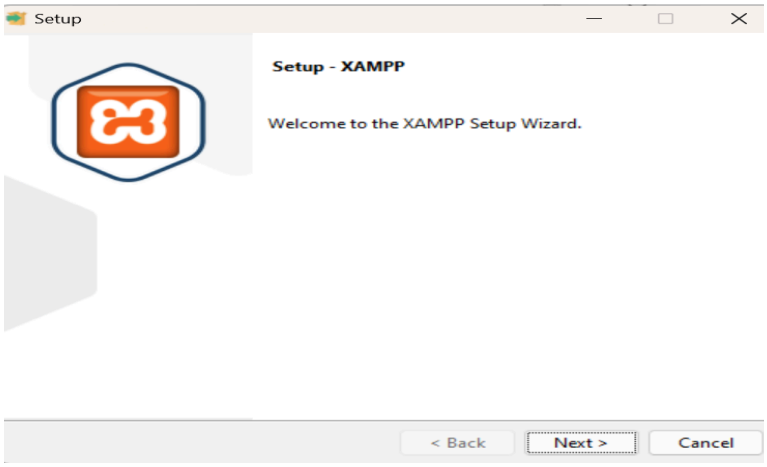
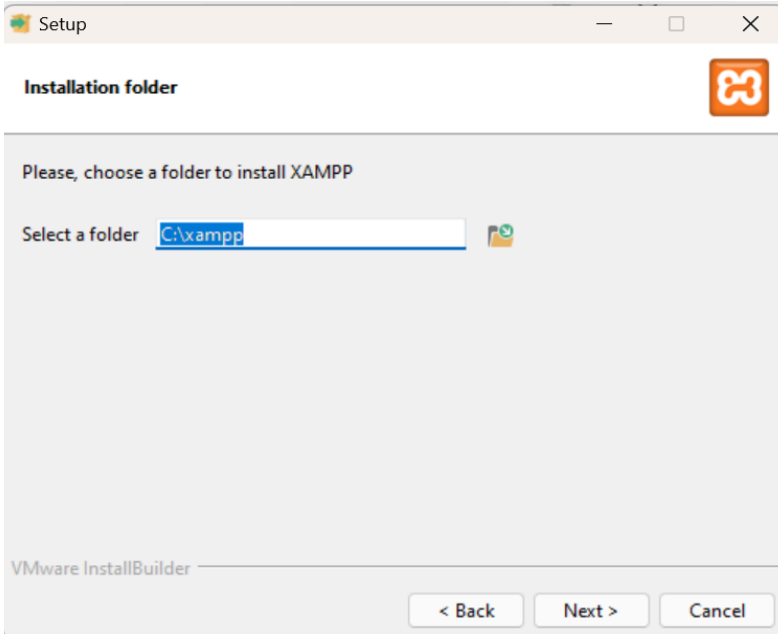
Experiment 1A

Hosting a static site using xampp

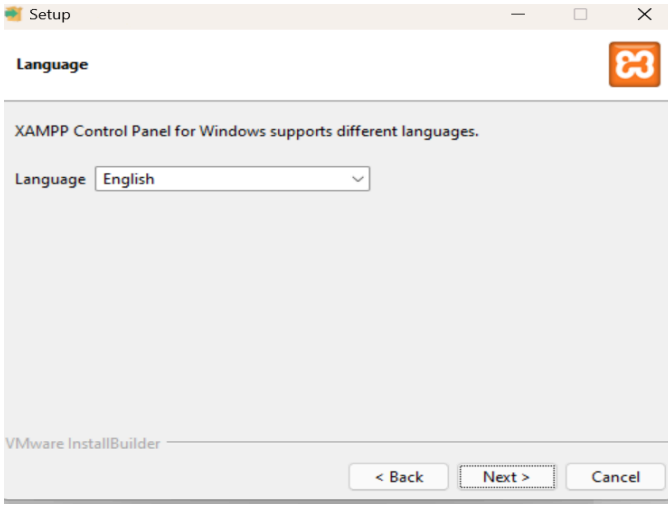
1. Download and open the xampp application and click on zip and extract it then click it.



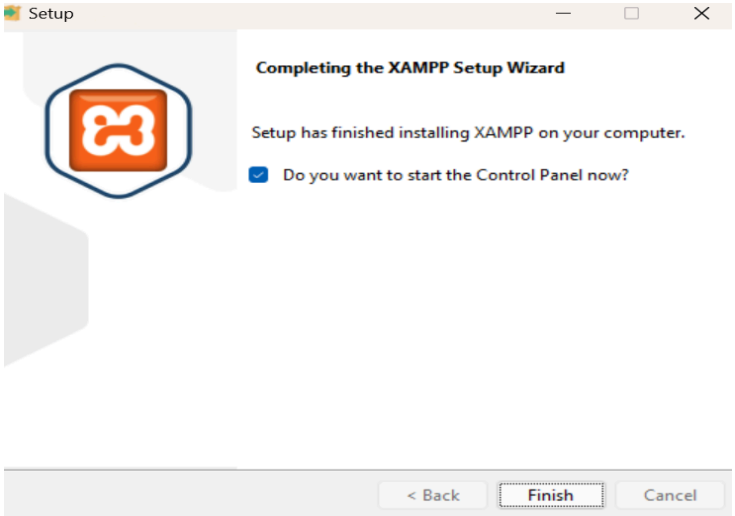
2. Then select the c directory.



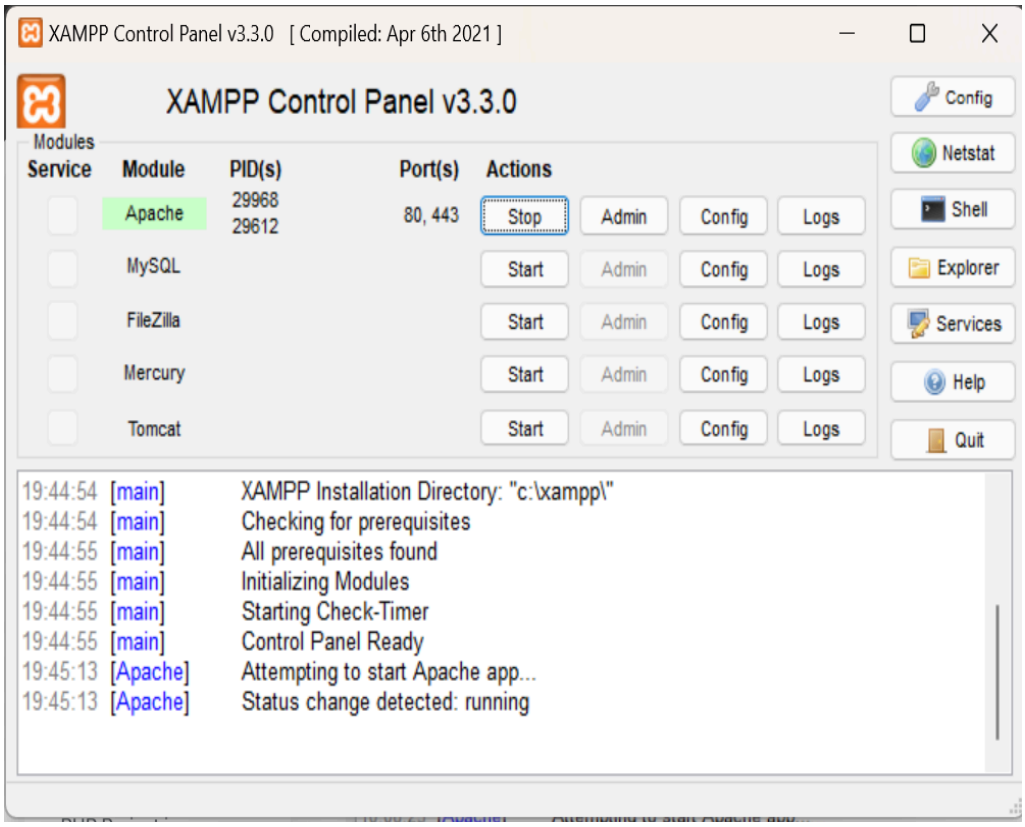
3. Then select language as english.



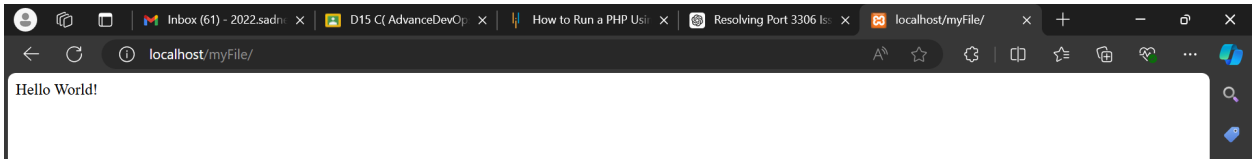
Then install and get started.



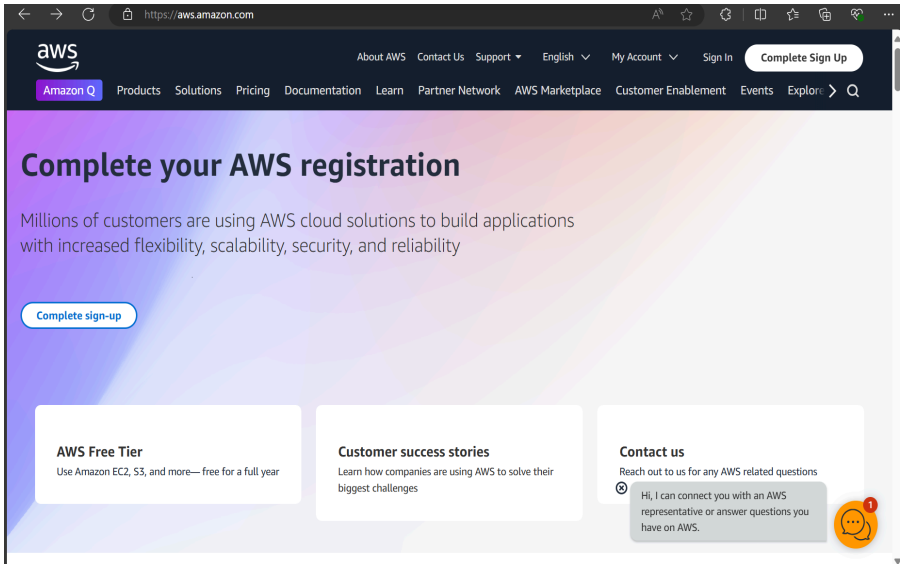
Then the final control panel opens.



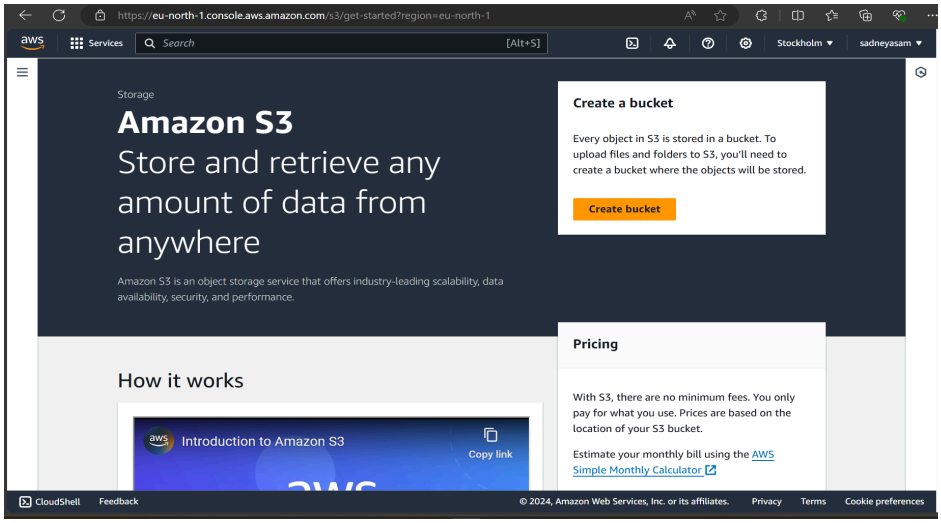
Then I created a file myFile inside htdocs folder and uploaded a sample file.and then opened https:/localhost/myFile/in.php

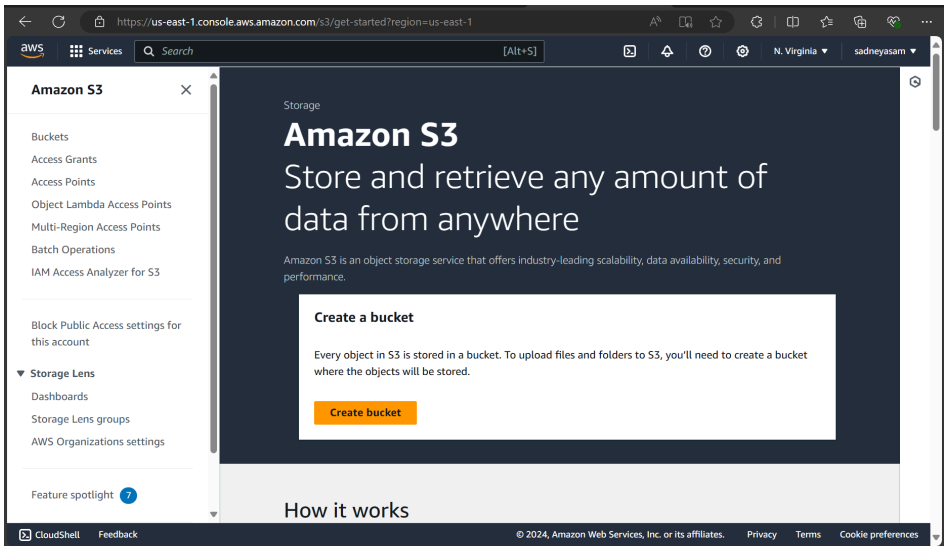


Hosting a static site using S3 bucket:



1. I clicked on create bucket.

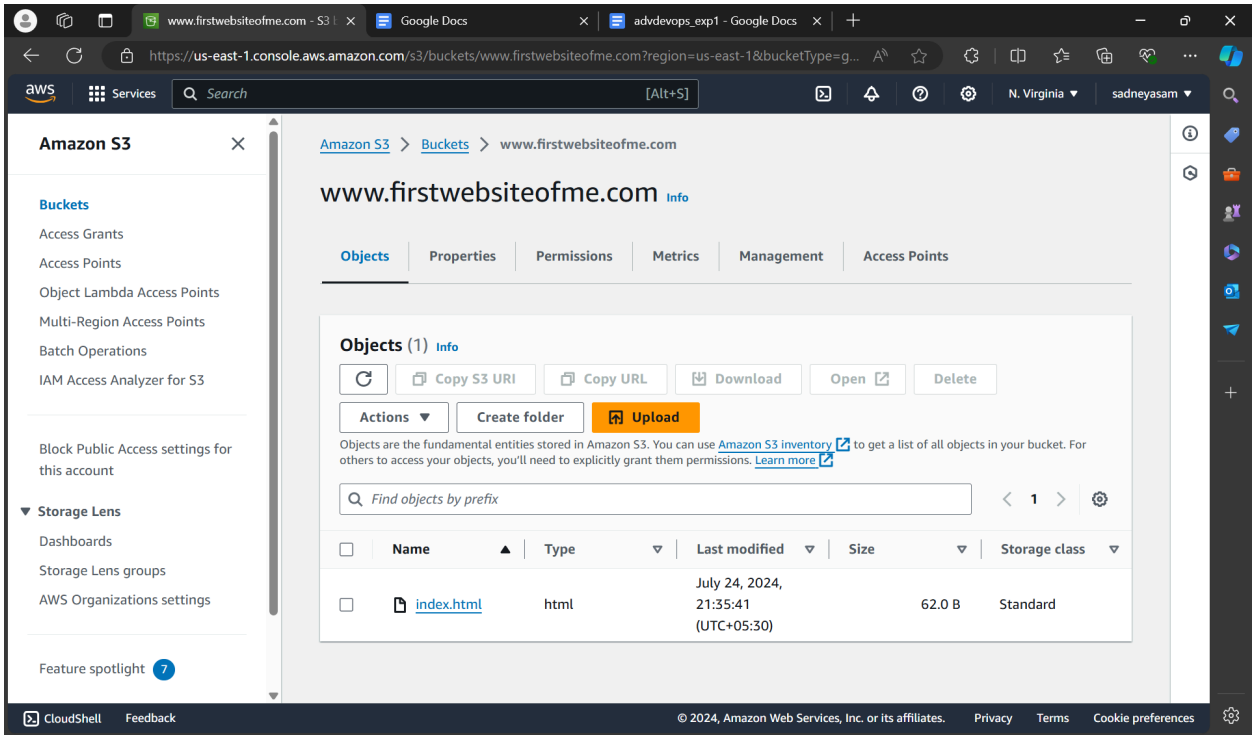




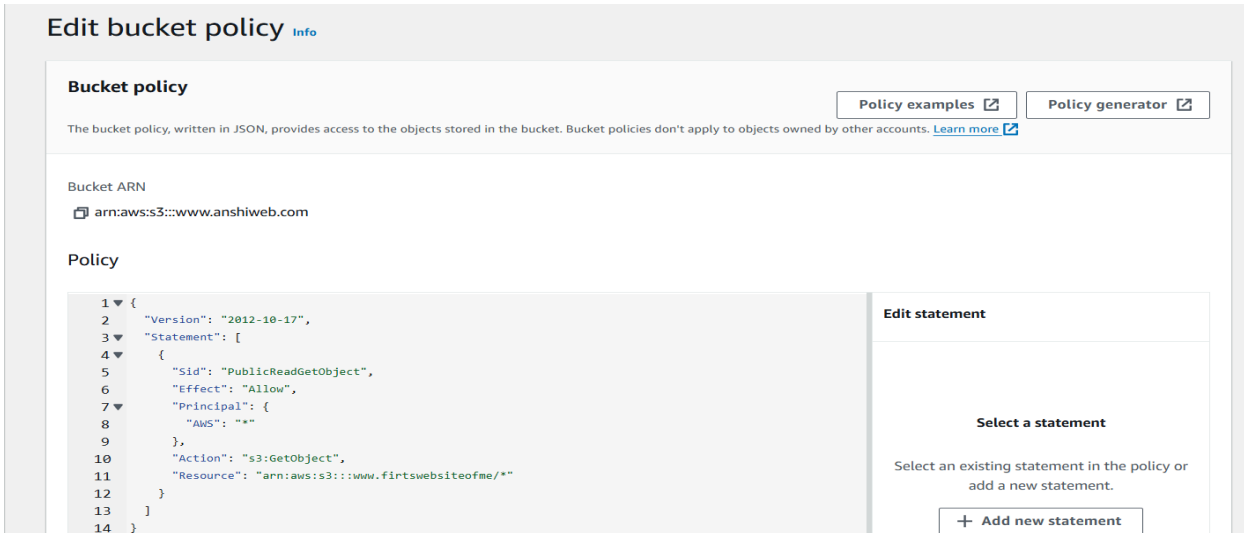
3. Then I created a bucket named www.firstwebsiteofme.com

○	www.firstwebsiteofme.com	US East (N. Virginia) us-east-1	View analyzer for us-east-1	July 24, 2024, 21:35:15 (UTC+05:30)

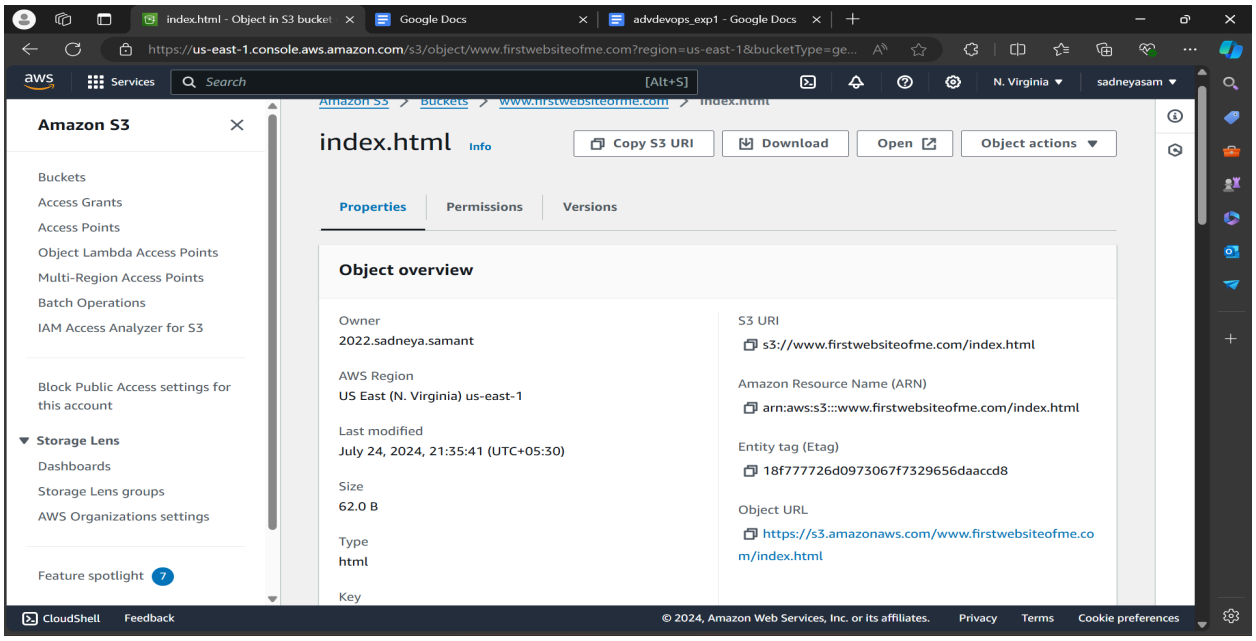
I clicked on the bucket.



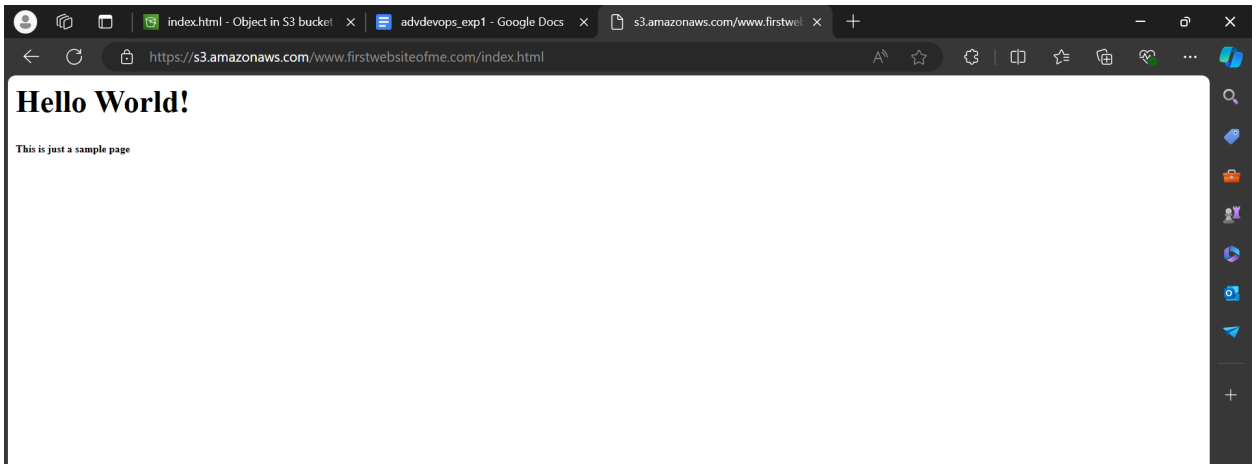
4. Then I also S3 webhosting-bucket policy with this code .I entered in the permission section there is an option edit bucket policy. Where I uploaded the code and saved it.



4. Then i clicked on index.html file and opened its properties section.



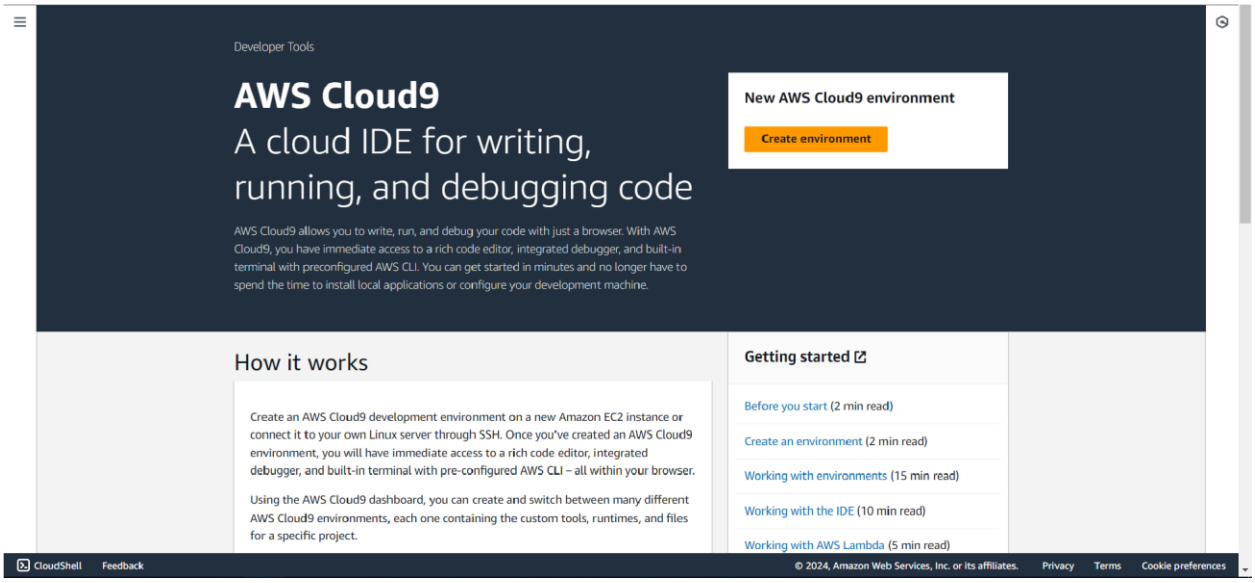
4. Then I clicked on https://s3.amazonaws.com/www.firstwebsiteofme.com/index.html which given me the final output.



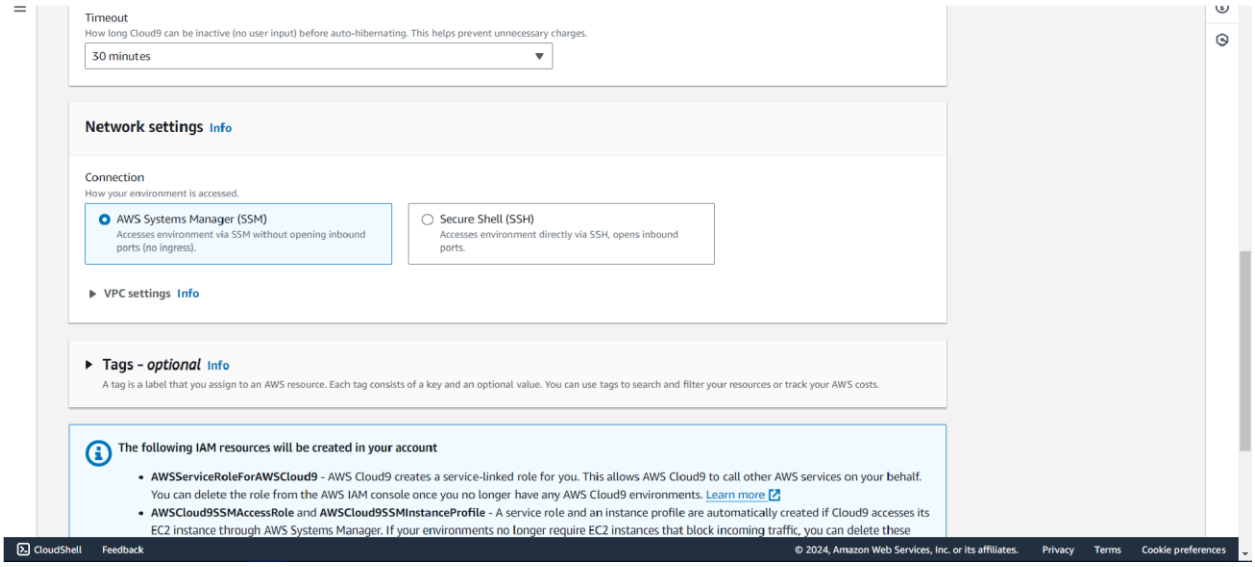
Name: Muskan Chandiramani
Div: D15C
Roll No.5

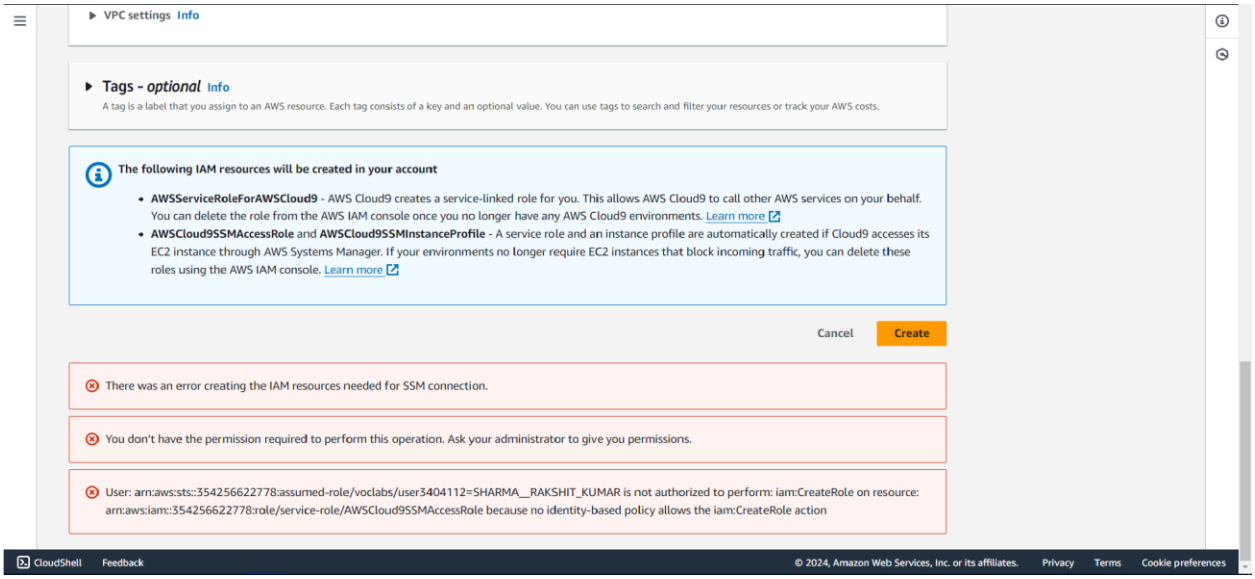
Experiment 1B: IAM and cloud9

1.Open the AWS account and search for Cloud9.

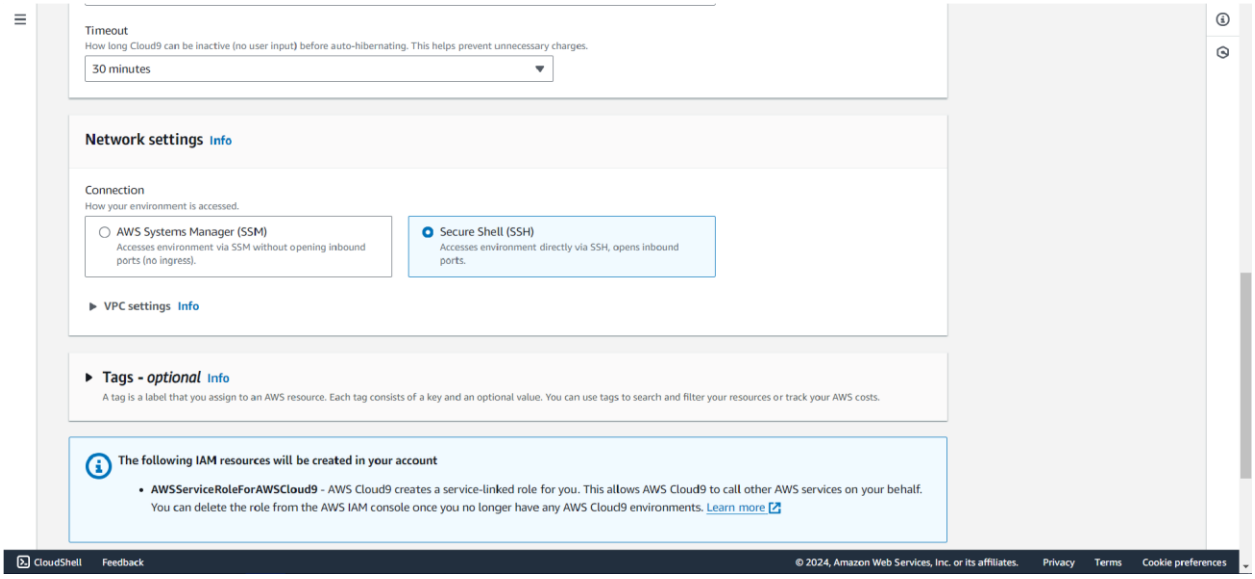


2. Enter the name and other required configuration for creating an environment. In network settings, using the AWS system manager gives an error while creating the environment

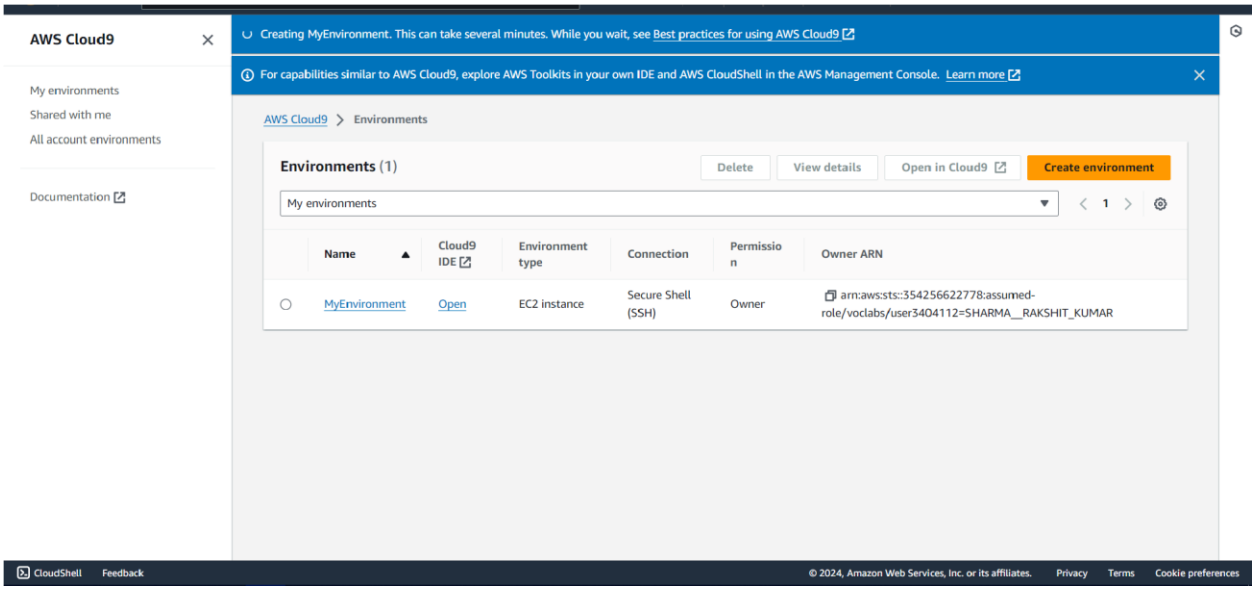




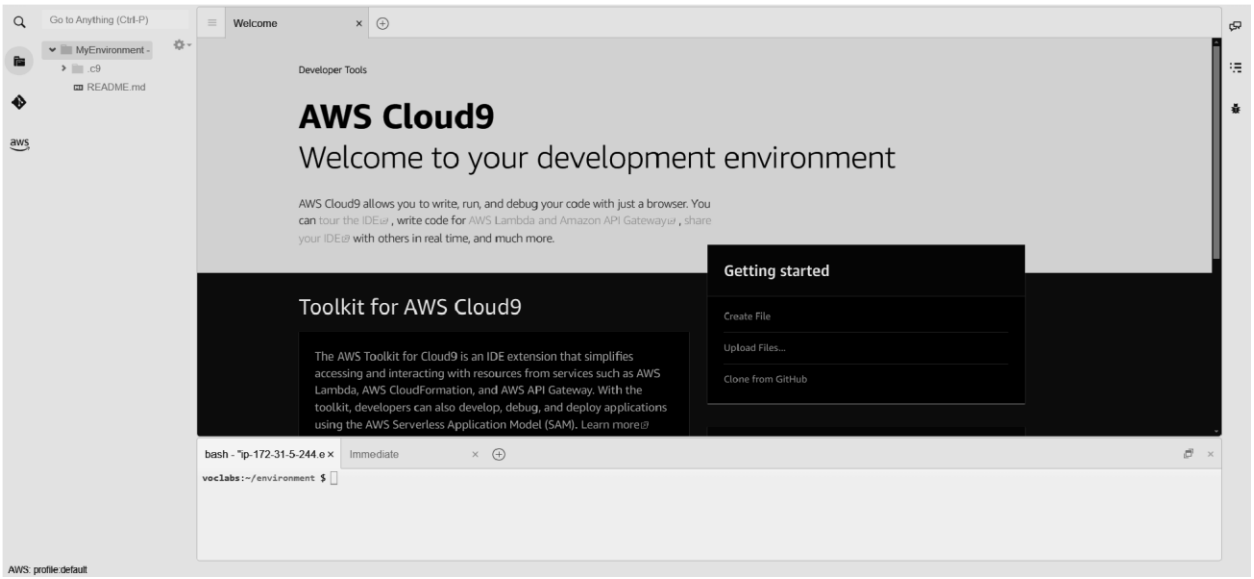
3. Use the Secure Shell option in Network settings.



4. Once the configuration is complete, click on create environment to create a Cloud9 environment.

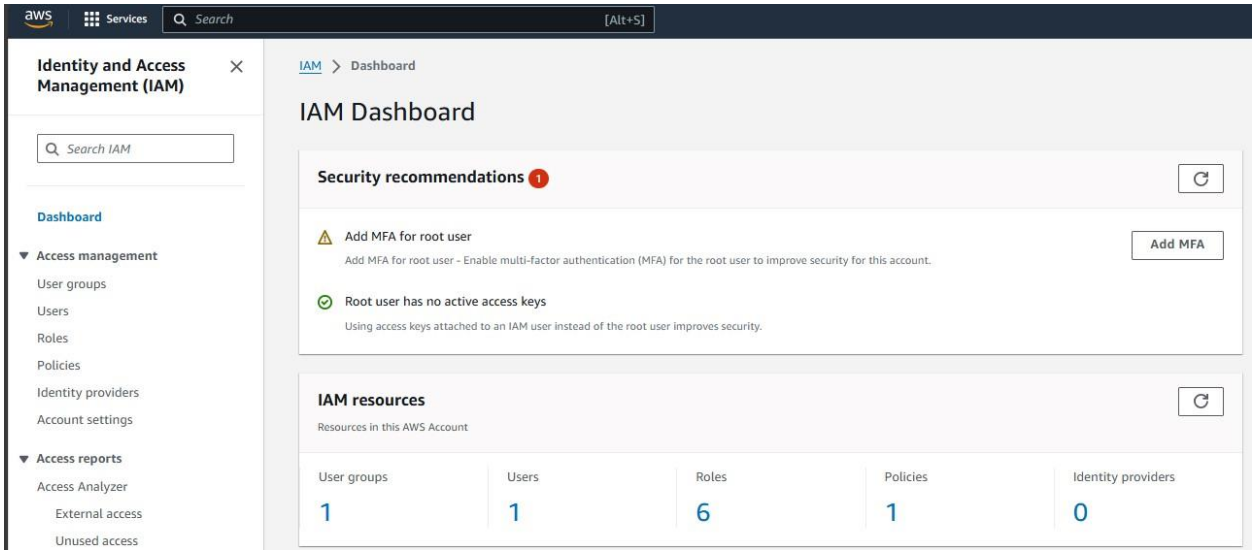


5. Cloud9 Environment.is opened when u click on the environment name

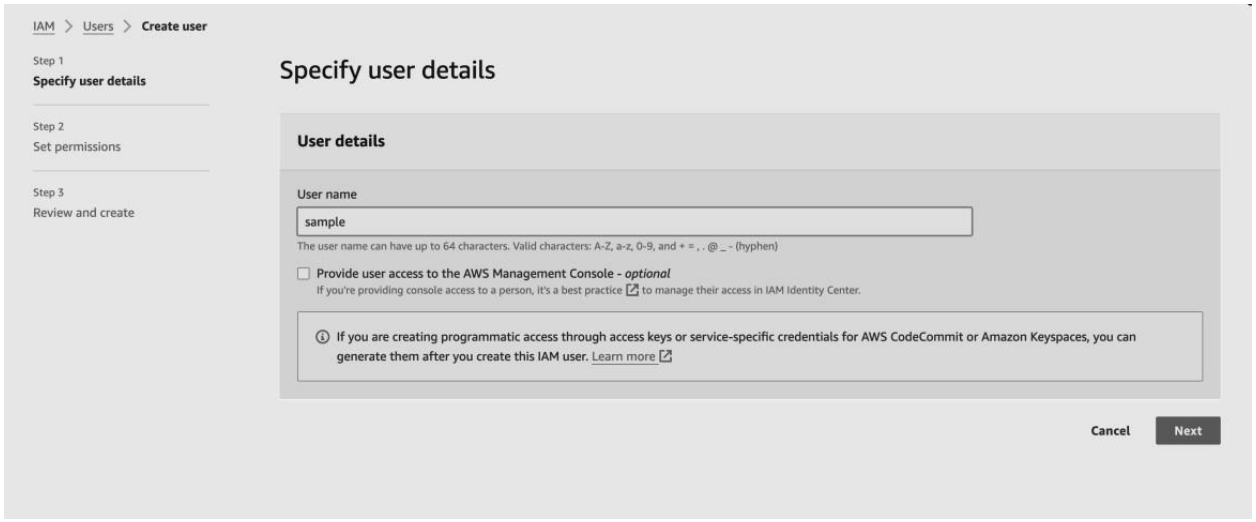


IAM user creation steps

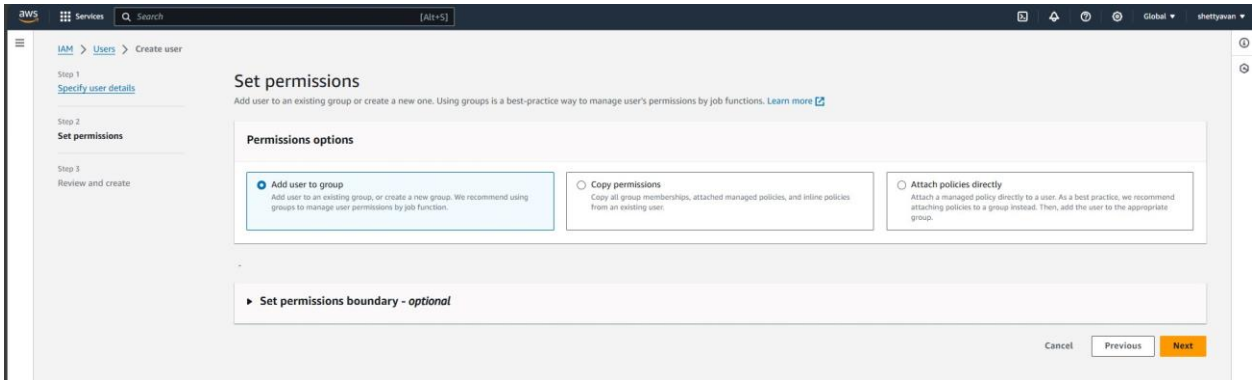
1. Open the aws account and search for IAM in service.



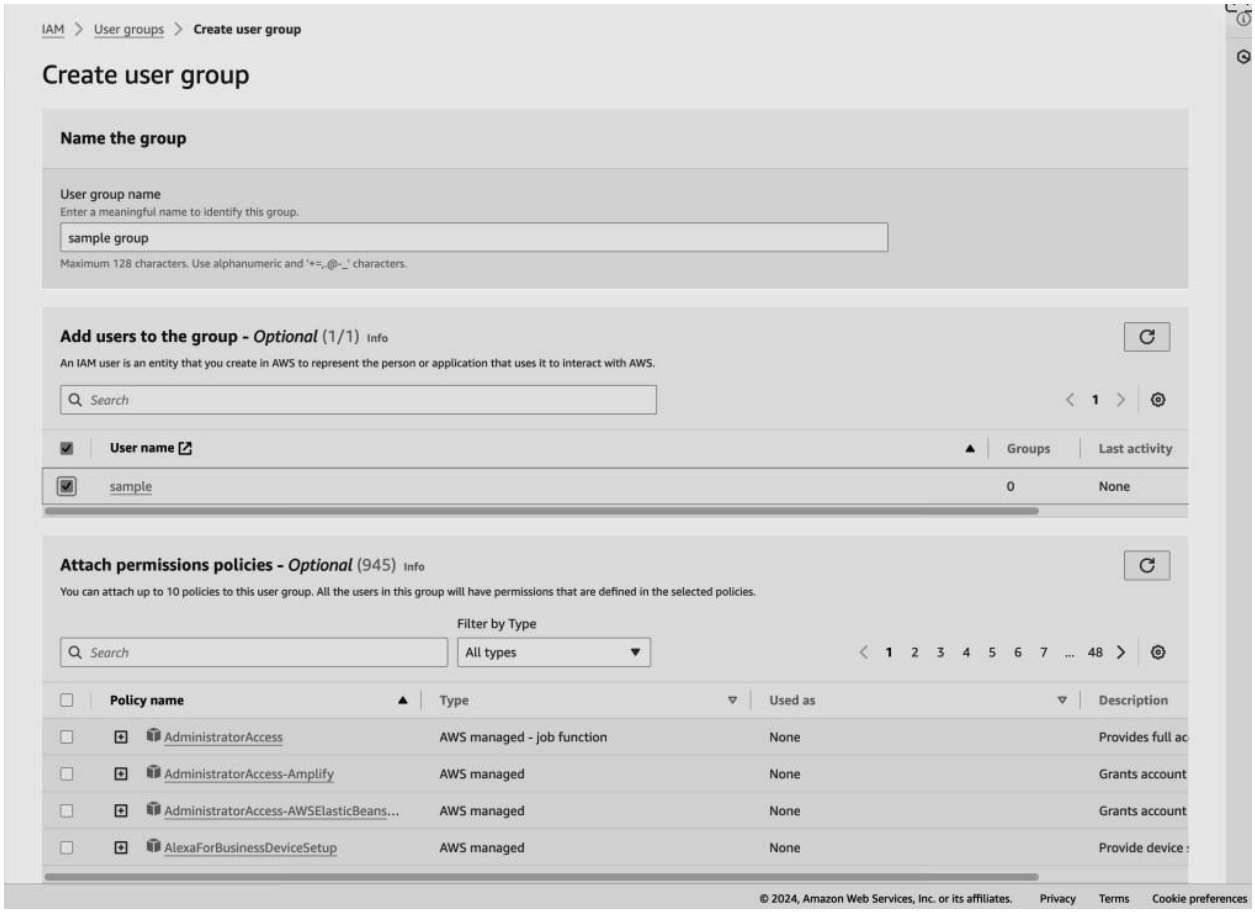
2. Select the users option from the left panel and click on create user button.Give the user name,



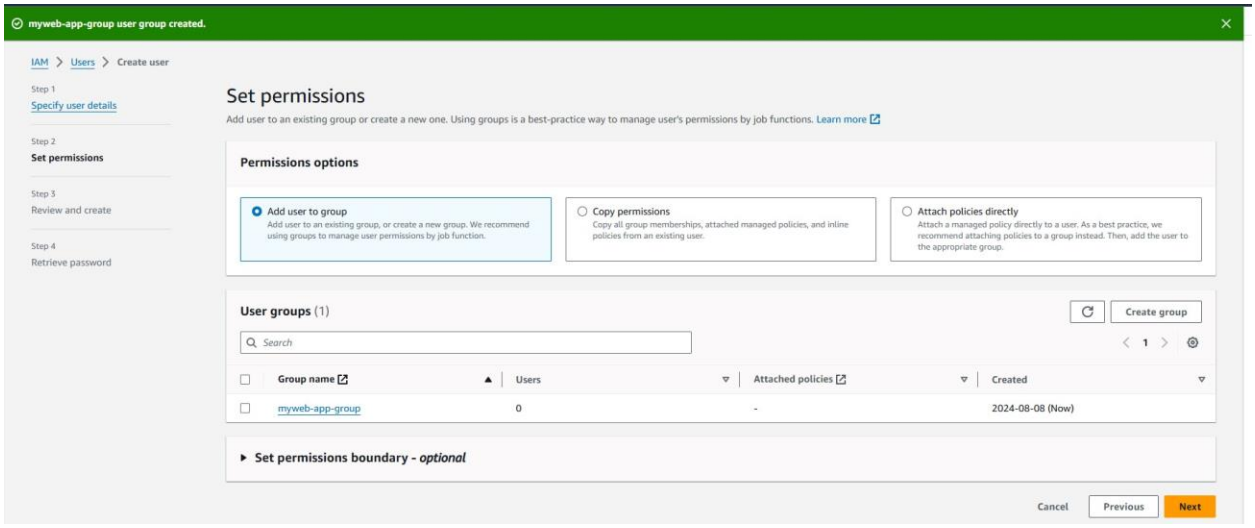
3. Click the add user option if you don't have an existing user group



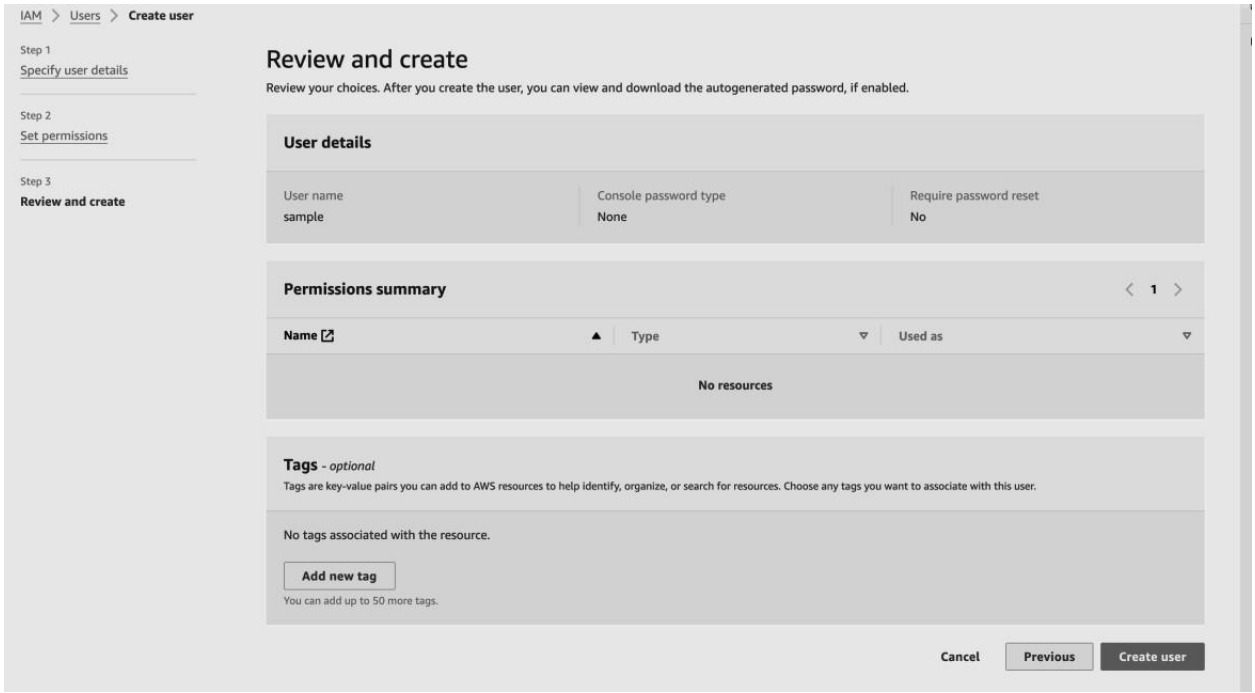
4. Give a name to your user group and check the policies if required any



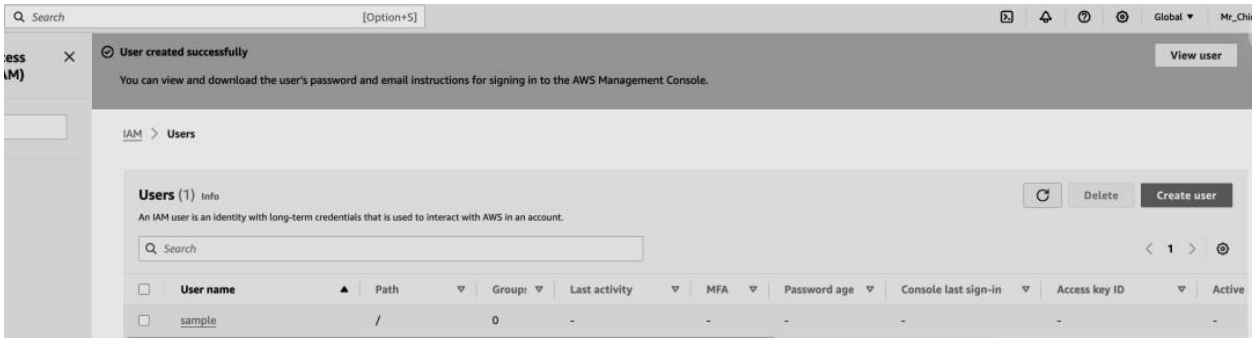
5. Once the user group is created select the name and click next to create your user



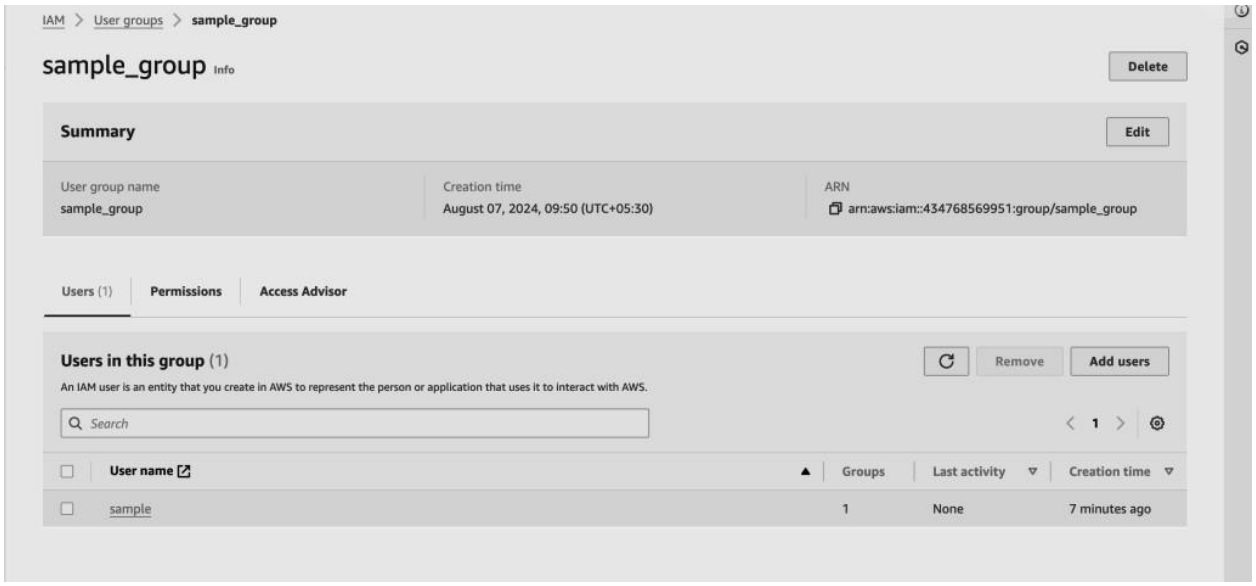
6. Review the configuration details and check if you have missed any steps and then click on ‘Create user’ button



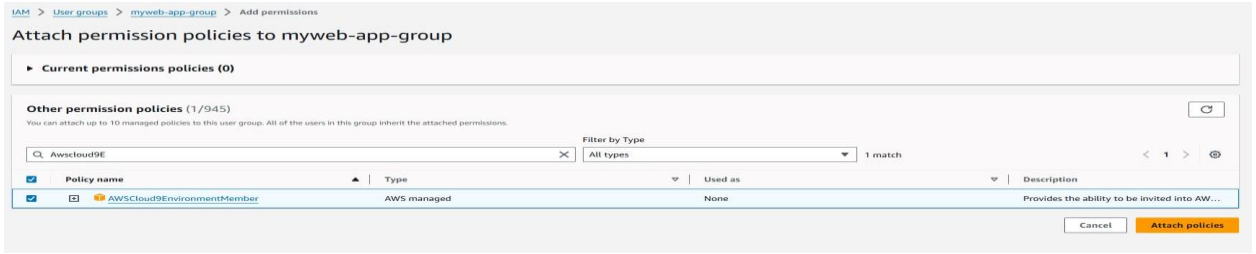
7. You will see the “user created successfully” message and incase you need then store your password by downloading the csv file



8. After creation of the user, go to the user groups tab and select the group in which the user has been added. Navigate to the permissions tab. Click on add permissions and attach policy.



