

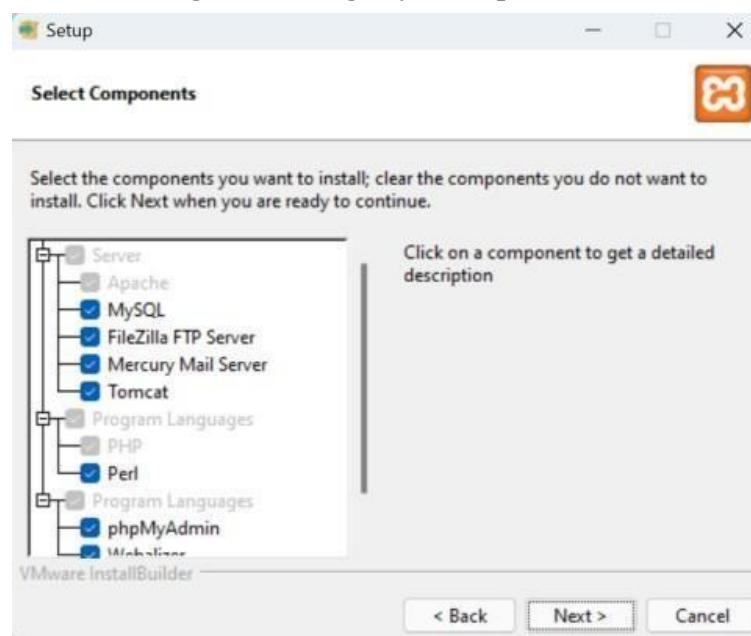
## EXPERIMENT 1A

### 1. Hosting Website using Xampp

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



Make the changes according to your requirements



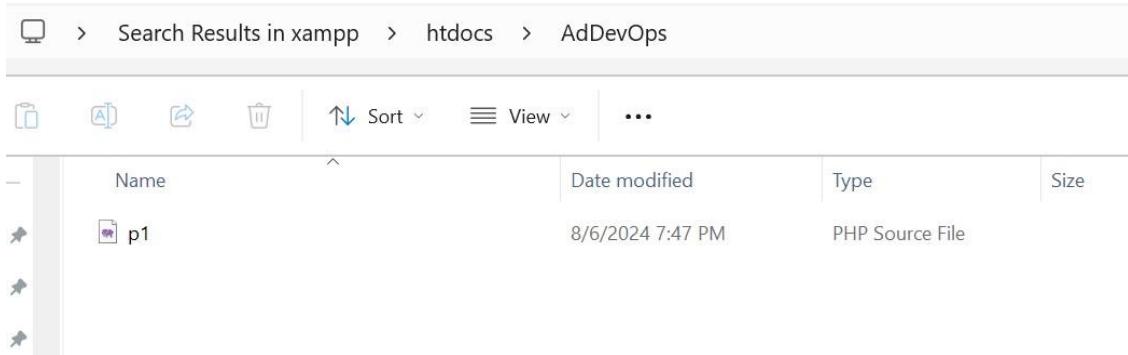


Open the control panel and click start for Apache



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

```
<html>
<title>Lab 1 Roshan</title>
<body>
    <?php
        echo""D15C";
    ?>
```



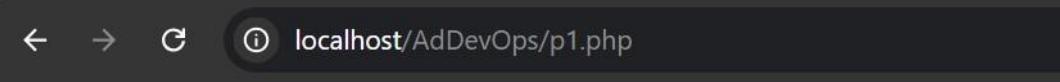
Go to your browser and type <localhost/filename/>



## Index of /AdDevOps

Name	Last modified	Size	Description
<a href="#">Parent Directory</a>	-	-	
<a href="#">p1.php</a>	2024-08-06 19:47	82	

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



D15C

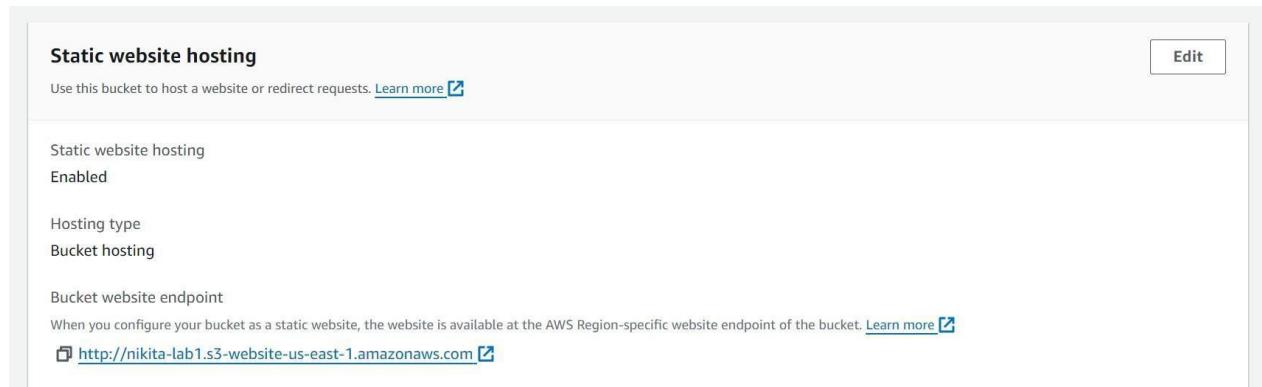
## Hosting website with Amazon S3

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

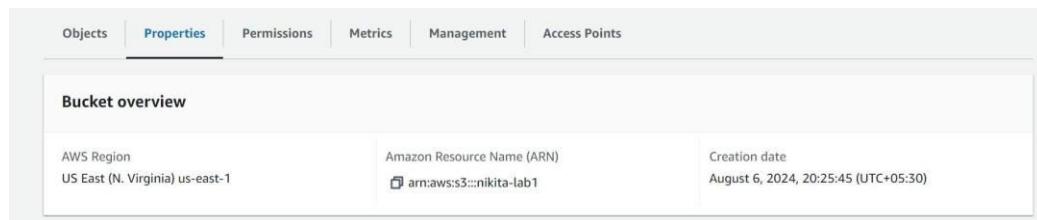


**Click on create bucket and make a bucket with any name**

**In properties go to static website hosting and enable it**



**Go to Uploads option and upload the code files in them**



The screenshot shows the 'Files and folders' section of an AWS S3 bucket. There are two items listed:

Name	Folder	Type	Size	Status	Error
<a href="#">error.html</a>	-	text/html	47.0 B	<span>✓ Succeeded</span>	-
<a href="#">index.html</a>	-	text/html	61.0 B	<span>✓ Succeeded</span>	-

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

The screenshot shows the 'Bucket policy' configuration page. It displays a JSON policy document:

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

Buttons for 'Edit' and 'Delete' are visible at the top right, and a 'Copy' button is located on the right side of the policy text area.

lab 1

The screenshot shows a web browser window with the following details:

- Address bar: Not secure | www.lab1a.com.s3-website-us-east-1.amazonaws.com
- Content area: Error Message



# Experiment 1 B

The screenshot shows the 'Create environment' wizard in the AWS Management Console. The current step is 'Details'. The 'Name' field is filled with 'WebAppIDE'. The 'Description - optional' field is empty. The 'Environment type' section shows two options: 'New EC2 instance' (selected) and 'Existing compute'. Under 'New EC2 instance', it lists three instance types: 't2.micro (1 GiB RAM + 1 vCPU)', 't3.small (2 GiB RAM + 2 vCPU)', and 'm5.large (8 GiB RAM + 2 vCPU)'. There is also a link to 'Additional instance types'.

For capabilities similar to AWS Cloud9, explore AWS Toolkits in your own IDE and AWS CloudShell in the AWS Management Console. [Learn more](#)

AWS Cloud9 > Environments > Create environment

## Create environment Info

**Details**

**Name**  
WebAppIDE  
Limit of 60 characters, alphanumeric, and unique per user.

**Description - optional**  
Limit 200 characters.

**Environment type** Info  
Determines what the Cloud9 IDE will run on.

**Environment type** Info  
Determines what the Cloud9 IDE will run on.

**New EC2 instance** Info  
Cloud9 creates an EC2 instance in your account. The configuration of your EC2 instance cannot be changed by Cloud9 after creation.

**Existing compute** Info  
You have an existing instance or server that you'd like to use.

**New EC2 instance**

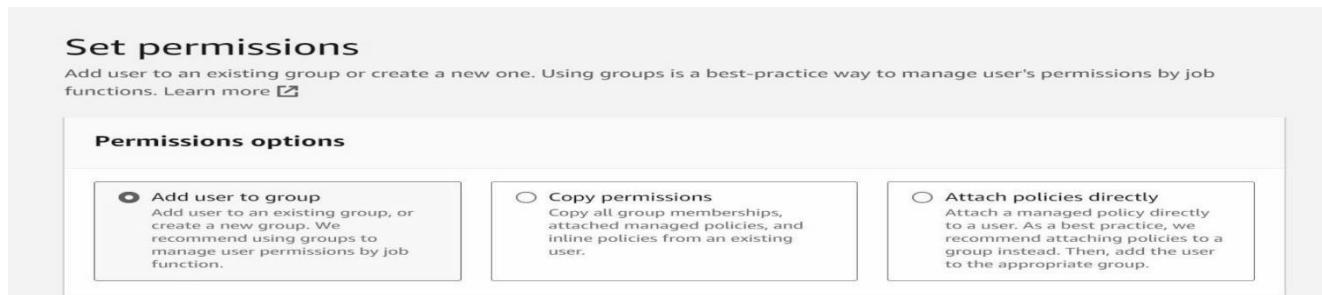
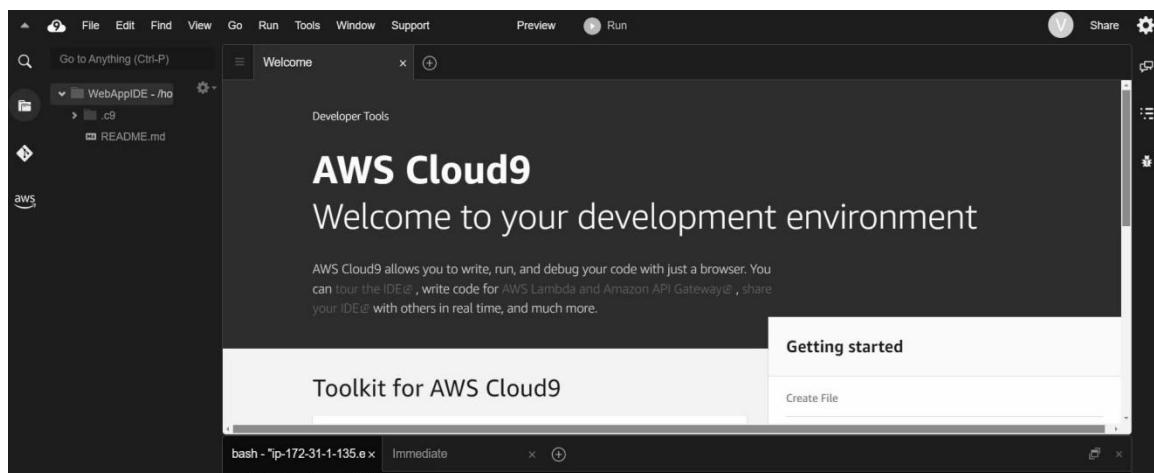
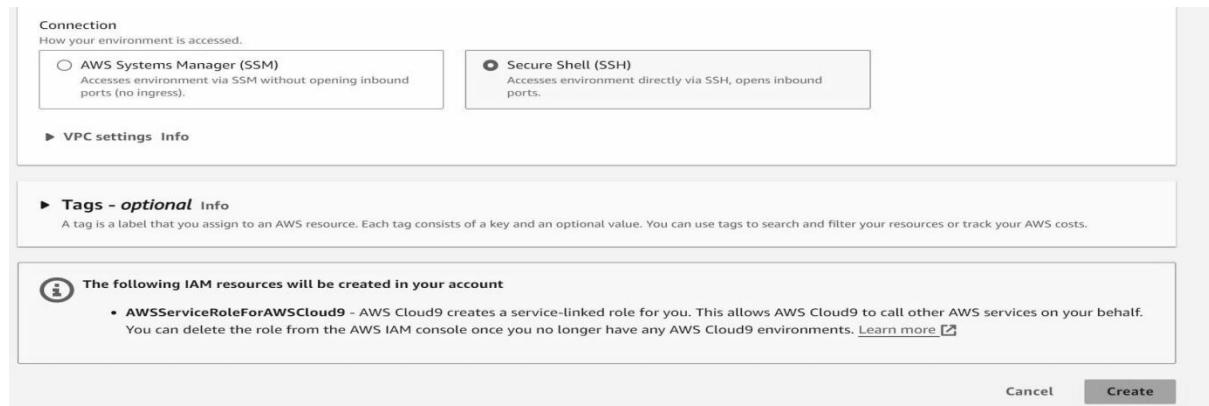
**Instance type** Info  
The memory and CPU of the EC2 instance that will be created for Cloud9 to run on.

**t2.micro (1 GiB RAM + 1 vCPU)**  
Free-tier eligible. Ideal for educational users and exploration.

**t3.small (2 GiB RAM + 2 vCPU)**  
Recommended for small web projects.

**m5.large (8 GiB RAM + 2 vCPU)**  
Recommended for production and most general-purpose development.

**Additional instance types**  
Explore additional instances to fit your need.



### Review and create

Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

User details		
User name user1	Console password type None	Require password reset No

### Identity and Access Management (IAM)

IAM > User groups > Create user group

#### Create user group

Name the group

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

Maximum 128 characters. Use alphanumeric and '+-=\_,@-' characters.

### Identity and Access Management (IAM)

IAM > User groups

group1 user group created.

Group name	Users	Permissions	Creation time
group1	0	Not defined	Now

The screenshot shows the final step of creating a new IAM user. It includes options for providing access to the AWS Management Console, selecting a user type (Identity Center or IAM), choosing a password (Autogenerated or Custom), and setting permissions by adding the user to a group. A summary of the selected options is shown on the left.

**Step 4: Set permissions**

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

**Are you providing console access to a person?**

User type

Specify a user in Identity Center - Recommended  
We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.

I want to create an IAM user  
We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keypairs, or a backup credential for emergency account access.

**Console password**

Autogenerated password  
You can view the password after you create the user.

Custom password  
Enter a custom password for the user.

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

**Permissions options**

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

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

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

**User groups (1/1)**

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

The screenshot shows the AWS Identity and Access Management (IAM) Policies page. A search bar at the top right contains the text "cloud9". Below the search bar, a table lists five AWS managed policies:

Policy name	Type	Used as	Description
AWSCloud9Admini...	AWS managed	None	Provides administrator access to AWS ...
AWSCloud9Enviro...	AWS managed	None	Provides the ability to be invited into A...
AWSCloud9Service...	AWS managed	None	Service Linked Role Policy for AWS Clo...
AWSCloud9SSMInsi...	AWS managed	None	This policy will be used to attach a rol...
AWSCloud9User	AWS managed	None	Provides permission to create AWS Clo...

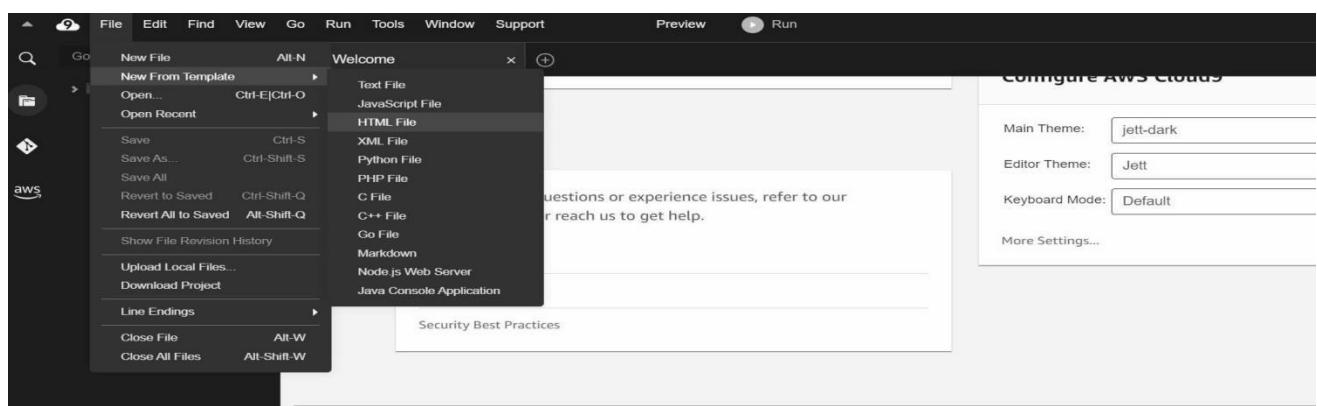
Below the table, a section titled "Cloud9User" provides a brief description of the policy's purpose. A "Copy JSON" button is located in the bottom right corner of this section.

Attach permissions policies - Optional (1/946) Info

You can attach up to 10 policies to this user group. All the users in this group will have permissions that are defined in the selected policies.

Policy name	Type	Used as	Description
AWSCloud9Admin...	AWS managed	None	Provides administrator access to A
AWSCloud9Enviro...	AWS managed	None	Provides the ability to be invited i
AWSCloud9SSMIns...	AWS managed	None	This policy will be used to attach a
<input checked="" type="checkbox"/> AWSCloud9User	AWS managed	None	Provides permission to create AWS

Filter by Type: All types | 4 matches | < 1 > |



Go to Anything (Ctrl-P)

