Genesys Customer Experience Insights Private Edition Guide

Upgrade, roll back, or uninstall GCXI

5/20/2026

Contents

Learn how to upgrade, roll back, or uninstall GCXI.

Related documentation:

-
-
-
-

RSS:

- For private edition

> **Important**
>
> The instructions on this page assume you have deployed the services in service-specific 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.

Supported upgrade strategies

Genesys Customer Experience Insights supports the following upgrade strategies:

Service	Upgrade Strategy	Notes
Genesys CX Insights	• Rolling Update	
Reporting and Analytics Aggregates	• Rolling Update	

The upgrade or rollback process to follow depends on how you deployed the service initially. Based on the deployment strategy adopted during initial deployment, refer to the corresponding upgrade or rollback section on this page for related instructions.

For a conceptual overview of the upgrade strategies, refer to Upgrade strategies in the Setting up Genesys Multicloud CX Private Edition guide.

In scenarios where you have previously deployed GCXI using Helm, use the instructions on this page to upgrade to a newer GCXI release.

Timing

A regular upgrade schedule is necessary to fit within the Genesys policy of supporting N-2 releases, but a particular release might warrant an earlier upgrade (for example, because of a critical security fix).

If the service you are upgrading requires a later version of any third-party services, upgrade the third-party service(s) before you upgrade the private edition service. For the latest supported versions of third-party services, see the Software requirements page in the suite-level guide.

Scheduling considerations

Genesys recommends that you upgrade the services methodically and sequentially: Complete the upgrade for one service and verify that it upgraded successfully before proceeding to upgrade the next service. If necessary, roll back the upgrade and verify successful rollback.

Monitoring

Monitor the upgrade process using standard Kubernetes and Helm metrics, as well as service-specific metrics that can identify failure or successful completion of the upgrade (see Observability in Genesys Customer Experience Insights).

Genesys recommends that you create custom alerts for key indicators of failure — for example, an alert that a pod is in pending state for longer than a timeout suitable for your environment. Consider including an alert for the absence of metrics, which is a situation that can occur if the Docker image is not available. Note that Genesys does not provide support for custom alerts that you create in your environment.

Preparatory steps

Ensure that your processes have been set up to enable easy rollback in case an upgrade leads to compatibility or other issues.

Each time you upgrade a service:

1. Review the release note to identify changes.
2. Ensure that the new package is available for you to deploy in your environment.
3. Ensure that your existing **-values.yaml** file is available and update it if required to implement changes.

The Helm installation package (IP) has a file name in the format gcxi-.tgz, for example **gcxi-100.0.029+0000.tgz**.

The following steps provide detailed information about preparing to upgrade GCXI:

1. On the Control Plane node, set the current directory to the helm folder, and run the following command to create a subfolder to differentiate this release from others:

```
mkdir helm_
```

where is the folder in which to extract the Helm package. For example: **018.00**.

2. To extract the Helm package, run the following command:

```
tar xvzf  -C helm_
```

where:
 is the folder you created in the previous step.
 is the name of the gcxi Helm package you are installing.

For example:

```
mkdir helm_018.00; tar xvzf gcxi-9.0.018.00.tgz -C helm_018.00
```

3. From the folder where you deployed the previous release of GCXI using Helm, copy the file **values-test.yaml** into the new folder.

Back up the meta database

The GCXI upgrade process automatically backs up your meta database before upgrading to the new release of GCXI. However, GCXI also provides additional options to back up the contents of your meta database, including:

- Scheduled backup using the container option BACKUP_HOUR.
- Immediate backup using a Kubernetes job.
- Immediate backup using PostgreSQL tools.

Backup files are stored in the gcxi-shared mount point, which is found in /genesys/gcxi/shared, unless you configure a different location. (By default, backups are stored in a subfolder called backup in /genesys/gcxi/shared).

Back up using the container variable

Genesys recommends that you configure the container option to automatically back up your data. To schedule automatic backups, set the following container variable:

```
BACKUP_HOUR=
```

Where digits in 24-hr format instruct the container to perform meta db backup in corresponding hours, for example to back up at 22:00, 6:00, and 15:00 hours:

```
BACKUP_HOUR=22,6,15
```

Back up using the Kubernetes job

You can perform a backup at any time using the included Kubernetes job, gcxi-backup, as follows:

```
kubectl delete job "gcxi-backup" || true
kubectl apply -f gcxi-backup.yaml
```

This creates two backup files (that is, service databases): mstr_meta_.pgdump and mstr_hist_.pgdump, where is a timestamp. You can optionally set another value for using the BACKUP_TAG variable in gcxi-backup job yaml.

Restoring from a backup

Ensure that backup files (pgdump) are present in the **gcxi-shared** mount location. If they are in another location, you must set the RESTORE_TAG variable in the **gcxi-restore.yaml** file.

You must stop the GCXI pods before performing a restore operation. Stop the pods using the following command:

```
kubectl scale --replicas=0 statefulset/gcxi
```

Execute the following commands to restore the meta database:

```
kubectl delete job "gcxi-restore" || true
kubectl apply -f gcxi-restore.yaml
```

Use the following command to restart the pods:

```
kubectl scale --replicas=1 statefulset/gcxi
```

Rolling Update

Rolling Update: Upgrade

Execute the following command to upgrade :

```
helm upgrade --install  -f -values.yaml  -n
```

Tip: If your review of Helm chart changes (see Preparatory Step 3) identifies that the only update you need to make to your existing **-values.yaml** file is to update the image version, you can pass the image tag as an argument by using the `--set` flag in the command:

```
helm upgrade --install  -f -values.yaml  --set .image.tag=
```

GCXI example

Set the command context by navigating to the new folder you created (see above) and run the following command:

```
helm upgrade --debug gcxi-helm gcxi/ --namespace gcxi --create-namespace -f values-test.yaml
```

RAA example

Run the following command:

```
helm upgrade gcxi-raa-lab helmrepo/gcxi-raa --values myvalues.yaml
```

Rolling Update: Verify the upgrade

Follow usual Kubernetes best practices to verify that the new service version is deployed. See the information about initial deployment for additional functional validation that the service has upgraded successfully.

Rolling Update: Rollback

Execute the following command to roll back the upgrade to the previous version:

```
helm rollback
```

or, to roll back to an even earlier version:

```
helm rollback
```

Alternatively, you can re-install the previous package:

1. Revert the image version in the `.image.tag` parameter in the **-values.yaml** file. If applicable, also revert any configuration changes you implemented for the new release.
2. Execute the following command to roll back the upgrade:

   ```
   helm upgrade --install  -f -values.yaml
   ```

   Tip: You can also directly pass the image tag as an argument by using the `--set` flag in the command:

   ```
   helm upgrade --install  -f -values.yaml  --set .image.tag=
   ```

RAA example

```
helm rollback gcxi-raa-lab --recreate-pods
```

GCXI example

> ### Important
> Do not use the `rollback` command to roll back a GCXI upgrade. You must re-install a previous version.

Rolling Update: Verify the rollback

Verify the rollback in the same way that you verified the upgrade (see Rolling Update: Verify the upgrade).

Uninstall

> **Warning**
> Uninstalling a service removes all Kubernetes resources associated with that service. Genesys recommends that you contact Genesys Customer Care before uninstalling any private edition services, particularly in a production environment, to ensure that you understand the implications and to prevent unintended consequences arising from, say, unrecognized dependencies or purged data.

Execute the following command to uninstall :

```
helm uninstall  -n
```

GCXI example

Execute the following command to remove GCXI:

```
helm uninstall gcxi-helm -n gcxi
```

RAA example

```
helm uninstall gcxi-raa-lab
```