 LTS or Ubuntu 24.04 LTS. • Python 3.10-venv (Ubuntu 22.04) or 3.12-venv (Ubuntu 24.04) • Docker version 28.3.2. • Helm version 3.18.4. package manager • Container version 1.7.27. • Subctl version 0.20.0—Submariner CLI utility. • Kubectl Karmada version 1.13.1—Kubectl karmada plug-in. • Compatible OpenShift and Kubernetes CLI (depends on the version of RHOCP that is installed on the cluster). You can use either of the following: • 4.16.50—https://mirror.openshift.com/pub/openshift-v4/x86_64/clients/ocp/4.16.50/openshift-client-linux.tar.gz • 4.18.31—https://mirror.openshift.com/pub/openshift-v4/x86_64/clients/ocp/4.18.31/openshift-client-linux.tar.gz
Storage	A storage class named jnpr-bbe-storage

Table 2: Multiple Geography Kubernetes Cluster Setup Requirements *(Continued)*

Category	Details
Network load balancer addresses	One on each workload cluster for APMi and one on the management cluster for bbe-observer.
Registry storage	Each APM release requires approximately 3 gibibytes (GiB) of container images. Required for each cluster.

NOTE: In a single geography APM setup, you can make some basic assumptions about the cluster's parameters. You can use a quick start tool like BBE Cloudsetup to create a single geography APM. The construction of a production environment APM setup with multiple geographies and multiple clusters requires much more input from you to build.

Additional Requirements

The BNG is a Juniper Networks MX Series router running Junos OS or a Juniper BNG CUPS Controller (BNG CUPS Controller).

We recommend the following releases:

- Junos OS Release 23.4R2-S5 or later
- BNG CUPS Controller 24.4R2 or later

For APM, confirm that you have a juniper.net user account with permissions to download the APM software package. Download and install the APM software from a machine that will not be part of the Kubernetes cluster.

Install a Single Geography APM

SUMMARY

Use the procedures in this section to install a single geography APM for the first time.

Before you begin, confirm that you have met the requirements for the APM installation.

We recommend that you use a secure connection between APM and the BNG.

You have the following two options for installing APM:

- "Install a Single Geography APM Using the APM installation Utility" on page 14—You can install APM using the APM utility, which streamlines the installation process.

  NOTE: BBE Cloudsetup is a utility that you can use to quickly get started with using APM. It is not a life cycle tool for the cluster. You cannot expand the width of the cluster, perform node maintenance, upgrade infrastructure components, and so on. A Kubernetes cluster for production purposes should be designed and constructed with the requirements of the production environment and with appropriate support to maintain its life cycle. (For information about BBE Cloudsetup, see BBE Cloudsetup Installation Guide.)

- "Install a Single Geography APM Without Using the APM Utility " on page 26—You can install APM on a preexisting Kubernetes cluster of your choice. This process is a manual process and does not use the APM utility that comes with the APM installation package.

Before starting the APM installation, make sure that you have the following information:

Required Information:

- Container registry details:
  - If you are using a BBE Cloudsetup created cluster.
    - External registry address.
    - External registry port number (usually 5000).
  - If you are using a Red Hat OpenShift Container Platform cluster:
    - External registry (fully qualified domain name)
    - Internal (Docker) registry address
    - Internal (Docker) registry port number

Optional Information:

- APM initial configuration file. If a configuration file is not supplied, a basic configuration file is automatically generated.

- Storage class name for persistent volume claim (PVC) creation (default is jnpr-bbe-storage).

- PVC Size (default is 90 MiB).
- Archival configuration details. This is required if you are planning to mirror a copy of the APM configuration to an external server.
  - Either the name of the SSH private key file or the name of the Kubernetes secret that is present in the jnpr-apm namespace containing the SSH private key.
  - The Secure Copy Protocol (SCP) URL of the server where the configuration file will be archived. An SCP URL takes the form of `scp://`*`user-login@server-fqdn`*`:`*`server-port/absolute-file-path`* (for example, `scp://`*`user@host1.mydomain.com`*`:30443/home/user/configs/apm`).
- Syslog server details. This is required if you are planning to export APM logs to an external syslog collector.

  > **NOTE**: If BBE Event Collection and Visualization is detected running on the target cluster, the address and port values of the ECAV deployment will be suggested as the default.

  - Syslog server address.
  - Sysylog server port number.
- Network load balancer details. This is required if you are planning to use a specific network load balancer pool and address for APMi.
  - Network load balancer pool name.
  - Network load balancer pool address.
- APMi Details:
  - Port (default is 20557)
  - TLS details. You will need one of the following:
    - None (insecure)
    - Either the key and certificate files or the name of the Kubernetes secret that is present in the jnpr-apm namespace that contains the key and certificate information.
- Service Account Name—The name of the Kubernetes service account used to bind certain operational privileges to the mgmt microservice. If a service account name is not provided, APM creates a service account named apm-svca during rollout.
- SSH service type—If SSH access to the mgmt microservice is specified (`--ssh <ip>:<port>`), you must specify whether the service should be created as a node port (**NodePort**) service or a load balancer (**LoadBalancer**) service. If **LoadBalancer** is selected, a MetalLB pool is created containing the supplied

external IP address. The load balancer service created at rollout is assigned the external IP address from the newly created MetalLB pool.

- DBSync service type—The `apm multicluster status` APM utility command collects the state to display from the DBSync microservice through a Kubernetes service. By default, a node port service is created for this purpose. If you select `LoadBalancer`, you are prompted for an external IP address and a MetalLB pool is created containing the supplied external IP address. The `LoadBalancer` service created at rollout is assigned the external IP address from the newly created MetalLB pool.

- Number of worker processes for the provman microservice (default is 3).

Install a Single Geography APM Using the APM installation Utility

SUMMARY

You use the procedure in this section if you are installing a single geography APM.

1. Download the APM software package from the Juniper Networks software download page to the jump host.

   APM is available as a compressed TAR (**.tgz**) file. The filename includes the release number as part of the name. The release number has the format: <Major>.<Minor>.<Maintenance>

   - *major* is the main release number of the product.

   - *minor* is the minor release number of the product.

   - *maintainance* is the revision number.
2. Unpack the APM TAR (**.tgz**) file on the jump host by entering:

```
$ tar -zxvf apm-m.m.m.tgz
apm/
apm/apm_loader
apm/images/
apm/images/apm_containerImages.tar.gz
apm/charts/
apm/charts/provman/
apm/charts/provman/templates/
apm/charts/provman/templates/apmProv-man.yaml
```

```
apm/charts/provman/templates/apmApmiSvc.yaml
apm/charts/provman/questions.yaml
apm/charts/provman/Chart.yaml
apm/charts/provman/containers.yaml
apm/charts/provman/values.yaml
apm/charts/provman/.helmignore
apm/charts/entman/
apm/charts/entman/templates/
apm/charts/entman/templates/apmEnt-man.yaml
.
.
.
```

3. Run the loader script after you unpack the TAR file.

```
$ sudo apm/apm_loader
Creating apm group... done.
Loading files... done.
Updating latest link... done
Setting up utility script... done.
Updating wrapper... done
Successfully loaded: 3.5.0
```

4. Use the `sudo -E apm link --context` *context-name* `--version` *apm-version* command to link to the cluster. The `link` command associates the loaded APM software package to the cluster context in preparation for the setup.

```
$ sudo -E apm link --context context-name --version 3.5.0
Linking context-name to 3.5.0 ... done.
Linking complete, please run apm setup.
```

- *context-name* is the Kubernetes context.
- *apm-version* is the software version.

5. If you are installing APM on a Red Hat OpenShift Container Platform cluster, log in with the OpenShift CLI and then proceed to the next step.

   If you are installing APM on a BBE Cloudsetup created cluster, proceed to the next step.

6. You must authenticate with the container registry in order to be able to push the APM container images. How you authenticate to the registry varies depending on if you are installing APM on a BBE Cloudsetup created cluster or on an Red Hat OpenShift Container Platform cluster (see the respective documentation for details).

7. Run `setup` to configure your installation. The `setup` command does the following:

   - Collects information about the cluster environment such as: container registry contact information, keys and certificates needed to secure external interfaces, persistent storage resources, and other information relevant to supporting APM features.

   - Establishes the operational parameters for the Kubernetes deployment.

     If you did not use either the `bbecloudsetup` option or the `template` *`file-name`* option with the setup command, you need to complete these prompts during the setup:

     - If you are using BBE Cloudsetup to create your cluster.

       - External registry address.

       - External registry port number.

     - If you are using a Red Hat OpenShift Container Platform cluster:

       - External registry (fully qualified domain name)

       - Internal (Docker) registry address

       - Internal (Docker) registry port number

   NOTE: When running `setup`, you can interact with the setup process by entering **^d**. If you want to change a value after entering it, enter **^d**. After entering **^d**, the value you previously entered is removed and the default value is automatically used for the question. You can use the **^d** operation for any setup questions that are optional or for which a list of values can be provided.

   ```
   $ sudo -E apm setup --context context-name [--help] [--log info] [--no color] [--
   bbecloudsetup] [--update] [--ssh host:port-number] [--secrets] [--verbose] [--config file-
   name] [--template file-name] [--mandatory] [--optional]
   ```

   NOTE: `context` *`context-name`* is the only required option for the `setup` command.

   The options that you can use with the `setup` command are listed in the following:

   - `context` *`context-name`*—The Kubernetes context name.

   - `h, help`—Shows the help message and exit.

   - `l, log [error, warning, info, debug]`—Adjusts the log level.

- `no-color`—Prints messages without colors.
- `bbecloudsetup`—Fills in operational parameters that align with a bbecloudsetup created cluster so that you do not have to interact with APM during the setup process (see the BBE Cloudsetup Installation Guide for cluster installation instructions).

  NOTE: Only use either the `bbecloudsetup` option or the `template` *`file-name`* option. Do not use both options.

- `update`—You will only be prompted for missing values during setup.
- `ssh` *`host:port`*—A hostname or IP address of the cluster (any of the cluster's nodes) and open port used for SSH access to the CLI.

  NOTE: Enabling SSH access requires the MGMT microservice to run in *privileged* mode.

- `secrets`—Updates the keys, certificates, and secrets used by APM.
- `verbose`—Provides a detailed description before each prompted question.
- `config` *config-file-path-name*—The name of the initial configuration file that you want APM to use during startup.

  NOTE: You can use an initial configuration file to start and roll out APM. You use the configuration file through the `--config` *`config-file-path-name`* switch on the utility script's `setup` command.

  ```
  sudo -E apm setup -context context-name --config config-file-path-name
  ```

  When APM is started or rolled out, the configuration file that you supply during setup is used to initialize APM. If you do not supply a configuration file, APM starts with the factory defaults. If the BBE Event Collection and Visualization application is running on the cluster, the factory defaults include the bbe-ecav syslog server configuration.

  The supplied configuration file is stored on the jumphost's context repository. This allows the configuration to be preserved across APM start and stop events. Commits to the initial configuration are not automatically saved to the persistent location on the jumphost. To update the configuration at the persistent location, use the utility script's save-config command.

```
sudo -E apm save-config --context context-name
```

Using the save-config command ensures that the latest configuration is used the next time that APM is started and rolled out. In order to restore the initial configuration back to its factory default, enter setup interactively and enter **^d** to the *startup config ...* question.

```
sudo -E apm setup -context context-name
.
.
.
Startup config to mount into mgmt pod on rollout (deployed: true) > ^d
```

The action in the parenthesis changes to *remove*. Press **Enter** to accept the removal of the deployed configuration. APM reverts back to the factory default configuration after a stop and then rollout command sequence.

When you change the initial configuration file using the utility script's setup command, you must perform a stop and then rollout command sequence for the change to take effect.

- template *file-name*—A YAML formatted file that contains a subset of the operational parameters file that is created during setup. The values that are entered in the template file are used automatically by the setup process. When you use the template option, you are not required to manually enter the information contained in the template file during the setup process. Use the template option when using Red Hat OpenShift Container Platform to create the cluster or when creating a multiple geography cluster. Table 3 on page 19 describes the information that you can enter into the template configuration file.

  NOTE: Only use either the bbecloudsetup option or the template *file-name* option. Do not use both options.

- mandatory—Only asks required questions during setup.

- optional—Only asks questions that are not required during setup.

Table 3: Setup File Field Descriptions

Field	Description
External registry address	The external registry address is a fully qualified domain name (FQDN) that the container images are pushed to.
Internal (Docker) registry transport address (fqdn:port)	The internal registry transport address is the address from which the container images are pulled from during rollout. This address is typically different than the external registry address.
(Optional) **Initial APM configuration file**	The configuration file that is used at APM startup.
(Optional) **Cluster storage-class name**	The name of the Kubernetes storage class to use for creating persistent volume claims (PVCs). The management microservice uses a PVC to record the configuration state.
(Optional) **Cluster storage size**	The PVC size in mebibytes (MiB).

Table 3: Setup File Field Descriptions *(Continued)*

Field	Description
(Optional) **Configuration archival server**	When you configure the **Configuration archival server** option, APM archives a copy of the updated configuration to an external server after each successful commit. To configure the server information where configuration file changes are archived, you must enter the following information: • ssh-key information. Provide information for one of the following: • The name of a Kubernetes Secret in the APM namespace that contains the SSH private key data. • The name of the SSH private-key file. **NOTE**: If a secret name is supplied, you will not be prompted for the SSH private-key file. • The Secure Copy Protocol (SCP) URL of the server where the configuration file will be archived. **NOTE**: The URL must use the following format: `scp://`*`user-login@server-fqdn`*`:`*`server-port/absolute-file-path`* (for example, `scp://`*`user@host1.mydomain.com`*`:30443/home/user/configs/apm`). Upon successful commit, an SCP transfer of the candidate configuration is transferred to the archival URL as a compressed file with the name: *`apm-identifier`*`_YYYYMMDD_HHMMSS_juniper.conf.`*`n`*`.gz` • *`apm-identifier`* is the external IP address of the APMi interface.

Table 3: Setup File Field Descriptions *(Continued)*

Field	Description
	- `YYYYMMDD_HHMMSS` is the time stamp in Coordinated Universal Time (UTC). - *n* is the number designation of the compressed configuration rollback file.
(Optional) **Syslog Details**	If you want to export APM log information to an external syslog collector,enter the following syslog server information: - IP address or fully qualified domain name - Port number Syslog information is included in the generated factory default configuration file. If you did not use the generated factory default configuration file, and used your own initial configuration file, you must include the `system syslog host` stanza containing the connection details for the syslog server.
(Optional) **Network Load Balancer Pool**	If you want the APMi external address to be allocated from a specific network load balancer address pool, enter the following network load balancer pool information: - Network load balancer address annotation (for example: metallb.io/address-pool: *myMetallpAddressPool*) - Network load balancer pool annotation (for example: metallb.io/loadBalancerIPs: 10.1.1.3)
(Optional) **APMi port**	The APMi port number (default is 20557).

