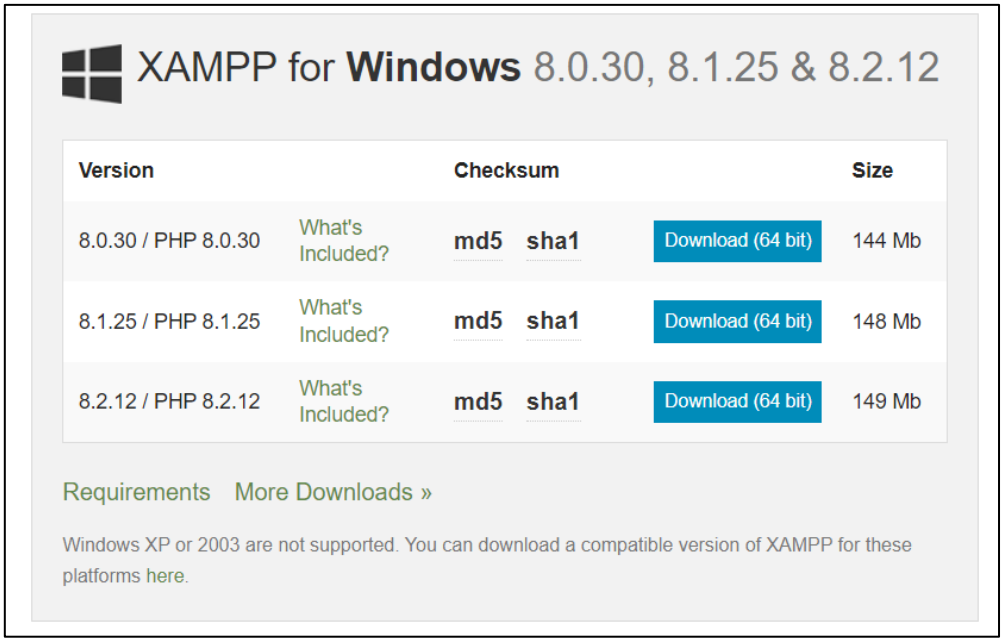
Advance DevOps

Experiment 1a

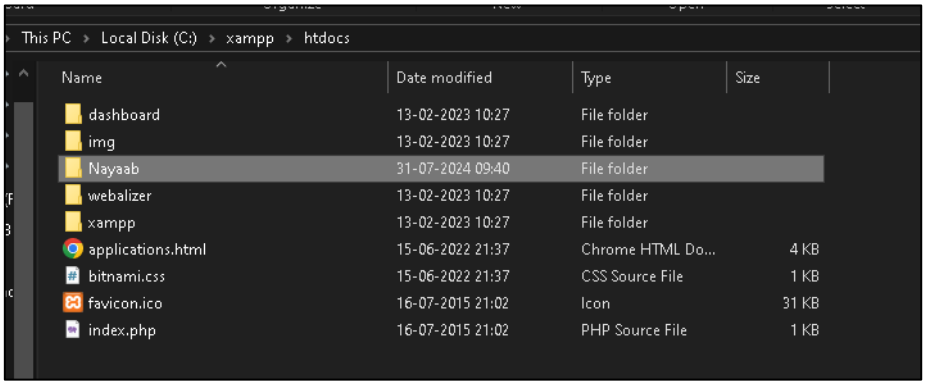
Part 1: To develop a website and host it on your local machine on a VM

Following are the steps to be followed:

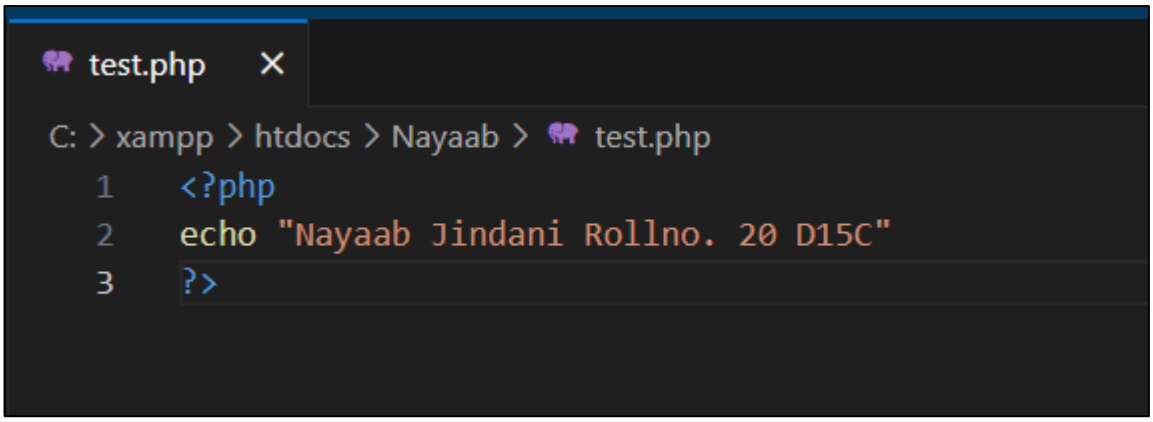
1. First step is to download XAMPP through the official website, https://www.apachefriends.org/download.html and then complete the installation process.



2. To run a php script go to C:\xampp\htdocs and create a folder (here my folder is named Nayaab).

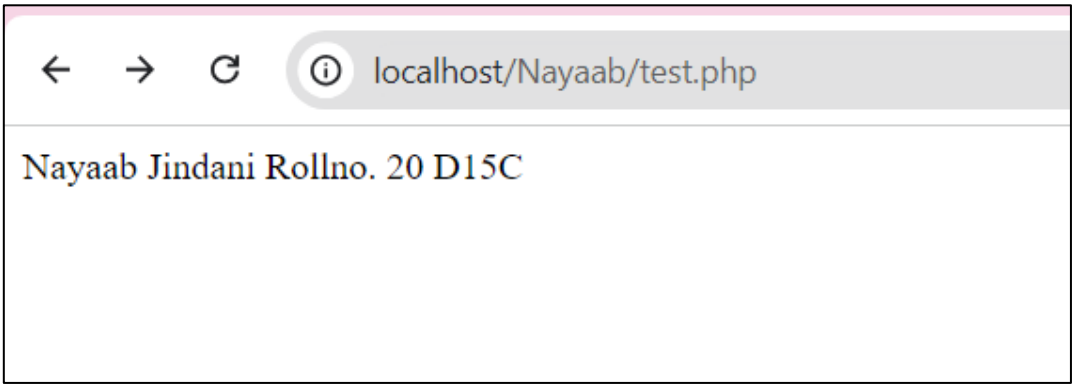


3. Inside the folder create a file and save it as filename.php and write php script inside this file.



4. Now go to XAMPP control panel and start Apache server.

5. Now type http://localhost/folder_name/filename.php in your browser and you will get to see the output.



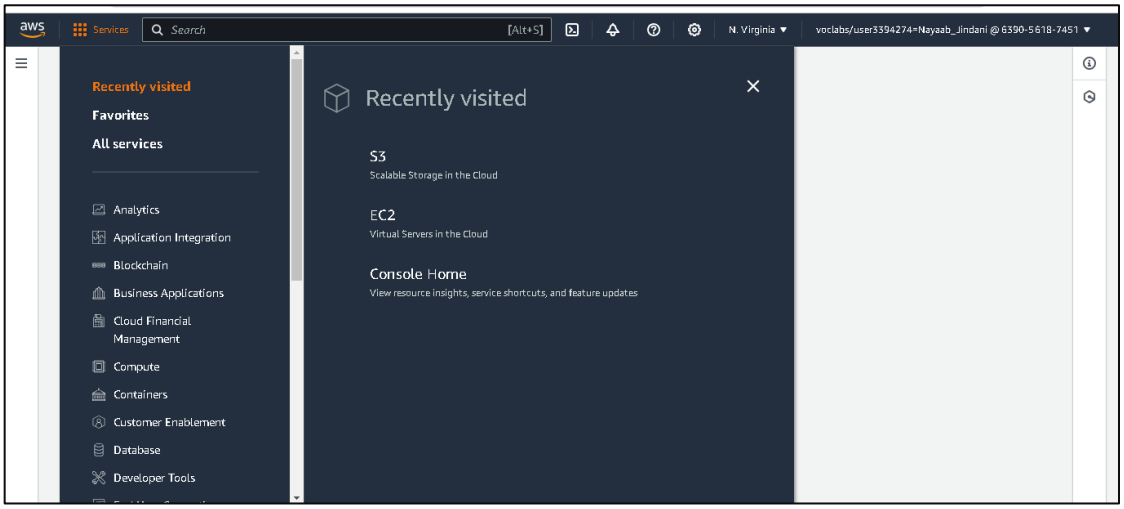
Advance DevOps

Experiment 1a

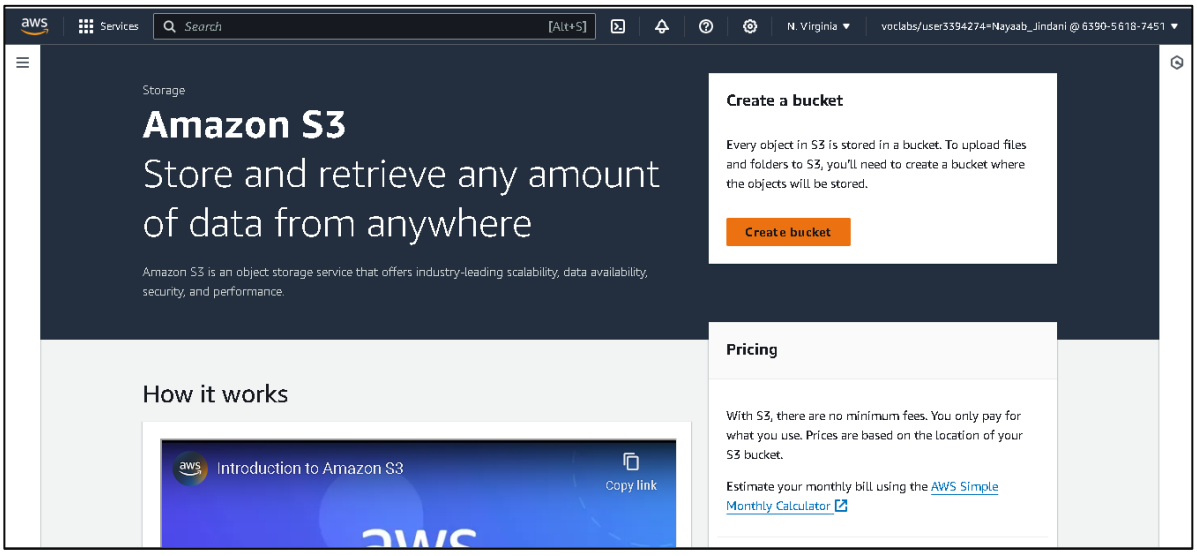
Part 2: Hosting a static website on Amazon S3

Steps:

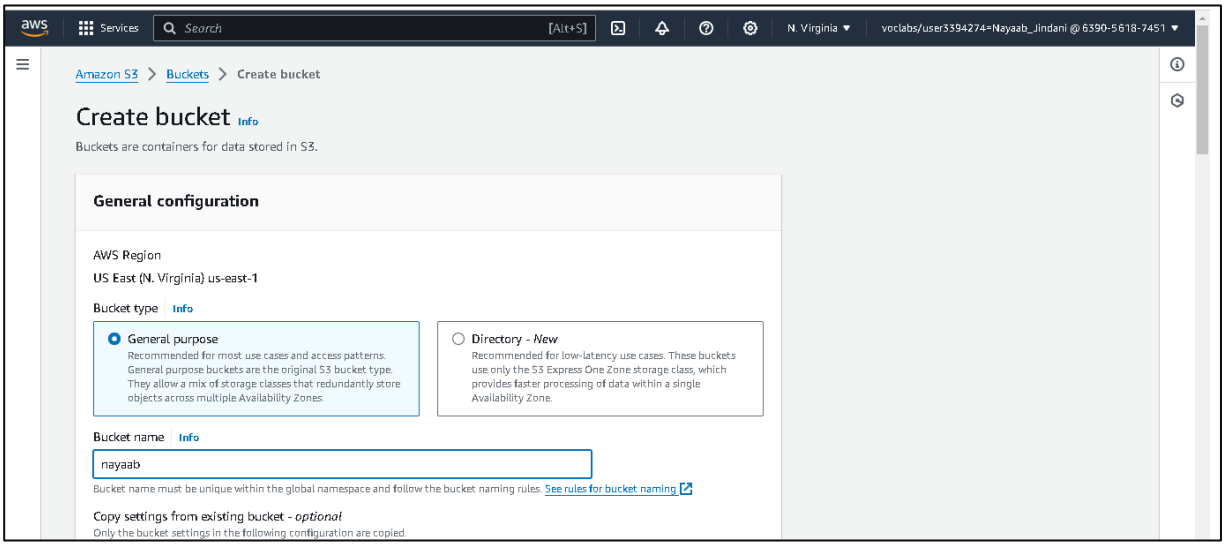
1. Go to AWS console home -> services and select S3.



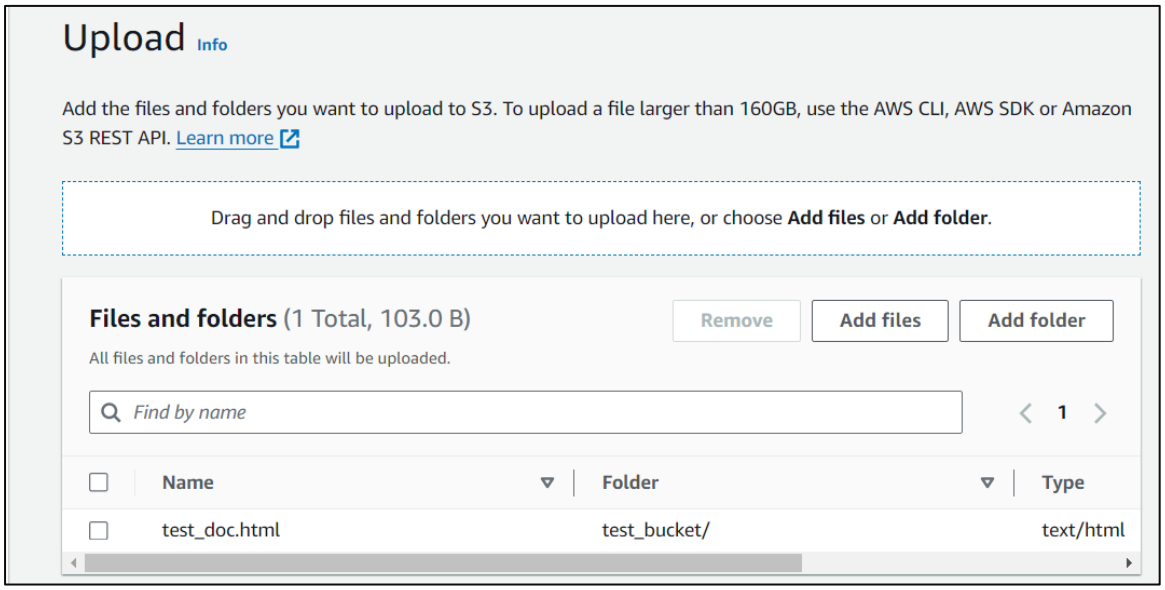
2. Select create a bucket



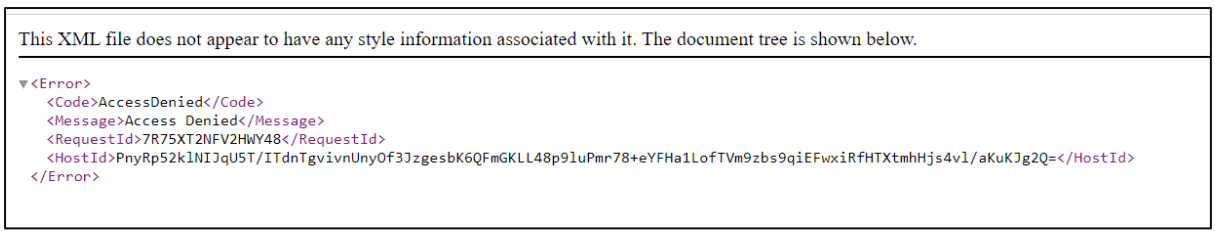
3. Enter bucket name and select create bucket.



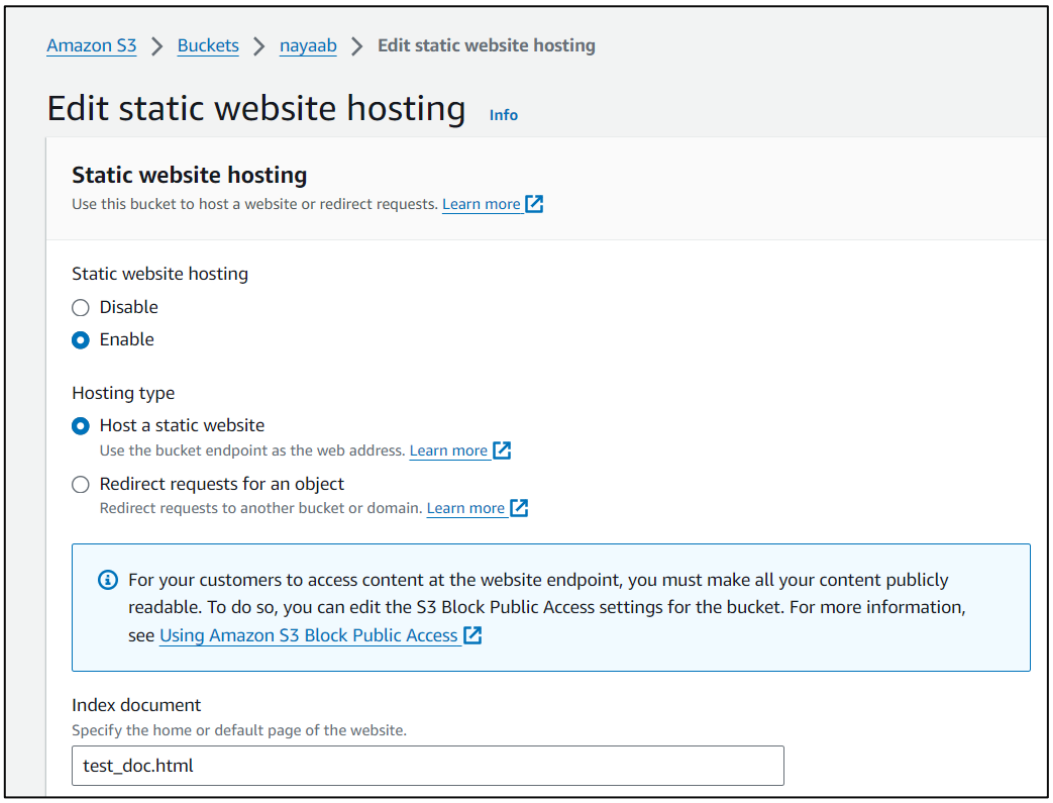
4. Now upload the folder or files of the website to be hosted.



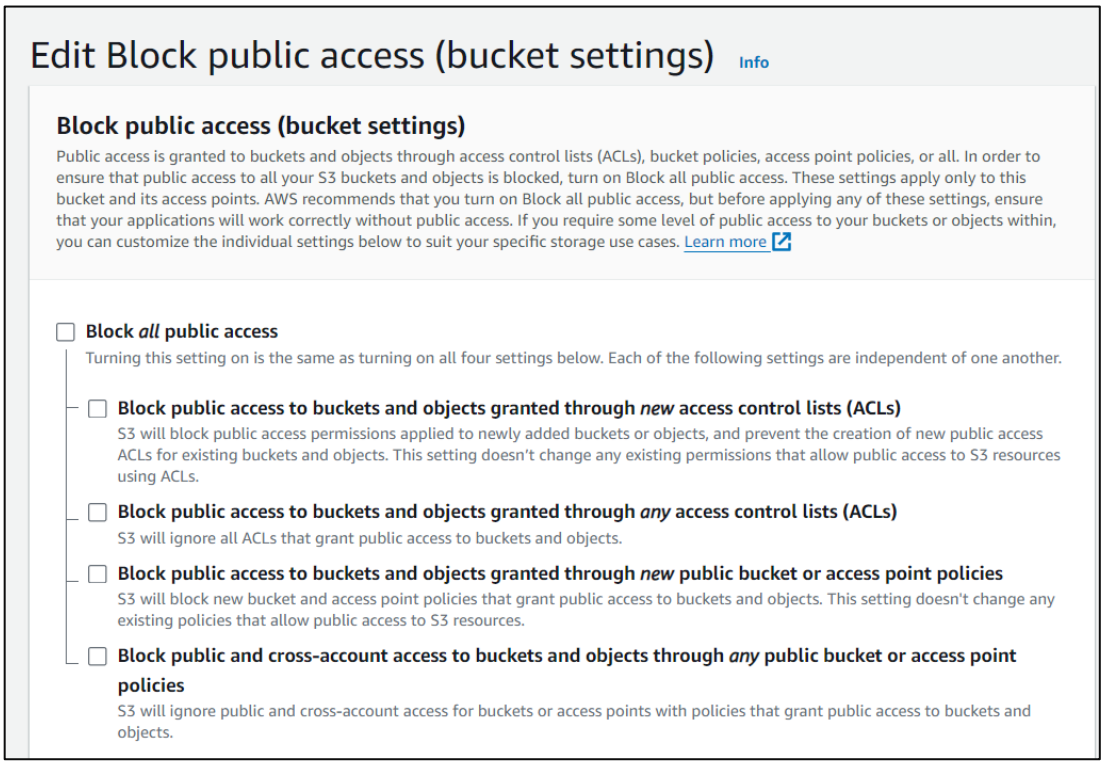
5. After uploading the folder or files when you select the object URL of your file it will show access denied:



- To provide access we need to enable static website hosting and to do so go to buckets -> your bucket -> properties -> static website hosting and enable static website hosting. Also Specify the home or default page of the website. Then click on save changes.

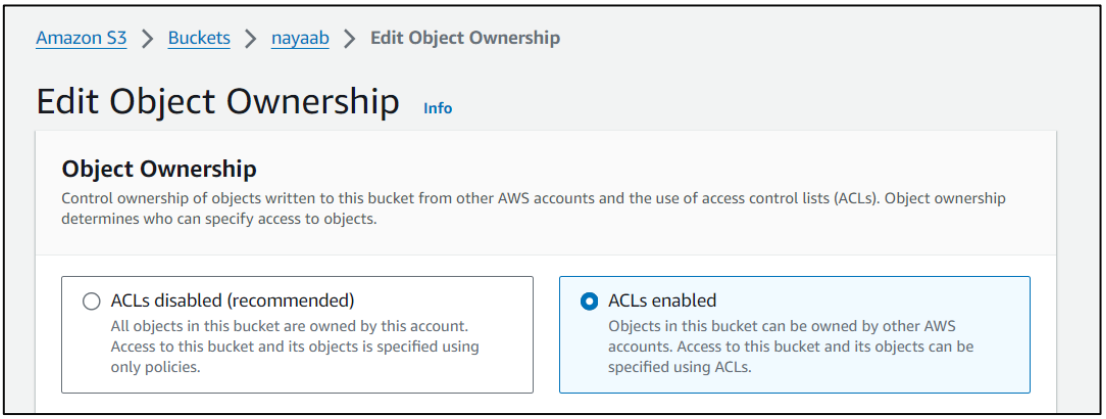


- Now select the permission tab which is besides properties and uncheck block all public access.

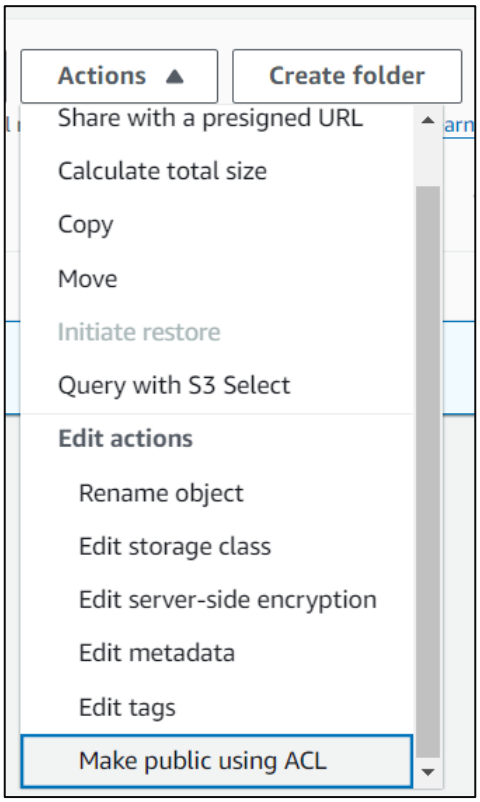


- Now we have to make all objects related to our website publicly accessible and that can be done in 2 ways: either we can select each object and provide permission or use the bucket policy and provide permission for all files at once.

  To select a file and give permission first go to permissions and enable ACL.



Now select the objects related to your website click on actions and then click on make public using ACL. Select make public.



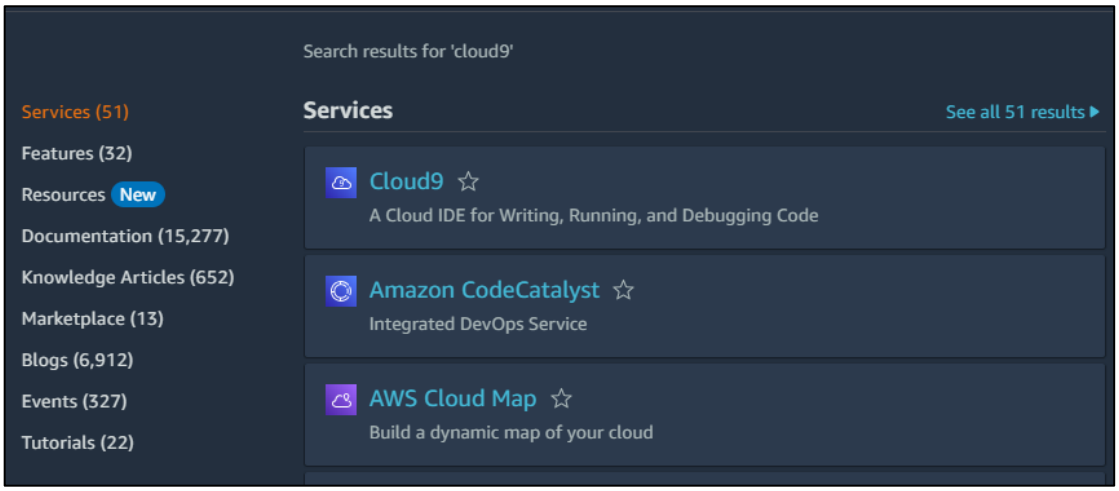
6. Now when you open the URL you will be able to see the website.

Advance DevOps

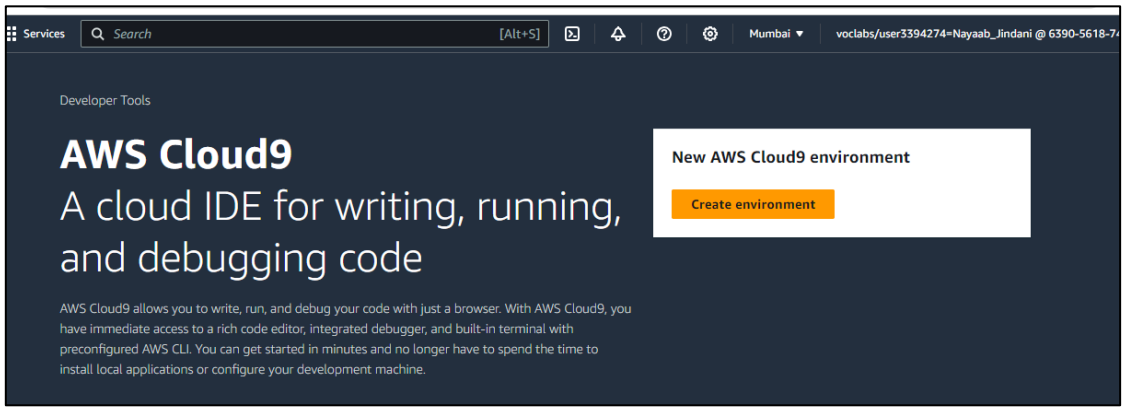
Experiment 1b

Steps to setup AWS Cloud9 IDE, Launch AWS Cloud9 IDE and perform collaboration:

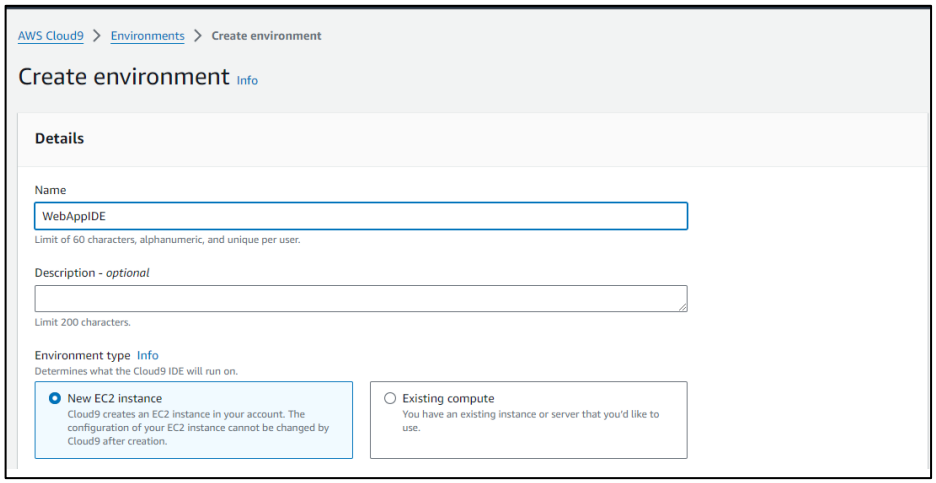
1. Login in your AWS account. Search Cloud9 in services.



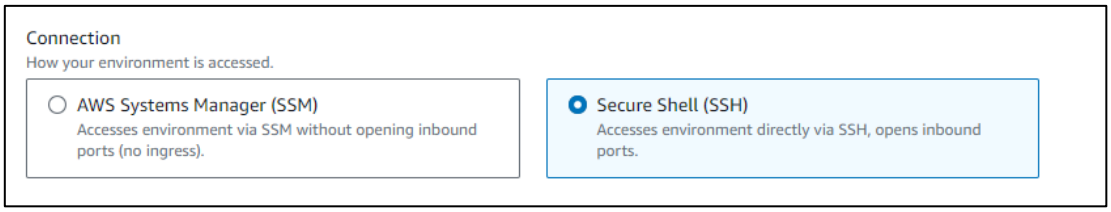
2. Select create environment.



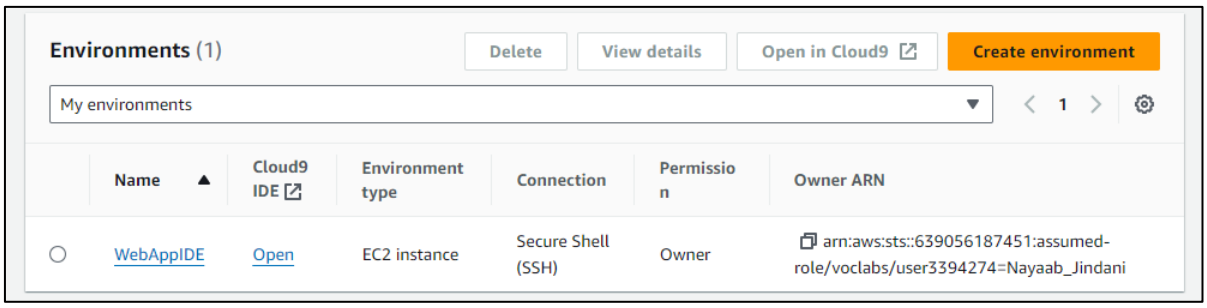
3. Provide the name, description (optional).



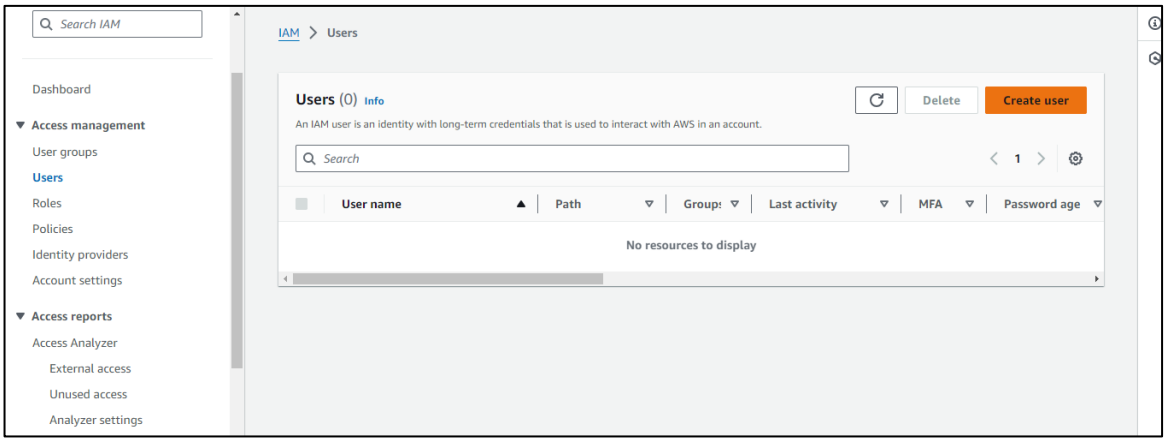
4. Select Secure Shell (SSH) as the connection type and click on create.



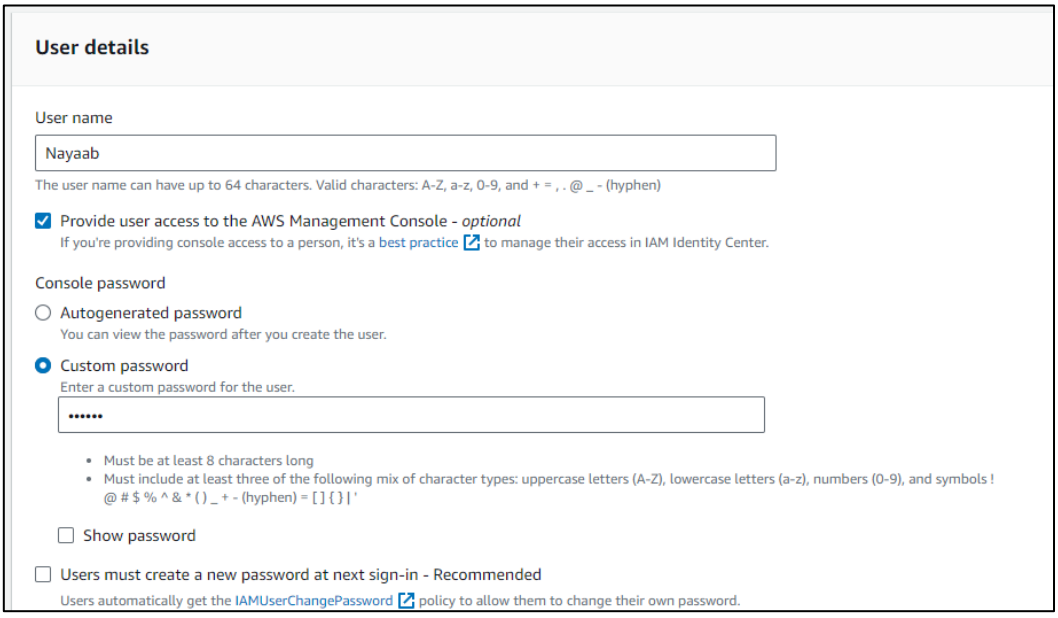
5. Your Cloud9 IDE has been created.



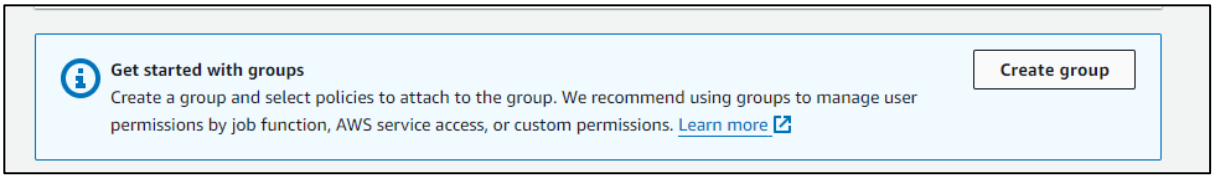
6. Now we will create user. For that go to IAM -> Users -> Create user.



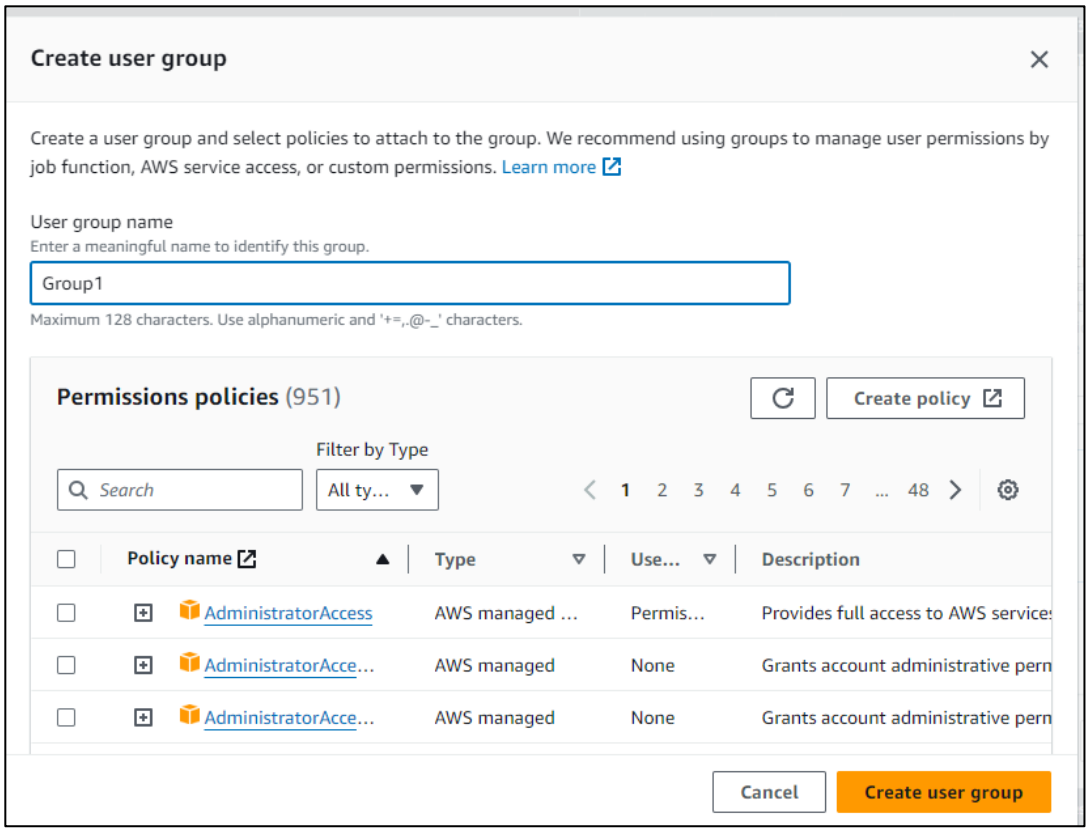
7. Enter user details like user name, password and click on next.



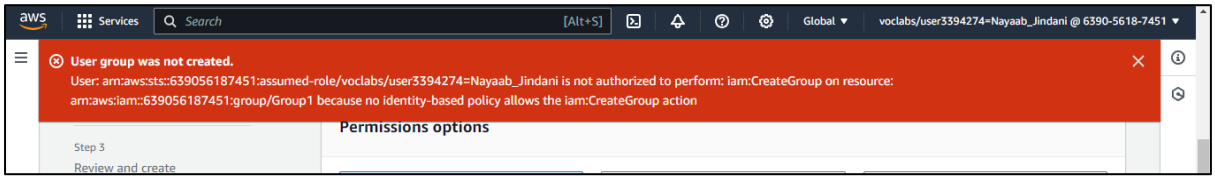
8. Select create group in the next section.



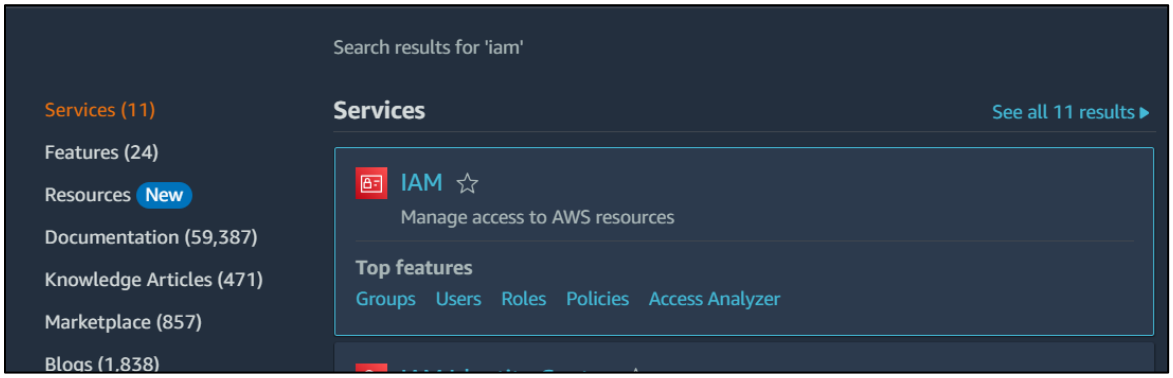
9. Enter group name, select permission policies and select create user group.



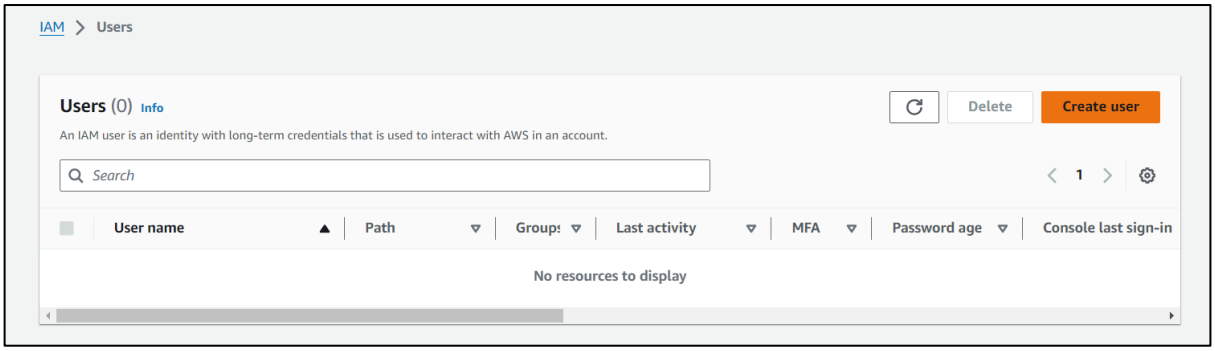
10. Due to authorization issues the group was not created and the below given error was displayed.

