

**Name: Shubham Jha**

**19**

**Div: D15C**

**Static Hosting:**

**Roll\_no:**

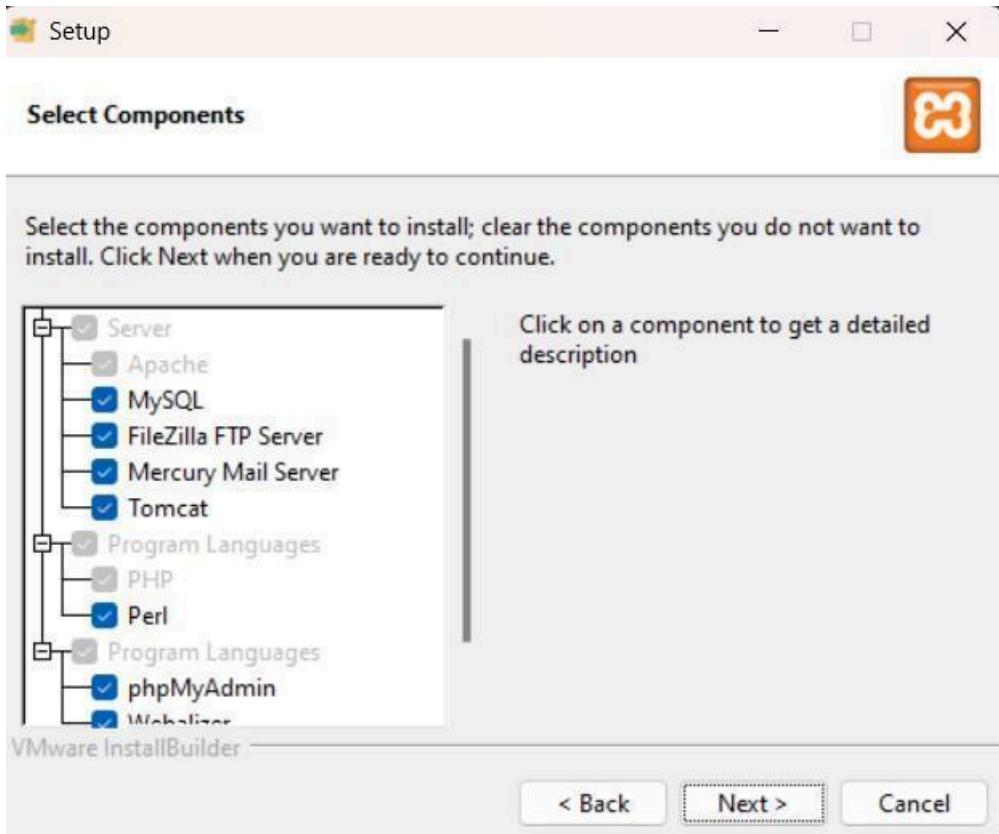
### **1) On local server (XAMPP)**

**Step 1: Install XAMPP from <https://www.apachefriends.org/>**

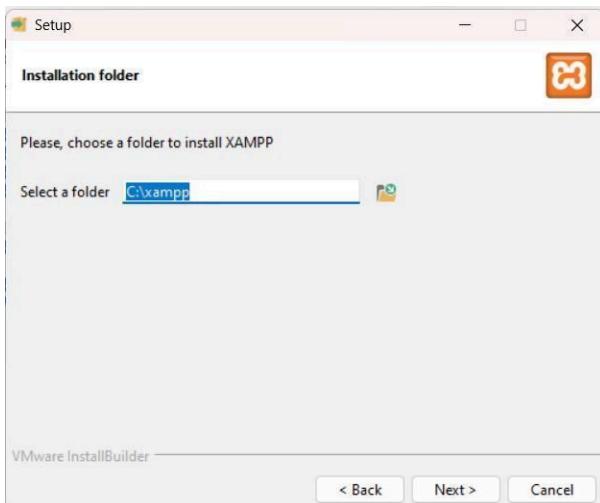
- 1) Select your OS. It will automatically start downloading.



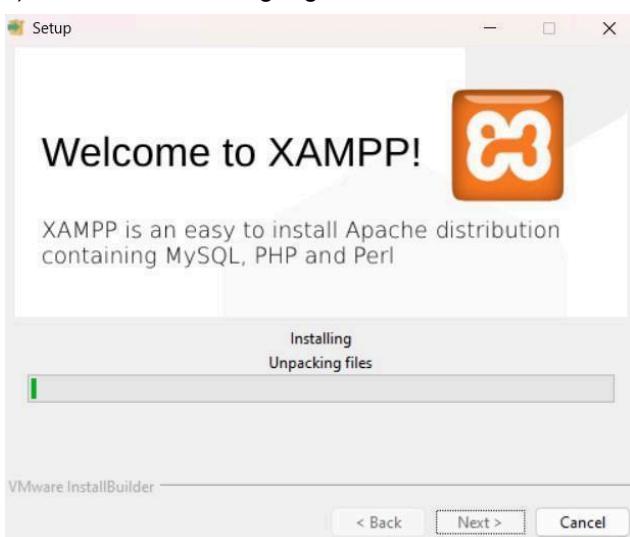
- 2) Open the setup file. Select all the required components and click next



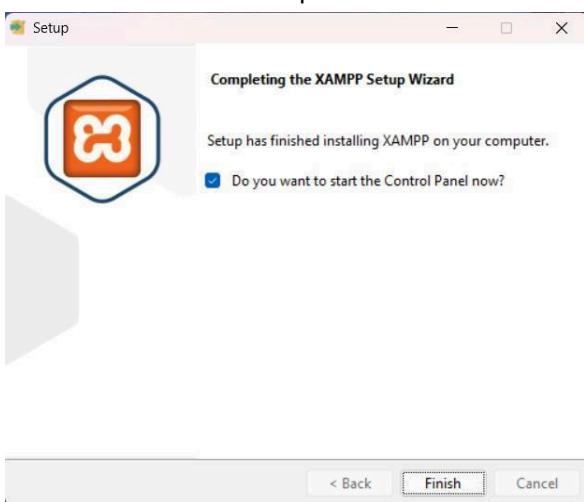
- 3) Choose the folder to install XAMPP in. Make sure the folder is empty. Click next



- 4) Select the language, click next. XAMPP starts to install



- 5) The installation is complete. Click Finish



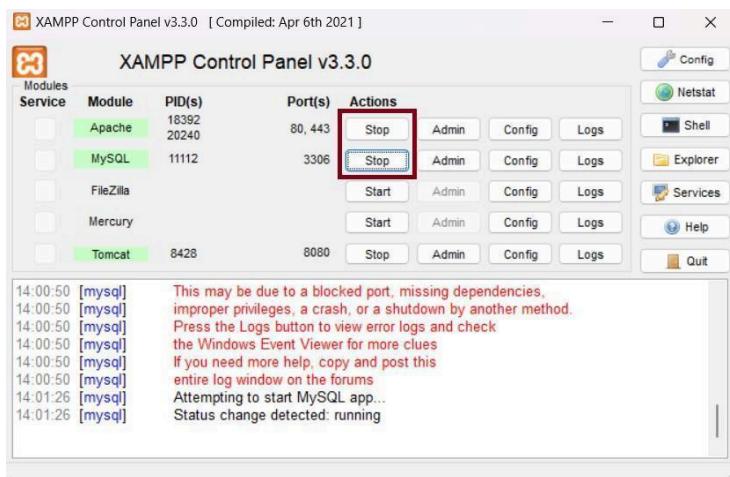
**Step 2:** Setup a file that is to be hosted on the server. Make sure the file has extension .php

test1	06-08-2024 22:48	PHP Source File	1 KB
-------	------------------	-----------------	------

**Step 3:** Go to the directory where XAMPP was installed. Go to **htdocs** folder. Place your folder in this directory.

Name	Date modified	Type	Size
dashboard	06-08-2024 20:42	File folder	
img	06-08-2024 20:42	File folder	
webalizer	06-08-2024 20:42	File folder	
xampp	06-08-2024 22:44	File folder	
applications	15-06-2022 21:37	Chrome HTML Do...	4 KB
bitnami	15-06-2022 21:37	CSS Source File	1 KB
favicon.ico	16-07-2015 21:02	ICO File	31 KB
index	16-07-2015 21:02	PHP Source File	1 KB
test1	06-08-2024 22:48	PHP Source File	1 KB
text	06-08-2024 22:23	PHP Source File	1 KB

**Step 4:** Open XAMPP Control Panel, start the Apache service (Required) and mySQL service (if needed)



**Step 5:** Open your web browser. Type localhost/YOUR\_FILENAME.php. This will open your website on your browser.

The screenshot shows a web browser window with the URL `localhost/portfoliowebsite1`. The page has a dark header with a user icon and the name "Shubham Jha". Below the header is a banner featuring a circular profile picture of a man (Shubham Jha) and the text "Hello! This is Shubham Jha". Under the banner is a bio section with the heading "About me:" followed by a short paragraph: "I am passionate about problem solving and teaching. Love to create new things and learn new things. Love to travel, Train and read." At the bottom of the page are three social media links: "Mailme", "LinkedIn", and "Github".

### Skills:

#### DSA

- Strong grasp on Algorithms and problem solving skills
- Languages : Java, C, Cpp

#### Frontend Development:

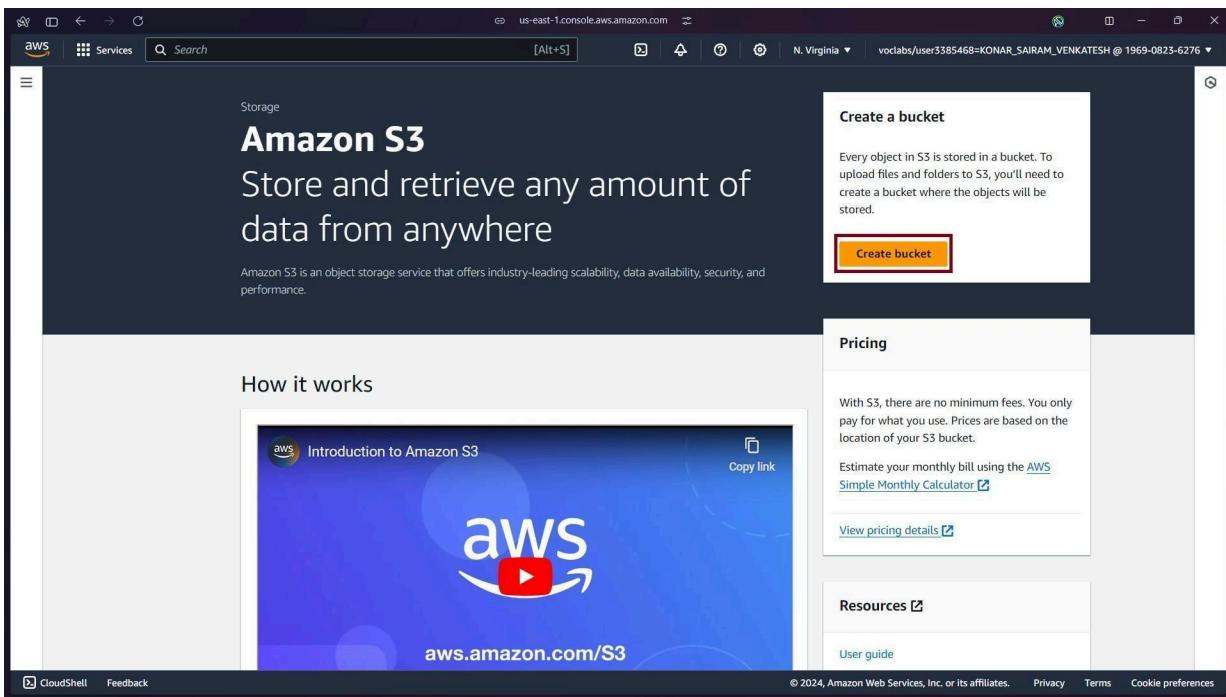
- React
- JavaScript

## 2) AWS S3

**Step 1:** Login to your AWS account. Go to services and open **S3**.

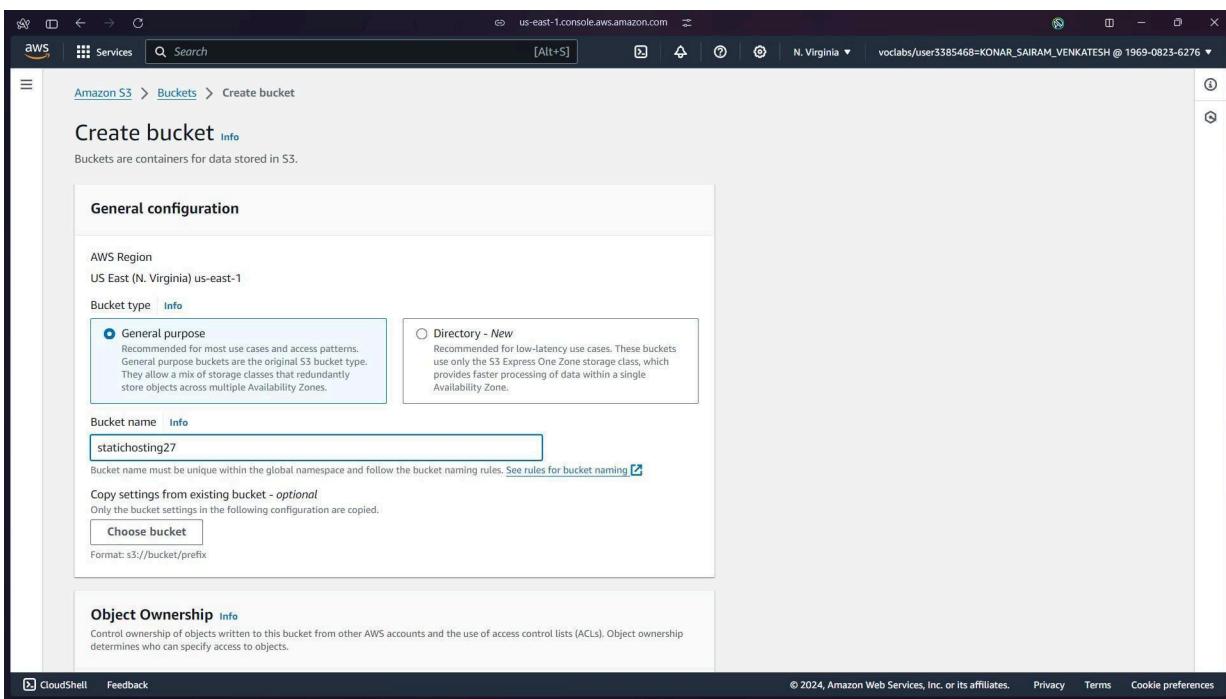
The screenshot shows the AWS Services console. On the left is a sidebar with "Recently visited" and "Favorites" sections, and a list of "All services" categorized into groups like Analytics, Application Integration, Blockchain, etc. On the right is a "Recently visited" panel containing links to various AWS services: S3 (Scalable Storage in the Cloud), IAM (Manage access to AWS resources), IAM Identity Center (Manage workforce user access to multiple AWS accounts and cloud applications), Resource Access Manager (Share AWS resources with other accounts or AWS Organizations), Cloud9 (A Cloud IDE for Writing, Running, and Debugging Code), and EC2 (Virtual Servers in the Cloud). The "S3" link is highlighted.

## Step 2: Click on Create Bucket



The screenshot shows the AWS S3 landing page. On the right side, there is a prominent call-to-action box titled "Create a bucket". Below it, a text explains that every object in S3 is stored in a bucket and provides instructions to upload files and folders. A large orange "Create bucket" button is centered in this box. To the left of this box, there's a section titled "How it works" featuring a video thumbnail for "Introduction to Amazon S3". To the right of the main content area, there are sections for "Pricing" and "Resources". The bottom of the page includes standard AWS navigation links like CloudShell, Feedback, and copyright information.

## Step 3: Give a name to your bucket, keeping other options default, scroll down and click on Create Bucket



The screenshot shows the "Create bucket" configuration page. The "General configuration" tab is active. It includes fields for "Bucket name" (set to "statichosting27") and "AWS Region" (set to "US East (N. Virginia) us-east-1"). There are two radio button options for "Bucket type": "General purpose" (selected) and "Directory - New". Below these, there are sections for "Copy settings from existing bucket - optional" and "Object Ownership". The "Object Ownership" section notes that object ownership determines who can specify access to objects. At the bottom of the page, there are standard AWS navigation links.

#### Step 4: Click on the name of your bucket and goto Properties

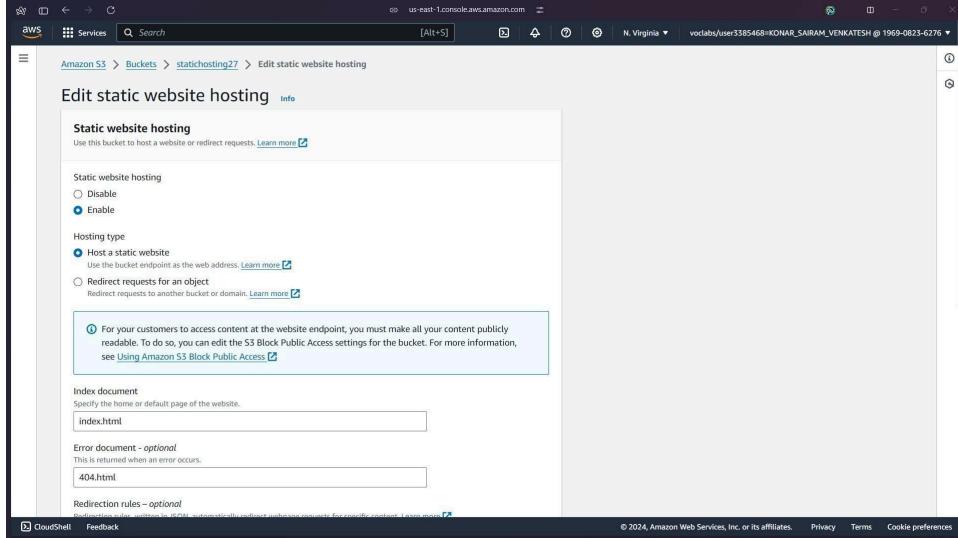
The screenshot shows the AWS S3 Buckets page. At the top, there's an account snapshot and a search bar. Below that, tabs for 'General purpose buckets' and 'Directory buckets' are visible. Under 'General purpose buckets', there's a table with one row. The first column contains a checkbox and the bucket name 'statichosting27'. The second column shows the AWS Region as 'US East (N. Virginia) us-east-1'. The third column shows the IAM Access Analyzer status as 'View analyzer for us-east-1'. The fourth column shows the creation date as 'August 4, 2024, 15:30:03 (UTC+05:30)'. At the bottom of the table, there are buttons for 'Create bucket' and other actions.

The screenshot shows the AWS S3 Bucket Properties page for 'statichosting27'. The top navigation bar includes 'Objects', 'Properties' (which is selected and highlighted with a red box), 'Permissions', 'Metrics', 'Management', and 'Access Points'. Below the navigation, there's a toolbar with buttons for 'Actions', 'Create folder', and 'Upload'. A search bar is present. The main area shows a table with one row under 'Objects (0)'. The columns are 'Name', 'Type', 'Last modified', 'Size', and 'Storage class'. A message says 'No objects' and 'You don't have any objects in this bucket.' At the bottom, there's a 'Upload' button.

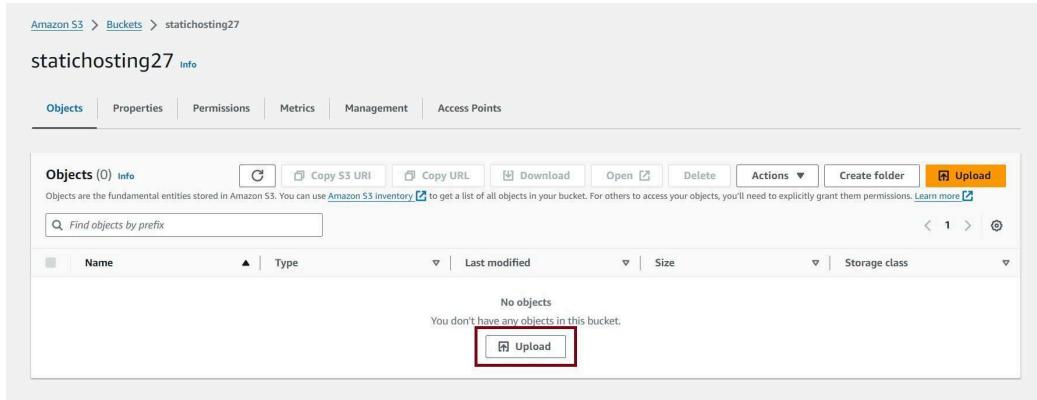
#### Step 5: Scroll down till you find Static website hosting, click on edit

The screenshot shows the 'Static website hosting' section of the AWS S3 Bucket Properties page. It includes a note: 'Use this bucket to host a website or direct requests. Learn more.' Below that, it says 'Static website hosting' and 'Disabled'. An 'Edit' button is located at the bottom right of this section, which is highlighted with a red box.

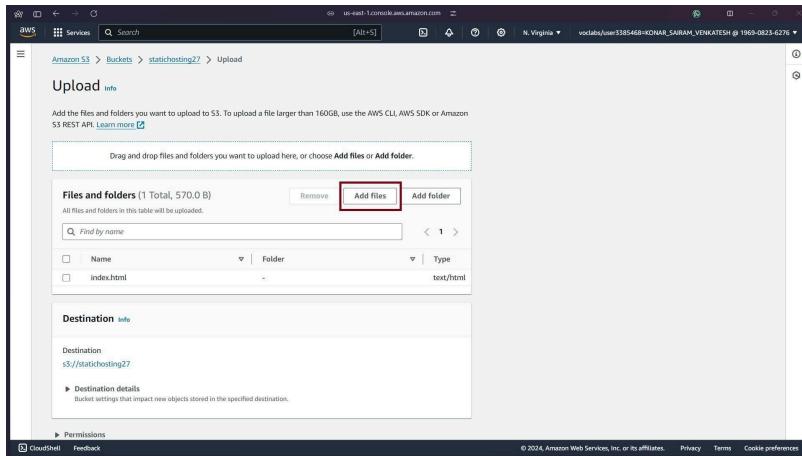
**Step 6:** Enable static website hosting, in Index document, write the name of your document and in error document, give name as 404.html. Save your changes.



**Step 7:** Go to Objects tab and click on upload file.



**Step 8:** Click on Add files. Add all the files you want to upload. Then scroll down and click on Upload



**Step 9:** This will take you to the Objects screen. Switch to Properties, scroll down to Static web hosting. There you would find the link (Bucket website endpoint) to your website.

The screenshot shows the 'Static website hosting' section of the AWS S3 Bucket Properties page. It includes fields for 'Hosting type' (set to 'Bucket hosting') and a 'Bucket website endpoint' (set to '<http://statichosting27.s3-website-us-east-1.amazonaws.com>'). A red box highlights the 'Bucket website endpoint' field and its value.

**Step 10:** Open the link. It will show a 403 forbidden error screen as the contents of the bucket are not available for the public users. To change this, go to Permissions tab, go to Block public access and click on edit

The screenshot shows a 403 Forbidden error page. The URL in the address bar is 'statichosting27.s3-website-us-east-1.amazonaws.com'. The main content area displays the error message '403 Forbidden' and a list of error details.

**Code: AccessDenied**

**Message: Access Denied**

**RequestId: 8TQ4EGP4TK06MVPB**

**HostId: hF+ToadQUoCuDM8H+iFrSXdA28TGp+xikYbjb4CICS/t+3it4ihA/tvgA1Xr1xo+JL5AhkT6hJs=**

**An Error Occurred While Attempting to Retrieve a Custom Error Document**

**Code: AccessDenied**

**Message: Access Denied**

**Step 11:** Uncheck the Block all public access checkbox and click on save changes

The screenshot shows the 'Edit Block public access (bucket settings)' page. At the top, there is a note about turning on 'Block all public access'. Below it is a list of four checkboxes under 'Block public access (bucket settings)'. The first checkbox, 'Block all public access', is highlighted with a red border. The other three checkboxes are not selected.

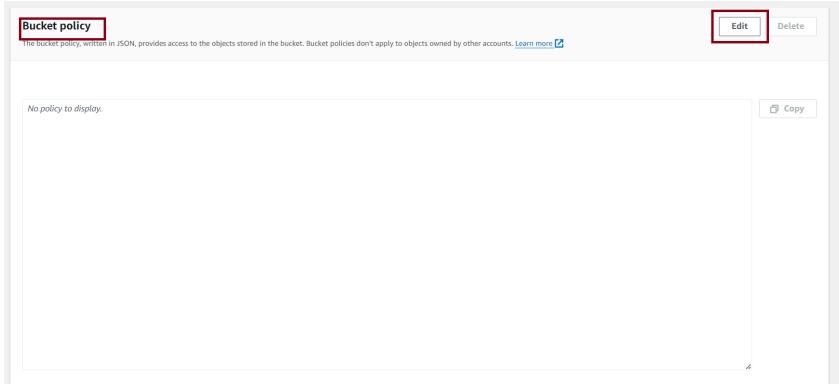
**Block all public access**

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

- Block public access to buckets and objects granted through new access control lists (ACLs)**  
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLs)**  
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies**  
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**  
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

At the bottom right are 'Cancel' and 'Save changes' buttons.