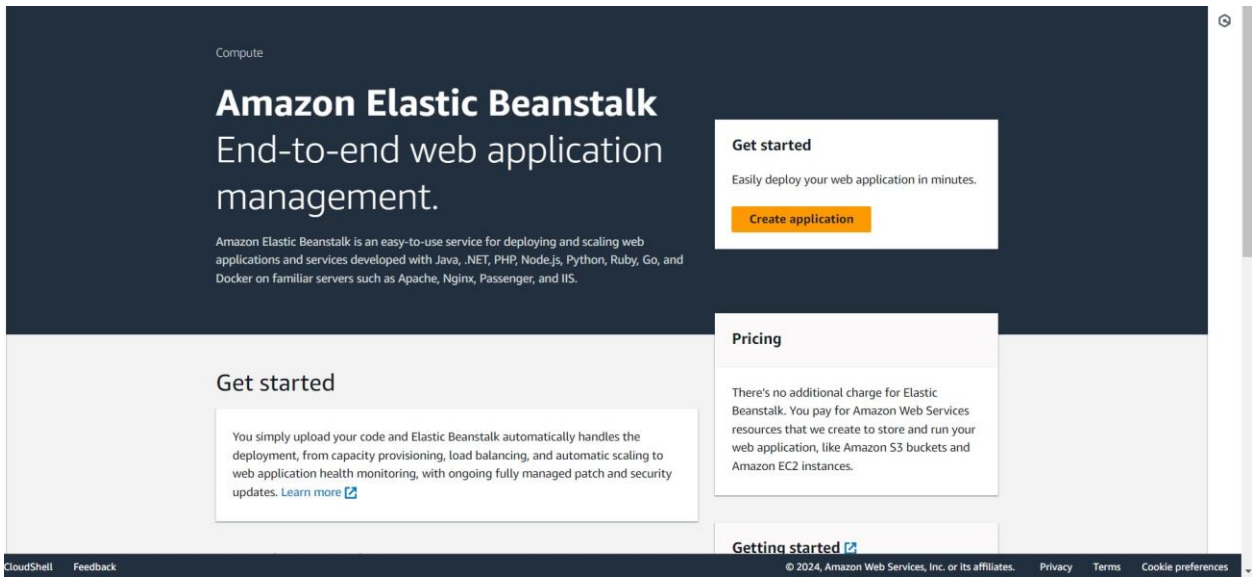
9. Search for the “AWSCloud9EnvironmentMember” policy and attach it.



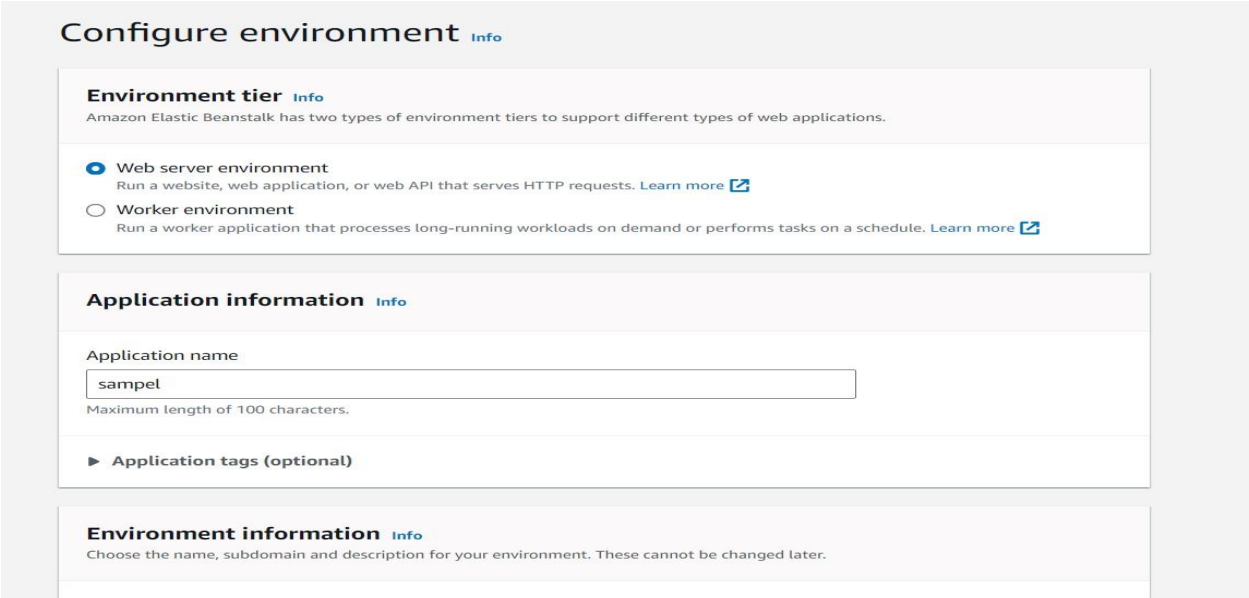
Name: Muskan Chandiramani
Div: D10C
Roll No:5

Practical No 2 : Elastic Beanstalk

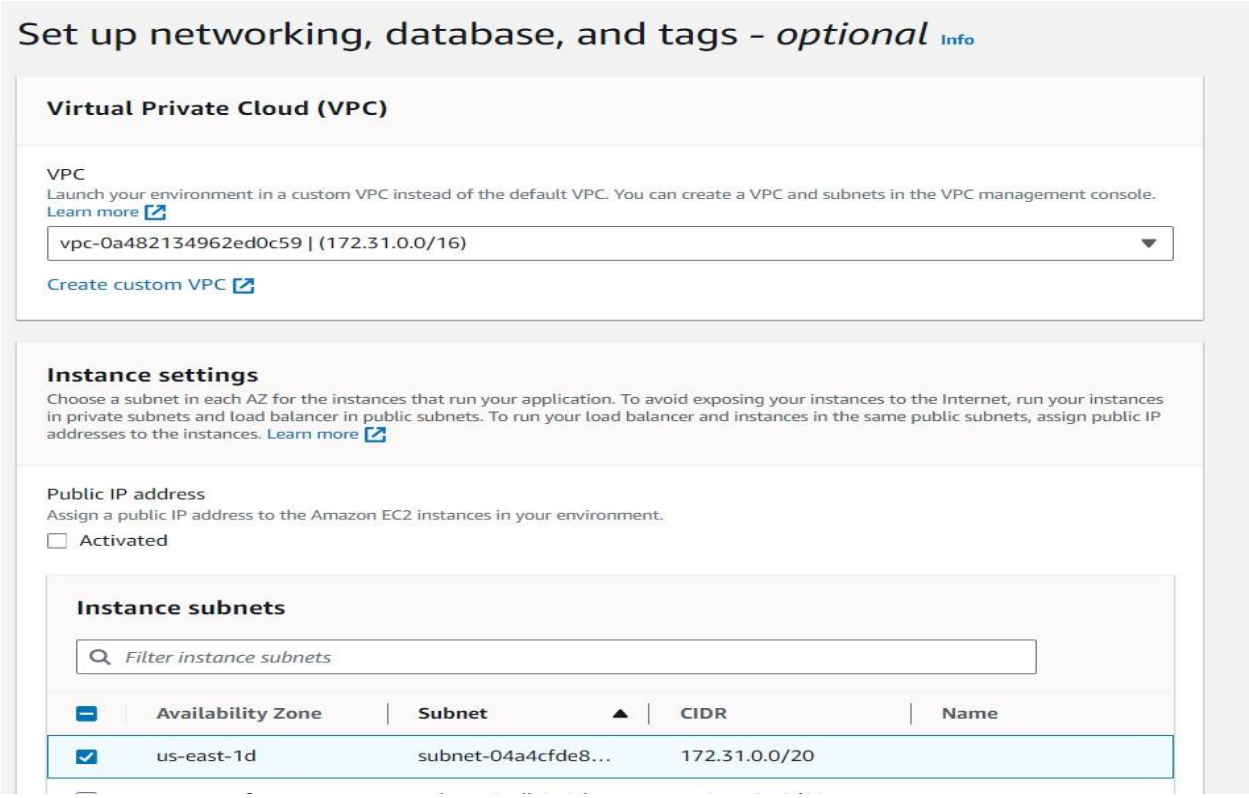
1) Go to services and choose elastic Beanstalk.following page will appear.



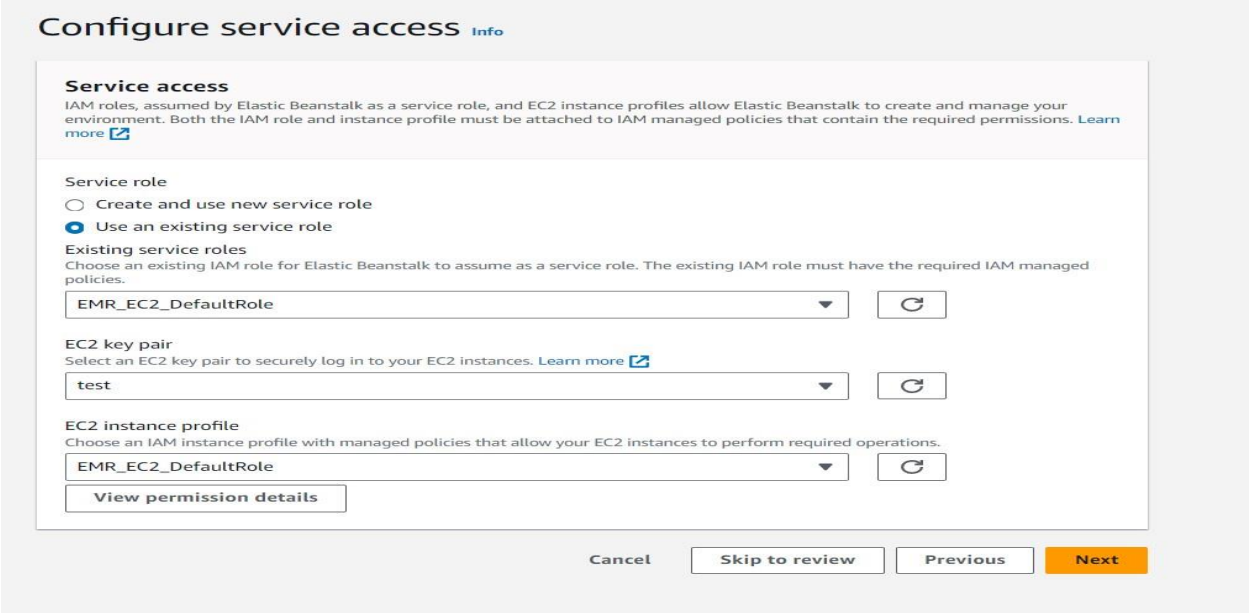
2) Configure the environment. Give the application name, check domain availability and choose PHP as platform.Then click next.

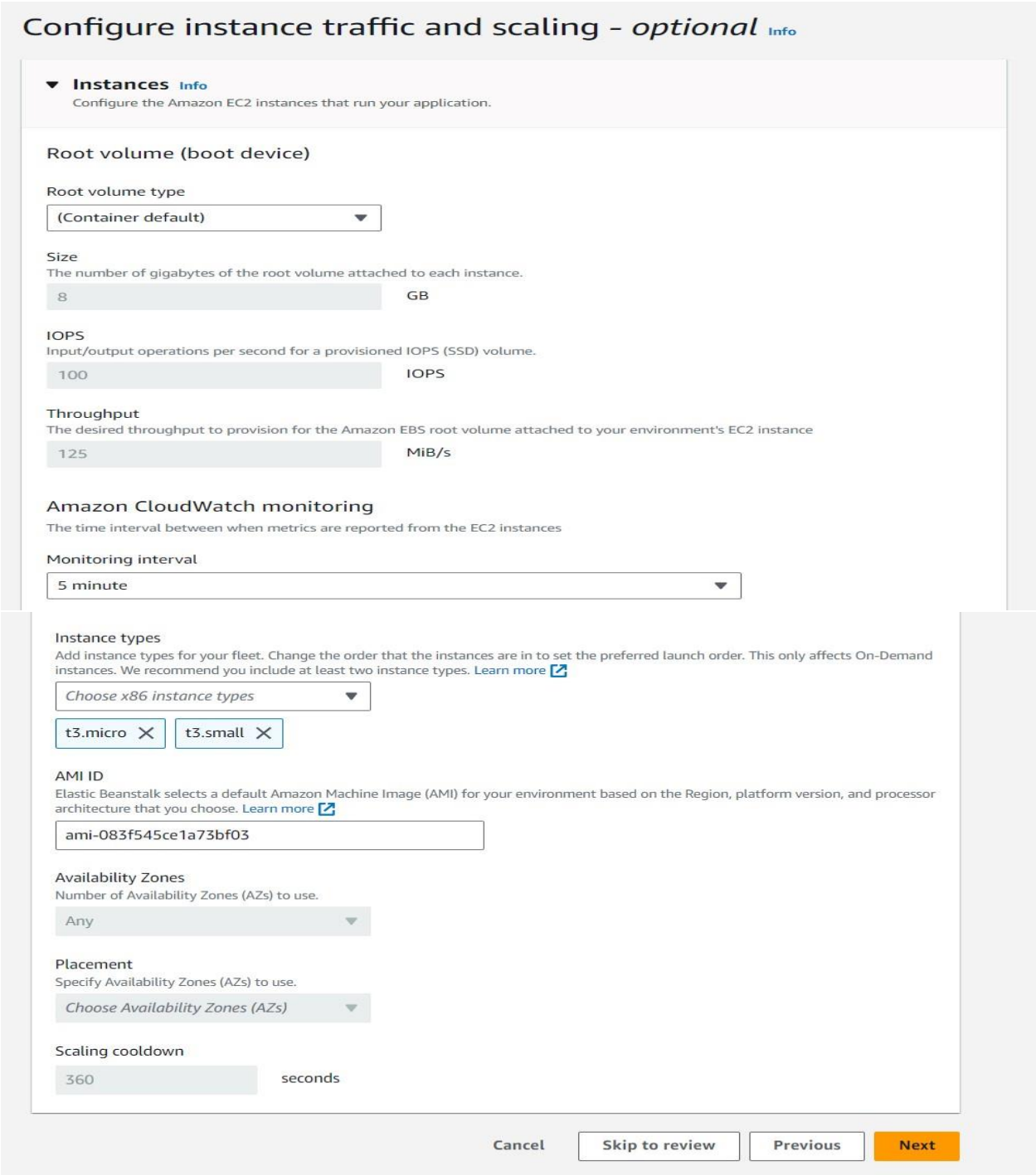


3) Configure the service access. 4) Choose one of the available VPC and instance subnet.Click next.

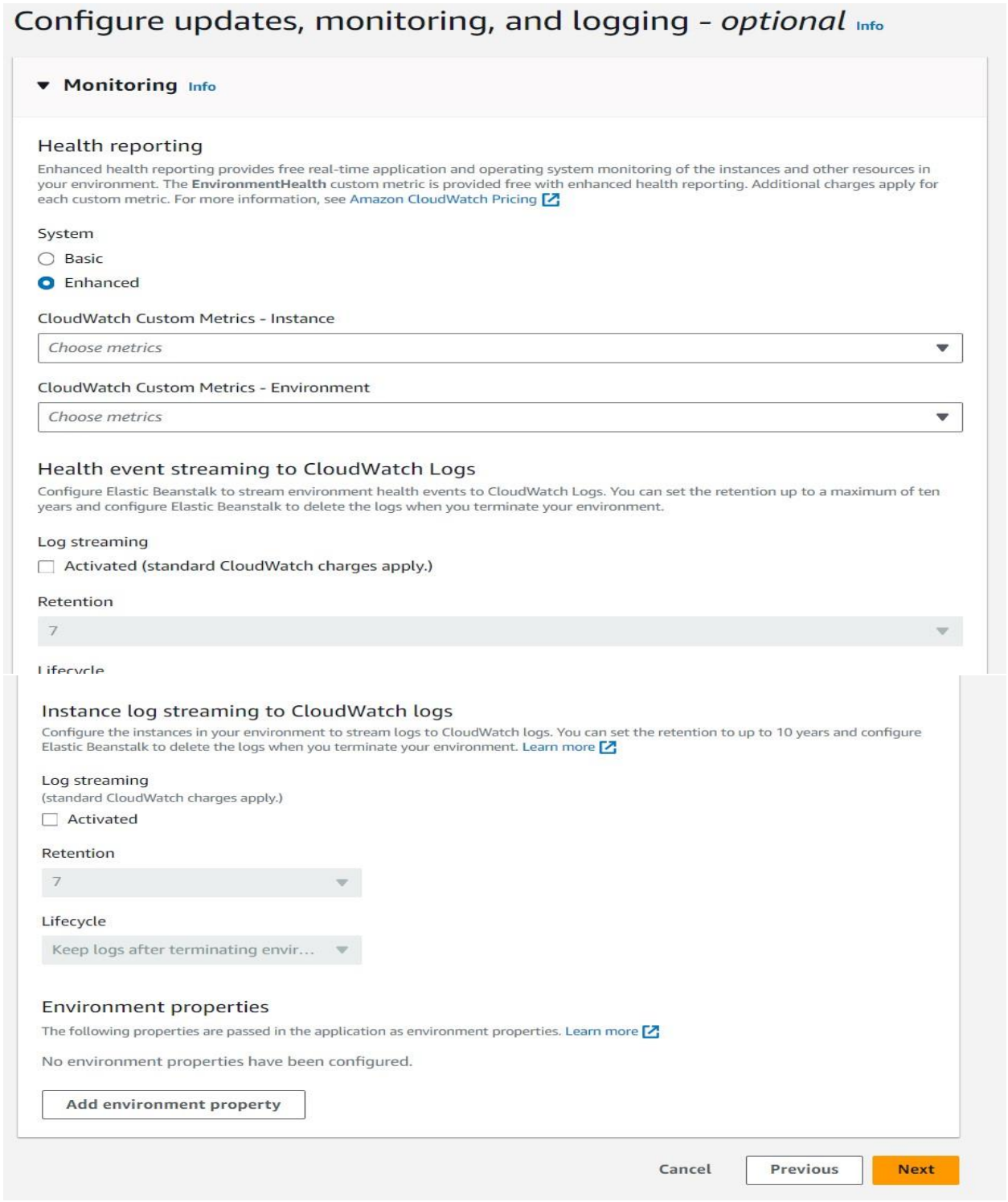


5) Configure instance traffic and scaling. Keep all the options as default.

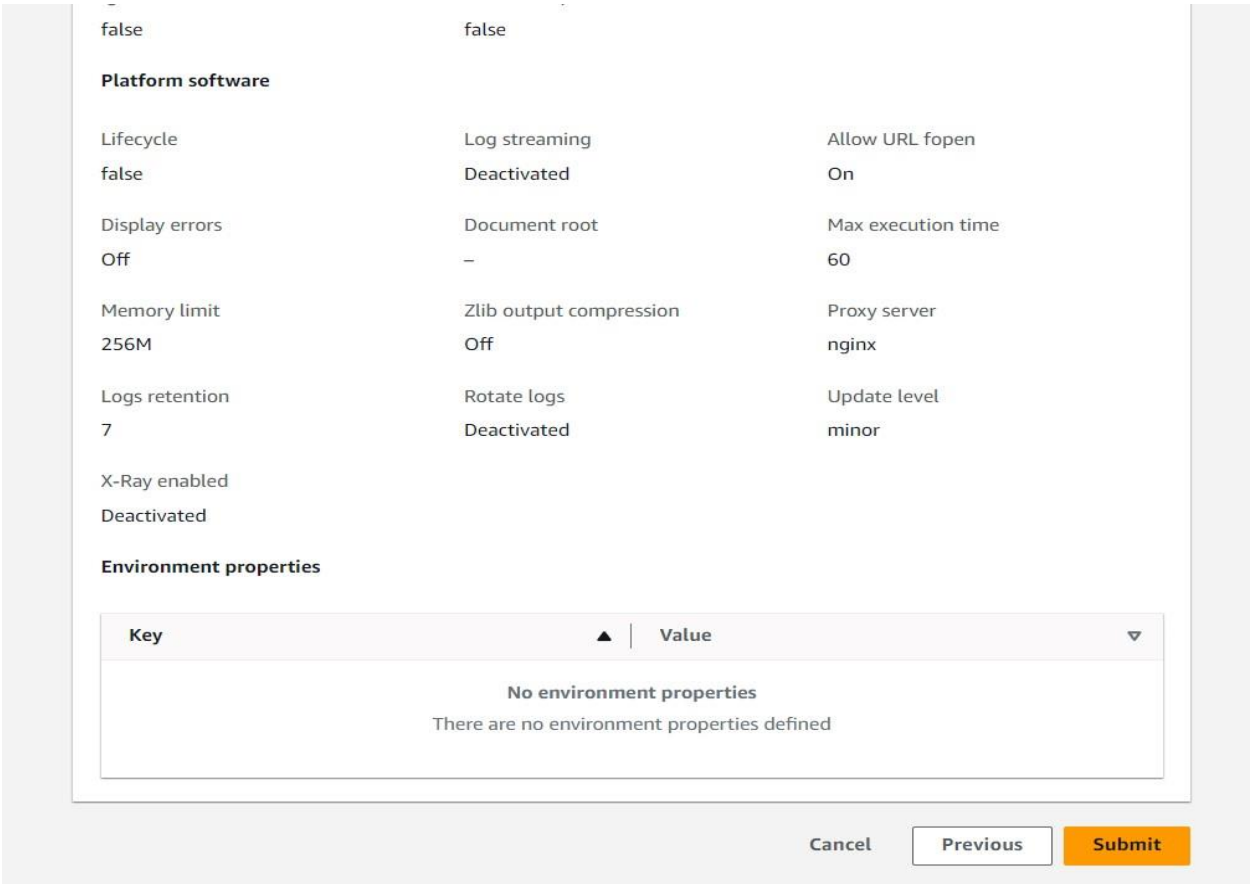




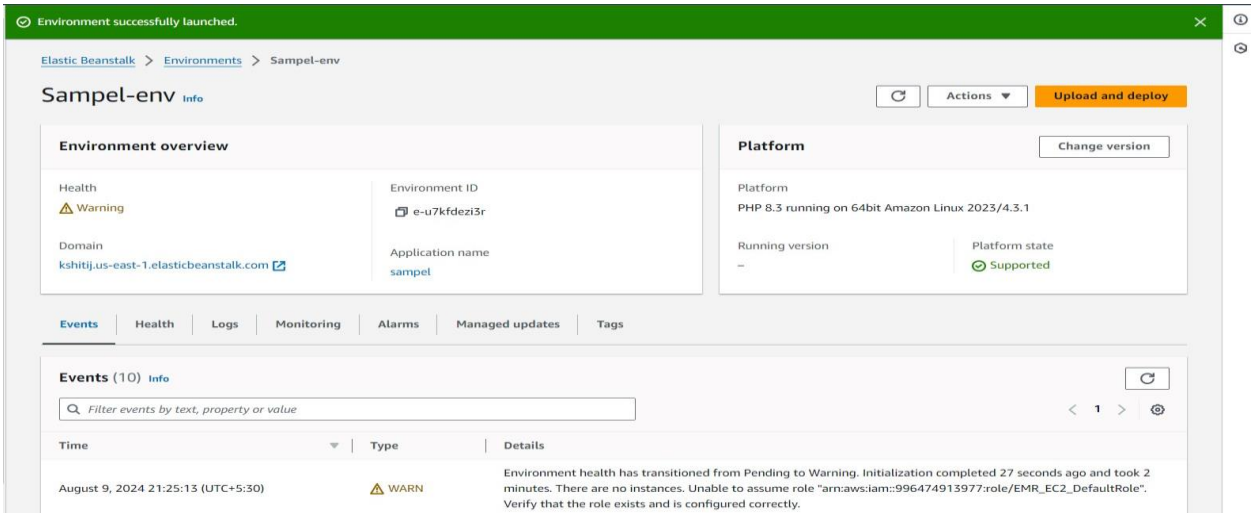
6) Configure updates, monitoring, and logging.Keep everything as default and click next.



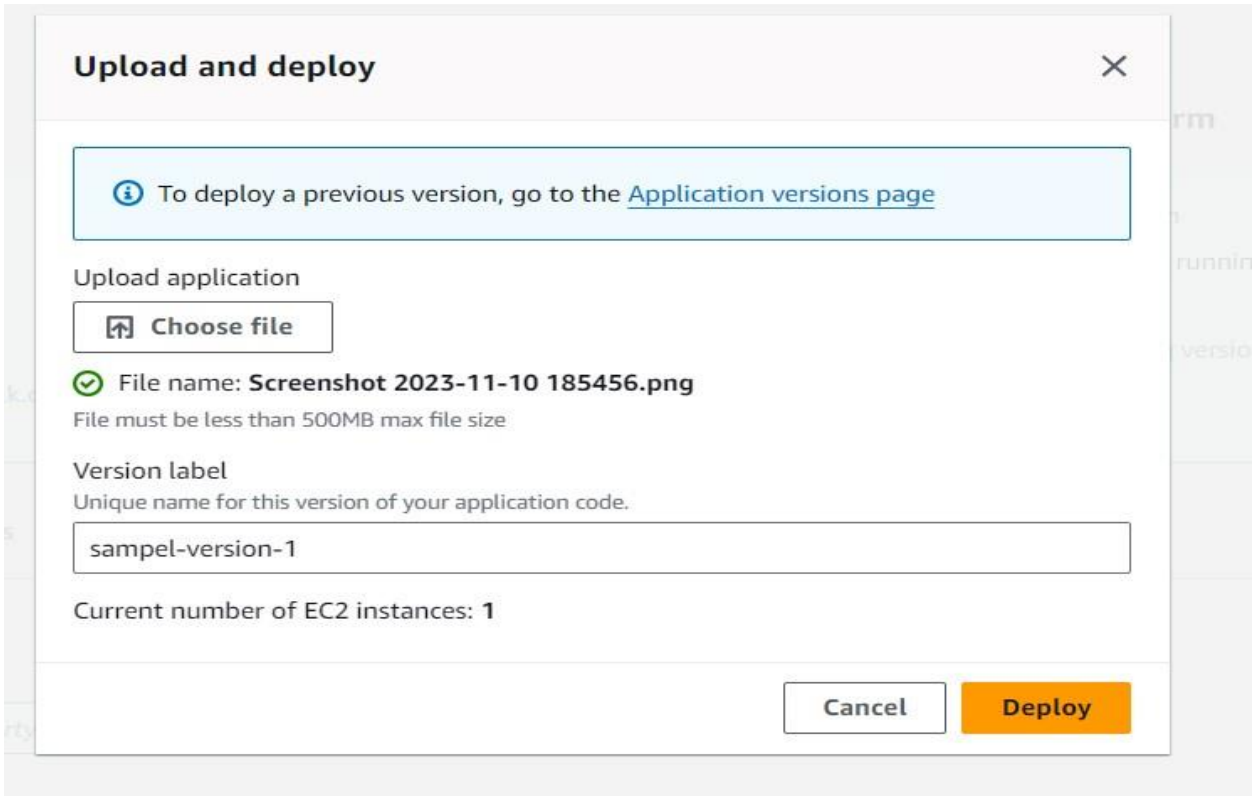
7) In the review section,click submit.



8) Environment has been created successfully.

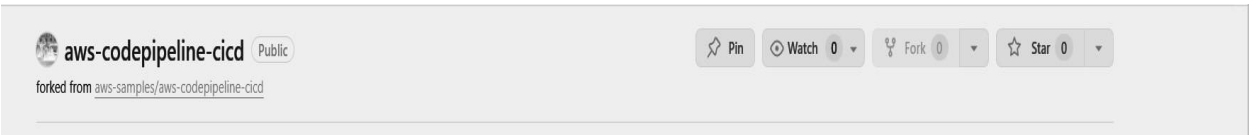


9) Deploy something on the recently created environment.

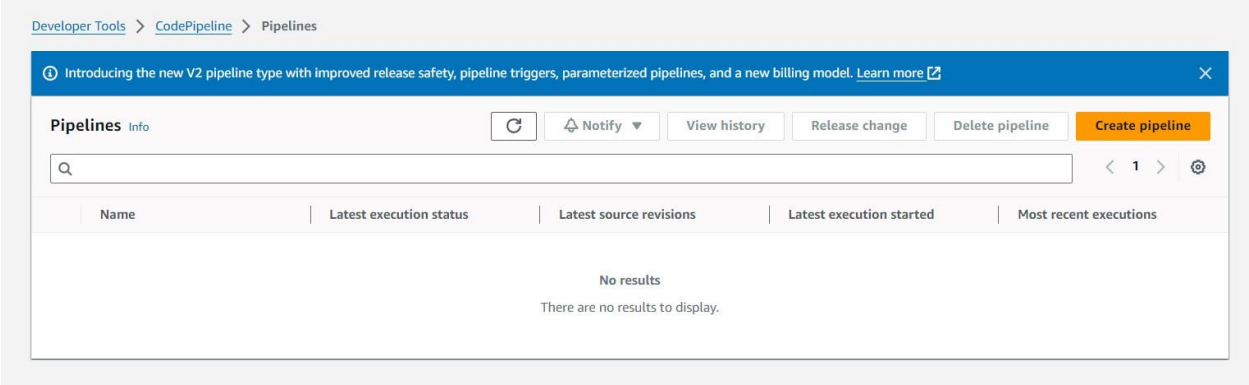

Upload and deploy
To deploy a previous version, go to the Application versions page
Upload application
Choose file
File name: Screenshot 2023-11-10 185456.png
File must be less than 500MB max file size
Version label
Unique name for this version of your application code.
sampel-version-1
Current number of EC2 instances: 1
Cancel
Deploy

Pipeline Creation:

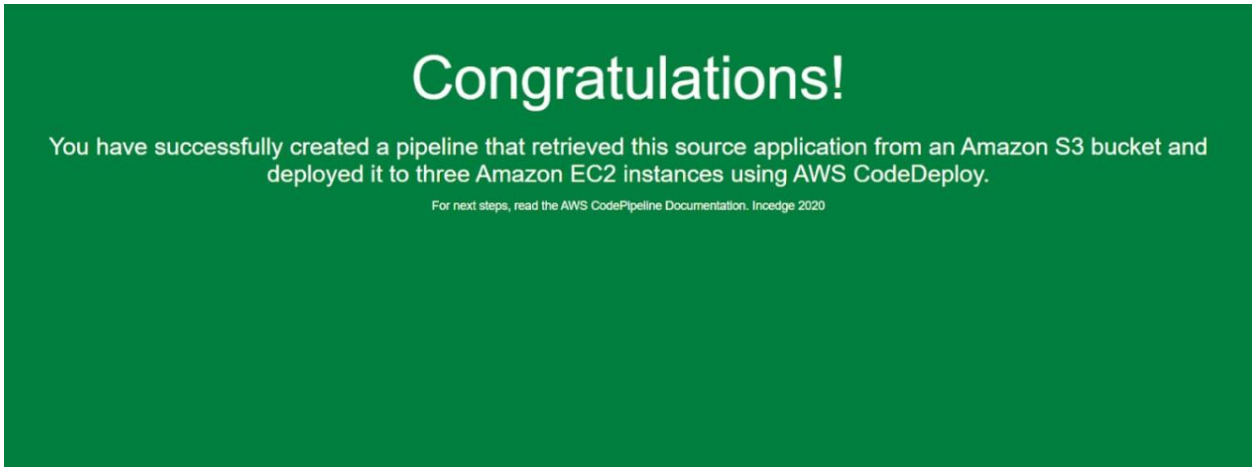
1) Fork a github repository. This forked repository will act as source for your code pipeline.



2) Go to developer tools and select CodePipeline and create a new pipeline



3) Create a pipeline:



Choose pipeline settings Info

Step 1 of 5

Pipeline settings

Pipeline name

Enter the pipeline name. You cannot edit the pipeline name after it is created.

firstpipeline

No more than 100 characters

Pipeline type

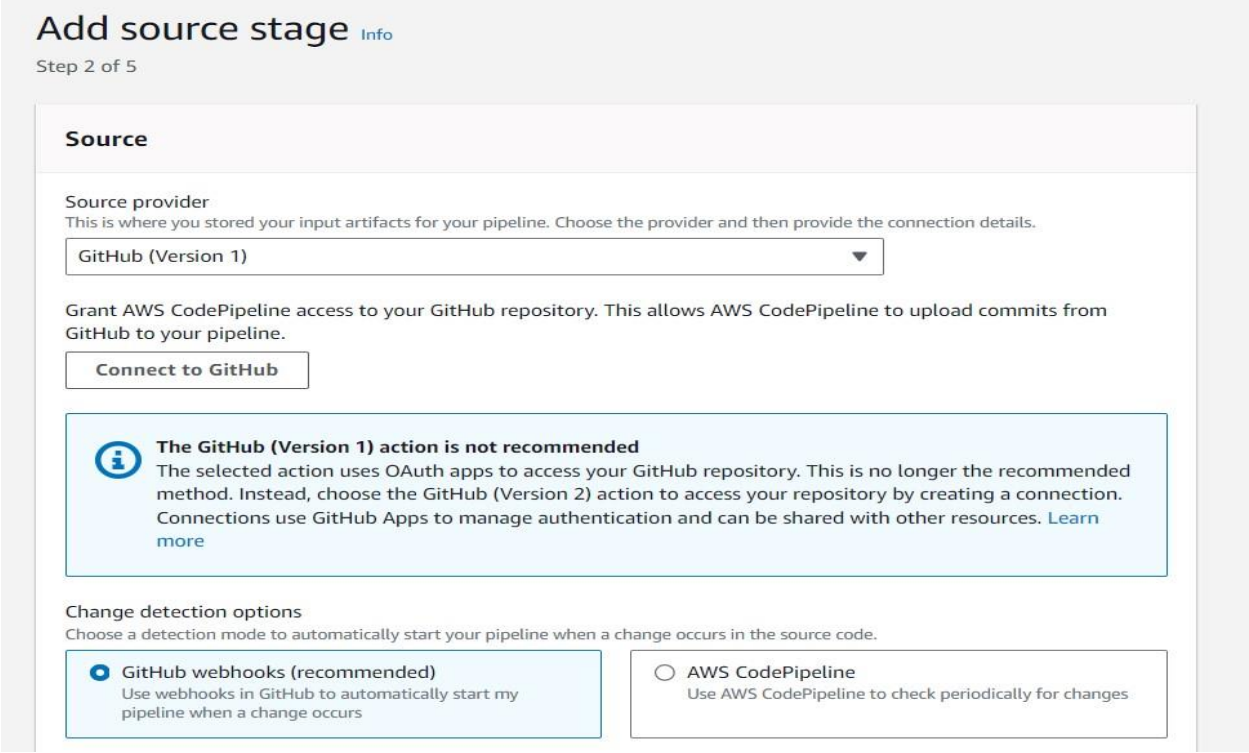
You can no longer create V1 pipelines through the console. We recommend you use the V2 pipeline type with improved release safety, pipeline triggers, parameterized pipelines, and a new billing model.

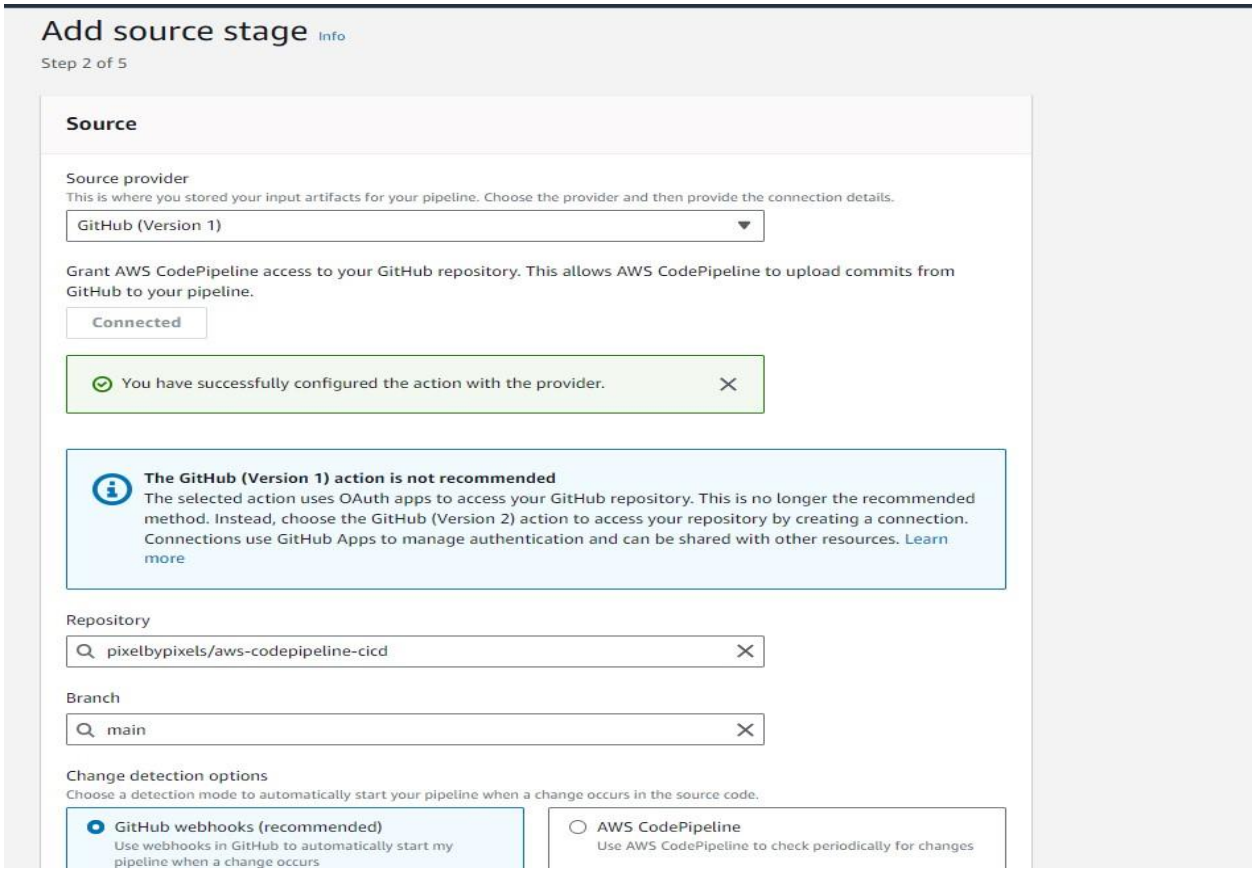
Execution mode

Choose the execution mode for your pipeline. This determines how the pipeline is run.

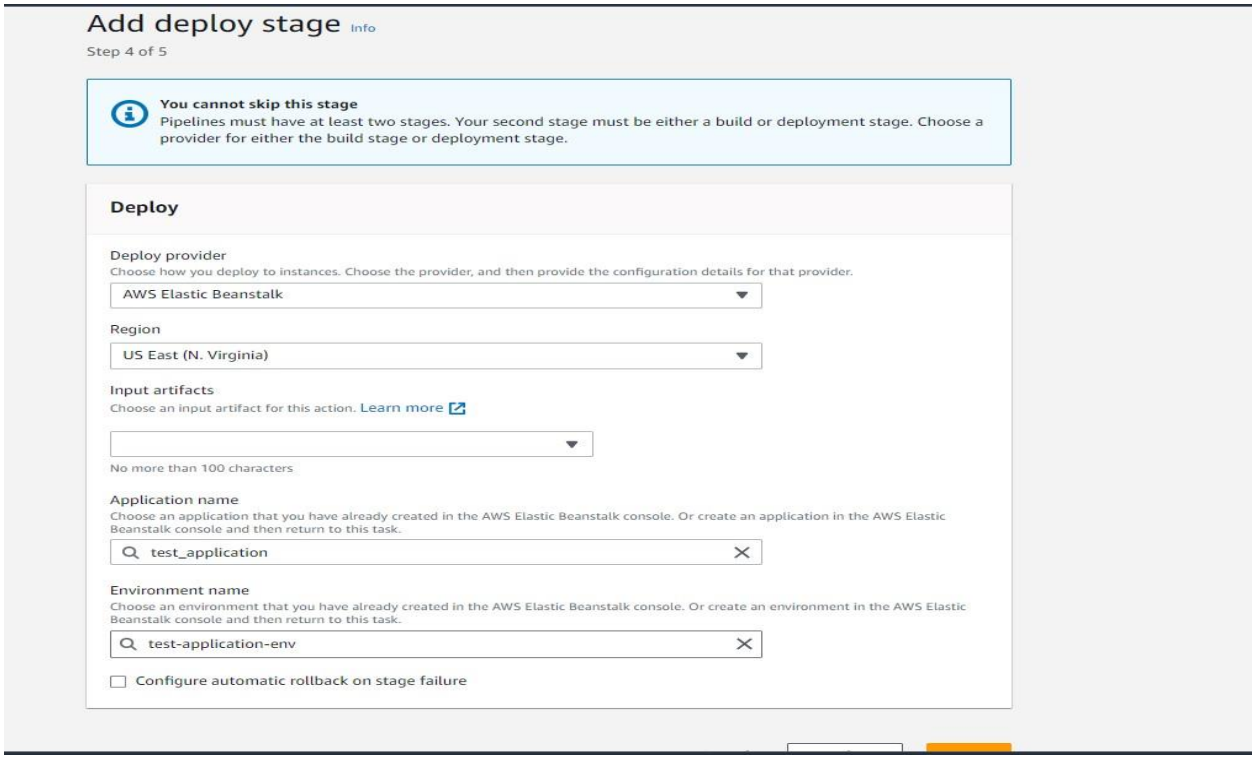
- ○ Superseded
  A more recent execution can overtake an older one. This is the default.
- ◉ Queued (Pipeline type V2 required)
  Executions are processed one by one in the order that they are queued.
- ○ Parallel (Pipeline type V2 required)

4)

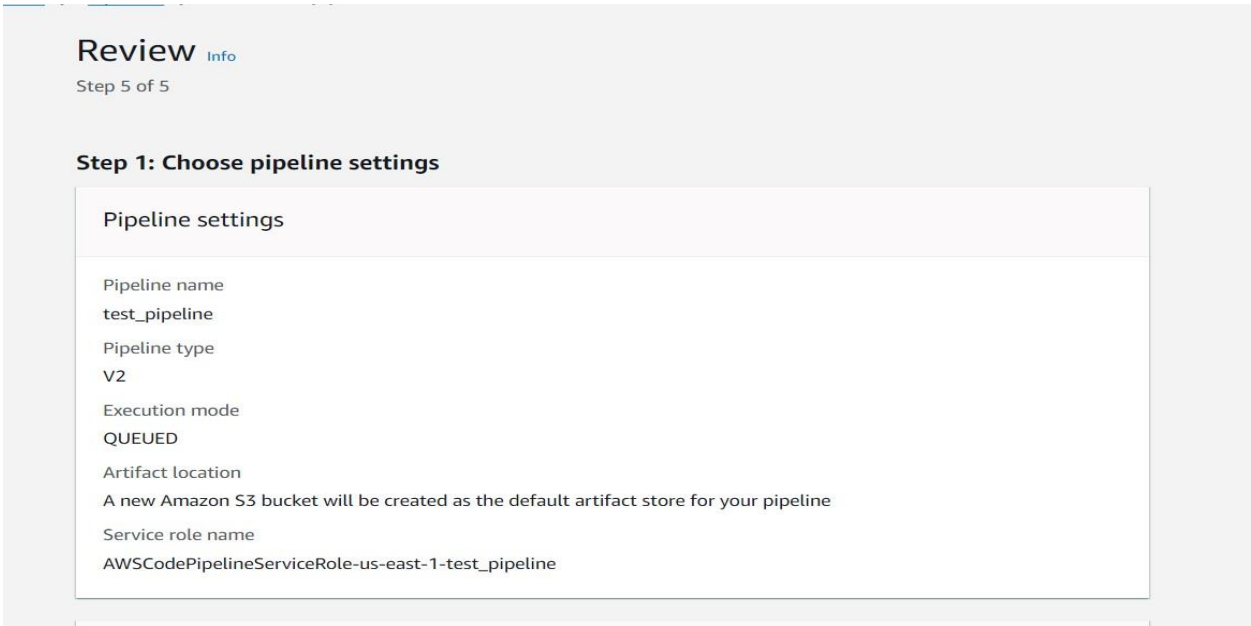

Add source stage Info
Step 2 of 5
Source
Source provider
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.
GitHub (Version 1)
Grant AWS CodePipeline access to your GitHub repository. This allows AWS CodePipeline to upload commits from GitHub to your pipeline.
Connect to GitHub
The GitHub (Version 1) action is not recommended
The selected action uses OAuth apps to access your GitHub repository. This is no longer the recommended method. Instead, choose the GitHub (Version 2) action to access your repository by creating a connection. Connections use GitHub Apps to manage authentication and can be shared with other resources. Learn more
Change detection options
Choose a detection mode to automatically start your pipeline when a change occurs in the source code.
GitHub webhooks (recommended)
Use webhooks in GitHub to automatically start my pipeline when a change occurs
AWS CodePipeline
Use AWS CodePipeline to check periodically for changes



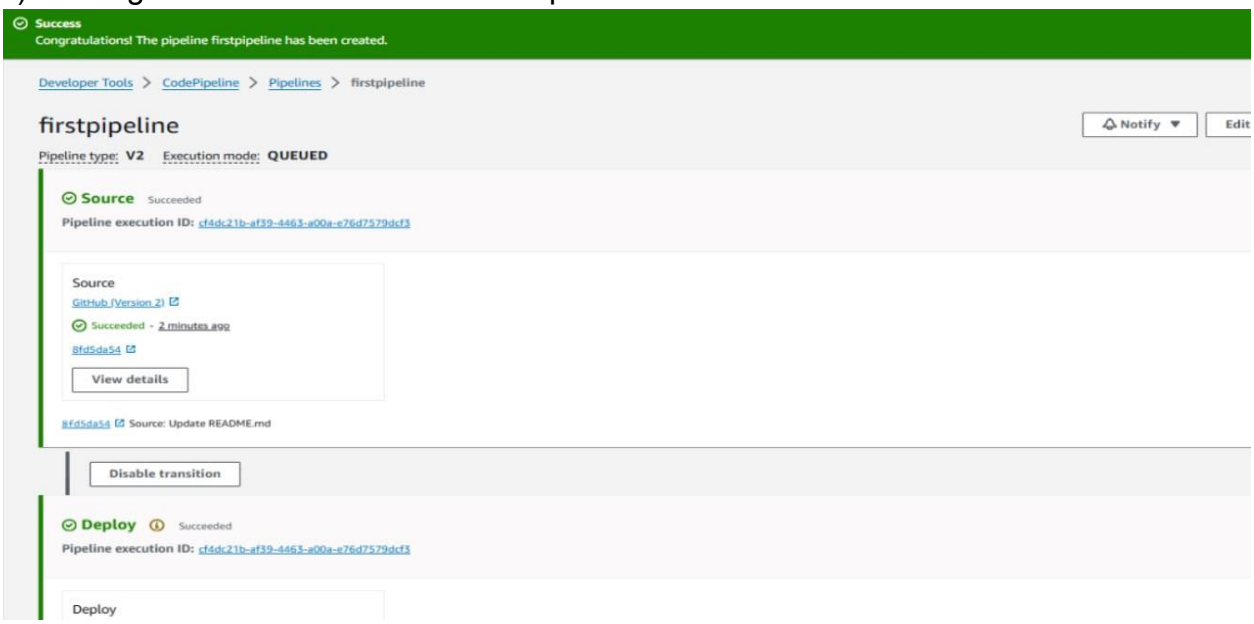
5) Go to the deploy stage and ensure the following settings.