WebAppIDE - /ho

Welcome Untitled1.html

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>AdDevOps Lab </title>
5   </head>
6   <body>
7     </body>
8   </html>
```

**Share this environment**

**Links to share**

Environment: <https://us-east-1.console.aws.amazon.com/cloud9/ide/dfa17a27c7754>

Application: 44.200.154.228

To make your application accessible from the internet, please follow [our documentation](#).

**Who has access**

ReadWrite  
● You (online) RW

Don't allow members to save their tab state

**Invite Members**

R RW Invite

Invite an existing IAM user or [create a new user](#).

# Experiment 2

**Configure environment** Info

**Environment tier** Info  
Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.

**Web server environment**  
Run a website, web application, or web API that serves HTTP requests. Learn more 

**Worker environment**  
Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. Learn more 

**Application information** Info

Application name  
  
Maximum length of 100 characters.

► Application tags (optional)

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

Service role  
 Create and use new service role  
 Use an existing service role

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

EC2 key pair  
Select an EC2 key pair to securely log in to your EC2 instances. Learn more   
  

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

**View permission details**

**Set up networking, database, and tags - optional** Info

**Virtual Private Cloud (VPC)**

VPC  
Launch your environment in a custom VPC instead of the default VPC. You can create a VPC and subnets in the VPC management console. Learn more   
 

Create custom VPC 

**Review Info**

**Step 1: Configure environment**

**Edit**

**Environment information**

Environment tier Web server environment	Application name: Application1
Environment name Application1-env	Application code Sample application
Platform arn:aws:elasticbeanstalk:us-east-1::platform/PHP 8.3 running on 64bit Amazon Linux 2023/4.3.1	

**Step 2: Configure service access**

**Edit**

**Service access Info**

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

Service role arn:aws:iam::011528263337:role/service-role/AWSCloud9SSMAccessRole	EC2 instance profile AWSCloud9SSMInstanceProfile
--	---

**Platform software**

Ignore health check false	Instance replacement false	
Lifecycle false	Log streaming Deactivated	Allow URL fopen On
Display errors Off	Document root —	Max execution time 60
Memory limit 256M	Zlib output compression Off	Proxy server nginx
Logs retention 7	Rotate logs Deactivated	Update level minor
X-Ray enabled Deactivated		

**Environment properties**

⌚ Environment successfully launched.

Elastic Beanstalk > Environments > Application1-env

### Application1-env Info

Environment overview

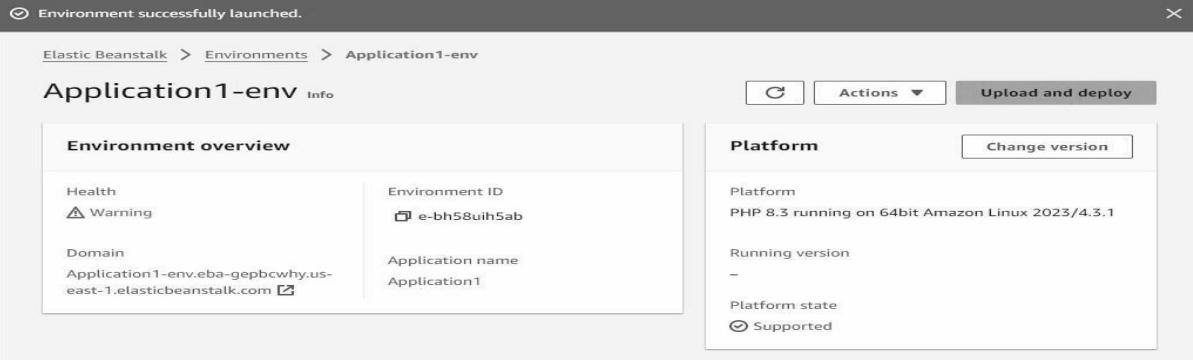
Health ⚠ Warning	Environment ID e-bh58uih5ab
Domain Application1-env.eba-gepbcbwhy.us-east-1.elasticbeanstalk.com	Application name Application1

Platform

Change version

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

Actions ▾ Upload and deploy



**Choose pipeline settings** Info

Step 1 of 5

**Pipeline settings**

**Pipeline name**  
Enter the pipeline name. You cannot edit the pipeline name after it is created.

pipeline1

No more than 100 characters

**Pipeline type**

ⓘ You can no longer create V1 pipelines through the console. We recommend you use the V2 pipeline type with improved release safety, pipeline triggers, parameterized pipelines, and a new billing model.

**Superseded**  
A more recent execution can overtake an older one. This is the default.

**Queued (Pipeline type V2 required)**

for these repositories:

**All repositories**

This applies to all current and future repositories owned by the resource owner.  
Also includes public repositories (read-only).

**Only select repositories**

Select at least one repository.  
Also includes public repositories (read-only).

with these permissions:

Read access to issues and metadata

Read and write access to administration, code, commit statuses, pull requests, and repository hooks

**Install**

**Cancel**

Next: you'll be directed to the GitHub App's site to complete setup.

## Deploy

Deploy provider  
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.

AWS Elastic Beanstalk ▾

Region  
US East (N. Virginia) ▾

Input artifacts  
Choose an input artifact for this action. Learn more

SourceArtifact ▾  
No more than 100 characters

Application name  
Choose an application that you have already created in the AWS Elastic Beanstalk console. Or create an application in the AWS Elastic Beanstalk console and then return to this task.

Application1 X

Environment name  
Choose an environment that you have already created in the AWS Elastic Beanstalk console. Or create an environment in the AWS Elastic Beanstalk console and then return to this task.

Application1-env X

Success  
Congratulations! The pipeline pipeline1 has been created. Create a notification rule for this pipeline

Developer Tools > CodePipeline > Pipelines > pipeline1

**pipeline1** Notify ▾ Edit Stop execution Clone pipeline Release change

Pipeline type: V2 Execution mode: QUEUED

<input checked="" type="checkbox"/> <b>Source</b> Succeeded Pipeline execution ID: <a href="#">db5e3336-41cc-486c-82b2-23dbf5ba2a13</a>	<input checked="" type="checkbox"/> <input type="radio"/>
<p>Source <a href="#">GitHub (Version 2)</a> </p> <p>Succeeded - <u>Just now</u> <a href="#">8be52cba</a> </p> <p><a href="#">View details</a></p>	

⌚ Deploy ⓘ In progress

Pipeline execution ID: [db5e3336-41cc-486c-82b2-23dbf5ba2a13](#)

Deploy

[AWS Elastic Beanstalk](#)

⌚ In progress - [Just now](#)

[View details](#)

[8be52cba](#) ⓘ Source: Adding template

### pipeline1

Pipeline type: V2 Execution mode: QUEUED

⌚ Source Succeeded

Pipeline execution ID: [db5e3336-41cc-486c-82b2-23dbf5ba2a13](#)

Source

[GitHub \(Version 2\)](#)

⌚ Succeeded - [1 minute ago](#)

[8be52cba](#)

[View details](#)

[8be52cba](#) ⓘ Source: Adding template

⌚ Deploy ⓘ Succeeded

Pipeline execution ID: [db5e3336-41cc-486c-82b2-23dbf5ba2a13](#)

[Start rollback](#)

Deploy

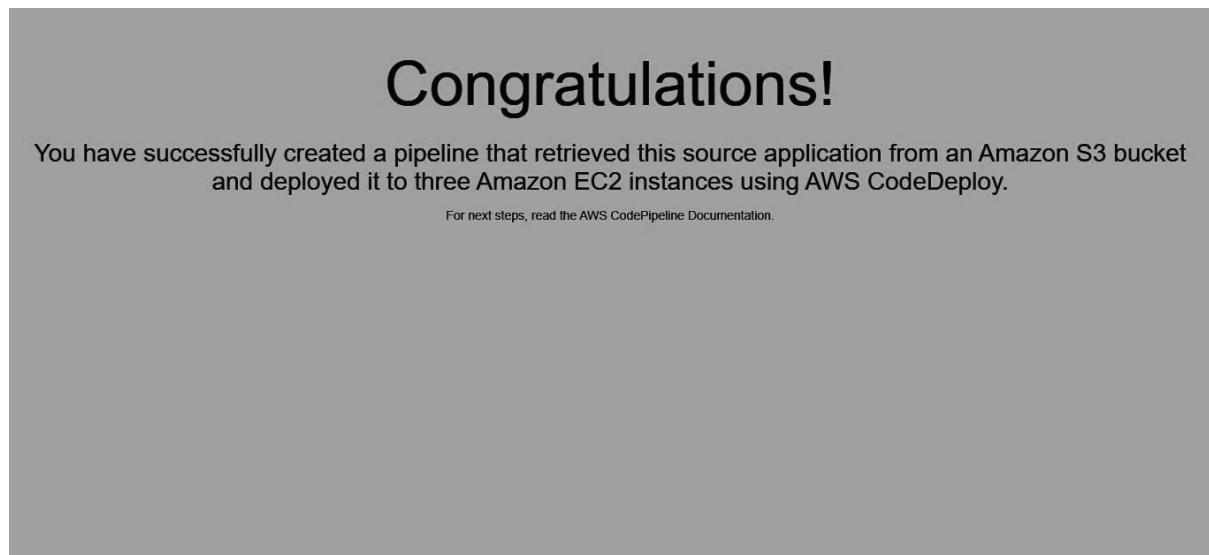
[AWS Elastic Beanstalk](#)

⌚ Succeeded - [Just now](#)

[View details](#)

[8be52cba](#) ⓘ Source: Adding template

Applications (1) <small>Info</small>		<small>C</small>	Actions ▾	Create application	
<small>Filter results matching the display value</small>		<small>&lt; 1 &gt; ⚙</small>			
Application name	▲   Environments	Date created	▼	Last modified	▼
● Application1	Application1-env	August 9, 2024 20:11:10 (...)		August 9, 2024 20:11:10 (...)	



# EXPERIMENT 3

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

Steps:

1. We will create 3 EC2 instances. One will be the master node and the other 2 will be slave/worker nodes.

Instances (3) <small>Info</small>		Last updated 43 minutes ago		Connect	Instance state ▾	Actions ▾	Launch instances ▾	▼
		Find Instance by attribute or tag (case-sensitive)	All states ▾				◀ 1 ▶	⚙️
<input type="checkbox"/>	Name ↗	Instance ID	Instance state	Instance type	Status check	Alarm status		
<input type="checkbox"/>	worker2	i-0f554e25913aa17a0	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a> +		
<input type="checkbox"/>	master	i-09878736747637d9a	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a> +		
<input type="checkbox"/>	worker1	i-063256e0e8d824e95	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a> +		

2. After the instances have been created, we will connect them one by one.

**Instances (1/3) Info** Last updated less than a minute ago

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
worker2	i-0f554e25913aa17a0	Running	t2.micro	2/2 checks passed	<a href="#">View alarms +</a>
<b>master</b>	i-09878736747637d9a	Running	t2.micro	2/2 checks passed	<a href="#">View alarms +</a>
worker1	i-063256e0e8d824e95	Running	t2.micro	2/2 checks passed	<a href="#">View alarms +</a>

Find Instance by attribute or tag (case-sensitive)

All states ▾

security group. For increased security, consider restricting access to only the EC2 Instance Connect service IP addresses for your Region: 13.239.158.0/29. [Learn more.](#)

Instance ID  
i-09878736747637d9a (master)

Connection Type

Connect using EC2 Instance Connect  
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.

Connect using EC2 Instance Connect Endpoint  
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address  
3.106.222.144

Username  
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ec2-user.

ec2-user

**Note:** In most cases, the default username, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Cancel **Connect**

### 3. Docker installation:

This step has to be performed on all the 3 instances. The following command has to be run:

```
yum install docker -y
```

```

Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-172-31-12-97 ~]$ sudo su
[root@ip-172-31-12-97 ec2-user]# yum install docker -y
Last metadata expiration check: 0:08:33 ago on Sat Sep 14 15:21:32 2024.
Dependencies resolved.

=====
| Package           | Architecture | Version      | Repository | Size |
|=====|
| Installing:      |             |              |            |       |
| docker            | x86_64       | 25.0.6-1.amzn2023.0.2 | amazonlinux | 44 M  |
| Installing dependencies: |             |              |            |       |
| containerd        | x86_64       | 1.7.20-1.amzn2023.0.1 | amazonlinux | 35 M  |
| iptables-libc     | x86_64       | 1.8.8-3.amzn2023.0.2 | amazonlinux | 401 k |
| iptables-nft      | x86_64       | 1.8.8-3.amzn2023.0.2 | amazonlinux | 183 k |
| libcgroup         | x86_64       | 3.0-1.amzn2023.0.1   | amazonlinux | 75 k  |
| libnetfilter_conntrack | x86_64       | 1.0.8-2.amzn2023.0.2 | amazonlinux | 58 k  |

=====
AWS Services Search [Alt+S] Sydney ▾ bhumish
=====
libn[...]k          x86_64    1.0.8-2.amzn2023.0.2      amazonlinux  58 k
libnfnetlink        x86_64    1.0.1-19.amzn2023.0.2    amazonlinux  30 k
libnftnl           x86_64    1.2.2-2.amzn2023.0.2    amazonlinux  84 k
pigz               x86_64    2.5-1.amzn2023.0.3      amazonlinux  83 k
runc               x86_64    1.1.13-1.amzn2023.0.1    amazonlinux  3.2 M

=====
Transaction Summary
=====
Install 10 Packages

Total download size: 84 M
Installed size: 317 M
Downloading Packages:
(1/10): iptables-libc-1.8.8-3.amzn2023.0.2.x86_64.rpm 3.0 MB/s | 401 kB 00:00
(2/10): iptables-nft-1.8.8-3.amzn2023.0.2.x86_64.rpm 6.6 MB/s | 183 kB 00:00
(3/10): libcgroup-3.0-1.amzn2023.0.1.x86_64.rpm 1.7 MB/s | 75 kB 00:00
(4/10): libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64.rpm 1.6 MB/s | 58 kB 00:00
(5/10): libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64.rpm 823 kB/s | 30 kB 00:00
(6/10): libnftnl-1.2.2-2.amzn2023.0.2.x86_64.rpm 2.9 MB/s | 84 kB 00:00
(7/10): pigz-2.5-1.amzn2023.0.3.x86_64.rpm 2.4 MB/s | 83 kB 00:00
(8/10): runc-1.1.13-1.amzn2023.0.1.x86_64.rpm 15 MB/s | 3.2 MB 00:00
(9/10): containerd-1.7.20-1.amzn2023.0.1.x86_64.rpm 36 MB/s | 35 kB 00:00
(10/10): docker-25.0.6-1.amzn2023.0.2.x86_64.rpm 30 MB/s | 44 kB 00:01

Total 56 MB/s | 84 MB 00:01

=====
Run      : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64 8/10
Installing : libcgroup-3.0-1.amzn2023.0.1.x86_64 9/10
Running scriptlet: docker-25.0.6-1.amzn2023.0.2.x86_64 10/10
Installing : docker-25.0.6-1.amzn2023.0.2.x86_64 10/10
Running scriptlet: docker-25.0.6-1.amzn2023.0.2.x86_64 10/10
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.

=====
Verifying : containerd-1.7.20-1.amzn2023.0.1.x86_64 1/10
Verifying : docker-25.0.6-1.amzn2023.0.2.x86_64 2/10
Verifying : iptables-libc-1.8.8-3.amzn2023.0.2.x86_64 3/10
Verifying : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64 4/10
Verifying : libcgroup-3.0-1.amzn2023.0.1.x86_64 5/10
Verifying : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64 6/10
Verifying : libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64 7/10
Verifying : libnftnl-1.2.2-2.amzn2023.0.2.x86_64 8/10
Verifying : pigz-2.5-1.amzn2023.0.3.x86_64 9/10
Verifying : runc-1.1.13-1.amzn2023.0.1.x86_64 10/10

=====
Installed:
containerd-1.7.20-1.amzn2023.0.1.x86_64  docker-25.0.6-1.amzn2023.0.2.x86_64  iptables-libc-1.8.8-3.amzn2023.0.2.x86_64
iptables-nft-1.8.8-3.amzn2023.0.2.x86_64  libcgroup-3.0-1.amzn2023.0.1.x86_64  libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64  libnftnl-1.2.2-2.amzn2023.0.2.x86_64  pigz-2.5-1.amzn2023.0.3.x86_64

=====
Complete!

```

4. After successfully docker has been installed it has to be started on all machines by using the command “`systemctl start docker`”

Complete!

`[root@ip-172-31-12-97 ec2-user]# systemctl start docker`

## 5. Kubernetes installation:

Search kubeadm installation on your browser and scroll down and select red hatbased distributions.

1. Set SELinux to `permissive` mode:

These instructions are for Kubernetes 1.31.

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

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

3. Install kubelet, kubeadm and kubectl:

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

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

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

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

```
[root@ip-172-31-12-97 ec2-user]# cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/repo/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
EOF
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/repo/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
[root@ip-172-31-12-97 ec2-user]# yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Kubernetes
Dependencies resolved.
=====
Package          Architecture Version      Repository  Size
=====
Installing:
kubeadm          x86_64      1.31.1-150500.1.1   kubernetes  11 M
kubectl          x86_64      1.31.1-150500.1.1   kubernetes  11 M

kube              x86_64      1.31.1-150500.1.1   kubernetes  15 M
Installing dependencies:
  conntrack-tools    x86_64      1.4.6-2.amzn2023.0.2   amazonlinux 208 k
  cri-tools          x86_64      1.31.1-150500.1.1   kubernetes  6.9 M
  kubernetes-cni     x86_64      1.5.1-150500.1.1   kubernetes  7.1 M
  libnetfilter_cthelper x86_64      1.0.0-21.amzn2023.0.2   amazonlinux 24 k
  libnetfilter_cttimeout x86_64      1.0.0-19.amzn2023.0.2   amazonlinux 24 k
  libnetfilter_queue   x86_64      1.0.5-2.amzn2023.0.2   amazonlinux 30 k

Transaction Summary
=====
Install 9 Packages

Total download size: 51 M
Installed size: 269 M
Downloading Packages:
(1/9): libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64.rpm           500 kB/s |  24 kB  00:00
(2/9): libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64.rpm           475 kB/s |  24 kB  00:00
(3/9): conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64.rpm                  3.6 MB/s | 208 kB  00:00
(4/9): libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64.rpm                1.4 MB/s | 30 kB  00:00
(5/9): kubeadm-1.31.1-150500.1.1.x86_64.rpm                          17 MB/s | 11 MB  00:00
(6/9): kubectl-1.31.1-150500.1.1.x86_64.rpm                         15 MB/s | 11 MB  00:00
(7/9): cri-tools-1.31.1-150500.1.1.x86_64.rpm                        8.0 MB/s | 6.9 MB  00:00
(8/9): kubernetes-cni-1.5.1-150500.1.1.x86_64.rpm                     14 MB/s | 7.1 MB  00:00
(9/9): kubelet-1.31.1-150500.1.1.x86_64.rpm                         25 MB/s | 15 MB  00:00
=====
Ins          ibnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64           5/9
Installing : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64                 6/9
Running scriptlet: conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64           6/9
Installing : kubelet-1.31.1-150500.1.1.x86_64                         7/9
Running scriptlet: kubelet-1.31.1-150500.1.1.x86_64                   7/9
Installing : kubeadm-1.31.1-150500.1.1.x86_64                         8/9
Running scriptlet: kubeadm-1.31.1-150500.1.1.x86_64                   9/9
Verifying   : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64               9/9
Verifying   : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64         2/9
Verifying   : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64        3/9
Verifying   : libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64            4/9
Verifying   : cri-tools-1.31.1-150500.1.1.x86_64                      5/9
Verifying   : kubeadm-1.31.1-150500.1.1.x86_64                      6/9
Verifying   : kubectl-1.31.1-150500.1.1.x86_64                      7/9
Verifying   : kubelet-1.31.1-150500.1.1.x86_64                      8/9
Verifying   : kubernetes-cni-1.5.1-150500.1.1.x86_64                  9/9
=====
Installed:
conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64                         cri-tools-1.31.1-150500.1.1.x86_64
kubeadm-1.31.1-150500.1.1.x86_64                           kubectl-1.31.1-150500.1.1.x86_64
kubelet-1.31.1-150500.1.1.x86_64                           kubernetes-cni-1.5.1-150500.1.1.x86_64
libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64           libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64
libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64
```

## 6. We can check if repository has been created by using yum repolist command.

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

7. Now we will be initializing the kubeadm. For that “kubeadm init” command has to be used. It may show errors but those can be ignored by using

**--ignore-preflight-errors=all**

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

```
aws Services Search [Alt+S] Sydney bhumishap
85 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-14-85.ap-southeast-2.compute.internal localhost] and IPs [172.31.14.85 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
[certs] Generating "sa" key and public key
[kubeconfig] Using kubeconfig folder "/etc/kubernetes"
[kubeconfig] Writing "admin.conf" kubeconfig file
[kubeconfig] Writing "super-admin.conf" kubeconfig file
[kubeconfig] Writing "kubelet.conf" kubeconfig file
[kubeconfig] Writing "controller-manager.conf" kubeconfig file
[kubeconfig] Writing "scheduler.conf" kubeconfig file
[etcd] Creating static Pod manifest for local etcd in "/etc/kubernetes/manifests"
[control-plane] Using manifest folder "/etc/kubernetes/manifests"
[control-plane] Creating static Pod manifest for "kube-apiserver"
[control-plane] Creating static Pod manifest for "kube-controller-manager"
[control-plane] Creating static Pod manifest for "kube-scheduler"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Starting the kubelet
[wait-control-plane] Waiting for the kubelet to boot up the control plane as static Pods from directory "/etc/kubernetes/manifests"
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 518.648244ms
[api-check] Waiting for a healthy API server. This can take up to 4m0s
[api-check] The API server is healthy after 10.001658622s
[upload-config] Storing the configuration used in ConfigMap "kubeadm-config" in the "kube-system" Namespace
[kubelet] Creating a ConfigMap "kubelet-config" in namespace kube-system with the configuration for the kubelets in the cluster
[upload-certs] Skipping phase. Please see --upload-certs
[mark-control-plane] Marking the node ip-172-31-14-85.ap-southeast-2.compute.internal as control-plane by adding the labels: [node-role.kubernetes.io/control-plane node.kubernetes.io/exclude-from-external-load-balancers]
[mark-control-plane] Marking the node ip-172-31-14-85.ap-southeast-2.compute.internal as control-plane by adding the taints [node-role.kubernetes.io/control-plane:NoSchedule]
[bootstrap-token] Using token: 6lysht.48enn4gmnhof6ex8
[bootstrap-token] Configuring bootstrap tokens, cluster-info ConfigMap, RBAC Roles
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to get nodes
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to post CSRs in order for nodes to get long term certificate credentials
[bootstrap-token] Configured RBAC rules to allow the csapprover controller automatically approve CSRs from a Node Bootstrap Token
[bootstrap-token] Configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
[bootstrap-token] Creating the "cluster-info" ConfigMap in the "kube-public" namespace
[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable kubelet client certificate and key
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy
```

Your Kubernetes control-plane has initialized successfully!

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

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

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

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

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

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

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

```
kubeadm join 172.31.14.85:6443 --token 61ysht.48enn4gmnhof6ex8 \
--discovery-token-ca-cert-hash sha256:461819c971fe032e04a78e18fde8e28755825e8468d468a2c86d88c52dba4945
```

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

This node has joined the cluster:

- \* Certificate signing request was sent to apiserver and a response was received.
- \* The Kubelet was informed of the new secure connection details.

Run '`kubectl get nodes`' on the control-plane to see this node join the cluster.

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

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

Conclusion:

This experiment successfully demonstrated the creation of a Kubernetes cluster and the successful addition of all three nodes using various commands. Errors encountered during initialization can be addressed in two ways: 1) by ignoring the errors, or 2) by

upgrading the instance type to t3.medium or t3.large if the issues are due to insufficient memory or CPU resources.

# Experiment 4

## Aim:

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

## Steps:

### 1. Create a key pair.

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

Name	<input type="text" value="rook"/>
The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.	
Key pair type	<input checked="" type="radio"/> RSA <input type="radio"/> ED25519
Private key file format	<input checked="" type="radio"/> .pem For use with OpenSSH
	<input type="radio"/> .ppk For use with PuTTY
Tags - optional	<input type="button" value="Add new tag"/> You can add up to 50 more tags.
<input type="button" value="Cancel"/> <input type="button" value="Create key pair"/>	

Key pairs (2) <small>Info</small>					
<input type="button" value="Actions ▾"/> <input type="button" value="Create key pair"/>					
<input type="checkbox"/>	Name	Type	Created	Fingerprint	ID
<input type="checkbox"/>	rook	rsa	2024/09/14 22:42 GMT+5:30	40:38:ad:9e:d0:9d:51:f4:f1:20:b0:...	key-04
<input type="checkbox"/>	ec2	rsa	2024/08/05 13:07 GMT+5:30	15:77:76:82:87:d2:40:e4:db:c7:2a...	key-0b

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

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

Instances (1) <a href="#">Info</a>		Last updated less than a minute ago	<a href="#">Connect</a>	Instance state	Actions	Launch instances
<input type="text"/> Find Instance by attribute or tag (case-sensitive)						All states
<input type="button" value="Instance state = running"/> <input type="button" value="Clear filters"/>						<a href="#">1</a>
	Name <a href="#">Edit</a>	Instance ID	Instance state	Instance type	Status check	Alarm status
<input type="checkbox"/>	instance	i-051e99d82072f03cd	<input checked="" type="radio"/> Running <input type="radio"/> <input type="radio"/>	t2.micro	<input checked="" type="checkbox"/> 2/2 checks passed <a href="#">View alarms</a> +	ap-so

3. Now edit the inbound rules to allow ssh.

Edit inbound rules [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-009a122cf85a62854	SSH	TCP	22	Cus... <input type="button" value="Info"/>	<input type="text"/> <input type="button" value="Delete"/>
-	All traffic	All	All	An... <input type="button" value="Info"/>	<input type="text"/> <input type="button" value="Delete"/>

[Add rule](#)

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

```
bhumi@LAPTOP-RVJC2CFS MINGW64 ~/Downloads
$ chmod 400 rook.pem

bhumi@LAPTOP-RVJC2CFS MINGW64 ~/Downloads
$
```

5. Now use this command on the terminal: ssh -i <keyname>.pem ubuntu@ and replace
- Keyname with the name of your key pair, in our case test1.
  - As we are using amazon Linux instead of ubuntu we will have ec2-user • Replace public ip address with its value. Go to your instance and scroll down and you will find the public ip address there.

