Docker and Kubernetes

# Task 1 – Install docker

Get the script file to install docker and its dependencies from docker website and pipe it to the shell.

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

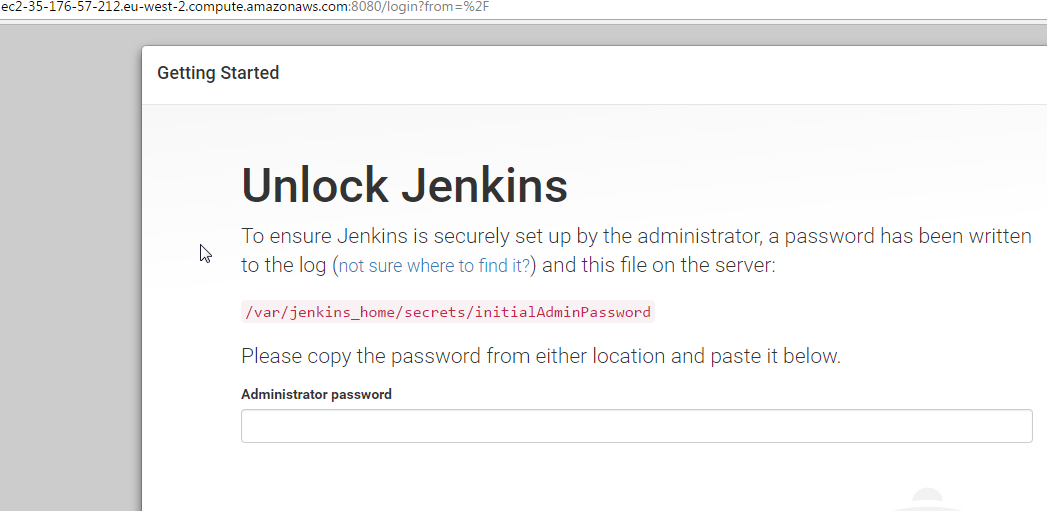
# Task 2 – Deploy a Jenkins Container

Pull a Jenkins image

**docker pull Jenkins**

Run Jenkins in a container

**docker run -p 8080:8080 -p 50000:50000 jenkins**



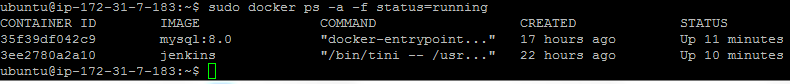
A container of Jenkins has been created, use ctrl+z to get out of the terminal.

**sudo docker ps –a**

**Lists all processes, find the container id of Jenkins**

**sudo docker start containerID**





# Task 3 – Create a dockerfile

Create dockerfile using the following contents:

FROM ubuntu:16.04

#Always update your running system

RUN sudo apt-get update -y

#You may or may not need to run these commands

RUN sudo apt-get install -y wget

RUN sudo apt-get install -y tar

#installs the libraries needed to run the GUI

RUN sudo apt-get install -y libgtk2.0

RUN sudo apt-get install -y mesa-utils

RUN sudo apt-get install -y libXtst6

#RUN sudo apt-get install -y openjdk-7-jre

#RUN java -version

#Now install the Java Compiler

#RUN sudo apt-get install -y openjdk-7-jdk

#RUN javac -version

#Add java from file and install

WORKDIR /opt

ADD files /opt

RUN sudo tar zxvf /opt/java.tar.gz

RUN sudo update-alternatives --install /usr/bin/java java /opt/jdk1.8.0\_74/bin/java 100

RUN sudo update-alternatives --install /usr/bin/javac javac /opt/jdk1.8.0\_74/bin/javac 100

# Install OpenJDK-8

RUN apt-get update && \

apt-get install -y openjdk-8-jdk && \

apt-get install -y ant && \

apt-get clean;

# Fix certificate issues

RUN apt-get update && \

apt-get install ca-certificates-java && \

apt-get clean && \

update-ca-certificates -f;

# Setup JAVA\_HOME -- useful for docker commandline

ENV JAVA\_HOME /usr/lib/jvm/java-8-openjdk-amd64/

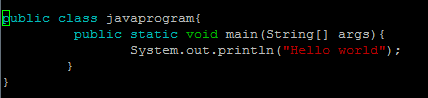
RUN export JAVA\_HOME

Run the file using the following command.

