NetBackup™ Snapshot Manager Install and Upgrade Guide

Ubuntu, RHEL, SLES

Release 10.1

August 2022

VERITAS™

Veritas NetBackup™ Snapshot Manager Install and Upgrade Guide

Last updated: 2022-08-30

Legal Notice

Veritas Technologies LLC
2625 Augustine Drive
Santa Clara, CA 95054

http://www.veritas.com

Technical Support

Technical Support maintains support centers globally. All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policies. For information about our support offerings and how to contact Technical Support, visit our website:

https://www.veritas.com/support

You can manage your Veritas account information at the following URL:

https://my.veritas.com

If you have questions regarding an existing support agreement, please email the support agreement administration team for your region as follows:

Worldwide (except Japan)	CustomerCare@veritas.com
Japan	CustomerCare_Japan@veritas.com

Documentation

Make sure that you have the current version of the documentation. Each document displays the date of the last update on page 2. The latest documentation is available on the Veritas website:

https://sort.veritas.com/documents

Documentation feedback

Your feedback is important to us. Suggest improvements or report errors or omissions to the documentation. Include the document title, document version, chapter title, and section title of the text on which you are reporting. Send feedback to:

NB.docs@veritas.com

You can also see documentation information or ask a question on the Veritas community site:

http://www.veritas.com/community/

Veritas Services and Operations Readiness Tools (SORT)

Veritas Services and Operations Readiness Tools (SORT) is a website that provides information and tools to automate and simplify certain time-consuming administrative tasks. Depending on the product, SORT helps you prepare for installations and upgrades, identify risks in your datacenters, and improve operational efficiency. To see what services and tools SORT provides for your product, see the data sheet:

https://sort.veritas.com/data/support/SORT_Data_Sheet.pdf

Contents

Chapter 1

Introduction

This chapter includes the following topics:

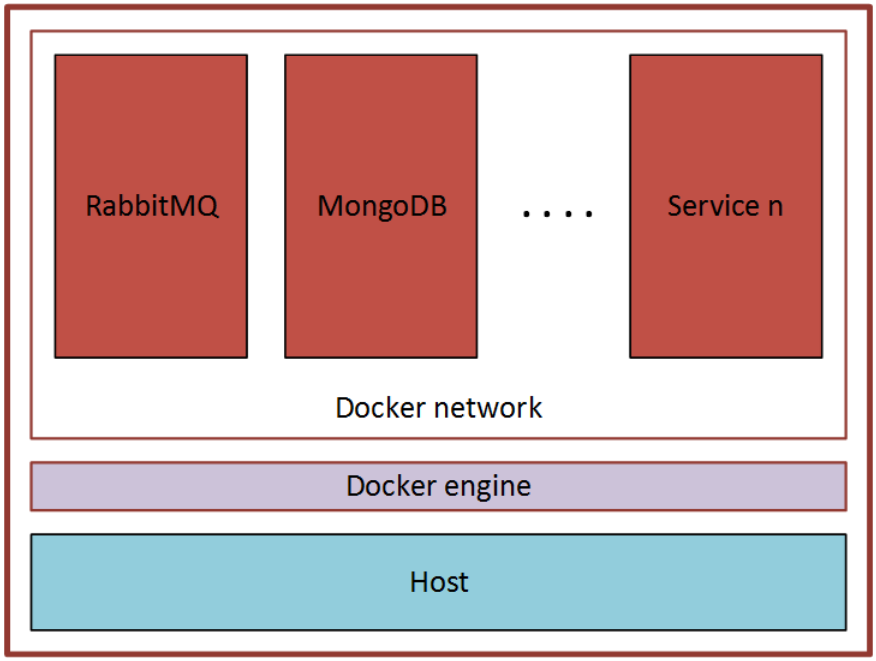
- About the deployment approach
- Deciding where to run Snapshot Manager
- About deploying Snapshot Manager in the cloud

About the deployment approach

Snapshot Manager uses a micro-services model of installation. When you load and run the Docker image, Snapshot Manager installs each service as an individual container in the same Docker network. All containers securely communicate with each other using RabbitMQ.

Two key services are RabbitMQ and MongoDB. RabbitMQ is Snapshot Manager's message broker, and MongoDB stores information on all the assets Snapshot Manager discovers. The following figure shows Snapshot Manager's micro-services model.

Figure 1-1 Snapshot Manager's micro-services model



This deployment approach has the following advantages:

- Snapshot Manager has minimal installation requirements.
- Deployment requires only a few commands.

Snapshot Manager solution can be deployed on Kubernetes Service Cluster environment. For more information, refer to *NetBackup Deployment Guide for Azure Kubernetes Services (AKS) Cluster*.

Deciding where to run Snapshot Manager

You can deploy Snapshot Manager in the following ways:

- Deploy Snapshot Manager in a cloud and manage assets in that cloud.
- Deploy Snapshot Manager in a cloud and manage assets in multiple clouds.

Veritas recommends that you deploy Snapshot Manager on cloud to protect your cloud assets. If you wish to protect assets in a cloud, deploy the Snapshot Manager host instance in the same cloud environment.

Similarly, if you wish to protect on-premise assets, deploy the Snapshot Manager host in the same on-premise environment. For detailed information about the on-premise content, refer to the *NetBackup Snapshot Manager for Data Center Administrator's Guide*.

You can deploy Snapshot Manager in a NetBackup media server, but not in a NetBackup primary server.

If you install Snapshot Manager on multiple hosts, we strongly recommend that each Snapshot Manager instance manage separate resources. For example, two Snapshot Manager instances should not manage the same AWS account or the same Azure subscription. The following scenario illustrates why having two Snapshot Manager instances manage the same resources creates problems:

- Snapshot Manager instance A and Snapshot Manager instance B both manage the assets of the same AWS account.
- On Snapshot Manager instance A, the administrator takes a snapshot of an AWS virtual machine. The database on Snapshot Manager instance A stores the virtual machine's metadata. This metadata includes the virtual machine's storage size and its disk configuration.
- Later, on Snapshot Manager instance B, the administrator restores the virtual machine snapshot. Snapshot Manager instance B does not have access to the virtual machine's metadata. It restores the snapshot, but it does not know the virtual machine's specific configuration. Instead, it substitutes default values for the storage size configuration. The result is a restored virtual machine that does not match the original.

If you host the Snapshot Manager and media server in the same host, do the following for proper functioning of the backup from snapshot jobs:

- Assign distinct IPs and NBU client names to the Snapshot Manager and the media server so that they can obtain different NetBackup Certificates. This is required so as have different NetBackup host ID certificates for communication. Use the following configuration:
  - Configure host with two network adapters
  - Edit the `/etc/hosts` file and make entry as mentioned in the example below:

    ```
    <IP Address MediaServer Host1>  < MediaServer Host1>
    <IP Address Snapshot Manager Host2>  <Snapshot Manager Host2>
    ```

  - Provide the MediaServer Host1 which is mentioned in the `/etc/hosts` file during the Media server installation for Media server name.
  - Similarly select the Snapshot Manager Host 2 from the `/etc/hosts` file during the Snapshot Manager installation with non-default port other than 443.

    - Start Snapshot Manager and Media services and register it with NetBackup primary server.
- Once the Snapshot Manager is registered, ensure that it has a different HOST DB entry.
- Before performing the backup from snapshot jobs, perform the following optimization: DISABLE SHM and NOSHM. See: https://www.veritas.com/support/en_US/article.100016170

This will ensure that NetBackup does not use shared memory for communicating between NetBackup data mover processes.

About deploying Snapshot Manager in the cloud

A common deployment approach for Snapshot Manager is to set up a Snapshot Manager instance in the cloud and then configure it to protect and manage all the assets in the cloud. You can deploy Snapshot Manager either manually or using the Snapshot Manager template available in the online marketplace.

In case of manual Snapshot Manager deployment, ensure the UUID of Snapshot Manager boot disk is unique and does not conflict with FS UUID of any other asset node.

Refer to Explore NetBackup section for more information on how to deploy a Snapshot Manager instance in the cloud.

Section 1

NetBackup Snapshot Manager installation and configuration

Chapter 2

Preparing for NetBackup Snapshot Manager installation

This chapter includes the following topics:

- Meeting system requirements
- Snapshot Manager host sizing recommendations
- Snapshot Manager extension sizing recommendations
- Creating an instance or preparing the host to install Snapshot Manager
- Installing container platform (Docker, Podman)
- Creating and mounting a volume to store Snapshot Manager data
- Verifying that specific ports are open on the instance or physical host
- Preparing Snapshot Manager for backup from snapshot jobs

Meeting system requirements

Snapshot Manager host requirements

The host on which you install Snapshot Manager must meet the following requirements.

See “Snapshot Manager host sizing recommendations” on page 25.

Table 2-1 Operating system, processor and package requirements for Snapshot Manager host

Category	Requirement
Operating system	■ Ubuntu 18.04 and 20.04 Server LTS ■ Red Hat Enterprise Linux (RHEL) 8.6, 8.4 and 7.x **Note:** Snapshot Manager deployment for RHEL 8.6 and 8.4 over IPV6 is not supported. ■ SUSE Linux Enterprise Server (SLES) 15 SP2
Processor architecture	x86_64 /64-bit processors
Packages on Snapshot Manager host	Following are the operating system specific respective required packages to be installed on Snapshot Manager host: ■ **Ubuntu**: lvm2, udev ■ **SUSE**: lvm2, udev ■ **RHEL 7**: lvm2, systemd ■ **RHEL 8**: podman-plugins, lvm2, systemd-udev

Table 2-2 System requirements for the Snapshot Manager host

Host on which Snapshot Manager is installed	Requirements
Amazon Web Services (AWS) instance	■ Elastic Compute Cloud (EC2) instance type: t3.large ■ vCPUs: 2 ■ RAM: 8 GB ■ Root disk: 64 GB with a solid-state drive (GP2) ■ Data volume: 50 GB Elastic Block Store (EBS) volume of type GP2 with encryption for the snapshot asset database; use this as a starting value and expand your storage as needed.

Table 2-2 System requirements for the Snapshot Manager host *(continued)*

Host on which Snapshot Manager is installed	Requirements
Microsoft Azure VM	■ Virtual machine type: D2s_V3 Standard ■ CPU cores: 2 ■ RAM: 8 GB ■ Root disk: 64 GB SSD ■ Data volume: 50 GB Premium SSD for the snapshot asset database; storage account type Premium_LRS; set Host Caching to Read/Write. Ensure that do the following before you deploy Snapshot Manager on an RHEL instance in the Azure cloud: ■ Register the RHEL instance with Red Hat using Red Hat Subscription Manager ■ Extend the default LVM partitions on the RHEL instance so that they fulfil the minimum disk space requirement

Microsoft Azure Stack Hub VM	■ Virtual machine types:
	  ■ DS2_v2 Standard - CPU cores 2, RAM 7 GB
	  ■ DS3_v2 Standard - CPU cores 4, RAM 14 GB
	■ Root disk: 64 GB SSD
	■ Data volume: 50 GB Premium SSD for the snapshot asset database; storage account type Premium_LRS; set Host Caching to Read/Write.
	
	Ensure that do the following before you deploy Snapshot Manager on an RHEL instance in the Azure Stack Hub cloud:
	
	■ Register the RHEL instance with Red Hat using Red Hat Subscription Manager
	■ Extend the default LVM partitions on the RHEL instance so that they fulfil the minimum disk space requirement

Google Cloud Platform (GCP) VM	■ Virtual machine type: n2-standard-4 ■ vCPUs: 2 ■ RAM: 16 GB ■ Boot disk: 64 GB standard persistent disk ■ Data volume: 50 GB SSD persistent disk for the snapshot asset database with automatic encryption **Note:** To support LVM indexing, ensure that the Multipath service is disabled on Snapshot Manager host.

Table 2-2 System requirements for the Snapshot Manager host *(continued)*

Host on which Snapshot Manager is installed	Requirements
VMware VM	■ Virtual machine type: 64-bit with a Snapshot Manager supported operating system ■ vCPUs: 8 ■ RAM: 16 GB or more ■ Root disk: 64 GB with a standard persistent disk ■ Data volume: 50 GB for the snapshot asset database
Physical host *(x86_64 / AMD64)*	■ Operating system: A 64-bit Snapshot Manager supported operating system ■ CPUs: x86_64 (64-bit), single-socket, multi-core, with at least 8 CPU count ■ RAM: 16 GB or more ■ Boot disk: 64 GB ■ Data volume: 50 GB for the snapshot asset database

Note: NetBackup Snapshot Manager is not fully FIPS compliant.

Disk space requirements

Snapshot Manager uses the following file systems on the host to store all the container images and files during installation:

- `/` *(root file system)*
- `/var`

The `/var` file system is further used for container runtimes. Ensure that the host on which you install or upgrade Snapshot Manager has sufficient space for the following components.

Table 2-3 Space considerations for Snapshot Manager components

Component	Space requirements
Snapshot Manager containers	30 GB free space
Snapshot Manager agents and plug-ins	350 MB free space, for every Snapshot Manager plug-in and agent configured

Additionally, Snapshot Manager also requires a separate volume for storing Snapshot Manager data. Ensure that you create and mount this volume to `/cloudpoint` on the Snapshot Manager host.

Table 2-4 Space consideration for Snapshot Manager data volume

Volume mount path	Size
`/cloudpoint`	50 GB or more

See “Snapshot Manager host sizing recommendations” on page 25.

Applications, operating systems, and cloud platforms supported by Snapshot Manager agents and plug-ins

Snapshot Manager supports the following applications, operating systems and cloud platforms.

These assets are supported irrespective of how you configure Snapshot Manager, whether using the Snapshot Manager cloud agents and plug-ins (earlier known as off-host plug-ins), or using the Snapshot Manager application configuration plug-ins (earlier known as on-host plug-ins), or using the Snapshot Manager agentless feature.

Table 2-5 Supported applications, operating systems, and cloud platforms

Category	Support

Applications	■ File systems
	  ■ Linux native file systems: ext3, ext4, and XFS
	  ■ Microsoft Windows: NTFS
	■ Microsoft SQL 2014, SQL 2016, SQL 2017, SQL 2019
	■ Windows Server 2022 and 2019
	■ Oracle 12c, Oracle 12c R1, Oracle 18c, Oracle 19c
	Single node configurations are supported.
	See “Oracle plug-in configuration requirements” on page 146.

Operating systems on supported assets	■ Red Hat Enterprise Linux (RHEL) 8.6 and 8.4 ■ Windows Server 2012 R2, 2016, 2019 and 2022 **Note:** Snapshot Manager agents are not supported on non-English operating systems.

Table 2-5 Supported applications, operating systems, and cloud platforms *(continued)*

Category	Support
Cloud platforms	

Table 2-5 Supported applications, operating systems, and cloud platforms *(continued)*

Category	Support
	■ Amazon Web Services (AWS) If you wish to protect applications, the applications must be hosted on a t2.large or a higher specification AWS instance type. Snapshot Manager currently does not support applications that are running on t2.medium or a lower instance type. The t2 series instances are supported only if the device naming conventions recommended by AWS are followed. For more details, refer to the following links: ■ Windows: Device names on Windows instances ■ Linux: Device names on Linux instances For protecting Microsoft Windows-based applications, use t2.xlarge or t3.xlarge or a higher specification instance type. For more information on the required permissions for configuring AWS, refer to the following link: See “AWS permissions required by Snapshot Manager” on page 97. ■ Microsoft Azure If you wish to protect applications, the applications must be hosted on a D2s_V3 Standard or a higher specification Azure virtual machine type. For protecting Microsoft Windows-based applications, use B4ms or D4s_V3 or a higher specification virtual machine. **Note:** The Snapshot Manager Azure plug-in supports disks of type Premium_LRS, Standard_LRS, and StandardSSD_LRS. All other disk types are defaulted to Standard_LRS during snapshot restore operations. For more information on the required permissions for configuring Azure, refer to the following link: See “Configuring permissions on Microsoft Azure” on page 117. ■ Microsoft Azure Stack Hub (2008 and later) If you wish to protect applications, the applications must be hosted on a DS2_v2 Standard or a higher specification Azure Stack Hub virtual machine type. For more information, see VM sizes supported in Azure Stack Hub.

Table 2-5 Supported applications, operating systems, and cloud platforms *(continued)*

Category	Support
	Note: The Snapshot Manager Azure Stack Hub plug-in supports disks of type Premium_LRS, Standard_LRS, and StandardSSD_LRS. All other disk types are defaulted to Standard_LRS during snapshot restore operations. For more information on the required permissions for configuring Microsoft Azure Stack, refer to the following link: See “Configuring permissions on Microsoft Azure Stack Hub” on page 122. ■ Google Cloud Platform (GCP) If you wish to protect applications, the applications must be hosted on a n2-standard-4 or a higher specification GCP virtual machine type. For more information on the required permissions for configuring Google cloud platform, refer to the following link: See “Google Cloud Platform permissions required by Snapshot Manager” on page 109.

Snapshot Manager time zone

Ensure that the time zone settings on the host where you wish to deploy Snapshot Manager are as per your requirement and synchronized with a public NTP server.

By default, Snapshot Manager uses the time zone that is set on the host where you install Snapshot Manager. The timestamp for all the entries in the logs are as per the clock settings of the host machine.

Proxy server requirements

If the instance on which you are deploying Snapshot Manager is behind a proxy server, that is, if the Snapshot Manager instance connects to the internet using a proxy server, you must specify the proxy server details during the Snapshot Manager installation. The Snapshot Manager installer stores the proxy server information in a set of environment variables that are specific for the Snapshot Manager containers.

The following table displays the environment variables and the proxy server information that you must provide to the Snapshot Manager installer. Make sure you keep this information ready; you are required to provide these details during Snapshot Manager installation.

Table 2-6 Proxy server details required by Snapshot Manager

Environment variables created by Snapshot Manager installer	Description
VX_HTTP_PROXY	Contains the HTTP proxy value to be used for all connections. For example, "http://proxy.mycompany.com:8080/".
VX_HTTPS_PROXY	Contains the HTTP proxy value to be used for all connections. For example, "http://proxy.mycompany.com:8080/".
VX_NO_PROXY	Contains the hosts that are allowed to bypass the proxy server. For example, "localhost,mycompany.com,192.168.0.10:80".

Snapshot Manager services that need to communicate externally via a proxy server use these predefined environment variables that are set during the Snapshot Manager installation.

Snapshot Manager host sizing recommendations

The Snapshot Manager host configuration depends primarily on the number of workloads and also the type of workloads that you wish to protect. It is also dependent on the maximum number of simultaneous operations running on the Snapshot Manager at its peak performance capacity.

Another factor that affects performance is how you use Snapshot Manager for protecting your assets. If you use the Snapshot Manager agentless option to discover and protect your assets, then the performance will differ depending on the type of workload.

With agentless, Snapshot Manager transfers the plug-in data to the application host, performs the discovery and configuration tasks, and then removes the plug-in package from the application host.

Veritas recommends the following configurations for the Snapshot Manager host:

Table 2-7 Typical Snapshot Manager host configuration based on the number of concurrent tasks

Workload metric	Snapshot Manager host configuration
Up to 16 concurrent operational tasks	CPU: 2 CPUs Memory: 16 GB For example, in the AWS cloud, the Snapshot Manager host specifications should be an equivalent of a **t3.xlarge** instance.
Up to 32 concurrent operational tasks	CPU: 4 - 8 CPUs Memory: 32 GB or more For example, in the AWS cloud, the Snapshot Manager host specifications should be an equivalent of a **t3.2xlarge** or a higher type of instance.

General considerations and guidelines:

Consider the following points while choosing a configuration for the Snapshot Manager host:

- To achieve better performance in a high workload environment, Veritas recommends that you deploy the Snapshot Manager host in the same location as that of the application hosts.
- If you are using the agentless option, Veritas recommends that you allocate enough space to the `/tmp` directory on the application host. Snapshot Manager uses this directory for extracting the plug-in configuration files.
- Depending on the number of workloads, the amount of plug-in data that is transmitted from the Snapshot Manager host can get really large in size. The network latency also plays a key role in such a case. You might see a difference in the overall performance depending on these factors.
- If you wish to configure multiple workloads using the agentless option, then the performance will be dependent on factors such as the network bandwidth and the location of the Snapshot Manager host with respect to the application workload instances. You can, if desired, bump up the Snapshot Manager host's CPU, memory, and network configuration to achieve a performance improvement in parallel configurations of agentless application hosts.
- In cases where the number of concurrent operations is higher than what the Snapshot Manager host configuration capacity can handle, Snapshot Manager automatically puts the operations in a job queue. The queued jobs are picked up only after the running operations are completed.

Snapshot Manager extension sizing recommendations

The Snapshot Manager extension serves the purpose of scaling the capacity of the Snapshot Manager host to service a large number of requests concurrently running on the Snapshot Manager at its peak performance capacity. You can install one or more Snapshot Manager extensions on-premise or in cloud, depending on your requirements to run the jobs without putting the host under additional stress. An extension can increase the processing capacity of the Snapshot Manager.

The Snapshot Manager extension can have the configuration same or higher as the Snapshot Manager host.

See “ Meeting system requirements” on page 17.

Supported Snapshot Manager extension environments:

- VM based extension for on-premise
- Cloud based extension with managed Kubernetes cluster

Note: For Snapshot Manager 10.0, the VM based extensions are supported on Azure Stack hub and Kubernetes based extension are supported on Azure, AWS and GCP.

Veritas recommends the following configurations for the Snapshot Manager extensions:

Table 2-8 Typical Snapshot Manager extension configuration for VM based extension (Azure stack)

Workload metric	Snapshot Manager extension configuration
Up to 16 concurrent operational tasks	CPU: 4 CPUs Memory: 16 GB For example, in Azure stack, the Snapshot Manager extension should be an equivalent of a **t3.xlarge** instance in AWS.

Table 2-8 Typical Snapshot Manager extension configuration for VM based extension (Azure stack) *(continued)*

Workload metric	Snapshot Manager extension configuration
Up to 32 concurrent operational tasks	CPU: 8 CPUs Memory: 32 GB or more For example, in Azure stack, the Snapshot Manager extension should be an equivalent of a **t3.2xlarge** or a higher type of instance in AWS.

Table 2-9 Typical Snapshot Manager extension configuration for Kubernetes based extension (Azure, AWS and GCP)

Workload metric	Snapshot Manager extension configuration
Up to 24 concurrent operational tasks	**For Azure** ■ For 2 CPU's and 8 GB RAM node configuration: CPU: More than 2 CPU's RAM per node: 8GB Maximum pods per node: 6 (system) + 4 (Static pods) + 8*2=16 (Dynamic pods) = 26 or more Autoscaling enabled, with minimum=1, maximim=3 ■ For 2/4/6 CPU's and 16 GB node configuration CPU per node: More than 2/4/6 CPU's RAM per node: 16 GB Maximum pods per node: 6 (system) + 4 (Static pods) + 16*2=32 (Dynamic pods) = 42 or more Autoscaling enabled, with minimum=1, maximum=3 **Note:** Above configuration would run 16 jobs per node at once.

Table 2-9 Typical Snapshot Manager extension configuration for Kubernetes based extension (Azure, AWS and GCP) *(continued)*

Workload metric	Snapshot Manager extension configuration
Up to 24 concurrent operational tasks	**For AWS** CPU: More than 2 CPU's RAM per node: 8 GB Autoscaling enabled, with minimum =1 and maximum =3 **Note:** Above configuration would run 8 jobs per node at once.
Up to 24 concurrent operational tasks	**For GCP** CPU: More than 2 CPU's per node Memory: 8 GB per node Autoscaling enabled, with minimum =1 and maximum =3

General considerations and guidelines:

Consider the following points while choosing a configuration for the Snapshot Manager extension:

- To achieve better performance in a high workload environment, Veritas recommends that you deploy the Snapshot Manager extension in the same location as that of the application hosts.
- The cloud-based extension on a managed Kubernetes cluster should be in the same VNet as that of the Snapshot Manager host. If it is not, then you can make use of the VNet peering mechanism available with the Azure cloud, to make sure that Snapshot Manager host and extension nodes can communicate with each other over the required ports
- Depending on the number of workloads, the amount of plug-in data that is transmitted from the Snapshot Manager host can get really large in size. The network latency also plays a key role in such a case. You might see a difference in the overall performance depending on these factors.
- In cases where the number of concurrent operations is higher than what the Snapshot Manager host and the extensions together can handle, Snapshot Manager automatically puts the operations in a job queue. The queued jobs are picked up only after the running operations are completed.

Creating an instance or preparing the host to install Snapshot Manager

If you are deploying Snapshot Manager in a public cloud, do the following:

- Choose a supported Ubuntu, RHEL, or SLES instance image that meets Snapshot Manager installation requirements.
- Add sufficient storage to the instance to meet the installation requirements.

If you are deploying Snapshot Manager on an on-premise instance, do the following:

- Install a supported Ubuntu, RHEL, or SLES operating system on a physical or a virtual x86 server.
- Add sufficient storage to the server to meet the installation requirements.

Installing container platform (Docker, Podman)

Table 2-10 Installing container platform

Platform	Description
Docker on Ubuntu	Supported version: Docker 18.09 and later For detailed instructions on installing the Docker on Ubuntu, see Install Docker Engine on Ubuntu.

Table 2-10 Installing container platform *(continued)*

Platform	Description

Docker on RHEL 7.x	Supported version: Docker 1.13.x and later
	
	Use the following process to install Docker on RHEL. Steps may vary depending on whether Snapshot Manager is being deployed on-premise or in the cloud.
	
	■ (If Snapshot Manager is being deployed in AWS cloud) Ensure that you enable the extra repos:
	`# sudo yum-config-manager --enable rhui-REGION-rhel-server-extras`
	■ (If Snapshot Manager is being deployed on-premise) Enable your subscriptions:
	`# sudo subscription-manager register --auto-attach --username=<username> --password=<password>`
	`# subscription-manager repos --enable=rhel-7-server-extras-rpms`
	`# subscription-manager repos --enable=rhel-7-server-optional-rpms`
	■ Install Docker using the following command:
	`# sudo yum -y install docker`
	■ Reload the system manager configuration using the following command:
	`# sudo systemctl daemon-reload`
	■ Enable and then restart the docker service using the following commands:
	`# sudo systemctl enable docker`
	`# sudo systemctl restart docker`
	■ If SELinux is enabled, change the mode to permissive mode.
	Edit the `/etc/selinux/config` configuration file and modify the `SELINUX` parameter value to `SELINUX=permissive`.
	■ Reboot the system for the changes to take effect.
	■ Verify that the SELinux mode change is in effect using the following command:
	`# sudo sestatus`
	The `Current Mode` parameter value in the command output should appear as `permissive`.
	
	For detailed instructions on installing Docker on RHEL, see Getting Docker in RHEL 7.
	
	If the docker is using default storage driver (overlay2 or overlay) on XFS backed file system, then ensure that XFS FS has `ftype` option set to *1*. Use `xfs_info` to verify. For details, see Use the OverlayFS storage driver. Otherwise, you can use different storage driver. For details, see Docker storage drivers.

Table 2-10 Installing container platform *(continued)*

Platform	Description

Podman on RHEL 8.6 and 8.4	Supported version: Podman 4.0.2 and later
	Notes:
	■ (If Snapshot Manager is being deployed in AWS cloud) Ensure that you enable the extra repos:
	`# sudo yum-config-manager --enable rhui-REGION-rhel-server-extras`
	■ (If Snapshot Manager is being deployed on-premise) Enable your subscriptions:
	`# sudo subscription-manager register --auto-attach --username=<username> --password=<password>`
	■ If SELinux is enabled, change the mode to permissive mode.
	Edit the `/etc/selinux/config` configuration file and modify the `SELINUX` parameter value to `SELINUX=permissive`.
	■ Reboot the system for the changes to take effect.
	■ Verify that the SELinux mode change is in effect using the following command:
	`# getenforce`
	The `Current Mode` parameter value in the command output should appear as `permissive`.

Creating and mounting a volume to store Snapshot Manager data

Before you deploy the Snapshot Manager or Snapshot Manager extension in a cloud environment:

- You must create and mount a volume of at least 50 GB to store Snapshot Manager data. The volume must be mounted to `/cloudpoint`.
- Ensure that UUID of the volume and the mount point (`/cloudpoint`) are mentioned in the `/etc/fstab` so that the volume is auto mounted when the host or the extension is rebooted.

Note: If you ever boot your instance without this volume attached (for example, after moving the volume to another instance), the `nofail` mount option enables the instance to boot even if there are errors mounting the volume.

Table 2-11 Volume creation steps for each supported cloud vendor

Vendor	Procedure
Amazon Web Services (AWS)	**1** On the EC2 dashboard, click **Volumes > Create Volumes**. **2** Follow the instructions on the screen and specify the following: ■ Volume type: General Purpose SSD ■ Size: 50 GB **3** Use the instructions provided in the Make an Amazon EBS volume available for use on Linux section to create a file system and mount the device to `/cloudpoint` on the instance host.
Google Cloud Platform	◆ Create the disk for the virtual machine, initialize it, and mount it to `/cloudpoint`. For more information, see Add a persistent disk to your VM.
Microsoft Azure	**1** Create a new disk and attach it to the virtual machine. For more information, see Use the portal to attach a data disk to a Linux VM. You should choose the managed disk option. For more information, see Use the portal to attach a data disk to a Linux VM. **2** Initialize the disk and mount it to `/cloudpoint`. For more information, see the "Connect to the Linux VM to mount the new disk" section of the Add a disk to a Linux VM.
Microsoft Azure Stack Hub	**1** Create a new disk and attach it to the virtual machine. For more information, see Create VM disk storage in Azure Stack Hub. You should choose the managed disk option. **2** Initialize the disk and mount it to `/cloudpoint`. For more information, see the "Connect to the Linux VM to mount the new disk" section of the Add a disk to a Linux VM.

Verifying that specific ports are open on the instance or physical host

Ensure that the following ports are open on the instance or physical host.

Table 2-12 Ports used by Snapshot Manager

Port	Description
443	The Snapshot Manager user interface uses this port as the default HTTPS port.
5671	The Snapshot Manager RabbitMQ server uses this port for communications. This port must be open to support multiple agents, extensions, backup from snapshot, and restore from backup jobs.

Keep in mind the following:

- If the instance is in a cloud, configure the ports information under required inbound rules for your cloud.
- Once you configure the port when you install Snapshot Manager, you cannot change it when you upgrade.

Preparing Snapshot Manager for backup from snapshot jobs

For backup from snapshot jobs, you must have media server 9.1 or later.

Note: Veritas recommends having swap space enabled on Snapshot Manager's and extensions that would be used to run backup from snapshot jobs for cloud assets. The recommended size for swap space must be greater than or equal to 1.5 times of the system memory. In scenarios where swap space enablement is not available, it is recommend to have systems with higher memory configuration.

Note: (*For AKS only*) To enable swap space on Azure Kubernetes cluster for NetBackup installation and Snapshot Manager deployment on kubernetes based extensions, follow the steps mentioned in Customize node configuration for Azure Kubernetes Service (AKS) node pools.

Required ports:

- Port required on NetBackup primary server: 1556 and 443
- Ports required on NetBackup media server for client side deduplication: 10082 and 10102

If you use private names for installing certificates and communicating with NetBackup, which have to be resolved using `/etc/hosts` follow these steps:

- Add entries similar to `/etc/hosts` file in the */cloudpoint/openv/etc/hosts* file.
- Ensure that you use the private name during Snapshot Manager installation, as well as Snapshot Manager registration.

Chapter 3

Deploying NetBackup Snapshot Manager using container images

This chapter includes the following topics:

- Before you begin installing Snapshot Manager
- Installing Snapshot Manager in the Docker/Podman environment
- Verifying that Snapshot Manager is installed successfully
- Restarting Snapshot Manager

Before you begin installing Snapshot Manager

Ensure that you complete the following before installing Snapshot Manager:

- Decide where to install Snapshot Manager.
  See “Deciding where to run Snapshot Manager” on page 12.

  Note: If you plan to install Snapshot Manager on multiple hosts, read this section carefully and understand the implications of this approach.

- Ensure that your environment meets system requirements.
  See “ Meeting system requirements” on page 17.
- Create the instance on which you install Snapshot Manager or prepare the physical host.

See “Creating an instance or preparing the host to install Snapshot Manager” on page 30.

- Install a container platform
  See Table 2-10 on page 30.
- Create and mount a volume to store Snapshot Manager data.
  See “Creating and mounting a volume to store Snapshot Manager data” on page 32.
- Verify that specific ports are open on the instance or physical host.
  See “Verifying that specific ports are open on the instance or physical host” on page 33.

Note: RedHat 8.x has replaced the Docker ecosystem with the Podman ecosystem. For RHEL 7.x hosts See “Installing Snapshot Manager in the Docker/Podman environment” on page 38.

Installing Snapshot Manager in the Docker/Podman environment

Note: When you deploy Snapshot Manager, you may want to copy the commands below and paste them in your command line interface. If you do, replace the information in these examples that is different from your own: the product and build version, the download directory path, and so on.

Snapshot Manager installation prerequisites on Podman:

- Run the following commands to install the required packages (`lvm2`, `udev` and `plugins`) on the hosts:

```
#yum install -y lvm2-<version>
#yum install -y lvm2-libs-<version>
#yum install -y python3-pyudev-<version>
#yum install -y systemd-udev-<version>
#yum install -y podman-plugins
```

Installing Snapshot Manager

Perform the following appropriate steps depending on the Docker or Podman environment.

To install Snapshot Manager

1. Download the Snapshot Manager image to the system on which you want to deploy Snapshot Manager. Navigate to the Veritas Support site.

   Note: You must log on to the support site to download.

   From the **Products** drop-down, select **NetBackup** and select the required version from the **Version** drop-down. Click **Explore**. Click **Base and upgrade** installers.

   The Snapshot Manager image name resembles the following format for Docker and Podman environment:

   ```
   NetBackup_SnapshotManager_<version>.tar.gz
   ```

   Note: The actual file name may vary depending on the release version.

2. Un-tar the image file and list the contents:

   ```
   # ls
   NetBackup_SnapshotManager_10.1.x.x.xxxx.tar.gz
   netbackup-flexsnap-10.1.x.x.xxxx.tar.gz
   flexsnap_preinstall.sh
   ```

3. Run the following command to prepare the Snapshot Manager host for installation:

   ```
   # sudo ./flexsnap_preinstall.sh
   ```

   (*For Podman on RHEL 8.x*) The output resembles as follows:

   ```
   NetBackup Snapshot Manager for installation:
   Validate SELINUX                          ...  done
   Check for Podman installation             ...  done
   Validate Podman version support           ...  done
   Checking for required packages            ...  done
   Validate Podman services health           ...  done
   Removing deprecated services              ...  done
   Loading Snapshot Manager service images   ...  done
   ```

4. Perform the following appropriate step depending on the Docker or Podman environment.

   - *(For Docker environment)*

Type the following command to run the Snapshot Manager container:

```
# sudo docker run -it --rm -u 0
-v /cloudpoint:/cloudpoint:/-v /var/run/docker.sock:/var/run/
docker.sock veritas/flexsnap-deploy:
<version> install
```

- *(For Podman environment)*
  Install NetBackup Snapshot Manager with the following command:

```
# podman run -it --rm -u 0
-v /cloudpoint:/cloudpoint -v /run/podman/podman.sock:/run/podman/
podman.sock veritas/flexsnap-deploy:
<version> install
```

Note: The command mentioned in the above step is a single command. Ensure that you enter the command without any line breaks.

If the Snapshot Manager host is behind a proxy server, use the following command instead:

- *(For Docker environment)*

```
# sudo docker run -it --rm -u 0
-v /<full_path_to_volume_name>:/<full_path_to_volume_name>
-e VX_HTTP_PROXY=<http_proxy_value>
-e VX_HTTPS_PROXY=<http_proxy_value>
-e VX_NO_PROXY=<no_proxy_value>
-v /var/run/docker.sock:/var/run/docker.sock
veritas/flexsnap-deploy:<version> install
```

- *(For Podman environment)*

```
# podman run -it --rm -u 0
-v /<full_path_to_volume_name>:/<full_path_to_volume_name>
-e VX_HTTP_PROXY=<http_proxy_value>
-e VX_HTTPS_PROXY=<http_proxy_value>
-e VX_NO_PROXY=<no_proxy_value>
-v /run/podman/podman.sock:/run/podman/podman.sock
veritas/flexsnap-deploy:<version> install
```

Replace the following parameters as per your environment:

Parameter	Description
<full_path_to_volume_name>	Represents the path to the Snapshot Manager data volume, which typically is `/cloudpoint`.
<version>	Represents the Snapshot Manager product version that you noted in the earlier step.
Following parameters are required only if the instance uses a proxy server	
<http_proxy_value>	Represents the value to be used as the HTTP proxy for all connections. For example, `"http://proxy.mycompany.com:8080/"`.
<https_proxy_value>	Represents the value to be used as the HTTPS proxy for all connections. For example, `"https://proxy.mycompany.com:8080/"`.
<no_proxy_value>	Represents the addresses that are allowed to bypass the proxy server. You can specify host names, IP addresses, and domain names in this parameter. Use commas to separate multiple entries. For example, `"localhost,mycompany.com,192.168.0.10:80"`. **Note:** If Snapshot Manager is being deployed in the cloud, ensure that you set the following respective values in this parameter: ■ For an AWS instance: 169.254.169.254 ■ For a GCP virtual machine: 169.254.169.254,metadata,metadata.google.internal ■ For an Azure virtual machine: 169.254.169.254 Snapshot Manager uses these addresses to gather instance metadata from the instance metadata service.

