# Installing Genesys CX Insights - Docker Compose

| Prepared by | Reviewer | Date | Version |
|---|---|---|---|
| Kanalarasan | Prabhakaran/Yuvaraj | 18-04-2022 | Draft |
| Kanalarasan | Prabhakaran/Yuvaraj | 29-04-2022 | Draft 2 |
| Kanalarasan | Prabhakaran/Yuvaraj | 06-05-2022 | 1.0 |

# Table of Contents

# About GCXI

Genesys Customer Experience Insights (CX Insights) draws aggregated historical information from the Genesys Info Mart data warehouse and presents the data in readable reports to enable business and contact centre managers to make better business decisions for streamlining operations, reducing costs, and providing better services.

This document describes a simplified steps to deploy **GCXI** using **Docker Compose** model.

# Prerequisites:

## GCXI Product Alert

Click here to check - GCXI product alert and its compatible components.

## System Requirements

| OS Family | OS | Release | Conditions |
|---|---|---|---|
| Linux | CentOS Linux 7 | 9.0+ | Note the following:<br>-CentOS Linux 7.5 is supported starting with GCXI 9.0.007.03<br>-CentOS Linux 7.9 is supported starting with GCXI 9.0.016.02 |
| Linux | CentOS Linux 8 | 9.0+ | Starting with GCXI 9.0.019.01 |
| Linux | RHEL 7 | 9.0+ | Note the following:<br>-RHEL7.5 is supported starting with GCXI 9.0.007.03<br>-RHEL 7.9 is supported starting with GCXI 9.0.016.02 |
| Linux | RHEL 8 | 9.0+ | Starting with GCXI 9.0.019.01 |

- For Linux distributions, we recommended to have **16 GB RAM** and **100 GB of disk space (40 GB free space needed for /var/lib/docker)** to accommodate GCXI. More is recommended, particularly if you plan to use this deployment as a production environment.
- Refer **Supported Operating Systems** and make sure to install the compatible components.
- Miss match installation leads to product start up failure and will create issues.
- Basic Linux administration skill is required to install docker and GCXI product.

## System Preparation

1. Keep your system up to date:

   $ yum update

```
[root@localhost ~]# yum update
Last metadata expiration check: 2:27:46 ago on Fri 29 Apr 2022 01:31:47 PM IST.
Dependencies resolved.
================================================================================
 Package                                 Architecture            Version
================================================================================
Installing:
 kernel                                  x86_64                  5.14.0-80.el9
Upgrading:
 NetworkManager                          x86_64                  1:1.39.2-1.el9
 NetworkManager-adsl                     x86_64                  1:1.39.2-1.el9
 NetworkManager-bluetooth                x86_64                  1:1.39.2-1.el9
 NetworkManager-config-server            noarch                  1:1.39.2-1.el9
 NetworkManager-initscripts-updown       noarch                  1:1.39.2-1.el9
 NetworkManager-libnm                    x86_64                  1:1.39.2-1.el9
 NetworkManager-team                     x86_64                  1:1.39.2-1.el9
 NetworkManager-tui                      x86_64                  1:1.39.2-1.el9
 NetworkManager-wifi                     x86_64                  1:1.39.2-1.el9
```

2. 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:

   a) 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

   b) Execute the following command:

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

   c) Then verify the values:

      $ cat /etc/sysctl.conf | grep -i -e "kernel.sem" -e "vm.max_map_count"

```
[root@localhost ~]# echo "kernel.sem = 250 1024000 250 4096" >> /etc/sysctl.conf
[root@localhost ~]#
[root@localhost ~]# echo "vm.max_map_count = 5242880" >> /etc/sysctl.conf
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# cat /etc/sysctl.conf | grep -i -e "kernel.sem" -e "vm.max_map_count"
kernel.sem = 250 1024000 250 4096
vm.max_map_count = 5242880
[root@localhost ~]#
```

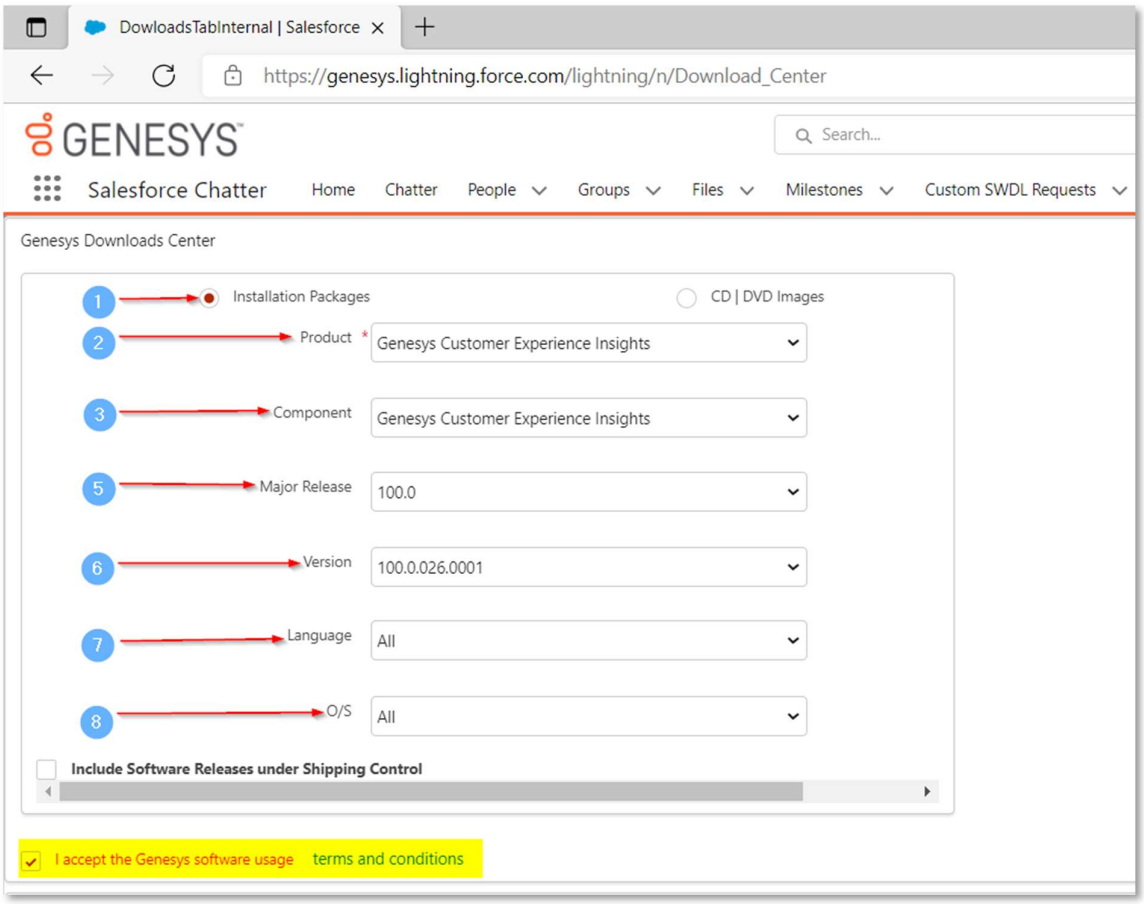
   d) Reboot the machine.

## Package Repository

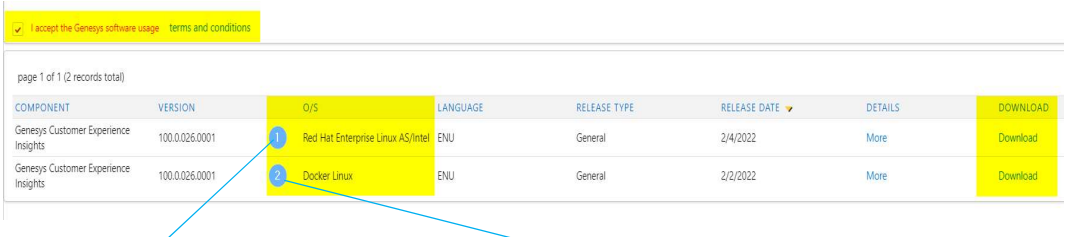
Latest Package available in Download Centre

Steps to Download :

1. Go to Download Centre
2. Choose/filter below options to download GCXI (Always download the latest version of GCXI)

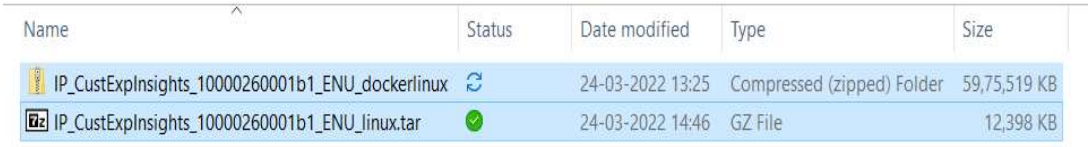



3. Once you checked Genesys Software Terms and Conditions you able to download below packages. Download both **RHEL Linux AS/Intel & Dcoker Linux** packages.

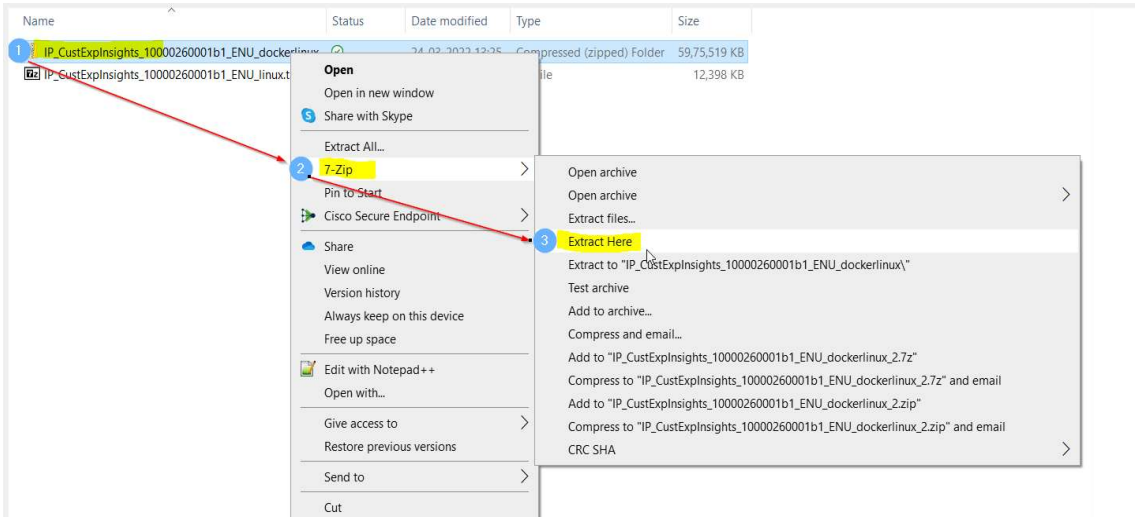



| RHEL Linux AS/Intel | Docker Linux |
|---|---|
| Package which contains .yml file | Contains GCXI & GCXI Control images |

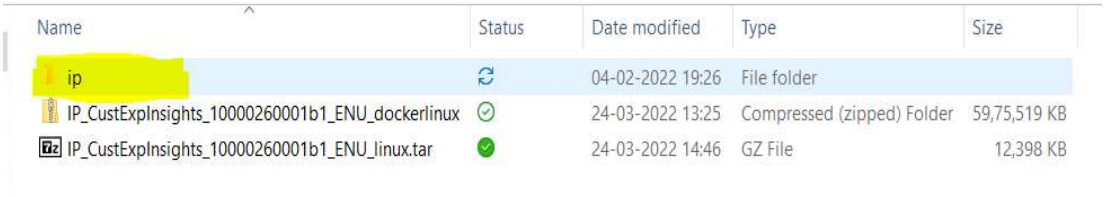
Your downloaded packages would look like as shown below




4. Extract the package **IP_CustExpInsights_10000260001b1_ENU_dockerlinux**

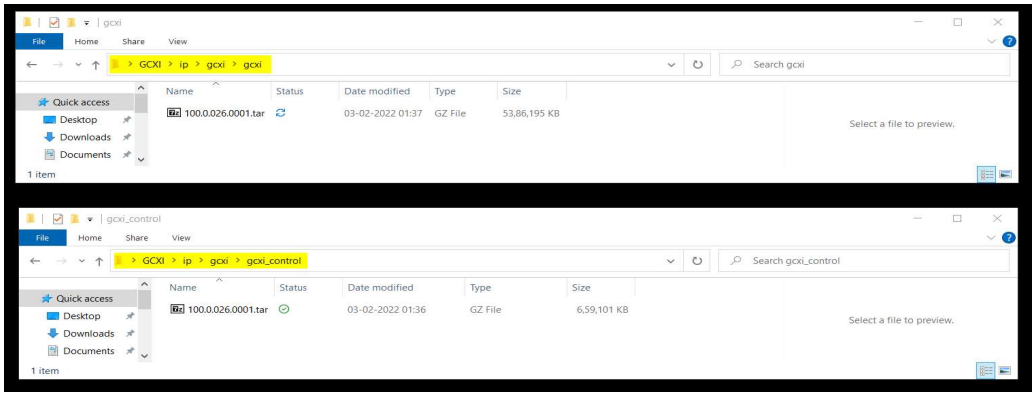



Now you'll get folder called **IP** which contains GCXI and GCXI Control images




Go to → **IP** folder then → **GCXI**

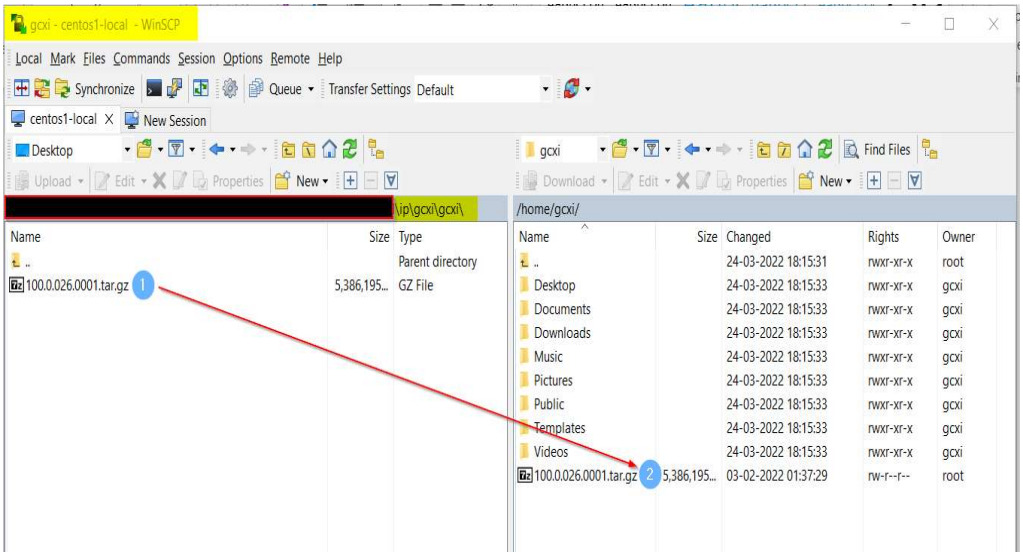
Each GCXI and GCXI_Contorl has its own .tar file which needs to be copied to linux machine.



