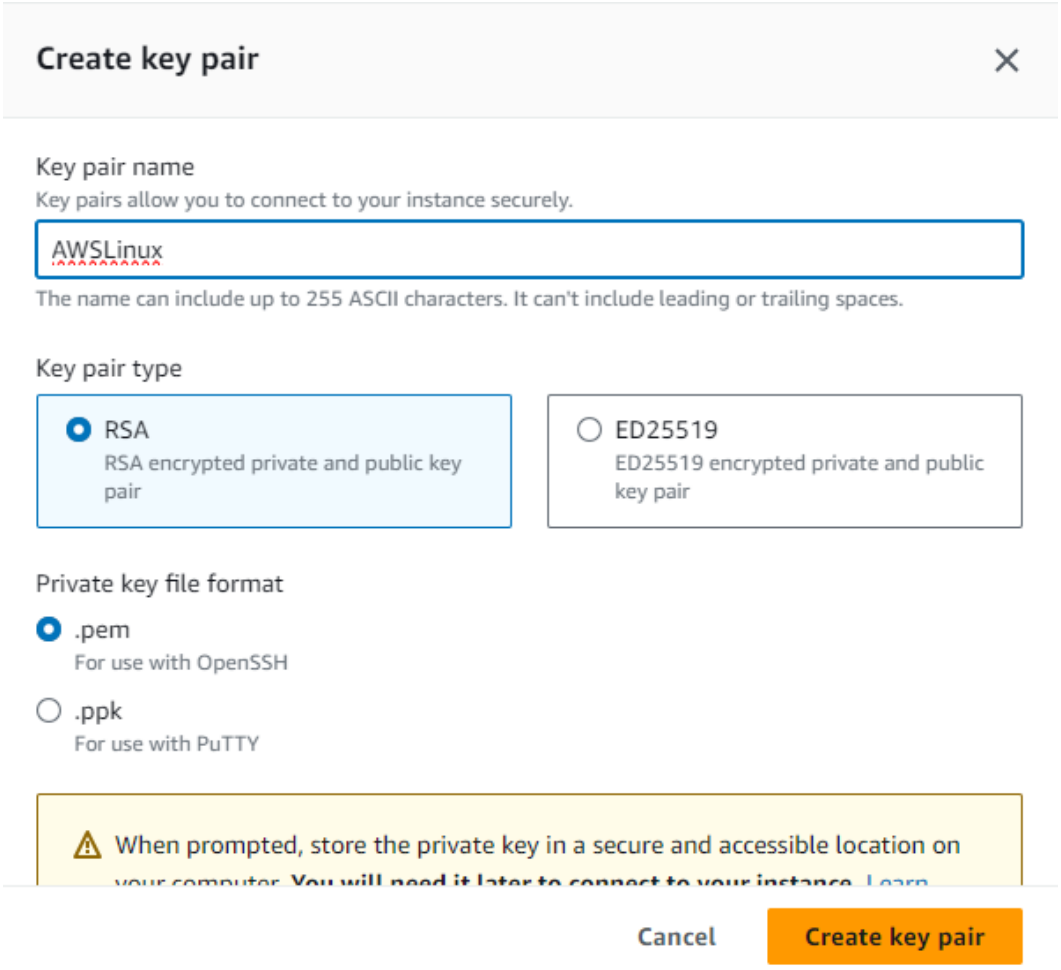
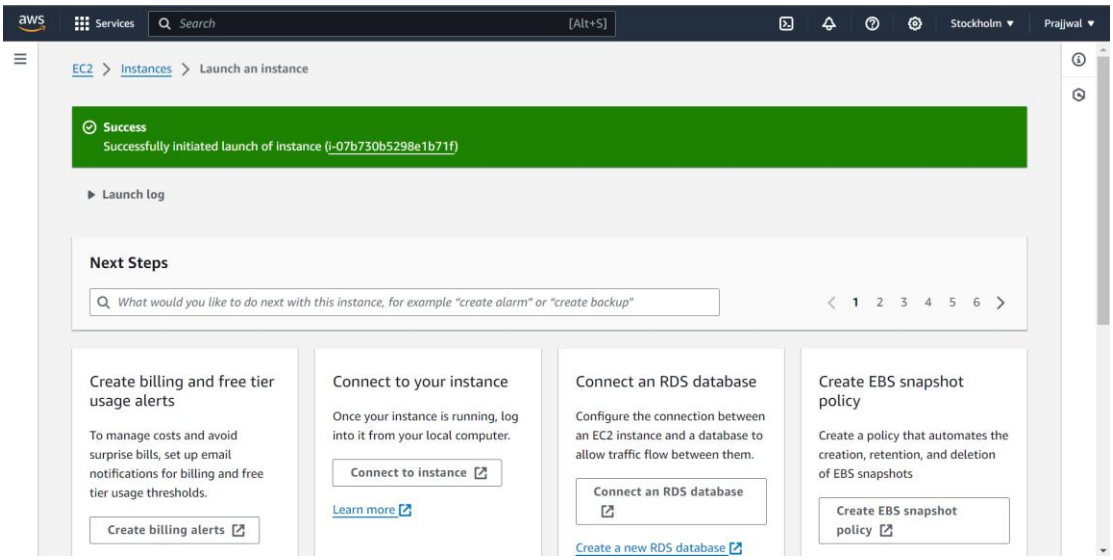
Adv Devops exp-1.a

Prajjwal Pandey
D15A - 33

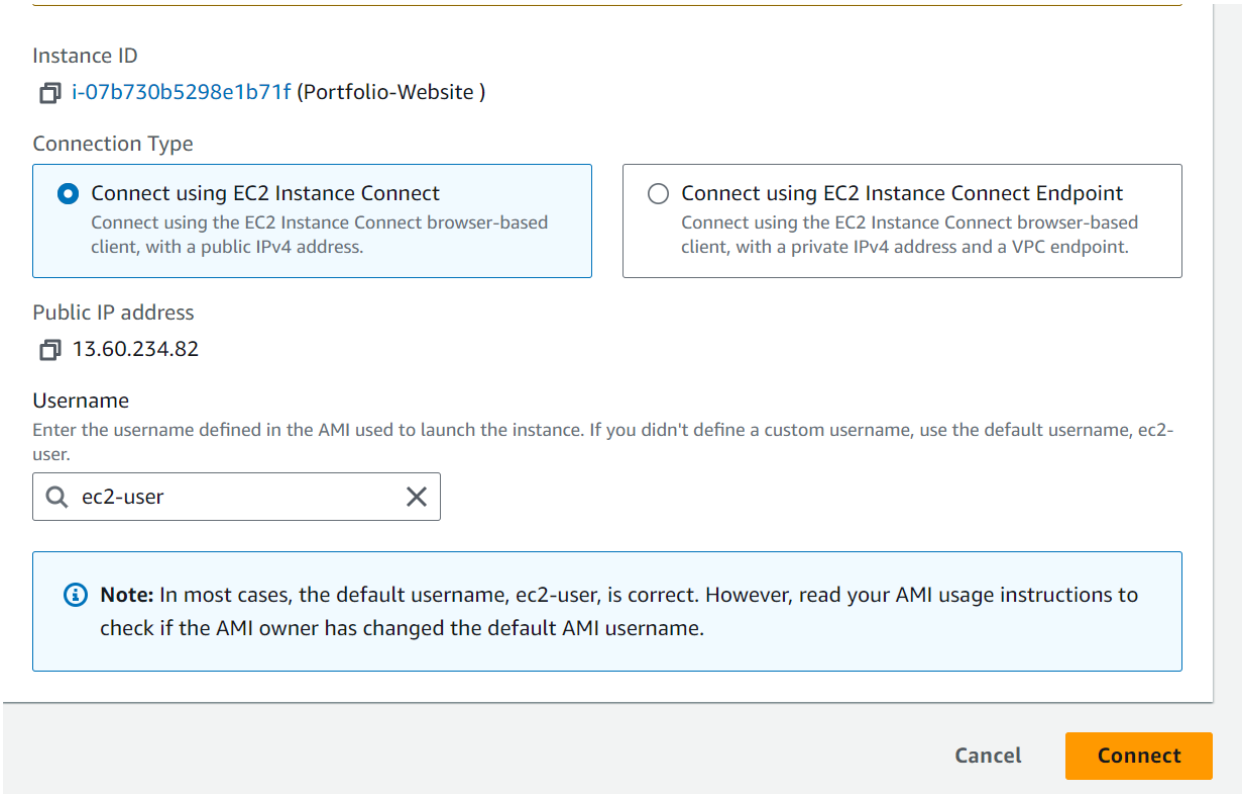
1. Creating key-pair



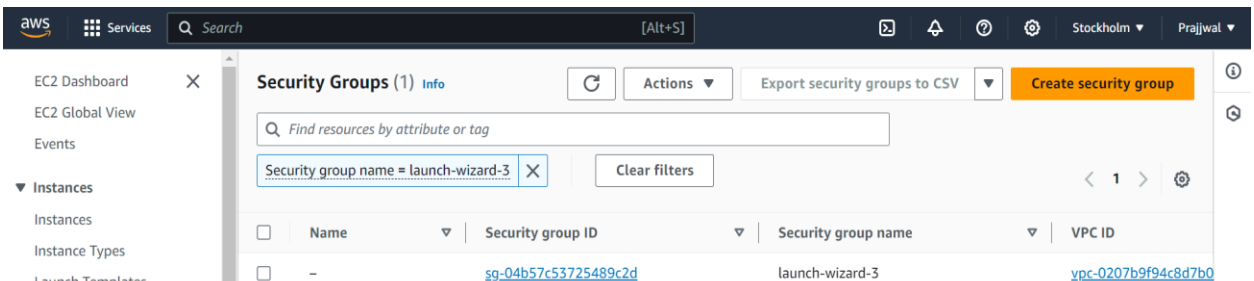
2. Launching Instance

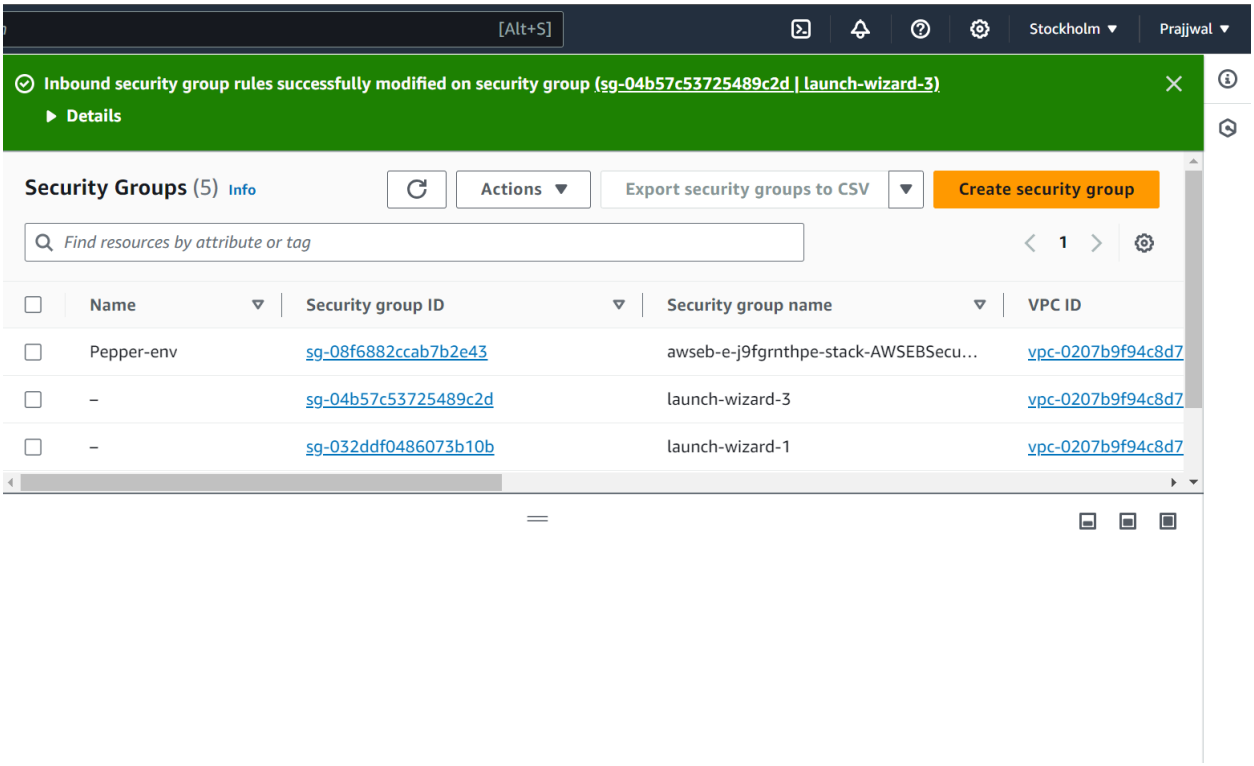


3. Connecting using EC2 instance

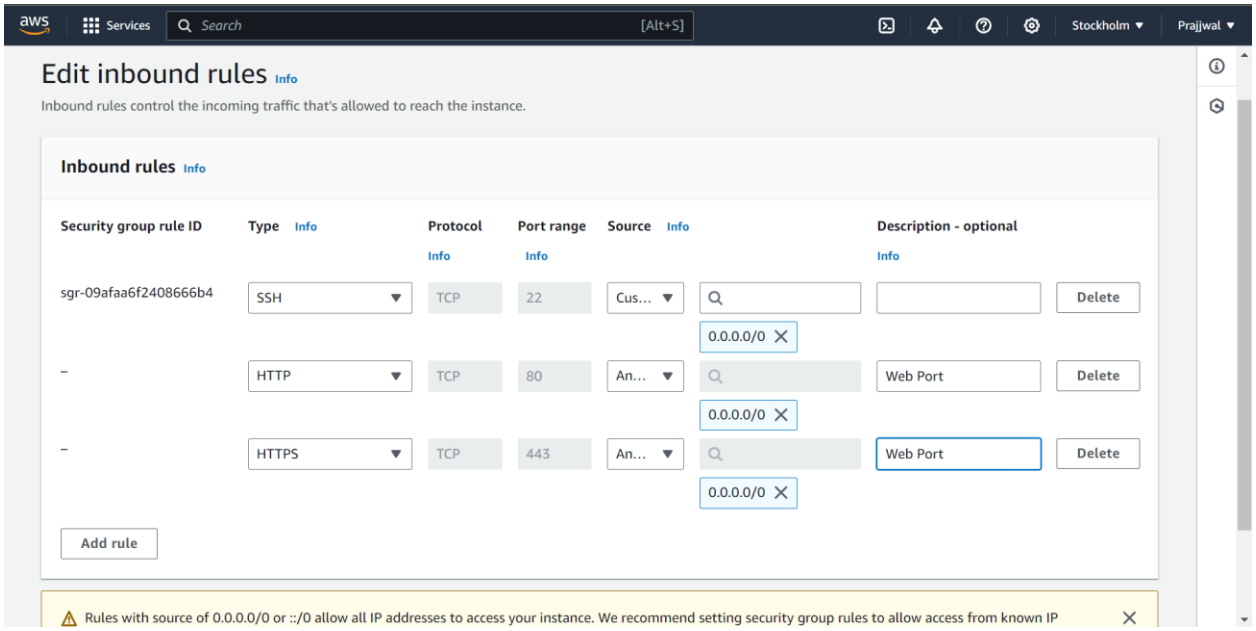


4. Launching wizard

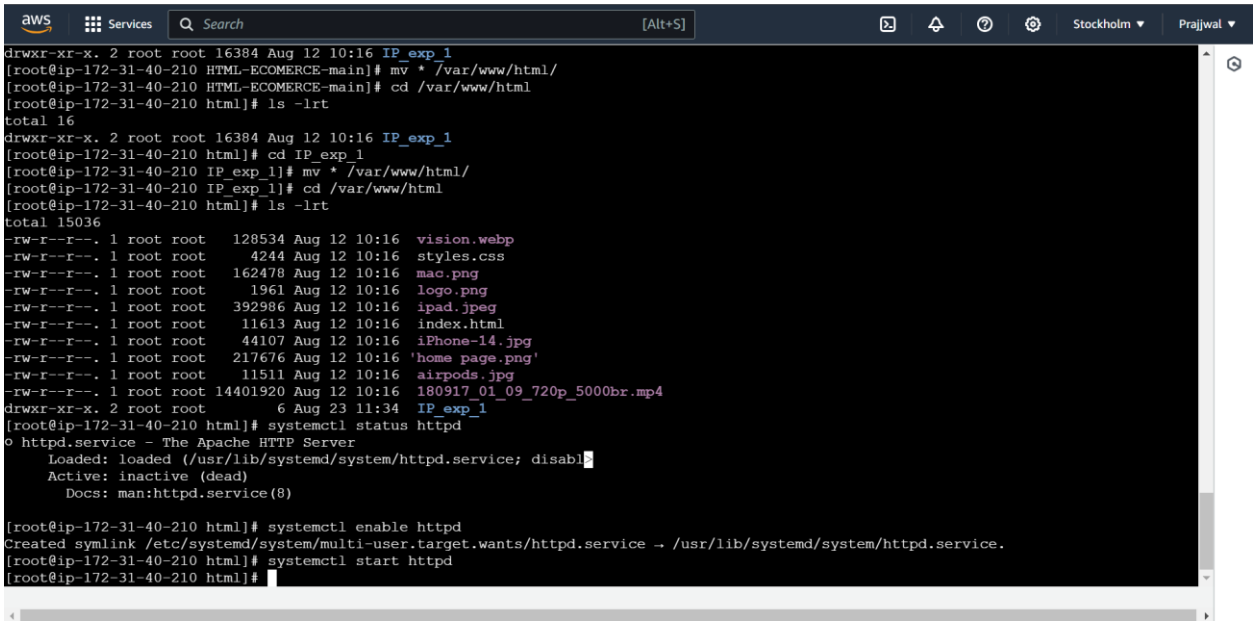




5. Adding Inbound rules


aws
Services
Search
[Alt+S]
Stockholm
Prajjwal
Edit inbound rules Info
Inbound rules control the incoming traffic that's allowed to reach the instance.
Inbound rules Info
Security group rule ID
Type Info
Protocol Info
Port range Info
Source Info
Description - optional Info
sgr-09afaa6f2408666b4
SSH
TCP
22
Cus...
0.0.0.0/0
Delete
HTTP
TCP
80
An...
0.0.0.0/0
Web Port
Delete
HTTPS
TCP
443
An...
0.0.0.0/0
Web Port
Delete
Add rule
Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP

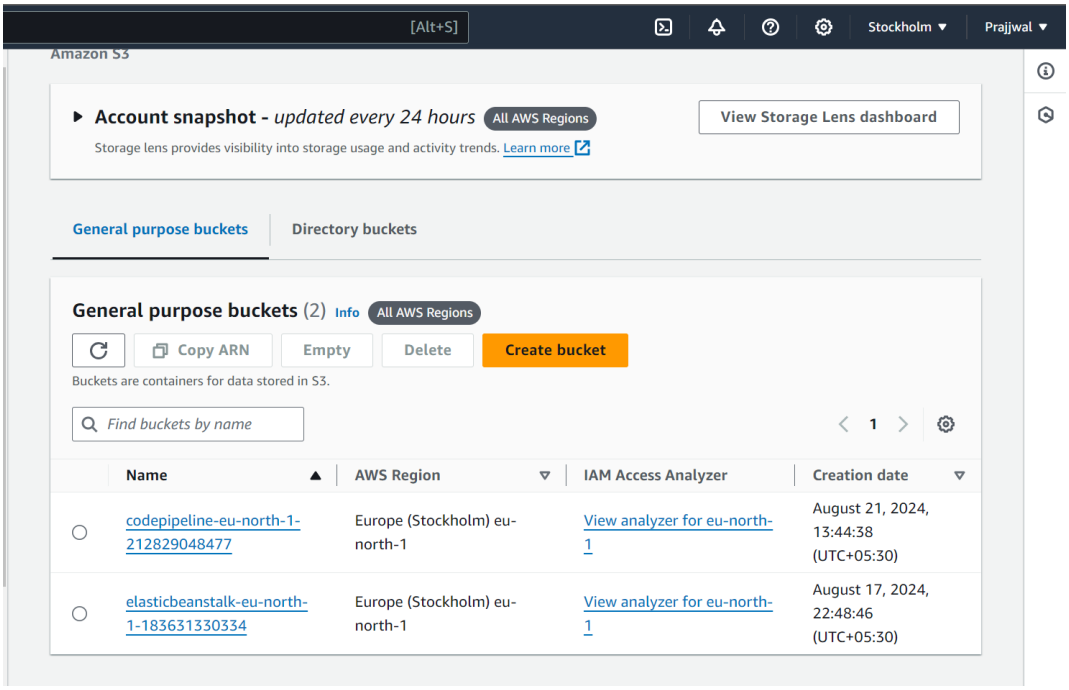
6. Starting

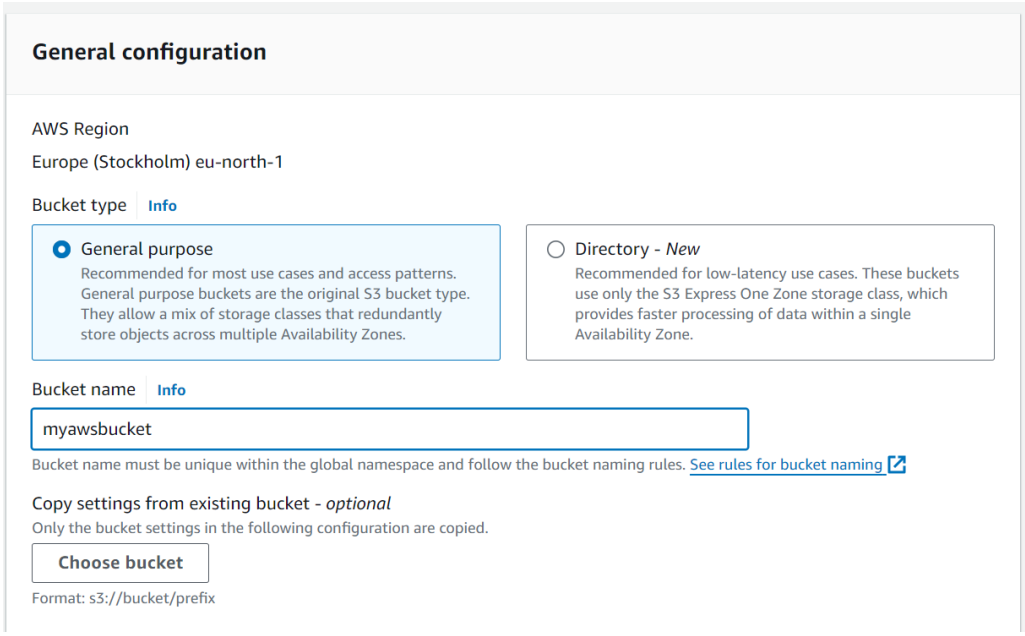

aws
Services
Search
[Alt+S]
Stockholm
Prajjwal
drwxr-xr-x. 2 root root 16384 Aug 12 10:16 IP_exp_1
[root@ip-172-31-40-210 HTML-ECOMERCE-main]# mv * /var/www/html/
[root@ip-172-31-40-210 HTML-ECOMERCE-main]# cd /var/www/html
[root@ip-172-31-40-210 html]# ls -lrt
total 16
drwxr-xr-x. 2 root root 16384 Aug 12 10:16 IP_exp_1
[root@ip-172-31-40-210 html]# cd IP_exp_1
[root@ip-172-31-40-210 IP_exp_1]# mv * /var/www/html/
[root@ip-172-31-40-210 IP_exp_1]# cd /var/www/html
[root@ip-172-31-40-210 html]# ls -lrt
total 15036
-rw-r--r--. 1 root root 128534 Aug 12 10:16 vision.webp
-rw-r--r--. 1 root root 4244 Aug 12 10:16 styles.css
-rw-r--r--. 1 root root 162478 Aug 12 10:16 mac.png
-rw-r--r--. 1 root root 1961 Aug 12 10:16 logo.png
-rw-r--r--. 1 root root 392986 Aug 12 10:16 ipad.jpeg
-rw-r--r--. 1 root root 11613 Aug 12 10:16 index.html
-rw-r--r--. 1 root root 44107 Aug 12 10:16 iPhone-14.jpg
-rw-r--r--. 1 root root 217676 Aug 12 10:16 'home page.png'
-rw-r--r--. 1 root root 11511 Aug 12 10:16 airpods.jpg
-rw-r--r--. 1 root root 14401920 Aug 12 10:16 180917_01_09_720p_5000br.mp4
drwxr-xr-x. 2 root root 6 Aug 23 11:34 IP_exp_1
[root@ip-172-31-40-210 html]# systemctl status httpd
o httpd.service - The Apache HTTP Server
Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabl
Active: inactive (dead)
Docs: man:httpd.service(8)
[root@ip-172-31-40-210 html]# systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-40-210 html]# systemctl start httpd
[root@ip-172-31-40-210 html]#

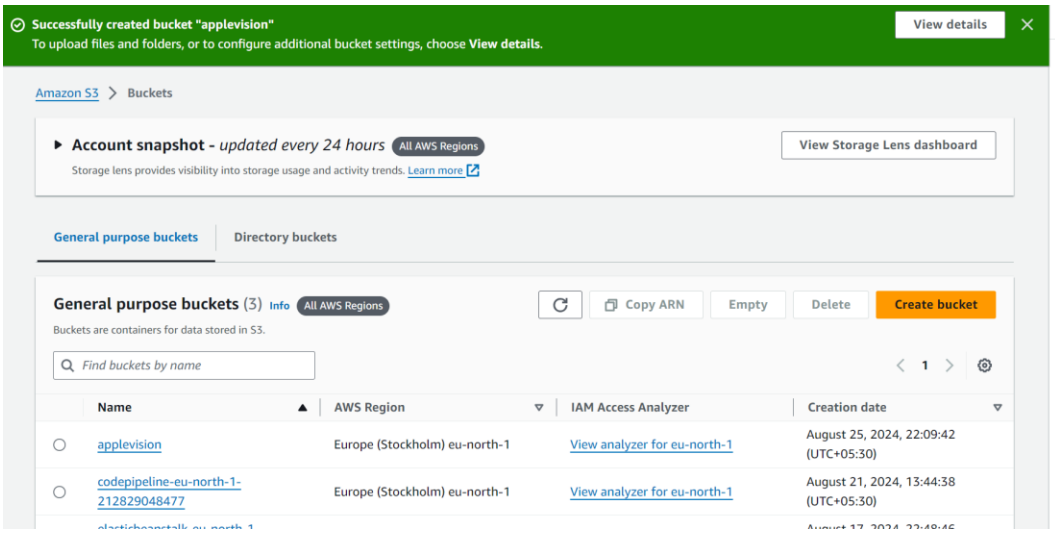

Store
Mac
iPad
iPhone
Airpods
Search
APPLE VISION PRO

B) Hosting a static website on Amazon S3.

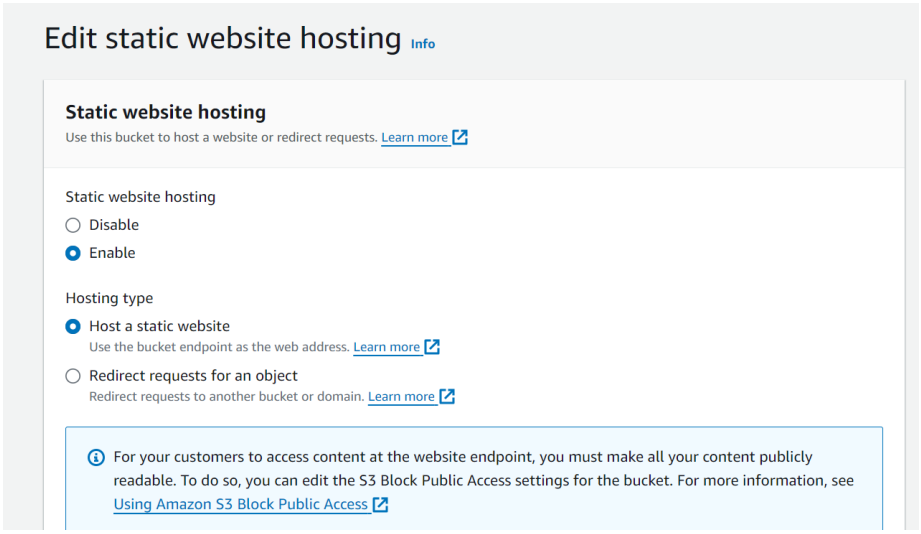
1.Creating a bucket.