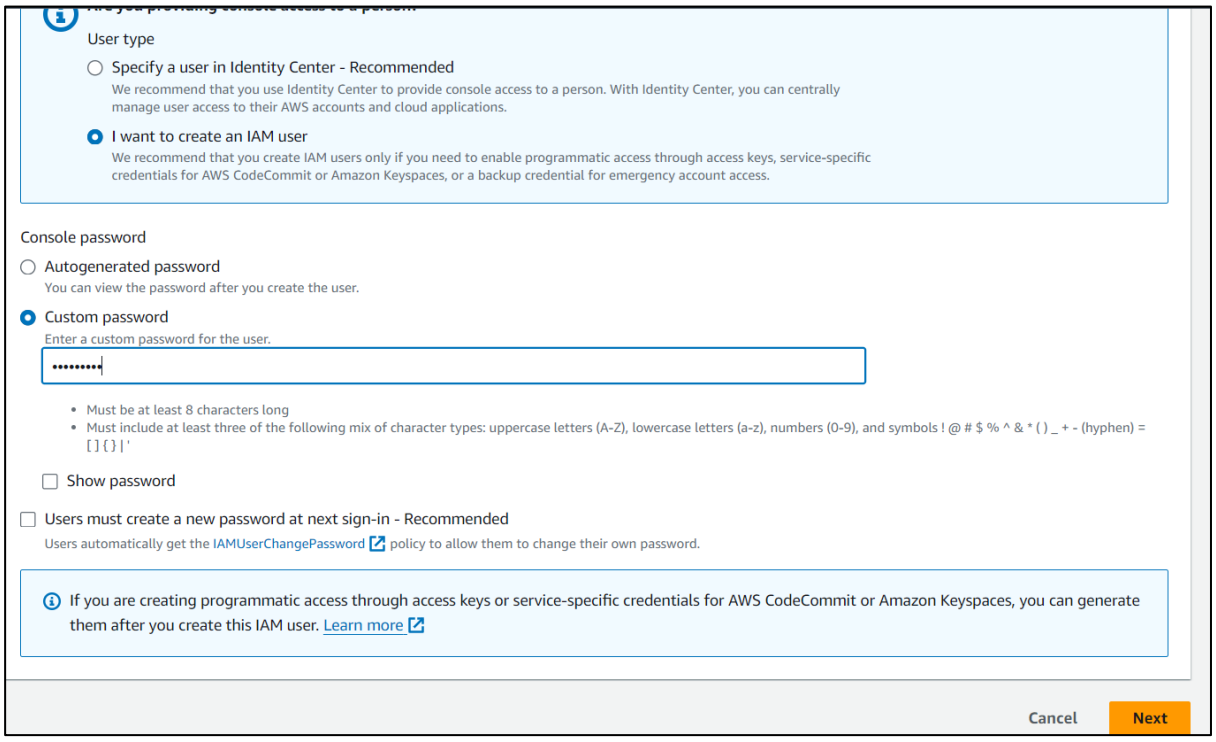


11. Now we will sign in to the AWS account to continue with the further procedure and sign out from the console we were working on earlier. After logging in search for IAM in services and click on it.

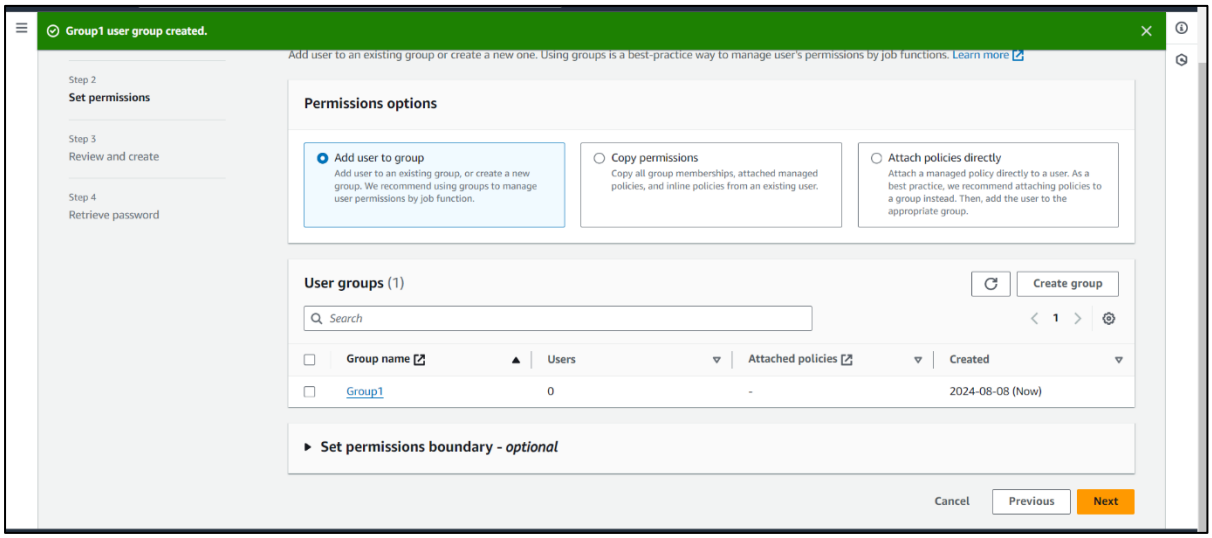


12. Now again we will follow the same procedure to create user and group as done above.

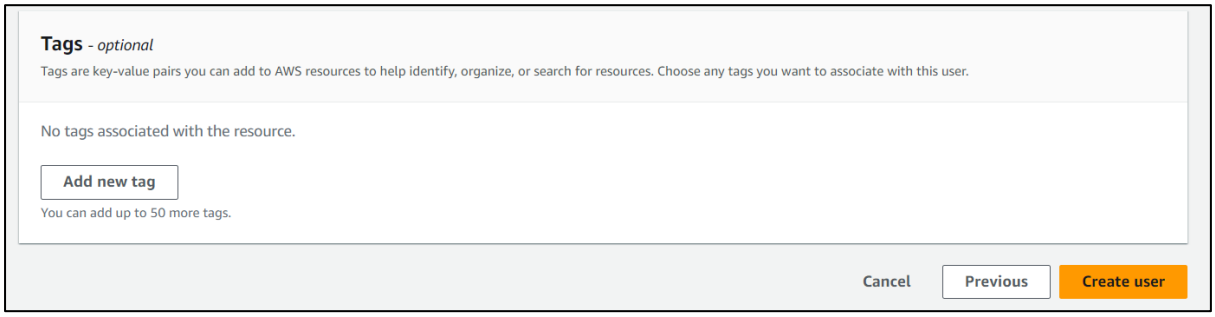




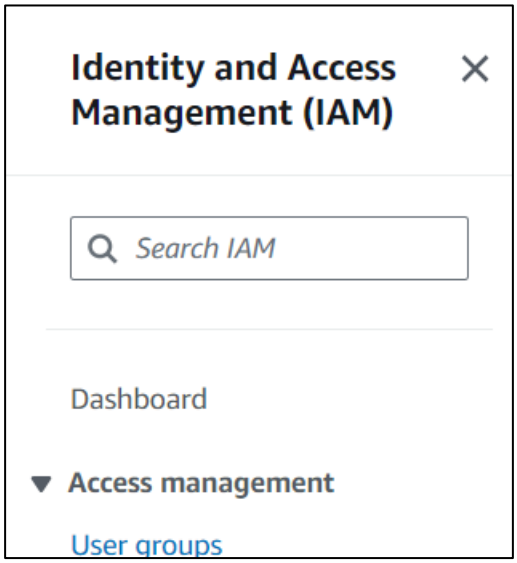
13. Select add user to group and click on next.



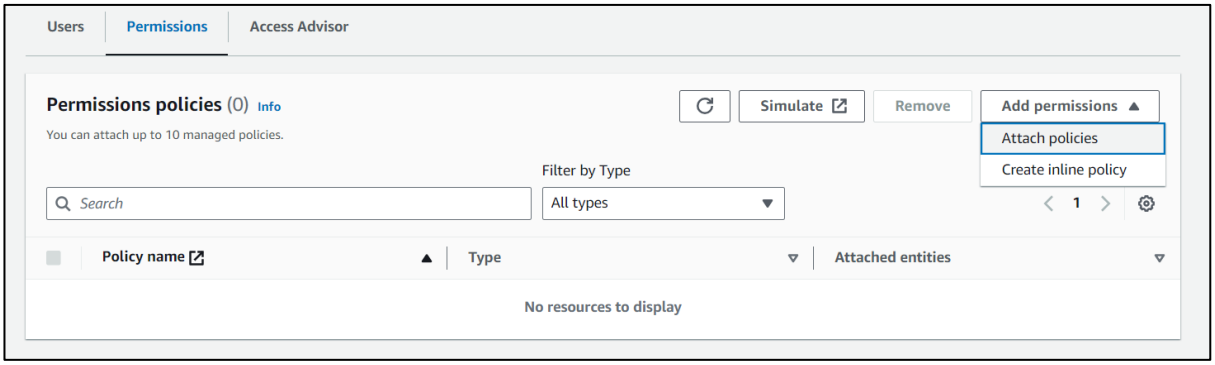
14. Add tags (optional) and select create user.



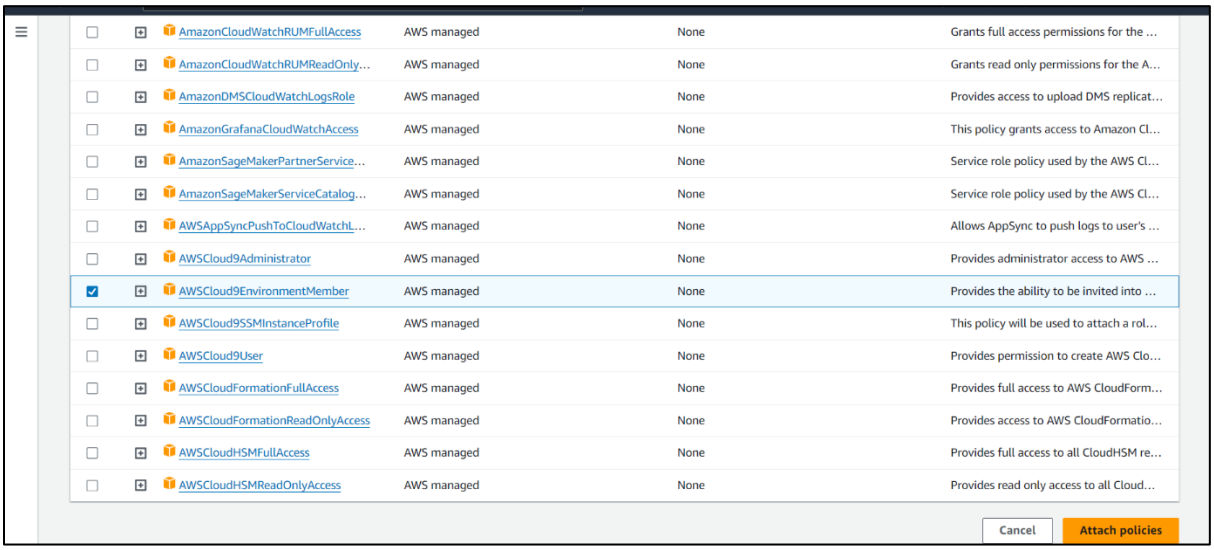
15. In the left pane select user groups.



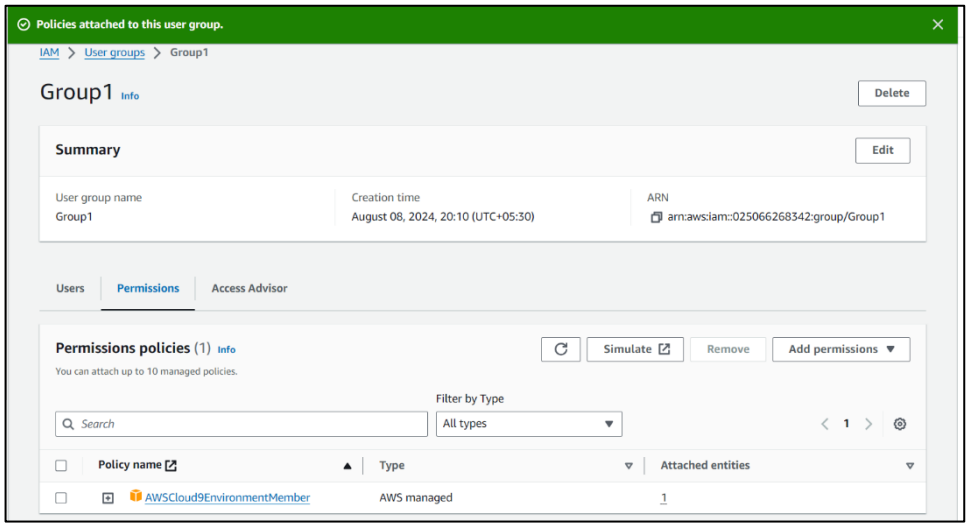
16. Click on the group you created and go to permissions tab and select attach policies from add permissions.



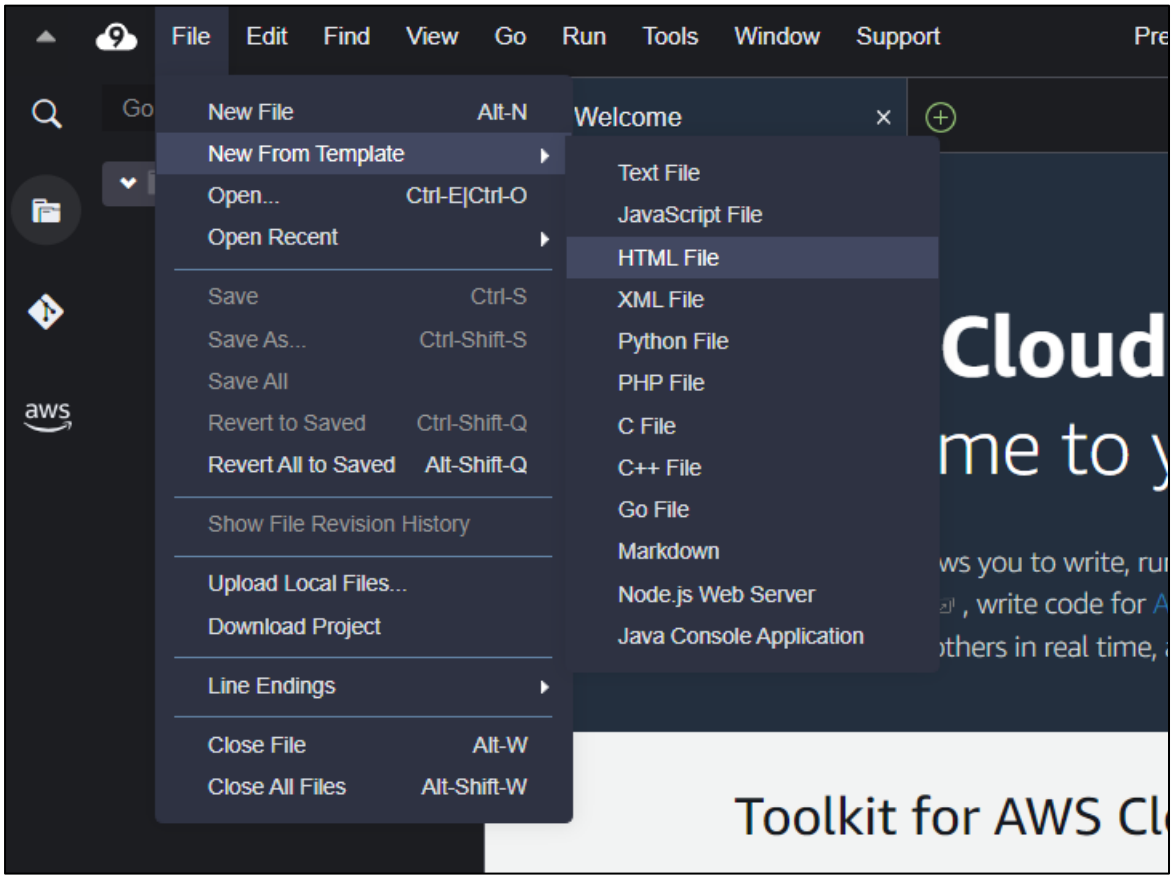
17. Attach the policy AWSCloud9EnvironmentMember by selecting it and click on attach policies.



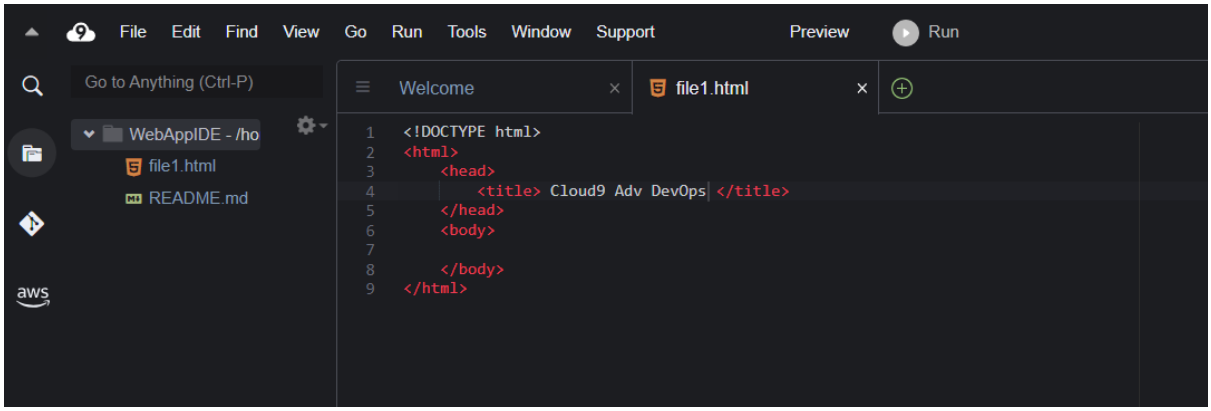
18. Policy has been successfully attached to the user group



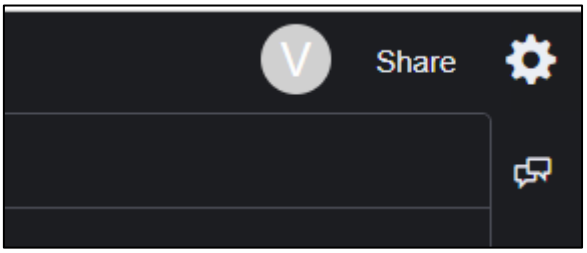
19. Now we will open the Cloud9 IDE we created earlier. Then go to file -> new from template -> HTML file (any file type can be selected).



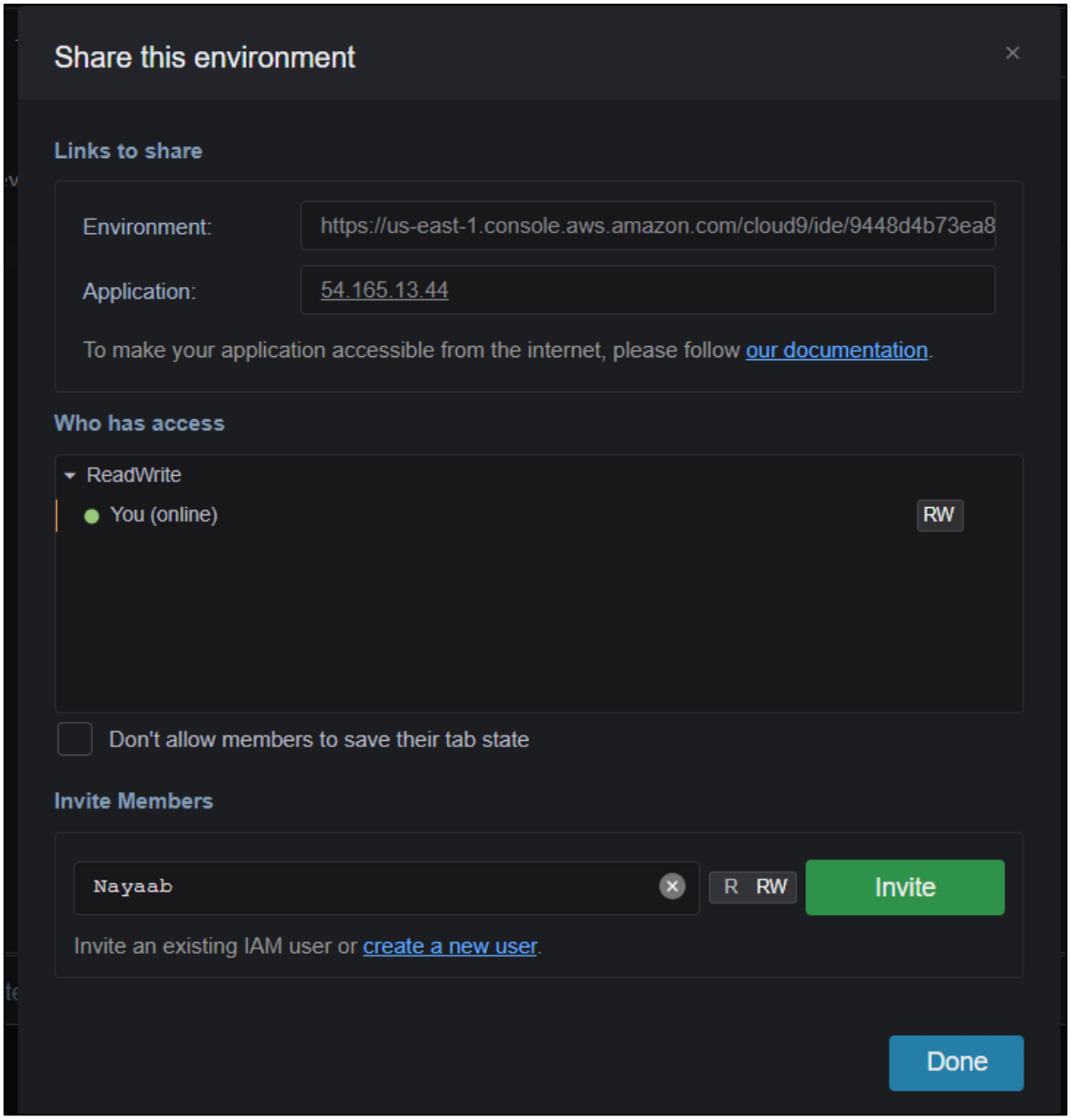
20. You can now edit the html file and save it.



21. We can collaborate with other members by sharing this file. To do so click on the share option to the top right of the screen.



22. Here you can give read or read/write permission to your team members and click on invite.
    Your teammate will get an invite in “shared with you” and after he/she selects open IDE they will be able to see the same interface as yours and now you and your team members can collaborate in real time.



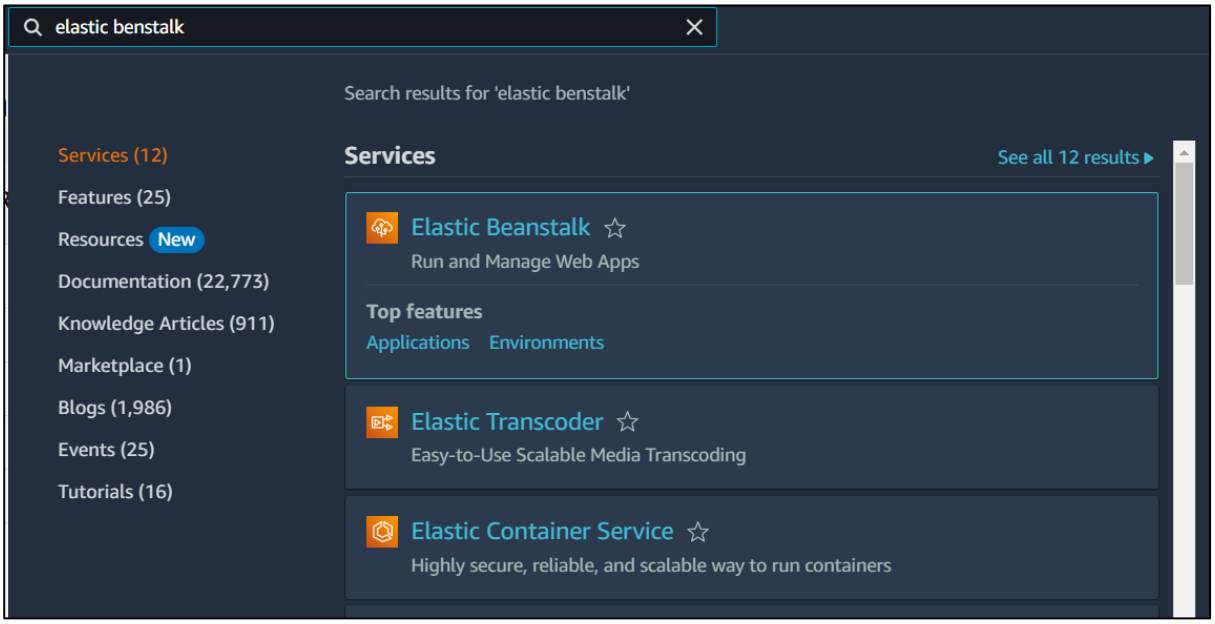
Advance DevOps

Experiment 2

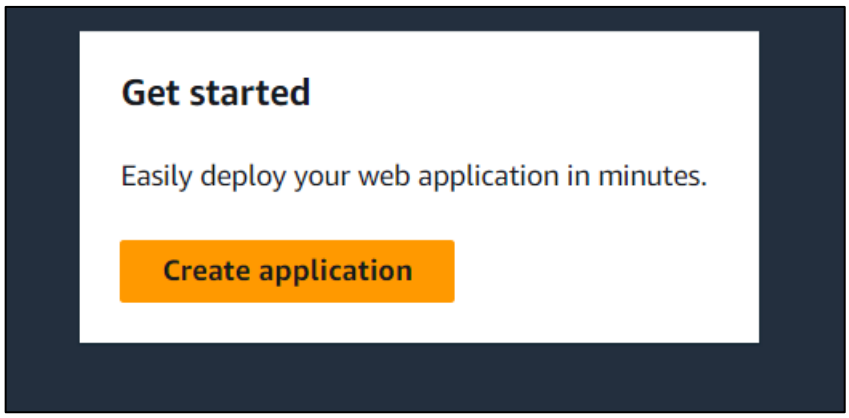
To Build Your Application using AWS CodeBuild and Deploy on S3 / SEBS using AWS CodePipeline, deploy Sample Application on EC2 instance using AWS CodeDeploy.

Steps:

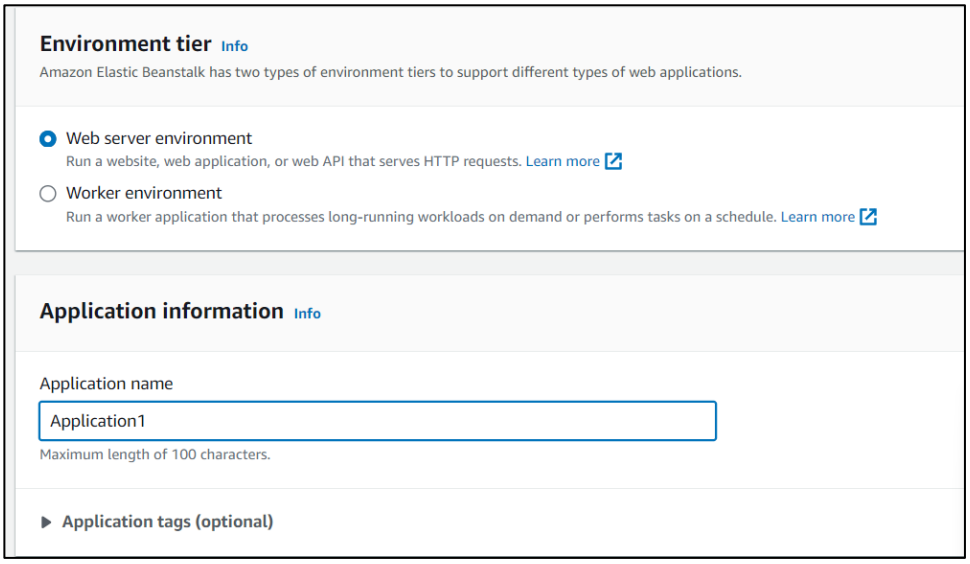
1. Firstly, we will create an environment and for that search elastic beanstalk in the services.



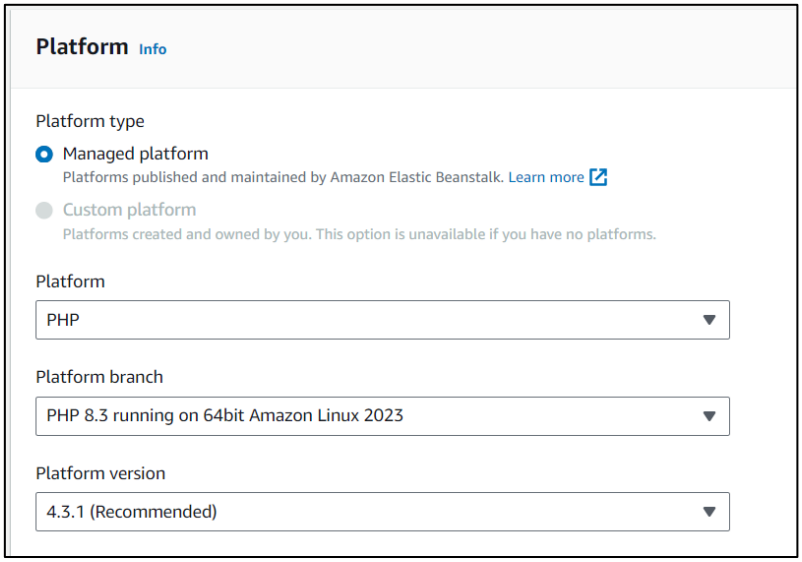
2. Select create application.



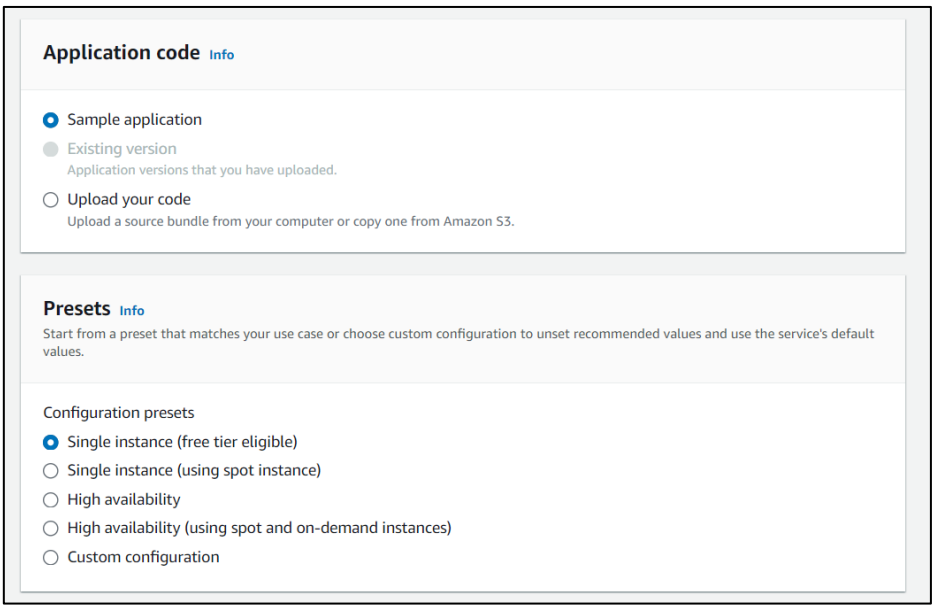
3. Now we will configure the environment. In environment tier select web server environment and give application name.



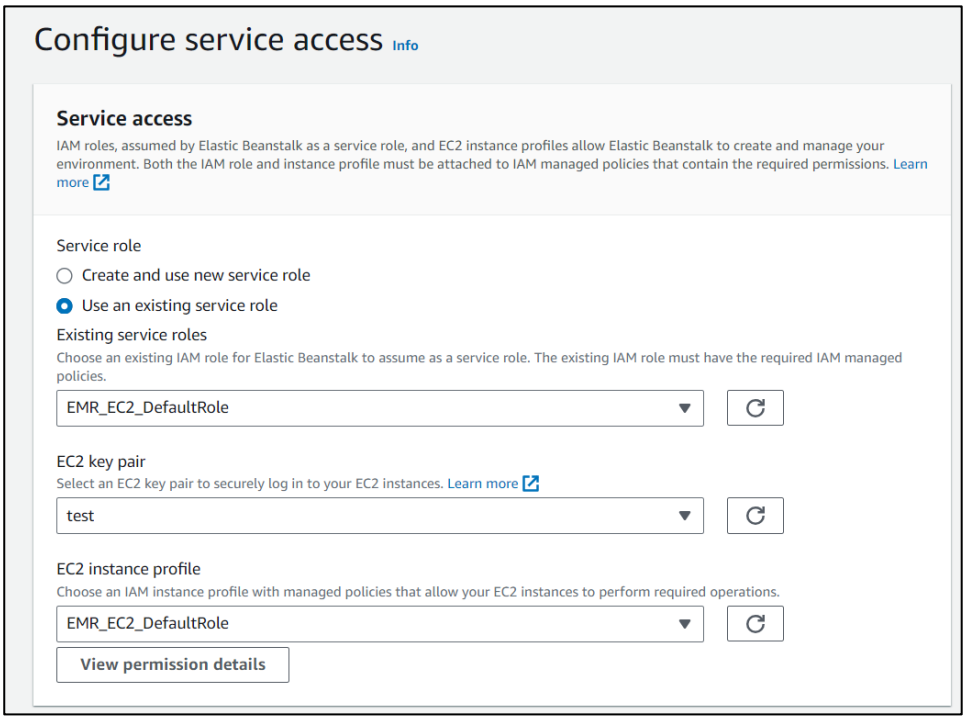
4. Select a platform, here I have selected php.



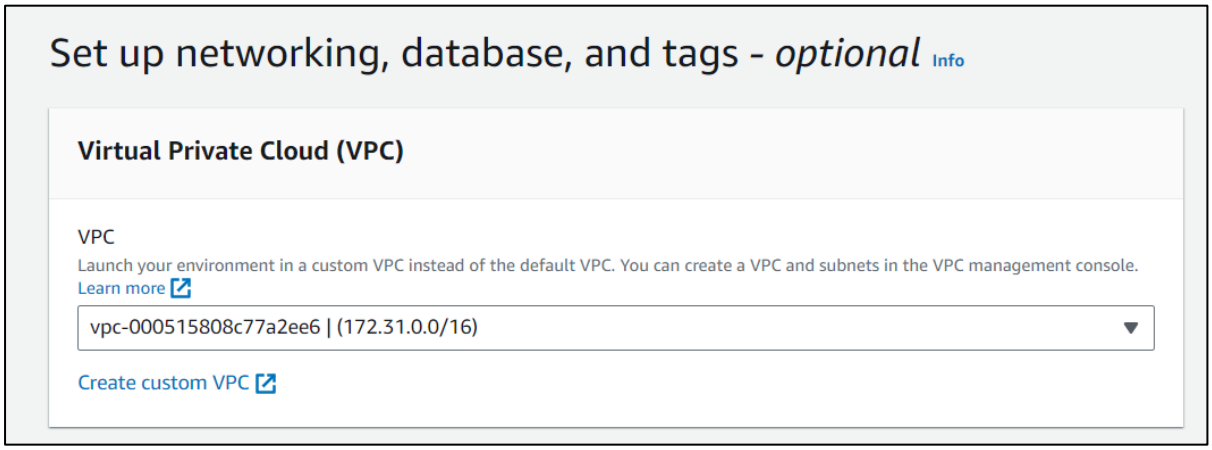
5. Keep all other settings at their default values

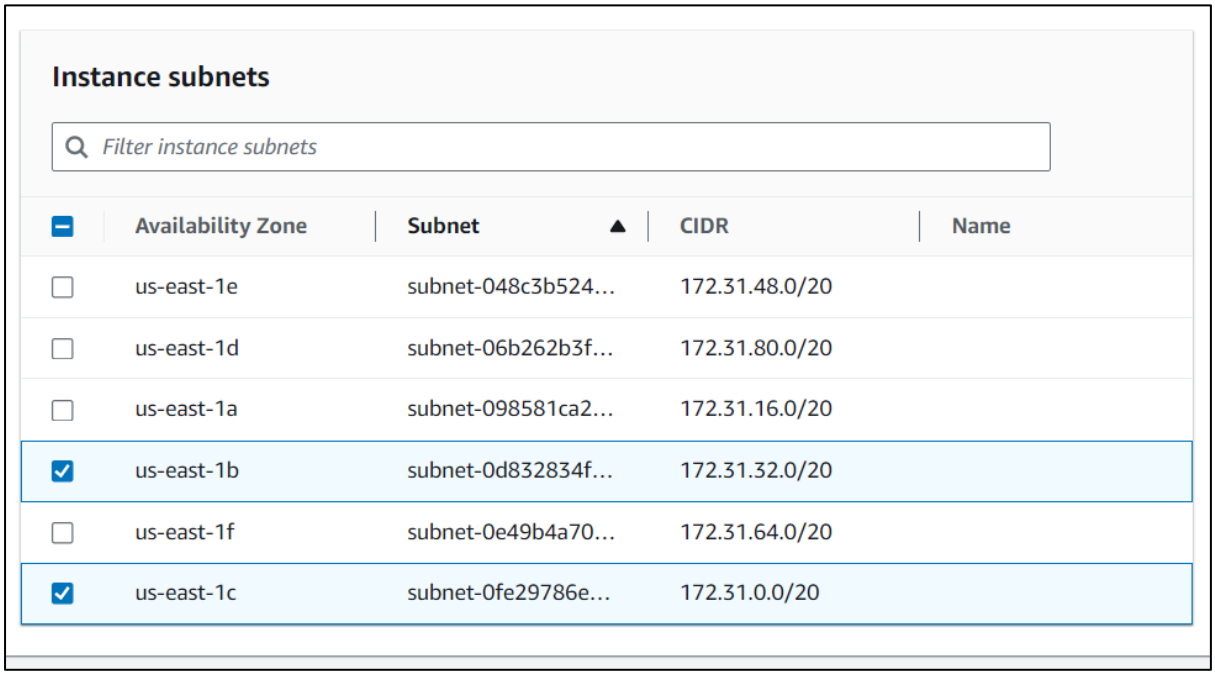


6. Now we will configure service access. Select an existing service role, EC2 key pair and EC2 instance profile from the dropdowns given.

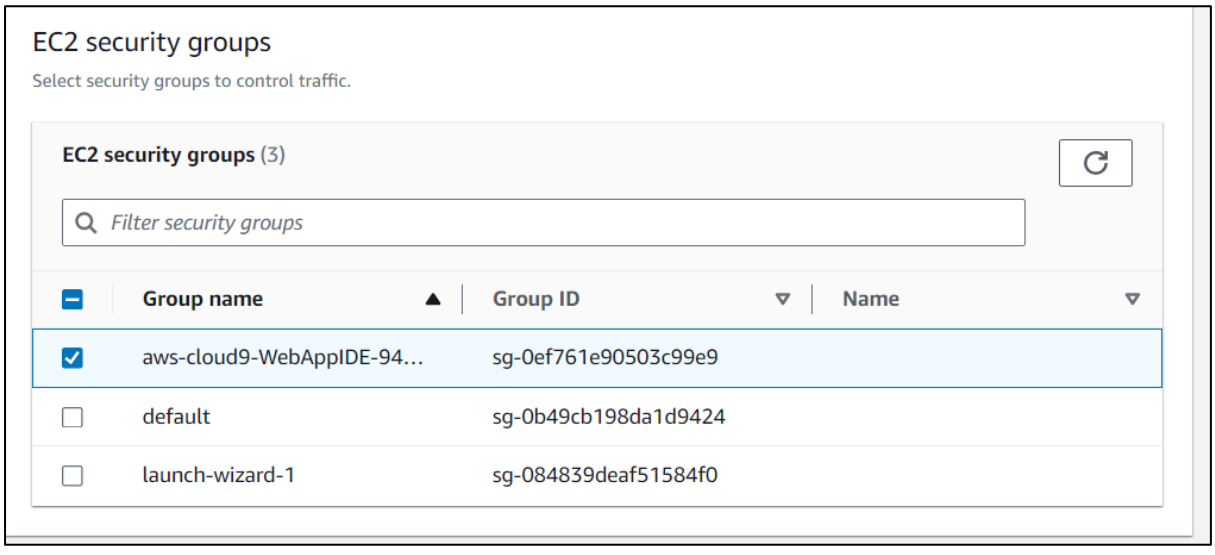


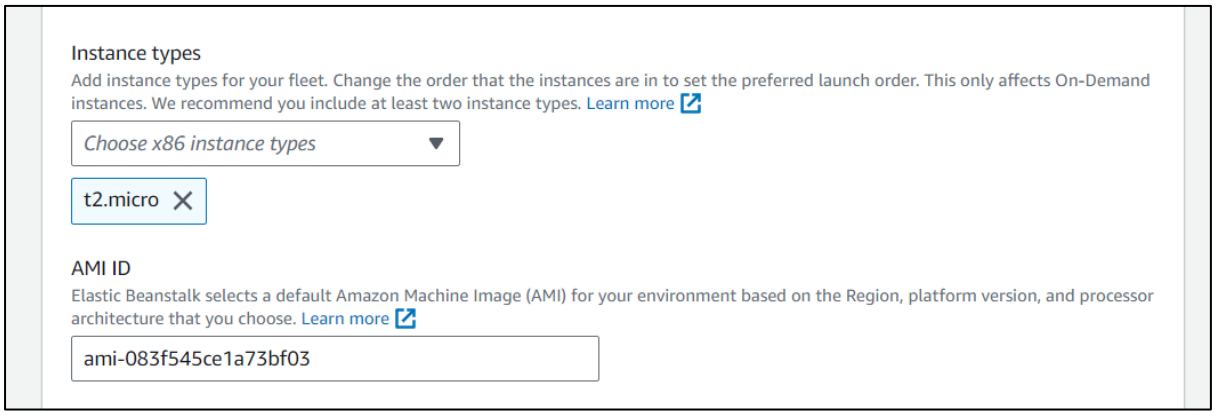
7. In the setup network section select a vpc and instance subnets then click on next.