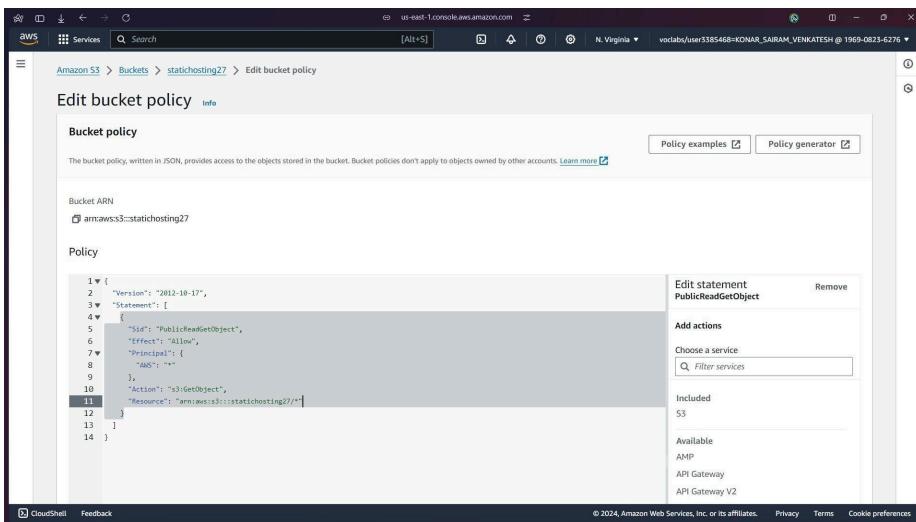
## Step 12: Scroll down to bucket policy and click edit



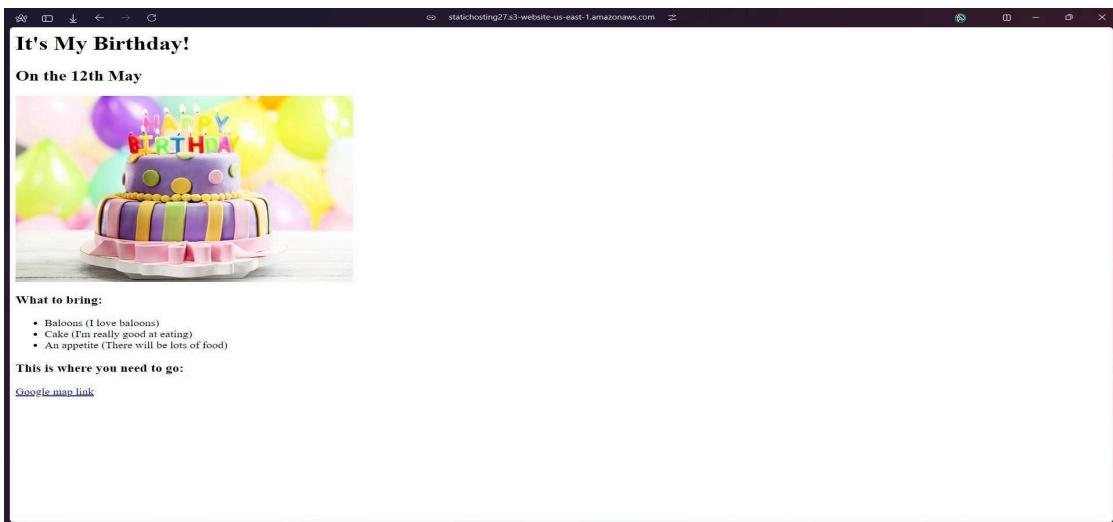
## Step 13:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "PublicReadGetObject",  
      "Effect": "Allow",  
      "Principal": {  
        "AWS": "*"  
      },  
      "Action": "s3:GetObject",  
      "Resource": "arn:aws:s3:::YOUR-BUCKET-NAME-HERE/*"  
    }  
  ]  
}
```

Paste this code snippet in the policy textarea. Replace YOUR-BUCKET-NAME-HERE with the name you have given to your bucket. Save the changes.



**Step 14:** Now reload the website. You can see your website



## Practical No 2 : Elastic Beanstalk

A screenshot of the AWS Elastic Beanstalk console showing the environment overview for "Sampel-env". The top navigation bar shows "Elastic Beanstalk &gt; Environments &gt; Sampel-env". The main area is divided into two sections: "Environment overview" and "Platform". The "Environment overview" section includes fields for Health (with a warning icon), Environment ID (e-u7kfdezi3r), Domain (kshitij.us-east-1.elasticbeanstalk.com), Application name (sampel), and Events tab. The "Platform" section shows PHP 8.3 running on 64bit Amazon Linux 2023/4.3.1, Running version (empty), Platform state (Supported), and a "Change version" button. Below these sections is an "Events" table with one entry: "Service role "arn:aws:siam::996474913977:role/EMR\_EC2\_DefaultRole" is missing permissions required to check for". The table has columns for Time (August 9, 2024 21:25:22 (UTC+5:30)), Type (WARN), and Details.

# Congratulations!

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

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

## Hello this is my first deployment D15C

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

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

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

1. Create 3 EC-2 instances with all running on Amazon Linux as OS with inbound SSH allowed and the proper key

The screenshot shows the 'Launch an instance' wizard in the AWS Management Console. The current step is 'Name and tags'. A 'Name' field contains the value 'master'. An 'Add additional tags' button is visible. Below this, a section titled 'Application and OS Images (Amazon Machine Image)' is expanded, showing a note about AMIs.

EC2 > Instances > Launch an instance

### Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

**Name and tags Info**

Name  
master Add additional tags

**Application and OS Images (Amazon Machine Image) Info**

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for

To efficiently run a Kubernetes cluster, select an instance type of at least t2.medium as Kubernetes recommends at least 2 vCPU to run smoothly.

The screenshot shows the AWS Lambda function configuration page. It includes sections for:

- Environment variables:** AWS\_LAMBDA\_FUNCTION\_NAME=hello-world, AWS\_LAMBDA\_FUNCTION\_MEMORY\_SIZE=128, AWS\_LAMBDA\_FUNCTION\_TIMEOUT=3, AWS\_LAMBDA\_HANDLER=index.handler, AWS\_LAMBDA\_SOURCE\_CODE\_BUCKET=aws-samples, AWS\_LAMBDA\_SOURCE\_CODE\_KEY=serverless/hello-world.zip
- Triggers:** CloudWatch Logs (CloudWatchLogsLogGroupArn), CloudWatch Metrics (CloudWatchMetricsNamespace), CloudWatch Metrics (CloudWatchMetricsMetricName)
- Logs:** CloudWatch Logs (CloudWatchLogsLogGroupArn) and CloudWatch Metrics (CloudWatchMetricsMetricName)

In this way, create 3 instances namely master, worker-1 and worker-2.

Instances (1/5) <a href="#">Info</a>								
		Last updated less than a minute ago	Actions	Launch instances				
Name		Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
<input type="checkbox"/>	invictus	i-01fcdb47bb4741b10	<span>Running</span>	t2.micro	<span>2/2 checks passed</span>		us-east-1d	ec2-54-92-
<input type="checkbox"/>	worker2	i-03f5af0634e5cb13b	<span>Running</span>	t2.micro	<span>Initializing</span>		us-east-1b	ec2-3-87-2
<input type="checkbox"/>	worker1	i-0c1f3a8de38c54ce2	<span>Running</span>	t2.micro	<span>Initializing</span>		us-east-1b	ec2-18-209
<input checked="" type="checkbox"/>	master	i-052d151ff7f7212807	<span>Running</span>	t2.micro	<span>2/2 checks passed</span>		us-east-1b	ec2-44-211
<input type="checkbox"/>	aws-cloud9-Cl...	i-05341cdc66d13354b	<span>Running</span>	t2.micro	<span>2/2 checks passed</span>		us-east-1a	ec2-107-23

2. SSH into all 3 machines each in separate terminal
- a. You can do it through the aws console directly

```
C:\Users\Avan>ssh -i "C:\Users\Avan\Downloads\kub1.pem" ec2-user@3.85.237.93
The authenticity of host '3.85.237.93 (3.85.237.93)' can't be established.
ED25519 key fingerprint is SHA256:Nz2iC26abFyhATf/8i4F0IgWmDoxTXBbzY9/NkMwYyM.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.85.237.93' (ED25519) to the list of known hosts.

      _#
      ~\_ #####_      Amazon Linux 2023
      ~~ \_#####\
      ~~   \###|
      ~~     \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023
      ~~       V~' '-->
      ~~~
      ~~..  /
      ~~/_ -/
      _/m/'

Last login: Thu Sep 12 13:11:49 2024 from 18.206.107.29
```

Or

- b. Locate your key from the Downloads folder and open it in cmd and paste this command  
**ssh -i <-your-key->.pem ec2-user<ip-address of instance>**

```
C:\Users\Avan\Downloads>ssh -i C:\Users\Avan\Downloads\kub1.pem ec2-user@54.85.79.186
The authenticity of host '54.85.79.186 (54.85.79.186)' can't be established.
ED25519 key fingerprint is SHA256:ffz946cqlxbNvBsPqtNcxLlfXmhW8VJhyLD4n4jStto.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '54.85.79.186' (ED25519) to the list of known hosts.

      _#
      ~\_ #####_      Amazon Linux 2023
      ~~ \_#####\
      ~~   \###|
      ~~     \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023
      ~~       V~' '-->
      ~~~
      ~~..  /
      ~~/_ -/
      _/m'

Last login: Sat Sep 14 06:03:27 2024 from 18.206.107.27
[ec2-user@ip-172-31-20-75 ~]$
```

With this you can continue your commands through local terminal

- From now on, until mentioned, perform these steps on all 3 machines.

## Install Docker

**sudo yum install docker -y**

```
[ec2-user@ip-172-31-31-212 ~]$ sudo yum install docker -y
Last metadata expiration check: 0:33:43 ago on Thu Sep 12 13:11:13 2024.
Dependencies resolved.
=====
Package          Architecture Version
=====
Installing:
docker           x86_64      25.0.6-1.amzn2023.0.2
Installing dependencies:
containerd        x86_64      1.7.20-1.amzn2023.0.1
iptables-libs    x86_64      1.8.8-3.amzn2023.0.2
iptables-nft     x86_64      1.8.8-3.amzn2023.0.2
libcgroup         x86_64      3.0-1.amzn2023.0.1
libnetfilter_conntrack x86_64      1.0.8-2.amzn2023.0.2
libnftnl          x86_64      1.0.1-19.amzn2023.0.2
runc              x86_64      1.2.2-2.amzn2023.0.2
pigz              x86_64      2.5-1.amzn2023.0.3
runc              x86_64      1.1.13-1.amzn2023.0.1
=====
Transaction Summary
```

Then, configure cgroup in a daemon.json file by using following commands

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

```
[ec2-user@ip-172-31-20-75 ~]$ cd /etc/docker
[ec2-user@ip-172-31-20-75 docker]$ cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
EOF
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
[ec2-user@ip-172-31-20-75 docker]$ ls
daemon.json  kubectl
```

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

```
[ec2-user@ip-172-31-212 docker]$ sudo systemctl enable docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
[ec2-user@ip-172-31-212 docker]$ sudo systemctl daemon-reload
[ec2-user@ip-172-31-212 docker]$ sudo systemctl restart docker
[ec2-user@ip-172-31-212 docker]$ docker -v
Docker version 25.0.5, build 5dc9bcc
[ec2-user@ip-172-31-212 docker]$ █
```

## 4. Install Kubernetes on all 3 machines

SELinux needs to be disabled before configuring kubelet

- sudo setenforce 0
- sudo sed -i 's/^SELINUX=enforcing\$/SELINUX=permissive/' /etc/selinux/config

```
[ec2-user@ip-172-31-26-2 docker]$ sudo setenforce 0
[ec2-user@ip-172-31-26-2 docker]$ sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
[ec2-user@ip-172-31-26-2 docker]$ █
```

Add kubernetes repository (paste in terminal)

```
cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/repo/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
EOF
```

Type following commands to install set of kubernetes packages:

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

```
[ec2-user@ip-172-31-212 docker]$ sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Last metadata expiration check: 0:00:23 ago on Thu Sep 12 14:09:10 2024.
Dependencies resolved.
=====
Package           Architecture Version
=====
Installing:
kubeadm          x86_64      1.30.5-150500.1.1
kubectl          x86_64      1.30.5-150500.1.1
kubelet          x86_64      1.30.5-150500.1.1
Installing dependencies:
conctrack-tools   x86_64      1.4.6-2.amzn2023.0.2
cri-tools         x86_64      1.30.1-150500.1.1
kubernetes-cni    x86_64      1.4.0-150500.1.1
libnetfilter_cthelper x86_64  1.0.0-21.amzn2023.0.2
libnetfilter_cttimeout x86_64  1.0.0-19.amzn2023.0.2
libnetfilter_queue x86_64      1.0.5-2.amzn2023.0.2
=====
Transaction Summary
=====
Install  9 Packages

Total download size: 53 M
Installed size: 292 M
Downloading Packages:
(1/9): libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64.rpm
(2/9): libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64.rpm
(3/9): libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64.rpm
(4/9): contrack-tools-1.4.6-2.amzn2023.0.2.x86_64.rpm
(5/9): kubeadm-1.30.5-150500.1.1.x86_64.rpm
(6/9): kubectl-1.30.5-150500.1.1.x86_64.rpm
(7/9): cri-tools-1.30.1-150500.1.1.x86_64.rpm
(8/9): kubernetes-cni-1.4.0-150500.1.1.x86_64.rpm
(9/9): kubelet-1.30.5-150500.1.1.x86_64.rpm
=====
Total
Kubernetes
Importing GPG key 0x9A296436:
  Userid : "isv:kubernetes OBS Project <isv:kubernetes@build.opensuse.org>"
  Fingerprint: DE15 B144 86CD 377B 9E87 6E1A 2346 54DA 9A29 6436
  From   : https://pkgs.k8s.io/core:/stable:/v1.30/rpm/repo/repomd.xml.key
Key imported successfully
Running transaction check
Transaction check succeeded.
```

After installing Kubernetes, we need to configure internet options to allow bridging.

- sudo swapoff -a
- echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf
- sudo sysctl -p

## 5. Perform this ONLY on the Master machine

Initialize kubernetes by typing below command

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

```
[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 172.31.26.2:6443 --token 6cjsz0.8ei243v0zn9k7erg \
  --discovery-token-ca-cert-hash sha256:abd917ec30e12c5616bf647a3d174bef3d271e92c30b8f2f7768cfb3181341d4
[ec2-user@ip-172-31-26-2 docker]$ █
```

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
```

**Copy this join link and save it in clipboard (copy from your output as it different for each master instance)**

Example :

```
kubeadm join 172.31.20.75:6443 --token 66kg9u.2bc0kze31hrwbzvr \
  --discovery-token-ca-cert-hash
sha256:5e478da328b199e17d9b5da68e78bc9a6daab2043b05860552f4c184a7b3cb66
```

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

**Command:**

```
kubectl apply -f
```

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

```
[ec2-user@ip-172-31-26-2 docker]$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
```

## 6. Perform this ONLY on the worker machines

Paste the below command on all 2 worker machines

- sudo yum install iproute-tc -y
- sudo systemctl enable kubelet
- sudo systemctl restart kubelet

```
[ec2-user@ip-172-31-212 docker]$ sudo yum install iproute-tc -y
Last metadata expiration check: 0:15:14 ago on Thu Sep 12 14:09:10 2024.
Dependencies resolved.
=====
Package           Architecture      Version       Repository   Size
=====
Installing:
iproute-tc        x86_64          5.10.0-2.amzn2023.0.5
amazonlinux        455 k
=====
Transaction Summary
=====
install 1 Package
=====
Total download size: 455 k
Installed size: 928 k
Downloading Packages:
iproute-tc-5.10.0-2.amzn2023.0.5.x86_64.rpm                               4.0 MB/s | 455 kB     00:00
2.8 MB/s | 455 kB     00:00
=====
Total
running transaction check
'release' check succeeded.
running transaction test
'transaction test succeeded.
running transaction
Preparing
  running scriptlet : iproute-tc-5.10.0-2.amzn2023.0.5.x86_64
  Running scriptlet: iproute-tc-5.10.0-2.amzn2023.0.5.x86_64
  Verifying       : iproute-tc-5.10.0-2.amzn2023.0.5.x86_64
=====
installed:
iproute-tc-5.10.0-2.amzn2023.0.5.x86_64
=====
Complete!
[ec2-user@ip-172-31-21-212 docker]$
[ec2-user@ip-172-31-17-184 docker]$ sudo systemctl enable kubelet
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /usr/lib/systemd/system/kubelet.service.
[ec2-user@ip-172-31-17-184 docker]$ sudo systemctl restart kubelet
[ec2-user@ip-172-31-17-184 docker]$
```

Now paste the hash that you copied in these worker nodes to connect to master cluster

- kubeadm join 172.31.20.75:6443 --token 66kg9u.2bc0kze31hrwbzvr \  
--discovery-token-ca-cert-hash  
sha256:5e478da328b199e17d9b5da68e78bc9a6daab2043b05860552f4c184a7b3cb66

Now we can see in the master/control node of Kubernetes that worker nodes are connected by this command

- **watch kubectl get nodes**  
(in the master node instance)

Errors faced during the execution :

1. In the end kubelet might not respond or the connectivity of nodes to master might not happen

2. You can see this error

```
[ec2-user@ip-172-31-20-75 docker]$ kubectl get nodes
E0914 06:14:55.956919 3650 memcache.go:265] couldn't get current server API group list: Get "https://172.31.20.75:6443/api?timeout=32s";
connection refused
E0914 06:14:55.957758 3650 memcache.go:265] couldn't get current server API group list: Get "https://172.31.20.75:6443/api?timeout=32s";
connection refused
E0914 06:14:55.959507 3650 memcache.go:265] couldn't get current server API group list: Get "https://172.31.20.75:6443/api?timeout=32s";
connection refused
E0914 06:14:55.960160 3650 memcache.go:265] couldn't get current server API group list: Get "https://172.31.20.75:6443/api?timeout=32s";
connection refused
E0914 06:14:55.961526 3650 memcache.go:265] couldn't get current server API group list: Get "https://172.31.20.75:6443/api?timeout=32s";
connection refused
```

3. Try to restart the kubelet from worker instance and try the commands again

### Conclusion :

- The experiment aimed to deploy Kubernetes on Docker by connecting a master node to two worker nodes.
- Encountered issues with misconfigured SSH inbound rules.
- Resolved the SSH issue by correctly enabling necessary access rules.
- Realized that using t2.medium or t3 instances was essential to provide sufficient resources for Kubernetes.
- Despite adjustments, the worker nodes could not join the cluster.
- Master node was successfully initialized, but worker nodes faced issues.
- Possible causes of worker node problems included:
  - Misconfigurations in kubelet setup.
  - Networking challenges.
  - Inability of worker nodes to communicate with the master node's API server.
  - Potential incorrect firewall settings.
  - Missing API server certificates.
  - Errors during the kubeadm join process on the worker nodes.

### 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.

# Steps:

Log in to your AWS Academy/personal account.

## 1. Create security group

Create security group with following configuration lets name it as exp4.

The screenshot shows the 'Create security group' wizard in the AWS Management Console. The 'Basic details' step is completed with the security group name 'exp4SecurityGroup' and a description 'security for exp4'. The VPC dropdown is set to 'vpc-07b6966cbfbba88ee3'. The 'Inbound rules' step shows three rules defined:

Type	Protocol	Port range	Source	Description - optional
SSH	TCP	22	Anywhere-0.0.0.0/0	
Custom TCP	TCP	6443	Anywhere-0.0.0.0/0	
All traffic	All	All	Anywhere-0.0.0.0/0	

An 'Add rule' button is visible at the bottom of the list.

click on create security group.

The screenshot shows the AWS EC2 Security Groups page. A success message at the top states: "Security group (sg-0f87a692e5cdada58 | exp4SecurityGroup) was created successfully". The main content area displays the details of the new security group "sg-Of87a692e5cdada58 - exp4SecurityGroup". The "Details" section includes fields for Security group name (sg-0f87a692e5cdada58), Description (security for exp4), Owner (209322483715), VPC ID (vpc-07b6966cbfba88ee3), Security group ID (sg-0f87a692e5cdada58), Inbound rules count (3 Permission entries), and Outbound rules count (1 Permission entry). Below this, there are tabs for "Inbound rules" (selected), "Outbound rules", and "Tags". The "Inbound rules" table has three rows:

Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
-	sgr-0495fce359ab0d547	IPv4	SSH	TCP	22	0.0.0.0/0	-
-	ec2-0f87a692e5cdada58.1	IPv4	All traffic	All	All	0.0.0.0/0	-

## 2. Create Instance

Launch an ec2 instance.

**Name and tags** [Info](#)

Name  
exp4 [Add additional tags](#)

**Application and OS Images (Amazon Machine Image)** [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

[Search our full catalog including 1000s of application and OS images](#)

[Recents](#) [Quick Start](#)

[Amazon Linux](#) [macOS](#) [Ubuntu](#) [Windows](#) [Red Hat](#) [SUSE Li](#) [Browse more AMIs](#)  
Including AMIs from AWS, Marketplace and the Community

**Amazon Machine Image (AMI)**

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type [Free tier eligible](#)

ami-0e86e20dae9224db8 (64-bit (x86)) / ami-096ea6a12ea24a797 (64-bit (Arm))  
Virtualization: hvm ENA enabled: true Root device type: ebs

**Description**  
Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Architecture	AMI ID	Username	Verified provider
64-bit (x86)	ami-0e86e20dae9224db8	ubuntu	

Select Ubuntu 22.04 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.

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type  
ami-0e86e20dae9224db8 (64-bit (x86)) / ami-096ea6a12ea24a797 (64-bit (Arm))  
Virtualization: hvm FNA enabled: true Root device type: ebs

Free tier eligible

**t2.small**

Family: t2 1 vCPU 2 GiB Memory Current generation: true  
On-Demand Windows base pricing: 0.032 USD per Hour  
On-Demand Linux base pricing: 0.023 USD per Hour  
On-Demand RHEL base pricing: 0.0376 USD per Hour  
On-Demand SUSE base pricing: 0.053 USD per Hour

**t2.medium**

Family: t2 2 vCPU 4 GiB Memory Current generation: true  
On-Demand Linux base pricing: 0.0464 USD per Hour  
On-Demand RHEL base pricing: 0.0752 USD per Hour  
On-Demand Windows base pricing: 0.0644 USD per Hour  
On-Demand SUSE base pricing: 0.1464 USD per Hour

**t2.large**

Family: t2 2 vCPU 8 GiB Memory Current generation: true  
On-Demand Windows base pricing: 0.1208 USD per Hour  
On-Demand RHEL base pricing: 0.1216 USD per Hour  
On-Demand SUSE base pricing: 0.1928 USD per Hour  
On-Demand Linux base pricing: 0.0928 USD per Hour

**t2.xlarge**

Family: t2 4 vCPU 16 GiB Memory Current generation: true  
On-Demand Windows base pricing: 0.2416 USD per Hour  
On-Demand RHEL base pricing: 0.2432 USD per Hour  
On-Demand SUSE base pricing: 0.3856 USD per Hour  
On-Demand Linux base pricing: 0.1856 USD per Hour

**t2.micro**

Family: t2 1 vCPU 1 GiB Memory Current generation: true  
On-Demand Windows base pricing: 0.0162 USD per Hour  
On-Demand SUSE base pricing: 0.0116 USD per Hour  
On-Demand RHEL base pricing: 0.026 USD per Hour  
On-Demand Linux base pricing: 0.0116 USD per Hour

Additional costs apply for AMIs with pre-installed software

Available from Canonical

All generations

Compare instance types

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

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.

Create a key-pair to login to the machine remotely and then select this newly generated key-pair.

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

exp4

 [Create new key pair](#)

▼ Network settings [Info](#)

[Edit](#)

Network | [Info](#)

vpc-07b6966cbfba88ee3

Subnet | [Info](#)

No preference (Default subnet in any availability zone)

Select existing security group and select the security group we created at the start.

**Network settings** [Info](#)

[Edit](#)

Network | [Info](#)  
vpc-07b6966cbfba88ee3

Subnet | [Info](#)  
No preference (Default subnet in any availability zone)

Auto-assign public IP | [Info](#)  
Enable  
Additional charges apply when outside of free tier allowance

Firewall (security groups) | [Info](#)  
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group     Select existing security group

Common security groups [Info](#)

Select security groups ▾

exp4SecurityGroup sg-0f87a692e5cdada58 X  
VPC: vpc-07b6966cbfba88ee3

Compare security group rules

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

**Configure storage** [Info](#) [Advanced](#)

Launch the instance.

### 3. Connect to the instance.

Select the instance created

click on Connect the instance and navigate to SSH Client.

