Day1 : 22-01-2022

Phase 4 : Testing and deployment

Software testing

Grunt task runner for JavaScript

Docker

CI and CD : Jenkins

Cloud computing

AWS

Ec2, S3 and EBS modules

Graph QL : Self learning

Elastic Stack : Self learning

Testing : Testing is use to find the defects or errors or bugs.

Read value a, b,

Process compute sum = a+b

Display value write b

Layer architecture

MVC

function operation() {

}

function add(a,b) {

// coding

var sum = a+b;

return sum;

}

Function sub() {

}

Function mul() {

}

Testing mainly divided into 2 types

Black box testing

Input -------------------🡪Process ---------------------🡪Output

Raj,123 success

Failure

Ravi, 123,

Age

21 60

Age between 21 to 60

>=21 <=60

<21 >60

Special symbol

-ve number

White box testing

Input --------🡪Process -----------🡪Output

Unit Testing :

Test suite and test case

describe(“”,()=> {

it(“”,()=> {

})

})

Unit testing : Unit testing is a kind of software testing method in which each individual and independent part of the source code or functions tested developed by developer.

Unit can be function or method or module or class.

If we do unit testing for UI side then we can call front end unit testing.

We can do backend side unit testing with respective framework.

Front end unit testing : Jasmine :

Jasmine is open source library which help to do the unit testing for Client side as a well as Server scripting language.

Jasmine provide pre-defined function which help to write test suite, test cases and expect functionality to check actual and expected output.

Old version of jasmine to generate the report or output we were depends upon Another ie Karma.

Karma is test runner for the Jasmine testing framework.

But new version Jasmine provide pre-defined plugin which help to generate the report or result without depending upon the Karma test runner.

Client Side JavaScript testing : Jasmine framework + Karma with old version Jasmine tool

Jasmine Framework with inbuilt plugin to run the tool.

Server Side JavaScript testing (Node JS): Node JS provided pre-defined external module jasmine which help to do the Testing for Server Side JavaScript (Node JS). In Server side JavaScript karma not required.

Mocha : Mocha is light weighted library which upon Jasmine. Using Mocha we can testing for Client side as well as server side testing.

Jasmine Vs Mocha

Mocha with Chai

Server Side JavaScript testing we can do with Jasmine or Mocha with Chai.

Client side testing Using Jasmine Plugin

Test suite : Test suite is use to write more than one test cases as well as another test suite.

To write the test suite using testing framework.

describe(“message”,callbackfunction);

Example

describe(“Operation Testing ”,()=> {

});

Test case : Test case is a type of test function which help to check the function functionality with the help of more than one except functions.

Syntax

it(“message”,callback);

describe(“Operation Testing”,()=> {

it(“Additing Testing”,()=> {

// write the code to call the function and

// using expect function we will check actual and expected output

})

})

Assert function : Jasmine framework provided lot of pre-defined assert function in the form of expectXXX() syntax to check actual and expected output.

Day 2 : 23-01-2022

.toBe(expected)

expect().toBeCloseTo(expected, precisionopt)

expect().toBeDefined()

expect().toBeFalse()

expect().toBeFalsy()

expect().toBeGreaterThan(expected)

expect().toBeGreaterThanOrEqual(expected)

expect().toBeInstanceOf(expected)

expect().toBeLessThan(expected)

expect().toBeLessThanOrEqual(expected)

expect().toBeNaN()

expect().toBeNegativeInfinity()

expect().toBeNull()

expect().toBePositiveInfinity()

expect().toBeTrue()

expect().toBeTruthy()

expect().toBeUndefined()

expect().toContain(expected)

expect().toEqual(expected)

expect().toHaveBeenCalled()

expect().toHaveBeenCalledBefore(expected)

expect().toHaveBeenCalledOnceWith()

expect().toHaveBeenCalledTimes(expected)

expect().toHaveBeenCalledWith()

expect().toHaveClass(expected)

expect().toHaveSize(expected)

expect().toMatch(expected)

expect().toThrow(expectedopt)

expect().toThrowError(expectedopt, messageopt)

expect().toThrowMatching(predicate)

expect().withContext(message)

Jasmine hook functions.

beforeEach() :This is life cycle or hook function it will call before each it function.

afterEach() : it will call after each it function

beforeAll() : it will call only once before all it function

afterAll() : it will call only once after all it function.

It()

deposit amount

It()

without amount

Jasmine client side testing using node js

First create the folder and install two dependencies

**Npm install –D jasmine-browser-runner jasmine-core**

**npm install --save-dev jasmine-browser-runner jasmine-core**

or

npm install –D **jasmine-browser-runner jasmine-core**

Next run the command as

npx jasmine-browser-runner init

This command is use to create the spec folder and jasmine.browser.json file.

npx jasmine-browser-runner serve

The application run on default port number ie 8888

<http://localhost:8888>

Day 3 : 29-01-2022

Angular Framework testing using Jasmine and Karma

Ionic framework for hybrid mobile application

React JS test using JEST library

With Redux or Flux

React native

Component angular as well as react js

@Input : parent to child

@Output or @ViewChild : child to parent

Shared service or sessionStorage or rxjs using observable : sibling component

In React JS using props we can share the data between one component to another component depending upon the their relationship.

Redux is state management tool which provide centralized tool to share the data between more than one component doesn’t matter what is the relationship between two components.

Create new angular project using ng command as

ng new angular-testing

routing no :

styling css

Jasmine is testing framework which provide pre-defined function ie describe, it and more than one expect.

It may be jasmine or jest or mocha.

Karma is a test runner which help to generate testing reports.

Angular Utilities testing classes.

TypeScript

We use special type of class using decorator @Component, @Service, etc

Angular providing pre-defined module ie @angular/core/testing

TestBed is pre-defined class which provide set of method which help to do the testing for Angular component and service.

To start the angular testing we have to use the command as

ng test

ng g class employee this command is use to create model or normal type script class.

Angular service testing

ng g s fake

Day 4 : 30-01-2022

Docker :

OS :

Window, Linux , Unix or Mac etc

VM : Virtual Machine

Base Machine : 16 GB RAM

Unix Guest OS or Virtual Machine

4 gb

12 GB

10 Guest OS

1gb

Docker : Docker is a advanced os virtualization software platform that makes it easier to create, deploy and run the application in a Docker Container.

Virtualization : Virtualization is the means of employing software such as hypervisor to create a virtual version of resources such as server, tool, data storage, or application.

Virtualization let you divide a system into series of separate section, each one acting as distinct individual system. The virtual environment is known as virtual machine.

MQ or

Python

Table

View

Docker is use to create containerization application.

Virtualization Vs Containerization

Virtualization is an abstract version of physical machine.

While containerization is the abstract version of an application or tool or application software/system software.

Docker providing images which help to run the application that application internally run on os.

Containerization is use create application, deploy the application and run the application on Docker engine.

Container : Container is a run time environment to the run the application.

Docker Container : Running the instance of Docker images container turn on or run the actual application. A container includes an application and all of its dependencies.

Docker Image : It is file system and configuration of our application which is use to create the container. Docker images are the source code of our container.

Docker file : A Docker file is blue print or set of instruction that defined how our images are built.

Docker registry Docker provide Docker hub it is like a git hub which help to publish the our own images as well as we can pull other open source images and run in our machine.

Docker hub account create.

docker --version

docker images : this command display all images present in local machine.

docker pull image-name : This command is use to pull the image from hocker hub to local machine.

docker run image-name/image-id : This command is use to run the image when we run the image the application present in image run.

Virtual Machine

1. It is not a container based model. They use user space along with kernel space on OS. Here the information store permanently.
2. It doesn’t share the host kernel.
3. Virtual machine don’t start quickly and lead to poor performance.
4. It can run only a limited number of VMs on a system depends upon the Base OS.

Docker Container

1. In Docker, the container share the host OS kernel space. Here the information store or hold temporary.
2. They share kernel space through images.
3. Docker container can start up quickly and result in less boot-up time.
4. With Docker container, user or programmer or developer can create the application and store it into a container images for temporary purpose.
5. Using Docker we can run multiple container at a time.

Create image to run the data information using busybox image

**Dockerfile**

#pull the image from local or docker hub

FROM busybox:latest

#run the date command.

CMD ["date"]

To build the image we have to execute the command as

Docker build –t imageName . –f Dockerfile

docker build –t my-busy222 . –f Dockerfile

Create image to run the node js application

App.js

let a = 10;

let b=20;

let sum = a+b;

console.log("Sum of two number is "+sum);

function sayHello(name){

    return "Welcome to Node js with docker"+name

}

console.log(sayHello("Ravi Kumar"));

**Dockerfile**

# pull the node js image base upon alpine image run time environemnt is ready

FROM node:latest

# copy app.js file into image in current path.

COPY app.js .

