NetWitness® Platform

Version 12.5.1

SASE Hybrid Cloud Installation Guide

NETWITNESS

Platform

Contact Information

NetWitness Community at https://community.netwitness.com contains a knowledge base that answers common questions and provides solutions to known problems, product documentation, community discussions, and case management.

Trademarks

RSA and other trademarks are trademarks of RSA Security LLC or its affiliates ("RSA"). For a list of RSA trademarks, go to https://www.rsa.com/en-us/company/rsa-trademarks. Other trademarks are trademarks of their respective owners.

License Agreement

This software and the associated documentation are proprietary and confidential to RSA Security LLC or its affiliates are furnished under license, and may be used and copied only in accordance with the terms of such license and with the inclusion of the copyright notice below. This software and the documentation, and any copies thereof, may not be provided or otherwise made available to any other person.

No title to or ownership of the software or documentation or any intellectual property rights thereto is hereby transferred. Any unauthorized use or reproduction of this software and the documentation may be subject to civil and/or criminal liability. This software is subject to change without notice and should not be construed as a commitment by RSA.

It is advised not to deploy third-party repos or perform any change to the underlying NetWitness Operating System that is not part of the supported NetWitness version. Any such change outside of the NetWitness approved image may result in a service or functionality conflict and require a reimage of the NetWitness system to bring NetWitness back to an optimized functional state. In the event a third-party repo is deployed, or other non-supported change is made by the customer without NetWitness approval, the customer takes full responsibility for any system malfunction until the issue can be remediated through troubleshooting efforts or a reimage of the service.

Third-Party Licenses

This product may include software developed by parties other than RSA. The text of the license agreements applicable to third-party software in this product may be viewed on the product documentation page on NetWitness Community. By using this product, a user of this product agrees to be fully bound by terms of the license agreements.

Note on Encryption Technologies

This product may contain encryption technology. Many countries prohibit or restrict the use, import, or export of encryption technologies, and current use, import, and export regulations should be followed when using, importing or exporting this product.

Distribution

Use, copying, and distribution of any RSA Security LLC or its affiliates ("RSA") software described in this publication requires an applicable software license.

RSA believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." RSA MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Miscellaneous

This product, this software, the associated documentations as well as the contents are subject to NetWitness' standard Terms and Conditions in effect as of the issuance date of this documentation and which can be found at https://www.netwitness.com/standard-form-agreements/.

November, 2024

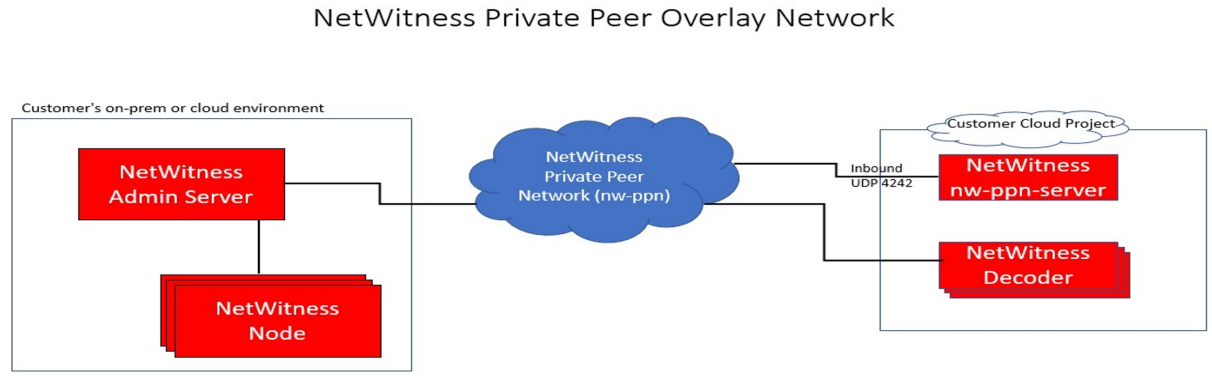
Contents

SASE Hybrid Cloud Overview

Customers opt for Secure Access Service Edge (SASE) technology to deliver WAN and security controls as a cloud computing service directly to the source connection device, using IOT or edge computing rather than a data center. This technology has accelerated recently, with the workforce opting to work remotely and from various locations. Customers opt for a more Hybrid Cloud model for SASE.

To provide hybrid cloud support for SASE, NetWitness has developed a new capability allowing the deployment of components (Decoder and Concentrator) in different cloud regions where the SASE vendor operates. Deployment is per tenant and can connect to an on-prem or cloud-based NetWitness Admin node over a secure network.

Core to this new NetWitness capability is the integration of an overlay network. The NetWitness Peer-to-Peer Network (NW-PPN) provides secure, mutually authenticated, PKI-based communication between NetWitness components. The NW-PPN is based on the Noise Protocol Framework, which leverages the open-source Slack Nebula implementation.



The only port required to be open is UDP port 4242 on the outbound NAT/Firewall specific to the Admin Server. Core to the NW-PPN is the NW-PPN Server. This is a Nebula Lighthouse service deployed on a NetWitness cloud image. The NW-PPN Server collects and provides all NW-PPN Nodes (AdminServer and SASE deployed cloud nodes) with UDP based networking connection information to support peer-to-peer communication via UDP hole punching technique. Additionally, where UDP hole punching is not supported, the NW-PPN Server supports fallback to relaying through the NetWitness NW-PPN Server.

This document describes the components and configuration options specific to the NetWitness SASE cloud deployment process. At the end of this document is a “Quick-Install" guide with an enumerated set of installation steps.

SASE Installation

The SASE Hybrid Cloud Configuration is a data driven design. The NetWitness Admin Server contains a script **nw-create-cloud-hybrid** that has a command **--enable-cloud-sase**, which will deploy the NetWitness Overlay Network, and the defined NetWitness Nodes in their respectively defined regions in the requested Cloud Platform. The 12.5.1.0 implementation supports deploying the overlay network and NetWitness nodes to the Google Cloud Platform (GCP). The recommended SASE deployment includes 3 cloud compute instances, a PPN-Server node to support Overlay Network communication, a NetWitness Decoder, and a NetWitness Concentrator. The deployed GCP based Decoder and Concentrator will have available storage based on the chosen host-model for each. The Decoder and Concentrator service and storage configuration must be completed after the SASE Deployment following the appropriate 12.5.1.0 NetWitness documentation.

GCP Prerequisites

1. Customer must have an On-Prem or cloud based NetWitness Admin Server at Version 12.5.1.0.
2. Customer must have a GCP Cloud Project.
3. Customer's GCP Cloud Project must have enabled the following Google APIs.
    a. Cloud Compute API
    b. Cloud Resource Manager API
    c. Identity and Access Management (IAM) API
    d. Cloud DNS API
4. Customer's on-prem Admin Server outbound network configuration must allow access to the external-ip of the ppn-server on UDP port 4242.
5. To deploy resources in the cloud, a service account credential file must be available on the NetWitness Admin Server that has the necessary permission to create vpcs, subnetworks, nats, disks, service accounts, and instances from images.
    a. To enable automated deployment of NetWitness resources to the customer's GCP Cloud Project, a JSON based Service Account Key File must be generated.
    b. Create a Service Account with the Roles mentioned below. To create a Service Account and its key in GCP, refer to https://cloud.google.com/iam/docs/service-accounts-create and https://cloud.google.com/iam/docs/keys-create-delete.

        Note: These roles are the default OOTB GCP roles. Permissions can be future limited with custom role(s).

        i. Compute Admin
        ii. IAP-secured Tunnel User
        iii. Service Account User
        iv. Storage Admin

c. Provide the service account email to NetWitness to give access to the gcp image. Email can be found in the token as client_email or from GCP console go to **IAM > Service Account**.

d. GCP JSON based service account file must be saved to the On-Prem NetWitness Admin Server to `/root/.gcp/gcp-auth-token.json`. This is the expected default location of the NetWitness Installation scripts. The JSON token file can be stored in another location on the NetWitness Admin Server, but it will require passing in that location during the SASE deployment installation.

e. GCP JSON Service Account File must be like:

```
{
"type": "service_account",
"project_id": "nw-nwp-xxx",
"private_key_id": "d143529509c56b28e7186fea465XXXXXXXXXXXXX",
"private_key": "-----BEGIN PRIVATE KEY-----<private key>-----END PRIVATE
KEY-----\n",
"client_email": "<email associated with service account>",
"client_id": "<client id>",
"auth_uri": "https://accounts.google.com/o/oauth2/auth",
"token_uri": "https://oauth2.googleapis.com/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/<service account
name>",
"universe_domain": "googleapis.com"
}
```

Note: The Service Account has significant privileges and should be disabled in GCP for security purposes when not required by a NetWitness SASE Deployment or Validation Action via the **nw-create-cloud-hybrid** script.

6. The customer's cloud project can access the NetWitness Production Project that hosts the cloud image that matches the version of the On-Prem NetWitness Admin Server. The minimum NetWitness version is 12.5.1.0.

GCP Details

a. The NetWitness GCP production project name is: **nw-onprem-images-prod**.

b. This project name MUST be provided as the value for the "image_project" attribute in the `sase-deployment-models.yml` configuration file prior to SASE Deployment. Details are provided in the later sections.

c. The NetWitness SASE Image Name will look like: **rsa-nw-12-5-1-0-<build #>-lite**. The build number can be found in the initial lines of the `/etc/netwitness/component-descriptor/data/nw-component-descriptor.json` file on the NetWitness Admin Server.

SASE Configuration

The number and type of NetWitness nodes broken out by region is defined in the `/opt/rsa/saTools/cloud/sase-deployment-models.yml` file on the NetWitness Admin Server. The file defines a templated configuration. This configuration requires customization based on the customer's exact cloud configuration. This template should be copied to the expected deployment location (`/root/.sase/sase-deployment-models.yml`) and customized to exactly align with the customer's deployment model preference(s). If using the default configuration, the template, `/opt/rsa/saTools/cloud/sase-deployment-models.yml` will be copied to the deployment location, `/root/.sase/sase-deployment-models.yml` when using the SASE Deployment Script (**nw-create-cloud-hybrid**).

The `sase-deployment-models.yml` file contains the cloud node deployment configuration. The template in `/opt/rsa/saTools/cloud/sase-deployment-models.yml` includes the following attribute definitions:

- **provider** – This is the cloud provider. Different providers will have different schemas. Currently only the Google Cloud Platform (GCP) is supported. Contains Cloud implementations.
- **gcp** - Google Cloud Platform provider. Contains model definitions.
- **default** – Out of the box model definition. The attributes in this model definition should be updated to suite the customer's requirements.
- **image** – This is the image name to be used as the base image in the gcp project. The image name is in the following format: "`rsa-nw-<version>-<build#>-lite`". If left blank, the SASE deployment configuration will default to the version and build number on the Admin Server (version and build number can be found in the initial lines of the `/etc/netwitness/component-descriptor/data/nw-component-descriptor.json` file on the admin server). The image name can be overridden with this attribute if required.
- **image_project** – NetWitness Cloud project that maintains the NetWitness SASE image. Must be set to: **nw-onprem-images-prod**.
- **vpn_provider** – This is the VPN provider for the SASE deployment model. The different providers have differing integration points with NetWitness which require varying configuration. Supported VPN providers: **Broadcom/PaloAlto/Netskope**.
- **vpc_ppn_cidr** – Virtual Private Cloud network address in C.I.D.R format. The value must not conflict with another defined VPC in the cloud project.
- **ppn_cidr** – NetWitness overlay network range to be used. It is in C.I.D.R format. This value must not conflict with networks on either the on-prem network that hosts the NetWitness Admin Server or in the cloud project. The default NetWitness Private Peer Network value is `172.30.30.0/24`.
- **admin_cidr_ip** – This is the Admin Server's nw-ppn ip address in C.I.D.R format. This address MUST be within the **ppn_cidr** range and suggested to be within the .2 value range.
- **default_region** – Region that is used by default to deploy the nw-ppn-server (Lighthouse server). Set to the VPN provider's region of deployment.

- **ppn_server** – This is the Nebula Lighthouse Server instance. It is based on a NetWitness image. The Nebula rpm and corresponding certs are installed, and the instance is configured as the nw-ppn-server (Lighthouse server in Nebula terminology).
  - **name** – Name of the instance known to the nw-ppn network.
  - **ppn_cidr_ip** – This node's ip in C.I.D.R format. This address MUST be within the **ppn_cidr** range and suggested to be within the .1 value range.
  - **zone_suffix** – This value is concatenated with the region to define the zone that the ppn-server will be installed into on GCP.
  - **machine_type** – The GCP machine type to use when creating the nw-ppn-server instance.
  - **boot_disk_size** – The size of the attached file system of the nw-ppn-server instance when it is created.
  - **boot_disk_type** – The type of attached boot disk.
  - **cloud_subnet** – The subnet address range in C.I.D.R format will be used to create a subnet for the nw-ppn-server node. This address MUST be in the **vpc_ppn_cidr** range.
  - **whitelist** - List of IPs or IP ranges in C.I.D.R. format that will be added to the ppn-server ingress firewall. This should be a comma-separated list of all externally facing outbound IPs that can be used to access the PPN server. It should be the complete list of edge proxy addresses that the AdminServer may try to route through to access the PPN server.
    For Example: '`10.11.12.13/32' or '10.11.12.13/32,10.11.12.0/24`'
- **regions** – Container for each region definition.

  Note: When adding new regions, the region's defined cloud_node_subnet address MUST be unique.

