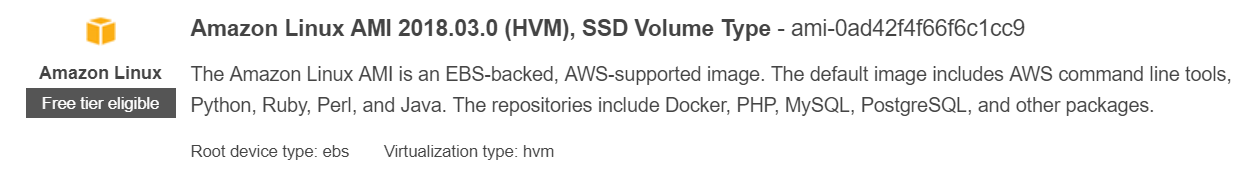
**To install Docker on an Amazon EC2 instance**

1. Create a ec2 instance on AWS select AMI as Redhat Linux



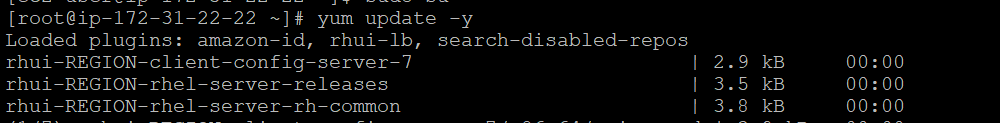
1. Turn from normal user to root user

#sudo su –



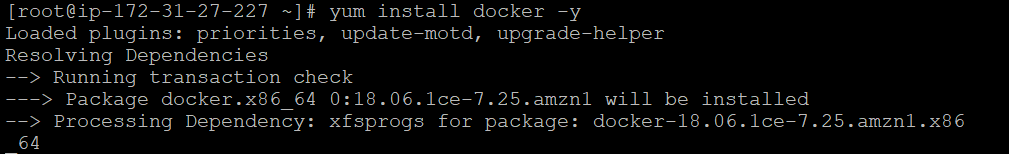
1. Update all the packages

#yum update –y



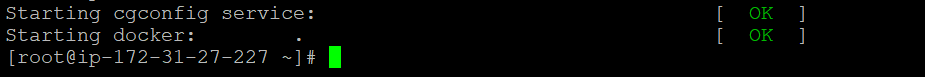
1. Install docker engine

# yum install docker –y



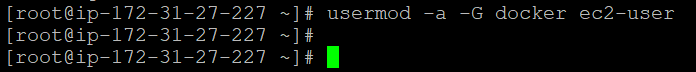
1. Start the docker engine

# service docker start



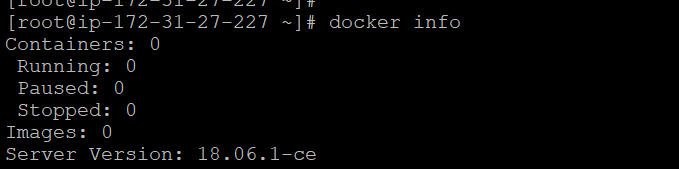
1. Add the ec2-user to the docker group so you can execute Docker commands without using sudo

#usermod -a -G docker ec2-user



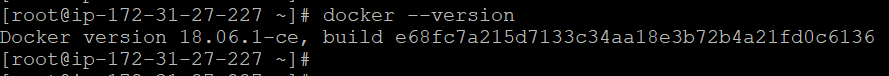
Verify that the ec2-user can run Docker commands without sudo.

1. #docker info



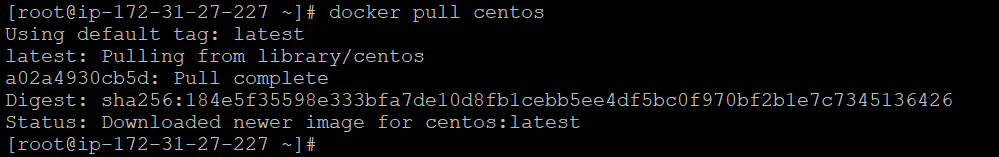
**Docker Basic Command**

1. How to check the version of Docker Installed.

# docker --version

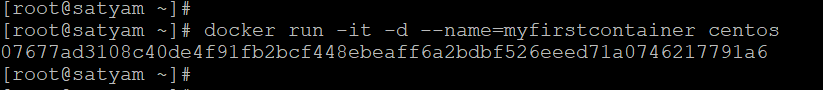
1. How to pull any image.