Example

- *(For Docker environment)* If the Snapshot Manager version is `10.1.xxxx`, the command syntax is as follows:

```
# sudo docker run -it --rm -u 0 -v /cloudpoint:/cloudpoint
-v /var/run/docker.sock:/var/run/docker.sock
veritas/flexsnap-deploy:10.0.1.xxxx install
```

If using a proxy server, then using the examples provided in the table earlier, the command syntax is as follows:

```
# sudo docker run -it --rm -u 0
-v /cloudpoint:/cloudpoint -e VX_HTTP_PROXY="http://proxy.mycompany.com:80
-e VX_HTTPS_PROXY="http://proxy.mycompany.com:8080/"
-e VX_NO_PROXY="localhost,mycompany.com,192.168.0.10:80"
-v /var/run/docker.sock:/var/run/docker.sock veritas/
flexsnap-deploy:10.0.1.xxxx install
```

The installer displays messages similar to the following:

```
Installing the services

Configuration started at time: Thu Jun  9 07:49:00 UTC 2022
docker server version: 20.10.12

This is a fresh install of NetBackup Snapshot Manager 10.1.x.x.xxxx
Snapshot Manager currently is not configured. Starting initial services
before configuration.
Creating network: flexsnap-network ...done
Starting container: flexsnap-fluentd ...done
Starting container: flexsnap-ipv6config ...done
Creating container: flexsnap-mongodb ...done

Creating container: flexsnap-rabbitmq ...done
Creating container: flexsnap-certauth ...done
Creating container: flexsnap-api-gateway ...done
Creating container: flexsnap-coordinator ...done
Creating container: flexsnap-listener ...done
Creating container: flexsnap-agent ...done
Creating container: flexsnap-onhostagent ...done
Creating container: flexsnap-scheduler ...done
Creating container: flexsnap-policy ...done
Creating container: flexsnap-notification ...done
Creating container: flexsnap-idm ...done
Starting container: flexsnap-config ...done
Creating self signed keys and certs for nginx ...done

Please provide Snapshot Manager admin credentials for configuration:
Admin username: admin
Admin password:
Confirm Admin password:
Host names for TLS certificate (space or comma separated):10.244.79.36
Port (default:443):

Starting container: flexsnap-nginx ...done
```

```
Configuring admin credentials ...done
Waiting for Snapshot Manager configuration to complete (22/22)...done
Configuration complete at time Thu Jun  9 07:54:00 UTC 2022!
Please register Snapshot Manager with NetBackup primary server
```

- *(For Podman environment)*
  The output resembles the following:

```
Installing the services
Configuration started at time: Thu Jun  9 08:42:41 UTC 2022
podman server version: 4.0.2

This is a fresh install of NetBackup Snapshot Manager 10.0.1.0.10014
Snapshot Manager currently is not configured. Starting initial services
before configuration.

Creating network: flexsnap-network ...done
Starting container: flexsnap-fluentd ...done
Creating container: flexsnap-mongodb ...done
Creating container: flexsnap-rabbitmq ...done
Creating container: flexsnap-certauth ...done
Creating container: flexsnap-api-gateway ...done
Creating container: flexsnap-coordinator ...done
Creating container: flexsnap-listener ...done
Creating container: flexsnap-agent ...done
Creating container: flexsnap-onhostagent ...done
Creating container: flexsnap-scheduler ...done
Creating container: flexsnap-policy ...done
Creating container: flexsnap-notification ...done
Creating container: flexsnap-idm ...done
Starting container: flexsnap-config ...done
Creating self signed keys and certs for nginx ...done

Please provide Snapshot Manager admin credentials for configuration:
Admin username: admin
Admin password:
Confirm Admin password:
Host names for TLS certificate (space or comma separated):10.239.154.24
Port (default:443):

Starting container: flexsnap-nginx ...done
Configuring admin credentials ...done
Waiting for Snapshot Manager configuration to complete (22/22)...done
```

```
Configuration complete at time Thu Jun  9 08:52:04 UTC 2022!
Please register Snapshot Manager with NetBackup primary server
```

In this step, Snapshot Manager does the following:

- Creates and runs the containers for each of the Snapshot Manager services.
- Creates self-signed keys and certificates for `nginx`.

Note: If you do not specify the volume as `-v` *`full_path_to_volume_name:/full_path_to_volume_name`*, the container writes to the Docker/Podman host file system.

---

5 Provide the following details when prompted on the command prompt:

Parameter	Description
Admin username	Specify a user name for the Snapshot Manager administrator user account.
Admin password	Specify a password for the administrator user.
Confirm Admin password	Confirm the administrator user password.
Host name for TLS certificate	Specify the IP address or the Fully Qualified Domain Name (FQDN) of the Snapshot Manager host. If you connect to the host using different names (for example, myserver, myserver.mydomain, or myserver.mydomain.mycompany.com), then ensure that you add all the names here if you want to enable Snapshot Manager acce ss using those names. Use commas to specify multiple entries. The names you specify here must point to the same Snapshot Manager host. The specified names or IP address are added to the list of host names to use for configuring Snapshot Manager. The installer uses these names to generate a server certificate for the Snapshot Manager host.
Port	Specify the port through which the Snapshot Manager can communicate. Default is port 443.

The installer then displays messages similar to the following:

```
Configuring admin credentials ...done
Waiting for Snapshot Manager configuration to complete (22/22) ...done
Configuration complete at time Thu Jun 9 06:15:43 UTC 2022!
```

6 This concludes the Snapshot Manager deployment process. The next step is to register the Snapshot Manager with the Veritas NetBackup primary server.

If Snapshot Manager is deployed in the cloud, refer to the *NetBackup Web UI Cloud Administrator's Guide* for instructions. If Snapshot Manager is deployed on-premise, refer to the *NetBackup Snapshot Manager for Data Center Administrator's Guide* for instructions.

Note: If you ever need to restart Snapshot Manager, use the `docker run` command so that your environmental data is preserved.

See “Restarting Snapshot Manager” on page 48.

Verifying that Snapshot Manager is installed successfully

Verify that Snapshot Manager is installed successfully by doing one of the following on the physical machine or the instance command line:

- Verify that a similar success message is displayed at the command prompt.

  ```
  Configuration complete at time Fri Mar 13 06:15:43 UTC 2020!
  ```

- Run the following command and verify that the Snapshot Manager services are running and the status is displayed as `UP`:

  For Docker environment: `# sudo docker ps -a`

  For Podman environment: `# podman ps -a`

  The command output resembles the following:

  ```
  CONTAINER ID   IMAGE     CREATED       STATUS
  076d3c2252fb veritas/
  flexsnap-core:10.0.1.0.10014 system 3 days ago Up 3 days ago
  flexsnap-workflow-system-0-min
  07df8d5d083e veritas/
  flexsnap-rabbitmq:10.0.1.0.10014 3 days ago Up 3 days ago
  flexsnap-rabbitmq
  1d30b1922dad veritas/
  flexsnap-core:10.0.1.0.10014 3 days ago Up 3 days ago
  flexsnap-onhostagent
  4ecca5996401 veritas/
  flexsnap-core:10.0.1.0.10014 3 days ago Up 3 days ago
  flexsnap-notification
  5c2763afe3bd veritas/
  flexsnap-nginx:10.0.1.0.10014 3 days ago Up 3 days ago
  0.0.0.0:443->443/tcp flexsnap-nginx
  5d5805787cda veritas/
  flexsnap-core:10.0.1.0.10014 3 days ago Up 3 days ago
  flexsnap-coordinator
  64ebf4083dbd veritas/
  flexsnap-deploy:10.0.1.0.10014 3 days ago Exited (15) 3 days ago
  ```

```
flexsnap-config
6ca231fc35c2 veritas/
flexsnap-certauth:10.0.1.0.10014 3 days ago Up 3 days ago
flexsnap-certauth
7356cabbb486 veritas/
flexsnap-core:10.0.1.0.10014 3 days ago Up 3 days ago
flexsnap-agent
756ba92314fb veritas/
flexsnap-mongodb:10.0.1.0.10014 3 days ago Up 3 days ago
flexsnap-mongodb
79b7ad032fb7 veritas/
flexsnap-core:10.0.1.0.10014 general 3 days ago Up 3 days ago
flexsnap-workflow-general-0-min
9018a4a7cb08 veritas/
flexsnap-core:10.0.1.0.10014 indexing general 3 days ago Up 3 days ago
flexsnap-workflow-indexing-0-min
b9db2708f7f6 veritas/
flexsnap-core:10.0.1.0.10014 3 days ago Up 3 days ago
flexsnap-policy
cb3e69c27ab1 veritas/
flexsnap-idm:10.0.1.0.10014 3 days ago Up 3 days ago
flexsnap-idm
d25d774ed2e8 veritas/
flexsnap-core:10.0.1.0.10014 3 days ago Up 3 days ago
flexsnap-scheduler
d58206a3c3d7 veritas/
flexsnap-api-gateway:10.0.1.0.10014 3 days ago Up 3 days ago
0.0.0.0:8472->8472/tcp flexsnap-api-gateway
f522cedea280 veritas/
flexsnap-core:10.0.1.0.10014 3 days ago Up 3 days ago
flexsnap-listener
feced68604cc veritas/
flexsnap-fluentd:10.0.1.0.10014 3 days ago Up 3 days ago
0.0.0.0:24224->24224/tcp flexsnap-fluentd
```

Note: The number (`10.0.1.0.10014`) displayed in the image name column represents the Snapshot Manager version. The version may vary depending on the actual product version being installed.

The command output displayed here may be truncated to fit the view. The actual output may include additional details such as container names and ports used.

Restarting Snapshot Manager

If you need to restart Snapshot Manager, it's important that you restart it correctly so that your environmental data is preserved.

To restart Snapshot Manager in the Docker environment

Warning: Do not use commands such as `docker restart` or `docker stop` and `docker start` to restart Snapshot Manager. Use the `docker run` command described below.

- On the instance where Snapshot Manager is installed, enter the following command:

  ```
  # sudo docker run -it --rm -u 0
  -v /cloudpoint:/cloudpoint
  -v /var/run/docker.sock:/var/run/docker.sock
  veritas/flexsnap-deploy:<version> restart
  ```

  Here, *version* represents the currently installed Snapshot Manager product version.

  For example:

  ```
  # sudo docker run -it -rm -u 0
  -v /cloudpoint:/cloudpoint
  -v /var/run/docker.sock:/var/run/docker.sock
  veritas/flexsnap-deploy:10.1.x.xxxx restart
  ```

  Note: Ensure that you enter the command without any line breaks.

To restart Snapshot Manager in the Podman environment

1. First, stop the Snapshot Manager by using the following command on the instance where Snapshot Manager is installed:

```
# podman run -it --rm -u 0 -v /cloudpoint:/cloudpoint
-v /run/podman/podman.sock:/run/podman/podman.sock
veritas/flexsnap-deploy:<version> stop
```

2. Then, start it again by using the following command:

```
# podman run -it --rm -u 0 -v /cloudpoint:/cloudpoint
-v /run/podman/podman.sock:/run/podman/podman.sock
veritas/flexsnap-deploy:<version> start
```

Note: Ensure that you enter the commands without any line breaks.

Chapter 4

Deploying NetBackup Snapshot Manager extensions

This chapter includes the following topics:

- Before you begin installing Snapshot Manager extensions
- Downloading the Snapshot Manager extension
- Installing the Snapshot Manager extension on a VM
- Installing the Snapshot Manager extension on a managed Kubernetes cluster (AKS) in Azure
- Installing the Snapshot Manager extension on a managed Kubernetes cluster (EKS) in AWS
- Installing the Snapshot Manager extension on a managed Kubernetes cluster (GKE) in GCP
- Install extension using the Kustomize and CR YAMLs
- Managing the extensions

Before you begin installing Snapshot Manager extensions

The Snapshot Manager extensions which can be installed on a VM or a managed Kubernetes cluster, can elastically scale out the compute infrastructure to service a large number of jobs, and then scale in as well when the jobs have completed.

Refer to the following appropriate preparatory steps for installing Snapshot Manager that also apply for installing Snapshot Manager extensions.

For a VM based extension

- Decide where to install Snapshot Manager extension.
  See “Deciding where to run Snapshot Manager” on page 12.
- Ensure that your environment meets system requirements.
  See “ Meeting system requirements” on page 17.
- Create the instance or prepare the VM on which you want to install the Snapshot Manager extension.
  See “Creating an instance or preparing the host to install Snapshot Manager” on page 30.
- Install Docker on the VM or the instance on which you want to deploy the extension.
  See Table 2-10 on page 30.
- Create and mount a volume to store Snapshot Manager data. For a VM based extension, the volume size can be 30 GB.
  See “Creating and mounting a volume to store Snapshot Manager data” on page 32.
- Verify that specific ports are open on the instance or the main Snapshot Manager host and ensure that the hosts being protected are reachable from the extensions on required ports. Port 5671 and 443 needs to be opened for RabbitMQ communication on the Snapshot Manager host.

About the extension installation and configuration process

For a Kubernetes based extension

- *For Azure*: The Snapshot Manager cloud-based extension can be deployed on a managed Kubernetes cluster in Azure for scaling the capacity of the Snapshot Manager host to service a large number of requests concurrently. For more information on preparing the host and the managed Kubernetes cluster in Azure:
  See “Prerequisites to install the extension on a managed Kubernetes cluster in Azure” on page 58.
- *For AWS*: The Snapshot Manager cloud-based extension can be deployed on a managed Kubernetes cluster in AWS for scaling the capacity of the Snapshot Manager host to service a large number of requests concurrently. For more information on preparing the host and the managed Kubernetes cluster in AWS:
  See “Prerequisites to install the extension on a managed Kubernetes cluster in AWS” on page 66.

- *For GCP*: The Snapshot Manager cloud-based extension can be deployed on a managed Kubernetes cluster in GCP (GKE) for scaling the capacity of the Snapshot Manager host to service a large number of requests concurrently. For more information on preparing the host and the managed Kubernetes cluster in GCP:

  See “Prerequisites to install the extension on a managed Kubernetes cluster in GCP” on page 74.

About the extension installation and configuration process

To install and configure the Snapshot Manager extension, perform tasks from the NetBackup user interface in your browser and on the command line interface of your local computer or the application host.

See “Installing the extension on a VM” on page 55.

See “Installing the Snapshot Manager extension on a managed Kubernetes cluster (AKS) in Azure” on page 57.

See “Installing the Snapshot Manager extension on a managed Kubernetes cluster (EKS) in AWS” on page 66.

See “Installing the extension on GCP (GKE)” on page 76.

Downloading the Snapshot Manager extension

To download the extension

1. Sign in to the NetBackup Web UI.
2. From the left navigation pane, click **Cloud** and then select the **Snapshot Manager** tab.

   All the Snapshot Manager servers that are registered with the primary server are displayed in this pane.
3. From the desired Snapshot Manager row, click the actions icon on the right and then select **Add extension**.

Note: For the VM-based extension you do not need to download the extension. Proceed directly to steps 7 and 8 to copy the token.

---

4 If you are installing the extension on a managed Kubernetes cluster, then on the **Add extension** dialog box, click the *download* hyperlink.

This launches a new web browser tab.

Do not close the **Add extension** dialog box yet. When you configure the extension, you will return to this dialog box to generate the validation token.

5 Switch to the new browser tab that opened and from the Add extension card, click **Download**. The extension file `nbu_flexsnap_extension.tar` will be downloaded.

6 Copy the downloaded file to the Snapshot Manager host, and untar it by running the `tar -xvf nbu_flexsnap_extension.tar` command .

See “Installing the extension on Azure (AKS)” on page 60.

See “Installing the extension on AWS (EKS)” on page 68.

See “Installing the extension on GCP (GKE)” on page 76.

7 Then to generate the validation token, on the **Add extension** dialog box, click **Create Token**

8 Click **Copy Token** to copy the displayed token. Then provide it on the command prompt while configuring the extension.

Note: The token is valid for 180 seconds only. If you do not use the token within that time frame, generate a new token.

Installing the Snapshot Manager extension on a VM

Note: Currently, the extension is supported only on the Azure Stack Hub environment.

Prerequisites to install the extension on VM

- Choose the Snapshot Manager image supported on Ubuntu or RHEL system that meets the Snapshot Manager installation requirements and create a host. See “Creating an instance or preparing the host to install Snapshot Manager” on page 30.
- Verify that you can connect to the host through a remote desktop.

See “Verifying that specific ports are open on the instance or physical host” on page 33.

- Install Docker or Podman container platforms on the host.
  See Table 2-10 on page 30.
- Download the OS-specific Snapshot Manager image from the Veritas support site.
  The Snapshot Manager image name resembles the following format for Docker and Podman environment:

  ```
  NetBackup_SnapshotManager_<version>.tar.gz
  ```

  Run the following command to prepare the Snapshot Manager host for installation:

  ```
  # sudo ./flexsnap_preinstall.sh
  ```

  Note: The actual file name varies depending on the release version.

- For the VM based extension installed on a RHEL OS the SElinux mode should be "*permissive*".
- Network Security Groups used by the host that is being protected should allow communication from the host where the extension is installed, on the specified ports.

Installing the extension on a VM

Before you install the Snapshot Manager extension on a VM, see Prerequisites to install the extension on VM.

To install the extension

1. Depending on the environment, run the following respective command:
   - *For docker environment*:

     ```
     # sudo docker run -it --rm -u 0
     -v /<full_path_to_volume_name>:/<full_path_to_volume_name>
     -v /var/run/docker.sock:/var/run/docker.sock
     veritas/flexsnap-deploy:<version> install_extension
     ```

   - *For podman environment*:

     ```
     # podman run -it --rm -u 0 --privileged
     -v /<full_path_to_volume_name>:/<full_path_to_volume_name>
     ```

```
-v /run/podman/podman.sock:/run/podman/podman.sock
veritas/flexsnap-deploy:<version> install_extension
```

Note: This is a single command without any line breaks.

In this step, Snapshot Manager does the following:

- Creates and runs the containers for each of the Snapshot Manager services.
- Creates self-signed keys and certificates for `nginx`.

2 Navigate to the NetBackup Web UI and follow the steps 7 and 8 described in the section *Downloading Snapshot Manager extension* to generate and copy the validation token.

See “Downloading the Snapshot Manager extension” on page 53.

Note: For the VM-based extension you do not need to download the extension. Proceed directly to steps 7 and 8 to copy the token.

3 Provide the following configuration parameters when prompted:

Parameter	Description
IP address / FQDN	Provide IP address or FQDN of the main Snapshot Manager host.
Token	Paste the token obtained in the previous step.
Extension Name Identifier	Name of the extension identifier to be visible on the NetBackup UI.

The installer then displays messages similar to the following:

```
Starting docker container: flexsnap-fluentd ...done
Starting docker container: flexsnap-ipv6config ...done
Starting docker container: flexsnap-listener ...done
```

This concludes the Snapshot Manager extension installation on a VM.

To verify that the extension is installed successfully:

- Verify that the success message is displayed at the command prompt.
- Verify that the extension is listed on the NetBackup Web UI.

Go to **Cloud** > **Snapshot Manager** tab > click **Advanced settings** > go to **Snapshot Manager extensions** tab and verify.

- Run the following command and verify that the Snapshot Manager containers are running and the status is displayed as `UP`:

  ```
  # sudo docker ps -a
  ```

  The command output resembles the following:

```
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
e67550304195 veritas/flexsnap-core:9.1.x.x.xxxx "/usr/bin/flexsnap-w…"
13 minutes ago Up 13 minutes flexsnap-core-system-b17e4dd9f6b04d41a08e3a638cd91f61-0
26472ebc6d39 veritas/flexsnap-core:9.1.x.x.xxxx "/usr/bin/flexsnap-w…"
13 minutes ago Up 13 minutes flexsnap-core-general-b17e4dd9f6b04d41a08e3a638cd91f61-0
4f24f6acd290 veritas/flexsnap-core:9.1.x.x.xxxx "/usr/bin/flexsnap-l…"
13 minutes ago Up 13 minutes flexsnap-core
4d000f2d117d veritas/flexsnap-:9.1.x.x.xxxx "/root/ipv6_configur…"
13 minutes ago Exited (137) 13 minutes ago flexsnap-deploy
92b5bdf3211c veritas/flexsnap-fluentd:9.1.x.x.xxxx   "/root/flexsnap-flue…"
13 minutes ago Up 13 minutes 5140/tcp, 0.0.0.0:24224->24224/tcp flexsnap-fluentd
db1f0bff1797 veritas/flexsnap-datamover:9.1.x.x.xxxx "/entrypoint.sh -c d…"
13 minutes ago Up 13 minutes flexsnap-datamover.134b6158ea5a443dba3c489d553098c5
c4ae0eb61fb0 veritas/flexsnap-datamover:9.1.x.x.xxxx "/entrypoint.sh -c d…"
13 minutes ago Up 13 minutes flexsnap-datamover.8e25f89f04e74b01b4fe04e7e5bf8644
1bcaa2b646fb veritas/flexsnap-datamover:9.1.x.x.xxxx "/entrypoint.sh -c d…"
13 minutes ago Up 13 minutes flexsnap-datamover.b08591bdde0f445f83f4ada479e6ddfd
```

Installing the Snapshot Manager extension on a managed Kubernetes cluster (AKS) in Azure

The Snapshot Manager cloud-based extension can be deployed on a managed Kubernetes cluster in Azure for scaling the capacity of the Snapshot Manager host to service a large number of requests concurrently.

Note: Veritas does not recommend the registration of kubernetes extensions for Snapshot Manager in Kubernetes cluster.

Overview

- Your Azure managed Kubernetes cluster should already be deployed with appropriate network and configuration settings, and with specific roles. The cluster must be able to communicate with Snapshot Manager.

The required roles are: `Azure Kubernetes Service RBAC Writer, AcrPush, Azure Kubernetes Service Cluster User Role`

For supported Kubernetes versions, refer to the *Snapshot Manager Hardware Compatibility List (HCL)*.

- Use an existing Azure Container Registry or create a new one, and ensure that the managed Kubernetes cluster has access to pull images from the container registry
- A dedicated nodepool for Snapshot Manager workloads needs to be created with manual scaling or 'Autoscaling' enabled in the Azure managed Kubernetes cluster. The autoscaling feature allows your nodepool to scale dynamically by provisioning and de-provisioning the nodes as required automatically.
- Snapshot Manager extension images (`flexsnap-deploy, flexsnap-core, flexsnap-fluentd, flexsnap-datamover`) need to be uploaded to the Azure container registry.

Prerequisites to install the extension on a managed Kubernetes cluster in Azure

- Choose the Snapshot Manager image supported on Ubuntu or RHEL system that meets the Snapshot Manager installation requirements and create a host.
  See “Creating an instance or preparing the host to install Snapshot Manager” on page 30.
- It is not recommended to scale the cluster up or down when a job is running. It might cause the job to fail. Set the cluster size beforehand.
- Verify that the port 5671 is open on the main Snapshot Manager host.
  See “Verifying that specific ports are open on the instance or physical host” on page 33.
- The public IP of the virtual machine scale set via which the node pool is configured has to be allowed to communicate through port 22, on the workloads being protected.
- Install a Docker or Podman container platform on the host and start the container service.
  See Table 2-10 on page 30.
- Prepare the Snapshot Manager host to access Kubernetes cluster within your Azure environment.
  - Install Azure CLI. For more information, refer to the Azure documentation.
  - Install Kubernetes CLI. For more information, refer to the Kubernetes site.

- Login to the Azure environment to access the Kubernetes cluster by running this command on Azure CLI:

  ```
  # az login --identity
  # az account set --subscription <subscriptionID>
  # az aks get-credentials --resource-group <resource_group_name>
  --name <cluster_name>
  ```

- Ensure that you create an Azure Container Registry or use the existing one if available, to which the Snapshot Manager images will be pushed (uploaded). See Azure documentation.
- To run the `kubectl` and container registry commands from the host system, assign the following role permissions to your VM and cluster. You can assign a 'Contributor', 'Owner', or any custom role that grants full access to manage all resources.
  - Navigate to your Virtual Machine > click **Identity** on the left > under **System assigned** tab, turn the **Status** to 'ON' > click **Azure role assignment** > click **Add role assignments** > select **Scope** as 'Subscription' or 'Resource Group' > select **Role** and assign the following roles : Azure Kubernetes Service RBAC Writer, AcrPush, Azure Kubernetes Service Cluster User Role, and **Save**.
  - Navigate to your Kubernetes cluster > click **Access Control (IAM)** on the left > click **Add role assignments** > select **Role** as 'Contributor ' > Select **Assign access to** as 'Virtual Machines' > select your VM from the drop-down and **Save**.
- Create a storage account in the same subscription and region your Kubernetes cluster is in, and create a file share into it. (Follow the default settings by Azure.) For more information, see Azure documentation.
- While defining **StorageClass** consider using CSI provisioner for `Azure Files` with NFS protocol.
  For example,

  ```
  apiVersion: storage.k8s.io/v1
  kind: StorageClass
  metadata:
    name: test-sc
  parameters:
    skuName: Premium_LRS
    protocol: nfs
  provisioner: file.csi.azure.com
  reclaimPolicy: Retain
  volumeBindingMode: WaitForFirstConsumer
  ```

- Create a namespace for Snapshot Manager from the command line interface on host system:

  ```
  # kubectl create namespace cloudpoint-system
  ```

- Then create a new or use an existing managed Kubernetes cluster in Azure, and add a new node pool dedicated for Snapshot Manager use. Configure Autoscaling as per your requirement.

- Ensure that Azure plug-in is configured.
  See “Microsoft Azure plug-in configuration notes” on page 113.

Installing the extension on Azure (AKS)

Before you install the Snapshot Manager extension on a managed Kubernetes cluster (AKS) in Azure:

- See “Downloading the Snapshot Manager extension” on page 53.
- See “Prerequisites to install the extension on a managed Kubernetes cluster in Azure” on page 58.

To install the extension

1. Download the extension script `nbu_flexsnap_extension.tar`.

   See “Downloading the Snapshot Manager extension” on page 53.

   Note: Do not create the authentication token yet, as it is valid only for 180 seconds.

2. If the host from which you want to install the extension is not the same host where your Snapshot Manager is installed, load the Snapshot Manager container images on the extension host (`flexsnap-deploy, flexsnap-core, flexsnap-fluentd, flexsnap-datamover`)

   The image names are in the following format:

   Example: `veritas/flexsnap-deploy`

3. Create image tags to map the source image with the target image, so that you can push the images to the Azure container registry. For more information, see Prerequisites to install the extension on a managed Kubernetes cluster in Azure.

   Gather the following parameters beforehand:

Parameter	Description
`container_registry_path`	To obtain the container registry path, go to your container registry in Azure and from the **Overview** pane, copy the 'Login server'. Example: `mycontainer.azurecr.io`
`tag`	Snapshot Manager image version. Example: `10.1.x.xxxx`

   - To tag the images, run the following command for each image, depending on the container platform running on your host:
     For Docker: `# docker tag source_image:tag target_image:tag`
     For Podman: `# podman tag source_image:tag target_image:tag`
     Where,
     - the source image tag is: `veritas/flexsnap-deploy:tag>`
     - the target image tag is:
       `<container_registry_path>/<source_image_name>:<SnapshotManager_version_tag>`

     Example:

```
# docker tag veritas/flexsnap-deploy:10.1.x.xxxx
mycontainer.azurecr.io/veritas/flexsnap-deploy:10.1.x.xxxx
# docker tag veritas/flexsnap-core:10.1.x.xxxx
mycontainer.azurecr.io/veritas/flexsnap-core:10.1.x.xxxx
# docker tag veritas/flexsnap-fluentd:10.1.x.xxxx
mycontainer.azurecr.io/veritas/flexsnap-fluentd:10.1.x.xxxx
# docker tag veritas/flexsnap-datamover:10.1.x.xxxx
mycontainer.azurecr.io/veritas/flexsnap-datamover:10.1.x.xxxx
```

4. Then to push the images to the container registry, run the following command for each image, depending on the container platform running on your host:

   For Docker: `# docker push target_image:tag`

   For Podman: `# podman push target_image:tag`

   Example:

```
# docker push mycontainer.azurecr.io/veritas/
flexsnap-deploy:10.1.x.xxxx
# docker push mycontainer.azurecr.io/veritas/
flexsnap-core:10.1.x.xxxx
# docker push mycontainer.azurecr.io/veritas/
flexsnap-fluentd:10.1.x.xxxx
# docker push mycontainer.azurecr.io/veritas/
flexsnap-datamover:10.1.x.xxxx
```

5. Once the images are pushed to the container registry, execute the extension script `cp_extension.sh` that was downloaded earlier, from the host where `kubectl` is installed. The script can be executed either by providing all the required input parameters in one command, or in an interactive way where you will be prompted for input.

   Gather the following parameters before running the script:

Parameter	Description
`snapshotmanager_ip`	Provide IP address or FQDN of the main Snapshot Manager host.
`target_image:tag`	Target image tag created for the `flexsnap-deploy` image in step 3. Example: 'mycontainer.azurecr.io/veritas/flexsnap-deploy:10.0.1.0.10014'
`namespace`	Snapshot Manager `namespace` that was created earlier in the preparation steps.

Parameter	Description

`tag_key=tag_val`	`tag_key` and `tag_val` can be retrieved by using these commands:
	1 Get the name of the node:
	`# kubectl get nodes | grep <node_name>`
	2 Get the tag key=value label:
	`# kubectl describe node <node_name> -n <namespace> | grep -i labels`
	Output example: `agentpool=cpuserpool`

`storage_class`	Kubernetes storage class that was created earlier in the preparation steps. Example: `cloudpoint-sc`
`Size in GB`	Volume size to be provisioned as per your scaling requirements.
`workflow_token`	Authentication token created from the NetBackup Web UI - Add extension dialog. See “Downloading the Snapshot Manager extension” on page 53.

Note: While deploying Snapshot Manager Kubernetes extension, create a storage class and provide it as an input to the Snapshot Manager extension installation script. By default file properties are open, hence it is recommended to create storage class by providing custom attributes in order to maintain the file/folder permissions created on extension under `/cloudpoint` directory. For more information, see Create a storage class section of the Azure product documentation.

Run the script as an executable file:

- Permit the script to run as an executable:
  `# chmod +x cp_extension.sh`
- Run the installation command with all the input parameters described in the above table:
  `./cp_extension.sh install -c <snapshotmanager_ip> -i <target_image:tag> -n <namespace> -p <tag_key=tag_val> -f <storage_class> -t <workflow_token>`
  Example:

```
./cp_extension.sh install
Snapshot Manager image repository path. Format=<Login-server/image:tag>:
 cpautomation.azurecr.io/veritas/flexsnap-deploy:10.1.x.xxxx
Snapshot Manager extension namespace: snapshot-manager
```

```
Snapshot Manager IP or fully-qualified domain name: 10.244.79.38
Node group/pool label with format key=value: agentpool=extpool
Storage class name: azurefile
Size in GiB (minimum 30 GiB, Please refer NetBackup Snapshot Manager
Install and Upgrade Guide for PV size): 50
Snapshot Manager extension token:
This is a fresh NetBackup Snapshot Manager Extension Installation

Starting Snapshot Manager service deployment
customresourcedefinition.apiextensions.k8s.io/
cloudpoint-servers.veritas.com unchanged
serviceaccount/cloudpoint-acc created
clusterrole.rbac.authorization.k8s.io/cloudpoint-cloudpoint-yj created
clusterrolebinding.rbac.authorization.k8s.io/
cloudpoint-rolebinding-cloudpoint-yj created
deployment.apps/flexsnap-operator created
Snapshot Manager service deployment ...done

Generating Snapshot Manager Custom Resource Definition object
Waiting for deployment "flexsnap-operator" rollout to finish:
0 of 1 updated replicas are available...
deployment "flexsnap-operator" successfully rolled out
cloudpointrule.veritas.com/cloudpoint-config-rule created
Snapshot Manager extension installation ...
Operator operations passed
Waiting for all components to come up ...Done
Waiting for all components to come up ...Done
```

Run the script as an interactive file:

- Run the following command:

  ```
  # ./cp_extension.sh install
  ```

- When the script runs, provide the input parameters as described in the above table:

```
./cp_extension.sh install
Snapshot Manager image repository path. Format=<Login-server/image:tag>:
cpautomation.azurecr.io/veritas/flexsnap-deploy:10.1.x.xxxx
Snapshot Manager extension namespace: snapshot-manager
Snapshot Manager IP or fully-qualified domain name: 10.244.79.38
Node group/pool label with format key=value: agentpool=extpool
Storage class name: azurefile
```

```
Size in GiB (minimum 30 GiB, Please refer NetBackup Snapshot Manager
Install and Upgrade Guide for PV size): 50
Snapshot Manager extension token:
This is a fresh NetBackup Snapshot Manager Extension Installation

Starting Snapshot Manager service deployment
customresourcedefinition.apiextensions.k8s.io/
cloudpoint-servers.veritas.com unchanged
serviceaccount/cloudpoint-acc created
clusterrole.rbac.authorization.k8s.io/
cloudpoint-cloudpoint-yj created
clusterrolebinding.rbac.authorization.k8s.io/
cloudpoint-rolebinding-cloudpoint-yj created
deployment.apps/flexsnap-operator created
Snapshot Manager service deployment ...done

Generating Snapshot Manager Custom Resource Definition object
Waiting for deployment "flexsnap-operator" rollout to finish:
 0 of 1 updated replicas are available...
deployment "flexsnap-operator" successfully rolled out
cloudpointrule.veritas.com/cloudpoint-config-rule created
Snapshot Manager extension installation ...
Operator operations passed
Waiting for all components to come up ...Done
Waiting for all components to come up ...Done
```

Note: The output examples have been formatted to fit the screen.

This concludes the Snapshot Manager extension installation on a managed Kubernetes cluster (in Azure cloud).

To verify that the extension is installed successfully:

- Verify that the success message is displayed at the command prompt.
- Verify that the extension is listed on the NetBackup Web UI.
  Go to **Cloud** > **Snapshot Manager** tab > click **Advanced settings** > go to **Snapshot Manager extensions** tab and verify.
- Run the following command and verify that there are five pods, namely, `flexsnap-deploy-xxx`, `flexsnap-fluentd-xxx`, `flexsnap-listener-xxx`, `flexsnap-fluentd-collector-xxx` and `flexsnap-datamover-xxxx` are in Running state:

```
# kubectl get pods -n <namespace>
```

Example: `# kubectl get pods -n cloudpoint-system`

Installing the Snapshot Manager extension on a managed Kubernetes cluster (EKS) in AWS

The Snapshot Manager cloud-based extension can be deployed on a managed Kubernetes cluster in AWS for scaling the capacity of the Snapshot Manager host to service a large number of requests concurrently.

Overview

- Your AWS managed Kubernetes cluster should already be deployed with appropriate network and configuration settings, and with specific roles. The cluster must be able to communicate with Snapshot Manager.
  The required roles are: `AmazonEKSClusterPolicy AmazonEKSWorkerNodePolicy AmazonEC2ContainerRegistryReadOnly AmazonEKS_CNI_Policy AmazonEKSServicePolicy`
  For supported Kubernetes versions, refer to the *Snapshot Manager Hardware Compatibility List (HCL)*.
- Use an existing AWS Elastic Container Registry or create a new one, and ensure that the EKS has access to pull images from the elastic container registry.
- A dedicated nodepool for Snapshot Manager workloads needs to be created in AWS managed Kubernetes cluster. The nodepool uses AWS autoscaling group feature which allows your nodepool to scale dynamically by provisioning and de-provisioning the nodes as required automatically.
- Snapshot Manager extension images (`flexsnap-cloudpoint, flexsnap-listener, flexsnap-workflow, flexsnap-fluentd, flexsnap-datamover`) need to be uploaded to the AWS container registry.

Prerequisites to install the extension on a managed Kubernetes cluster in AWS

- Choose the Snapshot Manager image supported on Ubuntu or RHEL system that meets the Snapshot Manager installation requirements and create a host.
  See “Creating an instance or preparing the host to install Snapshot Manager” on page 30.
- Verify that the port 5671 is open on the main Snapshot Manager host.
  See “Verifying that specific ports are open on the instance or physical host” on page 33.

- Install a Docker or Podman container platform on the host and start the container service.
  See Table 2-10 on page 30.
- It is not recommended to scale the cluster up or down when a job is running. It might cause the job to fail. Set the cluster size beforehand.
- Prepare the Snapshot Manager host to access Kubernetes cluster within your AWS environment.
  - Install AWS CLI. For more information, refer to the AWS Command Line Interface.
  - Install Kubernetes CLI. For more information, refer to the Installing kubectl documentation.
  - Create an AWS Container Registry or use the existing one if available, to which the Snapshot Manager images will be pushed (uploaded). Configure the minimum and maximum nodes as per the requirement.
    For more information, refer to the AWS documentation Amazon Elastic Container Registry documentation.
  - Create the OIDC provider for the AWS EKS cluster. For more information, refer to the Create an IAM OIDC provider for your cluster section of the Amazon EKS User Guide.
  - Create an IAM service account for the AWS EKS cluster. For more information, refer to the Amazon EKS User Guide.
  - If an IAM role needs an access to the EKS cluster, run the following command from the system that already has access to the EKS cluster:
    ```
    kubectl edit -n kube-system configmap/aws-auth
    ```
    For more information, refer to the Enabling IAM user and role access to your cluster section of the Amazon EKS User Guide.
  - Install Amazon EFS driver. For more information, refer to the Amazon EFS CSI driver section of the Amazon EKS User Guide.
  - Login to the AWS environment to access the Kubernetes cluster by running this command on AWS CLI:
    ```
    # aws eks --region <region_name> update-kubeconfig --name
    <cluster_name>
    ```
- Create a storage class. For more information, refer to the Storage classes section of the Amazon EKS User Guide.
- Create a namespace for Snapshot Manager from the command line on host system:
  ```
  # kubectl create namespace cloudpoint-system
  ```

- Then create a new or use an existing managed Kubernetes cluster in AWS, and add a new node pool dedicated for Snapshot Manager use. Configure Autoscaling as per your requirement.

Installing the extension on AWS (EKS)

Before you install the Snapshot Manager extension:

- See “Prerequisites to install the extension on a managed Kubernetes cluster in AWS” on page 66.
- See “Downloading the Snapshot Manager extension” on page 53.

To install the extension

1. The extension file `nbu_flexsnap_extension.tar` must be downloaded beforehand.

   See “Downloading the Snapshot Manager extension” on page 53.

   Note: Do not create the authentication token yet, as it is valid only for 180 seconds.

2. If the host from which you want to install the extension is not the same host where your Snapshot Manager is installed, load the Snapshot Manager container images on the extension host (`flexsnap-deploy, flexsnap-core, flexsnap-fluentd, flexsnap-datamover`)

   The image names are in the following format:

   Example: `veritas/flexsnap-deploy`

3. Create image tags to map the source image with the target image, so that you can push the images to the AWS container registry.

   See “Prerequisites to install the extension on a managed Kubernetes cluster in AWS” on page 66.

   Gather the following parameters beforehand:

Parameter	Description
container_registry_path	To obtain the container registry path, go to your Amazon ECR and copy the URI of each repo. Example: <accoint_id>.dkr.ecr.us-east-2.amazonaws.com/veritas/flexsnap-datamover

Parameter	Description
tag	Snapshot Manager image version. Example: `10.1.x.xxxx`

- To tag the images, run the following command for each image, depending on the container platform running on your host:
  For Docker: `# docker tag source_image:tag target_image:tag`
  For Podman: `# podman tag source_image:tag target_image:tag`
  Where,
  - the source image tag is: `veritas/flexsnap-deploy:tag>`
  - the target image tag is:
    <container_registry_path>/<source_image_name>:<SnapshotManager_version_tag>
    Example:

```
docker tag veritas/flexsnap-deploy:10.1.x.xxxx
<account_id>.dkr.ecr.us-east-2.amazonaws.com/veritas/flexsnap-deploy:10.1.x.xxxx
docker tag veritas/flexsnap-core:10.1.x.xxxx
<account_id>.dkr.ecr.us-east-2.amazonaws.com/veritas/flexsnap-core:10.1.x.xxxx
docker tag veritas/flexsnap-fluentd:10.1.x.xxxx
<account_id>.dkr.ecr.us-east-2.amazonaws.com/veritas/flexsnap-fluentd:10.1.x.xxxx
docker tag veritas/flexsnap-datamover:10.1.x.xxxx
<account_id>.dkr.ecr.us-east-2.amazonaws.com/veritas/flexsnap-datamover:10.1.x.xxxx
```

4 Then to push the images to the container registry, run the following command for each image, depending on the container platform running on your host:

For Docker: `# docker push target_image:tag`

For Podman: `# podman push target_image:tag`

Example:

```
docker push <account-id>.dkr.ecr.us-east-2.amazonaws.com/veritas/
flexsnap-datamover:10.1.x.xxxx
docker push <account-id>.dkr.ecr.us-east-2.amazonaws.com/veritas/
flexsnap-deploy:10.1.x.xxxx
docker push <account-id>.dkr.ecr.us-east-2.amazonaws.com/veritas/
flexsnap-fluentd:10.1.x.xxxx
docker push <account-id>.dkr.ecr.us-east-2.amazonaws.com/veritas/
flexsnap-core:10.1.x.xxxx
```

Note: The command/output examples may be formatted or truncated to fit the screen.