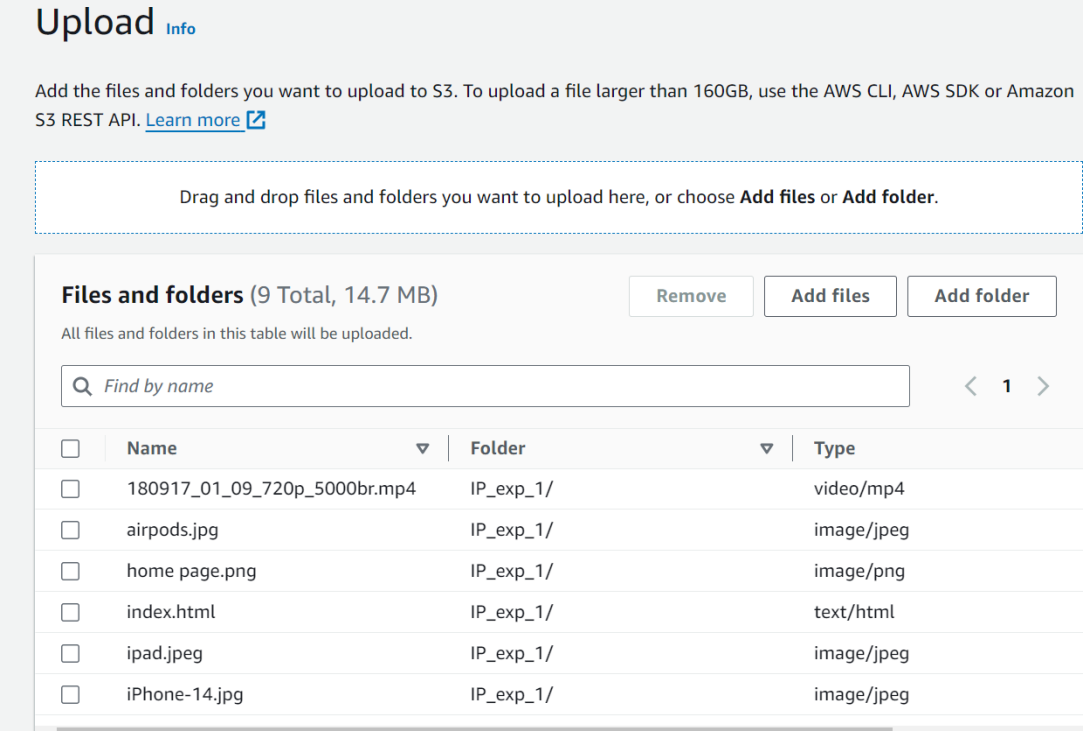


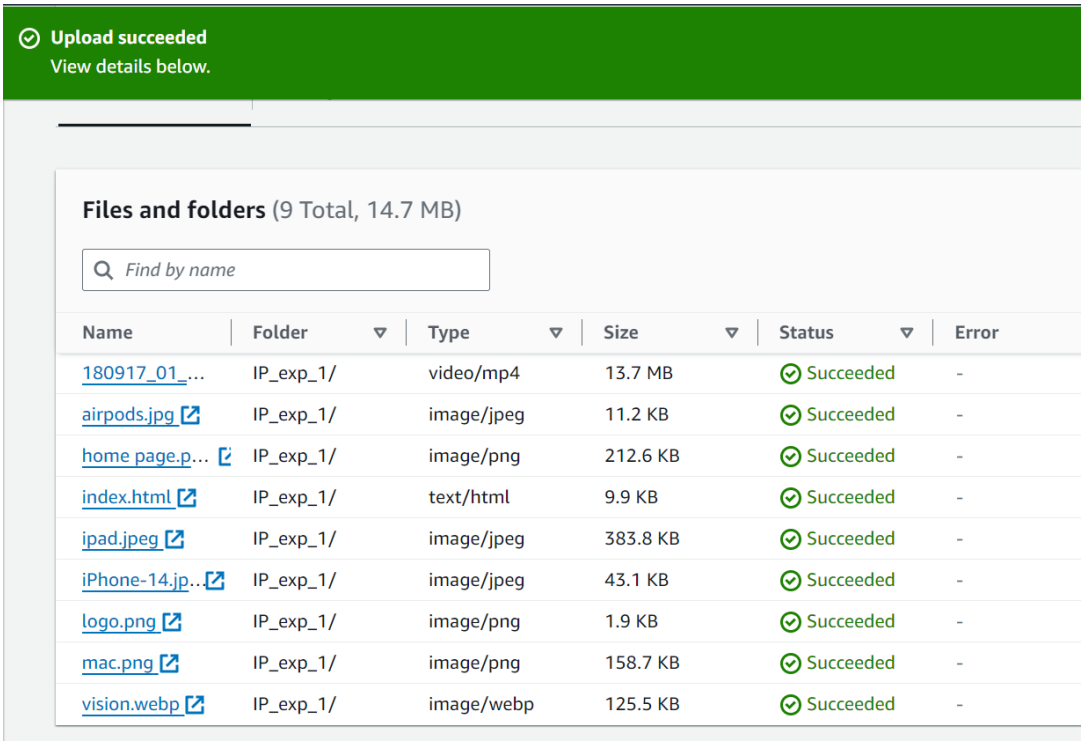


2. Enable static website hosting

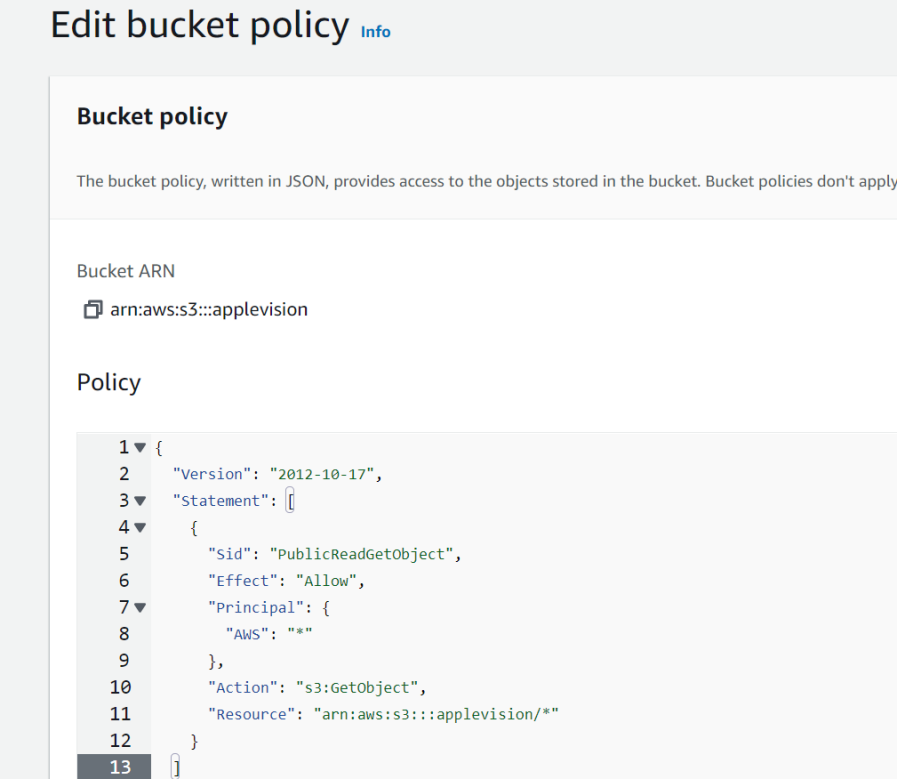


3. Upload your files

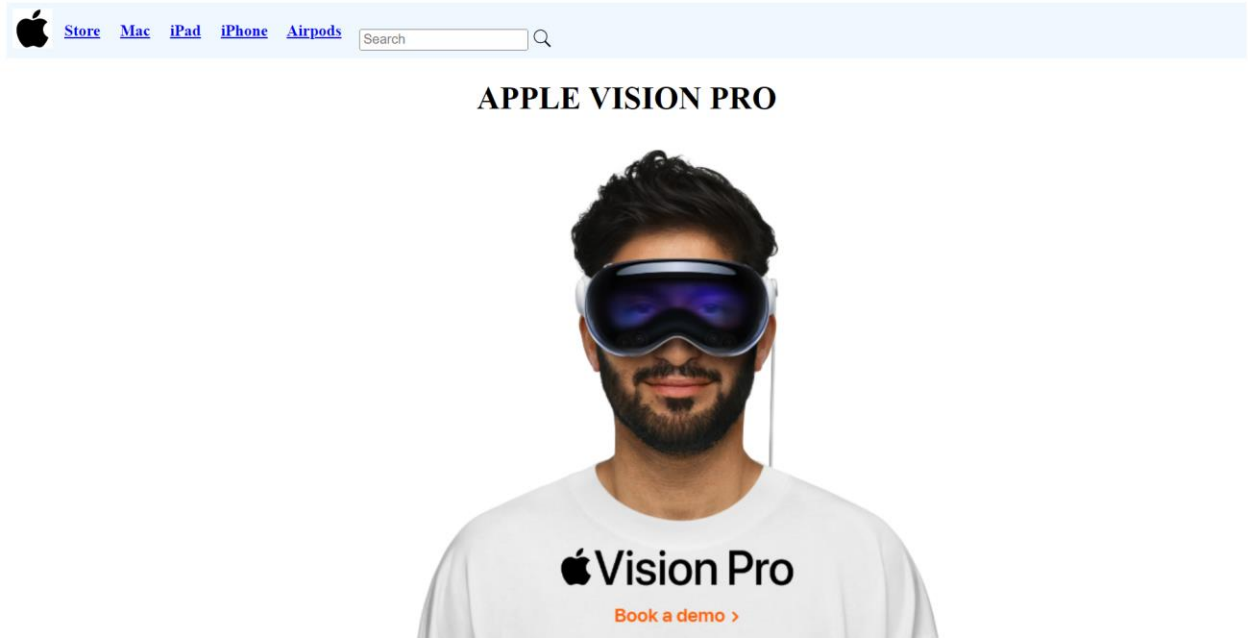




4. Add bucket policy



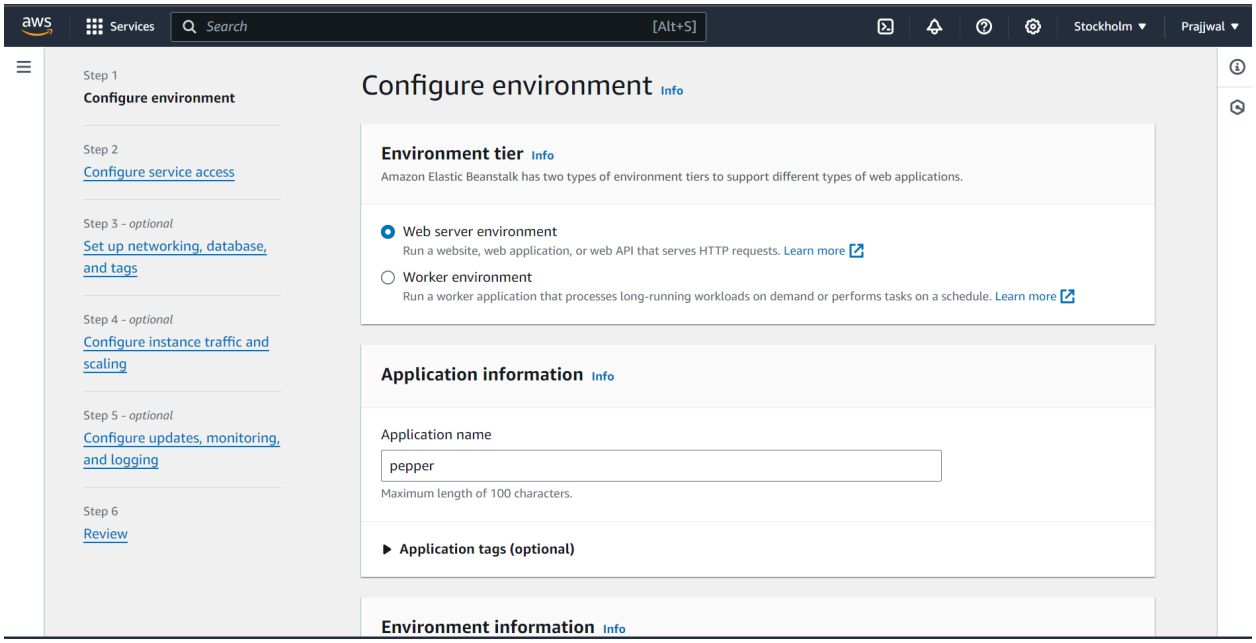
5. Go to URL in permissions



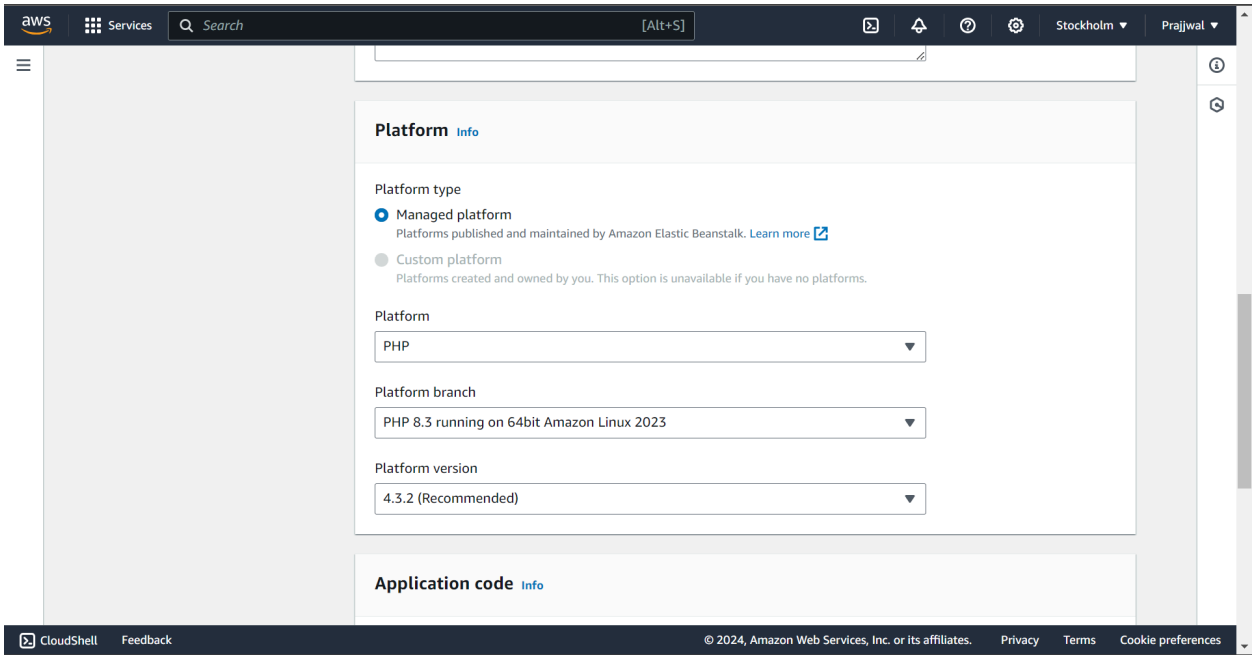
AdvDevops
Exp-2 Elastic Beanstalk

Prajjwal Pandey D15A - 33

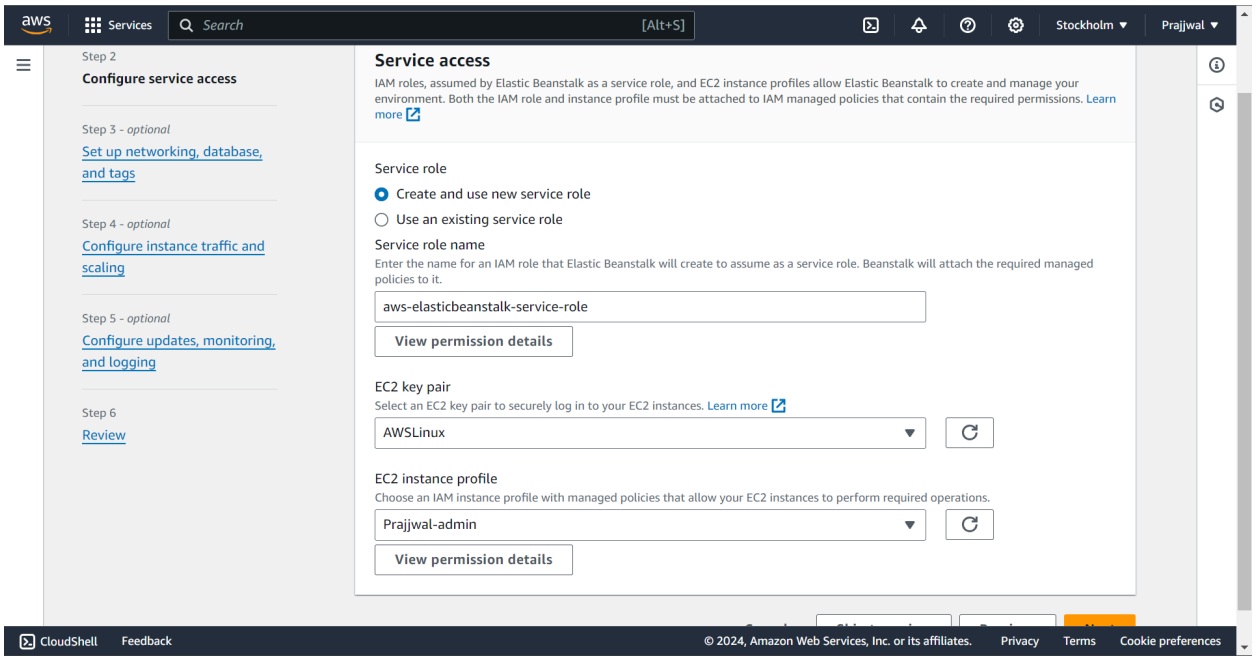
1. Open up Elastic Beanstalk and name your web app.



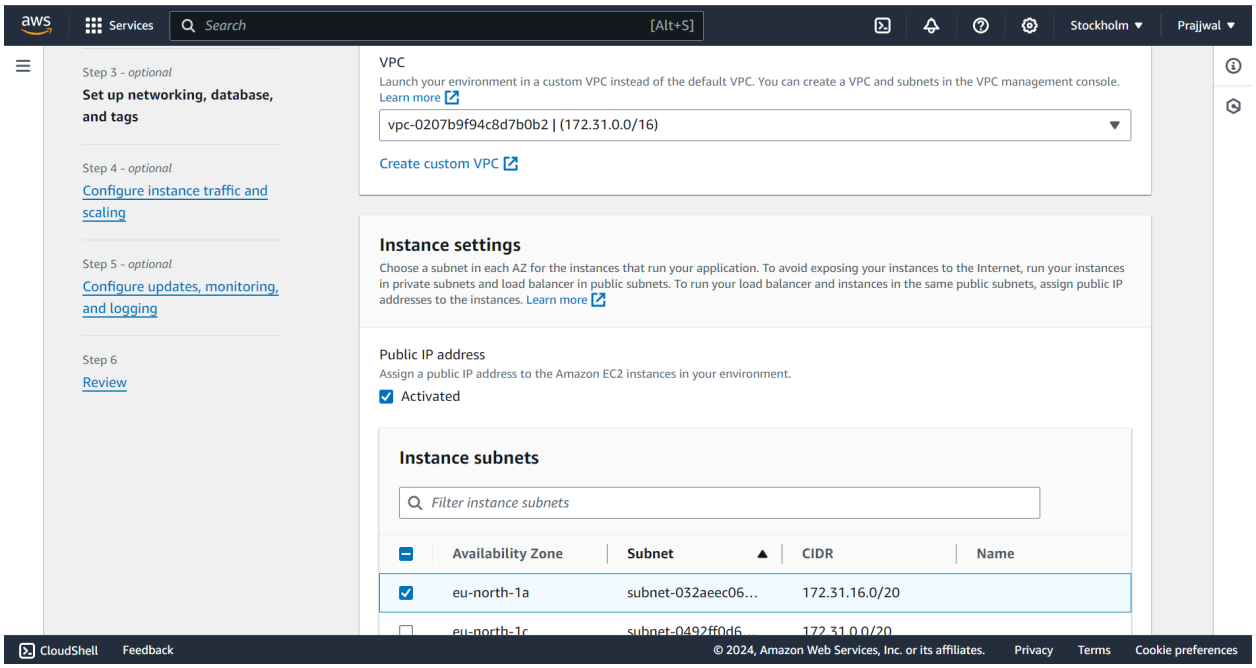
2. Choose PHP from the drop-down menu and then click Create Application.



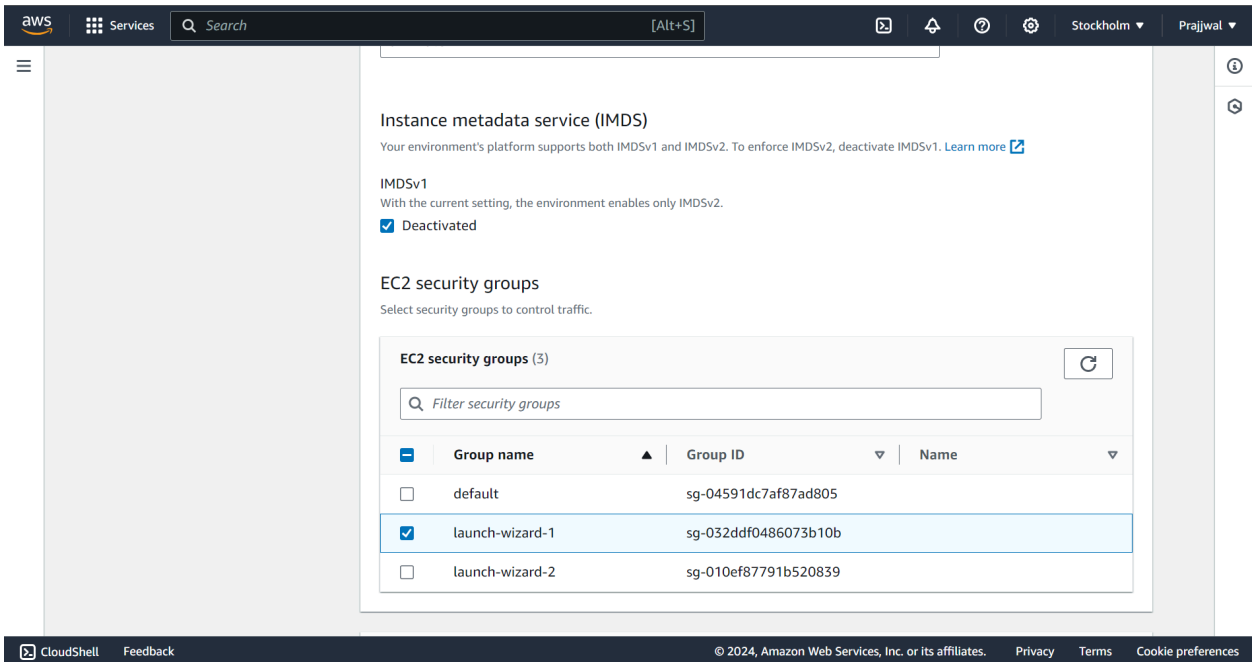
3. Give Key pair



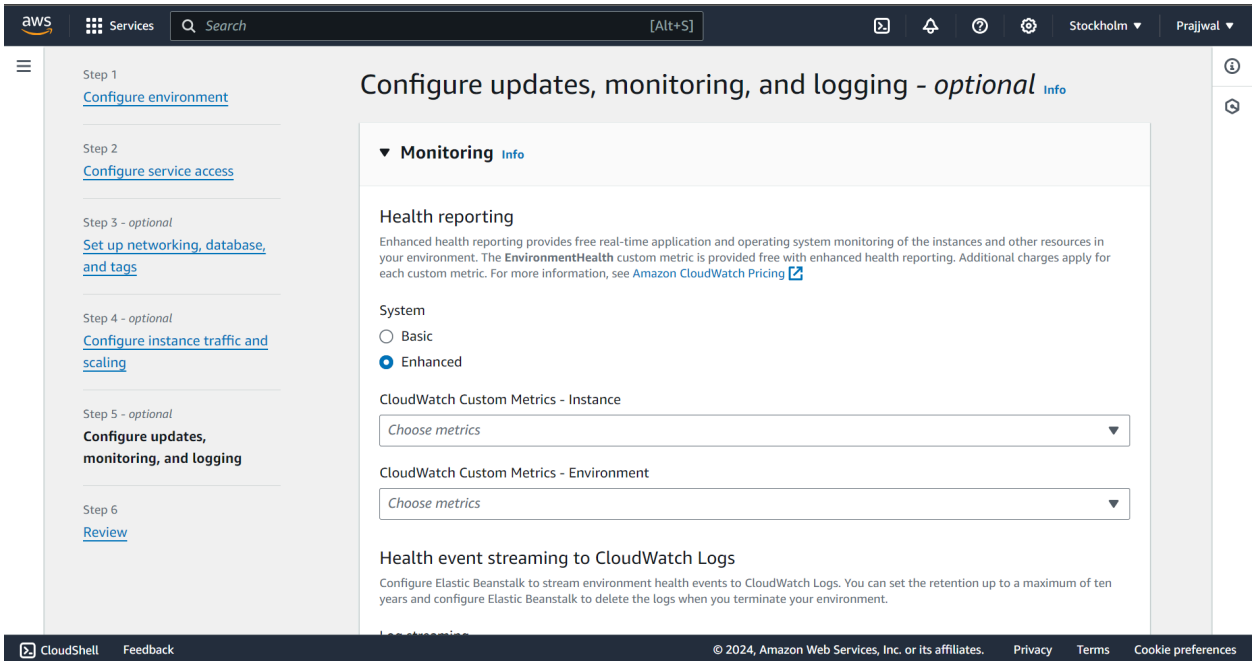
4. Instance settings



5. Select security groups



6. Configure updates, monitoring and logging



Congratulations!

Your AWS Elastic Beanstalk *PHP* application is now running on your own dedicated environment in the AWS Cloud

You are running PHP version 8.3.7

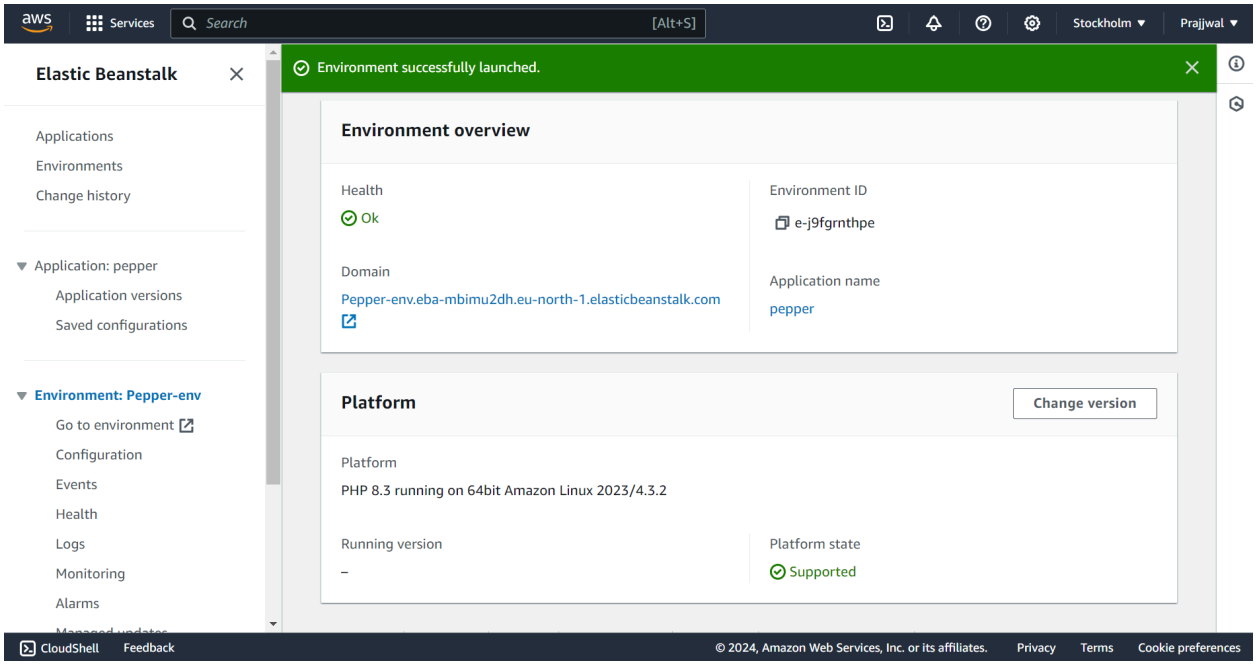
This environment is launched with Elastic Beanstalk PHP Platform

What's Next?

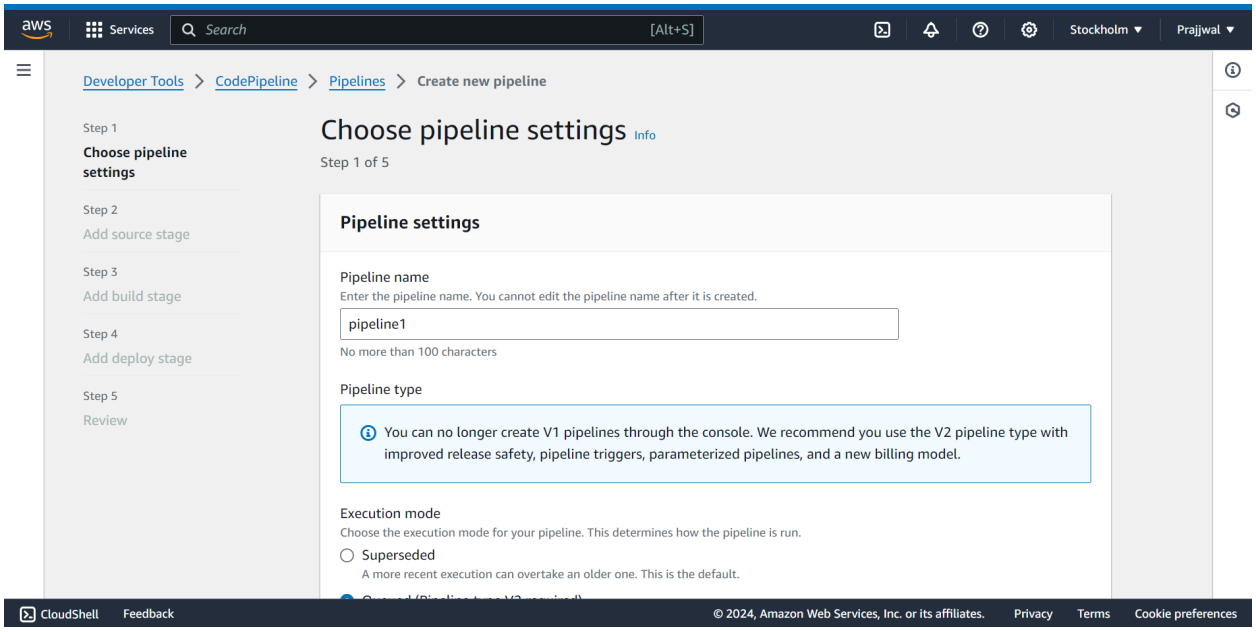
- AWS Elastic Beanstalk overview
- Deploying AWS Elastic Beanstalk Applications in PHP Using Eb and Git
- Using Amazon RDS with PHP
- Customizing the Software on EC2 Instances
- Customizing Environment Resources

AWS SDK for PHP

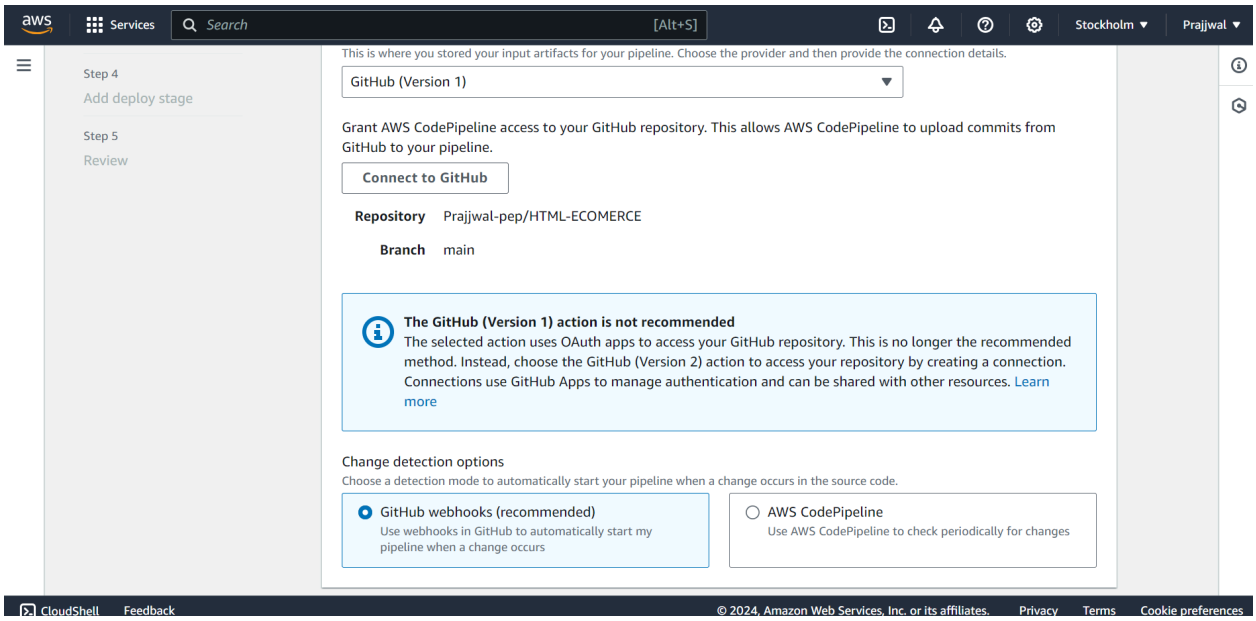
- AWS SDK for PHP home
- PHP developer center
- AWS SDK for PHP on GitHub



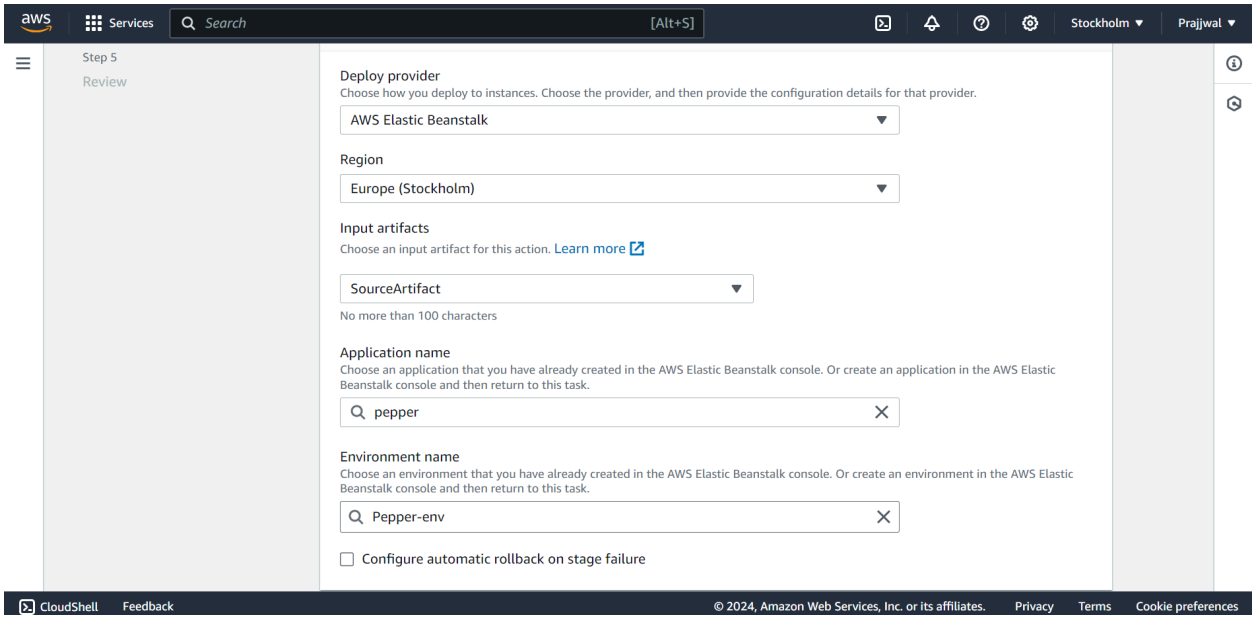
7. Choose pipeline settings

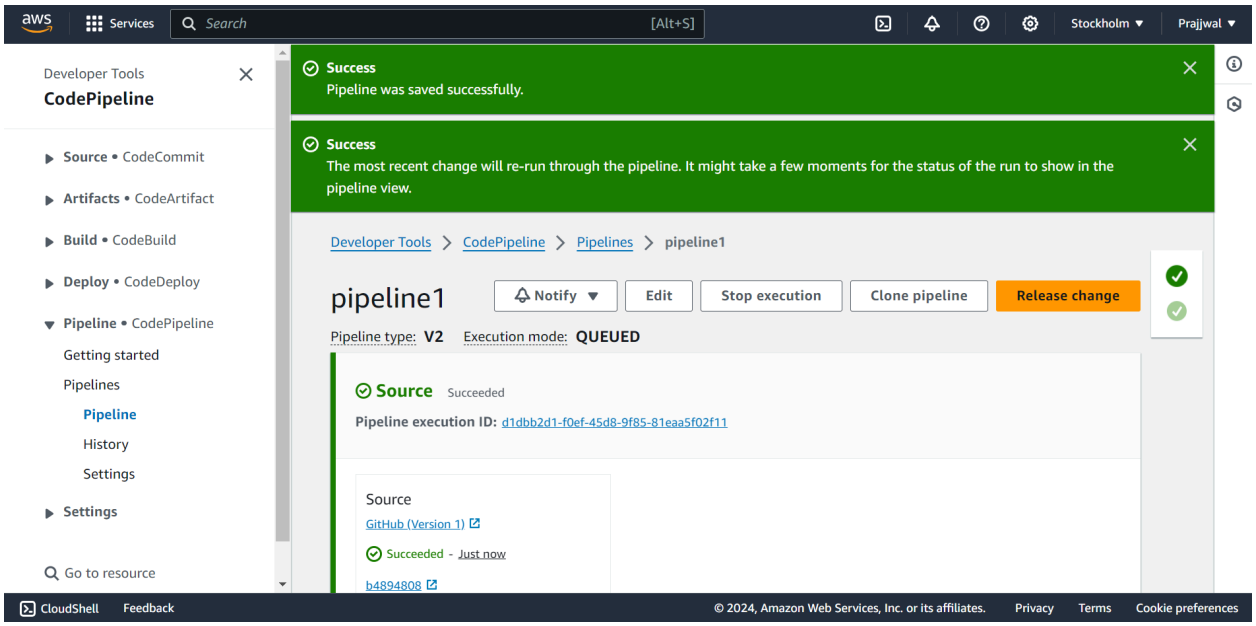


8. Connect to GitHub



9. Choose input artifacts and application name



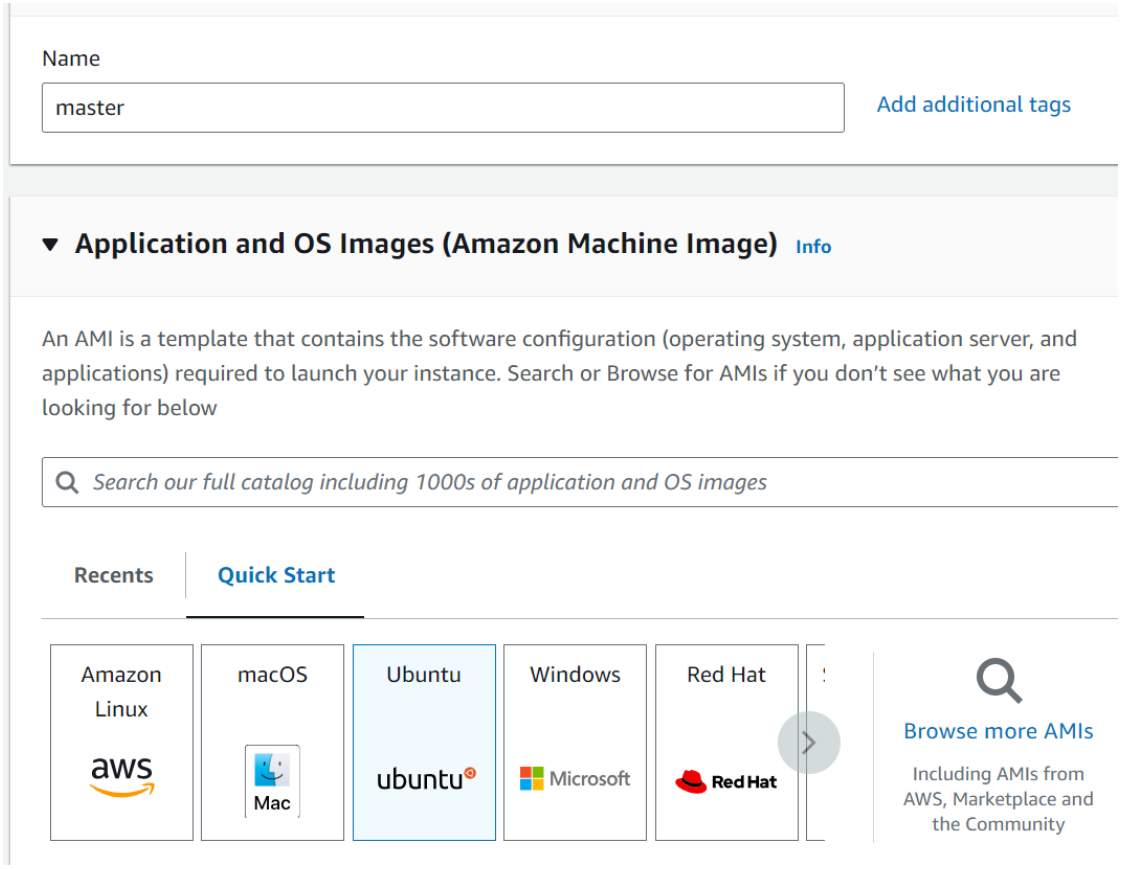


ADVANCE DEVOPS EXPERIMENT 3

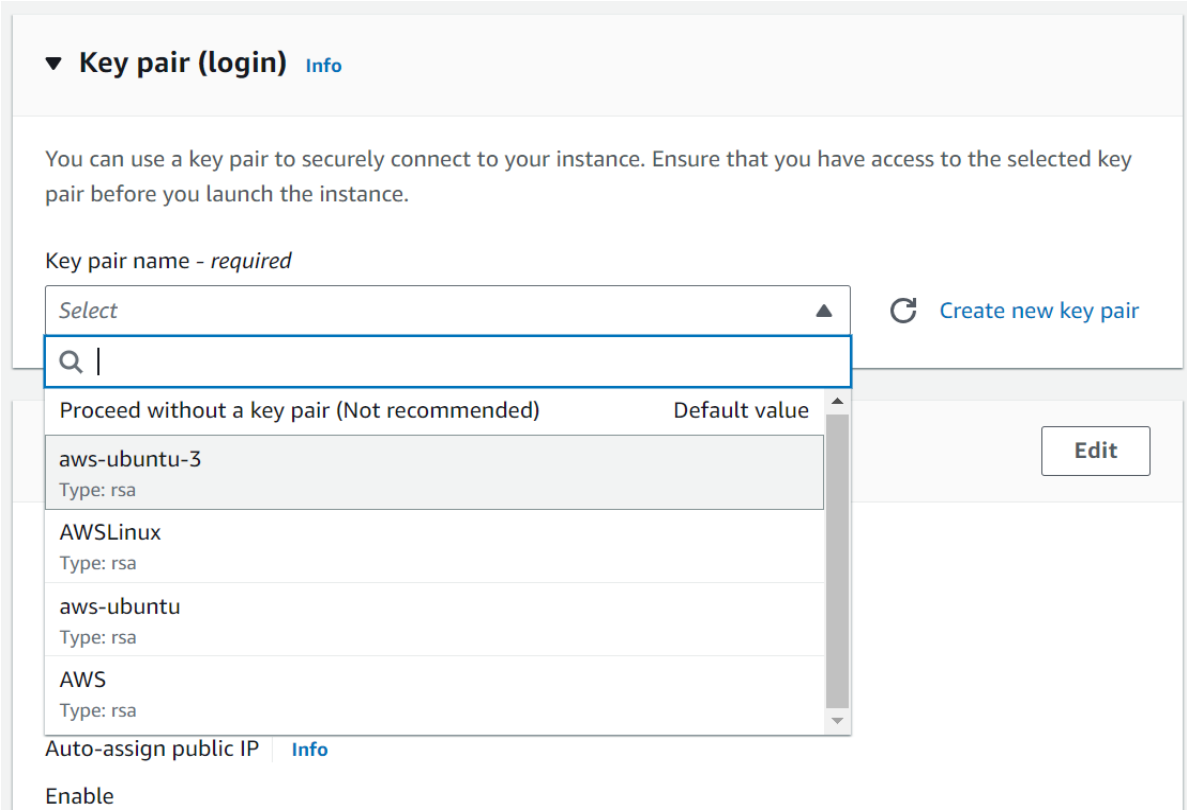
Prajjwal Pandey – D15A/33

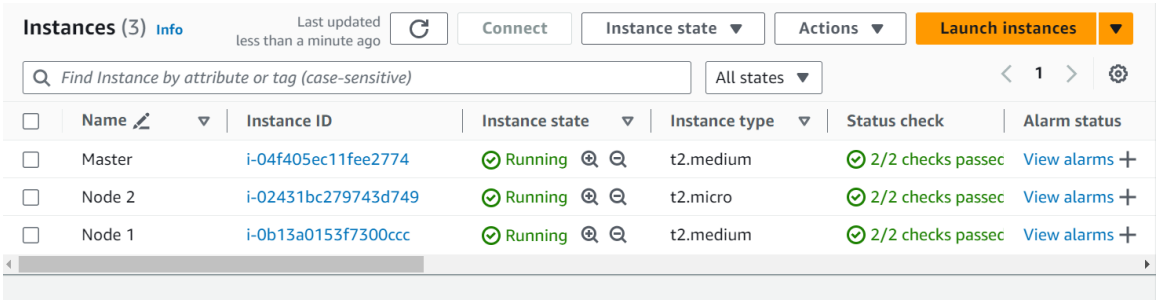
Aim:To understand the Kubernetes Cluster Architecture, install and Spin Up a Kubernetes Cluster on Linux Machines/Cloud Platforms.

1.1 Create 3 EC2 instances master, node 1 and node 2.

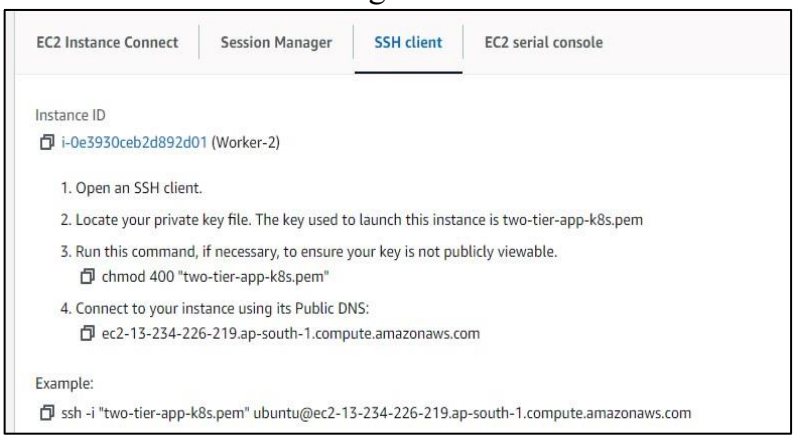


1.2 Create a new key pair (use the same key pair for all 3 instances)





1.3 After the instances have been created, copy the text given in the example part of each of the three instances into git bash.