  - **us-east1** – Region definition. Add additional regions under the parent Regions node for additional region usage. Set to the VPN provider's region of deployment.
  - **region_name** – Exact name of region as defined by cloud provider, i.e., us-east1. Set to the VPN provider's region of deployment.
    - **cloud_node_subnet** – The subnet address range in C.I.D.R format that will be used to create a subnet for the nodes defined in the nw-nodes attribute in this region. This address MUST be in the **vpc_ppn_cidr** range and MUST be unique if multiple regions are defined.
    - **nw-nodes** – Container element defining all nodes that will be created within this region. Nodes do not necessarily get created in the order listed. The default set of nw-nodes is limited to a NetWitness Decoder and Concentrator. Node defined here using the following defined attributes will be created under the parent region.
  - **<node-name>** – Node to be created. This element is just an arbitrary name for the type of node to be created and provisioned in this region's subnet.
    - **name** – Name of the instance known to the nw-ppn network.
    - **zone_suffix** – This value is concatenated with the region to define the zone that the NetWitness node will be installed into on GCP.
    - **boot_disk_size** – The size of the attached boot disk.

- **boot_disk_type** – The type of attached boot disk.
- **nic_type** - The type of vNIC to be used on this interface.
  Possible values: `GVNIC, VIRTIO_NET`
- **egress_tier** - A selection of available network options that controls the egress bandwidth.
  Possible values: `TIER_1, DEFAULT`. The **nic-type** must be `GVNIC` for this setting to take effect.

  Note: machine_type must be a supported type.
- **model_name** – This is the host configuration model name. These configuration models are defined in `/root/.sase/host-models.yml` which provides the host drive and machine type configuration attributes.
- **additional_storage** – This flag dictates the deployment of the drive model defined in `host-models.yml` file. Must be set to true in production.
- **bootstrap** – Used to determine if the calling script will automatically bootstrap and accept the node keys in the Admin Server. This allows for either automated or manual orchestration of a NetWitness Category to the node.
- **orchestrate** – Used to determine if the calling script will automatically orchestrate a NetWitness Category to the node.
- **category** – The NetWitness Category to be orchestrated. Must be an exact value (Case Sensitive).
- **Configure_block_storage** – Automates the block storage configurations for the SASE Cloud node.
- **Configure_warm_storage** – Automates the warm packet storage configurations for the SASE Cloud node.

The `/opt/rsa/saTools/cloud/host-models.yml` defines the available and tested storage models for the specific NetWitness version. The `/opt/rsa/saTools/cloud/host-models.yml` will be copied to the deployment location, `/root/.sase/host-models.yml` when using the SASE Deployment Script (**nw-create-cloud-hybrid**) on first use. The host model chosen from the `host-models.yml` file is very customer specific and MUST be specified by updating the node configuration's host_model value of the `sase-deployment-models.yml` as defined above. The following is the available **production** host_model options for Decoder and Concentrator SASE Nodes:

Model	Description
c1r6m30	Defines storage configuration for decoder at 1gbps capture (c1) with 6-day retention (r6) and 30-day meta retention (m30) for concentrator.
c1r12m60	Defines storage configuration for decoder at 1gbps capture (c1) with 12-day retention (r12) and 60-day meta retention (m60) for concentrator.
c1r23m120	Defines storage configuration for decoder at 1gbps capture (c1) with 23-day retention (r23) and 120-day meta retention (m120) for concentrator.

To help determine the appropriate model to choose, see the SASE section in the *NetWitness Storage Guide* for more details.

Each host model defines the following:

models – Container for all the above models.

- **<model-name> i.e., c1r6m30** – Storage model for 1gpbs capture with 6-day decoder packet retention and 30-day concentrator retention.
- **Decoder** – Contains attributes specific to a Decoder deployment.
  - **machine_type** – Defines the virtual machine type. Ex: **n2-standard-32**
  - **storage_class** – Defines the type of cloud storage. Ex: **STANDARD**. This attribute is not currently used but is defined for future use.
  - **warm_retention** – Size (in TB) of cloud/bucket storage used for warm retention. This attribute is not currently used but is defined for future use.
  - **disks** – Defines the block storage disk properties for NW services. Disk properties for each NW service (Decoder, Concentrator) must be defined separately.
  - **decodersmall** – Decoder service storage volume name for meta/session/index databases. Multiple volumes must be created when the disk size exceeds 65000GB. The volume names are incremented starting with 0. Ex: **decodersmall0**
    - **disk_name** – Unique name for disk. Ex: **decodersmall**
    - **disk_type** – Type of disk. Ex: **pd-standard**
    - **disk_size** – Size (in GB) for the above disk.
  - **decoder** – Decoder service storage volume name for packet database. Multiple volumes must be created when the disk size exceeds 65000GB. The volume names are incremented starting with 0. Ex: **decoder0**
    - **disk_name** – Unique name for disk. Ex: **decoder**
    - **disk_type** – Type of disk. Ex: **pd-standard**
    - **disk_size** – Size (in GB) for the above disk.
- **Concentrator** – Contains attributes specific to a Concentrator deployment.
  - **machine_type** – Defines the virtual machine type. Ex: **n2-standard-32**
  - **storage_class** – Defines the type of cloud storage. Ex: **STANDARD**
  - **warm_retention** – Size (in TB) of cloud/bucket storage used for warm retention. This attribute is not currently used but is defined for future use.
  - **disks** – Defines the block storage disk properties for NW services. Disk properties for each NW service (Decoder, Concentrator) must be defined separately.
  - **concentrator** – Concentrator service storage volume name for meta/session databases. Multiple volumes must be created when the disk size exceeds 65000GB. The volume names are incremented starting with 0. Ex: **concentrator0**
    - **disk_name** – Unique name for disk. Ex: **concentrator**
    - **disk_type** – Type of disk. Ex: **pd-standard**
    - **disk_size** – Size (in GB) for the above disk.

- **index** – Concentrator service storage volume name for index database. Multiple volumes must be created when the disk size exceeds 65000GB. The volume names are incremented starting with 0. Ex: **index0**
  - **disk_name** – Unique name for disk. Ex: **decoder**
  - **disk_type** – Type of disk. Ex: **pd-standard**
  - **disk_size** – Size (in GB) for the above disk.

nw-create-cloud-hybrid Command Help

Run the following command as root on the NetWitness Admin Server via the command line for all available options:

nw-create-cloud-hybrid *--help*

```
[root@1191Standby ~]# nw-create-cloud-hybrid --help
Usage:
 nw-create-cloud-hybrid command [options]

Commands:
 --enable-cloud-sase            Deploys NetWitness SASE focused assets to the cloud
 --disable-cloud-sase           Undeploys NetWitness SASE focused assets from the cloud
 --upgrade-overlay-network      Upgrades overlay network resources
 --reissue-all-certs            reissues all overlay network certificates
 --reissue-node-certs           reissues overlay network certificates for a specific node
     required parameter:
       --uuid                   UUID of NetWitness Node (see nw-manage -l)
 --backup-cloud-nodes, -b       Backup configuration of all cloud nodes
 --restore-cloud-node, -r       Restores configuration of specified Cloud node
     required parameter:
       --uuid                   UUID of NetWitness Cloud Node (see nw-manage -l)
 --check-overlay-status, -c     Checks inter-connectivity of all nw-ppn overlay network hosts
 --check-cert-status, -s        Checks overlay network certificate expiration status

Command Options:
     --cloud-provider           Required: Destination cloud provider (gcp|aws)
     --deployment-model         Optional Name of deployment model in template
                                  defaults to pre-defined '(gcp|aws) default'
     --cloud-key-path           Optional Cloud based Service Account key data path
                                  defaults to .(gcp|aws) specific file
```

SASE Deployment

To deploy your SASE based nodes in GCP, run the following command as root on the NetWitness Admin Server via the command line:

nw-create-cloud-hybrid *--enable-cloud-sase*

Options	Description
--deployment-model	*Optional Name of deployment model in template defaults to pre-defined 'gcp default'*
--cloud-key-path	*Optional Cloud Service Account Json-based key data path GCP will default to /root/.gcp/gcp-auth-token.json*

This will deploy the nodes defined in the `sase-deployment-models.yml` file. The default installation will deploy a NetWitness Decoder, Concentrator, and PPN Server Instance and required ancillary resources (vpc, subnetworks, persistent drives, buckets, etc.). The deployed instances will utilize the 'nw' namespace by default for their naming convention prefix. i.e., nw-ppn-server, nw-decoder<region><zone>, nw-concentrator<region><zone>.

SASE Undeployment

To undeploy your SASE based nodes in the cloud provider, run the following command as root on the NetWitness Admin Server via the command line.

nw-create-cloud-hybrid *--disable-cloud-sase*

Note: The undeploy action will remove all objects in the cloud. It will request removal confirmation of specific objects during the undeploy. These objects include each node, subnet, vpcs, etc.

Options	Description
--deployment-model	*Optional Name of deployment model in template* *defaults to pre-defined 'gcp default'*
--cloud-key-path	*Optional Cloud Service Account Json-based key data path* *GCP will default to /root/.gcp/gcp-auth-token.json*

This undeploys all NetWitness SASE created cloud-based assets.

SASE Backup

Deployed SASE Nodes can be backed up using the same approach as all other NetWitness Nodes. The **nw-create-cloud-hybrid** script has a **--backup-cloud-nodes** option that will utilize the underlying NetWitness backup capability but will support and facilitate the automated restoration of a cloud based NetWitness SASE node. The following command must be executed using the **nw-create-cloud-hybrid** script on the Admin Server:

nw-create-cloud-hybrid *--backup-cloud-nodes*

or

nw-create-cloud-hybrid *--b*

Options	Description
--deployment-model	*Optional Name of deployment model in template* *defaults to pre-defined 'gcp default'*
--cloud-key-path	*Optional Cloud Service Account Json-based key data path* *GCP will default to /root/.gcp/gcp-auth-token.json*

This command will take a backup of the configuration of all SASE based deployed NetWitness Cloud Nodes and will store the backup configurations in the `/root/.sase/backups` directory.

Note: The Admin Server must be backed up using the standard disaster recovery/recovery-tool procedures.

Note: This process only backs up the node configuration of the cloud nodes. This command must be executed after any configuration updates to the cloud deployed NetWitness SASE Nodes.

SASE Restore

Deployed SASE Nodes can be restored if the deployed SASE NetWitness Node is in an unrecoverable state. To recover/restore a previously deployed SASE NetWitness Cloud Node, the node's configuration must have been previously backed up in the `/root/.sase/backups/` directory using the **nw-create-cloud-hybrid *--backup-cloud-nodes*** command, and the instance must be completely removed/deleted from the cloud provider project. Additionally, the node must still exist in the NetWitness Configuration (see ***nw-manage --l***). The **nw-create-cloud-hybrid** script has a ***--restore-cloud-node*** option that will utilize the underlying NetWitness recovery capability and facilitate the automated restoration of a cloud based NetWitness node. The following command must be executed using the **nw-create-cloud-hybrid** script on the Admin Server:

nw-create-cloud-hybrid *--restore-cloud-node*

or

nw-create-cloud-hybrid *--r*

Options	Description
--uuid	*UUID of the specific node (Required)*
--deployment-model	*Optional Name of deployment model in template* *defaults to pre-defined 'gcp default'*
--cloud-key-path	*Optional Cloud Service Account Json-based key data path* *GCP will default to /root/.gcp/gcp-auth-token.json*

This command will recreate the previously deployed SASE based cloud instance and apply the configuration of that node based on the last backup available in the `/root/.sase/backups` directory.

Note: This process only restores one node based on the required *--uuid* parameter. The uuid for a specific node can be found using the **nw-manage *--l*** command..

SASE Upgrade

The NetWitness Peer-to-Peer network (Nebula) is not upgraded by default when doing a NetWitness system upgrade. Additional steps are required. After a NetWitness upgrade of the Admin Server, the following command must be executed using the **nw-create-cloud-hybrid** script on the Admin Server:

nw-create-cloud-hybrid *--upgrade-overlay-network*

Options	Description
--deployment-model	*Optional Name of deployment model in template defaults to pre-defined 'gcp default'*
--cloud-key-path	*Optional Cloud Service Account Json-based key data path GCP will default to /root/.gcp/gcp-auth-token.json*

This upgrades the version of nebula on all NetWitness SASE created cloud-based assets and the Admin Server to the version installed when the NetWitness Admin Server was upgraded.

Reissue All Certificates

The NetWitness Peer-to-Peer network (Nebula) is secured with PKI based transport encryption. A Certificate Authority (CA) is created on the Admin Server and all SASE based nodes, the PPN-Server (Lighthouse) and the Admin Server are all issued and configured with node certificates to enable secure internode communication.

The Nebula certificates are created with expirations that match the Platform based certificate policy. The Nebula CA Certificate is issued with a 10-year expiration while the node certificates have a 3-year expiration.

All Certificates (CA and node certs) can be reissued via the following command:

nw-create-cloud-hybrid *--reissue-all-certs*

Options	Description
--deployment-model	*Optional Name of deployment model in template defaults to pre-defined 'gcp default'*
--cloud-key-path	*Optional Cloud Service Account Json-based key data path GCP will default to /root/.gcp/gcp-auth-token.json*

This command replaces all Nebula specific certificates in the SASE Deployment.

Reissue Node Certificates

The NetWitness Peer-to-Peer network (Nebula) is secured with PKI based transport encryption. A Certificate Authority (CA) is created on the Admin Server and all SASE based nodes, the PPN-Server (Lighthouse) and the Admin Server are all issued and configured with node certificates to enable secure internode communication.