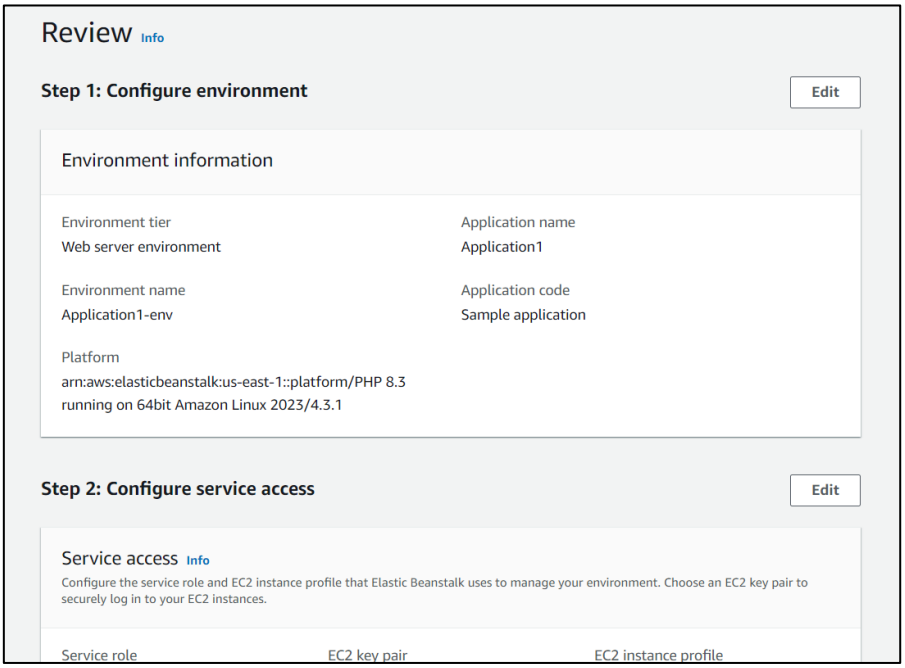


8. In the next step select EC2 security groups and set instance type as t2.micro

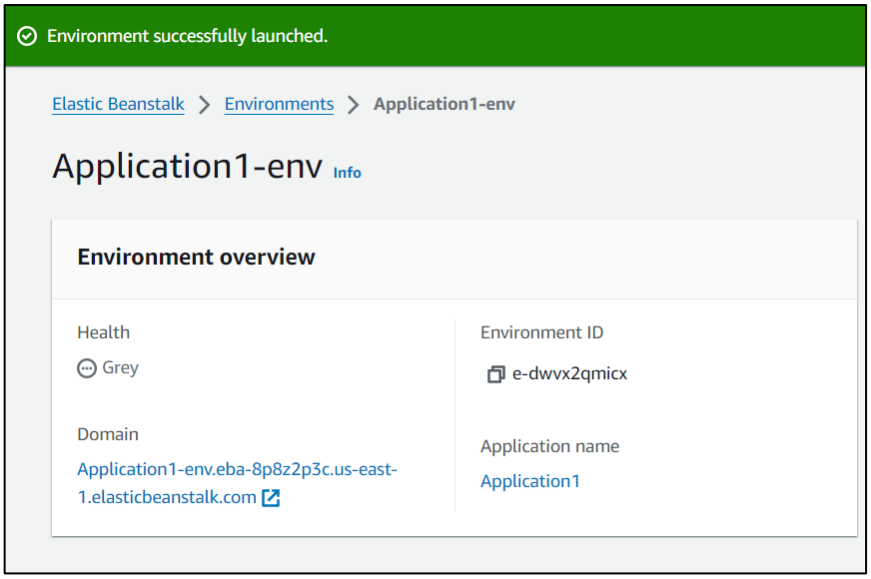




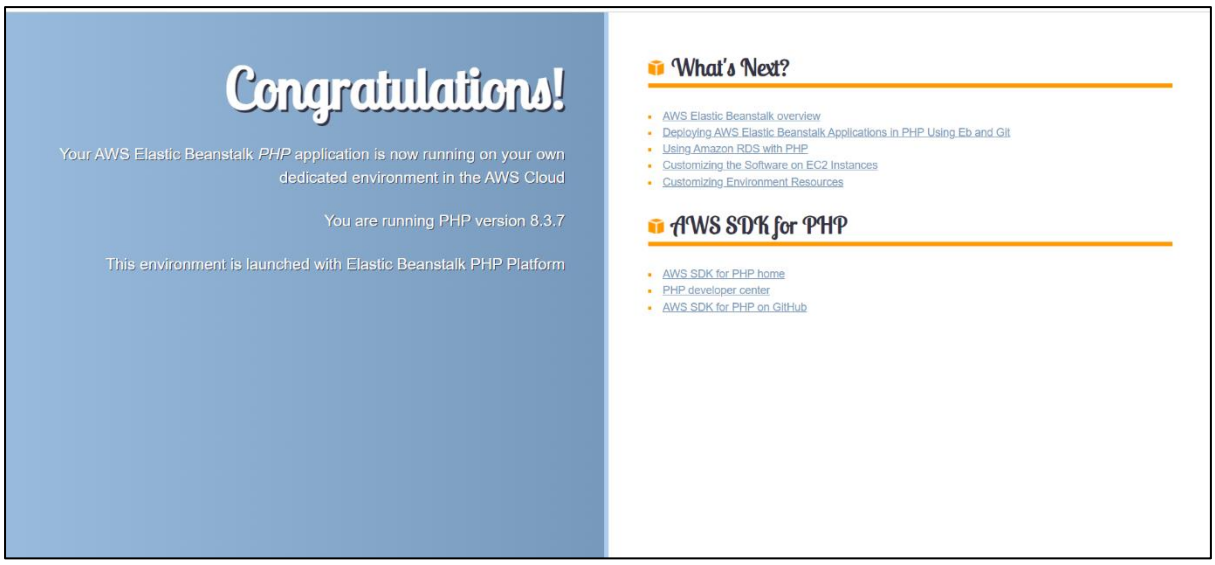
9. Next is the review page where you can check the configurations that have been set in the previous steps. Click on submit.



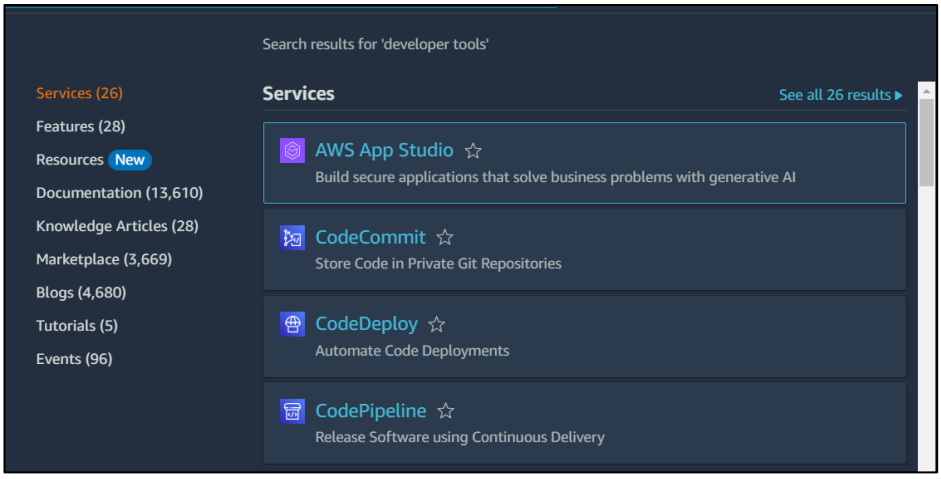
10. Your environment will be successfully created.



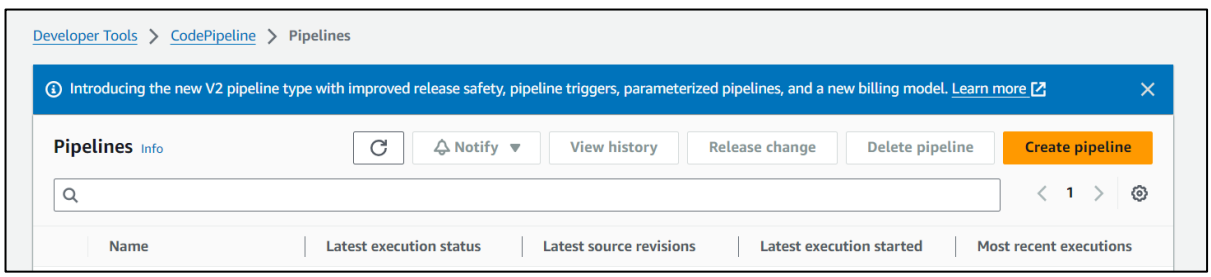
11. After clicking on domain link the page given below will open:



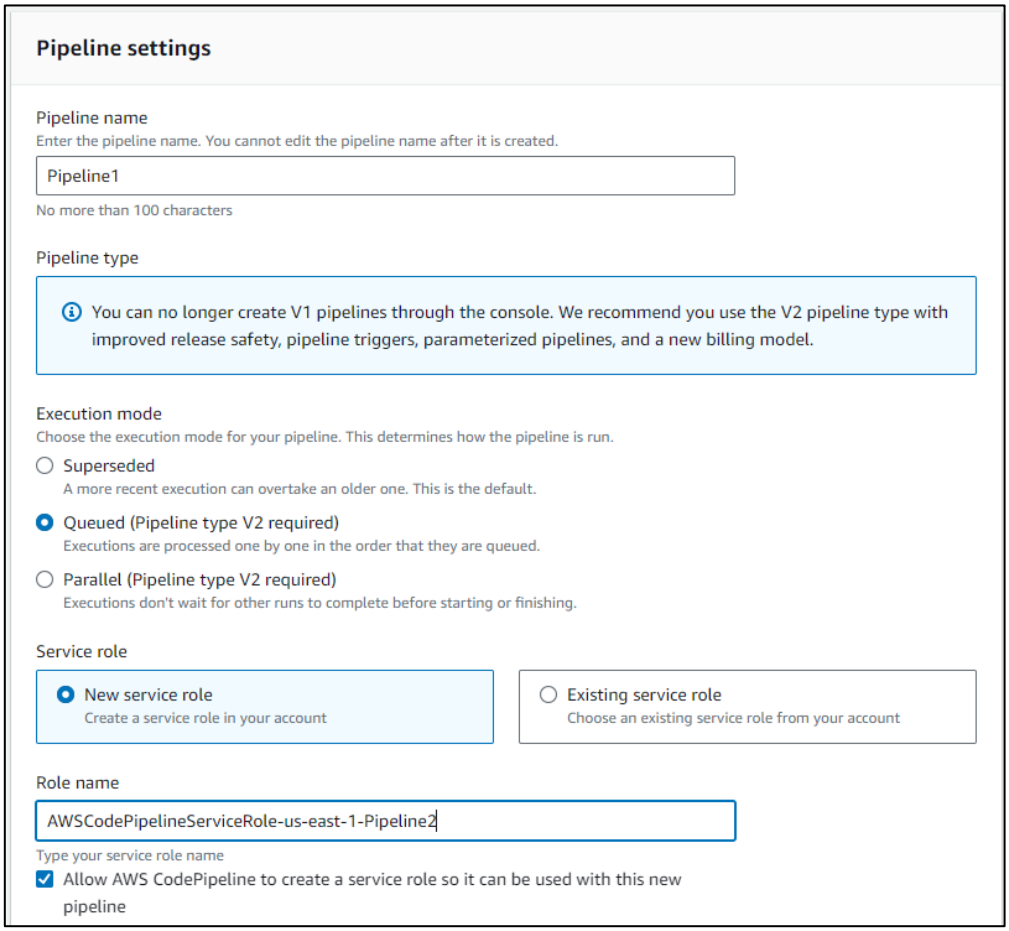
12. Now we will create a pipeline. Go to services and select CodePipeline.



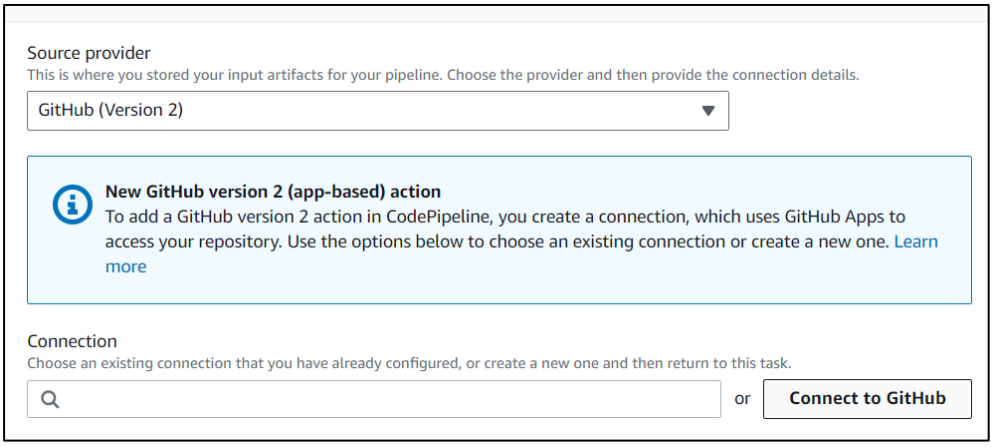
13. Select create pipeline.



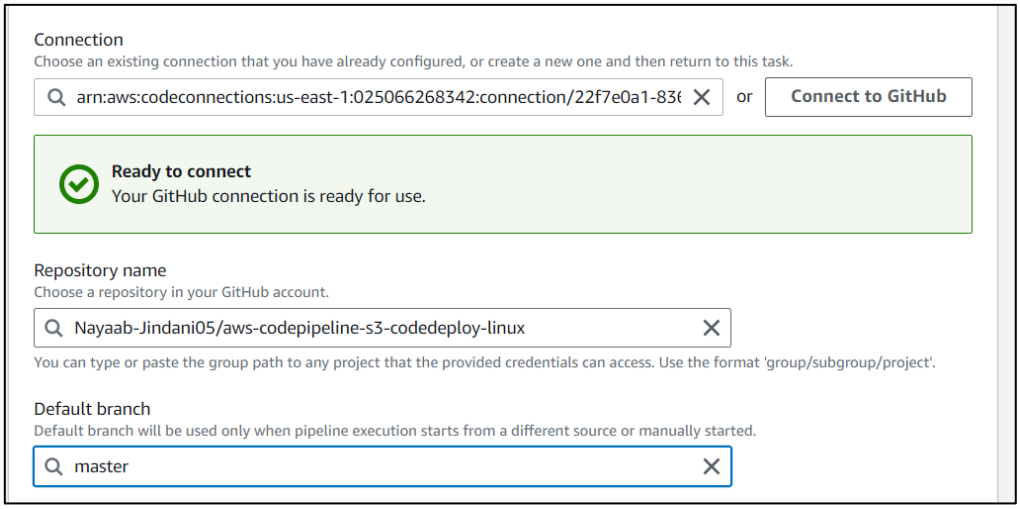
14. In step 1 give the pipeline name



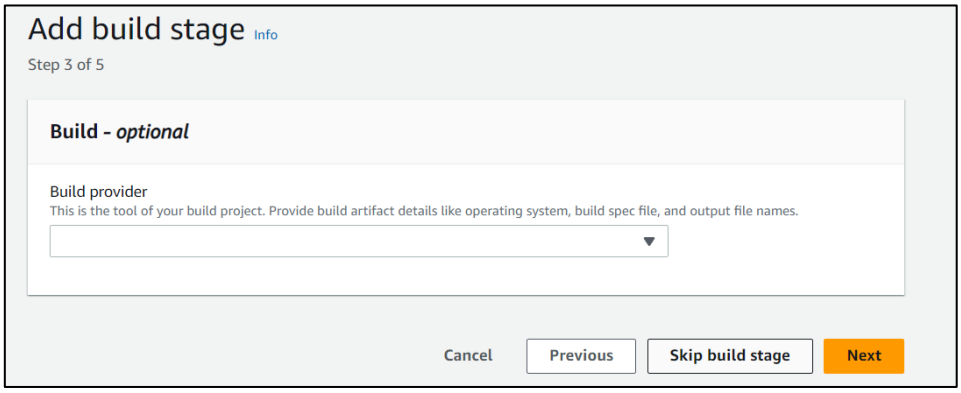
15. In Add source stage select GitHub version2 as the source provider and then connect your AWS to GitHub account and if there is a connection which already exists then select that.



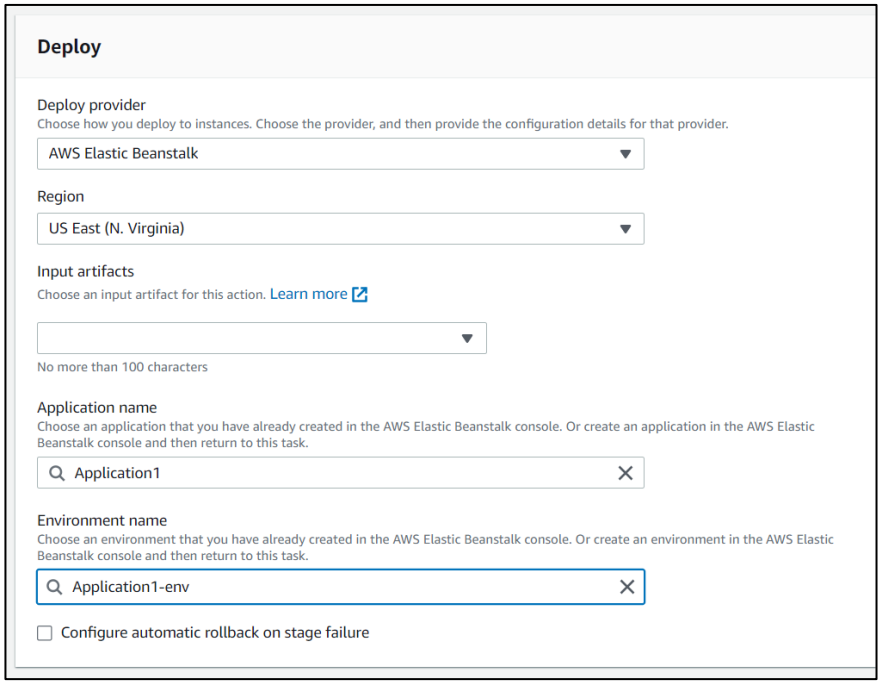
16. After connecting to GitHub select the repository name and default branc



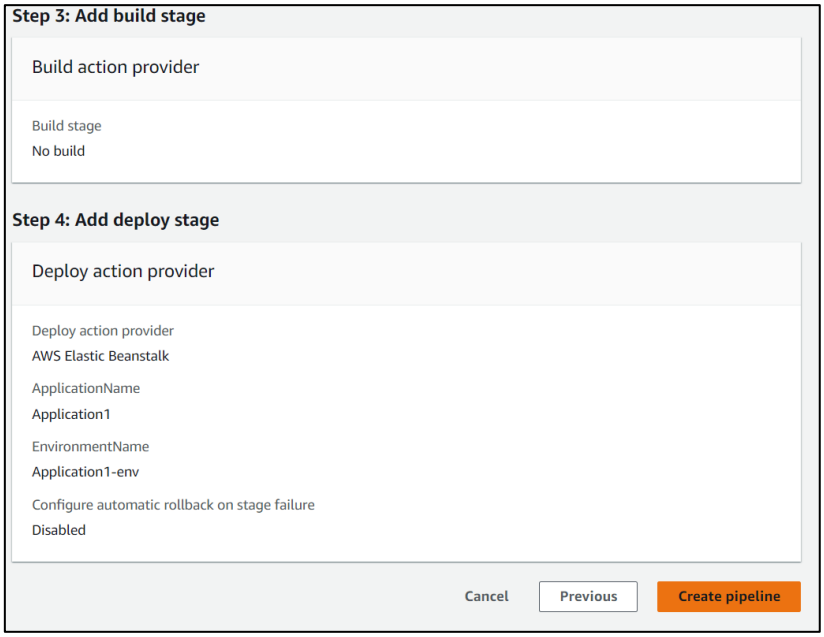
17. Skip the build stage and go to deploy stage



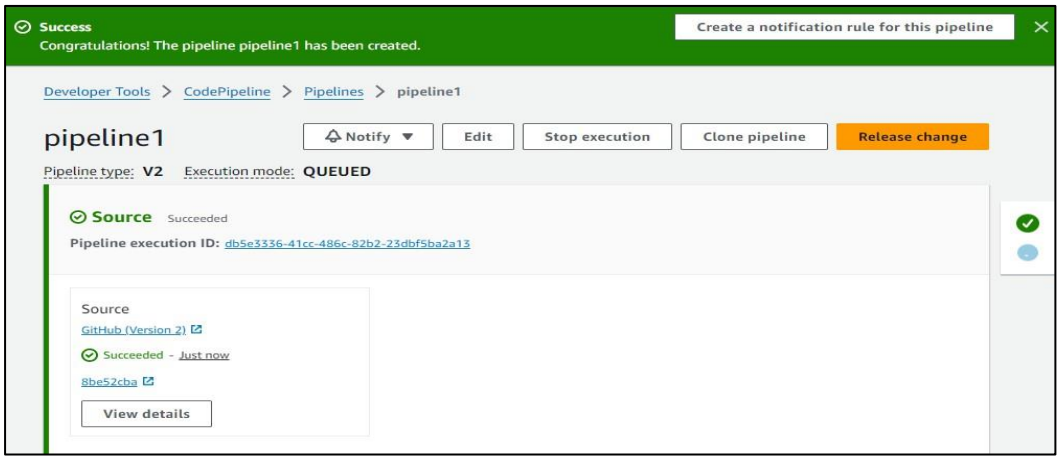
18. In the deploy stage select AWS Elastic Beanstalk as the deploy provider and also select the application name, environment name created earlier.



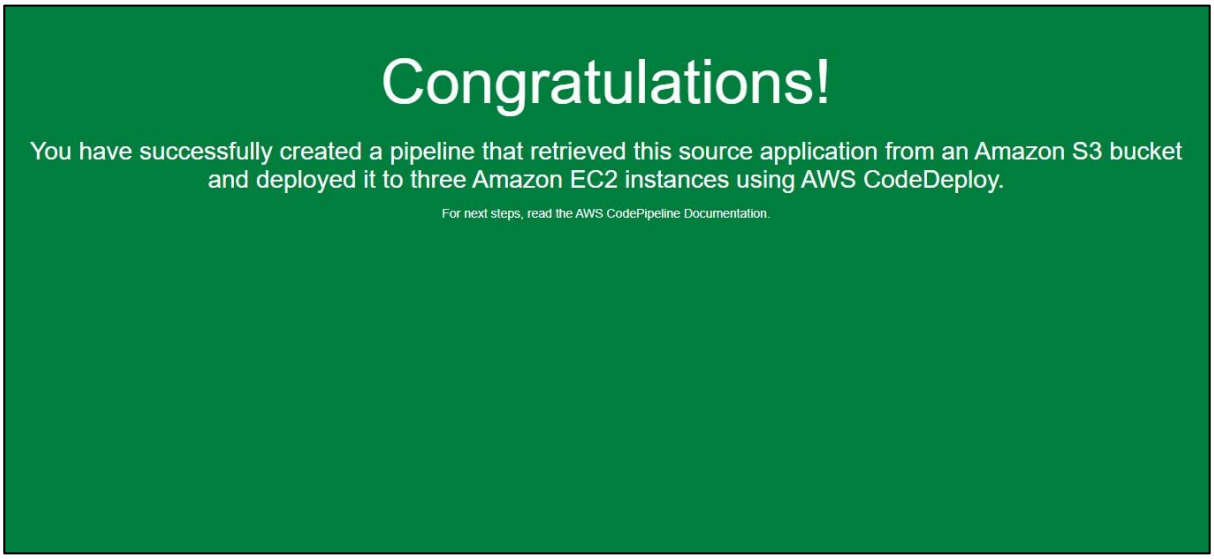
19. In the review stage check all the settings that have been done and select create pipeline.



20. This screen means that pipeline creation is successful



21. Now we can select the URL and it will open a sample website that we have created.



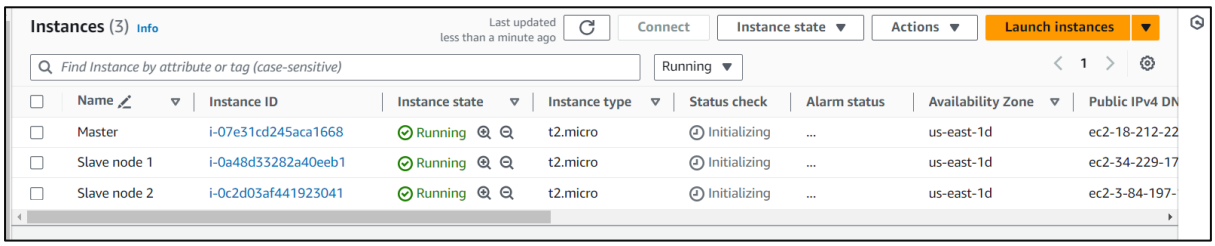
Advance DevOps

Experiment 3

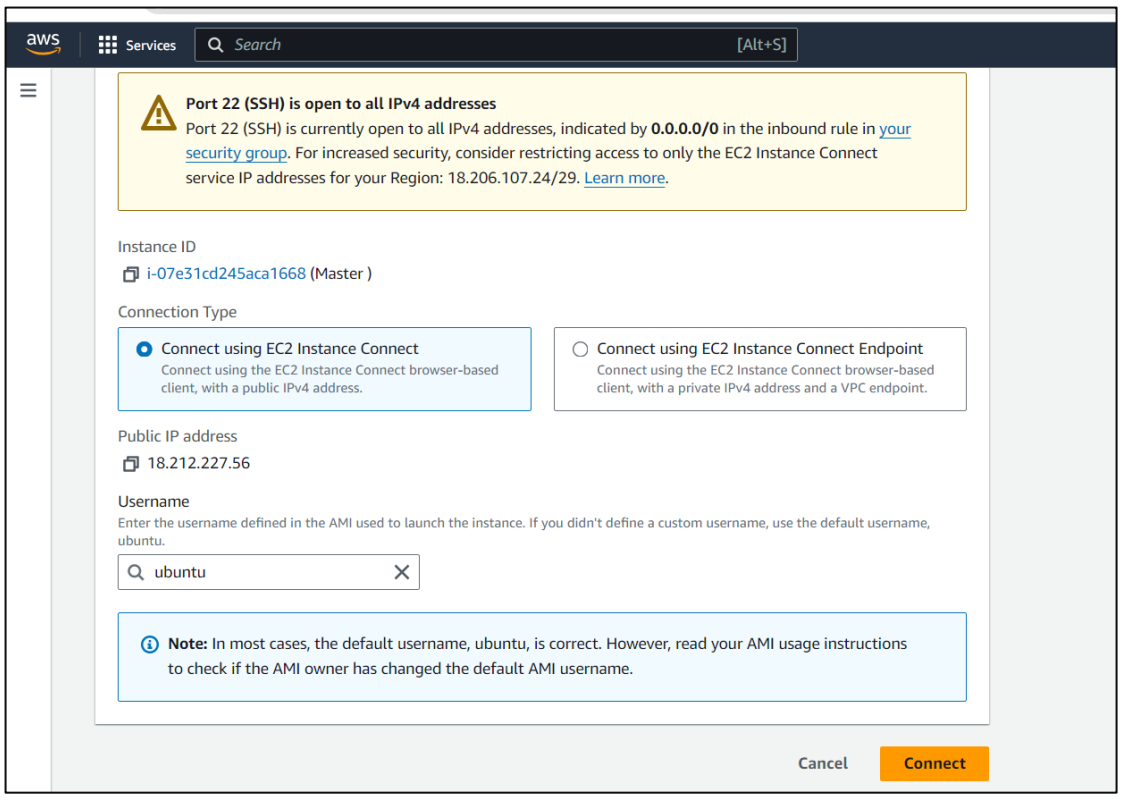
Aim: To understand the Kubernetes Cluster Architecture, install and Spin Up a Kubernetes Cluster on Linux Machines/Cloud Platforms.

Steps:

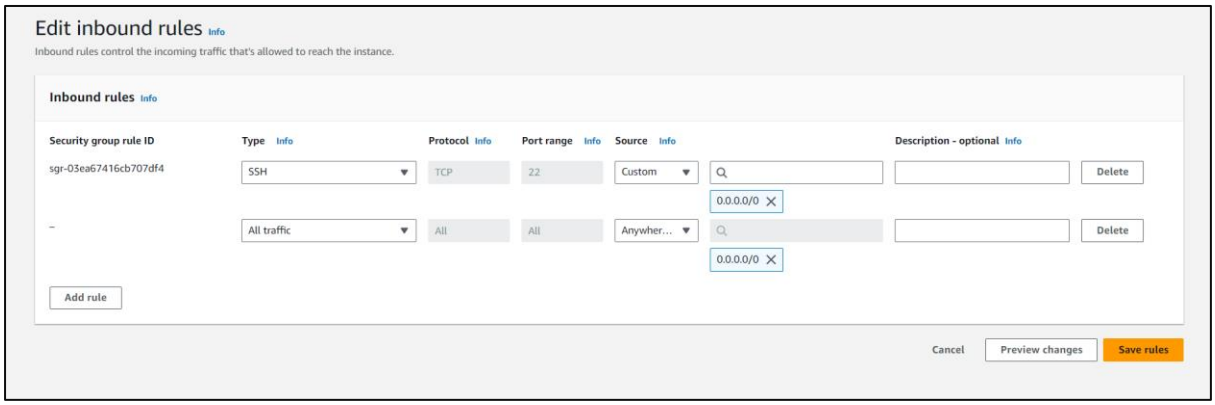
1. We will create 3 EC2 instances. One will be the master node and the other 2 will be slave/worker nodes.



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

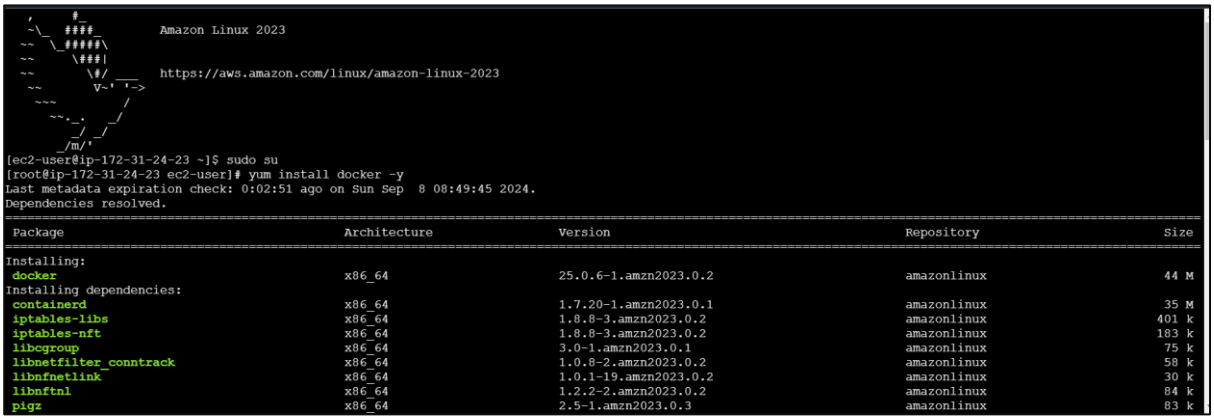


3. Edit the inbound rule to allow all traffic

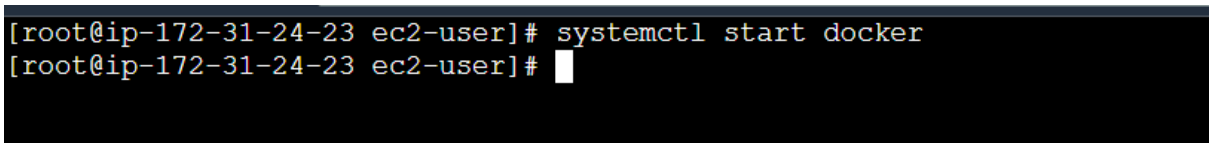


4. Docker installation:

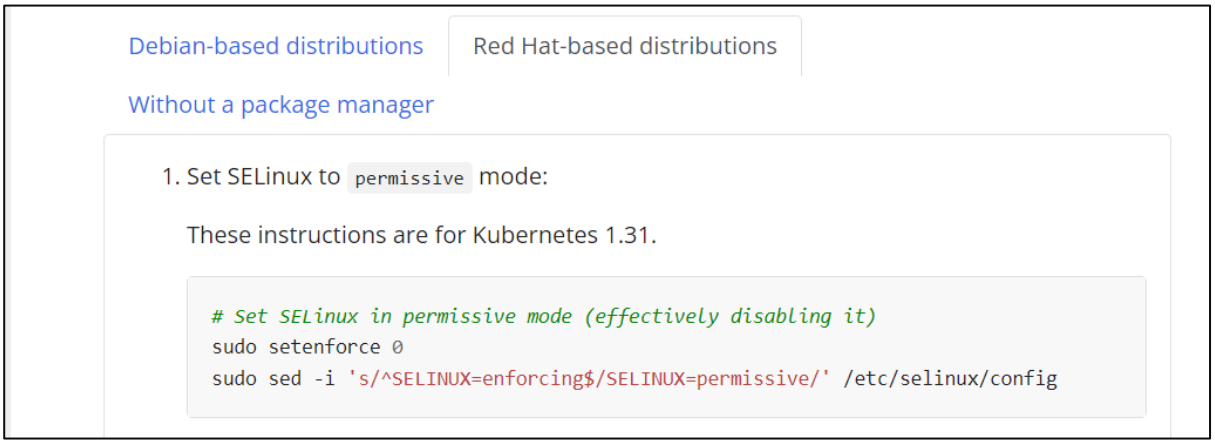
   This step has to be performed on all the 3 instances.
   The following command has to be run:
   yum install docker -y



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



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



```
# This overwrites any existing configuration in /etc/yum.repos.d/kubernetes.repo
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
```

3. Install kubelet, kubeadm and kubectl:

```
sudo 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@ip-172-31-24-23 ec2-user]# sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Kubernetes                                                                      52 kB/s | 7.3 kB     00:00
Dependencies resolved.
================================================================================================================
 Package                     Architecture      Version                          Repository            Size
================================================================================================================
Installing:
 kubeadm                     x86_64            1.31.0-150500.1.1                kubernetes            11 M
 kubectl                     x86_64            1.31.0-150500.1.1                kubernetes            11 M
 kubelet                     x86_64            1.31.0-150500.1.1                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
 socat                       x86_64            1.7.4.2-1.amzn2023.0.2           amazonlinux          303 k

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

Total download size: 51 M
Installed size: 270 M
Downloading Packages:
(1/10): libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64.rpm                 459 kB/s |  24 kB     00:00
(2/10): libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64.rpm                     1.4 MB/s |  30 kB     00:00
```

```
  Installing       : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64                                        7/10
  Running scriptlet: conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64                                        7/10
  Installing       : kubelet-1.31.0-150500.1.1.x86_64                                                   8/10
  Running scriptlet: kubelet-1.31.0-150500.1.1.x86_64                                                   8/10
  Installing       : kubeadm-1.31.0-150500.1.1.x86_64                                                   9/10
  Installing       : kubectl-1.31.0-150500.1.1.x86_64                                                  10/10
  Running scriptlet: kubectl-1.31.0-150500.1.1.x86_64                                                  10/10
  Verifying        : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64                                        1/10
  Verifying        : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64                                 2/10
  Verifying        : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64                                3/10
  Verifying        : libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64                                     4/10
  Verifying        : socat-1.7.4.2-1.amzn2023.0.2.x86_64                                                5/10
  Verifying        : cri-tools-1.31.1-150500.1.1.x86_64                                                 6/10
  Verifying        : kubeadm-1.31.0-150500.1.1.x86_64                                                   7/10
  Verifying        : kubectl-1.31.0-150500.1.1.x86_64                                                   8/10
  Verifying        : kubelet-1.31.0-150500.1.1.x86_64                                                   9/10
  Verifying        : kubernetes-cni-1.5.1-150500.1.1.x86_64                                            10/10

Installed:
  conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64        cri-tools-1.31.1-150500.1.1.x86_64                   kubeadm-1.31.0-150500.1.1.x86_64
  kubectl-1.31.0-150500.1.1.x86_64                   kubelet-1.31.0-150500.1.1.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
  socat-1.7.4.2-1.amzn2023.0.2.x86_64

Complete!
[root@ip-172-31-24-23 ec2-user]# sudo systemctl enable --now kubelet
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /usr/lib/systemd/system/kubelet.service.
[root@ip-172-31-24-23 ec2-user]#
```

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

```
[root@ip-172-31-24-23 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-172-31-24-23 ec2-user]#
```

8. 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-preflight-errors=all

```
[root@ip-172-31-24-23 ec2-user]# kubeadm init
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
        [WARNING FileExisting-tc]: tc not found in system path
error execution phase preflight: [preflight] Some fatal errors occurred:
        [ERROR NumCPU]: the number of available CPUs 1 is less than the required 2
        [ERROR Mem]: the system RAM (949 MB) is less than the minimum 1700 MB
[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
[root@ip-172-31-24-23 ec2-user]# ^C
[root@ip-172-31-24-23 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-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'
W0908 09:11:46.376173   28683 checks.go:846] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent with that used b
y 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-172-31-24-23.ec2.internal kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.c
luster.local] and IPs [10.96.0.1 172.31.24.23]
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
```

```
[root@ip-172-31-82-191 ec2-user]# kubeadm init --ignore-preflight-errors=NumCPU --ignore-preflight-errors=Mem
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
W0913 15:43:09.666583    2255 checks.go:1080] [preflight] WARNING: Couldn't create the interface used for talking to the container runtime: failed to create new C
RI runtime service: validate service connection: validate CRI v1 runtime API for endpoint "unix:///var/run/containerd/containerd.sock": rpc error: code = Unavaila
ble desc = connection error: desc = "transport: Error while dialing: dial unix /var/run/containerd/containerd.sock: connect: no such file or directory"
        [WARNING FileExisting-socat]: socat not found in system path
        [WARNING FileExisting-tc]: tc not found in system path
error execution phase preflight: [preflight] Some fatal errors occurred:
        [ERROR FileContent--proc-sys-net-ipv4-ip_forward]: /proc/sys/net/ipv4/ip_forward contents are not set to 1
[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
[root@ip-172-31-82-191 ec2-user]# ^C
[root@ip-172-31-82-191 ec2-user]# systemctl start docker
[root@ip-172-31-82-191 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 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'
W0913 15:44:18.815161    2565 checks.go:846] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent with that used b
y 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-172-31-82-191.ec2.internal kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.
```

9. 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
```

10. 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 172.31.82.191:6443 --token 8450pt.tdprcovwa61rqyo1 \
        --discovery-token-ca-cert-hash sha256:b11f191f3df19a2e9112a5c19b4461bffeaddd8b5be8625ad8451019aecc043c
```

11. We can check the nodes that have joined the cluster using kubectl get nodes. Right now, there is only one node which is the master node.

```
[root@ip-172-31-85-89 ec2-user]# kubectl get nodes
NAME                          STATUS     ROLES           AGE   VERSION
ip-172-31-85-89.ec2.internal   NotReady   control-plane   72s   v1.26.0
```

12. 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.
```

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