# open the command and run app.js file in node js image

CMD [ "node","app.js" ]

docker –t my-node222 . –f Dockerfile

. : current location of file

-f : file

Day 5

05-02-2022

git clone URL

git pull (in existing repository to get new update we have to use git pull);

creating image to run the express js application (web application).

npm init To create package.json file

**Dockerfile**

FROM node:latest

RUN mkdir /usr/src/app

WORKDIR /usr/src/app

COPY package.json /usr/src/app

RUN npm install

COPY app.js /usr/src/app

#COPY . .

CMD ["node","app.js" ]

docker build –t imageName . –f Dockerfile

**if image contains web application then we have to run the image using command as**

**docker run –p 9090:9090 imageName**

**9090 : actual application running port number in red color**

**9090 : expose port number to run the application in yellow color it may be same or different.**

docker run –d –p 9191:9090 imageName

-d : detached mode or background

To check all running images

docker image : this command is use to display all images

To check all running container

docker ps : This command is use to display only running container

To stop the container command is

docker stop container\_id : This command is use to stop specific container

docker start container\_id: : This command is use to start specific container

docker ps –a : This command is use to display all container (it may running or stopped).

docker stop $(docker ps -a -q) : This command is use to stop all container

(please run this command in gitbash)

To remove the image we have to use the command as

docker rmi image\_id/imageName

If we get the error then use –f

docker rmi image\_id/imageName –f

Stop the container

docker stop container\_id

Then

docker rm container\_id

tomcat

web logic

jboss

iis

apache

nginx

Creating the image to run static html, css and Javascript program

Create html or css or js file

**Dockerfile**

FROM nginx:latest

COPY index.html /usr/share/nginx/html

docker build –t my-web222 . –f Dockerfile

tomcat : 8080

nginx default port number is 80

Creating the image for Angular application and add build file in nginx server and create the image.

Create new project

ng new angular-docker-app

After angular development we have to build angular project and we have to give this build file to backend technologies like express js or spring boot or python or asp .net technologies.

Angular build file + spring boot project and create war file and deploy in web logic or tomcat server.

Angular build file + express js file and build on node js server.

We can deploy angular or react js project separately in different server

Backend technologies develop in different server

Both are interact to each other using REST API.

How to build the angular project

ng build : Build angular project

after build successfully we can see dist/projectname and build files.

nginx

then create the Dockerfile

FROM nginx:latest

COPY /dist/angular-docker-app /usr/share/nginx/html

docker build -t my-angular222 . -f Dockerfile

docker image

docker run -d -p 81:80 my-angular222

Then run the application on

<http://localhost:81>

06-02-2022

Before publish the image in docker hub we have to create the tag.

Tag version of images generally we write latest:

Syntax to give tag for local images

docker tag imageName:latest dockerhubid/imageName

Ex :

docker tag my-angular222:latest akashkale/my-angular222

docker push dockerhubid/imageName

Ex:

docker push akashkale/my-angular222

if you get the error as access denied :

docker login

docker run –d –p 87:80 akashkale/my-angular222

CI and CD :

Manager -🡪 Project Structure

Push in git repository

He or she verify master or main branch code with user-defined branch code if correct it will merge in master or main branch else reject.

Dev1 git clone default branch it may master or main

If we want to do any changes we have to do in user-defined branch

User must be push this branch in remote repository.

Dev2 git clone

Git pull

Dev3 git clone

Git pull

Dev n git clone

Code changes made by individual team members and merged together into working software.

This phase is known as integration phase.

Manager or other team member must be build the project after merged by any of team member code in remote master or main branch.

CI and CD

There are lot of tools are available in market

1. Jenkin

Jenkin is a type of CI and CD tool. It is a open source developed using Java language.

