Genesys Customer Experience Insights Private Edition Guide

Metrics

2/5/2026

Contents

Learn about Prometheus metrics you should monitor for Genesys Customer Experience Insights (GCXI).

Related documentation:

-
-
-
-

RSS:

- For private edition

The metrics described on this page pertain to the internal monitoring of GCXI and 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 Historical Reporting with Genesys CX Insights

GCXI metrics

GCXI provides internal monitoring metrics through a Prometheus endpoint on port 8180.

See **gcxi-service-monitor.yaml**:

```
{{ if eq (.Values.gcxi.deployment.deployMain | default false) true }}
{{- if .Values.gcxi.deployment.deployServiceMonitor -}}
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  labels:
    {{- include "gcxi.labels" . | nindent 4 }}
  name: gcxi-servicemonitor{{ template "deploymentCode" . }}
  namespace: {{ .Values.gcxi.app.namespace }}
spec:
  selector:
    matchLabels:
      gcxi/deployment-code: gcxi{{ template "deploymentCode" . }}
  endpoints:
  - interval: 30s
    port: metrics
{{- end }}
{{- end }}
```

RAA metrics

metric name	labels	description
gcxi_raa_health_level		a health status metric extracted using separate port of monitor container. The metric value is a sum of values from two different health-checks: • a database based health-check results 2 when RAA is working or in maintenance (have received STOP and waits for START command from GIM) and 0 when RAA is not working according to this check. • a local health-check results 1 when RAA is processing aggregation requests according to local health files and 0 otherwise.
gcxi_raa_command_count	cmd	amount of commands received from GIM since previous scrape moment. Label reflects name of the command. The supported commands are: START, QUIT, EXIT, UPDATE_CONFIG, REAGGREGATE
gcxi_raa_dispatch_count		amount of dispatch (moving aggregation requests from AGR_NOTIFICATION to PENDING_ARG) events since previous scrape moment. Dispatch event occurs usually each 15 seconds. Such events are used for aggregation health check based on local files.
gcxi_raa_heartbeat_count	version	amount of heartbeats since previous scrape moment. Usually heartbeat performed once per 5 minutes and used for health check based on local files. Label is just a current RAA version.
gcxi_raa_relaunched_count		amount of RAA relaunches happened since previous scrape moment. Aggregation process usually exits on error. GIM always sends START command each 15 minutes during aggregation period. Such a command causes RAA relaunch after exit.

metric name	labels	description
gcxi_raa_launched_count	version	amount of RAA launches happened since previous scrape moment
gcxi_raa_error_count		amount of errors registered since previous scrape moment
gcxi_raa_notification_count	fact	amount of fact change notifications received from GIM since previous scrape moment
gcxi_raa_notification_period_ms	fact	sum of changed fact periods in notifications received from GIM since previous scrape moment
gcxi_raa_notification_delay_ms	fact	sum of fact notification delays in notifications received from GIM since previous scrape moment. Notification delay is measured as difference between moment of notification and start of changed period.
gcxi_raa_aggregated_count	hierarchy, level, mediaType	amount of aggregations completed by RAA since previous scrape moment. The data is grouped by aggregation hierarchy name, materialised level (usually SUBHOUR, HOUR, DAY, MONTH) and media type (Online, Offline)
gcxi_raa_aggregated_period_ms	hierarchy, level, mediaType	sum of periods aggregated by RAA since previous scrape moment
gcxi_raa_aggregated_duration_ms	hierarchy, level, mediaType	total duration of aggregations completed by RAA since previous scrape moment.
gcxi_raa_aggregated_delay_ms	hierarchy, level, mediaType	sum of delays for aggregations completed by RAA since previous scrape moment. Aggregation delay is measured as difference between moment of aggregation compete and start of aggregation range.
gcxi_raa_purged_count	table	amount of records purged by RAA since previous scrape moment. The data is grouped by purged table name
gcxi_raa_purged_duration_ms	table	total purging duration since previous scrape moment.

Metrics for Alarming

- **gcxi_raa_health_level** — a zero value for a recent period (several scrape intervals) is an alarm signal

that RAA not works.

- **gcxi_raa_error_count** — a non zero value is a signal for logs investigation between the corresponding scrape intervals.
- **gcxi_raa_aggregated_duration_ms/gcxi_raa_aggregated_count** — reflects an average duration of an aggregation query specified by the hierarchy, level, mediaType labels. Such the formula suggested since several query events can be registered between scrape intervals. It makes sense to get notifications when the formula value exceeds warning-threshold because the plan of the SQL query of aggregation can be found in logs (written with the WARNING log level). If query execution is longer than the value of deadlock-threshold then RAA cause an error which triggers the metric above.

Enabling monitoring using Prometheus Endpoints

A monitor side car optional container is responsible for exposing RAA metrics; it is disabled by default.

The following example of a **myvalues.yaml** file enables monitoring:

```
raa:
...
  env:
    STAT_SCRAPE_INTERVAL: 15
  ...
  statefulset:
  ...
    containers:
    ...
      monitor:
        name: "{{ $.Chart.Name }}-monitor"
        ...
        metrics:
          portName: "metrics"
          containerPort: "9100"

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

Health port is responsible for retrieving of health status. Health status analyses can take up to 3 minutes. Ensure that correct values of scrape interval and timeout are configured for this port in Prometheus or other monitoring tool.

The rest of metrics contains statistical information and extracted using the metrics port. Their values are simply aggregated for period since the previous scrape or from start moment (for scraping at first time) from a local file. The stats aggregation takes not more that several seconds. It is because the local source file always lists events having place only 2-3 scrape intervals behind. Therefore a scrape interval value configured in Prometheus (or other monitoring tool) for this port should be duplicated in seconds here using the raa.env.STAT_SCRAPE_INTERVAL value.

Support of Prometheus CRDs

If Prometheus Operator configured in k8s environment then it is possible to enable PodMonitor and PrometheusRule using values.yaml:

```
raa:
  ...
```

```
  podMonitor:
    name : "{{ tpl $.Values.raa.serviceName $ }}-monitor"
    podMetricsEndpoints:
      - port: "{{ $.Values.raa.containers.monitor.metrics.portName }}"
        path: "/{{ tpl $.Values.raa.containers.monitor.metrics.portName $ }}"
        interval: "{{ $.Values.raa.env.STAT_SCRAPE_INTERVAL }}s"
      - port: "{{ tpl $.Values.raa.containers.monitor.health.portName $ }}"
        path: "/{{ tpl $.Values.raa.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.
```