```
[root@ip-172-31-34-212 ec2-user]# kubectl get nodes
NAME                           STATUS   ROLES           AGE   VERSION
ip-172-31-34-212.ec2.internal   Ready    control-plane   18m   v1.31.1
ip-172-31-37-229.ec2.internal   Ready    <none>          13m   v1.31.1
ip-172-31-45-98.ec2.internal    Ready    <none>          14m   v1.31.1
[root@ip-172-31-34-212 ec2-user]#
```

Conclusion:

This experiment enabled the creation of a Kubernetes cluster and the successful joining of all 3 nodes using various commands. Errors during initialization can be handled in two ways: 1. By ignoring the errors, or 2. By changing the instance type to t3.medium or t3.large if the issue is related to insufficient memory space or CPU resources. Also, it is to be ensured that the inbound rules and outbound rules allow all traffic or else it leads to connectivity issues between master node and worker nodes.

Advance DevOps

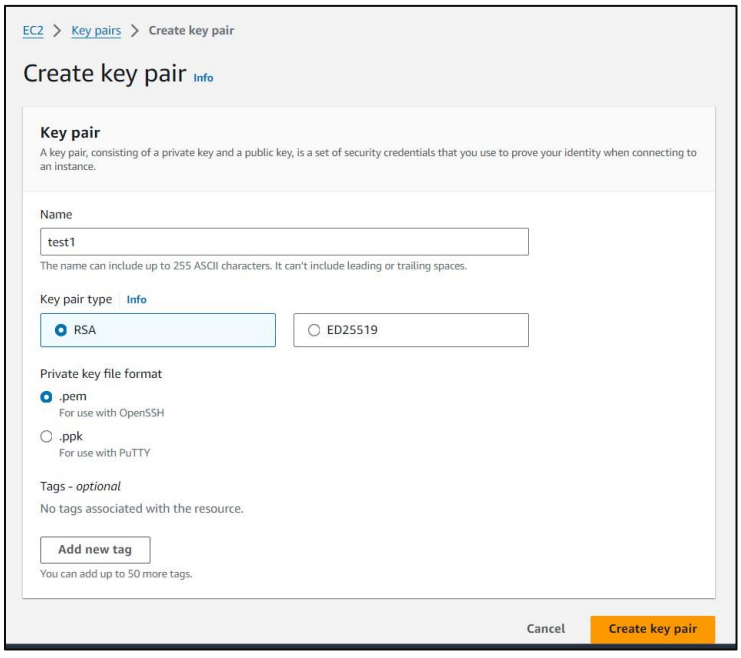
Experiment 4

Aim:

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

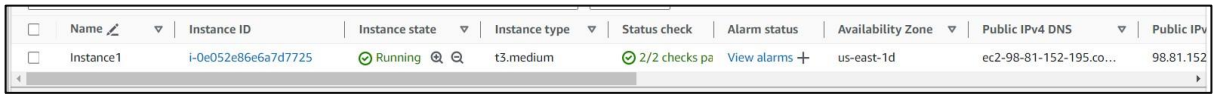
Steps:

1. Create a key pair

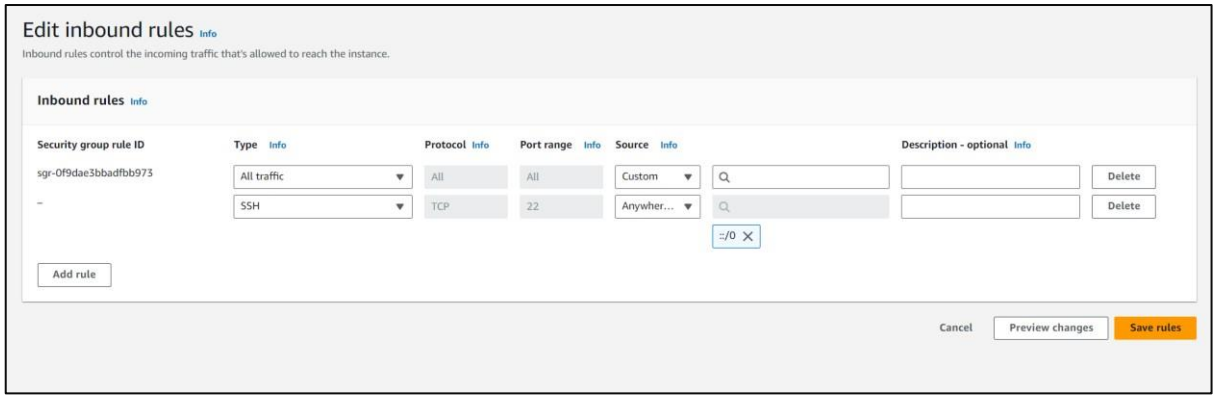


The .pem file will be downloaded on your machine and will be required in the further steps.

2. Now we will create an EC2 Ubuntu instance. Select the key pair which you just created while creating this instance.

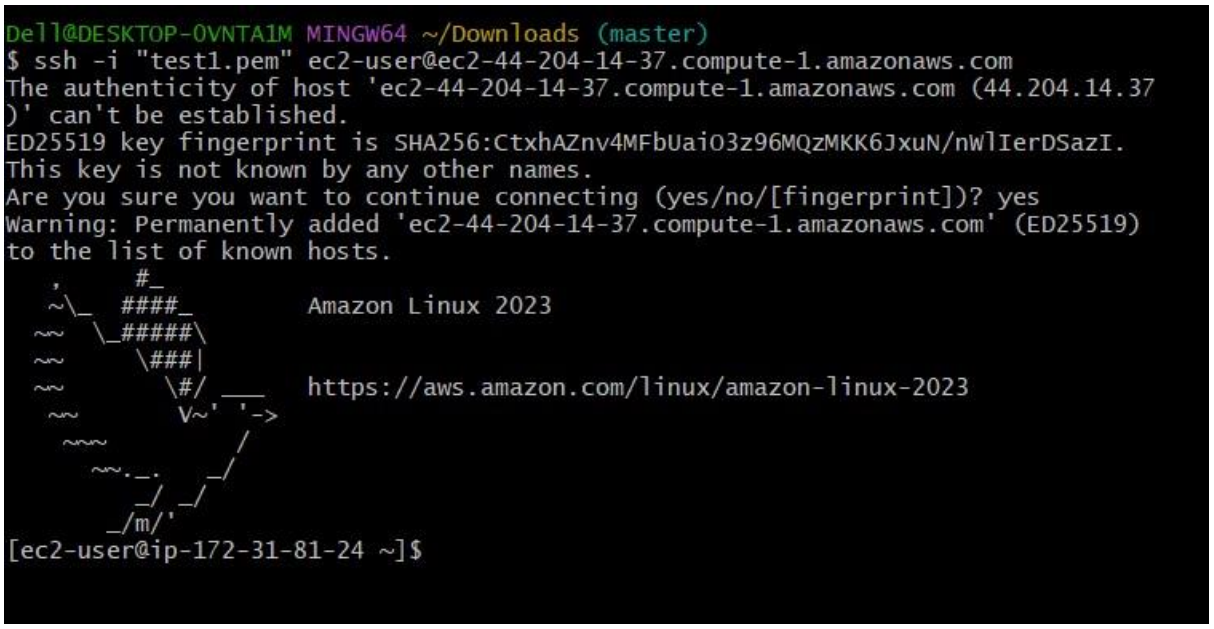


3. Now edit the inbound rules to allow ssh

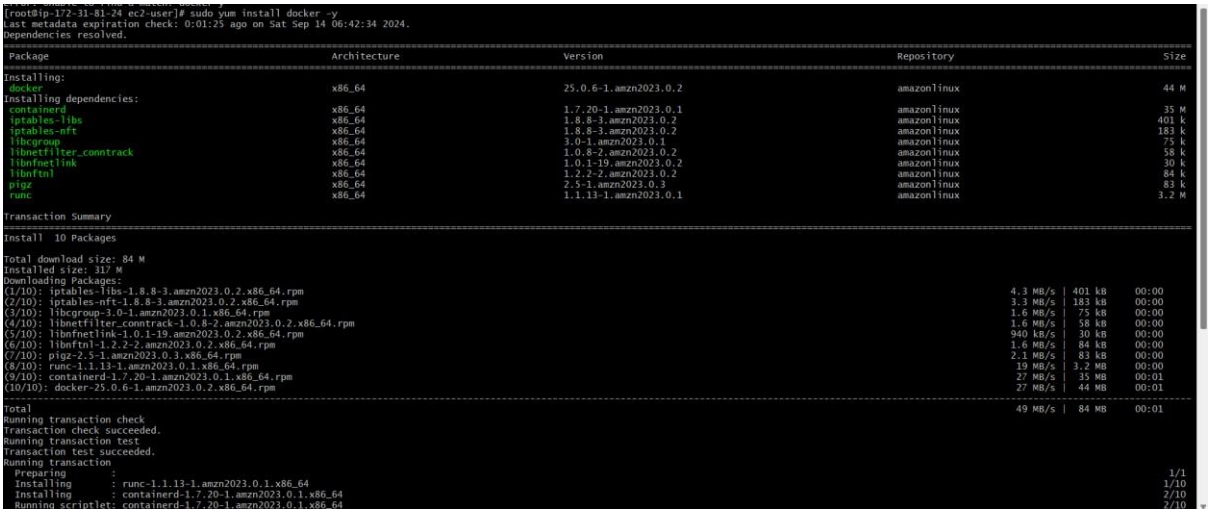


4. Open git bash and go to the directory where pem file is located and use chmod to provide permissions.

5. Now use this command on the terminal: ssh -i <keyname>.pem ubuntu@<public_ip_address> and replace
   - Keyname with the name of your key pair, in our case test1.
   - As we are using amazon Linux instead of ubuntu we will have ec2-user
   - Replace public ip address with its value. Go to your instance and scroll down you will find the public ip address there.



6. Docker installation:
   We will be installing docker by using “sudo yum install docker -y”



7. Then to configure cgroup in a daemon json file we will run

cd /etc/docker

cat <<EOF | sudo tee /etc/docker/daemon.json

{

"exec-opts": ["native.cgroupdriver=systemd"]

}

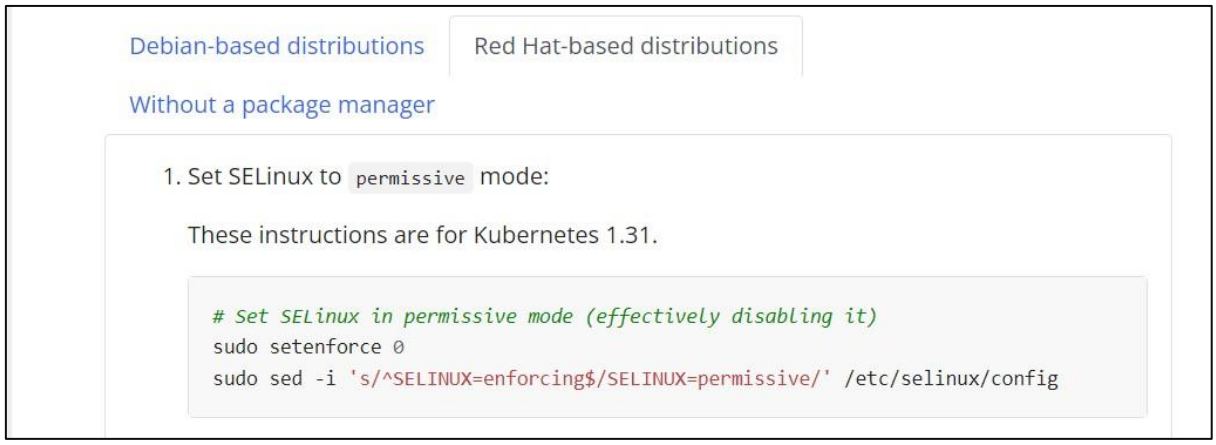
EOF

sudo systemctl enable docker sudo

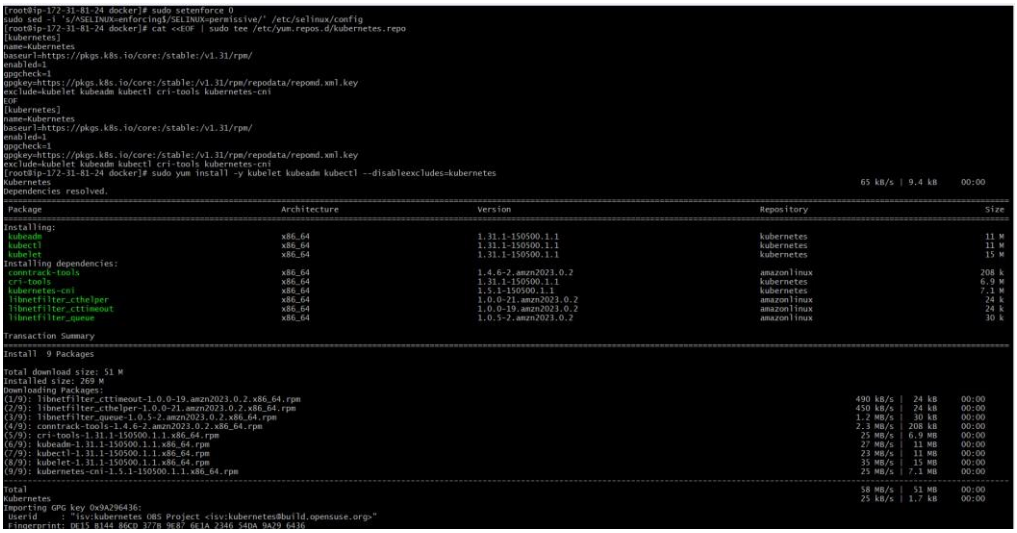
systemctl daemon-reload sudo

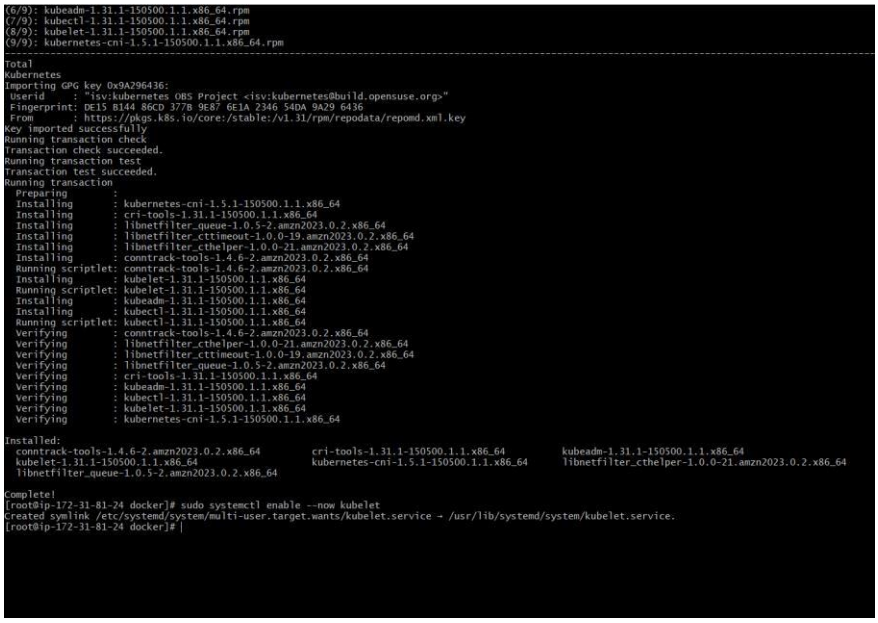
systemctl restart docker

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

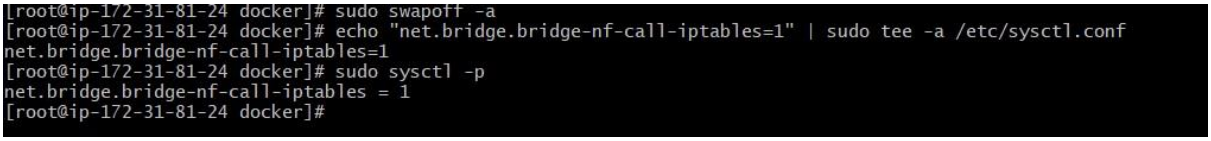


Copy the above given steps and paste in the terminal. This will create a Kubernetes repository, install kubelet,kubeadm and kubectl and also enable the services.





9. After installing Kubernetes, we need to configure internet options to allow bridging.
   sudo swapoff -a
   echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a
   /etc/sysctl.conf sudo
   sysctl -p



10. Initializing kubecluster: sudo kubeadm init --pod-network-cidr=10.244.0.0/16

```
[root@ip-172-31-81-24 docker]# sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
        [WARNING FileExisting-socat]: socat not found in system path
        [WARNING FileExisting-tc]: tc not found in system path
error execution phase preflight: [preflight] Some fatal errors occurred:
        [ERROR NumCPU]: the number of available CPUs 1 is less than the required 2
        [ERROR Mem]: the system RAM (949 MB) is less than the minimum 1700 MB
[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
[root@ip-172-31-81-24 docker]# sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=NumCPU,Mem
[init] Using Kubernetes version: v1.31.0
```

```
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 172.31.81.24:6443 --token 4a91z3.yz6rwmmkf9yncyd2 \
        --discovery-token-ca-cert-hash sha256:3404bd1bcdd9cf90a003673f622d1672acb4c6ce7c15c4738c80a0a1560fe70d
[root@ip-172-31-81-24 docker]#
```

11. The mkdir command that is generated after initialization has to be copy pasted in the terminal.

```
[root@ip-172-31-81-24 docker]#  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

12. Then, add a common networking plugin called flannel:
    kubectl apply -f
    https://raw.githubusercontent.com/coreos/flannel/master/Documentation/ kube-flannel.yml

```
[root@ip-172-31-81-24 docker]# 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
[root@ip-172-31-81-24 docker]#
```

13. 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

```
[root@ip-172-31-81-24 docker]# kubectl apply -f https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
[root@ip-172-31-81-24 docker]#
```

14. Use kubectl get pods to check if pod is working correctly

```
[root@ip-172-31-81-24 docker]# kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-8jdlf    0/1     Pending   0          18s
```

15. To change status from pending to running use following command: kubectl describe pod nginx.

```
Name:             nginx-deployment-d556bf558-gw8v8
Namespace:        default
Priority:         0
Service Account:  default
Node:             <none>
Labels:           app=nginx
                  pod-template-hash=d556bf558
Annotations:      <none>
Status:           Pending
IP:
IPs:              <none>
Controlled By:    ReplicaSet/nginx-deployment-d556bf558
Containers:
  nginx:
    Image:        nginx:1.14.2
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:
```

```
Conditions:
  Type           Status
  PodScheduled   False
Volumes:
  kube-api-access-f9k9s:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    ConfigMapOptional:       <nil>
    DownwardAPI:             true
QoS Class:                   BestEffort
Node-Selectors:              <none>
Tolerations:                 node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type     Reason            Age    From               Message
  ----     ------            ----   ----               -------
  Warning  FailedScheduling  114s   default-scheduler  0/1 nodes are available: 1 node(s) had untolerated taint {node-role.kubernetes.io/control-
: }. preemption: 0/1 nodes are available: 1 Preemption is not helpful for scheduling.
  Warning  FailedScheduling  3m18s  default-scheduler  0/1 nodes are available: 1 node(s) had untolerated taint {node-role.kubernetes.io/control-
: }. preemption: 0/1 nodes are available: 1 Preemption is not helpful for scheduling.
```

Use the below command to remove taints

```
[root@ip-172-31-81-24 docker]# kubectl taint nodes --all node-role.kubernetes.io
/control-plane-
```

16. Check the pod status

```
[root@ip-172-31-81-24 docker]# kubectl get pods
NAME    READY   STATUS    RESTARTS      AGE
nginx   1/1     Running   1 (6s ago)    90s
```

17. port forward the deployment to your localhost so that you can view it.

```
[root@ip-172-31-81-24 docker]# kubectl port-forward nginx 8081:80
Forwarding from 127.0.0.1:8081 -> 80
Forwarding from [::1]:8081 -> 80
```

18. 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

    If you see your nginx server name and response code is 200 then the deployment was successful.

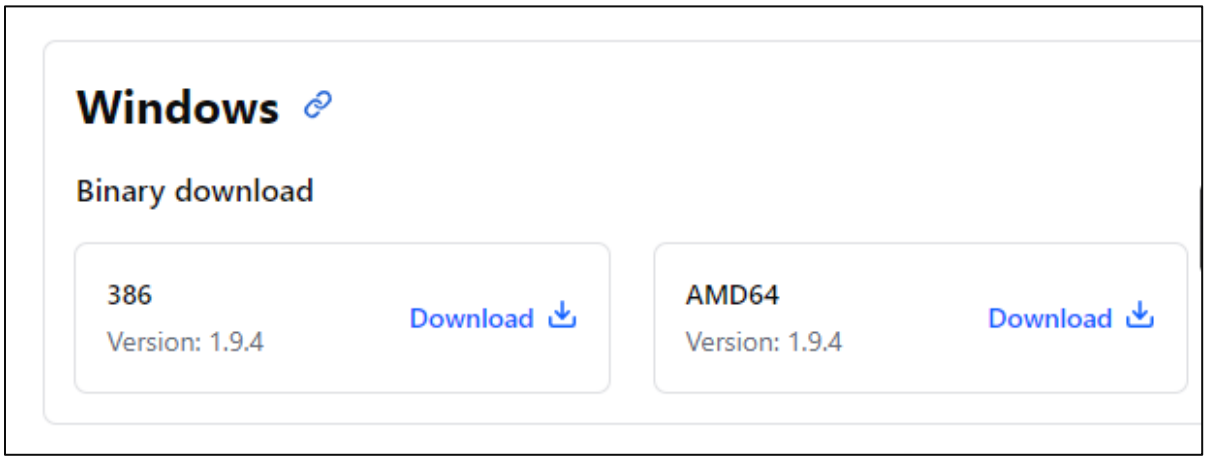
```
HTTP/1.1 200 OK
Server: nginx/1.14.2
Date: Sat, 14 Sep 2024 06:54:21 GMT
Content-Type: text/html
Content-Length: 612
Last-Modified: Tue, 04 Dec 2018 14:44:49 GMT
Connection: keep-alive
ETag: "5c0692e1-264"
Accept-Ranges: bytes
```

Conclusion: In this experiment we created an ec2 instance, enabled ssh by editing the inbound rules. After that we installed docker and Kubernetes and configured internet options to allow bridging. Once this setup got completed, we added a common networking plugin called flannel. Once the cluster started running we deployed nginx server on this cluster and verified deployment.

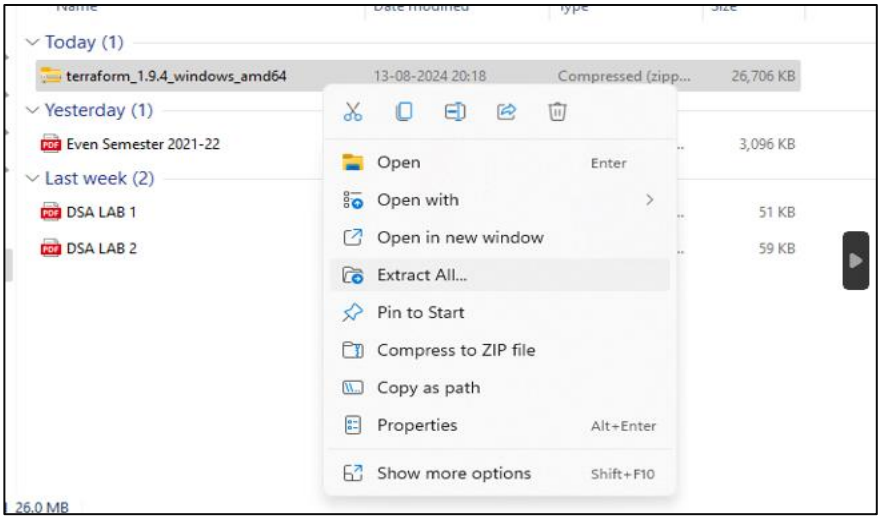
Advance DevOps

Experiment 5

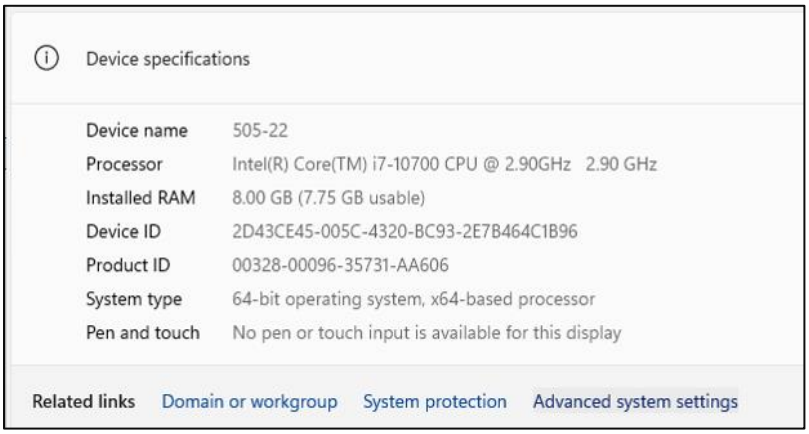
Step 1: To install terraform first download the Terraform CLI Utility for Windows from terraform official website www.terraform.io/downloads.html



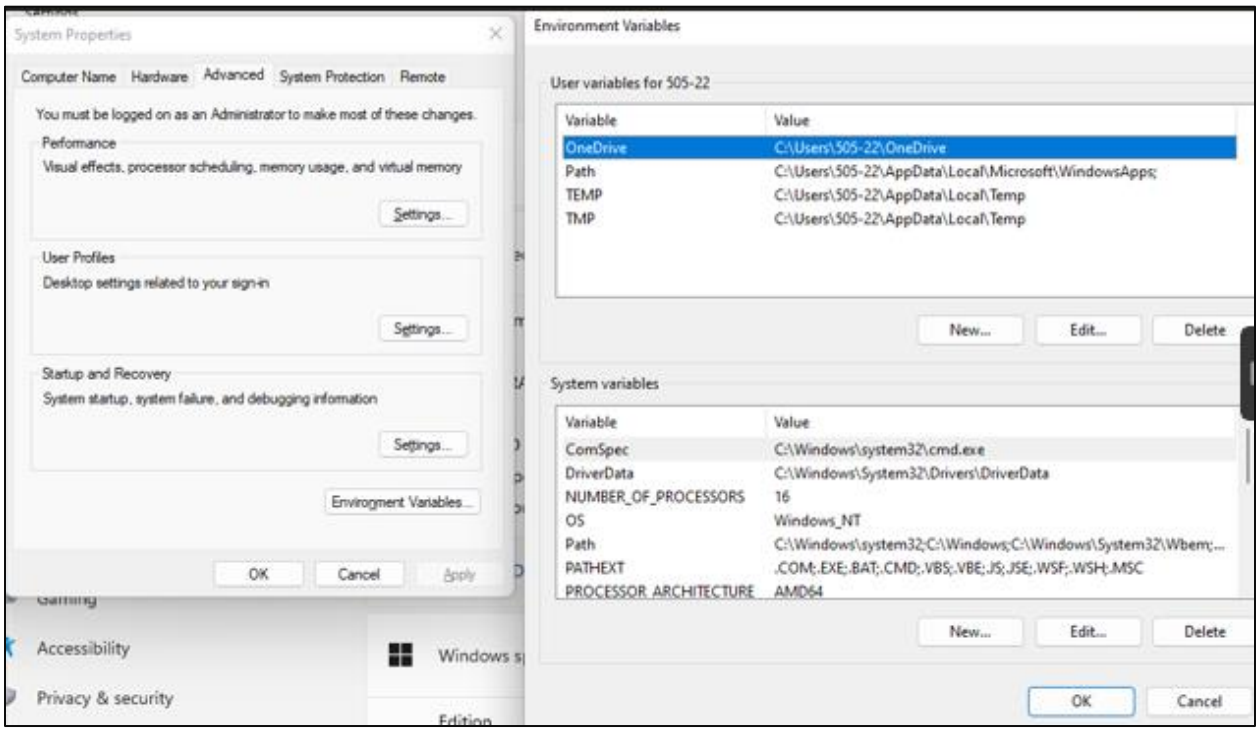
Step 2: After the zip file is downloaded, we will right click on it and click on extract all files.

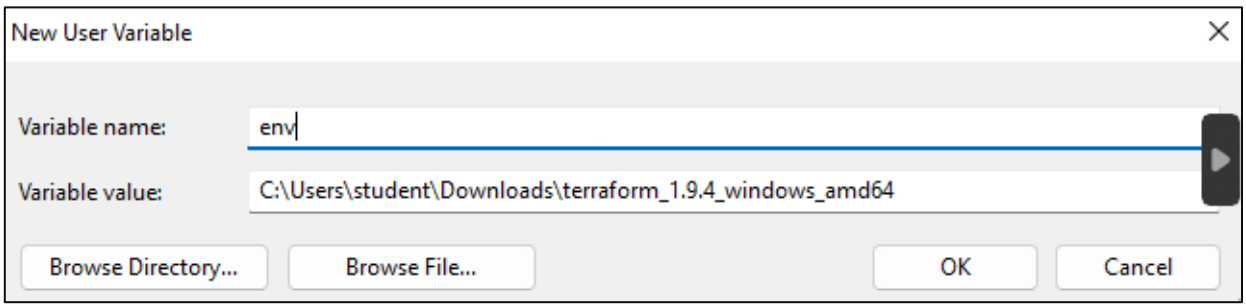


Step 3: Go to advance system settings in settings



Step 4: In advance settings select environment variables and set the path of Terraform.





Step 5: In order to check whether terraform was successfully installed we have to run the command “terraform -v” which will give us the version of our terraform and hence terraform has been downloaded correctly.

```
C:\Users\Dell\Downloads\terraform_1.9.4_windows_amd64>terraform -v
Terraform v1.9.4
on windows_amd64
```

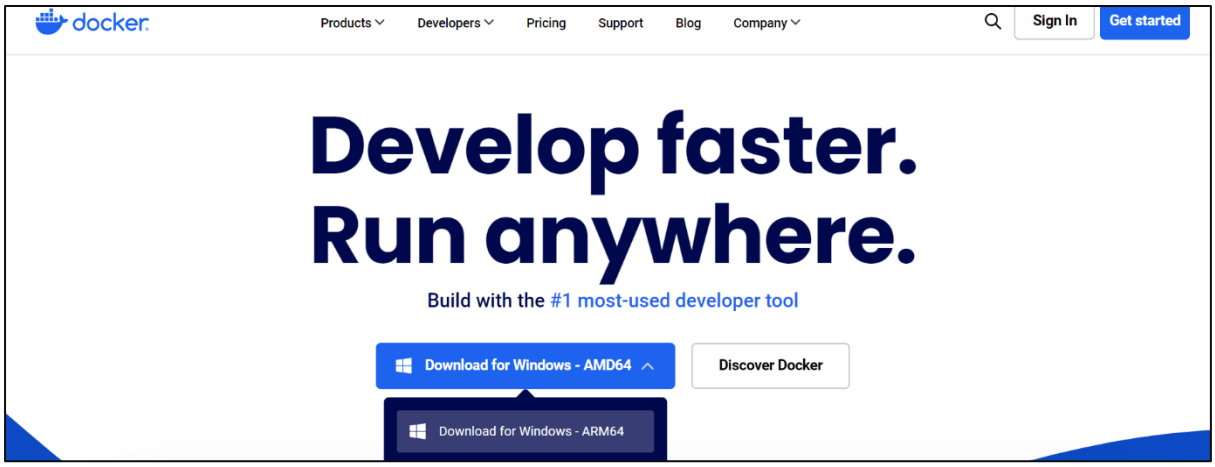
Advance DevOps

Experiment 6

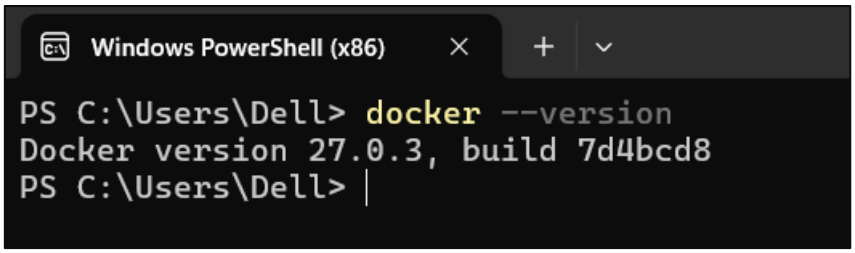
Creating docker image using terraform

Steps:

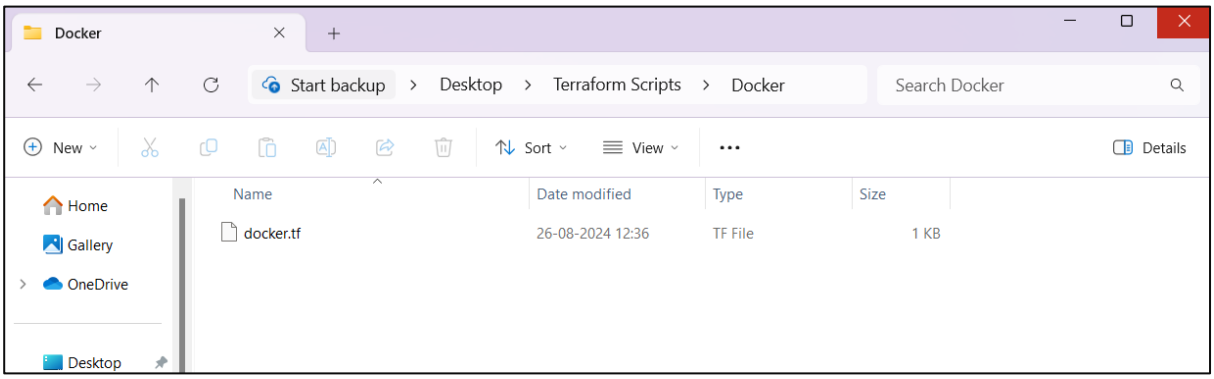
1. Firstly, we will have to download and install docker from its official website.



2. To check if docker has been correctly installed we can use the command “docker –version”. If it returns the version then installation has been successful.



3. Now we will create a folder “Terraform Scripts” and inside this folder we will create another folder “Docker” where our docker script will be saved. Using an editor create a file named docker and save it with .tf extension inside the above-mentioned directory structure.

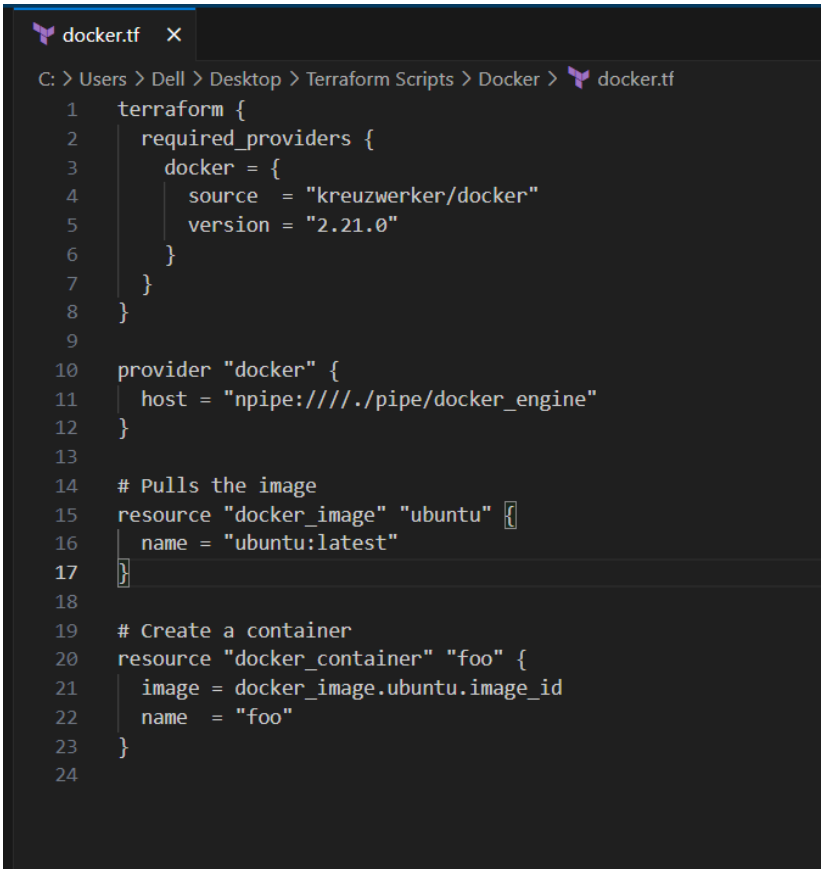


4. The docker.tf file will contain the following:

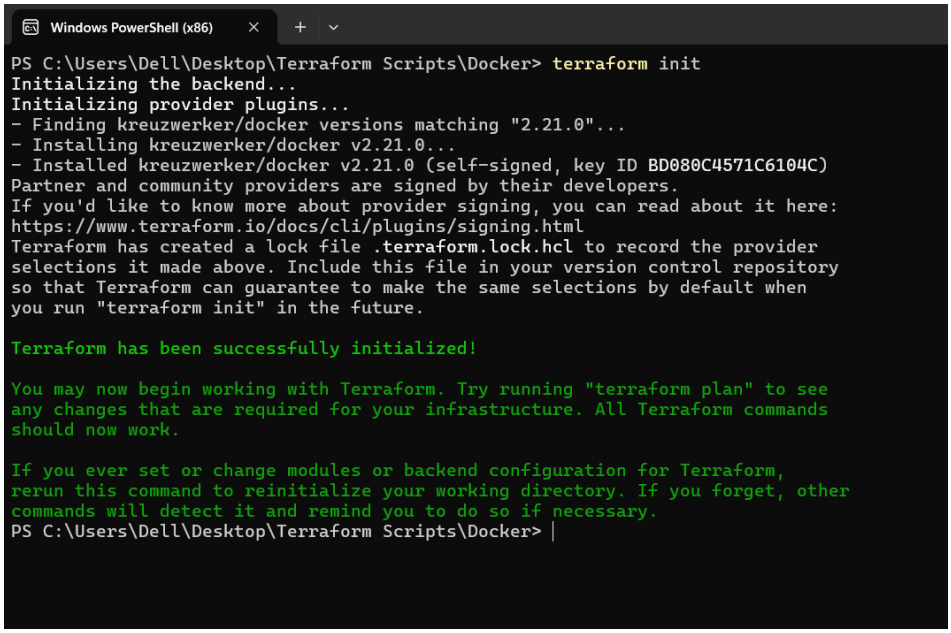
```
terraform {
  required_providers {
    docker = {
      source  = "kreuzwerker/docker"
      version = "2.21.0"
    }
  }
}

provider "docker" {
  host = "npipe:////./pipe/docker_engine"
}

# Pulls the image
resource "docker_image" "ubuntu" {
  name = "ubuntu:latest"
}
```

```
# Create a container
resource "docker_container" "foo" {
  image = docker_image.ubuntu.image_id
  name  = "foo"
}
```

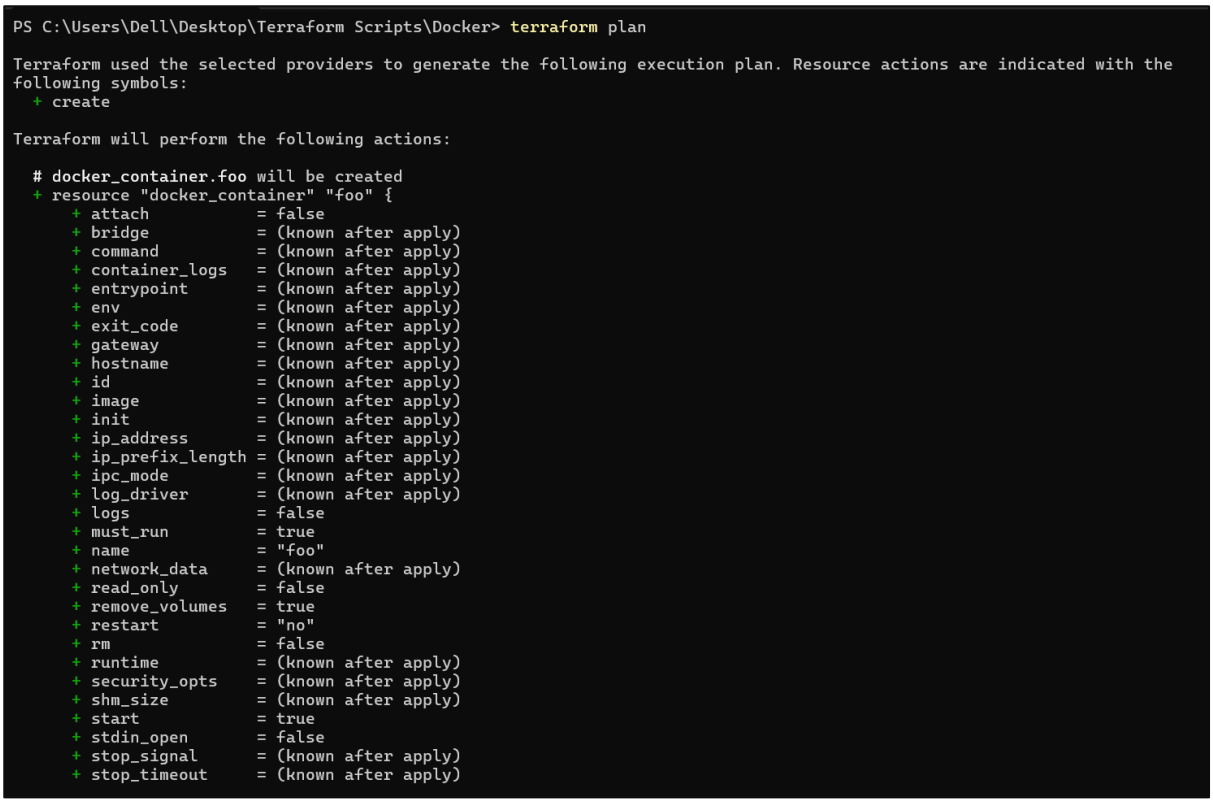
Ensure correct block formatting to avoid errors.



5. Now execute “terraform init” command in power shell. This command will initialize your working directory and download the necessary provider plugins. (Make sure you are in the “Docker” directory)



6. Next execute “terraform plan” command. The terraform plan command is used to create an execution plan for terraform. This plan shows you what changes terraform will make to your infrastructure based on your current configuration files and the state of your existing infrastructure.



7. After this execute the “terraform apply” command. The terraform apply command executes the actions proposed in terraform plan. It is used to deploy your infrastructure.

   In this step I got an error saying “container exited immediately” which indicates that the Docker container created by terraform configuration exited immediately after being started. In order to keep the container running I updated the script file with the following code:

   command = ["sleep", "infinity"] # Keeps the container running

```
terraform {
  required_providers {
    docker = {
      source  = "kreuzwerker/docker"
      version = "2.21.0"
    }
  }
}

provider "docker" {
  host = "npipe:////./pipe/docker_engine"
}

# Pulls the image
resource "docker_image" "ubuntu" {
  name = "ubuntu:latest"
}

  resource "docker_container" "foo" {
  image = docker_image.ubuntu.image_id
  name  = "foo"
  command = ["sleep", "infinity"]  # Keeps the container running indefinitely
}
```

8. After updating the script, the error will be solved and we can again run apply command.

```
PS C:\Users\Dell\Desktop\Terraform Scripts\Docker> terraform apply
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]

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          = [
          + "sleep",
          + "infinity",
        ]
      + 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            = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a"
      + 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
```

```
      + 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)
    }

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

docker_container.foo: Creating...
docker_container.foo: Creation complete after 0s [id=334f576e574fb630507997954b3a9f5c39af4e6aa10b085a0a3a5122ee19ef81]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

Docker image has been created.

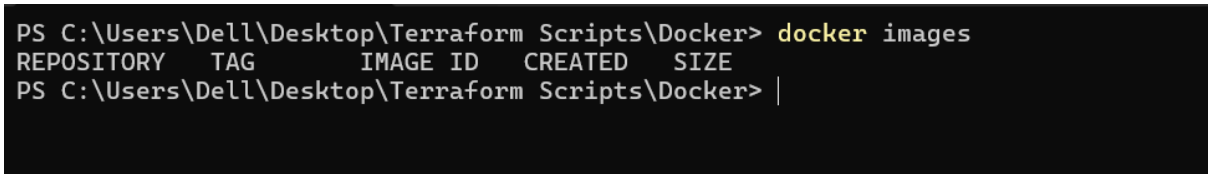
9. The image created can be checked by using docker images command. It will show the repository, tag, image id, time of creation and size as shown below.

```
PS C:\Users\Dell\Desktop\Terraform Scripts\Docker> docker images
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
ubuntu       latest    edbfe74c41f8   3 weeks ago   78.1MB
PS C:\Users\Dell\Desktop\Terraform Scripts\Docker>
```

10. Using terraform destroy we can destroy the container.

```
PS C:\Users\Dell\Desktop\Terraform Scripts\Docker> terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Refreshing state... [id=334f576e574fb630507997954b3a9f5c39af4e6aa10b085a0a3a5122ee19ef81]

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

Terraform will perform the following actions:

  # docker_container.foo will be destroyed
  - resource "docker_container" "foo" {
      - attach           = false -> null
      - command          = [
          - "sleep",
          - "infinity",
        ] -> null
      - cpu_shares       = 0 -> null
      - dns              = [] -> null
      - dns_opts         = [] -> null
      - dns_search       = [] -> null
      - entrypoint       = [] -> null
      - env              = [] -> null
      - gateway          = "172.17.0.1" -> null
      - group_add        = [] -> null
      - hostname         = "334f576e574f" -> null
      - id               = "334f576e574fb630507997954b3a9f5c39af4e6aa10b085a0a3a5122ee19ef81" -> null
      - image            = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> null
      - init             = false -> null
      - ip_address       = "172.17.0.2" -> null
      - ip_prefix_length = 16 -> null
      - ipc_mode         = "private" -> null
      - links            = [] -> null
      - log_driver       = "json-file" -> null
      - log_opts         = {} -> null
      - logs             = false -> null
      - max_retry_count  = 0 -> null
      - memory           = 0 -> null
      - memory_swap      = 0 -> null
      - must_run         = true -> null
      - name             = "foo" -> null
      - network_data     = [
      - rm               = false -> null
      - runtime          = "runc" -> null
      - security_opts    = [] -> null
      - shm_size         = 64 -> null
      - start            = true -> null
      - stdin_open       = false -> null
      - stop_timeout     = 0 -> null
      - storage_opts     = {} -> null
      - sysctls          = {} -> null
      - tmpfs            = {} -> null
      - tty              = false -> null
        # (8 unchanged attributes hidden)
    }

  # docker_image.ubuntu will be destroyed
  - resource "docker_image" "ubuntu" {
      - id          = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest" -> null
      - image_id    = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> null
      - latest      = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> null
      - name        = "ubuntu:latest" -> null
      - repo_digest = "ubuntu@sha256:8a37d68f4f73ebf3d4efafbcf66379bf3728902a8038616808f04e34a9ab63ee" -> null
    }

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

Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

docker_container.foo: Destroying... [id=334f576e574fb630507997954b3a9f5c39af4e6aa10b085a0a3a5122ee19ef81]
docker_container.foo: Destruction complete after 0s
docker_image.ubuntu: Destroying... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_image.ubuntu: Destruction complete after 0s

Destroy complete! Resources: 2 destroyed.
```

Now when we run docker images we will see that the container information is not visible that means it has been deleted successfully.



Advance DevOps

Experiment 7

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

Steps:

1. Firstly, we will ensure whether docker is installed or not by running docker -v in the command prompt.

```
C:\Users\Dell>docker -v
Docker version 27.1.1, build 6312585
```

2. Run docker login command and add your username and password for docker.

```
C:\Users\Dell>docker login
Authenticating with existing credentials...
Stored credentials invalid or expired
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, head
over to https://hub.docker.com/ to create one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security and
is required for organizations using SSO. Learn more at https://docs.docker.com/go/access-tokens/

Username (dimple866): dimple866
Password:

Login Succeeded
```

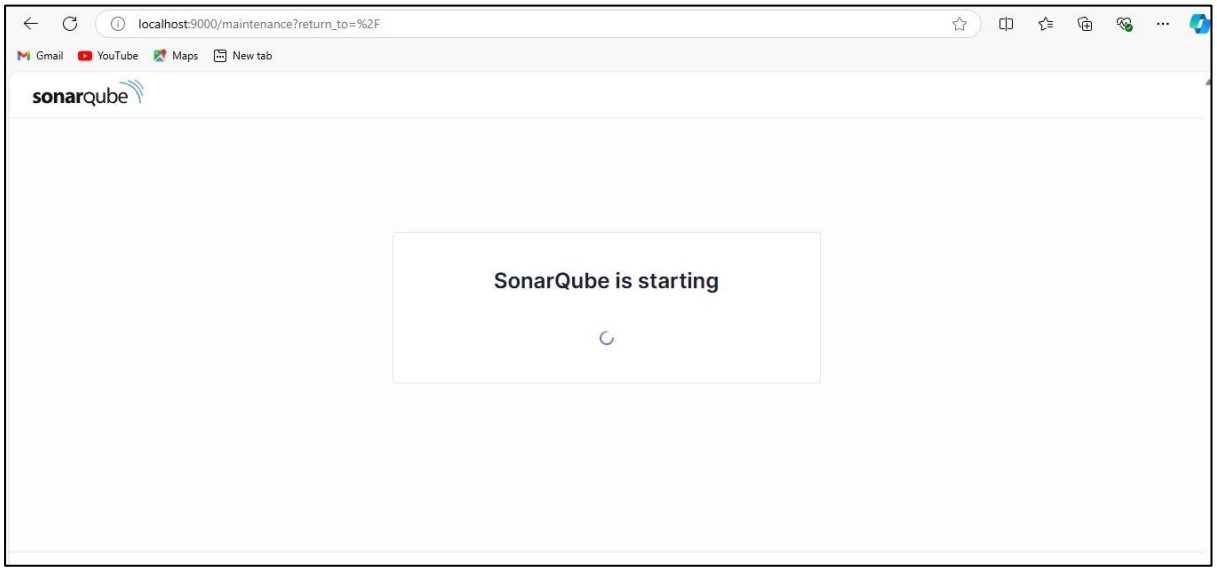
3. Run docker pull SonarQube command to install SonarQube image.

```
C:\Users\Dell>docker pull sonarqube
Using default tag: latest
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
docker.io/library/sonarqube:latest

What's next:
    View a summary of image vulnerabilities and recommendations → docker scout quickview sonarqube
```

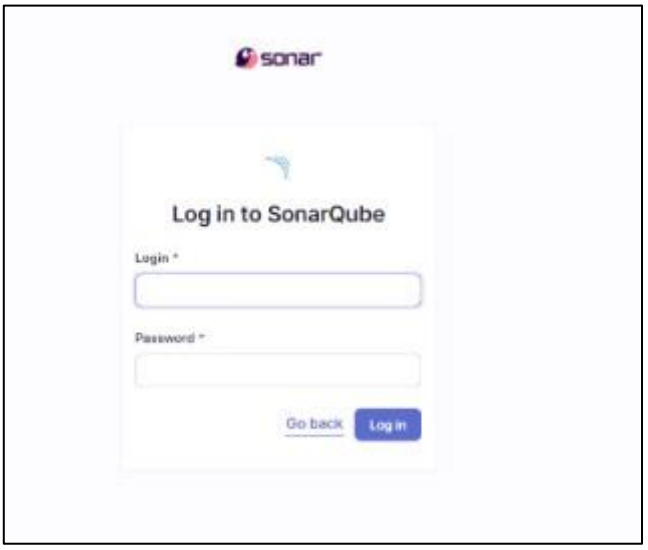
4. Run docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest Command to run the sonarqube.

