GENESYS™

Genesys Customer Experience Insights Deployment Guide

Installing Genesys CX Insights - Podman Compose

12/15/2025

Contents

Installing Genesys CX Insights - Podman Compose

GCXI is agnostic to container environments. It can work on top of Docker, Podman, or Kubernetes with any valid container runtime (like containerd or cri-o). The Podman setup is similar to a GCXI installation with Docker with minor syntactic differences between both types.

Prerequisites

GCXI deployment on Podman shares the same requirements to the host system, as GCXI on Docker.

- requirements to IPC / shared memory `sysctl` settings, namely `kernel.sem` and `vm.max_map_count`
- `selinux` must be set to permissive mode or disabled

Install Podman

1. Install Podman version 4.5 or greater. See Installation for more information.
2. Verify if the installation was successful by running `podman version`.
3. Enable Podman socket and verify.

```
systemctl enable --now podman.socket
systemctl status podman.socket
```

Install Podman Plugins

1. Install Podman plugins package. For RedHat 8/9, or CentOS Stream 8/9, the command will be: `yum install -y podman-plugins.x86_64`. See the Podman documentation for more information on installing plugins for your operating system.

Install Docker Compose

1. Install the docker-compose standalone binary following the instructions in Install Compose standalone.

2. Verify if the installation was successful by running `podman compose version`.

Prepare File Volumes

Similar to GCXI docker-compose deployment, GCXI on Podman uses certain folders on the host machine to store persistent data.

Before running GCXI on Podman, review all *host-path file volumes*, mentioned in the compose file, and ensure that the corresponding folders on the host

1. exist and

2. are owned by the user under which GCXI container runs (by default, user 500).

```
# compose file syntax reference
# https://docs.docker.com/compose/compose-file/

# excerpt from GCXI compose yaml file, describing host-path volumes
# note that particular host paths (left part before colon) are merely examples
# if needed, you may change them to whatever paths are convenient to you
volumes:
- "/genesys/gcxi_config:/genesys/gcxi_config"
- "/genesys/gcxi_shared:/genesys/gcxi_shared"
- "mstr_log_01:/mnt/log"

# NB: in this example, the log volume is defined as a temporary volume
# it will disappear after the GCXI deployment is brought down
# if you want to preserve GCXI logs permanently in the file system,
# you can redefine the log volume as a host-path volume
# '/mnt/log/gcxi' is an example path, you can choose any
volumes:
- "/mnt/log/gcxi:/mnt/log"

# make sure, that all host-path volumes folders, mentioned in the compose file:
# - exist
# - are owned by the container user (user 500 by default)

# for the example above, the needed commands will look like this:
mkdir -p "/genesys/gcxi_config" "/genesys/gcxi_shared"
chown -R 500:0 "/genesys/gcxi_config" "/genesys/gcxi_shared"
```

Prepare GCXI Images

Newer versions of Podman client work similar to Docker. Podman supports all necessary commands for working with images, like `podman pull`, `podman load`, or `podman tag`.

In most cases, GCXI images are shipped as tar.gz archives. Use `podman load` to create a usable OCI image, which may be consumed by Podman.

```
podman load -i gcxi.tar.gz
podman load -i gcxi_control.tar.gz

# (optional)
# after unpack, the images appear under full names, containing repo and version
# like 'pureengage-docker-production.jfrog.io/gcxi/gcxi:100.0.035.0000'
# if needed, retag them to more convenient names
# it allows to refer to these shorter names in your compose files or other tools
podman tag pureengage-docker-production.jfrog.io/gcxi/gcxi:100.0.035.0000 gcxi

# list images to ensure successful image load
podman images
```

Review GCXI Compose File

GCXI IP ships a sample **docker-compose.yaml** file.

This file may be **fully reused** by Podman versions > 4.5, no changes are needed. Configure GCXI variables in the compose file referring to the standard GCXI documentation. Ensure that the `image` references in your compose file are configured similar to how GCXI images are tagged on the host. General GCXI requirements about host IPC settings apply as usual.

Run GCXI on Podman Compose

1. Run GCXI on Podman Compose. `podman compose --file docker-compose.yml up`

Podman Compose almost fully mirrors behavior of Docker Compose, hence operating `podman compose` client shouldn't comprise any difficulty.