```
bhumibhumi@LAPTOP-RVJC2CFS MINGW64 ~/Downloads
$ ssh -i "rook.pem" ec2-user@ec2-3-106-253-36.ap-southeast-2.compute.amazonaws.com
  _#_
  ~\_\_#####
  ~~~ \###\
  ~~ \###|      Amazon Linux 2023
  ~~ \#/ ,__-> https://aws.amazon.com/linux/amazon-linux-2023
  ~~~ / \
  ~~~ .- / \
  ~~~ / , /
  ~m/ , /
Last login: Sat Sep 14 17:41:50 2024 from 152.57.238.229
[ec2-user@ip-172-31-3-16 ~]$ |
```

## 6. Docker installation:

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

```
Last login: Mon Sep 18 11:00:49 UTC 2023 from 172.31.1.1
[ec2-user@ip-172-31-3-16 ~]$ sudo yum install docker -y
Last metadata expiration check: 0:05:38 ago on Sat Sep 14 17:38:25 2024.
Dependencies resolved.

    Package          Architecture Version       Repository      Size
Installing:
  docker           x86_64      25.0.6-1.amzn2023.0.2      amazonlinux   44 M
Installing dependencies:
  containerd        x86_64      1.7.20-1.amzn2023.0.1      amazonlinux   35 M
  iptables-libs     x86_64      1.8.8-3.amzn2023.0.2      amazonlinux   401 K
  iptables-nft      x86_64      1.8.8-3.amzn2023.0.2      amazonlinux   183 K
  libcgroup         x86_64      3.0-1.amzn2023.0.1      amazonlinux   75 K
  libnetfilter_conntrack x86_64  1.0.8-2.amzn2023.0.2      amazonlinux   58 K
  libnftnl          x86_64      1.0.1-19.amzn2023.0.2     amazonlinux   30 K
  libnftnl          x86_64      1.2.2-2.amzn2023.0.2     amazonlinux   84 K
  pigz              x86_64      2.5-1.amzn2023.0.3      amazonlinux   83 K
  runc              x86_64      1.1.13-1.amzn2023.0.1     amazonlinux   3.2 M

Transaction Summary
Install 10 Packages

Total download size: 84 M
Installed size: 317 M
Downloading Packages:
(1/10): iptables-libs-1.8.8-3.amzn2023.0.2.x86_64.rpm      4.1 MB/s | 401 kB  00:00
(2/10): iptables-nft-1.8.8-3.amzn2023.0.2.x86_64.rpm      6.8 MB/s | 183 kB  00:00
(3/10): libcgroup-3.0-1.amzn2023.0.1.x86_64.rpm          1.4 MB/s | 75 kB   00:00
(4/10): libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64.rpm 1.2 MB/s | 30 kB   00:00
(5/10): libnftnl-link-1.0.1-19.amzn2023.0.2.x86_64.rpm     3.1 MB/s | 58 kB   00:00
(6/10): libnftnl-link-1.2.2-2.amzn2023.0.2.x86_64.rpm      2.0 MB/s | 84 kB   00:00
(7/10): pigz-2.5-1.amzn2023.0.3.x86_64.rpm                1.4 MB/s | 83 kB   00:00
(8/10): runc-1.1.13-1.amzn2023.0.1.x86_64.rpm            15 MB/s | 3.2 MB  00:00
(9/10): containerd-1.7.20-1.amzn2023.0.1.x86_64.rpm      34 MB/s | 35 MB  00:01
(10/10): docker-25.0.6-1.amzn2023.0.2.x86_64.rpm        32 MB/s | 44 MB  00:01

Total                                         59 MB/s | 84 MB  00:01

Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing :                                                 1/10
  Installing : runc-1.1.13-1.amzn2023.0.1.x86_64          1/10
  Installing : containerd-1.7.20-1.amzn2023.0.1.x86_64      2/10
  Running scriptlet: containerd-1.7.20-1.amzn2023.0.1.x86_64 2/10
  Installing : pigz-2.5-1.amzn2023.0.3.x86_64             3/10
  Installing : libnftnl-link-1.2.2-2.amzn2023.0.2.x86_64    4/10
  Installing : libnftnl-link-1.0.1-19.amzn2023.0.2.x86_64    5/10
  Installing : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64 6/10

Total                                         59 MB/s | 84 MB  00:01

Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing :                                                 1/10
  Installing : runc-1.1.13-1.amzn2023.0.1.x86_64          1/10
  Installing : containerd-1.7.20-1.amzn2023.0.1.x86_64      2/10
  Running scriptlet: containerd-1.7.20-1.amzn2023.0.1.x86_64 2/10
  Installing : pigz-2.5-1.amzn2023.0.3.x86_64             3/10
  Installing : libnftnl-link-1.2.2-2.amzn2023.0.2.x86_64    4/10
  Installing : libnftnl-link-1.0.1-19.amzn2023.0.2.x86_64    5/10
  Installing : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64 6/10

Total                                         59 MB/s | 84 MB  00:01

Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing :                                                 1/10
  Installing : runc-1.1.13-1.amzn2023.0.1.x86_64          1/10
  Installing : containerd-1.7.20-1.amzn2023.0.1.x86_64      2/10
  Running scriptlet: containerd-1.7.20-1.amzn2023.0.1.x86_64 2/10
  Installing : pigz-2.5-1.amzn2023.0.3.x86_64             3/10
  Installing : libnftnl-link-1.2.2-2.amzn2023.0.2.x86_64    4/10
  Installing : libnftnl-link-1.0.1-19.amzn2023.0.2.x86_64    5/10
  Installing : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64 6/10
  Installing : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64    7/10
  Installing : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64    8/10
  Running scriptlet: iptables-nft-1.8.8-3.amzn2023.0.2.x86_64 8/10
  Installing : libcgroup-3.0-1.amzn2023.0.1.x86_64         9/10
  Running scriptlet: docker-25.0.6-1.amzn2023.0.2.x86_64   10/10
  Installing : docker-25.0.6-1.amzn2023.0.2.x86_64         10/10
  Running scriptlet: docker-25.0.6-1.amzn2023.0.2.x86_64   10/10

Created symlink /etc/systemd/system/sockets.target/wants/docker.socket → /usr/lib/systemd/system/docker.socket.

Verifying : containerd-1.7.20-1.amzn2023.0.1.x86_64          1/10
Verifying : docker-25.0.6-1.amzn2023.0.2.x86_64            2/10
Verifying : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64      3/10
Verifying : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64      4/10
Verifying : libcgroup-3.0-1.amzn2023.0.1.x86_64          5/10
Verifying : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64 6/10
Verifying : libnftnl-link-1.0.1-19.amzn2023.0.2.x86_64     7/10
Verifying : libnftnl-link-1.2.2-2.amzn2023.0.2.x86_64     8/10
Verifying : pigz-2.5-1.amzn2023.0.3.x86_64               9/10
Verifying : runc-1.1.13-1.amzn2023.0.1.x86_64            10/10

Installed:
  containerd-1.7.20-1.amzn2023.0.1.x86_64
  iptables-nft-1.8.8-3.amzn2023.0.2.x86_64
  libnftnl-link-1.0.1-19.amzn2023.0.2.x86_64
  runc-1.1.13-1.amzn2023.0.1.x86_64

docke-25.0.6-1.amzn2023.0.2.x86_64
libcgroup-3.0-1.amzn2023.0.1.x86_64
libnftnl-link-1.2.2-2.amzn2023.0.2.x86_64

iptables-libs-1.8.8-3.amzn2023.0.2.x86_64
libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
pigz-2.5-1.amzn2023.0.3.x86_64

Complete!
[ec2-user@ip-172-31-3-16 ~]
```

- Then to configure cgroup in a daemon json file we will run cd /etc/docker cat <<EOF | sudo tee /etc/docker/daemon.json{  
"exec-opts": ["native.cgroupdriver=systemd"]

```

}

EOF

sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker
[ec2-user@ip-172-31-3-16 ~]$ cd /etc/docker
cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"]
}
EOF
sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker
{
  "exec-opts": ["native.cgroupdriver=systemd"]
}
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
[ec2-user@ip-172-31-3-16 docker]$ |

```

## 8. Kubernetes installation:

Search kubeadm installation on your browser and scroll down and select red hatbased distributions.

1. Set SELinux to `permissive` mode:

These instructions are for Kubernetes 1.31.

```

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

```

```

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

```

3. Install kubelet, kubeadm and kubectl:

```

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

```

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

```

sudo systemctl enable --now kubelet

```

```
Error: This command has to be run with superuser privileges (under the root user on most systems).
[ec2-user@ip-172-31-3-16 docker]$ sudo yum install -y kubelet kubeadm kubectl --disablerelease=kubernetes
Kubernetes
Last metadata expiration check: 0:00:02 ago on Sat Sep 14 17:47:29 2024.
Dependencies resolved.
=====
Package           Architecture      Version            Repository        Size
=====
Installing:
kubernetes       x86_64          1.31.1-150500.1.1  kubernetes       11 M
kubectl          x86_64          1.31.1-150500.1.1  kubernetes       11 M
kubelet          x86_64          1.31.1-150500.1.1  kubernetes       15 M
Installing dependencies:
conntrack-tools   x86_64          1.4.6-2.amzn2023.0.2  amazonlinux     208 K
cri-tools         x86_64          1.31.1-150500.1.1  kubernetes       6.9 M
kubernetes-cni   x86_64          1.5.1-150500.1.1  kubernetes       7.1 M
libnetfilter_cthelper x86_64    1.0.0-21.amzn2023.0.2  amazonlinux     24 K
libnetfilter_cttimeout x86_64   1.0.0-19.amzn2023.0.2  amazonlinux     24 K
libnetfilter_queue x86_64    1.0.5-2.amzn2023.0.2  amazonlinux     30 K
=====
Transaction Summary
Install 9 Packages

Total download size: 51 M
Installed size: 269
Downloading Packages:
(1/9): libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64.rpm          499 kB/s |  24 kB  00:00
(2/9): libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64.rpm          376 kB/s |  24 kB  00:00
(3/9): libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64.rpm          1.6 MB/s | 30 kB  00:00
(4/9): conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64.rpm          1.7 MB/s | 208 kB  00:00
(5/9): cri-tools-1.31.1-150500.1.1.x86_64.rpm          15 MB/s | 6.9 MB  00:00
(6/9): kubeadm-1.31.1-150500.1.1.x86_64.rpm          21 MB/s | 11 MB  00:00
(7/9): kubelet-1.31.1-150500.1.1.x86_64.rpm          17 MB/s | 11 MB  00:00
(8/9): kubernetes-cni-1.5.1-150500.1.1.x86_64.rpm          21 MB/s | 7.1 MB  00:00
(9/9): kubelet-1.31.1-150500.1.1.x86_64.rpm          29 MB/s | 15 MB  00:00
=====
Total                                         45 MB/s | 51 MB  00:01
Kubernetes
Importing GPG key 0x9A296436:
  Userid : DE15:kubernetes OBS Project <4592:kubernetes@build.opensuse.org>
  Fingerprint: DE15 B144 86CD 377B 9E87 2346 54DA 9A29 6436
  URL: https://pkgs.io/core/stable/v1.31/rpm/repodata/repomd.xml.key
Key imported successfully
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing :
  Installing : kubernetes-cni-1.5.1-150500.1.1.x86_64
  Installing : cri-tools-1.31.1-150500.1.1.x86_64
  Installing : libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64
  1/1
  1/9
  2/9
  3/9
  4/9
  5/9
  6/9
  7/9
  8/9
  9/9
  -----
  Running transaction
    Preparing :
    Installing : kubernetes-cni-1.5.1-150500.1.1.x86_64
    Installing : cri-tools-1.31.1-150500.1.1.x86_64
    Installing : libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64
    1/1
    1/9
    2/9
    3/9
    4/9
    5/9
    6/9
    7/9
    8/9
    9/9
    -----
    Running scriptlet: conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64
    Installing : kubelet-1.31.1-150500.1.1.x86_64
    1/1
    1/9
    2/9
    3/9
    4/9
    5/9
    6/9
    7/9
    8/9
    9/9
    -----
    Running scriptlet: kubelet-1.31.1-150500.1.1.x86_64
    Installing : kubeadm-1.31.1-150500.1.1.x86_64
    1/1
    1/9
    2/9
    3/9
    4/9
    5/9
    6/9
    7/9
    8/9
    9/9
    -----
    Running scriptlet: kubelet-1.31.1-150500.1.1.x86_64
    Verifying  : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64
    Verifying  : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64
    Verifying  : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64
    Verifying  : libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64
    Verifying  : cri-tools-1.31.1-150500.1.1.x86_64
    Verifying  : kubeadm-1.31.1-150500.1.1.x86_64
    Verifying  : kubelet-1.31.1-150500.1.1.x86_64
    Verifying  : kubelet-1.31.1-150500.1.1.x86_64
    Verifying  : kubernetes-cni-1.5.1-150500.1.1.x86_64
    1/1
    2/9
    3/9
    4/9
    5/9
    6/9
    7/9
    8/9
    9/9
    -----
    Complete!
[ec2-user@ip-172-31-3-16 docker]$
```

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

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

```
[ec2-user@ip-172-31-3-16 docker]$ sudo swapoff -a  
[ec2-user@ip-172-31-3-16 docker]$ echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf  
net.bridge.bridge-nf-call-iptables=1  
[ec2-user@ip-172-31-3-16 docker]$ sudo sysctl -p  
net.bridge.bridge-nf-call-iptables = 1  
net.bridge.bridge-nf-call-iptables = 1  
[ec2-user@ip-172-31-3-16 docker]$
```

## 10. Initializing kubecluster:

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

Your Kubernetes control-plane has initialized successfully!

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

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

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

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

You should now deploy a pod network to the cluster.

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

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

```
kubeadm join 172.31.3.16:6443 --token ekhyop.xkge2agz07jxxqqs \
--discovery-token-ca-cert-hash sha256:8206263b4e2632eb03dafa4819c7c8505d47b21e8ba8c4901d5802c791c806f7
[ec2-user@ip-172-31-3-16 docker]$ |
```

11. The mkdir command that is generated after initialization has to be copied and pasted in the terminal.

```
[ec2-user@ip-172-31-3-16 docker]$ mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
[ec2-user@ip-172-31-3-16 docker]$
```

12. Then, add a common networking plugin called flannel: kubectl apply -f <https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml>

```
[ec2-user@ip-172-31-3-16 docker]$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
[ec2-user@ip-172-31-3-16 docker]$ |
```

13. Now that the cluster is up and running, we can deploy our nginx server on this cluster. Apply this deployment file using this command to create a deployment kubectl apply -f

```
[ec2-user@ip-172-31-3-16 docker]$ kubectl apply -f https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
[ec2-user@ip-172-31-3-16 docker]$ |
```

<https://k8s.io/examples/application/deployment.yaml>

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

```
[ec2-user@ip-172-31-3-16 docker]$ kubectl get pods
NAME                      READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-mvnj7   0/1     Pending   0          18s
nginx-deployment-d556bf558-w2pd8   0/1     Pending   0          18s
[ec2-user@ip-172-31-3-16 docker]$
```

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

```
nginx-deployment-d556bf558-w2pd8  0/1  Pending   0          18s
[ec2-user@ip-172-31-3-16 docker]$ kubectl describe pod nginx
Name:           nginx-deployment-d556bf558-mvnj7
Namespace:      default
Priority:       0
Service Account: default
Node:           <none>
Labels:          app=nginx
                 pod-template-hash=d556bf558
Annotations:    <none>
Status:         Pending
IP:
IPs:
Controlled By: ReplicaSet/nginx-deployment-d556bf558
Containers:
  nginx:
    Image:        nginx:1.14.2
    Port:         80/TCP
    Host Port:   0/TCP
    Environment: <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-8cms7 (ro)
Conditions:
  Type            Status
  PodScheduled    False
Volumes:
  kube-api-access-8cms7:
    Type:           Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:   kube-root-ca.crt
    ConfigMapOptional: <nil>
    DownwardAPI:    true
    QoS Class:      BestEffort
    Node-Selectors: <none>
    Tolerations:    node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                    node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type      Reason     Age   From           Message
  ----      ----     --   --            --
  Warning  FailedScheduling  57s  default-scheduler  0/1 nodes are available: 1 node(s) had untolerated taint {node-role.kubernetes.io/control-plane: }. preemption: 0/1 nodes are available: 1 Preemption is not helpful for scheduling.
Name:           nginx-deployment-d556bf558-w2pd8
Namespace:      default
Priority:       0
Service Account: default
Node:           <none>
Labels:          app=nginx
                 pod-template-hash=d556bf558
Annotations:    <none>
Status:         Pending
IP:
IPs:
Controlled By: ReplicaSet/nginx-deployment-d556bf558
Containers:
  nginx:
    Image:        nginx:1.14.2
    Port:         80/TCP
    Host Port:   0/TCP
    Environment: <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-6fl8b (ro)
Conditions:
  Type            Status
  PodScheduled    False
Volumes:
  kube-api-access-6fl8b:
    Type           Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:   kube-root-ca.crt
    ConfigMapOptional: <nil>
    DownwardAPI:    true
    QoS Class:      BestEffort
    Node-Selectors: <none>
    Tolerations:    node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                    node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type      Reason     Age   From           Message
  ----      ----     --   --            --
  Warning  FailedScheduling  57s  default-scheduler  0/1 nodes are available: 1 node(s) had untolerated taint {node-role.kubernetes.io/control-plane: }. preemption: 0/1 nodes are available: 1 Preemption is not helpful for scheduling.
```

Use the below command to remove taints.

```
[ec2-user@ip-172-31-3-16 docker]$ kubectl taint nodes --all node-role.kubernetes.io/control-plane-
node/ip-172-31-3-16.ap-southeast-2.compute.internal untainted
```

16. Check the pod status.

NAME	READY	STATUS	RESTARTS	AGE
nginx	1/1	Running	1 (6s ago)	90s

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

```
Forwarding from 127.0.0.1:8081 -> 80
Forwarding from [::1]:8081 -> 80
```

---

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

Then, use this curl command to check if the Nginx server is running. curl --head <http://127.0.0.1:8080>

Conclusion: In this experiment, we launched an EC2 instance and configured SSH access by updating the inbound rules. Next, we installed Docker and Kubernetes, and adjusted network settings to enable bridging. After completing the setup, we installed the Flannel networking plugin to ensure proper communication within the cluster. Once the cluster was up and running, we successfully deployed an NGINX server and verified its deployment.

# EXPERIMENT 5

## Step 1) Install Terraform

The screenshot shows the Terraform download page. At the top right, it says "1.9.4 (latest)". Below that, there's a "macOS" section with a "Package manager" option containing the command "brew tap hashicorp/tap" and "brew install hashicorp/tap/terraform". There's also a "Binary download" section for "macOS" with "AMD64" and "ARM64" options, each with a "Download" button. Below this is a "Windows" section with a "Binary download" section for "Windows" with "386" and "AMD64" options, each with a "Download" button.

## Step 2) Setup path in environment variables.

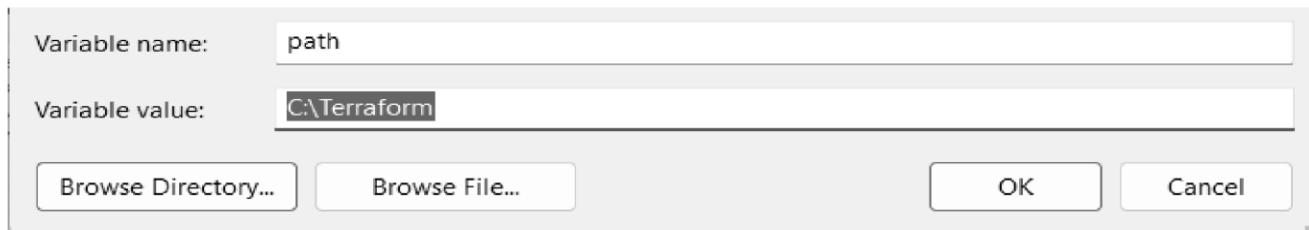
The screenshot shows the Windows Environment Variables dialog box. It has two main sections: "User variables for <username>" and "System variables".  
**User variables:**

Variable	Value
IntelliJ IDEA Community E...	C:\Users\91773\OneDrive\Desktop\java_neew\IntelliJ IDEA Co...
JAVA_HOME	C:\Users\91773\AppData\Local\Programs\Eclipse Adoptium\j...
OneDrive	C:\Users\91773\OneDrive
OneDriveConsumer	C:\Users\91773\OneDrive
Path	C:\Users\91773\AppData\Local\Programs\Eclipse Adoptium\j...
TEMP	C:\Users\91773\AppData\Local\Temp
TMP	C:\Users\91773\AppData\Local\Temp

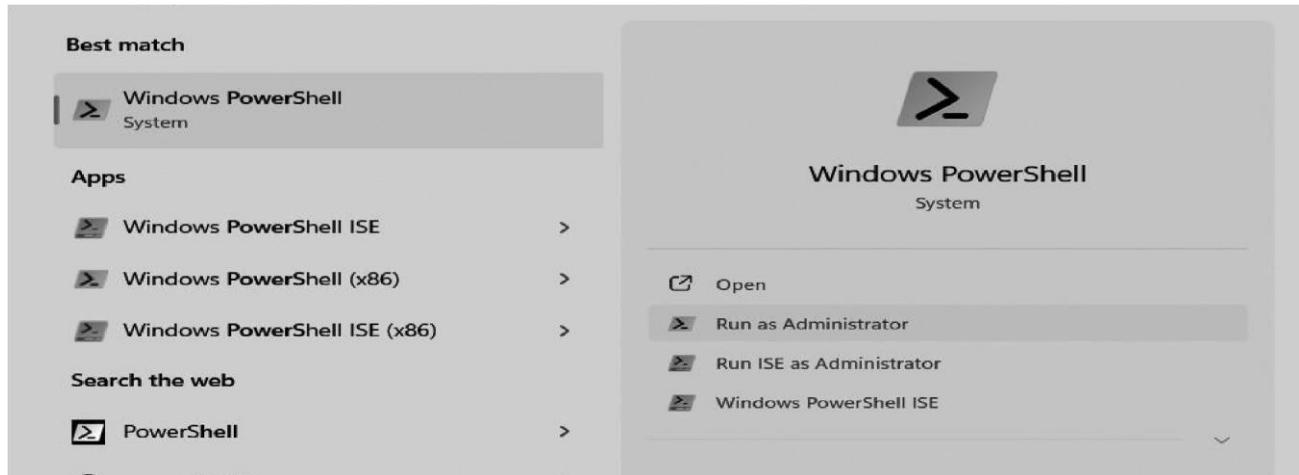
  
**System variables:**

Variable	Value
ACSetupSvcPort	23210
ACSvcPort	17532
ANDROID_HOME	D:\Flutter Dev\ANDROID_SDK
ComSpec	C:\WINDOWS\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
EnableLog	INFO
JAVA_HOME	D:\Flutter Dev\JDK

Step 3) Select the C drive Terraform folder as variable value.



Step 4) Open Windows powershell as Administrator.



Step 5) Run the Terraform command in powershell.

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

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

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

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

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

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

Global options (use these before the subcommand, if any):
  -chdir=DIR  Switch to a different working directory before executing the
             given subcommand.
  -help      Show this help output, or the help for a specified subcommand.
  -version   An alias for the "version" subcommand.

PS C:\WINDOWS\system32> -
```