#docker pull centos



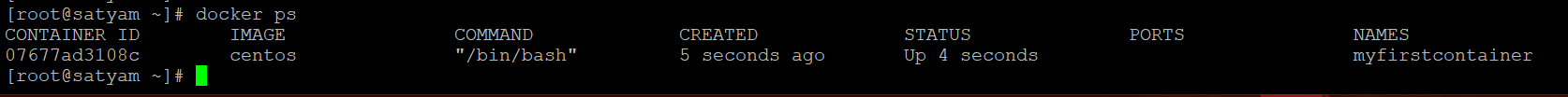
1. How to make container from Image

#docker run –it –d –name=myfirstcontainer centos

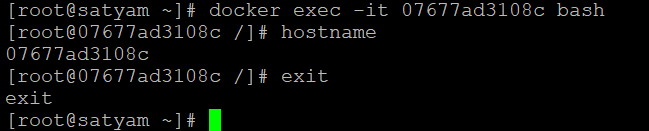


1. How to check running container’s

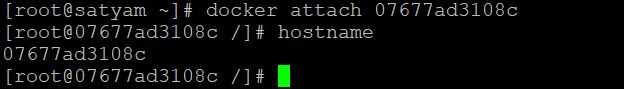
#docker ps



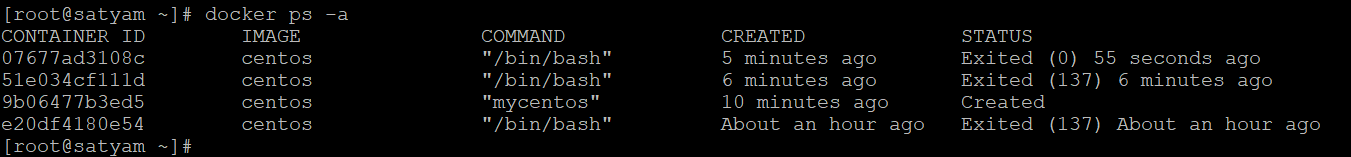
1. How to login inside container



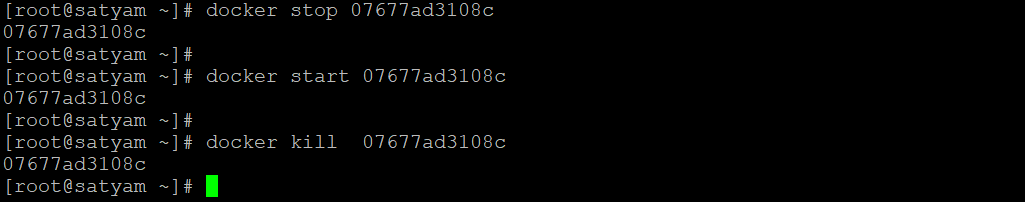
1. How to attach with a running container.



1. How to all docker process running in background.



1. How to stop,start,kill a container.



**Docker Labs**

Creating a Ngnix Docker Container:

1. Create the Docker file 1st

========

#Create a Nginx Container

FROM ubuntu

MAINTAINER demousr@gmail.com

RUN apt-get update \

&& apt-get install -y nginx \

&& apt-get clean \

=========

1. Docker build is the command to build Docker images from Docker file

# docker build -t myngniximage .

Here, myngniximage is the name we are giving to the Image , Since the Docker File is in the present working directory, we used "." at the end of the command to signify the present working directory.



Here is the Docker Image, which you just created:

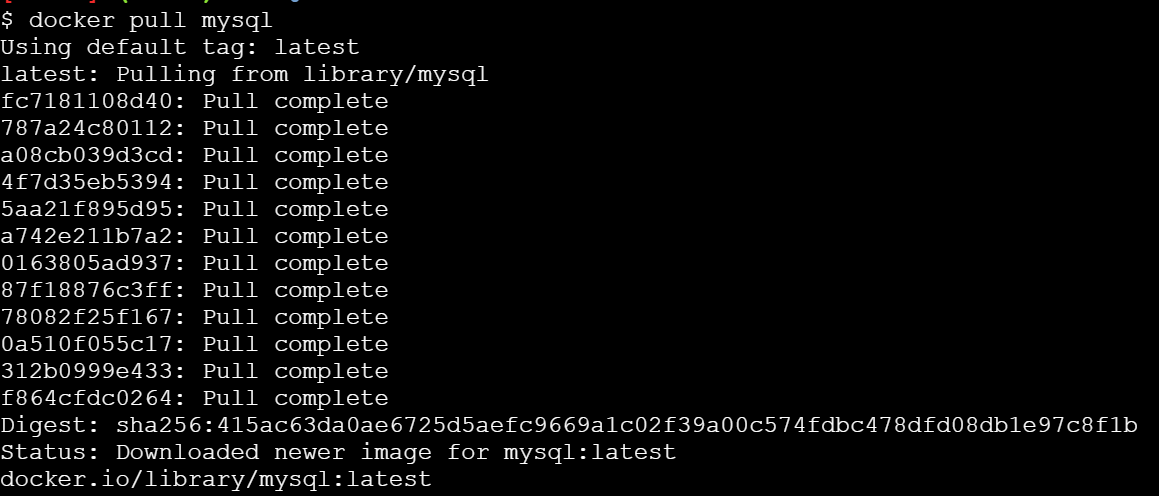


1. Now create the docker container from the image which you have created

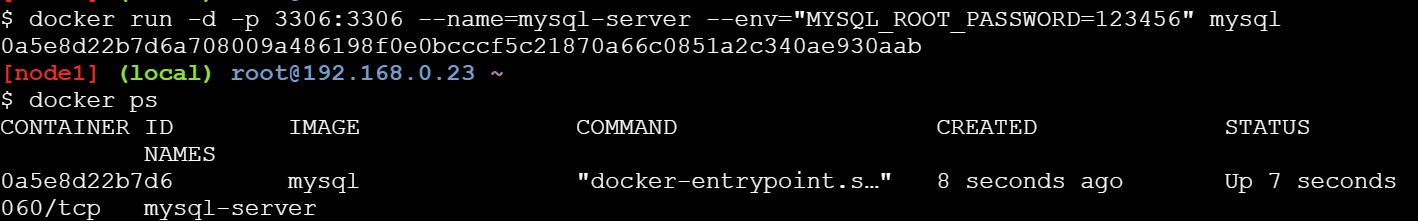


Creating mysql inside docker container

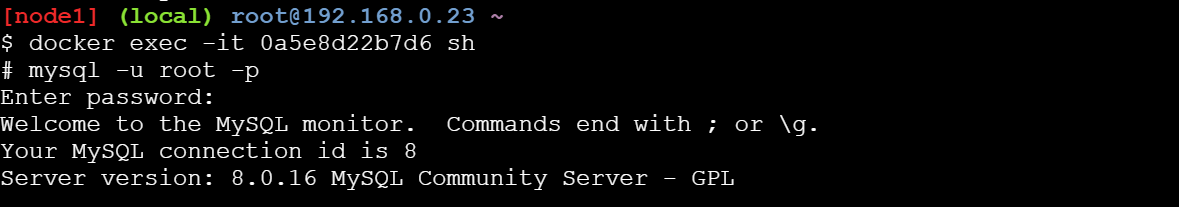
1. Docker pull mysql



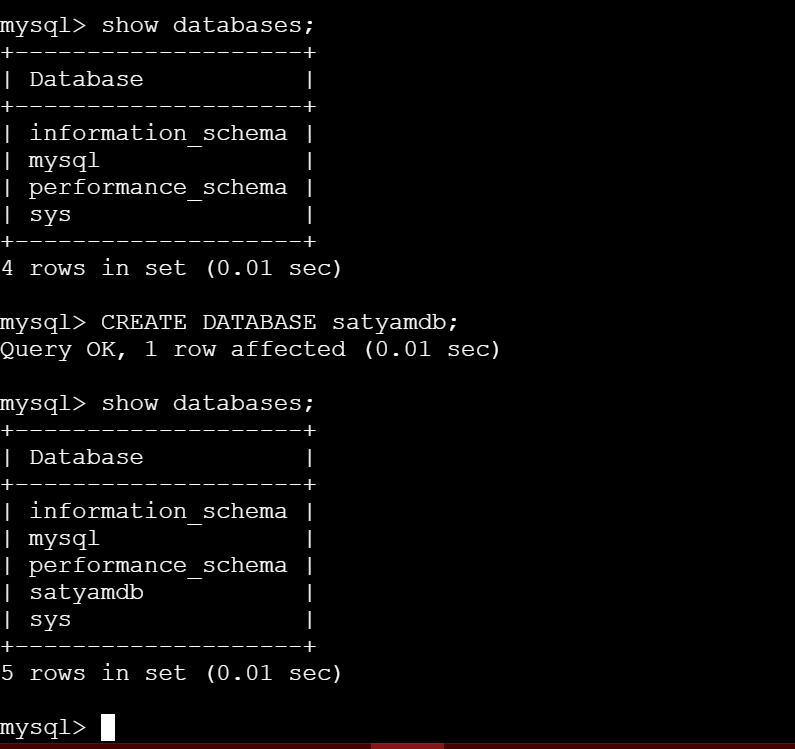
1. Create a container from the pulled mysql image



1. Login to the container and connect to the db.

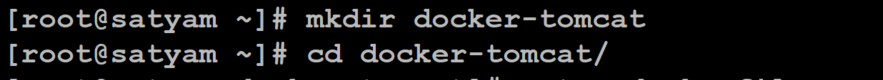


1. Use show DB command to list database and create a DB



Dockerfile for Tomcat

we are going to learn how to install a Tomcat Application Server or Web Container on Docker and Deploy web applications into the Tomcat running inside Docker.  This post is all about Docker Tomcat and deploying war web application into tomcat docker



============

FROM centos

MAINTAINER email2satyam88@gmail.com

RUN mkdir /opt/tomcat/

WORKDIR /opt/tomcat

RUN curl -O https://www-eu.apache.org/dist/tomcat/tomcat-8/v8.5.43/bin/apache-tomcat-8.5.43.tar.gz

RUN tar xvfz apache\*.tar.gz

RUN mv apache-tomcat-8.5.43/\* /opt/tomcat/.

RUN yum -y install java

RUN java -version

WORKDIR /opt/tomcat/webapps

RUN curl -O -L https://github.com/satyam88/SampleWebApp.git

EXPOSE 8080

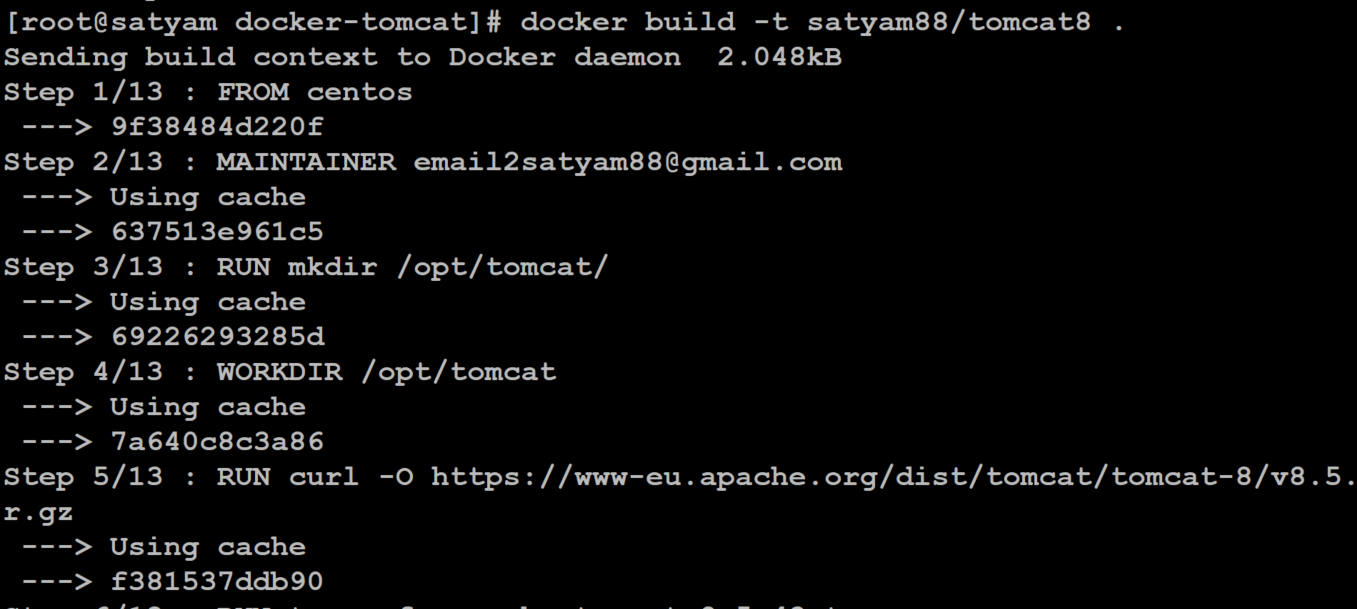
CMD ["/opt/tomcat/bin/catalina.sh", "run"]

