NetWitness® Platform

Version 12.5.2

Netskope SASE Configuration 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

License Agreement

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 NetWitness. 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

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/.

January, 2026

Contents

Getting Started

NetWitness SASE, combined with Netskope, provides unprecedented visibility into behavior and communication among devices and services in remote and distributed networks across on-premises, hybrid, and cloud deployments.

What NetWitness SASE does:

- **Streamline searches and investigations**: Log into a single user interface to perform index searches, pivot through metadata, and reconstruct network sessions to receive results quickly.
- **Leverage retained data**: Empower analysts to perform forensic examinations on a triggered detection and threat hunt for unknown threats against retained raw network communications.
- **Correlate disparate data sets**: Enrich the context of investigations by correlating data from the actual network traffic of remote users with other access by those same users for a complete end-to-end story of what transpired.
- **Minimize costs**: Optimize storage and reduce operating costs using new compression algorithms, selective retention, and the ability to split network decoder components to limit what must run in the cloud.

About NetWitness SASE

NetWitness supports SASE and critical hybrid use cases across on-premises and in the cloud by partnering with Netskope on technical integrations. NetWitness SASE Integrations give organizations complete visibility into encrypted traffic, remote users, and cloud workloads. With NetWitness SASE integrations, customers can achieve SASE flexibility, inherent security advantages, and complete visibility into threat detection and response.

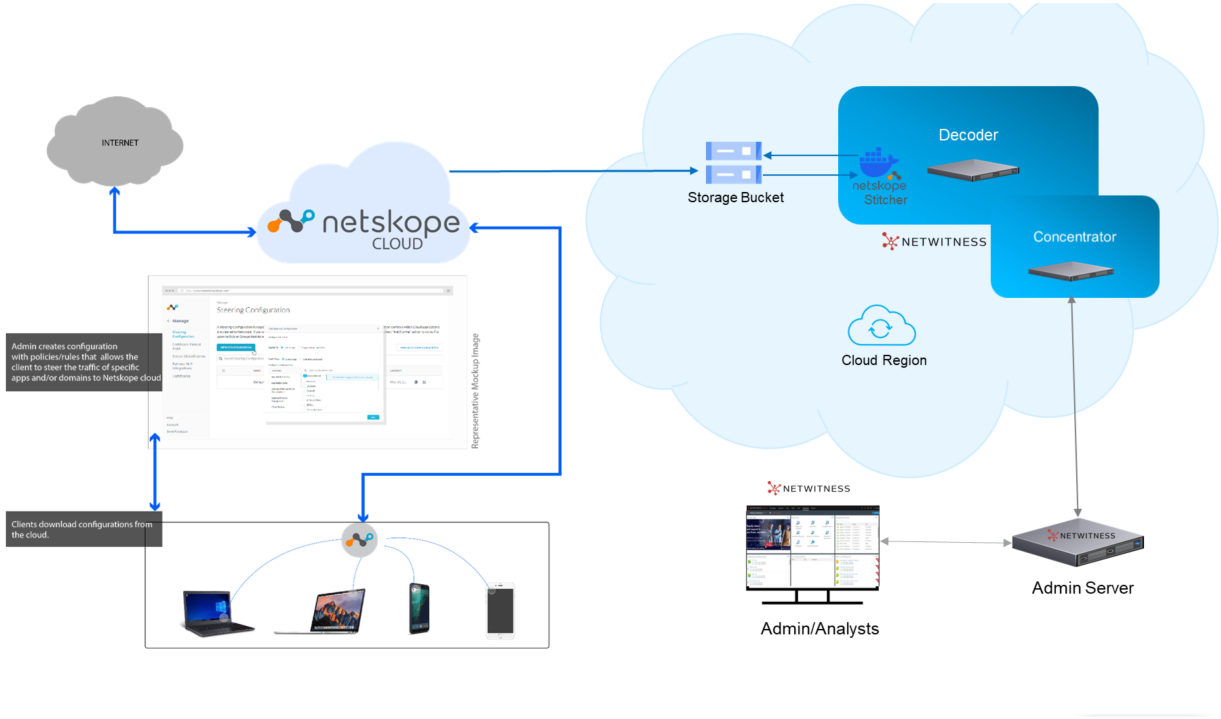
NetWitness SASE provides the following capabilities:

- **Flexible**, **secure**, **real-time traffic monitoring**: NetWitness SASE integrations capture all network traffic from remote users in near real-time, enabling immediate response to any potential threats. Regardless of the location of the data collected, the data is available in the detection engine, and analysts can easily find the anomalies. The customization opens available in NetWitness SASE reduce the risk of storing sensitive, personally identifiable information.
- **Get scalable**, **high-performance cloud security**: With NetWitness SASE integrations, enhance total visibility and threat detection capabilities across your enterprise using well-known on-premises mechanisms such as rules, parsers, feeds, and machine learning. Perform searches and investigations and swiftly receive results with a single user interface. The integration supports forensic examinations on triggered detections and facilitates threat hunting against retained network communications, empowering analysts to combat unknown threats effectively.
- **Eliminate blind spots**: NetWitness SASE integrations empower organizations to retain complete visibility into their cloud security stack, cost-effectively eliminating blind spots in their cloud traffic and maximizing the effectiveness of their security infrastructure investments. Organizations have the visibility and control they need over encrypted traffic to ensure compliance with their privacy, regulatory, and acceptable use policies, whether on-premises or in the cloud.

- **Unparalleled network visibility to strengthen SASE security**: The improved visibility provided by the integration allows organizations to close gaps in their zero trust security posture and enable better detection capabilities.

Deployment Diagram

The following diagram depicts a deployment model for NetWitness Netskope SASE Integration.

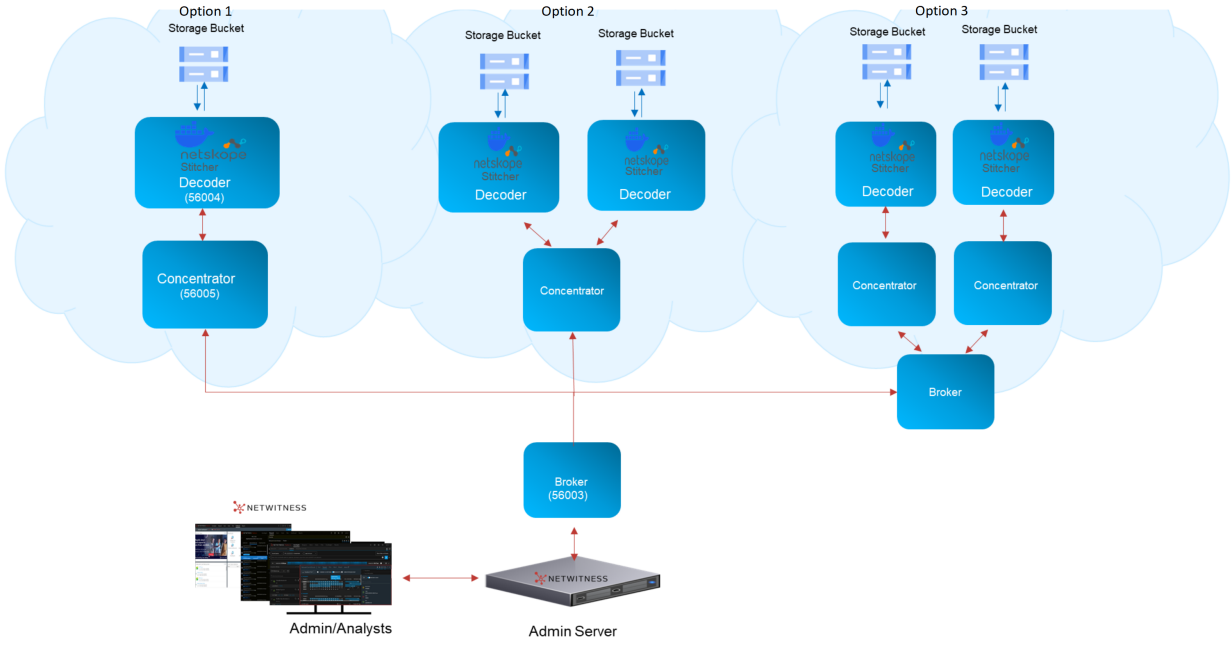


- The first half of the model shows the Netskope deployment and configurations. The Netskope Administrator creates configurations with policies and rules that allows the client to steer the traffic of specific applications and domains to Netskope cloud and storage bucket. The Netskope clients download the configurations and steer the traffic to configured Storage bucket through Netskope cloud.
- The second half of the model shows the NetWitness deployment , components and configurations.
- The NetWitness `Decoder` cloud instance supports traffic capture from the `Storage Bucket`, parses the traffic and generates meta which is used for threat detections and investigations.
  - The Decoder runs Netskope `Cloud-Tap Stitcher container` which connects to the configured `Storage Bucket`, captures the traffic and replays it on a local network interface of the instance.
  - The Decoder then captures the traffic from that local network interface and generates meta.
  - The Decoder uses the default decryption provided by `Cloud-Tap Stitcher` for the encrypted traffic.

- The NetWitness `Concentrator` cloud instance aggregates the meta from `Decoder` and indexes the meta to support querying and investigations.
- It is recommended to be deploy the `Decoder` and `Concentrator` instances in the same cloud provider region of the `Storage Bucket`. This would minimize latency and cloud transfer costs of the traffic data.
- The NetWitness `Admin Server` supports configuring and managing the `Decoder` and `Concentrator` instances. It also provides centralized platform for Analysts in threat detections, investigations and response.
- The `Admin Server` can be deployed in the cloud or on-premise. The hybrid-cloud deployment can connect the on-prem installed `Admin Server` and the cloud instances of `Decoder` and `Concentrator`.

Scaling Options

The following diagram shows various deployment scaling options with NetWitness Components.



- The NetWitness `Broker` is a service that aggregates database ranges from the configured devices (Concentrators) and provides an API to treat multiple devices as a single query-able entity.
- The `Broker` service can be run as part of Admin Server and also as a standalone instance for additional scaling. It can be deployed in the cloud or on-premise.

Following are the few sample deployment options and topology of NetWitness components:

- **Option 1**: A `Concentrator` aggregates from a single `Decoder` in the cloud, and that `Concentrator` can be configured to a `Broker` .

- **Option 2**: A `Concentrator` aggregates from multiple `Decoders` in the cloud, and that `Concentrator` can be configured to a `Broker`.
- **Option 3**: A `Broker` aggregates from multiple `Concentrators` in the cloud, and those `Concentrators` aggregate from their paired `Decoders`. This `Broker` can be configured to a Top Level `Broker`.
- Investigating and Querying the Top Level Broker would run queries against its configured devices (`Concentrators`, `Brokers`) and return the results which can be visualized in the `Admin Server` for the entire topology that was queried.
- The hybrid-cloud deployment can connect the on-prem `Admin Server` and `Broker` to the cloud instances of `Decoders`, `Concentrators` and `Brokers`.
- The Decoders use the Cloud-Tap Stitcher multi threading option for scaling vertically in the capture from Storage Bucket.
- The Decoder supports the Cloud-Tap Stitcher Origin Filter (Netskope POP, where POP is Point of Presence) option to scale horizontally on the same Storage Bucket.

Port Information

There are several ports involved in the NetWitness SASE deployment. These default ports are used for the components described above.

Note: If your SASE deployment utilizes an Overlay network, the ports are opened for internal networks as necessary. For external networks, the Overlay network provides encapsulation.

For trusted communication, the services listen on following tcp ports:

- Decoder 56004, Concentrator 56005, Broker 56003
- Appliance 56006
  - It is a service that run as part of every NetWitness instance for Appliance configurations, storage configurations and instance statistics.

These ports are required to be open and available in the network.

For non trusted communication, the services listen on following tcp ports:

- Decoder 50004, Concentrator 50005, Broker 50003
- Appliance 50006

These ports are optional to be open and available in the network.

For secure RESTful communication, the services listen on following tcp ports:

- Decoder 50104, Concentrator 50105, Broker 50103
- Appliance 50106

These ports are optional to be open and available in the network. It is recommended to provide a way to reach these ports when needed as RESTful services help advanced troubleshooting and accessing diagnostic tools.

For trusted AMQP (Advanced Message Queuing Protocol) connections, the services use tcp ports 15671. These ports are required to be open and available in the network.

For more more information on open ports for SASE deployment, refer to Network Architecture and Ports.

Limitations

- Running multiple Cloud-Tap Stitcher containers on a single Decoder instance is not supported. Each Decoder supports running only one Cloud-Tap Stitcher container.

Configure Netskope Integration

Note:
- If you have not deployed NW instances in the GCP cloud, refer to the document https://community.netwitness.com/s/article/DeploymentGuidefor12-5-1?tabset-87e51=2 and for SASE deployment, refer to https://community.netwitness.com/s/article/SASEConfiguration.
- For GCP Instance configuration, refer to https://community.netwitness.com/s/article/GCPInstanceConfigurationRecommendations

Note:
- If you have not deployed NW instances in the AWS cloud, refer to the document https://community.netwitness.com/s/article/AWSInstallationMarketplaceGuidefor12-5
- For AWS Instance configurations, refer to https://community.netwitness.com/s/article/InstanceConfigurationRecommendations.

Note: To prepare cloud storage, refer to https://community.netwitness.com/s/article/PrepareVirtualorCloudStorage.

There are two methods to configure Netskope Integration from NetWitness Platform.

Note: NetWitness recommends you to use the Centralized Content Management (CCM) method for a more streamlined deployment process.

- Deploy Netskope Integration using CCM
- Deploy Netskope Integration Using NwConsole

Deploy Netskope Integration using CCM

This topic describes how to deploy the Netskope Integration for users using the Policy based CCM.

Prerequisites

Before proceeding, it is important to make sure the following:

- The NetWitness Platform (Admin Server and Packet Decoder Host) is on version 12.5.1 or later.
- You are connected to Live Services under the **(Admin)** > **System** > **Live Services** page.
- Ensure you have a network connection between the Decoder and Google Cloud Platform (GCP) or Amazon Web Services (AWS).
- The Decoder services are managed by CCM. If CCM does not manage it, you can enable CCM for the particular decoder service. For more information, see the topic Enable or Disable CCM for Individual Decoder Services.
- You must have the GCP or AWS bucket names available for configuration. Bucket Authentication (.JSON file) is optional for GCP and AWS:
  - The Bucket Authentication Key (.JSON file) is used to authenticate access to a bucket in GCP or AWS. Creating a Bucket Authentication Key (.JSON file) is a two-step process:
    - Create a service account in GCP with the role **Storage Object Viewer (roles/storage.objectViewer)**. For more information, see the topic Create service accounts.
    - Create a service account key in GCP. For more information, see the topic Create and delete service account keys.
- In case of AWS, you must have an IAM user with role assigned having below access permissions to the bucket.

  ```
  "Action": [ "s3:GetObject", "s3:ListBucket ]
  "Resource": ["arn:aws:s3:::<bucket-name>", "arn:aws:s3:::<bucket-name>/*"]
  ```

To create the .json file

- Access IAM user and obtain the following keys.
  - **Access Key**
  - **Secret Key**
- Create the **Auth Key JSON file**

  For Example: `AWS-Bucket-Auth-Key.json`
- Update the keys as shown below.

  ```
  {
  "access_key_id": "<copy access key>",
  ```

```
"secret_access_key": "<copy secret key>"
}
```

You must perform the following tasks to deploy the Netskope Integration on NetWitness Platform.