The Nebula certificates are created with expirations that match the Platform based certificate policy. The Nebula CA Certificate is issued with a 10-year expiration while the node certificates have a 3-year expiration. A specific node's certificates (private/public) can be reissued and applied via the following command:

nw-create-cloud-hybrid *--reissue-node-certs*

Options	Description
--uuid	*UUID of the specific node (Required)*
--deployment-model	*Optional Name of deployment model in template defaults to pre-defined 'gcp default'*
--cloud-key-path	*Optional Cloud Service Account Json-based key data path GCP will default to /root/.gcp/gcp-auth-token.json*

This command replaces the Nebula certificates for the specified node in the SASE Deployment.

Check Certificate Status

The NetWitness Peer-to-Peer network (Nebula) is secured with PKI based transport encryption. A Certificate Authority (CA) is created on the Admin Server and all SASE based nodes, the PPN-Server (Lighthouse) and the Admin Server are all issued and configured with node certificates to enable secure internode communication.

The Nebula certificates are created with expirations that match the Platform based certificate policy. The Nebula CA Certificate is issued with a 10-year expiration while the node certificates have a 3-year expiration. All issued Nebula Certificates can be shown with their respective creation and expiration dates with the following command:

nw-create-cloud-hybrid *--check-cert-status*

or

nw-create-cloud-hybrid *--s*

Options	Description
--deployment-model	*Optional Name of deployment model in template defaults to pre-defined 'gcp default'*
--cloud-key-path	*Optional Cloud Service Account Json-based key data path GCP will default to /root/.gcp/gcp-auth-token.json*

Check Overlay Network Status

The NetWitness Peer-to-Peer network (Nebula) provides secure NetWitness inter-node communication. There can be a multitude of reasons for communication failures between nodes that includes but is not limited to expired certificates, improper network configuration, nodes shutdown, firewall configuration, required services shutdown or not responding, etc.

This check tests “Full SASE inter-node” communication. It checks the connectivity from the Admin Server and each SASE based node to the Admin Server or other SASE nodes, not just connectivity from the Admin Server to the remote cloud nodes. Full inter-node NetWitness Overlay node communication can be easily checked with following command:

nw-create-cloud-hybrid *--check-overlay-status*

or

nw-create-cloud-hybrid *--c*

Options	Description
--deployment-model	*Optional Name of deployment model in template defaults to pre-defined 'gcp default'*
--cloud-key-path	*Optional Cloud Service Account Json-based key data path GCP will default to /root/.gcp/gcp-auth-token.json*

GCP Cloud Quick-Install Guide

Perform the following steps to deploy the NetWitness SASE Nodes:

1. Install/Update NetWitness version 12.5.1.0 on On-Prem NetWitness Admin Server.
2. To create a Service Account and assign Roles, see **GCP Prerequisites 5. b.** under the SASE Installation section.
3. Copy the GCP Authentication Credential JSON File on the On-Prem Admin Server to `/root/.gcp/gcp-auth-token.json`. See **GCP Prerequisites 5. e.** under the SASE Installation section.
4. Copy `/opt/rsa/saTools/cloud/sase-deployment-models.yml` to `/root/.sase/sase-deployment-models.yml`. See SASE Installation section for more details.
5. Edit the `/root/.sase/sase-deployment-models.yml` file to customize the deployment. The deployment is data driven and very flexible. Most, **but not all**, attributes have been defaulted and do not need to be updated for the standard 12.5.1.0 NetWitness SASE deployment. See SASE Configuration section for full attributes list and definitions. The following attributes though, **MUST** be updated for a successful deployment to GCP in the **provider** > **gcp** > **default** node:
   a. **image**: Set to latest available image name for version. i.e., **rsa-nw-12-5-1-0-<build #>-lite**
   b. **image_project**: Set to **nw-onprem-images-prod**. This is the NetWitness GCP Project that contains the available NetWitness Product images.
   c. **vpn_provider**: Set to one of the available supported customers specific vpn providers (**Broadcom/PaloAlto/Netskope**).
   d. **default_region**: Set to the VPN provider's region of deployment.
   e. **ppn_server** > **cloud_subnet**: Update if default value conflicts with currently deployed sub-network.
   f. **regions**: Update region node name from default (us-east1) to the VPN provider's region of deployment.
   g. **regions** > **<preferred region>.region_name**: Update region name value from default (us-east1) the VPN provider's region of deployment.
   h. **regions** > **<preferred region>.cloud_node_subnet**: Update if default value conflicts with currently deployed sub-network.
   i. Under the node: **provider** > **gcp** > **default** > **regions** > **<preferred region>.region_name** > **nw_nodes**: Update the following attributes.
      i. **decoder** > **model_name**: Set to available production value. See Host Models subsection in the SASE Configuration Section above.
      ii. **decoder** > **additional_storage**: Set to true to add required persistent disks defined based on the model_name above.
      iii. **concentrator** > **model_name**: Set to available production value. See Host Models subsection in the SASE Configuration Section above.
      iv. **concentrator** > **additional_storage**: Set to true to add required persistent disks defined based on the model_name above.

6. From Admin Server command line, execute: **nw-create-cloud-hybrid *--enable-cloud-sase***.

Limitations

1. SASE deployment does not support an Admin Server that utilizes the NetWitness NAT capability.
2. Nebula-rpm is not automatically updated if newer version on subsequent upgrades (use **nw-create-cloud-hybrid *--upgrade-overlay-network*** to upgrade).
3. Certificate/PKI infrastructure independent of NetWitness PKI:
4. Full Overlay Network Certificate reissue is automated with **nw-create-cloud-hybrid *--reissue-all-certs***.
5. The **nw-create-cloud-hybrid *--restore-cloud-node*** command uses the active `sase-deployment-models.yml` and `host-models.yml` files located in `/root/.sase/` directory. The ***--restore-cloud-node*** command restores the cloud nodes based on their respective backup tar files located in `/root/.sase/backups` directory. This directory also contains the `sase-deployment-models.yml` and `host-models.yml` files which were also backed up at the time the ***--backup-cloud-nodes*** command was executed. The `sase-deployment-models.yml` and `host-models.yml` files in the `/root/.sase/backups` directory must be used when executing the ***--restore-cloud-node*** command. To resolve, do the following:

   a. Make a copy of the active `sase-deployment-models.yml` and `host-models.yml` files located in `/root/.sase/` directory and name them `sase-deployment-models.yml.orig` and `host-models.yml.orig`:

      ```
      - cp /root/.sase/sase-deployment-models.yml /root/.sase/sase-deployment-models.yml.orig
      - cp /root/.sase/host-models.yml /root/.sase/host-models.yml.orig
      ```

   b. Override the `sase-deployment-models.yml` and `host-models.yml` files located in `/root/.sase/` with the ones in `/root/.sase/backup` directory:

      ```
      - yes | cp /root/.sase/backup/sase-deployment-models.yml /root/.sase/
      - yes | cp /root/.sase/backup/host-models.yml /root/.sase/
      ```

   c. Execute restore command(s):

      ```
      - nw-create-cloud-hybrid --restore-cloud-node --uuid <uuid of cloud node>
      ```

   d. Restore original `sase-deployment-models.yml` and `host-models.yml` files:

      ```
      - yes | cp /root/.sase/sase-deployment-models.yml.orig /root/.sase/sase-deployment-models.yml
      - yes | cp /root/.sase/host-models.yml.orig /root/.sase/host-models.yml
      ```

SASE Node(s) Backup and Restore using GCP Snapshots

NetWitness provides backup/restore of SASE node(s) in GCP that follows the same process as on-prem node(s). GCP provides a snapshot feature to backup GCP instances periodically and restore them in case of node destruction or failure.

SASE Node-X backup and restore uses Google snapshots.

Refer to Snapshots overview | Filestore | Google Cloud for more details on snapshots.

Note: Snapshots are taken only for VM boot disk instances and not configured for storage disks (For example, disks defined in `host-models.yml` configured for NW databases such as packetdb/index).

Perform the following tasks to backup/restore the SASE node(s):

- Task 1. Create Snapshot of Orchestrated SASE Node-X
- Task 2. Create Snapshot Schedule for Orchestrated SASE Node-X
- Task 3. Restore SASE Node-X (For example, concentrator or decoder) from Snapshot

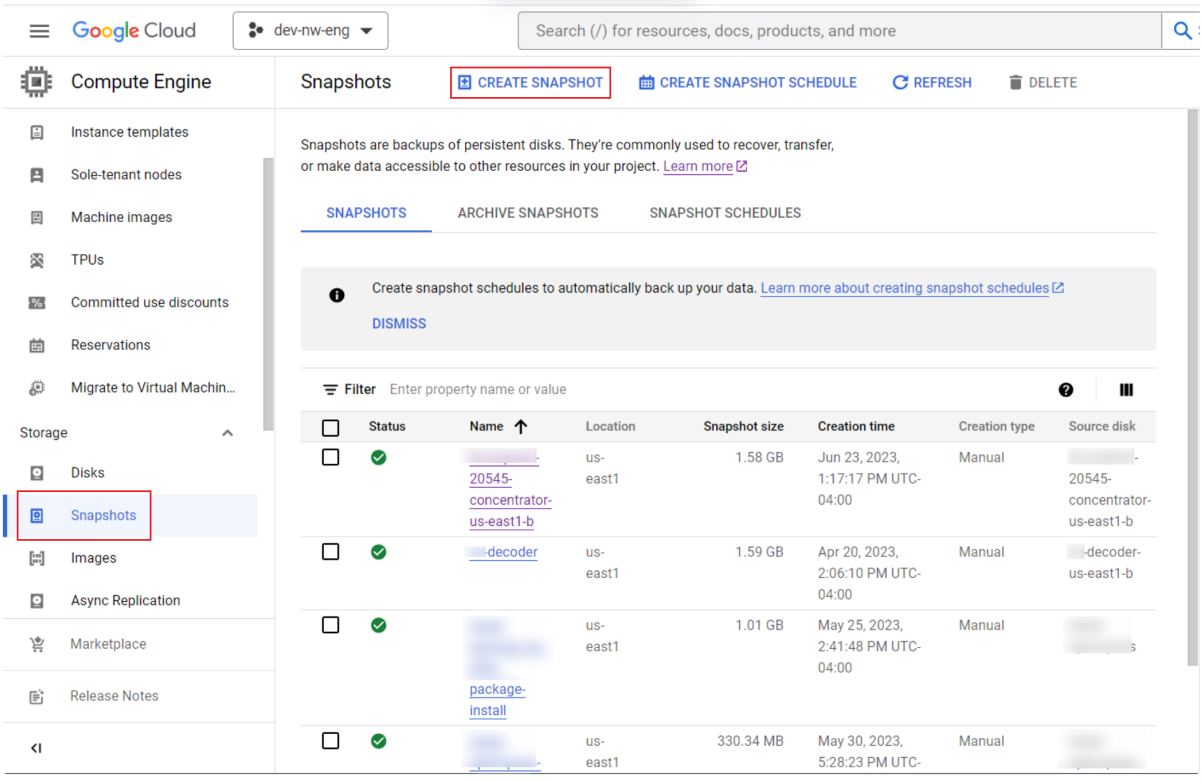
Task 1. Create Snapshot of Orchestrated SASE Node-X

During the orchestration of SASE Node-X, if the storage is set to `true` under the `Storage` section of `/opt/rsa/saTools/cloud/sase-deployment-models.yml`, additional disks are created with the specifications defined under the `Storage` section. These disks are configured as storage devices for the orchestrated Node-X service. Currently, both the `Decoder` and `Concentrator` are supported. The snippet below describes the creation of two disks, `concentrator0` and `index0`, with disk type and size.

