Designer Private Edition Guide

Observability in Designer

1/26/2025

Contents

Learn about the logs, metrics, and alerts you should monitor for Designer.

Related documentation:

-
-
-

RSS:

- For private edition

Monitoring

Private edition services expose metrics that can be scraped by Prometheus, to support monitoring operations and alerting.

- As described on Monitoring overview and approach, you can use a tool like Grafana to create dashboards that query the Prometheus metrics to visualize operational status.
- As described on Customizing Alertmanager configuration, you can configure Alertmanager to send notifications to notification providers such as PagerDuty, to notify you when an alert is triggered because a metric has exceeded a defined threshold.

The services expose a number of Genesys-defined and third-party metrics. The metrics that are defined in third-party software used by private edition services are available for you to use as long as the third-party provider still supports them. For descriptions of available Designer metrics, see:

- Designer Application Server metrics
- Designer metrics

See also System metrics.

Desginer and DAS generate application related metrics at the `/metric` API in the standard Prometheus client format.

> **Important**
>
> In addition to the metrics listed in the DES and DAS metrics topics, you can also obtain infrastructure related metrics by installing standard Prometheus clients in the Kubernetes cluster.

Enable monitoring

To enable monitoring you must configure the various Prometheus related options. For more details on these various options, refer to the *Designer deployment settings* and *DAS deployment settings* sections in the Configure Designer topic.

Service	CRD or annotations?	Port	Endpoint/ Selector	Metrics update interval
Designer Application Server	ServiceMonitor	8081	See selector details on the Designer Application Server metrics and alerts page	10 seconds
Designer	ServiceMonitor	8888	See selector details on the Designer metrics and alerts page	10 seconds

Configure metrics

The metrics that are exposed by the DES and DAS services are available by default. No further configuration is required in order to define or expose these metrics. You cannot define your own custom metrics.

The Metrics pages linked to above show some of the metrics the DES and DAS services expose. You can also query Prometheus directly or via a dashboard to see all the metrics available from the DES and DAS services.

What do Designer metrics monitor?

The exposed DES and DAS metrics help you monitor a number of data points that are important in a production environment. For more details on the individual metrics, refer to the Metrics pages.

Alerting

Private edition services define a number of alerts based on Prometheus metrics thresholds.

> **Important**
>
> You can use general third-party functionality to create rules to trigger alerts based on metrics values you specify. Genesys does not provide support for custom alerts that you create in your environment.

For descriptions of available Designer alerts, see:

- Designer Application Server alerts
- Designer alerts

Configure alerts

Private edition services define a number of alerts by default (for Designer, see the pages linked to above). No further configuration is required.

Enable alerts in Designer

To enable alerts in Designer, use either of the following methods:

Method 1: Enable Prometheus alerts in the values.yaml file.

```
designer:
    prometheus:
        alerts:
            enabled: true # this will be false by default.
```

Method 2: Find out the active deployment color and execute the below command in the corresponding deployment:

```
helm upgrade --install designer-blue -f designer-values.yaml designer-9.0.xx.tgz --
set designer.deployment.strategy=blue-green --set
designer.prometheus.alerts.enabled=true
```

Disable alerts in Designer

To disable or delete alerts, use either of the following methods:

Method 1: Disable Prometheus alerts in the values.yaml file.

```
designer:
    prometheus:
        alerts:
            enabled: false # this will be false default.
```

Method 2: Pass the below parameter along with the Helm upgrade command.

```
helm upgrade --install designer-blue -f designer-values.yaml designer-9.0.xx.tgz --
set designer.deployment.strategy=blue-green --set
designer.prometheus.alerts.enabled=false
```

Enable alerts in DAS

To enable alerts, use either of the following methods:

Method 1: Enable Prometheus alerts in the values.yaml file.

```
das:
    prometheus:
```

```
        alerts:
            enabled: true # this will be false default.
```

Method 2: Pass the below parameter along with the Helm upgrade command.

```
helm upgrade --install designer-das-blue -f designer-values.yaml designer-
das-9.0.xx.tgz --set das.deployment.strategy=blue-green --set
das.prometheus.alerts.enabled=true
```

Disable alerts in DAS

To disable or delete alerts, use either of the following methods:

Method 1: Disable Prometheus alerts in the values.yaml file.

```
das:
    prometheus:
        alerts:
            enabled: false # this will be false default.
```

Method 2: Pass the below parameter along with the Helm upgrade command.

```
helm upgrade --install designer-das-blue -f designer-values.yaml designer-
das-9.0.xx.tgz --set das.deployment.strategy=blue-green --set
das.prometheus.alerts.enabled=false
```

Update alert parameters

The following alert parameters can be updated:

- Alert Threshold (ALERT_PARAMETER_NAME: threshold)
- Alert Interval (ALERT_PARAMETER_NAME: interval)
- Alert Severity (ALERT_PARAMETER_NAME: AlertPriority)

Perform the following steps to update the above alerts:

1. Refer to the list of alerts and identify the name of the alert you want to update or modify.
2. Update the alert by adding a parameter in the below format in the values.yaml file:

```
designer:
     prometheus:
         alerts:
             :
                 :
                 :
```

For example, consider the CPU utilization alert. The alert name is CPUUtilization with a default threshold of 75, severity set to CRITICAL and interval set to 180s. To modify its threshold to 80, severity to HIGH, and interval to 120 seconds, you will have to make the following changes in the values.yaml file:

```
designer:
```

```
prometheus:
    alerts:
        CPUUtilization:
            threshold: 80
            interval: 120
            AlertPriority: HIGH
```

> **Important**
>
> Though the ability to create custom alerts and some dashboards are packaged with the Designer Helm charts, these are not documented and are not supported as of now.

Logging

Refer to the Logging topic for information on configuring logging for the DES and DAS services.