- Task 1. Download and Install the Docker Image on the Decoder Host
- Task 2. Create and Publish Policy for Netskope Integration
- Task 3. Configure Netskope Integration from Policy Details View
- Task 4. Enable and Start the Container Service
- Task 5. Capture Interface in Decoder
- Task 6. Verify Netskope Events Received at Decoder
- Task 7. Verify Events Meta from Netskope in Investigate View

Task 1. Download and Install the Docker Image on the Decoder Host

This topic explains how to download and install the Docker image from the Netskope Docker repository on the Decoder Host.

To Download and Install the Docker Image on the Decoder Host

1. SSH to the decoder host.
2. Download the image `docker.io/nsteam/cloudtap-stitcher:125` from the Netskope Docker repository.

   Note: This image tag will change for future updates. Refer to the NetSkope Plugin description in CCM for applicable/supported versions.

3. Do one of the following:
   - If the internet is available on the Decoder host, run the following command:

     `podman pull docker.io/nsteam/cloudtap-stitcher:125`
   - If the internet is unavailable on the Decoder host, then download the image from a machine with the internet as a tar file using `podman save` or `docker save`. Then, load that image on the decoder host using `podman load` or `docker load`:
     - `podman save -o cloudtap-stitcher-125.tar docker.io/nsteam/cloudtap-stitcher:125`
     - Compress the tar file using the command: `gzip cloudtap-stitcher-125.tar` and copy the image archive to the decoder host.
     - On the decoder host, run the following command to load the image from the file: `podman load -i cloudtap-stitcher-125.tar.gz`

Task 2. Create and Publish Policy for Netskope Integration

You must create a policy with Netskope Integration plugin and assign it to one or more groups having a decoder service and publish the policy.

Prerequisites

- Ensure that the **Netskope Integration** plugin type is available at the **SASE Integration Plugin** tab.
- Ensure that the group with one or more decoder services is created.

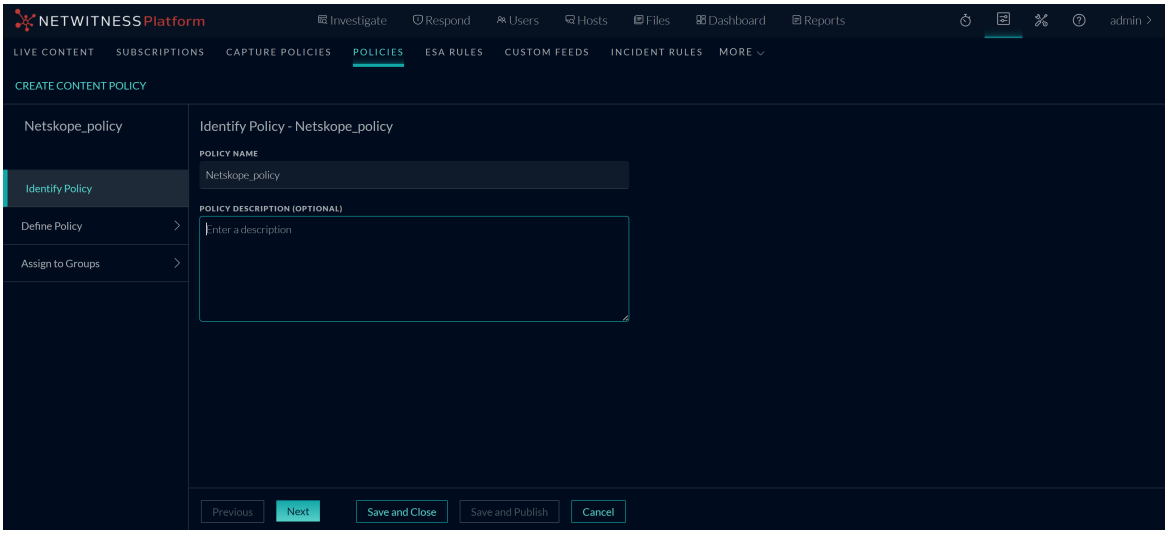
Supported Hosts

- Packet Decoder
- Packet Hybrid

To Create and Publish Policy for Netskope Integration

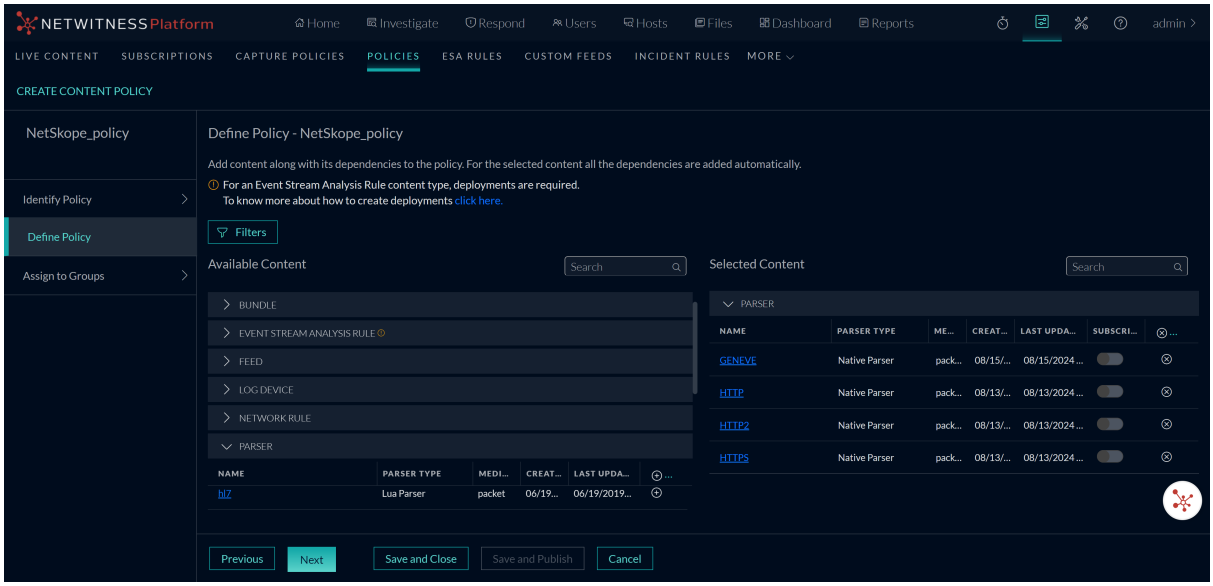
1. Go to (Configure) > **Policies**.
2. In the policies panel, click **Content**.
3. Click **Policies**.

   The available policies are displayed.
4. Click + **Create New** to add a new policy.
5. In the **New Policy** panel, do the following:
   a. Enter a unique policy name.
   b. (Optional) Enter a description for the policy.



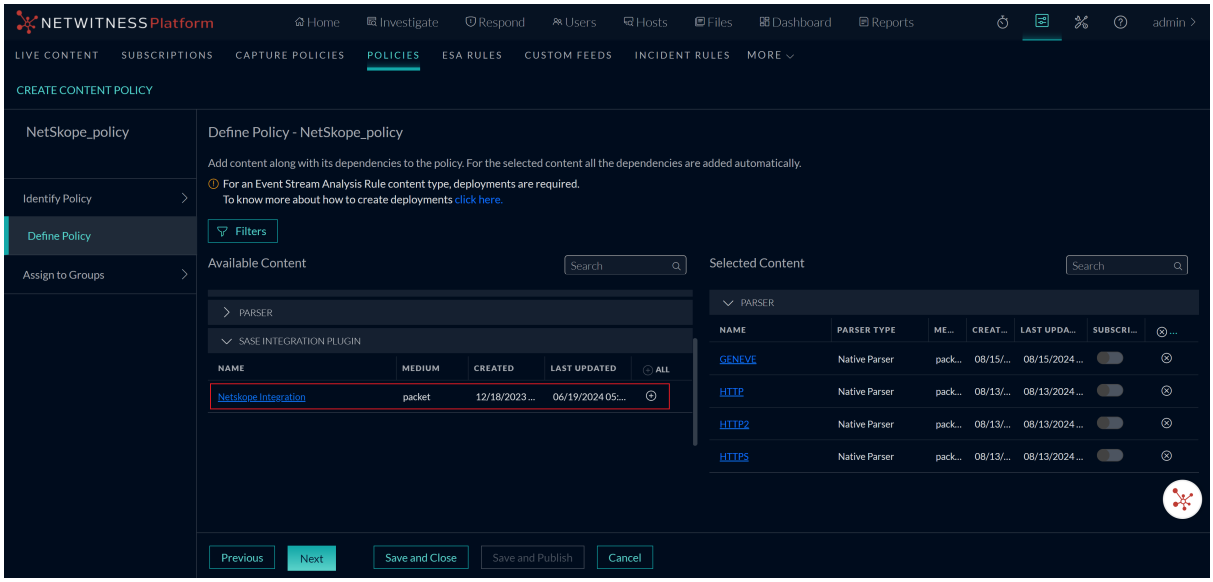
6. Click **Next**.

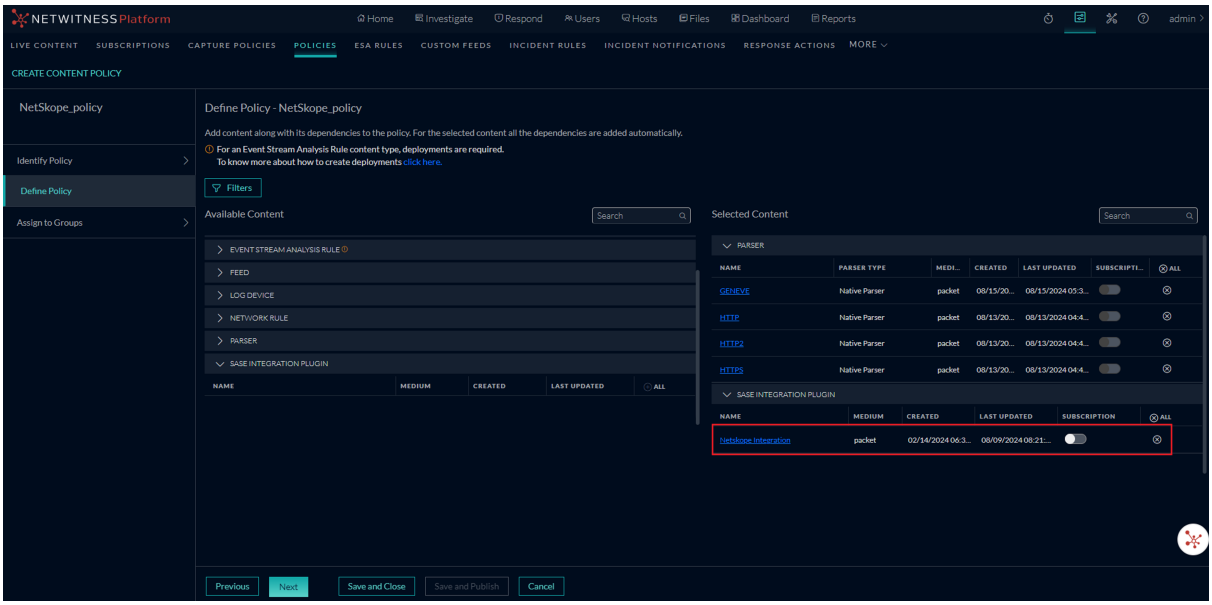
7. In the **Available Content**, first select **Parser** and click + to add the required Geneve, HTTP, HTTP2, and HTTPS parsers. Customers can also select other system and Lua parsers for more visibility.



8. Select the **Sase Integration Plugin** and click + to add the **Netskope Integration** plugin to the policy.

Note: You can add only one **SASE Integration Plugin** to any particular policy.



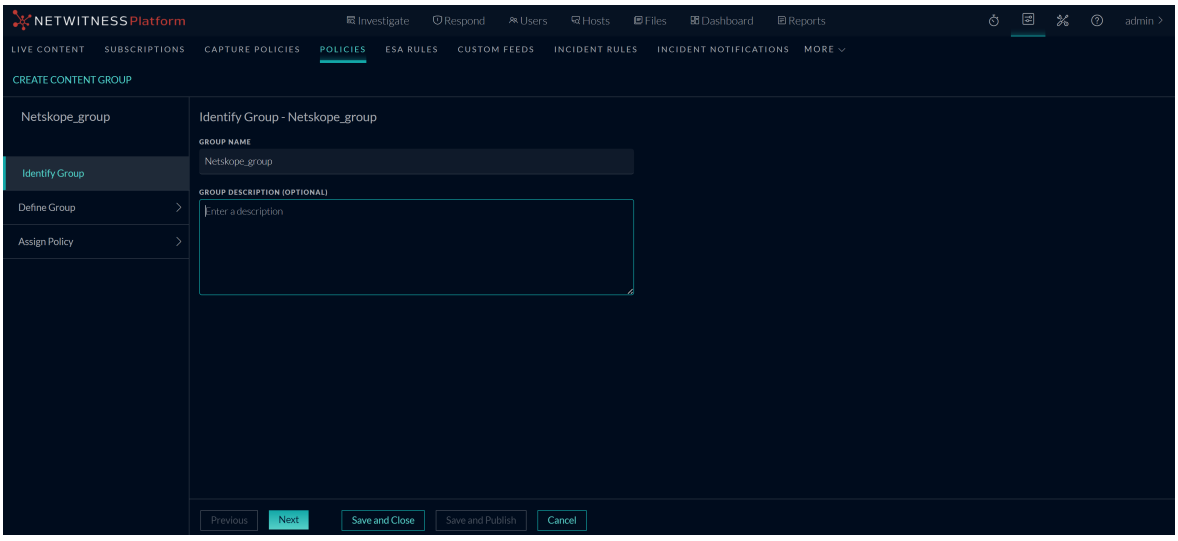


9. Click **Next**.

10. If there are no unassigned groups available, click + Create Group to save the policy and redirect you to the **Create Content Group** screen.

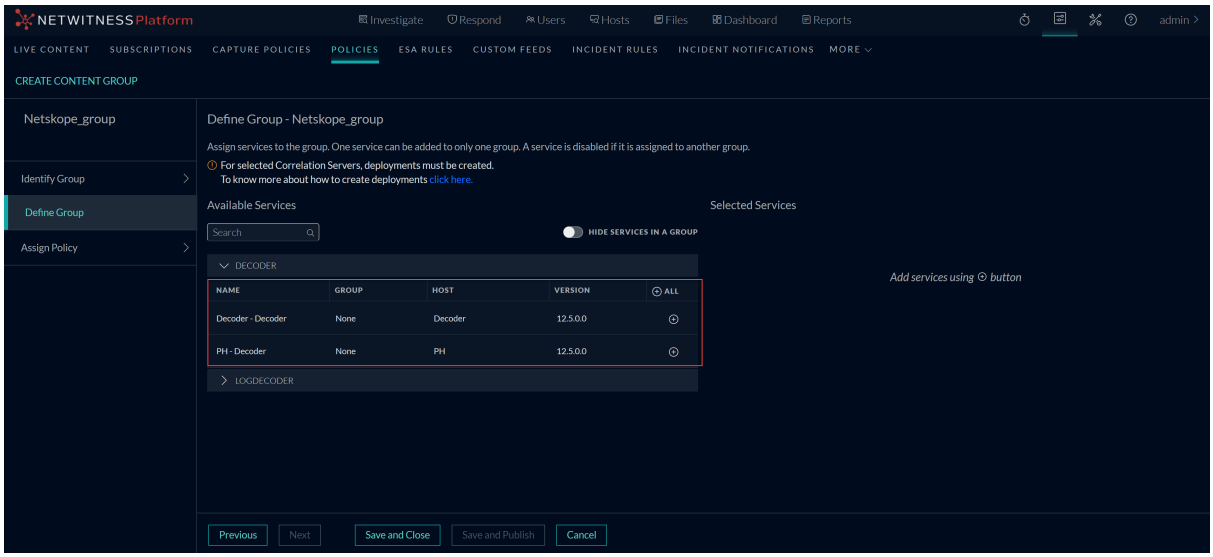
11. In the New Group panel, do the following:

    - Enter the name of the group.
    - (Optional) Enter the description for the group.



