Genesys Customer Experience Insights Deployment Guide

Prerequisites: Before you begin installation

4/4/2026

Prerequisites: Before you begin installation

There are several supported deployment methods for Genesys CX Insights:

- **Kubernetes using descriptors** — Prerequisites for this deployment type are described in detail on this page. Complete the steps on this page as required in your environment, and then proceed to deploy GCXI as described later on this page.
- **Kubernetes using Helm** — Prerequisites for this deployment type are described in detail on this page. Complete the steps on this page as required in your environment, and then see Deploying GCXI using Helm.
- **OpenShift using Helm** — Prerequisites and information about how to install OpenShift, are provided on the Red Hat OpenShift site. For GCXI deployment instructions, see Deploying GCXI using OpenShift.
- **Docker Compose** — Installing Genesys CX Insights - Docker Compose, which is suitable for testing or development, or for very small production environments. Prerequisites for this deployment method are described on the Installing Genesys CX Insights - Docker Compose page, which also describes deployment steps.

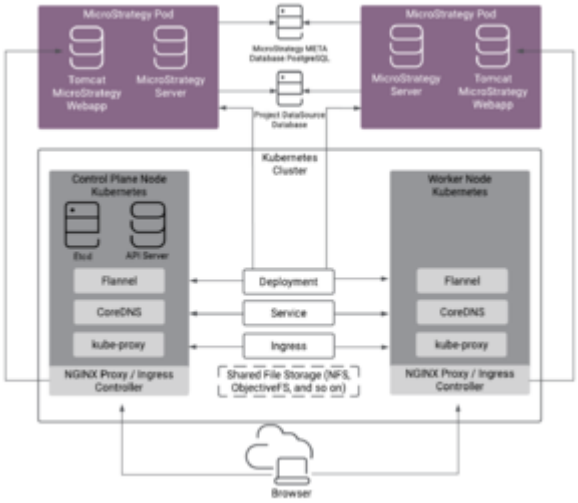
This page describes prerequisites that must be met before you can install Genesys Customer Experience Insights (Genesys CX Insights) in a Kubernetes production environment. For example, you must prepare a suitable Linux server environment, and install Kubernetes, you must plan how you will handle the meta database, and you must identify the compatible releases of the software you will need.

> **Important**
>
> The MicroStrategy server instance that runs in the container includes a temporary pre-activated license key, which is required for the operation of MicroStrategy. Request a replacement key from your customer care representative; you will install the new key during the deployment process.

1. Ensure that your system meets minimum hardware and system requirements

- **Required number of machines** — These could be real machines, virtual machines, or cloud machines such as EC2 instances in AWS, and must be configured as required to support the indicated Linux version. Ensure, for example, that IPV6 is enabled.
  - **In non-High-Availability (HA) deployments**, at least two machines (nodes), each with a supported version of Linux with the **systemd** suite installed. Genesys recommends Red Hat Enterprise Linux 7.5 (or a later 7.x release) / CentOS Linux 7.5 (or a later 7.x release). Other clones of Red Hat Enterprise Linux 7, such as Oracle Linux 7.5, should also work correctly.

- In a two-machine configuration, the Control plane node hosts mstr-01, and a worker node hosts PostgreSQL and mstr-02.
- Optionally, add a third machine if you prefer to deploy each mstr service on a separate node.



High -Availability (HA) Deployment Architecture

- **In High-Availability (HA) deployments**, the following minimums apply:
  - Small/medium customers — Two nodes for MicroStrategy, plus one node for custom cluster PostgreSQL installation.
  - Large customer — Two nodes with MicroStrategy, plus external custom cluster PostgreSQL installation.

    In HA deployments with Kubernetes, Genesys recommends:

  - A 3-node Control plane cluster, which can be either:
    - Three tiny Control plane Kubernetes nodes (meeting minimum Kubernetes requirements), plus two worker nodes (meeting Genesys CX Insights minimum requirements).

      OR

    - One tiny node for standalone Kubernetes Control plane node, plus two worker nodes that also host Kubernetes Control plane nodes.

      See About HA deployments for more information.
- In all scenarios:
  - Identify one machine that you designate the mstr-01; other machine(s) are designated as mstr-02, mstr-03, and so on.
  - Ensure that you have access to an account with root access.
- **Specifications of each machine** — In general, you can deploy MicroStrategy and Genesys CX Insights on any Linux platform with appropriate resources to deploy and run Kubernetes and Docker. However, MicroStrategy / Genesys CX Insights can require significant resources, so note the following minimums for each machine:
  - 64-bit compatible CPU architecture. (2 or more CPUs).
  - 10 GB for each GCXI container, and 2 GB for the PostgreSQL container. If your deployment scenario

involves more than one container on a machine, then the memory requirements/recommendations increase accordingly; for example, if you were to deploy the GCXI container and PostGreSQL on a single host, a minimum of 10 GB + 2 GB = 12 GB is required. Deployment of a GCXI container on a machine with less than 10 GB is not recommended, and requires changes in the `gcxi.yaml` file. Production deployments commonly reserve 16 – 64 GB RAM for each container.

