Genesys Customer Experience Insights Deployment Guide

Installing Genesys CX Insights - Docker Compose

12/18/2025

Contents

Installing Genesys CX Insights - Docker Compose

This page describes a simplified procedure that you can optionally use to deploy Genesys CX Insights for demonstration, testing, evaluation, or development purposes; when deployed on CentOS, this method is also a suitable choice for small to medium production environments. This page describes deploying Genesys CX Insights and supporting software on Windows or CentOS using a single docker-compose file on a single virtual machine (VM).

For most production environments, Genesys recommends deploying Docker with Kubernetes or OpenShift; see Choose a deployment type and Before you install Genesys CX Insights. It is possible to install using other configurations; refer to the Docker website for information about other scenarios.

Before You Begin

Genesys CX Insights requires a suitably-prepared environment in order to operate successfully, including properly-configured installations of each of the following:

- A supported release of Microsoft Windows or CentOS. For information about what releases are supported, see the Docker website: Microsoft Windows / CentOS.
- If you plan to link Genesys CX Insights to your Info Mart, install Genesys Info Mart and RAA:
  - Genesys Info Mart release 8.5 database — The Genesys Info Mart documentation set 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 set describes how to deploy RAA, and how to configure the aggregation process.
- 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 installation packages.

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**	**gcxi.tar.gz** — contains the **gcxi** Docker image, which contains a fully installed Microstrategy

	(where XXX is the latest release number.)	Server 10.x (the latest supported release of MicroStrategy: 11.2.1, 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.
	Regular Linux IP (Linux platform) **IP_CustExpInsights_9000XXX_ENU_linux.tar.gz** (where XXX is the latest release number.)	**data.tar.gz** — contains the various YAML files (Kubernetes script files), such as gcxi.yaml, gcxi-postgres-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 (Discontinued beginning with GCXI release 9.0.019.01)	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. This image is discontinued beginning with GCXI release 9.0.019.01
MSSecEntPltf64 — MicroStrategy Secure Enterprise Platform for Windows	MicroStrategy software for Windows (server and client / editing tools) **MicroStrategy_XXX_IntelligentEnterprise_Windows_XXX.zip** (where XX is the current MicroStrategy release. For example, **MicroStrategy_11.3_IntelligentEnterprise_Windows_11.3.0560.0066.zip**.)	MicroStrategy_Secure_Enterprise_Platform_11_3_Win/cpe1705/PI
MSWrkstn — MicroStrategy Workstation	MicroStrategy Workstation software for Windows **workstation-win-ent_XXX.zip** (where XXX is the current MicroStrategy release. For example, **workstation-win-**	MicroStrategy_Workstation_11.3.63/ The MicroStrategy Workstation package is available beginning with GCXI release 100.0.029.0000

	ent_11.3.63.zip.)	

*Important:

In some releases, the names of the container images in the installation package differ from the description in the **Installation packages for GCXI** table. In these scenarios, rename the container images as described in the table **Renaming the images**:

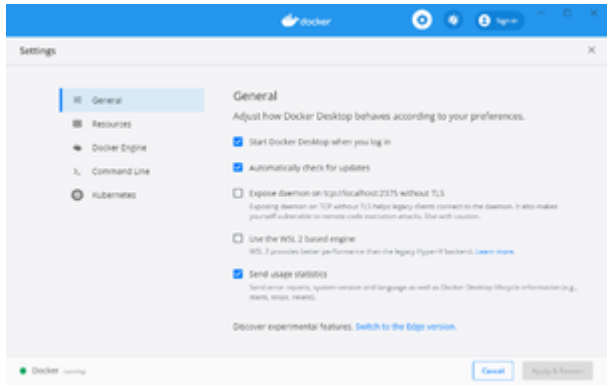
Renaming the images (if your release requires it)

Copy this file from this folder to a convenient location on your local hard drive (for example `C:\GCXI_temp`):	Rename it as:
CustExpInsights ... **dockerlinux**... **9.0.010.04.tar.gz**	**gcxi.tar.gz**
CustExpInsightsDB ... **9.0.010.04.tar.gz** (This image is discontinued beginning with GCXI release 9.0.019.01)	**gcxi_postgres.tar.gz**
CustExpInsightsOps ... **9.0.010.04.tar.gz**	**gcxi_control.tar.gz**

Note that 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.

