Genesys Customer Experience Insights Private Edition Guide

Observability in Genesys Customer Experience Insights

1/29/2026

Contents

Learn about the logs, metrics, and alerts you should monitor for Genesys Customer Experience Insights.

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 Genesys Customer Experience Insights metrics, see:

- Genesys CX Insights metrics
- Reporting and Analytics Aggregates metrics

See also System metrics.

The metrics described in this document pertain to the internal monitoring of Genesys CX Insights (GCXI) and Reporting and Analytics Aggregates (RAA) performance and containers, and are intended for administrator use only. These metrics are distinct from the metrics which GCXI provides for monitoring contact center activity, which are described in Work with Genesys CX Insights Reports

About monitoring in GCXI

GCXI container provides TCP metrics endpoint (Prometheus format) on port 8180, and HTTP endpoint at: **/gcxi/monitor/metrics**, for example https://gcxi-22-30.genhtcc.com/gcxi/monitor/metrics.

About monitoring in RAA

In RAA, the **Health port** retrieves the health status. It can take as much as three minutes for RAA to analyze Health status.

In addition to health metrics, RAA provides other metrics that capture statistical information and other data extracted using the **Metrics port**. The first time RAA scrapes this metric data, it aggregates data beginning from the the earliest data available. On subsequent occasions, RAA aggregates the data over the time period that has elapsed since the previous scrape. RAA reads this data from a local file that contains only two or three scrape intervals, which speeds the process; the aggregation of this statistical information typically takes only a few seconds.

Enable monitoring

Monitoring is not enabled by default for GCXI or RAA.

Enable GCXI monitoring

The GCXI Helm Chart provides a standard ServiceMonitor object for integration with Prometheus Operator, in gcxi-service-monitor.yaml.

The following Helm Chart values indicate whether to deploy ServiceMonitor object for integration with Prometheus Operator:

```
gcxi.deployment.deployServiceMonitor = false|true
```

```
gcxi.ports.worker.metrics = 8180 (default)
container port for metrics endpoint
```

Enable RAA monitoring

RAA makes metrics available with the help of a sidecar container.

1. To start the sidecar container, you specify the **raa.statefulset.containers.monitor** element, including ports for metrics and/or health endpoints, in the **values.yaml** file. Ensure that the value of **raa.env.STAT_SCRAPE_INTERVAL** (in seconds) exceeds the scrape interval configured in Prometheus for the metrics port.
   For example:

```
raa: ...
  env:
    STAT_SCRAPE_INTERVAL: 15
  ...
  statefulset:
  ...
    containers:
    ...
      monitor:
        name: "{{$.Chart.Name }}-monitor"
        ...
        metrics:
          portName: "metrics"
```

```
                    containerPort: "9100"

                  health:
                    portName: "health"
                    containerPort: "9101"
                ...
```

2. If Prometheus Operator is configured in your Kubernetes environment, then you can configure Prometheus using the parameters **PodMonitor** and **PrometheusRule** in your **values.yaml** file. When configuring **interval** and **scrapeTimeout** for the health port, keep in mind that health check usually takes around a minute.
   For example, set the following values:

```
raa:
  ...
  podMonitor:
    name : "{{ tpl $.Values.raa.serviceName $ }}-monitor"
    podMetricsEndpoints:
      - port: "{{ $.Values.raa.statefulset.containers.monitor.metrics.portName }}"
        path: "/{{ tpl $.Values.raa.statefulset.containers.monitor.metrics.portName
$ }}"
        interval: "{{ $.Values.raa.env.STAT_SCRAPE_INTERVAL }}s"
      - port: "{{ tpl $.Values.raa.statefulset.containers.monitor.health.portName $
}}"
        path: "/{{ tpl $.Values.raa.statefulset.containers.monitor.health.portName $
}}"
        interval: 4m
        scrapeTimeout: 3m
    podTargetLabels:
      - service
      - servicename

  prometheusRule:
    name : "{{ tpl $.Values.raa.serviceName $ }}-alerts"
    alerts:
      health:
        for: "30m"
        labels:
          service: "{{ $.Values.raa.namespace }}"
          component: "{{ $.Chart.Name }}"
          release: "{{ tpl $.Values.raa.serviceName $ }}"
          severity: "severe"
        annotations:
          summary: RAA is unhealthy for more than hour
          description: |-
            Pod {{ "{{ $labels.pod }}" }} reports about unhealthy RAA for more than
hour already.
      error:
        labels:
          service: "{{ $.Values.raa.namespace }}"
          component: "{{ $.Chart.Name }}"
          release: "{{ tpl $.Values.raa.serviceName $ }}"
          severity: "warning"
        annotations:
          summary: RAA error detected
          description: |-
            Pod {{ "{{ $labels.pod }}" }} reports about new RAA errors detected.
      longAggregation:
        thresholdSec: 300
        labels:
          service: "{{ $.Values.raa.namespace }}"
          component: "{{ $.Chart.Name }}"
          release: "{{ tpl $.Values.raa.serviceName $ }}"
          severity: "warning"
        annotations:
          summary:  Aggregation query is pretty slow
          description: |-
            Pod {{ "{{ $labels.pod }}" }} reports that query {{ "{{
$labels.hierarchy }}-{{ $labels.level }}--{{ $labels.mediaType }}" }}
            was running more than {{
$.Values.raa.prometheusRule.alerts.longAggregation.thresholdSec }} seconds.
```

Monitoring summary

Service	CRD or annotations?	Port	Endpoint/ Selector	Metrics update interval
Genesys CX Insights	ServiceMonitor	8180	See selector details on the Genesys CX Insights metrics and alerts page	15 minutes
Reporting and Analytics Aggregates	PodMonitor and PrometheusRule	metrics: 9100, health: 9101	See selector details on the Reporting and Analytics Aggregates metrics and alerts page	metrics: several seconds, health: up to 3 minutes

Configure metrics

The metrics that are made available by these services are available by default. No further configuration is required to define or make available these metrics. You cannot define your own custom metrics.

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 Genesys Customer Experience Insights alerts, see:

- Genesys CX Insights alerts
- Reporting and Analytics Aggregates alerts

Configure alerts

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

The alerts are defined as **PrometheusRule** objects in a **prometheus-rule.yaml** file in the Helm charts. As described above, Genesys Customer Experience Insights does not support customizing the alerts or defining additional **PrometheusRule** objects to create alerts based on the service-provided metrics.

Set the severity levels and some thresholds for alerts by specifying various **raa.prometheusRule.alerts.*** parameters in the **values.yaml** file. See RAA alerts for details.

Logging

For more information, see Logging