

Name: Chinmay Chaudhari

Div: D15C

Static Hosting:

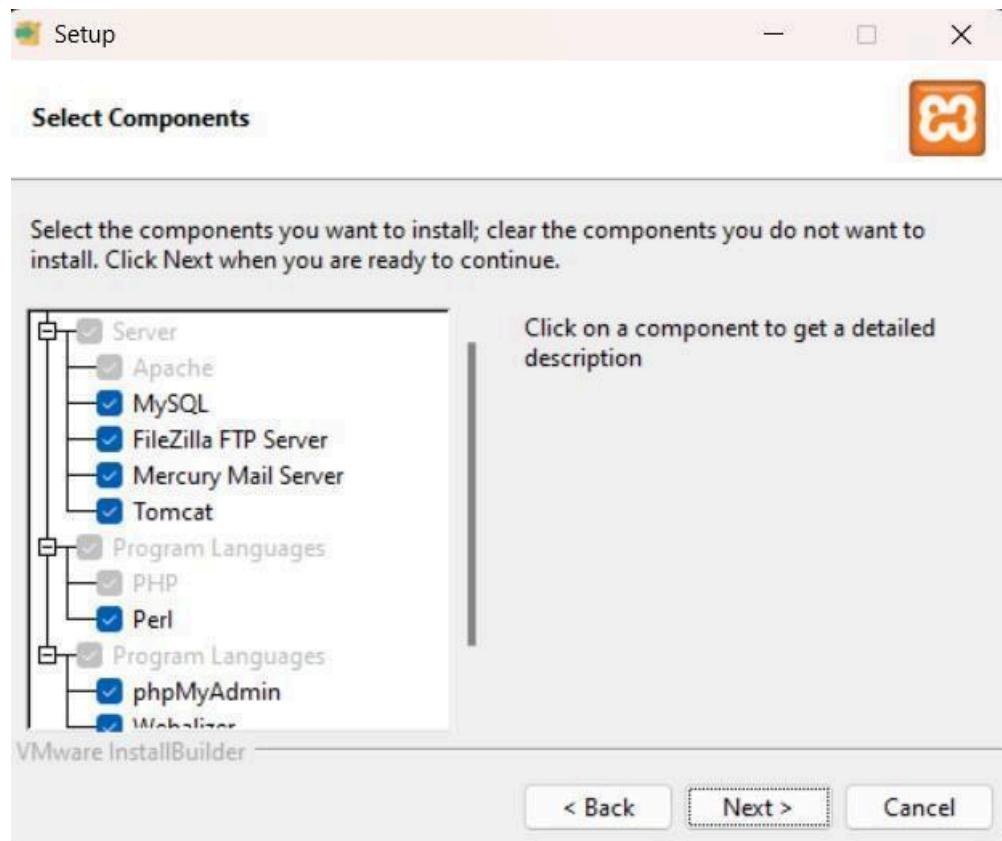
1) On local server (XAMPP)

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

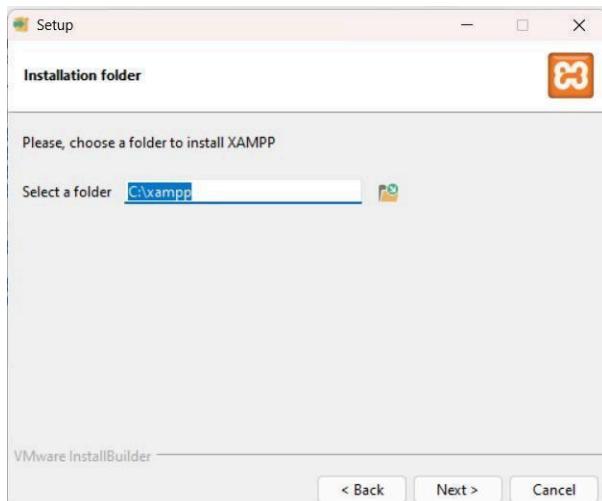
- 1) Select your OS. It will automatically start downloading.



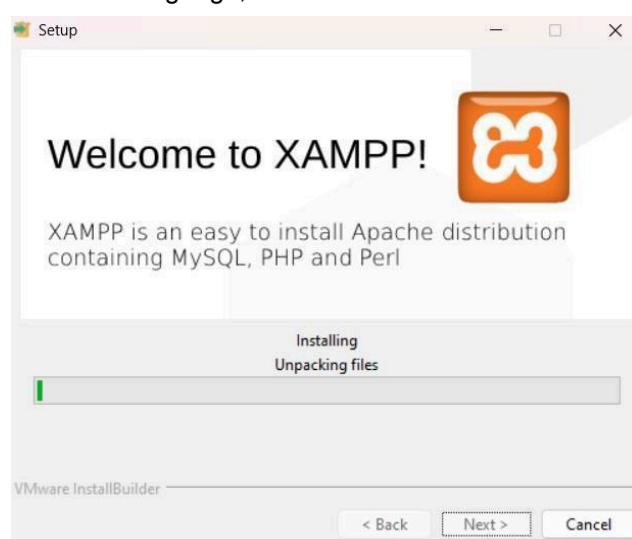
- 2) Open the setup file. Select all the required components and click next



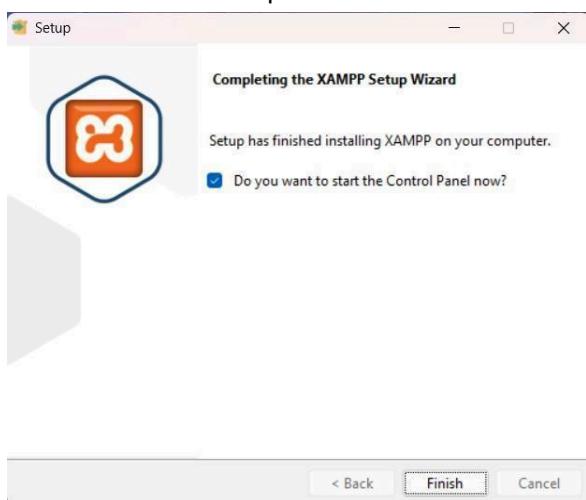
- 3) Choose the folder to install XAMPP in. Make sure the folder is empty. Click next



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



- 5) The installation is complete. Click Finish



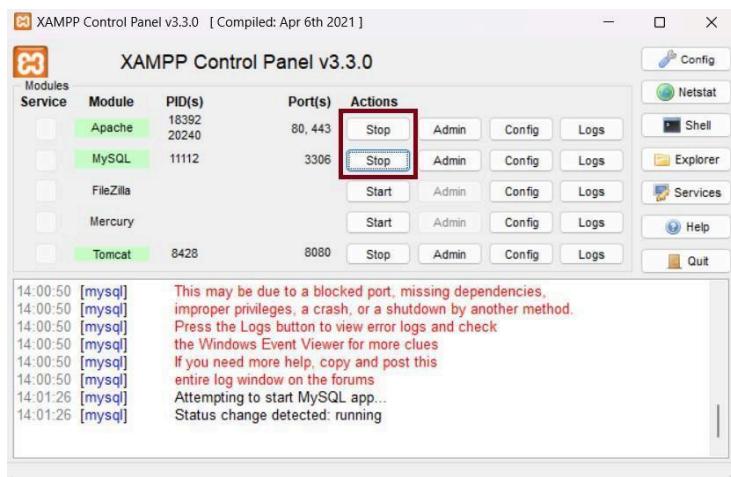
Step 2: Setup a file that is to be hosted on the server. Make sure the file has extension .php

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

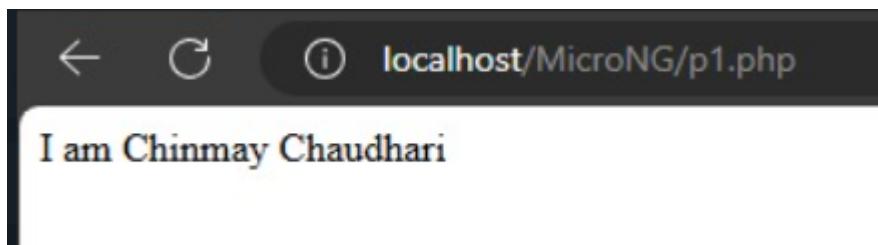
Step 3: Go to the directory where XAMPP was installed. Go to **htdocs** folder. Place your folder in this directory.

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)

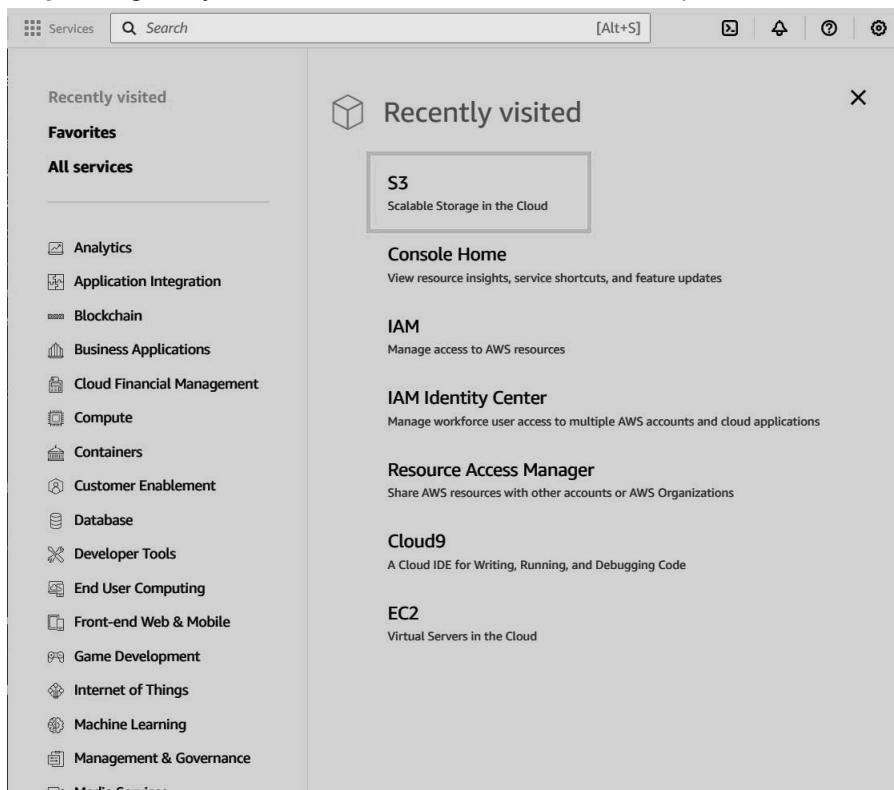


Step 5: Open your web browser. Type localhost/YOUR_FILENAME.php. This will open your website on your browser.



2) AWS S3

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



Step 2: Click on Create Bucket

The screenshot shows the AWS S3 service page. On the left, there's a video thumbnail titled "Introduction to Amazon S3" with a "Copy link" button. To the right of the video, a large call-to-action box contains the text: "Create a bucket" and "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 prominent orange "Create bucket" button. To the right of the main content area, there are sections for "Pricing" (with a note about no minimum fees) and "Resources" (linking to the User guide). At the bottom of the page, there are links for CloudShell, Feedback, and various AWS terms like Privacy, Terms, and Cookie preferences.

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 top navigation bar includes "Amazon S3 > Buckets > Create bucket". The main section is titled "Create bucket" with an "Info" link. It says "Buckets are containers for data stored in S3." Below this is a "General configuration" section. Under "AWS Region", it shows "US East (N. Virginia) us-east-1". Under "Bucket type", two options are listed: "General purpose" (selected) and "Directory - New". The "General purpose" option is described as recommended for most use cases and access patterns. The "Bucket name" field is filled with "statichosting27". There's also a "Choose bucket" button and a note about copy settings from existing buckets. At the bottom, there's an "Object Ownership" section with an "Info" link and a note about controlling ownership of objects. The footer includes links for CloudShell, Feedback, and standard AWS terms.

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

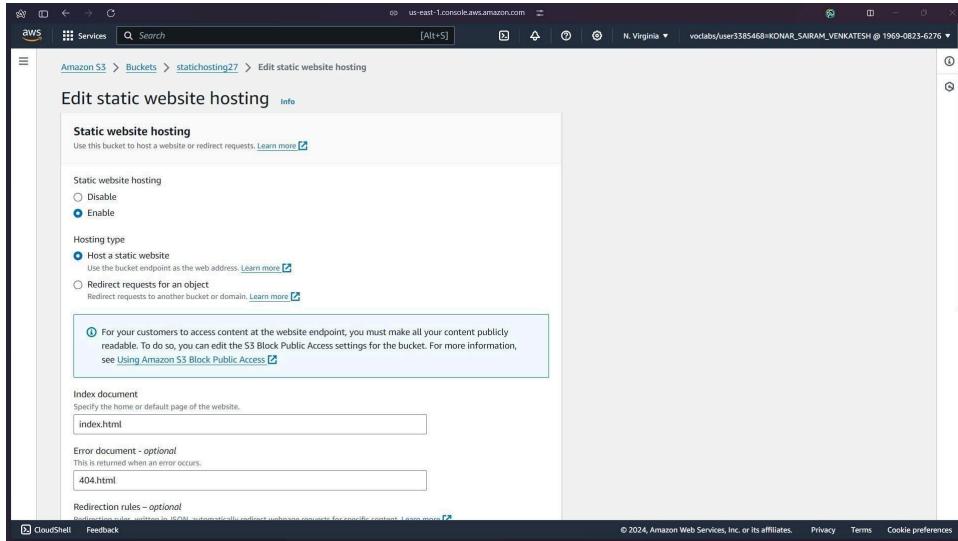
The screenshot shows the AWS S3 Buckets page. At the top, there's an account snapshot and a link to the Storage Lens dashboard. Below that, tabs for 'General purpose buckets' and 'Directory buckets' are shown, with 'General purpose buckets' selected. A search bar allows finding buckets by name. A table lists one bucket: 'statichosting27'. The table columns include Name, AWS Region, IAM Access Analyzer, and Creation date. The 'Name' column shows 'statichosting27', the 'AWS Region' column shows 'US East (N. Virginia) us-east-1', the 'IAM Access Analyzer' column shows 'View analyzer for us-east-1', and the 'Creation date' column shows 'August 4, 2024, 15:30:03 (UTC+05:30)'. At the bottom right of the table, the 'Create bucket' button is highlighted with a red box.

The screenshot shows the properties page for the 'statichosting27' bucket. The 'Properties' tab is selected and highlighted with a red box. Below it, other tabs include 'Objects', 'Permissions', 'Metrics', 'Management', and 'Access Points'. The 'Objects' section displays '0 objects' and includes buttons for 'Copy', 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Actions', 'Create folder', and 'Upload'. A search bar for 'Find objects by prefix' is present. A message states 'No objects' and 'You don't have any objects in this bucket.' An 'Upload' button is located at the bottom of this section.

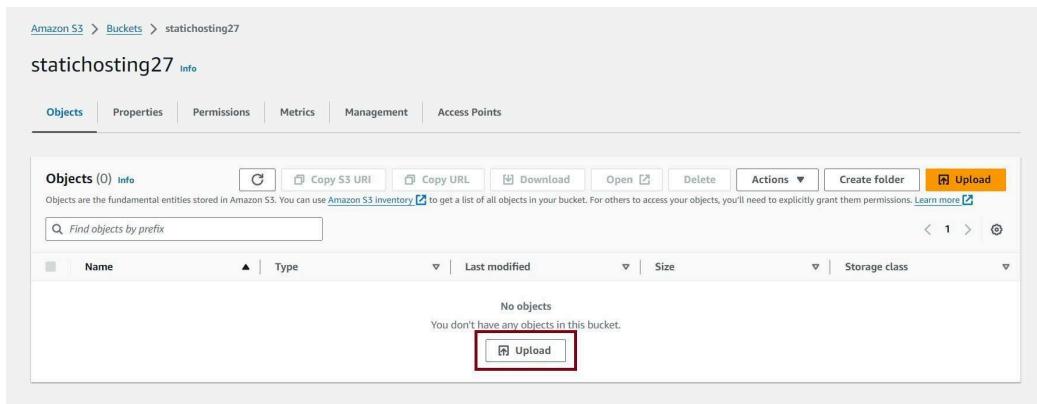
Step 5: Scroll down till you find Static website hosting, click on edit

The screenshot shows the properties page for the 'statichosting27' bucket, specifically the 'Static website hosting' section. This section is highlighted with a red box. It contains a note: 'Use this bucket to host a website or direct requests. Learn more.' Below this, the status is 'Disabled'. An 'Edit' button is located at the bottom right of this section, also highlighted with a red box.

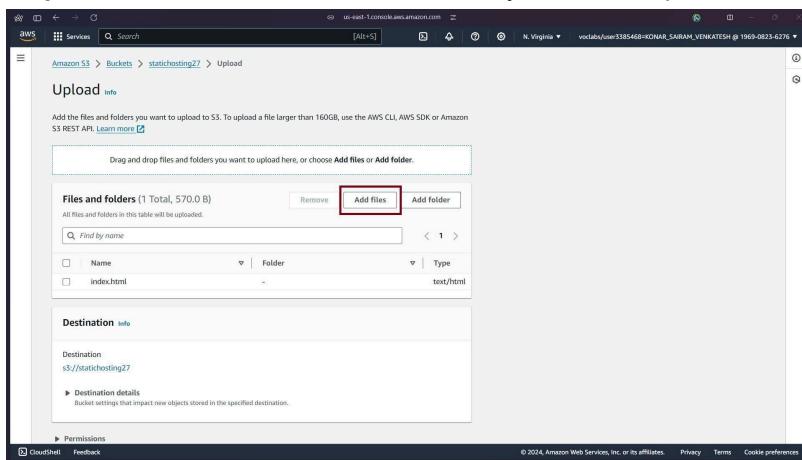
Step 6: Enable static website hosting, in Index document, write the name of your document and in error document, give name as 404.html. Save your changes.



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' (Bucket hosting) and 'Bucket website endpoint'. The endpoint URL is highlighted with a red box: <http://statichosting27.s3-website-us-east-1.amazonaws.com>.

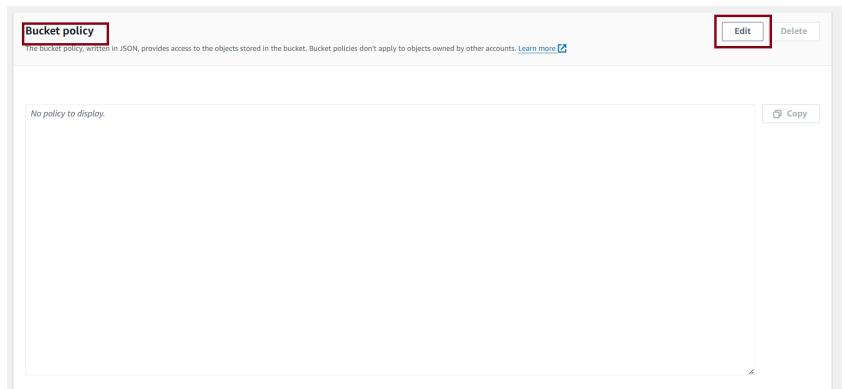
Step 10: Open the link. It will show a 403 forbidden error screen as the contents of the bucket are not available for the public users. To change this, go to Permissions tab, go to Block public access and click on edit

The screenshot shows a 403 Forbidden error page. The browser address bar shows the URL: statichosting27.s3-website-us-east-1.amazonaws.com. The main content of the page lists several error codes and messages, including 'Code: AccessDenied' and '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. It features a 'Block public access (bucket settings)' section with a note about setting up public access. A checkbox for 'Block all public access' is checked and highlighted with a red box. Below it are four other unchecked options: 'Block public access to buckets and objects granted through new access control lists (ACLs)', 'Block public access to buckets and objects granted through new public bucket or access point policies', and 'Block public and cross-account access to buckets and objects through any public bucket or access point policies'. At the bottom are 'Cancel' and 'Save changes' buttons.

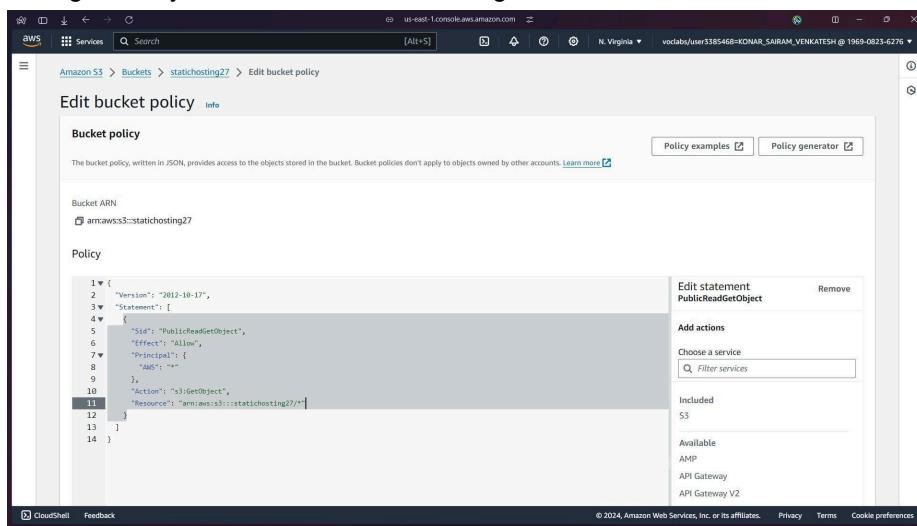
Step 12: Scroll down to bucket policy and click edit



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



Name: Chinmay Chaudhari
Div: D15C
Roll No.6

Experiment 1B: IAM and cloud9

Outputs:

The screenshot displays two main sections of the AWS Management Console.