To Jenin in our machine we can use

1. Install the Jenkin software
2. Run the Jenkin.war file
3. We can use Jenkin image

docker pull jenkins/jenkins:lts-jdk11

docker run –d –p 8080:8080 jenkins/jenkins:lts-jdk11

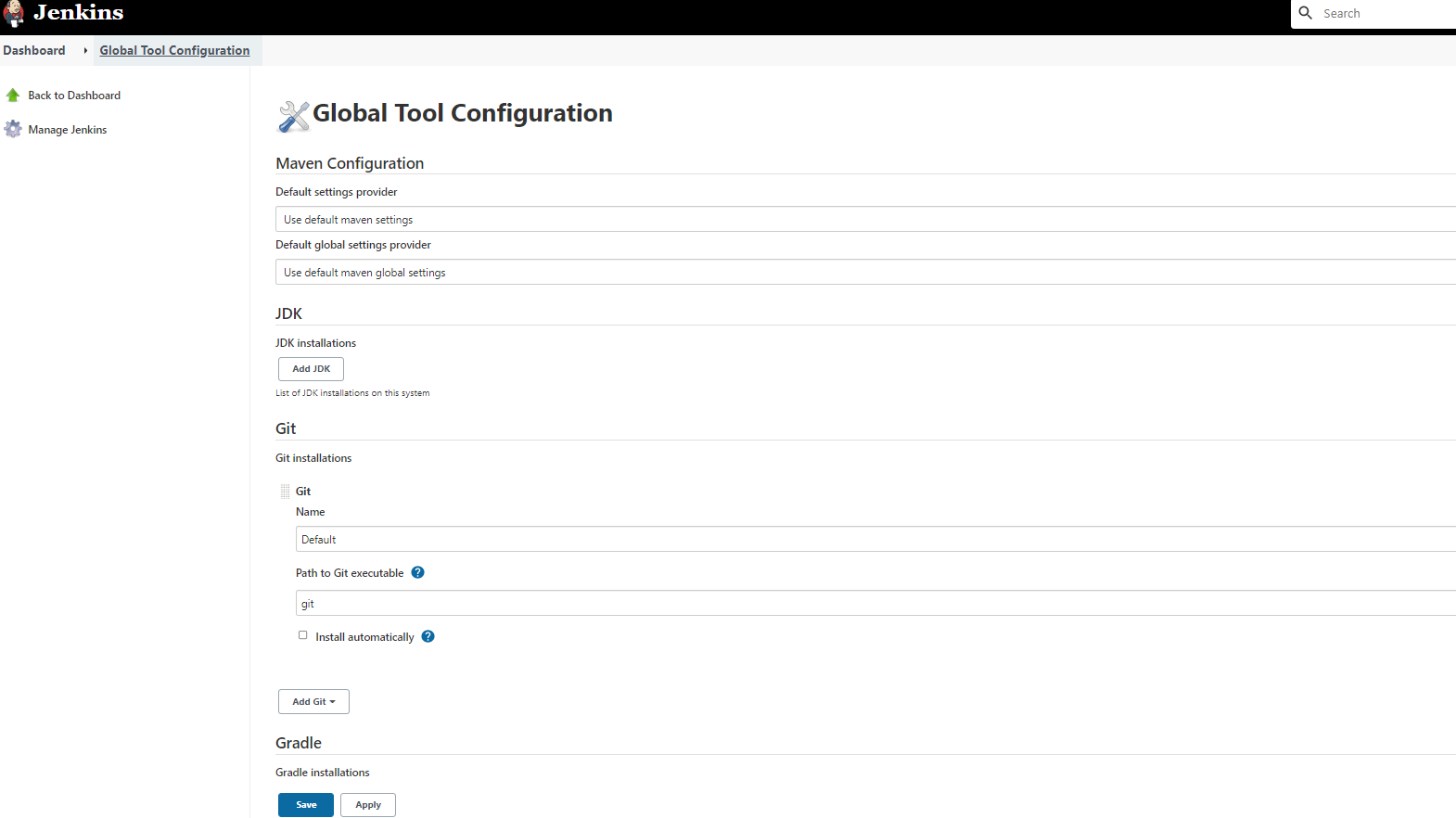
run without –d

docker run –p 8282:8080 jenkins/jenkins:lts-jdk11

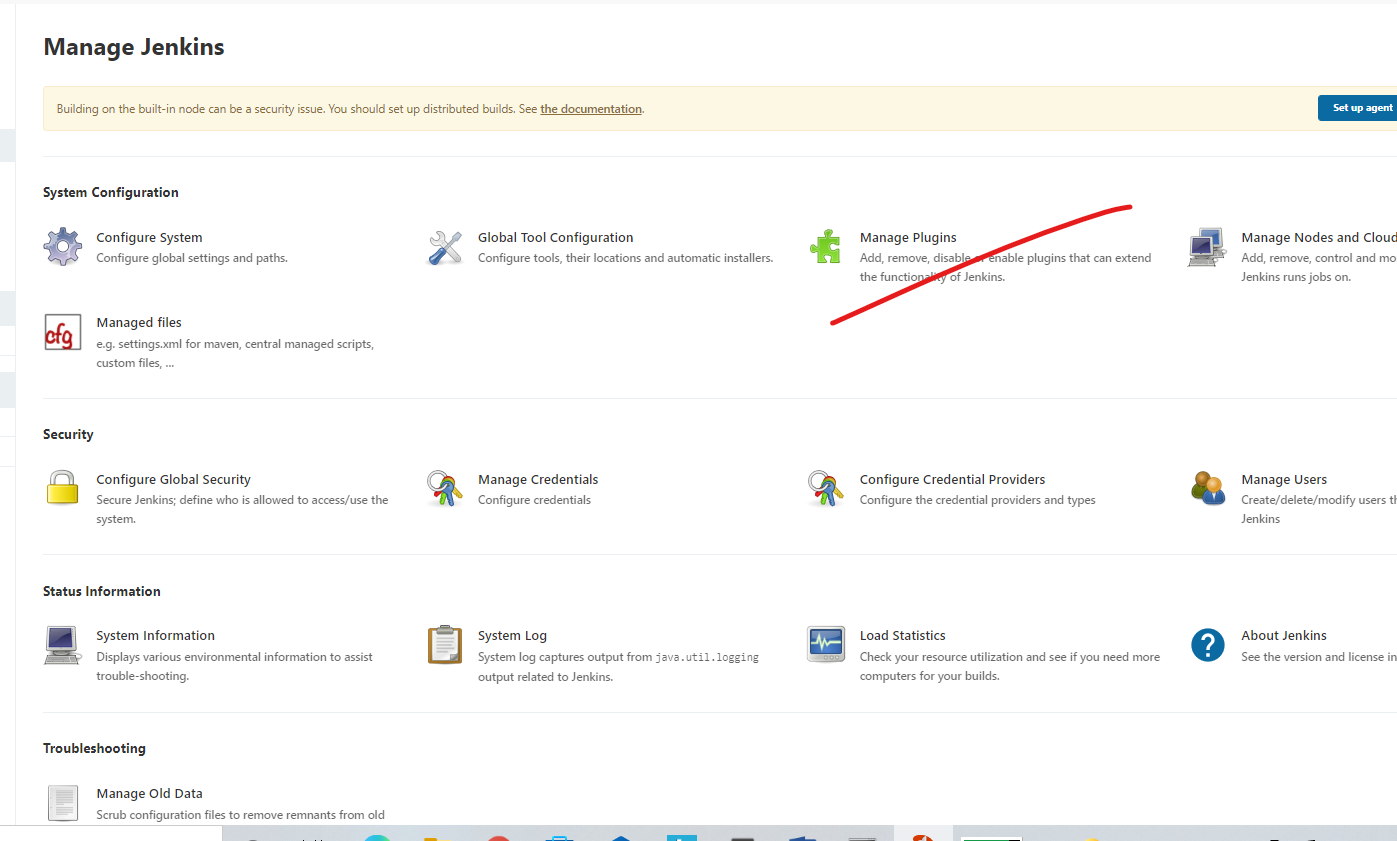
with above command it will display the temporary password on console.