Instances (1/1) [Info](#)

Last updated less than a minute ago [C](#) [Connect](#) [Inst](#)

Find Instance by attribute or tag (case-sensitive) All states ▾

Instance ID: i-09426999c36422602 X Clear filters

<input checked="" type="checkbox"/>	Name <a href="#">D</a>	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	<a href="#">F</a>
<input checked="" type="checkbox"/>	exp4	i-09426999c36422602	<a href="#">Running</a> <a href="#">Q</a> <a href="#">Q</a>	t2.medium	<a href="#">Initializing</a>	<a href="#">View alarms +</a>	us-east-1b	<a href="#">E</a>

Copy the command that comes to your dashboard at the bottom.

## Connect to instance Info

Connect to your instance i-09426999c36422602 (exp4) using any of these options

EC2 Instance Connect | Session Manager | **SSH client** | EC2 serial console

Instance ID

i-09426999c36422602 (exp4)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is exp4.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.  
 chmod 400 "exp4.pem"

4. Connect to your instance using its Public DNS:

ec2-44-212-57-152.compute-1.amazonaws.com

 Command copied

ssh -i "exp4.pem" ubuntu@ec2-44-212-57-152.compute-1.amazonaws.com

**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

Now copy the path to the file where our .pem key is stored and replace the pem file in the command copied from the ssh dashboard.

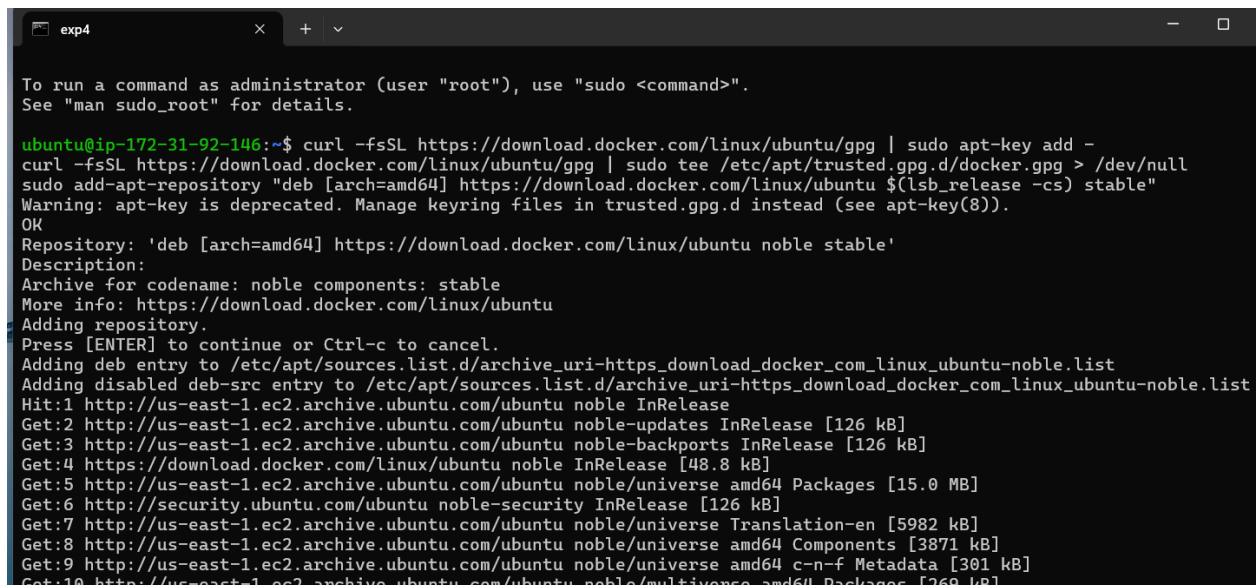
```
C:\Users\Lenovo>ssh -i "C:\Users\Lenovo\Downloads\exp4.pem" ubuntu@ec2-44-212-57-152.compute-1.amazonaws.com
The authenticity of host 'ec2-44-212-57-152.compute-1.amazonaws.com (44.212.57.152)' can't be established.
ED25519 key fingerprint is SHA256:SYWntsQatiMJ2x6vE4Nabz7KWXcSDPgjer2N22WJ7eU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes|
```

## 4. Install and set-up Docker

Run the following commands:

1. We have to install and setup Docker. Run these commands

```
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"
```

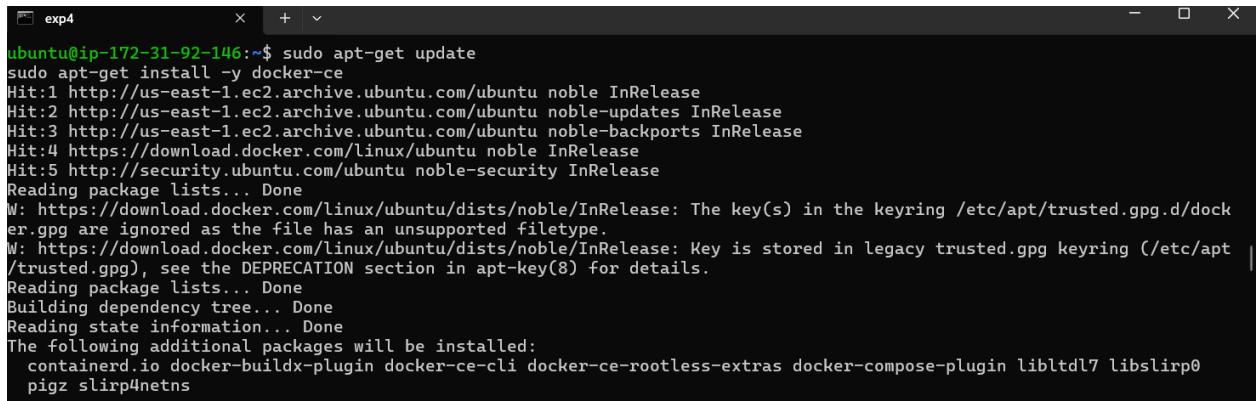


```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-92-146:~$ 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"
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
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.
Adding deb entry to /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
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 https://download.docker.com/linux/ubuntu noble InRelease [48.8 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
```

2. Update

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



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

```
exp4 x + -  
Setting up docker-ce (5:27.3.1~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.  
ubuntu@ip-172-31-92-146:~$ |
```

3. Configure Docker to use the `systemd` cgroup driver by creating the necessary configuration file in the `/etc/docker` directory.

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

```
exp4 x + -  
ubuntu@ip-172-31-92-146:~$ sudo mkdir -p /etc/docker  
cat <<EOF | sudo tee /etc/docker/daemon.json  
{  
    "exec-opts": ["native.cgroupdriver=systemd"]  
}  
EOF  
{  
    "exec-opts": ["native.cgroupdriver=systemd"]  
}  
ubuntu@ip-172-31-92-146:~$ |
```

4. Restart and enable docker:

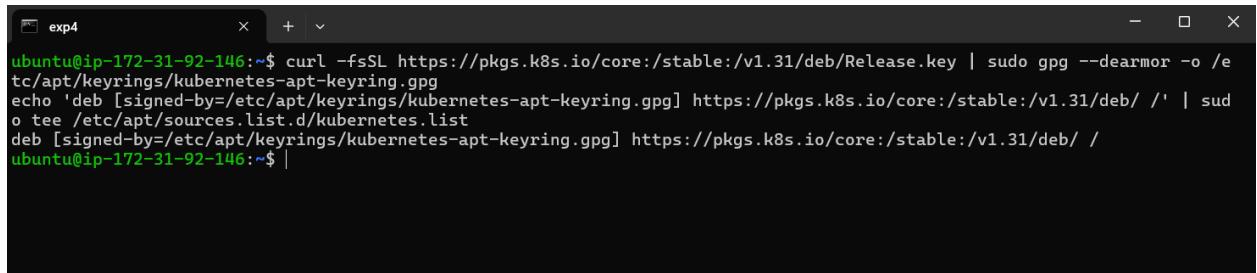
```
sudo systemctl enable docker  
sudo systemctl daemon-reload  
sudo systemctl restart docker
```

```
ubuntu@ip-172-31-92-146:~$ sudo systemctl enable docker  
sudo systemctl daemon-reload  
sudo systemctl restart 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  
ubuntu@ip-172-31-92-146:~$ |
```

## 5. Set-up Kubernetes

1. Add the Kubernetes signing key and repository to your APT sources for package installation.

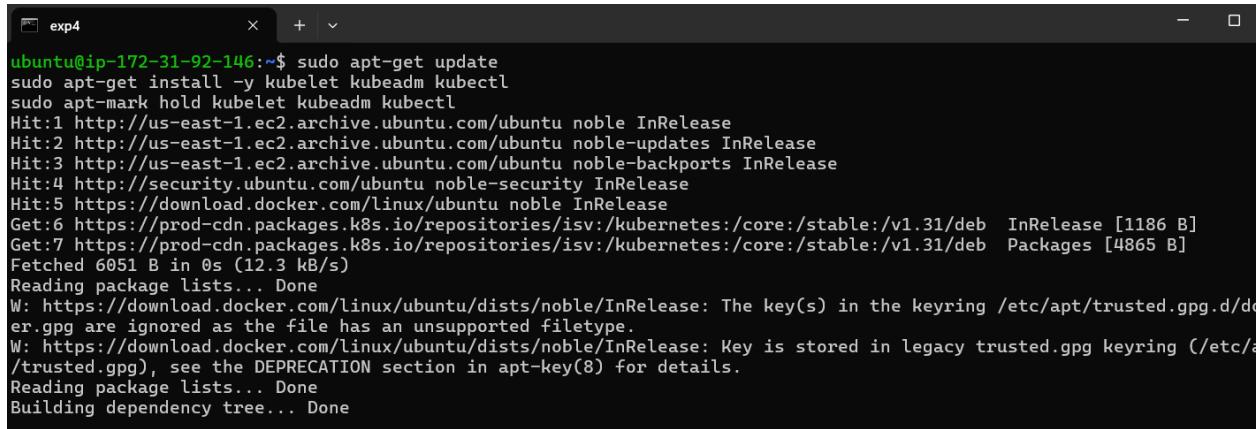
```
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key |  
sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg  
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]  
https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /' | sudo tee  
/etc/apt/sources.list.d/kubernetes.list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



```
ubuntu@ip-172-31-92-146:~$ 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  
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /  
ubuntu@ip-172-31-92-146:~$
```

2. Update APT package lists, install Kubernetes tools (`kubelet`, `kubeadm`, `kubectl`), and mark them to prevent automatic updates.

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



```
ubuntu@ip-172-31-92-146:~$ 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 (12.3 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 an unsupported filetype.  
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.  
Reading package lists... Done  
Building dependency tree... Done
```

```
exp4 x + v
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.

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

## 6. Initialize the kubernetes cluster

1. Enable and start the `kubelet` service, then initialize the Kubernetes cluster with a specified pod network CIDR.

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

```
exp4 x + v
ubuntu@ip-172-31-92-146:~$ sudo systemctl enable --now kubelet
sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
W1001 08:58:04.356900    4249 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 f
or 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[p
reflight] 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
ubuntu@ip-172-31-92-146:~$ |
```

Here, we encounter an error as a few of the dependencies for running the command are not installed. So, run the following commands

```
sudo apt-get install -y containerd
```

```
exp4 x + v
See https://ubuntu.com/esm or run: sudo pro status

Last login: Tue Oct  1 08:26:47 2024 from 125.99.93.18
ubuntu@ip-172-31-92-146:~$ history
 1 history
ubuntu@ip-172-31-92-146:~$ 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:
```

```
exp4 x + v
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.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-92-146:~$ |
```

2. Create the `/etc/containerd` directory and generate the default `containerd` configuration file (`config.toml`).

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

```
exp4 x + v
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-92-146:~$ 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 = ""
```

```
[timeouts]
  "io.containerd.timeout.bolt.open" = "0s"
  "io.containerd.timeout.metrics.shimstats" = "2s"
  "io.containerd.timeout.shim.cleanup" = "5s"
  "io.containerd.timeout.shim.load" = "5s"
  "io.containerd.timeout.shim.shutdown" = "3s"
  "io.containerd.timeout.task.state" = "2s"

[ttrpc]
  address = ""
  gid = 0
  uid = 0
ubuntu@ip-172-31-92-146:~$ |
```

3. Restart, enable, and check the status of the `containerd` service.

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

```
[ exp4 ] exp4 x + v
ubuntu@ip-172-31-92-146:~$ sudo systemctl restart containerd
sudo systemctl enable containerd
sudo systemctl status containerd
● containerd.service - containerd container runtime
  Loaded: loaded (/usr/lib/systemd/system/containerd.service; enabled; preset: enabled)
  Active: active (running) since Tue 2024-10-01 09:18:05 UTC; 274ms ago
    Docs: https://containerd.io
   Main PID: 5581 (containerd)
      Tasks: 7
     Memory: 13.6M (peak: 13.9M)
        CPU: 71ms
      CGroup: /system.slice/containerd.service
              └─5581 /usr/bin/containerd

Oct 01 09:18:05 ip-172-31-92-146 containerd[5581]: time="2024-10-01T09:18:05.013214307Z" level=info msg=serving...
Oct 01 09:18:05 ip-172-31-92-146 containerd[5581]: time="2024-10-01T09:18:05.013246107Z" level=info msg=serving...
Oct 01 09:18:05 ip-172-31-92-146 containerd[5581]: time="2024-10-01T09:18:05.013300897Z" level=info msg="Start subscrib...
Oct 01 09:18:05 ip-172-31-92-146 containerd[5581]: time="2024-10-01T09:18:05.013325717Z" level=info msg="Start recoveri...
```

4. Install the `socat` package using APT with no prompts.

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

```
[ exp4 ] exp4 x + v
ubuntu@ip-172-31-92-146:~$ 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 143 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 (9246 kB/s)
Selecting previously unselected package socat.
dpkg: warning: ignoring file /var/cache/apt/archives/socat_1.8.0.0-4build3_amd64.deb in favor of a newer version, /var/cache/apt/archives/socat_1.8.0.0-4build3_amd64.deb
```

5. Initialize the Kubernetes cluster with a specified pod network CIDR of `10.244.0.0/16`.

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

The screenshot shows two terminal windows side-by-side. Both windows have a title bar 'exp4' and a close button 'X'. The left terminal window displays the output of the 'kubeadm init' command, which includes logs about Kubernetes version v1.31.0, pre-flight checks, image pulling, certificate generation, and API server configuration. The right terminal window shows the configuration of the kubeconfig file, including setting \$HOME/.kube/config, and provides instructions for deploying a pod network using 'kubectl apply -f [podnetwork].yaml' and joining worker nodes.

```
ubuntu@ip-172-31-92-146:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[kubelet] 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'
W1001 09:21:47.290415      5902 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-172-31-92-146 kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 172.31.92.146]
[certs] Generating "apiserver-kubebundle-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key

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 172.31.92.146:6443 --token vg6cqy.kx64j8i2bp7qf776 \
    --discovery-token-ca-cert-hash sha256:812f3da588c8ecd9e96cf40a0ea5d99360e518299e5ec7b026f8e228c2017904
ubuntu@ip-172-31-92-146:~$ |
```

6. Deploy the Flannel network add-on for Kubernetes by applying the specified YAML configuration file from the provided URL.

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

7. Connect nginx server to pod.

The screenshot shows a terminal window with a title bar 'exp4' and a close button 'X'. The command 'kubectl get nodes' is run, and the output shows a single node named 'ip-172-31-81-27.ec2.internal' in status 'Ready', with 'control-plane' role and age '15m'. The version is 'v1.31.1'.

```
[ec2-user@ip-172-31-81-27 bin]$ kubectl get nodes
NAME                  STATUS   ROLES      AGE     VERSION
ip-172-31-81-27.ec2.internal   Ready   control-plane   15m   v1.31.1
[ec2-user@ip-172-31-81-27 bin]$ |
```

```
[ec2-user@ip-172-31-81-27 bin]$ kubectl apply -f https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
[ec2-user@ip-172-31-81-27 bin]$ |
```

```
[ec2-user@ip-172-31-81-27 bin]$ kubectl get pods
NAME                      READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-4prm9   0/1     Pending   0          94s
nginx-deployment-d556bf558-d6dld   0/1     Pending   0          94s
[ec2-user@ip-172-31-81-27 bin]$ |
```

## Conclusion:

In this experiment, we have learned how to deploy an nginx server to a kubernetes cluster. We also learned how to tackle any intolerable taints that tend to give issues while deploying the server. We also learned how to set the port on which you want to host the server.

## Installation guide of Teraform (Windows).

### Step 1: Download terraform

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

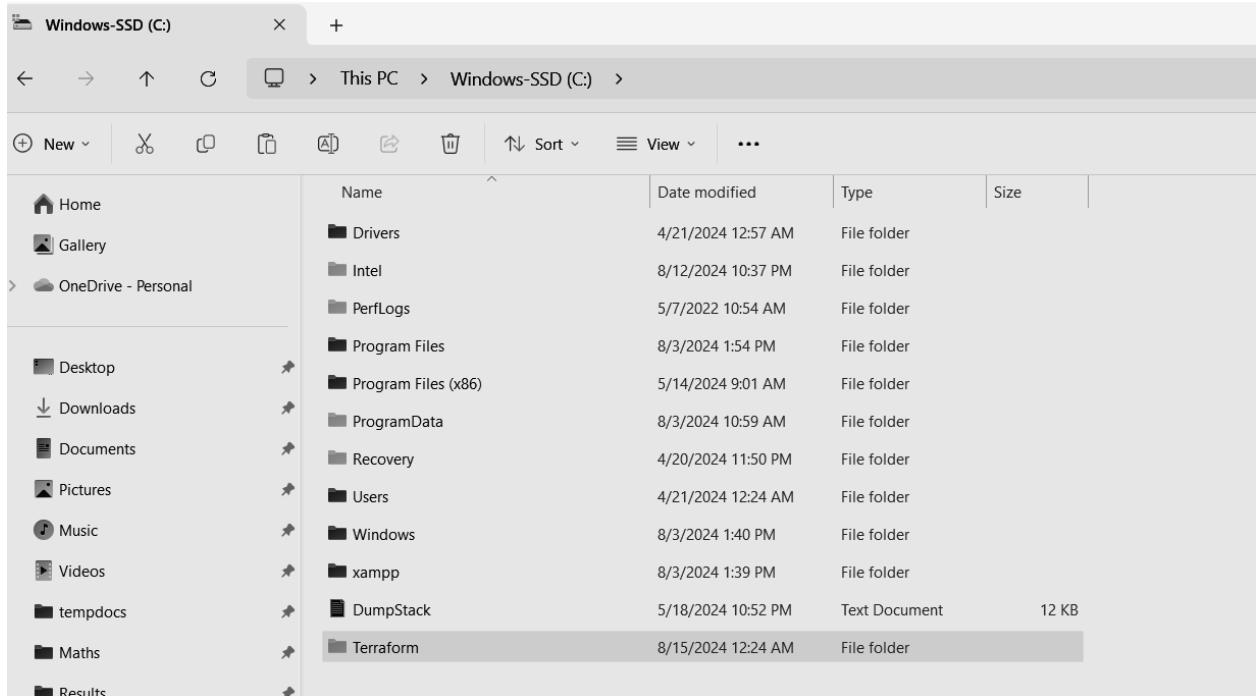
website:<https://www.terraform.io/downloads.html>

Select the Operating System Windows followed by either 32bit (386) or 64 bit (AMD64) based on your OS type.

The screenshot shows the Terraform download page. On the left, there's a sidebar with links for macOS, Windows (which is selected), Linux, FreeBSD, OpenBSD, Solaris, Release information, and Next steps. The main content area has tabs for Windows and Linux. Under the Windows tab, it says "Binary download" and shows two options: "386 Version: 1.9.4" and "AMD64 Version: 1.9.4", each with a "Download" button. Under the Linux tab, it says "Package manager" and lists Ubuntu/Debian, CentOS/RHEL, Fedora, Amazon Linux, and Homebrew.

## Step 2: Create a folder for Terraform:

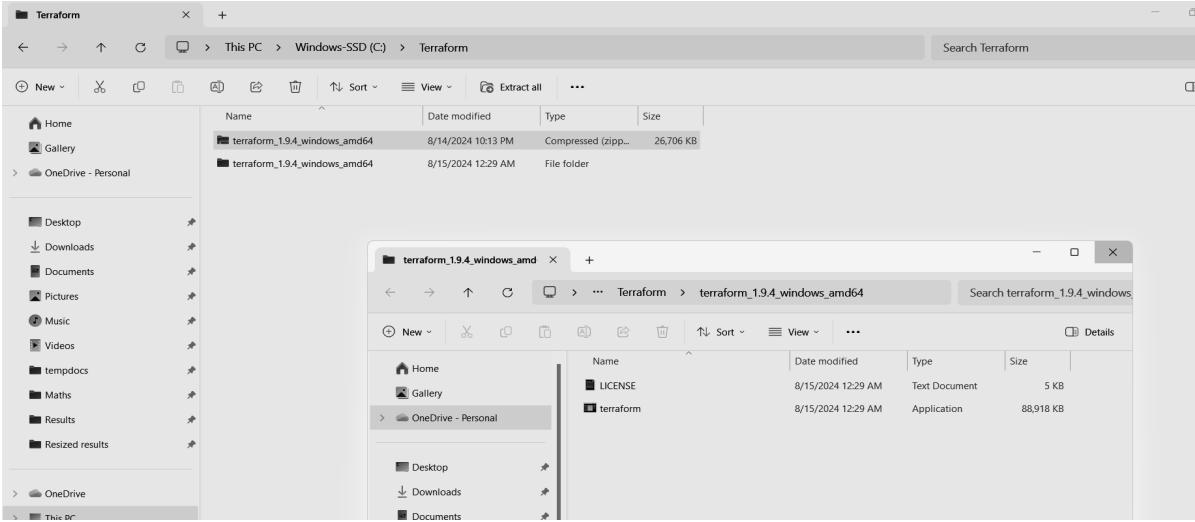
- Open **File Explorer** and navigate to **C:\** or another location where you want to store Terraform.
- Right-click in the folder, select **New > Folder**, and name the folder something like **Terraform**.
- For example, the folder might be **C:\Terraform**.



## Step 3: Extract Terraform into this folder:

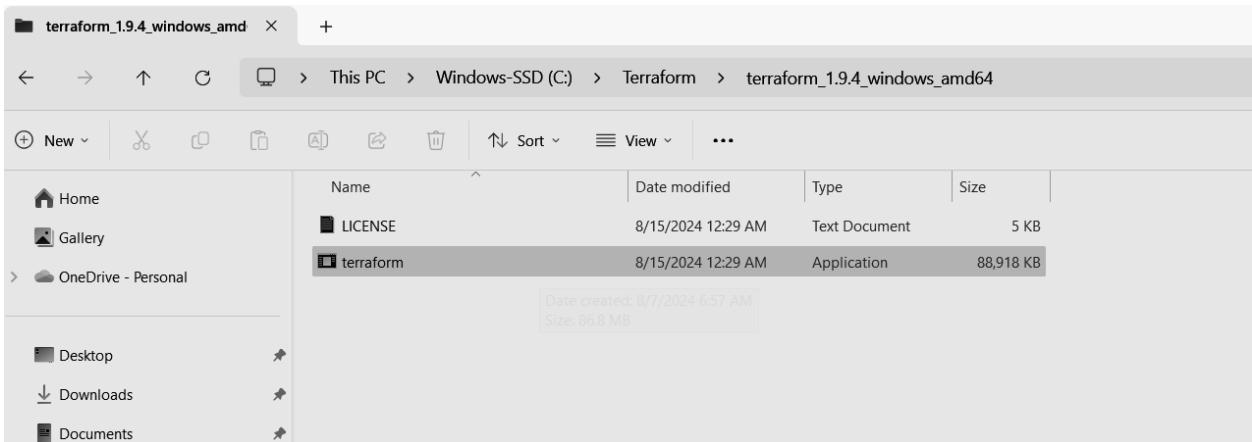
- Make sure to move the zip file from downloads to this new folder if it is not there already.
- After downloading the **.zip** file, right-click the **.zip** and choose **Extract All**.
- Extract the contents of the **.zip** file into the folder you just created (e.g., **C:\Terraform**).

\*the background is the terraform folder in the c drive where we unzip the file and the forefront image is of contents inside this unzipped folder.

