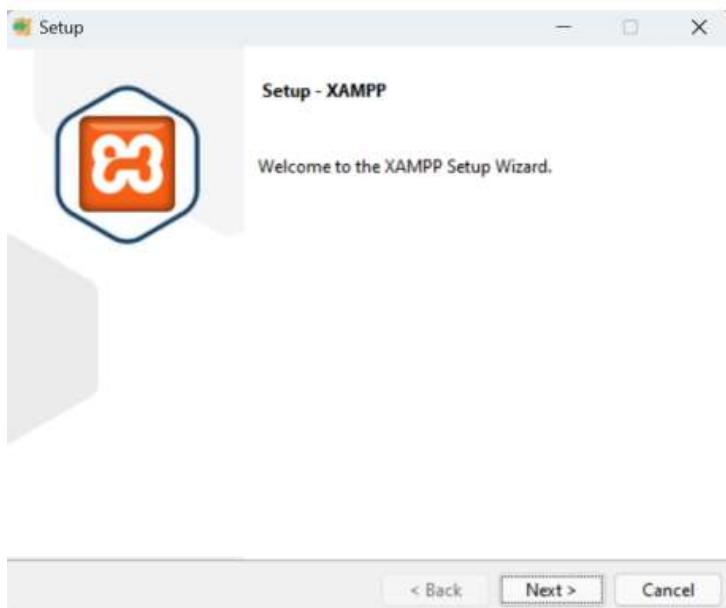


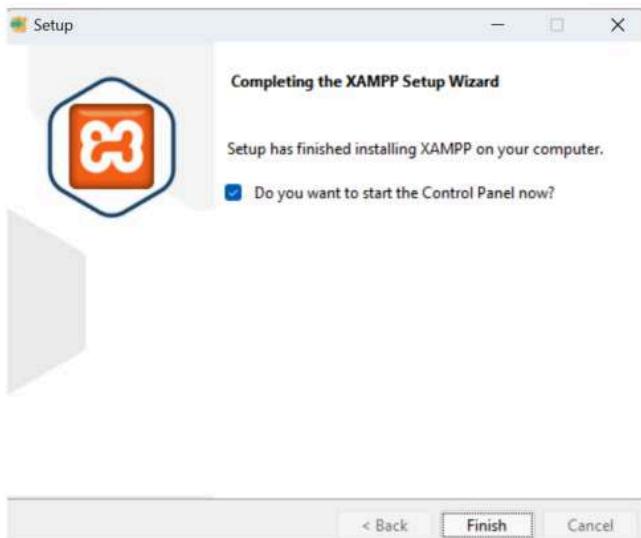
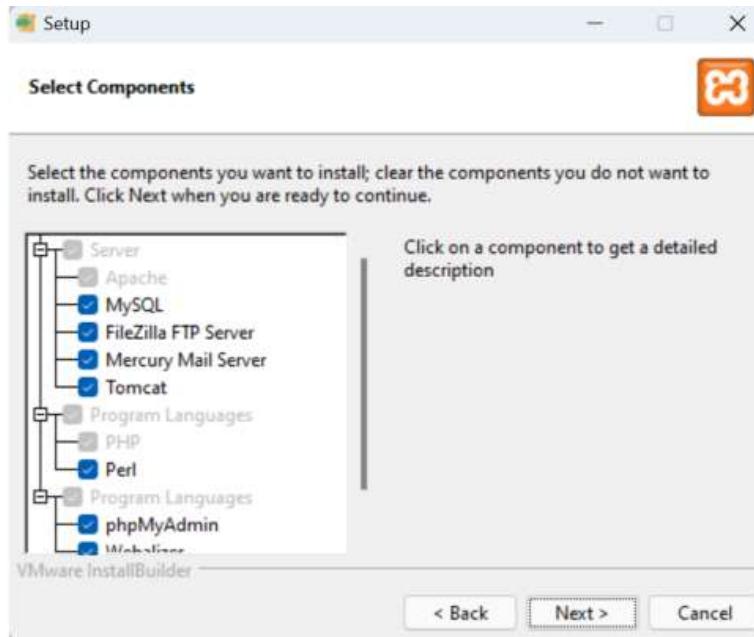
Aim: To develop a website and host it on your local machine and Amazon S3

1. Hosting Website using Xampp

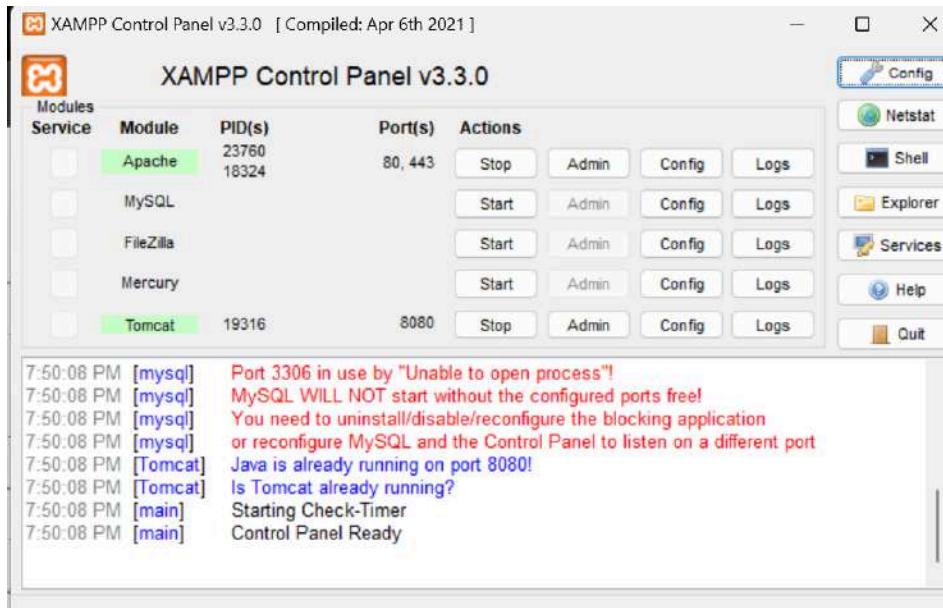
Install the required files from the internet and open up the setup



Make the changes according to your requirements

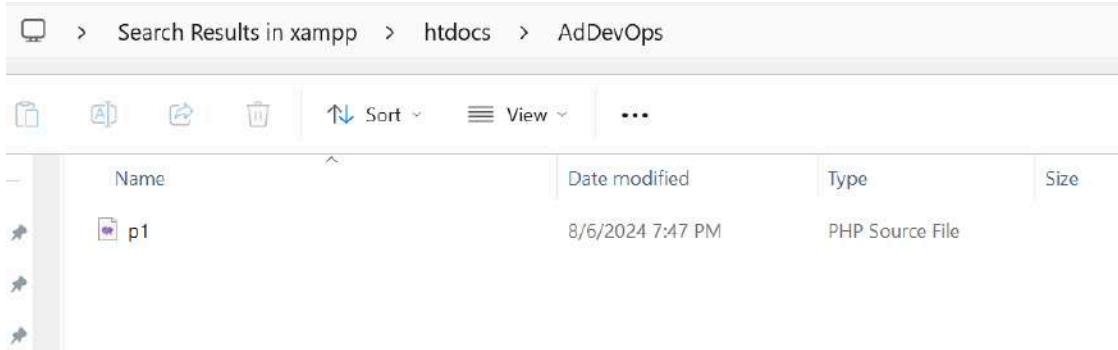


Open the control panel and click start for Apache



Write down your code and save it inside the htdocs folder in xampp files

```
C: > xampp > htdocs > AdDevOps > p1.php
1 <html>
2   <title> LAB 1 NIKITA CHHABRIA </title>
3   <body>
4     <?php
5     echo"D15C";
6     ?>
7
8
```



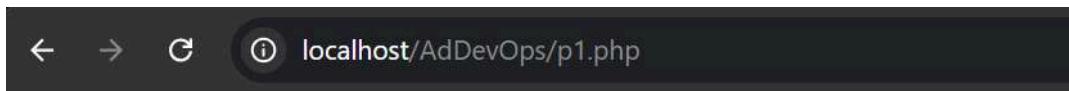
Go to your browser and type <localhost/filename/>



Index of /AdDevOps

Name	Last modified	Size	Description
Parent Directory		-	
p1.php	2024-08-06 19:47	82	

Apache/2.4.56 (Win64) OpenSSL/1.1.1t PHP/8.2.4 Server at localhost Port 80



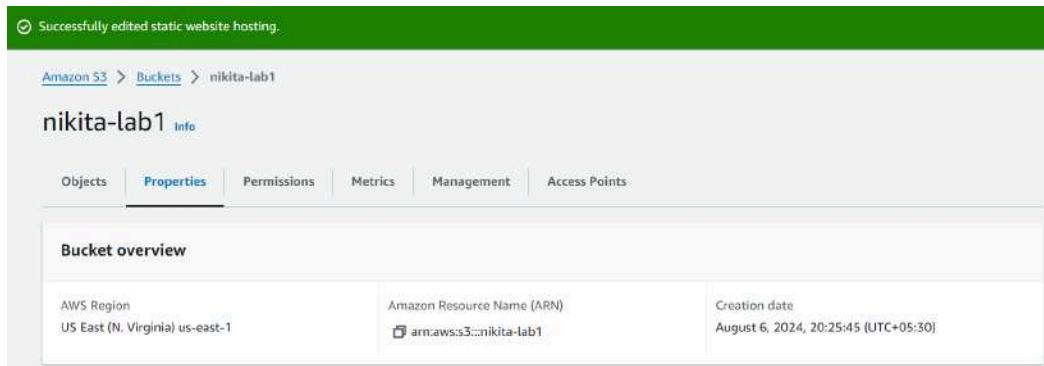
D15C

Hosting website with Amazon S3

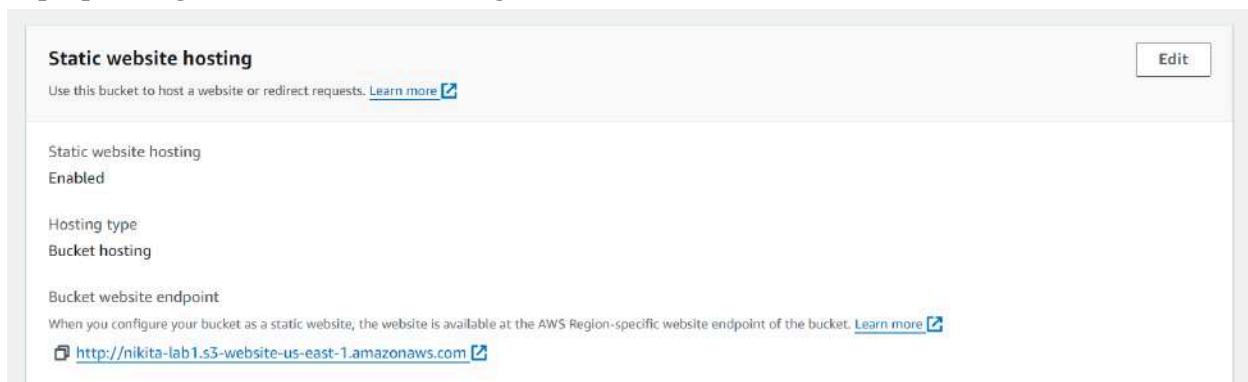
Open your learner lab's and Go to Amazon S3 in services



Click on create bucket and make a bucket with any name



In properties go to static website hosting and enable it



Go to Uploads option and upload the code files in them

The screenshot shows the AWS S3 console. At the top, a green header bar indicates "Upload succeeded" with a link to "View details below". Below this, a "Summary" section shows the destination as "s3://nikita-lab1" and two categories: "Succeeded" (2 files, 108.0 B (100.00%)) and "Failed" (0 files, 0 B (0%)). A navigation bar at the bottom has tabs for "Files and folders" (selected) and "Configuration". The main content area displays a table titled "Files and folders (2 Total, 108.0 B)". The table includes a search bar and pagination controls. The data is as follows:

Name	Folder	Type	Size	Status	Error
error.html	-	text/html	47.0 B	Succeeded	-
index.html	-	text/html	61.0 B	Succeeded	-

To host the website we need some permissions which can be changed using the following code

The screenshot shows the "Bucket policy" section of the AWS S3 console. It displays a JSON policy document with a "Copy" button. The policy is as follows:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::nikita-lab1/*"
    }
  ]
}
```

At the bottom, a browser-like interface shows the URL "nikita-lab1.s3.amazonaws.com/index.html".

lab 1



Aim: To understand the benefits of Cloud Infrastructure and Setup AWS Cloud9 IDE, Launch AWS Cloud9 IDE and Perform Collaboration Demonstration.

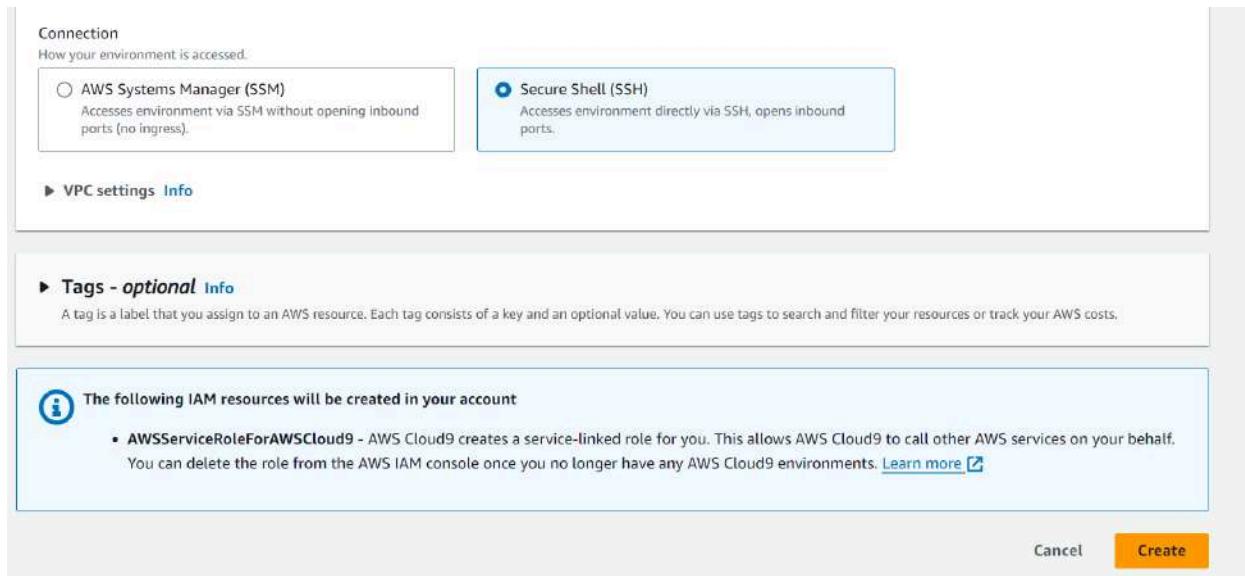
Login in to your Aws canva account and search for AWS Cloud9 in services

The screenshot shows the AWS Management Console with the AWS Cloud9 service selected. The top navigation bar includes 'Services', 'Search', and 'N. Virginia'. A banner at the top right provides information about AWS Toolkits and CloudShell. Below the banner, the breadcrumb navigation shows 'AWS Cloud9 > Environments > Create environment'. The main section is titled 'Create environment' with a 'Details' tab selected. Under 'Name', the value 'WebAppIDE' is entered. There is a note about character limits and uniqueness. The 'Description - optional' field is empty. The 'Environment type' section is expanded, showing two options: 'New EC2 instance' (selected) and 'Existing compute'. The 'New EC2 instance' section details the choice of 't2.micro' instance type, noting it's free-tier eligible and ideal for educational users. Other options like 't3.small' and 'm5.large' are also listed. The 'Existing compute' section notes that an existing instance or server can be used.

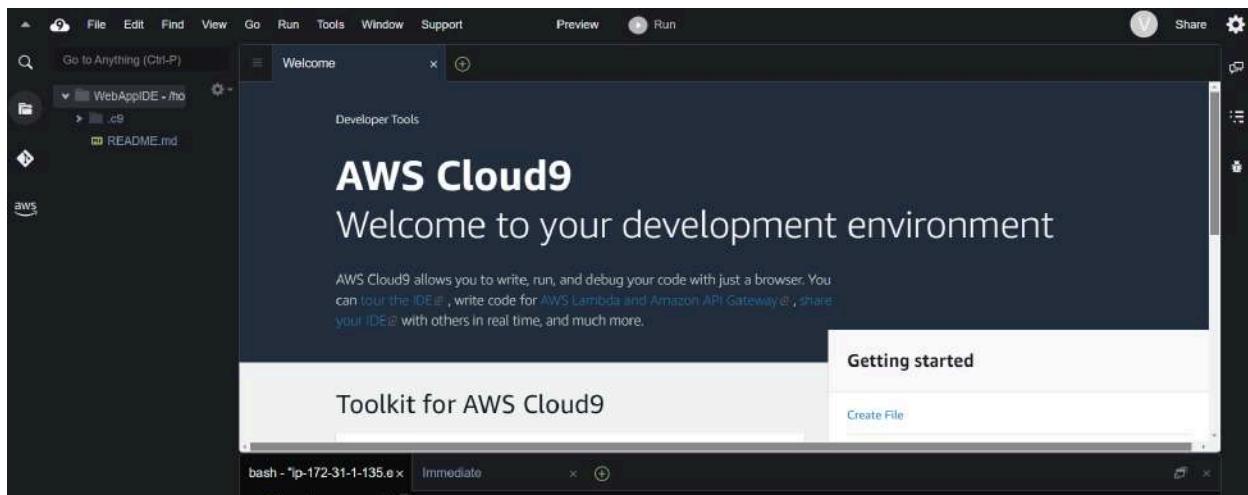
Select Environment type to be EC2

This screenshot shows the 'New EC2 instance' configuration step. It starts with a general note about determining what the Cloud9 IDE will run on. Two options are shown: 'New EC2 instance' (selected) and 'Existing compute'. The 'New EC2 instance' section then asks for the 'Instance type'. Three options are listed: 't2.micro (1 GiB RAM + 1 vCPU)' (selected), 't3.small (2 GiB RAM + 2 vCPU)', and 'm5.large (8 GiB RAM + 2 vCPU)'. Each option has a brief description. At the bottom, there is a link to 'Additional instance types'.

A few other options depending on your use case and then **Create environment**



The environment is ready



Now, to enable collaboration in cloud environment, we need to add a user group. Only IAM users can be part of this group.

Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Permissions options

- Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
- Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.
- Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Identity and Access Management (IAM)

IAM > User groups > Create user group

Create user group

Name the group

User group name
Enter a meaningful name to identify this group.

Group created without any hassle

Identity and Access Management (IAM)

IAM > User groups

group1 user group created.

User groups (1) [Info](#)

A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.

Group name	Users	Permissions	Creation time
group1	0	Not defined	Now

User name
nikita

Provide user access to the AWS Management Console - optional
If you're providing console access to a person, it's a best practice [to manage their access in IAM Identity Center.](#)

Are you providing console access to a person?

User type
 Specify a user in Identity Center - Recommended
We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.
 I want to create an IAM user
We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keypairs, or a backup credential for emergency account access.

Console password
 Autogenerated password
You can view the password after you create the user.
 Custom password
Enter a custom password for the user.

Specify user details: Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Permissions options

Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.

Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

User groups (1/1)

Group name	Users	Attached policies	Created
group1	0	-	2024-08-09 (2...)

Now for creating an IAM user, you need to create an account as in the academy you don't have the access to create one.

User created successfully

You can view and download the user's password and email instructions for signing in to the AWS Management Console.

IAM > Users > Create user

Step 1 Specify user details

Step 2 Set permissions

Step 3 Review and create

Step 4 Retrieve password

Retrieve password

You can view and download the user's password below or email users instructions for signing in to the AWS Management Console. This is the only time you can view and download this password.

Console sign-in details		Email sign-in instructions
Console sign-in URL	https://026090558619.signin.aws.amazon.com/console	
User name	nikita	
Console password	***** Show	

Add policies as per your requirement> I've added AWSCloud9 services

Identity and Access Management (IAM)

Policies (1222) Info

A policy is an object in AWS that defines permissions.

Filter by Type

Policy name	Type	Used as	Description
AWSCloud9Admin...	AWS managed	None	Provides administrator access to AWS ...
AWSCloud9Enviro...	AWS managed	None	Provides the ability to be invited into A...
AWSCloud9Service...	AWS managed	None	Service Linked Role Policy for AWS Clo...
AWSCloud9SSMIn...	AWS managed	None	This policy will be used to attach a rol...
AWSCloud9User	AWS managed	None	Provides permission to create AWS Clo...

Cloud9User

Cloud9User permission to create AWS Cloud9 development environments and to manage owned environments.

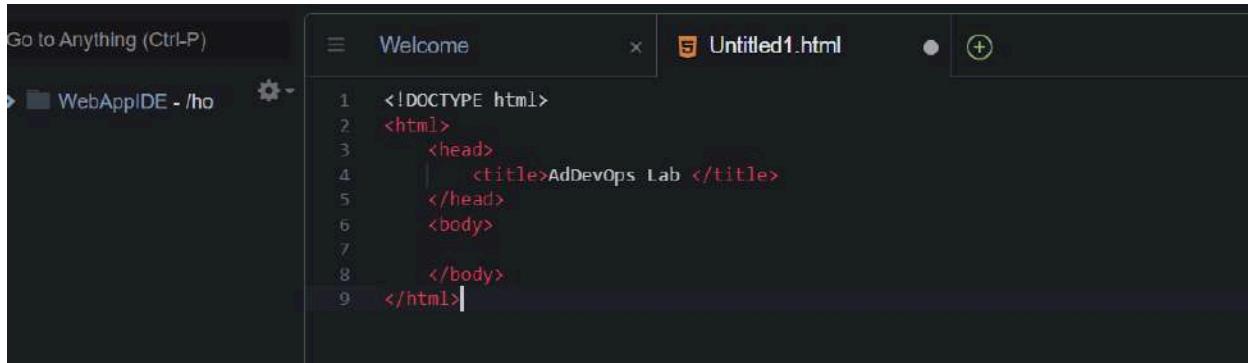
[Copy JSON](#)

Click on create and done

The screenshot shows the 'Attach permissions policies' dialog for a user named 'nikita'. At the top, there's a table with columns: 'User name' (with a checkbox), 'Groups', 'Last activity', and 'Creation time'. Below the table, a message says: 'You can attach up to 10 policies to this user group. All the users in this group will have permissions that are defined in the selected policies.' A 'Filter by Type' search bar contains 'cloud9' and shows 'All types' with '4 matches'. The main list table has columns: 'Policy name', 'Type', 'Used as', and 'Description'. It lists four AWS managed policies: 'AWSCloud9Admin...', 'AWSCloud9Enviro...', 'AWSCloud9SSMIn...', and 'AWSCloud9User'. The 'AWSCloud9User' policy is selected, indicated by a checked checkbox. At the bottom right are 'Cancel' and 'Create user group' buttons.

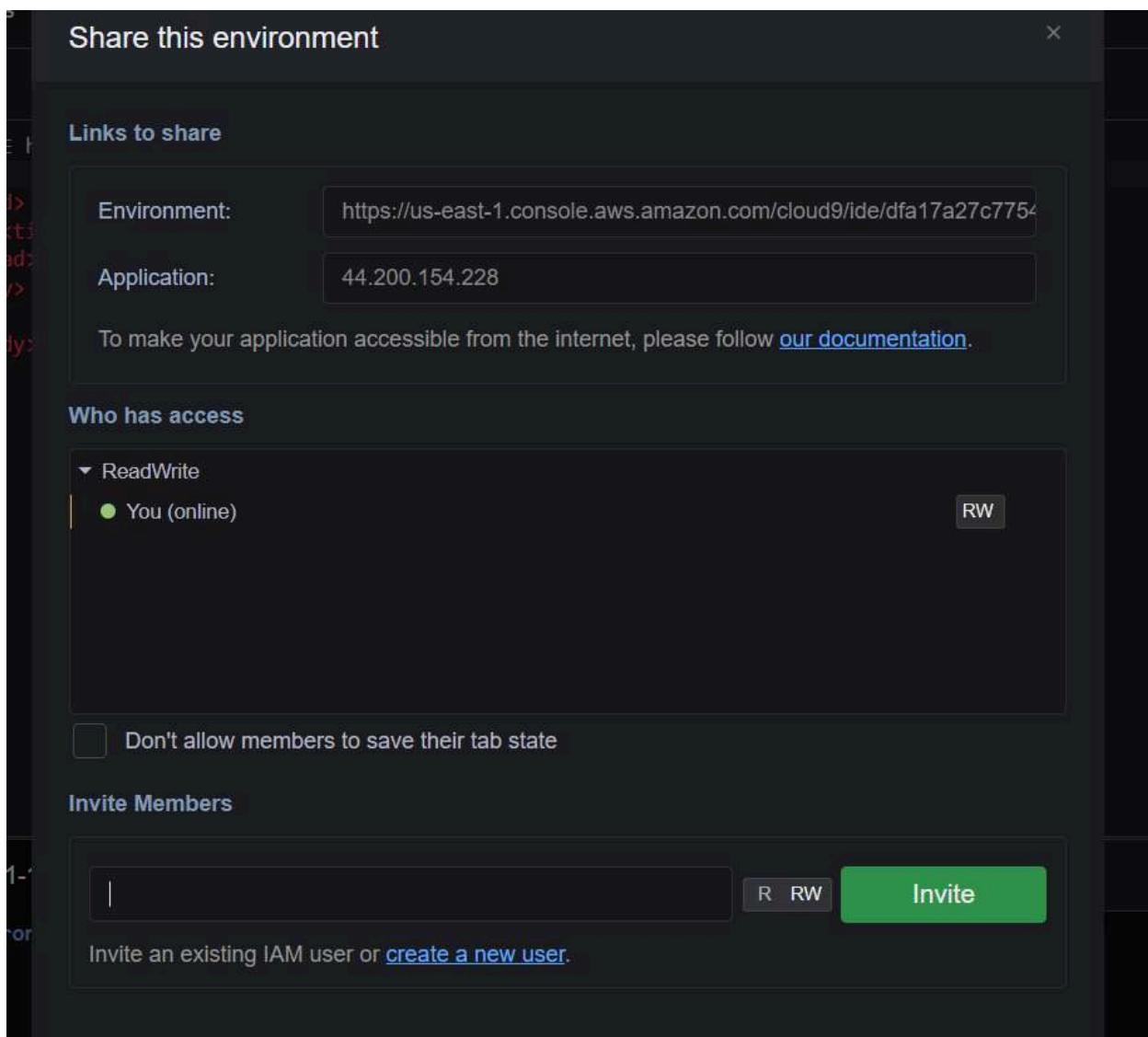
Now, go back to the academy account and deploy your code

The screenshot shows the AWS Cloud9 IDE interface. The top navigation bar includes File, Edit, Find, View, Go, Run, Tools, Window, Support, Preview, and Run buttons. The 'File' menu is open, showing options like New File, New From Template, Open, Open Recent, Save, Save As, Revert to Saved, Revert All to Saved, Show File Revision History, Upload Local Files, Download Project, Line Endings, Close File, and Close All Files. A context menu for 'New From Template' is open, showing options: Text File, JavaScript File, HTML File (which is selected), XML File, Python File, PHP File, C File, C++ File, Go File, Markdown, Node.js Web Server, and Java Console Application. On the right side, there's a 'Configure AWS Cloud9' sidebar with sections for Main Theme (set to 'jett-dark'), Editor Theme (set to 'Jett'), and Keyboard Mode (set to 'Default'). A 'More Settings...' link is also present.