AWS Cloud9 (Top Left): Shows the creation of an environment named "MyEnvironment". A progress bar indicates "Creating MyEnvironment. This can take several minutes. While you wait, see Best practices for using AWS Cloud9". The status message below says "For capabilities similar to AWS Cloud9, explore AWS Toolkits in your own IDE and AWS CloudShell in the AWS Management Console. Learn more". The "Environments" table shows one entry: "MyEnvironment" (Open, EC2 instance, Secure Shell (SSH), Owner, ARN: arn:aws:sts::354256622778:assumed-role/vclabs/user3404112=SHARMA_RAKSHIT_KUMAR).

IAM Management Console (Bottom Right):

- User Creation:** Shows a success message: "User created successfully". It includes a "View user" button and a summary table for the new user "sample".
- User Groups:** Shows the "sample_group" info page. Summary details: User group name "sample_group", Creation time "August 07, 2024, 09:50 (UTC+05:30)", ARN "arn:aws:iam::434768569951:group/sample_group".
- Users in this group:** Shows the "sample" user is associated with the "sample_group".

Name: Chinmay Chaudhari
Div: D15C
Roll No:6

Practical No 2 : Elastic Beanstalk

The screenshot shows the AWS Elastic Beanstalk console for a environment named "Sampel-env". At the top, two green success notifications are displayed: "Environment successfully launched." and "Successfully uploaded file Screenshot 2023-11-10 185456.png to S3, created application version and started deployment with new application version". Below the notifications, the environment overview section shows the following details:

Health	Environment ID
⚠ Warning	e-u7kfdezi3r
Domain	Application name
kshitij.us-east-1.elasticbeanstalk.com	sampel

The "Platform" section indicates "Platform: PHP 8.3 running on 64bit Amazon Linux 2023/4.3.1", "Running version: -", and "Platform state: Supported".

Below the overview, there are tabs for Events, Health, Logs, Monitoring, Alarms, Managed updates, and Tags. The "Events" tab is selected, showing 11 events. One event is listed:

Time	Type	Details
August 9, 2024 21:25:22 (UTC+5:30)	WARN	Service role "arn:aws:iam::996474913977:role/EMR_EC2_DefaultRole" is missing permissions required to check for pending updates. Verify if the update policy is correctly defined.

A large "Congratulations!" message is displayed, stating: "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." A note below says: "For next steps, read the AWS CodePipeline Documentation. Incedge 2020".

The screenshot shows the AWS CodePipeline console for a pipeline that has just completed a deployment. The main message reads: "Hello this is my first deployment D15C". Below this, a summary message states: "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." A note at the bottom says: "For next steps, read the AWS CodePipeline Documentation. Incedge 2020".

Aim:

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

Theory:

Container-based microservices architectures have revolutionized how development and operations teams test and deploy modern software. Containers allow companies to scale and deploy applications more efficiently, but they also introduce new challenges, adding complexity by creating a whole new infrastructure ecosystem.

Today, both large and small software companies are deploying thousands of container instances daily. Managing this level of complexity at scale requires advanced tools. Like Kubernetes.

Originally developed by Google, Kubernetes is an open-source container orchestration platform designed to automate the deployment, scaling, and management of containerized applications. Kubernetes has quickly become the de facto standard for container orchestration and is the flagship project of the Cloud Native Computing Foundation (CNCF), supported by major players like Google, AWS, Microsoft, IBM, Intel, Cisco, and Red Hat.

Kubernetes simplifies the deployment and operation of applications in a microservice architecture by providing an abstraction layer over a group of hosts. This allows development teams to deploy their applications while Kubernetes takes care of key tasks, including:

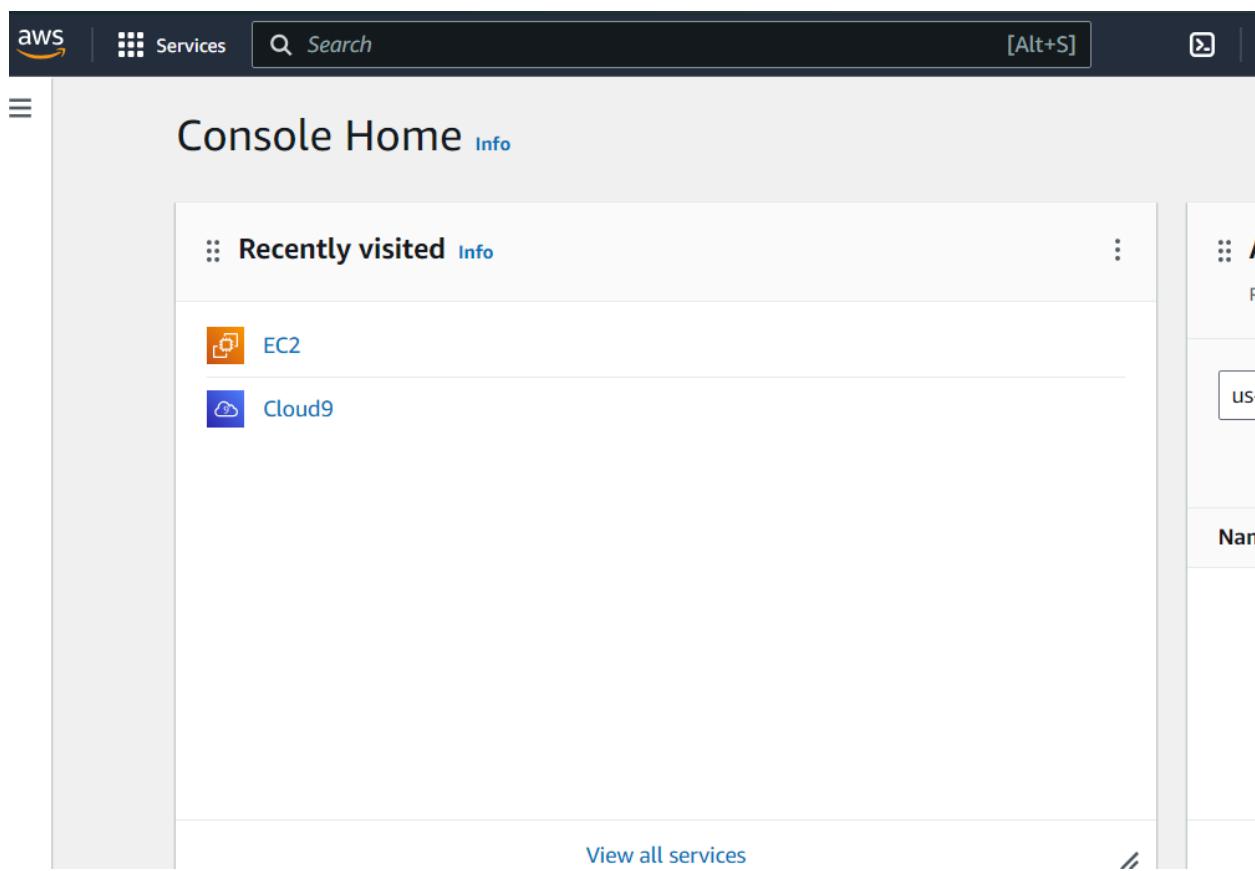
- Managing resource consumption by applications or teams
- Distributing application load evenly across the infrastructure
- Automatically load balancing requests across multiple instances of an application
- Monitoring resource usage to prevent applications from exceeding resource limits and automatically restarting them if needed
- Moving application instances between hosts when resources are low or if a host fails
- Automatically utilizing additional resources when new hosts are added to the cluster
- Facilitating canary deployments and rollbacks with ease.

Steps:

Set Up the instances of each machine

1. open the aws academy.

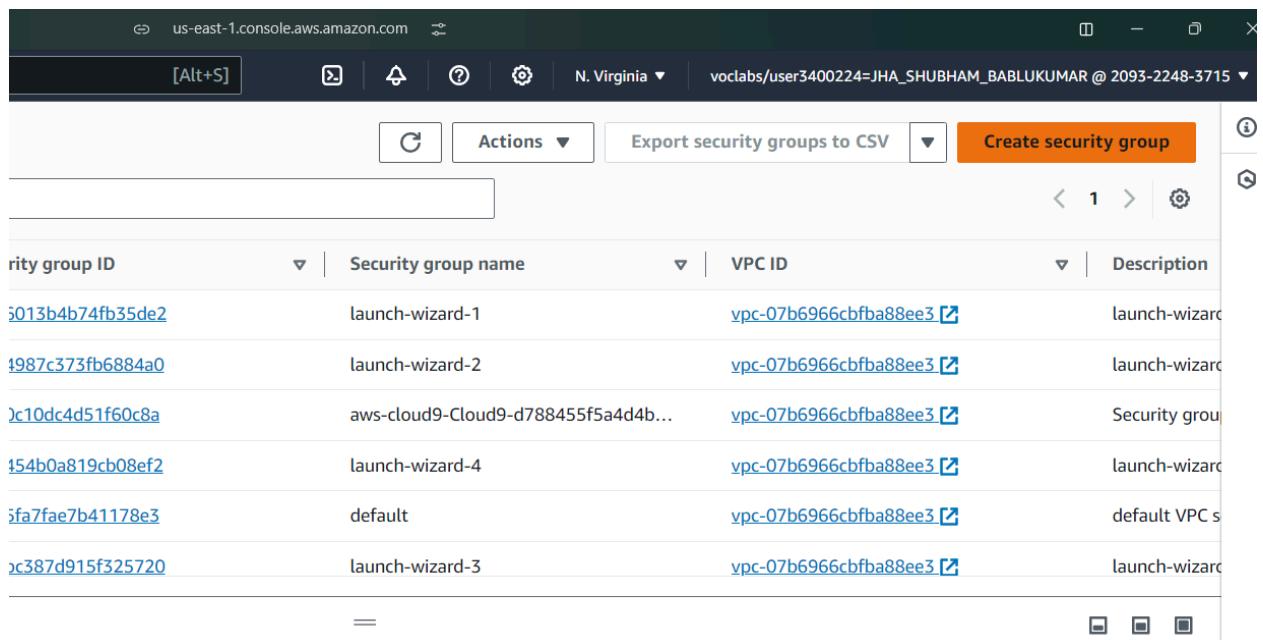
The screenshot shows the AWS Academy Learner interface. At the top, there's a navigation bar with a search bar containing "awsacademy.instruction". Below it, a breadcrumb trail reads "≡ ALLv2EN-US... > Modules > AWS Academ... > Launch AWS Academy Learner I". On the left, a sidebar has links for "Home", "Modules" (which is selected and highlighted in blue), "Discussions", "Grades" (with a red notification badge showing "1"), and "Lucid (Whiteboard)". The main content area has a header "AWS" with a green progress bar. To the right of the progress bar, it says "Used \$4.8 of \$50". The main content area contains a terminal-like window with the following text:
eee_W_3409509@runweb137029:~\$ ec2
bash: ec2: command not found
eee_W_3409509@runweb137029:~\$



2. Click on security groups

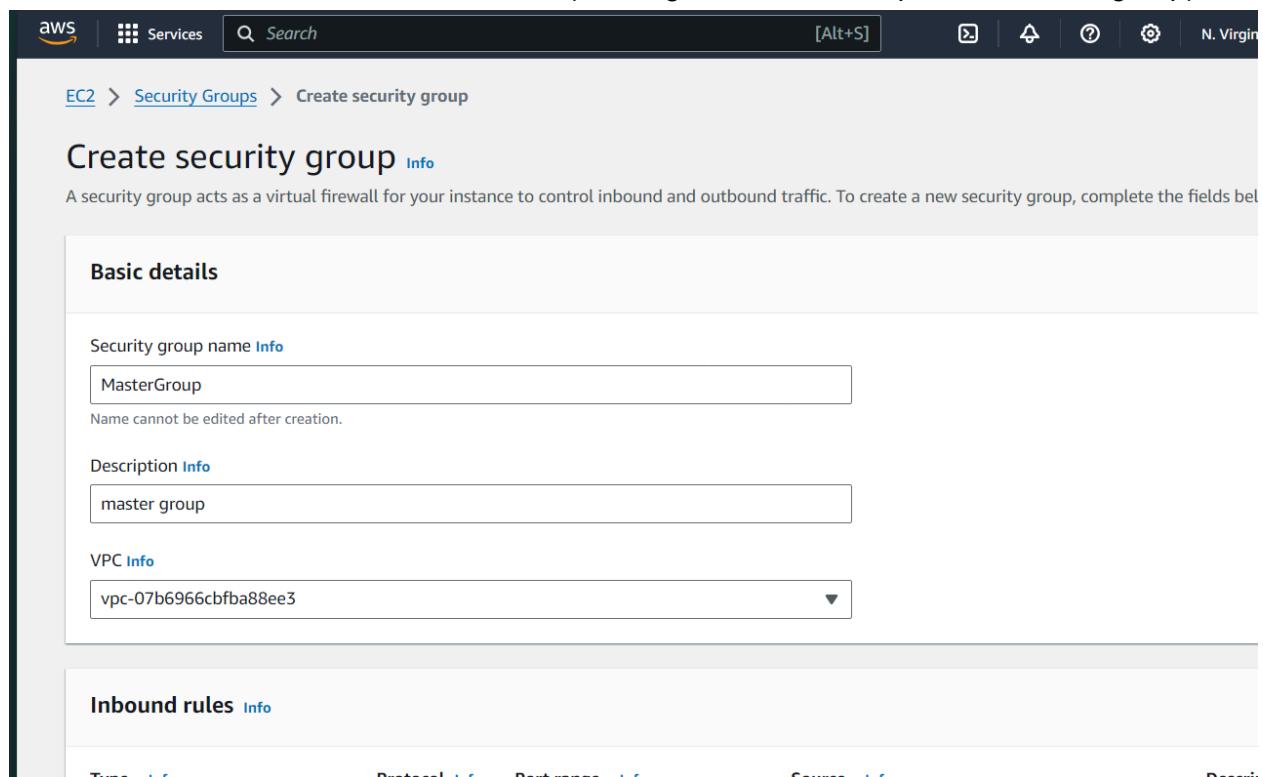
The screenshot shows the EC2 Dashboard. The left sidebar has sections for 'EC2 Dashboard', 'Events', 'Console-to-Code', 'Instances' (with sub-options like 'Instances', 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Capacity Reservations'), and 'Images'. The main content area is titled 'Resources' and displays the following information: 'You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:' with a table of counts: Instances (running) 5, Auto Scaling Groups 0, Capacity Reservations 0, Dedicated Hosts 0, Elastic IPs 0, Instances 5, Key pairs 2, Load balancers 0, Placement groups 0, Security groups 6, Snapshots 0, and Volumes 5. Below this is a 'Launch instance' section with a 'Launch instance' button, and a 'Service health' section with a 'AWS Health Dashboard' button. The right sidebar contains account-related information: 'Default vpc-07b6f', 'Settings Data prote Zones EC2 Serial Default cr EC2 conso', and 'Explore Save up to Optimize'.

3. create two secure groups one for master and other for the two nodes.



Security group ID	Security group name	VPC ID	Description
5013b4b74fb35de2	launch-wizard-1	vpc-07b6966cbfba88ee3	launch-wizard-1
1987c373fb6884a0	launch-wizard-2	vpc-07b6966cbfba88ee3	launch-wizard-2
Jc10dc4d51f60c8a	aws-cloud9-Cloud9-d788455f5a4d4b...	vpc-07b6966cbfba88ee3	Security group
154b0a819cb08ef2	launch-wizard-4	vpc-07b6966cbfba88ee3	launch-wizard-4
5fa7fae7b41178e3	default	vpc-07b6966cbfba88ee3	default VPC s
Jc387d915f325720	launch-wizard-3	vpc-07b6966cbfba88ee3	launch-wizard-3

4. enter details and add inbound rules (I have given MasterGroup for the master group)



EC2 > Security Groups > Create security group

Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name Info

Name cannot be edited after creation.

Description Info

VPC Info

Inbound rules Info

Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Descrip...

You have to look for the particular configuration which I did (in the image below)

Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>
HTTP	TCP	80	Anywhere... ▾	0.0.0.0/0 0.0.0.0/X
Custom TCP	TCP	6443	Anywhere... ▾	0.0.0.0/0 0.0.0.0/X
All traffic	All	All	Anywhere... ▾	0.0.0.0/0 0.0.0.0/X
Custom TCP	TCP	10251	Anywhere... ▾	0.0.0.0/0 0.0.0.0/X
Custom TCP	TCP	10252	Anywhere... ▾	0.0.0.0/0 0.0.0.0/X
Custom TCP	TCP	10250	Anywhere... ▾	0.0.0.0/0 0.0.0.0/X
All TCP	TCP	0 - 65535	Anywhere... ▾	0.0.0.0/0 0.0.0.0/X
SSH	TCP	22	Anywhere... ▾	0.0.0.0/0 0.0.0.0/X

click on create security group below.

now do the same for a node group.

Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>
All traffic	All	All	Anywhere... ▾	0.0.0.0/0 0.0.0.0/X
SSH	TCP	22	Anywhere... ▾	0.0.0.0/0 0.0.0.0/X
Custom TCP	TCP	10250	Anywhere... ▾	0.0.0.0/0 0.0.0.0/X
All TCP	TCP	0 - 65535	Anywhere... ▾	0.0.0.0/0 0.0.0.0/X
Custom TCP	TCP	30000 - 32767	Anywhere... ▾	0.0.0.0/0 0.0.0.0/X
HTTP	TCP	80	Anywhere... ▾	0.0.0.0/0 0.0.0.0/X

5. now go to ec2 and launch an instance

The screenshot shows the AWS EC2 Dashboard. On the left, a sidebar lists navigation options: EC2 Global View, Events, Console-to-Code (Preview), Instances (with sub-options: Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations (New)), Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes, Snapshots, Lifecycle Manager). The main content area is titled 'Resources' and displays usage statistics for the US East (N. Virginia) Region:

Category	Count
Instances (running)	5
Dedicated Hosts	0
Key pairs	2
Security groups	8
Auto Scaling Groups	
Elastic IPs	
Load balancers	
Snapshots	

Below the resources section is a 'Launch instance' panel with the following content:

- A large orange button labeled 'Launch instance'.
- A smaller button labeled 'Migrate a server'.
- A note: "To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud."
- A note: "Note: Your instances will launch in the US East (N. Virginia) Region".
- A 'Instance alarms' section with a 'View in CloudWatch' button.

add name and set ubuntu:

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
 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.

Recents | **Quick Start**

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Linux Enterprise Server

Amazon Machine Image (AMI)

▼ Summary

Number of instances
1

Software Image (/)
Canonical, Ubuntu
ami-0e86e20dae922

Virtual server type
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: Includes 750 hours of compute in the Region, up to 250 GB of storage, and 1000 requests per second for public IPv4. This tier is available for new accounts in the US East (N. Virginia) Region for the first month, 30 million IOs per month, 100 GB of data transfer, and 100 GB of internet.

create a key if you want

Create key pair

Key pair name
Key pairs allow you to connect to your instance securely.

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

Key pair type

RSA
RSA encrypted private and public key pair

ED25519
ED25519 encrypted private and public key pair

Private key file format

.pem
For use with OpenSSH

.ppk
For use with PuTTY

⚠ When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#) 

[Cancel](#) [Create key pair](#)

If you want you can reuse the key pair generated earlier.

Select the security group for master.

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

Search bar

Security Group Name	SG ID	VPC
launch-wizard-1	sg-06013b4b74fb35de2	VPC: vpc-07b6966cbfba88ee3
MasterGroup	sg-00c39d8526dda67f7	VPC: vpc-07b6966cbfba88ee3
launch-wizard-2	sg-04987c373fb6884a0	VPC: vpc-07b6966cbfba88ee3
aws-cloud9-Cloud9-d788455f5a4d4b4083454091233a80eb-		
InstanceSecurityGroup-OjiPSymDkJTu	sg-00c10dc4d51f60c8a	VPC: vpc-07b6966cbfba88ee3
launch-wizard-4	sg-0454b0a819cb08ef2	VPC: vpc-07b6966cbfba88ee3
default	sg-05fa7fae7b41178e3	VPC: vpc-07b6966cbfba88ee3
NodeGroup	sg-0cd15001d7a12fbdf	v the first 0 instance store

Compare security group rules

Advanced

or Magnetic

X

Name:Chinmay Chaudhari

D15C

Roll. No 6

then launch:

The screenshot shows the AWS Launch Wizard interface. On the left, under 'Configure storage', it shows a root volume of 8 GiB (gp3) and a note about free tier eligible customers getting up to 30 GB of EBS storage. Below this, there's a note about selecting AMI instance store volumes and a link to refresh backup information. On the right, the 'Software Image (AMI)' section is selected, showing Canonical, Ubuntu 24.04, amd64. It also shows the 'Virtual server type (instance type)' as t2.micro, 'Firewall (security group)' as MasterGroup, and 'Storage (volumes)' as 1 volume(s) - 8 GiB. A large callout box highlights the free tier benefits for the first year. At the bottom right are 'Cancel', 'Launch instance' (in orange), and 'Review commands' buttons.

do the same for node instance just select the number of instance as 2.
and select custom security group as node group.

