



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.

## Tenant Service Private Edition Guide

[Upgrade, roll back, or uninstall Tenant Service](#)

---

# Contents

- **1 Supported upgrade strategies**
  - **1.1 Single region/location/cluster**
  - **1.2 Multiple regions/locations/clusters**
- **2 Backend upgrade**
- **3 Timing**
  - **3.1 Scheduling considerations**
- **4 Monitoring**
- **5 Preparatory steps**
- **6 Rolling Update**
  - **6.1 Rolling Update: Upgrade**
  - **6.2 Rolling Update: Verify the upgrade**
  - **6.3 Rolling Update: Rollback**
  - **6.4 Rolling Update: Verify the rollback**
- **7 Uninstall**

---

Learn how to upgrade, roll back, or uninstall Tenant Service.

## 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

Tenant Service supports the following upgrade strategies:

Service	Upgrade Strategy	Notes
Tenant Service	Rolling Update	

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

Tenant Service uses a rolling upgrade process. When multiple Tenant nodes are deployed, you can perform an upgrade to your existing deployment without causing a complete service outage. Tenant instances are rolled over, one by one, affecting only a portion of your agents each time.

Upgrading the master Tenant instance (node "0", which is responsible for write access to the provisioning data) causes a temporary degradation of functionality with no write access until instance "0" is restored.

Single region/location/cluster

Genesys recommends performing a backup of the backend database for Tenant Service before upgrading.

---

To perform an upgrade, follow the deployment process – use the same `helm upgrade` command and the same mandatory parameters. Remember to update the Tenant image(s) and Helm charts version tags in the **values.yaml** file. The upgrade is performed automatically, one node at a time (if **node count** is > 1). When you upgrade the primary node, the Tenant configuration updates automatically.

For information about the deployment process, see [Deployment scenarios and Deploy the service](#).

#### Multiple regions/locations/clusters

Genesys recommends performing a backup of the backend database for Tenant Service before you upgrade the master region.

To perform an upgrade, follow the deployment process – use the same `helm upgrade` command and the same mandatory parameters. Remember to update the Tenant image(s) and Helm charts version tags in the **values.yaml** file for every location. When you upgrade the master node in the master region, the Tenant configuration updates automatically.

For information about the deployment process, see [Deployment scenarios and Deploy the service](#).

## Backend upgrade

When you perform an upgrade to Tenant Service, Tenant Service provisioning is done automatically, as part of the `helm upgrade` step. There is no need to make any provisioning changes, unless noted in Release Notes for a particular version.

If you must perform an upgrade or maintenance to the database management system (DBMS) that Tenant Service uses to store its provisioning data, then the DBMS upgrade or maintenance can be performed in place, separate from Tenant Service. Perform the DBMS upgrade according to the vendor's instructions, making sure that:

- Tenant Service is disconnected from the backend during maintenance, and
- you specify the same parameters that were provisioned during the Tenant Service upgrade in order for Tenant Service to access the updated database.

Be aware that the Tenant Service is in a degraded state (no write access to provisioning) during the database upgrade.

If the database upgrade involves moving to a new DBMS instance with new parameters, then you trigger this as part of the Tenant Service upgrade. Both the new database and the new values for the database parameters (specified in the **values.yaml** file used for the `helm upgrade` step) must be ready and available before you trigger the Tenant Service upgrade.

If you're performing an upgrade or maintenance to the Redis backend that is shared with the core Voice platform, and you require a change to the Redis connection parameters, first scale Tenant Service down to a single instance (setting the `replicaCount` parameter in the **values.yaml** file), then perform the Redis backend maintenance or upgrade, and finally, trigger a restart of the Tenant Service. After that's complete, you can scale up Tenant Service again.

---

## 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 Tenant Service).

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.

---

# 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=
```

When you perform an upgrade to Tenant Service, Tenant Service provisioning is done automatically, as part of the `helm upgrade` step. There is no need to make any provisioning changes, unless noted in Release Notes for a particular version.

## 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=
```

## 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
```