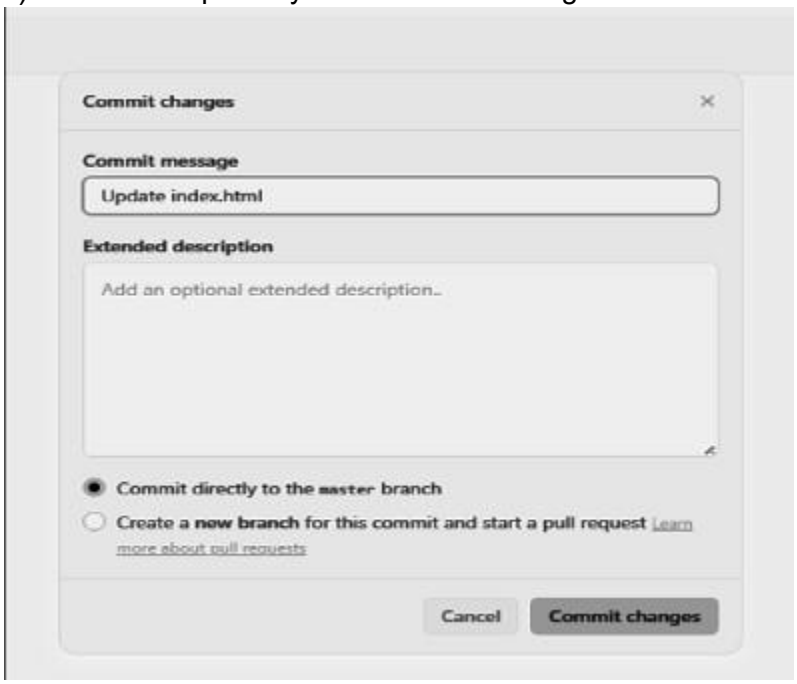
6) review the pipeline settings.



7) Then go ahead and check the URL provided in the EBS environment.



8) Go to the repository and make the changes in the index.html file and commit them

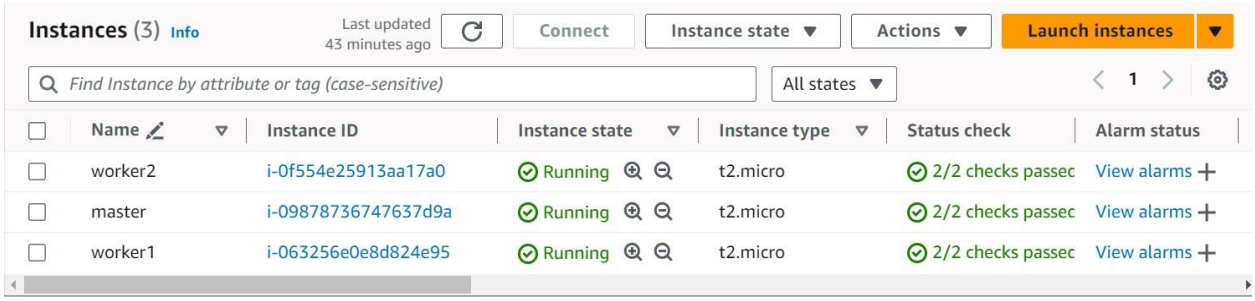


9)The changes that are committed can be noticed in the source panel in real time and to view the changes check the url (refresh it) and you can view the changes once the deployment section shows success.

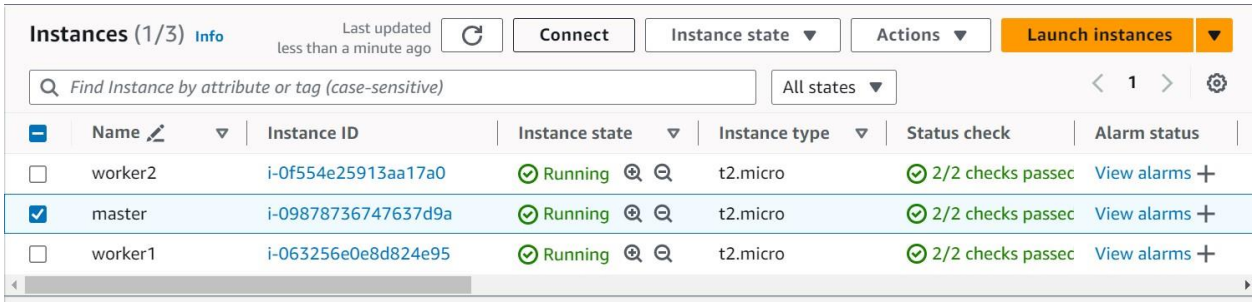
Aim: To understand the Kubernetes Cluster Architecture, install and Spin Up a Kubernetes Cluster on Linux Machines/Cloud Platforms.

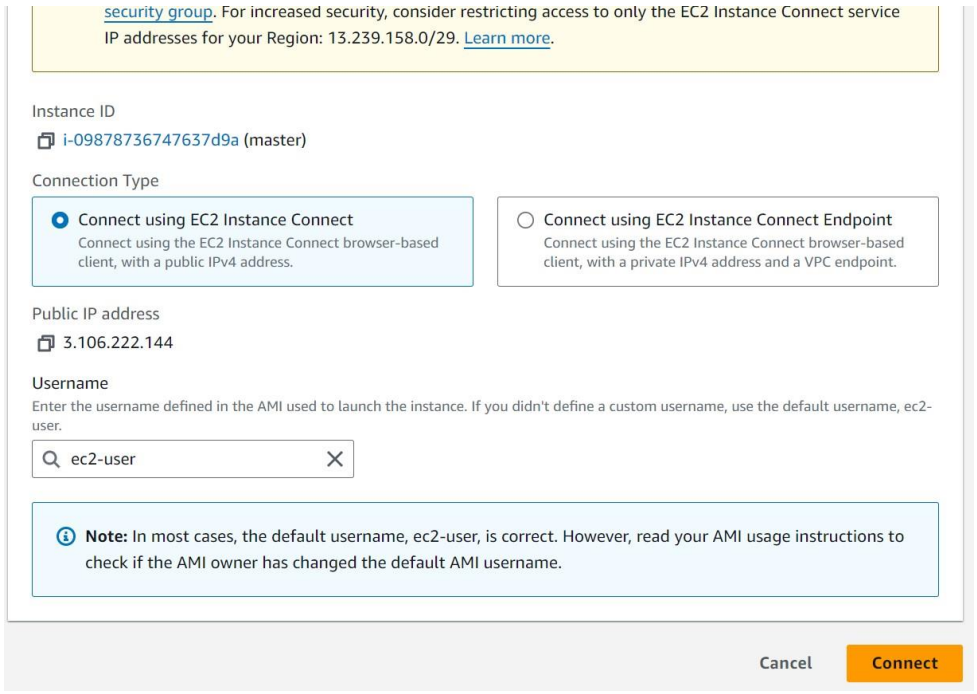
Steps:

1. We will create 3 EC2 instances. One will be the master node and the other 2 will be slave/worker nodes.



2. After the instances have been created, we will connect them one by one.





3. Docker installation:
   This step has to be performed on all the 3 instances. The following command has to be run:
   yum install docker -y

```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-12-97 ~]$ sudo su
[root@ip-172-31-12-97 ec2-user]# yum install docker -y
Last metadata expiration check: 0:08:33 ago on Sat Sep 14 15:21:32 2024.
Dependencies resolved.
================================================================================
 Package                 Architecture    Version                 Repository    Size
================================================================================
Installing:
 docker                  x86_64          25.0.6-1.amzn2023.0.2   amazonlinux   44 M
Installing dependencies:
 containerd              x86_64          1.7.20-1.amzn2023.0.1   amazonlinux   35 M
 iptables-libs           x86_64          1.8.8-3.amzn2023.0.2    amazonlinux  401 k
 iptables-nft            x86_64          1.8.8-3.amzn2023.0.2    amazonlinux  183 k
 libcgroup               x86_64          3.0-1.amzn2023.0.1      amazonlinux   75 k
 libnetfilter_conntrack  x86_64          1.0.8-2.amzn2023.0.2    amazonlinux   58 k
```

```
aws  Services  Search [Alt+S]                                  Sydney ▾  bhumish
 libn                    k  x86_64  1.0.8-2.amzn2023.0.2   amazonlinux   58 k
 libnfnetlink            x86_64  1.0.1-19.amzn2023.0.2   amazonlinux   30 k
 libnftnl                x86_64  1.2.2-2.amzn2023.0.2    amazonlinux   84 k
 pigz                    x86_64  2.5-1.amzn2023.0.3      amazonlinux   83 k
 runc                    x86_64  1.1.13-1.amzn2023.0.1   amazonlinux  3.2 M

Transaction Summary
================================================================================
Install  10 Packages

Total download size: 84 M
Installed size: 317 M
Downloading Packages:
(1/10): iptables-libs-1.8.8-3.amzn2023.0.2.x86_64.rpm      3.0 MB/s | 401 kB  00:00
(2/10): iptables-nft-1.8.8-3.amzn2023.0.2.x86_64.rpm       6.6 MB/s | 183 kB  00:00
(3/10): libcgroup-3.0-1.amzn2023.0.1.x86_64.rpm            1.7 MB/s |  75 kB  00:00
(4/10): libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64.rpm 1.6 MB/s |  58 kB  00:00
(5/10): libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64.rpm      823 kB/s |  30 kB  00:00
(6/10): libnftnl-1.2.2-2.amzn2023.0.2.x86_64.rpm           2.9 MB/s |  84 kB  00:00
(7/10): pigz-2.5-1.amzn2023.0.3.x86_64.rpm                 2.4 MB/s |  83 kB  00:00
(8/10): runc-1.1.13-1.amzn2023.0.1.x86_64.rpm               15 MB/s | 3.2 MB  00:00
(9/10): containerd-1.7.20-1.amzn2023.0.1.x86_64.rpm         36 MB/s |  35 MB  00:00
(10/10): docker-25.0.6-1.amzn2023.0.2.x86_64.rpm            30 MB/s |  44 MB  00:01
--------------------------------------------------------------------------------
Total                                                       56 MB/s |  84 MB  00:01
```

```
  Run                     ptables-nft-1.8.8-3.amzn2023.0.2.x86_64                 8/10
  Installing       : libcgroup-3.0-1.amzn2023.0.1.x86_64                          9/10
  Running scriptlet: docker-25.0.6-1.amzn2023.0.2.x86_64                         10/10
  Installing       : docker-25.0.6-1.amzn2023.0.2.x86_64                         10/10
  Running scriptlet: docker-25.0.6-1.amzn2023.0.2.x86_64                         10/10
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.

  Verifying        : containerd-1.7.20-1.amzn2023.0.1.x86_64                      1/10
  Verifying        : docker-25.0.6-1.amzn2023.0.2.x86_64                          2/10
  Verifying        : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64                    3/10
  Verifying        : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64                     4/10
  Verifying        : libcgroup-3.0-1.amzn2023.0.1.x86_64                          5/10
  Verifying        : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64           6/10
  Verifying        : libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64                    7/10
  Verifying        : libnftnl-1.2.2-2.amzn2023.0.2.x86_64                         8/10
  Verifying        : pigz-2.5-1.amzn2023.0.3.x86_64                               9/10
  Verifying        : runc-1.1.13-1.amzn2023.0.1.x86_64                           10/10

Installed:
  containerd-1.7.20-1.amzn2023.0.1.x86_64    docker-25.0.6-1.amzn2023.0.2.x86_64   iptables-libs-1.8.8-3.amzn2023.0.2.x86_64
  iptables-nft-1.8.8-3.amzn2023.0.2.x86_64   libcgroup-3.0-1.amzn2023.0.1.x86_64   libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
  libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64  libnftnl-1.2.2-2.amzn2023.0.2.x86_64  pigz-2.5-1.amzn2023.0.3.x86_64
  runc-1.1.13-1.amzn2023.0.1.x86_64

Complete!
```

4. After successfully docker has been installed it has to be started on all machines by using the command “systemctl start docker”

```
Complete!
[root@ip-172-31-12-97 ec2-user]# systemctl start docker
```

5. Kubernetes installation:

Search kubeadm installation on your browser and scroll down and select red hat-based distributions.

1. Set SELinux to `permissive` mode:

These instructions are for Kubernetes 1.31.

```
Linux in permissive mode (effectively disabling it)
enforce 0
 -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
```

```
# This overwrites any existing configuration in /etc/yum.repos.d/
cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/repodata/repom
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
EOF
```

3. Install kubelet, kubeadm and kubectl:

```
yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
```

4. (Optional) Enable the kubelet service before running kubeadm:

```
sudo systemctl enable --now kubelet
```

Copy the above given steps and paste in the terminal. This will create a Kubernetes repository, install kubelet, kubeadm and kubectl and also enable the services.

```
[root                ec2-user]# cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
EOF
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
[root@ip-172-31-12-97 ec2-user]# yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Kubernetes                                                                                    21 kB/s | 9.4 kB     00:00
Dependencies resolved.
================================================================================================================================
 Package                          Architecture          Version                          Repository                    Size
================================================================================================================================
Installing:
 kubeadm                          x86_64                1.31.1-150500.1.1                kubernetes                    11 M
 kubectl                          x86_64                1.31.1-150500.1.1                kubernetes                    11 M
 kube                             x86_64                1.31.1-150500.1.1                kubernetes                    15 M
Installing dependencies:
 conntrack-tools                  x86_64                1.4.6-2.amzn2023.0.2             amazonlinux                  208 k
 cri-tools                        x86_64                1.31.1-150500.1.1                kubernetes                   6.9 M
 kubernetes-cni                   x86_64                1.5.1-150500.1.1                 kubernetes                   7.1 M
 libnetfilter_cthelper            x86_64                1.0.0-21.amzn2023.0.2            amazonlinux                   24 k
 libnetfilter_cttimeout           x86_64                1.0.0-19.amzn2023.0.2            amazonlinux                   24 k
 libnetfilter_queue               x86_64                1.0.5-2.amzn2023.0.2             amazonlinux                   30 k

Transaction Summary
================================================================================================================================
Install  9 Packages

Total download size: 51 M
Installed size: 269 M
Downloading Packages:
(1/9): libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64.rpm                                 500 kB/s |  24 kB     00:00
(2/9): libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64.rpm                                475 kB/s |  24 kB     00:00
(3/9): conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64.rpm                                        3.6 MB/s | 208 kB     00:00
(4/9): libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64.rpm                                     1.4 MB/s |  30 kB     00:00
(5/9): kubeadm-1.31.1-150500.1.1.x86_64.rpm                                                    17 MB/s |  11 MB     00:00
(6/9): kubectl-1.31.1-150500.1.1.x86_64.rpm                                                    15 MB/s |  11 MB     00:00
(7/9): cri-tools-1.31.1-150500.1.1.x86_64.rpm                                                 8.0 MB/s | 6.9 MB     00:00
(8/9): kubernetes-cni-1.5.1-150500.1.1.x86_64.rpm                                              14 MB/s | 7.1 MB     00:00
(9/9): kubelet-1.31.1-150500.1.1.x86_64.rpm                                                    25 MB/s |  15 MB     00:00
  Ins                ibnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64                                                  5/9
  Installing       : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64                                                        6/9
  Running scriptlet: conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64                                                        6/9
  Installing       : kubelet-1.31.1-150500.1.1.x86_64                                                                   7/9
  Running scriptlet: kubelet-1.31.1-150500.1.1.x86_64                                                                   7/9
  Installing       : kubeadm-1.31.1-150500.1.1.x86_64                                                                   8/9
  Installing       : kubectl-1.31.1-150500.1.1.x86_64                                                                   9/9
  Running scriptlet: kubectl-1.31.1-150500.1.1.x86_64                                                                   9/9
  Verifying        : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64                                                        1/9
  Verifying        : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64                                                 2/9
  Verifying        : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64                                                3/9
  Verifying        : libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64                                                     4/9
  Verifying        : cri-tools-1.31.1-150500.1.1.x86_64                                                                 5/9
  Verifying        : kubeadm-1.31.1-150500.1.1.x86_64                                                                   6/9
  Verifying        : kubectl-1.31.1-150500.1.1.x86_64                                                                   7/9
  Verifying        : kubelet-1.31.1-150500.1.1.x86_64                                                                   8/9
  Verifying        : kubernetes-cni-1.5.1-150500.1.1.x86_64                                                             9/9

Installed:
  conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64                cri-tools-1.31.1-150500.1.1.x86_64
  kubeadm-1.31.1-150500.1.1.x86_64                           kubectl-1.31.1-150500.1.1.x86_64
  kubelet-1.31.1-150500.1.1.x86_64                           kubernetes-cni-1.5.1-150500.1.1.x86_64
  libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64         libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64
  libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64
```

6. We can check if repository has been created by using yum repolist command.

```
[root@ip-172-31-14-85 ec2-user]# yum repolist
repo id                                                repo name
amazonlinux                                            Amazon Linux 2023 repository
kernel-livepatch                                       Amazon Linux 2023 Kernel Livepatch repository
kubernetes                                             Kubernetes
[root@ip-172-31-14-85 ec2-user]#
```

7. Now we will be initializing the kubeadm. For that “kubeadm init” command has to be used. It may show errors but those can be ignored by using --ignore-preflighterrors=all

```
[root@ip-172-31-14-85 ec2-user]# kubeadm init --ignore-preflight-errors=NumCPU --ignore-preflight-errors=Mem
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
        [WARNING NumCPU]: the number of available CPUs 1 is less than the required 2
        [WARNING Mem]: the system RAM (949 MB) is less than the minimum 1700 MB
        [WARNING FileExisting-socat]: socat not found in system path
        [WARNING FileExisting-tc]: tc not found in system path
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
W0914 15:50:31.271160   29520 checks.go:846] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inc
onsistent with that used by kubeadm.It is recommended to use "registry.k8s.io/pause:3.10" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-14-85.ap-southeast-2.compute.internal kubernetes kubernetes.default ku
bernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 172.31.14.85]
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key
[certs] Generating "etcd/ca" certificate and key
[certs] Generating "etcd/server" certificate and key
[certs] etcd/server serving cert is signed for DNS names [ip-172-31-14-85.ap-southeast-2.compute.internal localhost] and IPs [172.31.14.
85 127.0.0.1 ::1]
```

```
aws  Services  Search [Alt+S]  Sydney ▾  bhumisha
85 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-14-85.ap-southeast-2.compute.internal localhost] and IPs [172.31.14.85
 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
[certs] Generating "sa" key and public key
[kubeconfig] Using kubeconfig folder "/etc/kubernetes"
[kubeconfig] Writing "admin.conf" kubeconfig file
[kubeconfig] Writing "super-admin.conf" kubeconfig file
[kubeconfig] Writing "kubelet.conf" kubeconfig file
[kubeconfig] Writing "controller-manager.conf" kubeconfig file
[kubeconfig] Writing "scheduler.conf" kubeconfig file
[etcd] Creating static Pod manifest for local etcd in "/etc/kubernetes/manifests"
[control-plane] Using manifest folder "/etc/kubernetes/manifests"
[control-plane] Creating static Pod manifest for "kube-apiserver"
[control-plane] Creating static Pod manifest for "kube-controller-manager"
[control-plane] Creating static Pod manifest for "kube-scheduler"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Starting the kubelet
[wait-control-plane] Waiting for the kubelet to boot up the control plane as static Pods from directory "/etc/kubernetes/manifests"
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 518.648244ms
[api-check] Waiting for a healthy API server. This can take up to 4m0s
```

```
aws  Services  Search [Alt+S]  Sydney ▾  bhumishap
[wait-control-plane] Waiting for the kubelet to boot up the control plane as static Pods from directory "/etc/kubernetes/manifests"
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 518.648244ms
[api-check] Waiting for a healthy API server. This can take up to 4m0s
[api-check] The API server is healthy after 10.001658622s
[upload-config] Storing the configuration used in ConfigMap "kubeadm-config" in the "kube-system" Namespace
[kubelet] Creating a ConfigMap "kubelet-config" in namespace kube-system with the configuration for the kubelets in the cluster
[upload-certs] Skipping phase. Please see --upload-certs
[mark-control-plane] Marking the node ip-172-31-14-85.ap-southeast-2.compute.internal as control-plane by adding the labels: [node-role.
kubernetes.io/control-plane node.kubernetes.io/exclude-from-external-load-balancers]
[mark-control-plane] Marking the node ip-172-31-14-85.ap-southeast-2.compute.internal as control-plane by adding the taints [node-role.k
ubernetes.io/control-plane:NoSchedule]
[bootstrap-token] Using token: 61ysht.48enn4gmnhof6ex8
[bootstrap-token] Configuring bootstrap tokens, cluster-info ConfigMap, RBAC Roles
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to get nodes
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to post CSRs in order for nodes to get long term certificate cred
entials
[bootstrap-token] Configured RBAC rules to allow the csrapprover controller automatically approve CSRs from a Node Bootstrap Token
[bootstrap-token] Configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
[bootstrap-token] Creating the "cluster-info" ConfigMap in the "kube-public" namespace
[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable kubelet client certificate and key
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy

Your Kubernetes control-plane has initialized successfully!
```

8. On successful initialization we need to copy and paste the following commands on the master machine itself:

```
To start using your cluster, you need to run the following as a regular user:

  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config

Alternatively, if you are the root user, you can run:

  export KUBECONFIG=/etc/kubernetes/admin.conf
```

9. Next copy and paste the join link in the worker nodes so that the worker nodes can join the cluster.

```
Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 172.31.14.85:6443 --token 61ysht.48enn4gmnhof6ex8 \
        --discovery-token-ca-cert-hash sha256:461819c971fe032e04a78e18fde8e28755825e8468d468a2c86d88c52dba4945
```

10. After performing join commands on the worker nodes, we will get following output:

```
This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.
```

11. Once again when you run kubectl get nodes you will now see all 3 nodes have joined the cluster.

```
NAME                                STATUS     ROLES           AGE    VERSION
ip-172-31-85-89.ec2.internal    NotReady   control-plane   119s   v1.26.0
ip-172-31-89-46.ec2.internal    NotReady   <none>          19s    v1.26.0
ip-172-31-94-70.ec2.internal    NotReady   <none>          12s    v1.26.0
```

Conclusion:

This experiment successfully demonstrated the creation of a Kubernetes cluster and the successful addition of all three nodes using various commands. Errors encountered during initialization can be addressed in two ways: 1) by ignoring the errors, or 2) by upgrading the instance type to t3.medium or t3.large if the issues are due to insufficient memory or CPU resources.

Name:Muskan chandiramani. **D15C.** **Roll No: 5**

Aim: To install Kubectl and execute Kubectl commands to manage the Kubernetes cluster and deploy Your First Kubernetes Application.

Theory:

Kubernetes, originally developed by Google, is an open-source container orchestration platform. It automates the deployment, scaling, and management of containerized applications, ensuring high availability and fault tolerance. Kubernetes is now the industry standard for container orchestration and is governed by the **Cloud Native Computing Foundation (CNCF)**, with contributions from major cloud and software providers like Google, AWS, Microsoft, IBM, Intel, Cisco, and Red Hat.

Kubernetes Deployment: Is a resource in Kubernetes that provides declarative updates for Pods and ReplicaSets. With a Deployment, you can define how many replicas of a pod should run, roll out new versions of an application, and roll back to previous versions if necessary. It ensures that the desired number of pod replicas are running at all times.

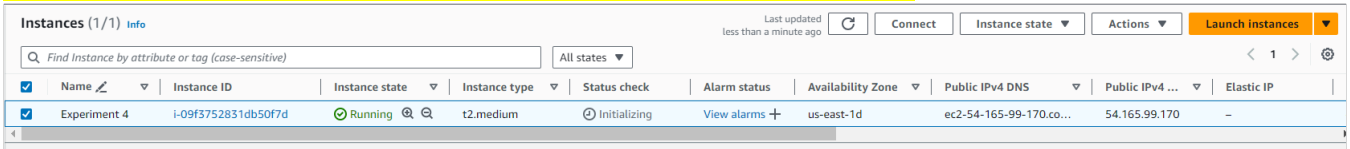
Necessary Requirements:

- **EC2 Instance:** The experiment required launching a t2.medium EC2 instance with 2 CPUs, as Kubernetes demands sufficient resources for effective functioning.
- **Minimum Requirements:**
    - **Instance Type:** t2.medium ○ **CPUs:** 2
    - **Memory:** Adequate for container orchestration.

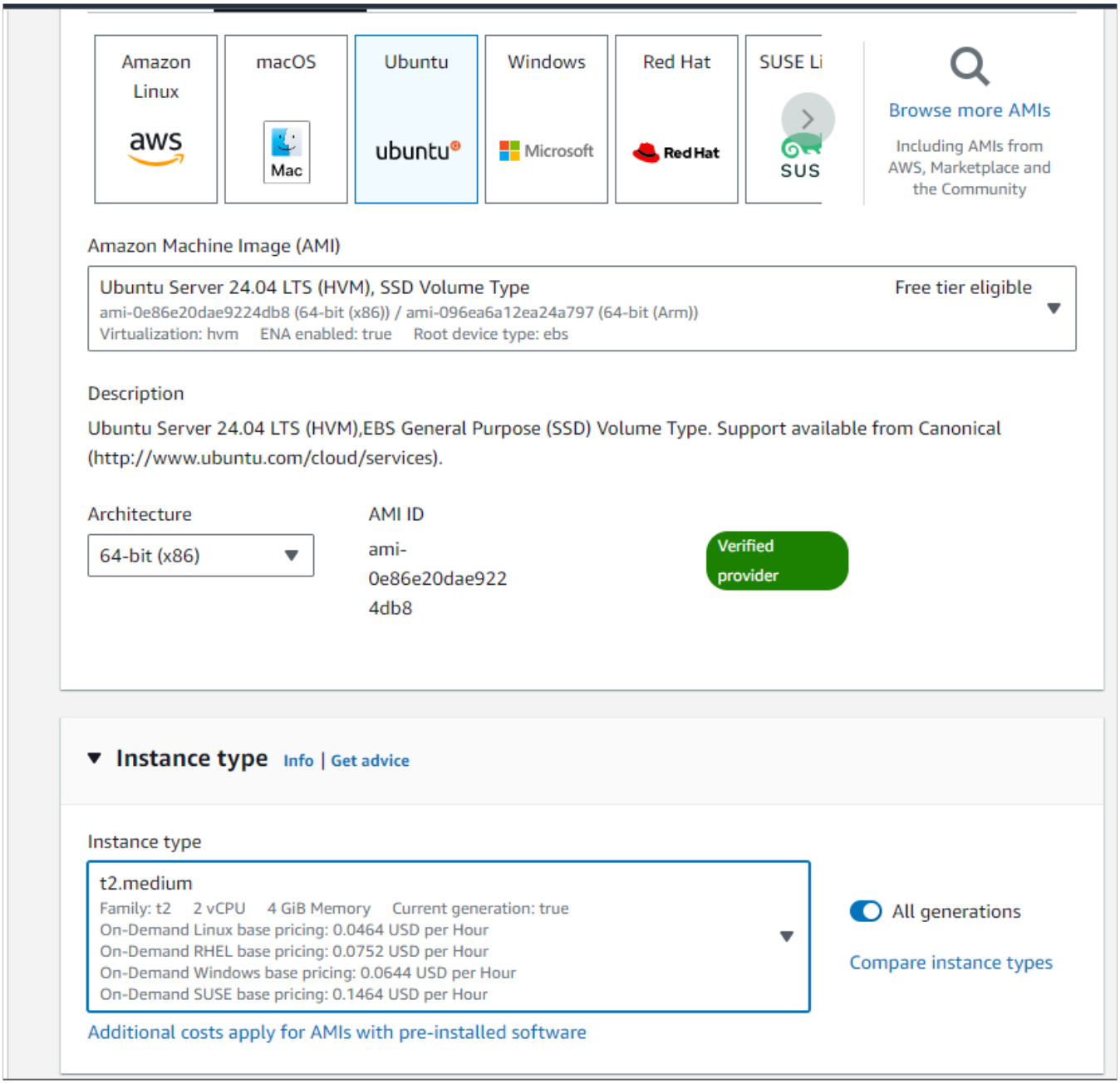
This ensured that the Kubernetes cluster had the necessary resources to function smoothly.

Note:
AWS Personal Account is preferred but we can also perform it on AWS Academy(adding some ignores in the command if any error occurs in below as the below experiment is performed on Personal Account .).
If You are using AWS Academy Account Errors you will face in kubeadm init command so you have to add some ignores with this command.

Step 1: Log in to your AWS Academy/personal account and launch a new Ec2 Instance.
Select Ubuntu as AMI and t2.medium as Instance Type, create a key of type RSA with .pem extension, and move the downloaded key to the new folder.
Note: A minimum of 2 CPUs are required so Please select t2.medium and do not forget to stop the instance after the experiment because it is not available in the free tier.

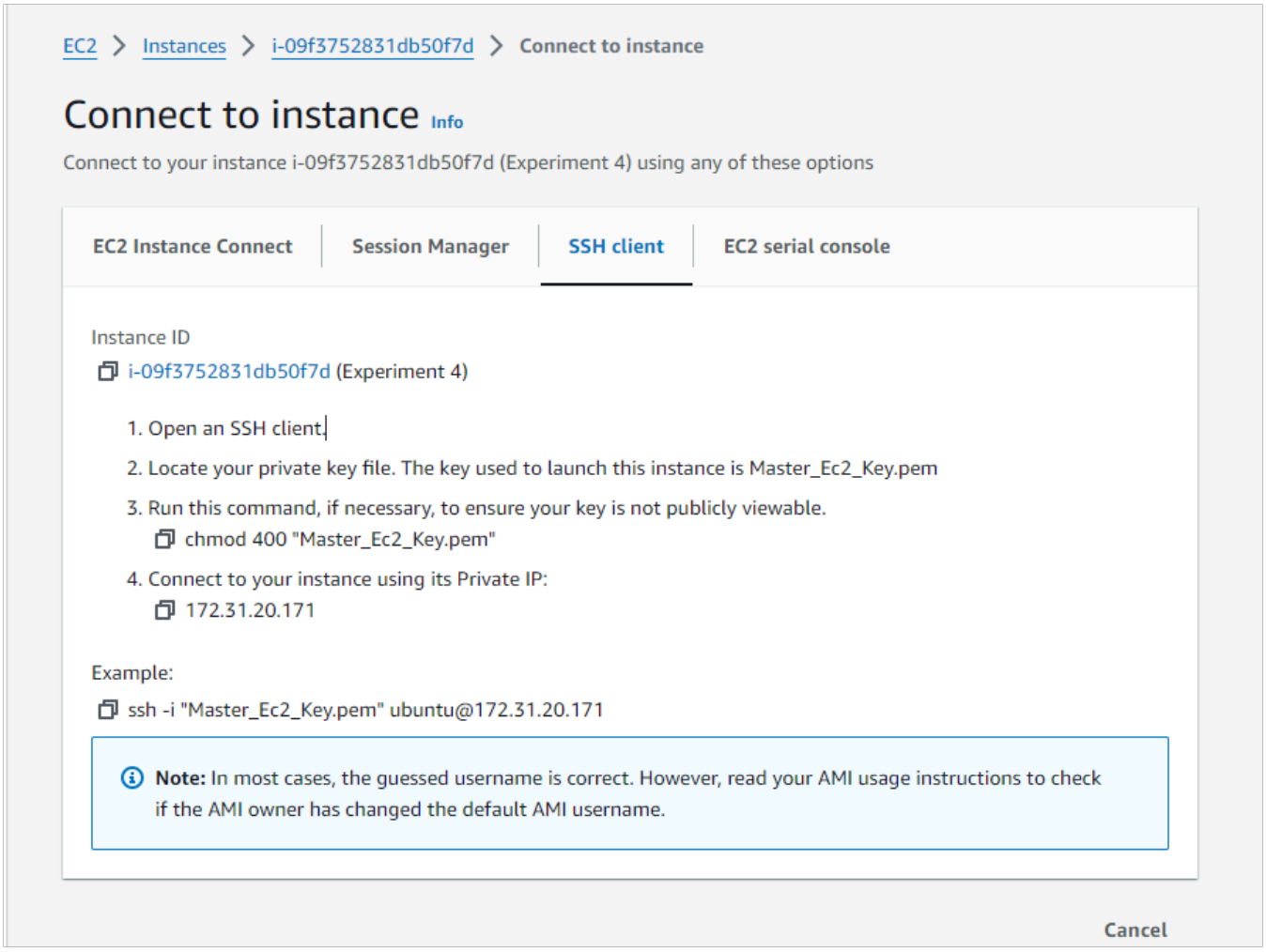


Name:Muskan chandiramani. **D15C.** **Roll No: 5**



Step 2: After creating the instance click on Connect the instance and navigate to SSH Client.

Name:Muskan chandiramani. **D15C.** **Roll No: 5**



Step 3: Now open the folder in the terminal where our .pem key is stored and paste the Example command (starting with ssh -i …..) in the terminal.(ssh -i "Master_Ec2_Key.pem" ubuntu@ec2-54-196-129-215.compute-1.amazonaws.com)

Name:Muskan chandiramani. D15C. Roll No: 5

Step 4: Run the below commands to install and setup Docker.

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee /etc/apt/trusted.gpg.d/docker.gpg > /dev/null

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"

```
ubuntu@ip-172-31-20-171:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee /etc/apt/trusted.gpg.d/docker.gpg > /dev/null
ubuntu@ip-172-31-20-171:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
Repository: 'deb [arch=amd64] https://download.docker.com/linux/ubuntu noble stable'
Description:
Archive for codename: noble components: stable
More info: https://download.docker.com/linux/ubuntu
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.
Found existing deb entry in /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-noble.list
Adding deb entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-noble.list
Found existing deb-src entry in /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-noble.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-noble.list
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Get:4 https://download.docker.com/linux/ubuntu noble InRelease [48.8 kB]
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:6 https://download.docker.com/linux/ubuntu noble/stable amd64 Packages [13.8 kB]
Fetched 62.6 kB in 0s (128 kB/s)
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: The key(s) in the keyring /etc/apt/trusted.gpg.d/docker.gpg are ignored as the file has a
n unsupported filetype.
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION s
ection in apt-key(8) for details.
```

Name:Muskan chandiramani. D15C. Roll No: 5

sudo apt-get update sudo apt-get install -y docker-ce

```
ubuntu@ip-172-31-20-171:~$ sudo apt-get update
sudo apt-get install -y docker-ce
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://download.docker.com/linux/ubuntu noble InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: The key(s) in the keyring /etc/apt/trusted.gpg.d/docker.gpg are ignored as the file has a
n unsupported filetype.
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION s
ection in apt-key(8) for details.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  containerd.io docker-buildx-plugin docker-ce-cli
  docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz
  slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli
  docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz
  slirp4netns
0 upgraded, 10 newly installed, 0 to remove and 133 not upgraded.
Need to get 122 MB of archives.
After this operation, 440 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libltdl7 amd64 2.4.7-7build1 [40.3 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libslirp0 amd64 4.7.0-1ubuntu3 [63.8 kB]
Get:4 https://download.docker.com/linux/ubuntu noble/stable amd64 containerd.io amd64 1.7.22-1 [29.5 MB]
```

Name:Muskan chandiramani. D15C. Roll No: 5

```
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 slirp4netns amd64 1.2.1-1build2 [34.9 kB]
Get:6 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-buildx-plugin amd64 0.16.2-1~ubuntu.24.04~noble [29.9 MB]
Get:7 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce-cli amd64 5:27.2.1-1~ubuntu.24.04~noble [15.0 MB]
Get:8 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce amd64 5:27.2.1-1~ubuntu.24.04~noble [25.6 MB]
Get:9 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce-rootless-extras amd64 5:27.2.1-1~ubuntu.24.04~noble [9571 kB]
Get:10 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-compose-plugin amd64 2.29.2-1~ubuntu.24.04~noble [12.5 MB]
Fetched 122 MB in 2s (71.3 MB/s)
Selecting previously unselected package pigz.
(Reading database ... 67741 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.8-1_amd64.deb ...
Unpacking pigz (2.8-1) ...
Selecting previously unselected package containerd.io.
Preparing to unpack .../1-containerd.io_1.7.22-1_amd64.deb ...
Unpacking containerd.io (1.7.22-1) ...
Selecting previously unselected package docker-buildx-plugin.
Preparing to unpack .../2-docker-buildx-plugin_0.16.2-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-buildx-plugin (0.16.2-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce-cli.
Preparing to unpack .../3-docker-ce-cli_5%3a27.2.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce-cli (5:27.2.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce.
Preparing to unpack .../4-docker-ce_5%3a27.2.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce (5:27.2.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce-rootless-extras.
Preparing to unpack .../5-docker-ce-rootless-extras_5%3a27.2.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce-rootless-extras (5:27.2.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-compose-plugin.
Preparing to unpack .../6-docker-compose-plugin_2.29.2-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-compose-plugin (2.29.2-1~ubuntu.24.04~noble) ...
Selecting previously unselected package libltdl7:amd64.
Preparing to unpack .../7-libltdl7_2.4.7-7build1_amd64.deb ...
Unpacking libltdl7:amd64 (2.4.7-7build1) ...
Selecting previously unselected package libslirp0:amd64.
Preparing to unpack .../8-libslirp0_4.7.0-1ubuntu3_amd64.deb ...
Unpacking libslirp0:amd64 (4.7.0-1ubuntu3) ...
Selecting previously unselected package slirp4netns.
Preparing to unpack .../9-slirp4netns_1.2.1-1build2_amd64.deb ...
Unpacking slirp4netns (1.2.1-1build2) ...
Setting up docker-buildx-plugin (0.16.2-1~ubuntu.24.04~noble) ...
Setting up containerd.io (1.7.22-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /usr/lib/systemd/system/containerd.service.
```

```
Unpacking slirp4netns (1.2.1-1build2) ...
Setting up docker-buildx-plugin (0.16.2-1~ubuntu.24.04~noble) ...
Setting up containerd.io (1.7.22-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /usr/lib/systemd/system/containerd.service.
Setting up docker-compose-plugin (2.29.2-1~ubuntu.24.04~noble) ...
Setting up libltdl7:amd64 (2.4.7-7build1) ...
Setting up docker-ce-cli (5:27.2.1-1~ubuntu.24.04~noble) ...
Setting up libslirp0:amd64 (4.7.0-1ubuntu3) ...
Setting up pigz (2.8-1) ...
Setting up docker-ce-rootless-extras (5:27.2.1-1~ubuntu.24.04~noble) ...
Setting up slirp4netns (1.2.1-1build2) ...
Setting up docker-ce (5:27.2.1-1~ubuntu.24.04~noble) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

sudo mkdir -p /etc/docker cat <<EOF | sudo
tee /etc/docker/daemon.json
{
"exec-opts": ["native.cgroupdriver=systemd"]
}

Name:Muskan chandiramani. D15C. Roll No: 5

EOF

```
ubuntu@ip-172-31-20-171:~$ sudo mkdir -p /etc/docker
ubuntu@ip-172-31-20-171:~$ cat <<EOF | sudo tee /etc/docker/daemon.json
{
    "exec-opts": ["native.cgroupdriver=systemd"]
}
EOF
{
    "exec-opts": ["native.cgroupdriver=systemd"]
}
```

sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker

```
ubuntu@ip-172-31-20-171:~$ sudo systemctl enable docker
Synchronizing state of docker.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable docker
```

Step 5: Run the below command to install Kubernets.
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg

echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list

```
ubuntu@ip-172-31-20-171:~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
ubuntu@ip-172-31-20-171:~$ echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /
```

Name:Muskan chandiramani. D15C. Roll No: 5

sudo apt-get update sudo apt-get install -y kubelet kubeadm kubectl sudo apt-mark hold kubelet kubeadm kubectl

```
ubuntu@ip-172-31-20-171:~$ sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:5 https://download.docker.com/linux/ubuntu noble InRelease
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  InRelease [1186 B]
Get:7 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  Packages [4865 B]
Fetched 6051 B in 0s (12.9 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  conntrack cri-tools kubernetes-cni
The following NEW packages will be installed:
  conntrack cri-tools kubeadm kubectl kubelet kubernetes-cni
0 upgraded, 6 newly installed, 0 to remove and 130 not upgraded.
Need to get 87.4 MB of archives.
After this operation, 314 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 conntrack amd64 1:1.4.8-1ubuntu1 [37.9 kB]
Get:2 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  cri-tools 1.31.1-1.1 [15.7 MB]
Get:3 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  kubeadm 1.31.1-1.1 [11.4 MB]
Get:4 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  kubectl 1.31.1-1.1 [11.2 MB]
Get:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  kubernetes-cni 1.5.1-1.1 [33.9 MB]
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  kubelet 1.31.1-1.1 [15.2 MB]
Fetched 87.4 MB in 1s (77.1 MB/s)
Selecting previously unselected package conntrack.
(Reading database ... 68011 files and directories currently installed.)
Preparing to unpack .../0-conntrack_1%3a1.4.8-1ubuntu1_amd64.deb ...
Unpacking conntrack (1:1.4.8-1ubuntu1) ...
Selecting previously unselected package cri-tools.
Preparing to unpack .../1-cri-tools_1.31.1-1.1_amd64.deb ...
Unpacking cri-tools (1.31.1-1.1) ...
Selecting previously unselected package kubeadm.
Preparing to unpack .../2-kubeadm_1.31.1-1.1_amd64.deb ...
Unpacking kubeadm (1.31.1-1.1) ...
```

Name:Muskan chandiramani. D15C. Roll No: 5

```
Unpacking cri-tools (1.31.1-1.1) ...
Selecting previously unselected package kubeadm.
Preparing to unpack .../2-kubeadm_1.31.1-1.1_amd64.deb ...
Unpacking kubeadm (1.31.1-1.1) ...
Selecting previously unselected package kubectl.
Preparing to unpack .../3-kubectl_1.31.1-1.1_amd64.deb ...
Unpacking kubectl (1.31.1-1.1) ...
Selecting previously unselected package kubernetes-cni.
Preparing to unpack .../4-kubernetes-cni_1.5.1-1.1_amd64.deb ...
Unpacking kubernetes-cni (1.5.1-1.1) ...
Selecting previously unselected package kubelet.
Preparing to unpack .../5-kubelet_1.31.1-1.1_amd64.deb ...
Unpacking kubelet (1.31.1-1.1) ...
Setting up conntrack (1:1.4.8-1ubuntu1) ...
Setting up kubectl (1.31.1-1.1) ...
Setting up cri-tools (1.31.1-1.1) ...
Setting up kubernetes-cni (1.5.1-1.1) ...
Setting up kubeadm (1.31.1-1.1) ...
Setting up kubelet (1.31.1-1.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
```

sudo systemctl enable --now kubelet sudo
kubeadm init --pod-network-cidr=10.244.0.0/16

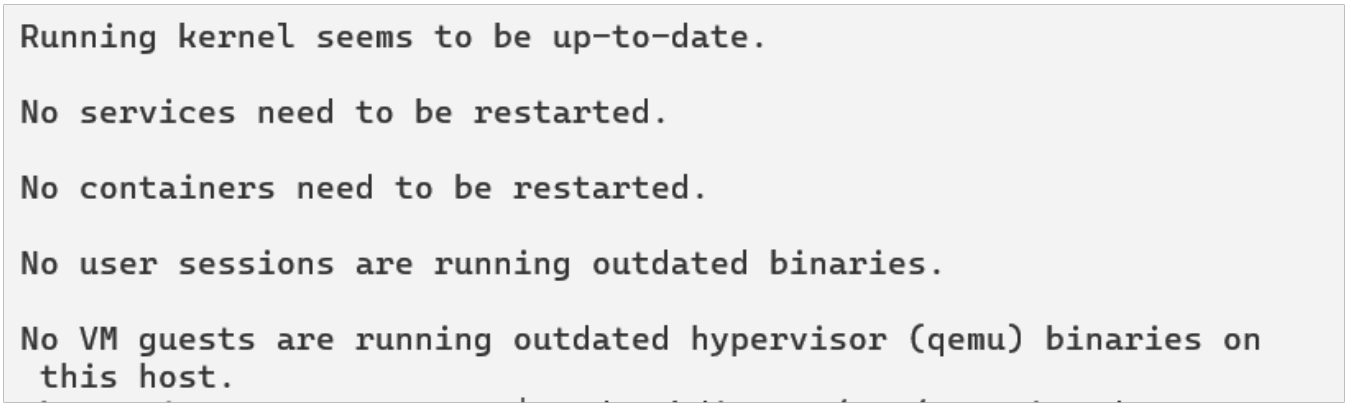
```
ubuntu@ip-172-31-20-171:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
W0915 07:47:37.419191    7952 checks.go:1080] [preflight] WARNING: Couldn't create the interface used for talking to the container runtime: failed to create
 new CRI runtime service: validate service connection: validate CRI v1 runtime API for endpoint "unix:///var/run/containerd/containerd.sock": rpc error: cod
e = Unimplemented desc = unknown service runtime.v1.RuntimeService
        [WARNING FileExisting-socat]: socat not found in system path
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
error execution phase preflight: [preflight] Some fatal errors occurred:
failed to create new CRI runtime service: validate service connection: validate CRI v1 runtime API for endpoint "unix:///var/run/containerd/containerd.sock"
: rpc error: code = Unimplemented desc = unknown service runtime.v1.RuntimeService[preflight] If you know what you are doing, you can make a check non-fatal
 with `--ignore-preflight-errors=...`
To see the stack trace of this error execute with --v=5 or higher
```

Now We have got an error.
So we have to perform some additional commands as follow.

Name:Muskan chandiramani. D15C. Roll No: 5

sudo apt-get install -y containerd

```
To see the stack trace of this error execute with --v=5 or higher   ubuntu@ip-172-31-20-171:~$ sudo apt-get install -y containerd
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras
  docker-compose-plugin libltdl7 libslirp0 pigz slirp4netns
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  runc
The following packages will be REMOVED:
  containerd.io docker-ce
The following NEW packages will be installed:
  containerd runc
0 upgraded, 2 newly installed, 2 to remove and 130 not upgraded.
Need to get 47.2 MB of archives.
After this operation, 53.1 MB disk space will be freed.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.1.12-0ubuntu3.1 [8599 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 containerd amd64 1.7.12-0ubuntu4.1 [38.6 MB]
Fetched 47.2 MB in 1s (74.5 MB/s)
(Reading database ... 68068 files and directories currently installed.)
Removing docker-ce (5:27.2.1-1~ubuntu.24.04~noble) ...
Removing containerd.io (1.7.22-1) ...
Selecting previously unselected package runc.
(Reading database ... 68048 files and directories currently installed.)
Preparing to unpack .../runc_1.1.12-0ubuntu3.1_amd64.deb ...
Unpacking runc (1.1.12-0ubuntu3.1) ...
Selecting previously unselected package containerd.
Preparing to unpack .../containerd_1.7.12-0ubuntu4.1_amd64.deb ...
Unpacking containerd (1.7.12-0ubuntu4.1) ...
Setting up runc (1.1.12-0ubuntu3.1) ...
Setting up containerd (1.7.12-0ubuntu4.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.
```



Name:Muskan chandiramani. D15C. Roll No: 5

sudo mkdir -p /etc/containerd sudo containerd config default | sudo tee /etc/containerd/config.toml

```
ubuntu@ip-172-31-20-171:~$ sudo mkdir -p /etc/containerd
sudo containerd config default | sudo tee /etc/containerd/config.toml
disabled_plugins = []
imports = []
oom_score = 0
plugin_dir = ""
required_plugins = []
root = "/var/lib/containerd"
state = "/run/containerd"
temp = ""
version = 2

[cgroup]
  path = ""

[debug]
  address = ""
  format = ""
  gid = 0
  level = ""
  uid = 0

[grpc]
  address = "/run/containerd/containerd.sock"
  gid = 0
  max_recv_message_size = 16777216
  max_send_message_size = 16777216
  tcp_address = ""
  tcp_tls_ca = ""
  tcp_tls_cert = ""
  tcp_tls_key = ""
  uid = 0

[metrics]
  address = ""
  grpc_histogram = false

[plugins]

  [plugins."io.containerd.gc.v1.scheduler"]
    deletion_threshold = 0
```

...

Name:Muskan chandiramani. D15C. Roll No: 5

sudo systemctl restart containerd
sudo systemctl enable containerd
sudo systemctl status containerd

```
ubuntu@ip-172-31-20-171:~$ sudo systemctl restart containerd
sudo systemctl enable containerd
ubuntu@ip-172-31-20-171:~$ sudo systemctl status containerd
● containerd.service - containerd container runtime
     Loaded: loaded (/usr/lib/systemd/system/containerd.service; en>
     Active: active (running) since Sun 2024-09-15 07:49:23 UTC; 5s>
       Docs: https://containerd.io
   Main PID: 8398 (containerd)
      Tasks: 7
     Memory: 13.5M (peak: 14.0M)
        CPU: 70ms
     CGroup: /system.slice/containerd.service
             └─8398 /usr/bin/containerd

Sep 15 07:49:23 ip-172-31-20-171 containerd[8398]: time="2024-09-15>
Sep 15 07:49:23 ip-172-31-20-171 containerd[8398]: time="2024-09-15>
Sep 15 07:49:23 ip-172-31-20-171 containerd[8398]: time="2024-09-15>
Sep 15 07:49:23 ip-172-31-20-171 containerd[8398]: time="2024-09-15>
Sep 15 07:49:23 ip-172-31-20-171 containerd[8398]: time="2024-09-15>
Sep 15 07:49:23 ip-172-31-20-171 containerd[8398]: time="2024-09-15>
Sep 15 07:49:23 ip-172-31-20-171 containerd[8398]: time="2024-09-15>
Sep 15 07:49:23 ip-172-31-20-171 containerd[8398]: time="2024-09-15>
Sep 15 07:49:23 ip-172-31-20-171 systemd[1]: Started containerd.ser>
Sep 15 07:49:23 ip-172-31-20-171 containerd[8398]: time="2024-09-15>
```

sudo apt-get install -y socat