- 40 GB of available disk space if loading images from a repository, 80 GB if loading from a local drive.
  Some deployments use a separate machine for PostgreSQL and gcxi_control, which requires a minimum configuration of 2GB RAM + 1 CPU. The recommended configuration varies depending on the load in your environment, but generally is 4 GB + 2 CPUs. This reflects the requirements for PostGreSQL, as gcxi_control doesn't demand significant resources.

For information about requirements to install Kubernetes, see Installing kubeadm.

About HA deployments

Because a Genesys CX Insights environment consists of several interoperating components, there are several levels of HA configuration that you can optionally deploy. Each type of HA configuration is independant from the others -- you can configure any, all, or none of the following components as HA:

- **GCXI HA** — In a Gensys CX Insights HA deployment, MicroStrategy clusters are used to distribute workers (at least 2, but not more than 8) across multiple machines. The MicroStrategy containers in an HA cluster use an *active-active* model, so if any container fails, Genesys CX Insights continues to operate normally.
- **Kubernetes platform HA** — In a Kubernetes HA deployment, redundant infrastructure is used to run the Kubernetes management infrastructure on several nodes, configured in such a way that the failure of any single node does not interfere with normal Kubernetes operation. Each redundant node stores all Kubernetes state information (etcd). Genesys CX Insights requires no configuration changes to work with HA Kubernetes.
  Kubernetes supports a wide variety of HA options, the simplest consisting of three Kubernetes Control plane nodes. For more information about specific HA deployment options, see Kubernetes HA topology. Note that each MicroStrategy instance that is part of an HA cluster must use a unique path to store log files. The creation of MicroStrategy clusters can fail in scenarios where more than one MicroStrategy instance tries to write logs into a common folder, such as **/mnt/log**.
- **PostgreSQL HA** — PostgreSQL is used for GCXI server metadata; in an HA PostgreSQL deployment, Genesys CX Insights can continue to operate when PostgreSQL container fails. Standard GCXI deployments use a single-node PostgreSQL deployment, and this documentation describes only single node (non-HA) PostgreSQL. To configure HA for the server metadata, follow PostgreSQL documentation, and use standard PostgreSQL database capabilities and administration procedures to configure an external database, and point Genesys CX Insights Kubernetes to it. Genesys CX Insights requires no configuration changes to work with HA PostgreSQL.

2. Plan to accommodate the meta database

Depending on whether you choose to deploy an external PostgreSQL server for the *meta* database, one of the following statements applies:

- If you use an external PostgreSQL server to store the MicroStrategy meta database, ensure that your

PostgreSQL server is configured with a compatible release:

- For new installations of release 100.0.020 and later, PostgreSQL 12 is required for the meta database. If you upgrade an existing deployment to release 100.0.020 or later, it will continue to work with existing PostreSQL releases.

- If you wish to avoid deploying and managing a PostgreSQL server, use the prepackaged PostgreSQL server provided in the Installation Package. For more information about this option, see Installing Genesys CX Insights in production environments.

3. Identify compatible software releases

Genesys CX Insights requires that your environment contain supported releases of the following components.

- Genesys recommends that you use the latest supported version wherever possible.
- See Genesys CX Insights Product Alert for detailed information about supported releases.

4. Pre-installation configuration

Complete the following configuration changes, referring to the operating system documentation for more information if needed:

1. Configure shared memory settings — MicroStrategy requires that you preconfigure shared-memory settings on the host operating system. See the MicroStrategy website for steps appropriate to your system.

> **Important**
>
> Changes to shared memory configuration can impact all applications and the operating system itself. These steps provide an example; follow them only if you are certain they apply to your environment.

Complete the following steps on each machine:

1. Log in as root, or execute the following command to switch to root:

```
sudo -i bash
```

2. Execute one of the following commands:
   Release 9.0.014 and later:

```
echo "kernel.sem = 250 1024000 250 4096" >> /etc/sysctl.conf
```

   Earlier releases:

```
echo "kernel.sem = 250 32000 32 4096" >> /etc/sysctl.conf
```

3. Execute the following command:

```
echo "vm.max_map_count = 5242880" >> /etc/sysctl.conf
```

4. Reboot the machine.

2. **Set up DNS** — Set up DNS for each machine in your cluster and (using, for example, 'ping', 'host', and 'hostname --fqdn' commands) verify that each machine can resolve itself and each other by DNS name.

3. Ensure that all machines in the Kubernetes cluster have access to the *shared* folder, which is used for internal purposes by MicroStrategy.
   - The shared folder must be accessible to each machine in the Kubernetes cluster. It can use SMB, NFS, a Kubernetes shared volume, or whatever other method of network share your environment permits. For HA deployments, you must create the shared folder on both node and replica hosts, share the folder on the node host, and mount to the previously created folder on the replica host, as read-only.
   - By default, the containers expect the shared folder to be located at **/genesys/gcxi/shared** on the host. Optionally, you can use another location for the shared folder. To do so, complete the following steps prior to container startup: Open the **gcxi.yaml** file for editing, find all instances of the string **genesys/gcxi/shared**, and replace them with the new folder path.

