Troubleshooting Cisco NIR App on Cisco DCNM

This chapter contains the following sections:

Troubleshooting Cisco NIR Common GUI Issues

The following are troubleshooting tips for GUI issues on Cisco NIR app in Cisco DCNM .

- The Cisco NIR app has the ability to display historical data. The specific time duration can be selected from the available calendar to see data within that particular time range.
- The majority of issues will be due to receiving data from the APIs other than what was expected. Opening the **Developer Tools Network** tab and repeating the last action will show the API data received. If the issue is with the APIs, then troubleshooting will need to continue on the backend.
- If the API requests and responses are accurate then check the **Developer Tools Console** tab for any errors.
- After initial installation the application needs time to start. During this time, the GUI may exhibit incomplete or unstable behavior. It is recommended to wait several minutes before starting to use the application.
- Take screenshots just before and just after reproducing an issue. Screenshots along with a full network capture saved as HAR with contents can be used to issue reports. If an issue report has a HAR recording attached then there is a significantly higher chance that the root cause can be identified and resolved quickly.
- If the Cisco NIR GUI page loads to a skeleton template with spinner then this means almost none of the APIs are responding.
- If the Cisco NIR GUI page is taking a while to load fabrics then this means the `fabrics.json` API is not responding or not returning any fabrics.

- If the fabric anomaly score does not agree with reported anomalies, or the node counts are incorrect, then check the `fabricsSummary.json` response for the fabric `anomalyScore` value, and check the `nodes.json` response for the types and counts of nodes reported.
- If the expected fabrics are not shown in the fabric selection dropdown, first verify that they are not included in the `fabrics.json` response entries, then rerun setup and edit the data collection setup configuration to view the state of the configured fabrics. Make sure the appropriate fabrics are enabled and that no errors are reported. This data comes from the `get_nir_fabrics` request.
- For Flow Analytics issues make sure the following requirements are met:
    - The `capability.json` request is made when the GUI loads and returns true. If it returns false, it means the fabric does not support this feature.
    - Navigate to **Application Settings** tab and make sure **Flow Collection** has been enabled, the Management In-Band EPG has been selected, and verify the flow collection filters have been correctly configured.
    - To verify the MOs are using `visore`, navigate to **uni** > **fabric** > **flowcol** to check the configuration and check the classes `telemetrySelector`, `telemetrySubnetFltGrp`, and `telemetrySubnetFilter`.
    - Navigate to **Collection Status** tab and check if the nodes are returning flow telemetry.

Troubleshooting at Cisco NIR App Level

The following techniques are useful for troubleshooting Cisco NIR app on Cisco DCNM.

- To view running service instances, navigate to **DCNM** > **Application** > **NIR** and right-click the settings icon.
- To view capacity usage, navigate to **NIR**, right-click the settings icon and choose **System Status**.
- If there is no data shown in Cisco NIR dashboard, check the **Setup** page. Navigate to **NIR**, right-click the settings icon and choose **Rerun Setup**.
    - If no fabric is enabled, try to enable a fabric from available fabrics.
    - If no fabric is available, check if Cisco DCNM fabric is setup properly.
- To view telemetry data collection status, navigate to **NIR**, right-click the settings icon and choose **Collection Status**. Any red dot indicates that the telemetry data is not streamed. Possibilities for red dot are:
    - Switch is in monitor mode and telemetry configuration may not be available.
    - Switch telemetry configuration CLI may be missing or causing error.

Troubleshooting at Switch Level

The following techniques are useful for troubleshooting switches for Cisco NIR app on Cisco DCNM.

- To check the telemetry connection status:

    - Login to the switch using SSH.
    - Execute the following command and verify status of telemetry connection.