```
ubuntu@ip-172-31-20-171:~$ sudo apt-get install -y socat
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras
  docker-compose-plugin libltdl7 libslirp0 pigz slirp4netns
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  socat
0 upgraded, 1 newly installed, 0 to remove and 130 not upgraded.
Need to get 374 kB of archives.
After this operation, 1649 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 socat amd64 1.8.0.0-4build3 [374 kB]
Fetched 374 kB in 0s (12.1 MB/s)
Selecting previously unselected package socat.
(Reading database ... 68112 files and directories currently installed.)
Preparing to unpack .../socat_1.8.0.0-4build3_amd64.deb ...
Unpacking socat (1.8.0.0-4build3) ...
Setting up socat (1.8.0.0-4build3) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on
 this host.
```

Step 6: Initialize the Kubecluster

Name:Muskan chandiramani. D15C. Roll No: 5

sudo kubeadm init --pod-network-cidr=10.244.0.0/16

```
ubuntu@ip-172-31-20-171:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
W0915 07:49:42.979051    8570 checks.go:846] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent with that used by kubeadm.It is recommende
d to use "registry.k8s.io/pause:3.10" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-20-171 kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 172.3
1.20.171]
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key
[certs] Generating "etcd/ca" certificate and key
[certs] Generating "etcd/server" certificate and key
[certs] etcd/server serving cert is signed for DNS names [ip-172-31-20-171 localhost] and IPs [172.31.20.171 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-20-171 localhost] and IPs [172.31.20.171 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
[certs] Generating "sa" key and public key
[kubeconfig] Using kubeconfig folder "/etc/kubernetes"
[kubeconfig] Writing "admin.conf" kubeconfig file
[kubeconfig] Writing "super-admin.conf" kubeconfig file
[kubeconfig] Writing "kubelet.conf" kubeconfig file
[kubeconfig] Writing "controller-manager.conf" kubeconfig file
[kubeconfig] Writing "scheduler.conf" kubeconfig file
[etcd] Creating static Pod manifest for local etcd in "/etc/kubernetes/manifests"
[control-plane] Using manifest folder "/etc/kubernetes/manifests"
[control-plane] Creating static Pod manifest for "kube-apiserver"
[control-plane] Creating static Pod manifest for "kube-controller-manager"
[control-plane] Creating static Pod manifest for "kube-scheduler"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Starting the kubelet
[wait-control-plane] Waiting for the kubelet to boot up the control plane as static Pods from directory "/etc/kubernetes/manifests"
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 502.777379ms
[api-check] Waiting for a healthy API server. This can take up to 4m0s
[api-check] The API server is healthy after 4.501245501s
[upload-config] Storing the configuration used in ConfigMap "kubeadm-config" in the "kube-system" Namespace
[kubelet] Creating a ConfigMap "kubelet-config" in namespace kube-system with the configuration for the kubelets in the cluster
[upload-certs] Skipping phase. Please see --upload-certs
[mark-control-plane] Marking the node ip-172-31-20-171 as control-plane by adding the labels: [node-role.kubernetes.io/control-plane node.kubernetes.io/exclude-from-external-load-balancers
]
[mark-control-plane] Marking the node ip-172-31-20-171 as control-plane by adding the taints [node-role.kubernetes.io/control-plane:NoSchedule]
[bootstrap-token] Using token: 7acddu.inheshzwxti0372v
```

```
[mark-control-plane] Marking the node ip-172-31-20-171 as control-plane by adding the taints [node-role.kubernetes.io/control-plane:NoSchedule]
[bootstrap-token] Using token: 7acddu.inheshzwxti0372v
[bootstrap-token] Configuring bootstrap tokens, cluster-info ConfigMap, RBAC Roles
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to get nodes
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to post CSRs in order for nodes to get long term certificate credentials
[bootstrap-token] Configured RBAC rules to allow the csrapprover controller automatically approve CSRs from a Node Bootstrap Token
[bootstrap-token] Configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
[bootstrap-token] Creating the "cluster-info" ConfigMap in the "kube-public" namespace
[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable kubelet client certificate and key
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy

Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config

Alternatively, if you are the root user, you can run:

  export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 172.31.20.171:6443 --token 7acddu.inheshzwxti0372v \
        --discovery-token-ca-cert-hash sha256:aed5faf97bac361d1bb7f33a89fb05d2bb28c7fc065024eac2302a734c330a36
```

Copy the mkdir and chown commands from the top and execute them.
mkdir -p $HOME/.kube sudo cp -i /etc/kubernetes/admin.conf
$HOME/.kube/config sudo chown $(id -u):$(id -g) $HOME/.kube/config

```
ubuntu@ip-172-31-20-171:~$  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

Add a common networking plugin called flannel as mentioned in the code.
kubectl apply -f
https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml

Name:Muskan chandiramani. D15C. Roll No: 5

```
ubuntu@ip-172-31-20-171:~$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
```

Step 7: Now that the cluster is up and running, we can deploy our nginx server on this cluster.Apply this deployment file using this command to create a deployment

kubectl apply -f https://k8s.io/examples/application/deployment.yaml

```
ubuntu@ip-172-31-20-171:~$ kubectl apply -f https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
```

kubectl get pods

```
ubuntu@ip-172-31-20-171:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-vz8rv    0/1     Pending   0          8s
nginx-deployment-d556bf558-wz5wc    0/1     Pending   0          8s
```

POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath="{.items[0].metadata.name}")

kubectl port-forward $POD_NAME 8080:80

```
ubuntu@ip-172-31-20-171:~$ POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath="{.items[0].metadata.name}")
ubuntu@ip-172-31-20-171:~$ kubectl port-forward $POD_NAME 8080:80
error: unable to forward port because pod is not running. Current status=Pending
```

Note : We have faced an error as pod status is pending so make it running run below commands then again run above 2 commands.

kubectl taint nodes --all node-role.kubernetes.io/control-plane-node/ip-172-31-20-171

untainted kubectl get nodes

```
ubuntu@ip-172-31-20-171:~$ kubectl taint nodes --all node-role.kubernetes.io/control-plane-
node/ip-172-31-20-171 untainted
ubuntu@ip-172-31-20-171:~$ kubectl get nodes
NAME              STATUS   ROLES           AGE     VERSION
ip-172-31-20-171   Ready    control-plane   5m23s   v1.31.1
```

kubectl get pods

```
ubuntu@ip-172-31-20-171:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-vz8rv    1/1     Running   0          3m4s
nginx-deployment-d556bf558-wz5wc    1/1     Running   0          3m4s
```

Name:Muskan chandiramani. D15C. Roll No: 5

POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath="{.items[0].metadata.name}")

kubectl port-forward $POD_NAME 8080:80

```
ubuntu@ip-172-31-20-171:~$ POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath="{.items[0].metadata.name}")
kubectl port-forward $POD_NAME 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
Handling connection for 8080
```

Step 8: Verify your deployment
Open up a new terminal and ssh to your EC2 instance.
Then, use this curl command to check if the Nginx server is running.

curl --head http://127.0.0.1:8080

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\bhush\one drive 2\OneDrive\Desktop\New folder (4)> ssh -i "Master_Ec2_Key.pem" ubuntu@ec2-54-196-129-215.compute-1.amazonaws.com
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Sun Sep 15 07:58:53 UTC 2024

  System load:  0.15              Processes:             152
  Usage of /:   55.3% of 6.71GB   Users logged in:       1
  Memory usage: 20%               IPv4 address for enX0: 172.31.20.171
  Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

132 updates can be applied immediately.
38 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


Last login: Sun Sep 15 07:27:23 2024 from 152.58.2.47
```

```
ubuntu@ip-172-31-20-171:~$ curl --head http://127.0.0.1:8080
HTTP/1.1 200 OK
Server: nginx/1.14.2
Date: Sun, 15 Sep 2024 07:59:03 GMT
Content-Type: text/html
Content-Length: 612
Last-Modified: Tue, 04 Dec 2018 14:44:49 GMT
Connection: keep-alive
ETag: "5c0692e1-264"
Accept-Ranges: bytes

ubuntu@ip-172-31-20-171:~$
```

Name:Muskan chandiramani. **D15C.** **Roll No: 5**

If the response is 200 OK and you can see the Nginx server name, your deployment was successful.

We have successfully deployed our Nginx server on our EC2 instance.

Conclusion:

In this experiment, we successfully installed Kubernetes on an EC2 instance and deployed an Nginx server using Kubectl commands. During the process, we encountered two main errors: the Kubernetes pod was initially in a pending state, which was resolved by removing the control-plane taint using kubectl taint nodes --all, and we also faced an issue with the missing containerd runtime, which was fixed by installing and starting containerd. We used a **t2.medium EC2 instance with 2 CPUs** to meet the necessary resource requirements for the Kubernetes setup and deployment.

Experiment 5

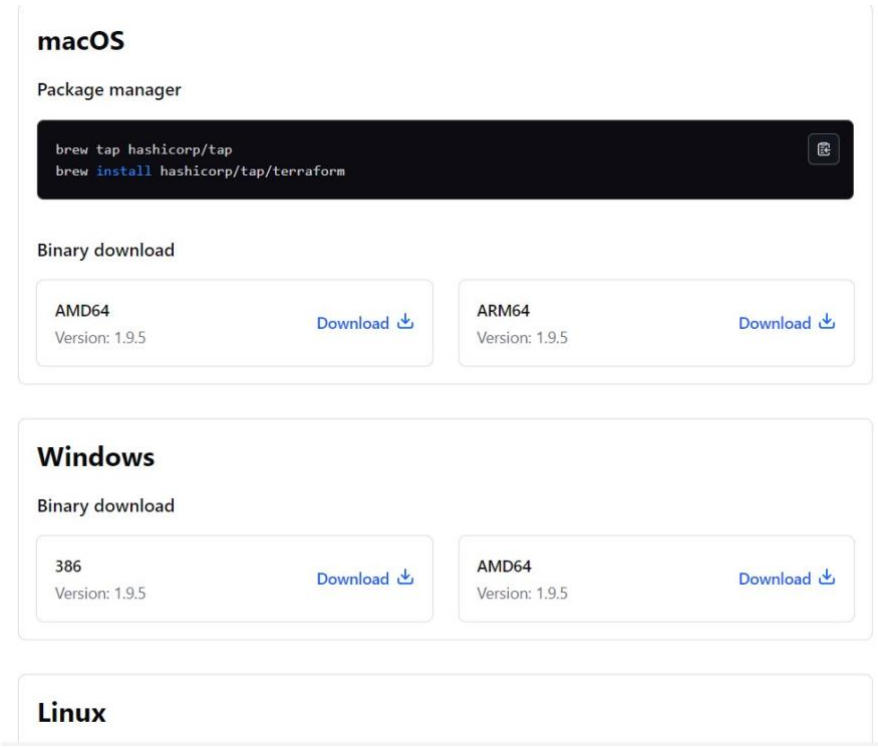
A) Installation and Configuration of Terraform in Windows

Step 1: Download terraform

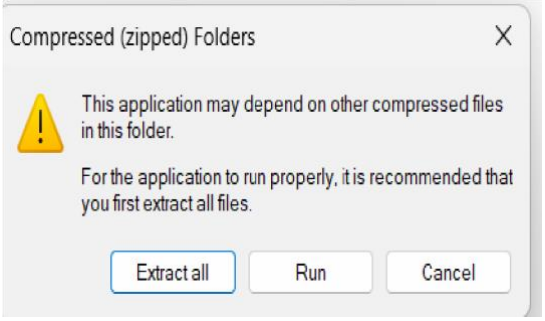
To install Terraform, First Download the Terraform Cli Utility for windows from terraforms official website

website:https://www.terraform.io/downloads.html

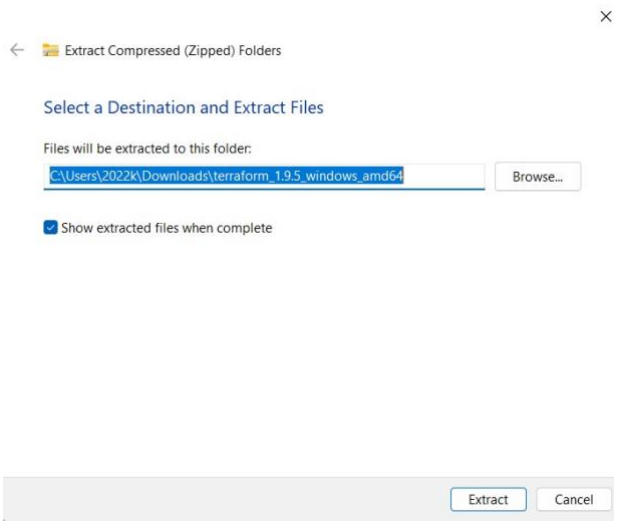
Select the Operating System Windows followed by either 32bit or 64 bit based on your OS type.



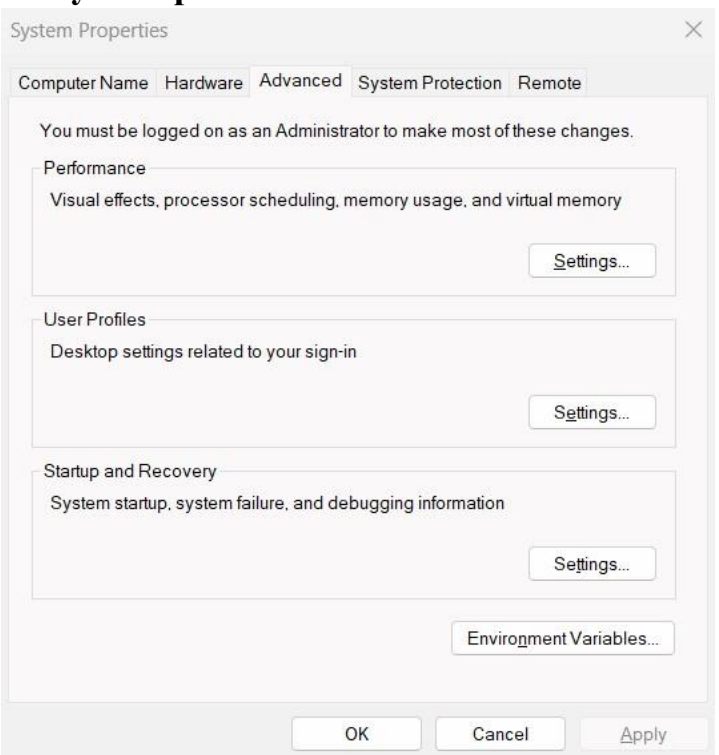
Step 2: Extract the downloaded setup file Terraform.exe in

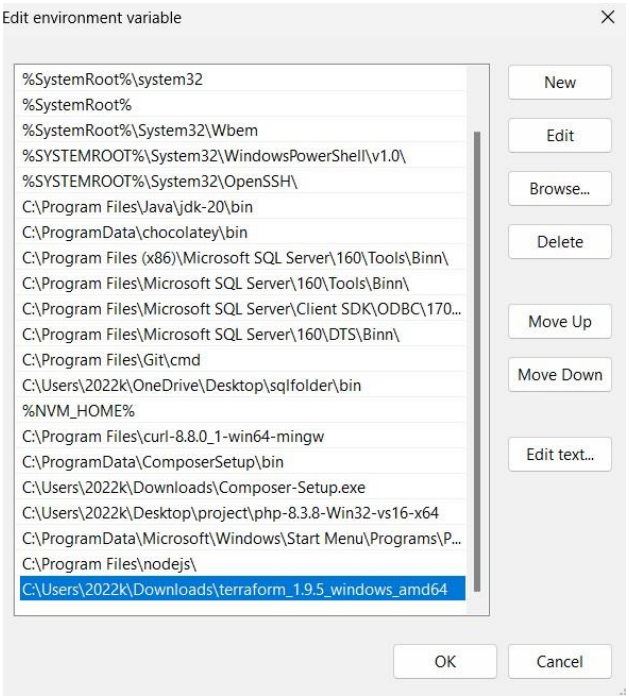


C:\Terraform directory

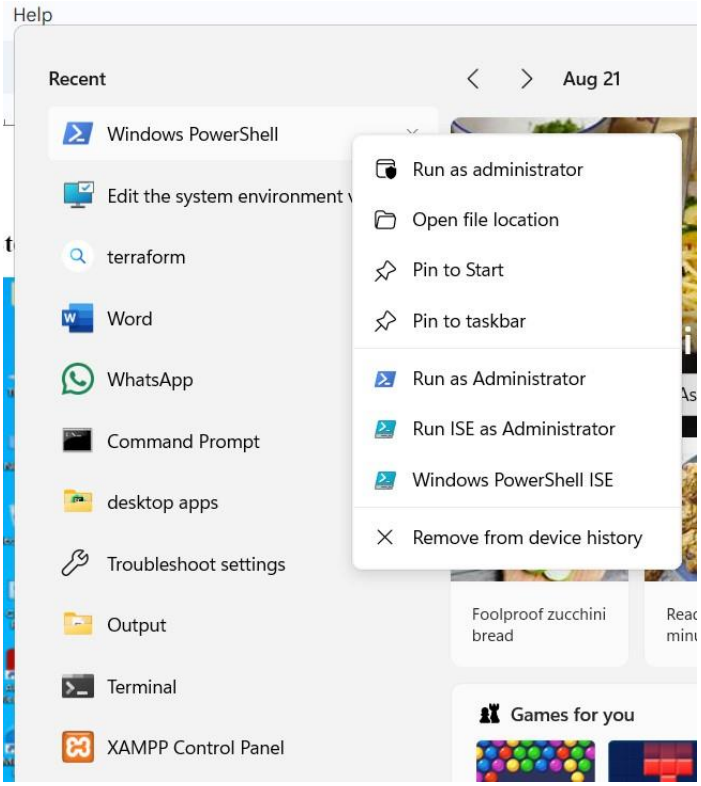


Step 3: Set the System path for Terraform in Environment Variables

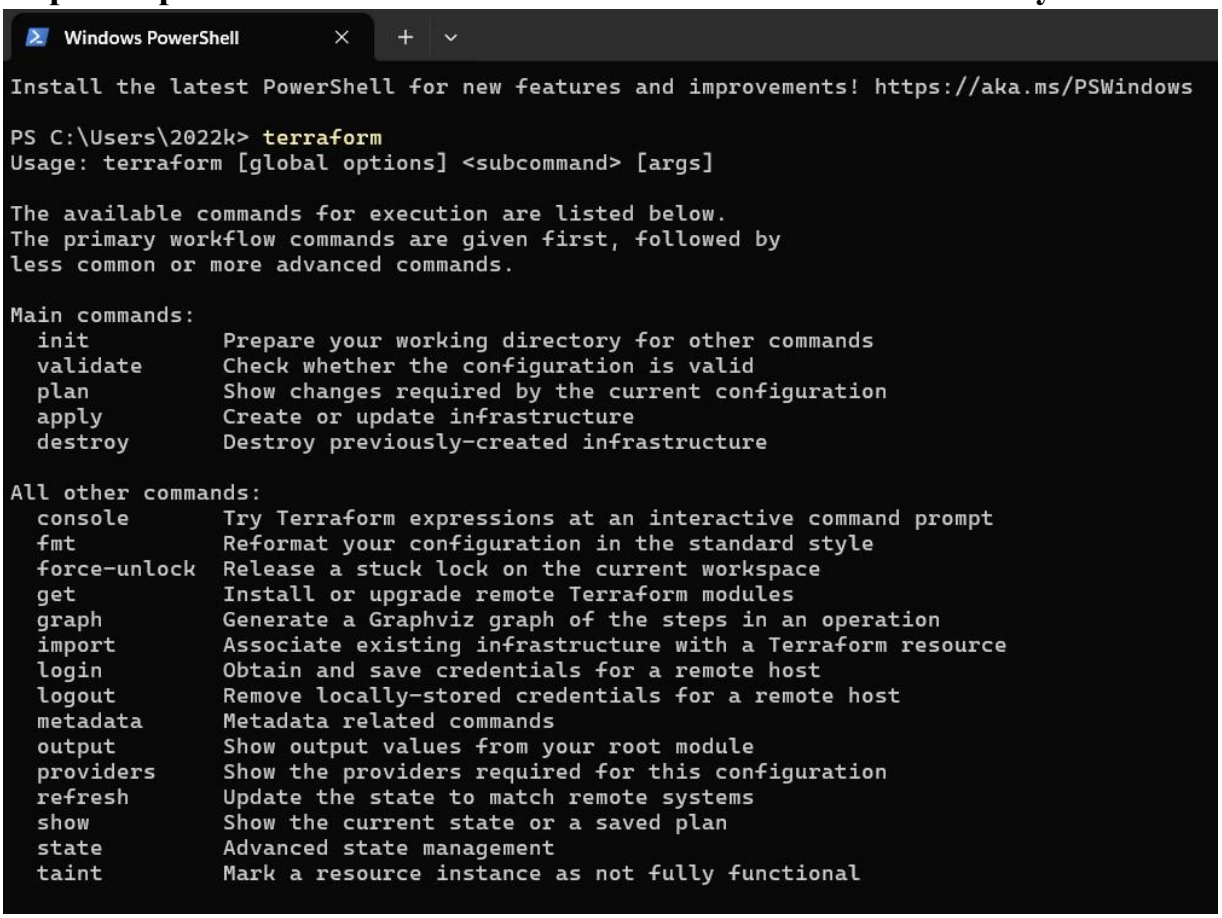




Step 4: Open PowerShell with Admin Access



Step 5 : Open Terraform in PowerShell and check its functionality



Name: Muskan Chandiramani D15C Roll No: 05

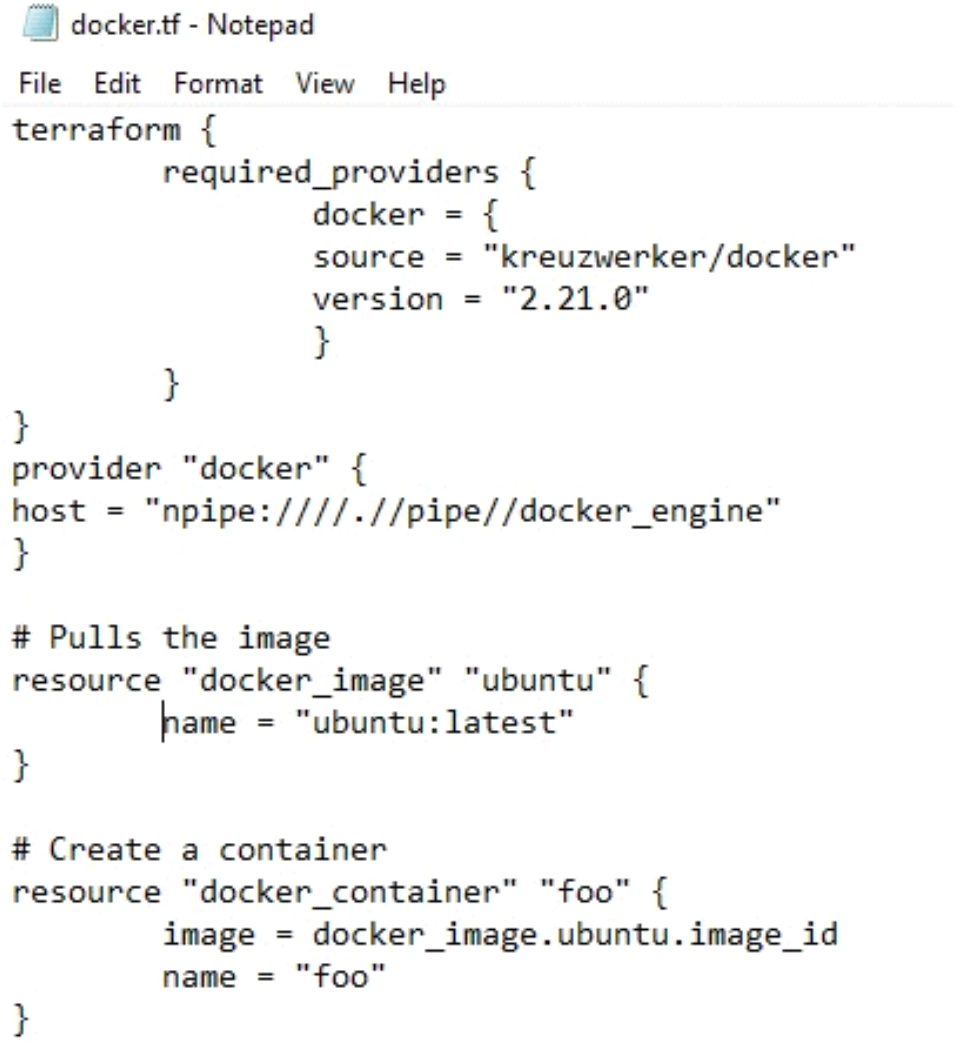
Experiment No. 6

- Creating a docker image using terraform

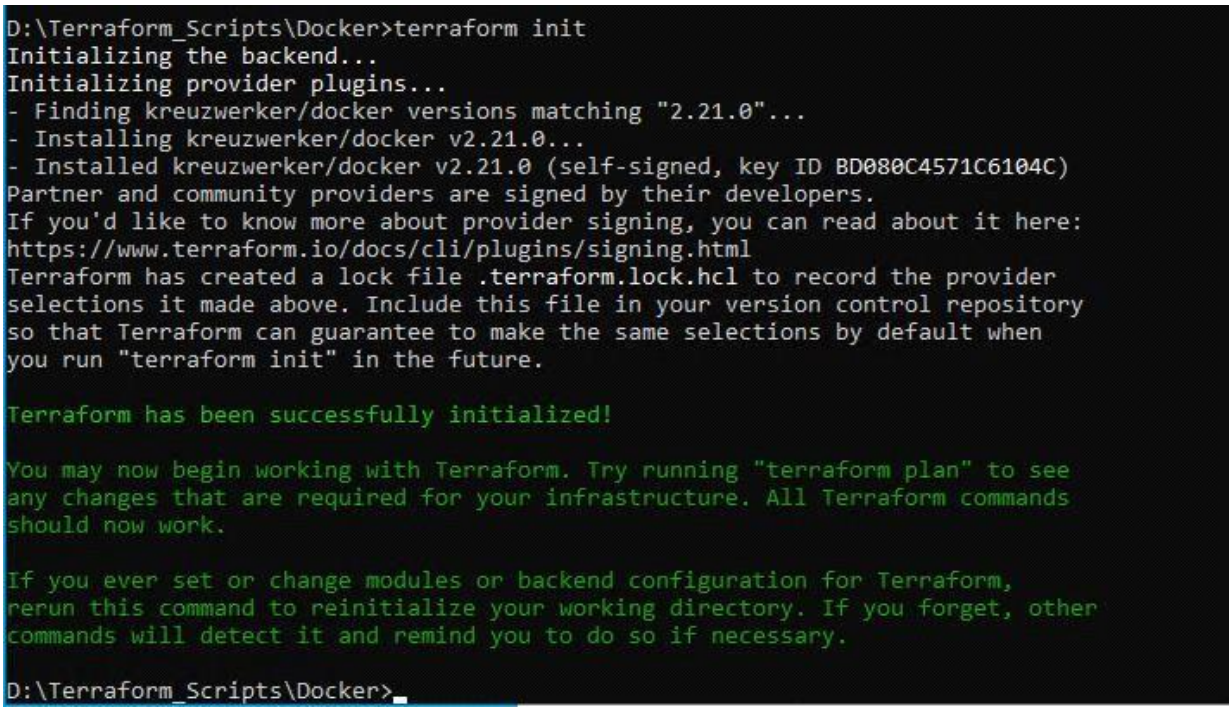
```
PS C:\ProgramData\Microsoft\Windows\Start Menu> Docker --version
Docker version 27.1.1, build 6312585
PS C:\ProgramData\Microsoft\Windows\Start Menu> docker

Usage:  docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Common Commands:
  run         Create and run a new container from an image
  exec        Execute a command in a running container
  ps          List containers
  build       Build an image from a Dockerfile
  pull        Download an image from a registry
  push        Upload an image to a registry
  images      List images
  login       Log in to a registry
  logout      Log out from a registry
  search      Search Docker Hub for images
  version     Show the Docker version information
  info        Display system-wide information

Management Commands:
  builder     Manage builds
  buildx*     Docker Buildx
  checkpoint  Manage checkpoints
  compose*    Docker Compose
  container   Manage containers
  context     Manage contexts
  debug*      Get a shell into any image or container
  desktop*    Docker Desktop commands (Alpha)
  dev*        Docker Dev Environments
  extension*  Manages Docker extensions
  feedback*   Provide feedback, right in your terminal!
  image       Manage images
```

- Create a folder named ‘Terraform Scripts’ in which we save our different types of scripts which will be further used in this experiment.

- Create a new folder named ‘Docker’ in the ‘TerraformScripts’ folder. Then create a new docker.tf file using a text editor and write the following contents into it to create a Ubuntu Linux container.



- Execute Terraform Init command to initialize the resources



- Execute Terraform plan to see the available resources

```
D:\Terraform_Scripts\Docker>terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the
following symbols:
  + create

Terraform will perform the following actions:

  # docker_container.foo will be created
  + resource "docker_container" "foo" {
      + attach           = false
      + bridge           = (known after apply)
      + command          = (known after apply)
      + container_logs   = (known after apply)
      + entrypoint       = (known after apply)
      + env              = (known after apply)
      + exit_code        = (known after apply)
      + gateway          = (known after apply)
      + hostname         = (known after apply)
      + id               = (known after apply)
      + image            = (known after apply)
      + init             = (known after apply)
      + ip_address       = (known after apply)
      + ip_prefix_length = (known after apply)
      + ipc_mode         = (known after apply)
      + log_driver       = (known after apply)
      + logs             = false
      + must_run         = true
      + name             = "foo"
      + network_data     = (known after apply)
      + read_only        = false
      + remove_volumes   = true
      + restart          = "no"
      + rm               = false
      + runtime          = (known after apply)
      + security_opts    = (known after apply)
      + shm_size         = (known after apply)
      + start            = true
      + stdin_open       = false
      + stop_signal      = (known after apply)
      + stop_timeout     = (known after apply)
      + tty              = false

      + healthcheck (known after apply)

      + labels (known after apply)
    }

  # docker_image.ubuntu will be created
  + resource "docker_image" "ubuntu" {
      + id          = (known after apply)
      + image_id    = (known after apply)
      + latest      = (known after apply)
      + name        = "ubuntu:latest"
      + output      = (known after apply)
      + repo_digest = (known after apply)
    }

Plan: 2 to add, 0 to change, 0 to destroy.
```

- Execute Terraform apply to apply the configuration, which will automatically create and run the Ubuntu Linux container based on our configuration. Using command : “terraform apply”

- Check Docker images, Before and After Executing Apply step

```
D:\Terraform_Scripts\Docker>docker images
REPOSITORY                                  TAG                                 IMAGE ID       CREATED       SIZE
mcr.microsoft.com/dotnet/framework/aspnet   4.8-windowsservercore-ltsc2022      0b1ef1176a57   6 weeks ago   5.43GB
mcr.microsoft.com/dotnet/framework/sdk      4.8-windowsservercore-ltsc2022      c3f8c2735565   6 weeks ago   9.04GB
mcr.microsoft.com/dotnet/framework/runtime  4.8-windowsservercore-ltsc2022      e69ea8a5ec1b   6 weeks ago   5.1GB
mcr.microsoft.com/windows/servercore        ltsc2022                            e60f47e635b7   7 weeks ago   4.84GB
mcr.microsoft.com/windows/nanoserver        ltsc2022                            f0ca29645006   7 weeks ago   292MB

D:\Terraform_Scripts\Docker>
```

```
D:\Terraform_Scripts\Docker>docker images
REPOSITORY                                  TAG                                 IMAGE ID       CREATED          SIZE
mcr.microsoft.com/dotnet/framework/aspnet   4.8-windowsservercore-ltsc2022      0b1ef1176a57   6 weeks ago      5.43GB
mcr.microsoft.com/dotnet/framework/sdk      4.8-windowsservercore-ltsc2022      c3f8c2735565   6 weeks ago      9.04GB
mcr.microsoft.com/dotnet/framework/runtime  4.8-windowsservercore-ltsc2022      e69ea8a5ec1b   6 weeks ago      5.1GB
mcr.microsoft.com/windows/servercore        ltsc2022                            e60f47e635b7   7 weeks ago      4.84GB
mcr.microsoft.com/windows/nanoserver        ltsc2022                            f0ca29645006   7 weeks ago      292MB
ubuntu                                      Latest                              2dc39ba859dc   2 minutes ago    77.8MB
```

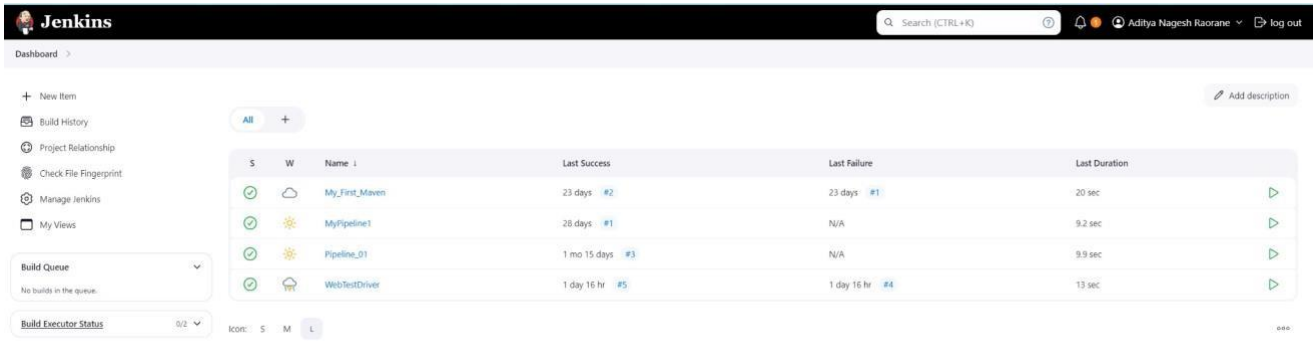
- Execute Terraform destroy to delete the configuration, which will automatically delete the Ubuntu Container.

- Check Docker images, After Executing Destroy step

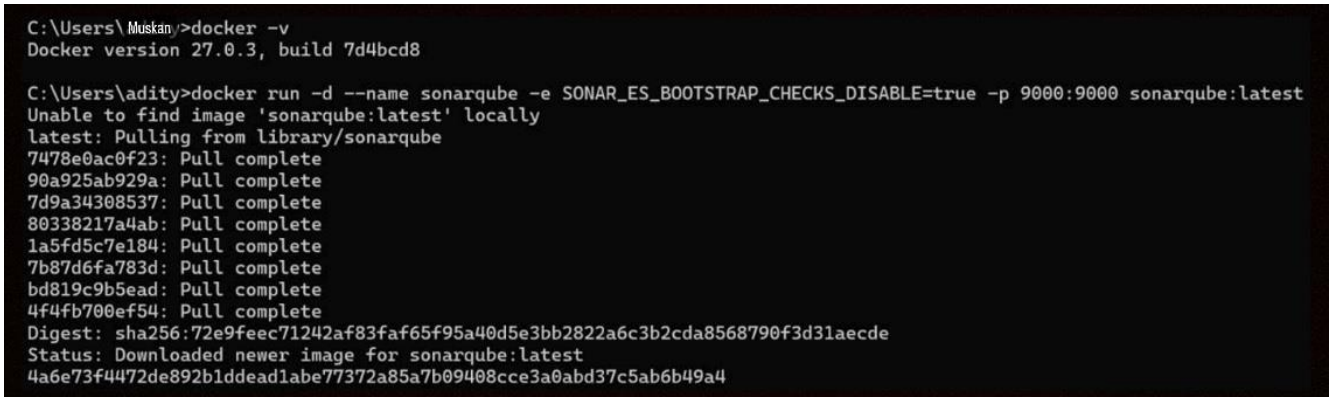
```
D:\Terraform_Scripts\Docker>docker images
REPOSITORY                                  TAG                                 IMAGE ID       CREATED       SIZE
mcr.microsoft.com/dotnet/framework/aspnet   4.8-windowsservercore-ltsc2022      0b1ef1176a57   6 weeks ago   5.43GB
mcr.microsoft.com/dotnet/framework/sdk      4.8-windowsservercore-ltsc2022      c3f8c2735565   6 weeks ago   9.04GB
mcr.microsoft.com/dotnet/framework/runtime  4.8-windowsservercore-ltsc2022      e69ea8a5ec1b   6 weeks ago   5.1GB
mcr.microsoft.com/windows/servercore        ltsc2022                            e60f47e635b7   7 weeks ago   4.84GB
mcr.microsoft.com/windows/nanoserver        ltsc2022                            f0ca29645006   7 weeks ago   292MB

