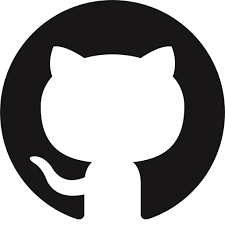
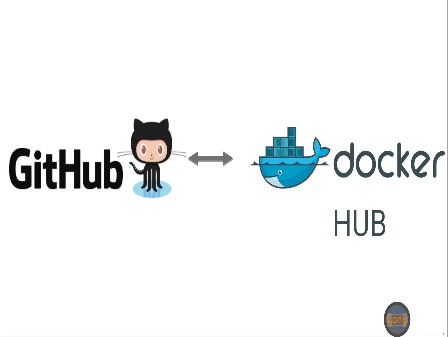
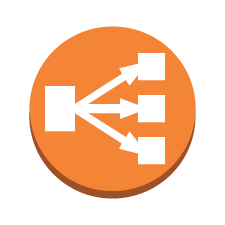
# Tools needs

* Github account
* AWS account
* Docker hub account

# Architectural Design.





**VPC subnet**

**VPC subnet**

**Commit and Sync to master**

**Local GitHub Repo**

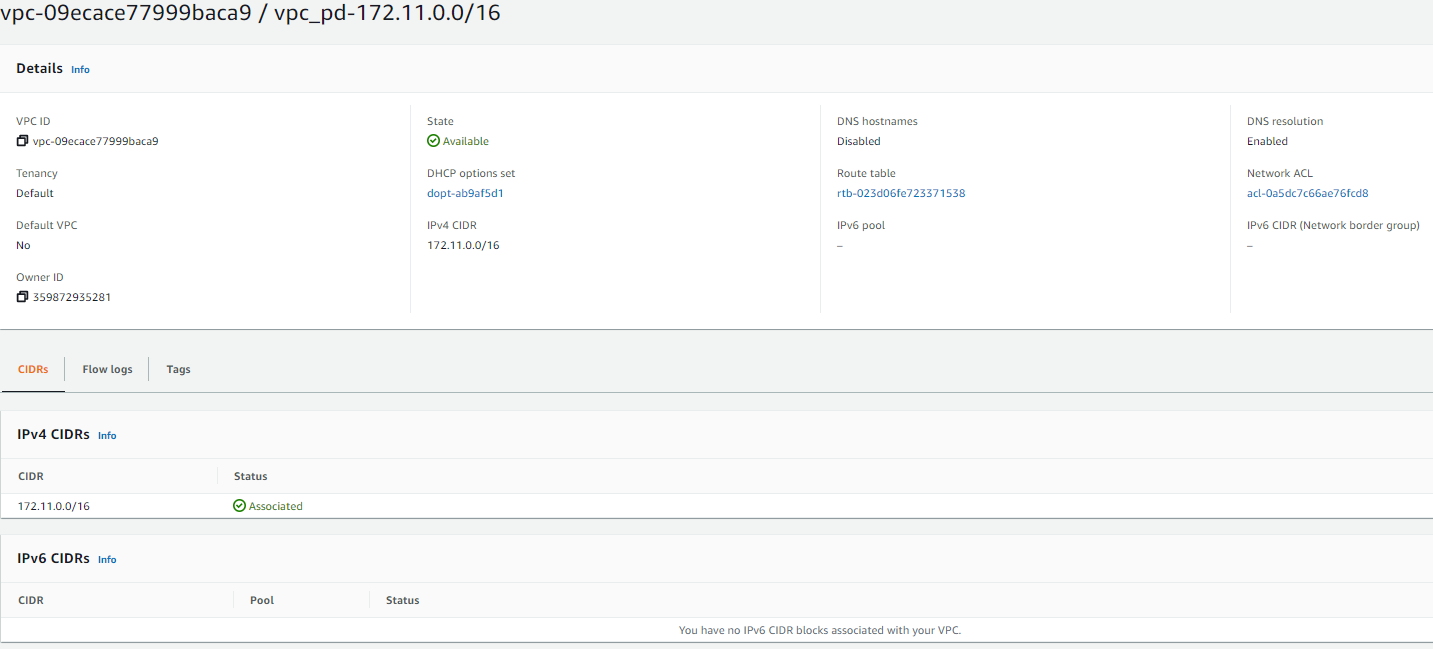
**User Commits**

# AWS

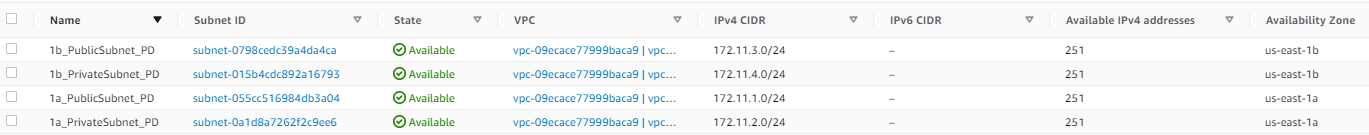
1. VPC

Primary **CIDR** block for PD VPC 172.11.0.0/16

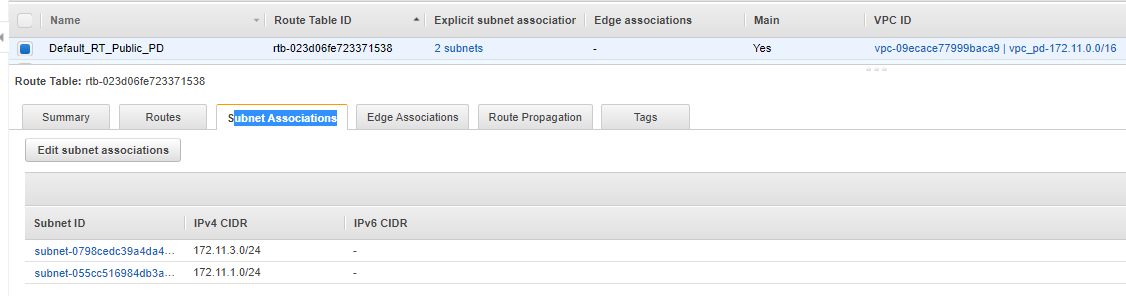
# VPC

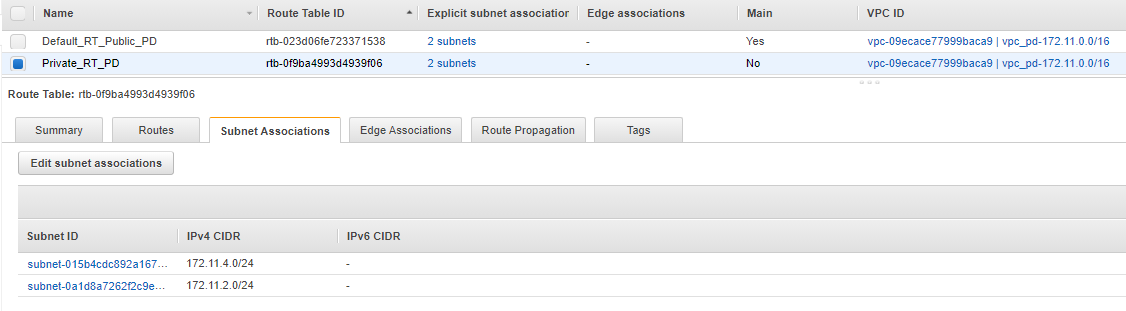


# Subnets

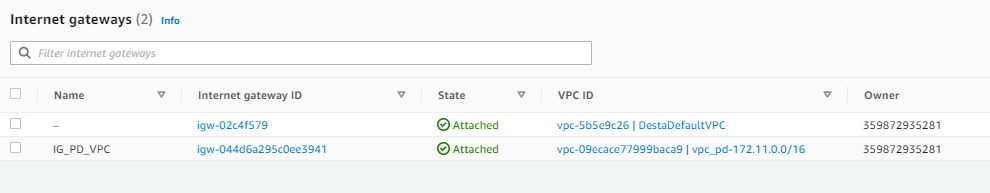


# Subnet Associations

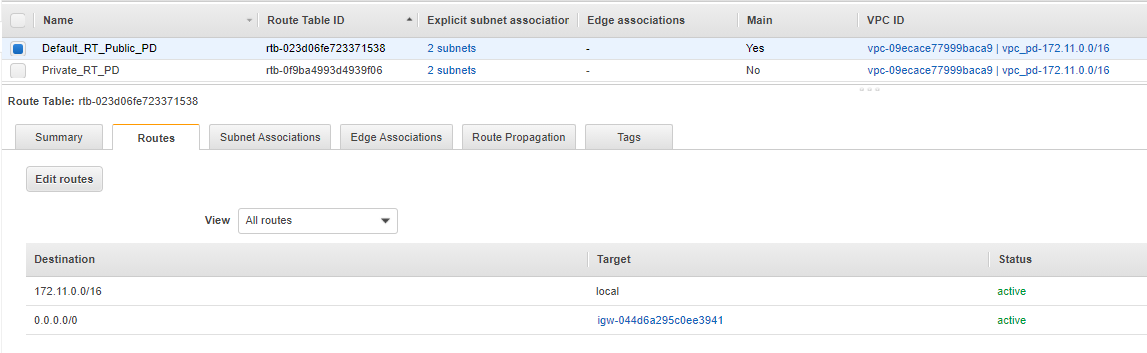