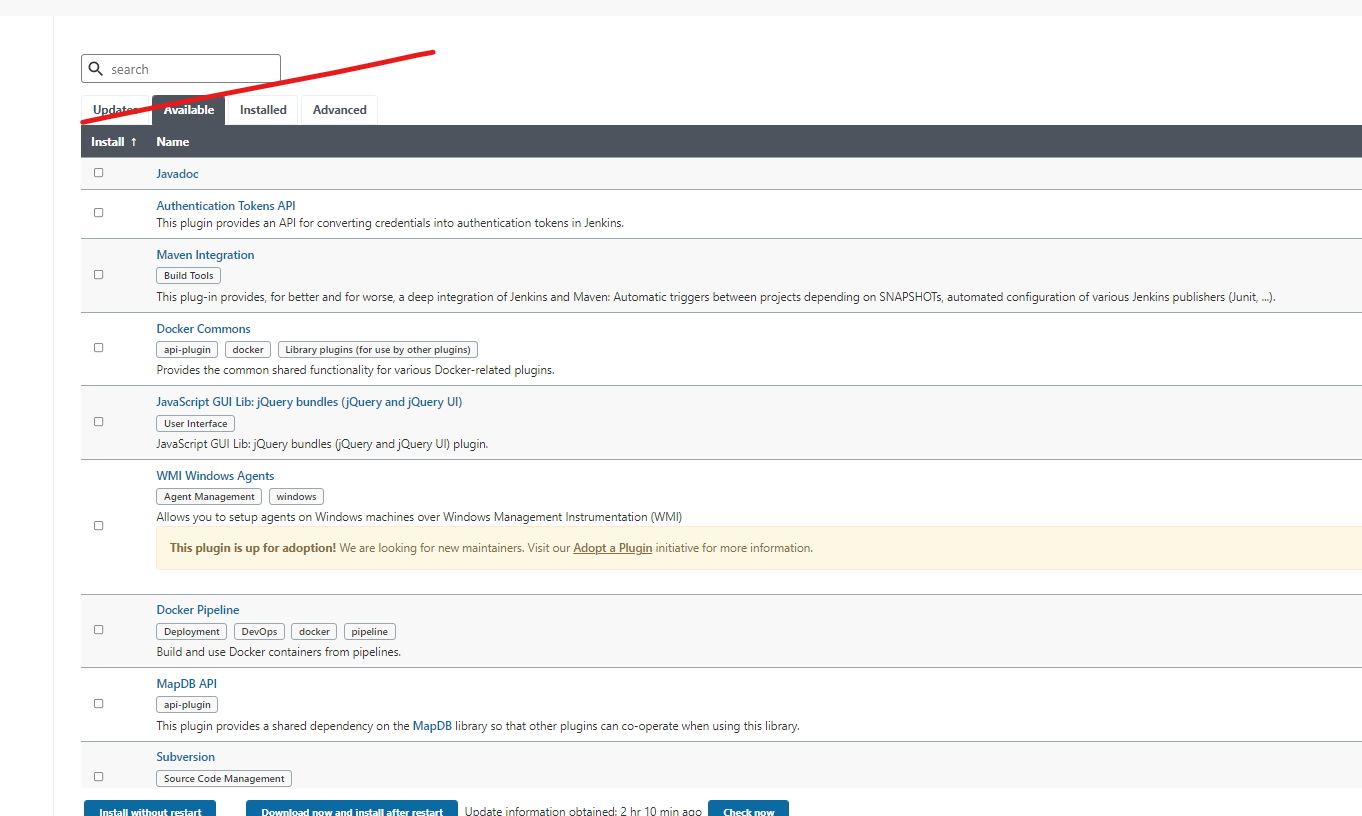


In Global tool configuration we have to verify the software present or not

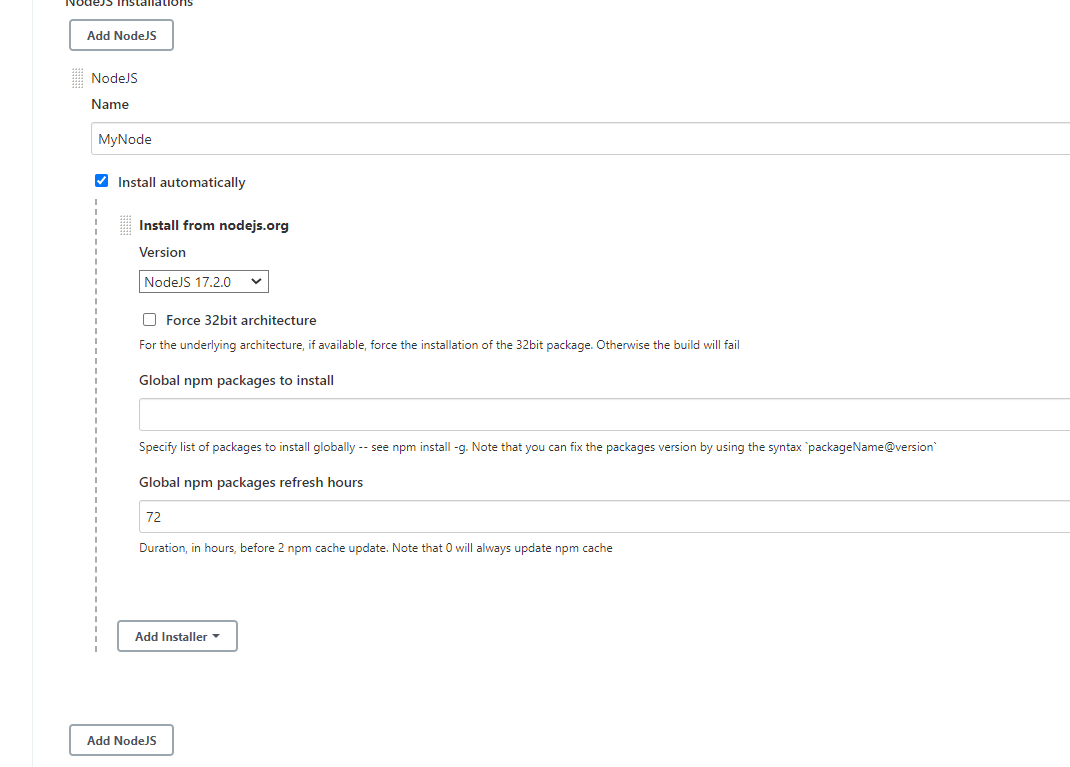


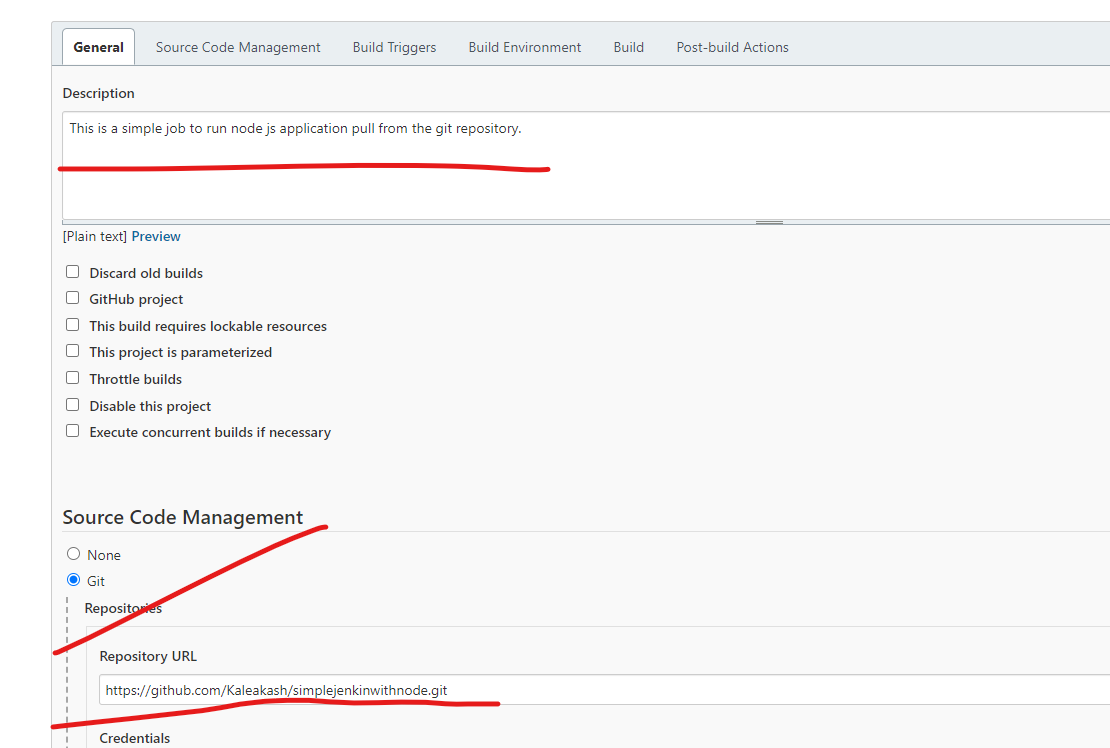
If it is not present then we have to install using manage plugin