============

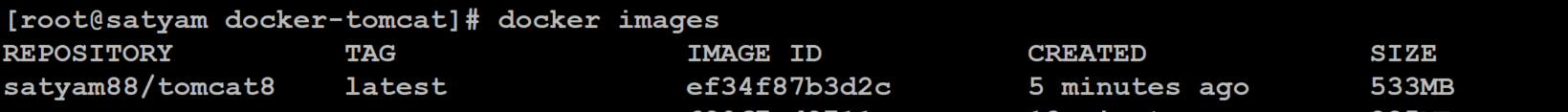
1. Create the dockerfile



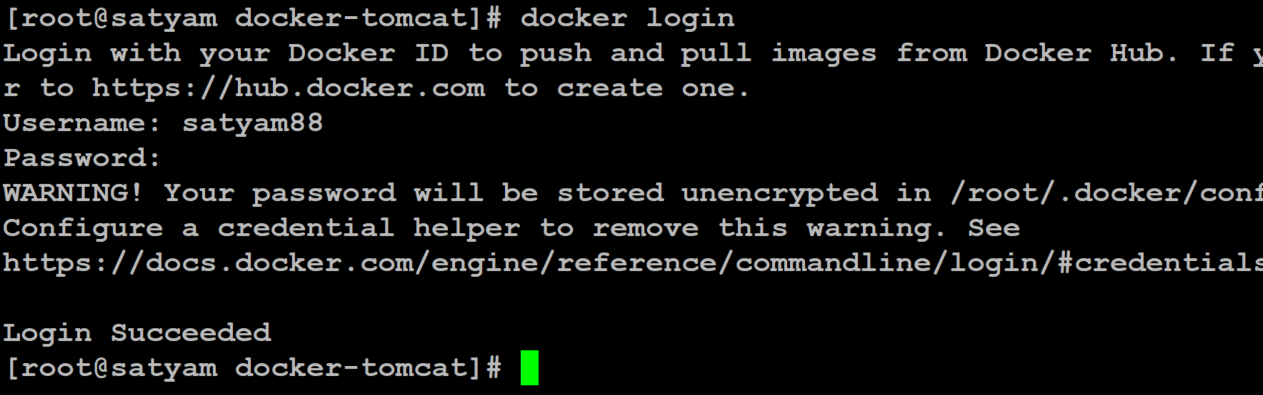
1. Create the docker image from the dockerfile