Installing Docker

Docker Compose deployments have the same general requirements as Kubernetes deployments, with two key differences: Only one VM is required, and instead of deploying Docker and Kubernetes on Linux, you deploy only Docker, and you do so on either Linux or Windows.



Docker Desktop

Procedure: 1. Install Docker

Purpose: This procedure provides an example deployment procedure for installing Docker, without Kubernetes. This is suitable for demo, test, or development environments, not for production.

Prerequisites

- If you have previous Docker images installed, optionally back them up, as the steps in this procedure will remove them.

Steps

Installing Docker for Windows

Before you begin, ensure that your environment meets the minimum hardware requirements for Docker for Windows. At least 16 GB of RAM is

required, of which 12 GB should be available for Docker (which provides 4 GB for Hyper-V, 10 GB for the GCXI container, and 2 GB for the PostgreSQL container). More is recommended, particularly if you plan to use this deployment as a production environment.

1. Download the latest stable release of Docker Desktop, from the Docker website.
2. Open **Docker Desktop Installer** and follow the Docker Desktop for Windows instructions. When prompted during the installation process, clear the "Enable WSL2 Windows Features" option.
3. Using **Run as administrator**, start **Docker Desktop**.
4. Choose **Settings** > **General**, and disable **Use the WSL 2 based engine**.
5. Click **Apply and Restart**.

OR

Installing Docker for CentOS

Before you begin, ensure that your environment meets the OS requirements. At least 12 GB of RAM is required (10 GB for the GCXI container, and 2 GB for the PostgreSQL container). More is recommended, particularly if you plan to use this deployment as a production environment.

1. Execute the following command to verify that the **centos-extras** repository is enabled:

   ```
   sudo yum repolist
   ```

   The **centos-extras** repository is enabled by default.
2. Execute the following command to uninstall old versions:

   ```
   sudo yum remove docker \
                     docker-client \
                     docker-client-latest \
                     docker-common \
                     docker-latest \
                     docker-latest-logrotate \
   ```

```
                    docker-logrotate \
                    docker-engine
```

3. Execute the following command to install the yum-utils package and set up the stable repository:

   ```
   sudo yum install -y yum-utils
    sudo yum-config-manager \
        --add-repo \
        https://download.docker.com/linux/centos/docker-ce.repo
   ```

4. Execute the following command to install the Docker engine:

   ```
   sudo yum install docker-ce docker-ce-cli containerd.io
   ```

5. Execute the following command to start Docker:

   ```
   sudo systemctl start docker
   ```

6. Execute the following command to verify that the engine is installed and running:

   ```
   sudo systemctl status docker
   ```

7. Create a group and user, to simplify management:
   1. Execute the following command to create the group 'docker': `sudo groupadd docker`
   2. Execute the following command to add a user to the group: `sudo usermod -aG docker $USER`
      where USER is the user name for the account you will use to install and manage Docker.
   3. Execute the following command to activate the change to the group: `newgrp docker`
   4. Log out, and log in using the USER account you added to the 'docker' group.
   5. Execute the following command to verify that you can now run docker commands without using sudo: `docker run hello-world`
      A container runs, displays the 'hello world' message, and exits.

   For more information, and other options for installing Docker for CentOS, see Install Docker Engine on CentOS on the Docker web site.

Procedure: 2. Cleaning up your Docker VM

Purpose: If you have previously installed Genesys CX Insights and supporting software or are preparing to reinstall or update it, use the instructions in this procedure to clean up your Docker Virtual Machine (VM).

> ### Important
> This procedure deletes all Genesys CX Insights content, including customizations you may have made to the Genesys CX Insights Project or reports.