This screenshot shows the AWS Launch Wizard for launching multiple instances. In the 'Network settings' section, the 'Number of instances' is set to 2. Under 'Security groups', the 'Select existing security group' button is selected, and 'NodeGroup' is chosen. The 'Software Image (AMI)', 'Virtual server type (instance type)', 'Firewall (security group)', and 'Storage (volumes)' sections are also visible. A callout box highlights the free tier benefits for the first year. At the bottom right are 'Cancel', 'Launch instance' (in orange), and 'Review commands' buttons.

don't give name now. and launch instance.

now go to instances and give name to the blank ones:

Instances (1/8) Info								
Last updated C Connect Instance state Actions Launch instances								
Find Instance by attribute or tag (case-sensitive) All states								
Name ↴	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IP
<input type="checkbox"/> Master	i-0ab175e9c60cc3a23	Running C Q	t2.micro	2/2 checks passed View alarms +	us-east-1b	ec2-184-73-147-45.co...	184.73.147.45	184.73.147.45
<input type="checkbox"/> node-1	i-08ad30b7114767ca2	Running C Q	t2.micro	Initializing View alarms +	us-east-1b	ec2-107-21-179-161.co...	107.21.179.161	107.21.179.161
<input checked="" type="checkbox"/> node-2	i-03c70d364fb762af5	Running C Q	t2.micro	Initializing View alarms +	us-east-1b	ec2-44-210-122-179.co...	44.210.122.179	44.210.122.179

6. select master and connect:

Instances (1/8) Info								
Last updated C Connect Instance state Actions Launch instances								
Find Instance by attribute or tag (case-sensitive) All states								
Name ↴	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IP
<input checked="" type="checkbox"/> Master	i-0ab175e9c60cc3a23	Running C Q	t2.micro	2/2 checks passed View alarms +	us-east-1b	ec2-184-73-147-45.co...	184.73.147.45	184.73.147.45

click on ssh client:

copy the command below the SSH client session

EC2 > Instances > i-0ab175e9c60cc3a23 > Connect to instance

Connect to instance [Info](#)

Connect to your instance i-0ab175e9c60cc3a23 (Master) using any of these options

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

Instance ID [i-0ab175e9c60cc3a23 \(Master\)](#)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is aws1331.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
`chmod 400 "aws1331.pem"`
4. Connect to your instance using its Public DNS:
`ec2-184-73-147-45.compute-1.amazonaws.com`

Command copied

[ssh -i "aws1331.pem" ubuntu@ec2-184-73-147-45.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](#)

7. Enter the copied command to a terminal window.

```
Microsoft Windows [Version 10.0.22631.4112]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Lenovo>ssh -i "C:\Users\Lenovo\Downloads\aws1331.pem" ubuntu@ec2-34-203-217-53.compute-1.amazonaws.com
The authenticity of host 'ec2-34-203-217-53.compute-1.amazonaws.com (34.203.217.53)' can't be established.
ED25519 key fingerprint is SHA256:3onu4BDyF+uS+Fwt16U1L99+OSyVYZbTNPiTw0Y074Y.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? |
```

It would prompt whether we want to continue connecting. type yes.

```
System information as of Fri Sep 27 15:25:58 UTC 2024

System load: 0.0          Processes:      104
Usage of /: 22.8% of 6.71GB  Users logged in:  0
Memory usage: 19%          IPv4 address for enX0: 172.31.87.211
Swap usage:  0%

Expanded Security Maintenance for Applications is not enabled.

No updates can be applied immediately.

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

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

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

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

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

ubuntu@ip-172-31-87-211:~$ |
```

The step is similar for node 1 and node 2 instances too. Just use different terminal windows.

node 1:

EC2 > Instances > i-08ad30b7114767ca2 > Connect to instance

Connect to instance Info

Connect to your instance i-08ad30b7114767ca2 (node-1) using any of these options

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

Instance ID
i-08ad30b7114767ca2 (node-1)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is aws1331.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
chmod 400 aws1331.pem
4. Connect to your instance using its Public DNS:
ec2-3-88-249-77.compute-1.amazonaws.com

Command copied

```
ssh -i "aws1331.pem" ubuntu@ec2-3-88-249-77.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

```
System load: 0.0          Processes:      104
Usage of /: 22.8% of 6.71GB  Users logged in:  0
Memory usage: 19%          IPv4 address for enX0: 172.31.89.24

Expanded Security Maintenance for Applications is not enabled.

No updates can be applied immediately.

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

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

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

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

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

ubuntu@ip-172-31-89-24:~$ |
```

node 2:

Connect to your instance i-03c70d364fb762af5 (node-2) using any of these options

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

Instance ID
[i-03c70d364fb762af5 \(node-2\)](#)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is aws1331.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
chmod 400 "aws1331.pem"
4. Connect to your instance using its Public DNS:
ec2-35-173-124-11.compute-1.amazonaws.com

Command copied

ssh -i "aws1331.pem" ubuntu@ec2-35-173-124-11.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.

```
System load: 0.08          Processes: 104
Usage of /: 22.8% of 6.71GB  Users logged in: 0
Memory usage: 19%          IPv4 address for enX0: 172.31.88.60
Swap usage: 0%
Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

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

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

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

ubuntu@ip-172-31-88-60:~$ |
```

8. From now on run the commands on all the 3 terminals unless instructed otherwise.

and the images (screen shots) will only be of master unless stated otherwise.

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

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

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

```
ubuntu@ip-172-31-87-211:~$ 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 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
```

now for node1 and node2:

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

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

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

ubuntu@ip-172-31-89-24:~$ 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)).
```

```
applicable law.

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

ubuntu@ip-172-31-88-60:~$ 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
```

sudo apt-get update

sudo apt-get install -y docker-ce

```
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.

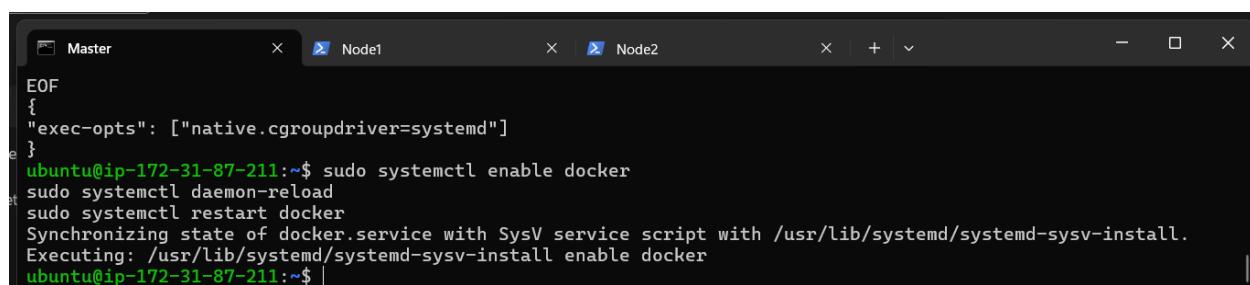
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-87-211:~$ |
```

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-87-211:~$ 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-172-31-87-211:~$ |
```

sudo systemctl enable docker

sudo systemctl daemon-reload

sudo systemctl restart docker



```
EOF
{
  "exec-opts": ["native.cgroupdriver=systemd"]
}
ubuntu@ip-172-31-87-211:~$ sudo systemctl enable docker
ubuntu@ip-172-31-87-211:~$ sudo systemctl daemon-reload
ubuntu@ip-172-31-87-211:~$ 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-172-31-87-211:~$ |
```

9. Run the below command to install Kubernets.

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

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

```
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-172-31-87-211:~$ 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-172-31-87-211:~$ |
```

Run the commands:

```
sudo apt-get update
```

```
sudo apt-get install -y kubelet kubeadm kubectl
```

```
sudo apt-mark hold kubelet kubeadm kubectl
```

```
ubuntu@ip-172-31-87-211:~$ 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
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
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]
```

```
No services need to be restarted.
```

```
No containers need to be restarted.
```

```
No user sessions are running outdated binaries.
```

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
```

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

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

```
ubuntu@ip-172-31-87-211:~$ sudo systemctl enable --now kubelet
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
  pigz slirp4netns
Use 'sudo apt autoremove' to remove them.

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-172-31-87-211:~$ |
```

sudo mkdir -p /etc/containerd

sudo containerd config default | sudo tee /etc/containerd/config.toml

```
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-172-31-87-211:~$ 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 = ""

[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-172-31-87-211:~$ |
```

sudo systemctl restart containerd

sudo systemctl enable containerd

sudo systemctl status containerd

```
Sep 27 16:24:28 ip-172-31-87-211 containerd[4900]: time="2024-09-27T16:24:28.133071347Z" level=info msg="serv>
Sep 27 16:24:28 ip-172-31-87-211 containerd[4900]: time="2024-09-27T16:24:28.133103323Z" level=info msg="serv>
Sep 27 16:24:28 ip-172-31-87-211 containerd[4900]: time="2024-09-27T16:24:28.133174859Z" level=info msg="Sta>
Sep 27 16:24:28 ip-172-31-87-211 containerd[4900]: time="2024-09-27T16:24:28.133199320Z" level=info msg="Sta>
Sep 27 16:24:28 ip-172-31-87-211 containerd[4900]: time="2024-09-27T16:24:28.133236780Z" level=info msg="Sta>
Sep 27 16:24:28 ip-172-31-87-211 containerd[4900]: time="2024-09-27T16:24:28.133244763Z" level=info msg="Sta>
Sep 27 16:24:28 ip-172-31-87-211 containerd[4900]: time="2024-09-27T16:24:28.133252776Z" level=info msg="Sta>
ubuntu@ip-172-31-87-211:~$ |
```

exit with **ctrl+c**.

sudo apt-get install -y socat

```
ubuntu@ip-172-31-87-211:~$ 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 libslirp4netns
  pigz slirp4netns
Use 'sudo apt autoremove' to remove them.
```

Run the following command in master only:

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

if it gives error use:

```
sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=NumCPU,Mem
```

```
to see the stack trace of this error execute with --v=5 or higher
ubuntu@ip-172-31-87-211:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=NumCPU,Mem
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
    [WARNING NumCPU]: the number of available CPUs 1 is less than the required 2
    [WARNING Mem]: the system RAM (957 MB) is less than the minimum 1700 MB
[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'
W0927 16:47:12.068193   6025 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.
```

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 172.31.87.211:6443 --token lhbvbx.kzlxu87vmv8hvxo \
    --discovery-token-ca-cert-hash sha256:1cef7709c45a42691a2ff0e44e3acf7f0e214fec7f4f822bb6818f3cf24ea4
3
ubuntu@ip-172-31-87-211:~$ |
```

Token and ca

Note: copy the text after kubeadm that you see at the later part like below:

```
kubeadm join 172.31.87.211:6443 --token lhbvbx.kzlxu87vmv8hvxo \
```

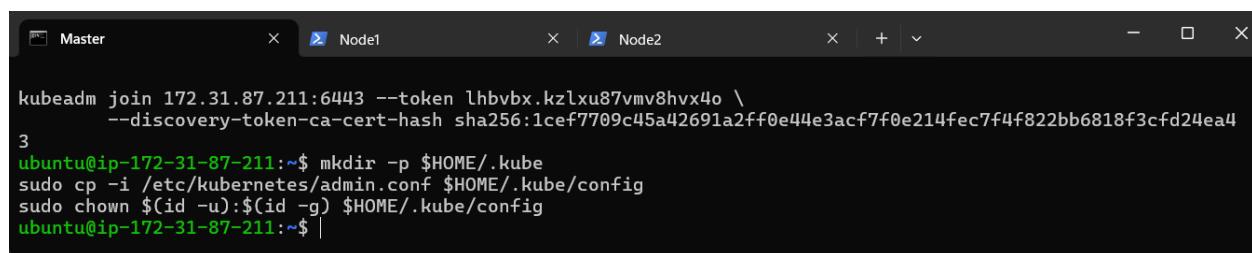
```
--discovery-token-ca-cert-hash
sha256:1cef7709c45a42691a2ff0e44e3acf7f0e214fec7f4f822bb6818f3cf24ea43
```

Run this command on master

```
mkdir -p $HOME/.kube
```

```
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
```

```
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```



```
Master          Node1          Node2
x  X  X
kubeadm join 172.31.87.211:6443 --token lhbvbx.kzlxu87vmv8hvxo \
    --discovery-token-ca-cert-hash sha256:1cef7709c45a42691a2ff0e44e3acf7f0e214fec7f4f822bb6818f3cf24ea4
3
ubuntu@ip-172-31-87-211:~$ mkdir -p $HOME/.kube
ubuntu@ip-172-31-87-211:~$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
ubuntu@ip-172-31-87-211:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@ip-172-31-87-211:~$ |
```

Now Run the command **kubectl get nodes** to see the nodes before executing Join

command on nodes.

```
ubuntu@ip-172-31-87-211:~$ kubectl get nodes
NAME           STATUS    ROLES      AGE   VERSION
ip-172-31-87-211  NotReady control-plane  3m48s  v1.31.1
```

Now paste the token and ca that I asked to copy earlier, on both the nodes.

use sudo before them.

it would be something like:

```
sudo kubeadm join <your-master-node-ip>:6443 --token <your-token>
--discovery-token-ca-cert-hash sha256:<your-ca-cert-hash>
(it has placeholders)
```

Node1:

```
ubuntu@ip-172-31-89-24:~$ sudo kubeadm join 172.31.87.211:6443 --token lhbvb8.kzlxu87vmv8hv4o \
--discovery-token-ca-cert-hash sha256:1cef7709c45a42691a2ff0e44e3acf7f0e214fec7f4f822bb6818f3cf24ea4
3
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
```

Node2:

```
ubuntu@ip-172-31-88-60:~$ sudo kubeadm join 172.31.87.211:6443 --token lhvbvx.kzlxu87vmv8hv4o \
--discovery-token-ca-cert-hash sha256:1cef7709c45a42691a2ff0e44e3acf7f0e214fec7f4f822bb6818f3cf24ea4
3
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 1.00236733s
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap
```

Step 9: Now Run the command on Master **kubectl get nodes** to see the nodes after executing Join command on nodes.

```
ubuntu@ip-172-31-87-211:~$ kubectl get nodes
NAME           STATUS    ROLES     AGE      VERSION
ip-172-31-87-211   NotReady   control-plane   7m35s   v1.31.1
ip-172-31-88-60   NotReady   <none>        12s     v1.31.1
ip-172-31-89-24   NotReady   <none>        32s     v1.31.1
ubuntu@ip-172-31-87-211:~$ |
```

Step 10: Since Status is NotReady we have to add a network plugin. And also we have to give the name to the nodes.

paste this command on master terminal. (and the following commands will be based on master unless states otherwise.

kubectl apply -f <https://docs.projectcalico.org/manifests/calico.yaml>

sudo systemctl status kubelet

again use ctrl+c to exit.

```

Master                               Node1                               Node2
Loaded: loaded (/usr/lib/systemd/system/kubelet.service; enabled; preset: enabled)
Drop-In: /usr/lib/systemd/system/kubelet.service.d
└─10-kubeadm.conf
Active: active (running) since Fri 2024-09-27 16:47:34 UTC; 17min ago
  Docs: https://kubernetes.io/docs/
Main PID: 6553 (kubelet)
  Tasks: 9 (limit: 1130)
 Memory: 72.2M (peak: 74.3M)
    CPU: 12.057s
   CGroup: /system.slice/kubelet.service
           └─6553 /usr/bin/kubelet --bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kubelet.conf --kubecon>
Sep 27 17:04:26 ip-172-31-87-211 kubelet[6553]: E0927 17:04:26.639597      6553 pod_workers.go:1301] "Error sy>
Sep 27 17:04:29 ip-172-31-87-211 kubelet[6553]: I0927 17:04:29.984518      6553 pod_container_deletor.go:80] ">
Sep 27 17:04:29 ip-172-31-87-211 kubelet[6553]: I0927 17:04:29.987738      6553 scope.go:117] "RemoveContainer>
Lines 1-16
ubuntu@ip-172-31-87-211:~$ |

```

Play a sound when a notification arrives

Now Run command **kubectl get nodes -o wide** we can see Status is ready.

NAME	STATUS	ROLES	AGE	VERSION	INTERNAL-IP	EXTERNAL-IP	OS-IMAGE
ip-172-31-87-211	Ready	control-plane	18m	v1.31.1	172.31.87.211	<none>	Ubuntu 24.04 LTS
ip-172-31-88-60	Ready	<none>	11m	v1.31.1	172.31.88.60	<none>	Ubuntu 24.04 LTS
6.8.0-1012-aws	containerd://1.7.12						
6.8.0-1012-aws	containerd://1.7.12						
ip-172-31-89-24	Ready	<none>	11m	v1.31.1	172.31.89.24	<none>	Ubuntu 24.04 LTS
6.8.0-1012-aws	containerd://1.7.12						

Lines 1-16
ubuntu@ip-172-31-87-211:~\$ |

Now to Rename run this command

Syntax: **kubectl label node <node-ip> kubernetes.io/role=worker**

examples:

Rename to Node 1:kubectl label node ip-<node1ip> kubernetes.io/role=Node1

Rename to Node 2:kubectl label node ip-<node2ip> kubernetes.io/role=Node2

```

ubuntu@ip-172-31-87-211:~$ kubectl label node ip-172-31-88-60 kubernetes.io/role=Node2
node/ip-172-31-88-60 labeled
ubuntu@ip-172-31-87-211:~$ kubectl label node ip-172-31-89-24 kubernetes.io/role=Node1
node/ip-172-31-89-24 labeled
ubuntu@ip-172-31-87-211:~$ |

```

Step 11: Run command **kubectl get nodes -o wide . And Hence we can see we have Successfully connected Node 1 and Node 2 to the Master.**

```
node/ip-172-31-89-24 labeled
ubuntu@ip-172-31-87-211:~$ kubectl get nodes -o wide
NAME           STATUS   ROLES      AGE    VERSION   INTERNAL-IP     EXTERNAL-IP   OS-IMAGE
KERNEL-VERSION CONTAINER-RUNTIME
ip-172-31-87-211   Ready    control-plane   24m   v1.31.1   172.31.87.211   <none>       Ubuntu 24.04 LTS
6.8.0-1012-aws    containerd://1.7.12
ip-172-31-88-60   Ready    Node2        17m   v1.31.1   172.31.88.60    <none>       Ubuntu 24.04 LTS
6.8.0-1012-aws    containerd://1.7.12
ip-172-31-89-24   Ready    Node1        17m   v1.31.1   172.31.89.24    <none>       Ubuntu 24.04 LTS
6.8.0-1012-aws    containerd://1.7.12
ubuntu@ip-172-31-87-211:~$ |
```

Conclusion:

In this Advanced DevOps Lab experiment, we began by setting up three EC2 Ubuntu instances on AWS, designating one as the Master node and the others as Worker nodes.

We then installed Docker and Kubernetes on all instances, ensuring Docker was properly configured.

The Kubernetes cluster was initialized on the Master node, and the Flannel networking plugin was applied to facilitate communication between nodes.

Finally, we joined the Worker nodes to the cluster using the provided token and hash, resulting in a fully operational Kubernetes cluster ready for managing and scaling containerized applications.

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.

[EC2](#) > [Security Groups](#) > Create security group

Create security group [Info](#)

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name [Info](#)
 Name cannot be edited after creation.

Description [Info](#)

VPC [Info](#)

Inbound rules [Info](#)

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

[Add rule](#)

click on create security group.

[AWS](#) [Services](#) [Search](#) [Alt+S]

EC2 Dashboard [X](#) Security group (sg-0f87a692e5cdada58 | exp4SecurityGroup) was created successfully [Details](#)

EC2 > Security Groups > sg-0f87a692e5cdada58 - exp4SecurityGroup

sg-0f87a692e5cdada58 - exp4SecurityGroup [Actions](#)

Details

Security group name exp4SecurityGroup	Security group ID sg-0f87a692e5cdada58	Description security for exp4	VPC ID vpc-07b6966cbfba88ee3
Owner 209322483715	Inbound rules count 3 Permission entries	Outbound rules count 1 Permission entry	

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

Inbound rules (3)

Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
-	sgr-0493fce359abad547	IPv4	SSH	TCP	22	0.0.0.0/0	-
-	ec2-0a91a7a0d0d0d0d0c	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 ENA 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

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

Free tier eligible ▾

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

▼ 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.

The screenshot shows the configuration interface for a Lambda function. It includes sections for 'Key pair (login)' and 'Network settings'.