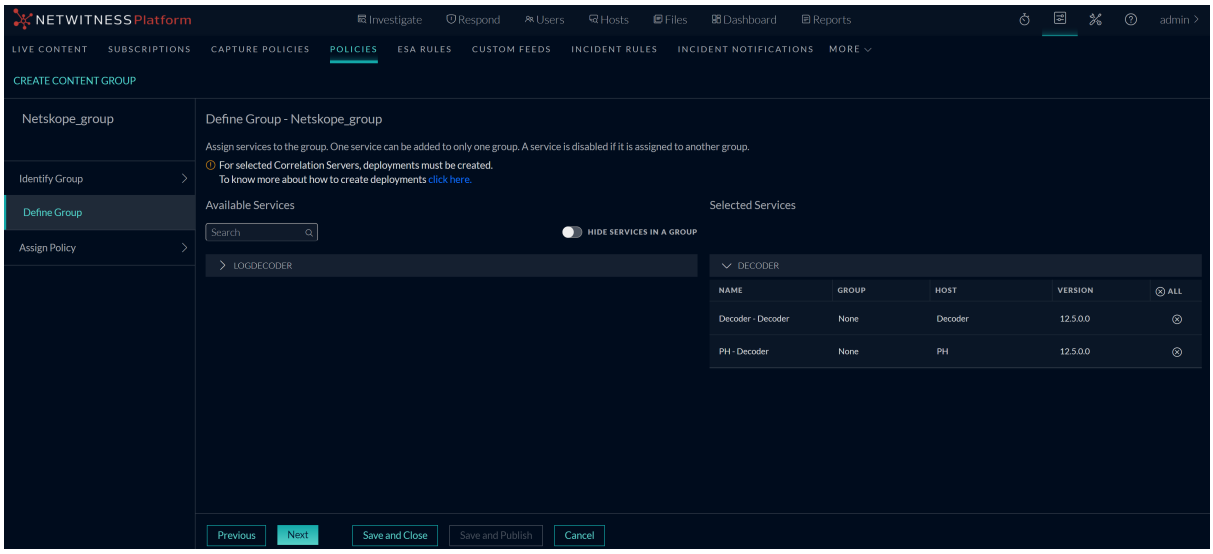
12. Click **Next**.

13. In the **Define Group**, click + to assign services to the group.

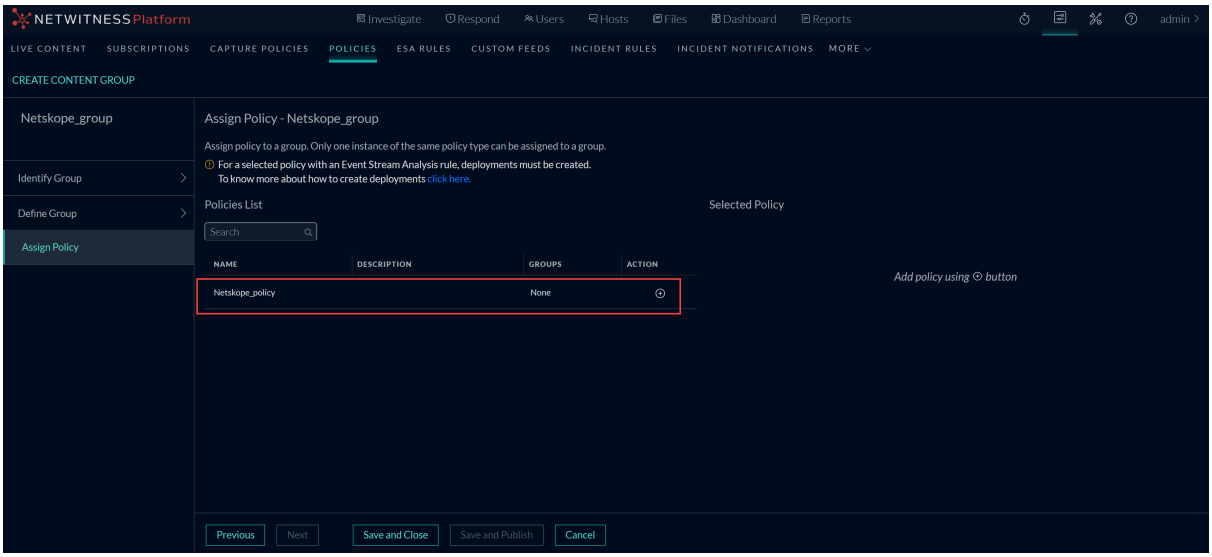


Note:
- You can add only Packet Decoder and Packet Hybrid services to the group.
- A service is disabled if it is assigned to another group.
- A service is disabled if it is not managed by Policy-based Centralized Content Management.

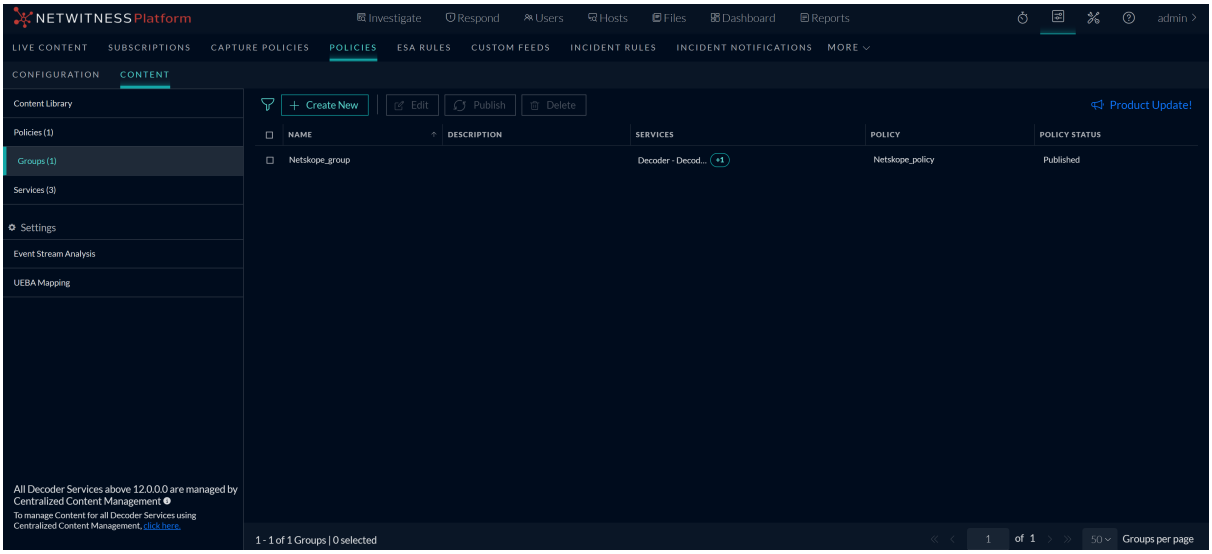
14. Click **Next**.



15. In the Assign Policies, click + to assign policies to a group. You can assign only one policy to any particular group.



16. Click **Save and Publish** to save and publish the settings.



IMPORTANT: Ensure that you always publish the policy after adding the **Netskope Integration** plugin to the policy to deploy the plugin to the Decoder service.

Note: You can also publish a policy from the **Policy Details** screen. For more information on publishing a policy from the **Policy Details** screen, refer to the View a Policy topic.

For more information on Policies, see Manage Policies.

For more information on Groups, see Manage Groups.

Next, go to the policy details view and configure the Netskope Integration.

Task 3. Configure Netskope Integration from Policy Details View

Administrators can configure the Netskope Integration to capture the network data from the decoder service within a policy. The data is then processed by NetWitness so that it can provide a comprehensive view of network traffic and malicious activity. Analysts can use this data to monitor network traffic, identify threats, and investigate any malicious behavior.

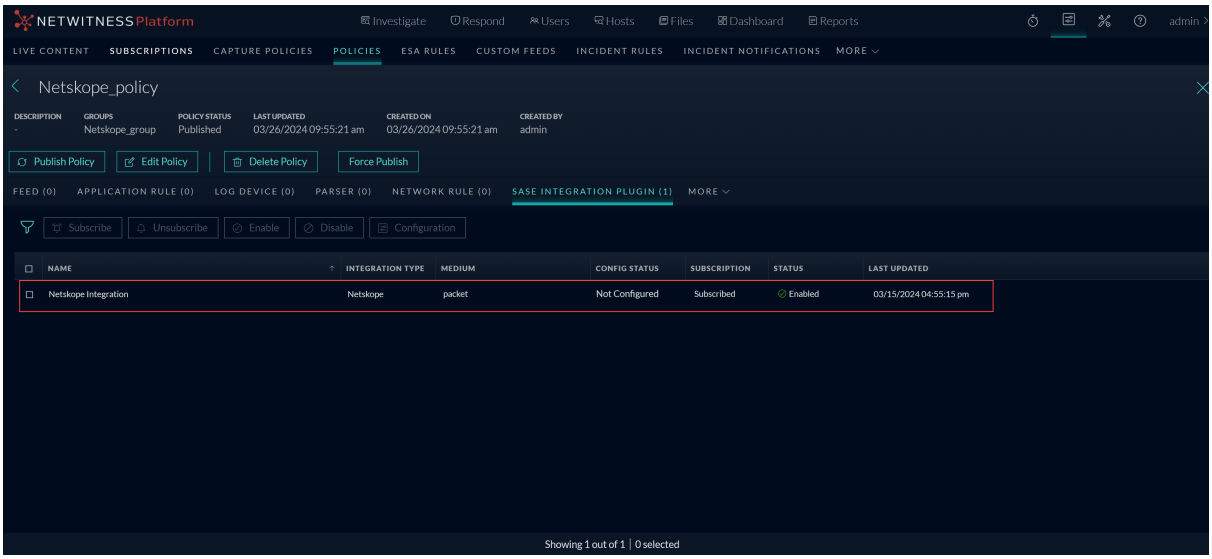
Prerequisites

Before you begin configuring the Netskope Integration, ensure that you have the following details:

- Ensure there is a policy created with the Netskope Integration plugin, and the policy is associated with the group that has a Decoder service configured, and the policy is published.
- Ensure you have the names of the Google Cloud Platform (GCP) or Amazon Web Services (AWS) buckets ready. The Bucket Authentication (.JSON file) for configuration is optional for GCP and AWS. The Region details are also necessary if you select AWS.

To Configure the Netskope Integration

1. Go to  **(Configure)** > **Policies**.
2. In the policies panel, click **Content**.
3. In the left panel, click **Policies**.
4. Do one of the following:
   a. Click the policy name containing the **Netskope Integration** plugin type to view the policy details.
   b. Click a row to view details about the selected policy and click **View Details**.
5. Click **More** > **SASE Integration Plugin** tab.



IMPORTANT: The **Configuration** button will be disabled when the policy status is **Unpublished**, **Failed**, or **N/A**. For more information, see Filter Policies.

6. Select the **Netskope Integration** plugin and click **Configuration**.

   The Configuration dialog is displayed.

7. In the **Add Bucket Configuration** section, do the following:

   a. Select the decoder service from the **Decoder** drop-down list.

   Note: A bucket can be configured with only one decoder at a time unless the origin filter option is used. With the origin filter option, you can configure the same bucket with different origin filters on different decoders simultaneously.

   b. Select either **Google Cloud Platform (GCP)** or **Amazon Web Services (AWS)** from the **Cloud Service Provider** drop-down list.

   c. Enter the GCP or AWS bucket name from which the decoder needs to fetch the data. For example, **netskope-traffic-capture**.

   Note: Bucket names must be more than two characters and can only contain lowercase letters, numeric characters, dashes (-), underscores (_), and dots (.). Spaces are not allowed. For example, **netskope-traffic-capture**, **netskope-traffic-capture1**.

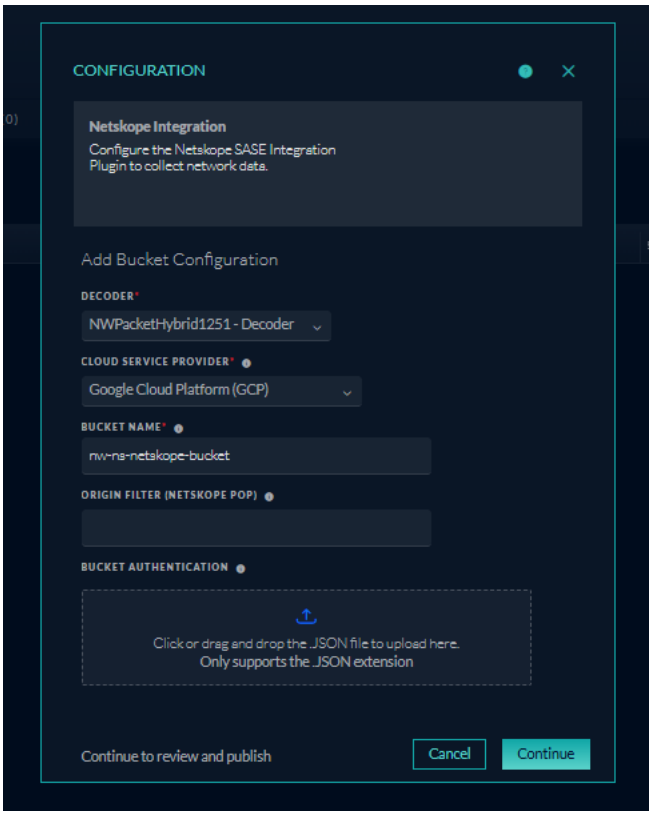
   IMPORTANT: Bucket Authentication is optional if you have already configured GCP / AWS default credentials.

