

# Lab 1 – Installing and Configuring Docker

## Introduction of Docker

Docker is an open platform for **developing**, **shipping**, and **running** applications. Docker is designed to deliver user applications faster. With Docker user can separate applications from one infrastructure and treat as infrastructure like a managed application. Docker helps user to ship code faster, test faster, deploy faster, and shorten the cycle between writing code and running code.

In this lab you will **install**, **configure** and **enable** the docker daemon on one of the host **aio110** and pull a simple **centos** image to get familiar with docker commands. Create a couple of containers based on this image and launch long-running processes inside each. This introduces docker as well as process isolation, one of the most important features of containers.

## 1. Environment Setup

**Caution!** Before starting this lab, and any time you are returning to start working on your lab exercises, make sure that you are logged in as the correct user. The default user- **user0** or **aio110** – will show up unless you have selected the appropriate user on the **Class IP Assignment** page. If you are not **user0** but you see **aio110** in the command below **step 1** or in the user details at the top of this page, take a moment to pick your correct user number on the right side of **Class IP Assignment** page.

### 1.1 Login as “root” user on **aio110** host:

Copy

```
ssh root@aio110
```

### 1.2 Make sure correct hostname is set:

Copy

```
hostnamectl set-hostname aio110
```

Copy

```
hostname
```

## Output:

```
aio110
```

**Note:** Hostname in the prompt, will change only after **reboot** and we will be reboot the host in a movement.

**1.3** Add entries to “**/etc/hosts**” file for local **hostname** resolution.

Copy

```
cat > /etc/hosts <<EOF

172.16.120.12 pod0-master.origin.com

172.16.120.22 pod0-node1.origin.com

10.1.64.1 gw.onecloud    gw

EOF
```

**1.4** Update the system with the latest packages and **reboot** the host to take effect:

Copy

```
yum update -y
```

**1.5** Delete the **set\_hostname** and **update\_hostname** attribute in “**/etc/cloud/cloud.cfg**” file.

Copy

```
sed -i '/set_hostname/d' /etc/cloud/cloud.cfg

sed -i '/update_hostname/d' /etc/cloud/cloud.cfg
```

**Note:** Above command is used to avoid appending default domain name (“**.novalocal**”) to the hostname by cloud\_config service as the servers used for this class are actually openstack instances .

**1.6** Reboot

Copy

```
reboot
```

**Note:** Login back as “**root**” user on **aio110**.  
`ssh root@aio110`

## 2. Pre-requisites

**2.1** Configure **EPEL** repository by installing the below package which provides additional packages for Docker.

**EPEL** (Extra Packages for Enterprise Linux) is open source and free community based repository project from Fedora team which provides 100% high quality add-on software packages for Linux distribution including **RHEL** (Red Hat Enterprise Linux), **CentOS**, and Scientific **Linux**.

EPEL project is not a part of **RHEL/CentOS** but it is designed for major Linux distributions by providing lots of open source packages like networking, sys admin, programming, monitoring and so on. Most of the epel packages are maintained by Fedora repo.

Copy

```
yum install epel-release -y
```

**2.2** Set up docker yum repository:

Copy

```
cat > /etc/yum.repos.d/docker.repo <<EOF

[dockerrepo]

name=Docker Repository

baseurl=https://yum.dockerproject.org/repo/main/centos/7/

enabled=1

gpgcheck=1

gpgkey=https://yum.dockerproject.org/gpg
```

EOF

## 2.3 Make sure SELinux is disabled:

**Security-Enhanced Linux (SELinux)** is a **mandatory access control (MAC)** security mechanism implemented in the kernel. SELinux was first introduced in **CentOS 4** and significantly enhanced in later CentOS releases.

**SELinux** has **three basic modes** of operation, of which **Enforcing** is set as the installation default mode.

1. **Enforcing** : The default mode which will enable and enforce the SELinux security policy on the system, denying access and logging actions
2. **Permissive** : In Permissive mode, SELinux is enabled but will not enforce the security policy, only warn and log actions. Permissive mode is useful for troubleshooting SELinux issues.
3. **Disabled** : SELinux is turned off

**Note:** In our environment “**firewalld**” and “**SELinux**” is **disabled** by default.

Copy

```
sestatus
```

**Output:**

```
SELinux status:                disabled
```

## 2.4: Install **Chrony** and update the “**Chrony**” (**NTP Server**) configuration to allow connections from our other nodes.