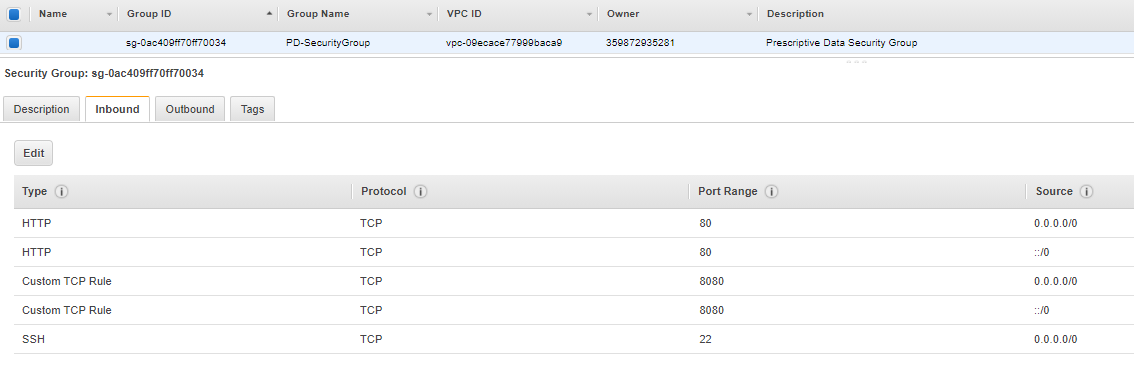
# Create Internet Gateway and attach to VPC



# Add Internet Gateway to Public Subnet Routes



# Define Security group



# Docker creation on AWS EC2 instances

1. Launch an EC2 instances, and configure which play role as Host Instance for running Docker on it
2. Run the following commands

sudo su

yum update -y

yum install docker -y

service docker start

1. Yum install git -y
2. git clone <https://github.com/destaalemu/github-pd.git>
3. docker build destaalemu .
4. docker build -t destaalemu .
5. docker run -p 80:80 destaalemu/dockerhub-pd
6. docker push destaalemu/dockerhub-pd
7. SSH to second instance …. As

**ssh -i "pd\_KeyPair.pem" ec2-user@54.166.160.180**

sudo su

1. yum update -y
2. yum install docker git -y
3. service docker start

create new repo github-pd-automatedbuild

git clone <https://github.com/destaalemu/github-pd-automatedbuild.git>

copy same code to new repository

make changes to the index

add, commit, push to github

goto docker hub and create automated repo linking to the github

make changes to the code

add, commit, push automatically it will be impacted autobuild

\*\*\*

sudo su

yum update -y

yum install docker git -y

service docker start

docker images

docker pull destaalemu/dockerhu-pd-automatedbuild

docker run -p 80:80 destaalemu/dockerhu-pd-automatedbuild

check the url ()

Note:

Make changes

Add, commit push to github

Verify that docker auto build is done

Down load new docker image and run

Alpine

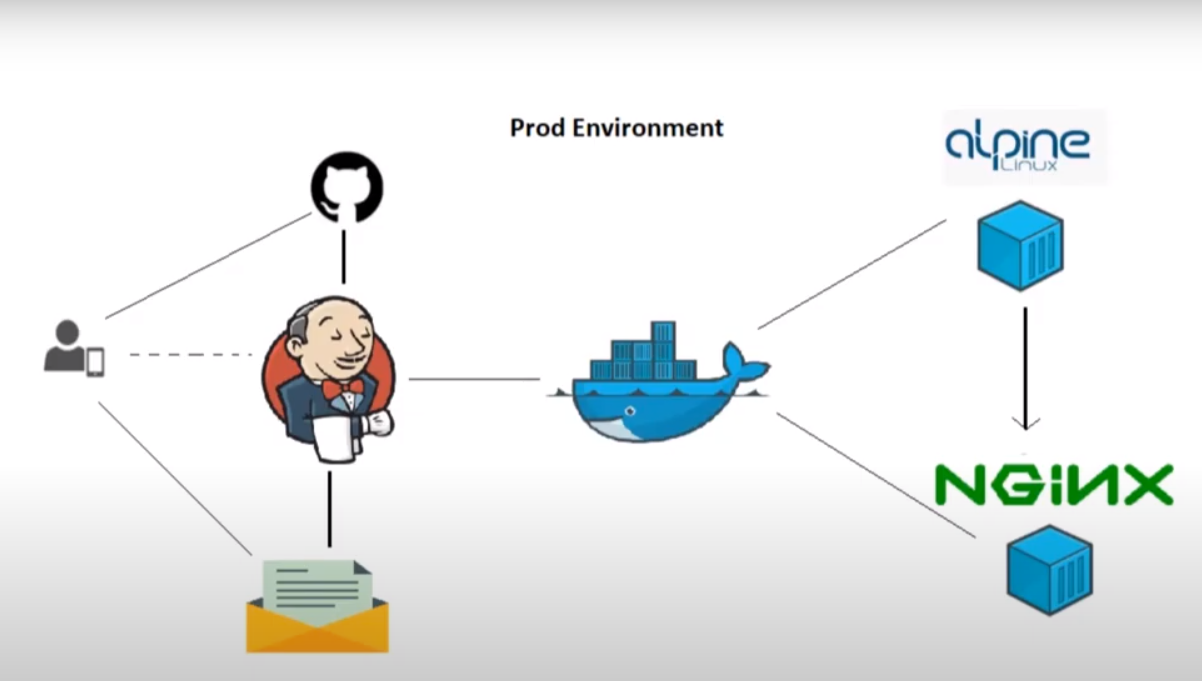
docker pull destaalemu/**dockerhub\_alpine**

docker images

docker run -p 80:80 destaalemu/**dockerhub\_alpine**

check the url ()

# Ubuntu



1. Launch three AWS ec2 instances, and configure based on the above VPC, Subnets, and Internet gateways

* Load Balancer
* Worker node 1 (Web App 1)
* And Worker Node2 (Web App 2)

1. Configure putty and connect

* **ssh -i "pd\_KeyPair.pem" ubuntu@3.87.65.220**
* **ssh -i "pd\_KeyPair.pem" ubuntu@107.21.10.162**
* **ssh -i "pd\_KeyPair.pem" ubuntu@54.227.182.231**

in my case am running from windows so, I used Putty, configured ssh connection with public IP, generated public private key “**pd\_KeyPair.ppm**” from “**pd\_KeyPair.ppm**” using PuttyGen

1. Install
   1. Install Jenjins
2. Run update

sudo apt-get update

1. Install open jdk

sudo apt-get install openjdk-8-jdk

1. open repository for jenkins and add key to repo

wget -q -O - <https://pkg.jenkins.io/debian/jenkins.io.key> | sudo apt-key add –

1. add Jenkins list entry to /etc/apt/sources.list

sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'