Table 3: Setup File Field Descriptions *(Continued)*

Field	Description
(Optional) **APMi secrets**	To secure the APMi (recommended),enter one of the following: • The name of a Kubernetes secret in the APM namespace that contains the TLS secret data (root Certificate Authority certificate, certificate, private-key) • Key files (root Certificate Authority certificate, certificate, and private key) **NOTE**: If a secret is provided, you will not be prompted for the Key files during installation.
(Optional) **Service Account Name**	The name of the Kubernetes service account used to bind certain operational privileges to the mgmt microservice. If a service account name is not provided, APM creates a service account named apm-svca during rollout.
(Optional) **Type of apm SSH service**	If SSH access to the mgmt microservice is specified (`--ssh <ip>:<port>`), you must specify whether the service should be created as a node port (**NodePort**) service or a load balancer (**LoadBalancer**) service. If **LoadBalancer** is selected, a MetalLB pool is created containing the supplied external IP address. The load balancer service created at rollout is assigned the external IP address from the newly created MetalLB pool.

Table 3: Setup File Field Descriptions *(Continued)*

Field	Description
(Optional) **DBSync service type**	The `apm multicluster status` APM utility command collects the state to display from the DBSync microservice through a Kubernetes service. By default, a node port service is created for this purpose. If you select `LoadBalancer`, you are prompted for an external IP address and a MetalLB pool is created containing the supplied external IP address. The `LoadBalancer` service created at rollout is assigned the external IP address from the newly created MetalLB pool.
(Optional) **Number of worker processes**	The number of provman worker processes determines how simultaneous processes provman deploys to handle the entity workload. We suggest that you plan for 20 entities per process. Each process can consume a CPU core on the node it is running on. Therefore, the nodes in the cluster must have sufficient CPU cores to support the number of provman processes (plus any other workloads that may be running on a node). You can configure 1 to 10 worker process (default is 3).
(Optional) **Provide service account for the Observer's Controller Manager**	The service account for the controller manager. If you select `True` for this prompt, you are prompted for a service account name. If you select `False`, the `bbe-observer-controller-manager` service account is created.
(Optional) **Provide service account for the Observer's gRPC server**	The service account for the gRPC server. If you select `True` for this prompt, you are prompted for a service account name. If you select `False`, the `bbe-observer-grpc-server` service account is created.

8. Verify the APM installation by running the `apm version` command.

```
$ apm version --context context-name --detail
NOTE: Address_Pool_Manager not deployed
Address_Pool_Manager versions:
  Microservice  Release              (version)
  apm:          3.5.0
  addrman:      (3.5.0)
  entman:       (3.5.0)
  mgmt:         (3.5.0)  (24.4R1.1)
  provman:      (3.5.0)
  redis:        (3.5.0)  (6.2.16-debian-12-r3)
```

- `context` *`context-name`*—The Kubernetes context name.
- `detail`—Displays all available software versions.

Start APM in a Single Geography Setup

SUMMARY

Use this procedure to configure and to start APM in a single geography setup.

1. Enter `rollout` to start the APM installation. You need to use the rollout command with sudo/as root. The `rollout` command also validates that all the values needed for the new releases are present and loads the new release container images to the registry. Use `sudo -E apm rollout --context` *`context-name`* to start APM services. For example:

```
$ sudo -E apm rollout --context context-name
  RHOCP: adding privileged SCC to apm service account
Validating registries... done.
Load container images to registry...
  Loading 3.5.0 redis images to local cache... done.
  Loading 3.5.0 mgmt images to local cache... done.
  Loading 3.5.0 addrman images to local cache... done.
  Loading 3.5.0 entman images to local cache... done.
  Loading 3.5.0 provman images to local cache... done.
  Pushing 3.5.0 redis images to registry... done.
  Pushing 3.5.0 mgmt images to registry... done.
  Pushing 3.5.0 addrman images to registry... done.
  Pushing 3.5.0 entman images to registry... done.
  Pushing 3.5.0 provman images to registry... done.
Loaded container images to registry.
Rollout Address_Pool_Manager... done.
```

- `context` *`contex-name`*—The Kubernetes context.

NOTE: By default, APM starts with the values that you provided during setup. Unless the configuration was saved, the initial configuration is what was entered during setup. All other persistent states (logs, database keys, and so on) are cleared.

2. Enter `apm status --context` *`context-name`* `[-o|--output json] [--detail]` to verify that the APM services are up and running. For example:

```
$ apm status --context context-name --detail
MICROSERVICE POD                               STATE  RESTARTS UPTIME          NODE
addrman      jnpr-apm-addrman-7cff87b557-gp8s7 Running 0       0:01:52.755513  test-node-1.juniper.net
entman       jnpr-apm-entman-67d9bf9498-bx8jj  Running 0       0:01:49.755557  test-node-2.juniper.net
mgmt         jnpr-apm-mgmt-6c76cc8dd7-pmlpv    Running 0       0:01:56.755587  test-node-3.juniper.net
provman      jnpr-apm-provman-75bc8d465d-czcfm Running 0       0:01:36.755613  test-node-3.juniper.net
redis        jnpr-apm-redis-0                  Running 0       0:02:28.755636  test-node-1.juniper.net
redis        jnpr-apm-redis-1                  Running 0       0:02:15.755658  test-node-3.juniper.net
redis        jnpr-apm-redis-sentinels-0        Running 0       0:02:14.755679  test-node-3.juniper.net
redis        jnpr-apm-redis-sentinels-1        Running 0       0:02:13.755700  test-node-2.juniper.net
redis        jnpr-apm-redis-sentinels-2        Running 0       0:02:13.755722  test-node-1.juniper.net

Storage: Healthy
```

NOTE: Collect the logs for a service and contact the Juniper Networks Technical Assistance Center (JTAC) when either of the following occurs:

- The service is not running.
- The service's uptime compared with other services indicates that it has restarted.

Install a Single Geography APM Without Using the APM Utility

The instructions in this section describes the installation steps for installing a single geography APM on a preexisting Kubernetes cluster of your choice. This process is a manual process and does not use the APM utility that comes with the APM installation package.

1. Download the APM software package from the Juniper Networks software download page to the jump host.

   APM is available as a compressed TAR (**.tgz**) file. The filename includes the release number as part of the name. The release number has the format: <Major>.<Minor>.<Maintenance>

   - *major* is the main release number of the product.

   - *minor* is the minor release number of the product.

   - *maintainance* is the revision number.

2. Unpack the APM TAR (**.tgz**) file on the jump host by entering:

```
$ tar -zxvf apm-m.m.m.tgz
apm/
apm/apm_loader
apm/images/
apm/images/apm_containerImages.tar.gz
apm/charts/
apm/charts/provman/
apm/charts/provman/templates/
apm/charts/provman/templates/apmProv-man.yaml
apm/charts/provman/templates/apmApmiSvc.yaml
apm/charts/provman/questions.yaml
apm/charts/provman/Chart.yaml
apm/charts/provman/containers.yaml
apm/charts/provman/values.yaml
apm/charts/provman/.helmignore
apm/charts/entman/
apm/charts/entman/templates/
apm/charts/entman/templates/apmEnt-man.yaml
.
.
.
```

3. The container images needed by APM are stored in the **images** subdirectory. You must push the images to the registry where the scheduled application images will be pulled from. Depending on the

type of container registry being used the commands may be different. The following commands illustrate one method of pushing container images to the registry:

```
% docker image load -i ./images/apm_addr-man_containerImages.tar.gz
Loaded image: apm_addr-man:3.4.0
docker image load -i ./images/apm_init_containerImages.tar.gz
Loaded image: apm_init:3.4.0
```

```
% docker tag apm_addr-man:3.4.0 <regHost>:<regPort>/apm_addr-man:3.4.0
docker tag apm_init:3.4.0 [regHost]:[regPort]/apm_init:3.4.0
```

```
% docker push <regHost>:<regPort>/apm_addr-man:3.4.0
docker push [regHost]:[regPort]/apm_init:3.4.0
```

4. To prepare APM for deployment, you must create a YAML configuration file for each microservice. Each microservice's configuration file contains the specific configuration settings for the microservice. The YAML configuration file is called **values.yaml** and the file is located under the **charts** subdirectory, with each microservice. You should create a separate **values.yaml** (for example, ***new-values.yaml***) specific to your configuration for each microservice. Table 4 on page 28 describes the fields in the microservice's configuration files (**values.yaml**).

   NOTE: If you do not want to create multiple **values.yaml** files, you can create a single **values.yaml** that contains information for all the microservices. The single **values.yaml** is located under the umbrella chart in the **apm/apm/charts/address_pool_manager** folder. The procedures in this section describe how to configure an individual YAML configuration file for each microservice.

   Create a new **values.yaml** file for each of the microservices, by making a copy of the file and then saving the new file. Update each file according to your Kubernetes cluster's information.

   Following are the microservices and their **values.yaml** file location:

   - **redis** microservice—Located at **apm/apm/charts/redis**
   - **mgmt** microservice—Located at **apm/apm/charts/mgmt**
   - **addrman** microservice—Located at **apm/apm/charts/addrman**
   - **entman** microservice—Located at **apm/apm/charts/entman**.
   - **provman** microservice—Located at **apm/apm/charts/provman**

Table 4: Microservices Configuration File Field Descriptions

Field	Description	Microservice
APMi port	The APMi exposed port number.	provman
APMi secrets	- **name**—Name space secret to mount - **certificate**—Certificate file name - **key**—Private key file name - **rootca**—CA certificate file name	provman
apmInitVersion	APM init software version.	- mgmt - redis
archivalUrl	The Secure Channel Protocol (SCP) URL of the server where the configuration file is archived.	mgmt
db master updateStrategy	Only **RollingUpdate** is supported.	redis
evictionToleration	The node's unreachable tolerance (in seconds).	- addrman - entman - mgmt - provman - redis

Table 4: Microservices Configuration File Field Descriptions *(Continued)*

Field	Description	Microservice
log_level	The default logging level.	• addrman • entman • mgmt • provman • redis
nlbPoolAnnotation	The network load balancer pool name.	provman
nlbPoolIpAnnotation	The network load balancer IP address	provman
pvcs config	• **meta**—Persistent volume claim (PVC) for configuration file storage. • **size**—PVC size (MiB).	mgmt
registry	Registry information: • **host**—The registry contact for the cluster pulls. • **port**—The registry port number for cluster pulls.	• addrman • entman • mgmt • provman • redis

Table 4: Microservices Configuration File Field Descriptions *(Continued)*

Field	Description	Microservice
resourceRequestsEnabled	Whether or not to accept the resource request.	• **addrman** • **entman** • **mgmt** • **provman** • **redis**
resourceRanges	Required resource ranges: • **cpuRequest**—The minimum millicores that are required to operate the system. • **memRequest**—The minimum mebibytes (MiB) that are required to operate the system.	• **addrman** • **entman** • **mgmt** • **provman** • **redis** • **redis**
sentinelCount	The number of sentinels to start.	**redis**
startup config	The configuration to use for system startup.	**mgmt**
storage_class	Name of the storage class for PVC.	**mgmt**
tlsEnabled	Indicates if TLS is enabled.	**provman**
workerProcs	The number of worker processes that you want started.	**provman**

5. After you have made all the desired changes to your new **values.yaml** files for each microservice, the microservices must be deployed with the new **values.yaml** files.

Run the following commands:

```
helm upgrade --dependency-update --kube-context <cluster-context> --namespace jnpr-apm --
create-namespace --atomic --install -f ./charts/redis/new-values.yaml redis ./charts/redis
```

```
helm upgrade --dependency-update --kube-context <cluster-context> --namespace jnpr-apm --
create-namespace --atomic --install -f ./charts/mgmt/new-values.yaml mgmt ./charts/mgmt
```

```
helm upgrade --dependency-update --kube-context <cluster-context> --namespace jnpr-apm --
create-namespace --atomic --install -f ./charts/addrman/new-values.yaml addrman ./charts/
addrman
```

```
helm upgrade --dependency-update --kube-context <cluster-context> --namespace jnpr-apm --
create-namespace --atomic --install -f ./charts/entman/new-values.yaml entman ./charts/entman
```

6. To complete the configure of the provman microservice, you must run the **helm upgrade** command twice. The double configuration is required because the provman microservice uses the external IP address assigned to the APMi load balancer service in the protocol exchanges with its entities. The first **helm upgrade** command establishes the external IP address and the second **helm upgrade** command passes the external IP address to the provman microservice, which allows it to initialize.

   Run the following commands:

```
helm upgrade --dependency-update --kube-context <cluster-context> --namespace jnpr-apm --
create-namespace --atomic --install -f ./charts/provman/new-values.yaml provman ./charts/
provman
```

```
helm upgrade --dependency-update --kube-context <cluster-context> --namespace jnpr-apm --
create-namespace --atomic --install -f ./charts/provman/new-values.yaml provman ./charts/
provman
```

7. Verify the APM installation by running the Kubernetes Command Line Tool command `kubectl get pods` and verify the APM pods are running.

```
$ kubectl get pods -n jnpr-apm -o wide
NAME                              READY STATUS  RESTARTS AGE  IP          NODE                      NOMINATED NODE READINESS GATES
```

```
jnpr-apm-addrman-7cff87b557-gp8s7 1/1   Running  0       124m 10.42.2.15   jib.englab.juniper.net      <none>          <none>
jnpr-apm-entman-67d9bf9498-bx8jj  1/1   Running  0       124m 10.42.1.141  keel.englab.juniper.net     <none>          <none>
jnpr-apm-mgmt-6c76cc8dd7-pmlpv    1/1   Running  0       124m 10.42.0.20   binnacle.englab.juniper.net <none>          <none>
jnpr-apm-provman-75bc8d465d-czcfm 1/1   Running  0       123m 10.42.0.22   binnacle.englab.juniper.net <none>          <none>
jnpr-apm-redis-0                  1/1   Running  0       124m 10.42.2.16   jib.englab.juniper.net      <none>          <none>
jnpr-apm-redis-1                  1/1   Running  0       123m 10.42.0.21   binnacle.englab.juniper.net <none>          <none>
jnpr-apm-redis-sentinels-0        1/1   Running  0       124m 10.42.0.19   binnacle.englab.juniper.net <none>          <none>
jnpr-apm-redis-sentinels-1        1/1   Running  0       124m 10.42.1.142  keel.englab.juniper.net     <none>          <none>
jnpr-apm-redis-sentinels-2        1/1   Running  0       124m 10.42.2.17   jib.englab.juniper.net      <none>          <none>
```

8. Verify that the services are present. Run the Kubernetes Command Line Tool command `kubectl get services`.

```
$ kubectl get services -n jnpr-apm
NAME                           TYPE           CLUSTER-IP  EXTERNAL-IP    PORT(S)         AGE
jnpr-apm-mgmt-svc              ClusterIP    10.43.131.131 <none>         8066/TCP        125m
jnpr-apm-redis-sentinels-0-svc ClusterIP    10.43.142.53  <none>         7381/TCP        125m
jnpr-apm-redis-sentinels-1-svc ClusterIP    10.43.109.206 <none>         7381/TCP        125m
jnpr-apm-redis-sentinels-2-svc ClusterIP    10.43.104.100 <none>         7381/TCP        125m
jnpr-apm-redis-svc             ClusterIP    10.43.6.207   <none>         7380/TCP        125m
provman-apmi                   LoadBalancer 10.43.221.12  198.19.224.212 20557:32553/TCP 125m
```

Install APM in a Multiple Geography Setup

SUMMARY

Use the installation procedures in this section for an APM setup that consists of multiple APMs that are located in different geographical locations.