# Experiment No:6

## Implementation:

### A. Creating docker image using terraform

#### Prerequisites:

1. Download and install Docker Desktop from Website: <https://www.docker.com>.

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

```
C:\Users\excel>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
  --help   Print this help message
```

#### Step 1:To Verify Docker Functionality

1. Create a folder named `Terraform Scripts` to store various scripts for this experiment.

#### Step 2:To Set Up Terraform Configuration

1. Inside the `Terraform Scripts` folder, create a new folder named `Docker`.
2. Within the `Docker` folder, create a file named `docker.tf` using Atom editor and insert the following content to configure an Ubuntu Linux container:

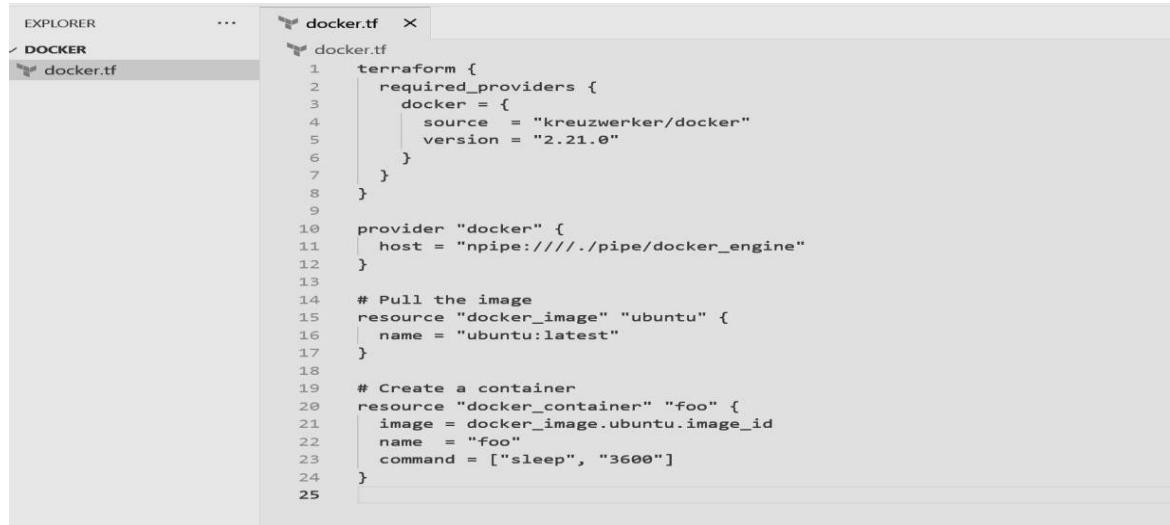
```
terraform {
```

```
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"]
}
```



The screenshot shows the Visual Studio Code interface with the 'EXPLORER' and 'DOCKER' panes on the left. The main pane displays the contents of the 'docker.tf' file. The code itself is as follows:

```
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:To Initialize Terraform

Run the command `terraform init` to initialize the Terraform configuration.

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

Terraform has been successfully initialized!

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

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

### Step 4:To Review Terraform Plan

Execute `terraform plan` to preview the resources that will be created.

```
C:\Users\excel\Documents\college\Terraform scripts\docker>terraform plan
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the
following symbols:
+ create

Terraform will perform the following actions:

# docker_container.foo will be created
+ resource "docker_container" "foo" {
    + attach                = false
    + bridge                 = (known after apply)
    + command               = [
        + "sleep",
        + "3600",
    ]
    + container_logs         = (known after apply)
    + entrypoint             = (known after apply)
    + env                    = (known after apply)
    + exit_code              = (known after apply)
    + gateway                = (known after apply)
    + hostname               = (known after apply)
    + id                     = (known after apply)
    + image                  = (known after apply)
    + init                   = (known after apply)
    + ip_address              = (known after apply)
    + ip_prefix_length       = (known after apply)
    + ipc_mode               = (known after apply)
    + log_driver              = (known after apply)
    + logs                   = false
    + must_run               = true
    + name                   = "foo"
    + network_data            = (known after apply)
    + read_only               = false
    + remove_volumes          = true
    + restart                = "no"
    + rm                      = false
    + runtime                = (known after apply)
    + security_opts           = (known after apply)
    + shm_size                = (known after apply)
    + start                  = true
}
```

```

    + 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: To Apply Terraform Configuration

Run `terraform apply` to apply the configuration and create the Ubuntu Linux container.

```
C:\Users\excel\Documents\college\Terraform scripts\docker>terraform apply
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# docker_container.foo will be created
+ resource "docker_container" "foo" {
    + attach           = false
    + bridge           = (known after apply)
    + command          = [
        + "sleep",
        + "3600",
    ]
    + container_logs   = (known after apply)
    + entrypoint       = (known after apply)
    + env              = (known after apply)
    + exit_code         = (known after apply)
    + gateway          = (known after apply)
    + hostname         = (known after apply)
    + id               = (known after apply)
    + image             = (known after apply)
    + init              = (known after apply)
    + ip_address        = (known after apply)
    + ip_prefix_length = (known after apply)
    + ipc_mode          = (known after apply)
    + log_driver        = (known after apply)
    + logs              = false
    + must_run          = true
    + name              = "foo"
    + network_data      = (known after apply)
    + read_only          = false
    + remove_volumes    = true
    + restart            = "no"
    + rm                = false
    + runtime            = (known after apply)
    + security_opts     = (known after apply)
    + shm_size           = (known after apply)
    + start              = true
}
```

```

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

+ healthcheck (known after apply)
+ labels (known after apply)
}

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

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

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

Enter a value: yes

docker_image.ubuntu: Creating...
docker_image.ubuntu: Still creating... [10s elapsed]
docker_image.ubuntu: Still creating... [20s elapsed]
docker_image.ubuntu: Still creating... [30s elapsed]
docker_image.ubuntu: Still creating... [40s elapsed]
docker_image.ubuntu: Creation complete after 43s [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Creating...
docker_container.foo: Creation complete after 1s [id=71bffb28b5cee3d1699c27dbcceb992b931000a847e6dfb219b3ca85ce5c6131]

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

```

Before executing `terraform apply`, list the Docker images.

```
C:\Users\excel\Documents\college\Terraform scripts\docker>docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
```

After executing `terraform apply`, list the Docker images again.

```
C:\Users\excel\Documents\college\Terraform scripts\docker>docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
ubuntu          latest    edbfe74c41f8    3 weeks ago  78.1MB
```

## Step 6: Clean Up

To delete the created Ubuntu container, run `terraform destroy`.

```
C:\Users\excel\Documents\college\Terraform scripts\docker>terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Refreshing state... [id=71bffb28b5cee3d1699c27dbcce992b931000a847e6dfb219b3ca85ce5c6131]

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

Terraform will perform the following actions:

# docker_container.foo will be destroyed
resource "docker_container" "foo" {
  attach           = false -> null
  command          = [
    - "sleep",
    - "3600",
  ] -> null
  cpu_shares       = 0 -> null
  dns              = [] -> null
  dns_opts         = [] -> null
  dns_search        = [] -> null
  entrypoint        = [] -> null
  env              = []
  gateway          = "172.17.0.1" -> null
  group_add        = [] -> null
  hostname          = "71bffb28b5cee3d1699c27dbcce992b931000a847e6dfb219b3ca85ce5c6131" -> null
  id               = "71bffb28b5cee3d1699c27dbcce992b931000a847e6dfb219b3ca85ce5c6131" -> null
  image             = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> null
  init              = false -> null
  ip_address        = "172.17.0.2" -> null
  ip_prefix_length  = 16 -> null
  ipc_mode          = "private" -> null
  links             = [] -> null
  log_driver         = "json-file" -> null
  log_opts           = {} -> null
  logot              = false -> null
  max_retry_count   = 0 -> null
  memory             = 0 -> null
  memory_swap        = 0 -> null
  must_run           = true -> null
  name              = "foo" -> null
}

- network_data      = [
  - {
    - gateway          = "172.17.0.1"
    - global_ipv6_prefix_length = 0
    - ip_address        = "172.17.0.2"
    - ip_prefix_length  = 16
    - network_name       = "bridge"
  } # (2 unchanged attributes hidden)
] -> null
network_mode        = "bridge" -> null
privileged          = false -> null
publish_all_ports   = false -> null
read_only            = false -> null
remove_volumes      = true -> null
restart              = "no" -> null
rm                  = false -> null
runtime              = "runc" -> null
security_opts       = [] -> null
shm_size             = 64 -> null
start                = true -> null
stdin_open           = false -> null
stop_timeout         = 0 -> null
storage_opts        = {} -> null
sysctls              = {} -> null
tmpfs                = {} -> null
tty                 = false -> null
} # (8 unchanged attributes hidden)

# docker_image.ubuntu will be destroyed
resource "docker_image" "ubuntu" {
  id               = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest" -> null
  image_id          = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> null
  latest             = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> null
  name              = "ubuntu:latest" -> null
  repo_digest        = "ubuntu@sha256:a837d68f4f73ebf3d4efafbcf66379bf3728902a8038616808f04e34a9ab63ee" -> null
}

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

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=71bffb28b5cee3d1699c27dbcce992b931000a847e6dfb219b3ca85ce5c6131]
docker_container.foo: Destruction complete after 0s
docker_image.ubuntu: Destroying... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_image.ubuntu: Destruction complete after 0s

Destroy complete! Resources: 2 destroyed.
```

After executing `terraform destroy` , list the Docker images one more time.

```
C:\Users\excel\Documents\college\Terraform scripts\docker>docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
```

Step 7: To check correctness of configured files.

Execute `terraform validate` to check the correctness of your Terraform configuration files.

```
C:\Users\excel\Documents\college\Terraform scripts\docker>terraform validate
Success! The configuration is valid.
```

Step 8: To verify the details.

Run `terraform providers` to list the providers used in your configuration and verify their details.

```
C:\Users\excel\Documents\college\Terraform scripts\docker>terraform providers
Providers required by configuration:
└── provider[registry.terraform.io/kreuzwerker/docker] 2.21.0
```

Step 9: To generate visual representation.

Generate a visual representation of the dependency graph of your Terraform resources.

```
C:\Users\excel\Documents\college\Terraform scripts\docker>terraform graph
digraph G {
    rankdir = "RL";
    node [shape = rect, fontname = "sans-serif"];
    "docker_container.foo" [label="docker_container.foo"];
    "docker_image.ubuntu" [label="docker_image.ubuntu"];
    "docker_container.foo" -> "docker_image.ubuntu";
}
```

## Experiment 7

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

### **Theory:**

#### **What is SAST?**

Static application security testing (SAST), or static analysis, is a testing methodology that analyzes source code to find security vulnerabilities that make your organization's applications susceptible to attack. SAST scans an application before the code is compiled. It's also known as white box testing.

#### **What problems does SAST solve?**

SAST takes place very early in the software development life cycle (SDLC) as it does not require a working application and can take place without code being executed. It helps developers identify vulnerabilities in the initial stages of development and quickly resolve issues without breaking builds or passing on vulnerabilities to the final release of the application.

SAST tools give developers real-time feedback as they code, helping them fix issues before they pass the code to the next phase of the SDLC. This prevents security-related issues from being considered an afterthought. SAST tools also provide graphical representations of the issues found, from source to sink. These help you navigate the code easier. Some tools point out the exact location of vulnerabilities and highlight the risky code. Tools can also provide in-depth guidance on how to fix issues and the best place in the code to fix them, without requiring deep security domain expertise.

It's important to note that SAST tools must be run on the application on a regular basis, such as during daily/monthly builds, every time code is checked in, or during a code release.

#### **Why is SAST important?**

Developers dramatically outnumber security staff. It can be challenging for an organization to find the resources to perform code reviews on even a fraction of its applications. A key strength of SAST tools is the ability to analyze 100% of the codebase. Additionally, they are much faster than manual secure code reviews performed by humans. These tools can scan millions of lines of code in a matter of minutes. SAST tools automatically identify critical vulnerabilities—such as buffer overflows, SQL injection, cross-site scripting, and others—with high confidence. Thus, integrating

static analysis into the SDLC can yield dramatic results in the overall quality of the code developed.

## **What are the key steps to run SAST effectively?**

There are six simple steps needed to perform SAST efficiently in organizations that have a very large number of applications built with different languages, frameworks, and platforms.

1. **Finalize the tool.** Select a static analysis tool that can perform code reviews of applications written in the programming languages you use. The tool should also be able to comprehend the underlying framework used by your software.
2. **Create the scanning infrastructure, and deploy the tool.** This step involves handling the licensing requirements, setting up access control and authorization, and procuring the resources required (e.g., servers and databases) to deploy the tool.
3. **Customize the tool.** Fine-tune the tool to suit the needs of the organization. For example, you might configure it to reduce false positives or find additional security vulnerabilities by writing new rules or updating existing ones. Integrate the tool into the build environment, create dashboards for tracking scan results, and build custom reports.
4. **Prioritize and onboard applications.** Once the tool is ready, onboard your applications. If you have a large number of applications, prioritize the high-risk applications to scan first. Eventually, all your applications should be onboarded and scanned regularly, with application scans synced with release cycles, daily or monthly builds, or code check-ins.
5. **Analyze scan results.** This step involves triaging the results of the scan to remove false positives. Once the set of issues is finalized, they should be tracked and provided to the deployment teams for proper and timely remediation.
6. **Provide governance and training.** Proper governance ensures that your development teams are employing the scanning tools properly. The software security touchpoints should be present within the SDLC. SAST should be incorporated as part of your application development and deployment process.

## **Integrating Jenkins with SonarQube:**

### **Prerequisites:**

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

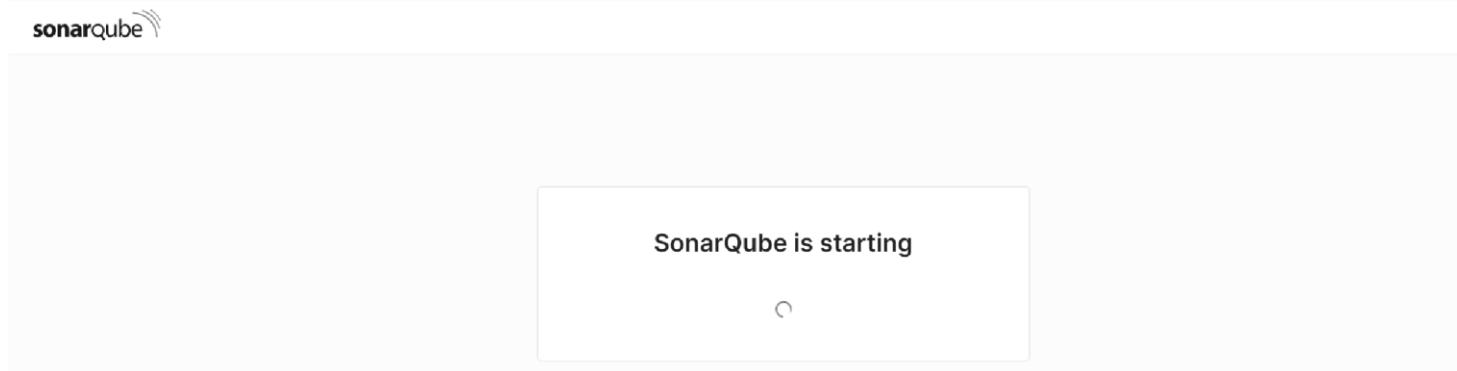
## Steps to integrate Jenkins with SonarQube

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

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

```
C:\Users\ADMIN>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9fec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
de76efbeef2054aeb442b86ba54c2916039b8757b388482d9780ffc69f5d8bbe
```

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



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

5. Create a manual project in SonarQube with the name **sonarqube**

1 of 2

### Create a local project

Project display name \*

sonarqube



Project key \*

sonarqube



Main branch name \*

main

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

[Cancel](#)

[Next](#)

Setup the project and come back to Jenkins Dashboard.

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

A screenshot of the Jenkins Marketplace search results. A search bar at the top contains the text 'sonar'. To the right of the search bar are an 'Install' button and a dropdown menu. Below the search bar, a table lists a single plugin: 'SonarQube Scanner 2.17.2'. The table has columns for 'Install', 'Name ↓', and 'Released'. The 'Install' column has a checkbox. The 'Name ↓' column shows 'SonarQube Scanner 2.17.2'. The 'Released' column shows '7 mo 9 days ago'. Below the table, a brief description of the plugin is visible.

7. Under Jenkins ‘Configure System’, look for SonarQube Servers and enter the details.

Enter the Server Authentication token if needed.

A screenshot of the Jenkins 'System' configuration page under 'Manage Jenkins'. The URL is 'Dashboard > Manage Jenkins > System >'. The page displays 'SonarQube installations' with a table. The table has columns for 'Name', 'Server URL', and 'Server authentication token'. One entry is shown: 'sonarqube' with 'http://localhost:9000' in the URL field and '- none -' in the token field. There is also an '+ Add' button. An 'Advanced' dropdown is visible at the bottom.

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

A screenshot of the Jenkins 'Global Tool Configuration' page under 'Manage Jenkins'. The URL is 'Dashboard > Manage Jenkins > Global Tool Configuration > SonarQube Scanner installations'. The page shows a table with one entry: 'sonarqube'. Below the table, there is a section for 'Install automatically'. A checkbox is checked next to 'Install automatically'. A 'Install from Maven Central' section is expanded, showing a 'Version' dropdown set to 'SonarQube Scanner 6.1.0.4477'. There is also an 'Add Installer' button.

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

## New Item

Enter an item name

SonarQube

Select an item type



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.

10. Choose this GitHub repository in Source Code Management.

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

Source Code Management

None

Git [?](#)

Repositories [?](#)

Repository URL [?](#)

Credentials [?](#)

- none -

[+ Add](#) [▼](#)

[Advanced](#) [▼](#)

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

11. Under Build-> 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

**Build Steps****Execute SonarQube Scanner**SonarQube Installation: [?](#)

sonarqube

JDK [?](#)

JDK to be used for this SonarQube analysis

(Inherit From Job)

Path to project properties [?](#)Analysis properties [?](#)

```
sonar.projectKey=sonarqube
sonar.login=admin
sonar.password=admin123
sonar.sources=C:\\ProgramData\\Jenkins\\jenkins\\workspace\\SonarQube
sonar.host.url=http://127.0.0.1:9000
```

Additional arguments [?](#)JVM Options [?](#)

-Dsonar.ws.timeout=300

**Save****Apply**

12. Go to [http://localhost:9000/<user\\_name>/permissions](http://localhost:9000/<user_name>/permissions) and allow Execute Permissions to the Admin user.

	<a href="#">Administer System</a> <a href="#">?</a>	<a href="#">Administer</a> <a href="#">?</a>	<a href="#">Execute Analysis</a> <a href="#">?</a>	<a href="#">Create</a> <a href="#">?</a>
A <b>Administrator</b> admin	<input type="checkbox"/>	<input type="checkbox"/> Quality Gates <input type="checkbox"/> Quality Profiles <input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Projects

13. Run The Build.

- Status**
- Changes
- Workspace
- Build Now**
- Configure
- Delete Project
- SonarQube
- Rename

Check the console output.