1. Update your local package index, then finally install Jenkins:

sudo apt-get update

sudo apt-get install jenkins

1. check if the service is working

service jenkins status

1. install docker

sudo apt-get install docker.io

* 1. Install Worker-1 and Worker-2

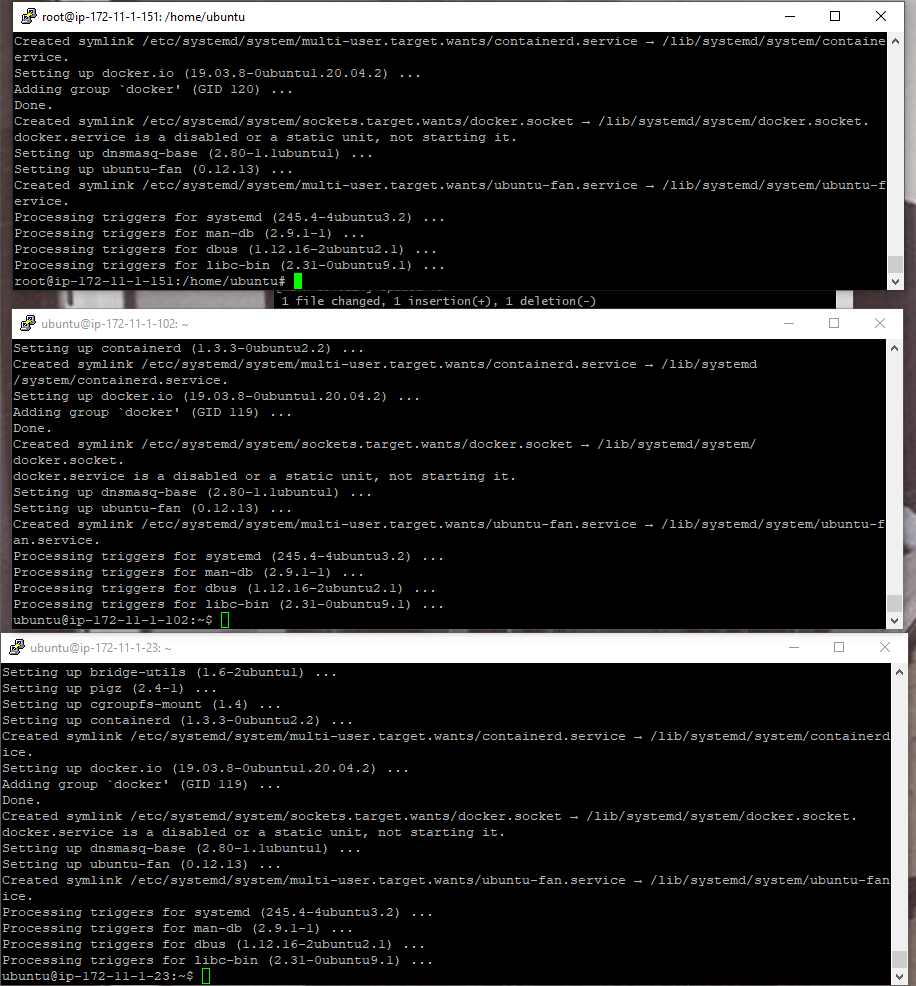
install open jdk , java jdk and docker

sudo add-apt-repository ppa:openjdk/ppa

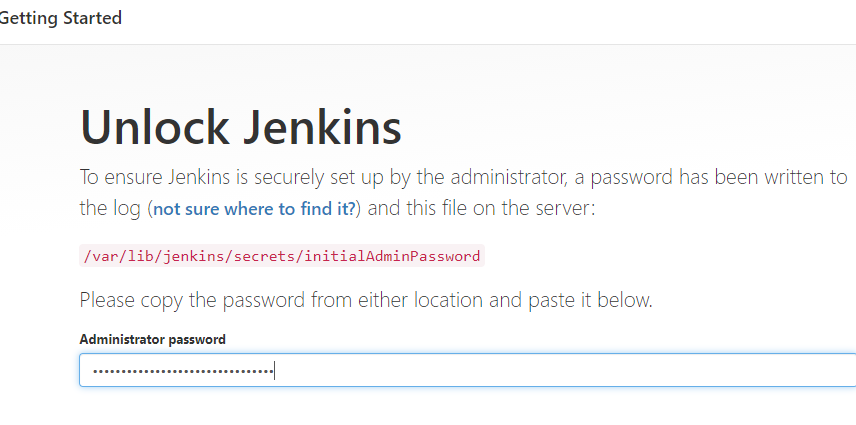
sudo apt-get update

sudo apt-get install openjdk-8-jdk

sudo apt-get install docker.io



Explore to Jenkins



Configure global security

* Set … to random

Create nodes

WorkerNode1 and WorkerNode2 for main and secondary connection

1. Alpin NGnix

install open jdk , java jdk and docker

sudo add-apt-repository ppa:openjdk/ppa

sudo apt-get update

sudo apt-get install openjdk-8-jdk

sudo apt-get install docker.io

Note:

Make changes

Add, commit push to github

Verify that docker auto build is done

Down load new docker image and run

Alpine

Sudo su

docker images

docker pull destaalemu/dockerhub\_alpine

docker images

docker run -p 80:80 destaalemu/dockerhub\_alpine

check the url ()