Before you begin, confirm that you meet the requirements for the APM installation (see Table 2 on page 7).

Prerequisites

Before starting the APM installation, make sure that you have the following information:

For descriptions of the following information, see Table 3 on page 19.

Required Information:

- The cluster context names of the workload clusters, the management cluster's karmada context and the management cluster's working context.

  For example, your context output might look like the following:

  ```
  kubectl config get-contexts
  CURRENT   NAME                CLUSTER             AUTHINFO          NAMESPACE
  *         karmada-apiserver   karmada-apiserver   karmada-admin
            mgmt                mgmt                mgmt-admin
            workload-a          workload-a          workload-a-admin
            workload-b          workload-b          workload-b-admin
  ```

- Karmada kubeconfig—The kubeconfig file for the Karmada context on the management cluster. You can extract the kubeconfig file for the Karmada context from the management cluster context in the karmada-system namespace.

  For an example of the command to run, see the following:

  ```
  kubectl get secrets -n karmada-system --context management-context-name -o
  jsonpath='{.data.karmada\.config}' | base64 -d > karmada-secret-file
  ```

- Container registry details for each cluster:

  NOTE: You must collect the following information for all three clusters.

  - External registry address
  - External registry port number (usually 5000)

Optional Information:

- Service account for the controller manager. If you select `True` for this prompt, you are prompted for a service account name. If you do not select `True`, a service account named `bbe-observer-controller-manager` is created.

- Service account for the gRPC server. If you select `True` for this prompt, you are prompted for a service account name. If you do not select `True`, a service account named `bbe-observer-grpc-server` is created.

- APM initial configuration file. If a configuration file is not supplied, a basic configuration file is automatically generated.

- Storage class name for persistent volume claim (PVC) creation (default is jnpr-bbe-storage).

- PVC Size (default is 90 MiB).

- Archival configuration details. This is required if you are planning to mirror a copy of the APM configuration to an external server.

  - Either the name of the SSH private key file or the name of the Kubernetes secret that is present in the jnpr-apm namespace containing the SSH private key.

  - The Secure Copy Protocol (SCP) URL of the server where the configuration file will be archived. An SCP URL takes the form of `scp://`*`user-login@server-fqdn`*`:`*`server-port`*`/`*`absolute-file-path`* (for example, `scp://`*`user@host1.mydomain.com`*`:30443/home/user/configs/apm`).

- Syslog server details. This is required if you are planning to export APM logs to an external syslog collector.

  > **NOTE**: If BBE Event Collection and Visualization is detected running on the target cluster, the address and port values of the ECAV deployment will be suggested as the default.

  - Syslog server address.

  - Sysylog server port number.

- Network load balancer details. This is required if you are planning to use a specific network load balancer pool and address for APMi.

  - Network load balancer pool name.

  - Network load balancer pool address.

- APMi Details:

  - Port (default is 20557)

  - TLS details. You will need one of the following:

    - None (insecure)

    - Either the key and certificate files or the name of the Kubernetes secret that is present in the jnpr-apm namespace that contains the key and certificate information.

- Service account name—The name of the Kubernetes service account used to bind certain operational privileges to the mgmt microservice. If a service account name is not provided, APM creates a service account named apm-svca during rollout.

- DBSync service type—The `apm multicluster status` APM utility command collects the state to display from the DBSync microservice through a Kubernetes service. By default, a node port service is created for this purpose. If you select `LoadBalancer`, you are prompted for an external IP address and a MetalLB pool is created containing the supplied external IP address. The `LoadBalancer` service created at rollout is assigned the external IP address from the newly created MetalLB pool.

- Number of worker processes for the provman microservice (default is 3).

Install the APM Application (Multiple Geography Setup)

1. Create the jnpr-apm namespace/project in the management context.

```
kubectl create namespace jnpr-apm --context management-context-name
```

   - `context` *`management-context-name`*—The context name for the management cluster. Be aware that this context is not the same as the Karmada context name that is associated with the management cluster.

2. Create the karmada-kconf secret. Create the secret on the management cluster in the APM namespace using the management cluster's kubeconfig.

NOTE: The karmada-kconf secret contains the kubeconfig that is used by the observer to monitor the status of the CPi. If the secret is not created, the observer (and APM) will not operate correctly.

```
kubectl create secret generic --context management-context-name -n jnpr-apm   karmada-kconf
--from-file=config=karmada-secret-file
```

   - `context` *`management-context-name`*—The context name for the management cluster. Be aware that this context is not the same as the Karmada context name that is associated with the management cluster.

   - *`karmada-secret-file`*—The management cluster's kubeconfig file.

3. Verify that the secret was created.

```
kubectl describe secrets --context mgmt -n jnpr-apm karmada-kconf
Name:         karmada-kconf
Namespace:    jnpr-apm
Labels:       <none>
Annotations:  <none>

Type:  Opaque

Data
====
config:  6426 bytes
```

4. Download the APM software package from Juniper Networks software download page to the jump host.

   APM is available as a compressed TAR (**.tgz**) file. The filename includes the release number as part of the name. The release number has the format: <Major>.<Minor>.<Maintenance>

   - *major* is the main release number of the product.
   - *minor* is the minor release number of the product.
   - *maintainance* is the revision number.

5. Unpack the APM TAR (**.tgz**) file on the jump host by entering:

```
$ tar -zxvf apm-m.m.m.tgz
apm/
apm/apm_loader
apm/images/
apm/images/apm_containerImages.tar.gz
apm/charts/
apm/charts/provman/
apm/charts/provman/templates/
apm/charts/provman/templates/apmProv-man.yaml
apm/charts/provman/templates/apmApmiSvc.yaml
apm/charts/provman/questions.yaml
apm/charts/provman/Chart.yaml
apm/charts/provman/containers.yaml
apm/charts/provman/values.yaml
apm/charts/provman/.helmignore
apm/charts/entman/
```

```
apm/charts/entman/templates/
apm/charts/entman/templates/apmEnt-man.yaml
.
.
.
```

6. Run the loader script after you unpack the TAR file.

```
$ sudo apm/apm_loader
Creating apm group... done.
Loading files... done.
Updating latest link... done
Setting up utility script... done.
Updating wrapper... done
Successfully loaded: 3.5.0
```

7. Use the `sudo -E apm link` command to link to the cluster. In preparation for running `setup`, the `link` command takes the list of workload cluster contexts and an observer context and associates them to the loaded APM software package.

```
$ sudo -E apm link --context karmada-context-name --version 3.5.0 --workload-contexts
workload-1-context-name workload-2-context-name --observer-context management-context-name
Linking karmada-context-name to 3.5.0...
  Adding workload-1-context-name to karmada-context-name workload context list... done.
  Adding workload-2-context-name to karmada-context-name workload context list... done.
done.
Linking BBE Observer to Address Pool Manager...
Linking management-context-name to 3.5.0 ... done.
done.
Linking complete, please run apm setup.

The following releases can be cleaned...
```

- `context` *`karmada-context-name`*—The context name of the Karmada context that is created when Karmada is installed on the management cluster.

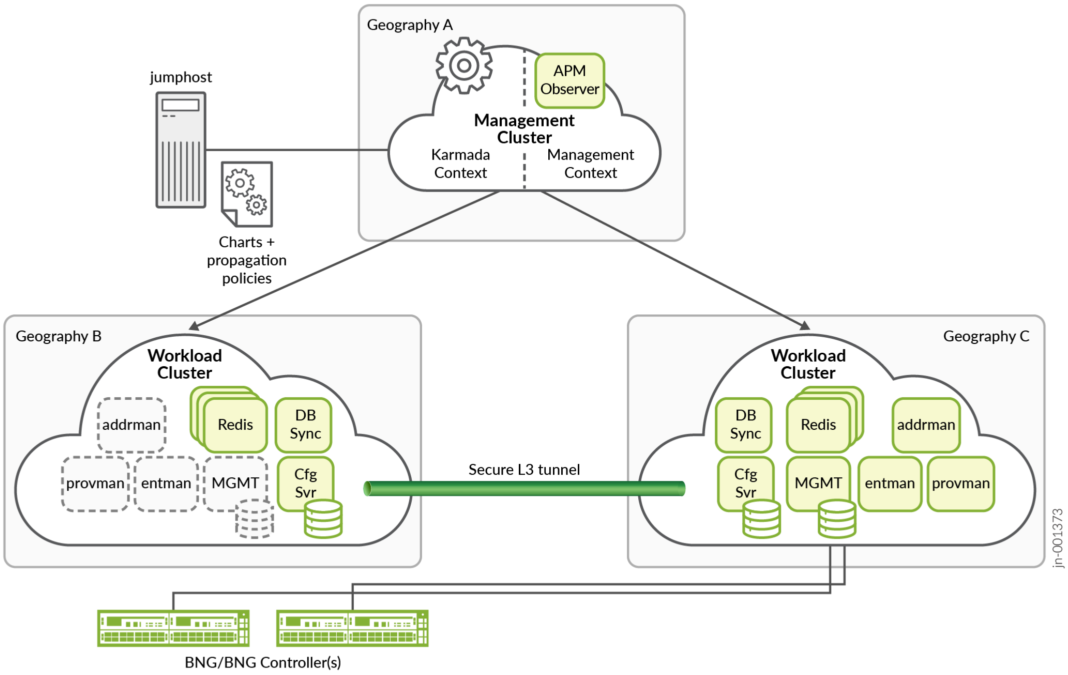
- `worload-contexts` *`workload-1-context-name workload-2-context-name`*—The two workload context names.

- `observer-context` *`management-context-name`*—The context name for the management cluster. Be aware that this context is not the same as the Karmada context name that is associated with the management cluster.

- `version` *`software-release`*—The APM software version, as displayed from the `apm_loader` output.

> **NOTE**: During installation, Karmada is installed on the management cluster and Karmada creates its own context on the management cluster. The Karmada context is used to target any operations involving the workload clusters. The management cluster also has its own context that is used for running noncritical centralized workloads (for example, BBE Event Collection and Visualization). You use this management cluster context for the *observer-context*.
>
> Figure 1 on page 38 shows where the different contexts are located in a multiple cluster setup.

Figure 1: Multiple Cluster Contexts



8. When using a RHOCP cluster, you can interact with it after authenticating the OpenShift cluster and the three RHOCP clusters (management and two workload clusters) using the OpenShift CLI.

   For an example of the command to run, see the following:

```
$ oc login -u regress
Console URL: https://api.<context>.<domain>:6443/console
Authentication required for https://api.<context>.<domain>:6443 (openshift)
```

```
Username: <clusterUser>
Password:
Login successful.

You have access to 80 projects, the list has been suppressed. You can list all projects
with 'oc projects'

Using project "default".
```

9. In order to push the APM container images, you must authenticate with the registry on each cluster in the multiple cluster setup. Authenticate with the registry by issuing a docker login as the **system user** (the **system user** entered in the BBE Cloudsetup configuration file) to the cluster's registry transport address (the FQDN supplied as the system address in the BBE Cloudsetup configuration file).

   For an example of the command to run, see the following:

```
 docker login default-route-openshift-image-registry.apps.<context><domain> -u $(oc whomai)
-p $(oc w
hoami -t)
WARNING! Using --password via the CLI is insecure. Use --password-stdin.

WARNING! Your credentials are stored unencrypted in '/home/regress/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/

Login Succeeded
```

10. Run `setup` to configure your installation. The `setup` command does the following:

    - Collects the following information about the cluster environment:

        - Names of storage classes or persistent volumes

        - Location of a container registry

        - Container and pod name of registry

        - TLS keys

        - Information relevant to supporting APM features.

    - Initializes the APM configuration.

    The following prompts will appear during the setup process:

    - Enter the following information for the management cluster:

  - External registry address and port number for each cluster. Press the **Enter** key after entering the information for each cluster.

  - Registry address and port number, for the observer (management cluster) to pull from.

  - Karmada kubeconfig secret name

  - Enable TLS (default is False)

  - TLS secret name

- Enter the following for the primary workload cluster. After entering the information, press the **Enter** key to enter the information for the backup workload cluster:

  - Name of cluster

  - Cluster registry address and port number

- Enter the following for the backup workload cluster:

  - Name of cluster

  - Cluster registry address and port number

```
$ sudo -E apm setup --context karmada-context-name [--help] [--log info] [--no color] [--
update] [--secrets] [--verbose] [--config file-name] [--template file-name] [--mandatory]
[--optional]
```

NOTE: To set up CLI access through SSH in a multiple geography deployment, you must use a template file (use the `template` option in the `setup` command). This enables you to configure the two workload cluster addresses.

To configure SSH, add the following information (YAML formatted) for each workload cluster to the template file that you provide during the setup process:

```
ssh:
- ip: ip-address            #Where the IP address is the workload cluster 1 cluster vip address
                              for the NodePort type or a reachable IP address for the LoadBalancer type
  karmada_workload_cluster: cluster-name   #For example, the name of workload cluster 1
  port: 'available-port-value'             #Port value should be in single quotes
  name: service-name                       #A user-defined name. For example, apm-ssh-workload-1
  type: NodePort                           #The type of service, either NodePort or LoadBalancer
- ip: ip-address            #Where the IP address is the workload cluster 2 cluster vip address
                              for the NodePort type or a reachable IP address for the LoadBalancer type
  karmada_workload_cluster: cluster-name   #For example, the name of workload cluster 2
  port: 'available-port-value'           #Port value should be in single quotes
  name: service-name                     #A user-defined name. For example, apm-ssh-workload-2
  type: NodePort                         #The type of service, either NodePort or LoadBalancer
```

For more information regarding the SSH configuration in the template file, see the following:

- *`ip-address`*—The IP address you use to manage the cluster from the jump host.