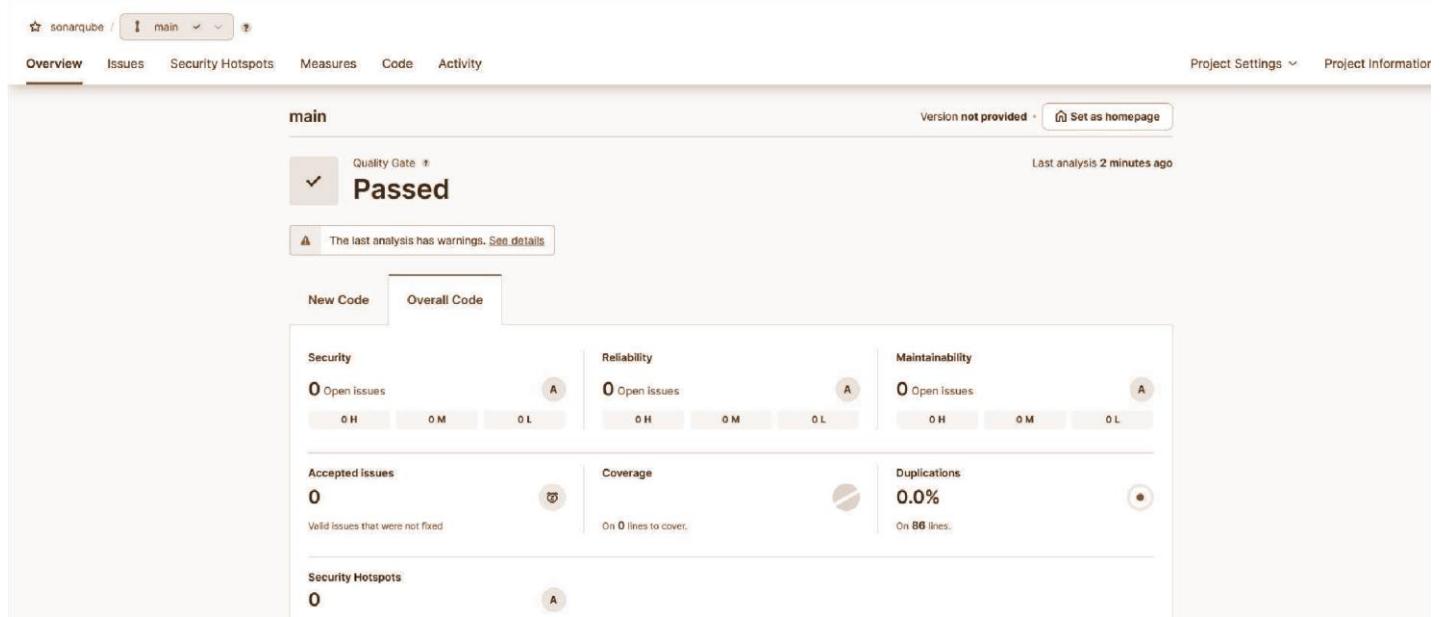
## Console Output

 Download  Copy View as plain text

```
Started by user Shiven Bansal
Running as SYSTEM
Building on the built-in node in workspace C:\ProgramData\Jenkins\.jenkins\workspace\SonarQube
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\SonarQube\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/shazforiot/MSBuild_firstproject.git # timeout=10
Fetching upstream changes from https://github.com/shazforiot/MSBuild_firstproject.git
> git.exe --version # timeout=10
> git --version # 'git version 2.45.2.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/shazforiot/MSBuild_firstproject.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision f2bc042c04c6e72427c380bcaee6d6fee7b49adf (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f f2bc042c04c6e72427c380bcaee6d6fee7b49adf # timeout=10
Commit message: "updated"
> git.exe rev-list --no-walk f2bc042c04c6e72427c380bcaee6d6fee7b49adf # timeout=10
[SonarQube] $ C:\ProgramData\Jenkins\.jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube\bin\sonar-scanner.bat -Dsonar.host.url=http://localhost:9000 -Dsonar.projectKey=sonarqube -Dsonar.login=admin -Dsonar.host.url=http://127.0.0.1:9000 -Dsonar.sources=C:\ProgramData\Jenkins\.jenkins\workspace\SonarQube -Dsonar.password=admin123 -Dsonar.projectBaseDir=C:\ProgramData\Jenkins\.jenkins\workspace\SonarQube
16:16:39.198 WARN Property 'sonar.host.url' with value 'http://localhost:9000' is overridden with value 'http://127.0.0.1:9000'
16:16:39.206 INFO Scanner configuration file: C:\ProgramData\Jenkins\.jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube\bin..\conf\sonar-scanner.properties
16:16:39.206 INFO Project root configuration file: NONE
16:16:39.230 INFO SonarScanner CLI 6.1.0.4477
16:16:39.230 INFO Java 21.0.4 Eclipse Adoptium (64-bit)
16:16:39.230 INFO Windows 11 10.0 amd64
16:16:39.230 INFO SONAR_SCANNER_OPTS=-Dsonar.ws.timeout=300
16:16:39.254 INFO User cache: C:\Windows\system32\config\systemprofile\.sonar\cache

16:16:58.734 INFO Using git CLI to retrieve untracked files
16:16:58.791 INFO Analyzing language associated files and files included via "sonar.text.inclusions" that are tracked by git
16:16:58.856 INFO 14 source files to be analyzed
16:16:59.154 INFO 14/14 source files have been analyzed
16:16:59.154 INFO Sensor TextAndSecretsSensor [text] (done) | time=1306ms
16:16:59.163 INFO ----- Run sensors on project
16:16:59.373 INFO Sensor C# [csharp]
16:16:59.373 WARN Your project contains C# files which cannot be analyzed with the scanner you are using. To analyze C# or VB.NET, you must use the SonarScanner for .NET 5.x or higher, see https://redirect.sonarsource.com/doc/install-configure-scanner-msbuild.html
16:16:59.373 INFO Sensor C# [csharp] (done) | time=0ms
16:16:59.373 INFO Sensor Analysis Warnings import [csharp]
16:16:59.379 INFO Sensor Analysis Warnings import [csharp] (done) | time=0ms
16:16:59.379 INFO Sensor C# File Caching Sensor [csharp]
16:16:59.379 WARN Incremental PR analysis: Could not determine common base path, cache will not be computed. Consider setting 'sonar.projectBaseDir' property.
16:16:59.379 INFO Sensor C# File Caching Sensor [csharp] (done) | time=6ms
16:16:59.379 INFO Sensor Zero Coverage Sensor
16:16:59.389 INFO Sensor Zero Coverage Sensor (done) | time=10ms
16:16:59.389 INFO SCM Publisher SCM provider for this project is: git
16:16:59.389 INFO SCM Publisher 4 source files to be analyzed
16:16:59.838 INFO SCM Publisher 4/4 source files have been analyzed (done) | time=449ms
16:16:59.846 INFO CPD Executor Calculating CPD for 0 files
16:16:59.846 INFO CPD Executor CPD calculation finished (done) | time=0ms
16:16:59.854 INFO SCM revision ID 'f2bc042c04c6e72427c380bcaee6d6fee7b49adf'
16:17:00.121 INFO Analysis report generated in 120ms, dir size=201.1 kB
16:17:00.195 INFO Analysis report compressed in 57ms, zip size=22.4 kB
16:17:00.393 INFO Analysis report uploaded in 195ms
16:17:00.394 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://127.0.0.1:9000/dashboard?id=sonarqube
16:17:00.395 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
16:17:00.395 INFO More about the report processing at http://127.0.0.1:9000/api/ce/task?id=acd819f5-9e70-42ab-bff7-3cc893e2cae4
16:17:00.405 INFO Analysis total time: 18.743 s
16:17:00.408 INFO SonarScanner Engine completed successfully
16:17:00.494 INFO EXECUTION SUCCESS
16:17:00.494 INFO Total time: 21.288s
Finished: SUCCESS
```

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



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

## Conclusion:

In this experiment, I learned how to integrate Jenkins with SonarQube for performing Static Application Security Testing (SAST). I set up SonarQube in a Docker container and configured Jenkins to use the SonarQube scanner. By creating a manual project in SonarQube and configuring the necessary authentication and tools in Jenkins, I established a seamless connection between Jenkins and SonarQube for static code analysis.

I tested the integration with a sample GitHub repository, successfully running a build and analyzing the project's code quality through SonarQube. This hands-on experience enhanced my understanding of the SAST process, Jenkins automation, and SonarQube's capabilities for identifying potential code vulnerabilities.

## Experiment - 8

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

### Theory:

#### **What is SAST?**

Static application security testing (SAST), or static analysis, is a testing methodology that analyzes source code to find security vulnerabilities that make your organization's applications susceptible to attack. SAST scans an application before the code is compiled. It's also known as white box testing.

#### **What problems does SAST solve?**

SAST takes place very early in the software development life cycle (SDLC) as it does not require a working application and can take place without code being executed. It helps developers identify vulnerabilities in the initial stages of development and quickly resolve issues without breaking builds or passing on vulnerabilities to the final release of the application.

SAST tools give developers real-time feedback as they code, helping them fix issues before they pass the code to the next phase of the SDLC. This prevents security-related issues from being considered an afterthought. SAST tools also provide graphical representations of the issues found, from source to sink. These help you navigate the code easier. Some tools point out the exact location of vulnerabilities and highlight the risky code. Tools can also provide in-depth guidance on how to fix issues and the best place in the code to fix them, without requiring deep security domain expertise.

It's important to note that SAST tools must be run on the application on a regular basis, such as during daily/monthly builds, every time code is checked in, or during a code release.

#### **Why is SAST important?**

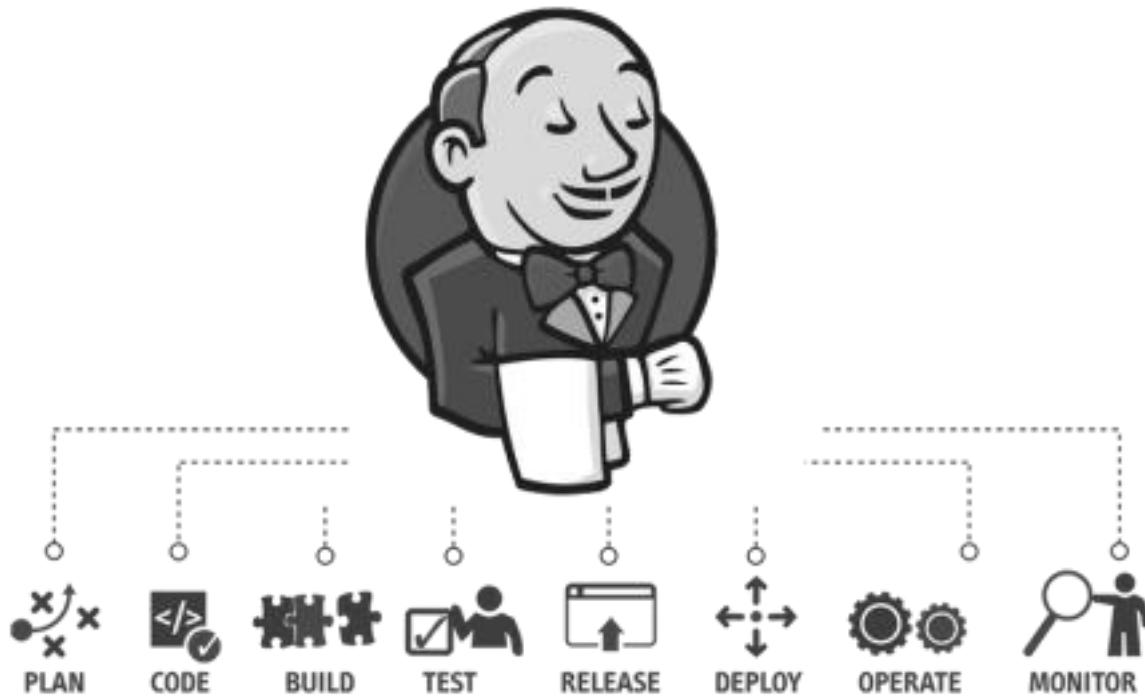
Developers dramatically outnumber security staff. It can be challenging for an organization to find the resources to perform code reviews on even a fraction of its applications. A key strength of SAST tools is the ability to analyze 100% of the codebase. Additionally, they are much faster than manual secure code reviews performed by humans. These tools can scan millions of lines of code

in a matter of minutes. SAST tools automatically identify critical vulnerabilities—such as buffer overflows, SQL injection, cross-site scripting, and others—with high confidence.

## What is a CI/CD Pipeline?

CI/CD pipeline refers to the Continuous Integration/Continuous Delivery pipeline. Before we dive deep into this segment, let's first understand what is meant by the term 'pipeline'?

A pipeline is a concept that introduces a series of events or tasks that are connected in a sequence to make quick software releases. For example, there is a task, that task has got five different stages, and each stage has got some steps. All the steps in phase one have to be completed, to mark the latter stage to be complete.



Now, consider the CI/CD pipeline as the backbone of the DevOps approach. This Pipeline is responsible for building codes, running tests, and deploying new software versions. The Pipeline executes the job in a defined manner by first coding it and then structuring it inside several blocks that may include several steps or tasks.

## What is SonarQube?

SonarQube is an open-source platform developed by SonarSource for continuous inspection of code quality. Sonar does static code analysis, which provides a detailed report of bugs, code smells, vulnerabilities, code duplications.

It supports 25+ major programming languages through built-in rulesets and can also be extended with various plugins.

## Benefits of SonarQube

- **Sustainability** - Reduces complexity, possible vulnerabilities, and code duplications, optimizing the life of applications.
- **Increase productivity** - Reduces the scale, cost of maintenance, and risk of the application; as such, it removes the need to spend more time changing the code
- **Quality code** - Code quality control is an inseparable part of the process of software development.
- **Detect Errors** - Detects errors in the code and alerts developers to fix them automatically before submitting them for output.
- **Increase consistency** - Determines where the code criteria are breached and enhances the quality
- **Business scaling** - No restriction on the number of projects to be evaluated
- **Enhance developer skills** - Regular feedback on quality problems helps developers to improve their coding skills

## Integrating Jenkins with SonarQube:

### Prerequisites:

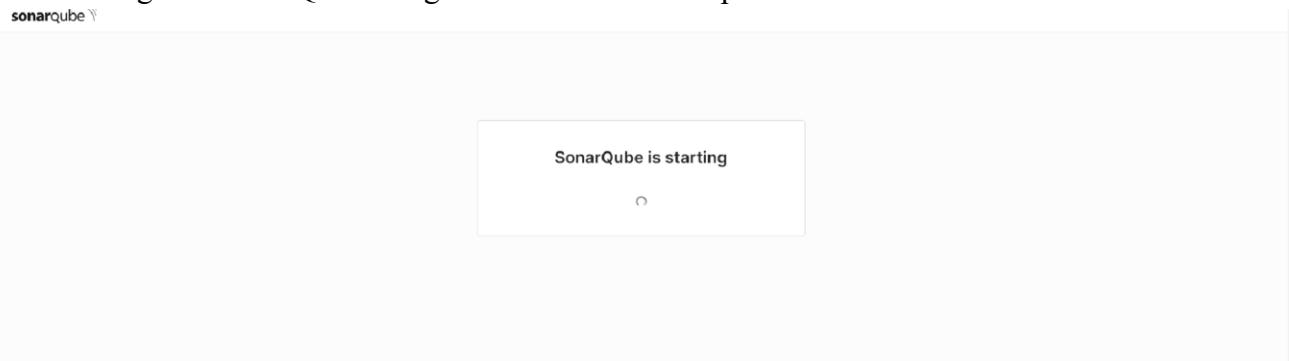
- Jenkins installed
- Docker Installed (for SonarQube)
- SonarQube Docker Image

### Steps to create a Jenkins CI/CD Pipeline and use SonarQube to perform SAST

1. Open up Jenkins Dashboard on localhost, port 8080 or whichever port it is at for you.
2. Run SonarQube in a Docker container using this command -

```
C:\Users\ADMIN>docker run -d --name sonarqube2 -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9001:9000 sonarqube:latest  
fda86b00e3989f3eb5aca8396b29b2a0adc95bcfe0fc5d85cf1237491e7678b9
```

- Once the container is up and running, you can check the status of SonarQube at localhost port 9000.
- Login to SonarQube using username *admin* and password *admin*.



- Create a manual project in SonarQube with the name **sonarqube-test**

1 of 2

## Create a local project

**Project display name \***

**Project key \***

**Main branch name \***

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

**Cancel** **Next**

Setup the project and come back to Jenkins Dashboard.

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

Enter an item name

Select an item type

- 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**

- Under Pipeline Script, enter the following -

```

node {
    stage('Cloning the GitHub Repo') {
        git 'https://github.com/shazforiot/GOL.git'
    }
    stage('SonarQube analysis') {
        withSonarQubeEnv('sonarqube') {
            sh "<PATH_TO SONARQUBE FOLDER>/bin//sonar-scanner \
                -D sonar.login=<SonarQube_USERNAME> \
                -D sonar.password=<SonarQube_PASSWORD> \
                -D sonar.projectKey=<Project_KEY> \
                -D sonar.exclusions=vendor/**,resources/**,**/*.java \
                -D sonar.host.url=http://127.0.0.1:9000/"
        }
    }
}

```

Pipeline

#### Definition

##### Pipeline script

```

Script ? 
1 stage('Cloning the GitHub Repo') {
2     git 'https://github.com/shazforiot/GOL.git'
3 }
4
5 stage('SonarQube analysis') {
6     withSonarQubeEnv('sonarqube') {
7         bat """
8             C:\ProgramData\Jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube\bin\sonar-scanner ^
9                 -D sonar.login=admin ^
10                -D sonar.password=admin123 ^
11                -D sonar.projectKey=sonarqube-test ^
12                -D sonar.exclusions=vendor/**,resources/**,**/*.java ^
13                -D sonar.host.url=http://127.0.0.1:9001/
14             """
15     }
16 }
17
18 }
19

```

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

8. Run The Build.

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

**SonarQube-8**

- Status
- </> Changes
- Build Now
- Configure
- Delete Pipeline
- Full Stage View
- Stages
- Rename
- Pipeline Syntax

Average stage times:  
(Average full run time: ~12min)

Cloning the GitHub Repo	SonarQube analysis
5s	31min 25s
1s	12min 7s
8s	50min 43s aborted

Build History

trend

Filter...