5 Once the images are pushed to the container registry, you can install the extension using one of the following methods:

- Kustomization and custom resource YAML files: Create and apply the `kustomization.yaml` and `cloudpoint_crd.yaml` files based on the samples provided.
  See “Install extension using the Kustomize and CR YAMLs” on page 81.
- Extension script: Execute the extension script `cp_extension.sh` that is packaged within the 'tar' file that was downloaded earlier. The script can be executed either by providing all the required input parameters in one command, or in an interactive way where you will be prompted for input.
  See “Install extension using the extension script” on page 71.

After following the above instructions, you can verify if the extension was installed successfully.

To verify that the extension is installed successfully:

- Verify that the success message is displayed at the command prompt.
- Verify that the extension is listed on the NetBackup Web UI.
  Navigate to **Cloud** > **Snapshot Manager** tab > click **Advanced settings** > go to **Snapshot Manager extensions** tab and verify.

- Run the following command and verify that there are four pods, namely, `flexsnap-cloudpoint-xxx, flexsnap-fluentd-xxx, flexsnap-listener-xxx, flexsnap-fluentd-collector-xxx` and `flexsnap-datamover-xxxx` are in Running state:

  ```
  # kubectl get pods -n <namespace>
  ```

  Example: `# kubectl get pods -n cloudpoint-system`

Install extension using the extension script

Gather the following parameters before running the extension script:

Parameter	Description
cloudpoint_ip	Specify the Snapshot Manager hostname or IP.
target_image:tag	Target image tag created for the `flexsnap-cloudpoint` image. Example: <account_id>.dkr.ecr.us-east-2.amazonaws.com/veritas/flexsnap-deploy:10.2.0.9129
namespace	The namespace that was created earlier in the preparation steps, in which to deploy Snapshot Manager.
tag_key=tag_val	`tag_key` and `tag_val` are the label key and value pair defined for the node on which you want to install the extension. The label key-value pair can be retrieved by using the command `kubectl describe node <node_name> -n <namespace>` Example: `eks.amazonaws.com/nodegroup=Demo-NG`
storage_class	Kubernetes storage class that was created earlier in the preparation steps. Example: `cloudpoint-sc`
Size in GB	Volume size to be provisioned as per your scaling requirements.
workflow_token	Authentication token created from the NetBackup Web UI - Add extension dialog. See “Downloading the Snapshot Manager extension” on page 53.

Run the script as an executable file:

- Permit the script to run as an executable:

  ```
  # chmod +x cp_extension.sh
  ```

- Run the installation command with all the input parameters described in the above table:

```
./cp_extension.sh install -c <snapshotmanager_ip> -i
<target_image:tag> -n <namespace> -p <tag_key=tag_val> -f
<storage_class> -t <workflow_token>
```

Example:

```
root@access-vm2-dnd:/home/cpuser/cp_ext# ./cp_extension.sh install
Snapshot Manager image repository path. Format=<Login-server/image:tag>: cpscale1.azurecr.io/veritas
Snapshot Manager extension namespace: ext
Snapshot Manager IP or fully-qualified domain name: 10.244.63.154
Node group/pool label with format key=value: agentpool=extpool1
Snapshot Manager extension token:
This is a fresh NetBackup Snapshot Manager Extension Installation

Starting Snapshot Manager service deployment
customresourcedefinition.apiextensions.k8s.io/cloudpoint-servers.veritas.com unchanged
serviceaccount/cloudpoint-acc unchanged
clusterrole.rbac.authorization.k8s.io/cloudpoint-ext unchanged
clusterrolebinding.rbac.authorization.k8s.io/cloudpoint-rolebinding-ext unchanged
deployment.apps/flexsnap-deploy created
Snapshot Manager service deployment ...done

customresourcedefinition.apiextensions.k8s.io/cloudpoint-servers.veritas.com condition met
Generating Snapshot Manager Custom Resource Definition object
Waiting for deployment "flexsnap-deploy" rollout to finish: 0 of 1 updated replicas are available...
deployment "flexsnap-deploy" successfully rolled out
cloudpointrule.veritas.com/cloudpoint-config-rule created
Snapshot Manager extension installation ...done
root@access-vm2-dnd:/home/cpuser/cp_ext# kubectl get pods -n ext
NAME                                        READY   STATUS    RESTARTS   AGE
flexsnap-cloudpoint-d8fb97c49-swp7v         1/1     Running   0          5m53s
flexsnap-fluentd-b6vxz                      1/1     Running   0          5m40s
flexsnap-fluentd-collector-867c9cf776-q58bw  1/1     Running   0          5m40s
flexsnap-listener-6f9f5cf7fd-9bsm4          1/1     Running   0          5m40s
```

Run the script as an interactive file:

- Run the following command:
  ```
  # ./cp_extension.sh install
  ```
- When the script runs, provide the input parameters as described in the above table:

Example:

```
Snapshot Manager image repository path. Format=<Login-server/image:tag>:
<account-id>.dkr.ecr.us-east-2.amazonaws.com/veritas/flexsnap-deploy:10.2.0.9129
Snapshot Manager extension namespace: cloudpoint-system
Snapshot Manager IP or fully-qualified domain name: 18.117.***.***
Node pool with format key=value: eks.amazonaws.com/nodegroup=td-nodepool-dnd
Snapshot Manager extension token:
This is a fresh NetBackup Snapshot Manager Extension Installation


Getting Snapshot Manager service file ...done
Getting Snapshot Manager CRD file ...done

Starting Snapshot Manager service deployment
namespace/cloudpoint-system configured
deployment.apps/flexsnap-deploy created
serviceaccount/cloudpoint-acc created

clusterrole.rbac.authorization.k8s.io/cloudpoint-cloudpoint-system unchanged
clusterrolebinding.rbac.authorization.k8s.io/cloudpoint-rolebinding-cloudpoint-system unchanged
customresourcedefinition.apiextensions.k8s.io/cloudpoint-servers.veritas.com created
Snapshot Manager service deployment ...done

customresourcedefinition.apiextensions.k8s.io/cloudpoint-servers.veritas.com condition met
Generating Snapshot Manager Custom Resource Definition object
cloudpointrule.veritas.com/cloudpoint-config-rule created
Snapshot Manager extension installation ...done
```

Note: The output examples may be formatted or truncated to fit the screen.

Installing the Snapshot Manager extension on a managed Kubernetes cluster (GKE) in GCP

Following are the permissions required for configuring the Google Kubernetes Engine (GKE) cluster:

- For pushing the images to google container registry, user must have the write permissions for cloud bucket storage. The `storage.admin` role covers all the required permissions.
  For more information on pushing the images, see Pushing images to a registry in your project.

- The user must have the **cluster-admin** IAM role assigned to it to configure the Kubernetes extension.

  For more information on the role based access control, see Define permissions using Roles or ClusterRoles.
- Account associated with GCP provider configuration must have the following permissions for GKE based Kubernetes extension operations:
  - Permissions for cluster access:

    `container.clusters.get`
  - Permissions for auto scale feature:

    `compute.instanceGroupManagers.get`

    `compute.instanceGroupManagers.update`

    `container.clusters.get`

    `container.clusters.update`

    `container.operations.get`

Prerequisites to install the extension on a managed Kubernetes cluster in GCP

The Snapshot Manager cloud-based extension can be deployed on a managed Kubernetes cluster in GCP for scaling the capacity of the Snapshot Manager host to service a large number of requests concurrently.

- The GCP managed Kubernetes cluster must be already deployed with appropriate network and configuration settings. The cluster must be able to communicate with Snapshot Manager and the filestore.

  Note: The Snapshot Manager and all the cluster nodepools must be in the same zone.

  For more information, see Google Kubernetes Engine overview.
- Use an existing container registry or create a new one, and ensure that the managed Kubernetes cluster has access to pull images from the container registry.
- A dedicated nodepool for Snapshot Manager workloads must be created with or without **Autoscaling** enabled in the GKE cluster. The autoscaling feature allows your nodepool to scale dynamically by provisioning and de-provisioning the nodes as required automatically.
- Snapshot Manager extension images (flexsnap-core, flexsnap-datamover , flexsnap-deploy, flexsnap-fluentd) must be uploaded to the container registry.

Prepare the host and the managed Kubernetes cluster in GCP

- Select the Snapshot Manager image supported on Ubuntu or RHEL system that meets the Snapshot Manager installation requirements and create a host.
  See “Creating an instance or preparing the host to install Snapshot Manager” on page 30.
- Verify that the port 5671 is open on the main Snapshot Manager host.
  See “Verifying that specific ports are open on the instance or physical host” on page 33.
- Install a docker or podman container platform on the host and start the container service.
  See “Installing container platform (Docker, Podman)” on page 30.
- Prepare the Snapshot Manager host to access Kubernetes cluster within your GCP environment.
  - Install gcloud CLI. For more information, see Install the gcloud CLI.
  - Install Kubernetes CLI.
    For more information, refer to the following documents:
    Install kubectl and configure cluster access
    Install and Set Up kubectl on Linux
  - Create a gcr container registry or use the existing one if available, to which the Snapshot Manager images will be uploaded (pushed).
    Container Registry overview.
  - Run the `gcloud init` to set the account. Ensure that this account has the required permissions to configure the Kubernetes cluster.
    For more information on the required permissions, see Installing the Snapshot Manager extension on a managed Kubernetes cluster (GKE) in GCP. For more information on `gcloud` command, refer to the following document:
    gcloud init
  - Connect to the cluster using the following command:
    ```
    gcloud container clusters get-credentials <cluster-name> --zone
    <zone-name> --project <project-name>
    ```
    For more information, refer to Install kubectl and configure cluster access.
  - Create a namespace for Snapshot Manager from the command line on host system:
    ```
    # kubectl create namespace <namespace-name>
    # kubectl config set-context --current
    --namespace=<namespace-name>
    ```

Note: User can provide any namespace name, it must be like `cloudpoint-system`.

Create a persistent volume

- Reuse existing filestore.
  Mount the filestore and create a directory (for example, dir_for_this_cp) only to be used by Snapshot Manager.
- Create a file (for example, `PV_file.yaml`) with the content as follows:

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: <name of the pv>
spec:
  capacity:
    storage: <size in GB>
  accessModes:
  - ReadWriteMany
  nfs:
    path: <path to the dir created above>
    server: <ip of the filestore>
```

  Run the following command to setup Persistent Volume:

```
kubectl apply -f <PV_file.yaml>
```

  For more information about using file store with kubernetes cluster, refer to Accessing file shares from Google Kubernetes Engine clusters.

Installing the extension on GCP (GKE)

Before you install the Snapshot Manager extension on a managed Kubernetes cluster (GKE) in GCP:

- See “Downloading the Snapshot Manager extension” on page 53.
- See “Prerequisites to install the extension on a managed Kubernetes cluster in GCP” on page 74.

To install the extension

1. Download the extension script `nbu_flexsnap_extension.tar`.

   See “Downloading the Snapshot Manager extension” on page 53.

   Note: Do not create the authentication token yet, as it is valid only for 180 seconds.

2. If the host from which you want to install the extension is not the same host where your Snapshot Manager is installed, load the Snapshot Manager container images on the extension host (`flexsnap-deploy, flexsnap-core, flexsnap-fluentd, flexsnap-datamover`)

   The image names are in the following format:

   Example: `veritas/flexsnap-deploy`

3. Tag the images to map the source image with the target image, so that you can push the images to the GCP container registry.

   Gather the following parameters beforehand:

Parameter	Description
container_registry_path	To obtain the container registry path, go to your container registry in GCP and from the **Overview** pane, copy the 'Login server'. Example: `gcr.io/<project-name>/<dir>`
tag	Snapshot Manager image version. Example: `10.1.x.xxxx`

   - To tag the images, run the following command for each image, depending on the container platform running on your host:
     For Docker: `# docker tag source_image:tag target_image:tag`
     For Podman: `# podman tag source_image:tag target_image:tag`
     Where,
     - the source image tag is: `veritas/flexsnap-deploy:tag>`
     - the target image tag is:
       `<container_registry_path>/<source_image_name>:<SnapshotManager_version_tag>`
       Example:

       ```
       # docker tag veritas/flexsnap-deploy:10.1.x.xxxx gcr.io/<project-nam
       veritas/flexsnap-deploy:10.1.x.xxxx
       ```

```
# docker tag veritas/flexsnap-core:10.1.x.xxxx gcr.io/<project-name>/
veritas/flexsnap-listener:10.1.x.xxxx
# docker tag veritas/flexsnap-fluentd:10.1.x.xxxx gcr.io/<project-name>/
veritas/flexsnap-fluentd:10.1.x.xxxx
# docker tag veritas/flexsnap-datamover:10.1.x.xxxx gcr.io/<project-name>
veritas/flexsnap-datamover:10.1.x.xxxx
```

4 To push the images to the container registry, run the following command for each image, depending on the container platform running on your host:

For Docker: `# docker push target_image:tag`

For Podman: `# podman push target_image:tag`

Example:

```
# docker push gcr.io/<project-name>/veritas/flexsnap-deploy:10.1.x.xxxx
# docker push gcr.io/<project-name>/veritas/flexsnap-core:10.1.x.xxxx
# docker push gcr.io/<project-name>/veritas/flexsnap-fluentd:10.1.x.xxxx
# docker push gcr.io/<project-name>/veritas/flexsnap-datamover:10.1.x.xxxx
```

5 Finally, run the script `cp_extension.sh` that was downloaded earlier.

See “Downloading the Snapshot Manager extension” on page 53.

The script can be executed either by providing all the required input parameters in one command, or in an interactive way where you will be prompted for input.

Gather the following parameters before running the script:

Parameter	Description
cloudpoint_ip	Provide IP address or FQDN of the main Snapshot Manager host.
target_image:tag	Target image tag created for the `flexsnap-deploy` image in step 3. Example: gcr.io/<project-name>/veritas/flexsnap-deploy:10.1.x.xxxx
namespace	Snapshot Manager `namespace` that was created earlier in the preparation steps.
tag_key=tag_val	`tag_key` and `tag_val` can be retrieved by using the following command: `# gcloud container node-pools list --cluster=<cluster-name> --zone=<zone-name>`

Parameter	Description
persistent_volume	Kubernetes persistent volume that was created earlier in the preparation steps.
Size in GiB	Volume size to be provisioned as per your scaling requirements.
workflow_token	Authentication token created from the NetBackup Web UI - Add extension dialog. See "Downloading the Snapshot Manager extension" on page 53.

Note: While deploying Snapshot Manager Kubernetes extension, create a persistent volume and provide it as an input to the Snapshot Manager extension installation script.

Run the script as an executable file:

- Permit the script to run as an executable:
  ```
  # chmod +x cp_extension.sh
  ```
- Run the installation command with all the input parameters described in the above table:
  ```
  ./cp_extension.sh install -c <cloudpoint-ip> -i
  <target-image:tag> -n <namespace> -p
  cloud.google.com/gke-nodepool=<nodepool-name> -v
  <persistent-volume-name> -k <size-in-GiB> -t <token>
  ```
  Example:

  ```
  ./cp_extension.sh install
  Snapshot Manager image repository path. Format=<Login-server/image:tag>
  <project-name>/veritas/flexsnap-deploy:10.0.1.0.10012
  Snapshot Manager extension namespace: <namespace-name>
  Snapshot Manager IP or fully-qualified domain name: xx.xxx.xx.xx
  Node group/pool label with format key=value: agentpool=extpool
  Persistent volume name:
  Size in GiB (minimum 30 GiB,
  Please refer NetBackup Snapshot Manager Install and Upgrade Guide for I
  Snapshot Manager extension token:
  This is a fresh NetBackup Snapshot Manager Extension Installation

  Starting Snapshot Manager service deployment
  customresourcedefinition.apiextensions.k8s.io/cloudpoint-servers.verita
  ```

```
serviceaccount/cloudpoint-acc created
clusterrole.rbac.authorization.k8s.io/cloudpoint-cloudpoint-yj created
clusterrolebinding.rbac.authorization.k8s.io/cloudpoint-rolebinding-cloudp
deployment.apps/flexsnap-operator created
Snapshot Manager service deployment ...done

Generating Snapshot Manager Custom Resource Definition object
Waiting for deployment "flexsnap-operator" rollout to finish:0 of 1 update
deployment "flexsnap-operator" successfully rolled out
cloudpointrule.veritas.com/cloudpoint-config-rule created
Snapshot Manager extension installation ...
Operator operations passed
Waiting for all components to come up ...Done
Waiting for all components to come up ...Done
```

Run the script as an interactive file:

- Run the following command:
  ```
  # ./cp_extension.sh install
  ```
- When the script runs, provide the input parameters as described in the above table:

```
./cp_extension.sh install
Snapshot Manager image repository path. Format=<Login-server/image:tag>: cpau
<project-name>/veritas/flexsnap-deploy:10.1.x.xxxx
Snapshot Manager extension namespace: snapshot-manager
Snapshot Manager IP or fully-qualified domain name: xx.xxx.xx.xx
Node group/pool label with format key=value: agentpool=extpool
Persistent volume name:
Size in GiB (minimum 30 GiB,
Please refer NetBackup Snapshot Manager Install and Upgrade Guide for PV size
Snapshot Manager extension token:
This is a fresh NetBackup Snapshot Manager Extension Installation

Starting Snapshot Manager service deployment
customresourcedefinition.apiextensions.k8s.io/cloudpoint-servers.veritas.com
serviceaccount/cloudpoint-acc created
clusterrole.rbac.authorization.k8s.io/cloudpoint-cloudpoint-yj created
clusterrolebinding.rbac.authorization.k8s.io/cloudpoint-rolebinding-cloudpoin
deployment.apps/flexsnap-operator created
Snapshot Manager service deployment ...done

Generating Snapshot Manager Custom Resource Definition object
```

```
Waiting for deployment "flexsnap-operator" rollout to finish:0 of 1 updat
deployment "flexsnap-operator" successfully rolled out
cloudpointrule.veritas.com/cloudpoint-config-rule created
Snapshot Manager extension installation ...
Operator operations passed
Waiting for all components to come up ...Done
Waiting for all components to come up ...Done
```

Note: The output examples have been formatted to fit the screen.

This concludes the Snapshot Manager extension installation on a managed Kubernetes cluster (in GCP).

To verify that the extension is installed successfully:

- Verify that the success message is displayed at the command prompt.
- Verify that the extension is listed on the NetBackup Web UI.
  Go to **Cloud** > **Snapshot Manager** tab > click **Advanced settings** > go to **Snapshot Manager extensions** tab and verify.
- Run the following command and verify that there are four pods, namely, `flexsnap-deploy-xxx, flexsnap-fluentd-xxx, flexsnap-listener-xxx` and `flexsnap-fluentd-collector-xxx` are in Running state:
  `# kubectl get pods -n <namespace>`
  Example: `# kubectl get pods -n cloudpoint-system`
  The `flexsnap-datamover-xxxx` pod will not run by-default after deployment, it will get created only if backup operation is triggered.

Install extension using the Kustomize and CR YAMLs

The extension folder contains the following samples based on which you need to create new YAMLs with the relevant values as per your environment:

- kustomization.yaml
- cloudpoint_crd.yaml
- node_select.yaml
- cloudpoint_service.yaml

kustomization.yaml

In the `kustomization.yaml`, update the parameters in the **Image** section with relevant values as described in the following table.

Parameter	Description
newName	Specify the Snapshot Manager image name, along with the container registry path. Example: <account_id>.dkr.ecr.us-east-2.amazonaws.com/veritas/flexsnap-deploy
newTag	Specify the tag of the Snapshot Manager image to be deployed. Example: `10.2.0.9129`
namespace	The namespace that was created earlier in the preparation steps, in which to deploy Snapshot Manager.

Example:

```
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization
resources:
- cloudpoint_service.yaml
patchesStrategicMerge:
- node_select.yaml
namespace: demo-cloudpoint-ns
images:
- name: CLOUDPOINT_IMAGE
  newName: <account_id>.dkr.ecr.us-east-2.amazonaws.com/veritas/flexsnap-deploy
  newTag: 10.2.0.9129
vars:
- name: ServiceAccount.cloudpoint-acc.metadata.namespace
  objref:
    kind: ServiceAccount
    name: cloudpoint-acc
    apiVersion: v1
  fieldref:
    fieldpath: metadata.namespace
configurations:
- cloudpoint_kustomize.yaml
```

cloudpoint_service.yaml

If deploying the extension on GCP platform, then in `cloudpoint_service.yaml`, replace the **storageClassName** with **volumeName**.

cloudpoint_crd.yaml

Edit the `cloudpoint_crd.yaml` manifest file as follows:

- For GCP platform: Delete the line with **storageClassName** word in it.
- For Non-GCP platform: Delete the line with **volumeName** word in it.

Now update the parameters in the **Spec** section with relevant values as described in the following table.

Parameter	Description
cloudpointHost	Specify the Snapshot Manager hostname or IP.
cloudpointExtensionToken	Paste the contents of the Snapshot Manager token that was downloaded earlier from NetBackup Web UI - Add extension dialog.
storageClassName	Kubernetes storage class that was created earlier in the preparation steps. Example: `efs-sc-new-root` **Note:** Not required for GCP platform.
size	Volume size in GB to be provisioned as per your scaling requirements.
namespace	The namespace that was created earlier in the preparation steps, in which to deploy Snapshot Manager.
volumeName	The name of the Persistent Volume created earlier in preparation steps. **Note:** Required for GCP platform.

Example:

```
apiVersion: veritas.com/v1
kind: CloudpointRule
metadata:
  name: cloudpoint-config-rule
  namespace: demo-cloudpoint-ns
spec:
  CLOUDPOINT_HOST: 3.17.**.*** .
  CLOUDPOINT_EXTENSION_TOKEN: workflow-3s3tlpwpy62dyoingxqmfeojlky7bub9rbzx8srh8kdgmsqo6f-q85lfl
  RENEW: false
  LOG_STORAGE:
```

```
STORAGE_CLASS_NAME: efs-sc-new-root
SIZE: 100
```

Then run the following commands from the folder where the YAML files are located.

- To apply the Kustomization YAML: `kubectl apply -k <location of the kustomization.yaml file>`
- To apply the Snapshot Manager CR: `kubectl apply -f cloudpoint_crd.yaml`

node_select.yaml

Navigate to **nodeSelector** under the **Spec** section and replace the values of NODE_AFFINITY_KEY and NODE_AFFINITY_VALUE in the `node_select.yaml` file. User can obtain these details using the following commands:

- Use the following command to obtain the name of any node from the dedicated node-pool for our extension:
  `# kubectl get nodes`
- Depending on the specific cloud provider, use the following respective commands based on the the **tag key=value** label:
  - For Azure: `# kubectl describe node <node_name> | grep -i labels`
    Output example: `agentpool=cpuserpool`
  - For AWS: `# kubectl describe node <node_name> | grep -i <node_group_name>`
    Output example: `eks.amazonaws.com/nodegroup=Demo-NG`
  - For GCP: `# kubectl describe node <node_name> | grep -i <node_pool_name>`
    Output example: `cloud.google.com/gke-nodepool=manik-node-pool`

Parameter	Description
NODE_AFFINITY_KEY	■ For AWS: eks.amazonaws.com/nodegroup ■ For Azure: agentpool ■ For GCP: cloud.google.com/gke-nodepool
NODE_AFFINITY_VALUE	Name of the node pool. ■ For AWS: Demo-NG ■ For Azure: cpuserpool ■ For GCP: manik-node-pool

Managing the extensions

After you have installed the VM-based or the managed Kubernetes cluster-based extensions, you may need to disable or enable them, stop, start, or restart them, or renew their certificates.

Refer to the following table that describes how to use these options to manage the extensions.

Table 4-1 Post-installation options for the extensions

Option	Procedure
Disable or enable the extension: ■ VM-based extension ■ Managed Kubernetes cluster-based extension	You can disable or enable the extensions from the NetBackup Web UI Go to **Cloud** > **Snapshot Managers** tab > click **Advanced settings** > go to **Snapshot Manager extensions** tab > then disable or enable the extension as required, and click **Save.** No jobs will be scheduled on the extension that is disabled. **Note:** When Snapshot Manager is upgraded, all the extensions are automatically disabled. Then you need to upgrade the extensions with the same Snapshot Manager version and enable them manually from the NetBackup Web UI.

Stop, start, or restart the VM-based extension
To stop the extension:
	Execute the following commands on the extension host VM to stop/start/restart the extension:
	For Docker:
	`# sudo docker run -it --rm -u 0 -v /<full_path_to_volume_name>:/<full_path_to_volume_name> -v /var/run/docker.sock:/var/run/docker.sock veritas/flexsnap-deploy:<version> stop`
	For Podman
	`# podman run -it --rm -u 0 --privileged -v /<full_path_to_volume_name>:/<full_path_to_volume_name> -v /run/podman/podman.sock:/run/podman/podman.sock veritas/flexsnap-deploy:<version> stop`

Table 4-1 Post-installation options for the extensions *(continued)*

Option	Procedure

To start the extension:	For Docker:
	`# sudo docker run -it --rm -u 0 -v /<full_path_to_volume_name>:/<full_path_to_volume_name> -v /var/run/docker.sock:/var/run/docker.sock veritas/flexsnap-deploy:<version> start`
	For Podman
	`# podman run -it --rm -u 0 --privileged -v /<full_path_to_volume_name>:/<full_path_to_volume_name> -v /run/podman/podman.sock:/run/podman/podman.sock veritas/flexsnap-deploy:<version> start`

To restart the extension:	For Docker:
	`# sudo docker run -it --rm -u 0 -v /<full_path_to_volume_name>:/<full_path_to_volume_name> -v /var/run/docker.sock:/var/run/docker.sock veritas/flexsnap-deploy:<version> restart`
	For Podman
	`# podman run -it --rm -u 0 --privileged -v /<full_path_to_volume_name>:/<full_path_to_volume_name> -v /run/podman/podman.sock:/run/podman/podman.sock veritas/flexsnap-deploy:<version> restart`

Renew certificate for a VM-based extension	**1** Run the following command on the extension host: `# sudo docker run -it --rm -u 0 -v /<full_path_to_volume_name>:/<full_path_to_volume_name> -v /var/run/docker.sock:/var/run/docker.sock veritas/flexsnap-deploy:<version> renew_extension` **2** Then provide the Snapshot Manager IP/FQDN, and the extension token which can be generated from NetBackup Web UI to begin renewal of the certificates. See “Installing the extension on a VM” on page 55.

Table 4-1 Post-installation options for the extensions *(continued)*

Option	Procedure

Renew certificate for a managed Kubernetes cluster-based extension	**1** Download the extension installation script `cp_extension.sh` from the NetBackup Web UI .
	2 Execute the script from the host where `kubectl` is installed. Run the following commands:
	`# chmod +x cp_extension.sh`
	`# ./cp_extension.sh renew`
	3 Then provide the Snapshot Manager IP/FQDN, extension token (which can be generated from NetBackup Web UI), and the extension namespace to begin renewal of the certificates.
	See “Installing the extension on Azure (AKS)” on page 60.

Chapter 5

NetBackup Snapshot Manager cloud plug-ins

This chapter includes the following topics:

- How to configure the Snapshot Manager cloud plug-ins?
- AWS plug-in configuration notes
- Google Cloud Platform plug-in configuration notes
- Microsoft Azure plug-in configuration notes
- Microsoft Azure Stack Hub plug-in configuration notes

How to configure the Snapshot Manager cloud plug-ins?

Snapshot Manager plug-ins are software modules that enable the discovery of your assets in the cloud or in an on-premise environment. After registering the Snapshot Manager with the NetBackup primary server, you must configure the Snapshot Manager plug-ins to be able to protect your workloads using NetBackup.

How you configure the plug-ins depends on the asset type and how Snapshot Manager is deployed. If the Snapshot Manager is deployed in the cloud and you want to protect workloads in the cloud, you must use the NetBackup Web UI to register the Snapshot Manager and configure the Snapshot Manager cloud and application plug-ins. The overall steps to configure the plug-ins are similar, regardless of the asset type. Only the configuration parameters vary.

Refer to the *NetBackup Web UI Cloud Administrator's Guide* for information on how to configure cloud plug-ins.

AWS plug-in configuration notes

The Amazon Web Services (AWS) plug-in lets you create, restore, and delete snapshots of the following assets in an Amazon cloud:

- Elastic Compute Cloud (EC2) instances
- Elastic Block Store (EBS) volumes
- Amazon Relational Database Service (RDS) instances
- Aurora clusters

Note: Before you configure the AWS plug-in, make sure that you have configured the proper permissions so Snapshot Manager can work with your AWS assets.

Snapshot Manager supports the following AWS regions:

Table 5-1 AWS regions supported by Snapshot Manager

AWS commercial regions	AWS GovCloud (US) regions
■ us-east-1, us-east-2, us-west-1, us-west-2 ■ ap-east-1, ap-south-1, ap-northeast-1, ap-northeast-2, ap-southeast-1, ap-southeast-2, ap-southeast-3 ■ eu-central-1, eu-west-1, eu-west-2, eu-west-3, eu-north-1, eu-south-1 Milan, eu-south-1 Cape Town ■ cn-north-1, cn-northwest-1 ■ ca-central-1 ■ me-south-1 ■ sa-east-1	■ us-gov-east-1 ■ us-gov-west-1

The following information is required for configuring the Snapshot Manager plug-in for AWS:

If Snapshot Manager is deployed on a on-premise host or a virtual machine:

Table 5-2 AWS plug-in configuration parameters

Snapshot Manager configuration parameter	AWS equivalent term and description
Access key	The access key ID, when specified with the secret access key, authorizes Snapshot Manager to interact with the AWS APIs.

Table 5-2 AWS plug-in configuration parameters *(continued)*

Snapshot Manager configuration parameter	AWS equivalent term and description
Secret key	The secret access key.
Regions	One or more AWS regions in which to discover cloud assets.

Note: Snapshot Manager encrypts credentials using AES-256 encryption.

If Snapshot Manager is deployed in the AWS cloud:

Table 5-3 AWS plug-in configuration parameters: cloud deployment

Snapshot Manager configuration parameter	Description
For Source Account configuration	
Regions	One or more AWS regions associated with the AWS source account in which to discover cloud assets. **Note:** If you deploy Snapshot Manager using the CloudFormation template (CFT), then the source account is automatically configured as part of the template-based deployment workflow.
For Cross Account configuration	
Account ID	The account ID of the other AWS account (cross account) whose assets you wish to protect using the Snapshot Manager instance configured in the Source Account.
Role Name	The IAM role that is attached to the other AWS account (cross account).
Regions	One or more AWS regions associated with the AWS cross account in which to discover cloud assets.

When Snapshot Manager connects to AWS, it uses the following endpoints. You can use this information to create a allowed list on your firewall.

- ec2.*.amazonaws.com
- sts.amazonaws.com

- rds.*.amazonaws.com
- kms. *.amazonaws.com
- ebs.*.amazonaws.com
- iam.amazonaws.com
- eks.*.amazonaws.com
- autoscaling.*.amazonaws.com
- (For DBPaaS protection) dynamodb.*.amazonaws.com

In addition, you must specify the following resources and actions:

- ec2.SecurityGroup.*
- ec2.Subnet.*
- ec2.Vpc.*
- ec2.createInstance
- ec2.runInstances

Configuring multiple accounts or subscriptions or projects

- If you are creating multiple configurations for the same plug-in, ensure that they manage assets from different Regions. Two or more plug-in configurations should not manage the same set of cloud assets simultaneously.
- When multiple accounts are all managed with a single Snapshot Manager, the number of assets being managed by a single Snapshot Manager instance might get too large and it would be better to space them out.
- To achieve application consistent snapshots, we would require agent/agentless network connections between the remote VM instance and Snapshot Manager. This would require setting up cross account/subscription/project networking.

AWS plug-in considerations and limitations

Before you configure the plug-in, consider the following:

- Snapshot Manager does not support AWS Nitro-based instances that use EBS volumes that are exposed as non-volatile memory express (NVMe) devices.
  To allow Snapshot Manager to discover and protect AWS Nitro-based Windows instances that use NVMe EBS volumes, ensure that the AWS NVMe tool executable file, `ebsnvme-id.exe`, is present in any of the following locations on the AWS Windows instance:
- To allow Snapshot Manager to discover and protect Windows instances created from custom/community AMI.

- AWS NVMe drivers must be installed on custom or community AMIs. See this link.
- Install the `ebsnvme-id.exe` either in `%PROGRAMDATA%\Amazon\Tools` or `%PROGRAMFILES%\Veritas\Cloudpoint`
- Friendly device name must contain the substring "NVMe", or update in Windows registry for all NVMe backed devices.
  Registry path:
  `Computer\HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Enum\SCSI\Disk&Ven_NVMe&Prod_Amazon_Elastic_B\`
  Property Name: `FriendlyName`
  Value: `NVMe Amazon Elastic B SCSI Disk Drive`

- You cannot delete automated snapshots of RDS instances and Aurora clusters through Snapshot Manager.
- The application consistency of AWS RDS applications depend on the behavior of AWS. (AWS suspends I/O while backing up the DB instance).
  This is a limitation from AWS and is currently outside the scope of Snapshot Manager.
- All automated snapshot names start with the pattern `rds:`.
- If you are configuring the plug-in to discover and protect AWS Nitro-based Windows instances that use NVMe EBS volumes, you must ensure that the AWS NVMe tool executable file, `ebsnvme-id.exe`, is present in any of the following locations on the AWS instance:
  - `%PROGRAMDATA%\Amazon\Tools`
    This is the default location for most AWS instances.
  - `%PROGRAMFILES%\Veritas\Cloudpoint`
    Manually download and copy the executable file to this location.
  - System PATH environment variable
    Add or update the executable file path in the system's PATH environment variable.

  If the NVMe tool is not present in one of the mentioned locations, Snapshot Manager may fail to discover the file systems on such instances. You may see the following error in the logs:
  `"ebsnvme-id.exe" not found in expected paths!"`
  This is required for AWS Nitro-based Windows instances only. Also, if the instance is launched using the community AMI or custom AMI, you might need to install the tool manually.

- Snapshot Manager does not support cross-account replication for AWS RDS instances or clusters, if the snapshots are encrypted using the default RDS encryption key (aws/rds). You cannot share such encrypted snapshots between AWS accounts.

  If you try to replicate such snapshots between AWS accounts, the operation fails with the following error:

  ```
  Replication failed The source snapshot KMS key [<key>] does not exist,
  is not enabled or you do not have permissions to access it.
  ```

  This is a limitation from AWS and is currently outside the scope of Snapshot Manager.

- If a region is removed from the AWS plug-in configuration, then all the discovered assets from that region are also removed from the Snapshot Manager assets database. If there are any active snapshots that are associated with the assets that get removed, then you may not be able perform any operations on those snapshots.

  Once you add that region back into the plug-in configuration, Snapshot Manager discovers all the assets again and you can resume operations on the associated snapshots. However, you cannot perform restore operations on the associated snapshots.

- Snapshot Manager supports commercial as well as GovCloud (US) regions. During AWS plug-in configuration, even though you can select a combination of AWS commercial and GovCloud (US) regions, the configuration will eventually fail.

- Snapshot Manager does not support IPv6 addresses for AWS RDS instances. This is a limitation of Amazon RDS itself and is not related to Snapshot Manager. For more information, refer to the AWS documentation.

- Snapshot Manager does not support application consistent snapshots and granular file restores for Windows systems with virtual disks or storage spaces that are created from a storage pool. If a Microsoft SQL server snapshot job uses disks from a storage pool, the job fails with an error. But if a snapshot job for virtual machine which is in a connected state is triggered, the job might be successful. In this case, the file system quiescing and indexing is skipped. The restore job for such an individual disk to original location also fails. In this condition, the host might move to an unrecoverable state and requires a manual recovery.

Prerequisites for configuring the AWS plug-in

If the Snapshot Manager instance is deployed in the AWS cloud, do the following before you configure the plug-in:

- Create an AWS IAM role and assign permissions that are required by Snapshot Manager.

  See “Configuring AWS permissions for Snapshot Manager” on page 96.

  For more information on how to create an IAM role, see AWS Identity and Access Management Documentation.

- Attach the IAM role to the Snapshot Manager instance.

  For more information on how to attach an IAM role, see AWS Identity and Access Management Documentation.

  Note: If you have deployed Snapshot Manager using the CloudFormation Template (CFT), then the IAM role is automatically assigned to the instance when the Snapshot Manager stack is launched.

- For cross account configuration, from the AWS IAM console (IAM Console > Roles), edit the IAM roles such that:

  - A new IAM role is created and assigned to the other AWS account (target account). Also, assign that role a policy that has the required permissions to access the assets in the target AWS account.
  - The IAM role of the other AWS account should trust the Source Account IAM role (**Roles > Trust relationships** tab).
  - The Source Account IAM role is assigned an inline policy (**Roles > Permissions** tab) that allows the source role to assume the role (`"sts:AssumeRole"`) of the other AWS account.
  - The validity of the temporary security credentials that the Source Account IAM role gets when it assumes the Cross Account IAM role is set to 1 hour, at a minimum (**Maximum CLI/API session duration** field).

  See “Before you create a cross account configuration” on page 103.

- If the assets in the AWS cloud are encrypted using AWS KMS Customer Managed Keys (CMK), then you must ensure the following:

  - If using an IAM user for Snapshot Manager plug-in configuration, ensure that the IAM user is added as a key user of the CMK.
  - For source account configuration, ensure that the IAM role that is attached to the Snapshot Managert instance is added as a key user of the CMK.
  - For cross account configuration, ensure that the IAM role that is assigned to the other AWS account (cross account) is added as a key user of the CMK.

Adding these IAM roles and users as the CMK key users allows them to use the AWS KMS CMK key directly for cryptographic operations on the assets. For more details, refer to the AWS documentation.

- If the Snapshot Manager instance has instance metadata service (IMDsv2) enabled, then ensure that the **HttpPutResponseHopLimit** parameter is set to 2 for the VM.
  If the value of **HttpPutResponseHopLimit** parameter is not set to 2, then the AWS calls to fetch the metadata from the Snapshot Manager containers created on the machine fails.
  For more information on the IMDsv2 service, refer to Use IMDSv2.

Configuring AWS permissions for Snapshot Manager

To protect your Amazon Web Services (AWS) assets, Snapshot Manager must first have access to them. You must associate a permission policy with each Snapshot Manager user who wants to work with AWS assets.

Ensure that the user account or role is assigned the minimum permissions required for Snapshot Manager.

See “AWS permissions required by Snapshot Manager” on page 97.

To configure permissions on Amazon Web Services

1. Create or edit an AWS user account from Identity and Access Management (IAM).
2. Perform one of the following.
   - To create a new AWS user account, perform the following:
     - From IAM, select the **Users** pane and click **Add user**.
     - In the **User name** field, enter a name for the new user.
     - Select the **Access** type. This value determines how AWS accesses the permission policy. (This example uses Programmatic access).
     - Select **Next: Permissions**.
     - On the **Set permissions for *username*** screen, select **Attach existing policies directly**.
     - Select the previously created permission policy (shown below) and select **Next: Review**.
     - On the **Permissions summary** page, select **Create user**.
     - Obtain the **Access Key** and **Secret Key** for the newly created user.
   - To edit an AWS user account, perform the following:

- Select **Add permissions**.
- On the **Grant permissions** screen, select **Attach existing policies directly**.
- Select the previously created permission policy (shown below), and select **Next: Review**.
- On the **Permissions summary** screen, select **Add permissions**.

3 To configure the AWS plug-in for the created or edited user, refer to the plug-in configuration notes.

See “AWS plug-in configuration notes” on page 90.

AWS permissions required by Snapshot Manager

The following is a IAM role definition (in JSON format) that gives Snapshot Manager the ability to configure AWS plugin and discover assets, manage the snapshots etc.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EC2AutoScaling",
            "Effect": "Allow",
            "Action": [
                "autoscaling:UpdateAutoScalingGroup",
                "autoscaling:AttachInstances",
                "autoscaling:DescribeScalingActivities",
                "autoscaling:TerminateInstanceInAutoScalingGroup"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "KMS",
            "Effect": "Allow",
            "Action": [
                "kms:ListKeys",
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncryptTo",
```

```
            "kms:DescribeKey",
            "kms:ListAliases",
            "kms:GenerateDataKey",
            "kms:GenerateDataKeyWithoutPlaintext",
            "kms:ReEncryptFrom",
            "kms:CreateGrant"
        ],
        "Resource": [
            "*"
        ]
    },
    {
        "Sid": "RDSBackup",
        "Effect": "Allow",
        "Action": [
            "rds:DescribeDBSnapshots",
            "rds:DescribeDBClusters",
            "rds:DescribeDBClusterSnapshots",
            "rds:DeleteDBSnapshot",
            "rds:CreateDBSnapshot",
            "rds:CreateDBClusterSnapshot",
            "rds:ModifyDBSnapshotAttribute",
            "rds:DescribeDBSubnetGroups",
            "rds:DescribeDBInstances",
            "rds:CopyDBSnapshot",
            "rds:CopyDBClusterSnapshot",
            "rds:DescribeDBSnapshotAttributes",
            "rds:DeleteDBClusterSnapshot",
            "rds:ListTagsForResource",
            "rds:AddTagsToResource"
        ],
        "Resource": [
            "*"
        ]
    },
    {
        "Sid": "RDSRecovery",
        "Effect": "Allow",
        "Action": [
            "rds:ModifyDBInstance",
            "rds:ModifyDBClusterSnapshotAttribute",
            "rds:RestoreDBInstanceFromDBSnapshot",
            "rds:ModifyDBCluster",
```

```
            "rds:RestoreDBClusterFromSnapshot",
            "rds:CreateDBInstance",
            "rds:RestoreDBClusterToPointInTime",
            "rds:CreateDBSecurityGroup",
            "rds:CreateDBCluster",
            "rds:RestoreDBInstanceToPointInTime",
            "rds:DescribeDBClusterParameterGroups"
        ],
        "Resource": [
            "*"
        ]
    },
    {
        "Sid": "EC2Backup",
        "Effect": "Allow",
        "Action": [
            "sts:GetCallerIdentity",
            "ec2:CreateSnapshot",
            "ec2:CreateSnapshots",
            "ec2:DescribeInstances",
            "ec2:DescribeInstanceStatus",
            "ec2:ModifySnapshotAttribute",
            "ec2:CreateImage",
            "ec2:CopyImage",
            "ec2:CopySnapshot",
            "ec2:DescribeSnapshots",
            "ec2:DescribeVolumeStatus",
            "ec2:DescribeVolumes",
            "ec2:RegisterImage",
            "ec2:DescribeVolumeAttribute",
            "ec2:DescribeSubnets",
            "ec2:DescribeVpcs",
            "ec2:DeregisterImage",
            "ec2:DeleteSnapshot",
            "ec2:DescribeInstanceAttribute",
            "ec2:DescribeRegions",
            "ec2:ModifyImageAttribute",
            "ec2:DescribeAvailabilityZones",
            "ec2:ResetSnapshotAttribute",
            "ec2:DescribeHosts",
            "ec2:DescribeImages",
            "ec2:DescribeSecurityGroups" ,
            "ec2:DescribeNetworkInterfaces"
```

```
        ],
        "Resource": [
            "*"
        ]
    },
    {
        "Sid": "EC2Recovery",
        "Effect": "Allow",
        "Action": [
            "ec2:RunInstances",
            "ec2:AttachNetworkInterface",
            "ec2:DetachVolume",
            "ec2:AttachVolume",
            "ec2:DeleteTags",
            "ec2:CreateTags",
            "ec2:StartInstances",
            "ec2:StopInstances",
            "ec2:TerminateInstances",
            "ec2:CreateVolume",
            "ec2:DeleteVolume",
            "ec2:DescribeIamInstanceProfileAssociations",
            "ec2:AssociateIamInstanceProfile",
            "ec2:AssociateAddress",
            "ec2:DescribeKeyPairs",
            "ec2:AuthorizeSecurityGroupEgress",
            "ec2:AuthorizeSecurityGroupIngress",
            "ec2:DescribeInstanceTypeOfferings",
            "ec2:GetEbsEncryptionByDefault"
        ],
        "Resource": [
            "*"
        ]
    },
    {
        "Sid": "EBS",
        "Effect": "Allow",
        "Action": [
            "ebs:ListSnapshotBlocks",
            "ebs:GetSnapshotBlock",
            "ebs:CompleteSnapshot",
            "ebs:PutSnapshotBlock",
            "ebs:ListChangedBlocks"
        ],
```

```
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "EKS",
            "Effect": "Allow",
            "Action": [
                "eks:DescribeNodegroup",
                "eks:DescribeUpdate",
                "eks:UpdateNodegroupConfig",
                "eks:ListClusters"
                "eks:DescribeCluster"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "IAM",
            "Effect": "Allow",
            "Action": [
                "iam:ListAccountAliases",
                "iam:SimulatePrincipalPolicy"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

If a Snapshot Manager extension is installed on a managed Kubernetes cluster in AWS, then enable the following polices for a user account or a role before configuring the plugin:

```
AmazonEKSClusterPolicy
AmazonEKSWorkerNodePolicy
AmazonEC2ContainerRegistryReadOnly
AmazonEKS_CNI_Policy
AmazonEKSServicePolicy
```

Additional IAM permissions required for marketplace deployment

```
{
  "Sid": "AWSMarketplacePermissions",
  "Effect": "Allow",
  "Action": [
     "autoscaling:UpdateAutoScalingGroup",
     "autoscaling:AttachInstances",
     "sns:Publish",
     "sns:GetTopicAttributes",
     "secretsmanager:GetResourcePolicy",
     "secretsmanager:GetSecretValue",
     "secretsmanager:DescribeSecret",
     "secretsmanager:RestoreSecret",
     "secretsmanager:PutSecretValue",
     "secretsmanager:DeleteSecret",
     "secretsmanager:UpdateSecret"
  ],
  "Resource": [
    "*"
  ]
}
```

Additional IAM permissions required by PaaS workloads

```
{
   "Sid": "DynamoDB",
   "Effect": "Allow",
   "Action": [
      "dynamodb:ListTables",
      "dynamodb:DescribeTable",
      "dynamodb:CreateTable",
      "dynamodb:BatchWriteItem",
      "dynamodb:DescribeContinuousBackups",
      "dynamodb:ExportTableToPointInTime",
      "dynamodb:DescribeExport",
      "dynamodb:DeleteTable",
      "dynamodb:UpdateTable",
      "dynamodb:UpdateContinuousBackups"
   ],
   "Resource": [
     "*"
   ]
},
{
  "Sid": "S3Permissions",
```

```
  "Effect": "Allow",
  "Action": [
     "s3:PutObject",
     "s3:GetObject",
     "s3:ListBucket",
     "s3:DeleteObject"
  ],
  "Resource": [
    "*"
   ]
}
```

Before you create a cross account configuration

For Snapshot Manager cross account configuration, you need to perform the following additional tasks before you can create the configuration:

- Create a new IAM role in the other AWS account (target account)
- Create a new policy for the IAM role and ensure that it has required permissions to access the assets in that target AWS account
- Establish a trust relationship between the source and the target AWS accounts
- In the source AWS account, create a policy that allows the IAM role in the source AWS account to assume the IAM role in the target AWS account
- In the target AWS account, set the maximum CLI/API session duration to 1 hour, at a minimum

Perform the following steps:

1. Using the AWS Management Console, create an IAM role in the additional AWS account (the target account) whose assets you want to protect using Snapshot Manager.

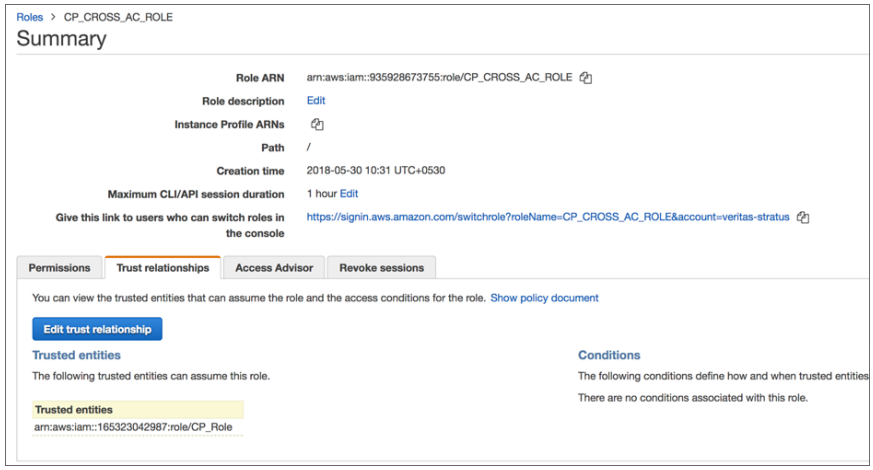
   While creating the IAM role, select the role type as **Another AWS account**.

2. Define a policy for the IAM role that you created in the earlier step.

   Ensure that the policy has the required permissions that allow the IAM role to access all the assets (EC2, RDS, and so on) in the target AWS account.

3 Set up a trust relationship between the source and target AWS accounts.

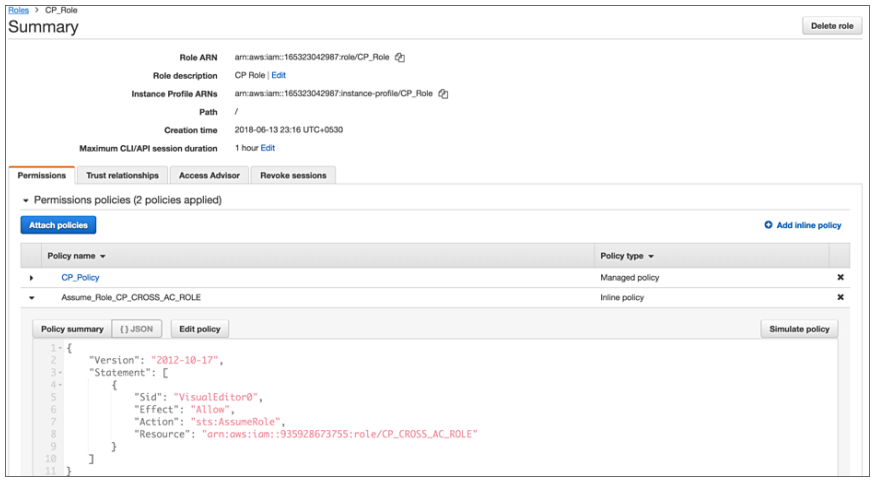
In the target AWS account, edit the trust relationship and specify source account number and source account role.



This action allows only the Snapshot Manager instance hosted in source AWS account to assume the target role using the credentials associated with source account's IAM role. No other entities can assume this role.

4 Grant the source AWS account access to the target role.

In the source AWS account, from the account Summary page, create an inline policy and allow the source AWS account to assume the target role (`"sts:AssumeRole"`).



5 From the target account's Summary page, edit the **Maximum CLI/API session duration** field and set the duration to **1 hour**, at a minimum.

This setting determines the amount of time for which the temporary security credentials that the source account IAM role gets when it assumes target account IAM role remain valid.

Google Cloud Platform plug-in configuration notes

The Google Cloud Platform plug-in lets you create, delete, and restore disk and host-based snapshots in all regions where Google Cloud is present.

Google Cloud Platform plug-in configuration prerequisites

- Before you configure the Google Cloud Platform plug-in, enable the following APIs under **APIs & Services** from Google Cloud console:
  - Cloud Resource Manager API
  - Compute Engine API
- The node pool provided while configuring Kubernetes cluster extension must have all nodes from same region, that is, the node-pool should be single zonal.

- The region of the Snapshot Manager host and node-pool should be same.
- For backup from snapshot use case, Snapshot Manager should be installed in cloud only. A provider must be configured for the zone in which Snapshot Manager is installed. If Snapshot Manager is installed in us-west1-b zone then a provider for us-west1 region must be configured.
- For manual installation (non marketplace) of Snapshot Manager, disable auto-activation of LVM's LV. This can be achieved by setting **auto_activation_volume_list** parameter to empty list or list of specific VG names which must be auto activated. The **auto_activation_volume_list** parameter can be set in `lvm.conf` configuration file.

Google Cloud Platform plug-in configuration parameters

The following parameters are required for configuring the Google Cloud Platform plug-in:

Table 5-4 Google Cloud Platform plug-in configuration parameters

Snapshot Manager configuration parameter	Google equivalent term and description
Project ID	The ID of the project from which the resources are managed. Listed as `project_id` in the JSON file.
Client Email	The email address of the Client ID. Listed as `client_email` in the JSON file.
Private Key	The private key. Listed as `private_key` in the JSON file. **Note:** You must enter this key without quotes (neither single quotes nor double quotes). Do not enter any spaces or return characters at the beginning or end of the key.
Region	A list of regions in which the plug-in operates.

Snapshot Manager supports the following GCP regions:

Table 5-5 GCP regions supported by Snapshot Manager

GCP regions
■ asia-east1 ■ asia-east2 ■ asia-northeast1 ■ asia-northeast2 ■ asia-south1 ■ asia-southeast1
■ australia-southeast1
■ europe-north1 ■ europe-west1 ■ europe-west2 ■ europe-west3 ■ europe-west4 ■ europe-west6
■ northamerica-northeast1 ■ southamerica-east1
■ us-central1 ■ us-east1 ■ us-east4 ■ us-west1 ■ us-west2 ■ us-west3- Utah ■ us-west4 Nevada
■ asia-south ■ australia-southeast2 ■ europe-central2 ■ northamerica-northeast2 ■ southamerica-west1

Configuring multiple accounts or subscriptions or projects

- If you are creating multiple configurations for the same plug-in, ensure that they manage assets from different Regions. Two or more plug-in configurations should not manage the same set of cloud assets simultaneously.

- When multiple accounts are all managed with a single Snapshot Manager, the number of assets being managed by a single Snapshot Manager instance might get too large and it would be better to space them out.
- To achieve application consistent snapshots, we would require agent/agentless network connections between the remote VM instance and Snapshot Manager. This would require setting up cross account/subscription/project networking.

GCP plug-in considerations and limitations

Consider the following before you configure this plug-in:

- If a region is removed from the GCP plug-in configuration, then all the discovered assets from that region are also removed from the Snapshot Manager assets database. If there are any active snapshots that are associated with the assets that get removed, then you may not be able perform any operations on those snapshots.

  Once you add that region back into the plug-in configuration, Snapshot Manager discovers all the assets again and you can resume operations on the associated snapshots. However, you cannot perform any restore operations on the associated snapshots.
- The maximum attachment points on GCP instances are 128 and Snapshot Manager host uses 2 attachment points, which leaves 126 attachment point for backup/restore jobs. So at any point in time Snapshot Manager can backup/restore instance as long as attachment points are available (which is 126 attachment points). If all the attachment points are used, backup/restore jobs start failing with following error message:

  ```
  Failed to attach disk.
  ```
- The maximum number of labels that can be attached to GCP instances are 64 and Snapshot Manager uses 2 labels. If any instance has more than 62 labels, backup/restore may fail

See “Google Cloud Platform permissions required by Snapshot Manager” on page 109.

See “Configuring a GCP service account for Snapshot Manager” on page 111.

See “Preparing the GCP service account for plug-in configuration” on page 112.

Google Cloud Platform permissions required by Snapshot Manager

Assign the following permissions to the service account that Snapshot Manager uses to access assets in the Google Cloud Platform:

```
compute.diskTypes.get
compute.diskTypes.list
```

```
compute.disks.create
compute.disks.createSnapshot
compute.disks.delete
compute.disks.get
compute.disks.list
compute.disks.setLabels
compute.disks.use
compute.globalOperations.get
compute.globalOperations.list
compute.images.get
compute.images.list
compute.instances.attachDisk
compute.instances.create
compute.instances.delete
compute.instances.detachDisk
compute.instances.get
compute.instances.list
compute.instances.setMetadata
compute.instances.setServiceAccount
compute.instances.setLabels
compute.instances.setTags
compute.instances.start
compute.instances.stop
compute.instances.use
compute.machineTypes.get
compute.machineTypes.list
compute.networks.get
compute.networks.list
compute.projects.get
compute.regionOperations.get
compute.regionOperations.list
compute.regions.get
compute.regions.list
compute.snapshots.create
compute.snapshots.delete
compute.snapshots.get
compute.snapshots.list
compute.snapshots.setLabels
compute.snapshots.useReadOnly
compute.subnetworks.get
compute.subnetworks.list
compute.subnetworks.use
compute.subnetworks.useExternalIp
```

```
compute.zoneOperations.get
compute.zoneOperations.list
compute.zones.get
compute.zones.list
iam.serviceAccounts.actAs
resourcemanager.projects.get
```

Additional permissions required by PaaS workloads:

```
cloudsql.databases.list
cloudsql.instances.list
```

Configuring a GCP service account for Snapshot Manager

To protect the assets in Google Cloud Platform (GCP), Snapshot Manager requires permissions to be able to access and perform operations on those cloud assets. You must create a custom role and assign it with the minimum permissions that Snapshot Manager requires. You then associate that custom role with the service account that you created for Snapshot Manager.

Perform the following steps:

1. Create a custom IAM role in GCP. While creating the role, add all the permissions that Snapshot Manager requires.

   See “Google Cloud Platform permissions required by Snapshot Manager” on page 109.

   For more information on creating and managing the custom roles, see Creating and managing custom roles section of Google documentation.

2. Create a service account in GCP.

   Grant the following roles to the service account:

   - The custom IAM role that you created in the earlier step. This is the role that has all the permissions that Snapshot Manager requires to access GCP resources.
   - The `iam.serviceAccountUser` role. This enables the service account to connect to the GCP using the service account context.

   For more information on creating and managing service accounts, see Creating and managing service accounts section of Google documentation.

Note: To use Shared VPC at GCP, additional **Compute Network User** named role assignment is required for the service account used to configure GCP plugin.

Preparing the GCP service account for plug-in configuration

To prepare for the Snapshot Manager GCP plug-in configuration

1. Gather the GCP configuration parameters that Snapshot Manager requires.

   See “Google Cloud Platform plug-in configuration notes” on page 106.

   Do the following:

   - From the Google Cloud console, navigate to **IAM & admin > Service accounts**.
   - Click the assigned service account. Click the three vertical buttons on the right side and select **Create key**.
   - Select **JSON** and click **CREATE**.
   - In the dialog box, click to save the file. This file contains the parameters you need to configure the Google Cloud plug-in. The following is a sample JSON file showing each parameter in context. The `private-key` is truncated for readability.

```
{
  "type": "service_account",
  "project_id": "some-product",
  "private_key": "-----BEGIN PRIVATE KEY-----\n
N11EvA18ADAN89kq4k199w08AQEFAA5C8KYw9951A9EAAo18AQCnvpuJ3oK974z4\n
.
.
.
weT9odE4ryl81tNU\nV3q1XNX4fK55QTpd6CNu+f7QjEw5x8+5ft05DU8ayQcNkX\n
4pXJoDol54N52+T4qV4WkoFD5uL4NLPz5wxf1y\nNWcNfru8K8a2q1/9o0U+99==\n
-----END PRIVATE KEY-----\n",
  "client_email": "email@xyz-product.iam.gserviceaccount.com",
  "auth_uri": "https://accounts.google.com/o/oauth2/auth",
  "token_uri": "https://accounts.google.com/o/oauth2/token",
  "auth_provider_x509_cert_url": "https://www.googleapis.com \
   /oauth2/v1/certs",
  "client_x509_cert_url": "https://www.googleapis.com/robot/v1 \
   /metadata/x509/ email%40xyz-product.iam.gserviceaccount.com"
}
```

2. Using a text editor, reformat the `private_key` so it can be entered in the Snapshot Manager user interface. When you look in the file you created, each line of the private key ends with `\n`. You must replace each instance of `\n` with an actual carriage return. Do one of the following:

- If you are a UNIX administrator, enter the following command in `vi`. In the following example, the `^` indicates the `Ctrl` key. Note that only the `^M` is visible on the command line.
  :g/\\n/s//^V^M/g
- If you are a Windows administrator, use WordPad or a similar editor to search on `\n` and manually replace each instance.

3 When you configure the plug-in from the NetBackup user interface, copy and paste the reformatted private key into the **Private Key** field. The reformatted private_key should look similar to the following:

```
-----BEGIN PRIVATE KEY-----\
N11EvA18ADAN89kq4k199w08AQEFAA5C8KYw9951A9EAAo18AQCnvpuJ3oK974z4
.
.
.
weT9odE4ryl81tNU\nV3q1XNX4fK55QTpd6CNu+f7QjEw5x8+5ft05DU8ayQcNkX
4pXJoDol54N52+T4qV4WkoFD5uL4NLPz5wxf1y\nNWcNfru8K8a2q1/9o0U+99==
-----END PRIVATE KEY-----
```

Microsoft Azure plug-in configuration notes

The Microsoft Azure plug-in lets you create, delete, and restore snapshots at the virtual machine level and the managed disk level.

Before you configure the Azure plug-in, complete the following preparatory steps:

- (*Applicable only if user proceeds with application service principal route*) Use the Microsoft Azure Portal to create an Azure Active Directory (AAD) application for the Azure plug-in.
- Assign the required permissions to a role to access resources.
  For more information on Azure plug-in permissions required by Snapshot Manager, See “Configuring permissions on Microsoft Azure” on page 117.
  In Azure you can assign permissions to the resources by one of the following methods:
  - Service principal: This permission can be assigned to user, group or an application.
  - Managed identity: Managed identities provide an automatically managed identity in Azure Active Directory for applications to use when connecting to resources that support Azure Active Directory (Azure AD) authentication. There are two types of managed identities:

- System-assigned
- User-assigned

For more details, follow the steps mentioned in the Azure documentation.

Table 5-6 Microsoft Azure plug-in configuration parameters

Snapshot Manager configuration parameter	Microsoft equivalent term and description
Credential type: **Application service principal** **Note:** Assign a role to the application service principal.	
Tenant ID	The ID of the Azure AD directory in which you created the application.
Client ID	The application ID.
Secret key	The secret key of the application.
Credential type: **System managed identity**	Enable system managed identity on Snapshot Manager host in Azure. **Note:** Assign a role to the system managed identity.
Credential type: **User managed identity** **Note:** Assign a role to the user managed identity.	
Client ID	The Client ID of the user managed identity connected to the Snapshot Manager host.
Following parameters are applicable for all the above credential type's	
Regions	One or more regions in which to discover cloud assets. **Note:** If you configure a government cloud, select US Gov Arizona, US Gov Texas US, or Gov Virginia.
Resource Group prefix	The prefix used to store the snapshots created for the assets in a different resource group other than the one in which the assets exist. For example, if an asset exists in **Snapshot Manager** and prefix for resource group is **snap**, then snapshots of assets in Snapshot Manager resource group would be stored in **snapSnapshot Manager** resource group.

Table 5-6 Microsoft Azure plug-in configuration parameters *(continued)*

Snapshot Manager configuration parameter	Microsoft equivalent term and description
Protect assets even if prefixed Resource Groups are not found	On selecting this check box, Snapshot Manager would not fail the snapshot operation if resource group does not exists. It tries to store the snapshot in the original resource group. **Note:** The prefixed resource group region must be same as the original resource group region.

Configuring multiple accounts or subscriptions or projects

- If you are creating multiple configurations for the same plug-in, ensure that they manage assets from different Subscriptions. Two or more plug-in configurations should not manage the same set of cloud assets simultaneously.
- When multiple accounts are all managed with a single Snapshot Manager server, the number of assets being managed by a single Snapshot Manager instance might get too large. Hence it would be better to segregate the assets across multiple Snapshot Manager servers for better load balancing.
- To achieve application consistent snapshots, we would require agent/agentless network connections between the remote VM instance and Snapshot Manager server. This would require setting up cross account/subscription/project networking.

Azure plug-in considerations and limitations

Consider the following before you configure the Azure plug-in:

- The current release of the plug-in does not support snapshots of blobs.
- Snapshot Manager currently only supports creating and restoring snapshots of Azure-managed disks and the virtual machines that are backed up by managed disks.
- Snapshot Manager does not support snapshot operations for Ultra SSD disk types in an Azure environment. Even though Snapshot Manager discovers the ultra disks successfully, any snapshot operation that is triggered on such disk assets fails with the following error:

  ```
  Snapshots of UltraSSD_LRS disks are not supported.
  ```

- If you are creating multiple configurations for the same plug-in, ensure that they manage assets from different Tenant IDs. Two or more plug-in configurations should not manage the same set of cloud assets simultaneously.

- When you create snapshots, the Azure plug-in creates an Azure-specific lock object on each of the snapshots. The snapshots are locked to prevent unintended deletion either from the Azure console or from an Azure CLI or API call. The lock object has the same name as that of the snapshot. The lock object also includes a field named "`notes`" that contains the ID of the corresponding VM or asset that the snapshot belongs to.
  Ensure that the `notes` field in the snapshot lock objects is not modified or deleted. Doing so will disassociate the snapshot from its corresponding original asset. The Azure plug-in uses the ID from the `notes` fields of the lock objects to associate the snapshots with the instances whose source disks are either replaced or deleted, for example, as part of the 'Original location' restore operation.
- Azure plug-in supports the following GovCloud (US) regions:
  - US Gov Arizona
  - US Gov Texas
  - US Gov Virginia
  - US Gov Iowa
  - US DoD Central
  - US DoD East
- Azure plug-in supports the following India regions:
  - Jio India West
  - Jio India Central
- Snapshot Manager Azure plug-in does not support the following Azure regions:

Location	Region
US	■ US DoD Central ■ US DoD East ■ US Sec West
China Snapshot Manager does not support any regions in China.	■ China East ■ China East 2 ■ China North ■ China North 2
Germany	■ Germany Central (Sovereign) ■ Germany Northeast (Sovereign)

- Snapshot Manager also supports Microsoft Azure generation 2 type of virtual machines.
- Snapshot Manager does not support application consistent snapshots and granular file restores for Windows systems with virtual disks or storage spaces that are created from a storage pool. If a Microsoft SQL server snapshot job uses disks from a storage pool, the job fails with an error. But if a snapshot job for virtual machine which is in a connected state is triggered, the job might be successful. In this case, the file system quiescing and indexing is skipped. The restore job for such an individual disk to original location also fails. In this condition, the host might move to an unrecoverable state and requires a manual recovery.

Configuring permissions on Microsoft Azure

Before Snapshot Manager can protect your Microsoft Azure assets, it must have access to them. You must associate a custom role that Snapshot Manager users can use to work with Azure assets.

The following is a custom role definition (in JSON format) that gives Snapshot Manager the ability to:

- Configure the Azure plug-in and discover assets.
- Create host and disk snapshots.
- Restore snapshots to the original location or to a new location.
- Delete snapshots.

```
{
    {
        "roleName": "CloudPoint-permissions",
        "description": "Necessary permissions for Azure plug-in operations in
        "assignableScopes": [
            "/subscriptions/<Subscriptions_ID>"
        ],
        "permissions": [
            {
                "actions": [
                    "Microsoft.Storage/*/read",
                    "Microsoft.Compute/*/read",
                    "Microsoft.Sql/*/read",
                    "Microsoft.Compute/disks/write",
                    "Microsoft.Compute/disks/delete",
                    "Microsoft.Compute/disks/beginGetAccess/action",
```

```
                    "Microsoft.Compute/disks/endGetAccess/action",
                    "Microsoft.Compute/snapshots/delete",
                    "Microsoft.Compute/snapshots/write",
                    "Microsoft.Compute/snapshots/beginGetAccess/action",
                    "Microsoft.Compute/snapshots/endGetAccess/action",
                    "Microsoft.Compute/virtualMachines/write",
                    "Microsoft.Compute/virtualMachines/delete",
                    "Microsoft.Compute/virtualMachines/start/action",
                    "Microsoft.Compute/virtualMachines/vmSizes/read",
                    "Microsoft.Compute/virtualMachines/powerOff/action",
                    "Microsoft.Network/*/read",
                    "Microsoft.Network/networkInterfaces/delete",
                    "Microsoft.Network/networkInterfaces/effectiveNetworkSecurity
                    "Microsoft.Network/networkInterfaces/join/action",
                    "Microsoft.Network/networkInterfaces/write",
                    "Microsoft.Network/networkSecurityGroups/join/action",
                    "Microsoft.Network/networkSecurityGroups/write",
                    "Microsoft.Network/publicIPAddresses/delete",
                    "Microsoft.Network/publicIPAddresses/join/action",
                    "Microsoft.Network/publicIPAddresses/write",
                    "Microsoft.Network/virtualNetworks/subnets/join/action",
                    "Microsoft.Resources/*/read",
                    "Microsoft.Resources/subscriptions/tagNames/tagValues/write",
                    "Microsoft.Resources/subscriptions/tagNames/write",
                    "Microsoft.Subscription/*/read",
                    "Microsoft.Authorization/locks/*",
                    "Microsoft.Authorization/*/read",
                    "Microsoft.ContainerService/managedClusters/agentPools/read",
                    "Microsoft.ContainerService/managedClusters/read",
                    "Microsoft.Compute/virtualMachineScaleSets/write",
                    "Microsoft.Compute/virtualMachineScaleSets/delete/action"
                ],
                "notActions": [],
                "dataActions": [],
                "notDataActions": []
            }
        ]
    }
}
```

If Snapshot Manager extension is installed on a managed Kubernetes cluster in Azure, then the following permissions can also be added before configuring the plugin:

```
"Microsoft.ContainerService/managedClusters/agentPools/read",
"Microsoft.ContainerService/managedClusters/read",
"Microsoft.Compute/virtualMachineScaleSets/write",
"Microsoft.Compute/virtualMachineScaleSets/delete/action"
```

Additional permissions required by PaaS workloads:

```
"Microsoft.DBforMySQL/servers/read",
"Microsoft.DBforMySQL/servers/databases/read",
"Microsoft.DBforMySQL/flexibleServers/read",
"Microsoft.DBforMySQL/flexibleServers/databases/read",
"Microsoft.DBforPostgreSQL/servers/read",
"Microsoft.DBforPostgreSQL/servers/databases/read",
"Microsoft.DBforPostgreSQL/flexibleServers/read",
"Microsoft.DBforPostgreSQL/flexibleServers/databases/read",
"Microsoft.Sql/*/write",
"Microsoft.Sql/*/delete"
```

To create a custom role using powershell, follow the steps mentioned in the Azure documentation.

For example:

```
New-AzureRmRoleDefinition -InputFile "C:\CustomRoles\ReaderSupportRole.json"
```

To create a custom role using Azure CLI, follow the steps mentioned in the Azure documentation.

For example:

```
az role definition create --role-definition "~/CustomRoles/
ReaderSupportRole.json"
```

Note: Before creating a role, you must copy the role definition given earlier (text in JSON format) in a .json file and then use that file as the input file. In the sample command displayed earlier, `ReaderSupportRole.json` is used as the input file that contains the role definition text.

To use this role, perform the following:

- Assign the role to an application running in the Azure environment.
- In Snapshot Manager, configure the Azure off-host plug-in with the application's credentials.

See “Microsoft Azure plug-in configuration notes” on page 113.

About Azure snapshots

NetBackup provides support for incremental snapshots in Azure. NetBackup creates the incremental snapshots for new changes to the disks, since the previous snapshot. The snapshots are independent of each other, for example, deletion of one snapshot, does not affect the subsequent snapshot that NetBackup creates. The incremental snapshots significantly reduce the cost of backup by reducing the required disk space, and using the Azure Standard HDD as storage, instead of Premium HDD.

Microsoft Azure Stack Hub plug-in configuration notes

The Microsoft Azure Stack Hub plug-in lets you create, delete, and restore snapshots at the virtual machine level and the managed disk level. You can configure the Azure Stack Hub plugin using AAD or ADFS authentication methods.

Before you configure the Azure Stack Hub plug-in, complete the following preparatory steps:

- Use the Microsoft Azure Stack Portal to create an application in the Azure Active Directory (AAD) if using AAD as the identify provider for the Azure Stack Hub plug-in.
  For more information on your identity provider options, refer to the Azure Stack documentation.
- Assign the service principal to a role that has access to the resources.

For details, follow the steps mentioned in the Azure Stack documentation.

Table 5-7 Azure Stack Hub plug-in configuration parameters using AAD

Snapshot Manager configuration parameter	Microsoft equivalent term and description
Azure Stack Hub Resource Manager endpoint URL	The endpoint URL in the following format, that allows Snapshot Manager to connect with your Azure resources. `https://management.<location>.<FQDN>`
Tenant ID	The ID of the AAD directory in which you created the application.
Client ID	The application ID.
Secret Key	The secret key of the application.

Table 5-8 Azure Stack Hub plug-in configuration parameters using AD FS

Snapshot Manager configuration parameter	Microsoft equivalent term and description
Azure Stack Hub Resource Manager endpoint URL	The endpoint URL in the following format, that allows Snapshot Manager to connect with your Azure resources. `https://management.<location>.<FQDN>`
Tenant ID (optional)	The ID of the AD FS directory in which you created the application.
Client ID	The application ID.
Secret Key	The secret key of the application.
Authentication Resource URL (optional)	The URL where the authentication token is sent to.

Azure Stack Hub plug-in limitations

- The current release of the plug-in does not support snapshots of blobs.
- Snapshot Manager currently only supports creating and restoring snapshots of Azure Stack managed disks and the virtual machines that are backed up by managed disks.
- Snapshot Manager currently only supports creating and restoring snapshots of Azure Stack managed disks and the virtual machines that are deployed using Azure Stack Resource Manager deployment model.
- Rollback restore operation is not supported for Azure Stack VM, because the OS disk swap not supported.
- Disk encryption is not possible with the Snapshot Manager Azure Stack Hub plug-in, because Azure Stack Hub 2008 does not support disk encryption.
- Snapshot Manager does not support disk-based protection for applications that store data on virtual disks or storage spaces that are created from a storage pool. While taking snapshots of such applications, the disk-based option is not available.
- Snapshot Manager does not support snapshot operations for Ultra SSD disk types in an Azure Stack environment.

Azure Stack Hub plug-in considerations

- If you are creating multiple configurations for the same plug-in, ensure that they manage assets from different Tenant IDs. Two or more plug-in configurations should not manage the same set of cloud assets simultaneously.

- When you create snapshots, the Azure Stack Hub plug-in creates an Azure Stack-specific lock object on each of the snapshots. The snapshots are locked to prevent unintended deletion either from the Azure console or from an Azure CLI or API call. The lock object has the same name as that of the snapshot. The lock object also includes a field named "`notes`" that contains the ID of the corresponding VM or asset that the snapshot belongs to.

  You must ensure that the "`notes`" field in the snapshot lock objects is not modified or deleted. Doing so will disassociate the snapshot from its corresponding original asset.

  The Azure Stack Hub plug-in uses the ID from the "`notes`" fields of the lock objects to associate the snapshots with the instances whose source disks are either replaced or deleted, for example, as part of the 'Original location' restore operation.

Configuring permissions on Microsoft Azure Stack Hub

Before Snapshot Manager can protect your Microsoft Azure Stack assets, it must have access to them. You must associate a custom role that Snapshot Manager users can use to work with Azure Stack assets.

The following is a custom role definition (in JSON format) that gives Snapshot Manager the ability to:

- Configure Azure Stack Hub plug-in and discover assets.
- Create host and disk snapshots.
- Restore snapshots to the original location or to a new location.
- Delete snapshots.