      ```
      switch# show telemetry transport
      ```

- To check the telemetry data collector details:
    - Login to the switch using SSH.
    - Execute the following command

      ```
      switch# show telemetry data collector details
      ```

- To check for any time sync issues from the switches:
    - Login to Cisco DCNM compute node using SSH.
    - Execute the following commands:

      ```
      node# docker ps|grep debug
      node# docker exec -it <<debugcontainerid>> sh
      node# show ntp-time-sync
      ```

      Check if the telemetry data is not synced with NTP server.

Troubleshooting at Services Level

The following techniques are useful for troubleshooting services for Cisco NIR app on Cisco DCNM.

- To check the basic services are running:

1. To check for the required telemetry services are up and running:
    - Login to the master node and execute the following command.

      ```
      node# docker service Is
      ```

    - Verify the services **apiserver, correlator, eventcollector, postprocessor, utr, redictor, scheduler, kafka, zookeeper**, and **elastic 6.1.4** are available and running.

2. To check all the topics are pre-created and are in place:
    - Login to the compute node and execute the following command.

      ```
      node# docker ps|grep debugtools
      ```

    - Get the container ID and execute the following command to debug the container.

      ```
      node# docker exec -it <<debugcontainerid>> sh
      ```

    - Execute the following command in shell prompt.

      ```
      sh# show kafka topics
      ```

- Verify the topics **cisco_nir-events, cisco_nir-operational, cisco_nir-stats-json**, and **cisco_nir-sw-telemetry-utr-out** are available.

3. To check all the indicies are present:
   - Login to the compute node and execute the following command.

     ```
     node# docker ps|grep debugtools
     ```

   - Get the container ID and execute the following command to debug the container.

     ```
     node# docker exec -it <<debugcontainerid>> sh
     ```

   - Execute the following command in shell prompt.

     ```
     sh# show elastic indices
     ```

   - Verify the indicies **cisco_nir-enrich*, cisco_nir-statsdb*, cisco_nir-anomalydb, cisco_nir-fabrics*, cisco_nir-eventsdb, Cisco_nir-operdb, cisco_nir-recourcecollectdb***, and **cisco_nir-recourcescores*** are available.

4. Other useful docker commands to check the basic services are running:
   - Login to the compute node and execute the following command.

     ```
     node# docker ps
     ```

   - To get the memory and CPU statistics of containers running in compute node:

     ```
     node# docker estats
     ```

   - Get the container ID to view the instant logs of any running container:

     ```
     node# docker logs <<containerid>> -f
     ```

   - To view the logs of any running service in the master mode:

     ```
     node# docker service logs <<servicecontainerid>> -f
     ```

- To check the individual services are running:
  - Telemetry receiver stage.
  - Postprocessor stage.
  - Event collector, Predictor and correlator stage.

Troubleshooting at UTR Telemetry Receiver Level

The following techniques are useful for troubleshooting UTR telemetry receiver for Cisco NIR app on Cisco DCNM.

1. Make sure the Debug Tools app is available and running to debug a service.
   - Login to the compute node and execute the following command.

```
node# docker ps|grep debug
```

- Get the container ID and execute the following command to debug container.

```
node# docker exec -it <<debugcontainerid>> sh
```

2. Execute the following command to verify that the data is flowing through UTR service.

```
sh-4.2# show kafka data-utr-out --help
Usage: show kafka data-utr-out [OPTIONS]

Options:
   -n, --node-name TEXT
   -o, --offset [latest|earliest]
   --help Show this message and exit.
```

3. Other useful commands to check UTR telemetry receiver data.

```
sh# show kafka data-utr-out -n leaf0 -o latest
```

```
sh# show kafka data-utr-out -n leaf0 -o latest | grep -i "show vlan
summary"
```

Troubleshooting at Post-Processor Level

The following techniques are useful for trubleshooting post-processor for Cisco NIR app on Cisco DCNM.

1. Make sure the Dubug Tools app is running to debug a service.
   - Login to the compute node and execute the following command.

   ```
   node# docker ps|grep debug
   ```

   - Get the container ID and execute the following command to debug the container.