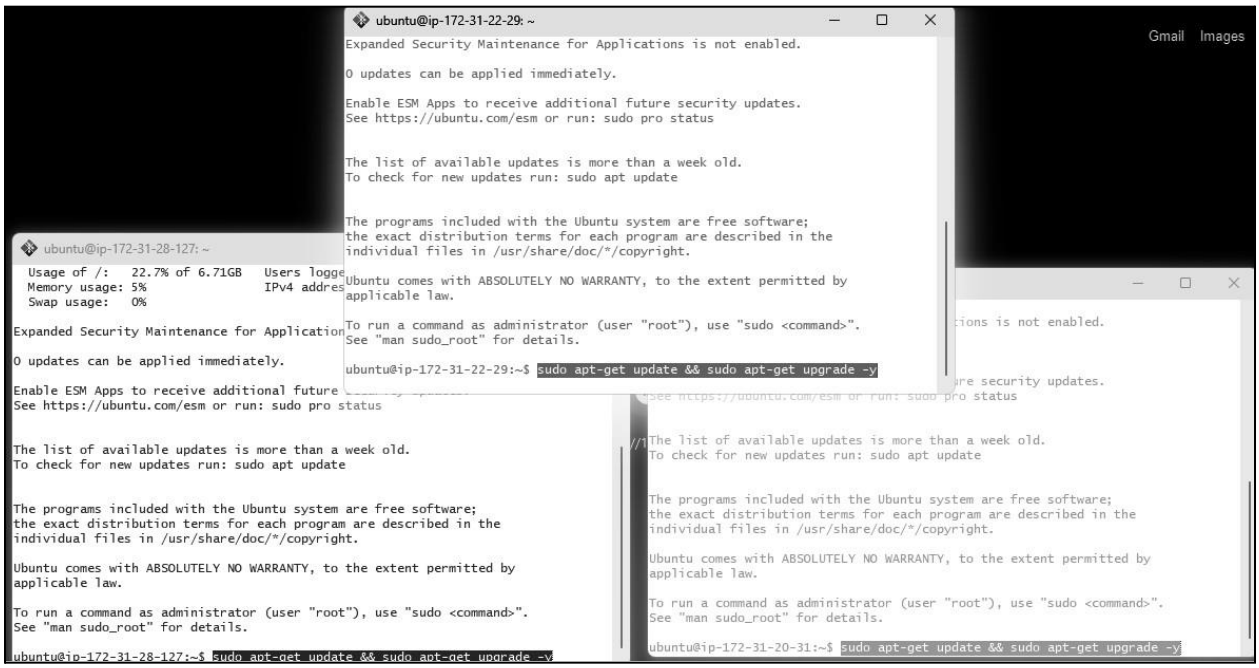


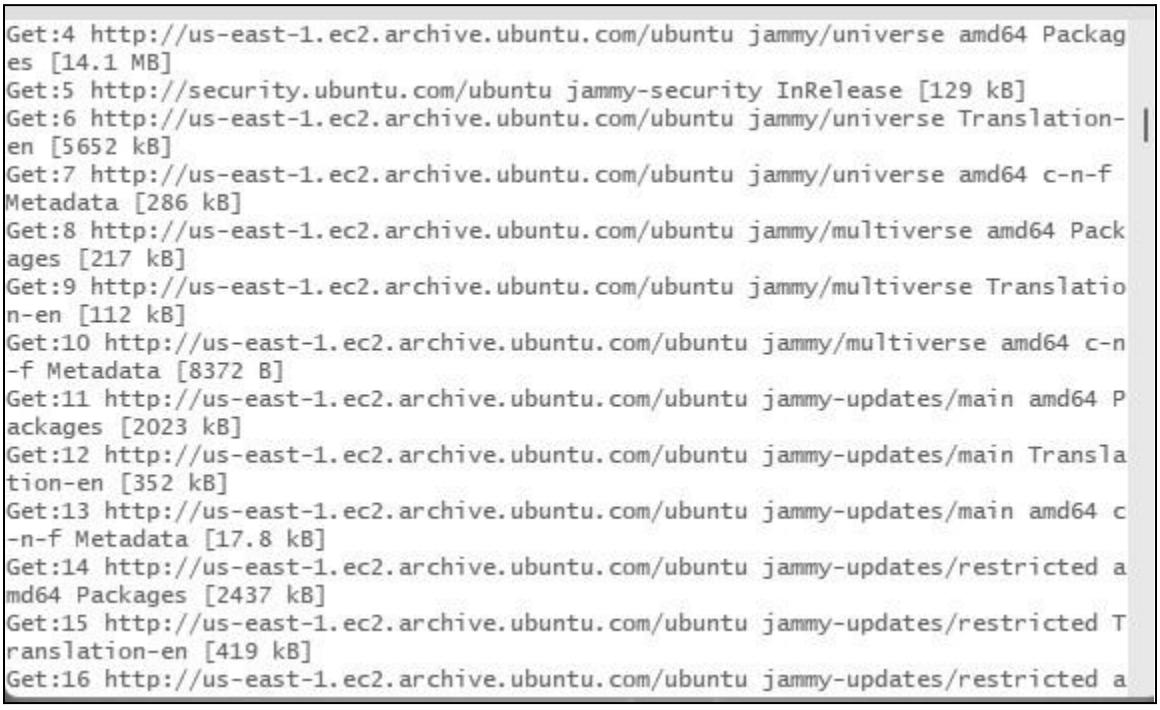

```
acer@TMP214-53 MINGW64 ~/Downloads
$ ssh -i "two-tier-app-k8s.pem" ubuntu@ec2-13-232-36-34.ap-south-1.compute.amazo
naws.com
The authenticity of host 'ec2-13-232-36-34.ap-south-1.compute.amazonaws.com (13.
232.36.34)' can't be established.
ED25519 key fingerprint is SHA256:uVGEO+FWYefj6Oj0ft7OSralv8NrzEi/IwxAtBY+EPE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-232-36-34.ap-south-1.compute.amazonaws.com' (
ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-1022-aws x86_64)

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

 System information as of Wed Sep 11 14:07:10 UTC 2024

  System load:  0.0               Processes:             106
  Usage of /:   20.7% of 7.57GB   Users logged in:       0
  Memory usage: 5%                IPv4 address for eth0: 172.31.45.227
  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
```

1.3 After the instances have been created,copy the text given in the example part of each of the three instances into git bash.

Step 2:Prepare Nodes

2.1. Update the package manager on all nodes:
sudo apt-get update && sudo apt-get upgrade -y





2.2. Disable Swap (Kubernetes requires swap to be off):
sudo swapoff -a sudo sed -i '/ swap / s/^/#/' /etc/fstab

```
ubuntu@ip-172-31-22-29:~$ sudo swapoff -a
sudo sed -i '/ swap / s/^/#/' /etc/fstab
```

2.3. Load necessary kernel modules for networking and iptables:
cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
overlay br_netfilter
EOF
sudo modprobe overlay
sudo modprobe br_netfilter

```
ubuntu@ip-172-31-22-29:~$ cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
overlay
br_netfilter
EOF
sudo modprobe overlay
sudo modprobe br_netfilter
overlay
br_netfilter
```

2.4. Configure sysctl settings for Kubernetes networking:
cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf net.bridge.bridge-nf-call-
ip6tables = 1 net.bridge.bridge-nf-call-iptables = 1
EOF
sudo sysctl --system

```
ubuntu@ip-172-31-22-29:~$ cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
overlay
br_netfilter
EOF
sudo modprobe overlay
sudo modprobe br_netfilter
overlay
br_netfilter
ubuntu@ip-172-31-22-29:~$ cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
EOF
sudo sysctl --system
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
* Applying /etc/sysctl.d/10-console-messages.conf ...
kernel.printk = 4 4 1 7
* Applying /etc/sysctl.d/10-ipv6-privacy.conf ...
net.ipv6.conf.all.use_tempaddr = 2
net.ipv6.conf.default.use_tempaddr = 2
* Applying /etc/sysctl.d/10-kernel-hardening.conf ...
kernel.kptr_restrict = 1
```

Step 3: Install Docker

Kubernetes uses container runtimes like Docker. Install Docker on all nodes.

sudo apt-get update sudo apt-get install -y apt-transport-https ca-certificates curl software-properties-common curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" sudo apt-get update sudo apt-get install -y docker-ce docker-ce-cli containerd.io

```
ubuntu@ip-172-31-22-29:~$ sudo apt-get update
sudo apt-get install -y apt-transport-https ca-certificates curl software-proper
ties-common
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubun
tu $(lsb_release -cs) stable"
sudo apt-get update
sudo apt-get install -y docker-ce docker-ce-cli containerd.io
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Fetched 129 kB in 1s (241 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20230311ubuntu0.22.04.1).
ca-certificates set to manually installed.
curl is already the newest version (7.81.0-1ubuntu1.17).
curl set to manually installed.
software-properties-common is already the newest version (0.99.22.9).
software-properties-common set to manually installed.
```

Configure Docker for Kubernetes:

```
cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
```

EOF
sudo systemctl restart docker

```
ubuntu@ip-172-31-22-29:~$ cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
EOF
sudo systemctl restart docker
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
```

Step 4: Install kubeadm, kubelet, kubectl

Install Kubernetes tools on all nodes.

4.1. Add Kubernetes APT repository:

sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list

```
ubuntu@ip-172-31-22-29:~$ sudo curl -fsSLo /usr/share/keyrings/kubernetes-archiv
e-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg
echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https:/
/apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/k
ubernetes.list
deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.k
ubernetes.io/ kubernetes-xenial main
```

4.2. Install kubeadm, kubelet, and kubectl:
sudo apt-get update sudo apt-get install -y kubelet kubeadm kubectl sudo apt-mark hold kubelet kubeadm kubectl

```
ubuntu@ip-172-31-22-29:~$ sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:5 https://download.docker.com/linux/ubuntu jammy InRelease
```

Step 5: Initialize the Kubernetes Cluster on Master Node

On the master node:

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

```
ubuntu@ip-172-31-22-29:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --v
=5
Found multiple CRI endpoints on the host. Please define which one do you wish to
 use by setting the 'criSocket' field in the kubeadm configuration file: unix://
/var/run/containerd/containerd.sock, unix:///var/run/crio/crio.sock
k8s.io/kubernetes/cmd/kubeadm/app/util/runtime.detectCRISocketImpl
        cmd/kubeadm/app/util/runtime/runtime.go:167
k8s.io/kubernetes/cmd/kubeadm/app/util/runtime.DetectCRISocket
        cmd/kubeadm/app/util/runtime/runtime.go:175
k8s.io/kubernetes/cmd/kubeadm/app/util/config.SetNodeRegistrationDynamicDefaults
        cmd/kubeadm/app/util/config/initconfiguration.go:118
k8s.io/kubernetes/cmd/kubeadm/app/util/config.SetInitDynamicDefaults
        cmd/kubeadm/app/util/config/initconfiguration.go:64
k8s.io/kubernetes/cmd/kubeadm/app/util/config.DefaultedInitConfiguration
        cmd/kubeadm/app/util/config/initconfiguration.go:248
k8s.io/kubernetes/cmd/kubeadm/app/util/config.LoadOrDefaultInitConfiguration
        cmd/kubeadm/app/util/config/initconfiguration.go:282
k8s.io/kubernetes/cmd/kubeadm/app/cmd.newInitData
        cmd/kubeadm/app/cmd/init.go:319
k8s.io/kubernetes/cmd/kubeadm/app/cmd.newCmdInit.func3
        cmd/kubeadm/app/cmd/init.go:170
k8s.io/kubernetes/cmd/kubeadm/app/cmd/phases/workflow.(*Runner).InitData
        cmd/kubeadm/app/cmd/phases/workflow/runner.go:183
k8s.io/kubernetes/cmd/kubeadm/app/cmd.newCmdInit.func1
```

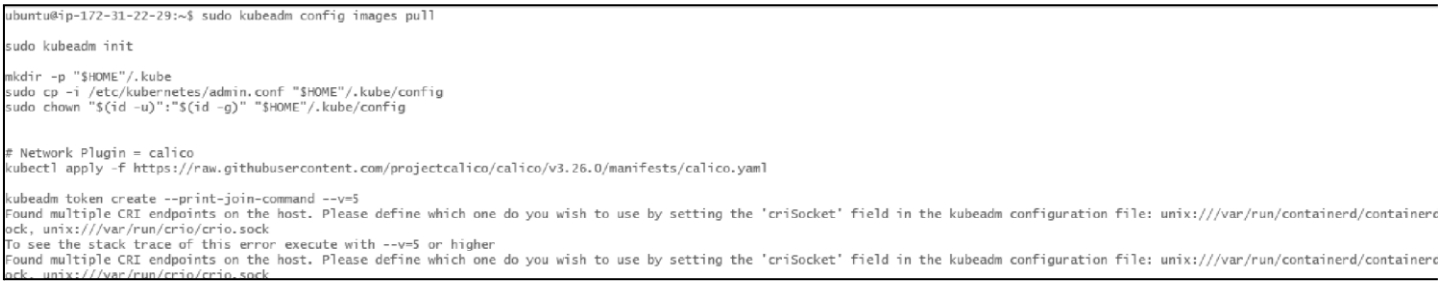
5.1. Set up kubectl on the master node:

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

Step 6: Install a Pod Network Add-on

To enable communication between pods, install a pod network plugin like Flannel or Calico.

Install Flannel:

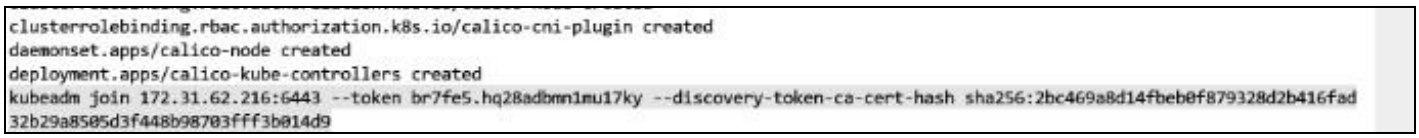


kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml

Step 7: Join Worker Nodes to the Cluster

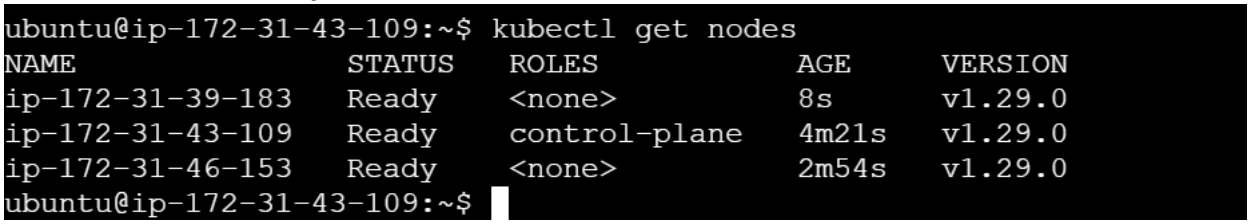
On the **worker nodes**, run the command provided by the master node during initialization . It looks something like this:

sudo kubeadm join <master-ip>:6443 --token <token> --discovery-token-ca-cert-hash sha256:<hash>



Step 8: Verify the Cluster

Once the worker node joins, check the status on the **master node**



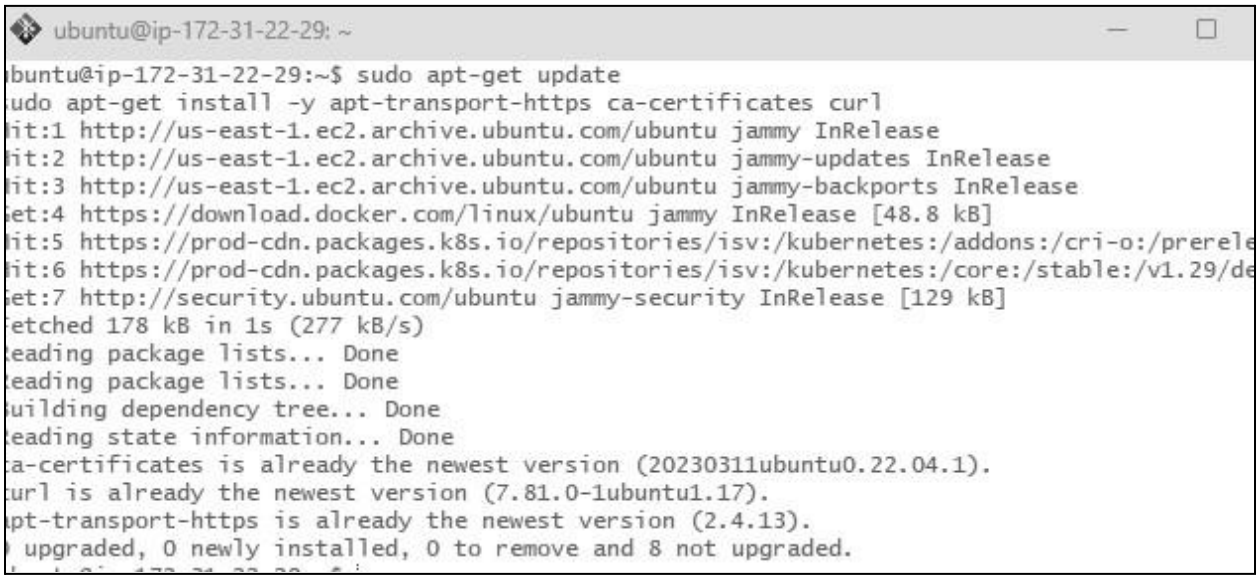
ADVANCE DEVOPS EXPERIMENT 4

Prajjwal Pandey – D15A/33

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

1. **Install prerequisites:** sudo apt-get update sudo apt-get install -y apt-transport-

   https ca-certificates curl

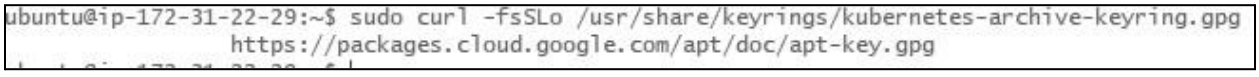


2. **Add the GPG key for Kubernetes:**

   sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg

3. **Add the Kubernetes repository:**

   echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-focal main" | sudo tee /etc/apt/sources.list.d/kubernetes.list



```
ubuntu@ip-172-31-22-29:~$  echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring
.gpg]  https://apt.kubernetes.io/ kubernetes-focal main" | sudo tee /etc/apt/sources.list.d/ku
bernetes.list
deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg]  https://apt.kubernetes.io/
 kubernetes-focal main
```

1.2 Install kubectl

sudo apt-get update

sudo apt-get install -y kubectl

```
ubuntu@ip-172-31-22-29:~$ sudo apt-get update
sudo apt-get install -y kubectl
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:5 https://download.docker.com/linux/ubuntu jammy InRelease
Hit:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/addons:/cri-o:/prerelease:/main/deb  InRelease
Ign:7 https://packages.cloud.google.com/apt kubernetes-focal InRelease
Err:8 https://packages.cloud.google.com/apt kubernetes-focal Release
  404  Not Found [IP: 172.253.62.138 443]
Reading package lists... Done
E: The repository 'https://apt.kubernetes.io kubernetes-focal Release' does not have a Release file.
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
kubectl is already the newest version (1.29.0-1.1).
0 upgraded, 0 newly installed, 0 to remove and 5 not upgraded.
```

Step 2: Deploying Your Application on Kubernetes

2.1 Set up Kubernetes Cluster

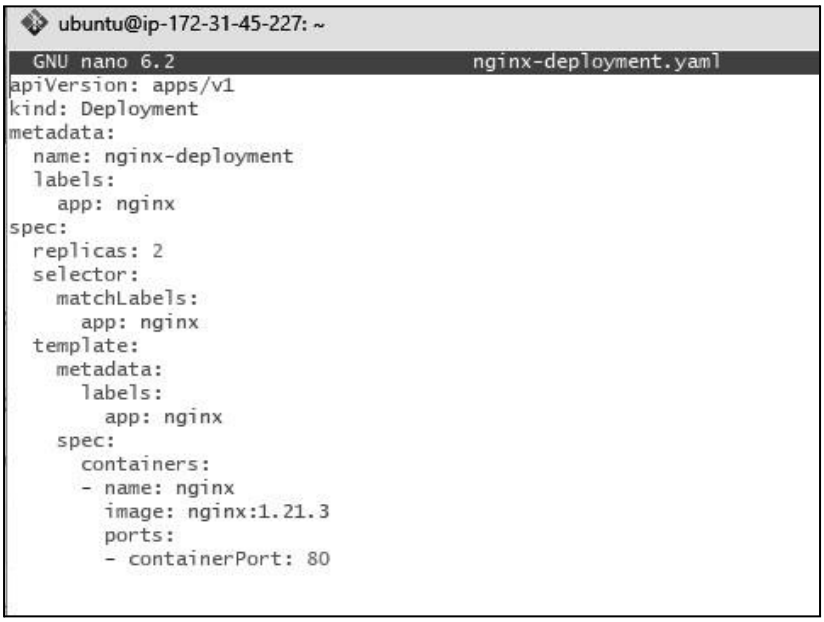
kubectl get nodes

```
ubuntu@ip-172-31-45-227:~$ kubectl get nodes
NAME               STATUS   ROLES           AGE     VERSION
ip-172-31-43-211   Ready    <none>          50s     v1.29.0
ip-172-31-45-13    Ready    <none>          34s     v1.29.0
ip-172-31-45-227   Ready    control-plane   5m17s   v1.29.0
ubuntu@ip-172-31-45-227:~$
```

Step 3:Create the Deployment YAML file

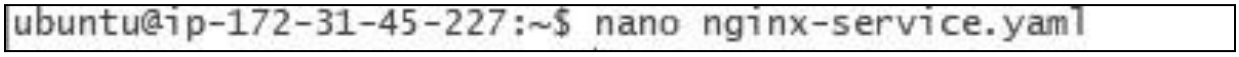
a)Create the YAML file: Use a text editor to create a file named nginx-deployment.yaml

b) Add the Deployment Configuration: Copy and paste the following YAML content into the file. Save and exit the editor (Press Ctrl+X, then Y, and Enter).

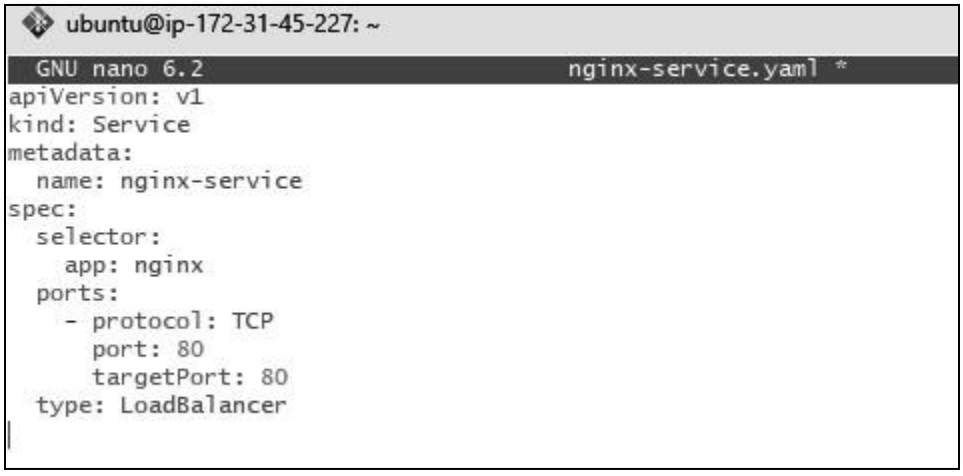


Step 4:Create the Service YAML File

a)Create the YAML File: Create another file named nginx-service.yaml



b)Add the Service Configuration: Copy and paste the following YAML content into the file given below.



Step 5:Apply the YAML Files

a)Deploy the Application: Use kubectl to create the Deployment and Service from the YAML files.

```
ubuntu@ip-172-31-45-227:~$ kubectl apply -f nginx-deployment.yaml
kubectl apply -f nginx-service.yaml
deployment.apps/nginx-deployment created
service/nginx-service created
```

b)Verify the Deployment: Check the status of your Deployment,Pods and Services.

```
ubuntu@ip-172-31-45-227:~$ kubectl get deployments
kubectl get pods
kubectl get services
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   2/2     2            2           40s
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-6b4d6fdbf-6k84m   1/1     Running   0          40s
nginx-deployment-6b4d6fdbf-9d8j6   1/1     Running   0          40s
NAME            TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
kubernetes      ClusterIP      10.96.0.1        <none>        443/TCP        40m
nginx-service   LoadBalancer   10.106.182.152   <pending>     80:32317/TCP   40s
```

Describe the deployment(Extra)

```
ubuntu@ip-172-31-45-227:~$ kubectl get deployments
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   1/1     1            1           14h
ubuntu@ip-172-31-45-227:~$ kubectl describe deployment
Name:                   nginx-deployment
Namespace:              default
CreationTimestamp:      Wed, 11 Sep 2024 17:16:17 +0000
Labels:                 <none>
Annotations:            deployment.kubernetes.io/revision: 2
Selector:               app=nginx
Replicas:               1 desired | 1 updated | 1 total | 1 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=nginx
  Containers:
   nginx:
    Image:        nginx:latest
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:
      /usr/share/nginx/html from website-volume (rw)
  Volumes:
   website-volume:
    Type:      ConfigMap (a volume populated by a ConfigMap)
    Name:      nginx-website
    Optional:  false
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable
OldReplicaSets:  nginx-deployment-6b4d6fdbf (0/0 replicas created)
NewReplicaSet:   nginx-deployment-776b8fd845 (1/1 replicas created)
Events:          <none>
```

Step 6:Ensure Service is Running

kubectl get service

```
ubuntu@ip-172-31-45-227:~$ kubectl get service
NAME            TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
kubernetes      ClusterIP   10.96.0.1        <none>        443/TCP        16h
nginx           NodePort    10.106.0.176     <none>        80:32618/TCP   76m
nginx-service   NodePort    10.106.182.152   <none>        80:30007/TCP   15h
nginx2          NodePort    10.99.32.156     <none>        80:31421/TCP   8s
```

Step 7:Forward the Service Port to Your Local Machine

kubectl port-forward allows you to forward a port from your local machine to a port on a service running in the Kubernetes cluster.

1. **Forward the Service Port**: Use the following command to forward a local port to the service's target port. kubectl port-forward service/<service-name> <local-port>:<service-port>

```
ubuntu@ip-172-31-45-227:~$ kubectl port-forward service/nginx-service 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
```

This command will forward local port 8080 on your machine to port 80 of the service nginx-service running inside the cluster.

2. This means port forwarding is now active, and any traffic to localhost:8080 will be routed to the nginx-service on port 80.

```
ubuntu@ip-172-31-45-227:~$ kubectl port-forward service/nginx-service 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
^Cubuntu@ip-172-31-45-227:~kubectl port-forward service/nginx-service 8081:8080
Forwarding from 127.0.0.1:8081 -> 80
Forwarding from [::1]:8081 -> 80
^Cubuntu@ip-172-31-45-227:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-776b8fd845-k9cx4   1/1     Running   0          113m
ubuntu@ip-172-31-45-227:~$ kubectl logs nginx-deployment-776b8fd845-k9cx4
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2024/09/12 06:35:51 [notice] 1#1: using the "epoll" event method
2024/09/12 06:35:51 [notice] 1#1: nginx/1.27.1
2024/09/12 06:35:51 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2024/09/12 06:35:51 [notice] 1#1: OS: Linux 6.5.0-1022-aws
2024/09/12 06:35:51 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2024/09/12 06:35:51 [notice] 1#1: start worker processes
2024/09/12 06:35:51 [notice] 1#1: start worker process 24
2024/09/12 06:35:51 [notice] 1#1: start worker process 25
```

Step 8: Access the Application Locally

1. **Open a Web Browser**: Now open your web browser and go to the following URL:

   http://localhost:8080

You should see the application (in this case, Nginx) that you have deployed running in the Kubernetes cluster, served locally via port 8080.

In case the port 8080 is unavailable, try using a different port like 8081


Welcome to nginx!
192.46.222.221:31672
Welcome to nginx!
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.
Thank you for using nginx.

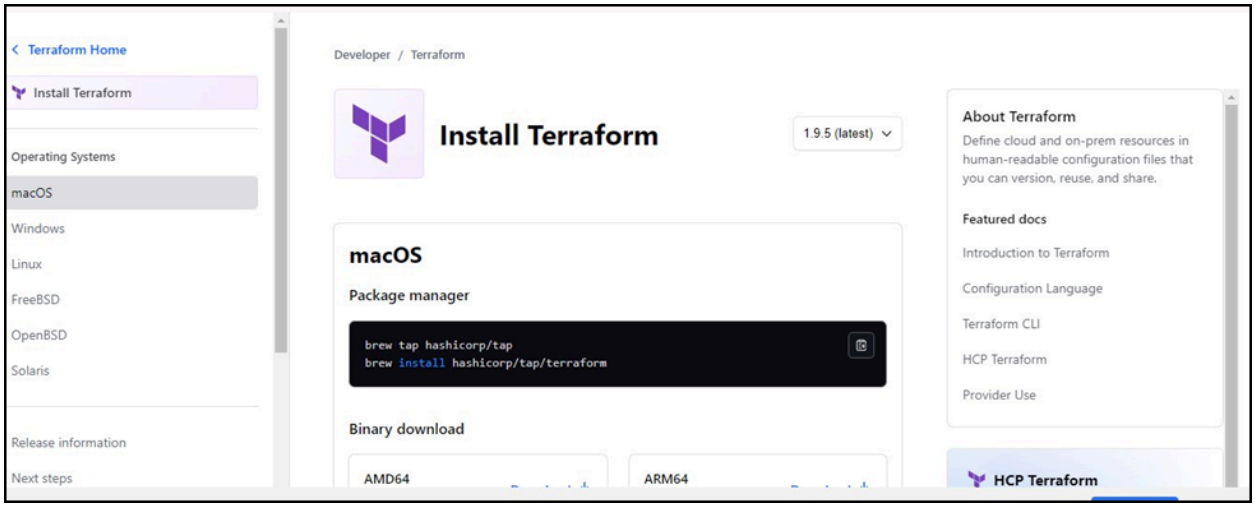
ADVANCE DEVOPS EXP 5

Name:Prajjwal Pandey
Class:D15A
Roll No:33

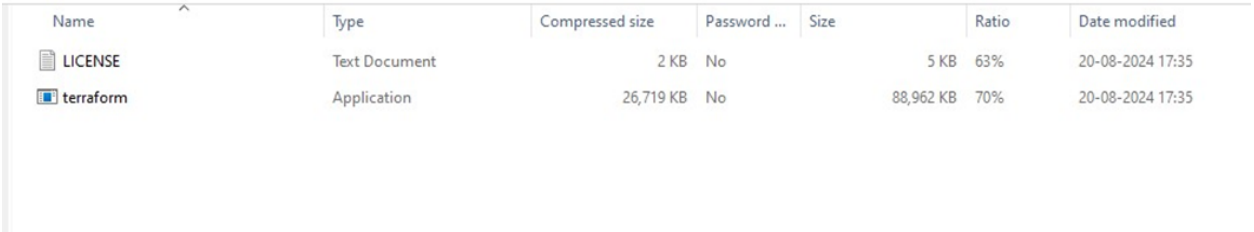
Aim:To understand terraform lifecycle, core concepts/terminologies and install it on a Linux Machine and Windows.

Installation for Windows:

Step 1:Go to the website terraform.io and install Terraform from there..Select the AMD64 option for Windows and download Terraform.

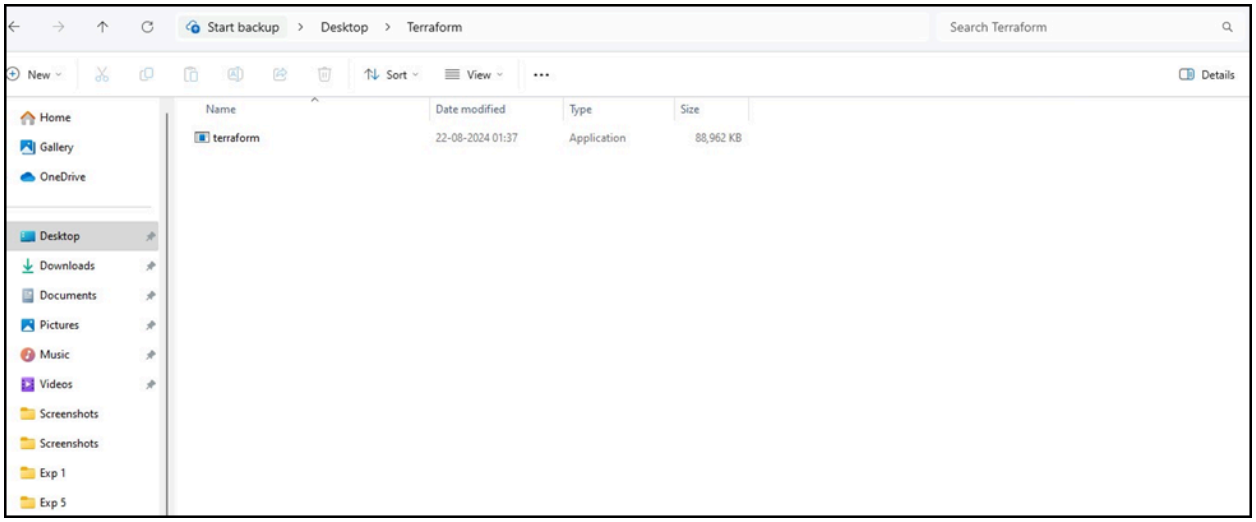


Step 2: Go to the zip file where Terraform is installed.

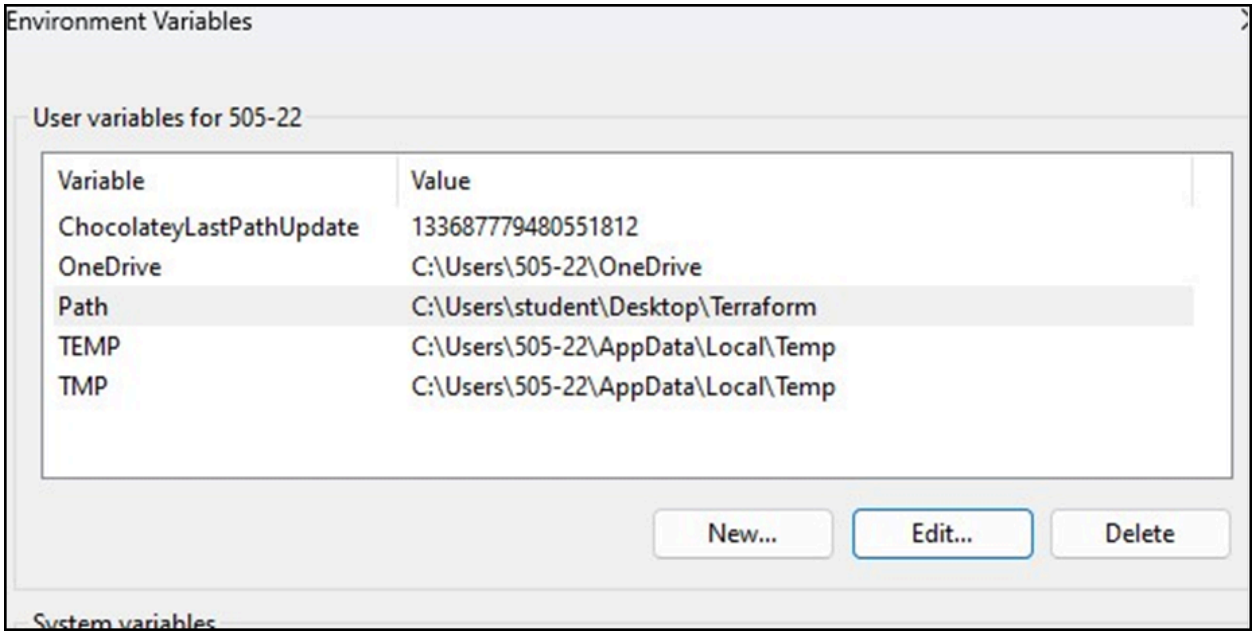


Name	Type	Compressed size	Password ...	Size	Ratio	Date modified
LICENSE	Text Document	2 KB	No	5 KB	63%	20-08-2024 17:35
terraform	Application	26,719 KB	No	88,962 KB	70%	20-08-2024 17:35

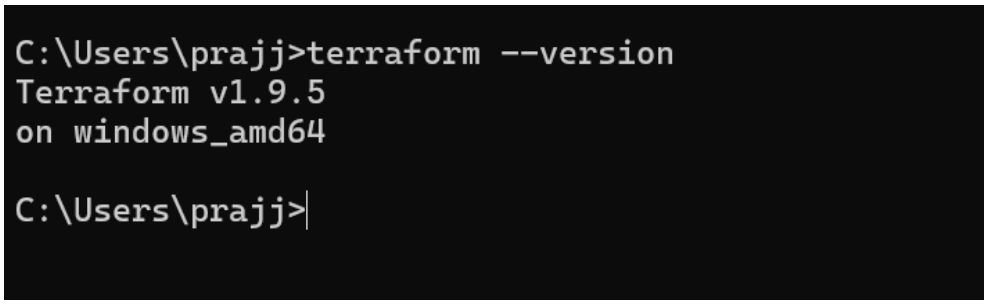
Step 3: Since the installed file is a zip file,create a new folder on desktop and copy the installed terraform application there.



Step 4: Now go to search bar,select edit environment variables option,then go to the path option.Now add the file path of the directory wherein we have installed the terraform application.



Step 5: Now go to the folder where we have installed terraform and open Powershell inside it.After this type ‘terraform’ to make sure that terraform has been installed on the system. The command ‘terraform –version simply checks the current version of terraform that has been installed.



ADVANCE DEVOPS EXP6

Name:Prajjwal Pandey
Class:D15A
Roll No:33

Aim:To Build, change, and destroy AWS / GCP /Microsoft Azure/ DigitalOcean infrastructure Using Terraform.
(S3 bucket or Docker) fdp.

Part A:Creating docker image using terraform

Prerequisite:
1) Download and Install Docker Desktop from https://www.docker.com/

Step 1: Check Docker functionality, Check for the docker version with the following command.

```
Microsoft Windows [Version 10.0.22631.3880]
(c) Microsoft Corporation. All rights reserved.

C:\Users\prajj>docker --version
Docker version 27.0.3, build 7d4bcd8

C:\Users\prajj>
```

Now, create a folder named 'Terraform Scripts' in which we save our different types of scripts which will be further used in this experiment.

