|  |  |
| --- | --- |
| **Reference** |  |
|  | [**https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-compose-on-centos-7**](https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-compose-on-centos-7) |
|  |  |
|  |  |

[**https://community.atlassian.com/t5/Bamboo-questions/Cannot-connect-to-the-Docker-daemon/qaq-p/310527**](https://community.atlassian.com/t5/Bamboo-questions/Cannot-connect-to-the-Docker-daemon/qaq-p/310527)

**Updating Your Instance**

When new EL OS updates are available, you can simply use Yum to update all of the packages:

$ sudo yum update

Major version upgrades will require users to deploy an instance of the new version and migrate their configuration and data to the new instance. No in-place major version upgrade is expected to be supported.

**Install Docker Compose (ref: https://docs.docker.com/compose/install/#upgrading)**

sudo curl -L "https://github.com/docker/compose/releases/download/1.22.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

sudo chmod +x /usr/local/bin/docker-compose

Please make sure –

sudo groupadd docker

sudo gpasswd -a $USER docker

sudo usermod -aG docker $USER

sudo service docker status

Redirecting to /bin/systemctl status docker.service

● docker.service - Docker Application Container Engine

Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)

Active: inactive (dead)

Docs: <https://docs.docker.com>

**systemctl show --property ActiveState docker**

**ActiveState=inactive**

**Step 1 — Installing Docker**

First, install Docker if you haven't already. The quickest way to install Docker is to download and install their installation script (you'll be prompted for a sudo password).

wget -qO- https://get.docker.com/ | sh

The above command downloads and executes a small installation script written by the Docker team. If you don't trust third party scripts or want more details about what the script is doing check out the instructions in the [DigitalOcean Docker tutorial](https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-getting-started) or Docker's own [installation documentation](https://docs.docker.com/engine/installation/centos/).

Working with Docker is a pain if your user is not configured correctly, so add your user to the dockergroup with the following command.

**sudo** usermod -aG docker $(whoami)

sudo reboot

Log out and log in from your server to activate your new groups.

Set Docker to start automatically at boot time:

sudo systemctl enable docker.service

Finally, start the Docker service:

sudo systemctl start docker.service

Now, docker should be up and running.

**Step 2 — Installing Docker Compose**

Now that you have Docker installed, let's go ahead and install Docker Compose. First, install python-pipas prerequisite:

sudo yum install epel-release

sudo yum install -y python-pip

Then you can install Docker Compose:

sudo pip install docker-compose

You will also need to upgrade your Python packages on CentOS 7 to get docker-compose to run successfully:

sudo yum upgrade python\*

Some Sanity checks

ls -l /var/run/docker.sock

srw-rw---- 1 root docker 0 Sep 12 13:11 /var/run/docker.sock

**Examples-**

docker run rickfast/hello-oreilly

Hello O'Reilly!

## Linux (<https://stackoverflow.com/questions/21871479/docker-cant-connect-to-docker-daemon> )

The [Post-installation steps for Linux](https://docs.docker.com/install/linux/linux-postinstall/) documentation reveals the following steps:

1. Create the **docker** group.

sudo groupadd docker

1. Add the user to the **docker** group.

sudo usermod -aG docker $(whoami)

1. Log out and log back in to ensure docker runs with correct permissions.
2. Start docker.

sudo service docker start

30th-Nov-18

[dc-user@ech-10-157-136-3 ~]$ docker info

Containers: 5

Running: 1

Paused: 0

Stopped: 4

Images: 6

Server Version: 1.12.3

Storage Driver: devicemapper

Pool Name: docker-251:0-70087976-pool

Pool Blocksize: 65.54 kB

Base Device Size: 10.74 GB

Backing Filesystem: xfs

Data file: /dev/loop0

Metadata file: /dev/loop1

Data Space Used: 455.4 MB

Data Space Total: 107.4 GB

Data Space Available: 57.74 GB

Metadata Space Used: 1.315 MB

Metadata Space Total: 2.147 GB

Metadata Space Available: 2.146 GB

Thin Pool Minimum Free Space: 10.74 GB

Udev Sync Supported: true

Deferred Removal Enabled: false

Deferred Deletion Enabled: false

Deferred Deleted Device Count: 0

Data loop file: /var/lib/docker/devicemapper/devicemapper/data

WARNING: Usage of loopback devices is strongly discouraged for production use. Use `--storage-opt dm.thinpooldev` to specify a custom block storage device.