#4 | Sep 26, 2024, 6:04PM

#3 | Sep 26, 2024, 5:13PM

Dashboard > SonarQube-8 > #4

### Console Output

Skipping 4,240 KB. Full Log

```

18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 798. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 810. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 823. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 844. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 509. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 1065. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 776. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 778. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 530. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 648. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 798. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 546. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 546. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 634. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 778. Keep only the first 100 references.
18:13:39.657 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at line 75. Keep only the first 100 references.
18:13:39.657 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at line 41. Keep only the first 100 references.
18:13:39.657 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at line 17. Keep only the first 100 references.
18:13:39.657 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at line 296. Keep only the first 100 references.
18:13:39.657 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at line 75. Keep only the first 100 references.
18:13:39.657 INFO CPD Executor CPD calculation finished (done) | time=158971ms
18:13:39.674 INFO SCM revision ID 'ba799ba7e1b576f04a4612322b0412c5e6e1e5e4'
18:15:49.694 INFO Analysis report generated in 5022ms, dir size=127.2 MB
18:16:08.759 INFO Analysis report compressed in 19048ms, zip size=29.6 MB
18:16:09.884 INFO Analysis report uploaded in 1125ms
18:16:09.887 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://127.0.0.1:9001/dashboard?id=sonarqube-test
18:16:09.887 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
18:16:09.887 INFO More about the report processing at http://127.0.0.1:9001/api/ce/task?id=6f22c333-3777-4a21-b058-0ab4c049625c
18:16:22.970 INFO Analysis total time: 12:02.242 s
18:16:22.975 INFO SonarScanner Engine completed successfully
18:16:23.699 INFO EXECUTION SUCCESS
18:16:23.708 INFO Total time: 12:05.758s
[Pipeline]
[Pipeline] // withSonarQubeEnv
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

## 10. After that, check the project in SonarQube.

The screenshot shows the SonarQube main dashboard for the project 'sonarqube-test'. At the top, there's a navigation bar with links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, More, and a search bar. Below the navigation is a breadcrumb trail: sonarqube-test / main. The main content area has tabs for Overview, Issues, Security Hotspots, Measures, Code, and Activity, with 'Overview' selected. A large 'main' section displays a 'Quality Gate' status as 'Passed' with a green checkmark icon. It also shows '683k Lines of Code' and 'Version not provided'. A button to 'Set as homepage' is present. Below the main title, it says 'Last analysis 15 minutes ago'. The dashboard includes sections for Security (0 Open Issues), Reliability (68k Open Issues), Maintainability (164k Open Issues), Accepted issues (0), Coverage (On 0 lines to cover), and Duplications (50.6% on 759k lines). There's also a 'Security Hotspots' section with 3 items.

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

## 11. Code Problems -

The screenshot shows the SonarQube 'Issues' tab for the project 'sonarqube-test'. The top navigation bar and breadcrumb trail are identical to the main dashboard. The left sidebar has dropdown menus for Software Quality, Severity, Type, and Scope. Under 'Type', 'Bug' is selected. The main content area lists several code problems:

- Add "lang" and/or "xml:lang" attributes to this "html>" element (Reliability, Not assigned, L1 + 2min effort - 4 years ago, Major)
- Insert a <!DOCTYPE> declaration to before this <html> tag. (Reliability, Not assigned, L1 + 5min effort - 4 years ago, Major)
- Add "<th>" headers to this "table>". (Reliability, Not assigned, L9 + 2min effort - 4 years ago, Major)

A warning message at the bottom states: 'Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.'

## Code Smells

sonarqube

Projects Issues Rules Quality Profiles Quality Gates Administration More Q

star sonarqube-test / main

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

**Software Quality**

- Security: 0
- Reliability: 21k
- Maintainability: 164k

**Severity**

**Type**

- Bug: 47k
- Vulnerability: 0
- Code Smell: 164k

Add to selection Ctrl + click

**Scope**

gameoflife-acceptance-tests/Dockerfile

Use a specific version tag for the image. Intentionality: No tags

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

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

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

**Embedded database should be used for evaluation purposes only**  
The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.

SonarQube™ technology is powered by SonarSource SA

Community Edition v18.6 (#2115) ACTIVE | LGPL v3 | Community | Documentation | Plugins | Web API

## Intentional Issues

sonarqube

Projects Issues Rules Quality Profiles Quality Gates Administration More Q

star sonarqube-test / main

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

**Filters**

**Issues in new code**

**Clean Code Attribute**

- Consistency: 164k
- Intentionality: 268
- Adaptability: 0
- Responsibility: 0

Add to selection Ctrl + click

**Software Quality**

- Security: 0
- Reliability: 253
- Maintainability: 15

gameoflife-acceptance-tests/Dockerfile

Use a specific version tag for the image. Intentionality: No tags

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

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

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

**Embedded database should be used for evaluation purposes only**  
The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.

## Reliability Issue

sonarqube

Projects Issues Rules Quality Profiles Quality Gates Administration More Q

star sonarqube-test / main

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

**My Issues All**

**Filters**

**Issues in new code**

**Clean Code Attribute**

- Consistency: 21k
- Intentionality: 253
- Adaptability: 0
- Responsibility: 0

Add to selection Ctrl + click

**Software Quality**

- Security: 0
- Reliability: 21k
- Maintainability: 164k

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

Anchors must have content and the content must be accessible by a screen reader. Consistency: accessibility

Anchors must have content and the content must be accessible by a screen reader. Consistency: accessibility

Anchors must have content and the content must be accessible by a screen reader. Consistency: accessibility

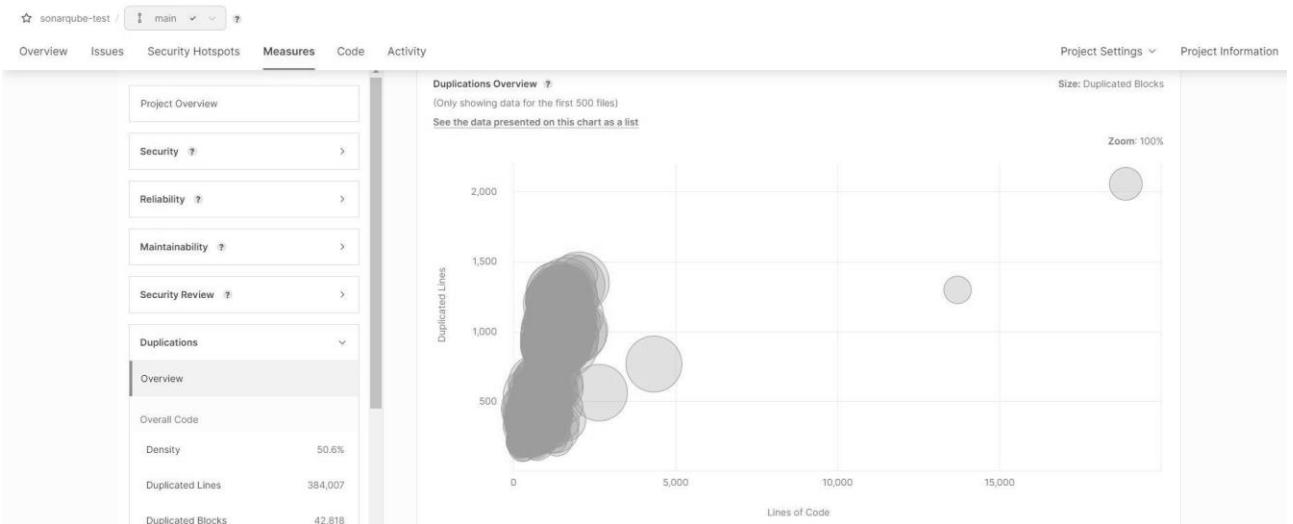
Anchors must have content and the content must be accessible by a screen reader. Consistency: accessibility

**Embedded database should be used for evaluation purposes only**

## Maintainability Issue

The screenshot shows the SonarQube Issues page for the project 'sonarqube-test'. The main navigation bar includes 'Projects', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', 'Administration', 'More', and a search icon. The 'Issues' tab is selected. On the left, there's a sidebar with 'My Issues' and 'All' buttons, and a 'Filters' section with 'Issues in new code' and 'Clean Code Attribute' and 'Software Quality' dropdowns. The main area displays a list of issues under 'gameoflife-core/build/reports/tests/all-tests.html'. Each issue has a checkbox, a title, a severity level (Consistency), and a detailed description. For example, one issue is 'Remove this deprecated "width" attribute.' with a 'Consistency' level of 'html5 obsolete'. The top right corner shows '163,786 issues' and '1705d effort'.

## Duplicates



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

## Conclusion:

In this experiment, I successfully created a CI/CD pipeline using Jenkins integrated with SonarQube for static code analysis on a sample Java application. I set up SonarQube in a Docker container and configured Jenkins to clone the GitHub repository and perform the analysis. The pipeline detected various issues, including bugs, code smells, and security vulnerabilities, which I reviewed in SonarQube. This experience enhanced my skills in configuring CI/CD tools and highlighted the importance of maintaining code quality through automation. Overall, I gained valuable insights into integrating tools for effective software development practices.

# Adv DevOps Practical 9

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

## Theory:

### What is Nagios?

Nagios is an open-source software for continuous monitoring of systems, networks, and infrastructures. It runs plugins stored on a server that is connected with a host or another server on your network or the Internet. In case of any failure, Nagios alerts about the issues so that the technical team can perform the recovery process immediately.

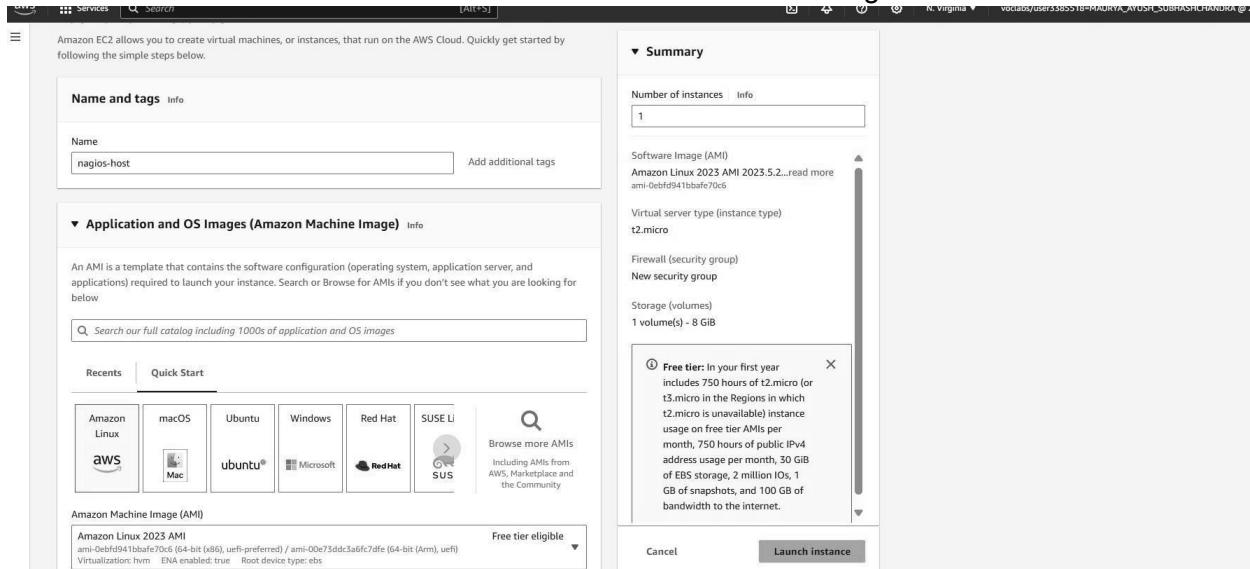
Nagios is used for continuous monitoring of systems, applications, service and business processes in a DevOps culture

### Installation of Nagios

**Prerequisites:** AWS Free Tier

### Steps:

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



The screenshot shows the AWS EC2 instance creation process. It includes:

- Instance type:** t2.micro (selected), showing details like Family: t2, 1 vCPU, 1 GiB Memory, and Current generation: true.
- Key pair (login):** exp\_09 (selected).
- Security Group:** A new security group named "New security group" is being created. It allows SSH traffic from anywhere and HTTPS traffic from the internet.
- Networking:** A new subnet is being created with a single public IPv4 address.
- Storage:** One volume (8 GiB) is attached.
- Free tier information:** A tooltip indicates the free tier covers 750 hours of t2.micro usage.
- EC2 Dashboard:** Shows the newly launched instances: "nagios-host" (running), "Master" (stopped), "node1" (stopping), "node2" (stopping), and "exp\_4" (stopped).

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

The screenshot shows the AWS Security Groups page with the following details:

Name	Security group ID	Security group name	VPC ID	Description	Owner
-	sg-070583550d576c53e	launch-wizard-2	vpc-0dd4c0d8f48c2e4508	launch-wizard-2 created 2024-09-27T...	217253764927
-	sg-030c0a1b62a1e9894	NodeGroup	vpc-0dd4c0d8f48c2e4508	Node	217253764927
-	sg-03f412e8ec9ec5946	launch-wizard-1	vpc-0dd4c0d8f48c2e4508	launch-wizard-1 created 2024-09-27T...	217253764927
-	sg-000c20590a5551206	default	vpc-0dd4c0d8f48c2e4508	default VPC security group	217253764927
-	sg-097fc30a345c1a537	MasterGroup	vpc-0dd4c0d8f48c2e4508	Master	217253764927
-	sg-09d51590eb1851b46	launch-wizard-3	vpc-0dd4c0d8f48c2e4508	launch-wizard-3 created 2024-09-29T...	217253764927

[EC2](#) > [Security Groups](#) > sg-09d51590eb1851b46

### sg-09d51590eb1851b46 - launch-wizard-3

[Actions ▾](#)

Details	
Security group name launch-wizard-3	Security group ID sg-09d51590eb1851b46
Owner 217253764927	Description launch-wizard-3 created 2024-09-29T06:49:51.498Z
	VPC ID vpc-0d4c0d8f48c2e4508
Owner 217253764927	Inbound rules count 1 Permission entry
	Outbound rules count 1 Permission entry

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

**Inbound rules (1)**

Inbound rules (1)									
<input type="button" value="C"/> Manage tags <a href="#">Edit inbound rules</a> <span style="float: right;">&lt; 1 &gt; ⌂</span>									
Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description		
-	sgr-0ec19557ab93305...	IPv4	SSH	TCP	22	0.0.0.0/0	-		

**Edit inbound rules** Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules									
<a href="#">Info</a> <span style="float: right;"><a href="#">Delete</a></span>									
Security group rule ID	Type	Info	Protocol	Info	Port range	Info	Source	Info	Description - optional
sgr-0ec19557ab9330565	SSH		TCP		22		Custom		<input type="text" value=""/> <a href="#">Delete</a>
-	HTTP		TCP		80		Anywhere-...		<input type="text" value=""/> <a href="#">Delete</a>
-	All ICMP - IPv6		IPv6 ICMP		All		Anywhere-...		<input type="text" value=""/> <a href="#">Delete</a>
-	HTTPS		TCP		443		Anywhere-...		<input type="text" value=""/> <a href="#">Delete</a>
-	All traffic		All		All		Anywhere-...		<input type="text" value=""/> <a href="#">Delete</a>
-	Custom TCP		TCP		5666		Anywhere-...		<input type="text" value=""/> <a href="#">Delete</a>
-	All ICMP - IPv4		ICMP		All		Anywhere-...		<input type="text" value=""/> <a href="#">Delete</a>

[Add rule](#)

Details	
Security group name launch-wizard-3	Security group ID sg-09d51590eb1851b46
Owner 217253764927	Description launch-wizard-3 created 2024-09-29T06:49:51.498Z
	VPC ID vpc-0d4c0d8f48c2e4508
Owner 217253764927	Inbound rules count 7 Permission entries
	Outbound rules count 1 Permission entry

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

**Inbound rules (7)**

Inbound rules (7)									
<input type="button" value="C"/> Manage tags <a href="#">Edit inbound rules</a> <span style="float: right;">&lt; 1 &gt; ⌂</span>									
Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description		
-	sgr-034c50eff5e5fa00	IPv4	All ICMP - IPv6	IPv6 ICMP	All	0.0.0.0/0	-		
-	sgr-038d0d3791dfcc60e	IPv4	HTTPS	TCP	443	0.0.0.0/0	-		
-	sgr-0e8ad1dd008b14...	IPv4	All ICMP - IPv4	ICMP	All	0.0.0.0/0	-		
-	sgr-0ec19557ab93305...	IPv4	SSH	TCP	22	0.0.0.0/0	-		
-	sgr-00a0e56d560959f45	IPv4	HTTP	TCP	80	0.0.0.0/0	-		
-	sgr-064c062d69916fa84	IPv4	Custom TCP	TCP	5666	0.0.0.0/0	-		
-	sgr-0613b7b6aa9d30def	IPv4	All traffic	All	All	0.0.0.0/0	-		

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

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

## Connect to instance Info

Connect to your instance i-0011127bbfdb2f467 (nagios-host) using any of these options

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

Instance ID

i-0011127bbfdb2f467 (nagios-host)

1. Open an SSH client.

2. Locate your private key file. The key used to launch this instance is `exp_09.pem`

3. Run this command, if necessary, to ensure your key is not publicly viewable.  
 `chmod 400 "exp_09.pem"`

4. Connect to your instance using its Public DNS:

`ec2-44-204-11-28.compute-1.amazonaws.com`

Example:

`ssh -i "exp_09.pem" ec2-user@ec2-44-204-11-28.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.

Or open command prompt and paste ssh command.

```
Microsoft Windows [Version 10.0.22631.4169]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Ayush Maurya>ssh -i "Downloads/exp_09.pem" ec2-user@ec2-44-204-11-28.compute-1.amazonaws.com
The authenticity of host 'ec2-44-204-11-28.compute-1.amazonaws.com (44.204.11.28)' can't be established.
ED25519 key fingerprint is SHA256:v20KH/ezl9iu7/RT6m8LWkgWzEJnnQIqrG9gKWzwC14.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-204-11-28.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

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

Last login: Sun Sep 29 07:11:40 2024 from 18.206.107.27
[ec2-user@ip-172-31-91-91 ~]$ |
```

## **sudo yum update**

```
[ec2-user@ip-172-31-91-91 ~]$  
sudo yum update  
Last metadata expiration check: 0:19:03 ago on Sun Sep 29 06:56:15 2024.  
Dependencies resolved.  
Nothing to do.  
Complete!  
[ec2-user@ip-172-31-91-91 ~]$ |
```

**sudo yum install httpd php**

```
[ec2-user@ip-172-31-91-91 ~]$ sudo yum install httpd php
Last metadata expiration check: 0:19:29 ago on Sun Sep 29 06:56:15 2024.
Dependencies resolved.
=====
Package           Architecture     Version          Repository      Size
=====
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-httd
noarch           18.0.0-12.amzn2023.0.3
httdp-core       x86_64          2.4.62-1.amzn2023
httdp-filesystem noarch           2.4.62-1.amzn2023
httdp-tools      x86_64          2.4.62-1.amzn2023
libbrotli        x86_64          1.0.9-4.amzn2023.0.2
libsodium        x86_64          1.0.19-4.amzn2023
libxml2          x86_64          1.1.34-5.amzn2023.0.2
mailcap          noarch           2.1.49-3.amzn2023.0.3
nginx-filesystem noarch           1:1.24.0-1.amzn2023.0.4
php8_3-cli       x86_64          8.3.10-1.amzn2023.0.1
php8_3-common    x86_64          8.3.10-1.amzn2023.0.1
php8_3-process   x86_64          8.3.10-1.amzn2023.0.1
php8_3-xml       x86_64          8.3.10-1.amzn2023.0.1
=====
Installing weak dependencies:
apr-util-openssl x86_64          1.6.3-1.amzn2023.0.1
mod_nginx        x86_64          2.0.27-1.amzn2023.0.3
mod_lua          x86_64          2.4.62-1.amzn2023
php8_3-fpm       x86_64          8.3.10-1.amzn2023.0.1
php8_3-mbstring  x86_64          8.3.10-1.amzn2023.0.1
php8_3-opcache   x86_64          8.3.10-1.amzn2023.0.1
php8_3-pdo       x86_64          8.3.10-1.amzn2023.0.1
php8_3-sodium    x86_64          8.3.10-1.amzn2023.0.1
=====
Transaction Summary
=====
Total download size: 22 MB/s | 10 MB 00:00
=====
Preparing : 1/1
Installing : php8_3-common-8.3.10-1.amzn2023.0.1.x86_64 2/25
Installing : apr-1.7.2-2.amzn2023.0.2.x86_64 3/25
Installing : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64 4/25
Installing : mod_nginx-2.1.49-3.amzn2023.0.3.noarch 5/25
Running scriptlet: httdp-filesystem-2.4.62-1.amzn2023.noarch 6/25
```

## **sudo yum install gcc glibc glibc-common**

```
[ec2-user@ip-172-31-91-91 ~]$ sudo yum install gcc glibc glibc-common
Last metadata expiration check: 0:20:41 ago on Sun Sep 29 06:56:15 2024.
Package glibc-2.34-52.amzn2023.0.11.x86_64 is already installed.
Package glibc-common-2.34-52.amzn2023.0.11.x86_64 is already installed.
Dependencies resolved.
=====
Package           Architecture     Version          Repository      Size
=====
Installing:
gcc              x86_64          11.4.1-2.amzn2023.0.2
=====
Installing dependencies:
annobin          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
gtl              x86_64          8.8.0-5.amzn2023.0.2
glibc-devel      noarch           2.34-52.amzn2023.0.11
glibc-headers-x86 x86_64          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
libcrypt-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:
annobin-docs-10.93-1.amzn2023.0.1.noarch
gcc-8.0.4-5.amzn2023.0.2.x86_64
glibc-headers-x86-2.34-52.amzn2023.0.11.noarch
libmpc-1.2.1-2.amzn2023.0.2.x86_64
make-1.4.3-5.amzn2023.0.2.x86_64
=====
cpp-11.4.1-2.amzn2023.0.2.x86_64
glibc-devel-2.34-52.amzn2023.0.11.x86_64
kernel-headers-6.1.109-118.189.amzn2023.x86_64
libcrypt-devel-4.4.33-7.amzn2023.x86_64
=====
Complete!
```

## **sudo yum install gd gd-devel**

```
[ec2-user@ip-172-31-91-91 ~]$ sudo yum install gd gd-devel
Last metadata expiration check: 0:21:30 ago on Sun Sep 29 06:56:15 2024.
Dependencies resolved.
=====
Package           Architecture     Version          Repository      Size
=====
Installing:
gd              x86_64          2.3.3-5.amzn2023.0.3
gd-devel        x86_64          2.3.3-5.amzn2023.0.3
=====
Installing dependencies:
brotli          x86_64          1.0.9-4.amzn2023.0.2
brotli-devel    x86_64          1.0.9-4.amzn2023.0.2
bz2              x86_64          1.0.9-4.amzn2023.0.2
cairo           x86_64          1.17.6-2.amzn2023.0.1
cmake-filesystem x86_64          3.22.2-1.amzn2023.0.4
fontconfig       x86_64          2.13.94-2.amzn2023.0.2
=====
amazonlinux      139 k
amazonlinux      38 k
amazonlinux      314 k
amazonlinux      31 k
amazonlinux      234 k
amazonlinux      684 k
amazonlinux      16 k
amazonlinux      273 k
```

```

Installed:
brotli-1.0.9-4.amzn2023.0.2.x86_64
cairo-1.17.6-2.amzn2023.0.1.x86_64
fontconfig-devel-2.13.94-2.amzn2023.0.2.x86_64
freetype-devel-2.13.2-5.amzn2023.0.1.x86_64
glib2-devel-2.74.7-689.amzn2023.0.2.x86_64
graphite2-1.3.14-7.amzn2023.0.2.x86_64
harfbuzz-devel-7.0.0-2.amzn2023.0.1.x86_64
langpacks-core-font-en-3.0-21.amzn2023.0.4.noarch
libDX11-1.7.2-2.amzn2023.0.4.x86_64
libEGL-1.4.10-10.amzn2023.0.4.x86_64
libExt2-1.3.4-6.amzn2023.0.2.x86_64
libXrender-0.9.10-14.amzn2023.0.2.x86_64
libffi-devel-3.4.4-1.amzn2023.0.1.x86_64
libjpeg-turbo-2.1.4-2.amzn2023.0.5.x86_64
libpng-2.1.6.37-18.amzn2023.0.6.x86_64
libsep0-devel-3.4-3.amzn2023.0.3.x86_64
libwebrtc-1.4-1.amzn2023.0.6.x86_64
libxcb-devel-1.13.1-7.amzn2023.0.2.x86_64
pcre2-utf16-10.40-1.amzn2023.0.3.x86_64
sysprof-capture-devel-10.40.1-2.amzn2023.0.2.x86_64
xz-devel-5.2.5-9.amzn2023.0.2.x86_64

brotli-devel-1.0.9-4.amzn2023.0.2.x86_64
cmake-fs-3.22.2-1.amzn2023.0.4.x86_64
fonts-fs-1.2.0.5-12.amzn2023.0.2.noarch
gd-2.3.3-5.amzn2023.0.3.x86_64
google-noto-fonts-common-20201206-2.amzn2023.0.2.noarch
graphite2-devel-1.3.14-7.amzn2023.0.2.x86_64
harfbuzz-icu-7.0.0-2.amzn2023.0.1.x86_64
jbigkit-libs-2.1-21.amzn2023.0.2.x86_64
libSM-1.2.3-8.amzn2023.0.2.x86_64
libX11-devel-1.7.2-3.amzn2023.0.4.x86_64
libXau-1.0.8-6.amzn2023.0.4.x86_64
libXext-1.3.4-6.amzn2023.0.4.x86_64
libXt-1.2.0-4.amzn2023.0.2.x86_64
libicu-67.1-7.amzn2023.0.3.x86_64
libjpeg-turbo-devel-2.1.4-2.amzn2023.0.5.x86_64
libpng-devel-2.1.6.37-18.amzn2023.0.6.x86_64
libtiff-4.0-4.amzn2023.0.18.x86_64
libwebrtc-devel-1.2.4-1.amzn2023.0.6.x86_64
libxml2-devel-2.18.4-1.amzn2023.0.6.x86_64
pcre2-utf32-10.40-1.amzn2023.0.3.x86_64
xml-common-0.6.3-56.amzn2023.0.2.noarch
zlib-devel-1.2.11-33.amzn2023.0.5.x86_64

bzip2-devel-1.0.8-6.amzn2023.0.2.x86_64
fontconfig-2.13.94-2.amzn2023.0.2.x86_64
freetype-2.13.2-5.amzn2023.0.1.x86_64
gd-devel-2.3.3-5.amzn2023.0.3.x86_64
google-noto-sans-vf-fonts-20201206-2.amzn2023.0.2.noarch
harfbuzz-7.0.0-2.amzn2023.0.1.x86_64
jbigkit-libs-2.1-21.amzn2023.0.2.x86_64
libSM-1.2.3-8.amzn2023.0.2.x86_64
libX11-devel-1.7.2-3.amzn2023.0.4.x86_64
libXau-1.0.8-6.amzn2023.0.4.x86_64
libXpm-devel-3.0.1-1.amzn2023.0.3.x86_64
libblkid-devel-2.37.4-1.amzn2023.0.4.x86_64
libicu-devel-67.1-7.amzn2023.0.3.x86_64
libmount-devel-2.37.4-1.amzn2023.0.4.x86_64
libselinux-devel-3.4-5.amzn2023.0.2.x86_64
libtiff-devel-4.0-4.amzn2023.0.18.x86_64
libxcb-1.13.1-7.amzn2023.0.2.x86_64
pcre2-devel-10.40-1.amzn2023.0.3.x86_64
pixman-0.40.0-3.amzn2023.0.3.x86_64
xorg-x11proto-devel-2021.4-1.amzn2023.0.2.noarch