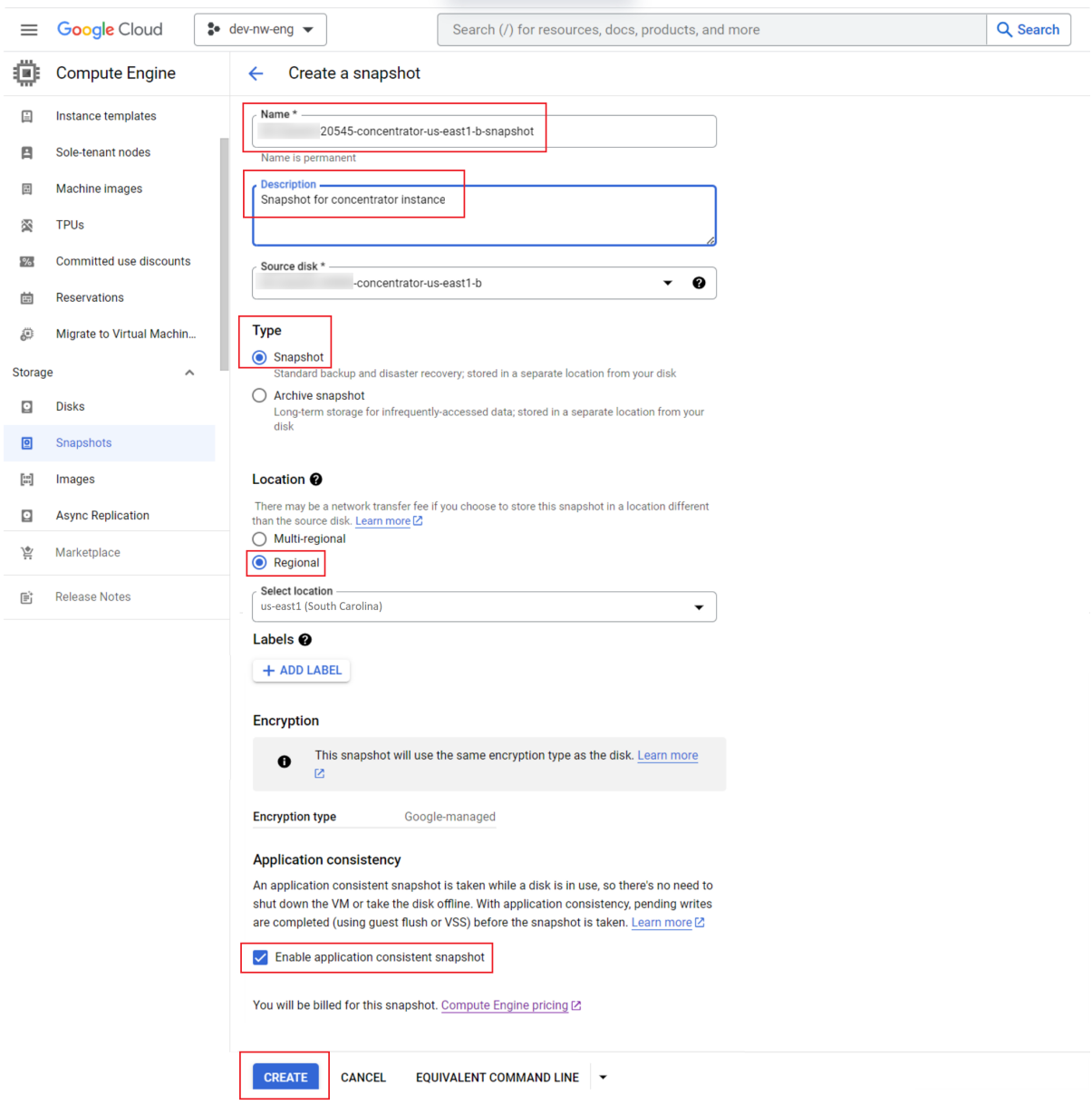
```
storage:
                boot_disk_size: 196
                boot_disk_type: pd-standard
                additional_disks: true
                disks:
                  # allocate to concentrator root, metadb, sessiondb
                  concentrator0:
                    disk_name: concentrator0
                    disk_type: pd-standard
                    disk_size: 100
                  # allocate to index
                  index0:
                    disk_name: index0
                    disk_type: pd-standard
                    disk_size: 10
```

Follow the below steps to create snapshot of orchestrated SASE Node-X:

1. Login to the GCP account, select `Snapshots > CREATE SNAPSHOT`.



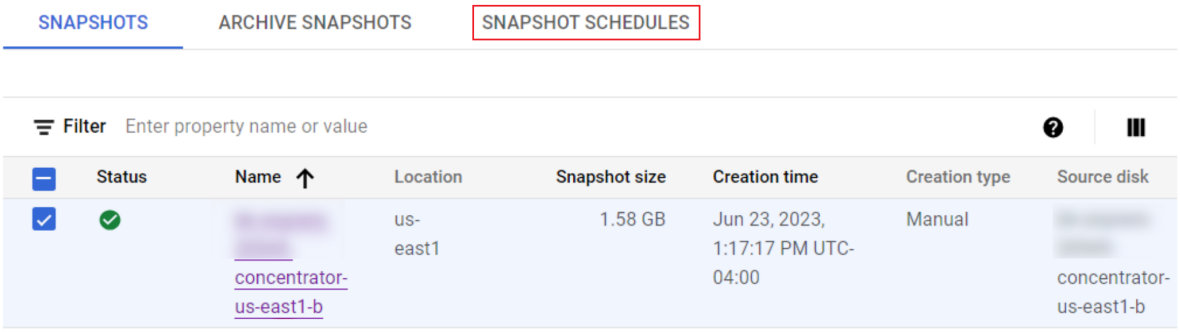
2. Fill in all the mandatory fields such as `Name` (For example, VM name + snapshot i.e., `20545-concentrator-us-east1-b-snapshot`, `Type` (Snapshot), `Location` - Regional (Same as the VM instance location. In this case: `us-east-1`), select `Enable application consistent snapshot` checkbox and click **Create** to create the snapshot.



`gcloud cli : Create a snapshot`

```
gcloud compute snapshots create  20545-concentrator-us-east1-b-snapshot-1
--project=dev-nw-eng
--source-disk=bk-onprem-20545-concentrator-us-east1-b
--source-disk-zone=us-east1-b
--storage-location=us
```

3. The new snapshot is displayed in the snapshot window.

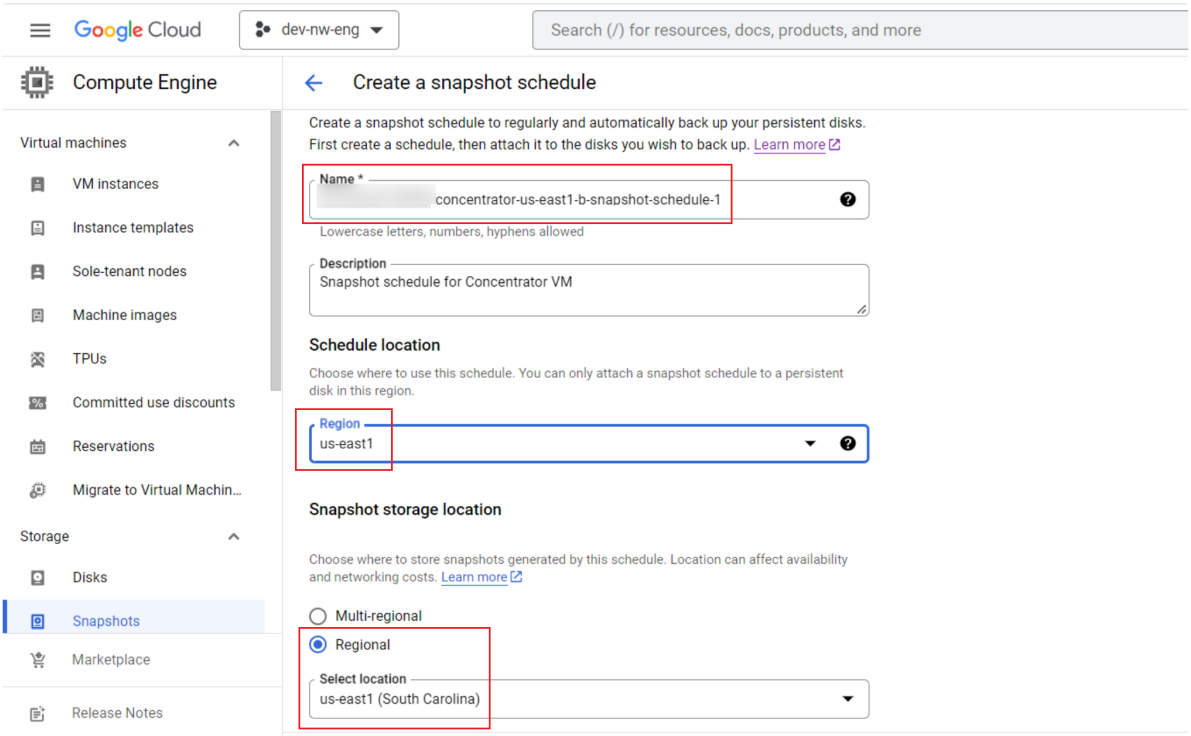


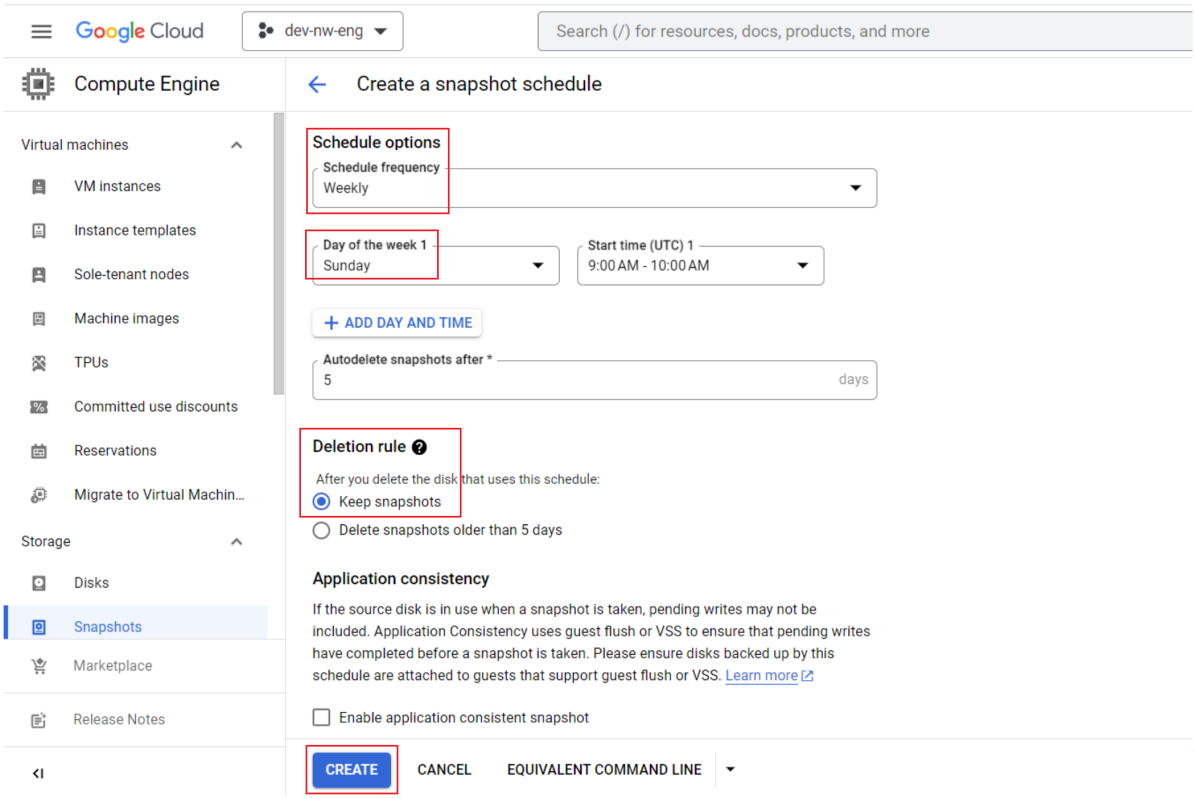
Task 2. Create Snapshot Schedule for Orchestrated SASE Node-X

Refer to Create schedules for disk snapshots | Compute Engine Documentation | Google Cloud for more details.

Follow the below steps to create snapshot schedule for orchestrated SASE Node-X:

1. Select the `SNAPSHOT SCHEDULES` tab to create snapshot schedules for this instance.

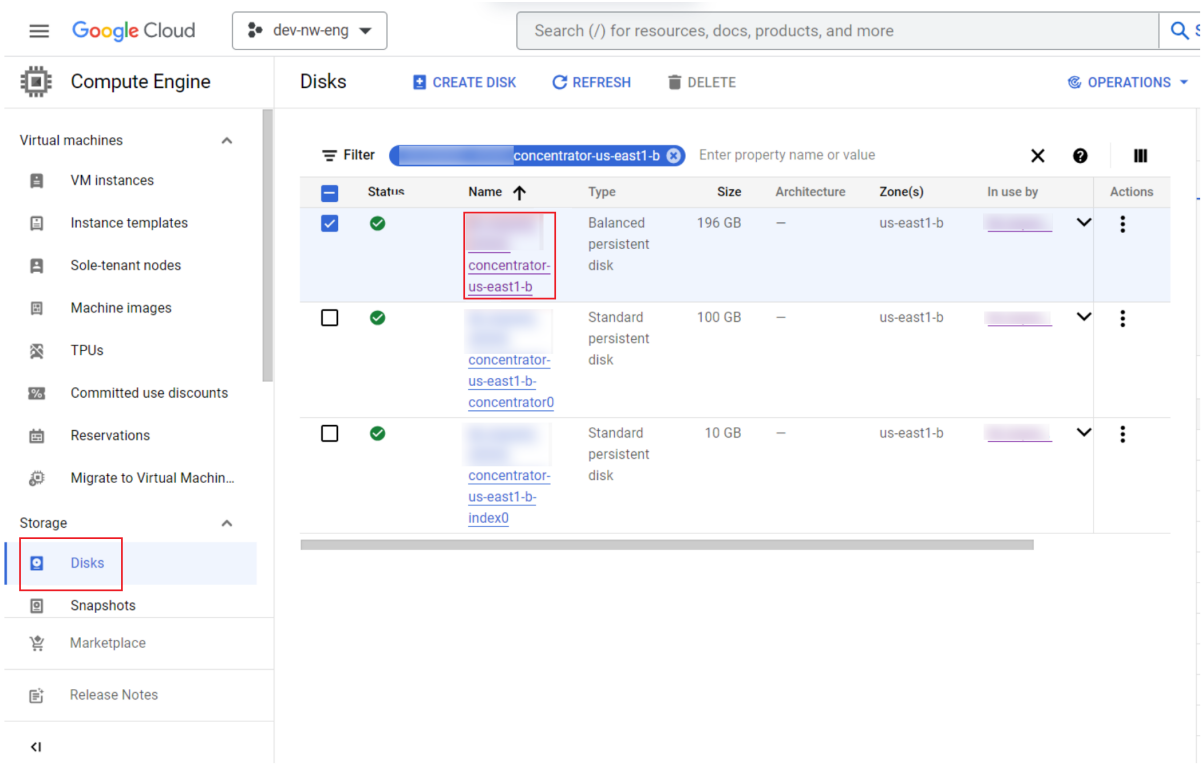




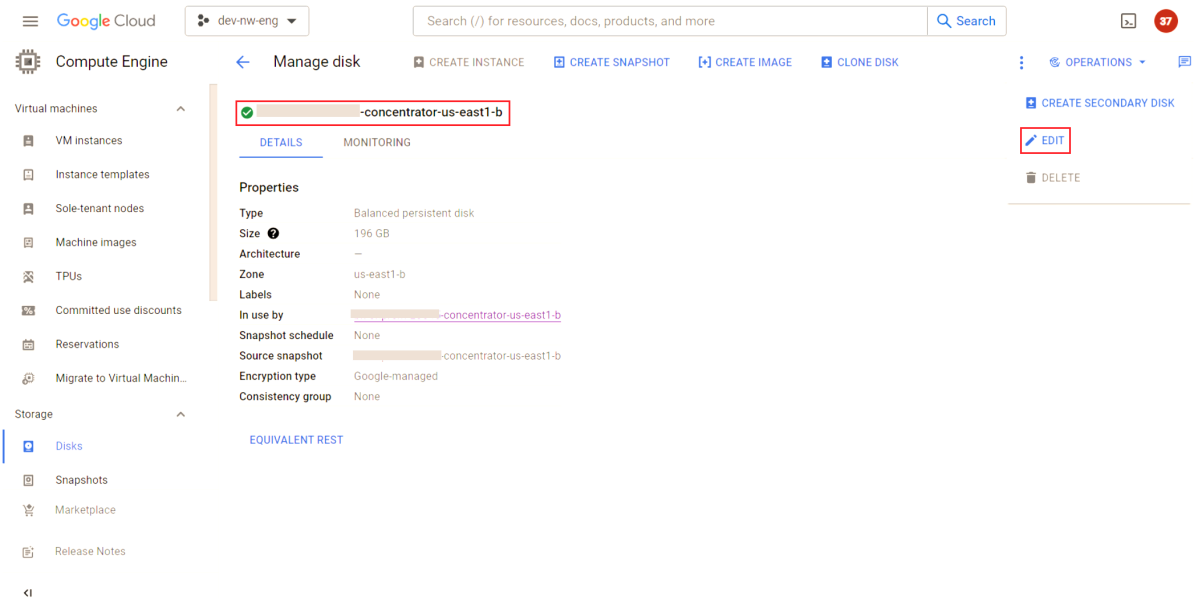
`gcloud cli:`

```
gcloud compute resource-policies create snapshot-schedule schedule-1
--project=dev-nw-eng
--region=us-central1
--max-retention-days=14
--on-source-disk-delete=keep-auto-snapshots
--daily-schedule
--start-time=11:00
--storage-location=us-central1
```

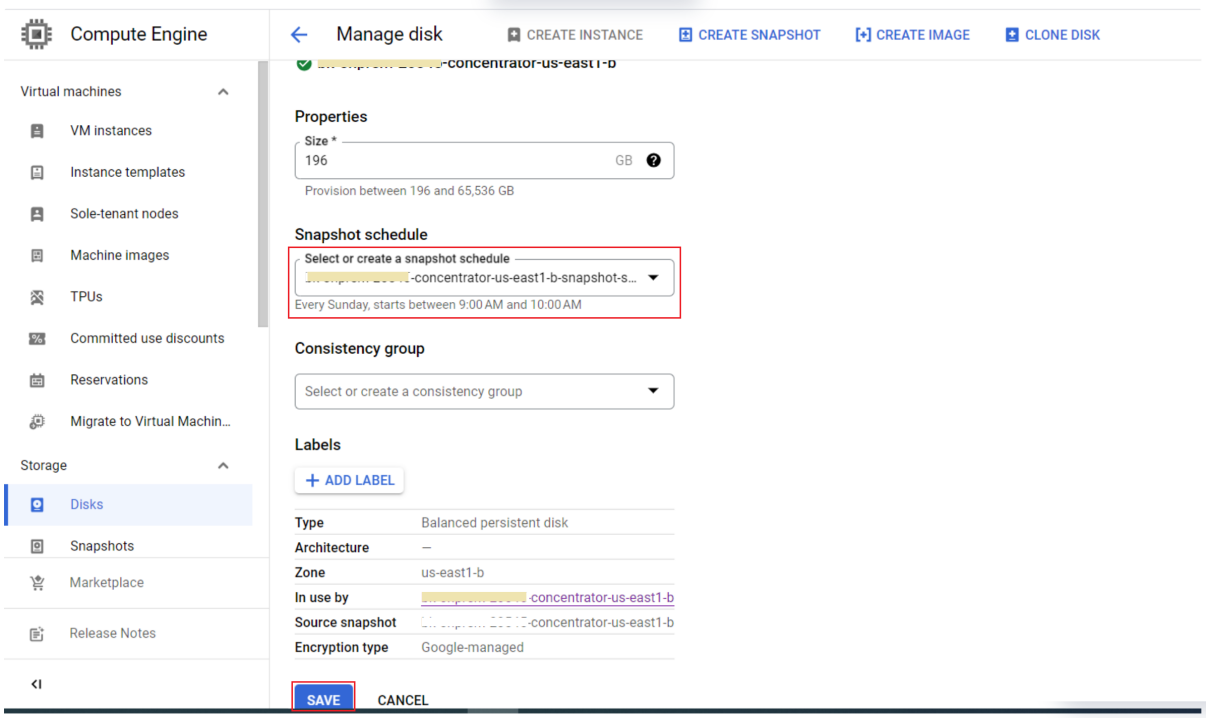
2. Refer to the link Create schedules for disk snapshots | Compute Engine Documentation | Google Cloud.



Click on the `Name` to select the disk. Click **Edit** to change the properties.



3. Select the `Snapshot schedule` from the drop-down. This is the schedule created in the previous step. The selected schedule details are described below the snapshot drop-down. Click **Save**.




```
gcloud cli: Attach a snapshot schedule to the persistent boot disk image.

gcloud compute disks add-resource-policies [DISK_NAME] \
    --resource-policies [SCHEDULE_NAME] \
    --zone [ZONE]
```

Task 3. Restore SASE Node-X (For example, concentrator or decoder) from Snapshot

1. Identify the snapshot corresponding to the image that needs restoration.
2. Note the failed Node-X `Name` and `Network` settings (`gcp - IP, Subnet,` and `gateway`). The new image MUST have the same name and IP settings as the failed node. The name/ip address are added to `/etc/hosts` of all the nodes in the NW environment. Fixed static IP assignment is an option.
3. Attach the `Storage` disks of the failed instance to the new instance restored from a snapshot. When restoring an image from a snapshot, the `/etc/fstab` is restored and contains the mounts for the `Storage` disks. These disks MUST be attached before the NW server (For example, concentrator or decoder) is booted.
4. `SSH` to the new instance and confirm that the service is running. Next, log in to the Admin server UI and confirm that the service on the new instance is online.

Original Settings

```
[root@20545-concentrator-us-east1-b ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000
      link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
      inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
      inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1460 qdisc mq state UP group
default qlen 1000
      link/ether 42:01:0a:0a:14:08 brd ff:ff:ff:ff:ff:ff
      inet 10.10.20.8/32 brd 10.10.20.8 scope global dynamic eth0
            valid_lft 2686sec preferred_lft 2686sec
      inet6 fe80::4001:aff:fe0a:1408/64 scope link
            valid_lft forever preferred_lft forever
3: nebula1: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1300 qdisc pfifo_
fast state UNKNOWN group default qlen 500
      link/none
      inet 172.30.30.4/24 scope global nebula1
            valid_lft forever preferred_lft forever
      inet6 fe80::e337:8b00:d411:132a/64 scope link flags 800
            valid_lft forever preferred_lft forever
[root@20545-concentrator-us-east1-b ~]# cat /etc/fstab
#
# /etc/fstab
# Created by anaconda on Tue May 23 18:49:10 2023
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/netwitness_vg00-root / xfs defaults 0 0
UUID=7e02d023-b265-4f18-9f4f-3aecf28681bf /boot xfs defaults 0 0
/dev/mapper/netwitness_vg00-usrhome /home xfs nosuid 0 0
/dev/mapper/netwitness_vg00-varlog /var/log xfs defaults 0 0
/dev/mapper/netwitness_vg00-nwhome /var/netwitness xfs nosuid,noatime 0 0
/dev/mapper/netwitness_vg00-swap swap swap defaults 0 0
/dev/concentrator/root /var/netwitness/concentrator xfs noatime,nosuid 1 2
/dev/concentrator/sessiondb /var/netwitness/concentrator/sessiondb xfs
noatime,nosuid 1 2
/dev/concentrator/metadb /var/netwitness/concentrator/metadb xfs
noatime,nosuid 1 2
```

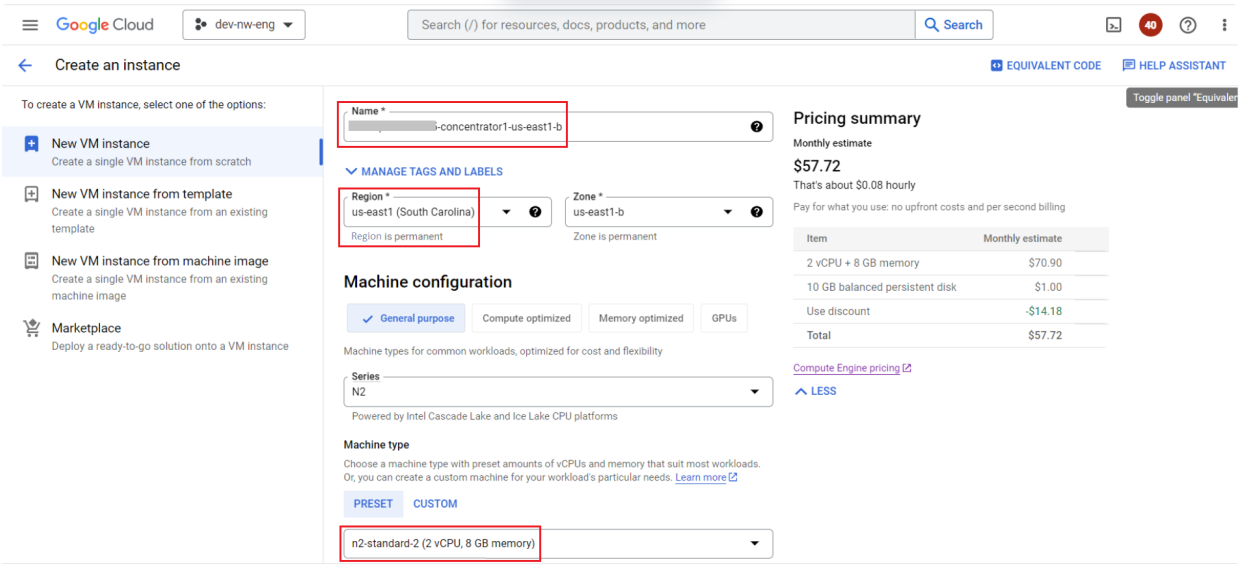
```
/dev/index/index /var/netwitness/concentrator/index xfs noatime,nosuid 1 2
[root@20545-concentrator-us-east1-b ~]#
```

