

SMF Interface for Metrics

- Feature Description, on page 1
- How it Works, on page 2
- Configuring Metrics Collection, on page 2

Feature Description

You can a monitor wide range of application and system statistics, and key performance indicators (KPI) within the infrastructure. KPIs are useful to gain insight into the overall health of the environment. Statistics offer a simplified representation of the configurations and utilization-specific data.

The integrates with Prometheus, a third-party monitoring and alerting solution to capture and preserve the performance data. This data is reported as statistics and can be viewed in the web-based dashboard. Grafana provides a graphical or text-based representation of statistics and counters, which the Prometheus database collects. The Grafana dashboard projects a comprehensive set of quantitative and qualitative data that encourages you to analyze metrics in the reporting tool of your choice and take informed decisions.

By default, the monitoring solution is enabled, which indicates that Prometheus continually monitors your environment and the Prometheus data source is associated with Grafana. You must have the administrative privileges to access Grafana. However, to view a specific dashboard, run the Prometheus queries. The queries are available in the built-in and custom format.

The following snapshot is a sample of the Grafana dashboard.

Figure 1: Grafana Dashboard



How it Works

KPIs constitute of metrics, such as statistics and counters. These metrics represent the performance improvement or degradation. By default, Prometheus is enabled on the system where is deployed, and configured with Grafana. Prometheus dynamically starts monitoring the data sources that are available on the system. For new dashboard panels, execute queries in Prometheus.

For more information about Prometheus, consult the Prometheus documentation at https://prometheus.io/docs/introduction/overview/.

Configuring Metrics Collection

The labels of each metrics are classified into the following three categories:

- Production
- Debug
- Granular

All the application metrics are controlled through the CLI command for performance optimization.

To collect the necessary metrics and labels, use the following sample configuration:

```
config
  infra metrics verbose { service | protocol | load-balancer | application
 } [ level { debug | off | production | trace } | metrics metrics_name [
granular-labels label_name| level { debug | off | production | trace } |
pod pod_name | level { debug | off | production | trace }  ] ]
   end
```

NOTES:

- If the metrics verbosity is not configured, then the default verbosity level for pod type is as follows.
  - LoadBalancer = Production
  - Protocol = Trace
  - Service = Trace
  - Application = Debug
- The order of the level for verbose metrics is in the following priority order:
  - **metrics [[metrics_name] level [production|debug|trace|off]**: [Priority 1]
  - **pod [[pod_Name]] level [production | debug | trace | off]]** [Priority 2]
  - **level [production | debug | trace | off]** [Priority 3]
- **infra metrics verbose { service | protocol | load-balancer | application }**: Enable the metric collection. This configuration helps to collect the required application metrics and labels. By default, this command captures the debug labels of metrics.
- **level { debug | off | production | trace }**: Specify the application metrics category to capture the required application metrics and labels.
  - **debug**: Capture all the labels that are classified as production and debug categories. This option is the default configuration.
  - **off**: Disable the application level metrics collection.

    For example, configuring the **infra metrics verbose application smf_service_stats level off** command disables the smf_service_stats application metrics.
  - **production**: Capture the labels that are classified as production category.
  - **trace**: This option is not supported for application metrics. If this option is configured, the treats this option as **debug**.
- If production and debug classification is empty for a metrics, then all the labels except granular-labels (if configured) are classified as debug.
- **metrics** *metrics_name*: Specify the metrics name to capture only the labels that correspond to the given metrics. The metric-level configuration takes precedence over the application-level configuration. If the metrics level is not configured, the labels are captured at the application level.
- **granular-labels**: Capture only the granular labels. By default, this option is disabled.

  If a granular label is required for KPI, then that label must be configured. For example, to capture dnn labels of smf_service_stats metrics, you must configure the following CLI command:

```
infra metrics verbose application metrics smf_service_stats level debug
 granular-labels [ dnn ]
```

Configuration Example

The following is an example configuration to enable only production level for all the application metrics.

```
infra metrics verbose application level production
```

The following is an example configuration to enable production level for smf_service_stats application metrics and debug level for all other application metrics.

```
infra metrics verbose application smf_service_stats level production
```

The following is an example configuration to enable debug level for smf_service_stats application metrics along with granular labels and production level for all other application metrics.

```
infra metrics verbose application level production smf_service_stats level
 debug granular-labels [ dnn ]
```

The following is an example configuration to enable production level for smf_service_stats application metrics along with granular labels and debug level for all other application metrics.

```
infra metrics verbose application smf_service_stats level production
granular-labels [ dnn ]
```

The following is an example configuration to disable smf_service_stats application metrics and debug level for all other application metrics.

```
infra metrics verbose application smf_service_stats level off
```

The following is an example configuration to configure NSSAI labels of smf_service_stats metrics.

```
infra metrics verbose application metrics smf_service_stats level debug
granular-labels [ snssai ]
```

Note The NSSAI statistics are not pegged without configuring the NSSAI label in the granular-labels configuration.

Configuration Verification

To verify the configuration, use the following show command:

```
show running-config infra metrics verbose application
```

The following are example outputs of the **show running-config infra metrics verbose application** command.

```
[smf] smf# show running-config infra metrics verbose application
infra metrics verbose application
 metrics smf_service_stats
  level production
  granular-labels [ dnn ]
 exit
exit
```

The preceding output indicates that the configuration to capture production labels for smf_service_stats application metrics along with granular labels and debug levels of all other application metrics is enabled.

```
[smf] smf# show running-config infra metrics verbose application
 infra metrics verbose application
  level production
```

```
  metrics smf_service_stats
   level debug
   granular-labels [ [dnn] ]
  exit
exit
```

The preceding output indicates that the configuration to capture debug labels for smf_service_stats application metrics along with granular labels and production level of all other application metrics is enabled.

To verify the slice information on procedure and session statistics, use the following show command:

```
show running-config infra metrics verbose application
infra metrics verbose application
 metrics smf_service_stats
  level debug
  granular-labels [ snssai ]
 exit
```