Genesys Customer Experience Insights Deployment Guide

Installing Genesys CX Insights

12/18/2025

Contents

Installing Genesys CX Insights

This page describes the steps required to deploy Genesys Customer Experience Insights (Genesys CX Insights) on Docker with Kubernetes, which is the supported and recommended scenario for deploying Genesys CX Insights in production environments.

Before You Begin

Before you complete the steps on this page, prepare the environment as described in Preparing to install Genesys CX Insights. In addition:

1. Acquire the Genesys CX Insights Installation package

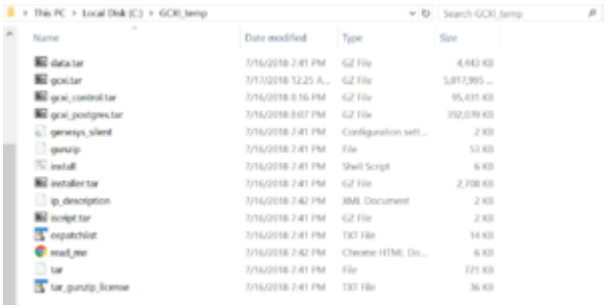
> **Warning**
> The Genesys CX Insights Installation Package comprises several files, some of which are too large to be downloaded using Genesys Download Center. Contact Customer Care for assistance.

Ensure that you have the latest Genesys CX Insights 9.0 Installation Packages (IP); talk to your Genesys representative for information about where to download the IP.

Installation Packages for GCXI

Component	IP / File	tar files
CustExInsights — Genesys Customer Experience Insights	Docker container (Docker Linux platform) **IP_CustExpInsights_9000XXX_ENU_dockerlinux.zip** (where XXX is the latest release number.)	**gcxi.tar.gz** — contains the **gcxi** Docker image, which contains a fully installed Microstrategy Server 10.x (the latest supported release of MicroStrategy: 10.11, for example.). This container provides a *stateless* deployment, where project data (reports, users, and other objects) is stored separately in a MicroStrategy *meta*

		database. This image is used for production deployments.
	Docker container (regular Linux IP [Linux platform]) **IP_CustExpInsights_9000XXX_ENU_linux.zip** (where XXX is the latest release number.)	**data.tar.gz** — contains the various YAML files (Kubernetes script files), such as gcxi.yaml, gcxi-posrtgres-yaml, and gcxi-init.yaml, and the gcxi.properties file (files which you must edit as part of the deployment procedure), PostgreSQL database dump with MicroStrategy meta-data database for GCXI project, and other files needed for GCXI
CustExInsightOps — Genesys Customer Experience Insights Ops	Docker container (Docker Linux platform) **IP_CustExpInsightsOPS_9000XXX_ENU_dockerlinux.zip** (where XXX is the latest release number.)	**gcxi_control.tar.gz** — contains the **gcxi_control** Docker image, which is used for deployment and configuration of the GCXI solution.
CustExInsightDB — Genesys Customer Experience Insights DB	Docker container (Docker Linux platform) **IP_CustExpInsightsDB_9000XXX_ENU_dockerlinux.zip** (where XXX is the latest release number.)	**gcxi_postgres.tar.gz** — contains the **gcxi_postgres** image, which contains a PostgreSQL database server with GCXI MicroStrategy Project, Meta data, and History databases pre-deployed.
MSSecEntPltf — MicroStrategy Secure Enterprise Platform for Linux	MicroStrategy software for Linux (server) **MicroStratedgy_Secure_Enterprise_Platform_10_X_Lin.zip** (where X is the current release. For example, 10.1.1000.00 — corresponds to 10.11 Microstrategy for Linux installation)	MicroStratedgy_Secure_Enterprise_Platform_10_X_Lin
MSSecEntPltf64 — MicroStrategy Secure Enterprise Platform for Windows	MicroStrategy software for Windows (client / editing tools) **MicroStratedgy_Secure_Enterprise_Platform_10_X_Win/cpe1705/PI.zip** (where X is the current release. For example, 10.1.1000.00 — corresponds to 10.11 Microstrategy for Windows installation)	MicroStratedgy_Secure_Enterprise_Platform_10_1_Win/cpe1705/PI
Note: Reporting and Analytics Aggregates (RAA) files are also available in the same location (**Reporting_and_Analytics_Aggregates_G231_850XXXX_ENU_ISO**). See the Reporting and Analytics Aggregates documentation for more information about deploying RAA.		



