



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Genesys Customer Experience Insights Deployment Guide

Installing Kubernetes and Docker in online scenarios

5/7/2025

Contents

- 1 Installing Kubernetes and Docker in online scenarios
 - 1.1 Before you begin
 - 1.2 Install Docker and Kubernetes

Installing Kubernetes and Docker in online 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), including the installation of Docker and Kubernetes on a Linux server. Use these instructions on this page in environments where it is possible to access the internet or other external networks from the machines / network where you plan to install Genesys CX Insights (*online scenarios*). If your deployment environment cannot access the internet (*offline scenarios*), follow the instructions on [Installing Kubernetes and Docker in offline scenarios](#) instead.

For additional information about Docker, see the [Genesys Docker Deployment Guide](#).

Important

Notes:

- This page provides an example scenario using Kubernetes release 1.26.3, Docker version 20.10-ce, and cri-dockerd adapter version 0.3.1 to integrate Docker Engine with Kubernetes, with CentOS Linux release 7.9.2009 (Core). Please make sure that your infrastructure components, such as Linux, Kubernetes, CRI engine and Docker (if you use Docker), are of supported versions and have an adequate update cadence.
- Components such as Kubernetes, Docker, and containerd are community-driven products with a very fast upgrade cycle. Installation instructions for these components, provided by this guide, may sometimes not be the most current. We recommend that you double-check the CRI/Docker/Kubernetes installation steps on their corresponding sites.
- This page does not describe all deployment scenarios, and is applicable only to the indicated software release (Operating System, Container Runtime, Kubernetes). For other releases or CRI, 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 7.9 (or a later 7.x release) / CentOS Linux 7.9 (or a later 7.x release) hosts with system suites installed.

Install Docker and Kubernetes

This section describes a typical production deployment of Docker and Kubernetes (K8S), which is an open-source system for automating deployment, scaling, and management of containerized Docker applications, sometimes called a container orchestration system. Before you deploy MicroStrategy and Genesys CX Insights, you must deploy both Docker and Kubernetes. Docker containers and Kubernetes descriptors simplify Genesys CX Insights deployment, and provide flexibility, scalability, and reliability, and simplified future maintenance of Genesys CX Insights.

To deploy Docker and Kubernetes, and configure Kubernetes clusters, follow the [Kubernetes installation instructions](#). The following section provides an example of a Kubernetes deployment process on Red Hat Enterprise Linux 7.9 (or a later 7.x release) / CentOS Linux 7.9 (or a later 7.x release).

Example steps to install Kubernetes

The exact installation procedure for Kubernetes varies significantly depending on numerous factors, including the type of machines in your environment (these could be real machines, virtual machines, or cloud machines), Operating System, networking model, planned load, and Kubernetes version. For information about the exact steps you must follow to install Kubernetes and Docker in your environment, see [Kubernetes installation instructions](#).

The following procedure outlines the steps to follow in one common deployment scenario.

Procedure: Example: Deploying Kubernetes clusters and loading Docker

Purpose: This example procedure illustrates one scenario for the installation of Kubernetes and Docker. For more information:

- Detailed information about the Operating System requirements, see documentation on the [Kubernetes web site](#).

- These instructions are based on [Kubernetes documentation](#).

Prerequisites

- This sample installation is intended for an Red Hat Enterprise Linux 7.9 (or a later 7.x release) / CentOS Linux 7.9 (or a later 7.x release) environment, as described in [Before you install Genesys CX Insights](#). The examples on this page use Docker CE, but Kubernetes supports multiple container runtimes, and you can choose any other Kubernetes-supported engine. Ensure that you have properly installed and configured a supported engine before you proceed.

Important

Some steps can take some time for processing to complete. Genesys does not recommend interrupting any of the processes initiated in this procedure.

Steps

Perform steps 1-9 on each machine, and then subsequent steps as directed.

1. **Prepare hosts** — Prepare CentOS Linux 7.9 (or a later 7.x release) hosts with system suites installed. Earlier 7.x releases of CentOS Linux may work, but have not been tested by Genesys, and may require additional updates or package installation; see the [Kubernetes documentation](#) for more information. These can be real machines, virtual machines (VM), or cloud instances.

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.

Note the reviewers: Moved "troubleshooting" step to the end of the procedure.

2. Log in with root access.
3. **Install Docker on each machine** by following the instructions in the [Docker installation documentation](#).
4. Execute the following command to verify the Docker CE installation:

```
docker --version
```

The Docker version appears, such as Docker version 20.10.22.

5. 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.