**chrony** is a versatile implementation of the **Network Time Protocol (NTP)**. It can synchronise the system clock with NTP servers, reference clocks (e.g. GPS receiver), and manual input using wristwatch and keyboard.

Copy

```
yum install chrony -y
```

Copy

```
sed -e '/server/s/^/#/g' -i /etc/chrony.conf
```

```
cat >> /etc/chrony.conf <<EOF  
  
server gw.onecloud iburst  
  
Allow 10.1.64.0/24  
  
Allow 10.1.65.0/24  
  
EOF
```

**2.5** Next enable, restart and check the status of Chrony service.

Copy

```
systemctl enable chronyd.service  
  
systemctl restart chronyd.service  
  
systemctl status chronyd.service
```

**2.6** Run this command to verify the NTP sources:

Copy

```
chronyc sources
```

**Sample output:**

```
210 Number of sources = 1  
  
MS Name/IP address          Stratum Poll Reach LastRx Last sample  
  
=====
```

MS Name/IP address	Stratum	Poll	Reach	LastRx	Last sample
^? gw.onecloud	3	6	3	0	-1291us[-1291us]
+/- 90ms					

```
=====
```

**2.7** To check current kernel version, run the below command :

Docker requires a **64-bit** installation regardless of user CentOS version. Also, kernel must be **3.10** at minimum, which CentOS 7 runs.

Copy

```
uname -r
```

**Output:**

```
3.10.0-693.17.1.el7.x86_64
```

Finally, it is recommended that fully update system. Please keep in mind that system should be fully patched to fix any potential kernel bugs. Any reported kernel bugs may have already been fixed on the latest kernel packages.

## 3. Install Docker

**3.1** Install the Docker package.

Copy

```
yum install docker-engine -y
```

Once docker is installed, will need to start the docker daemon.

**3.2** Run the below command to start the docker daemon on boot.

Copy

```
systemctl enable docker
```

**Output:**

```
Created symlink from /etc/systemd/system/multi-  
user.target.wants/docker.service to  
/usr/lib/systemd/system/docker.service.
```

**3.3** Start the Docker daemon.

Copy

```
systemctl start docker
```

### 3.4 Verify the Status of the Docker.

Copy

```
systemctl status docker
```

#### Output:

- docker.service - Docker Application Container Engine

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

Active: active (running) since Sat 2018-02-17 11:40:44 UTC; 2s ago

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

Main PID: 10508 (dockerd)

Memory: 14.4M

CGroup: /system.slice/docker.service

└─10508 /usr/bin/dockerd

└─10514 docker-containerd -l  
unix:///var/run/docker/libcontaine...

Feb 17 11:40:41 aio110 systemd[1]: Starting Docker Application  
Container.....

Feb 17 11:40:41 aio110 dockerd[10508]: time="2018-02-  
17T11:40:41.92854899..."

....

....

```
Feb 17 11:40:44 aio110 dockerd[10508]: time="2018-02-17T11:40:44.00961340..."
```

Hint: Some lines were ellipsized, use `-l` to show in full.

## 4. Verify the Docker Installation

**4.1** Verify Docker is installed correctly by running a test image in a container.

Copy

```
docker run hello-world
```

**Output:**

```
Hello from Docker!
```

```
This message shows that your installation appears to be working correctly.
```

**4.2** The “**docker ps**” command only shows running containers by default. To see all containers, use the `-a`.

Copy

```
docker ps -a
```

**Output:**

CONTAINER ID	IMAGE	COMMAND	CREATED
STATUS	PORTS	NAMES	
eb20c240f350	hello-world	"/hello"	47 seconds ago
Exited (0) 46 seconds ago			
blissful_jennings			

**4.3** As you noticed in the previous steps, Docker was “**Unable to find image ‘hello-world:latest’ locally**” and it fetched it from one of **Public registry** and saved it **locally**.



List all the available Docker images stored in locally.