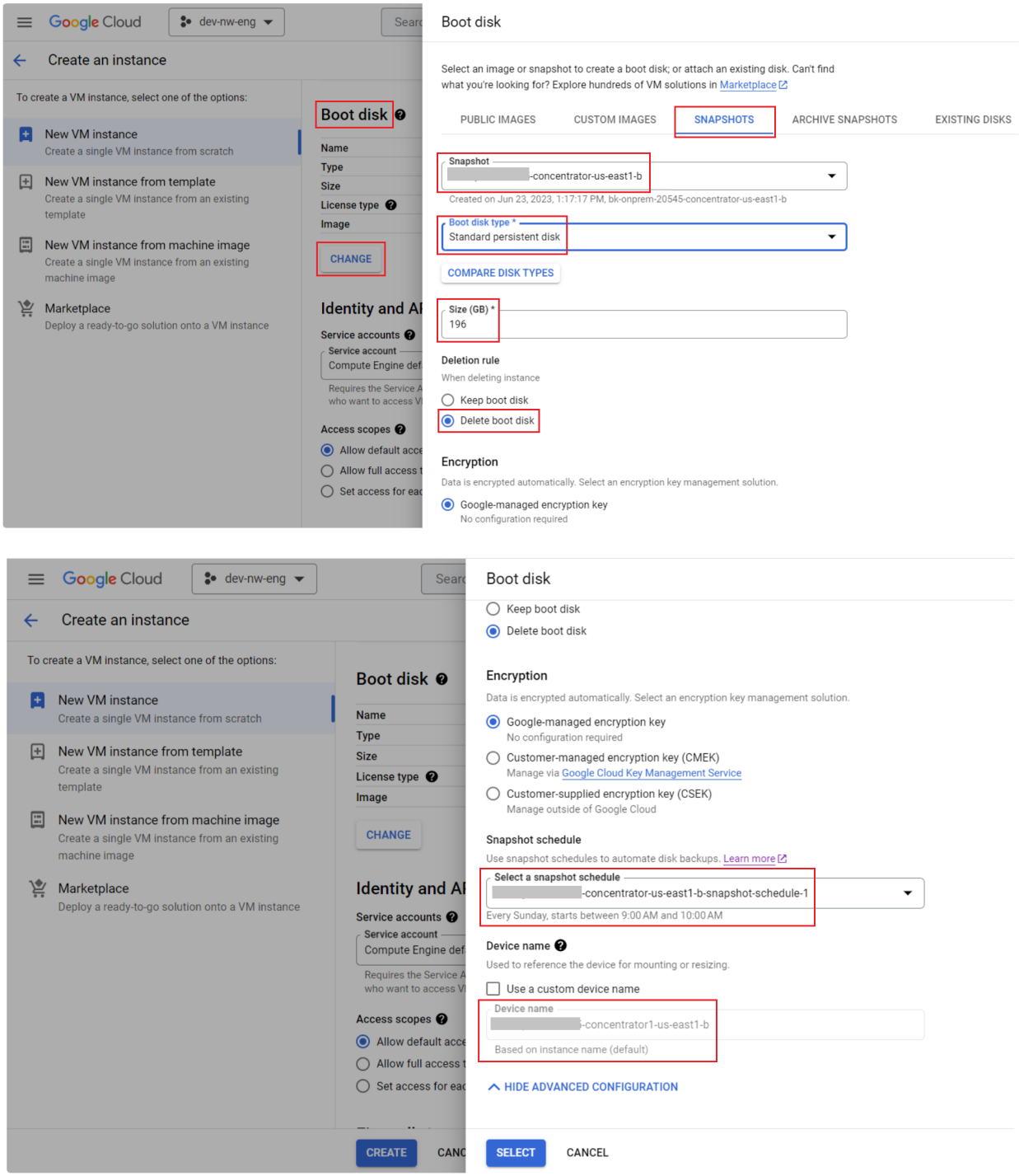
Detailed Steps

Step 1. Create a GCP VM instance from the snapshot with the same name and specifications

```
gcloud compute instances create 20545-concentrator1-us-east1-b
--project=dev-nw-eng
--zone=us-east1-b
--machine-type=n2-standard-4
--network-interface=network-tier=PREMIUM,stack-type=IPV4_ONLY,subnet=20545-ppn-node-subnetwork-us-east1
--maintenance-policy=MIGRATE
--provisioning-model=STANDARD
--service-account=97611879986-compute@developer.gserviceaccount.com
--scopes=https://www.googleapis.com/auth/devstorage.read_only,https://www.googleapis.com/auth/logging.write,https://www.googleapis.com/auth/monitoring.write,https://www.googleapis.com/auth/servicecontrol,https://www.googleapis.com/auth/service.management.readonly,https://www.googleapis.com/auth/trace.append
--create-disk=auto-delete=yes,boot=yes,device-name=20545-concentrator1-us-east1-b,disk-resource-policy=projects/dev-nw-eng/regions/us-east1/resourcePolicies/20545-concentrator-us-east1-b-snapshot-schedule-1,mode=rw,size=196,source-snapshot=projects/dev-nw-eng/global/snapshots/20545-concentrator-us-east1-b,type=projects/dev-nw-eng/zones/us-east1-b/diskTypes/pd-balanced
--labels=goog-ec-src=vm_add-gcloud
--reservation-affinity=any
```

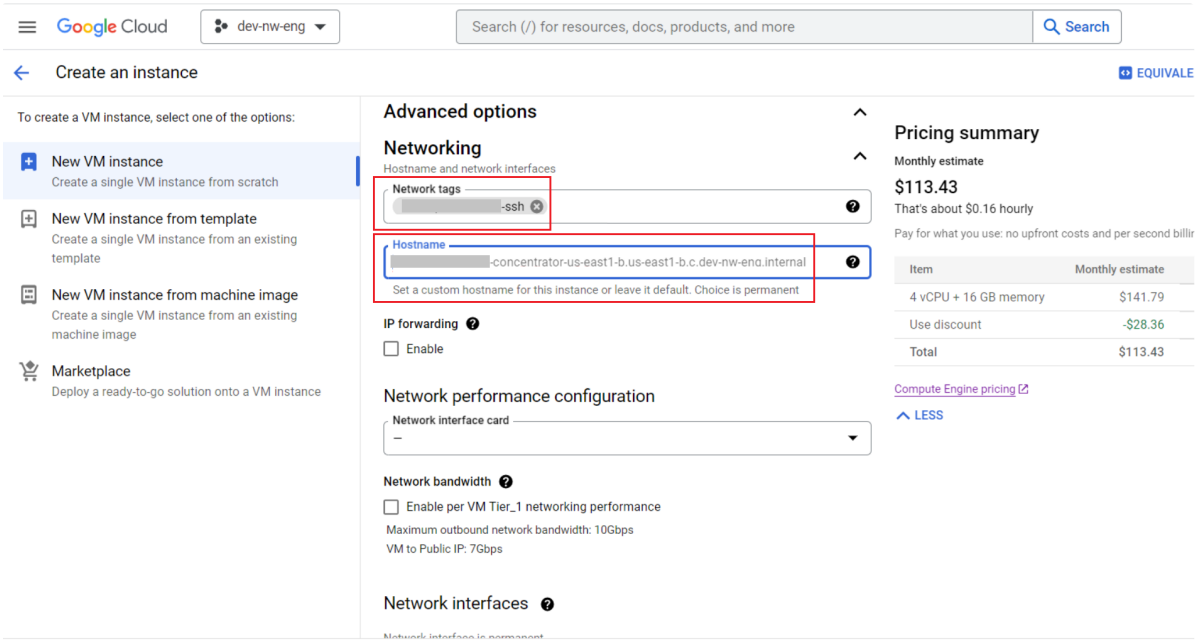
Step 2. Create a VM instance from the snapshot





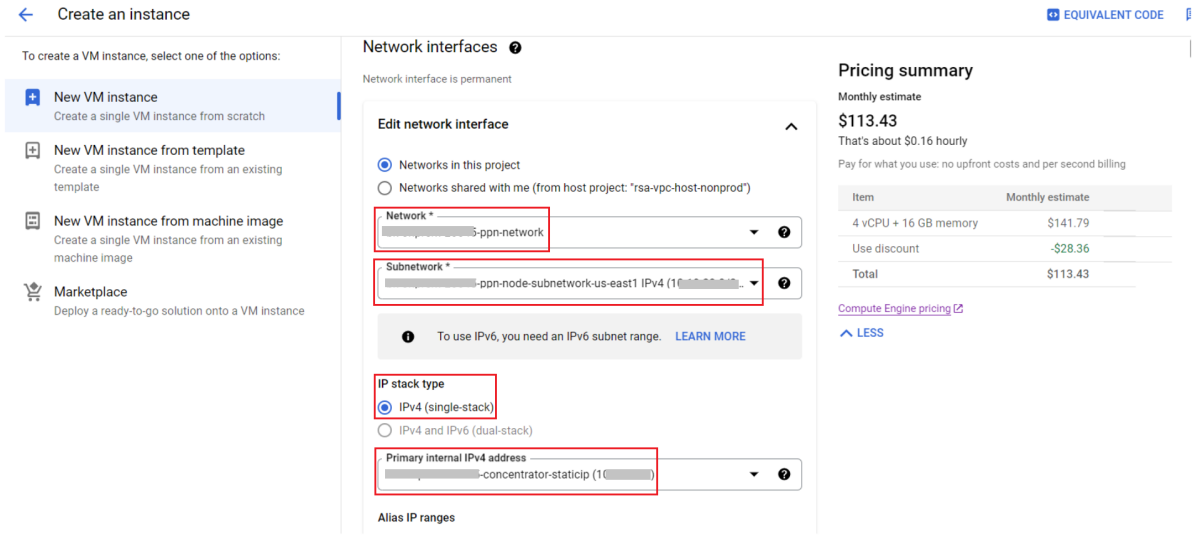
Step 3. Configure Networking

a. Select `Advanced Options > Networking > Select the required Network Tags`. Retain the default value for `HostName`.



b. Select `Network Interfaces > Edit network interface > Networks` in this project.

c. Select the appropriate `Network` and `Subnetwork` from the drop-down.



d. Select `IPv4 (single-stack)` and `Primary Internal IPv4 Address` from the drop-down.

> **Note:** This address MUST match the IPv4 address of the failed node being replaced.

Reserving Static Internal IPv4 Address

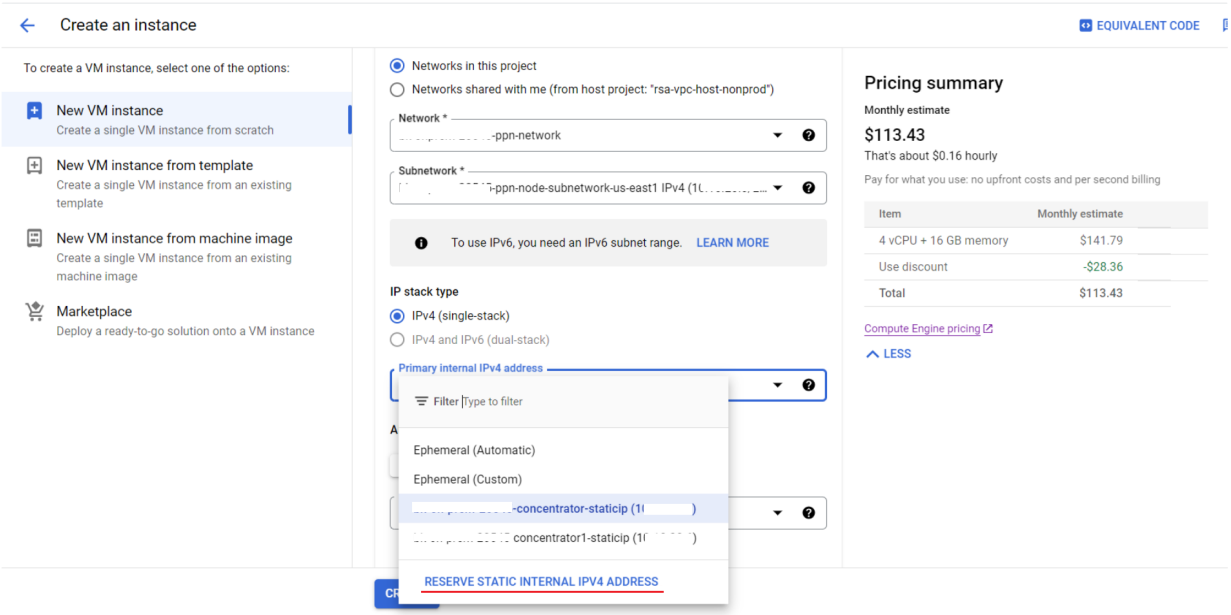
Select the `Primary internal IPv4 address` drop-down and select `RESERVE STATIC INTERNAL IPV4 ADDRESS`.

Configure static internal IP addresses | Compute Engine Documentation | Google Cloud

terra form: Configure static internal IP addresses | Compute Engine Documentation | Google Cloud

gcloud cli:

```
gcloud compute addresses create ADDRESS_NAME [ADDRESS_NAME..] \
    --region REGION --subnet SUBNETWORK \
    --addresses IP_ADDRESS
```



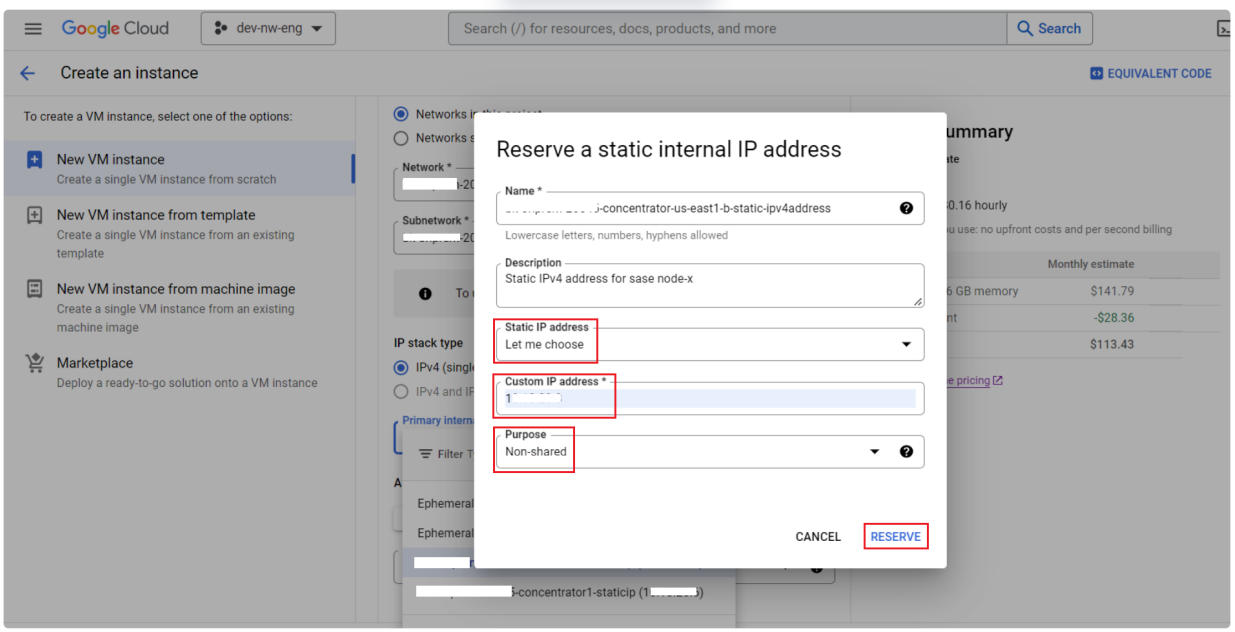
Enter the below details and click **Reserve**:

`Name` - lower case text to reflect the VM instance Name IPv4 address. For example: `20545-concentrator-us-east1-b-ipv4address`