```
C:\Users\Dell>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
ac1f985dedebc00a642a4c69a502d611389e8f9fa46610febe75aa5021767cab
```

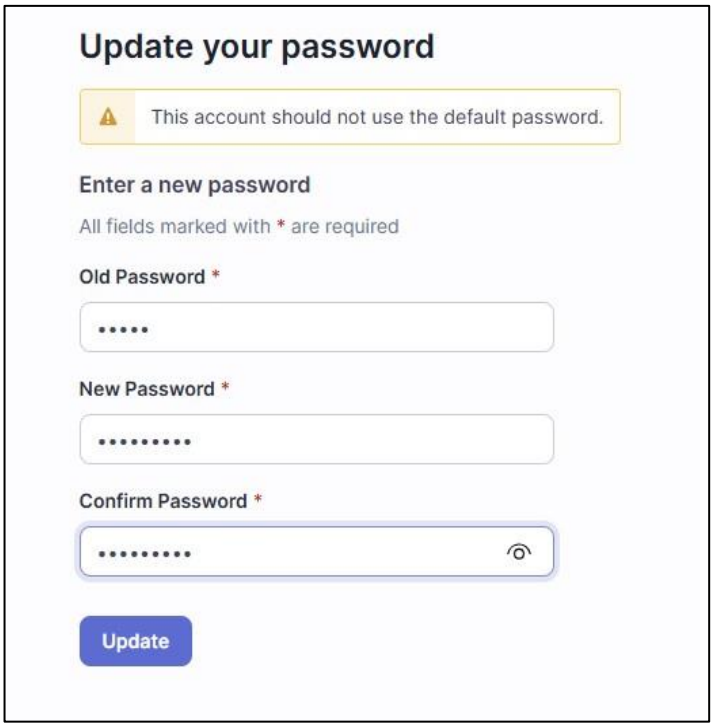
5. Once the container is running go to your web browser and check status of SonarQube at port 9000.



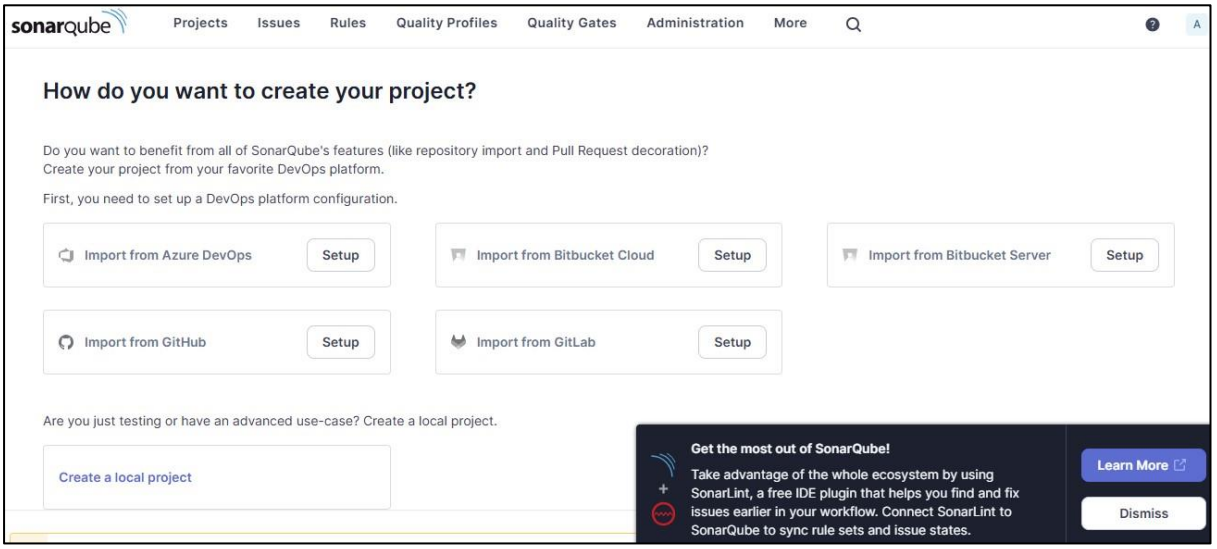
6. Once SonarQube is started it will redirect you to login page. The login and password both for SonarQube is ‘admin’



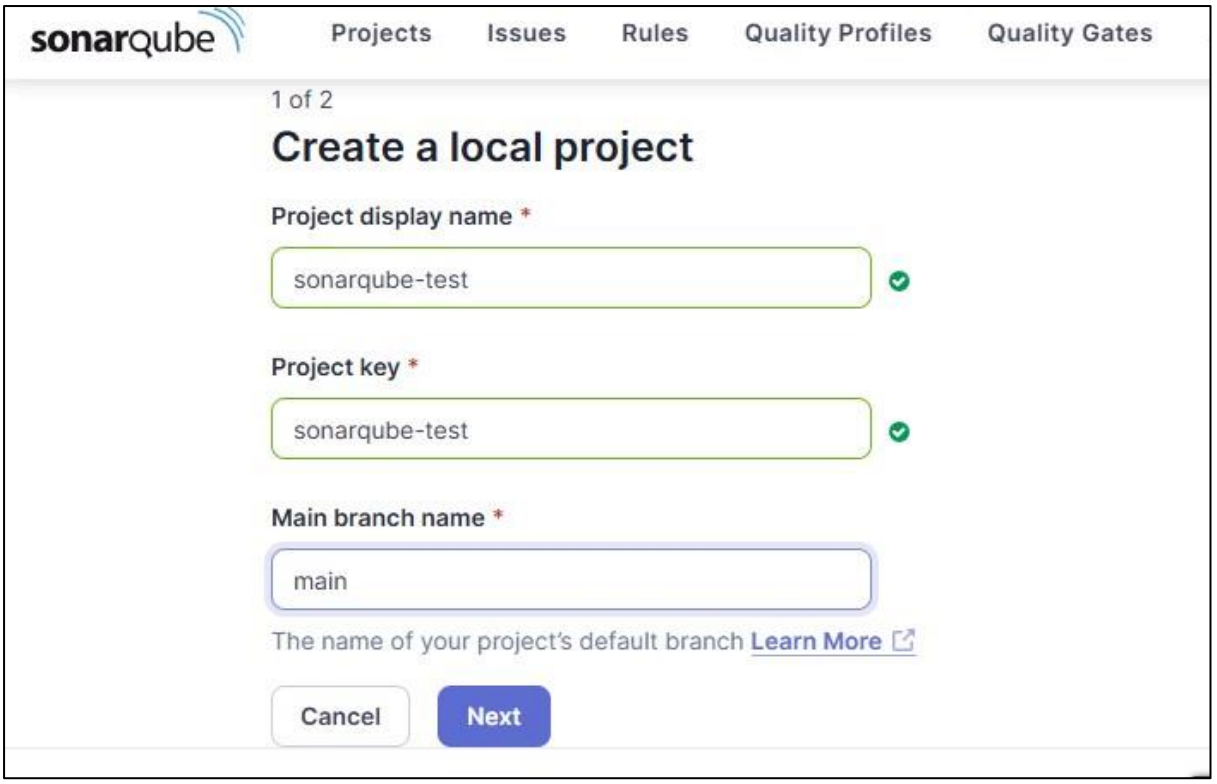
7. Change the password for your SonarQube account.



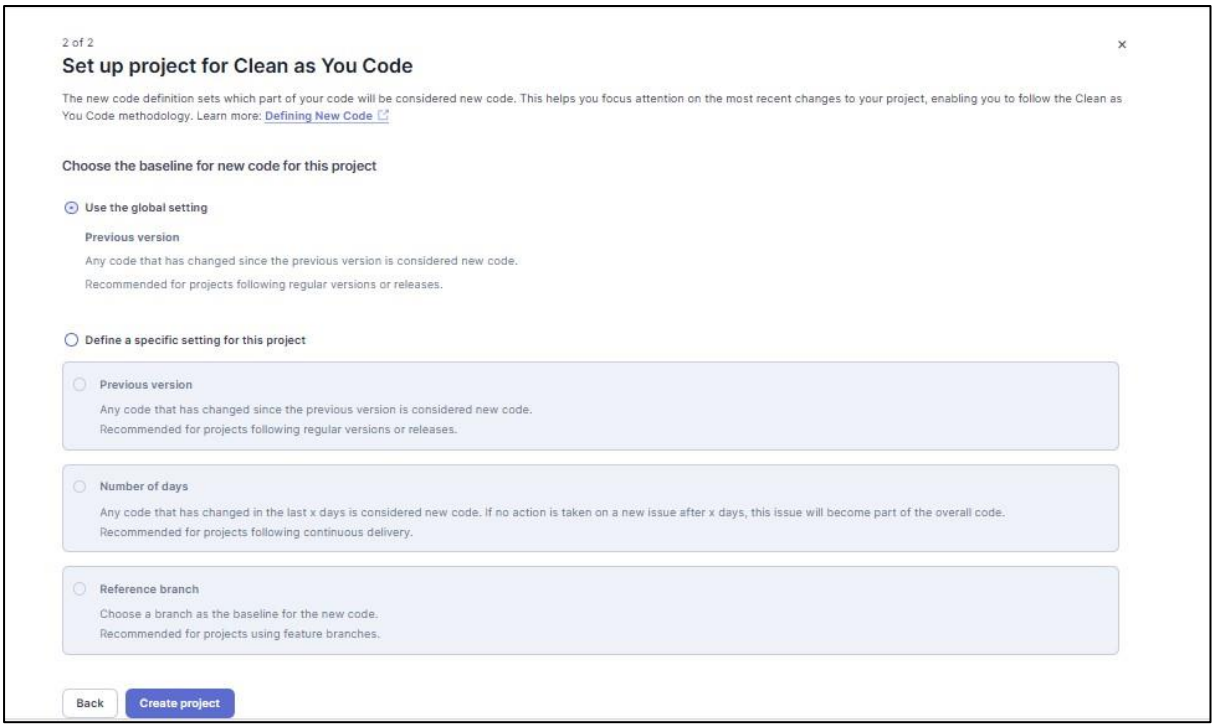
8. After changing the password, you will be directed to this screen. Click on Create a Local Project.



9. Add name of the project and project key and select the main branch name and click on next.

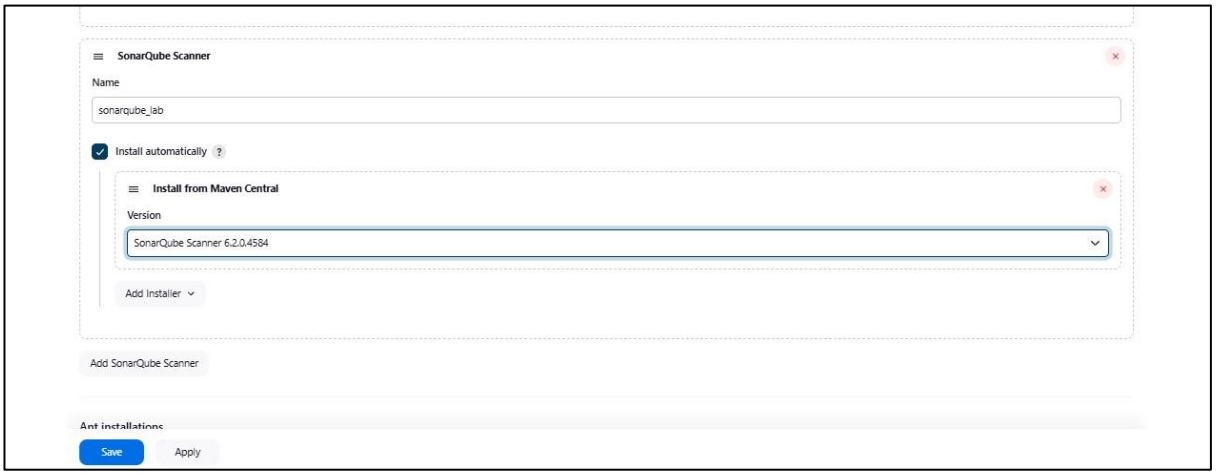


10. Set up the project as required and click on create.

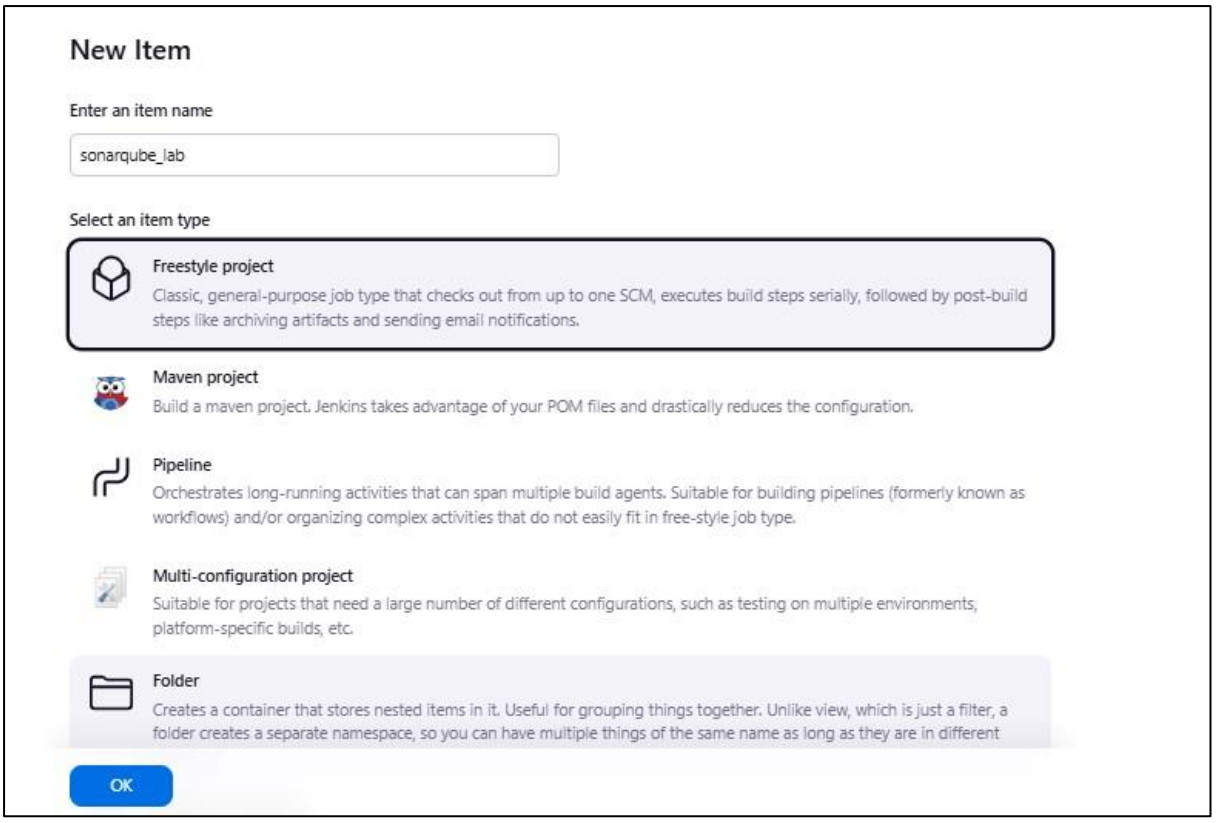


11. Go to Jenkins dashboard->Manage Jenkins->System and scroll down to SonarQube installations. Enter the name and URL in the fields and save the changes.

12. In SonarQube Scanner add the latest version then apply the changes and save it.



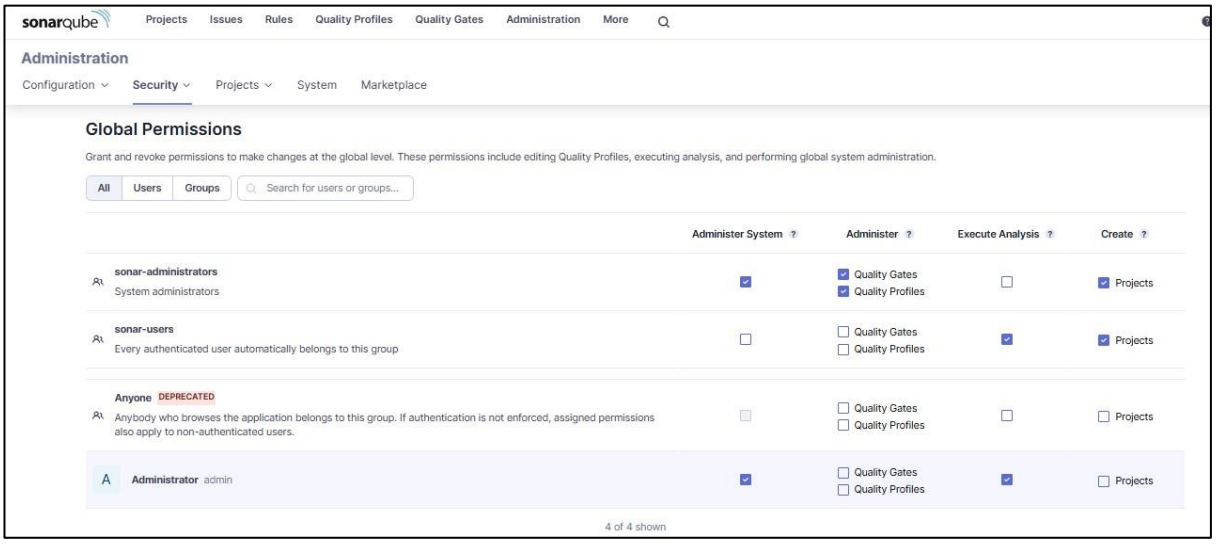
13. Go to Jenkins and then create a new item, enter the item name and select “Freestyle project” and then click on ok.



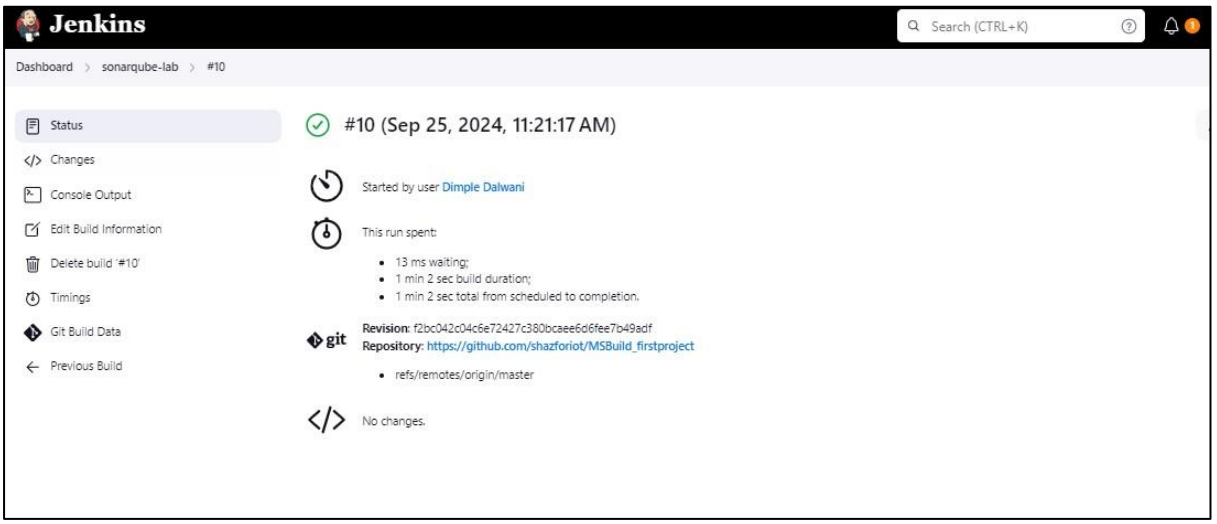
14. Use this GitHub repository in Source Code Management.
https://github.com/shazforiot/MSBuild_firstproject

15. In Analysis properties, mention the SonarQube Project Key, Login, Password, Source path and Host URL.

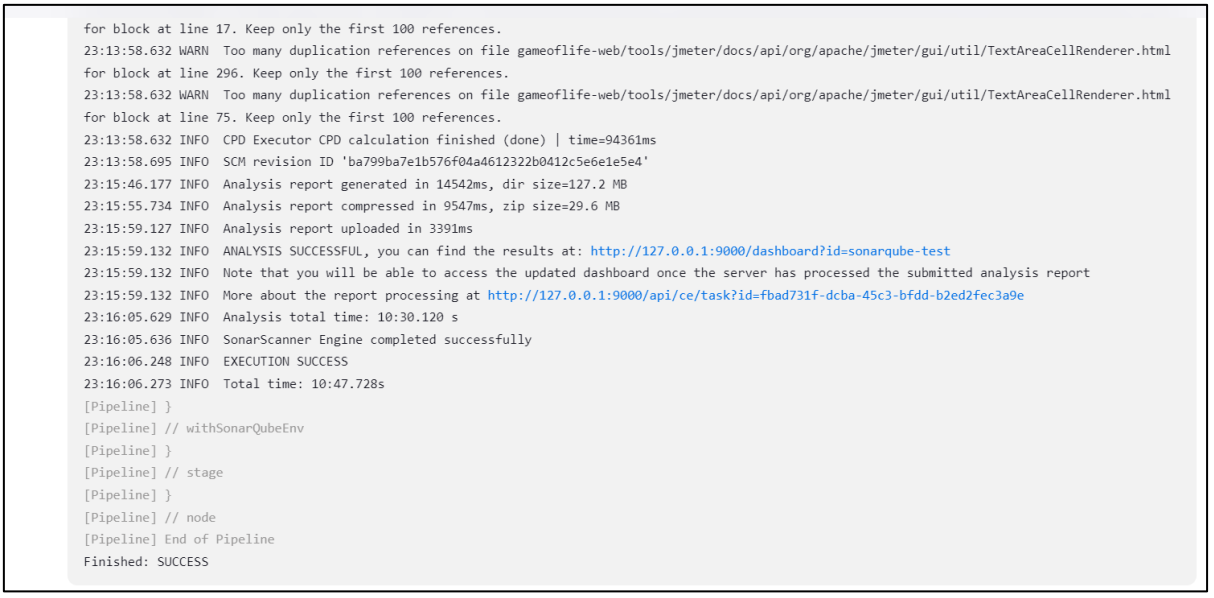
16. Now, you need to grant the local user (here admin user) permissions to Execute the Analysis stage on SonarQube. For this go to http://localhost:<port_number>/admin/permissions and check the ‘Execute Analysis’ checkbox under Administrator.



17. Go to the job you have just built and click on Build Now.

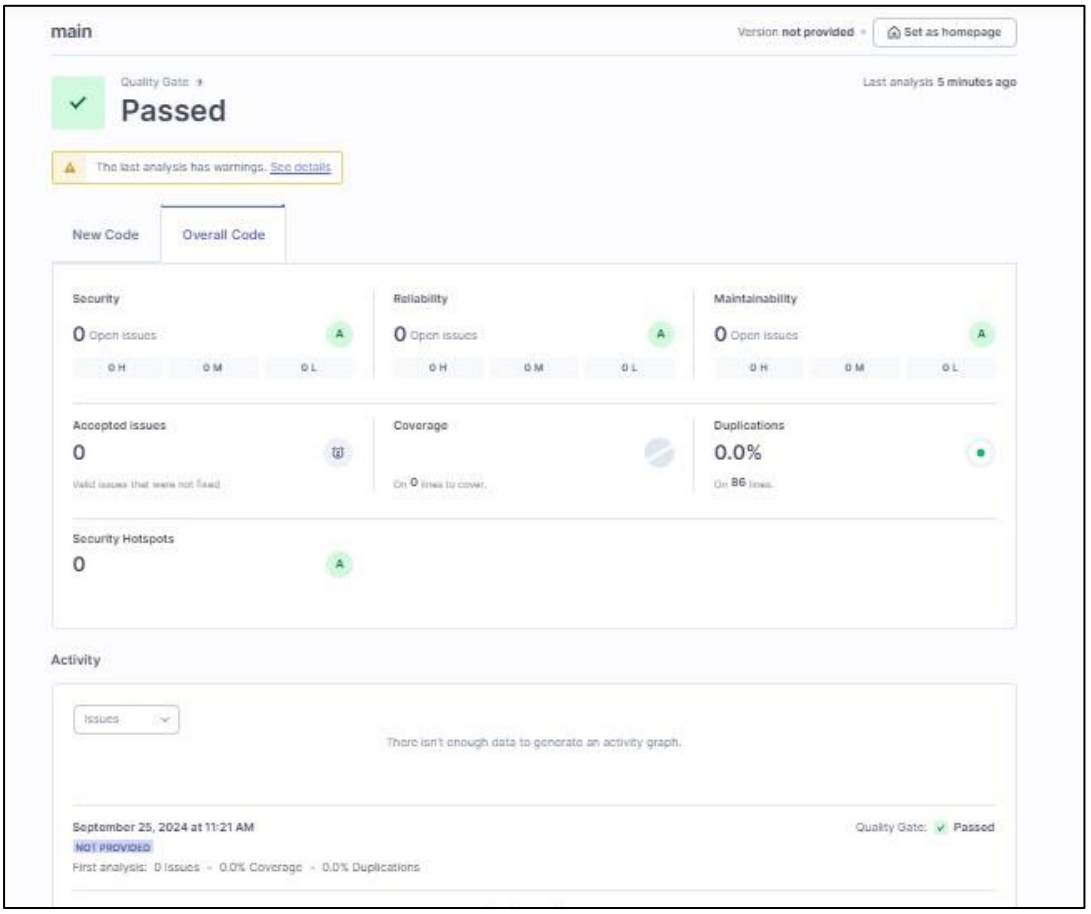


18. Check the console Output



```
for block at line 17. Keep only the first 100 references.
23:13:58.632 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html
for block at line 296. Keep only the first 100 references.
23:13:58.632 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html
for block at line 75. Keep only the first 100 references.
23:13:58.632 INFO  CPD Executor CPD calculation finished (done) | time=94361ms
23:13:58.695 INFO  SCM revision ID 'ba799ba7e1b576f04a4612322b0412c5e6e1e5e4'
23:15:46.177 INFO  Analysis report generated in 14542ms, dir size=127.2 MB
23:15:55.734 INFO  Analysis report compressed in 9547ms, zip size=29.6 MB
23:15:59.127 INFO  Analysis report uploaded in 3391ms
23:15:59.132 INFO  ANALYSIS SUCCESSFUL, you can find the results at: http://127.0.0.1:9000/dashboard?id=sonarqube-test
23:15:59.132 INFO  Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
23:15:59.132 INFO  More about the report processing at http://127.0.0.1:9000/api/ce/task?id=fbad731f-dcba-45c3-bfdd-b2ed2fec3a9e
23:16:05.629 INFO  Analysis total time: 10:30.120 s
23:16:05.636 INFO  SonarScanner Engine completed successfully
23:16:06.248 INFO  EXECUTION SUCCESS
23:16:06.273 INFO  Total time: 10:47.728s
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

19. Go back to SonarQube and check the project.



Conclusion: While performing this experiment there was an issue in creating sonarqube docker image and we resolved it by logging in to the docker desktop and performing it through the terminal. Other than this we created a freestyle project and entered the sonarqube credentials and then performed build.

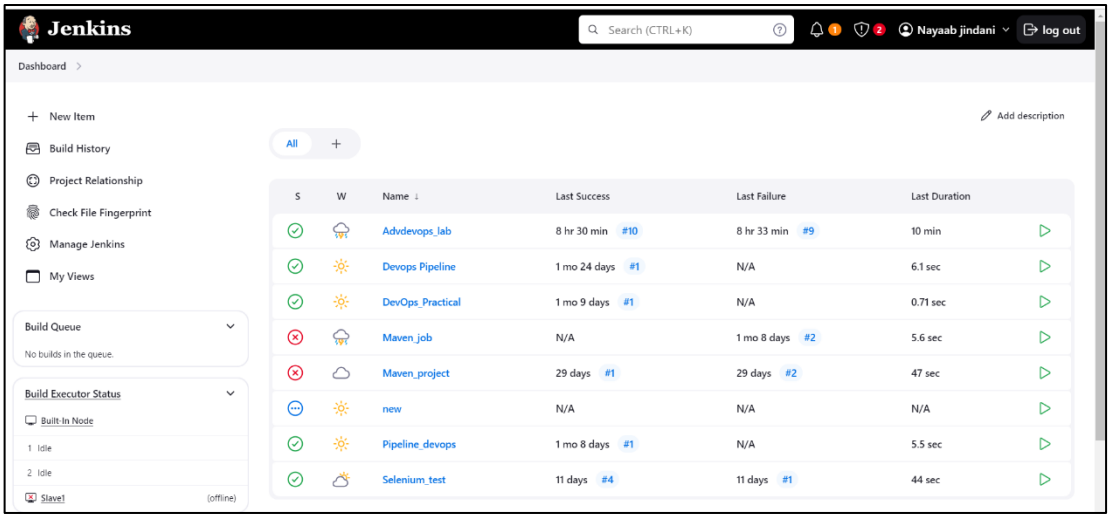
Advance DevOps

Experiment 8

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.

Steps:

1. Open the Jenkins dashboard.



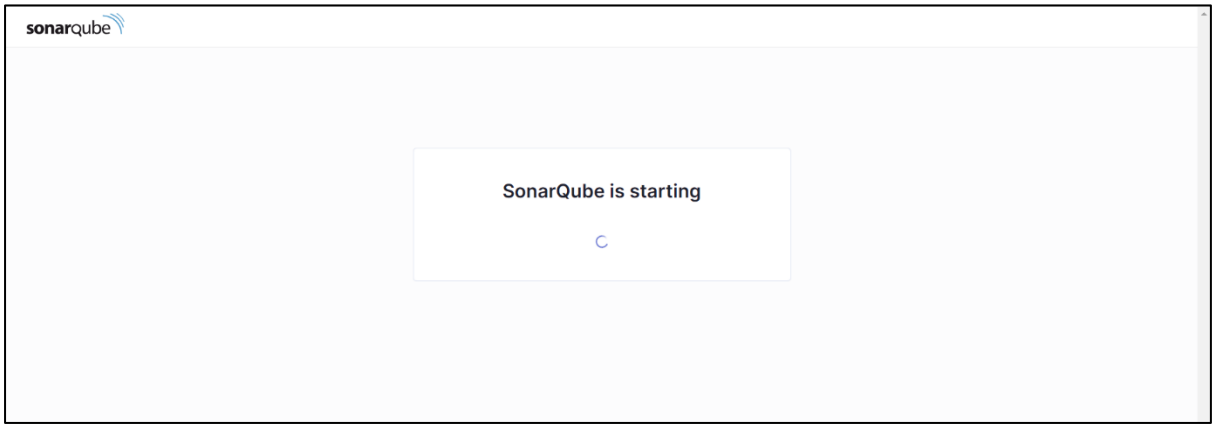
2. First, we will pull the latest version of sonar qube image from the docker hub using the command:
   docker pull sonarqube:latest

```
docker pull sonarqube:latest
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
docker.io/library/sonarqube:latest

What's next:
    View a summary of image vulnerabilities and recommendations → docker scout quickview sonarqube:latest
```

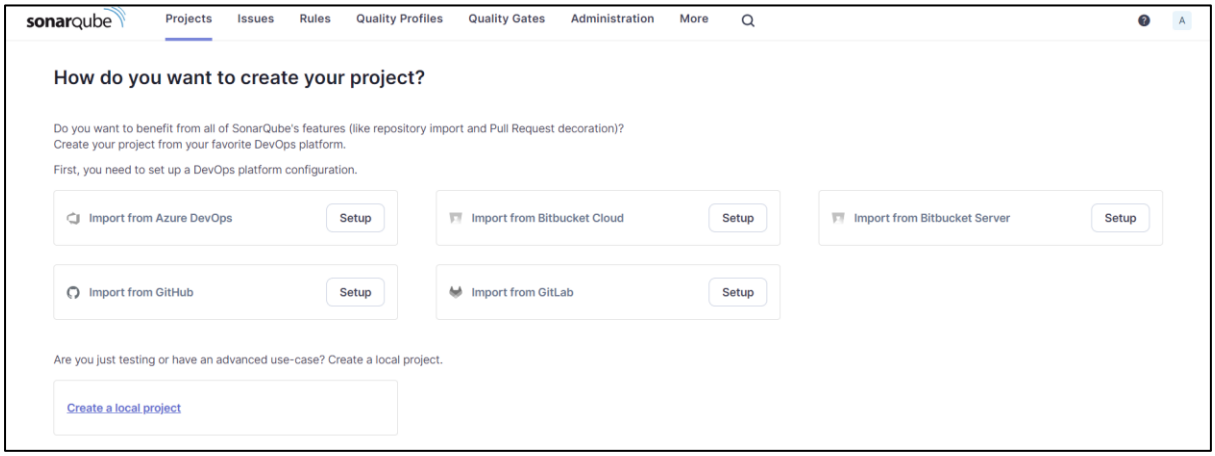
3. Next, we will run SonarQube in a docker container

```
PS C:\Users\Dell> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
67aeea599cfc48e12d50da592eff01d8257f58e6c1bffdf50446066e5f2a8844
```

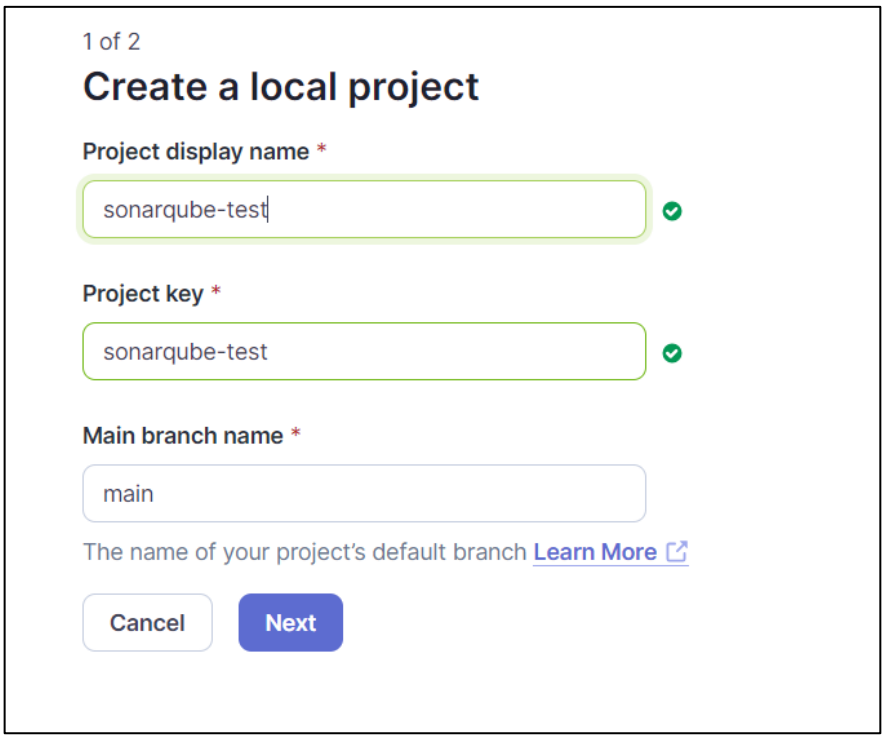
4. Once the container is running, we will check the status of sonar qube on the port 9000. It will show “Sonar qube is starting”



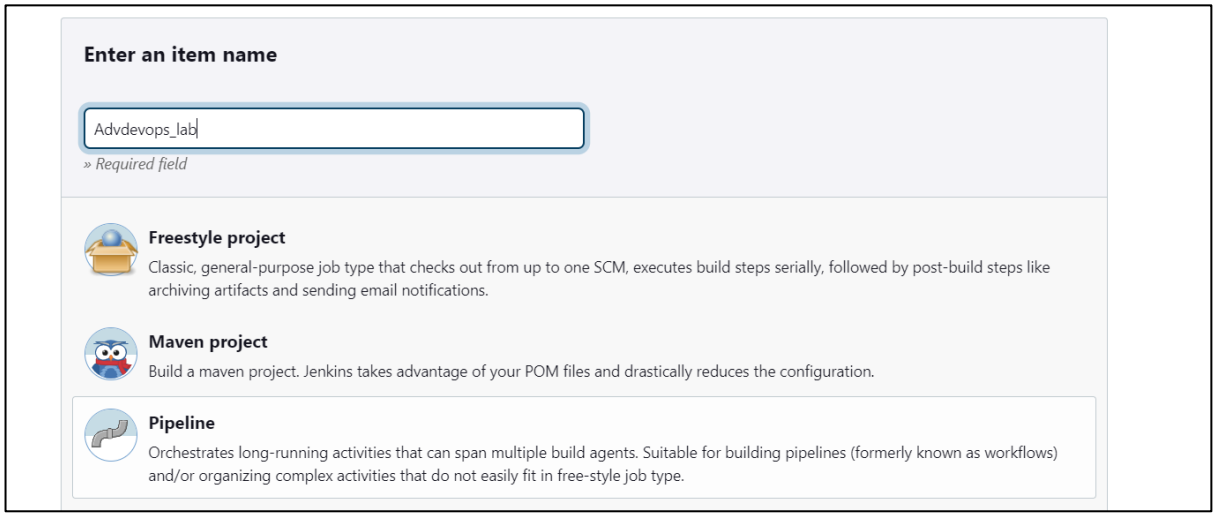
5. Now login to SonarQube using username and password.



6. Click on create a local project option from the dashboard and give a name to the project, click on next and complete the setup.



7. Go to Jenkins dashboard and create a new item by giving a name and select pipeline option.



8. Scroll down to pipeline script and enter the following script:

```
node {
stage('Cloning the GitHub Repo') {
git 'https://github.com/shazforiot/GOL.git'
}
stage('SonarQube analysis') {
withSonarQubeEnv('sonarqube') {
sh "<PATH_TO_SONARQUBE_FOLDER>//bin//sonar-scanner \
-D sonar.login=<SonarQube_USERNAME> \
-D sonar.password=<SonarQube_PASSWORD> \
-D sonar.projectKey=<Project_KEY> \
-D sonar.exclusions=vendor/**,resources/**,**/*.java \
-D sonar.host.url=http://127.0.0.1:9000/"
}
}
}
```

(Change the path and credentials)

9. Now run the build.

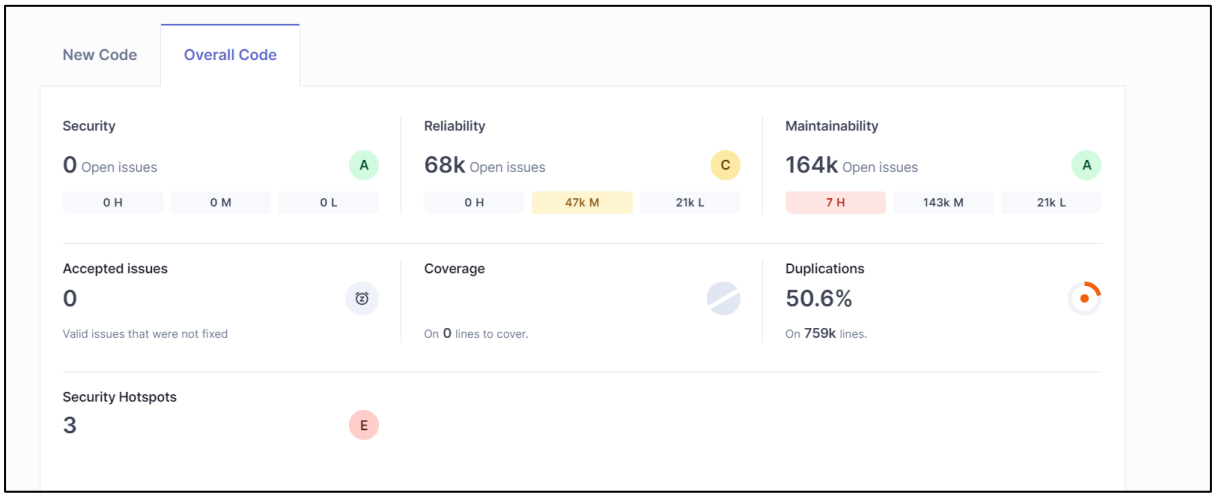
Console Output

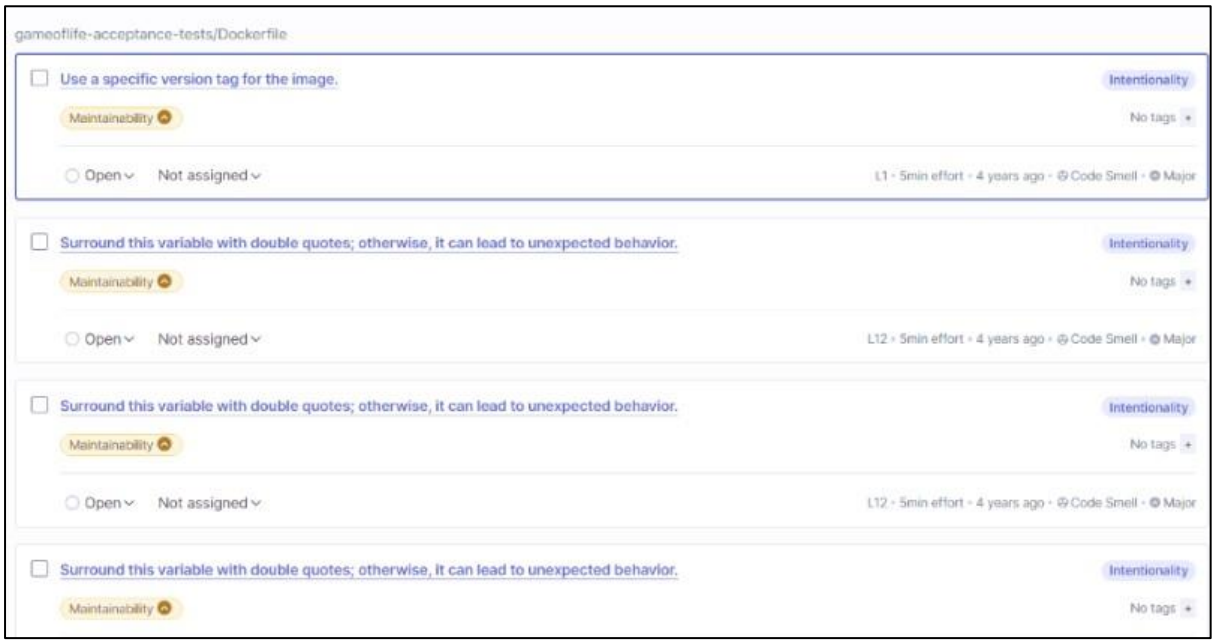
```
Started by user Nayaab jindani
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\Advdevops_lab
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Cloning the GitHub Repo)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
 > git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\Advdevops_lab\.git # timeout=10
Fetching changes from the remote Git repository
 > git.exe config remote.origin.url https://github.com/shazforiot/GOL.git # timeout=10
Fetching upstream changes from https://github.com/shazforiot/GOL.git
 > git.exe --version # timeout=10
 > git --version # 'git version 2.45.2.windows.1'
 > git.exe fetch --tags --force --progress -- https://github.com/shazforiot/GOL.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision ba799ba7e1b576f04a4612322b0412c5e6e1e5e4 (refs/remotes/origin/master)
 > git.exe config core.sparsecheckout # timeout=10
 > git.exe checkout -f ba799ba7e1b576f04a4612322b0412c5e6e1e5e4 # timeout=10
 > git.exe branch -a -v --no-abbrev # timeout=10
 > git.exe branch -D master # timeout=10
```