Key pair (login)

- Description: 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

- Network: vpc-07b6966cbfba88ee3
- Subnet: No preference (Default subnet in any availability zone)
- Edit

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

Network settings [Info](#)

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 [↻](#) [Connect](#) [Inst](#)

Find Instance by attribute or tag (case-sensitive) [X](#) All states ▾

Instance ID = i-09426999c36422602 [X](#) Clear filters

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

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

EC2 > Instances > i-09426999c36422602 > Connect to instance

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.

```
Command Prompt - ssh -i C:\Users\Lenovo\Downloads\exp4.pem ubuntu@ec2-44-212-57-152.compute-1.amazonaws.com
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"
```

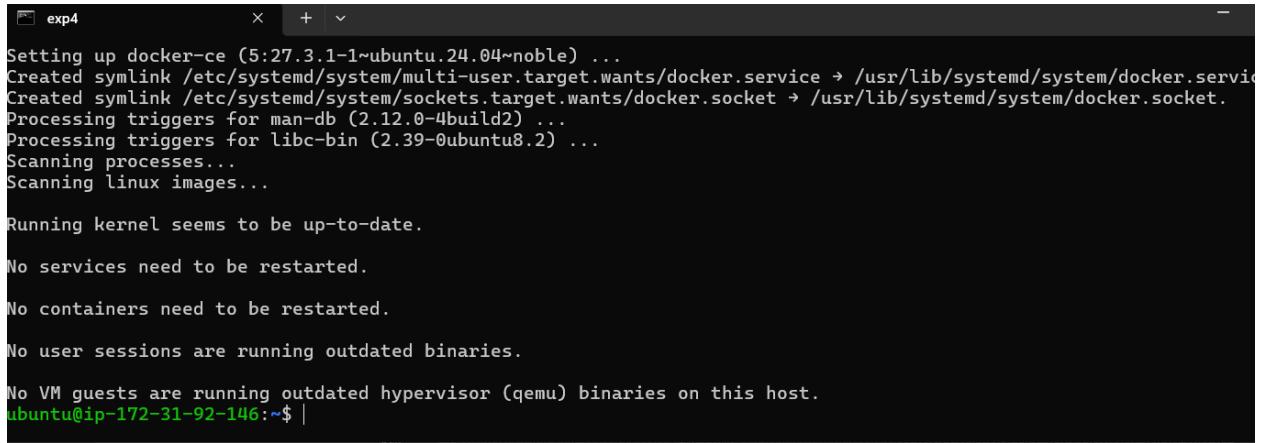
```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

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 [269 kB]
```

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
Setting up docker-ce (5:27.3.1-1~ubuntu.24.04~noble) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

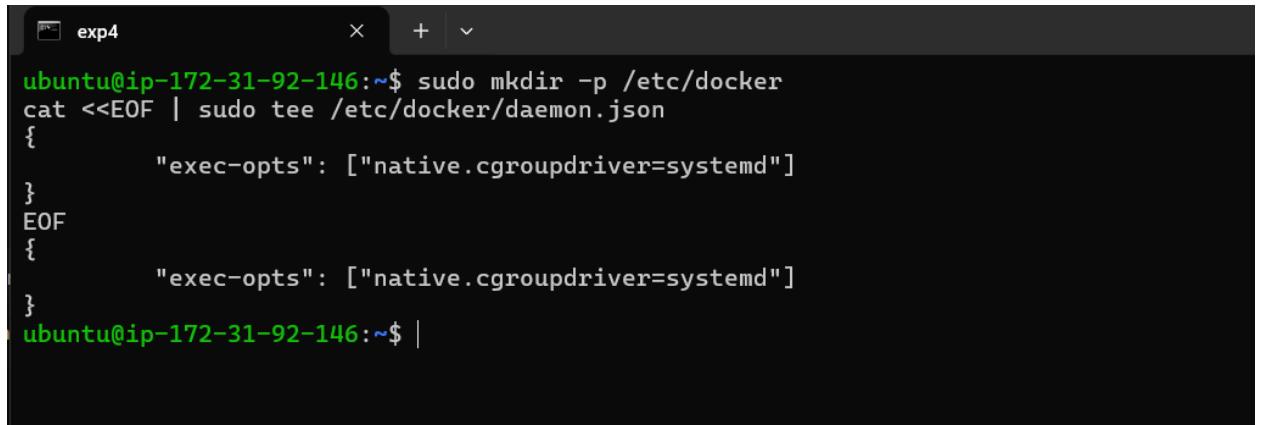
No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
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
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/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
Reading package lists... Done
Building dependency tree... Done
```

```

exp4
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 kubecluster

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

```

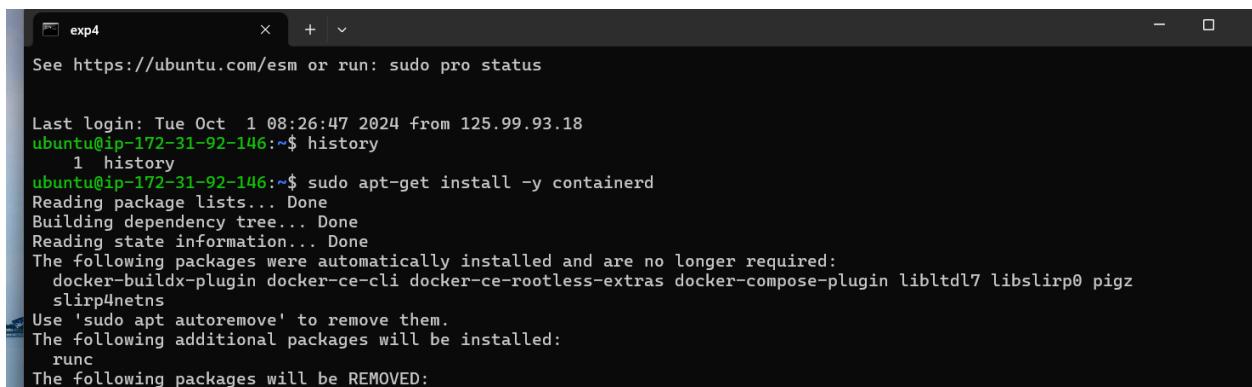
```

exp4
ubuntu@ip-172-31-92-146:~$ sudo systemctl enable --now kubelet
sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[kinit] 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 for endpoint "unix:///var/run/containerd/containerd.sock": rpc error: code = Unimplemented desc = unknown service runtime.v1.RuntimeService
[WARNINg FileExisting-socat]: socat not found in system path
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
error execution phase preflight: [preflight] Some fatal errors occurred:
failed to create new CRI runtime service: validate service connection: validate CRI v1 runtime API for endpoint "unix:///var/run/containerd/containerd.sock": rpc error: code = Unimplemented desc = unknown service runtime.v1.RuntimeService[preflight] If you know what you are doing, you can make a check non-fatal with '--ignore-preflight-errors=...'
To see the stack trace of this error execute with --v=5 or higher
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
```



```
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
  slirp4nets
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  runc
The following packages will be REMOVED:

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 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 subscri...
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
```

```
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 amd64 socat amd64 1.8.0.0-4build3 [374 kB]
Fetched 374 kB in 0s (9246 kB/s)
Selecting previously unselected package socat.
(Reading database ... 6019 files and directories currently installed.)
```

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
```

```
ubuntu@ip-172-31-92-146:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
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-kubelet-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.

```
[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.

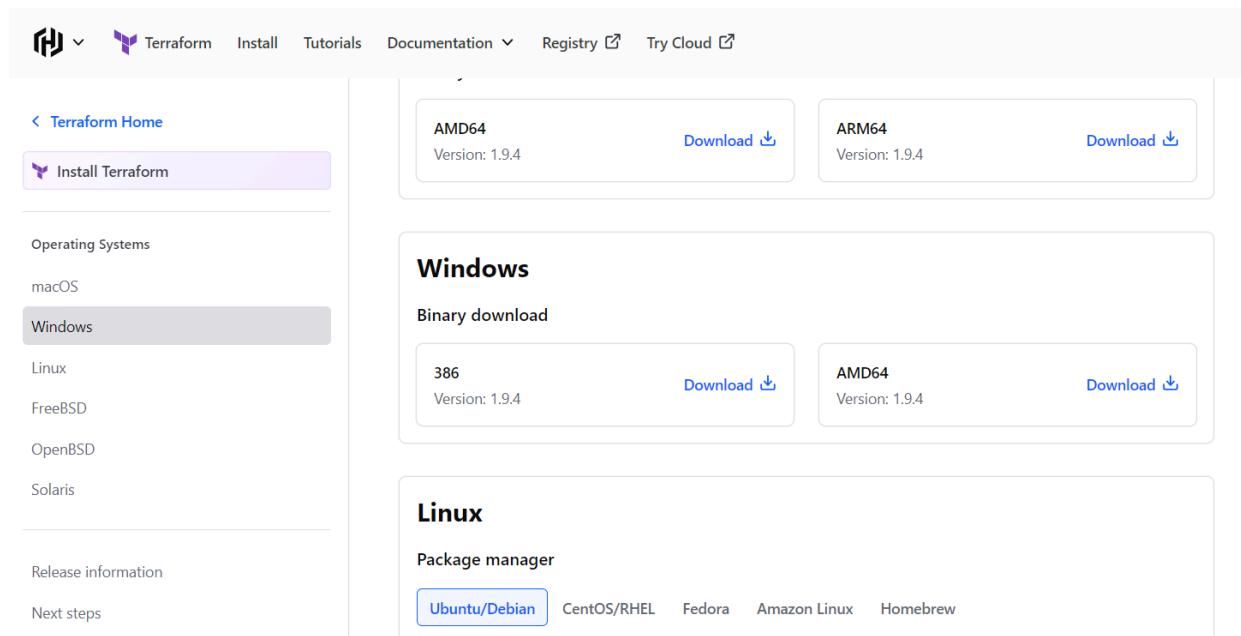
Installation guide of Terraform (Windows).

Step 1: Download terraform

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

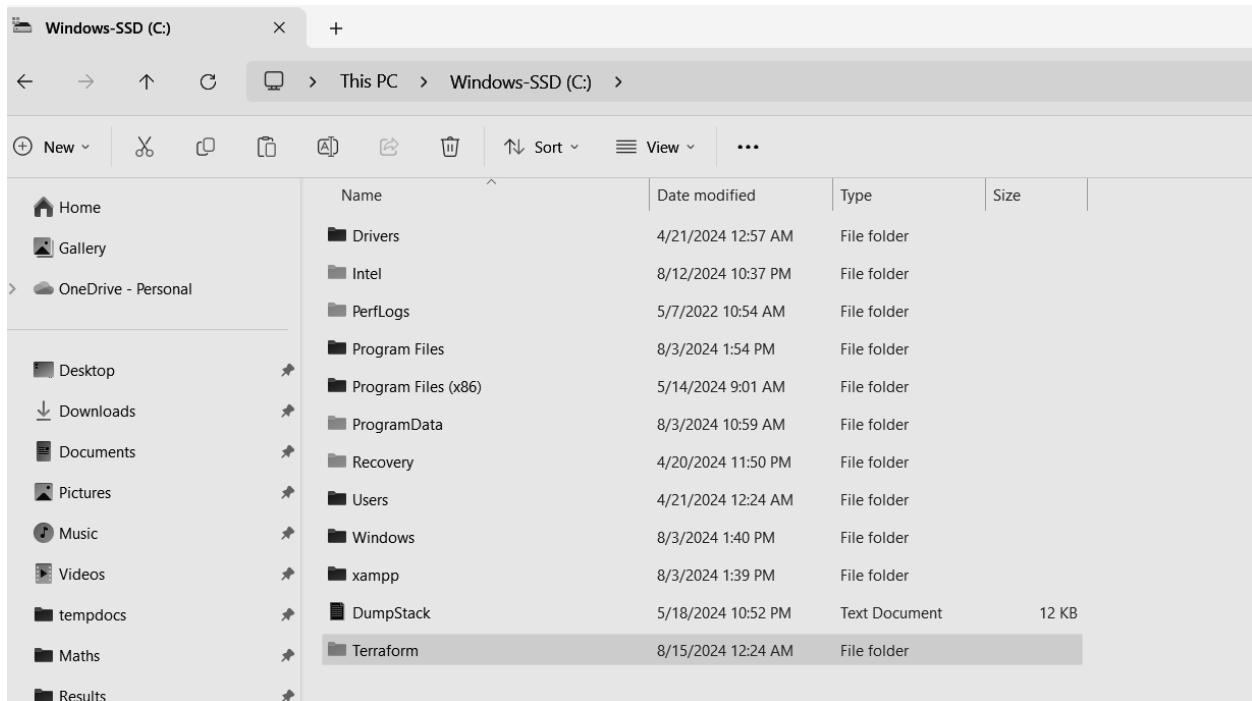
website:<https://www.terraform.io/downloads.html>

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



Step 2: Create a folder for Terraform:

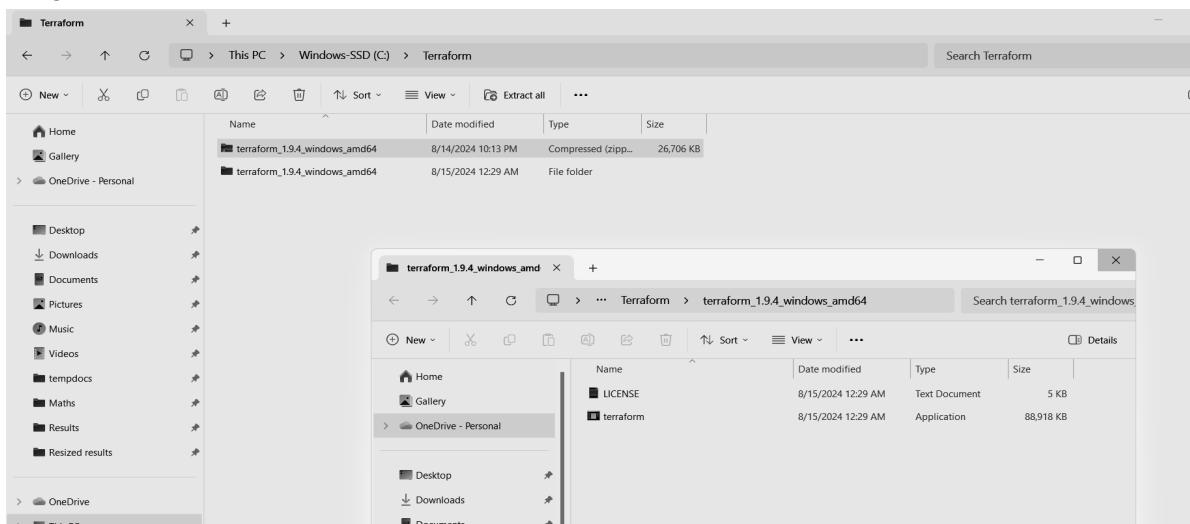
- Open **File Explorer** and navigate to **C:** or another location where you want to store Terraform.
- Right-click in the folder, select **New > Folder**, and name the folder something like **Terraform**.
- For example, the folder might be **C:\Terraform**.



Step 3: Extract Terraform into this folder:

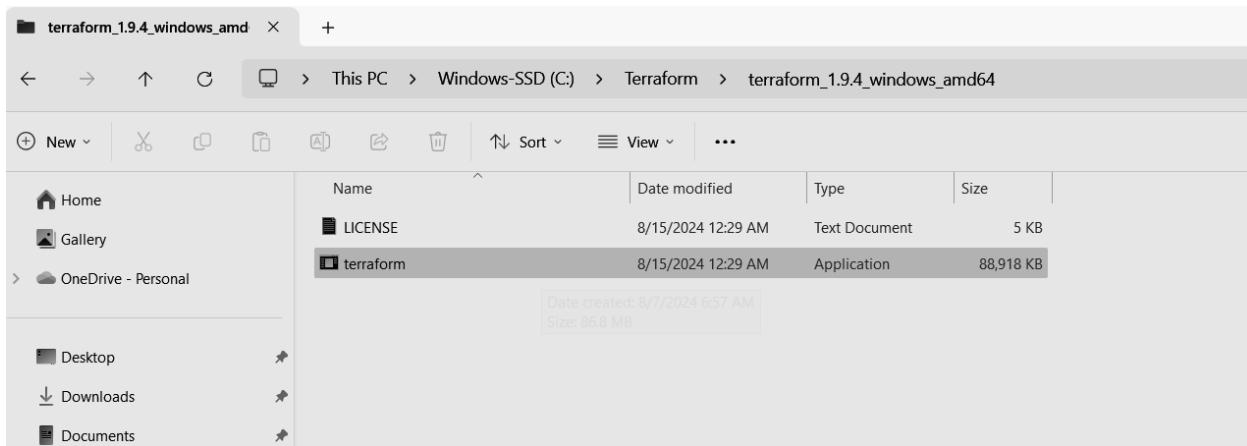
- Make sure to move the zip file from downloads to this new folder if it is not there already.
- After downloading the .zip file, right-click the .zip and choose **Extract All**.
- Extract the contents of the .zip file into the folder you just created (e.g., **C:\Terraform**).

*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.

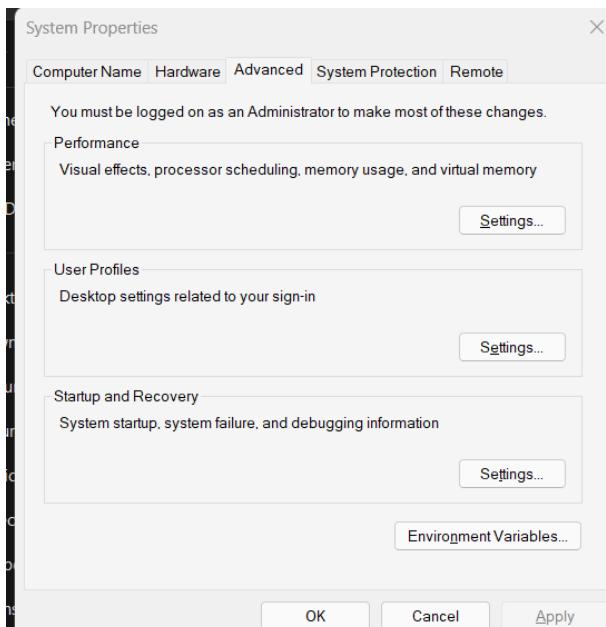


Step 4: Add to path

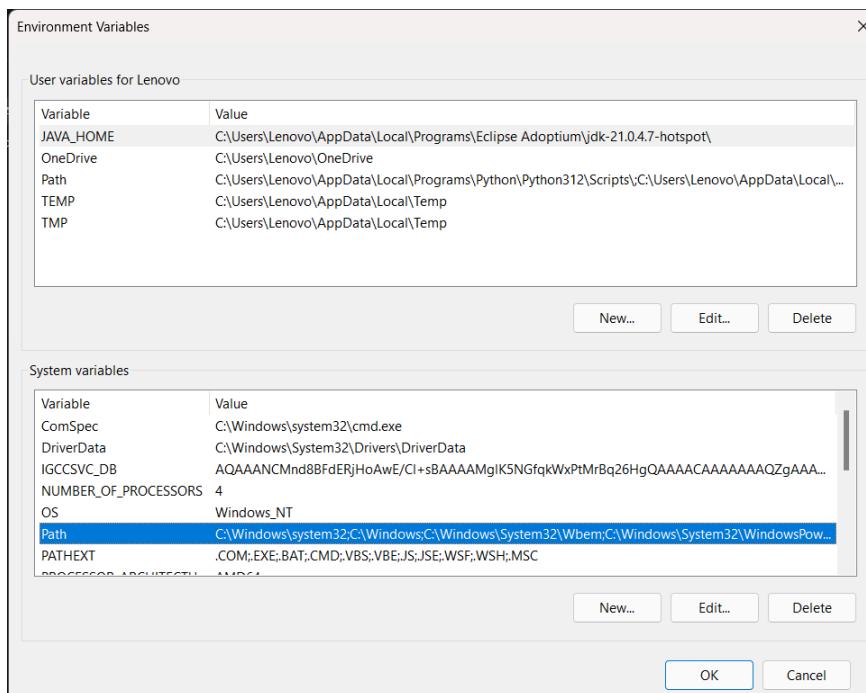
- Theory : the system uses top level search for finding the executable files, so rule of thumb is to add the directory which is a direct parent to the terraform exe.
- Hence, we first locate the terraform.exe and only copy the path till its parent directory. the highlighted file is the executable file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

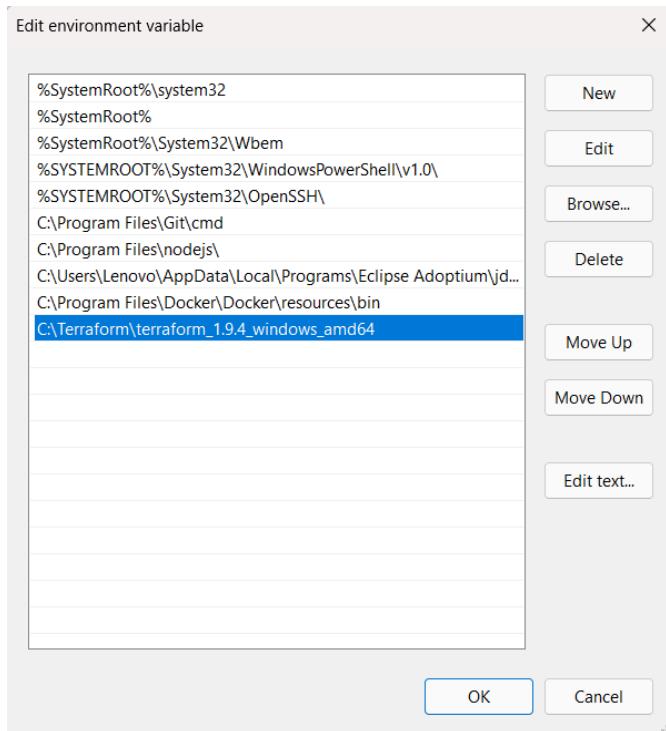


- Now select path from the system variables and click on edit.