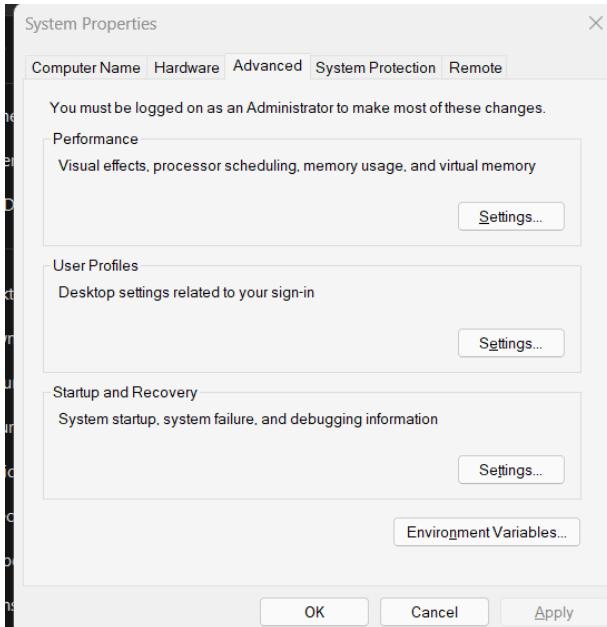


## Step 4: Add to path

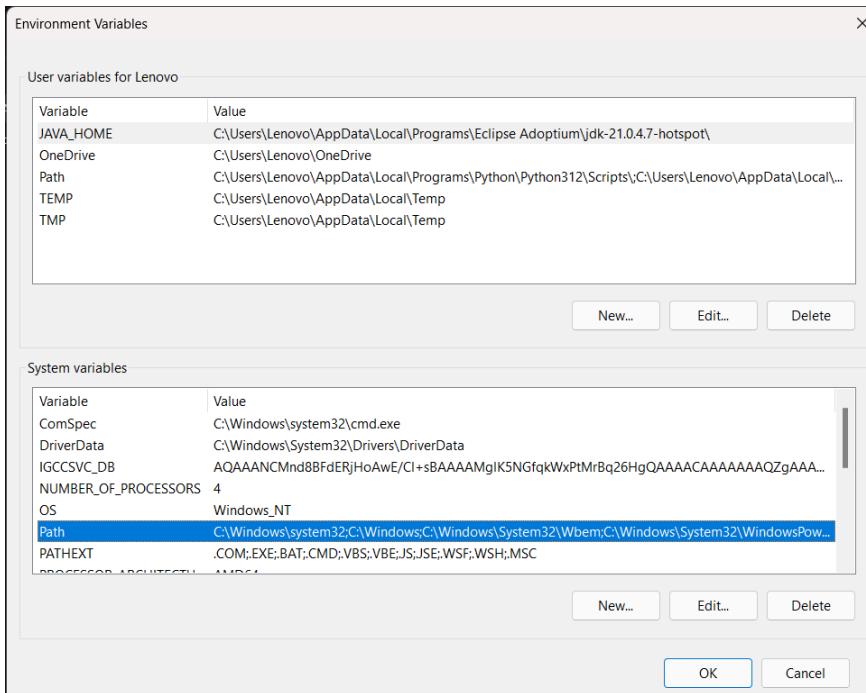
- Theory : the system uses top level search for finding the executable files, so rule of thumb is to add the directory which is a direct parent to the terraform exe.
- Hence, we first locate the terraform.exe and only copy the path till its parent directory. the highlighted file is the executable file and we will just copy that `terraform_1.9.4_windows_amd64` visible in the path above.



- Now search for edit environment variables and click on it.
- Now click on environment variables.

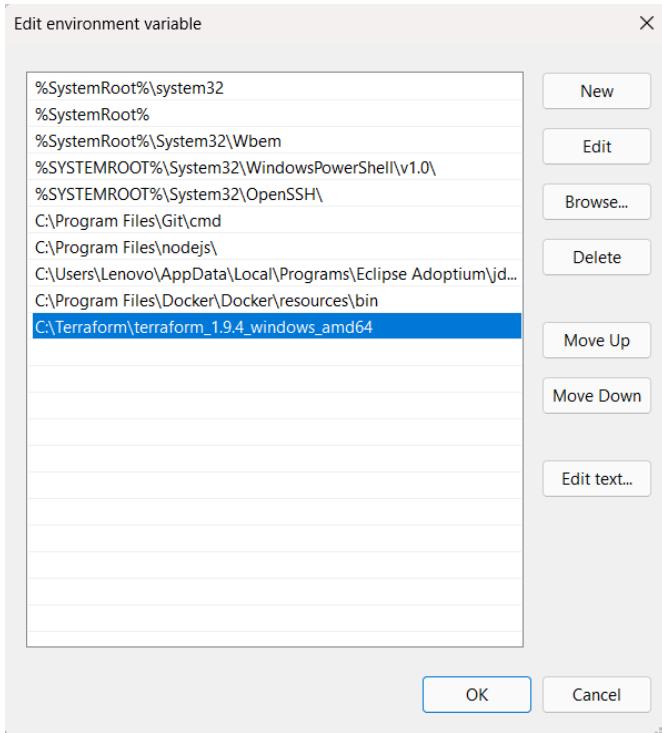


- Now select path from the system variables and click on edit.



\*Note click on edit and not new!

- Now paste the path to the directory that we copied.



## Step 5: Run terraform command in shell to see if it works

```
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\Lenovo> 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
  test       Execute integration tests for Terraform modules
  untaint   Remove the 'tainted' state from a resource instance
  version    Show the current Terraform version
  workspace  Workspace management
```

Name: Shubham Jha

Div: D15C

# Experiment No.: 6

## Implementation:

### A. Creating docker image using terraform

Prerequisite:

- 1) Download and Install Docker Desktop from <https://www.docker.com/>

**Step 1:** Check the docker functionality

```
PS C:\Users\91773> docker
Usage: docker [OPTIONS] COMMAND
A self-sufficient runtime for containers

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

Management Commands:
  builder        Manage builds
  buildx*        Docker Buildx
  compose*       Docker Compose
  container      Manage containers
  context         Manage contexts
  debug*         Get a shell into any image or container
  desktop*       Docker Desktop commands (Alpha)
  dev*          Docker Dev Environments
```

```
PS C:\Users\91773> docker --version
Docker version 27.0.3, build 7d4bcd8
PS C:\Users\91773> |
```

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

**Step 2:** Firstly create a new folder named ‘Docker’ in the ‘TerraformScripts’ folder. Then create a new docker.tf file using Atom editor and write the

following contents into it to create a Ubuntu Linux container.

Script:

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

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

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

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

docker.tf X

```
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

```
PS C:\Users\91773\Desktop\College Resources\TerraformScripts\Docker> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding kreuzwerker/docker versions matching "2.21.0"...
- Installing kreuzwerker/docker v2.21.0...
- Installed kreuzwerker/docker v2.21.0 (self-signed, key ID BD080C4571C6104C)
  Partner and community providers are signed by their developers.
  If you'd like to know more about provider signing, you can read about it here:
    https://www.terraform.io/docs/cli/plugins/signing.html
  Terraform has created a lock file .terraform.lock.hcl to record the provider
  selections it made above. Include this file in your version control repository
  so that Terraform can guarantee to make the same selections by default when
  you run "terraform init" in the future.

Terraform has been successfully initialized!

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

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

### Step 4: Execute Terraform plan to see the available resources

```
PS C:\Users\91773\Desktop\College Resources\TerraformScripts\Docker> terraform plan
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the
following symbols:
+ create

Terraform will perform the following actions:

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

```

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

+ healthcheck (known after apply)

+ labels (known after apply)
}

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

```

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

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

```

PS C:\Users\91773\Desktop\College Resources\TerraformScripts\Docker> terraform apply
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubun
tu:latest]

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

Terraform will perform the following actions:

# docker_container.foo will be created
+ resource "docker_container" "foo" {
  + attach          = false
  + bridge          = (known after apply)
  + command         = [
    + "sleep",
    + "3600",
  ]
}
```

Docker images, Before Executing Apply step:

```

PS C:\Users\91773\Desktop\College Resources\TerraformScripts\Docker> docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE

```

Docker images, After Executing Apply step:

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

PS C:\Users\91773\Desktop\College Resources\TerraformScripts\Docker> docker images
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
ubuntu          latest       edbfe74c41f8   2 weeks ago   78.1MB
PS C:\Users\91773\Desktop\College Resources\TerraformScripts\Docker> |
```

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

```
PS C:\Users\91773\Desktop\College Resources\TerraformScripts\Docker> terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Refreshing state... [id=f03a28e4658896c23c9992f7a98eb1011befc7d014e997ea9fc6372da70b7903]

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
}

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

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

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

Enter a value: yes

docker_container.foo: Destroying... [id=f03a28e4658896c23c9992f7a98eb1011befc7d014e997ea9fc6372da70b7903]
docker_container.foo: Destruction complete after 0s
docker_image.ubuntu: Destroying... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_image.ubuntu: Destruction complete after 1s

Destroy complete! Resources: 2 destroyed.
```

Docker images After Executing Destroy step

```
PS C:\Users\91773\Desktop\College Resources\TerraformScripts\Docker> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
------------	-----	----------	---------	------

## **Adv DevOps Practical 7**

### **Aim:**

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

### **Integrating Jenkins with SonarQube:**

### **Prerequisites:**

- Jenkins installed
- Docker Installed (for SonarQube)
- SonarQube Docker Image

### **Steps**

#### **to integrate Jenkins with SonarQube**

Prerequisites: Make sure you have docker and jenkins installed.

Run **docker -v** to check the docker installation.

Run

1. Open up Jenkins Dashboard on localhost, port 8090 or whichever port it is at for you.

The screenshot shows the Jenkins dashboard. On the left, there are two collapsed sections: 'Build Queue' (No builds in the queue) and 'Build Executor Status' (1 Idle, 2 Idle, 1 offline). The main area displays a table of build history. The columns are: S (Status), W (Last Build), Name, Last Success, Last Failure, and Last Duration. The table contains four rows:

S	W	Name	Last Success	Last Failure	Last Duration
Green checkmark	Sun icon	sahil 7	24 days #2	N/A	96 ms
Green checkmark	Sun icon	Sahil exp6	24 days #3	N/A	1 sec
Blue circle with dots	Sun icon	SahilExp6	N/A	N/A	N/A
Red X	Cloud icon	sahiljob	N/A	24 days #1	1.5 sec

At the bottom left of the table, it says 'Icon: S M L'. At the top right, there are buttons for 'Add description' and 'log out'.

2. Run SonarQube in a Docker container using this command -

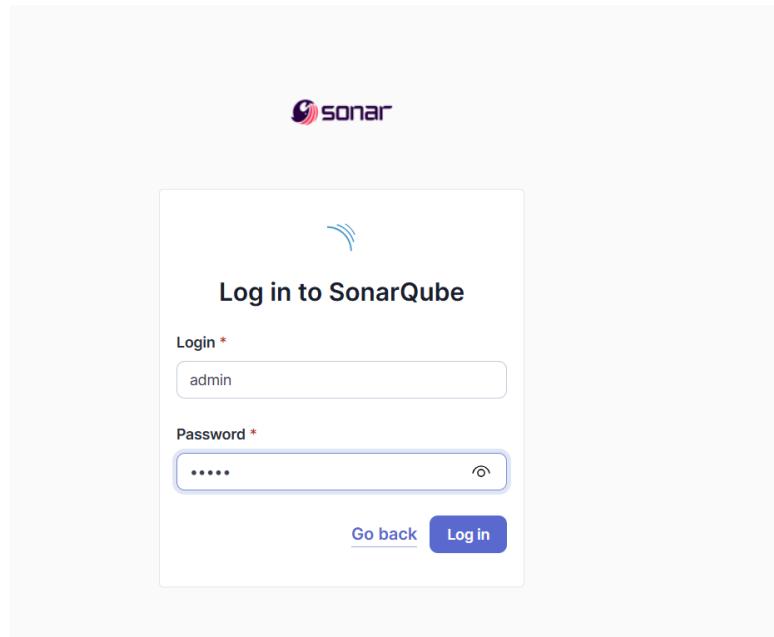
```
docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
```

-----**Warning: run below command only once**

```
C:\Users\Lenovo>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest  
47f6db8dbf2ed99dbe304bc0ebdf47b9d4144c4e4add42055ba44ce231058272
```

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

localhost port 9000.



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

(do change the password as you cannot use the default one)

## Update your password

 This account should not use the default password.

### Enter a new password

All fields marked with \* are required

Old Password \*

New Password \*

Confirm Password \*

**Update**



Projects Issues Rules Quality Profiles Quality Gates Administration More Q



### How do you want to create your project?

Do you want to benefit from all of SonarQube's features (like repository import and Pull Request decoration)?  
Create your project from your favorite DevOps platform.

First, you need to set up a DevOps platform configuration.

 Import from Azure DevOps Setup

 Import from Bitbucket Cloud Setup

 Import from Bitbucket Server Setup

 Import from GitHub Setup

 Import from GitLab Setup

Are you just testing or have an advanced use-case? Create a local project.

Create a local project

5. Create a manual project in SonarQube with the name sonarqube  
(Click on create local project)

1 of 2

## Create a local project

Project display name \*

exp7



Project key \*

exp7



Main branch name \*

main

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

[Cancel](#)

[Next](#)

## Setup the project

Projects Issues Rules Quality Profiles Quality Gates Administration More Q

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 baseline.  
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](#)

And come back to Jenkins Dashboard.

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



6. Under Jenkins 'Manage Jenkins' then go to 'system', scroll and look for **SonarQube Servers** and click on **add SonarQube** and then enter the details.

Enter the Server Authentication token if needed.(I didn't do it)

In SonarQube installations: Under **Name** add <project name of sonarqube> for me its sonarqube\_exp7

In Server URL Default is <http://localhost:9000>



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

**Dashboard > Manage Jenkins > Tools**



Click on **Add SonarQube Scanner**.

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



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



9. Choose this GitHub repository in Source Code Management.

[https://github.com/shazforiot/MSBuild\\_firstproject](https://github.com/shazforiot/MSBuild_firstproject)

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



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

## Configure

General

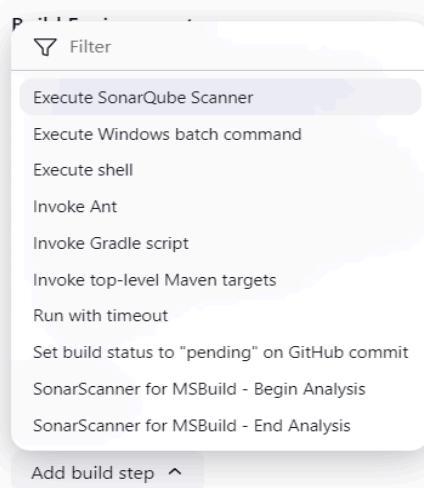
Source Code Management

Build Triggers

**Build Environment**

Build Steps

Post-build Actions



### Post-build Actions

Add post-build action ▾

Save

Apply

You will see something like this:



Open sonarQube again and go to Project Information appearing in the right side. Click on it and you can copy the project key from About the Project Section.



Use this key in place of <projectKey> in the following code

**sonar.projectKey =<projectKey>**

**sonar.login =admin**

**sonar.password =<yourpassword for sonar qube>**

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

**sonar.sources =.**

I personally preferred not keeping any spaces after '=' .

(Inherit From Job)

Path to project properties ?

Analysis properties ?

```
sonar.projectKey =exp7
sonar.login =admin
sonar.password =2923
sonar.hosturl =http://localhost:9000
sonar.sources =.
```

Additional arguments ?

Apply and save.

11. Go to sonarQube and go to administration → Security (dropdown) → Global Permissions.

See the administrator below and check the boxes which i checked.



12. Go to jenkins and click build:



# Conclusion:

In this project, I successfully integrated Jenkins with SonarQube to establish a robust automated static application security testing (SAST) pipeline. The setup involved deploying SonarQube using Docker, ensuring smooth container orchestration and efficient resource management. A key component was configuring Jenkins with the appropriate SonarQube plugins, authentication mechanisms, and linking it to a GitHub repository for continuous integration.

One of the challenges I faced was configuring Docker on the Jenkins environment, which required resolving networking issues between the Docker containers and ensuring that the SonarQube server was reachable from Jenkins. Additionally, setting up secure authentication between Jenkins and SonarQube involved troubleshooting token-based authentication and resolving environment path issues, particularly with the `JAVA_HOME` setup for the SonarQube scanner.

After overcoming these obstacles, I integrated the SonarQube scanner as a build step, allowing for continuous code analysis. This setup provided automated detection of code vulnerabilities, code smells, and quality issues. It helped ensure that any new commits triggered immediate analysis, generating detailed reports and promoting continuous improvement in code quality.

## 08 Advanced DevOps Lab

# Aim:

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

# Steps

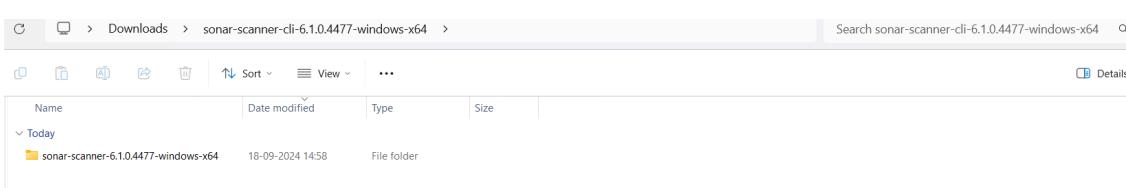
## Step 1: Download sonar scanner

<https://docs.sonarsource.com/sonarqube/latest/analyzing-source-code/scanners/sonarscanner/>

Visit this link and download the sonarqube scanner CLI.

The screenshot shows a web browser displaying the SonarScanner CLI page. The URL in the address bar is <https://docs.sonarsource.com/sonarqube/latest/analyzing-source-code/scanners/sonarscanner/>. The page title is "SonarScanner CLI". On the left, there is a sidebar with navigation links for SonarQube documentation, including "Homepage", "Try out SonarQube", "Server installation and setup", "Analyzing source code", "Scanners", and specific links for "SonarScanner CLI", "SonarQube extension for Azure DevOps", "SonarQube extension for Jenkins", "SonarScanner for .NET", and "SonarScanner for Maven". The main content area features a section titled "6.1" with a release date of "2024-06-27". It mentions "macOS and Linux AArch64 distributions" and provides download links for "Linux x64", "Linux AArch64", "Windows x64", "macOS x64", "macOS AArch64", "Docker Any (Requires a pre-installed JVM)", and "Release notes". Below this, a note states: "The SonarScanner CLI is the scanner to use when there is no specific scanner for your build system." Another note below it says: "The SonarScanner does not yet officially support ARM architecture. Still, early adopters reported it is working fine. If you encounter problems, don't hesitate to share your experience with us on the [SonarQube](#) or [SonarCloud](#) Community Forum but keep in mind that there is no support at this time."

Extract the downloaded zip file in a folder.



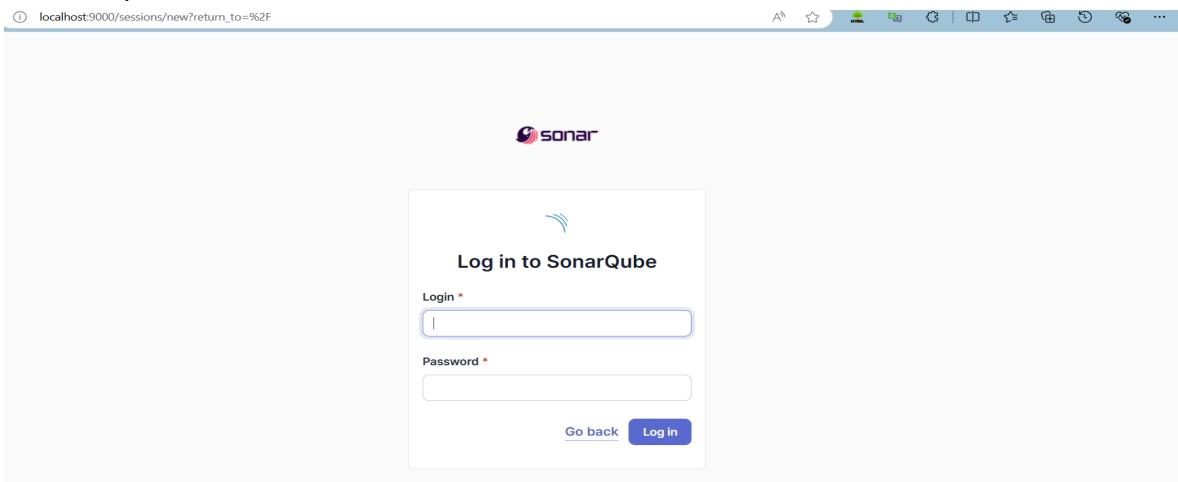
### 1. Install sonarqube image

Command: **docker pull sonarqube** (skip if already installed we did install it in exp 7)

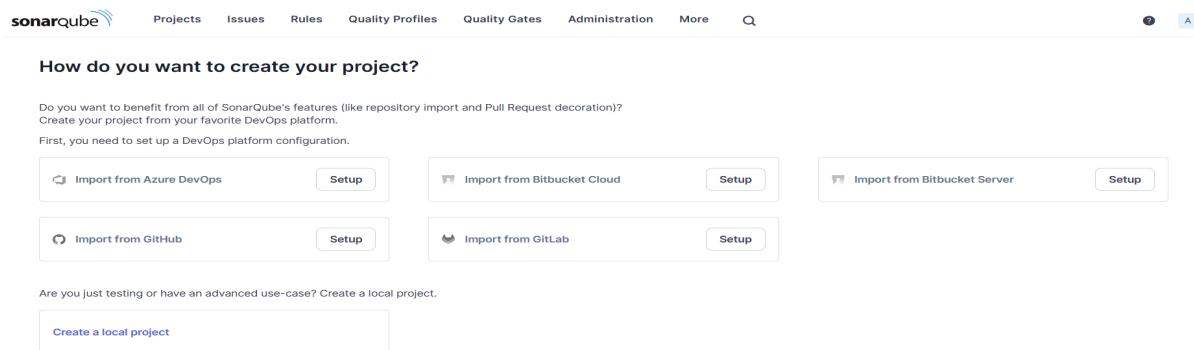
Then run the image

```
C:\Users\Lenovo>docker run -d -p 9000:9000 sonarqube  
5007285df5d17d62fef087bc6b74409e37fff333d6308ee62bd323fed5716d5d  
C:\Users\Lenovo>
```

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



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



### 4. Create a 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 at 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.

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

The Jenkins dashboard displays a list of build items:

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀️	DevOps Pipeline	1 mo 23 days #4	N/A	6.9 sec
✓	☀️	DevOps-NewJob	1 mo 14 days #1	N/A	0.67 sec
✓	☀️	DevOpsPipeLineExp6	1 mo 7 days #1	N/A	2.7 sec
✓	☁️	exp7	20 hr #5	22 hr #4	53 sec
✗	☁️	exp72	N/A	20 hr #3	2 sec
...	☀️	maven-project	N/A	N/A	N/A
✗	☁️	MavenDemo	N/A	1 mo 23 days #2	25 sec
✓	☁️	webApp	1 mo 0 days #5	1 mo 0 days #4	11 sec

6. Go to Manage Jenkins and search for SonarQube Scanner for Jenkins and install it.(we already installed it for exp 7 so you can skip )

The Jenkins Plugins page shows the SonarQube Scanner plugin listed for installation:

Install	Name	Released
<input type="checkbox"/>	SonarQube Scanner 2.17.2	6 mo 29 days ago

Details for the SonarQube Scanner plugin:

- Version: 2.17.2
- Released: 6 mo 29 days ago
- Description: This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.

**Plugins****Download progress**

Updates

25

- Preparation
- Checking internet connectivity
  - Checking update center connectivity
  - Success

Available plugins

SonarQube Scanner

Success

Loading plugin extensions

Success

Installed plugins

Advanced settings

Download progress

[Go back to the top page](#)

(you can start using the installed plugins right away)

 Restart Jenkins when installation is complete and no jobs are running

7. Under Jenkins 'Manage Jenkins' then go to 'system', scroll and look for **SonarQube Servers** and enter the details. Enter the Server Authentication token if needed.(we dont need the token and this step was done in previous exp but jic)

In SonarQube installations: Under **Name** add <project name of sonarqube> for me **sonarqube\_exp8**

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

**Name**  
sonarqube\_exp8

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

**Server authentication token**  
SonarQube authentication token. Mandatory when anonymous access is disabled.  
- none -  
+ Add ▾

**Advanced** ▾

Add SonarQube

**Saved**

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

**Dashboard > Manage Jenkins > Tools > SonarQube Scanner** (i kept the default setting from last experiment and just changed the name.)

#### SonarQube Scanner installations

SonarQube Scanner installations ^ Edit

Add SonarQube Scanner

**SonarQube Scanner**

Name  
sonarqube\_scanner\_exp8

Install automatically ?

Install from Maven Central

Version  
SonarQube Scanner 6.2.0.4584

Add Installer ▾

Add SonarQube Scanner



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

SonarQube Scanner

Name  
sonarqube\_exp8

Install automatically ?

Install from Maven Central

Version  
SonarQube Scanner 6.2.0.4584

Add Installer ▾

Add SonarQube Scanner



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

## Enter an item name

adDevOps\_exp8

» Required field



### Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.



### Maven project

Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.



### Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.



### Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.



### Folder

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.



### Multibranch Pipeline

Creates a set of Pipeline projects according to detected branches in one SCM repository.



### Organization Folder

Creates a set of multibranch project subfolders by scanning for repositories.

OK

## 10. Under Pipeline script, enter the following:

```
node {  
    stage('Cloning the GitHub Repo') {  
        git 'https://github.com/shazforiot/GOL.git'  
    }  
  
    stage('SonarQube Analysis') {  
        withSonarQubeEnv('exp8') {  
            bat """  
                "C:\\Program Files\\Sonar Scanner\\sonar-scanner-6.2.0.4584-windows-x64\\bin\\sonar-scanner.bat" ^  
                -Dsonar.login=<username> ^  
                -Dsonar.password=<password> ^  
                -Dsonar.projectKey=<project-key> ^  
                -Dsonar.exclusions=vendor/*,resources/,./java ^  
                -Dsonar.host.url=http://127.0.0.1:9000/  
            """  
        }  
    }  
}
```

```
    }
}
}
```

\*Note that the code has placeholders

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

Definition

Pipeline script

```
Script ?  
1 node {  
2   stage('Cloning the GitHub Repo') {  
3     git 'https://github.com/shazforiot/GOL.git'  
4   }  
5  
6   stage('SonarQube Analysis') {  
7     withSonarQubeEnv('sonarqube_exp8') {  
8       bat """  
9         "C:\sonar-scanner-6.2.0.4584-windows-x64\bin\sonar-scanner.bat" ^  
10        -Dsonar.login=admin ^  
11        -Dsonar.password=2923 ^  
12        -Dsonar.projectKey=sonarqube ^  
13        -Dsonar.exclusions=vendor/**,resources/.java ^  
14        -Dsonar.host.url=http://127.0.0.1:9000/  
15        """  
16     }  
17   }  
18 }
```

Use Groovy Sandbox ?

Pipeline Syntax

Save Apply

## 11. Build project



## 12. Check console

Dashboard > adDevOps\_exp8 > #5

```

Status
Changes
Console Output
View as plain text
Edit Build Information
Delete build "#5"
Timings
Git Build Data
Pipeline Overview
Pipeline Console
Replay
Pipeline Steps
Workspaces
Previous Build
Next Build

```

Skipping 4,248 KB. [Full Log](#)

12:35:00.367 WARN Too many duplication references on file gameoflife-web/tools/Jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControloui.html for block at line 810. Keep only references.  
12:35:00.367 WARN Too many duplication references on file gameoflife-web/tools/Jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControloui.html for block at line 512. Keep only references.  
12:35:00.367 WARN Too many duplication references on file gameoflife-web/tools/Jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControloui.html for block at line 789. Keep only references.  
12:35:00.367 WARN Too many duplication references on file gameoflife-web/tools/Jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControloui.html for block at line 512. Keep only references.  
12:35:00.367 WARN Too many duplication references on file gameoflife-web/tools/Jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControloui.html for block at line 248. Keep only references.  
12:35:00.367 WARN Too many duplication references on file gameoflife-web/tools/Jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControloui.html for block at line 886. Keep only references.  
12:35:00.367 WARN Too many duplication references on file gameoflife-web/tools/Jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControloui.html for block at line 249. Keep only references.  
12:35:00.367 WARN Too many duplication references on file gameoflife-web/tools/Jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControloui.html for block at line 662. Keep only references.  
12:35:00.367 WARN Too many duplication references on file gameoflife-web/tools/Jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControloui.html for block at line 615. Keep only references.  
12:35:00.367 WARN Too many duplication references on file gameoflife-web/tools/Jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControloui.html for block at line 664. Keep only references.  
12:35:00.367 WARN Too many duplication references on file gameoflife-web/tools/Jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControloui.html for block at line 913. Keep only references.  
12:35:00.367 WARN Too many duplication references on file gameoflife-web/tools/Jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControloui.html for block at line 810. Keep only references.  
12:35:00.367 WARN Too many duplication references on file gameoflife-web/tools/Jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControloui.html for block at line 668. Keep only references.

## 13. Now, check the project in SonarQube

sonarqube / main

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

main

Quality Gate: **Passed** (Last analysis: 30 minutes ago)

New Code Overall Code

Security	Reliability	Maintainability
0 open issues (A)	68K open issues (C)	164K open issues (A)
0 H 0 M 0 L	0 H 47k M 21k L	7 H 143k M 21k L

Accepted issues	Coverage	Duplications
0 (D)	Coverage: 50.6% (On 0 lines to cover)	50.6% (On 750k lines)

Security Hotspots: 3 (E)

## 14. Code Problems

- Consistency

sonarqube / main

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

My Issues All

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: Security 0, Reliability 54k, Maintainability 164k

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

Insert a <!DOCTYPE> declaration to before this <html> tag. **Reliability**

Remove this deprecated "width" attribute. **Maintainability**

Remove this deprecated "align" attribute. **Maintainability**

Introducing Clean Code Attributes

Clean Code attributes are the characteristics that your code must have to be considered Clean Code.

You can now filter by these attributes to evaluate why your code is breaking away from being clean.

1 of 5 Next

L11 - 5min effort - 4 years ago - 0 Code Smell - 0 Major

- Intentionality

sonarqube

Projects Issues Rules Quality Profiles Quality Gates Administration More Q

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

My Issues All

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

- Security 0
- Reliability 14k
- Maintainability 15

> Severity ?

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

gameoflife-acceptance-tests/Dockerfile

- Use a specific version tag for the image. Intentionality Maintenance Open Not assigned L1 - 5min effort - 4 years ago - ⚡ Code Smell - ⚡ Major
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. Intentionality Maintenance Open Not assigned L12 - 5min effort - 4 years ago - ⚡ Code Smell - ⚡ Major
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. Intentionality Maintenance Open Not assigned L12 - 5min effort - 4 years ago - ⚡ Code Smell - ⚡ Major
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. Intentionality Maintenance Open Not assigned L12 - 5min effort - 4 years ago - ⚡ Code Smell - ⚡ Major

● Bugs

sonarqube

Projects Issues Rules Quality Profiles Quality Gates Administration More Q

Overview Issues Security Hotspots Measures Code Activity Project Se

Software Quality

- Security 0
- Reliability 14k
- Maintainability 0

> Severity ?

Type

- Bug 14k
- Vulnerability 0
- Code Smell 268

Add to selection Ctrl + click

Scope

Status

Security Category

Bulk Change Select issues Navigate to issue 13,619 issues 5

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

- Add "lang" and/or "xmtlang" attributes to this "<chimb>" element Intent Reliability accessibility wor Open Not assigned L1 - 2min effort - 4 years ago - ⚡ A Bug -
- Add "etho" headers to this "<table>". Intent Reliability accessibility wor Open Not assigned L9 - 2min effort - 4 years ago - ⚡ A Bug -

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

- Add "lang" and/or "xmtlang" attributes to this "<htmd>" element Intent Reliability accessibility wor Open Not assigned L1 - 2min effort - 4 years ago - ⚡ A Bug -
- Add "etho" headers to this "<table>". Intent Reliability accessibility wor Open Not assigned L1 - 2min effort - 4 years ago - ⚡ A Bug -

● Code Smells

sonarqube

Projects Issues Rules Quality Profiles Quality Gates Administration More Q

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

Software Quality

- Security 0
- Reliability 253
- Maintainability 15

> Severity ?

Type

- Bug 1dk
- Vulnerability 0
- Code Smell 268

Add to selection Ctrl + click

Scope

Status

Security Category

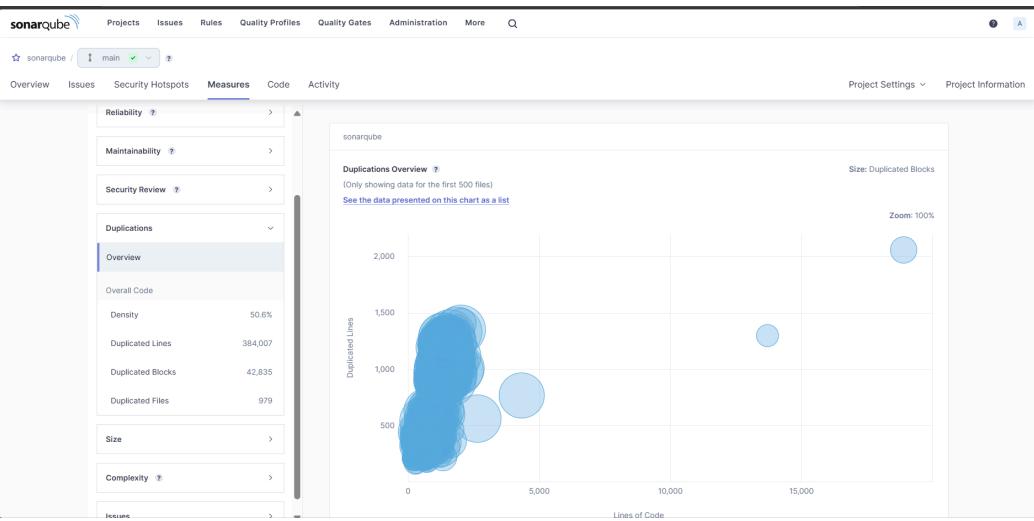
Bulk Change Select issues Navigate to issue 268 issues 2d 5h effort

gameoflife-acceptance-tests/Dockerfile

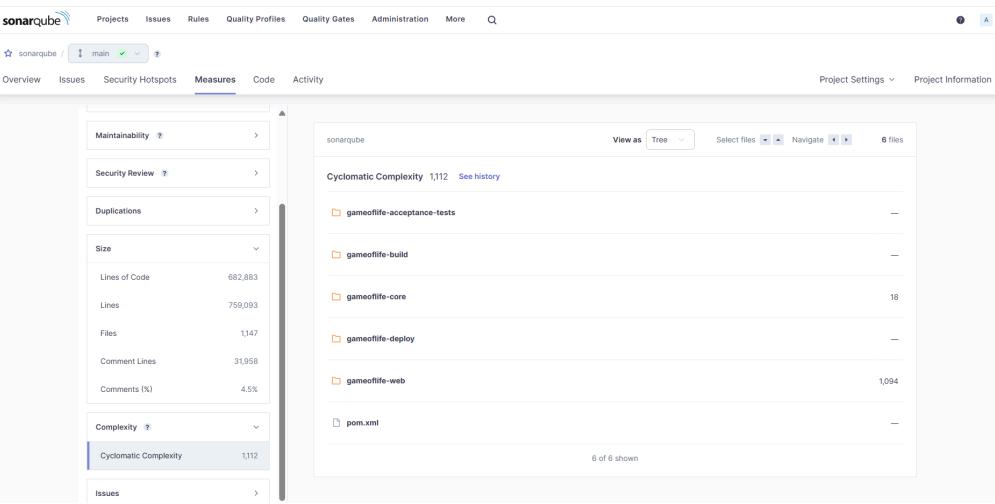
- Use a specific version tag for the image. Intentionality Maintenance Open Not assigned L1 - 5min effort - 4 years ago - ⚡ Code Smell - ⚡ Major
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. Intentionality Maintenance Open Not assigned L12 - 5min effort - 4 years ago - ⚡ Code Smell - ⚡ Major
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. Intentionality Maintenance Open Not assigned L12 - 5min effort - 4 years ago - ⚡ Code Smell - ⚡ Major
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. Intentionality Maintenance Open Not assigned L12 - 5min effort - 4 years ago - ⚡ Code Smell - ⚡ Major

● Duplications

Embedded database should be used for evaluation purposes only



## Cyclomatic Complexities



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

## Conclusion:

In this experiment, we successfully integrated Jenkins with SonarQube to automate continuous code quality monitoring within our CI/CD pipeline. This process involved deploying SonarQube via Docker, setting up a project for analysis, and configuring Jenkins with the SonarQube Scanner plugin. After configuring the necessary tools and adding SonarQube server details, we created a Jenkins pipeline that automates cloning from GitHub and running static analysis on the code. This integration allows us to detect potential bugs, code smells, and security vulnerabilities at every stage of development, ensuring improved code quality and streamlined development workflows.

# Steps

Launch an ec2 instance

Give name use the default OS

EC2 > Instances > Launch an instance

## Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

**Name and tags Info**

Name  
nagios\_host\_exp\_9kcs Add additional tags

**▼ Application and OS Images (Amazon Machine Image) Info**

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents Quick Start

Amazon Linux 	macOS 	Ubuntu 	Windows 	Red Hat 	S 
--	---	--	---	---	---

 **Browse more AMIs**  
Including AMIs from AWS, Marketplace and the Community

Make a key pair and use it.

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

▼ ⟳ [Create new key pair](#)

▼ Network settings [Info](#) Edit

Network [Info](#)  
vpc-07b6966cbfba88ee3

Subnet [Info](#)  
No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)  
Enable

**Additional charges apply** when outside of free tier allowance

Firewall (security groups) [Info](#)  
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

[Create security group](#)  [Select existing security group](#)

We'll create a new security group called '**launch-wizard-5**' with the following rules:

**Allow SSH traffic from**  
Helps you connect to your instance Anywhere ▼  
0.0.0.0/0

**Allow HTTPS traffic from the internet**

Note the name of the security group that was created for future use:  
here it is ' **launch-wizard-5** '

Success

Successfully initiated launch of instance ([i-0820376be204a7fcb](#))

▶ Launch log

Next Steps

go to security groups:

The screenshot shows the AWS EC2 Instances page. The left sidebar lists various EC2 services: EC2 Dashboard, EC2 Global View, Events, Console-to-Code (Preview), Instances (selected), Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations (New), Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, and Network & Security (Security Groups, Elastic IPs). The main content area displays the 'Instances (3)' section with a table:

	Name	Instance ID
<input type="checkbox"/>	Master	i-0ab175e9c60cc3a23
<input type="checkbox"/>	node-1	i-08ad30b7114767ca2
<input type="checkbox"/>	node-2	i-03c70d364fb762af5

Below the table, a section titled 'Select an instance' is visible.

click the security group id which was created while you created the ec2 instance of this experiment.

X Security Groups (9) [Info](#)

[C](#) Actions ▾ Export security groups to CSV ▾ [Create security group](#)

Find resources by attribute or tag

< 1 > [@](#)

<input type="checkbox"/>	Name	Security group ID	Security group name	VPC ID
<input type="checkbox"/>	-	<a href="#">sg-06013b4b74fb35de2</a>	launch-wizard-1	<a href="#">vpc-07b6966cbfba88e</a>
<input type="checkbox"/>	-	<a href="#">sg-00c39d8526dda67f7</a>	MasterGroup	<a href="#">vpc-07b6966cbfba88e</a>
<input type="checkbox"/>	-	<a href="#">sg-04987c373fb6884a0</a>	launch-wizard-2	<a href="#">vpc-07b6966cbfba88e</a>
<input type="checkbox"/>	aws-cloud9-Cloud9...	<a href="#">sg-00c10dc4d51f60c8a</a>	aws-cloud9-Cloud9-d788455f5a4d4b...	<a href="#">vpc-07b6966cbfba88e</a>
<input type="checkbox"/>	-	<a href="#">sg-0454b0a819cb08ef2</a>	launch-wizard-4	<a href="#">vpc-07b6966cbfba88e</a>
<input type="checkbox"/>	-	<a href="#">sg-05fa7fae7b41178e3</a>	default	<a href="#">vpc-07b6966cbfba88e</a>
<input type="checkbox"/>	-	<a href="#">sg-06ac4c5a9779ecaf9</a>	launch-wizard-5	<a href="#">vpc-07b6966cbfba88e</a>

now click on edit inbound rules

[Inbound rules](#) [Outbound rules](#) [Tags](#)

[C](#) Manage tags [Edit inbound rules](#)

Search

< 1 > [@](#)

<input type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol
<input type="checkbox"/>	-	<a href="#">sgr-0d6a171458e586b3e...</a>	IPv4	SSH	TCP

now do the following configurations:

Inbound rules [Info](#)

Security group rule ID	Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Source <a href="#">Info</a>	Description - optional <a href="#">Info</a>
sgr-0d6a171458e586b3e	SSH	TCP	22	Custom	<input type="text"/> 0.0.0.0/0 <a href="#">X</a>
-	HTTP	TCP	80	Anywhere...	<input type="text"/> ::/0 <a href="#">X</a>
-	All ICMP - IPv6	IPv6 ICMP	All	Anywhere...	<input type="text"/> ::/0 <a href="#">X</a>
-	HTTPS	TCP	443	Anywhere...	<input type="text"/> 0.0.0.0/0 <a href="#">X</a>
-	All traffic	All	All	Anywhere...	<input type="text"/> 0.0.0.0/0 <a href="#">X</a>
-	Custom TCP	TCP	5666	Anywhere...	<input type="text"/> 0.0.0.0/0 <a href="#">X</a>
-	All ICMP - IPv4	ICMP	All	Anywhere...	<input type="text"/> 0.0.0.0/0 <a href="#">X</a>

[Add rule](#)

by clicking "add rules"

then click on save rules.

EC2 > Security Groups > sg-06ac4c5a9779ecaf9 - launch-wizard-5

sg-06ac4c5a9779ecaf9 - launch-wizard-5

**Details**

Security group name <a href="#">launch-wizard-5</a>	Security group ID <a href="#">sg-06ac4c5a9779ecaf9</a>	Description <a href="#">launch-wizard-5 created 2024-09-28T03:55:31.506Z</a>	VPC ID <a href="#">vpc-07b6965chfb88ee5</a>
Owner <a href="#">209322483715</a>	Inbound rules count 7 Permission entries	Outbound rules count 1 Permission entry	

**Inbound rules (7)**

<input type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
<input type="checkbox"/>	-	sgr-05f6798fda0b60bb	IPv6	All ICMP - IPv6	IPv6 ICMP	All	::/0	-
<input type="checkbox"/>	-	sgr-0d6a171458e586...	IPv4	SSH	TCP	22	0.0.0.0/0	-
<input type="checkbox"/>	-	sgr-0b17ca9a9d9e6e3b5...	IPv4	HTTPS	TCP	443	0.0.0.0/0	-
<input type="checkbox"/>	-	sgr-0d3d582940a2ebaf0	IPv4	Custom TCP	TCP	5666	0.0.0.0/0	-
<input type="checkbox"/>	-	sgr-0e782e66d47b344f5	IPv4	All ICMP - IPv4	ICMP	All	0.0.0.0/0	-
<input type="checkbox"/>	-	sgr-00bb8da767cde375...	IPv6	HTTP	TCP	80	::/0	-
<input type="checkbox"/>	-	sgr-0c81dac37aa4a6020e	IPv4	All traffic	All		0.0.0.0/0	-

now navigate to instances, click on the instance which was created earlier and click on connect.

Instances (1/4) **Info**

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs
<input type="checkbox"/>	Master	i-0ab175e9c60cc3a23	<span>Running</span> <a href="#">Q</a> <a href="#">Q</a>	t2.micro	<span>2/2 checks passec</span> <a href="#">View alarms</a> <a href="#">+</a>	us-east-1b	ec2-54-165-203-193.co...	54.165.203.193	-	-	
<input type="checkbox"/>	node-1	i-08ad50b7114767ca2	<span>Running</span> <a href="#">Q</a> <a href="#">Q</a>	t2.micro	<span>2/2 checks passec</span> <a href="#">View alarms</a> <a href="#">+</a>	us-east-1b	ec2-52-23-200-179.co...	52.23.200.179	-	-	
<input type="checkbox"/>	node-2	i-03c70d364fb762af5	<span>Running</span> <a href="#">Q</a> <a href="#">Q</a>	t2.micro	<span>2/2 checks passec</span> <a href="#">View alarms</a> <a href="#">+</a>	us-east-1b	ec2-3-85-164-72.com...	3.85.164.72	-	-	
<input checked="" type="checkbox"/>	nagios_host_e...	i-0820376be204a7fc...	<span>Running</span> <a href="#">Q</a> <a href="#">Q</a>	t2.micro	<span>2/2 checks passec</span> <a href="#">View alarms</a> <a href="#">+</a>	us-east-1b	ec2-54-205-31-174.co...	54.205.31.174	-	-	

now copy the ssh command and just replace the .pem file with its actual location in your computer.

## Connect to instance Info

Connect to your instance i-0820376be204a7fcb (nagios\_host\_exp\_9kcs) using any of these options

EC2 Instance Connect | Session Manager | **SSH client** | EC2 serial console

Instance ID  
i-0820376be204a7fcb (nagios\_host\_exp\_9kcs)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is nagios\_exp\_9.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.  
chmod 400 "nagios\_exp\_9.pem"
4. Connect to your instance using its Public DNS:  
ec2-54-205-31-174.compute-1.amazonaws.com

Example:  
ssh -i "nagios\_exp\_9.pem" ec2-user@ec2-54-205-31-174.compute-1.amazonaws.com

ⓘ 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.

paste the command in your terminal and enter after replacing the .pem file with its actual location in your system.

```
C:\Users\Lenovo>ssh -i "C:\Users\Lenovo\Downloads\nagios_exp_9.pem" ec2-user@ec2-54-205-31-174.compute-1.amazonaws.com
The authenticity of host 'ec2-54-205-31-174.compute-1.amazonaws.com (54.205.31.174)' can't be established.
ED25519 key fingerprint is SHA256:+oIS6lcV6qE12x8gFgYVvMsB+yc9vN7UEpF6oBt0jw0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-205-31-174.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

      _#
      _###_
      Amazon Linux 2023
      _#####
      \###|
      \#/ __ https://aws.amazon.com/linux/amazon-linux-2023
      V~'__>
      ~~' /'
      ~~' /'
      _/m' /'
[ec2-user@ip-172-31-80-137 ~]$ |
```

now paste the following commands in your connected terminal:

`sudo yum update`

```
~/m/
[ec2-user@ip-172-31-80-137 ~]$ sudo yum update
Last metadata expiration check: 2:21:45 ago on Sat Sep 28 03:59:04 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-80-137 ~]$ |
```

sudo yum install httpd php

```
[ec2-user@ip-172-31-80-137 ~]$ sudo yum install httpd php
Last metadata expiration check: 2:22:53 ago on Sat Sep 28 03:59:04 2024.
Dependencies resolved.
=====
Package           Architecture   Version      Repository
=====
Installing:
httpd            x86_64        2.4.62-1.amzn2023
php8.3           x86_64        8.3.10-1.amzn2023.0.1
=====
Installing dependencies:
apr              x86_64        1.7.2-2.amzn2023.0.2
apr-util         x86_64        1.6.3-1.amzn2023.0.1
generic-logos-httpd    noarch      18.0.0-12.amzn2023.0.3
httpd-core       x86_64        2.4.62-1.amzn2023
httpd-filesystem  noarch      2.4.62-1.amzn2023
httpd-tools      x86_64        2.4.62-1.amzn2023
=====
(type y when prompted)
```

sudo yum install gcc glibc glibc-common

```
Dependencies resolved.
=====
Package           Architecture   Version      Repository
=====
Installing:
gcc              x86_64        11.4.1-2.amzn2023.0.2
=====
Installing dependencies:
annobin-docs     noarch      10.93-1.amzn2023.0.1
annobin-plugin-gcc x86_64        10.93-1.amzn2023.0.1
cpp              x86_64        11.4.1-2.amzn2023.0.2
gc               x86_64        8.0.4-5.amzn2023.0.2
glibc-devel      x86_64        2.34-52.amzn2023.0.11
glibc-headers-x86  noarch      2.34-52.amzn2023.0.11
guile22          x86_64        2.2.7-2.amzn2023.0.3
kernel-headers   x86_64        6.1.109-118.189.amzn2023
libmpc           x86_64        1.2.1-2.amzn2023.0.2
libtool-ltdl     x86_64        2.4.7-1.amzn2023.0.3
libxcrypt-devel  x86_64        4.4.33-7.amzn2023
make             x86_64        1:4.3-5.amzn2023.0.2
=====
Transaction Summary
=====
Install 13 Packages

Total download size: 52 M
Installed size: 168 M
Is this ok [y/N]: y|
```

sudo yum install gd gd-devel