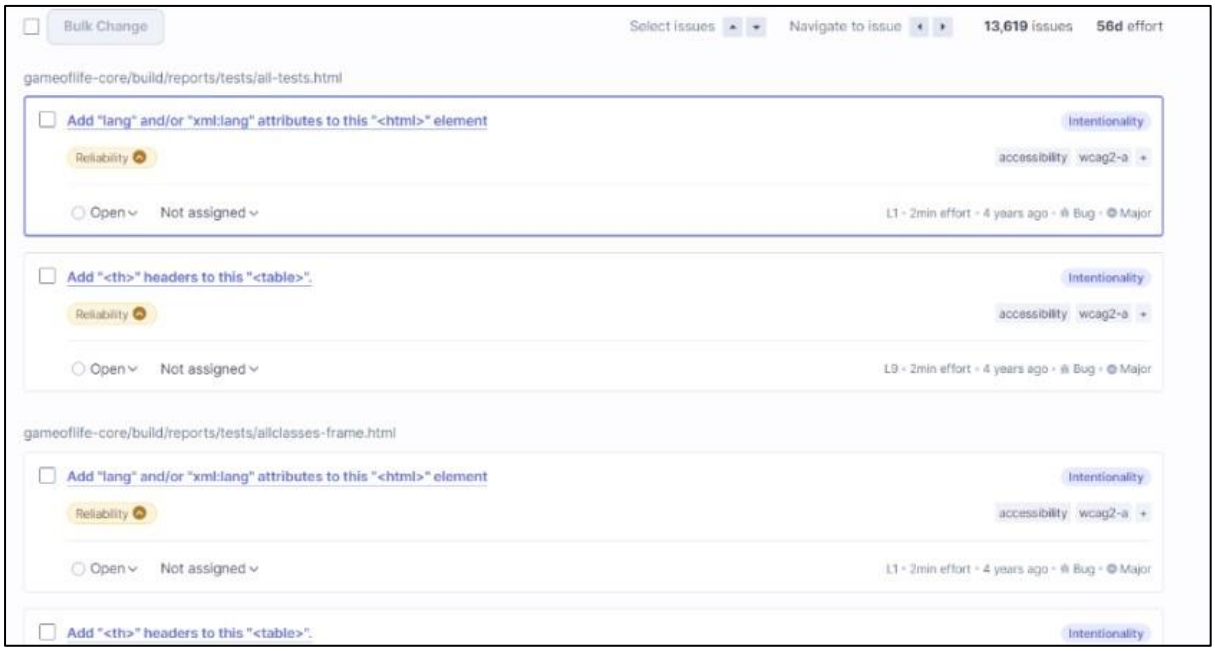
```
for block at line 17. Keep only the first 100 references.
23:13:58.632 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html
for block at line 296. Keep only the first 100 references.
23:13:58.632 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html
for block at line 75. Keep only the first 100 references.
23:13:58.632 INFO  CPD Executor CPD calculation finished (done) | time=94361ms
23:13:58.695 INFO  SCM revision ID 'ba799ba7e1b576f04a4612322b0412c5e6e1e5e4'
23:15:46.177 INFO  Analysis report generated in 14542ms, dir size=127.2 MB
23:15:55.734 INFO  Analysis report compressed in 9547ms, zip size=29.6 MB
23:15:59.127 INFO  Analysis report uploaded in 3391ms
23:15:59.132 INFO  ANALYSIS SUCCESSFUL, you can find the results at: http://127.0.0.1:9000/dashboard?id=sonarqube-test
23:15:59.132 INFO  Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
23:15:59.132 INFO  More about the report processing at http://127.0.0.1:9000/api/ce/task?id=fbad731f-dcba-45c3-bfdd-b2ed2fec3a9e
23:16:05.629 INFO  Analysis total time: 10:30.120 s
23:16:05.636 INFO  SonarScanner Engine completed successfully
23:16:06.248 INFO  EXECUTION SUCCESS
23:16:06.273 INFO  Total time: 10:47.728s
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

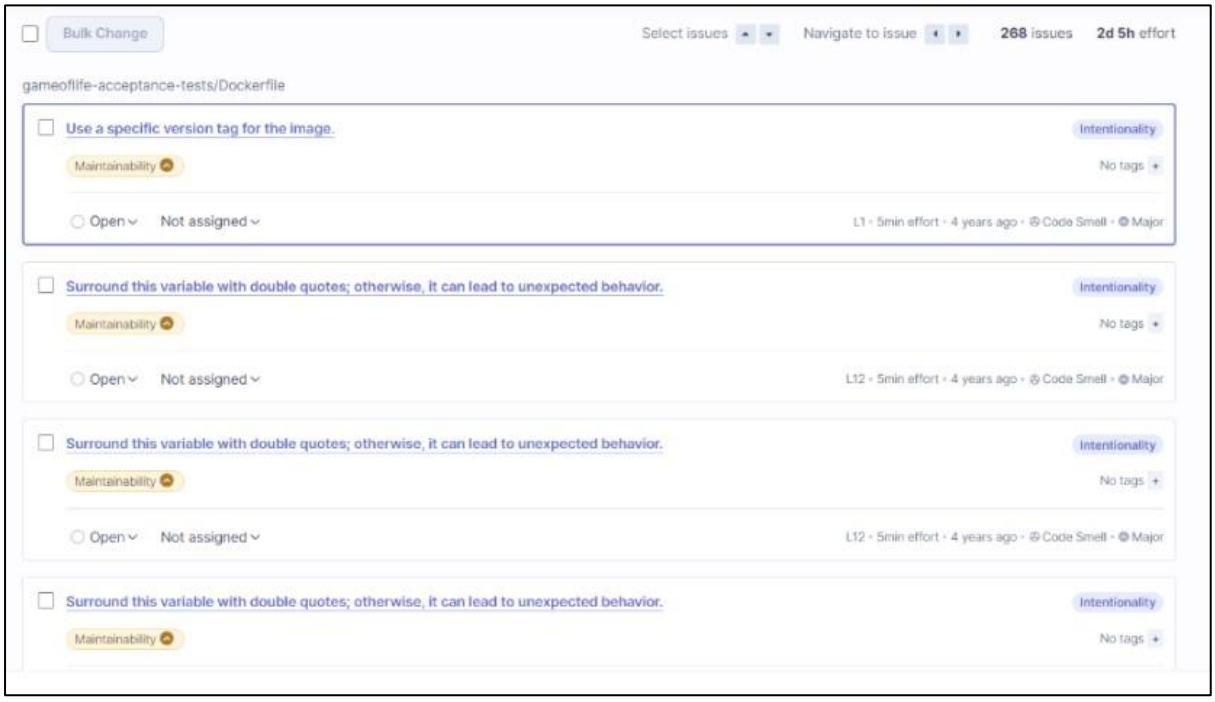
The build is successful.

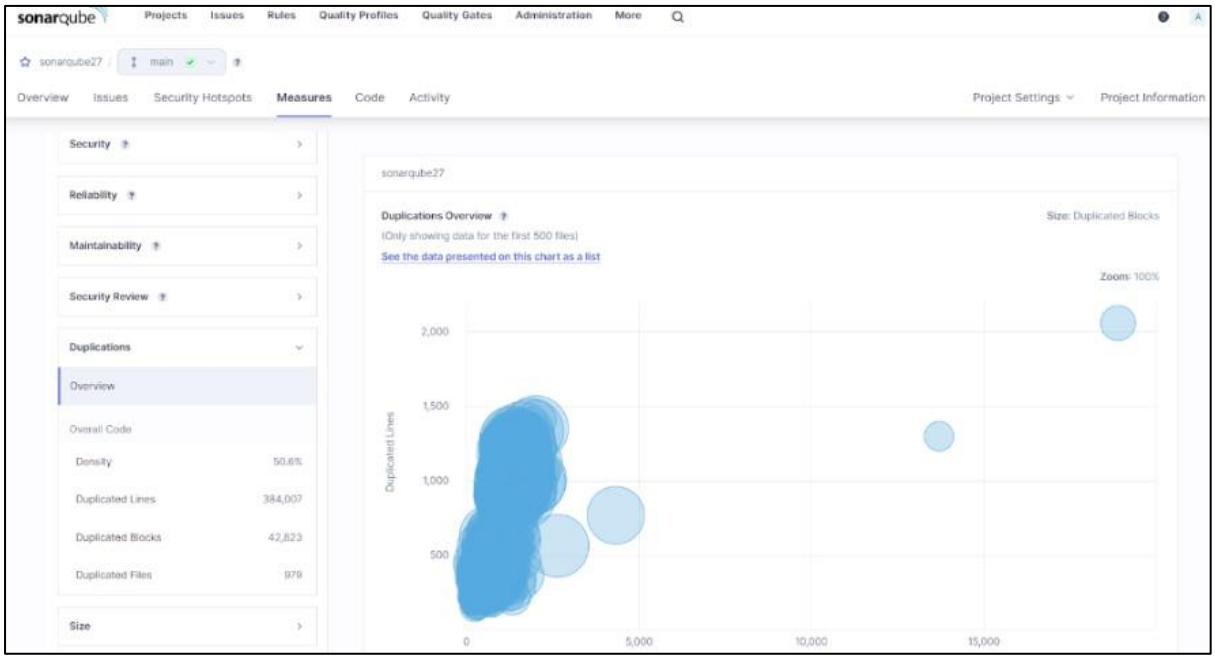
10. Go to sonar qube and check the different errors, code problems, bugs present in the code.






Bulk Change
Select issues
Navigate to issue
13,619 issues
56d effort
gameoflife-core/build/reports/tests/all-tests.html
Add "lang" and/or "xml:lang" attributes to this "<html>" element
Intentionality
Reliability
accessibility wcag2-a
Open
Not assigned
L1 · 2min effort · 4 years ago · Bug · Major
Add "<th>" headers to this "<table>".
Intentionality
Reliability
accessibility wcag2-a
Open
Not assigned
L9 · 2min effort · 4 years ago · Bug · Major
gameoflife-core/build/reports/tests/allclasses-frame.html
Add "lang" and/or "xml:lang" attributes to this "<html>" element
Intentionality
Reliability
accessibility wcag2-a
Open
Not assigned
L1 · 2min effort · 4 years ago · Bug · Major
Add "<th>" headers to this "<table>".
Intentionality


Bulk Change
Select issues
Navigate to issue
268 issues
2d 5h effort
gameoflife-acceptance-tests/Dockerfile
Use a specific version tag for the image.
Intentionality
Maintainability
No tags
Open
Not assigned
L1 · 5min effort · 4 years ago · Code Smell · Major
Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.
Intentionality
Maintainability
No tags
Open
Not assigned
L12 · 5min effort · 4 years ago · Code Smell · Major
Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.
Intentionality
Maintainability
No tags
Open
Not assigned
L12 · 5min effort · 4 years ago · Code Smell · Major
Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.
Intentionality
Maintainability
No tags



Conclusion:

In this experiment we created a Jenkins CICD Pipeline to perform a static analysis of the code to detect bugs, code smells, and security vulnerabilities on a sample code. It is to be checked whether the sonar scanner plugin is installed in Jenkins or not and also provide the correct path and credentials in the pipeline script or else it leads to the failure of the build.

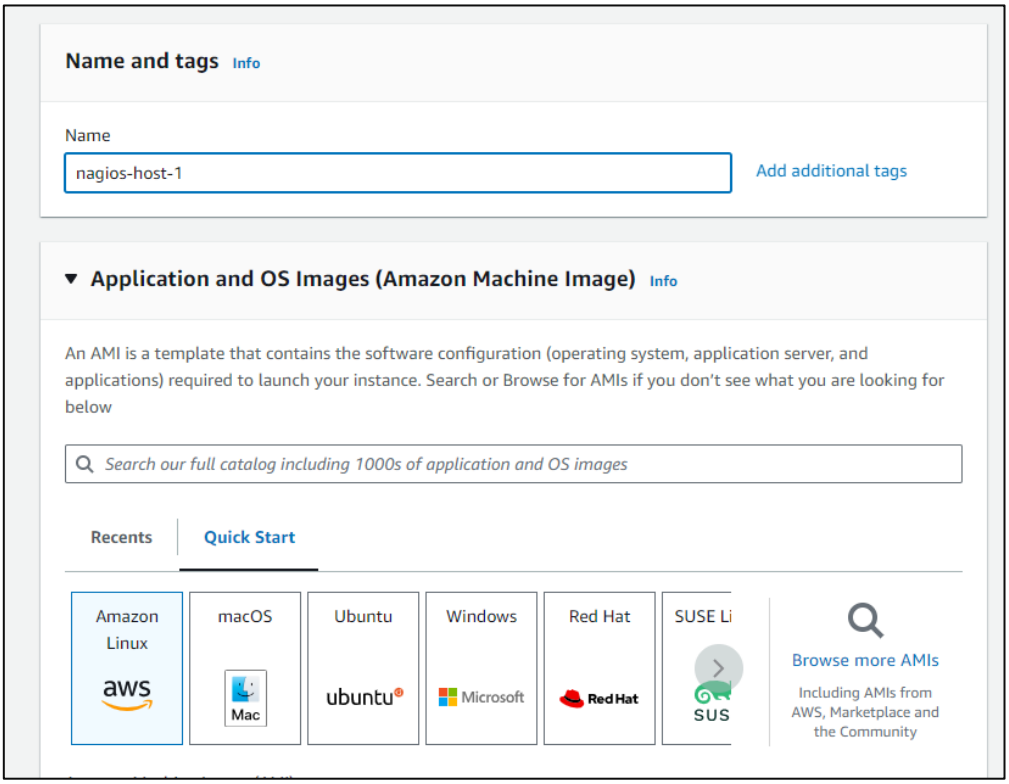
Advance DevOps

Experiment 9

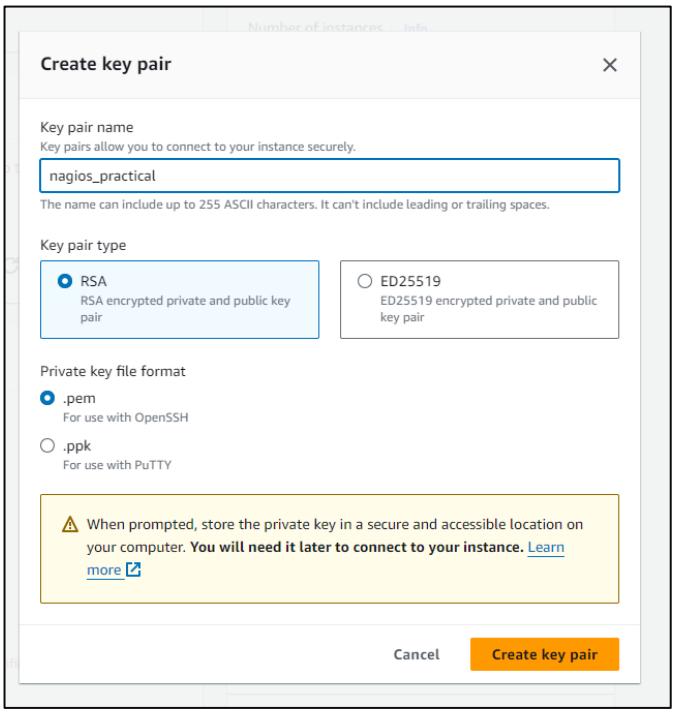
Aim: To understand continuous monitoring and installation and configuration of Nagios Core, Nagios Plugins and NRPE (Nagios Remote Plugin Executor) on Linux Machine.

Steps:

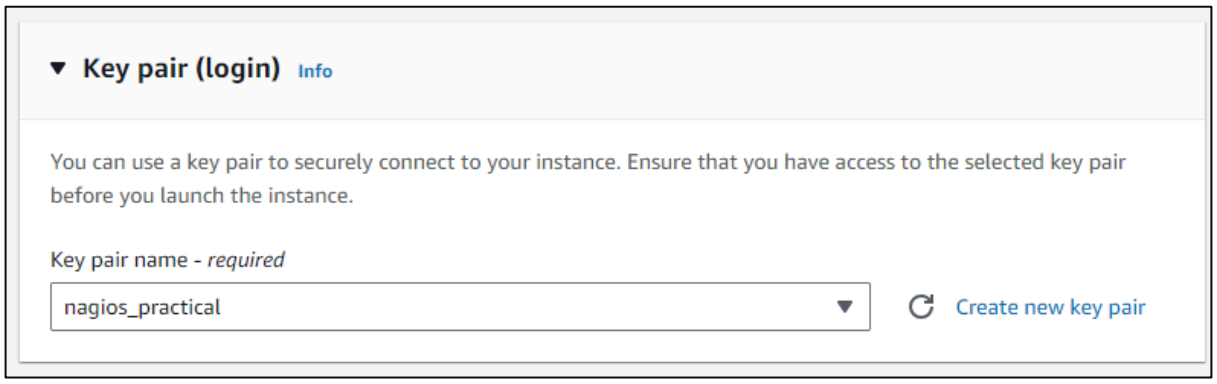
1. Create an ec2 instance and select amazon Linux as the OS



2. Now we will create a key pair.

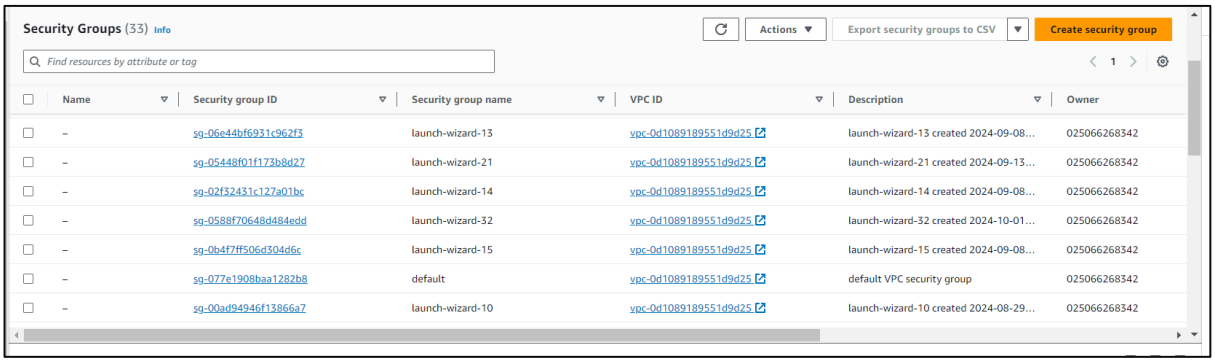


3. Use the created key pair while creating the instance

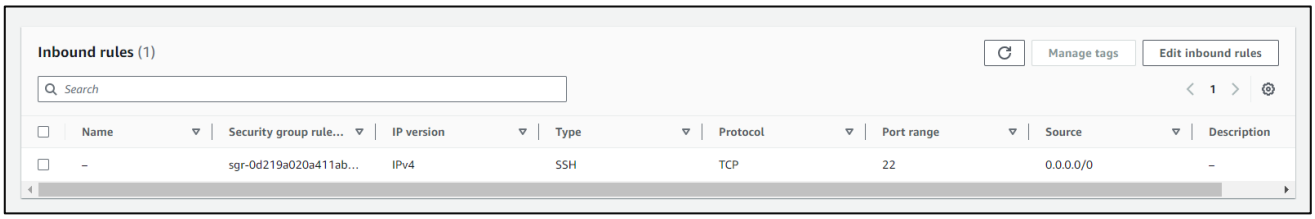


4. Once the instance is successfully initiated go to security groups and select the security group id of the instance you just created.

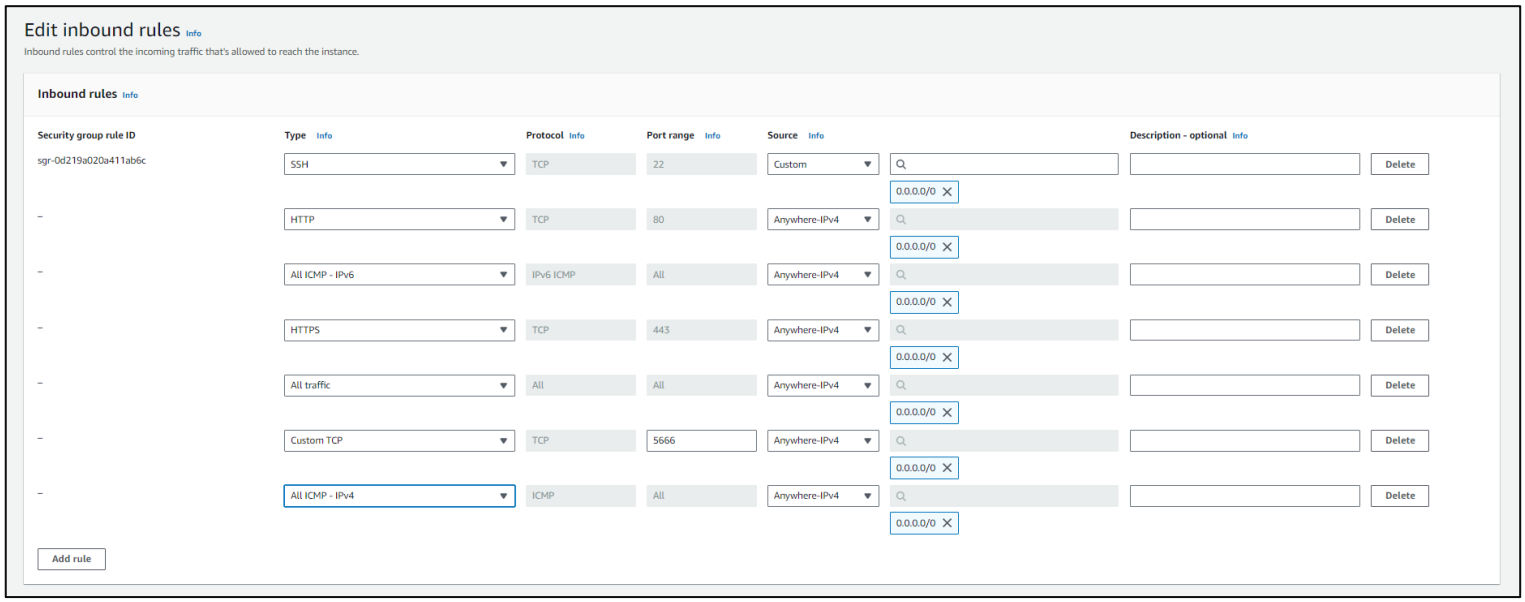
   (The security group name is visible during instance creation and also on the ec2 instances dashboard)



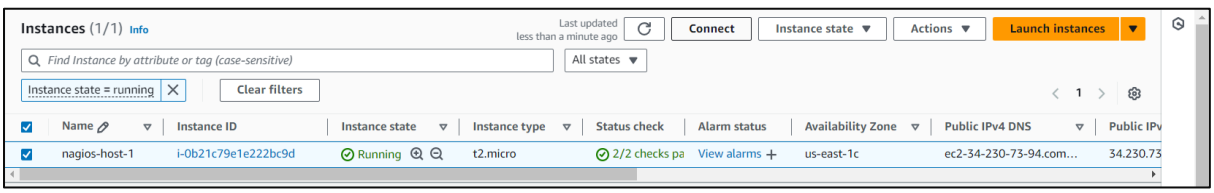
5. To edit the inbound rules select the “Edit inbound rules” button



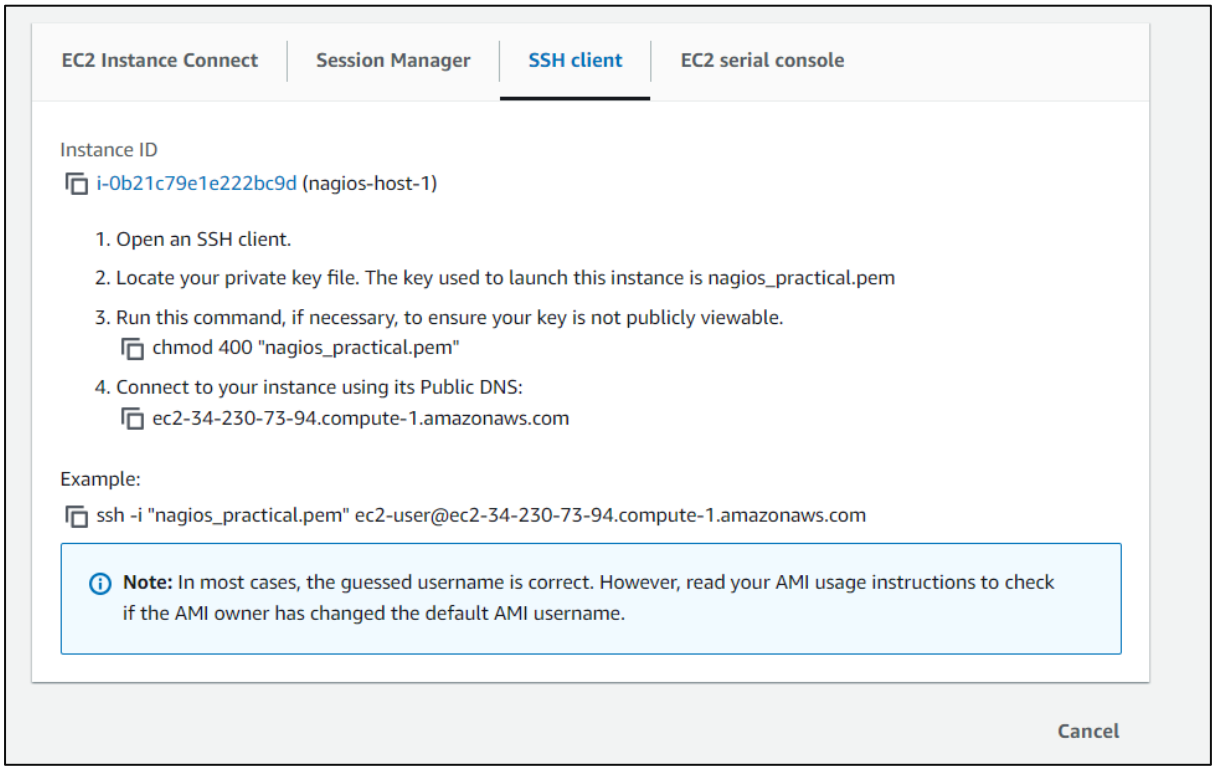
6. Add the rules as given in the screenshot below



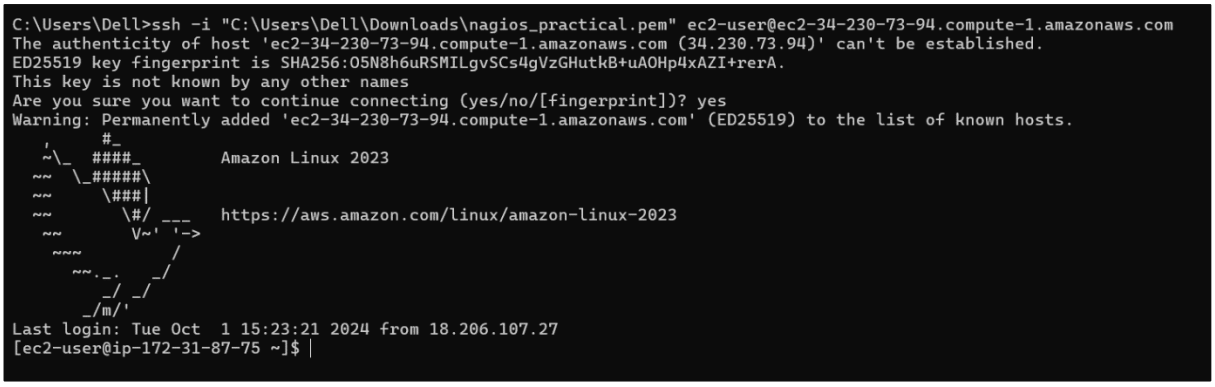
7. Connect the instance



8. Copy the ssh command given in the ssh client section.



9. In your terminal paste the copied command, just replace the .pem file name with the actual location where the .pem file is downloaded in your system



10. Now install the following packages using yum:

sudo yum update

```
[ec2-user@ip-172-31-87-75 ~]$ sudo yum update
Last metadata expiration check: 0:30:09 ago on Tue Oct  1 15:04:44 2024.
Dependencies resolved.
Nothing to do.
Complete!
```

sudo yum install httpd php

```
[ec2-user@ip-172-31-87-75 ~]$ sudo yum install httpd php
Last metadata expiration check: 0:31:22 ago on Tue Oct  1 15:04:44 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
```

```
Installed:
  apr-1.7.2-2.amzn2023.0.2.x86_64
  apr-util-1.6.3-1.amzn2023.0.1.x86_64
  apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
  generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
  httpd-2.4.62-1.amzn2023.x86_64
  httpd-core-2.4.62-1.amzn2023.x86_64
  httpd-filesystem-2.4.62-1.amzn2023.noarch
  httpd-tools-2.4.62-1.amzn2023.x86_64
  libbrotli-1.0.9-4.amzn2023.0.2.x86_64
  libsodium-1.0.19-4.amzn2023.x86_64
  libxslt-1.1.34-5.amzn2023.0.2.x86_64
  mailcap-2.1.49-3.amzn2023.0.3.noarch
  mod_http2-2.0.27-1.amzn2023.0.3.x86_64
  mod_lua-2.4.62-1.amzn2023.x86_64
  nginx-filesystem-1:1.24.0-1.amzn2023.0.4.noarch
  php8.3-8.3.10-1.amzn2023.0.1.x86_64
  php8.3-cli-8.3.10-1.amzn2023.0.1.x86_64
  php8.3-common-8.3.10-1.amzn2023.0.1.x86_64
  php8.3-fpm-8.3.10-1.amzn2023.0.1.x86_64
  php8.3-mbstring-8.3.10-1.amzn2023.0.1.x86_64
  php8.3-opcache-8.3.10-1.amzn2023.0.1.x86_64
  php8.3-pdo-8.3.10-1.amzn2023.0.1.x86_64
  php8.3-process-8.3.10-1.amzn2023.0.1.x86_64
  php8.3-sodium-8.3.10-1.amzn2023.0.1.x86_64
  php8.3-xml-8.3.10-1.amzn2023.0.1.x86_64

Complete!
```

sudo yum install gcc glibc glibc-common

```
[ec2-user@ip-172-31-87-75 ~]$ sudo yum install gcc glibc glibc-common
Last metadata expiration check: 0:35:45 ago on Tue Oct  1 15:04:44 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
```

```
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-172-31-87-75 ~]$ sudo yum install gd gd-devel
Last metadata expiration check: 0:38:32 ago on Tue Oct  1 15:04:44 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
 fontconfig-devel                           x86_64                  2.13.94-2.amzn2023.0.2                     amazonlinux                        128 k
 fonts-filesystem                           noarch                  1:2.0.5-12.amzn2023.0.2                    amazonlinux                        9.5 k
 freetype                                   x86_64                  2.13.2-5.amzn2023.0.1                      amazonlinux                        423 k
 freetype-devel                             x86_64                  2.13.2-5.amzn2023.0.1                      amazonlinux                        912 k
 glib2-devel                                x86_64                  2.74.7-689.amzn2023.0.2                    amazonlinux                        486 k
 google-noto-fonts-common                   noarch                  20201206-2.amzn2023.0.2                    amazonlinux                         15 k
 google-noto-sans-vf-fonts                  noarch                  20201206-2.amzn2023.0.2                    amazonlinux                        492 k
 graphite2                                  x86_64                  1.3.14-7.amzn2023.0.2                      amazonlinux                         97 k
 graphite2-devel                            x86_64                  1.3.14-7.amzn2023.0.2                      amazonlinux                         21 k
 harfbuzz                                   x86_64                  7.0.0-2.amzn2023.0.1                       amazonlinux                        868 k
 harfbuzz-devel                             x86_64                  7.0.0-2.amzn2023.0.1                       amazonlinux                        404 k
 harfbuzz-icu                               x86_64                  7.0.0-2.amzn2023.0.1                       amazonlinux                         18 k
 jbigkit-libs                               x86_64                  2.1-21.amzn2023.0.2                        amazonlinux                         54 k
 langpacks-core-font-en                     noarch                  3.0-21.amzn2023.0.4                        amazonlinux                         10 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!
[ec2-user@ip-172-31-87-75 ~]$
```

11. Create a new Nagios User with its password using the below given commands.

    sudo adduser -m nagios
    sudo passwd nagios

```
[ec2-user@ip-172-31-87-75 ~]$ sudo adduser -m nagios
sudo passwd nagios
Changing password for user nagios.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-87-75 ~]$
```

12. Create a new user group

sudo groupadd nagcmd

```
[ec2-user@ip-172-31-87-75 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-87-75 ~]$
```

13. Next execute 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-172-31-87-75 ~]$ sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
[ec2-user@ip-172-31-87-75 ~]$
```

14. Create a new directory for Nagios downloads

mkdir ~/downloads
cd ~/downloads

```
[ec2-user@ip-172-31-87-75 ~]$ mkdir ~/downloads
cd ~/downloads
[ec2-user@ip-172-31-87-75 downloads]$
```

15. 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-172-31-87-75 downloads]$ wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
--2024-10-01 15:55:47--  https://assets.nagios.com/downloads/nagioscore/releases/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... connected.
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.gz                 100%[=========================================================================>]   1.97M  5.54MB/s    in 0.4s

2024-10-01 15:55:47 (5.54 MB/s) - 'nagios-4.5.5.tar.gz' saved [2065473/2065473]

[ec2-user@ip-172-31-87-75 downloads]$
```

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

```
[ec2-user@ip-172-31-87-75 downloads]$ wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
--2024-10-01 15:57:19--  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-2.4.11.tar.gz          100%[=================================================================================>]   2.62M  4.14MB/s    in 0.6s

2024-10-01 15:57:20 (4.14 MB/s) - 'nagios-plugins-2.4.11.tar.gz' saved [2753049/2753049]
```

16. Use tar to unzip and change to that directory.

tar zxvf nagios-4.5.5.tar.gz

```
[ec2-user@ip-172-31-87-75 downloads]$ tar zxvf nagios-4.5.5.tar.gz
nagios-4.5.5/
nagios-4.5.5/.github/
nagios-4.5.5/.github/workflows/
nagios-4.5.5/.github/workflows/test.yml
nagios-4.5.5/.gitignore
nagios-4.5.5/CONTRIBUTING.md
nagios-4.5.5/Changelog
nagios-4.5.5/INSTALLING
nagios-4.5.5/LEGAL
nagios-4.5.5/LICENSE
nagios-4.5.5/Makefile.in
nagios-4.5.5/README.md
nagios-4.5.5/THANKS
nagios-4.5.5/UPGRADING
nagios-4.5.5/aclocal.m4
nagios-4.5.5/autoconf-macros/
nagios-4.5.5/autoconf-macros/.gitignore
nagios-4.5.5/autoconf-macros/CHANGELOG.md
nagios-4.5.5/autoconf-macros/LICENSE
nagios-4.5.5/autoconf-macros/LICENSE.md
nagios-4.5.5/autoconf-macros/README.md
nagios-4.5.5/autoconf-macros/add_group_user
nagios-4.5.5/autoconf-macros/ax_nagios_get_distrib
nagios-4.5.5/autoconf-macros/ax_nagios_get_files
```

17. We have to now change the directory to nagios-4.5.5, for this first verify whether nagios-4.5.5 exists by using ls command.

```
[ec2-user@ip-172-31-87-75 downloads]$ ls
nagios-4.5.5  nagios-4.5.5.tar.gz  nagios-plugins-2.4.11.tar.gz
```

18. As nagios-4.5.5 is present we will now use cd command to change directory.

```
[ec2-user@ip-172-31-87-75 downloads]$ cd nagios-4.5.5
[ec2-user@ip-172-31-87-75 nagios-4.5.5]$
```

19. Now we will install openssl dev library by using the command:
sudo yum install openssl-devel

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

Total download size: 3.0 M
Installed size: 4.7 M
Is this ok [y/N]: y
Downloading Packages:
openssl-devel-3.0.8-1.amzn2023.0.14.x86_64.rpm                                                                                  23 MB/s | 3.0 MB     00:00
----------------------------------------------------------------------------------------------------------------------------------------------------------------
Total                                                                                                                           18 MB/s | 3.0 MB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                                                                                                       1/1
  Installing       : openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64                                                                                          1/1
  Running scriptlet: openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64                                                                                          1/1
  Verifying        : openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64                                                                                          1/1

Installed:
  openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64
```

20. Run the configuration script

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

```
[ec2-user@ip-172-31-87-75 nagios-4.5.5]$ ./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
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 the compiler supports GNU C... yes
checking whether gcc accepts -g... yes
checking for gcc option to enable C11 features... none needed
checking whether make sets $(MAKE)... yes
checking whether ln -s works... yes
checking for strip... /usr/bin/strip
checking for sys/wait.h that is POSIX.1 compatible... yes
checking for stdio.h... yes
checking for stdlib.h... yes
checking for string.h... yes
```

```
*** Configuration summary for nagios 4.5.5 2024-09-17 ***:

 General Options:
 -------------------------
        Nagios executable:  nagios
        Nagios user/group:  nagios,nagios
       Command user/group:  nagios,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.
```

21. To compile the source code run “make all”

```
[ec2-user@ip-172-31-87-75 nagios-4.5.5]$ make all
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-overflow=]
  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_CONFIG_H -DNSCORE -c -o commands.o commands.c
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE_CONFIG_H -DNSCORE -c -o events.o events.c
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE_CONFIG_H -DNSCORE -c -o flapping.o flapping.c
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE_CONFIG_H -DNSCORE -c -o logging.o logging.c
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE_CONFIG_H -DNSCORE -c -o macros-base.o ../common/macros.c
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE_CONFIG_H -DNSCORE -c -o netutils.o netutils.c
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE_CONFIG_H -DNSCORE -c -o notifications.o notifications.c
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE_CONFIG_H -DNSCORE -c -o sehandlers.o sehandlers.c
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE_CONFIG_H -DNSCORE -c -o utils.o utils.c
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE_CONFIG_H -DNSCORE -c -o retention-base.o ./sretention.c
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE_CONFIG_H -DNSCORE -c -o xretention-base.o ../xdata/xrddefault.c
gcc -Wall -I.. -I. -I../lib -I../include  -I../include -I..  -g -O2  -DHAVE_CONFIG_H -DNSCORE -c -o comments-base.o ../common/comments.c
```

22. To install binaries, init script and sample config files run

    sudo make install

    sudo make install-init

    sudo make install-config

    sudo make install-commandmode

```
[ec2-user@ip-172-31-87-75 nagios-4.5.5]$ sudo make install
sudo make install-init
sudo make install-config
sudo make install-commandmode
cd ./base && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagios /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagiostats /usr/local/nagios/bin
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/base'
cd ./cgi && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
make install-basic
make[2]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/sbin
for file in *.cgi; do \
        /usr/bin/install -c -s -m 775 -o nagios -g nagios $file /usr/local/nagios/sbin; \
done
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
cd ./html && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/html'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/media
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/stylesheets
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/contexthelp
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/docs
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/docs/images
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/js
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/images
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/images/logos
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/includes
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/ssi
/usr/bin/install -c -m 664 -o nagios -g nagios ./robots.txt /usr/local/nagios/share
/usr/bin/install -c -m 664 -o nagios -g nagios ./jsonquery.html /usr/local/nagios/share
rm -f /usr/local/nagios/share/index.html
rm -f /usr/local/nagios/share/main.html
rm -f /usr/local/nagios/share/side.html
```

23. In the config file edit 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                   d2022.nayaab.jindani@ves.ac.in ; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS ******
}
```

24. To configure the web interface run:

    sudo make install-webconf

```
[ec2-user@ip-172-31-87-75 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/nagios.conf; \
fi

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

[ec2-user@ip-172-31-87-75 nagios-4.5.5]$
```

25. Create a nagios admin account and password
    sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin

```
[ec2-user@ip-172-31-87-75 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
```

26. Restart apache

    sudo service httpd restart

```
[ec2-user@ip-172-31-87-75 nagios-4.5.5]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-87-75 nagios-4.5.5]$
```

27. Go back to the downloads folder by using “cd ~/downloads” and unzip the plugins zip file using

    tar zxvf nagios-plugins-2.4.11.tar.gz

```
[ec2-user@ip-172-31-87-75 nagios-4.5.5]$ cd ~/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/
nagios-plugins-2.4.11/config_test/Makefile
nagios-plugins-2.4.11/config_test/run_tests
nagios-plugins-2.4.11/config_test/child_test.c
nagios-plugins-2.4.11/gl/
```

28. Compile and install the plugins

    cd nagios-plugins-2.0.3

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

```
[ec2-user@ip-172-31-87-75 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
```

make

sudo make install

```
[ec2-user@ip-172-31-87-75 nagios-plugins-2.4.11]$ sudo make install
Making install in gl
make[1]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/gl'
make  install-recursive
make[2]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/gl'
make[3]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/gl'
make[4]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/gl'
if test yes = no; then \
  case 'linux-gnu' in \
    darwin[56]*) \
      need_charset_alias=true ;; \
    darwin* | cygwin* | mingw* | pw32* | cegcc*) \
      need_charset_alias=false ;; \
    *) \
      need_charset_alias=true ;; \
  esac ; \
else \
  need_charset_alias=false ; \
fi ; \
if $need_charset_alias; then \
  /bin/sh ../build-aux/mkinstalldirs /usr/local/nagios/lib ; \
fi ; \
if test -f /usr/local/nagios/lib/charset.alias; then \
  sed -f ref-add.sed /usr/local/nagios/lib/charset.alias > /usr/local/nagios/lib/charset.tmp ; \
  /usr/bin/install -c -o nagios -g nagios -m 644 /usr/local/nagios/lib/charset.tmp /usr/local/nagios/lib/charset.alias ; \
  rm -f /usr/local/nagios/lib/charset.tmp ; \
else \
  if $need_charset_alias; then \
    sed -f ref-add.sed charset.alias > /usr/local/nagios/lib/charset.tmp ; \
    /usr/bin/install -c -o nagios -g nagios -m 644 /usr/local/nagios/lib/charset.tmp /usr/local/nagios/lib/charset.alias ; \
```

29. Run below given commands to start nagios:
    sudo chkconfig --add nagios
    sudo chkconfig nagios on

```
[ec2-user@ip-172-31-87-75 nagios-plugins-2.4.11]$ sudo chkconfig --add nagios
sudo chkconfig nagios on
error reading information on service nagios: No such file or directory
Note: Forwarding request to 'systemctl enable nagios.service'.
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service → /usr/lib/systemd/system/nagios.service.
```

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

```
[ec2-user@ip-172-31-87-75 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

Things look okay - No serious problems were detected during the pre-flight check
[ec2-user@ip-172-31-87-75 nagios-plugins-2.4.11]$
```

If the message says no errors detected then run “sudo service nagios start”

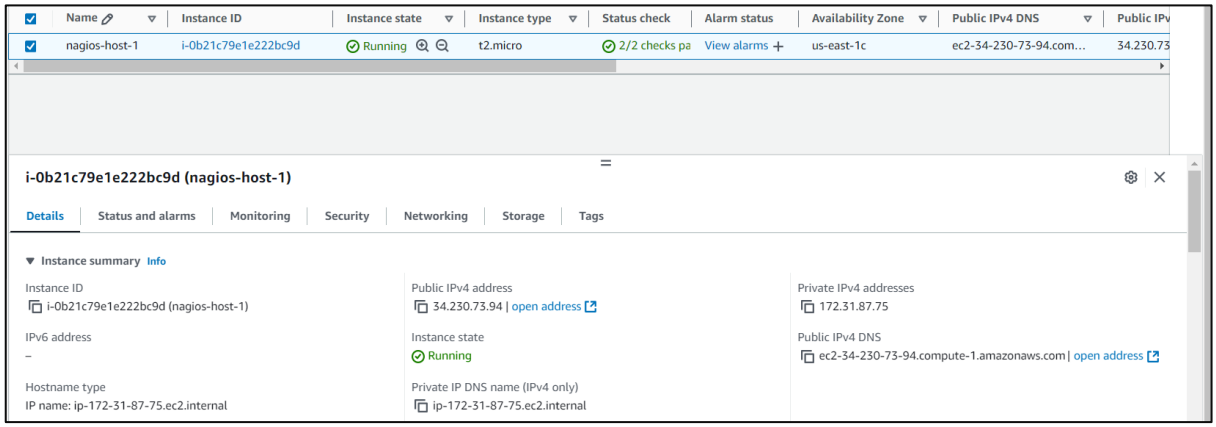
```
[ec2-user@ip-172-31-87-75 nagios-plugins-2.4.11]$ sudo service nagios start
Redirecting to /bin/systemctl start nagios.service
[ec2-user@ip-172-31-87-75 nagios-plugins-2.4.11]$
```

sudo systemctl status nagios

