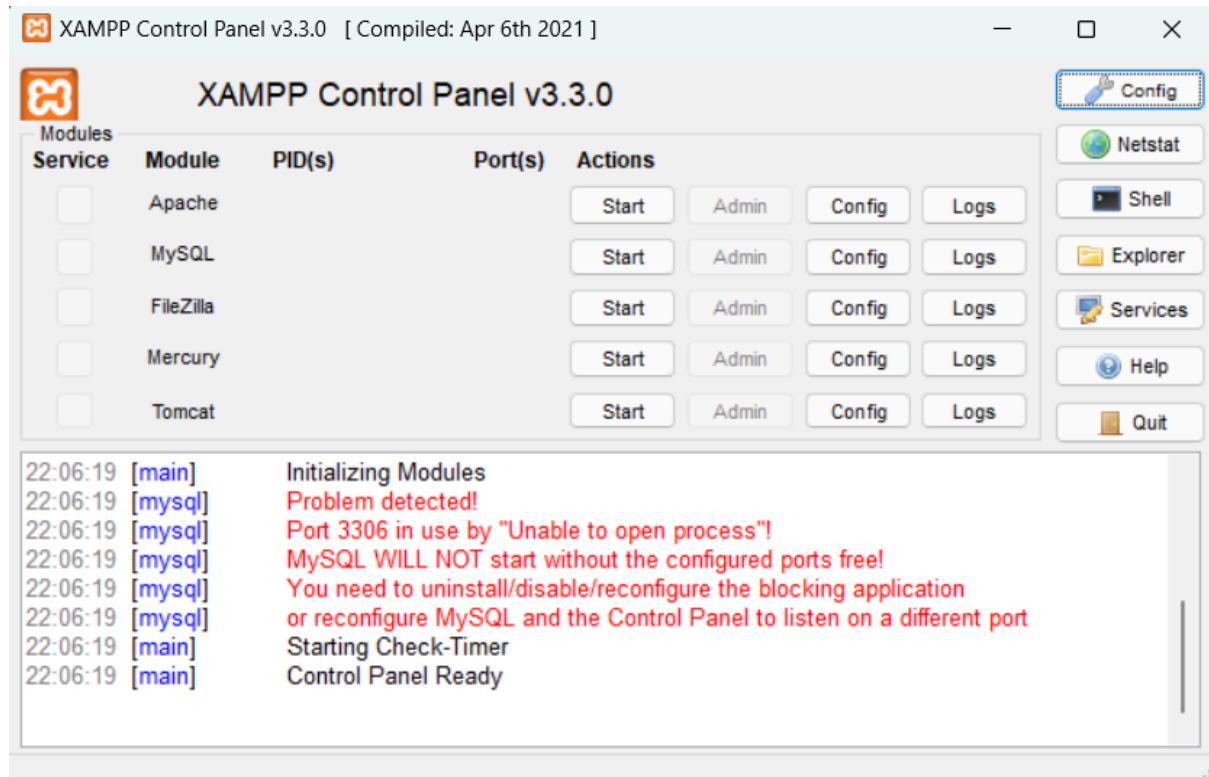


Part 1: To develop a website and host on Xampp

Step 1: Go to the official website of Xampp.

Select the suitable version and compatible installation.

Step 2: After installation open the Xampp control panel, we need to start Apache server.



Step 3: setup Php project. Create a php project. Save the file

Save it in the htdocs that is on the Xampp folder.

The screenshot shows a code editor with a dark theme. The menu bar includes 'File', 'Edit', and 'View'. The code in the editor is:

```
<html>
<head>
<title>First PHP Program</title>
</head>
<body>
<?php
echo "This is advDeveops lab";
?>
```

Step 4: Start the Xampp server and go to local host in browser.
It should be localhost/filename.php

← ⌛ ⓘ localhost/LAB/

Index of /LAB

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
_parent	-	-	Parent Directory
p1.php	2024-08-08 22:12	108	

Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.0.30 Server at localhost Port 80

Output:

← ⌛ ⓘ localhost/LAB/p1.php

Nikita Thadani

Jaipur
Rajasthan

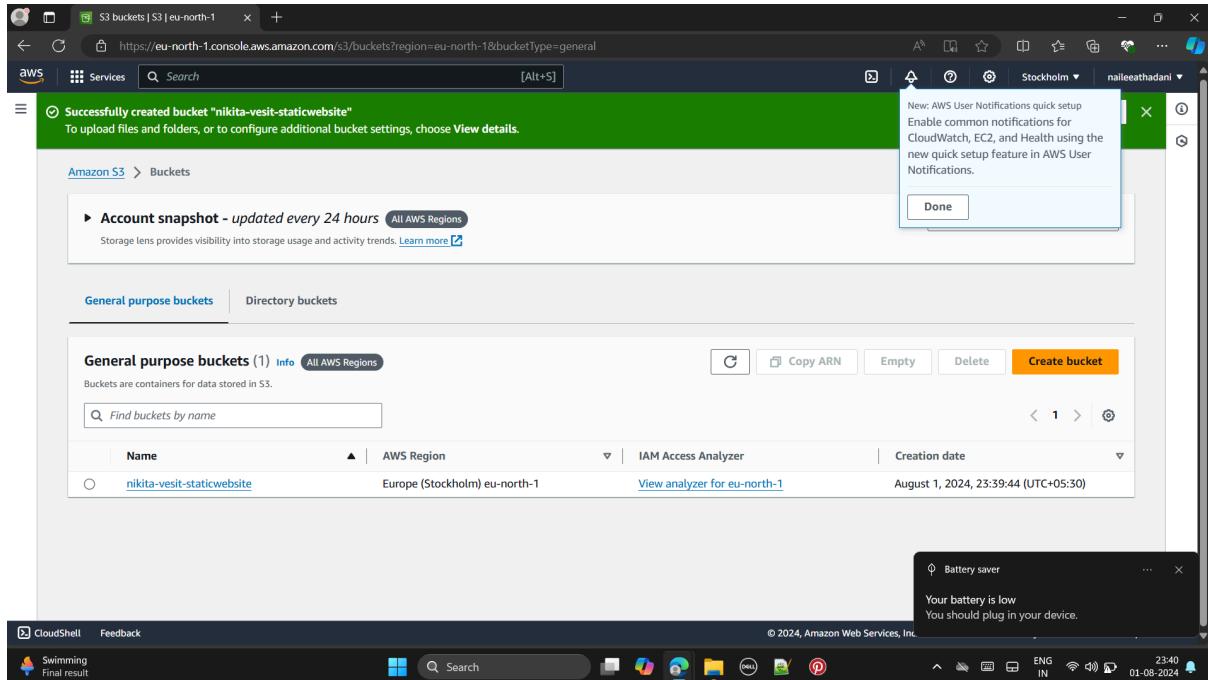
Contact

Part 2: Hosting A static website on Aws S3

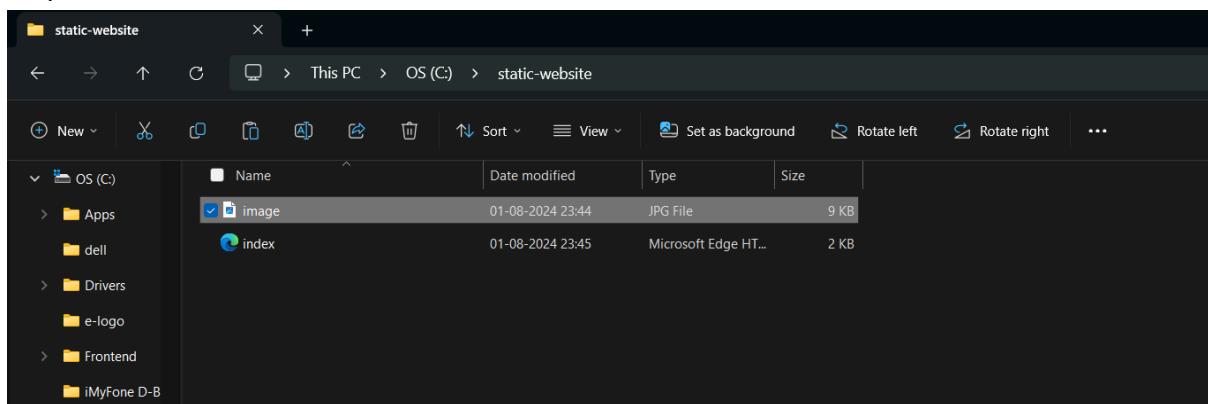
Step 1: Login onto AWS Academy and start learners lab

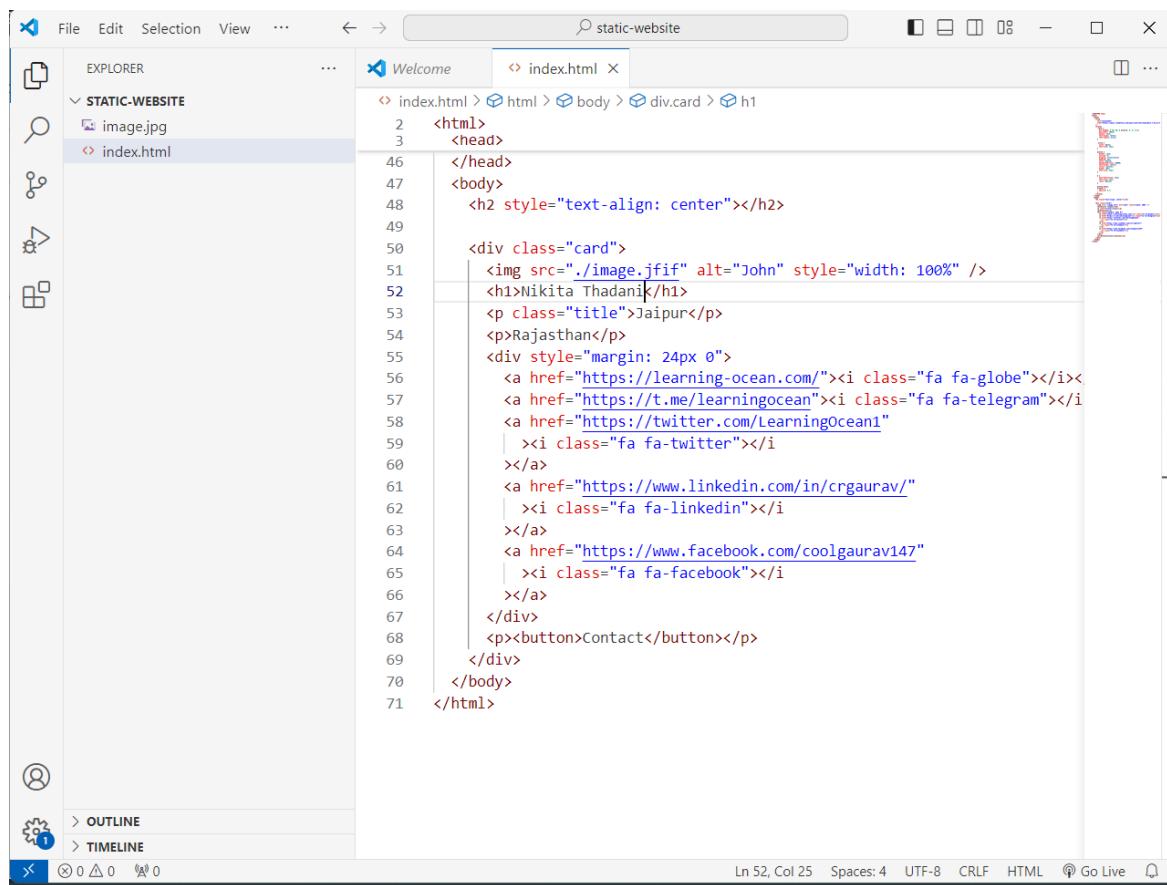
Step 2: Search for S3 Service

Step 3: Create a bucket



Step 4: Add contents inside bucket





```
<html>
  <head>
    <h2 style="text-align: center"></h2>
    <div class="card">
      
      <h1>Nikita Thadani</h1>
      <p class="title">Jaipur</p>
      <p>Rajasthan</p>
      <div style="margin: 24px 0">
        <a href="https://learning-ocean.com/"><i class="fa fa-globe"></i></a>
        <a href="https://t.me/learningocean"><i class="fa fa-telegram"></i></a>
        <a href="https://twitter.com/LearningOcean1" style="margin-left: 10px;"><i class="fa fa-twitter"></i></a>
        <a href="https://www.linkedin.com/in/crgaurav/" style="margin-left: 10px;"><i class="fa fa-linkedin"></i></a>
        <a href="https://www.facebook.com/coolgaurav147" style="margin-left: 10px;"><i class="fa fa-facebook"></i></a>
      </div>
      <p><button>Contact</button></p>
    </div>
  </body>
</html>
```

Step 5: Fill in the details and name your bucket. Use default settings. Uncheck 'Block all public access'.

Step 6: Once completed, click 'Create Bucket'

Step 7: Open the bucket and click upload in the objects tab. After adding files click on upload files.

Step 8: go to the properties tab and navigate to 'Static website hosting'

The screenshot shows the AWS S3 console with the bucket 'nikita-vesit-staticwebsite' selected. A green success message at the top states 'Successfully edited static website hosting.' Below it, under the 'Requester pays' section, the setting is shown as 'Requester pays' with 'Disabled' underneath. In the 'Static website hosting' section, 'Static website hosting' is set to 'Enabled'. Under 'Hosting type', 'Bucket hosting' is selected. The 'Bucket website endpoint' section shows the URL <http://nikita-vesit-staticwebsite.s3-website.eu-north-1.amazonaws.com>.

Step 9: Head to permissions tab and navigate 'Bucket policy'

The screenshot shows the AWS Lambda function editor with a policy document. The code is as follows:

```
1 ▾ {
2   "Id": "Policy1722572303678",
3   "Version": "2012-10-17",
4   "Statement": [
5     {
6       "Sid": "Stmt1722572291946",
7       "Action": [
8         "s3:GetObject"
9       ],
10      "Effect": "Allow",
11      "Resource": "arn:aws:s3:::nikita-vesit-staticwebsite/*",
12      "Principal": "*"
13    }
14  ]
15 }
```

Step 10: Once updated, a link will be available

Static website hosting[Edit](#)Use this bucket to host a website or redirect requests. [Learn more](#)

Static website hosting

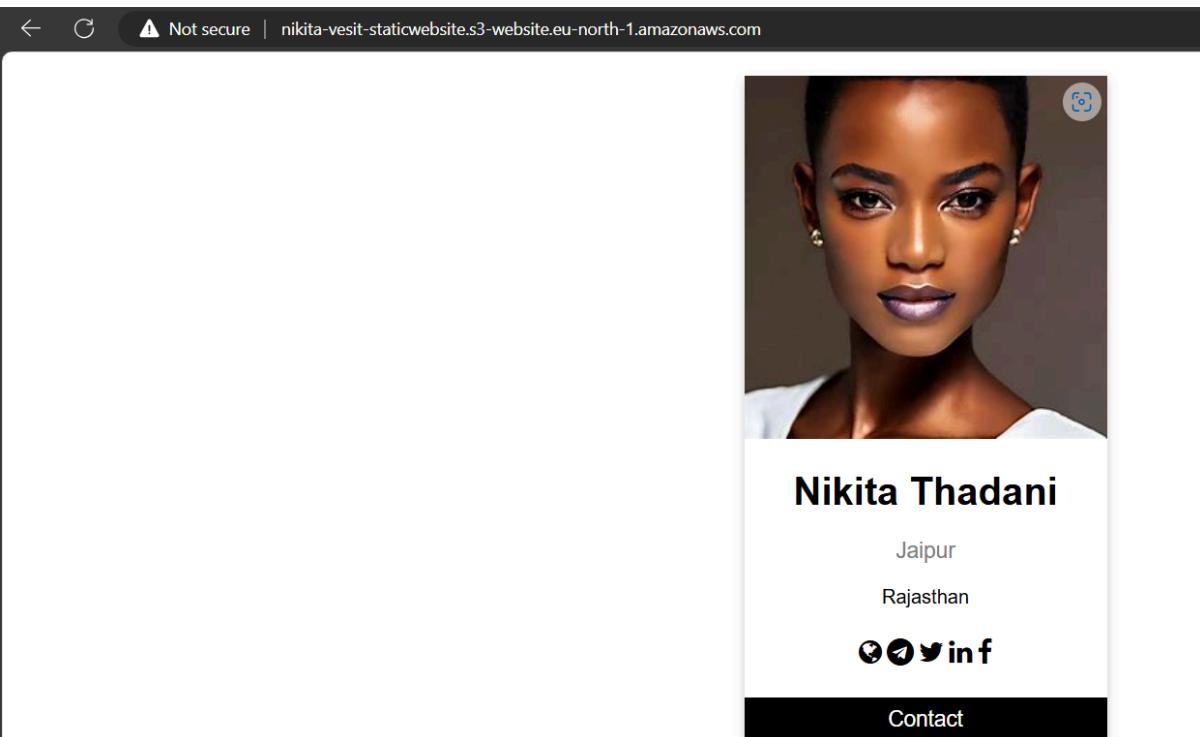
Enabled

Hosting type

Bucket hosting

Website endpoint copied

nt

bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket. [Learn more](#)<http://nikita-visit-staticwebsite.s3-website.eu-north-1.amazonaws.com>**Output:**

Experiment 1B

To understand the benefits of cloud infrastructure and Setup cloud9 IDE, and perform collaboration demonstration.

Step 1: Login into your AWS account.