The installation package contents

2. Gather information about data sources

For Genesys CX Insights to produce meaningful reports, you must have installed and properly configured both Genesys Info Mart and Reporting and Analytics Aggregates (RAA):

- Genesys Info Mart release 8.5 database — The Genesys Info Mart documentation describes how to deploy and configure Genesys Info Mart, including information about hardware sizing requirements to support Gensys Info Mart. Genesys CX Insights can provide meaningful reports only if the Info Mart database is regularly populated by a Genesys Info Mart 8.5 application. Genesys Info Mart must be properly configured and installed before Genesys CX Insights runs the aggregation process (RAA). Refer to the *Genesys Info Mart Deployment Guide* or the *Genesys Migration Guide* for information that pertains to configuring, installing, or upgrading Genesys Info Mart.
- Reporting and Analytics Aggregates (RAA) — The RAA documentation describes how to deploy RAA, and how to configure the aggregation process. You must have available all relevant Genesys Info Mart information, including the RDBMS type (Microsoft SQL Server, PostgreSQL, Oracle), hostname, and user credentials.

3. Decide how to handle the meta database

Choose whether to deploy an external PostgreSQL server for the *meta* database:

- Deploying with an external meta database — Requires that you create an external PostgreSQL server on which the MicroStrategy meta database resides.
- Deploying the pre-packaged meta database — Uses a pre-packaged meta database, so that you do not need to deploy or manage a separate PostgreSQL server. This option uses the standard PostgreSQL container. If you choose this option, note the following:

- Only one instance of the PostgreSQL container should be run any time.
- By default, this database container is configured to always run on the master node.
- Postgres containers store database data (including *all of your reporting data*) in a Docker volume, which is a physical directory that is mapped to the container. By default, this directory is on the primary (master) node of the cluster, in the directory `/genesys/gcxi/data`), and the container is configured to run only on the master node. Prior to starting the container, you can change the directory that is used to store the data by editing the **gcxi-postgres.yaml.** file, and changing the `/genesys/gcxi/data` to another valid path.

> **Important**
>
> In scenarios where you use the pre-packaged meta database, Genesys strongly recommends that you regularly back up the contents of the **Docker volume** directory, as it contains your Genesys CX Insights database data.

Deploying the containers

Use the steps in this section to deploy the Docker containers. Note that Genesys does not ship Docker or Kubernetes as a part of Genesys CX Insights. You must install Docker in your environment before you can load the Genesys CX Insights containers; see Preparing to install Gensys CX Insights. Install Docker according to the instructions on the Docker site. A Docker deployment provides a complete self-contained environment, so that you do not need to manually configure ports or address compatibility issues. Genesys CX Insights docker images may be available from two sources:

- from the tar.gz archive described above
- from the Genesys docker repository (contact your Genesys representative for the repository address and credentials)

Depending on the source from which you acquired the images, follow the appropriate procedure:

- Load Docker images from tar.gz archives
- Load Docker images from from the Genesys Docker repository

Procedure: Load Docker images from tar.gz archives

Purpose: Use the steps in this procedure to prepare the Docker containers.

Prerequisites

Make sure this is the method you want to use. Consider Load Docker images from from a Docker repository as an alternative.

Steps

Complete the following steps on EACH machine, except where noted otherwise:

1. Copy the docker containers from the IPs (gcxi_control.tar.gz, and either gcxi.tar.gz or gcxi_postgres.tar.gz) onto each machine in the cluster.
2. Log in as root, or execute the following command to switch to root:

   ```
   sudo -i bash
   ```
3. On each machine in the cluster, execute the following command:

   ```
   docker load < gcxi_control.tar.gz
   ```
4. On each machine in the cluster, execute the following command:

   ```
   docker load < gcxi.tar.gz
   ```
5. If you plan to use the pre-packaged meta database, execute the following command on the MASTER machine:

   ```
   docker load < gcxi_postgres.tar.gz
   ```

   Note that this is a very large image, so expect it to take some time to complete.

