Genesys Customer Experience Insights Private Edition Guide

Deploy Reporting and Analytics Aggregates

6/7/2026

Contents

Learn how to deploy Reporting and Analytics Aggregates (RAA) into a private edition environment.

Related documentation:

-
-
-
-

RSS:

- For private edition

Assumptions

- The instructions on this page assume you are deploying the service in a service-specific namespace, named in accordance with the requirements on Creating namespaces. If you are using a single namespace for all private edition services, replace the namespace element in the commands on this page with the name of your single namespace or project.
- Similarly, the configuration and environment setup instructions assume you need to create namespace-specific (in other words, service-specific) secrets. If you are using a single namespace for all private edition services, you might not need to create separate secrets for each service, depending on your credentials management requirements. However, if you do create service-specific secrets in a single namespace, be sure to avoid naming conflicts.

Prepare to deploy RAA

Configure the following resources to support the RAA pod:

- Database connection details specified as `raa.env.GCXI_GIM_DB__JSON` value, or as a Kubernetes or vendor-specific secret. For more information, see Genesys Info MartGenesys info Mart secret volume.
- Config volume — Ensure that space is provisioned and available for RAA to mount the Config Volume to the POD containers (using specific helm values, or separately from the RAA helm chart, with the help of Kubernetes or vendor-specific tools).
- Health volume — Ensure that space is provisioned and available for RAA to mount the Health Volume mounting to the POD containers (using specific helm values, or separately from the RAA helm chart, with the help of Kubernetes or vendor-specific tools).

Configure at least the following values in **values-raa.yaml**:

```
raa:
  deployment:
    targetPlatform: "local"
  env:
    GCXI_GIM_DB__JSON:  |-

eyJqZGJjX3VybCI6ImpkYmM6cG9zdGdyZXNxbDovLzxob3N0Pjo1NDMyLzxnaW1fZGF0YWJhc2U+IiwgImRiX3VzZXJuYW1lIjoiPHVzZXI+Iiw
    XML_CONF: "default_conf.xml"
    STAT_SCRAPE_INTERVAL: 15
  image:
    registry: pureengage-docker-staging.jfrog.io
    pullSecrets:
      - pullsecret

  statefulset:
    containers:
      configDelivery:
        name: "{{ $.Chart.Name }}-conf-delivery"
      monitor:
        name: "{{$.Chart.Name }}-monitor"
        metrics:
          portName: "metrics"
          containerPort: "9100"
        health:
          portName: "health"
          containerPort: "9101"

  volumes:
    config:
      pv: {}
      storageClassName:
      pvc:
        name: "gcxi-raa-config-pvc"
        volumeName: ""
    health:
      pv: {}
      storageClassName:
      pvc:
        name: "gcxi-raa-health-pvc"
        volumeName: ""

  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
```

Install RAA

Using the values you configured in **values-raa.yaml**, install RAA:

1. To deploy RAA, run the following bash script :

```
helm install -n gcxi gcxi-raa ./gcxi-raa-0.1.364.tgz -f values-raa.yaml
```

2. To check the installed Helm release, run the following bash script:

   ```
   helm list -n gcxi
   ```

   Release information appears, similar to the following:

   ```
   NAME     NAMESPACE       REVISION        UPDATED
    STATUS          CHART           APP VERSION
    gcxi-raa      gcxi                                 1                    2021-06-22
    18:08:30.5041926 +0300 +0300                      deployed             gcxi-
    raa-0.1.364
   ```

3. To check the RAA project status, run the following bash script:

   ```
   helm status gcxi-raa -n gcxi
   ```

   Status information appears, similar to the following (with STATUS: deployed):

   ```
   NAME: gcxi-raa
    LAST DEPLOYED: Tue Jun 22 18:08:30 2021
    NAMESPACE: gcxi
    STATUS: deployed
    REVISION: 1
   ```

Validate the deployment

The **testRun** init container activates validation testing. Helm test command runs all the test containers: one of the test containers tests execution results, while a second performs a healthCheck.

```
   containers:

      ## optional
      ## enables config delivery init container.
      ## container delivers required xml configuration and *.ss files to RAA work dir
      ## those files delivered from gzip archive encoded as base64 and specified via {{
.Values.raa.deplyment.configTar }} helm value
      ## default conf.xml and user-data-map.ss are supplied with chart
      ## they are copied to work dir when absent and helm value above is not specified
      configDelivery: {}

      #  ## container name template
      #  ## should be uncommented when configDelivery is enabled
      #  name: "{{ $.Chart.Name }}-conf-delivery"

      ## optional
      ## enables test run init container (GCXI-3463).
      ## it performs running of all the aggregates on empty data
      ## to test the correctness of configrugation and customization
      testRun:

        ## container name template
        name: "{{ $.Chart.Name }}-test-run"

      ## main aggregation container
      aggregation:

        ## container name template
```

```
        name: "{{ $.Chart.Name }}"

      ## optional
      ## enable side car container for monitoring and processing of:
      ## - new tool command requests from a file in work dir
      ## - new prometheus scrape requests
      monitor: {}

      #  ## container name template
      #  name: "{{ $.Chart.Name }}-monitor"

      #  ## optional
      #  ## enables command processing from file placed into work dir (GCXI-4052)
      #  ## it is helpful when kubect exec is forbidden by environment restrictions.
      #  toolcmd:

      #    ## interval of checking for a new file with command
      #    intervalSec: "20"

      #  ## optional
      #  ## enables raa metrics scraping in Prometheus format via http
      #  metrics:

      #    ## k8s port name
      #    portName: "metrics"

      #    ## port number
      #    containerPort: "9100"

      #  ## optional
      #  ## enables raa health metric scraping in Prometheus format via http
      #  health:

      #    ## k8s port name
      #    portName: "health"

      #    ## port number
      #    containerPort: "9101"


  ## optional
  ## defines if any test pod will be declared for helm release testing
  ## see links:
  ## - https://helm.sh/docs/topics/chart_tests/
  ## - https://helm.sh/docs/helm/helm_test/
  testPods:

    ## optional
    ## checks the result of test run init container execution
    ## actual when {{ .Values.raa.statefulset.containers.testRun }} is specified.
    testRunCheck:

      ## pod name template
      name: "{{ tpl .Values.raa.serviceName . }}-test-run-check"

      container:
        ## container name template
        name: "{{ $.Chart.Name }}-test-run-check"

      ## optional
      ## test pod specific labels are adjusted to common labels
      labels: {}
      ## test pod specific annotations are adjusted to common annotations
      annotations:
        "helm.sh/hook-weight": "100"
        "helm.sh/hook": "test-success"
        "helm.sh/hook-delete-policy": "before-hook-creation"

    ## optional
    ## performs RAA health check
    healthCheck:
      ## pod name template
      name: "{{ tpl .Values.raa.serviceName . }}-health-check"

      container:
        ## container name template
        name: "{{ $.Chart.Name }}-health-check"

      ## optional
      ## test pod specific labels are adjusted to common labels
      labels: {}

      ## test pod specific annotations are adjusted to common annotations
      annotations:
        "helm.sh/hook-weight": "200"
        "helm.sh/hook": "test-success"
        "helm.sh/hook-delete-policy": "before-hook-creation"
```