Steps

1. If it's not already running, start Docker (On CentOS, log in using an account in the 'docker' user group, and run `systemctl start docker`. On Windows, use **Run as administrator** to start **Docker Desktop**), and open a command terminal, such as PowerShell.
2. Execute the following command to see what containers are running:

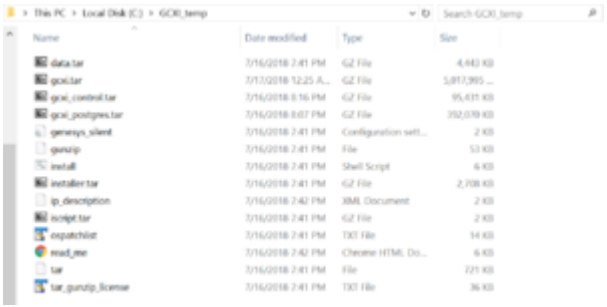
   ```
   docker ps -a
   ```

3. If any containers are running, make note of the container IDs or names, and execute the following command to remove them:

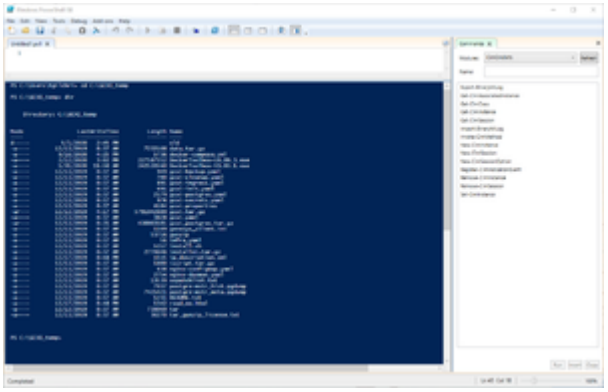
   ```
   docker rm -f <container ID or name>
   ```

4. Execute the following command to remove all existing containers, images and volumes:

   ```
   docker system prune -af --volumes
   ```



The installation package contents



GCXI_temp

Procedure: 3. Loading Docker images

Purpose: This procedure describes the steps you take to load the Genesys CX Insights installation files into Docker.

Prerequisites

- You must have available the Genesys CX Insights 9.0 installation package (IP), which includes the Genesys CX Insights repository file, containing the files listed in the table **The Genesys CX Insights installation package**; talk to your Genesys representative for information about where to download the installation packages. Genesys recommends that you save these files into a folder near the root, to make the path easier to type; for example, `C:\GCXI_temp`.

The Genesys CX Insights installation package

File	Description
gcxi.tar.gz container image	Contains a fully installed Microstrategy Server 10.x (the latest release of MicroStrategy: 10.11, for example.), providing a *stateless* deployment, where project data (reports, users, other objects) is stored separately in a MicroStrategy *meta* database. This option requires that you have configured an external PostgreSQL server.
gcxi_postgres.tar.gz container image	Contains a Genesys CX Insights Project with a PostgreSQL database server running with GCXI meta / history databases deployed; so it includes reports, users, and other objects in a single container.
data.tar.gz	Assorted deployment descriptor (YAML) files, including: • **docker-compose.yml** — A file that you use to start the solution, and that you optionally edit in some procedures.
gcxi_control.tar.gz	Contains the **gcxi_control** Docker image, which is used for deployment and configuration of the GCXI solution.

- In Release 9.0.010.04 and later, the names of the container images in the installation package differ from the description in the **The Genesys CX Insights installation package** table. If necessary, rename the container images so they match the names given in the following table:

Renaming the images in 9.0.010.04 and later

Copy the file from this folder to a convenient location on your local hard drive (for example `C:\GCXI_temp`):	Rename it as:
CustExpInsights ... **dockerlinux**... **9.0.015.01.tar.gz**	**gcxi.tar.gz**
CustExpInsightsDB ... **9.0.015.01.tar.gz**	**gcxi_postgres.tar.gz**

Steps

1. If it's not already running, start Docker (On CentOS, log in using an account in the 'docker' user group, and run `systemctl start docker`. On Windows, use **Run as administrator** to start **Docker Desktop**), and open a command terminal, such as PowerShell.
2. Change the current directory to the folder where you saved the Genesys CX Insights installation files. For example:

   ```
   cd C:\GCXI_temp
   ```

3. In release 9.0.019.00 and earlier, execute the following commands to load the PostgreSQL Docker image:

   ```
   docker load -i gcxi_postgres.tar.gz
   ```

4. Execute the following commands to load the Docker images:

   ```
   docker load -i gcxi.tar.gz
   ```

5. Execute the following command to verify that the images loaded correctly:

   ```
   docker images
   ```

   The console lists the installed Docker images:

   ```
   $ docker images
   REPOSITORY                                                TAG                 IMAGE ID            CREATED             SIZE
   pureengage-docker-production.jfrog.io/gcxi/gcxi           9.0.015.01          e7a7216f2f2f        4 months ago        11.7GB
   pureengage-docker-production.jfrog.io/gcxi/gcxi_postgres  9.0.015.01          068b8c6ba06c        4 months ago        3.53GB
   ```

6. Execute the following commands to retag the images (note that the PostgreSQL image is not used in release 9.0.019.01 and later):

```
docker tag <REPOSITORY>/gcxi:<RELEASE> gcxi

docker tag <REPOSITORY>/gcxi_postgres:<RELEASE> gcxi_postgres
```

   where:
   <REPOSITORY> is the full repository path shown in the preceding step (such as pureengage-docker-production.jfrog.io/gcxi/). The repository path varies depending on the release, and is not present or required in some releases.

   <RELEASE> is the a string corresponding to the release you are installing (such as 9.0.015.01),

   For example:

```
docker tag pureengage-docker-production.jfrog.io/gcxi/gcxi:9.0.019.01 gcxi

docker tag pureengage-docker-production.jfrog.io/gcxi/gcxi_postgres:9.0.019.01 gcxi_postgres
```

7. Execute the following command to verify that the images loaded correctly, and have correct tagging:

```
docker images
```

   The console lists the Docker images:

```
$ docker images
REPOSITORY                                                TAG                 IMAGE ID            CREATED             SIZE
gcxi                                                      latest              e7a7216f2f2f        4 months ago        11.7GB
pureengage-docker-production.jfrog.io/gcxi/gcxi           9.0.015.01          e7a7216f2f2f        4 months ago        11.7GB
gcxi_postgres                                             latest              068b8c6ba06c        4 months ago        3.53GB
pureengage-docker-production.jfrog.io/gcxi/gcxi_postgres  9.0.015.01          068b8c6ba06c        4 months ago        3.53GB
```

   Compare the result to the figure; each image must have a name in the REPOSITORY column with no preceding path, and a value of LATEST in the TAG column. Note that each image appears twice in the list; this is expected behavior, because each one has two tags.

> **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 on the last day of each year; when this happens, download the latest release of the Genesys CX Insights installation package, and restart your containers using the new image.

Procedure: 4. Specify a database

Purpose: Tell Genesys CX Insights what database to use — either the included sample / demo database, or your external Genesys Info Mart database.

Steps

Choose one the of the following methods:

Use the provided sample / demo database

This method uses the images **gcxi** and **gcxi_postgres**.

1. Open the (CustExpInsights\linux\b1\ip\) **data.tar.gz** package, and copy the file **docker-compose.yml** into the folder where you stored the installation package, for example **C:\GCXI_temp**.
2. Open the **docker-compose.yml** file for editing.

3. In the **services:** section, below the line that begins *old version of gcxi-postgres service*, uncomment the following lines:

```
# gcxi-postgres:
   #   image: gcxi_postgres:9.0.015.00
   #   hostname: gcxi-postgres
   #   volumes:
   #   - "gcxi_postgres:/var/lib/postgresql/data"
   #   ports:
   #   - "5432:5432"
   #   networks:
   #   - gcxi
```

   On Windows deployments, the port 8080 is sometimes used by another process. In this scenario, change the port mapping; Edit the line **8080:8080**, changing the first value to **<unused_ port_in_windows>:8080", where <unused_ port_in_windows> is an unused port, for example** 8280:8080**. If you remap this port, be sure to use the new port value when accessing MicroStrategy web interface.**

4. Comment out from the line that begins *new version of gcxi-postgres service*, down to the end of the **gcxi-postgres:** and **gcxi-control:** sections. Leave **gcxi-0:** and subsequent sections uncommented.
5. If you are connecting the GCXI Docker Compose deployment to a demo Info Mart database that resides in a PostgreSQL container, comment out the DSNDEF lines.
6. Save the **docker-compose.yml** file.

