Genesys Info Mart Private Edition Guide

Configure GCA

11/28/2025

Contents

Learn how to configure GIM Config Adapter (GCA).

Related documentation:

-
-
-
-

RSS:

- For private edition

GCA Helm chart overrides

The GCA requires some configuration for deployment that must be done by modifying the GCA's default Helm chart. You do this by creating override entries in the GCA's **values.yaml** file.

Download the **gca** and **gca-monitoring** Helm charts from your image registry, using the appropriate credentials.

For information about how to download the Helm charts, see Downloading your Genesys Multicloud CX containers. To find the correct Helm chart version for your release, see Helm charts and containers for Genesys Info Mart. For general information about Helm chart overrides, see Overriding Helm chart values in the *Genesys Multicloud CX Private Edition Guide*.

At minimum, you must create entries in the **values.yaml** file to specify key system information, as described in the following sections.

> **Important**
>
> Treat your modified **values.yaml** file as source code, which you are responsible to maintain so that your overrides are preserved and available for reuse when you upgrade.

Image registry and pull secret

Image registry

Create an entry in the GSP's **values.yaml** file to specify the location of the Genesys JFrog image registry. This is the repository from which Kubernetes will pull images.

The location of the Genesys JFrog image registry is defined when you set up the environment for the GSP. It is represented in the system as the `docker-registry`. In the GSP Helm chart, the repository is represented as `image: registry`, as shown below. You can optionally set a container version for the image.

```
image: # The repository from which Kubernetes will pull images
  registry:  # The default registry is pureengage-docker-staging.jfrog.io
  tag:  # The container image tag/version
```

Pull secret

When you set up your environment, you provision a pull secret for Genesys JFrog image registry (`docker-registry`). Each service must supply the credentials for the repository in order for Kubernetes to be able to pull from the repository. Each of the three Info Mart services (GIM, GSP, and GCA) must be configured with the pull secret. You do this **values.yaml** for the service.

```
imagePullSecrets:
  docker-registry: {} # The credentials Kubernetes will use to pull the image from the
registry
```

Note that other services use a different syntax than this to configure the repository pull secret, as follows:

```
imagePullSecrets:
  name: docker-registry
```

Genesys Info Mart, GIM Stream Processor, and GIM Configuration Adaptor helm charts all support advanced templating that allow the helm to create the pull secret automatically; hence the variation in syntax.

Tenant ID

The Tenant microservice provides direct access to the configuration server database. Configure a Helm override entry for the Tenant ID.

```
tenant_id:  # The TenantID of the tenant in use
```

Kafka

Kafka secret

The Kafka secret is necessary for GCA to access Kafka. The Kafka secret is provisioned in the system

as `kafka-secrets` when you set up the environment for GCA. Configure the Kafka secret by creating a Helm chart override in the **values.yaml** file.

```
kafka:
  password:  # Credentials for accessing Kafka. This secret is created during deployment.
```

Kafka bootstrap

To allow the Kafka service on GCA to align with the infrastructure Kafka service, make a Helm override entry with the location of the Kafka bootstrap.

```
kafka:
  bootstrap:  # the Kafka address to align with the infrastructure Kafka
```

S3-compatible storage

If you are using S3-compatible object storage on GCP to store the GCA snapshot, modify the following `storage: s3` entries in the **values.yaml** file:

```
storage:
  ...
  s3:
    bucket:  # The bucket name
    gcaSnapshots:  # The volume or folder in the bucket where the GCA snapshot will be stored
    accessKey:  # The access key created when you created the bucket
    secretKey:  # The secret created when the bucket was provisioned
    endPoint:  # The bucket host
```

GKE example

```
storage:
  ...
  s3:
    bucket: "test-example-bucket-one"
    gcaSnapshots: "/gca"
    accessKey: ""
    secretKey: ""
    useSSL: true
    endPoint: "storage.googleapis.com"
    port: 443
    Insecure: true
```

Configuration database

Specify applicable details for the configuration database. The password you configure here is provisioned as `cfgdb-secrets` in the system.

```
cfgdb: # The applicable details for the Configuration Database, created before you deployed
the Tenant service
  name:  # The name of the database
  host:  # The host on which the DBMS is running
  username:  # The user account for GCA to access the database. The user account must have at
```

```
least read permissions
  password:  # The password for the user account
```

GIM database

Specify the applicable details for the Info Mart database. The password you specify here is provisioned in the system as `gimdb-secrets`.

```
gimdb: # The applicable details for the Info Mart database
  name:  # The name of the database
  host:  # The host on which the DBMS is running
  username:  # The user account created when you created the GIM database
  password:  # The password for the user account
```

Config Maps

There are no Config Maps for GCA you can configure directly.