Day 1 :

26/08/2023

SDLC : Software Development Life cycle

Water fall model

Requirement gathering 6 months

Plan

Design

Coding / testing

Build the project development mode

Deploy the project production mode

Provide the service

Increment model

V model

Agile model

Sprint : time duration to develop small module 1 or 2 week. Login page, feedback page

Dashboard

DevOps : Development and Operation

Development people develop the application using any language like java, python, etc

Operation team interact with customer or end user client and they are responsible to maintain the application.

Admin

Developer

Tester

Admin

Database designer

Architecture etc

Devops tools

Git and git hub

Maven and Gradle (Java development)

CI and CD tool using Jenkin (Continuous Integration and Continuous deployment or delivery)

Selenium tool (testing)

Ansible tool (monitoring tool as well as configuration tool)

Docker container

Nagios tool

Kubernetes

Git and github

Sub version control which help to record the application flow.

Dev1	login page	
Dev2	java or python code	merge the code
Dev3	database	

Unix commands

ls : this command display all files and folder present in current directory

pwd : print working directory : it show current path of terminal or command prompt.

mkdir foldername: this command is use to create the folder

cd foldername : move inside a folder

cd .. : come outside a folder or move to parent directory of current folder

rmdir foldername : to remove folder

touch filename : to create empty file

vi filename : open the file in vi mode

once open hit i key to move inside a vi editor mode

write the contents

esc : to come out from editor mode to normal mode

:wq : write and q quite (save and exit)

cat filename : it is use to read contents from a file

cat means concatenate

Git : Git is version control system which help to track or record changes done in the application or project or app.

Git also known a distributed sub version control.

First create the folder

Then create the file and write the contents

git --version

git init it is use to make local folder as git repository

init command create .git folder insider that current directory

ls -a : it display all files and folder with hidden folder.

git status : this command is use to check the current status of your repository

git add filename : to add files or folder normal local folder to stagging area.

Or

git add . : this command is use to add all files and folder present in

Current directory.

git commit -m “message” : this command use to pass the task from stagging area to

local repository.

git config --global user.email “akash300383@gmail.com”

git config --global user.name "akash"

steps

1. Create folder with any name ie Demo
2. Then create the file with any name ie test and write the contents insider that file.
3. Then open the terminal inside that folder please use pwd
4. git init
5. git status
6. git add .
7. git status
8. git commit -m “message”
9. first time we need to set config details as emailed and name
10. git config --global user.email “akash300383@gmail.com”
11. git config --global user.name "akash"
12. Then please commit using command as git commit -m “done”
13. git status

git branch : branch is like a pointer which hold more than one commit details.

By default git provide default branch. Default branch name may be master or main.

If we want to check branch details present in local repository

git branch

command to create user defined branch

git branch branchname	this command is use to create user defined branch
git checkout branchname	this command is use to switch from one branch to another branch.

Current branch is master or any other branch

git merge branchname	this command add all task in current branch

git branch -D branchname this command is use to remove the branch

Demo.java

int a; akash branch

int b; Vikash branch

Remote repository help us to share the code between two or more than one tabme.

Git hub

Git lab

Bitbucket

Aws

Azure

Private cloud etc

git hub : it is a type of remote repository provided by micro soft organization.

Day 2 :

27/08/2023

We want to connect local repository with remote repository

1. Token base authentication
2. SSH Client
   More

   To connect local repository with remote repository

   git remote add origin URL

git remote add origin https://github.com/Kaleakash/test_rep.git

git remote add origin https://token@github.com/Kaleakash/test_rep.git

git push -u origin main (it is use to push the code)

how to resolve the conflict

1. first create Repo2 folder
2. then open terminal inside that folder
3. create sample file
4. add some data 1st, 2nd
5. using git init make folder as repository
6. git add .
7. git commit -m “done changes in master branch”
8. create the branch
9. git branch akash
10. git checkout akash
11. in akash branch we will add 3rd and 4th message.
12. Then git add .
13. Then git commit -m “in akash branch done some changes in sample file”
14. Create another branch with name as Vikash

15. git checkout -b Vikash (it will create the branch and switch to that branch)
16. in sample file in Vikash we will add the message as 5th and 6th.
17. Then git add .
18. Git commit -m “done change in sample file by Vikash branch”
19. Please move the master branch ie git chechout master.
20. Please verify current branch using command as git branch
21. Then in master branch merge the code from akash branch
22. Git merge akash
23. Using cat sample read the data from sample file
24. Out must be 1st, 2nd, 3rd, 4th
25.

Download or clone the repository

1. Create the folder with any name ie devopstrainig In VM
2. Then open the terminal
3. git clone URL
4. git clone https://github.com/Kaleakash/devops_aug_2023_trainig_batch.git
5. use ls command to see downloaded folder
6. using cd command please move inside that folder.
7. cd devops_aug_2023_trainig_batch
8. ls command to see the more than one file.

if we do any changes in local repository
we need to add, commit and push

```
git add .
git commit -m “done some changes in file”
git push -u origin main
```

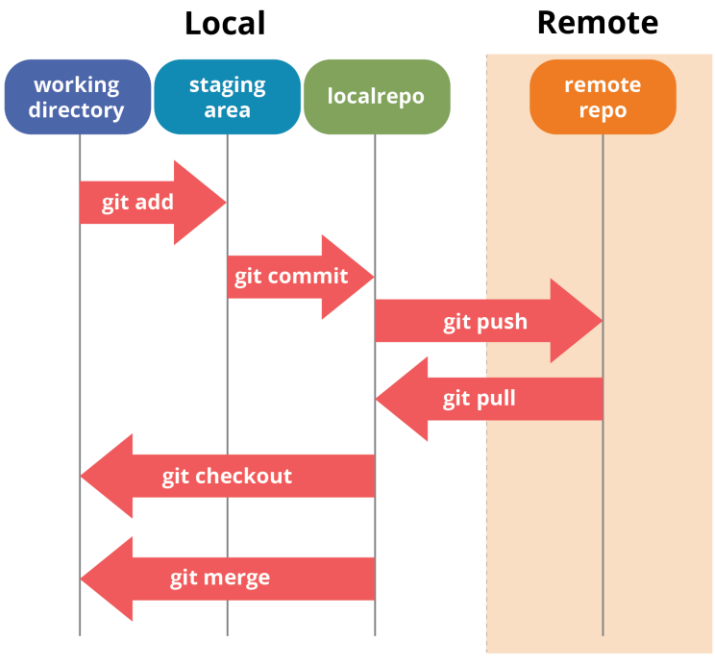
1. open the terminal inside a repository folder
2. make sure .git folder present using command verify ls -a
3. git pull

git clone URL : it help to download fresh repository in local machine

git pull : it will pull new updated from remote to local repository

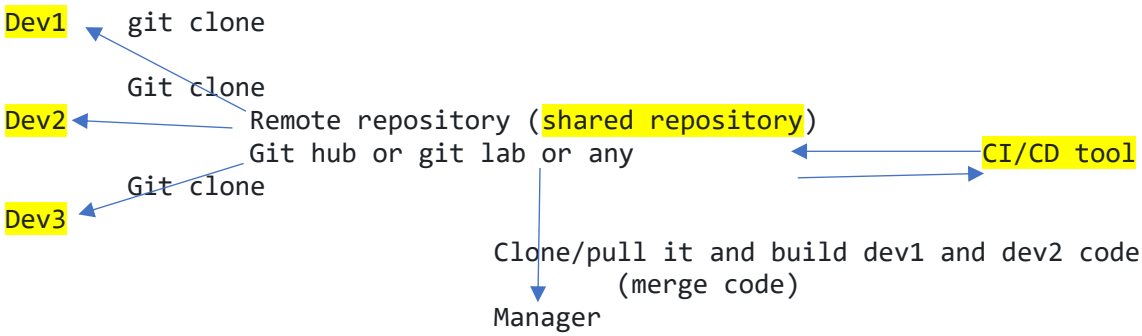
Git Vs Git hub

Git is Remote repository provided by micro soft.
Command line or GUI tool which
Help to interact with any remote repository



CI and CD tools

Continuous Integration and Continuous delivery or deployment



java	Git hub	Jenkin
python	Bit bucket	GoCD
angular		

Generally shared repository can be private or public.
While creating or after created that repository we will send the invitation
To all developer to join that repository.

Default branch can be master or main.
Dev1 need to create login page using html and css

Dev2 need to create java or python code
Dev3 need to create database tables.
If they do all their task in main or master branch after changes done they can add, commit and push the code to shared repository.

Don’t do your task in main or master branch. While doing your task please create user defined branch and push that branch in remote repository.

In remote repository we will check user defined branch if code is correct we will merge that code into master or main branch.

Build phase :

You need to compile and run the application using that language.

Javac

Java	java
Py	python
Ng	angular

Dev1 push the code in remote repository

In remote repository we need to verify the code and merge the code. And we need build the application.

After dev2 or dev3 code merge in master or main branch we can build successfully or it generate some error.

CI and CD tools.

Jenkin : it is a type of CI and CD tools. It is an open source ci/cd tool base upon java technologies. Plugin base ci and cd tool. GUI based tool.

Open the browser

http://localhost:8080

it will ask login details

username : admin

password : admin

in Jenkin we need to create the job. Every job responsible to build the project.

Day 3 :

02/09/2023

Open the terminal in VM.

git clone URL

next

git pull (but make sure terminal open inside that folder).

Web Service : Giving the service for web application when both the application running using different technologies

API : Application Programming interface.

Web Service

Java python

XML/JSON

eXtensible markup language

JSON : JavaScript object notation

HDFC XML/JSON HSBC

Micro service :

Please refer the website the set the trigger time using crons

https://crontab.guru/

Day 4 :

03/09/2023

Open the terminal and start /stop jenkin service

sudo service jenkins stop unix

sudo service jenkins start start

user name : admin

password : admin

java -jar jenkins.war

http://localhost:8080

Jenkin provide few pre installed tools like Maven, Gradel, Git, ant etc. These tools help us to run java technologies.

Jenkin Pipeline a pipeline is a collection of event or job which interconnected with each other to perform a specific task.

Maven goal

Clean

Compile

Install jar or war unit test

Test

Package

```
pipeline {
   agent any
   stages {
      stage('Hello') {
         steps {
            echo 'Hello World'
            sh ‘git --version’
           sh ‘v’
         }
      }
   }
}
```

Maven : Maven is open source build tool we use in java technologies to build the project.

Maven goal

Clean : clean complete project doesn’t matter it contains one file or more than one file

Compile

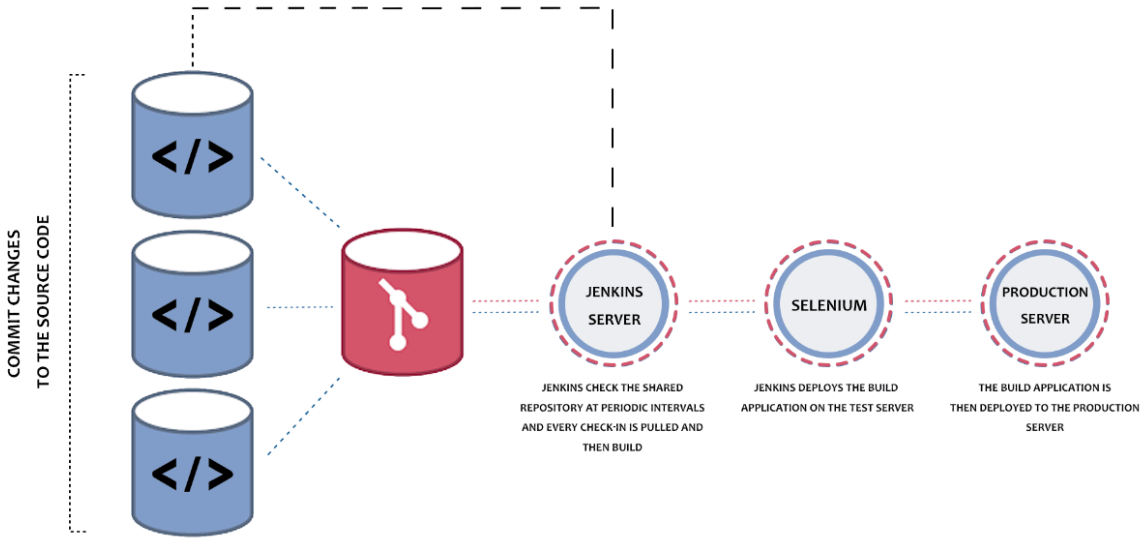
Install : install some dependencies

Test : test the project

Package : creating jar or war file

```
pipeline {
    agent any
    tools {
        // Install the Maven version configured as "M3" and add it to the path.
        maven "M3"
    }

    stages {
        stage('Build') {
            steps {
                // Get some code from a GitHub repository
                git 'https://github.com/jglick/simple-maven-project-with-tests.git'

                // Run Maven on a Unix agent.
                sh "mvn -Dmaven.test.failure.ignore=true clean package"

                // To run Maven on a Windows agent, use
                //bat "mvn -Dmaven.test.failure.ignore=true clean package"
            }
            post {
                // If Maven was able to run the tests, even if some of the test
                // failed, record the test results and archive the jar file.
                success {
                    junit '**/target/surefire-reports/TEST-*.xml'
                    archiveArtifacts 'target/*.jar'
                }
```

```
        }
      }
    }
}
```



If we want to run more than one command with help of normal job.

Day 5 :

Install python in VM

sudo apt-get install python3

sudo apt-get install python3-pytest

py ops.py

py *.py

or

python3 ops.py

https://github.com/Kaleakash/python_jenkins_file.git

git URL which contains jenkin pipe line script to build python program as well as

run python program.

Post build

We can send notification through email

We can push this project in production environment.

We can push this project to testing environment ie Selenium

Build can be success or failure.

Docker

Few we to run any application or server or tools.

We need system software. OS ie window, liux, Unix or Mac etc.

Server name : tomcat, IIS, nginx , apache, web logic, jboss, WAS etc

Database server : mysql, oracle, db2 RDBMS

Mongo db, HBase, Neo4j no SQL etc

Tools : SAP, Info metica, IIB, ESB, Portal server etc.

VM : Virtual Machine etc.

VMWare .ios

VMWare help use to do Virtualization.

Guest OS ; it can be unix, linux, window XP etc

Limitation of VMWare or Virtualization

Base machine is Window 11 : with RAM 16 hard disk 1tb

If I want to ru Guest OS with the help of VMWare software

Window XP --→boot up the window XP, we need provide RAM 4 gb and external hard disk

50gb. We need share the base machine resources. Etc.

We want to run 10 VM

Docker : Docker is an advanced OS Virtualization software platform which makes it easy to create, deploy and run the application in Docker container.

Container : run time environment or engine.

JRE : Java Run environment

Node JS : JavaScript run time environment.

Web Container : web container provide run time environment to run the application.

Database Container : it help to store the data in table format.

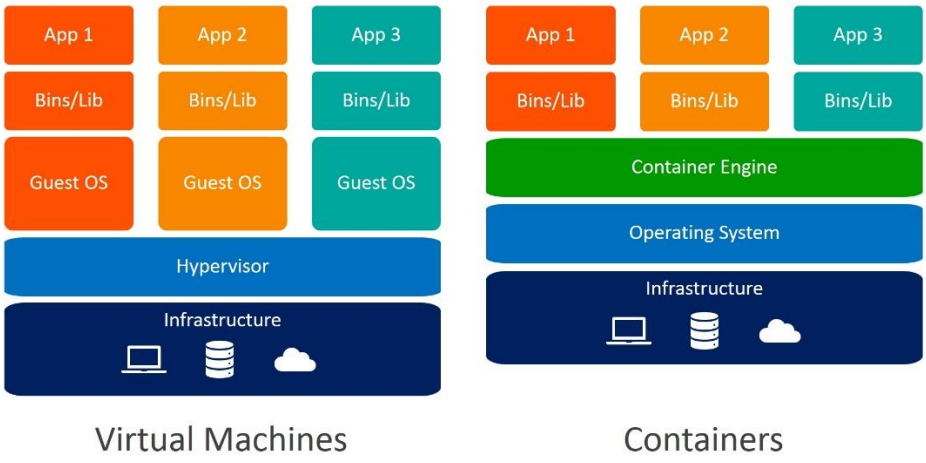
Docker Container : it is unit of deployment or software. Which contains everything to run the application. Ie code, runtime (software), tool and system libraries or database or server etc.

Docker is use to create Containerization application

Virtualization Vs Containerization

Virtualization is an abstract version of physical machine or OS or Guest OS.

Containerization is the abstract version of an application or server or tool or etc.



But run Docker in base machine we require docker engine.

Mkdir

Open the Terminal and

Write the command as

docker --version this command provide docker version

docker info this command provide docker details

docker pull imageName : this command is use to pull the image from docker hub to local machine.

docker images

docker pull hello-word this command is use to pull the image

docker run imageName/imageId : this command is use to run the application using docker image.

Docker pull the image by default from Docker.hub

Docker hub is an open source remote repository which contains lot of images

Which we can pull as well as push.

Docker hub provide private as well as public remote repository.

Docker hub is like a git hub.

In Git hub we can push any types of file or folder.

Docker hub contains docker images which is responsible to run the application

Using container.

Like other repository provided by AWS or Azure or Google cloud or private cloud etc.

Please docker hub account.

Dockerfile : A Docker file is a blue print or set of instruction that defines

How our images is build. Or Docker file use to create the image.

Docker image : Docker images contains everything to run the application.

Or

Doker image are the source code for your containers.

Using docker file we can create the image

Docker Container : instance of images or running process etc.