Metadata loop file: /var/lib/docker/devicemapper/devicemapper/metadata

Library Version: 1.02.135-RHEL7 (2016-09-28)

Logging Driver: json-file

Cgroup Driver: cgroupfs

Plugins:

Volume: local

Network: host null bridge overlay

Swarm: inactive

Runtimes: runc

Default Runtime: runc

Security Options: seccomp

Kernel Version: 4.1.12-61.1.18.el7uek.x86\_64

Operating System: Oracle Linux Server 7.3

OSType: linux

Architecture: x86\_64

CPUs: 2

Total Memory: 7.545 GiB

Name: ech-10-157-136-3

ID: R5W3:BPSK:7LWA:W534:BW3T:MYJN:ACFP:SEIW:KS5R:PXOU:6EHV:IOD3

Docker Root Dir: /var/lib/docker

Debug Mode (client): false

Debug Mode (server): false

Registry: https://index.docker.io/v1/

Insecure Registries:

127.0.0.0/8

[dc-user@ech-10-157-136-3 ~]$

**Install Java**

cd /opt/

sudo wget --no-cookies --no-check-certificate --header "Cookie: gpw\_e24=http%3A%2F%2Fwww.oracle.com%2F; oraclelicense=accept-securebackup-cookie" <http://download.oracle.com/otn-pub/java/jdk/8u181-b13/96a7b8442fe848ef90c96a2fad6ed6d1/jdk-8u181-linux-x64.tar.gz>

sudo tar xzf jdk-8u181-linux-x64.tar.gz

cd /opt/jdk1.8.0\_181/

sudo alternatives --install /usr/bin/java java /opt/jdk1.8.0\_181/bin/java 2

sudo alternatives --config java

sudo alternatives --install /usr/bin/jar jar /opt/jdk1.8.0\_181/bin/jar 2

sudo alternatives --install /usr/bin/javac javac /opt/jdk1.8.0\_181/bin/javac 2

sudo alternatives --set jar /opt/jdk1.8.0\_181/bin/jar

sudo alternatives --set javac /opt/jdk1.8.0\_181/bin/javac

Install Apache Maven

cd /usr/local

sudo wget <http://www-eu.apache.org/dist/maven/maven-3/3.5.4/binaries/apache-maven-3.5.4-bin.tar.gz>

sudo vi /etc/profile.d/maven.sh

export M2\_HOME=/usr/local/maven

export PATH=${M2\_HOME}/bin:${PATH}

**Problem(Abstract)**

sshd start failed cause by "start sshd:/var/empty/sshd must be owned by root and not gruop or world-writable"

**Cause**

file dir /var/empty/sshd 's access permissions not been setup correct

**Resolving the problem**

Follow below instruction to do some change for access permissions of /var/empty/sshd

1) su - root  
  
2) chmod 755 /var/empty/sshd  
  
3) service sshd restart

docker swarm init --advertise-addr 10.157.136.3:2377 --listen-addr 10.157.136.3:2377

Swarm initialized: current node (8a0c04r09h5uf1kp7oxlnh31s) is now a manager.

To add a worker to this swarm, run the following command:

docker swarm join \

--token SWMTKN-1-6bq1kdfqkbmydkp33l27s9dc4awbtjtbg8ctsrph0jfeq5o5la-3b1c1ea8vm269talim6lgsu7f \

10.157.136.3:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

[dc-user@ech-10-157-136-3 ~]$ docker swarm join-token manager

To add a manager to this swarm, run the following command:

docker swarm join \

--token SWMTKN-1-6bq1kdfqkbmydkp33l27s9dc4awbtjtbg8ctsrph0jfeq5o5la-1xbywis55eqhyvnkvwqb8jd1l \

10.157.136.3:2377

[dc-user@ech-10-157-136-3 ~]$ docker swarm join-token worker

To add a worker to this swarm, run the following command:

docker swarm join \

--token SWMTKN-1-6bq1kdfqkbmydkp33l27s9dc4awbtjtbg8ctsrph0jfeq5o5la-3b1c1ea8vm269talim6lgsu7f \

10.157.136.3:2377

Now, grab the manager whole token and append below details of advertise-addr and --listen-addr.

docker swarm join --token SWMTKN-1-6bq1kdfqkbmydkp33l27s9dc4awbtjtbg8ctsrph0jfeq5o5la-1xbywis55eqhyvnkvwqb8jd1l 10.157.136.3:2377 --advertise-addr 10.157.136.96:2377 --listen-addr 10.157.136.96:2377

This node joined a swarm as a manager.

$ docker node ls

ID HOSTNAME STATUS AVAILABILITY MANAGER STATUS ENGINE VERSION

8a0c04r09h5uf1kp7oxlnh31s ech-10-157-136-3 Ready Active Leader 17.05.0-ce

jjgng5s4cbgbygm487gjuiig3 \* ech-10-157-136-96 Ready Active Reachable 18.09.1