```
google-noto-sans-vf-fonts-20201206-2.amzn2023.0.2.noarch
graphite2-devel-1.3.14-7.amzn2023.0.2.x86_64
harfbuzz-7.0.0-2.amzn2023.0.1.x86_64
jbigkit-libs-2.1-21.amzn2023.0.2.x86_64
libICE-1.0.10-6.amzn2023.0.2.x86_64
libX11-1.7.2-3.amzn2023.0.4.x86_64
libX11-devel-1.7.2-3.amzn2023.0.4.x86_64
libXau-1.0.9-6.amzn2023.0.2.x86_64
libXext-1.3.4-6.amzn2023.0.2.x86_64
libXpm-devel-3.5.15-2.amzn2023.0.3.x86_64
libXt-1.2.0-4.amzn2023.0.2.x86_64
libffi-devel-3.4-4.amzn2023.0.1.x86_64
libicu-devel-67.1-7.amzn2023.0.3.x86_64
libjpeg-turbo-devel-2.1.4-2.amzn2023.0.5.x86_64
libpng-2:1.6.37-10.amzn2023.0.6.x86_64
libselinux-devel-3.4-5.amzn2023.0.2.x86_64
libtiff-4.4.0-4.amzn2023.0.18.x86_64
libwebp-1.2.4-1.amzn2023.0.6.x86_64
libxcb-1.13.1-7.amzn2023.0.2.x86_64
libxml2-devel-2.10.4-1.amzn2023.0.6.x86_64
pcre2-utf16-10.40-1.amzn2023.0.3.x86_64
pixman-0.40.0-3.amzn2023.0.3.x86_64
xml-common-0.6.3-56.amzn2023.0.2.noarch
xz-devel-5.2.5-9.amzn2023.0.2.x86_64
graphite2-1.3.14-7.amzn2023.0.2.x86_64
harfbuzz-7.0.0-2.amzn2023.0.1.x86_64
langpacks-core-font-en-3.0-21.amzn2023.0.4.noarch
libSM-1.2.3-8.amzn2023.0.2.x86_64
libX11-common-1.7.2-3.amzn2023.0.4.noarch
libX11-xcb-1.7.2-3.amzn2023.0.4.x86_64
libXau-devel-1.0.9-6.amzn2023.0.2.x86_64
libXpm-3.5.15-2.amzn2023.0.3.x86_64
libXrender-0.9.10-14.amzn2023.0.2.x86_64
libblkid-devel-2.37.4-1.amzn2023.0.4.x86_64
libicu-67.1-7.amzn2023.0.3.x86_64
libjpeg-turbo-2.1.4-2.amzn2023.0.5.x86_64
libmount-devel-2.37.4-1.amzn2023.0.4.x86_64
libpng-devel-2:1.6.37-10.amzn2023.0.6.x86_64
libsepol-devel-3.4-3.amzn2023.0.3.x86_64
libtiff-devel-4.4.0-4.amzn2023.0.18.x86_64
libwebp-devel-1.2.4-1.amzn2023.0.6.x86_64
libxcb-devel-1.13.1-7.amzn2023.0.2.x86_64
pcre2-devel-10.40-1.amzn2023.0.3.x86_64
pcre2-utf32-10.40-1.amzn2023.0.3.x86_64
sysprof-capture-devel-3.40.1-2.amzn2023.0.2.x86_64
xorg-x11proto-devel-2021.4-1.amzn2023.0.2.noarch
zlib-devel-1.2.11-33.amzn2023.0.5.x86_64
```

```
complete!
[ec2-user@ip-172-31-80-137 ~]$
```

sudo adduser -m nagios

sudo passwd nagios

```
Complete!
[ec2-user@ip-172-31-80-137 ~]$ sudo adduser -m nagios
sudo passwd nagios
Changing password for user nagios.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-80-137 ~]$ |
```

( add a password here)

sudo groupadd nagcmd

```
[ec2-user@ip-172-31-80-137 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-80-137 ~]$ |
```

sudo usermod -a -G nagcmd nagios

sudo usermod -a -G nagcmd apache

```
[ec2-user@ip-172-31-80-137 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-80-137 ~]$ sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
[ec2-user@ip-172-31-80-137 ~]$ |
```

mkdir ~/downloads

cd ~/downloads

```
[ec2-user@ip-172-31-80-137 ~]$ mkdir ~/downloads  
cd ~/downloads  
[ec2-user@ip-172-31-80-137 downloads]$ |
```

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

```
cd ~/downloads  
[ec2-user@ip-172-31-80-137 downloads]$ wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz  
--2024-09-28 06:27:51-- https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz  
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 2600:3c00::f03c:92ff:fe7:45ce  
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 2065473 (2.0M) [application/x-gzip]  
Saving to: 'nagios-4.5.5.tar.gz'  
  
nagios-4.5.5.tar.gz      100%[=====] 1.97M 5.30MB/s    in 0.4s  
2024-09-28 06:27:52 (5.30 MB/s) - 'nagios-4.5.5.tar.gz' saved [2065473/2065473]  
[ec2-user@ip-172-31-80-137 downloads]$ |
```

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

```
[ec2-user@ip-172-31-80-137 downloads]$ wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz  
--2024-09-28 06:28:14-- https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz  
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251  
Connecting to nagios-plugins.org (nagios-plugins.org)|45.56.123.251|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 2753049 (2.6M) [application/x-gzip]  
Saving to: 'nagios-plugins-2.4.11.tar.gz'  
  
nagios-plugins-2.4.11.tar.gz 100%[=====] 2.62M 5.90MB/s    in 0.4s  
2024-09-28 06:28:15 (5.90 MB/s) - 'nagios-plugins-2.4.11.tar.gz' saved [2753049/2753049]  
[ec2-user@ip-172-31-80-137 downloads]$ |
```

tar zxvf nagios-4.5.5.tar.gz

```
[ec2-user@ip-172-31-80-137 downloads]$ tar zxvf nagios-4.5.5.tar.gz  
nagios-4.5.5/  
nagios-4.5.5/.github/  
nagios-4.5.5/.github/workflows/  
nagios-4.5.5/.github/workflows/test.yml  
nagios-4.5.5/.gitignore  
nagios-4.5.5/CONTRIBUTING.md  
nagios-4.5.5/Changelog  
nagios-4.5.5/INSTALLING  
nagios-4.5.5/LEGAL  
nagios-4.5.5/LICENSE  
nagios-4.5.5/Makefile.in
```

---

Now we have to first navigate to the nagios-4.5.5 folder in downloads.

- commands to enter:

ls (verify whether nagios-4.5.5 exists)

```
nagiosexp9 x + v
[ec2-user@ip-172-31-80-137 downloads]$ ls
nagios-4.5.5 nagios-4.5.5.tar.gz nagios-plugins-2.4.11.tar.gz
[ec2-user@ip-172-31-80-137 downloads]$ |
```

```
cd nagios-4.5.5
nagios-4.5.5 nagios-4.5.5.tar.gz nagios-plugins-2.4.11.
[ec2-user@ip-172-31-80-137 downloads]$ cd nagios-4.5.5
[ec2-user@ip-172-31-80-137 nagios-4.5.5]$ |
```

we now have to install openssl dev library

commands to enter:

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

```
[ec2-user@ip-172-31-80-137 nagios-4.5.5]$ sudo yum install openssl-devel
Last metadata expiration check: 2:31:25 ago on Sat Sep 28 03:59:04 2024.
Dependencies resolved.
=====
 Package           Architecture      Version       Repository      Size
 =====
 Installing:
 openssl-devel     x86_64          1:3.0.8-1.amzn2023.0.14      amazonlinux    3.0 M
 Transaction Summary
 =====
 Install 1 Package

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

```
Total                                         18 MB/s | 3.0 MB   00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing          :                                1/1
  Installing         : openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64 1/1
  Running scriptlet: openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64 1/1
  Verifying          : openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64 1/1

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

Complete!
[ec2-user@ip-172-31-80-137 nagios-4.5.5]$ |
```

---

Then finally we can run the commands like usual.

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

```
nagiosexp9 x + v
[ec2-user@ip-172-31-80-137 nagios-4.5.5]$ ./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether the compiler supports GNU C... yes
checking whether gcc accepts -g... yes
checking for gcc option to enable C11 features... none needed
checking whether make sets $(MAKE)... yes
checking whether ln -s works... yes
checking for strip... /usr/bin/strip
checking for sys/wait.h that is POSIX.1 compatible... yes
checking for stdio.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
```

make all

```
[ec2-user@ip-172-31-80-137 nagios-4.5.5]$ make all
cd ./base && make
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
gcc -Wall -I.. -I./lib -I.. ./include -I.. -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nagios.o ./nagios.c
gcc -Wall -I.. -I./lib -I.. ./include -I.. -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o broker.o broker.c
gcc -Wall -I.. -I.. -I./lib -I.. ./include -I.. -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nebmods.o nebmod
gcc -Wall -I.. -I.. -I./lib -I.. ./include -I.. -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o ./common/shared
./common/shared.c
gcc -Wall -I.. -I.. -I./lib -I.. ./include -I.. -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o query-handler.o
query-handler.c
gcc -Wall -I.. -I.. -I./lib -I.. ./include -I.. -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o workers.o worker
In function `get_wproc_list',
      what version of the plugins you are using
      - Relevant snippets from your config files
      - Relevant error messages from the Nagios log file
```

For more information on obtaining support for Nagios, visit:

<https://support.nagios.com>

\*\*\*\*\*

Enjoy.

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

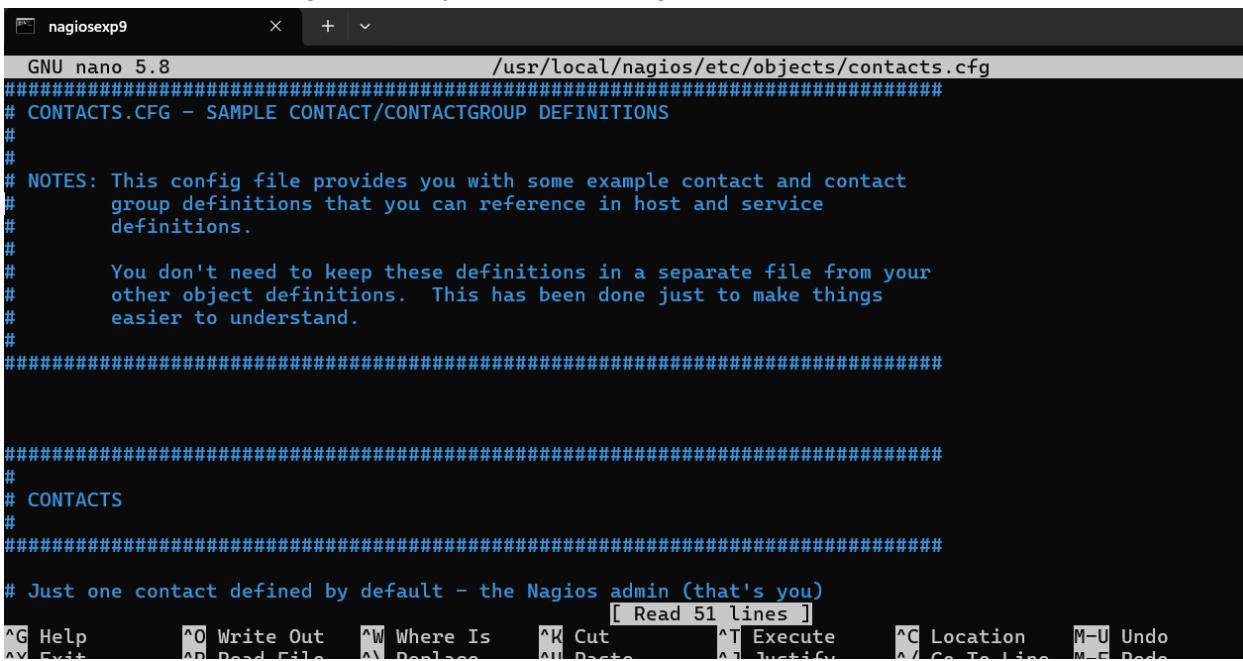
```
[ec2-user@ip-172-31-80-137 nagios-4.5.5]$ sudo make install
sudo make install-init
sudo make install-config
sudo make install-commandmode
cd ./base && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagios /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagiostats /usr/local/nagios/bin
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/base'
cd ./cgi && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
make install-basic
make[2]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/sbin
for file in *.cgi; do \

```

**Now the next command will take us to nano editor:**

---

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



```
GNU nano 5.8                               /usr/local/nagios/etc/objects/contacts.cfg
#####
# CONTACTS.CFG - SAMPLE CONTACT/CONTACTGROUP DEFINITIONS
#
#
# NOTES: This config file provides you with some example contact and contact
# group definitions that you can reference in host and service
# definitions.
#
# You don't need to keep these definitions in a separate file from your
# other object definitions. This has been done just to make things
# easier to understand.
#
#####
#
# CONTACTS
#
#####
# Just one contact defined by default - the Nagios admin (that's you)
[ Read 51 lines ]
^G Help      ^O Write Out    ^W Where Is    ^K Cut        ^T Execute    ^C Location   M-U Undo
^V Exit      ^P Read File    ^R Replace     ^U Paste      ^L Justify    ^I Go To Line M-F Redo
```

navigate down to email: and change it to your email address.

```
# 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 *****

#####

```

```
# 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
    alias             Nagios Admin        ; Full name of user
    email            2022.shubham.jha@ves.ac.in| ; <***** CHANGE THIS TO YOUR EMAIL ADDRESS *****
}

#####
#
# CONTACT GROUPS

press Ctrl+O and then enter.
then press Ctrl +X
```

chmod g+s /usr/local/nagios/var/rw

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

[ec2-user@ip-172-31-80-137 nagios-4.5.5]$ sudo nano /usr/local/nagios/etc/objects/contacts.cfg
[ec2-user@ip-172-31-80-137 nagios-4.5.5]$ |
```

---

```
sudo make install-webconf
```

```
[ec2-user@ip-172-31-80-137 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-172-31-80-137 nagios-4.5.5]$ |
```

## Adding password for nagios admin

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

```
[ec2-user@ip-172-31-80-137 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-172-31-80-137 nagios-4.5.5]$ |
```

```
sudo service httpd restart
```

```
[ec2-user@ip-172-31-80-137 nagios-4.5.5]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-80-137 nagios-4.5.5]$ |
```

cd ~/downloads

tar zxvf nagios-plugins-2.4.11.tar.gz

```
[ec2-user@ip-172-31-80-137 downloads]$ cd ~/downloads
tar zxvf nagios-plugins-2.4.11.tar.gz
nagios-plugins-2.4.11/
nagios-plugins-2.4.11/build-aux/
nagios-plugins-2.4.11/build-aux/compile
nagios-plugins-2.4.11/build-aux/config.guess
nagios-plugins-2.4.11/build-aux/config.rpath
nagios-plugins-2.4.11/build-aux/config.sub
nagios-plugins-2.4.11/build-aux/install-sh
nagios-plugins-2.4.11/build-aux/ltmain.sh
nagios-plugins-2.4.11/build-aux/missing
nagios-plugins-2.4.11/build-aux/mkinstalldirs
nagios-plugins-2.4.11/build-aux/depcomp
nagios-plugins-2.4.11/build-aux/snippet/
```

cd nagios-plugins-2.4.11

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

```
nagiosexp9  × + ▾
[ec2-user@ip-172-31-80-137 downloads]$ cd nagios-plugins-2.4.11
./configure --with-nagios-user=nagios --with-nagios-group=nagios
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether to enable maintainer-specific portions of Makefiles... yes
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether gcc understands -c and -o together... yes
checking whether make supports the include directive... yes (GNU style)
```

make

sudo make install

```
[ake[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/plugins-root'
ake[1]: Making install in po
ake[1]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/po'
usr/bin/mkdir -p /usr/local/nagios/share
nstalling fr.gmo as /usr/local/nagios/share/locale/fr/LC_MESSAGES/nagios-plugins.mo
nstalling de.gmo as /usr/local/nagios/share/locale/de/LC_MESSAGES/nagios-plugins.mo
f test "nagios-plugins" = "gettext-tools"; then \
/usr/bin/mkdir -p /usr/local/nagios/share/gettext/po; \
for file in Makefile.in.in remove-potcdate.sin      Makevars.template; do \
/usr/bin/install -c -o nagios -g nagios -m 644 ./${file} \
/usr/local/nagios/share/gettext/po/${file}; \
done; \
for file in Makevars; do \
rm -f /usr/local/nagios/share/gettext/po/${file}; \
done; \
lse \
:; \
i
ake[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/po'
ake[1]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
ake[2]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
ake[2]: Nothing to be done for 'install-exec-am'.
ake[2]: Nothing to be done for 'install-data-am'.
ake[2]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
ake[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
```

sudo chkconfig --add nagios

sudo chkconfig nagios on

```
[ec2-user@ip-172-31-80-137 nagios-plugins-2.4.11]$ sudo chkconfig --add nagios
sudo chkconfig nagios on
error reading information on service nagios: No such file or directory
Note: Forwarding request to 'systemctl enable nagios.service'.
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service → /usr/lib/systemd/system/nagios.service.
[ec2-user@ip-172-31-80-137 nagios-plugins-2.4.11]$ |
```

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

```
[ec2-user@ip-172-31-80-137 nagios-plugins-2.4.11]$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
error reading information on service nagios: no such file or directory
Note: Forwarding request to 'systemctl enable nagios.service'.
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service → /usr/lib/systemd/system/nagios.service.
[ec2-user@ip-172-31-80-137 nagios-plugins-2.4.11]$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

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

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

Running pre-flight check on configuration data...

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

---

**If this command is giving error! (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!)

**The solution:**

**Create the missing directory, set the permissions, verify it.**

```
sudo mkdir -p /usr/local/nagios/var/spool/checkresults      (this is for creation)
sudo chown nagios:nagios /usr/local/nagios/var/spool/checkresults
sudo chmod 775 /usr/local/nagios/var/spool/checkresults      (this is for permissions)
```

---

**Now rerun the commmad (also given below) and continue:**

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

sudo service nagios start

```
Things look okay - No serious problems were detected during the pre-flight check
[ec2-user@ip-172-31-80-137 nagios-plugins-2.4.11]$ sudo service nagios start
Redirecting to /bin/systemctl start nagios.service
[ec2-user@ip-172-31-80-137 nagios-plugins-2.4.11]$ |
```

---

sudo systemctl status nagios

```
[ec2-user@ip-172-31-80-137 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 Sat 2024-09-28 07:40:16 UTC; 35s ago
     Docs: https://www.nagios.org/documentation
  Process: 71009 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
  Process: 71010 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 71011 (nagios)
    Tasks: 6 (limit: 1112)
   Memory: 5.6M
      CPU: 82ms
     CGroup: /system.slice/nagios.service
             ├─71011 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─71012 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─71013 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─71014 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─71015 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─71016 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 28 07:40:16 ip-172-31-80-137.ec2.internal nagios[71011]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfully initialized
Sep 28 07:40:16 ip-172-31-80-137.ec2.internal nagios[71011]: qh: core query handler registered
Sep 28 07:40:16 ip-172-31-80-137.ec2.internal nagios[71011]: qh: echo service query handler registered
Sep 28 07:40:16 ip-172-31-80-137.ec2.internal nagios[71011]: qh: help for the query handler registered
Sep 28 07:40:16 ip-172-31-80-137.ec2.internal nagios[71011]: wproc: Successfully registered manager as @wproc with query handler
Sep 28 07:40:16 ip-172-31-80-137.ec2.internal nagios[71011]: wproc: Registry request: name=Core Worker 71015;pid=71015
Sep 28 07:40:16 ip-172-31-80-137.ec2.internal nagios[71011]: wproc: Registry request: name=Core Worker 71014;pid=71014
Sep 28 07:40:16 ip-172-31-80-137.ec2.internal nagios[71011]: wproc: Registry request: name=Core Worker 71013;pid=71013
Sep 28 07:40:16 ip-172-31-80-137.ec2.internal nagios[71011]: wproc: Registry request: name=Core Worker 71012;pid=71012
Sep 28 07:40:17 ip-172-31-80-137.ec2.internal nagios[71011]: Successfully launched command file worker with pid 71016
lines 1-28/28 (END)
```

(ignore if no error was found)

**Again if this is giving an error then it is primarily because Nagios monitoring tool is unable to create or write to a temporary file in the "/usr/local/nagios/var/"**

**To debug it lets start by  
checking the permissions:**

ls -ld /usr/local/nagios/var

**Changing the ownership**

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

**Modify permissions**

sudo chmod -R 755 /usr/local/nagios/var

**Restart Nagios service**

sudo systemctl restart nagios

**check status of nagios, Rerun the command**

**(the command which gave the recent error)**

sudo systemctl status nagios

Now, go to EC2 instance and click on instance id. Then, click on the copy icon just before the public ip address on public IP.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with various navigation options like EC2 Dashboard, Events, Instances, Launch Templates, and Capacity Reservations. The main area displays an instance summary for 'i-0820376be204a7fcb (nagios host exp\_9kcs)'. Key details shown include:

- Instance ID: i-0820376be204a7fcb
- IPv6 address: -
- Hostname type: IP name: ip-172-31-80-137.ec2.internal
- Answer private resource DNS name: IPv4 (A)
- Auto-assigned IP address: 54.224.175.95 [Public IP]
- IAM Role: -
- IMDSv2 Required
- VPC ID: vpc-07b6966cbfa8ee3
- Subnet ID: subnet-029be9bc13a1f9f65
- Instance ARN: arnaws:ec2:us-east-1:209322483715:instance/i-0820376be204a7fcb

A tooltip on the right side of the screen highlights the copied Public IPv4 address (54.224.175.95). Other visible information includes Private IP4 addresses (172.31.80.137), Public IP4 DNS (ec2-54-224-175-95.compute-1.amazonaws.com), and AWS Compute Optimizer findings.

Enter the username password set above. (in the section of adding password for nagios admin)

The screenshot shows a web browser window with the URL '54.224.175.95/nagios/' in the address bar. A modal dialog box is displayed, prompting for login credentials:

**Sign in to access this site**  
Authorization required by http://54.224.175.95  
Your connection to this site is not secure

Username:  Password:

Buttons: Sign in | Cancel

The screenshot shows the Nagios Core web interface. The left sidebar contains a navigation menu with sections like General, Current Status, Problems, Reports, and System. The main content area features the Nagios Core logo and a message indicating the daemon is running with PID 3152. It also displays the version (Version 4.5.5), the date (September 17, 2024), and a link to check for updates. Below this are two boxes: 'Get Started' with a list of monitoring steps and 'Quick Links' with links to various Nagios resources. At the bottom, there are sections for 'Latest News' and 'Don't Miss...', both currently empty. A copyright notice at the very bottom states: 'Copyright © 2010-2024 Nagios Core Development Team and Community Contributors. Copyright © 1999-2009 Ethan Galstad. See the THANKS file for more information on contributors.'

## Conclusion:

Setting up Nagios on an EC2 instance was a rewarding yet challenging experience for me. I began by launching an instance using the default operating system and configuring it to monitor my network. The installation process went smoothly at first; I installed essential packages, created users, and configured Nagios as planned.

However, I encountered a few hurdles along the way. One significant issue arose when the Apache server was not running, which prevented me from accessing the Nagios web interface. After some troubleshooting, I realized that restarting the Apache service was necessary to resolve this.

Additionally, I faced permission issues that initially hindered Nagios from creating or writing to temporary files. By checking the ownership and permissions of the necessary directories, I managed to address this issue effectively.

Here in the document nagios host machine (interchangibly also referred as exp9 machine or host machine) refers to the instance which was connected to the terminal in previous experiment.

(so if the previous instance was closed do connect with that instance and run the httpd status command to check whether the apache server was closed. if its closed run the start httpd command (google it or use ctrl+f to search for the key word in previous doc).)

And the client machine refers to the machine created just for this experiment.

# Steps

## 1) Launch an instance

Launch an ec2 instance.

Select Ubuntu as the os give a meaningful name of the instance.

The screenshot shows the 'Launch an instance' wizard. In the 'Name and tags' step, the name 'exp10client' is entered. In the 'Application and OS Images (Amazon Machine Image)' step, the 'Ubuntu' option is selected. A search bar at the top right of this section is empty. Below the search bar, there are tabs for 'Recents' and 'Quick Start'. Under 'Quick Start', there are icons for Amazon Linux, macOS, Ubuntu (selected), Windows, Red Hat, and SUSE Linux. To the right of these icons is a search icon and a link to 'Browse more AMIs'. At the bottom left of this section, it says 'Amazon Machine Image (AMI)'. On the far right, a summary panel shows the following details:

Summary	
Number of instances	1
Software Image (AMI)	Canonical, Ubuntu, 24.04, ami-0e86e20dae9224db8
Virtual server type (instance)	t2.micro
Firewall (security group)	launch-wizard-5
Storage (volumes)	1 volume(s) - 8 GiB

Select the same security group as given to the exp9 machine.

▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Search our full catalog including 1000s of application and OS images

Recents Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Linux

Browse more AMIs Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type  
ami-0e86e20dae9224db8 (64-bit (x86)) / ami-096ea6a12ea24a797 (64-bit (Arm))  
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

Description  
Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Architecture 64-bit (x86) AMI ID ami-0e86e20dae9224db8 Username ubuntu Verified provider

▼ Summary

Number of instances 1

Software Images Canonical, Ubuntu ami-0e86e20dae9224db8

Virtual server type t2.micro

Firewall (security groups) launch-wizard-1

Storage (volumes) 1 volume(s) - 8

Free tier eligible 750 hours in the Region are available for this tier AMI. Public IP address per month, up to 1 million IP addresses. 100 GB of internet bandwidth per month.

Cancel

Make sure to select the same key-pair login used in the exp9 machine.

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required  
 nagios\_exp\_9 [Create new key pair](#)

▼ Network settings [Info](#) Edit

Network [Info](#)  
vpc-07b6966cbfba88ee3

Subnet [Info](#)  
No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)  
Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) [Info](#)  
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group  Select existing security group

Common security groups [Info](#)  
 Select security groups

click on launch instance.

