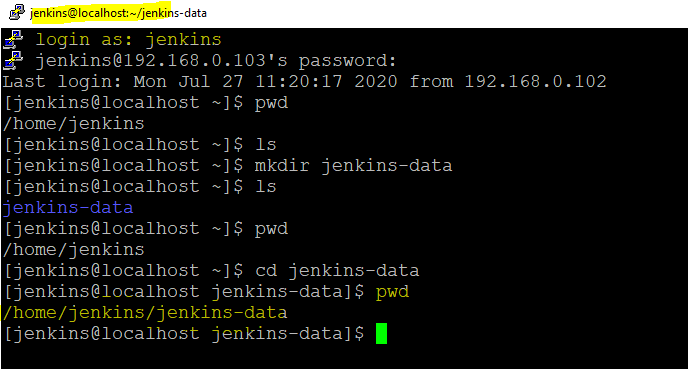
1. **Create a docker compose file for Jenkins**
2. Create a folder to Jenkins compose, Folder name Jenkins-Data



1. Created the docker-compose file

version: '3'

services:

jenkins:

container\_name: jenkins

image: jenkins/jenkins

ports:

- "8080:8080"

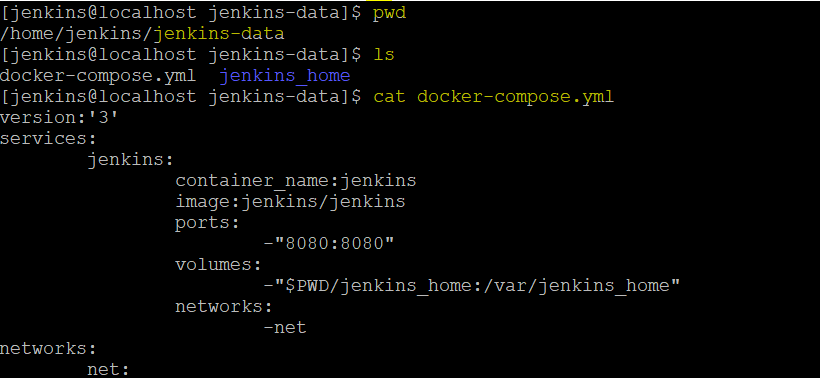
volumes:

- $PWD/jenkins\_home:/var/jenkins\_home

networks:

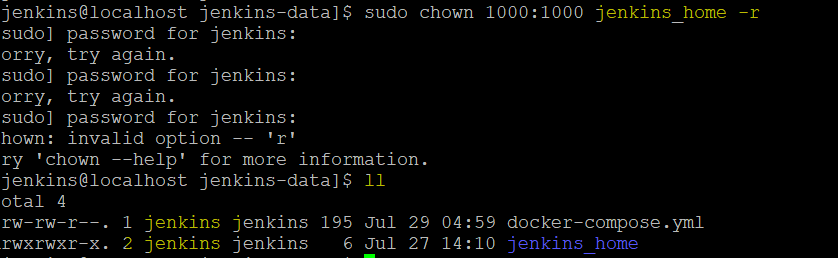
- net

networks:

net:

1. We have to make linux user(1000) have a rights to write in the Jenkins\_home directory ,

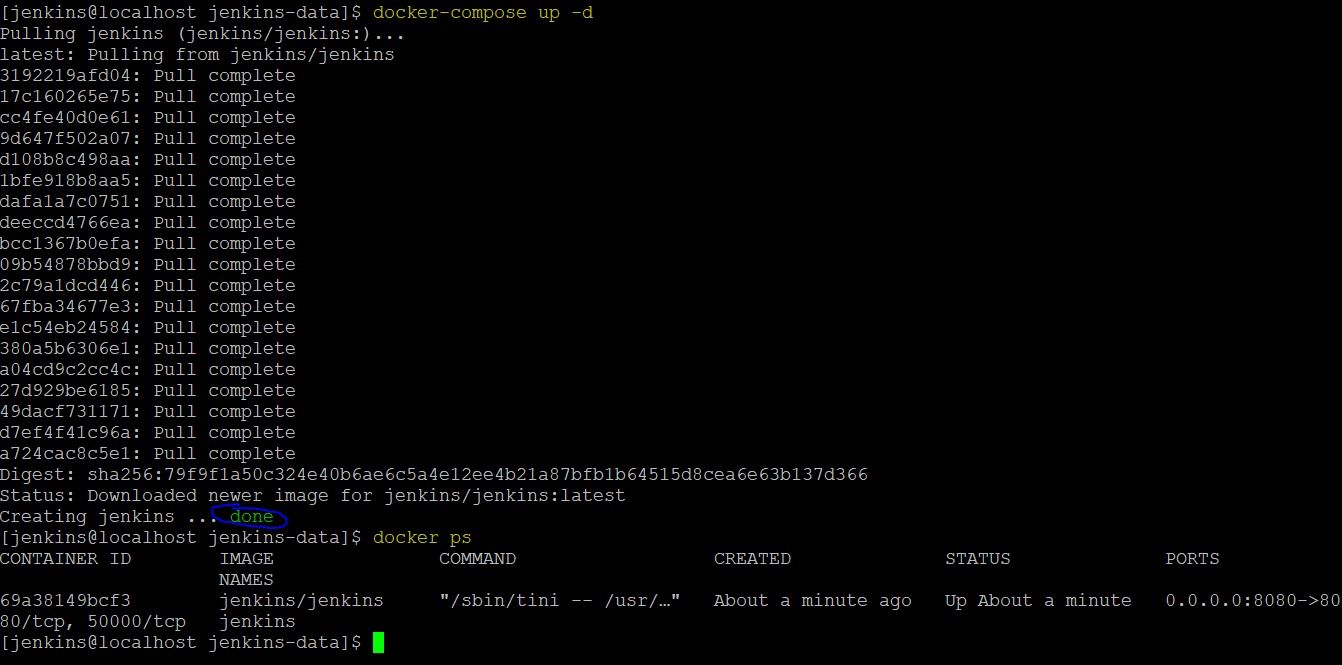
We can achieve it with sudo chown 1000:1000 jenkins\_home -r(r – recursively )



1. We can start docker service up by  
   **docker-compose up -d**

By this , docker will start service by reading the commands in the docker-compose.yml file,

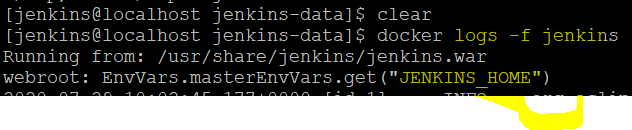
We can check what server running in docker, once we up it  
**docker ps**

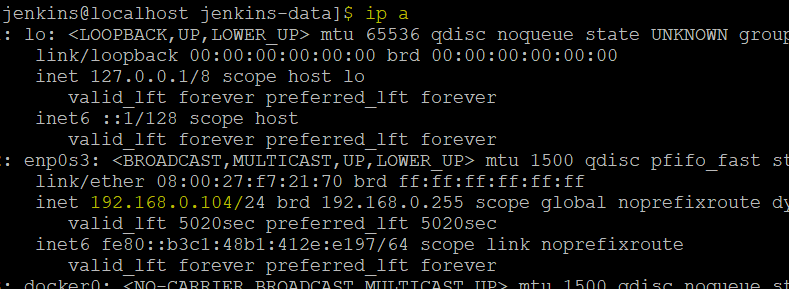


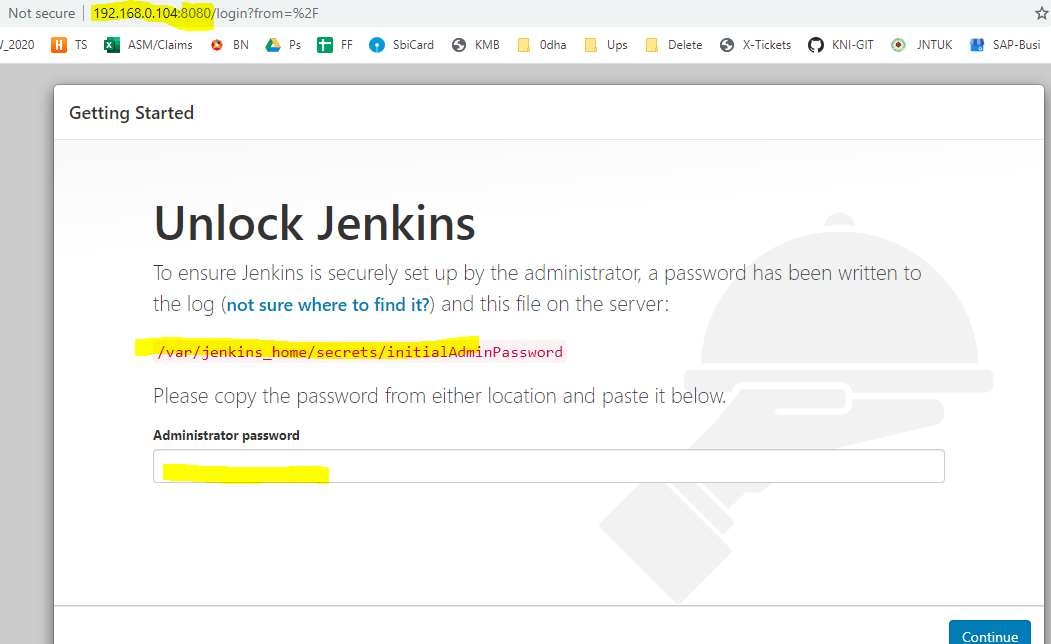
1. Get the key from the container by using the