Complete!

```

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

**sudo adduser -m nagios**

**sudo passwd nagios**

**(password : ayushmau)**

```

Complete!
[ec2-user@ip-172-31-91-91 ~]$ 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:
Sorry, passwords do not match.
New password:
Sorry, passwords do not match.
The password contains the user name in some form
Retype new password:
Sorry, passwords do not match.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-91-91 ~]$

```

6. Create a new user group **sudo groupadd nagcmd**

```

[ec2-user@ip-172-31-91-91 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-91-91 ~]$

```

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

```

[ec2-user@ip-172-31-91-91 ~]$ sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
[ec2-user@ip-172-31-91-91 ~]$

```

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

```

[ec2-user@ip-172-31-91-91 ~]$ mkdir ~/downloads
cd ~/downloads
~/downloads

```

9. Use wget to download the source zip files.

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

```
[ec2-user@ip-172-31-91-91 downloads]$ cd ..
[ec2-user@ip-172-31-91-91 ~]$ cd ~/downloads
[ec2-user@ip-172-31-91-91 downloads]$ wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
--2024-09-29 09:11:59-- 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 5.07MB/s in 0.4s
2024-09-29 09:11:59 (5.07 MB/s) - 'nagios-4.5.5.tar.gz' saved [2065473/2065473]

[ec2-user@ip-172-31-91-91 downloads]$ |
```

```
wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
[ec2-user@ip-172-31-91-91 nagios-4.5.5]$ cd ..
[ec2-user@ip-172-31-91-91 downloads]$ wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
--2024-09-29 09:14:28-- 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-2.4 100%[=====] 2.62M 6.92MB/s in 0.4s
```

10. Use tar to unzip and change to that directory. tar zxvf nagios-4.5.5.tar.gz

```
[ec2-user@ip-172-31-91-91 downloads]$ tar zxvf nagios-4.0.8.tar.gz
nagios-4.0.8/
nagios-4.0.8/.gitignore
nagios-4.0.8/Changelog
nagios-4.0.8/INSTALLING
nagios-4.0.8/LEGAL
nagios-4.0.8/LICENSE
nagios-4.0.8/Makefile.in
nagios-4.0.8/README
nagios-4.0.8/README.asciidoc
nagios-4.0.8/THANKS
nagios-4.0.8/UPGRADING
nagios-4.0.8/base/
nagios-4.0.8/base/.gitignore
```

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

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

Here we go an error

```
[ec2-user@ip-172-31-91-91 downloads]$ ./configure --with-command-group=nagcmd  
-bash: ./configure: No such file or directory  
[ec2-user@ip-172-31-91-91 downloads]$ |
```

### Solution

Navigate to nagios folder in downloads

```
[ec2-user@ip-172-31-91-91 downloads]$ ls  
nagios-4.0.8  nagios-4.0.8.tar.gz  nagios-plugins-2.0.3.tar.gz  
[ec2-user@ip-172-31-91-91 downloads]$ cd nagios-4.0.8  
[ec2-user@ip-172-31-91-91 nagios-4.0.8]$ |
```

Error 2: Cannot find SSL headers.

Solution: Install openssl dev library

Steps: sudo yum install

openssl-devel

```
[ec2-user@ip-172-31-91-91 nagios-4.5.5]$ sudo yum install openssl-devel  
Last metadata expiration check: 2:24:05 ago on Sun Sep 29 06:56:15 2024.  
Dependencies resolved.  
=====  
 Package          Arch      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  
Downloading Packages:
```

Now run

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

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

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

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

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

12. Compile the source code. **make all**

```
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
gcc -Wall -I.. -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
```

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

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

sudo make install
sudo make install-init
sudo make install-config
sudo make install-commandmode
cd ./base && make
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
gcc -Wall -I.. -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 nebmods.c
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 workers.c
In function 'get_wproc_list',
  inlined from 'get_worker' at workers.c:277:12:
workers.c:253:17: warning: '%s' directive argument is null [-Wformat-overflo
w=]
  253 |           log_debug_info(DEBUGL_CHECKS, 1, "Found specialized
worker(s) for '%s'", (slash && *slash != '/') ? slash : cmd_name);
  |
gcc -Wall -I.. -I. -I./lib -I./include -I../include -I.. -g -O2 -DHAVE_
CONFIG_H -DNSCORE -c -o checks.o checks.c
gcc -Wall -I.. -I. -I./lib -I./include -I../include -I.. -g -O2 -DHAVE_
CONFIG_H -DNSCORE -c -o config.o config.c
gcc -Wall -T.. -T. -T./lib -T../include -T.. -g -O2 -DHAVF
```

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

```

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

define contact {
    contact_name      nagiosadmin          ; Short name of user
    use               generic-contact       ; Inherit default values from generic-contact template (defined above)
    alias             Nagios Admin        ; Full name of user
    email             2022.ayush.maurya@ves.ac.in ; <***** CHANGE THIS TO YOUR EMAIL ADDRESS *****
}

#####
# CONTACT GROUPS
#
#####

# We only have one contact in this simple configuration file, so there is
# no need to create more than one contact group.

define contactgroup {
    contactgroup_name   admins
    alias              Nagios Administrators
    members            nagiosadmin
}

```

And change email with your email

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

```

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

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

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

```

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

```

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

```

Password: Ayushmau

### 17. Restart Apache **sudo service httpd restart**

```
Adding password for user nagiosadmin
[ec2-user@ip-172-31-91-91 nagios-4.5.5]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-91-91 nagios-4.5.5]$ |
```

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

```
cd ~/downloads tar zxvf nagios-
plugins-2.4.11.tar.gz
```

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

19. Compile and install plugins **cd nagios-plugins-2.4.11**

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

```
[ec2-user@ip-172-31-91-91 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)
checking dependency style of gcc... gcc3
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
checking for Minix Amsterdam compiler... no
checking for ar... ar
checking for ranlib... ranlib
```

**make sudo make**

**install**

```
[ec2-user@ip-172-31-91-91 nagios-plugins-2.4.11]$ make
sudo make install
make all-recursive
make[1]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
Making all in gl
make[2]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/
gl'
rm -f alloca.h-t alloca.h && \
{ echo '/* DO NOT EDIT! GENERATED AUTOMATICALLY! */'; \
cat ./alloca.in.h; \
} > alloca.h-t && \
mv -f alloca.h-t alloca.h
rm -f c++defs.h-t c++defs.h && \
sed -n -e '/_GL_CXXDEFS/, $p' \
< ./build-aux/snippet/c++defs.h \
> c++defs.h-t && \
mv c++defs.h-t c++defs.h
rm -f warn-on-use.h-t warn-on-use.h && \
sed -n -e '/^.ifndef/, $p' \
< ./build-aux/snippet/warn-on-use.h \
> warn-on-use.h-t && \
mv warn-on-use.h-t warn-on-use.h
rm -f arg-nonnull.h-t arg-nonnull.h && \
sed -n -e '/GL_ARG_NONNULL/, $p' \
< ./build-aux/snippet/arg-nonnull.h \
> arg-nonnull.h-t && \
mv arg-nonnull.h-t arg-nonnull.h
/usr/bin/mkdir -p arpa
u
make[1]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[2]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
[ec2-user@ip-172-31-91-91 nagios-plugins-2.4.11]$
```

## 20. Start Nagios

Add Nagios to the list of system services

**sudo chkconfig --add nagios sudo**

**chkconfig nagios on**

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

Verify the sample configuration files

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

*Error*

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

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

Solution:

**# Create the missing directory:** If the directory is missing, create it with the necessary permissions:

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

```
[ec2-user@ip-172-31-91-91 nagios-plugins-2.0.3]$ sudo mkdir -p /usr/local/nagios/var/spool/checkresults
sudo chown nagios:nagios /usr/local/nagios/var/spool/checkresults
sudo chmod 775 /usr/local/nagios/var/spool/checkresults
[ec2-user@ip-172-31-91-91 nagios-plugins-2.0.3]$
```

Now run again **sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg**

```
[ec2-user@ip-172-31-91-91 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.
    Checked 0 service escalations.
Checking for circular paths...
    Checked 1 hosts
    Checked 0 service dependencies
    Checked 0 host dependencies
    Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors: 0
```

**sudo service nagios start**

```
[ec2-user@ip-172-31-91-91 nagios-plugins-2.4.11]$ sudo service nagios start
Starting nagios (via systemctl): [ OK ]
```

21. Check the status of Nagios

**sudo systemctl status nagios**

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

Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: wproc: Registry request: name=Core Worker 68063;pid=68063
Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: wproc: Registry request: name=Core Worker 68062;pid=68062
Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: wproc: Registry request: name=Core Worker 68064;pid=68064
Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: wproc: Registry request: name=Core Worker 68061;pid=68061
Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: Warning: Could not open object cache file '/usr/local/nagios/var/objec>
Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmxp2N>
Sep 29 08:04:30 ip-172-31-91-91.ec2.internal nagios[68059]: Successfully launched command file worker with pid 68065
Sep 29 08:04:39 ip-172-31-91-91.ec2.internal nagios[68059]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmpTng>
Sep 29 08:04:49 ip-172-31-91-91.ec2.internal nagios[68059]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmpAfy>
Sep 29 08:04:59 ip-172-31-91-91.ec2.internal nagios[68059]: Error: Unable to create temp file '/usr/local/nagios/var/nagios.tmpCtq>
lines 1-26/26 (END)
```

### Error:

The log messages suggest that Nagios is unable to create temporary files, particularly in the directory [/usr/local/nagios/var/](#). This is typically caused by permission issues, or the directory might not exist. **Solution:**

Firstly check whether [/usr/local/nagios/var/](#) is there or not. If yes.....

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

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

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

Set permissions: Ensure the directory has the right permissions:

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

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

**sudo systemctl restart nagios**

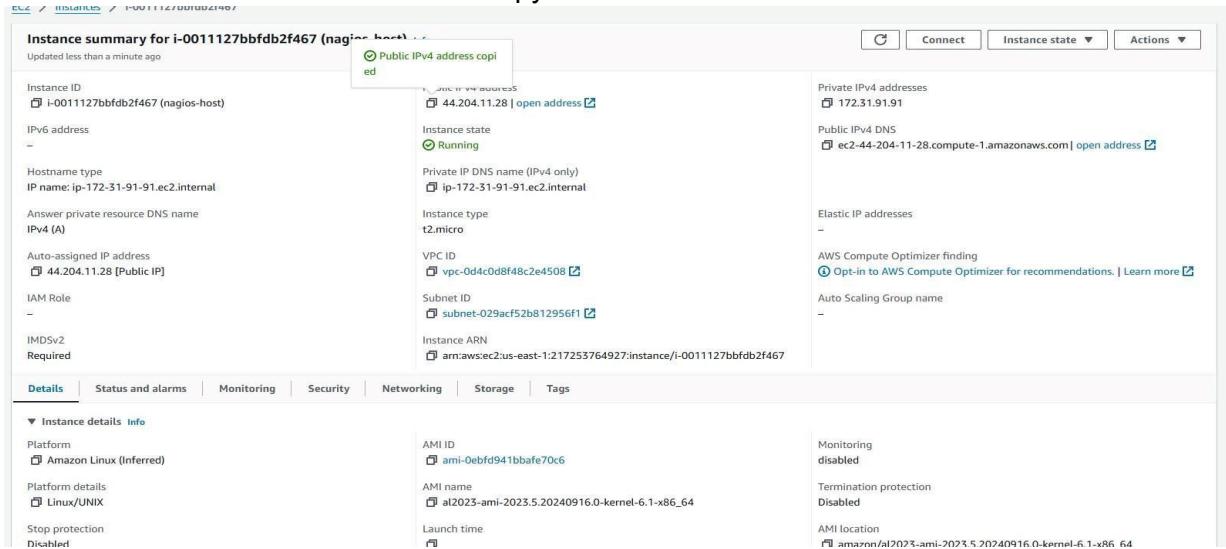
```
drwxr-xr-x. 4 root root 112 Sep 29 08:04 /usr/local/nagios/var/
[ec2-user@ip-172-31-91-91 nagios-plugins-2.0.3]$ sudo chown -R nagios:nagcmd /usr/local/nagios/var
[ec2-user@ip-172-31-91-91 nagios-plugins-2.0.3]$ sudo chmod -R 775 /usr/local/nagios/var
[ec2-user@ip-172-31-91-91 nagios-plugins-2.0.3]$ sudo systemctl restart nagios
[ec2-user@ip-172-31-91-91 nagios-plugins-2.0.3]$ |
```

Now run again

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

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

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



## 23. Open up your browser and look for

<http://<your public ip address>/nagios> Enter username as nagiosadmin and

password which you set in Step 16.

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

The screenshot shows the Nagios Core web interface at the URL 44.204.11.28/nagios/. The page title is "Nagios® Core™ Version 4.5.5". A banner at the top right says "Daemon running with PID 71188". The left sidebar has a "General" tab selected, showing links for Home, Documentation, Current Status (Tactical Overview, Map, Hosts, Services, Host Groups, Inventory, Grid), Service Groups (Summary, Grid), Problems (Services, 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 sections for Get Started (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), Latest News (empty), and Don't Miss... (empty). A "Quick Links" sidebar on the right lists Nagios Library, Nagios Labs, Nagios Exchange, Nagios Support, Nagios.com, and Nagios.org. At the bottom, there are copyright notices and a "Page Tour" link.

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

### Conclusion:

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

# Adv DevOps Practical 10

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

## Theory:

### Port and Service Monitoring

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

### Windows/Linux Server Monitoring

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

## Prerequisites:

### AWS Academy or Personal account.

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

## Monitoring Using Nagios:

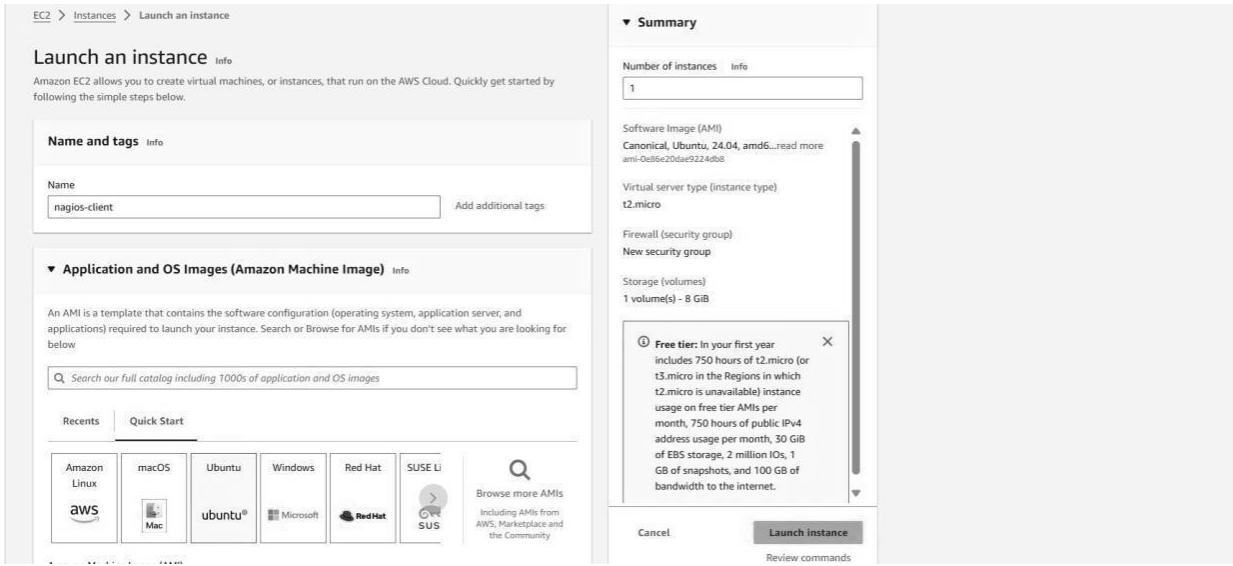
**Step 1:** To Confirm Nagios is running on the server side Perform the following command on your Amazon Linux Machine (Nagios-host). **sudo systemctl status nagios**

```
tee2-user@ip-172-31-91-91 ~$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
   Active: active (running) since Sun 2024-09-29 16:18:08 UTC; 21min ago
     Docs: https://www.nagios.org/documentation
 Process: 1942 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Process: 1944 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 1946 (nagios)
   Tasks: 8 (limit: 1112)
    Memory: 7.7M
      CPU: 387ms
     CGroup: /system.slice/nagios.service
             └─1946 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
                 ├─1947 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                 ├─1948 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                 ├─1949 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                 ├─1950 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                 ├─1956 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
                 ├─3088 /usr/local/nagios/libexec/check_ping -H 127.0.0.1 -w 3000.0,80% -c 5000.0,100% -p 5
                 └─3089 /usr/bin/ping -n -U -w 30 -c 5 127.0.0.1

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

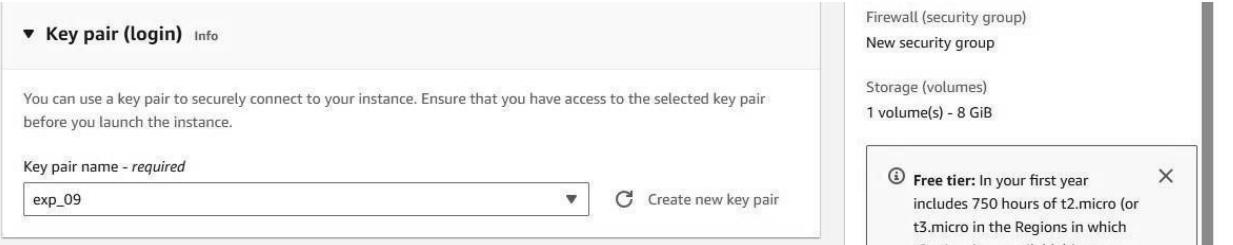
You can now proceed if you get the above message/output.

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

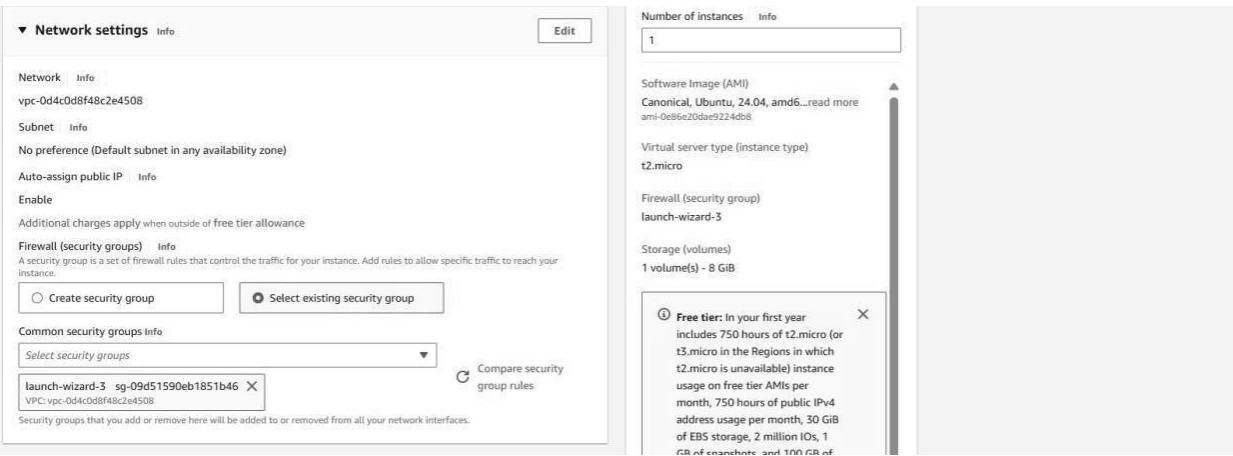


**For Key pair :** Click on create key and make key of type RSA with extension .pem . Key will be downloaded to your local machine.

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



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



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

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

```
PS C:\Users\ MUSKAANNN > ssh -i "Downloads/exp_09.pem" ubuntu@ec2-44-206-245-149.compute-1.amazonaws.com
The authenticity of host 'ec2-44-206-245-149.compute-1.amazonaws.com (44.206.245.149)' can't be established.
ED25519 key fingerprint is SHA256:Dt+A+mkydh3kOJ2vEpn4ZsA6FL+LM4m1QSImddAHg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-206-245-149.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

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

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

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

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

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

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

**ps -ef | grep nagios**

```
[ec2-user@ip-172-31-91-91 ~]$ ps -ef | grep nagios
nagios 1946 1 0 16:18 ? 00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios 1947 1946 0 16:18 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 1948 1946 0 16:18 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 1949 1946 0 16:18 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 1950 1946 0 16:18 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 1956 1946 0 16:18 ? 00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
root 3890 3855 0 16:40 pts/0 00:00:00 sudo systemctl status nagios
root 3892 3890 0 16:40 pts/1 00:00:00 sudo systemctl status nagios
root 3893 3892 0 16:40 pts/1 00:00:00 systemctl status nagios
[ec2-user 3914 3890 0 16:59 pts/2 00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-91-91 ~]$ |
```

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

**sudo su**

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

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

**Step 6:** Copy the sample localhost.cfg to linuxhost.cfg by running the following command.(Below command should come in one line see screenshot below) **cp /usr/local/nagios/etc/objects/localhost.cfg**

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

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

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

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

Change hostname to **linuxserver**.

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

Set hostgroup\_name to **linux-servers1**.