- *`cluster-name`*—The name of the workload cluster as it appears in the output of the `kubectl get clusters` command.

- *`available-port-value`*—If the **NodePort** option is entered in the `type` field, the port value must be a TCP port that is not used on any of the workload cluster's nodes. A best practice is to avoid ports that are already in use (like the often used SSH port 22), but below the ephemeral port range (port 49152 and higher). This avoids possible port contention with the node itself.

- *`service-name`*—The name you want the created service to use. A best practice is to include the application name, the purpose, and the workload cluster in the name (for example, apm-ssh-workload1).

The options that you can use with the `setup` command are listed in the following:

- `context` *`karmada-context-name`*—The context name of the Karmada context that is created when Karmada is installed on the management cluster.

- `h, help`—Shows the help message and exit.

- `l, log [error, warning, info, debug]`—Adjusts the log level.

- `no-color`—Prints messages without colors.

- `update`—You will only be prompted for missing values during setup.

- `secrets`—Updates the keys, certificates, and secrets used by APM.

- `verbose`—Provides a detailed description before each prompted question.

- `config` *file-name*—The name of the initial configuration file that you want APM to use during startup.

- `template` *`file-name`*—A YAML formatted file that contains a subset of the configuration file that is created during setup. The values that are entered in the template file are used automatically by the setup process. When you use the `template` option, you are not required to manually enter the information contained in the template file during the setup process. You should only use the `template` option when using Red Hat OpenShift Container Platform to create the cluster or when creating a multiple geography cluster. Table 3 on page 19 describes the information that you need to enter into the `template` configuration file.

- `mandatory`—Only asks required questions during setup.
- `optional`—Only asks questions that are not required during setup.

11. Verify the APM installation by running the `apm version` command.

```
$ apm version --context karmada-context-name --detail
Address Pool Manager versions:
  Microservice            Release            (version)
  apm:                    3.5.0
  addrman:                3.5.0
  configserver:           3.5.0
  dbsync:                 3.5.0
  entman:                 3.5.0
  mgmt:                   3.5.0              (24.4R1.1)
  provman:                3.5.0
  redis-swwf-fedora-wl1:  3.5.0              (6.2.16-debian-12-r3)
  redis-swwf-fedora-wl2:  3.5.0              (6.2.16-debian-12-r3)
Available releases for Address Pool Manager:
  3.5.0 (used by 1 installations)
  contexts:    karmada-context-name
  components:  apm             redis  mgmt  addrman  entman  provman  configserver  dbsync
```

- `context` *`karmada-context-name`*—The context name of the Karmada context that is created when Karmada is installed on the management cluster.
- `detail`—Displays all available software versions.

12. Create the karmada-kubeconfig secret from the kubeconfig file for the Karmada context (for additional information, see "Prerequisites" on page 32). The secret is necessary for the APM observer microservice, which runs in the management context, to monitor the workload cluster scheduling events in order to calculate a generation number used by APM in the cluster switchover process.

    Use the following command to create the karmada-kubeconfig secret (name the secret **karmada-kconf**):

```
kubectl create secret generic --context -n jnpr-apm karmada-kconf --from-
file=config=kdSecretFile
```

Start APM in a Multiple Geography Setup

SUMMARY

Use this procedure to configure and to start APM in a multiple geography setup.

1. Enter `rollout` to start the APM installation. The APM utility allows you to roll out different software versions for all microservices that are part of your APM multiple geography setup. You need to use the `rollout` command with sudo as root. The `rollout` command also validates that all the values needed for the new release are present and loads the new release container images to the registry. Use `sudo -E apm rollout --context` *karmada-context-name* `--version` *software-release* `--service` *service-name* to start APM services. For example:

```
$ sudo -E apm rollout --context karmada-context-name
Validating registries... done.
Deploying BBE Observer on swwf-fedora-kd...
Load container images to registry... done.
Rollout BBE Observer... done.
done.
Deploying APM Service Account on swwf-fedora-wl1... done.
Deploying APM Service Account on swwf-fedora-wl2... done.
Load container images to registry...
Loading 3.5.0-2 addrman images to local cache... done.
Loading 3.5.0-2 entman images to local cache... done.
Loading 3.5.0-2 provman images to local cache... done.
Loading 3.5.0-2 configserver images to local cache... done.
Loading 3.5.0-2 dbsync images to local cache... done.
Loading 3.5.0-2 redis-swwf-fedora-wl1 images to local cache... done.
Loading 3.5.0-2 redis-swwf-fedora-wl2 images to local cache... done.
Pushing 3.5.0-2 addrman images to registry... done.
Pushing 3.5.0-2 entman images to registry... done.
Pushing 3.5.0-2 provman images to registry... done.
Pushing 3.5.0-2 configserver images to registry... done.
Pushing 3.5.0-2 dbsync images to registry... done.
Pushing 3.5.0-2 redis-swwf-fedora-wl1 images to registry... done.
Pushing 3.5.0-2 redis-swwf-fedora-wl2 images to registry... done.
done.
```

```
Rollout Address Pool Manager... done.
```

- *karmada-context-name*—The context name of the Karmada context that is created when Karmada is installed on the management cluster.

- `service` *service-name*—The microservice name to rollout.

- `version` *software-release*—The software release to rollout (defaults to the release that links to the cluster).

> **NOTE**: On the, first rollout `service` is not required. The `service` is used with the `version` to rollout (upgrade) specific versions of specific services.

> **NOTE**: By default, APM starts with the values that you provided during setup. Unless the configuration was saved, the initial configuration is what was provided during setup. All other persistent states (logs, database keys, and so on) are cleared.

2. Enter `apm status --context` *karmada-context-name* `--detail` to verify that the APM CUPS Controller services are up and running. For example:

```
$ apm status --context karmada-context-name --detail

MICROSERVICE           WORKLOAD CLUSTER  POD                                                              STATE   RESTARTS  UPTIME         NODE

addrman                swwf-fedora-wl1   jnpr-apm-addrman-5757ffdcb5-sg5hx                                Running  0        0:02:55.401917  swwf-fedora-wl1-n-1.englab.juniper.net

configserver           swwf-fedora-wl1   jnpr-apm-configserver-swwf-fedora-wl1-65b5866785-xdtzq           Running  0        0:03:18.401953  swwf-fedora-wl1-n-1.englab.juniper.net

configserver           swwf-fedora-wl2   jnpr-apm-configserver-swwf-fedora-wl2-6889796559-5ppwp           Running  0        0:03:22.401973  swwf-fedora-wl2-n-1.englab.juniper.net

dbsync                 swwf-fedora-wl1   jnpr-apm-dbsync-swwf-fedora-wl1-574cc8f54f-kj9gh                 Running  0        0:02:55.401991  swwf-fedora-wl1-n-1.englab.juniper.net

dbsync                 swwf-fedora-wl2   jnpr-apm-dbsync-swwf-fedora-wl2-65db78c55c-kpdq8                 Running  0        0:02:57.402007  swwf-fedora-wl2-n-1.englab.juniper.net

entman                 swwf-fedora-wl1   jnpr-apm-entman-7c84b4b96b-cxrpl                                 Running  0        0:02:57.402023  swwf-fedora-wl1-n-1.englab.juniper.net

mgmt                   swwf-fedora-wl1   jnpr-apm-mgmt-848b8b894f-nq9nt                                   Running  0        0:02:54.402038  swwf-fedora-wl1-n-1.englab.juniper.net

provman                swwf-fedora-wl1   jnpr-apm-provman-57568b7b46-lrbzj                                Running  0        0:02:40.402053  swwf-fedora-wl1-n-1.englab.juniper.net

redis-swwf-fedora-wl1  swwf-fedora-wl1   jnpr-apm-redis-swwf-fedora-wl1-0                                 Running  0        0:03:17.402069  swwf-fedora-wl1-n-1.englab.juniper.net

redis-swwf-fedora-wl1  swwf-fedora-wl1   jnpr-apm-redis-swwf-fedora-wl1-1                                 Running  0        0:03:07.402084  swwf-fedora-wl1-n-1.englab.juniper.net

redis-swwf-fedora-wl1  swwf-fedora-wl1   jnpr-apm-redis-swwf-fedora-wl1-sentinels-0                       Running  0        0:03:07.402098  swwf-fedora-wl1-n-1.englab.juniper.net

redis-swwf-fedora-wl1  swwf-fedora-wl1   jnpr-apm-redis-swwf-fedora-wl1-sentinels-1                       Running  0        0:03:07.402113  swwf-fedora-wl1-n-1.englab.juniper.net

redis-swwf-fedora-wl1  swwf-fedora-wl1   jnpr-apm-redis-swwf-fedora-wl1-sentinels-2                       Running  0        0:03:07.402127  swwf-fedora-wl1-n-1.englab.juniper.net

redis-swwf-fedora-wl2  swwf-fedora-wl2   jnpr-apm-redis-swwf-fedora-wl2-0                                 Running  0        0:03:18.402142  swwf-fedora-wl2-n-1.englab.juniper.net

redis-swwf-fedora-wl2  swwf-fedora-wl2   jnpr-apm-redis-swwf-fedora-wl2-1                                 Running  0        0:03:08.402157  swwf-fedora-wl2-n-1.englab.juniper.net

redis-swwf-fedora-wl2  swwf-fedora-wl2   jnpr-apm-redis-swwf-fedora-wl2-sentinels-0                       Running  0        0:03:08.402171  swwf-fedora-wl2-n-1.englab.juniper.net

redis-swwf-fedora-wl2  swwf-fedora-wl2   jnpr-apm-redis-swwf-fedora-wl2-sentinels-1                       Running  0        0:03:09.402185  swwf-fedora-wl2-n-1.englab.juniper.net

redis-swwf-fedora-wl2  swwf-fedora-wl2   jnpr-apm-redis-swwf-fedora-wl2-sentinels-2                       Running  0        0:03:08.402200  swwf-fedora-wl2-n-1.englab.juniper.net
```

```
bbe-observer           swwf-fedora-kd    address-pool-manager-observer-controller-manager-559f5f4c9gqxpq  Running  0        0:04:12.402214  swwf-fedora-kd-n-1.englab.juniper.net
bbe-observer           swwf-fedora-kd    address-pool-manager-observer-controller-manager-559f5f4c9hw9s8  Running  0        0:04:12.402229  swwf-fedora-kd-n-1.englab.juniper.net
bbe-observer           swwf-fedora-kd    address-pool-manager-observer-controller-manager-559f5f4c9wmq95  Running  0        0:04:12.402242  swwf-fedora-kd-n-1.englab.juniper.net
bbe-observer           swwf-fedora-kd    address-pool-manager-observer-grpc-server-7594fc9fcd-jqcsf       Running  0        0:04:12.402256  swwf-fedora-kd-n-1.englab.juniper.net
bbe-observer           swwf-fedora-kd    address-pool-manager-observer-grpc-server-7594fc9fcd-jxcpj       Running  0        0:04:11.402271  swwf-fedora-kd-n-1.englab.juniper.net
bbe-observer           swwf-fedora-kd    address-pool-manager-observer-grpc-server-7594fc9fcd-k72lm       Running  0        0:04:12.402284  swwf-fedora-kd-n-1.englab.juniper.net

Storage:
  swwf-fedora-wl1: Healthy
  swwf-fedora-wl2: Healthy
```

NOTE: Collect the logs for a service and contact the Juniper Networks Technical Assistance Center (JTAC) when either of the following occurs:

- The service is not running.
- The service’s up time compared with other services indicates that it has restarted.

How to Use Command Line Tools to Administer APM

SUMMARY

After you've installed the Address Pool Manager (APM) application, you can perform the following administration functions.

IN THIS SECTION

Address Pool Manager gives you two command line options for perform administrator tasks. You can either use the APM utility script (`apm`) or the Kubernetes Command Line tool to administer APM.

Access APM Utility Commands

SUMMARY

Use the APM utility commands to perform administration functions.

You can use the APM utility script (`apm`) to administer the application and to access the CLI that you use to configure the address management functions. The Juniper APM installation places the utility script in **/var/local/apm** and creates a symbolic link to the script in **/usr/local/bin/apm**.

You can use the `apm` utility script (which uses the Kubernetes command line tool and Helm commands) to do the following:

- Create and delete objects.
- Provide log access.
- Conduct interactive sessions with pod containers.
- Display the status of the APM objects.

Using the `apm` utility script simplifies many of your administrative duties. The script performs the tasks you need to manage APM, while masking the complexity of the `kubectl` command.

Table 5 on page 48 lists the commands that you can invoke with the `apm` utility script and describes the action that occurs. Many of the individual commands have options that you can specify.

NOTE: If you are using these commands in a multiple geographically located setup, for the `context` *`context-name`* option, you must enter the context name of the Karmada context that is associated with the management cluster.

Table 5: APM Utility Script Commands

Name	Action
`sudo -E apm clean --context` *`context-name`* `[--docker] [--release` *`software-release`*`] [--dry-run] [--uninstall]`	Clean up unneeded releases and Docker cache. To run this command, you need sudo root privileges. This command offers the following options: • `context` *`context-name`*—The Kubernetes context name. Enter the name of the context. • `log [error, warning, info, debug]`—Adjusts the log level. • `no-color`—Prints messages without colors. • `docker`—Only cleans the local Docker cache, all other files remain. • `release` *`software-release`*—Specify a release to clean or clean all possible releases. • `dry-run`—Identifies releases and docker images for removal and prints them to console. This command does not actually clean any releases or the Docker cache. • `uninstall`—Uninstalls all APM materials from the disk. The command does not affect the running application. • `clustr-repos`—Clean the cluster repository of the clusters that have been removed.
`apm cli --context` *`context-name`* `[-p|--pipe]`	Gives you access to the CLI that you can use to configure APM features. This command offers the following options: • `context` *`context-name`*—The Kubernetes context name. Enter the name of the context. • `pipe`—Allows you to pipe input into the command.