Procedure: Load Docker images from a Docker repository

Purpose: Use the steps in this procedure to prepare the Docker containers by loading them from a Docker repository. This example uses the Genesys Docker repository, but for your deployment, you are more likely to use your own, which may require modifications to the .yaml files.

Prerequisites

Make sure this is the method you want to use. Consider Load Docker images from tar.gz archives as an alternative.

Steps

Complete the following steps on EACH machine, except where noted otherwise:

1. On each machine in the cluster, execute the following command:

   ```
   docker pull <repo_name>/gcxi_control:<gcxi_version>
   ```

   where <repo name> is the repository name, and <gcxi_version> is the release number of the software.

   For example,

   ```
   docker pull sdocker-registry.genhtcc.com/gcxi_control:9.0.007.03
   ```

2. On each machine in the cluster, execute the following command:

   ```
   docker pull <repo_name>/gcxi:<gcxi_version>
   ```

   For example,

   ```
   docker pull sdocker-registry.genhtcc.com/gcxi:9.0.007.03
   ```

Note that this is a very large image, so expect it to take some time to complete.

3. On the MASTER machine, execute the following command only if you plan to use the pre-packaged meta database:

   ```
   docker pull <repo_name>/gcxi_postgres:<gcxi_version>
   ```

   For example:

   ```
   docker pull sdocker-registry.genhtcc.com/gcxi_postgres:9.0.007.03
   ```

4. Images downloaded from repositories contain tagging strings that can cause installation errors. Execute the following commands to retag each of the images:

   ```
   docker tag <repository name>/<image>:<release> <image>:<release>
   ```

   where <repository name> is the identifier for the repository from which you downloaded the files, <image> is the name of the image file, and <release> is the release number.

   For example:

   ```
   docker tag sdocker-registry.genhtcc.com/gcxi_control:9.0.007.03 gcxi_control:9.0.007.03

   docker tag sdocker-registry.genhtcc.com/gcxi:9.0.007.03 gcxi:9.0.007.03

   docker tag sdocker-registry.genhtcc.com/gcxi_postgres:9.0.007.03 gcxi_postgres:9.0.007.03
   ```

Procedure: Install a new License key

Purpose: Use the steps in this procedure to install a new license key. The MicroStrategy server instance that runs in the container includes a pre-activated key, which is required for the operation of MicroStrategy. Beginning with

Genesys CX Insights release 9.0.009, this key is temporary. This procedure describes the steps required to update the key. Note that the key you install in this procedure will continue to operate for one year; if you have not upgraded GCXI within that time, plan to install a new license key.

Prerequisites

Contact your Genesys representative to obtain a new license key.

Steps

1. Execute the following command to back up the GCXI meta db:

   ```
   kubectl -f apply k8s/gcxi-backup.yaml
   ```

2. Execute the following commands to stop currently running containers

   ```
   kubectl scale deploy/gcxi-slave --replicas=0
   ```

   ```
   kubectl scale deploy/gcxi-master --replicas=0
   ```

3. Edit the **gcxi.properties** file, and add the line

   ```
   MSTR_LICENSE=<your new license>
   ```

   where `<your new license>` is the new license key value

   This adds the MSTR_LICENSE environment variable to your Genesys CX Insights environment.

4. Execute the following commands to load gcxi.properties into Kubernetes:

   ```
   kubectl delete configmap gcxi-config
   ```

   ```
   kubectl create configmap gcxi-config --from-env-file=k8s/gcxi.properties --namespace genesys
   ```

5. Execute the following command to start the master container:

```
kubectl scale deploy/gcxi-master --replicas=1
```

Wait until master node is done (wait until Tomcat is up, and MicroStrategyWeb page is available).

6. Execute the following command to start the slave container:

```
kubectl scale deploy/gcxi-slave --replicas=1
```

Procedure: Install the CustExInsights component

Purpose: Use the steps in this procedure to install the Genesys Customer Experience Insights Docker Linux (CustExInsights) component.