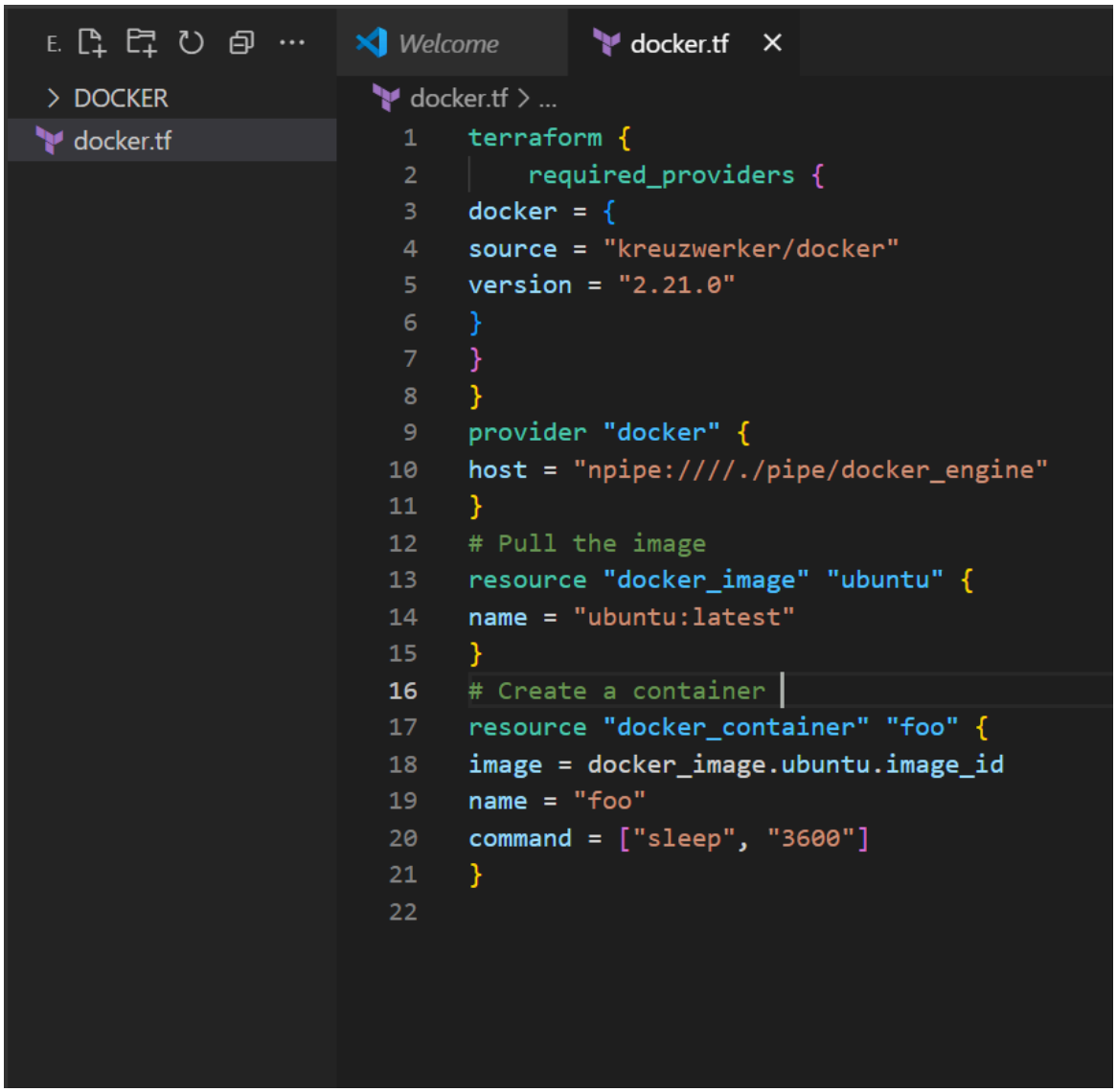
Step 2: Firstly create a new folder named 'Docker' in the 'TerraformScripts' folder. Then create a new docker.tf file using Atom editor and write the followingcontents into it to create a Ubuntu Linux container. Script:

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

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

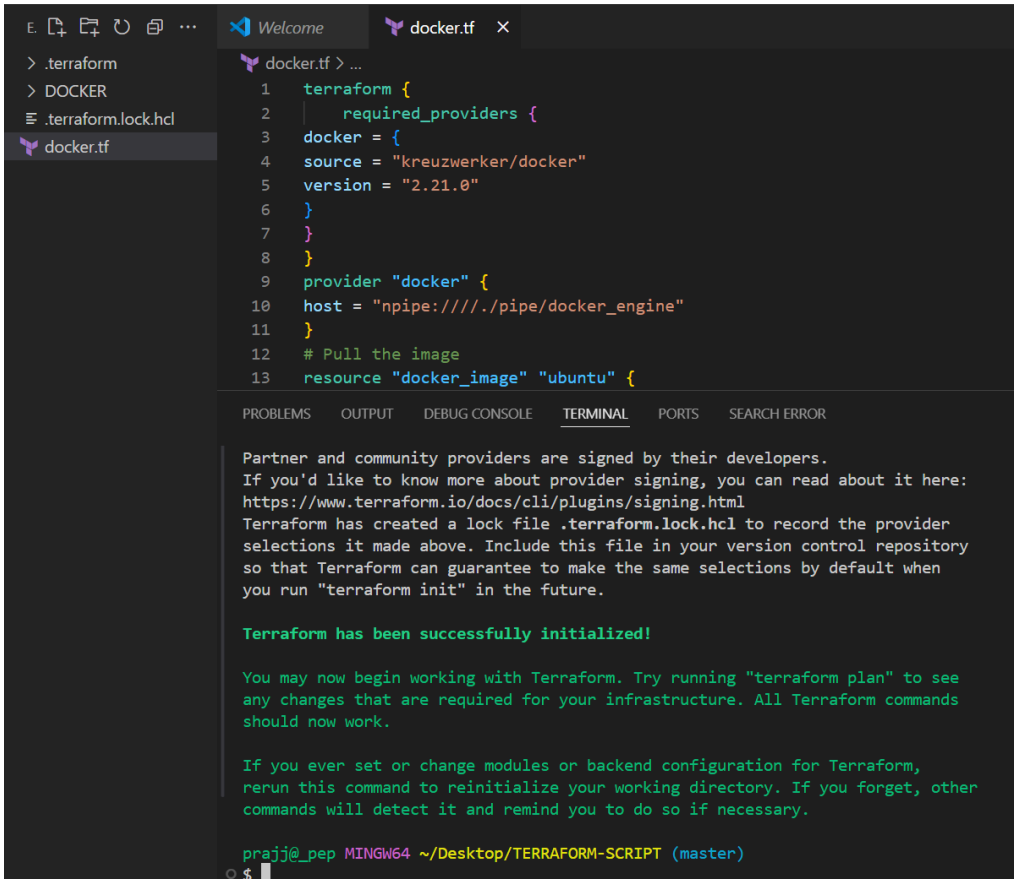
}

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

# Create a container resource
"docker_container" "foo" {
  image = docker_image.ubuntu.image_id
  name = "foo" command = ["sleep", "3600"]
}



Step 3: Execute Terraform Init command to initialize the resources



Step 4: Execute Terraform plan to see the available resources

```
PS C:\Users\Admin\TerraformScripts\Docker> terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the fol
  + 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",
          + "3600",
        ]
      + container_logs   = (known after apply)
      + entrypoint       = (known after apply)
      + env              = (known after apply)
      + exit_code        = (known after apply)
      + gateway          = (known after apply)
      + hostname         = (known after apply)
      + id               = (known after apply)
      + image            = (known after apply)
      + init             = (known after apply)
      + ip_address       = (known after apply)
      + ip_prefix_length = (known after apply)
      + ipc_mode         = (known after apply)
      + log_driver       = (known after apply)
      + logs             = false
      + must_run         = true
      + name             = "foo"
      + network_data     = (known after apply)
      + read_only        = false
      + remove_volumes   = true
      + restart          = "no"
      + rm               = false
```

```
      + runtime          = (known after apply)
      + security_opts    = (known after apply)
      + shm_size         = (known after apply)
      + start            = true
      + stdin_open       = false
      + stop_signal      = (known after apply)
      + stop_timeout     = (known after apply)
      + tty              = false

      + healthcheck (known after apply)

      + labels (known after apply)
    }

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

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

Step 5: Execute Terraform apply to apply the configuration, which will automatically create and run the Ubuntu Linux container based on our configuration. Using command :

```
docker_image.ubuntu: Creating...
docker_image.ubuntu: Creation complete after 9s [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Creating...
docker_container.foo: Creation complete after 2s [id=01adf07e5918931fee9b90073726a03671037923dd92032ce0e15bbb764a6f24]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
```

Docker images, Before Executing Apply step:

```
PS C:\Users\Admin\TerraformScripts\Docker> docker images
REPOSITORY                          TAG       IMAGE ID       CREATED         SIZE
```

Docker images, After Executing Apply step:

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

Step 6: Execute Terraform destroy to delete the configuration, which will automatically delete the Ubuntu Container.

```
  # 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=01adf07e5918931fee9b90073726a03671037923dd92032ce0e15bbb764a6f24]
docker_container.foo: Destruction complete after 0s
docker_image.ubuntu: Destroying... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_image.ubuntu: Destruction complete after 1s

Destroy complete! Resources: 2 destroyed.
```

Docker images After Executing Destroy step

```
PS C:\Users\Admin\TerraformScripts\Docker> docker images
REPOSITORY                          TAG       IMAGE ID       CREATED         SIZE
```

ADVANCE DEVOPS EXP 7

Prajjwal Pandey - D15A - 33

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

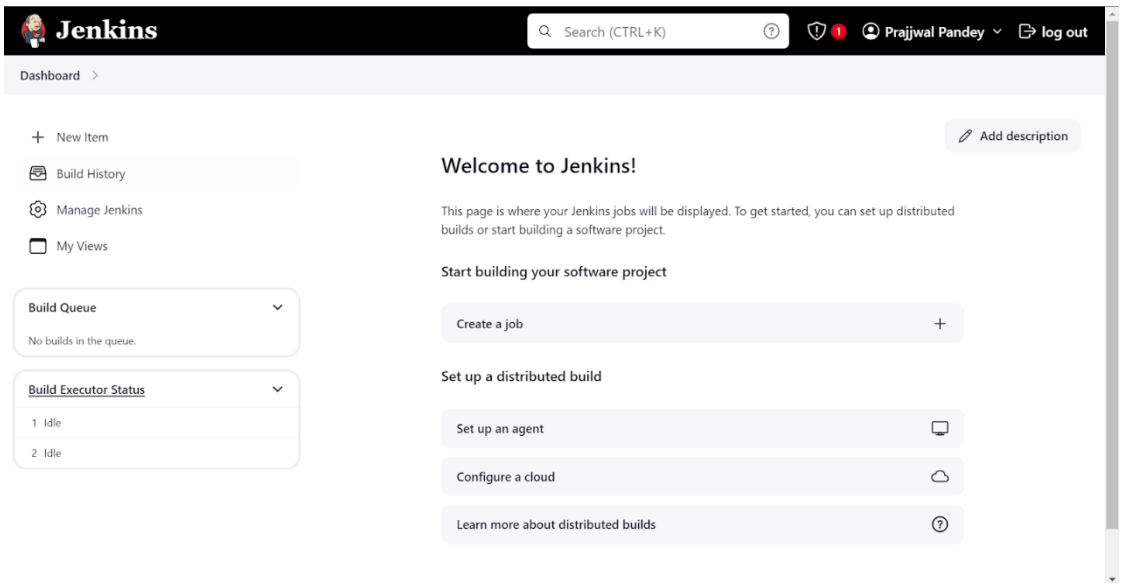
Integrating Jenkins with SonarQube:

- Jenkins installed
- Docker Installed (for SonarQube)

- SonarQube Docker Image

Steps to integrate Jenkins with SonarQube

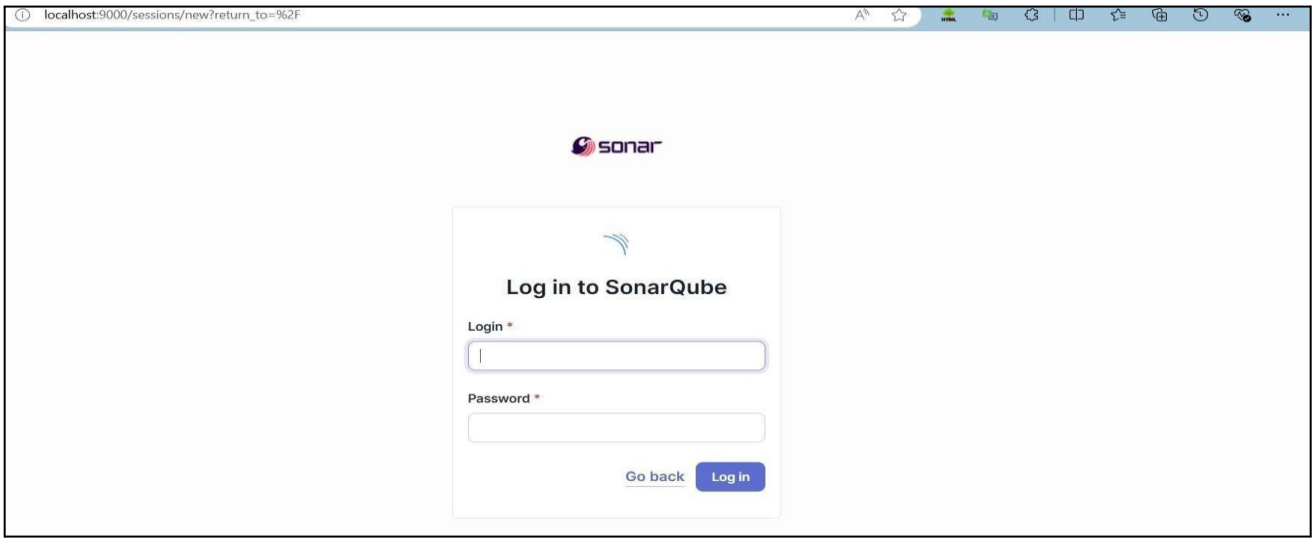
1.Open up Jenkins Dashboard on localhost, port 8080 or whichever port it is at for you.



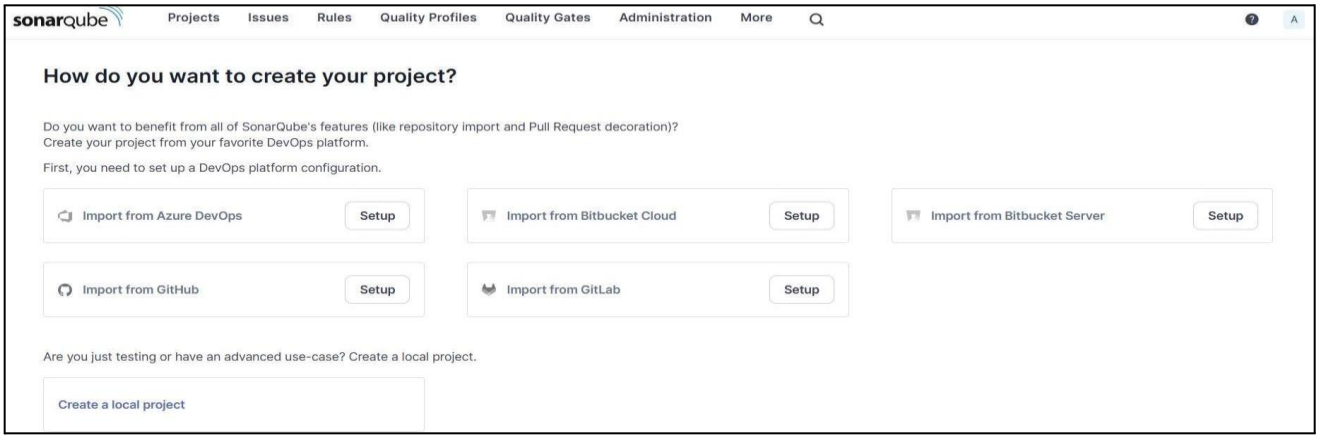
2.Run SonarQube in a Docker container using this command **docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest**

```
C:\Windows\System32>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:late
st
Unable to find image 'sonarqube:latest' locally
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
5bf67120a14329b758ad55b8a8e7495495f619936e93dd5aefd3e624c65e43b6

C:\Windows\System32>
```

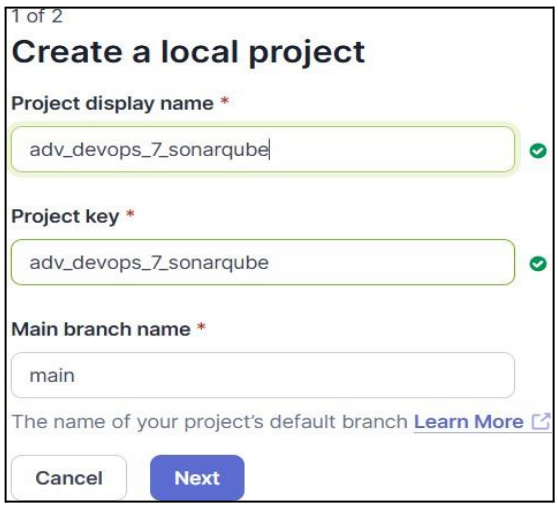
3. Once the container is up and running, you can check the status of SonarQube at localhost port 9000.

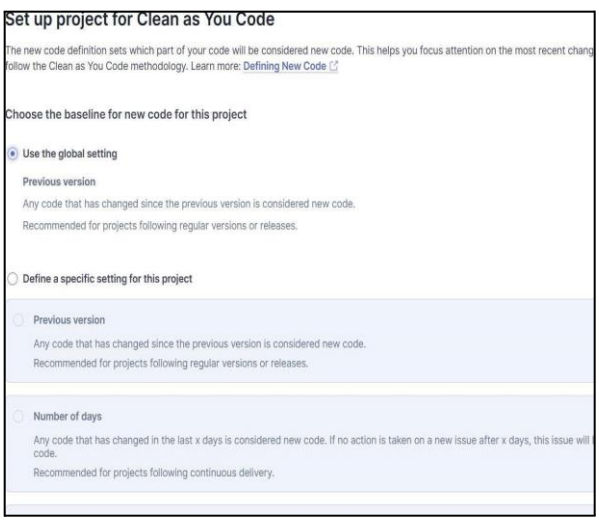


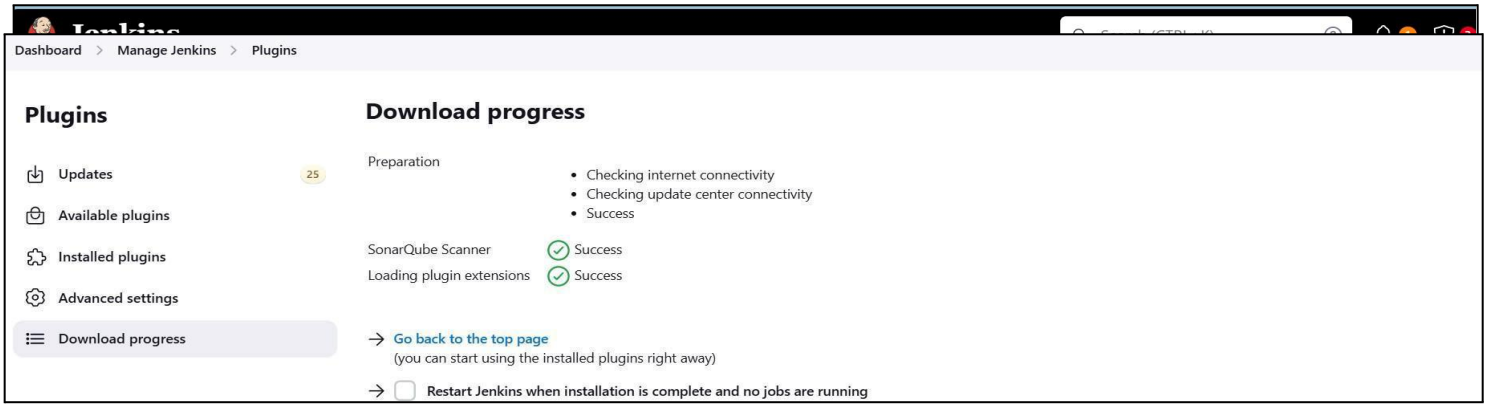
4.Login to SonarQube using username admin and password admin.



5.Create a manual project in SonarQube with the name sonarqube
Setup the project and come back to Jenkins Dashboard.
Go to Manage Jenkins and search for SonarQube Scanner for Jenkins and install it.







6.Under Jenkins 'Manage Jenkins' then go to 'system', scroll and look for **SonarQube Servers** and enter
the details.
Enter the Server Authentication token if needed.

In SonarQube installations: Under **Name** add <project name of sonarqube>,here we have named it as **adv_devops_7_sonarqube**
In **Server URL** Default is **http://localhost:9000**

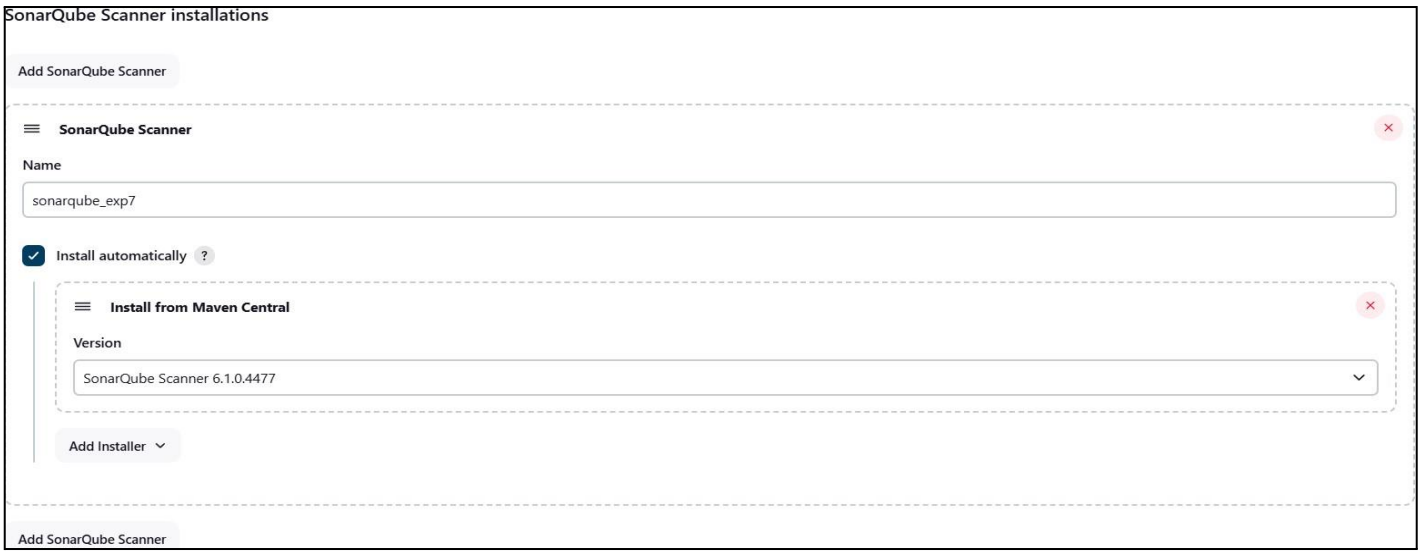
7. Search for SonarQube Scanner under Global Tool Configuration.

Choose the latest configuration and choose Install automatically.

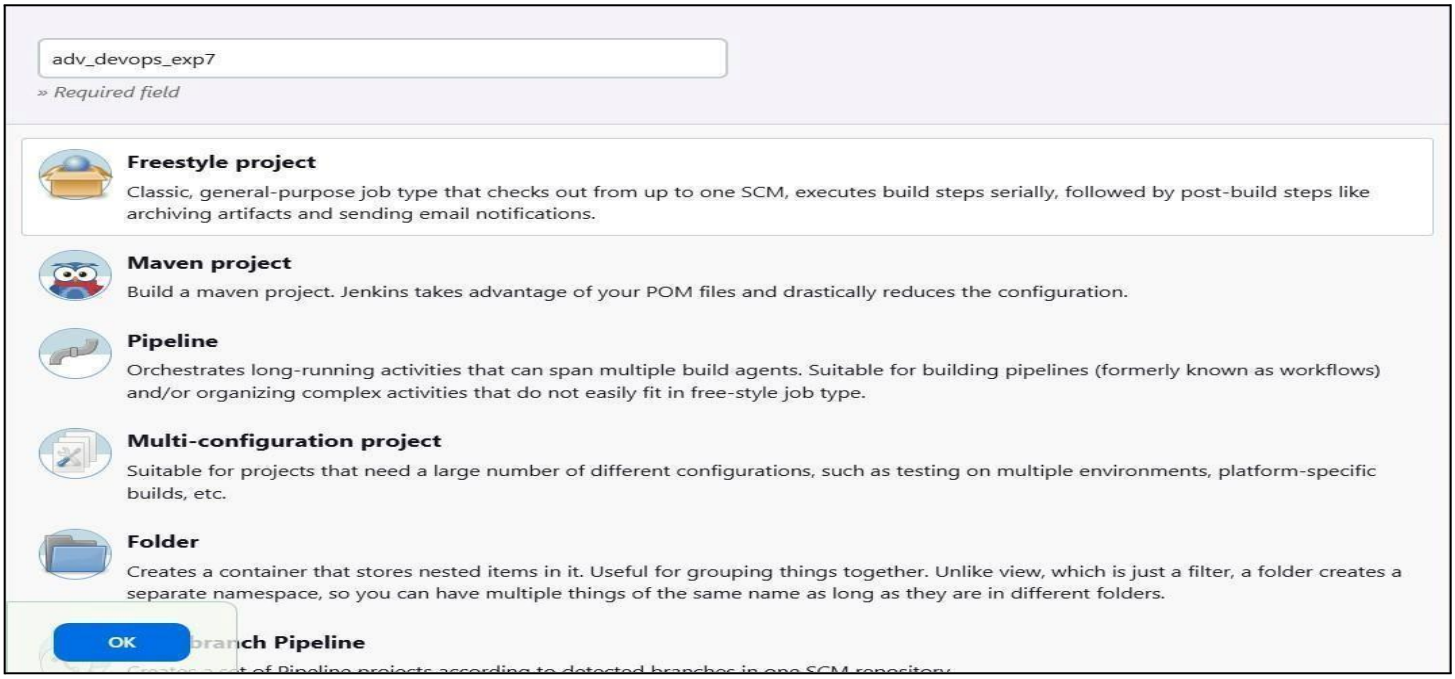
Dashboard > Manage Jenkins > Tools

Check the “Install automatically” option. → Under name any name as identifier →

Check the “Install automatically” option.

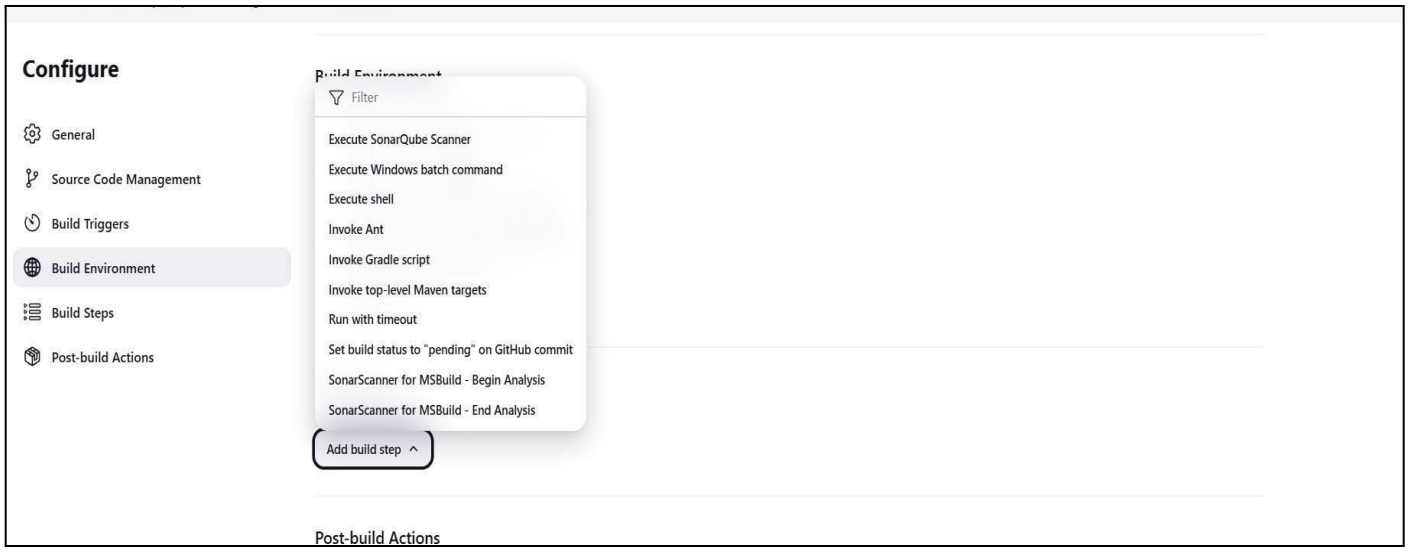


8.After the configuration, create a New Item in Jenkins, choose a freestyle project.

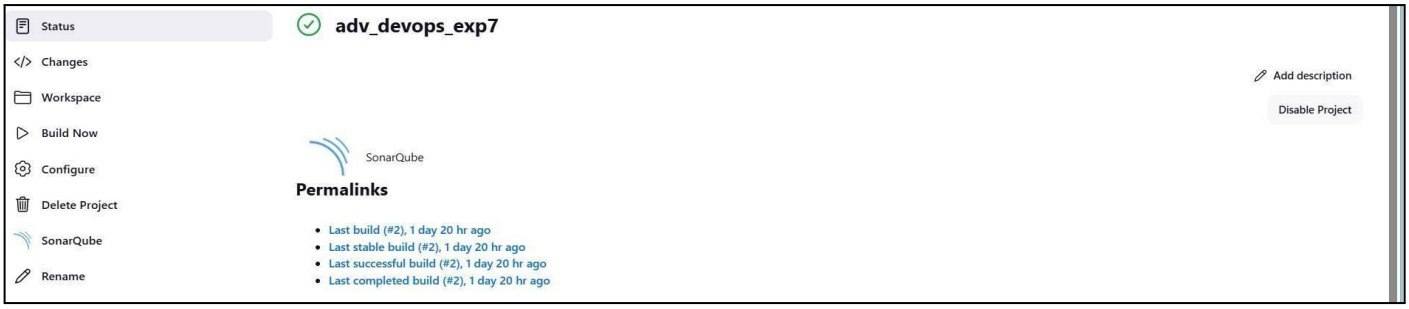

adv_devops_exp7
» Required field
Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
OK

9. Choose this GitHub repository in Source Code Management.
https://github.com/shazforiot/MSBuild_firstproject.git
It is a sample hello-world project with no vulnerabilities and issues, just to test the integration.

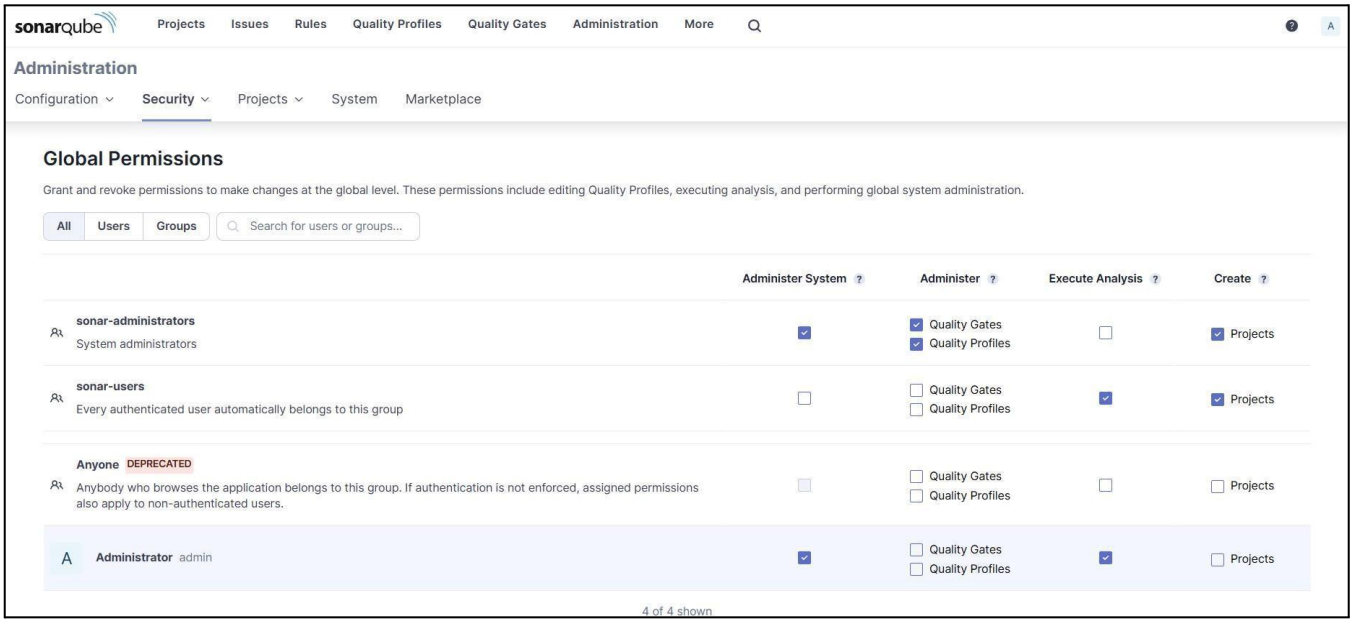
10. Under **Select project → Configuration → Build steps → Execute SonarQube Scanner**, enter these Analysis properties. Mention the SonarQube Project Key, Login, Password, Source path and Host URL.


Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Filter
Execute SonarQube Scanner
Execute Windows batch command
Execute shell
Invoke Ant
Invoke Gradle script
Invoke top-level Maven targets
Run with timeout
Set build status to "pending" on GitHub commit
SonarScanner for MSBuild - Begin Analysis
SonarScanner for MSBuild - End Analysis
Add build step
Post-build Actions

Then save



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



IF CONSOLE OUTPUT FAILED:

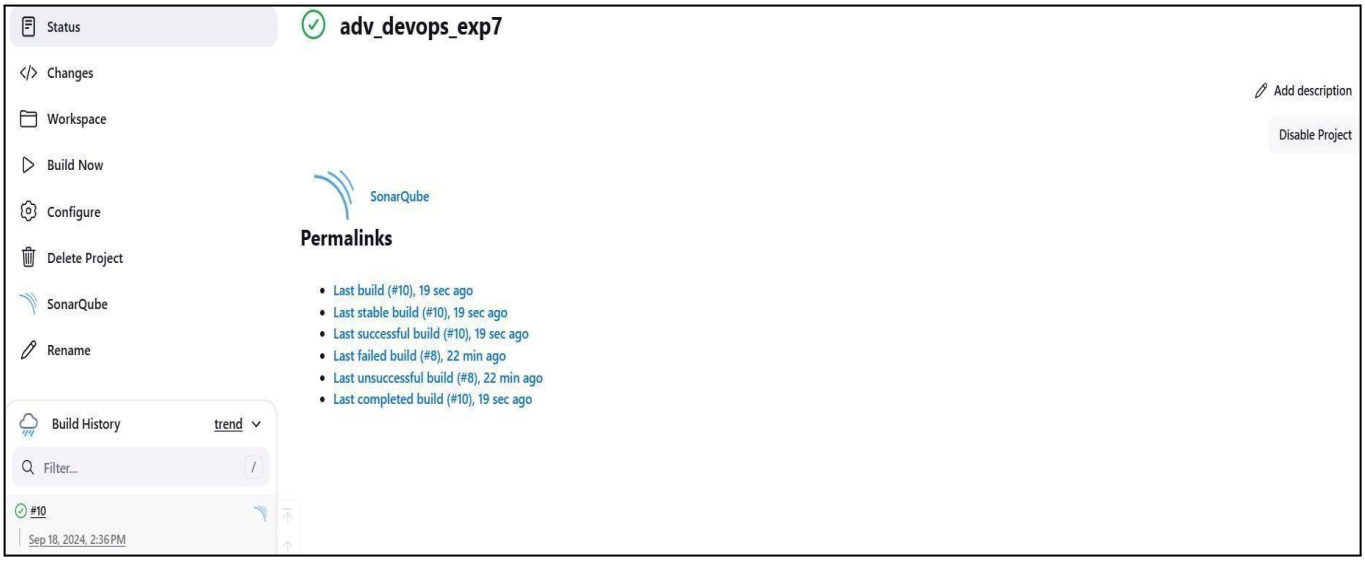
Step 1: Generate a New Authentication Token in SonarQube

1. **Login to SonarQube:**
   - Open your browser and go to **http://localhost:9000**.
   - Log in with your admin credentials (default username is **admin**, and the password is either **admin** or your custom password if it was changed).
2. **Generate a New Token:**
   - Click on your **username** in the top-right corner of the SonarQube dashboard.
   - Select **My Account** from the dropdown menu.
   - Go to the **Security** tab.
   - Under **Generate Tokens**, type a name for the token (e.g., "Jenkins-SonarQube").
   - Click **Generate**.
   - Copy the token and save it securely. You will need it in Jenkins.

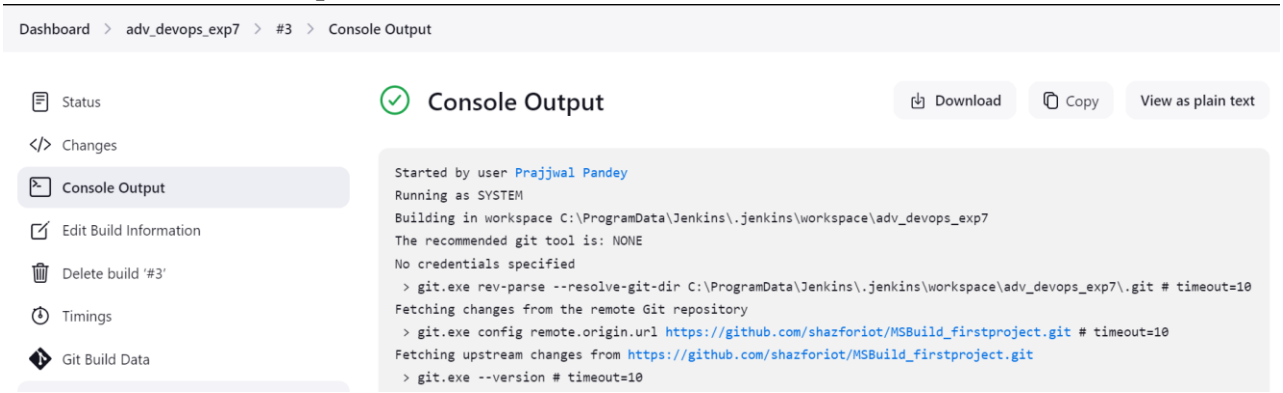
Step 2: Update the Token in Jenkins

1. **Go to Jenkins Dashboard:**
   - Open Jenkins and log in with your credentials.
2. **Configure the Jenkins Job:**
   - Go to the job that is running the SonarQube scanner (adv_devops_exp7). ◦ Click **Configure**.
3. **Update the SonarQube Token:**
   - In the SonarQube analysis configuration (either in the pipeline script or under "Build" section, depending on your job type), update the sonar.login parameter with the new token.

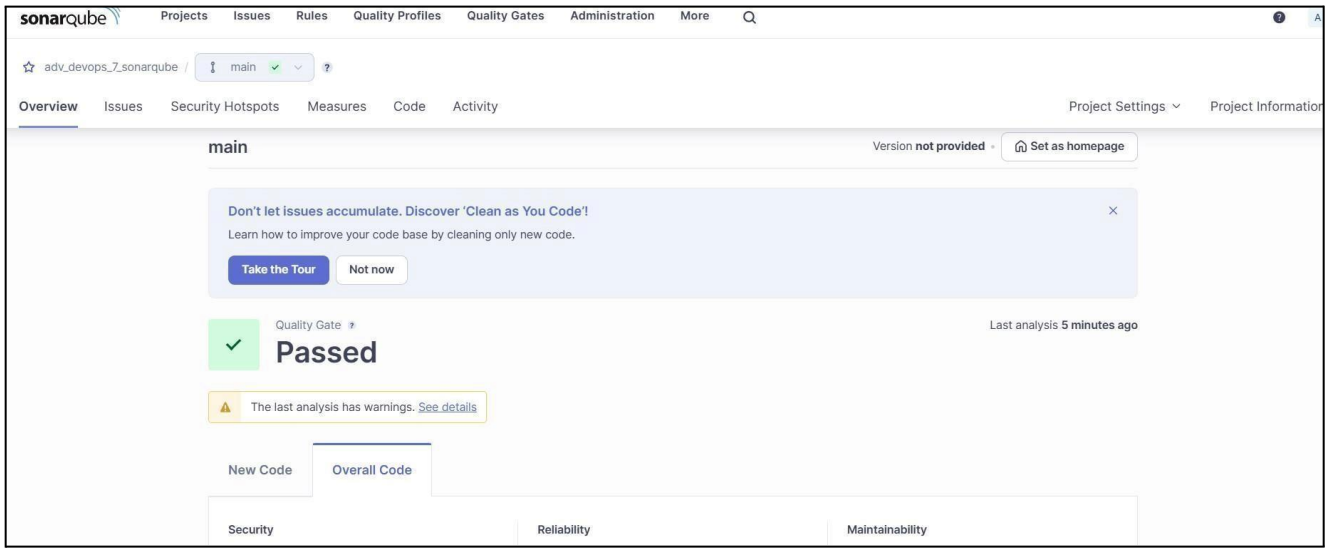
12.Run the Jenkins build.



Check the console Output



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

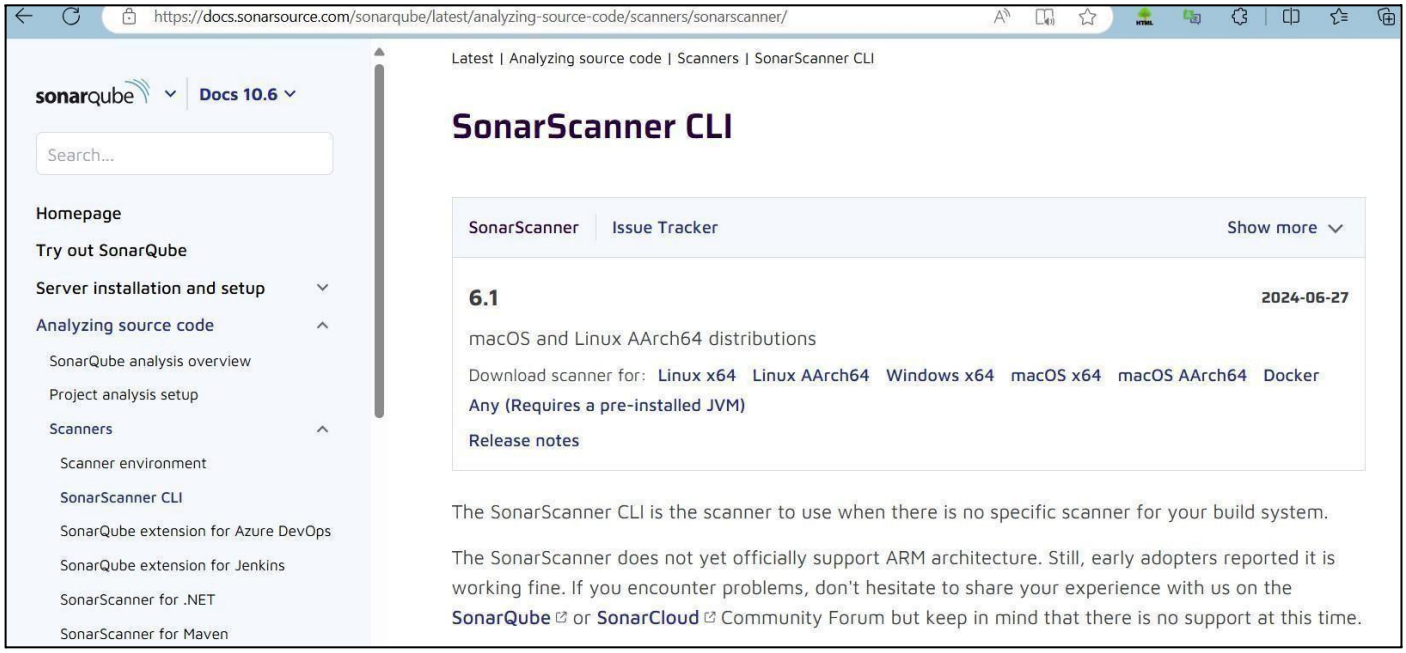


ADVANCE DEVOPS EXP 8

Prajjwal Pandey – D15A / 33

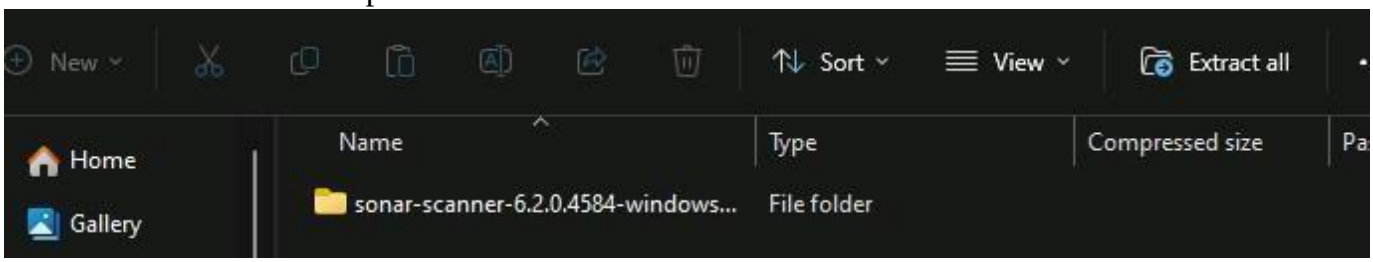
Aim: Create a Jenkins CICD Pipeline with SonarQube / GitLab Integration to perform a static analysis of the code to detect bugs, code smells, and security vulnerabilities on a sample Web / Java / Python application.

Step 1: Download sonar scanner https://docs.sonarsource.com/sonarqube/latest/analyzing-source-code/scanners/sonarscan



ner/ Visit this link and download the sonarqube scanner CLI.

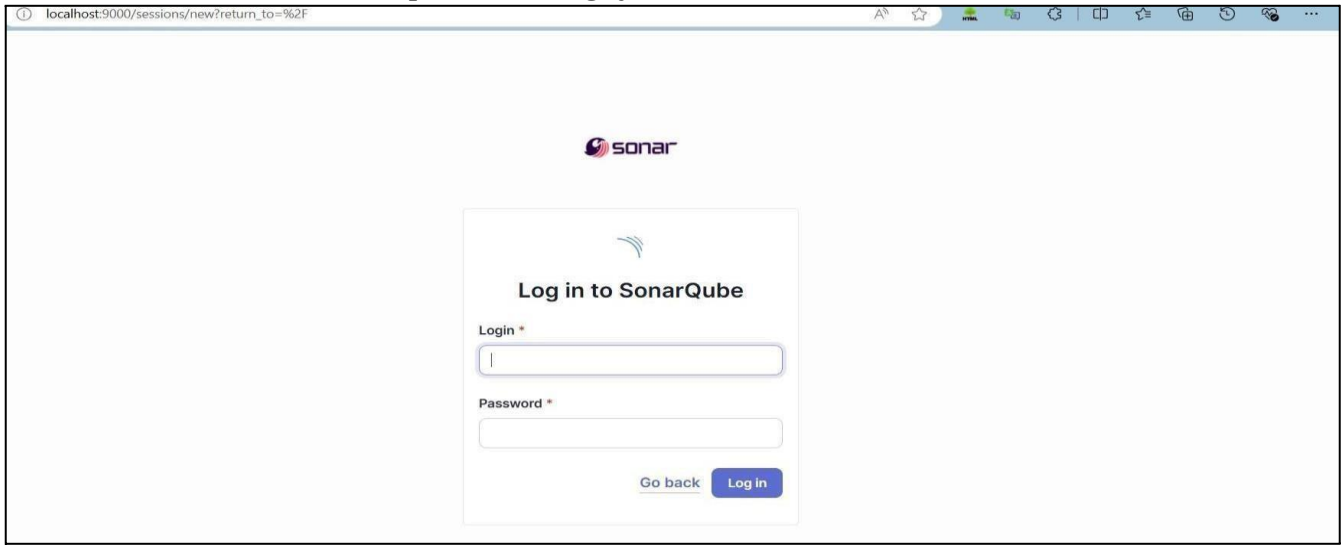
Extract the downloaded zip file in a folder.



1. Install sonarqube image Command: **docker pull sonarqube**

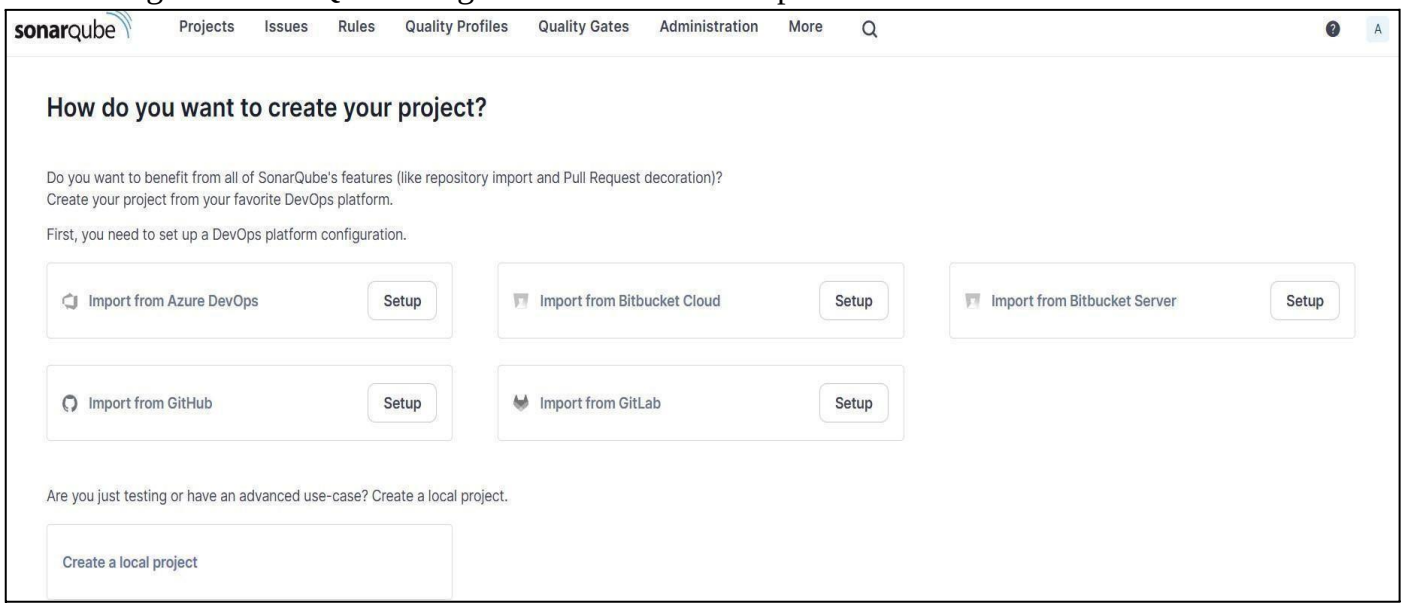
```
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindo

PS C:\Users\Soham Satpute> docker pull sonarqube
Using default tag: latest
latest: Pulling from library/sonarqube
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Image is up to date for sonarqube:latest
docker.io/library/sonarqube:latest
```

2. Once the container is up and running, you can check the status of

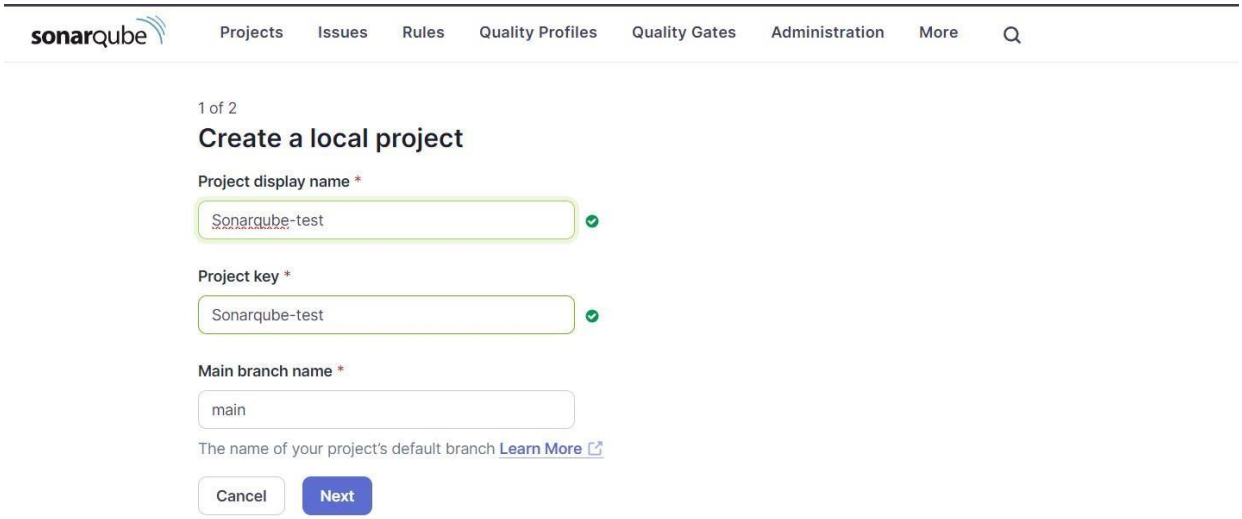


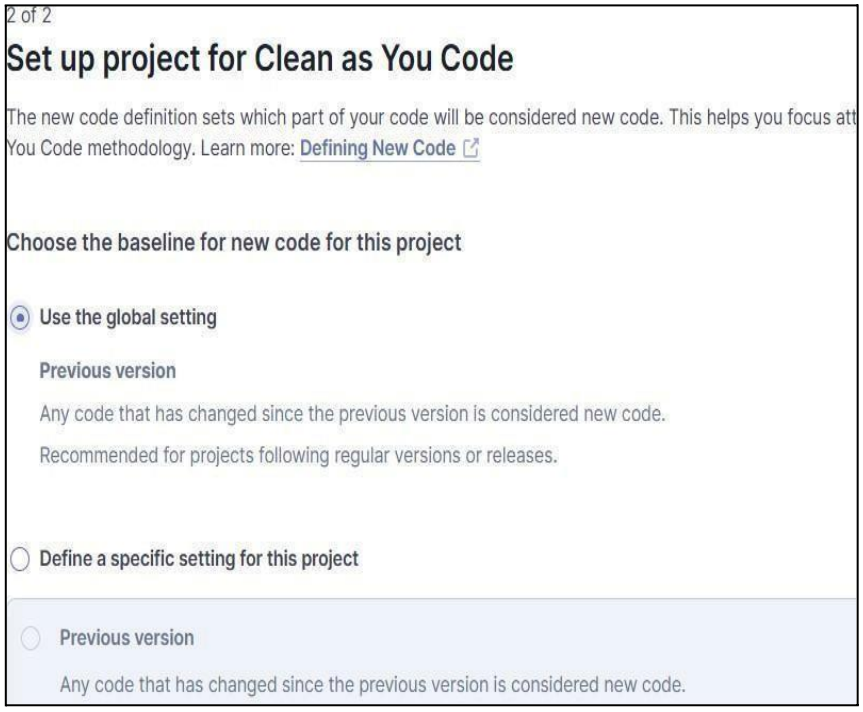
SonarQube at localhost port 9000.

3. Login to SonarQube using username admin and password admin.

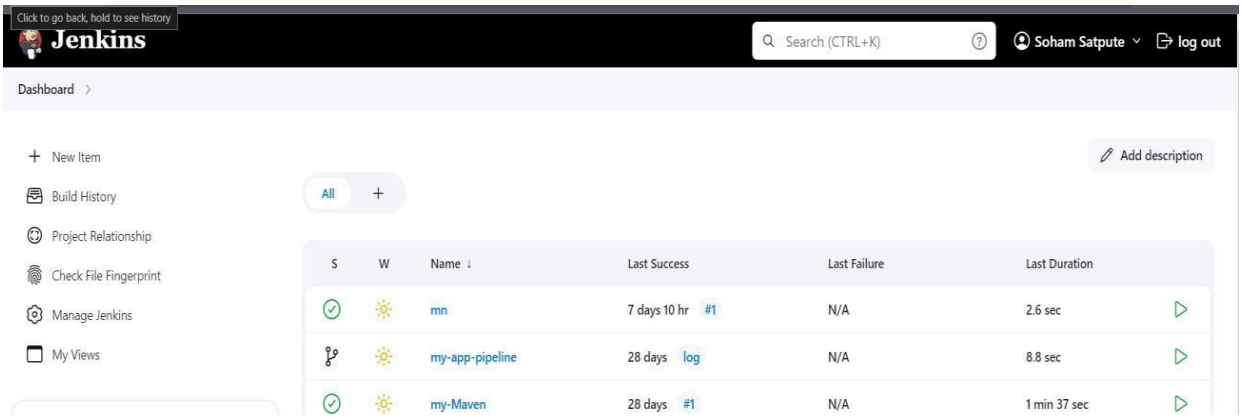


4. Create a manual project in SonarQube with the name sonarqube

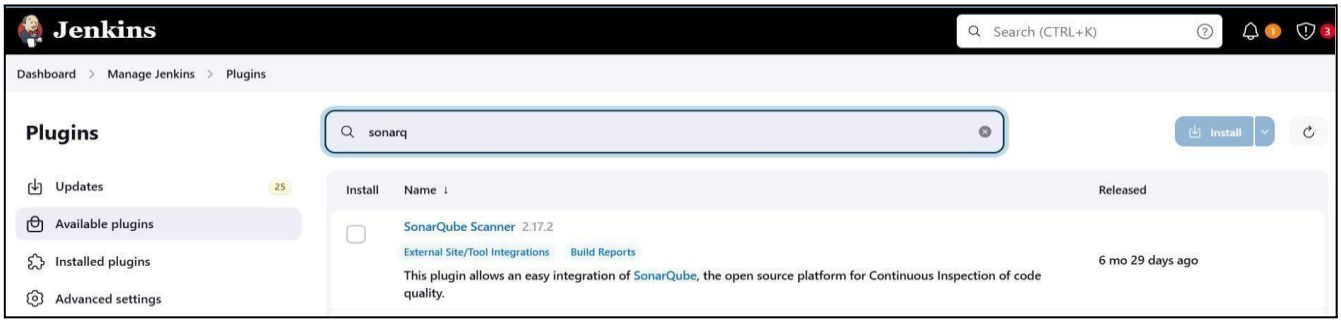


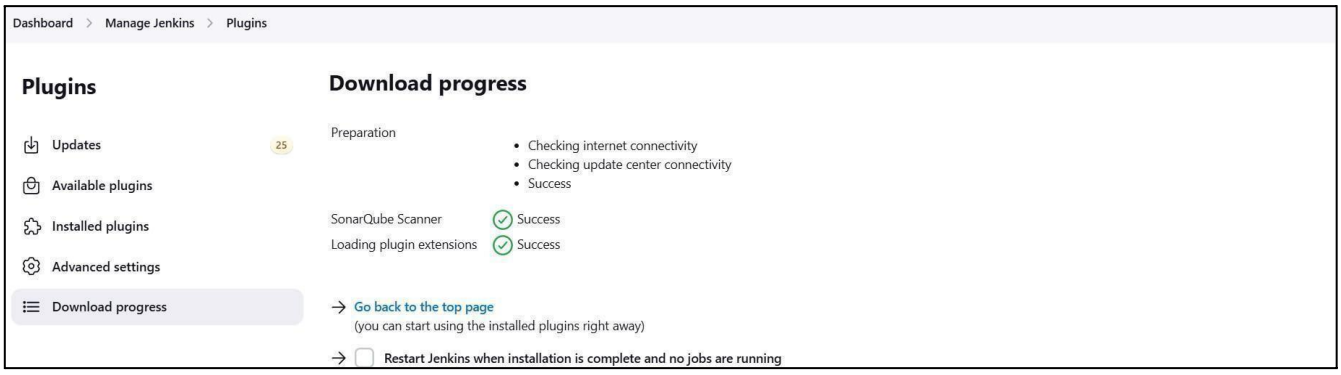


5. Open up Jenkins Dashboard on localhost, port 8080 or whichever port it is at for you.



6. Go to Manage Jenkins and search for SonarQube Scanner for Jenkins and install it.





7. Under Jenkins ‘Manage Jenkins’ then go to ‘system’, scroll and look for **SonarQube Servers** and enter
the details.

Enter the Server Authentication token if needed.

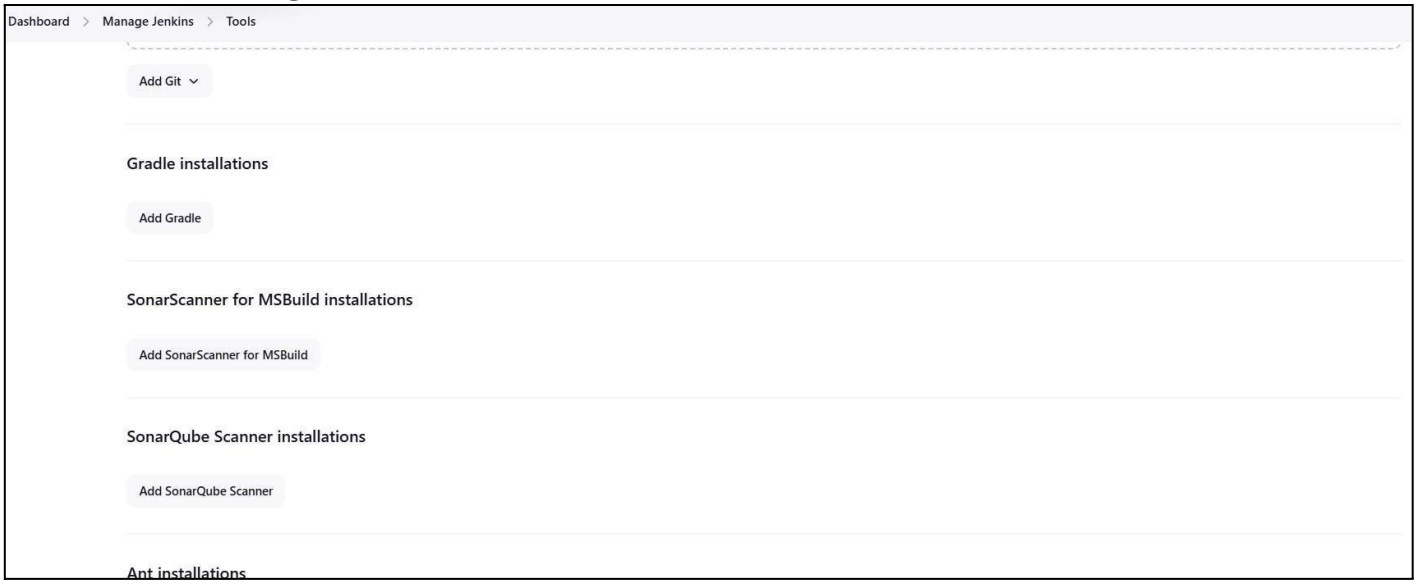
In SonarQube installations: Under **Name** add <project name of sonarqube> for me **adv_devops_7_sonarqube**

In **Server URL** Default is **http://localhost:9000**

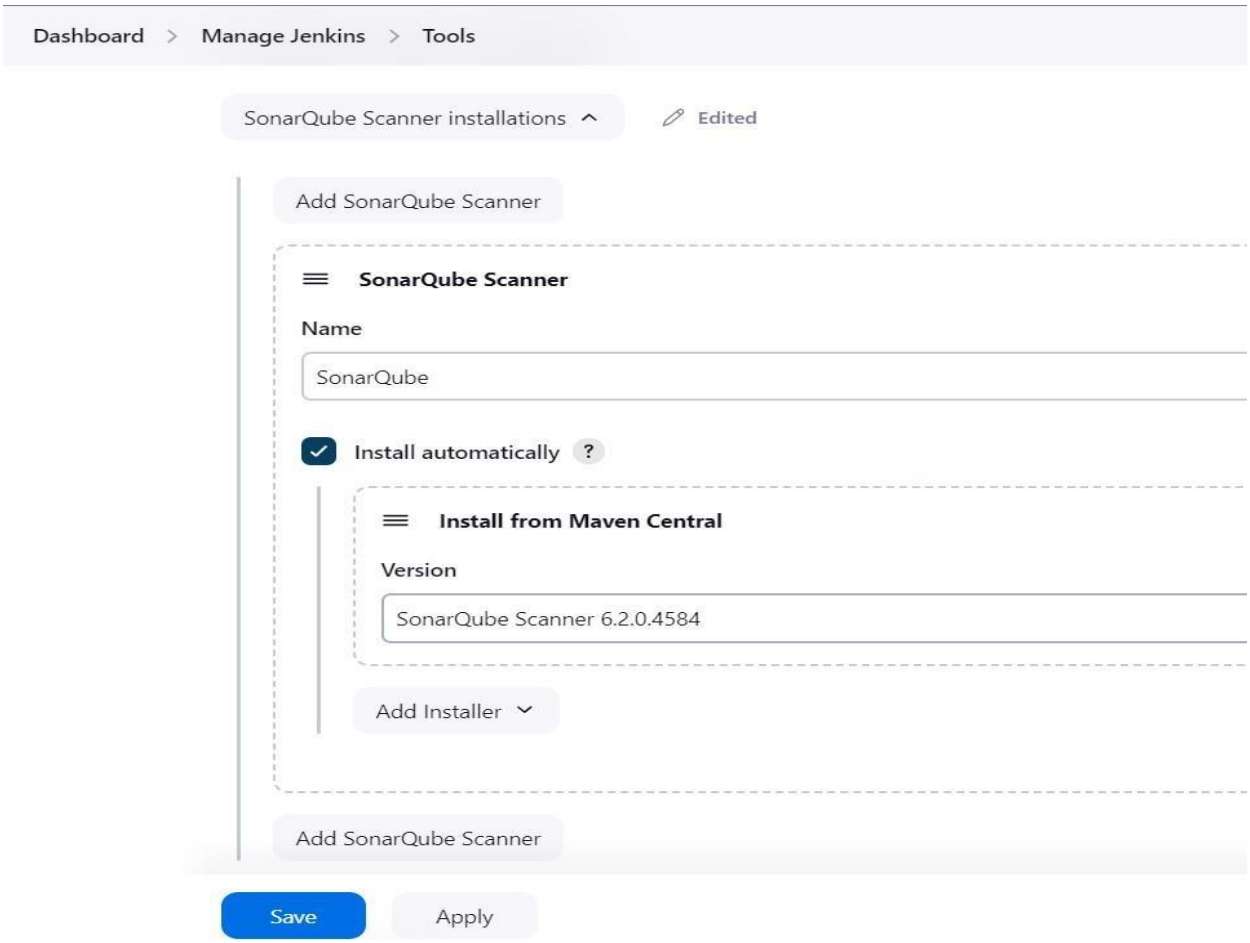
8. Search for SonarQube Scanner under Global Tool Configuration.

Choose the latest configuration and choose Install automatically.

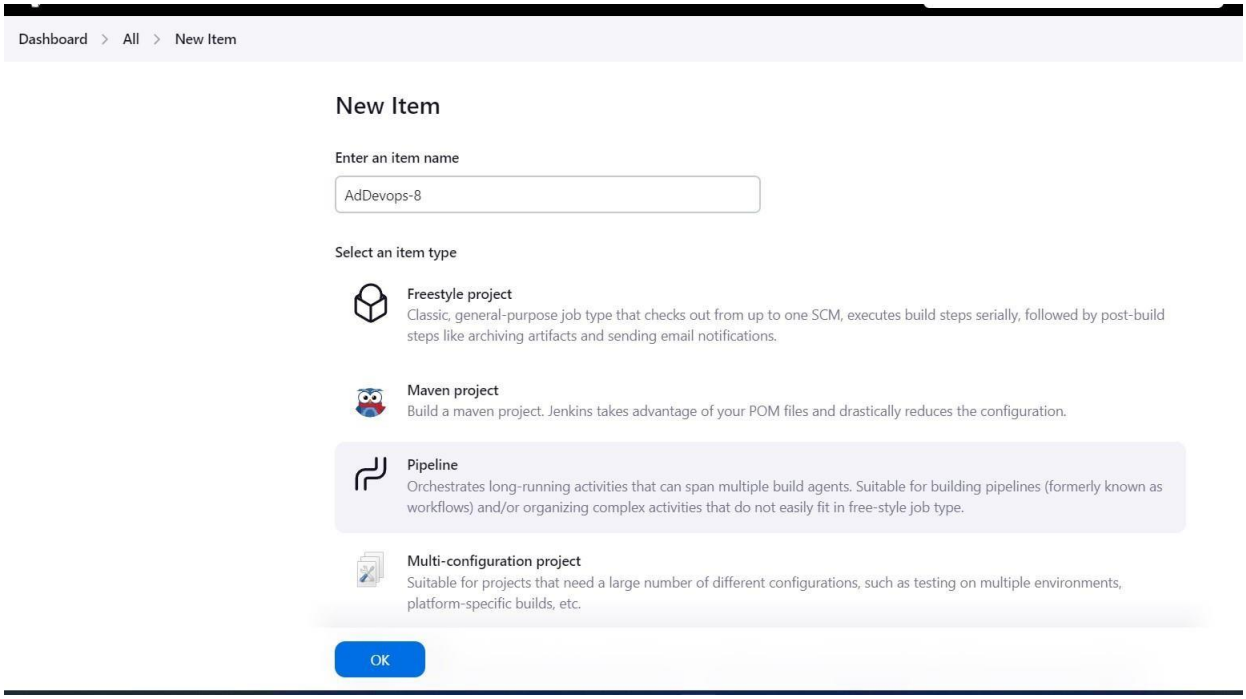
Dashboard > Manage Jenkins > Tools



Check the “Install automatically” option. → Under name any name as identifier → Check



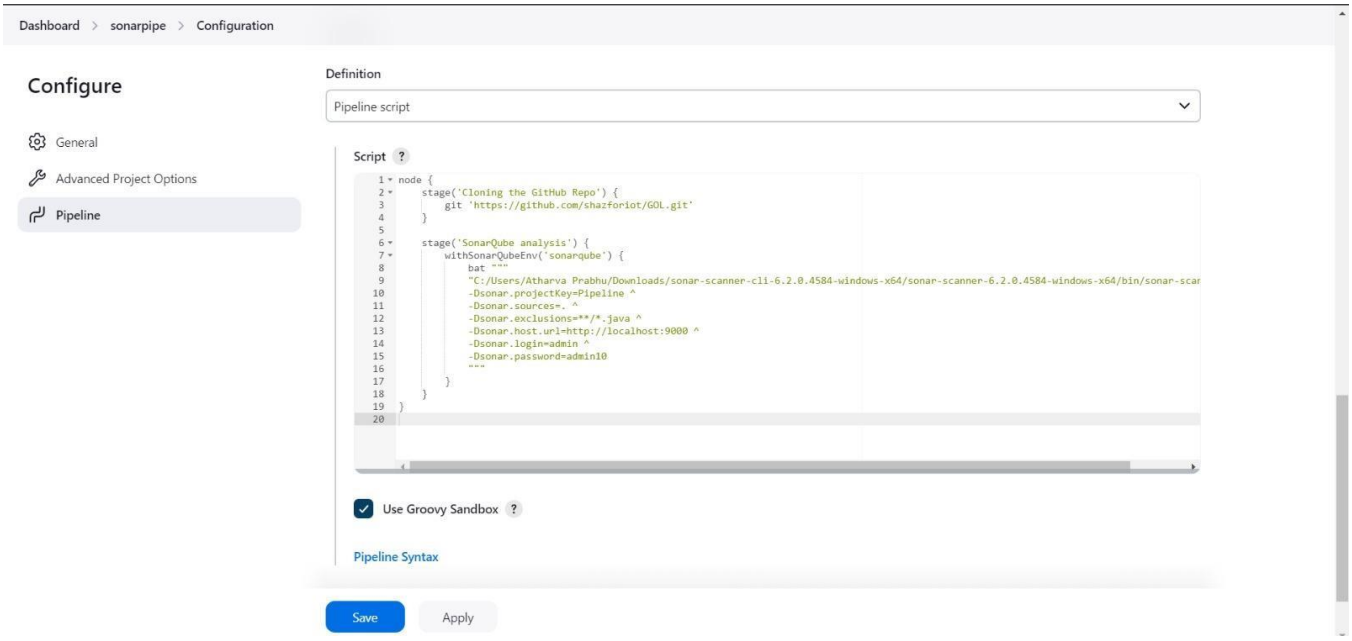
9. After configuration, create a New Item → choose a pipeline project.

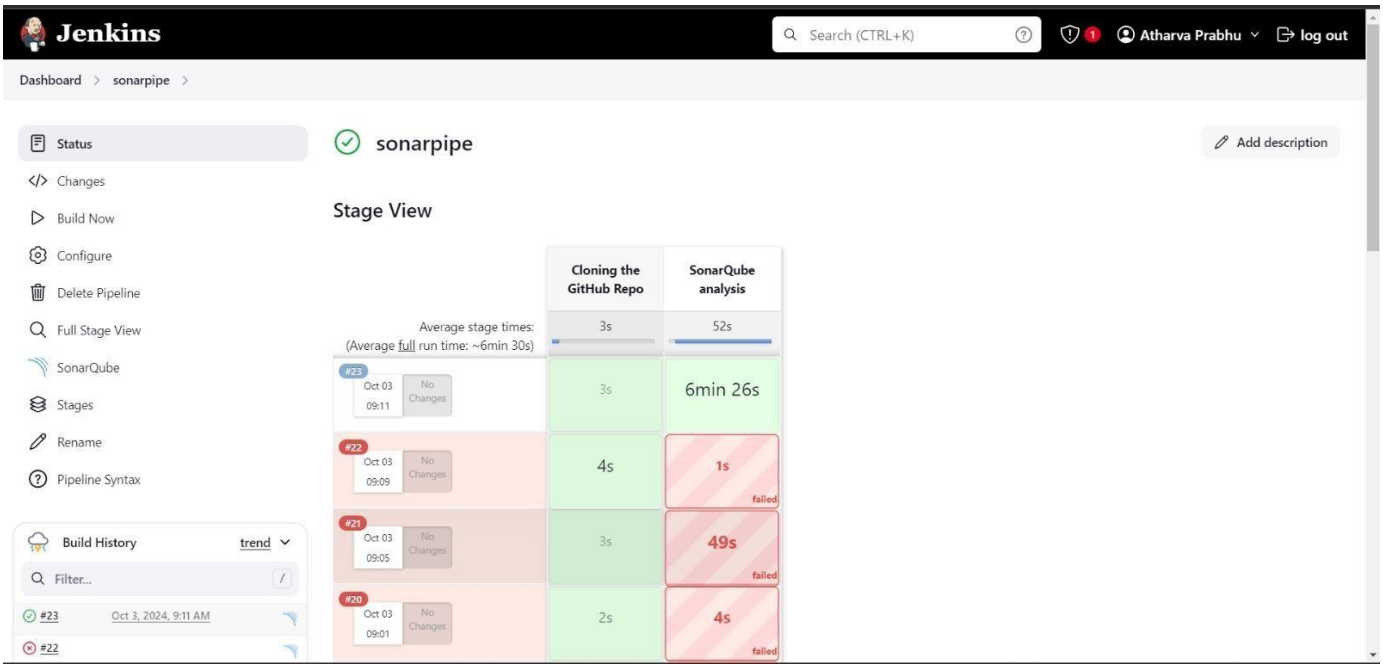


10. Under Pipeline script, enter the following:

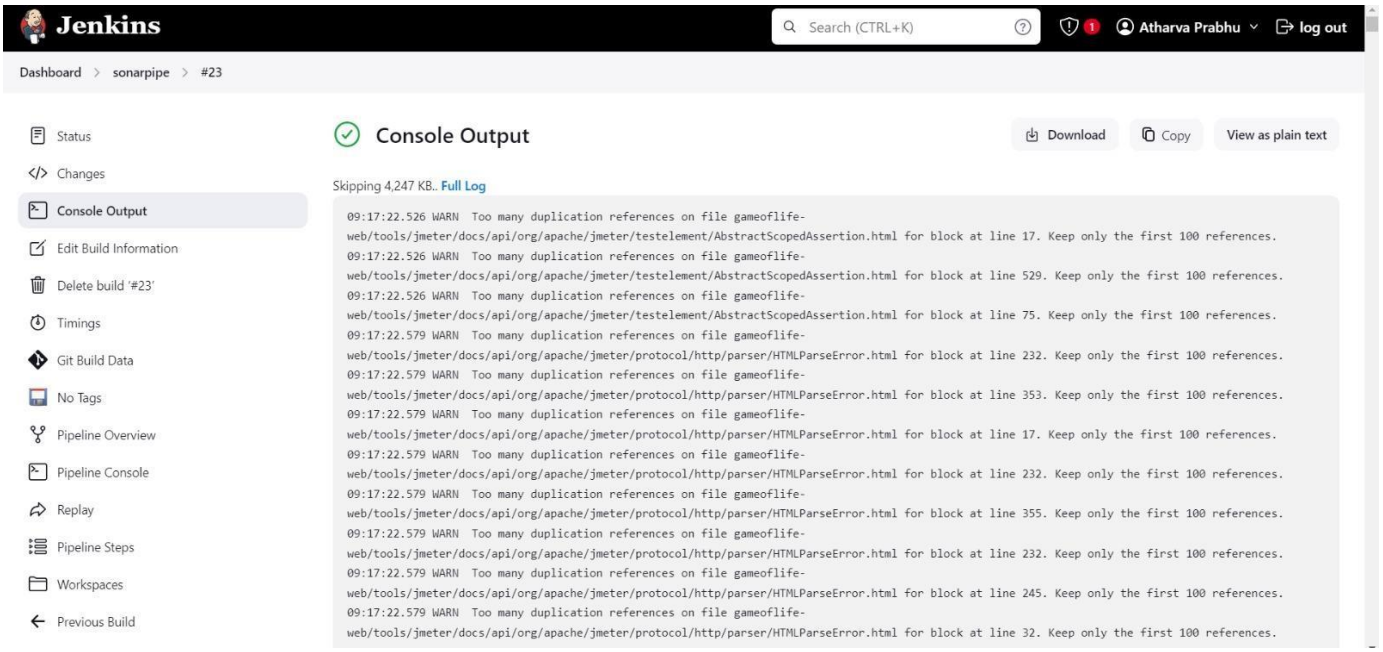
```
        node {
    stage('Cloning the GitHub Repo') { git
      'https://github.com/shazforiot/GOL.git'
    } stage('SonarQube