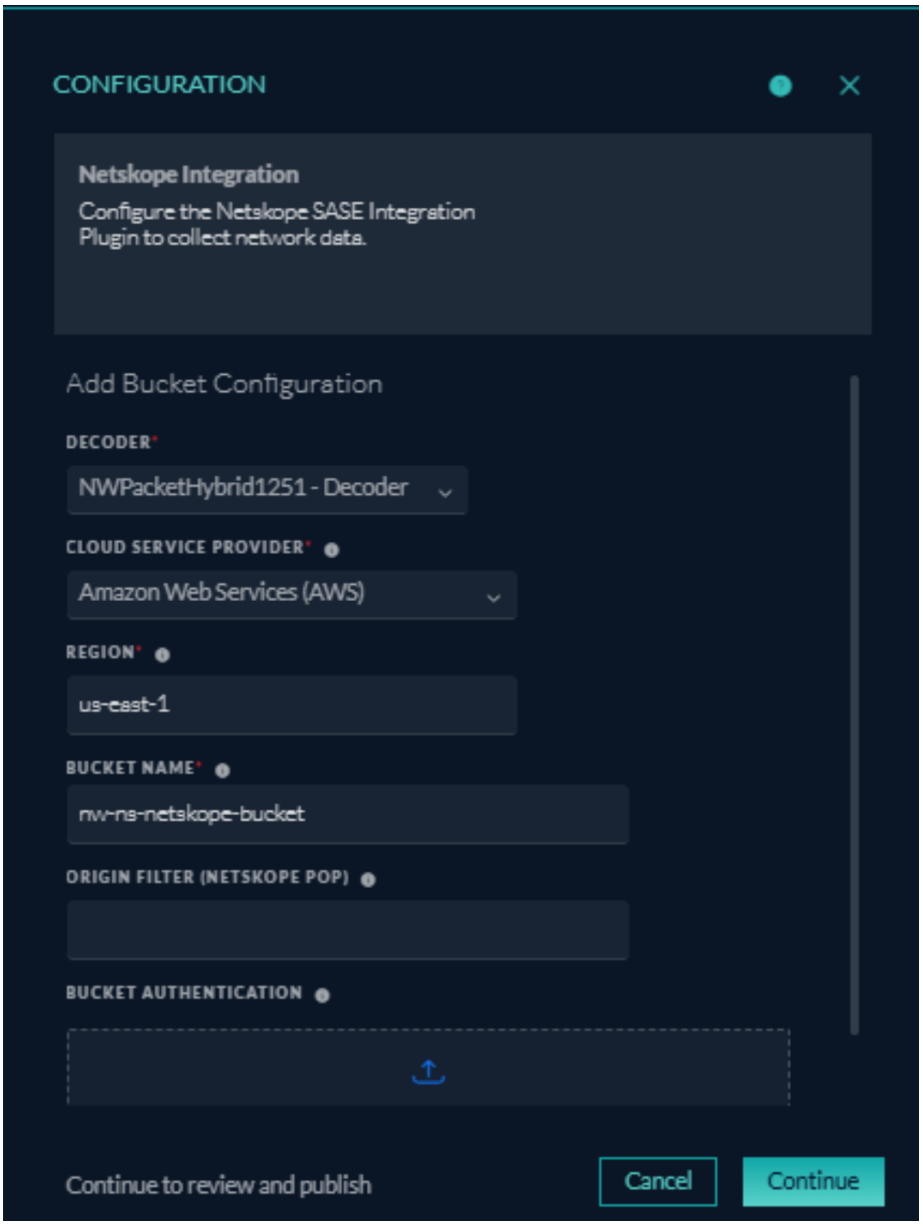
   d. Enter the optional Origin Filter (Netskope POP, where POP is Point of Presence). The default setting selects all the sources from the GCP or AWS bucket. The Origin Filter configures a specific traffic source to process the traffic. For more information on Origin Filter, refer to *Scaling with Multiple Stitcher Instances* under Configuring the Cloud TAP Stitcher.

   e. (Optional for GCP and AWS) In the **Bucket Authentication** area, click or drag and drop the (.JSON) file to upload. Bucket Authentication is used to authenticate access to a bucket in GCP or AWS.

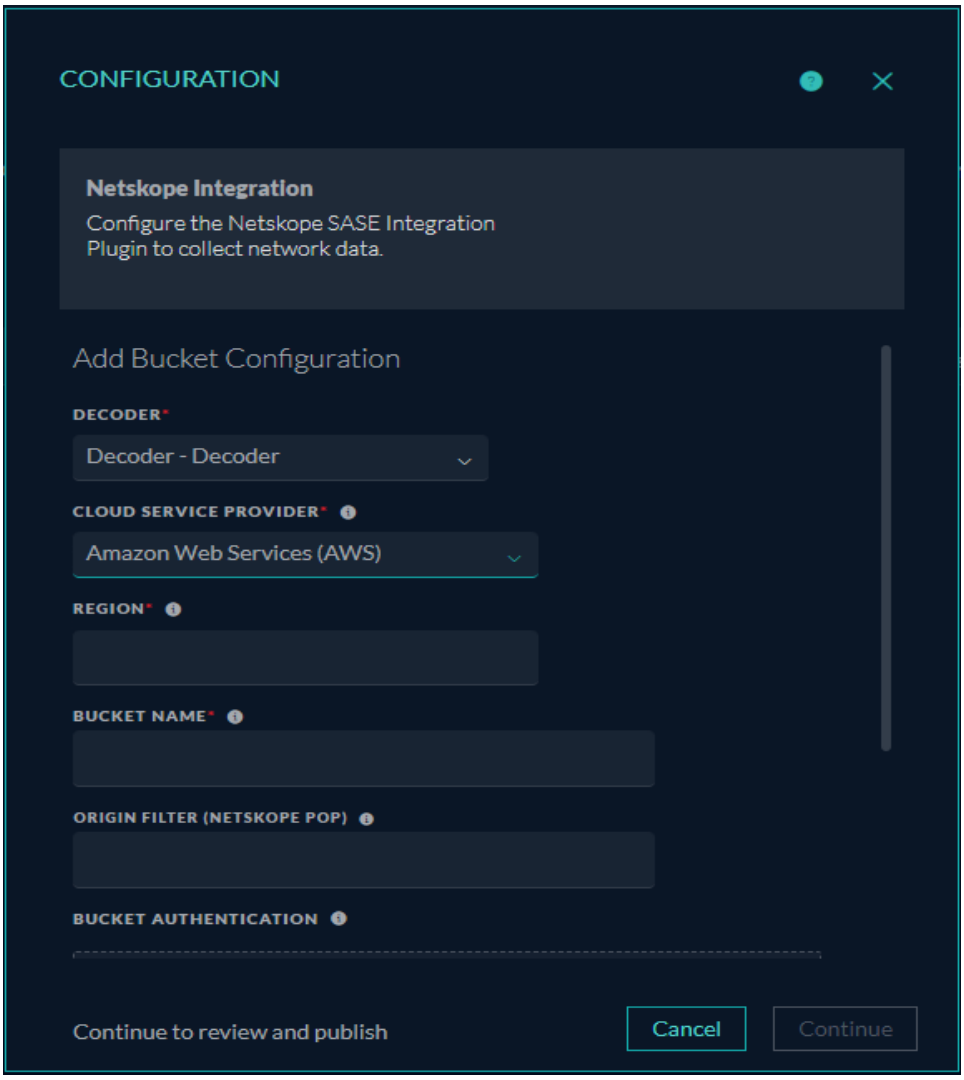
Note:
• The Bucket Authentication must have a valid bucket authentication key format extension (.JSON).
• The size of the Bucket Authentication (.JSON file) must not exceed 8 MB.

f. Click **Continue**.



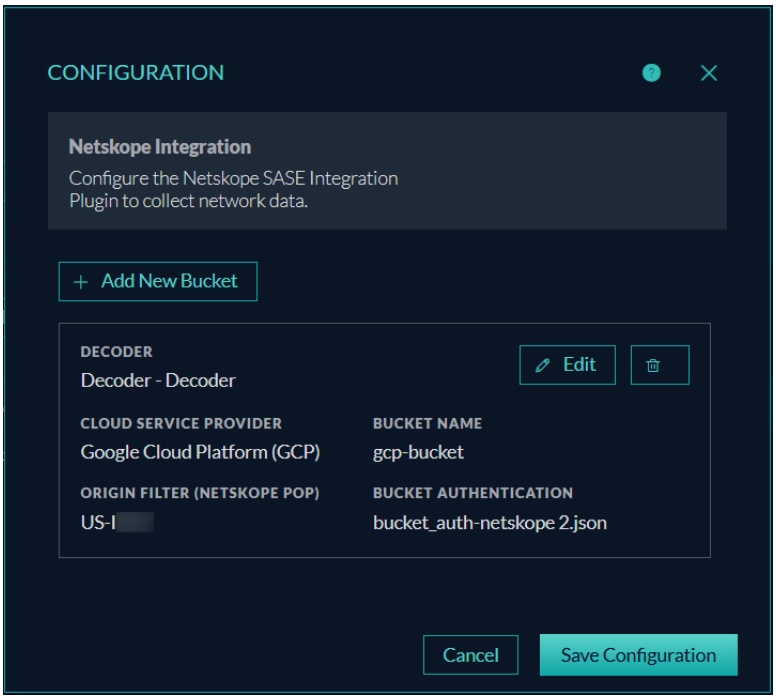


Note: Enter the Region details, if you select AWS as the Cloud Service Provider.



8. Click + **Add New Bucket** to add new buckets, which navigates to the **Add Bucket Configuration** section. Follow **step 7** to configure the bucket details.

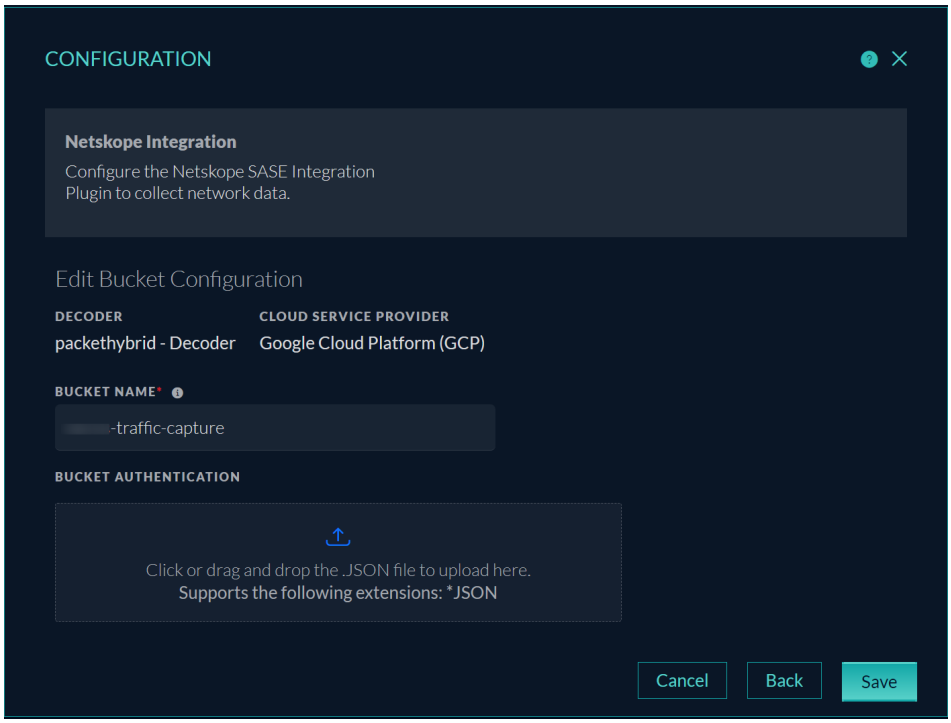
Note: The NetWitness Netskope plugin supports just one bucket per decoder.



9. If you want to modify the existing bucket configuration, perform the following steps:
   - Click **Edit** will navigate to the **Edit Bucket Configuration** Section.
   - Modify the details and click **Save**.

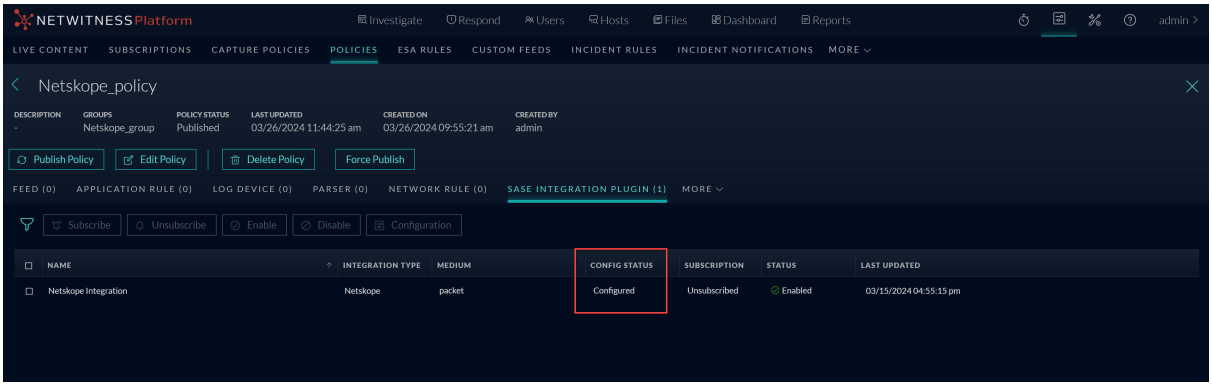
   Note: You cannot edit the decoder or cloud service for an existing bucket configuration. If you need to change the decoder, you must delete the existing bucket configuration and add a new

bucket.



10. Click **(Delete)** to remove the bucket configuration permanently.
11. Review the bucket configuration details and click **Save Configuration**.

    To verify if the configuration was completed successfully, ensure that the **Config Status** column displays **Configured** for the Netskope Integration.

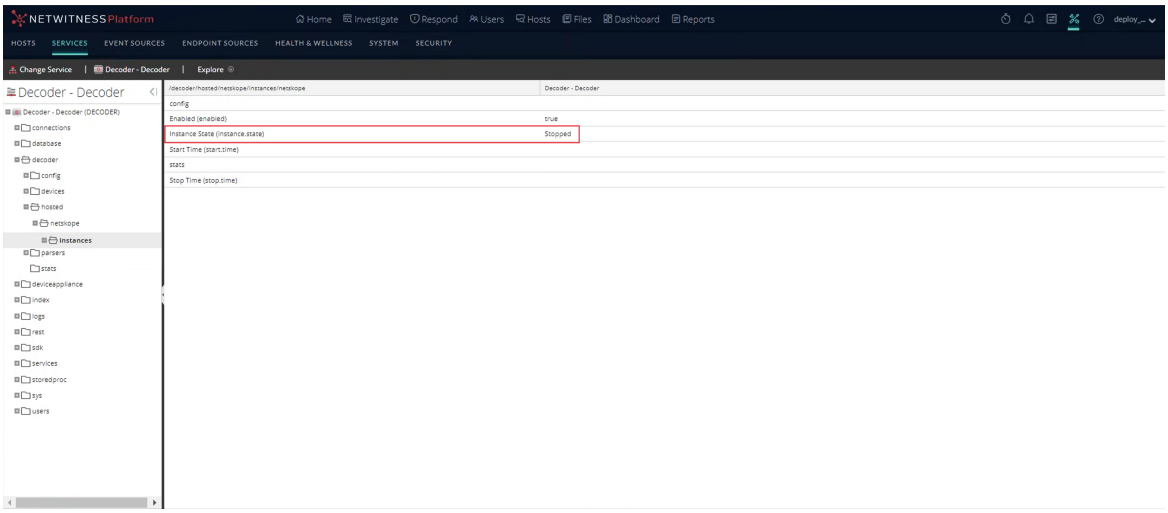


12. Start the netskope instance by performing the following steps:

    a. Go to **(Admin)** > **Services**.

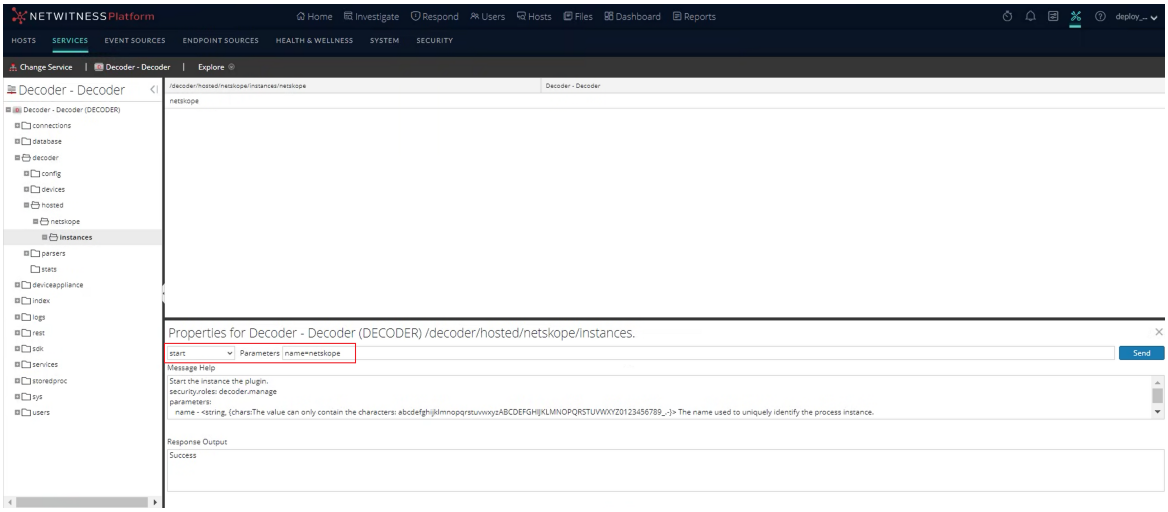
    b. Select the **Decoder** service and click > **View** > **Explore**.

    c. Select **hosted > netskope > instances**. Check if the **Instance State** status is **Stopped**.