```
curl -X POST https://us-east-1.console.aws.amazon.com/cloud9/ide/dfa17a27c7754
```

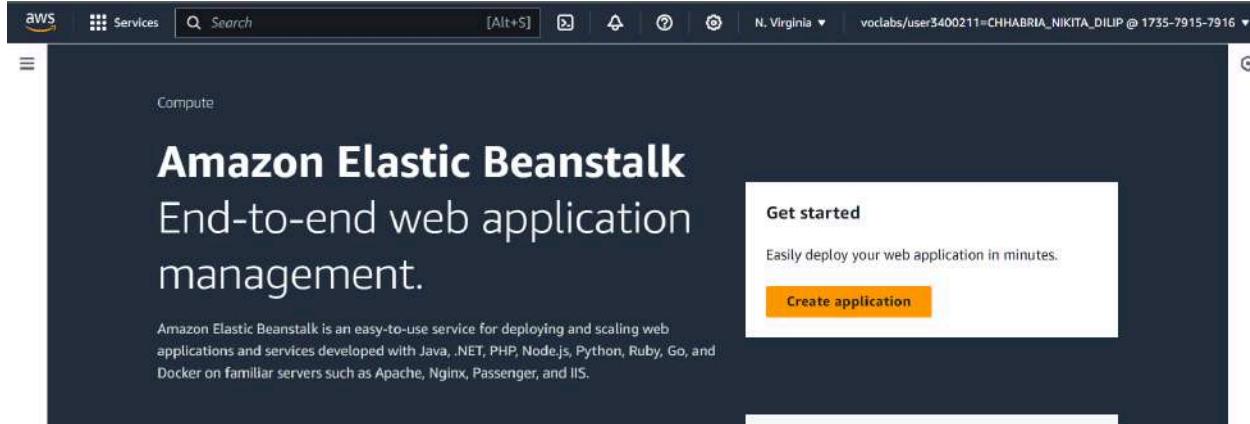
Click on share this environment and update the <who can access> section



And done!

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

Go to your amazon academy account and search for <Elastic Beanstalk>



Click on create application and configure the environment

A screenshot of the 'Configure environment' step in the AWS Elastic Beanstalk creation wizard. On the left, a sidebar lists steps: Step 1 (Configure environment), Step 2 (Configure service access), Step 3 (optional: Set up networking, database, and tags), Step 4 (optional: Configure instance traffic and scaling), Step 5 (optional: Configure updates, monitoring, and logging), and Step 6 (Review). The main area shows the 'Configure environment' step. It has two sections: 'Environment tier' and 'Application information'. Under 'Environment tier', it says 'Web server environment' is selected. Under 'Application information', the 'Application name' field contains 'FirstWebApp'. There's also a section for 'Application tags (optional)'.

Now, manage the platform type as shown below

Platform Info

Platform type

Managed platform
Platforms published and maintained by Amazon Elastic Beanstalk. [Learn more](#)

Custom platform
Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform

PHP

Platform branch

PHP 8.3 running on 64bit Amazon Linux 2023

Platform version

4.3.1 (Recommended)

A few more customizations

Service access

IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. [Learn more](#)

Service role

Create and use new service role
 Use an existing service role

Existing service roles

Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.

EMR_EC2_DefaultRole

EC2 key pair

Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

vockey

EC2 instance profile

Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

EMR_EC2_DefaultRole

[View permission details](#)

And environment created successfully!

The screenshot shows the AWS Elastic Beanstalk console. On the left, there's a sidebar with options like Applications, Environments, Change history, Application: FirstWebApp (Application versions, Saved configurations), and Environment: FirstWebApp-env (Go to environment). The main area is titled 'FirstWebApp-env' and shows an 'Environment successfully launched.' message. It includes sections for Environment overview, Health (with a warning icon), Domain (FirstWebApp-env.eba-esxry2an.us-east-1.elasticbeanstalk.com), Environment ID (e-jdhrq9uum), Application name (FirstWebApp), and Actions (Upload and deploy).

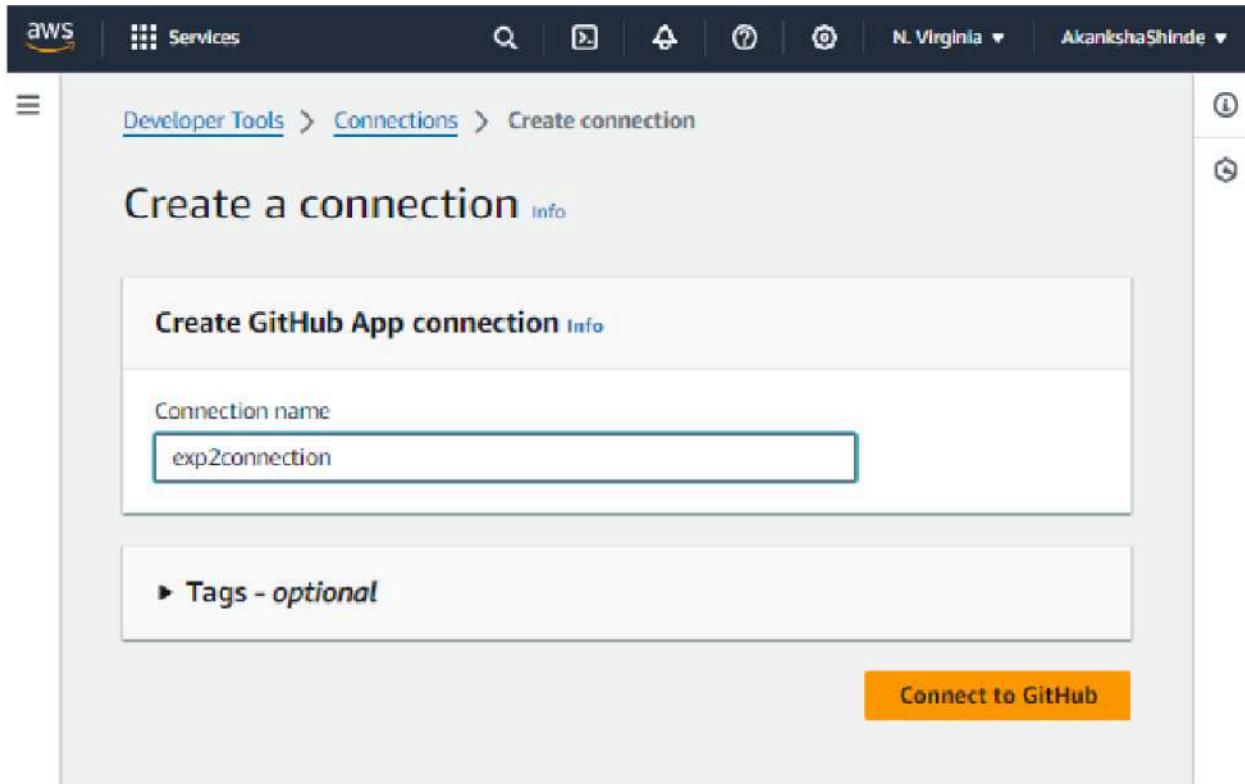
Now, go to github > <https://github.com/aws-samples/aws-codepipeline-s3-codedeploy-linux>

The screenshot shows a GitHub repository page for 'aws-codepipeline-s3-codedeploy-linux'. The repository is public and has 30 forks, 5.6k stars, and 159 issues. The page includes standard GitHub navigation like Code, Issues, Pull requests, Actions, Projects, Security, and Insights.

Fork this Repository and come back to the account and go to code pipeline

The screenshot shows the 'Choose pipeline settings' step of the AWS CodePipeline setup wizard. The pipeline name is set to 'pipeline1'. The pipeline type is 'Queued (Pipeline type V2 required)'. The execution mode is 'Superseded'. Under Service role, the 'New service role' option is selected. A note indicates that V1 pipelines cannot be created through the console and recommends using V2 pipeline type.

Follow through all the steps given



2. In the source stage, choose GitHub v2 as the provider, then connect your GitHub account to AWS by creating a connection. You'd need your GitHub credentials and then you'd need to authorize and install AWS on the forked GitHub Repository.

Source

Source provider
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.

GitHub (Version 2) ▾

New GitHub version 2 (app-based) action

To add a GitHub version 2 action in CodePipeline, you create a connection, which uses GitHub Apps to access your repository. Use the options below to choose an existing connection or create a new one. [Learn more](#)

Connection
Choose an existing connection that you have already configured, or create a new one and then return to this task.

Q arnaws:codeconnections:us-east-1:010928214902:connection/511a6a10-63 X or [Connect to GitHub](#)

Ready to connect
Your GitHub connection is ready for use.

Repository name
Choose a repository in your GitHub account.

Q

Trigger

Trigger type
Choose the trigger type that starts your pipeline.

No filter
Starts your pipeline on any push and clones the HEAD.

Specify filter
Starts your pipeline on a specific filter and clones the exact commit. Pipeline type V2 is required.

Do not detect changes
Don't automatically trigger the pipeline.

You can add additional sources and triggers by editing the pipeline after it is created.

[Cancel](#) [Previous](#) [Next](#)

3. Then, simply choose this forked repository and the branch which you will be able to find in the search box. After that, click Continue and skip the build stage. Proceed to the Deployment stage.

Add build stage Info

Step 3 of 5

Build - optional

Build provider
This is the tool of your build project. Provide build artifact details like operating system, build spec file, and output file names.

▼

Cancel Previous Skip build stage **Next**

Review all the settings

Review Info

Step 5 of 5

Step 1: Choose pipeline settings

Pipeline settings

Pipeline name: firstPipeline

Pipeline type: V2

Execution mode: QUEUED

Artifact location: A new Amazon S3 bucket will be created as the default artifact store for your pipeline

Service role name: AWSCodePipelineServiceRole-us-east-1-firstPipeline

Variables

The pipeline is created successfully

The screenshot shows the AWS CodePipeline Create Pipeline wizard. It is currently on Step 4: Add deploy stage. The interface includes a sidebar with navigation links like 'Create pipeline', 'My pipelines', 'Actions', 'Lambda functions', and 'AWS Lambda'. The main area displays configuration options for a deploy stage:

Step 4: Add deploy stage

Deploy action provider

- Deploy action provider: AWS Elastic Beanstalk
- ApplicationName: firstWebApp
- EnvironmentName: FirstWebApp-env
- Configure automatic rollback on stage failure: Disabled

At the bottom right are buttons for 'Cancel', 'Previous', and a prominent orange 'Create pipeline' button.

Now, you will be able to see the code deployed

The screenshot shows the AWS CodePipeline interface for a pipeline named "pipeline1". The pipeline type is V2 and the execution mode is QUEUED. The execution ID is [9c0d9b58-b454-4c61-b04e-90d7021f06fa](#). The pipeline consists of two stages: "Source" and "Deploy".

- Source Stage:** Status: Succeeded. Sub-steps include GitHub (Version 2) (Succeeded), S3 (Succeeded - 1 minute ago), and Artifacts (Succeeded). A "View details" button is available.
- Deploy Stage:** Status: Succeeded. Sub-step is AWS Elastic Beanstalk (Succeeded). A "Disable transition" button is available. A "Start rollback" button is located to the right of the stage.

Now we can see the URL provided for our application, we can simply click on it to lead us to our app

The screenshot shows the AWS Elastic Beanstalk interface for environments. There are two environments listed:

Environment name	Health	Application name	Platform	Domain	Running versions	Tier name	Date created
FirstApp-env	Ok	firstApp	PHP 8.1 running on...	FirstApp-env.eba-n34wyhg.us...	code-pipeline-1723...	WebServer	August 14, 2024 22
MyfirstWebApp-env	Unknown	myfirstWebApp	PHP 8.1 running on...	-	-	WebServer	August 14, 2024 22

Congratulations!

You have successfully created a pipeline that retrieved this source application from an Amazon S3 bucket and deployed it to three Amazon EC2 instances using AWS CodeDeploy.

For next steps, read the AWS CodePipeline Documentation. Incedge 2020

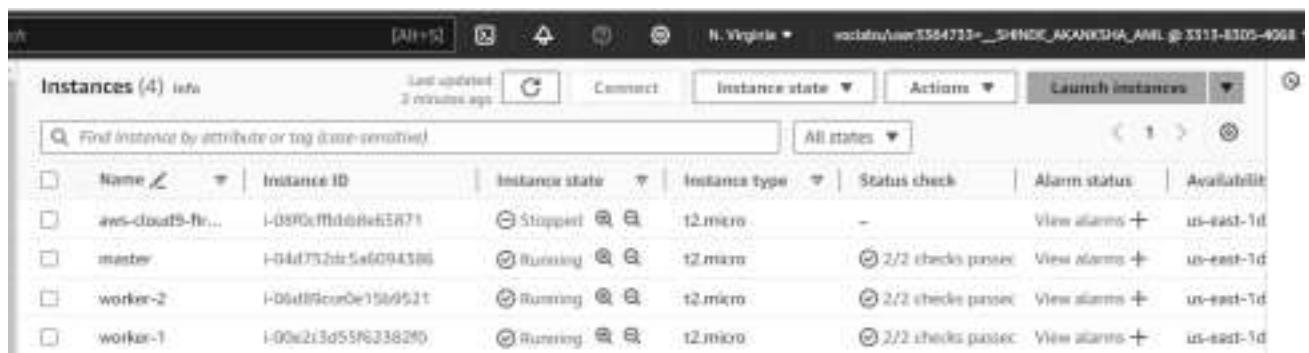
Experiment: 3

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

Steps:

1. Create 3 EC2 Ubuntu Instances on AWS.

(Name 1 as Master, the other 2 as worker-1 and worker-2)



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability zone
aws-cloud9-fn...	i-08f0cffdbbe65871	Stopped	t2.micro	-	View alarms +	us-east-1d
master	i-04d732bc5a6094586	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d
worker-2	i-00a09cc0e1509521	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d
worker-1	i-00e213d55f82382f0	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d

2. Edit the Security Group Inbound Rules to allow SSH and do it for all the three machines.

```
Akanksha.Shinde@AkankshaShinde MINGW64 ~/Downloads (master)
$ ssh -i "server.pem" ec2-user@ec2-54-174-206-93.compute-1.amazonaws.com
The authenticity of host 'ec2-54-174-206-93.compute-1.amazonaws.com (54.174.206.93)' can't be established.
ED25519 key fingerprint is SHA256:T+tsGyI11gAvUvjeAZ7GjDIWXHOaI4EPF5g5oICrk0Q.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-174-206-93.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

#_ _ _ _ _          Amazon Linux 2023
-- \##\           https://aws.amazon.com/linux/amazon-linux-2023
```

3. From now on, until mentioned, perform these steps on all 3 machines. Install Docker for all the 3 machines

```
[root@ip-172-31-90-172 ec2-user]# yum install docker -y
Last metadata expiration check: 0:21:16 ago on Fri Aug 30 04:01:12 2024.
Dependencies resolved.
```

Package	Architecture	Version
Installing:		
docker	x86_64	25.0.6-1.amzn2023.0.1
Installing dependencies:		
containerd	x86_64	1.7.20-1.amzn2023.0.1
iptables-libx	x86_64	1.8.8-3.amzn2023.0.2
iptables-nft	x86_64	1.8.8-3.amzn2023.0.2
libcgroup	x86_64	3.0-1.amzn2023.0.1
libnetfilter_conntrack	x86_64	1.0.8-2.amzn2023.0.2
libnfnetlink	x86_64	1.0.1-19.amzn2023.0.2
libnftnl	x86_64	1.2.2-2.amzn2023.0.2
pign	x86_64	2.5-1.amzn2023.0.3

Start the docker by running the command systemctl start docker in the terminal of all the ec2 instance.

```
Complete!
[root@ip-172-31-82-133 ec2-user]# systemctl start docker
[root@ip-172-31-82-133 ec2-user]#
```

4. Install the kubernetes on all 3 machines by searching for kubeadm and click on install kubernetes.

Select the red hat based distribution. This process will automatically disable SELinux before configuring kubelet so no need to run it separately in terminal.

version.

Debian-based distributions

Red Hat-based distributions

Without a package manager

1. Set SELinux to permissive mode:

These instructions are for Kubernetes 1.31.

```
# Set SELinux in permissive mode (effectively disabling it)
sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/
```

Copy the below script, to install kubernetes we need a kubernetes repo so this script helps us in getting that and paste it in the terminal.

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

```

Installed:
  containerd-1.7.20-1.amzn2023.0.1.x86_64      docker-25.0.6-1.amzn2023.0.1.x86_64      iptables-libc-1
  iptables-nft-1.8.8-3.amzn2023.0.2.x86_64    libcgroup-3.0-1.amzn2023.0.1.x86_64      libnetfilter_cc
  libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64    libnftnl-1.2.2-2.amzn2023.0.2.x86_64     pigz-2.5-1.amzn
  runc-1.1.11-1.amzn2023.0.1.x86_64

Complete!
[root@ip-172-31-90-172 ec2-user]# systemctl start docker
[root@ip-172-31-90-172 ec2-user]# sudo su
[root@ip-172-31-90-172 ec2-user]# yum repolist
repo id                                repo name
amazonlinux                             Amazon Linux 2023 repository

```

Run the command yum repolist to check whether the kubernetes repo has installed or not if successful installed then you can see a repo named as kubernetes

```

[root@ip-172-31-90-172 ec2-user]# yum repolist
repo id                                repo name
amazonlinux                            Amazon Linux 2023 repository
kernel-livepatch                        Amazon Linux 2023 Kernel Livepatch repository
kubernetes                             Kubernetes
[root@ip-172-31-90-172 ec2-user]#

```

Do the above steps for all the instances i.e for worker-1 and worker-2.

5. Perform this ONLY on the Master machine. Initialize the Kubecluster

```
sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=all
```

```
[addons] Applied essential addon: kube-proxy

Your Kubernetes control-plane has initialized successfully!

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

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

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

Then you can join any number of worker nodes by running the following on each as root:
kubeadm join 172.31.12.28:6443 --token 4bqw8.lua2ud0llr02uu55 \
  --discovery-token-ca-cert-hash sha256:b4edc7948be9bca50767f623b58e0612feedc144a7364f95be0dbd8c4614a169
```

Copy the join command and keep it in a notepad, we'll need it later.

Copy the mkdir and chown commands from the top and execute them

```
[ec2-user@ip-172-31-12-28 docker]$ mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

Then, add a common networking plugin called flammel file as mentioned in the code.

kubectl apply -f

<https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml>

```
[ec2-user@ip-172-31-12-28 docker]$ kubectl apply -f
```

<https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml>

Check the created pod using this command

Now, keep a watch on all nodes using the following command - watch kubectl get nodes

6. Perform this ONLY on the worker machines

Run the following command

```
sudo yum install iproute-tc -y
```

```
sudo systemctl enable kubelet  
sudo systemctl restart kubelet
```

Check the status of the pods using the following command

This command will show the status of all the pods.

```
kubectl get pods -n kube-system
```

Following command will show the status of the pod named daemonset.

```
kubectl get daemonset -n kube-system
```

```
[ec2-user@ip-172.31.12.28 docker]$ kubectl get pods -n kube-system  
NAME                               READY   STATUS    RESTARTS   AGE  
coredns-55cb5b8774-fx12f           1/1     Running   0          100s  
coredns-55cb5b8774-xn14v           1/1     Running   0          100s  
etcd-ip-172.31.12.28.ec2.internal 1/1     Running   0          75s  
kube-apiserver-ip-172.31.12.28.ec2.internal 1/1     Running   1          2m  
kube-controller-manager-ip-172.31.12.28.ec2.internal 0/1 CrashLoopBackOff 1          70s  
kube-proxy-4dv8m                   1/1     Running   2          100s  
kube-scheduler-ip-172.31.12.28.ec2.internal 1/1     Running   1          76s
```

```
[ec2-user@ip-172.31.12.28 docker]$ kubectl get daemonset -n kube-system  
NAME      DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE  
kube-proxy 1         1         1         1           1           kubernetes.io/os=linux 3m
```

That's it, we now have a Kubernetes cluster running across 3 AWS EC2 Instances. This cluster can be used to further deploy applications and their loads being distributed across these machines.

Conclusion:

Kubernetes cluster was successfully established using three AWS EC2 instances, which includes one Master and two Worker nodes. The process began with the creation of instances and configuration of settings to begin the communication. Docker was installed on all machines followed by the installation of Kubernetes components and the necessary repositories. The Master node was initialized with the 'kubeadm init' command, and a plugin called Flannel was deployed to enable pod communication. Error incurred during initialization can be solved by changing the instance type to t3.medium or t3.large and thus 3 nodes were connected successfully

Experiment 4

Aim:

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

Create a key pair

The screenshot shows the AWS EC2 Key Pairs page with one key pair listed:

Name	Type	Created	Fingerprint	ID
test	rsa	2024/08/28 09:58 GMT+5:30	35:80:ad:22:3d:50:d3:0f:77:ff:47:ea:5b:ec:e2:7...	key-07554cb45a9...

