



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

# Genesys Customer Experience Insights Private Edition Guide

Configure GCXI

---

## Contents

- 1 Override Helm chart values
- 2 Configure Kubernetes
- 3 ConfigMaps
- 4 Secrets
- 5 Configure security
  - 5.1 Arbitrary UID

---

Learn how to configure Genesys Customer Experience Insights (GCXI).

### Related documentation:

- 
- 
- 
- 

### RSS:

- [For private edition](#)

## Override Helm chart values

Before you begin, download the latest yaml files from the repository, or examine the attached files: Sample GCXI yaml files . Helm values are described in **values.yaml**. See the comments accompanying each Helm value.

You can override values in the Helm charts to configure Private Edition. For more information, see the "suite-level" documentation about how to override Helm chart values: [Overriding Helm chart values](#).

## Configure Kubernetes

This section provides information about Kubernetes configuration.

## ConfigMaps

Configuration information is stored in ConfigMap.

See the **gcxi-worker-statefulset.yaml** file:

```
envFrom:
  - configMapRef:
      name: gcxi-config{{ template "deploymentCode" . }}
      optional: true
  - configMapRef:
      name: gcxi-config-ext{{ template "deploymentCode" . }}
      optional: true
  {{- range $cm := .Values.gcxi.configMaps }}
  - configMapRef:
      name: {{ tpl $cm.name $ }}
      optional: true
```

---

## Secrets

GCXI supports the following methods of secret injection:

- CSI driver
- Kubernetes secrets
- Environment Variables

See the **values.yaml** file:

```
secrets:  
  - name: gcxi-secret-pg
```

See the **gcxi-worker-statefulset.yaml** file:

```
- name: gcxi-var  
  projected:  
    sources:  
      - secret:  
          name: gcxi-secret{{ template "deploymentCode" . }}  
          optional: true  
      - secret:  
          name: gcxi-secret-ext{{ template "deploymentCode" . }}  
          optional: true  
    {{- range $secret := .Values.gcxi.secrets }}  
      - secret:  
          name: {{ tpl $secret.name $ }}  
          {{- with $secret.items }}  
            items:  
              {{- range $item := $secret.items }}  
                - key: {{ tpl $item.key $ }}  
                  path: {{ tpl $item.path $ }}  
              {{- end }}  
          {{- end }}  
          optional: true  
    {{- end }}
```

## Configure security

GCXI is based on a 3rd-party product (MicroStrategy), and as result has some special considerations:

- The main container is about 12 GB.
- Genesys recommends that you enable hostIPC. However, hostIPC is disabled by default and in some deployments, GCXI can operate successfully without it.

1. Enable hostIPC by setting:

```
hostIPC: true
```

2. Configure hostIPC at the node level as follows:

---

```
echo "kernel.sem = 250 1024000 250 4096" >> /etc/sysctl.conf
echo "vm.max_map_count = 5242880" >> /etc/sysctl.conf
sysctl -p
```

## Arbitrary UID

- The deployment routine assigns an arbitrary user ID (UID) and group ID to pods during deployment. Default file ownership is **genesys:root (500:0)**.
- If your OpenShift deployment uses arbitrary UIDs, you must override the securityContext settings in the **values.yaml** file (see line 456) as follows:

```
secrets:
  podSecurityContext:
    fsGroup: null
    runAsUser: null
    runAsGroup: 0
    runAsNonRoot: true

  securityContext:
    fsGroup: null
    runAsUser: null
    runAsGroup: 0
    runAsNonRoot: true
```

The default values (user ID = 500) are suitable for many other deployment scenarios:

```
secrets:
  securityContext:
    control:
      fsGroup: null
      runAsUser: 500
      runAsGroup: 500
    worker:
      fsGroup: null
      runAsUser: 500
      runAsGroup: 500
```