6. 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
```

7. Ensure that relevant **sysctl** config option are 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
```

8. **Install kubeadm, kubelet and kubectl** — On each machine, log in as a user having root privileges, (`sudo -i bash`), and execute the following commands:

```
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=0
gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
EOF

yum install -y kubelet kubeadm kubectl
```

9. Install cri-dockerd adapter by following the instructions [here](#) or follow the steps provided below; On each machine, log in as a user having root privileges, (`sudo -i bash`), and execute the following commands:

```
# install and configure cri-dockerd service
wget https://github.com/Mirantis/cri-dockerd/releases/download/v0.3.1/cri-dockerd-0.3.1-3.el7.x86_64.rpm
yum install cri-dockerd-0.3.1-3.el7.x86_64.rpm

wget https://raw.githubusercontent.com/Mirantis/cri-dockerd/master/packaging/systemd/cri-docker.service
wget https://raw.githubusercontent.com/Mirantis/cri-dockerd/master/packaging/systemd/cri-docker.socket
mv cri-docker.socket cri-docker.service /etc/systemd/system/

systemctl daemon-reload
systemctl enable cri-docker.service
systemctl enable --now cri-docker.socket

# check cri-dockerd socket
systemctl status cri-docker.socket
```



```
# configure the kubelet to use cri-dockerd
a. Open /var/lib/kubelet/kubeadm-flags.env on each affected node (create the file if it does not exist).
b. Modify the --container-runtime-endpoint flag to unix:///var/run/cri-dockerd.sock
i.e. KUBELET_KUBEADM_ARGS="--container-runtime-endpoint=unix:///var/run/cri-dockerd.sock --pod-infra-container-
image=registry.k8s.io/pause:3.9"

# restart the kubelet
systemctl restart kubelet
```

10. Optionally, configure kubectl autocompletion:

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

Complete the preceding steps on each machine before continuing.

11. **On the Control plane machine only, create a cluster, and deploy the Flannel network:**

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

```
kubeadm init --pod-network-cidr=10.244.0.0/16 --cri-socket=unix:///var/run/cri-dockerd.sock
```

Note that this command produces large volumes of output, and should include a string similar to: `kubeadm join --token <token> <cpnode-ip>:<cpnode-port> --discovery-token-ca-cert-hash sha256:<hash>` For example: `kubeadm join 10.51.29.20:6443 --token dmijep.elqmgc4o3sh22pwd --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.

2. 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          v1.26.3
```

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

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

4. Deploy the Flannel overlay network on the Control plane node machine:

1. Execute the following command to initiate the Flannel network:

```
kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
```

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
```

12. On the worker machines only, join the worker machines to the cluster.

Executing 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> --cri-socket=unix:///var/run/cri-dockerd.sock
```

(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 --cri-socket=unix:///var/run/cri-dockerd.sock`

13. On the Control plane node machine, verify nodes:

1. Ensure that all nodes are in the **ready** state.
2. Execute the following command on the Control plane node machine:

```
kubectl get nodes
```

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

```
gcxi-doc-kube0   Ready    master    3m      v1.26.3
gcxi-doc-kube1   Ready    <none>    22s     v1.26.3
```

Troubleshooting tips

Execute the following troubleshooting commands if you encounter configuration issues with the CentOS Linux packages:

1. This step prevents the error `Requires: container-selinux >= 2.9:`

```
sudo yum install -y http://mirror.centos.org/centos/7/extras/x86_64/Packages/container-selinux-2.107-3.el7.noarch.rpm
```

It may take a little time for this step to complete.

2. This step prevents the error `libtool-ltdl-2.4.2-22.el7_3.x86_64 FAILED:`

```
sudo yum install http://mirror.centos.org/centos/7/os/x86_64/Packages/libtool-ltdl-2.4.2-22.el7_3.x86_64.rpm
```

A message appears, similar to the following:

```
Total size: 66 k   Installed size: 66 k   Is this ok [y/d/N]:
```

Enter y to continue.

3. This step prevents errors similar to `Package: docker-ce-18.03.1.ce-1.el7.centos.x86_64 (docker-ce-stable) Requires: pigz:`

```
sudo yum install http://mirror.centos.org/centos/7/extras/x86_64/Packages/pigz-2.3.3-1.el7.centos.x86_64.rpm
```

A message appears, similar to the following:

```
Total size: 123 k   Installed size: 123k   Is this ok [y/d/N]:
```

Enter y to continue.

Next Steps

After you have installed and configured Docker and Kubernetes, proceed to [Installing Genesys CX Insights](#).

After completing the steps on this page, complete the following:

- [Installing Genesys CX Insights](#)
- [Installing report editing software](#)
- [Post-Installation steps](#)