Below this is a 'Create key pair' dialog box:

Create key pair

Key pair
A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name: test

Key pair type: rsa

Private key file format: .pem

Tags - optional:

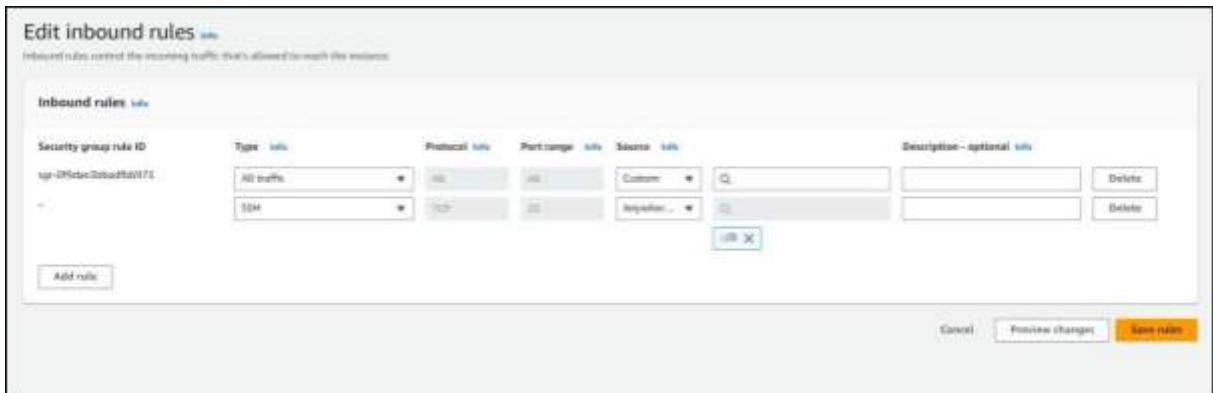
Create key pair

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

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

The screenshot shows the AWS EC2 Instances page with one instance listed:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IP
Instance1	i-0f12e86e5e7f7725	Running	t1.medium	2/2 checks green	View alarms	us-east-1a	m2-98-81-152-395.cs...	88.81.152...



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

```
Dell@DESKTOP-OVNTAIM MINGW64 ~/Downloads (master)
$ chmod 400 test1.pem
```

3. Now use this command on the terminal: ssh -i <keyname>.pem
ubuntu@<public_ip_address> and replace

```
Dell@DESKTOP-OVNTAIM MINGW64 ~/Downloads (master)
$ ssh -i "test1.pem" ec2-user@ec2-44-204-14-37.compute-1.amazonaws.com
The authenticity of host 'ec2-44-204-14-37.compute-1.amazonaws.com (44.204.14.37)' can't be established.
ED25519 key fingerprint is SHA256:CtxhAZnv4MFbUai03z96MQzMKK6JxuN/nWlIerDSazI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-204-14-37.compute-1.amazonaws.com' (ED25519)
to the list of known hosts.

      #
      ####_
      Amazon Linux 2023
      \###\
      '#'
      '#/  _-->
      V~'   /-
      .-.  /
      /m/,-/
[ec2-user@ip-172-31-81-24 ~]$
```

4. Docker installation:

We will be installing docker by using “`sudo yum install docker -y`”

```
[root@ip-172-31-81-24 ec2-user]# sudo yum install docker -y
Last metadata expiration check: 0:01:25 ago on Sat Sep 14 06:42:34 2024.
Dependencies resolved.

-- Package           Architecture      Version          Repository      Size
Installing:
  docker             x86_64          25.0.6-1.amzn2023.0.2
  iproute            x86_64          1.7.20-1.amzn2023.0.1
  iproute-tls         x86_64          1.6.8-3.amzn2023.0.2
  iproute-eft        x86_64          1.8.8-3.amzn2023.0.2
  libcapgroup        x86_64          3.0-1.amzn2023.0.1
  libcapwriter_comtrack x86_64          1.0.0-1.amzn2023.0.2
  libnfntnl          x86_64          1.1.19.amzn2023.0.2
  libnfntnl          x86_64          1.2.2-7.amzn2023.0.2
  pigrx              x86_64          2.5-1.amzn2023.0.3
  runc               x86_64          1.1.19-1.amzn2023.0.1

Transaction Summary

Install 10 Packages

Total download size: 84 M
Installed size: 317 M
Dependencies resolved.

Prepending repositories to rpmdb...
(1/10): iproute-tls-1.x86_64.amzn2023.0.2.x86_64.rpm
(2/10): iproute-eft-1.x86_64-3.amzn2023.0.2.x86_64.rpm
(3/10): libcapgroup-3.0-1.amzn2023.0.1.x86_64.rpm
(4/10): libcapwriter-comtrack-1.0-1.amzn2023.0.2.x86_64.rpm
(5/10): libnfntnl-1.1.19.amzn2023.0.2.x86_64.rpm
(6/10): libnfntnl-1.2.2-2.x86_64.amzn2023.0.1.x86_64.rpm
(7/10): pigrx-2.5-1.amzn2023.0.1.x86_64.rpm
(8/10): runc-1.1.19-1.amzn2023.0.1.x86_64.rpm
(9/10): containerd-1.20-1.amzn2023.0.1.x86_64.rpm
[10/10]: docker-25.0.6-1.amzn2023.0.2.x86_64.rpm

Total:
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing:
  Installing : runc-1.1.19-1.amzn2023.0.1.x86_64
  Installing : containerd-1.20-1.amzn2023.0.1.x86_64
Running transaction
  1/10
```

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

```
cd /etc/docker
```

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

{

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

}

EOF

```
sudo systemctl enable docker
```

`sudo systemctl daemon-reload`

```
sudo systemctl restart docker
```

```
[root@ip-172-31-81-24 ec2-user]# cd /etc/docker  
cat <<EOF | sudo tee /etc/docker/daemon.json  
{  
    "exec-opts": ["native.cgroupdriver=systemd"]  
}  
EOF  
sudo systemctl enable docker  
sudo systemctl daemon-reload  
sudo systemctl restart docker  
{  
    "exec-opts": ["native.cgroupdriver=systemd"]  
}  
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.  
[root@ip-172-31-81-24 ec2-user]#
```

6. Kubernetes installation:

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

Debian-based distributions Red Hat-based distributions

Without a package manager

1. Set SELinux to `permissive` mode:

These instructions are for Kubernetes 1.31.

```
# Set SELinux in permissive mode (effectively disabling it)
sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
```

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

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

3. Install kubelet, kubeadm and kubectl:

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

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

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

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

```
(/9/): kubeadm-1.31.1-150500.1.1.x86_64.rpm
(/9/): kubelet-1.31.1-150500.1.1.x86_64.rpm
(/9/): kubernetes-cni-1.5.1-150500.1.1.x86_64.rpm

total 0
kubernetes
  Importing GPG key 0xA9296436:
    Isidir : "sys-kubernetes OBS Project .csv/kubernetesBuild/opensuse.orgs"
    Fingerprint: https://pkgs.k8s.io/core/stable/v1.31/rpm/repo/epel.repo.gpg.key
Key Imported successfully
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing:
    Installing : kubernetes-cni-1.5.1-150500.1.1.x86_64
    Installing : cri-tools-1.31.1-150500.1.1.x86_64
    Installing : libnetfilter_qdisc-1.0.5-2.xzvnc023.0.2.x86_64
    Installing : libnetfilter_cttimeout-1.0.0-19.xzvnc2023.0.2.x86_64
    Installing : libnetfilter_cthelper-1.0.0-21.xzvnc023.0.2.x86_64
    Installing : conntrack-tools-1.4.6-2.xzvnc2023.0.2.x86_64
    Running scriptlets:
      conntrack-tools-1.4.6-2.xzvnc2023.0.2.x86_64
    Installing : kubelet-1.31.1-150500.1.1.x86_64
    Running scriptlets:
      kubelet-1.31.1-150500.1.1.x86_64
    Installing : kubernetes-cni-1.5.1-150500.1.1.x86_64
    Running scriptlets:
      kubelet-1.31.1-150500.1.1.x86_64
    Verifying:
      conntrack-tools-1.4.6-2.xzvnc2023.0.2.x86_64
      libnetfilter_qdisc-1.0.5-2.xzvnc023.0.2.x86_64
      libnetfilter_cttimeout-1.0.0-19.xzvnc2023.0.2.x86_64
      libnetfilter_cthelper-1.0.0-21.xzvnc023.0.2.x86_64
      cri-tools-1.31.1-150500.1.1.x86_64
    Verifying:
      kubelet-1.31.1-150500.1.1.x86_64
    Verifying:
      kubernetes-cni-1.5.1-150500.1.1.x86_64
    Verifying:
      kubelet-1.31.1-150500.1.1.x86_64
    Verifying:
      kubernetes-cni-1.5.1-150500.1.1.x86_64

Installed:
  cri-tools-1.31.1-150500.2.0.x86_64
  kubelet-1.31.1-150500.1.1.x86_64
  libnetfilter_qdisc-1.0.5-2.xzvnc023.0.2.x86_64

Complete!
[root@ip-177-31-81-24 docker]# sudo systemctl enable --now kubelet
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /usr/lib/systemd/system/kubelet.service.
[root@ip-177-31-81-24 docker]#
```

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

```
[root@ip-172-31-81-24 docker]# sudo swapoff -a
[root@ip-172-31-81-24 docker]# echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf
net.bridge.bridge-nf-call-iptables=1
[root@ip-172-31-81-24 docker]# sudo sysctl -p
net.bridge.bridge-nf-call-iptables = 1
[root@ip-172-31-81-24 docker]#
```

8. Initializing kubecluster:

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

```
[root@ip-172-31-81-24 docker]# sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
  [WARNING FileExisting-socat]: socat not found in system path
  [WARNING FileExisting-tc]: tc not found in system path
error execution phase preflight: [preflight] Some fatal errors occurred:
  [ERROR NumCPU]: the number of available CPUs is less than the required 2
  [ERROR Mem]: the system RAM (949 MB) is less than the minimum 1700 MB
[preflight] If you know what you are doing, you can make a check non-fatal with '--ignore-preflight-errors=...'
To see the stack trace of this error execute with --v=5 or higher
[root@ip-172-31-81-24 docker]# sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=NumCPU,Mem
[init] Using Kubernetes version: v1.31.0
```

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

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

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

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

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

```
You should now deploy a pod network to the cluster.
```

```
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/
```

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

```
kubeadm join 172.31.81.24:6443 --token 4a91z3.yz6rwmmkf9yncyd2 \
    --discovery-token-ca-cert-hash sha256:3404bd1bcd9cf90a003673f622d1672acb4c6ce7c15c4738c80a0a1560fe70d
[root@ip-172-31-81-24 docker]# |
```

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

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

10. Then, add a common networking plugin called flannel:

```
kubectl apply -f
```

```
https://raw.githubusercontent.com/coreos/flannel/master/Documentation/
kube-flannel.yml
```

```
[root@ip-172-31-81-24 docker]# kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-Flannel-Cfg created
daemonset.apps/kube-flannel-ds created
[root@ip-172-31-81-24 docker]# |
```

11. Apply this deployment file using this command to create a

```
deployment kubectl apply -f
```

```
https://k8s.io/examples/application/deployment.yaml
```

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

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

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

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

```
[root@ip-172-31-16-56 ~]# kubectl describe pod nginx
Name:           nginx-deployment-d556bf558-gw8v8
Namespace:      default
Priority:       0
Service Account: default
Node:           <none>
Labels:         app=nginx
                pod-template-hash=d556bf558
Annotations:    <none>
Status:         Pending
IP:
IPs:
Controlled By: ReplicaSet/nginx-deployment-d556bf558
Containers:
  nginx:
    Image:        nginx:1.14.2
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:
Conditions:
  Type        Status
  PodScheduled  False
Volumes:
  kube-api-access-f9k9s:
    Type:           Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:   kube-root-ca.crt
    ConfigMapOptional: <nil>
    DownwardAPI:    true
QoS Class:      BestEffort
Node-Selectors:  <none>
Tolerations:
  node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
  node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type      Reason     Age      From            Message
  Warning   FailedScheduling  114s   default-scheduler  0/1 nodes are available: 1 node(s) had untolerated taint {node-role.kubernetes.io/control-plane}: 1. preemption: 0/1 nodes are available: 1 Preemption is not helpful for scheduling.
  Warning   FailedScheduling  3m18s  default-scheduler  0/1 nodes are available: 1 node(s) had untolerated taint {node-role.kubernetes.io/control-plane}: 1. preemption: 0/1 nodes are available: 1 Preemption is not helpful for scheduling.
```

Use the below command to remove taints

```
[ec2-user@ip-172-31-26-174 ~]$ kubectl taint nodes --all node-role.kubernetes.io/control-plane-
node/ip-172-31-26-174.ec2.internal untainted
```

13. Check the pod status

```
[ec2-user@ip-172-31-26-174 ~]$ kubectl get pods
NAME    READY   STATUS    RESTARTS   AGE
nginx   1/1     Running   1 (6s ago)  90s
```

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

```
[ec2-user@ip-172-31-26-174 ~]$ kubectl port-forward nginx 8081:80
Forwarding from 127.0.0.1:8081 -> 80
Forwarding from [::1]:8081 -> 80
```

15. Verify your deployment

Open up a new terminal and ssh to your EC2 instance.

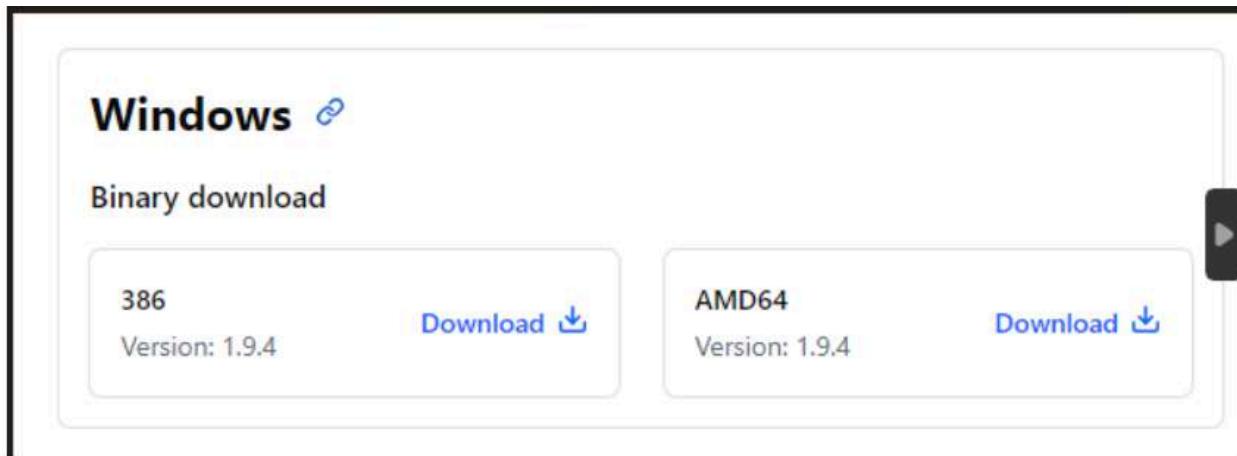
Then, use this curl command to check if the Nginx server is running.

```
curl --head http://127.0.0.1:8080
```

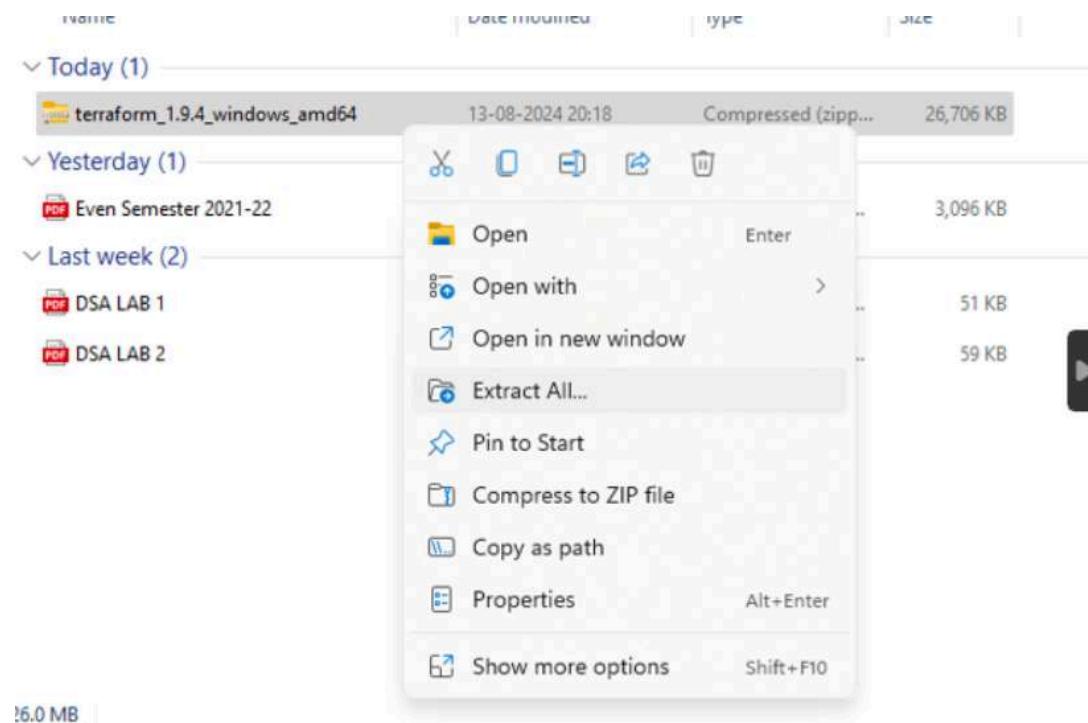
Conclusion: Firstly I created an EC2 AWS Linux instance successfully.then installed docker and kubernetes successfully. Initialized kubernetes and execute mkdir and chown command successfully. Then I tried to deploy nginx which initially gave an error. Then I deployed (simple-pod.yml) nginx successfully and also checked by using the get pods command.

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

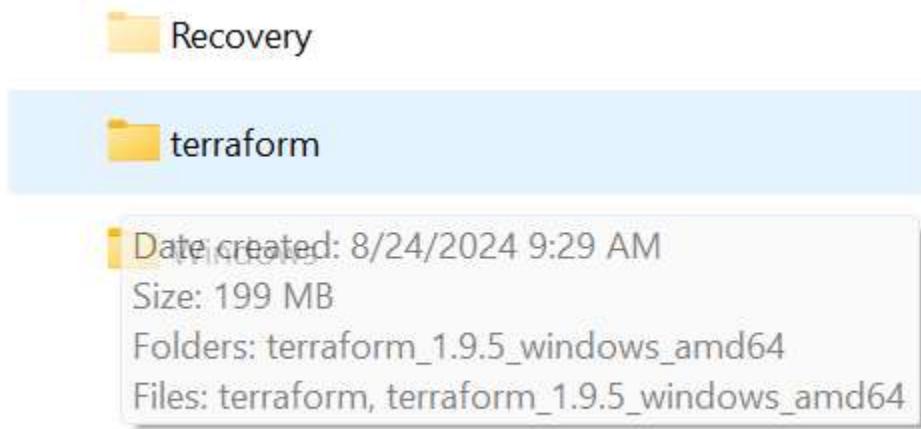
Step 1: Go to <www.terraform.io/downloads.html> and download the amd file compatible with your device



Step 2: It will download a zip file



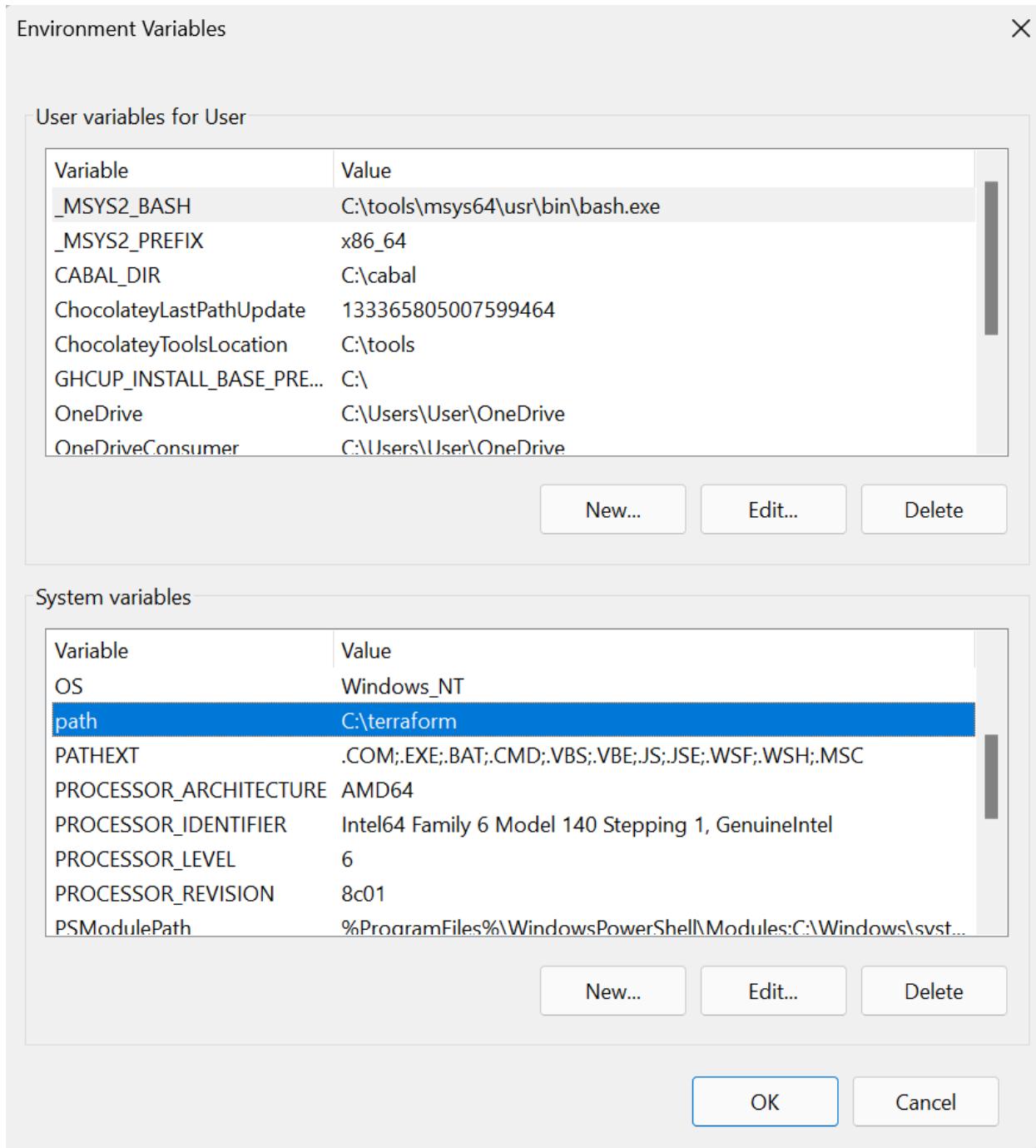
Step 3: Create a folder and name it <terraform> on C drive



Step 4: Move the extracted files in this folder

Name	Date modified	Type	Size
terraform_1.9.5_windows_amd64	8/24/2024 9:32 AM	File folder	
terraform	8/24/2024 9:36 AM	Application	88,962 KB
terraform_1.9.5_windows_amd64	8/24/2024 9:28 AM	Compressed (zipped)...	26,721 KB

Step 5: Go to Settings > System Properties > Environment variables and then add a new system variable



Step 6: To confirm proper installation, Open Windows Powershell as administrator and run terraform

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

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

PS C:\Windows\system32> terraform
Usage: terraform [global options] <subcommand> [args]

The available commands for execution are listed below.
The primary workflow commands are given first, followed by
less common or more advanced commands.