    analysis') {

      withSonarQubeEnv('<Name_of_SonarQube_environment_on_Jenk
      i ns>') { sh """
            <PATH_TO_SONARQUBE_SCANNER_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=<SonarQube_URL>(default: http://localhost:9000/)
          """
      }
    }
}
```

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

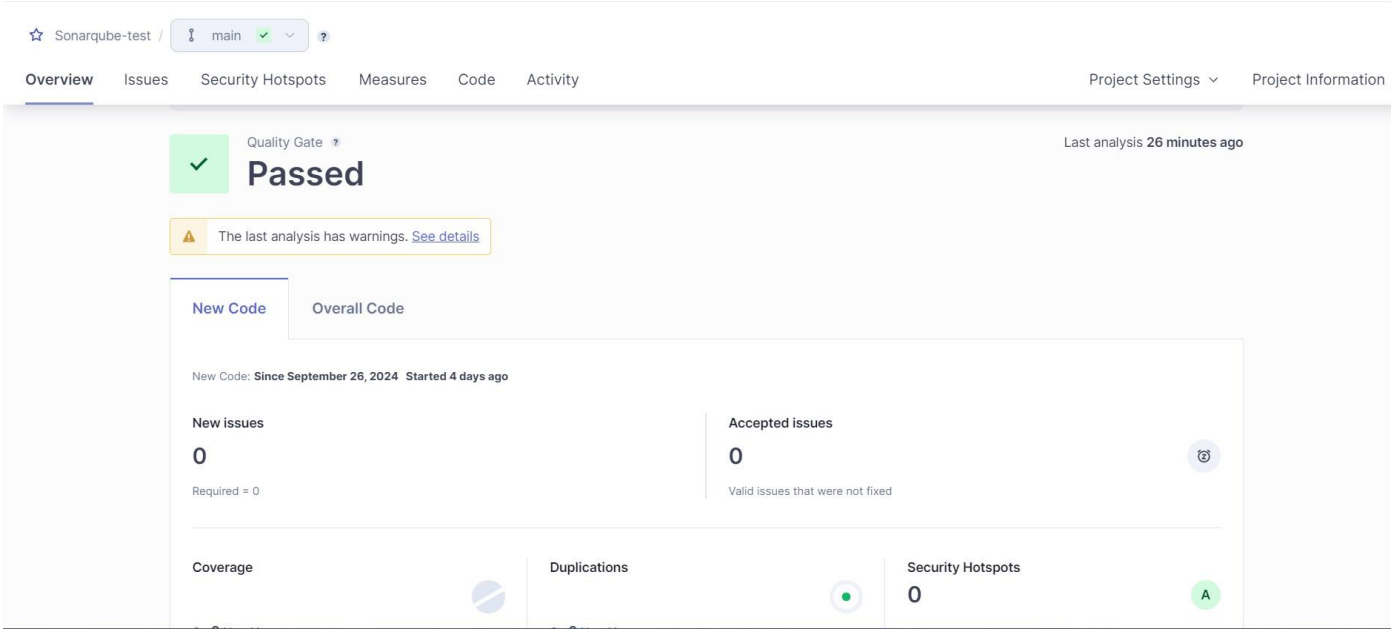




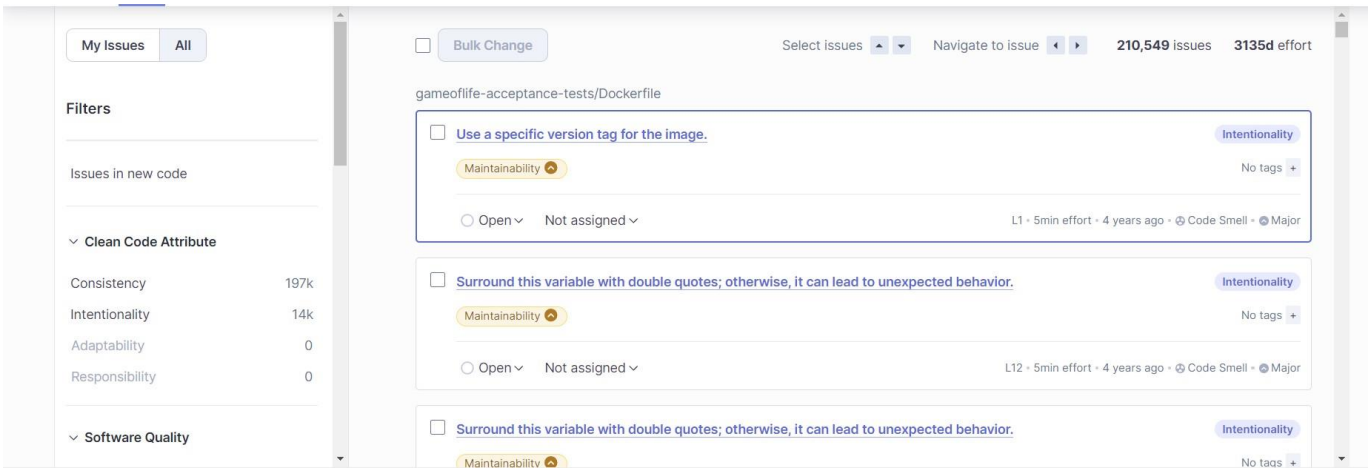
11. Check console



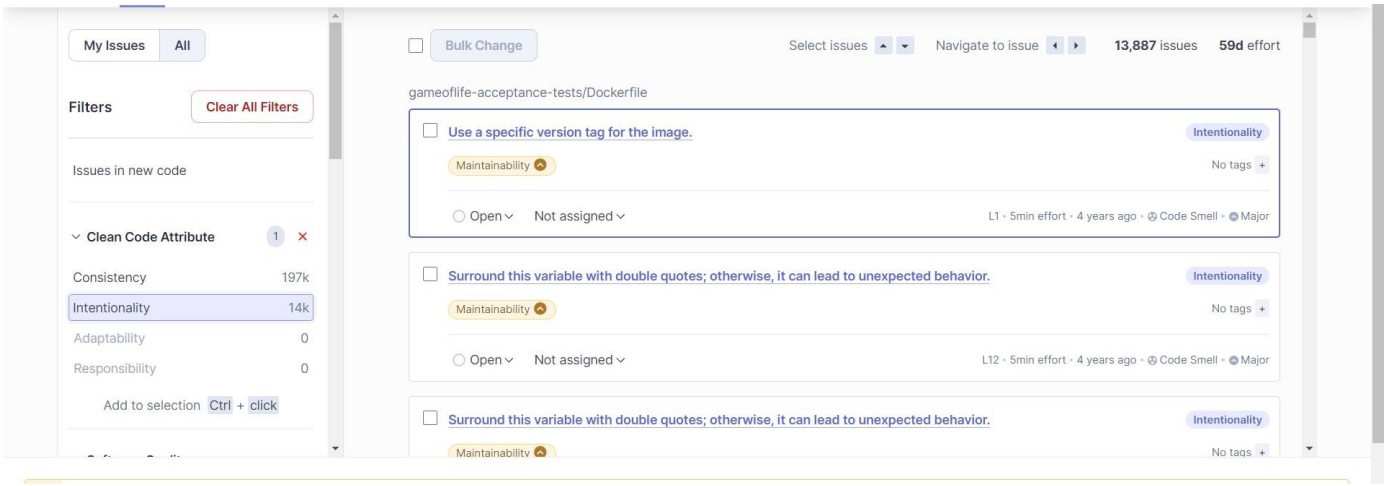
12. Now, check the project in SonarQube:



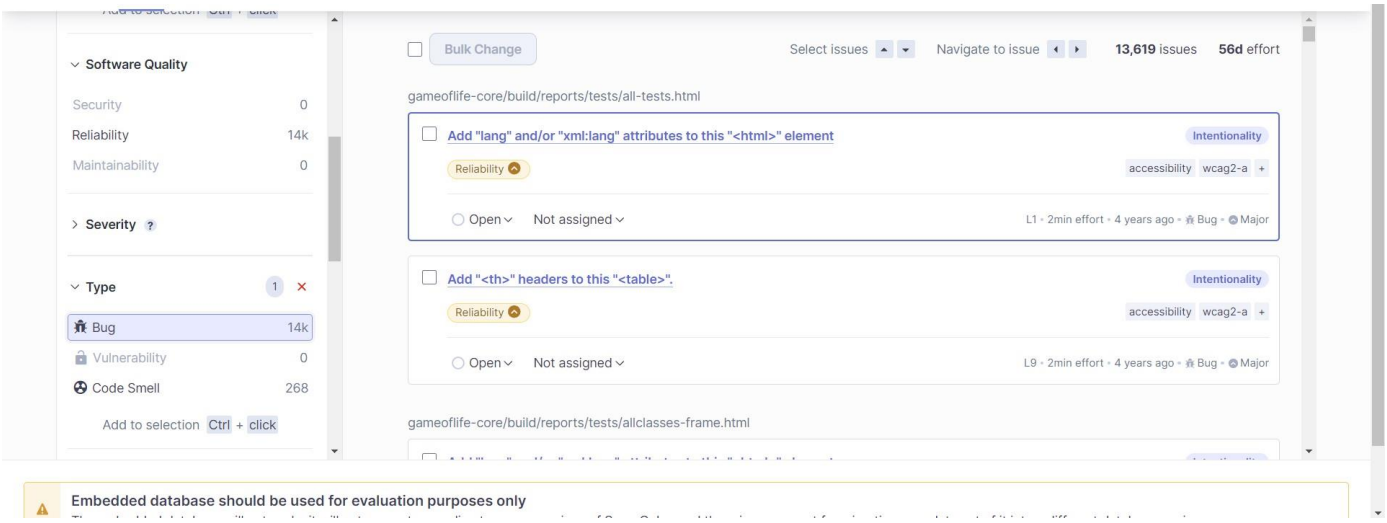
13. code problems consistency:



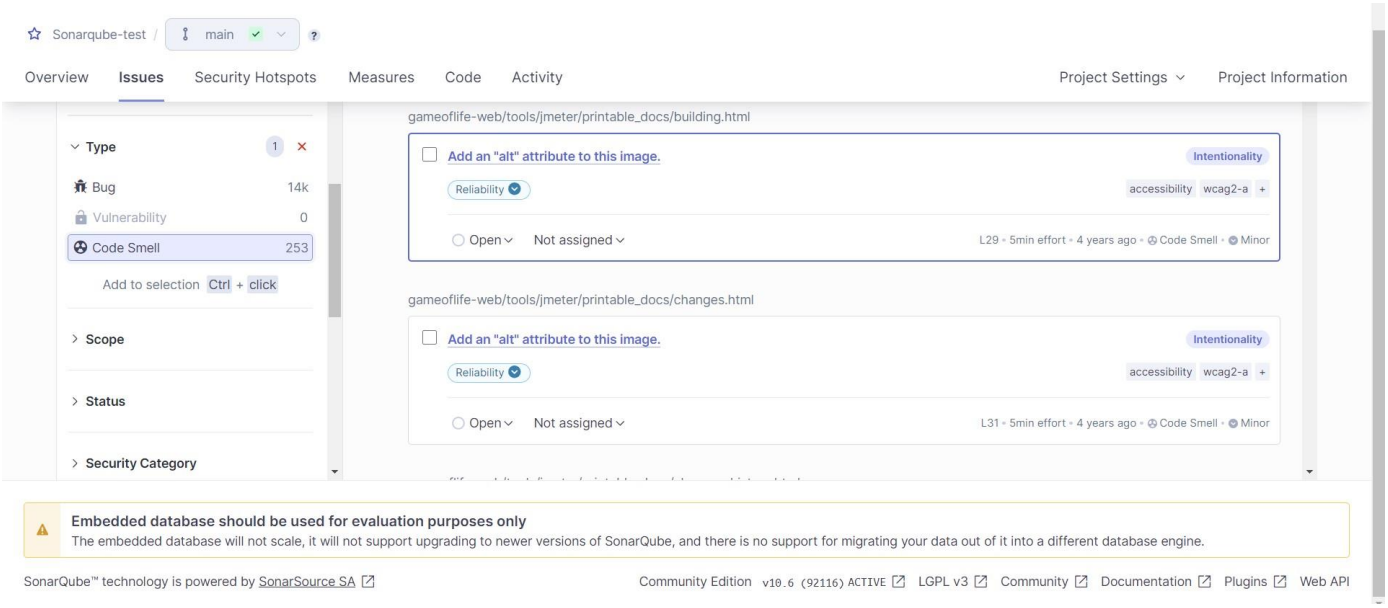
14. Intentionality:



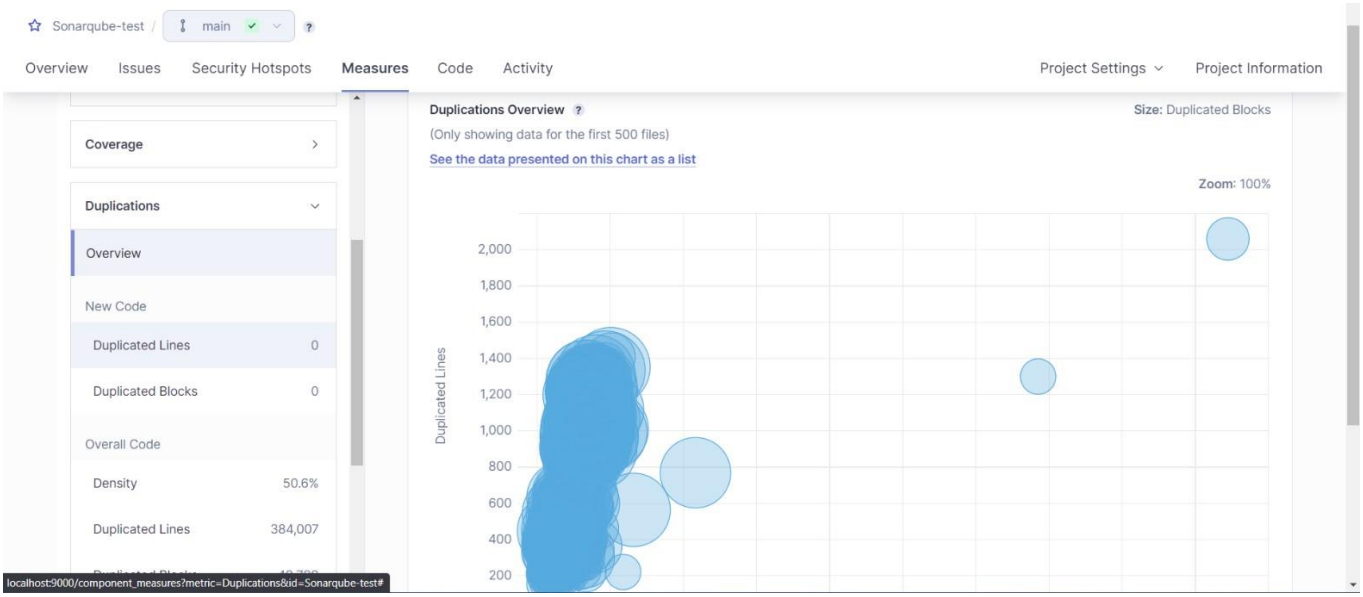
15. Bugs



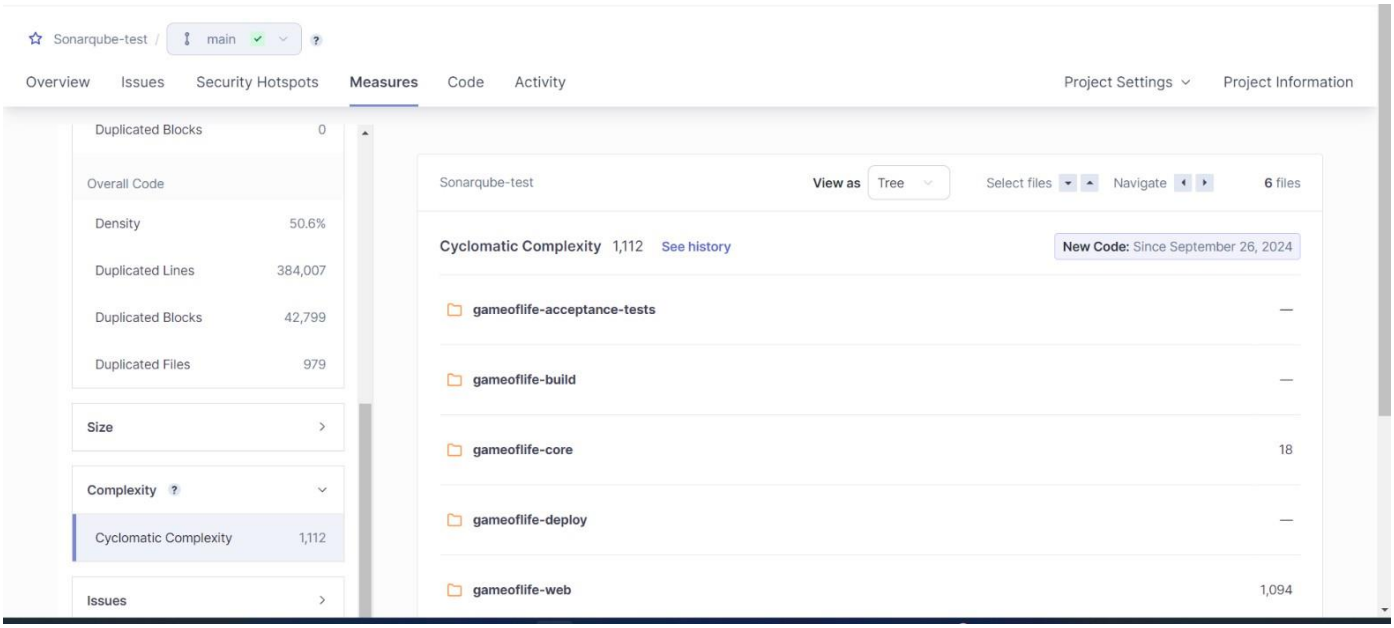
Code smells:



Duplications:



Cyclomatic Complexities:



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

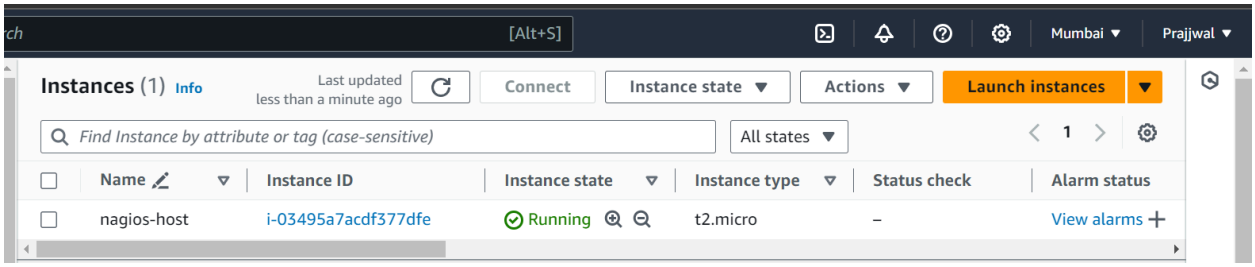
ADVANCE DEVOPS EXPERIMENT - 9

Prajjwal Pandey D15A / 33

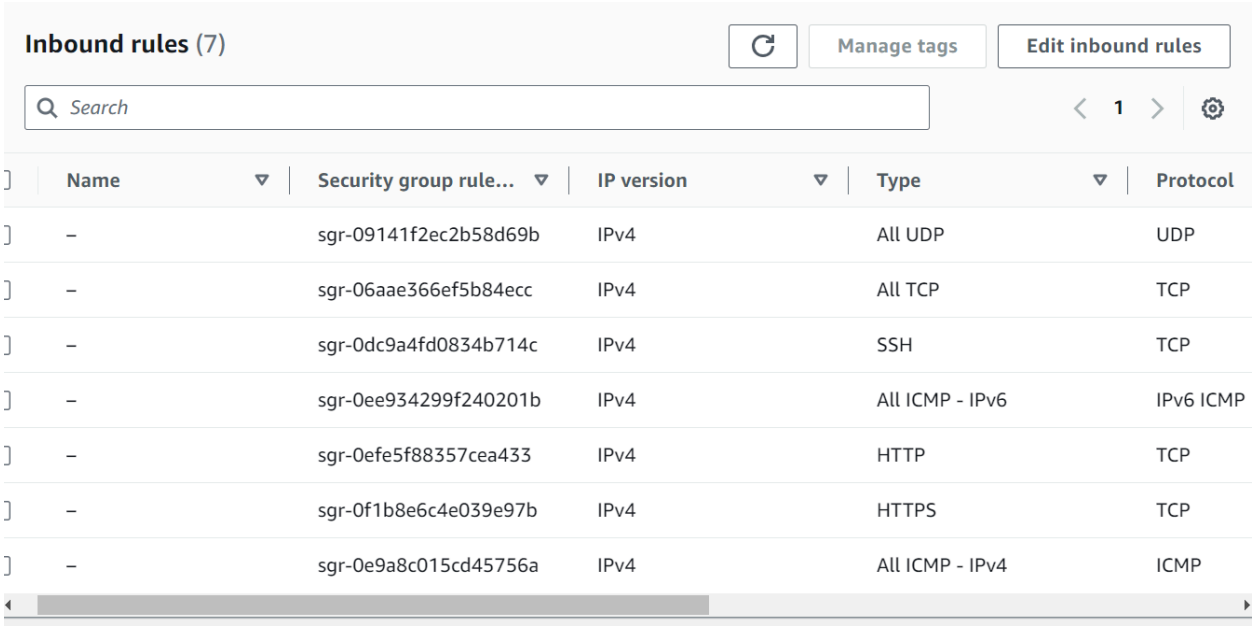
AIM: Installation of Nagios

Prerequisites: AWS Free Tier

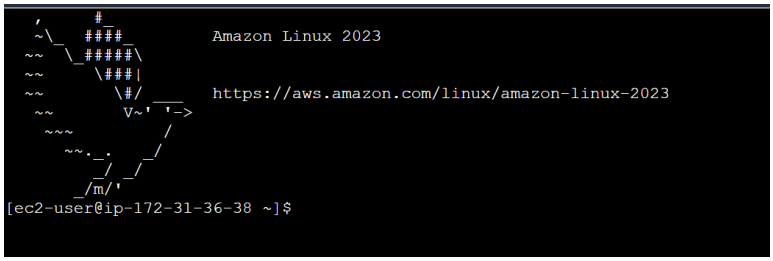
1. Create an Amazon Linux EC2 Instance in AWS and name it - nagios-host



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

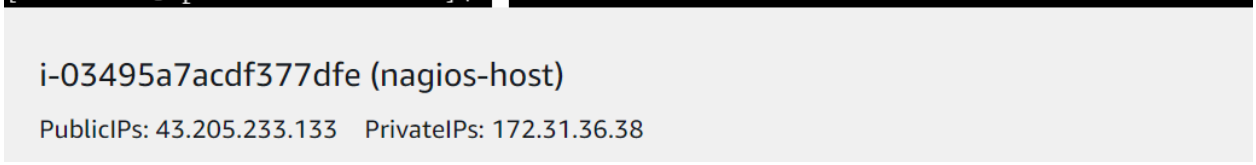


3. SSH into Your EC2 instance or simply use EC2 Instance Connect from the browser.



4. Update the package indices and install the following packages using yum
   sudo yum update
   sudo yum install httpd php
   sudo yum install gcc glibc glibc-common
   sudo yum install gd gd-devel

```
  libtiff-4.4.0-4.amzn2023.0.18.x86_64                                  libtiff
  libwebp-1.2.4-1.amzn2023.0.6.x86_64                                   libwebp
  libxcb-1.13.1-7.amzn2023.0.2.x86_64                                   libxcb-
  libxml2-devel-2.10.4-1.amzn2023.0.6.x86_64                            pcre2-d
  pcre2-utf16-10.40-1.amzn2023.0.3.x86_64                               pcre2-u
  pixman-0.40.0-3.amzn2023.0.3.x86_64                                   sysprof
  xml-common-0.6.3-56.amzn2023.0.2.noarch                               xorg-x1
  xz-devel-5.2.5-9.amzn2023.0.2.x86_64                                  zlib-de

Complete!
[ec2-user@ip-172-31-36-38 ~]$
```



5. Create a new Nagios User with its password. You'll have to enter the password twice for Confirmation.
   sudo adduser -m nagios
   sudo passwd nagios

```
Sorry, passwords do not match.
New password:
BAD PASSWORD: The password contains the user name in some form
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-36-38 ~]$
```

6. Create a new user group
   sudo groupadd nagcmd

7. Use these commands so that you don't have to use sudo for Apache and Nagios
   sudo usermod -a -G nagcmd nagios
   sudo usermod -a -G nagcmd apache

8. 8. Create a new directory for Nagios downloads
   mkdir ~/downloads
   cd ~/downloads

9. Use wget to download the source zip files.
   wget
   http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.0.8.tar.gz

wget http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz

```
nagios-plugins-2.0.3.tar.gz       100%[=============================================================>]

2024-09-29 14:00:39 (1.10 MB/s) - 'nagios-plugins-2.0.3.tar.gz' saved [2659772/2659772]

[ec2-user@ip-172-31-36-38 ~]$
```

i-03495a7acdf377dfe (nagios-host)

PublicIPs: 43.205.233.133 PrivateIPs: 172.31.36.38

10. Use tar to unzip and change to that directory.
    tar zxvf nagios-4.0.8.tar.gz

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

```
           Init directory:  /etc/rc.d/init.d
  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.
```

12. Compile the source code.
    make all

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

```
*** Config files installed ***

Remember, these are *SAMPLE* config files.  You'll need to read
the documentation for more information on how to actually define
services, hosts, etc. to fit your particular needs.

/usr/bin/install -c -m 775 -o nagios -g nagcmd -d /usr/local/nagios/var/rw
chmod g+s /usr/local/nagios/var/rw

*** External command directory configured ***
```

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

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

# 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 templat
       alias                           Nagios Admin            ; Full name of user

^O
       email                           prajjwal0904@gmail.com  ; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS ******
       }
^X
```

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

```
*** External command directory configured ***

[ec2-user@ip-172-31-36-38 nagios-4.0.8]$ sudo nano /usr/local/nagios/etc/objects/contacts.cfg
[ec2-user@ip-172-31-36-38 nagios-4.0.8]$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf

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

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

```
[ec2-user@ip-172-31-36-38 nagios-4.0.8]$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
[ec2-user@ip-172-31-36-38 nagios-4.0.8]$
```

17. Restart Apache
  sudo service httpd restart

18. Go back to the downloads folder and unzip the plugins zip file.
cd ~/downloads
tar zxvf nagios-plugins-2.0.3.tar.gz

19. Compile and install plugins
cd nagios-plugins-2.0.3
./configure --with-nagios-user=nagios --with-nagios-group=nagios
make
sudo make install

20. Start Nagios

Add Nagios to the list of system services
sudo chkconfig --add nagios

sudo chkconfig nagios on

Verify the sample configuration files

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

If there are no errors, you can go ahead and start Nagios.

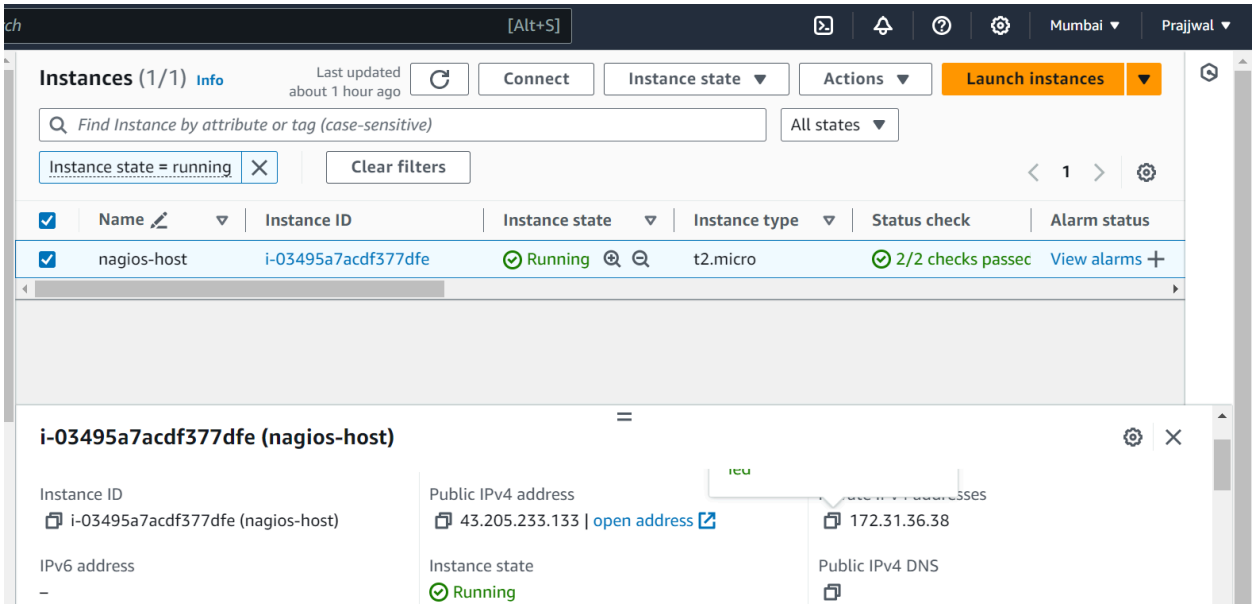
sudo service nagios start

```
Total Warnings: 0
Total Errors:   0

Things look okay - No serious problems were detected during the pre-flight check
[ec2-user@ip-172-31-36-38 bin]$ sudo service nagios start
Reloading systemd:                                         [  OK  ]
Starting nagios (via systemctl):                           [  OK  ]
[ec2-user@ip-172-31-36-38 bin]$
```

21. Check the status of Nagios
sudo systemctl status nagios

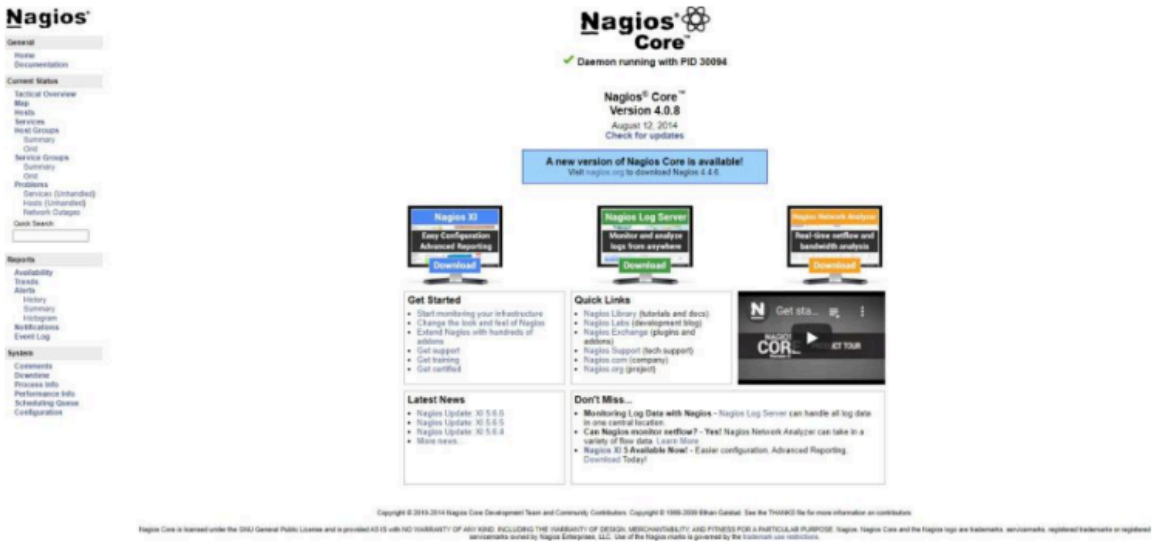
```
Starting nagios (via systemctl):                           [  OK  ]
[ec2-user@ip-172-31-36-38 bin]$ sudo systemctl status nagios
● nagios.service - LSB: Starts and stops the Nagios monitoring server
     Loaded: loaded (/etc/rc.d/init.d/nagios; generated)
     Active: active (running) since Sun 2024-09-29 15:15:32 UTC; 45s ago
       Docs: man:systemd-sysv-generator(8)
    Process: 72490 ExecStart=/etc/rc.d/init.d/nagios start (code=exited, status=0/SUCCESS)
      Tasks: 6 (limit: 1112)
     Memory: 2.2M
        CPU: 56ms
     CGroup: /system.slice/nagios.service
```

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



23. Open up your browser and look for http://<your_public_ip_address>/nagios
    Enter username as nagiosadmin and password which you set in Step 16.

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



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

ADVANCE DEVOPS EXPERIMENT - 10

Prajjwal Pandey D15A / 33

Prerequisites

- AWS Free Tier
- Nagios Server running on an Amazon Linux Machine

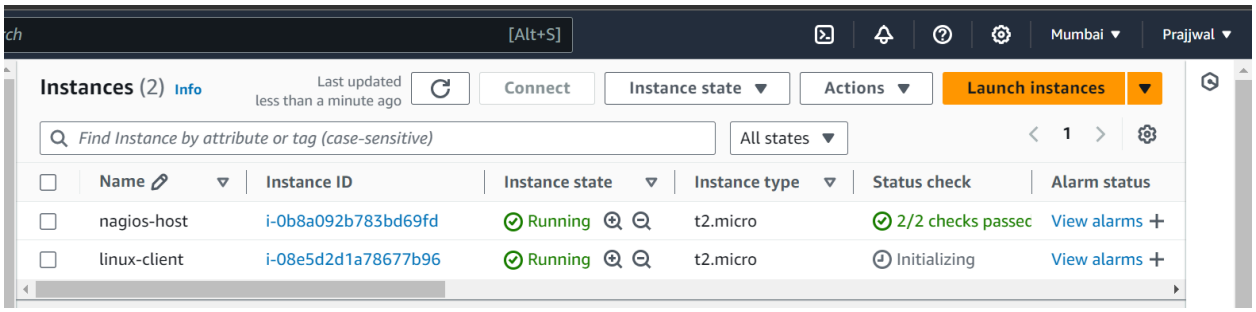
Steps

1. Confirm Nagios is Running
   sudo systemctl status nagios

```
● nagios.service - Nagios Core 4.4.6
     Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
     Active: active (running) since Tue 2024-10-01 16:21:57 UTC; 8s ago
       Docs: https://www.nagios.org/documentation
    Process: 67457 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
    Process: 67458 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (c
   Main PID: 67459 (nagios)
      Tasks: 6 (limit: 1112)
     Memory: 2.0M
        CPU: 17ms
     CGroup: /system.slice/nagios.service
             ├─67459 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─67460 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─67461 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─67462 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─67463 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─67464 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
```

2. Create a Linux Client

   - Create an Ubuntu 20.04 server EC2 instance in AWS.
   - Assign it the same security group as the Nagios Host.
   - Name it `linux-client`.



3. Verify Nagios Process
ps -ef | grep nagios

```
[ec2-user@ip-172-31-37-184 nagios-plugins-2.3.3]$ ps -ef | grep nagios
nagios     67459       1  0 16:21 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios     67460   67459  0 16:21 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios     67461   67459  0 16:21 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios     67462   67459  0 16:21 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios     67463   67459  0 16:21 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios     67464   67459  0 16:21 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
ec2-user   68320    2728  0 16:37 pts/0    00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-37-184 nagios-plugins-2.3.3]$
```

4. Create Directories for Monitoring Hosts
sudo su
mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts

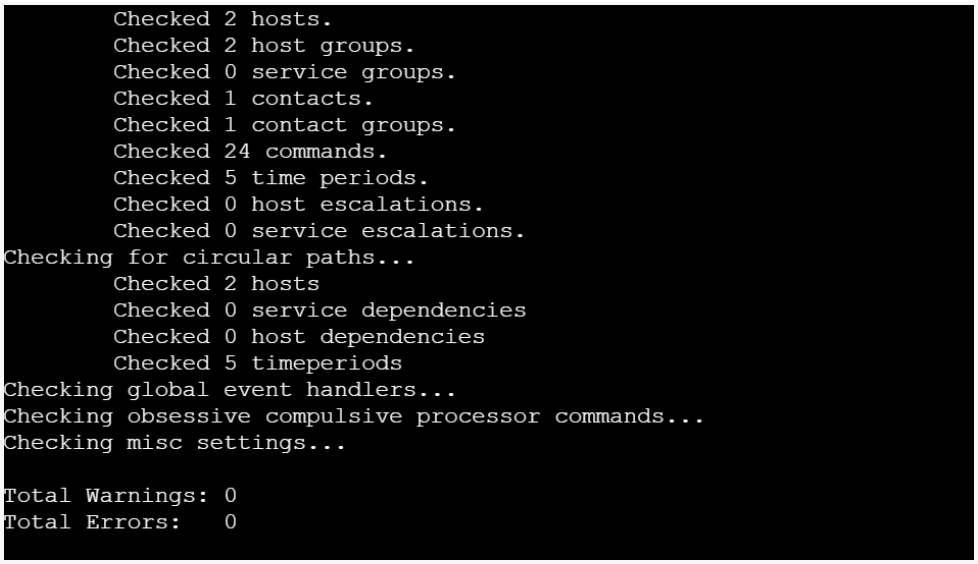
5. Copy and Modify Configuration File
cp /usr/local/nagios/etc/objects/localhost.cfg
/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
nano /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg

- Change `hostname` to `linuxserver` everywhere in the file.
- Change `address` to the public IP address of your `linux-client`.
- Change `hostgroup_name` under hostgroup to `linux-servers1`.

6. Update Nagios Configuration
Add the following line: cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/

```
# Definitions for monitoring a router/switch
#cfg_file=/usr/local/nagios/etc/objects/switch.cfg