## Copy and Renaming Files

Copy .tar files from gcxi and gcxi_control folder into Linux Machine using FTP tools **WinSCP** or **FileZila**

1. Copy **100.0.026.0001.tar.gz** file from **\ip\gcxi\gcxi** to your desired location of Linux Machine




**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:

| **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 |
| CustExpInsightsOps ... 9.0.010.04.tar.gz | gcxi_control.tar.gz |

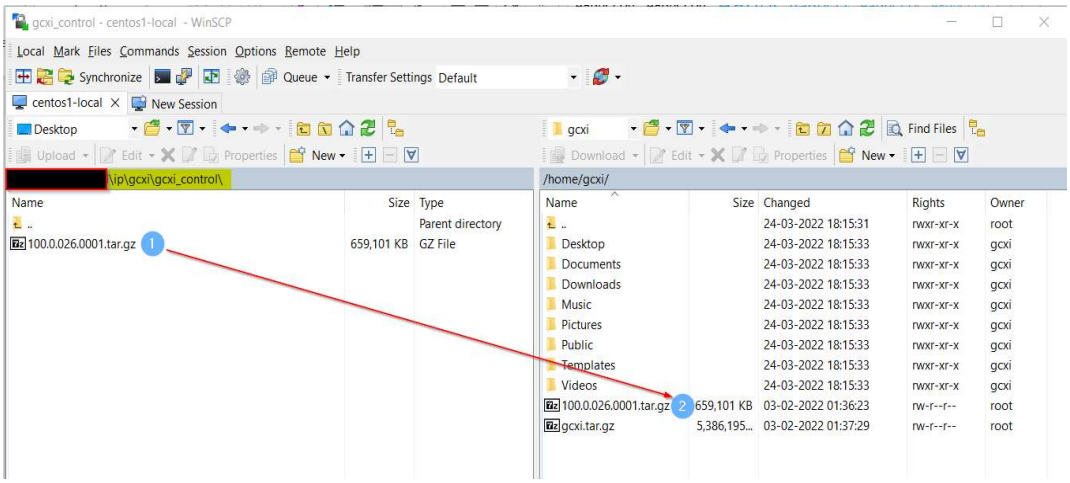
Rename the file as soon as you copied into Linux machine

$ mv 100.0.026.0001.tar.gz gcxi.tar.gz

$ ls -lrth | grep -i gcxi.tar.gz

```
[gcxi@localhost ~]$ mv 100.0.026.0001.tar.gz gcxi.tar.gz
[gcxi@localhost ~]$ ls -lrth | grep -i gcxi.tar.gz
-rw-r--r--. 1 root root 5.2G Feb  3 01:37 gcxi.tar.gz
[gcxi@localhost ~]$
```

2. Now copy **100.0.026.0001.tar.gz** from **\ip\gcxi\gcxi_control** to Linux Machine



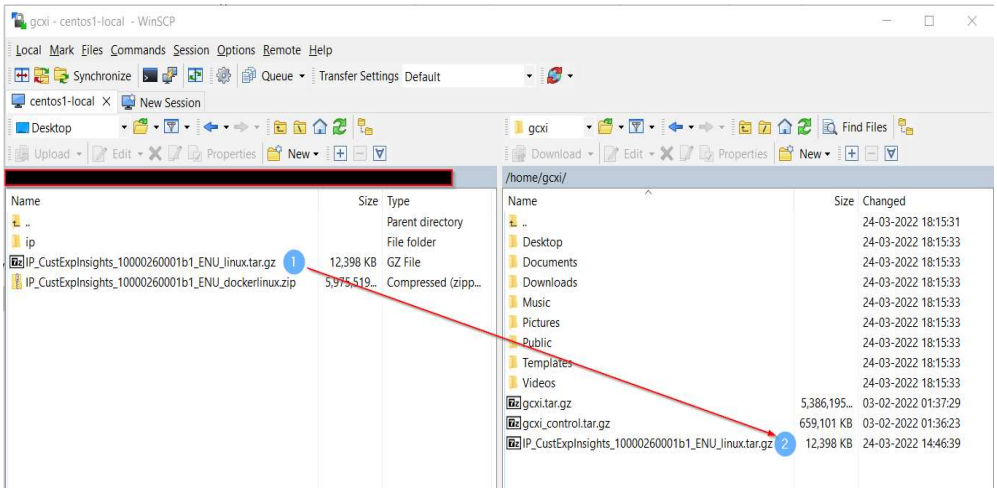

3. Rename the file as shown below

$ mv 100.0.026.0001.tar.gz gcxi_control.tar.gz

$ ls -lrth | grep -i gcxi_control.tar.gz

```
[gcxi@localhost ~]$
[gcxi@localhost ~]$ mv 100.0.026.0001.tar.gz gcxi_control.tar.gz
[gcxi@localhost ~]$ ls -lrth | grep -i gcxi_control.tar.gz
-rw-r--r--. 1 root root 644M Feb  3 01:36 gcxi_control.tar.gz
[gcxi@localhost ~]$
```

4. Then Copy **IP_CustExpInsights_10000260001b1_ENU_linux.tar.gz** to Linux machine



Once copied validate the files using below command

$ ls -lrth | grep -i -e gcxi_control.tar.gz -e gcxi.tar.gz -e IP_CustExpInsights_10000260001b1_ENU_linux.tar.gz

```
[gcxi@localhost ~]$
[gcxi@localhost ~]$ ls -lrth | grep -e gcxi_control.tar.gz -e gcxi.tar.gz -e IP_CustExpInsights_10000260001b1_ENU_linux.tar.gz
-rw-r--r--. 1 root root 644M Feb  3 01:36 gcxi_control.tar.gz
-rw-r--r--. 1 root root 5.2G Feb  3 01:37 gcxi.tar.gz
-rw-r--r--. 1 root root  13M Mar 24 14:46 IP_CustExpInsights_10000260001b1_ENU_linux.tar.gz
[gcxi@localhost ~]$
```

# Docker Installation

## Installing Docker for CentOS

1. Execute the following command to verify that the **centos-extras** repository is enabled:
   Note:- Use **sudo,** only when you're not logged in as super user.
   $ yum repolist

```
[root@localhost ~]# yum repolist
repo id                                                                repo name
appstream                                                              CentOS Stream 9 - AppStream
baseos                                                                 CentOS Stream 9 - BaseOS
docker-ce-stable                                                       Docker CE Stable - x86_64
extras-common                                                          CentOS Stream 9 - Extras packages
[root@localhost ~]#
```

   The centos-extras repository is enabled by default.

2. Execute the following command to uninstall old versions:

```
$ yum remove docker \docker-client \
        docker-client-latest \
        docker-common \
        docker-latest \
        docker-latest-logrotate \
        docker-logrotate \
        docker-engine
```

```
[root@localhost ~]# yum remove docker \docker-client \
                  docker-client-latest \
                  docker-common \
                  docker-latest \
                  docker-latest-logrotate \
                  docker-logrotate \
                  docker-engine
No match for argument: docker
No match for argument: docker-client
No match for argument: docker-client-latest
No match for argument: docker-common
No match for argument: docker-latest
No match for argument: docker-latest-logrotate
No match for argument: docker-logrotate
No match for argument: docker-engine
No packages marked for removal.
Dependencies resolved.
Nothing to do.
Complete!
[root@localhost ~]#
```

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

$ yum install -y yum-utils

```
[root@localhost ~]# yum install -y yum-utils
Last metadata expiration check: 3:22:51 ago on Fri 29 Apr 2022 01:31:47 PM IST.
Dependencies resolved.
================================================================================
 Package                    Architecture                Version
================================================================================
Installing:
 yum-utils                  noarch                      4.0.24-3.el9

Transaction Summary
================================================================================
Install  1 Package

Total download size: 42 k
Installed size: 23 k
Downloading Packages:
yum-utils-4.0.24-3.el9.noarch.rpm
--------------------------------------------------------------------------------
Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :
  Installing       : yum-utils-4.0.24-3.el9.noarch
  Running scriptlet: yum-utils-4.0.24-3.el9.noarch
  Verifying        : yum-utils-4.0.24-3.el9.noarch

Installed:
  yum-utils-4.0.24-3.el9.noarch

Complete!
[root@localhost ~]#
```

$ yum-config-manager \

--add-repo \

https://download.docker.com/linux/centos/docker-ce.repo

```
[root@localhost ~]# yum-config-manager \
    --add-repo \
    https://download.docker.com/linux/centos/docker-ce.repo
Adding repo from: https://download.docker.com/linux/centos/docker-ce.repo
[root@localhost ~]#
```

4. Execute the following command to install the Docker engine:
   $ yum install docker-ce docker-ce-cli containerd.io

```
[root@localhost ~]# yum install docker-ce docker-ce-cli containerd.io
Last metadata expiration check: 0:00:17 ago on Fri 29 Apr 2022 04:59:20 PM IST.
Dependencies resolved.
================================================================================
 Package                    Architecture    Version                  Repository
================================================================================
Installing:
 containerd.io              x86_64          1.5.11-3.1.el7           docker-ce-stable
 docker-ce                  x86_64          3:20.10.14-3.el7         docker-ce-stable
 docker-ce-cli              x86_64          1:20.10.14-3.el7         docker-ce-stable
Installing dependencies:
 docker-ce-rootless-extras  x86_64          20.10.14-3.el7           docker-ce-stable
 docker-scan-plugin         x86_64          0.17.0-3.el7             docker-ce-stable

Transaction Summary
================================================================================
Install  5 Packages

Total download size: 93 M
Installed size: 377 M
Is this ok [y/N]: y
Downloading Packages:
(1/5): docker-ce-20.10.14-3.el7.x86_64.rpm
(2/5): docker-ce-cli-20.10.14-3.el7.x86_64.rpm
(3/5): containerd.io-1.5.11-3.1.el7.x86_64.rpm
(4/5): docker-scan-plugin-0.17.0-3.el7.x86_64.rpm
(5/5): docker-ce-rootless-extras-20.10.14-3.el7.x86_64.rpm
--------------------------------------------------------------------------------
Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
```

```
Running transaction
  Preparing        :
  Installing       : docker-scan-plugin-0.17.0-3.el7.x86_64
  Running scriptlet: docker-scan-plugin-0.17.0-3.el7.x86_64
  Installing       : docker-ce-cli-1:20.10.14-3.el7.x86_64
  Running scriptlet: docker-ce-cli-1:20.10.14-3.el7.x86_64
  Installing       : containerd.io-1.5.11-3.1.el7.x86_64
  Running scriptlet: containerd.io-1.5.11-3.1.el7.x86_64
  Installing       : docker-ce-rootless-extras-20.10.14-3.el7.x86_64
  Running scriptlet: docker-ce-rootless-extras-20.10.14-3.el7.x86_64
  Installing       : docker-ce-3:20.10.14-3.el7.x86_64
  Running scriptlet: docker-ce-3:20.10.14-3.el7.x86_64
  Verifying        : containerd.io-1.5.11-3.1.el7.x86_64
  Verifying        : docker-ce-3:20.10.14-3.el7.x86_64
  Verifying        : docker-ce-cli-1:20.10.14-3.el7.x86_64
  Verifying        : docker-ce-rootless-extras-20.10.14-3.el7.x86_64
  Verifying        : docker-scan-plugin-0.17.0-3.el7.x86_64

Installed:
  containerd.io-1.5.11-3.1.el7.x86_64          docker-ce-3:20.10.14-3.el7.x86_64       docker-ce-cli-1:20.10.14-3.el7.x86_64       docker-ce-rootle
  docker-scan-plugin-0.17.0-3.el7.x86_64

Complete!
[root@localhost ~]#
```