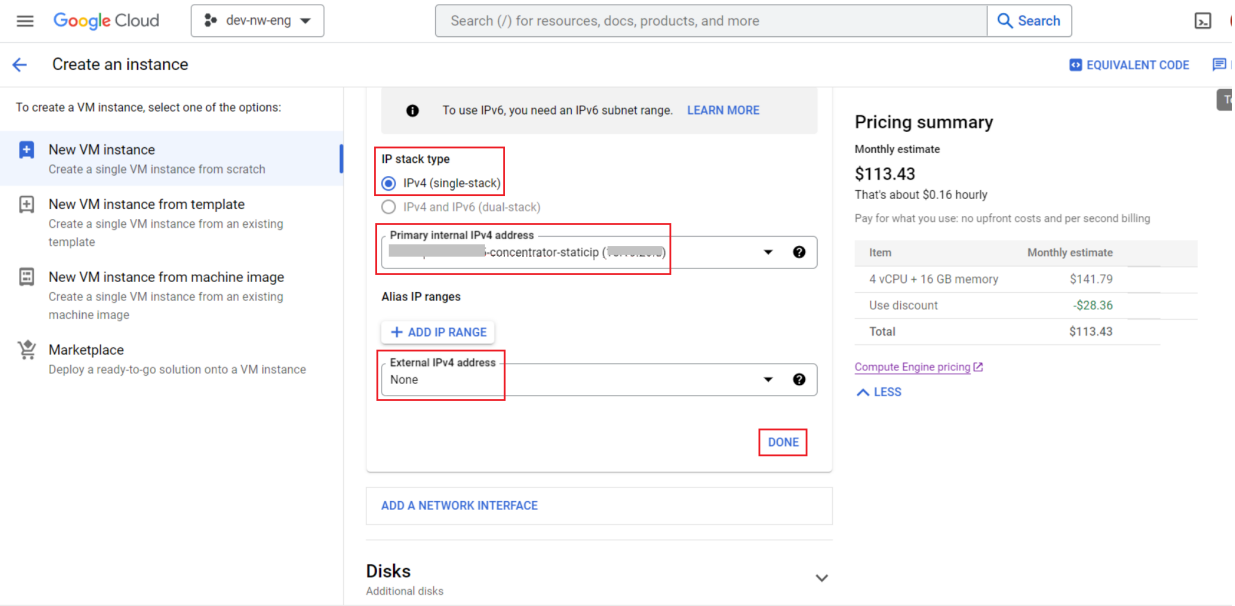
`Static IP address - Let me choose`

`Custom IP address` - IPv4 address in the `cloud_node_subnet` defined in `/opt/rsa/saTools/cloud/sase-deployment-models.yml`.

`Purpose - Non Shared`

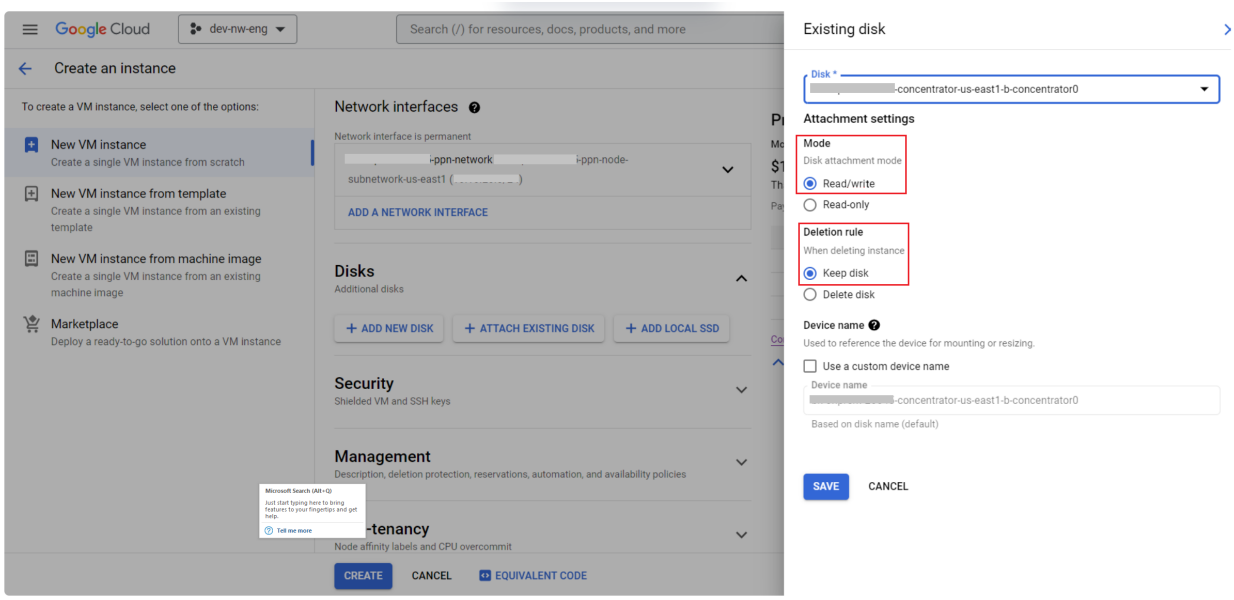


Under Alias IP ranges, select `None` for the External IPv4 address and click **Done**.

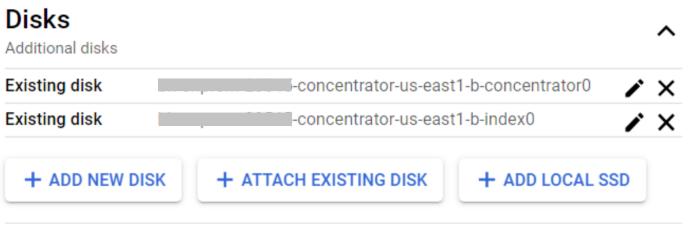


Step 4. Configure Storage Disks

Select `Disks > ATTACH EXISTING DISK`. Select the appropriate storage disk from the `Disk` drop-down. Select the `Read/Write` Mode and `Keep disk` Deletion Rule and `Save`. Repeat this (adding Disks) for all the remaining Storage disks for this instance.



Example: For concentrator service, at least two Storage disks are attached.



gcloud commands

Click **CREATE** to complete the VM instance creation from the snapshot.

Equivalent gcloud code to create a new VM instance with Storage disks attached from snapshot:

```
gcloud compute instances create 20545-concentrator-us-east1-b \

    --project=dev-nw-eng \

    --zone=us-east1-b \

    --machine-type=n2-standard-4 \

    --network-interface=private-network-ip=10.10.20.8,stack-type=IPV4_
    ONLY,subnet=20545-ppn-node-subnetwork-us-east1,no-address \

    --maintenance-policy=MIGRATE \

    --provisioning-model=STANDARD \

    --service-account=97611879986-compute@developer.gserviceaccount.com
    \

    --

scopes=https://www.googleapis.com/auth/devstorage.read_
only,https://www.googleapis.com/auth/logging.write,https://www.googleapis.com
/auth/monitoring.write,https://www.googleapis.com/auth/servicecontrol,https:/
/www.googleapis.com/auth/service.management.readonly,https://www.googleapis.c
om/auth/trace.append \
```

```
--tags=20545-ssh \

--create-disk=auto-delete=yes,boot=yes,device-name=20545-
concentrator-us-east1-b,mode=rw,size=196,source-
snapshot=projects/dev-nw-eng/global/snapshots/20545-concentrator-us-
east1-b,type=projects/dev-nw-eng/zones/us-east1-b/diskTypes/pd-
standard \

--disk=boot=no,device-name=20545-concentrator-us-east1-b-
concentrator0,mode=rw,name=20545-concentrator-us-east1-b-
concentrator0 \

--disk=boot=no,device-name=20545-concentrator-us-east1-b-
index0,mode=rw,name=20545-concentrator-us-east1-b-index0 \

--labels=goog-ec-src=vm_add-gcloud \

--reservation-affinity=any
```

Terraform

```
# This code is compatible with Terraform 4.25.0 and versions that are
backwards compatible to 4.25.0.

# For information about validating this Terraform code, see
https://developer.hashicorp.com/terraform/tutorials/gcp-get-started/google-
cloud-platform-build#format-and-validate-the-configuration

resource "google_compute_instance" "20545-concentrator-us-east1-b" {

     attached_disk {

          device_name = "20545-concentrator-us-east1-b-
          concentrator0"

          mode = "READ_WRITE"

          source = "projects/dev-nw-eng/zones/us-east1-
          b/disks/20545-concentrator-us-east1-b-concentrator0"

     }

     attached_disk {

          device_name = "20545-concentrator-us-east1-b-index0"

          mode = "READ_WRITE"

          source = "projects/dev-nw-eng/zones/us-east1-
          b/disks/20545-concentrator-us-east1-b-index0"

     }

     boot_disk {

          auto_delete = true

          device_name = "20545-concentrator-us-east1-b"

          initialize_params {

          size = 196

          type = "pd-standard"

          }

          mode = "READ_WRITE"
```

```
    }
    can_ip_forward = false
    deletion_protection = false
    enable_display = false
    labels = {
          goog-ec-src = "vm_add-tf"
    }
    machine_type = "n2-standard-4"
    name = "20545-concentrator-us-east1-b"
    network_interface {
         network_ip = "10.10.20.8"
         subnetwork = "projects/dev-nw-eng/regions/us-
         east1/subnetworks/20545-ppn-node-subnetwork-us-east1"
    }
    scheduling {
         automatic_restart = true
         on_host_maintenance = "MIGRATE"
         preemptible = false
         provisioning_model = "STANDARD"
    }
    service_account {
         email = "97611879986-
         compute@developer.gserviceaccount.com"
         scopes =
         ["https://www.googleapis.com/auth/devstorage.read_only",
         "https://www.googleapis.com/auth/logging.write",
         "https://www.googleapis.com/auth/monitoring.write",
         "https://www.googleapis.com/auth/service.management.reado
         nly", "https://www.googleapis.com/auth/servicecontrol",
         "https://www.googleapis.com/auth/trace.append"]
    }
    tags = ["20545-ssh"]
    zone = "us-east1-b"
}
```

Troubleshooting NetWitness SASE Deployment

The following sections outline potential errors that may arise during the deployment of SASE. Each section includes example outputs of these errors. It is essential to consult the relevant installation and deployment steps for additional details before attempting to resolve the issues.

Missing cloud credentials

Output example

nw-create-cloud-hybrid was aborted since the token file was not found in its default location.

```
[root@js116-adminserver errorCapture]# nw-create-cloud-hybrid --enable-cloud-sase --cloud-provider gcp --namespace js116
[2024-10-29T15:44:17+00:00] <16789> (INFO) Creating /var/log/netwitness/sase directory.
[2024-10-29T15:44:25+00:00] <16789> (INFO) Using cloud provider: gcp
[2024-10-29T15:44:25+00:00] <16789> (ERROR) Please specify the gcp Service Account Key Token file.
[root@js116-adminserver errorCapture]#
```

Solution:

- Generate the cloud credentials.
- Issue appropriate permissions to the identity of the credentials.
- Install the credentials on the Administration server.
  - For GCP:
    - Place the credentials in `/root/.gcp/gcp-auth-token.json`
    - Ensure no world or group read on the token: `chmod 600/root/.gcp/gcp-auth-token.json`

Invalid cloud credentials

Output example

This is an example of an initial attempt to deploy SASE with a malformed or bad token.

```
[2024-10-29T15:44:25+00:00] <16789> (INFO) Using cloud provider: gcp
[2024-10-29T15:44:25+00:00] <16789> (ERROR) Please specify the gcp Service Account Key Token file.
[2024-10-29T15:49:53+00:00] <19177> (ERROR) Failed to get attribute: hybrid_deployment from Ohai node: nw_host
[2024-10-29T15:51:18+00:00] <19670> (INFO) Using cloud provider: gcp
[2024-10-29T15:51:18+00:00] <19670> (INFO) Template File: /root/.sase/sase-deployment-models.yml is properly formatted.
```

```
[2024-10-29T15:51:19+00:00] <19670> (INFO) Template File: /root/.sase/host-
models.yml is properly formatted.
[2024-10-29T15:51:45+00:00] <19670> (ERROR) Unable to set the gcp project for
the gcloud cli
[2024-10-29T15:53:39+00:00] <21137> (INFO) Using cloud provider: gcp
[2024-10-29T15:53:40+00:00] <21137> (INFO) Template File: /root/.sase/sase-
deployment-models.yml is properly formatted.
[2024-10-29T15:53:40+00:00] <21137> (INFO) Template File: /root/.sase/host-
models.yml is properly formatted.
[2024-10-29T15:53:40+00:00] <21137> (ERROR) Unable to set the gcp project for
the gcloud cli
[2024-10-29T15:56:58+00:00] <79418> (INFO) Using cloud provider: gcp
[2024-10-29T15:56:58+00:00] <79418> (INFO) Template File: /root/.sase/sase-
deployment-models.yml is properly formatted.
[2024-10-29T15:56:58+00:00] <79418> (INFO) Template File: /root/.sase/host-
models.yml is properly formatted.
[2024-10-29T15:56:59+00:00] <79418> (ERROR) Unable to set the gcp project for
the gcloud cli
```

Subsequent attempts that use a malformed or bad token display this behavior.

```
nw-create-cloud-hybrid --enable-cloud-sase --cloud-provider gcp --namespace
js116
[2024-10-29T15:56:58+00:00] <79418> (INFO) Using cloud provider: gcp
[2024-10-29T15:56:58+00:00] <79418> (INFO) Template File: /root/.sase/sase-
deployment-models.yml is properly formatted.
[2024-10-29T15:56:58+00:00] <79418> (INFO) Template File: /root/.sase/host-
models.yml is properly formatted.
parse error: Invalid literal at line 2, column 0
[2024-10-29T15:56:59+00:00] <79418> (ERROR) Unable to set the gcp project for
the gcloud cli
[root@js116-adminserver ~]#
```

Solution

- For GCP
  - Check with your cloud administrator to ensure the token was correctly generated and transmitted to you.
  - Install a corrected token.
  - Ensure no world or group read on the token: `chmod 600/root/.gcp/gcp-auth-token.json`

Improperly formed sase-deployment-models.yml file

Output example

An example syntax error (a missing single quote) in the models file results in the below error.

```
vpn_provider: 'Netskope
```

The output resulting from an error in the models file.

```
...

File "/usr/lib64/python3.6/site-packages/yaml/parser.py", line 439, in parse_
block_mapping_key
"expected <block end>, but found %r" % token.id, token.start_mark)
 yaml.parser.ParserError: while parsing a block mapping
  in "/root/.sase/sase-deployment-models.yml", line 18, column 7
expected <block end>, but found '<scalar>'
  in "/root/.sase/sase-deployment-models.yml", line 61, column 23
[2024-10-29T16:11:44+00:00] <84831> (ERROR) Unable to parse yaml to json
[root@js116-adminserver ~]#
```

Solution

- Correct the structure of the model file found in `/root/.sase/sase-deployment-models.yml`.
- Ensure any host file key references in the model file are found in the host file at `/root/.sase/host-models.yml`, and correct any discrepancies.