Once you run the image the running container become up and it will run that application which we

Mention in docker file which create image.

docker ps	this command is use to display running container
docker ps -a	this command is use to display all container ie running as well as stopped mode.

docker run --name c-container hello-world this command is use

run the container with specific name or custom name

Day 6 :

10/09/2023

```
File  Edit  View  Search  Terminal  Help
akash300383gmai@ip-172-31-16-158:~/Desktop$ docker --version
Docker version 20.10.12, build e91ed57
akash300383gmai@ip-172-31-16-158:~/Desktop$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
hello-world   latest    9c7a54a9a43c   4 months ago   13.3kB
akash300383gmai@ip-172-31-16-158:~/Desktop$ docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
Digest: sha256:dcba6daec718f547568c562956fa47e1b03673dd010fe6ee58ca806767031d1c
Status: Image is up to date for hello-world:latest
docker.io/library/hello-world:latest
akash300383gmai@ip-172-31-16-158:~/Desktop$ docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
```

Creating images

1. creating simple image to display welcome message.

Dockerfile

FROM busybox

CMD ["echo","Welcome docker! This image created by akash"]

docker build -t my-bosybox . -f Dockerfile

docker images

docker run bosybox

-t : tag

-f : file

2. image to run console base application

```
a=10
b=20
sum=a+b
print("sum is",sum)
```

Then create the Dockerfile

```
FROM python:3
ADD myfile.py /
RUN pip install pystrich
CMD ["python","./myfile.py"]
```

1st file pull python image with version 3

2nd add myfile in python image

3rd if required install some plugin base upon requirement.

4th open command prompt and run the python program

then create the image

```
docker build -t my-python . -f Dockerfile
```

docker image

```
docker run my-python
```

Web Application Image

Html, css, JavaScript, typescript, angular or react js or jquery etc.

Html etc.

Html is use to create the web page or web application.

First create the image for web application and inside the folder create the file with

Below code

```
<html>
    <head>

    </head>
    <body>
        <h1>Welcome to my web page created by Akash Kale!</h1>
    </body>

</html>
```

save the file with name as index.html

and run or open this file in chrome browser.

http://www.google.com	: production env or live server
http://localhost:8080	run on local machine with local server dev env
or	
http://127.0.0.1:8080	default ip for every machine.

To run the web application we need server

Tomcat

Apache

IIS

Nginx

Etc

We use nginx open source server to create image for web application.

Server always run on port number :

Tomcat	8080
Nginx	80
MySQL Database	3306
Jenkins	8080

Etc

If image contains web application with run server then we need to use the command as

docker run -d -p 80:80 imageName/imageId

right side port actual port number 80

publish port number can be same or different 80

-d : background or detached mode

-p : publish

index.html

```
<html>
    <head>

    </head>
    <body>
        <h1>Welcome to my web page created by Akash Kale!</h1>
    </body>

</html>
```

save the file with name as index.html

and run or open this file in chrome browser.

Dockerfile

```
FROM nginx:1.6
COPY index.html /usr/share/nginx/html
```

docker build -t my-web . -f Dockerfile

docker run -d -p 80:80 my-web

using docker ps

if run please open browser and type as http://localhost:80

docker stop containerId/containerName stop container

docker start containerId/containerName start container

docker rm containerId/containerName remove container but first stop then remove

docker rm containerId/containerName -f without stop we can remove

docker rmi imageName/imageId if image is not link with any container or running container we can remove if we get error please first remove that container and then remove image

docker rmi imageName/imageId -f

now we will publish our local image into remote repository ie docker hub

docker login

it will ask your docker hub account

id and password

my-web

docker tag imageName dockerHubAccountId/imageName:version

version is like a tag

docker tag my-web akashkale/my-web:1.0

after created tag now you can push the image

docker push dockerHubAccount/imageName:tag

docker push akashkale/my-web:1.0

Application

Docker image

To run html page

Docker image

run my spring or python program,

REST API

frontend technology ⟷ backend technology

http://178.0.0.:80

html, css, js, typescript

angular or react or vue js

java (spring boot)

asp.net

python

node with express js

etc

store the data in file system or database ie mysql or oracle

we want store and retrieve the data from database. mysql database

Front end	backend	database
Angular	java	mysql
React	php	mysql

Network network

Public private

Public -→ frontend and backend container

Private → backend and database container

One image is responsible to run one application or modules (micro service)

Front end	backend	database
Image	image	image
Container1	Container2	Container3

http: TCP

docker compose and Docker swarm and Kubernetes

docker compose it a toolkit which help to run more than one container with help of yaml or yml etc.

Docker compose, Docker Swarm and Kubernetes are responsible to run more than one container.

Those container execute independently as well as they can communicate with each others base upon

Their requirements.

Day 7 :

16/09/2023

Docker compose it a tool kit which is responsible to run more than one container using configuration file .yml or .yaml

docker-compose build it build custom images

docker images

docker-compose up pull pre defined images if required

and run all images part of that docker compose file

docker compose up --build -d

it will build it and run in background ie detached mode.

please open another terminal

docker ps

please verify all three container running or not.

docker network ls it is use to verify all network

```
version: '3.3'
services:
  my-first-container:
    image: nginx
    ports:
      - 80:80
  my-another-container:
    image: akashkale/my-web:1.0
    ports:
      - 81:80
```

```
docker-compose build
docker-compose up
or
 docker-compose up --build -d                    detached mode
docker-compose down
```

Docker Swarm or Docker Kubernetes

Docker compose is use to run more than one container and all container must be running in same node / same machine ie desktop or cloud machine.

Node word refer to physical machine or cloud machine or device.

Atul if we use different machine then we need to use different docker compose.

All machine are connected using ip address.

Front end Machine	backend machine	database machine
Container	container	container

If number of client increase to access the application may be front end or backend or database.

We need up scale up upon on demand.

Kubernetes : Kubernetes is container management tool or K8S. It is also known as orchestration tool

Orchestration tool is responsible to deploying more than one container, scheduling, scaling and load balancing, configuration etc Etc.

Kubernetes is responsible to maintain more than one container those container can be run in same machine or difference machine ie node.

Docker Swarm Vs Kubernetes

Docker Swarm is part of Docker

1. No auto scaling
2. Does auto load balancing
3. Easy to develop the application
4. No GUI

Kubernetes is part of google

1. Auto scaling (up and down on demand)
2. We can do manually auto load balancing
3. Complicate to do configuration.
4. We can use GUI base upon tools.

Node : Node refer to machine or device or physical machine or cloud machine.

Cluster : it is a collection of host or combination of node (server or client). That helps you to aggregate their availability of resources. Like RAM, CPU, Disk, pool etc.

Public and private id address

http://198.78.56.45:80 public outside a cluster.

http://198.1.2.56 private

http://198.1.2.57

http://198.1.2.58

http://198.1.2.59

http://198.78.56.45:80

Namespace : it is a logical cluster or environment. Namespace is like package. It is widely used method which is scoping or dividing a cluster.

Java application → java-app

Python application → python-app

Angular application → angular -app

Pods : Kubernetes is responsible to run more than one container. In Kubernetes container can't communicate with each other directly. All container must be wrap in a functional unit and that unit is known as pods.

Each pods are responsible to run one container or more than one container.

Node contains more than one pods. Each pods can contains more than one container and each container responsible to run one application.

By default, the pods in only accessible by its internal IP address within a cluster.

to communicate more than one pods within a cluster we need service.

Service helps us to expose container from pods

Project

We can create one cluster or more than one cluster

Inside each cluster we can add one or more than one node.(machine).

Each node contains one or more than one pods. Those pods are part of same namespace or different namespace.

Each pods contains more than one container. And each container responsible to run the application. That application can be java, python, php or node js.

Kubeadm : tools provided by Kubernetes which help to develop Kubernetes application

Unix or Linux non window. Kubeadm support cluster features.

Minikube in your local machine.

Minikube ie open source tools which provide single cluster environment for Kubernetes

To deploy the application.

It is GUI base.

kubeadm

kind

Docker desktop

Kubectl : is a command line interface which help to interact with Kubernetes

Cluster

Private cloud provider provide Kubernetes cluster.

Service

Day 8 :

17/09/2023

Please Pull repository in your local machine or VM if Git present or download

https://github.com/Kaleakash/docker-compose-repository.git

then open the terminal

docker-compose --version

docker-compose up --build -d

docker images

docker ps

after running on container

open the browser

http://localhost:81

then application open

store the data

if you want to stop

docker-compose down

docker run -it alpine

apk add openjdk11

apk add git

git clone https://github.com/Kaleakash/jenkinjava.git

cd jenkinjava

javac Demo.java

java Demo

Dockerfile

docker run -it alpine size

apk add openjdk11 size

apk add git size

git clone https://github.com/Kaleakash/jenkinjava.git

cd jenkinjava

javac Demo.java

java Demo

Dockerfile

FROM openjdk:8

COPY Demo.java .

RUN javac Demo.java

CMD [“java”,”Demo”]

Git --→ Git Hub -→ Jenkin -→

In Jenkin We can run docker image (Jenkin Pipe Line)

VSCode editor

https://code.visualstudio.com/download

Please create Jenkin pipeline job

And provide Git URL of my project

https://github.com/Kaleakash/docker-compose-repository.git

docker-compose up --build -d

but make sure docker running as well as docker-compose running.

Or

Create index.html

Create the image for index.html

Please create docker compose file to create image and run the container.

Push this code in git hub

Then create jenkin job or pipeline with trigger and run dockerc-compose file in jenkin environment

Then in Jenkin we need to configure authentication details for Docker hub.

AWS we need to create three instance

All required software we need to install.

Instance type must be medium 2 CPU

Master Node

Worker1 node

Worker2 node

Day 9:

Installed docker

Installed minikube local machine

We need to start minikube start it will download all required images

and start the container.

To start minikube we required minimum

CPUs=2, Memory=4000MB

docker image

docker ps

to open the minikube dashboard

minikube dashboard

kubectl it is a command line interface which help to interact with cluster.

kubectl cluster-info

```
akashkale/my-simple-kuberneties:tagname:1.0
```

using these 2 ways we can deploy our application in cluster environment.

using imperative command

using declarative command

kubectl create deployment my-app --image= akashkale/my-simple-kuberneties:1.0

kubectl get deployment

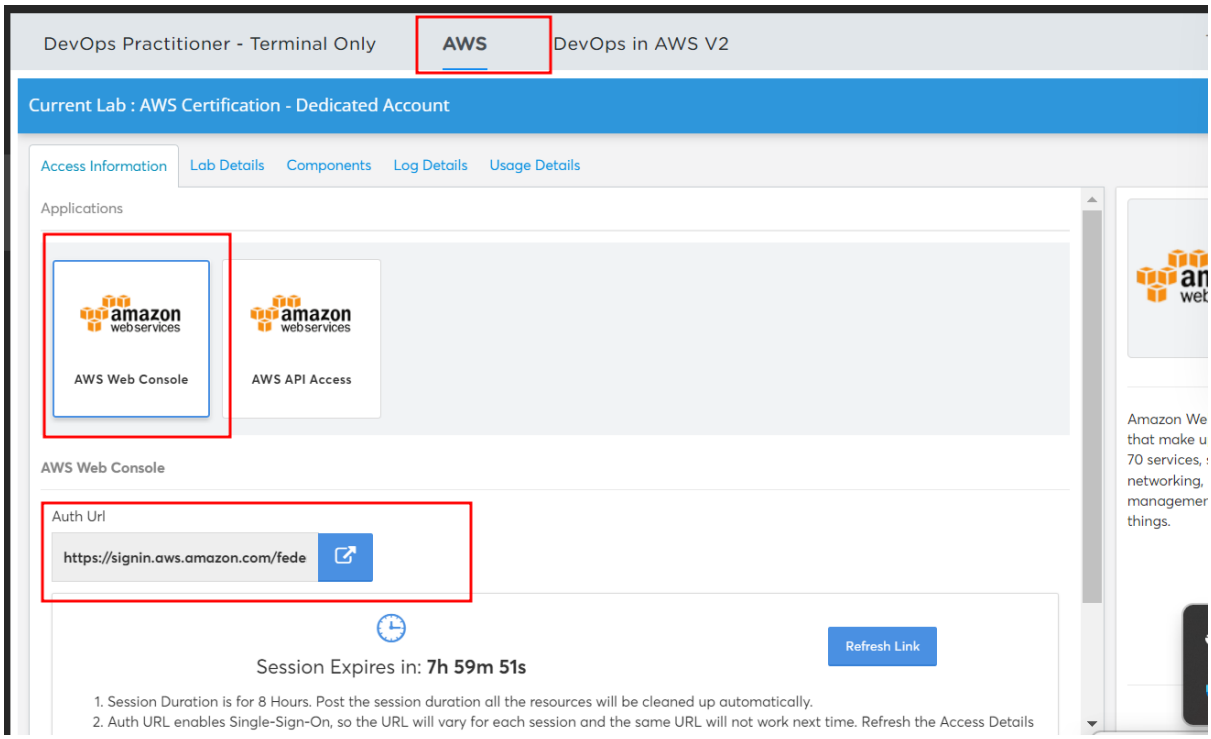
kubectl get pods

kubectl expose deployment my-app –type=LoadBalancer --port=80

kubectl get service

minikube service my-app

creating instance in AWS



EC2 instance

(Amazon Elastic Compute Cloud)

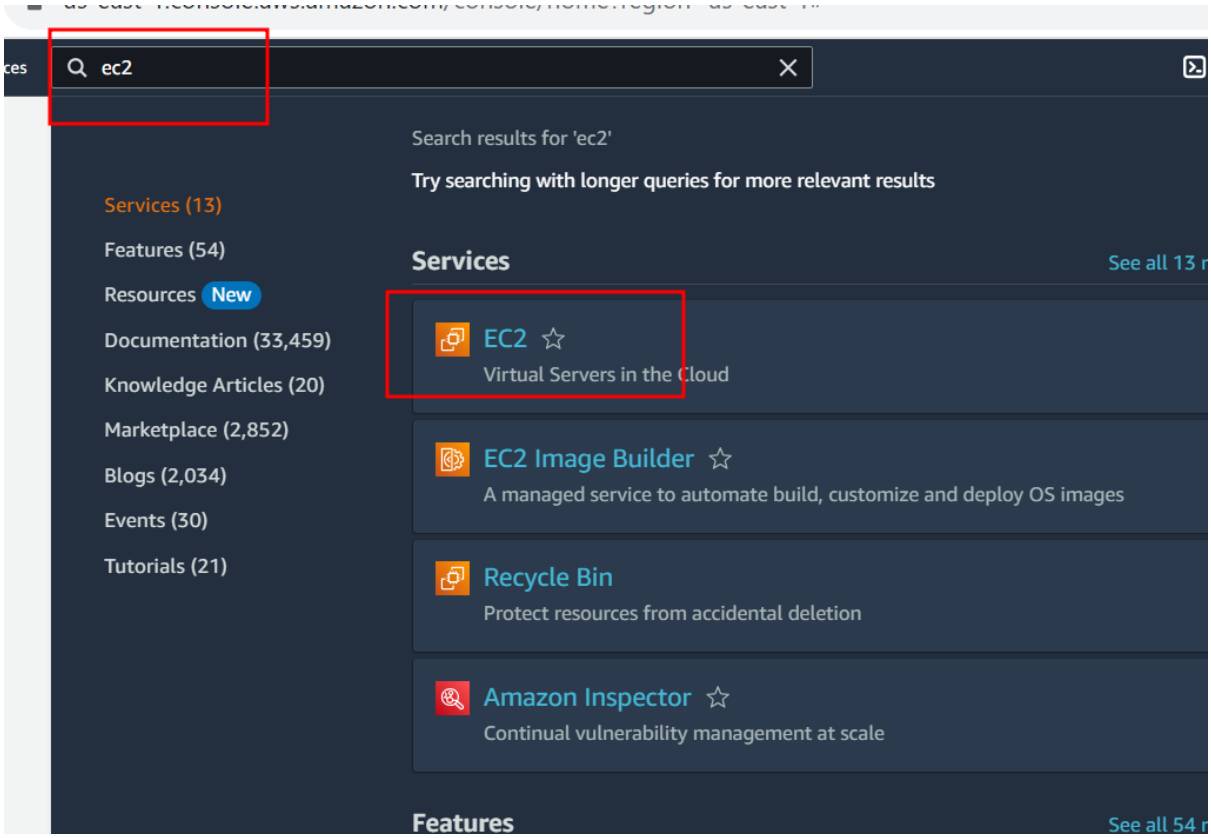
It is use to create Virtual Lab machine using any OS.

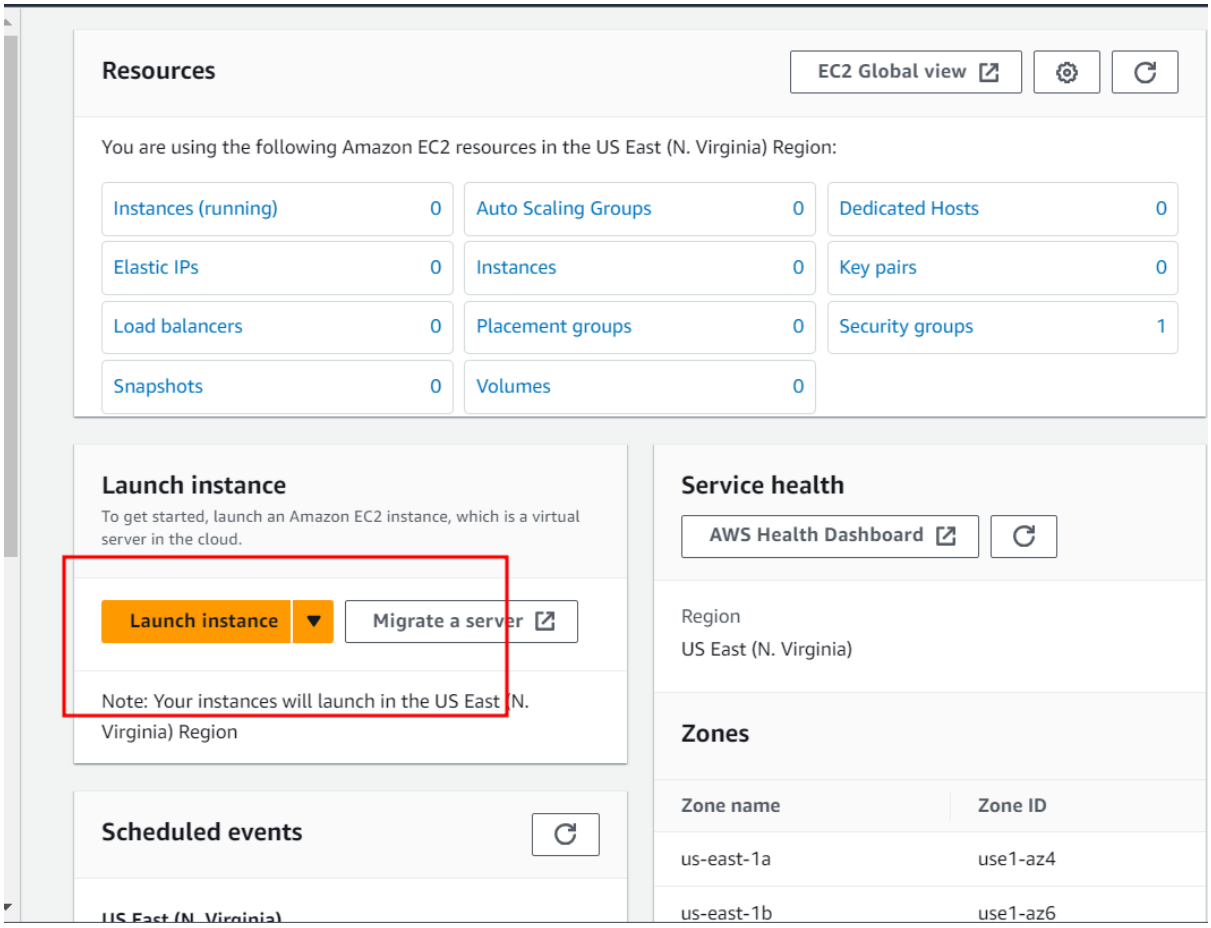
To connect that OS we can use command prompt or GUI application.