D:\Terraform_Scripts\Docker>
```

Aim: To understand Static Analysis SAST process and learn to integrate Jenkins SAST to SonarQube/GitLab.

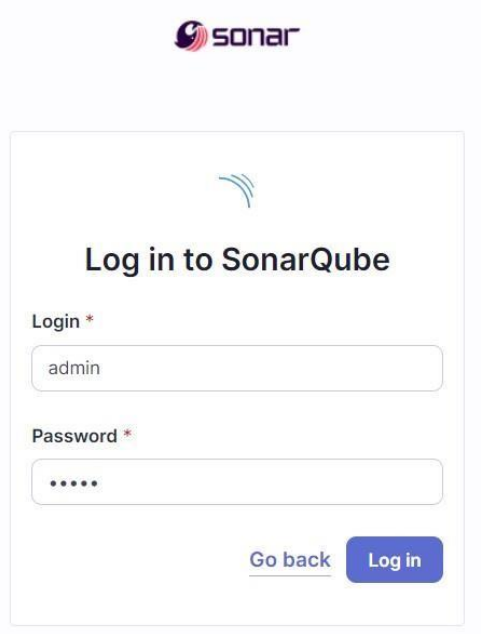
1. Open up Jenkins Dashboard on localhost, port 8080 or whichever port it is at for you.



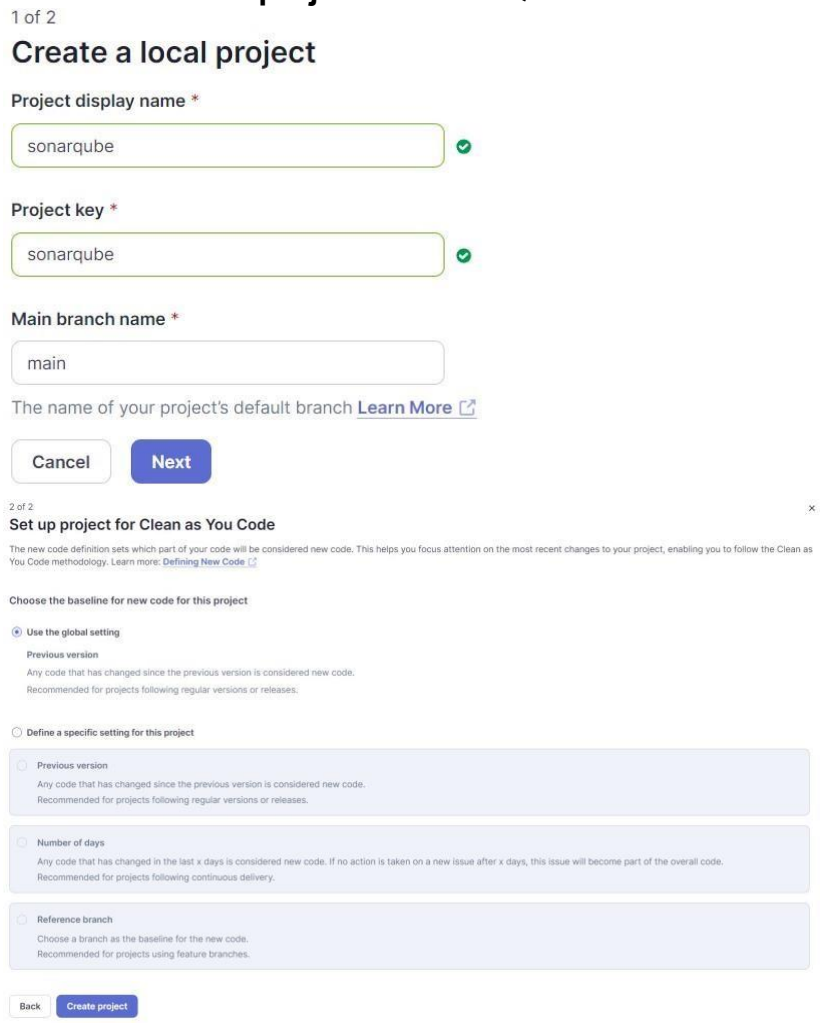
2. Run SonarQube in a Docker container using this command :a] docker -v b] docker pull sonarqube c] docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest



3. Once the container is up and running, you can check the status of SonarQube at **localhost port 9000**. The login id is "**admin**" and the password is **mus12**



4. **Create a local project in SonarQube** with the name **sonarqube**

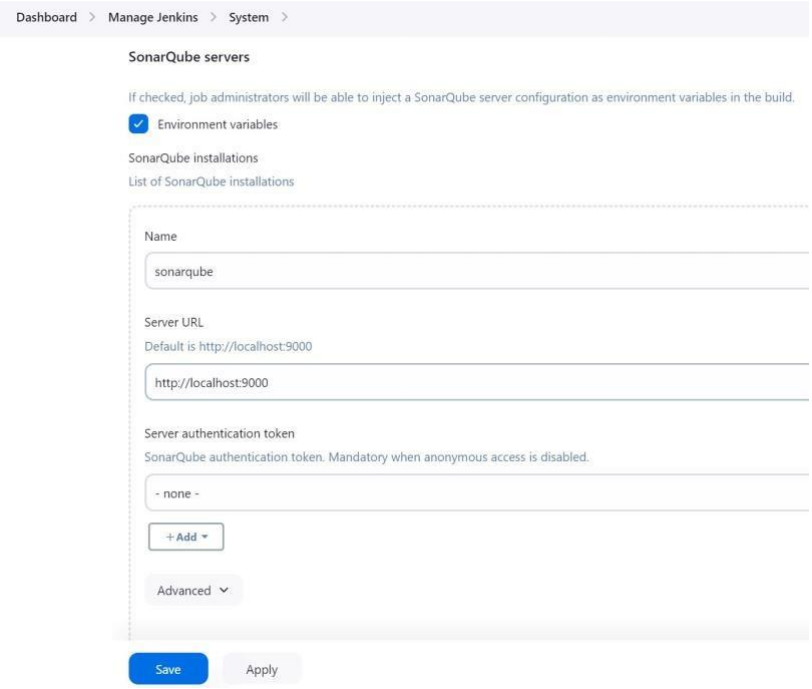


5. Setup the project and come back to Jenkins Dashboard. Go to **Manage Jenkins → Plugins** and search for **SonarQube Scanner** in **Available Plugins** and install it.

6. Under '**Manage Jenkins → System**', look for **SonarQube Servers** and enter these details.

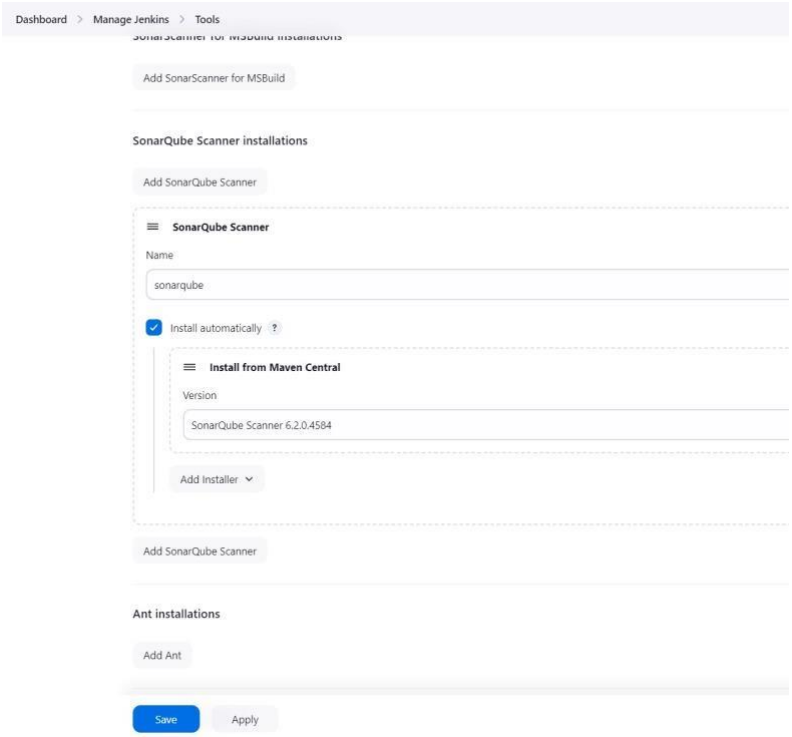
Name : sonarqube

Server URL : http://localhost:9000

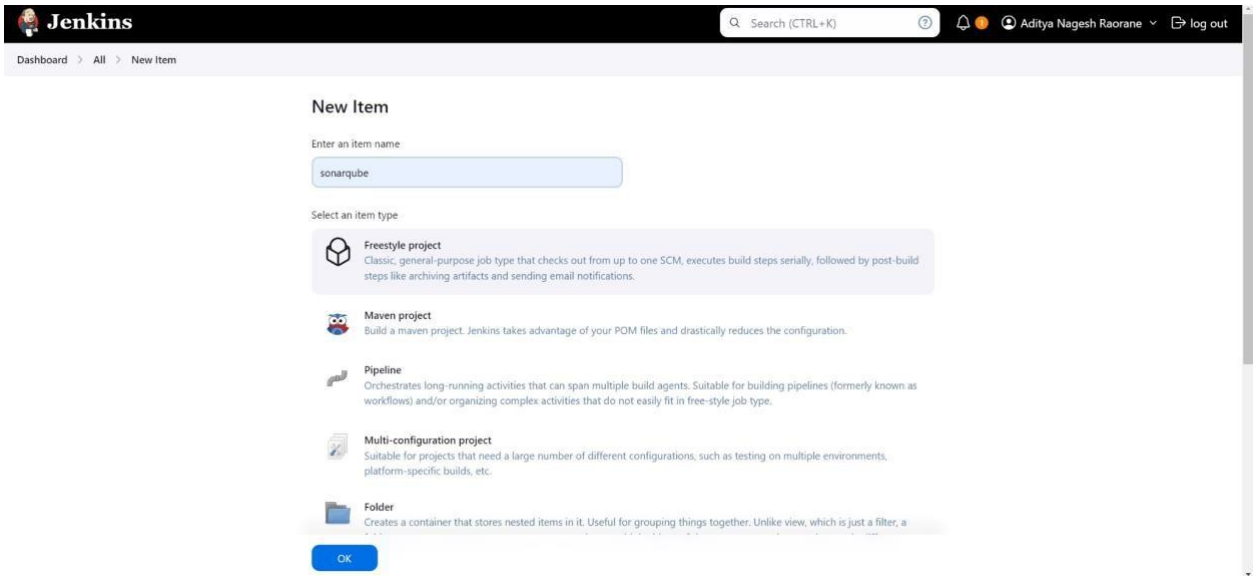


7. Search for SonarQube Scanner under Global Tool Configuration. Choose the latest configuration and choose Install automatically.

Manage Jeknins → Tools → SonarQube Scanner Installation

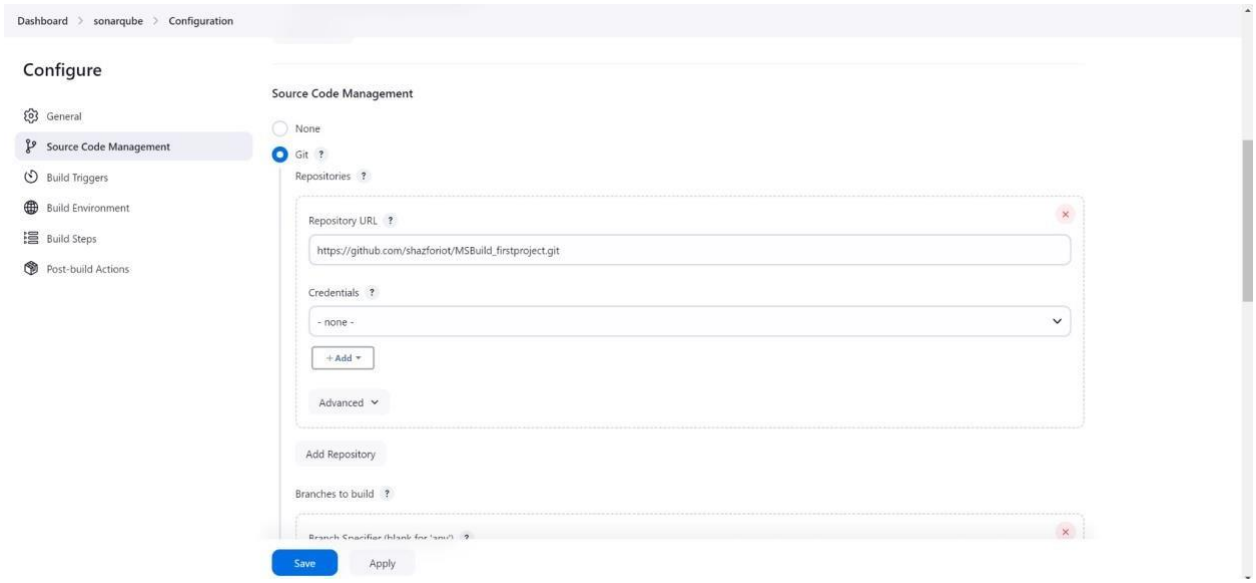


8. After the configuration, create a **New Item** in Jenkins, choose a **freestyle project** named **sonarqube**.



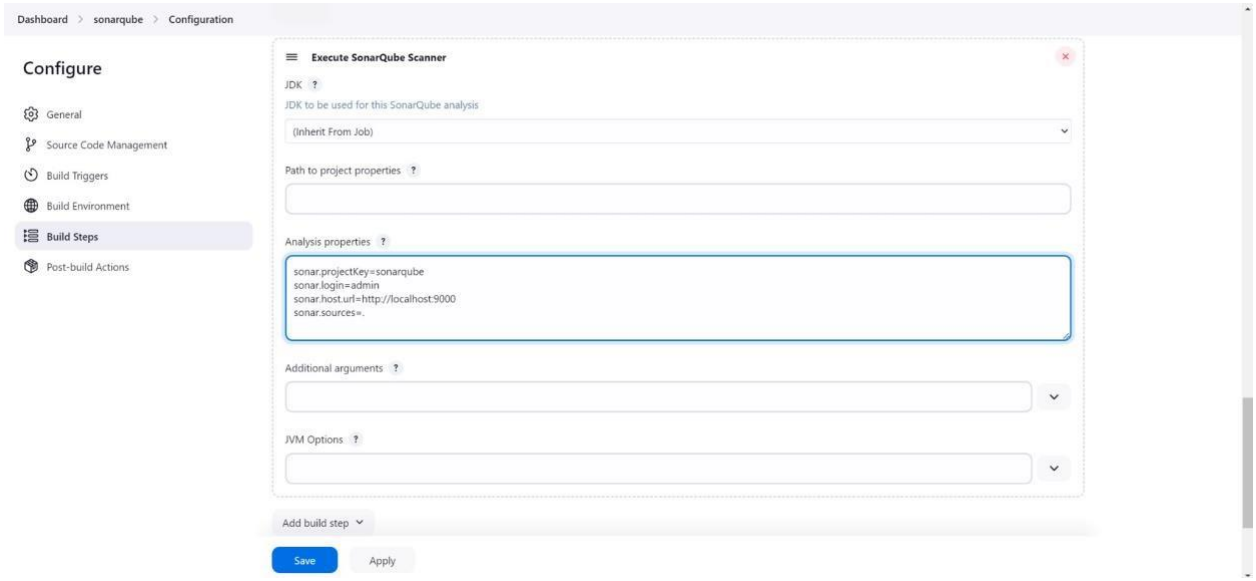
9. Choose this GitHub repository in **Source Code Management**.
   https://github.com/shazforiot/MSBuild_firstproject.git

It is a sample hello-world project with no vulnerabilities and issues, just to test the integration.

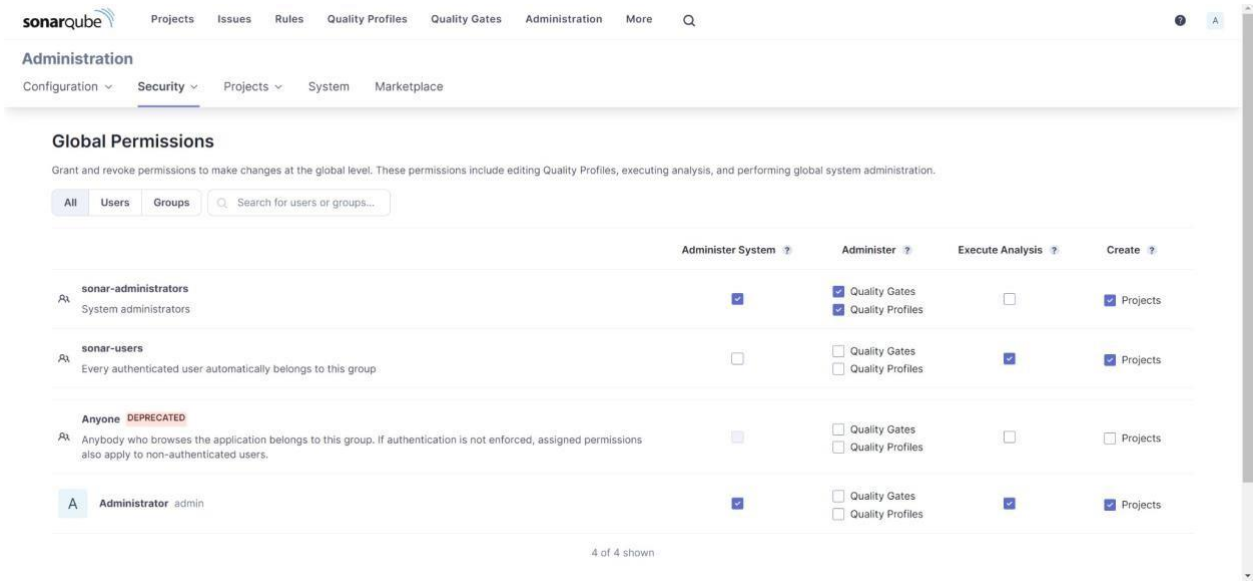


10. Under **Build-> Execute SonarQube Scanner**, enter these **Analysis Properties**. Mention the SonarQube Project Key, Login, Password, Source path and Host URL. sonar.projectKey=sonarqube sonar.login=admin sonar.password=aditya sonar.sources=.

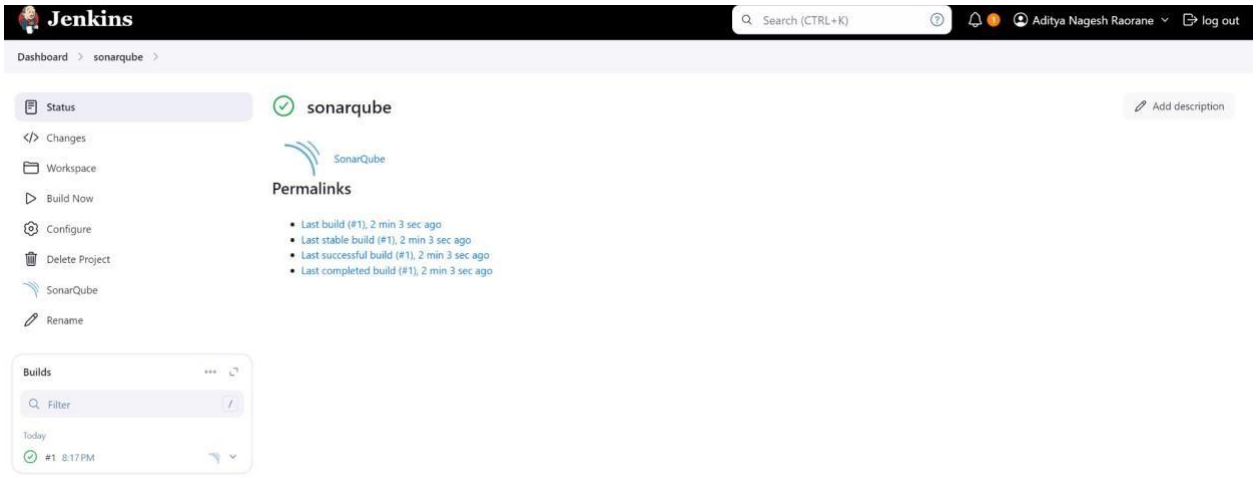
sonar.host.url=http://localhost:9000

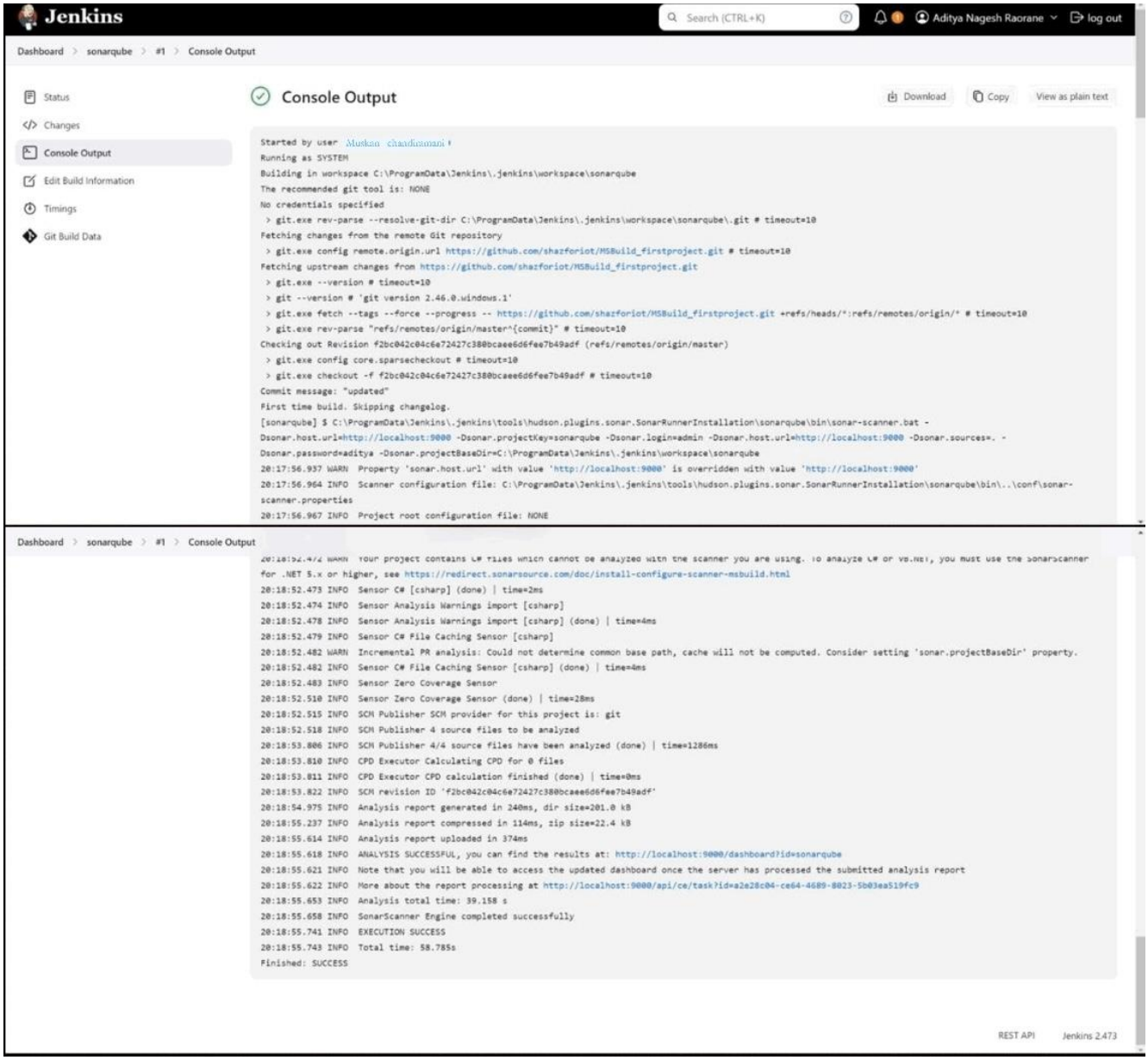


11. Go to http://localhost:9000/admin/permissions and allow Execute Permissions to the Admin user.

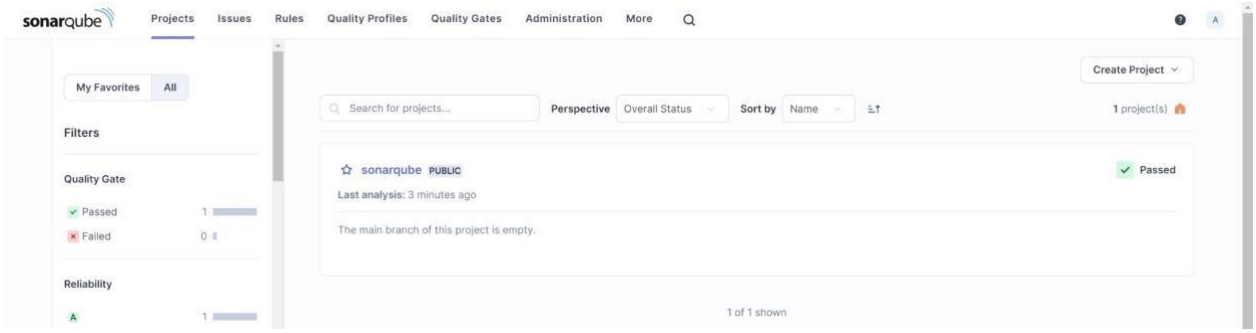


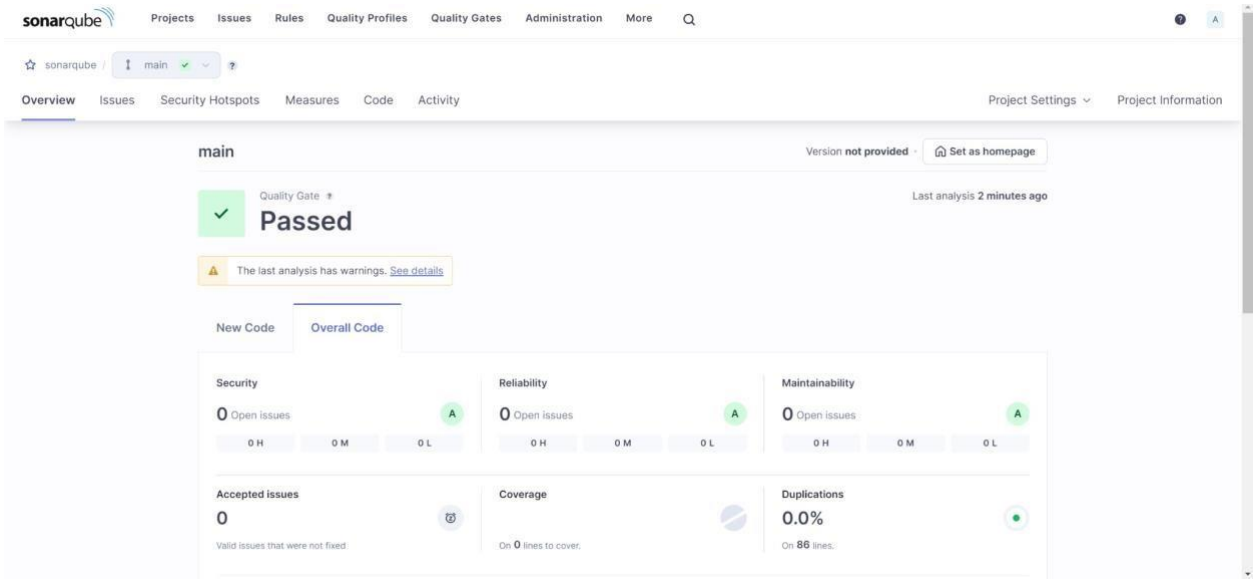
12. Run The **Build** and check the **console output**.





13. Once the build is complete, check the project in SonarQube.





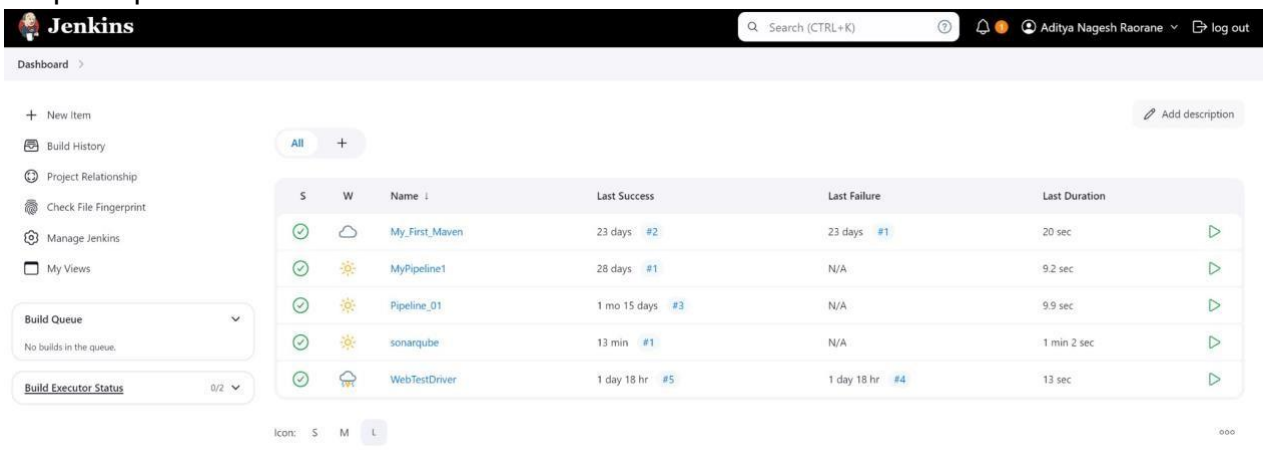
In this way, we have integrated Jenkins with SonarQube for SAST.

Conclusion:

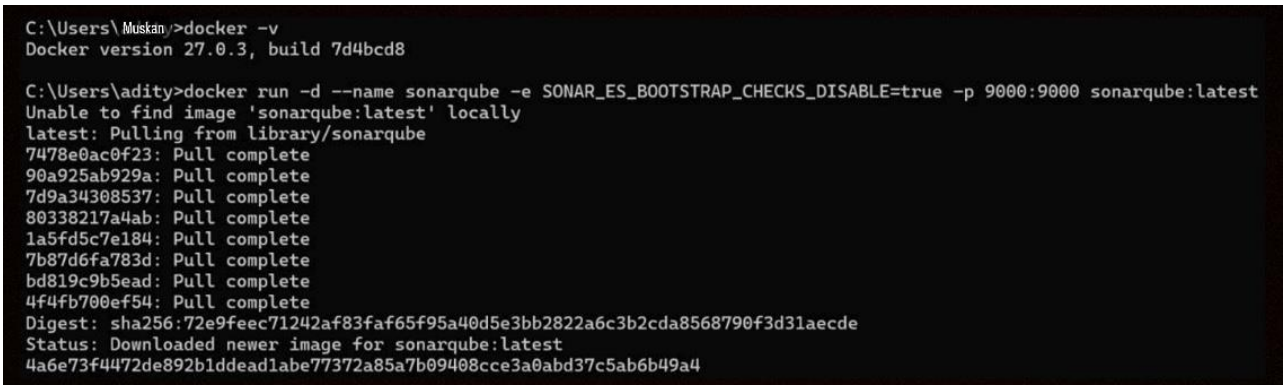
In this experiment, we have understood the importance of SAST and have successfully integrated Jenkins with SonarQube for Static Analysis and Code Testing.

Aim: Create a Jenkins CICD Pipeline with SonarQube / GitLab Integration to perform a static analysis of the code to detect bugs, code smells, and security vulnerabilities on a sample Web / Java / Python application.

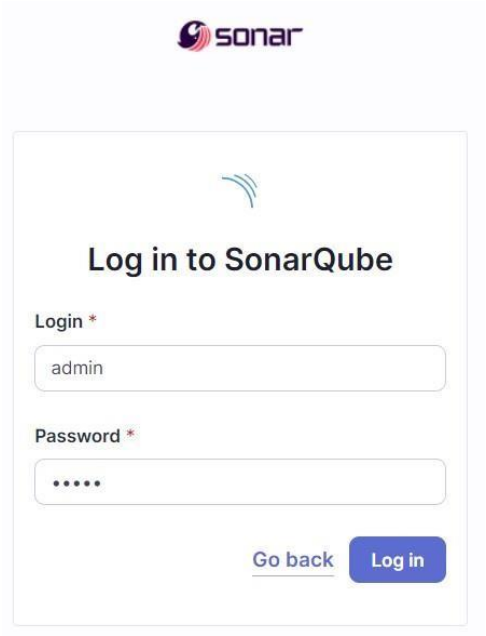
1. Open up Jenkins Dashboard on localhost:8080.



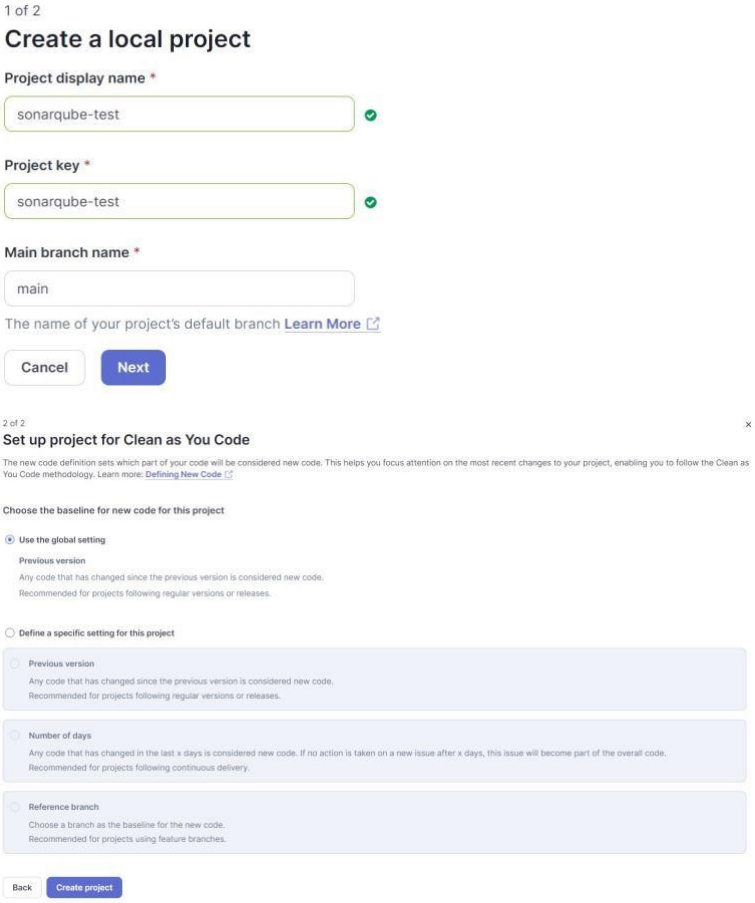
2. Run SonarQube in a Docker container using this command: a] docker -v b] docker pull sonarqube c] docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest



3. Once the container is up and running, you can check the status of SonarQube at **localhost port 9000**. The login id is “**admin**” and the password is “**mus12**”.

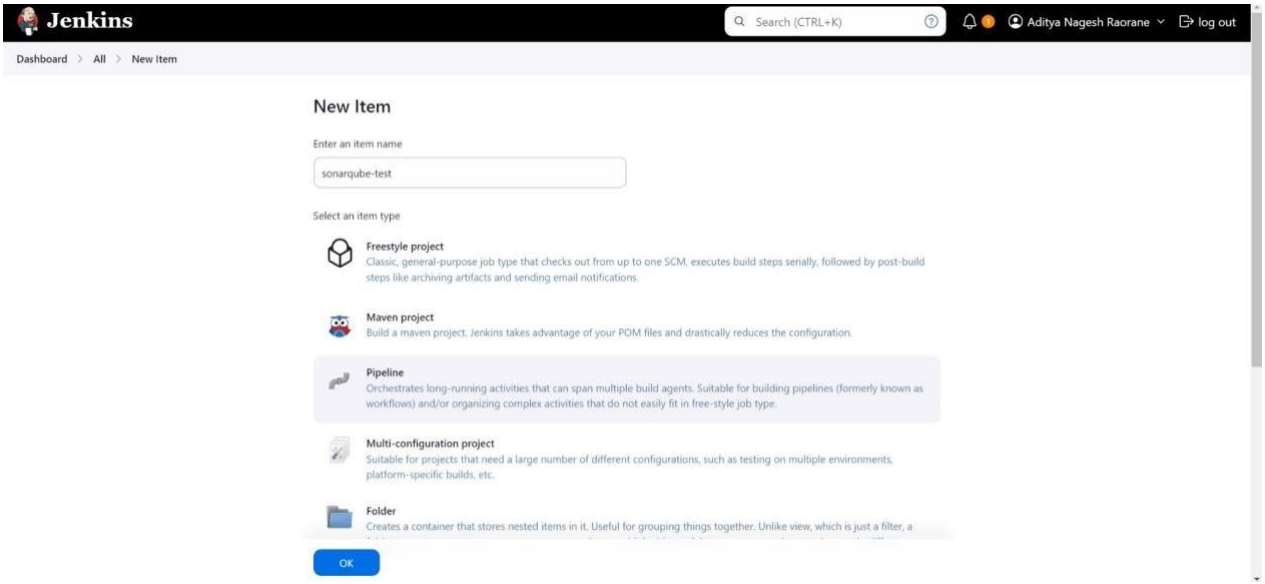


4. **Create a local project in SonarQube** with the name **sonarqube-test**.



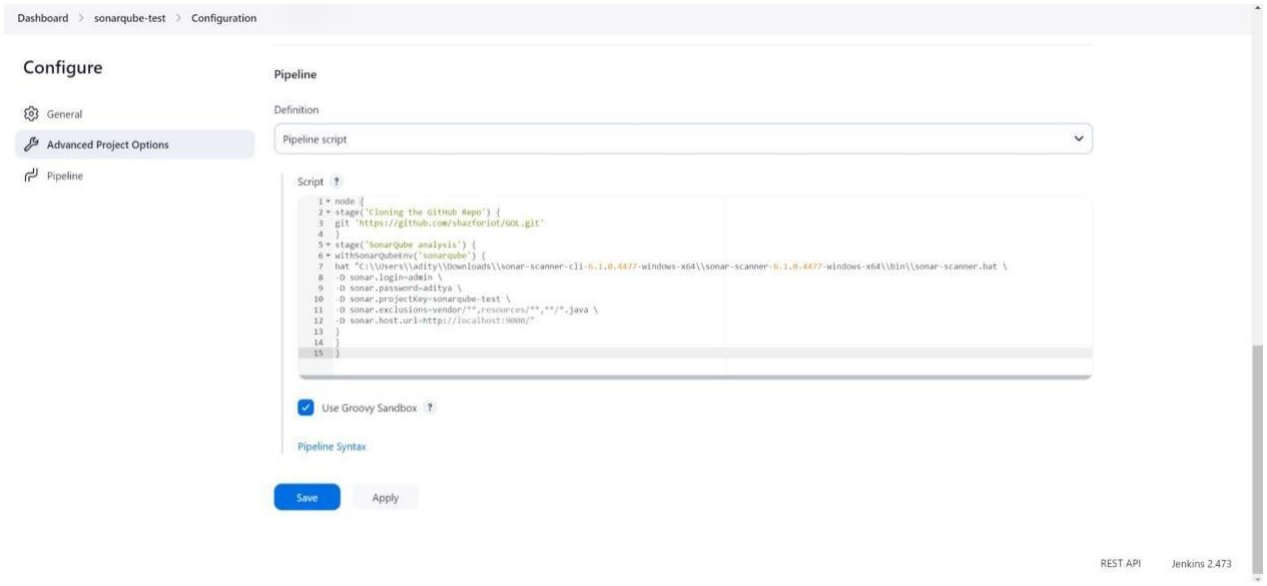
Setup the project and come back to Jenkins Dashboard.

6. Create a New Item in Jenkins, choose **Pipeline**.



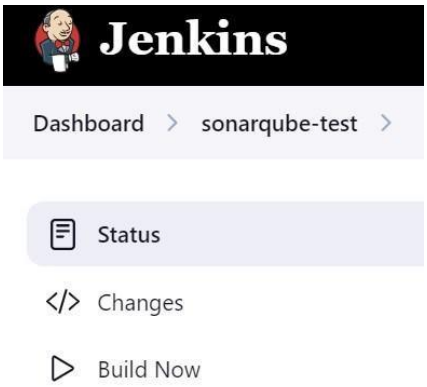
7. Under **Pipeline Script**, enter the following -

```
node { stage('Cloning the GitHub
        Repo')
        {
                git 'https://github.com/shazforiot/GOL.git'
        }
stage('SonarQube analysis') {
        withSonarQubeEnv('sonarqube') { bat
        "C:\\Users\\adity\\Downloads\\sonar-scanner-cli-6.1.0.4477-windows-x64\\sonar-s
        canner-6.1.0.4477-windows-x64\\bin\\sonar-scanner.bat \
                -D sonar.login=<YOUR ID> \
                -D sonar.password=<YOUR PASSWORD> \
                -D sonar.projectKey=<YOUR PROJECT KEY> \
                -D sonar.exclusions=vendor/**,resources/**,**/*.java \
                -D sonar.host.url=http://localhost:9000/"
                }
        }
}
```

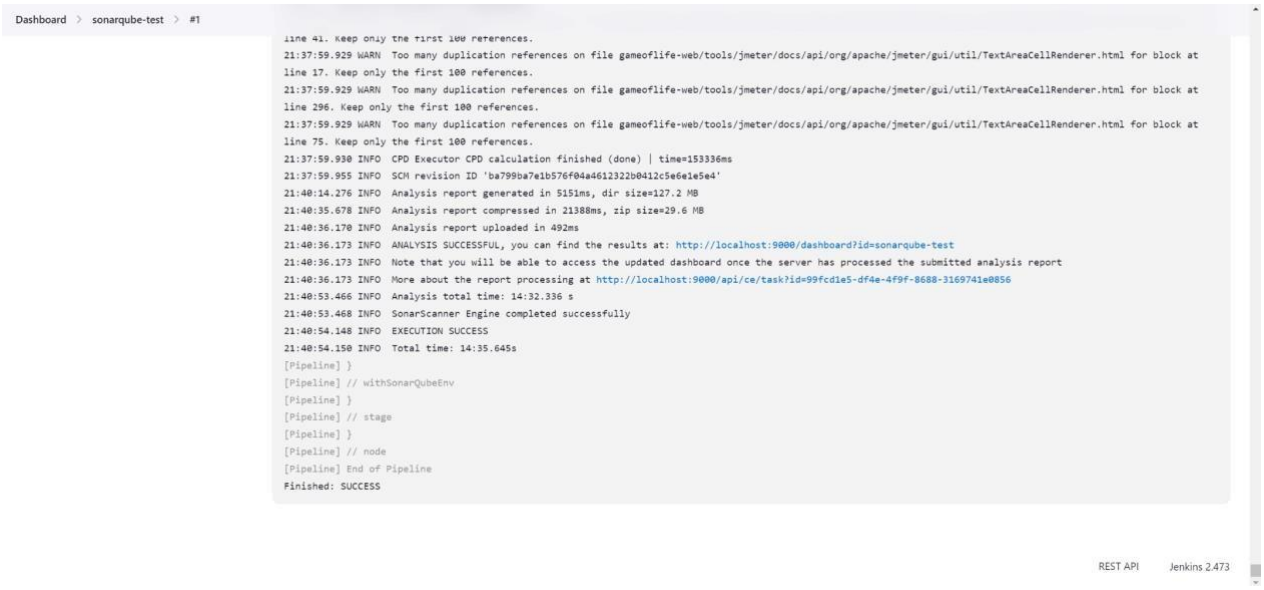


It is a java sample project which has a lot of repetitions and issues that will be detected by SonarQube.

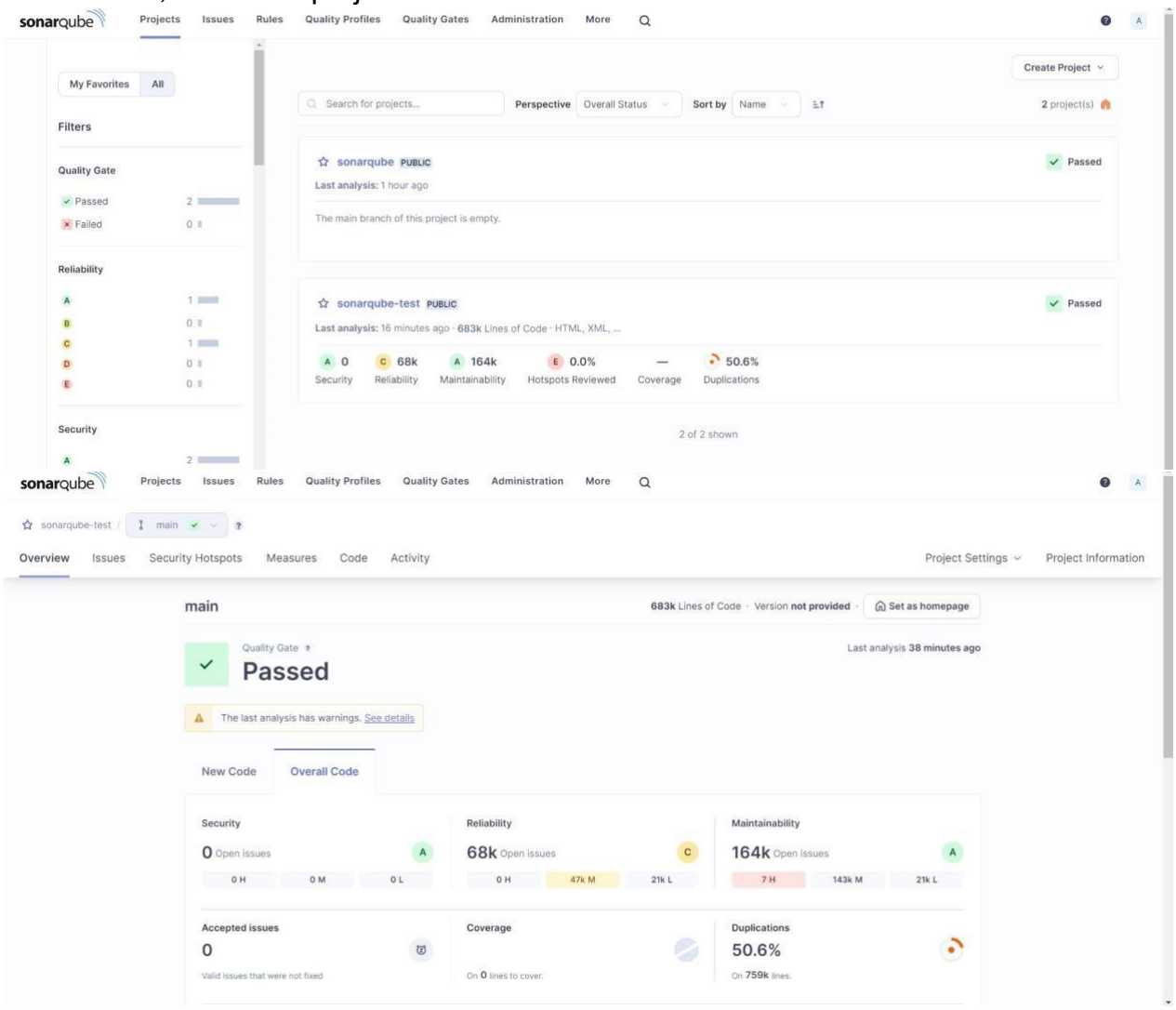
8. Run The Build.



9. Check the console output once the build is complete.

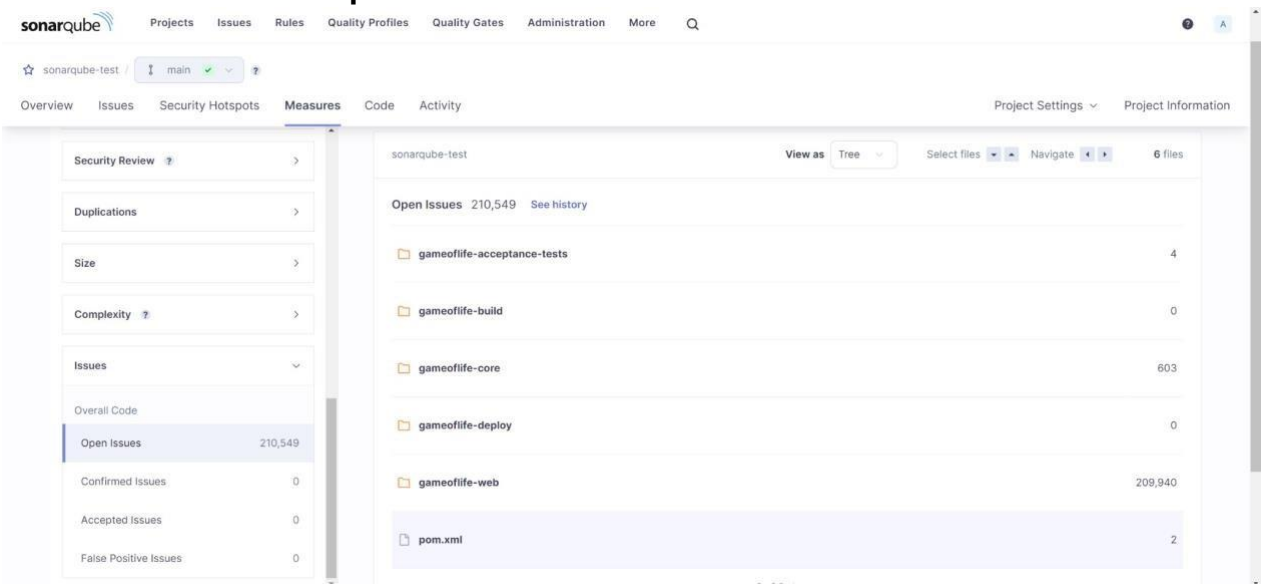


10. After that, check the project in SonarQube.

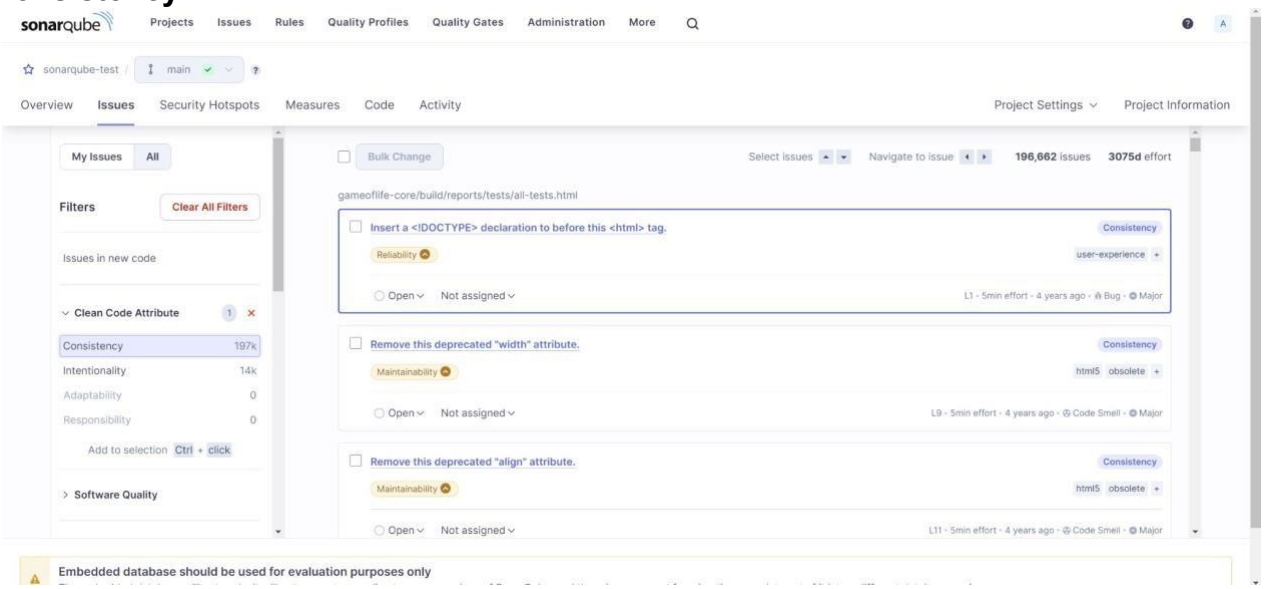


Under different tabs, check all different issues with the code.

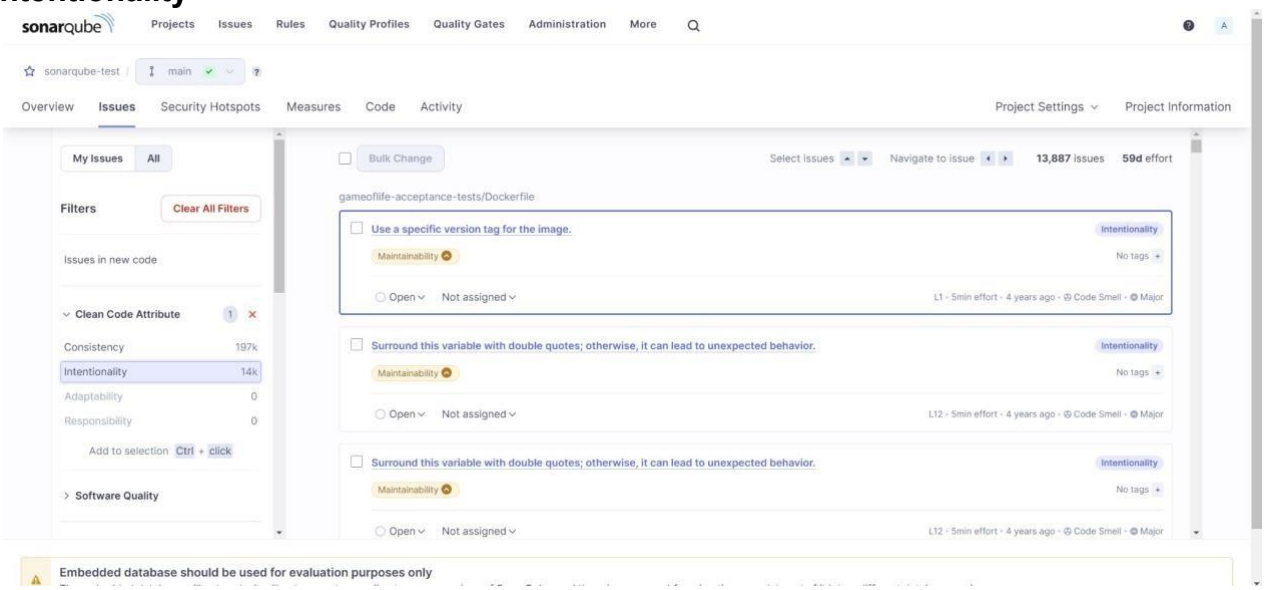
11. **Code Problems Open Issues**



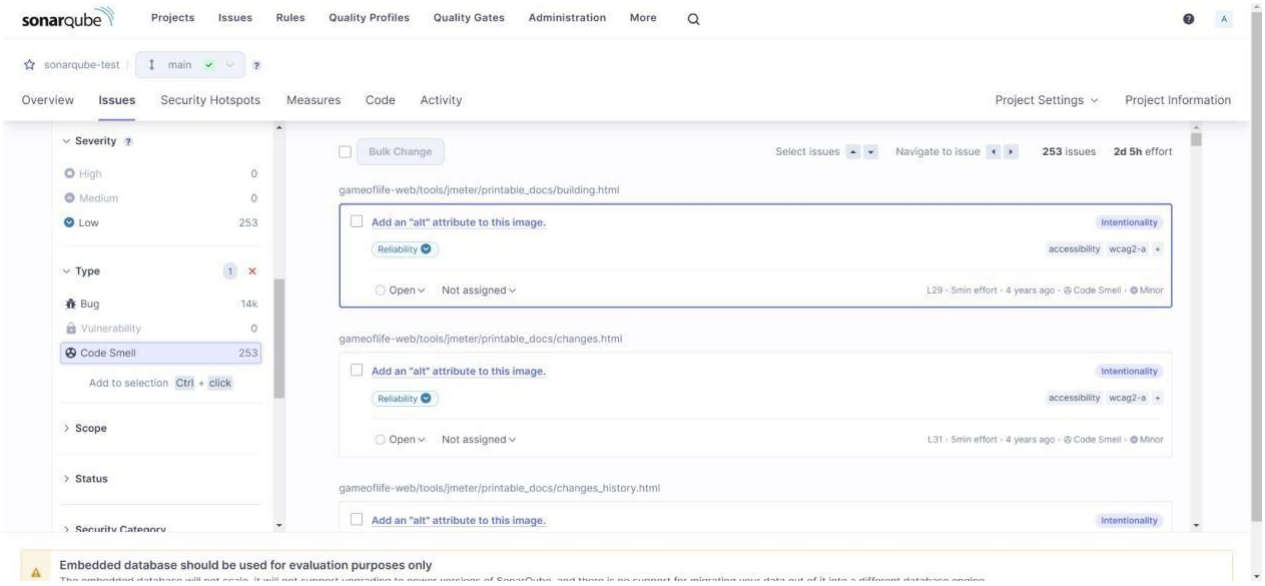
Consistency



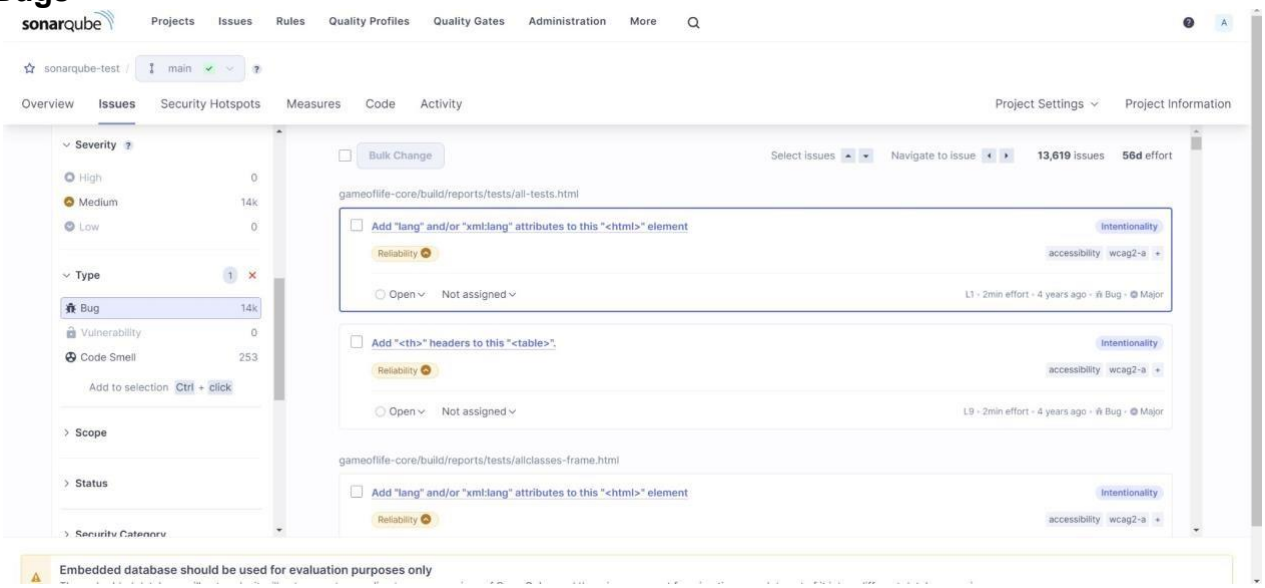
Intentionality



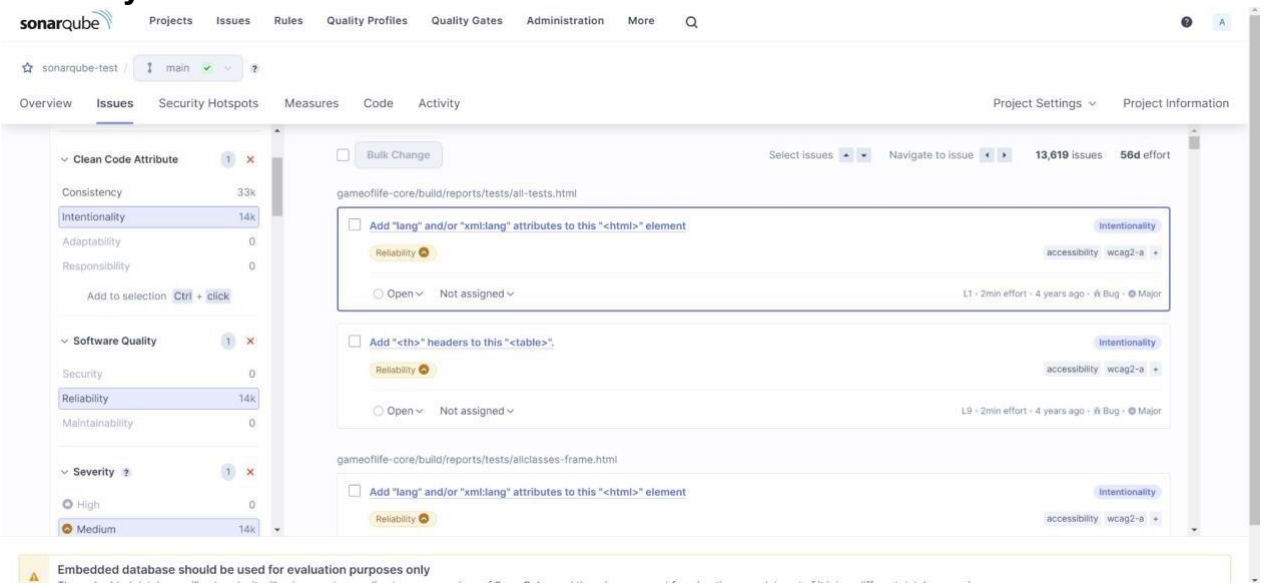
Code Smells



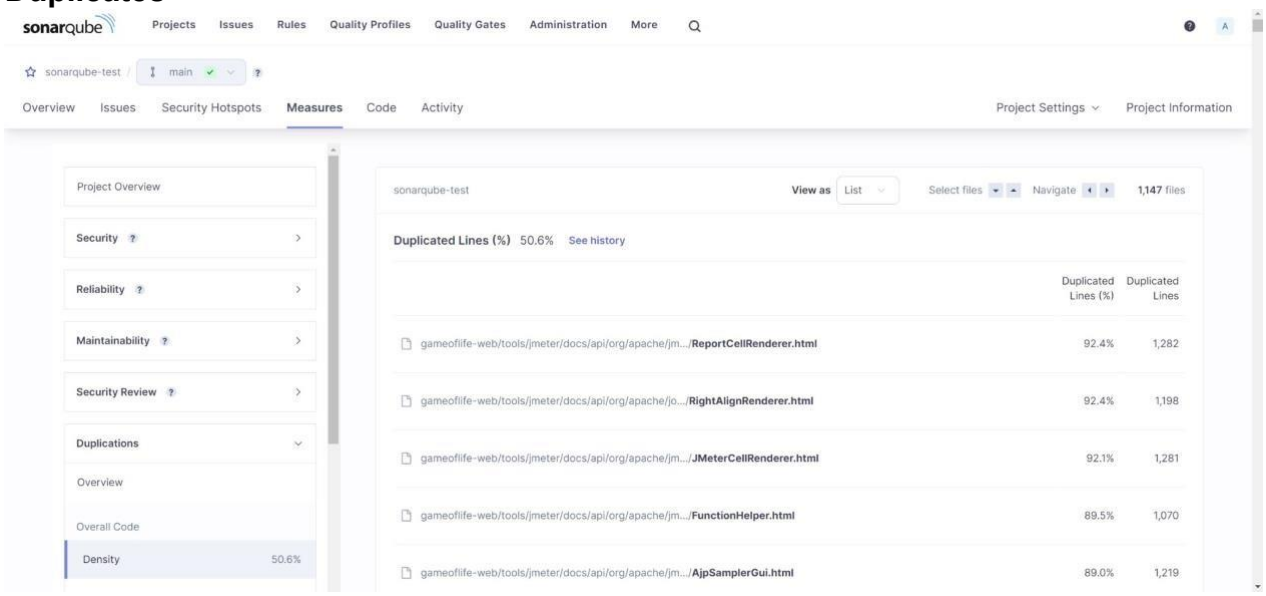
Bugs



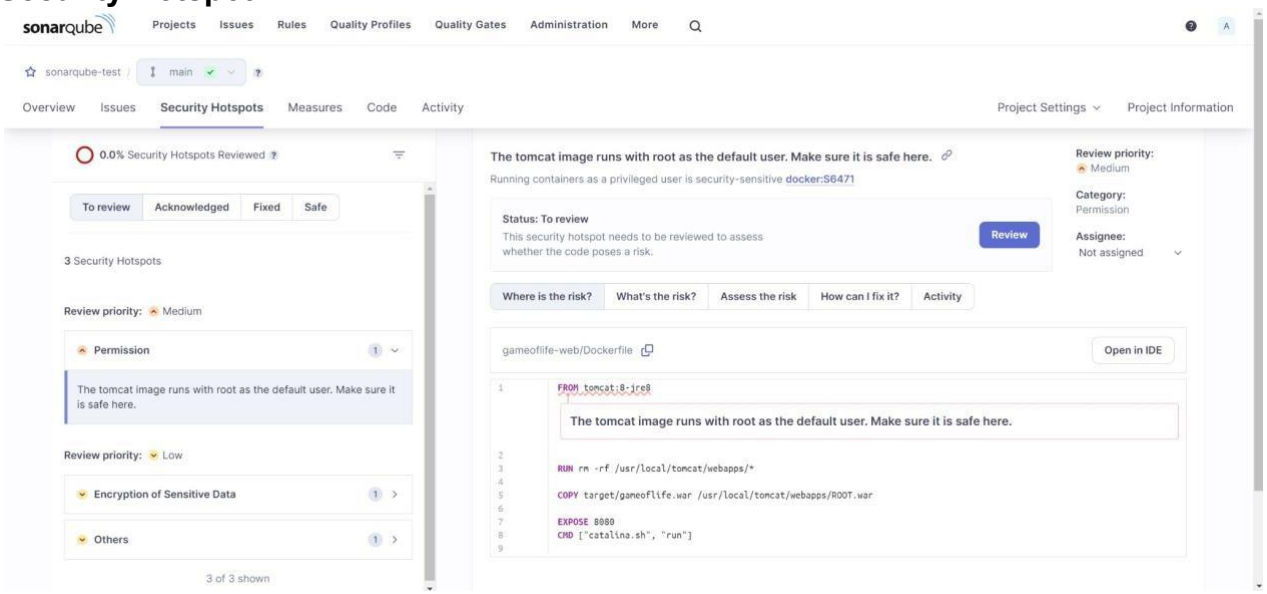
Reliability



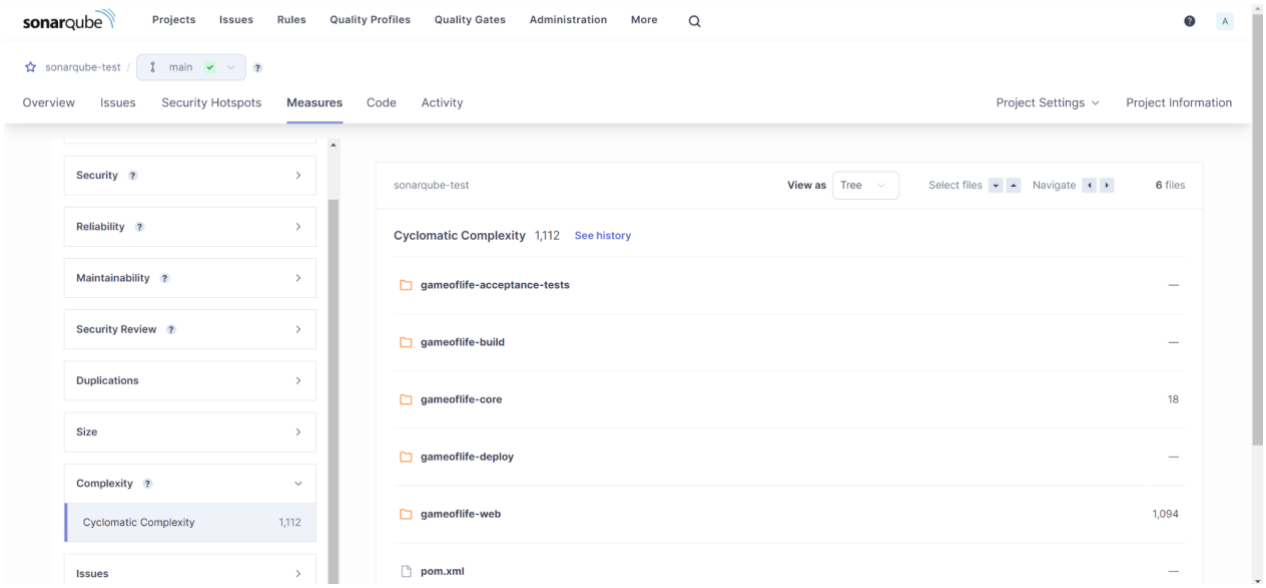
Duplicates



Security Hotspot



Cyclomatic Complexity



In this way, we have created a CI/CD Pipeline with Jenkins and integrated it with SonarQube to find issues in the code like bugs, code smells, duplicates, cyclomatic complexities, etc.

Conclusion:

In this experiment, we performed a static analysis of the code to detect bugs, code smells, and security vulnerabilities on our sample Java application.

Adv DevOps Practical 9

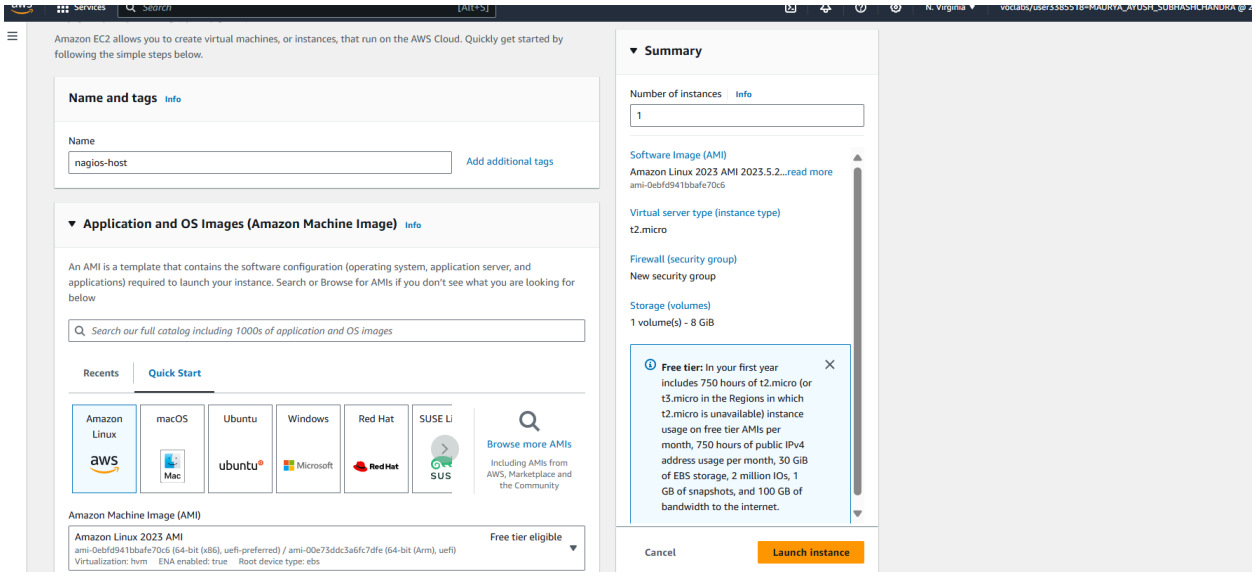
Aim: To Understand Continuous monitoring and Installation and configuration of Nagios Core, Nagios Plugins and NRPE (Nagios Remote Plugin Executor) on Linux Machine.

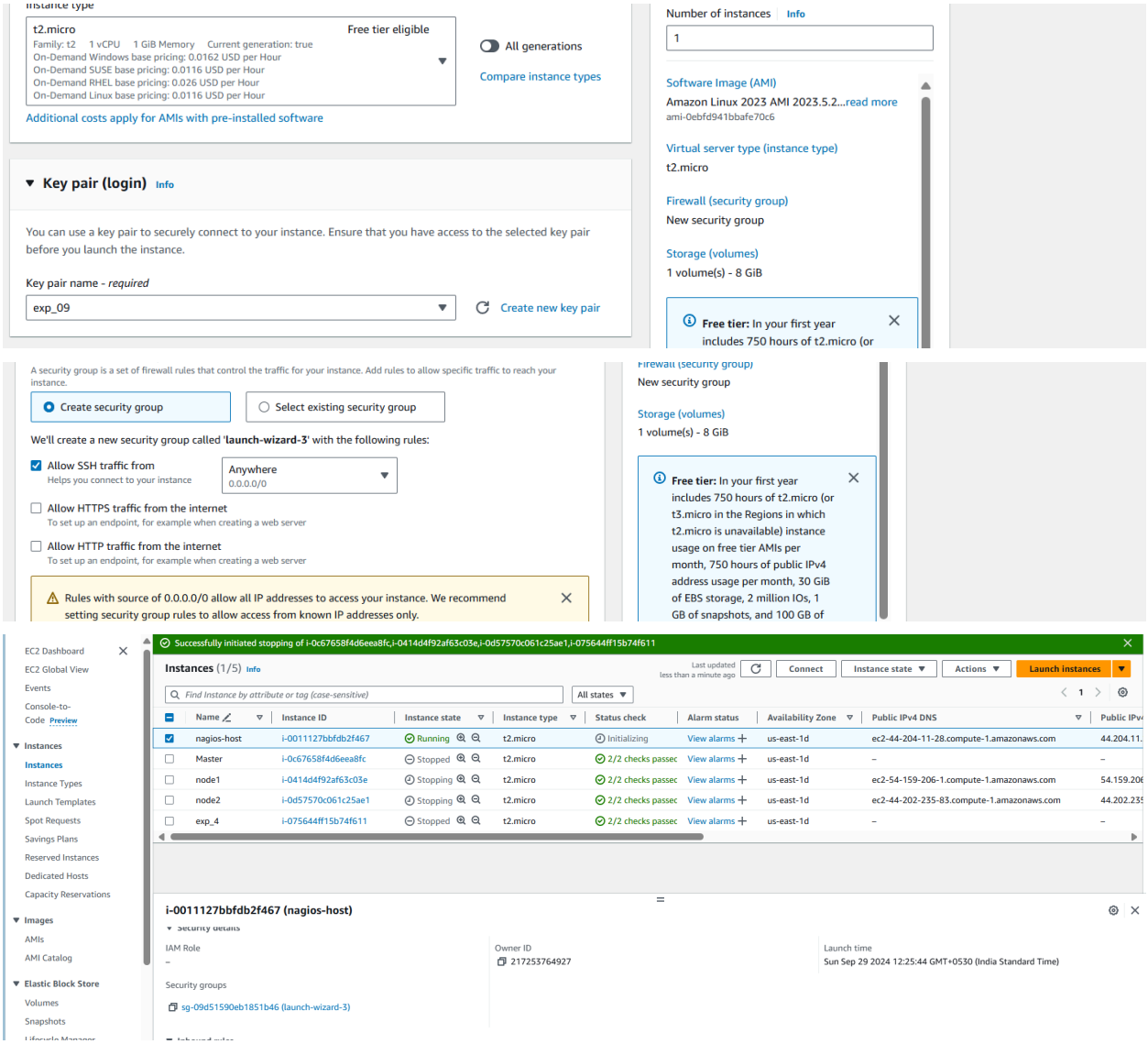
Theory:

What is Nagios?
Nagios is an open-source software for continuous monitoring of systems, networks, and infrastructures. It runs plugins stored on a server that is connected with a host or another server on your network or the Internet. In case of any failure, Nagios alerts about the issues so that the technical team can perform the recovery process immediately.
Nagios is used for continuous monitoring of systems, applications, service and business processes in a DevOps culture
.
Installation of Nagios

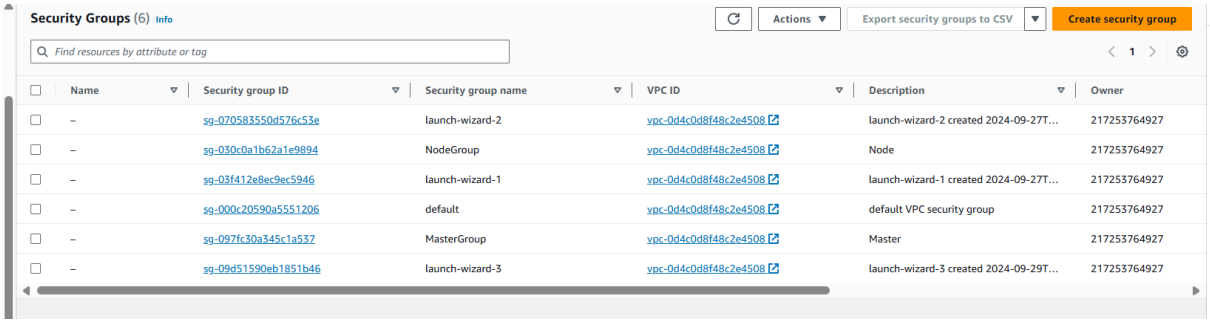
Prerequisites: AWS Free Tier

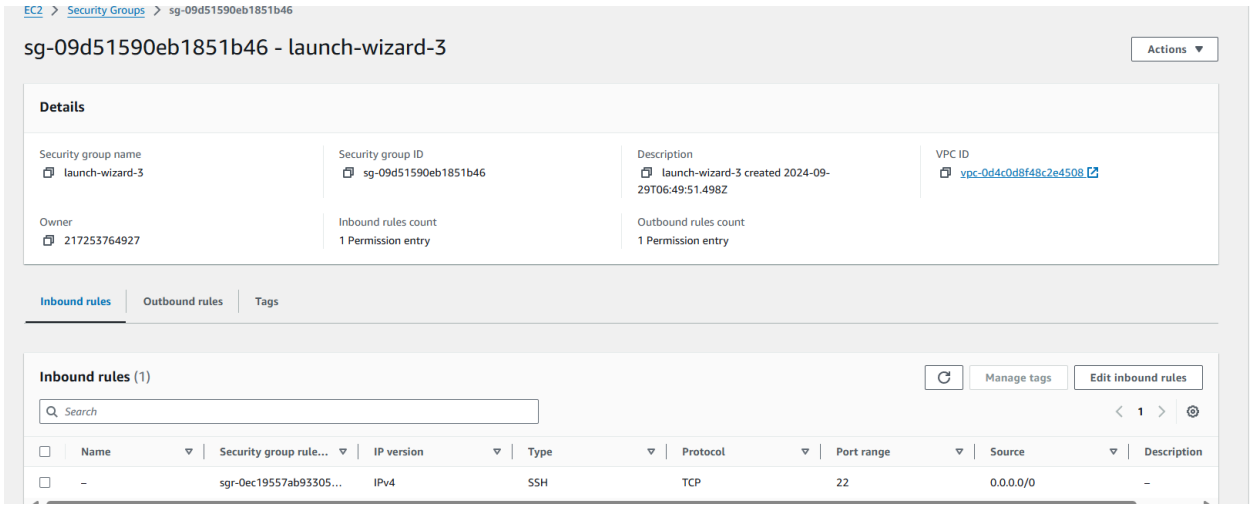
Steps:
1. Create an Amazon Linux EC2 Instance in AWS and name it - nagios-host

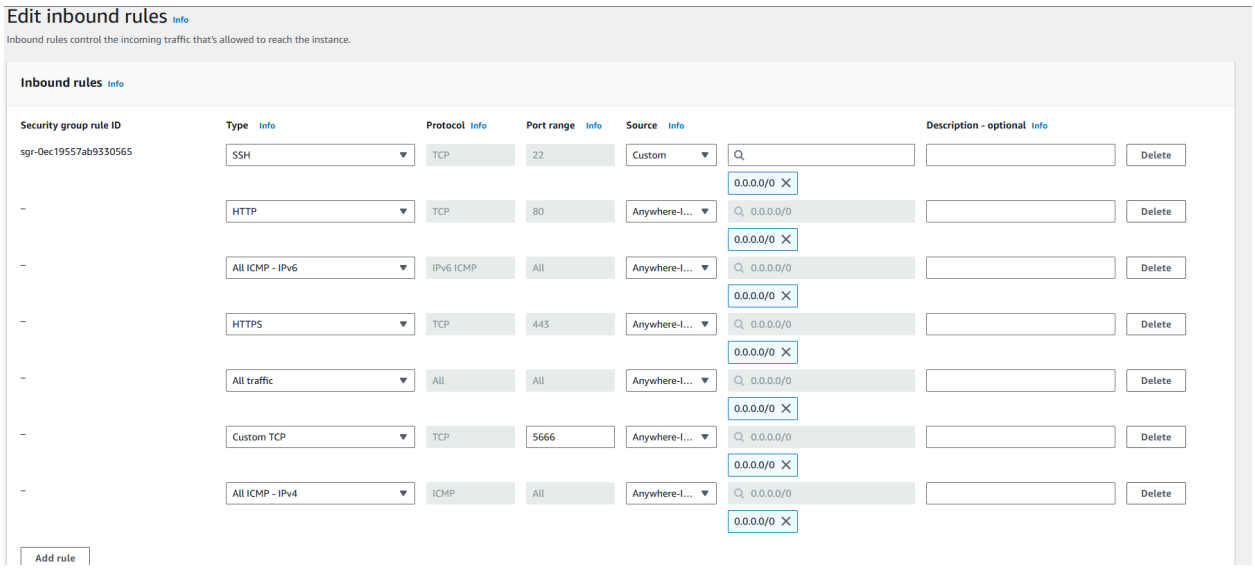


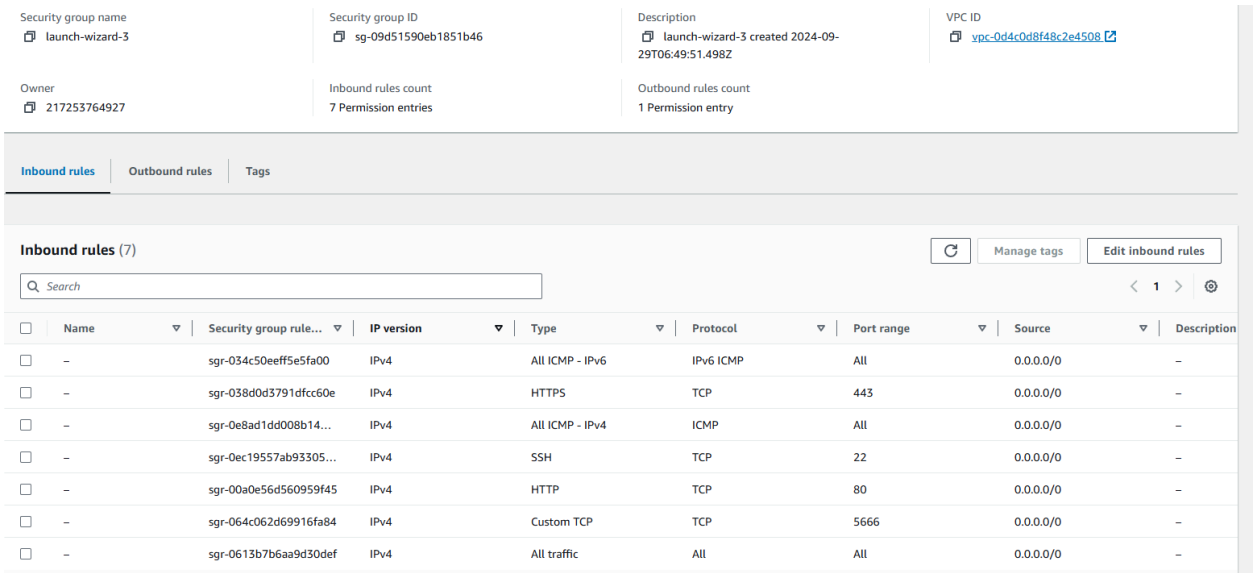


2. Under Security Group, make sure HTTP, HTTPS, SSH, ICMP are open from everywhere.



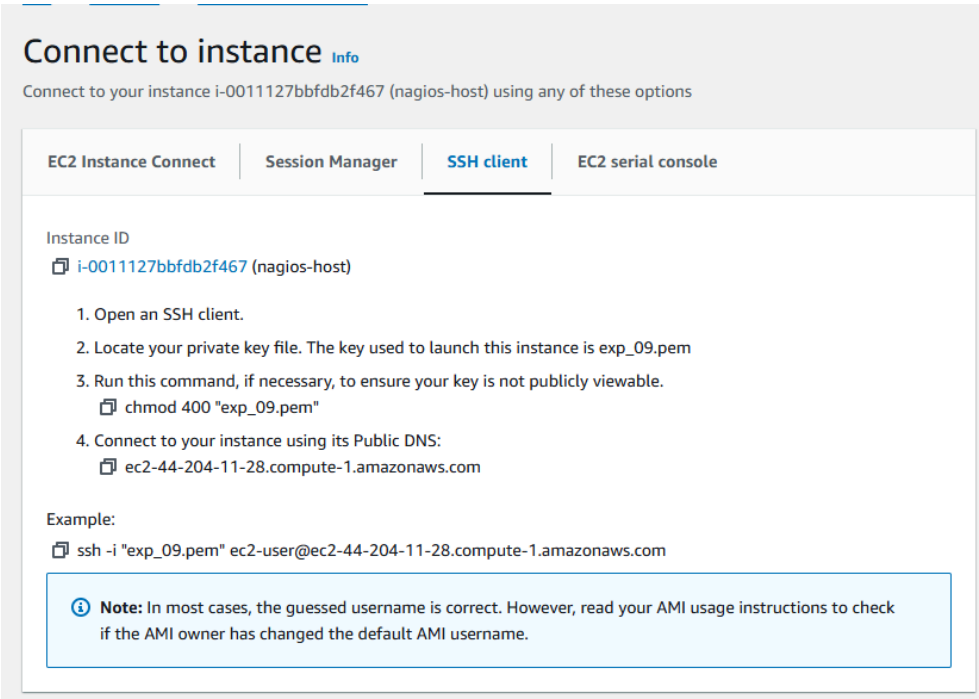






You have to edit the inbound rules of the specified Security Group for this.

3. SSH into Your EC2 instance or simply use EC2 Instance Connect from the browser.



Or open command prompt and paste ssh command.