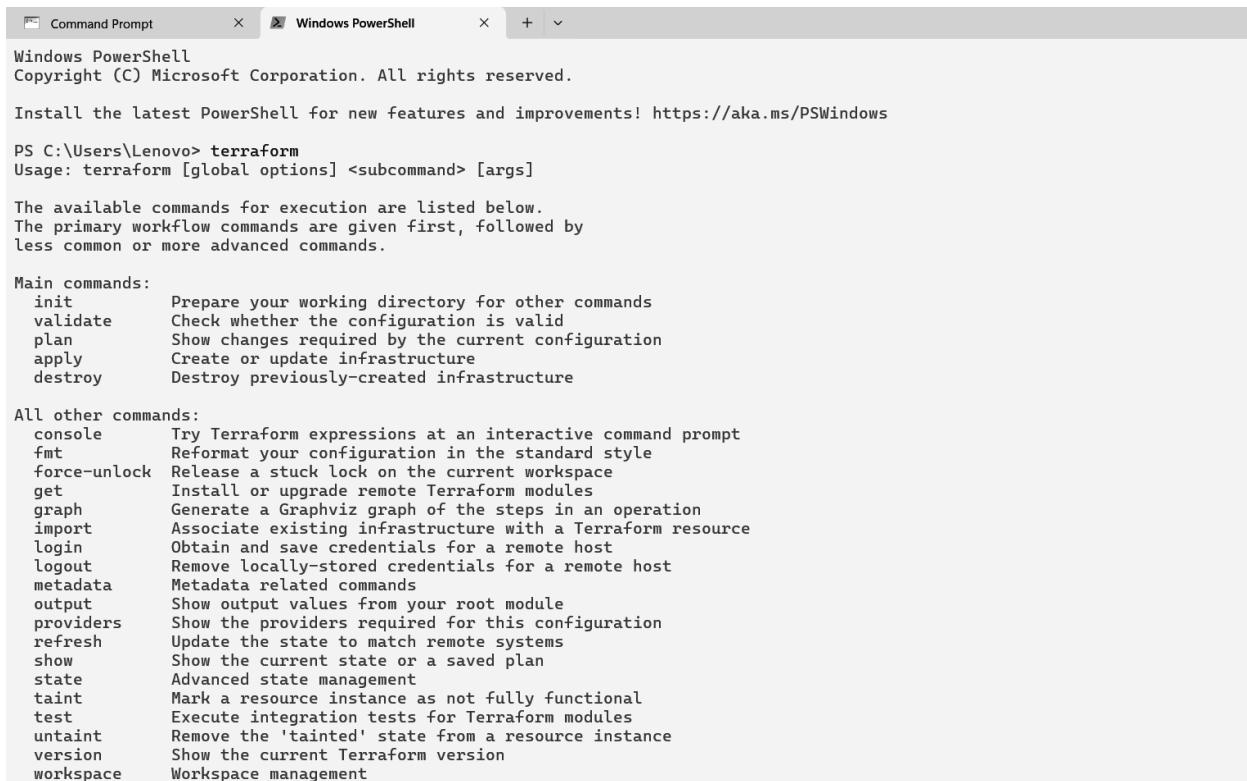


*Note click on edit and not new!

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



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



The screenshot shows a Windows PowerShell window titled "Windows PowerShell". The content displays the usage and available commands for Terraform. It includes sections for "Main commands:" and "All other commands:", each listing a command name and its description.

```
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: Chinmay Chaudhari
Div: D15C
Roll NO:06

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
```

```
docker.tf
```

```
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
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:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad76
tu:latest]

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

```

appy 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:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a]
ubuntu:latest]
docker_container.foo: Refreshing state... [id=f03a28e4658896c23c9992f7a98eb1011befc7d014e997ea9fc6372da70b7903]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated
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:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a"
  - 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:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a]
est]
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 links for 'New Item', 'Build History', 'Manage Jenkins', and 'My Views'. Below these are two expandable sections: 'Build Queue' (empty) and 'Build Executor Status'. The 'Build Executor Status' section shows one 'Built-In Node' named 'Sahil' which is 'Idle'. On the right, a table lists four builds: 'sahil 7' (last success 24 days ago, last failure N/A), 'Sahil exp6' (last success 24 days ago, last failure N/A), 'SahilExp6' (N/A), and 'sahiljob' (N/A). A search bar at the top right says 'Search (CTRL+K)'.

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀️	sahil 7	24 days #2	N/A	96 ms
✓	☀️	Sahil exp6	24 days #3	N/A	1 sec
🕒	☀️	SahilExp6	N/A	N/A	N/A
✗	🌧️	sahiljob	N/A	24 days #1	1.5 sec

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

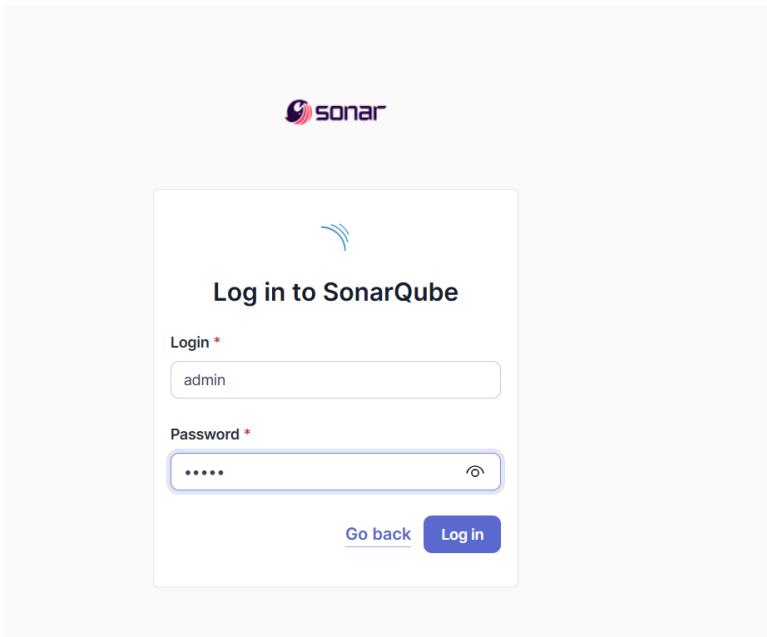
```
docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
```

-----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.

Name:Chinmay Chaudhari

Div:D15C

Roll No.6

(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

The screenshot shows the SonarQube interface for creating a project. At the top, there's a navigation bar with links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, More, and a search icon. Below the navigation, a heading asks "How do you want to create your project?". It provides options for importing from various platforms: Azure DevOps, Bitbucket Cloud, Bitbucket Server, GitHub, and GitLab, each with a "Setup" button. A note below these says "Are you just testing or have an advanced use-case? Create a local project." with a corresponding input field containing the text "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 *

 ✓

Project key *

 ✓

Main branch name *

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

CancelNext

Setup the project

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 pa
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.

Dashboard > Manage Jenkins > Plugins

Search (CTRL+K)

Plugins

Updates 29

Available plugins

Installed plugins

Advanced settings

Install	Name	Released
<input checked="" type="checkbox"/>	SonarQube Scanner 2.17.2	7 mo 8 days ago
<input type="checkbox"/>	External Site/Tool Integrations	
<input type="checkbox"/>	Build Reports	
<input type="checkbox"/>	Sonar Gerrit 388.v9b_f1cb_e42306	3 mo 22 days ago
<input type="checkbox"/>	External Site/Tool Integrations	
<input type="checkbox"/>	This plugin allows to submit issues from SonarQube to Gerrit as comments directly.	
<input type="checkbox"/>	SonarQube Generic Coverage 1.0	5 yr 1 mo ago
<input type="checkbox"/>	TODO	

Install

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>

SonarQube servers

If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

Environment variables

SonarQube installations

List of SonarQube installations

Name	sonarqube_exp7
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
ed

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

Dashboard > Manage Jenkins > Tools

The screenshot shows the Jenkins 'Tools' configuration page. At the top, there's a breadcrumb navigation: Dashboard > Manage Jenkins > Tools. Below this, there are five main sections, each with an 'Add [Tool Name]' button:

- Gradle installations**
- SonarScanner for MSBuild installations**
- SonarQube Scanner installations**
- Ant installations**
- Maven installations**

Below the Maven installations section, there are two buttons: **Save** (highlighted in blue) and **Apply**.

Click on **Add SonarQube Scanner** .

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

SonarScanner for MSBuild installations

Add SonarScanner for MSBuild

SonarQube Scanner installations

Add SonarQube Scanner

☰ SonarQube Scanner

Name

sonarqube_scanner_exp7

Install automatically ?

☰ Install from Maven Central

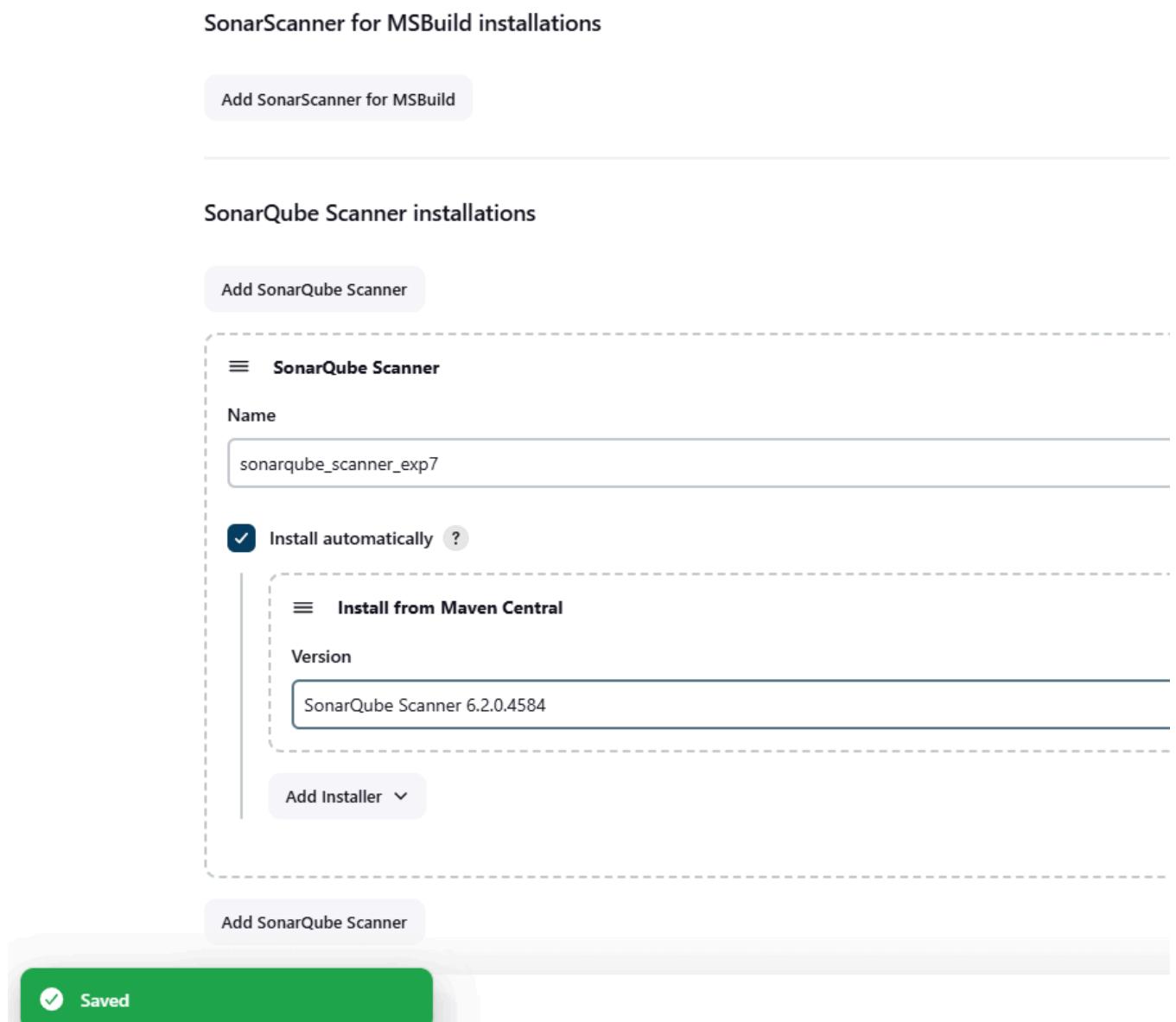
Version

SonarQube Scanner 6.2.0.4584

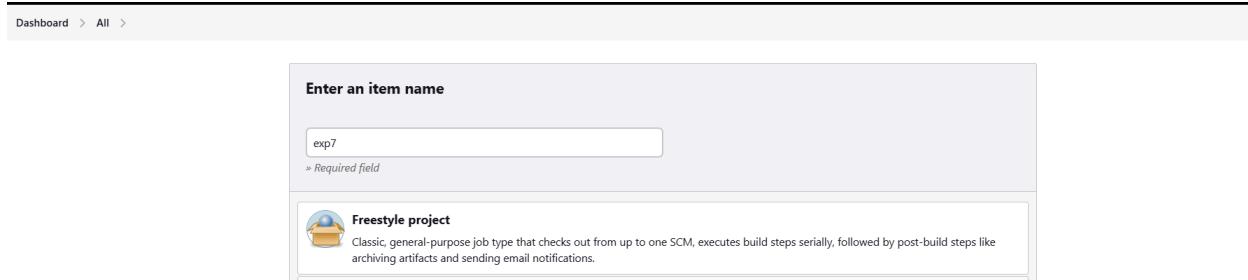
Add Installer ▾

Add SonarQube Scanner

 Saved



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

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

The screenshot shows the Jenkins configuration interface for the 'exp7' job. On the left, there's a sidebar with options: General, Source Code Management (which is selected), Build Triggers, Build Environment, Build Steps, and Post-build Actions. The main area is titled 'Source Code Management' and shows 'Git' selected. It has fields for 'Repository URL' (containing 'https://github.com/shazforiot/MSBuild_firstproject.git'), 'Credentials' (set to '- none -'), and 'Advanced' settings. At the bottom are 'Save' and 'Apply' buttons.

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

Execute SonarQube Scanner

Execute Windows batch command

Execute shell

Invoke Ant

Invoke Gradle script

Invoke top-level Maven targets

Run with timeout

Set build status to "pending" on GitHub commit

SonarScanner for MSBuild - Begin Analysis

SonarScanner for MSBuild - End Analysis

Add build step ^

Post-build Actions

Add post-build action ▾

Save **Apply**

You will see something like this:

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment**
- Build Steps**
- Post-build Actions

Build Steps

Execute SonarQube Scanner

JDK ? (Inherit From Job)

Path to project properties ?

Analysis properties ?

Additional arguments ?

JVM Options ?

Add build step ▾

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.

About this Project

Quality Gate used
(Default) [Sonar way](#)

Project Key ?
exp7

Visibility
Public

Description
No description added for this project.

Tags
No tags +

Notifications

A notification is never sent to the author of the event.

Send me an email for:

- Background tasks in failure
- Changes in issues/hotspots assigned to me
- Quality gate changes
- Issues resolved as false positive or accepted
- New issues
- My new issues

Badges

Show the status of your project metrics on your DOORSTEP or website. [Link your studio](#)

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

Additional arguments ?

Apply and save.

Name:Chinmay Chaudhari

Div:D15C

Roll No.6

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

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

Group/User	Administer System	Administer Quality Gates	Administer Quality Profiles	Create Projects
sonar-administrators	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
sonar-users	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Anyone DEPRECATED	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Administrator admin	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

12. Go to jenkins and click build:

The screenshot shows the Jenkins dashboard for the 'exp7' project. On the left, a sidebar lists project management options: Status, Changes, Workspace, Build Now (highlighted in grey), Configure, Delete Project, SonarQube, and Rename. Below this is the 'Build History' section, which displays a single build entry: '#1 Sep 25, 2024, 11:37 AM'. To the right of the build history is a SonarQube logo and a 'Permalinks' section with three downward-pointing arrows.

The screenshot shows the Jenkins console output for build #5. The sidebar on the left includes Status, Changes, Console Output (highlighted in grey), View as plain text, Edit Build Information, Delete build #5, Timings, Git Build Data, and Previous Build. The main area displays the command-line log of the SonarQube scanner execution. Key logs include:

```

Started by user Shubham Jha
Running as SYSTEM
Building on the built-in node in workspace C:\ProgramData\Jenkins\jenkins\workspace\exp7
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\jenkins\workspace\exp7\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/shafioriot/MSBuild_FirstProject # timeout=10
Fetching upstream changes from https://github.com/shafioriot/MSBuild_FirstProject
> git.exe --version # timeout=10
> git --version # git version 2.45.0.windows.1'
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision 12b042cb4c6e72427c380bcaceeddee7b49ad7 (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f f2b042cb4c6e72427c380bcaceeddee7b49ad7 # timeout=10
Commit message: "updated"
> git.exe rev-list --no-walk f2b042cb4c6e72427c380bcaceeddee7b49ad7 # timeout=10
[exp7] $ C:\ProgramData\Jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube_scanner_exp7\bin\sonar-scanner.bat -Dsonar.host.url=http://localhost:9000 -Dsonar.projectKey=exp7 -Dsonar.login=admin -Dsonar.host.url=http://localhost:9000 -Dsonar.sources=. -Dsonar.password=2923 -Dsonar.projectBaseDir=C:\ProgramData\Jenkins\jenkins\workspace\exp7
13:47:35.366 WARN Property 'sonar.host.url' with value 'http://localhost:9000' is overridden with value 'http://localhost:9000'
13:47:35.382 INFO Scanner configuration file: C:\ProgramData\Jenkins\jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube_scanner_exp7\bin..\conf\sonar-scanner.properties
13:47:35.382 INFO Project root configuration file: NONE
13:47:35.403 INFO SonarScanner CLI 6.2.0.4584
13:47:35.403 INFO Java 21.0.4 Oracle Corporation (64-bit)
13:47:35.403 INFO Windows 11 10.0 amd64
13:47:35.428 INFO User cache: C:\Windows\system32\config\systemprofile\.sonar\cache
13:47:37.015 INFO JRE provisioning: os[windows], arch[amd64]
13:47:47.097 INFO Communicating with SonarQube Server 10.6.0.92116
13:47:47.665 INFO Starting SonarScanner Engine...

```

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

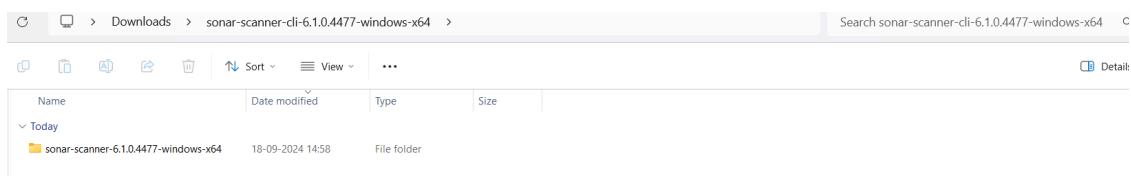
Step 1: Download sonar scanner

<https://docs.sonarsource.com/sonarqube/latest/analyzing-source-code/scanners/sonarscanner/>
Visit this link and download the sonarqube scanner CLI.



The screenshot shows the 'SonarScanner CLI' page from the SonarQube documentation. The left sidebar has a 'Scanners' section expanded, showing options like 'SonarScanner CLI'. The main content area displays version 6.1, released on 2024-06-27, which supports macOS and Linux AArch64 distributions. It provides download links for Linux x64, Linux AArch64, Windows x64, macOS x64, macOS AArch64, Docker, and Any (Requires a pre-installed JVM). Below this is a 'Release notes' section. A note at the bottom states: 'The SonarScanner CLI is the scanner to use when there is no specific scanner for your build system. 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.