Main commands:
  init      Prepare your working directory for other commands
  validate   Check whether the configuration is valid
  plan       Show changes required by the current configuration
  apply      Create or update infrastructure
  destroy    Destroy previously-created infrastructure

All other commands:
  console    Try Terraform expressions at an interactive command prompt
  fmt        Reformat your configuration in the standard style
  force-unlock Release a stuck lock on the current workspace
  get        Install or upgrade remote Terraform modules
  graph      Generate a Graphviz graph of the steps in an operation
  import     Associate existing infrastructure with a Terraform resource
  login      Obtain and save credentials for a remote host
  logout     Remove locally-stored credentials for a remote host
  metadata   Metadata related commands
  output     Show output values from your root module
```

Experiment No. 6

Aim: To create a docker image using terraform

1. Check if the docker is functioning properly on your device by running some commands

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

Usage: docker [OPTIONS] COMMAND
      A self-sufficient runtime for containers

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

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

2. To write the code for our pipeline create a new folder



8/22/2024 4:10 PM

File folder

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

Name	Date modified	Type	Size
Docker	8/22/2024 4:10 PM	File folder	

Name	Date modified	Type	Size
docker.tf	8/22/2024 4:12 PM	TF File	1 KB

4. Write the code in that docker.tf file and save it



docker.tf - Notepad

```
File Edit Format View Help
terraform {
    required_providers {
        docker = {
            source = "kreuzwerker/docker"
            version = "2.21.0"
        }
    }
    provider "docker" {
        host = "npipe:///./pipe/docker_engine"
    }

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

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

5. Open the folder in command prompt to run terraform commands. The first step is to initialize terraform using the command > terraform init

```
D:\Terraform_Scripts\Docker>terraform init
Initializing the backend...
Initializing provider plugins...
- Finding kreuzwerker/docker versions matching "2.21.0"...
- Installing kreuzwerker/docker v2.21.0...
- Installed kreuzwerker/docker v2.21.0 (self-signed, key ID BD080C4571C6104C)
  Partner and community providers are signed by their developers.
  If you'd like to know more about provider signing, you can read about it here:
    https://www.terraform.io/docs/cli/plugins/signing.html
  Terraform has created a lock file .terraform.lock.hcl to record the provider
  selections it made above. Include this file in your version control repository
  so that Terraform can guarantee to make the same selections by default when
  you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.

D:\Terraform_Scripts\Docker>
```

5. The next command to run > terraform plan to check the available resources

```
D:\Terraform_Scripts\Docker>terraform plan
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the
following symbols:
+ create

Terraform will perform the following actions:

# docker_container.foo will be created
+ resource "docker_container" "foo" {
  + attach          = false
  + bridge          = (known after apply)
  + command         = (known after apply)
  + container_logs = (known after apply)
  + entrypoint      = (known after apply)
  + env              = (known after apply)
  + exit_code        = (known after apply)
  + gateway          = (known after apply)
  + hostname         = (known after apply)
  + id               = (known after apply)
  + image             = (known after apply)
  + init              = (known after apply)
  + ip_address       = (known after apply)
  + ip_prefix_length = (known after apply)
  + ipc_mode          = (known after apply)
  + log_driver        = (known after apply)
  + logs              = false
  + must_run          = true
  + name              = "foo"
  + network_data     = (known after apply)
```

```

+ my           = <resource>
+ 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.

```

6. Now to finally execute the code, run the command terraform apply

```

$ ./Terraform_Scripts/Docker/terraform apply

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

Terraform will perform the following actions:

# docker_container.foo will be created
resource "docker_container" "foo" {
+ attach          = false
+ bridge          = (known after apply)
+ command         = (known after apply)
+ container_logs = (known after apply)
+ entrypoint      = (known after apply)
+ env             = (known after apply)
+ exit_code       = (known after apply)
+ gateway         = (known after apply)
+ hostname        = (known after apply)
+ id              = (known after apply)
+ image           = (known after apply)
+ init            = (known after apply)
+ ip_address      = (known after apply)
+ ip_prefix_length = (known after apply)
+ ipc_mode        = (known after apply)
+ log_driver      = (known after apply)
+ logs            = false
+ max_retry       = 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)
+ 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.

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

Enter a value: yes

docker image.ubuntu: Creating...
docker image.ubuntu: Still creating... [19s elapsed] docker image.ubuntu: Still creating... [20s elapsed] docker image.ubuntu: Still creating... [30s elapsed]
docker image.ubuntu: Creation complete after 30s [id=sha256:263966596cd12ad38aa99140716692777ba9ff8779a62ad93a704fe82e3e14
ubuntu:latest] docker_container.foo: Creating...

```

7. Here you can see the ubuntu image created

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
mscr.microsoft.com/dotnet/framework/aspnet	4.8-windowsservercore-ltsv2022	0b1ef1176a57	6 weeks ago	5.43GB
mscr.microsoft.com/dotnet/framework/sdk	4.8-windowsservercore-ltsv2022	c3f8c2735565	6 weeks ago	9.84GB
mscr.microsoft.com/dotnet/framework/runtime	4.8-windowsservercore-ltsv2022	e69ea8a5ec1b	6 weeks ago	5.16GB
mscr.microsoft.com/windows/servercore	ltsv2022	e60f47e635b7	7 weeks ago	4.84GB
mscr.microsoft.com/windows/nanoserver	ltsv2022	f0ca29645006	7 weeks ago	29.2GB
ubuntu	Latest	3de39ba859dc	2 minutes ago	77.8MB

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

```
docker_image.ubuntu: Refreshing state... [id=sha256:2dc29b50ddc2d301014750927770a887762d93ed0221fubuntu:latest]
Terraform used the selected provider to generate the following execution plan. Resource actions are indicated with the following symbols:
destroy: Terraform will perform the following actions:

# docker_image.ubuntu will be destroyed
resource "docker_image" "ubuntu" {
  - id          = "sha256:2dc29b50ddc2d301014750927770a887762d93ed0221fubuntu:latest" => null
  - image_id    = "sha256:2dc29b50ddc2d301014750927770a887762d93ed0221fubuntu:latest" => null
  - latest      = "sha256:2dc29b50ddc2d301014750927770a887762d93ed0221fubuntu:latest" => null
  - name        = "ubuntu:latest" => null
  - repo_digest = "ubuntu@sha256:2dc29b50ddc2d301014750927770a887762d93ed0221fubuntu:latest" => null
}

Plan: 0 to add, 0 to change, 1 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_image.ubuntu: Destroying... [id=sha256:2dc29b50ddc2d301014750927770a887762d93ed0221fubuntu:latest]
docker_image.ubuntu: Destruction complete after 0s

Destroy complete! Resources: 1 destroyed.
```

9. The docker image is destroyed

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
mscr.microsoft.com/dotnet/framework/aspnet	4.8-windowsservercore-ltsv2022	0b1ef1176a57	6 weeks ago	5.43GB
mscr.microsoft.com/dotnet/framework/sdk	4.8-windowsservercore-ltsv2022	c3f8c2735565	6 weeks ago	9.84GB
mscr.microsoft.com/dotnet/runtime	4.8-windowsservercore-ltsv2022	e69ea8a5ec1b	6 weeks ago	5.16GB
mscr.microsoft.com/windows/servercore	ltsv2022	e60f47e635b7	7 weeks ago	4.84GB
mscr.microsoft.com/windows/nanoserver	ltsv2022	f0ca29645006	7 weeks ago	29.2GB

Advance DevOps

Experiment 7

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

Steps:

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

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

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

```
C:\Users\DELL>docker login
Authenticating with existing credentials...
Stored credentials invalid or expired
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com/ to create one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security and is required for organizations using SSO. Learn more at https://docs.docker.com/go/access-tokens/
Username (dimple866): dimple866
Password:
Login Succeeded
```

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

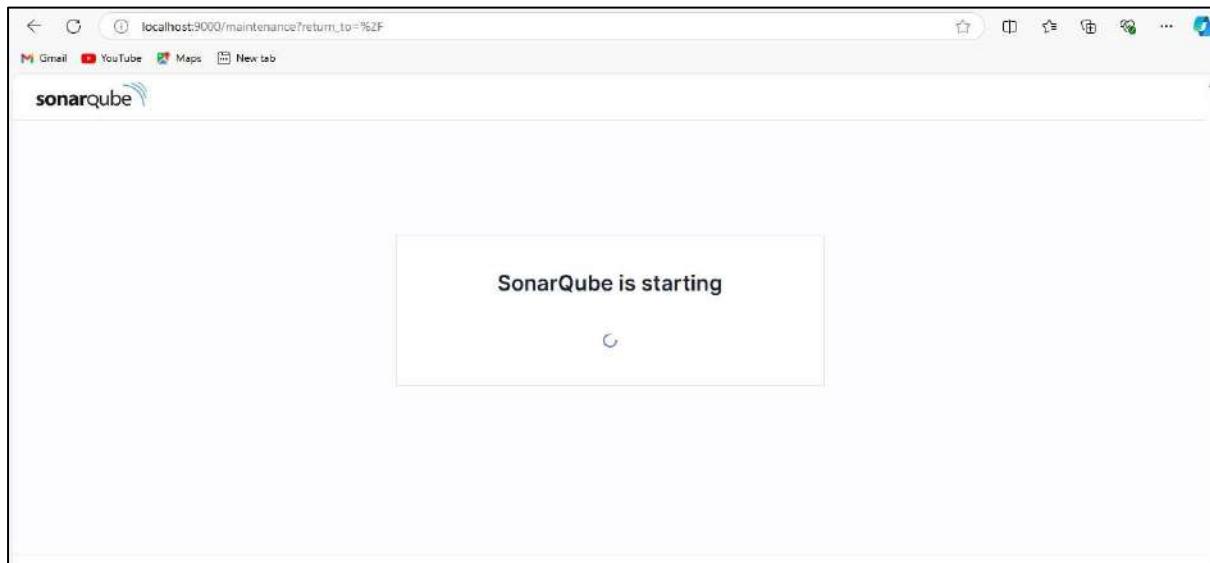
```
C:\Users\DELL>docker pull sonarqube
Using default tag: latest
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31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

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

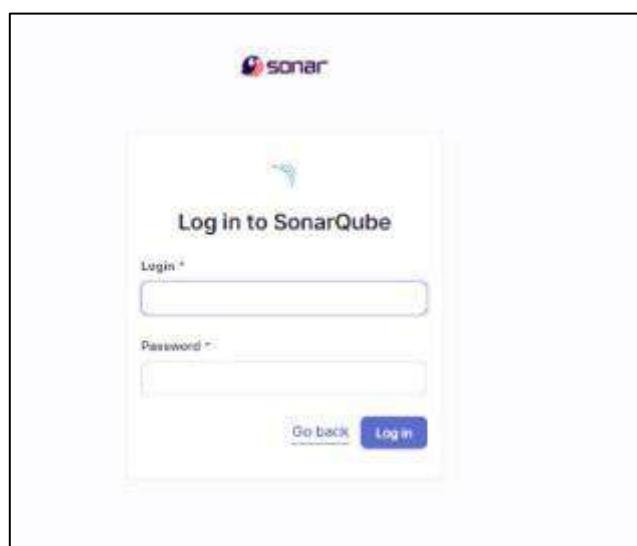
4. Run docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
Command to run the sonarqube.

```
C:\Users\DELL>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
cacf985dedebc00a642a4c69a502d611389e8f9fa46610febe75aa5021767cab
```

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



- Once SonarQube is started it will redirect you to login page. The login and password both for SonarQube is 'admin'



- Change the password for your SonarQube account.

Update your password

⚠️ This account should not use the default password.

Enter a new password

All fields marked with * are required

Old Password *

New Password *

Confirm Password *

Update

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

The screenshot shows the SonarQube interface for creating a local project. At the top, there's a navigation bar with links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, More, and a search bar. Below the navigation, a section titled "How do you want to create your project?" lists several import options:

- Import from Azure DevOps (Setup button)
- Import from Bitbucket Cloud (Setup button)
- Import from Bitbucket Server (Setup button)
- Import from GitHub (Setup button)
- Import from GitLab (Setup button)

Below these options, a message asks if the user is just testing or has an advanced use-case, and prompts them to "Create a local project".

A dark overlay at the bottom right contains promotional text for SonarLint, a free IDE plugin. It says: "Get the most out of SonarQube! Take advantage of the whole ecosystem by using SonarLint, a free IDE plugin that helps you find and fix issues earlier in your workflow. Connect SonarLint to SonarQube to sync rule sets and issue states." It includes "Learn More" and "Dismiss" buttons.

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

1 of 2

Create a local project

Project display name *

Project key *

Main branch name *

The name of your project's default branch [Learn More](#) 

CancelNext

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

2 of 2

Set up project for Clean as You Code

The new code definition sets which part of your code will be considered new code. This helps you focus attention on the most recent changes to your project, enabling you to follow the Clean as You Code methodology. Learn more: [Defining New Code](#) 

Choose the baseline for new code for this project

Use the global setting

Previous version

Any code that has changed since the previous version is considered new code.
Recommended for projects following regular versions or releases.

Define a specific setting for this project

Previous version

Any code that has changed since the previous version is considered new code.
Recommended for projects following regular versions or releases.

Number of days

Any code that has changed in the last x days is considered new code. If no action is taken on a new issue after x days, this issue will become part of the overall code.
Recommended for projects following continuous delivery.

Reference branch

Choose a branch as the baseline for the new code.
Recommended for projects using feature branches.

BackCreate project

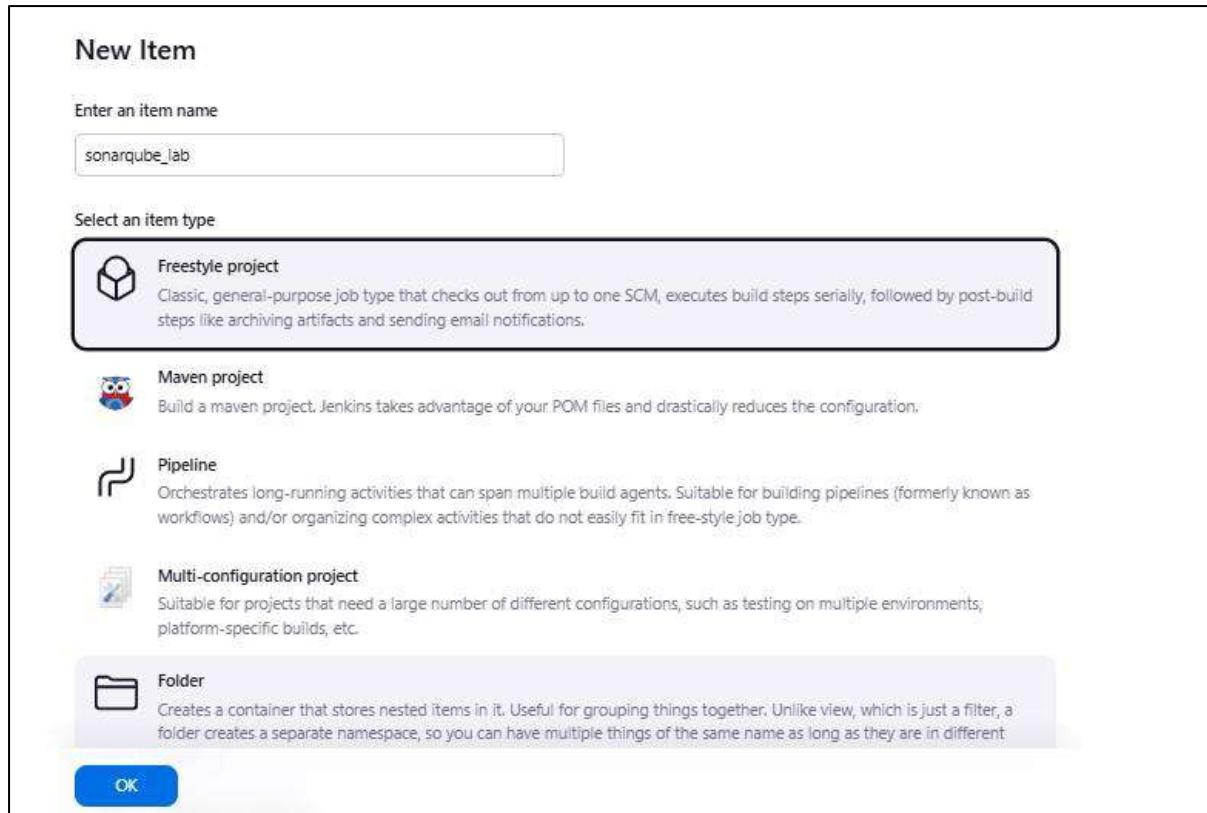
11. Go to Jenkins dashboard->Manage Jenkins->System and scroll down to SonarQube installations. Enter the name and URL in the fields and save the changes.

The screenshot shows the Jenkins configuration interface for SonarQube installations. It includes sections for SonarQube servers and SonarQube installations. In the SonarQube installations section, there is a table with one row. The row contains fields for Name (set to 'sonarqube'), Server URL (set to 'http://localhost:9000'), and Server authentication token (set to '- none -'). There is also an 'Advanced' button and a 'Save' button at the bottom.

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

The screenshot shows the Jenkins configuration interface for SonarQube Scanner. It includes sections for SonarQube Scanner and Ant installations. In the SonarQube Scanner section, there is a table with one row. The row contains fields for Name (set to 'sonarqube_sb'), Install automatically (checked), and Version (set to 'SonarQube Scanner 6.2.0.4584'). There is also an 'Add Installer' button and an 'Add SonarQube Scanner' button at the bottom. At the very bottom, there is a 'Save' button.

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



14. Use this GitHub repository in Source Code Management.

https://github.com/shazforiot/MSBuild_firstproject



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

The screenshot shows the 'Execute SonarQube Scanner' dialog box. It includes fields for 'JDK' (selected 'Inherit From Job'), 'Path to project properties' (empty), 'Analysis properties' containing configuration like sonar.projectKey=sonarqube, sonar.login=admin, sonar.password=123456, sonar.hostUrl=http://localhost:9000, and sonar.sources=., and sections for 'Additional arguments' and 'JVM Options' (both empty).

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

The screenshot shows the 'Global Permissions' section of the SonarQube administration interface. It lists groups and users with their assigned permissions. The 'Administrator' group has checked boxes for 'Administer System', 'Execute Analysis', and 'Create'. Other groups like 'sonar-administrators' and 'sonar-users' have different permission sets. The 'Administrator' user 'admin' also has these permissions checked. A note at the bottom indicates that 'Anyone' (DEPRECATED) also has these permissions.

Group/User	Administer System	Execute Analysis	Create
sonar-administrators	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
sonar-users	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Anyone DEPRECATED	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Administrator admin	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

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

Dashboard > sonarqube-lab > #10

Status: #10 (Sep 25, 2024, 11:21:17 AM)

Started by user Dimple Dalwani

This run spent:

- 13 ms waiting;
- 1 min 2 sec build duration;
- 1 min 2 sec total from scheduled to completion;

Git Build Data

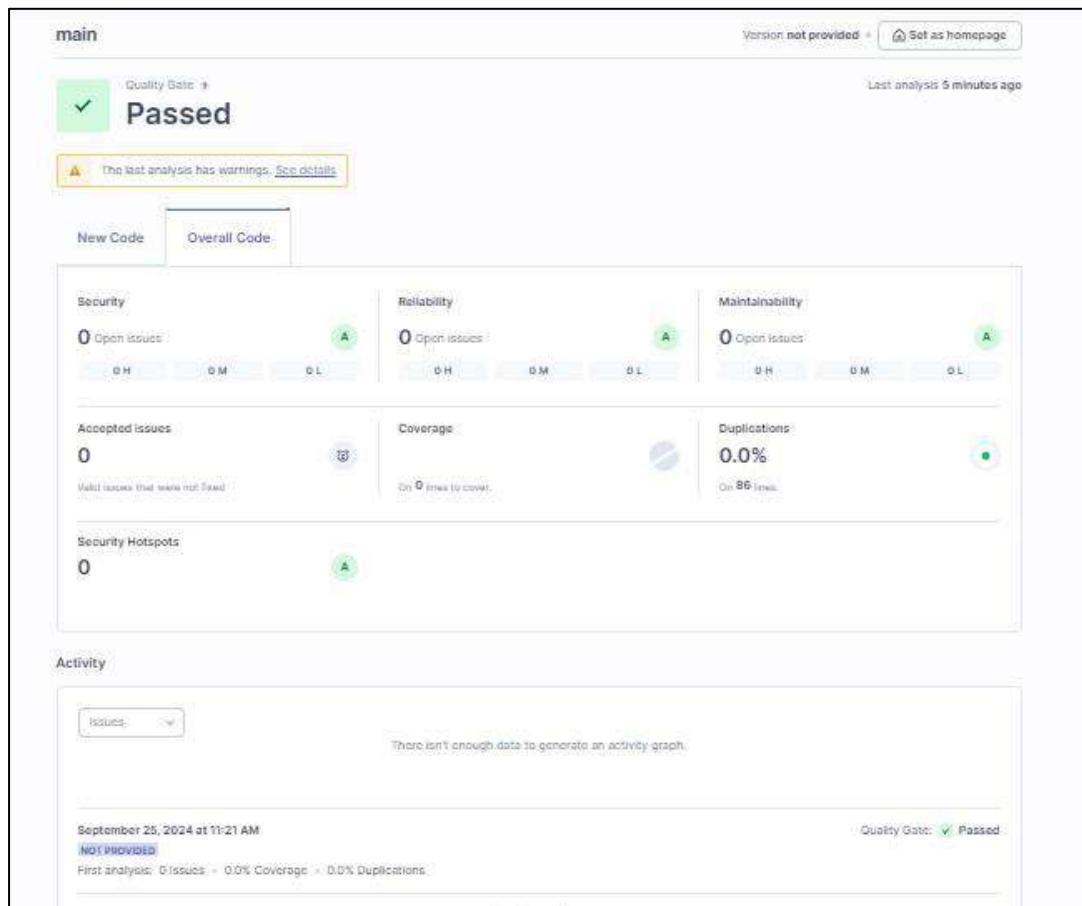
Revision: f2bc042c6e72427c380bcace6a6fe87b40ad1
Repository: <https://github.com/shozfciid/MSBuildFirstProject>
refs/remotes/origin/master

No changes.

18. Check the console Output

```
for block at line 17. Keep only the first 100 references.
23:13:58.632 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html
for block at line 296. Keep only the first 100 references.
23:13:58.632 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html
for block at line 75. Keep only the first 100 references.
23:13:58.632 INFO CPD Executor CPD calculation finished (done) | time=94361ms
23:13:58.695 INFO SCM revision ID 'ba799ba7e1b576f04a4612322b0412c5e6e1e5e4'
23:15:46.177 INFO Analysis report generated in 14542ms, dir size=127.2 MB
23:15:55.734 INFO Analysis report compressed in 9547ms, zip size=29.6 MB
23:15:59.127 INFO Analysis report uploaded in 3301ms
23:15:59.132 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://127.0.0.1:9000/dashboard?id=sonarqube-test
23:15:59.132 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
23:15:59.132 INFO More about the report processing at http://127.0.0.1:9000/api/ce/task?id=fbad731f-dcba-45c3-bfdd-b2ed2fec3a0e
23:16:05.629 INFO Analysis total time: 10:30.120 s
23:16:05.636 INFO SonarScanner Engine completed successfully
23:16:06.248 INFO EXECUTION SUCCESS
23:16:06.273 INFO Total time: 10:47.728s
[Pipeline]
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

19. Go back to SonarQube and check the project.



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

Advance DevOps

Experiment 8

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

Steps:

1. Open the Jenkins dashboard.

The screenshot shows the Jenkins dashboard with the following interface elements:

- Left Sidebar:** Includes links for "New Item", "Build History", "Project Relationship", "Check File Fingerprint", "Manage Jenkins", and "My Views". It also displays sections for "Build Queue" (No builds in the queue) and "Build Executor Status" (1 idle, 2 idle, 0 stuck).
- Top Bar:** Features a search bar ("Search (CTRL+K)"), notification icons, and user information ("Nayaab Jindani" and "log out").
- Main Content Area:** A table listing Jenkins projects. The columns are: Status (S), Last Build (W), Name, Last Success, Last Failure, and Last Duration.