   ```
   node# docker exec -it <<debugcontainerid>> sh
   ```

2. Execute the following command to verify that the data is flowing through post-processor service.

```
sh-4.2# show kafka data-processed --help
Usage: show kafka data-processed [OPTIONS]

Options:
   -r, --stat-type [hardware|protocol|environmental|config|operational|interface|To be
removed]
   -n, --node-name TEXT
   -f, --fabric-name TEXT
   -o, --offset [latest|earliest]
   --help Show this message and exit.
```

3. Execute the following command to verify that the fabric node is flowing through post-processor service.

```
sh-4.2# show kafka data-fabricnodes --help
Usage: show kafka data-fabricnodes [OPTIONS]

Options:
   -n, --node-name TEXT
   -f, --fabric-name TEXT
   -o, --offset [latest|earliest]
   --help Show this message and exit.
```

4. Other useful commands to troubleshoot post-processor

   ```
   sh# show kafka data-processed -r environmental -n leaf0 -o earlie
   VXLAN-1
   ```

   ```
   sh# show kafka data-processed -r environmental -n leaf0 -o earlie
   VXLAN-1 | grep -l "powerDrawn"
   ```

Troubleshooting at Event Collector, Predictor, and Correlator Services Level

The following techniques are useful for troubleshooting collector, predictor, and correlator services for Cisco NIR app on Cisco DCNM.

- Execute the following command to verify the processed data is available in time series database.

  ```
  sh-4.2# show nir es stats
  Usage: show nir es stats [OPTIONS] COMMAND [ARGS]...

  Options:
     --help Show this message and exit.

  Commands:
     aggregated
     anomaly
     raw

  sh-4.2# show nir es stats raw --help
  Usage: show nir es stats raw [OPTIONS]

  Options:
     -r, --stat-type [utilization|l2_protocol|interface]
     -n, --node-name TEXT
     -f, --fabric-name TEXT
     -s, --start-time [now-15m|now-1h|now-6h|now-1d|now-1w|now-1M]
     -p, --pretty
     --help Show this message and exit.
  ```

- Other useful commands to troubleshoot collector, predictor, and correlator services.
  - Execute the following commands to check for the available fabrics.

    ```
    # show nir es fabrics
    ```

    ```
    # show nir es fabrics -f Simulation
    ```

  - Execute the following commands to check for the available fabrics nodes.

    ```
    # show nir es fabrics-nodes -f Simulation
    ```

    ```
    # show nir es fabrics-nodes -f Simulation -n N9Kv-2
    ```

  - Execute the following command to verify the Statistics data.

    ```
    # show nir es stats raw -f Simulation -n N9Kv-1 -r utilization
    ```

  - Execute the following command to verify the Aggregated data.

    ```
    # show nir es stats aggregated -f Simulation -n N9Kv-2 -r config
    ```

- Execute the following command to verify the Anomaly data.

  ```
  # show nir es stats anomaly -f Simulation -n N9Kv-1 -r config
  ```

Debugging Cisco NIR App on Cisco DCNM

The following techniques are useful for troubleshooting Cisco NIR app on Cisco DCNM.

- To fetch the software telemetry logs from the devices streaming the data execute the following command.

  ```
  N9k-ToR2# show tech-support telemetry > ts_telemetry.log
  ```

- To fetch the logs for single node Cisco DCNM server execute the following command.

  ```
  # appmgr afw fetch-logs Cisco:NIR
  ```

- To fetch the logs for Cisco DCNM server with HA (active-standby):
  - Login to the active Cisco DCNM server.
  - Execute the following command.

    ```
    # appmgr afw fetch-logs Cisco:NIR:2.0 <<password>>
    ```

- To collect the logs for appliations such as Kafka and Elastic execute the following command.

  ```
  # appmgr afw fetch-logs Cisco:Kafka
  ```

- To collect the logs for all applications execute the following command.

  ```
  # appmgr techsupport
  ```