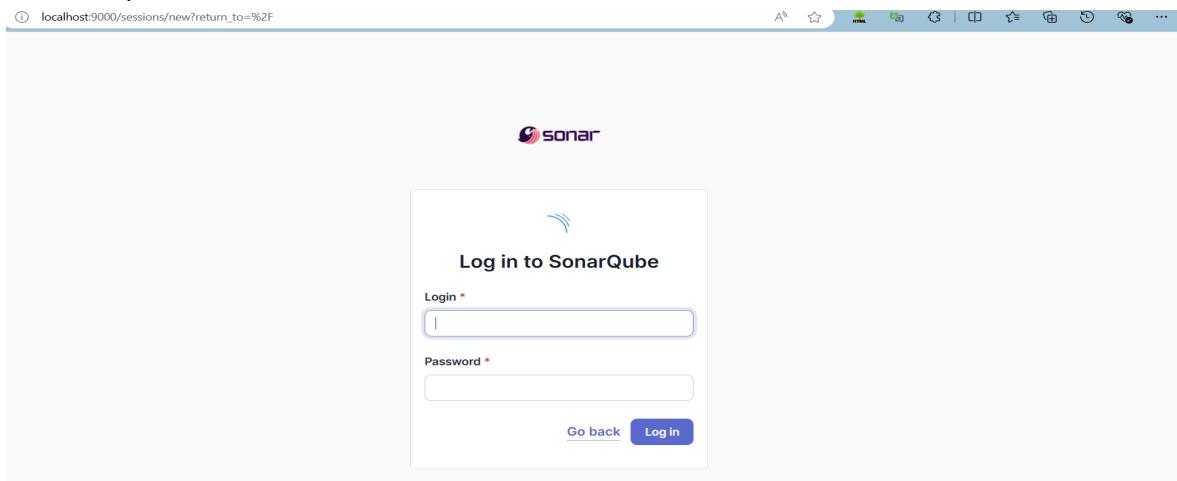


1. Install sonarqube image

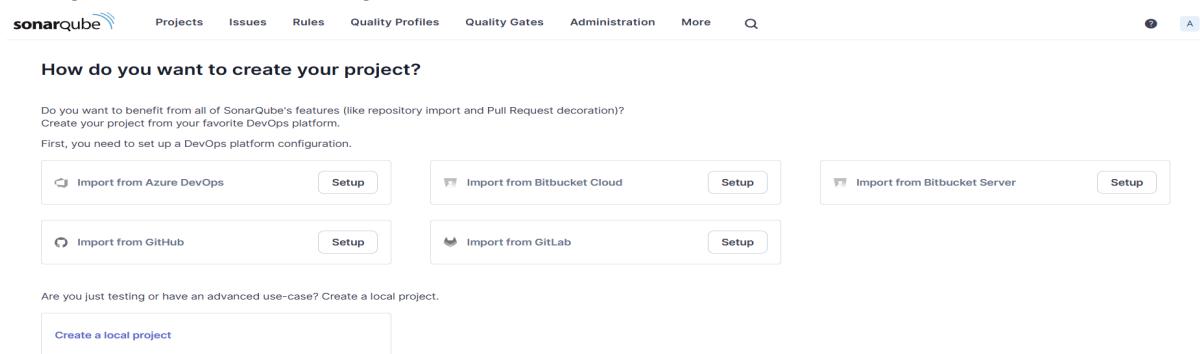
Command: **docker pull sonarqube** (skip if already installed we did install it in exp 7)
Then run the image

```
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 *

Project key *

Main branch name *

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

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 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

Build Queue: No builds in the queue.

Build Executor Status:

- 1 Idle
- 2 Idle
- Slave1 (offline)

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 Manage Jenkins > Plugins page shows the search results for "sonarq".

Install	Name	Released
<input type="checkbox"/>	SonarQube Scanner 2.17.2	6 mo 29 days ago
External Site/Tool Integrations Build Reports		
This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.		

Available plugins (25):

- Updates
- Available plugins
- Installed plugins
- Advanced settings

The screenshot shows the Jenkins 'Manage Jenkins' interface under the 'Plugins' section. On the left, there's a sidebar with options like 'Updates', 'Available plugins', 'Installed plugins', 'Advanced settings', and 'Download progress'. The 'Download progress' option is selected. On the right, it says 'Preparation' and lists three items: 'Checking internet connectivity', 'Checking update center connectivity', and 'Success'. Below that, it shows 'SonarQube Scanner' with a green checkmark and 'Success', and 'Loading plugin extensions' with another green checkmark and 'Success'. At the bottom, there are two links: 'Go back to the top page' and '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**

The screenshot shows the 'Add SonarQube' configuration form. It has fields for 'Name' (containing 'sonarqube_exp8'), 'Server URL' (containing 'http://localhost:9000'), and 'Server authentication token' (with a dropdown menu showing '- none -'). There's also an 'Advanced' button. At the bottom, there's a green 'Saved' button.

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.



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



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

Enter an item name

» Required field

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

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

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

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

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

Skiping 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/PropertyControlGui.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/PropertyControlGui.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/PropertyControlGui.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/PropertyControlGui.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/PropertyControlGui.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/PropertyControlGui.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/PropertyControlGui.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/PropertyControlGui.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/PropertyControlGui.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/PropertyControlGui.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/PropertyControlGui.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/PropertyControlGui.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/PropertyControlGui.html for block at line 668. Keep only references.

13. Now, check the project in SonarQube

Quality Gate: **Passed**

683k Lines of Code - Version not provided - Set as homepage

Last analysis 30 minutes ago

Security: 0 Open issues (A)

Reliability: 68k Open issues (C)

Maintainability: 164k Open issues (A)

Accepted issues: 0

Coverage: 0% lines to cover

Duplications: 50.6% on 759k lines

Security Hotspots: 3

14. Code Problems

- Consistency

Issues in new code

Consistency: 10%

Intentionality: 14k

Adaptability: 0

Responsibility: 0

Software Quality

Security: 0

Reliability: 54k

Maintainability: 164k

Bulk Change

Select issues: 196,662 Issues | 307sd effort

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

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

Reliability: Open Not assigned

Remove this deprecated "width" attribute.

Maintainability: Open Not assigned

Remove this deprecated "align" attribute.

Maintainability: Open Not assigned

Remove this deprecated "align" attribute.

Maintainability: Open Not assigned

1 of 5 Next

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

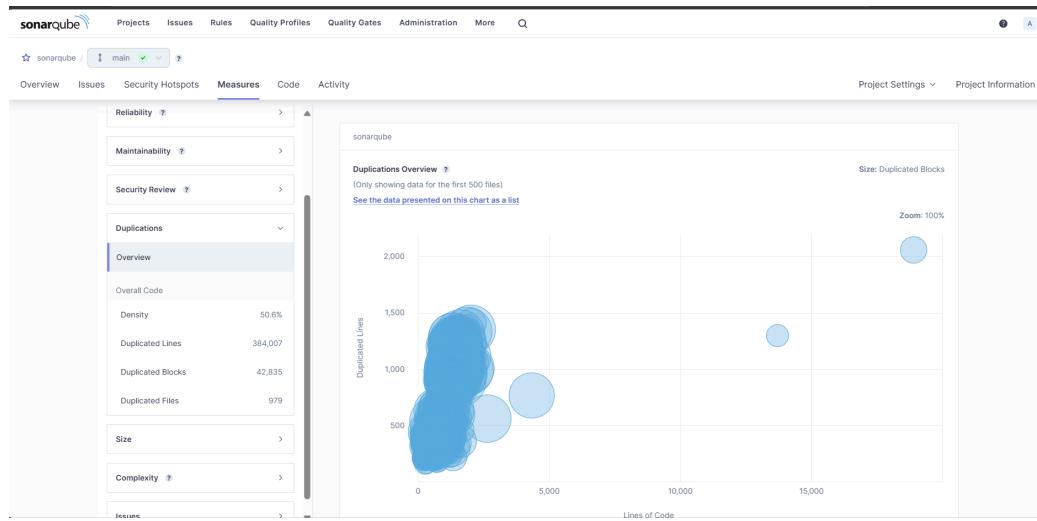
- Intentionality

Bugs

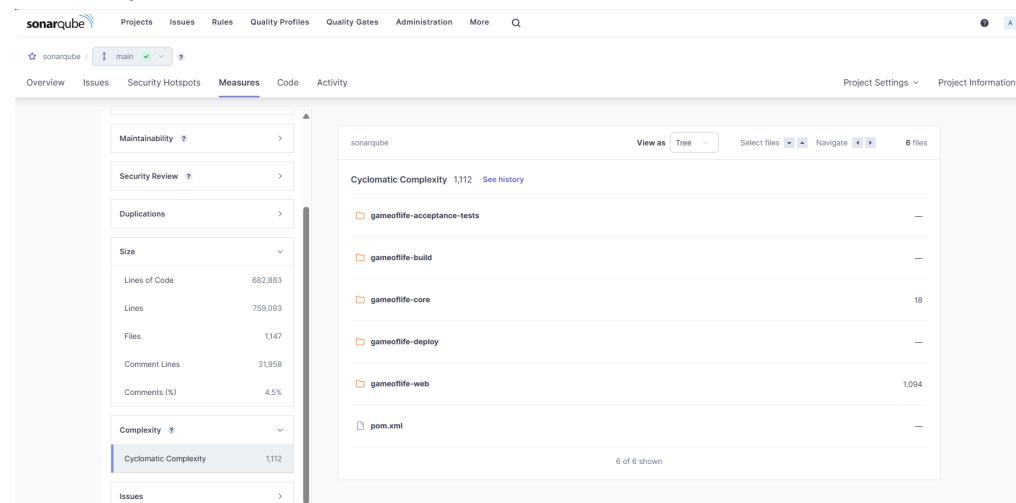
Bugs

Code Smells

- Duplications**



● 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
 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

Recents Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat S >

aws Mac ubuntu® Microsoft Red Hat S >

Q [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** '

Name : Chinmay Chaudhari

Div : D15C

Roll no:6



go to security groups:

The screenshot shows the AWS EC2 Instances page. The left sidebar has a collapsed navigation bar with the following items visible: 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, Network & Security, Security Groups (selected), and Elastic IPs.

The main content area displays the following table for Instances (3):

<input type="checkbox"/>	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 shown.

Name : Chinmay Chaudhari

Div : D15C

Roll no:6

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

The screenshot shows the AWS Management Console interface for managing security groups. At the top, there's a header with 'Security Groups (9)' and a 'Create security group' button. Below the header is a search bar with placeholder text 'Find resources by attribute or tag'. The main area displays a table with columns: 'Name', 'Security group ID', 'Security group name', and 'VPC ID'. The table lists nine entries:

Name	Security group ID	Security group name	VPC ID
-	sg-06013b4b74fb35de2	launch-wizard-1	vpc-07b6966cbfba88e
-	sg-00c39d8526dda67f7	MasterGroup	vpc-07b6966cbfba88e
-	sg-04987c373fb6884a0	launch-wizard-2	vpc-07b6966cbfba88e
aws-cloud9-Cloud9...	sg-00c10dc4d51f60c8a	aws-cloud9-Cloud9-d788455f5a4d4b...	vpc-07b6966cbfba88e
-	sg-0454b0a819cb08ef2	launch-wizard-4	vpc-07b6966cbfba88e
-	sg-05fa7fae7b41178e3	default	vpc-07b6966cbfba88e
-	sg-06ac4c5a9779ecaf9	launch-wizard-5	vpc-07b6966cbfba88e

now click on edit inbound rules

The screenshot shows the AWS Management Console interface for managing inbound rules. At the top, there's a header with 'Inbound rules (1)' and buttons for 'Manage tags' and 'Edit inbound rules'. Below the header is a search bar with placeholder text 'Search'. The main area displays a table with columns: 'Name', 'Security group rule...', 'IP version', 'Type', and 'Protocol'. The table lists one entry:

Name	Security group rule...	IP version	Type	Protocol
-	sgr-0d6a171458e586...	IPv4	SSH	TCP

Name : Chinmay Chaudhari

Div : D15C

Roll no:6

now do the following configurations:
by clicking "add rules"

Security group rule ID	Type	Info	Protocol	Info	Port range	Info	Source	Info	Description - optional	Info
sgr-0d6a171458e586b3e	SSH	TCP	22	Custom	0.0.0.0/0	X	Anywhere...	Custom	Delete	
-	HTTP	TCP	80	Anywhere...	::/0	X	Anywhere...	Custom	Delete	
-	All ICMP - IPv6	IPv6 ICMP	All	Anywhere...	::/0	X	Anywhere...	Custom	Delete	
-	HTTPS	TCP	443	Anywhere...	0.0.0.0/0	X	Anywhere...	Custom	Delete	
-	All traffic	All	All	Anywhere...	0.0.0.0/0	X	Anywhere...	Custom	Delete	
-	Custom TCP	TCP	5666	Anywhere...	0.0.0.0/0	X	Anywhere...	Custom	Delete	
-	All ICMP - IPv4	ICMP	All	Anywhere...	0.0.0.0/0	X	Anywhere...	Custom	Delete	

then click on save rules.

Details							
Security group name	Security group ID	Description	VPC ID				
launch-wizard-5	sg-06ac4c5a9779ecaf9	launch-wizard-5 created 2024-09-28T03:55:31.506Z	vpc-07b6966cbfa88ee5				
Owner	Inbound rules count	Outbound rules count					
209322483715	7 Permission entries	1 Permission entry					

Inbound rules (7)											
	Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description	Actions	Manage tags	Edit inbound rules
sg-057f6798fd0b60bb	sgr-057f6798fd0b60bb	IPv6	All ICMP - IPv6	IPv6 ICMP	All	::/0	-	-			
sgr-0d6a171458e586b3e	sgr-0d6a171458e586b3e	IPv4	SSH	TCP	22	0.0.0.0/0	-	-			
sgr-0b17ca9a96e3b5...	sgr-0b17ca9a96e3b5...	IPv4	HTTPS	TCP	443	0.0.0.0/0	-	-			
sgr-0d5d582940a2eba0	sgr-0d5d582940a2eba0	IPv4	Custom TCP	TCP	5666	0.0.0.0/0	-	-			
sgr-0e782e66d47b344f5	sgr-0e782e66d47b344f5	IPv4	All ICMP - IPv4	ICMP	All	0.0.0.0/0	-	-			
sgr-008da76767cd375...	sgr-008da76767cd375...	IPv6	HTTP	TCP	80	::/0	-	-			
sgr-0c81dac37a4a6020e	sgr-0c81dac37a4a6020e	IPv4	All traffic	All	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											
Find Instance by attribute or tag (case-sensitive)				Actions							
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs	
Master	i-0ab175e9c60cc5a23	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	ec2-54-165-203-193.co...	54.165.203.193	-	-	
node-1	i-08ad30b7114767ca2	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	ec2-52-23-200-179.co...	52.23.200.179	-	-	
node-2	i-03c70d364fb762af5	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	ec2-3-85-164-72.comp...	3.85.164.72	-	-	
nagios_host_e...	i-0820376be204a7fb	Running	t2.micro	2/2 checks passed	View alarms +	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.

The screenshot shows the 'Connect to instance' page for an EC2 instance. The instance ID is i-0820376be204a7fcb (nagios_host_exp_9kcs). The 'SSH client' tab is selected. The page provides instructions for connecting:

- Open an SSH client.
- Locate your private key file. The key used to launch this instance is nagios_exp_9.pem.
- Run this command, if necessary, to ensure your key is not publicly viewable.
`chmod 400 "nagios_exp_9.pem"`
- 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-ltdl       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`

Name : Chinmay Chaudhari

Div : D15C

Roll no:6

```
google-noto-sans-vf-fonts-20201206-2.amzn2023.0.2.noarch
graphite2-devel-1.3.14-7.amzn2023.0.2.x86_64
harfbuzz-devel-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.amzn2023.0.1.x86_64
libffi-devel-3.4-1.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
harfbuzz-icu-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

Name : Chinmay Chaudhari

Div : D15C

Roll no:6

```
[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:fef7:45ce
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2065473 (2.0M) [application/x-gzip]
Saving to: 'nagios-4.5.5.tar.gz'

nagios-4.5.5.tar.gz      100%[=====] 1.97M 5.30MB/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)

Name : Chinmay Chaudhari

Div : D15C

Roll no:6

```
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
[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    :
  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
```

Name : Chinmay Chaudhari

Div : D15C

Roll no:6

```
nagiosexp9 * + 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../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nagios.o ./nagios.c
gcc -Wall -I.. -I.. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o broker.o broker.c
gcc -Wall -I.. -I.. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nebmods.o nebmod.o
gcc -Wall -I.. -I.. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o ../common/shared.o
./common/shared.c
gcc -Wall -I.. -I.. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o query-handler.o
query-handler.c
gcc -Wall -I.. -I.. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o workers.o worker.o
In function 'get_wproc_list':
    while 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

Name : Chinmay Chaudhari

Div : D15C

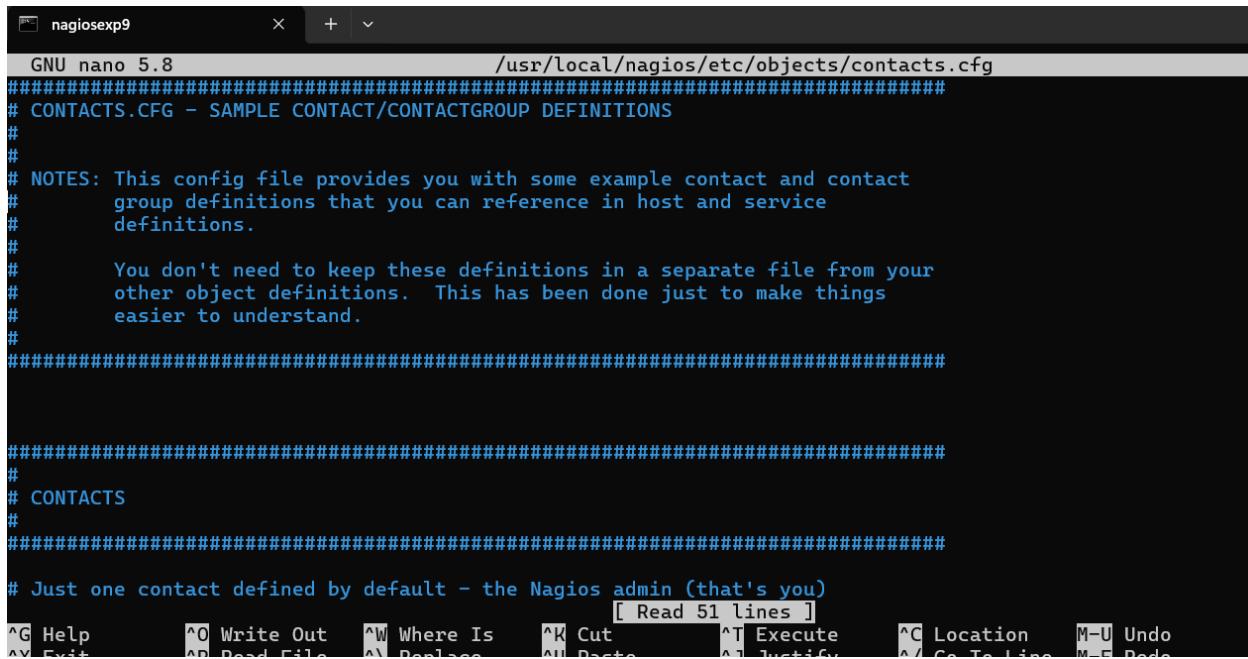
Roll no:6

```
[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 nagiosstats /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
```



```
nagiosexp9      x  +  v
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
^X Exit      ^R Read File   ^L Replace    ^U Paste      ^J Justify    ^V 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 *****

#####
#
```

Name : Chinmay Chaudhari

Div : D15C

Roll no:6

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

}

#####
#
# CONTACT GROUPS

^G Help      ^O Write Out   ^W Where Is   ^K Cut      ^T Execute   ^C Location   M-U Undo   M-A Set M
```

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

Name : Chinmay Chaudhari

Div : D15C

Roll no:6