Steps

Complete the following steps on the MASTER machine:

1. Download the CustExpInsights Installation Package (IP), extract its contents, and copy the `\9.0.00x.0x\linux\ip` sub folder to a directory on the MASTER node.
2. Execute one of the following commands to install the `data.tar.gz` IP:
   - On Red Hat Enterprise Linux 7.5 deployments, use the following commands to install the software automatically:
     1. Execute the following command to set permissions on the installation script:

```
chmod a+x install.sh
```

     2. Execute the installation script:

```
./install.sh
```

OR

- For Genesys CS Insights 9.0.008 and earlier releases with CentOS Linux 7.5, manually extract and install the software as follows:
  1. Execute the following command to create a folder in which to install the Genesys Customer Experience Insights Docker Linux (CustExInsights) component:
  2. `mkdir -p /<destination path>/gcxi`

     where:

     `<destination path>` is the folder path and name.

     For example:

     ```
     mkdir -p /genesys/gcxi
     ```
  3. Execute the following command to extract the IP:

     ```
     tar -xzvf <ip path>/data.tar.gz -C /<destination path>/gcxi
     ```

     where:

     `<ip path>` is the path to the folder where you copied `data.tar.gz` .

     For example:

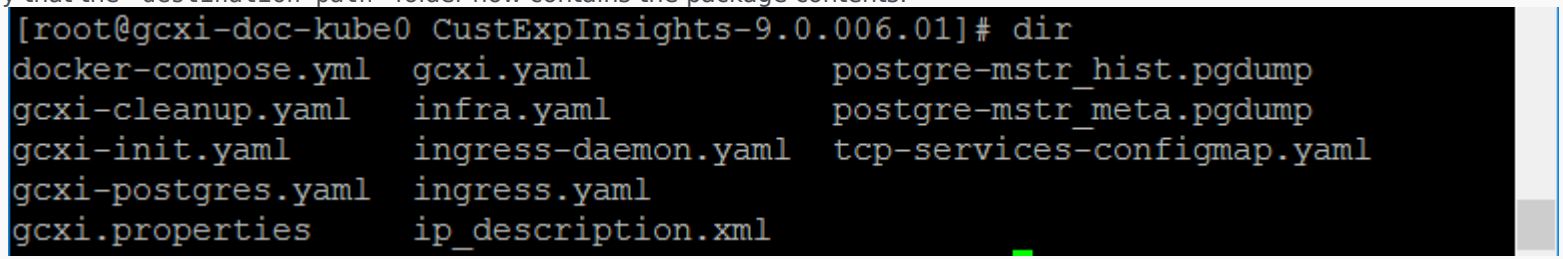
     ```
     sudo tar -xzvf /var/tmp/ip/data.tar.gz -C /genesys/gcxi
     ```
  4. Execute the following command to find all the YAML files that contain the string `[ToBeChanged: GCXI_VERSION]`:

     ```
     find <destination path>  -type f -exec grep -H 'ToBeChanged: GCXI_VERSION' {} \;
     ```

     where `<destination path>` is the folder in which you extracted the IP, for example `/genesys/gcxi/`

5. In each place where `[ToBeChanged: GCXI_VERSION]` is found, replace it with the release number (for example 9.0.007.03).

3. Verify that the <destination path> folder now contains the package contents:



Procedure: Enter database information in the properties file

Purpose: Use the steps in this procedure to populate the **gcxi.properties** file with information about your Info Mart.

> ### Tip
> To identify the folder where the Genesys CX Insights files are installed, execute the following command:
>
> ```
> sudo find -name gcxi.properties
> ```
>
> which returns the path to the `gcxi.properties` file, such as:

```
./genesys/gcxi/gcxi.properties
```

Steps

Complete the following steps on the MASTER machine:

1. Open the **gcxi.properties** file for editing.
   Genesys CX Insights database properties

   This section of the file provides information about the GCXI database properties:

   1. For Oracle RDBMS, skip this step. For other RDBMS, populate the `GIM_DB` property to reflect the actual Genesys Info Mart DB name.
      For example: `GIM_DB=gim_host1`
   2. Populate the `GIM_DB_TYPE` property with your RDBMS type. The following values are supported: POSTGRESQL, SQLSERVER, ORCLW
      For example: `GIM_DB_TYPE=POSTGRESQL`
   3. Populate the `GIM_DB_TYPE_EX` property with your extended RDBMS type. The following values are supported at the time of writing: `Microsoft SQL Server 2012`, `Microsoft SQL Server 2014`, `Microsoft SQL Server 2016`, `PostgreSQL 8.1`, `PostgreSQL 8.2/8.3`, `PostgreSQL 8.4`, `PostgreSQL 9.x`, `Oracle 11g`, `Oracle 11gR2`, `Oracle 12c`, `Oracle 12c R2`. For an up-to-date list of supported values, see the MicroStrategy documentation.
      For example: `GIM_DB_TYPE_EX=PostgreSQL 9.x`
   4. Populate the `GIM_HOST` property with your Genesys Info Mart RDBMS host name.
   5. Populate the `GIM_LOGIN` and `GIM_PASSWORD` properties.
      For example: `GIM_LOGIN=gim_username` and `GIM_PASSWORD=gim_PassWord123`.
   6. For Oracle RDBMS, complete this step (for other RDBMS, skip this step): Populate either the `GIM_ORCL_SID` or the `GIM_ORCL_SNAME` (not both).

7. Populate the GIM_PORT property with your Genesys Info Mart RDBMS port number.
   For example: GIM_PORT=1433

MicroStrategy database properties

This section of the file provides information about MicroStrategy *meta* and *history* database properties. These are databases where Microstrategy stores internal information, and are created automatically during Genesys CX Insights deployment:

Choose one of the following:

- If you plan to use the pre-packaged meta database, leave all the META* properties empty.
- If you are using an external PostgreSQL server to host the meta database complete the following steps:

1. Populate the META_DB_ADMIN, META_DB_ADMINDB, and META_DB_ADMINPWD properties with an existing user name, database name, and password. The user name specified must correspond to an account that has the necessary permissions to create databases and database users, and to assign ownership.
   For example: META_DB_ADMIN=postgres, META_DB_ADMINDB=postgres_db, and META_DB_ADMINPWD=postgres_pwd.
2. Populate the META_DB_HOST and META_DB_PORT properties host and port where the meta and history databases will be created:

   ```
   META_DB_HOST=<Host name></tt> and <tt>META_DB_PORT=<port>
   ```

   where <Host name> is the host name where external PostgreSQL server located, and <port number> is the port of the external PostgreSQL server (usually 5432).
3. Populate the META_DB_LOGIN and META_DB_PASSWORD properties with credentials for a new user account to be created for the MicroStrategy meta database. The deployment routine uses the information you provide to create a new user account for the meta database.
   For example: META_DB_LOGIN=mstr_meta_kube and META_DB_PASSWORD=g1n2s3s4
4. Populate the META_HIST_LOGIN and META_HIST_PASSWORD properties with appropriate credentials for the MicroStrategy history database. The deployment routine uses the information you provide to create a user account for the history database.
   For example: META_HIST_LOGIN=mstr_hist_kube and META_HIST_PASSWORD=g1n2s3s4.

Other properties

This section of the file provides information about other relevant properties:

1. Populate the `MSTR_PASSWORD` property with a suitable administrative password (minimum 8 characters, with 1 each of upper case, lower case, and numeric). The deployment routine uses the information you provide to set the administrator password for Microstrategy.
   For example: `MSTR_PASSWORD=Pa55word`
2. Ensure that the `GCXI_VERSION` property was set correctly by the installer. It must correspond to the release you are installing, for example `GCXI_VERSION=9.0.007.03`.
3. Ensure that the `IMAGE` property was set correctly by the installer. It must correspond to the release you are installing, for example `IMAGE=gcxi:9.0.007.03`.

Procedure: Deploy Genesys CX Insights

Purpose: Use the steps in this procedure to deploy Genesys CX Insights into Kubernetes.