```
Microsoft Windows [Version 10.0.22631.4169]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Ayush Maurya>ssh -i "Downloads/exp_09.pem" ec2-user@ec2-44-204-11-28.compute-1.amazonaws.com
The authenticity of host 'ec2-44-204-11-28.compute-1.amazonaws.com (44.204.11.28)' can't be established.
ED25519 key fingerprint is SHA256:v2OKH/ezl9iu7/RT6m8LWkgWzEJnnQIqrG9gKWzwC14.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-204-11-28.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
       #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Sun Sep 29 07:11:40 2024 from 18.206.107.27
[ec2-user@ip-172-31-91-91 ~]$
```

sudo yum update

```
[ec2-user@ip-172-31-91-91 ~]$
sudo yum update
Last metadata expiration check: 0:19:03 ago on Sun Sep 29 06:56:15 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-91-91 ~]$
```

sudo yum install httpd php

```
[ec2-user@ip-172-31-91-91 ~]$ sudo yum install httpd php
Last metadata expiration check: 0:19:29 ago on Sun Sep 29 06:56:15 2024.
Dependencies resolved.
================================================================================
 Package                 Architecture  Version                    Repository  Size
================================================================================
Installing:
 httpd                   x86_64        2.4.62-1.amzn2023          amazonlinux  48 k
 php8.3                  x86_64        8.3.10-1.amzn2023.0.1      amazonlinux  10 k
Installing dependencies:
 apr                     x86_64        1.7.2-2.amzn2023.0.2       amazonlinux 129 k
 apr-util                x86_64        1.6.3-1.amzn2023.0.1       amazonlinux  98 k
 generic-logos-httpd     noarch        18.0.0-12.amzn2023.0.3     amazonlinux  19 k
 httpd-core              x86_64        2.4.62-1.amzn2023          amazonlinux 1.4 M
 httpd-filesystem        noarch        2.4.62-1.amzn2023          amazonlinux  14 k
 httpd-tools             x86_64        2.4.62-1.amzn2023          amazonlinux  81 k
 libbrotli               x86_64        1.0.9-4.amzn2023.0.2       amazonlinux 315 k
 libsodium               x86_64        1.0.19-4.amzn2023          amazonlinux 176 k
 libxslt                 x86_64        1.1.34-5.amzn2023.0.2      amazonlinux 241 k
 mailcap                 noarch        2.1.49-3.amzn2023.0.3      amazonlinux  33 k
 nginx-filesystem        noarch        1:1.24.0-1.amzn2023.0.4    amazonlinux 9.8 k
 php8.3-cli              x86_64        8.3.10-1.amzn2023.0.1      amazonlinux 3.7 M
 php8.3-common           x86_64        8.3.10-1.amzn2023.0.1      amazonlinux 737 k
 php8.3-process          x86_64        8.3.10-1.amzn2023.0.1      amazonlinux  45 k
 php8.3-xml              x86_64        8.3.10-1.amzn2023.0.1      amazonlinux 154 k
Installing weak dependencies:
 apr-util-openssl        x86_64        1.6.3-1.amzn2023.0.1       amazonlinux  17 k
 mod_http2               x86_64        2.0.27-1.amzn2023.0.3      amazonlinux 166 k
 mod_lua                 x86_64        2.4.62-1.amzn2023          amazonlinux  61 k
 php8.3-fpm              x86_64        8.3.10-1.amzn2023.0.1      amazonlinux 1.9 M
 php8.3-mbstring         x86_64        8.3.10-1.amzn2023.0.1      amazonlinux 528 k
 php8.3-opcache          x86_64        8.3.10-1.amzn2023.0.1      amazonlinux 379 k
 php8.3-pdo              x86_64        8.3.10-1.amzn2023.0.1      amazonlinux  89 k
 php8.3-sodium           x86_64        8.3.10-1.amzn2023.0.1      amazonlinux  41 k

Transaction Summary
```

```
Total                                                          22 MB/s |  10 MB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                              1/1
  Installing       : php8.3-common-8.3.10-1.amzn2023.0.1.x86_64                   1/25
  Installing       : apr-1.7.2-2.amzn2023.0.2.x86_64                              2/25
  Installing       : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64                 3/25
  Installing       : apr-util-1.6.3-1.amzn2023.0.1.x86_64                         4/25
  Installing       : mailcap-2.1.49-3.amzn2023.0.3.noarch                         5/25
  Running scriptlet: httpd-filesystem-2.4.62-1.amzn2023.noarch                    6/25
```

sudo yum install gcc glibc glibc-common

```
[ec2-user@ip-172-31-91-91 ~]$ sudo yum install gcc glibc glibc-common
Last metadata expiration check: 0:20:41 ago on Sun Sep 29 06:56:15 2024.
Package glibc-2.34-52.amzn2023.0.11.x86_64 is already installed.
Package glibc-common-2.34-52.amzn2023.0.11.x86_64 is already installed.
Dependencies resolved.
================================================================================
 Package                 Architecture  Version                    Repository  Size
================================================================================
Installing:
 gcc                     x86_64        11.4.1-2.amzn2023.0.2      amazonlinux  32 M
Installing dependencies:
 annobin-docs            noarch        10.93-1.amzn2023.0.1       amazonlinux  92 k
 annobin-plugin-gcc      x86_64        10.93-1.amzn2023.0.1       amazonlinux 887 k
 cpp                     x86_64        11.4.1-2.amzn2023.0.2      amazonlinux  10 M
 gc                      x86_64        8.0.4-5.amzn2023.0.2       amazonlinux 105 k
 glibc-devel             x86_64        2.34-52.amzn2023.0.11      amazonlinux  27 k
 glibc-headers-x86       noarch        2.34-52.amzn2023.0.11      amazonlinux 427 k
 guile22                 x86_64        2.2.7-2.amzn2023.0.3       amazonlinux 6.4 M
 kernel-headers          x86_64        6.1.109-118.189.amzn2023   amazonlinux 1.4 M
 libmpc                  x86_64        1.2.1-2.amzn2023.0.2       amazonlinux  62 k
 libtool-ltdl            x86_64        2.4.7-1.amzn2023.0.3       amazonlinux  38 k
 libxcrypt-devel         x86_64        4.4.33-7.amzn2023          amazonlinux  32 k
 make                    x86_64        1:4.3-5.amzn2023.0.2       amazonlinux 534 k

Transaction Summary
================================================================================
Install  13 Packages

Total download size: 52 M
```

```
Installed:
  annobin-docs-10.93-1.amzn2023.0.1.noarch
  annobin-plugin-gcc-10.93-1.amzn2023.0.1.x86_64
  cpp-11.4.1-2.amzn2023.0.2.x86_64
  gc-8.0.4-5.amzn2023.0.2.x86_64
  gcc-11.4.1-2.amzn2023.0.2.x86_64
  glibc-devel-2.34-52.amzn2023.0.11.x86_64
  glibc-headers-x86-2.34-52.amzn2023.0.11.noarch
  guile22-2.2.7-2.amzn2023.0.3.x86_64
  kernel-headers-6.1.109-118.189.amzn2023.x86_64
  libmpc-1.2.1-2.amzn2023.0.2.x86_64
  libtool-ltdl-2.4.7-1.amzn2023.0.3.x86_64
  libxcrypt-devel-4.4.33-7.amzn2023.x86_64
  make-1:4.3-5.amzn2023.0.2.x86_64

Complete!
```

sudo yum install gd gd-devel

```
[ec2-user@ip-172-31-91-91 ~]$ sudo yum install gd gd-devel
Last metadata expiration check: 0:21:30 ago on Sun Sep 29 06:56:15 2024.
Dependencies resolved.
================================================================================
 Package                 Architecture  Version                    Repository  Size
================================================================================
Installing:
 gd                      x86_64        2.3.3-5.amzn2023.0.3       amazonlinux 139 k
 gd-devel                x86_64        2.3.3-5.amzn2023.0.3       amazonlinux  38 k
Installing dependencies:
 brotli                  x86_64        1.0.9-4.amzn2023.0.2       amazonlinux 314 k
 brotli-devel            x86_64        1.0.9-4.amzn2023.0.2       amazonlinux  31 k
 bzip2-devel             x86_64        1.0.8-6.amzn2023.0.2       amazonlinux 214 k
 cairo                   x86_64        1.17.6-2.amzn2023.0.1      amazonlinux 684 k
 cmake-filesystem        x86_64        3.22.2-1.amzn2023.0.4      amazonlinux  16 k
 fontconfig              x86_64        2.13.94-2.amzn2023.0.2     amazonlinux 273 k
```

```
Installed:
  brotli-1.0.9-4.amzn2023.0.2.x86_64                  brotli-devel-1.0.9-4.amzn2023.0.2.x86_64                  bzip2-devel-1.0.8-6.amzn2023.0.2.x86_64
  cairo-1.17.6-2.amzn2023.0.1.x86_64                  cmake-filesystem-3.22.2-1.amzn2023.0.4.x86_64             fontconfig-2.13.94-2.amzn2023.0.2.x86_64
  fontconfig-devel-2.13.94-2.amzn2023.0.2.x86_64      fonts-filesystem-1:2.0.5-12.amzn2023.0.2.noarch           freetype-2.13.2-5.amzn2023.0.1.x86_64
  freetype-devel-2.13.2-5.amzn2023.0.1.x86_64         gd-2.3.3-5.amzn2023.0.3.x86_64                            gd-devel-2.3.3-5.amzn2023.0.3.x86_64
  glib2-devel-2.74.7-689.amzn2023.0.2.x86_64          google-noto-fonts-common-20201206-2.amzn2023.0.2.noarch   google-noto-sans-vf-fonts-20201206-2.amzn2023.0.2.noarch
  graphite2-1.3.14-7.amzn2023.0.2.x86_64              graphite2-devel-1.3.14-7.amzn2023.0.2.x86_64              harfbuzz-7.0.0-2.amzn2023.0.1.x86_64
  harfbuzz-devel-7.0.0-2.amzn2023.0.1.x86_64          harfbuzz-icu-7.0.0-2.amzn2023.0.1.x86_64                  jbigkit-libs-2.1-21.amzn2023.0.2.x86_64
  langpacks-core-font-en-3.0-21.amzn2023.0.4.noarch   libICE-1.0.10-6.amzn2023.0.2.x86_64                       libSM-1.2.3-8.amzn2023.0.2.x86_64
  libX11-1.7.2-3.amzn2023.0.4.x86_64                  libX11-common-1.7.2-3.amzn2023.0.4.noarch                 libX11-devel-1.7.2-3.amzn2023.0.4.x86_64
  libX11-xcb-1.7.2-3.amzn2023.0.4.x86_64              libXau-1.0.9-6.amzn2023.0.2.x86_64                        libXau-devel-1.0.9-6.amzn2023.0.2.x86_64
  libXext-1.3.4-6.amzn2023.0.2.x86_64                 libXpm-3.5.15-2.amzn2023.0.3.x86_64                       libXpm-devel-3.5.15-2.amzn2023.0.3.x86_64
  libXrender-0.9.10-14.amzn2023.0.2.x86_64            libXt-1.2.0-4.amzn2023.0.2.x86_64                         libblkid-devel-2.37.4-1.amzn2023.0.4.x86_64
  libffi-devel-3.4.4-1.amzn2023.0.1.x86_64            libicu-67.1-7.amzn2023.0.3.x86_64                         libicu-devel-67.1-7.amzn2023.0.3.x86_64
  libjpeg-turbo-2.1.4-2.amzn2023.0.5.x86_64           libjpeg-turbo-devel-2.1.4-2.amzn2023.0.5.x86_64           libmount-devel-2.37.4-1.amzn2023.0.4.x86_64
  libpng-2:1.6.37-10.amzn2023.0.6.x86_64              libpng-devel-2:1.6.37-10.amzn2023.0.6.x86_64              libselinux-devel-3.4-5.amzn2023.0.2.x86_64
  libsepol-devel-3.4-3.amzn2023.0.3.x86_64            libtiff-4.4.0-4.amzn2023.0.18.x86_64                      libtiff-devel-4.4.0-4.amzn2023.0.18.x86_64
  libwebp-1.2.4-1.amzn2023.0.6.x86_64                 libwebp-devel-1.2.4-1.amzn2023.0.6.x86_64                 libxcb-1.13.1-7.amzn2023.0.2.x86_64
  libxcb-devel-1.13.1-7.amzn2023.0.2.x86_64           libxml2-devel-2.10.4-1.amzn2023.0.6.x86_64                pcre2-devel-10.40-1.amzn2023.0.3.x86_64
  pcre2-utf16-10.40-1.amzn2023.0.3.x86_64             pcre2-utf32-10.40-1.amzn2023.0.3.x86_64                   pixman-0.40.0-3.amzn2023.0.3.x86_64
  sysprof-capture-devel-3.40.1-2.amzn2023.0.2.x86_64  xml-common-0.6.3-56.amzn2023.0.2.noarch                   xorg-x11-proto-devel-2021.4-1.amzn2023.0.2.noarch
  xz-devel-5.2.5-9.amzn2023.0.2.x86_64                zlib-devel-1.2.11-33.amzn2023.0.5.x86_64

Complete!
```

5. Create a new Nagios User with its password. You'll have to enter the password twice for confirmation.

sudo adduser -m nagios

sudo passwd nagios

(password : *ayushmau)*

```
Complete!
[ec2-user@ip-172-31-91-91 ~]$ sudo adduser -m nagios
sudo passwd nagios
Changing password for user nagios.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
Sorry, passwords do not match.
New password:
BAD PASSWORD: The password contains the user name in some form
Retype new password:
Sorry, passwords do not match.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-91-91 ~]$
```

6. Create a new user group

sudo groupadd nagcmd

```
[ec2-user@ip-172-31-91-91 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-91-91 ~]$
```

7. Use these commands so that you don't have to use sudo for Apache and Nagios

sudo usermod -a -G nagcmd nagios

sudo usermod -a -G nagcmd apache

```
[ec2-user@ip-172-31-91-91 ~]$ sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
[ec2-user@ip-172-31-91-91 ~]$
```

8. Create a new directory for Nagios downloads

mkdir ~/downloads

cd ~/downloads

```
[ec2-user@ip-172-31-91-91 ~]$ mkdir ~/downloads
cd ~/downloads
```

9. Use wget to download the source zip files.

wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz

```
[ec2-user@ip-172-31-91-91 downloads]$ cd ..
[ec2-user@ip-172-31-91-91 ~]$ cd ~/downloads
[ec2-user@ip-172-31-91-91 downloads]$ wget https://assets.nagios.com/downloa
ds/nagioscore/releases/nagios-4.5.5.tar.gz
--2024-09-29 09:11:59--  https://assets.nagios.com/downloads/nagioscore/rele
ases/nagios-4.5.5.tar.gz
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 2600:3c00::
f03c:92ff:fef7:45ce
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... con
nected.
HTTP request sent, awaiting response... 200 OK
Length: 2065473 (2.0M) [application/x-gzip]
Saving to: ‘nagios-4.5.5.tar.gz’

nagios-4.5.5.tar.g 100%[=================>]   1.97M  5.07MB/s    in 0.4s

2024-09-29 09:11:59 (5.07 MB/s) - ‘nagios-4.5.5.tar.gz’ saved [2065473/20654
73]

[ec2-user@ip-172-31-91-91 downloads]$
```

wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz

```
[ec2-user@ip-172-31-91-91 nagios-4.5.5]$ cd ..
[ec2-user@ip-172-31-91-91 downloads]$ wget https://nagios-plugins.org/downlo
ad/nagios-plugins-2.4.11.tar.gz
--2024-09-29 09:14:28--  https://nagios-plugins.org/download/nagios-plugins-
2.4.11.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251
Connecting to nagios-plugins.org (nagios-plugins.org)|45.56.123.251|:443...
connected.
HTTP request sent, awaiting response... 200 OK
Length: 2753049 (2.6M) [application/x-gzip]
Saving to: ‘nagios-plugins-2.4.11.tar.gz’

nagios-plugins-2.4 100%[=================>]   2.62M  6.92MB/s    in 0.4s
```

10. Use tar to unzip and change to that directory.
tar zxvf nagios-4.5.5.tar.gz

```
[ec2-user@ip-172-31-91-91 downloads]$ tar zxvf nagios-4.0.8.tar.gz
nagios-4.0.8/
nagios-4.0.8/.gitignore
nagios-4.0.8/Changelog
nagios-4.0.8/INSTALLING
nagios-4.0.8/LEGAL
nagios-4.0.8/LICENSE
nagios-4.0.8/Makefile.in
nagios-4.0.8/README
nagios-4.0.8/README.asciidoc
nagios-4.0.8/THANKS
nagios-4.0.8/UPGRADING
nagios-4.0.8/base/
nagios-4.0.8/base/.gitignore
```

11. Run the configuration script with the same group name you previously created.
./configure --with-command-group=nagcmd

Here we go an error

```
[ec2-user@ip-172-31-91-91 downloads]$ ./configure --with-command-group=nagcmd
-bash: ./configure: No such file or directory
[ec2-user@ip-172-31-91-91 downloads]$
```

Solution

Navigate to nagios folder in downloads

```
[ec2-user@ip-172-31-91-91 downloads]$ ls
nagios-4.0.8  nagios-4.0.8.tar.gz  nagios-plugins-2.0.3.tar.gz
[ec2-user@ip-172-31-91-91 downloads]$ cd nagios-4.0.8
[ec2-user@ip-172-31-91-91 nagios-4.0.8]$
```

Error 2: Cannot find SSL headers.
Solution: Install openssl dev library

Steps:
sudo yum install openssl-devel

```
[ec2-user@ip-172-31-91-91 nagios-4.5.5]$ sudo yum install openssl-devel
Last metadata expiration check: 2:24:05 ago on Sun Sep 29 06:56:15 2024.
Dependencies resolved.
================================================================================
 Package          Arch      Version                     Repository      Size
================================================================================
Installing:
 openssl-devel    x86_64    1:3.0.8-1.amzn2023.0.14     amazonlinux    3.0 M

Transaction Summary
================================================================================
Install  1 Package

Total download size: 3.0 M
Installed size: 4.7 M
Is this ok [y/N]: y
Downloading Packages:
```

Now run
./configure --with-command-group=nagcmd

```
             Event Broker:  yes
         Install ${prefix}:  /usr/local/nagios
     Install ${includedir}:  /usr/local/nagios/include/nagios
                 Lock file:  /run/nagios.lock
    Check result directory:  /usr/local/nagios/var/spool/checkresults
            Init directory:  /lib/systemd/system
   Apache conf.d directory:  /etc/httpd/conf.d
              Mail program:  /bin/mail
                   Host OS:  linux-gnu
          IOBroker Method:  epoll

 Web Interface Options:
 ------------------------
                 HTML URL:  http://localhost/nagios/
                  CGI URL:  http://localhost/nagios/cgi-bin/
 Traceroute (used by WAP):  /usr/bin/traceroute


Review the options above for accuracy.  If they look okay,
type 'make all' to compile the main program and CGIs.

[ec2-user@ip-172-31-91-91 nagios-4.5.5]$
```

12. Compile the source code.
make all

```
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE_
CONFIG_H -DNSCORE -c -o nagios.o ./nagios.c
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE_
CONFIG_H -DNSCORE -c -o broker.o broker.c
```

13. Install binaries, init script and sample config files. Lastly, set permissions on the external command directory.
sudo make install
sudo make install-init
sudo make install-config
sudo make install-commandmode

```
[ec2-user@ip-172-31-91-91 nagios-4.5.5]$ make all

sudo make install
sudo make install-init
sudo make install-config
sudo make install-commandmode
cd ./base && make
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE_
CONFIG_H -DNSCORE -c -o nagios.o ./nagios.c
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE_
CONFIG_H -DNSCORE -c -o broker.o broker.c
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE_
CONFIG_H -DNSCORE -c -o nebmods.o nebmods.c
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE_
CONFIG_H -DNSCORE -c -o ../common/shared.o ../common/shared.c
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE_
CONFIG_H -DNSCORE -c -o query-handler.o query-handler.c
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE_
CONFIG_H -DNSCORE -c -o workers.o workers.c
In function ‘get_wproc_list’,
    inlined from ‘get_worker’ at workers.c:277:12:
workers.c:253:17: warning: ‘%s’ directive argument is null [-Wformat-overflo
w=]
  253 |                 log_debug_info(DEBUGL_CHECKS, 1, "Found specialized
worker(s) for '%s'", (slash && *slash != '/') ? slash : cmd_name);
      |                 ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE_
CONFIG_H -DNSCORE -c -o checks.o checks.c
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE_
CONFIG_H -DNSCORE -c -o config.o config.c
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE
```

14. Edit the config file and change the email address.
sudo nano /usr/local/nagios/etc/objects/contacts.cfg

```
#
# CONTACTS
#
###############################################################################

# Just one contact defined by default - the Nagios admin (that's you)
# This contact definition inherits a lot of default values from the
# 'generic-contact' template which is defined elsewhere.

define contact {

    contact_name            nagiosadmin             ; Short name of user
    use                     generic-contact         ; Inherit default values from generic-contact template (defined above)
    alias                   Nagios Admin            ; Full name of user
    email                   2022.ayush.maurya@ves.ac.in ; <****** CHANGE THIS TO YOUR EMAIL ADDRESS ******
}



###############################################################################
#
# CONTACT GROUPS
#
###############################################################################

# We only have one contact in this simple configuration file, so there is
# no need to create more than one contact group.

define contactgroup {

    contactgroup_name       admins
    alias                   Nagios Administrators
    members                 nagiosadmin
}

^G Help          ^O Write Out     ^W Where Is      ^K Cut           ^T Execute       ^C Location      M-U Undo         M-A Set Mark     M-] To Br
^X Exit          ^R Read File     ^\ Replace       ^U Paste         ^J Justify       ^/ Go To Line    M-E Redo         M-6 Copy         ^Q Where
```

And change email with your email

15. Configure the web interface.
sudo make install-webconf

```
[ec2-user@ip-172-31-91-91 nagios-4.5.5]$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios
.conf
if [ 0 -eq 1 ]; then \
        ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagi
s.conf; \
fi

*** Nagios/Apache conf file installed ***

[ec2-user@ip-172-31-91-91 nagios-4.5.5]$
```

16. Create a nagiosadmin account for nagios login along with password. You'll have to specify the password twice.
sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin

```
[ec2-user@ip-172-31-91-91 nagios-4.5.5]$ sudo htpasswd -c /usr/local/nagios/
etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
[ec2-user@ip-172-31-91-91 nagios-4.5.5]$
```

Password: Ayushmau

17. Restart Apache
sudo service httpd restart

```
Adding password for user nagiosadmin
[ec2-user@ip-172-31-91-91 nagios-4.5.5]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-91-91 nagios-4.5.5]$
```

18. Go back to the downloads folder and unzip the plugins zip file.

cd ~/downloads

tar zxvf nagios-plugins-2.4.11.tar.gz

```
[ec2-user@ip-172-31-91-91 downloads]$ cd ~/downloads
[ec2-user@ip-172-31-91-91 downloads]$ tar zxvf nagios-plugins-2.4.11.tar.gz
nagios-plugins-2.4.11/
nagios-plugins-2.4.11/build-aux/
nagios-plugins-2.4.11/build-aux/compile
nagios-plugins-2.4.11/build-aux/config.guess
nagios-plugins-2.4.11/build-aux/config.rpath
nagios-plugins-2.4.11/build-aux/config.sub
nagios-plugins-2.4.11/build-aux/install-sh
nagios-plugins-2.4.11/build-aux/ltmain.sh
nagios-plugins-2.4.11/build-aux/missing
nagios-plugins-2.4.11/build-aux/mkinstalldirs
nagios-plugins-2.4.11/build-aux/depcomp
nagios-plugins-2.4.11/build-aux/snippet/
nagios-plugins-2.4.11/build-aux/snippet/_Noreturn.h
nagios-plugins-2.4.11/build-aux/snippet/arg-nonnull.h
nagios-plugins-2.4.11/build-aux/snippet/c++defs.h
nagios-plugins-2.4.11/build-aux/snippet/warn-on-use.h
nagios-plugins-2.4.11/build-aux/test-driver
nagios-plugins-2.4.11/config_test/
```

19. Compile and install plugins

cd nagios-plugins-2.4.11

./configure --with-nagios-user=nagios --with-nagios-group=nagios

```
[ec2-user@ip-172-31-91-91 downloads]$ cd nagios-plugins-2.4.11
./configure --with-nagios-user=nagios --with-nagios-group=nagios
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether to enable maintainer-specific portions of Makefiles... yes
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether gcc understands -c and -o together... yes
checking whether make supports the include directive... yes (GNU style)
checking dependency style of gcc... gcc3
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
checking for Minix Amsterdam compiler... no
checking for ar... ar
checking for ranlib... ranlib
```

make
sudo make install

```
[ec2-user@ip-172-31-91-91 nagios-plugins-2.4.11]$ make
sudo make install
make  all-recursive
make[1]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
Making all in gl
make[2]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/
gl'
rm -f alloca.h-t alloca.h && \
{ echo '/* DO NOT EDIT! GENERATED AUTOMATICALLY! */'; \
  cat ./alloca.in.h; \
} > alloca.h-t && \
mv -f alloca.h-t alloca.h
rm -f c++defs.h-t c++defs.h && \
sed -n -e '/_GL_CXXDEFS/,$p' \
  < ../build-aux/snippet/c++defs.h \
  > c++defs.h-t && \
mv c++defs.h-t c++defs.h
rm -f warn-on-use.h-t warn-on-use.h && \
sed -n -e '/^.ifndef/,$p' \
  < ../build-aux/snippet/warn-on-use.h \
  > warn-on-use.h-t && \
mv warn-on-use.h-t warn-on-use.h
rm -f arg-nonnull.h-t arg-nonnull.h && \
sed -n -e '/GL_ARG_NONNULL/,$p' \
  < ../build-aux/snippet/arg-nonnull.h \
  > arg-nonnull.h-t && \
mv arg-nonnull.h-t arg-nonnull.h
/usr/bin/mkdir -p arpa
```

```
make[1]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[2]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
[ec2-user@ip-172-31-91-91 nagios-plugins-2.4.11]$
```

20. Start Nagios
Add Nagios to the list of system services
sudo chkconfig --add nagios
sudo chkconfig nagios on

```
[ec2-user@ip-172-31-91-91 nagios-plugins-2.4.11]$ sudo chkconfig --add nagio
s
sudo chkconfig nagios on
Note: Forwarding request to 'systemctl enable nagios.service'.
Synchronizing state of nagios.service with SysV service script with /usr/lib
/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable nagios
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service →
 /usr/lib/systemd/system/nagios.service.
[ec2-user@ip-172-31-91-91 nagios-plugins-2.4.11]$
```

Verify the sample configuration files
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Error

```
[ec2-user@ip-172-31-91-91 nagios-plugins-2.0.3]$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.0.8
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 08-12-2014
License: GPL

Website: http://www.nagios.org
Reading configuration data...
Error in configuration file '/usr/local/nagios/etc/nagios.cfg' - Line 452 (Check result path '/usr/local/nagios/var/spool/checkres
lts' is not a valid directory)
   Error processing main config file!
```

Solution:

 # Create the missing directory: If the directory is missing, create it with the necessary permissions:

sudo mkdir -p /usr/local/nagios/var/spool/checkresults

sudo chown nagios:nagios /usr/local/nagios/var/spool/checkresults

sudo chmod 775 /usr/local/nagios/var/spool/checkresults

```
[ec2-user@ip-172-31-91-91 nagios-plugins-2.0.3]$ sudo mkdir -p /usr/local/nagios/var/spool/checkresults
sudo chown nagios:nagios /usr/local/nagios/var/spool/checkresults
sudo chmod 775 /usr/local/nagios/var/spool/checkresults
```

Now run again

sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

```
[ec2-user@ip-172-31-91-91 nagios-plugins-2.4.11]$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.5.5
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2024-09-17
License: GPL

Website: https://www.nagios.org
Reading configuration data...
   Read main config file okay...
   Read object config files okay...

Running pre-flight check on configuration data...