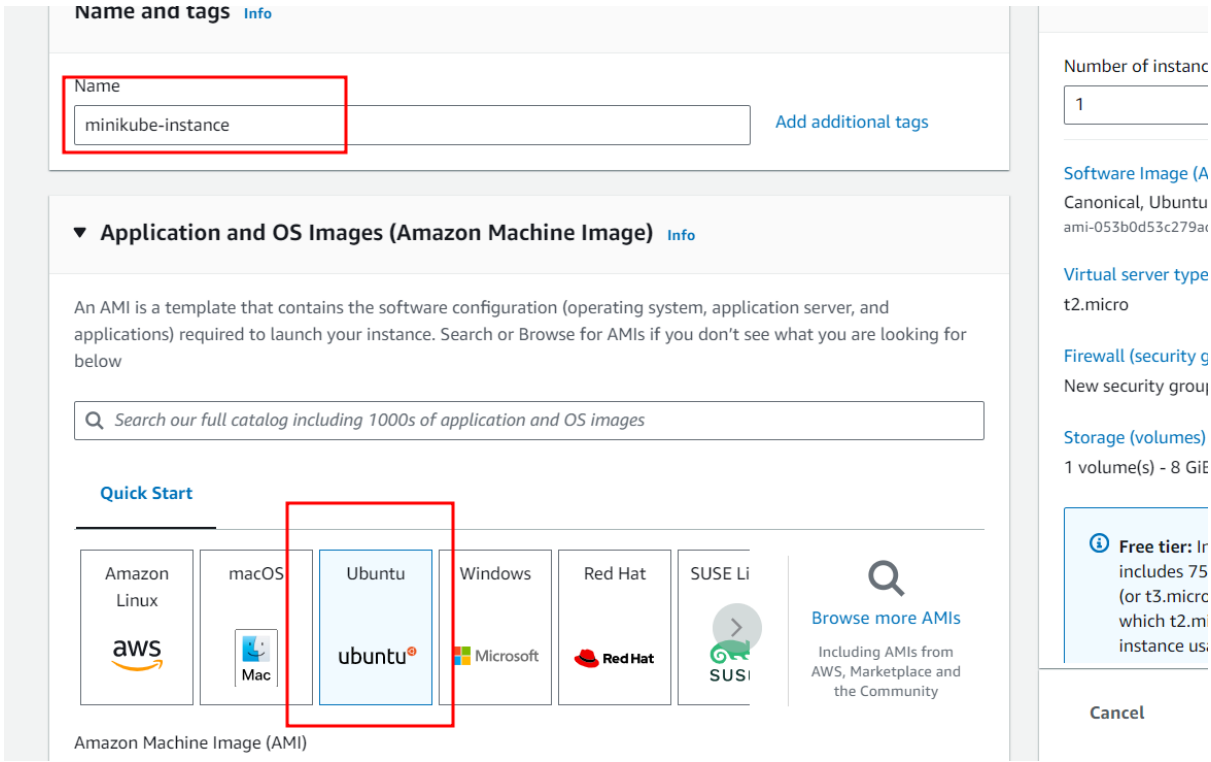
Then we installed all required software which help deploy our application.

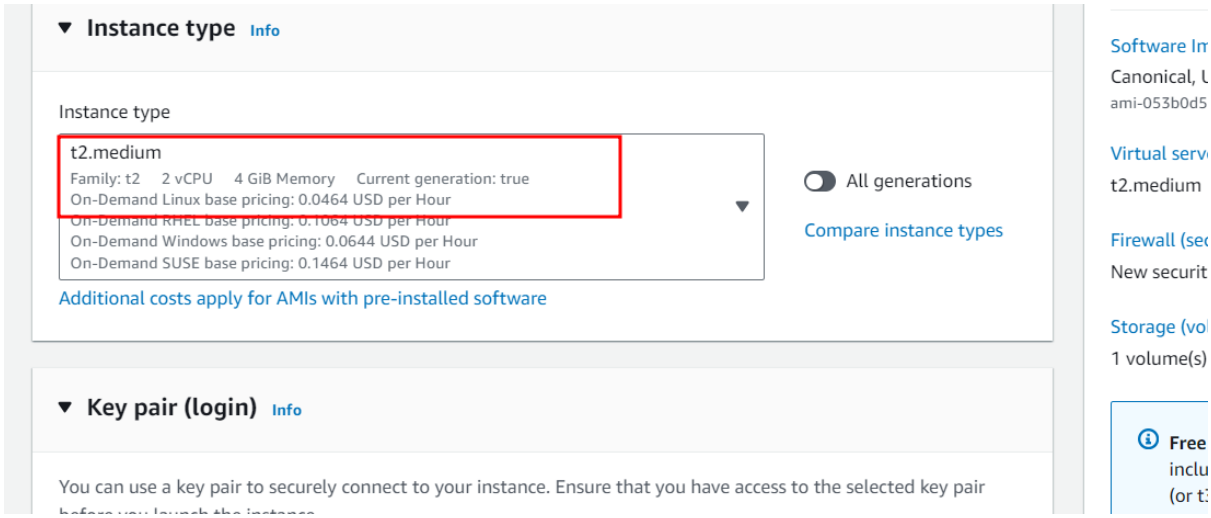
That VM provide Unique IP Address.

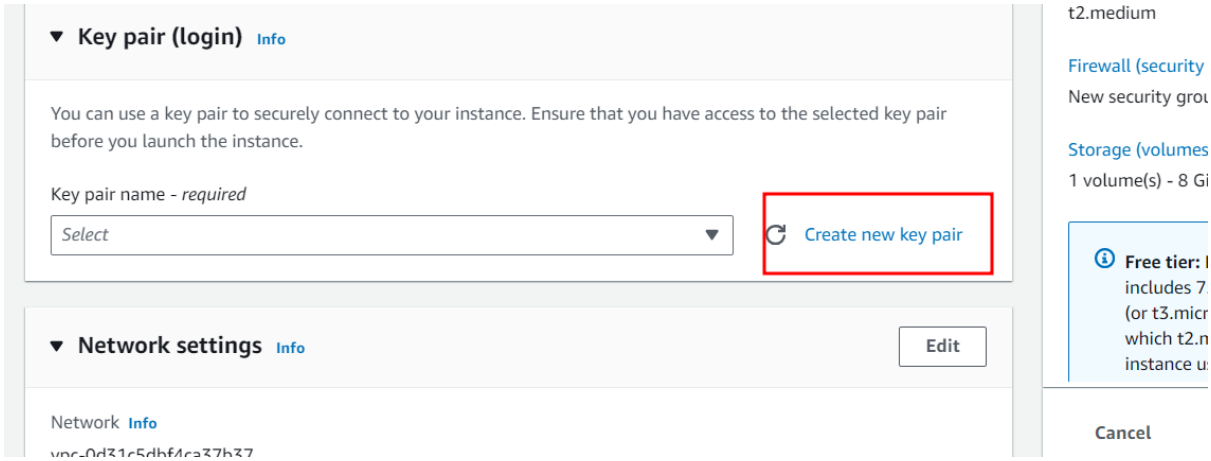
Public and private IP Address.

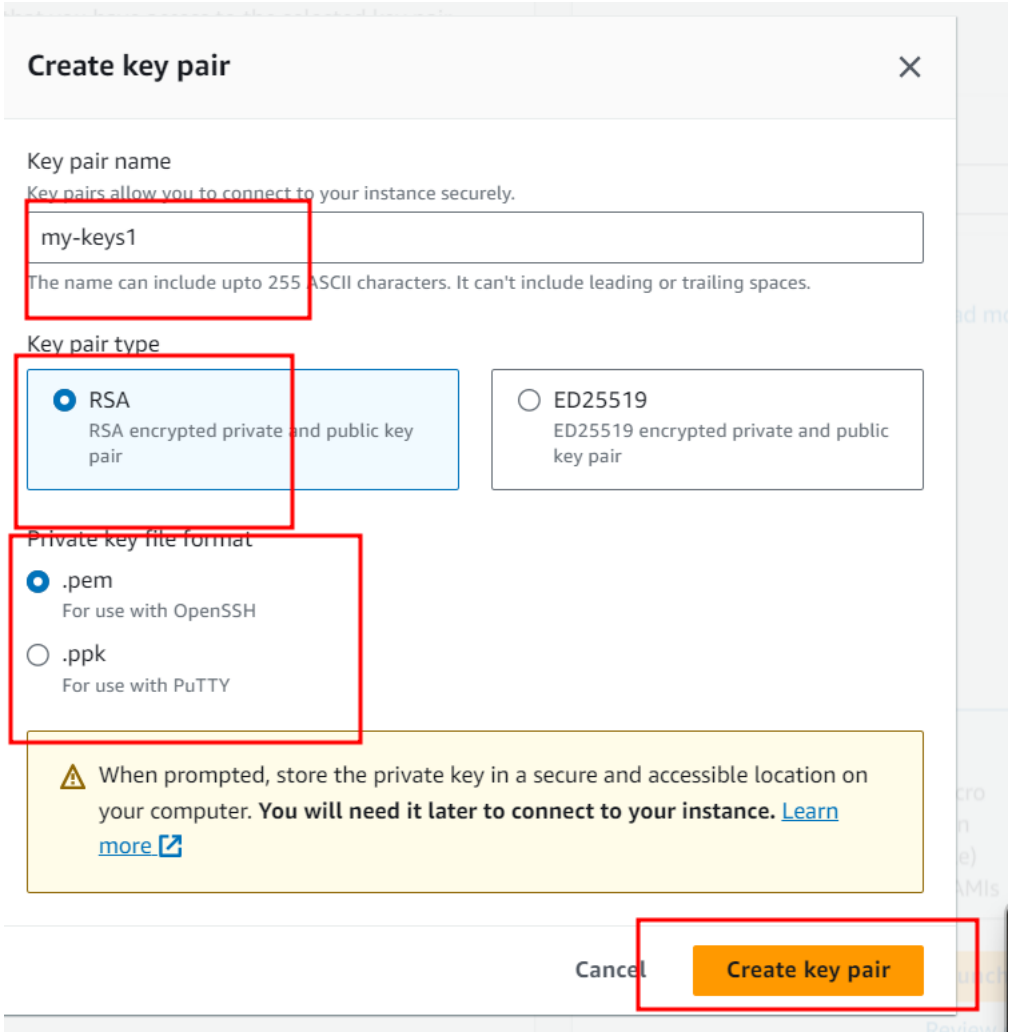


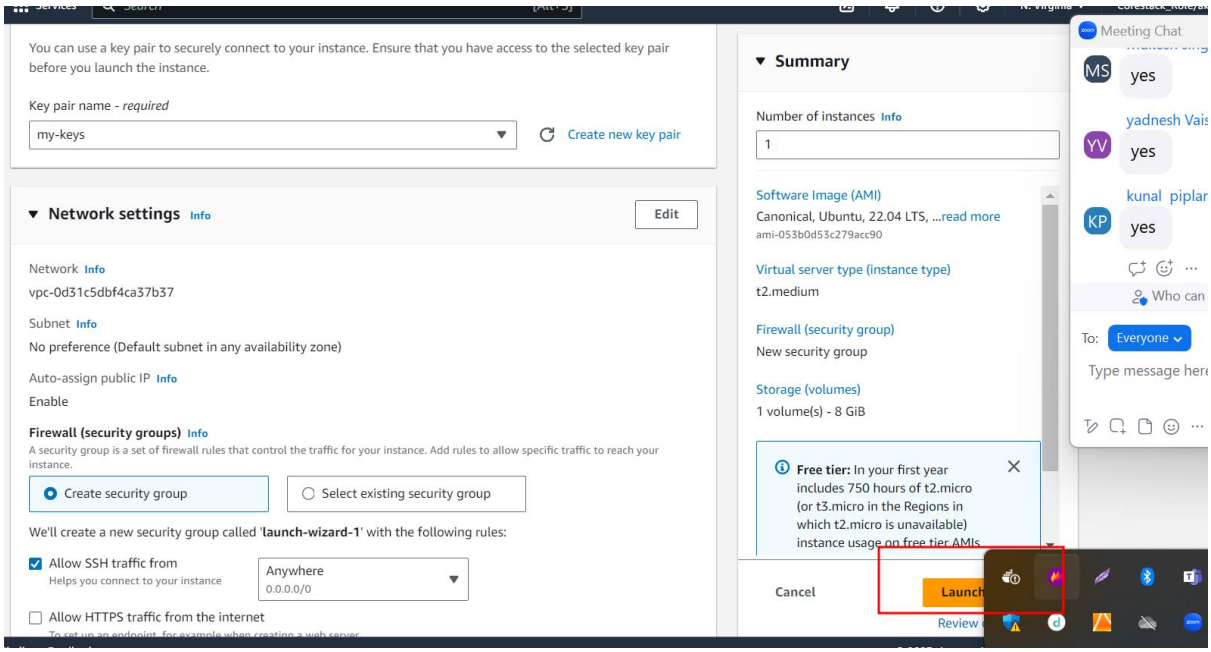

Resources
EC2 Global view
You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:
Instances (running) 0
Auto Scaling Groups 0
Dedicated Hosts 0
Elastic IPs 0
Instances 0
Key pairs 0
Load balancers 0
Placement groups 0
Security groups 1
Snapshots 0
Volumes 0
Launch instance
To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.
Launch instance
Migrate a server
Note: Your instances will launch in the US East (N. Virginia) Region
Service health
AWS Health Dashboard
Region
US East (N. Virginia)
Zones
Zone name
Zone ID
us-east-1a
use1-az4
us-east-1b
use1-az6
Scheduled events

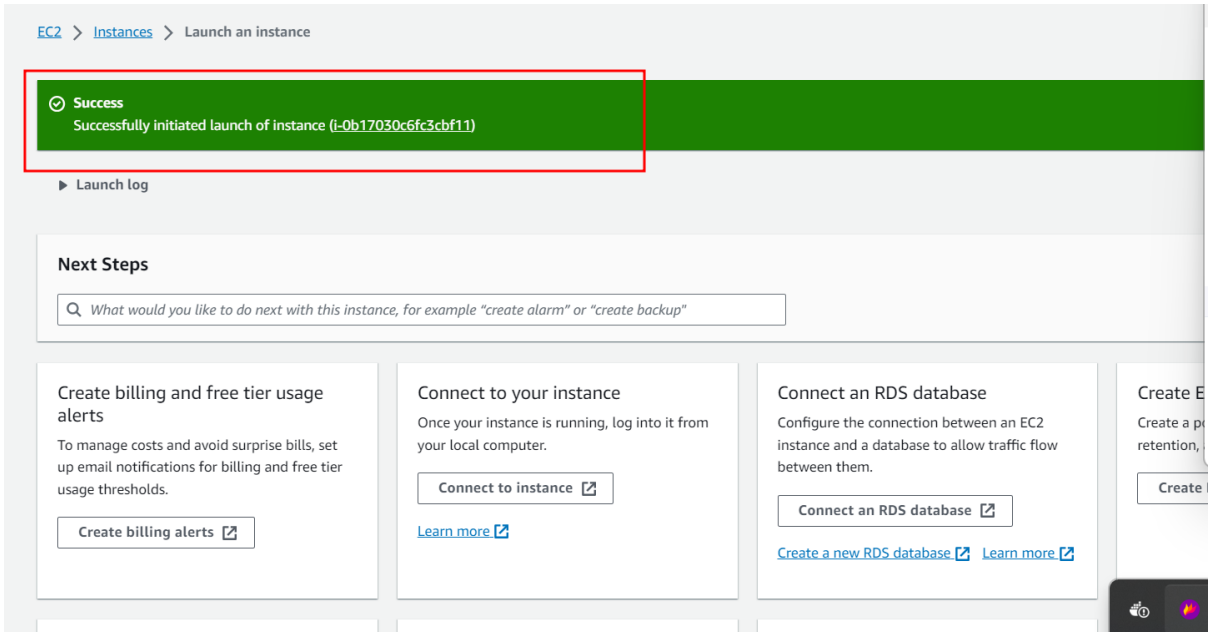

Name and tags Info
Name
minikube-instance
Add additional tags
Application and OS Images (Amazon Machine Image) Info
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below
Search our full catalog including 1000s of application and OS images
Quick Start
Amazon Linux
macOS
Ubuntu
Windows
Red Hat
SUSE Li
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community
Amazon Machine Image (AMI)
Number of instanc
1
Software Image (A
Canonical, Ubuntu
Virtual server type
t2.micro
Firewall (security g
New security grou
Storage (volumes)
1 volume(s) - 8 GiE
Free tier: In
includes 75
(or t3.micro
which t2.mi
instance us
Cancel