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

   $ systemctl start docker

   $ systemctl status docker

```
[root@localhost ~]# systemctl start docker
[root@localhost ~]#
[root@localhost ~]# systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)
     Active: active (running) since Fri 2022-04-29 17:05:33 IST; 1min 5s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 33358 (dockerd)
      Tasks: 10
     Memory: 31.9M
        CPU: 482ms
     CGroup: /system.slice/docker.service
             └─33358 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Apr 29 17:05:33 localhost.localdomain dockerd[33358]: time="2022-04-29T17:05:33.351850799+05:30" level=
Apr 29 17:05:33 localhost.localdomain dockerd[33358]: time="2022-04-29T17:05:33.465643421+05:30" level=
Apr 29 17:05:33 localhost.localdomain dockerd[33358]: time="2022-04-29T17:05:33.564692041+05:30" level=
Apr 29 17:05:33 localhost.localdomain dockerd[33358]: time="2022-04-29T17:05:33.653881972+05:30" level=
Apr 29 17:05:33 localhost.localdomain dockerd[33358]: time="2022-04-29T17:05:33.709035538+05:30" level=
Apr 29 17:05:33 localhost.localdomain dockerd[33358]: time="2022-04-29T17:05:33.774801150+05:30" level=
Apr 29 17:05:33 localhost.localdomain dockerd[33358]: time="2022-04-29T17:05:33.793516294+05:30" level=
Apr 29 17:05:33 localhost.localdomain dockerd[33358]: time="2022-04-29T17:05:33.793617169+05:30" level=
Apr 29 17:05:33 localhost.localdomain systemd[1]: Started Docker Application Container Engine.
Apr 29 17:05:33 localhost.localdomain dockerd[33358]: time="2022-04-29T17:05:33.816123574+05:30" level=
[root@localhost ~]#
```

6. Create a group and user, to simplify management (**Optional steps, if you're planning to deploy and manage the containers using normal user privilege**):

   a) Execute the following command to create the group 'docker':

   $ groupadd docker

```
[root@localhost ~]# groupadd docker
groupadd: group 'docker' already exists
[root@localhost ~]#
```

   b) Execute the following command to add a user to the **docker** group:

   $ usermod -aG docker gcxi

```
[root@localhost ~]# usermod -aG docker gcxi
[root@localhost ~]#
```

   where **"gcxi"** is the user name for the account you will use to install and manage Docker.

   c) Execute the following command to activate the change to the group:

   $ newgrp docker

```
[root@localhost ~]# newgrp docker
[root@localhost ~]#
```

d) Log out, and log in using the "**gcxi**" account you added to the 'docker' group.

```
[root@localhost ~]# su - gcxi
[gcxi@localhost ~]$
[gcxi@localhost ~]$ id
uid=1000(gcxi) gid=1000(gcxi) groups=1000(gcxi),974(docker) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[gcxi@localhost ~]$
```

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

## Installation of docker-compose

**Docker Compose** is a tool that allows you to run multi-container application environments based on definitions set in a **YAML file**. It uses service definitions to build fully customizable environments with multiple containers that can share networks and data volumes.

Following commands you have to run as **root**.

1. Run this command to download the current stable release of Docker Compose:
   $ curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

```
[root@localhost ~]# curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   664  100   664    0     0   2886      0 --:--:-- --:--:-- --:--:--  2886
100 12.1M  100 12.1M    0     0  3269k      0  0:00:03  0:00:03 --:--:-- 3499k
[root@localhost ~]#
```

2. Apply executable permissions to the binary:
   $ chmod +x /usr/local/bin/docker-compose

```
[root@localhost ~]#
[root@localhost ~]# chmod +x /usr/local/bin/docker-compose
[root@localhost ~]#
```

   **Note**: If the command docker-compose fails after installation, check your path. You can also create a symbolic link to /usr/bin or any other directory in your path.
   For example:
   $ ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose

```
[root@localhost ~]#
[root@localhost ~]# ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
[root@localhost ~]#
```

3. Test the installation.

$ docker-compose --version

```
[root@localhost ~]#
[root@localhost ~]# docker-compose --version
docker-compose version 1.29.2, build 5becea4c
[root@localhost ~]#
```

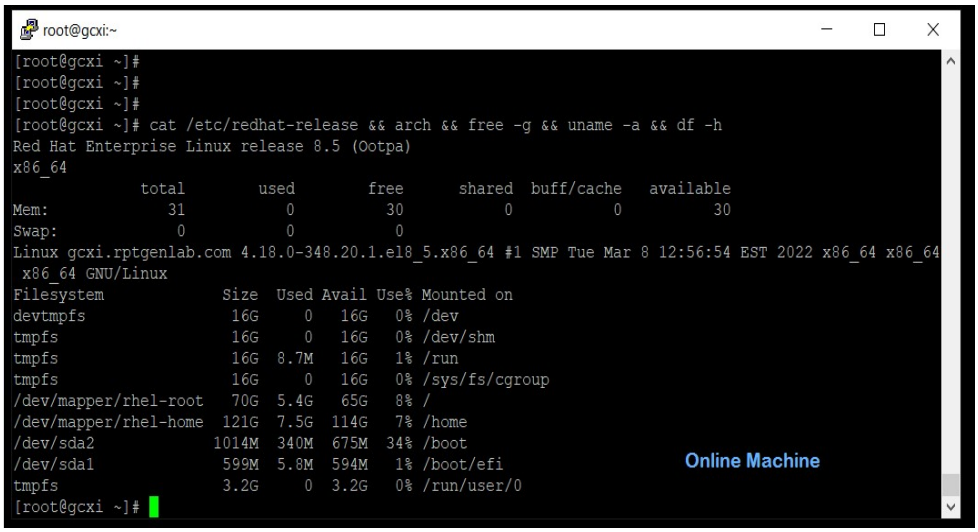
Refer docker documentation for latest updates.

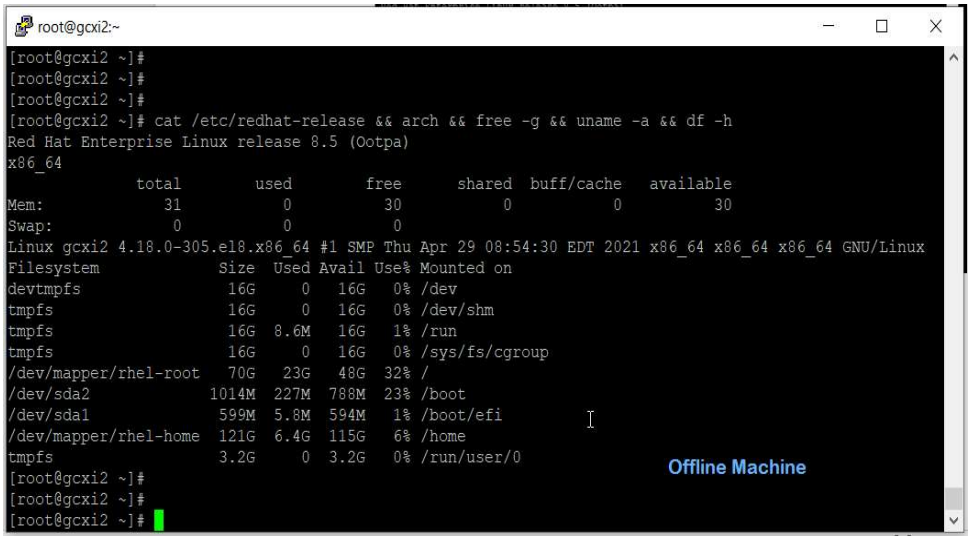
## Installing Docker Offline Scenario

Following instructions will help you to install the docker components where the system is not able/restricted to access internet.

## In addition, you require:

- **Online machine** - A machine with online access, which must have:
  - Sufficient space to download the YUM packages.
  - The same operating system version as the offline machines where Genesys CX Insights will be installed. Packages downloaded on a different OS version may not work, because the dependencies that determine which files to download vary depending on the operating system version, components and packages that already exist on the machine. Failure to follow this recommendation may cause installation on the target machine to fail.






## Perform below steps in online machine:

1. Execute the following command to configure YUM so it can download packages correctly:

$ yum install -y yum-utils

```
[root@gcxi ~]# yum install -y yum-utils
Updating Subscription Management repositories.
Last metadata expiration check: 0:01:18 ago on Thursday 14 April 2022 01:44:03 PM IST.
Dependencies resolved.
================================================================================
 Package              Architecture          Version                Repository
================================================================================
Installing:
 yum-utils            noarch                4.0.21-4.el8_5         rhel-8-for-x86_64-baseos-rpms

Transaction Summary
================================================================================
Install  1 Package

Total download size: 73 k
Installed size: 23 k
Downloading Packages:
yum-utils-4.0.21-4.el8_5.noarch.rpm                                        156
--------------------------------------------------------------------------------
Total                                                                      156
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :
  Installing       : yum-utils-4.0.21-4.el8_5.noarch
  Running scriptlet: yum-utils-4.0.21-4.el8_5.noarch
  Verifying        : yum-utils-4.0.21-4.el8_5.noarch
Installed products updated.

Installed:
  yum-utils-4.0.21-4.el8_5.noarch

Complete!
[root@gcxi ~]#
```

2. Execute the following commands to install the Docker repository:

   $ yum-config-manager --add-repo \
   https://download.docker.com/linux/centos/docker-ce.repo

```
[root@gcxi ~]#
[root@gcxi ~]# yum-config-manager --add-repo \
> https://download.docker.com/linux/centos/docker-ce.repo
Updating Subscription Management repositories.
Adding repo from: https://download.docker.com/linux/centos/docker-ce.repo
[root@gcxi ~]#
[root@gcxi ~]#
```

3. Execute the following commands to download required files:

   $ yumdownloader --assumeyes --destdir=<your_rpm_dir>/yum --resolve yum-utils
   $ yumdownloader --assumeyes --destdir=<your_rpm_dir>/dm --resolve device-mapper-persistent-data
   $ yumdownloader --assumeyes --destdir=<your_rpm_dir>/lvm2 --resolve lvm2
   $ yumdownloader --assumeyes --destdir=<your_rpm_dir>/docker-ce --resolve docker-ce
   $ yumdownloader --assumeyes --destdir=<your_rpm_dir>/se --resolve container-selinux

   where <your_rpm_dir> is a directory on your online machine.

```
root@gcxi:~
[root@gcxi ~]#
[root@gcxi ~]# ^C
[root@gcxi ~]# ^C
[root@gcxi ~]#
[root@gcxi ~]#
[root@gcxi ~]# yumdownloader --assumeyes --destdir=/docker_rpm/yum --resolve yum-utils 1
Updating Subscription Management repositories.
Docker CE Stable - x86_64                                                    14 kB/s | 3.5 kB     00:00
yum-utils-4.0.21-4.el8_5.noarch.rpm                                          74 kB/s |  73 kB     00:00
[root@gcxi ~]# yumdownloader --assumeyes --destdir=/docker_rpm/dm --resolve device-mapper-persistent-data 2
Updating Subscription Management repositories.
Last metadata expiration check: 0:00:09 ago on Wednesday 13 April 2022 01:44:33 PM IST.
device-mapper-persistent-data-0.9.0-4.el8.x86_64.rpm                        283 kB/s | 925 kB     00:03
[root@gcxi ~]# yumdownloader --assumeyes --destdir=/docker_rpm/lvm2 --resolve lvm2 3
Updating Subscription Management repositories.
Last metadata expiration check: 0:00:19 ago on Wednesday 13 April 2022 01:44:33 PM IST.
lvm2-2.03.12-11.el8_5.x86_64.rpm                                            458 kB/s | 1.6 MB     00:03
[root@gcxi ~]# yumdownloader --assumeyes --destdir=/docker_rpm/docker-ce --resolve docker-ce 4
Updating Subscription Management repositories.
Last metadata expiration check: 0:00:30 ago on Wednesday 13 April 2022 01:44:33 PM IST.
(1/18): libcgroup-0.41-19.el8.x86_64.rpm                                    101 kB/s |  70 kB     00:00
(2/18): docker-ce-20.10.14-3.el8.x86_64.rpm                                  16 MB/s |  22 MB     00:01
(3/18): docker-ce-rootless-extras-20.10.14-3.el8.x86_64.rpm                 6.0 MB/s | 4.6 MB     00:00
(4/18): container-selinux-2.173.0-1.module+el8.5.0+13852+150547f7.noarch.rpm 142 kB/s |  57 kB     00:00
(5/18): fuse-overlayfs-1.8-1.module+el8.5.0+13754+92ec836b.x86_64.rpm        117 kB/s |  73 kB     00:00
(6/18): fuse3-3.2.1-12.el8.x86_64.rpm                                        132 kB/s |  50 kB     00:00
(7/18): python3-setools-4.3.0-2.el8.x86_64.rpm                               984 kB/s | 626 kB     00:00
(8/18): checkpolicy-2.9-1.el8.x86_64.rpm                                     515 kB/s | 346 kB     00:00
(9/18): libslirp-4.4.0-1.module+el8.5.0+12582+56d94c81.x86_64.rpm            156 kB/s |  70 kB     00:00
(10/18): docker-scan-plugin-0.17.0-3.el8.x86_64.rpm                           23 MB/s | 3.8 MB     00:00
(11/18): python3-audit-3.0-0.17.20191104git1c2f876.el8.x86_64.rpm            224 kB/s |  86 kB     00:00
(12/18): fuse3-libs-3.2.1-12.el8.x86_64.rpm                                   92 kB/s |  94 kB     00:01
(13/18): python3-libsemanage-2.9-6.el8.x86_64.rpm                            288 kB/s | 127 kB     00:00
(14/18): policycoreutils-python-utils-2.9-16.el8.noarch.rpm                  376 kB/s | 252 kB     00:00
(15/18): slirp4netns-1.1.8-1.module+el8.5.0+12582+56d94c81.x86_64.rpm        117 kB/s |  51 kB     00:00
(16/18): docker-ce-cli-20.10.14-3.el8.x86_64.rpm                              20 MB/s |  30 MB     00:01
(17/18): containerd.io-1.5.11-3.1.el8.x86_64.rpm                              16 MB/s |  29 MB     00:01
(18/18): python3-policycoreutils-2.9-16.el8.noarch.rpm                       231 kB/s | 2.2 MB     00:09
[root@gcxi ~]# yumdownloader --assumeyes --destdir=/docker_rpm/se --resolve container-selinux 5
Updating Subscription Management repositories.
Last metadata expiration check: 0:00:47 ago on Wednesday 13 April 2022 01:44:33 PM IST.
(1/7): container-selinux-2.173.0-1.module+el8.5.0+13852+150547f7.noarch.rpm   94 kB/s |  57 kB     00:00
(2/7): checkpolicy-2.9-1.el8.x86_64.rpm                                      319 kB/s | 346 kB     00:01
(3/7): policycoreutils-python-utils-2.9-16.el8.noarch.rpm                    471 kB/s | 252 kB     00:00
(4/7): python3-policycoreutils-2.9-16.el8.noarch.rpm                         1.4 MB/s | 2.2 MB     00:01
(5/7): python3-audit-3.0-0.17.20191104git1c2f876.el8.x86_64.rpm              102 kB/s |  86 kB     00:00
(6/7): python3-setools-4.3.0-2.el8.x86_64.rpm                                1.6 MB/s | 626 kB     00:00
(7/7): python3-libsemanage-2.9-6.el8.x86_64.rpm                              141 kB/s | 127 kB     00:00
[root@gcxi ~]#
[root@gcxi ~]#
```