Step 2: Navigate to Cloud 9 service

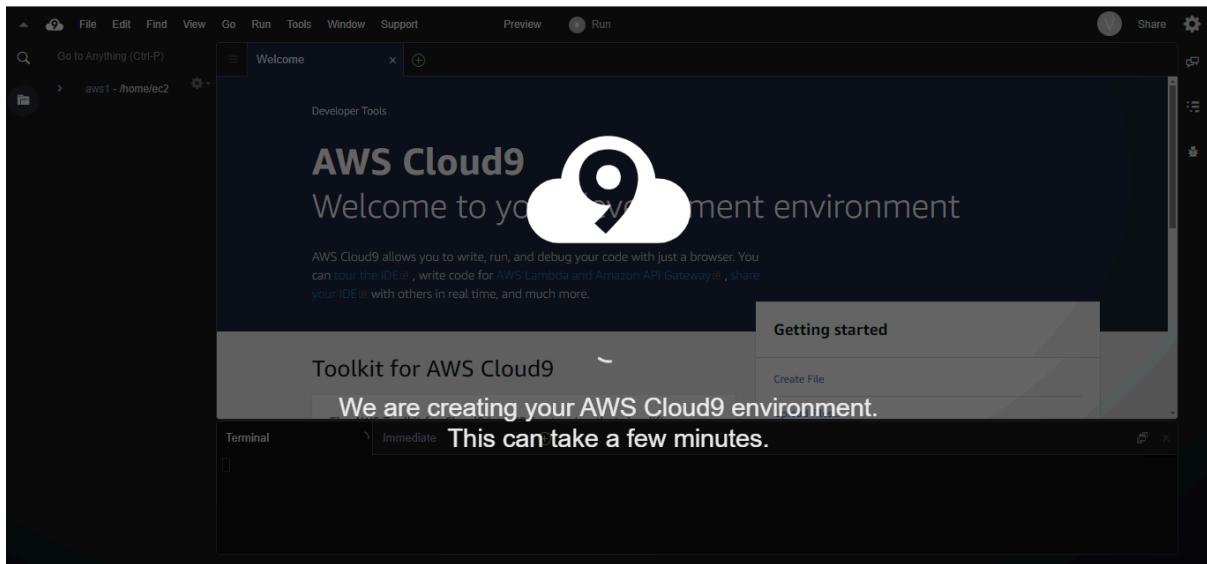
Step 3: Create a Instance

The screenshot shows the AWS EC2 Instances Launch an instance page. A green success banner at the top states "Successfully initiated launch of instance (i-06a85947b2d9e4072)". Below it, a "Launch log" button is visible. The "Next Steps" section contains four cards: "Create billing and free tier usage alerts", "Connect to your instance", "Connect an RDS database", and "Create EBS snapshot policy". Each card has a corresponding "Create" or "Learn more" button. At the bottom, there's a terminal window showing the launch of an Ubuntu instance with IP 172.31.93.2 and port 80 open. The terminal also displays system updates and license information.

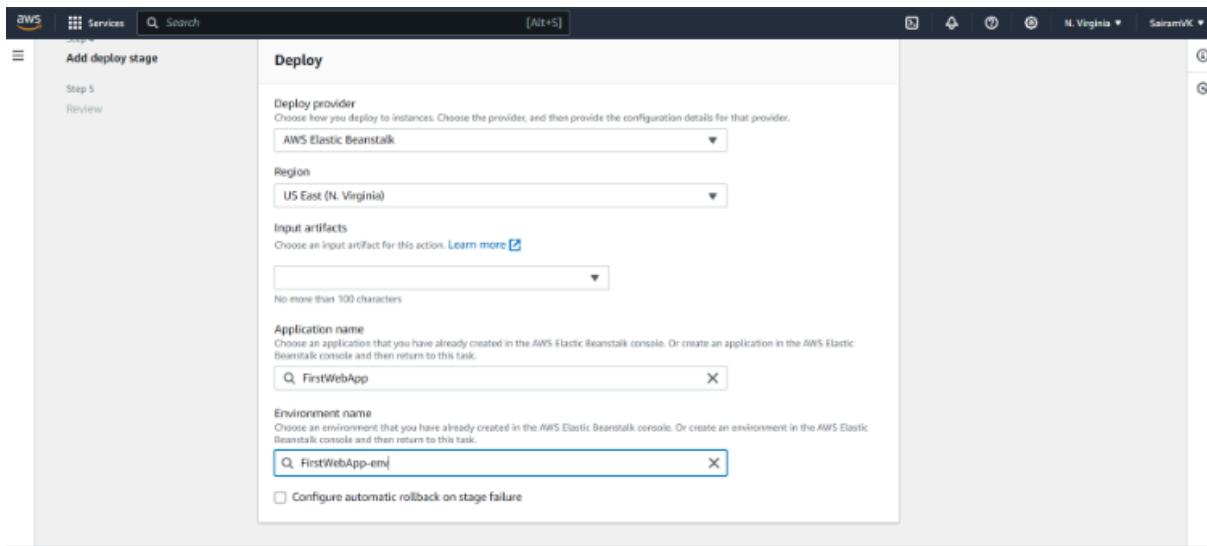
Step 4: Create a bucket

The screenshot shows the AWS S3 Buckets page. A green success banner at the top states "Successfully created bucket "nikita-visit"". Below it, a "View details" button is visible. The main area shows an account snapshot and a "General purpose buckets" table. The table lists one bucket: "nikita-visit" (Info, All AWS Regions). The bucket is described as a container for data stored in S3. It has options to "Copy ARN", "Empty", "Delete", and "Create bucket". The table includes columns for Name, AWS Region, IAM Access Analyzer, and Creation date. The creation date is August 1, 2024, 21:36:55 (UTC-07:00).

Step 5:Create an Environment



Step 6:Deploy



Step 7:Choose a Pipeline Settings

Step 1: Choose pipeline settings

Pipeline settings

- Pipeline name: Firstpipeline
- Pipeline type: V2
- Execution mode: QUEUED
- Artifact location: codepipeline-us-east-1-191647336071
- Service role name: AWSCodePipelineServiceRole-us-east-1-Firstpipeline

Variables

Name	Default value	Description
No variables		

Step 8: Add source stage

Source action provider

Source action provider: GitHub (Version 2)

OutputArtifactFormat: CODE_ZIP

DetectChanges: false

ConnectionArn: arn:aws:codeconnections:us-east-1:011528263337:connection/c61655f4-361c-4c99-8981-9d9869dde30f

FullRepositoryId: SURG30N6997/aws-codepipeline-s3-codedeploy-linux

Default branch: master

Trigger configuration

You can add additional pipeline triggers after the pipeline is created.

Trigger type:

Step 9: Add Deploy stage

No build

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

Create pipeline

Aim:To build your application using AWS code, build and deploy on S3 or SEBS using AWS code pipeline, diploid sample application on EC to instance, using AWS codedeploy.

Step 1:Open the AWS console and then search elastic beanstalk

The screenshot shows the AWS Elastic Beanstalk landing page. At the top, there's a navigation bar with 'Services' and a search bar. Below the header, the title 'Amazon Elastic Beanstalk' is displayed with the subtitle 'End-to-end web application management.' A 'Get started' button is prominent on the right. The main content area includes sections for 'Get started', 'Benefits and features' (with sub-sections for 'Easy to get started' and 'Complete resource control'), and 'Pricing'. On the far right, there's a sidebar with 'Getting started' and 'Launch a web application' options.

Step 2:Click on create application and configure the environment.

This screenshot shows the 'Application information' step of the 'Create Application' wizard. The left sidebar lists steps 1 through 6. Step 1 is 'Set up networking, database, and tags'. Step 2 is 'Worker environment', which is currently selected. Step 3 is 'Application information', Step 4 is 'Environment information', Step 5 is 'Configure updates, monitoring, and logging', and Step 6 is 'Review'. In the main panel, the 'Application name' field is filled with 'NikitaWebApp'. Below it, there's a note about the maximum length of 100 characters. The 'Environment name' field is also filled with 'NikitaWebApp-env', with a note about character restrictions and uniqueness.

Step 1: Configure environment

Environment information

Environment tier	Application name
Web server environment	nikita-webapp
Environment name	Application code
Nikita-webapp-env	Sample application
Platform	arm:aws:elasticbeanstalk:eu-north-1::platform/PHP 8.3 running on 64bit Amazon Linux 2023/4.3.1

Step 2: Configure service access

Service access Info
Configure the service role and EC2 instance profile that Elastic Beanstalk uses to manage your environment. Choose an EC2 key pair to securely log in to your EC2 instances.

Service role	EC2 key pair	EC2 instance profile
--------------	--------------	----------------------

Elastic Beanstalk

Nikita-webapp-env Info

Environment overview

Health	Environment ID
⊖ Unknown	e-ciuruu2dd4
Domain	Application name
-	nikita-webapp

Platform

Platform	PHP 8.3 running on 64bit Amazon Linux 2023/4.3.1
Running version	-
Platform state	Supported

Events Health Logs Monitoring Alarms Managed updates Tags

Events (2) Info

Step 3: Choose PHP from top down menu

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)

Step 4: Create a key

Step 5: Go to easy to instance, slap from the lead panel and create a keypad

EC2 > Key pairs > Create key pair

Create key pair [Info](#)

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 to an instance.

Name

new-key

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type [Info](#)

RSA ED25519

Private key file format

.pem
For use with OpenSSH

In the same fashion go to IAM and then under role section, click create role and then select AWS service and under instances select EC2

Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case

EC2

Step 6:Now come back to Elastic beanstalk page



EXPERIMENT 3

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

1.Create 3 instances and name them.

Master

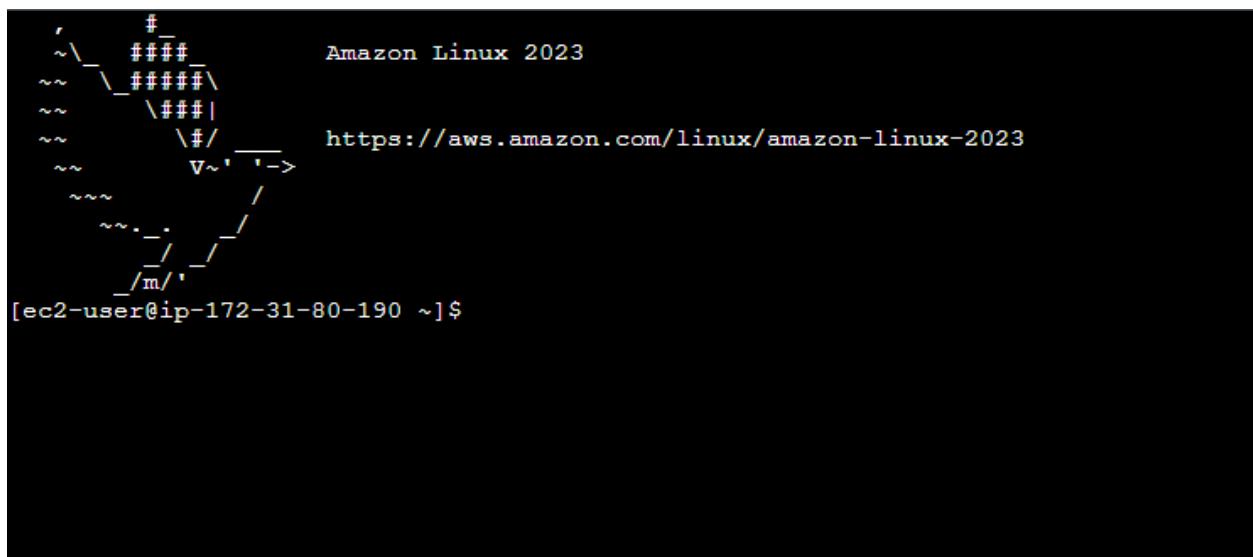
Worker-1

worker-2

Select a Key pair. Allow SSH

Instances (3) Info		Last updated less than a minute ago	C	Connect	Instance state ▾	Actions ▾	Launch instances	▼
<input type="text"/> Find Instance by attribute or tag (case-sensitive)					All states ▾	< 1 >		
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability	
<input type="checkbox"/>	worker-2	i-0aa9f89768e3a199c	Running	t2.micro	Initializing	View alarms	us-east-1b	
<input type="checkbox"/>	worker-1	i-05c2b3ba79c1d85ab	Running	t2.micro	Initializing	View alarms	us-east-1b	
<input type="checkbox"/>	master	i-0bc867eb06b7964b7	Running	t2.micro	Initializing	View alarms	us-east-1b	

2. Connect the instances and open the terminal to run commands to install docker and kubernetes.



3. Install Docker for all 3 instances. Repeat all the steps for all three instances.

```

Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
[root@ip-172-31-80-190 ec2-user]# sudo service docker status
Redirecting to /bin/systemctl status docker.service
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: disabled)
     Active: active (running) since Fri 2024-09-13 04:23:51 UTC; 25s ago
TriggeredBy: ● docker.socket
   Docs: https://docs.docker.com
 Main PID: 29009 (dockerd)
   Tasks: 7
  Memory: 29.6M
    CPU: 330ms
   CGroup: /system.slice/docker.service
           └─29009 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nofile=32768:65536

Sep 13 04:23:50 ip-172-31-80-190.ec2.internal systemd[1]: Starting docker.service - Docker Application Container Engine...
Sep 13 04:23:50 ip-172-31-80-190.ec2.internal dockerd[29009]: time="2024-09-13T04:23:50.998251422Z" level=info msg="Starting up"
Sep 13 04:23:51 ip-172-31-80-190.ec2.internal dockerd[29009]: time="2024-09-13T04:23:51.068745150Z" level=info msg="Loading containers: start."
Sep 13 04:23:51 ip-172-31-80-190.ec2.internal dockerd[29009]: time="2024-09-13T04:23:51.526546118Z" level=info msg="Loading containers: done."
Sep 13 04:23:51 ip-172-31-80-190.ec2.internal dockerd[29009]: time="2024-09-13T04:23:51.555765635Z" level=info msg="Docker daemon" commit=b08a51
Sep 13 04:23:51 ip-172-31-80-190.ec2.internal dockerd[29009]: time="2024-09-13T04:23:51.556129762Z" level=info msg="Daemon has completed initialization"
Sep 13 04:23:51 ip-172-31-80-190.ec2.internal dockerd[29009]: time="2024-09-13T04:23:51.598513281Z" level=info msg="API listen on /run/docker.sock"
Sep 13 04:23:51 ip-172-31-80-190.ec2.internal systemd[1]: Started docker.service - Docker Application Container Engine.
lines 1-20/20 (END)

```

4. Install kubernetes using intsal kubeadms and get code from there.

1. Set SELinux to `permissive` mode:

These instructions are for Kubernetes 1.31.

```

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

```

```

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

```

3. Install kubelet, kubeadm and kubectl:

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

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

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

```

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

Installed:
  conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64
  kubeadm-1.31.1-150500.1.1.x86_64
  kubelet-1.31.1-150500.1.1.x86_64
  libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64
  libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64
  libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64

Complete!
[root@ip-172-31-83-1 ec2-user]# sudo systemctl enable --now kubelet
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /usr/lib/systemd/system/kubelet.service.
[root@ip-172-31-83-1 ec2-user]#

```

5.check the repositories

```

[root@ip-172-31-83-1 ec2-user]# yum repolist
repo id                                repo name
amazonlinux                             Amazon Linux 2023 repository
kernel-livepatch                         Amazon Linux 2023 Kernel Livepatch repository
kubernetes                             Kubernetes

```

6.run command kubeadm

```

[root@ip-172-31-83-1 ec2-user]# kubeadm init
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
  [WARNING FileExisting-socat]: socat not found in system path
  [WARNING FileExisting-tc]: tc not found in system path
error execution phase preflight: [preflight] Some fatal errors occurred:
  [ERROR NumCPU]: the number of available CPUs 1 is less than the required 2
  [ERROR Mem]: the system RAM (949 MB) is less than the minimum 1700 MB
[preflight] If you know what you are doing, you can make a check non-fatal with '--ignore-preflight-errors=All'
To see the stack trace of this error execute with --v=5 or higher
[root@ip-172-31-83-1 ec2-user]#

```

7.check repo

```

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

```

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

```

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

```

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

```
[wait-control-plane] Waiting for the kubelet to boot up the control plane as static Pods from directory "/etc/kubernetes/manifests"
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 518.648244ms
[api-check] Waiting for a healthy API server. This can take up to 4m0s
[api-check] The API server is healthy after 10.001658622s
[upload-config] Storing the configuration used in ConfigMap "kubeadm-config" in the "kube-system" Namespace
[kubelet] Creating a ConfigMap "kubelet-config" in namespace kube-system with the configuration for the kubelets in the cluster
[upload-certs] Skipping phase. Please see --upload-certs
[mark-control-plane] Marking the node ip-172-31-14-85.ap-southeast-2.compute.internal as control-plane by adding the labels: [node-role.kubernetes.io/control-plane node.kubernetes.io/exclude-from-external-load-balancers]
[mark-control-plane] Marking the node ip-172-31-14-85.ap-southeast-2.compute.internal as control-plane by adding the taints [node-role.kubernetes.io/control-plane:NoSchedule]
[bootstrap-token] Using token: 6lysht.48enn4gmnhof6ex8
[bootstrap-token] Configuring bootstrap tokens, cluster-info ConfigMap, RBAC Roles
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to get nodes
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to post CSRs in order for nodes to get long term certificate credentials
[bootstrap-token] Configured RBAC rules to allow the csrapprover controller automatically approve CSRs from a Node Bootstrap Token
[bootstrap-token] Configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
[bootstrap-token] Creating the "cluster-info" ConfigMap in the "kube-public" namespace
[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable kubelet client certificate and key
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy
Your Kubernetes control-plane has initialized successfully!
```

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

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

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

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

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

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

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

```
kubeadm join 172.31.14.85:6443 --token 6lysht.48enn4gmnhof6ex8 \
--discovery-token-ca-cert-hash sha256:461819c971fe032e04a78e18fde8e28755825e8468d468a2c86d88c52dba4945
```

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

```
This node has joined the cluster:  
* Certificate signing request was sent to apiserver and a response was received.  
* The Kubelet was informed of the new secure connection details.  
  
Run 'kubectl get nodes' on the control-plane to see this node join the cluster.
```

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

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

Conclusion: This experiment successfully demonstrated the creation of a Kubernetes cluster and the successful addition of all three nodes using various commands

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

Theory:

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

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

Necessary Requirements:

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

Memory: Adequate for container orchestration.

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

Note:

AWS Personal Account is preferred but we can also perform it on AWS Academy(adding some ignores in the command if any error occurs in below as the below experiment is performed on Personal Account.).

If You are using AWS Academy Account Errors you will face in kubeadm init command so you have to add some ignores with this command.

Step 1: Log in to your AWS Academy/personal account and launch a new Ec2 Instance.

Select Ubuntu as AMI and t2.medium as Instance Type, create a key of type RSA with .pem extension, and move the downloaded key to the new folder.

Note: A minimum of 2 CPUs are required so Please select t2.medium and do not forget to stop the instance after the experiment because it is not available in the free tier.

Instances (1/1) Info										
Last updated less than a minute ago Launch instances										
<input type="text" value="Q Find Instance by attribute or tag (case-sensitive)"/> All states										
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	
<input checked="" type="checkbox"/> Experiment 4	i-09f3752831db50f7d	Running	t2.medium	Initializing	View alarms +	us-east-1d	ec2-54-165-99-170.co...	54.165.99.170	-	

The screenshot shows the AWS Lambda console interface. At the top, there's a navigation bar with 'Lambda' and other service links. Below it, a search bar and a 'Create Function' button are visible. The main area is titled 'Create New Function' with the sub-section 'From scratch'. A large text input field contains the code for a Lambda function:

```
function handler(event) {  
    // Your code here  
}
```

Below the code editor, there are sections for 'Runtime' (Node.js 14.x), 'Handler' (index.handler), and 'Role' (lambda-role). A 'Create' button is at the bottom right.