Docker logs -f containerName

Ex: docker logs -f Jenkins



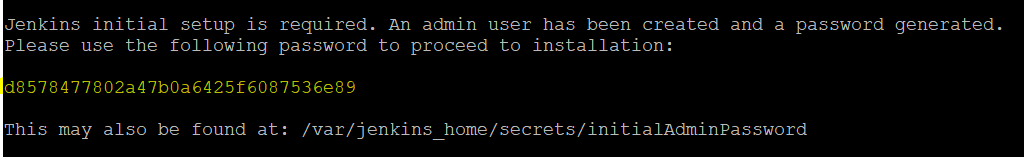
1. Call the service with help of Ip address
2. 



The password will be the key , we can get it by  
**Docker logs -f containerName**

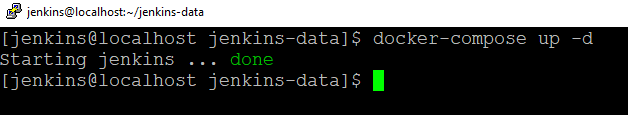
**Ex: docker logs -f Jenkins**

d8578477802a47b0a6425f6087536e89



1. Once password given , click on continue
2. Click on the plug-ins
3. Create a admin and set password
4. Now, create a local dns for Jenkins , instead of calling Jenkins with the IP address   
   open the notepad as admin, click on open ->goto system32->driver->etc->hosts-> give the Ip address and name of dns like [**192.168.0.104**](http://192.168.0.104/) **jenkins.local**

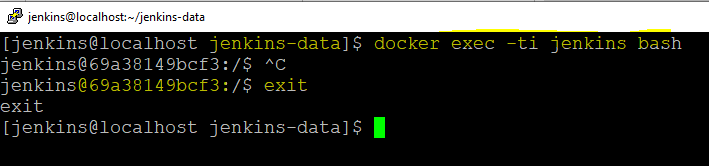
**Jenkins-Docker:**

1. We have to start the Jenkins service with help of putty  
   
2. To go into the container we will use

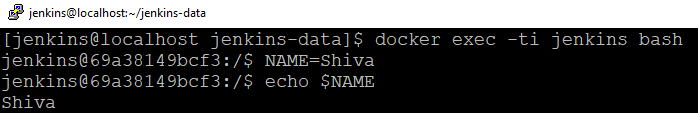
**$ docker exec -ti containerName bash**

**Ex: Docker exec -ti Jenkins bash**

1. Come out from the docker we use **exit**



1. We can define the variables and assign the values , can retriever

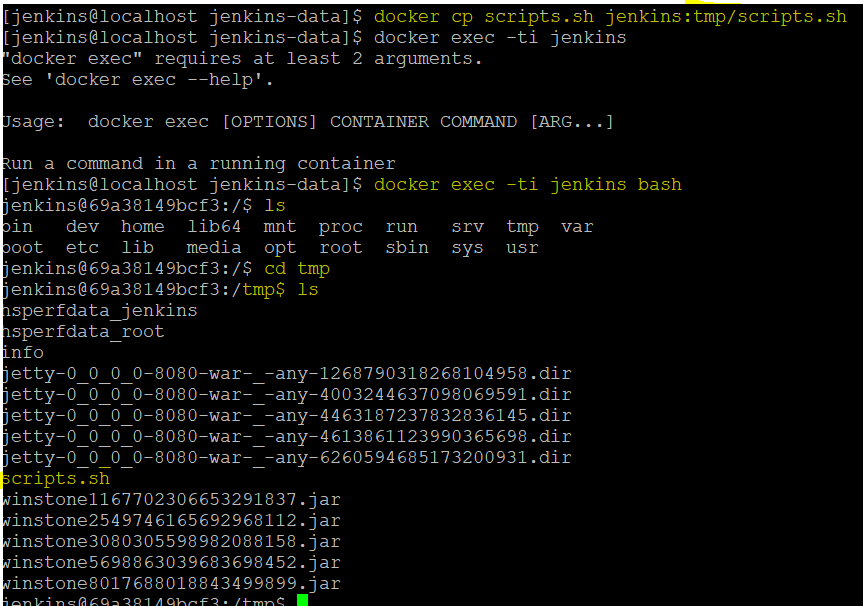


1. We can redirect the output to the file   
   NAME=Shiva  
   echo " Hello World,$NAME. Currrent Data and Time is . . . $(date**)"> /tmp/info**
2. Execute bash script from the Jenkins

