Genesys Customer Experience Insights Deployment Guide

Installing Kubernetes and Docker in offline scenarios

12/16/2025

Contents

Installing Kubernetes and Docker in offline scenarios

Disclaimer

Genesys is committed to diversity, equality, and inclusivity. This includes using appropriate terms in our software and documentation. Therefore, Genesys is removing non-inclusive terms. For third-party products leveraged by Genesys that include such terms, Genesys uses the following as replacements.

- For the terms master/slave, Genesys uses “primary” and “secondary” or “primary” and “replica,” with exceptions for their use in third-party commands.
- For the terms blacklist/whitelist, Genesys uses blocklist/allowlist.
- For the term master, when used on its own, Genesys uses main wherever possible.

This page describes example steps to prepare a system for the installation of Genesys Customer Experience Insights (Genesys CX Insights). Use these instructions in environments where it is not possible to access the internet or other external networks from the machines / network where you plan to install Genesys CX Insights (*offline scenarios*). If your deployment environment has access to the internet (*online scenarios*), use the (simplified) instructions on Installing Kubernetes and Docker in online scenarios.

For additional information about Docker, see the *Genesys Docker Deployment Guide*.

Important

Notes:

- This page provides an example scenario, using Kubernetes release 1.21.2 and Docker version 19.03-ce, on CentOS Linux release 7.5.1804 (Core) on Amazon Web Services (AWS). **Always use the latest approved releases of Kubernetes and Docker.**
- This page does not describe all deployment scenarios, and is applicable to only the indicated software release (Operating System, Docker, Kubernetes). For other releases, the required steps may vary.

Before You Begin

Ensure that you have a suitably-prepared environment, as described in Before you install Genesys CX Insights, which must include suitably-prepared hosts (real machines, virtual machines (VM), or cloud instances) with Red Hat Enterprise Linux 7.5 (or a later 7.x release) / CentOS Linux 7.5 (or a later 7.x release) hosts with system suites installed.

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. If you must use mismatched operating systems, see the YUM and Docker documentation.

- **File transfer** — Some method of transferring files from the online machine to the offline network where you will install Genesys CX Insights.

Download and Install Docker

Use the procedures in this section to download and install Docker.

Procedure: Download Docker (online machine)

Purpose: Use the steps in this procedure to download Docker.

Steps

Complete the following steps on the **online** machine:

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

   ```
   yum install -y yum-utils
   ```

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

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

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.

Procedure: Install Docker (offline machine)

Purpose: Use the steps in this procedure to install Docker.

Steps

Complete the following steps on each machine in the cluster on the **offline** machine (where <your_rpm_dir> is a directory on your offline machine where you placed the files):

1. Copy the Docker files, downloaded in the Download Docker (online machine), from the online machine to the offline machine.
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
```

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
   ```

6. Execute the following command to install Docker:

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

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

   ```
   systemctl enable docker
   systemctl start docker
   ```

8. Execute the following commands to verify docker:

   ```
   systemctl status docker
   docker version
   ```

Download and Install Kubernetes

Use the procedures in this section to download and install Kubernetes utilities.

Procedure: Download Kubernetes utilities (online machine)

Purpose: Use the steps in this procedure to download Kubernetes utilities.

Steps

Complete the following steps on the **online** machine:

1. Execute the following commands to install the Kubernetes repository:

```
cat <<EOF > /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
enabled=1
gpgcheck=1
repo_gpgcheck=1
gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
EOF
```

2. Execute the following command to download the Kubernetes utilities:

```
yumdownloader --assumeyes --destdir=<your_rpm_dir> --resolve yum-utils kubeadm-1.21.* kubelet-1.21.* kubectl-1.21.* ebtables
```

Procedure: Install Kubernetes utilities (offline machine)

Purpose: Use the steps in this procedure to install Kubernetes utilities.

Steps

Complete the following steps on each machine in the cluster on the **offline** machine:

1. Copy the Kubernetes utilities files, downloaded in the preceding steps, from the online machine to the offline machine.
2. Execute the following commands to install Kubernetes:

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

   where <your_rpm_dir> is a directory on your offline machine where you placed the files.
3. Execute the following command to run **kubeadm**, which returns a list of required images:

   ```
   kubeadm config images list
   ```

   A list of the required images appears, similar to the following:

   ```
   k8s.gcr.io/kube-apiserver:v1.21.2
   k8s.gcr.io/kube-controller-manager:v1.21.2
   k8s.gcr.io/kube-scheduler:v1.21.2
   k8s.gcr.io/kube-proxy:v1.21.2
   k8s.gcr.io/pause:3.4.1
   k8s.gcr.io/etcd:3.4.13-0
   k8s.gcr.io/coredns/coredns:v1.8.0
   ```

Download and Install Kubernetes Images

Use the procedures in this section to download and install Kubernetes images.

Procedure: Download Kubernetes Images (online machine)

Purpose: Use the steps in this procedure to download the Kubernetes images.