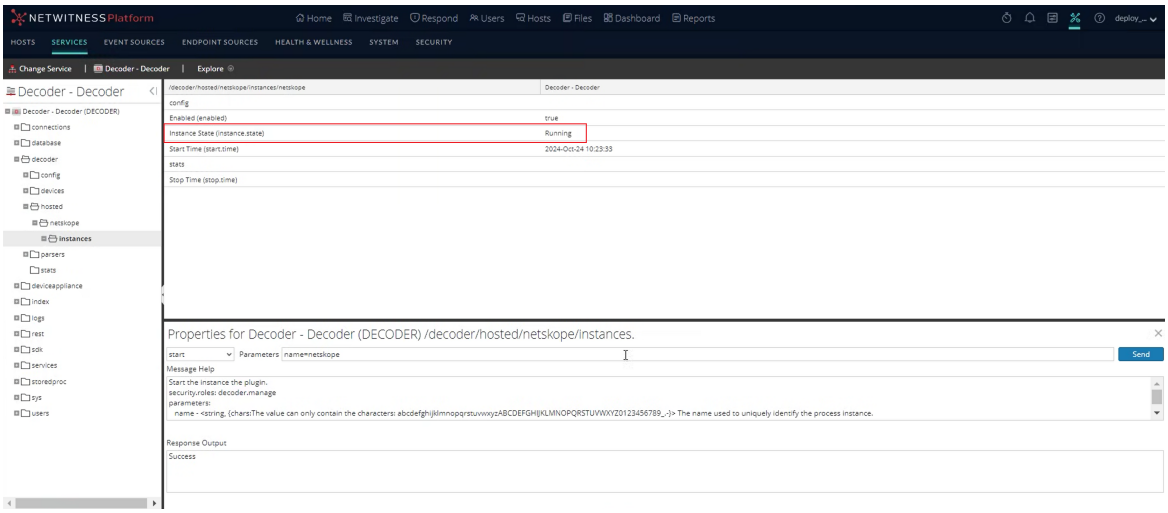


d. On the left panel, select **instances** and right-click **properties**.

e. In the **Properties** section for the instance, choose the **start** option from the drop-down list, and in the Parameters field, type **name=netskope** and click **Send**.



The **Instance State** status displayed as **Running** ensures the instance is manually started successfully.



Task 4. Enable and Start the Container Service

Enable and start the systemd container service which is a one-time operation. When you enable it, on the system restart, the Container will start up before the Decoder.

Note: By default, the systemd container service is deployed on a Decoder but is not enabled unless you configure the Plugin. On enabling it, the Container Service will start on reboot.

First, SSH to decoder as root user and run the below commands to enable and start the container:

1. `systemctl enable nw-netskope-container.service`
2. `systemctl start nw-netskope-container.service`

The Container Service creates Container which then creates a virtual interface, for example; `cni-podman1`. In this interface, the Netskope plugin ingest packets which the Decoder captures.

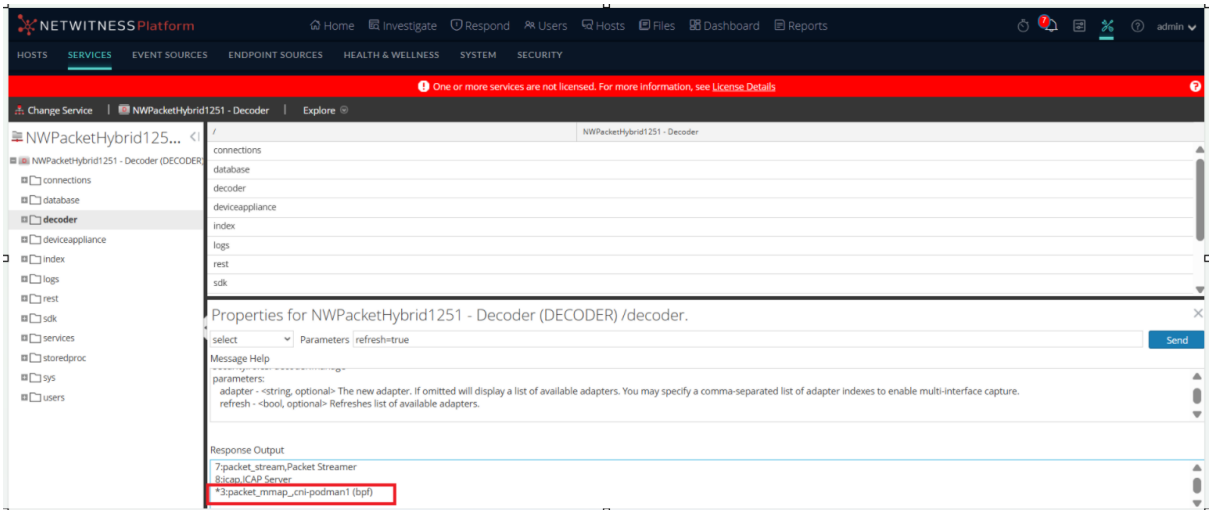
Task 5. Capture Interface in Decoder

You must select a network adapter **(packet_mmap_,cni-podman1 (bpf))** and enable **Capture Autostart** option through which the Decoder captures packets and processes the data.

Perform the following to Capture the Interface in Decoder:

1. Log in to the NetWitness Platform.
2. Go to  **(Admin)** > **Services**.
3. Select the **Packet Decoder** service and click  > **View** > **Explore**.
4. On the left panel, select **decoder** and right-click **properties**.
5. In the **Properties** section for the Decoder service, choose **select** option from the drop-down list, and in the **Parameters** field, type **refresh=true** and click the **Send** button.

Using the **refresh** option will detect the new capture interfaces available on Decoder. This will add `cni-podman1` interface to the list.



6. Navigate to the **Explore** page of the Decoder service.

   The Configure view for the Decoder service is displayed with the **General** tab open.

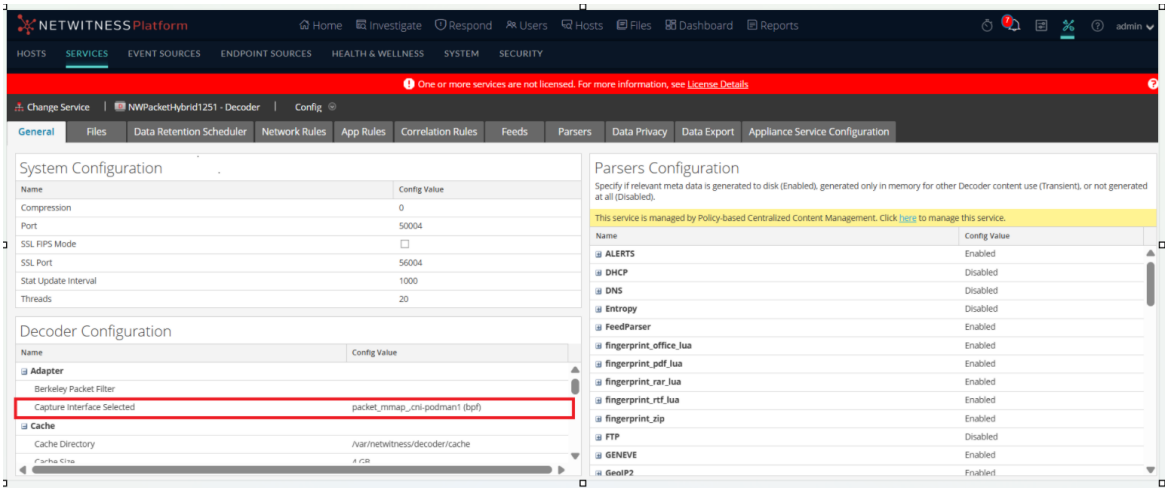
7. Under the **Decoder Configuration** section, do the following:

   a. Configure the BPF to filter port 6081 to capture Geneve traffic.

   b. Set the **Capture Interface Selected** to **(packet_mmap_,cni-podman1 (bpf))** network adapter.

      The decoder uses the **cni-podman1** network interface, monitoring the network traffic flowing through it. The decoder can capture and analyze the packets sent and received by the containers.

   c. Enable the **Capture Autostart** option.



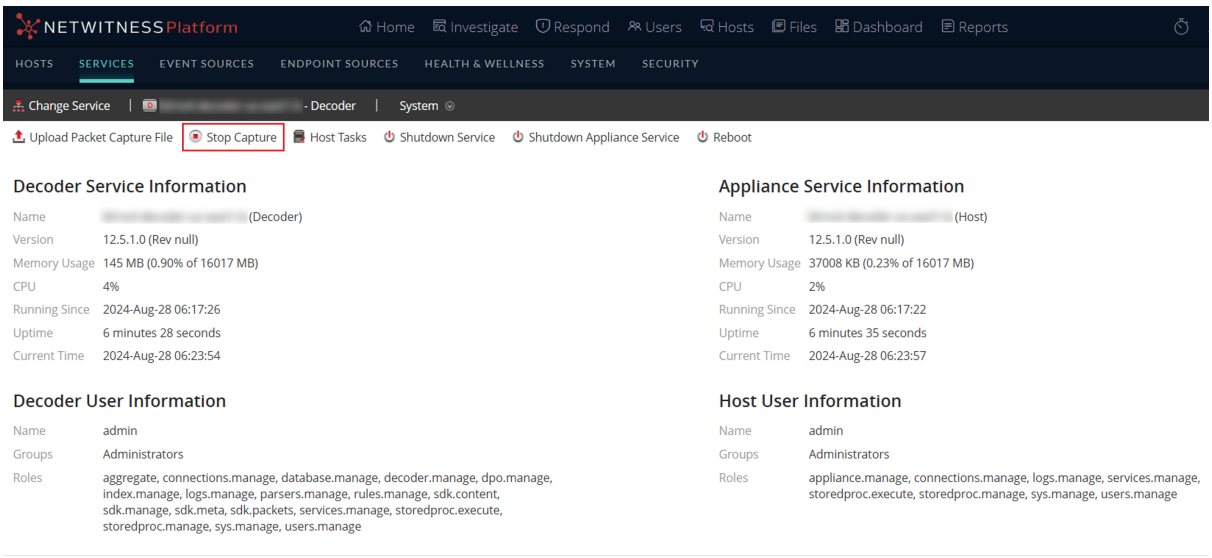
8. Click **Apply** to save the changes.

9. To restart the Decoder service, go to the **Services** view, select the Decoder service, and click  > **Restart**.

10. A Confirmation dialog request is displayed. To restart the service, click **Yes**.

11. (Optional) Navigate to the System view of the Decoder service and check if the Decoder is capturing the data.

    This option ensures the decoder has already started capturing the packets.

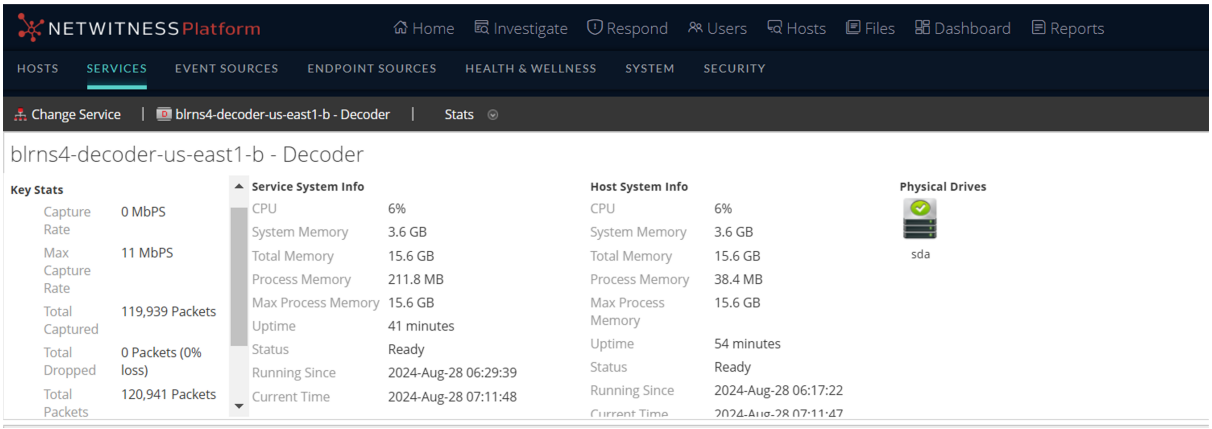


Task 6. Verify Netskope Events Received at Decoder

You can analyze the Netskope events that have been received by the Decoder and verify their accuracy.

To Verify the Netskope Events Received at Decoder

1. Log in to the NetWitness Platform.
2. Go to  **(Admin)** > **Services**.
3. Select the **Packet Decoder** service and click  > **View** > **Stats**.
4. Under the **Key Stats** section, check the values for **Capture Rate**, **Max Capture Rate**, **Total Dropped**, and **Total Captured packets** for the decoder service.



Task 7. Verify Events Meta from Netskope in Investigate View

To verify Netskope events, you must first aggregate the Decoder service into the Concentrator and then go to the **Investigate** > **Events** page to view the Netskope events.

- Add the Decoder Service in the Concentrator
- Verify from the Investigate > Events View

Add the Decoder Service in the Concentrator

1. Log in to the NetWitness Platform.
2. Go to **(Admin)** > **Services**.
3. In the **Services** list, select the **Concentrator** service.
4. Click > **View** > **Config**.

   The Services Config View of the Concentrator is displayed.
5. Select the **Sources** tab.
6. Click and select Available Services.

   The Available Services dialog is displayed.
7. Select the **Decoder** service and click **OK**.

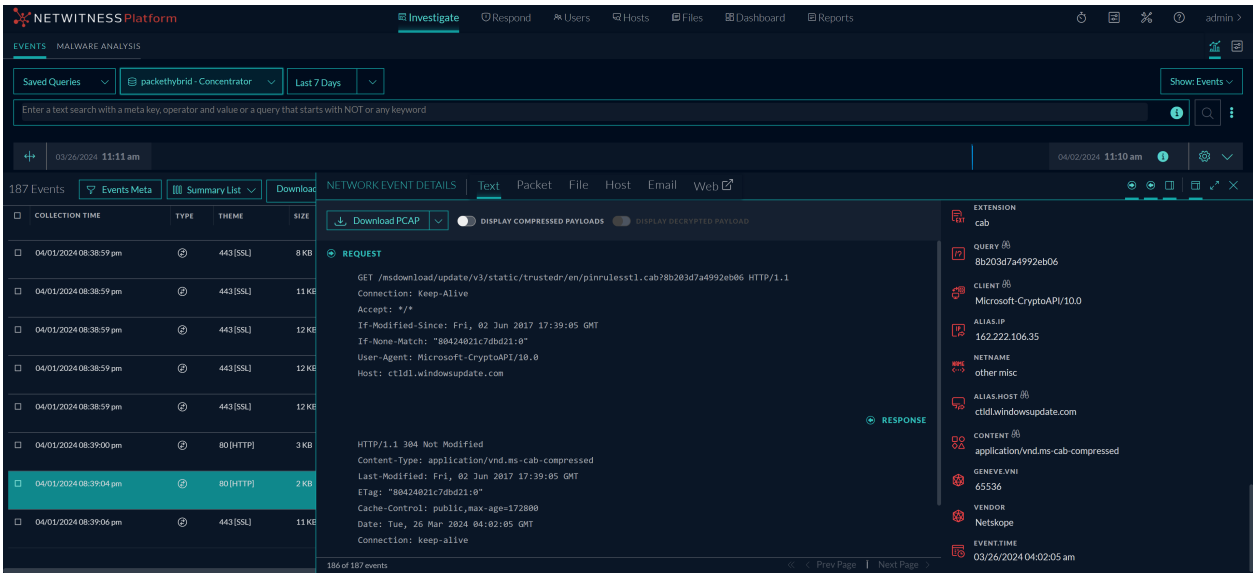
   The service authentication dialog box is displayed.