OR

Connect to your Genesys Info Mart database

You must have available all relevant Genesys Info Mart information, including the RDBMS type (Microsoft SQL Server, PostgreSQL, Oracle), hostname, and user credentials. This method uses the images **gcxi** and **gcxi_control**.

1. Change the current directory to the folder where you saved the Genesys CX Insights installation files. For example:

```
cd C:\GCXI_temp
```

2. Execute the following command to load the gcxi_control Docker image:

   ```
   docker load -i gcxi_control.tar.gz
   ```

3. Execute the following command to retag the image:

   ```
   docker tag <REPOSITORY>/gcxi_control:<RELEASE> gcxi_control
   ```

   where:
   <REPOSITORY> is the full repository path shown in the preceding step (such as pureengage-docker-production.jfrog.io/gcxi/). The repository path varies depending on the release, and is not present or required in some releases.

   <RELEASE> is the a string corresponding to the release you are installing (such as 9.0.015.01),

   For example:

   ```
   docker tag pureengage-docker-production.jfrog.io/gcxi/gcxi_control:9.0.015.01 gcxi_control
   ```

4. Open the (CustExpInsights\linux\b1\ip\) **data.tar.gz** package, and copy the file **docker-compose.yml** into the folder where you stored the installation package, for example **C:\GCXI_temp**.

5. Open the *docker-compose.yml* file for editing.

6. Some lines in the file are commented out with a single #. Uncomment lines with **DSNDEF_** variables, or those with **GIM_** variables as appropriate, and populate them with suitable values. Note that values of Database type (DB_TYPE) and Database Type extended (DB_TYPE_EX) are allowed, as described in the **docker-compose.yml** file. For example, if you plan to use your own MSSQL Info Mart database, instead of the built-in that comes as part of the container, starting with the line that begins *environment*, uncomment the lines that have a single #:

   ```
   ## DSNDEF* is a new DSN definition format suitable for GCXI v. >= 9.0.010.00
   ## If at least one DSNDEF* variable is defined, GIM_* variables are ignored
   ## Each DSNDEF node represents one DSN definition
   ## As of GCXI v. 9.0.010.00 DSN_NAMEs must be predefined:
   ## GCXI_GIM_DB = for project GCXI / CX Insights
   ## IWD_DB = for project IWD
   ## DSNs defined with other names will be created in MSTR, but not used by default
   ## Password notice: if GIM password contains semicolon, it must be escaped with \
   ```

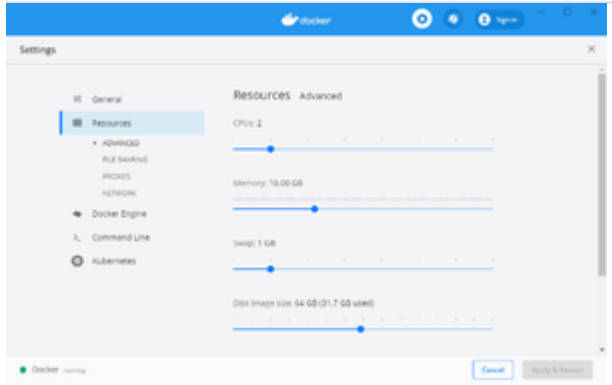
```
## Eg: PASSWORD=my;passwd;.. => PASSWORD=my\;passwd;..
## NB: password notice does not apply to old GIM_* DSN format, no need to escape anything there
## For DB_TYPE and DB_TYPE_EX values see explanation below
          - DSNDEF1=DSN_NAME=GCXI_GIM_DB;DB_TYPE=SQLSERVER;DB_TYPE_EX=Microsoft SQL Server
2012;HOST=gi2-qadb;PORT=1433;DB_NAME=gim85test2voice;LOGIN=gim85test2voice;PASSWORD=gim85test2voice
          - DSNDEF2=DSN_NAME=SOME_NAME;DB_TYPE=SQLSERVER;DB_TYPE_EX=Microsoft SQL Server
2012;HOST=gi2-qadb;PORT=1433;DB_NAME=gim85test2mm;LOGIN=gim85test2mm;PASSWORD=gim85test2mm
          -
DSNDEF3=DSN_NAME=IWD_DB;DB_TYPE=POSTGRESQL;DB_TYPE_EX=PostgreSQL;HOST=gi2-cent7-2;PORT=5435;DB_NAME=gim;LOGIN=gim_db;PASSWORD=gim_db
## Legacy GIM_* syntax - for v. < 9.0.010.00
## Database Type: values allowed: SQLSERVER POSTGRESQL ORCLW
          # - GIM_DB_TYPE=POSTGRESQL
## Database Type extended  (must correspond Type above), values allowed:
## 'Microsoft SQL Server 2012' 'Microsoft SQL Server 2014' 'Microsoft SQL Server 2016'
## 'PostgreSQL'
## 'Oracle 12cR2'  'Oracle 18c'  'Oracle 19c'
## sometimes these types change with new MSTR release
## if values above don't work (e. g. outdated), refer to file '$MSTR_INSTALL_HOME/install/DATABASE.PDS'
## in this file MSTR keeps DB type aliases for the current release
## search for 'DSSOBJECT' element, 'NAME' attribute
## another way: try to create DB Connection in MSTR Developer, and refer to the list of values it suggests
          # - GIM_DB_TYPE_EX=PostgreSQL
          # - GIM_HOST=gi2-qadb
          # - GIM_PORT=5432
## For Postgre and MS SQL this is GIM database name, for Oracle this is not set
          # - GIM_DB=gim85test2voice
## GIM Oracle SID - for GIM Oracle only (set either SID or Service name)
          # - GIM_ORCL_SID=
## GIM Oracle Service name - for GIM Oracle only (set either SID or Service name)
          # - GIM_ORCL_SNAME=
          # - GIM_LOGIN=gim85test2voice
          # - GIM_PASSWORD=gim85test2voice
```

