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.

Setting up Genesys Multicloud CX Private Edition

Upgrade process

4/23/2024

Contents

- 1 Prerequisites
- 2 Upgrade process

Provides at a glance view of the processes involved in upgrading a Genesys Multicloud CX service.

Related documentation:

-
-

RSS:

- For private edition

Genesys delivers its container images and Helm Charts in JFrog Artifactory Edge repository, a publicly accessible repository. When there is a new version of the service available, you will receive notifications in your JFrog account.

> **Tip**
> If you have not subscribed to receive JFrog notifications, then visit Downloading your Genesys Multicloud CX containers page and set up your JFrog account to receive notifications.

You can easily pull the latest version of a Genesys Multicloud CX service from JFrog Artifactory Edge repository to your designated (quarantine) location for security scans or Continuous Delivery (CD) pipeline.

The following section details the upgrade process at a high level. It helps you to plan, prepare, and perform the upgrade of Genesys Multicloud CX services in cloud private edition infrastructure.

Prerequisites

- Access to JFrog account
- Established CD pipeline
- Established Backup process

Upgrade process

1. Select the Kubernetes upgrade strategy for the Genesys Multicloud CX service you are upgrading. Refer Upgrade strategies and select a strategy that best suits your production environment for the specific service.
2. Backup the data before starting the upgrade. If something goes wrong, you can always restore or rollback to the previous point before the upgrade.

3. Pull the latest containers and Helm charts from JFrog into your container registry.

> **Important**
>
> You can perform security scans on the pulled in containers and Helm charts from within the container registry. Security scanning depends on your organization's security policy and might not be applicable for all users.

4. Prepare your environment for the new upgrade. This step depends on the upgrade particular to that release. For example, you might have to create a new directory or pass a modified a yaml file.
5. Modify your Helm charts with appropriate overridable values.
6. Set up the CD pipeline in your environment.
7. Depending on the upgrade strategy you selected for the service, you will either upgrade a complete infrastructure, a subset of pod instances, or one pod instance at a time.
8. Run the `helm upgrade` command for your service by following the steps specific to the upgrade strategy you selected for the service. Keep in mind that the upgrade procedure varies for each upgrade strategy. Refer the service level documentation of the service you are upgrading for comprehensive explanations.
9. Test the upgrade by using the instructions given in the service level documentation of the service you have upgraded.

> **Important**
>
> If any of your test case fails or if you observe performance degradation, you can always rollback to the previous release.

10. Roll out the new version to all the users.