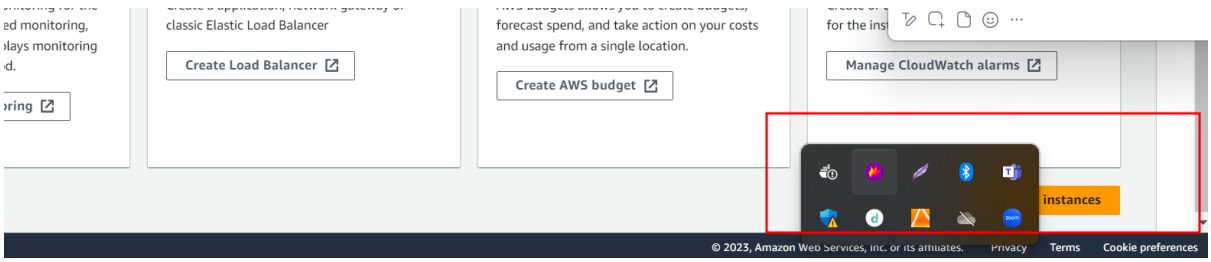

Instance type Info
Instance type
t2.medium
Family: t2 2 vCPU 4 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0464 USD per Hour
On-Demand RHEL base pricing: 0.1064 USD per Hour
On-Demand Windows base pricing: 0.0644 USD per Hour
On-Demand SUSE base pricing: 0.1464 USD per Hour
All generations
Compare instance types
Additional costs apply for AMIs with pre-installed software
Key pair (login) Info
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair
Software Im
Canonical, U
ami-053b0d5
Virtual serv
t2.medium
Firewall (sec
New securit
Storage (vo
1 volume(s)
Free
inclu
(or t

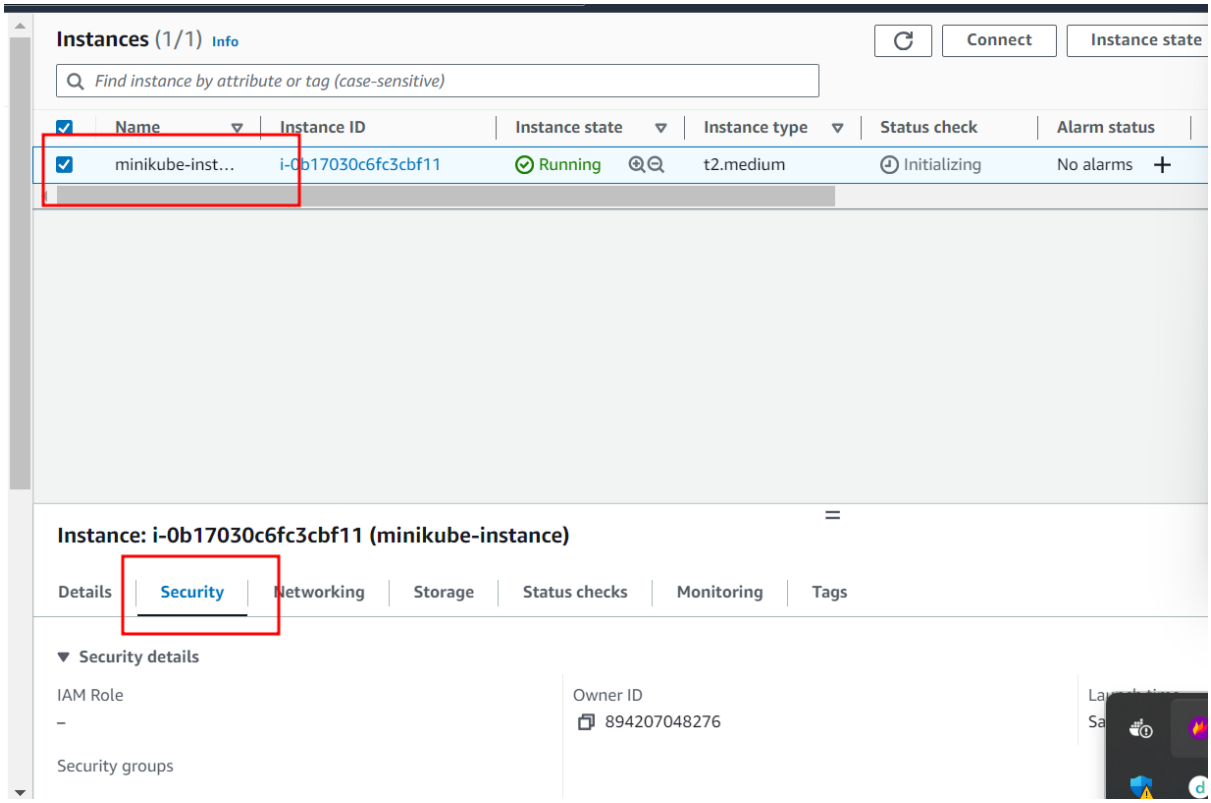

Key pair (login) Info
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.
Key pair name - required
Select
Create new key pair
Network settings Info
Edit
Network Info
t2.medium
Firewall (security
New security grou
Storage (volumes
1 volume(s) - 8 Gi
Free tier:
includes 7
(or t3.micr
which t2.n
instance u
Cancel


Create key pair
Key pair name
Key pairs allow you to connect to your instance securely.
my-keys1
The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.
Key pair type
RSA
RSA encrypted private and public key pair
ED25519
ED25519 encrypted private and public key pair
Private key file format
.pem
For use with OpenSSH
.ppk
For use with PuTTY
When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. Learn more
Cancel
Create key pair

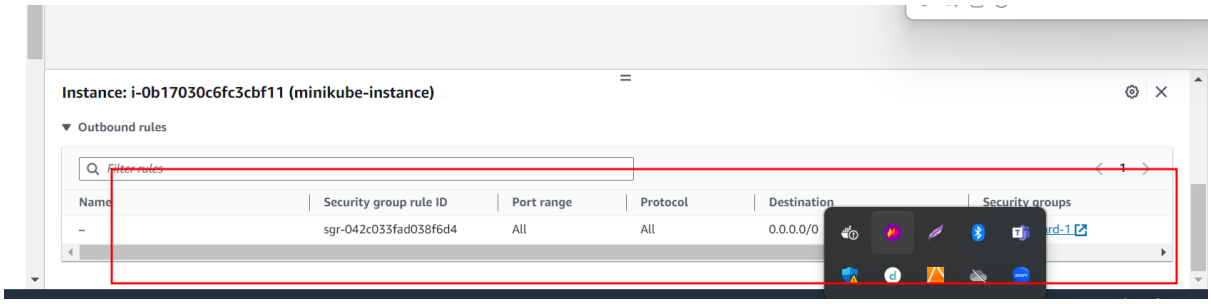

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.
Key pair name - required
my-keys
Create new key pair
Network settings Info
Edit
Network Info
vpc-0d31c5dbf4ca37b37
Subnet Info
No preference (Default subnet in any availability zone)
Auto-assign public IP Info
Enable
Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.
Create security group
Select existing security group
We'll create a new security group called 'launch-wizard-1' with the following rules:
Allow SSH traffic from
Helps you connect to your instance
Anywhere
0.0.0.0/0
Allow HTTPS traffic from the internet
Summary
Number of instances Info
1
Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ...read more
ami-053b0d53c279acc90
Virtual server type (instance type)
t2.medium
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs
Cancel
Launch
Review
Meeting Chat
MS
yes
yadnesh Vais
YV
yes
kunal piplar
KP
yes
Who can
To:
Everyone
Type message here


EC2 > Instances > Launch an instance
Success
Successfully initiated launch of instance (i-0b17030c6fc3cbf11)
Launch log
Next Steps
What would you like to do next with this instance, for example "create alarm" or "create backup"
Create billing and free tier usage alerts
To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds.
Create billing alerts
Connect to your instance
Once your instance is running, log into it from your local computer.
Connect to instance
Learn more
Connect an RDS database
Configure the connection between an EC2 instance and a database to allow traffic flow between them.
Connect an RDS database
Create a new RDS database
Learn more

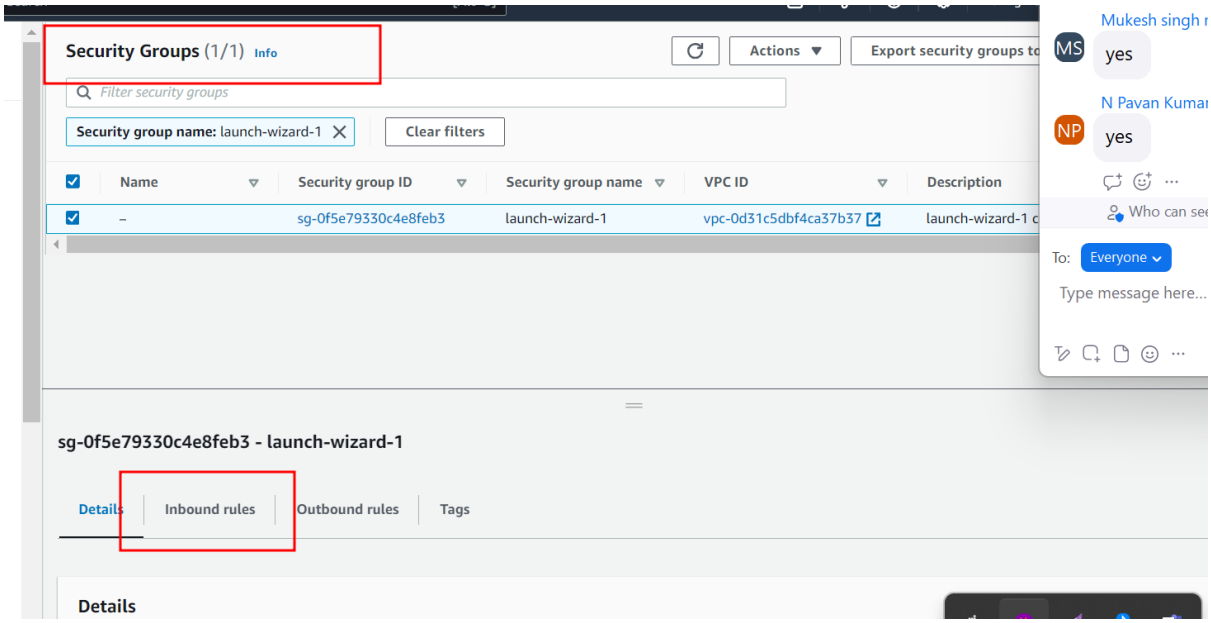

classic Elastic Load Balancer
Create Load Balancer
forecast spend, and take action on your costs and usage from a single location.
Create AWS budget
Manage CloudWatch alarms
instances
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

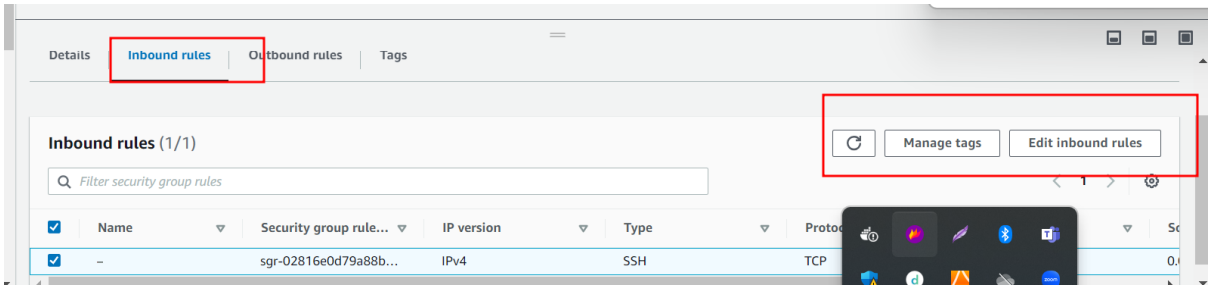

Instances (1/1) Info
Connect
Instance state
Find instance by attribute or tag (case-sensitive)
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
minikube-inst...
i-0b17030c6fc3cbf11
Running
t2.medium
Initializing
No alarms
Instance: i-0b17030c6fc3cbf11 (minikube-instance)
Details
Security
Networking
Storage
Status checks
Monitoring
Tags
Security details
IAM Role
–
Owner ID
894207048276
Security groups