Checking objects...
        Checked 8 services.
        Checked 1 hosts.
        Checked 1 host groups.
        Checked 0 service groups.
        Checked 1 contacts.
        Checked 1 contact groups.
        Checked 24 commands.
        Checked 5 time periods.
        Checked 0 host escalations.
        Checked 0 service escalations.
Checking for circular paths...
        Checked 1 hosts
        Checked 0 service dependencies
        Checked 0 host dependencies
        Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors:   0
```

sudo service nagios start

```
[ec2-user@ip-172-31-91-91 nagios-plugins-2.4.11]$ sudo service nagios start
Starting nagios (via systemctl):                           [  OK  ]
```

21. Check the status of Nagios

sudo systemctl status nagios

```
[ec2-user@ip-172-31-91-91 nagios-plugins-2.0.3]$ sudo systemctl status nagios
● nagios.service - LSB: Starts and stops the Nagios monitoring server
     Loaded: loaded (/etc/rc.d/init.d/nagios; generated)
     Active: active (running) since Sun 2024-09-29 08:04:30 UTC; 37s ago
       Docs: man:systemd-sysv-generator(8)
    Process: 68037 ExecStart=/etc/rc.d/init.d/nagios start (code=exited, status=0/SUCCESS)
      Tasks: 6 (limit: 1112)
     Memory: 2.0M
        CPU: 47ms
     CGroup: /system.slice/nagios.service
             ├─68059 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─68061 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─68062 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─68063 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─68064 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─68065 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: wproc: Registry request: name=Core Worker 68063;pid=68063
Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: wproc: Registry request: name=Core Worker 68062;pid=68062
Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: wproc: Registry request: name=Core Worker 68064;pid=68064
Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: wproc: Registry request: name=Core Worker 68061;pid=68061
Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: Warning: Could not open object cache file '/usr/local/nagios/var/objec>
Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmpx2N>
Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: Successfully launched command file worker with pid 68065
Sep 29 08:04:39 ip-172-31-91-91.ec2.internal nagios[68059]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmpTmg>
Sep 29 08:04:49 ip-172-31-91-91.ec2.internal nagios[68059]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmpAfy>
Sep 29 08:04:59 ip-172-31-91-91.ec2.internal nagios[68059]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmpCtQ>
lines 1-26/26 (END)
```

Error:

The log messages suggest that Nagios is unable to create temporary files, particularly in the directory `/usr/local/nagios/var/`. This is typically caused by permission issues, or the directory might not exist.

Solution:

Firstly check whether `/usr/local/nagios/var/` is there or not. If yes.....

`ls -ld /usr/local/nagios/var/`

Change ownership: Set the correct ownership for the Nagios user and group:

sudo chown -R nagios:nagcmd /usr/local/nagios/var

Set permissions: Ensure the directory has the right permissions:

sudo chmod -R 775 /usr/local/nagios/var

Restart Nagios: After adjusting the ownership and permissions, restart the Nagios service:

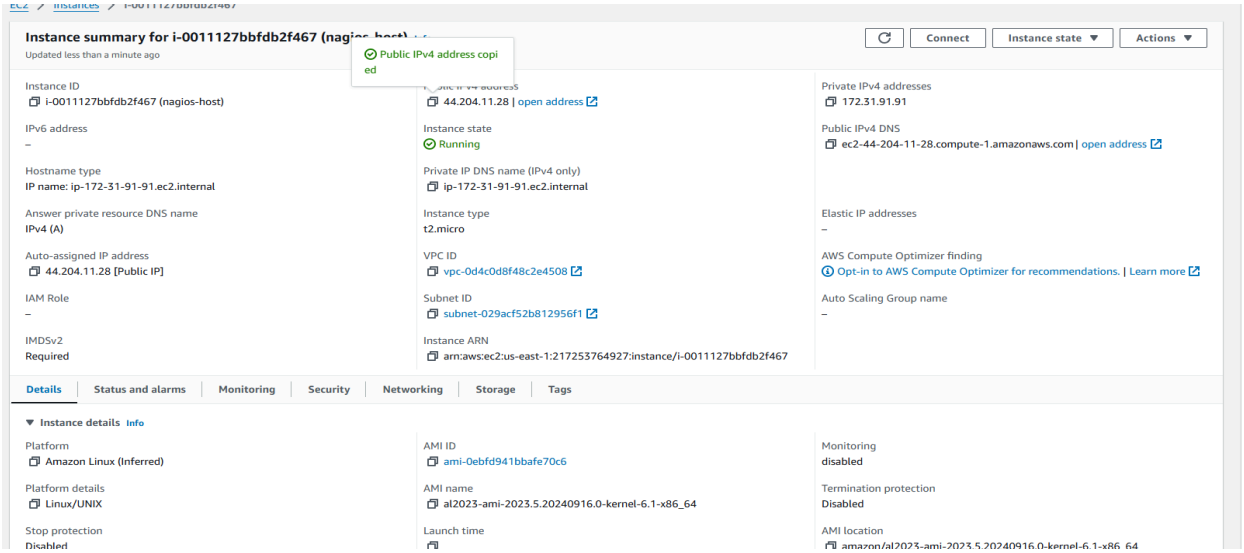
sudo systemctl restart nagios

```
drwxr-xr-x. 4 root root 112 Sep 29 08:04 /usr/local/nagios/var/
[ec2-user@ip-172-31-91-91 nagios-plugins-2.0.3]$ sudo chown -R nagios:nagcmd /usr/local/nagios/var
[ec2-user@ip-172-31-91-91 nagios-plugins-2.0.3]$ sudo chmod -R 775 /usr/local/nagios/var
[ec2-user@ip-172-31-91-91 nagios-plugins-2.0.3]$ sudo systemctl restart nagios
[ec2-user@ip-172-31-91-91 nagios-plugins-2.0.3]$
```

Now run again

```
[ec2-user@ip-172-31-91-91 nagios-plugins-2.4.11]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
     Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
     Active: active (running) since Sun 2024-09-29 08:51:47 UTC; 42min ago
       Docs: https://www.nagios.org/documentation
      Tasks: 6 (limit: 1112)
     Memory: 2.9M
        CPU: 562ms
     CGroup: /system.slice/nagios.service
             ├─71188 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─71190 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─71191 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─71192 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─71193 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─71194 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 29 08:51:47 ip-172-31-91-91.ec2.internal nagios[71188]: wproc: Registry request: name=Core Worker 71191;pid=71191
Sep 29 08:51:47 ip-172-31-91-91.ec2.internal nagios[71188]: wproc: Registry request: name=Core Worker 71190;pid=71190
Sep 29 08:51:47 ip-172-31-91-91.ec2.internal nagios[71188]: Successfully launched command file worker with pid 71194
Sep 29 08:59:22 ip-172-31-91-91.ec2.internal nagios[71188]: SERVICE ALERT: localhost;HTTP;WARNING;HARD;4;HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes i>
Sep 29 09:11:52 ip-172-31-91-91.ec2.internal nagios[71188]: SERVICE NOTIFICATION: nagiosadmin;localhost;Swap Usage;CRITICAL;notify-service-by-email;SWAP CR>
Sep 29 09:11:52 ip-172-31-91-91.ec2.internal nagios[71188]: wproc: NOTIFY job 10 from worker Core Worker 71192 is a non-check helper but exited with return>
Sep 29 09:11:52 ip-172-31-91-91.ec2.internal nagios[71188]: wproc:   host=localhost; service=Swap Usage; contact=nagiosadmin
Sep 29 09:11:52 ip-172-31-91-91.ec2.internal nagios[71188]: wproc:   early_timeout=0; exited_ok=1; wait_status=32512; error_code=0;
Sep 29 09:11:52 ip-172-31-91-91.ec2.internal nagios[71188]: wproc:   stderr line 01: /bin/sh: line 1: /bin/mail: No such file or directory
Sep 29 09:11:52 ip-172-31-91-91.ec2.internal nagios[71188]: wproc:   stderr line 02: /usr/bin/printf: write error: Broken pipe
lines 1-25/25 (END)
```

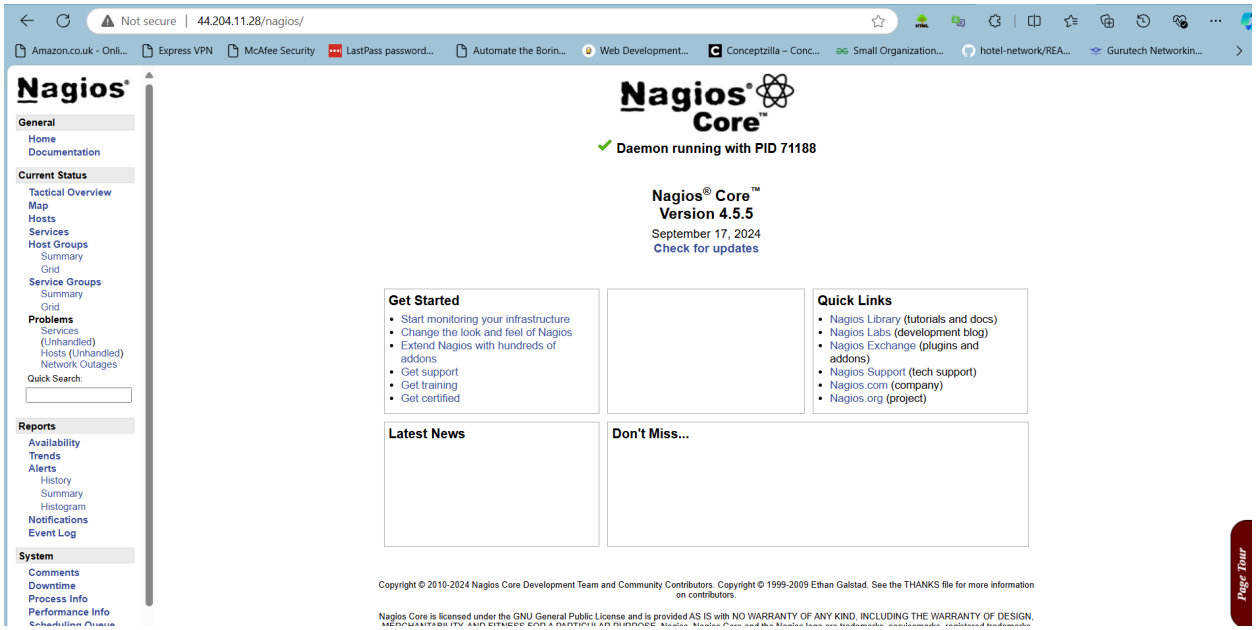
22. Go back to EC2 Console and copy the Public IP address of this instance



23. Open up your browser and look for http://<your_public_ip_address>/nagios

Enter username as nagiosadmin and password which you set in Step 16.

24. After entering the correct credentials, you will see this page.



This means that Nagios was correctly installed and configured with its plugins so far.

Conclusion:

In this practical, we successfully installed and configured Nagios Core along with Nagios plugins and NRPE on an Amazon EC2 instance. We created a Nagios user, set up necessary permissions, and resolved common installation errors. Finally, we verified the setup by accessing the Nagios web interface, confirming that our monitoring system was fully operational.

Adv DevOps Practical 10

Aim: To perform Port, Service monitoring, and Windows/Linux server monitoring using Nagios.

Theory:

Port and Service Monitoring

Port and service monitoring in Nagios involves checking the availability and responsiveness of network services running on specific ports. This ensures that critical services (like HTTP, FTP, or SSH) are operational. Nagios uses plugins to ping the ports and verify whether services are up and responding as expected, allowing administrators to be alerted in case of outages.

Windows/Linux Server Monitoring

Windows/Linux server monitoring with Nagios entails tracking the performance and health of servers running these operating systems. It includes monitoring metrics such as CPU usage, memory consumption, disk space, and system logs. Nagios employs various plugins to gather data, enabling administrators to ensure optimal performance, identify potential issues, and maintain uptime across their server infrastructure.

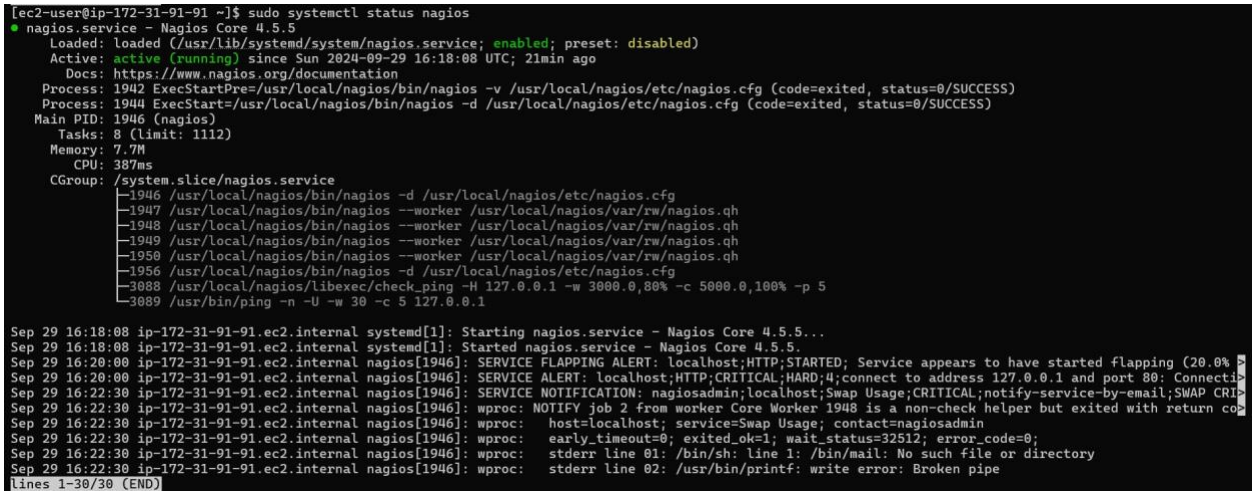
Prerequisites:

AWS Academy or Personal account.

Nagios Server running on Amazon Linux Machine. (Refer Experiment No 9)

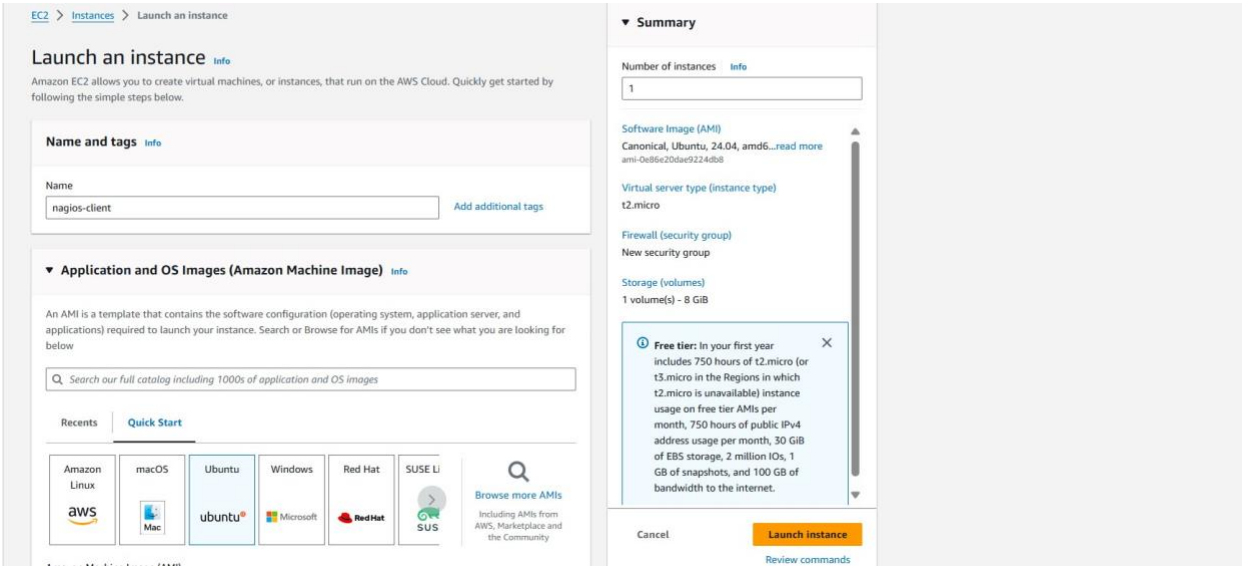
Monitoring Using Nagios:

Step 1: To Confirm Nagios is running on the server side Perform the following command on your Amazon Linux Machine (Nagios-host). **sudo systemctl status nagios**



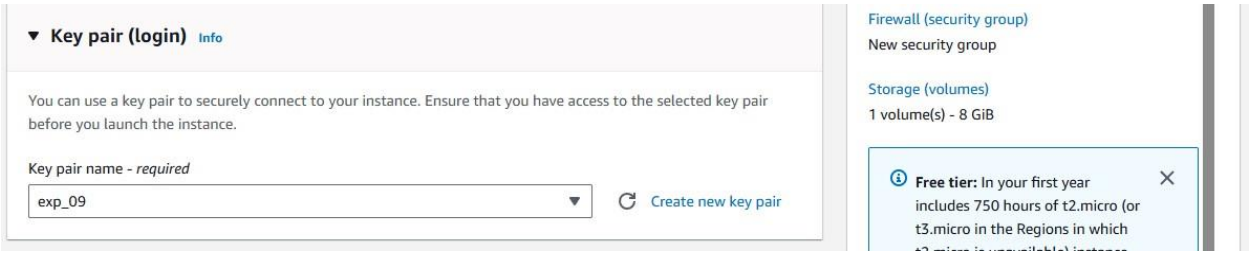
You can now proceed if you get the above message/output.

Step 2: Now Create a new EC2 instance. Name: Nagios-client, AMI: Ubuntu Instance Type: t2.micro.

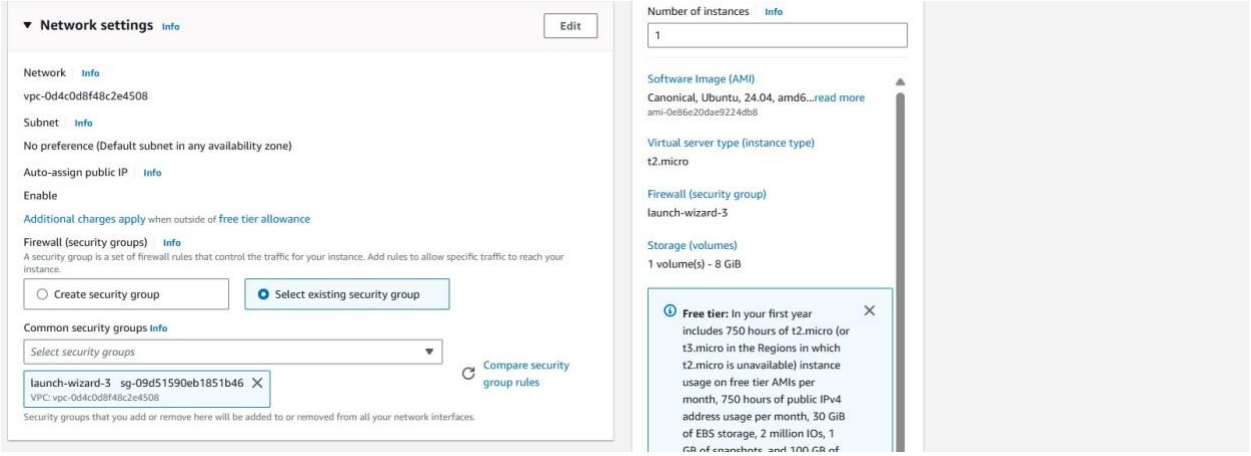


For Key pair : Click on create key and make key of type RSA with extension .pem . Key will be downloaded to your local machine.

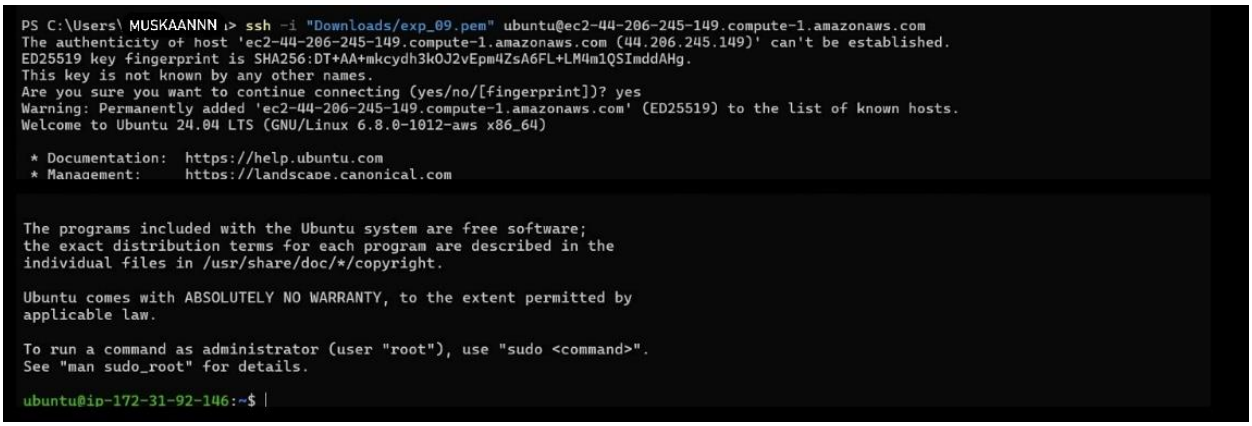
Now select that key in key pair if you already have key with type RSA and extension .pem no need to create new key but you must have that key downloaded.



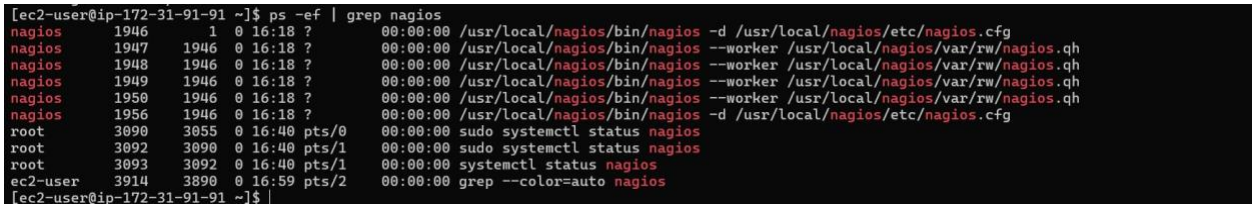
Select the Existing Security Group and select the Security Group that we have created in Experiment no 9 or the same one you have used for the Nagios server (Nagios-host).



Step 3: Now After creating the EC2 Instance click on connect and then copy the command which is given as example in the SSH Client section .
Now open the terminal in the folder where your key(RSA key with .pem) is located. and paste that copied command.



Now perform all the commands on the Nagios-host till step 10
Step 4: Now on the server Nagios-host run the following command.
ps -ef | grep nagios



Step 5: Now Become root user and create root directories.
sudo su
mkdir /usr/local/nagios/etc/objects/monitorhosts mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts

Step 6: Copy the sample localhost.cfg to linuxhost.cfg by running the following command.(Below command should come in one line see screenshot below)
cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg

Step 7: Open linuxserver.cfg using nano and make the following changes in all positions?everywhere in file.

> **nano /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg**

Change hostname to **linuxserver.**
Change address to the **public IP of your Linux client**.
Set hostgroup_name to **linux-servers1.**

```
##############################################################################
#
# HOST DEFINITION
#
##############################################################################

# Define a host for the local machine

define host {

    use                     linux-server            ; Name of host template to use
                                                    ; This host definition will inherit all variables that are defined
                                                    ; in (or inherited by) the linux-server host template definition.
    host_name               linuxserver
    alias                   localhost
    address                 172.31.92.146
}



##############################################################################
#
# HOST GROUP DEFINITION
#
##############################################################################

# Define an optional hostgroup for Linux machines

define hostgroup {

    hostgroup_name          linux-servers1          ; The name of the hostgroup
    alias                   Linux Servers           ; Long name of the group
    members                 localhost               ; Comma separated list of hosts that belong to this group
}
```

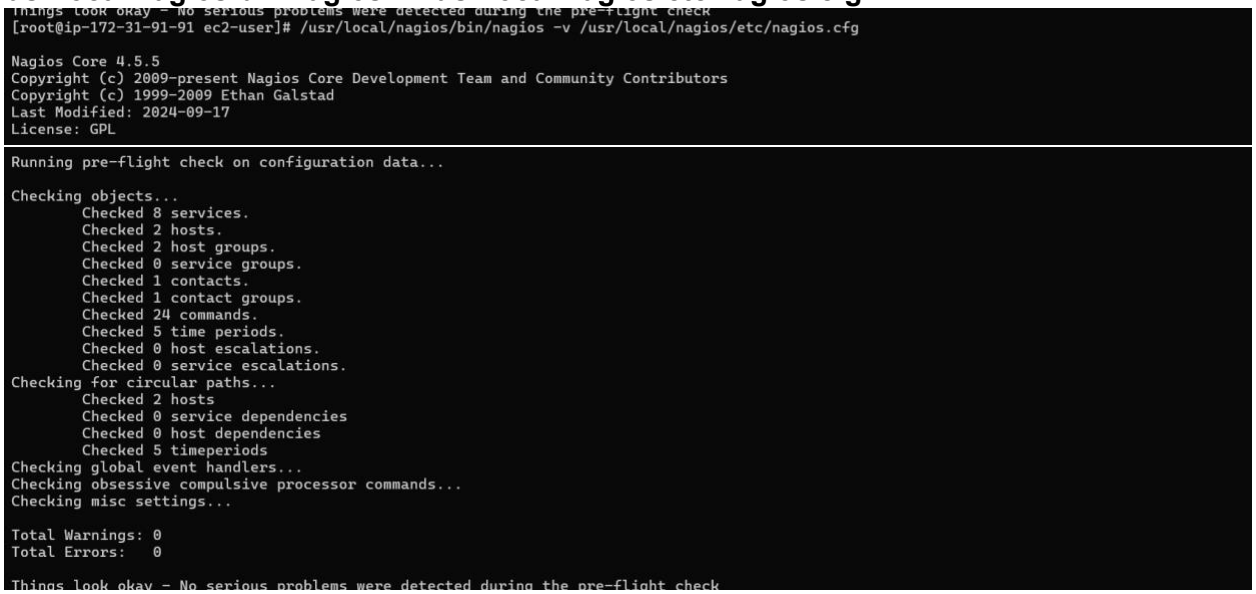
Step 8: Now update the Nagios config file .Add the following line in the file. Line to add :
> nano /usr/local/nagios/etc/nagios.cfg

cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/

```
#
# NAGIOS.CFG - Sample Main Config File for Nagios 4.5.5
#
# Read the documentation for more information on this configuration
# file.  I've provided some comments here, but things may not be so
# clear without further explanation.
#
#
##############################################################################


# LOG FILE
# This is the main log file where service and host events are logged
# for historical purposes.  This should be the first option specified
# in the config file!!!

log_file=/usr/local/nagios/var/nagios.log



# OBJECT CONFIGURATION FILE(S)
# These are the object configuration files in which you define hosts,
# host groups, contacts, contact groups, services, etc.
# You can split your object definitions across several config files
# if you wish (as shown below), or keep them all in a single config file.

# You can specify individual object config files as shown below:
cfg_file=/usr/local/nagios/etc/objects/commands.cfg
cfg_file=/usr/local/nagios/etc/objects/contacts.cfg
cfg_file=/usr/local/nagios/etc/objects/timeperiods.cfg
cfg_file=/usr/local/nagios/etc/objects/templates.cfg
cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/
# Definitions for monitoring the local (Linux) host
cfg_file=/usr/local/nagios/etc/objects/localhost.cfg
```

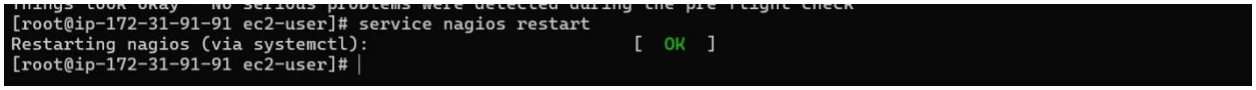
Step 9: Now Verify the configuration files by running the following commands.

/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg



Step 10: Now restart the services of nagios by running the following command.

service nagios restart



Step 11: Now Go to the Nagios-client ssh terminal and update and install the packages by running the following command.

sudo apt update -y sudo apt install gcc -y sudo apt install -y nagios-nrpe-server nagios-plugins

```
ubuntu@ip-172-31-92-146:~$ sudo apt update -y
sudo apt install gcc -y
sudo apt install -y nagios-nrpe-server nagios-plugins

Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [380 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [82.9 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4560 B]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [272 kB]
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [115 kB]
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [8632 B]
Get:14 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [10.3 kB]
Get:15 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [353 kB]
Get:16 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [68.1 kB]
Get:17 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 c-n-f Metadata [428 B]
Get:18 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [10.9 kB]
Get:19 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [2808 B]
Get:20 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:21 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [344 B]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
```

```
Service restarts being deferred:
 /etc/needrestart/restart.d/dbus.service
 systemctl restart getty@tty1.service
 systemctl restart networkd-dispatcher.service
 systemctl restart serial-getty@ttyS0.service
 systemctl restart systemd-logind.service
 systemctl restart unattended-upgrades.service

No containers need to be restarted.

User sessions running outdated binaries:
 ubuntu @ session #2: sshd[992,1102]
 ubuntu @ session #7: sshd[1190,1248]
ubuntu@ip-172-31-92-146:~$
```

Step 12: Open nrpe.cfg file to make changes.Under allowed_hosts, add your nagios host IP address. **sudo nano /etc/nagios/nrpe.cfg**

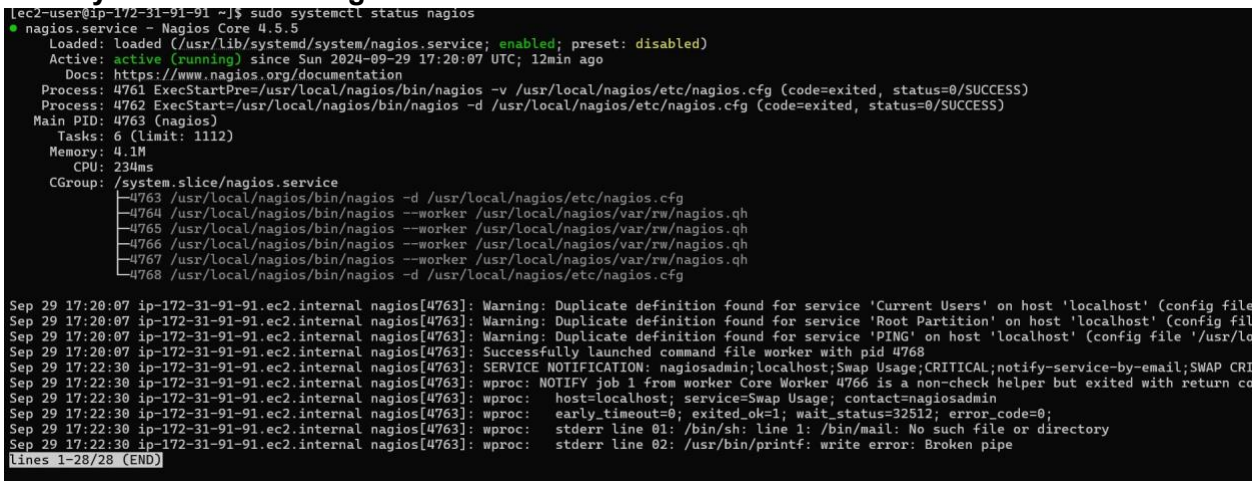
```
# NRPE GROUP
# This determines the effective group that the NRPE daemon should run as.
# You can either supply a group name or a GID.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd

nrpe_group=nagios



# ALLOWED HOST ADDRESSES
# This is an optional comma-delimited list of IP address or hostnames
# that are allowed to talk to the NRPE daemon. Network addresses with a bit mask
# (i.e. 192.168.1.0/24) are also supported. Hostname wildcards are not currently
# supported.
#
# Note: The daemon only does rudimentary checking of the client's IP
# address.  I would highly recommend adding entries in your /etc/hosts.allow
# file to allow only the specified host to connect to the port
# you are running this daemon on.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd

allowed_hosts=127.0.0.1,::1,34.207.68.187



# COMMAND ARGUMENT PROCESSING
# This option determines whether or not the NRPE daemon will allow clients
# to specify arguments to commands that are executed.  This option only works
# if the daemon was configured with the --enable-command-args configure script
# option.
#
# *** ENABLING THIS OPTION IS A SECURITY RISK! ***
```

Step 13: Now restart the NRPE server by this command.
sudo systemctl restart nagios-nrpe-server

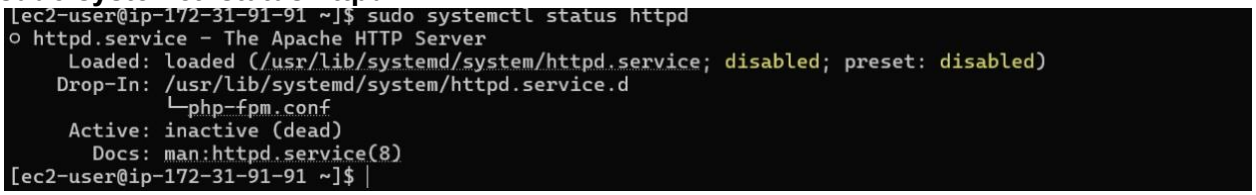
```
0 upgraded, 0 newly installed, 0 to remove and 139 not upgraded.
ubuntu@ip-172-31-92-146:~$ sudo nano /etc/nagios/nrpe.cfg
ubuntu@ip-172-31-92-146:~$ sudo systemctl restart nagios-nrpe-server
ubuntu@ip-172-31-92-146:~$
```

Step 14: Now again check the status of Nagios by running this command on Nagios-host and also check httpd is active and run the command to active it.

sudo systemctl status nagios



sudo systemctl status httpd

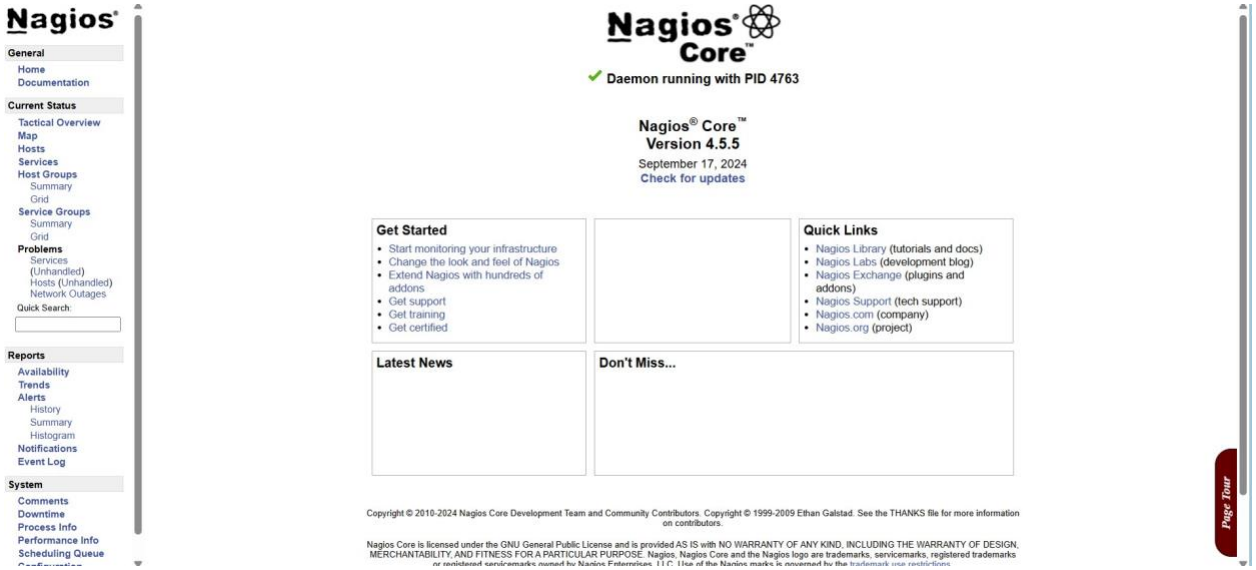


sudo systemctl start httpd

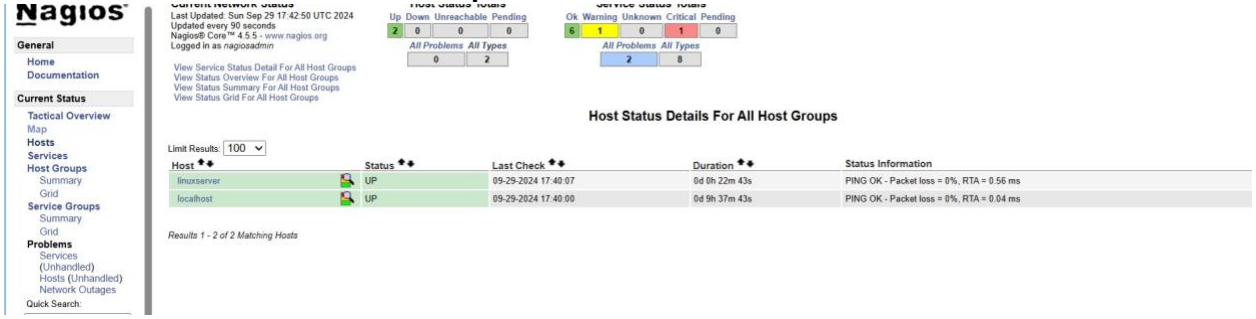
sudo systemctl enable httpd

Step 15: Now to check Nagios dashboard go to http://<nagios host ip>/nagios Eg. http://34.207.68.187/nagios

Enter username as nagiosadmin and password which you set in Exp 9.



Now Click on Hosts from left side panel

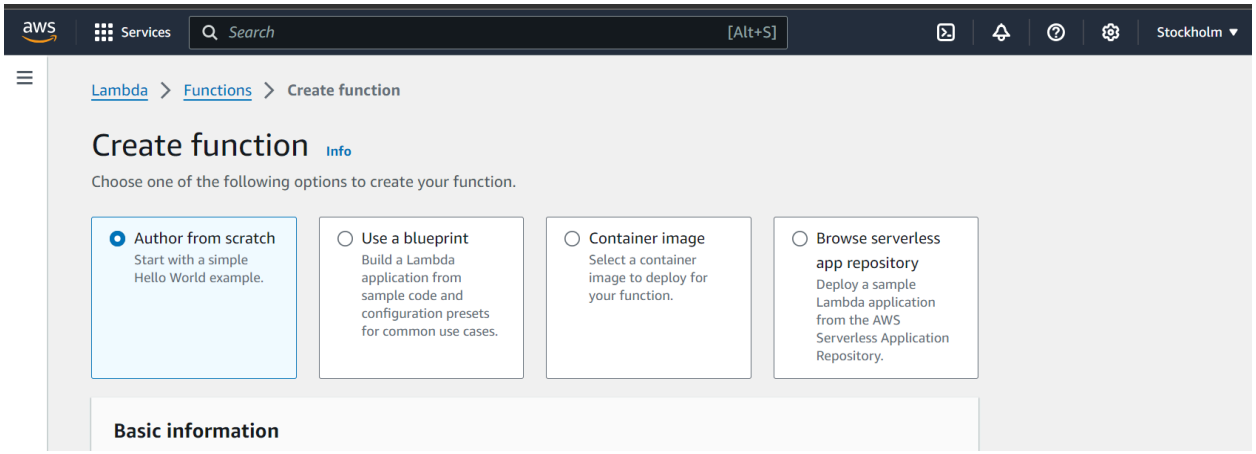


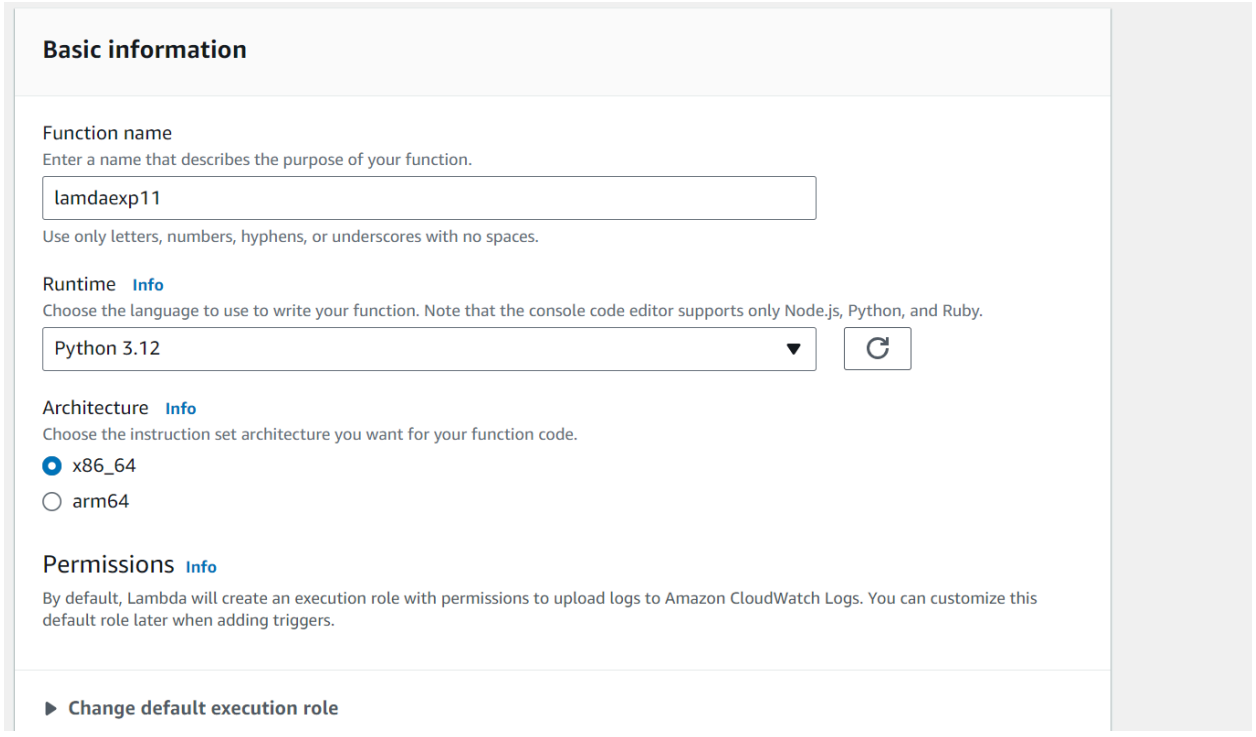
Conclusion:

In this practical, we set up a Nagios host and client to monitor services and server performance on both Linux and Windows servers. We configured Nagios on an Amazon Linux machine to monitor critical services like HTTP, SSH, and system resources, ensuring their availability and health. By creating and configuring a new EC2 instance as the Nagios client, we enabled seamless communication between the client and host for efficient service monitoring. This setup helps ensure uptime and quick detection of issues across the infrastructure.

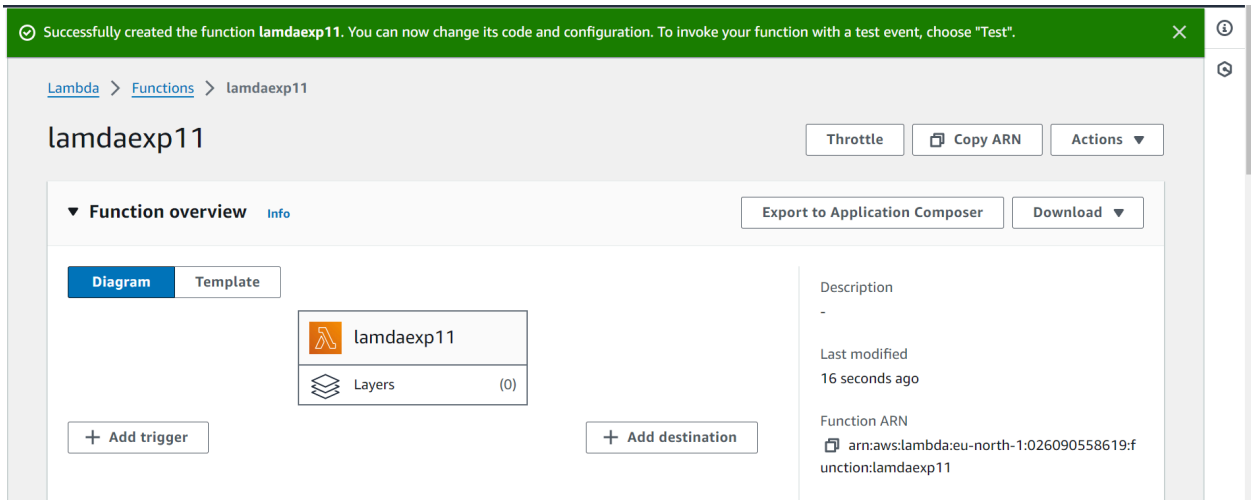
AIM:To understand AWS Lambda, its workflow, various functions and create your first Lambda functions using Python / Java / Nodejs.

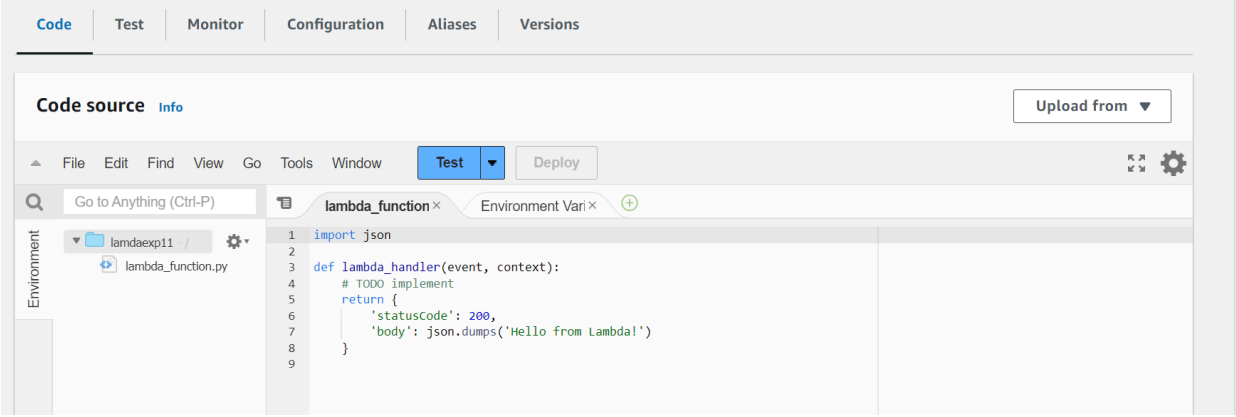
STEP1:Go on your AWS console account and search for lamba and then go on create function Select the author from scratch, add function name and then, choose a runtime env for your function, under the dropdown, you can see all the options AWS supports, Python, Nodejs, .NET and Java being the most popular ones.

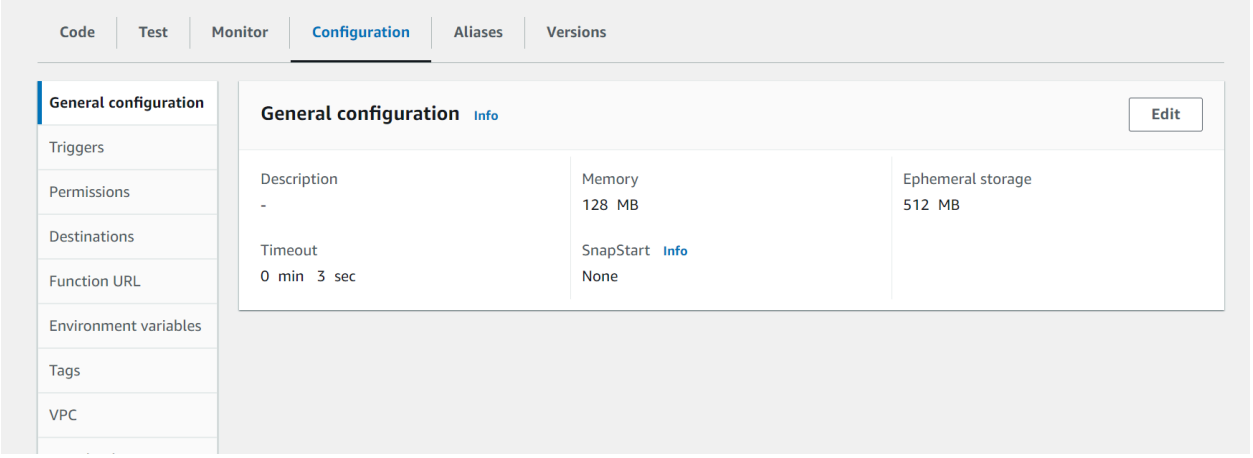




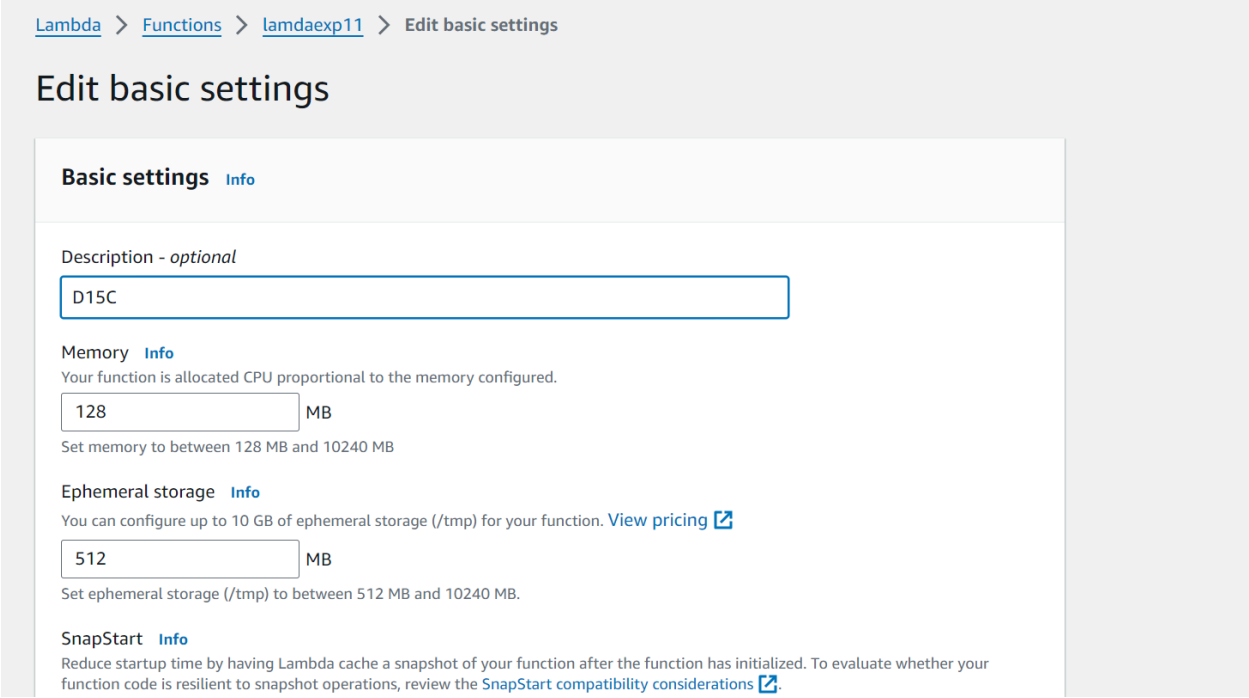
STEP 2:After the function is created successfully go on code write the default code and then configure them.



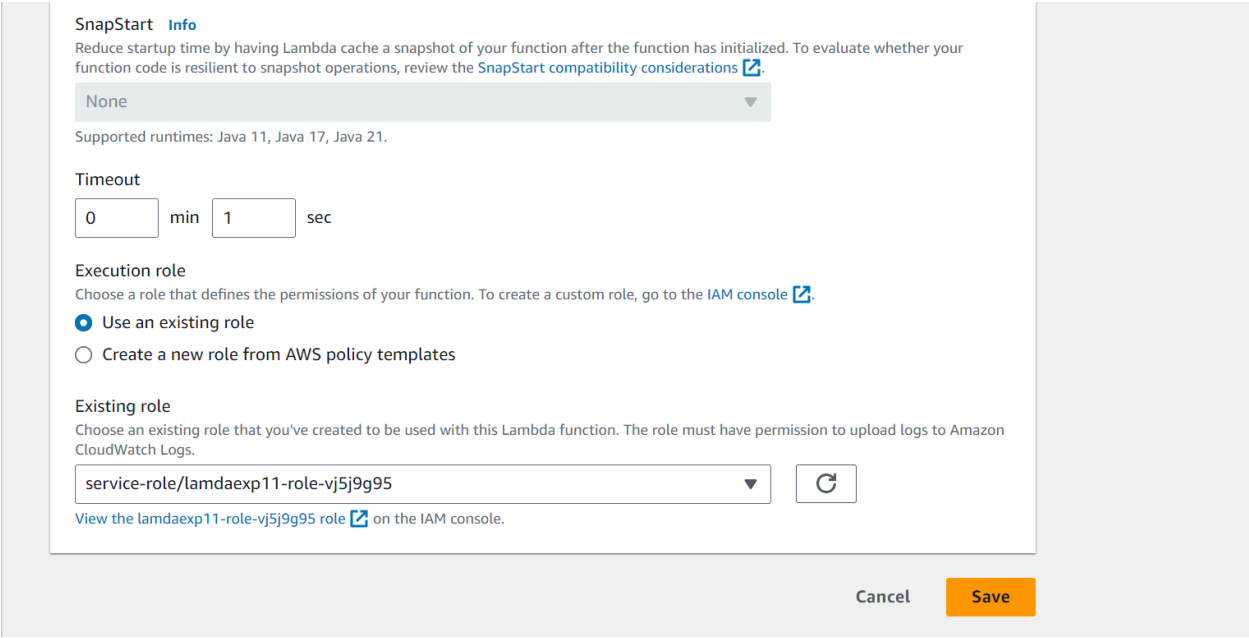




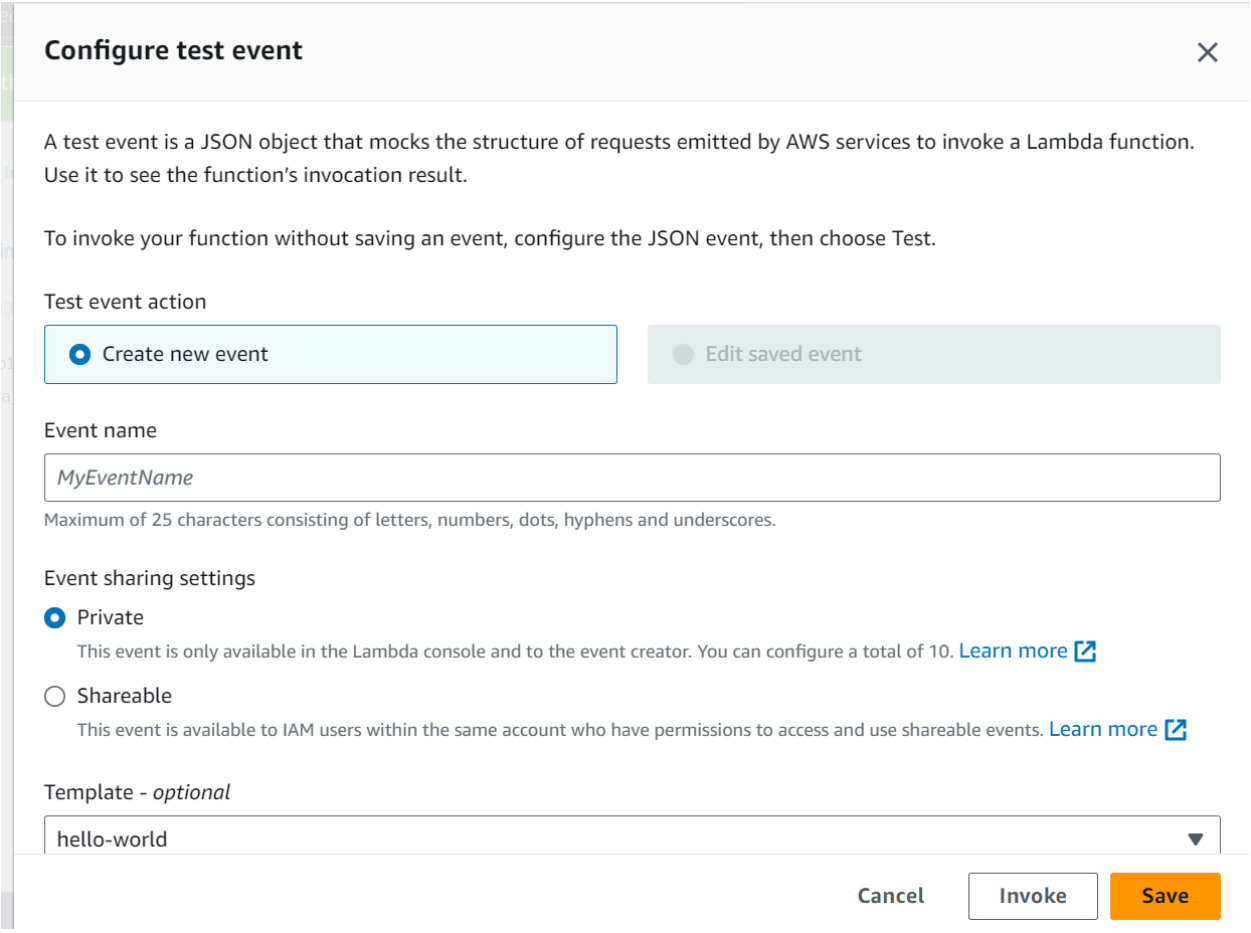
STEP 3: Then go on edit basic settings and add the description and then save it .



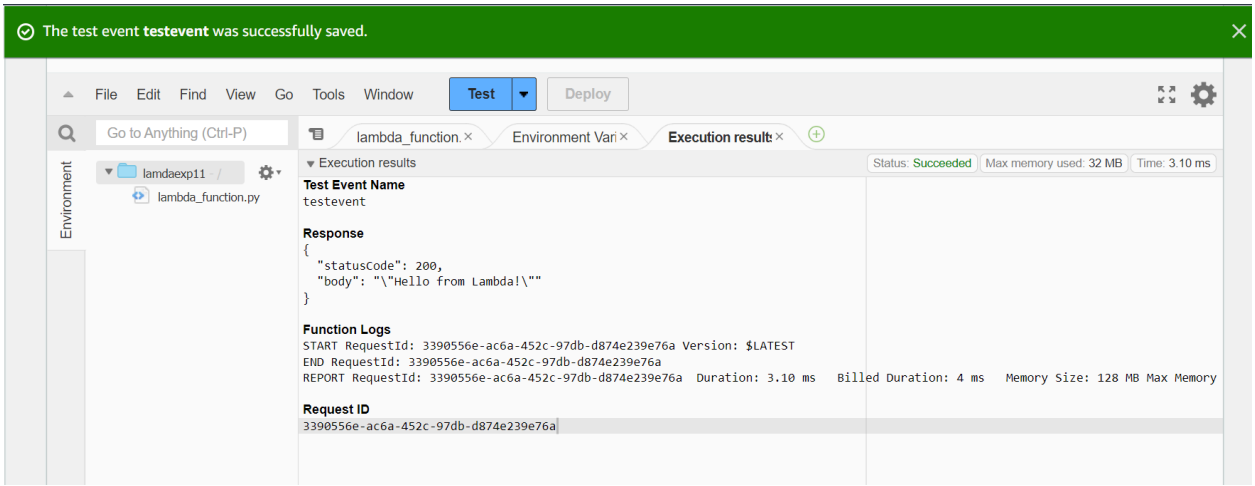
STEP 4:Click on “use an existing role “option and then ahead add the role and save it.



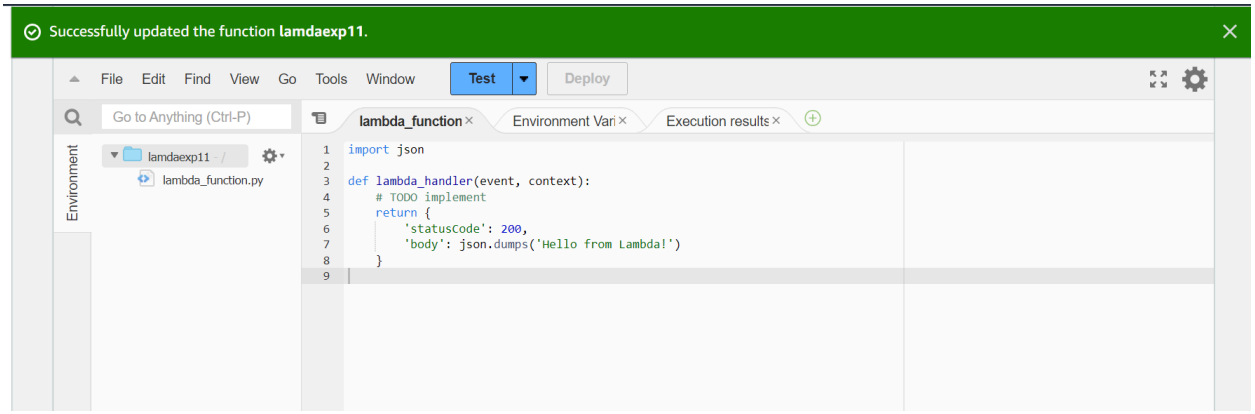
STEP 5:Go on configure test event click on “create new event” edit the event sharing accordingly and select hello world template for template option and then save it.



STEP 6: Click on the test and test the code.



STEP 7: The function is successfully added .

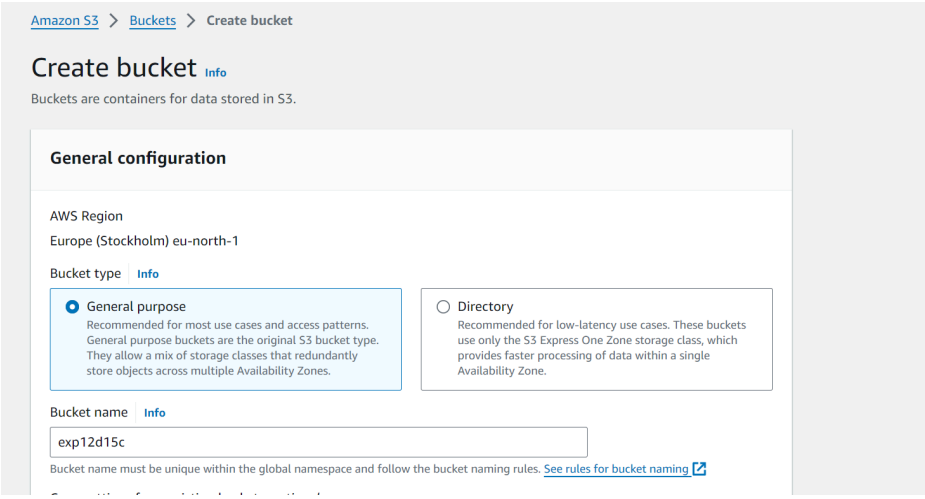


Conclusion:In conclusion, the experiment successfully involved the creation, coding, and deployment of AWS Lambda function. By writing and refining the source code, we demonstrated the ability to implement specific functionality within the Lambda environment. The successful testing of the function confirmed its operational integrity and effectiveness in executing the desired tasks

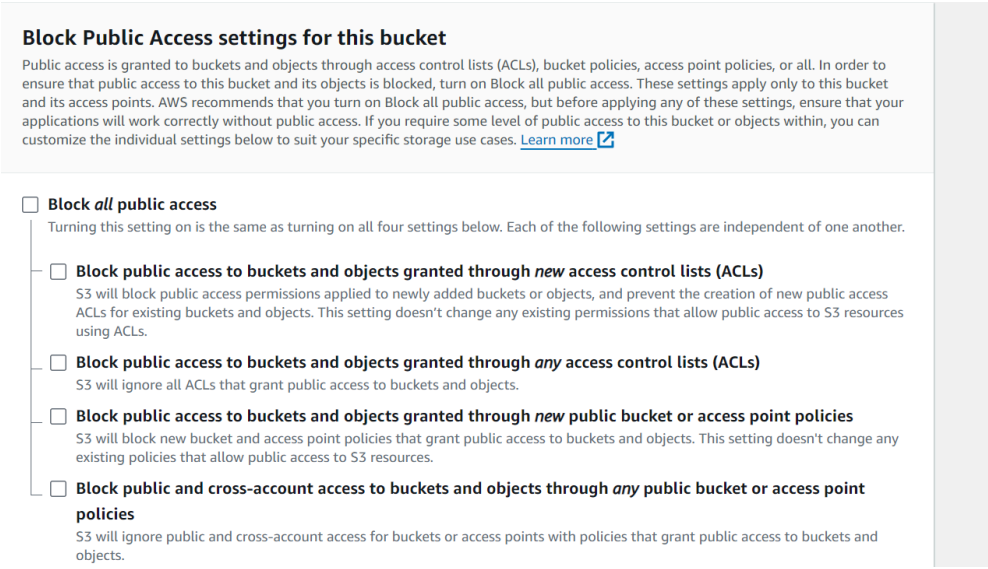
Aim: To create a Lambda function which will log “An Image has been added” once you add an object to a specific bucket in S3

STEPS:

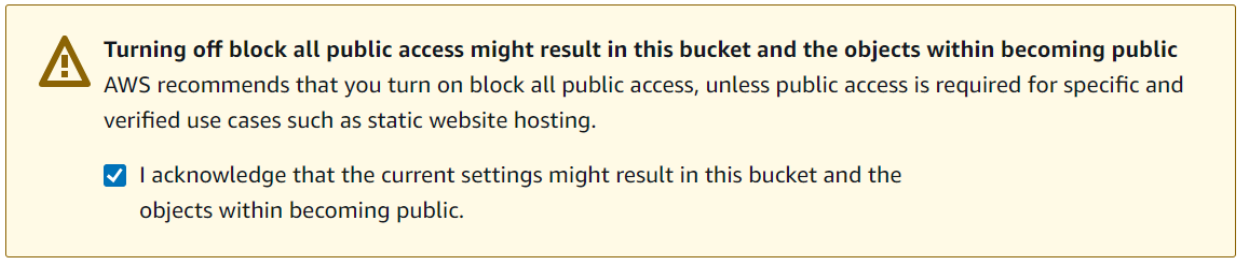
1. Create a S3 bucket and give it a bucket name



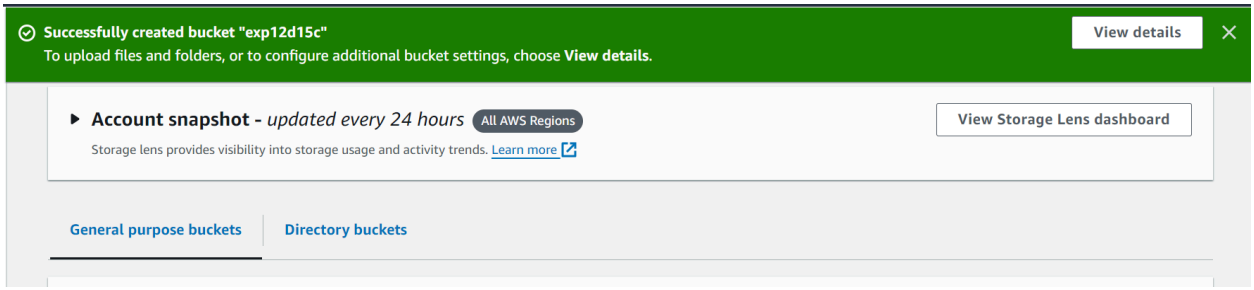
2. Allow public access to the bucket as we are going to add this bucket as a trigger for our lamda function



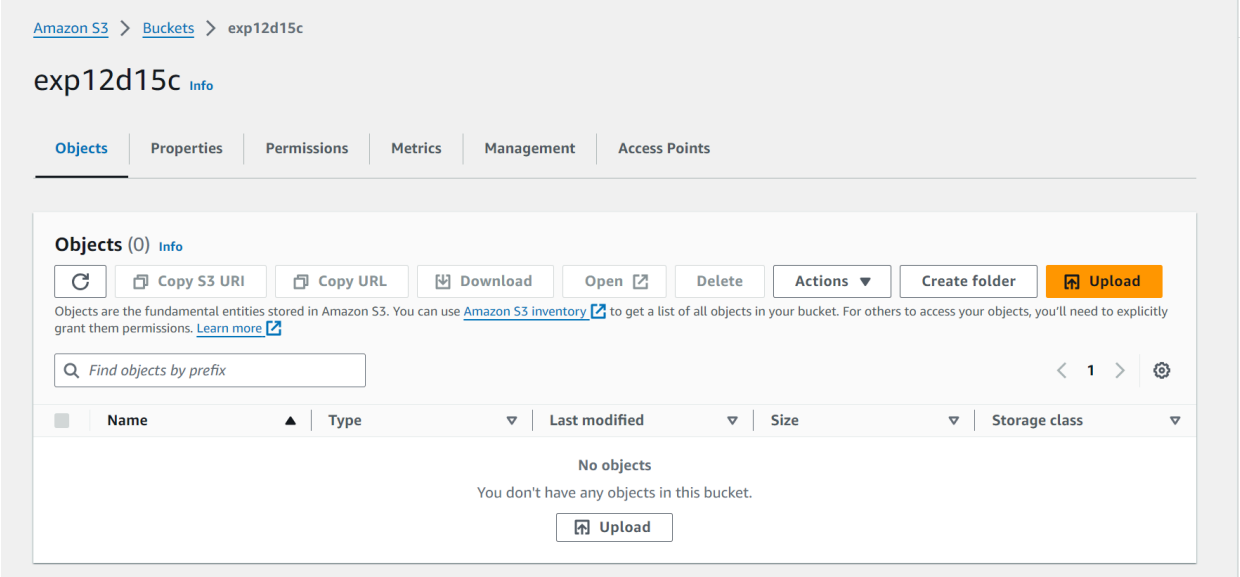
3. Give confirmation that you want to allow full public access and create the bucket



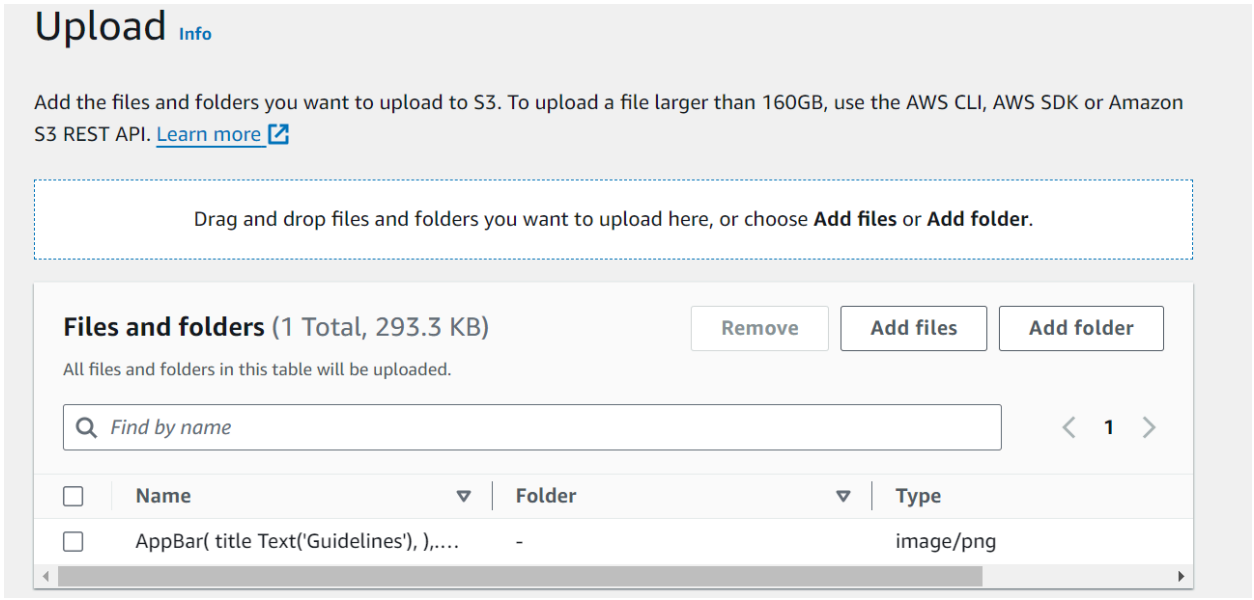
4. You will see the confirmation that the bucket is created successfully



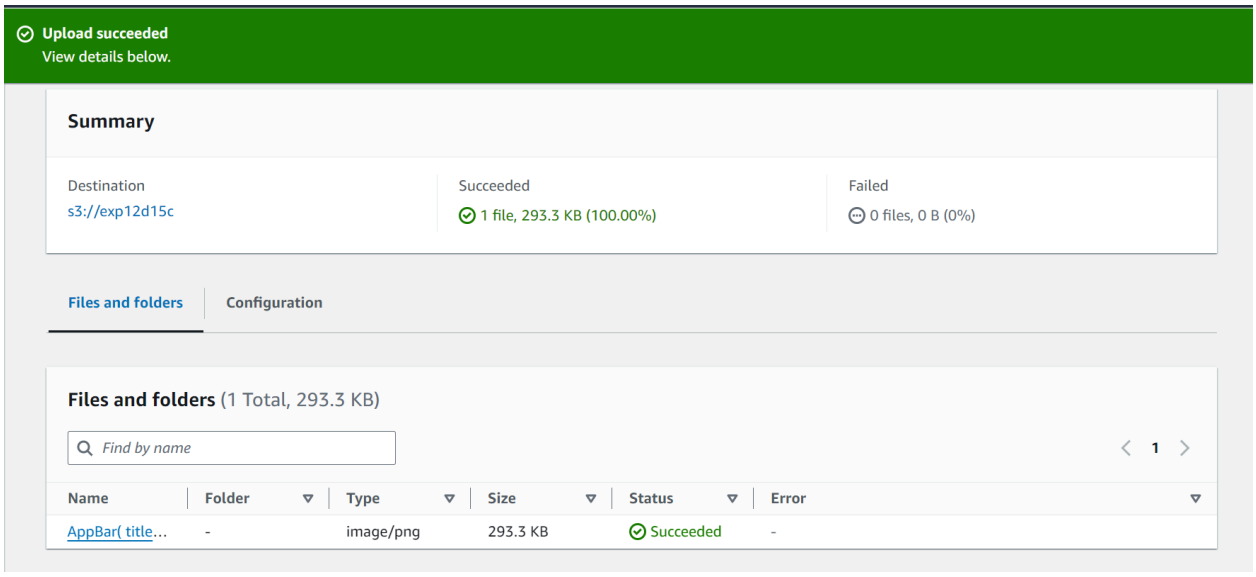
5. Now we need to upload something in the bucket so click on the upload button and add a file



6. I have added a .png extension file; You can upload a .txt file as well

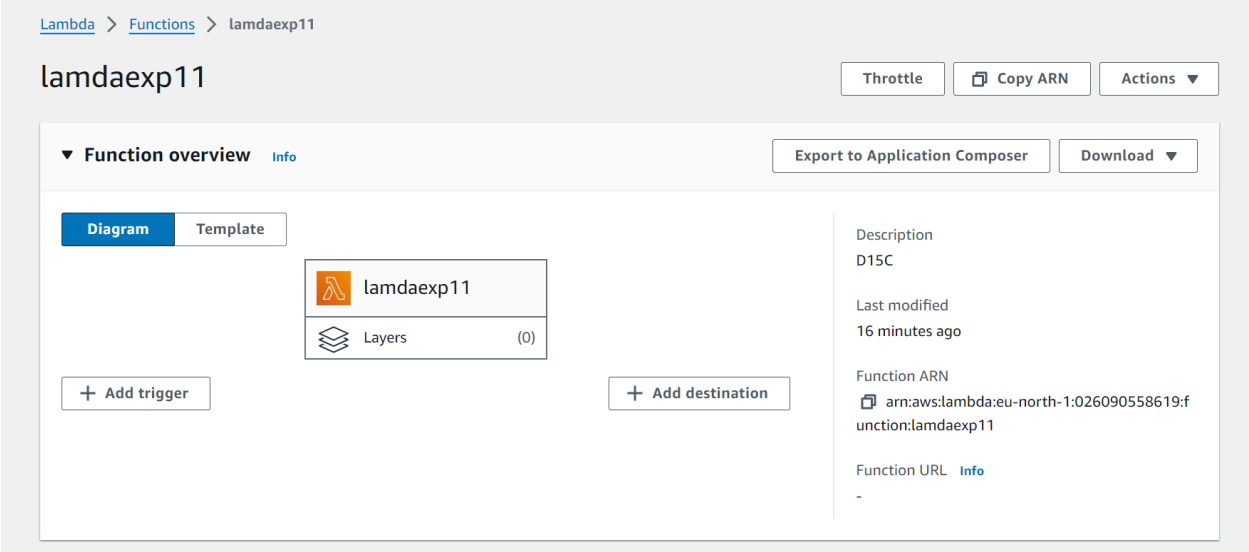


7. Here you can see the confirmation that the upload was a success

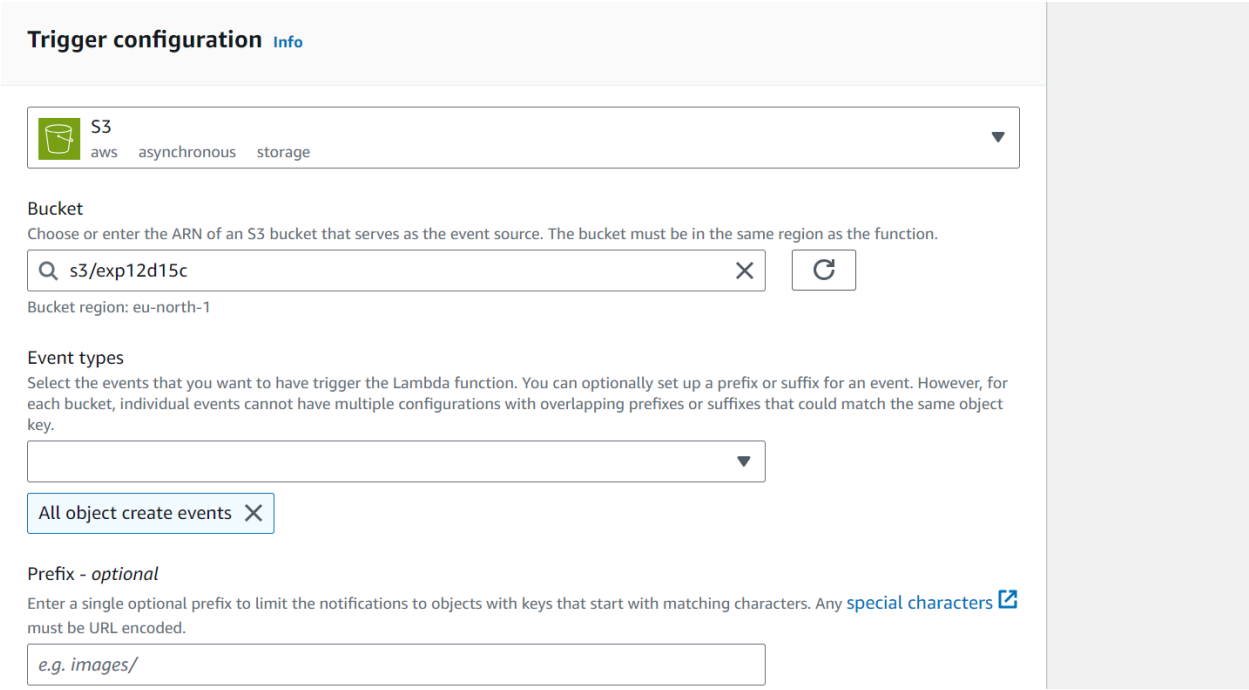


8. Now go back to the aws dashboard and search for lamda function service, Open the function we created in experiment 10. We are going to add this bucket as a trigger to this function

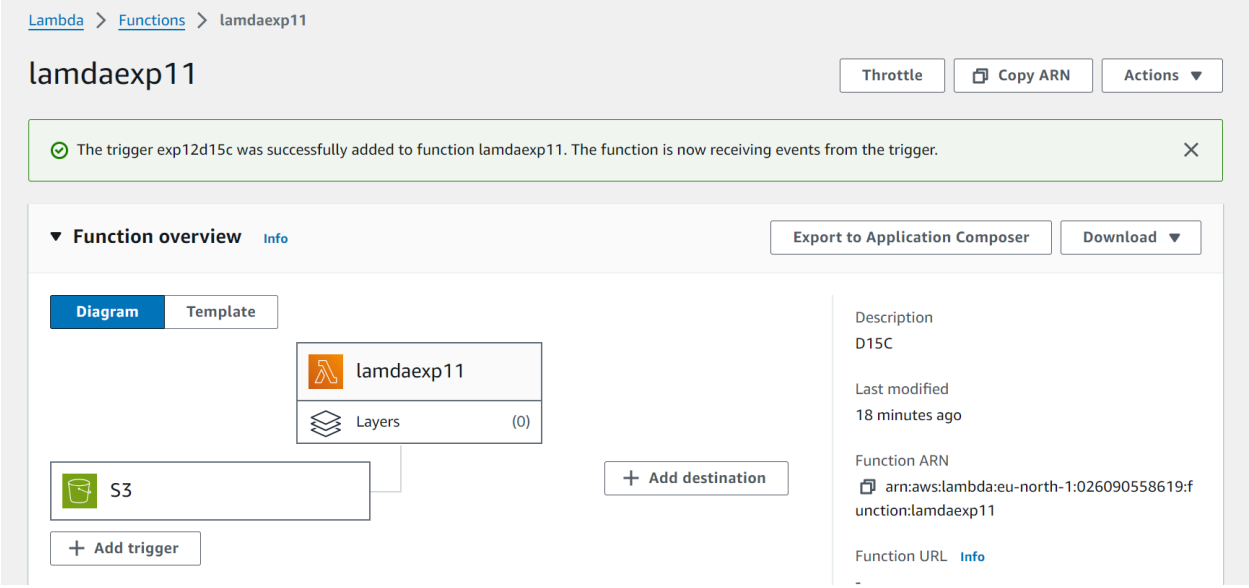
9. On the function overview section of the dashboard you can see the “Add trigger” button. Click on that



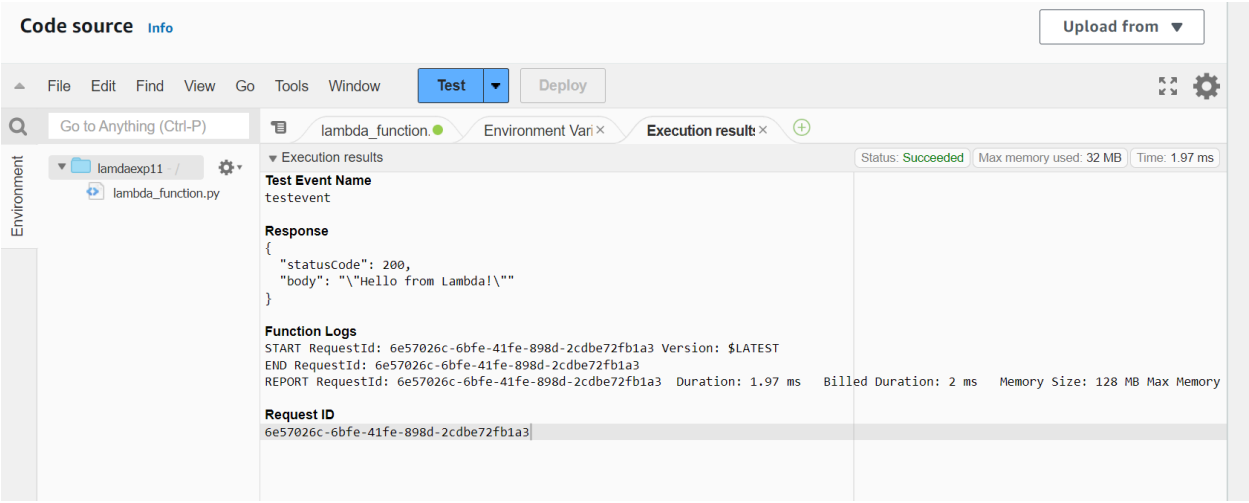
10. It will lead you to the trigger configuration tab; Where you have to select the service and the bucket you created. Add the required configuration information and then save.



11. Here you can see we have the confirmation message as well the the s3 bucket added to our triggers



12. Test the code by clicking on the Test tab ; Here as you can see our code ran successfully

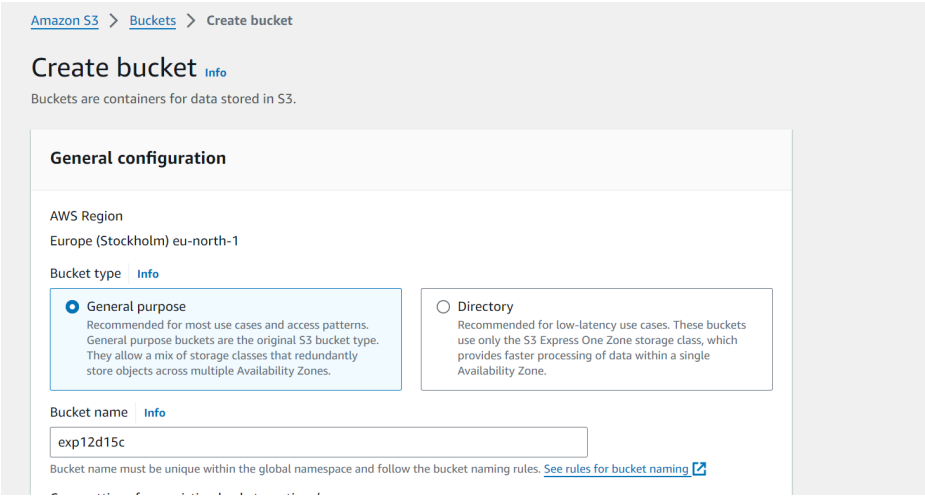


Conclusion: In conclusion, the experiment successfully demonstrated the integration of an S3 bucket with an AWS Lambda function as a trigger. By creating the S3 bucket and configuring it to invoke the Lambda function upon object uploads, we established a seamless workflow for automated processing.

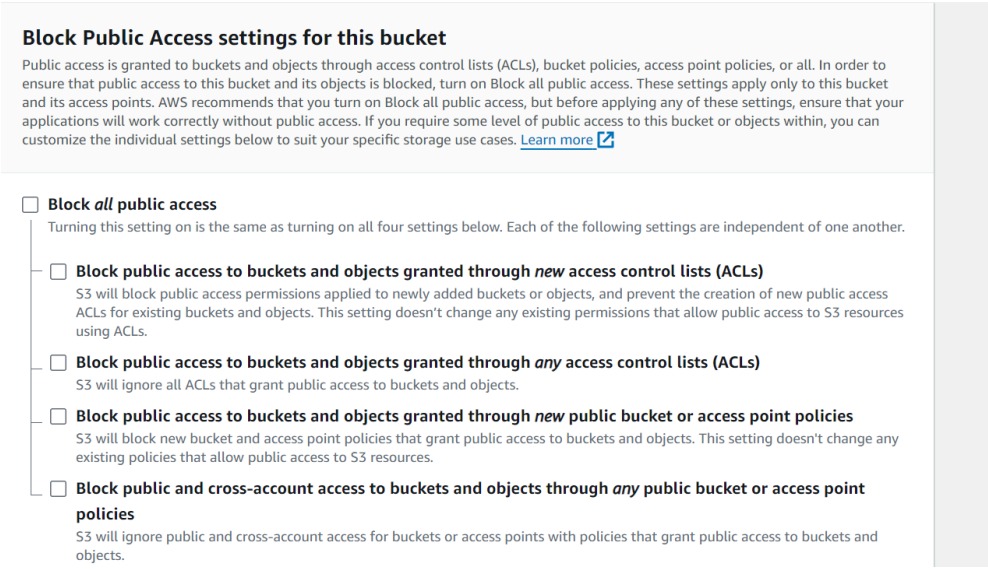
Aim: To create a Lambda function which will log “An Image has been added” once you add an object to a specific bucket in S3

STEPS:

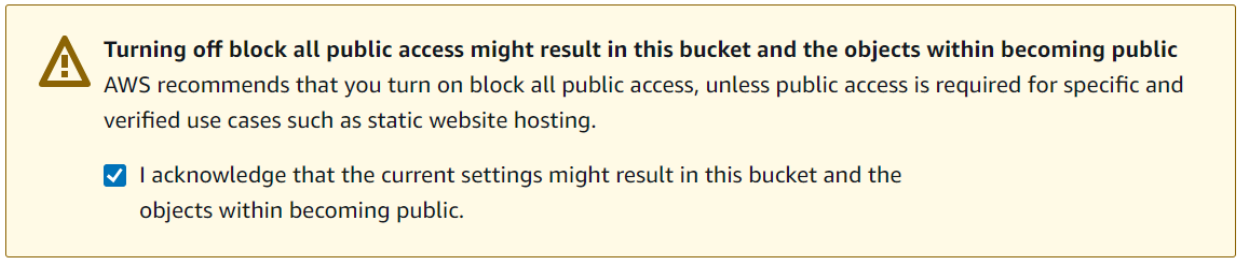
1. Create a S3 bucket and give it a bucket name



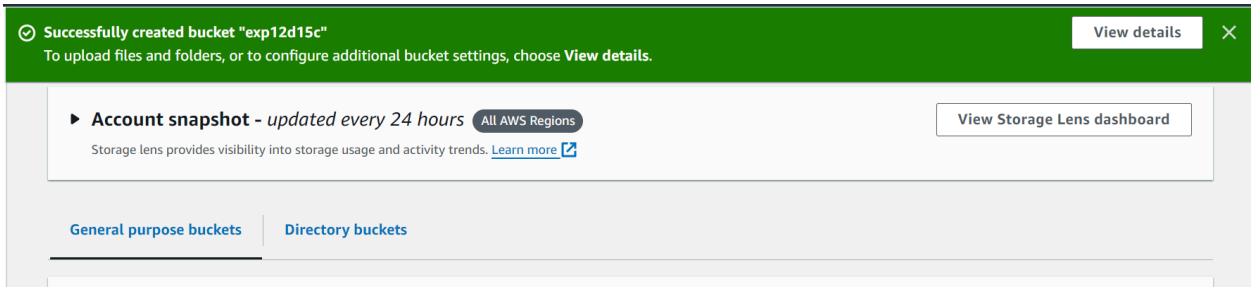
2. Allow public access to the bucket as we are going to add this bucket as a trigger for our lamda function



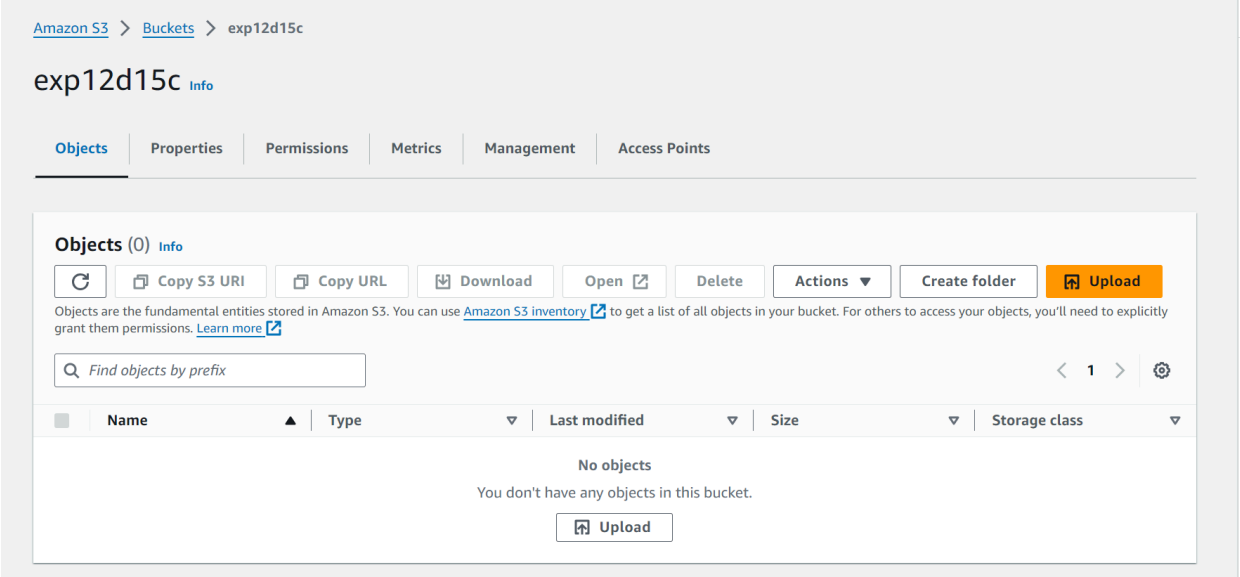
3. Give confirmation that you want to allow full public access and create the bucket



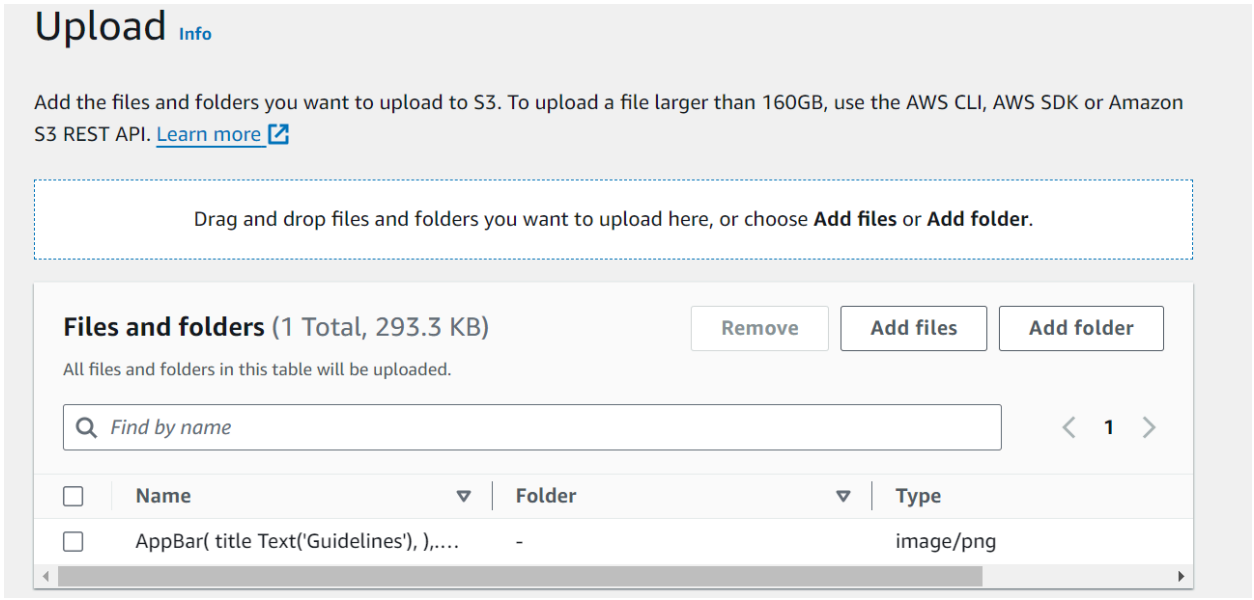
4. You will see the confirmation that the bucket is created successfully



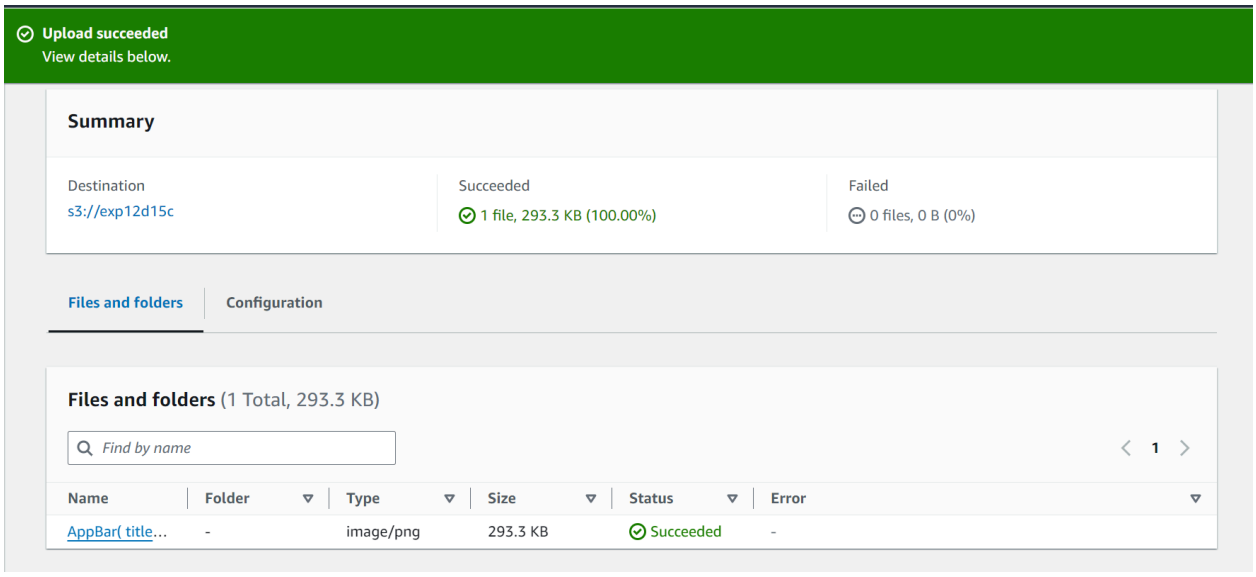
5. Now we need to upload something in the bucket so click on the upload button and add a file



6. I have added a .png extension file; You can upload a .txt file as well

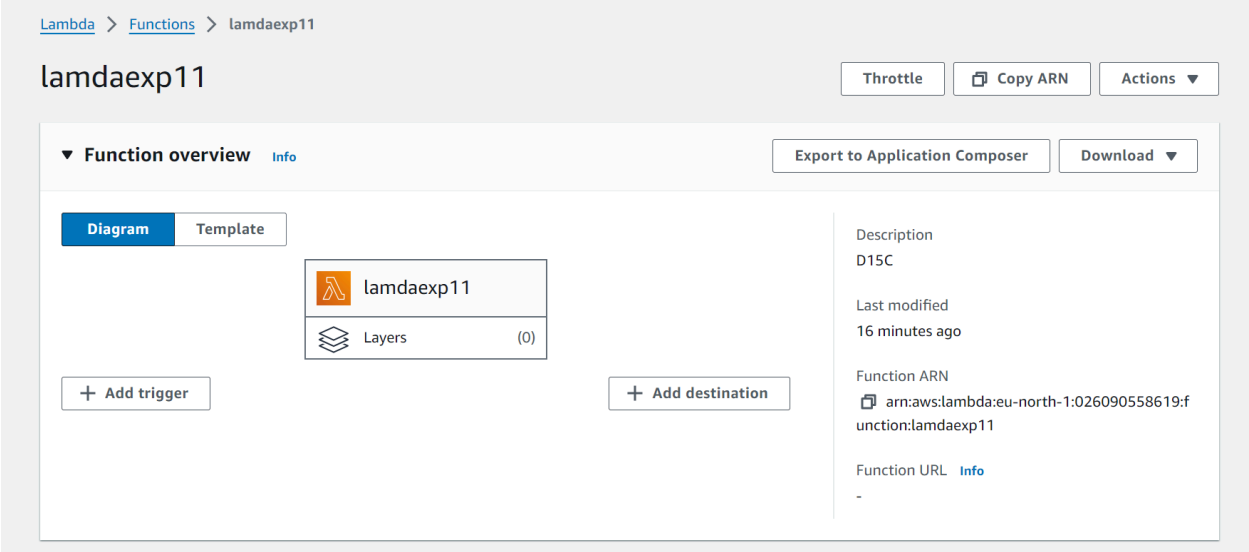


7. Here you can see the confirmation that the upload was a success

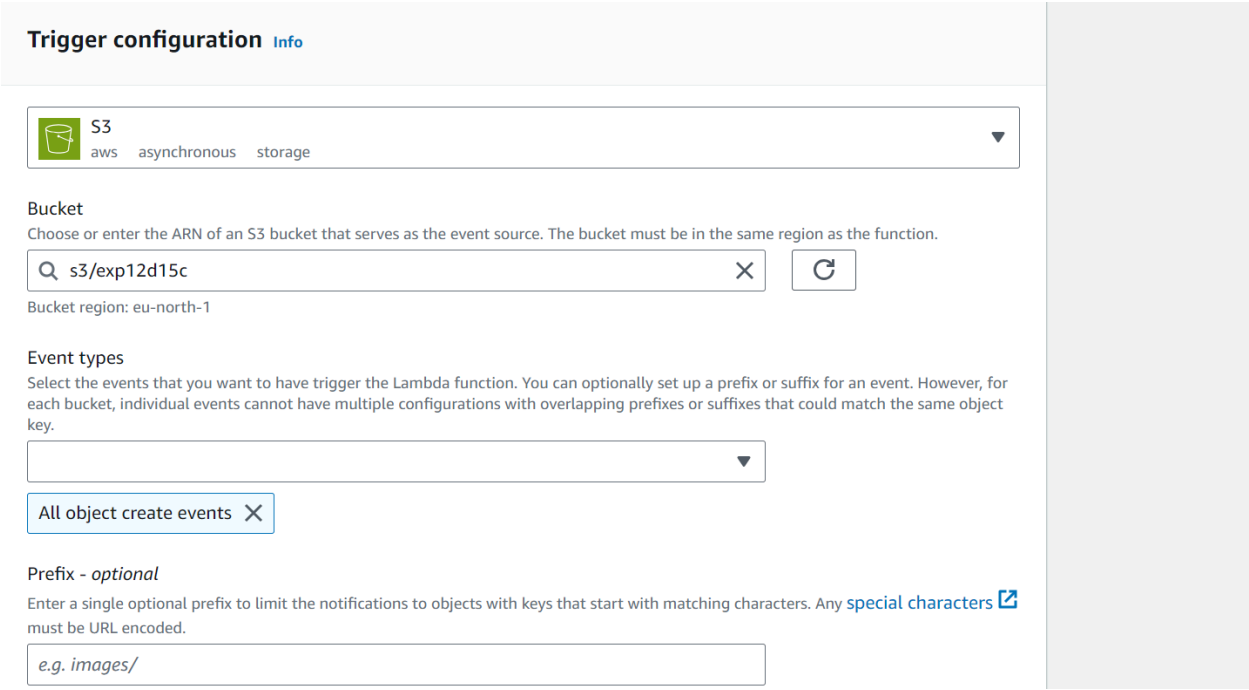


8. Now go back to the aws dashboard and search for lamda function service, Open the function we created in experiment 10. We are going to add this bucket as a trigger to this function

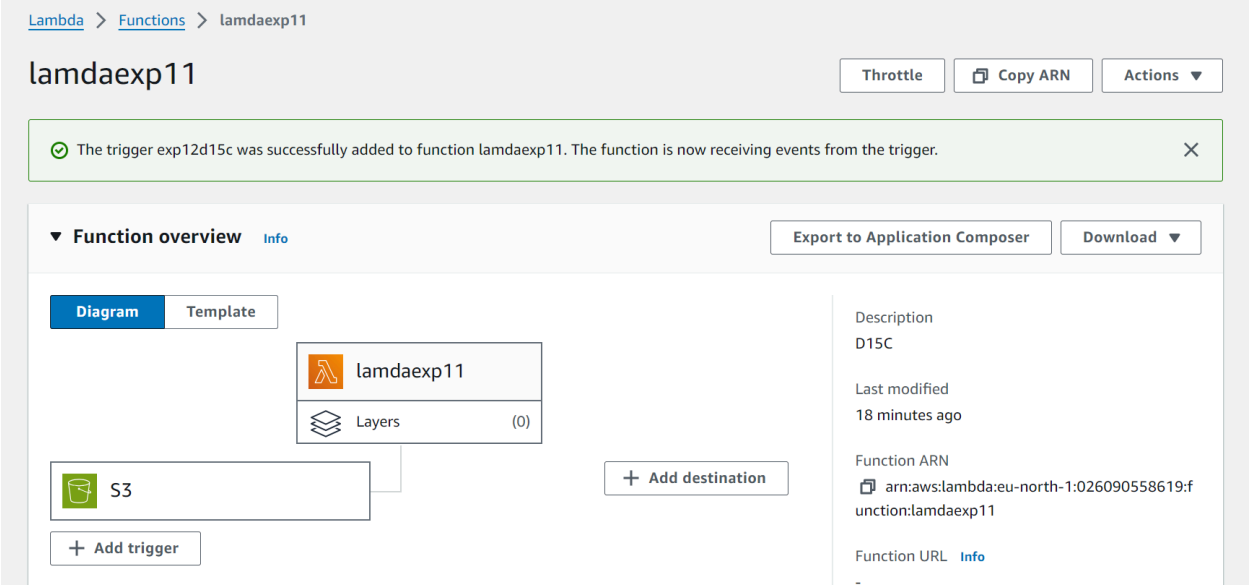
9. On the function overview section of the dashboard you can see the “Add trigger” button. Click on that



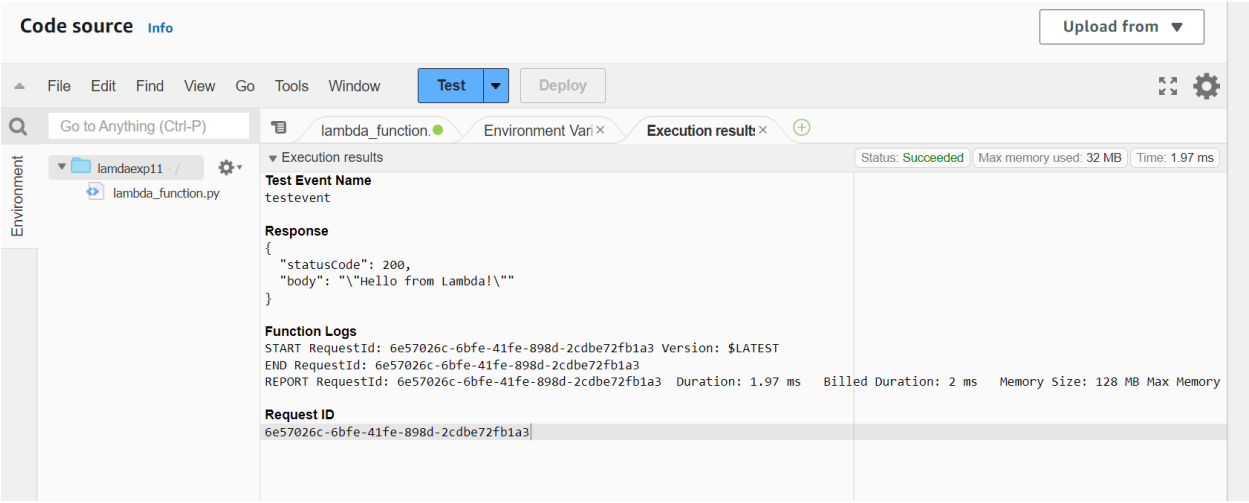
10. It will lead you to the trigger configuration tab; Where you have to select the service and the bucket you created. Add the required configuration information and then save.



11. Here you can see we have the confirmation message as well the the s3 bucket added to our triggers



12. Test the code by clicking on the Test tab ; Here as you can see our code ran successfully



Conclusion: In conclusion, the experiment successfully demonstrated the integration of an S3 bucket with an AWS Lambda function as a trigger. By creating the S3 bucket and configuring it to invoke the Lambda function upon object uploads, we established a seamless workflow for automated processing.