```
{ "Name": "CloudPoint Admin",
"IsCustom": true,
"Description": "Necessary permissions for
 Azure Stack Hub plug-in operations in CloudPoint",
"Actions": [
"Microsoft.Storage/*/read",
"Microsoft.Storage/storageAccounts/listKeys/action",
"Microsoft.Storage/storageAccounts/ListAccountSas/action",
"Microsoft.Compute/*/read",
"Microsoft.Compute/disks/write",
"Microsoft.Compute/disks/delete",
"Microsoft.Compute/images/write",
"Microsoft.Compute/images/delete",
"Microsoft.Compute/snapshots/delete",
```

```
"Microsoft.Compute/snapshots/write",
"Microsoft.Compute/snapshots/beginGetAccess/action",
"Microsoft.Compute/snapshots/endGetAccess/action",
"Microsoft.Compute/virtualMachines/capture/action",
"Microsoft.Compute/virtualMachines/write",
"Microsoft.Compute/virtualMachines/delete",
"Microsoft.Compute/virtualMachines/generalize/action",
"Microsoft.Compute/virtualMachines/restart/action",
"Microsoft.Compute/virtualMachines/runCommand/action",
"Microsoft.Compute/virtualMachines/start/action",
"Microsoft.Compute/virtualMachines/vmSizes/read",
"Microsoft.Compute/virtualMachines/powerOff/action",
"Microsoft.Authorization/locks/*",
"Microsoft.Network/*/read",
"Microsoft.Network/networkInterfaces/delete",
"Microsoft.Network/networkInterfaces/effectiveNetworkSecurityGroups/action",
"Microsoft.Network/networkInterfaces/join/action",
"Microsoft.Network/networkInterfaces/write",
"Microsoft.Network/networkSecurityGroups/join/action",
"Microsoft.Network/networkSecurityGroups/securityRules/write",
"Microsoft.Network/networkSecurityGroups/write",
"Microsoft.Network/publicIPAddresses/delete",
"Microsoft.Network/publicIPAddresses/join/action",
"Microsoft.Network/publicIPAddresses/write",
"Microsoft.Network/routeTables/join/action",
"Microsoft.Network/virtualNetworks/delete",
"Microsoft.Network/virtualNetworks/subnets/delete",
"Microsoft.Network/virtualNetworks/subnets/join/action",
"Microsoft.Network/virtualNetworks/write",
"Microsoft.Resources/*/read",
"Microsoft.Resources/subscriptions/resourceGroups/write",
"Microsoft.Resources/subscriptions/resourceGroups/ \
validateMoveResources/action",
"Microsoft.Resources/subscriptions/tagNames/tagValues/write",
"Microsoft.Resources/subscriptions/tagNames/write",
"Microsoft.Subscription/*/read",
"Microsoft.Authorization/*/read" ],
"NotActions": [ ],
"AssignableScopes": [
"/subscriptions/subscription_GUID",
"/subscriptions/subscription_GUID/ \
resourceGroups/myCloudPointGroup" ] }
```

To create a custom role using Powershell, follow the steps mentioned in the Azure Stack documentation.

For example:

```
New-AzRoleDefinition -InputFile "C:\CustomRoles\registrationrole.json"
```

To create a custom role using Azure CLI, follow the steps mentioned in the Azure documentation.

For example:

```
az role definition create --role-definition "~/CustomRoles/
registrationrole.json"
```

Note: Before creating a role, you must copy the role definition (text in JSON format) in a .json file and then use that file as the input file. In the sample command displayed earlier, `registrationrole.json` is used as the input file that contains the role definition text.

To use this role, perform the following:

- Assign the role to an application running in the Azure Stack environment.
- In Snapshot Manager, configure the Azure Stack off-host plug-in with the application's credentials.

See “Microsoft Azure Stack Hub plug-in configuration notes” on page 120.

Configuring staging location for Azure Stack Hub VMs to restore from backup

The Azure Stack Hub requires you to create a container, inside your storage account, and use it as a staging location when you restore from backup images. The staging location is used to stage the unmanaged disks in the container during restores. Once the data is written to the disk, the disks are converted to managed disks. This is a requirement from the Azure Stack Hub platform. This is a mandatory configuration, before you can use Azure Stack Hub with NetBackup.

The `azurestack.conf` file should contain staging location details of the subscription ID, where the VMs are restored. If you plan to restore to any target subscription ID, other than the source subscription ID, then details of the target subscription ID must be present in the `azurestack.conf` file.

If you are using snapshot images for restore, you do not need to create this staging location.

Note: The staging location is specific to the subscription ID, you must create one staging location for each subscription that you are using to restore VMs.

To configure a staging location for a subscription ID:

1. In the Snapshot Manager, navigate to:

   `/cloudpoint/azurestack.conf`, and open the file in a text editor. This file is created, only after you have added Azure Stack Hub as a cloud service provider in NetBackup.

2. Add the following details in the file:

   [subscription/`<subscription ID>`]

   storage_container = `<name of the storage container>`

   storage_account = `/resourceGroup/<name of the resource group where the storage account exists>/storageaccount/<name of storage account>`

   For example:
   `/resourceGroup/Harsha_RG/storageaccount/harshastorageacc`

3. Repeat step 2, for each subscription ID that you are using. Save and close the file.

Chapter 6

NetBackup Snapshot Manager application agents and plug-ins

This chapter includes the following topics:

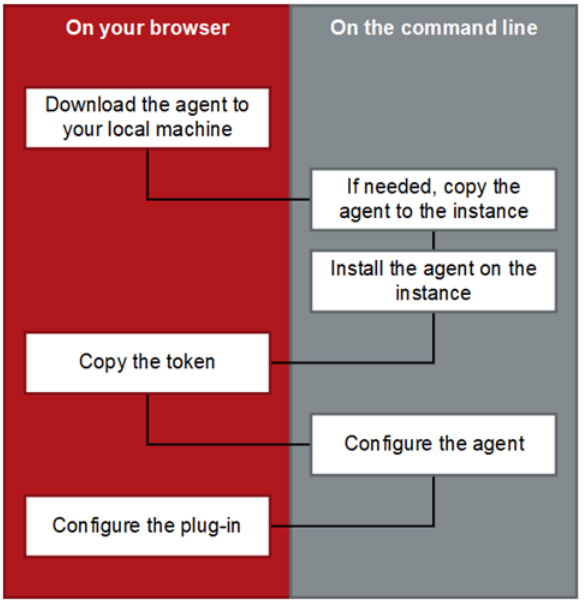
- About the installation and configuration process
- Installing and configuring Snapshot Manager agent
- Configuring the Snapshot Manager application plug-in
- NetBackup protection plan
- Configuring VSS to store shadow copies on the originating drive
- Additional steps required after restoring an AWS RDS database instance

About the installation and configuration process

To install and configure a Snapshot Manager agent and plug-in, use the NetBackup user interface in your browser and on the command line interface of your local computer or the application host.

You can also establish the agent connection using agentless connection mechanism, See “About the agentless feature” on page 155.

Figure 6-1 Snapshot Manager agent installation and configuration process



See “Downloading and installing the Snapshot Manager agent” on page 128.

See “Preparing to install the Windows-based agent” on page 135.

See “Preparing to install the Linux-based agent” on page 131.

Installing and configuring Snapshot Manager agent

This section describes the procedure for downloading, installing and configuring the Snapshot Manager agent.

Downloading and installing the Snapshot Manager agent

Download and install the appropriate Snapshot Manager agent depending on the application that you wish to protect. Whether you install the Linux-based agent or the Windows-based agent, the steps are similar.

Before you perform the steps described in this section, do the following:

- Ensure that you have administrative privileges on the application host on which you want to install the agent.

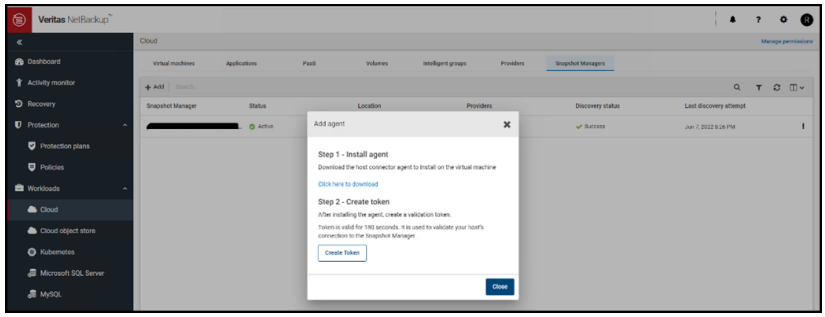
If a non-admin user attempts the installation, the installer displays the Windows UAC prompt where the user must specify the credentials of an admin user.

- Complete the preparatory steps and install all the dependencies for the respective agent.
  See “Preparing to install the Linux-based agent” on page 131.
  See “Preparing to install the Windows-based agent” on page 135.

To download and install the agent

1. Sign in to the NetBackup Web UI.
2. From the left navigation pane, click **Cloud** and then select the **Snapshot Managers** tab.

   All the Snapshot Manager servers that are registered with the primary server are displayed in this pane.
3. From the desired Snapshot Manager server row, click the actions icon on the right and then select **Add agent**.



4. On the Add agent dialog box, click the 'download' link.

   This launches a new browser window.

   Do not close the existing Add agent dialog box on the NetBackup Web UI as yet. When you configure the agent, you will return to this dialog box to get the authentication token.
5. Switch to the new web page browser window and from the Add Agent section, click on the download link to download the desired Snapshot Manager agent installation package.

   The web page provides separate links to download the Linux and Windows agents.
6. If necessary, copy the downloaded agent package to the application host on which you want to install the agent.

7 Install the agent.

- For the Linux-based agent, type the following command on the Linux host:

  ```
  # sudo yum -y install <snapshotmanager_agent_rpm_name>
  ```

  Here, <*snapshotmanager_agent_rpm_name*> is the name of the agent rpm package you downloaded earlier.

  For example:

  ```
  # sudo yum -y install
  VRTSflexsnap-agent-10.0.1.0.1005-RHEL.x86_64.rpm
  ```

- For the Windows-based agent, run the agent package file and follow the installation wizard workflow to install the agent on the Windows application host.

  Note: To allow the installation, admin users will have to click Yes on the Windows UAC prompt. Non-admin users will have to specify admin user credentials on the UAC prompt.

  The installer installs the agent at `C:\Program Files\Veritas\CloudPoint` by default and the path cannot be modified.

  Alternatively, you can also install the Windows-based agent in a silent mode by running the following command on the Windows host:

  ```
  msiexec /i <installpackagefilepath> /qn
  ```

  Here, <*installpackagefilepath*> is the absolute path of the installation package. For example, if the installer is kept at `C:\temp`, then the command syntax is as follows:

  ```
  msiexe /i C:\temp\VRTSflexsnap-core-<ver>-Windows.x64.msi /qn
  ```

  In this mode, the installation package does not display any UI and also does not require any user intervention. The agent is installed at `C:\Program Files\Veritas\CloudPoint` by default and the path cannot be modified.

  The silent mode of installation is useful if you want to automate the agent installation using a third-party deployment tool.

8 This completes the agent installation. You can now proceed to register the agent.

See “Registering the Linux-based agent” on page 131.

See “Registering the Windows-based agent” on page 135.

Linux-based agent

This section describes the procedures for preparing and registering the Linux-based agent.

Preparing to install the Linux-based agent

Before you install the Linux-based agent on the application host, ensure that you perform the following:

- If you are installing the Linux-based agent to discover Oracle applications, optimize your Oracle database files and metadata files.
  See “Optimizing your Oracle database data and metadata files” on page 147.
  See “About the installation and configuration process” on page 127.

Registering the Linux-based agent

Verify the following before you register the Linux-based agent:

- Ensure that you have downloaded and installed the agent on the application host.
  See “Downloading and installing the Snapshot Manager agent” on page 128.
- Ensure that you have root privileges on the Linux instance.
- If the Snapshot Manager Linux-based agent was already configured on the host earlier, and you wish to re-register the agent with the same Snapshot Manager instance, then do the following on the Linux host:
  - Remove the `/opt/VRTcloudpointr/keys` directory from the Linux host.
    Type the following command on the host where the agent is running:
    ```
    # sudo rm -rf /opt/VRTScloudpoint/keys
    ```
- If the Snapshot Manager Linux-based agent was already registered on the host earlier, and you wish to register the agent with a different Snapshot Manager instance, then do the following on the Linux host:
  - Uninstall the agent from the Linux host.
    See “Removing the Snapshot Manager agents” on page 214.
  - Remove the `/opt/VRTScloudpoint/keys` directory from the Linux host.
    Type the following command:
    ```
    # sudo rm -rf /opt/VRTScloudpoint/keys
    ```
  - Remove the `/etc/flexsnap.conf` configuration file from the Linux host.
    Type the following command:
    ```
    sudo rm -rf /etc/flexsnap.conf
    ```
  - Re-install the agent on the Linux host.
    See “Downloading and installing the Snapshot Manager agent” on page 128.

  If you do not perform these steps, then the on-host agent registration may fail with the following error:

```
On-host registration has failed. The agent is already registered
with Snapshot Manager instance <instance>.
```

To register the Linux-based agent

1. Return to the NetBackup Web UI, and on the Add agent dialog box, click **Create Token**.

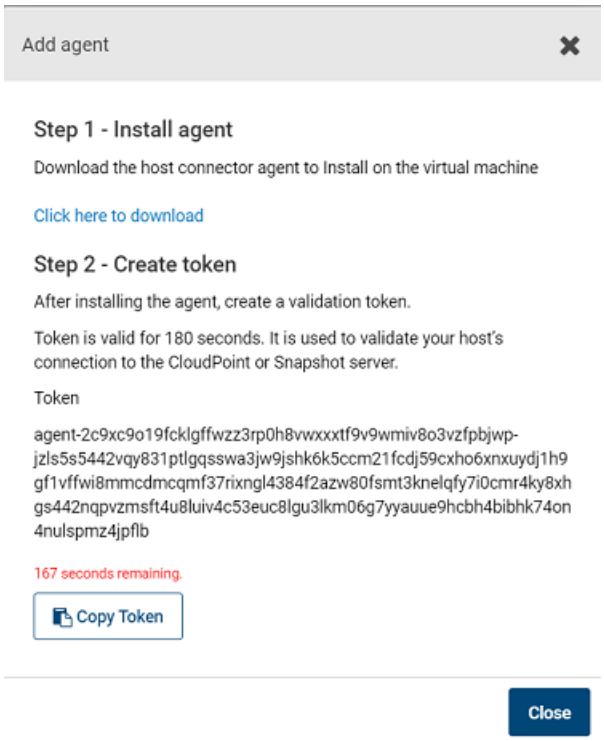
   If you have closed the dialog box, sign in to the NetBackup Web UI again and do the following:

   - Click **Cloud** from the left navigation menu, and select the **Snapshot Managers** tab.
   - From the desired Snapshot Manager server row, click the actions button on the right and then select **Add agent**.

- On the Add agent dialog box, click **Create Token**.

2 Click **Copy Token** to copy the displayed Snapshot Manager validation token.

The token is a unique sequence of alpha-numeric characters and is used as an authentication token to authorize the host connection with Snapshot Manager.



Note: The token is valid for 180 seconds only. If you do not copy the token within that time frame, generate a new token again.

3. Connect to the Linux host and register the agent using the following command:

   ```
   # sudo flexsnap-core --ip <snapshotmanager_host_FQDN_or_IP>
   --token <authtoken>
   ```

   Here, *<snapshotmanager_host_FQDN_or_IP>* is the Snapshot Manager server's Fully Qualified Domain Name (FQDN) or IP address that was specified during the Snapshot Manager configuration.

   <authtoken> is the authentication token that you copied in the earlier step.

   Note: You can use `flexsnap-core --help` to see the command help.

   Snapshot Manager performs the following actions when you run this command:

   Note: If you encounter an error, check the `flexsnap-core` logs to troubleshoot the issue.

4. Return to the NetBackup Web UI, close the Add agent dialog box, and then from the Snapshot Manager server row, click the actions button on the right and then click **Discover**.

   This triggers a manual discovery of all the assets that are registered with the Snapshot Manager server.

5. Click on the **Virtual machines** tab.

   The Linux host where you installed the agent should appear in the discovered assets list.

   Click to select the Linux host. If the host status is displayed as **VM Connected** and a **Configure Application** button appears, it confirms that the agent registration is successful.

6. This completes the agent registration. You can now proceed to configure the application plug-in.

   See “Configuring an application plug-in” on page 138.

Windows-based agent

This section describes the procedures for preparing and registering the Windows-based agent.

Preparing to install the Windows-based agent

Before you install the Windows-based agent, do the following on the Windows application host:

- Verify that the required ports are enabled on the Snapshot Manager host.
  See “Verifying that specific ports are open on the instance or physical host” on page 33.
- Verify that you can connect to the host through Remote Desktop.
- Verify that the `pagefile.sys` is not present on the drive or volume that you wish to protect using Snapshot Manager. If the file exists on such drives, move it to an alternate location.
  Restore of the snapshot will fail to revert the shadow copy if the `pagefile.sys` resides on the same drive or volume on which the operations are being performed.

Registering the Windows-based agent

Verify the following before you register the Windows-based agent:

- Ensure that you have downloaded and installed the agent on the Windows application host.
  See “Downloading and installing the Snapshot Manager agent” on page 128.
- Ensure that you have administrative privileges on the Windows host.

To register the Windows-based agent

1. Return to the NetBackup Web UI, and on the Add agent dialog box, click **Create Token**.

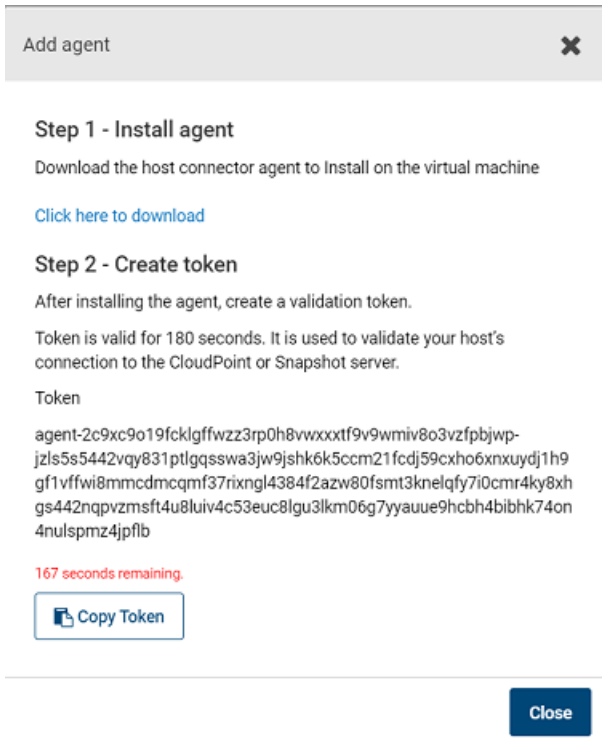
   If you have closed the dialog box, sign in to the NetBackup Web UI again and do the following:

   - Click **Cloud** from the left navigation menu, and select the **Snapshot Managers** tab.
     From the desired Snapshot Manager server row, click the actions button on the right and then select **Add agent**.

- On the Add agent dialog box, click **Create Token**.

2 Click **Copy Token** to copy the displayed Snapshot Manager validation token.

The token is a unique sequence of alpha-numeric characters and is used as an authentication token to authorize the host connection with Snapshot Manager.



Note: The token is valid for 180 seconds only. If you do not copy the token within that time frame, generate a new token again.

3 Connect to the Windows instance and register the agent.

From the command prompt, navigate to the agent installation directory and type the following command:

```
flexsnap-agent.exe --ip <snapshotmanager_host_FQDN_or_IP> --token <authtoken>
```

The default path is `<System Drive>\Program Files\Veritas\CloudPoint\`.

Here, <*snapshotmanager_host_FQDN_or_IP*> is the NetBackup host's Fully Qualified Domain Name (FQDN) or IP address that was used during the NetBackup initial configuration.

<*authtoken*> is the authentication token that you copied in the earlier step.

Note: You can use `flexsnap-agent.exe --help` to see the command help.

NetBackup performs the following actions when you run this command:

- registers the Windows-based agent
- creates a `<System Drive>\ProgramData\Veritas\CloudPoint\etc\flexsnap.conf` configuration file on the Windows instance and updates the file with NetBackup host information
- enables and then starts the agent service on the Windows host

---

Note: If you intend to automate the agent registration process using a script or a 3rd-party deployment tool, then consider the following:

Even if the agent has been registered successfully, the Windows agent registration command may sometimes return error code 1 (which generally indicates a failure) instead of error code 0.

An incorrect return code might lead your automation tool to incorrectly indicate that the registration has failed. In such cases, you must verify the agent registration status either by looking in to the flexsnap-agent-onhost logs or from the NetBackup Web UI.

---

4 Return to the NetBackup Web UI, close the Add agent dialog box, and then from the Snapshot Manager server row, click the actions button on the right and then click **Discover**.

This triggers a manual discovery of all the assets that are registered with the Snapshot Manager server.

5 Click on the **Virtual machines** tab.

The Windows host where you installed the agent should appear in the discovered assets list.

Click to select the Windows host. If the host status is displayed as **VM Connected** and a **Configure Application** button appears, it confirms that the agent registration is successful.

6 This completes the agent registration. You can now proceed to configure the application plug-in.

See “Configuring an application plug-in” on page 138.

Configuring the Snapshot Manager application plug-in

After installing and registering the Snapshot Manager agent on the application host, the next step is to configure the application plug-in on the host.

Before you proceed, ensure that you perform the following:

- Verify that you have configured the agent on the host.
  See “Registering the Linux-based agent” on page 131.
  See “Registering the Windows-based agent” on page 135.
- Review the configuration requirements for the plug-in you want to configure.
  See “Oracle plug-in configuration requirements” on page 146.
  See “Microsoft SQL plug-in configuration requirements” on page 139.

Configuring an application plug-in

To configure an application plug-in

1 Sign in to the NetBackup Web UI and from the left navigation pane, click **Cloud** and then select the **Virtual machines** tab.

2 From the list of assets, search for the application host where you installed and registered the Snapshot Manager agent.

Click to select the application host and verify that the **Configure application** button appears in the top bar.

3 Click **Configure application** and from the drop-down list, select the application plug-in that you want to configure, and then click **Configure**.

For example, if you want to configure the Snapshot Manager plug-in for Microsoft SQL, choose **Microsoft SQL Server**.

4 After the plug-in is configured, trigger an assets discovery cycle.

Click the **Snapshot Managers** tab and then from the desired Snapshot Manager server row, click the action button from the right and then click **Discover**.

5 After the discovery is completed, click the **Virtual machines** tab and verify the state of the application host. The Application column in the assets pane displays a value as **Configured** and this confirms that the plug-in configuration is successful.

6 Click on the **Applications** tab and verify that the application assets are displayed in the assets list.

For example, if you have configured the Microsoft SQL plug-in, the Applications tab displays the SQL Server instances, databases, and SQL Availability Group (AG) databases that are running on the host where you configured the plug-in.

You can now select these assets and start protecting them using protection plans.

Microsoft SQL plug-in

You can configure the Snapshot Manager plug-in for Microsoft SQL to discover SQL application instances and databases and protect them using disk-level snapshots. After you configure the plug-in, Snapshot Manager automatically discovers all the file system assets, SQL instances and databases that are configured on the SQL server host. The discovered SQL assets then appear in the NetBackup user interface (UI) from where you can protect the assets by subscribing them to a protection plan or by taking snapshots manually.

Microsoft SQL plug-in configuration requirements

Before you configure the plug-in, ensure that your environment meets the following requirements:

- This plug-in is supported in Microsoft Azure, Google Cloud Platform and Amazon AWS environments.
- A supported version of Microsoft SQL server is installed on the Windows instance.
  See “ Meeting system requirements” on page 17.
- The SQL server instances that you want to protect must be running on a non-system drive.
  Snapshot Manager also does not support SQL server instances that are installed on a mount point.
- Snapshot Manager uses the Microsoft Volume Shadow Copy Service (VSS).

Ensure that you configure VSS to store shadow copies on the same drive (the originating drive) where the database resides.
See “Configuring VSS to store shadow copies on the originating drive” on page 151.

Restore requirements and limitations for Microsoft SQL Server

Consider the following before you restore a SQL Server snapshot:

- Ensure that you close SQL Management Studio before you restore a SQL Server snapshot.
  This is applicable only if you are restoring the snapshot to replace the current asset (Overwrite existing option) or restoring the snapshot to the same location as the original asset (Original Location option).
- In case of a SQL instance disk-level restore to a new location fails if the target host is connected or configured.
  In such a case, to complete the SQL Server snapshot restore to a new location successfully, you must perform the restore in the following order:
  - First, perform a SQL Server disk-level snapshot restore.
    Ensure that you restore the disk snapshots of all the disks that are used by SQL Server. These are the disks on which SQL Server data is stored.
    See “Steps required before restoring SQL AG databases” on page 141.
  - Then, after the disk-level restore is successful, perform the additional manual steps.
    See “Additional steps required after a SQL Server instance snapshot restore” on page 142.
- Snapshot Manager does not support discovery, snapshot, and restore operations for SQL databases that contain leading or trailing spaces or non-printable characters. This is because the VSS writer goes into an error state for such databases.
  Refer to the following for more details:
  Microsoft SQL Server database documentation
- Before you restore a SQL Availability Group (AG) database, perform the pre-restore steps manually.
  See “Steps required before restoring SQL AG databases” on page 141.
- New location restore of system database is not supported.
- If destination instance has AG configured, restore is not supported.
- If database exists on new location destination and the overwrite existing option is not selected, the restore job will fail.

- If the overwrite existing option is selected for database that is a part of an AG, the restore job will fail.
- For system database restore, the SQL Server version must be same. For user databases, restore from a higher SQL version to a lower version is not allowed.
- Default timeout of 6 hours is not allowing restore of larger database (size more than 300 GB). Configurable timeout parameter value can be set to restore larger database.

  See “Troubleshooting Snapshot Manager” on page 227.

Steps required before restoring SQL AG databases

You must perform the following steps before you restore a SQL Availability Group (AG) database:

Note: If you are restoring the AG database to multiple replicas, perform the entire restore process on the primary replica first, and then repeat the steps for each secondary replica.

---

1. For the database that you want to restore, suspend data movement from the replica.

   From the SQL Server Management Studio, right-click on the database and select **Suspend Data Movement**.

2. Remove the database from the AG on the replica.

   From the SQL Server Management Studio, right-click on the database and select **Remove Database from Availability Group**.

   Confirm that the database is no longer part of the AG. Observe that the database on the primary replica is no longer in synchronized mode, and the status of the corresponding database on the secondary replica appears as `(Restoring...)`.

3. Delete the database from the replica.

   From the SQL Server Management Studio, right-click on the database and select **Delete**.

Additional steps required after restoring SQL AG databases

You must perform the following steps after restoring a SQL Availability Group (AG) database:

Note: If you are restoring the AG database to multiple replicas, perform the entire restore process on the primary replica first, and then repeat the steps for each secondary replica.

- Add the restored database to the AG on the primary replica.
  From the SQL Server Management Studio, right-click on the AG entry and select **Add Database**. In the wizard workflow, select the database, and on the Initial Data Synchronisation page, select the **Skip Initial Data Synchronization** option. You can select the other options depending on the requirement.

If you restoring the same database to a secondary replica, perform the following steps:

1. Restore database to the secondary SQL instance in "Not recovered" state. Restore with no recovery should be successful.
2. Join the database to the AG on the secondary replica.

   From the SQL Server Management Studio, connect to the secondary replica node, then right-click on the database and select **Join Availability Group**.

   Observe that the database status on the secondary replica change from `(Restoring...)` to `(Synchronized)`, indicating that AG database snapshot restore is successful.

You must repeat these steps for each replica where you wish to restore an AG database.

Additional steps required after a SQL Server instance snapshot restore

The following steps are required after you restore a SQL Server instance snapshot from the NetBackup user interface (UI). Even though the restore operation is successful, these steps are required for the application database to be available for normal use again.

Steps required after a SQL Server host-level restore

Perform these steps after you have restored a host-level SQL Server snapshot from the NetBackup UI. These steps are required irrespective of whether you are restoring the snapshot to the original location or to a new location.

Before you proceed, verify the following:

- Ensure that the SQL Server user account on the Windows host where you intend to revert the shadow copy, has full access to the restore data.
- Ensure that the `pagefile.sys` is not present on the drive that is selected for the snapshot creation or snapshot restore.

The snapshot creation and snapshot restore operations will fail if the file is present on the selected drives.

Perform the following steps to revert the shadow copy

1. Connect to the Windows host where the SQL Server instance is running.

   Ensure that you use an account that has administrator privileges on the host.

2. Stop the SQL Server service on the Windows host.

3. Open a command prompt window. If Windows UAC is enabled on the host, open the command prompt in the **Run as administrator** mode.

4. Navigate to `%programdata%\Veritas\CloudPoint\tmp\tools\windows\tools\` directory, and then run the following command from there:

   ```
   vss_snapshot.exe  --revertSnapshot
   ```

   The command displays a json output with Status = 0 that confirms that the operation is successful.

   This command reverts the shadow copies for all the drives, except the system drive. The SQL Server service is stopped before the snapshot is reverted and automatically started after the revert operation is successful.

5. Start the SQL Server service on the Windows host.

Steps required after a SQL Server instance disk-level snapshot restore to new location

Perform these steps after you have restored a disk-level SQL Server instance snapshot from the NetBackup UI. These steps are required only if the snapshot is restored to a new location. New location refers to a new host that is different from the one where the SQL instance is running.

Note: These steps are applicable only in case of a SQL Server instance snapshot restore to a new location. These are not applicable for a SQL Server database snapshot restore.

Clear the read-only mode of the new disk attached to the host

Perform the following steps

1. Connect to the new Windows host where the SQL Server instance is running.

   Ensure that you use an account that has administrator privileges on the host.

2. Open a command prompt window. If Windows UAC is enabled on the host, open the command prompt in the **Run as administrator** mode.

3. Start the diskpart utility using the following command:

   ```
   diskpart
   ```

4. View the list of disks on the new host using the following command:

   ```
   list disk
   ```

   Identify the new disk that is attached due to the snapshot restore operation and make a note of the disk number. You will use it in the next step.

5. Select the desired disk using the following command:

   ```
   select disk <disknumber>
   ```

   Here, *<disknumber>* represents the disk that you noted in the earlier step.

6. View the attributes of the selected disk using the following command:

   ```
   attributes disk
   ```

   The output displays a list of attributes for the disk. One of the attributes is `read-only`, which we will modify in the next step.

7. Modify the read-only attribute for the selected disk using the following command:

   ```
   attributes disk clear readonly
   ```

   This command changes the disk to read-write mode.

8. Bring the disk online.

   From the Windows Server Manager console, navigate to **Files and Storage Devices > Disks** and then right click on the newly attached disk and select **Bring online**.

9. Assign drive letters to the volumes on the disk that you brought online in the earlier step. Drive letters are required to view the shadow copies associated with each volume on the disk.

   Go back to the command prompt window and perform the following steps:

   - View the list of volumes on the new host using the following command:

     ```
     list volume
     ```

From the list of volumes displayed, identify the volume for which you want to assign, modify, or remove a drive letter.

- Select the desired volume using the following command:

  ```
  select volume <volnumber>
  ```

  Here, *<volnumber>* represents the volume that you noted in the earlier step.

- Assign a drive letter to the selected volume using the following command:

  ```
  assign letter=<driveletter>
  ```

  Here, *<driveletter>* is the drive letter that you wish to assign to the volume. Ensure that the specified drive letter is not already in use by another volume.

- Repeat these steps to assign a drive letter to all the SQL Server volumes on the disk.

10 Quit the diskpart utility using the following command:

```
exit
```

Do not close the command prompt yet; you can use the same window to perform the remaining steps described in the next section.

Revert shadow copy using the Microsoft DiskShadow utility

Perform the following steps

1 From the same command window used earlier, start the diskshadow command interpreter in the interactive mode using the following command:

```
diskshadow
```

2 View the list of all the shadow copies that exist on the new host. Type the following command:

```
list shadows all
```

Identify the shadow copy that you want to use for the revert operation and make a note of the shadow copy ID. You will use the shadow ID in the next step.

3 Revert the volume to the desired shadow copy using the following command:

```
revert <shadowcopyID>
```

Here, *<shadowcopyID>* is the shadow copy ID that you noted in the earlier step.

4 Exit the DiskShadow utility using the following command:

```
exit
```

Attach .mdf and .ldf files to the instance database

Perform the following steps:

1. Ensure that the disk-level snapshot restore operation has completed successfully and a new disk is created and mounted on the application host.
2. Log on to Microsoft SQL Server Management Studio as a database administrator.
3. From the Object Explorer, connect to an instance of the SQL Server Database Engine and then click to expand the instance view.
4. In the expanded instance view, right-click **Databases** and then click **Attach**.
5. In the Attach Databases dialog box, click **Add** and then in the Locate Database Files dialog box, select the disk drive that contains the database and then find and select all the .mdf and .ldf files associated with that database. Then click **OK**.

   The disk drive you selected should be the drive that was newly created by the disk-level snapshot restore operation.
6. Wait for the requested operations to complete and then verify that the database is available and is successfully discovered by NetBackup.

Oracle plug-in

You can configure the Oracle plug-in to discover and protect your Oracle database applications with disk-level snapshots.

Oracle plug-in configuration requirements

Before you configure the Oracle plug-in, make sure that your environment meets the following requirements:

- A supported version of Oracle is installed in a supported Red Hat Enterprise Linux (RHEL) host environment.
  See “ Meeting system requirements” on page 17.
- Oracle standalone instance is discoverable.
- Oracle binary and Oracle data must be on separate volumes.
- Log archiving is enabled.
- The `db_recovery_file_dest_size` parameter size is set as per Oracle recommendation.
  For more information, refer to the Oracle Database Backup and Recovery Basics.
- The databases are running, mounted, and open.

- Snapshot Manager supports discovery and snapshot operations on databases that are in a backup mode. After taking snapshots, the state of the databases is retained as is; Snapshot Manager does not change the status of such databases. However, in-place restore for such databases is not supported.

Optimizing your Oracle database data and metadata files

Veritas recommends that you do not keep the Oracle configuration files on a boot or a root disk. Use the following information to know more about how to move those files and optimize your Oracle installation.

Veritas takes disk snapshots. For better backup and recovery, you should optimize your Oracle database data and metadata files.

Each Oracle database instance has a control file. The control file contains information about managing the database for each transaction. For faster and efficient backup and recovery, Oracle recommends that you put the control file in the same file system as the database redo log file. If the database control file resides on the file system that is created on top of the boot disk or root disk, contact your database administrator to move the control file to the appropriate location.

For more information on control files and how to move them, contact your database administrator, or see the Oracle documentation.

After you use a snapshot to restore an application, do not perform any operations. Allow some time for Oracle to read new data and bring up the database. If the database does not come up, contact the database administrator to determine the cause of the problem.

Restore requirements and limitations for Oracle

Consider the following before you restore an Oracle snapshot:

- The destination host where you wish to restore the snapshot must have the same Oracle version installed as that at the source.
- If you are restoring the snapshot to a new location, verify the following:
  - Ensure that there is no database with the same instance name running on the target host.
  - The directories that are required to mount the application files are not already in use on the target host.
- Disk-level restore to a new location fails if the NetBackup plug-in for Oracle is not configured on the target host.
  In such a case, to complete the Oracle snapshot restore to a new location successfully, you must perform the restore in the following order:
  - First, perform a Oracle disk-level snapshot restore.

Ensure that you restore the disk snapshots of all the disks that are used by Oracle. These are the disks on which Oracle data is stored.

- Then, after the disk-level restore is successful, perform the additional manual steps.
  See “Additional steps required after an Oracle snapshot restore” on page 148.

- In an Azure environment, it is observed that the device mappings may sometimes get modified after performing a host-level restore operation. As a result, the Oracle application may fail to come online on the new instance, after the restore.
  To resolve this issue after the restore, you have to manually unmount the file systems and then mount them again appropriately as per the mappings on the original host.
  If you are using the `/etc/fstab` file to store file systems, mount points, and mount settings, Veritas recommends that you use the disk UUID instead of device mappings. Using disk UUIDs ensures that the file systems are mounted correctly on their respective mount points.
- Snapshots of application data residing on a filesystem that is part of an LVM type of partition are not supported. If you try to take a snapshot of such a filesystem, the following error is displayed:
  ```
  *flexsnap.GenericError: Unable to protect asset *
  ```

Additional steps required after an Oracle snapshot restore

The following steps are required after you restore an Oracle snapshot. Even though the restore operation itself is successful, these steps are required for the application database to be available for normal use again.

These manual steps are not required in case of a disk-level restore in the following scenario:

- You are performing a disk-level restore to the original location or an alternate location
- The target host is connected to the Snapshot Manager host
- The Snapshot Manager Oracle plug-in is configured on the target host

Perform the following steps:

1. Ensure that the snapshot restore operation has completed successfully and a new disk is created and mounted on the application host (in case of a disk-level restore) or the application host is up and running (in case of a host-level restore).
2. Connect to the virtual machine and then log on to the Oracle database as a database administrator (sysdba).

3 Start the Oracle database in mount mode using the following command:

```
# STARTUP MOUNT
```

Verify that the database is mounted successfully.

4 Remove the Oracle database from the backup mode using the following command:

```
# ALTER DATABASE END BACKUP
```

5 Open the Oracle database for normal usage using the following command:

```
# ALTER DATABASE OPEN
```

6 Add an entry of the newly created database in the Oracle `listerner.ora` and `tnsnames.ora` files.

7 Restart the Oracle listener using the following command:

```
# lsnrctl start
```

NetBackup protection plan

A protection plan defines when backups are performed, how long the backups are retained, and the type of storage to use. Once you have set up a protection plan, you can subscribe assets to that protection plan.

Creating a NetBackup protection plan for cloud assets

To create a protection plan

1 Sign in to the NetBackup Web UI.

2 From the left navigation pane, click **Protection plans** and then click **Add** from the right hand side.

3 On the Basic properties panel, do the following:

- Enter a **Name** and **Description** for the plan.
- From the **Workload** drop-down, select **Cloud**.
- From the **Cloud Provider** drop-down, select a cloud provider. NetBackup supports homogenous cloud asset subscriptions. While subscribing an asset to a protection plan, the cloud provider of the asset must be the same as the cloud provider defined in the protection plan.
- Click **Next**.

4 On the Schedules and retention panel, specify the desired backup schedule and then click **Next**.

5 Configure the remaining options as per your requirement and click **Finish** to create the protection plan.

The Protection plans pane displays the plan you created.

6 You can now proceed to assign assets to this protection plan.

See “Subscribing cloud assets to a NetBackup protection plan” on page 150.

For detailed information about managing protection plans, refer to the *NetBackup Web UI Backup Administrator's Guide*.

Subscribing cloud assets to a NetBackup protection plan

You can subscribe a single asset or a group of assets to a protection plan. For example, you can create a plan to create weekly snapshots and assign the policy to all your database applications. Also, an asset can have more than one policy. For example ,in addition to weekly snapshots, you can assign a second policy to your database applications to take a snapshot once a month.

NetBackup supports homogenous cloud asset subscriptions. While subscribing an asset to a protection plan, the cloud provider of the asset must be the same as the cloud provider defined in the protection plan.

Before you proceed, ensure that you have sufficient privileges to assign assets to a protection plan from the NetBackup Web UI.

To subscribe cloud assets to a protection plan

1 Sign in to the NetBackup Web UI.

2 From the left navigation pane, click **Cloud** and then select the **Applications** tab.

The Application tab displays a list of assets that you can protect.

3 On the Applications tab, search and select the asset that you wish protect and then click **Add Protection**.

For example, to protect Microsoft SQL, you can select a SQL instance, a standalone database, or an Availability Group (AG) database.

Note: If instance level SQL server backup is selected, only the databases that are online are included in the snapshot. The snapshot does not include databases that are offline or in an erroneous state.

---

4 On the Choose a protection plan panel, search and select the appropriate protection plan and then click **Protect**.

Verify that on the Applications tab, the Protected by column for the selected asset displays the protection plan that you just assigned. This indicates that the asset is now being protected by the configured protection plan.

The backup jobs should automatically get triggered as per the schedule defined in the plan. You can monitor the backup jobs from the Activity monitor pane.

For more detailed information on how to subscribe assets to a protection plan, refer to the *NetBackup Web UI Backup Administrator's Guide*.

Configuring VSS to store shadow copies on the originating drive

If you want to take disk-level, application-consistent snapshots of a Windows file system or Microsoft SQL application, you must configure Microsoft Volume Shadow Copy Service (VSS). VSS lets you take volume snapshots while applications continue to write to the volume.

When you configure VSS, keep in mind the following;

- Snapshot Manager currently has a limitation that you must manually configure the shadow copy creation location to the same drive or volume as the originating drive. This approach ensures that an application-consistent snapshot is created.
- If shadow storage already exists on an alternate drive or a dedicated drive, you must disable that storage and replace it with the configuration in the following procedure.
- Snapshot Manager does not support discovery, snapshot, and restore operations for SQL databases that contain leading or trailing spaces or non-printable characters. This is because the VSS writer goes into an error state for such databases.

For more information, see Microsoft Documentation.

To configure VSS to store shadow copies on the originating drive

1. On the Windows host, open the command prompt. If User Account Control (UAC) setting is enabled on the server, launch the command prompt in the **Run as administrator** mode.
2. For each drive letter on which you want to take disk-level, application-consistent snapshots using Snapshot Manager, enter a command similar to the following:

   ```
   vssadmin add shadowstorage /for=<drive being backed up> ^
   /on=<drive to store the shadow copy> ^
   /maxsize=<percentage of disk space allowed to be used>
   ```

   Here, `maxsize` represents the maximum free space usage allowed on the shadow storage drive. The caret (^) character in the command represents the Windows command line continuation character.

   For example, if the VSS shadow copies of the `D:` drive are to be stored on the `D:` drive and allowed to use up to 80% of the free disk space on `D:`, the command syntax is as follows:

   ```
   vssadmin add shadowstorage /for=d: /on=d: /maxsize=80%
   ```

   The command prompt displays a message similar to the following:

   ```
   Successfully added the shadow copy storage association
   ```

3. Verify your changes using the following command:

   ```
   vssadmin list shadowstorage
   ```

Additional steps required after restoring an AWS RDS database instance

The following steps are required after you restore an AWS RDS database instance snapshot. Even though the restore operation is successful, these manual steps are required so that the instance is available for normal use.

After restoring an AWS RDS database instance successfully, you have to manually check and reassign certain properties of the restored instance. This is required because even though the restore operation itself is successful, one or more instance properties are not restored completely. In some cases, NetBackup resets the property values to their default settings.

The following RDS database instance or cluster properties are not restored completely and will need modification:

- **VPC security groups** value (*AWS Management Console > RDS Database instance > Connectivity & security tab*)
- **Deletion protection** setting (*AWS Management Console > RDS Database instance > Configuration tab*)
- **Copy tags to snapshots** setting (*AWS Management Console > RDS Database instance > Maintenance & backups tab*)

Perform the following steps:

1. Verify that the RDS database instance snapshot restore is successful.
2. Sign in to the AWS Management Console and from the top right corner, select the region in which you have restored the RDS instance.
3. From the Services menu, under Database, click **RDS**.
4. From the Dashboard menu on the left, click **Databases**.
5. In the Databases panel, select the restored RDS database instance and then click **Modify** from the menu bar on the top right.
6. On the Modify DB panel, check for the following properties and ensure that the attribute values match with those of the original instance:
   - Under Network & Security, verify that the **Security group** attribute has the correct security group name assigned.
   - Under Backup, verify that the **Copy tags to snapshots** option is set as per the original instance.
   - Under Deletion protection, verify that the **Enable deletion protection** option is set as per the original instance.
   - If required, verify all the other parameter values and set them as per your preference.
7. Once you have modified the desired RDS instance properties, click **Continue**.
8. Under Scheduling of modifications, choose an appropriate option depending on when you wish to apply the modifications to the instance and then click **Modify DB instance**.
9. Verify the RDS instance properties and ensure that the changes have taken effect.

Chapter 7

Protecting assets with NetBackup Snapshot Manager's agentless feature

This chapter includes the following topics:

- About the agentless feature
- Prerequisites for the agentless configuration
- Configuring the agentless feature
- Configuring the agentless feature after upgrading Snapshot Manager

About the agentless feature

If you want NetBackup to discover and protect assets on a host, but you want to minimize the vendor software footprint on the hosts, consider Snapshot Manager's agentless feature. Typically, when you use an agent, the software remains on the host at all times. In contrast, the agentless feature works as follows:

- The Snapshot Manager software accesses the host through SSH on Linux and WMI and SMB in case of Windows.
- Snapshot Manager performs the specified task, such as creating a snapshot.
- When the task completes, Snapshot Manager software stops the process.

The Snapshot Manager agentless feature currently discovers and operates on Windows or Linux file system assets, Oracle database and Microsoft SQL database assets.

See “Prerequisites for the agentless configuration” on page 156.

See “Configuring the agentless feature” on page 158.

Prerequisites for the agentless configuration

Prerequisites for using the agentless feature in Linux

- Have the following information with you:
  - Host user name
  - Host password or SSH key

  Snapshot Manager requires these details to gain access to the host and perform requested operations.
- On hosts where you wish to configure this feature, grant password-less sudo access to the host user account that you provide to Snapshot Manager.

Granting password-less sudo access to host user account

Snapshot Manager requires a host user account to connect and perform operations on the host. You must grant password-less sudo access to the user account that you provide to Snapshot Manager. This is required for all the hosts where you wish to configure the agentless feature.

Note: The following steps are provided as a general guideline. Refer to the operating system or the distribution-specific documentation for detailed instructions on how to grant password-less sudo access to a user account.

---

1. Perform the following steps on a host where you want to configure the agentless feature
2. Verify that the host user name that you provide to Snapshot Manager is part of the `wheel` group.

   Log on as a root user and run the following command:

   ```
   # usermod -aG wheel hostuserID
   ```

   Here, *hostuserID* is the host user name that you provide to Snapshot Manager.
3. Log out and log in again for the changes to take effect.
4. Edit the `/etc/sudoers` file using the `visudo` command:

```
# sudo visudo
```

5. Add the following entry to the `/etc/sudoers` file:

   ```
   hostuserID ALL=(ALL) NOPASSWD: ALL
   ```

6. In the `/etc/sudoers` file, edit the entries for the `wheel` group as follows:

   - Comment out (add a # character at the start of the line) the following line entry:
     # %wheel ALL=(ALL) ALL
   - Uncomment (remove the # character at the start of the line) the following line entry:
     %wheel ALL=(ALL) NOPASSWD: ALL

   The changes should appear as follows:

   ```
   ## Allows people in group wheel to run all commands
   # %wheel ALL=(ALL) ALL

   ## Same thing without a password
   %wheel ALL=(ALL) NOPASSWD: ALL
   ```

7. Save the changes to the `/etc/sudoers` file.
8. Log out and log on to the host again using the user account that you provide to Snapshot Manager.
9. Run the following command to confirm that the changes are in effect:

   ```
   # sudo su
   ```

   If you do not see any prompt requesting for a password, then the user account has been granted password-less sudo access.

   You can now proceed to configure the Snapshot Manager agentless feature.

Prerequisites for using the agentless feature in Windows

- The user account used to connect to remote instance should be able to:
  - Access remote admin share (ADMIN$). Enabled by default.
  - Access to `root\cimv2`
- Configure the following ports:
  - Modify the security group to allow inbound traffic on the ports 135, 445 and dynamic port or fixed port for WMI .
  - Enable inbound rules in the firewall for the ports 135, 445 and the dynamic or fixed WMI-IN ports on Windows hosts.

Note: The dynamic range for the ports is 49152-65535.

- You can use fixed or dynamic WMI-IN ports. If you want to configure a fixed WMI-IN port, see Setting Up a Fixed Port for WMI.

- Disable User Account Control for the users groups accessing the agentless feature.
- For protecting SQL applications, the user account used for connecting to the cloud host, must have the required admin privileges to access the SQL server.

Configuring SMB for Windows (Optional)

Perform the following Server Message Block (SMB) configurations before configuring the agentless feature on Windows.

- Restrict unencrypted access to SMB share by setting the value to `True`.