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

The screenshot shows the 'Connect to instance' page for an EC2 instance. The instance ID is i-09f3752831db50f7d (Experiment 4). The 'SSH client' tab is selected. The page provides instructions for connecting via SSH:

1. Open an SSH client
2. Locate your private key file. The key used to launch this instance is Master_Ec2_Key.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
chmod 400 "Master_Ec2_Key.pem"
4. Connect to your instance using its Private IP:
172.31.20.171

Example command:

```
ssh -i "Master_Ec2_Key.pem" ubuntu@172.31.20.171
```

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

Step 3: Now open the folder in the terminal where our .pem key is stored and paste the Example command (starting with ssh -i) in the terminal.(ssh -i "Master_Ec2_Key.pem" ubuntu@ec2-54-196129-215.compute-1.amazonaws.com)

```

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

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

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

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

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

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

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

  https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

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

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

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

```

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

```

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

```

```

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

```

```

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

```

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

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

```

Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 slirp4netns amd64 1.2.1-1build2 [34.9 kB]
Get:6 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-buildx-plugin amd64 0.16.2-1~ubuntu.24.04~noble [29.9 MB]
Get:7 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce-cli amd64 5:27.2.1-1~ubuntu.24.04~noble [15.0 MB]
Get:8 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce amd64 5:27.2.1-1~ubuntu.24.04~noble [25.6 MB]
Get:9 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce-rootless-extras amd64 5:27.2.1-1~ubuntu.24.04~noble [9571 kB]
Get:10 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-compose-plugin amd64 2.29.2-1~ubuntu.24.04~noble [12.5 MB]
Fetched 122 MB in 2s (71.3 MB/s)
Selecting previously unselected package pigz.
(Reading database ... 67741 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.8-1_amd64.deb ...
Unpacking pigz (2.8-1) ...
Selecting previously unselected package containerd.io.
Preparing to unpack .../1-containedr.io_1.7.22-1_amd64.deb ...
Unpacking containerd.io (1.7.22-1) ...
Selecting previously unselected package docker-buildx-plugin.
Preparing to unpack .../2-docker-buildx-plugin_0.16.2-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-buildx-plugin (0.16.2-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce-cli.
Preparing to unpack .../3-docker-ce-cli_5%3a27.2.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce-cli (5:27.2.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce.
Preparing to unpack .../4-docker-ce_5%3a27.2.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce (5:27.2.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce-rootless-extras.
Preparing to unpack .../5-docker-ce-rootless-extras_5%3a27.2.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce-rootless-extras (5:27.2.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-compose-plugin.
Preparing to unpack .../6-docker-compose-plugin_2.29.2-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-compose-plugin (2.29.2-1~ubuntu.24.04~noble) ...
Selecting previously unselected package libltdl7:amd64.
Preparing to unpack .../7-libltdl7_2.4.7-7build1_amd64.deb ...
Unpacking libltdl7:amd64 (2.4.7-7build1) ...
Selecting previously unselected package libslirp0:amd64.
Preparing to unpack .../8-libslirp0_4.7.0-1ubuntu3_amd64.deb ...
Unpacking libslirp0:amd64 (4.7.0-1ubuntu3) ...
Selecting previously unselected package slirp4netns.
Preparing to unpack .../9-slirp4netns_1.2.1-1build2_amd64.deb ...
Unpacking slirp4netns (1.2.1-1build2) ...
Setting up docker-buildx-plugin (0.16.2-1~ubuntu.24.04~noble) ...
Setting up containerd.io (1.7.22-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /usr/lib/systemd/system/containerd.service.
Setting up docker-compose-plugin (2.29.2-1~ubuntu.24.04~noble) ...
Setting up libltdl7:amd64 (2.4.7-7build1) ...
Setting up docker-ce-cli (5:27.2.1-1~ubuntu.24.04~noble) ...
Setting up libslirp0:amd64 (4.7.0-1ubuntu3) ...
Setting up pigz (2.8-1) ...
Setting up docker-ce-rootless-extras (5:27.2.1-1~ubuntu.24.04~noble) ...
Setting up slirp4netns (1.2.1-1build2) ...
Setting up docker-ce (5:27.2.1-1~ubuntu.24.04~noble) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

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

```

```

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

```

EOF

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

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

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

Step 5: Run the below command to install Kubernets.

```
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o
/etc/apt/keyrings/kubernetes-apt-keyring.gpg
```

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

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

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

```

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

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

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

kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.

```

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

```

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

```

Now We have got an error.

So we have to perform some additional commands as follow.

sudo apt-get install -y containerd

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

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.
```

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

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

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

```
ubuntu@ip-172-31-20-171:~$ sudo mkdir -p /etc/containerd  
sudo containerd config default | sudo tee /etc/containerd/config.toml  
disabled_plugins = []  
imports = []  
oom_score = 0  
plugin_dir = ""  
required_plugins = []  
root = "/var/lib/containerd"  
state = "/run/containerd"  
temp = ""  
version = 2  
  
[cgroup]  
path = ""  
  
[debug]  
address = ""  
format = ""  
gid = 0  
level = ""  
uid = 0  
  
[grpc]  
address = "/run/containerd/containerd.sock"  
gid = 0  
max_recv_message_size = 16777216  
max_send_message_size = 16777216  
tcp_address = ""  
tcp_tls_ca = ""  
tcp_tls_cert = ""  
tcp_tls_key = ""  
uid = 0  
  
[metrics]  
address = ""  
grpc_histogram = false  
  
[plugins]  
  
[plugins."io.containerd.gc.v1.scheduler"]  
deletion_threshold = 0
```

...

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

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

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

```
sudo apt-get install -y socat
```

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

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

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

Step 6: Initialize the Kubecluster

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

```
ubuntu@ip-172-31-20-171:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using "kubeadm config images pull"
W0915 07:49:42.979651      8570 checks.go:846] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent with that used by kubeadm. It is recommended to use "registry.k8s.io/pause:3.10" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-20-171 kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 172.31.20.1]
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key
[certs] Generating "etcd/ca" certificate and key
[certs] Generating "etcd/server" certificate and key
[certs] etcd/server serving cert is signed for DNS names [ip-172-31-20-171 localhost] and IPs [172.31.20.171 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-20-171 localhost] and IPs [172.31.20.171 127.0.0.1 ::1]
[certs] Generating "healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
[certs] Generating "sa" key and public key
[kubeconfig] Using kubeconfig folder "/etc/kubernetes"
[kubeconfig] Writing "admin.conf" kubeconfig file
[kubeconfig] Writing "super-admin.conf" kubeconfig file
[kubeconfig] Writing "kubelet.conf" kubeconfig file
[kubeconfig] Writing "controller-manager.conf" kubeconfig file
[kubeconfig] Writing "scheduler.conf" kubeconfig file
[etcd] Creating static Pod manifest for local etcd in "/etc/kubernetes/manifests"
[control-plane] Creating static Pod manifest for "kube-apiserver"
[control-plane] Creating static Pod manifest for "kube-controller-manager"
[control-plane] Creating static Pod manifest for "kube-scheduler"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Starting the kubelet
[wait-control-plane] Waiting for the kubelet to boot up the control plane as static Pods from directory "/etc/kubernetes/manifests"
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[api-check] The kubelet is healthy after 502.777379ms
[api-check] Waiting for a healthy API server. This can take up to 4m0s
[api-check] The API server is healthy after 4.501245581s
[upload-config] Storing the configuration used in ConfigMap "kubeadm-config" in the "kube-system" Namespace
[kubelet] Creating a ConfigMap "kubelet-config" in namespace kube-system with the configuration for the kubelets in the cluster
[upload-certs] Skipping phase. Please see --upload-certs
[mark-control-plane] Marking the node ip-172-31-20-171 as control-plane by adding the labels: [node-role.kubernetes.io/control-plane node.kubernetes.io/exclude-from-external-load-balancers]
[mark-control-plane] Marking the node ip-172-31-20-171 as control-plane by adding the taints [node-role.kubernetes.io/control-plane:NoSchedule]
[bootstrap-token] Using token: 7acddu.inheshzwxti0372v
```

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

Your Kubernetes control-plane has initialized successfully!

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

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

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

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

You should now deploy a pod network to the cluster.

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

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

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

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

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

Add a common networking plugin called flannel as mentioned in the code.

kubectl apply -f

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

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

Step 7: Now that the cluster is up and running, we can deploy our nginx server on this cluster. Apply this deployment file using this command to create a deployment kubectl apply -f <https://k8s.io/examples/application/deployment.yaml>

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

kubectl get pods

NAME	READY	STATUS	RESTARTS	AGE
nginx-deployment-d556bf558-vz8rv	0/1	Pending	0	8s
nginx-deployment-d556bf558-wz5wc	0/1	Pending	0	8s

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

kubectl port-forward \$POD_NAME 8080:80

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

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

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

kubectl get nodes

NAME	STATUS	ROLES	AGE	VERSION
ip-172-31-20-171	Ready	control-plane	5m23s	v1.31.1

kubectl get pods

NAME	READY	STATUS	RESTARTS	AGE
nginx-deployment-d556bf558-vz8rv	1/1	Running	0	3m4s
nginx-deployment-d556bf558-wz5wc	1/1	Running	0	3m4s

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

```
kubectl port-forward $POD_NAME 8080:80
```

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

Step 8: Verify your deployment

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

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

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

```
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows  
  
PS C:\Users\bhush\OneDrive\Desktop\New folder (4)> ssh -i "Master_Ec2_Key.pem" ubuntu@ec2-54-196-129-215.compute-1.amazonaws.com  
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)  
  
* Documentation: https://help.ubuntu.com  
* Management: https://landscape.canonical.com  
* Support: https://ubuntu.com/pro  
  
System information as of Sun Sep 15 07:58:53 UTC 2024  
  
System load: 0.15 Processes: 152  
Usage of /: 55.3% of 6.71GB Users logged in: 1  
Memory usage: 20% IPv4 address for enX0: 172.31.20.171  
Swap usage: 0%  
  
* Ubuntu Pro delivers the most comprehensive open source security and  
compliance features.  
  
https://ubuntu.com/aws/pro  
  
Expanded Security Maintenance for Applications is not enabled.  
  
132 updates can be applied immediately.  
38 of these updates are standard security updates.  
To see these additional updates run: apt list --upgradable  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status
```

```
Last login: Sun Sep 15 07:27:23 2024 from 152.58.2.47  
ubuntu@ip-172-31-20-171:~$ curl --head http://127.0.0.1:8080  
HTTP/1.1 200 OK  
Server: nginx/1.14.2  
Date: Sun, 15 Sep 2024 07:59:03 GMT  
Content-Type: text/html  
Content-Length: 612  
Last-Modified: Tue, 04 Dec 2018 14:44:49 GMT  
Connection: keep-alive  
ETag: "5c0692e1-264"  
Accept-Ranges: bytes
```

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

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

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

Conclusion:

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

Experiment 5

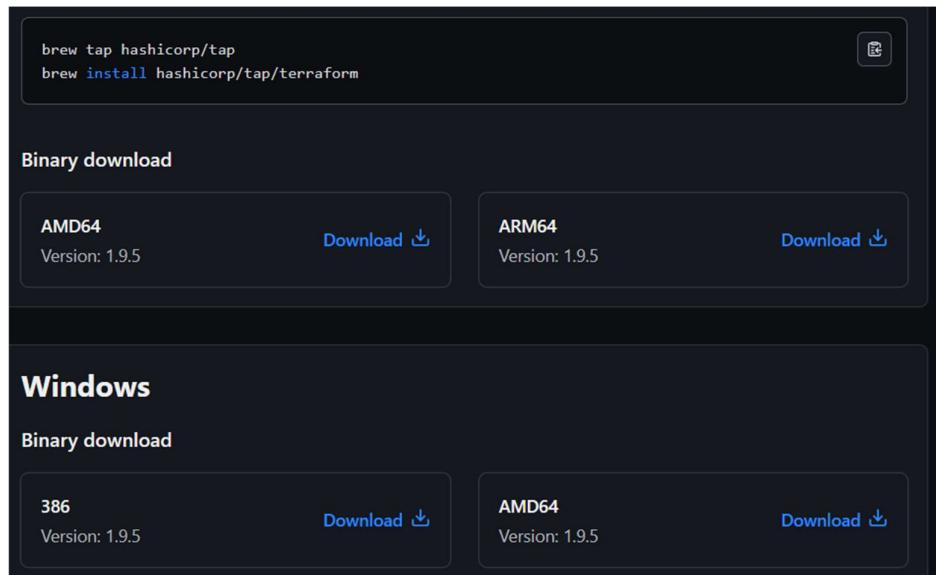
A) Installation and Configuration of Terraform in Windows

Step 1: Download terraform

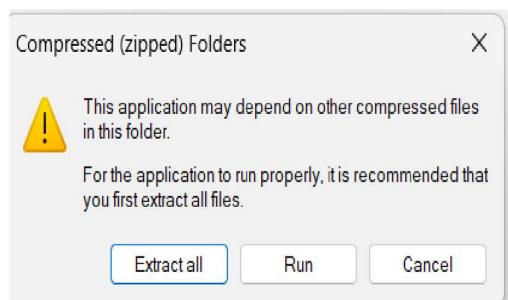
To install Terraform, First Download the Terraform Cli Utility for windows from terraforms official website

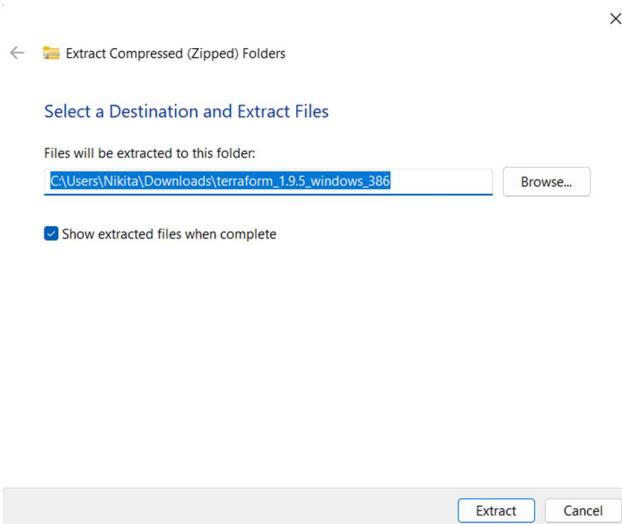
website:<https://www.terraform.io/downloads.html>

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

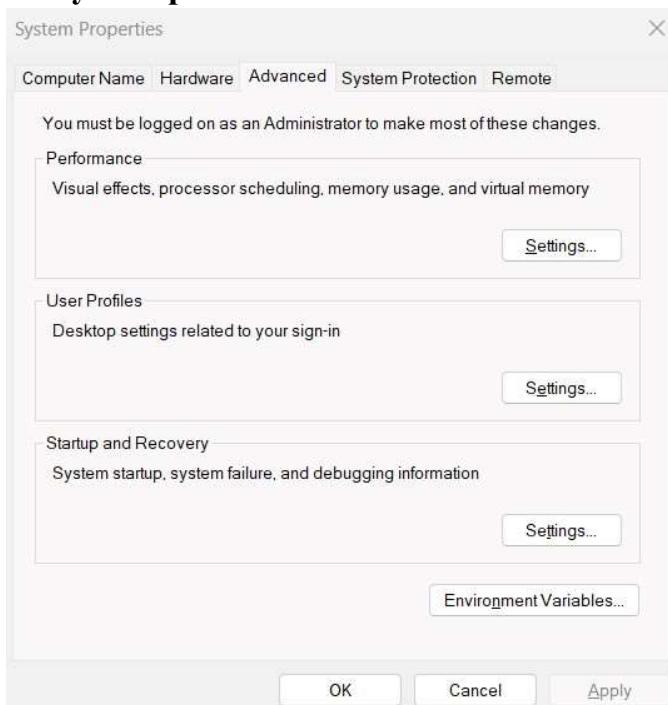


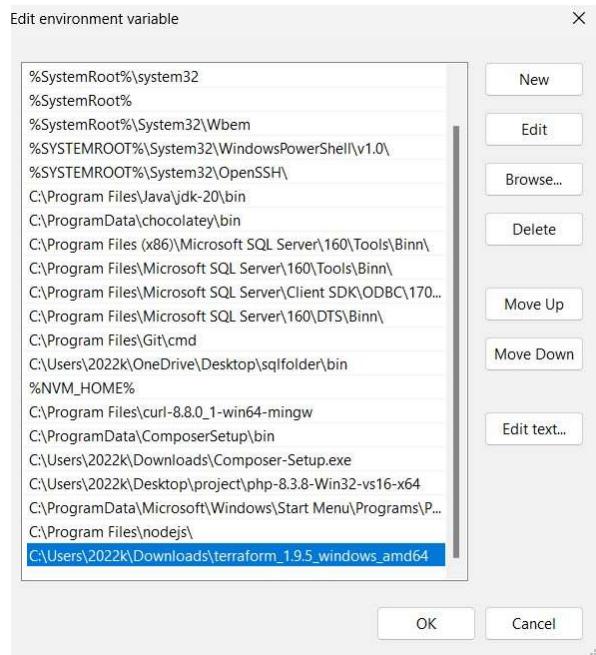
Step 2: Extract the downloaded setup file Terraform.exe in C:\Terraform directory



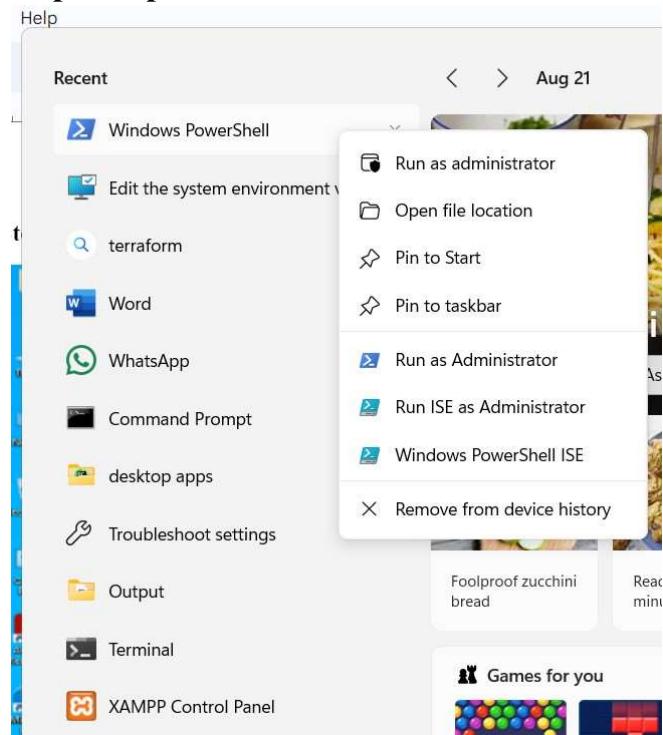


Step 3: Set the System path for Terraform in Environment Variables

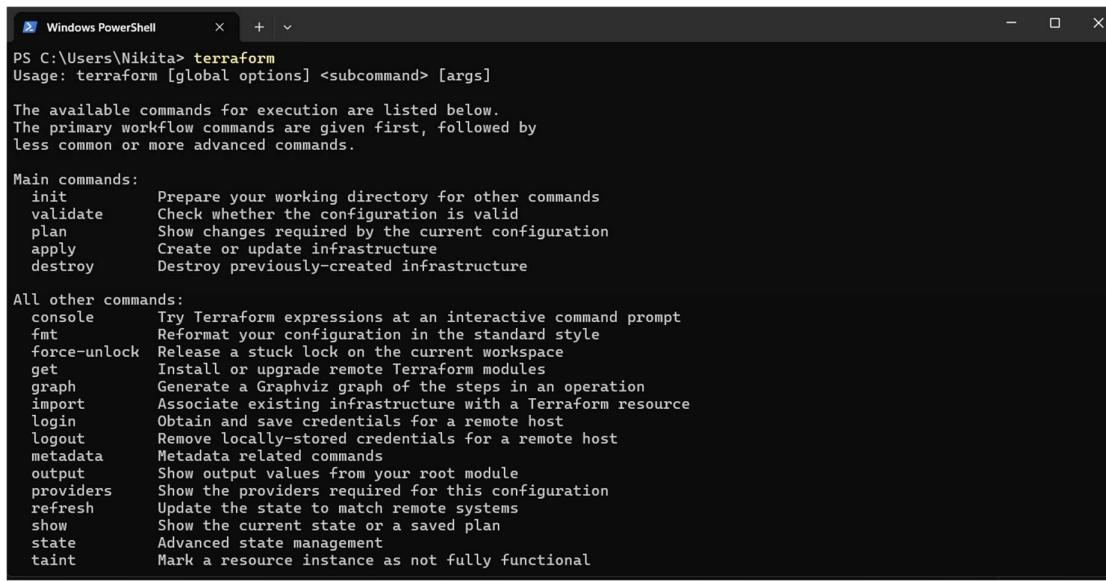




Step 4: Open PowerShell with Admin Access



Step 5 : Open Terraform in PowerShell and check its functionality



The screenshot shows a Windows PowerShell window titled "Windows PowerShell". The command "terraform" is run from the path "PS C:\Users\Nikita>". The output displays the usage information for the Terraform command-line interface, including the main commands (init, validate, plan, apply, destroy) and all other commands (console, fmt, force-unlock, get, graph, import, login, logout, metadata, output, providers, refresh, show, state, taint). Each command is followed by a brief description.