**docker build -t [imagename] .**

*Note: the dot at the end of command must be present; imagename is the name of the image to be run e.g. Ubuntu:16.04*

Create java program



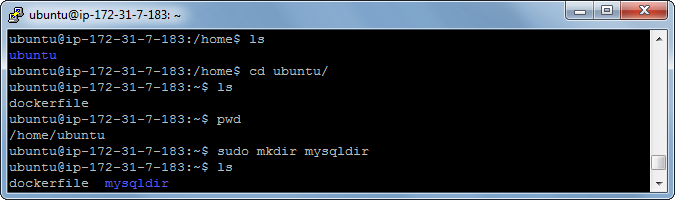
Compile and run it.





## Task 4 – Create your own linked container

To create mysql container that stores data in the host volume, create a directory



**docker run --name some-mysql -v /home/ubuntu/mysqldir:/var/lib/mysql -e MYSQL\_ROOT\_PASSWORD=my-secret-pw -d mysql:tag**

where **my-secret-pw** is password, **tag** is version number e.g. 8.0

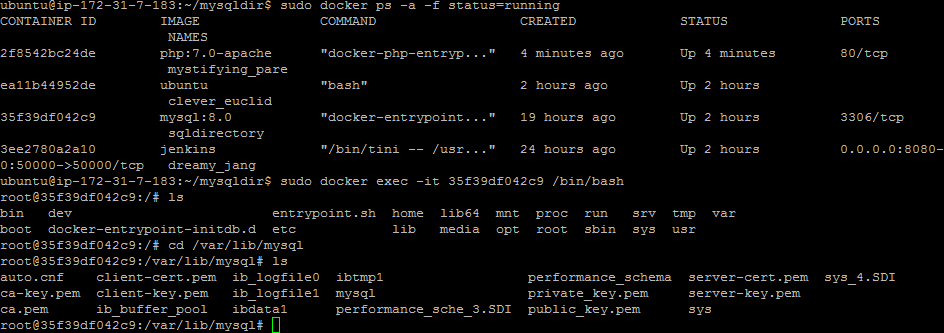
e.g. **docker run --name some-mysql -v /home/ubuntu/mysqldir:/var/lib/mysql -e MYSQL\_ROOT\_PASSWORD=abcde -d mysql:8.0**

Find running process with

**sudo docker ps -a -f status=running**

Execute interactive terminal –it inside a container using exec and containerID

**sudo docker exec -it containerID /bin/bash**

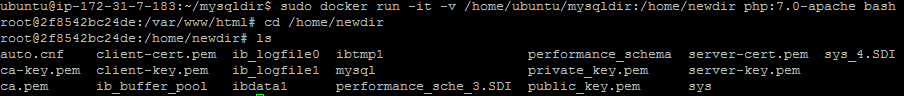
****

For PHP project, pull php image:

**sudo docker pull php**

When running the container mount the directory to a newdirectory in php file path

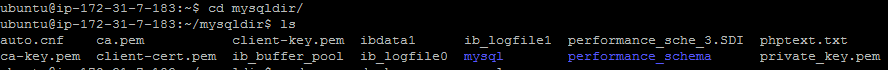
**sudo docker run -it -v /home/ubuntu/mysqldir:/home/newdir php:7.0-apache bash**

****

## Task 5 - Create your own docker-compose file

Create a directory to be used as shared folder.