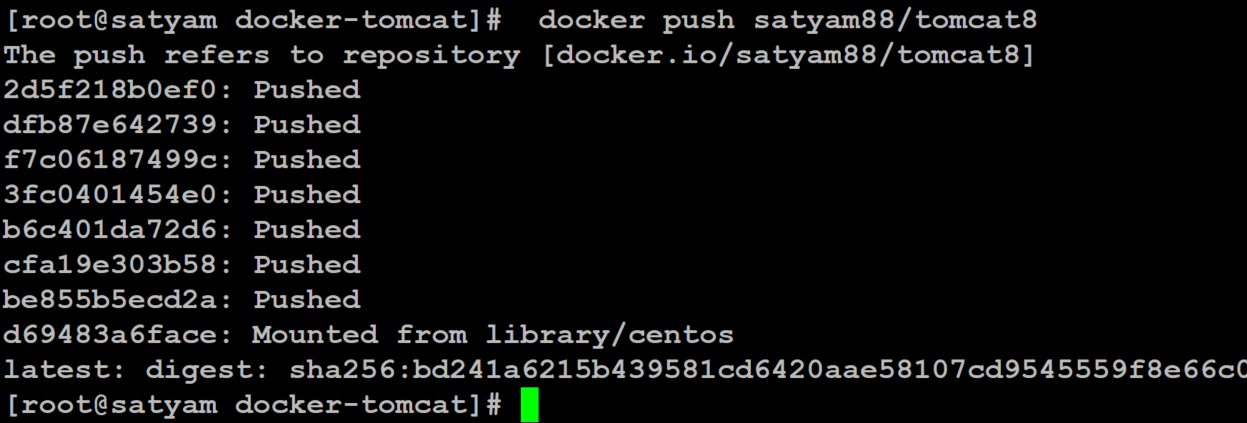
1. List your docker file



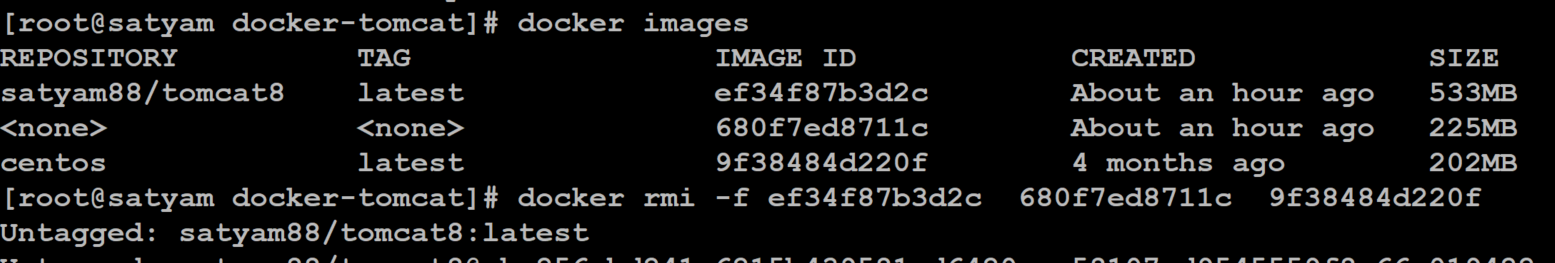
1. Login to the docker hub



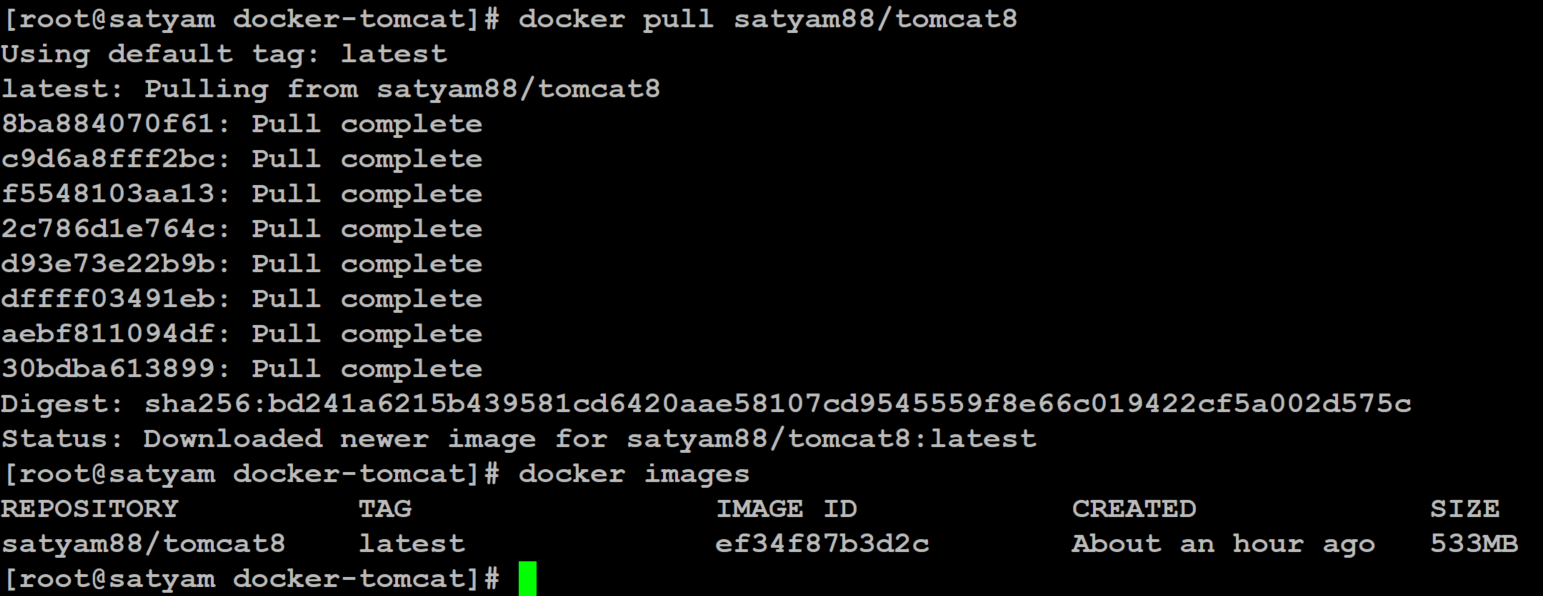
1. Push the docker image to the docker repository



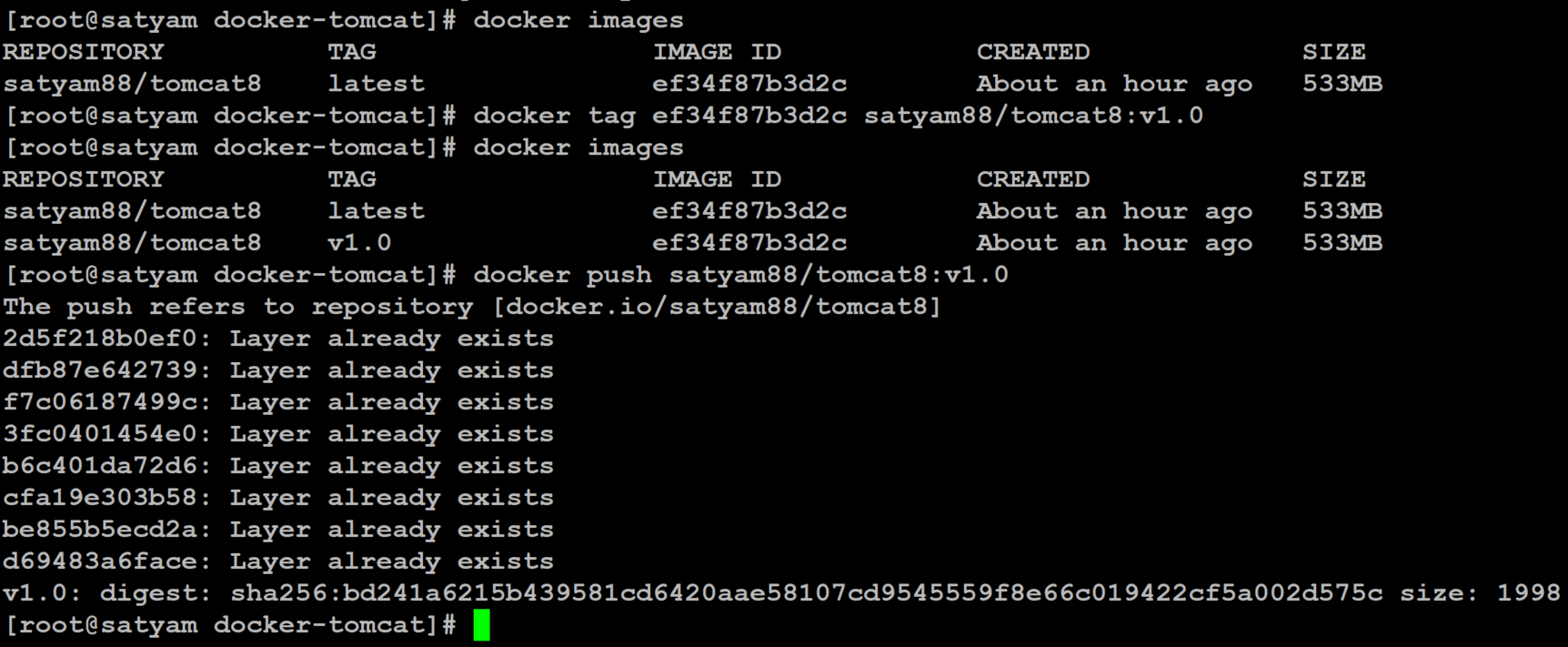
1. Check the docker image and delete the same