   Note: The services with the Trust Model enabled must be added individually. You are prompted to provide a username and password for the selected service.
8. Enter the Username and Password for the service.

9. Click **OK**.
10. Click **Apply**.

Verify from the Investigate > Events View

1. Go to **Investigate** > **Events**.
2. Select the **Concentrator** Service from the **Services** selection drop-down list.
3. Click  to load the Netskope events data.



To verify other events in the Investigate view refer to Begin an Investigation in the Events View.

Deploy Netskope Integration Using NwConsole

This topic describes how to deploy the Netskope Integration for users using the NwConsole.

Prerequisites

Before proceeding, it is important to make sure the following:

- The NetWitness Platform (Admin Server and Packet Decoder Host) is on version 12.5.1 or later.
- You are connected to Live Services under the **(Admin)** > **System** > **Live Services** page.
- Ensure that you have a network connection between the Google Cloud Platform (GCP) or Amazon Web Services (AWS).
- You must have the GCP or AWS bucket names available for configuration. Bucket Authentication (.JSON file) is optional for GCP and AWS:
  - The Bucket Authentication Key (.JSON file) is used to authenticate access to a bucket in GCP or AWS. Creating a Bucket Authentication Key (.JSON file) is a two-step process:
    - Create a service account in GCP with the role **Storage Object Viewer (roles/storage.objectViewer)**. For more information, see the topic Create service accounts.
    - Create a service account key in GCP. For more information, see the topic Create and delete service account keys.
- In case of AWS, you must have an IAM user with role assigned having below access permissions to the bucket.

  ```
  "Action": [ "s3:GetObject", "s3:ListBucket ]
  "Resource": ["arn:aws:s3:::<bucket-name>", "arn:aws:s3:::<bucket-name>/*"]
  ```

  To create the .json file

- Access IAM user and obtain the following keys.
  - **Access Key**
  - **Secret Key**
- Create the **Auth Key JSON file**

  For Example: `AWS-Bucket-Auth-Key.json`
- Update the keys as shown below.

  ```
  {
  "access_key_id": "<copy access key>",
  "secret_access_key": "<copy secret key>"
  }
  ```
- You must move the `AWS-Bucket-Auth-Key.json` file into Decoder node.

You must perform the following tasks to deploy the Netskope Integration on NetWitness Platform.

Task 1. Download and Install the Docker Image on the Decoder Host

This topic explains how to download and install the Docker image from the Netskope Docker repository on the Decoder Host.

To Download and Install the Docker Image on the Decoder Host

1. SSH to the decoder host.
2. Download the image `docker.io/nsteam/cloudtap-stitcher:125` from the Netskope Docker repository.
3. Do one of the following:
   - If the internet is available on the Decoder host, run the following command:

     ```
     podman pull docker.io/nsteam/cloudtap-stitcher:125
     ```
   - If the internet is unavailable on the Decoder host, then download the image from a machine with the internet as a tar file using `podman save` or `docker save`. Then, load that image on the decoder host using `podman load` or `docker load`:
     - `podman save -o cloudtap-stitcher-125.tar docker.io/nsteam/cloudtap-stitcher:125`
     - Compress the tar file using the command: `gzip cloudtap-stitcher-125.tar` and copy the image archive to the decoder host.
     - On the decoder host, run the following command to load the image from the file: `podman load -i cloudtap-stitcher-125.tar.gz`

Task 2. Deploy and Configure the NetSkope Plugin Using NwConsole

You can search for the NetSkope integration Plugin from the Live Content view and deploy it on the decoder services using NWconsole.

To Deploy NetSkope Integration Plugin on Decoder

1. Log in to the NetWitness Platform.
2. Go to  (Configure) > **Live Content**.
3. Select the **NetSkope Integration Plugin** from the **Resource Types** drop-down list in the Search Criteria panel.

   Note: To narrow the results further, you can use the different options available in the Search Criteria panel.
4. Click **Search**. The available SASE Integration plugins are displayed.
5. In the **Matching Resources** panel, select **Show Results > Grid**.
6. Select the **NetSkope Integration Plugin** checkbox and click **Package > Create**. The resource bundle gets downloaded to your local system.
7. Extract the resource bundle to view the **nw-ns-monitor.zip** package.
8. SSH to the Packet Decoder host.
9. Create a directory by running the following command:

   ```
   mkdir /opt/netskope
   ```
10. Copy the **nw-ns-monitor.zip** package and Bucket Authentication Key (.Json) file (optional for GCP and AWS) to the newly created `/opt/netskope` directory.
11. Connect to the NwConsole utility with the following command: `NwConsole`
12. Login to the Decoder service, using one of the following methods:

    Option 1: `Trusted Auth`

    ```
    > tlogin server=localhost port=56004 username=admin group=Administrators
    cert=/etc/pki/nw/node/node-cert.pem key=/etc/pki/nw/node/node-key.pem

    Successfully logged in to localhost:56004 as session 42376
    ```

    Option 2: Using Password

    ```
    > login localhost:56004:ssl admin

    Successfully logged in to localhost:56004 as session 1561
    ```
13. Navigate to the following directory: `cd /decoder/hosted`
14. Run the following command to install the plugin:

    ```
    upload /opt/netskope/nw-ns-monitor.zip
    ```

    ```
    Transfering nw-ns-monitor.zip, Please Wait...
    Success
    [          :56004] /decoder/hosted>
    ```

    Note: When the installation is complete, exit the NWConsole and ensure that a success message appears.
15. Next, run the following command in SSH: `cd /etc/netwitness/ng/hosted/netskope`

16. Run the following command to create an instance:

    ```
    sh configure-ns-instance.sh
    ```

17. Enter the values in the following fields:

    a. **Instance name**: Enter the instance name. Only alpha-numeric characters are allowed, and spaces are not allowed. For example, `netskope`.

    b. **Bucket name**: Enter the GCP or AWS bucket name from which the decoder needs to fetch the data. For example, ns-traffic-capture.

    > **Note:** Bucket names must be more than two characters and can only contain lowercase letters, numeric characters, dashes (-), underscores (_), and dots (.). Spaces are not allowed. For example, ns-traffic-capture.

    > **Note:** Enter the Region details, if you choose AWS as the Cloud Service Provider.

    > **IMPORTANT:** Bucket Authentication is optional if you have already configured GCP / AWS default credentials.

    c. **Origin Filter**: Enter the optional Origin Filter details. The default setting selects all the sources from the GCP or AWS bucket. The Origin Filter configures a specific traffic source to process the traffic. For more information on Origin Filter, refer to *Scaling with Multiple Stitcher Instances* under Configuring the Cloud TAP Stitcher.

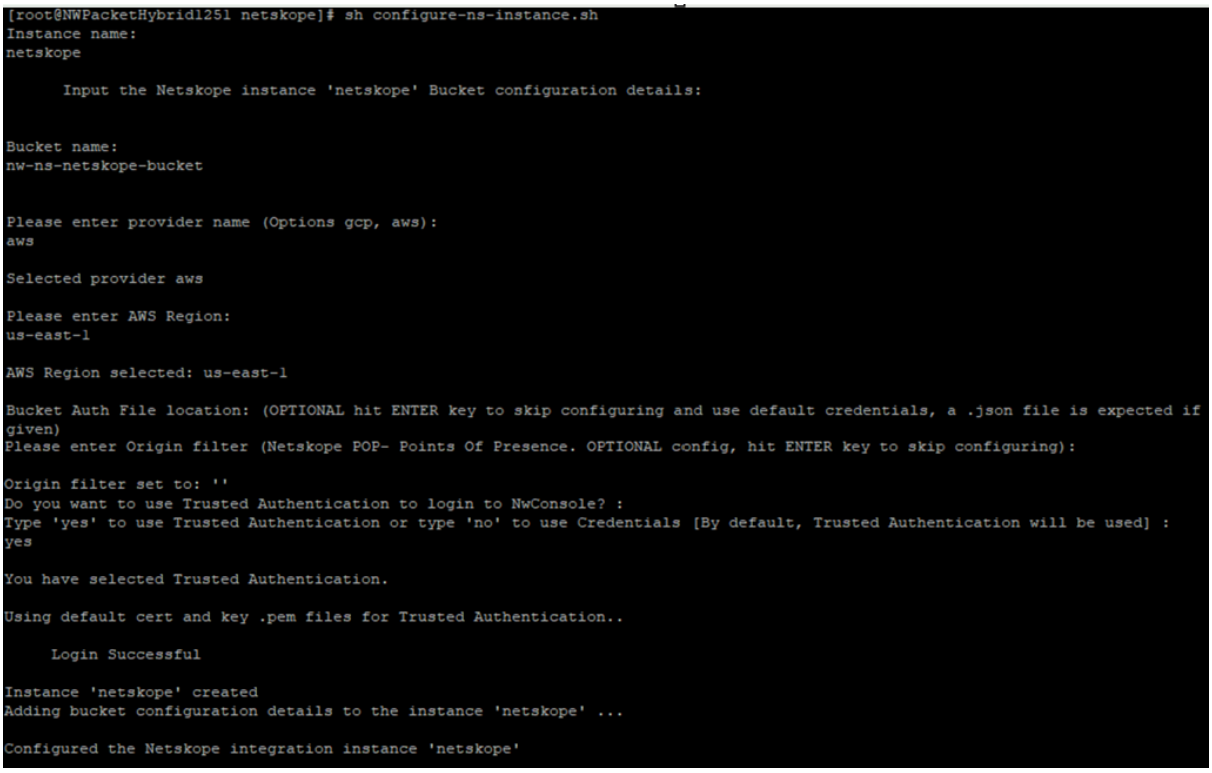
    d. **Bucket Auth File Location**: Specify the path where the Bucket Authentication Key (.JSON) file is placed. For example, /opt/netskope/bucket_auth_key.json. The Bucket Authentication Key (.JSON file) is used to authenticate access to a bucket in GCP or AWS.

    > **Note:** Bucket Authentication file is optional for GCP / AWS, if the customer chooses to not include this file, just press Enter when the Bucket auth file location is prompted by the script. The Bucket Authentication must have a valid bucket authentication key format extension (.JSON). The size of the Bucket Authentication (.JSON file) must not exceed 8 MB.

18. Type **yes** to use the Trusted authentication mechanism to log in to NwConsole; if you type **no**, you need to specify decoder credentials (username and password).

    > **Note:** NetWitness recommends that you use a trusted authentication mechanism for communication with the decoder service, where the option is enabled by default.

19. Type **yes** to enable the instance.



The instance is now configured successfully.

Task 3. Enable and Start the Container Service

Enable and start the systemd container service which is a one-time operation. When you enable it, on the system restart, the Container will start up before the Decoder.

> **Note:** By default, the systemd container service is deployed on a Decoder but is not enabled unless you configure the Plugin. On enabling it, the Container Service will start on reboot.

First, SSH to decoder as root user and run the below commands on the Explore view to enable and start the container:

1. `systemctl enable nw-netskope-container.service`
2. `systemctl start nw-netskope-container.service`

The Container Service creates Container which then creates a virtual interface, for example; `cni-podman1`. In this interface, the Netskope plugin ingest packets which the Decoder captures.

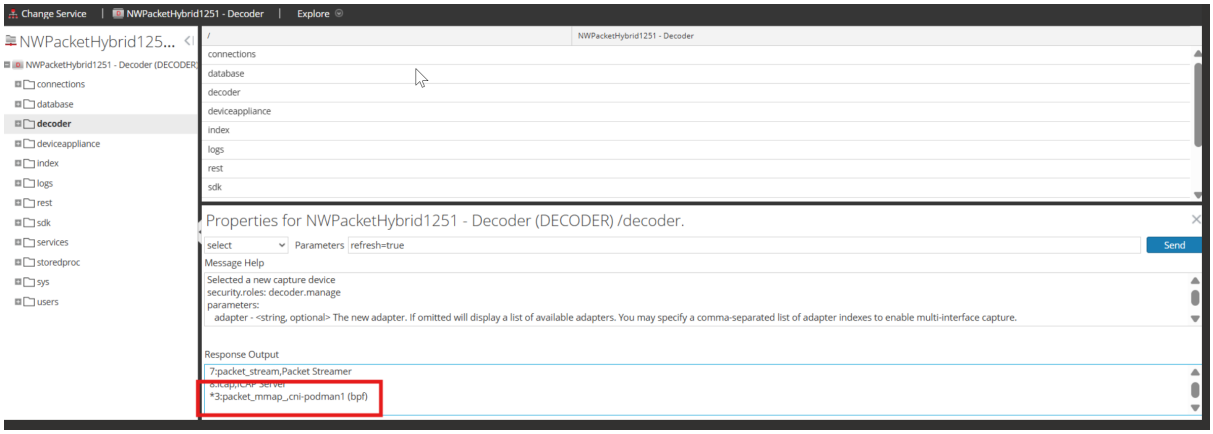
Task 4. Capture Interface in Decoder

You must select a podman network adapter **(for example, packet_mmap_,cni-podman1 (bpf))** and enable **Capture Autostart** option through which the Decoder captures packets and processes the data.

Perform the following to Capture the Interface in Decoder:

1. Log in to the NetWitness Platform.
2. Go to **(Admin)** > **Services**.
3. Select the **Packet Decoder** service and click > **View** > **Explore**.
4. On the left panel, select **decoder** and right-click **properties**.
5. In the **Properties** section for the Decoder service, choose **select** option from the drop-down list, and in the **Parameters** field, type **refresh=true** and click the **Send** button.

   Using the **refresh** option will detect the new capture interfaces available on Decoder and update the podman interface. Select the podman interface (for example, `cni-podman1`).



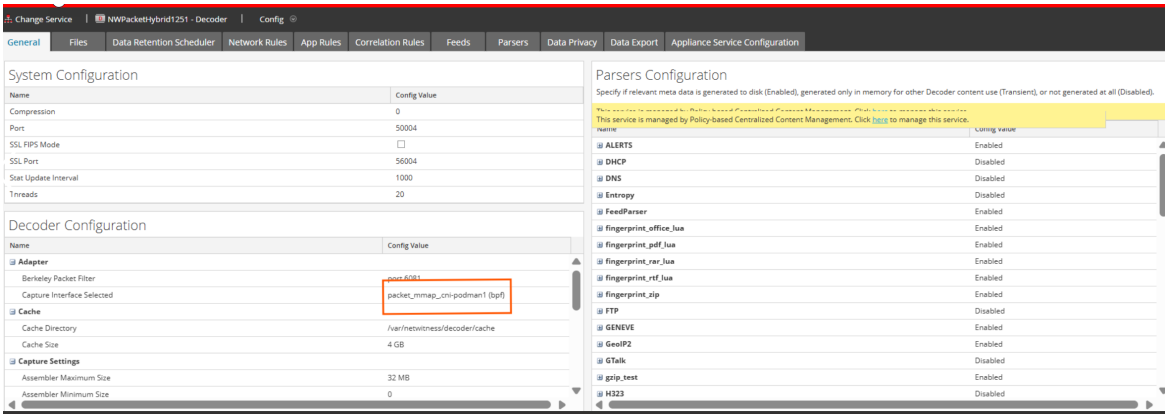
6. Navigate to the **Explore** page of the Decoder service.