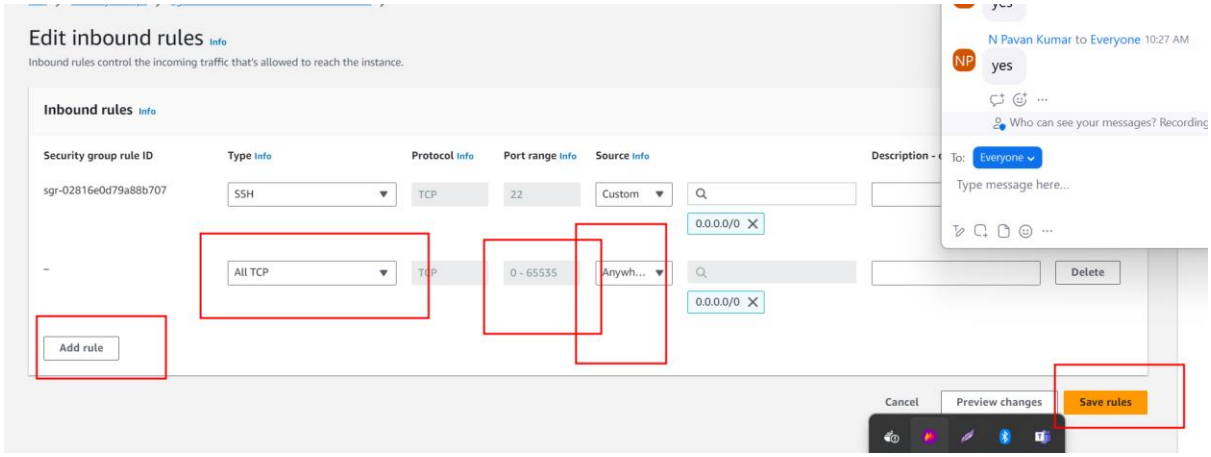
Scroll down

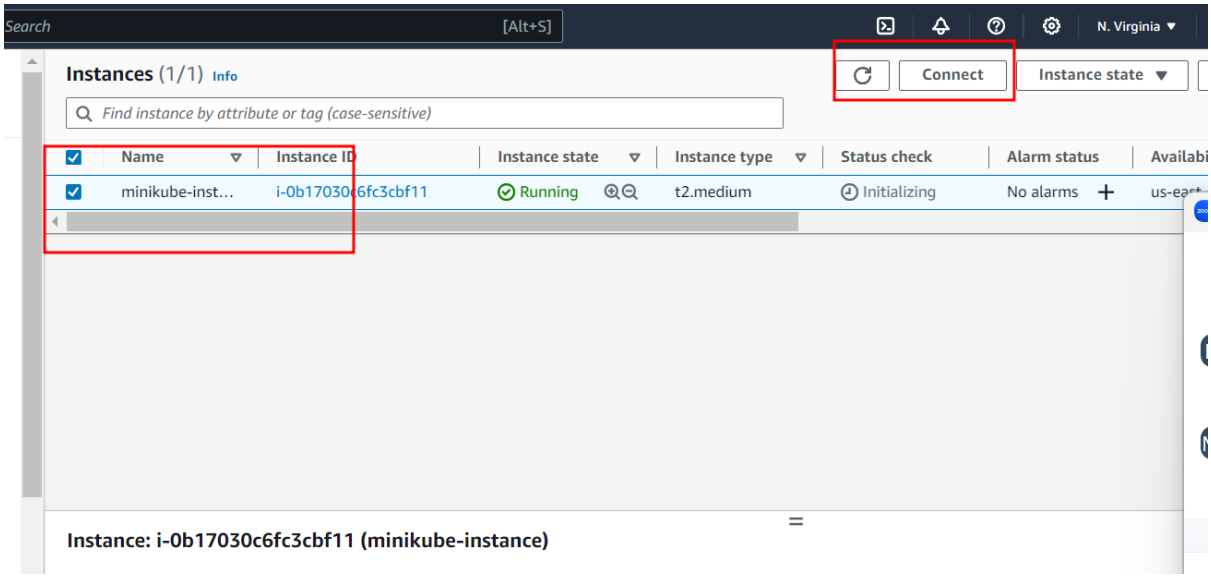


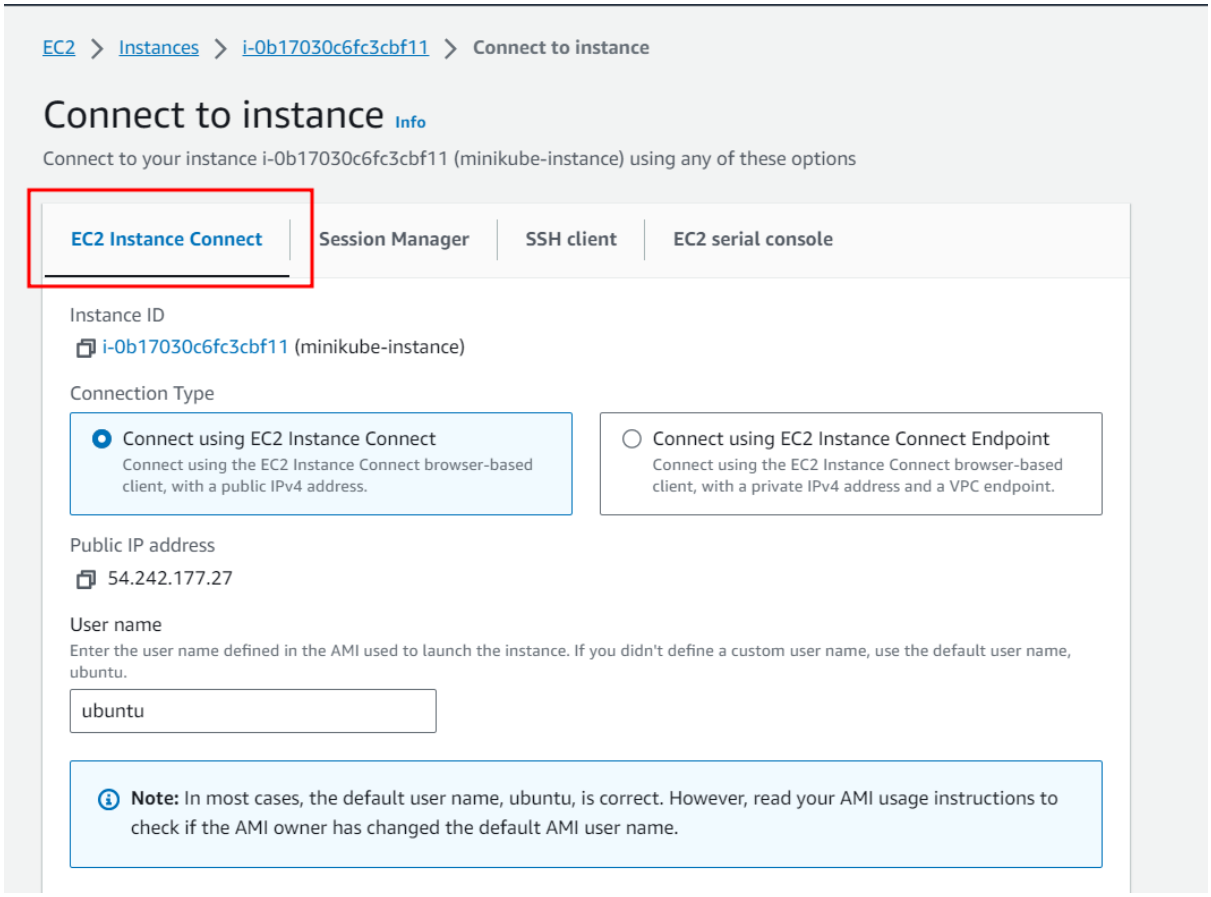
Launch security group

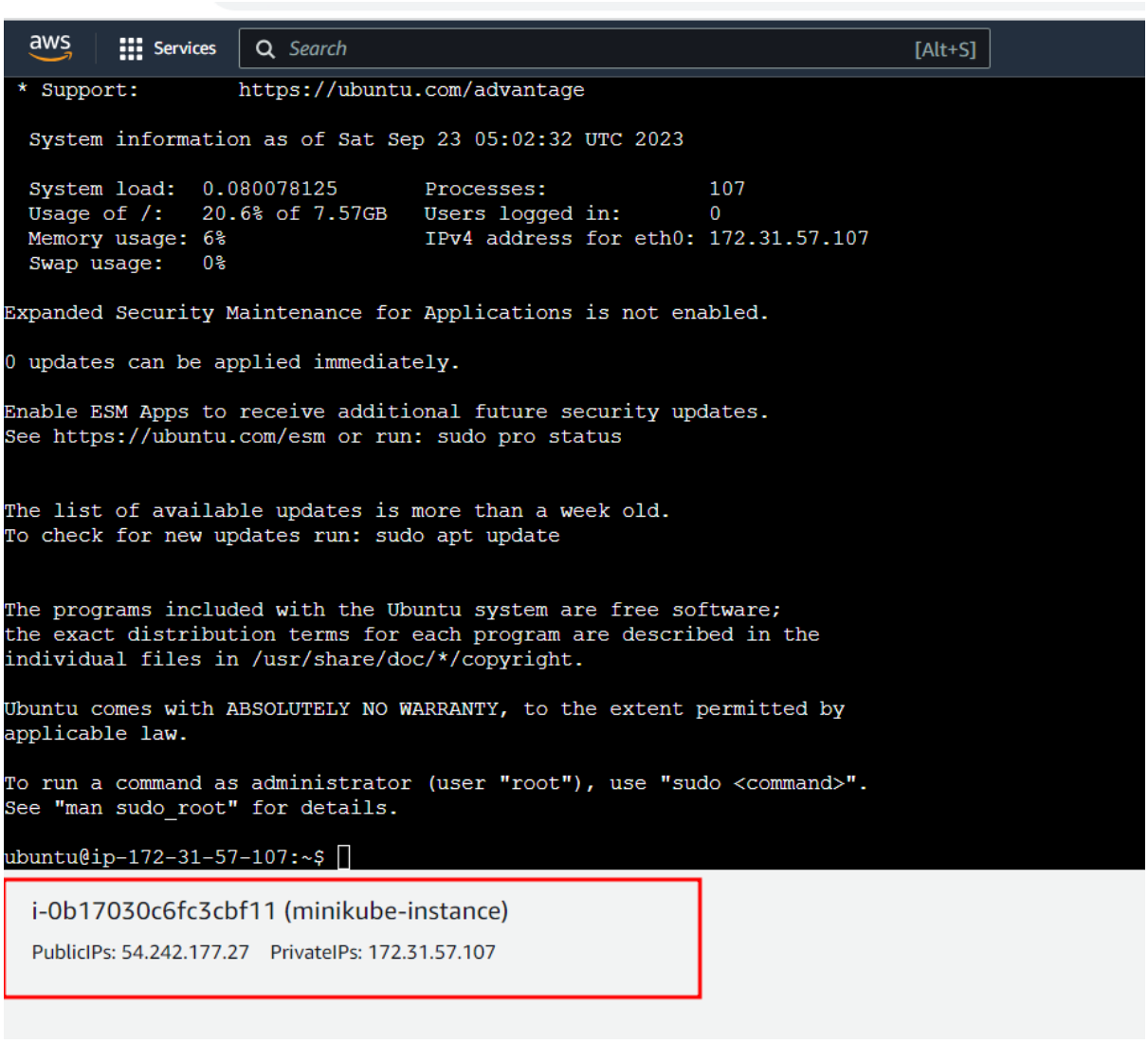





Edit inbound rules Info
Inbound rules control the incoming traffic that's allowed to reach the instance.
Inbound rules Info
Security group rule ID
Type Info
Protocol Info
Port range Info
Source Info
sgr-02816e0d79a88b707
SSH
TCP
22
Custom
0.0.0.0/0
All TCP
TCP
0 - 65535
Anywh...
0.0.0.0/0
Delete
Add rule
Cancel
Preview changes
Save rules
N Pavan Kumar to Everyone 10:27 AM
yes
Who can see your messages? Recording
To: Everyone
Type message here...


Search
[Alt+S]
N. Virginia
Instances (1/1) Info
Connect
Instance state
Find instance by attribute or tag (case-sensitive)
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
minikube-inst...
i-0b17030c6fc3cbf11
Running
t2.medium
Initializing
No alarms
Instance: i-0b17030c6fc3cbf11 (minikube-instance)


EC2 > Instances > i-0b17030c6fc3cbf11 > Connect to instance
Connect to instance Info
Connect to your instance i-0b17030c6fc3cbf11 (minikube-instance) using any of these options
EC2 Instance Connect
Session Manager
SSH client
EC2 serial console
Instance ID
i-0b17030c6fc3cbf11 (minikube-instance)
Connection Type
Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.
Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.
Public IP address
54.242.177.27
User name
Enter the user name defined in the AMI used to launch the instance. If you didn't define a custom user name, use the default user name, ubuntu.
ubuntu
Note: In most cases, the default user name, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.



Installed required software in EC2 instance.

First create the EC2 instance

move to root user

sudo su

root -> curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add -
download

root -> sudo echo deb http://apt.kubernetes.io/ kubernetes-xenial main > /etc/apt/sources.list.d/kubernetes.list

root -> sudo apt-get update

root -> sudo apt install docker.io kubectl install docker and kubectl

Now check the status of docker

root -> sudo systemctl status docker

Cntr + C : exit terminal

root -> sudo systemctl restart docker

root - > sudo systemctl stop docker

root -> sudo systemctl start docker

root -> sudo systemctl daemon-reload

Test the program

root -> docker run hello-world

To check the kubectl version

root -> kubectl version

Now we will install docker-compose

root --> sudo curl -L "https://github.com/docker/compose/releases/download/v2.20.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

root -> sudo chmod +x /usr/local/bin/docker-compose

install the minikube

root -> sudo apt install -y curl wget apt-transport-https

root -> curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64

root --> sudo install minikube-linux-amd64 /usr/local/bin/minikube

root --> minikube version

root --> sudo apt-get install -y conntrack

root --> sudo apt update && sudo apt upgrade

now exits from root user using command as

root -> exit

Then add the docker in user group using the command as

ubuntu -> sudo usermod -aG docker $USER && newgrp docker

Now check the

ubuntu --> docker images

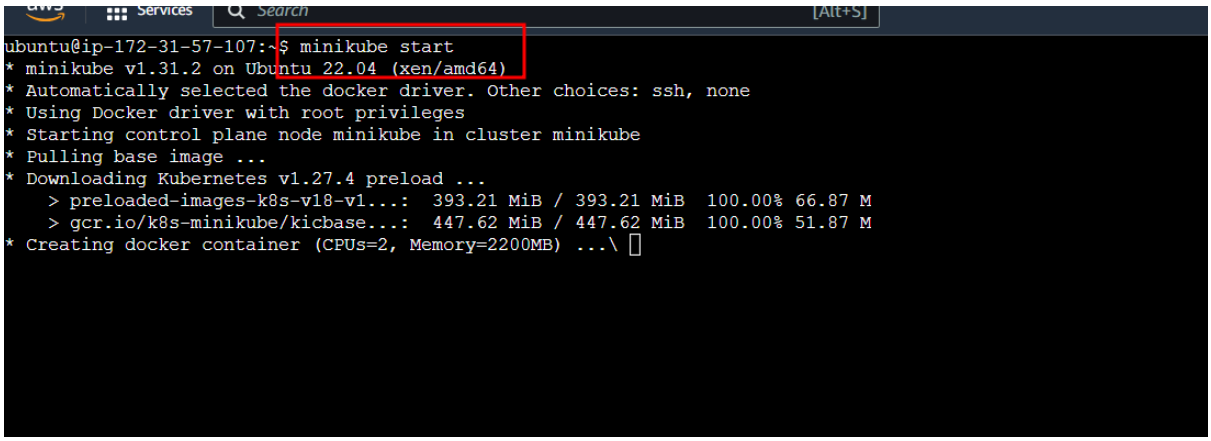
ubuntu --> docker ps

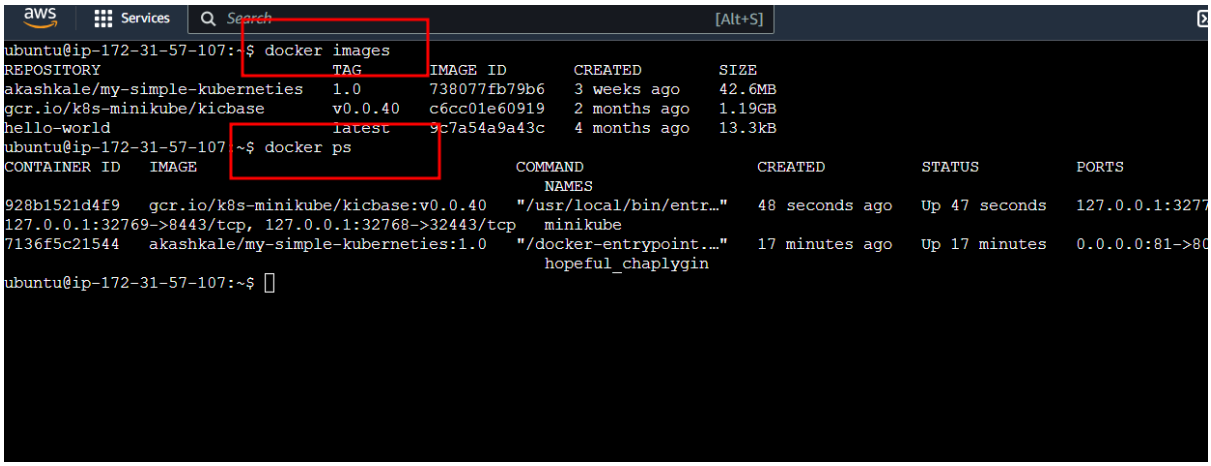
ubuntu --> minikube start

After installed all required software or tool

Then run the command as

minikube start





kubectl cluster-info it provide cluster information ie minikube

we need to create 3 instance

1 master node

kubeadm init

it will provide us token which help to join that cluster

public ip and private ip

2 workder1node and worker2node

Worker node 1

Kubectl join token with IPAddress

Worker node 2

Kubectl join token with IPAddress

kubectl get pods

kubectl get service

kubectl get deployment

kubectl get namespace

```
akashkale/my-simple-kuberneties:1.0
```

```
nginx:latest
```

```
kubectl create deployment my-deploy –-image=akashkale/my-simple-kuberneties:1.0
```

```
deployment provide meta data for pods
```

```
pods are use to run more than one container in Kubernetes cluster
```

kubectl delete deployment deploymentname

kubectl delete pod podname

kubectl create deployment my-deploy2 --image=akashkale/my-simple-kuberneties:1.0 --replicas=3

we created totally 3 pods using replicas option

this command is use to find details about specific pods.

kubectl describe pod my-deploy2-58f6c6545b-lzlr4

if we want to expose our pods which is part of cluster we need to use service with type of service.

kubectl expose deployment my-deploy1 --name=my-ser1 --type=NodePort --port=80

my-demploy1 deployment name

my-ser1 service name

type of serie NodePort

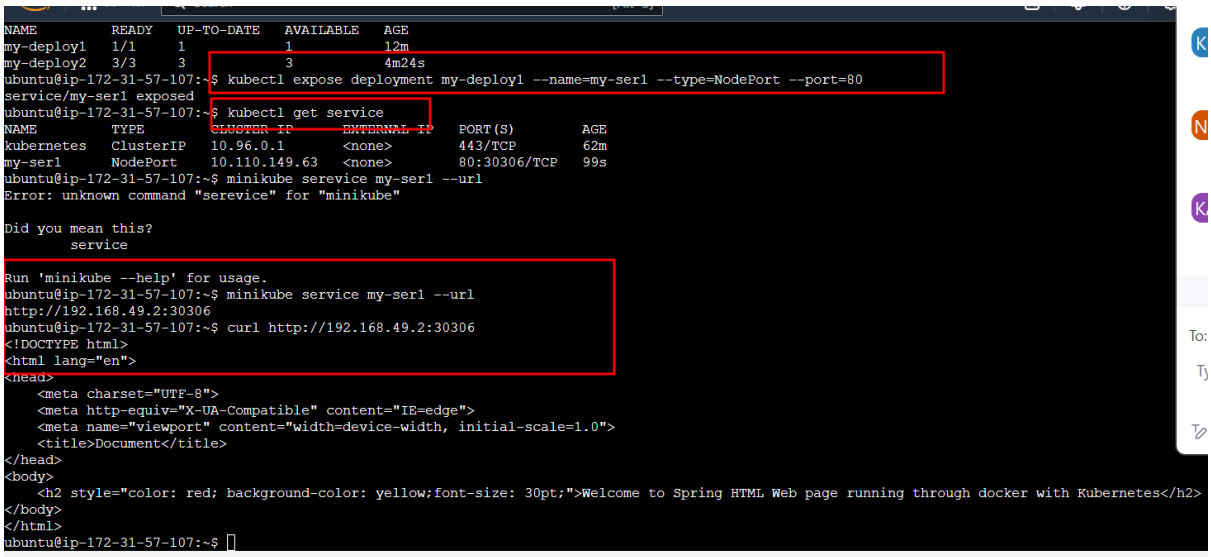
port number of my application 80

kubectl get service

if type is NodePort we can access that service within a cluster environment.

To check that service IP Address we need to run the command as

minikube service servicename —-ur



NodePort we can access service within cluster node ip address ie minikube

ClusterIp we can access that service using cluster Ip Address.

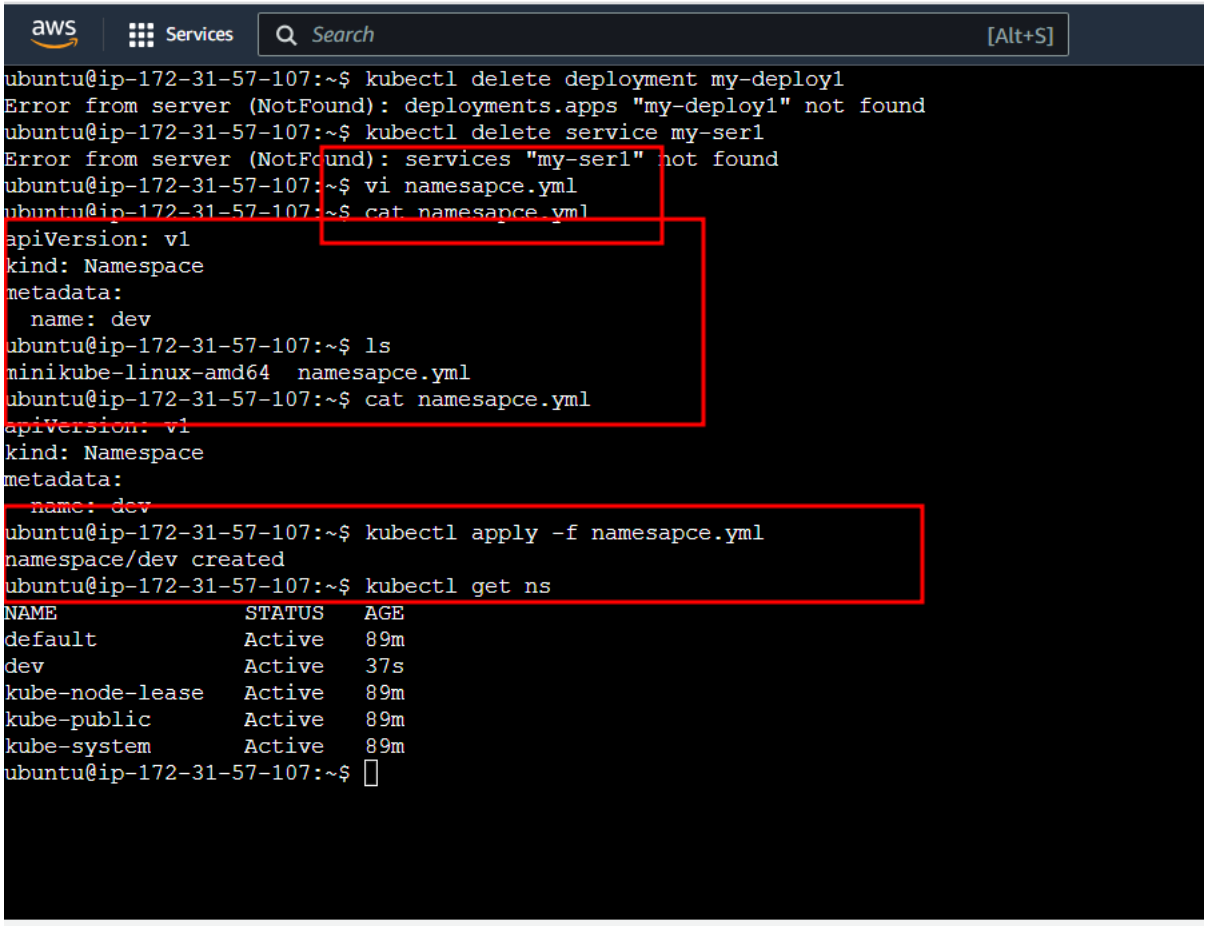
LoadBalancer then we can access that application using external Ip address outside cluster.