Copy

```
docker images
```

**Output:**

REPOSITORY SIZE	TAG	IMAGE ID	CREATED
hello-world ago 1.85kB	latest	f2a91732366c	2 months

**4.4** To search for a Docker image, **centos** for instance.

Copy

```
docker search centos
```

**Output:**

NAME STARS	OFFICIAL	AUTOMATED	DESCRIPTION
centos 4038	[OK]		The official build of CentOS.
ansible/centos7-ansible 105		[OK]	Ansible on Centos7
jdeathe/centos-ssh 7.4.1708 x8...	90		CentOS-6 6.9 x86_64 / CentOS-7 [OK]
consol/centos-xfce-vnc VNC sessi...	43		Centos container with "headless" [OK]
....			
....			

```
pivotaldata/centos
with a ... 0
```

Base centos, freshened up a little

```
smartentry/centos
0
```

[OK]

centos with smartentry

**4.5** Download it locally by running the below command (in this case centos image is downloaded and used).

Copy

```
docker pull centos
```

**Output:**

```
Using default tag: latest
```

```
latest: Pulling from library/centos
```

```
af4b0a2388c6: Pull complete
```

```
Digest:
```

```
sha256:2671f7a3eea36ce43609e9fe7435ade83094291055f1c96d9d1d1d7c0b986a5d
```

```
Status: Downloaded newer image for centos:latest
```

**4.6** Check Docker image on host.

Copy

```
docker images
```

**Output:**

REPOSITORY SIZE	TAG	IMAGE ID	CREATED
centos ago	latest 207MB	ff426288ea90	5 weeks

hello-world ago	latest 1.85kB	f2a91732366c	2 months
--------------------	------------------	--------------	----------

#### 4.7 Run an interactive session into a container.

Copy

```
docker run -dit centos
```

#### Sample Output:

```
89d2c6c2b29e201d13f8b89e8334c502b022113f24089946447c51853bf9a653
```

**Note:** “dit” – daemon interactive terminal (to run in active state).

#### 4.8 Check all the running containers.

Copy

```
docker ps -a
```

#### Output:

CONTAINER ID STATUS	IMAGE PORTS	COMMAND NAMES	CREATED
89d2c6c2b29e ago Up 22 seconds heuristic_bhaskara	centos	"/bin/bash"	23 seconds
eb20c240f350 ago Exited (0) 5 minutes ago blissful_jennings	hello-world	"/hello"	5 minutes

## 5. Cleanup

#### 5.1 To remove all the containers run the below commands:

Copy

```
docker rm `docker ps -a -q` -f
```

### Sample Output:

```
89d2c6c2b29e
```

```
eb20c240f350
```

**5.2** To remove all the images run the below commands:

Copy

```
docker rmi `docker images -q` -f
```

### Sample Output:

```
Untagged: centos:latest
```

```
Untagged:
```

```
centos@sha256:2671f7a3eea36ce43609e9fe7435ade83094291055f1c96d9d1d1d7c  
0b986a5d
```

```
Deleted:
```

```
sha256:ff426288ea903fcf8d91aca97460c613348f7a27195606b45f19ae91776ca23  
d
```

```
Deleted:
```

```
sha256:e15afa4858b655f8a5da4c4a41e05b908229f6fab8543434db79207478511ff  
7
```

```
Untagged: hello-world:latest
```

```
Untagged: hello-
```

```
world@sha256:083de497cff944f969d8499ab94f07134c50bcf5e6b9559b27182d3fa  
80ce3f7
```

```
Deleted:
```

```
sha256:f2a91732366c0332ccd7afd2a5c4ff2b9af81f549370f7a19acd460f87686bc  
7
```

Deleted:  
sha256:f999ae22f308fea973e5a25b57699b5daf6b0f1150ac2a5c2ea9d7fecee50fd  
f

### 5.3 Verify that containers are removed:

Copy

```
docker ps
```

**Output:**

CONTAINER ID STATUS	IMAGE PORTS	COMMAND NAMES	CREATED
------------------------	----------------	------------------	---------

### 5.4 Verify that docker images are removed:

Copy

```
docker images
```

**Output:**

REPOSITORY SIZE	TAG	IMAGE ID	CREATED
--------------------	-----	----------	---------

### 5.5 Run the below command for more information on a command:

Information available like **options** and **Management commands**.

Copy

```
docker --help
```