Table 5: APM Utility Script Commands *(Continued)*

Name	Action
`apm contexts [-o|--output json]`	Displays the available contexts for control with APM. This command offers the following options: - `contexts`—Lists the available contexts. - `output json`—Allows you to request the output in JSON format.
`sudo -E apm db --context` *`context-name`* `--service` *`redis-microservice-instance-name`*	Provides access to the Redis database CLI. To run this command, you need sudo root privileges. This command offers the following options: - `context` *`context-name`*—The Kubernetes context name. Enter the name of the context. **NOTE**: If you are using this command in a multiple geographically located setup, you must enter the context name of the Karmada context that is associated with the management cluster. - `service` *`redis-microservice-instance-name`*—Directs the command to the master database instance in the specified Redis microservice instance. **NOTE**: This option is required in a multiple geographically located setup.

Table 5: APM Utility Script Commands *(Continued)*

Name	Action
`apm db-info --context` *`context-name`* `--service` *`redis-microservice-instance-name`*	Displays current state of APM's database microservice including the current version, stateful set pods, and their roles. **NOTE**: For a multiple geographically located setup, the `apm db-info` displays the redis roles for workload clusters. This command offers the following options: • `context` *`context-name`*—The Kubernetes context name. Enter the name of the context. **NOTE**: If you are using this command in a multiple geographically located setup, you must enter the context name of the Karmada context that is associated with the management cluster. • `service` *`redis-microservice-instance-name`*—Displays database information for the specified Redis microservice. If the `service` argument is specified, the output also includes the database information for the Redis microservice instances in both workload clusters

Table 5: APM Utility Script Commands *(Continued)*

Name	Action
`sudo -E apm db-switchover --context` *`context-name`* `--services` *`redis-micro-service-instance-name`*	Forces the primary database pod to switchover to an eligible backup database pod. To run this command, you need sudo privileges. This command offers the following options: • `context` *`context-name`*—The Kubernetes context name. Enter the name of the context. **NOTE**: If you are using this command in a multiple geographically located setup, you must enter the context name of the Karmada context that is associated with the management cluster. • `services` *`redis-micro-service-instance-name`*—Directs the command to the master database instance in the specified Redis microservice instance. **NOTE**: `db-switchover` is a service disrupting event and you only use it with the upgrade procedure.
`apm ip --context` *`context-name`* `[-o| --output json] [--detail]`	Displays the IP addresses of every service with an external IP address. This command offers the following options: • `context` *`context-name`*—The Kubernetes context name. Enter the name of the context. **NOTE**: If you are using this command in a multiple geographically located setup, you must enter the context name of the Karmada context that is associated with the management cluster. • `output json`—Allows you to request the output in JSON format. • `detail`—Displays detailed IP information.

Table 5: APM Utility Script Commands *(Continued)*

Name	Action

Single geographically located setup use:
`sudo -E apm link [--version` *`software-release`*`] [--context` *`context-name`*`] [--log` *`info`*`] [--no-color] [--from-running]`
Multiple geographically located setup use:
`sudo -E apm link [--context` *`karmada-context-name`*`] [--log` *`info`*`] [--no-color][--workload-contexts` *`workload-1-context-name workload-2-context-name`*`] [--observer-context` *`observer-context-name`*`] [--version` *`software-release`*`] [--from-running]`
	Links a cluster to a specific software version. To run this command, you need sudo root privileges.
	This command offers the following options:
	• `version` *`software-release`*—Specify the software release to link to the cluster specific repository.
	• `context` *`context-name`*—The Kubernetes context name to link to the software release. Enter the name of the context.
	• `context` *`karmada-context-name`*—The context name of the Karmada context that is created when Karmada is installed on the management cluster. Used with a multiple geographically located setup only. Enter the name of the context.
	• `workload-contexts` *`workload-1-context-name workload-2-context-name`*—The two workload context names. Used with a multiple geographically located setup only. Enter the name of the context.
	• `l, log [error, warning, info, debug]`—Adjusts the log level.
	• `no-color`—Prints messages without colors.
	• `observer-context` *`management-context-name`*—The context name for the management cluster. Used with a multiple geographically located setup only. Enter the name of the context.
	• `from-running`—Attempts to match the software releases of the running Address Pool Manager.
	The `from-running` option is used to recover the version and user settings from a running application deployment. This can help you recover a failed jump host or for synchronizing a network of jump hosts.

Table 5: APM Utility Script Commands *(Continued)*

Name	Action
`apm logs --logset` *`logset-type`* `--services` *`services-names`* `--context` *`context-name`* `--log` *`info`* `--follow --previous --nocolor`	Displays logs for APM microservices. If you run the command without the `services` option, logs for the `addrman`, `entman`, and `provman` microservices are displayed. This command offers the following options: • `context` *`context-name`*—The Kubernetes context name. Enter the name of the context. • `logset` *`logset-type`*—Used to indicate what type of microservice for which you want the logs displayed. This command has the following options: • `apm` —To display the `addrman`, `entman`, and `provman` microservices logs. If the `logset` option is not configured, the `apm`option is the default operation and you can only see the logs for the above microservices. • `apm-infra` —To display the `configserver`, `dbsync`, `mgmt` and `redis` instances microservices logs. • `services` *`services-names`*—List the specific microservices for which you want the logs displayed. Logs are displayed chronologically. • `l, log [error, warning, info, debug]`—Adjusts the output log level. • `nocolor`—Disables colorized output. • `f, follow`—To follow the logs for all microservices. • `p, previous`—To view logs from previous container instances. **NOTE**: If you are using this command in a multiple geographically located setup, you must enter the context name of the Karmada context that is

Table 5: APM Utility Script Commands *(Continued)*

Name	Action
	associated with the management cluster. Also, as there are multiple instances of `redis` in a multiple geographical deployment, you must specify the `redis` instance. For example: `$ apm logs --context karmada-context --services` *`redis-workload1`* `--logset apm-infra`
`apm multi-cluster dbsync-stats --context` *`karmada-context-name`*	Displays the status of the database synchronization between the workload clusters. This command offers the following option: • `context` *`karmada-context-name`*—The context name of the Karmada context that is associated with the management cluster. Enter the name of the context. **NOTE**: This command is only supported in a multiple geographically located setup.
`apm multi-cluster status --context` *`karmada-context-name`*	Displays the status of the workload clusters from the application's perspective. This command offers the following option: • `context` *`karmada-context-name`*—The context name of the Karmada context that is associated with the management cluster. Enter the name of the context. **NOTE**: This command is only supported in a multiple geographically located setup.

Table 5: APM Utility Script Commands *(Continued)*

Name	Action
`sudo -E apm multi-cluster switchover --context` *`karmada-context-name`* `[--force]`	Initiates a switchover to the other workload cluster. The APM microservices are rescheduled on the other workload cluster. To run this command, you need sudo root privileges. This command offers the following option: • `context` *`karmada-context-name`*—The context name of the Karmada context that is associated with the management cluster. Enter the name of the context. The output of this command indicates success or failure. If a failure occurs, the command stops with a failure status of **`Multi-cluster switchover failed (database is not synchronizing)`**. **NOTE**: This command is only supported in a multiple geographically located setup.
`sudo -E apm rename-context --context` *`context-name`* `--new-name` *`new-name`*	Renames a context. The command does not affect the APM that is currently running on the cluster. To run this command, you need sudo root privileges. This command offers the following options: • `context` *`context-name`*—The old Kubernetes context name to rename. Enter the name of the context. • `new-name` *`new-name`*—The new name of the Kubernetes context. Enter a new name.

Table 5: APM Utility Script Commands *(Continued)*

Name	Action
`sudo -E apm restart --context` *`context-name`* `[--force] [--wait]` *`microservice-name`*	Restart one or more specified services. To run this command, you need sudo root privileges. This command offers the following options: • `context` *`context-name`*—The Kubernetes context name on which to restart the service. Enter the name of the context. • `force`—Forcibly restart the micro-service without validating that it can be safely restarted. • `wait`—Wait for the new pod to fully come up. • *microservice-name*—Enter the microservice name to restart.
`sudo -E apm rollout --context` *`context-name`* `[--service` *`service name`* `--version` *`software-release`*`]`	Upgrade or start an APM service. To run this command, you need sudo root privileges. This command offers the following options: • `context` *`context-name`*—The Kubernetes context name on which to roll out the new software version. Enter the name of the context. **NOTE**: If you are using this command in a multiple geographically located setup, you must enter the context name of the Karmada context that is associated with the management cluster. • `sevice` *`service name`*—The microservice name to roll out. Enter the microservice's name. • `version` *`software-release`*—The software release to roll out. Enter the software release number.

Table 5: APM Utility Script Commands *(Continued)*

Name	Action
`sudo -E apm save-config -- context` *`context-name`*	Saves the current configuration of the Address Pool Manager to a file outside the pod. To run this command, you need sudo root privileges. This command offers the following options: • `context` *`context-name`*—The Kubernetes context name. Enter the name of the context. **NOTE**: If you are using this command in a multiple geographically located setup, you must enter the context name of the Karmada context that is associated with the management cluster.

Table 5: APM Utility Script Commands *(Continued)*

Name	Action
`sudo -E apm setup --context` *`context-name`* `[--help] [--log info] [--no color] [--bbecloudsetup] [--update] [--ssh` *`ip-address:port-number`*`] [--secrets] [--verbose] [--config` *`file-name`*`] [--template` *`file-name`*`] [--mandatory] [--optional]`	Set up the APM application as part of the installation process. To run this command, you need sudo root privileges. This command offers the following options: • `context` *`context-name`*—The Kubernetes context name on which to run startup. Enter the name of the context. **NOTE**: If you are using this command in a multiple geographically located setup, you must enter the context name of the Karmada context that is associated with the management cluster. • `h, help`—Shows the help message and exit. • `l, log [error, warning, info, debug]`—Adjusts the log level. • `no-color`—Prints messages without colors. • `bbecloudsetup`—Fills in operational parameters that align with a bbecloudsetup created cluster so that you do not have to interact with APM during the setup process (see the BBE Cloudsetup Installation Guide for cluster installation instructions). **NOTE**: Only use either the `bbecloudsetup` option or the `template` *`file-name`* option. Do not use both options. • `update`—You will only be prompted for missing values during setup. • `ssh` *`ip-address:port-number`*—Enables SSH towards the control plane instance. Enter the SSH IP address and port number on which the control plane instance is listening for SSH (when enabled in the configuration). The IP address can also be a DNS name.

Table 5: APM Utility Script Commands *(Continued)*

Name	Action
	• `secrets`—Updates the keys, certificates, and secrets used by APM. • `verbose`—Provides a detailed description before each prompted question. • `config` *`file-name`*—The initial configuration file that you want APM to use at startup. • `template` *`file-name`*—A YAML formatted file that contains a subset of the configuration file that is created during setup. The values that are entered in the template file are used automatically by the setup process. When you use the `template` option, you are not required to manually enter the information contained in the template file during the setup process. You should only use the `template` option when using Red Hat OpenShift Container Platform to create the cluster or when creating a multiple geographical cluster. Table 3 on page 19 describes the information that you need to enter into the `template` configuration file. **NOTE**: Only use either the `bbecloudsetup` option or the `template` *`file-name`* option. Do not use both options. • `mandatory`—Only asks required questions during setup. • `optional`—Only asks questions that are not required during setup.

Table 5: APM Utility Script Commands *(Continued)*

Name	Action
`sudo -E apm shell --context` *`context-name`* `[-p|--pipe]` *`microservice-name`*	Connects you to a running microservice. To run this command, you need sudo root privileges. This command offers the following options: • *microservice-name*—The name of the microservice that you want to connect to. • `context` *`context-name`*—The Kubernetes context name. Enter the name of the context. **NOTE**: If you are using this command in a multiple geographically located setup, you must enter the context name of the Karmada context that is associated with the management cluster. • `pipe`—Allows you to pipe input into the command.
`sudo -E apm start --context` *`context-name`*	Starts all APM services. To run this command, you need sudo root privileges. This command offers the following option: • `context` *`context-name`*—The Kubernetes context name on which to start APM. Enter the name of the context. **NOTE**: If you are using this command in a multiple geographically located setup, you must enter the context name of the Karmada context that is associated with the management cluster.

Table 5: APM Utility Script Commands *(Continued)*

Name	Action
`apm status --context` *`context-name`* `[-o|--output json] [--terse] [--detail]`	Display the current status of the APM services. This command offers the following options: • `context` *`context-name`*—The Kubernetes context name. Enter the name of the context. **NOTE**: If you are using this command in a multiple geographically located setup, you must enter the context name of the Karmada context that is associated with the management cluster. • `output`—Allows you to request the output in JSON format. • `terse`—Displays a summarized output of the health of the system. • `detail`—Displays information for each pod.
`sudo -E apm stop --context` *`context-name`* `[--now]`	Stops all APM services. To run this command, you need sudo root privileges. This command offers the following option: • `context` *`context-name`*—The Kubernetes context name on which to stop APM. Enter the name of the context. **NOTE**: If you are using this command in a multiple geographically located setup, you must enter the context name of the Karmada context that is associated with the management cluster. • `now`—Stops APM immediately, instead of waiting for the two minute delay.

Table 5: APM Utility Script Commands *(Continued)*

Name	Action
`apm storage --context` *`context-name`*	Provides the status of the storage drivers for APM. This command offers the following option: • `context` *`context-name`*—The Kubernetes context name. Enter the name of the context. **NOTE**: If you are using this command in a multiple geographically located setup, you must enter the context name of the Karmada context that is associated with the management cluster.
`sudo -E apm unlink --context` *`context-name`*	Unlink components associated with the context. To run this command, you need sudo root privileges. This command offers the following option: • `context` *`context-name`*—The Kubernetes context name. Enter the name of the context. **NOTE**: If you are using this command in a multiple geographically located setup, you must enter the context name of the Karmada context that is associated with the management cluster.
`apm version --context` *`context-name`*	Displays the version of every running microservice in the APM instance as well as the APM utility. It also lists all available APM software releases on the system. This command offers the following option: • `context` *`context-name`*—The Kubernetes context name. Enter the name of the context. **NOTE**: If you are using this command in a multiple geographically located setup, you must enter the context name of the Karmada context that is associated with the management cluster.

Use the following general syntax to issue a command:

- For a short option:

  ```
  $ apm command-name -option
  ```

- For a long option:

  ```
  $ apm command-name --option
  ```

To target a command at a particular cluster context, use the **context** option.

```
$ apm command-name --context context-name
```

To display a list of available commands with a brief description, use either the `h` or `help` option:

```
$ apm -h
```

```
$ apm --help
```

To display the options for a specific command:

```
$ apm command-name -h
```

To specify the `no-color` option to disable colored text output (used to distinguish logs from different microservices):

```
$ apm command-name --no-color
```

Upgrade APM to a New Version Using the APM installation Utility

Use this procedure to upgrade to a new version of APM which is installed on a similar cluster deployment (single geography or multiple geography). This procedure assumes APM is running on your system.