S	W	Name	Last Success	Last Failure	Last Duration
Green	Cloudy	Advdevops_lab	8 hr 30 min #10	8 hr 33 min #9	10 min
Green	Sunny	Devops Pipeline	1 mo 24 days #1	N/A	6.1 sec
Green	Sunny	DevOps_Practical	1 mo 9 days #1	N/A	0.71 sec
Red	Cloudy	Maven_job	N/A	1 mo 8 days #2	5.6 sec
Red	Cloudy	Maven_project	29 days #1	29 days #2	47 sec
Blue	Sunny	new	N/A	N/A	N/A
Green	Sunny	Pipeline_devops	1 mo 8 days #1	N/A	5.5 sec
Green	Cloudy	Selenium_test	11 days #4	11 days #1	44 sec

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

```
docker pull sonarqube:latest
```

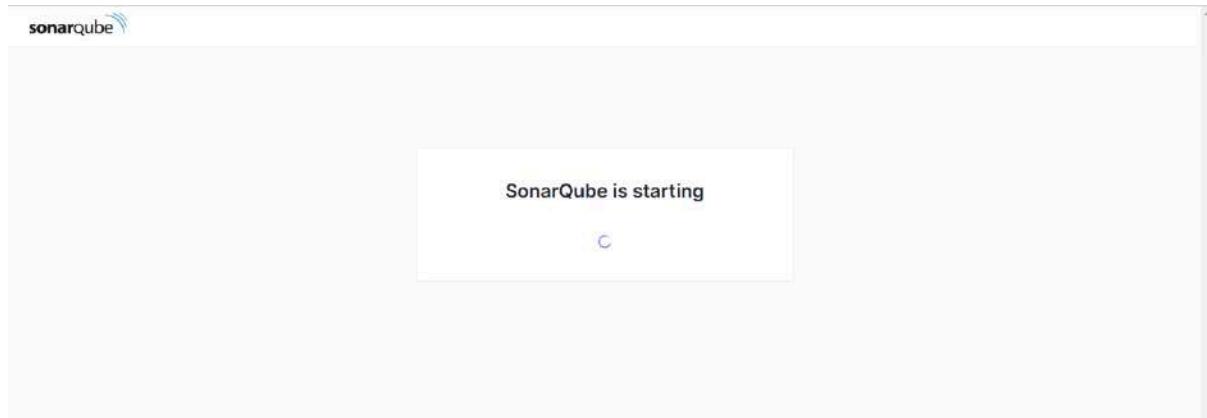
```
docker pull sonarqube:latest
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: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

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

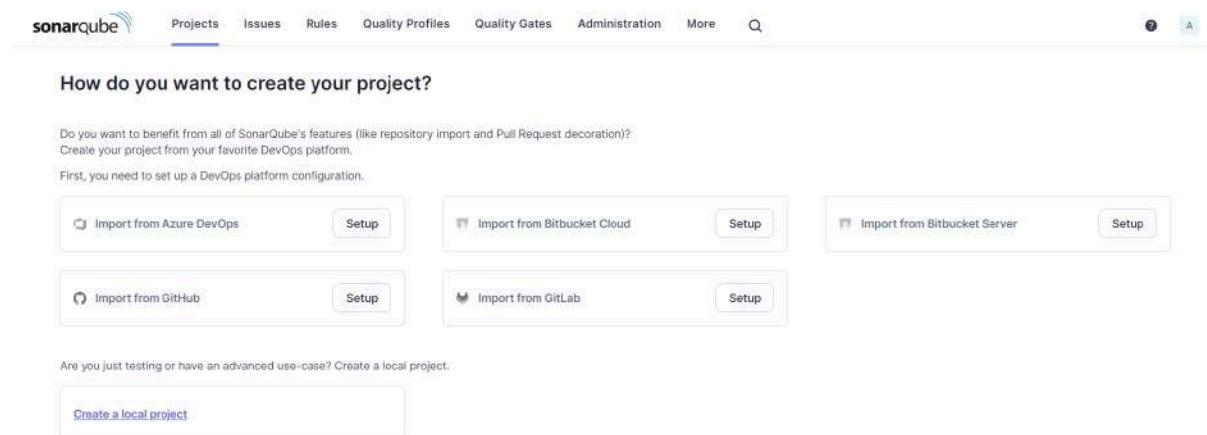
3. Next, we will run SonarQube in a docker container

```
PS C:\Users\DELL> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest  
67aeea599cf48e12d50da592eff01d8257f58e6c1bffd50446066e5f2a8844
```

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



- Now login to SonarQube using username and password.



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

1 of 2

Create a local project

Project display name *

 ✓

Project key *

 ✓

Main branch name *

The name of your project's default branch [Learn More](#)

CancelNext

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

Enter an item name

 » 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.

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

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

```
}
```

```
stage('SonarQube analysis') {
```

```
    withSonarQubeEnv('sonarqube') {
```

```
        sh "<PATH_TO SONARQUBE FOLDER>/bin//sonar-scanner \
```

```
            -D sonar.login=<SonarQube_USERNAME> \
```

```
            -D sonar.password=<SonarQube_PASSWORD> \
```

```
            -D sonar.projectKey=<Project_KEY> \
```

```
            -D sonar.exclusions=vendor/**,resources/**,**/*.java \
```

```
            -D sonar.host.url=http://127.0.0.1:9000/"
```

```
    }
```

```
}
```

```
}
```

(Change the path and credentials)

Definition

Pipeline script

Script ?

```
1 * node {  
2     stage('Cloning the GitHub Repo') {  
3         git 'https://github.com/shazforiot/GOL.git'  
4     }  
5     stage('SonarQube analysis') {  
6         withSonarQubeEnv('sonarqube') {  
7             sh "";  
8             C:/Users/Dell/Downloads/sonar-scanner-cli-6.2.0.4584-windows-x64/bin/sonar-scanner \  
9                 -D sonar.login=admin \  
10                -D sonar.password=nayab \  
11                -D sonar.projectKey=sonarqube-test \  
12                -D sonar.exclusions=vendor/**,resources/**,**/*.java \  
13                -D sonar.host.url=http://127.0.0.1:9000/  
14                ""  
15         }  
16     }  
17 }
```

Use Groovy Sandbox ?

Pipeline Syntax

Save Apply

9. Now run the build.

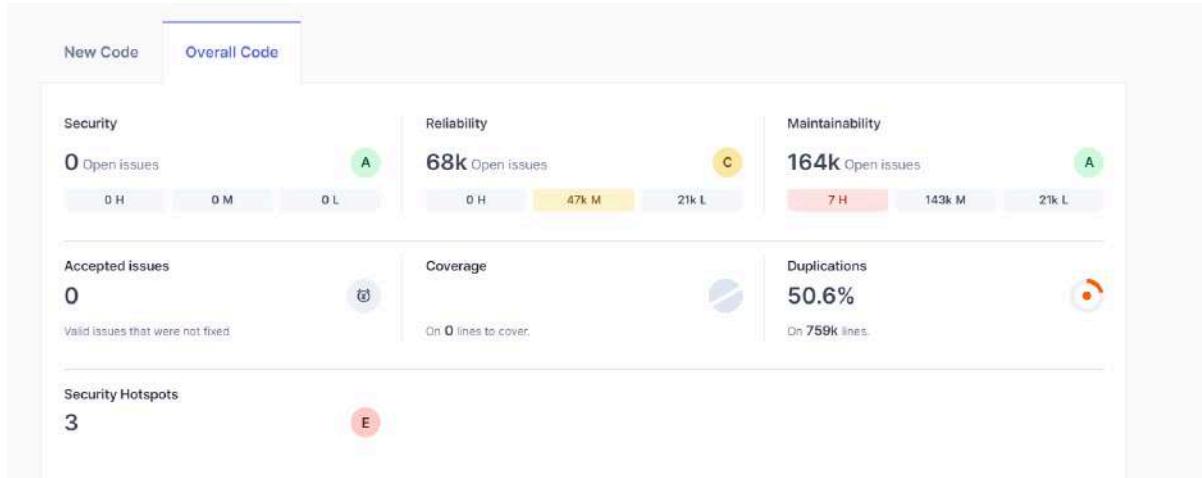
Console Output

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

The build is successful.

```
for block at line 17. Keep only the first 100 references.
23:13:58.632 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html
for block at line 296. Keep only the first 100 references.
23:13:58.632 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html
for block at line 75. Keep only the first 100 references.
23:13:58.632 INFO CPD Executor CPD calculation finished (done) | time=9436ms
23:13:58.695 INFO SCM revision ID 'ba799ba7e1b576f04a4612322b0412c5e6e1e5e4'
23:15:46.177 INFO Analysis report generated in 14542ms, dir size=127.2 MB
23:15:55.734 INFO Analysis report compressed in 9547ms, zip size=29.6 MB
23:15:59.127 INFO Analysis report uploaded in 3391ms
23:15:59.132 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://127.0.0.1:9000/dashboard?id=sonarqube-test
23:15:59.132 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
23:15:59.132 INFO More about the report processing at http://127.0.0.1:9000/api/ce/task?id=fbad731f-dcba-45c3-bfdd-b2ed2fec3a9e
23:16:05.629 INFO Analysis total time: 10:30.120 s
23:16:05.636 INFO SonarScanner Engine completed successfully
23:16:06.248 INFO EXECUTION SUCCESS
23:16:06.273 INFO Total time: 10:47.728s
[Pipeline]
[Pipeline] // withSonarQubeEnv
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

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



gametoflife-acceptance-tests/Dockerfile	
<input type="checkbox"/> Use a specific version tag for the image.	Intentionality Maintainability (C)
<input type="radio"/> Open ✓ <input type="radio"/> Not assigned ✓	L1 - 5min effort - 4 years ago - ⚡ Code Smell - ⚡ Major
<input type="checkbox"/> Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.	Intentionality Maintainability (C)
<input type="radio"/> Open ✓ <input type="radio"/> Not assigned ✓	L12 - 5min effort - 4 years ago - ⚡ Code Smell - ⚡ Major
<input type="checkbox"/> Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.	Intentionality Maintainability (C)
<input type="radio"/> Open ✓ <input type="radio"/> Not assigned ✓	L12 - 5min effort - 4 years ago - ⚡ Code Smell - ⚡ Major
<input type="checkbox"/> Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.	Intentionality Maintainability (C)
	No tags +

gameoflife-core/build/reports/tests/all-tests.html

Add "lang" and/or "xml:lang" attributes to this "<html>" element Intentionality
Reliability ● accessibility wcag2-a +

 Open ▾ Not assigned ▾ L1 - 2min effort - 4 years ago - ⚡ Bug - ⚡ Major

Add "<th>" headers to this "<table>". Intentionality
Reliability ● accessibility wcag2-a +

 Open ▾ Not assigned ▾ L1 - 2min effort - 4 years ago - ⚡ Bug - ⚡ Major

gameoflife-core/build/reports/tests/allclasses-frame.html

Add "lang" and/or "xml:lang" attributes to this "<html>" element Intentionality
Reliability ● accessibility wcag2-a +

 Open ▾ Not assigned ▾ L1 - 2min effort - 4 years ago - ⚡ Bug - ⚡ Major

Add "<th>" headers to this "<table>". Intentionality

gameoflife-acceptance-tests/Dockerfile

Use a specific version tag for the image. Intentionality
Maintainability ● No tags +

 Open ▾ Not assigned ▾ L1 - 5min effort - 4 years ago - ⚡ Code Smell - ⚡ Major

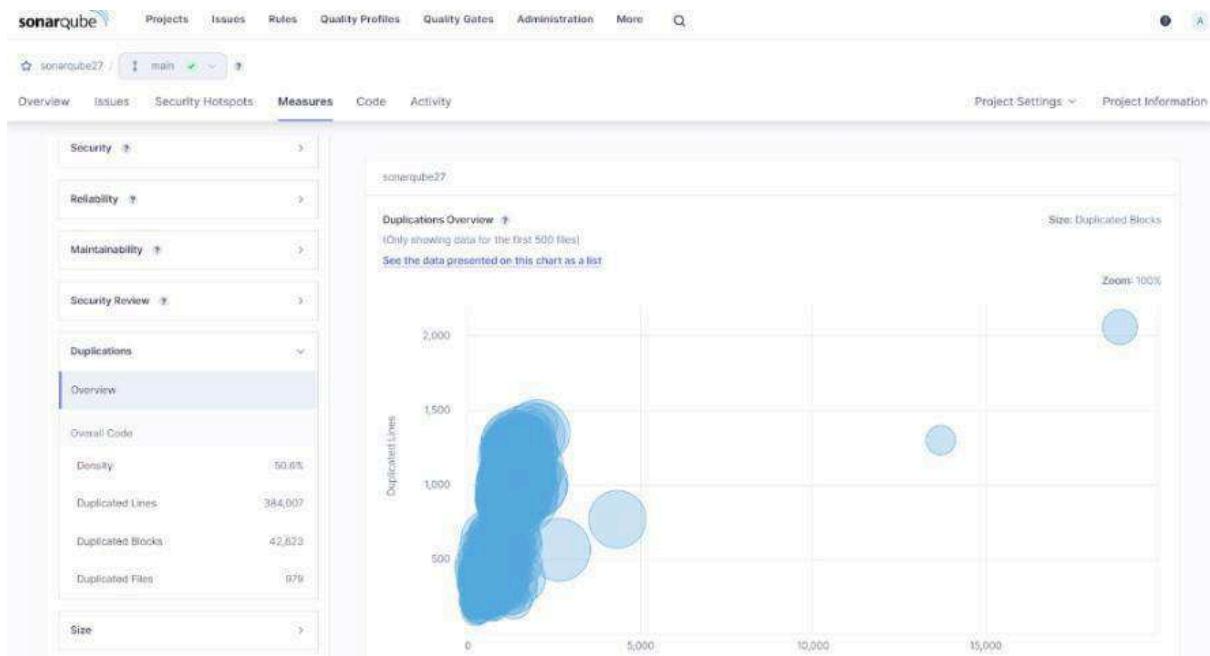
Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. Intentionality
Maintainability ● No tags +

 Open ▾ Not assigned ▾ L12 - 5min effort - 4 years ago - ⚡ Code Smell - ⚡ Major

Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. Intentionality
Maintainability ● No tags +

 Open ▾ Not assigned ▾ L12 - 5min effort - 4 years ago - ⚡ Code Smell - ⚡ Major

Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. Intentionality
Maintainability ● No tags +



Conclusion:

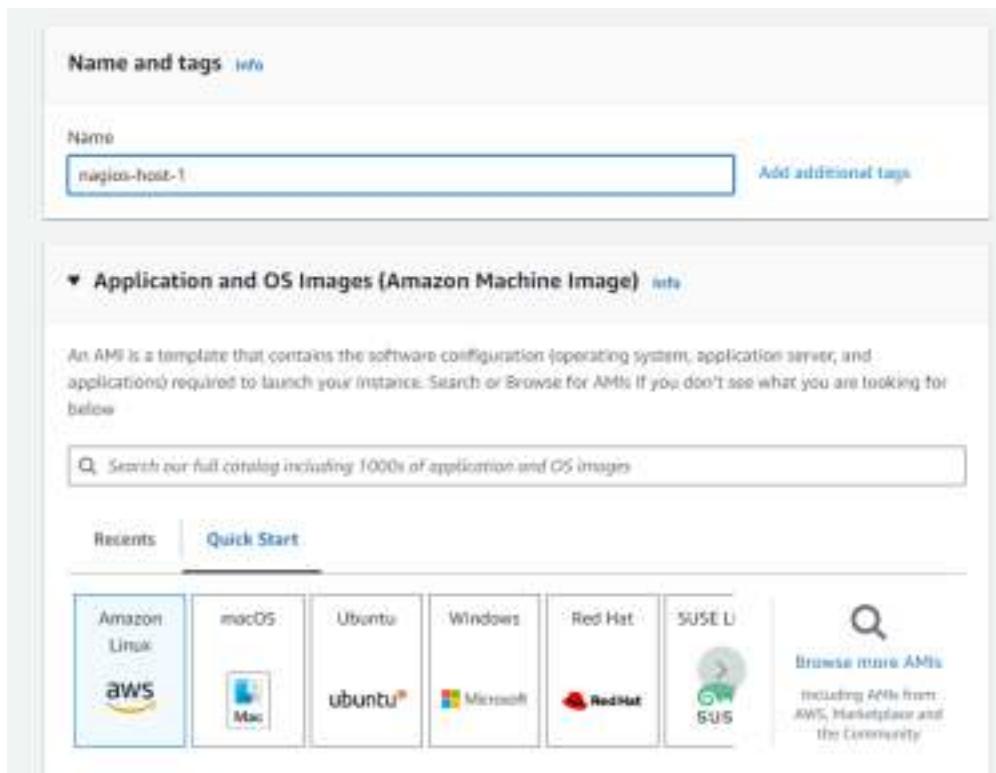
In this experiment we created a Jenkins CICD Pipeline to perform a static analysis of the code to detect bugs, code smells, and security vulnerabilities on a sample code. It is to be checked whether the sonar scanner plugin is installed in Jenkins or not and also provide the correct path and credentials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

Experiment 9

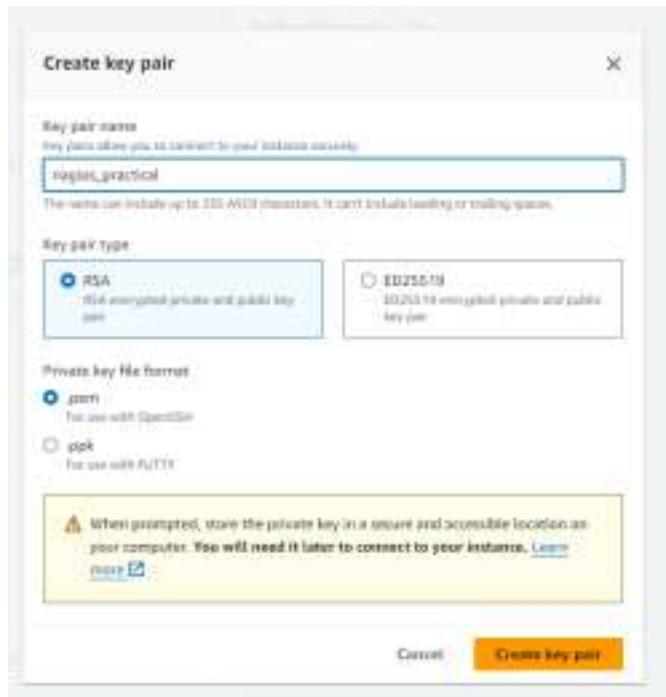
Aim: To understand continuous monitoring and installation and configuration of Nagios Core, Nagios Plugins and NRPE (Nagios Remote Plugin Executor) on Linux Machine.

Steps:

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



2. Now we will create a key pair.



3. Use the created key pair while creating the instance

The screenshot shows the 'Key pair (login)' configuration page. It displays the selected key pair name 'nagios_practical'. A 'Create new key pair' button is visible at the bottom right.

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

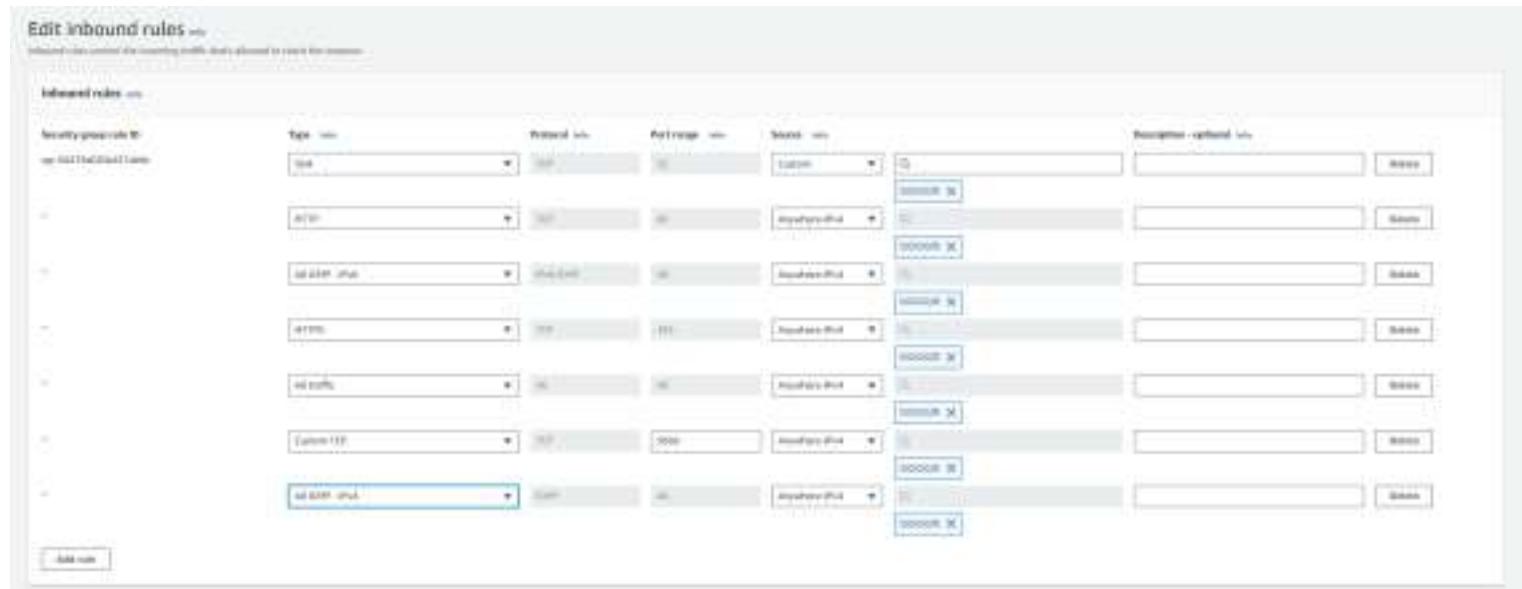
(The security group name is visible during instance creation and also on the ec2 instances dashboard)

Security Groups (1) View resources Edit						
Create security group on VPC Create community group						
#	Name	Security group ID	Security group rules	IP range	Description	Last modified
1	lambda-2014-09-06	sg-0000000000000000	lambda-2014-09-06	0.0.0.0/0	lambda-2014-09-06	2014-09-06 10:10:10
2	lambda-2014-09-06-1	sg-0000000000000001	lambda-2014-09-06-1	0.0.0.0/0	lambda-2014-09-06-1	2014-09-06 10:10:10
3	lambda-2014-09-06-2	sg-0000000000000002	lambda-2014-09-06-2	0.0.0.0/0	lambda-2014-09-06-2	2014-09-06 10:10:10
4	lambda-2014-09-06-3	sg-0000000000000003	lambda-2014-09-06-3	0.0.0.0/0	lambda-2014-09-06-3	2014-09-06 10:10:10
5	lambda-2014-09-06-4	sg-0000000000000004	lambda-2014-09-06-4	0.0.0.0/0	lambda-2014-09-06-4	2014-09-06 10:10:10
6	lambda-2014-09-06-5	sg-0000000000000005	lambda-2014-09-06-5	0.0.0.0/0	lambda-2014-09-06-5	2014-09-06 10:10:10
7	lambda-2014-09-06-6	sg-0000000000000006	lambda-2014-09-06-6	0.0.0.0/0	lambda-2014-09-06-6	2014-09-06 10:10:10

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



6. Add the rules as given in the screenshot below



7. Connect the instance



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

EC2 Instance Connect	Session Manager	SSH client	EC2 serial console
Instance ID			
<input type="checkbox"/>	i-0b21c79e1e222bc9d (nagios-host-1)		
1.	Open an SSH client.		
2.	Locate your private key file. The key used to launch this instance is <code>nagios_practical.pem</code> .		
3.	Run this command, if necessary, to ensure your key is not publicly viewable.		
	<input type="checkbox"/> <code>chmod 400 "nagios_practical.pem"</code>		
4.	Connect to your instance using its Public DNS:		
	<input type="checkbox"/> <code>ec2-34-230-73-94.compute-1.amazonaws.com</code>		
Example:			
	<input type="checkbox"/> <code>ssh -i "nagios_practical.pem" ec2-user@ec2-34-230-73-94.compute-1.amazonaws.com</code>		
① Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.			
Cancel			