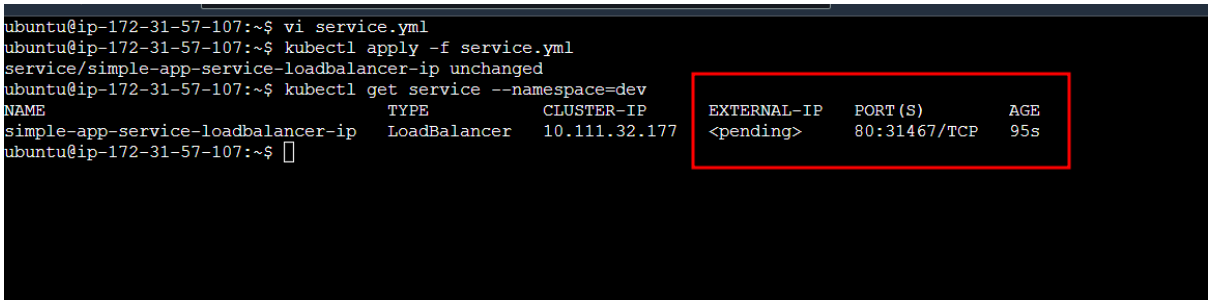
Please do clean up activity

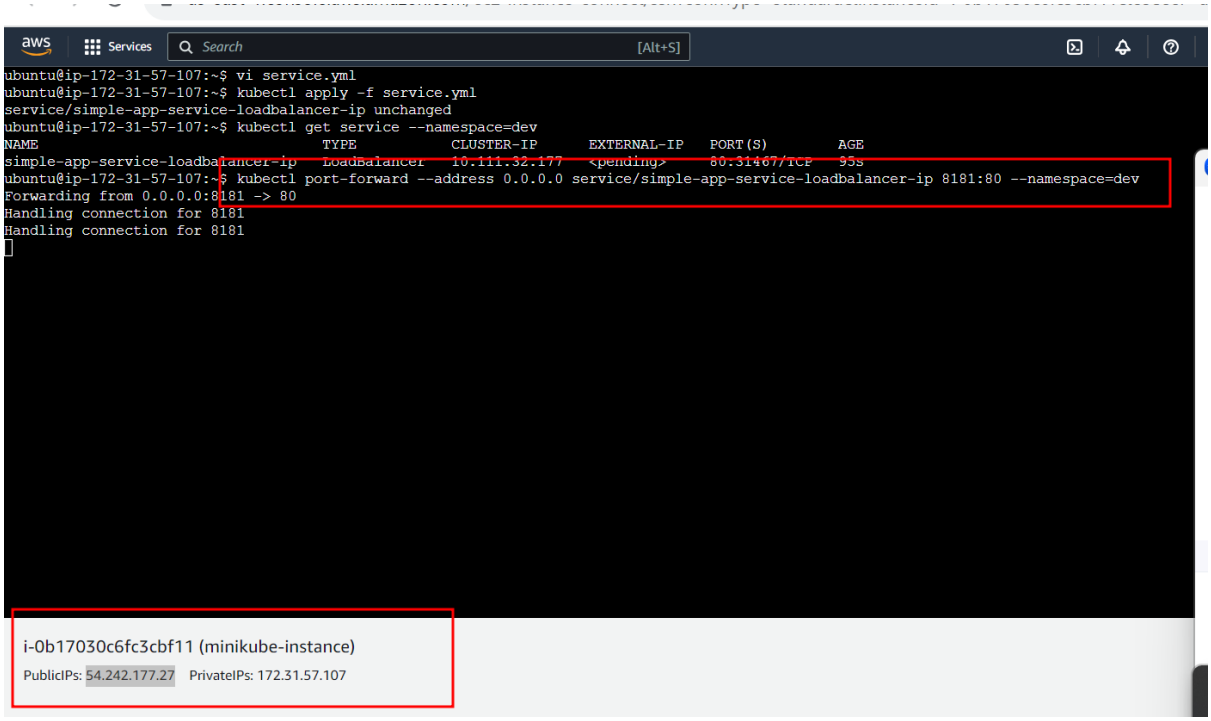
Delete service, deployment

We need to use declarative mode

Write all deployment, service, namespace, pods in yml file.


aws
Services
Search
[Alt+S]
ubuntu@ip-172-31-57-107:~$ kubectl delete deployment my-deploy1
Error from server (NotFound): deployments.apps "my-deploy1" not found
ubuntu@ip-172-31-57-107:~$ kubectl delete service my-ser1
Error from server (NotFound): services "my-ser1" not found
ubuntu@ip-172-31-57-107:~$ vi namesapce.yml
ubuntu@ip-172-31-57-107:~$ cat namesapce.yml
apiVersion: v1
kind: Namespace
metadata:
name: dev
ubuntu@ip-172-31-57-107:~$ ls
minikube-linux-amd64 namesapce.yml
ubuntu@ip-172-31-57-107:~$ cat namesapce.yml
apiVersion: v1
kind: Namespace
metadata:
name: dev
ubuntu@ip-172-31-57-107:~$ kubectl apply -f namesapce.yml
namespace/dev created
ubuntu@ip-172-31-57-107:~$ kubectl get ns
NAME STATUS AGE
default Active 89m
dev Active 37s
kube-node-lease Active 89m
kube-public Active 89m
kube-system Active 89m
ubuntu@ip-172-31-57-107:~$


ubuntu@ip-172-31-57-107:~$ vi service.yml
ubuntu@ip-172-31-57-107:~$ kubectl apply -f service.yml
service/simple-app-service-loadbalancer-ip unchanged
ubuntu@ip-172-31-57-107:~$ kubectl get service --namespace=dev
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
simple-app-service-loadbalancer-ip LoadBalancer 10.111.32.177 <pending> 80:31467/TCP 95s
ubuntu@ip-172-31-57-107:~$



Now open your application with ec2 instance Ip Address with expose port number

https://github.com/Kaleakash/kuberentes_yml_files

Day 10

Ansible tool

Ansible is also part of devops tools.

Ansible tool is an open source IT engine tool that take care for application deployment,

Orchestration tool, configuration management and many others IT Process.

Server : web Server, Application server, db serve, firewall server, security setting.

Ansible Server → Machine from ansible server machine we can control

All server node machine to do any configuration base upon our requirements.

SSH Client we connect all machine together

Ansible provide playbook using playbook we can write some task

To install java or python.

Ansible provided more than 500 modules those module help to do task.

Fetch

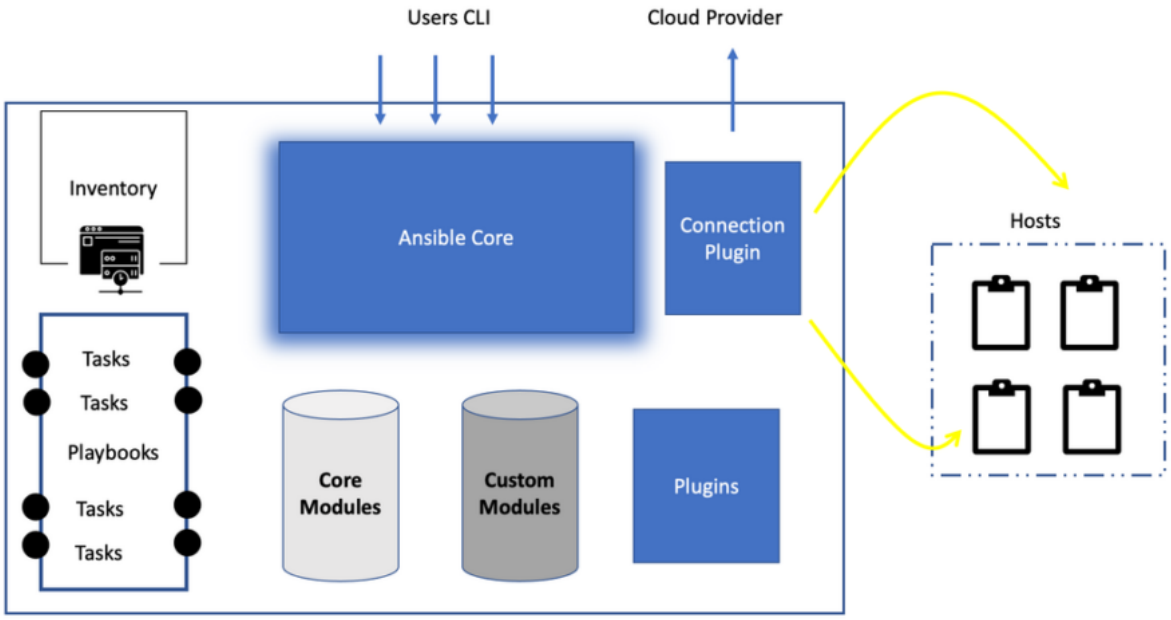
Node1

Node2

Node3

Node4

Node5



Controller machine or master node or Ansible server : This is where ansible get installed. This machine is responsible to control more than one server machine or host.

All host machine connect using SSH client configuration.

Inventory This is basically an initializing file that contains information about the server that we are managing. This file generally is yml file.

Playbook : it is an organized unit of script defining an automated work for the configuration management of server.

Module: ansible provide more than 500 modules. Each modules are contains unit of code that control system resource or execute system command.

Task : A task block defines a single procedure to be execute on the node or server or worker node like installing, start, stop, create file, delete folder,

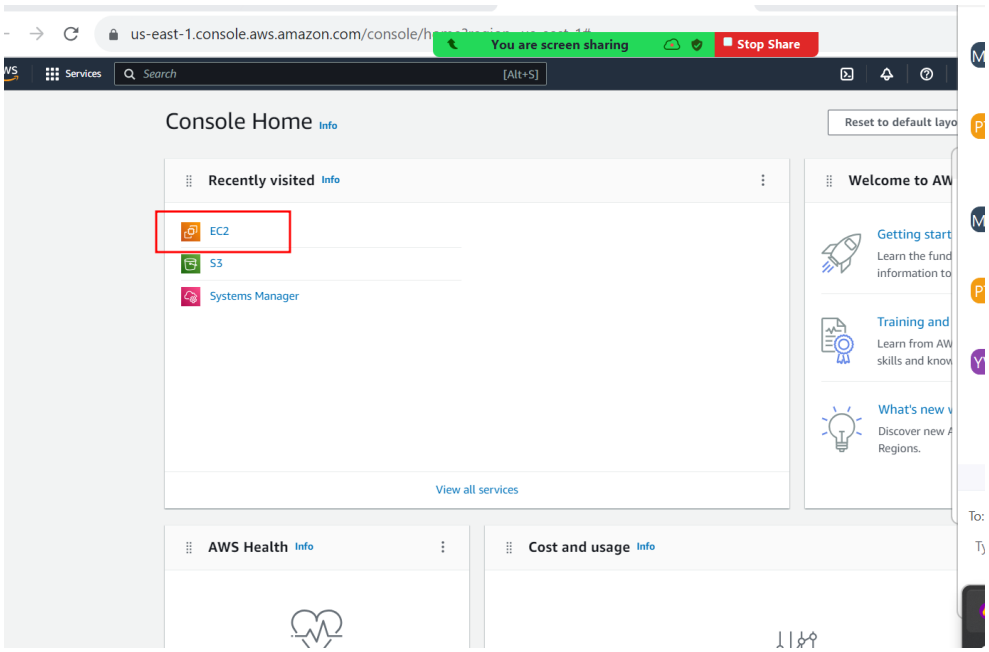
We will create three nodes

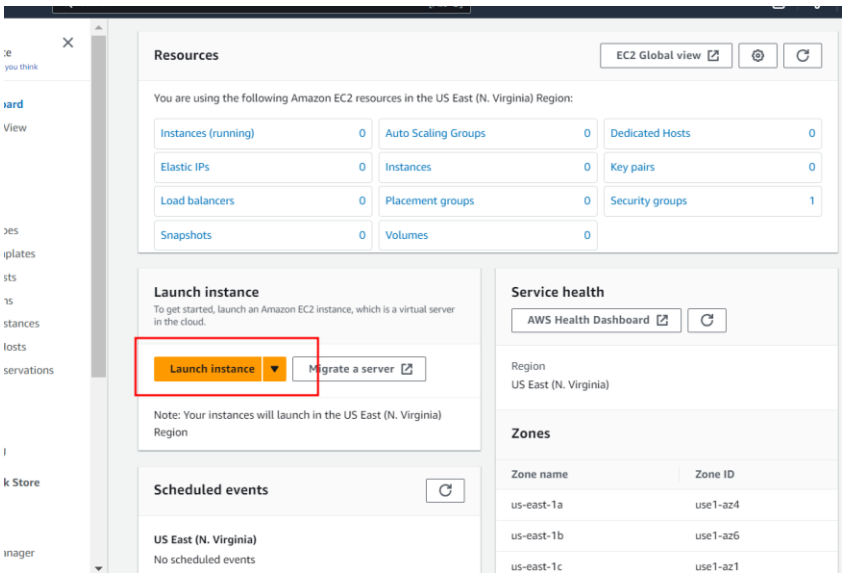
Ansible Server – Node : we install ansible server