> **NOTE**: You cannot upgrade between a single geography and a multiple geography deployment.

Use this procedure to upgrade to a new version of APM which is installed on a cluster that was created by the BBE Cloudsetup utility or by Red Hat OpenShift Container Platform Console. This procedure assumes APM is running on your system.

1. Download the APM software package from the Juniper Networks software download page to the jump host.

   APM is available as a compressed tarball image (**.tgz**). The filename includes the release number as part of the name. The release number has the format: <Major>.<Minor>.<Maintenance>

   - *major* is the main release number of the product.
   - *minor* is the minor release number of the product.
   - *maintenance* is the revision number.

2. Unpack the APM tarball (**.tgz**) file on the jump host by entering:

```
$ tar -zxvf apm-m.m.m.tgz
apm/
apm/apm_loader
apm/images/
apm/images/apm_containerImages.tar.gz
apm/charts/
apm/charts/provman/
apm/charts/provman/templates/
apm/charts/provman/templates/apmProv-man.yaml
apm/charts/provman/templates/apmApmiSvc.yaml
apm/charts/provman/questions.yaml
apm/charts/provman/Chart.yaml
apm/charts/provman/containers.yaml
apm/charts/provman/values.yaml
apm/charts/provman/.helmignore
apm/charts/entman/
apm/charts/entman/templates/
apm/charts/entman/templates/apmEnt-man.yaml
.
.
.
```

3. Run the loader script after you unpack the tarball.

   ```
   $ sudo apm/apm_loader
   Creating apm group... done.
   Loading files... done.
   Updating latest link... done
   Setting up utility script... done.
   Updating wrapper... done
   Successfully loaded: 3.5.0
   ```

4. Link to the cluster by using the `link` command. The `link` command associates the loaded APM software package to the cluster in preparation for the setup. The commands that you enter for a single geography or multiple geography deployment are different. Run the following command that is appropriate for your deployment:

   - Single geography:

     ```
     $ sudo -E apm link --context context-name --version apm-version
     ```

   - Multiple geography:

     ```
     $ sudo -E apm link --context karmada-context-name --workload-contexts workload-1-context-name workload-2-context-name --observer-context management-context-name --version apm-version
     ```

   - *context-name*—The Kubernetes context name.

   - *apm-version*—The APM software version.

   - *karmada-context-name*—The context name of the Karmada context that is created when Karmada is installed on the management cluster.

   - *workload-1-context-name workload-2-context-name*—The two workload context names.

   - *management-context-name*—The context name for the management cluster. Be aware that this context is not the same as the Karmada context name that is associated with the management cluster.

5. If you are upgrading APM on a Red Hat OpenShift Container Platform cluster, log in with the OpenShift CLI and then proceed to the next step.

   If you are installing APM on a BBE Cloudsetup created cluster, proceed to the next step.

6. You must authenticate with each cluster's container registry in order to push the APM container images. For a single geography deployment. you authenticate with the single cluster's container

registry. For a multiple geography deployment, you authenticate with the container registries of the management, and both workload clusters. How you authenticate to the registry varies depending on if you are installing APM on a BBE Cloudsetup created cluster or on an Red Hat OpenShift Container Platform cluster (see the respective documentation for details).

7. Run the `setup` command to complete the setup of any additional environment values. The commands that you enter for a single geography or multiple geography deployment are different. Run the following command that is appropriate for your deployment:

   - Single geography:

     ```
     $ sudo -E apm setup --context context-name --update
     ```

   - Multiple geography:

     ```
     $ sudo -E apm setup --context karmada-context-name --update
     ```

   - *context-name*—The Kubernetes context name.

   - *karmada-context-name*—The context name of the Karmada context that is created when Karmada is installed on the management cluster.

   - **update** are the prompts for only missing values (primarily used after loading a new software release).

8. Display the running database to see which pod is the primary pod and to determine whether to upgrade the persistent state database. The commands that you enter for a single geography or multiple geography deployment are different. Run the following command that is appropriate for your deployment:

   - Single geography:

     ```
     apm db-info -context cluster-name
     ```

   - Multiple geography:

     ```
     apm db-info --context karmada-context-name --service redis-service-instance-name
     ```

9. Display the database version in the new package:

   ```
   apm version --compare 3.5.0 --context cluster-name
   components:
   ```

```
apm: 3.4.0 -> 3.5.0
addrman: 3.4.0 -> 3.5.0
entman: 3.4.0 -> 3.5.0
mgmt: 3.4.0 -> 3.5.0
provman: 3.4.0 -> 3.5.0
redis: 6.2.14 -> 6.2.16-debian-12-r3
```

NOTE: For example, the database version 6.2.16-debian-12-r3 is later than what is running (6.2.14), so you need to upgrade the database.

10. Initiate a database switchover if the `jnpr-apm-redis-0` is not the primary database. The commands that you enter for a single geography or multiple geography deployment are different. Run the following command that is appropriate for your deployment:

    - Single geography:

      ```
      sudo -E apm db-switchover --context context-name
      ```

    - Multiple geography:

      ```
      sudo -E apm db-switchover --context karmada-context-name --service redis-service-
      instance-name
      ```

11. Upgrade the microservices using the `rollout` command. Enter the commands in the following order for your deployment (single or multiple geography):

    NOTE: If the versions of the `observer`, `dbSync`, `configsvr`, or `db`(redis) microservices are not changing, you do not need to run the `rollout` command for them.

    - Single geography:

      ```
      sudo -E apm rollout --context context-name --version new version --service db
      sudo -E apm rollout --context context-name --version new version --service mgmt
      sudo -E apm rollout --context context-name --version new version --service addrman
      sudo -E apm rollout --context context-name --version new version --service entman
      sudo -E apm rollout --context context-name --version new version --service provman
      ```

- Multiple geography

```
sudo -E apm rollout --context karmada-context-name --version new version --service
observer
sudo -E apm rollout --context karmada-context-name --version new version --service
configsvr
sudo -E apm rollout --context karmada-context-name --version new version --service redis-
service-instance-workload2
sudo -E apm rollout --context karmada-context-name --version new version --service redis-
service-instance-workload1
sudo -E apm rollout --context karmada-context-name --version new version --service dbsync
sudo -E apm rollout --context karmada-context-name --version new version --service mgmt
sudo -E apm rollout --context karmada-context-name --version new version --service
addrman
sudo -E apm rollout --context karmada-context-name --version new version --service entman
sudo -E apm rollout --context karmada-context-name --version new version --service
provman
```

12. Verify that all microservices are running the new version of software:

```
$ apm version --context context-name --detail
Address_Pool_Manager versions:
  Microservice  Release            (version)
  apm:          3.5.0
  addrman:      3.5.0
  entman:       3.5.0
  mgmt:         3.5.0 (24.4R2)
  provman:      3.5.0
  redis:        3.5.0  (6.2.16-debian-12-r3)
```

Upgrade APM to a New Version Without Using the APM Utility

The instructions in this section describes the upgrade steps for installing APM on a preexisting single geography Kubernetes cluster of your choice. This process is a manual process and does not use the APM utility that comes with the APM installation package.

NOTE: This upgrade procedure is only for a single geographical deployment.

1. Download the APM software package from the Juniper Networks software download page to the jump host.

   APM is available as a compressed tarball image (**.tgz**). The filename includes the release number as part of the name. The release number has the format: <Major>.<Minor>.<Maintenance>

   - *major* is the main release number of the product.

   - *minor* is the minor release number of the product.

   - *maintainance* is the revision number.

2. Unpack the APM tarball (**.tgz**) file on the jump host by entering:

```
$ tar -zxvf apm-m.m.m.tgz
apm/
apm/apm_loader
apm/images/
apm/images/apm_containerImages.tar.gz
apm/charts/
apm/charts/provman/
apm/charts/provman/templates/
apm/charts/provman/templates/apmProv-man.yaml
apm/charts/provman/templates/apmApmiSvc.yaml
apm/charts/provman/questions.yaml
apm/charts/provman/Chart.yaml
apm/charts/provman/containers.yaml
apm/charts/provman/values.yaml
apm/charts/provman/.helmignore
apm/charts/entman/
apm/charts/entman/templates/
apm/charts/entman/templates/apmEnt-man.yaml
.
.
.
```

3. The container images needed by APM are stores in the **images** subdirectory. You must push the images to the registry where the scheduled application images will be pulled from. Depending on the

type of container registry being used the commands may be different. The following commands illustrate one method of pushing container images to the registry:

```
% docker image load -i ./images/apm_addr-man_containerImages.tar.gz
Loaded image: apm_addr-man:3.4.0
```

```
% docker tag apm_addr-man:3.4.0 <regHost>:<regPort>/apm_addr-man:3.4.0
```

```
% docker push <regHost>:<regPort>/apm_addr-man:3.4.0
```

4. To prepare APM for upgrade, create a new YAML configuration file for each microservice. (Table 4 on page 28 describes the fields in the microservice's configuration files.)

   NOTE: You may have create a single **values.yaml**, during your intial installation,that contains information for all the microservices. The single **values.yaml** is located under the umbrella chart in the **apm/apm/charts/address_pool_manager** folder. The procedures in this section only describe how to upgrade if created individual YAML configuration files for each microservice.

   Create a new **values.yaml** file for each of the microservices, by making a copy of the file and then saving the new file. Update each file according to your Kubernetes cluster's information.

   Following are the microservices and their **values.yaml** file location:

   - **redis** microservice—Located at **apm/apm/charts/redis**
   - **mgmt** microservice—Located at **apm/apm/charts/mgmt**
   - **addrman** microservice—Located at **apm/apm/charts/addrman**
   - **entman** microservice—Located at **apm/apm/charts/entman**
   - **provman** microservice—Located at **apm/apm/charts/provman**

5. Create the APM `ServiceAccount` object:

```
kubectl create serviceaccount -n jnpr-apm apm
```

6. Run the dependency update command:

```
helm dependency update --kube-context context-name --namespace jnpr-apm ./charts/redis/
```

7. After you have made all the desired changes to your new **values.yaml** files for each microservice, the microservices must be deployed with the new **values.yaml** files.

   Run the following commands:

```
helm upgrade --dependency-update --kube-context context-name --namespace jnpr-apm --create-namespace --atomic --install -f ./charts/redis/new-values.yaml redis ./charts/redis
```

```
helm upgrade --dependency-update --kube-context context-name --namespace jnpr-apm --create-namespace --atomic --install -f ./charts/mgmt/new-values.yaml mgmt ./charts/mgmt
```

```
helm upgrade --dependency-update --kube-context context-name --namespace jnpr-apm --create-namespace --atomic --install -f ./charts/addrman/new-values.yaml addrman ./charts/addrman
```

```
helm upgrade --dependency-update --kube-context context-name --namespace jnpr-apm --create-namespace --atomic --install -f ./charts/entman/new-values.yaml entman ./charts/entman
```

```
helm upgrade --dependency-update --kube-context context-name --namespace jnpr-apm --create-namespace --atomic --install -f ./charts/provman/new-values.yaml provman ./charts/provman
```

8. Verify the APM installation by running the Kubernetes Command Line Tool command `kubectl get pods` and verify the APM pods are running.

```
$ kubectl get pods -n jnpr-apm -o wide
NAME                              READY STATUS  RESTARTS AGE   IP          NODE                        NOMINATED NODE READINESS GATES
jnpr-apm-addrman-7cff87b557-gp8s7 1/1   Running  0       124m  10.42.2.15  jib.englab.juniper.net      <none>         <none>
jnpr-apm-entman-67d9bf9498-bx8jj  1/1   Running  0       124m  10.42.1.141 keel.englab.juniper.net     <none>         <none>
jnpr-apm-mgmt-6c76cc8dd7-pmlpv    1/1   Running  0       124m  10.42.0.20  binnacle.englab.juniper.net <none>         <none>
jnpr-apm-provman-75bc8d465d-czcfm 1/1   Running  0       123m  10.42.0.22  binnacle.englab.juniper.net <none>         <none>
jnpr-apm-redis-0                  1/1   Running  0       124m  10.42.2.16  jib.englab.juniper.net      <none>         <none>
jnpr-apm-redis-1                  1/1   Running  0       123m  10.42.0.21  binnacle.englab.juniper.net <none>         <none>
jnpr-apm-redis-sentinels-0        1/1   Running  0       124m  10.42.0.19  binnacle.englab.juniper.net <none>         <none>
```

```
jnpr-apm-redis-sentinels-1        1/1    Running   0       124m   10.42.1.142 keel.englab.juniper.net      <none>        <none>

jnpr-apm-redis-sentinels-2        1/1    Running   0       124m   10.42.2.17  jib.englab.juniper.net       <none>        <none>
```

9. Verify that the services are present. Run the Kubernetes Command Line Tool command `kubectl get services`.

```
$ kubectl get services -n jnpr-apm
NAME                              TYPE         CLUSTER-IP     EXTERNAL-IP     PORT(S)
AGE
jnpr-apm-mgmt-svc                ClusterIP    10.43.131.131  <none>          8066/TCP
125m
jnpr-apm-redis-sentinels-0-svc  ClusterIP    10.43.142.53   <none>          7381/TCP
125m
jnpr-apm-redis-sentinels-1-svc  ClusterIP    10.43.109.206  <none>          7381/TCP
125m
jnpr-apm-redis-sentinels-2-svc  ClusterIP    10.43.104.100  <none>          7381/TCP
125m
jnpr-apm-redis-svc               ClusterIP    10.43.6.207    <none>          7380/TCP
125m
provman-apmi                     LoadBalancer 10.43.221.12   198.19.224.212  20557:32553/TCP
125m
```

Start or Stop APM Services Using the APM Utility

Use the `apm` utility script to start or stop all APM services. The services start in order of dependency. Essential services (db and mgmt) start first, followed by the other services. The services stop in reverse order of dependency.

- To start all APM services:

```
$ sudo -E apm start --context context-name
```

NOTE: We recommend that you use the `sudo -E apm start --services` option to start individual services or a set of services only for troubleshooting. Use under the guidance of a Juniper Networks support representative. Use with caution as this command is like rebooting to factory-default.

NOTE: APM starts from it's initial settings when you execute the `apm setup` command. Any persistent state is lost when the `apm stop` command is executed. The current configuration can be saved using the `apm save-config` command. The saved configuration is the configuration that is used the next time APM is started.