4. Download PostgreSQL Image

   Execute the following command pull the image and save it as a TAR archive:

   $ docker pull postgres:<version>

   $ docker save postgres:<version> > postgres.tar.gz

   *where <version> is the PostgreSQL release of the image to pull.*

```
[root@gcxi ~]# docker pull postgres:12 (1)
12: Pulling from library/postgres
1fe172e4850f: Pull complete
c2bb685f623f: Pull complete
3027ff705410: Pull complete
062371e3461d: Pull complete
39d54e944de7: Pull complete
6530357dda9a: Pull complete
b1d302dc78c6: Pull complete
f6d91cb1d3c1: Pull complete
85c75cecd287: Pull complete
ce327230a313: Pull complete
55277580dd99: Pull complete
2f48dddf94f1: Pull complete
1f949a245982: Pull complete
Digest: sha256:886b50cdce26f025a02b2107521462341ffb74f16e7766837545ee61a7aa295c
Status: Downloaded newer image for postgres:12
docker.io/library/postgres:12
[root@gcxi ~]# docker save postgres:12 | gzip > /docker_rpm/postgres.tar.gz (2)

[root@gcxi ~]#
```

- **File transfer** — Copy the files from online machine to offline using file transfer tools.

  *We used SCP utility to transfer the files.*

  $ scp /docker_rpm/* root@10.31.193.74:/docker_rpm/

```
[root@gcxi ~]# cd /docker_rpm/
[root@gcxi docker_rpm]# ls -lrth (1)  Verify the contetnts in online machine
total 127M
drwxr-xr-x. 2 root root   49 Apr 13 13:44 yum
drwxr-xr-x. 2 root root   66 Apr 13 13:44 dm
drwxr-xr-x. 2 root root   46 Apr 13 13:44 lvm2
drwxr-xr-x. 2 root root 4.0K Apr 13 13:45 docker-ce
drwxr-xr-x. 2 root root 4.0K Apr 13 13:45 se
-rwxr-xr-x. 1 root root 127M May 11 15:30 postgres.tar.gz
[root@gcxi docker_rpm]#
[root@gcxi docker_rpm]# scp -r /docker_rpm/* root@10.31.193.74:/docker_rpm/ (2)  Copy the contents from online machine to offline machine
root@10.31.193.74's password:
/root/.bashrc: line 13: kubectl: command not found
device-mapper-persistent-data-0.9.0-4.el8.x86_64.rpm                                  100%  925KB  86.2MB/s   00:00
python3-policycoreutils-2.9-16.el8.noarch.rpm                                         100% 2297KB 162.3MB/s   00:00
libcgroup-0.41-19.el8.x86_64.rpm                                                      100%   70KB  33.2MB/s   00:00
docker-ce-20.10.14-3.el8.x86_64.rpm                                                   100%   22MB 324.5MB/s   00:00
docker-ce-rootless-extras-20.10.14-3.el8.x86_64.rpm                                   100% 4741KB 235.4MB/s   00:00
container-selinux-2.173.0-1.module+el8.5.0+13852+150547f7.noarch.rpm                  100%   57KB  28.7MB/s   00:00
fuse-overlayfs-1.8-1.module+el8.5.0+13754+92ec836b.x86_64.rpm                         100%   73KB  32.7MB/s   00:00
fuse3-3.2.1-12.el8.x86_64.rpm                                                         100%   50KB  15.2MB/s   00:00
checkpolicy-2.9-1.el8.x86_64.rpm                                                      100%  346KB  81.0MB/s   00:00
python3-setools-4.3.0-2.el8.x86_64.rpm                                                100%  626KB 106.1MB/s   00:00
libslirp-4.4.0-1.module+el8.5.0+12582+56d94c81.x86_64.rpm                             100%   70KB  27.9MB/s   00:00
fuse3-libs-3.2.1-12.el8.x86_64.rpm                                                    100%   94KB  30.3MB/s   00:00
docker-scan-plugin-0.17.0-3.el8.x86_64.rpm                                            100% 3841KB 152.5MB/s   00:00
python3-audit-3.0-0.17.20191104git1c2f876.el8.x86_64.rpm                              100%   86KB  36.5MB/s   00:00
python3-libsemanage-2.9-6.el8.x86_64.rpm                                              100%  127KB  45.0MB/s   00:00
policycoreutils-python-utils-2.9-16.el8.noarch.rpm                                    100%  252KB  75.8MB/s   00:00
slirp4netns-1.1.8-1.module+el8.5.0+12582+56d94c81.x86_64.rpm                          100%   51KB  28.0MB/s   00:00
docker-ce-cli-20.10.14-3.el8.x86_64.rpm                                               100%   30MB 162.2MB/s   00:00
containerd.io-1.5.11-3.1.el8.x86_64.rpm                                               100%   29MB 195.9MB/s   00:00
lvm2-2.03.12-11.el8_5.x86_64.rpm                                                      100% 1673KB 214.0MB/s   00:00
postgres.tar.gz                                                                       100%  126MB 255.5MB/s   00:00
python3-policycoreutils-2.9-16.el8.noarch.rpm                                         100% 2297KB 164.6MB/s   00:00
checkpolicy-2.9-1.el8.x86_64.rpm                                                      100%  346KB  73.2MB/s   00:00
container-selinux-2.173.0-1.module+el8.5.0+13852+150547f7.noarch.rpm                  100%   57KB  22.7MB/s   00:00
policycoreutils-python-utils-2.9-16.el8.noarch.rpm                                    100%  252KB  72.0MB/s   00:00
python3-audit-3.0-0.17.20191104git1c2f876.el8.x86_64.rpm                              100%   86KB  32.5MB/s   00:00
python3-libsemanage-2.9-6.el8.x86_64.rpm                                              100%  127KB  44.2MB/s   00:00
python3-setools-4.3.0-2.el8.x86_64.rpm                                                100%  626KB 124.2MB/s   00:00
yum-utils-4.0.21-4.el8_5.noarch.rpm                                                   100%   73KB  26.3MB/s   00:00
[root@gcxi docker_rpm]#
```

## Install Docker (offline machine):

1. Verify the copied contents in offline machine

   $ ls -lrth /docker_rpm/

```
[root@gcxi2 ~]# hostname
gcxi2
[root@gcxi2 ~]# ls -lrth /docker_rpm/
total 127M
drwxr-xr-x. 2 root root   66 May 11 15:41 dm
drwxr-xr-x. 2 root root 4.0K May 11 15:41 docker-ce
drwxr-xr-x. 2 root root   46 May 11 15:41 lvm2
-rwxr-xr-x. 1 root root 127M May 11 15:41 postgres.tar.gz
drwxr-xr-x. 2 root root 4.0K May 11 15:41 se
drwxr-xr-x. 2 root root   49 May 11 15:41 yum
[root@gcxi2 ~]#
```

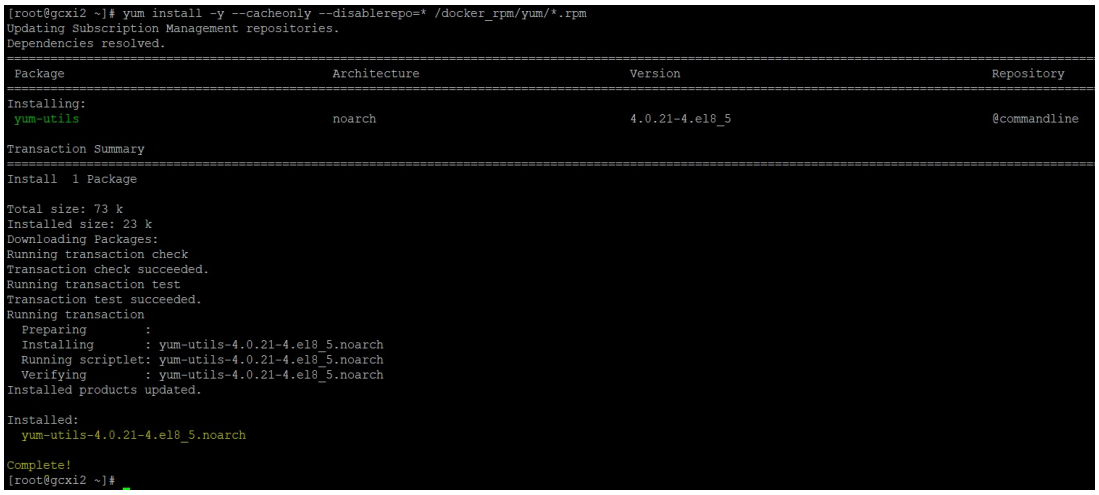
2. Execute the following commands to uninstall any old docker software:

   $ yum remove docker \
   docker-client \
   docker-client-latest \
   docker-common \
   docker-latest \
   docker-latest-logrotate \
   docker-logrotate \
   docker-selinux \
   docker-engine-selinux \
   docker-engine

```
[root@gcxi2 ~]# yum remove -y docker \
>     docker-client \
>     docker-client-latest \
>     docker-common \
>     docker-latest \
>     docker-latest-logrotate \
>     docker-logrotate \
>     docker-selinux \
>     docker-engine-selinux \
>     docker-engine
Updating Subscription Management repositories.
No match for argument: docker
No match for argument: docker-client
No match for argument: docker-client-latest
No match for argument: docker-common
No match for argument: docker-latest
No match for argument: docker-latest-logrotate
No match for argument: docker-logrotate
No match for argument: docker-selinux
No match for argument: docker-engine-selinux
No match for argument: docker-engine
No packages marked for removal.
Dependencies resolved.
Nothing to do.
Complete!
[root@gcxi2 ~]#
```

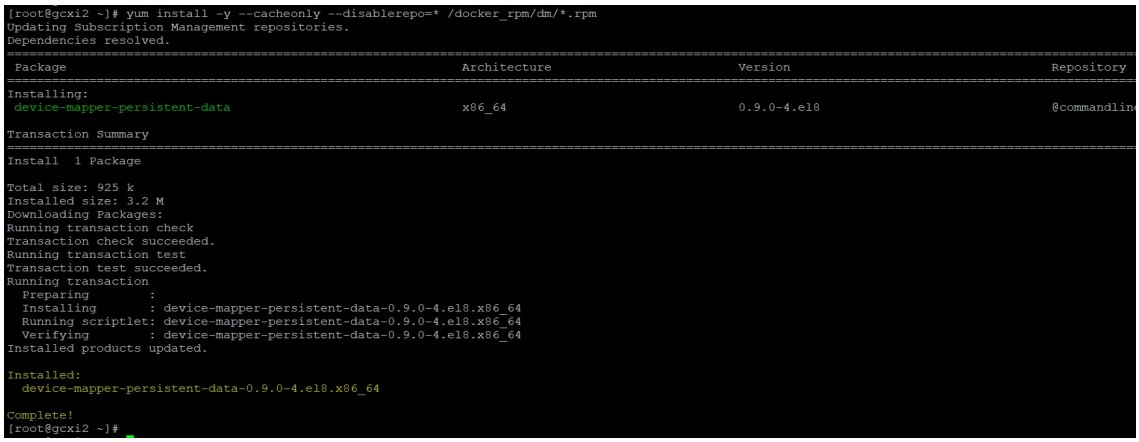
3. Execute the following command to install yum utilities:

$ yum install -y --cacheonly --disablerepo=* <your_rpm_dir>/yum/*.rpm

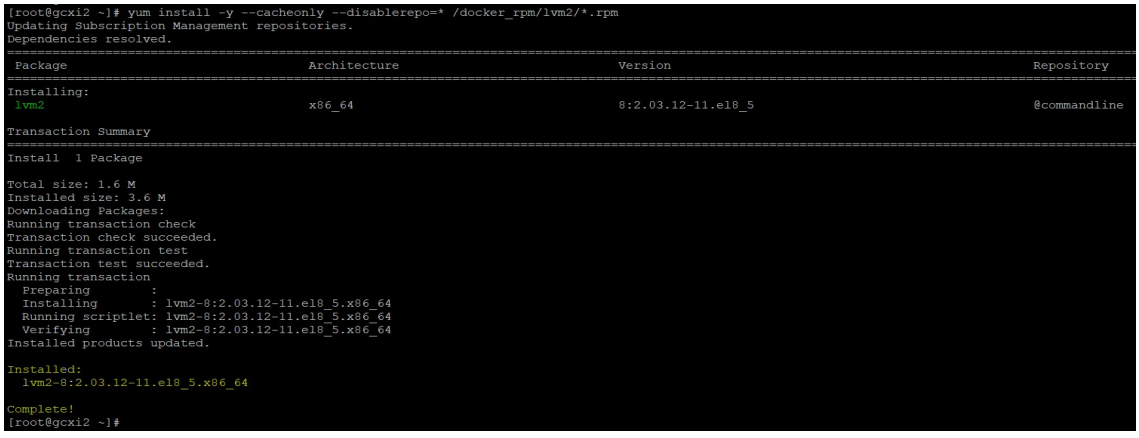



4. Execute the following commands to install Docker file drivers:

$ yum install -y --cacheonly --disablerepo=* <your_rpm_dir>/dm/*.rpm




$ yum install -y --cacheonly --disablerepo=* <your_rpm_dir>/lvm2/*.rpm



5. Execute the following command to install container-selinux:

   $ yum install -y --cacheonly --disablerepo=* <your_rpm_dir>/se/*.rpm

```
[root@gcxi2 ~]# yum install -y --cacheonly --disablerepo=* /docker_rpm/se/*.rpm
Updating Subscription Management repositories.
Dependencies resolved.
================================================================================
 Package                 Architecture  Version                                  Repository      Size
================================================================================
Installing:
 checkpolicy             x86_64        2.9-1.el8                                @commandline   346 k
 container-selinux       noarch        2:2.173.0-1.module+el8.5.0+13852+150547f7 @commandline   57 k
 policycoreutils-python-utils noarch   2.9-16.el8                               @commandline   252 k
 python3-audit           x86_64        3.0-0.17.20191104git1c2f876.el8          @commandline    86 k
 python3-libsemanage     x86_64        2.9-6.el8                                @commandline   127 k
 python3-policycoreutils noarch        2.9-16.el8                               @commandline   2.2 M
 python3-setools         x86_64        4.3.0-2.el8                              @commandline   626 k

Transaction Summary
================================================================================
Install  7 Packages

Total size: 3.7 M
Installed size: 10 M
Downloading Packages:
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Installing       : python3-setools-4.3.0-2.el8.x86_64                     1/7
  Installing       : python3-libsemanage-2.9-6.el8.x86_64                   2/7
  Installing       : python3-audit-3.0-0.17.20191104git1c2f876.el8.x86_64   3/7
  Installing       : checkpolicy-2.9-1.el8.x86_64                           4/7
  Installing       : python3-policycoreutils-2.9-16.el8.noarch              5/7
  Installing       : policycoreutils-python-utils-2.9-16.el8.noarch         6/7
  Running scriptlet: container-selinux-2:2.173.0-1.module+el8.5.0+13852+150547f7.noarch 7/7
  Installing       : container-selinux-2:2.173.0-1.module+el8.5.0+13852+150547f7.noarch 7/7
  Running scriptlet: container-selinux-2:2.173.0-1.module+el8.5.0+13852+150547f7.noarch 7/7
```

```
  Verifying        : checkpolicy-2.9-1.el8.x86_64                           1/7
  Verifying        : container-selinux-2:2.173.0-1.module+el8.5.0+13852+150547f7.noarch 2/7
  Verifying        : policycoreutils-python-utils-2.9-16.el8.noarch         3/7
  Verifying        : python3-audit-3.0-0.17.20191104git1c2f876.el8.x86_64   4/7
  Verifying        : python3-libsemanage-2.9-6.el8.x86_64                   5/7
  Verifying        : python3-policycoreutils-2.9-16.el8.noarch              6/7
  Verifying        : python3-setools-4.3.0-2.el8.x86_64                     7/7
Installed products updated.

Installed:
  checkpolicy-2.9-1.el8.x86_64                container-selinux-2:2.173.0-1.module+el8.5.0+13852+150547f7.noarch  policycoreutils-python-utils-2.9-16.el8.noarch
  python3-audit-3.0-0.17.20191104git1c2f876.el8.x86_64  python3-libsemanage-2.9-6.el8.x86_64  python3-policycoreutils-2.9-16.el8.noarch
  python3-setools-4.3.0-2.el8.x86_64

Complete!
[root@gcxi2 ~]#
```

6. Execute the following command to install Docker:

   $ yum install -y --cacheonly --disablerepo=* <your_rpm_dir>/docker-ce/*.rpm

```
[root@gcxi2 ~]# yum install -y --cacheonly --disablerepo=* /docker_rpm/docker-ce/*.rpm
Updating Subscription Management repositories.
Package checkpolicy-2.9-1.el8.x86_64 is already installed.
Package container-selinux-2:2.173.0-1.module+el8.5.0+13852+150547f7.noarch is already installed.
Package policycoreutils-python-utils-2.9-16.el8.noarch is already installed.
Package python3-audit-3.0-0.17.20191104git1c2f876.el8.x86_64 is already installed.
Package python3-libsemanage-2.9-6.el8.x86_64 is already installed.
Package python3-policycoreutils-2.9-16.el8.noarch is already installed.
Package python3-setools-4.3.0-2.el8.x86_64 is already installed.
Dependencies resolved.
================================================================================================================================================================
 Package                                    Architecture                Version                                                  Repository                  Size
================================================================================================================================================================
Installing:
 containerd.io                              x86_64                      1.5.11-3.1.el8                                           @commandline                29 M
 docker-ce                                  x86_64                      3:20.10.14-3.el8                                         @commandline                22 M
 docker-ce-cli                              x86_64                      1:20.10.14-3.el8                                         @commandline                30 M
 docker-ce-rootless-extras                  x86_64                      20.10.14-3.el8                                           @commandline               4.6 M
 docker-scan-plugin                         x86_64                      0.17.0-3.el8                                             @commandline               3.8 M
 fuse-overlayfs                             x86_64                      1.8-1.module+el8.5.0+13754+92ec836b                      @commandline                73 k
 fuse3                                      x86_64                      3.2.1-12.el8                                             @commandline                50 k
 fuse3-libs                                 x86_64                      3.2.1-12.el8                                             @commandline                94 k
 libcgroup                                  x86_64                      0.41-19.el8                                              @commandline                70 k
 libslirp                                   x86_64                      4.4.0-1.module+el8.5.0+12582+56d94c81                    @commandline                70 k
 slirp4netns                                x86_64                      1.1.8-1.module+el8.5.0+12582+56d94c81                    @commandline                51 k

Transaction Summary
================================================================================================================================================================
Install  11 Packages

Total size: 90 M
Installed size: 374 M
Downloading Packages:
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                                                                                                       1/1
  Installing       : docker-scan-plugin-0.17.0-3.el8.x86_64                                                                                                1/11
  Running scriptlet: docker-scan-plugin-0.17.0-3.el8.x86_64                                                                                                1/11
  Installing       : docker-ce-cli-1:20.10.14-3.el8.x86_64                                                                                                 2/11
  Running scriptlet: docker-ce-cli-1:20.10.14-3.el8.x86_64                                                                                                 2/11
  Installing       : libslirp-4.4.0-1.module+el8.5.0+12582+56d94c81.x86_64                                                                                 3/11
  Installing       : slirp4netns-1.1.8-1.module+el8.5.0+12582+56d94c81.x86_64                                                                              4/11
  Running scriptlet: libcgroup-0.41-19.el8.x86_64                                                                                                          5/11
  Installing       : libcgroup-0.41-19.el8.x86_64                                                                                                          5/11
  Running scriptlet: libcgroup-0.41-19.el8.x86_64                                                                                                          5/11
  Installing       : fuse3-libs-3.2.1-12.el8.x86_64                                                                                                        6/11
  Running scriptlet: fuse3-libs-3.2.1-12.el8.x86_64                                                                                                        6/11
  Installing       : fuse3-3.2.1-12.el8.x86_64                                                                                                             7/11
  Installing       : fuse-overlayfs-1.8-1.module+el8.5.0+13754+92ec836b.x86_64                                                                             8/11
  Running scriptlet: fuse-overlayfs-1.8-1.module+el8.5.0+13754+92ec836b.x86_64                                                                             8/11
  Installing       : containerd.io-1.5.11-3.1.el8.x86_64                                                                                                   9/11
  Running scriptlet: containerd.io-1.5.11-3.1.el8.x86_64                                                                                                   9/11
  Installing       : docker-ce-rootless-extras-20.10.14-3.el8.x86_64                                                                                      10/11
  Running scriptlet: docker-ce-rootless-extras-20.10.14-3.el8.x86_64                                                                                      10/11
  Installing       : docker-ce-3:20.10.14-3.el8.x86_64                                                                                                    11/11
  Running scriptlet: docker-ce-3:20.10.14-3.el8.x86_64                                                                                                    11/11
  Verifying        : containerd.io-1.5.11-3.1.el8.x86_64                                                                                                   1/11
  Verifying        : docker-ce-3:20.10.14-3.el8.x86_64                                                                                                     2/11
  Verifying        : docker-ce-cli-1:20.10.14-3.el8.x86_64                                                                                                 3/11
  Verifying        : docker-ce-rootless-extras-20.10.14-3.el8.x86_64                                                                                       4/11
  Verifying        : docker-scan-plugin-0.17.0-3.el8.x86_64                                                                                                5/11
  Verifying        : fuse3-3.2.1-12.el8.x86_64                                                                                                             6/11
  Verifying        : fuse3-libs-3.2.1-12.el8.x86_64                                                                                                        7/11
  Verifying        : fuse-overlayfs-1.8-1.module+el8.5.0+13754+92ec836b.x86_64                                                                             8/11
  Verifying        : libcgroup-0.41-19.el8.x86_64                                                                                                          9/11
  Verifying        : libslirp-4.4.0-1.module+el8.5.0+12582+56d94c81.x86_64                                                                                10/11
  Verifying        : slirp4netns-1.1.8-1.module+el8.5.0+12582+56d94c81.x86_64                                                                             11/11
Installed products updated.

Installed:
  containerd.io-1.5.11-3.1.el8.x86_64                  docker-ce-3:20.10.14-3.el8.x86_64                    docker-ce-cli-1:20.10.14-3.el8.x86_64
  docker-ce-rootless-extras-20.10.14-3.el8.x86_64      docker-scan-plugin-0.17.0-3.el8.x86_64               fuse-overlayfs-1.8-1.module+el8.5.0+13754+92ec836b.x86_64
  fuse3-3.2.1-12.el8.x86_64                            fuse3-libs-3.2.1-12.el8.x86_64                       libcgroup-0.41-19.el8.x86_64
  libslirp-4.4.0-1.module+el8.5.0+12582+56d94c81.x86_64 slirp4netns-1.1.8-1.module+el8.5.0+12582+56d94c81.x86_64

Complete!
[root@gcxi2 ~]#
[root@gcxi2 ~]#
```

7. Execute the following commands to enable and start docker service:

   $ systemctl enable docker

   $ systemctl start docker

   $ systemctl status docker

```
[root@gcxi2 ~]# systemctl enable docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/do
[root@gcxi2 ~]# systemctl start docker
[root@gcxi2 ~]# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2022-04-13 16:26:12 IST; 7s ago
     Docs: https://docs.docker.com
 Main PID: 6826 (dockerd)
    Tasks: 13
   Memory: 36.5M
   CGroup: /system.slice/docker.service
           └─6826 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Apr 13 16:26:11 gcxi2 dockerd[6826]: time="2022-04-13T16:26:11.848491942+05:30" level=error msg="Failed
Apr 13 16:26:11 gcxi2 dockerd[6826]: time="2022-04-13T16:26:11.876343760+05:30" level=warning msg="Your
Apr 13 16:26:11 gcxi2 dockerd[6826]: time="2022-04-13T16:26:11.876432607+05:30" level=warning msg="Your
Apr 13 16:26:11 gcxi2 dockerd[6826]: time="2022-04-13T16:26:11.876809064+05:30" level=info msg="Loading
Apr 13 16:26:12 gcxi2 dockerd[6826]: time="2022-04-13T16:26:12.410434800+05:30" level=info msg="Default
Apr 13 16:26:12 gcxi2 dockerd[6826]: time="2022-04-13T16:26:12.561409725+05:30" level=info msg="Loading
Apr 13 16:26:12 gcxi2 dockerd[6826]: time="2022-04-13T16:26:12.589331928+05:30" level=info msg="Docker
Apr 13 16:26:12 gcxi2 dockerd[6826]: time="2022-04-13T16:26:12.589521941+05:30" level=info msg="Daemon
Apr 13 16:26:12 gcxi2 systemd[1]: Started Docker Application Container Engine.
Apr 13 16:26:12 gcxi2 dockerd[6826]: time="2022-04-13T16:26:12.636964277+05:30" level=info msg="API lis
[root@gcxi2 ~]#
[root@gcxi2 ~]#
```