1. Pull the same image which has been pushed earlier



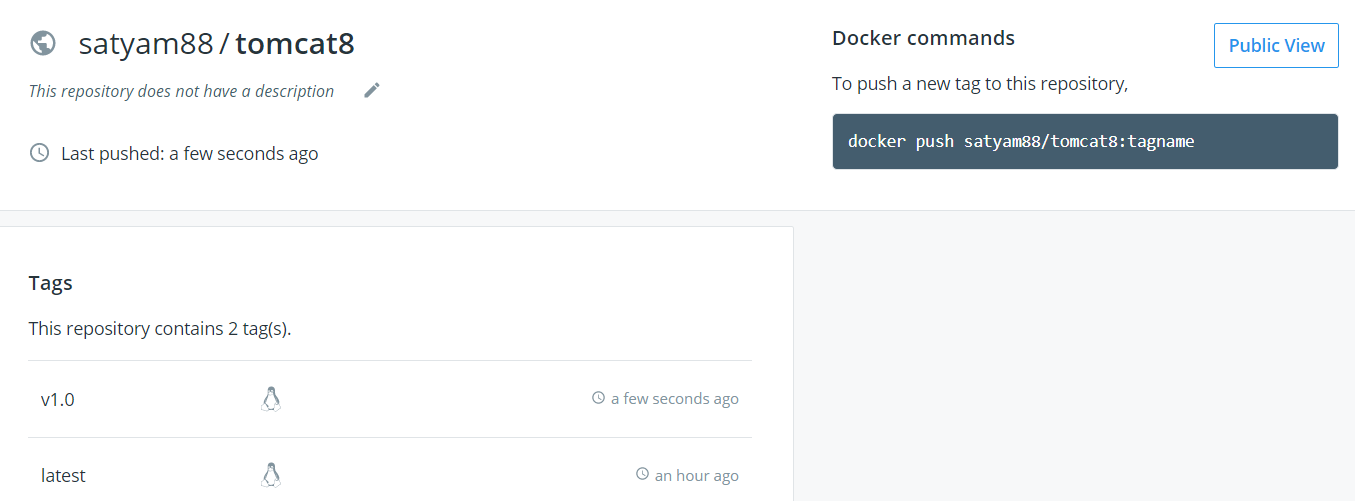
1. This time we are going to tag the docker image and going to push the same.