- To stop all APM services:

```
$ sudo -E apm stop --context context-name
WARNING Shutting down your Address Pool Manager will reset it to factory defaults and you
will lose all state on the system.

Shutdown will begin in 2 minutes. Please use CTRL+C to cancel.
```

Restart APM Services Using the Kubernetes Command Line Tool

Use the `kubectl delete pods` Kubernetes command to restart APM services. For example:

```
$ kubectl delete pod -n jnpr-apm jnpr-apm-addrman-758cc8885
pod "jnpr-apm-addrman-7585cc8885" deleted
```

NOTE: To determine the pod name, you can use the `kubectl get pods -n jnpr-apm` Kubernetes command (see "Check the Status of APM Services Using the Kubernetes Command Line Tool" on page 79).

Setup Secrets Using the APM Utility

You can setup secrets during setup or run the `sudo -E apm setup --context` *`context-name`* `--secrets` to setup secrets or update them.

```
$ sudo -E apm setup --context context-name --secrets
APMi Secret Name (deployed: ) >
APMi certificate (default: ) > ./apm.crt
Copied /home/user/apm.crt to /var/local/apm/e476597324/secrets/apmi with 600 permissions
APMi private key (default: ) > ./apm.key
Copied /home/user/apm.keyto /var/local/apm/e476597324/secrets/apmi with 600 permissions
APMi root certificate (default: ) > ./rootCA.crt
Copied /home/user/rootCA.crt to /var/local/apm/e476597324/secrets/apmi with 600 permissions
```

NOTE: If you enter a value for the secret name, you will not be asked for the key or certification files.

Display Database Information Using the APM Utility

The `apm db-info` command displays current state of APM's database microservice including the current version, stateful set pods, and their roles.

NOTE: For a multiple geographically located setup, the `apm db-info` displays the redis roles for workload clusters.

Single Geographically Located Setup

```
$ apm db-info  [--context context-name] [-o|--output json]
Version: 6.2.16
Primary: jnpr-apm-redis-0
Backup(s): jnpr-apm-redis-1
```

Multiple Geographically Located Setup

```
$ apm db-info --context wf-mg-jh-flair
redis-workload-1:
```

```
version: 6.2.16-debian-12-r3
Primary: jnpr-apm-redis-workload-1-0
Backup(s): jnpr-apm-redis-workload-1-1
redis-workload-2:
version: 6.2.16-debian-12-r3
Primary: jnpr-apm-redis-workload-2-0
Backup(s): jnpr-apm-redis-workload-2-1
```

Display the Running Database Using the Kubernetes Command Line Tool

Use the `kubectl exec` Kubernetes command to display the running database to see which pod is the primary pod and to determine whether to upgrade the persistent state database. You should run the `kubectl exec` on both the primary and secondary redis server instances. For example:

```
$ kubectl exec -i -n jnpr-apm jnpr-apm-redis-0 -- redis-cli -p 7380 role
master
174738135
10.42.1.152
7380
174738135$
```

```
$ kubectl exec -i -n jnpr-apm jnpr-apm-redis-1 -- redis-cli -p 7380 role
slave
10.42.2.22
7380
connected
174740637
```

Perform a Database Switchover Using the Kubernetes Command Line Tool

To force the persistent state database primary pod to switchover to an eligible backup pod, perform the following:

1. Run the `kubectl get pods` Kubernetes command to determine the name of the sentinel pods.

```
$ kubectl get pods -n jnpr-apm
NAME                              READY   STATUS    RESTARTS   AGE
jnpr-apm-entman-5dcf659676-4mq6g  1/1     Running   0          5d22h
jnpr-apm-mgmt-6d7c4f47dc-2v8ss    1/1     Running   0          5d22h
jnpr-apm-provman-65c66bc689-bvhb6 1/1     Running   0          5d22h
jnpr-apm-redis-0                  1/1     Running   0          5d22h
jnpr-apm-redis-1                  1/1     Running   0          5d22h
jnpr-apm-redis-sentinels-0        1/1     Running   0          5d22h
jnpr-apm-redis-sentinels-1        1/1     Running   0          5d22h
jnpr-apm-redis-sentinels-2        1/1     Running   0          5d22h
```

2. Pick any of the three sentinel pods to use with the `kubectl exec` command. The following example uses the **jnpr-apm-redis-sentinels-0** sentinel pod. After running following commands, the roles of the redis instances are reveresed (redis-0 is the secondary and redis-1 is the primary).

```
$ kubectl exec -i -n jnpr-apm jnpr-apm-redis-sentinels-0 -c redis-sentinel -- redis-cli -p
7381 sentinel failover jnpr-apm-redis-masters
OK
```

```
$ kubectl exec -i -n jnpr-apm jnpr-apm-redis-0 -- redis-cli -p 7380
role slave
```

Check the Status of APM Services Using the APM Utility

Use the `apm status` utility script to check the status of each APM service (functional component) listed in Table 6 on page 77. The status shows whether a service is running, has exited, or has not started. It also displays the service name on the Kubernetes pod. You can compare uptime for the services to quickly see whether any service has been restarted.

> **NOTE**: You use the `apm status` utility script to check the status of each APM service for a multiple geographic, mutiple cluster setup.

Table 6: Services Displayed with the Status Command

Microservice	Pod Prefix
addrman—Address manager.	jnpr-apm-addrman
mgmt—CLI management.	jnpr-apm-mgmt
redis (Remote Dictionary Server)—consists of a set of pods which provide the persistent database.	jnpr-apm-redis
entman—Entity manager.	jnpr-apm-entman
provman—Provisioning manager.	jnpr-apm-provman
dbSync—Database synchronization.	jnpr-apm-dbSync
cfgServer—Configuration file replication.	jnpr-apm-cfgServer
bbe-observer—Observes multiple cluster scheduling events for generation number calculation.	jnpr-apm-bbe-observer

To check the status:

1. Display the service status.

   ```
   $ apm status --context contextName [-o|--output json] [--detail]
   ```

2. (Optional) Render the version information in JavaScript Object Notation (JSON) format, which is useful for scripting interfaces.

   ```
   $ apm status [-o|--output json]
   ```

For example:

```
$ apm status --context context-name --detail

MICROSERVICE          WORKLOAD        CLUSTER POD                                                       STATE   RESTARTS  UPTIME         NODE
```

```
addrman               swwf-fedora-wl1 jnpr-apm-addrman-d5f946495-pjv2t                                Running 0        0:00:42.643899 swwf-fedora-wl1-n-1.englab.juniper.net

configserver          swwf-fedora-wl1 jnpr-apm-configserver-swwf-fedora-wl1-65b5866785-k52tg          Running 0        0:01:03.643932 swwf-fedora-wl1-n-1.englab.juniper.net

configserver          swwf-fedora-wl2 jnpr-apm-configserver-swwf-fedora-wl2-6889796559-vwwr2          Running 0        0:01:05.643950 swwf-fedora-wl2-n-1.englab.juniper.net

dbsync                swwf-fedora-wl1 jnpr-apm-dbsync-swwf-fedora-wl1-669c58cc7b-q9h9g                Running 0        0:00:42.643965 swwf-fedora-wl1-n-1.englab.juniper.net

dbsync                swwf-fedora-wl2 jnpr-apm-dbsync-swwf-fedora-wl2-745cd74bf6-rkbbt                Running 0        0:00:41.643980 swwf-fedora-wl2-n-1.englab.juniper.net

entman                swwf-fedora-wl1 jnpr-apm-entman-fcc7cb9b7-fd9vl                                 Running 0        0:00:42.643994 swwf-fedora-wl1-n-1.englab.juniper.net

mgmt                  swwf-fedora-wl1 jnpr-apm-mgmt-5d6695dcfd-zfl92                                  Running 0        0:00:44.644007 swwf-fedora-wl1-n-1.englab.juniper.net

provman               swwf-fedora-wl1 jnpr-apm-provman-5cd4f48ccc-gvtb8                               Running 0        0:00:26.644021 swwf-fedora-wl1-n-1.englab.juniper.net

redis-swwf-fedora-wl1 swwf-fedora-wl1 jnpr-apm-redis-swwf-fedora-wl1-0                                Running 0        0:01:04.644034 swwf-fedora-wl1-n-1.englab.juniper.net

redis-swwf-fedora-wl1 swwf-fedora-wl1 jnpr-apm-redis-swwf-fedora-wl1-1                                Running 0        0:00:53.644048 swwf-fedora-wl1-n-1.englab.juniper.net

redis-swwf-fedora-wl1 swwf-fedora-wl1 jnpr-apm-redis-swwf-fedora-wl1-sentinels-0                      Running 0        0:00:53.644062 swwf-fedora-wl1-n-1.englab.juniper.net

redis-swwf-fedora-wl1 swwf-fedora-wl1 jnpr-apm-redis-swwf-fedora-wl1-sentinels-1                      Running 0        0:00:53.644092 swwf-fedora-wl1-n-1.englab.juniper.net

redis-swwf-fedora-wl1 swwf-fedora-wl1 jnpr-apm-redis-swwf-fedora-wl1-sentinels-2                      Running 0        0:00:53.644107 swwf-fedora-wl1-n-1.englab.juniper.net

redis-swwf-fedora-wl2 swwf-fedora-wl2 jnpr-apm-redis-swwf-fedora-wl2-0                                Running 0        0:01:03.644122 swwf-fedora-wl2-n-1.englab.juniper.net

redis-swwf-fedora-wl2 swwf-fedora-wl2 jnpr-apm-redis-swwf-fedora-wl2-1                                Running 0        0:00:53.644136 swwf-fedora-wl2-n-1.englab.juniper.net

redis-swwf-fedora-wl2 swwf-fedora-wl2 jnpr-apm-redis-swwf-fedora-wl2-sentinels-0                      Running 0        0:00:53.644151 swwf-fedora-wl2-n-1.englab.juniper.net

redis-swwf-fedora-wl2 swwf-fedora-wl2 jnpr-apm-redis-swwf-fedora-wl2-sentinels-1                      Running 0        0:00:53.644165 swwf-fedora-wl2-n-1.englab.juniper.net

redis-swwf-fedora-wl2 swwf-fedora-wl2 jnpr-apm-redis-swwf-fedora-wl2-sentinels-2                      Running 0        0:00:53.644180 swwf-fedora-wl2-n-1.englab.juniper.net

bbe-observer          swwf-fedora-kd address-pool-manager-observer-controller-manager-559f5f4c94gr6z Running 0        0:01:51.644194 swwf-fedora-kd-n-1.englab.juniper.net

bbe-observer          swwf-fedora-kd address-pool-manager-observer-controller-manager-559f5f4c9csxmp Running 0        0:01:51.644208 swwf-fedora-kd-n-1.englab.juniper.net

bbe-observer          swwf-fedora-kd address-pool-manager-observer-controller-manager-559f5f4c9gs67d Running 0        0:01:51.644228 swwf-fedora-kd-n-1.englab.juniper.net

bbe-observer          swwf-fedora-kd address-pool-manager-observer-grpc-server-7594fc9fcd-6n9jt      Running 0        0:01:50.644250 swwf-fedora-kd-n-1.englab.juniper.net

bbe-observer          swwf-fedora-kd address-pool-manager-observer-grpc-server-7594fc9fcd-cckhb      Running 0        0:01:50.644280 swwf-fedora-kd-n-1.englab.juniper.net

bbe-observer          swwf-fedora-kd address-pool-manager-observer-grpc-server-7594fc9fcd-r952s      Running 0        0:01:51.644313 swwf-fedora-kd-n-1.englab.juniper.net


Storage:

swwf-fedora-wl1: Healthy

swwf-fedora-wl2: Healthy
```

```
$ apm status --context context-name --detail
MICROSERVICE POD                                STATE    RESTARTS UPTIME                     NODE
addrman      jnpr-apm-addrman-7b778979b6-5vk44  Running  0        11 days, 23:25:14.629150  test-node-1
mgmt         jnpr-apm-mgmt-6b4cf98d4d-hmjd9     Running  0        11 days, 23:25:33.629206  test-node-1
entman       jnpr-apm-entman-7d66d89d6b-5295d   Running  0        11 days, 23:25:14.629224  test-node-1
provman      jnpr-apm-provman-849fb9cc4-vswm6   Running  0        11 days, 22:50:05.629258  test-node-1
redis        jnpr-apm-redis-0                   Running  0        11 days, 23:25:36.629275  test-node-1
redis        jnpr-apm-redis-1                   Running  0        11 days, 23:25:23.629290  test-node-1
redis        jnpr-apm-redis-sentinels-0         Running  0        11 days, 23:25:26.629306  test-node-1
redis        jnpr-apm-redis-sentinels-1         Running  0        11 days, 23:25:26.629322  test-node-1
redis        jnpr-apm-redis-sentinels-2         Running  0        11 days, 23:25:26.629337  test-node-1
```

```
Storage: Healthy
```

```
$ apm status --context context-name
MICROSERVICE PODS RESTARTS
addrman 1/1 0
mgmt 2/2 0/0
entman 1/1 0
provman 1/1 0
redis 5/5 0/0/0/0/0

Storage: Healthy
```

Check the Status of APM Services Using the Kubernetes Command Line Tool

Use the Kubernetes Command Line tool to check the status of each APM service (functional component) listed in Table 6 on page 77. The status shows whether a service is running, has exited, or has not started. It also displays the service name on the Kubernetes pod. You can compare uptime for the services to quickly see whether any service has been restarted.

NOTE: You use the procedure in this section only for single geographically located APM setups.