Now connect with this client machine using the ssh through your terminal(open a new terminal in your local machine and we will need both of the terminals open)

Instances (1/5) <a href="#">Info</a>							
				Last updated 2 minutes ago		<a href="#">Connect</a>	<a href="#">Instance state ▾</a>
<input type="text"/> Find Instance by attribute or tag (case-sensitive)		<a href="#">All states ▾</a>					
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
Master	i-0ab175e9c60cc3a23	<span>Running</span>	t2.micro	<span>2/2 checks passed</span>	<a href="#">View alarms</a>	us-east-1b	ec2-3-82-156-160.com...
node-1	i-08ad30b7114767ca2	<span>Running</span>	t2.micro	<span>2/2 checks passed</span>	<a href="#">View alarms</a>	us-east-1b	ec2-3-85-110-80.comp...
node-2	i-03c70d364fb762af5	<span>Running</span>	t2.micro	<span>2/2 checks passed</span>	<a href="#">View alarms</a>	us-east-1b	ec2-54-226-209-38.co...
nagios_host_e...	i-0820376be204a7fcf	<span>Running</span>	t2.micro	<span>2/2 checks passed</span>	<a href="#">View alarms</a>	us-east-1b	ec2-54-224-175-95.co...
exp10client	i-0994ca5a178801a54	<span>Running</span>	t2.micro	<span>Initializing</span>	<a href="#">View alarms</a>	us-east-1b	ec2-54-173-58-143.co...

[EC2](#) > [Instances](#) > [i-0994ca5a178801a54](#) > [Connect to instance](#)

## Connect to instance [Info](#)

Connect to your instance i-0994ca5a178801a54 (exp10client) using any of these options

[EC2 Instance Connect](#) [Session Manager](#) [SSH client](#) [EC2 serial console](#)

Instance ID  
 [i-0994ca5a178801a54 \(exp10client\)](#)

1. Open an SSH client.  
2. Locate your private key file. The key used to launch this instance is nagios\_exp\_9.pem  
3. Run this command, if necessary, to ensure your key is not publicly viewable.  
 `chmod 400 "nagios_exp_9.pem"`  
4. Connect to your instance using its Public DNS:  
 [ec2-54-173-58-143.compute-1.amazonaws.com](#)

**Command copied**

`ssh -i "nagios_exp_9.pem" ubuntu@ec2-54-173-58-143.compute-1.amazonaws.com`

**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](#)

Note to change the path of the .pem file.

```

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\Lenovo> ssh -i "C:\Users\Lenovo\Downloads\nagios_exp_9.pem" ubuntu@ec2-54-173-58-143.compute-1.amazonaws.com
The authenticity of host 'ec2-54-173-58-143.compute-1.amazonaws.com (54.173.58.143)' can't be established.
ED25519 key fingerprint is SHA256:IA3XH7f011spK084wDcZFmqRgNn0iJZ7itI2pBMmHP4.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-173-58-143.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
 * Support:        https://ubuntu.com/pro

System information as of Sat Sep 28 10:43:28 UTC 2024

System load:  0.01      Processes:          107
Usage of /:   22.8% of 6.71GB  Users logged in:    0
Memory usage: 19%           IPv4 address for enx0: 172.31.82.77

```

## 2) Go to nagios host machine (Host machine)

Perform the following commands

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

```

[ec2-user@ip-172-31-80-137 ~]$ ps -ef | grep nagios
nagios  3152      1  0 08:36 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios  3153      3152  0 08:36 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios  3154      3152  0 08:36 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios  3155      3152  0 08:36 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios  3156      3152  0 08:36 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios  3160      3152  0 08:36 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
ec2-user 11528     2972  0 10:44 pts/0    00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-80-137 ~]$ |

```

```
sudo su
```

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

```
[root@ip-172-31-80-137 ec2-user]# mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-80-137 ec2-user]# ls
```

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

```
[root@ip-172-31-80-137 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
```

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

```
[root@ip-172-31-80-137 ec2-user]# nano /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
```

Change hostname and alias to linuxserver

Change address to public ip address of client instance (Ubuntu instance) you can get

the ip address by clicking on the instance id on the instances section there you will get the public ipv4 address

Instance summary for i-0994ca5a178801a54 (exp10client) Updated less than a minute ago

Instance ID: i-0994ca5a178801a54 (exp10client)

IPv6 address: -

Hostname type: IP name: ip-172-31-82-77.ec2.internal

Answer private resource DNS name: IPv4 (A)

Auto-assigned IP address: -

Public IP address: 54.173.58.143 | open address

Instance state: Running

Private IP DNS name (IPv4 only): ip-172-31-82-77.ec2.internal

Instance type: t2.micro

VPC ID: -

Private IPv4 addresses: 172.31.82.77

Public IPv4 DNS: ec2-54-173-58-143.compute-1.amazonaws.com | open address

Elastic IP addresses: -

AWS Compute Optimizer finding: -

```
# HOST DEFINITION
#####
# Define a host for the local machine
define host {
    use          linux-server           ; Name of host template to use
    host_name    linuxserver           ; This host definition will inherit
    alias        linuxserver           ; in (or inherited by) the linux-
    address      54.173.58.143         ; server template
}
```

Change hostgroup\_name to linux-servers1

```
# 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
}
```

Change the occurrences of hostname further in the document from localhost to linuxserver example like:

host_name	localhost
service_descriptions	BING

changed to

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

This is the last one

```
define service {  
    use local-service ; Name of service template to >  
    host_name linuxserver  
    service_description HTTP  
    check_command check_http  
    notifications_enabled 0
```

now **ctrl+O** and **enter** to save and then **ctrl+X** for exiting.

Open nagios configuration file and add the line shown below  
**nano /usr/local/nagios/etc/nagios.cfg**

```
[root@ip-172-31-80-137 ec2-user]# nano /usr/local/nagios/etc/nagios.cfg
```

##Add this line below the opened nano interface where similar lines are commented.  
**cfg\_dir=/usr/local/nagios/etc/objects/monitorhosts/**

```

GNU nano 5.8                               /usr/local/nagios/etc/nagios.cfg
# These are the object configuration files in which you define hosts,
# host groups, contacts, contact groups, services, etc.
# You can split your object definitions across several config files
# if you wish (as shown below), or keep them all in a single config file.

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

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

# Definitions for monitoring a Windows machine
:cfg_file=/usr/local/nagios/etc/objects/windows.cfg

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

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

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

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

# OBJECT CACHE FILE
# This option determines where object definitions are cached when
# Nagios starts up. The SCG need object definitions from

```

ctrl+o and enter for saving and ctrl+x to exit nano editor.

## Verify configuration files

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

```

[root@ip-172-31-80-137 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

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

Running pre-flight check on configuration data...

Checking objects...
```

```
Checked 0 service dependencies
Checked 0 host dependencies
Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
[root@ip-172-31-80-137 ec2-user]# |
```

Restart nagios service.

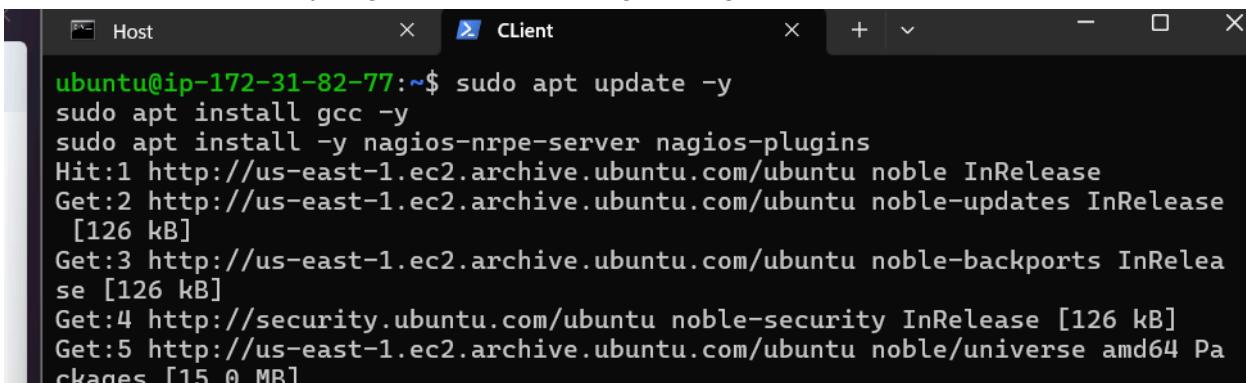
```
service nagios restart
```

```
Things look okay - No serious problems were detected during the pre-flight check
[root@ip-172-31-80-137 ec2-user]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
[root@ip-172-31-80-137 ec2-user]# |
```

### 3) Go to client machine (ubuntu machine)

Perform the following commands

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



A screenshot of a terminal window titled "Client". The terminal shows the following command sequence:

```
ubuntu@ip-172-31-82-77:~$ 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]
```

```
Running kernel seems to be up-to-date.

Restarting services...

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 #1: sshd[990,1101]
ubuntu @ user manager service: systemd[996]

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-82-77:~$ |
```

Open the nrpe.cfg file in nano editor  
sudo nano /etc/nagios/nrpe.cfg

Under allowed\_hosts, add the nagios host ip address (public)

```
# You can either supply a username or a UID.  
#  
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd  
nrpe_user=nagios  
  
# 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  
# (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=127.0.0.1,54.224.175.95  
  
# COMMAND ARGUMENT PROCESSING  
# This option determines whether or not the NRPE daemon will allow clients
```

again save and exit the nano editor.

## 4) Go to nagios dashboard and click on hosts

The screenshot shows the Nagios Core dashboard at the URL <https://54.224.175.95/nagios/>. The top navigation bar indicates "Not secure" and the address. The main header features the Nagios Core logo with a gear icon. Below the header, a green checkmark icon and the text "Daemon running with PID 13935" are displayed. The left sidebar contains several menu sections: General (Home, Documentation), Current Status (Tactical Overview, Map, Hosts, Services, Host Groups, Service Groups), Problems (Services (Unhandled), Hosts (Unhandled), Network Outages), Reports (Availability, Trends, Alerts, History, Summary, Histogram, Notifications, Event Log), and System (Comments, Downtime, Process Info, Performance Info, Scheduling Queue, Configuration). The central content area includes a "Get Started" box with links to monitoring infrastructure, changing look, and addons; a "Quick Links" box with links to Nagios Library, Labs, Exchange, Support, and official sites; and two empty boxes for "Latest News" and "Don't Miss...". At the bottom, there is copyright information and a note about the GNU General Public License.

Click on hosts

This screenshot shows the "Current Status" page under the "Tactical Overview" section. The left sidebar lists several navigation items: Tactical Overview (selected), Map, Hosts, Services, and Host Groups. The main content area is currently empty, indicating no hosts are present.

## 5) Click on linux server

**Nagios\***

**Current Network Status**  
Last Updated: Sat Sep 28 11:33:24 UTC 2024  
Updated every 50 seconds  
Nagios® Core™ 4.5.5 - www.nagios.org  
Logged in as nagiosadmin

**General**  
Home Documentation

**Current Status**  
Tactical Overview Map Hosts Services Host Groups Summary Grid Service Groups Summary Grid Problems Services (Unhandled) Hosts (Unhandled) Network Outages Quick Search:

**Reports**  
Availability Trends Alerts History Summary Histogram Notifications Event Log

**Host Status Totals**  

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

**Service Status Totals**  

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

**Host Status Details For All Host Groups**  
Limit Results: 100 Host Status Last Check Duration Status Information  

Host	Status	Last Check	Duration	Status Information
linuxserver	UP	09-28-2024 11:29:10	0d 0h 8m 36s	PING OK - Packet loss = 0%; RTA = 1.18 ms
localhost	UP	09-28-2024 11:32:18	0d 3h 53m 7s	PING OK - Packet loss = 0%; RTA = 0.03 ms

Results 1 - 2 of 2 Matching Hosts

**Nagios\***

**Host Information**  
Last Updated: Sat Sep 28 11:33:39 UTC 2024  
Updated every 50 seconds  
Nagios® Core™ 4.5.5 - www.nagios.org  
Logged in as nagiosadmin

**General**  
Home Documentation

**Current Status**  
Tactical Overview Map Hosts Services Host Groups Summary Grid Service Groups Summary Grid Problems Services (Unhandled) Hosts (Unhandled) Network Outages Quick Search:

**Reports**  
Availability Trends Alerts History Summary Histogram Notifications Event Log

**Host**  
linuxserver (linuxserver)  
Member of No hostgroups  
54.173.58.143

**Host State Information**  
Host Status: UP (for 0d 0h 8m 51s)  
Status Information: PING OK - Packet loss = 0%, RTA = 1.18 ms  
Performance Data: rta=1.184000ms;3000.000000;5000.000000;0.000000 p=0%;80;100;0  
Current Attempt: 1/10 (HARD state)  
Last Check Time: 09-28-2024 11:29:10  
Check Type: ACTIVE  
Check Latency / Duration: 0.000 / 4.066 seconds  
Next Scheduled Active Check: 09-28-2024 11:34:10  
Last State Change: 09-28-2024 11:24:48  
Last Modification: N/A (notification 0)  
Is This Host Flapping? NO (0.00% state change)  
In Scheduled Downtime? NO  
Last Update: 09-28-2024 11:33:37 ( 0d 0h 0m 2s ago)

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

**Host Comments**  
Add a new comment Delete all comments  
Entry Time Author Comment Comment ID Persistent Type Expires Actions  
This host has no comments associated with it

## 6) Click on nagios services

### Documentation

#### Current Status

##### Tactical Overview

##### Map

##### Hosts

##### Services

##### Host Groups

###### Summary

###### Grid

##### Service Groups

#### Nagios\*

##### General

##### Home

##### Documentation

##### Current Status

###### Tactical Overview

###### Map

###### Hosts

###### Services

###### Host Groups

###### Summary

###### Grid

###### Service Groups

###### Problems

###### Services (Unhandled)

###### Hosts (Unhandled)

###### Network Outages

###### Quick Search:

###### Reports

###### Availability

###### Trends

###### Alerts

###### History

###### Summary

###### Histogram

###### Notifications

###### Event Log

###### System

###### Comments

Current Network Status  
Last Updated: Sat Sep 28 11:33:58 UTC 2024  
Updated every 90 seconds  
Nagios® Core™ 4.5.5 - www.nagios.org

Logged in as nagiosadmin

[View History For all hosts](#)  
[View Notifications For All Hosts](#)  
[View Host Status Detail For All Hosts](#)

Host Status Totals  
Up Down Unreachable Pending

All Problems All Types

Service Status Totals  
Ok Warning Unknown Critical Pending

All Problems All Types

[View Service Status Details For All Hosts](#)

Limit Results: 100

Host \*\* Service \*\* Status \*\* Last Check \*\* Duration \*\* Attempt \*\* Status Information

localhost Current Load OK 09-28-2024 11:30:25 0d 0h 8m 33s 1/4 OK - load average: 0.01, 0.00, 0.00

localhost Current Users OK 09-28-2024 11:31:03 0d 0h 7m 55s 1/4 USERS OK - 2 users currently logged in

HTTP CRITICAL 09-28-2024 11:29:40 0d 0h 4m 18s 4/4 connect to address 54.173.58.143 and port 80: Connection refused

PING OK 09-28-2024 11:32:18 0d 0h 6m 40s 1/4 PING OK - Packet loss = 0%, RTA = 1.03 ms

Root Partition OK 09-28-2024 11:32:55 0d 0h 6m 3s 1/4 DISK OK - free space: /6105 MB (75.23% inode=98%)

SSH OK 09-28-2024 11:33:33 0d 0h 5m 25s 1/4 SSH OK - OpenSSH\_9.6p1 Ubuntu-3ubuntu13.4 (protocol 2.0)

Swap Usage CRITICAL 09-28-2024 11:32:10 0d 0h 1m 48s 4/4 SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size

Total Processes OK 09-28-2024 11:29:48 0d 0h 9m 10s+ 1/4 PROCESSES OK, 37 processes with STATE : R/SZDT

localhost Current Load OK 09-28-2024 11:29:39 0d 3h 53m 5s 1/4 OK - load average: 0.02, 0.01, 0.00

localhost Current Users OK 09-28-2024 11:30:17 0d 3h 52m 27s 1/4 USERS OK - 2 users currently logged in

HTTP WARNING 09-28-2024 11:29:46 0d 2h 49m 12s 4/4 HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.001 second response time

PING OK 09-28-2024 11:31:32 0d 3h 51m 12s 1/4 PING OK - Packet loss = 0%, RTA = 0.03 ms

Root Partition OK 09-28-2024 11:32:09 0d 3h 50m 35s 1/4 DISK OK - free space: /6105 MB (75.23% inode=98%)

SSH OK 09-28-2024 11:32:47 0d 3h 49m 57s 1/4 SSH OK - OpenSSH\_8.7 (protocol 2.0)

Swap Usage CRITICAL 09-28-2024 11:31:24 0d 3h 12m 34s 4/4 SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size

Total Processes OK 09-28-2024 11:29:02 0d 3h 14m 56s 1/4 PROCESSES OK, 37 processes with STATE : R/SZDT

Results 1 - 16 of 16 Matching Services

## Conclusion:

In this lab, we successfully configured a monitoring setup between a Nagios host machine (referred to as "exp9 machine") and a client machine (created specifically for this experiment). The goal was to set up Nagios to monitor a remote Linux server, which involved configuring both the Nagios host and client machine (Ubuntu instance) in an EC2 environment.

We started by launching an Ubuntu EC2 instance as the client machine, ensuring that we used the same security group and key-pair as the Nagios host machine to maintain consistent access and permissions. After establishing SSH connections to both machines, we worked in parallel, using one terminal for the host and another for the client.

On the Nagios host machine, we created a new directory structure, then copied and modified the localhost.cfg file to set up a configuration for monitoring the remote client machine. This included specifying the public IP address of the client

machine and updating the hostgroup and hostname. After editing the Nagios configuration file to recognize the new monitoring host directory, we verified the changes and restarted the Nagios service.

On the client machine, we installed the necessary Nagios packages (`nagios-nrpe-server` and `nagios-plugins`), configured the `nrpe.cfg` file, and allowed communication between the Nagios host and client by updating the `allowed_hosts` configuration.

After these steps, we were able to successfully monitor the remote Linux server from the Nagios dashboard, confirming that our setup was correct. This experiment demonstrated the core concepts of configuring Nagios to monitor a remote machine, providing practical insight into network monitoring and server management in a real-world scenario.

1.

## Adv DevOps Exp-11

### Aim:

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

### Theory:

#### AWS Lambda

A fully managed, serverless computing service where you run code without provisioning or managing servers. Lambda automatically scales your application based on the number of incoming requests or events, ensuring efficient resource utilization. You are only charged for the time your code is running, with no upfront cost, making it cost-effective for on-demand workloads.

#### Lambda Workflow:

- **Create a Function:** Write the function code and define its handler (entry point). You can use the AWS Console, CLI, or upload a deployment package.
- **Set Event Sources:** Define how the function is triggered (e.g., when an object is uploaded to S3 or a DynamoDB table is updated).
- **Execution:** When triggered, Lambda runs your function, executes the logic, and automatically scales to handle the incoming event volume.
- **Scaling and Concurrency:** Lambda scales automatically by launching more instances of the function to handle simultaneous invocations. There are also options for configuring reserved

concurrency to manage traffic.

- **Monitoring and Logging:** Lambda integrates with Amazon CloudWatch for logging and monitoring. Logs for each invocation are sent to CloudWatch, allowing you to track performance and troubleshoot errors.

## AWS Lambda Functions:

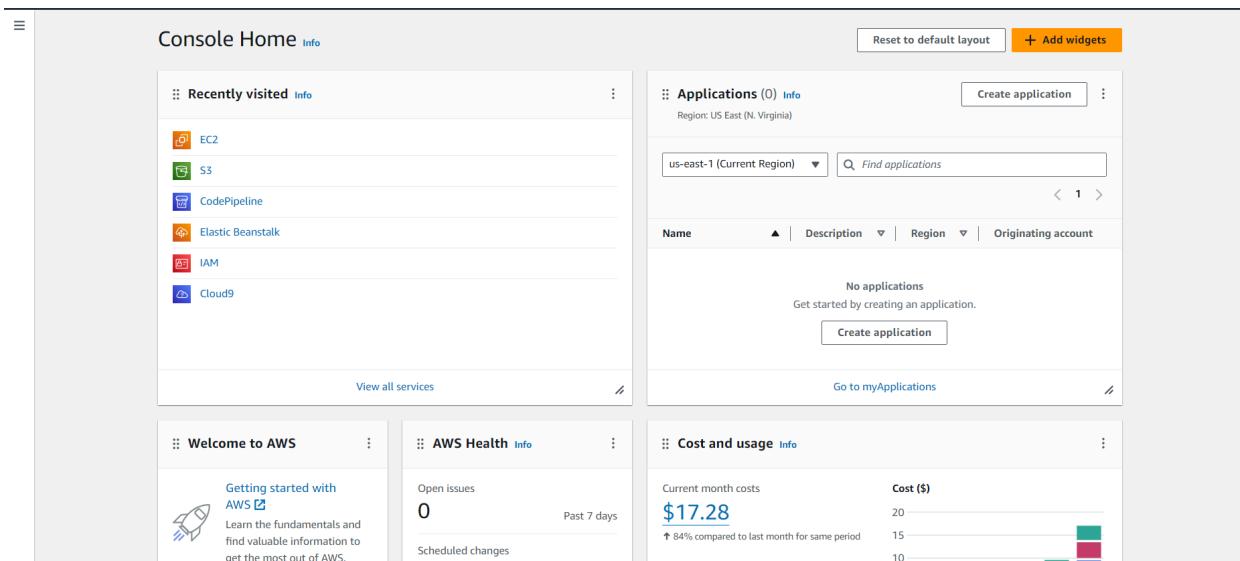
- **Python:** Great for quick development with its rich standard library and support for lightweight tasks.
- **Java:** Typically used for more complex, compute-intensive tasks. While it's robust, cold start times can be higher.
- **Node.js:** Excellent for I/O-bound tasks like handling APIs or streaming data, with fast startup times and efficient memory usage.

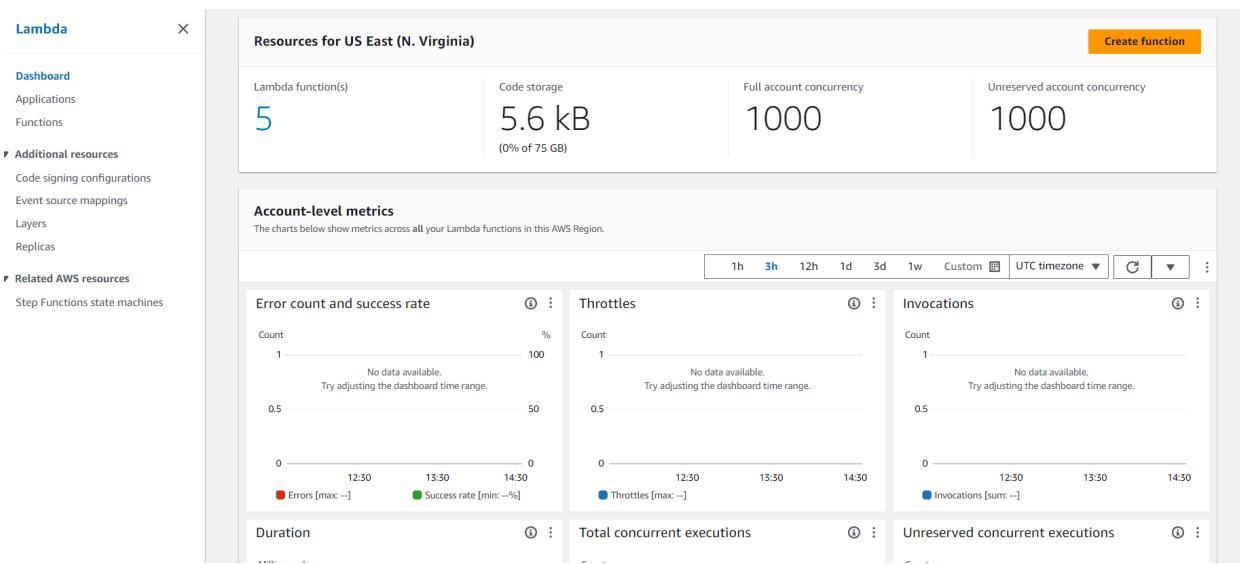
Prerequisites: AWS Personal/Academy Account

Prerequisites: AWS Personal/Academy Account

## Steps To create the lambda function:

**Step 1:** Login to your AWS Personal/Academy Account. Open lambda and click on create function button.





**Step 2:** Now Give a name to your Lambda function, Select the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby. So will select Python 3.12, Architecture as x86, and Execution role to Create a new role with basic Lambda permissions.

AWS Services Search [Alt+S]

Lambda > Functions > Create function

## 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 Info  
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

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

ⓘ Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in this role.

Lambda will create an execution role named KCS\_Lambda-role-kssqesm9, with permission to upload logs to Amazon CloudWatch Logs.