```
PS C:\Users\Nikita> 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
  providers  Show the providers required for this configuration
  refresh    Update the state to match remote systems
  show       Show the current state or a saved plan
  state      Advanced state management
  taint     Mark a resource instance as not fully functional
```

Conclusion:

We learned about installation of installation of terraform

Experiment No: 6

Aim : Exp 6 To Build, change, and destroy AWS / GCP /Microsoft Azure/DigitalOcean infrastructure Using Terraform.(S3 bucket or Docker)

Creating the docker image using terraform

1: Check the docker version and functionality if its not downloaded you can download it from <https://www.docker.com/>

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

```
C:\Users\Nikita>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
```

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

Step 2: Firstly create a new folder named ‘Docker’ in the ‘TerraformScripts’ folder.

Then create a new docker.tf

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

```
provider "docker" {
```

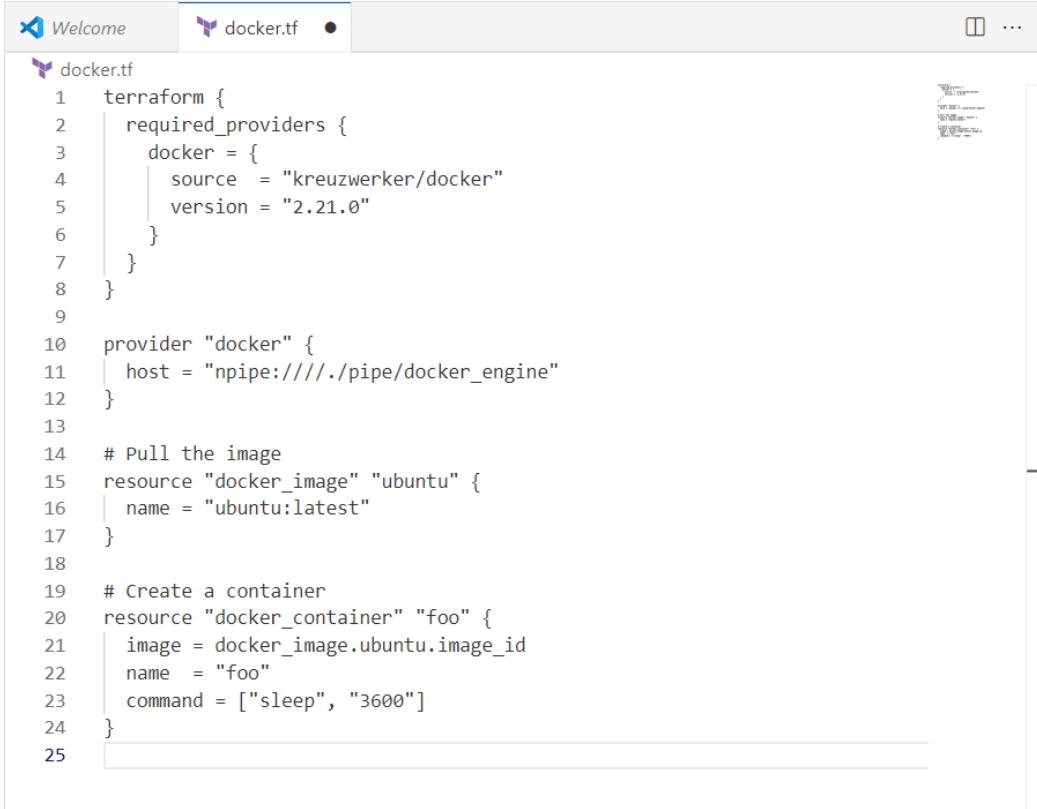
```

host = "npipe:///./pipe/docker_engine"
}

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

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

```



```

1 terraform {
2   required_providers {
3     docker = {
4       source  = "kreuzwerker/docker"
5       version = "2.21.0"
6     }
7   }
8 }
9
10 provider "docker" {
11   host = "npipe:///./pipe/docker_engine"
12 }
13
14 # Pull the image
15 resource "docker_image" "ubuntu" {
16   name = "ubuntu:latest"
17 }
18
19 # Create a container
20 resource "docker_container" "foo" {
21   image = docker_image.ubuntu.image_id
22   name  = "foo"
23   command = ["sleep", "3600"]
24 }
25

```

Step 3: Execute **Terraform Init** command to initialize the resources

```
C:\Users\Nikita\Downloads\terraform scripts\Docker>terraform init
Initializing the backend...
Initializing provider plugins...
      + provider registry.terraform.io/hashicorp/docker v2.0.0
      + provider registry.terraform.io/hashicorp/random v2.1.0
      + provider registry.terraform.io/hashicorp/local v2.0.0

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.

C:\Users\Nikita\Downloads\terraform scripts\Docker>
```

4. Execute Terraform plan to see the available resources

```
C:\Users\Nikita\Downloads\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         = [
        + "sleep",
        + "3600",
    ]
    + container_logs = (known after apply)
    + entrypoint     = (known after apply)
    + env             = (known after apply)
    + exit_code       = (known after apply)
    + gateway         = (known after apply)
    + hostname        = (known after apply)
    + id              = (known after apply)
    + image           = (known after apply)
    + init            = (known after apply)
    + ip_address      = (known after apply)
    + ip_prefix_length = (known after apply)
    + ipc_mode        = (known after apply)
    + log_driver      = (known after apply)
    + logs            = false
}
```

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

```
C:\Users\Nikita\Downloads\terraform scripts\Docker>terraform apply

Terraform used the selected providers to generate the following execution plan.
following symbols:
+ create

Terraform will perform the following actions:

# docker_container.foo will be created
+ resource "docker_container" "foo" {
  + attach           = false
  + bridge           = (known after apply)
  + command          = [
    + "sleep",
    + "3600",
  ]
  + container_logs   = (known after apply)
  + entrypoint       = (known after apply)
  + env               = (known after apply)
  + exit_code         = (known after apply)
  + gateway           = (known after apply)
  + hostname          = (known after apply)
  + id                = (known after apply)
  + image              = (known after apply)
  + init               = (known after apply)
  + ip_address         = (known after apply)
  + ip_prefix_length  = (known after apply)
  + ipc_mode           = (known after apply)
  + log_driver         = (known after apply)
}

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... [10s elapsed]
docker_image.ubuntu: Still creating... [20s elapsed]
docker_image.ubuntu: Creation complete after 21s [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2
598aubuntu:latest]
docker_container.foo: Creating...
docker_container.foo: Creation complete after 1s [id=0699033230c10aac18cab0a18b29bba4b202f29d75f7083a2496c269ea10bd44]

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

Step 6. Docker images before executing this command

```
C:\Users\Nikita\Downloads\terraform scripts\Docker>docker images
REPOSITORY      TAG          IMAGE ID      CREATED      SIZE
```

Docker images after the execution of command

```
C:\Users\Nikita\Downloads\terraform scripts\Docker>docker images
REPOSITORY      TAG          IMAGE ID      CREATED      SIZE
ubuntu          latest        edbfe74c41f8  3 weeks ago  78.1MB
```

Step 7: Execute Terraform destroy to delete the configuration, which will automatically delete the Ubuntu Container.

```
C:\Users\Nikita\Downloads\terraform scripts\Docker>terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Refreshing state... [id=0699033230c10aac18cab0a18b29bba4b202f29d75f7083a2496c269ea10bd44]

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

Terraform will perform the following actions:

# docker_container.foo will be destroyed
- resource "docker_container" "foo" {
  - attach           = false -> null
  - command          = [
    - "sleep",
    - "3600",
  ] -> null
  - cpu_shares       = 0 -> null
  - dns              = [] -> null
  - dns_opts          = [] -> null
  - dns_search         = [] -> null
  - entrypoint        = [] -> null
  - env              = [] -> null
  - gateway           = "172.17.0.1" -> null
  - group_add         = [] -> null
  - hostname          = "0699033230c1" -> null
  - id               = "0699033230c10aac18cab0a18b29bba4b202f29d75f7083a2496c269ea10bd44" -> null
}
```

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

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

Enter a value: yes

docker_container.foo: Destroying... [id=0699033230c10aac18cab0a18b29bba4b202f29d75f7083a2496c269ea10bd44]
docker_container.foo: Destruction complete after 1s
docker_image.ubuntu: Destroying... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_image.ubuntu: Destruction complete after 0s

Destroy complete! Resources: 2 destroyed.
```

Docker images After Executing Destroy step

```
C:\Users\Nikita\Downloads\terraform scripts\Docker>docker images
REPOSITORY      TAG          IMAGE ID      CREATED      SIZE
```

```
C:\Users\Nikita\Downloads\terraform scripts\Docker>terraform validate
Success! The configuration is valid.
```

```
C:\Users\Nikita\Downloads\terraform scripts\Docker>terraform providers
```

```
Providers required by configuration:
```

```
|__ provider[registry.terraform.io/kreuzwerker/docker] 2.21.0
```

Conclusion:

We learned to use terraform and run commands using it

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

1. Open up Jenkins on port 8080

S	W	Name	Last Success	Last Failure	Last Duration
		My_First_Maven	23 days: #2	23 days: #1	20 sec
		MyPipeline1	28 days: #1	N/A	9.2 sec
		Pipeline_01	1 mo 15 days: #3	N/A	9.9 sec
		WebTestDriver	1 day 16 hr: #5	1 day 16 hr: #4	13 sec

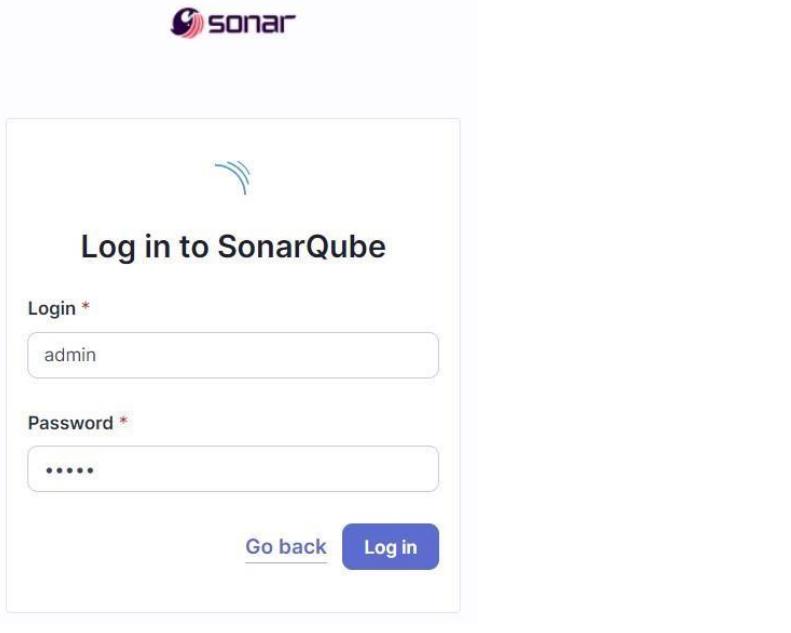
2. In a Docker container run SonarQube using this command :

- a] docker -v
- b] docker pull sonarqube
- c] docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest

```
C:\Users\adity>docker -v
Docker version 27.0.3, build 7d4bcd8

C:\Users\adity>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
4a6e73f4472de892b1ddead1abe77372a85a7b09408cce3a0abd37c5ab6b49a4
```

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



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

1 of 2

Create a local project

Project display name *

sonarqube



Project key *

sonarqube



Main branch name *

main

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

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.

[Back](#)[Create project](#)

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

The screenshot shows the Jenkins Plugins page. In the top navigation bar, there are links for 'Dashboard', 'Manage Jenkins', and 'Plugins'. Below this, there are three tabs: 'Updates' (disabled), 'Available plugins' (selected), and 'Installed plugins'. A search bar at the top right contains the text 'sonarqube scanner'. The results list shows one item: 'SonarQube Scanner 2.17.2'. This entry includes a 'Install' button, a 'Released' timestamp ('7 mo 1 day ago'), and a 'Install after restart' button. Below the main list, there are links for 'External Site/Test Integrations' and 'Build Reports'. A note at the bottom states: 'This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.'

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

Name : sonarqube

Server URL : http://localhost:9000

The screenshot shows the Jenkins System configuration page. The top navigation bar includes 'Dashboard', 'Manage Jenkins', 'System', and other system-related links. The main content area is titled 'SonarQube servers'. It contains a note: 'If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.' A checkbox labeled 'Environment variables' is checked. Below this, there is a section for 'SonarQube installations' with a link to 'List of SonarQube installations'. A new installation configuration is being added, with the following fields filled in:

- Name:** sonarqube
- Server URL:** http://localhost:9000
- Server authentication token:** - none - (dropdown menu)
- Add:** + Add ▾ (button)
- Advanced:** Advanced ▾ (dropdown menu)

At the bottom are 'Save' and 'Apply' buttons.

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

Manage Jenkins → Tools → SonarQube Scanner Installation

The screenshot shows the Jenkins interface under the 'Manage Jenkins' section, specifically the 'Tools' configuration. It displays the 'SonarQube Scanner installations' section. A new installation named 'sonarqube' is being added. The 'Install automatically' checkbox is checked. The 'Install from Maven Central' section shows the selected version as 'SonarQube Scanner 6.2.0.4584'. Below this, there's an 'Add installer' dropdown. At the bottom of the page, there are 'Save' and 'Apply' buttons.

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

The screenshot shows the 'New Item' dialog. In the 'Enter an item name' field, 'sonarqube' is typed. Under 'Select an item type', the 'Freestyle project' option is selected, shown with its icon and description: '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.' Other options like 'Maven project', 'Pipeline', 'Multi-configuration project', and 'Folder' are also listed. At the bottom right of the dialog is an 'OK' button.

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

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

The screenshot shows the 'Configure' section of the SonarQube interface under 'sonarqube > Configuration'. The 'Source Code Management' tab is selected. Under 'Source Code Management', the 'Git' option is chosen. The 'Repository URL' field contains 'https://github.com/shazforiot/MSBuild_firstproject.git'. The 'Branches to build' field is empty. At the bottom are 'Save' and 'Apply' buttons.

10. Under Build-> Execute SonarQube Scanner, enter these Analysis Properties.

Mention the SonarQube Project Key, Login, Password, Source path and Host

URL. sonar.projectKey=sonarqube sonar.login=admin sonar.password=aditya
sonar.sources=.

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

The screenshot shows the 'Configure' section of the SonarQube interface under 'sonarqube > Configuration'. The 'Build Steps' tab is selected. A new build step is being configured for 'Execute SonarQube Scanner'. In the 'Analysis properties' field, the following values are entered:
 sonar.projectKey=sonarqube
 sonar.login=admin
 sonar.host.url=http://localhost:9000
 sonar.sources=.

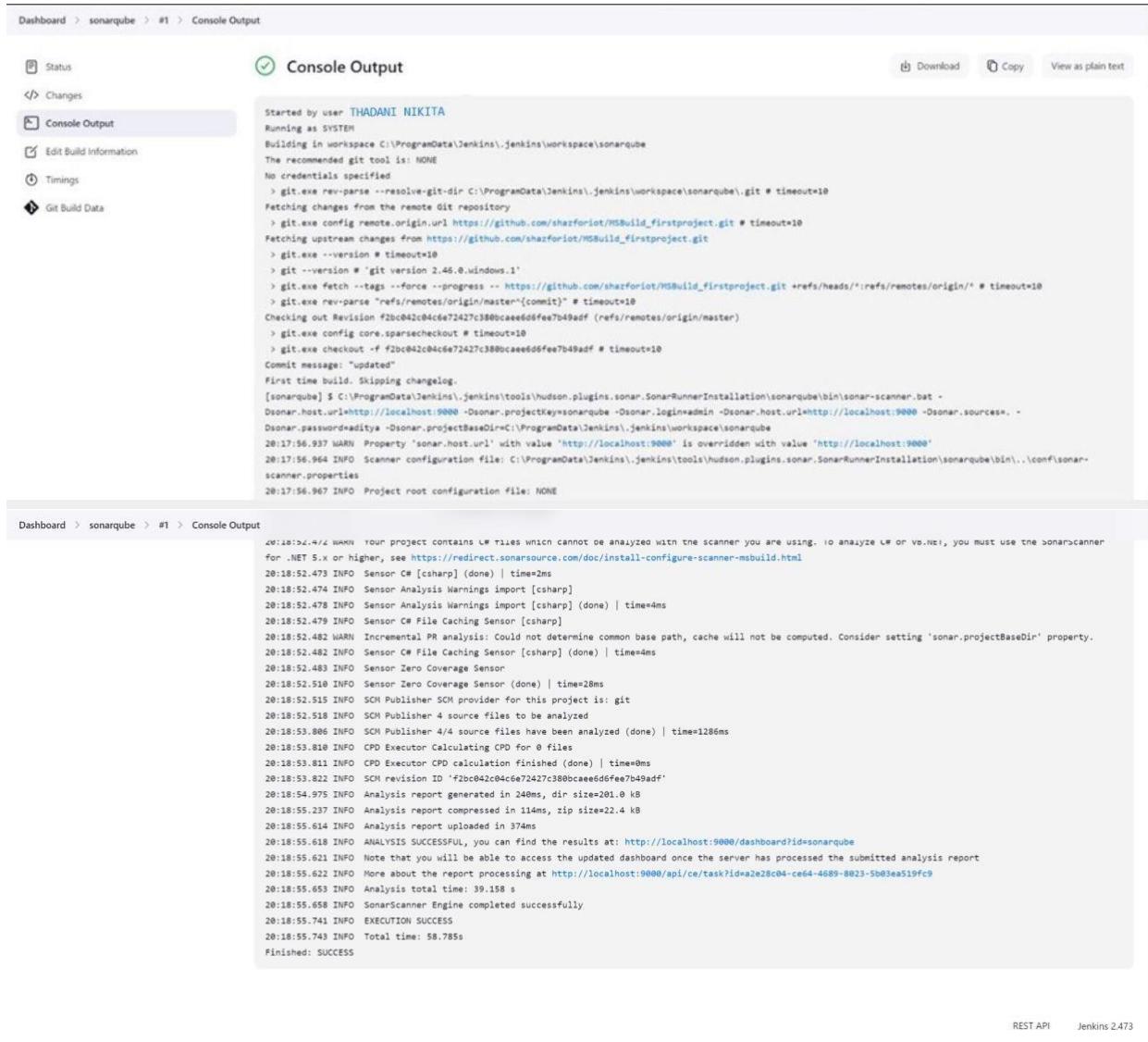
11. Go to <http://localhost:9000/admin/permissions> and allow Execute Permissions to the Admin user.

The screenshot shows the SonarQube Administration interface under the Security tab. It displays a table of global permissions for groups and users. The 'Execute Analysis' column contains checkboxes for Quality Gates and Quality Profiles. The 'Admin' user (admin) has checked both boxes for Quality Gates and Quality Profiles. Other groups like 'sonar-administrators' and 'sonar-users' have different permission sets.

	Administrator System	Administer	Execute Analysis	Create
sonar-administrators System administrators	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
sonar-users Every authenticated user automatically belongs to this group	<input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Anyone DEPRECATED Anybody who browses the application belongs to this group. If authentication is not enforced, assigned permissions also apply to non-authenticated users.	<input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
A Administrator admin	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

12. Run The Build and check the console output.

The screenshot shows the Jenkins Pipeline configuration for 'MyPipeline1'. The 'Build Now' button is visible. On the right, there is a 'sonarqube' configuration section with a 'Permalinks' link. Below it, a 'Builds' card shows a single build entry from today at 8:17PM.



The screenshot shows the Jenkins console output for a SonarQube build. The top section displays the build configuration and the command-line logs of the SonarScanner execution. The logs show the scanner analyzing the project, identifying Warnings, and generating an SCM report. The bottom section shows the detailed log output, including the analysis report generation and the final 'EXECUTION SUCCESS' message.

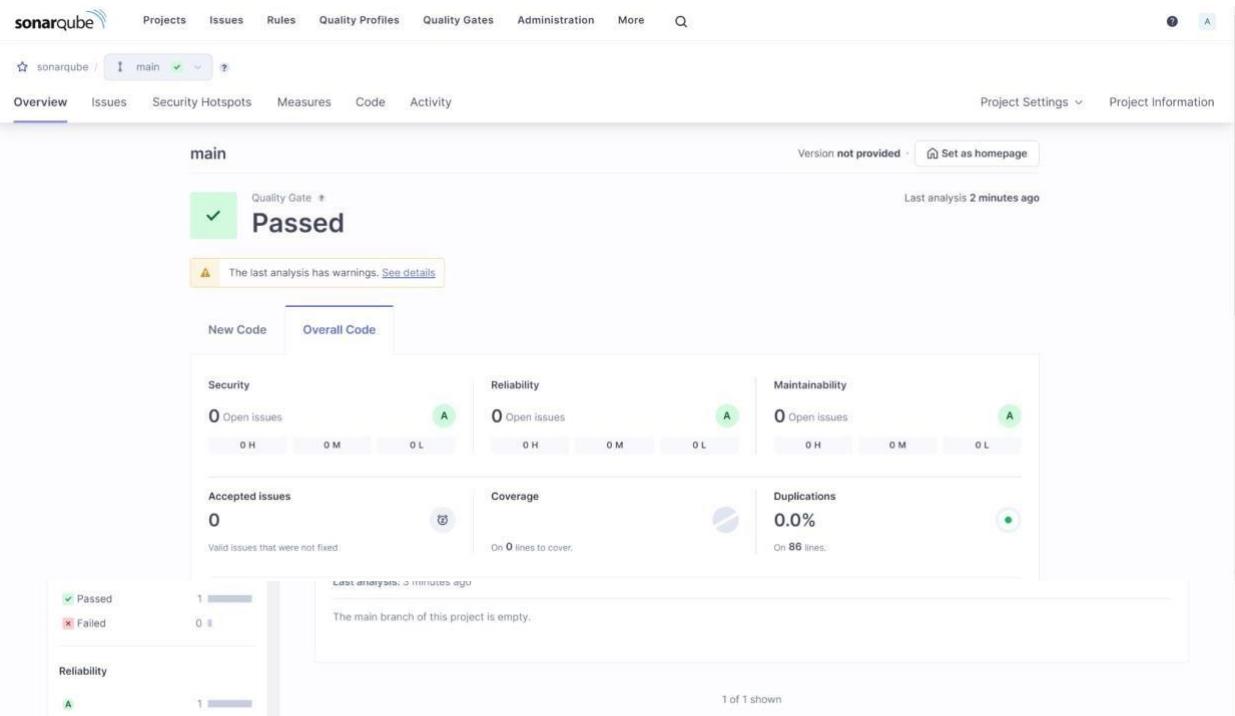
```

Started by user THADANI_NIKITA
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\sonarqube
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\jenkins\workspace\sonarqube\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/shazforiot/MSBuild_firstproject.git # timeout=10
Fetching upstream changes from https://github.com/shazforiot/MSBuild_firstproject.git
> git.exe --version # timeout=10
> git --version # 'git version 2.48.0.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/shazforiot/MSBuild_firstproject.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision f2bc042c84c6e72427c3800cae6d6fee7b49adf (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f f2bc042c84c6e72427c3800cae6d6fee7b49adf # timeout=10
Commit message: "updated"
First time build. Skipping changelog.
[sonarqube] $ C:\ProgramData\Jenkins\jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube\bin\sonar-scanner.bat -
Sonar.host.url=http://localhost:9000 -Dsonar.projectKey=sonarqube -Dsonar.login=admin -Dsonar.host.url=http://localhost:9000 -Dsonar.sources=.. -Dsonar.password=admin -Dsonar.projectBaseDir=C:\ProgramData\Jenkins\jenkins\workspace\sonarqube
20:17:56.937 WARN Property 'sonar.host.url' with value 'http://localhost:9000' is overridden with value 'http://localhost:9000'
20:17:56.964 INFO Scanner configuration file: C:\ProgramData\Jenkins\jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube\bin..\conf\sonar-scanner.properties
20:17:56.967 INFO Project root configuration file: NONE

20:18:52.472 WARN Your project contains L# files which cannot be analyzed with the scanner you are using. To analyze L# or VB.NET, you must use the sonarScanner for .NET 5.x or higher, see https://redirect.sonarsource.com/doc/install-configure-scanner-msbuild.html
20:18:52.473 INFO Sensor CM [csharp] (done) | time=2ms
20:18:52.474 INFO Sensor Analysis Warnings import [csharp]
20:18:52.476 INFO Sensor Analysis Warnings import [csharp] (done) | time=4ms
20:18:52.478 INFO Sensor CM File Cache Sensor [csharp]
20:18:52.482 WARN Incremental PR analysis: Could not determine common base path, cache will not be computed. Consider setting 'sonar.projectBaseDir' property.
20:18:52.482 INFO Sensor CM File Cache Sensor [csharp] (done) | time=4ms
20:18:52.483 INFO Sensor Zero Coverage Sensor
20:18:52.518 INFO Sensor Zero Coverage Sensor (done) | time=28ms
20:18:52.515 INFO SCM Publisher SCM provider for this project is: git
20:18:52.518 INFO SCM Publisher 4 source files to be analyzed
20:18:53.808 INFO SCM Publisher 4/4 source files have been analyzed (done) | time=1286ms
20:18:53.810 INFO CPD Executor Calculating CPD for 0 files
20:18:53.811 INFO CPD Executor CPD calculation finished (done) | time=0ms
20:18:53.822 INFO SCM revision ID 'f2bc042c84c6e72427c3800cae6d6fee7b49adf'
20:18:54.975 INFO Analysis report generated in 240ms, dir size=201.0 kB
20:18:55.237 INFO Analysis report compressed in 114ms, zip size=22.4 kB
20:18:55.614 INFO Analysis report uploaded in 37ms
20:18:55.616 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube
20:18:55.622 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
20:18:55.622 INFO More about the report processing at http://localhost:9000/api/ce/task?id=a2e28c04-ce64-4689-8023-5b03ea519fc9
20:18:55.653 INFO Analysis total time: 39.158 s
20:18:55.656 INFO SonarScanner Engine completed successfully
20:18:55.744 INFO EXECUTION SUCCESS
20:18:55.743 INFO Total time: 58.785s
Finished: SUCCESS

```

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



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

Conclusion:

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

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

1. Open up Jenkins Dashboard on localhost:8080.

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

- Header:** Jenkins logo, search bar (Search (CTRL+K)), user info (Aditya Nagesh Raorane), and log out button.
- Left Sidebar:**
 - New Item
 - Build History
 - Project Relationship
 - Check File Fingerprint
 - Manage Jenkins
 - My Views
 - Build Queue: Shows "No builds in the queue."
 - Build Executor Status: Shows "0/2"
- Main Content:**

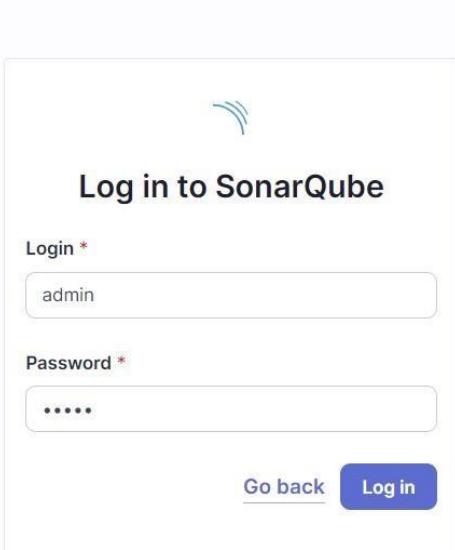
S	W	Name	Last Success	Last Failure	Last Duration
✓	Cloud	My_First_Maven	23 days #2	23 days #1	20 sec
✓	Sun	MyPipeline1	28 days #1	N/A	9.2 sec
✓	Sun	Pipeline_01	1 mo 15 days #3	N/A	9.9 sec
✓	Sun	sonarqube	13 min #1	N/A	1 min 2 sec
✓	Cloud	WebTestDriver	1 day 18 hr #5	1 day 18 hr #4	13 sec
- Bottom:** Icon selection buttons (S, M, L) and a "ooo" link.

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

```
C:\Users\adity>docker -v
Docker version 27.0.3, build 7d4bcd8

C:\Users\adity>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87defa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
4a6e73f4472de892b1dead1abe77372a85a7b09408cce3a0abd37c5ab6b49a4
```

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



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

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

2 of 2

Set up project for Clean as You Code

This 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.

[Back](#)[Create project](#)

Setup the project and come back to Jenkins Dashboard.

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

7. Under Pipeline Script, enter the following -

```

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

```

The screenshot shows the Jenkins Pipeline configuration page for a project named 'sonarqube-test'. The 'Pipeline' tab is selected. In the 'Definition' section, the 'Pipeline script' dropdown is set to 'Script'. Below it, a code editor displays a Groovy script:

```

1 * node {
2 *   stage('Cloning the GitHub Repo') {
3 *     git "https://github.com/shafioriot/qa.git"
4 *
5 *   stage('SonarQube analysis') {
6 *     withSonarQubeEnv('sonarqube') {
7 *       bat "C:\Users\aditya\Downloads\sonar-scanner-cli-6.1.0.4477-windows-x64\sonar-scanner-6.1.0.4477-windows-x64\bin\sonar-scanner.bat \
8 * -D sonar.login=admin \
9 * -D sonar.password=aditya \
10 * -D sonar.projectKey=sonarqube-test \
11 * -D sonar.exclusions=vendor/**,resources/**/*,java \
12 * -D sonar.host.url=http://localhost:9000/"
13 *     }
14 *   }
15 * }

```

Below the script, there is a checked checkbox for 'Use Groovy Sandbox' and a link to 'Pipeline Syntax'. At the bottom are 'Save' and 'Apply' buttons.

REST API Jenkins 2.473

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

8. Run The Build.



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

The screenshot shows the Jenkins Pipeline 'sonarqube-test' dashboard. On the left, there's a sidebar with various options: Status (selected), Changes, Build Now, Configure, Delete Pipeline, Full Stage View, SonarQube, Stages, Rename, and Pipeline Syntax. Below this is a 'Builds' section with a search bar and a filter icon, showing a single build entry: '#1 Sep 17 21:26 No Changes'. To the right, the main area displays the 'Stage View' for build #1. It shows two stages: 'Cloning the GitHub Repo' (1s) and 'SonarQube analysis' (14min 37s). Below the stage view is a summary: 'Average stage times: (Average full run time: ~14min 39s)'. Underneath the stage view is a 'Permalinks' section with a bulleted list of links:

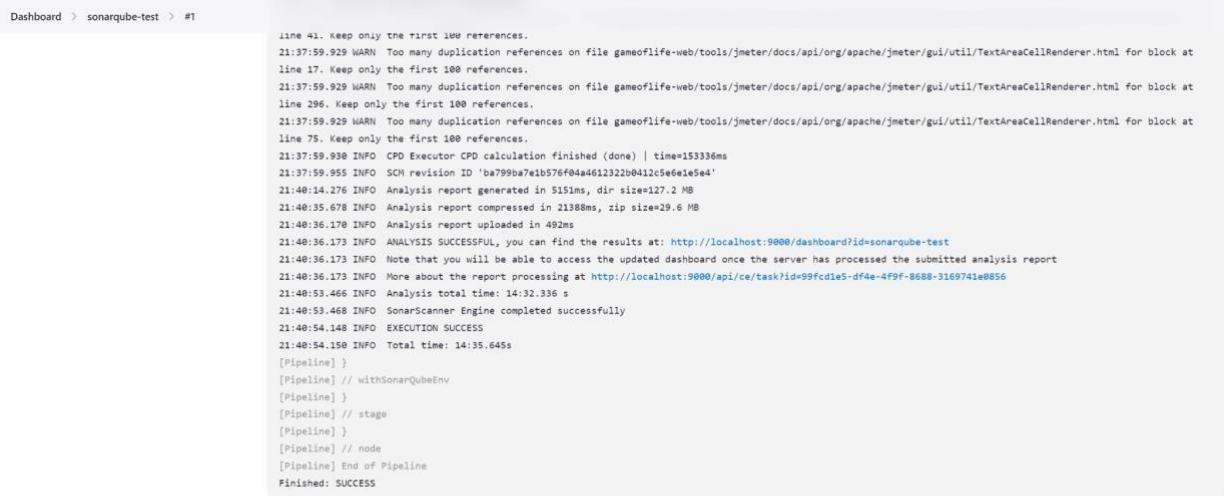
- Last build (#1), 15 min ago
- Last stable build (#1), 15 min ago
- Last successful build (#1), 15 min ago
- Last completed build (#1), 15 min ago

The screenshot shows the Jenkins Pipeline 'MyPipeline1' configuration page. The left sidebar includes options: Status, Changes, Console Output (selected), View as plain text, Edit Build Information, Timings, Git Build Data, Pipeline Overview, Pipeline Console, Thread Dump, Pause/resume, Replay, Pipeline Steps, and Workspaces. The main area is titled 'Console Output' and shows the command-line output of the pipeline's execution. The output starts with 'Started by user THADANI NIKITA' and continues through several git commands for cloning, fetching, and checking out code from a GitHub repository.