Table 7: Services Displayed with the Status Command

Microservice	Pod Prefix
addrman—Address manager.	jnpr-apm-addrman
mgmt—CLI management.	jnpr-apm-mgmt
redis (Remote Dictionary Server)—consists of a set of pods which provide the persistent database.	jnpr-apm-redis

Table 7: Services Displayed with the Status Command *(Continued)*

Microservice	Pod Prefix
entman—Entity manager.	jnpr-apm-entman
provman—Provisioning manager.	jnpr-apm-provman
dbSync—Database synchronization.	jnpr-apm-dbSync
cfgServer—Configuration file replication.	jnpr-apm-cfgServer
bbe-observer—Observes multiple cluster scheduling events for generation number calculation.	jnpr-apm-bbe-observer

To check the status, run the following command:

```
$ kubectl get pods -n jnpr-apm -o wide
```

For example:

```
$ kubectl get pods -n jnpr-apm -o wide

NAME                              READY STATUS  RESTARTS AGE   IP          NODE                        NOMINATED NODE READINESS GATES

jnpr-apm-addrman-7585cc8885-5xr24 1/1   Running  0       11m   10.42.0.31  binnacle.englab.juniper.net <none>         <none>

jnpr-apm-entman-5dcf659676-4mq6g  1/1   Running  0       5d22h 10.42.2.24  jib.englab.juniper.net      <none>         <none>

jnpr-apm-mgmt-6d7c4f47dc-2v8ss    1/1   Running  0       5d22h 10.42.0.30  binnacle.englab.juniper.net <none>         <none>

jnpr-apm-provman-65c66bc689-bvhb6 1/1   Running  0       5d22h 10.42.1.153 keel.englab.juniper.net     <none>         <none>

jnpr-apm-redis-0                  1/1   Running  0       5d22h 10.42.2.22  jib.englab.juniper.net      <none>         <none>

jnpr-apm-redis-1                  1/1   Running  0       5d22h 10.42.1.152 keel.englab.juniper.net     <none>         <none>

jnpr-apm-redis-sentinels-0        1/1   Running  0       5d22h 10.42.1.151 keel.englab.juniper.net     <none>         <none>

jnpr-apm-redis-sentinels-1        1/1   Running  0       5d22h 10.42.0.28  binnacle.englab.juniper.net <none>         <none>

jnpr-apm-redis-sentinels-2        1/1   Running  0       5d22h 10.42.2.23  jib.englab.juniper.net      <none>         <none>
```

Display APM IP Addresses Using the APM Utility

Use the `apm ip` utility script to display the Kubernetes objects that are necessary for the orchestration of the APM pods. For example:

```
$ apm ip --context context-name --detail
SERVICE                         MICROSERVICE  EXTERNAL IP     INTERNAL IP    PORT(S)
apm-ssh                         mgmt          198.19.224.215  10.43.35.110   22
jnpr-apm-mgmt-svc               mgmt                          10.43.131.131  8066
jnpr-apm-redis-sentinels-0-svc  redis                         10.43.142.53   7381
jnpr-apm-redis-sentinels-1-svc  redis                         10.43.109.206  7381
jnpr-apm-redis-sentinels-2-svc  redis                         10.43.104.100  7381
jnpr-apm-redis-svc              redis                         10.43.6.207    7380
apm-apmi                        provman       198.19.224.212  10.43.221.12   20557
```

Display the APM IP Addresses Using the Kubernetes Command Line Tool

Use the `kubectl get services` Kubernetes command to display the Kubernetes objects that are necessary for the orchestration of the APM pods. For example:

```
$ kubectl get services -n jnpr-apm | egrep "TYPE|LoadBalancer"
NAME      TYPE           CLUSTER-IP      EXTERNAL-IP      PORT(S)         AGE
apm-apmi  LoadBalancer   10.43.99.79     198.19.224.212   20557:32136/TCP 5d22h
apm-ssh   LoadBalancer   10.43.35.110    198.19.224.215   22:30261/TCP    146d
```

Display Logging Using the APM Utility

SUMMARY

Use the `apm logs` utility script to display the logs of events that occur while APM is running. You can also use the BBE Event Collection and Visualization utility to display file-based logs collected and stored

IN THIS SECTION

since the time APM is started. BBE Event Collection and Visualization is a cloud-based centralized utility that provides a way to capture APM logs that span the life-cycle of APM micro-services. You link to the BBE Event Collection and Visualization logging utility when you set up APM. See the Broadband Edge Event Collection and Visualization Installation Guide.

- Display Logs Using the Kubernetes Command Line Tool | **85**

Display APM Logging

BBE Event Collection and Visualization is a cloud-based centralized utility that provides a way to capture APM logs that span the life-cycle of APM micro-services. If BBE Event Collection and Visualization is running, during the APM setup you can point BBE Event Collection and Visualization to perform the logging. BBE Event Collection and Visualization has a web-based interface to OpenSearch's capabilities for advanced searching, aggregation, viewing, and data analysis of collected syslog events.

Use the `apm logs` utility script to display the logs of events that occur while APM is running. The event logs include events such as those shown in the following non-exhaustive list:

- Pool-domain registration events
- Address allocation failures
- Network entity connection failures
- Startup messages
- Network entity resynchronization events
- Pool and partition utilization threshold and depletion warnings

By default, APM sends logs to the standard output (stdout) of the service. The output displays the circular buffer of all services or of a specified service. You can also enable logging to follow the log output of the running services. Following the log output creates an open session that continuously streams the logs to stdout.

The APM logging functions mask the underlying complexities of the `kubectl log` command that is collecting the log information. You can still use the `kubectl log` command, but that is outside the scope of this documentation.

NOTE: You can use third-party applications to capture and redirect the stdout stream for the container. Refer to your third-party documentation for assistance. You can also configure Docker with different logging drivers to redirect stdout. Refer to your Docker documentation for assistance.

To display APM logs for all services:

```
$ apm logs [--context contextName] [-f] [--previous] [--services SERVICES [SERVICES ...]] [--
logset LOGSET] [--nocolor]
```

BEST PRACTICE: Use the `apm logs --services` option only when you are troubleshooting under the guidance of a Juniper Networks support representative.

To follow the logs for all services, specify the follow option (`-f`):

```
$ apm logs -f
```

To view logs from previous instances of the containers in a Kubernetes pod, specify the previous container option (`-p`):

```
$ apm logs -p
```

You can use standard Ubuntu conventions to redirect the logs to a file or to the terminal and to a file. Refer to the Ubuntu documentation for more information, but you can use the following examples as a starting point:

- Redirect all logs to only a file.

  ```
  $ apm logs > file-path
  ```

- Redirect all logs to both the screen and to a file.

  ```
  $ apm logs | tee file-path
  ```

BEST PRACTICE: Use the `--services` option only for troubleshooting under the guidance of a Juniper Networks support representative.

You can specify any of the following severity levels, in order of increasing severity:

Table 8: Severity Level

Severity Level	Description
`debug`	Detailed information that is typically of interest only when you are trying to diagnose a problem. These logs are often very frequent.
`info`	Events or non-error conditions of interest. Logs at this level provide confirmation that everything is working as expected. These logs are generally not very frequent.
`warning`	Indicates that something unexpected happened or that some problem might occur in the near future. A simple example of the latter is the *disk space low* warning that indicates that you might run out of disk space soon. In either case, the software is still working as expected, but you might want to monitor it more closely. These logs are generally not very frequent.
`error`	Indicates that a more serious problem has prevented the software from performing some function, but the software has handled the problem as gracefully as possible to continue functioning.
`critical`	A serious error that indicates that the program itself might be unable to continue running.

You can use the `--logset` option to display logs either for only APM services or for only prebuilt services. If you do not use this option, then only the APM services logs are displayed.

To display logs for only APM services:

```
$ apm logs --logset apm
$ apm logs
```

To display logs for prebuilt services:

```
$ apm logs --logset apm-infra
```

Prebuilt services are services borrowed from other sources to provide infrastructure functions for APM. These sources include mgmt, redis, and redis-sentinel. For example, redis provides database and messaging services, mgmt provides configuration and CLI services, and so on.

Display Logs Using the Kubernetes Command Line Tool

To display all the logs of events that occur while APM is running, run the following command:

```
$ kubectl logs -n jnpr-apm -l jnpr/logset=jnpr-apm --tail=-1
```

To display logs for a specific microservice, replace the label selector (`-l jnpr/logset=jnpr-apm`) with the pod name. For example:

```
$ kubectl logs -n jnpr-apm microservice-pod-name --tail=-1
```

NOTE: To determine the pod name, you can use the `kubectl get pods -n jnpr-apm` Kubernetes command (see "Check the Status of APM Services Using the Kubernetes Command Line Tool" on page 79).

Determine the APM Version Using the APM Utility

Use the `apm version [--context <context name>] [-o|--output json] [--detail| --compare <software-version>]` utility script to determine the version number of the installed APM release.

To display the release version:

```
$ apm version --context context-name --detail
Address_Pool_Manager versions:
  Microservice  Release            (version)
  apm:          3.5.0
  addrman:      3.5.0
  entman:       3.5.0
  mgmt:         3.5.0 (24.4R2)
  provman:      3.5.0
  redis:        3.5.0  (6.2.16-debian-12-r3)
```

To compare the specified software release versions against the current deployed release for the specified context:

```
apm version --compare 3.3.0 --context context-name
components:
apm: 3.4.0 -> 3.5.0
addrman: 3.4.0 -> 3.5.0
entman: 3.4.0 -> 3.5.0
mgmt: 3.4.0 -> 3.5.0
provman: 3.4.0 -> 3.5.0
redis: 6.2.13 -> 6.2.16-debian-12-r3
```

Use the `-j` option to render the version information in JavaScript Object Notation (JSON) format.

Archive the APM Configuration Using the Kubernetes Command Line Tool

To archive a copy of the currently running APM configuration, enter the following command:

```
$ kubectl cp jnpr-apm/$( kubectl get pods -n jnpr-apm -l jnpr/cli=cli --no-headers=true | awk
'{print $1}'):config/juniper.conf.gz ./juniper.conf.gz
```

Uninstall and Remove APM Using the APM Utility

Use the `apm` utility script to uninstall the APM configuration. The `uninstall` command reverts the actions you performed when setting up APM. Use this command to return APM to the state it was in immediately after you installed the application but before you did any setup configuration.

To uninstall APM:

```
$ sudo -E apm stop --context context-name
sudo -E apm unlink --context context-name
```

After you uninstall APM, we recommend that you use the Debian uninstall procedure to remove the entire package.

```
$ sudo sudo -E apm clean [-h] [--log {error | warn | info | debug}] [--no-color] [--docker] [--
release release-number] [--dry-run] [--uninstall] [--cluster-repos] ]
```

For the `apm clean` command options, see the following:

- `-h` or `--help`—Show the help message and exit.
- `--log` or `-l` {error | warn | info | debug}—Adjust the log level of the utility scrip.
- `--no-color`—Print messages without colors.
- `--docker`—Clean the local docker cache.
- `--release` *`release-number`*—The release to clean (defaults to unused releases).
- `--dry-run`—List releases or containers that will be removed.
- `--uninstall`—Uninstall all software releases and remove APM from the system.
- `--cluster-repos`—Clean the cluster repos for the clusters that have been removed.

Uninstall and Remove APM Without Using the APM Utility

This is the uninstall procedure for APM deployments that were not created with the APM utility. Use the `helm uninstall` command to uninstall your APM configuration. To completely remove APM, you must run the `helm uninstall` command for each microservice.

To uninstall APM, run the following:

```
helm uninstall --kube-context context-name -n jnpr-apm provman
```

```
helm uninstall --kube-context context-name -n jnpr-apm entman
```

```
helm uninstall --kube-context context-name -n jnpr-apm addrman
```

```
helm uninstall --kube-context context-name -n jnpr-apm mgmt
```

```
helm uninstall --kube-context context-name -n jnpr-apm redis
```

How to Access APM Configuration and Operational Commands Using the APM Utility

IN THIS SECTION

Access the APM CLI Using the APM Utility

To access the CLI prompt, enter the following `apm` utility script command:

```
$ apm cli --context context-name [-p|--pipe]
```

```
root@jnpr-apm-mgmt
```

Enter a question mark to see the available top-level CLI commands. This list of commands is a subset of the Junos OS top-level commands.

```
root@jnpr-apm-mgmt> ?
Possible completions:
  clear                Clear information in the system
  configure            Manipulate software configuration information
  file                 Perform file operations
  help                 Provide help information
  monitor              Show real-time debugging information
  op                   Invoke an operation script
  quit                 Exit the management session
  request              Make system-level requests
  restart              Restart software process
  set                  Set CLI properties, date/time, craft interface message
  show                 Show system information
  ssh                  Start secure shell on another host
  start                Start shell
  telnet               Telnet to another host
  test                 Perform diagnostic debugging
  traceroute           Trace route to remote host
```

For an overview of Junos OS CLI basics, see Day One: Exploring the Junos CLI. For more detailed information, see the CLI User Guide.

Access and Use CLI Configuration Statements Using the APM Utility

1. Use the APM utility command `apm cli` to access the top-level CLI prompt.

   ```
   $ apm cli --context context-name
   root@jnpr-apm-mgmt>
   ```

2. Access configuration mode to configure APM and the information that APM uses to configure a managed router.

```
root@jnpr-apm-mgmt> configure
root@jnpr-apm-mgmt#
```

3. Enter CLI statements to configure the APM-managed BNGs, pool domains, pools, and system attributes.
4. Save and activate the configuration. This command succeeds only when there are no configuration syntax errors.

```
root@jnpr-apm-mgmt# commit
commit complete
```

5. (Optional) Exit configuration mode and return to the top-level CLI prompt.

```
root@jnpr-apm-mgmt# exit
root@jnpr-apm-mgmt>
```

Access and Use CLI Operational Commands Using the APM Utility

To monitor APM, view APM configuration and statistics, or run certain operations manually:

1. Use the APM utility command `apm cli` to access the top-level CLI prompt.

```
$ $ apm cli --context context-name
root@jnpr-apm-mgmt>
```

2. Enter specific commands.

   - Use `show` commands to display statistics and the relationships between partitions, BNGs, pool domains, and pools.

   - Use `request` commands to manually initiate certain APM operations.

How to Use the APM Command Line Tool Without Using the APM Utility

SUMMARY

This section describes how you use the Kubernetes Command Line tool commands to perform administration functions.

IN THIS SECTION

You can use the Kubernetes Command Line tool to administer the application and to access the CLI that you use to configure the address management functions.

You can use the Kubernetes Command Line tool to do the following:

- Create and delete objects.
- Provide log access.
- Conduct interactive sessions with pod containers.
- Display the status of the APM objects.

Access the APM Command Line Tool Without Using the APM Utility

To access the APM Command Line tool using the Kubernetes commands, enter the following:

```
$ kubectl exec -it -n jnpr-apm $(kubectl get pods -n jnpr-apm -l jnpr/cli=cli --no-headers=true
| awk '{print $1}') -- cli
```

Enter a question mark to see the available top-level CLI commands.

```
 ?
Possible completions:
  clear                Clear information in the system
  configure            Manipulate software configuration information
  file                 Perform file operations
  help                 Provide help information
  monitor              Show real-time debugging information
  op                   Invoke an operation script
  quit                 Exit the management session
  request              Make system-level requests
  restart              Restart software process
  set                   Set CLI properties, date/time, craft interface message
  show                  Show system information
  ssh                   Start secure shell on another host
  start                 Start shell
  telnet                Telnet to another host
  test                  Perform diagnostic debugging
  traceroute            Trace route to remote host
```