4. Configure RAA — Some Genesys CX Insights reporting features and the associated objects (including certain folders and reports) are not needed in all deployments, or may require additional configuration steps. Beginning with Genesys CX Insights release 9.0.010, the Genesys CX Insights deployment routine automatically enables these reporting features based on the features you enable in RAA. If an RAA feature in the following table is enabled when you deploy Genesys CX Insights (or restart the container), the corresponding feature is enabled in Genesys CX Insights, and relevant folders and reports are visible. To enable or disable one of these features in Genesys CX Insights, you can enable or disable the the corresponding option in RAA, and:
   - In release 9.0.010, restart the Genesys CX Insights container to enable or disable the indicated feature.
   - In release 9.0.011 or later, wait one hour, and Genesys CX Insights automatically commits your changes, enabling or disabling the indicated feature.

   If you disable a feature that has been enabled for a period of time, and there is data in the tables that you wish to retain (for example, to use in reports), copy the tables into the Custom folder before restarting the container (9.0.010) or before an hour has passed (9.0.011).

Configure these RAA options to automatically enable GCXI features

In RAA release 8.5.011.02 and later, a new configuration option, enable-available-features, in the **[agg-feature]** section, enables all features that are supported in the current RAA release, except, in some releases, **enable-gpr-fcr**, as noted below. If you set the **enable-available-features** option, you do not also need to set the individual options here.

If this RAA feature is enabled:	These objects appear in GCXI:
enable-bgs	Chat Bot folder and reports
enable-callback	Callback folder and reports
enable-chat	Chat folder and reports
enable-chat-thread	Chat folder > Chat Thread Report
enable-cobrowse	Co-browse folder and reports
enable-gpr	Predictive Routing folder and reports

enable-gpr-fcr	The Predictive Routing folder, and the following reports: Predictive Routing AB Testing Report, Predictive Routing - AHT & QUEUE. This option is enabled by enable-available-features only in release 8.5.011.02. In later releases, it is not enabled by enable-available-features, and must be enabled manually.
enable-media-neutral	In the Agents folder: Agent Omnichannel Activity Report
enable-sdr	Designer folder and reports
enable-sdr-bot	In the Designer folder: Bot Analytical Dashboard and Final Disposition Dashboard
enable-sdr-survey	In the Designer folder: Survey Answer Report and Survey Statistics Report
eServicesSM	In the Agents folder: Agent Social Engagement Report
post-call-survey	All reports in the Agent folder except details and interval-based reports, and all reports in the Business Results folder.
user-data-gen-dim	All reports in the Agent folder except details and interval-based reports, and all reports in the Business Results, Outbound Contact, and Queue folders.

For information about enabling features in RAA, see the *Reporting and Analytics Aggregates Options Reference* guide.

Contact Center Sizing Categories

Where this document refers to contact centers as *small*, *medium*, or *large*, these terms refer to environments of approximately the sizes listed in the table **Sizing Categories**:

Sizing Categories

Sizing Category	Number of Agents	Number of Agent Groups	Number of Queues	Daily Call Volume
Small	Fewer than 500	Fewer than 50	Fewer than 50	On the order of tens of thousands
Medium	Fewer than 5000	Fewer than 400	Fewer than 1000	Up to 500000
Large	Fewer than 30000	Fewer than 1000	Fewer than 8000	Up to 4000000

Installing Docker and Kubernetes

In most scenarios, you can deploy Docker and Kubernetes by accessing the installation packages and other files directly from the internet (*online scenarios*). However, it is also possible to deploy the software in environments where it is not possible to access the internet or other external networks

from the machines / network where you plan to install Genesys CX Insights (*offline scenarios*). Choose one of the following options:

- Installing Kubernetes and Docker in online scenarios — for most deployments, **Genesys recommends this option**.
- Installing Kubernetes and Docker in offline scenarios — for scenarios where your deployment environment cannot access the internet.

Tomcat Version

GCXI is shipped with the latest Tomcat version available at the moment of the release build.

To find out the Tomcat version included in your version of GCXI, do either of the following:

- Run the following command:
  ```
  docker run --rm <gcxi_image> /opt/tomcat/bin/version.sh
  ```
  For example, `docker run --rm gcxi:100.0.034.0000 /opt/tomcat/bin/version.sh`.

(Or)

- Run the above command through `docker exec` or `kubectl exec` in the running container.
  For example, `kubectl exec <gcxi_running_pod> /opt/tomcat/bin/version.sh`.

> **Tip**
> If you are using a tool other than `docker` or `kubectl`, refer to the vendor's documentation for instructions on how to run the command inside the container.