8. Check docker version:

   $ docker version

```
[root@gcxi2 ~]# docker version
Client: Docker Engine - Community
 Version:           20.10.14
 API version:       1.41
 Go version:        go1.16.15
 Git commit:        a224086
 Built:             Thu Mar 24 01:47:44 2022
 OS/Arch:           linux/amd64
 Context:           default
 Experimental:      true

Server: Docker Engine - Community
 Engine:
  Version:          20.10.14
  API version:      1.41 (minimum version 1.12)
  Go version:       go1.16.15
  Git commit:       87a90dc
  Built:            Thu Mar 24 01:46:10 2022
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.5.11
  GitCommit:        3df54a852345ae127d1fa3092b95168e4a88e2f8
 runc:
  Version:          1.0.3
  GitCommit:        v1.0.3-0-gf46b6ba
 docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0
[root@gcxi2 ~]#
[root@gcxi2 ~]#
```

9. Load PostgreSQL Image:

   $ docker load < postgres.tar.gz

```
[root@gcxi2 ~]# docker load < /docker_rpm/postgres.tar.gz
9c1b6dd6c1e6: Loading layer [==================================================>]   83.9MB/83.9MB
f121e8841357: Loading layer [==================================================>]  10.18MB/10.18MB
8f6516bbd7c3: Loading layer [==================================================>]    340kB/340kB
289452c265f5: Loading layer [==================================================>]  4.192MB/4.192MB
9829cf46cc7b: Loading layer [==================================================>]   25.7MB/25.7MB
eb4827143dd5: Loading layer [==================================================>]  3.554MB/3.554MB
54ccbeacfc8c: Loading layer [==================================================>]  2.048kB/2.048kB
12e6d5c212c5: Loading layer [==================================================>]  8.704kB/8.704kB
7f8a486203c0: Loading layer [==================================================>]  252.6MB/252.6MB
f5128df9c90c: Loading layer [==================================================>]  62.46kB/62.46kB
6e4364b15a4b: Loading layer [==================================================>]  2.048kB/2.048kB
e5f201922232: Loading layer [==================================================>]  3.584kB/3.584kB
b3d91dbf04e6: Loading layer [==================================================>]  15.36kB/15.36kB
Loaded image: postgres:12
[root@gcxi2 ~]#
```

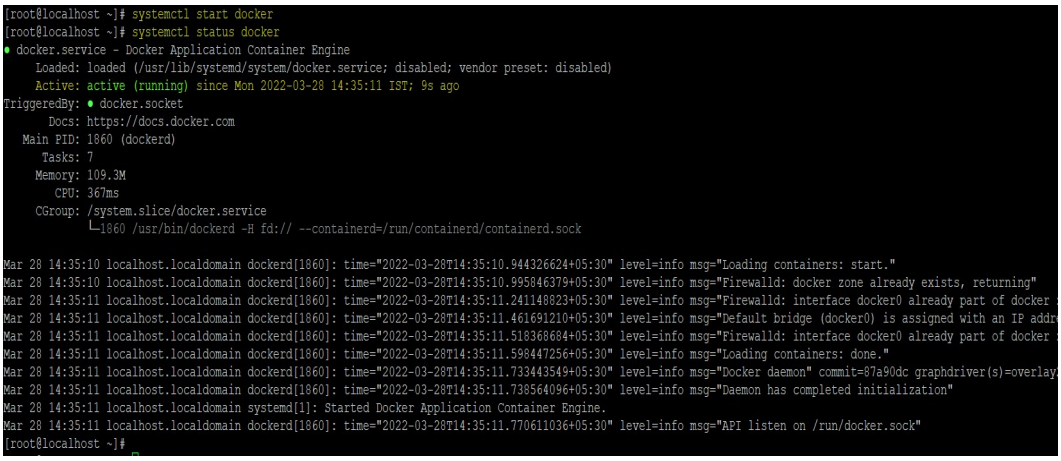
## Loading Docker images

Required files already in place (Linux Machine). If not please refer, how to download and prepare the package for docker deployment.
Following steps will help you to load the GCXI .tar images into Docker.

1. check and start Docker (as a root) if it's not running already:

   $ systemctl start docker

   $ systemctl status docker




   Note :- Proceeding commands you have to execute as a user which is added to the 'docker' group.

   $ su – gcxi

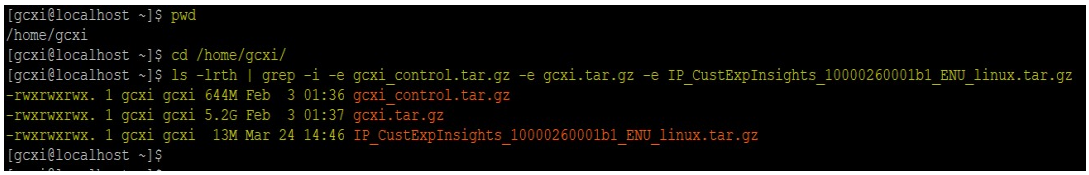
   $ id


2. Change the current directory where you copied GCXI .tar images and verify its permissions:

   $ pwd

   $ cd /home/gcxi

   $ ls -lrth | grep -i -e gcxi_control.tar.gz -e gcxi.tar.gz -e IP_CustExpInsights_10000260001b1_ENU_linux.tar.gz




   If required change the permissions then proceed further.
   Note:- Prior to loading images, please make sure **40 GB free space needed for "/var/lib/docker" to accommodate GCXI images.**

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

$ docker load -i gcxi_control.tar.gz

```
[gcxi@localhost ~]$ docker load -i gcxi_control.tar.gz
352ba846236b: Loading layer [==================================================>]  234.7MB/234.7MB
3ba8c926eef9: Loading layer [==================================================>]  20.48kB/20.48kB
425e0ee24ad9: Loading layer [==================================================>]  173.9MB/173.9MB
29f1eebe2b74: Loading layer [==================================================>]    242MB/242MB
68f431db43cb: Loading layer [==================================================>]  293.9MB/293.9MB
1511cf30ff30: Loading layer [==================================================>]  70.15MB/70.15MB
a2e571c3dce7: Loading layer [==================================================>]  30.83MB/30.83MB
b1f847bd1e74: Loading layer [==================================================>]  4.096kB/4.096kB
1e47535fbe8c: Loading layer [==================================================>]    159MB/159MB
10dfb07d63be: Loading layer [==================================================>]  3.584kB/3.584kB
d6e4d24bbb53: Loading layer [==================================================>]  9.029MB/9.029MB
f020b24153fd: Loading layer [==================================================>]  3.072kB/3.072kB
1ae3d9204e81: Loading layer [==================================================>]  3.584kB/3.584kB
dcd21507efd8: Loading layer [==================================================>]   35.3MB/35.3MB
c70da534398c: Loading layer [==================================================>]  667.1kB/667.1kB
64540cf9e46a: Loading layer [==================================================>]  4.096kB/4.096kB
be4aead4ef84: Loading layer [==================================================>]  157.7kB/157.7kB
6ffa4cdacf9d: Loading layer [==================================================>]  218.1kB/218.1kB
dde33ac29c07: Loading layer [==================================================>]    144MB/144MB
689f7f31c4f0: Loading layer [==================================================>]    144MB/144MB
Loaded image: pureengage-docker-production.jfrog.io/gcxi/gcxi_control:100.0.026.0001
[gcxi@localhost ~]$
```

$ docker load -i gcxi.tar.gz

```
[gcxi@localhost ~]$
[gcxi@localhost ~]$ docker load -i gcxi.tar.gz
a54caf4d666a: Loading layer [==================================================>]  127.2MB/127.2MB
997776166c21: Loading layer [==================================================>]    171MB/171MB
0f286f34877f: Loading layer [==================================================>]  4.096kB/4.096kB
baaac0977c1a: Loading layer [==================================================>]    159MB/159MB
b7e7625bf892: Loading layer [==================================================>]  2.048kB/2.048kB
df95a229d38f: Loading layer [==================================================>]  13.82kB/13.82kB
710e891a1c01: Loading layer [==================================================>]  4.608kB/4.608kB
61e57cf993be: Loading layer [==================================================>]  9.201GB/9.201GB
aca939ff47c7: Loading layer [==================================================>]  8.138MB/8.138MB
747db58af066: Loading layer [==================================================>]  95.11MB/95.11MB
9acbc57a9b66: Loading layer [==================================================>]  13.57MB/13.57MB
cb5b97b1731f: Loading layer [==================================================>]  5.632kB/5.632kB
978d82ead718: Loading layer [==================================================>]  6.656kB/6.656kB
d3d2b0255a2c: Loading layer [==================================================>]  2.057GB/2.057GB
970b55c3a75f: Loading layer [==================================================>]  10.83MB/10.83MB
122ff794b9a9: Loading layer [==================================================>]  88.09MB/88.09MB
3432ba131bb3: Loading layer [==================================================>]  88.36MB/88.36MB
4bab2fe8ea52: Loading layer [==================================================>]  1.409MB/1.409MB
27488beff492: Loading layer [==================================================>]  1.409MB/1.409MB
5561ef16970e: Loading layer [==================================================>]  759.3kB/759.3kB
d6d1a171d48c: Loading layer [==================================================>]  37.89kB/37.89kB
fb29051d9485: Loading layer [==================================================>]  4.096kB/4.096kB
5ba7fa772d45: Loading layer [==================================================>]   35.3MB/35.3MB
3e4f0c8f2ccb: Loading layer [==================================================>]  48.64kB/48.64kB
e0778fa0d6ec: Loading layer [==================================================>]  158.7kB/158.7kB
a7283eec1515: Loading layer [==================================================>]  32.46MB/32.46MB
2d0e6405f09b: Loading layer [==================================================>]  4.571MB/4.571MB
ebc5b409ac92: Loading layer [==================================================>]  487.4kB/487.4kB
722ca8d4386c: Loading layer [==================================================>]   7.68kB/7.68kB
04a6503737c2: Loading layer [==================================================>]  57.34kB/57.34kB
3d502634e2a8: Loading layer [==================================================>]  13.35MB/13.35MB
c6a82ab05adf: Loading layer [==================================================>]  195.1kB/195.1kB
7211828da706: Loading layer [==================================================>]  9.029MB/9.029MB
48017de0deab: Loading layer [==================================================>]  39.71MB/39.71MB
941e9af6b248: Loading layer [==================================================>]  8.163MB/8.163MB
d1454d6e5ec8: Loading layer [==================================================>]   15.3MB/15.3MB
b680eae73b0b: Loading layer [==================================================>]  11.78kB/11.78kB
8cbab1932ff9: Loading layer [==================================================>]  16.17MB/16.17MB
4004ae50ff22: Loading layer [==================================================>]  1.105MB/1.105MB
09e9466b9548: Loading layer [==================================================>]  391.2kB/391.2kB
6143980f8f5c: Loading layer [==================================================>]  22.53kB/22.53kB
6eb7b0e66a2f: Loading layer [==================================================>]  260.1kB/260.1kB
a52fdd7e31b6: Loading layer [==================================================>]  30.76MB/30.76MB
e822b63f6089: Loading layer [==================================================>]  11.26kB/11.26kB
Loaded image: pureengage-docker-production.jfrog.io/gcxi/gcxi:100.0.026.0001
[gcxi@localhost ~]$
```

4. Execute the following command to verify that the images loaded correctly:
   $ docker images

```
[gcxi@localhost ~]$ docker images
REPOSITORY                                          TAG             IMAGE ID      CREATED       SIZE
pureengage-docker-production.jfrog.io/gcxi/gcxi           100.0.026.0001  bd4cb8c63f7c  7 weeks ago   13.1GB
pureengage-docker-production.jfrog.io/gcxi/gcxi_control   100.0.026.0001  f7154ec5b736  2 months ago  1.5GB
[gcxi@localhost ~]$
```

   These images are repository path, so retag the images

5. Execute the following commands to retag the images:
   $ docker tag pureengage-docker-production.jfrog.io/gcxi/gcxi:100.0.026.0001 gcxi:100.0.026.0001
   $ docker tag pureengage-docker-production.jfrog.io/gcxi/gcxi_control:100.0.026.0001 gcxi_control:100.0.026.0001

```
[gcxi@localhost ~]$
[gcxi@localhost ~]$ docker tag pureengage-docker-production.jfrog.io/gcxi/gcxi:100.0.026.0001 gcxi:100.0.026.0001
[gcxi@localhost ~]$
[gcxi@localhost ~]$
[gcxi@localhost ~]$ docker tag pureengage-docker-production.jfrog.io/gcxi/gcxi_control:100.0.026.0001 gcxi_control:100.0.026.0001
[gcxi@localhost ~]$
[gcxi@localhost ~]$
```