Steps

Complete the following steps on the **online** machine:

1. **Install Docker on each machine** by following the instructions in the Docker installation documentation.
2. For each image that appears in the list that was returned in the preceding step, execute the following commands to pull the image and save it as a TAR archive:

   ```
   docker pull k8s.gcr.io/<image name>
   docker save k8s.gcr.io/<image name> > <image name>.tar
   ```

   where <image name> is the name of the image to pull, for example:

   ```
   docker pull k8s.gcr.io/kube-apiserver:v1.21.2
   docker save k8s.gcr.io/kube-apiserver:v1.21.2 > kube-apiserver_v1.21.2.tar
   ```

Procedure: Load Kubernetes Images (offline machine)

Purpose: Use the steps in this procedure to load the Kubernetes Images.

Steps

Complete the following steps on each machine in the cluster on the **offline** machine:

1. Copy the Kubernetes images, downloaded in Download Kubernetes Images (online machine), from the online machine to the offline machine.
2. For each image, execute the following command to unpack the image:

   ```
   docker load < <image name>.tar
   ```

   where <image name> is the name of the image to unpack, for example:

   ```
   docker load < kube-apiserver_v1.21.2.tar
   ```

Download and Install Kubernetes Network

Use the procedures in this section to download and install Kubernetes networking files.

Procedure: Download networking files (online machine)

Purpose: Use the steps in this procedure to download Kubernetes networking files.

Steps

Complete the following steps on the **online** machine:

1. Execute the following command to download the yaml descriptor:

   ```
   wget https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
   ```

2. Open the **kube-flannel.yml** file, and find the line that indicates the flannel image version. For example, in the following string, the version is **v0.14.0**:

   ```
   image: quay.io/coreos/flannel:v0.14.0-amd64
   ```

3. Execute the following commands to download and save the image:

   ```
   docker pull quay.io/coreos/flannel:v<version>-amd64
   docker save quay.io/coreos/flannel:v<version>-amd64 > flannel_v<version>-amd64.tar
   ```

   where <version> is the flannel image version you found in the previous step. For example:

   ```
   docker pull quay.io/coreos/flannel:v0.14.0-amd64
   docker save quay.io/coreos/flannel:v0.14.0-amd64 > flannel_v0.14.0_v1.tar
   ```

Procedure: Install Kubernetes networking files (offline machine)

Purpose: Use the steps in this procedure to install Kubernetes networking files.

Steps

Complete the following steps on each machine in the cluster on the **offline** machine:

1. Copy the networking files, downloaded in Download networking files (online machine), from the online machine to the offline machine.
2. Execute the following command to unpack the networking image:

   ```
   docker load < flannel_v<version>_v1.tar
   ```

   where <version> is the version of the flannel software, for example:

   ```
   docker load < flannel_v0.14.0_v1.tar
   ```

Download and Install NGINX

Use the procedures in this section to download and install NGINX.

Procedure: Download NGINX images (online machine)

Purpose: Use the steps in this procedure to download NGINX images.

Steps

Complete the following steps on the **online** machine:

1. Execute one of the following commands to download yaml descriptors:

```
wget https://raw.githubusercontent.com/kubernetes/ingress-nginx/nginx-0.30.0/deploy/static/namespace.yaml && \
 wget https://raw.githubusercontent.com/kubernetes/ingress-nginx/nginx-0.30.0/deploy/static/rbac.yaml
```

2. Execute the following command to download and save the image:

```
docker pull quay.io/kubernetes-ingress-controller/nginx-ingress-controller:0.30.0
 docker save quay.io/kubernetes-ingress-controller/nginx-ingress-controller:0.30.0 > nginx-ingress-controller_0.30.0.tar
```

Procedure: Load NGINX images (offline machine)

Purpose: Use the steps in this procedure to load NGINX images.

Steps

Complete the following steps on each machine in the cluster on the **offline** machine:

1. Copy the NGINX images, downloaded in Download NGINX images (online machine), from the online machine to the offline machine.
2. Execute the following command to unpack the NGINX image:

```
docker load < nginx-ingress-controller_0.30.0.tar
```

Deploy cluster

Procedure: Example: Deploying Kubernetes cluster

Purpose: This example procedure illustrates one scenario to complete the installation and preparation of Kubernetes and Docker in offline scenarios. For more information, see Kubernetes documentation.

Prerequisites

Ensure that you have a suitably-prepared environment, as described in Before you install Genesys CX Insights.

Steps

Perform steps 1-5 on each machine, and then complete subsequent steps as indicated:

1. Log in with root access.
2. Complete the following steps to disable swap:
    1. Execute the following command to disable swap for the current session:

       ```
       swapoff -a
       ```

    2. To permanently disable swap, remove the swap partition using `fdisk` or `parted`. Be sure to remove only the swap partition, as removing another partition could cause serious problems. The changes take effect after system restart.
3. Ensure that SELinux is in permissive mode. For example, execute the following commands:

   ```
   setenforce 0
   sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
   ```

4. Ensure that the config option **sysctl > net.bridge.bridge-nf-call-iptables** is set to 1. For example, execute the following command:

```
cat <<EOF >  /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
EOF
sysctl --system
```

5. Optionally, configure kubectl autocompletion:

```
echo "source <(kubectl completion bash)" >> ~/.bashrc
```

Complete the preceding steps on each PRIMARY and SECONDARY machine before continuing.