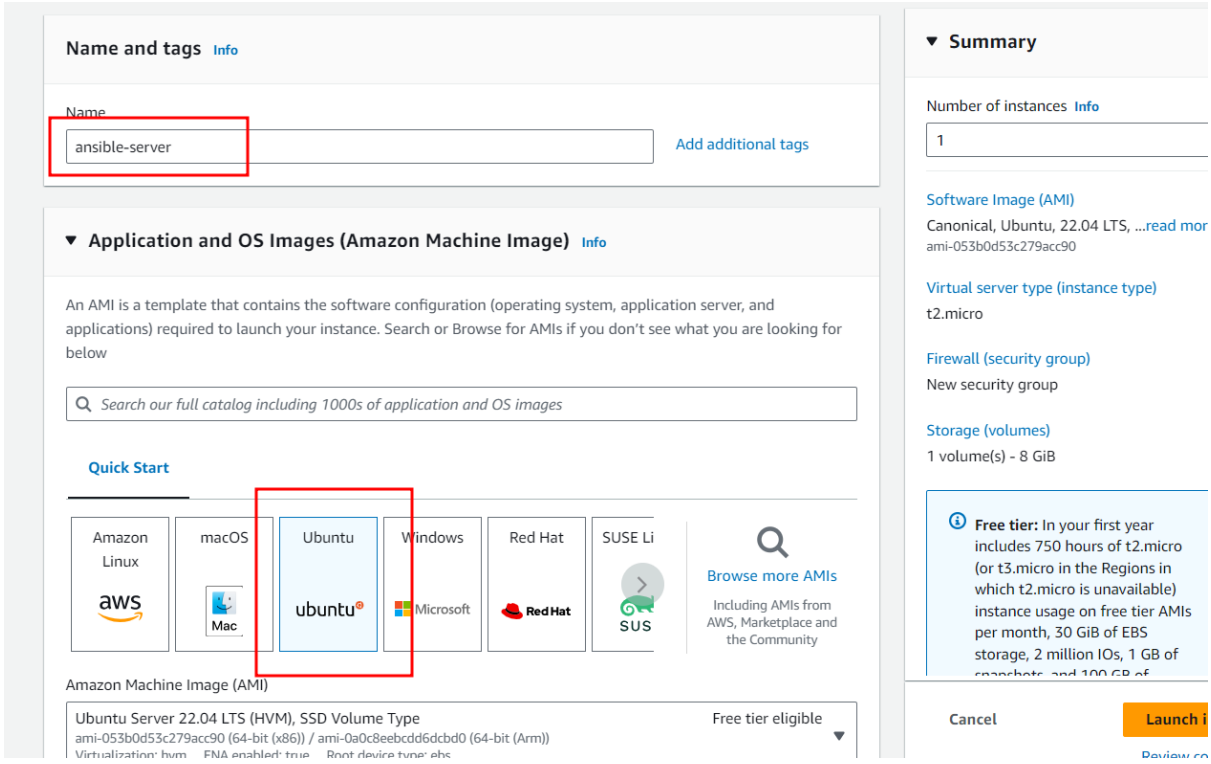
Node1 normal server

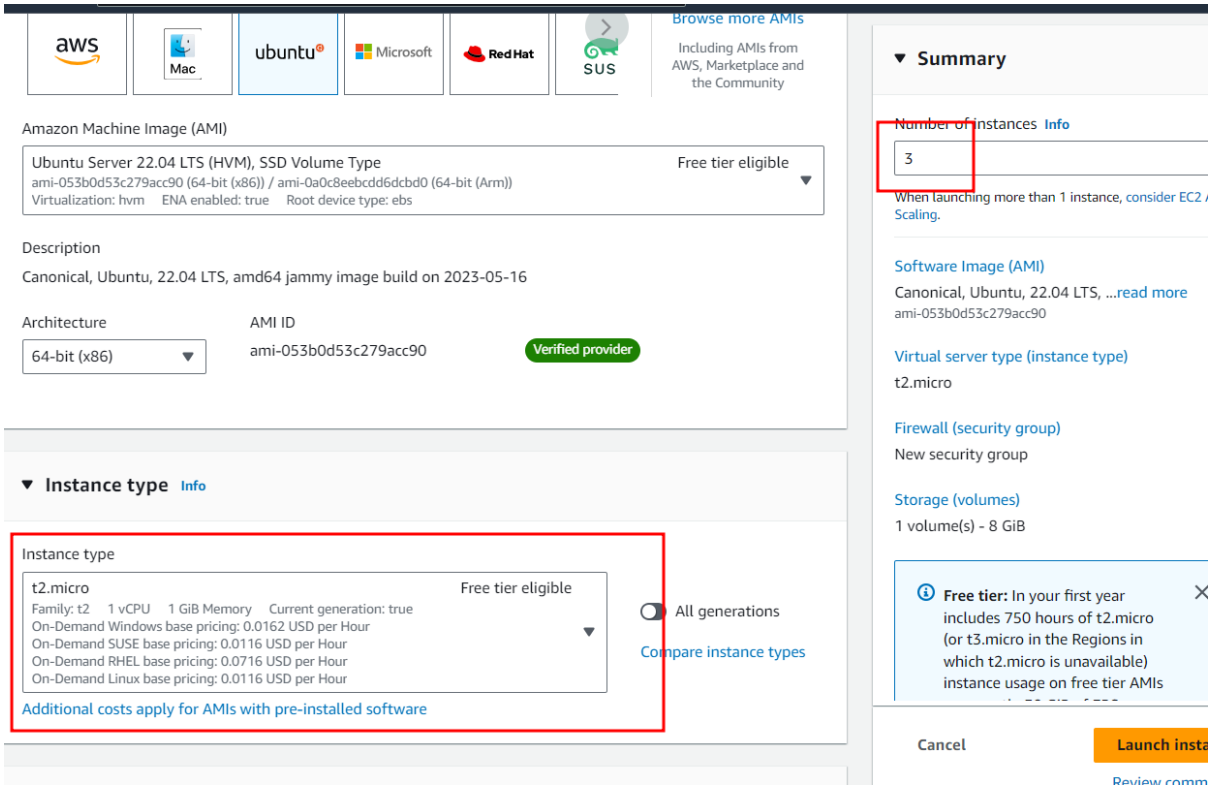
Node2 normal server

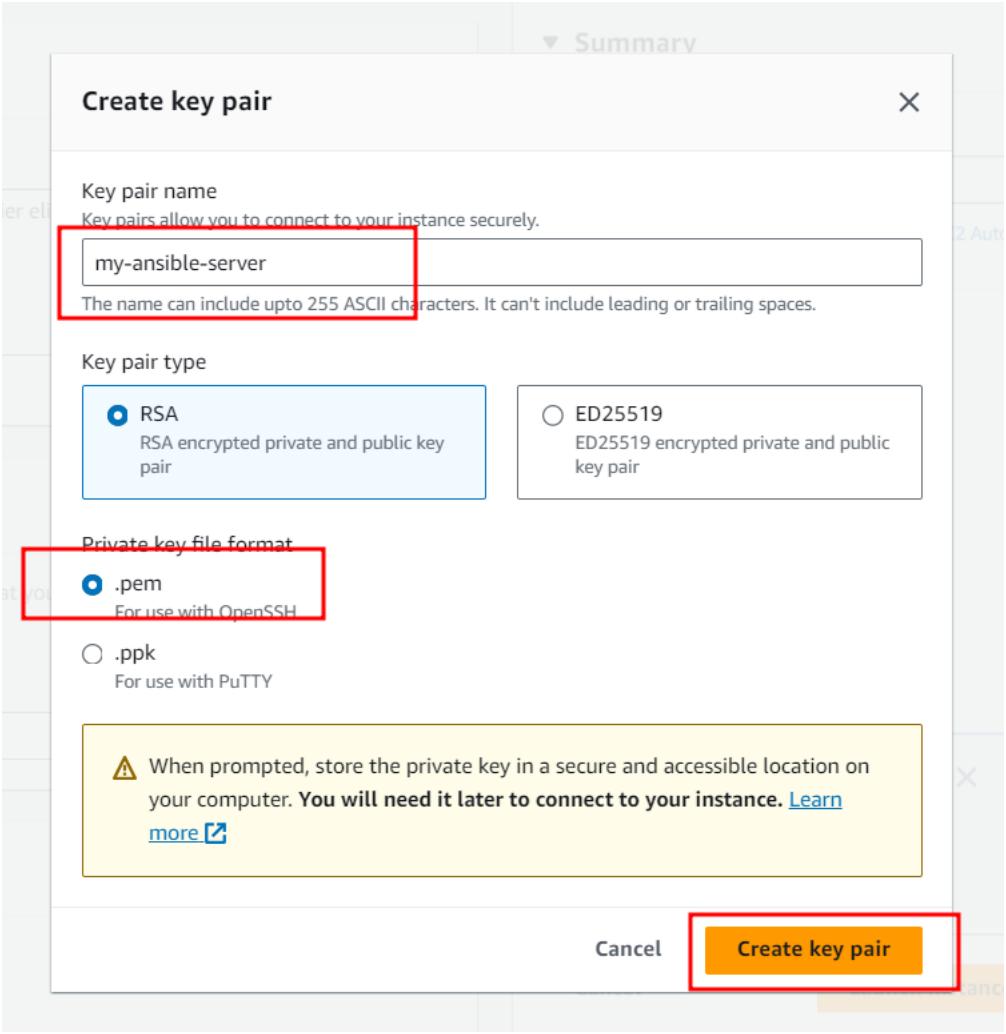
Login to AWS account

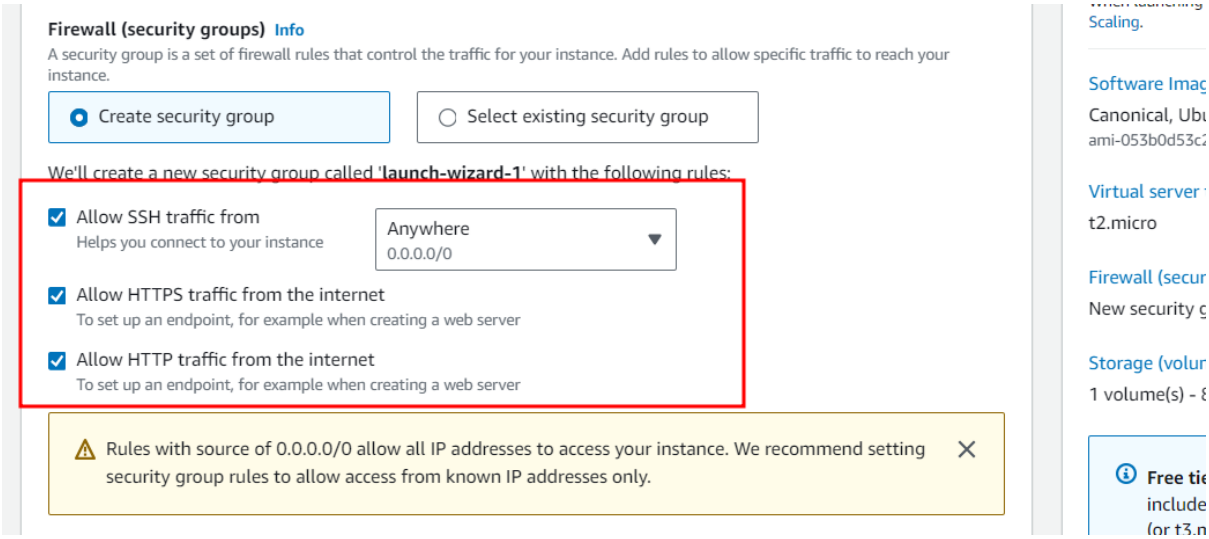


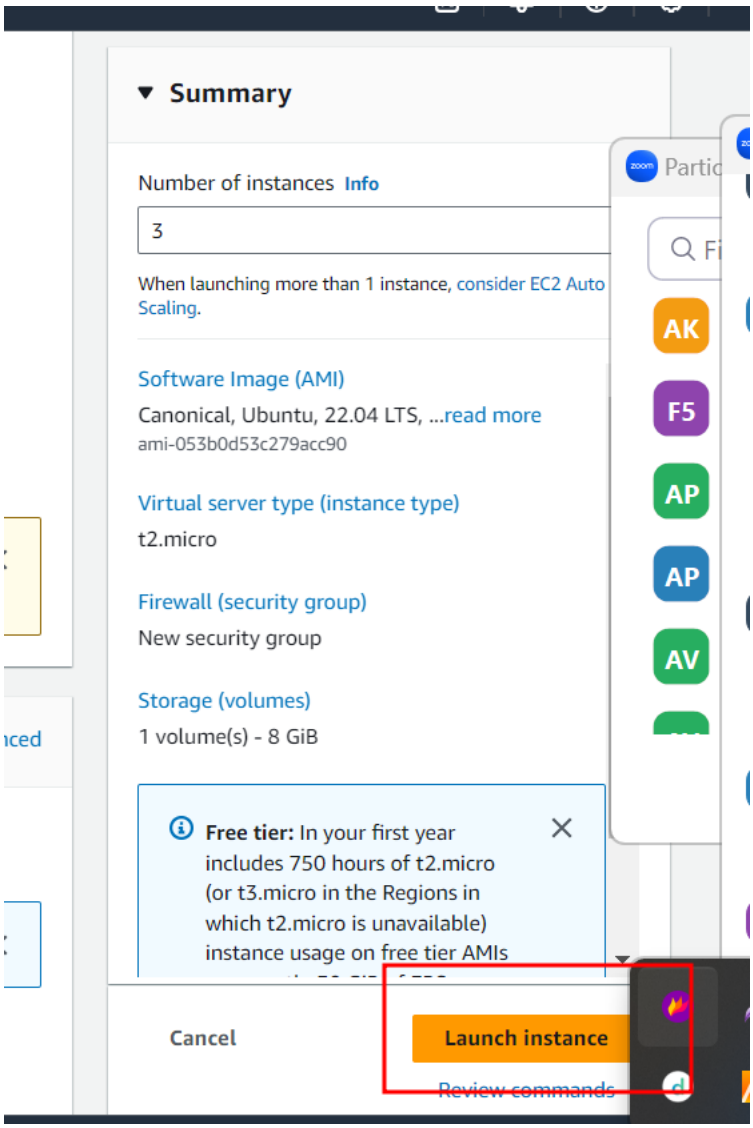

Resources
EC2 Global view
You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:
Instances (running) 0
Auto Scaling Groups 0
Dedicated Hosts 0
Elastic IPs 0
Instances 0
Key pairs 0
Load balancers 0
Placement groups 0
Security groups 1
Snapshots 0
Volumes 0
Launch instance
To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.
Launch instance
Migrate a server
Note: Your instances will launch in the US East (N. Virginia) Region
Service health
AWS Health Dashboard
Region
US East (N. Virginia)
Zones
Zone name
Zone ID
us-east-1a
use1-az4
us-east-1b
use1-az6
us-east-1c
use1-az1
Scheduled events
US East (N. Virginia)
No scheduled events

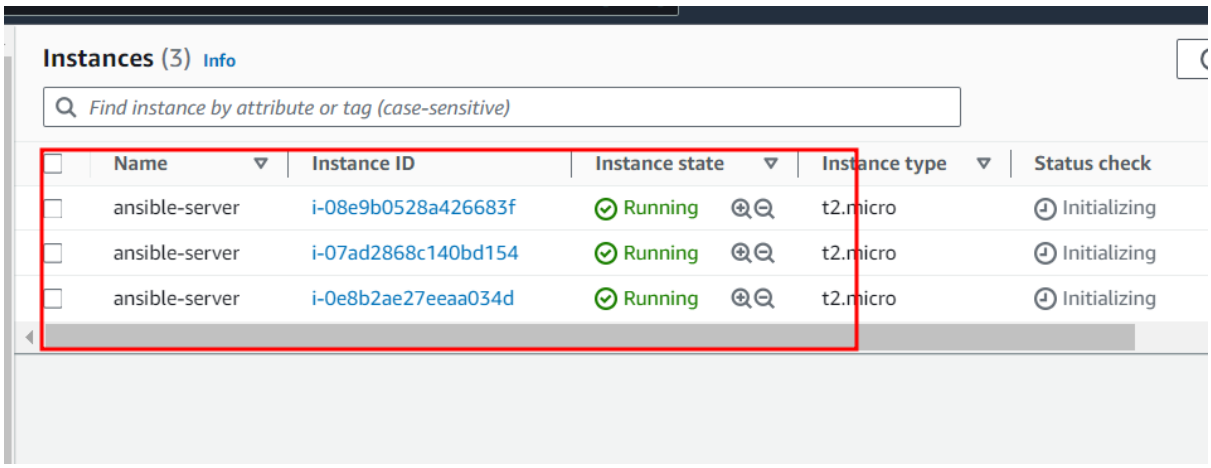

Name and tags Info
Name
ansible-server
Add additional tags
Application and OS Images (Amazon Machine Image) Info
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below
Search our full catalog including 1000s of application and OS images
Quick Start
Amazon Linux
macOS
Ubuntu
Windows
Red Hat
SUSE Li
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community
Amazon Machine Image (AMI)
Ubuntu Server 22.04 LTS (HVM), SSD Volume Type
ami-053b0d53c279acc90 (64-bit (x86)) / ami-0a0c8eebcdd6dcbd0 (64-bit (Arm))
Free tier eligible
Summary
Number of instances Info
1
Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ...read mor
ami-053b0d53c279acc90
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of
Cancel
Launch i

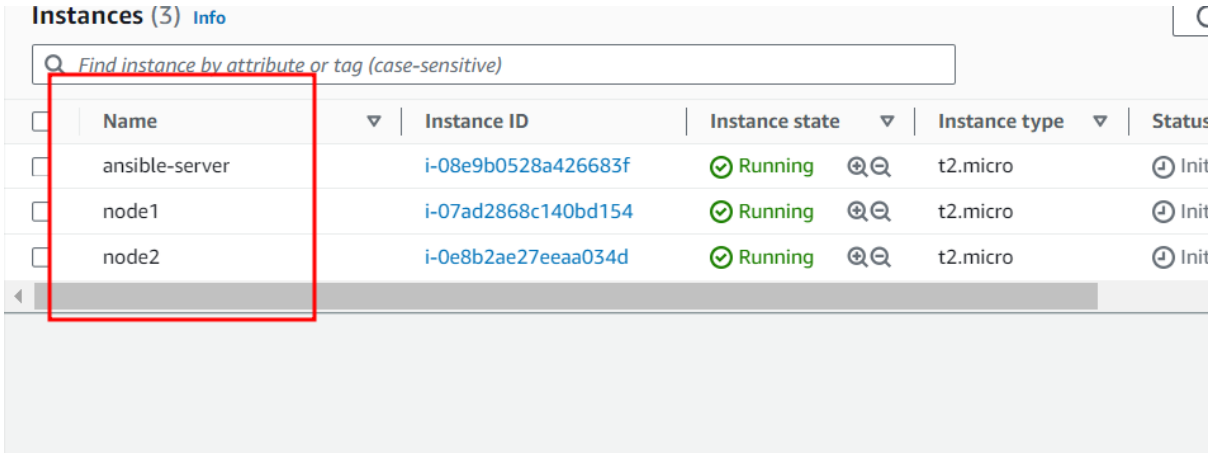

aws
Mac
ubuntu
Microsoft
Red Hat
SUS
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community
Amazon Machine Image (AMI)
Ubuntu Server 22.04 LTS (HVM), SSD Volume Type
ami-053b0d53c279acc90 (64-bit (x86)) / ami-0a0c8eebcdd6dcbd0 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs
Free tier eligible
Description
Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2023-05-16
Architecture
64-bit (x86)
AMI ID
ami-053b0d53c279acc90
Verified provider
Instance type Info
Instance type
t2.micro
Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand RHEL base pricing: 0.0716 USD per Hour
On-Demand Linux base pricing: 0.0116 USD per Hour
All generations
Compare instance types
Additional costs apply for AMIs with pre-installed software
Summary
Number of instances Info
3
When launching more than 1 instance, consider EC2 Scaling.
Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ...read more
ami-053b0d53c279acc90
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs
Cancel
Launch insta

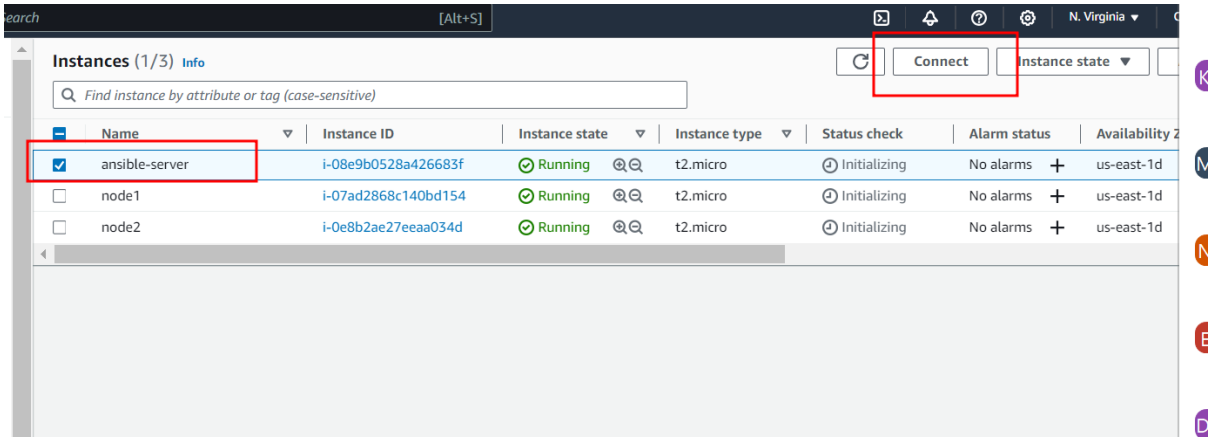

Summary
Create key pair
Key pair name
Key pairs allow you to connect to your instance securely.
my-ansible-server
The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.
Key pair type
RSA
RSA encrypted private and public key pair
ED25519
ED25519 encrypted private and public key pair
Private key file format
.pem
For use with OpenSSH
.ppk
For use with PuTTY
When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. Learn more
Cancel
Create key pair