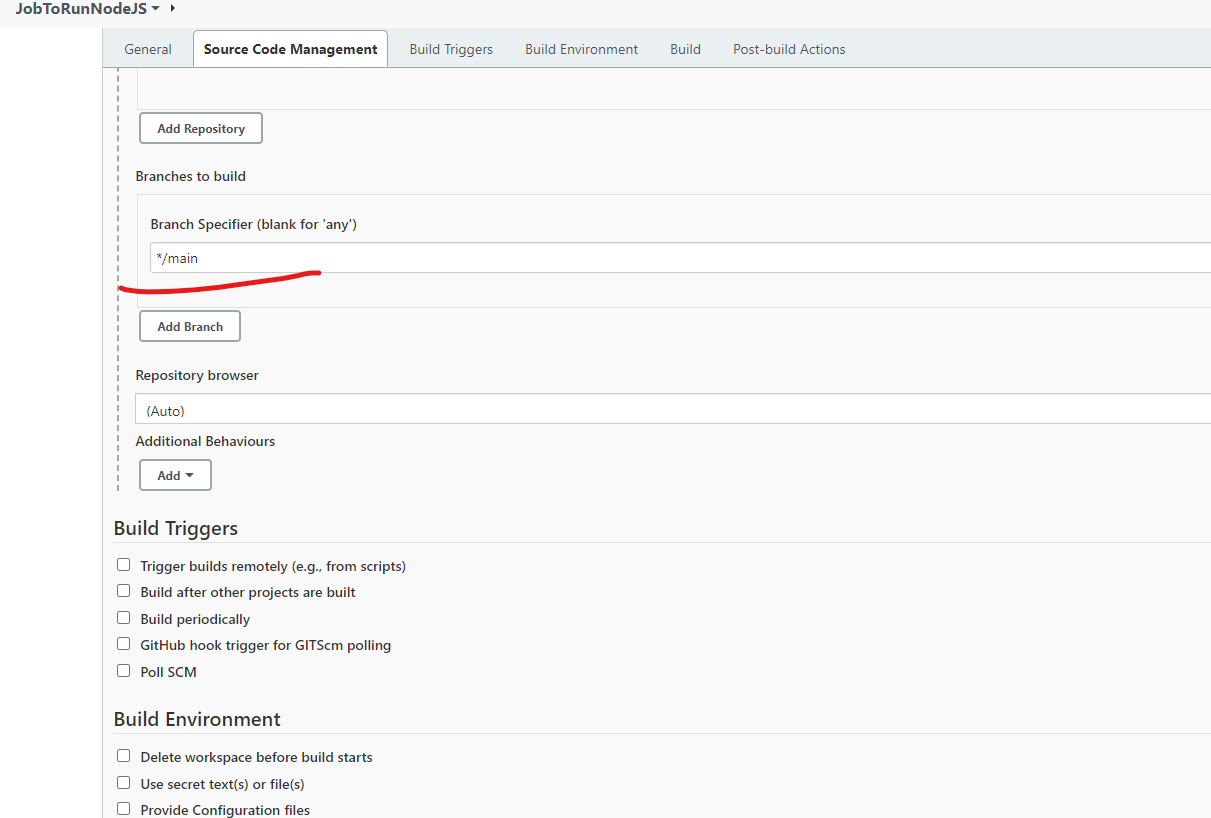


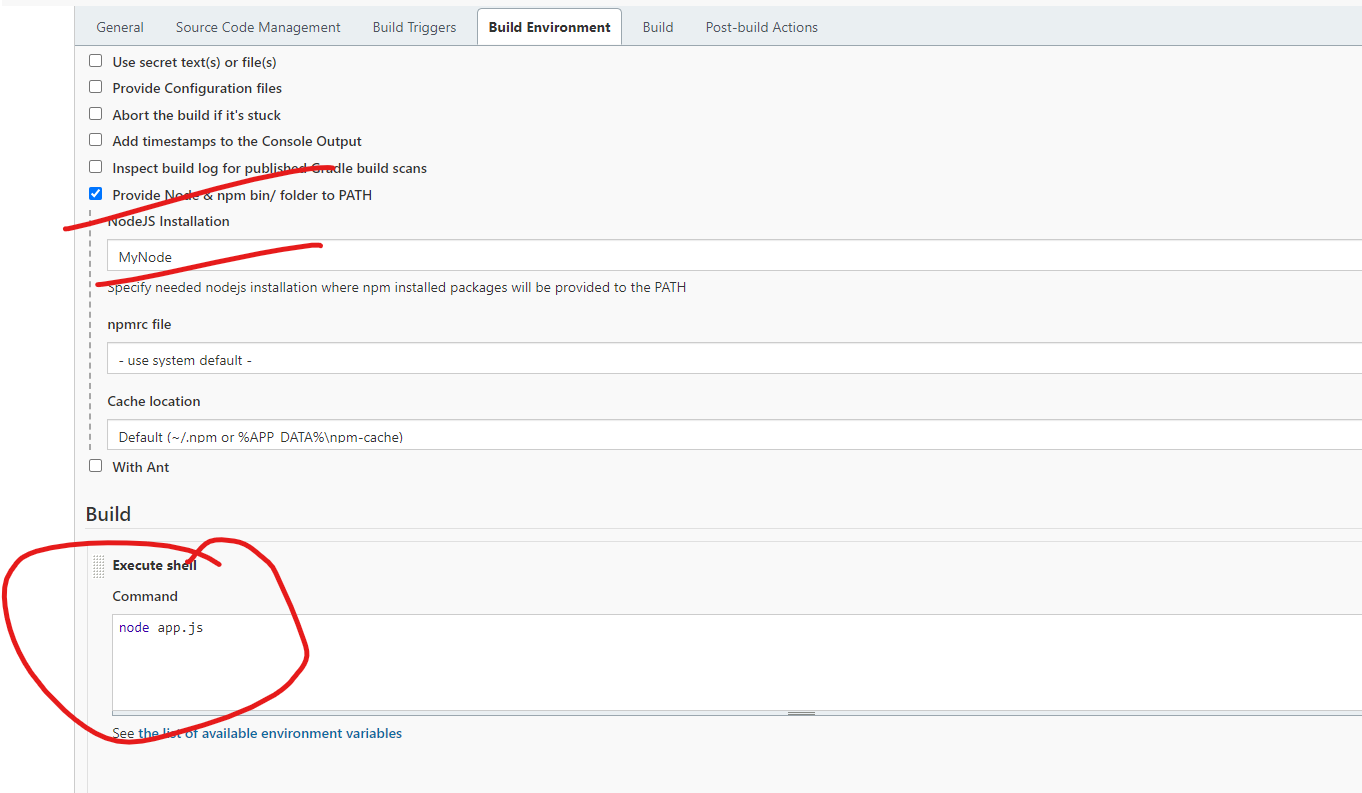


After downloaded you have to add this software to global tool configuration.









Jenkin pipe line : it is use to execute more than one jobs which are connected to each others.

Check environment

Build the project

Test project

Deploy the project.

Day 7

12-02-2022

Grunt : Grunt is known as JavaScript task runner.

1. CSS file compression
2. Converting ts to js
3. Unit testing
4. Build deployment

Etc

To do all above task we have to use command prompt or any other tools. But every tools is use to run any one specific task. Using Grunt we can take the help of Gruntfile and run more than one task depending upon our application requirements.

Grunt use command line tool which run on node js.

Grunt project folder and inside folder create sub folder as SimpleTask

npm install grunt-cli –g : to enable grunt command.

npm init : to create the package.json file

grunt locally not installed

npm install grunt : This is a module to run the task.

Create separate project as frontend app

Create package.json file using command as npm init

Now install grunt locally

npm install grunt

Minification : Minification also known as minimization, it is a process of removing all unnecessary character from html, css and Js or ts files.

Grunt provide lot of pre-defined minification modules

1. Html minification
2. Css minification
3. Js minification

Etc

npm install grunt-contrib-clean -D : This module is responsible to clean all build files.

npm install grunt-contrib-copy –D : this module is responsible to copy all files from src folder to dist folder.

npm install grunt-contrib-uglify–D : this module is use to combine all js file into one build js file.

Graph QL (Query Language):

Rest API has become the dominant API style for building backend web service. With REST we request the resources using GET, POST, PUT and Delete etc.

But REST full Web service has few limitation

1. Over fetching
2. Multiple request for multiple resources
3. Water fall network request on nested data.

Graph QL let ask for what you want in a single query , saving bandwidth and reduce waterfall request.

Using REST API we can’t retrieve partial data or payload. But using Graph QL we can retrieve partial payload data.

Graph QL Folder

Create the package.json file using npm init command

npm install express graphql express-graphql

**We created Docker images (Docker container for individual application)**

When we develop big application we require multiple container to run more than one application.

Those container want to communicate with each others.

Container -🡪 Express JS : node js image

Container 🡪 Mongo DB : alpine or busybox

Container -🡪 Angular JS using nginx server ngnix : internally using alpine image

Angular container have to call Express JS container and Express js container must be call mongo db container.

Docker compose : Docker compose is a toolkit provided by Docker to build, ship and run multi container application.

Docker provide docker-compose command to multi container application.

To provide all container details we have to use docker-compose.yml file (yet another markup language).

We have to configure our all container details.

Docker compose is used for configuration and starting multiple container on the same host or same machine rather than start each container separately.

Docker swarm : It is a container orchestration tool that allow you to run and connect more than one container running on different host or machine or node.

Kubernetes : it is known as container management tool or system develop by google platform Kubernetes help to manager containerized application in different machine or node or platform or cloud machine.

Day 8

13-02-2022

Cloud computing Introduction

Cloud computing provides us a means by which we can access the application as utilities over the internet. It allow us to create, configure and customize the application.

With the cloud computing user or developer or programmer can access database or software or tools or OS or sever resources via internet from anywhere for long as they need without worrying about their maintenance.

Cloud computing is both a combination of software and hardware based computing resources delivered as a network service.

Private cloud : within an organization.

Public cloud : any one can access. AWS or Google or Oracle

Hybrid cloud : combination of private and public

Community cloud : it support by more than one organization.

Iaas : Infrastructure as a Service : OpSource At & T

Paas : Platform as a Service : Aws or Azure

Saas : Software as a Service : sale forces and google

Advantage

1. Lower computer cost
2. Reduce the software cost
3. Pay base upon the usages.
4. Scale on demand.
5. Unlimited storage

Etc

Disadvantage

1. Security issues
2. Can be slow.
3. Require constant high internet speed.

AWS : Amazon Web Service : it is type Paas cloud service. Which provide n number of service.

Azure

Google

Oracle

Etc

AWS : 1 year account for Student.

Etc

S3

EC2

Amazon S3 (Simple Storage Service). It is designed to make web – scale computing easier for developer to share the data through internet.

It is like google drive.

It gives any developer access the same highly scalable, reliable, fast and secure environment to share the data.

Amazon EC2 (Amazon Elastic Compute Cloud). This service is use to create server machine which help to install the software and deploy the application.

sudo yum install git –y

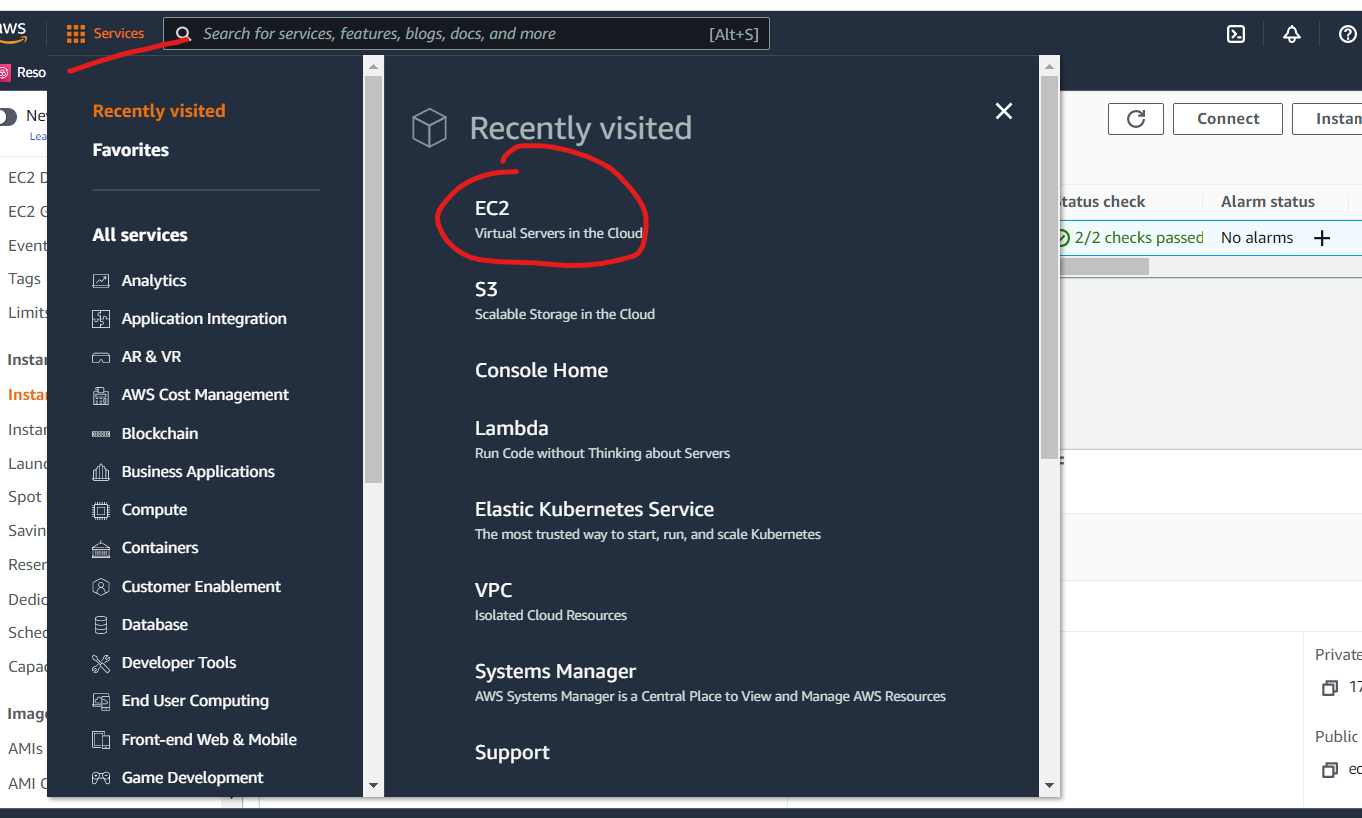
if we run the project in standalone machine