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

10. Now install the following packages using yum:

`sudo yum update`

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

```
sudo yum install httpd php
```

```
[ec2-user@ip-172-31-87-75 ~]$ sudo yum install httpd php
Last metadata expiration check: 0:31:23 ago on Tue Oct 3 18:40:44 2024.
Dependencies resolved.
=====
Transaction Summary
=====
Installed:
  httpd           x86_64    2.4.62-1.amzn2023.0.1
  php8.1          x86_64    8.1.18-1.amzn2023.0.1
  Installing dependencies:
    apr            x86_64    1.7.3-2.amzn2023.0.1
    apr-util        x86_64    1.6.3-1.amzn2023.0.1
    generic-https   noarch   14.0.0-12.amzn2023.0.1
    httpd-cave     x86_64    2.4.62-1.amzn2023.0.1
    httpsys        noarch   2.4.62-1.amzn2023.0.1
    libevent       x86_64    1.0.9-4.amzn2023.0.1
    libev          x86_64    1.0.19-6.amzn2023.0.1
    libevasio      x86_64    1.1.14-6.amzn2023.0.1
    libedit        x86_64    2.1.89-2.amzn2023.0.1
    libxml         noarch   1.1.24.0-1.amzn2023.0.1
    stat64-bit    x86_64    0.3.10-1.amzn2023.0.1
    stdc89-common  x86_64    0.3.10-1.amzn2023.0.1
    stdc89-process x86_64    0.3.10-1.amzn2023.0.1
    stdc89-devel   x86_64    0.3.10-1.amzn2023.0.1
  Installing weak dependencies:
    apr-util-memset x86_64    1.6.3-1.amzn2023.0.1
    mod_http2      x86_64    2.0.27-1.amzn2023.0.1
    mod_lsapi      x86_64    2.4.62-1.amzn2023.0.1
    stdc89-fp     x86_64    0.3.10-1.amzn2023.0.1
    stdc89-memctrl x86_64    0.3.10-1.amzn2023.0.1
    stdc89-pco     x86_64    0.3.10-1.amzn2023.0.1
    stdc89-random  x86_64    0.3.10-1.amzn2023.0.1
  Transaction Summary
  =====
  Installed:
    httpd           x86_64    2.4.62-1.amzn2023.0.1
    php8.1          x86_64    8.1.18-1.amzn2023.0.1
  apr             x86_64    1.7.3-2.amzn2023.0.1
  apr-util        x86_64    1.6.3-1.amzn2023.0.1
  generic-https   noarch   14.0.0-12.amzn2023.0.1
  httpd-cave     x86_64    2.4.62-1.amzn2023.0.1
  httpsys        noarch   2.4.62-1.amzn2023.0.1
  libevent       x86_64    1.0.9-4.amzn2023.0.1
  libev          x86_64    1.0.19-6.amzn2023.0.1
  libevasio      x86_64    1.1.14-6.amzn2023.0.1
  libedit        x86_64    2.1.89-2.amzn2023.0.1
  libxml         noarch   1.1.24.0-1.amzn2023.0.1
  stat64-bit    x86_64    0.3.10-1.amzn2023.0.1
  stdc89-common  x86_64    0.3.10-1.amzn2023.0.1
  stdc89-process x86_64    0.3.10-1.amzn2023.0.1
  stdc89-devel   x86_64    0.3.10-1.amzn2023.0.1
  apr-util-memset x86_64    1.6.3-1.amzn2023.0.1
  mod_http2      x86_64    2.0.27-1.amzn2023.0.1
  mod_lsapi      x86_64    2.4.62-1.amzn2023.0.1
  stdc89-fp     x86_64    0.3.10-1.amzn2023.0.1
  stdc89-memctrl x86_64    0.3.10-1.amzn2023.0.1
  stdc89-pco     x86_64    0.3.10-1.amzn2023.0.1
  stdc89-random  x86_64    0.3.10-1.amzn2023.0.1
  Transaction Summary
  =====
  Complete!
```

sudo yum install gcc glibc glibc-common

```
[ec2-user@ip-172-31-87-75 ~]$ sudo yum install gcc glibc glibc-common
Last metadata expiration check: 0:31:23 ago on Tue Oct 3 18:40:44 2024.
Package glibc-2.34-52.amzn2023.0.11.x86_64 is already installed.
Package glibc-common-2.34-52.amzn2023.0.11.x86_64 is already installed.
Dependencies resolved.
=====
Transaction Summary
=====
Installed:
  gcc             x86_64    11.4.0-2.amzn2023.0.1
  glibc           x86_64    2.34-52.amzn2023.0.11
  glibc-common    x86_64    2.34-52.amzn2023.0.11
  libgcc          x86_64    11.4.0-2.amzn2023.0.1
  libgcc_s        x86_64    11.4.0-2.amzn2023.0.1
  libstdc++       x86_64    11.4.0-2.amzn2023.0.1
  libstdc++-devel x86_64    11.4.0-2.amzn2023.0.1
  libgccabi       x86_64    2.34-52.amzn2023.0.1
  libgccabi-devel x86_64    2.34-52.amzn2023.0.1
  libgccabi-headers x86_64   2.34-52.amzn2023.0.1
  libgccabi-headers-devel x86_64  2.34-52.amzn2023.0.1
  libgccabi-headers-x86_64 x86_64  2.34-52.amzn2023.0.1
  libgccabi-x86_64 x86_64   2.34-52.amzn2023.0.1
  libgcc_s         x86_64    2.34-52.amzn2023.0.1
  libgcc_s-devel   x86_64    2.34-52.amzn2023.0.1
  libgcc_s-headers x86_64    2.34-52.amzn2023.0.1
  libgcc_s-headers-devel x86_64  2.34-52.amzn2023.0.1
  libgcc_s-x86_64 x86_64   2.34-52.amzn2023.0.1
  libstdc++-devel x86_64    11.4.0-2.amzn2023.0.1
  libstdc++-headers x86_64   11.4.0-2.amzn2023.0.1
  libstdc++-headers-devel x86_64  11.4.0-2.amzn2023.0.1
  libstdc++-headers-x86_64 x86_64  11.4.0-2.amzn2023.0.1
  libstdc++-x86_64 x86_64   11.4.0-2.amzn2023.0.1
  Transaction Summary
  =====
  Complete!
```

sudo yum install gd gd-devel

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

```
sudo adduser -m nagios  
sudo passwd nagios
```

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

12. Create a new user group

```
sudo groupadd nagcmd
```

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

13. Next execute these commands so that you don't have to use sudo for Apache and Nagios:

```
sudo usermod -a -G nagcmd nagios
```

```
sudo usermod -a -G nagcmd apache
```

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

14. Create a new directory for Nagios downloads

```
mkdir ~/downloads
cd ~/downloads
```

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

15. Use wget to download the source zip files.

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

```
[ec2-user@ip-172-31-87-75 downloads]$ wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
--2024-10-01 15:55:47-- https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.128, 2606:3c00::f03c:92ff:feff:45ce
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.128|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2065473 (2.0M) [application/x-gzip]
Saving to: 'nagios-4.5.5.tar.gz'

nagios-4.5.5.tar.gz          100%[=====] 1.97M  5.54MB/s   in 0.4s
2024-10-01 15:55:47 (5.54 MB/s) - 'nagios-4.5.5.tar.gz' saved [2065473/2065473]
[ec2-user@ip-172-31-87-75 downloads]$ |
```

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

```
[ec2-user@ip-172-31-87-75 downloads]$ wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
--2024-10-01 15:57:19-- https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251
Connecting to nagios-plugins.org (nagios-plugins.org)|45.56.123.251|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2753049 (2.6M) [application/x-gzip]
Saving to: 'nagios-plugins-2.4.11.tar.gz'

nagios-plugins-2.4.11.tar.gz      100%[=====] 2.62M  4.14MB/s   in 0.6s
2024-10-01 15:57:20 (4.14 MB/s) - 'nagios-plugins-2.4.11.tar.gz' saved [2753049/2753049]
[ec2-user@ip-172-31-87-75 downloads]$ |
```

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

```
tar zxvf nagios-4.5.5.tar.gz
```

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

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

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

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

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

19. Now we will install openssl dev library by using the command:

```
sudo yum install openssl-devel
```

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

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

20. Run the configuration script

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

```
[ec2-user@ip-172-31-87-75 nagios-4.5.5]$ ./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether the compiler supports GNU C... yes
checking whether gcc accepts -g... yes
checking for gcc option to enable C11 features... none needed
checking whether make sets $(MAKE)... yes
checking whether ln -s works... yes
checking for strip... /usr/bin/strip
checking for sys/wait.h that is POSIX.1 compatible... yes
checking for stdio.h... yes
checking for stdlib.h... yes
checking for string.h... yes
```

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

General Options:

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

Web Interface Options:

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

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

```
[ec2-user@ip-172-31-87-75 nagios-4.5.5]$ make all
cd ./base && make
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
gcc -Wall -I.. -I./lib -I./include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o nagios.o ./nagios.c
gcc -Wall -I.. -I.. -I./lib -I./include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o broker.o broker.c
gcc -Wall -I.. -I.. -I./lib -I./include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o nebmods.o nebmods.c
gcc -Wall -I.. -I.. -I./lib -I./include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o ..//common/shared.o ..//common/shared.c
gcc -Wall -I.. -I.. -I./lib -I./include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o query-handler.o query-handler.c
gcc -Wall -I.. -I.. -I./lib -I./include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o workers.o workers.c
In function 'get_prcp_list',
  inlined from 'get_worker' at workers.c:277:12:
workers.c:253:17: warning: '%s' directive argument is null [-Wformat-overflow=]
  253 |         log_debug_info(DEBUGL_CHECKS, 1, "Found specialized worker(s) for '%s'", (slash @@ *slash != '/') ? slash : cmd_name);
gcc -Wall -I.. -I.. -I./lib -I./include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o checks.o checks.c
gcc -Wall -I.. -I.. -I./lib -I./include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o config.o config.c
gcc -Wall -I.. -I.. -I./lib -I./include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o commands.o commands.c
gcc -Wall -I.. -I.. -I./lib -I./include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o events.o events.c
gcc -Wall -I.. -I.. -I./lib -I./include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o flapping.o flapping.c
gcc -Wall -I.. -I.. -I./lib -I./include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o logging.o logging.c
gcc -Wall -I.. -I.. -I./lib -I./include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o macros-base.o ..//common/macros.c
gcc -Wall -I.. -I.. -I./lib -I./include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o netutils.o netutils.c
gcc -Wall -I.. -I.. -I./lib -I./include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o notifications.o notifications.c
gcc -Wall -I.. -I.. -I./lib -I./include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o sehandlers.o sehandlers.c
gcc -Wall -I.. -I.. -I./lib -I./include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o utils.o utils.c
gcc -Wall -I.. -I.. -I./lib -I./include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o retention-base.o ./retention.c
gcc -Wall -I.. -I.. -I./lib -I./include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o xretention-base.o ..//xdata/xrddefaul.c
gcc -Wall -I.. -I.. -I./lib -I./include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o comments-base.o ..//common/comments.c
```

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

sudo make install

```
sudo make install-init
```

```
sudo make install-config
```

sudo make install-commandmode

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

23. In the config file edit the email address

```
sudo nano /usr/local/nagios/etc/objects/contacts.cfg
```

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

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

define contact {
    contact_name          nagiosadmin           ; Short name of user
    use                   generic-contact        ; Inherit default values from generic-contact template (defined above)
    alias                Nagios Admin          ; Full name of user
    email                d2022.nayaab.jindani@ves.ac.in ; <***** CHANGE THIS TO YOUR EMAIL ADDRESS *****
}
```

24. To configure the web interface run:

```
sudo make install-webconf
```

```
[ec2-user@ip-172-31-87-75 nagios-4.5.5]$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf
if [ 0 -eq 1 ]; then \
    ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagios.conf
fi

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

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

25. Create a nagios admin account and password

```
sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

```
[ec2-user@ip-172-31-87-75 nagios-4.5.5]$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
```

26. Restart apache

```
sudo service httpd restart
```

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

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

```
tar zxvf nagios-plugins-2.4.11.tar.gz
```

```
[ec2-user@ip-172-31-87-75 nagios-4.5.5]$ cd ~/downloads
tar zxvf nagios-plugins-2.4.11.tar.gz
nagios-plugins-2.4.11/
nagios-plugins-2.4.11/build-aux/
nagios-plugins-2.4.11/build-aux/compile
nagios-plugins-2.4.11/build-aux/config.guess
nagios-plugins-2.4.11/build-aux/config.rpath
nagios-plugins-2.4.11/build-aux/config.sub
nagios-plugins-2.4.11/build-aux/install-sh
nagios-plugins-2.4.11/build-aux/ltmain.sh
nagios-plugins-2.4.11/build-aux/missing
nagios-plugins-2.4.11/build-aux/mkinstalldirs
nagios-plugins-2.4.11/build-aux/depcomp
nagios-plugins-2.4.11/build-aux/snippet/
nagios-plugins-2.4.11/build-aux/snippet/_Noreturn.h
nagios-plugins-2.4.11/build-aux/snippet/arg-nonnull.h
nagios-plugins-2.4.11/build-aux/snippet/c++defs.h
nagios-plugins-2.4.11/build-aux/snippet/warn-on-use.h
nagios-plugins-2.4.11/build-aux/test-driver
nagios-plugins-2.4.11/config_test/
nagios-plugins-2.4.11/config_test/Makefile
nagios-plugins-2.4.11/config_test/run_tests
nagios-plugins-2.4.11/config_test/child_test.c
nagios-plugins-2.4.11/gl/
```

28. Compile and install the plugins

```
cd nagios-plugins-2.0.3
```

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

```
[ec2-user@ip-172-31-87-75 downloads]$ cd nagios-plugins-2.4.11
./configure --with-nagios-user=nagios --with-nagios-group=nagios
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether to enable maintainer-specific portions of Makefiles
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
```

make

sudo make install

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

```

29. Run below given commands to start nagios:

sudo chkconfig --add nagios

sudo chkconfig nagios on

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

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

```
[ec2-user@ip-172-31-87-75 nagios-plugins-2.4.11]$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagi  
Nagios Core 4.5.5  
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors  
Copyright (c) 1999-2009 Ethan Galstad  
Last Modified: 2024-09-17  
License: GPL  
  
Website: https://www.nagios.org  
Reading configuration data...  
  Read main config file okay...  
  Read object config files okay...  
  
Running pre-flight check on configuration data...  
  
Checking objects...  
  Checked 8 services.  
  Checked 1 hosts.  
  Checked 1 host groups.  
  Checked 0 service groups.  
  Checked 1 contacts.  
  Checked 1 contact groups.  
  Checked 24 commands.  
  Checked 5 time periods.  
  Checked 0 host escalations.  
  Checked 0 service escalations.  
Checking for circular paths...  
  Checked 1 hosts  
  Checked 0 service dependencies  
  Checked 0 host dependencies  
  Checked 5 timeperiods  
Checking global event handlers...  
Checking obsessive compulsive processor commands...  
Checking misc settings...  
  
Total Warnings: 0  
Total Errors: 0  
  
Things look okay - No serious problems were detected during the pre-flight check  
[ec2-user@ip-172-31-87-75 nagios-plugins-2.4.11]$ |
```

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

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

sudo systemctl status nagios

```

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

Total Warnings: 0
Total Errors: 0

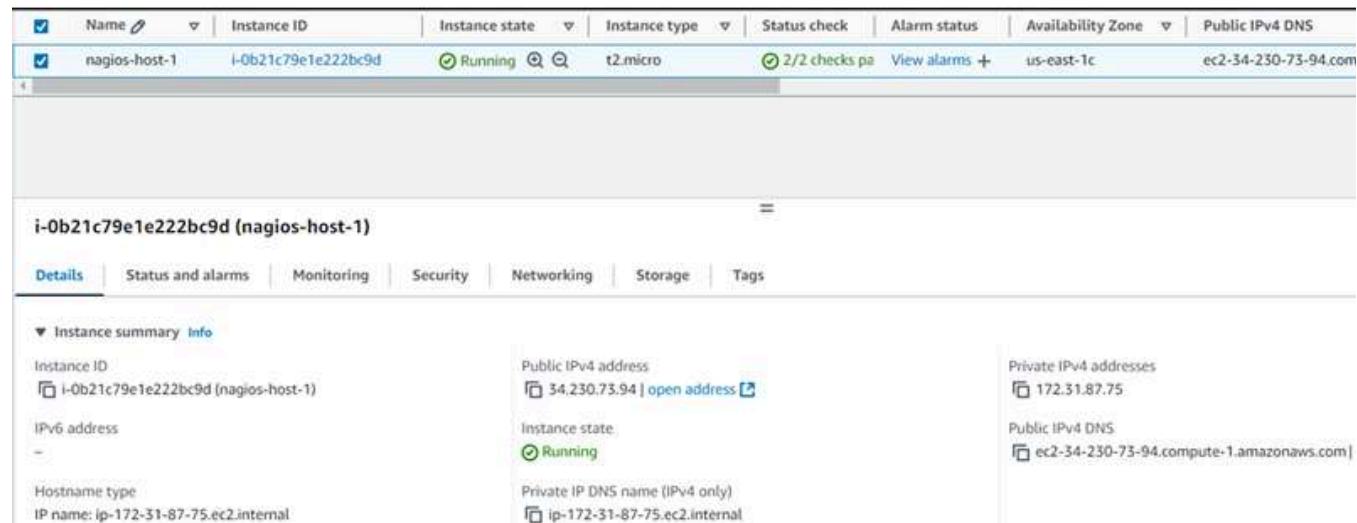
Things look okay - No serious problems were detected during the pre-flight check
[ec2-user@ip-172-31-87-75 nagios-plugins-2.4.11]$ sudo service nagios start
Redirecting to /bin/systemctl start nagios.service
[ec2-user@ip-172-31-87-75 nagios-plugins-2.4.11]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
    Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
    Active: active (running) since Tue 2024-10-01 16:32:27 UTC; 48s ago
      Docs: https://www.nagios.org/documentation
   Process: 66684 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
  Process: 66693 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 66694 (nagios)
    Tasks: 6 (limit: 1112)
   Memory: 5.7M
      CPU: 82ms
     CGroup: /system.slice/nagios.service
             └─66694 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
                  ├─66695 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                  ├─66696 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                  ├─66697 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                  ├─66698 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                  └─66739 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

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

```

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

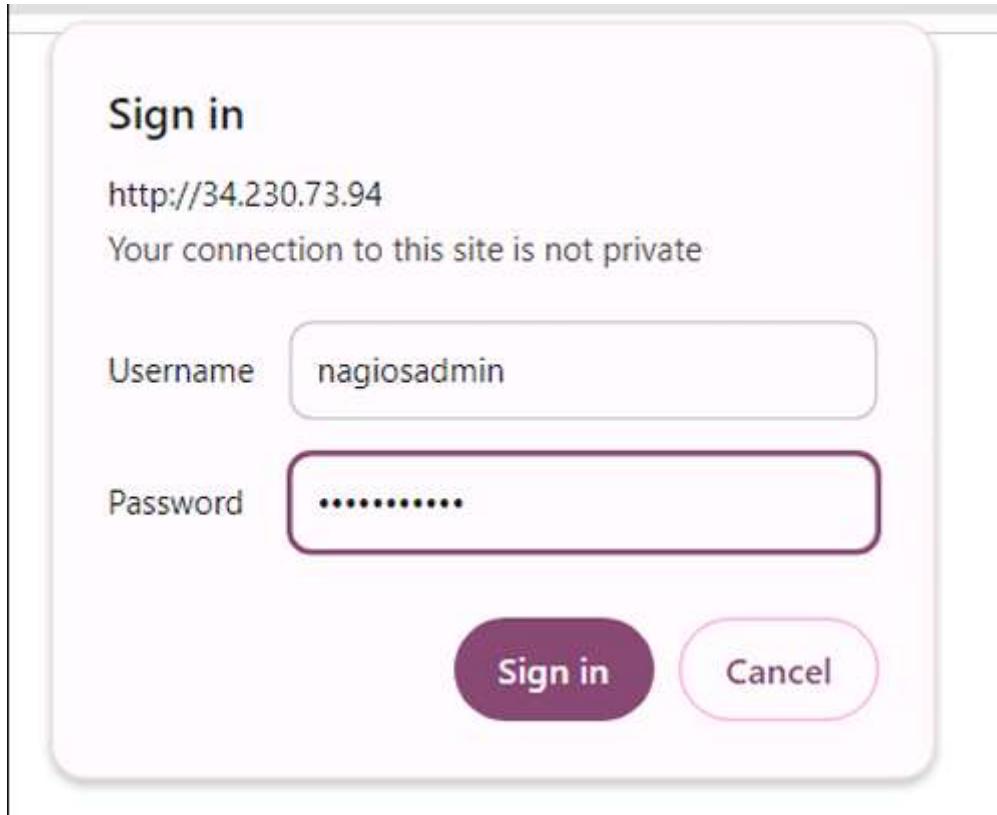
30. Go back to your instances and copy the public IPv4 address



31. Lastly, go to your web browser and type “<http://<public-IPv4-address>/nagios>”

Replace public-IPv4-address with the public ip address of your instance which you copied.

You will get a prompt to enter the username and password that have been set for nagios admin in step 25.



You will see the below shown page after entering credentials.

Conclusion: In this experiment, the primary challenge I encountered was accessing the Nagios web interface due to a "Forbidden: You do not have permission to access this resource" error. This issue was resolved by modifying the inbound security rules and verifying that all necessary files were installed in the correct directories. Additionally, it is crucial to restart Apache after making any changes to ensure they take effect. Once Nagios was activated without any errors, the output was successfully displayed on the web interface.

