Custom Course content:

1. Install Ubuntu container – Manually
2. Install Ubuntu container – Dockerfile
3. Manually install Jenkins on this container
4. Attach Centos container as node
5. Share Jenkins default job mouth point with Host
6. Change default Jenkins port
7. Jenkins behind reverse proxy
8. Github integration with Jenkins
9. AWS integration with Jenkins container
10. Plugin installation
11. Plugin downgrade
12. Plugin Different server installation
13. Plugin installation via hpi
14. Configuration slicing
15. Gradle installation
16. Maven installation
17. Make two jobs ( Basic compile and jar creation) with maven and gradle
18. Archive this jar
19. Publish this jar on AWS bucket
20. Admin to see all the jobs
21. One user to see maven jobs ( maven view)
22. Sonarqube
23. Java docs
24. Nodes ( centos slave to Ubuntu master)
25. Try to provision a container in Centos during build time
26. Jenkins File
27. Job DSL many jobs
28. Jenkins Backup.

So here is the thing. We are going to create three docker containers inside windows docker engine

One nginx to be used as reverse proxy named as Falcon-proxy

One Ubuntu that will be out master Jenkins named as Falcon-master

One Centos container that will be slave named Falcon-slave

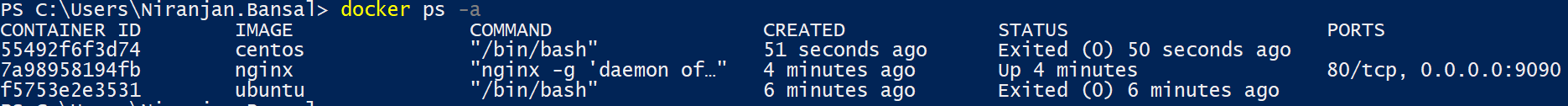
Have one Docker network and attach all these containers to same network.

Commands:

docker run -p 8080:8080 --name falconmaster ubuntu

docker run -p 9090:900 --name falconproxy nginx

docker run -p 7070:7070 --name falconslave centos



All these containers will be in stopped state. So lets ssh

<https://stackoverflow.com/questions/45356985/how-to-run-an-existing-stopped-container-and-get-inside-the-bash>

Note: The problem I think you're running into is that the command you provide is exiting immediately and for the container to keep running it needs a command that will not exit. One way I've found to keep containers running is to use the -d option like so

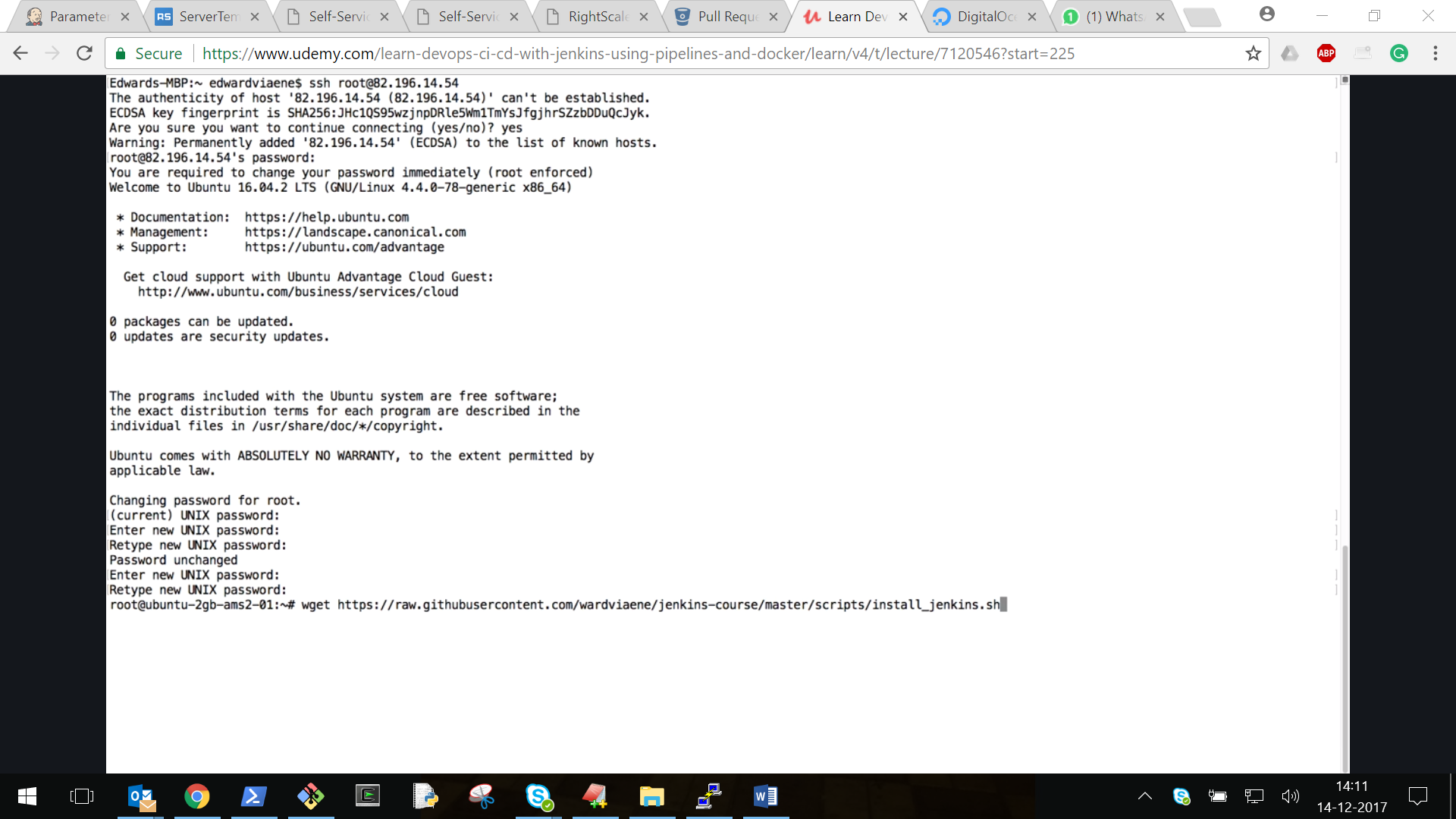
<https://stackoverflow.com/questions/29599632/docker-container-is-not-running>

NOTE:

|  |  |
| --- | --- |
|  | **By default, docker container will exit immediately if you do not have any task running on the container.** |

Jenkins:

Install Jenkins:



Install\_jenkins.sh on ubuntu:

#!/bin/bash

# this script is only tested on ubuntu xenial

# install docker

apt-key adv --keyserver hkp://p80.pool.sks-keyservers.net:80 --recv-keys 58118E89F3A912897C070ADBF76221572C52609D

apt-add-repository 'deb https://apt.dockerproject.org/repo ubuntu-xenial main'

apt-get update

apt-get install -y docker-engine

systemctl enable docker

systemctl start docker

usermod -aG docker ubuntu

# run jenkins

mkdir -p /var/jenkins\_home

chown -R 1000:1000 /var/jenkins\_home/

docker run -p 8080:8080 -p 50000:50000 -v /var/jenkins\_home:/var/jenkins\_home -d --name jenkins jenkins

# show endpoint

echo 'Jenkins installed'

echo 'You should now be able to access jenkins at: http://'$(curl -s ifconfig.co)':8080'

All the code related to Jenkins course is at

<https://github.com/wardviaene/jenkins-course>

Run install\_jenkins.sh script to install Jenkins. This is very good script for Jenkins installation on Ubuntu

What is CI and CD