7. Save the **docker-compose.yml** file.

Tip

- When starting the container, if you encounter an error about "exited with status 1", verify that all variables listed above are correctly populated.
- If you are connecting the GCXI Docker Compose deployment to a demo Info Mart database that resides in a PostgreSQL container, comment out the DSNDEF lines.
- If you are connecting to a Named Instance on a Microsoft SQL Server DBMS using DSNDEF environment variables, add 4 backslashes between the hostname and the instance name:

```
HOST=<serverName>\\\\<InstanceName>
```



Docker Desktop Memory Settings

Procedure: 5. Configure memory settings (Windows deployments)

Purpose: This procedure describes the steps you take to configure the virtual environment on Windows deployments.

Steps

1. Open Hyper-V Manager, and change the memory settings for the virtual machine:
    1. Click **Turn Off** to stop the virtual machine.
    2. Open the **Settings** of your virtual machine, and in the section **Memory** > **Hardware**, mark the checkbox **Enable Dynamic Memory**, and click **OK**..
    3. Click **Start** to start the virtual machine.
    4. Close Hyper-V Manager.
2. Open Docker Desktop, and change the memory settings for Docker:
    1. Click **Settings** > **Resources**.
    2. Increase the value of **Memory** to 10GB or more.
    3. Click **Apply & Restart**.

Procedure: 6. Starting Genesys CX Insights containers

Purpose: This procedure describes the steps you take to start, stop, or reset the containers.

Steps

1. If you haven't already done so, open the (CustExpInsights\linux\b1\ip\) **data.tar.gz** package, and copy the file **docker-compose.yml** into the folder where you stored the installation package, for example **C:\GCXI_temp**.
2. If it's not already running, start Docker (On CentOS, log in using an account in the 'docker' user group, and run `systemctl start docker`. On Windows, use **Run as administrator** to start **Docker Desktop**), and open a command terminal, such as PowerShell.
3. Change the current directory to the location where the Genesys CX Insights installation package is stored, for example:

   ```
   cd C:\GCXI_temp
   ```

4. To start the whole solution, enter the following command:

   ```
   docker-compose -f docker-compose.yml up
   ```

   After several minutes (as few as two, but sometimes more than ten, depending on your environment), summary information appears in the console:

   ```
   gcxi-0_1        | Attempt to start PDF Export Service...
   gcxi-0_1        | PDF Export Service is started.
   ...
   gcxi-0_1        | Tomcat started.
   gcxi-0_1        | USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
   gcxi-0_1        | root      1459  3.0  0.0  51760  3864 ?        R    16:33   0:00 ps -auxw --sort pmem
   gcxi-0_1        | root         1  0.0  0.0  12932  4032 ?        Ss   16:32   0:00 /bin/bash /genesys/gcxi/mstr_start.sh
   ...
   ```

   [followed by several more lines]

   > **Tip**

> After docker-compose.yml completes, you will not see a blinking insertion point in the terminal. This is expected behavior. Once you see the output shown above, which may be followed by additional lines, you can open another terminal and proceed to the next step Accessing MicroStrategy web interface.