```
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors:   0

Things look okay - No serious problems were detected during the pre-flight check
[ec2-user@ip-172-31-87-75 nagios-plugins-2.4.11]$ sudo service nagios start
Redirecting to /bin/systemctl start nagios.service
[ec2-user@ip-172-31-87-75 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 Tue 2024-10-01 16:32:27 UTC; 48s ago
       Docs: https://www.nagios.org/documentation
    Process: 66684 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
    Process: 66693 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Main PID: 66694 (nagios)
      Tasks: 6 (limit: 1112)
     Memory: 5.7M
        CPU: 82ms
     CGroup: /system.slice/nagios.service
             ├─66694 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─66695 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─66696 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─66697 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─66698 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─66739 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 01 16:32:27 ip-172-31-87-75.ec2.internal nagios[66694]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfully initialized
Oct 01 16:32:27 ip-172-31-87-75.ec2.internal nagios[66694]: qh: core query handler registered
Oct 01 16:32:27 ip-172-31-87-75.ec2.internal nagios[66694]: qh: echo service query handler registered
Oct 01 16:32:27 ip-172-31-87-75.ec2.internal nagios[66694]: qh: help for the query handler registered
Oct 01 16:32:27 ip-172-31-87-75.ec2.internal nagios[66694]: wproc: Successfully registered manager as @wproc with query handler
Oct 01 16:32:27 ip-172-31-87-75.ec2.internal nagios[66694]: wproc: Registry request: name=Core Worker 66696;pid=66696
Oct 01 16:32:27 ip-172-31-87-75.ec2.internal nagios[66694]: wproc: Registry request: name=Core Worker 66695;pid=66695
Oct 01 16:32:27 ip-172-31-87-75.ec2.internal nagios[66694]: wproc: Registry request: name=Core Worker 66697;pid=66697
Oct 01 16:32:27 ip-172-31-87-75.ec2.internal nagios[66694]: wproc: Registry request: name=Core Worker 66698;pid=66698
Oct 01 16:32:27 ip-172-31-87-75.ec2.internal nagios[66694]: Successfully launched command file worker with pid 66739
[ec2-user@ip-172-31-87-75 nagios-plugins-2.4.11]$
```

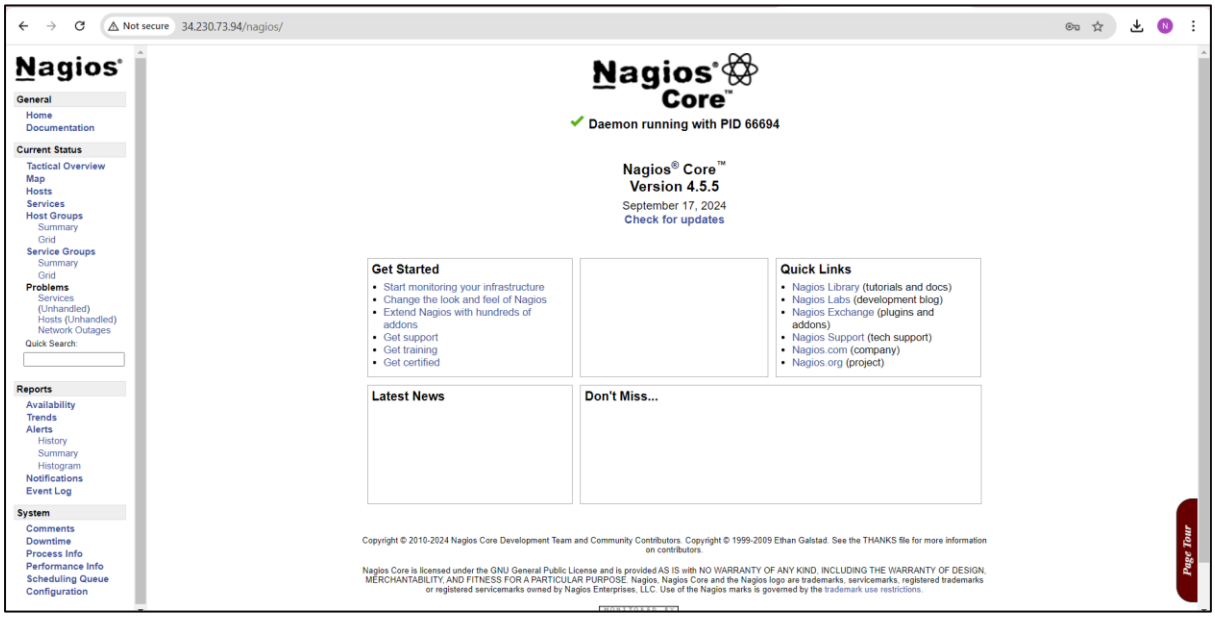
We can see that Nagios has been initialized correctly and its status is active.

30. Go back to your instances and copy the public IPv4 address



31. Lastly, go to your web browser and type “http://<public-IPv4-address>/nagios” Replace public-IPv4-address with the public ip address of your instance which you copied.
    You will get a prompt to enter the username and password that have been set for nagios admin in step 25.

You will see the below shown page after entering credentials.



Conclusion: In this experiment the main issue that I faced was accessing the nagios web interface due to a “Forbidden: You do not have permission to access this resource” error This issue was resolved by adjusting the inbound security rules and ensuring that all necessary files were installed in the correct directories. It is also important to restart apache after doing any changes for the changes to get reflected. Once Nagios was activated without errors, the output was successfully displayed on the web interface.

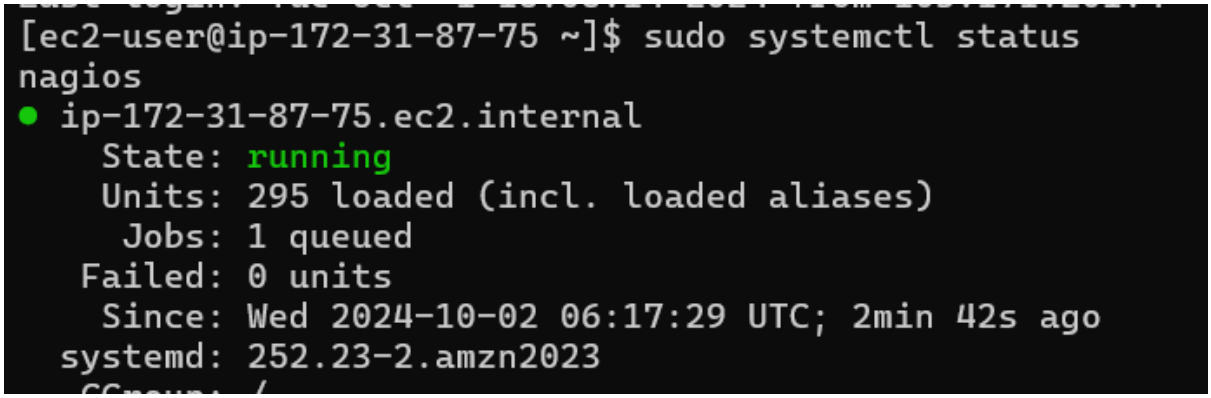
Advance DevOps

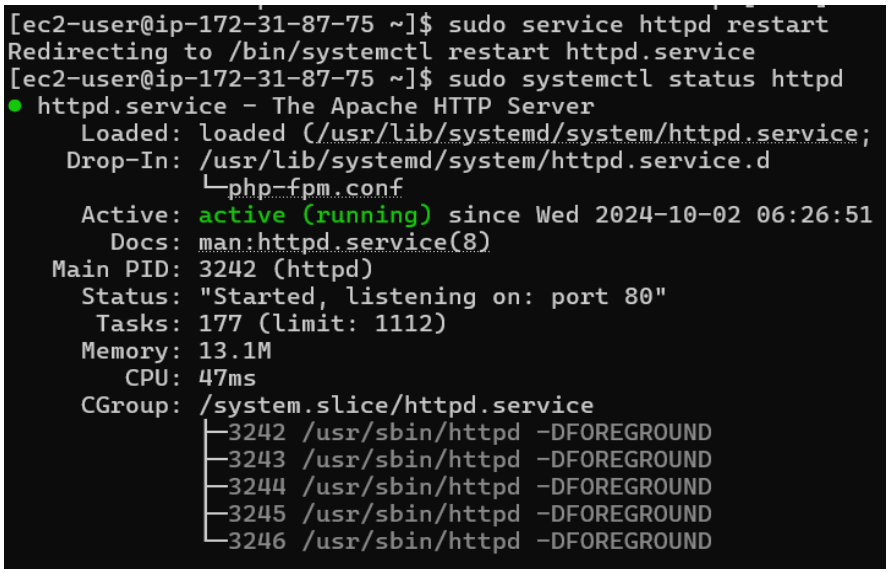
Experiment 10

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

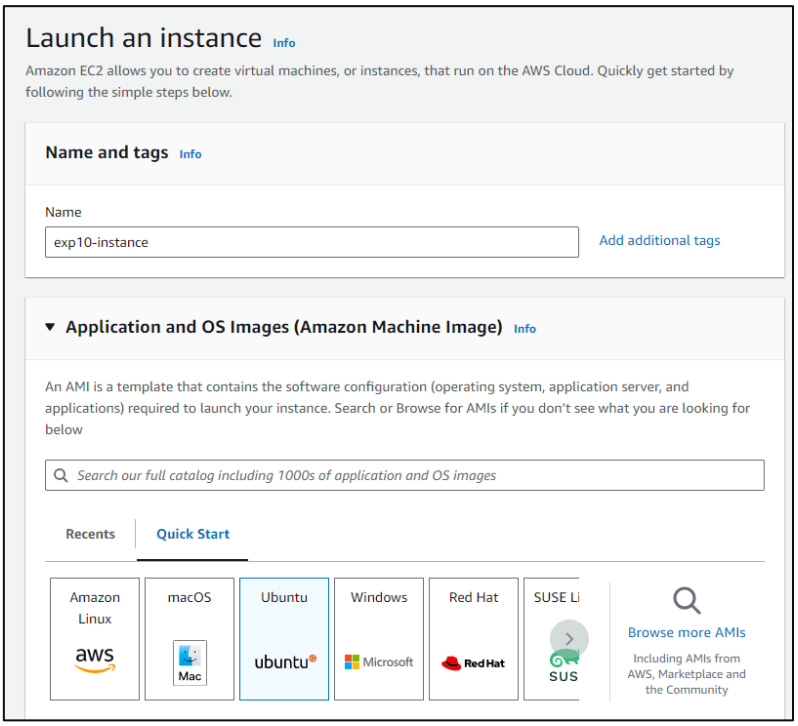
Steps:

1. Firstly, we will check whether nagios is running on the server side by using the command “sudo systemctl status nagios” on the host machine (host machine is the instance connected to the terminal in experiment 9, ensure that you have started the instance created for exp9, also check status of apache).

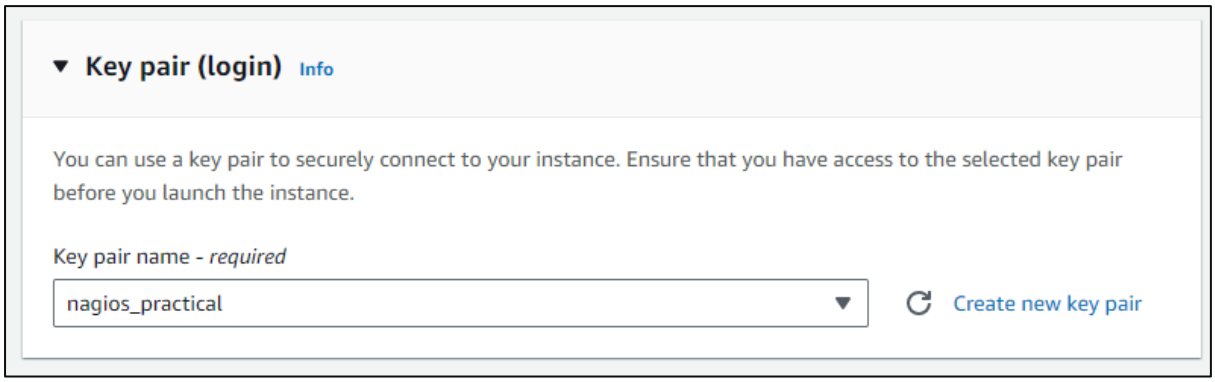




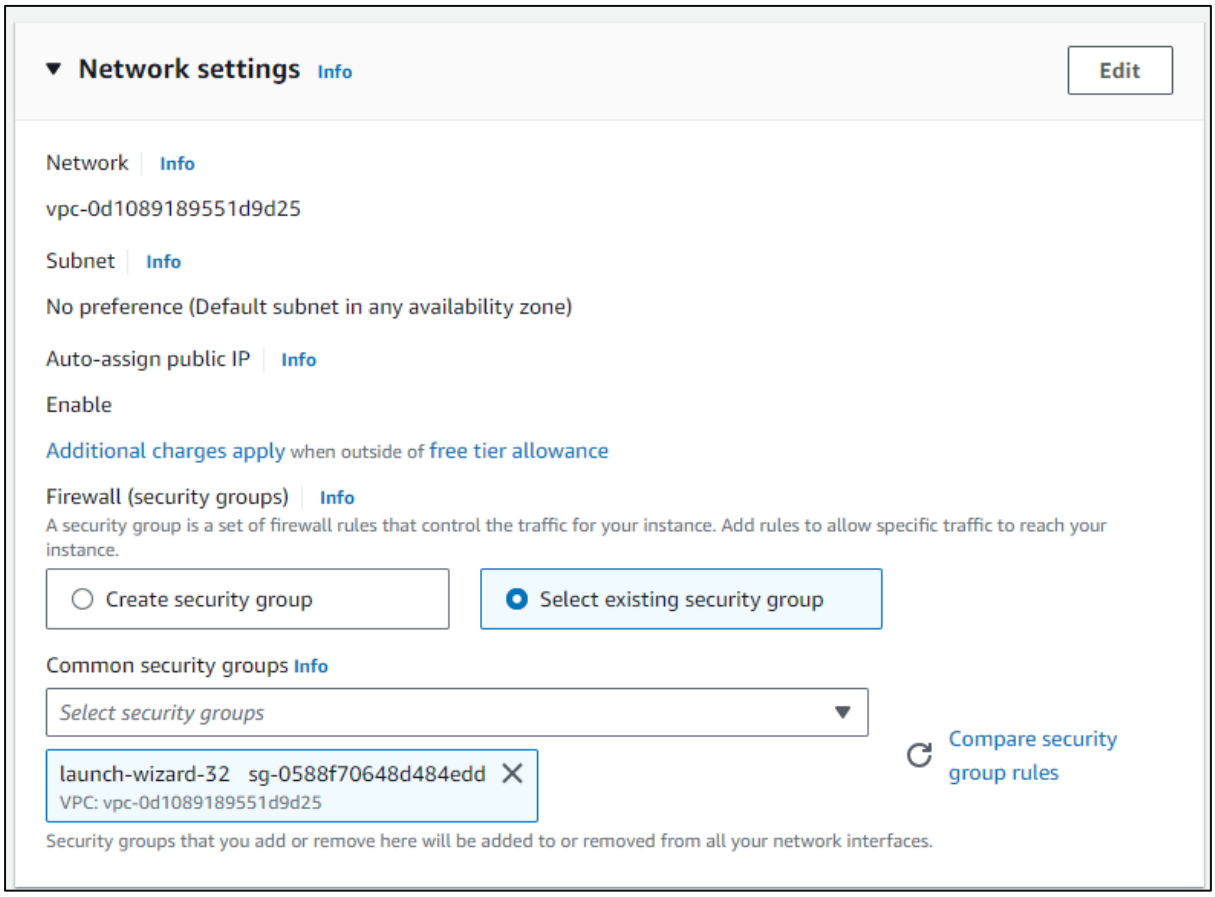
2. Now we will launch a new instance. Select ubuntu for the OS.



3. Select the key pair which was created and used in the exp 9.



4. Select existing security group and from the list of options select the security group created for exp 9. Previously it was launch wizard 32 and so here I have selected the same.



5. Open a new terminal to connect to the client machine. Copy the SSH command provided in the SSH client section during connection of instance. When pasting the command into your terminal, ensure you specify the full path to your .pem file instead of just the file name.

```
PS C:\Users\Dell> ssh -i "C:\Users\Dell\Downloads\nagios_practical.pem" ubuntu@ec2-18-207-191-20.compute-1.amazonaws.com

The authenticity of host 'ec2-18-207-191-20.compute-1.amazonaws.com (18.207.191.20)' can't be established.
ED25519 key fingerprint is SHA256:NPJPUfsGZXQUXK8GQ9sw/fIzAFXOnabRJiCiAFdyYiU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-207-191-20.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1016-aws x86_64)

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

 System information as of Wed Oct  2 06:36:10 UTC 2024

  System load:  0.15              Processes:             106
  Usage of /:   22.9% of 6.71GB   Users logged in:       0
  Memory usage: 21%               IPv4 address for enX0: 172.31.40.130
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

6. Now go back to your host machine and run the following command

   ps -ef | grep nagios

```
[ec2-user@ip-172-31-87-75 ~]$ ps -ef | grep nagios
nagios      2002       1  0 06:17 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios      2003    2002  0 06:17 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios      2004    2002  0 06:17 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios      2005    2002  0 06:17 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios      2006    2002  0 06:17 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios      2007    2002  0 06:17 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
ec2-user    4389    2306  0 06:43 pts/0    00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-87-75 ~]$
```

7. Now perform these commands on the host terminal
   sudo su
   mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts

```
[root@ip-172-31-87-75 ec2-user]# mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-87-75 ec2-user]#
```

   cp /usr/local/nagios/etc/objects/localhost.cfg
   /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg

```
[root@ip-172-31-87-75 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
[root@ip-172-31-87-75 ec2-user]#
```

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

   The above given command will open the nano text editor wherein you have to do the following changes:

i. Change the hostgroup name to linux-servers1

```
##############################################################################
#
# 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                 linuxserver               ; Comma separated list of hosts that belong to this group
}
```

ii. Change host name and alias from localhost to linuxserver everywhere in the file

```
# Define a service to "ping" the local machine

define service {

    use                     local-service           ; Name of service template to use
    host_name               linuxserver
    service_description     PING
    check_command           check_ping!100.0,20%!500.0,60%
}
```

iii. Change the address to the public IPv4 address of the ubuntu instance (You will find the ip address when you select the instance on the ec2 instances dashboard)

```
# 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                   linuxserver
    address                 18.207.191.20
}
```

8. Open the Nagios Config file by using this command:
   nano /usr/local/nagios/etc/nagios.cfg
   nano text editor will get opened

```
##############################################################################
#
# 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

# Definitions for monitoring the local (Linux) host
cfg_file=/usr/local/nagios/etc/objects/localhost.cfg

# Definitions for monitoring a Windows machine
```

9. In the text editor add “cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/” this line

```
# You can also tell Nagios to process all config files (with a .cfg
# extension) in a particular directory by using the cfg_dir
# directive as shown below:

#cfg_dir=/usr/local/nagios/etc/servers
#cfg_dir=/usr/local/nagios/etc/printers
#cfg_dir=/usr/local/nagios/etc/switches
#cfg_dir=/usr/local/nagios/etc/routers
cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/
```

10. Now we will verify the configuration files

    /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 16 services.
        Checked 2 hosts.
        Checked 2 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 2 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

Things look okay - No serious problems were detected during the pre-flight check
[root@ip-172-31-87-75 ec2-user]#
```

If there are no errors we can proceed further

11. We will now restart the nagios service
    service nagios restart

```
[root@ip-172-31-87-75 ec2-user]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
[root@ip-172-31-87-75 ec2-user]#
```

12. Now on the client machine (The ubuntu machine we created for this experiment) run the following command:

    sudo apt update -y

```
ubuntu@ip-172-31-40-130:~$ sudo apt update -y
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://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [535 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [130 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [8676 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [380 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [156 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [45.0 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [14.9 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [14.4 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [3608 B]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [212 B]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [532 B]
```

    sudo apt install gcc -y

```
ubuntu@ip-172-31-40-130:~$ sudo apt install gcc -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu cpp cpp-13 cpp-13-x86-64-linux-gnu cpp-x86-64-linux-gnu fontconfig-config fonts-dejavu-core
  fonts-dejavu-mono gcc-13 gcc-13-base gcc-13-x86-64-linux-gnu gcc-x86-64-linux-gnu libaom3 libasan8 libatomic1 libbinutils libc-dev-bin libc-devtools
  libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libde265-0 libdeflate0 libfontconfig1 libgcc-13-dev libgd3 libgomp1 libgprofng0
  libheif-plugin-aomdec libheif-plugin-aomenc libheif-plugin-libde265 libheif1 libhwasan0 libisl23 libitm1 libjbig0 libjpeg-turbo8 libjpeg8 liblerc4
  liblsan0 libmpc3 libquadmath0 libsframe1 libsharpyuv0 libtiff6 libtsan2 libubsan1 libwebp7 libxpm4 linux-libc-dev manpages-dev rpcsvc-proto
Suggested packages:
  binutils-doc gprofng-gui cpp-doc gcc-13-locales cpp-13-doc gcc-multilib make autoconf automake libtool flex bison gdb gcc-doc gcc-13-multilib gcc-13-doc
  gdb-x86-64-linux-gnu glibc-doc libgd-tools libheif-plugin-x265 libheif-plugin-ffmpegdec libheif-plugin-jpegdec libheif-plugin-jpegenc
  libheif-plugin-j2kdec libheif-plugin-j2kenc libheif-plugin-rav1e libheif-plugin-svtenc
The following NEW packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu cpp cpp-13 cpp-13-x86-64-linux-gnu cpp-x86-64-linux-gnu fontconfig-config fonts-dejavu-core
  fonts-dejavu-mono gcc gcc-13 gcc-13-base gcc-13-x86-64-linux-gnu gcc-x86-64-linux-gnu libaom3 libasan8 libatomic1 libbinutils libc-dev-bin libc-devtools
  libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libde265-0 libdeflate0 libfontconfig1 libgcc-13-dev libgd3 libgomp1 libgprofng0
  libheif-plugin-aomdec libheif-plugin-aomenc libheif-plugin-libde265 libheif1 libhwasan0 libisl23 libitm1 libjbig0 libjpeg-turbo8 libjpeg8 liblerc4
  liblsan0 libmpc3 libquadmath0 libsframe1 libsharpyuv0 libtiff6 libtsan2 libubsan1 libwebp7 libxpm4 linux-libc-dev manpages-dev rpcsvc-proto
0 upgraded, 57 newly installed, 0 to remove and 6 not upgraded.
Need to get 62.8 MB of archives.
After this operation, 222 MB of additional disk space will be used.
```

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

```
ubuntu@ip-172-31-40-130:~$ sudo apt install -y nagios-nrpe-server nagios-plugins
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'monitoring-plugins' instead of 'nagios-plugins'
The following additional packages will be installed:
  libavahi-client3 libavahi-common-data libavahi-common3 libcups2t64 libdbi1t64 libldb2 libmysqlclient21 libnet-snmp-perl libpq5 libradcli4 libsmbclient0
  libsnmp-base libsnmp40t64 libtalloc2 libtdb1 libtevent0t64 liburiparser1 libwbclient0 monitoring-plugins-basic monitoring-plugins-common
  monitoring-plugins-standard mysql-common python3-gpg python3-ldb python3-markdown python3-samba python3-talloc python3-tdb rpcbind samba-common
  samba-common-bin samba-dsdb-modules samba-libs smbclient snmp
Suggested packages:
  cups-common libcrypt-des-perl libdigest-hmac-perl libio-socket-inet6-perl snmp-mibs-downloader icinga2 nagios-plugins-contrib fping postfix
  | sendmail-bin | exim4-daemon-heavy | exim4-daemon-light qstat xinetd | inetd python-markdown-doc heimdal-clients python3-dnspython cifs-utils
The following NEW packages will be installed:
  libavahi-client3 libavahi-common-data libavahi-common3 libcups2t64 libdbi1t64 libldb2 libmysqlclient21 libnet-snmp-perl libpq5 libradcli4 libsmbclient0
  libsnmp-base libsnmp40t64 libtalloc2 libtdb1 libtevent0t64 liburiparser1 libwbclient0 monitoring-plugins monitoring-plugins-basic
  monitoring-plugins-common monitoring-plugins-standard mysql-common nagios-nrpe-server python3-gpg python3-ldb python3-markdown python3-samba
  python3-talloc python3-tdb rpcbind samba-common samba-common-bin samba-dsdb-modules samba-libs smbclient snmp
0 upgraded, 37 newly installed, 0 to remove and 6 not upgraded.
Need to get 16.1 MB of archives.
After this operation, 72.0 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 nagios-nrpe-server amd64 4.1.0-1ubuntu3 [356 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 rpcbind amd64 1.2.6-7ubuntu2 [46.5 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libavahi-common-data amd64 0.8-13ubuntu6 [29.7 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libavahi-common3 amd64 0.8-13ubuntu6 [23.3 kB]
```

13. Open nrpe.cfg file to make changes.
    sudo nano /etc/nagios/nrpe.cfg
    Under allowed_hosts, add your nagios host public IPv4 address:

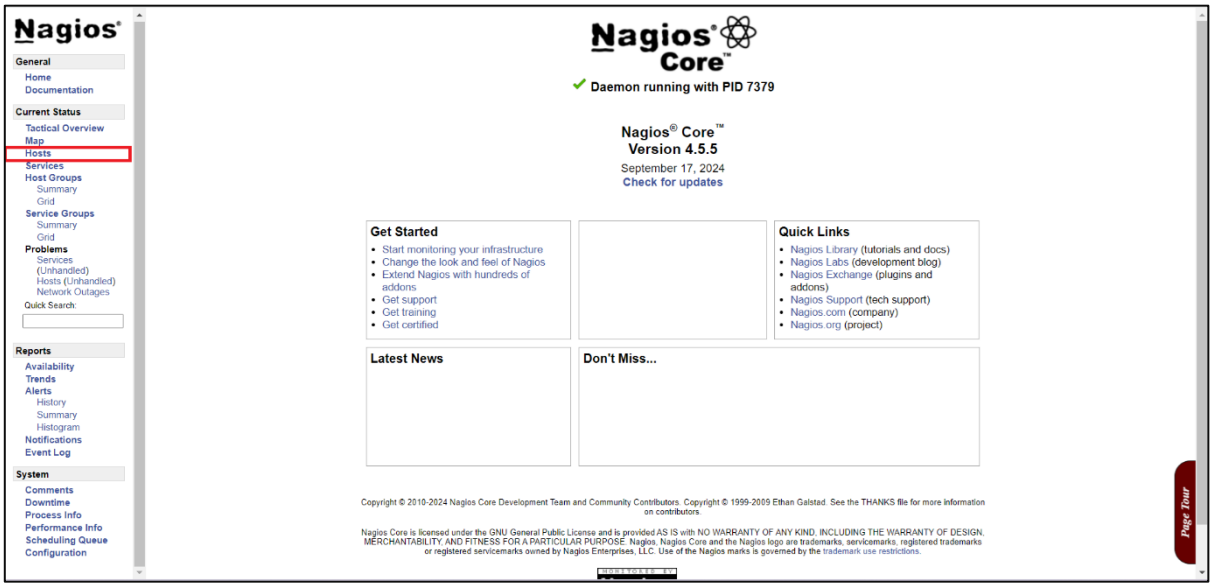
```
#
# Note: The daemon only does rudimentary checking of the client's IP
# address.  I would highly recommend adding entries in your /etc/hosts.allow
# file to allow only the specified host to connect to the port
# you are running this daemon on.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd

allowed_hosts=127.0.0.1,54.163.184.143


# 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! ***
# Read the SECURITY file for information on some of the security implications
# of enabling this variable.
#
# Values: 0=do not allow arguments, 1=allow command arguments
```

14. Now restart the NRPE server

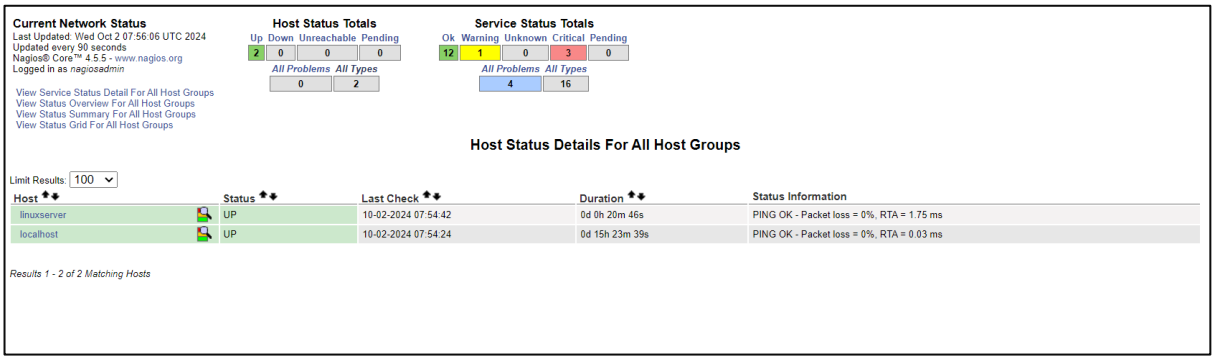
    sudo systemctl restart nagios-nrpe-server