1. Create the container from docker images , here we are exposing the port



1. Verify the image which has been pushed

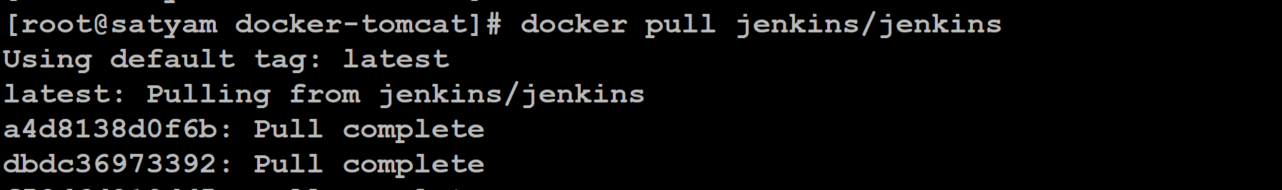


Deploy Jenkins inside Docker Container

====================================

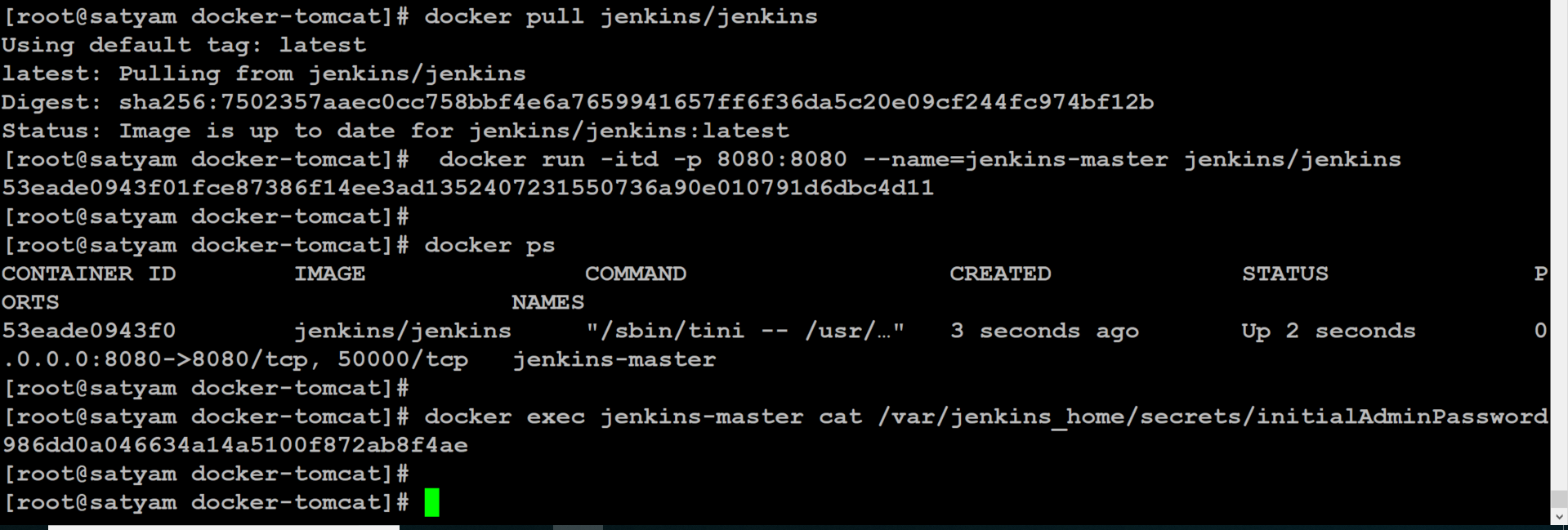
1. Download the image from docker hub

Docker pull Jenkins/jenkins

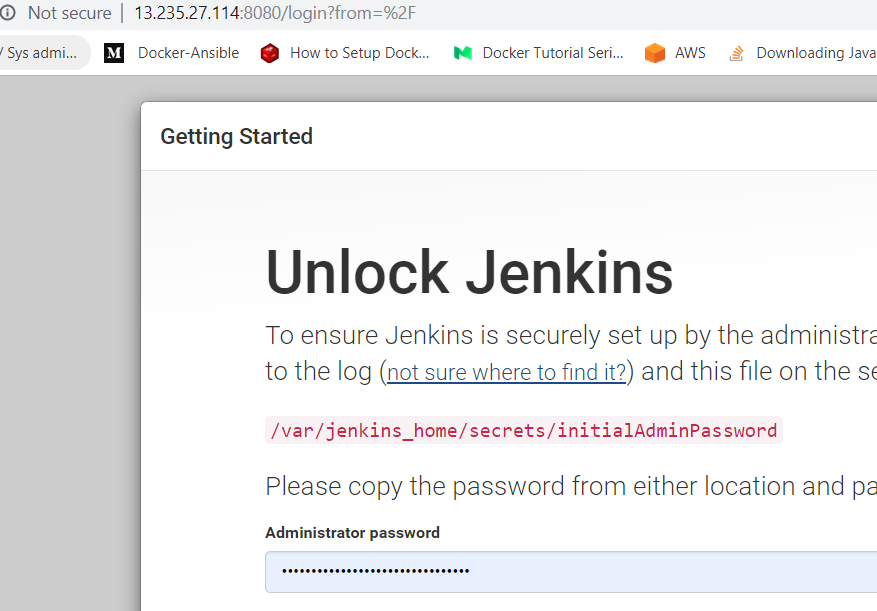


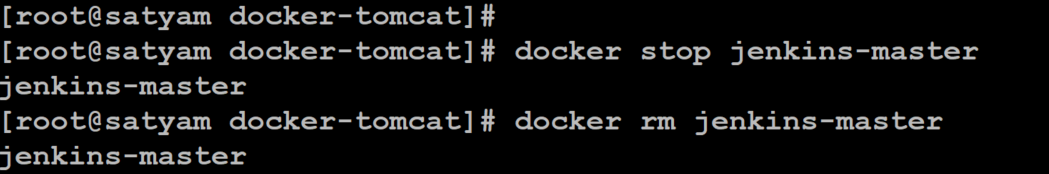
docker run -itd -p 8080:8080 --name=jenkins-master jenkins/jenkins

docker exec jenkins-master cat /var/jenkins\_home/secrets/initialAdminPassword



Use AWS ec2 instance public IP



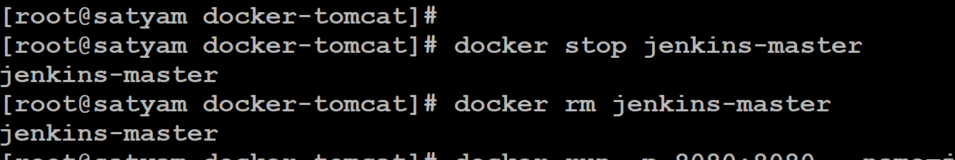


docker run -p 8080:8080 --name=jenkins-master -d --env JAVA\_OPTS="-Xmx512m" jenkins/jenkins