/home/Ubuntu/mysqldir contains many files from task4

****

Create **docker-compose.yml** file as below:

**version : '2'**

**services:**

**db:**

**image: mysql:8.0**

**ports:**

**- "3333:3333"**

**volumes:**

**- /home/ubuntu/mysqldir:/var/www/html**

**environment:**

**MYSQL\_ROOT\_PASSWORD: passwerd**

**php:**

**image: php:7.0-apache**

**links:**

**- db:db**

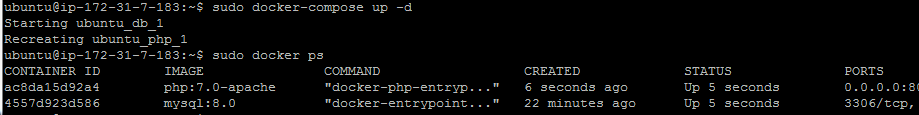
**ports:**

**- "80:80"**

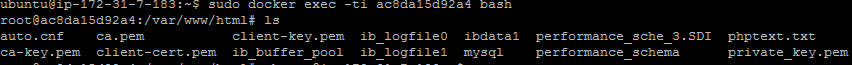
**volumes:**

**- /home/ubuntu/mysqldir:/var/www/html**

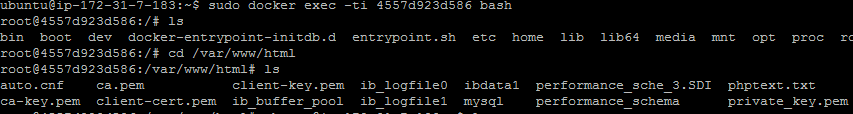
PHP and mysql are running



Execute the first container and check shared directory which contains files as expected.



Execute the second container and it can be seen that the directory is successfully mounted as the files expected are present.



## Task 6 – Install Kubernetes

Install conjure up which also installs juju

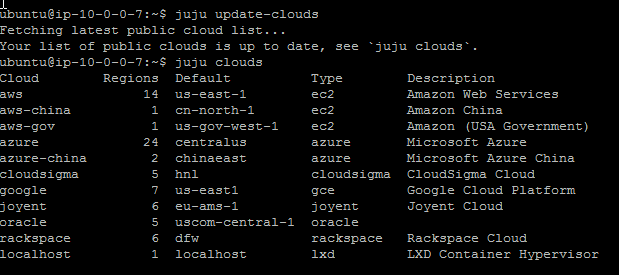
**sudo snap install conjure-up --classic**

Add credentials of aws using juju

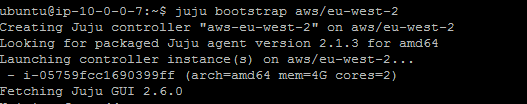
**juju add-credential aws**



Update cloud and check it

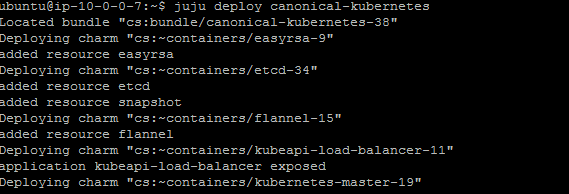


Bootstrap a controller to manage our cluster

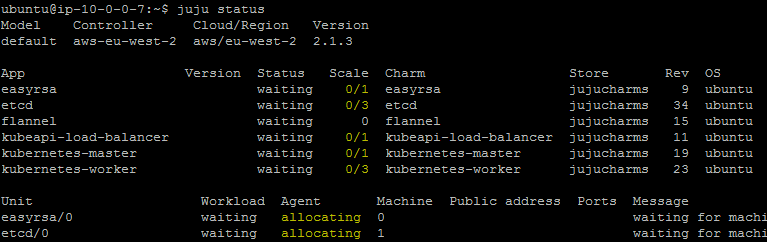


Deploy a cluster of 9 nodes

**juju deploy canonical-kubernetes**



Check the status



## Task 7 – Creating your first Single Container Pod

Install kubectl using curl

**curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl**

Make it executable

**chmod +x ./kubectl**

Move it to environmental PATH

**sudo mv ./kubectl /usr/local/bin/kubectl**