to access express js application we use

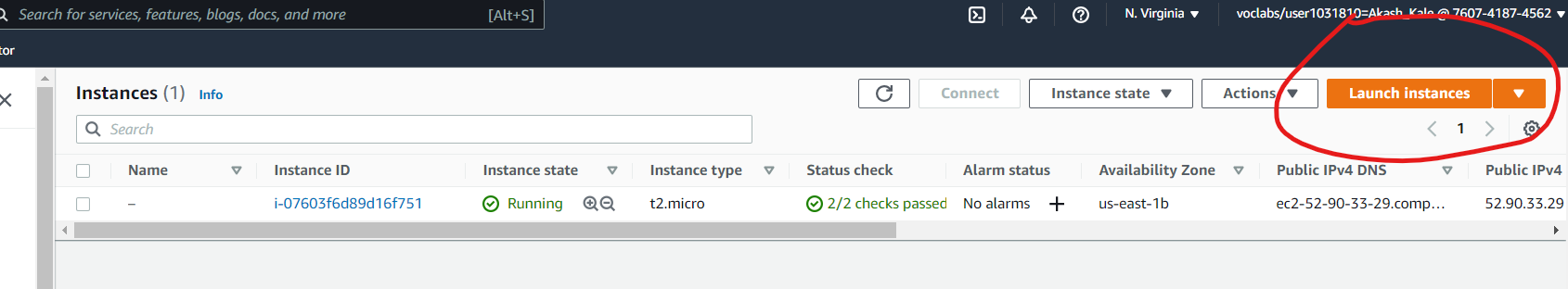
<http://localhost:9090>

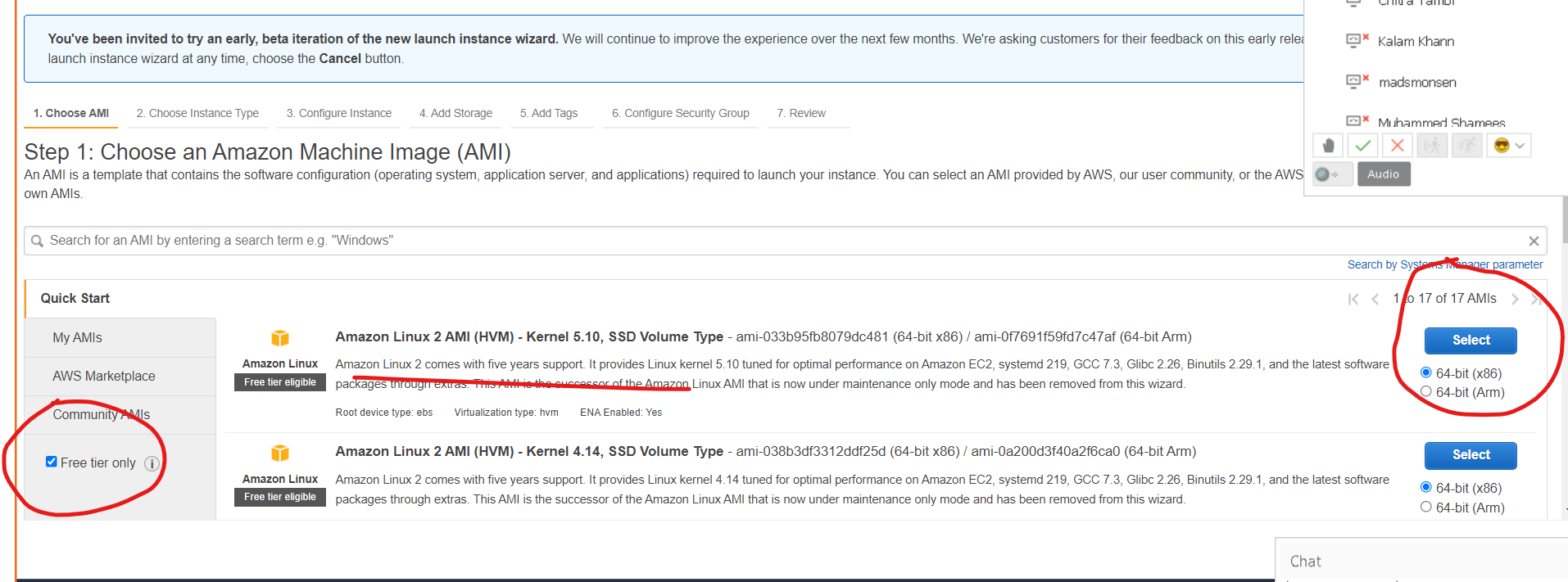
Creating EC2 instance in AWS

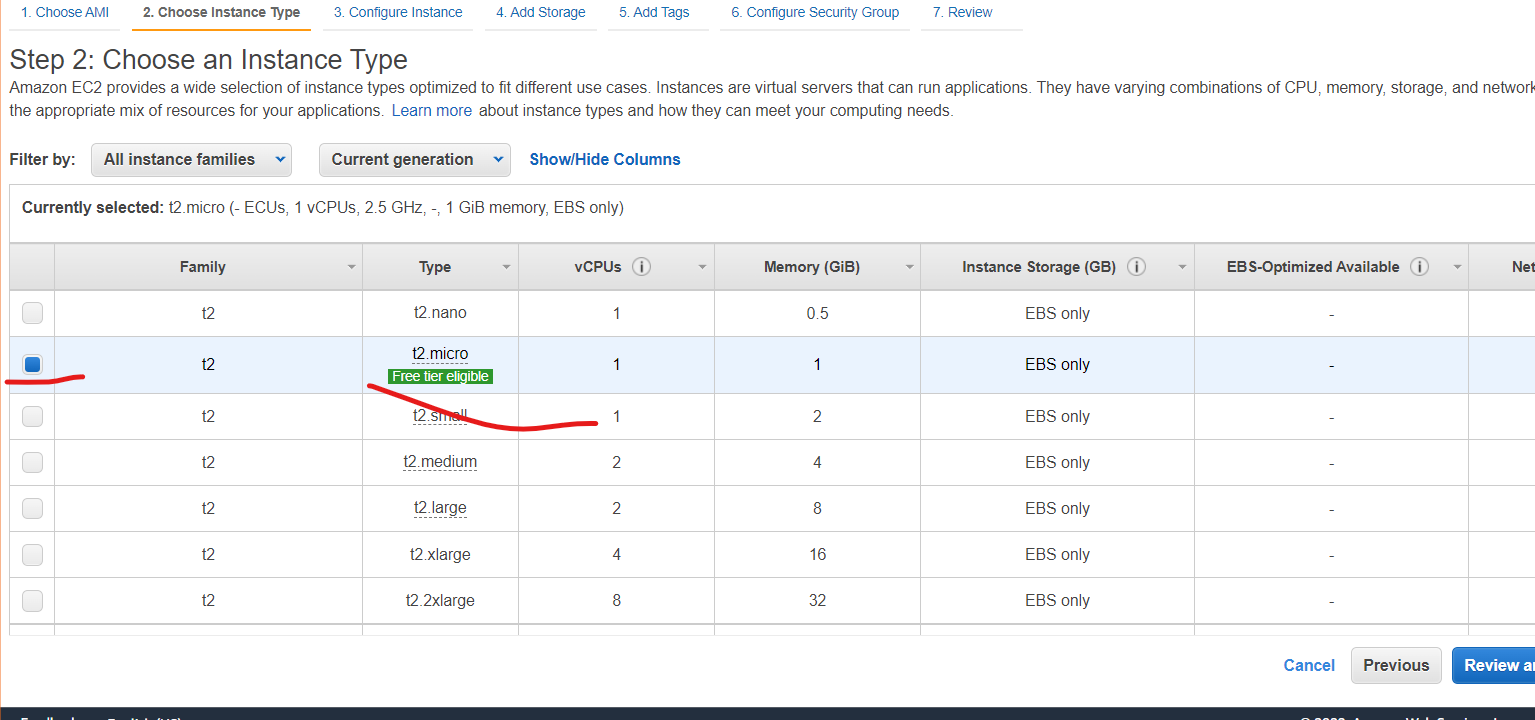