docker exec jenkins-master ps -ef | grep java





Create your own custom docker file for Jenkins

========

FROM jenkins/jenkins:2.112

MAINTAINER email2satyam88@gmail.com

USER root

RUN mkdir /var/log/jenkins

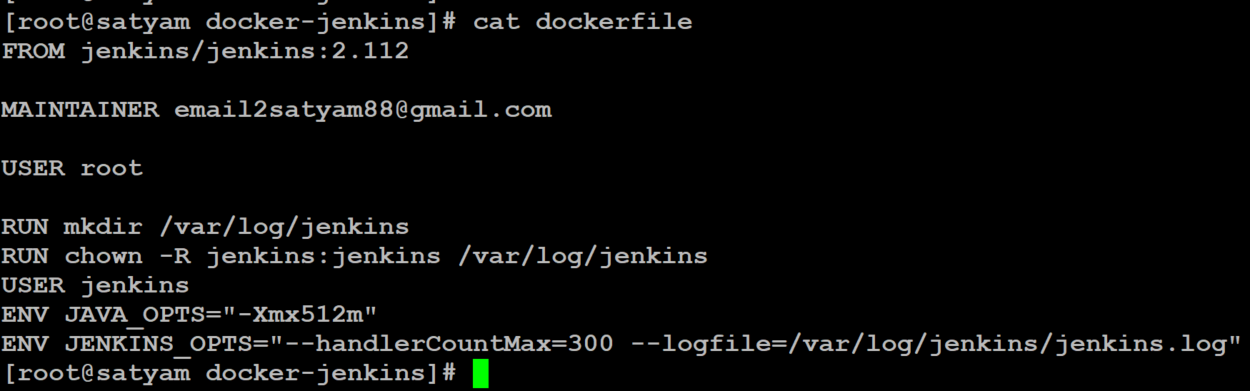
RUN chown -R jenkins:jenkins /var/log/jenkins

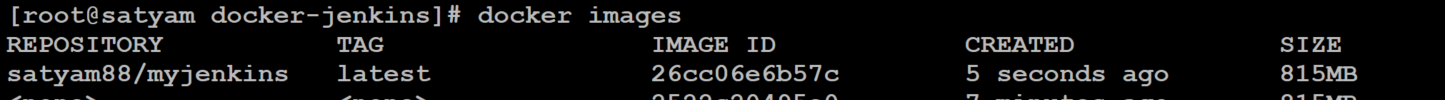
USER jenkins

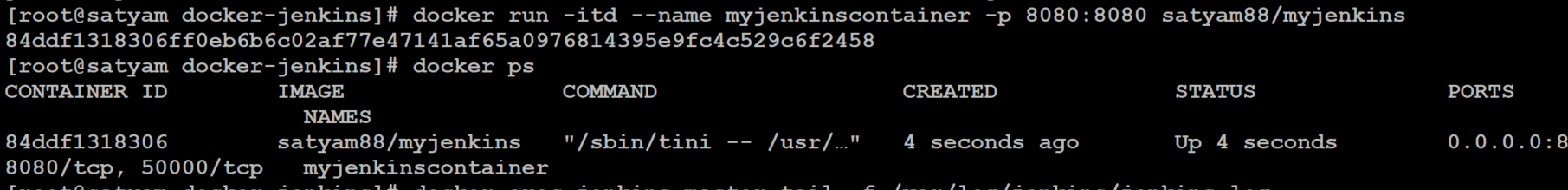
ENV JAVA\_OPTS="-Xmx512m"

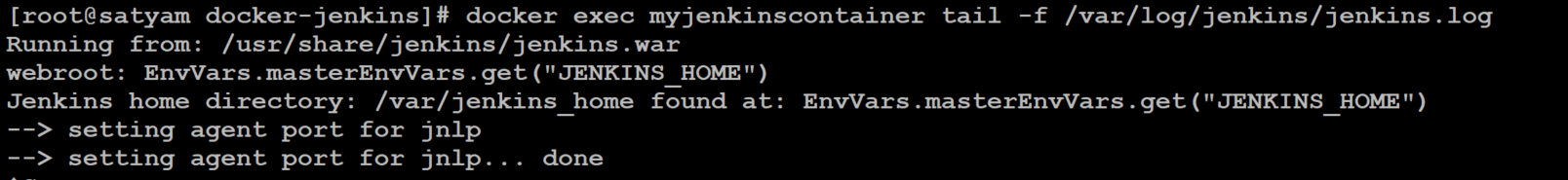
ENV JENKINS\_OPTS="--handlerCountMax=300 --logfile=/var/log/jenkins/jenkins.log"

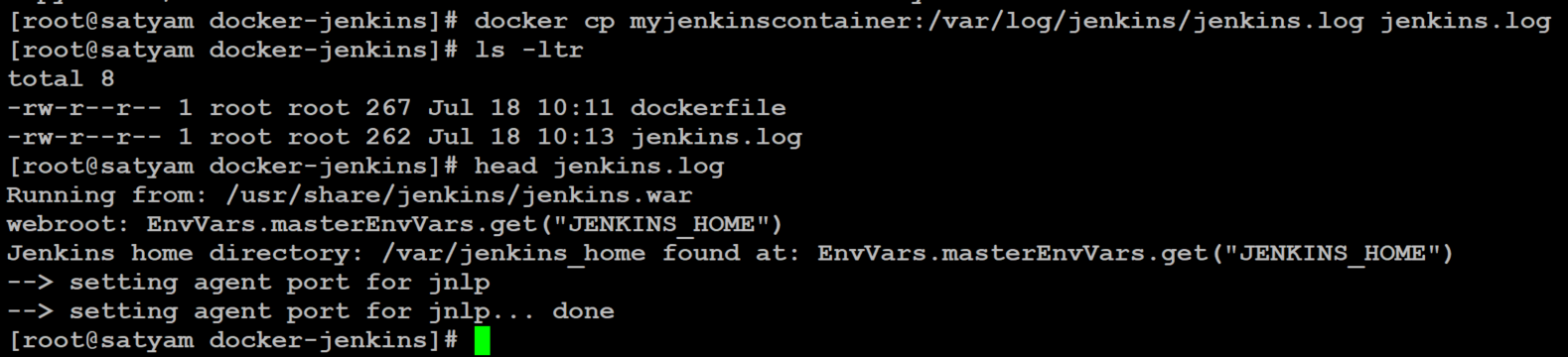
=========











=============================DockerFile