  ```
  RejectUnencryptedAccess: True
  ```

- Disable SMB 1.0 by running the following command on Windows powershell:

  ```
  Set-SmbServerConfiguration –EnableSMB1Protocol $false
  ```

  For more details, see Disabling SMB 1.0

For more details on SMB security, see: SMB security enhancements.

Configuring WMI security for Windows (optional)

Windows Management Instrumentation (WMI) security protects access to the namespace data. Snapshot Manager uses the `root\cimv2` namespace. This name space must be accessible to only those users that are configured using the connect option. For details, see Maintaining WMI Security.

Configuring the agentless feature

Verify all the prerequisites before you configure the Snapshot Manager agentless feature.

See “Prerequisites for the agentless configuration” on page 156.

To configure the agentless feature

1. Sign in to the NetBackup Web UI and from the left navigation pane, click **Cloud** and then select the **Virtual machines** tab.
2. From the list of assets, search for the host on which you want to use the agentless feature.

   Note: The Snapshot Manager agentless feature currently discovers and operates on Windows or Linux file system assets, Oracle database and MS SQL database assets.

3. Click to select the host and then click **Connect** in the top bar.

Note: If you have not assigned any credential to the VM, a message prompts you to assign the credentials before you can connect the VM. See the *Managing Credentials* section, in the *Web UI Administrator's Guide.*

Configuring the agentless feature after upgrading Snapshot Manager

After upgrade the cloud assets which were already in connected state, continues to work. If you want to change the asset's credentials for Linux agentless instance(s), which are already in connected state, the credentials must be associated and updated for the asset(s) from credential management.

Chapter 8

Volume Encryption in NetBackup Snapshot Manager

This chapter includes the following topics:

- About volume encryption support in Snapshot Manager
- Volume encryption for Azure
- Volume encryption for GCP
- Volume encryption for AWS

About volume encryption support in Snapshot Manager

NetBackup Snapshot Manager supports disk volume encryption for AWS, Azure, and Google Cloud Platform. Volume encryption is provided using customer keys or system keys from the cloud provider Key Management Service (KMS).

For more information on the cross account replication, refer to the *Support matrix for account replication* section of the *NetBackup™ Web UI Cloud Administrator's Guide*.

Volume encryption for Azure

You can encrypt disks in Azure using the following methods:

- Default encryption, using Platform Managed Key (PMK)

- Customer Managed Key (CMK) using Azure Key vault

For more information on Azure encryption, see: Data encryption models.

Table 8-1 Encryption for creating snapshots

Disk encryption	Snapshot encryption
Platform Managed Key (PMK)	Same PMK is used as the source disk.
Customer Managed Key (CMK)	Same CMK is used as the source disk.

Table 8-2 Encryption for restoring snapshots

Snapshot encryption	Restored disk encryption
PMK	Same PMK is used as the snapshot.
CMK	Same CMK is used as the snapshot.

Table 8-3 Encryption for restoring from backup

Snapshot encryption	Restored disk encryption
PMK	Same PMK is used as the source disk.
CMK	Same CMK is used as the source disk, else PMK is used.

Note: With this release disks would be restored with CMK if the same DES name is present in target subscription.

Volume encryption for GCP

You can encrypt disks in GCP using the following methods:

- Encryption by default (PMK or Google Managed Key)
- Customer Managed Encryption Key (CMEK) using Google Cloud KMS

For more information on GCP encryption, see: Google Cloud Encryption.

Table 8-4 Encryption for creating snapshots

Disk encryption	Snapshot encryption
Platform Managed Key (PMK)	Same PMK is used as the source disk.

Table 8-4 Encryption for creating snapshots *(continued)*

Disk encryption	Snapshot encryption
CMEK	Same CMEK is used as the source disk.

Table 8-5 Encryption for restoring snapshots

Snapshot encryption	Restored disk encryption
PMK	Same PMK is used as the snapshot.
CMEK	Same CMEK is used as the snapshot, if the target restore location is within the scope of the key.

Table 8-6 Encryption for restoring from backup

Snapshot encryption	Restored disk encryption
PMK	Same PMK is used as the source disk.
CMEK	Same CMEK is used as the source disk, else PMK is used.

Note: For successful restoration, the target restore location must be inside the scope of the key during restoration.

Volume encryption for AWS

You can encrypt disks in AWS using the following methods:

- Default encryption, using Platform Managed Key (PMK).
- Customer Managed Encryption Key (CMEK), using AWS KMS.

For more information on AWS encryption, see: Amazon EBS encryption.

Table 8-7 Encryption for creating snapshots

Disk encryption	Snapshot encryption
Platform Managed Key (PMK)	Same PMK is used as the source disk.
CMEK	Same CMEK is used as the source disk.

Table 8-8 Encryption for restoring snapshots

Snapshot encryption	Restored disk encryption
PMK	Same PMK is used as the snapshot.
CMEK	Same CMEK is used as the snapshot.

Table 8-9 Encryption for restoring from backup

Snapshot encryption	Restored disk encryption
PMK	Same PMK is used as the source disk.
CMK	Same CMK is used as the source disk, else PMK is used.

Chapter 9

NetBackup Snapshot Manager security

This chapter includes the following topics:

- Configuring security for Azure Stack
- Configuring the cloud connector for Azure Stack
- CA configuration for Azure Stack
- Securing the connection to Snapshot Manager

Configuring security for Azure Stack

You can connect to Azure Stack workload in two ways.

- The Snapshot Manager can connect to the cloud workload using provider plugins.
- The data mover container present in the Snapshot Manager, can connect to the workload, through the cloud connector plug-in component.

For Azure Stack workload, these components connect using the HTTPS protocol. By default, peer and hosts validations are always enabled.

See the section called “Proxy server requirements” on page 24.

See “Verifying that specific ports are open on the instance or physical host” on page 33.

Configuring the cloud connector for Azure Stack

The cloud connector component connects to the workloads through a secure mechanism. You need to perform the following configurations.

SSL peer and host validations

By default, peer and host validations are enabled. You can disable peer and host validations only for Azure Stack.

To disable peer and host validation, set the parameter *VIRTUALIZATION_HOSTS_SECURE_CONNECT_ENABLED=NO* in the `/cloudpoint/openv/netbackup/bp.conf` file in the Snapshot Manager. You must use HTTPS protocol, even after you disable peer and host validation.

For cloud workloads, the public root certificates are a part of the container image. NetBackup maintains the `cacert.pem` file which has root certificates of public cloud, at the following location:

`/usr/openv/var/global/wmc/cloud/cacert.pem`

For Azure Stack, you must specify the file path of the root certificates using the *ECA_TRUST_STORE_PATH* parameter in the `/cloudpoint/openv/netbackup/bp.conf` file in the Snapshot Manager. The value of *ECA_TRUST_STORE_PATH* must be in the */cloudpoint/eca/trusted/cacerts.pem* file.

Configuring CRL validations

From release 10.1 onwards Snapshot Manager will be treated as NetBackup entity while communicating with NetBackup. Certificate Revocation List (CRL) check is enabled by default while communication happens between NetBackup entities.

- ECA_CRL_CHECK: This flag is used while communicating between two NetBackup entities. By default CRL check is enabled for ECA_CRL_CHECK flag. In case Snapshot Manager machines certificate revoked then communication between NetBackup and Snapshot Manager will fail with the following error:

  `"The Snapshot Manager's certificate is not valid or doesn't exist.(9866)"`

- VIRTUALIZATION_CRL_CHECK: Before 10.1 Snapshot Manager was considered as workload while communication happens with NetBackup. Value of VIRTUALIZATION_CRL_CHECK flag is used for CRL check whenever communication happens between NetBackup and workload. By default CRL check is disabled for VIRTUALIZATION_CRL_CHECK flag.

Note: If NetBackup is upgraded from version 9.1 to 10.1, then user can delete the VIRTUALIZATION_CRL_CHECK flag which was enabled for CRL check between NetBackup and Snapshot Manager.

Specifying the CRL path

If you enable CRL validations, you need to specify the path to the directory containing revoked certificates of the external CA.

In the *ECA_CRL_PATH* parameter in the `/cloudpoint/openv/netbackup/bp.conf` file in the Snapshot Manager, specify the path to the directory where the certificate revocation lists (CRL) of the external CA are located. The path must be `/cloudpoint/eca/crl`.

If the *ECA_CRL_PATH* option is not specified, NetBackup downloads the CRLs from the URLs that are specified in the CRL Distribution Point (CDP) and uses them to verify revocation status of the peer host's certificate.

CA configuration for Azure Stack

You can sign the Azure Stack workloads with a different ECA than NetBackup. You can also configure in NBCA mode. You can have the following configurations:

1. **NetBackup configured with NBCA, Snapshot Manager configured with FlexsnapCA and Azure Stack configured with ECA:**
   - You need to configure the *ECA_TRUST_STORE_PATH* parameter in the `/cloudpoint/openv/netbackup/bp.conf` file.
   - The trust store file is available in `/cloudpoint/eca/trusted/cacerts.pem`. The trust store file is in PEM format.
   - Only the Azure stack appliance public root certificates must be present in the `/cloudpoint/eca/trusted/cacerts.pem` file. Manually append the Azure Stack appliance root public certificates in this file.
2. **NetBackup, Snapshot Manager configured with ECA-1 and Azure Stack is also configured with ECA-1:**
   - No manual step required since Snapshot Manager registration with NetBackup will take care of adding *ECA_TRUST_STORE_PATH* in `/cloudpoint/openv/netbackup/bp.conf` file.
   - Required CA certificates are already present in `/cloudpoint/eca/trusted/cacerts.pem` file.
3. **NetBackup, Snapshot Manager configured with ECA-1 and Azure Stack is configured with ECA-2:**
   - Required NetBackup CA certificates are already present in `/cloudpoint/eca/trusted/cacerts.pem` file.

- Manually append the Azure Stack appliance root public certificates in the same file.
- The file must now contain NetBackup and Azure Stack appliance public root certificates.

4. **Azure Stack is configured with well known public CA**:

   No manual steps are required at Snapshot Manager end.

Securing the connection to Snapshot Manager

In the Snapshot Manager, you can upload CRLs of the external CA at `/cloudpoint/eca/crl`. The uploaded CRL does not work, if the `crl` directory is not present or empty.

For data mover container, add this path against the *ECA_CRL_PATH* parameter in the `/cloudpoint/openv/netbackup/bp.conf` file.

Following three parameters are tuneable, you can add the entry under *eca* section in the `/cloudpoint/flexsnap.conf` file.

Table 9-1 ECA parameters

Parameter	Default	Value	Remarks
eca_crl_check	0 (Disabled)	0 (disabled) 1 (leaf) 2 (chain)	Certificate check level. Used to control the CRL/OCSP validation level for Snapshot Manager host connecting to On-prem/cloud workloads. ■ **0 (disabled)**: No CRL/OCSP is performed during validation ■ **1 (leaf)**: CRL/OSCP validation is performed only for leaf ■ **2 (chain)**: CRL/OSCP validation is performed for the whole chain
eca_crl_refresh_hours	24	Numerical value between 0 and 4830	Time interval in hours to update the Snapshot Manager CRLs cache from CA through the certificate CDP URL. Option is not applicable if `/cloudpoint/eca/crl` is present and contains CRL files. If it is set as 0, cache does not refresh.

Table 9-1 ECA parameters *(continued)*

Parameter	Default	Value	Remarks
eca_crl_path_sync_hours	1	Numerical value between 1 and 720	Time interval in hours to update the Snapshot Manager CRL cache from `/cloudpoint/eca/crl`. Option is not applicable if `/cloudpoint/eca/crl` is not present or empty.

Note: Cache is invalidated if any of ECA tuneable are added or modified manually inside the /cloudpoint/flexsnap.conf .

Note: The scope of CRL check is limited to Azure Stack only.

Section 2

NetBackup Snapshot Manager maintenance

Chapter 10

NetBackup Snapshot Manager logging

This chapter includes the following topics:

- About Snapshot Manager logging mechanism
- How Fluentd-based Snapshot Manager logging works
- Snapshot Manager logs
- Agentless logs
- Troubleshooting Snapshot Manager logging

About Snapshot Manager logging mechanism

Snapshot Manager uses the Fluentd-based logging framework for log data collection and consolidation. Fluentd is an open source data collector that provides a unified logging layer for structured log data collection and consumption.

For more information on Fluentd, refer to Fluentd website.

All the Snapshot Manager container services generate and publish service logs to the configured Docker logging driver. The logging driver is the fluentd framework that is running as a separate `flexsnap-fluentd` container on the Snapshot Manager host. With the Fluentd framework, these individual service logs are now structured and routed to the Fluentd data collector from where they are sent to the configured output plug-ins. The MongoDB collection and the flexsnap-fluentd container logs are the two output plug-ins that are configured by default.

Using Fluentd-based logging provides several benefits including the following:

- A persistent structured repository that stores the logs of all the Snapshot Manager services

- A single stream of all Snapshot Manager logs (vs disparate individual log files) makes it easy to trail and monitor specific logs
- Metadata associated with the logs allow for a federated search that speeds up troubleshooting
- Ability to integrate and push Snapshot Manager logs to a third-party tool for analytics and automation

How Fluentd-based Snapshot Manager logging works

When you install or upgrade Snapshot Manager, the following changes occur on the Snapshot Manager host:

- A new container service named `flexsnap-fluentd` is started on the Snapshot Manager host. This service is started before all the other Snapshot Manager container services. The `flexsnap-fluentd` service serves as the `fluentd` daemon on the host.
- All the Snapshot Manager container services are then started with `fluentd` as the Docker logging driver.
- A `fluentd` configuration file is created at `/cloudpoint/fluent/fluent.conf`. This file contains the output plug-in definitions that are used to determine where the Snapshot Manager logs are redirected for consumption.

Once all the infrastructure components are ready, each of the Snapshot Manager services begin to send their respective log messages to the configured Docker `fluentd` logging driver. The `fluentd` daemon then redirects the structured logs to the output plug-ins configured in the `fluentd` configuration file. These logs are then sent to the `/cloudpoint/logs/flexsnap.log` file on the Snapshot Manager host.

Note that the `flexsnap.log` file gets rotated after the file size reaches a maximum of 100 MB. A total of 30 generations (rotated files) of the `flexsnap.log` file are maintained. These conditions are applicable because of the new log file rotate (`log-rotate-age`) and log size (`log-rotate-size`) command options that are introduced in the fluentd command.

About the Snapshot Manager fluentd configuration file

Fluentd uses a configuration file that defines the source of the log messages, the set of rules and filters to use for selecting the logs, and the target destinations for delivering those log messages.

The `fluentd` daemon running on the Snapshot Manager host is responsible for sending the Snapshot Manager logs to various destinations. These target destinations, along with the other details such as input data sources and required fluentd parameters are defined in the plug-in configuration file. For Snapshot Manager, these plug-in configurations are stored in a `fluentd` configuration file that is located at `/cloudpoint/fluent/fluent.conf` on the Snapshot Manager host. The fluentd daemon reads the output plug-in definition from this configuration file to determine where to send the Snapshot Manager log messages.

The following output plug-in definition is added to the configuration file by default:

`STDOUT`: This is used to send the Snapshot Manager log messages to `/cloudpoint/logs/flexsnap.log`.

The plug-in is defined as follows:

```
# Send to fluentd docker logs
<store>
@type stdout
</store>
```

Additionally, the Snapshot Manager fluentd configuration file includes plug-in definitions for the following destinations:

- MongoDB
- Splunk
- ElasticSearch

These plug-in definitions are provided as a template and are commented out in the file. To configure an actual MongoDB, Splunk, or ElasticSearch target, you can uncomment these definitions and replace the parameter values as required.

Modifying the fluentd configuration file

Modify the `fluent.conf` configuration file if you want to modify the existing plug-in definitions.

To modify the fluent.conf file

1. On the Snapshot Manager host, open the `/cloudpoint/fluent/fluent.conf` configuration file in a text editor of your choice and then edit the contents to add or remove a plug-in definition.
2. Save all the changes to the file.
3. Restart the `flexsnap-fluentd` container service using the following command:

   ```
   # sudo docker restart flexsnap-fluentd
   ```

Note that the changes take effect immediately and are applicable only to the newer log messages that get generated after the change. The file changes do not apply to the older logs that were generated before the configuration file was updated.

Snapshot Manager logs

Snapshot Manager maintains the following logs that you can use to monitor Snapshot Manager activity and troubleshoot issues, if any. The logs are stored at `<install_path>/cloudpoint/logs` on the Snapshot Manager host.

Table 10-1 Snapshot Manager log files

Log	Description
`/cloudpoint/logs/flexsnap.log`	This log file contains all the product logs.
`/cloudpoint/logs/flexsnap-cloudpoint.log`	This log file contains all the Snapshot Manager installation related logs.
`/cloudpoint/logs/flexsnap-ipv6config.log`	This log file contains all the IPv6 related logs.

Logs for backup from snapshot and restore from backup jobs.

Navigate to: `/cloudpoint/openv/dm/datamover.<id>`

Here, logs can be found in the following directories: `logs`, `opt` and the `netbackup`.

- `nbpxyhelper` and `nbsubscriber` logs can be found inside the `logs` directory
- `VRTSpbx` logs can be found inside the `opt` directory
- `bpbkar, bpcd, bpclntcmd, nbcert, vnetd, vxms` and all other services logs can be found inside `netbackup` directory

To increase logging verbosity, `bp.conf` and `nblog.conf` files can be updated on Snapshot Manager at `/cloudpoint/openv/netbackup`. See *NetBackup Logging Reference Guide*

Changes to the `bp.conf` and `nblog.conf` files come to effect when the next backup from snapshot or restore job runs.

Log retention

The default configuration for datamover logs is as follows:

- Log retention maximum period is 30 days. Logs older than 30 days are deleted.

- The default configuration for high and low water marks for datamover logs is 70% and 30% of the size of "/cloudpoint" mount point. For example, if the usable size of the `/cloudpoint` folder is 30 GB, then the high water mark is 21 GB (70%) and low water mark is 9GB (30%). In case, the logs directory (`/cloudpoint/openv/dm/`) size reaches to high water mark, older logs for which the datamover containers are cleaned up and no longer running are considered for deletion. The logs are deleted for such datamover containers until low water mark is reached or no logs are remaining for the datamover containers cleaned up or no longer running.

Modifying the default configuration:

You can modify the default configuration for log retention by adding such a section in the `flexsnap.conf` on the primary Snapshot Manager. Open the `flexsnap.conf` file from the path `/cloudpoint/flexsnap.conf` and add the following section:

```
[datamover]
high_water_mark = 50
low_water_mark = 20
log_retention_in_days = 60
```

In case of Snapshot Manager extensions, the configuration from the primary server are used. Once the configuration is changed in primary Snapshot Manager, the configuration is updated on each Snapshot Manager extension within one hour. It is not possible to have separate custom configurations for primary Snapshot Manager or the Snapshot Manager extensions and configurations should only be changed in the primary Snapshot Manager. Though the configuration is same for primary Snapshot Manager and Snapshot Manager extensions, the high water mark and low water mark for log size are calculated based on the `/cloudpoint` directory mounted on each primary Snapshot Manager or Snapshot Manager extensions.

Snapshot Manager extension logs

Each Snapshot Manager extension maintains the logs under its own `/cloudpoint/logs` location.

- VM-based extension logs: Under the directory `/cloudpoint/logs`.
- Managed Kubernetes cluster-based extension logs: Under the directory `/cloudpoint/logs` which belongs to a file share.

Agentless logs

Logs for agentless connection to cloud instance(s) are present on the cloud instance at following locations based on the platform:

- **Linux**: `/tmp/ directory`
- **Windows**: `C:\\ProgramData\\Veritas\\CloudPoint\\logs\\`

Troubleshooting Snapshot Manager logging

You can retrieve the logs of a Snapshot Manager service from the `/cloudpoint/logs/flexsnap.log` file by running the following command:

```
# sudo cat /cloudpoint/logs/flexsnap.log | grep <flexsnap-service name>
```

Chapter 11

Upgrading NetBackup Snapshot Manager

This chapter includes the following topics:

- About Snapshot Manager upgrades
- Supported upgrade path
- Upgrade scenarios
- Preparing to upgrade Snapshot Manager
- Upgrading Snapshot Manager
- Upgrading Snapshot Manager using patch or hotfix
- Migrating and upgrading Snapshot Manager
- Post-upgrade tasks
- Post-migration tasks

About Snapshot Manager upgrades

You should not use two versions of Snapshot Manager on two different hosts to manage the same assets.

When you upgrade Snapshot Manager, all the snapshot data and configuration data from your previous version is maintained in the external `/cloudpoint` data volume. Veritas recommends that you upgrade Snapshot Manager on the same host or on a different host to which the Snapshot Manager data volume of the previous version is attached.

Supported upgrade path

Table 11-1 Snapshot Manager upgrade path

Upgrade from version	Upgrade to version
9.1/9.1.0.1	10.0
8.3/9.0/9.0.0.1	9.1/9.1.0.1 upgraded to 10.0
9.1/9.1.0.1	10.0.0.1 upgraded to 10.1

Notes:

- Direct upgrade from older versions to 10.1 is not supported. We need to first upgrade to 9.1 before upgrading to 10.1 for any upgrade path.
- Upgrading Snapshot Manager across the OS versions is not supported. If you are using Snapshot Manager on a RHEL7.x host, then you can only migrate it to a RHEL 8.6 or 8.4 host. Then follow the upgrade paths mentioned in the above table for upgrading Snapshot Manager on a RHEL 8.6 or 8.4 host. See “Migrating and upgrading Snapshot Manager” on page 191., for more information on migrating and upgrading Snapshot Manager on RHEL.
- See “Upgrade scenarios” on page 180., for more information on upgrading NetBackup 8.3.x to NetBackup 10.1.

Upgrade scenarios

The following table lists the Snapshot Manager upgrade scenarios.

Note: For the NetBackup version 10.1, NetBackup (primary, media) server and Snapshot Manager version should be at the same level. During upgrade, first upgrade Snapshot Manager and then upgrade NetBackup server.

Table 11-2 Upgrade scenarios

Scenario	Description	Action
Full upgrade from NetBackup 8.3 or 9.0 to NetBackup 9.1 or later	If you plan to upgrade NetBackup to 9.1 or later that includes upgrading all Snapshot Manager servers.	■ Disable Snapshot Manager servers ■ Upgrade Snapshot Manager servers ■ Upgrade NetBackup primary server ■ Then enable Snapshot Manager servers See “Upgrading Snapshot Manager” on page 183. **Note:** If you do not plan to upgrade one or more Snapshot Manager servers, then you must disable them using the NetBackup Web UI. In that case, any assets associated with the disabled Snapshot Manager servers cannot be protected by NetBackup.
Only Snapshot Manager upgrades to version 9.1 or later	If you plan to upgrade only the Snapshot Manager servers to 9.1 or later, but do not plan to upgrade NetBackup to 9.1 or later.	■ Contact Veritas Technical Support to obtain an Emergency Engineering Binary (EEB) to support the incompatibility between the Snapshot Manager and NetBackup versions. ■ Disable Snapshot Manager servers ■ Apply the EEB patch on the NetBackup primary server and associated media servers. ■ Upgrade Snapshot Manager servers ■ Then enable Snapshot Manager servers See “Upgrading Snapshot Manager using patch or hotfix” on page 189.
Upgrading to NetBackup version 10.1	If your NetBackup 8.3.x server has Snapshot Manager, you must first upgrade Snapshot Manager to NetBackup 9.1.x before you upgrade to NetBackup 10.1. Then you can proceed to upgrade NetBackup 8.3.x to NetBackup 10.1.	The process for this upgrade is: ■ Disable the Snapshot Manager server for maintenance in the NetBackup web UI. ■ Upgrade the Snapshot Manager server from NetBackup 8.3.x to NetBackup 9.1.x. ■ Upgrade the Snapshot Manager server from NetBackup 9.1.x to NetBackup 10.1. ■ Enable the Snapshot Manager server in the NetBackup web UI. ■ Upgrade the NetBackup server from 8.3.x directly to 10.1. ■ Upgrade the media server to 10.1 if it has been configured with storage units.
Migrating VM based Snapshot Manager to Kubernetes deployment	If you plan to migrate your VM based Snapshot Manager to a managed Kubernetes cluster.	For the the complete procedure, refer to the "Migration and upgrade of Snapshot Manager" section of *NetBackup™ Deployment Guide for Azure Kubernetes Services (AKS) Cluster*.

Table 11-2 Upgrade scenarios *(continued)*

Scenario	Description	Action
Migrating and upgrading the Snapshot Manager on RHEL	If you plan to migrate and upgrade Snapshot Manager on RHEL 8.6 or 8.4	See “Migrating and upgrading Snapshot Manager” on page 191.

Preparing to upgrade Snapshot Manager

Note the following before you upgrade

- Ensure that the Snapshot Manager instance, virtual machine, or physical host meets the requirements of the Snapshot Manager version you are upgrading to.
  See “ Meeting system requirements” on page 17.
- Ensure that the ports required by NetBackup server meet the requirements as mentioned in the *Required Ports* section of the following chapter:
  See “Preparing Snapshot Manager for backup from snapshot jobs” on page 34.
- When you upgrade Snapshot Manager, all the snapshot data and configuration data from your previous version is maintained in the external `/cloudpoint` data volume. This information is external to the Snapshot Manager container and the image and is preserved during the upgrade.
  However, you can take a backup of all the data in the `/cloudpoint` volume during the upgrade process when prompted or manually, if required.
  See “Backing up Snapshot Manager” on page 209.
- Ensure that no jobs are running on Snapshot Manager.
  - If you are using NetBackup Web UI, disable the Snapshot Manager server and wait for all the in-progress jobs to complete. Use the `nbstlutil` command to cancel all the pending SLP operations. Use one of the following commands:
    - To cancel the pending SLP operation for a specific image, use `nbstlutil cancel -backupid <value>`
    - To cancel the pending SLP operation for images that belong to specific lifecycle, use `nbstlutil cancel -lifecycle <name>`
  - If you are using NetBackup Administration console (Java UI), on the NetBackup primary server, run the following command to stop all NetBackup processes:
    - UNIX: `/usr/openv/netbackup/bin/bp.kill_all`

  - Windows: `install_path\NetBackup\bin\bpdown -f`

- After you upgrade Snapshot Manager, if required you can upgrade the NetBackup primary server. Also, you must enable the Snapshot Manager server from NetBackup Web UI.
- After upgrading, all the Snapshot Manager that you want to use for backup from snapshot or restore from backup jobs, must be re-edited by providing a token so that NetBackup certificates are generated in the Snapshot Manager. See *Edit a Snapshot Manager* section, in the *NetBackup Web UI Cloud Administrator's Guide*.

Upgrading Snapshot Manager

The following procedures describe how to upgrade your Snapshot Manager deployment. During the upgrade, you replace the container that runs your current version of Snapshot Manager with a newer container.

To upgrade Snapshot Manager server in Podman/Docker environment

1. Download the Snapshot Manager upgrade installer.

   On the Snapshot Manager download page, click **Download Now** to download the Snapshot Manager installer.

   The Snapshot Manager software components are available in a package form. The file name has the following format:

   ```
   NetBackup_SnapshotManager_<version>.tar.gz
   ```

   Note: The actual file name may vary depending on the release version.

2. Copy the downloaded compressed image file to the computer on which you want to deploy Snapshot Manager.
3. Un-tar the image file and list the contents:

   ```
   # ls
   NetBackup_SnapshotManager_10.1.x.x.xxxx.tar.gz
   netbackup-flexsnap-10.1.x.x.xxxx.tar.gz
   flexsnap_preinstall.sh
   ```

4 Run the following command to prepare the Snapshot Manager host for installation:

```
# sudo ./flexsnap_preinstall.sh
```

The output resembles the following:

```
Executing the following changes on this node to prepare the

NetBackup Snapshot Manager for installation:
     1) Validate SELINUX
    2) Loading Snapshot Manager service images.
9a585888b624: Loading layer [==================================================
2e62066f7e63: Loading layer [==================================================
f685725593dc: Loading layer [==================================================
696db75055f2: Loading layer [==================================================
Loaded image: veritas/flexsnap-core:10.1.0.0.1005
231ab0b2c170: Loading layer [==================================================
8c28932ae9d0: Loading layer [==================================================
Loaded image: veritas/flexsnap-certauth:10.1.0.0.1005
943d68324d6c: Loading layer [==================================================
191891ecc4f9: Loading layer [==================================================
Loaded image: veritas/flexsnap-nginx:10.1.0.0.1005
ceec3f55b2db: Loading layer [==================================================
Loaded image: veritas/flexsnap-idm:10.1.0.0.1005
1dea31649399: Loading layer [==================================================
fed7c9c63244: Loading layer [==================================================
085e29ab3a40: Loading layer [==================================================
db18b0c91f2b: Loading layer [==================================================
Loaded image: veritas/flexsnap-deploy:10.1.0.0.1005
439f0da098cd: Loading layer [==================================================
330348b98074: Loading layer [==================================================
Loaded image: veritas/flexsnap-rabbitmq:10.1.0.0.1005
2070f36290f4: Loading layer [==================================================
faca6ad364c3: Loading layer [==================================================
b0f22c31174d: Loading layer [==================================================
Loaded image: veritas/flexsnap-api-gateway:10.1.0.0.1005
7bb7b547ef29: Loading layer [==================================================
244ad9d09146: Loading layer [==================================================
c10ebc736986: Loading layer [==================================================
Loaded image: veritas/flexsnap-fluentd:10.1.0.0.1005
e48739b330f1: Loading layer [==================================================
5686c5f93e1a: Loading layer [==================================================
8ecac31f1564: Loading layer [==================================================
5fd91c9cd7b3: Loading layer [==================================================
```

```
ff641dd08e00: Loading layer [==================================================
Loaded image: veritas/flexsnap-datamover:10.1.0.0.1005
6f063a66d20a: Loading layer [==================================================
0ae89ddd7a56: Loading layer [==================================================
00fd9132896d: Loading layer [==================================================
Loaded image: veritas/flexsnap-mongodb:10.1.0.0.1005
```

Note: The output is truncated to fit the page.

5 Verify that there are no protection policy snapshots or other operations in progress and then stop Snapshot Manager by running the following command:

For Podman

```
# podman run -it --rm --privileged
-v /cloudpoint:/cloudpoint
-v /run/podman/podman.sock:/run/podman/podman.sock
veritas/flexsnap-deploy:<current_version> stop
```

For Docker

```
# docker run -it --rm --privileged
-v /cloudpoint:/cloudpoint
-v /run/docker/docker.sock:/run/docker/docker.sock
veritas/flexsnap-deploy:<current_version> stop
```

Here, *current_version* represents the currently installed Snapshot Manager version.

Note: Ensure that you enter the command without any line breaks.

The Snapshot Manager containers are stopped one by one. Messages similar to the following appear on the command line:

```
Stopping the services
Stopping container: flexsnap-core-system-0-0 ...done
Stopping container: flexsnap-core-indexing-0-0 ...done
Stopping container: flexsnap-core-general-0-0 ...done
Stopping container: flexsnap-core ...done
Stopping container: flexsnap-nginx ...done
Stopping container: flexsnap-core ...done
Stopping container: flexsnap-core ...done
Stopping container: flexsnap-scheduler ...done
Stopping container: flexsnap-idm ...done
Stopping container: flexsnap-core ...done
Stopping container: flexsnap-core ...done
Stopping container: flexsnap-core ...done
Stopping container: flexsnap-api-gateway ...done
Stopping container: flexsnap-certauth ...done
Stopping container: flexsnap-rabbitmq ...done
Stopping container: flexsnap-mongodb ...done
Stopping container: flexsnap-fluentd ...done
```

Wait for all the Snapshot Manager containers to be stopped and then proceed to the next step.

6 Depending on the environment, upgrade Snapshot Manager by running the following command:

- *For Podman*

```
# podman run -it --rm -u 0
-v /cloudpoint:/cloudpoint
-v /run/podman/podman.sock:/run/podman/podman.sock
veritas/flexsnap-deploy:<new_version> install
```

For an unattended installation, use the following command:

```
# podman run -it --rm -u 0
-v /cloudpoint:/cloudpoint
-v /run/podman/podman.sock:/run/podman/podman.sock
veritas/flexsnap-deploy:<new_version> install -y
```

- *For Docker*

```
# sudo docker run -it --rm --privileged -u 0 -v /cloudpoint:/cloudpoint
-v /var/run/docker.sock:/var/run/docker.sock
veritas/flexsnap-deploy:<new_version> install
```

For an unattended installation, use the following command:

```
# sudo docker run -it --rm --privileged -u 0 -v /cloudpoint:/cloudpoint
-v /var/run/docker.sock:/var/run/docker.sock
veritas/flexsnap-deploy:<new_version> install -y
```

Here, *new_version* represents the Snapshot Manager version you are upgrading to, for example '10.1.0.0.1005'

The `-y` option passes an approval for all the subsequent installation prompts and allows the installer to proceed in a non-interactive mode.

Note: Ensure that you enter the command without any line breaks.

---

7 The installer first loads the individual service images and then launches them in their respective containers.

The output resembles the following, here as an example the Podman environment output is provided:

```
Installing the services
Configuration started at time: Mon May  3 11:57:33 UTC 2021
podman server version: 2.0.5 Supported: true
This is an upgrade to NetBackup Snapshot Manager 10.1.0.0.1005
Previous CloudPoint version: 10.0.0.0.9800
Do you want to take a backup of the Snapshot Manager metadata prior to upgrade?
(y/n): y
Taking backup of Snapshot Manager metadata...done
Backup completed successfully.
Backup file located at /cloudpoint/backup/cloudpoint_9.0.0.0.9234.tar.gz.
[Storing /cloudpoint/keys/idm_store]
[Storing /cloudpoint/keys/flexsnap-idm_store]
Creating network: flexsnap-network ...done
Starting container: flexsnap-fluentd ...done
Starting container: flexsnap-mongodb ...done
Starting container: flexsnap-rabbitmq ...done
Starting container: flexsnap-certauth ...done
Starting container: flexsnap-api-gateway ...done
Starting container: flexsnap-core ...done
Starting container: flexsnap-core ...done
Starting container: flexsnap-core ...done
Starting container: flexsnap-core ...done
Starting container: flexsnap-scheduler ...done
Starting container: flexsnap-core ...done
Starting container: flexsnap-core ...done
Starting container: flexsnap-idm ...done
Starting container: flexsnap-deploy ...done
Starting container: flexsnap-nginx ...done
Upgrade finished at time: Mon May  3 11:58:51 UTC 2021
Before using backups from cloud snapshots, re-register Snapshot Manager with the
NetBackup primary server
```

8 (Optional) Run the following command to remove the previous version images.

(*For Podman*) `# podman rmi -f <imagename>:<oldimage_tagid>`

(*For Docker*) `# docker rmi -f <imagename>:<oldimage_tagid>`

9 To verify that the new Snapshot Manager version is installed successfully:

See “Verifying that Snapshot Manager is installed successfully” on page 46.

10 This concludes the upgrade process. Verify that your Snapshot Manager configuration settings and data are preserved as is.

Upgrading Snapshot Manager using patch or hotfix

You can also upgrade your current Snapshot Manager server using a patch or a hotfix. All the considerations and steps that apply for a normal upgrade, also apply to the upgrade being done using a patch or a hotfix, except that instead of downloading a new Snapshot Manager image, you download the patch/hotfix binaries.

Contact Veritas Technical Support at https://www.veritas.com/content/support/en_US/contact-us to obtain an Emergency Engineering Binary (EEB) for patch/hotfix.

Following are the brief steps explained with an example. For the detailed upgrade procedures

See “Upgrading Snapshot Manager” on page 183.

Consider that the currently installed version is Snapshot Manager `10.1.0.0` and you are upgrading to a Snapshot Manager patch version `10.1.0.0.1005` on a RHEL8.6 system in a Podman/Docker environment.

To upgrade Snapshot Manager using a patch or a hotfix

1 Download the Snapshot Manager EEB obtained from Veritas Technical Support.

Example: `NetBackup_SnapshotManager_<version>.tar.gz`

2 Un-tar the image file and list the contents:

```
# ls
NetBackup_SnapshotManager_10.1.x.x.xxxx.tar.gz
netbackup-flexsnap-10.1.x.x.xxxx.tar.gz
flexsnap_preinstall.sh
```

3 Run the following command to prepare the Snapshot Manager host for installation:

```
# sudo ./flexsnap_preinstall.sh
```

4 Verify that there are no protection policy snapshots or other operations in progress and then stop Snapshot Manager by running the following command:

For Podman

```
# podman run -it --rm --privileged
-v /cloudpoint:/cloudpoint
-v /run/podman/podman.sock:/run/podman/podman.sock
veritas/flexsnap-deploy:<current_version> stop
```

For Docker

```
# docker run -it --rm --privileged
-v /cloudpoint:/cloudpoint
-v /run/docker/docker.sock:/run/docker/docker.sock
veritas/flexsnap-deploy:<current_version> stop
```

Here, *current_version* represents the currently installed Snapshot Manager version.

5 Depending on the environment, upgrade Snapshot Manager by running the following command:

- *For Podman*

```
# podman run -it --rm -u 0
-v /cloudpoint:/cloudpoint
-v /run/podman/podman.sock:/run/podman/podman.sock
veritas/flexsnap-deploy:<EEB_version> install
```

For an unattended installation, use the following command:

```
# podman run -it --rm -u 0
-v /cloudpoint:/cloudpoint
-v /run/podman/podman.sock:/run/podman/podman.sock
veritas/flexsnap-deploy:<EEB_version> install -y
```

- *For Docker*

```
# sudo docker run -it --rm --privileged -u 0 -v /cloudpoint:/cloudpoint -v
-v /var/run/docker.sock:/var/run/docker.sock
veritas/flexsnap-deploy:<new_version> install
```

For an unattended installation, use the following command:

```
# sudo docker run -it --rm --privileged -u 0 -v /cloudpoint:/cloudpoint -v
-v /var/run/docker.sock:/var/run/docker.sock
veritas/flexsnap-deploy:<new_version> install -y
```

Here, *EEB_version* represents the Snapshot Manager patch/hotfix version you are upgrading to.

The `-y` option passes an approval for all the subsequent installation prompts and allows the installer to proceed in a non-interactive mode.

Note: Ensure that you enter the command without any line breaks.

The installer first loads the individual service images and then launches them in their respective containers.

6 (Optional) Run the following command to remove the previous version images.

(*For Podman*) `# podman rmi -f <imagename>:<oldimage_tagid>`

(*For Docker*) `# docker rmi -f <imagename>:<oldimage_tagid>`

7 To verify that the new Snapshot Manager version is installed successfully:

See “Verifying that Snapshot Manager is installed successfully” on page 46.

8 This concludes the Snapshot Manager upgrade process using a patch or a hotfix . Verify that your Snapshot Manager configuration settings and data are preserved as is.

Migrating and upgrading Snapshot Manager

This section describes the procedure for migrating and upgrading the Snapshot Manager on RHEL.

Before you begin migrating Snapshot Manager

Make sure that you complete the following before installing Snapshot Manager:

- Ensure that your environment meets system requirements.
  See “ Meeting system requirements” on page 17.
- Create the instance on which you install Snapshot Manager or prepare the physical host.
  See “Verifying that specific ports are open on the instance or physical host” on page 33.
  See “Creating an instance or preparing the host to install Snapshot Manager” on page 30.
- Prepare a RHEL 8.6 or 8.4 host for installation. You can either upgrade your existing RHEL 7.x OS to RHEL 8.6 or 8.4 OS, or create a new system with RHEL 8.6 or 8.4.

- For upgrading the system from RHEL 7.x to RHEL 8.6 or 8.4, follow the Red Hat documentation.
- For creating a new system with RHEL 8.6 or 8.4, configure a Podman container platform
  See Table 2-10 on page 30.
  The brief steps include:
  - Setup the RHEL repos
    For AWS cloud, enable the extra repos
    ```
    # sudo yum-config-manager --enable
    rhui-REGION-rhel-server-extras
    ```
    For on-premise, enable your subscriptions:
    ```
    # sudo subscription-manager register --auto-attach
    --username=<username> --password=<password>
    ```
  - Install Podman if required:
    ```
    # sudo yum install -y podman
    ```
  - If SELinux is enabled, change the mode to permissive mode and restart the system.
    Edit the `/etc/selinux/config` configuration file and modify the `SELINUX` parameter value to `SELINUX=permissive`.

- Run the following commands to install the required packages (`lvm2`, `udev` and `dnsmaq`) on the hosts:
  ```
  #yum install -y lvm2-<version>
  #yum install -y lvm2-libs-<version>
  #yum install -y python3-pyudev-<version>
  #yum install -y systemd-udev-<version>
  #yum install -y podman-plugins
  ```
- Run the following commands to lock the Podman and Conmon versions to the supported versions, so that they do not get updated with the *yum* update:
  ```
  sudo yum install -y podman-2.2.1-7.module+el8.3.1+9857+68fb1526
  sudo yum install -y python3-dnf-plugin-versionlock
  ```
- Verify that specific ports are open on the instance or physical host.
  See “Verifying that specific ports are open on the instance or physical host” on page 33.

Next, you migrate Snapshot Manager from the RHEL 7.x host to the newly prepared RHEL 8.6 or 8.4 host.

See “Migrate and upgrade Snapshot Manager on RHEL 8.6 or 8.4” on page 193.

Migrate and upgrade Snapshot Manager on RHEL 8.6 or 8.4

Perform the following steps to migrate Snapshot Manager 10.0 or 10.0.0.1 from your RHEL 7.x host to the new RHEL 8.6 or 8.4 host.

To upgrade Snapshot Manager in docker environment

1. Download the Snapshot Manager upgrade installer.

   Example: `NetBackup_SnapshotManager_<version>.tar.gz`

2. Un-tar the image file and list the contents:

   ```
   # ls
   NetBackup_SnapshotManager_10.1.x.x.xxxx.tar.gz
   netbackup-flexsnap-10.1.x.x.xxxx.tar.gz
   flexsnap_preinstall.sh
   ```

3. Run the following command to prepare the Snapshot Manager host for installation:

   ```
   # sudo ./flexsnap_preinstall.sh
   ```

4. Upgrade Snapshot Manager by running the following command:

   ```
   # podman run -it --rm -u 0
   -v /cloudpoint:/cloudpoint
   -v /run/podman/podman.sock:/run/podman/podman.sock
   veritas/flexsnap-deploy:<new_version> install
   ```

   For an unattended installation, use the following command:

   ```
   # podman run -it --rm -u 0
   -v /cloudpoint:/cloudpoint
   -v /run/podman/podman.sock:/run/podman/podman.sock
   veritas/flexsnap-deploy:<new_version> install -y
   ```

   Here, *new_version* represents the Snapshot Manager version you are upgrading to.

   The `-y` option passes an approval for all the subsequent installation prompts and allows the installer to proceed in a non-interactive mode.

   Note: Ensure that you enter the command without any line breaks.

   The installer first loads the individual service images and then launches them in their respective containers.

5 (Optional) Run the following command to remove the previous version images.

```
# podman rmi -f <imagename>:<oldimage_tagid>
```

6 To verify that the new Snapshot Manager version is installed successfully:

See “Verifying that Snapshot Manager is installed successfully” on page 46.

To migrate Snapshot Manager in Podman environment

1 On the RHEL 7.x host, verify that there are no protection policy snapshots or other operations in progress and then stop Snapshot Manager by running the following command:

```
# sudo docker run -it --rm
-v /cloudpoint:/cloudpoint
-v /var/run/docker.sock:/var/run/docker.sock
veritas/flexsnap-deploy:<current_version> stop
```

Here, *current_version* represents the currently installed Snapshot Manager version.

Example:

```
# sudo docker run -it --rm -v /cloudpoint:/cloudpoint
-v /var/run/docker.sock:/var/run/docker.sock
veritas/flexsnap-deploy:9.1.0.0.9349 stop
```

Note: This is a single command. Ensure that you enter the command without any line breaks.

The Snapshot Manager containers are stopped one by one. Messages similar to the following appear on the command line:

```
Stopping the services
Stopping container: flexsnap-core.8a51aac1848c404ab61e4625d7b88703 ...done
Stopping container: flexsnap-core-long-15 ...done
Stopping container: flexsnap-core-long-14 ...done
Stopping container: flexsnap-core-long-13 ...done
Stopping container: flexsnap-core-long-12 ...done
Stopping container: flexsnap-core-long-11 ...done
Stopping container: flexsnap-core-long-10 ...done
Stopping container: flexsnap-core-long-9 ...done
Stopping container: flexsnap-core-long-8 ...done
Stopping container: flexsnap-core-long-7 ...done
Stopping container: flexsnap-core-long-6 ...done
Stopping container: flexsnap-core-long-5 ...done
Stopping container: flexsnap-core-long-4 ...done
Stopping container: flexsnap-core-long-3 ...done
Stopping container: flexsnap-core-long-2 ...done
Stopping container: flexsnap-core-long-1 ...done
Stopping container: flexsnap-core-long-0 ...done
Stopping container: flexsnap-core-15 ...done
Stopping container: flexsnap-core-14 ...done
```

```
Stopping container: flexsnap-core-13 ...done
Stopping container: flexsnap-core-12 ...done
Stopping container: flexsnap-core-11 ...done
Stopping container: flexsnap-core-10 ...done
Stopping container: flexsnap-core-9 ...done
Stopping container: flexsnap-core-8 ...done
Stopping container: flexsnap-core-7 ...done
Stopping container: flexsnap-core-6 ...done
Stopping container: flexsnap-core-5 ...done
Stopping container: flexsnap-core-4 ...done
Stopping container: flexsnap-core-3 ...done
Stopping container: flexsnap-core-2 ...done
Stopping container: flexsnap-core-1 ...done
Stopping container: flexsnap-core-0 ...done
Stopping container: flexsnap-nginx ...done
Stopping container: flexsnap-core ...done
Stopping container: flexsnap-core ...done
Stopping container: flexsnap-scheduler ...done
Stopping container: flexsnap-idm ...done
Stopping container: flexsnap-core ...done
Stopping container: flexsnap-core ...done
Stopping container: flexsnap-core ...done
Stopping container: flexsnap-api-gateway ...done
Stopping container: flexsnap-certauth ...done
Stopping container: flexsnap-rabbitmq ...done
Stopping container: flexsnap-mongodb ...done
Stopping container: flexsnap-fluentd ...done
```

Wait for all the Snapshot Manager containers to be stopped and then proceed to the next step.

2 Migrate the Snapshot Manager configuration data to the RHEL 8.6 or 8.4 host:

- If you have created a new system with RHEL 8.6 or 8.4:
  - Run the following command to unmount `/cloudpoint` from the current host.

    ```
    # umount /cloudpoint
    ```

  - Detach the data disk that was mounted on `/cloudpoint` mountpoint.

    Note: For detailed instructions to detach or attach the data disks, follow the documentation provided by your cloud or storage vendor.

- On the RHEL8.6 or 8.4 host, run the following commands to create and mount the disk:

  ```
  # mkdir /cloudpoint
  # mount /dev/<diskname> /cloudpoint
  ```

  For vendor-specific details
  See “Creating and mounting a volume to store Snapshot Manager data” on page 32.

- If you have upgraded from RHEL 7.x to RHEL 8.6 or 8.4, copy the `/cloudpoint` mountpoint data from RHEL 7.x system and move it to the RHEL8.6 or 8.4 system under `/cloudpoint` folder.

This concludes the Snapshot Manager migration process.

After migration, install the `new_version` on the new host by following the steps mentioned in the To upgrade Snapshot Manager in docker environment.

3 During migration process, if Snapshot Manager is migrated to another system or IP address is changed, then regenerate the certificates as follows:

- Stop the Snapshot Manager services using the following command:

  ```
  [root@ip-172-31-24-178 ec2-user]# podman run -it --rm
  --privileged -v /cloudpoint:/cloudpoint -v
  /run/podman/podman.sock:/run/podman/podman.sock
  veritas/flexsnap-deploy:10.0.0.9818 stop
  ```

- Regenerate the certificates using the following command:

  ```
  /cloudpoint/scripts/cp_regenerate_certs.sh -i <CP_IP_ADDRESS>
  -h <CP_HOSTNAME>

  Setting up certificate authority ...done
  Generating certificates for servers ...done
  Generating certificates for clients ...done
  Adding MongoDB and RabbitMQ certificate to the trust store ...[Storing /cl
  [Storing /cloudpoint/keys/flexsnap-idm_store]
  done
  Creating symlinks for nginx certificates ...done
  ```

- Start the Snapshot Manager services using the following command:

  ```
  [root@ip-172-31-24-178 ec2-user]# podman run -it --rm
  --privileged -v /cloudpoint:/cloudpoint -v
  /run/podman/podman.sock:/run/podman/podman.sock
  veritas/flexsnap-deploy:10.0.0.9818 start
  ```

4 Depending on the following appropriate scenario, update the `/cloudpoint/openv/netbackup/bp.conf` file to update the value of **CLIENT_NAME** to new Snapshot Manager IP/hostname.

- If IP address does not change, then edit the Snapshot Manager server entry and provide a reissue token generated for the Snapshot Manager host.
- If IP address changes, then disable the previous Snapshot Manager host, and add a Snapshot Manager host with new IP address. Then perform the following steps:
  - Revoke the certificate of previous Snapshot Manager host.
  - Add the mapping of new Snapshot Manager host IP/hostname into the previous Snapshot Manager host using the host mappings.
  - Generate reissue token by selecting previous Snapshot Manager host, then use that token to edit the new Snapshot Manager host. Old Snapshot Manager host **Certificate** entry and **Host Mapping** would be replaced.

5 After migrating Snapshot Manager to a RHEL 8.6 or 8.4 host, perform the following steps to upgrade Snapshot Manager to 10.1.

See “Upgrading Snapshot Manager” on page 183.

6 This concludes the migration and upgrade process for Snapshot Manager. Verify that your Snapshot Manager configuration settings and data are preserved as is.

Post-upgrade tasks

You may need to perform the following tasks after a successful upgrade of the Snapshot Manager server.

Post-upgrade tasks

1 Upgrade the Snapshot Manager agents on the Linux and Windows application hosts.

Note: If you are upgrading from Snapshot Manager 8.3 to 9.0 or 9.1, then you must manually upgrade the on-host agents. If you are upgrading from Snapshot Manager 9.0 to 9.1, upgrading the on-host agents is optional.

Perform the following steps to upgrade the agent on Linux hosts:

- Sign in to NetBackup UI and download the newer agent package. Navigate to **Cloud > Snapshot Managers > Actions > Add agent**.

- Stop the flexsnap agent service on the Linux host where you want to upgrade the agent.
  Run the following command on the Linux host:
  ```
  # sudo systemctl stop flexsnap-core.service
  ```
- Upgrade the agent on the Linux host.
  Run the following command on the Linux host:
  ```
  # sudo rpm -Uvh --force flexsnap_agent_rpm_name
  ```
  Here, *flexsnap_agent_rpm_name* is the name of the agent rpm package you downloaded earlier.
- Reload the daemon, if prompted.
  Run the following command on the Linux host:
  ```
  # sudo systemctl daemon-reload
  ```
- Repeat these steps on all the Linux hosts where you wish to upgrade the Linux-based agent.

Note the following:

When upgrading from CloudPoint agent to Flexsnap agent, uninstall CloudPoint agent first and then install the Flexsnap agent using the following recommended uninstallation and installation commands:

- Uninstallation: `sudo yum -y remove cloudpoint_agent_rpm_name`
- Installation: `sudo yum -y install flexnsap_agent_rpm_name`

Perform the following steps to upgrade the agent on Windows hosts:

- Sign in to NetBackup UI and download the newer agent package.
  Navigate to **Cloud > Snapshot Managers > Actions > Add agent**.
- Stop the Veritas Snapshot Manager Agent service that is running on the host.
- Run the newer version of the agent package file and follow the installation wizard workflow to upgrade the on-host agent on the Windows host.
  The installer detects the existing installation and upgrades the package to the new version automatically.
- Generate the token for agent configuration. Navigate to **NetBackup Web UI > Cloud > Snapshot Managers > Actions > Add agent > Create Token**.
- Repeat these steps on all the Windows hosts where you wish to upgrade the Windows-based agent.

For details on how to download the agent installation package from the NetBackup UI, refer to the following:

See “Downloading and installing the Snapshot Manager agent” on page 128.

2 If you want to run backup from snapshot and restore from backup jobs after upgrade, you must update the NetBackup configuration so that the upgraded Snapshot Manager configuration details are available with NetBackup. After upgrading, all the Snapshot Manager that you want to use for backup from snapshot or restore from backup jobs, must be re-edited by providing a token so that NetBackup certificates are generated. See *Edit a Snapshot Manager* section, in the *NetBackup Web UI Cloud Administrator's Guide*.

Perform one of the following actions:

- From the NetBackup Web UI, edit the Snapshot Manager server information.
  - In the Web UI, click **Workloads > Cloud** from the left navigation pane and then click the **Snapshot Manager servers** tab.
  - Select the Snapshot Manager server that you just upgraded, and then click **Edit** from the ellipsis action button on the right.
  - In the Edit Snapshot Manager server dialog, specify all the requested details.
  - Click **Validate** to validate the Snapshot Manager server certificate.
  - In the **Token** field enter the Standard Host Token.
  - Click **Save** to update the Snapshot Manager server configuration.
- Or, on the NetBackup primary server, run the following command:

```
./tpconfig -update -snapshot_manager <snapshot_manager_name>
-snapshot_manager_user_id <user_ID> -manage_workload
<manage_workload> [-requiredport <IP_port_number>]
[-security_token <token_value>]
```

Note: Additional option `-security_token` is required for updating Snapshot Manager which is managing cloud workloads. The token must be Standard host token. This is required for NetBackup certificates generation on Snapshot Manager.

On UNIX systems, the directory path to this command is `/usr/openv/volmgr/bin/`. On Windows systems, the directory path to this command is `install_path\Volmgr\bin\`. Refer to the *Veritas NetBackup Commands Reference Guide* for details.

- Or, make a PATCH API call to the NetBackup primary server using the following URL:

https://*primaryserver.domain.com*/netbackup/config/servers/snapshot-mgmt-servers/*cp-hostname*

3 After upgrading Snapshot Manager to version 10.0, the on-host agent must be restarted to discover and protect assets on LVM storage.

For more details about the `tpconfig` command and its options, refer to the *Veritas NetBackup Commands Reference Guide*.

Upgrading Snapshot Manager extensions

When Snapshot Manager is upgraded, all the extensions are automatically disabled. You must upgrade the extensions with the required Snapshot Manager version and enable them manually from the NetBackup Web UI.

Upgrading Snapshot Manager extensions on a managed Kubernetes cluster (AKS)

1 Permit the script to run as an executable:

```
# chmod +x cp_extension_start.sh
```

2 Run the command as follows:

```
# ./cp_extension.sh install

NetBackup Snapshot Manager image repository path. Format=<Login-server/image:
Snapshot Manager extension namespace: cloudpoint-system
Snapshot Manager extension token:
This is an upgrade of NetBackup Snapshot Manager Extension

Starting Snapshot Manager service deployment
customresourcedefinition.apiextensions.k8s.io/cloudpoint-servers.veritas.com
serviceaccount/cloudpoint-acc unchanged
clusterrole.rbac.authorization.k8s.io/cloudpoint-cloudpoint-system unchanged
clusterrolebinding.rbac.authorization.k8s.io/cloudpoint-rolebinding-cloudpoin
deployment.apps/flexsnap-deploy unchanged
Snapshot Manager service deployment ...done

customresourcedefinition.apiextensions.k8s.io/cloudpoint-servers.veritas.com
Generating Snapshot Manager Custom Resource Definition object
deployment "flexsnap-deploy" successfully rolled out
cloudpointrule.veritas.com/cloudpoint-config-rule configured
Snapshot Manager extension installation ...done
```

Executable way

- Permit the script to run as an executable:

  ```
  # chmod +x cp_extension_start.sh
  ```

- Run the installation command as follows:

  ```
  # ./cp_extension_start.sh install -i <target_image:tag> -n
  <namespace> -t <workflow_token>
  ```

  For example:

  ```
  # ./cp_extension_start.sh install -i
  mycontainer.azurecr.io/veritas/flexsnap-deploy:9.0.1.0.9271
  -n cloudpoint-system -t workflow
  3q3ou4jxiircp9tk0eer2g9jx7mwuypwz10k4i3sms2e7k4ee7-.....
  ```

Upgrade of Snapshot Manager extension on a managed Kubernetes cluster (AKS) in Azure

To improve the security in NetBackup 10.1, the process in data mover container are configured to launch with service (non-root) user. If file share is created with the **SMB** protocol then Backup from Snapshot, Index from Snapshot operations and so on would fail when data mover is launched for data movement operation. To resolve this issue, perform the following:

1. Take a backup of the logs from old file share or retain the old file share.
2. Uninstall the Snapshot Manager extension. Delete **Persistent Volume**, **ConfigMap** and **Secrets** from AKS extensions.
3. Install Snapshot Manager extension. While defining **StorageClass** consider using CSI provisioner for `Azure Files` with NFS protocol.

   See “Installing the Snapshot Manager extension on a managed Kubernetes cluster (AKS) in Azure” on page 57.

Upgrading Snapshot Manager extensions on a VM

1 Load required images:

- For docker environments: `# sudo docker load -i SnapshotManager_image_name`
- For podman environment, un-tar the image file:

  ```
  # gunzip VRTSflexsnap-podman-9.x.x.x.x.tar.gz
  ```

2 Run the following command to prepare the Snapshot Manager host for installation:

```
# ./flexsnap_preinstall.sh
```

3 Run the following respective command to upgrade VM extension:

- For docker environment:

  ```
  # sudo docker run -it --rm -u 0 -v
  /<full_path_to_volume_name>:/<full_path_to_volume_name> -v
  ```

```
/var/run/docker.sock:/var/run/docker.sock
veritas/flexsnap-deploy:<new_version> install_extension
```

- For podman environment:

```
# sudo podman run -it --rm -u 0 --privileged -v
/<full_path_to_volume_name>:/<full_path_to_volume_name> -v
/run/podman/podman.sock:/run/podman/podman.sock
veritas/flexsnap-deploy:<new_version> install_extension
```

Post-migration tasks

After migration, if the name is changed to Snapshot Manager, then perform the following steps for Linux and Windows on-host agent renews and then perform the plugin level discovery:

For Linux:

- Edit the `/etc/flexsnap.conf` file and update the targeted field with new IP/host of Snapshot Manager.
  For example,

```
[root@testVM]# cat /etc/flexsnap.conf
[global]
target = nbuxqa-alphaqa-10-250-172-172.vxindia.veritas.com
hostid = azure-vm-b5c2b769-256a-4488-a71d-f809ce0fec5d

[agent]
id = agent.c2ec74c967e043aaae5818e50a939556
```

- Perform the Linux on-host agent renew using the following command:
  `/opt/VRTScloudpoint/bin/flexsnap-agent--renew--token <auth_token>`
- Restart linux on-host agent using the following command:
  `sudo systemctl restart flexsnap-core.service`

For Windows:

- Edit the `\etc\flexsnap.conf` and update the targeted field with new IP/host of Snapshot Manager.
  For example,

```
[global]
target = nbuxqa-alphaqa-10-250-172-172.vxindia.veritas.com
hostid = azure-vm-427a67a0-6f91-4a35-abb0-635e099fe9ad
```

```
[agent]
id = agent.3e2de0bf17d54ed0b54d4b33530594d8
```

- Perform the Windows on-host agent renew using the following command:

  ```
  "c:\ProgramFiles\Veritas\CloudPoint\flexsnap-agent.exe"--renew--token
  <auth_token>
  ```

Chapter 12

Uninstalling NetBackup Snapshot Manager

This chapter includes the following topics:

- Preparing to uninstall Snapshot Manager
- Backing up Snapshot Manager
- Unconfiguring Snapshot Manager plug-ins
- Unconfiguring Snapshot Manager agents
- Removing the Snapshot Manager agents
- Removing Snapshot Manager from a standalone Docker host environment
- Removing Snapshot Manager extensions - VM-based or managed Kubernetes cluster-based
- Restoring Snapshot Manager

Preparing to uninstall Snapshot Manager

Note the following before you uninstall Snapshot Manager:

- Ensure that there are no active Snapshot Manager operations in progress. For example, if there are any snapshot, replication, restore or indexing jobs running, wait for them to complete.
  If you have configured policies, ensure that you stop the scheduled policy runs. You may even want to delete those policies.
- Ensure that you remove the Snapshot Manager agents that are installed on the application hosts. The application hosts are the systems where the applications that are being protected by Snapshot Manager are running.

See “Removing the Snapshot Manager agents” on page 214.

- Ensure that you disable the Snapshot Manager server from NetBackup. Depending on how you have set up your Snapshot Manager server, whether on-premise or in the cloud, you can disable Snapshot Manager server from the NetBackup Web UI .
  Refer to the *NetBackup Snapshot Manager for Data Center Administrator's Guide* for instructions.
- All the snapshot data and configuration data from your existing installation is maintained in the external `/cloudpoint` data volume. This information is external to the Snapshot Manager containers and images and is deleted after the uninstallation.
  You can take a backup of all the data in the `/cloudpoint` volume, if desired.
  See “Backing up Snapshot Manager” on page 209.

Backing up Snapshot Manager

If Snapshot Manager is deployed in a cloud

To back up Snapshot Manager when it is deployed in a cloud

1 Stop Snapshot Manager services.

Depending on the environment, use the following respective commands:

(*For Docker*) `# sudo docker run -it --rm -u 0 -v /full_path_to_volume_name:/full_path_to_volume_name -v /var/run/docker.sock:/var/run/docker.sock veritas/flexsnap-deploy:version stop`

(*For Podman*) `# sudo podman run -it --rm -u 0 -v /full_path_to_volume_name:/full_path_to_volume_name -v /var/run/podman.sock:/var/run/podman.sock veritas/flexsnap-deploy:version stop`

Here, *version* represents the currently installed Snapshot Manager product version. You can retrieve the version using the following command:

```
# cat /cloudpoint/version
```

As an example following is the command for docker environment:

```
# sudo docker run -it --rm -u 0 -v /cloudpoint:/cloudpoint -v
/var/run/docker.sock:/var/run/docker.sock
veritas/flexsnap-deploy:10.1.0.0.1005 stop
```

Note: This is a single command. Ensure that you enter the command without any line breaks.

---

2 Ensure that all Snapshot Manager containers are stopped. This step is important because all activity and connections to and from Snapshot Manager must be stopped to get a consistent Snapshot Manager backup.

Enter the following:

(*For Docker*) `# sudo docker ps | grep veritas`

(*For Podman*) `# sudo podman ps | grep veritas`

This command should not return any actively running Snapshot Manager containers.

3. (Optional) If you still see any active containers, repeat step 2. If that does not work, run the following command on each active container:

   (*For Docker*) `# sudo docker kill container_name`

   (*For Podman*) `# sudo podman kill container_name`

   As an example following is the command for docker environment:

   ```
   # sudo docker kill flexsnap-api
   ```

4. After all the containers are stopped, take a snapshot of the volume on which you installed Snapshot Manager. Use the cloud provider's snapshot tools.

5. After the snapshot completes, restart Snapshot Manager services.

   Use the following command:

   (*For Docker*) `# sudo docker run -it --rm -u 0 -v /full_path_to_volume_name:/full_path_to_volume_name-v /var/run/docker.sock:/var/run/docker.sock veritas/flexsnap-deploy:version start`

   (*For Podman*) `# sudo podman run -it --rm -u 0 -v /full_path_to_volume_name:/full_path_to_volume_name-v /var/run/podman.sock:/var/run/podman.sock veritas/flexsnap-deploy:version start`

   Here, *version* represents the currently installed Snapshot Manager product version.

   As an example following is the command for docker environment:

   ```
   # sudo docker run -it --rm -u 0 -v /cloudpoint:/cloudpoint -v
   /var/run/docker.sock:/var/run/docker.sock
   veritas/flexsnap-deploy:10.1.0.0.1005 start
   ```

   Note: This is a single command. Ensure that you enter the command without any line breaks.

If Snapshot Manager is deployed on-premises

To backup Snapshot Manager when it is deployed on-premise

1 Stop Snapshot Manager services.

Use the following command:

(*For Docker*) `# sudo docker run -it --rm -u 0 -v /full_path_to_volume_name:/full_path_to_volume_name -v /var/run/docker.sock:/var/run/docker.sock veritas/flexsnap-deploy:version stop`

(*For Podman*) `# sudo podman run -it --rm -u 0 -v /full_path_to_volume_name:/full_path_to_volume_name -v /var/run/podman.sock:/var/run/podman.sock veritas/flexsnap-deploy:version stop`

Here, *version* represents the currently installed Snapshot Manager product version.

As an example following is the command for docker environment:

```
# sudo docker run -it --rm -u 0 -v /cloudpoint:/cloudpoint -v
/var/run/docker.sock:/var/run/docker.sock
veritas/flexsnap-deploy:10.1.0.0.1005 stop
```

Note: This is a single command. Ensure that you enter the command without any line breaks.

---

2 Ensure that all Snapshot Manager containers are stopped. This step is important because all activity and connections to and from Snapshot Manager must be stopped to get a consistent Snapshot Manager backup.

Enter the following:

(*For Docker*) `# sudo docker ps | grep veritas`

(*For Podman*) `# sudo podman ps | grep veritas`

This command should not return any actively running Snapshot Manager containers.

3 (Optional) If you still see any active containers, repeat step 2. If that does not work, run the following command on each active container:

(*For Docker*) `# sudo docker kill container_name`

(*For Podman*) `# sudo podman kill container_name`

As an example following is the command for docker environment:

```
# sudo docker kill flexsnap-api
```

4 Back up the folder `/cloudpoint`. Use any backup method you prefer.

For example:

```
# tar -czvf cloudpoint_dr.tar.gz /cloudpoint
```

This command creates a compressed archive file named `cloudpoint_dr.tar.gz` that contains the data in the `/cloudpoint` directory.

Unconfiguring Snapshot Manager plug-ins

Snapshot Manager plug-ins allow Snapshot Manager to discover the assets on the host so that you can protect those assets by taking snapshots. If required, you can remove a Snapshot Manager plug-in configuration using the NetBackup UI.

Before you remove a plug-in configuration from the host, consider the following:

- You must remove all the snapshots of the assets that are related to the plug-in that you wish to unconfigure.
  Plug-in unconfiguration fails if asset snapshots exist.
- Unconfiguring a plug-in removes the plug-in from the selected host. To protect the plug-in related assets on the same host again, you will have to reconfigure that plug-in on the host.
- Once you unconfigure a plug-in, all the assets that are related to the plug-in are removed from the Snapshot Manager configuration and you will no longer be able to protect those assets.

To unconfigure a plug-in from a host

1 Sign in to the NetBackup UI.

2 Verify that you have removed all the plug-in related asset snapshots.

3. From the menu on the left, click **Workloads > Cloud** and then click the **Virtual machines** tab.
4. On the Virtual machines tab, select the host where you want unconfigure the agent and then from the menu bar that appears at the top, click **Unconfigure**.

   Snapshot Manager unconfigures the plug-in from the host. Observe that the **Unconfigure** button now changes to **Configure**. This indicates that the plug-in unconfiguration is successful on the host.

Unconfiguring Snapshot Manager agents

To enable Snapshot Manager to protect assets on a remote host, you first need to establish a connection between the Snapshot Manager server and the remote host. Depending on how the connection is configured (either with agents or using the agentless feature), Snapshot Manager uses agents that manage the plug-ins that are used to discover all the assets and perform the operations on the host.

Whenever you configure a remote host for protection, the agent registration and the plug-in configuration information is added to the Snapshot Manager database on the Snapshot Manager server. You can, if required, remove an agent entry from the Snapshot Manager database by performing the disconnect operation from the NetBackup UI.

Before you unconfigure an agent, consider the following:

- Once you unconfigure an agent, you cannot re-configure a Snapshot Manager plug-in on the same host, if you had installed the Snapshot Manager agent on that host. To be able to configure a plug-in on the host again, you must first uninstall the agent package from the host, connect the host and install and register the agent with the Snapshot Manager server again.
- You must first unconfigure the Snapshot Manager plug-in from the host before you proceed with the disconnect operation. The disconnect option is not enabled if a Snapshot Manager plug-in is configured on the host.
- Unconfiguring an agent entry from the Snapshot Manager server does not uninstall the agent package from the host. You have to manually remove the agent binaries from the host after completing the disconnect operation.
- Once you unconfigure an agent, all the file system assets that belong to that host are removed from the Snapshot Manager configuration.

To unconfigure the agent entry from the Snapshot Manager server

1. Sign in to the NetBackup UI.
2. Remove Snapshot Manager plug-in configuration from the host that you wish to disconnect.

   See “Unconfiguring Snapshot Manager plug-ins” on page 212.
3. From the menu on the left, click **Workloads > Cloud** and then click the **Virtual machines** tab.
4. On the Virtual machines tab, select the host where you want unconfigure the agent and then from the menu bar that appears at the top, click **Disconnect**.

   Snapshot Manager begins to unconfigure the agent. Observe that the Disconnect button now changes to Connect. This indicates that the disconnect operation is successful and the agent has been unconfigured successfully.

   The agent registration and all the assets information about that host is completely removed from the database.
5. The next step is to manually uninstall the agent from the host on which you performed the disconnect operation. This is required if you wish to protect this host and its assets using Snapshot Manager at a later time.

   See “Removing the Snapshot Manager agents” on page 214.

Removing the Snapshot Manager agents

You must first remove the Snapshot Manager agents before you remove Snapshot Manager. The agents are installed directly on the host where the applications are running. Snapshot Manager agents manage the Snapshot Manager plug-ins that discover assets and perform snapshot operations on the host.

To uninstall the Snapshot Manager on-host agents

1. Connect to the host where you have installed the Snapshot Manager agent.

   Ensure that the user account that you use to connect has administrative privileges on the host.

2. For Linux-based agent, perform the following:

   Remove the .rpm package using the following command:

   ```
   # sudo yum -y remove <snapshotmanager_agent_package>
   ```

   Here, <*snapshotmanager_agent_package*> is the name of the agent rpm package, without the version number and the file extension (`.rpm`).

   For example, if the name of the agent rpm package is `VRTSflexsnap-agent-10.1.0.0.1005-RHEL.x86_64.rpm`, the command syntax is as follows:

   ```
   # sudo yum -y remove VRTSflexsnap-agent
   ```

3. For Windows-based agent, do the following:

   From Windows Control Panel > Programs and Features, select the entry for the Snapshot Manager agent (**Veritas Snapshot Manager Agent**) and then click **Uninstall**.

   Follow the wizard workflow to uninstall the agent from the Windows instance.

   Note: To allow the uninstallation, admin users will have to click Yes on the Windows UAC prompt. Non-admin users will have to specify admin user credentials on the UAC prompt.

4. This completes the agent uninstallation.

   You can now proceed to uninstall Snapshot Manager.

   See “Removing Snapshot Manager from a standalone Docker host environment” on page 215.

Removing Snapshot Manager from a standalone Docker host environment

The process for uninstalling Snapshot Manager is the same as that followed for installation. The only difference is that you specify `"uninstall"` in the command, which tells the installer to remove the components from the host.

During uninstallation, the installer performs the following tasks on the Snapshot Manager host:

- Stops all the Snapshot Manager containers that are running
- Removes the Snapshot Manager containers
- Unloads and removes the Snapshot Manager images

To uninstall Snapshot Manager

1. Ensure that you have uninstalled the Snapshot Manager agents from all the hosts that are part of the Snapshot Manager configuration.

   See “Removing the Snapshot Manager agents” on page 214.

2. Verify that there are no protection policy snapshots or other operations in progress, and then uninstall Snapshot Manager by running the following command on the host:

   (*For Docker*)

   ```
   # sudo docker run -it --rm -u 0
   -v /full_path_to_volume:/full_path_to_volume
   -v /var/run/docker.sock:/var/run/docker.sock
   veritas/flexsnap-deploy:<version> uninstall
   ```

   (*For Podman*)

   ```
   # sudo podman run -it --rm -u 0
   -v /full_path_to_volume:/full_path_to_volume
   -v /var/run/podman.sock:/var/run/podman.sock
   veritas/flexsnap-deploy:<version> uninstall
   ```

   Replace the following parameters as per your environment:

Parameter	Description
<version>	Represents the Snapshot Manager product version that is installed on the host.
<full_path_to_volume>	Represents the path to the Snapshot Manager data volume, which typically is `/cloudpoint`.

   For example, if the product version is `10.1.0.0.1005`, the command syntax for docker is as follows:

```
# sudo docker run -it --rm -u 0 -v /cloudpoint:/cloudpoint -v
/var/run/docker.sock:/var/run/docker.sock
veritas/flexsnap-deploy:10.1.0.0.1005 uninstall
```

If using a proxy server, then using the examples provided in the table earlier, the command syntax for docker is as follows:

```
# sudo docker run -it --rm -u 0 -v /cloudpoint:/cloudpoint -e
VX_HTTP_PROXY="http://proxy.mycompany.com:8080/" -e
VX_HTTPS_PROXY="http://proxy.mycompany.com:8080/" -e
VX_NO_PROXY="localhost,mycompany.com,192.168.0.10:80" -v
/var/run/docker.sock:/var/run/docker.sock
veritas/flexsnap-deploy:10.1.0.0.1005 uninstall
```

Note: This is a single command. Ensure that you enter the command without any line breaks.

The installer begins to unload the relevant Snapshot Manager container packages from the host. Messages similar to the following indicate the progress status:

```
Uninstalling NetBackup Snapshot Manager
------------------------------------------
Stopping flexsnap-mongodb ... done
Stopping flexsnap-rabbitmq ... done
Stopping flexsnap-auth ... done
Stopping flexsnap-core ... done
Removing flexsnap-mongodb ... done
Removing flexsnap-rabbitmq ... done
Removing flexsnap-auth ... done
Removing flexsnap-core ... done
Unloading flexsnap-mongodb ... done
Unloading flexsnap-rabbitmq ... done
Unloading flexsnap-auth ... done
Unloading flexsnap-core ... done
```

3. Confirm that the Snapshot Manager containers are removed.

   Use the following docker command:

   (*For Docker*) `# sudo docker ps -a`

   (*For Podman*) `# sudo podman ps -a`

4. If desired, remove the Snapshot Manager container images from the host.

Use the following docker command to view the docker images that are loaded on the host:

- (*For Docker*) `# sudo docker images -a`
- (*For Podman*) `# sudo podman images -a`

Use the following respective commands to remove the Snapshot Manager container images from the host:

- (*For Docker*) `# sudo docker rmi` *<image ID>*
- (*For Podman*) `# sudo podman rmi` *<image ID>*

5. This completes the Snapshot Manager uninstallation on the host.

   Possible next step is to re-deploy Snapshot Manager.

   See “Installing Snapshot Manager in the Docker/Podman environment” on page 38.

Removing Snapshot Manager extensions - VM-based or managed Kubernetes cluster-based

During uninstallation, the installer performs the following tasks on the Snapshot Manager extension host:

- Stops all the Snapshot Manager containers that are running
- Removes the Snapshot Manager containers

To uninstall a VM-based extension

1 **For Docker environment:**

Run the following command:

```
# sudo docker run -it --rm -u 0
-v /<full_path_to_volume_name>:/<full_path_to_volume_name>
-v /var/run/docker.sock:/var/run/docker.sock
veritas/flexsnap-deploy:<version> uninstall
```

Example:

```
# sudo docker run -it --rm -u 0
-v /cloudpoint:/cloudpoint
-v /var/run/docker.sock:/var/run/docker.sock
veritas/flexsnap-deploy:<10.1.x.x.xxx> uninstall
```

Note: This is a single command without any line breaks.

For Podman environment:

Run the following command:

```
# podman run -it --rm -u 0
-v /<full_path_to_volume_name>:/<full_path_to_volume_name>
-v /run/podman/podman.sock:/run/podman/podman.sock
veritas/flexsnap-deploy:<version> uninstall
```

Example:

```
# podman run -it --rm -u 0
-v /cloudpoint:/cloudpoint
-v /run/podman/podman.sock:/run/podman/podman.sock
veritas/flexsnap-deploy:<10.1.x.x.xxx> uninstall
```

Note: This is a single command without any line breaks.

Replace the following parameters as per your environment:

Parameter	Description
<version>	Represents the Snapshot Manager product version that is installed on the host.
<full_path_to_volume>	Represents the path to the Snapshot Manager data volume, which typically is `/cloudpoint`.

2 If desired, remove the Snapshot Manager container images from the extension host.

Use the following docker command to view the docker images that are loaded on the host and remove the Snapshot Manager images based on their IDs.