```
#####
# HOST DEFINITION
#####
# Define a host for the local machine

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

    host_name    linuxserver
    alias        localhost
    address      172.31.92.146
}

#####
# HOST GROUP DEFINITION
#####
# Define an optional hostgroup for Linux machines

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

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

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

**cfg\_dir=/usr/local/nagios/etc/objects/monitorhosts/**

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

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

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

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

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

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

```
[root@ip-172-31-91-91 ec2-user]# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
Nagios Core 4.5.5
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2024-09-17
License: GPL

Running pre-flight check on configuration data...

Checking objects...
    Checked 8 services.
    Checked 2 hosts.
    Checked 2 host groups.
    Checked 0 service groups.
    Checked 1 contacts.
    Checked 1 contact groups.
    Checked 24 commands.
    Checked 5 time periods.
    Checked 0 host escalations.
    Checked 0 service escalations.
Checking for circular paths...
    Checked 2 hosts
    Checked 0 service dependencies
    Checked 0 host dependencies
    Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors: 0

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

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

**service nagios restart**

```
[root@ip-172-31-91-91 ec2-user]# service nagios restart
Restarting nagios (via systemctl): [ OK ]
[root@ip-172-31-91-91 ec2-user]# |
```

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

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

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

Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [380 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [82.9 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4568 B]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [272 kB]
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [115 kB]
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [8632 B]
Get:14 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [10.3 kB]
Get:15 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [353 kB]
Get:16 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [68.1 kB]
Get:17 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 c-n-f Metadata [428 B]
Get:18 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [10.9 kB]
Get:19 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [2888 B]
Get:20 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:21 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [344 B]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
```

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

No containers need to be restarted.

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

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

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

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

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

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

**sudo systemctl restart nagios-nrpe-server**

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

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

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

Sep 29 17:20:07 ip-172-31-91-91.ec2.internal nagios[4763]: Warning: Duplicate definition found for service 'Current Users' on host 'localhost' (config file
Sep 29 17:20:07 ip-172-31-91-91.ec2.internal nagios[4763]: Warning: Duplicate definition found for service 'Root Partition' on host 'localhost' (config fil
Sep 29 17:20:07 ip-172-31-91-91.ec2.internal nagios[4763]: Warning: Duplicate definition found for service 'PING' on host 'localhost' (config file '/usr/lo
Sep 29 17:20:07 ip-172-31-91-91.ec2.internal nagios[4763]: Successfully launched command file worker with pid 4768
Sep 29 17:22:38 ip-172-31-91-91.ec2.internal nagios[4763]: SERVICE NOTIFICATION: nagiosadmin;localhost;Swap Usage;CRITICAL;notify-service-by-email;SWAP CRI
Sep 29 17:22:38 ip-172-31-91-91.ec2.internal nagios[4763]: wproc: NOTIFY job 1 from worker Core Worker 4766 is non-check helper but exited with return co
Sep 29 17:22:38 ip-172-31-91-91.ec2.internal nagios[4763]: wproc: host=localhost; service=Swap Usage; contact=nagiosadmin
Sep 29 17:22:38 ip-172-31-91-91.ec2.internal nagios[4763]: wproc: early_timeout=0; exited_ok=1; wait_status=32512; error_code=0;
Sep 29 17:22:38 ip-172-31-91-91.ec2.internal nagios[4763]: wproc: stderr line 01: /bin/sh: line 1: /bin/mail: No such file or directory
Sep 29 17:22:38 ip-172-31-91-91.ec2.internal nagios[4763]: wproc: stderr line 02: /usr/bin/printf: write error: Broken pipe
[lines 1-28 (END)]
```

**sudo systemctl status httpd**

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

**sudo systemctl start httpd**

**sudo systemctl enable httpd**

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

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

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

**Nagios® Core™ Version 4.5.5**

September 17, 2024 Check for updates

Get Started

- Start monitoring your infrastructure
- Change the look and feel of Nagios
- Addons
- Get support
- Get training
- Get certified

Latest News

Don't Miss...

Quick Links

- Nagios Library (tutorials and docs)
- Nagios Labs (development blog)
- Nagios Exchange (plugins and addons)
- Nagios Support (tech support)
- Nagios.com (company)
- Nagios.org (project)

Copyright © 2010-2024 Nagios Core Development Team and Community Contributors. Copyright © 1999-2009 Ethan Galstad. See the THANKS file for more information on contributors.

Nagios Core is licensed under the GNU General Public License and is provided AS IS with NO WARRANTY OF ANY KIND, INCLUDING THE WARRANTY OF DESIGN, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. Nagios, Nagios Core and the Nagios logo are trademarks, service marks, registered trademarks or unregistered trademarks owned by Naxos Technologies, LLC. Use of the Naxos mark is reserved by the trademark owner.

### Now Click on Hosts from left side panel

Up	Down	Unreachable	Pending	Ok	Warning	Unknown	Critical	Pending
2	0	0	0	6	1	0	1	0
All Problems	All Types			All Problems	All Types			
0	2			2	8			

Host Status Details For All Host Groups

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

Results 1 - 2 of 2 Matching Hosts

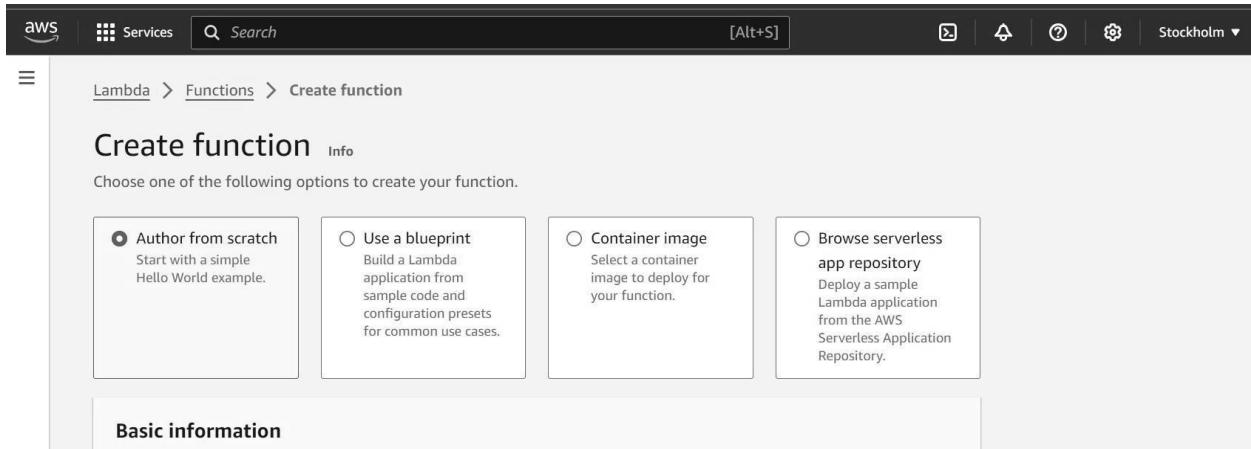
### Conclusion:

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

# EXPERIMENT-11

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

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



**Basic information**

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

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

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

**Architecture Info**  
Choose the instruction set architecture you want for your function code.  
 x86\_64  
 arm64

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

▶ Change default execution role

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

Code | Test | Monitor | Configuration | Aliases | Versions

**General configuration**

General configuration <small>Info</small>			
<small>Description</small> -	<small>Memory</small> 128 MB	<small>Ephemeral storage</small> 512 MB	<small>Edit</small>
<small>Timeout</small> 0 min 3 sec	<small>SnapStart Info</small> None		

The screenshot shows the AWS Lambda Functions overview for a function named 'lambdaexp11'. At the top, a success message states: "Successfully created the function lambdaexp11. You can now change its code and configuration. To invoke your function with a test event, choose 'Test'." Below this, the function name 'lambdaexp11' is displayed, along with buttons for 'Throttle', 'Copy ARN', and 'Actions'. A 'Function overview' section is expanded, showing a diagram with the function icon and the text 'Layers (0)'. Buttons for '+ Add trigger' and '+ Add destination' are present. To the right, there are fields for 'Description' (empty), 'Last modified' (16 seconds ago), and 'Function ARN' (arn:aws:lambda:eu-north-1:026090558619:function:lambdaexp11). Other tabs like 'Code', 'Test', 'Monitor', 'Configuration', 'Aliases', and 'Versions' are visible at the bottom.

Code source Info

File Edit Find View Go Tools Window Test Deploy

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 from Lambda!')
8     }
9
```

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

Lambda > Functions > lamdaexp11 > Edit basic settings

## Edit basic settings

**Basic settings** Info

Description - optional  
D15C

Memory Info  
Your function is allocated CPU proportional to the memory configured.  
128 MB  
Set memory to between 128 MB and 10240 MB.

Ephemeral storage Info  
You can configure up to 10 GB of ephemeral storage (/tmp) for your function. View pricing

512 MB  
Set ephemeral storage (/tmp) to between 512 MB and 10240 MB.

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

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

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

None

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

**Timeout**  
0 min 1 sec

**Execution role**  
Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console .

Use an existing role

Create a new role from AWS policy templates

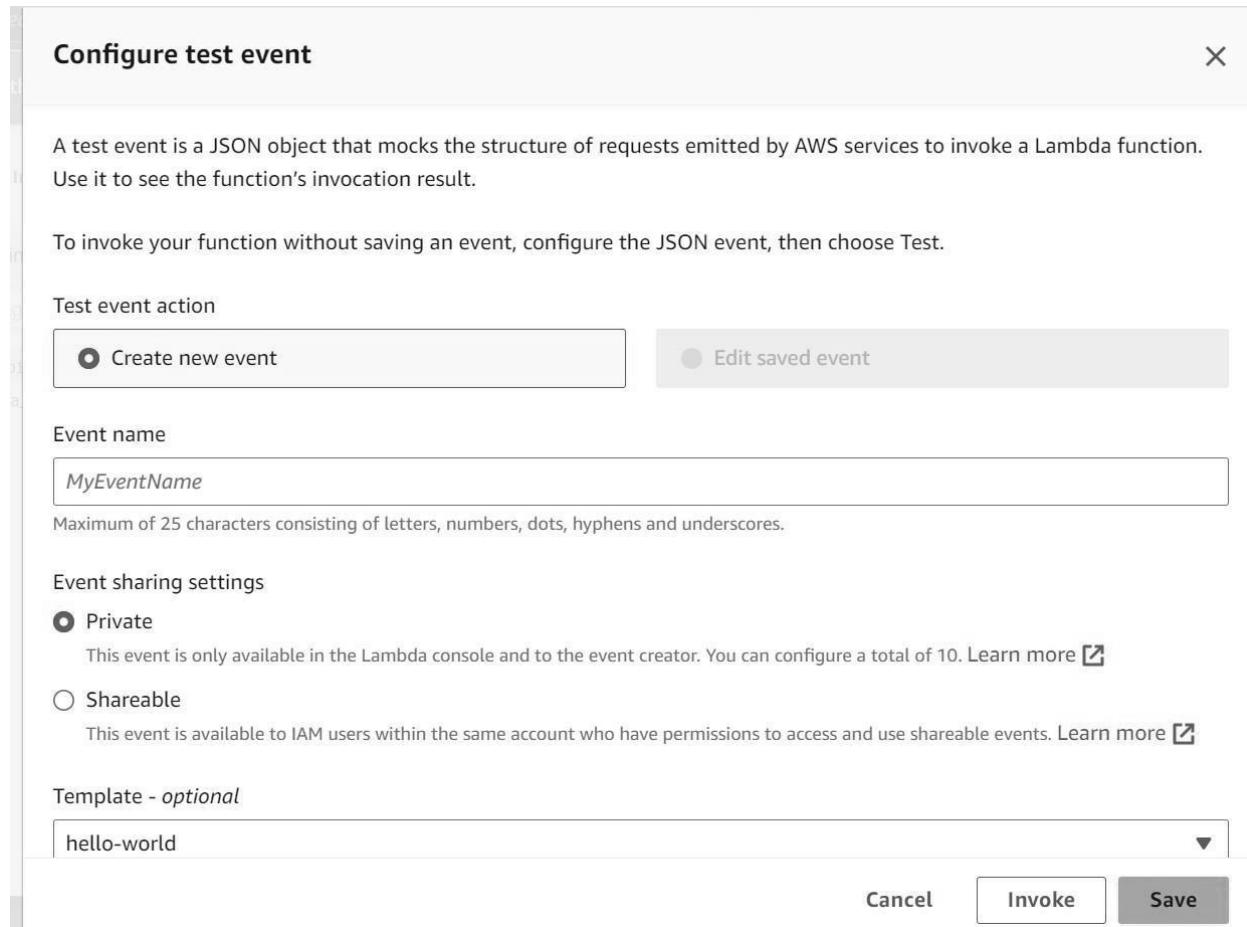
**Existing role**  
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

service-role/lamdaexp11-role-vj5j9g95

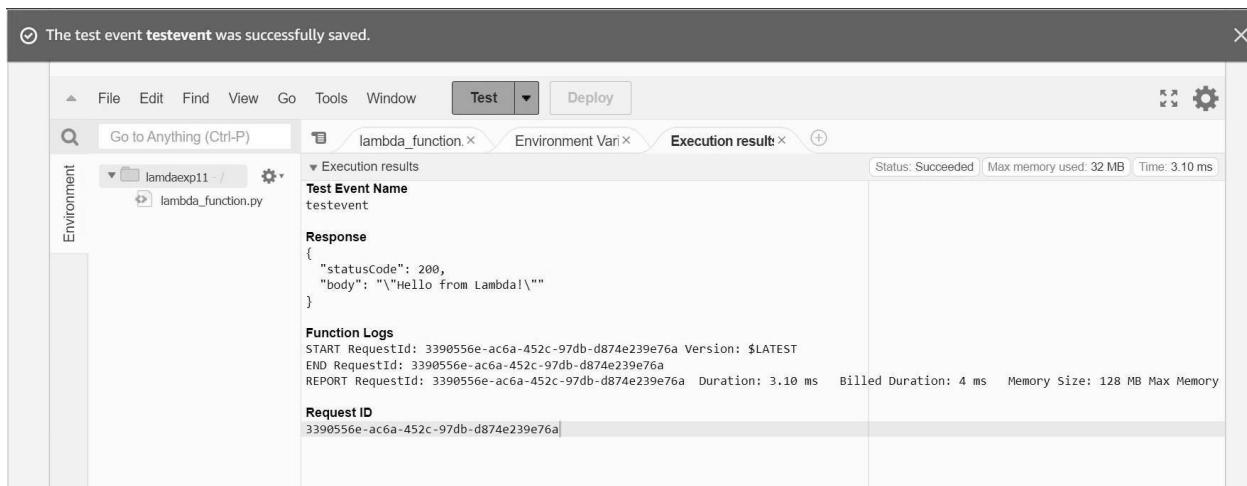
View the lamdaexp11-role-vj5j9g95 role on the IAM console.

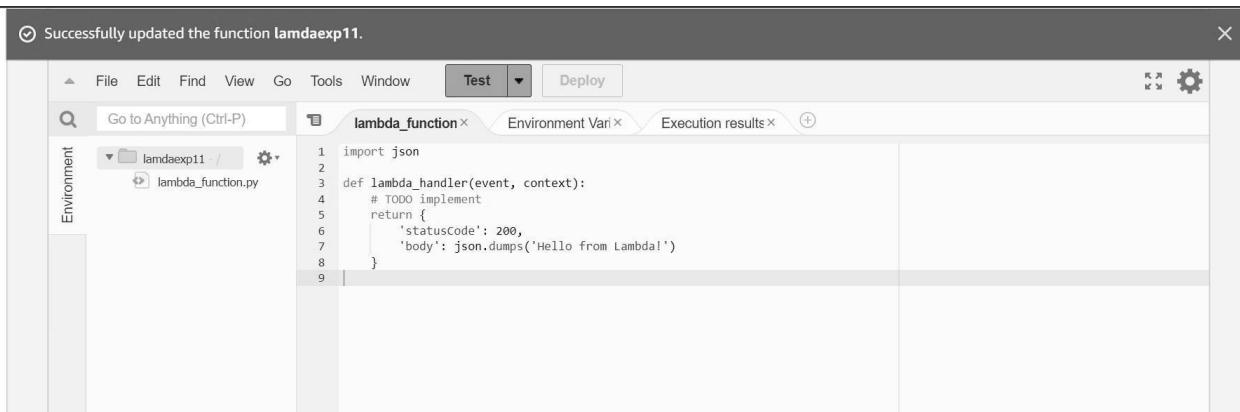
**Cancel** **Save**

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



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



**STEP 7: The function is successfully added .**

The screenshot shows the AWS Lambda console interface. At the top, a message says "Successfully updated the function lamdaexp11." Below the message, there are tabs for "Test" and "Deploy". The main area shows the "lambda\_function" tab, which contains the following Python code:

```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     return {
6         'statusCode': 200,
7         'body': json.dumps('Hello from Lambda!')
8     }
9
```

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

# EXPERIMENT 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 STEPS:

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

Amazon S3 > Buckets > Create bucket

**Create bucket** Info

Buckets are containers for data stored in S3.

**General configuration**

AWS Region  
Europe (Stockholm) eu-north-1

Bucket type Info

- General purpose
 

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

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

Bucket name Info

exp12d15c

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

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

**Block Public Access settings for this bucket**

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

**Block all public access**

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

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

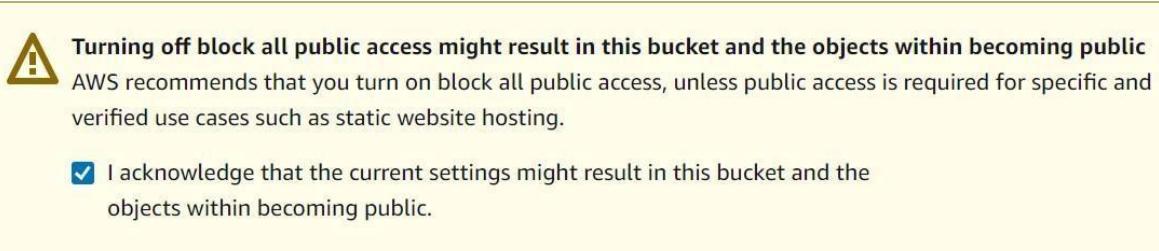
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets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

S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through *new* public bucket or access point policies**

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

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

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



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

The screenshot shows the AWS S3 console with a confirmation message: "Successfully created bucket 'exp12d15c'. To upload files and folders, or to configure additional bucket settings, choose View details." Below this, there's an "Account snapshot - updated every 24 hours" section and a "General purpose buckets" list.

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

The screenshot shows the AWS S3 "Objects (0)" page for the "exp12d15c" bucket. At the top, there's a toolbar with buttons for Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, and Upload. Below the toolbar, a message says: "Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions." A search bar labeled "Find objects by prefix" is followed by a pagination area with page 1 of 1. The main table header includes columns for Name, Type, Last modified, Size, and Storage class. A message at the bottom states: "No objects" and "You don't have any objects in this bucket." A prominent "Upload" button is located at the bottom center.

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

## Upload Info

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

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

<b>Files and folders (1 Total, 293.3 KB)</b>		<b>Remove</b>	<b>Add files</b>	<b>Add folder</b>
All files and folders in this table will be uploaded.				
<input type="text"/> <b>Find by name</b>		<span style="float: right;">&lt; 1 &gt;</span>		
<input type="checkbox"/>	<b>Name</b>	<b>Folder</b>	<b>Type</b>	
<input type="checkbox"/>	AppBar( title Text('Guidelines'), ),....	-	image/png	

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

⌚ Upload succeeded  
View details below.

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

---

<b>Files and folders</b>	<b>Configuration</b>												
<b>Files and folders (1 Total, 293.3 KB)</b>													
<input type="text"/> <b>Find by name</b>													
<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th><b>Name</b></th> <th><b>Folder</b></th> <th><b>Type</b></th> <th><b>Size</b></th> <th><b>Status</b></th> <th><b>Error</b></th> </tr> </thead> <tbody> <tr> <td>AppBar( title...</td> <td>-</td> <td>image/png</td> <td>293.3 KB</td> <td> Succeeded</td> <td>-</td> </tr> </tbody> </table>		<b>Name</b>	<b>Folder</b>	<b>Type</b>	<b>Size</b>	<b>Status</b>	<b>Error</b>	AppBar( title...	-	image/png	293.3 KB	 Succeeded	-
<b>Name</b>	<b>Folder</b>	<b>Type</b>	<b>Size</b>	<b>Status</b>	<b>Error</b>								
AppBar( title...	-	image/png	293.3 KB	 Succeeded	-								

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

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

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

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

The trigger exp12d15c was successfully added to function lambdaexp11. The function is now receiving events from the trigger.

**Function overview**

Description: D15C  
Last modified: 18 minutes ago  
Function ARN: arn:aws:lambda:eu-north-1:026090558619:function:lambdaexp11  
Function URL: -

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

Execution result: Status: Succeeded | Max memory used: 32 MB | Time: 1.97 ms

Test Event Name: testevent

Response:

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

Function Logs:

```
START RequestId: 6e57026c-6bfe-41fe-898d-2cdbe72fb1a3 Version: $LATEST
END RequestId: 6e57026c-6bfe-41fe-898d-2cdbe72fb1a3
REPORT RequestId: 6e57026c-6bfe-41fe-898d-2cdbe72fb1a3 Duration: 1.97 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory
```

Request ID: 6e57026c-6bfe-41fe-898d-2cdbe72fb1a3

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