   The Configure view for the Decoder service is displayed with the **General** tab open.

7. Under the **Decoder Configuration** section, do the following:
   a. Configure the BPF to filter port 6081 to capture Geneve traffic.
   b. Set the **Capture Interface Selected** to **(packet_mmap_,cni-podman1 (bpf))** network adapter.

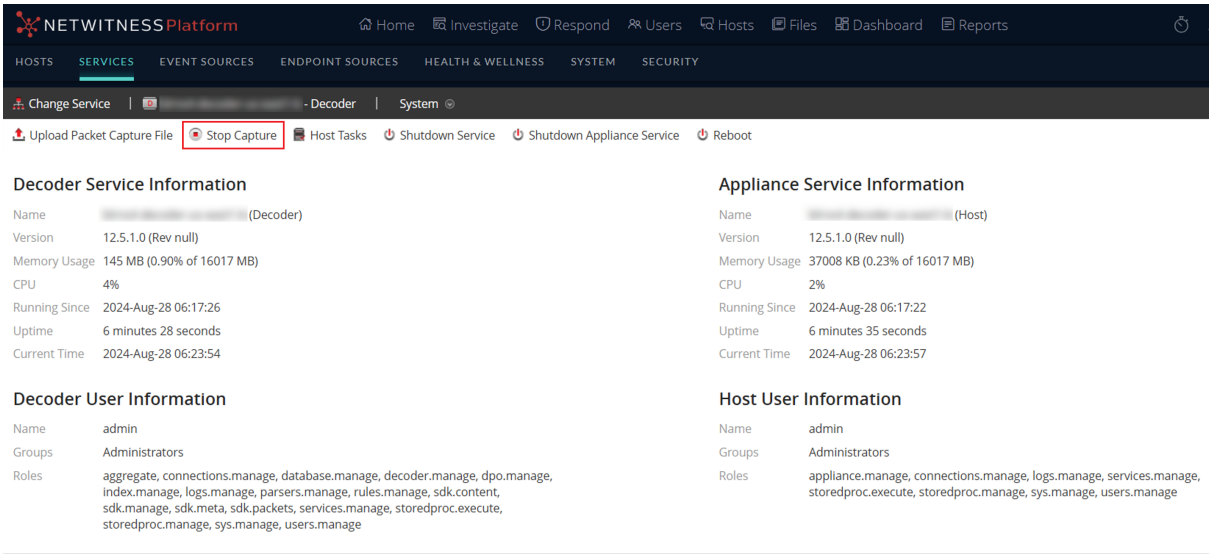
      The decoder uses the **cni-podman1** network interface, monitoring the network traffic flowing through it. The decoder can capture and analyze the packets sent and received by the containers.

   c. Enable the **Capture Autostart** option.



8. Click **Apply** to save the changes.
9. To restart the Decoder service, go to the **Services** view, select the Decoder service, and click > **Restart**.
10. A Confirmation dialog request is displayed. To restart the service, click **Yes**.
11. (Optional) Navigate to the System view of the Decoder service and check if the Decoder is capturing the data.

    This option ensures the decoder has already started capturing the packets.



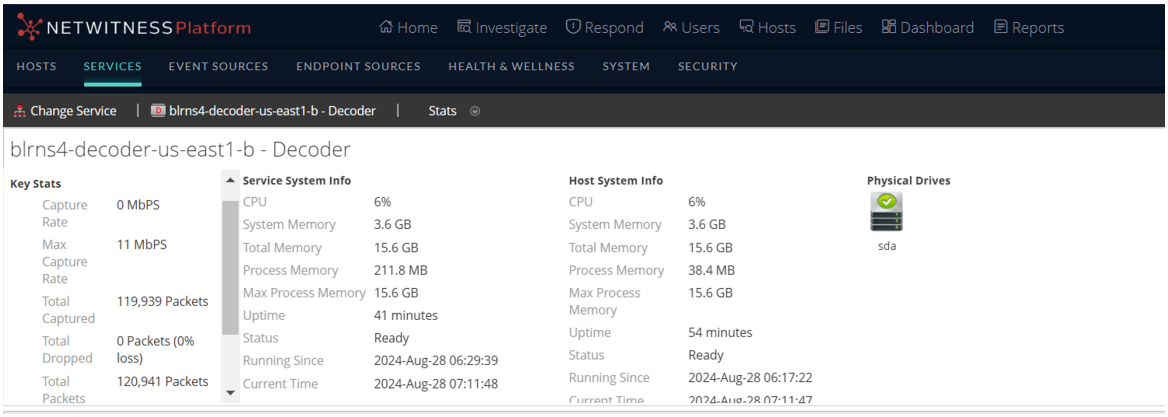
Task 5. Verify Netskope Events Received at Decoder

You can analyze the Netskope events that have been received by the Decoder and verify their accuracy.

To Verify the Netskope Events Received at Decoder

1. Log in to the NetWitness Platform.
2. Go to **(Admin)** > **Services**.
3. Select the **Packet Decoder** service and click > **View** > **Stats**.
4. Under the **Key Stats** section, check the values for **Capture Rate**, **Max Capture Rate**, **Total**

Dropped, and **Total Captured packets** for the decoder service.



Task 6. Verify Events Meta from Netskope in Investigate View

To verify Netskope events, you must first aggregate the Decoder service into the Concentrator and then go to the **Investigate** > **Events** page to view the Netskope events.

- Add the Decoder Service in the Concentrator
- Verify from the Investigate > Events View

Add the Decoder Service in the Concentrator

1. Log in to the NetWitness Platform.
2. Go to **(Admin)** > **Services**.
3. In the **Services** list, select the **Concentrator** service.
4. Click > **View** > **Config**.

   The Services Config View of the Concentrator is displayed.
5. Select the **Sources** tab.
6. Click and select Available Services.

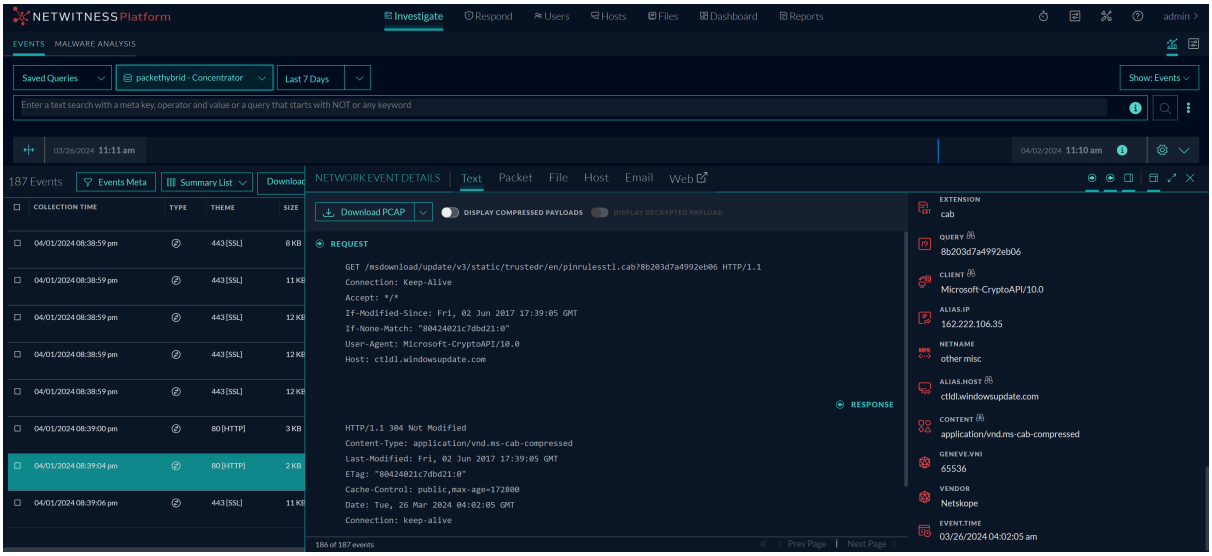
   The Available Services dialog is displayed.
7. Select the **Decoder** service and click **OK**.

   The service authentication dialog box is displayed.

   Note: The services with the Trust Model enabled must be added individually. You are prompted to provide a username and password for the selected service.
8. Enter the Username and Password for the service.
9. Click **OK**.
10. Click **Apply**.

Verify from the Investigate > Events View

1. Go to **Investigate** > **Events**.
2. Select the **Concentrator** Service from the **Services** selection drop-down list.
3. Click  to load the Netskope events data.



Remove Netskope Integration Plugin

If you have Netskope Integration plugin deployed on a policy and no longer want to use it, perform the following steps to delete it.

To remove the Netskope Integration completely, first delete the policy containing the plugin from **Policies** view, and then delete the plugin details on the Decoder host.

IMPORTANT: If you have deployed the plugin using CCM, you must perform steps **1** and **2** procedures. If you have deployed the plugin using NwConsole, you can proceed directly to step 2 and complete the procedure.

- Step 1: Remove the Policy Containing Netskope Integration
- Step 2: Remove the Netskope Plugin Details from Decoder Host

Step 1: Remove the Policy Containing Netskope Integration

1. Go to  **(Configure) > Policies**.
2. In the policies panel, click **Content**.
3. Click **Policies**. The available policies are displayed.
4. Select one or more policies and in the **More Actions** drop-down list in the tool bar, click **Delete**.

   The **Delete Policies** dialog is displayed.
5. To delete the deployed content from the group's services upon deleting the policy, select the option **Delete deployed content from the group's services on policy removal**.

   Note: Removing the policy will delete only the Netskope Configuration details and not the plugin on the Decoder host.
6. Click **Delete** to permanently delete the selected policy.

   Deletion will take immediate effect and the policy will no longer be available in any group.

   Note:
   - You can also delete a policy from the **Policy Details** view. For more information on deleting a policy from the **Policy Details** view, see View a Policy topic.
   - The policy status changes to **Failed** if policy deletion fails for any particular reason.

Step 2: Remove the Netskope Plugin Details from Decoder Host

1. SSH to the Packet Decoder Host.
2. Run the following command to stop the Decoder service:

   ```
   systemctl stop nwdecoder
   ```
3. Navigate to the following path:

   ```
   /etc/netwitness/ng/hosted
   ```
4. Delete the **netskope** folder.

5. Run the following command to start the Decoder service:

   ```
   systemctl start nwdecoder
   ```

6. Navigate to the **Explore** view of the Packet Decoder service.
7. On the left panel, select **decoder** > **hosted** > **netskope** > **instances**.
8. Select the **delete** operation from the drop-down list and click **Send**.

The plugin details are removed from the decoder.

Remove Container

Perform the following to remove the Container:

Step 1. Remove the Container

Remove the Container when upgrading the Netskope CloudTap Image.

a. First stop the Container using the command: `podman stop <container_name>`
b. Next, remove the container using the command: `podman rm -f <container_name>`

Example:

```
podman stop netskope
podman rm -f netskope
```

Step 2. Remove the CloudTap Image

Remove the CloudTap Image using the command: `podman rmi -f <image id>`

Example: `podman rmi -f c4912b553a73`

Note: When you upgrade the plugin, you can download the new supported image. For more information, refer to Deploy Netskope Integration using CCM.

(Optional) NetSkope Container Setup (without plugin support mode)

Overview

This topic highlights the procedure for manually deploying the Netskope Cloud-Tap Container on the NetWitness Decoder cloud instance without plugin support.

Prerequisites

- Deploy and Configure NetWitness Decoder and Concentrator cloud instances using the Netwitness SASE Deployment guide. For more information, refer to the SASE Hybrid Cloud Installation Guide.
- Download and Copy the Netskope Cloud-Tap Stitcher Image to the Decoder host.
  - Download the image example `docker.io/nsteam/cloudtap-stitcher:latest` from Netskope Docker repository.
    - If internet is available on the Decoder host, run the podman pull command:

      - `podman pull docker.io/nsteam/cloudtap-stitcher:latest`
    - If internet is unavailable on the Decoder host, then download the image from a machine with internet as a tar file using `podman save` or `docker save`. Then load that image on the Decoder host using `podman load` or `docker load`:

      - `podman save -o cloudtap-stitcher-latest.tar docker.io/nsteam/cloudtap-stitcher:latest`

      - Compress the tar file using the command: `gzip cloud-stitcher-latest.tar` and copy the image archive to the Decoder host.

      - On the decoder host, run the following command to load the image from the file:

        `podman load -i cloudtap-stitcher-latest.tar.gz`

Perform the following to deploy the NetSkope container setup:

- Task 1. Configure the Container
- Task 2. Create the Container
- Task 3. Capture Interface in the Decoder
- Task 4. Start the Decoder Capture
- Task 5. Start the Stitcher Capture

Task 1. Configure the Container

Perform the following to configure the container

1. Choose a unique Container name for this Decoder instance. For example: `netskope`
2. SSH to the NetWitness Decoder host as a root.
3. Create a folder to share the bucket credentials with the Netskope Container. For example:

   `/etc/netwitness/ng/netskope/bucketcreds`
4. Copy the cloud bucket credentials json file to the newly created folder:

   `/etc/netwitness/ng/netskope/bucketcreds`
5. Create a bookmark folder to be shared with Container to save the progress. For example:

   `/etc/netwitness/ng/netskope/bookmark`

   This path for progress will be available for the NetWitness backup and restoring the process automatically.

Note: The Bucket Authentication mentioned in the steps 3 and 4 is optional if you have already configured GCP / AWS default credentials for the instance.

Task 2. Create the Container

First, SSH to the NetWitness Decoder host to create the container.

Get the Image Id

Run the following command:

```
podman image list
```

Output example:

```
REPOSITORY                          TAG      IMAGE ID       CREATED     SIZE
docker.io/nsteam/cloudtap-          latest   c4912b553a73   11 days     883
stitcher                                                    ago         MB
```

In this example, c4912b553a73 is the Image ID.

Create and Setup the Container

1. Run the following command to create and setup the Container. Replace the fields listed in <> with the relevant values.

   ```
   podman run -itd --name <container_name> --cpus=<max_cpus_for_
   container> -v <netskope_host_credentials_directory_path>:r -v
   <netskope_host_bookmark_path>:<netskope_container_mount_dir>:rw,z
   <image_id>
   ```

   Example: `podman run -itd --name netskope --cpus=2 -v /etc/netwitness/ng/netskope/bucketcreds:r -v /etc/netwitness/ng/netskope/bookmark:/bookmark:rw,z c4912b553a73`

   Note: The Bucket credentials path is optional if you have already configured GCP / AWS default credentials to the instance.