```
# sudo docker images -a
```

```
# sudo docker rmi <image ID>
```

This completes the Snapshot Manager extension uninstallation on a VM host.

To uninstall a managed Kubernetes cluster-based extension

- Execute the extension script `cp_extension.sh` that was downloaded at the time of extension installation, from the host where `kubectl` is installed.

  Run the following command:

  ```
  bash cp_extension.sh uninstall
  ```

  Once the uninstallation is triggered, provide the namespace as an input, from which the extension services need to be uninstalled.

After the uninstallation, the provisioned cloud resources associated with the uninstalled extension can be terminated or removed.

Restoring Snapshot Manager

You can restore Snapshot Manager using any of the following methods:

- Recover Snapshot Manager using a snapshot you have in the cloud
- Recover Snapshot Manager using a backup located on-premises
- (*Only for GCP cloud provider*) Recover Snapshot Manager using GCP cross-project restore

Using Snapshot Manager snapshot located in the cloud

To recover Snapshot Manager using a snapshot you have in the cloud

1 Using your cloud provider's dashboard or console, create a volume from the existing snapshot.

2 Create a new virtual machine with specifics equal to or better than your previous Snapshot Manager server.

3 Install Docker/Podman on the new server.

See “Installing container platform (Docker, Podman)” on page 30.

4 Attach the newly-created volume to this Snapshot Manager server instance.

5 Create the Snapshot Manager installation directory on this server.

Use the following command:

```
# mkdir /full_path_to_cloudpoint_installation_directory
```

For example:

```
# mkdir /cloudpoint
```

6 Mount the attached volume to the installation directory you just created.

Use the following command:

```
# mount /dev/device-name
/full_path_to_cloudpoint_installation_directory
```

For example:

```
# mount /dev/xvdb /cloudpoint
```

7 Verify that all Snapshot Manager related configuration data and files are in the directory.

Enter the following command:

```
# ls -l /cloudpoint
```

8 Download or copy the Snapshot Manager installer binary to the new server.

9 Install Snapshot Manager.

Use the following command:

(*For Docker*)

```
# sudo docker run -it --rm -u 0
-v /cloudpoint:/cloudpoint
-v /var/run/docker.sock:/var/run/docker.sock
veritas/flexsnap-deploy:10.1.0.0.1005 install
```

(*For Podman*)

```
# sudo podman run -it --rm -u 0
-v /cloudpoint:/cloudpoint
-v /var/run/podman.sock:/var/run/podman.sock
veritas/flexsnap-deploy:10.1.0.0.1005 install
```

Here, `10.1.0.0.1005` represents the Snapshot Manager version. Replace it as per your currently installed product version.

Note: This is a single command. Ensure that you enter the command without any line breaks.

The installation program detects an existing version of Snapshot Manager and re-installs all Snapshot Manager services without overwriting existing content.

Messages similar to the following are displayed on the command prompt:

```
Configuration started at time Wed May 13 22:20:47 UTC 2020
This is a re-install.
Checking if a 1.0 release container exists ...
```

Note the line that indicates that the operation is a re-install.

10 When the installation completes, you can resume working with Snapshot Manager using your existing credentials.

Using Snapshot Manager backup located on-premise

To recover Snapshot Manager using a backup located on-premise

1. Copy the existing Snapshot Manager backup to the new Snapshot Manager server and extract it to the Snapshot Manager installation directory.

   In the following example, because `/cloudpoint` was backed up, the command creates a new `/cloudpoint` directory.

   ```
   # tar -zxf cloudpoint_dr.tar.gz -C /cloudpoint/
   ```

2. Download or copy the Snapshot Manager installer binary to the new server.

3 Install Snapshot Manager.

Use the following command:

(*For Docker*)

```
# sudo docker run -it --rm -u 0
-v /cloudpoint:/cloudpoint
-v /var/run/docker.sock:/var/run/docker.sock
veritas/flexsnap-deploy:10.1.0.0.1005 install
```

(*For Podman*)

```
# sudo podman run -it --rm -u 0
-v /cloudpoint:/cloudpoint
-v /var/run/podman.sock:/var/run/podman.sock
veritas/flexsnap-deploy:10.1.0.0.1005 install
```

Here, `10.1.0.0.1005` represents the Snapshot Manager version. Replace it as per your currently installed product version.

Note: This is a single command. Ensure that you enter the command without any line breaks.

The installation program detects an existing version of Snapshot Manager and re-installs all Snapshot Manager services without overwriting existing content.

Messages similar to the following are displayed on the command prompt:

```
Configuration started at time Wed May 13 22:20:47 UTC 2020
This is a re-install.
Checking if a 1.0 release container exists ...
```

Note the line that indicates that the operation is a re-install.

4 When the installation completes, you can resume working with Snapshot Manager using your existing credentials.

Using Snapshot Manager GCP cross-project restore

Note: The zone of Snapshot Manager and node-pools of the extension must be same.

In case of cross-project restore, a provider must be configured for the region in which Snapshot Manager is installed. If Snapshot Manager is installed in **us-west1-b** zone then a provider for **us-west1** region must be configured.

Let the details of the first project in which Snapshot Manager is installed be:

- Service-account = **cp-host-service-account**
- Project-name = **cp-host-project**

Let the details of the second project be:

- Service-account = **other-service-account**
- Project-name = **other-project**

To recover Snapshot Manager using GCP cross-project restore

1. Create a cross project role in **other-service-account** with the following permissions:
   - `compute.snapshots.useReadOnly`
   - `compute.disks.create`
   - `Cloud KMS CryptoKey Encrypter/Decrypter`
2. Assign the above role to **cp-host-service-account** under the **other-project** project.

Chapter 13

Troubleshooting NetBackup Snapshot Manager

This chapter includes the following topics:

- Troubleshooting Snapshot Manager
- SQL snapshot or restore and granular restore operations fail if the Windows instance loses connectivity with the Snapshot Manager host
- Disk-level snapshot restore fails if the original disk is detached from the instance
- Discovery is not working even after assigning system managed identity to the control node pool
- Performance issue with GCP backup from snapshot
- Post migration on host agents fail with an error message
- File restore job fails with an error message

Troubleshooting Snapshot Manager

Refer to the following troubleshooting scenarios:

- **Snapshot Manager agent fails to connect to the Snapshot Manager server if the agent host is restarted abruptly.**
  This issue may occur if the host where the Snapshot Manager agent is installed is shut down abruptly. Even after the host restarts successfully, the agent fails to establish a connection with the Snapshot Manager server and goes into an offline state.

The agent log file contains the following error:

```
Flexsnap-agent-onhost[4972] mainthread
flexsnap.connectors.rabbitmq: error - channel 1 closed
unexpectedly: (405) resource_locked - cannot obtain exclusive
access to locked queue '
flexsnap-agent.a1f2ac945cd844e393c9876f347bd817' in vhost '/'
```

This issue occurs because the RabbitMQ connection between the agent and the Snapshot Manager server does not close even in case of an abrupt shutdown of the agent host. The Snapshot Manager server cannot detect the unavailability of the agent until the agent host misses the heartbeat poll. The RabbitMQ connection remains open until the next heartbeat cycle. If the agent host reboots before the next heartbeat poll is triggered, the agent tries to establish a new connection with the Snapshot Manager server. However, as the earlier RabbitMQ connection already exists, the new connection attempt fails with a resource locked error.

As a result of this connection failure, the agent goes offline and leads to a failure of all snapshot and restore operations performed on the host.

Workaround:

Restart the Veritas Snapshot Manager Agent service on the agent host.

- On a Linux hosts, run the following command:

  ```
  # sudo systemctl restart flexsnap-agent.service
  ```

- On Windows hosts:

  Restart the `Veritas Snapshot Manager™ Agent` service from the Windows Services console.

- **Snapshot Manager agent registration on Windows hosts may time out or fail.**

  For protecting applications on Windows, you need to install and then register the Snapshot Manager agent on the Windows host. The agent registration may sometimes take longer than usual and may either time out or fail.

  Workaround:

  To resolve this issue, try the following steps:

  - Re-register the agent on the Windows host using a fresh token.
  - If the registration process fails again, restart the Snapshot Manager services on the Snapshot Manager server and then try registering the agent again.

  Refer to the following for more information:

  See “Registering the Windows-based agent” on page 135.
  See “Restarting Snapshot Manager” on page 48.

- **Disaster recovery when DR package is lost or passphrase is lost.**

  This issue may occur if the DR package is lost or the passphrase is lost.

In case of Catalog backup, 2 backup packages are created:

- DR package which contains all the certs
- Catalog package which contains the data base

The DR package contains the NetBackup UUID certs and Catalog DB also has the UUID. When you perform disaster recovery using the DR package followed by catalog recovery, both the UUID cert and the UUID are restored. This allows NetBackup to communicate with Snapshot Manager since the UUID is not changed.

However if the DR package is lost or the Passphrase is lost the DR operation cannot be completed. You can only recover the catalog without DR package after you reinstall NetBackup. In this case, a new UUID is created for NetBackup which is not recognised by Snapshot Manager. The one-to-one mapping of NetBackup and Snapshot Manager is lost.

Workaround:

To resolve this issue, you must update the new NBU UUID and Version Number after NetBackup primary is created.

- The NetBackup administrator must be logged on to the NetBackup Web Management Service to perform this task. Use the following command to log on:
  `/usr/openv/netbackup/bin/bpnbat -login -loginType WEB`
- Execute the following command on the primary server to get the NBU UUID:
  `/usr/openv/netbackup/bin/admincmd/nbhostmgmt -list -host <primary server host name> | grep "Host ID"`
- Execute the following command to get the Version Number:
  `/usr/openv/netbackup/bin/admincmd/bpgetconfig -g <primary Ssrver host name> -L`

After you get the NBU UUID and Version number, execute the following command on the Snapshot Manager host to update the mapping:

`/cloudpoint/scripts/cp_update_nbuuid.sh -i <NBU UUID> -v <Version Number>`

- **The snapshot job is successful but backup job fails with error "The Snapshot Managers certificate is not valid or doesn't exist.(9866)" when ECA_CRL_CHECK disabled on master server.**

  If ECA_CRL_CHECK is configured on master server and is disabled then it must be configured in `bp.conf` on Snapshot Manager setup with same value.

  For example, considering a scenario of backup from snapshot where NetBackup is configured with external certificate and certificate is revoked. In this case, if ECA_CRL_CHECK is set as DISABLE on master then set the same value in

`bp.conf` of Snapshot Manager setup, otherwise snapshot operation will be successful and backup operation will fail with the certificate error.
See “Configuring security for Azure Stack ” on page 165.

- **Snapshot Manager fails to establish connection using agentless to the Windows cloud instance**

  Error 1: `<Instance_name>: network connection timed out.`

  Case 1: Snapshot Manager server log message:

  ```
  WARNING - Cannot connect to the remote host. SMB Connection timeout
   <IP address> <user>

  …

  flexsnap.OperationFailed: Could not connect to the remote server
  <IP address>
  ```

  Workaround:

  To resolve this issue, try the following steps:

  - Verify if the SMB port 445 is added in the Network security group and is accessible from the Snapshot Manager.
  - Verify if the SMB port 445 is allowed through cloud instance firewall.

  Case 2: Snapshot Manager log message:

  ```
  WARNING - Cannot connect to the remote host. WMI Connection
  timeout <IP address> <user>

  …

  flexsnap.OperationFailed: Could not connect to the remote
  server <IP address>
  ```

  Workaround:

  To resolve this issue, try the following steps:

  - Verify if the DCOM port (135) is added in the Network security group and is accessible from Snapshot Manager.
  - Verify if the port 135 is allowed through cloud instance firewall.

  Case 3: Snapshot Manager log message:

  ```
  Exception while opening SMB connection, [Errno Connection error
  (<IP address>:445)] [Errno 113] No route to host.
  ```

Workaround:: Verify if the cloud instance is up and running or not in inconsistent state.

Case 4: Snapshot Manager log message:

```
Error when closing dcom connection: 'Thread-xxxx'"
```

Where, *xxxx* is the thread number.

Workaround::

To resolve this issue, try the following steps:

- Verify if the WMI-IN dynamic port range or the fixed port as configured is added in the Network security group.
- Verify and enable WMI-IN port from the cloud instance firewall.

Error 2: `<Instance_name>: Could not connect to the virtual machine.`

Snapshot Manager log message:

```
Error: Cannot connect to the remote host. <IP address> Access denied.
```

Workaround::

To resolve this issue, try the following steps:

- Verify if the user is having administrative rights.
- Verify if the UAC is disabled for the user.

- **Snapshot Manager cloud operations fail on a RHEL system if a firewall is disabled**

  The Snapshot Manager operations fail for all the supported cloud plugins on a RHEL system, if a firewall is disabled on that system when the Snapshot Manager services are running. This is a network configuration issue that prevents the Snapshot Manager from accessing the cloud provider REST API endpoints.

  Workaround:

  - Stop Snapshot Manager

    ```
    # docker run --rm -it
    -v /var/run/docker.sock:/var/run/docker.sock
    -v /cloudpoint:/cloudpoint veritas/flexsnap-deploy:<version>
    stop
    ```

  - Restart Docker

    ```
    # systemctl restart docker
    ```

  - Restart Snapshot Manager

    ```
    # docker run --rm -it
    -v /var/run/docker.sock:/var/run/docker.sock
    ```

```
-v /cloudpoint:/cloudpoint veritas/flexsnap-deploy:<version>
start
```

- **Backup from Snapshot job and Indexing job fails with the errors**

```
Jun 10, 2021 2:17:48 PM - Error mqclient (pid=1054) SSL
Connection failed with string, broker:<hostname>
Jun 10, 2021 2:17:48 PM - Error mqclient (pid=1054) Failed SSL
handshake, broker:<hostname>
Jun 10, 2021 2:19:16 PM - Error nbcs (pid=29079) Invalid
operation for asset: <asset_id>
Jun 10, 2021 2:19:16 PM - Error nbcs (pid=29079) Acknowledgement
not received for datamover <datamover_id>
```

and/or

```
Jun 10, 2021 3:06:13 PM - Critical bpbrm (pid=32373) from client
<asset_id>: FTL - Cannot retrieve the exported snapshot details
for the disk with UUID:<disk_asset_id>
Jun 10, 2021 3:06:13 PM - Info bptm (pid=32582) waited for full
buffer 1 times, delayed 220 times
Jun 10, 2021 3:06:13 PM - Critical bpbrm (pid=32373) from client
<asset_id>: FTL - cleanup() failed, status 6
```

This can happen when the inbound access to Snapshot Manager on port 5671 and 443 port gets blocked at the OS firewall level (firewalld). Hence, from the datamover container (used for the Backup from Snapshot and Indexing jobs), communication to Snapshot Manager gets blocked. This results in the datamover container not being able to start the backup or indexing.

Workaround:

Modify the rules in OS firewall to allow the inbound connection from 5671 and 443 port.

- **Agentless connection fails for a VM with an error message.**

  Agentless connection fails for a VM with the following error message when user changes the authentication type from SSH Key based to password based for a VM through the portal:

```
User does not have the required privileges to establish an agentless connection
```

This issue occurs when the permissions are not defined correctly for the user in the sudoers file as mentioned in the above error message.

Workaround:

Resolve the sudoers file issue for the user by providing the required permissions to perform the passwordless sudo operations.

- **When Snapshot Manager is deployed in private subnet (without internet) Snapshot Manager function fails**

  This issue occurs when Snapshot Manager is deployed in private network where firewall is enabled or public IP which is disabled. The customer's information security team would not allow full internet access to the virtual machine's.

  Workaround:

  Enable the ports from the firewall command line using the following commands:

  ```
  firewall-cmd --add-port=22/tcp
  firewall-cmd --add-port=5671/tcp
  firewall-cmd --add-port=443/tcp
  ```

- **Restoring asset from backup copy fails**

  In some of the scenarios it is observed that the connection resets intermittently in Docker container. Due to this the server sends more tcp payload than the advertised client window. Sometimes Docker container drops **SYN+ACK** packet from new TCP connection handshake. To allow these packets, use the `nf_conntrack_tcp_be_liberal` option.

  If `nf_conntrack_tcp_be_liberal = 1` then the following packets are allowed:

  - ACK is under the lower bound (possible overly delayed ACK)
  - ACK is over the upper bound (ACKed data not seen yet)
  - SEQ is under the lower bound (already ACKed data retransmitted)
  - SEQ is over the upper bound (over the window of the receiver)

  If `nf_conntrack_tcp_be_liberal = 0` then those are also rejected as invalid.

  Workaround:

  To resolve the issue of restore from backup copy, use the `nf_conntrack_tcp_be_liberal = 1` option and set this value on node where datamover container is running.

  Use the following command for setting the value of `nf_conntrack_tcp_be_liberal`:

  ```
  sysctl -w net.netfilter.nf_conntrack_tcp_be_liberal=1
  ```

- **Some pods on Kubernetes extension progressed to completed state**

  Workaround:

  Disable Kubernetes extension.

  Delete listener pod using the following command:

  ```
  #kubectl delete pod flexnsap-listener-xxxxx -n <namespace>
  ```

  Enable Kubernetes extension.

- **User is not able to customize a cloud protection plan**

  Workaround:

Create a new protection plan with the desired configuration and assign it to the asset.

- **Podman container not starting or containers are not up after reboot**
  On RHEL 8.x platform, restarting container or machine reboot, the container displays the following error message:

```
# podman restart flexsnap-coordinator 47ca97002e53de808cb8d0526ae033d4b317d538
 "2022-02-05T04:53:42.265084989+00:00 Feb 05 04:53:42 flexsnap-coordinator fle
agent_container_health_check flexsnap.container_manager: INFO - Response: b'{"
""error creating container storage: the container name \\""flexsnap-agent.15bd
\\""30f031d586b1ab524511601aad521014380752fb127a9440de86a81b327b6777\\"". You
 name.: that name is already in use"","response"":500}\n'"
```

  Workaround:
  Check if there is a file with IP address entry mapping to the container that could not be started at `/var/lib/cni/networks/flexsnap-network/` file system location.

```
[ec2-user@ip-172-31-44-163 ~]$ ls -latr
/var/lib/cni/networks/flexsnap-network/ total 16 -rwxr-x---. 1
root root 0 Jan 22 12:30 lock drwxr-xr-x. 4 root root 44 Jan 22
12:30 .. -rw-r--r--. 1 root root 70 Feb 4 14:47 10.89.0.150
-rw-r--r--. 1 root root 70 Feb 4 14:47 10.89.0.151 -rw-r--r--. 1
root root 70 Feb 4 14:47 10.89.0.152 -rw-r--r--. 1 root root 11
Feb 7 11:09 last_reserved_ip.0 drwxr-xr-x. 2 root root 101 Feb 7
11:13 . [ec2-user@ip-172-31-44-163 ~]$
```

  From the above directory , delete the duplicate IP address file and perform the stop and start operation as follows:
  Stop the container: `#podman stop <container_name>`
  Start the container:`#podman start <container_name>`

- **After starting the start/stop services, Snapshot Manager, RabbitMQ and MongoDB containers are still in the starting state**
  It was observed that flexsnap-mongodb and flexsnap-rabbitmq containers did not go into healthy state. Following is the Below is the state of flexsnap-mongodb container:

```
[ec2-user@ip-172-31-23-60 log]$ sudo podman container inspect --format='{{json
flexsnap-mongodb {"Test":["CMD-SHELL","echo 'db.runCommand({ping: 1}).ok'
| mongo --ssl --sslCAFile /cloudpoint/keys/cacert.pem
--sslPEMKeyFile /cloudpoint/keys/mongodb.pem flexsnap-mongodb:27017/zenbrain -
"Interval":60,"Timeout":30000000000,"Retries":3} [ec2-user@ip-172-31-23-60 log
{{json .State.Healthcheck}}' flexsnap-mongodb {"Status":"starting","FailingStr
```

Workaround:

Run the following #podman CLI(s) command:

```
[ec2-user@ip-172-31-23-60 log]$ sudo podman healthcheck run flexsnap-mongo

[ec2-user@ip-172-31-23-60 log]$ sudo podman ps -a

CONTAINER ID  IMAGE                            COMMAND  CREATED      STATUS

fe8cf001032b  localhost/veritas/

  flexsnap-fluentd:10.0.0.0.9817         2 days ago    Up 45 hours ago

2c00500c1ac6  localhost/veritas/

  flexsnap-mongodb:10.0.0.0.9817         2 days ago    Up 45 hours ago (he

7ab3e248024a  localhost/veritas/

  flexsnap-rabbitmq:10.0.0.0.9817        2 days ago    Up 45 hours ago (st

[ec2-user@ip-172-31-23-60 log]$ sudo podman healthcheck run flexsnap-rabbi

[ec2-user@ip-172-31-23-60 log]$ sudo podman ps -a

CONTAINER ID  IMAGE                              COMMAND  CREATED       STATU

fe8cf001032b  localhost/veritas/

  flexsnap-fluentd:10.0.0.0.9817        2 days ago    Up 45 hours ago

2c00500c1ac6  localhost/veritas/

  flexsnap-mongodb:10.0.0.0.9817        2 days ago    Up 45 hours ago (hea

7ab3e248024a  localhost/veritas/

  flexsnap-rabbitmq:10.0.0.0.9817       2 days ago    Up 45 hours ago (hea

 [ec2-user@ip-172-31-23-60 log]$ sudo podman container inspect --format='{

{"Status":"healthy","FailingStreak":0,"Log":

[{"Start":"2022-02-14T07:32:13.051150432Z","End":"2022-02-14T07:32:13.4446

[ec2-user@ip-172-31-23-60 log]$ sudo podman container inspect --format='{{
```

```
{"Status":"healthy","FailingStreak":0,"Log":

[{"Start":"2022-02-14T07:32:46.537804403Z","End":"2022-02-14T07:32:47.29369574

[ec2-user@ip-172-31-23-60 log]$
```

- **Certificate generation would fail while registering Snapshot Manager with NetBackup**

  Starting Snapshot Manager release 9.1.2, NetBackup certificate generation will happen synchronously with registration in register API of Snapshot Manager. Hence, any failure in certificate generation will cause failure while registering Snapshot Manager with NetBackup, that is ading or editing the Snapshot Manager entry from Web UI. These certificates are used for datamover which is launched for operations like backup from snapshot, restore from backup, indexing (VxMS based), and so on. Hence, if certificate generation fails, these jobs cannot be performed. Hence Snapshot Manager on cloud VMs cannot connect to NetBackup on lab VMs, hence the registration will fail, and hence Snapshot Manager cannot be added to NetBackup.

  Workaround:

  To add Snapshot Manager in such scenario requires to skip certificate generation on Snapshot Manager by adding the following entry in `/cloudpoint/flexsnap.conf` file:

  ```
  [client_registration] skip_certificate_generation = yes
  ```

- **Default timeout of 6 hours is not allowing restore of larger database (size more than 300 GB)**

  Workaround:

  Configurable timeout parameter value can be set to restore larger database. The timeout value can be specified in `/etc/flexsnap.conf` file of `flexsnap-coordinator` container. It does not require restart of the coordinator container. Timeout value would be picked up in next database restore job.

  User must specify the timeout value in seconds as follows:

  ```
  docker exec -it flexsnap-coordinator bash
  root@flexsnap-coordinator:/# cat /etc/flexsnap.conf [global] target
  = flexsnap-rabbitmq grt_timeout = 39600
  ```

- **Agentless connection and granular restore to restored host fails when the VM restored from backup has 50 tags attached to it**

  Workaround:

  (For AWS) If a Windows VM restored from backup has 50 tags and platform tag does not exists, user can remove any tag that is not required and add the **Platform: windows** tag.

- **For few versions of GKE versions, failed pod issues are observed in namespace**

  Following few failed pods in namespace is observed with failure status as `NodeAffinity`:

  ```
  $ kubectl get pods -n <cp_extension_namespace>

  NAME                                          READY     STATUS       RESTART
  flexsnap-datamover-
  2fc2967943ba4ded8ef653318107f49c-664tm        0/1     Terminating    0
  flexsnap-fluentd-collector-c88f8449c-5jkqh    0/1     NodeAffinity   0
  flexsnap-fluentd-collector-c88f8449c-ph8mx    0/1     NodeAffinity   0
  flexsnap-fluentd-collector-c88f8449c-rqw7w    1/1     Running        0
  flexsnap-fluentd-collector-c88f8449c-sswzr    0/1     NodeAffinity   0
  flexsnap-fluentd-ftlnv                        1/1     Running        3 (10
  flexsnap-listener-84c66dd4b8-6l4zj            1/1     Running        0
  flexsnap-listener-84c66dd4b8-ls4nb            0/1     NodeAffinity   0
  flexsnap-listener-84c66dd4b8-x84q8            0/1     NodeAffinity   0
  flexsnap-listener-84c66dd4b8-z7d5m            0/1     NodeAffinity   0
  flexsnap-operator-6b7dd6c56c-cf4pc            1/1     Running        0
  flexsnap-operator-6b7dd6c56c-qjsbs            0/1     NodeAffinity   0
  flexsnap-operator-6b7dd6c56c-xcsgj            0/1     NodeAffinity   0
  flexsnap-operator-6b7dd6c56c-z86tc            0/1     NodeAffinity   0
  ```

  However, these failures do not affect the functionality of Snapshot Manager Kubernetes extension.

  Workaround:

  Manually clean-up the failed pods using the following command:

  ```
  kubectl get pods -n <cp_extension_namespace> | grep NodeAffinity
  | awk '{print $1}' | xargs kubectl delete pod -n
  <cp_extension_namespace>
  ```

- **Plugin information is duplicated, if Snapshot Manager registration has failed in previous attempts**

  This occurs only when Snapshot Manager has been deployed using the MarketPlace Deployment Mechanism. This issue is observed when the plugin information is added before the registration. This issue creates duplicate plugin information in the **CloudPoint_plugin.conf** file.

  Workaround:

  Manually delete the duplicated plugin information from the **CloudPoint_plugin.conf** file.

  For example, consider the following example where the duplicate entry for GCP plugin config is visible (in bold) in **CloudPoint_plugin.conf** file:

```
{
    "CPServer1": [
      {
        "Plugin_ID": "test",
        "Plugin_Type": "aws",
        "Config_ID": "aws.8dda1bf5-5ead-4d05-912a-71bdc13f55c4",
        "Plugin_Category": "Cloud",
        "Disabled": false
      }
    ]
  },
  {
    "CPServer2": [
      {
        "Plugin_ID": "gcp.2080179d-c149-498a-bf1f-4c9d9a76d4dd",
        "Plugin_Type": "gcp",
        "Config_ID": "gcp.2080179d-c149-498a-bf1f-4c9d9a76d4dd",
        "Plugin_Category": "Cloud",
        "Disabled": false
      },
      {
        "Plugin_ID": "gcp.2080179d-c149-498a-bf1f-4c9d9a76d4dd",
        "Plugin_Type": "gcp",
        "Config_ID": "gcp.2080179d-c149-498a-bf1f-4c9d9a76d4dd",
        "Plugin_Category": "Cloud",
        "Disabled": false
      }
    ]
  }
```

- **Plugin information is duplicated, if cloned Snapshot Manager is added into NetBackup**

  This occurs only when cloned Snapshot Manager is added into NetBackup during migration of Snapshot Manager to RHEL 8.6 VM. Cloning of Snapshot Manager uses existing Snapshot Manager volume to create new Snapshot Manager. This creates duplicate entry into **CloudPoint_plugin.conf** file.

  Workaround:

  Manually edit and delete the duplicated plugin information from the **CloudPoint_plugin.conf** file.

  For example, consider the following example where the duplicate entry for Azure plugin config is visible (in bold) in **CloudPoint_plugin.conf** file:

```
{
    "CPServer1": [
      {
        "Plugin_ID": "config10",
        "Plugin_Type": "azure",
        "Config_ID": "azure.327ec7fc-7a2d-4e94-90a4-02769a2ba521",
        "Plugin_Category": "Cloud",
        "Disabled": false
      }
    ]
  },
  {
    "CPServer2": [
      {
        "Plugin_ID": "azure.327ec7fc-7a2d-4e94-90a4-02769a2ba521",
        "Plugin_Type": "azure",
        "Config_ID": "azure.327ec7fc-7a2d-4e94-90a4-02769a2ba521",
        "Plugin_Category": "Cloud",
        "Disabled": false
      },
       {
    "cpserver101.yogesh.joshi2-dns-zone": [
      {
        "Plugin_ID": "azure.327ec7fc-7a2d-4e94-90a4-02769a2ba521",
        "Plugin_Type": "azure",
        "Config_ID": "azure.327ec7fc-7a2d-4e94-90a4-02769a2ba521",
        "Plugin_Category": "Cloud",
        "Disabled": false
      },
      {
        "Plugin_ID": "AZURE_PLUGIN",
        "Plugin_Type": "azure",
        "Config_ID": "azure.4400a00a-8d2b-4985-854a-74f48cd4567e",
        "Plugin_Category": "Cloud",
        "Disabled": false
      }
    ]
  }
 ]
}
```

SQL snapshot or restore and granular restore operations fail if the Windows instance loses connectivity with the Snapshot Manager host

This issue occurs if the Snapshot Manager agent that is configured on a Windows instance loses network connectivity with the Snapshot Manager host. Snapshot Manager operations such as snapshot creation or restore for SQL Server and granular restore begin to fail for the Windows instance.

The connectivity failure may occur due to various reasons such as a services restart on the Snapshot Manager host as part of a Snapshot Manager software upgrade or a general network disruption.

The flexsnap-agent logs may contain messages similar to the following:

```
flexsnap-agent-onhost[2720] MainThread flexsnap.connectors.rabbitmq:
ERROR - Unexpected exception() in main loop
flexsnap-agent-onhost[2720] MainThread agent: ERROR - Agent failed
unexpectedly
```

If Snapshot Manager is deployed in a Veritas NetBackup environment, the NetBackup logs may contain messages similar to the following:

```
Error nbcs (pid=5997) Failed to create snapshot for asset: <sqlassetname>
Error nbcs (pid=5997) Operation failed. Agent is unavailable.
```

Workaround:

To resolve this issue, restart the `Veritas Snapshot Manager Agent` service on the Windows instance.

Disk-level snapshot restore fails if the original disk is detached from the instance

This issue occurs if you are performing a disk-level snapshot restore to the same location.

When you trigger a disk-level snapshot restore to the same location, NetBackup first detaches the existing original disk from the instance, creates a new volume from the disk snapshot, and then attaches the new volume to the instance. The original disk is automatically deleted after the restore operation is successful.

However, if the original disk whose snapshot is being restored is manually detached from the instance before the restore is triggered, the restore operation fails.

You may see the following message on the NetBackup UI:

```
Request failed unexpectedly: [Errno 17] File exists: '/<app.diskmount>'
```

The NetBackup coordinator logs contain messages similar to the following:

```
flexsnap.coordinator: INFO - configid : <app.snapshotID> status changed to
 {u'status': u'failed', u'discovered_time': <time>, u'errmsg': u'
Could not connect to <application> server localhost:27017:
[Errno 111]Connection refused'}
```

Workaround:

If the restore has already failed in the environment, you may have to manually perform a disk cleanup first and then trigger the restore job again.

Perform the following steps:

1. Log on to the instance for which the restore operation has failed.

   Ensure that the user account that you use to connect has administrative privileges on the instance.

2. Run the following command to unmount the application disk cleanly:

   ```
   # sudo umount /<application_diskmount>
   ```

   Here, <*application_diskmount*> is the original application disk mount path on the instance.

   If you see a `"device is busy"` message, wait for some time and then try the `umount` command again.

3. From the NetBackup UI, trigger the disk-level restore operation again.

In general, if you want to detach the original application disks from the instance, use the following process for restore:

1. First take a disk-level snapshot of the instance.
2. After the snapshot is created successfully, manually detach the disk from the instance.

   For example, if the instance is in the AWS cloud, use the AWS Management Console and edit the instance to detach the data disk. Ensure that you save the changes to the instance.

3. Log on to the instance using an administrative user account and then run the following command:

   ```
   # sudo umount /<application_diskmount>
   ```

If you see a `"device is busy"` message, wait for some time and then try the `umount` command again.

4. Now trigger a disk-level restore operation from the NetBackup UI.

Discovery is not working even after assigning system managed identity to the control node pool

If **System managed identity** is not enabled on Snapshot Manager (deployed on Kubernetes cluster) and user adds Azure cloud provider (with **User managed identity** already added) using **System managed identity**, then **User managed identity** is automatically selected for the addition of Azure cloud provider and plugin addition is successful.

But it could not discover the assets if there are insufficient permissions added in **System managed identity**. Discovery and Snapshot Manager related operations would not work even if **System managed identity** is enabled and required permission/role is added to **System managed identity** later on. Because it will always use **User managed identity** at the backend of Snapshot Manager.

To resolve this issue, perform the following steps

1 Update the required permission/role and then add the permissions to **User managed identity** and run the required operations again.

2 Edit the corresponding Azure provider configuration in NetBackup Web UI and run the required operations again.

The following table lists the scenarios and expected outcomes of different Azure plug-in configurations:

Table 13-1 Scenarios and expected outcomes of different Azure plug-in configurations

Snapshot Manager configuration	VM configuration in Azure		Snapshot
	System managed identity (MI)	User managed identity (MI)	
System MI	CP-Permissions	N/A	Yes
	N/A	CP-Permissions	Yes
	N/A	■ CP-Permissions ■ Reader	N/A
	Reader	CP-Permissions	No
	CP-Permissions	Reader	Yes
	Reader	Reader	No
	CP-Permissions	CP-Permissions	Yes
User MI	CP-Permissions	N/A	N/A
	N/A	CP-Permissions	Yes
	Reader	CP-Permissions	Yes
	CP-Permissions	Reader	No
	Reader	Reader	No
	CP-Permissions	CP-Permissions	Yes
User MI (Reader)	N/A	■ CP-Reader ■ CP-Permissions	No

Note: In the above table, **CP-Permissions** is a role that has permission to take snapshot and**Reader** is a role that does not have permission to take the snapshot.

Performance issue with GCP backup from snapshot

During GCP backup from snapshot operation the data is read from persistent disks attached to the Snapshot Manager. Persistent disk IOPS speed gets split between disks if read operation is going on multiple disks on the same machine.

For GCP backup from snapshot operation, a maximum number of 15 jobs can be launched (on machine whose capability is more than 15) and if the capability of the machine is less than 15, then those many backup from snapshot operation can run parallel on Snapshot Manager.

If multiple backup from snapshot jobs are running, then **Effective IOPS for single disk = total disk input/output operations per second (IOPS) for read operation on machine/number of disk on which read operation is going on**. This results in longer backup times for the VM which have large size when large number of parallel backup jobs are going on.

Perform the following steps to improve the performance

1. Select higher configuration for the Snapshot Manager:

   GCP disk IOPS depends on number of factors like VM type, Disk type, Disk size, CPU and so on.

   Select higher configuration to get better IOPS. For more information, see Configure disks to meet performance requirements.

2. Limit the number of jobs running on Snapshot Manager:

   Use the following settings to limit the number of parallel jobs running on Snapshot Manager:

   ```
   [capability_limit]

   max_backup_jobs = 4
   ```

   If Snapshot Manager machines capability is less than `max_backup_jobs` then machines capability would be considered. If machines capability is more than `max_backup_jobs` then value of `max_backup_jobs` would be used to decide the number of backup from snapshot jobs to be run on machine. After changing the configuration restart the Snapshot Manager and complete manual discovery on NetBackup.

Post migration on host agents fail with an error message

Post migration on host agents fail with the following error mesage:

```
[1864] Failed to execute script flexsnap-agent
```

To resolve this issue, run the following respective commands:

- For Windows: From the command prompt navigate to the agent installation directory (`C:\Program Files\Veritas\CloudPoint\`) and run the following command:

```
#flexsnap-agent.exe --renew --token <auth_token> renew
```

This command fails in the first attempt. Rerun the command for successful attempt.

- For Linux: Rerun the following command on Linux host:

```
#sudo systemctl start flexsnap-agent.service --renew --token
<auth_token>
```

File restore job fails with an error message

The file restore job fails with the following error message in the job **Activity** monitor:

```
Unable to detect volume for disk <disk_name>
```

check if there is any network devise attached to the host, detach If exists. 2 open the command prompt in admin privileges. 3. run the command: diskpart 4. inside diskpart prompt type "rescan" and enter 5. exit the diskpart prompt and command line. 6. try restore again

To resolve this issue, perform the following:

- If any network device is attached to the device, detach it.
- Open the command prompt in admin privileges and run the following command:

```
diskpart
```

- Inside the diskpart prompt, type **rescan** and press enter.
- Exit the diskpart prompt and the command line.
- Perform the file restore operation again.