First logic in AWS account



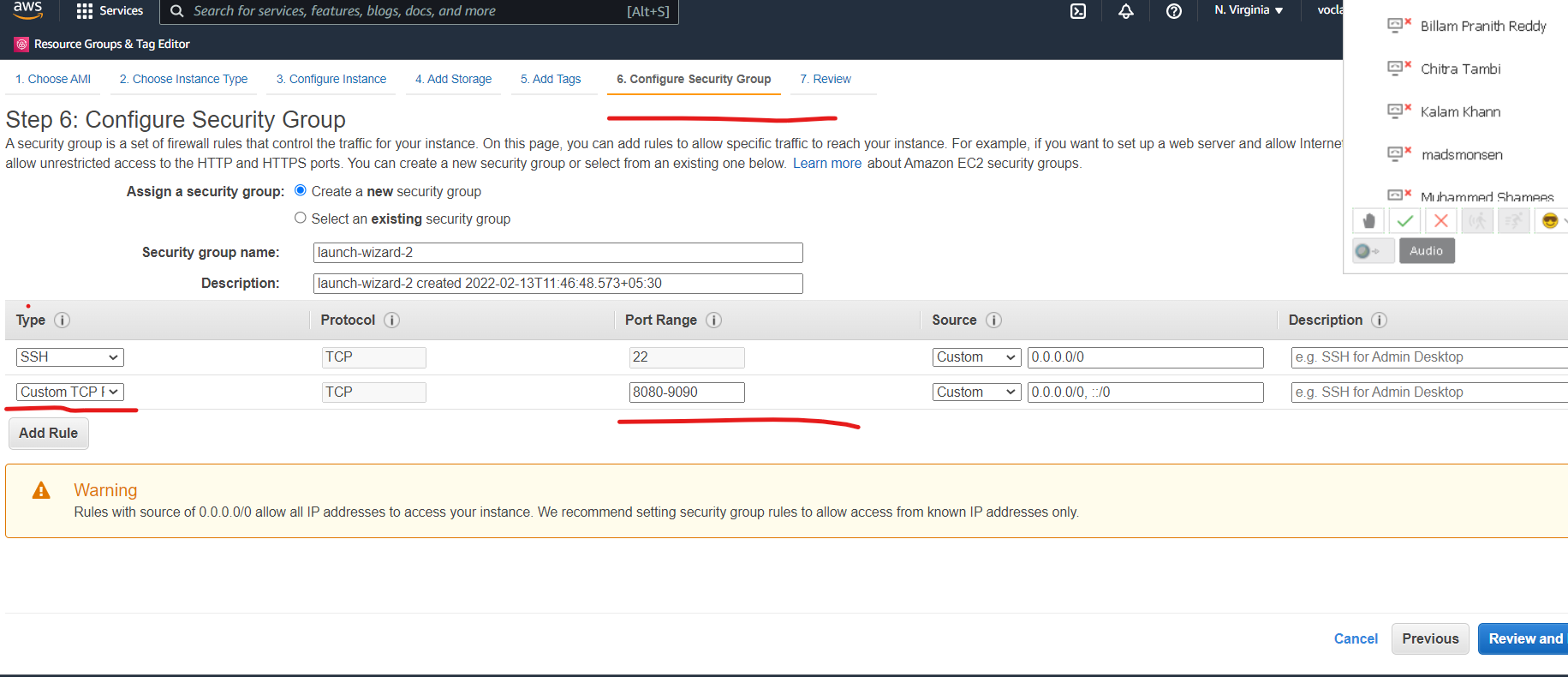
Click on launch instance





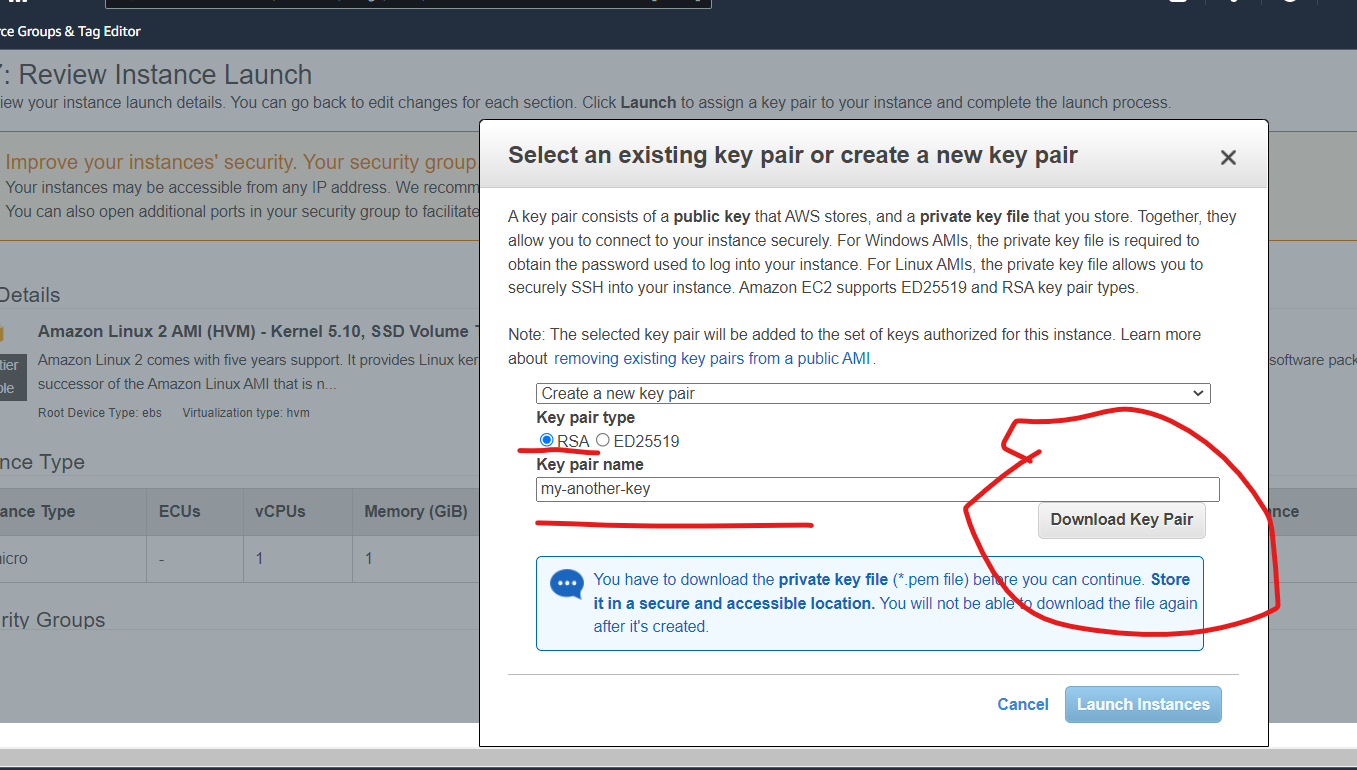


Port number range 8080-9090 : so you can run the application between this range.