Steps

Complete the following steps on the MASTER machine:

1. To create Kubernetes namespace 'genesys', execute the following command:

   ```
   kubectl create -f <destination path>/infra.yaml
   ```

   where <destination path> is the folder in which the Geneys Installation Package (IP) is stored, for example:

   ```
   kubectl create -f /genesys/gcxi/infra.yaml
   ```

2. To set 'genesys' namespace as the default namespace, execute the following command:

   ```
   kubectl config set-context $(kubectl config current-context) --namespace=genesys
   ```

3. To load the variables into Kubernetes, executethe following command:

   ```
   kubectl create configmap gcxi-config --from-env-file=<destination path>/gcxi.properties --namespace genesys
   ```

   where <destination path> is the folder in which the Geneys IP is stored, for example:

   ```
   kubectl create configmap gcxi-config --from-env-file=/genesys/gcxi/gcxi.properties --namespace genesys
   ```

4. If you are hosting the meta database on an external server, skip this step. If you are using the pre-packaged meta database, complete the following steps:

   1. Execute the following command to start the PostgreSQL database container, which is required so that your GCXI meta database to run in the PostgreSQL container as a part of the Kubernetes cluster:

      ```
      kubectl create -f <destination path>/gcxi-postgres.yaml
      ```

      where <destination path> is the folder in which the Geneys IP is stored, for example:

      ```
      kubectl create -f /genesys/gcxi/gcxi-postgres.yaml
      ```

   2. Execute the following command to verify the state of the gcxi-postgres pod:

      ```
      kubectl get pods | grep 'gcxi-postgres*'
      ```

      The pod status should be Running, for example:

      ```
      gcxi-postgres-5cd4d45754-mss6p   1/1       Running     0          6d
      ```

      If it has any other state, wait a few minutes and check again (it may take some time).

5. Edit the **gcxi.yaml** file as follows:

   - Change the `image: gcxi:[ToBeChanged: GCXI_VERSION]` line so that it specifies the correct release number, for example `9.0.007.03`.

- Ensure that the volume mounted to /genesys/gcxi/shared is a shared folder visible to both MASTER and SLAVE machines.

6. To create the MicroStrategy meta database, complete the following steps:
   1. Execute the following command to create the database:

      ```
      kubectl create -f <destination path>/gcxi-init.yaml
      ```

      , for example:

      ```
      kubectl create -f /genesys/gcxi/gcxi-init.yaml
      ```

      where <destination path> is the folder in which the Geneys IP is stored. Note that this step is required even if you use the pre-packaged PostgreSQL container.
   2. Execute the following command to verify the state of the gcxi-init pod:

      ```
      kubectl get pods | grep 'gcxi-init*'
      ```

      The pod status should be Completed, for example:

      ```
      gcxi-init-l4b4x                  0/1       Completed   0          6d
      ```

      If it has any other state, wait a few minutes and check again (it may take some time).
7. To deploy the MicroStrategy containers, execute the following command:

   ```
   kubectl create -f <destination path>/gcxi.yaml
   ```

   where <destination path> is the folder in which the Geneys IP is stored, for example:

   ```
   kubectl create -f /genesys/gcxi/gcxi.yaml
   ```
8. Complete the following steps to verify that both GCXI pods are running:
   1. Execute the following command to verify the state of the master gcxi pod:

```
kubectl get pods | grep 'gcxi-master*'
```

The master pod status should be Running, for example:

```
gcxi-master-549f6897f-zghqf      1/1       Running     0          6d
```

If it has any other state, wait a few minutes and check again (it may take some time).

2. Execute the following command to verify the state of the slave gcxi pod:

```
kubectl get pods | grep 'gcxi-slave*'
```

The slave pod status should be Running, for example:

```
gcxi-slave-75fdb444df-z5nbq      1/1       Running     0          6d
```

If it has any other state, wait a few minutes and check again (it may take some time).

> **Important**
>
> The MicroStrategy server instance that runs in the container includes a pre-activated key, which is required for the operation of MicroStrategy. The key expires periodically; when this happens, download the latest release of the Genesys CX Insights installation package, and restart your containers using the new image.

Configuring Ingress

By default, Kubernetes does not expose any app ports publicly. To make your app accessible, you must configure a special entity called *Ingress*. As

for any Kubernetes entity, a variety of Ingress methods are supported, for more information see Kubernetes documentation. The following section provides an example of a simple case where an NGINX daemon is run on each cluster node.

Procedure: Configuring Ingress on selected ports

Purpose: Use the steps in this example procedure to configure Ingress.

Steps

Complete the following steps on the MASTER machine:

1. To deploy ingress http rules, execute the following command:

   ```
   kubectl apply -f <destination path>/ingress.yaml
   ```

   where <destination path> is the folder in which the Geneys IP is stored.