Check the Container Status

Run the following command:

```
podman ps
```

Output example:

```
CONTAINER ID  IMAGE                                COMMAND  CREATED      STATUS         PORTS  NAMES
bfdd05026f6d  docker.io/nsteam/cloudtap-stitcher           0 days ago   Up 12 seconds         Netskope
```

Check the Container Network Interface cni-podman1

Run the following command:

```
ifconfig
builds]# ifconfig
cni-podman1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9000
      inet 10.88.0.1 netmask 255.255.0.0 broadcast 10.88.255.255
      inet6 fe80::98b7:c6ff:fe0c:65f0 prefixlen 64 scopeid 0x20<link>
      ether 9a:b7:c6:0c:65:f0 txqueuelen 1000 (Ethernet)
      ...
```

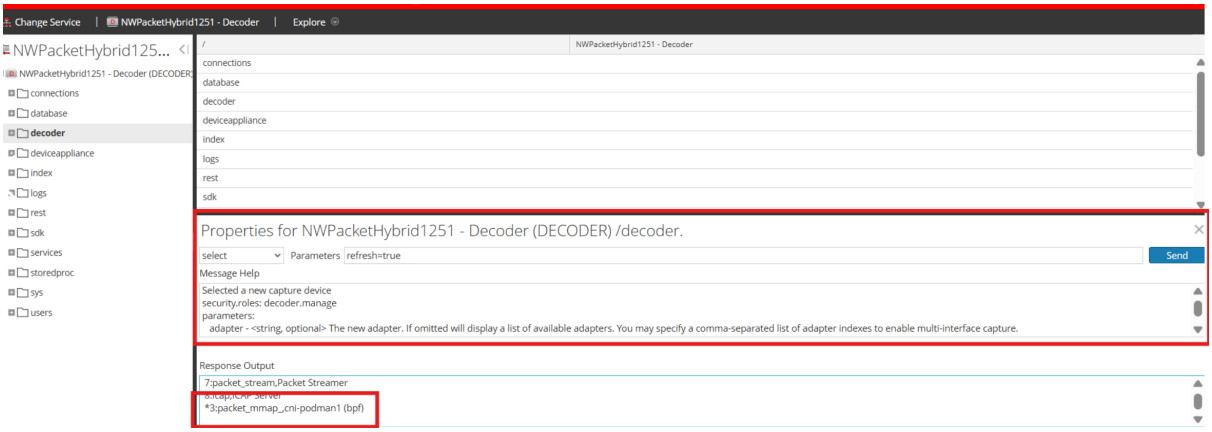
Task 3. Capture Interface in the Decoder

You must select a network adapter **(packet_mmap_,cni-podman1 (bpf))** through which the Decoder captures packets and processes the data.

Perform the following to capture the interface in the decoder:

1. Log in to the NetWitness Platform.
2. Go to  (**Admin**) > **Services**.
3. Select the **Packet Decoder** service and click  > **View** > **Explore**.
4. On the left panel, select decoder and right-click properties.
5. In the **Properties** section for the Decoder service, choose **select** option from the drop-down list, and in the **Parameters** field, type **refresh=true** and click the **Send** button.

   Using the **refresh** option will detect the new capture interfaces available on Decoder. This will add `cni-podman1` interface to the list.



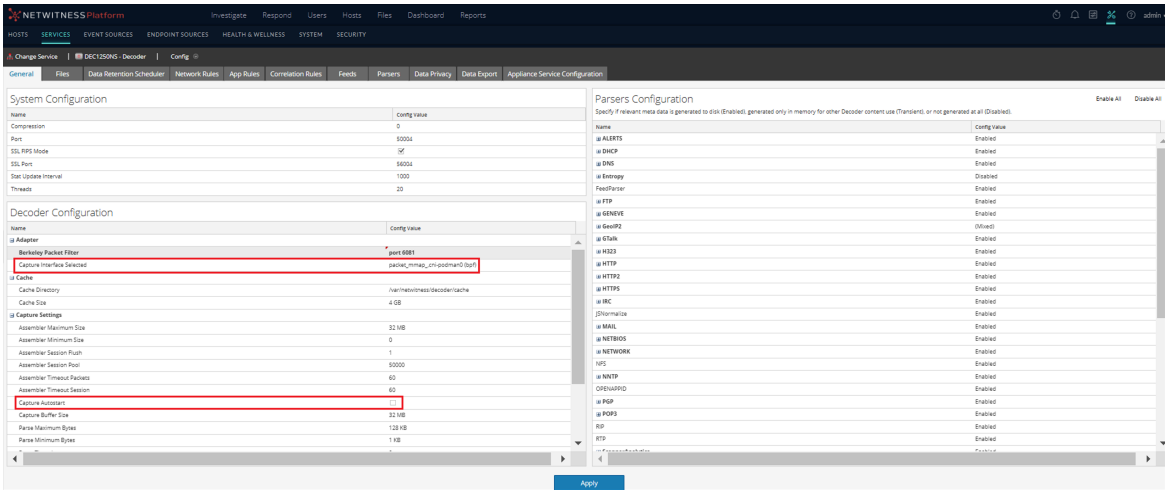
6. Navigate to the **Explore** page of the Decoder service.

   The Configure view for the Decoder service is displayed with the **General** tab open.
7. Under the Decoder Configuration section, do the following:
   a. Configure the BPF to filter port 6081 to capture Geneve traffic.
   b. Set the **Capture Interface Selected** to **(packet_mmap_,cni-podman1 (bpf))** network adapter.

      The decoder uses the **cni-podman1** network interface, monitoring the network traffic flowing through it. The decoder can capture and analyze the packets sent and received by the containers.

   c. Disable the **Capture Autostart** option on the Decoder as the Netskope Container state is independent of the Decoder capture state, and the Decoder auto capture can fail if the capture

interface does not appear.



8. Click Apply to save the changes.

9. To restart the Decoder service, go to the **Services** view, select the Decoder service, and click  > Restart.

10. Confirmation dialog request is displayed. To restart the service, click **Yes**.

Task 4. Start the Decoder Capture

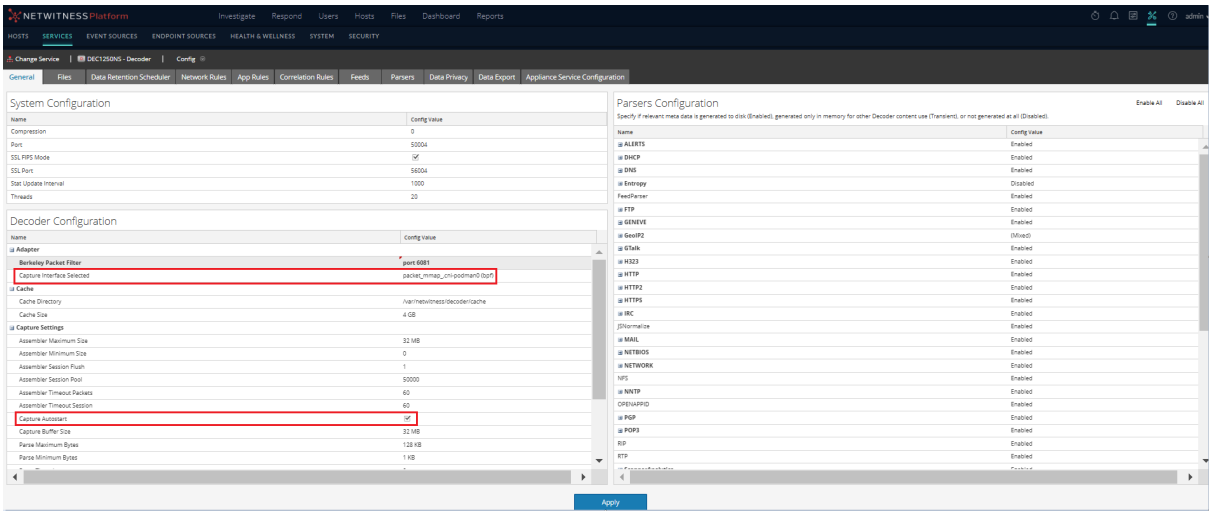
Start the Container

1. Check if the NetSkope Container is running. To start the Netskope Container if it is not running, use the command:

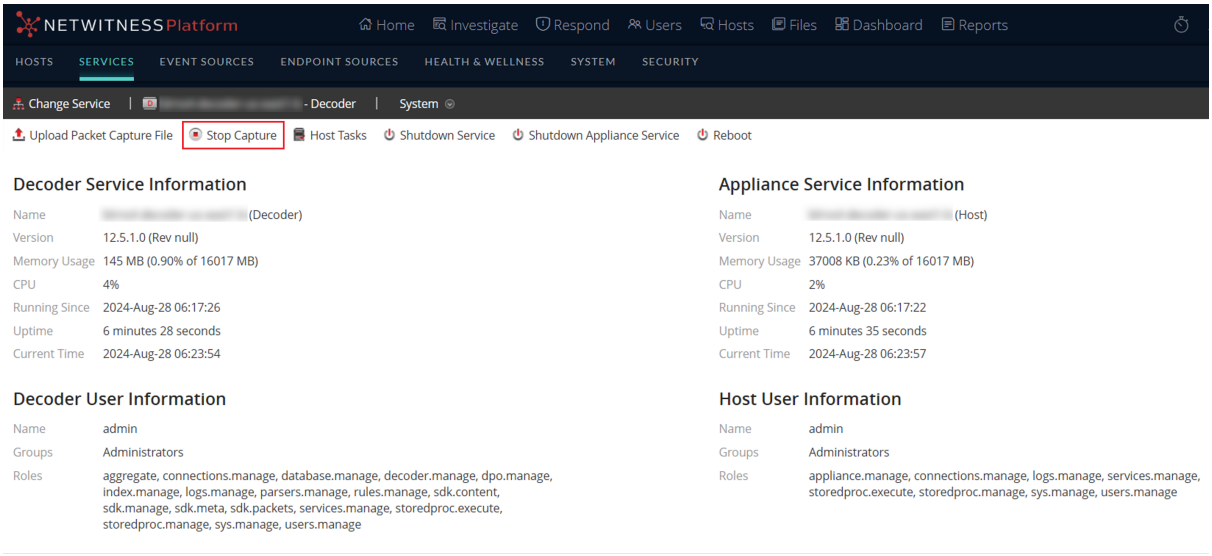
   `podman start <container-name>`

   Example: `podman start netskope`

2. Navigate to **NetWitness Admin Server UI** > **Decoder** > **System** > **Start Capture**. Enable the **Capture Autostart** option to start the Decoder capture on the host container interface `cni-podman1`.



3. (Optional) Navigate to the System view of the Decoder service and check if the Decoder is capturing the data.



Task 5. Start the Stitcher Capture

The Netskope Stitcher program connects to the cloud bucket, downloads the packets from the bucket using the credentials specified, and replays the packets to the host interface of the Container, e.g., `cni-podman1`, where the NetWitness Decoder service can capture packets.

Note: The Bucket credentials config is optional if you have already configured GCP / AWS default credentials to the instance.

Start the Stitcher Capture in Detach Mode (default)

Run the following command to start the Stitcher Capture in Detach mode. Replace the fields listed in <> with the relevant values.

```
podman exec -d <container_name> stitcher -b <bucket_name> --provider
gcp -c <access_credentials_file> --geneve-host host.containers.internal
--with-timestamps --log-progress <bookmark_path> -n --mt --connections
```

Example:

```
podman exec -d netskope stitcher -b nw-ns-traffic-capture --provider
gcp -c /bucketcreds/creds.json --geneve-host host.containers.internal -
-with-timestamps --log-progress /bookmark/ -n --mt --connections
```

Start the Stitcher Capture in Attach Mode (View Logs and Debugging)

Run the following command to start the Stitcher Capture in Attach mode.

```
podman exec <container_name> stitcher -b <bucket_name> --provider gcp -
c <access_credentials_file> --geneve-host host.containers.internal --
with-timestamps --log-progress <bookmark_path> -n --mt --connections
```

Example:

```
podman exec netskope stitcher -b nw-ns-traffic-capture --provider gcp -
c /bucketcreds/creds.json --geneve-host host.containers.internal --
with-timestamps --log-progress /bookmark/ -n --mt --connections
```

Note: When the session terminates, the Stitcher process may still be running in the background in the Container. Stop the Container to stop the Stitcher process completely. Refer the below steps to stop the Stitcher capture.

Host Reboot

After the host reboot, follow the steps below to ensure that the host container interface is available before the Decoder starts the capture. If the interface is unavailable, the Decoder capture will fail.

1. Task 2. Create the Container
2. Task 4. Start the Decoder Capture
3. Task 5. Start the Stitcher Capture

Upgrade the Container

Perform the following to Upgrade the Container:

Step 1. Remove the Container

Remove the Container when upgrading the Netskope CloudTap Image.

a. First stop the Container using the command: `podman stop <container_name>`

b. Next, remove the container using the command: `podman rm -f <container_name>`

Example:

```
podman stop netskope
podman rm -f netskope
```

Step 2. Remove the CloudTap Image

Remove the CloudTap Image using the command: `podman rmi -f <image id>`

Example: `podman rmi -f c4912b553a73`

Step 3. Download the new image (Refer to Prerequisites topic for more information)

Step 4. Task 2. Create the Container

Step 5. Task 3. Capture Interface in the Decoder

Step 6. Task 4. Start the Decoder Capture

Step 7. Task 5. Start the Stitcher Capture

Additional Option

Stop the Stitcher Capture

To stop the Stitcher Capture for maintenance, run the following command:

`podman stop <container_name>`

Example: `podman stop netskope`

Limitations

Since this is a manual process for managing a Container, the User deploying the Container must be aware that the Container's running state is independent of the NetWitness Decoder service state. The following are the possibilities:

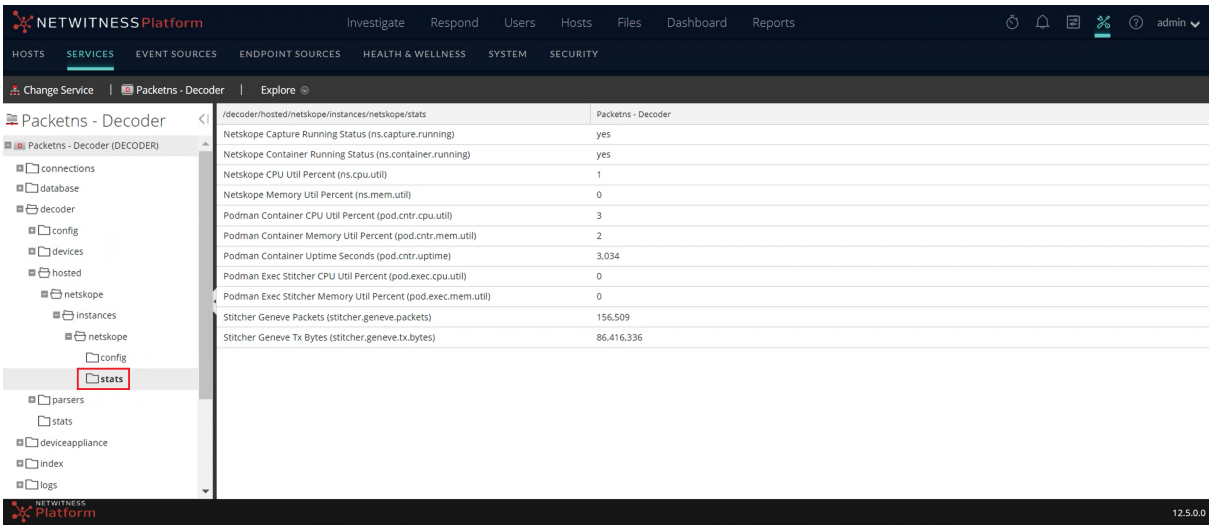
1. The Container will run the Stitcher Capture, but the Decoder may be down for maintenance. This could cause a packet loss as the Container might continue ingesting packets while the Decoder is down.
2. Container to host network interface may not be available if the Container does not start before the Decoder service starts. This may cause the Decoder capture failure as the Container network interface is unavailable for capture.

Plugin Stats

Note: Stats are also subjected to change based on future image releases.

Perform the following to view the plugin stats:

1. Log in to the NetWitness Platform.
2. Go to **(Admin)** > **Services**.
3. Select the **Packet Decoder** service and click > **View** > **Explore**.
4. On the left panel, select **decoder > hosted > netskope > instance > netskope > stats**.



The stats for the Decoder service is displayed.