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

```
[gcxi@localhost ~]$
[gcxi@localhost ~]$ docker images
REPOSITORY                                          TAG             IMAGE ID      CREATED       SIZE
gcxi                                                100.0.026.0001  bd4cb8c63f7c  2 months ago  13.1GB
pureengage-docker-production.jfrog.io/gcxi/gcxi     100.0.026.0001  bd4cb8c63f7c  2 months ago  13.1GB
gcxi_control                                        100.0.026.0001  f7154ec5b736  3 months ago  1.5GB
pureengage-docker-production.jfrog.io/gcxi/gcxi_control  100.0.026.0001  f7154ec5b736  3 months ago  1.5GB
[gcxi@localhost ~]$
```

7. Now you can remove the unused and keep only required images :-

   $ docker rmi pureengage-docker-production.jfrog.io/gcxi/gcxi:100.0.026.0001
   $ docker rmi pureengage-docker-production.jfrog.io/gcxi/gcxi_control:100.0.026.0001
   $ docker images

```
[gcxi@localhost ~]$
[gcxi@localhost ~]$ docker rmi pureengage-docker-production.jfrog.io/gcxi/gcxi:100.0.026.0001
Untagged: pureengage-docker-production.jfrog.io/gcxi/gcxi:100.0.026.0001
[gcxi@localhost ~]$ docker rmi pureengage-docker-production.jfrog.io/gcxi/gcxi_control:100.0.026.0001
Untagged: pureengage-docker-production.jfrog.io/gcxi/gcxi_control:100.0.026.0001
[gcxi@localhost ~]$
[gcxi@localhost ~]$ docker images
REPOSITORY     TAG             IMAGE ID       CREATED        SIZE
gcxi           100.0.026.0001  bd4cb8c63f7c   2 months ago   13.1GB
gcxi_control   100.0.026.0001  f7154ec5b736   3 months ago   1.5GB
[gcxi@localhost ~]$
```

# Starting Genesys CX Insights containers

For docker-compose deployment, we need only **docker-compose.yml** from the package "**IP_CustExpInsights_10000260001b1_ENU_linux.tar.gz"**.

1. Extract the **IP_CustExpInsights_10000260001b1_ENU_linux.tar.gz**

   $ tar -xvf IP_CustExpInsights_10000260001b1_ENU_linux.tar.gz

```
[gcxi@localhost ~]$ tar -xvf IP_CustExpInsights_10000260001b1_ENU_linux.tar.gz
ip/
ip/data.tar.gz
ip/gunzip
ip/install.sh
ip/installer.tar.gz
ip/ip_description.xml
ip/iscript.tar.gz
ip/ospatchlist.txt
ip/read_me.html
ip/tar
ip/tar_gunzip_license.txt
[gcxi@localhost ~]$
```

2. Then go to extracted directory called "**ip**" and extract "**data.tar.gz**".

```
[gcxi@localhost ip]$ tar -xvf data.tar.gz
docker-compose.yml
gcxi-backup.yaml
gcxi-cleanup.yaml
gcxi-ingress.yaml
gcxi-init.yaml
gcxi-postgres.yaml
gcxi-restore.yaml
gcxi-secrets.yaml
gcxi.properties
gcxi.yaml
infra.yaml
nginx-configmap.yaml
nginx-daemon.yaml
postgre-mstr_hist.pgdump
postgre-mstr_meta.pgdump
[gcxi@localhost ip]$
```

3. Now copy **docker-compose.yml** file to the path where the **gcxi.tar.gz & gcxi_control.tar.gz** is placed**.**

   $ pwd

   $ cp docker-compose.yml /home/gcxi/

```
[gcxi@localhost ip]$ pwd
/home/gcxi/ip
[gcxi@localhost ip]$ cp docker-compose.yml /home/gcxi/
[gcxi@localhost ip]$
```

4. Verify the files using below command.

   $ ls -lrth /home/gcxi/ | grep -i -e gcxi_control.tar.gz -e gcxi.tar.gz -e IP_CustExpInsights_10000260001b1_ENU_linux.tar.gz -e docker-compose.yml

```
[gcxi@localhost ip]$ ls -lrth /home/gcxi/ | grep -i -e gcxi_control.tar.gz -e gcxi.tar.gz -e IP_CustExpInsights_10000260001b1_ENU_linux.tar.gz -e docker-compose.yml
-rwxrwxrwx. 1 gcxi gcxi 644M Feb  3 01:36 gcxi_control.tar.gz
-rwxrwxrwx. 1 gcxi gcxi 5.2G Feb  3 01:37 gcxi.tar.gz
-rwxrwxrwx. 1 gcxi gcxi  13M Mar 24 14:46 IP_CustExpInsights_10000260001b1_ENU_linux.tar.gz
-rwxr-x---. 1 gcxi gcxi 9.0K Mar 28 18:29 docker-compose.yml
```

## GCXI Build Procedure

To build GCXI - open **"docker-compose"** file and update the values as follows. We recommended to use preferred YAML file editors to avoid white/blank space, tab, line order issues.

- Set necessary versions of gcxi in gcxi-control and gcxi-0 images.

  Example:

  If you're using below GCXI versions:

  $ docker images

```
[gcxi@localhost ~]$
[gcxi@localhost ~]$ docker images
REPOSITORY      TAG              IMAGE ID       CREATED         SIZE
gcxi            100.0.026.0001   bd4cb8c63f7c   8 weeks ago     13.1GB
gcxi_control    100.0.026.0001   f7154ec5b736   2 months ago    1.5GB
[gcxi@localhost ~]$
```

  Then docker-compose.yml should have below values

  $ cat docker-compose.yml | grep -i image

```
[gcxi@localhost ~]$ cat docker-compose.yml | grep -i image
  # ## contains built-in sample gim db, but requires 4 gb image gcxi_postgres
  #   image: gcxi_postgres
  ## postgres image + init job from gcxi_contorl image, no embedded gim db
    image: postgres:12
    image: gcxi_control:100.0.026.0001
    image: gcxi:100.0.026.0001
[gcxi@localhost ~]$
```

- Set necessary database information's in gcxi-0 block below environment section.
- Check the (**telnet**) connectivity between GIM DB and GCXI host server.
- DSNDEF1 would consider as primary.
- <DSN_NAME> is the DSN type. only following values are supported: **GCXI_GIM_DB** or **IWD_DB**

**Sample MSSQL Definition: -**

- DSNDEF1=DSN_NAME=GCXI_GIM_DB;DB_TYPE=SQLSERVER;DB_TYPE_EX=Microsoft SQL Server 2012;HOST=192.168.29.192;PORT=1433;DB_NAME=GIM_MSSQL;LOGIN=login_id;PASSWORD=Password

**Sample Oracle Definition:-**

- DSNDEF1=DSN_NAME=IWD_DB;DB_TYPE=ORCLW;DB_TYPE_EX=Oracle 12c;HOST=192.168.124.105;PORT=1521;ORCL_SID=ci;LOGIN=login_id;PASSWORD=password;ENCODING=UTF8

**Sample Postgres Definition:-**

-
DSNDEF1=DSN_NAME=IWD_DB;DB_TYPE=POSTGRESQL;DB_TYPE_EX=PostgreSQL;HOST=192.168.124.105;PORT=5432;DB_NAME=dm_gcxi_85;LOGIN=login_id;PASSWORD=password;ENCODING=NONUTF8

**Sample docker-compose.yml**

Sample yaml file .txt

Tip:

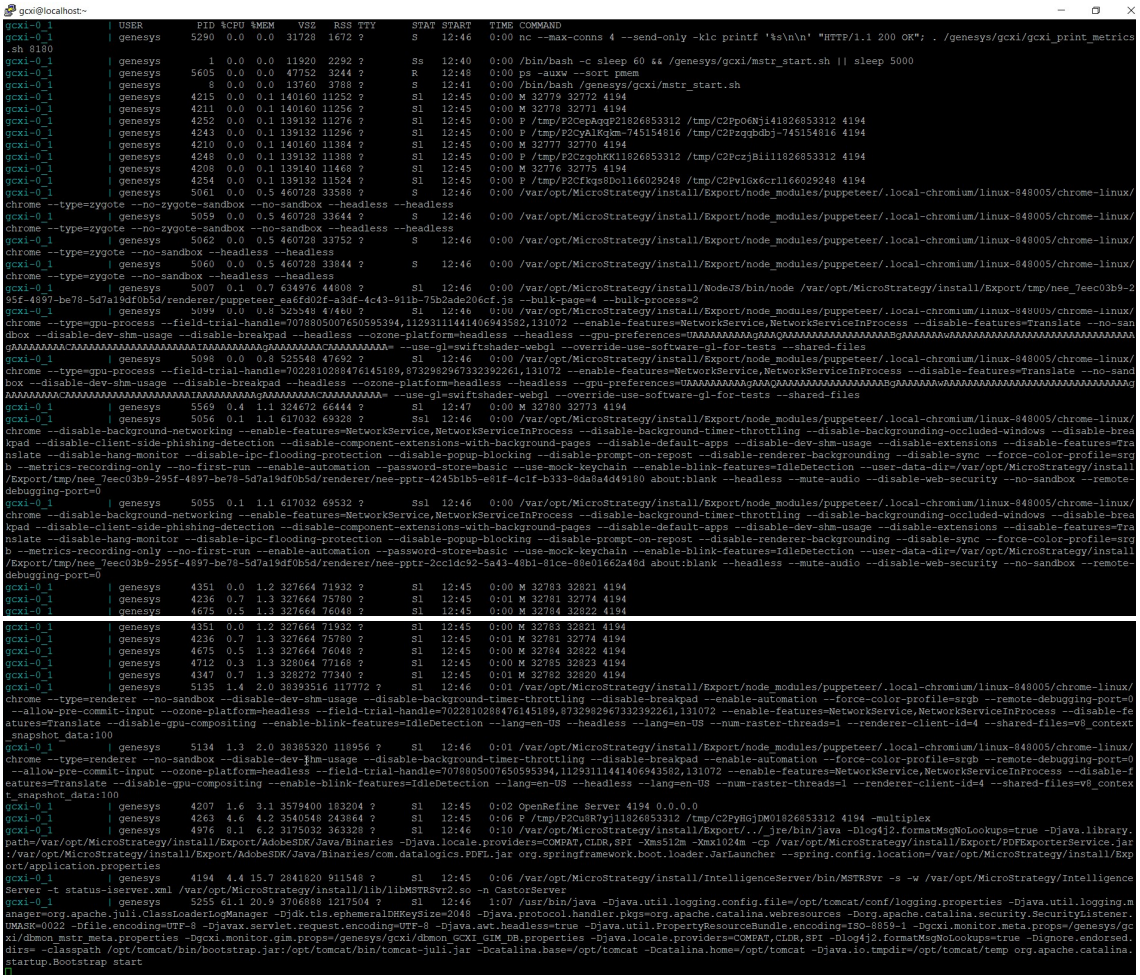
- When starting the container, if you encounter an error about "exited with status 1", verify that all variables in the docker-compose.yml file.
- Beginning with release 100.0.020.0000 (9.0.020), the default user for all Genesys CX Insights images has changed to genesys (id = 500); previously, the default user was root. this can result in an error due to volume mount permission settings;
  To prevent we recommend below steps:
  $ mkdir /mnt/log
  $ mkdir -p /genesys/gcxi/shared
  $ chown -R 500:500 /mnt/log
  $ chown -R 500:500 /genesys/gcxi/shared

```
[root@localhost ~]# mkdir /mnt/log
[root@localhost ~]# mkdir -p /genesys/gcxi/shared
[root@localhost ~]# chown -R 500:500  /mnt/log
[root@localhost ~]# chown -R 500:500  /genesys/gcxi/shared
[root@localhost ~]#
```

Once you done with prerequisites and necessary changes to docker-compose.yml file, you can start the containers using below command:

$ docker-compose -f docker-compose.yml up -d

```
[gcxi@localhost ~]$
[gcxi@localhost ~]$ docker-compose -f docker-compose.yml up -d
Starting gcxi_gcxi-postgres_1 ... done
Starting gcxi_gcxi-control_1  ... done
Recreating gcxi_gcxi-0_1      ... done
[gcxi@localhost ~]$
```



Sample deployment output for reference:

Sample output.txt

Check the logs and make sure below services up and running.

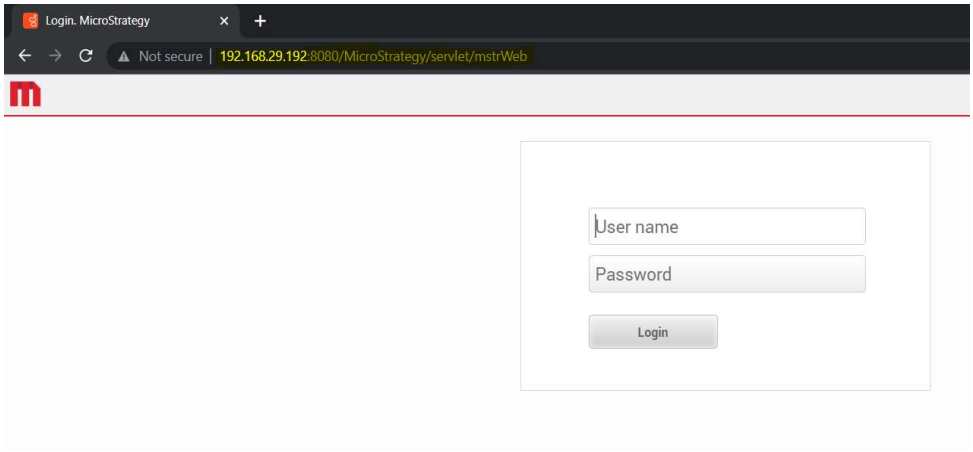
$ docker-compose logs | grep -i -e "gcxi_gcxi-control_1 exited with code 0" -e "The IntelligenceServer is starting" -e "PDF Export Service is started." -e "Tomcat started."