The containers are now loaded and ready to use.

Procedure: 7. Accessing MicroStrategy web interface

Purpose: Use the following procedure to access the various web interfaces, where you can view the reports and dashboards or manage the software.

Prerequisites

If you remapped the 8080 port in **Procedure: 4. Specify a database**, be sure to use the new port value in this procedure, instead of 8080.

Steps

1. Open a command terminal, such as PowerShell.
2. Execute the following command to learn the IP address of the VM:
   Windows:

   ```
   ipconfig
   ```

CentOS:

```
ip addr show
```

Mac:

```
localhost
```

3. To access the various web interfaces, enter the following addresses (where <VM IP> is the IP address you obtained in the preceding step):
   - To view reports and dashboards, visit `http://<VM IP>:8080/MicroStrategy/servlet/mstrWeb`, and log in as an administrator.
   - To manage users and security roles, visit `http://<VM IP>:8080/MicroStrategy/servlet/mstrServerAdmin`, and log in as an administrator.

     Where <VM IP> is the hostname or ip-address where docker-compose is running.
   - To manage MSTR Web server settings, visit `http://<VM IP>:8080/MicroStrategy/servlet/mstrWebAdmin`, and log in as `admin`.

Procedure: 8. Installing MicroStrategy tools

Purpose: If you are setting up a development environment, you can use the instructions in this section to add administrative tools. This is required only if you expect to use these tools.

Prerequisites

Ensure that you have a copy of the latest MicroStrategy tools (for example, **MicroStrategy_11.1_Windows.zip**).

Steps

1. Extract the .zip archive to a temporary location on your hard drive, and run the **MICROSTRATEGY.EXE** file, to start the MicroStrategy Installation WIzard.
2. Follow the steps in the installation wizard.
   On the **Select Components** tab, clear all components *except* the following:
   - **MicroStrategy Developer Products**
   - **MicroStrategy Object Manager**
   - **MicroStrategy Command Manager**
   - **MicroStrategy Integrity Manager**
   - **MicroStrategy System Manager**
   - **MicroStrategy Analytics Module**.

   > **Important**
   > Install only the components indicated here. If you install additional components, you may encounter installation difficulties or performance issues.

Managing your environment

Use the procedures in this section to manage the environment.

Procedure: Stopping Genesys CX Insights containers and resetting the environment

Purpose: If, for any reason, you need to stop the containers, use the instructions in this section to stop the containers and clear any customizations.

Steps

1. Open a command terminal, such as PowerShell.
2. Change the current directory to the folder where you saved the Genesys CX Insights installation files. For example:

   ```
   cd C:\GCXI_temp
   ```

3. To stop the containers, execute the following command:

   ```
   docker-compose -f docker-compose.yml down
   ```

   Note that when you stop the containers, your customizations are preserved, and will still be available when you restart the containers.

4. To clear any customizations, and restore everything to the original state, first stop the containers, and then execute the following command:

   ```
   docker volume rm gcxi_mstr_log_01 gcxi_mstr_shared gcxi_gcxi_postgres
   ```

5. To restart the containers, execute the following command:

   ```
   docker-compose -f docker-compose.yml up
   ```

After installation

See Accessing CX Insights GUIs for information about accessing the reports using MicroStrategy Web. Note that, in the demo / sample database, data is available for a limited time period:

- For Genesys CX Insights reports, September 2015 to October 2016.
- For Genesys CX Insights for iWD reports, February 12, 2019 to February 21, 2019.

When you run reports in such environments, choose dates within that range, or simply remove the default value from the first prompt (Pre-set Date/Day) before you run the report.