```
ubuntu@ip-172-31-40-130:~$ sudo systemctl restart nagios-nrpe-server
ubuntu@ip-172-31-40-130:~$
```

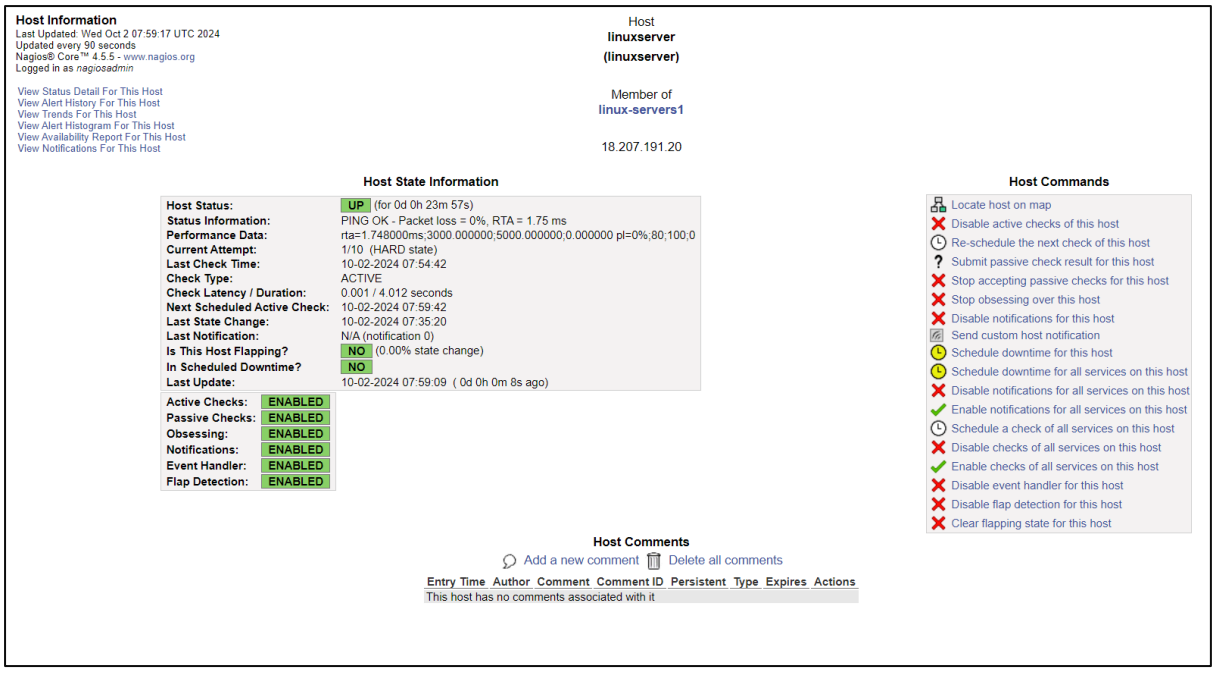
15. Go to the nagios dashboard and click on hosts



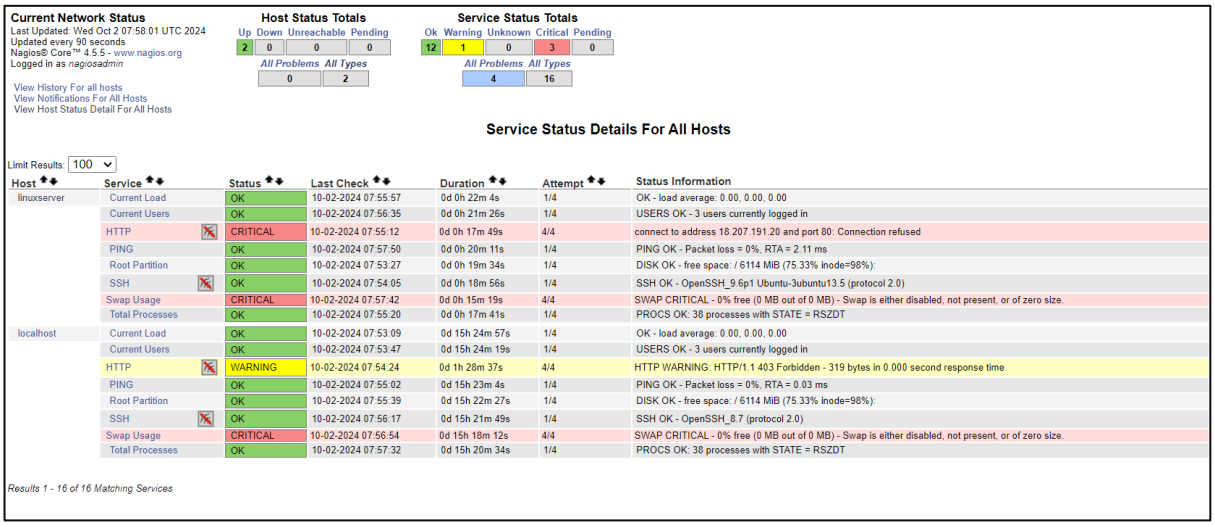
Click on linux server



We can see the host state information:



If you want to see all the services and ports being monitored then select the services option and you will see the page as shown below:



Current Network Status
Last Updated: Wed Oct 2 07:58:01 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.5.5 - www.nagios.org
Logged in as *nagiosadmin*

View History For all hosts
View Notifications For All Hosts
View Host Status Detail For All Hosts

Host Status Totals

Up	Down	Unreachable	Pending
2	0	0	0

All Problems	All Types
0	2

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
12	1	0	3	0

All Problems	All Types
4	16

Service Status Details For All Hosts

Limit Results: 100

Host	Service	Status	Last Check	Duration	Attempt	Status Information
linuxserver	Current Load	OK	10-02-2024 07:55:57	0d 0h 22m 4s	1/4	OK - load average: 0.00, 0.00, 0.00
	Current Users	OK	10-02-2024 07:56:35	0d 0h 21m 26s	1/4	USERS OK - 3 users currently logged in
	HTTP	CRITICAL	10-02-2024 07:55:12	0d 0h 17m 49s	4/4	connect to address 18.207.191.20 and port 80: Connection refused
	PING	OK	10-02-2024 07:57:50	0d 0h 20m 11s	1/4	PING OK - Packet loss = 0%, RTA = 2.11 ms
	Root Partition	OK	10-02-2024 07:53:27	0d 0h 19m 34s	1/4	DISK OK - free space: / 6114 MiB (75.33% inode=98%):
	SSH	OK	10-02-2024 07:54:05	0d 0h 18m 56s	1/4	SSH OK - OpenSSH_9.6p1 Ubuntu-3ubuntu13.5 (protocol 2.0)
	Swap Usage	CRITICAL	10-02-2024 07:57:42	0d 0h 15m 19s	4/4	SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size.
	Total Processes	OK	10-02-2024 07:55:20	0d 0h 17m 41s	1/4	PROCS OK: 38 processes with STATE = RSZDT
localhost	Current Load	OK	10-02-2024 07:53:09	0d 15h 24m 57s	1/4	OK - load average: 0.00, 0.00, 0.00
	Current Users	OK	10-02-2024 07:53:47	0d 15h 24m 19s	1/4	USERS OK - 3 users currently logged in
	HTTP	WARNING	10-02-2024 07:54:24	0d 1h 28m 37s	4/4	HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.000 second response time
	PING	OK	10-02-2024 07:55:02	0d 15h 23m 4s	1/4	PING OK - Packet loss = 0%, RTA = 0.03 ms
	Root Partition	OK	10-02-2024 07:55:39	0d 15h 22m 27s	1/4	DISK OK - free space: / 6114 MiB (75.33% inode=98%):
	SSH	OK	10-02-2024 07:56:17	0d 15h 21m 49s	1/4	SSH OK - OpenSSH_8.7 (protocol 2.0)
	Swap Usage	CRITICAL	10-02-2024 07:56:54	0d 15h 18m 12s	4/4	SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size.
	Total Processes	OK	10-02-2024 07:57:32	0d 15h 20m 34s	1/4	PROCS OK: 38 processes with STATE = RSZDT

Results 1 - 16 of 16 Matching Services

Conclusion: For performing this experiment it is necessary to start the instance of the previous experiment as that will act as the host and the instance created in this experiment will be the client machine. There were errors when I tried to run the command to verify the Nagios configuration file and in order to resolve those errors I reinstalled the nagios plugins and restarted the nagios service after which the errors were fixed.

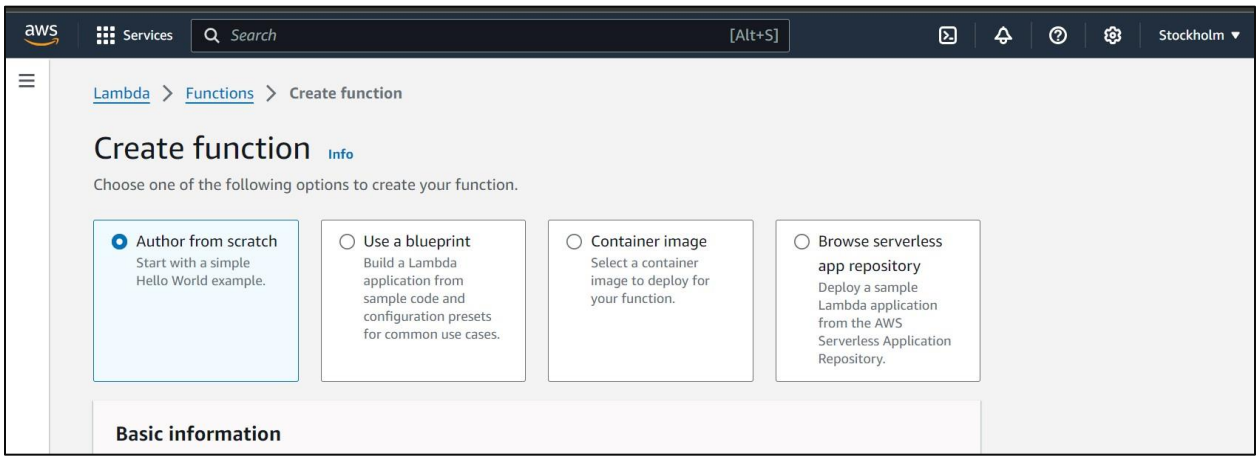
Advance DevOps

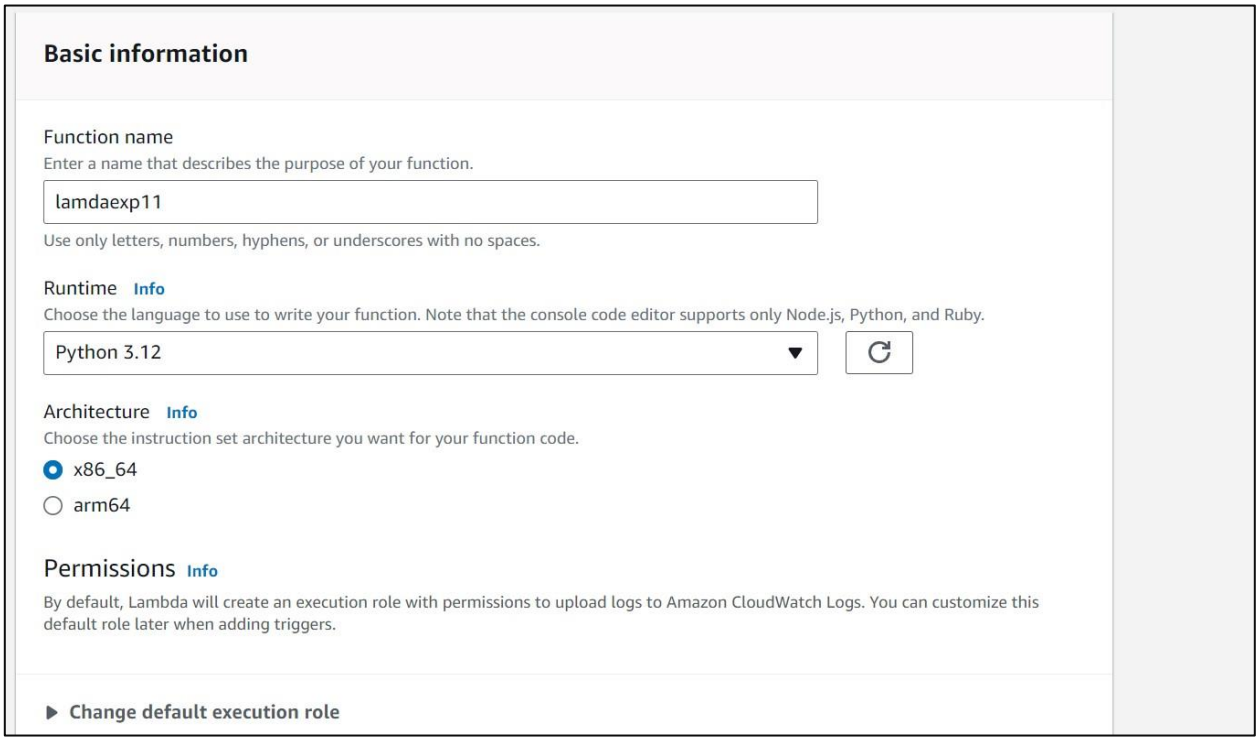
Experiment 11

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

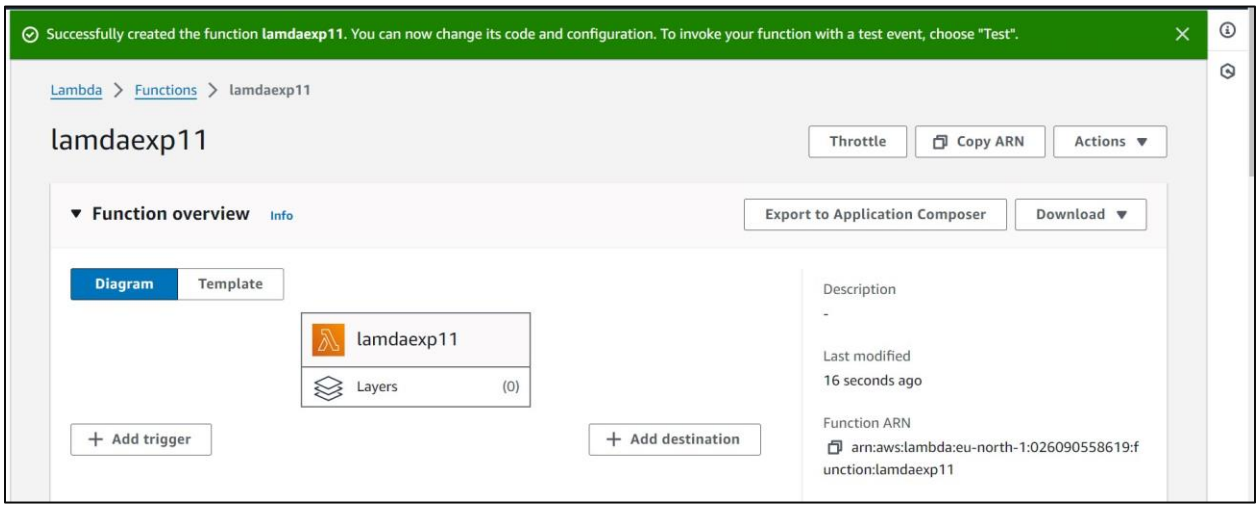
Steps:

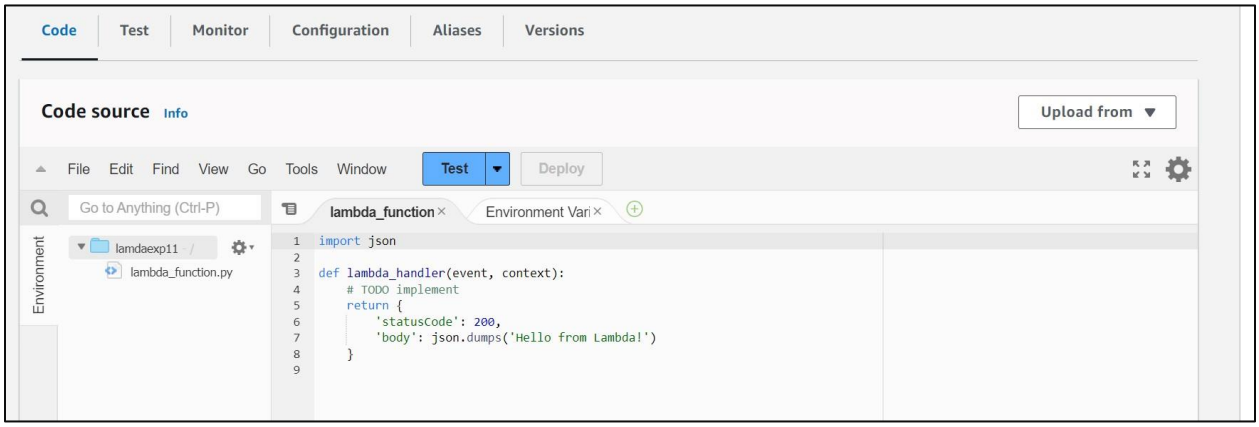
1. Go on your AWS console account and search for lambda and then go on create function Select the author from scratch, add function name and then, choose a runtime env for your function, under the dropdown, you can see all the options AWS supports, Python, Nodejs, .NET and Java being the most popular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



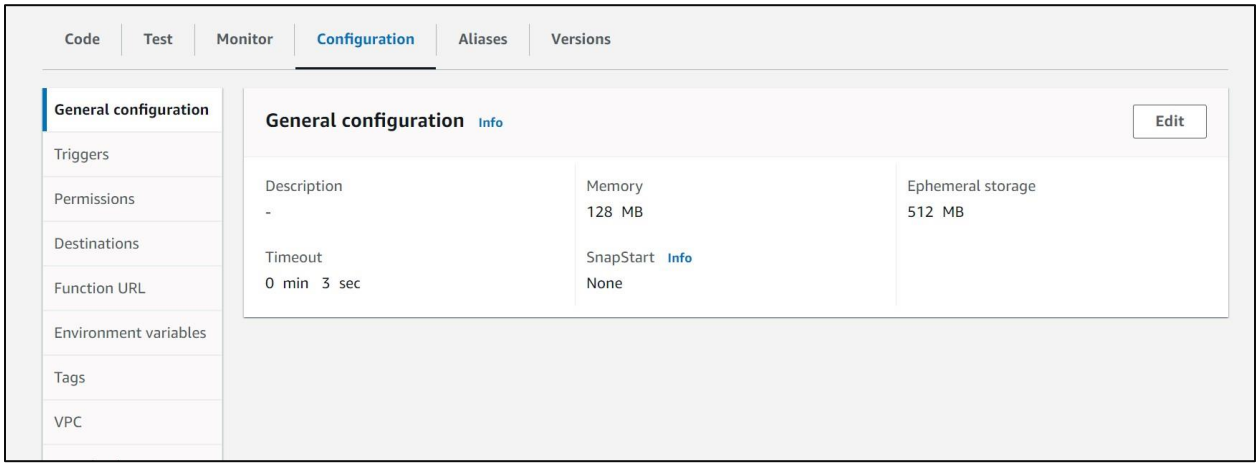


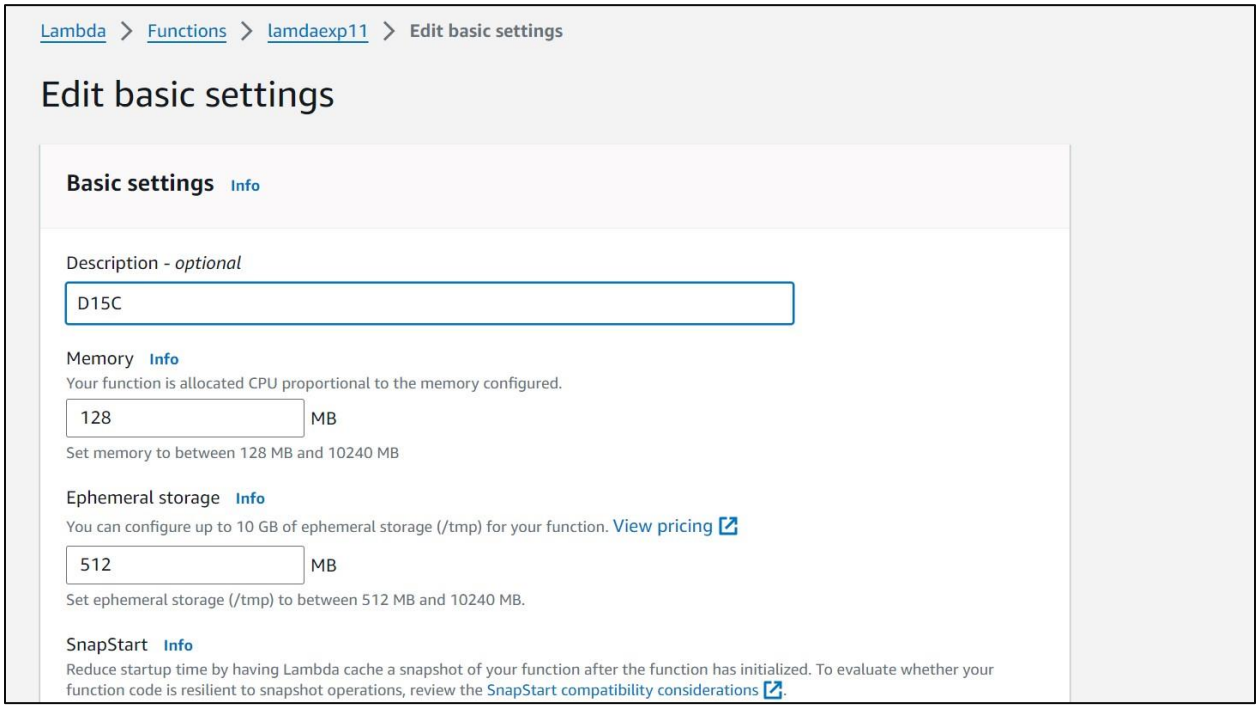
2. After the function is created successfully go on code, write the default code and then configure it.



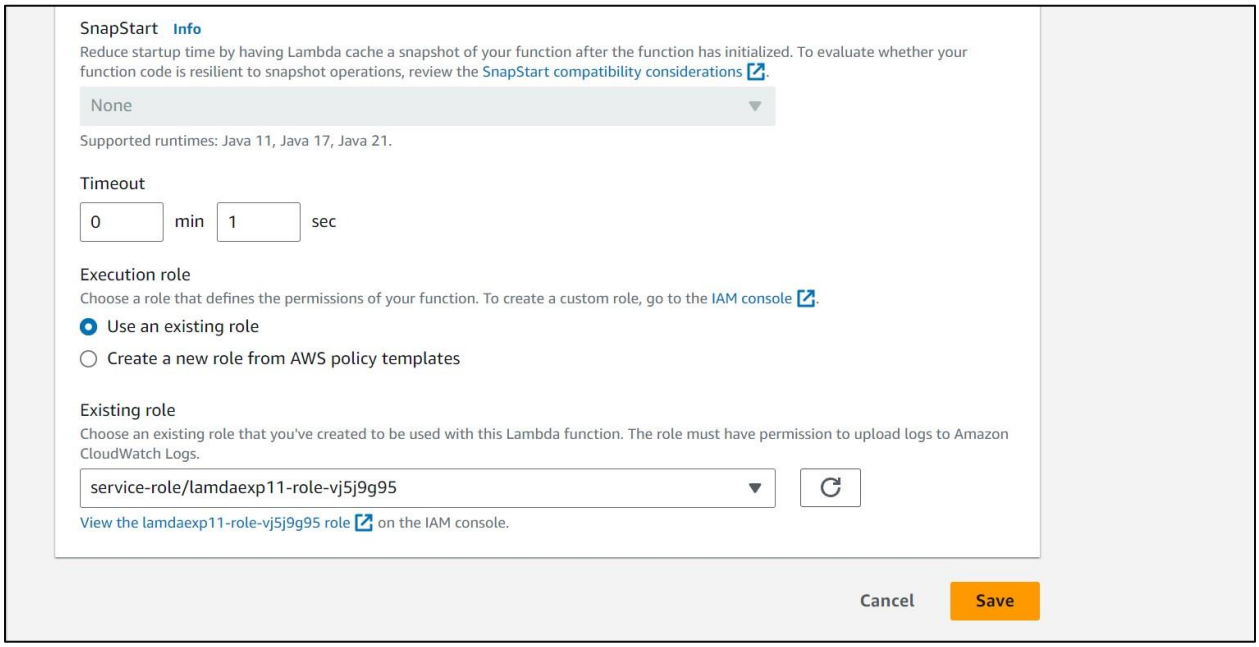


3. Then go on edit basic settings, add the description and then save it.

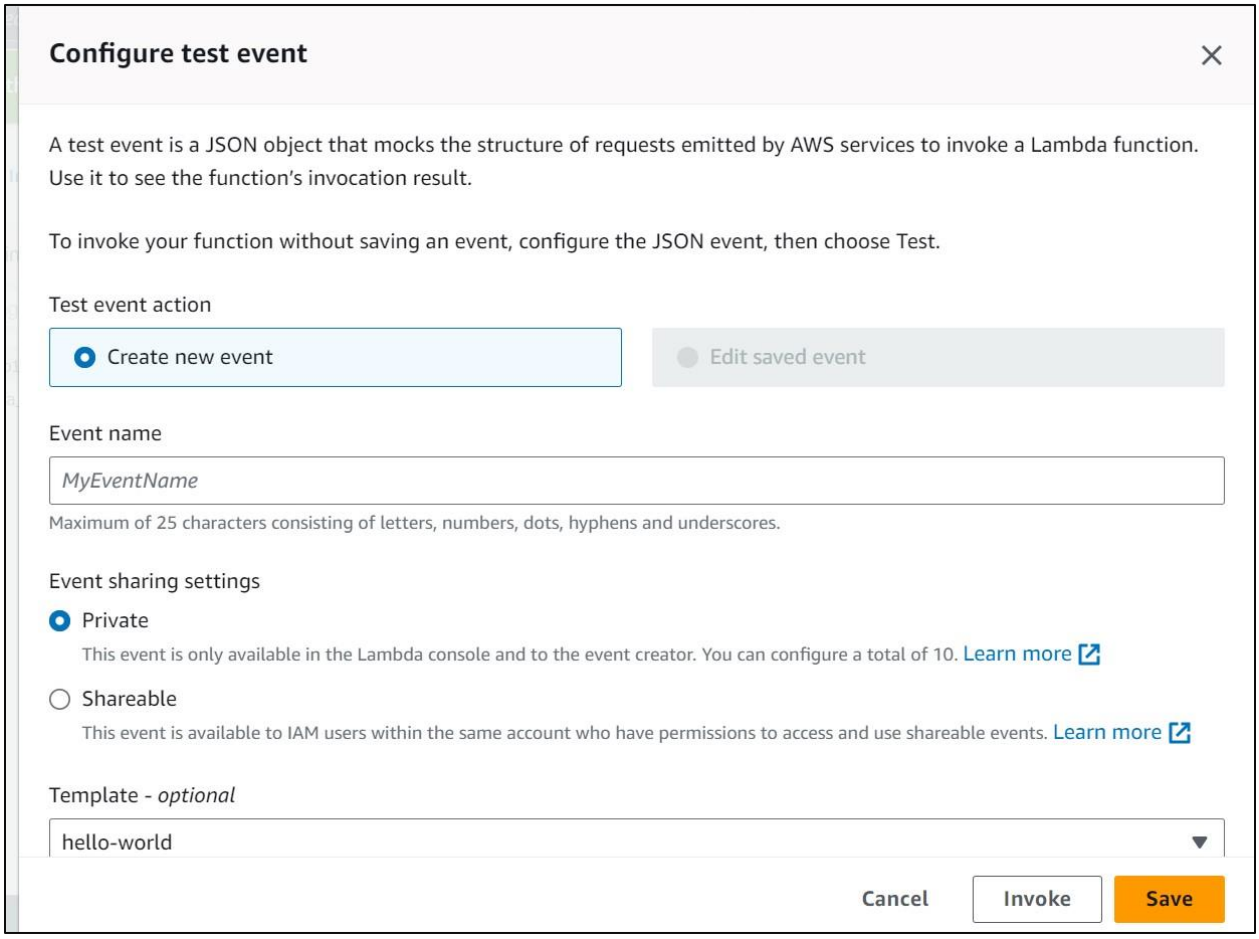




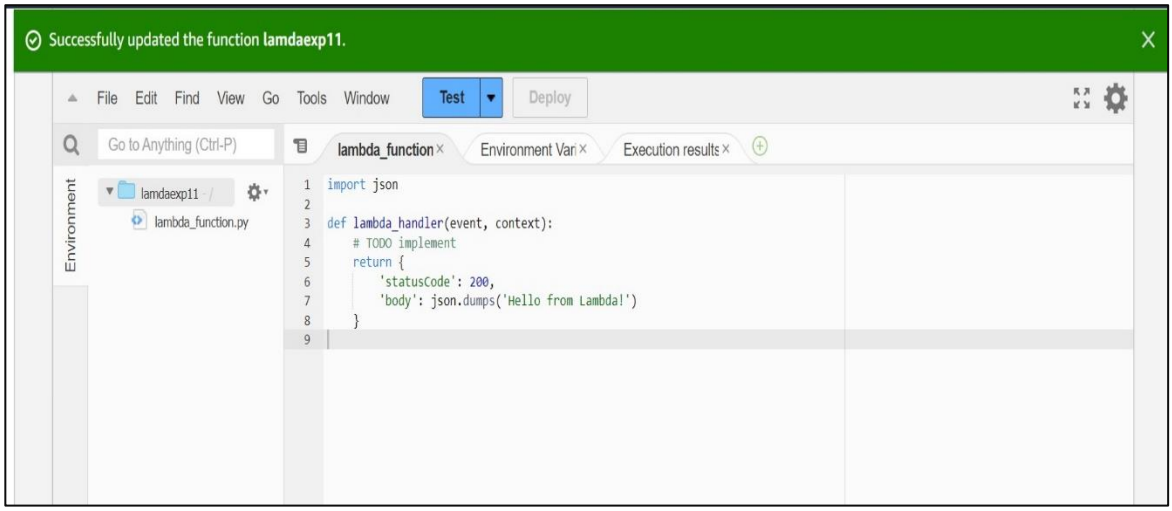
4. Click on “use an existing role“ option and then add the role and save it.



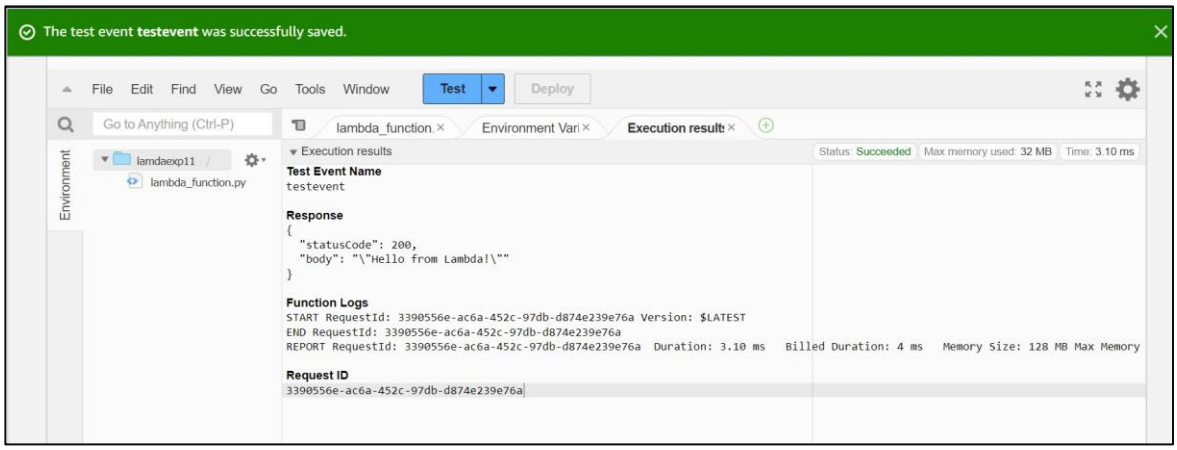
5. Go on configure test event click on “create new event”. Add an event name, set the event sharing settings and select hello-world template option. After this select save.



6. Once the test event is successfully created and saved click on test option to test the code.



7. The function has been added successfully.



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.

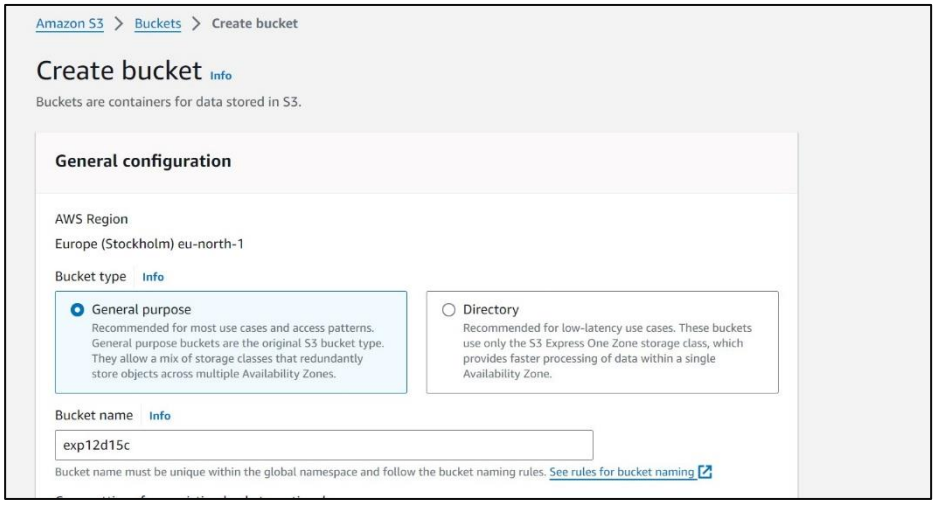
Advance DevOps

Experiment 12

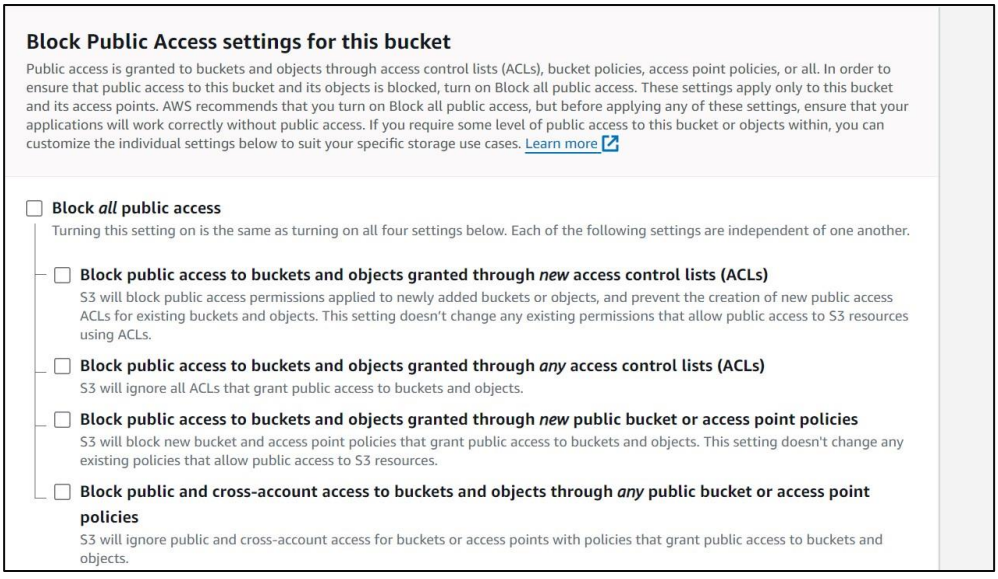
Aim: To create a Lambda function which will log “An Image has been added” once you add an object to a specific bucket in S3.

Steps:

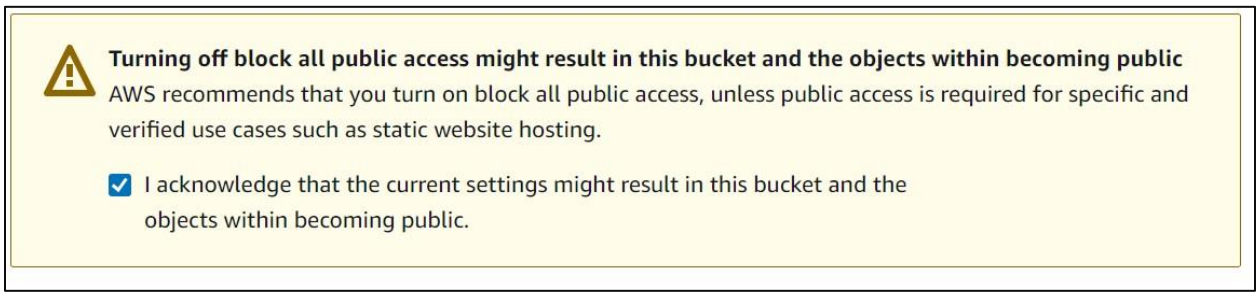
1. First, we have to create a S3 bucket for which give a name to the bucket.



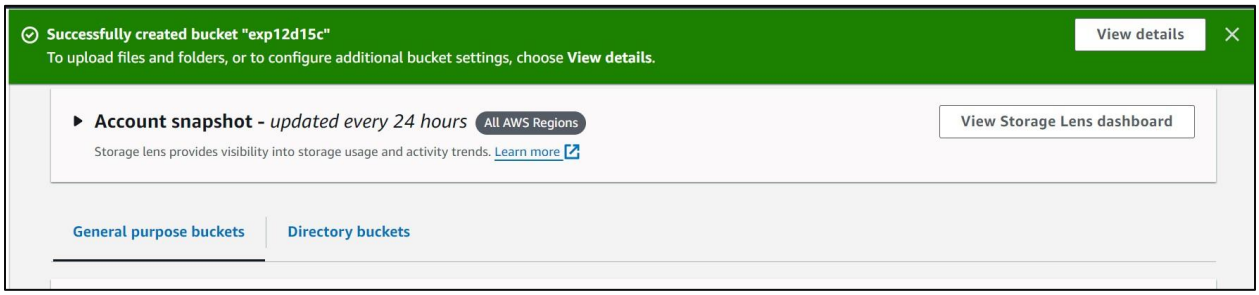
2. To allow public access to the bucket, uncheck the block all public access option.



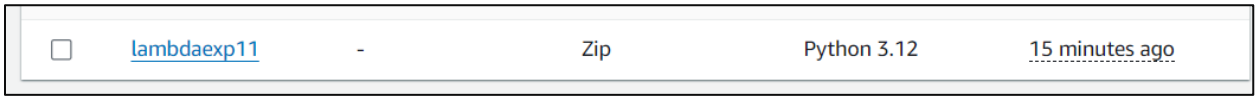
3. Give confirmation that you want to allow full public access and create the bucket



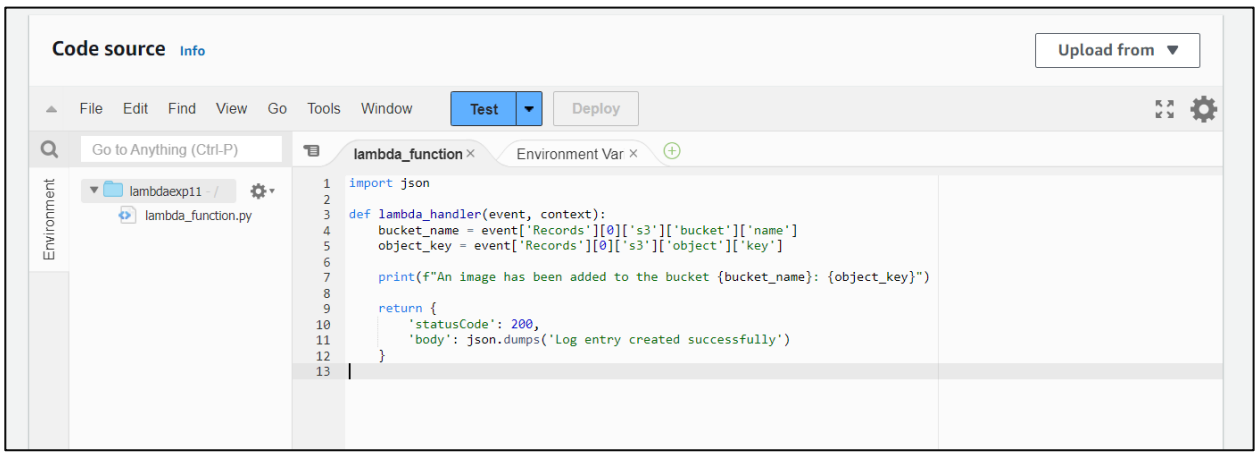
4. You will see the confirmation that the bucket is created successfully



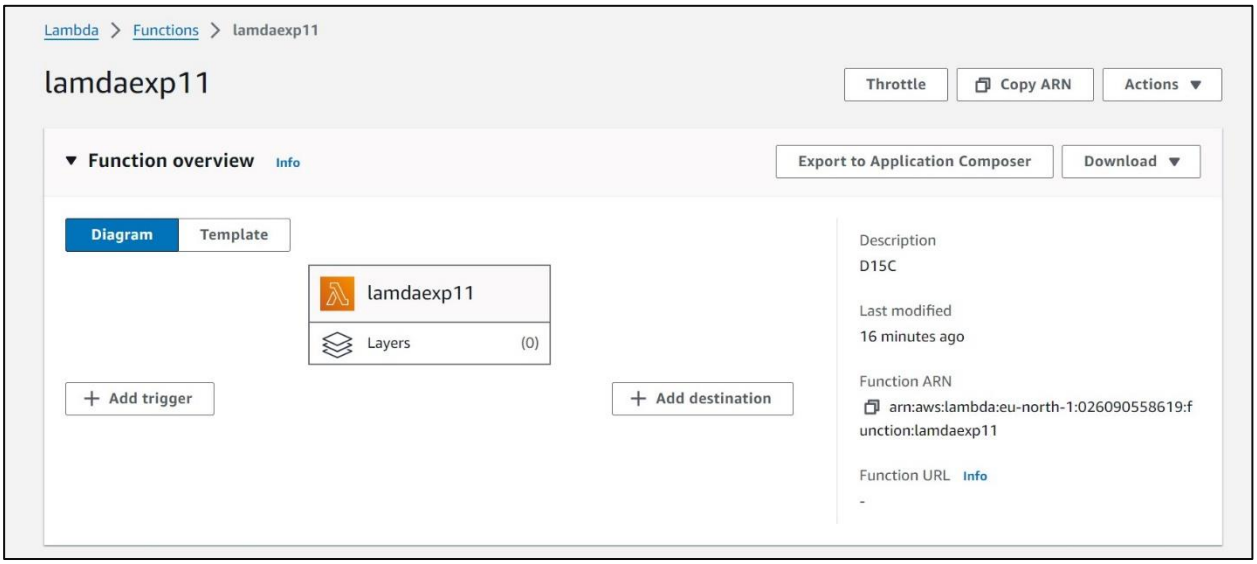
5. After creating the bucket go to the AWS dashboard and search for lambda function service, select the function we created in experiment 11 (lambdaexp11). We are going to add this bucket as a trigger to this function.



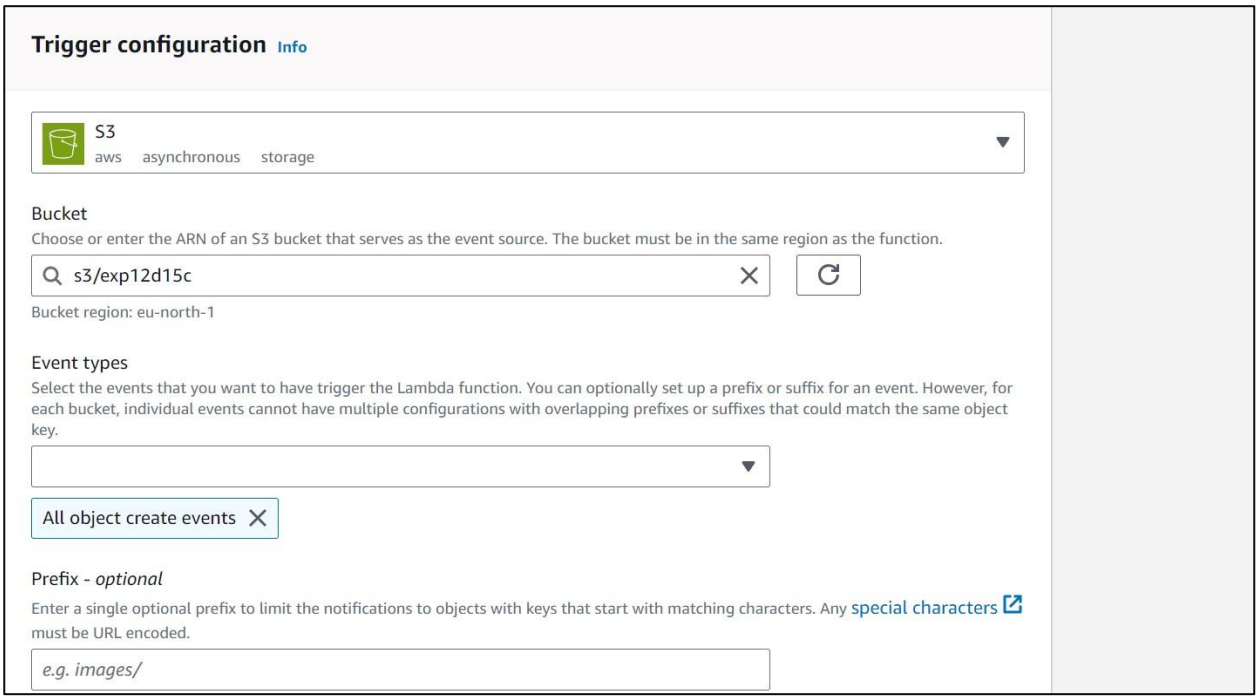
6. After selecting the function scroll down to the code section and add the following lines of code to display the required message when image is added to the bucket.



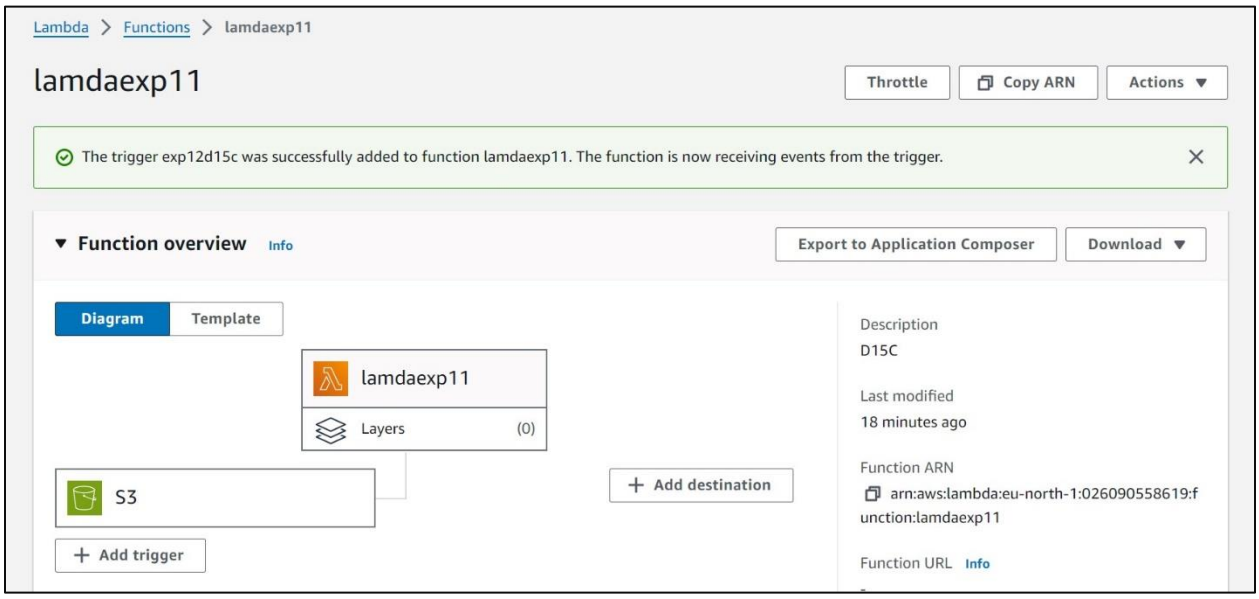
7. Now on the function overview section of the dashboard you can see the “Add trigger” button. Click on that.



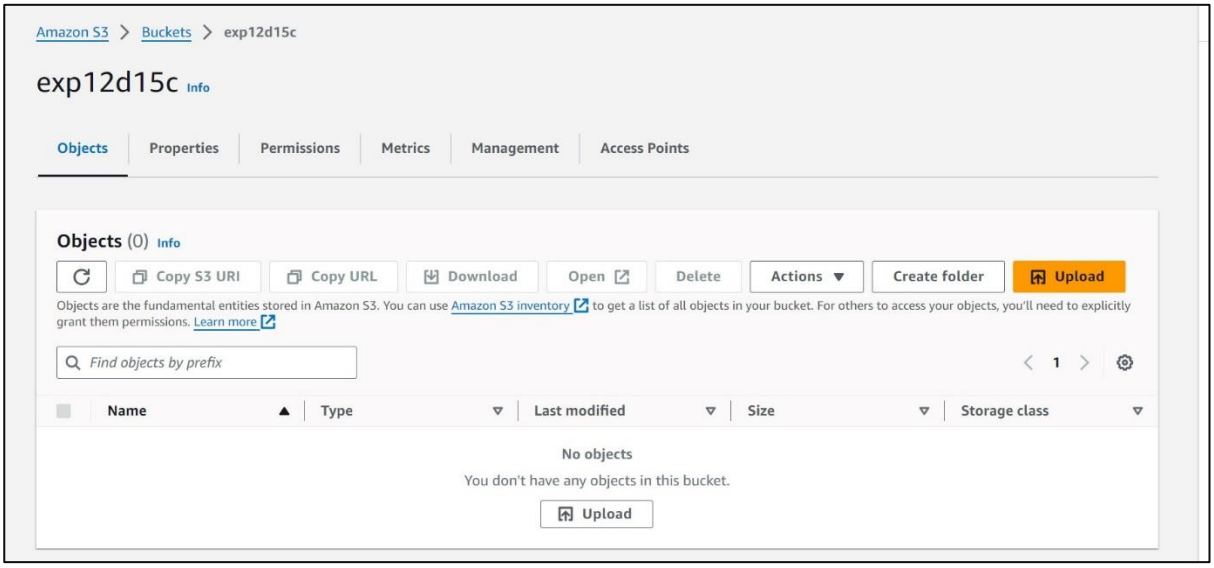
8. It will lead you to the trigger configuration tab. Here select the service as S3 and choose the bucket you just created.

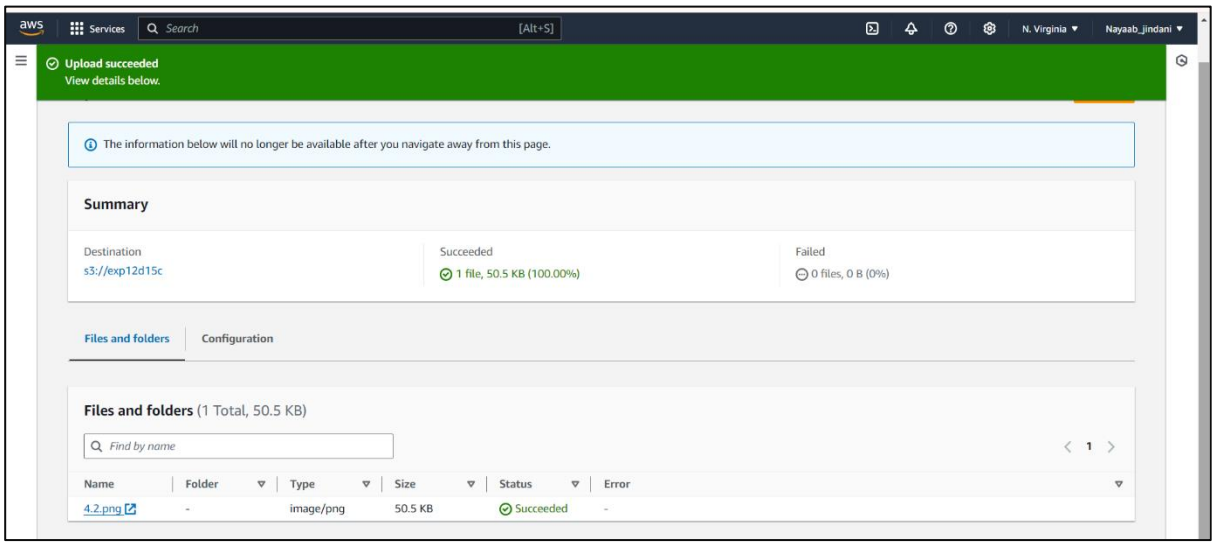


9. Here you can see we have the confirmation message as well as the S3 bucket has been added to our triggers.

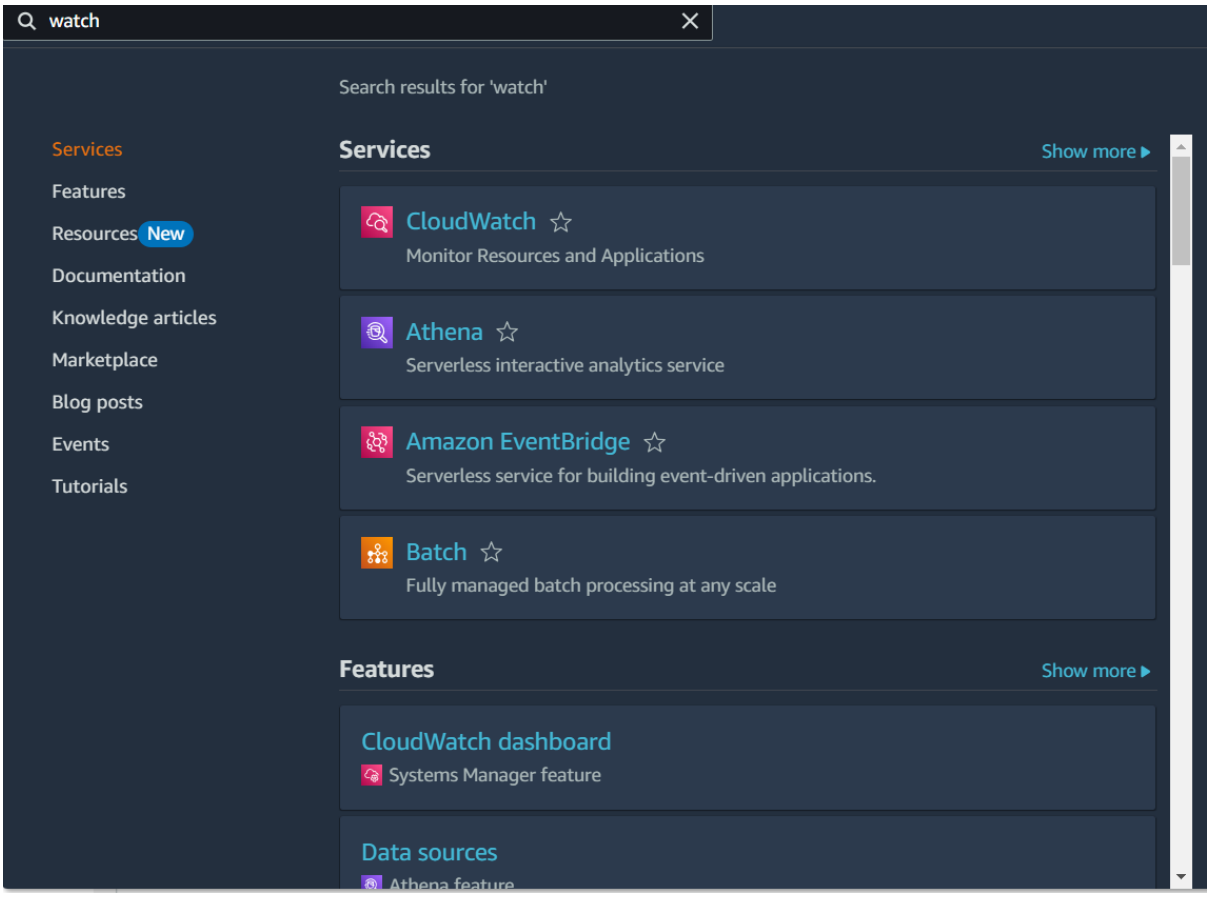


10. Now we will upload a file in the S3 bucket to test. Here I have uploaded an image file.

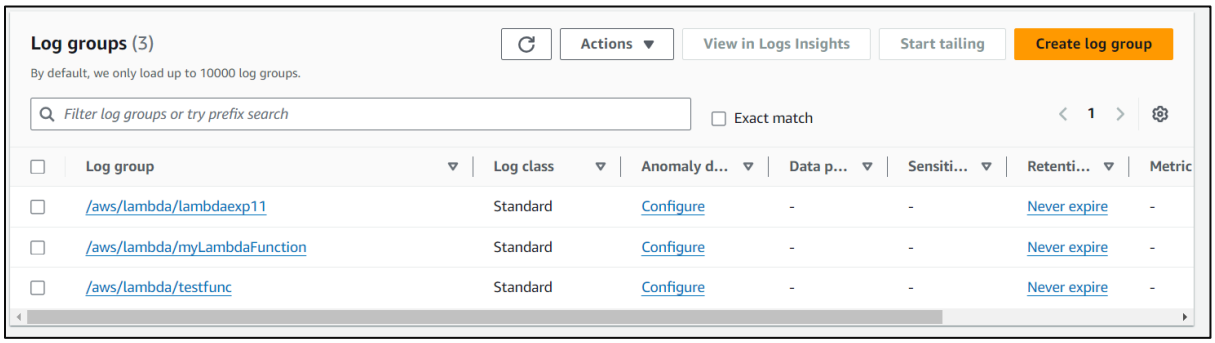




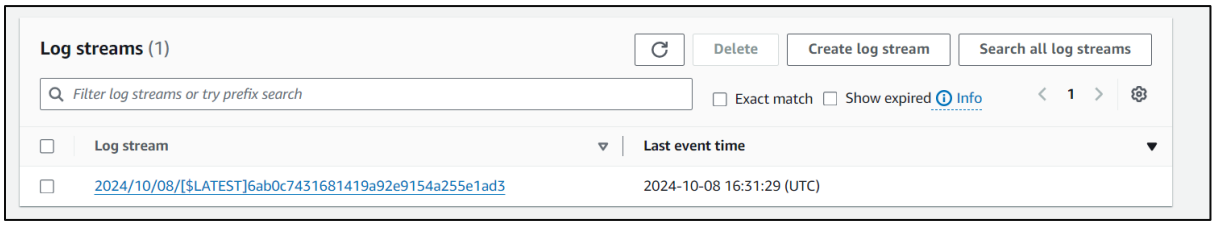
11. To check the log search for CloudWatch in services



12. Select the function created in log groups



13. Select the log stream



14. Here we can see that the message has been displayed in the logs.

Conclusion: In this experiment, we successfully created an S3 bucket and configured it as a trigger for the Lambda function created in the previous experiment. Upon the occurrence of an event, such as uploading a file to the S3 bucket, the Lambda function was triggered, producing the output message: "An image has been added to the bucket." We have successfully demonstrated the effective integration of S3 and Lambda, allowing automatic responses to file uploads in the bucket.