▶ Advanced settings

Cancel Create function

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

Lambda > Functions > KCS\_Lambda

## KCS\_Lambda

Throttle

Copy ARN

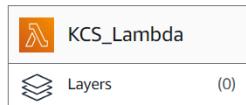
Actions ▾

### Function overview Info

Export to Application Composer

Download ▾

Diagram Template



+ Add trigger

+ Add destination

#### Description

-

#### Last modified

3 seconds ago

#### Function ARN

arn:aws:lambda:us-east-1:235494807211:function:KCS\_Lambda

#### Function URL Info

-

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

Code Test Monitor Configuration Aliases Versions

### Code source Info

Upload from ▾

File Edit Find View Go Tools Window

Test Deploy Changes not deployed

Environment  
Go to Anything (Ctrl-P)  
KCS\_Lambda /  
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 KCS from Lambda!')
8     }
9
```

To See or Edit the basic settings go to configuration then click on edit general configuration.

Code Test Monitor Configuration Aliases Versions

### General configuration

#### General configuration Info

Edit

Description

Memory

128 MB

Ephemeral storage

512 MB

Timeout

SnapStart Info

0 min 3 sec

None

Here, you can enter a description and change Memory and Timeout. I've changed the Timeout period to 2 sec since that is sufficient for now.

**Basic settings** [Info](#)

**Description - optional**  
The supreme leader(KCS) wants to change the basic settings

**Memory** [Info](#)  
Your function is allocated CPU proportional to the memory configured.  
 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](#)  
 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](#).  
 None

Supported runtimes: Java 11, Java 17, Java 21.

**Timeout**  
 min  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.  
 [View the KCS\\_Lambda-role-9nzyyxbk role](#) on the IAM console.

Successfully updated the function **KCS\_Lambda**.

Lambda > Functions > KCS\_Lambda

**KCS\_Lambda**

Throttle [Copy ARN](#) Actions ▾

**Function overview** [Info](#) Export to Application Composer Download ▾

**Diagram** **Template**

 **KCS\_Lambda**  
 Layers (0)

+ Add trigger + Add destination

Description  
The supreme leader(KCS) wants to change the basic settings

Last modified  
2 seconds ago

Function ARN  
 arn:aws:lambda:us-east-1:235494807211:function:KCS\_Lambda

Function URL [Info](#)

**Step 3:** Now Click on the Test tab then select Create a new event, give a name to the event and select Event Sharing to private, and select hello-world template.

Code | **Test** | Monitor | Configuration | Aliases | Versions

**Test event** Info

To invoke your function without saving an event, configure the JSON event, then choose Test.

Test event action

Create new event    Edit saved event

Event name  
KCS\_Event

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

Event sharing settings

Private  
This event is only available in the Lambda console and to the event creator. You can configure a total of 10. [Learn more](#)

Shareable  
This event is available to IAM users within the same account who have permissions to access and use shareable events. [Learn more](#)

Template - optional  
hello-world

Event JSON

```
1 • [{}  
2   "key1": "value1",  
3   "key2": "value2",  
4   "key3": "value3"  
5 ]
```

Format JSON

The test event KCS\_Event was successfully saved. X

Code | **Test** | Monitor | Configuration | Aliases | Versions

**Test event** Info

To invoke your function without saving an event, configure the JSON event, then choose Test.

Test event action

Create new event    Edit saved event

Event name  
KCS\_Event

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

Event sharing settings

Private  
This event is only available in the Lambda console and to the event creator. You can configure a total of 10. [Learn more](#)

Shareable  
This event is available to IAM users within the same account who have permissions to access and use shareable events. [Learn more](#)

Template - optional

**Step 4:** Now In Code section select the created event from the dropdown of test then click on test . You will see the below output.

The screenshot shows the AWS Lambda code editor interface. The top navigation bar includes tabs for Code, Test, Monitor, Configuration, Aliases, and Versions. The Code tab is selected. Below the tabs is a toolbar with File, Edit, Find, View, Go, Tools, and Window. The main area displays a file named lambda\_function.py with the following code:

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

A context menu is open over the code, with the 'Test' option highlighted. A dropdown menu shows 'Configure test event' (Ctrl-Shift-C) and 'Private saved events'. The 'KCS\_Event' option is selected and highlighted in blue.

The screenshot shows the AWS Lambda code editor interface after testing the function. The top navigation bar and toolbar are identical to the previous screenshot. The main area displays the execution results for the 'lambda\_function' test event. The results show a successful execution with a status of 'Succeeded', a maximum memory usage of 32 MB, and a duration of 2.07 ms. The response body is displayed as:

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

Below the response, the 'Function Logs' section shows the log entries for the request:

```
START RequestId: 9b8874c5-da6e-4026-9098-134c4fee787f Version: $LATEST
END RequestId: 9b8874c5-da6e-4026-9098-134c4fee787f
REPORT RequestId: 9b8874c5-da6e-4026-9098-134c4fee787f Duration: 2.07 ms Billed Duration: 3 ms Memory Size: 128 MB Max Memory Used: 32 MB
```

The 'Request ID' is also listed as 9b8874c5-da6e-4026-9098-134c4fee787f.

The screenshot shows the AWS Lambda code editor interface. At the top, there are tabs for Code, Test, Monitor, Configuration, Aliases, and Versions. The Code tab is selected. Below the tabs is a toolbar with File, Edit, Find, View, Go, Tools, Window, Test, Deploy, and a status message Changes not deployed. On the left, there's an Environment sidebar. The main area shows a file tree under KCS\_Lambda / with lambda\_function.py selected. The code editor displays the following Python code:

```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     new_string = "Hey there. I am KCS!"
6     return {
7         'statusCode': 200,
8         'body': json.dumps(new_string)
9     }
10
```

Now ctrl+s to save and click on deploy to deploy the changes

The screenshot shows the AWS Lambda code editor interface after deployment. The Test button is now highlighted. The Execution result section shows a successful deployment with the following details:

- Status: Succeeded
- Max memory used: 32 MB
- Time: 2.13 ms

The Response section shows the output of the function:

```
{ "statusCode": 200, "body": "\"Hey there. I am KCS!\""} 
```

The Function Logs section shows the request and response details:

```
START RequestId: 8cc12d43-7137-4c05-9ecf-315440b7226d Version: $LATEST
END RequestId: 8cc12d43-7137-4c05-9ecf-315440b7226d
REPORT RequestId: 8cc12d43-7137-4c05-9ecf-315440b7226d Duration: 2.13 ms Billed Duration: 3 ms Memory Size: 128 MB Max Memory Used: 32 MB
```

The Request ID is listed as 8cc12d43-7137-4c05-9ecf-315440b7226d.

You can see the desired output.

**Conclusion:** In this experiment, we successfully developed an AWS Lambda function, covering the key steps involved. Starting with the Python-based setup, we configured the function's fundamental settings, including setting the timeout to 1 second. We proceeded to create a test event, deployed the function, and verified its output. Additionally, we made updates to the Lambda function's code and redeployed it, observing the real-time changes. This hands-on experience highlighted AWS Lambda's

efficiency and adaptability, enabling rapid serverless application development while AWS handles infrastructure and scaling effortlessly.

## Adv DevOps Exp-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: [Exp12](#)

#### 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. Prerequisites: AWS Personal Account

**Prerequisites:** AWS Personal Account

#### Steps To create the lambda function:

**Step 1:** Login to your AWS Personal account. Now open S3 from services and click on create S3 bucket.

The screenshot shows the Amazon S3 service page. At the top left, it says "Storage" and "Amazon S3". Below that, it says "Store and retrieve any amount of data from anywhere". A small note below states: "Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance." On the right, there is a white box with a dark border containing the text "Create a bucket" and a description: "Every object in S3 is stored in a bucket. To upload files and folders to S3, you'll need to create a bucket where the objects will be stored." Below this is a large orange "Create bucket" button. To the right of this box is another white box with a dark border titled "Pricing". It contains the text: "With S3, there are no minimum fees. You only pay for what you use. Prices are based on the location of your S3 bucket." Below this is a link: "Estimate your monthly bill using the [AWS Simple Monthly Calculator](#)". At the bottom of the pricing box is a link: "View pricing details".

**Step 2:** Now Give a name to the Bucket, select general purpose project and deselect the Block public access and keep other this to default.

**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](#)

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*  
Only the bucket settings in the following configuration are copied.  
[Choose bucket](#)

Format: s3://bucket/prefix

## Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

### ACLs disabled (recommended)

All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

### ACLs enabled

Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership

Bucket owner enforced

## Block Public Access settings for this bucket

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

### Block all public access

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

#### Block public access to buckets and objects granted through new access control lists (ACLs)

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

#### Block public access to buckets and objects granted through any access control lists (ACLs)

S3 will ignore all ACLs that grant public access to buckets and objects.

#### Block public access to buckets and objects granted through new public bucket or access point policies

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

#### Block public and cross-account access to buckets and objects through any public bucket or access point policies

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



Turning off block all public access might result in this bucket and the objects within becoming

Successfully created bucket "weareeks" View details X

To upload files and folders, or to configure additional bucket settings, choose View details.

Amazon S3 > Buckets

Account snapshot - updated every 24 hours All AWS Regions View Storage Lens dashboard

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

General purpose buckets Create bucket

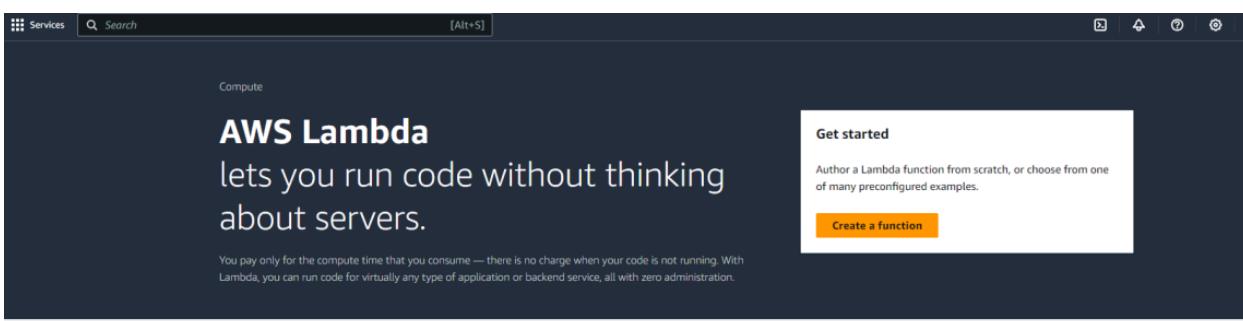
General purpose buckets (1) Info All AWS Regions

Buckets are containers for data stored in S3.

Find buckets by name

Name	AWS Region	IAM Access Analyzer	Creation date
weareeks	US East (N. Virginia) us-east-1	<a href="#">View analyzer for us-east-1</a>	October 1, 2024, 13:40:40 (UTC+05:30)

**Step 3:** Open lambda console and click on create function button

This screenshot shows the 'How it works' section of the Lambda landing page. It includes a code editor with a green header containing the text 'Hello from Lambda!' and a Node.js runtime selected. The code editor displays the following Node.js code:

```
1 * exports.handler = async (event) => {
2     console.log(event);
3     return 'Hello from Lambda!';
4 };
5
```

Below the code editor are tabs for '.NET', 'Java', 'Node.js', 'Python', 'Ruby', and 'Custom runtime'. To the right of the code editor are two buttons: 'Run' and 'Next: Lambda responds to events'.

**Step 4:** Now Give a name to your Lambda function, Select the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby. So will select Python 3.12 , Architecture as x86, and Execution role to Create a new role with basic Lambda permissions.

This screenshot shows the 'Create function' wizard. The first step, 'Choose one of the following options to create your function.', has three options: 'Author from scratch' (selected), 'Use a blueprint', and 'Container image'. The second step, 'Basic information', includes fields for 'Function name' (KCS\_Exp12), 'Runtime' (Python 3.12), 'Architecture' (x86\_64 selected), and 'Permissions'. The third step, 'Advanced settings', is partially visible at the bottom.

Successfully created the function KCS\_Exp12. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Lambda > Functions > KCS\_Exp12

### KCS\_Exp12

Function overview [Info](#)

[Diagram](#) [Template](#)

 KCS\_Exp12  
 Layers (0)

+ Add trigger + Add destination

Description  
-

Last modified  
3 seconds ago

Function ARN  
 arn:aws:lambda:us-east-1:235494807211:function:KCS\_Exp12

Function URL [Info](#)  
-

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

[Code source](#) [Info](#)

File Edit Find View Go Tools Window Test Deploy

Upload from [▼](#)

Code

Code source [Info](#)

File Edit Find View Go Tools Window Test Deploy

Go to Anything (Ctrl-P)

KCS\_Exp12 / 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
```

Upload from [▼](#)

To See or Edit the basic settings go to configuration then click on edit general setting

Code Test Monitor Configuration Aliases Versions

General configuration

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

Description	Memory	Ephemeral storage
KCS exp12	128 MB	512 MB
Timeout	SnapStart <a href="#">Info</a>	
0 min 2 sec	None	

Change any setting of your choice. Here I have set a timeout of 2 secs. Then save changes

## Edit basic settings

**Basic settings** [Info](#)

Description - *optional*  
KCS exp12

**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](#).  
None  
Supported runtimes: Java 11, Java 17, Java 21.

**Timeout**  
0 min 2 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/KCS\_Exp12-role-0q6h1t4r [View the KCS\\_Exp12-role-0q6h1t4r role](#) on the IAM console.

**Step 5:** Now Click on the Test tab then select Create a new event, give a name to the event and select Event Sharing to private, and select s3 put template.

Code | **Test** | Monitor | Configuration | Aliases | Versions

**Test event** [Info](#)

To invoke your function without saving an event, configure the JSON event, then choose Test.

Test event action  
 Create new event [Edit saved event](#)

Event name  
KCS\_Bucket

Event sharing settings  
 Private  
This event is only available in the Lambda console and to the event creator. You can configure a total of 10. [Learn more](#)  
 Shareable  
This event is available to IAM users within the same account who have permissions to access and use shareable events. [Learn more](#)

Template - *optional*  
s3-put [Format JSON](#)

Event JSON

```

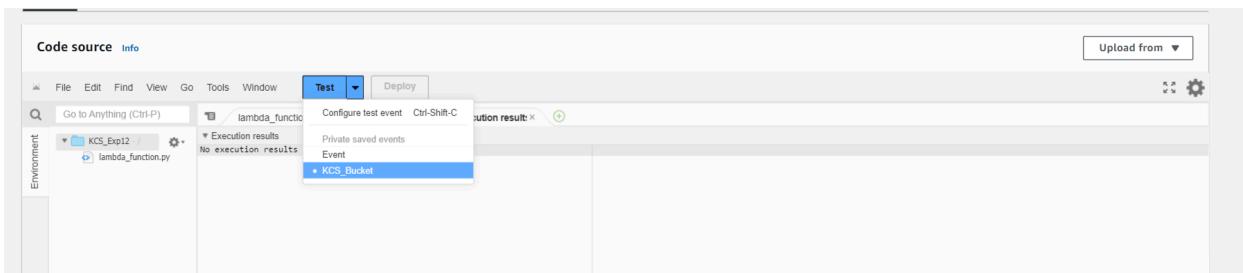
1  [
2    "Records": [
3      {
4        "eventVersion": "2.0",
5        "eventSource": "aws:s3",
6        "awsRegion": "us-east-1",
7        "eventTime": "1970-01-01T00:00:00.000Z",
8        "eventName": "ObjectCreated:Put",
9        "userIdentity": {
10          "principalId": "EXAMPLE"
11        },
12        "requestParameters": {
13          "sourceIPAddress": "127.0.0.1"
14        },
15        "responseElements": {
16          "x-amz-request-id": "EXAMPLE123456789",
17          "x-amz-id-2": "EXAMPLE123/5678abcdefghijklmnaqrstuvwxyzABCDEFH"
18        },
19        "s3": {
20          "s3SchemaVersion": "1.0",
21          "configurationId": "testConfigRule",
22          "bucket": {
23            "name": "example-bucket",
24            "ownerIdentity": {
25              "principalId": "EXAMPLE"
26            },
27            "arn": "arn:aws:s3:::example-bucket"
28          },
29          "object": {
30            "key": "test%2Fkey",
31          }
32        }
33      }
34    ]
35  ]

```

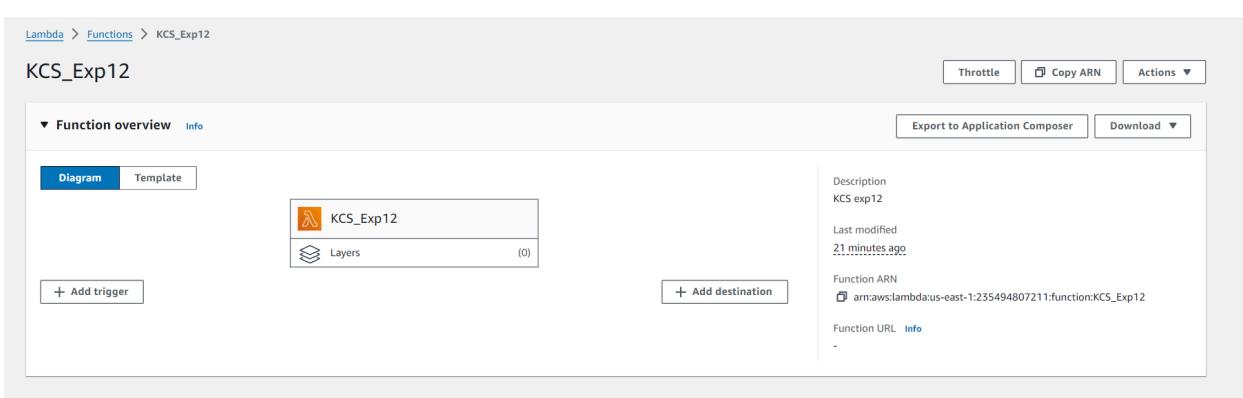
Format JSON

1:1 JSON Spaces: 2

**Step 6:** Now In the Code section select the created event from the dropdown .



**Step 7:** Now In the Lambda function click on add trigger



Now select the source as S3 then select the bucket name from the dropdown, keep other things to default and also you can add prefix to image.

## Add trigger

### Trigger configuration [Info](#)



#### Bucket

Choose or enter the ARN of an S3 bucket that serves as the event source. The bucket must be in the same region as the function.

 X C

Bucket region: us-east-1

#### Event types

Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.



All object create events X

#### Prefix - optional

Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters. Any [special characters](#) must be URL encoded.

KCS Image

#### Suffix - optional

Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters. Any [special characters](#) must be URL encoded.

e.g. jpg

#### Recursive invocation

X Throttle Copy ARN Actions ▾

### KCS\_Exp12

The trigger wearekcs was successfully added to function KCS\_Exp12. The function is now receiving events from the trigger.

#### Function overview [Info](#)

[Export to Application Composer](#)

[Download](#) ▾

Diagram Template



+ Add trigger

Description  
KCS exp12

Last modified  
26 minutes ago

Function ARN  
[arn:aws:lambda:us-east-1:235494807211:function:KCS\\_Exp12](#)

Function URL [Info](#)

+ Add destination

The screenshot shows the AWS Lambda Configuration page. The left sidebar contains links for General configuration, Triggers, Permissions, Destinations, Function URL, Environment variables, Tags, VPC, RDS databases, Monitoring and operations tools, Concurrency and recursion detection, Asynchronous invocation, Code signing, File systems, and State machines. The main area is titled 'Triggers (1) Info' and shows a single trigger named 'S3: wearekcs' which is triggered by 'arn:aws:s3:::wearekcs'. There are buttons for 'Edit', 'Delete', and 'Add trigger'.

**Step 8:** Now Write code that logs a message like "An Image has been added" when triggered. Save the file and click on deploy.

The screenshot shows the AWS Lambda Code source editor. The top navigation bar includes links for Code, Test, Monitor, Configuration, Aliases, and Versions. The main area displays the code for 'lambda\_function.py' under the 'KCS\_Exp12 /' folder. The code is as follows:

```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     bucket_name = event['Records'][0]['s3']['bucket']['name']
6     object_key = event['Records'][0]['s3']['object']['key']
7
8     print(f"An image has been added to the bucket {bucket_name} : {object_key}")
9     return {
10         'statusCode': 200,
11         'body': json.dumps('Log entry created successfully')
12     }
13
14
```

Successfully updated the function KCS\_Exp12.

Code | Test | Monitor | Configuration | Aliases | Versions

Code source Info

File Edit Find View Go Tools Window Test Deploy

Go to Anything (Ctrl-P)

KCS\_Exp12 - / lambda\_function Environment Var Execution result

Status: Succeeded | Max memory used: 32 MB | Time: 2.00 ms

Test Event Name: KCS\_Bucket

Response:

```
{ "statusCode": 200, "body": "\"Log entry created successfully\""} 
```

Function Logs:

```
START RequestId: 28cd6419-7e4f-40fd-a8e8-2e44bf2320d2 Version: $LATEST
An image has been added to the bucket example-bucket : test%2Fkey
END RequestId: 28cd6419-7e4f-40fd-a8e8-2e44bf2320d2
REPORT RequestId: 28cd6419-7e4f-40fd-a8e8-2e44bf2320d2 Duration: 2.00 ms Billed Duration: 3 ms Memory Size: 128 MB Max Memory Used: 32 MB Init Duration: 92.40 ms
```

Request ID: 28cd6419-7e4f-40fd-a8e8-2e44bf2320d2

**Step 9:** Now upload any image to the bucket.

## Upload Info

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

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

### Files and folders (1 Total, 957.0 KB)

All files and folders in this table will be uploaded.

<input type="checkbox"/> Find by name		<	1	>
<input type="checkbox"/>	Name	▼   Folder		
<input type="checkbox"/>	F_i0UxsXgAAXB2s.jpg	-		

### Destination Info

#### Destination

[s3://wearekcs](#)

#### ▶ Destination details

Bucket settings that impact new objects stored in the specified destination.

#### ▶ Permissions

Grant public access and access to other AWS accounts.

#### ▶ Properties

Specify storage class, encryption settings, tags, and more.

Cancel

Upload

Upload succeeded  
View details below.

Upload: status

The information below will no longer be available after you navigate away from this page.

**Summary**

Destination	Succeeded	Failed
s3://woreks	1 file, 957.0 KB (100.00%)	0 files, 0 B (0%)

**Files and folders** (1 Total, 957.0 KB)

Name	Folder	Type	Size	Status	Error
F_000sXgA...	-	image/jpeg	957.0 KB	Succeeded	-

**Step 10:** Now to click on test in lambda to check whether it is giving log when image is added to S3

Code | Test | Monitor | Configuration | Aliases | Versions

**Code source** Info

File Edit Find View Go Tools Window Test Deploy

Go to Anything (Ctrl-P) lambda\_function Environment Var Execution result

Status: Succeeded | Max memory used: 32 MB | Time: 1.88 ms

Execution results

Test Event Name KCS\_Bucket

Response

```
{
  "statusCode": 200,
  "body": "\"Log entry created successfully\""
}
```

Function Logs

```
START RequestId: b624cc5-6862-4d62-84ca-6a1bf867d831 Version: $LATEST
An image has been added to the bucket example-bucket : test%2Fkey
END RequestId: ba624cc5-6862-4d62-84ca-6a1bf867d831
REPORT RequestId: ba624cc5-6862-4d62-84ca-6a1bf867d831 Duration: 1.88 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 32 MB
```

Request ID

```
ba624cc5-6862-4d62-84ca-6a1bf867d831
```

**Step 11:** Now Lets see the log on Cloud watch. To see it go to monitor section and then click on view cloudwatch logs.

**Log events**

Actions ▾

Start tailing

Create metric filter

You can use the filter bar below to search for and match terms, phrases, or values in your log events. [Learn more about filter patterns](#)

<input type="text"/> Filter events - press enter to search	Clear	1m	30m	1h	12h	Custom	UTC timezone	Display	
--	-------	----	-----	----	-----	--------	--------------	---------	--

▶	Timestamp	Message
No older events at this moment. <a href="#">Retry</a>		
▶	2024-10-01T08:55:09.068Z	INIT_START Runtime Version: python:3.12.v36 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:188d9ca2e2714ff5637bd2bbe..
▶	2024-10-01T08:55:09.163Z	START RequestId: 28cd6419-7e4f-40fd-a8e8-2e44bf2320d2 Version: \$LATEST
▶	2024-10-01T08:55:09.164Z	An image has been added to the bucket example-bucket : test%2Fkey
▶	2024-10-01T08:55:09.174Z	END RequestId: 28cd6419-7e4f-40fd-a8e8-2e44bf2320d2
▶	2024-10-01T08:55:09.174Z	REPORT RequestId: 28cd6419-7e4f-40fd-a8e8-2e44bf2320d2 Duration: 2.00 ms Billed Duration: 3 ms Memory Size: 128 MB Max Memor...
▶	2024-10-01T08:59:18.675Z	START RequestId: ba624cc5-6862-4d62-84ca-6a1bf867d831 Version: \$LATEST
▶	2024-10-01T08:59:18.676Z	An image has been added to the bucket example-bucket : test%2Fkey
▶	2024-10-01T08:59:18.678Z	END RequestId: ba624cc5-6862-4d62-84ca-6a1bf867d831
▶	2024-10-01T08:59:18.678Z	REPORT RequestId: ba624cc5-6862-4d62-84ca-6a1bf867d831 Duration: 1.88 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memor...
No newer events at this moment. Auto retry paused. <a href="#">Resume</a>		