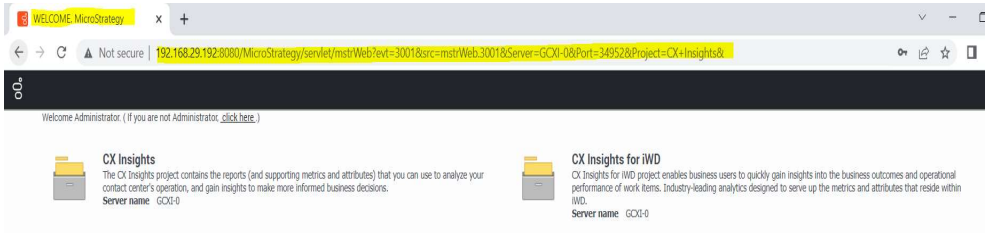
```
[gcxi@localhost ~]$
[gcxi@localhost ~]$ docker-compose logs | grep -i -e "gcxi_gcxi-control_1 exited with code 0" -e "The In
telligenceServer is starting" -e "PDF Export Service is started." -e "Tomcat started."
gcxi-0_1         | The IntelligenceServer is starting
gcxi-0_1         | PDF Export Service is started.
gcxi-0_1         | Tomcat started.
[gcxi@localhost ~]$
[gcxi@localhost ~]$
```

## Landing Page:

- **To view reports and dashboards**, visit http://<VM IP>:8080/MicroStrategy/servlet/mstrWeb, and log in as an administrator. Where <VM IP> is the hostname or ip-address where docker-compose is running.

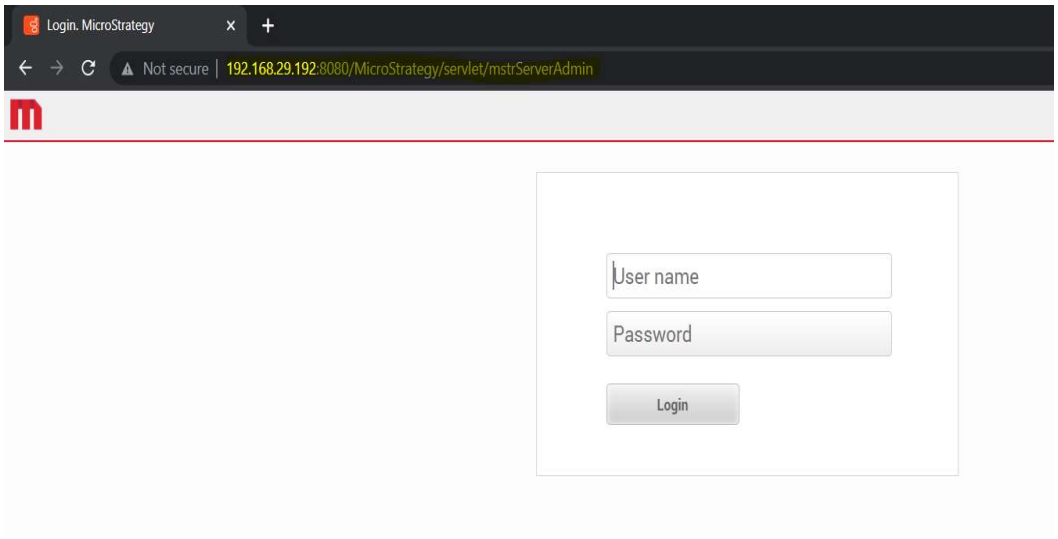
  ***Username = administrator | Password = Genesys_0***

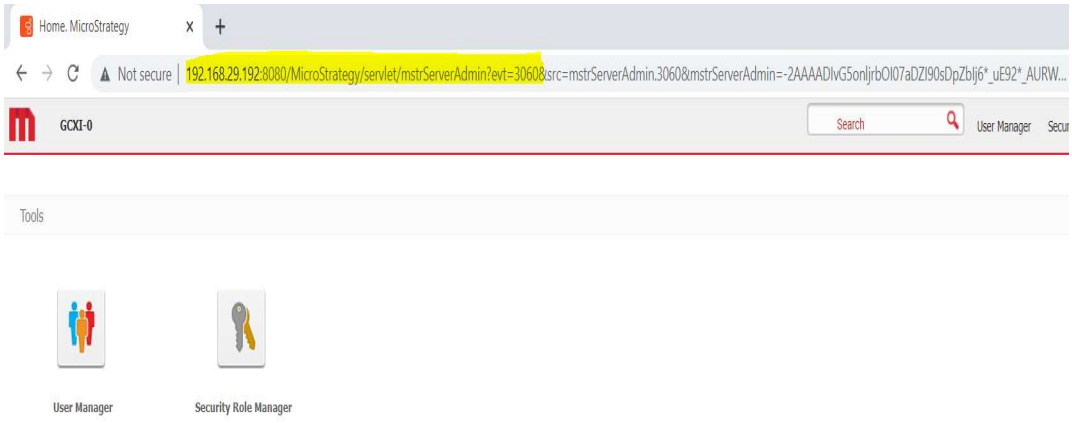


- **To manage users and security roles**, visit http://<VM IP>:8080/MicroStrategy/servlet/mstrServerAdmin, and log in as an administrator.

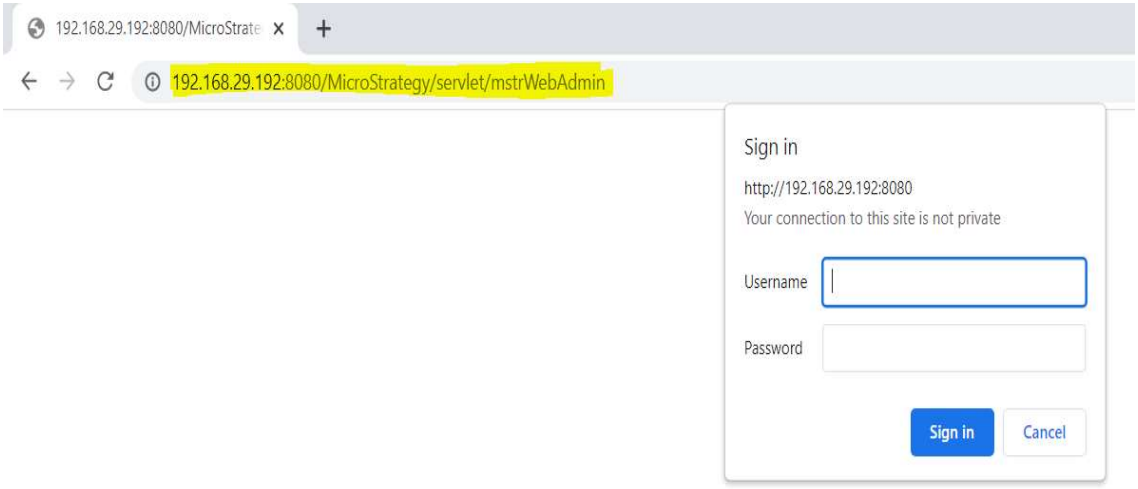
  ***Username = administrator | Password = Genesys_0***

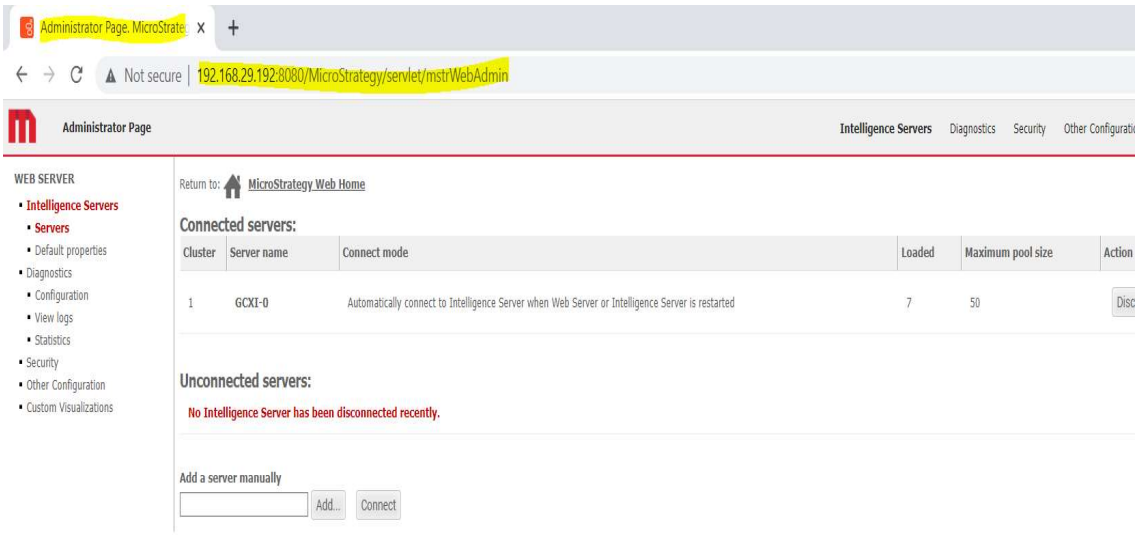


- **To manage MSTR Web server settings**, visit http://<VM IP>:8080/MicroStrategy/servlet/mstrWebAdmin, and log in as admin.

  ***Username = admin | Password = gcxi_admin***



## Cleaning up your Machine

**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 to Clean up your Machine:

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.

```
[root@localhost ~]#
[root@localhost ~]# sudo systemctl start docker
[root@localhost ~]#
```

2. Execute the following command to see what containers are running:
   $ docker ps -a

```
[gcxi@localhost ~]$ docker ps -a
CONTAINER ID   IMAGE                                      COMMAND                  CREATED      STATUS                       PORTS                                       NAMES
928d23fe099a   gcxi:100.0.026.0001                        "/bin/bash -c 'sleep…"   4 days ago   Exited (0) 4 days ago                                                    gcxi_gcxi-0_1
57a19d396e0e   mcr.microsoft.com/mssql/server:2017-latest "/opt/mssql/bin/nonr…"   4 days ago   Exited (255) 2 minutes ago   0.0.0.0:1433->1433/tcp, :::1433->1433/tcp   sql-server-db
8d6aa4cc40d3   gcxi_control:100.0.026.0001                "/bin/bash -c 'sleep…"   4 days ago   Exited (0) 4 days ago                                                    gcxi_gcxi-control_
1
fa4e04f23cf5   postgres:12                                "docker-entrypoint.s…"   4 days ago   Exited (255) 2 minutes ago   0.0.0.0:5432->5432/tcp, :::5432->5432/tcp   gcxi_gcxi-postgres
_1
996f1398d606   gcxi:100.0.026.0001                        "/bin/bash -c ${GCXI…"   4 days ago   Exited (143) 4 days ago                                                  sweet_engelbart
[gcxi@localhost ~]$
```

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>

```
[gcxi@localhost ~]$ docker rm -f 928d23fe099a 57a19d396e0e 8d6aa4cc40d3
928d23fe099a
57a19d396e0e
8d6aa4cc40d3
[gcxi@localhost ~]$
```

4. Execute the following command to remove all existing containers, images and volumes:
   $ docker system prune -af --volumes

```
[gcxi@localhost ~]$ docker system prune -af --volumes
Deleted Containers:
fa4e04f23cf512f3e5856c43b1c363f714ae314aa8bd1e92c3c711dde7b41fa0

Deleted Networks:
gcxi_gcxi
mssql_default

Deleted Volumes:
4e15bf1689b65a40d2bcd74e7fd493a0b143692bfdccb2c3fbca2fbffa85ef91
8e83b8884429fbdd65bf32eabf45d18491f8129cf0e4e9711381da45329a0b1f
gcxi_gcxi_postgres
gcxi_mstr_log_01
gcxi_mstr_shared
2a1f24aff9dbfd8869e436a97227b768f3a69aa91d47a19b4eaa118ae7cac47e

Deleted Images:
untagged: postgres:12
untagged: postgres@sha256:dccefcef098597b666c0a7012ffdb0187a84fd48868c1165cb72d031f0e36e7c
deleted: sha256:cb268752044768ac45106e08fdde76d264de57f0404d3552ee91ba2953e80f28
deleted: sha256:22714ca6d2f490db2e3d98577c48bfbf2f0ad172a3ee8d556bd5d818bc3bd0de
deleted: sha256:c293280a73cc3264efeed85ddd2dc9b52b25bf250fe32ffc274045c60a564e29
deleted: sha256:197d62e22045a3c434cdd804ee4a7a2c240966d86e95e12728fdb222b41dd00c
```

Refer the entire output in below attachment.

Docker Prune Volumes.txt