Improperly formed host-models.yml file

Output example

A missing colon in the disk name raises the below error.

```
 decodersmall:
 disk_name decodersmall
 disk_type: pd-standard
 disk_size: 690
```

An incorrect host-models file will result in the error below.

```
 ...

 File "/usr/lib64/python3.6/site-packages/yaml/scanner.py", line 116, in
check_token
self.fetch_more_tokens()
  File "/usr/lib64/python3.6/site-packages/yaml/scanner.py", line 220, in
fetch_more_tokens
return self.fetch_value()
  File "/usr/lib64/python3.6/site-packages/yaml/scanner.py", line 576, in
fetch_value
self.get_mark())
yaml.scanner.ScannerError: mapping values are not allowed here
in "/root/.sase/host-models.yml", line 13, column 20
[2024-10-29T17:03:28+00:00] <101933> (ERROR) Unable to parse yaml to json
```

Solution

- Correct the structure of the `/root/.sase/host-models.yml` file.
- Ensure any host file key references that are used within the model file exist in the hosts file and are properly referenced.

Missing image file (lite)

Output example

An example of the error message displayed when the image file is missing. The ‘lite’ image is used to create the specified nodes.

```
...


Plan: 6 to add, 0 to change, 0 to destroy.


Changes to Outputs:
+ network = "js116-ppn-network"
+ ppn_server_ext_ip = (known after apply)

╷
│ Error: error retrieving image information: googleapi: Error 404: The
resource 'projects/nw-nwp-dev/global/images/rsa-nw-12-5-1-0-21738-lite' was
not found, notFound
│
│ with data.google_compute_image.nw_image,
│ on compute.tf line 12, in data "google_compute_image" "nw_image":
│ 12: data "google_compute_image" "nw_image" {
│
╵

[2024-10-29T17:08:35+00:00] <104309> (ERROR) Deployment of js116-ppn-server
failed
[2024-10-29T17:08:35+00:00] <103741> (ERROR) Failed to create ppn-server in
region us-east1
[root@js116-adminserver ~]#
```

Solution

- Ensure that a lite image exists in your image repository.
- Contact support to ensure that the correct image has been copied to your project's cloud image repository.
- Contact your Cloud administrator to ensure the identity of the token has rights to use the image.

Failure of nw-create-cloud-hybrid --disable-cloud-sase

Output example

In this example, a simulated failure was created by terminating the the ssh connection to the adminserver while nw-create-cloud-hybrid --disable-cloud-sase was running. In this situation running the command again would see the already removed assets and continue on successfully. However, for resons below, a failure could occur, leaving assets unremoved.

A simulated network failure occurred while a subnet was being removed.

```
[2024-10-29T20:08:05+00:00] <193469> (ERROR) Undeployment of subnet failed
[2024-10-29T20:08:05+00:00] <187807> (ERROR) Failed to delete subnet
10.10.20.0/24 in region us-east1
```

Solution

- Depending upon the point of failure of `--disable-cloud-sase`, various cloud assets may be left unremoved.
- Possible cause:
  - Network disruption.
    - Attempt to run the `--disable-cloud-sase` again. If still failing, see below for manual removal of cloud assets.
  - Deletion of or removal of access to state data from the state data bucket.
    - See below for manual removal of cloud assets.
  - Removal of rights from the service account.
    - restore the rights, re-run, and then remove the rights once all cloud assets are removed.
- Manual removal of cloud assets should repeated runs of `--disable-cloud-sase` fail.
  - The following is list of assets that might be left unremoved. Using your projects cloud console, carefully remove the cloud assets. The asset names will be prefixed with a namespace “nw”. This helps identify cloud assets created by sase automation. Depending upon your model file, you may have assets in multiple regions. Ensure you check all regions deployed to for assurance that all assets are removed. If unsure, check your model file to see where your hosts were configured to be deployed.
    - For GCP, if exists, delete:
      - the Warm Storage Buckets (Cloud Storage / Buckets).
      - the Compute Instances (Compute Engine / VM Instances).
      - the Cloud router, and thus the cloud nat. (search for: Cloud Routers in the navigation search bar).
      - the VPC, and thus the subnets.
      - the static IP addresses (VPC Network / IP addresses).
      - the service account.

Firewall Rule(s) to allow UDP 4242 egress from adminserver to ppn-server not present or are malformed

Output Example

```
nw-create-cloud-hybrid --enable-cloud-sase
```

… (truncated)

The error output seen when the adminserver cannot reach the ppn-server.

```
Waiting for nebula service to start. in 5 seconds.....[2024-11-
05T20:08:53+00:00] <95647> (INFO) nebula service is running
[2024-11-05T20:08:53+00:00] <96262> (INFO) nebula service is running
[2024-11-05T20:08:53+00:00] <96262> (INFO) Successfully connected to
172.30.30.2 on port 22.
[2024-11-05T20:08:58+00:00] <96262> (ERROR) Unable to connect to 172.30.30.1
on ssh port
[2024-11-05T20:08:58+00:00] <93656> (ERROR) Failed to validate NetWitness
Overlay Network connections
[root@js116-adminserver ~]#
```

Solution

- Correct the firewall settings to allow egress of UDP 4242 from adminserver to the ppn server.

Firewall Rule(s) to allow TCP 443 egress to cloud api endpoints are not present or are malformed

Output Example

```
nw-create-cloud-hybrid --enable-cloud-sase
```

… (truncated)

The output appears frozen when TCP 443 egress is not allowed from the adminserver. Sometimes this occurs at the line `Template File: /root/.sase/host-models.yml is properly formatted.`

```
[root@js116-adminserver ~]# nw-create-cloud-hybrid --enable-cloud-sase --
cloud-provider gcp --namespace js116
[2024-11-05T20:30:28+00:00] <106035> (INFO) Using cloud provider: gcp
[2024-11-05T20:30:28+00:00] <106035> (INFO) Template File: /root/.sase/sase-
deployment-models.yml is properly formatted.
[2024-11-05T20:30:28+00:00] <106035> (INFO) Template File: /root/.sase/host-
models.yml is properly formatted.
[2024-11-05T20:30:29+00:00] <106412> (INFO) ssh key-pair is already created,
skipping..
```

```
[2024-11-05T20:30:29+00:00] <106412> (INFO) terraform rpm already installed,
skipping...
[2024-11-05T20:30:30+00:00] <106412> (INFO) google-cloud-cli rpm already
installed, skipping...
[2024-11-05T20:30:31+00:00] <106412> (INFO) Installing package: nebula
```

Solution

- Correct the firewall settings to allow egress of TCP 443 from adminserver to the cloud api endpoints.

Insufficient permissions/roles on the cloud service account

Output

... (truncated)

```
google_compute_subnetwork.nw_ppn_server_subnetwork: Creating...
google_compute_firewall.nw_ppn_ingress: Creating...
google_compute_firewall.nw_ssh: Creating...
google_compute_firewall.nw_ppn_egress: Creating...
google_compute_firewall.nw_ppn_ingress: Still creating... [10s elapsed]
google_compute_subnetwork.nw_ppn_server_subnetwork: Still creating... [10s
elapsed]
google_compute_firewall.nw_ssh: Still creating... [10s elapsed]
google_compute_firewall.nw_ppn_egress: Still creating... [10s elapsed]

╷
│ Error: Failed to save state
│
│ Error saving state: Failed to upload state to
│ gs://nw-cloud-artifacts-1c1013f4/terraform/js116-ppn-server/default.tfstate:
googleapi: Error 403:
│ nw-sase-automation@nw-nwp-dev.iam.gserviceaccount.com does not have
storage.objects.create access to the Google
│ Cloud Storage object. Permission 'storage.objects.create' denied on resource
(or it may not exist)., forbidden
╵

╷
│ Error: Failed to persist state to backend
│
│ The error shown above has prevented Terraform from writing the updated state
to the configured backend. To
│ allow for recovery, the state has been written to the file "errored.tfstate"
in the current working directory.
│
│ Running "terraform apply" again at this point will create a forked state,
making it harder to recover.
│
```

```
| To retry writing this state, use the following command:
|  terraform state push errored.tfstate
|
'

'
| Error: Error waiting to create Subnetwork: Error waiting for Creating
Subnetwork: error while retrieving operation:
googleapi: Error 403: Required 'compute.regionOperations.get' permission for
'projects/nw-nwp-dev/regions/us-east1/operations/operation-1730910031081-
62640e58a6e7f-3b2b21c4-c8362697', forbidden
|
| with google_compute_subnetwork.nw_ppn_server_subnetwork,
| on network.tf line 11, in resource "google_compute_subnetwork" "nw_ppn_
server_subnetwork":
| 11: resource "google_compute_subnetwork" "nw_ppn_server_subnetwork" {
|
'

'
| Error: Error waiting to create Firewall: Error waiting for Creating
Firewall: error while retrieving operation: googleapi: Error 403: Required
'compute.globalOperations.get' permission for 'projects/nw-nwp-
dev/global/operations/operation-1730910031083-62640e58a78a8-1055e424-
a8fbfd4a', forbidden
|
| with google_compute_firewall.nw_ssh,
| on network.tf line 21, in resource "google_compute_firewall" "nw_ssh":
| 21: resource "google_compute_firewall" "nw_ssh" {
|
'

'
| Error: Error waiting to create Firewall: Error waiting for Creating
Firewall: error while retrieving operation: googleapi: Error 403: Required
'compute.globalOperations.get' permission for 'projects/nw-nwp-
dev/global/operations/operation-1730910031083-62640e58a78ad-e0ed2c5a-
83b29a5a', forbidden
|
| with google_compute_firewall.nw_ppn_ingress,
| on network.tf line 36, in resource "google_compute_firewall" "nw_ppn_
ingress":
| 36: resource "google_compute_firewall" "nw_ppn_ingress" {
|
'

'
```

```
| Error: Error waiting to create Firewall: Error waiting for Creating
Firewall: error while retrieving operation: googleapi: Error 403: Required
'compute.globalOperations.get' permission for 'projects/nw-nwp-
dev/global/operations/operation-1730910031084-62640e58a7b66-08f858df-
24b6bfc1', forbidden
|
| with google_compute_firewall.nw_ppn_egress,
| on network.tf line 51, in resource "google_compute_firewall" "nw_ppn_
egress":
| 51: resource "google_compute_firewall" "nw_ppn_egress" {
|
'

Releasing state lock. This may take a few moments...

'
| Error: Error releasing the state lock
|
| Error message: 2 errors occurred:
| * googleapi: Error 403: nw-sase-automation@nw-nwp-
dev.iam.gserviceaccount.com does not have
| storage.objects.delete access to the Google Cloud Storage object. Permission
'storage.objects.delete' denied on
| resource (or it may not exist)., forbidden
| * googleapi: got HTTP response code 403 with body: <?xml version='1.0'
| encoding='UTF-8'?><Error><Code>AccessDenied</Code><Message>Access
| denied.</Message><Details>nw-sase-automation@nw-nwp-
dev.iam.gserviceaccount.com does not have
| storage.objects.get access to the Google Cloud Storage object. Permission
'storage.objects.get' denied on
| resource (or it may not exist).</Details></Error>
|
|
|
| Terraform acquires a lock when accessing your state to prevent others
| running Terraform to potentially modify the state at the same time. An
| error occurred while releasing this lock. This could mean that the lock
| did or did not release properly. If the lock didn't release properly,
| Terraform may not be able to run future commands since it'll appear as if
| the lock is held.
|
| In this scenario, please call the "force-unlock" command to unlock the
| state manually. This is a very dangerous operation since if it is done
| erroneously it could result in two people modifying state at the same time.
| Only call this command if you're certain that the unlock above failed and
| that no one else is holding a lock.
```

```
[2024-11-06T16:20:42+00:00] <19843> (ERROR) Deployment of js116-ppn-server
failed
[2024-11-06T16:20:42+00:00] <19348> (ERROR) Failed to create ppn-server in
region us-east1
[root@js116-adminserver ~]#
```

Shows insufficient permission by the service account, in particular, storage (bucket) permissions are missing: '`does not have storage.objects.create`'.

Solution

- For GCP
  - Ensure all permissions for the service account have been created and rerun.