```

Started by user THADANI NIKITA
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\jenkins\workspace\sonarqube-test
[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\sonarqube-test\.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 2.46.0.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/shazforiot/GOL.git +refs/heads/*:refs/remotes/origin/*
> git.exe config remote.origin.master.refspec +refs/heads/master:refs/remotes/origin/master
Checking out Revision ba7990a7e1b576f04a461232208412c5e6e1e5e4 (refs/remotes/origin/master)
> git.exe config core.sparsecheckout true
> git.exe checkout -f ba7990a7e1b576f04a461232208412c5e6e1e5e4 # timeout=10
> git.exe branch -a -v --no-abbrev # timeout=10
> git.exe branch -D master # timeout=10
> git.exe checkout -b master ba7990a7e1b576f04a461232208412c5e6e1e5e4 # timeout=10
Commit message: "Update Jenkinsfile"

```



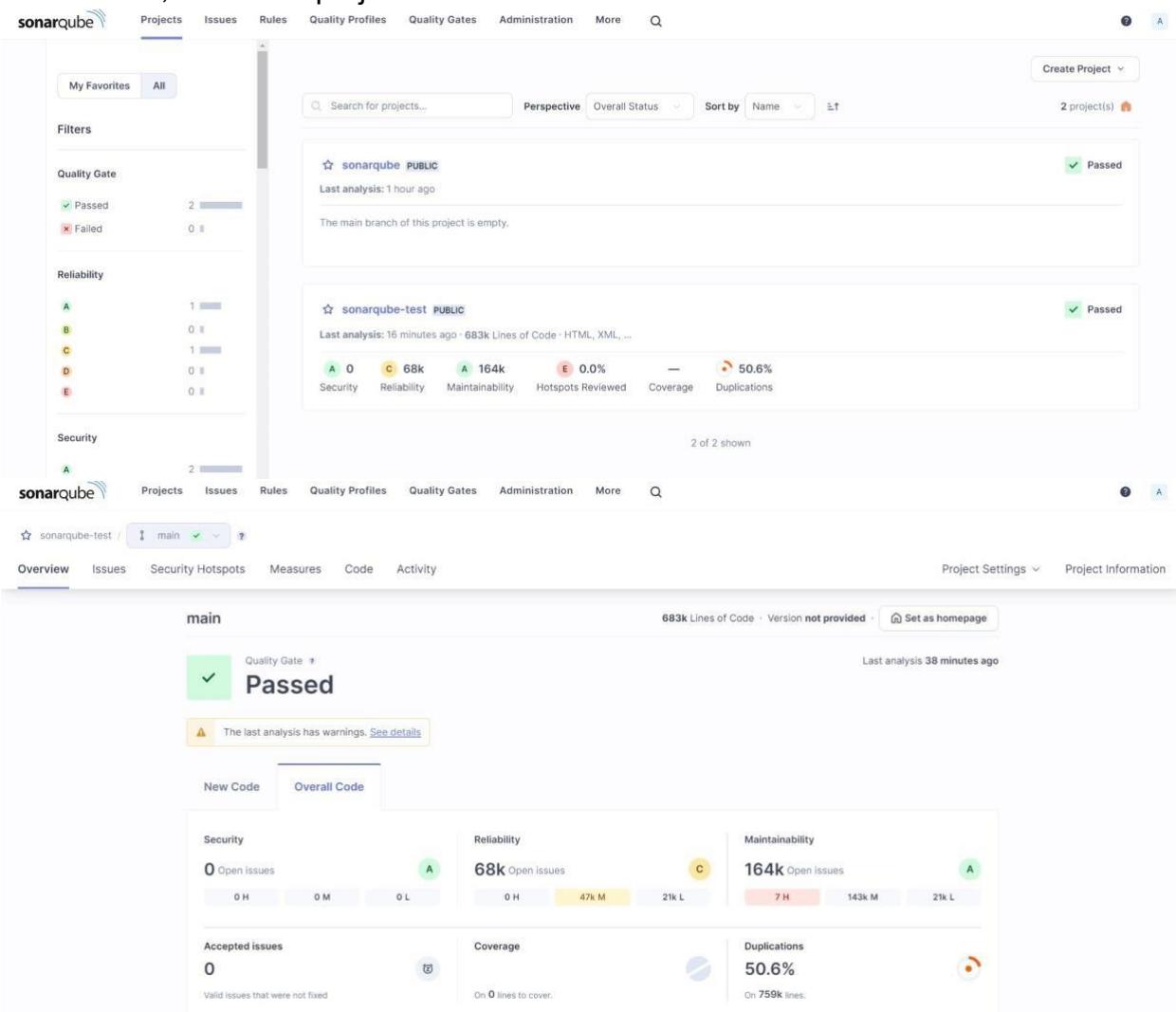
```

line 41. Keep only the first 100 references.
21:37:59.929 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at
line 17. Keep only the first 100 references.
21:37:59.929 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.
21:37:59.929 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.
21:37:59.930 INFO CPD Executor CPD calculation finished (done) | time=153336ms
21:37:59.955 INFO SCH revision ID 'ba799ba7e1b576f04ad61232b0d412c5e61e5e4'
21:40:14.276 INFO Analysis report generated in 515ms, dir size=127.2 MB
21:40:35.678 INFO Analysis report compressed in 2138ms, zip size=29.6 MB
21:40:36.178 INFO Analysis report uploaded in 492ms
21:40:36.173 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube-test
21:40:36.173 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
21:40:36.173 INFO More about the report processing at http://localhost:9000/api/ce/task?id=99fcde5-df4e-4f9f-8688-3169741e0856
21:40:53.466 INFO Analysis total time: 14:32.336 s
21:40:53.468 INFO SonarScanner Engine completed successfully
21:40:54.148 INFO EXECUTION SUCCESS
21:40:54.150 INFO Total time: 14:35.645s
[Pipeline]
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

REST API Jenkins 2.473

10. After that, check the project in SonarQube.



The screenshot shows the SonarQube dashboard with two projects listed:

- sonarqube PUBLIC**: Last analysis: 1 hour ago. Status: Passed. The main branch is empty.
- sonarqube-test PUBLIC**: Last analysis: 16 minutes ago - 683k Lines of Code - HTML, XML, ... Status: Passed. Key metrics: Security (0), Reliability (68k), Maintainability (164k), Hotspots Reviewed (0.0%), Coverage (50.6%), Duplications (50.6%).

On the main dashboard, the project **main** is selected. It shows a Quality Gate status of **Passed**. A warning message indicates there are warnings. The dashboard provides an overview of code quality metrics:

Category	Value	Grade
Security	0 Open issues	A
Reliability	68k Open issues	C
Maintainability	164k Open issues	A
Accepted issues	0	
Coverage	50.6%	
Duplications	50.6%	

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

11. Code Problems Open Issues

SonarQube interface showing the 'Measures' tab. The left sidebar displays metrics such as Security Review, Duplications, Size, Complexity, and Issues. The main panel shows an overall code count of 210,549 open issues. A detailed tree view on the right lists issues categorized by file and package, with counts ranging from 0 to 603.

Consistency

SonarQube interface showing the 'Issues' tab. The left sidebar filters for 'Consistency' issues. The main panel lists specific consistency-related code smells, such as 'Insert a <!DOCTYPE> declaration to before this <html> tag.' and 'Remove this deprecated "width" attribute.', along with their severity and impact levels.

Intentionality

sonarqube-test / main

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

My Issues All

Bulk Change Select Issues Navigate to issue 13,887 issues 59d effort

Filters Clear All Filters

Issues in new code

Clean Code Attribute

- Consistency 197k
- Intentionality 14k
- Adaptability 0
- Responsibility 0

Add to selection Ctrl + click

Software Quality

gameoflife-acceptance-tests/Dockerfile

- Use a specific version tag for the image. Maintainability Intentionality 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. Maintainability Intentionality 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. Maintainability Intentionality No tags
- Open Not assigned L12 - 5min effort - 4 years ago - ⚡ Code Smell ⚡ Major

Embedded database should be used for evaluation purposes only

Code Smells

sonarqube-test / main

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

My Issues All

Bulk Change Select Issues Navigate to issue 253 issues 2d 5h effort

Severity ?

- High 0
- Medium 0
- Low 253

Type ?

- Bug 14k
- Vulnerability 0
- Code Smell 253

Add to selection Ctrl + click

Scope

Status

Security Category

gameoflife-web/tools/jmeter/printable_docs/building.html

- Add an "alt" attribute to this image. Reliability Intentionality accessibility wcag2-a
- Open Not assigned L29 - 5min effort - 4 years ago - ⚡ Code Smell ⚡ Minor

gameoflife-web/tools/jmeter/printable_docs/changes.html

- Add an "alt" attribute to this image. Reliability Intentionality accessibility wcag2-a
- Open Not assigned L31 - 5min effort - 4 years ago - ⚡ Code Smell ⚡ Minor

gameoflife-web/tools/jmeter/printable_docs/changes_history.html

- Add an "alt" attribute to this image. Reliability Intentionality accessibility wcag2-a
- Open Not assigned L31 - 5min effort - 4 years ago - ⚡ Code Smell ⚡ Minor

Embedded database should be used for evaluation purposes only

The embedded database will not scale. It will not support connections to remote instances of SonarQube, and there is no support for retrieving issue data out of its own database instance.

Bugs

sonarqube-test / main

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

Severity

- High: 0
- Medium: 14k
- Low: 0

Type

- Bug: 14k
- Vulnerability: 0
- Code Smell: 253

Scope

Status

Security Category

Select issues Navigate to issue 13,619 issues 56d effort

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

Add "lang" and/or "xmllang" attributes to this "<html>" element
Reliability (Yellow) Intentionality (Blue)
Open Not assigned L1 - 2min effort - 4 years ago - ⚡ Bug ⚡ Major accessibility wcag2-a

Add "<th>" headers to this "<table>".
Reliability (Yellow) Intentionality (Blue)
Open Not assigned L9 - 2min effort - 4 years ago - ⚡ Bug ⚡ Major accessibility wcag2-a

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

Add "lang" and/or "xmllang" attributes to this "<html>" element
Reliability (Yellow) Intentionality (Blue)
Open Not assigned L1 - 2min effort - 4 years ago - ⚡ Bug ⚡ Major accessibility wcag2-a

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

Add "lang" and/or "xmllang" attributes to this "<html>" element
Reliability (Yellow) Intentionality (Blue)
Open Not assigned L9 - 2min effort - 4 years ago - ⚡ Bug ⚡ Major accessibility wcag2-a

Embedded database should be used for evaluation purposes only

Reliability

sonarqube-test / main

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

Clean Code Attribute

- Consistency: 33k
- Intentionality: 14k
- Adaptability: 0
- Responsibility: 0

Add to selection Ctrl + click

Software Quality

- Security: 0
- Reliability: 14k
- Maintainability: 0

Severity

- High: 0
- Medium: 14k

Select issues Navigate to issue 13,619 issues 56d effort

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

Add "lang" and/or "xmllang" attributes to this "<html>" element
Reliability (Yellow) Intentionality (Blue)
Open Not assigned L1 - 2min effort - 4 years ago - ⚡ Bug ⚡ Major accessibility wcag2-a

Add "<th>" headers to this "<table>".
Reliability (Yellow) Intentionality (Blue)
Open Not assigned L9 - 2min effort - 4 years ago - ⚡ Bug ⚡ Major accessibility wcag2-a

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

Add "lang" and/or "xmllang" attributes to this "<html>" element
Reliability (Yellow) Intentionality (Blue)
Open Not assigned L1 - 2min effort - 4 years ago - ⚡ Bug ⚡ Major accessibility wcag2-a

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

Add "lang" and/or "xmllang" attributes to this "<html>" element
Reliability (Yellow) Intentionality (Blue)
Open Not assigned L9 - 2min effort - 4 years ago - ⚡ Bug ⚡ Major accessibility wcag2-a

Embedded database should be used for evaluation purposes only

Duplicates

	File Path	Duplicated Lines (%)	Duplicated Lines
1	gameoflife-web/tools/jmeter/docs/api/org/apache/jm.../ReportCellRenderer.html	92.4%	1,282
2	gameoflife-web/tools/jmeter/docs/api/org/apache/jo.../RightAlignRenderer.html	92.4%	1,198
3	gameoflife-web/tools/jmeter/docs/api/org/apache/jm.../JMeterCellRenderer.html	92.1%	1,281
4	gameoflife-web/tools/jmeter/docs/api/org/apache/jm.../FunctionHelper.html	89.5%	1,070
5	gameoflife-web/tools/jmeter/docs/api/org/apache/jm.../AjpSamplerGui.html	89.0%	1,219

Security Hotspot

The tomcat image runs with root as the default user. Make sure it is safe here. [Review](#)

Running containers as a privileged user is security-sensitive docker:S6471

Status: To review

This security hotspot needs to be reviewed to assess whether the code poses a risk.

Where is the risk? What's the risk? Assess the risk How can I fix it? Activity

```
gameoflife-web/Dockerfile
1 FROM tomcat:8-jre8
2
3
4
5
6
7
8
9
```

Open in IDE

Cyclomatic Complexity

The screenshot shows the SonarQube interface for a project named "sonarqube-test". The left sidebar has a "Measures" tab selected, showing various metrics: Security, Reliability, Maintainability, Security Review, Duplications, Size, Complexity, and Issues. The "Complexity" section is expanded, showing a sub-section for "Cyclomatic Complexity" with a value of 1,112. The main panel displays a hierarchical tree of files under "sonarqube-test". The files and their cyclomatic complexity values are:

- gameoflife-acceptance-tests: 1,112
- gameoflife-build: 18
- gameoflife-core: 1,094
- gameoflife-deploy: 1,094
- gameoflife-web: 1,094
- pom.xml: 1,094

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

Conclusion:

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

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

Prerequisites: AWS Free Tier

Steps:

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

The screenshot shows the AWS EC2 Instances page with three instances listed:

- aws-cloud9-a... (Instance ID: i-031453e9a581a8b02, Status: Running, Type: t2.micro, Checks: 2/2 passed, Zone: us-east-1d, Public IP: ec2-44-22)
- My Web server (Instance ID: i-06a85947b2d9e4072, Status: Running, Type: t2.micro, Checks: 2/2 passed, Zone: us-east-1a, Public IP: ec2-35-17)
- nagios-host (Instance ID: i-07ae956bdd4ee100c, Status: Initializing, Type: t2.micro, Checks: 0/2, Zone: us-east-1c, Public IP: ec2-34-20)

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

The screenshot shows the AWS Security Groups Inbound rules page for a specific security group. It lists two rules:

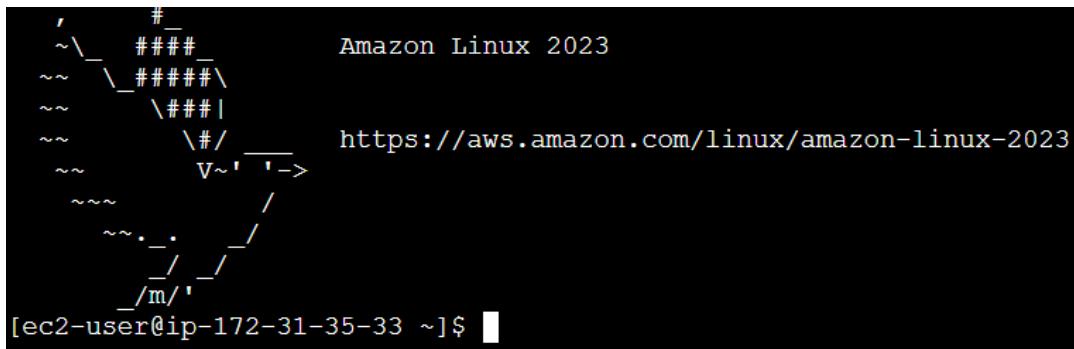
- (sgr-0d3a046f7b9fc67b5) IPv4 All traffic All
- (sgr-0d3a046f7b9fc67b5) IPv4 All traffic All

The "Edit inbound rules" dialog box shows the detailed configuration of these rules:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0ec19557ab9350565	SSH	TCP	22	Custom	
-	HTTP	TCP	80	Anywhere-...	
-	All ICMP - IPv6	IPv6 ICMP	All	Anywhere-...	
-	HTTPS	TCP	443	Anywhere-...	
-	All traffic	All	All	Anywhere-...	
-	Custom TCP	TCP	5666	Anywhere-...	
-	All ICMP - IPv4	ICMP	All	Anywhere-...	

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

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



4. Update the package indices and install the following packages using yum

```
sudo yum update  
sudo yum install httpd php  
sudo yum install gcc glibc glibc-common  
sudo yum install gd gd-devel
```

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

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

```
Re-type new password:  
[ec2-user@ip-172-31-35-33 ~]$ sudo passwd nagios  
Changing password for user nagios.  
New password:  
Re-type new password:  
passwd: all authentication tokens updated successfully.  
[ec2-user@ip-172-31-35-33 ~]$
```

6. Create a new user group

```
sudo groupadd nagcmd
```

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

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

8. Create a new directory for Nagios downloads

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

9. Use wget to download the source zip files.

```
wget
```

<http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.0.8.tar.gz>

wget

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

```
[ec2-user@ip-172-31-35-33 downloads]$ wget http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz
--2024-10-04 04:01:10-- http://nagios-plugins.org/download/nagios-plugins-2.0.3.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|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2659772 (2.5M) [application/x-gzip]
Saving to: 'nagios-plugins-2.0.3.tar.gz'

nagios-plugins-2.0.3.tar.gz      0%[=====]   287.47K  1.33MB/s
nagios-plugins-2.0.3.tar.gz    11%[=====]  2.54M  6.88MB/s  in 0.4s
nagios-plugins-2.0.3.tar.gz    100%[=====]  2.54M  6.88MB/s  in 0.4s

2024-10-04 04:01:10 (6.88 MB/s) - 'nagios-plugins-2.0.3.tar.gz' saved [2659772/2659772]
```

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

tar zxvf nagios-4.0.8.tar.gz

11. Run the configuration script with the same group name you previously created.

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

```
*** Configuration summary for nagios 4.0.8 08-12-2014 ***:

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: ${prefix}/var/nagios.lock
  Check result directory: ${prefix}/var/spool/checkresults
  Init directory: /etc/rc.d/init.d
  Apache conf.d directory: /etc/httpd/conf.d
  Mail program: /bin/mail
  Host OS: linux-gnu
  IOBroker Method: epoll

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

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

12. Compile the source code.

```
make all
```

13. Install binaries, init script and sample config files. Lastly, set permissions on the external command directory.