6. **On the PRIMARY machine only, create a cluster, deploy the Flannel network, and schedule pods:**
   1. Execute the following command to retrieve the version number for Kubernetes:

      ```
      kubectl version
      ```

      The Kubernetes version is displayed.

   2. Execute the following command to set up a Kubernetes cluster:

      ```
      kubeadm init --pod-network-cidr=10.244.0.0/16 --kubernetes-version=v<version>
      ```

      Where `<version>` is the version of Kubernetes retrieved in the preceding step. For example:

      ```
      kubeadm init --pod-network-cidr=10.244.0.0/16 --kubernetes-version=v1.21.2
      ```

      Note that this command produces large volumes of output, and should include a string similar to: `kubeadm join --token <token> <primary-ip>:<primary-port> --discovery-token-ca-cert-hash sha256:<hash>` For example: `kubeadm join 10.51.29.20:6443 --token dmijep.e1qmgc4o3sh22pwd --discovery-token-ca-cert-hash sha256:ef846cf825d6234aa7b123723bc312a7ff72a14facf9e3a02bc34a708fb3c877`
      IMPORTANT: This string is required in a later step. Find the string in the output, then copy and save it. Alternatively, redirect command output to a file before completing this step.

   3. Execute the following command to verify the node is running:

```
kubectl get nodes
```

The Control plane node should have a status of NotReady, similar to the following output:

```
gcxi-doc-kube0   NotReady   master    3m        v1.21.2
```

4. Execute the following commands to configure kubectl to manage your cluster:

```
grep -q "KUBECONFIG" ~/.bashrc || {
    echo 'export KUBECONFIG=/etc/kubernetes/admin.conf' >> ~/.bashrc
    . ~/.bashrc
}
```

5. Deploy the Flannel overlay network on the PRIMARY machine:
   1. Execute the following command to initiate the Flannel network:

```
kubectl apply -f <destination path>/kube-flannel.yml
```

Where <destination path>

This may take several minutes to complete; avoid interrupting the process.

   2. Execute the following command to ensure that **kube-dns*** pods (or **coredns**, depending on the release of kubernetes you are using) are running (not **pending** or any other status):

```
kubectl get pods --all-namespaces
```

6. Schedule pods — Note that, for security reasons, the cluster does not schedule pods on the PRIMARY by default. Optionally, to configure the cluster to schedule on the PRIMARY machine, execute the following command on the PRIMARY machine. Executing this command removes the `node-role.kubernetes.io/master` taint from any nodes that have it, so that the scheduler can schedule pods everywhere:

```
kubectl taint nodes --all node-role.kubernetes.io/master-
```

7. **On the SECONDARY machines only, join the SECONDARY to the cluster:**
   Execute the following command (replacing <token>, <primary port>, and <hash> with appropriate values):

```
kubeadm join --token <token> <primary-ip>:<primary-port> --discovery-token-ca-cert-hash sha256:<hash>
```

(or paste in the string you saved in a previous step).

For example: `kubeadm join 10.51.29.20:6443 --token dmijep.e1qmgc4o3sh22pwd --discovery-token-ca-cert-hash sha256:ef846cf825d6234aa7b123723bc312a7ff72a14facf9e3a02bc34a708fb3c877`

8. **On the PRIMARY machine, verify nodes:**
   1. Ensure that all nodes are in the **ready** state.
   2. Execute the following command on the PRIMARY machine:

      ```
      kubectl get nodes
      ```

      The nodes should have a status of Ready, similar to the following output:

      ```
      gcxi-doc-kube0   Ready     master    3m        v1.21.2
      gcxi-doc-kube1   Ready     <none>    22s       v1.21.2
      ```

Next Steps

After you have installed and configured Docker and Kubernetes, proceed to Installing Genesys CX Insights.

Download and Install PostgreSQL Image

Use the procedures in this section to download and install the PostgreSQL image.

Procedure: Download PostgreSQL Image (online machine)

Purpose: Use the steps in this procedure to download the PostgreSQL image.

Steps

Complete the following steps on the **online** machine:

1. 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, for example:

   ```
   docker pull postgres:12
   docker save postgres:12 | gzip > postgres.tar.gz
   ```

Procedure: Load PostgreSQL Image (offline machine)

Purpose: Use the steps in this procedure to load the PostgreSQL Image.

Steps

Complete the following steps on the **offline** machine:

1. Copy the PostgreSQL images, downloaded in Download PostgreSQL Images (online machine), from the online machine to the offline machine.
2. Execute the following command to unpack the image:

   ```
   docker load < postgres.tar.gz
   ```