```
[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

```
[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
```

```
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/plugins-root'
make install in po
make[1]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/po'
/usr/bin/mkdir -p /usr/local/nagios/share
installing fr.gmo as /usr/local/nagios/share/locale/fr/LC_MESSAGES/nagios-plugins.mo
installing de.gmo as /usr/local/nagios/share/locale/de/LC_MESSAGES/nagios-plugins.mo
if 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; \
else \
:; \
fi
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/po'
make[1]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[2]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[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
```

```
** 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 commad (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

Name : Chinmay Chaudhari

Div : D15C

Roll no:6

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. A tooltip 'Public IPv4 address copied' is displayed over the copy icon next to the public IP address 54.224.175.95. Other visible instance details include Instance ID (i-0820376be204a7fcf), Hostname type (IP name: ip-172-31-80-137.ec2.internal), and Instance type (t2.micro).

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/. A 'Sign in to access this site' dialog box is open, prompting for a Username (nagiosadmin) and a Password (****). The browser interface includes standard navigation buttons like back, forward, and search.

The screenshot shows the Nagios Core web interface. At the top, there's a navigation bar with a 'Sign in' button, a search bar containing 'Nagios: 54.224.175.95', and a note 'Not secure | 54.224.175.95/nagios/'. Below the header, the Nagios logo is displayed with the text 'Nagios® Core™ Version 4.5.5' and the date 'September 17, 2024'. A green checkmark indicates 'Daemon running with PID 3152'. The left sidebar contains several sections: '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), and 'System' (Comments, Downtime, Process Info, Performance Info). The main content area includes a 'Get Started' section with a bulleted list: Start monitoring your infrastructure, Change the look and feel of Nagios, Extend Nagios with hundreds of addons, Get support, Get training, Get certified. There are also 'Latest News' and 'Don't Miss...' sections, both of which are currently empty. On the right, a 'Quick Links' section lists various Nagios resources. At the bottom of the page, a copyright notice reads: '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 AWS EC2 'Launch an instance' wizard. On the left, there's a navigation bar with 'EC2 > Instances > Launch an instance'. The main area has two tabs: 'Name and tags' (selected) and 'Application and OS Images (Amazon Machine Image)'. In the 'Name and tags' tab, the 'Name' field contains 'exp10client'. Below it, there's a 'Search our full catalog including 1000s of application and OS images' input field and a 'Quick Start' button. Under 'Application and OS Images (Amazon Machine Image)', there's a list of AMI categories: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE. To the right, a summary panel shows: 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), and a note about the Free tier. A 'Cancel' button is at the bottom right of the summary panel.

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.

Recents **Quick Start**

[Browse more AMIs](#)

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-0e86e20dae...

Virtual server type: t2.micro

Firewall (security groups): launch-wizard-1

Storage (volumes): 1 volume(s) - 8

Free tier
750 hours in the Region. The Region is unavailable for tier AMI. Public IP address per month, up to 1 million IP addresses. 100 GB of internet bandwidth.

[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 [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](#)

Free tier
750 hours in the Region. The Region is unavailable for tier AMI. Public IP address per month, up to 1 million IP addresses. 100 GB of internet bandwidth.

[Cancel](#)

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)

The screenshot shows the AWS EC2 Instances page with the following details:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
Master	i-0ab175e9c60cc3a23	Running	t2.micro	2/2 checks passed	View alarms	us-east-1b	ec2-3-82-156-160.com...
node-1	i-08ad30b7114767ca2	Running	t2.micro	2/2 checks passed	View alarms	us-east-1b	ec2-3-85-110-80.comp...
node-2	i-03c70d364fb762af5	Running	t2.micro	2/2 checks passed	View alarms	us-east-1b	ec2-54-226-209-38.co...
nagios_host_e...	i-0820376be204a7fcf	Running	t2.micro	2/2 checks passed	View alarms	us-east-1b	ec2-54-224-175-95.co...
exp10client	i-0994ca5a178801a54	Running	t2.micro	Initializing	View alarms	us-east-1b	ec2-54-173-58-143.co...

Below the table, the 'Connect to instance' dialog is displayed for the instance 'exp10client'. It includes tabs for EC2 Instance Connect, Session Manager, SSH client (selected), and EC2 serial console. The SSH client tab shows the instance ID 'i-0994ca5a178801a54 (exp10client)' and a list of connection steps. Step 4 provides a command to run: 'ssh -i "nagios_exp_9.pem" ubuntu@ec2-54-173-58-143.compute-1.amazonaws.com'. A tooltip indicates the command has been copied. A note at the bottom states: '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.'

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

The screenshot shows the AWS CloudWatch Metrics Insights interface with the following details:

- 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 IPv4 address:** 54.173.58.143 (highlighted and copied)
- 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
- 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
    ; This host definition will inherit
    ; in (or inherited by) the linux-server
    host_name       linuxserver
    alias           linuxserver
    address         54.173.58.143
}
```

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 hostnames
}
```

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

host_name	localhost
service_description	PING

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 CGTs read 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]# |
```

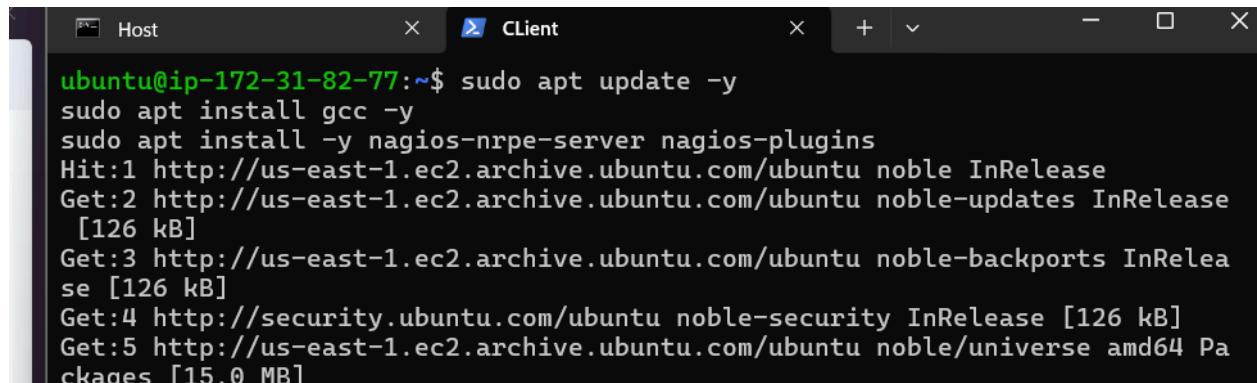
3) Go to client machine (ubuntu machine)

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 "Host" and "Client". The terminal shows the following command and its execution:

```
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 the text "Nagios® Core™ Version 4.5.5" and the date "September 17, 2024". A green checkmark message says "Daemon running with PID 13935". On the left, there's a sidebar with sections for General (Home, Documentation), Current Status (Tactical Overview, Map, Hosts, Services, Host Groups, Grid, Service Groups, Summary, Grid), Reports (Availability, Trends, Alerts, History, Summary, Histogram, Notifications, Event Log), and System (Comments, Downtime Info, Process Info, Performance Info, Scheduling Queue, Configuration). The main content area has a "Get Started" box with links to monitoring infrastructure, changing look, extending Nagios, getting support, training, and certification. It also includes "Latest News" and "Don't Miss..." boxes. A "Quick Links" box lists Nagios Library, Labs, Exchange, Support, and the official website. At the bottom, there's a copyright notice and a "Nagios" footer logo.

Click on hosts

This screenshot shows the "Current Status" page of the Nagios Core interface. The left sidebar has a vertical list of links: "Tactical Overview" (highlighted in blue), "Map", "Hosts" (highlighted in blue), "Services", and "Host Groups". The main content area is currently empty, showing a large white space.

5) Click on linux server

Nagios®

Current Network Status

Last Updated: Sat Sep 28 11:33:24 UTC 2024
Updated every 90 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	Down	Unreachable	Pending
2	0	0	0
All Problems	All Types		
0	2		

Service Status Totals

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

Host Status Details For All Host Groups

Host	Status	Last Check	Duration	Status Information
linuxserver	UP	09-28-2024 11:29:10	0d 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

Host Information

Last Updated: Sat Sep 28 11:33:39 UTC 2024
Updated every 90 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 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 Downtime Change:	09-28-2024 11:24:48
Last Notification:	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)

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

Host Commands

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

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

Current Network Status

Last Update: Sat Sep 29 11:33:58 UTC 2024

Updated every 90 seconds

Nagios Core™ 4.5.6 - www.nagios.org

Logged in as nagiostest

View History For All hosts

View Notifications For All Hosts

View Host Status Detail For All Hosts

Host Status Totals

Up	Down	Unreachable	Pending
2	0	0	0

All Problems All Types

0	2
---	---

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
12	1	0	3	0

All Problems All Types

4	16
---	----

Service Status Details For All Hosts

Limit Results: 100 ▾

Host	Service	Status	Last Check	Duration	Attempt	Status Information
linuxserver	Current Load	OK	09-28-2024 11:30:25	0d 0h 6m 33s	1/4	OK - load average: 0.01, 0.00, 0.00
	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-Subuntu13.4 (protocol 2.0)
localhost	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	PROCS OK. 37 processes with STATE = RSZDT
	Current Load	OK	09-28-2024 11:29:39	0d 0h 53m 5s	1/4	OK - load average: 0.02, 0.01, 0.00
	Current Users	OK	09-28-2024 11:30:17	0d 0h 52m 27s	1/4	USERS OK - 2 users currently logged in
localhost	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 5m 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	PROCS OK. 37 processes with STATE = RSZDT

Results 1 - 16 of 16 Matching Services

Reports

Availability

Trends

Alerts

 History

 Summary

 Histogram

Notifications

Event Log

System

Comments

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.

Errors and Solutions

During the setup, a few common issues were encountered:

1. **Apache Server Was Not Running:** When connecting to the Nagios host machine after restarting an instance, the Apache server was found to be inactive. This was resolved by running the `sudo service httpd start` command to restart the server, ensuring that the Nagios web interface was accessible.
2. **Directory Creation Issue:** While copying the configuration files, we initially encountered a "No such file or directory" error. This was resolved by creating the necessary directory structure (`/usr/local/nagios/etc/objects/monitorhosts/linuxhosts`) before proceeding with the file copy.
3. **NRPE Configuration Error:** On the client machine, forgetting to update the `allowed_hosts` field in the `nrpe.cfg` file with the Nagios host's public IP resulted in connectivity issues. This was fixed by editing the file to include the correct IP address.

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.

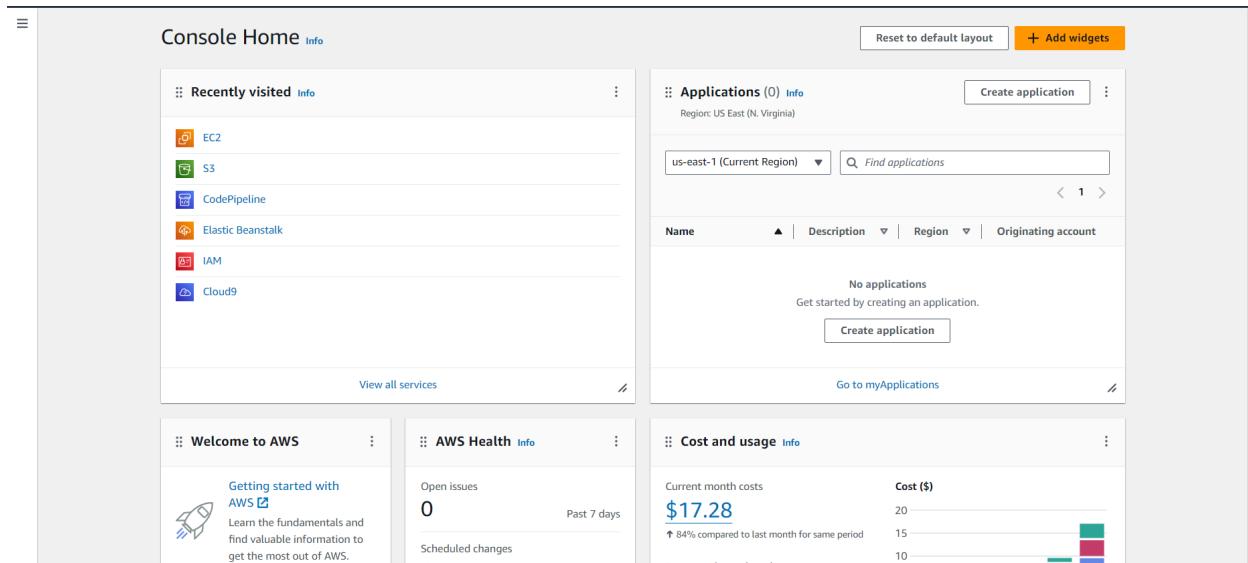
• **Node.js:** Excellent for I/O-bound tasks like handling APIs or streaming data, with fast startup times and efficient memory usage.

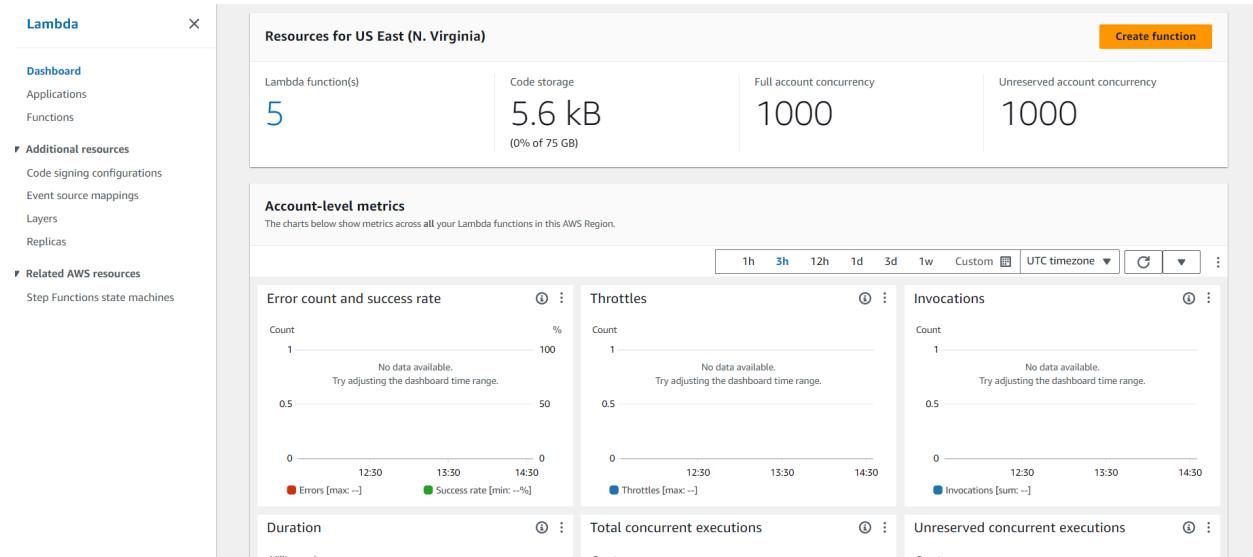
Prerequisites: AWS Personal/Academy Account

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.

The screenshot shows the 'Create function' wizard in the AWS Lambda console. The top navigation bar includes the AWS logo, 'Services' (with 'Lambda' selected), a search bar, and a 'Create function' button. The breadcrumb trail shows 'Lambda > Functions > Create function'. The main section is titled 'Create function' with an 'Info' link. It prompts the user to choose a creation option: 'Author from scratch' (selected), 'Use a blueprint', or 'Container image'. Below this is a 'Basic information' section where the 'Function name' is set to 'KCS_Lambda'. The 'Runtime' is chosen as 'Python 3.12'. The 'Architecture' is set to 'x86_64'. Under 'Permissions', it notes that Lambda will create an execution role with CloudWatch Logs permissions. A note states that role creation might take a few minutes. In the 'Advanced settings' section, there are options for memory, timeout, and layers. At the bottom are 'Cancel' and 'Create function' buttons.

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.
KCS_Lambda

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.
Python 3.12

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 [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".

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".

Code Test Monitor Configuration Aliases Versions

Code source Info Upload from ▾

File Edit Find View Go Tools Window Test Deploy Changes not deployed

Environment Var λ lambda_function Environment

```
lambda_function
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 Info Edit

Triggers	Description -	Memory 128 MB	Ephemeral storage 512 MB
Permissions	Timeout 0 min 3 sec	SnapStart Info None	
Destinations			
Function URL			
Environment variables			
Tags			
VPC			
RDS databases			
Monitoring and operations tools			

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.

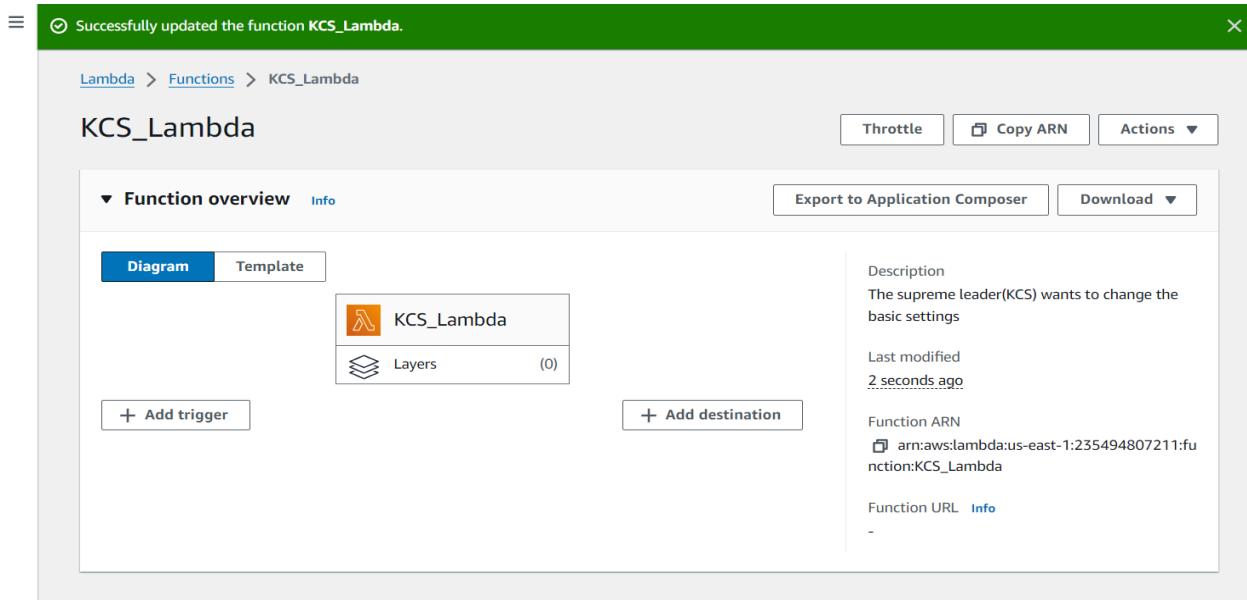
SnapStart [Info](#)
Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized. To evaluate whether your function code is resilient to snapshot operations, review the [SnapStart compatibility considerations](#).

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

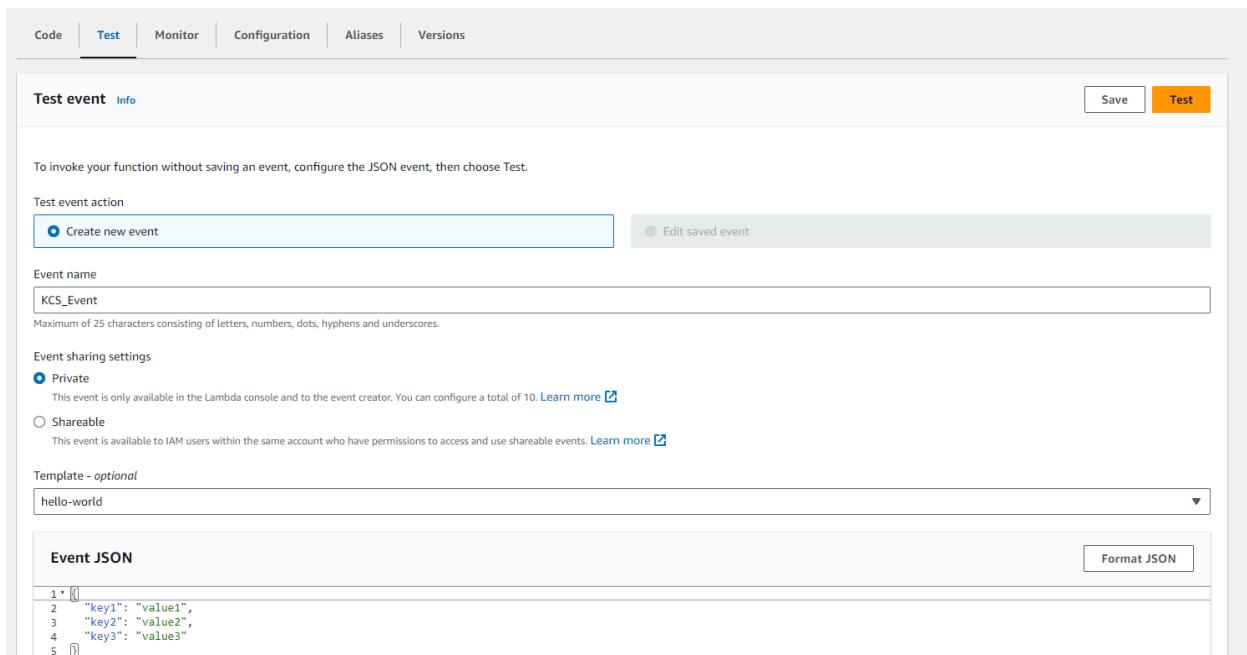
Timeout
 min sec

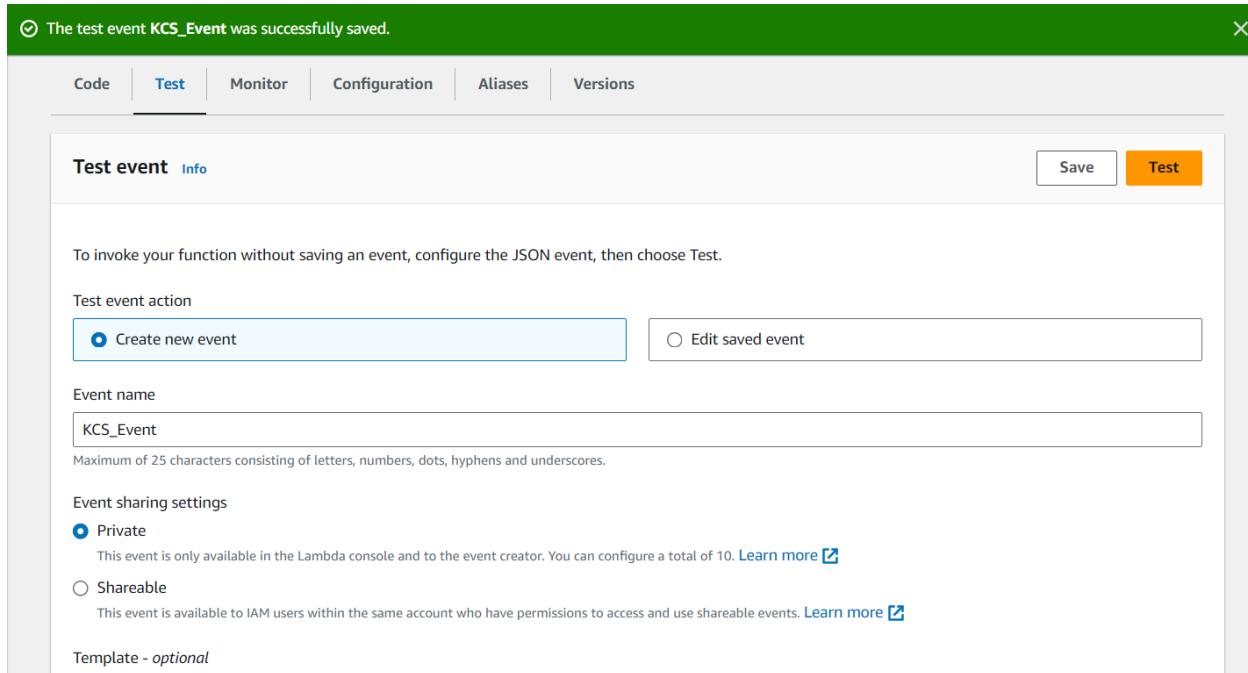
Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).
 Use an existing role
 Create a new role from AWS policy templates

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.