```
sudo make install
sudo make install-init
sudo make install-config
sudo make install-commandmode
```

```
[ec2-user@ip-172-31-35-33 nagios-4.0.8]$ sudo make install-commandmode
/usr/bin/install -c -m 775 -o nagios -g nagcmd -d /usr/local/nagios/var/rw
chmod g+s /usr/local/nagios/var/rw

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

14. Edit the config file and change the email address.

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

```
#####
#
# 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                nagios@localhost       ; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS *****
```

15. Configure the web interface.

```
sudo make install-webconf
```

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

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

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

16. Create a nagiosadmin account for nagios login along with password. You'll have to specify the password twice.

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

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

17. Restart Apache

```
sudo service httpd restart
```

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

```
cd ~/downloads
tar zxvf nagios-plugins-2.0.3.tar.gz
```

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

19. Compile and install plugins

```
cd nagios-plugins-2.0.3
./configure --with-nagios-user=nagios --with-nagios-group=nagios
make
sudo make install
```

20. Start Nagios

Add Nagios to the list of system services

```
sudo chkconfig --add nagios
sudo chkconfig nagios on
```

Verify the sample configuration files

```
sudo /usr/local/nagios/bin/nagios -v
```

/usr/local/nagios/etc/nagios.cfg If there are no errors, you can go ahead and

start Nagios.

```
sudo service nagios start
```

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

If facing error like this:

```
[ec2-user@ip-172-31-91-91 nagios-plugins-2.0.3]$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
Nagios Core 4.0.8
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 08-12-2014
License: GPL

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

Run these commands:

```
sudo mkdir -p /usr/local/nagios/var/spool/checkresults sudo
chown nagios:nagios /usr/local/nagios/var/spool/checkresults sudo
chmod 775 /usr/local/nagios/var/spool/checkresults
```

21. Check the status of Nagios

```
sudo systemctl status nagios
```

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

Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: wproc: Registry request: name=Core Worker 68063;pid=68063
Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: wproc: Registry request: name=Core Worker 68062;pid=68062
Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: wproc: Registry request: name=Core Worker 68064;pid=68064
Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: wproc: Registry request: name=Core Worker 68061;pid=68061
Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: Warning: Could not open object cache file '/usr/local/nagios/var/objectcache'
Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmx2N'
Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: Successfully launched command file worker with pid 68065
Sep 29 08:04:39 ip-172-31-91-91.ec2.internal nagios[68059]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmxTmQ'
Sep 29 08:04:49 ip-172-31-91-91.ec2.internal nagios[68059]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmpAfY'
Sep 29 08:04:59 ip-172-31-91-91.ec2.internal nagios[68059]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmxCtQ'
[lines 1-26/26 (END)]
```

If you are facing error again:

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

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

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

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

Set permissions: Ensure the directory has the right permissions:

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

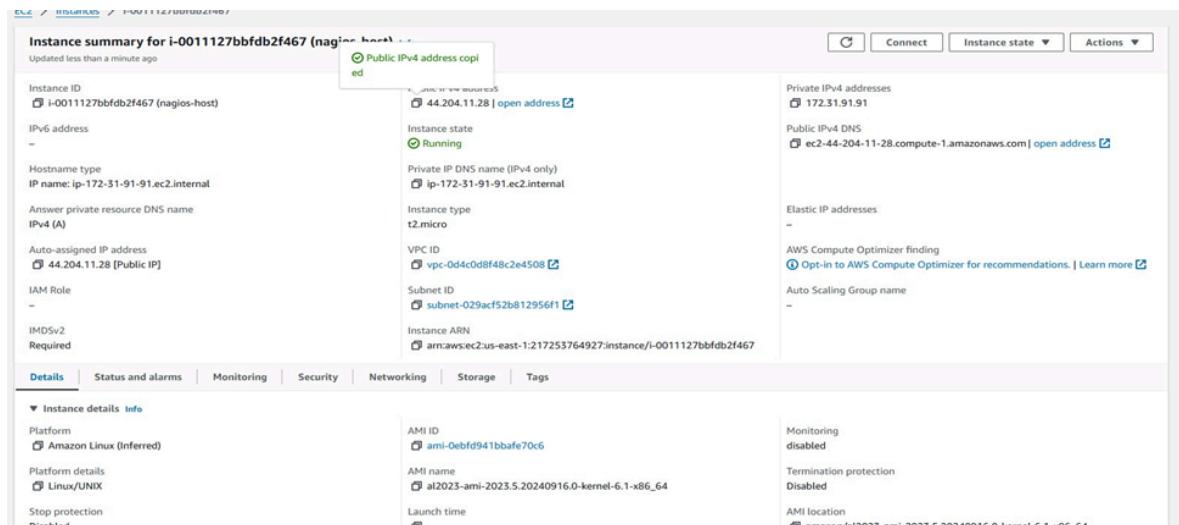
Restart Nagios: After adjusting the ownership and permissions, restart the Nagios service:

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

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

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

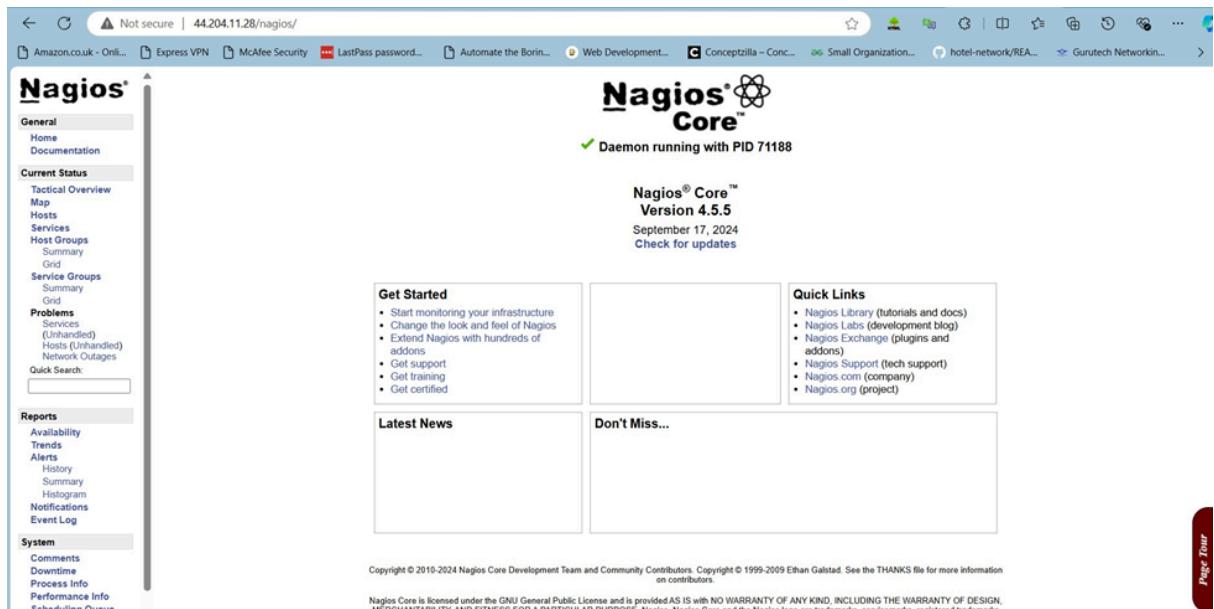
22. Go back to EC2 Console and copy the Public IP address of this instance



23. Open up your browser and look for

http://<your_public_ip_address>/nagios Enter username as nagiosadmin and password which you set in Step 16.

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



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

Conclusion:

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

Adv DevOps Practical 10

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

Theory:

Port and Service Monitoring

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

Windows/Linux Server Monitoring

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

Prerequisites:

AWS Academy or Personal account.

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

Monitoring Using Nagios:

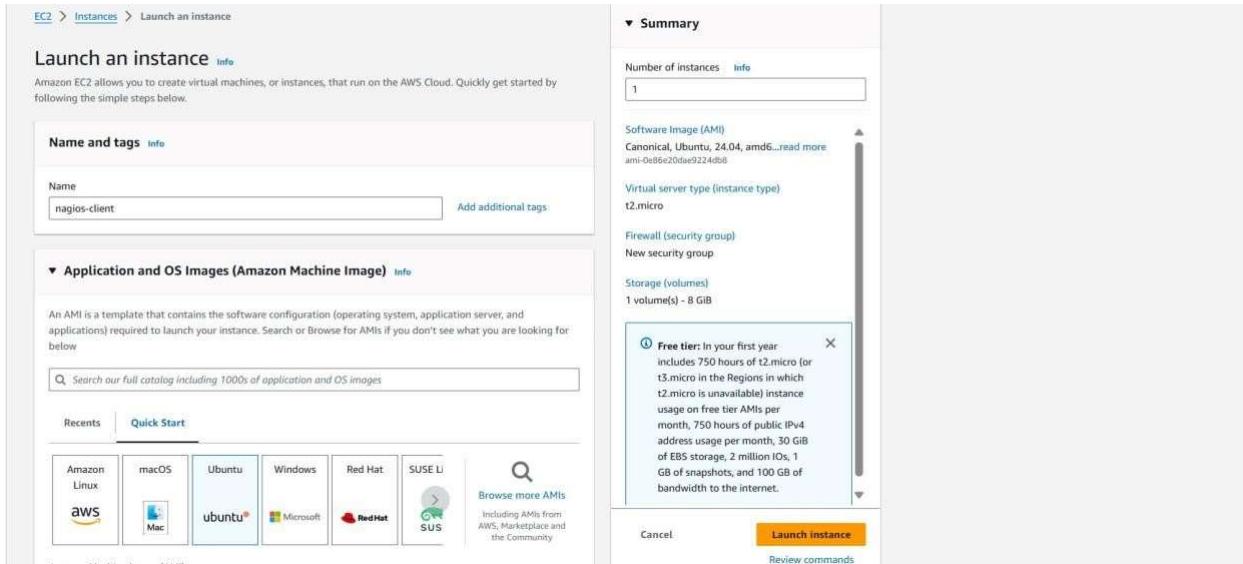
Step 1: To Confirm Nagios is running on the server side Perform the following command on your Amazon Linux Machine (Nagios-host). **sudo systemctl status nagios**

```
[ec2-user@ip-172-31-91-91 ~]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
   Active: active (running) since Sun 2024-09-29 16:18:08 UTC; 21min ago
     Docs: https://www.nagios.org/documentation
 Process: 1942 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Process: 1944 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 1946 (nagios)
   Tasks: 8 (limit: 1112)
  Memory: 7.7M
    CPU: 387ms
   CGroup: /system.slice/nagios.service
           ├─1946 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           ├─1947 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─1948 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─1949 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─1950 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─1956 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           ├─3088 /usr/local/nagios/libexec/check_ping -H 127.0.0.1 -w 3000.0,80% -c 5000.0,100% -p 5
           └─3089 /usr/bin/ping -n -U -w 30 -c 5 127.0.0.1

Sep 29 16:18:08 ip-172-31-91-91.ec2.internal systemd[1]: Starting nagios.service - Nagios Core 4.5.5...
Sep 29 16:18:08 ip-172-31-91-91.ec2.internal systemd[1]: Started nagios.service - Nagios Core 4.5.5
Sep 29 16:20:00 ip-172-31-91-91.ec2.internal nagios[1946]: SERVICE FLAPPING ALERT: localhost;HTTP:STARTED; Service appears to have started flapping (20.0%
Sep 29 16:20:00 ip-172-31-91-91.ec2.internal nagios[1946]: SERVICE ALERT: localhost;HTTP;CRITICAL;HARD;4;connect to address 127.0.0.1 and port 80: Connecti
Sep 29 16:22:30 ip-172-31-91-91.ec2.internal nagios[1946]: SERVICE NOTIFICATION: nagiosadmin;localhost;Swap Usage;CRITICAL;notify-service-by-email;SWAP CRITI
Sep 29 16:22:30 ip-172-31-91-91.ec2.internal nagios[1946]: wproc: NOTIFY job 2 from worker Core Worker 1948 is a non-check helper but exited with return co
Sep 29 16:22:30 ip-172-31-91-91.ec2.internal nagios[1946]: wproc: host=localhost; service=Swap Usage; contact=nagiosadmin
Sep 29 16:22:30 ip-172-31-91-91.ec2.internal nagios[1946]: wproc: early_timeout=0; exited_ok=1; wait_status=32512; error_code=0;
Sep 29 16:22:30 ip-172-31-91-91.ec2.internal nagios[1946]: wproc: stderr line 01: /bin/sh: line 1: /bin/mail: No such file or directory
Sep 29 16:22:30 ip-172-31-91-91.ec2.internal nagios[1946]: wproc: stderr line 02: /usr/bin/printf: write error: Broken pipe
lines 1-30/30 (END)
```

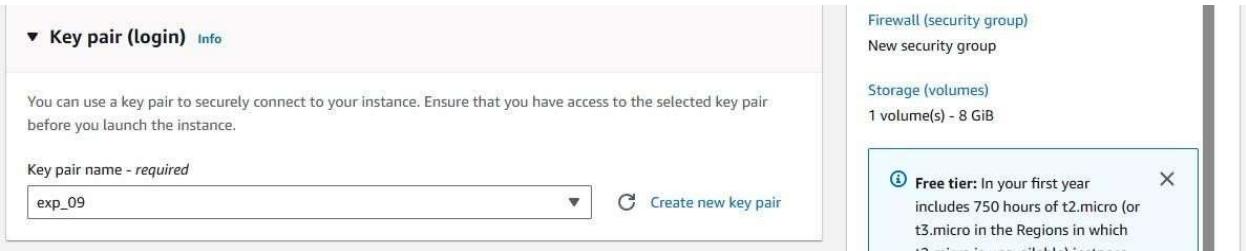
You can now proceed if you get the above message/output.

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

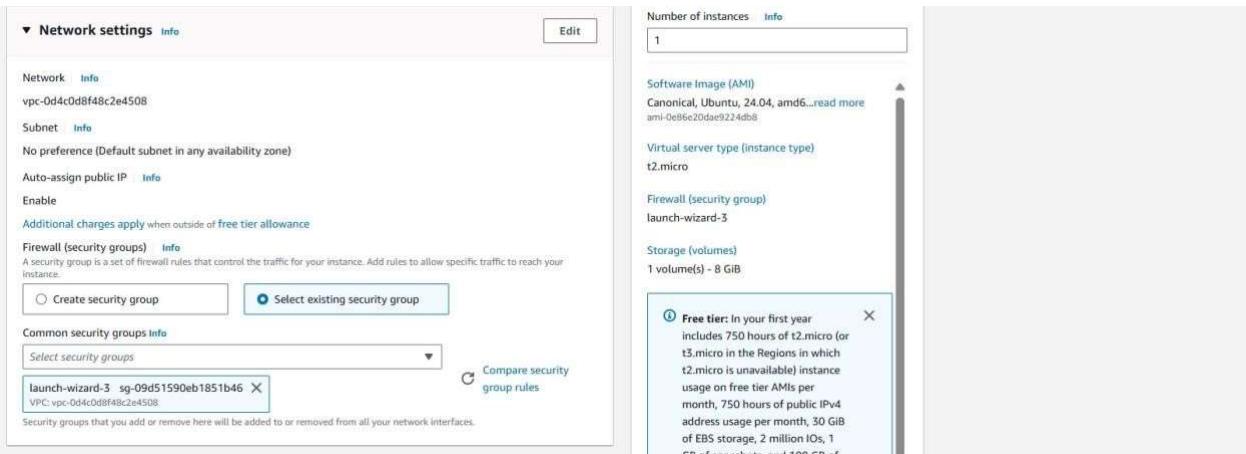


For Key pair : Click on create key and make key of type RSA with extension .pem . Key will be downloaded to your local machine.

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



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



Step 3: Now After creating the EC2 Instance click on connect and then copy the command which is given as example in the SSH Client section .

Now open the terminal in the folder where your key(RSA key with .pem) is located. and paste that copied command.

```
The authenticity of host 'ec2-44-206-245-149.compute-1.amazonaws.com (44.206.245.149)' can't be established.
ED25519 key fingerprint is SHA256:DT+AA+mkydh3kOJ2vEpw4ZsA6FL+LM4m1QSIddAHg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-206-245-149.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-92-146:~$ |
```

Now perform all the commands on the Nagios-host till step 10

Step 4: Now on the server Nagios-host run the following command.

ps -ef | grep nagios

```
[ec2-user@ip-172-31-91-91 ~]$ ps -ef | grep nagios
nagios 1946 1 0 16:18 ? 00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios 1947 1946 0 16:18 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 1948 1946 0 16:18 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 1949 1946 0 16:18 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 1950 1946 0 16:18 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 1956 1946 0 16:18 ? 00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
root 3099 3055 0 16:40 pts/0 00:00:00 sudo systemctl status nagios
root 3092 3099 0 16:40 pts/1 00:00:00 sudo systemctl status nagios
root 3093 3092 0 16:40 pts/1 00:00:00 systemctl status nagios
[ec2-user 3914 3890 0 16:59 pts/2 00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-91-91 ~]$ |
```

Step 5: Now Become root user and create root directories.

sudo su

**mkdir /usr/local/nagios/etc/objects/monitorhosts mkdir
/usr/local/nagios/etc/objects/monitorhosts/linuxhosts**

```
[ec2-user@ip-172-31-91-91 ~]$ sudo su
[root@ip-172-31-91-91 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-91-91 ec2-user]# |
```

Step 6: Copy the sample localhost.cfg to linuxhost.cfg by running the following command.(Below command should come in one line see screenshot below)

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

```
[root@ip-172-31-91-91 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
[root@ip-172-31-91-91 ec2-user]# |
```

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

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

Change hostname to **linuxserver**.

Change address to the **public IP** of your Linux client.

Set hostgroup_name to **linux-servers1**.