Experiment 10

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

Nagios. Steps:

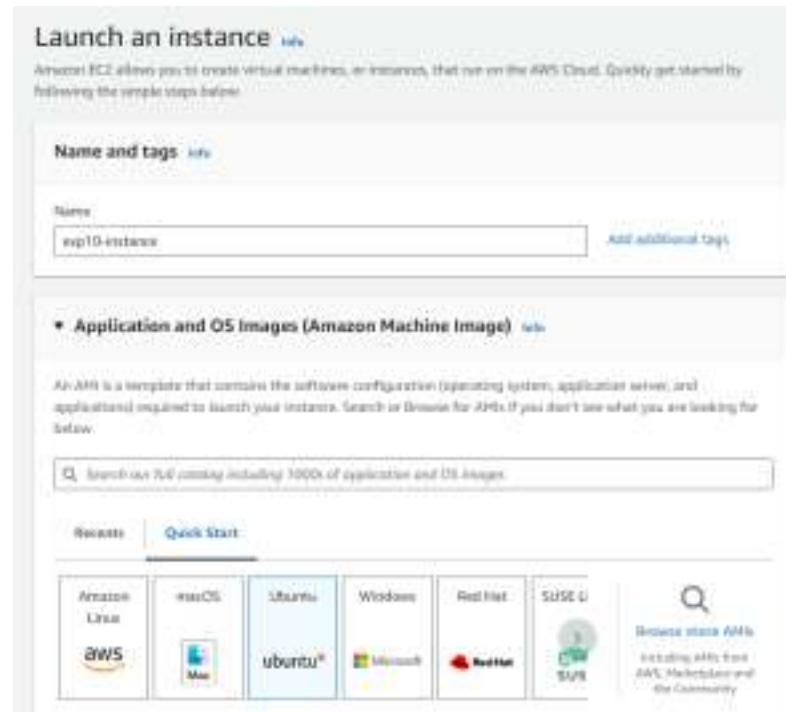
1. Firstly, we will check whether nagios is running on the server side by using the command “sudo systemctl status nagios” on the host machine (host machine is the instance connected to the terminal in experiment 9, ensure that you have started the instance created for exp9, also check status of apache).

```
[ec2-user@ip-172-31-87-75 ~]$ sudo systemctl status
nagios
● ip-172-31-87-75.ec2.internal
  State: running
    Units: 295 loaded (incl. loaded aliases)
      Jobs: 1 queued
     Failed: 0 units
      Since: Wed 2024-10-02 06:17:29 UTC; 2min 42s ago
    systemd: 252.23-2.amzn2023
    CC=us, /
```

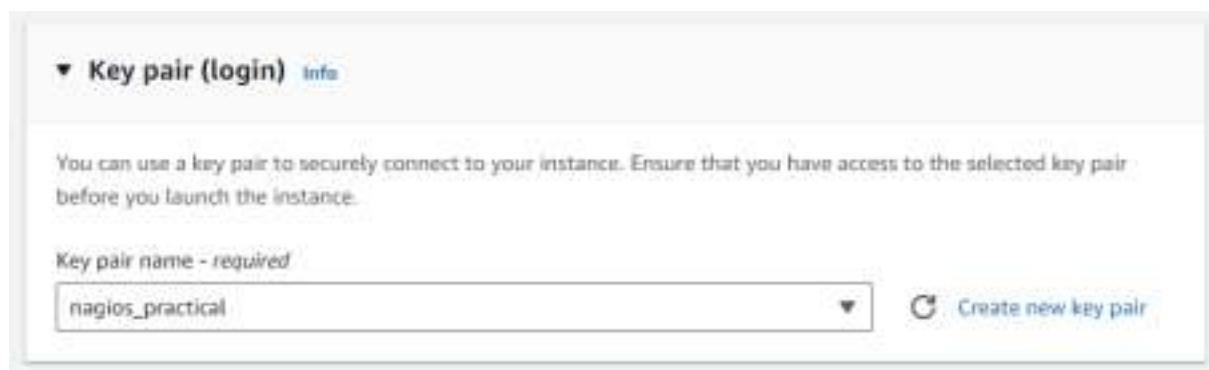


```
[ec2-user@ip-172-31-87-75 ~]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-87-75 ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service;
  Drop-In: /usr/lib/systemd/system/httpd.service.d
            └─php-fpm.conf
  Active: active (running) since Wed 2024-10-02 06:26:51
    Docs: man:httpd.service(8)
  Main PID: 3242 (httpd)
    Status: "Started, listening on: port 80"
      Tasks: 177 (limit: 1112)
     Memory: 13.1M
        CPU: 47ms
      CGroup: /system.slice/httpd.service
              ├─3242 /usr/sbin/httpd -DFOREGROUND
              ├─3243 /usr/sbin/httpd -DFOREGROUND
              ├─3244 /usr/sbin/httpd -DFOREGROUND
              ├─3245 /usr/sbin/httpd -DFOREGROUND
              ├─3246 /usr/sbin/httpd -DFOREGROUND
```

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



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



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

Network [Info](#)

vpc-0d1089189551d9d25

Subnet [Info](#)

No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

Common security groups [Info](#)

Select security groups

launch-wizard-32 sg-0588f70648d484edd X

VPC: vpc-0d1089189551d9d25

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

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

```
PS C:\Users\BELL> ssh -i "C:\Users\BELL\Downloads\nagios_practical.pem" ubuntu@ec2-18-207-191-28.compute-1.amazonaws.com
The authenticity of host 'ec2-18-207-191-28.compute-1.amazonaws.com (18.207.191.28)' can't be established.
ED25519 key fingerprint is SHA256:NPJP0FwGZXU0XNkQ9aw/1zAFX0nabRJiCiAfdyYi0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-207-191-28.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1816-ans x86_64)

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

System information as of Wed Oct  3 06:36:18 UTC 2024

System load: 0.18      Processes:          196
Usage of /: 22.9% of 6.71GB   Users logged in:   8
Memory usage: 21%           IPv4 address for enx8: 172.31.48.136
Swap usage: 0%
```

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

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

```
ps -ef | grep nagios
```

```
[ec2-user@ip-172-31-87-75 ~]$ ps -ef | grep nagios
nagios   2003      1  0 86:17 ?    00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios   2002      2  0 86:17 ?    00:00:00 /usr/local/nagios/bin/nagios --master /usr/local/nagios/var/rw/nagios.oh
nagios   2004      2  0 86:17 ?    00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.oh
nagios   2005      2  0 86:17 ?    00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.oh
nagios   2006      2  0 86:17 ?    00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.oh
nagios   2007      2  0 86:17 ?    00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
ec2-user  2000      2  0 86:03 pts/0    00:00:00 qprep --coloramae nagios
[ec2-user@ip-172-31-87-75 ~]$
```

7. Now perform these commands on the host terminal

sudo su

mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts

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

cp /usr/local/nagios/etc/objects/localhost.cfg

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

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

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

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

- Change the hostgroup name to linux-servers1

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

# Define an optional hostgroup for Linux machines

define hostgroup {
        hostgroup_name      Linux-servers1           ; The name of the hostgroup
        alias                Linux Servers            ; long name of the group
        members              Linuxserver             ; Comma separated list of hosts that belong to this group
}
```

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

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

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

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

```
# Define a host for the local machine
define host {
    use           Linux-server          ; Name of host template to use
                                         ; This host definition will inherit all variables that are defined
                                         ; in (or inherited by) the linux-server host-template definition.
    host_name     Linuxserver
    alias         Linuxserver
    address       18.207.192.20
}
```

8. Open the Nagios Config file by using this command:

```
nano /usr/local/nagios/etc/nagios.cfg
nano text editor will get opened
```

```
#####
#
# NAGIOS.CFG - Sample Main Config File for Nagios 4.5.5
#
# Read the documentation for more information on this configuration
# file. I've provided some comments here, but things may not be so
# clear without further explanation.
#
#
#####
#
# LOG FILE
# This is the main log file where service and host events are logged
# for historical purposes. This should be the first option specified
# in the config file!!!
log_file=/usr/local/nagios/var/nagios.log

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

# You can specify individual object config files as shown below:
cfg_file=/usr/local/nagios/etc/objects/commands.cfg
cfg_file=/usr/local/nagios/etc/objects/contacts.cfg
cfg_file=/usr/local/nagios/etc/objects/timeperiods.cfg
cfg_file=/usr/local/nagios/etc/objects/templates.cfg

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

# Definitions for monitoring a Windows machine
```

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

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

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

10. Now we will verify the configuration files

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

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

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

Running pre-flight check on configuration data...

Checking objects...
  Checked 16 services.
  Checked 2 hosts.
  Checked 2 host groups.
  Checked 8 service groups.
  Checked 1 contacts.
  Checked 1 contact groups.
  Checked 24 commands.
  Checked 5 time periods.
  Checked 8 host escalations.
  Checked 8 service escalations.
Checking for circular paths...
  Checked 2 hosts
  Checked 8 service dependencies
  Checked 8 host dependencies
  Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors: 0

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

If there are no errors we can proceed further

11. We will now restart the nagios service

```
service nagios restart
```

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

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

sudo apt update -y

```
ubuntu@ip-172-31-40-130:~$ sudo apt update -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [380 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.8 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [535 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [130 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [8676 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [380 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [156 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [45.0 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [14.9 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [14.4 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [3608 B]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [212 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [532 B]
```

sudo apt install gcc -y

```
ubuntu@ip-172-31-40-130:~$ sudo apt install gcc -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86_64-linux-gnu cpp cpp-13 cpp-13-x86_64-linux-gnu cpp-x86_64-linux-gnu fontconfig-config fonts-dejavu-core
  fonts-dejavu-mono gcc-13 gcc-13-base gcc-13-x86_64-linux-gnu glibaom libasan8 libatomic libbinutils libc-dev-bin libc-dev-tools
  libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libde265-0 libdeflate0 libfontconfig1 libgcc-13-dev libgd3 libgnoml1 libprofg9
  libheif-plugin-aomdec libheif-plugin-gmenc libheif-plugin-libde265 libheif1 libwasan0 libis23 libitm1 libjbig0 libjpeg-turbo8 libjpeg8 liblerc4
  liblsan8 libmpc3 libquadmath0 libsharpframe1 libsharpuyv0 libtiff6 libtsan2 libubsan1 libwebp7 libxml2 linux-lhc-dev manpages-dev rpcsvc-proto
Suggested packages:
  binutils-doc gprofng-gui cpp-doc gcc-13-locales cpp-13-doc gcc-multilib make autoconf automake libtool flex bison gdb gcc-doc gcc-13-multilib gcc-13-doc
  gdb-x86_64-linux-gnu glibc-doc libgdg-tools libheif-plugin-x265 libheif-plugin-ffmpegdec libheif-plugin-jpegdec libheif-plugin-jpegenc
  libheif-plugin-j2kdec libheif-plugin-j2kenc libheif-plugin-rav1e libheif-plugin-svtenc
The following NEW packages will be installed:
  binutils binutils-common binutils-x86_64-linux-gnu cpp cpp-13 cpp-13-x86_64-linux-gnu cpp-x86_64-linux-gnu fontconfig-config fonts-dejavu-core
  fonts-dejavu-mono gcc-13 gcc-13-base gcc-13-x86_64-linux-gnu glibaom libasan8 libatomic libbinutils libc-dev-bin libc-dev-tools
  libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libde265-0 libdeflate0 libfontconfig1 libgcc-13-dev libgd3 libgnoml1 libprofg9
  libheif-plugin-aomdec libheif-plugin-gmenc libheif-plugin-libde265 libheif1 libwasan0 libis23 libitm1 libjbig0 libjpeg-turbo8 libjpeg8 liblerc4
  liblsan8 libmpc3 libquadmath0 libsharpuyv0 libtiff6 libtsan2 libubsan1 libwebp7 libxml2 linux-lhc-dev manpages-dev rpcsvc-proto
0 upgraded, 57 newly installed, 0 to remove and 6 not upgraded.
Need to get 62.8 MB of archives.
After this operation, 222 MB of additional disk space will be used.
```

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

```
ubuntu@ip-172-31-40-130:~$ sudo apt install -y nagios-nrpe-server nagios-plugins
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'monitoring-plugins' instead of 'nagios-plugins'
The following additional packages will be installed:
  libavahi-client3 libavahi-common-data libavahi-common3 libcurls2t64 libdbilt64 libldb2 libmysqlclient21 libnet-snmp-perl libpq5 libradcli4 libsmbclient0
  libsnmp-base libsnmp40t64 libtalloc2 libtdb1 libtevent0t64 liburiparser1 libwbcclient0 monitoring-plugins-basic monitoring-plugins-common
  monitoring-plugins-standard mysql-common python3-gpg python3-ldb python3-markdown python3-samba python3-talloc python3-tdb rpcbind samba-common
  samba-common-bin samba-dsdb-modules samba-libs smbclient snmp
Suggested packages:
  cups-common libcrypt-des-perl libdigest-hmac-perl libio-socket-inet6-perl snmp-mibs-downloader icinga2 nagios-plugins-contrib fping postfix
  | sendmail-bin | exim4-daemon-heavy | exim4-daemon-light qstat xinetd | inetd python-markdown-doc heimdal-clients python3-dnspython cifs-utils
The following NEW packages will be installed:
  libavahi-client3 libavahi-common-data libavahi-common3 libcurls2t64 libdbilt64 libldb2 libmysqlclient21 libnet-snmp-perl libpq5 libradcli4 libsmbclient0
  libsnmp-base libsnmp40t64 libtalloc2 libtdb1 libtevent0t64 liburiparser1 libwbcclient0 monitoring-plugins monitoring-plugins-basic
  monitoring-plugins-common monitoring-plugins-standard mysql-common nagios-nrpe-server python3-gpg python3-ldb python3-markdown python3-samba
  python3-talloc python3-tdb rpcbind samba-common samba-common-bin samba-dsdb-modules samba-libs smbclient snmp
0 upgraded, 37 newly installed, 0 to remove and 6 not upgraded.
Need to get 16.1 MB of archives.
After this operation, 72.0 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 nagios-nrpe-server amd64 4.1.0-1ubuntu3 [356 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu/noble/main amd64 rpcbind amd64 1.2.6-7ubuntu2 [46.5 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu/noble/main amd64 libavahi-common-data amd64 0.8-13ubuntu6 [29.7 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu/noble/main amd64 libavahi-common3 amd64 0.8-13ubuntu6 [23.3 kB]
```

13. Open nrpe.cfg file to make changes.

sudo nano /etc/nagios/nrpe.cfg

Under allowed_hosts, add your nagios host public IPv4 address:

```

#
# Note: The daemon only does rudimentary checking of the client's IP
# address. I would highly recommend adding entries in your /etc/hosts.allow
# file to allow only the specified host to connect to the port
# you are running this daemon on.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd
allowed_hosts=127.0.0.1,54.163.184.143

# COMMAND ARGUMENT PROCESSING
# This option determines whether or not the NRPE daemon will allow clients
# to specify arguments to commands that are executed. This option only works
# if the daemon was configured with the --enable-command-args configure script
# option.
#
# *** ENABLING THIS OPTION IS A SECURITY RISK! ***
# Read the SECURITY file for information on some of the security implications
# of enabling this variable.
#
# Values: 0=do not allow arguments, 1=allow command arguments

```

14. Now restart the NRPE server

sudo systemctl restart nagios-nrpe-server

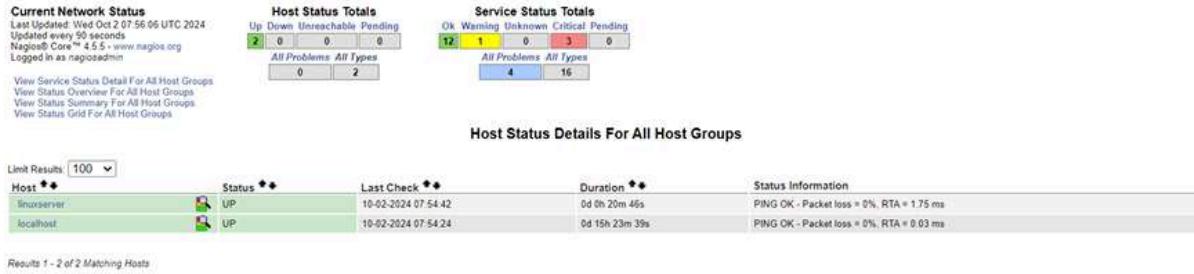
```

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

```

15. Go to the nagios dashboard and click on hosts

Click on linux server



We can see the host state information:

Host Information

Last Updated: Wed Oct 2 07:59:17 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.5.5 - www.nagios.org
Logged in as nagiosadmin

Host
linuxserver
(linuxserver)

Member of
linux-servers1

18.207.191.20

Host State Information

Host Status:	UP (for 0d 0h 23m 57s)
Status Information:	PING OK - Packet loss = 0%, RTA = 1.75 ms
Performance Data:	rta=1.748000ms;3000.000000;5000.000000;0.000000 pl=0%;80,100,0
Current Attempt:	1/10 (HARD state)
Last Check Time:	10-02-2024 07:54:42
Check Type:	ACTIVE
Check Latency / Duration:	0.001 / 4.012 seconds
Next Scheduled Active Check:	10-02-2024 07:59:42
Last State Change:	10-02-2024 07:35:20
Last Notification:	N/A (notification 0)
Is This Host Flapping?	NO (0.00% state change)
In Scheduled Downtime?	NO
Last Update:	10-02-2024 07:59:09 (0d 0h 0m 8s ago)
Active Checks:	ENABLED
Passive Checks:	ENABLED
Obsessing:	ENABLED
Notifications:	ENABLED
Event Handler:	ENABLED
Flap Detection:	ENABLED

Host Commands

- Locate host on map
- Disable active checks of this host
- Re-schedule the next check of this host
- Submit passive check result for this host
- Stop accepting passive checks for this host
- Stop obsessing over this host
- Disable notifications for this host
- Send custom host notification
- Schedule downtime for this host
- Schedule downtime for all services on this host
- Disable notifications for all services on this host
- Enable notifications for all services on this host
- Schedule a check of all services on this host
- Disable checks of all services on this host
- Enable checks of all services on this host
- Disable event handler for this host
- Disable flap detection for this host
- Clear flapping state for this host

Host Comments

Add a new comment Delete all comments

Entry Time Author Comment Comment ID Persistent Type Expires Actions

This host has no comments associated with it.

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

Current Network Status

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

Host Status Totals

Up	Down	Unreachable	Pending
2	0	0	0
All Problems	All Types		
0	2		

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
12	1	0	3	0
All Problems	All Types			
4	16			

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

Service Status Details For All Hosts

Limit Results: 100 ▾

Host **	Service **	Status **	Last Check **	Duration **	Attempt **	Status Information
linuxserver	Current Load	OK	10-02-2024 07:55:57	0d 0h 22m 4s	1/4	OK - load average: 0.00, 0.00, 0.00
	Current Users	OK	10-02-2024 07:56:35	0d 0h 21m 26s	1/4	USERS OK - 3 users currently logged in
	HTTP	CRITICAL	10-02-2024 07:55:12	0d 0h 17m 49s	4/4	connect to address 18.207.191.26 and port 80: Connection refused
	PING	OK	10-02-2024 07:57:50	0d 0h 20m 11s	1/4	PING OK - Packet loss = 0%, RTA = 2.11 ms
	Root Partition	OK	10-02-2024 07:53:27	0d 0h 19m 34s	1/4	DISK OK - free space: / 6114 MB (75.33% inode=98%)
	SSH	OK	10-02-2024 07:54:05	0d 0h 18m 56s	1/4	SSH OK - OpenSSH_9_6p1 Ubuntu-Subuntu13.5 (protocol 2.0)
	Swap Usage	CRITICAL	10-02-2024 07:57:42	0d 0h 15m 19s	4/4	SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size
	Total Processes	OK	10-02-2024 07:55:20	0d 0h 17m 41s	1/4	PROCS OK: 38 processes with STATE = RSZDT
localhost	Current Load	OK	10-02-2024 07:53:09	0d 1h 24m 57s	1/4	OK - load average: 0.00, 0.00, 0.00
	Current Users	OK	10-02-2024 07:53:47	0d 1h 24m 19s	1/4	USERS OK - 3 users currently logged in
	HTTP	WARNING	10-02-2024 07:54:24	0d 1h 28m 37s	4/4	HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.000 second response time
	PING	OK	10-02-2024 07:55:02	0d 1h 23m 4s	1/4	PING OK - Packet loss = 0%, RTA = 0.03 ms
	Root Partition	OK	10-02-2024 07:55:39	0d 1h 22m 27s	1/4	DISK OK - free space: / 6114 MB (75.33% inode=98%)
	SSH	OK	10-02-2024 07:56:17	0d 1h 21m 49s	1/4	SSH OK - OpenSSH_8.7 (protocol 2.0)
	Swap Usage	CRITICAL	10-02-2024 07:56:54	0d 1h 18m 12s	4/4	SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size
	Total Processes	OK	10-02-2024 07:57:32	0d 1h 20m 34s	1/4	PROCS OK: 38 processes with STATE = RSZDT

Results 1 - 16 of 16 Matching Services

Conclusion: To conduct this experiment, it's necessary to start the instance from the previous experiment, as it will serve as the host, while the instance created in this experiment will act as the client machine. When I attempted to run the command to verify the Nagios configuration file, I encountered errors. To resolve these errors, I reinstalled the Nagios plugins and restarted the Nagios service, which fixed the issues.

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

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

The screenshot shows the AWS Lambda 'Create function' wizard. At the top, there are four options: 'Author from scratch' (selected), 'Use a blueprint', 'Container image', and 'Browse serverless app repository'. Below this is a 'Basic information' section. In the 'Function name' field, 'lambdaexp11' is entered. Under 'Runtime', 'Python 3.12' is selected. In the 'Architecture' section, 'x86_64' is chosen. The 'Permissions' section notes that Lambda will create an execution role with CloudWatch Logs permissions by default. A link to 'Change default execution role' is available.

Choose one of the following options to create your function.

- Author from scratch
Start with a simple Hello World example.
- Use a blueprint
Build a Lambda application from sample code and configuration presets for common use cases.
- Container image
Select a container image to deploy for your function.
- Browse serverless app repository
Deploy a sample Lambda application from the AWS Serverless Application Repository.

Basic information

Function name
Enter a name that describes the purpose of your function.

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime [Info](#)
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
 ▼ C

Architecture [Info](#)
Choose the instruction set architecture you want for your function code.
 x86_64
 arm64

Permissions [Info](#)
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▶ [Change default execution role](#)

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

The screenshot shows the 'Function overview' section of the AWS Lambda console. At the top, a green banner indicates 'Successfully created the function lamdaexp11. You can now change its code and configuration. To invoke your function with a test event, choose "Test".' Below the banner, the function name 'lamdaexp11' is displayed. On the left, there are tabs for 'Diagram' (selected) and 'Template'. The 'Diagram' view shows a single function icon labeled 'lamdaexp11' and a 'Layers' section with '(0)'. On the right, there are buttons for 'Throttle', 'Copy ARN', and 'Actions'. Below these are buttons for 'Export to Application Composer' and 'Download'. A 'Description' field is present with placeholder text 'A brief description of your function.' and a note that it was last modified 16 seconds ago. The 'Function ARN' is listed as arn:aws:lambda:eu-north-1:026090558619:function:lamdaexp11.

The screenshot shows the 'Code source' editor for the 'lambda_function' file. The interface includes a toolbar with 'File', 'Edit', 'Find', 'View', 'Go', 'Tools', 'Window', 'Test' (selected), and 'Deploy'. The code editor displays the following Python code:

```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     return {
6         'statusCode': 200,
7         'body': json.dumps('Hello from Lambda!')
8     }
```

On the left, a sidebar shows the project structure with a folder 'lamdaexp11' containing 'lambda_function.py'. On the right, there is a 'Upload from' button and a settings gear icon.

The screenshot shows the 'Configuration' tab of the AWS Lambda console. The left sidebar lists 'General configuration' and other sections: Triggers, Permissions, Destinations, Function URL, Environment variables, Tags, and VPC. The main area displays the 'General configuration' settings:

General configuration		
Description	Memory	Ephemeral storage
-	128 MB	512 MB
Timeout	SnapStart	
0 min 3 sec	Info	
	None	

An 'Edit' button is located in the top right corner of the configuration table.

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

Lambda > Functions > lamdaexp11 > Edit basic settings

Edit basic settings

Basic settings [Info](#)

Description - optional
D15C

Memory [Info](#)
Your function is allocated CPU proportional to the memory configured.
128 MB
Set memory to between 128 MB and 10240 MB

Ephemeral storage [Info](#)
You can configure up to 10 GB of ephemeral storage (/tmp) for your function. [View pricing](#)
512 MB
Set ephemeral storage (/tmp) to between 512 MB and 10240 MB.

SnapStart [Info](#)
Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized. To evaluate whether your function code is resilient to snapshot operations, review the [SnapStart compatibility considerations](#).

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

SnapStart [Info](#)
Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized. To evaluate whether your function code is resilient to snapshot operations, review the [SnapStart compatibility considerations](#).
None
Supported runtimes: Java 11, Java 17, Java 21.

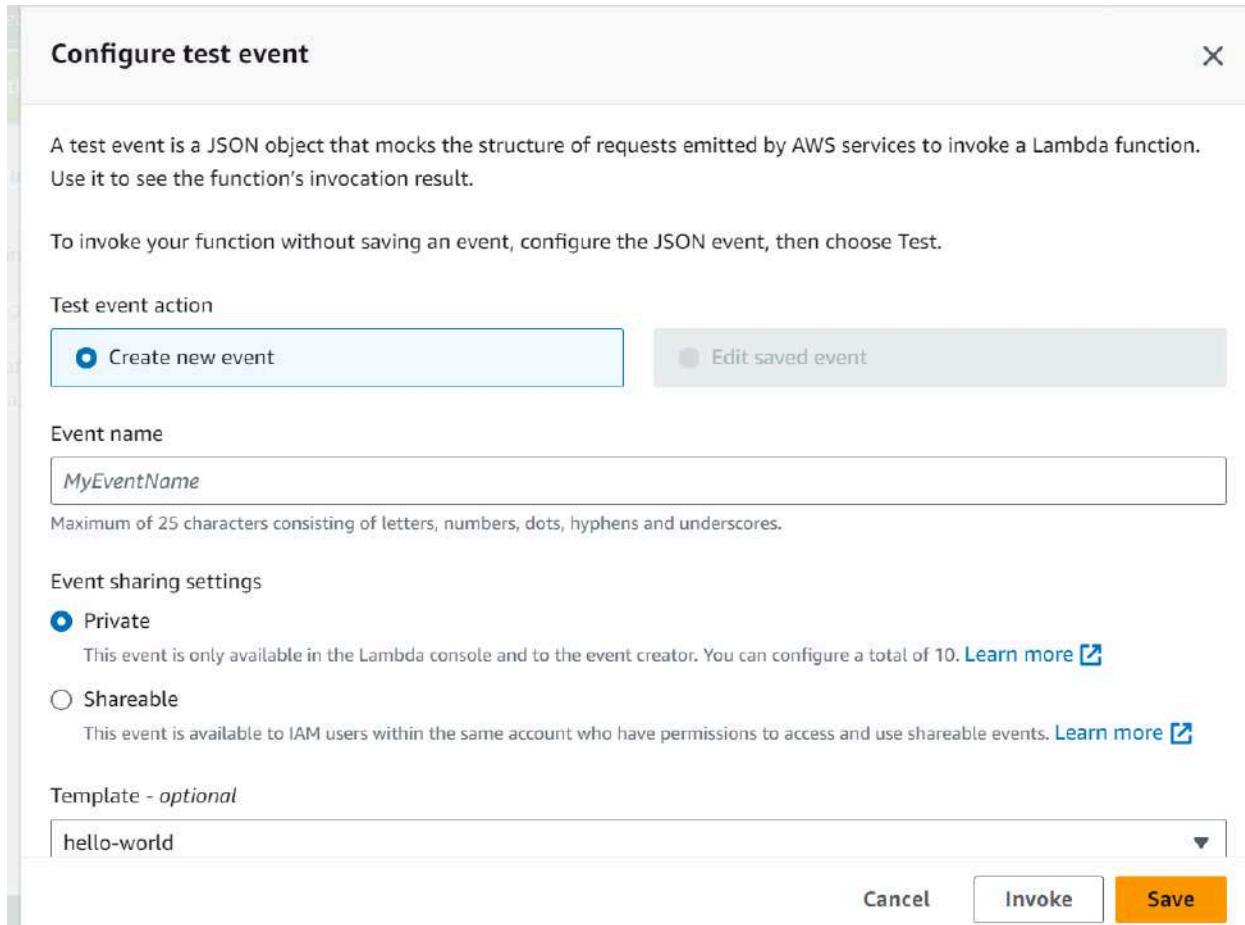
Timeout
0 min 1 sec

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).
 Use an existing role
 Create a new role from AWS policy templates

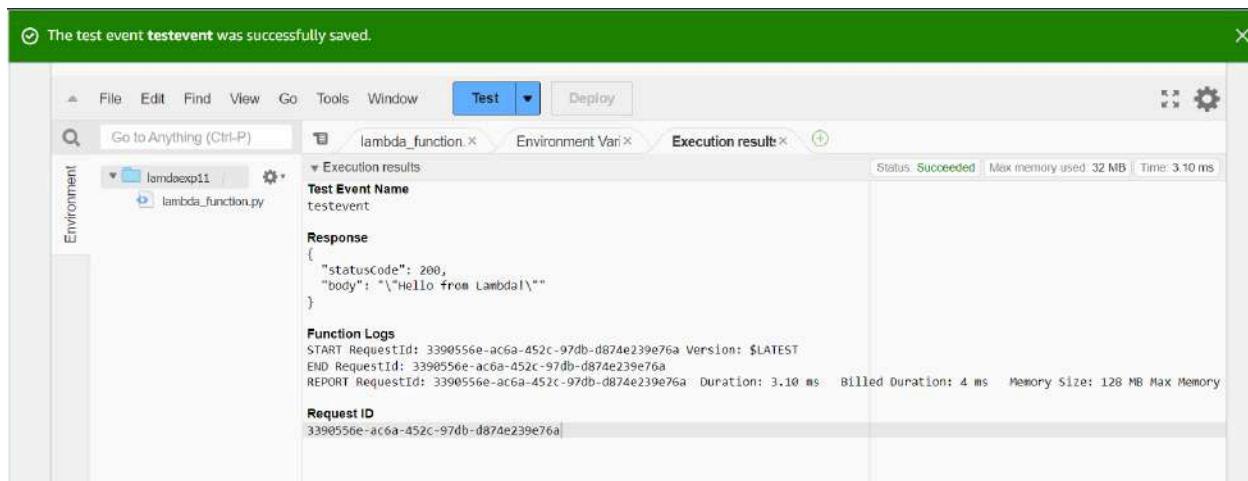
Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.
service-role/lamdaexp11-role-vj5j9g95 [View the lamdaexp11-role-vj5j9g95 role](#) on the IAM console.

Cancel **Save**

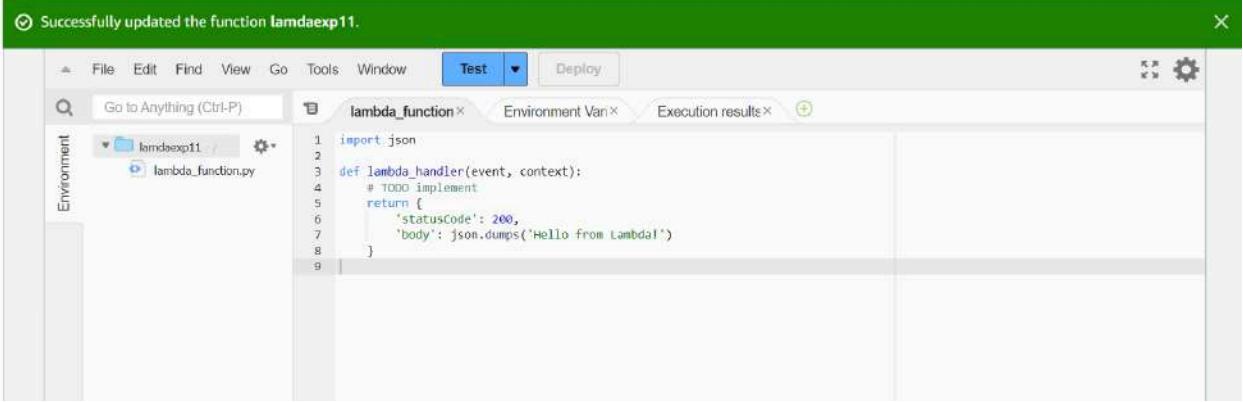
STEP 5: Go on configure test event click on “create new event” edit the event sharing accordingly and select hello world template for template option and then save it.



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



STEP 7: The function is successfully added .



The screenshot shows the AWS Lambda function editor interface. At the top, a green banner displays the message "Successfully updated the function lambdaexp11.". Below the banner, the menu bar includes File, Edit, Find, View, Go, Tools, Window, Test (which is currently selected), and Deploy. On the left, there's a sidebar labeled "Environment" and a file tree showing a folder named "lambdaexp11" containing a file "lambda_function.py". The main workspace is titled "lambda_function" and contains the following Python code:

```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     return [
6         'statusCode': 200,
7         'body': json.dumps('Hello from Lambda!')
8     ]
```

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

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

STEPS:

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

Amazon S3 > Buckets > Create bucket

Create bucket [Info](#)

Buckets are containers for data stored in S3.

General configuration

AWS Region
Europe (Stockholm) eu-north-1

Bucket type [Info](#)

General purpose
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

Directory
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name [Info](#)
exp12d15c

Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

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

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

Block all public access
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

Block public access to buckets and objects granted through new access control lists (ACLs)
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

Block public access to buckets and objects granted through any access control lists (ACLs)
S3 will ignore all ACLs that grant public access to buckets and objects.

Block public access to buckets and objects granted through new public bucket or access point policies
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

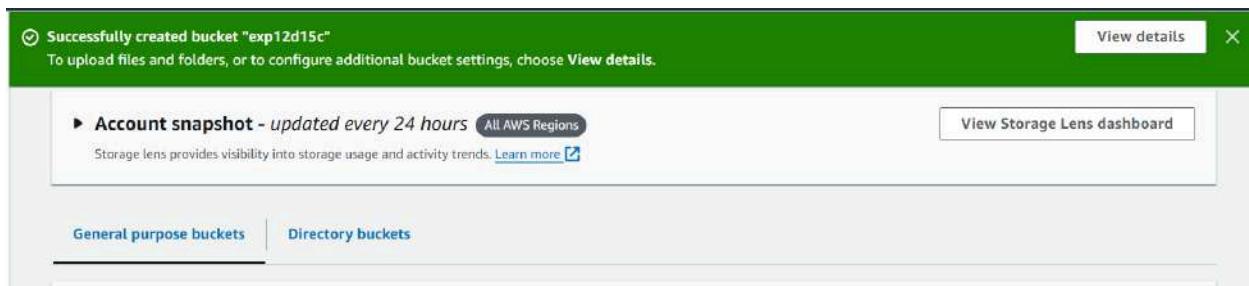
Block public and cross-account access to buckets and objects through any public bucket or access point policies
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

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

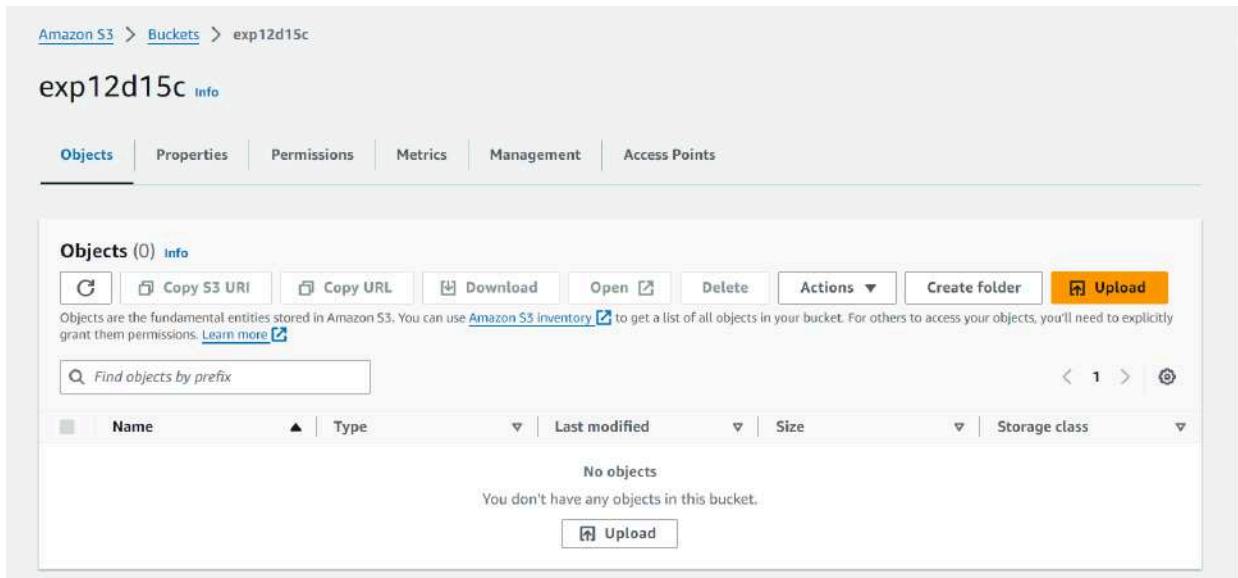
⚠️ Turning off block all public access might result in this bucket and the objects within becoming public
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

I acknowledge that the current settings might result in this bucket and the objects within becoming public.

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



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



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

Upload Info

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#) 

Drag and drop files and folders you want to upload here, or choose **Add files** or **Add folder**.

Files and folders (1 Total, 293.3 KB)

All files and folders in this table will be uploaded.

<input type="checkbox"/>	Name	Folder	Type
<input type="checkbox"/>	AppBar(title Text('Guidelines'),),....	-	image/png

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

 **Upload succeeded**
View details below.

Summary

Destination	Succeeded	Failed
s3://exp12d15c	 1 file, 293.3 KB (100.00%)	 0 files, 0 B (0%)

Files and folders (1 Total, 293.3 KB)

Name	Folder	Type	Size	Status	Error
AppBar(title...	-	image/png	293.3 KB	 Succeeded	-

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

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

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

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

The screenshot shows the AWS Lambda Functions console. The top navigation bar shows 'Lambda > Functions > lamdaexp11'. The main title is 'lamdaexp11'. On the right, there are buttons for 'Throttle', 'Copy ARN', and 'Actions'. A green success message box says: 'The trigger exp12d15c was successfully added to function lamdaexp11. The function is now receiving events from the trigger.' Below this, the 'Function overview' section is expanded, showing a 'Diagram' tab selected. The diagram shows a box labeled 'lamdaexp11' with a downward arrow pointing to a box labeled 'S3'. There are buttons '+ Add destination' and '+ Add trigger'. To the right, there is detailed information: Description 'D15C', Last modified '18 minutes ago', Function ARN 'arn:aws:lambda:eu-north-1:026090558619:function:lamdaexp11', and Function URL with an 'Info' link.

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

The screenshot shows the 'Test' tab for the 'lamdaexp11' function. The top navigation bar shows 'Code source' and 'Info'. The main area has tabs 'Test' (selected), 'Deploy', and 'Environment'. The 'Test Event Name' dropdown is set to 'testevent'. The 'Execution results' section shows a single entry: 'Status: Succeeded' with 'Max memory used: 32 MB' and 'Time: 1.97 ms'. The 'Response' field contains the JSON object: { "statusCode": 200, "body": "\"Hello from Lambda!\""}'. The 'Function Logs' section shows the log output: START RequestId: 6e57026c-6bfe-41fe-898d-2cdbe72fb1a3 Version: \$LATEST END RequestId: 6e57026c-6bfe-41fe-898d-2cdbe72fb1a3 REPORT RequestId: 6e57026c-6bfe-41fe-898d-2cdbe72fb1a3 Duration: 1.97 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Request ID 6e57026c-6bfe-41fe-898d-2cdbe72fb1a3].

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

Adv. DevOps

Assignment 1

Q1 Use S3 bucket and host Video Streaming
sol:-

Step 1: Set up S3 bucket

- Search for S3 bucket on your AWS account. Click on create bucket.
- Maintain all the options as default; give your bucket a name.
- Bucket is created. Now, add a video to that bucket. For that, click on the ~~name~~ of the bucket, this will redirect you to the Objects screen which shows the object of your bucket.
- Click on upload. Here now select an .mp4 extension file for video; once selected click on upload.

Step 2: Set up CloudFront

- Search for CloudFront on the Services tab. On the left dashboard you will find origin access = under the security > tab.
- Under origin access, click on create origin access identity; give identity a name & click create.

- (c) Now go to Distribution from the left pane
and click on create a cloudfront distribution
- (d) In the origin field select the s3 bucket
we created
- (e) Under origin access, select legacy access
identities
- (f) Under origin access identities, select the identity
that we created
- (g) Update the bucket policy ; In default cache
behavior > review > Redirect HTTP or HTTPS policy
hosting secure
- (h) keep the remaining options as default & click
on create distribution

Step 3: Accessing the hosted video

- (a) Once the distribution is deployed, copy the domain
name
- (b) Go to s3 bucket, copy the key for the video
- (c) Combine them like <domain name>/<key of video>
- (d) The video is now streaming.

Q2

Discuss BMW & Hotelsor case study using AWS

Sol:-

BMW and Hotelsor are prime examples of how organisations can leverage the power of Amazon Web Services (AWS) to enhance their operations and deliver exceptional customer experiences. Let's explore how these companies have used AWS to achieve their goals.

BMW : Driving Innovation in the Automotive Industry

BMW faced the challenge of managing vast amounts of vehicle data, ensuring data security, and accelerating the development of new connected car features. By migrating its on-premises data lake to AWS, BMW created a centralized platform for processing & analyzing data from millions of vehicles. This enabled them to:

- Gain deeper insights into driving patterns, vehicle performance & customer preferences
- Develop innovative connected car features like over-the-air updates, voice-activated assistants and predictive maintenance

- strengthens data security with robust measures on the AWS platform

Holstar: Revolutionizing Digital Entertainment

Holstar, a leading digital entertainment platform in India, sought to deliver high-quality streaming content to millions of users with varying internet speeds and devices. By adopting AWS, Holstar built a scalable and reliable streaming platform capable of:

- Delivering seamless streaming experiences to users worldwide, regardless of their internet connections or devices
- Scaling to meet peak demand during major sporting events & other popular content releases
- Reducing latency and improving user experience through AWS's global network.

Key Takeaways:

- ① AWS empowers innovation: Both BMW & Holstar have used AWS to drive innovation & enhance their offerings

- ② Scalability is essential: AWS's elastic infrastructure has enabled these companies to scale their operations to meet increasing demands
- ③ Data-driven insights: By leveraging AWS, BMW has been able to gain valuable insights from its vehicle data
- ④ Exceptional user experiences: Hotstar has used AWS to deliver a superior streaming experience to its customers.

In conclusion, BMW & Hotstar demonstrate how organisations can effectively leverage AWS to achieve their business objectives & provide exceptional value to their customers. By embracing cloud technology, these companies have been able to stay ahead of the curve & drive innovation in their respective industries.

Q3

Why Kubernetes and advantages and disadvantages of Kubernetes. How adidas uses Kubernetes
Sol:-

Kubernetes, an open-source platform for managing containerized applications, has revolutionized the way businesses deploy, scale and operate their software. For Adidas, a global sportswear giant, Kubernetes has been a game-changer enabling them to enhance their e-commerce operations and deliver a superior customer experience.

Advantages of Kubernetes

- Scalability: Kubernetes can effortlessly scale applications up or down to meet fluctuating demands. For Adidas, this means their e-commerce platform can handle peak traffic during sales seasons like Black Friday without compromising performance.
- Portability: Kubernetes applications can be deployed across various cloud platforms and on-premises infrastructure. This flexibility allows Adidas to choose the most suitable environment based on their specific needs & cost considerations.

Q3

- Efficiency: Kubernetes optimizes resource utilization by dynamically allocating resources based on application demands. This helps Adidas reduce costs & improve overall performance.

Disadvantages of Kubernetes

- Complexity: Kubernetes can be complex to learn & manage, especially for organizations new to containerization.
- Operational overhead: Managing Kubernetes requires specialized knowledge and can introduce additional operational costs.
- Vendor lock-in: Some Kubernetes distributions may be tied to specific cloud providers or vendors, potentially limiting flexibility & increasing dependency.

How Adidas leverages Kubernetes

- Scalability: Adidas can dynamically scale their e-commerce platform to handle peak traffic during sale seasons.
- Reliability: Kubernetes ensures high availability by minimizing downtime, guaranteeing a consistent

customer experiences

- portability: Adhere can deploy their e-commerce platform and on-premises infrastructure.
- Efficiency: Kubernetes optimizes resource utilization, reducing costs & improving overall performance.

Q4 What are Nagios and explain how Nagios are used in E-services

Ans:-

Nagios is a popular open-source network monitoring system that has become an essential tool for organizations that rely on e-services. It provides real-time information on the health and performance of computer systems, applications & network infrastructure, allowing administrators to proactively identify & resolve potential issues.

Key Features of Nagios

- ① Comprehensive Monitoring: Nagios can monitor a wide range of systems and services, including servers, networks, applications and databases.

Q 1

- (1) Customizable Alerts: Users can configure Nagios to send alerts via email, SMS or other notification methods.
- (2) Detailed Reporting: Nagios generates detailed reports on system performance, allowing administrators to analyze trends and identify areas for improvement.
- (3) Extensibility: Nagios supports a vast ecosystem of plugins, extending its capabilities to monitor specific systems & applications.

Uses in E-services

In world of e-services, Nagios plays a crucial role in ensuring the reliability and performance of online applications. Here are some common uses cases:

- Website Monitoring: Nagios can monitor the availability & response time of websites, ensuring that they are accessible to users.
- Application performance monitoring: Nagios can track the performance of key application metrics such as CPU usage, memory consumption, database query times, etc.

Assignment 2

Create a REST API with the serverless framework

Step 1 : Install Prerequisites

- Install Node.js from the official website
- Install the serverless framework globally using npm
`npm install -g serverless`

Step 2 : Create a Serverless project

- Create a new directory for your project & navigate into it
- Create a new serverless service using the AWS Node.js template

Step 3 : Set up AWS credentials

- Configure AWS CLI if not already set up
- Set up your AWS Access Key, Secret Key, Region & output Format

Step 4: Modify serverless.yml configuration

- (a) Open serverless.yml & define your REST endpoints, Lambda functions, and event triggers.
- (b) Add HTTP event triggers for the CRUD operations (POST, GET, PUT, DELETE)

Step 5: Write Lambda Function Handlers

- In handler.js, write the function to handle the API requests (e.g. create, retrieve, update, delete)

Step 6: Deploy your service

- Deploy the REST API to AWS.
- After deployment, note down the API gateway endpoints URLs from the console.

Step 7: Test the API

Use tools like curl, Postman or browser to test the deployed API using the provided URLs.

- Create your own profile in sonarqube
- Use sonarcloud to analyse your github code
- REST API to install sonarlint in your Java IntelliJ or Eclipse IDE to analyse your Java code
- CRUD operation: Analyse python project with sonarqube
- Analyse node.js project with sonarqube
- Analyse node.js project with sonarqube

• 6

In modern software development, maintaining code quality is paramount for ensuring the long-term sustainability & reliability of a project. SonarQube & SonarCloud offer powerful tools for static code analysis, helping developers identify and fix issues in their code.

gate

① Setting up Sonarqua for local testing

- Installation : ~~Download & install sample from~~
~~sonarqube.org~~
set up a database for sonarqube
start the sonarqube server by running the
sonar.sh script in the sonarqube installation
directory
Open <http://localhost:9000> in your browser &
create a new profile
login and create a new project.

② Use sonarcloud to Analyse your github code

- Go to sonarcloud
- Sign up with your github account & get sonarcloud access to your repositories
- Create a new project & select the github repository to analyse
- Configure the project key and scan configuration
- Follow the provided instruction to set up sonar-project.properties file in the root of repository
- To use Github actions, Jenkins or manual trigger the analysis using Sonarcloud analyse the code quality in the cloud

③ Install Sonarlint in IntelliJ IDEA or Eclipse for Java code Analysis

- Open IntelliJ IDEA and navigate to File → Settings → Plugins
- Search for "Sonarlint" and install it
- Restart IntelliJ after the installation is complete
- Open your java project, right-click and select Analyse → Sonarlint. This will analyse the code and highlight any issues

① We can cloud to Analyse your git hub code

- Go to sonarcloud
- Sign up with your git hub account & get sonarcloud access to your repository
- Create a new project & select the git hub repository to analyse
- Configure the project key and scan key
- Follow the provided instruction to set up cover-project.properties file in the root of repository
- Use Github actions, Jenkins or manual trigger the analysis using sonarcloud analyse the code quality in the cloud

② Install sonarlint in IntelliJ Idea or for Java code Analysis

- Open IntelliJ IDEA and navigate to File → Settings → Plugins
- Search for "sonarlint" and install it
- Restart IntelliJ after the installation is complete
- Open your java project, right-click and select Analyse → sonarlint. This will analyse the code and highlight any issues

Analyse Python Project with SonarQube

Install the SonarScanner scanner on your machine
In your Python project directory, create
a sonar-project.properties file with the
following content

sonar.projectKey = your-project-key

sonar.sources = .

sonar.language = py

sonar.python.version = 3.x

sonar.sourceEncoding = UTF-8

Run the analysis by executing the following
command in the terminal

- sonar-scanner

After analysis, view the results on your
SonarQube dashboard

Analyse Node.js project with SonarQube

In your Node.js project directory, create a
sonar-project.properties file with the same
content as mentioned in the previous case

Install SonarScanner globally if not already

Run the analysis using sonar-scanner

Check the analysis results on the SonarQube
dashboard to identify and fix potential
issues

(Q3)

Creating a self-service infrastructure model in Terraform for a large organization involves the following steps:

- Step 1:- Define Infrastructure standards:- Establish clear standards and best practices for infrastructure deployment, including naming conventions, resource types, tagging, policies, and security compliance. This foundation ensures consistency across the organization.
- Step 2:- Create Terraform module:- Develop reusable Terraform modules based on your organization's standards. Each module defines resources and configuration, allowing teams to deploy infrastructure efficiently & consistently.
- Step 3:- Set up Terraform cloud or enterprise:- Use Terraform Cloud for centralized management of configurations & state files, enabling collaboration & access control for infrastructure change.
- Step 4:- Configure version control:- Integrate Terraform modules with version control (e.g. GitHub). This tracks changes, facilitates collaboration & ensures proper versioning for updates & compatibility.

Step 5:- Integrate with service now to automate infrastructure requests. This trigger runs terraform and thus streamlining the process for teams

Step 6:- Provide Documentation & training :-

Create Documentation & trainig for using terraform modules & submitting requests helping teams understand & follow best practices

Step 7:- Monitor & support :-

Monitor the usage of the self-service model & provide ongoing support to users. Gathering feedbacks helps identify that the infrastructure remains compliant and super efficient.

By following above steps, organisations can enable product teams to manage their own infrastructure through a standardised terraform approach

SK