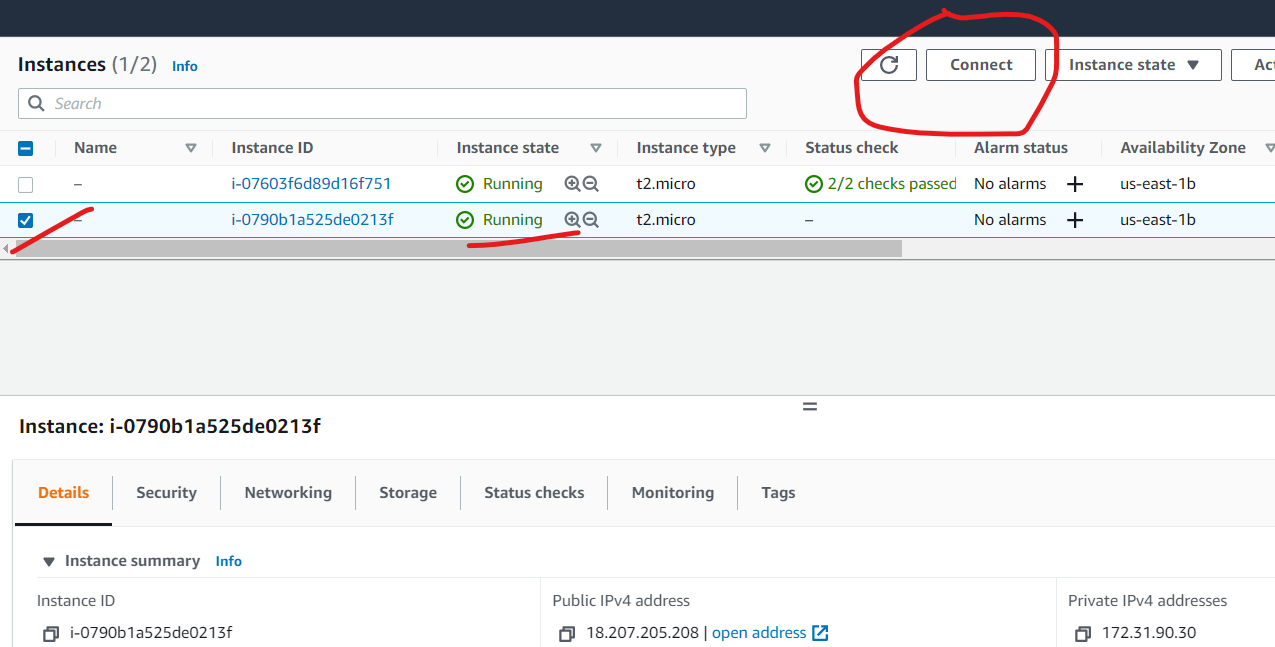
Then click review and launch

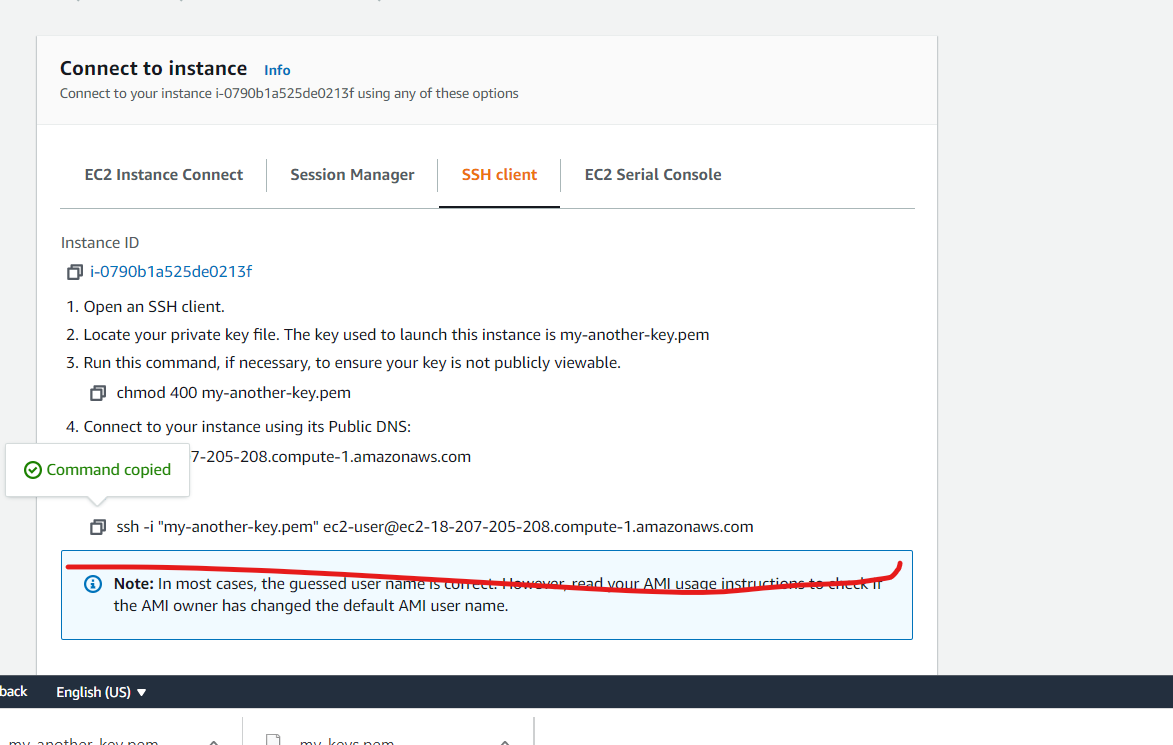
Then create PEM key (private key to access EC2 instance)



Then launch instance

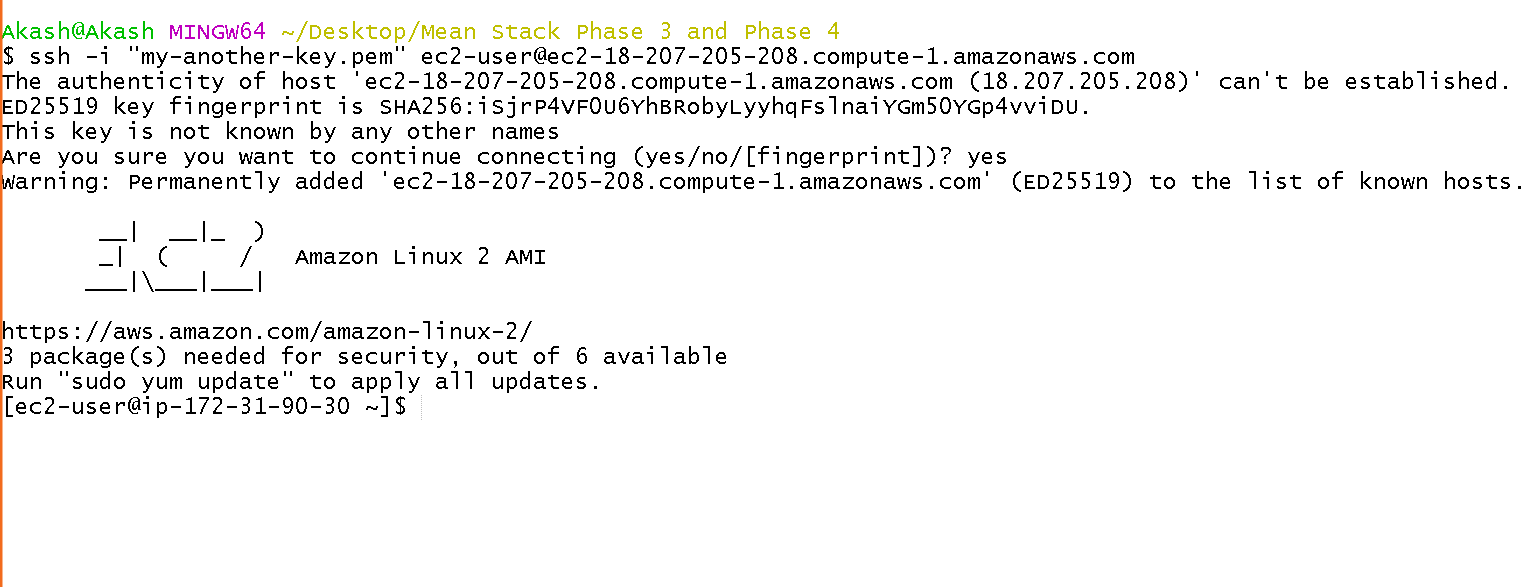
Wait for few second to get the instance to up





Open the git bash terminal in place where the pem file downloaded and paste the command ie ssh to connect Virtual server using git bash terminal





To update the EC2 instance run sudo yum update

Then in Ec2 terminal install the necessary software depending upon our requirement.

First we will create the project in local machine

Build the project

Using nginx server create the image

1st option

Publish this image in docker hub

In EC2 instance we will install docker

And pull image and run this image.

2nd option

Push image file in git

Clone from git

Install docker in Ec2 instance

Build the image and run it.

First create the project in local machine and do the changes according your requirement

Then build the project using command as ng build

Inside build file create Dockerfile

FROM nginx:latest

COPY . /usr/share/nginx/html

Then push the dist folder into git repository

Inside dist folder

git init

git add .

git commit –m “done”

Then create remote repository

Connect local repository to remote repository and push the code to remote.

git remote add origin https://github.com/Kaleakash/angular-build-on-ec2.git

git push -u origin HEAD

Then in EC2 instance pull the project

Then install docker in EC2 instance using command

As

install docker

sudo yum install docker

sudo service docker start

sudo docker info

sudo docker --version (in ES2 all docker command start with sudo command)

Then build the image

Sudo docker build –t my-angular . –f Dockerfile

sudo docker images

Then run the image

sudo docker run -d -p 80:80 my-angular