# Definitions for monitoring a network printer
#cfg_file=/usr/local/nagios/etc/objects/printer.cfg


# 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/
```

7. Verify Configuration Files
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg



8. Restart Nagios Service
sudo systemctl restart nagios

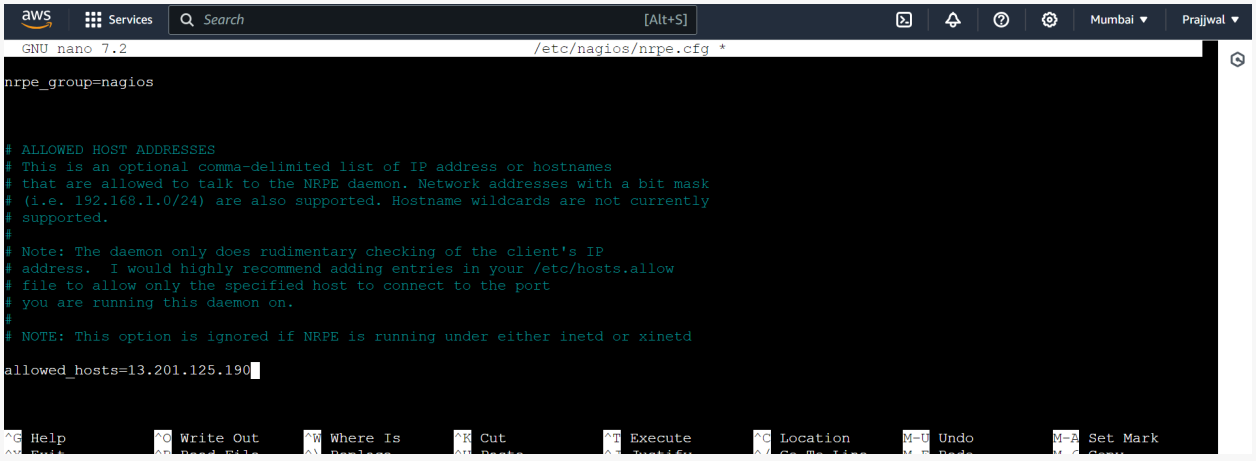
9. Switch to Client Machine

- SSH into the `linux-client` machine or use the EC2 Instance Connect feature.

10. Install Required Packages on Client.
sudo apt update -y
sudo apt install gcc -y
sudo apt install -y nagios-nrpe-server nagios-plugins

11. Configure NRPE
sudo nano /etc/nagios/nrpe.cfg

- Under `allowed_hosts`, add your Nagios host IP address.



12. Restart NRPE Server

`sudo systemctl restart nagios-nrpe-server`

13. Check Nagios Dashboard

- Go to your Nagios dashboard.
- Click on `Hosts` and then on `linuxserver` to see the host details.
- Click on `Services` to see all services and ports being monitored.

Host Information

Last Updated: Tue Oct 1 16:52:19 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.4.6 - www.nagios.org
Logged in as *nagiosadmin*

View Status Detail For This Host
View Alert History For This Host
View Trends For This Host
View Alert Histogram For This Host
View Availability Report For This Host
View Notifications For This Host

Host
localhost
(linuxserver)

Member of
No hostgroups

13.234.202.182

Host State Information

Host Status:	**DOWN** (for 0d 0h 6m 37s)
Status Information:	PING CRITICAL - Packet loss = 100%
Performance Data:	rta=5000.000000ms;3000.000000;5000.000000;0.000000 pl=100%;80;100;0
Current Attempt:	4/10 (SOFT state)
Last Check Time:	10-01-2024 16:50:12
Check Type:	ACTIVE
Check Latency / Duration:	0.000 / 30.003 seconds
Next Scheduled Active Check:	10-01-2024 16:51:42
Last State Change:	10-01-2024 16:45:42
Last Notification:	N/A (notification 0)
Is This Host Flapping?	**NO** (5.86% state change)
In Scheduled Downtime?	**NO**
Last Update:	10-01-2024 16:52:11 (0d 0h 0m 8s ago)

Active Checks:	**ENABLED**
Passive Checks:	**ENABLED**
Obsessing:	**ENABLED**
Notifications:	**ENABLED**
Event Handler:	**ENABLED**
Flap Detection:	**ENABLED**

Current Network Status

Last Updated: Tue Oct 1 16:53:13 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.4.6 - 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
1	1	0	0

All Problems	All Types
1	2

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
6	1	0	1	0

All Problems	All Types
2	8

Service Status Details For All Hosts

Limit Results: 100

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	10-01-2024 16:52:34	0d 0h 30m 39s	1/4	OK - load average: 0.00, 0.00, 0.00
	Current Users	OK	10-01-2024 16:48:12	0d 0h 30m 1s	1/4	USERS OK - 2 users currently logged in
	HTTP	WARNING	10-01-2024 16:51:49	0d 0h 26m 24s	4/4	HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.001 second response time
	PING	OK	10-01-2024 16:49:27	0d 0h 28m 46s	1/4	PING OK - Packet loss = 0%, RTA = 0.05 ms
	Root Partition	OK	10-01-2024 16:50:04	0d 0h 28m 9s	1/4	DISK OK - free space: / 6080 MiB (74.92% inode=98%):
	SSH	OK	10-01-2024 16:51:57	0d 0h 27m 31s	1/4	SSH OK - OpenSSH_8.7 (protocol 2.0)
	Swap Usage	CRITICAL	10-01-2024 16:49:19	0d 0h 23m 54s	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	10-01-2024 16:51:57	0d 0h 26m 16s	1/4	PROCS OK: 39 processes with STATE = RSZDT

Results 1 - 8 of 8 Matching Services

ADVANCE DEVOPS EXPERIMENT - 11

Prajjwal Pandey D15A / 33

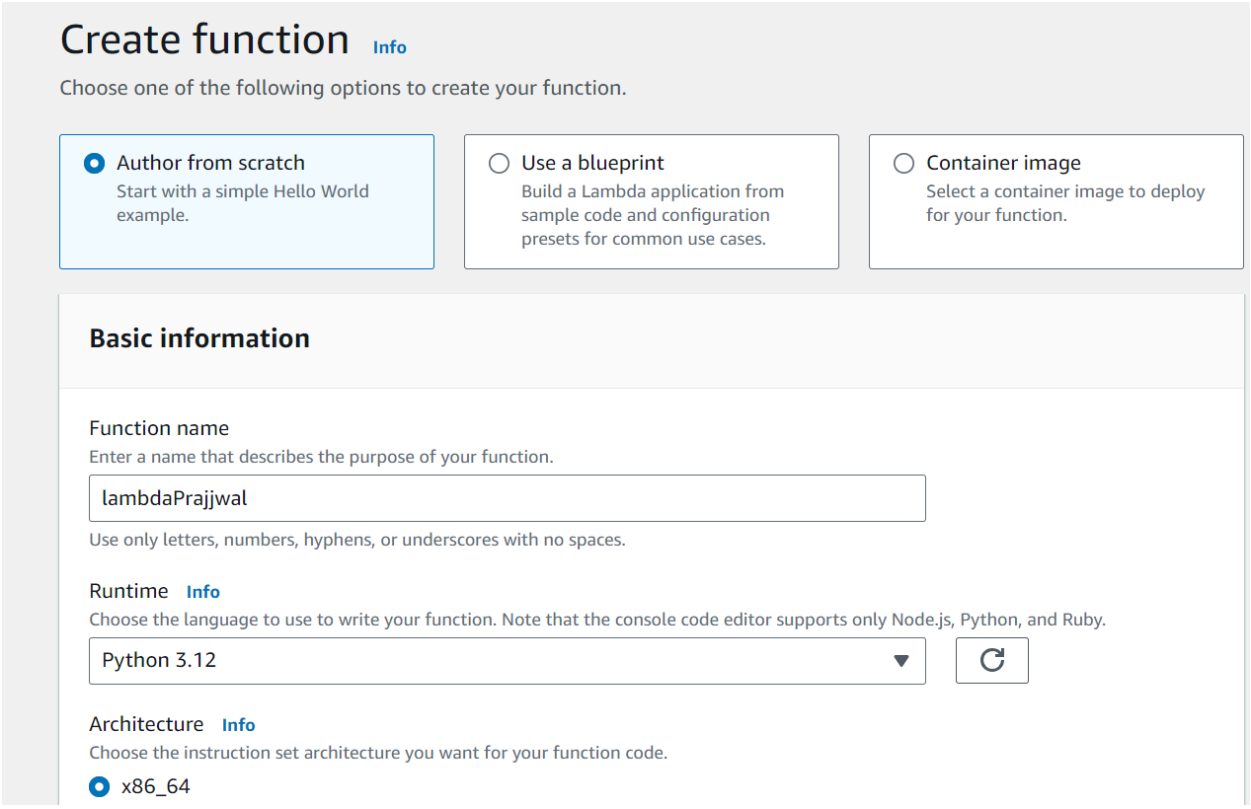
Steps:

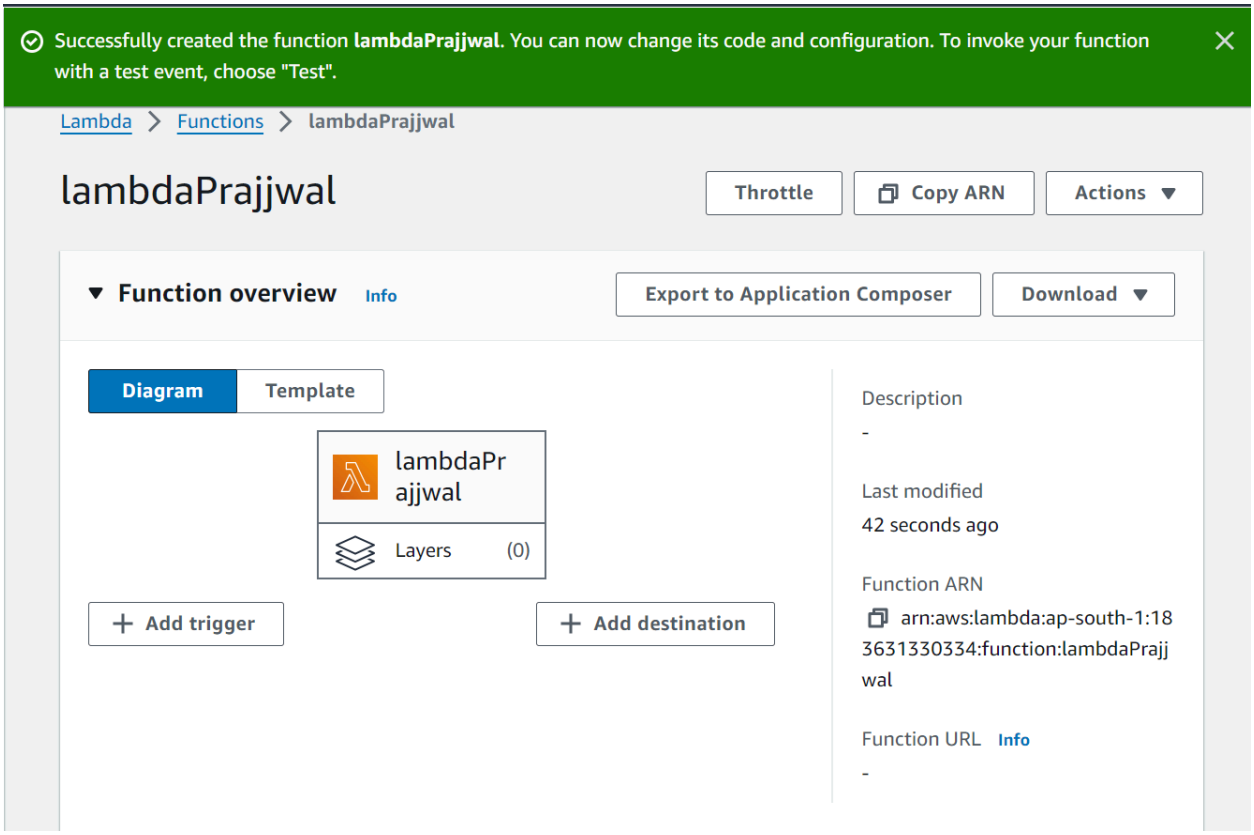
1. Open up the Lambda Console and click on the Create button.

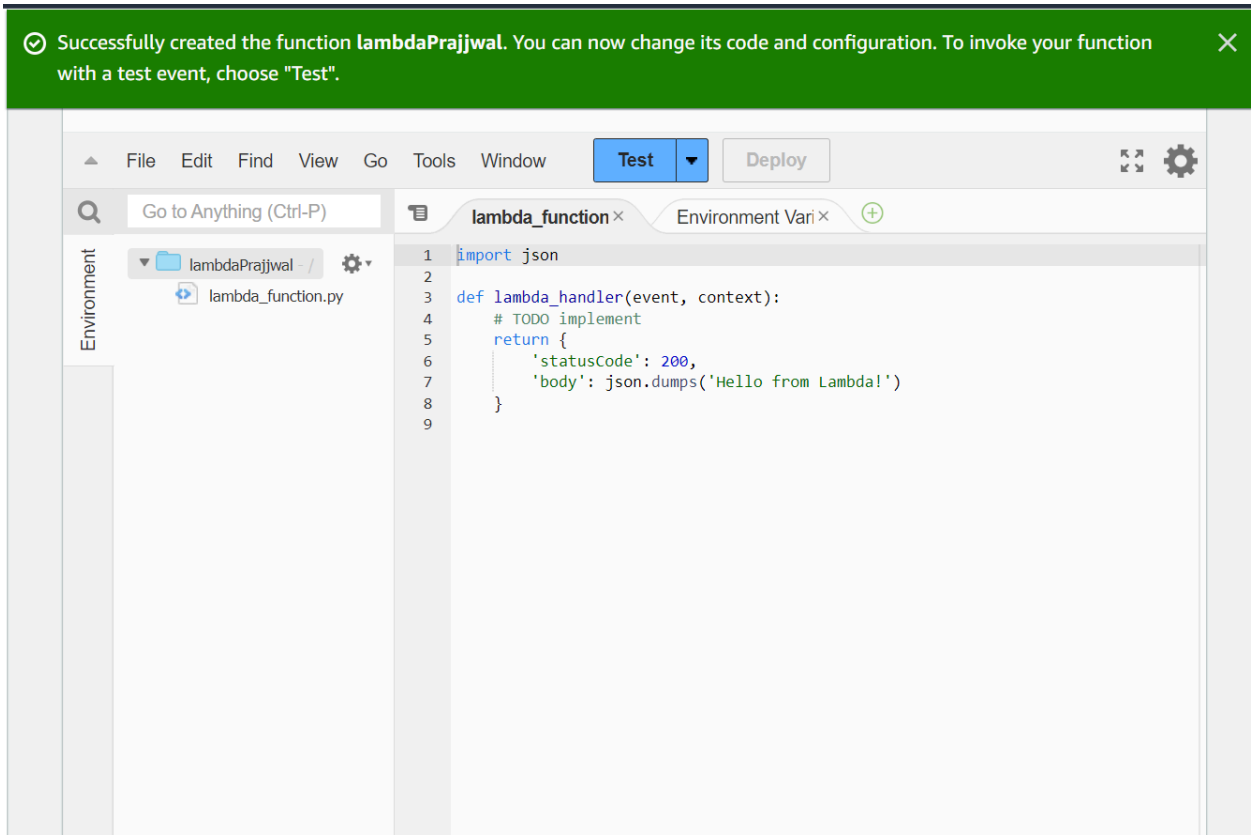
2. Choose to create a function from scratch or use a blueprint, i.e templates defined by AWS for you with all configuration presets required for the most common use cases.

   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.

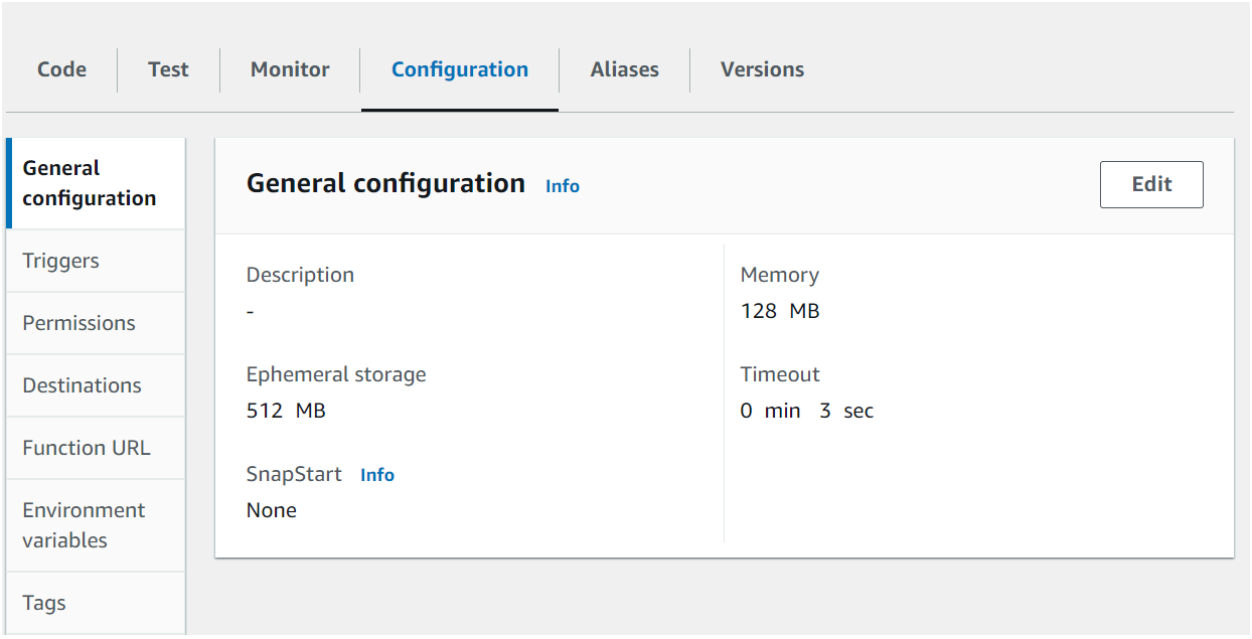
   After that, choose to create a new role with basic Lambda permissions if you don't have an existing one.

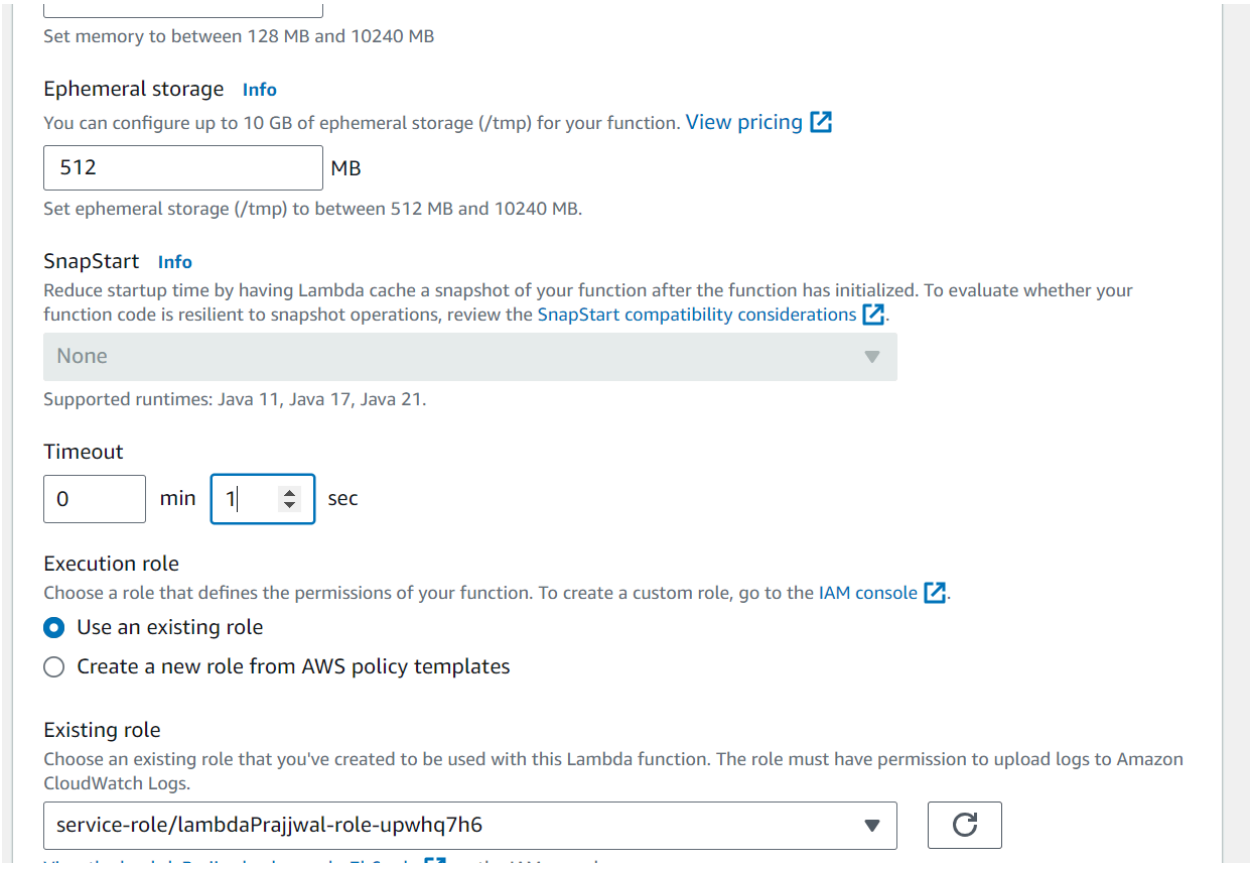



Successfully created the function lambdaPrajjwal. You can now change its code and configuration. To invoke your function with a test event, choose "Test".
Lambda > Functions > lambdaPrajjwal
lambdaPrajjwal
Throttle
Copy ARN
Actions
Function overview Info
Export to Application Composer
Download
Diagram
Template
lambdaPr ajjwal
Layers (0)
Add trigger
Add destination
Description
-
Last modified
42 seconds ago
Function ARN
arn:aws:lambda:ap-south-1:183631330334:function:lambdaPrajjwal
Function URL Info
-

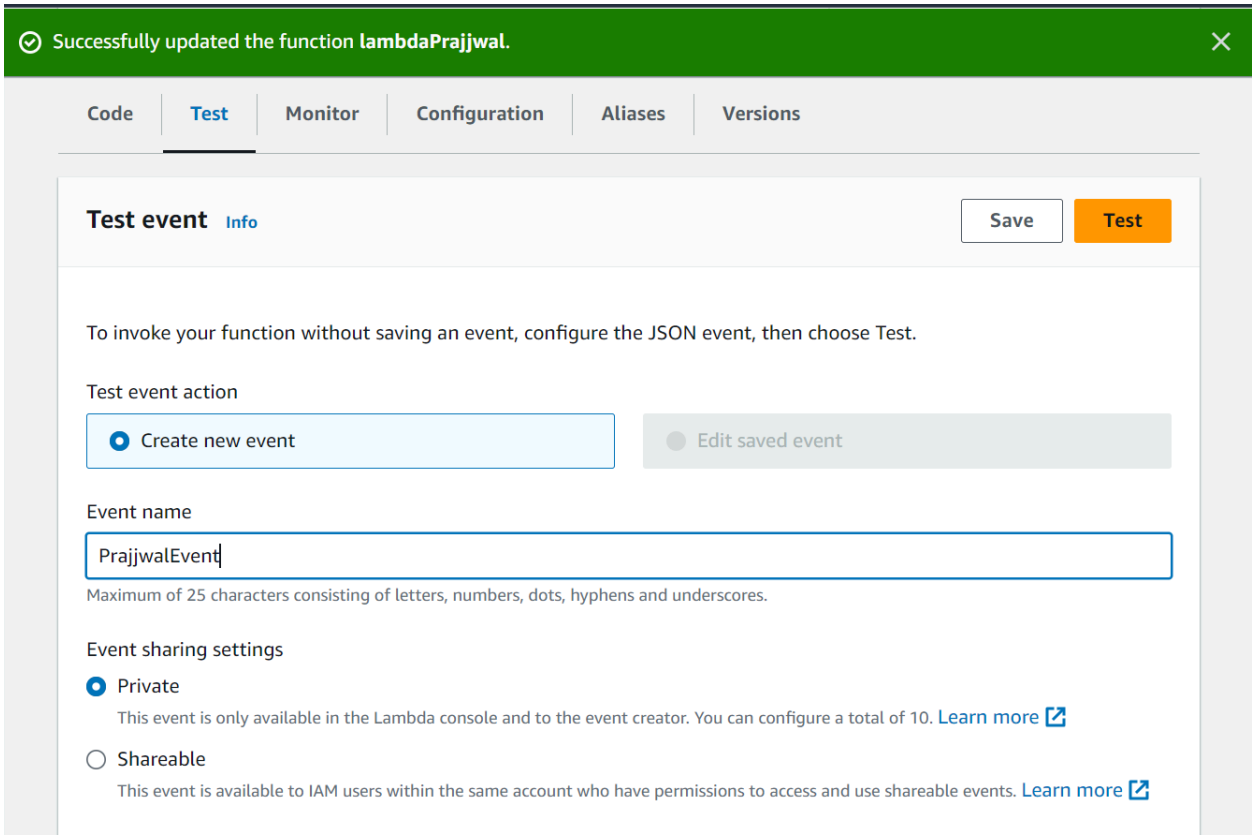

Successfully created the function lambdaPrajjwal. You can now change its code and configuration. To invoke your function with a test event, choose "Test".
File Edit Find View Go Tools Window
Test
Deploy
Go to Anything (Ctrl-P)
lambda_function
Environment Vari
Environment
lambdaPrajjwal
lambda_function.py
import json
def lambda_handler(event, context):
# TODO implement
return {
'statusCode': 200,
'body': json.dumps('Hello from Lambda!')
}

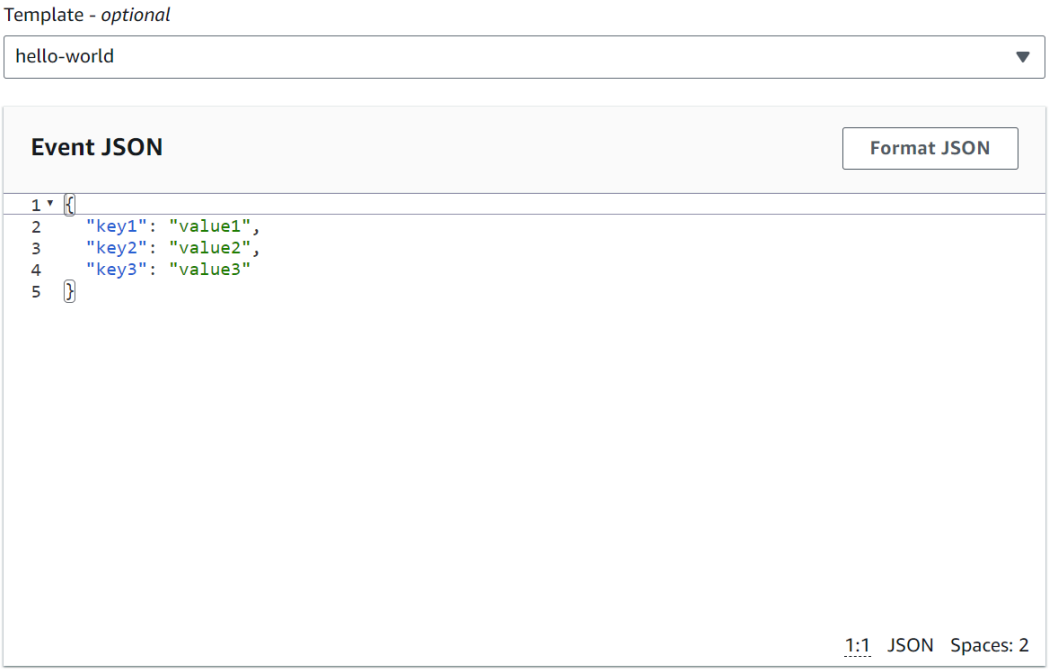
3. To change the configuration, open up the Configuration tab and under General Configuration, choose Edit. Here, you can enter a description and change Memory and Timeout. I've changed the Timeout period to 1 sec since that is sufficient for now.



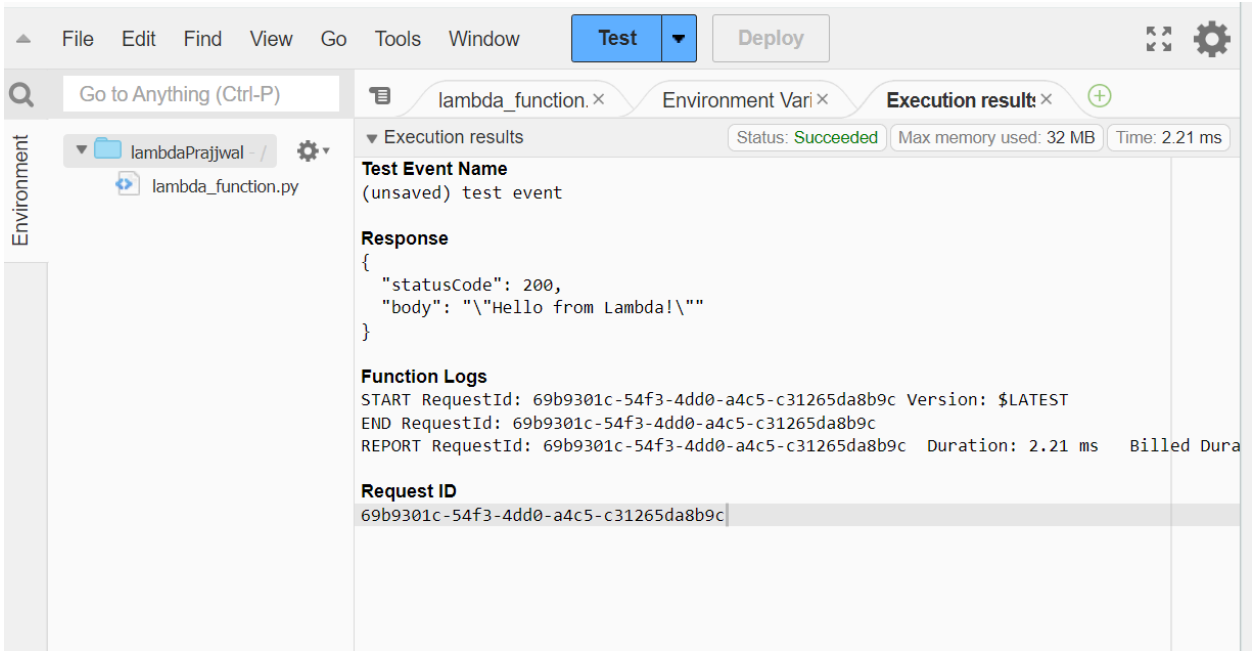


4. Click on Test and you can change the configuration, like so. If you do not have anything in the request body, it is important to specify two curly braces as valid JSON, so make sure they are there.





5. Now click on Test and you should be able to see the results.



ADVANCE DEVOPS EXPERIMENT - 12

Prajjwal Pandey D15A / 33

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

Theory:
AWS Lambda and S3 Integration: AWS Lambda allows you to execute code in response to various events, including those triggered by Amazon S3. When an object is added to an S3 bucket, it can trigger a Lambda function to execute, allowing for event-driven processing without managing servers.