2. Execute the following commands to deploy the ingress controller infrastructure:

   ```
   kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/nginx-0.20.0/deploy/namespace.yaml && \
   kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/nginx-0.20.0/deploy/default-backend.yaml && \
   kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/nginx-0.20.0/deploy/configmap.yaml && \
   kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/nginx-0.20.0/deploy/udp-services-configmap.yaml &&
   \
   kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/nginx-0.20.0/deploy/rbac.yaml
   ```

3. Execute the following command to deploy ingress tcp rules:

   ```
   kubectl apply -f <destination path>/tcp-services-configmap.yaml
   ```

   where <destination path> is the folder in which the Geneys IP is stored. The **tcp-services-configmap.yaml** file contains the port values (see hostPort) to which your traffic will be exposed. The default values are http=80, https=443, mstr-tcp=34952, where mstr-tcp is the port used by MSTR client tools, such as Developer.

4. Execute the following command to complete the configuration:

   ```
   kubectl apply -f <destination path>/ingress-daemon.yaml
   ```

   where `<destination path>` is the folder in which the Geneys IP is stored.

```
..
docker-compose.yml
gcxi.properties
gcxi.yaml
gcxi-cleanup.yaml
gcxi-init.yaml
infra.yaml
ingress.yaml
ingress-daemon.yaml
ip_description.xml
ospatchlist.txt
postgre-mstr_hist.pgdump
postgre-mstr_meta.pgdump
read_me.html
release_notes.html
tcp-services-configmap.yaml
```

The installed Genesys CX Insights folder

Procedure: Verify Genesys CX Insights installation

Purpose: Use the steps in this procedure verify the installations. The Genesys CX Insights installation routine creates the folder shown in the figure *The installed Genesys CX Insights folder*.

Steps

After you have successfully installed Genesys CX Insights, you can verify the installation by completing the following steps:

1. Execute the following command and examine the output:

   ```
   kubectl get nodes
   ```

   The output should be similar to the following:

   ```
   NAME             STATUS    ROLES     AGE       VERSION
    spb-rhel-mstr1   Ready     master    6d        v1.11.1
    spb-rhel-mstr2   Ready     <none>    6d        v1.11.1
   ```

2. Execute the following command and examine the output:

   ```
   kubectl get pods
   ```

   The output should be similar to the following:

   ```
   NAME                            READY     STATUS      RESTARTS   AGE
    gcxi-init-2qvcd                 0/1       Completed   0          6d
    gcxi-master-587dc679c-fn2w4     1/1       Running     0          6d
    gcxi-postgres-77b7f946c-drck4   1/1       Running     0          6d (this line appears only if prepackaged PostgreSQL server
   is used)
    gcxi-slave-5d9f4485bb-d8v25     1/1       Running     1          6d
   ```

3. Execute the following command and examine the output:

   ```
   kubectl get services
   ```

The output should be similar to the following:

```
NAME            TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)              AGE
 gcxi           ClusterIP   10.98.156.54    <none>        34952/TCP,8080/TCP   6d
 gcxi-postgres  NodePort    10.101.64.127   <none>        5432:31642/TCP       6d (this line appears only if prepackaged
 PostgreSQL server is used)
 mstr-01        ClusterIP   None            <none>        34952/TCP,8080/TCP   6d
 mstr-02        ClusterIP   None            <none>        34952/TCP,8080/TCP   6d
```

4. View the Genesys CX Insights reports in MicroStrategy Web to confirm that the reports are installed, by pointing your web browser to `http://<servername>:<port>/MicroStrategy/servlet/mstrWeb`, where `<servername>` is the IP or host name of the master or slave host, and `<port>` is the port number (usually `80`).
5. Verify the schema version.
6. Verify the Genesys CX Insights Release number.
7. View the GCXI Project in MicroStrategy Developer.

> **Important**
>
> Unlike most other Genesys applications, Genesys CX Insights is not configured as an application within Genesys Configuration Server, nor is it started (or stopped) by using the Genesys Solution Control Interface.

Keep in mind that you must perform additional post-installation setup steps before actively using the report and universe elements. After completing the steps on this page, complete the following:

- Installing report editing software
- Post-Installation steps