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

# Define a host for the local machine

define host {
    use            linux-server           ; Name of host template to use
                                         ; This host definition will inherit all variables that are defined
                                         ; in (or inherited by) the linux-server host template definition.

    host_name      linuxserver
    alias          localhost
    address        172.31.92.146
}

#####

# HOST GROUP DEFINITION
#
#####

# Define an optional hostgroup for Linux machines

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

Step 8: Now update the Nagios config file .Add the following line in the file. Line to add :

> nano /usr/local/nagios/etc/nagios.cfg

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

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

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

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

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

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

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

```
Things look okay - No serious problems were detected during the pre-flight check
[root@ip-172-31-91-91 ec2-user]# /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

Running pre-flight check on configuration data...

Checking objects...
    Checked 8 services.
    Checked 2 hosts.
    Checked 2 host groups.
    Checked 0 service groups.
    Checked 1 contacts.
    Checked 1 contact groups.
    Checked 24 commands.
    Checked 5 time periods.
    Checked 0 host escalations.
    Checked 0 service escalations.

Checking for circular paths...
    Checked 2 hosts
    Checked 0 service dependencies
    Checked 0 host dependencies
    Checked 5 timeperiods

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

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
```

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

service nagios restart

```
Things look okay - No serious problems were detected during the pre-flight check
[root@ip-172-31-91-91 ec2-user]# service nagios restart
Restarting nagios (via systemctl): [ OK ]
[root@ip-172-31-91-91 ec2-user]# |
```

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

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

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

Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [380 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [82.9 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4566 B]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [272 kB]
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [115 kB]
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [8632 B]
Get:14 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [10.3 kB]
Get:15 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [353 kB]
Get:16 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [68.1 kB]
Get:17 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 c-n-f Metadata [428 B]
Get:18 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [10.9 kB]
Get:19 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [2808 B]
Get:20 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 kB]
Get:21 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [344 B]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
```

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

No containers need to be restarted.

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

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

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

# ALLOWED HOST ADDRESSES
# This is an optional comma-delimited list of IP address or hostnames
# that are allowed to talk to the NRPE daemon. Network addresses with a bit mask
# (i.e. 192.168.1.0/24) are also supported. Hostname wildcards are not currently
# supported.
#
# Note: The daemon only does rudimentary checking of the client's IP
# address. I would highly recommend adding entries in your /etc/hosts.allow
# file to allow only the specified host to connect to the port
# you are running this daemon on.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd
allowed_hosts=127.0.0.1,:1,34.207.68.187

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

Step 13: Now restart the NRPE server by this command.

sudo systemctl restart nagios-nrpe-server

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

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

```
[ec2-user@ip-172-31-91-91 ~]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
     Active: active (running) since Sun 2024-09-29 17:20:07 UTC; 12min ago
       Docs: https://www.nagios.org/documentation
   Process: 4761 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
  Process: 4762 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 4763 (nagios)
   Tasks: 6 (limit: 1112)
  Memory: 4.1M
    CPU: 234ms
   CGroup: /system.slice/nagios.service
           └─4763 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
               ├─4764 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
               ├─4765 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
               ├─4766 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
               ├─4767 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
               └─4768 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 29 17:20:07 ip-172-31-91-91.ec2.internal nagios[4763]: Warning: Duplicate definition found for service 'Current Users' on host 'localhost' (config file
Sep 29 17:20:07 ip-172-31-91-91.ec2.internal nagios[4763]: Warning: Duplicate definition found for service 'Root Partition' on host 'localhost' (config fil
Sep 29 17:20:07 ip-172-31-91-91.ec2.internal nagios[4763]: Warning: Duplicate definition found for service 'PING' on host 'localhost' (config file '/usr/lo
Sep 29 17:20:07 ip-172-31-91-91.ec2.internal nagios[4763]: Successfully launched command file worker with pid 4768
Sep 29 17:22:30 ip-172-31-91-91.ec2.internal nagios[4763]: SERVICE NOTIFICATION: nagiosadmin;localhost;Swap Usage;CRITICAL;notify-service-by-email;SWAP CRI
Sep 29 17:22:30 ip-172-31-91-91.ec2.internal nagios[4763]: wproc: NOTIFY job 1 from worker Core Worker 4766 is a non-check helper but exited with return co
Sep 29 17:22:30 ip-172-31-91-91.ec2.internal nagios[4763]: wproc: host=localhost; service=Swap Usage; contact=nagiosadmin
Sep 29 17:22:30 ip-172-31-91-91.ec2.internal nagios[4763]: wproc: early_timeout=0; exited_ok=1; wait_status=32512; error_code=0;
Sep 29 17:22:30 ip-172-31-91-91.ec2.internal nagios[4763]: wproc: stderr line 01: /bin/sh: line 1: /bin/mail: No such file or directory
Sep 29 17:22:30 ip-172-31-91-91.ec2.internal nagios[4763]: wproc: stderr line 02: /usr/bin/printf: write error: Broken pipe
[lines 1-28/28 (END)]
```

sudo systemctl status httpd

```
[ec2-user@ip-172-31-91-91 ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
  Drop-In: /usr/lib/systemd/system/httpd.service.d
            └─php-fpm.conf
  Active: inactive (dead)
    Docs: man:httpd.service(8)
[ec2-user@ip-172-31-91-91 ~]$ |
```

sudo systemctl start httpd sudo systemctl enable httpd

```
[ec2-user@ip-172-31-91-91 ~]$ sudo systemctl start httpd
[ec2-user@ip-172-31-91-91 ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[ec2-user@ip-172-31-91-91 ~]$ |
```

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

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

The screenshot shows the Nagios Core dashboard. On the left is a vertical navigation menu with sections like General, Current Status, Reports, and System. The main area has several boxes: 'Get Started' with monitoring tips, 'Latest News' (empty), 'Don't Miss...' (empty), and 'Quick Links' with links to Nagios Library, Labs, Exchange, Support, and the official website. At the bottom, there's copyright information and a note about the license.

Now Click on Hosts from left side panel

The screenshot shows the 'Hosts' section of the Nagios Core interface. It displays a table of host status details for two hosts: 'linusserver' and 'localhost'. The table includes columns for Host, Status, Last Check, Duration, and Status Information.

Host	Status	Last Check	Duration	Status Information
linusserver	UP	09-29-2024 17:40:07	0d 0h 22m 03s	PING OK - Packet loss = 0%, RTA = 0.56 ms
localhost	UP	09-29-2024 17:40:00	0d 9h 37m 43s	PING OK - Packet loss = 0%, RTA = 0.04 ms

Conclusion:

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

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

STEP1:Go on your AWS console account and search for 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 where the function name is set to 'lambdaexp11', runtime is 'Python 3.12', architecture is 'x86_64', and permissions are managed via a 'Change default execution role' link.

Create function Info

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.

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.

Successfully created the function lamdaexp11. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Lambda > Functions > lamdaexp11

lamdaexp11

Function overview [Info](#)

[Throttle](#) [Copy ARN](#) [Actions ▾](#)

[Export to Application Composer](#) [Download ▾](#)

Diagram [Template](#)

lamdaexp11

Layers (0)

+ Add trigger [+ Add destination](#)

Description
-

Last modified
16 seconds ago

Function ARN
arn:aws:lambda:eu-north-1:026090558619:function:lamdaexp11

Code source [Info](#)

[Upload from ▾](#)

File **Edit** **Find** **View** **Go** **Tools** **Window** **Test** [Deploy](#) [⚙️](#)

lambda_function Environment Vari
lambda_function.py

```

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     }
9

```

Code [Test](#) [Monitor](#) [Configuration](#) [Aliases](#) [Versions](#)

General configuration [Info](#) [Edit](#)

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

General configuration

- Triggers
- Permissions
- Destinations
- Function URL
- Environment variables
- Tags
- VPC

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

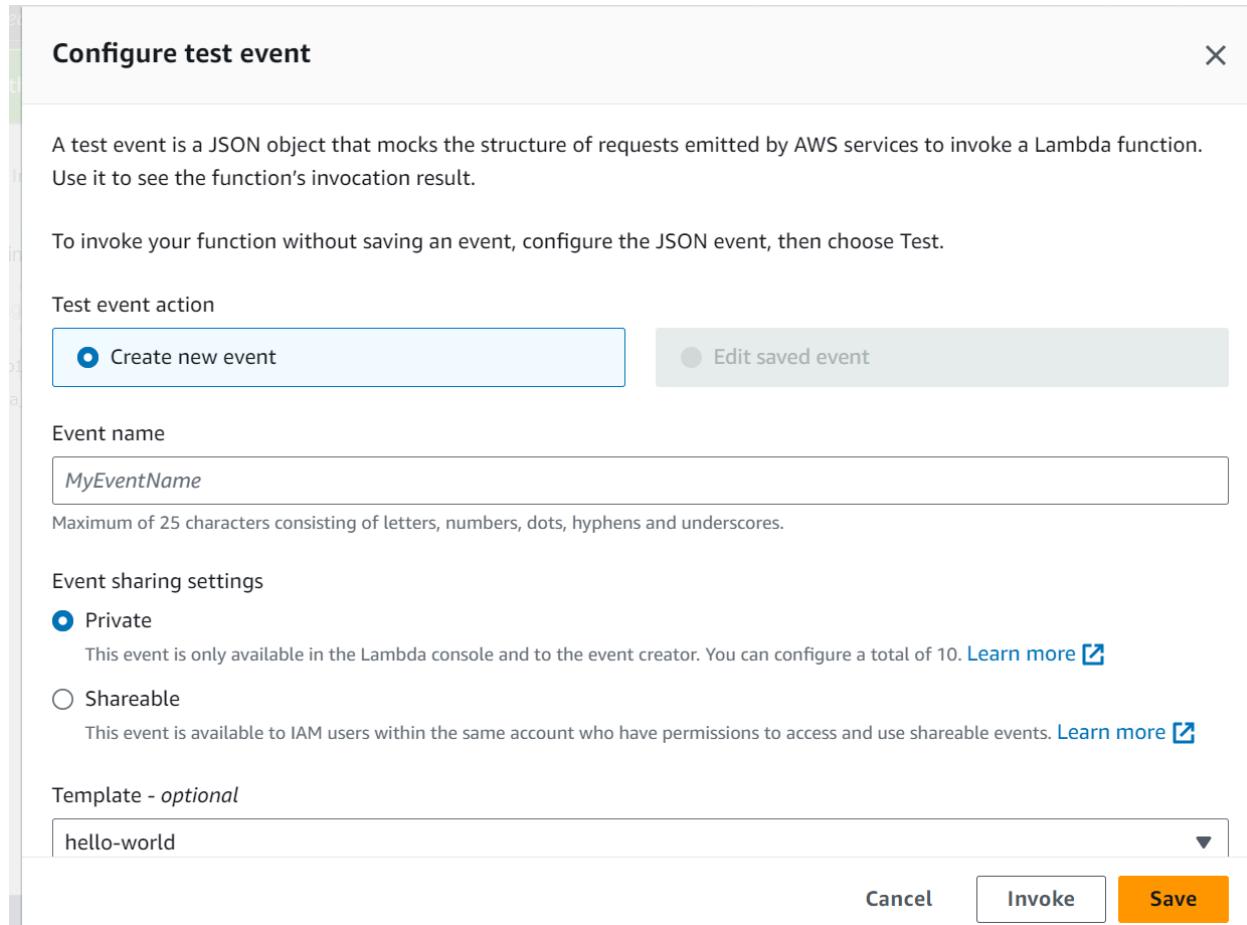
Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

service-role/lamdaexp11-role-vj5j9g95

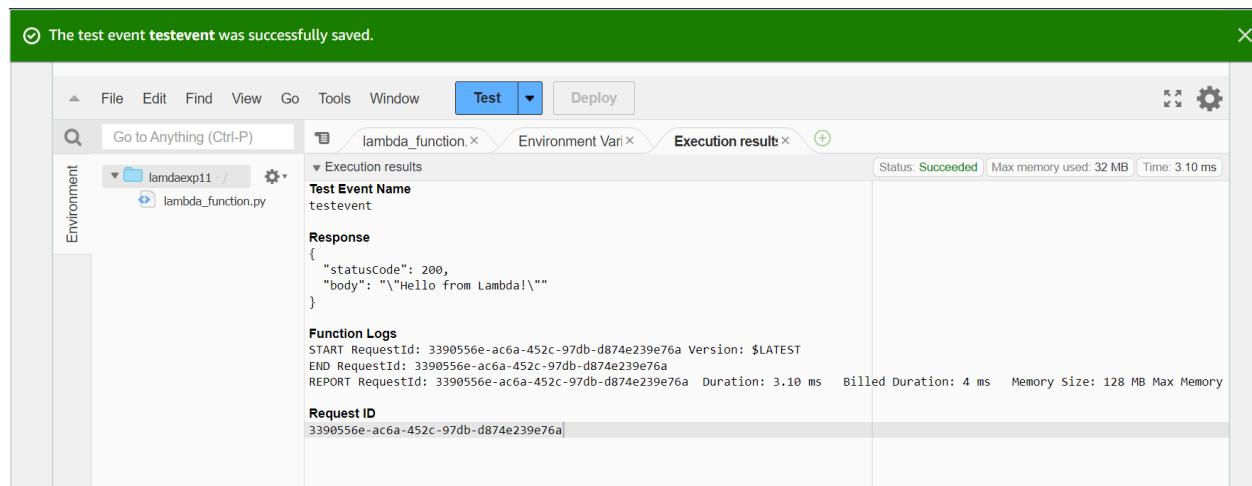
[View the lamdaexp11-role-vj5j9g95 role](#) on the IAM console.

Cancel **Save**

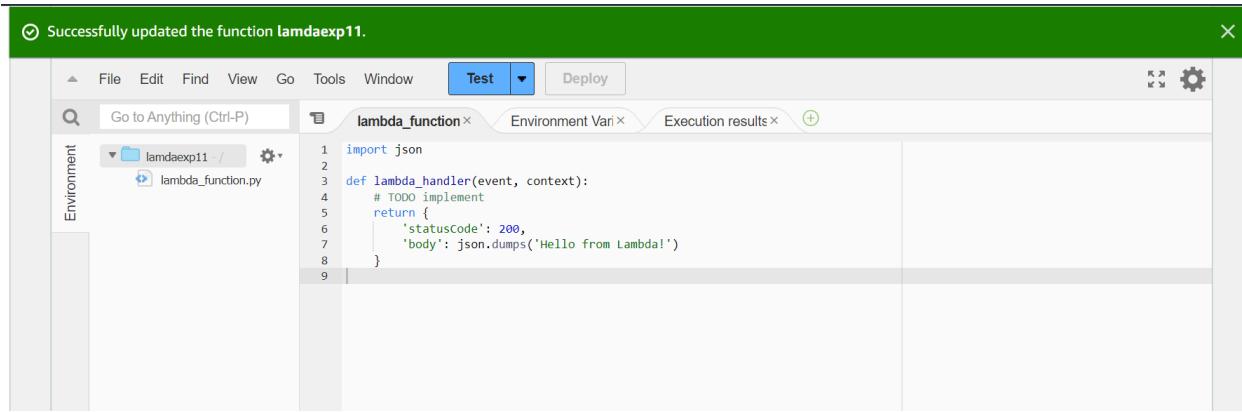
STEP 5: Go on configure test event click on “create new event” edit the event sharing accordingly and select hello world template for template option and then save it.



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 main window has a toolbar with "File", "Edit", "Find", "View", "Go", "Tools", "Window", "Test" (which is currently selected), and "Deploy". To the right of the toolbar are three small icons: a gear, a magnifying glass, and a plus sign. The left sidebar is titled "Environment" and contains a search bar labeled "Go to Anything (Ctrl-P)" and a folder icon labeled "lambdaexp11 - /". The main content area is titled "lambda_function" and shows 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     }
9
```

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

Advance Devops-12

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

Theory:

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

Workflow:

1. Create an S3 Bucket:

- First, create an S3 bucket that will store the objects. This bucket will act as the trigger source for the Lambda function.

2. Create the Lambda Function:

- Set up a new Lambda function using AWS Lambda’s console. You can choose a runtime environment like Python, Node.js, or Java.
- Write code that logs a message like “An Image has been added” when triggered.

3. Set Up Permissions:

- Ensure that the Lambda function has the necessary permissions to access S3. You can do this by attaching an IAM role with policies that allow reading from the bucket and writing logs to CloudWatch.

4. Configure S3 Trigger:

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

5. Test the Setup:

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

Outcomes:

Step 1:create a s3 bucket

Create bucket [Info](#)

Buckets are containers for data stored in S3.

General configuration

AWS Region
US East (N. Virginia) us-east-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](#)
`exp12`

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

Copy settings from existing bucket - optional

Step 2:

Create a function

And select Python 3 as Routine

Successfully created bucket "exp-no-12"

To upload files and folders, or to configure additional bucket settings, choose [View details](#).

View details [X](#)

Amazon S3 > Buckets

▶ Account snapshot - updated every 24 hours [All AWS Regions](#)

Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

[View Storage Lens dashboard](#)

Create function [Info](#)

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.

Basic information

Function name

Enter a name that describes the purpose of your function.

`nikita-12`

Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

Runtime [Info](#)

Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Python 3.9

Step 3: Add a trigger

Add trigger

Step 4:

Test it

The test event nikita-test was successfully saved.

Code source [Info](#)

File Edit Find View Go Tools Window Test Deploy

Execution results

Test Event Name nikita-test

Status: Succeeded | Max memory used: 30 MB | Time: 1.56 ms

Response

```
{
  "statusCode": 200,
  "body": "\"Hello from Lambda!\""
}
```

Function Logs

```
START RequestId: 83864db6-5abe-404f-935f-1088bf50dba6 Version: $LATEST
END RequestId: 83864db6-5abe-404f-935f-1088bf50dba6
REPORT RequestId: 83864db6-5abe-404f-935f-1088bf50dba6 Duration: 1.56 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 30 MB
Request ID 83864db6-5abe-404f-935f-1088bf50dba6
```

The screenshot shows the AWS S3 console. At the top, a green banner indicates a successful upload: "Upload succeeded" with "View details below." Below this, a summary table shows the destination as "s3://nikita-visit". The status column shows "Succeeded" for one file (1.6 MB) and "Failed" for zero files (0 B). Under the "Files and folders" tab, a table lists "demo.jpg" as 1.6 MB, type "image/jpeg", and status "Succeeded".

Step 5:**Upload a image****It should be in jpg format**

[CloudWatch](#) > [Log groups](#) > [/aws/lambda/nikita-12](#) > 2024/10/11/[[\\$LATEST](#)]8148efef2713489a960d35b23bad25b1

The screenshot shows the AWS CloudWatch Logs interface for the log group "/aws/lambda/nikita-12". The log events table has columns for "Timestamp" and "Message". It displays four log entries: INIT_START, START, END, and REPORT. The REPORT entry includes detailed metrics like Duration and Billed Duration. A message at the bottom indicates "Auto retry paused".

Step 6:**You can see the image you uploaded with the time**

The screenshot shows the AWS CloudWatch Log Events interface. The URL in the address bar is [CloudWatch > Log groups > /aws/lambda/nikita-12 > 2024/10/11/\[LATEST\]8148fefef2713489a960d35b23bad25b1](#). The main heading is "Log events". Below it, a sub-header says "You can use the filter bar below to search for and match terms, phrases, or values in your log events. [Learn more about filter patterns](#)". The filter bar includes a search input ("Filter events - press enter to search"), time range ("1m", "1h", "Local timezone"), "Display" dropdown, and a gear icon. The table has columns "Timestamp" and "Message". The first row of the table says "No older events at this moment. [Retry](#)". The second row shows a log entry for 2024-10-11T10:13:51.831+05:30: "INIT_START Runtime Version: python:3.9.v62 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:4b9...". The third row shows a log entry for 2024-10-11T10:13:51.910+05:30: "START RequestId: 83864db6-5abe-404f-935f-1088bf50dba6 Version: \$LATEST". The fourth row shows a log entry for 2024-10-11T10:13:51.914+05:30: "END RequestId: 83864db6-5abe-404f-935f-1088bf50dba6". The fifth row shows a log entry for 2024-10-11T10:13:51.914+05:30: "REPORT RequestId: 83864db6-5abe-404f-935f-1088bf50dba6 Duration: 1.56 ms Billed Duration: 2 ms Memory...". The last row of the table says "No newer events at this moment. Auto retry paused. [Resume](#)".

Conclusion:

Integrating AWS Lambda with S3 allows for real-time, automated processing of events such as file uploads. In this example, a Lambda function is configured to log a message whenever an image is added to a specific S3 bucket. This setup demonstrates the power and flexibility of serverless computing by automating tasks without requiring manual intervention or server management. By leveraging AWS Lambda, developers can efficiently handle event-driven workflows, reduce operational overhead, and quickly deploy scalable solutions that respond to specific actions within cloud environments.