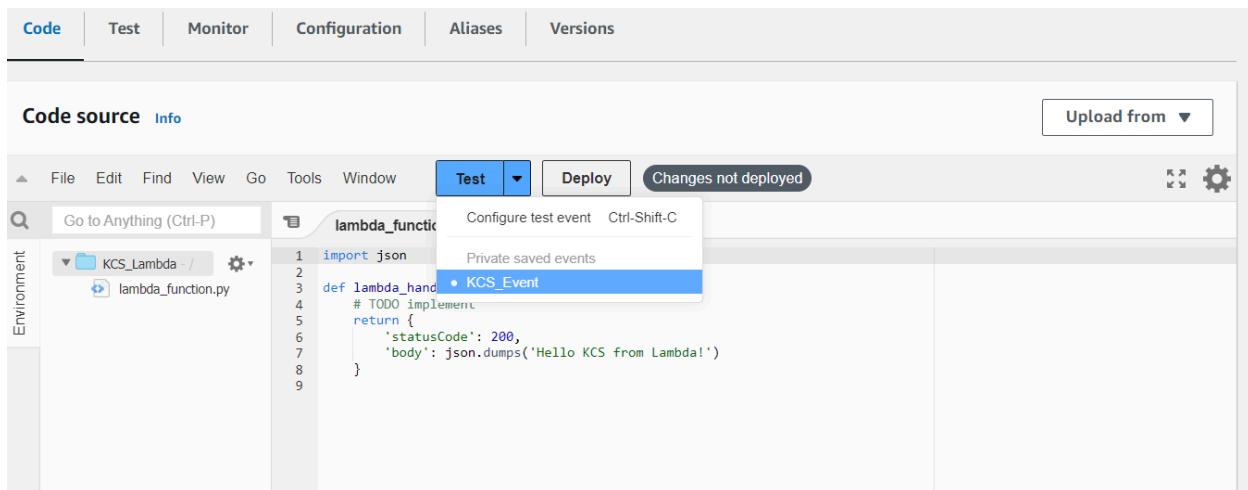


Step 3: Now Click on the Test tab then select Create a new event, give a name to the event and select Event Sharing to private, and select hello-world template.





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 two views of the AWS Lambda function editor for a function named 'lambda_function'.

Top View (Test Results):

- Code source:** Info
- Test:** Deploy
- Execution result:** Status: Succeeded | Max memory used: 32 MB | Time: 2.07 ms
- Test Event Name:** KCS_Event
- Response:**

```
{ "statusCode": 200, "body": "\"Hello KCS from Lambda!\""}  
Function Logs  
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 Mem
```
- Request ID:** 9b8874c5-da6e-4026-9098-134c4fee787f

Bottom View (Code Editor):

- Code:** Test | Monitor | Configuration | Aliases | Versions
- Code source:** Info
- Test:** Deploy | Changes not deployed
- Execution results:**
- lambda_function.py:**

```
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 Test console interface. At the top, there are tabs for 'Code source' and 'Info'. Below the tabs, a navigation bar includes File, Edit, Find, View, Go, Tools, Window, a 'Test' dropdown, Deploy, and settings icons. A search bar says 'Go to Anything (Ctrl-P)'. On the left, an 'Environment' sidebar lists 'KCS_Lambda - /' and 'lambda_function.py'. The main area has tabs for 'lambda_function.x', 'Environment Var x', and 'Execution result x'. Under 'Execution results', it shows a 'Test Event Name' of 'KCS_Event'. The 'Response' section displays the following JSON:

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

The 'Function Logs' section shows log entries:

```
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, it says "Storage" and "Amazon S3". Below that is a large heading "Amazon S3" with the subtext "Store and retrieve any amount of data from anywhere". A small note below says "Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance." To the right, there's a box titled "Create a bucket" containing text about buckets and a "Create bucket" button. Below the main content, there's a section titled "How it works" featuring a video thumbnail with the title "Introduction to Amazon S3" and a "Copy link" button.

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.

The screenshot shows the "Create bucket" configuration page. It starts with a header "Create bucket" and a note that "Buckets are containers for data stored in S3." Below this is a "General configuration" section. Under "AWS Region", "US East (N. Virginia) us-east-1" is selected. Under "Bucket type", "General purpose" is selected (indicated by a blue outline). There are two options: "General purpose" (selected) and "Directory". The "General purpose" option has a note: "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." The "Directory" option has a note: "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." Below this is a "Bucket name" field containing "wearekcs". A note says "Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)". Under "Copy settings from existing bucket - optional", it says "Only the bucket settings in the following configuration are copied." with a "Choose bucket" button. A note at the bottom says "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.

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

Successfully created bucket "wearekes"
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
Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

[View Storage Lens dashboard](#)

General purpose buckets | Directory buckets

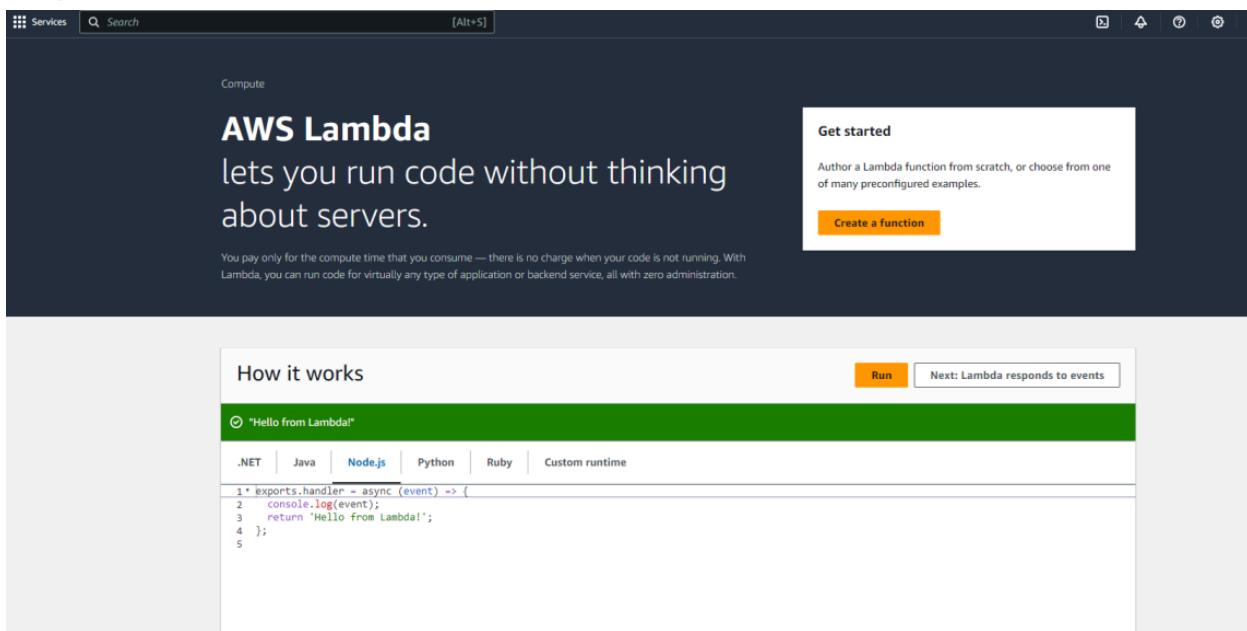
General purpose buckets (1) [Info](#) All AWS Regions
Buckets are containers for data stored in S3.

Find buckets by name

Name	AWS Region	IAM Access Analyzer	Creation date
wearekes	US East (N. Virginia) us-east-1	View analyzer for us-east-1	October 1, 2024, 13:40:40 (UTC+05:30)

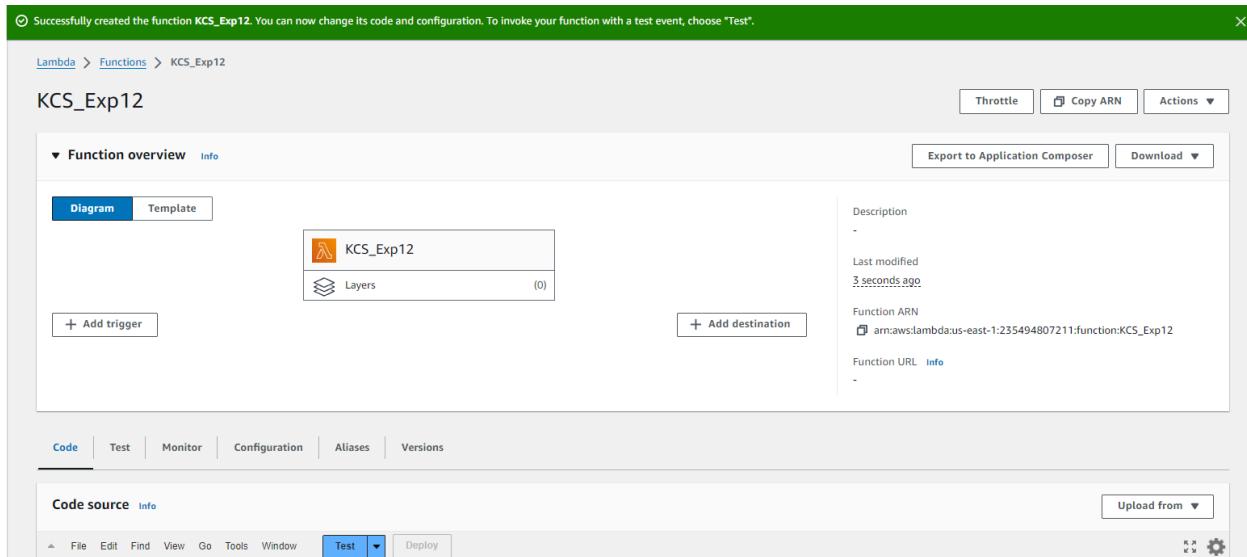
[Create bucket](#)

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

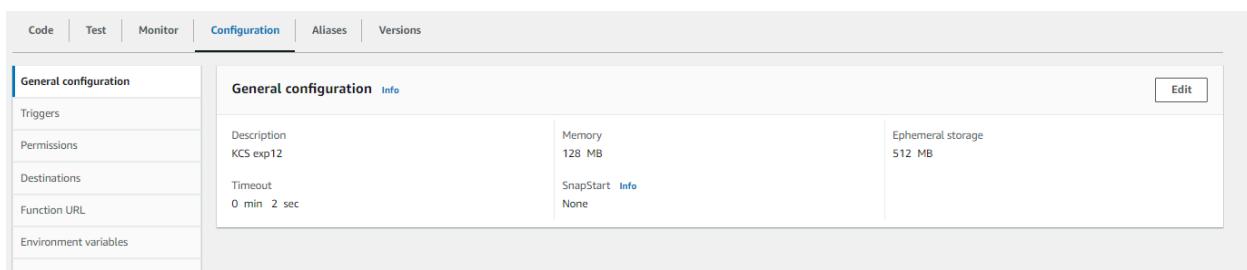


Step 4: Now Give a name to your Lambda function, Select the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby. So will select Python 3.12 , Architecture as x86, and Execution role to Create a new role with basic Lambda permissions.

The screenshot shows the 'Create function' wizard. In the first step, 'Author from scratch' is selected. The second step, 'Basic information', is shown with a 'Function name' field containing 'KCS_Exp12'. Under 'Runtime', 'Python 3.12' is chosen. In the 'Architecture' section, 'x86_64' is selected. The 'Permissions' section at the bottom indicates that a default execution role will be created. The 'Advanced settings' section is partially visible at the bottom.



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



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

The screenshot shows the 'Edit basic settings' page for a Lambda function named 'KCS_Exp12'. The 'Basic settings' tab is selected. Key configuration details include:

- Description - optional:** KCS exp12
- Memory:** 128 MB (Set memory to between 128 MB and 10240 MB)
- Ephemeral storage:** 512 MB (Set ephemeral storage (/tmp) to between 512 MB and 10240 MB)
- SnapStart:** 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:** service-role/KCS_Exp12-role-0q6h1t4r (View the KCS_Exp12-role-0q6h1t4r role on the IAM console.)

At the bottom right of the main content area, there is an orange 'Save' button.

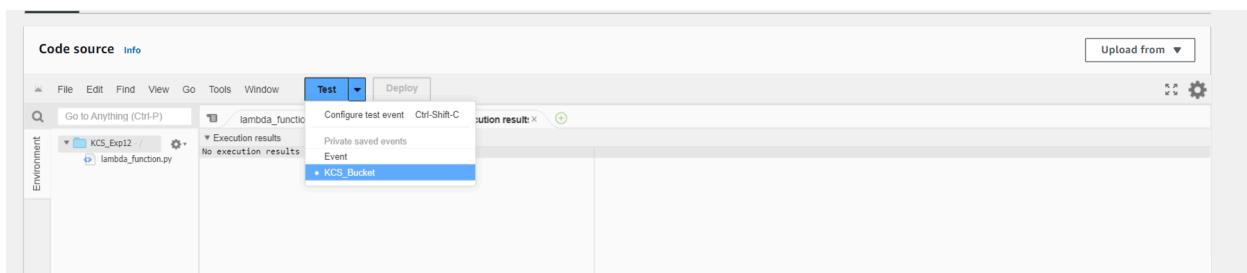
Step 5: Now Click on the Test tab then select Create a new event, give a name to the event and select Event Sharing to private, and select s3 put template.

The screenshot shows the 'Test' tab configuration page for the 'KCS_Exp12' function. The tabs at the top are 'Code', 'Test' (which is selected), 'Monitor', 'Configuration', 'Aliases', and 'Versions'. The 'Test' tab has the following configuration:

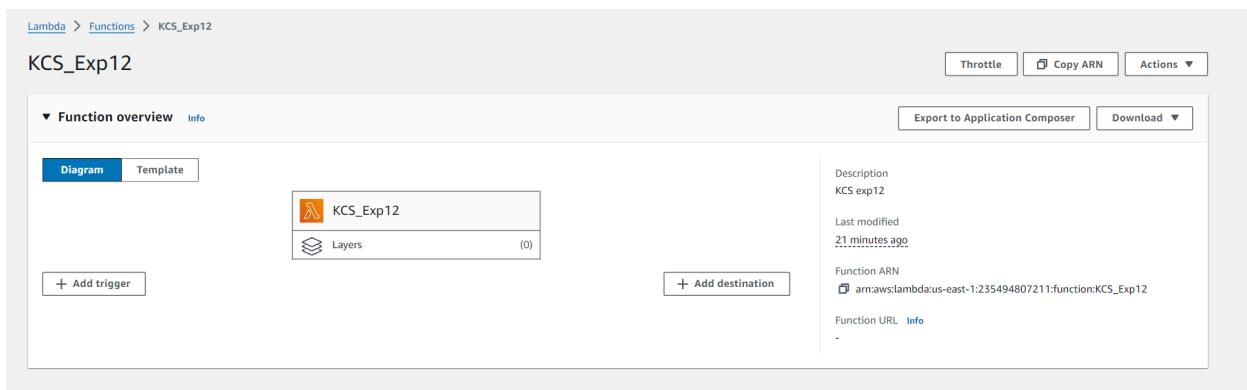
- Test event:** Info
- Test event action:** Create new event
- Event name:** KCS_Bucket
- Event sharing settings:**
 - Private
 - This event is only available in the Lambda console and to the event creator. You can configure a total of 10. [Learn more](#)
 - Shareable
 - This event is available to IAM users within the same account who have permissions to access and use shareable events. [Learn more](#)
- Template - optional:** s3-put
- Event JSON:** (Empty field)
- Buttons:** Save (grayed out), Test (orange)

```
Event JSON
[{"Records": [
    {
        "eventVersion": "2.0",
        "eventSource": "aws:s3",
        "awsRegion": "us-east-1",
        "eventTime": "1970-01-01T00:00:00.000Z",
        "eventName": "ObjectCreated:Put",
        "userIdentity": {
            "principalId": "EXAMPLE"
        },
        "requestParameters": {
            "sourceIPAddress": "127.0.0.1"
        },
        "responseElements": {
            "x-amz-request-id": "EXAMPLE123456789",
            "x-amz-id-2": "EXAMPLEE123/5678abcdefghijklmabaisawesome/mnopqrstuvwxyzABCDEFGH"
        },
        "s3": {
            "s3SchemaVersion": "1.0",
            "configurationId": "testConfigRule",
            "bucket": {
                "name": "example-bucket",
                "ownerIdentity": {
                    "principalId": "EXAMPLE"
                },
                "arn": "arn:aws:s3:::example-bucket"
            },
            "object": {
                "key": "test%2Fkey",
                "size": 1024,
                "eTag": "d41d8cd98f00b204e9800998ecf8427e",
                "sequencer": "00550B3D9D12345678"
            }
        }
    }
]}]
```

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.

Lambda > Add triggers

Add trigger

Trigger configuration [Info](#)

S3 aws asynchronous storage

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.

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.

Recursive invocation

KCS_Ex12

The trigger wearekcs was successfully added to function KCS_Ex12. The function is now receiving events from the trigger. [X](#)

Function overview [Info](#)

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

KCS_Ex12

Description KCS_Ex12

Last modified 26 minutes ago

Function ARN arn:aws:lambda:us-east-1:235494807211:function:KCS_Ex12

Function URL [Info](#)

S3 [+ Add destination](#)

[+ Add trigger](#)

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

The screenshot shows the AWS Lambda Configuration page. The left sidebar contains a list of configuration options: General configuration, Triggers (which is selected), 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 panel is titled 'Triggers (1) Info' and shows one trigger entry:

- Trigger:** S3: wearekcs
- ARN:** arnawsS3:wearekcs
- Details**

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 code in `lambda_function.py` is:

```

import json
def lambda_handler(event, context):
    # TODO implement
    bucket_name = event['Records'][0]['s3']['bucket']['name']
    object_key = event['Records'][0]['s3']['object']['key']
    print(f'An image has been added to the bucket {bucket_name} : {object_key}')
    return {
        'statusCode': 200,
        'body': json.dumps('Log entry created successfully')
    }

```

Below the editor, a green status bar says "Successfully updated the function KCS_Exp12." The bottom part of the screenshot shows the AWS Lambda Configuration page again, with the deployment successful message visible.

Step 9: Now upload any image to the bucket.

Amazon S3 > Buckets > wearekcs > Upload

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)		Remove	Add files	Add folder
All files and folders in this table will be uploaded.				
<input type="text"/> 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 status

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

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

Files and folders (1 Total, 957.0 KB)

Name	Folder	Type	Size	Status	Error
f_0JusXgjA...	-	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

Execution results: lambda_function Environment Var: Execution result: Status: Succeeded Max memory used: 32 MB Time: 1.88 ms

Test Event Name: KCS_Bucket

Response:

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

Function Logs:

```
START RequestId: ba624cc5-6862-4d62-84ca-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.

CloudWatch > Log groups > /aws/lambda/KCS_Exp12 > 2024/10/01/[\$LATEST]b93d5fc4cf3b4bf8802c5b106ff03bde

Log events

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

Filter events - press enter to search

Actions ▾ Start tailing Create metric filter

Timestamp	Message
No older events at this moment. Retry	
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 Mem...
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 Mem...
No newer events at this moment. Auto retry paused. Resume	