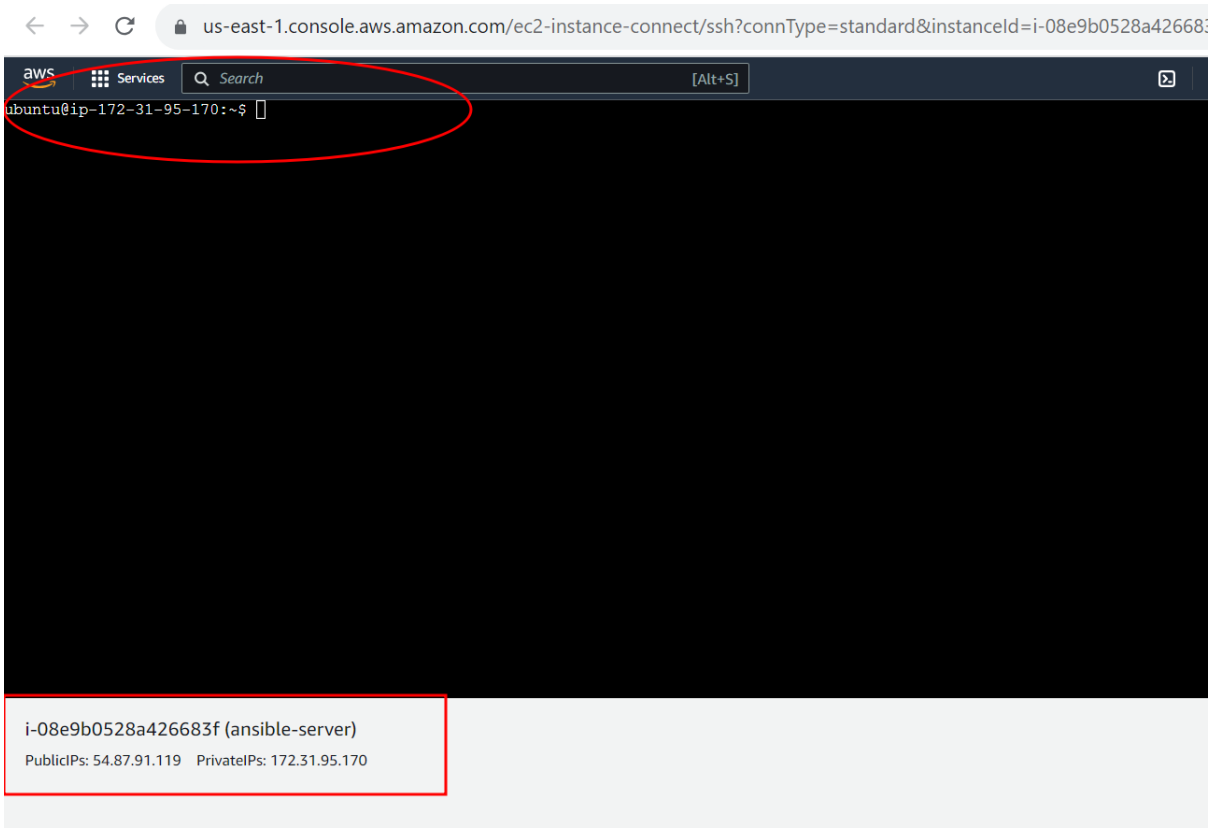

Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.
Create security group
Select existing security group
We'll create a new security group called 'launch-wizard-1' with the following rules:
Allow SSH traffic from
Helps you connect to your instance
Anywhere
0.0.0.0/0
Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server
Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.
Scaling.
Software Imag
Canonical, Ubu
ami-053b0d53c2
Virtual server
t2.micro
Firewall (secur
New security g
Storage (volur
1 volume(s) - 8
Free tie
include
(or t3.n


Summary
Number of instances Info
3
When launching more than 1 instance, consider EC2 Auto Scaling.
Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ...read more
ami-053b0d53c279acc90
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs
Cancel
Launch instance
Review commands


Instances (3) Info
Find instance by attribute or tag (case-sensitive)
Name
Instance ID
Instance state
Instance type
Status check
ansible-server
i-08e9b0528a426683f
Running
t2.micro
Initializing
ansible-server
i-07ad2868c140bd154
Running
t2.micro
Initializing
ansible-server
i-0e8b2ae27eeaa034d
Running
t2.micro
Initializing


Instances (3) Info
Find instance by attribute or tag (case-sensitive)
Name
Instance ID
Instance state
Instance type
Status
ansible-server
i-08e9b0528a426683f
Running
t2.micro
node1
i-07ad2868c140bd154
Running
t2.micro
node2
i-0e8b2ae27eeaa034d
Running
t2.micro


[Alt+S]
N. Virginia
Instances (1/3) Info
Connect
Instance state
Find instance by attribute or tag (case-sensitive)
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
ansible-server
i-08e9b0528a426683f
Running
t2.micro
Initializing
No alarms
us-east-1d
node1
i-07ad2868c140bd154
Running
t2.micro
Initializing
No alarms
us-east-1d
node2
i-0e8b2ae27eeaa034d
Running
t2.micro
Initializing
No alarms
us-east-1d



Please open ansible node

And install the ansible software

sudo su to move root user

or

sudo -i

sudo apt-get update

sudo apt-get install software-properties-common

sudo apt-add-repository ppa:ansible/ansible

sudo apt-get update

sudo apt-get install ansible

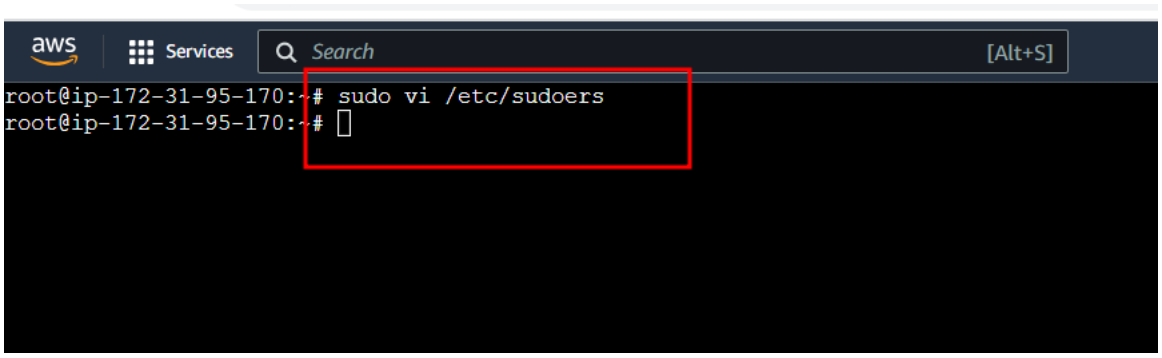
```
  python version = 3.10.6 (main, Mar 10 2023, 10:55:28) [GCC 11.3.0]
  jinja version = 3.0.3
  libyaml = True
root@ip-172-31-95-170:~# sudo vi /etc/ansible/hosts
root@ip-172-31-95-170:~# sudo vi /etc/ansible/hosts
root@ip-172-31-95-170:~# sudo vi /etc/ansible/hosts
```

```
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
#   - Comments begin with the '#' character
#   - Blank lines are ignored
#   - Groups of hosts are delimited by [header] elements
#   - You can enter hostnames or ip addresses
#   - A hostname/ip can be a member of multiple groups

# Ex 1: Ungrouped hosts, specify before any group headers:
[default]
172.31.83.224
172.31.81.187

## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10

# Ex 2: A collection of hosts belonging to the 'webservers' group:

## [webservers]
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110

# If you have multiple hosts following a pattern, you can specify
# them like this:
```

sudo vi /etc/ansible/ansible.cfg

```
# ansible-config init --disabled -t all > ansible.cfg

# For previous versions of Ansible you can check for examples in the
# Note that this file was always incomplete  and lagging changes to
# for example, for 2.9: https://github.com/ansible/ansible/blob/stabl
#
[defaults]
inventory=/etc/ansible/hosts
sudo-user=root
```

Open the file



All configuration ie installing and connect node1 and nod2 using ssh client please follow below steps.

create more than one EC2 instance.

please install these command in ansible server instance

sudo apt-get update

sudo apt-get install software-properties-common

sudo apt-add-repository ppa:ansible/ansible

sudo apt-get update

sudo apt-get install ansible

Now we will add the host details in Ansible Server

sudo vi /etc/ansible/hosts

Add all nodes ip address details inside this file ie private IP Address

now open this file

sudo vi /etc/ansible/ansible.cfg

copy and paste this code

[defaults]

inventory=/etc/ansible/hosts

sudo-user=root

save

esc

:wq!

Now create a user do this steps in all nodes

sudo adduser ansible

This ansible user don't have sudo privileges right now

we need to give the all privileges

sudo vi /etc/sudoers do this step in all nodes

inside this file add below line

%ansible ALL=(ALL:ALL) NOPASSWD:ALL

in all nodes we need to switch to ansible user

su - ansible

After set all details in all nodes

now switch to ansible user in all nodes and run the command as

sudo apt-get update

Now we need to establish the conntion

Now in Ansible node type as

ssh PrivateIPAddressOfNode to connect node1 or node2 to do some task

May you will get the Output as permission denied.

This is due to SSH restrictions.

move the root user ie exit to this task in all nodes

sudo vi /etc/ssh/sshd_config

and edit the information

PasswordAuthentication yes

PubkeyAuthentication yes

Now please restart the service to this task in all nodes

sudo service ssh restart

sudo service ssh status

Now switch to the Ansible server node

then write

ssh NodeIpAddress

Now from ansible server or master node if you want to connect to node1 or node2 we need to enter the password

So now we will see how to configure ssh key to connect to node1 or node2 from server node

now in master or ansible node write the command as

ssh privateIPAddress

then it will ask passsword

then do some taks like create the file

and then check in node1 in user ansible file created or not.

in ansible server node

ssh-keygen (ssh means Secure Socket Shell,Ssh-keygen is a tool for creating new authentication key pairs for SSH)

please hit enter key again and again

after key generated

ls -a (to check the .ssh folder)

cd .ssh (move inside that folder)

ssh-copy-id ansible@PrivateIPAddressOfNode1

ssh-copy-id ansible@172.31.39.251

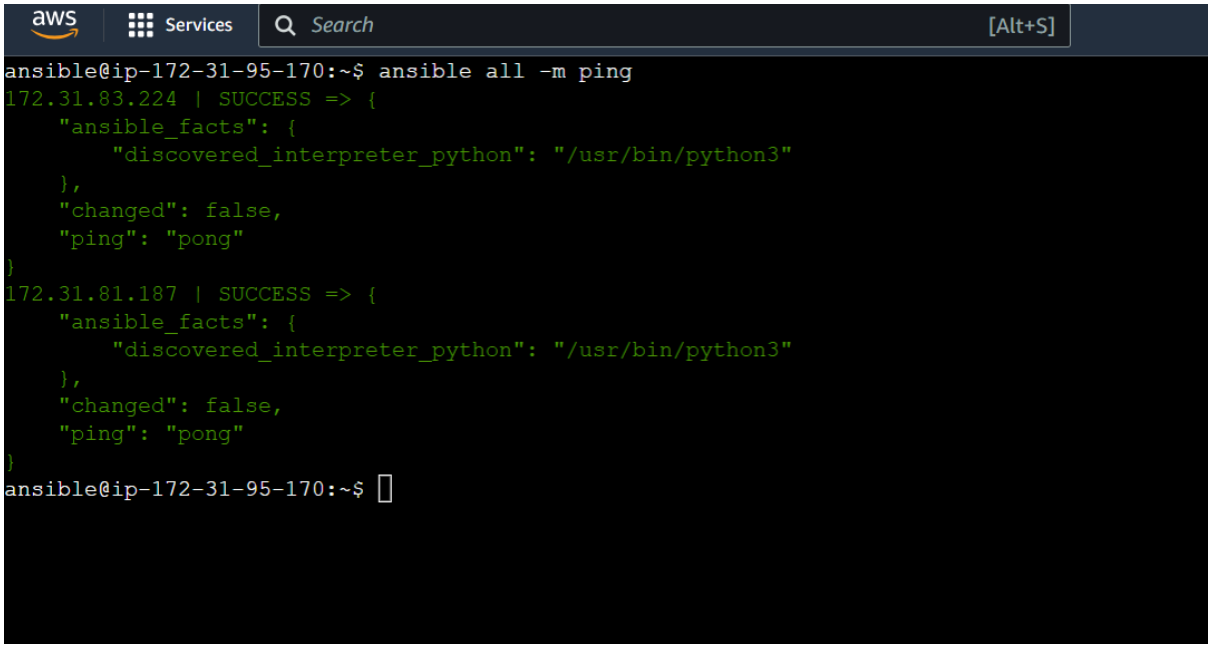
After this we can connect from ansible node to node1 using one command as

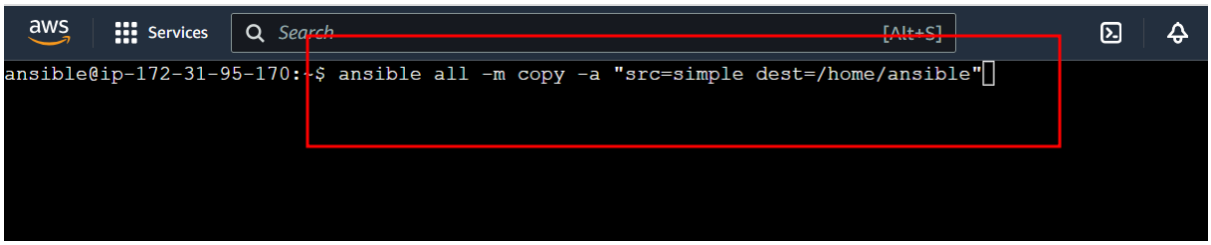
ssh 'ansible@172.31.39.251'

Now we will connect node2 using ssh-copy-id command

please do same for all nodes but while creating ssh-keygen create new file for each node.

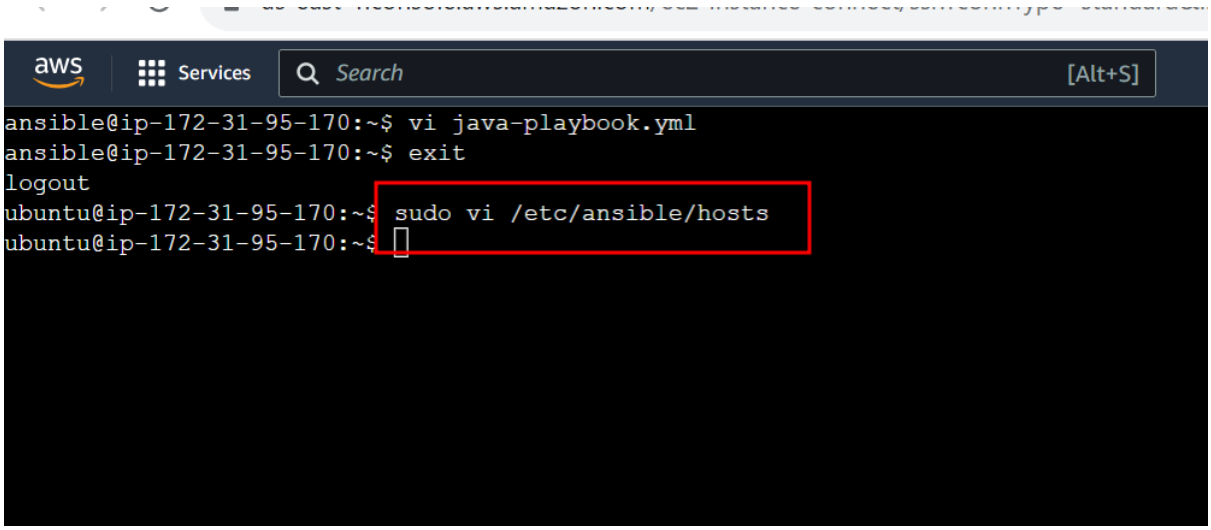
ansible all -m ping





In ansible playbook we can write all module and task details.

To create playbook we need to use yml file



ansible all -m ping

ansible all -m copy -a "src=simple dest=/home/ansible"

vi createfile.yml

```
---
 - hosts: all
   become: true
```

```
  tasks:
    - name: create the files in nodes
      file: path=/home/ansible/test.txt state=touch
```

ansible-playbook createfile.yml

vi java-playbook

```
---
- hosts: all
  tasks:
    - name: Task - 1 Update APT package manager repositories cache
      become: true
      apt:
        update_cache: yes
    - name: Task -2 Install Java using Ansible
      become: yes
      apt:
        name: "{{ packages }}"
        state: present
      vars:
        packages:
          - openjdk-11-jdk
```

ansible-playbook java-playbook.yml

vi nginx-server.yml

```
---
- hosts: webserver
  become: true
  tasks:
    - name: install nginx
      apt: name=nginx state=latest
    - name: start nginx
      service:
          name: nginx
          state: started
```

ansible-playbook nginx-server.yml

this time this playbook install nginx server on node1

please open node1 public ip address to check nginx server running or not.

vi nginx-server-stop.yml

```
---
- hosts: webserver
  become: true
  tasks:
    - name: stop nginx
      service:
          name: nginx
          state: stopped
```

ansible-playbook nginx-server-stop.yml