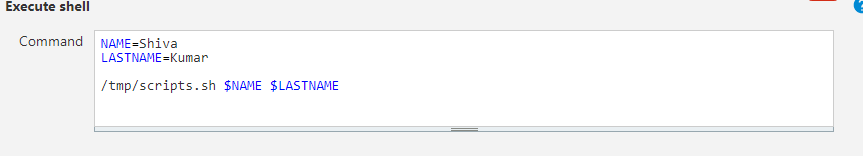
First we have to copy the bash script file to Docker container ,   
**docker cp scriptFile.extension container:/full path where to paste**

EX: docker cp script.sh Jenkins:/tmp/script.sh

After copy , we can execute it in the container **/tmp/script.sh**



After copy file to docker container , we will give the file path in the Jenkins to execute it.



1. **Docker + Jenkins+ssh**

Use case : when we have two servers to work on, trying to access/run/execute the Jenkins job the one server from another server, we need SSH key.

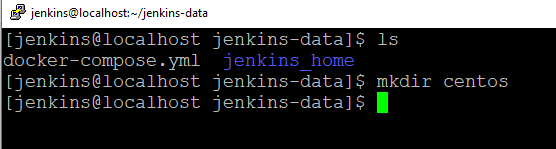
Lets work on this use case . . .

To work on this use case , we have to have 2 servers , as of now

1. We have 1 container called Jenkins in current linux
2. Lets create a remote server and install the docker image in it.

2.1 let us create a folder to save the new server

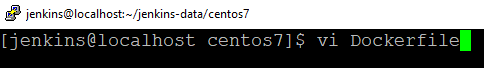
**mkdir centos7**



Move to the created folder **cd centos7**

**2.2**  create a docker image in it, with help of dockerfile.  
 with help of dockerfile command we can easily create an image , and then we can create a container in it.

**Vi dockerfile**



2.2.1 So let's create this file and in docker the first instruction to create a name which is to say what distribution we want to take as a container. In this case we use **centos**  as container

**FROM centos**2.2.2 from 2nd line , we can start write the commands . . .

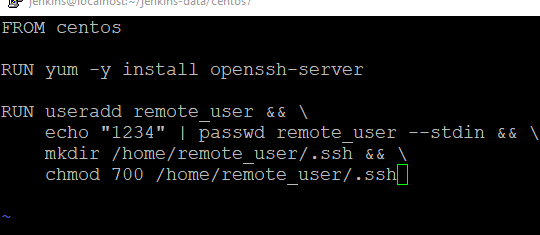
2.3 we will install the ssh server in it.  
**RUN yum – install opnssh-server**

2.4 adding the user into this distribution

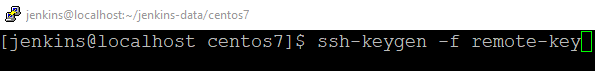
RUN useradd remote\_user && \

Echo “1234” | passwd remote\_user –stdin && \

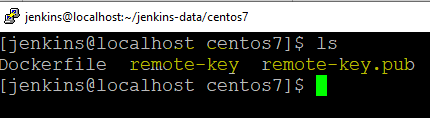
Mkdir /home/remote\_user/.ssh   
 chmod 700/home/remote\_user/.ssh



2.5 lets create a ssh key in centos7 , we can use this inorder to access the centos7.  
**ssh-keygen -f remote-key**



After this click on enter two times  
now , we can able to see the two files were created



2.6. lets edit the Dockerfile ,  
 add the **remote-key.pub**  to the **.ssh** folder and saving this as “authorized\_keys”

COPY remote-key.pub /home/remote\_user/.ssh/authorized\_keys

RUN chown remote\_user:remote\_user -R /home/remote\_user/.ssh/ && \  
 chmod 600/home/remote\_user/.ssh/authorized\_keys

Run /usr/sbin/sshd-keygen

CMD /usr/sbin/sshd -D