Workflow:
1. Create an S3 Bucket:
○ First, create an S3 bucket that will store the objects. This bucket will act as the trigger source for the Lambda function.

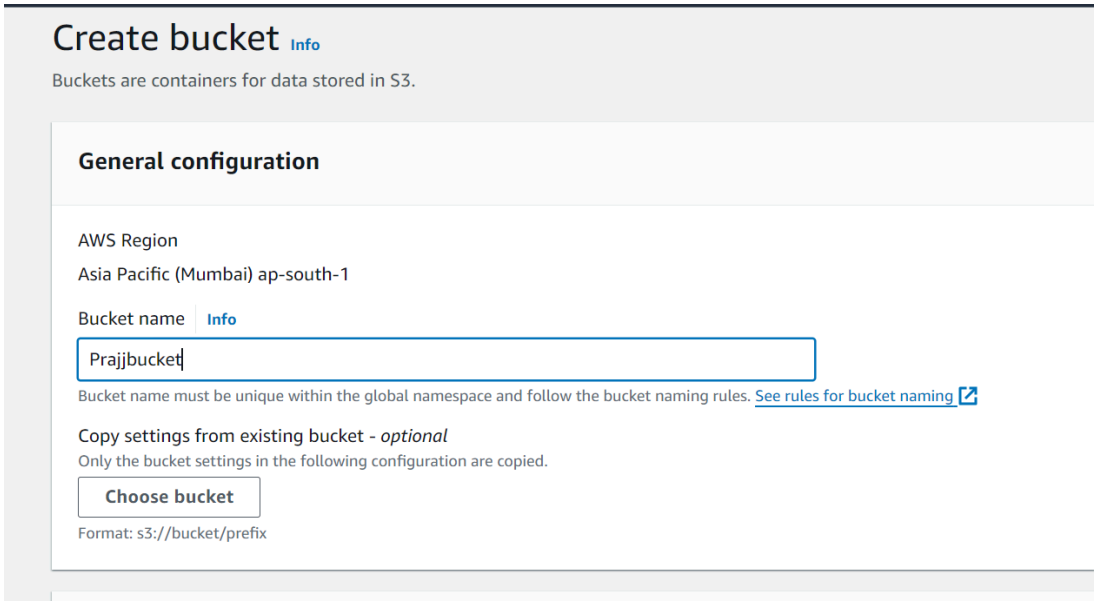
2. Create the Lambda Function:
○ Set up a new Lambda function using AWS Lambda’s console. You can choose a runtime environment like Python, Node.js, or Java.
○ Write code that logs a message like “An Image has been added” when triggered.
3. Set Up Permissions:
○ Ensure that the Lambda function has the necessary permissions to access S3. You can do this by attaching an IAM role with policies that allow reading from the bucket and writing logs to CloudWatch.

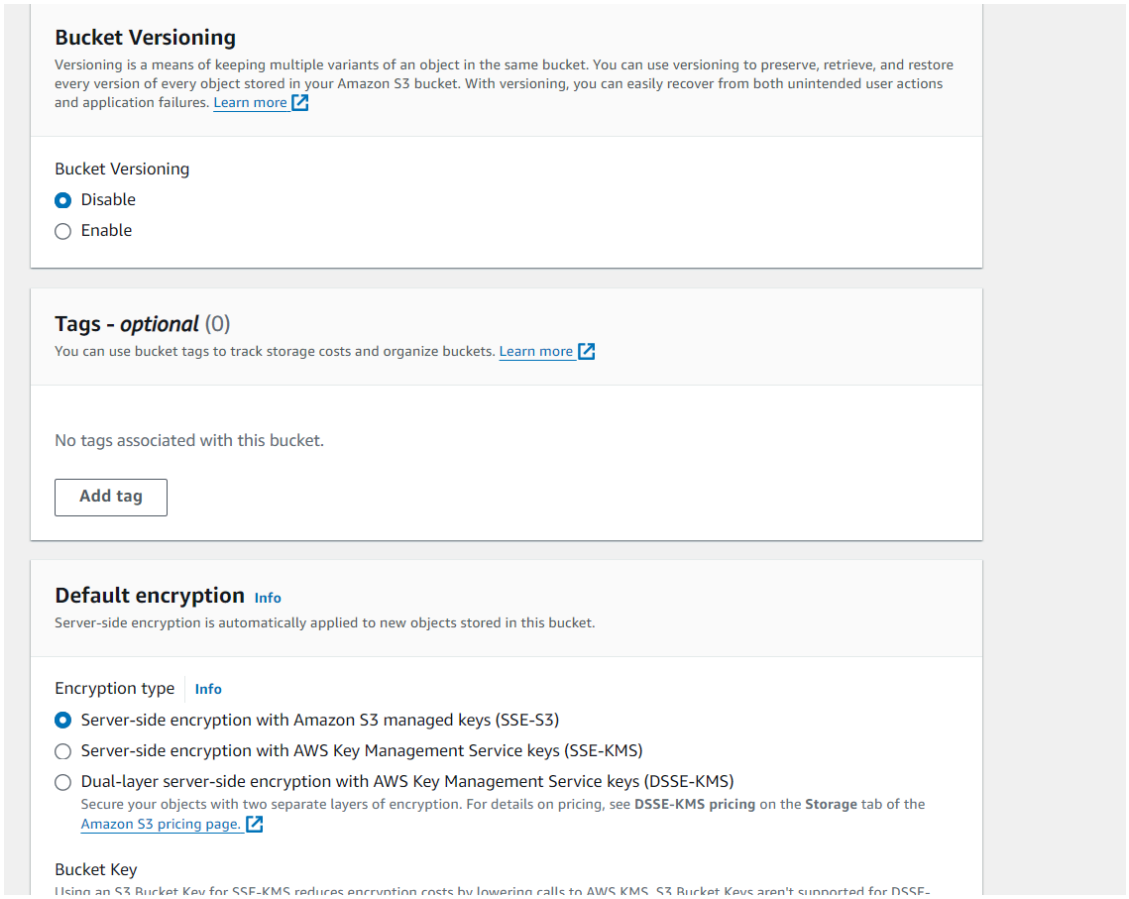
4. Configure S3 Trigger:
○ Link the S3 bucket to the Lambda function by setting up a trigger. Specify that the function should be triggered when an object is created in the bucket (e.g., when an image is uploaded).

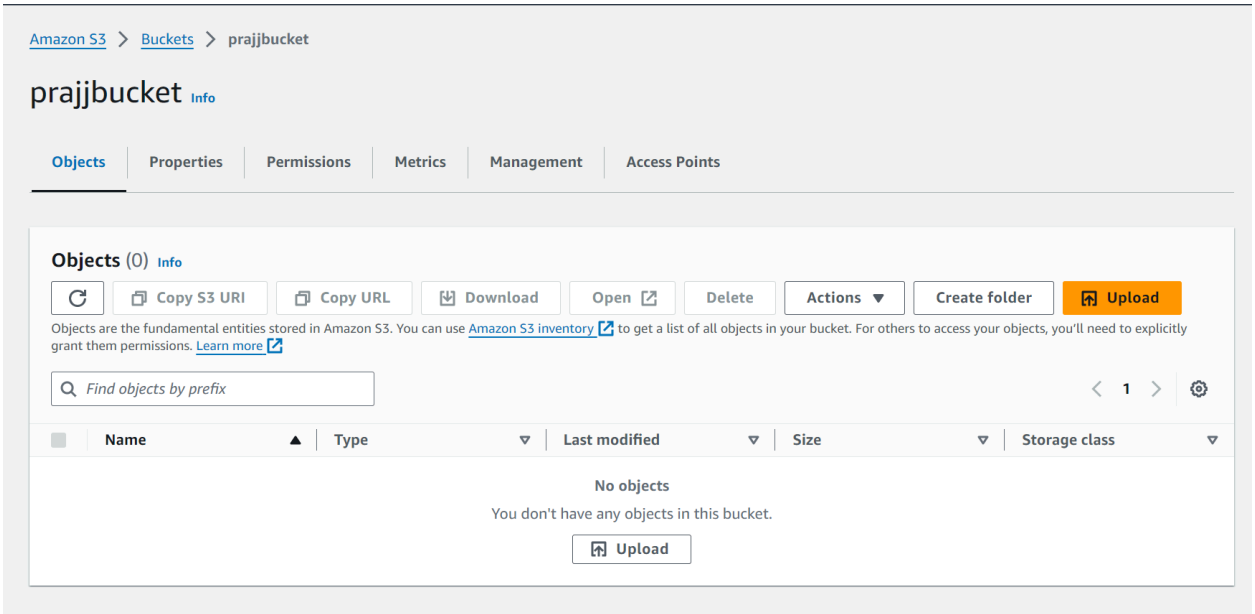
5. Test the Setup:
○ Upload an object (e.g., an image) to the S3 bucket to test the trigger. The Lambda function should execute and log the message “An Image has been added” in AWS CloudWatch Logs.

STEPS:-

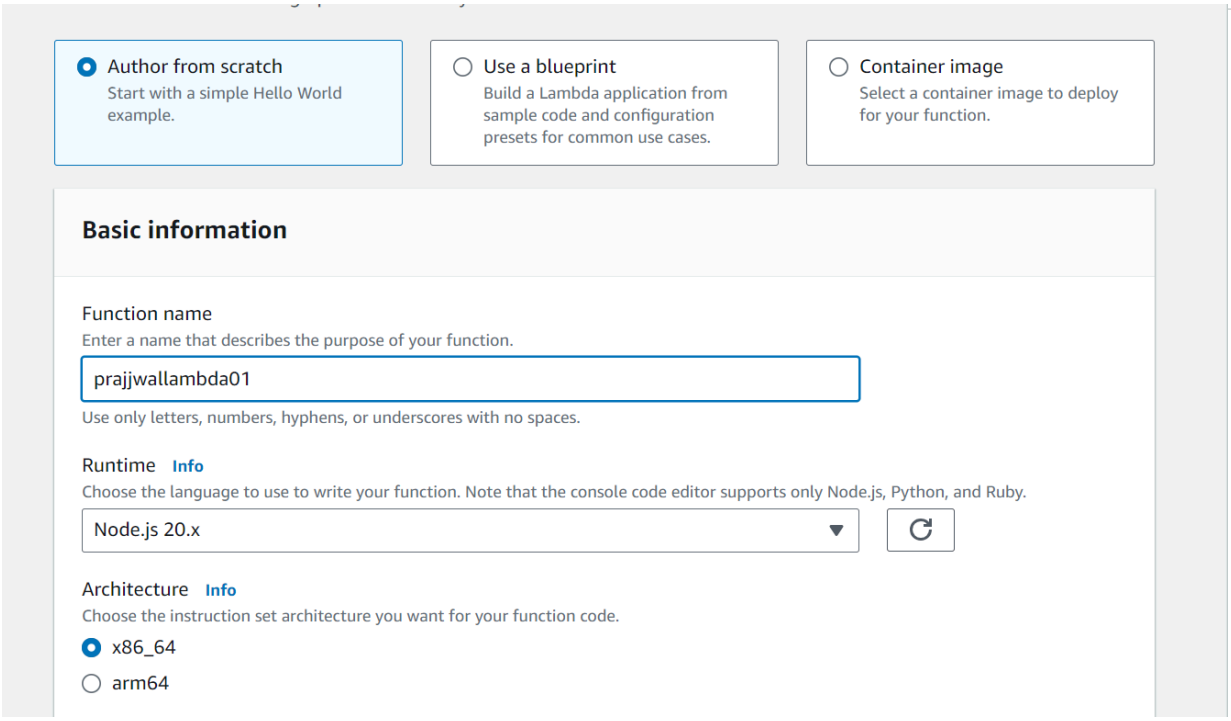
1. Create an S3 bucket of the same location as that of the Lambda function.

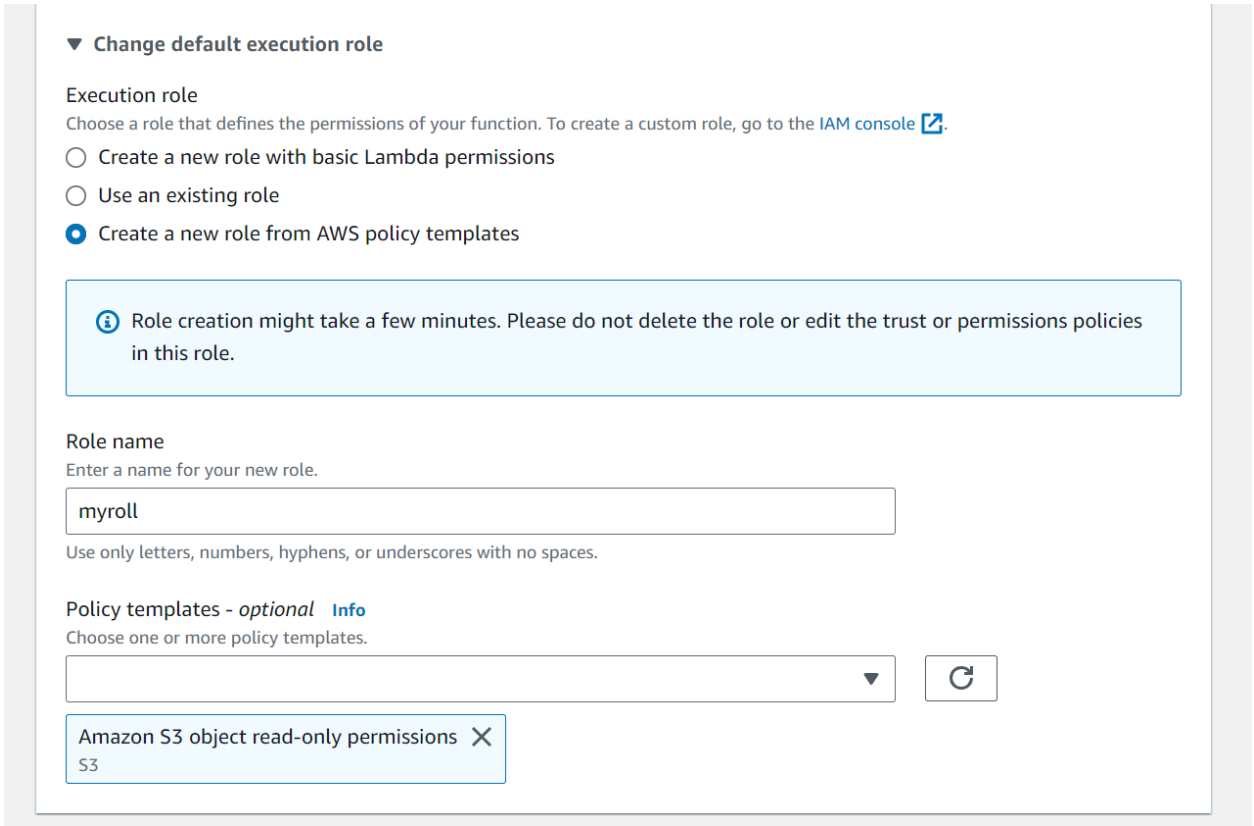


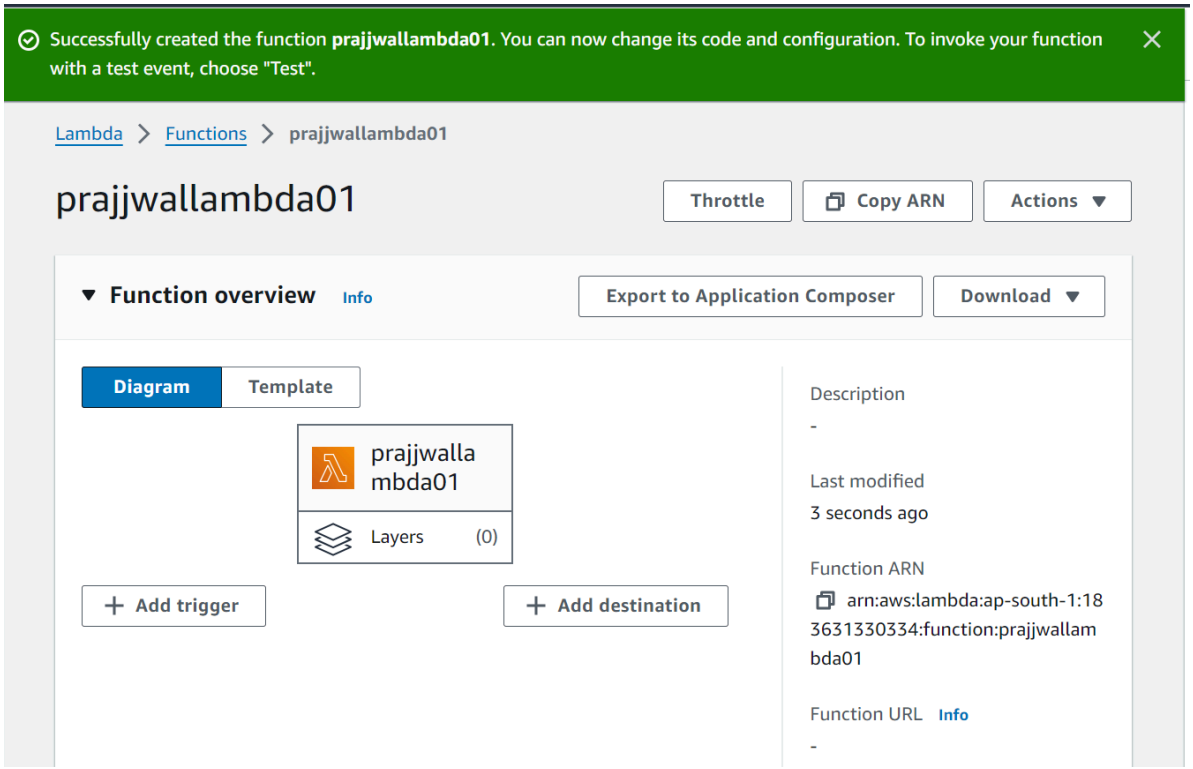




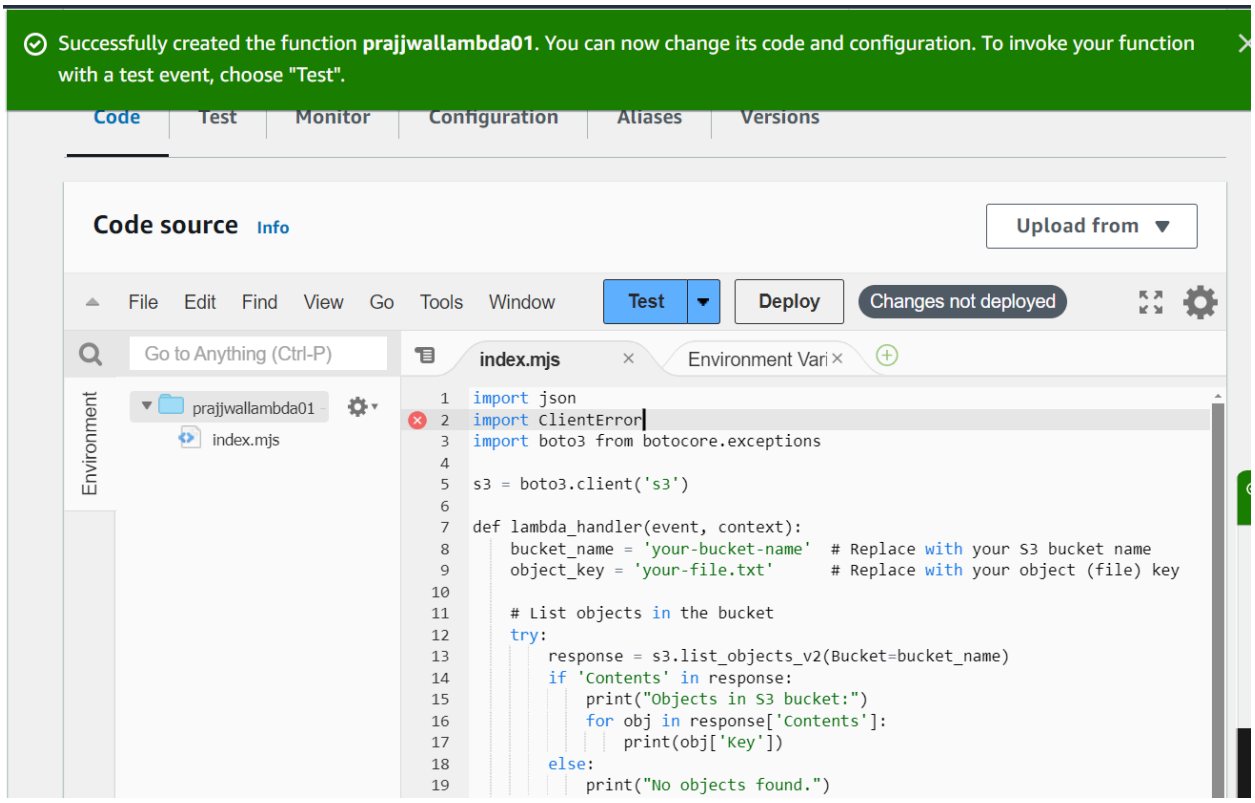
2. Add roles while creating the Lambda function and give permissions for accessing the S3 bucket.



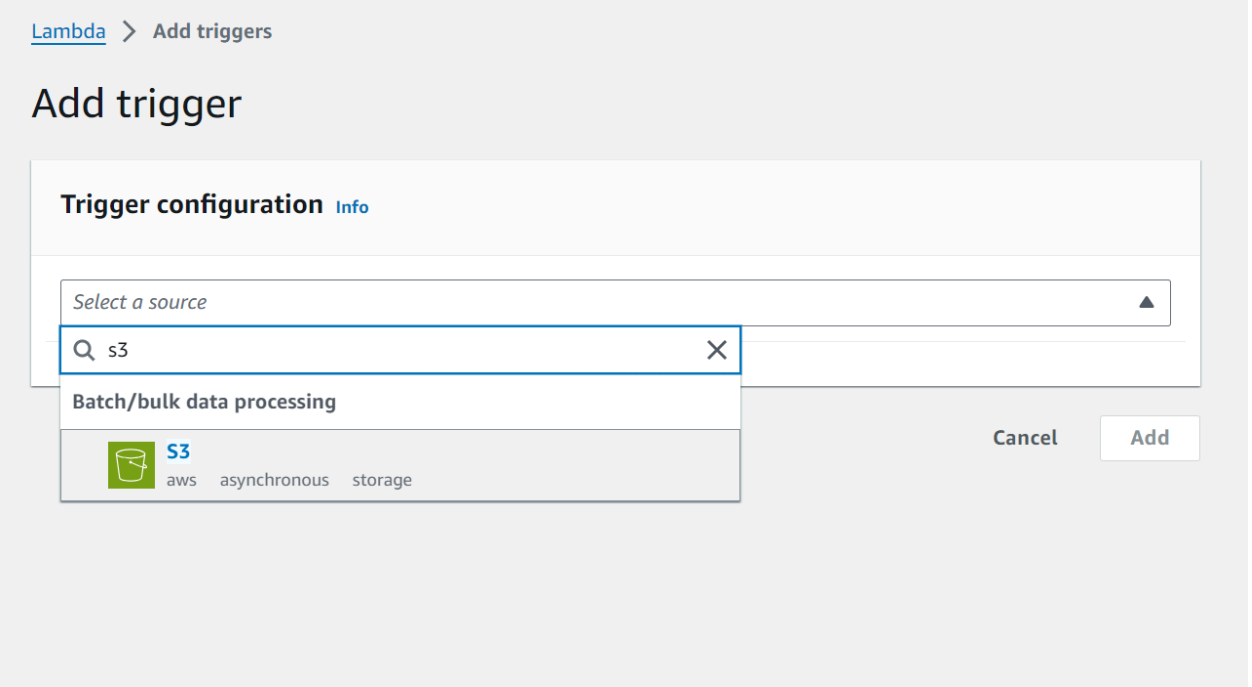

Change default execution role
Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.
Create a new role with basic Lambda permissions
Use an existing role
Create a new role from AWS policy templates
Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in this role.
Role name
Enter a name for your new role.
myroll
Use only letters, numbers, hyphens, or underscores with no spaces.
Policy templates - optional Info
Choose one or more policy templates.
Amazon S3 object read-only permissions
S3

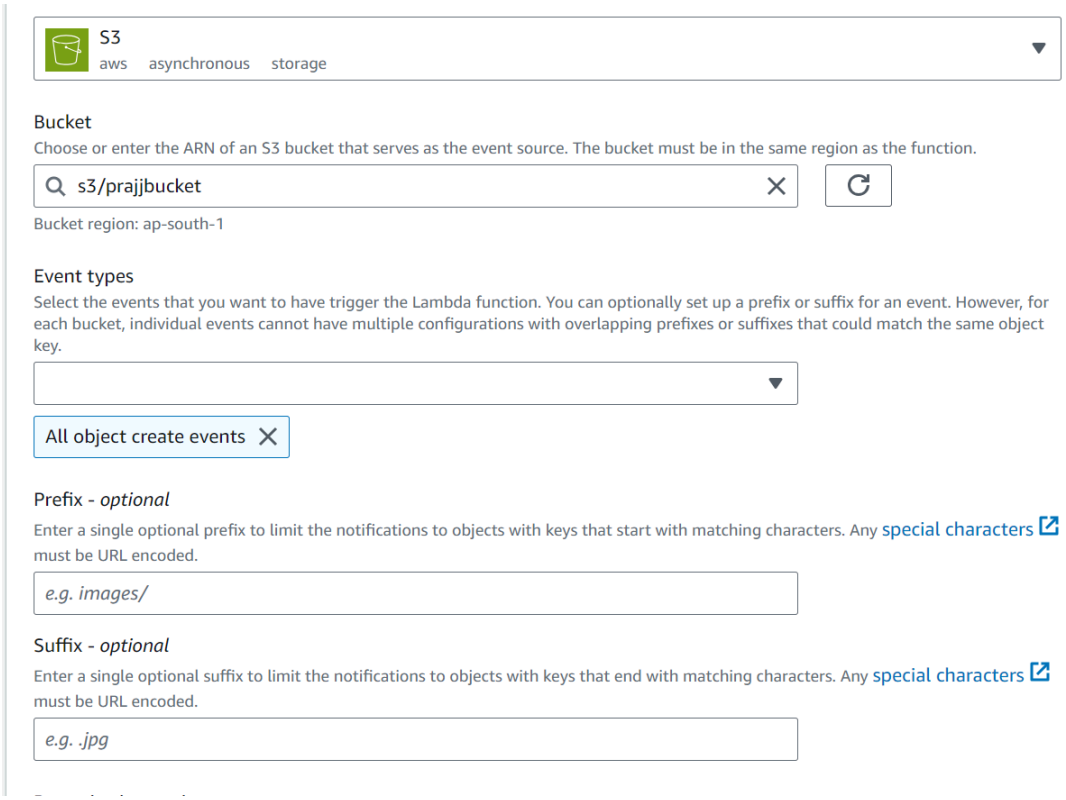

Successfully created the function prajjwallambda01. You can now change its code and configuration. To invoke your function with a test event, choose "Test".
Lambda > Functions > prajjwallambda01
prajjwallambda01
Throttle
Copy ARN
Actions
Function overview Info
Export to Application Composer
Download
Diagram
Template
prajjwallambda01
Layers (0)
Add trigger
Add destination
Description
-
Last modified
3 seconds ago
Function ARN
arn:aws:lambda:ap-south-1:183631330334:function:prajjwallambda01
Function URL Info
-

3. After creating the Lambda function copy a code available on the internet which allows the Lambda function to access the S3 bucket contents.

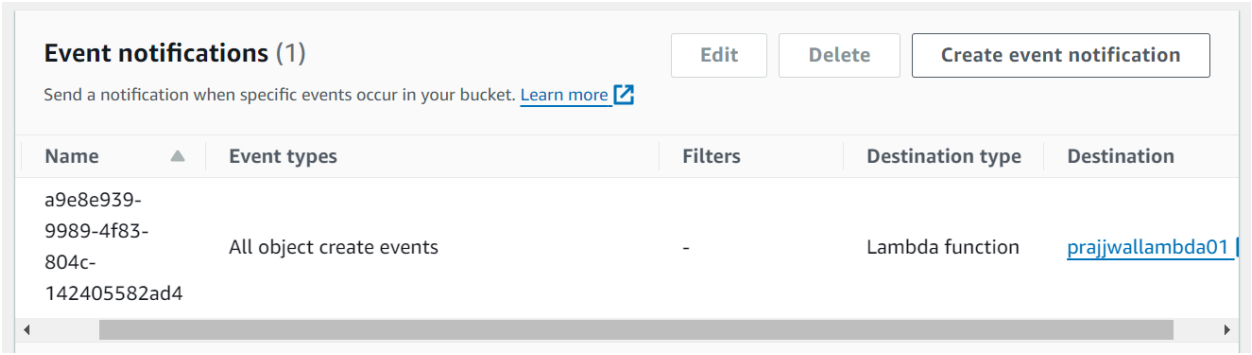


4. Add a trigger to the Lambda function so any changes in the S3 bucket will be first visible to the user.

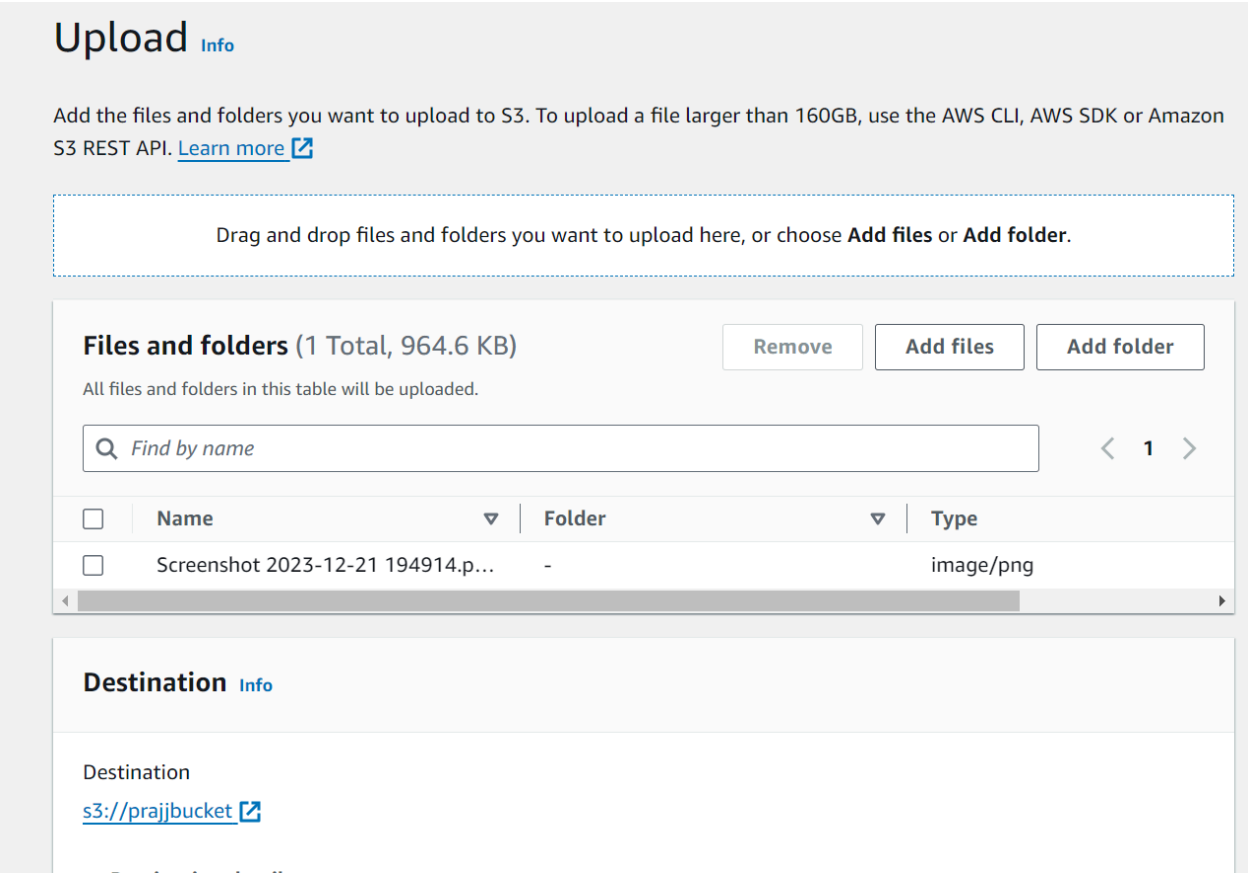


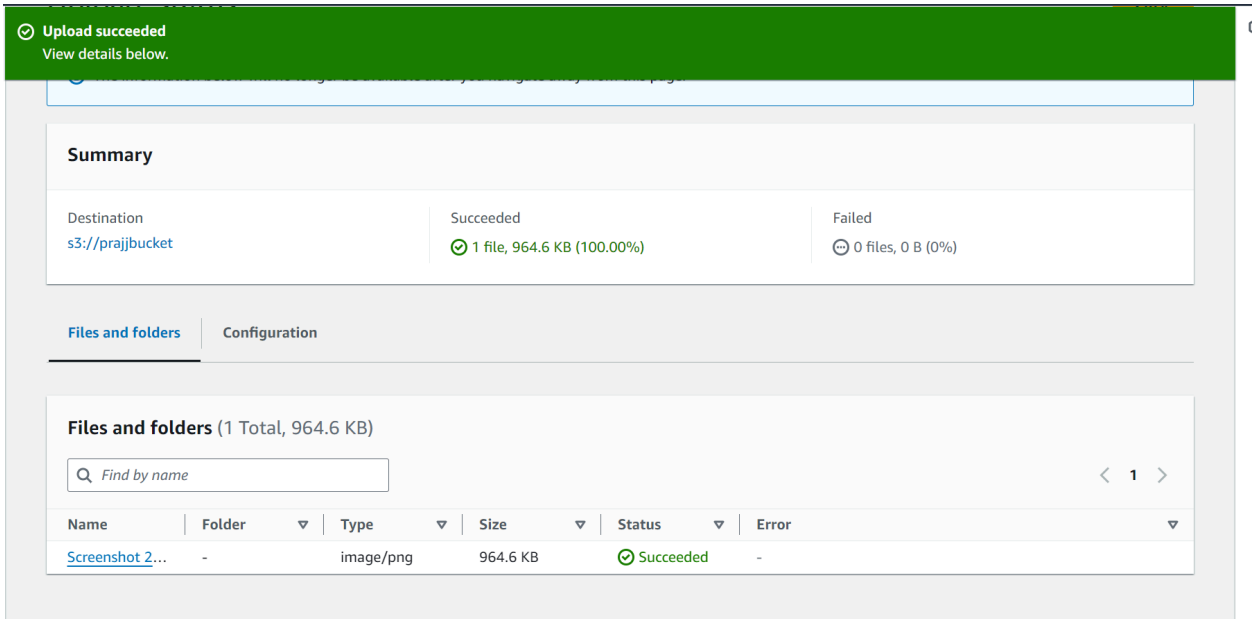


5. In the event notification of the S3 bucket we can see that it has been connected to the Lambda function .

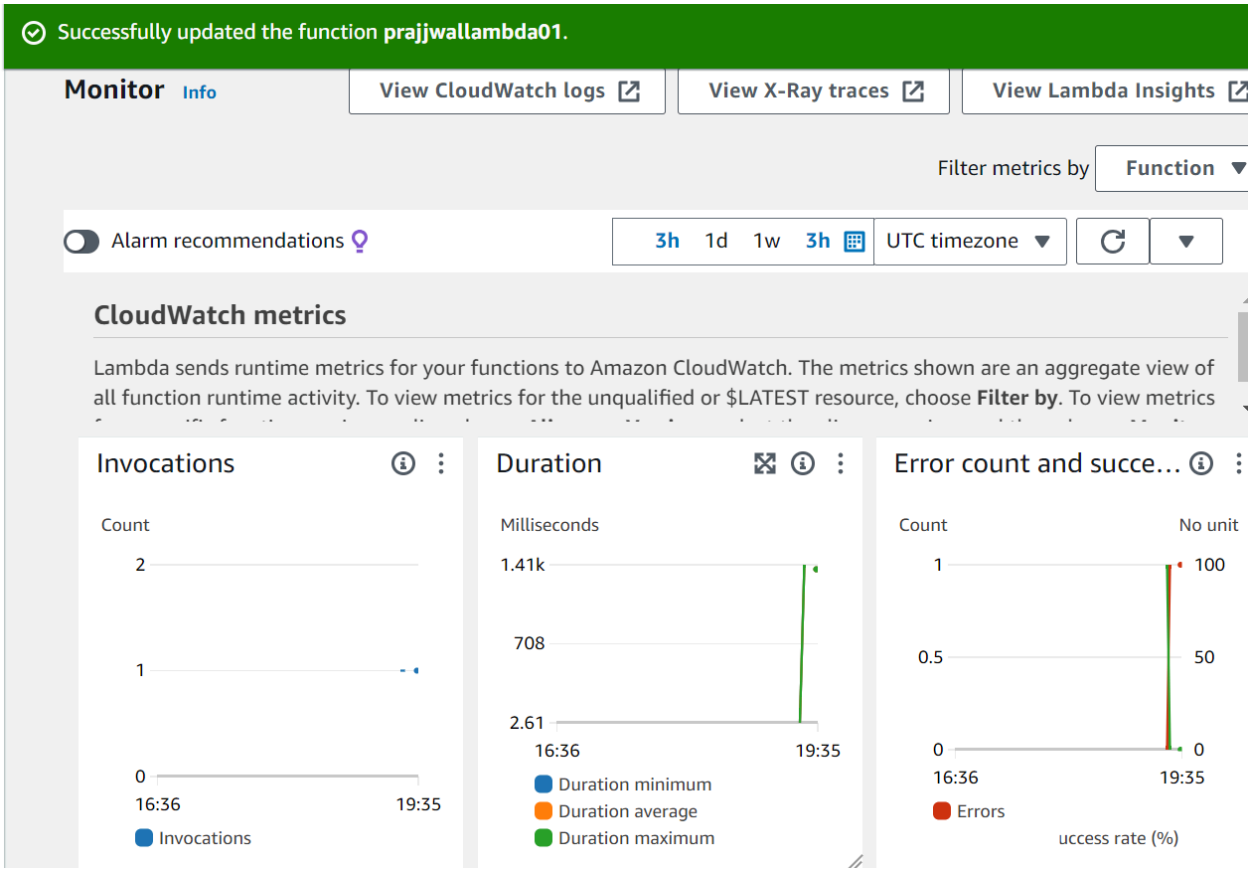


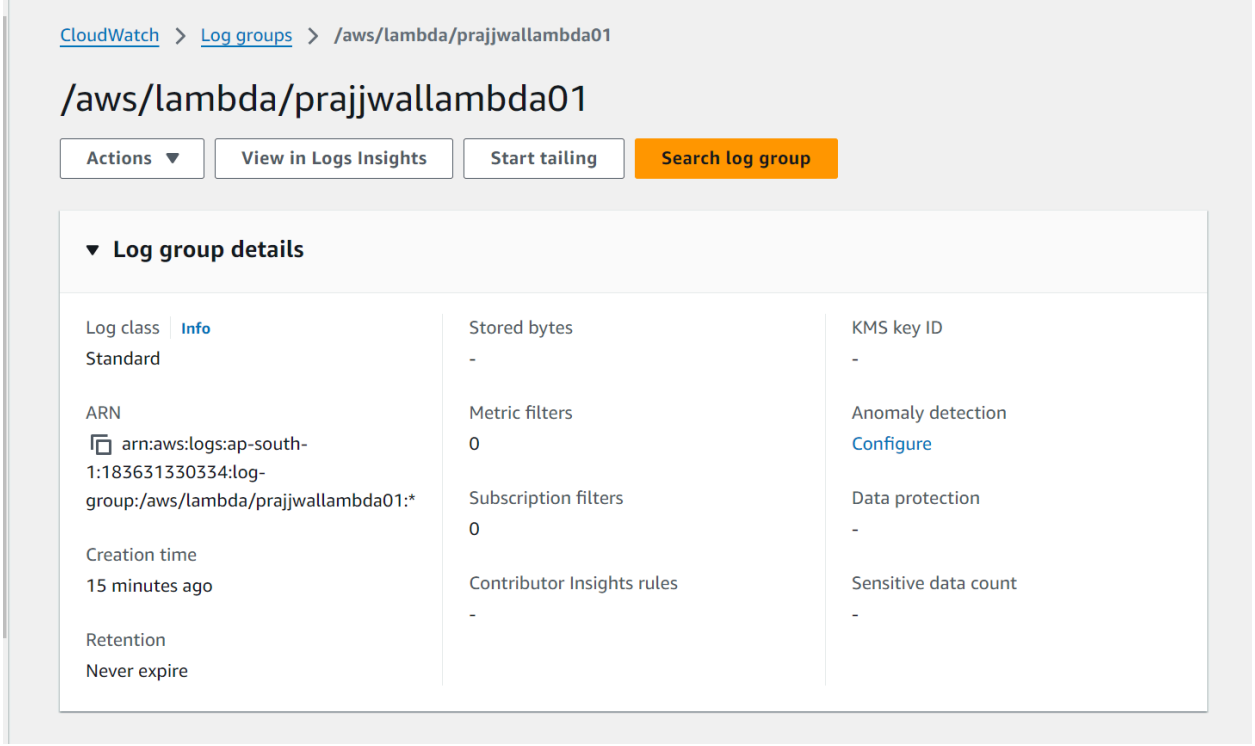
6. Upload a photo to the S3 bucket

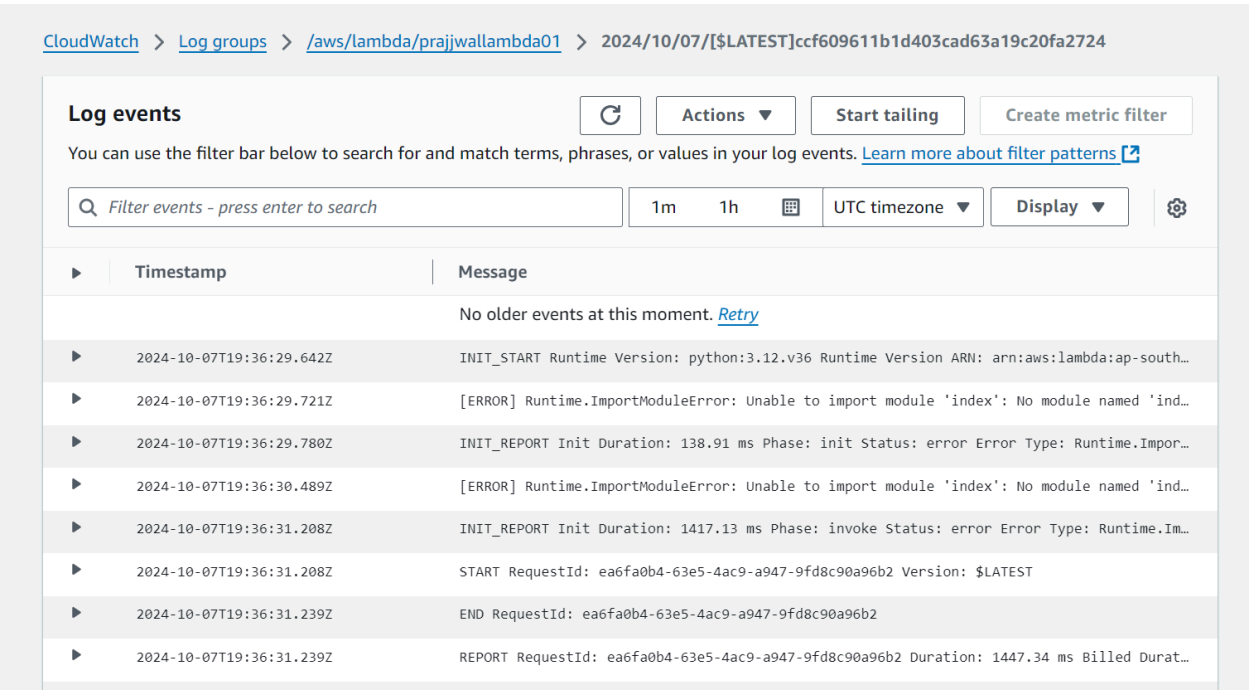




7. Now run the function and in the cloud watch logs of AWS you can see the message printed and all the other details of the working of the Lambda function.


Successfully updated the function prajjwallambda01.
Monitor Info
View CloudWatch logs
View X-Ray traces
View Lambda Insights
Filter metrics by
Function
Alarm recommendations
3h 1d 1w 3h
UTC timezone
CloudWatch metrics
Lambda sends runtime metrics for your functions to Amazon CloudWatch. The metrics shown are an aggregate view of all function runtime activity. To view metrics for the unqualified or $LATEST resource, choose Filter by. To view metrics
Invocations
Count
2
1
0
16:36
19:35
Invocations
Duration
Milliseconds
1.41k
708
2.61
16:36
19:35
Duration minimum
Duration average
Duration maximum
Error count and succe...
Count
No unit
1
0.5
0
100
50
0
16:36
19:35
Errors
uccess rate (%)


CloudWatch > Log groups > /aws/lambda/prajjwallambda01
/aws/lambda/prajjwallambda01
Actions
View in Logs Insights
Start tailing
Search log group
Log group details
Log class Info
Standard
ARN
arn:aws:logs:ap-south-1:183631330334:log-group:/aws/lambda/prajjwallambda01:*
Creation time
15 minutes ago
Retention
Never expire
Stored bytes
-
Metric filters
0
Subscription filters
0
Contributor Insights rules
-
KMS key ID
-
Anomaly detection
Configure
Data protection
-
Sensitive data count
-


CloudWatch > Log groups > /aws/lambda/prajjwallambda01 > 2024/10/07/[$LATEST]ccf609611b1d403cad63a19c20fa2724
Log events
Actions
Start tailing
Create metric filter
You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns
Filter events - press enter to search
1m
1h
UTC timezone
Display
Timestamp
Message
No older events at this moment. Retry
2024-10-07T19:36:29.642Z
INIT_START Runtime Version: python:3.12.v36 Runtime Version ARN: arn:aws:lambda:ap-south…
2024-10-07T19:36:29.721Z
[ERROR] Runtime.ImportModuleError: Unable to import module 'index': No module named 'ind…
2024-10-07T19:36:29.780Z
INIT_REPORT Init Duration: 138.91 ms Phase: init Status: error Error Type: Runtime.Impor…
2024-10-07T19:36:30.489Z
[ERROR] Runtime.ImportModuleError: Unable to import module 'index': No module named 'ind…
2024-10-07T19:36:31.208Z
INIT_REPORT Init Duration: 1417.13 ms Phase: invoke Status: error Error Type: Runtime.Im…
2024-10-07T19:36:31.208Z
START RequestId: ea6fa0b4-63e5-4ac9-a947-9fd8c90a96b2 Version: $LATEST
2024-10-07T19:36:31.239Z
END RequestId: ea6fa0b4-63e5-4ac9-a947-9fd8c90a96b2
2024-10-07T19:36:31.239Z
REPORT RequestId: ea6fa0b4-63e5-4ac9-a947-9fd8c90a96b2 Duration: 1447.34 ms Billed Durat…