

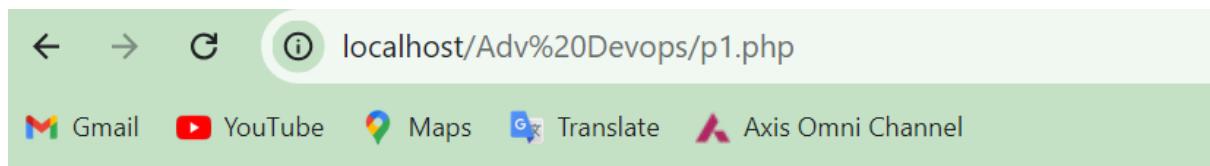
EXPERIMENT NO 1

NAME-SPANDAN DEB

CLASS-D15A

ROLL NO-13

```
FTP p1.php X
C: > xampp > htdocs > Adv Devops > FTP p1.php
1   <html>
2   <head>
3   <title>First PHP Program </title>
4   </head>
5   <body>
6   <?php
7   echo "MY NAME IS SPANDAN DEB";
8   ?>
```



MY NAME IS SPANDAN DEB

Create bucket Info

Buckets are containers for data stored in S3.

General configuration

AWS Region

Asia Pacific (Sydney) ap-southeast-2

Bucket name Info

myawsbucket

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

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

[Choose bucket](#)

Format: s3://bucket/prefix

Objects Properties Permissions Metrics Management Access Points

Objects (0) Info



[Copy S3 URI](#)

[Copy URL](#)

[Download](#)

[Open](#)

[Delete](#)

[Actions ▾](#)

[Create folder](#)

[Upload](#)

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. [Learn more](#)

[Find objects by prefix](#)

< 1 >

[Name](#) [Type](#) [Last modified](#) [Size](#) [Storage class](#)

No objects

You don't have any objects in this bucket.

[Upload](#)

☰ **Upload succeeded**
View details below.

Destination s3://spandan1	Succeeded 1 file, 300.0 B (100.00%)	Failed 0 files, 0 B (0%)
----------------------------------------------	----------------------------------------	-----------------------------

Files and folders Configuration

Files and folders (1 Total, 300.0 B)

Name	Folder	Type	Size	Status	Error
index.html	AdvD/	text/html	300.0 B	Succeeded	-

Object overview

Owner	s andan.deb04	S3 URI	s3://spandan1/AdvD/index.html
AWS Region	Asia Pacific (Sydney) ap-southeast-2	Amazon Resource Name (ARN)	arn:aws:s3:::spandan1/AdvD/index.html
Last modified	August 6, 2024, 23:20:15 (UTC+05:30)	Entity tag (Etag)	eab4f4741f8a8b61cefd3e889d2699e7
Size	300.0 B	Object URL	https://spandan1.s3.ap-southeast-2.amazonaws.com/AdvD/index.html
Type	html		
Key	AdvD/index.html		

Hi! My Name is Spandan Deb

Static website hosted on aws

EXPERIMENT NO 2

NAME-SPANDAN DEB

CLASS-D15A

ROLLNO-13

AIM-

To Build Your Application using AWS Code Build and Deploy on S3 / SEBS using AWS CodePipeline, deploy Sample Application on EC2 instance using AWS Code Deploy.

Trusted entity type

<input checked="" type="radio"/> AWS service Allow AWS services like EC2, Lambda, or others to perform actions in this account.	<input type="radio"/> AWS account Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.	<input type="radio"/> Web identity Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
<input type="radio"/> SAML 2.0 federation Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.	<input type="radio"/> Custom trust policy Create a custom trust policy to enable others to perform actions in this account.	

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

Service or use case

EC2 ▾

Choose a use case for the specified service.
Use case

EC2
Allows EC2 instances to call AWS services on your behalf.

EC2 Role for AWS Systems Manager

Trusted entity type

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

AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

Web identity
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

SAML 2.0 federation
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.

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

Service or use case
 CodeDeploy

Choose a use case for the specified service.
Use case

CodeDeploy
Allows CodeDeploy to call AWS services such as Auto Scaling on your behalf.

CodeDeploy for Lambda
Allows CodeDeploy to route traffic to a new version of an AWS Lambda function version on your behalf.

CodeDeploy - ECS
Allows CodeDeploy to read S3 objects, invoke Lambda functions, publish to SNS topics, and update ECS services on your behalf.

Cancel **Next**

Role details

Role name

Enter a meaningful name to identify this role.

CodeDeployRole

Maximum 64 characters. Use alphanumeric and '+=-,.@-_' characters.

Description

Add a short explanation for this role.

Allows CodeDeploy to call AWS services such as Auto Scaling on your behalf.

Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following char

Step 1: Select trusted entities

⌚ Role CodeDeployRole1 created.

[IAM](#) > [Roles](#) > [CodeDeployRole1](#)

CodeDeployRole1 [Info](#)

Allows CodeDeploy to call AWS services such as Auto Scaling on your behalf.

Summary

Creation date

August 11, 2024, 22:43 (UTC+05:30)

ARN

[arn:aws:iam::0109281845](#)

Last activity

-

Maximum session duration

1 hour

=

Launch an instance [Info](#)

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

Name and tags [Info](#)

Name Add additional tags

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

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

▼ Summary

Number of instances [Info](#)
1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.5.2...[read more](#)
ami-02346a771f34de8ac

Virtual server type (instance type)
t2.micro

Firewall (security group)
default

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

[Cancel](#) [Launch instance](#) [Review commands](#)

Create application

Application configuration

Application name

Enter an application name

100 character limit

Compute platform

Choose a compute platform



Tags

Create deployment group

Application

Application

SD-CICD

Compute type

EC2/On-premises

Deployment group name

Enter a deployment group name

100 character limit

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.

No more than 100 characters

Pipeline type

(i) 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.

Execution mode

Choose the execution mode for your pipeline. This determines how the pipeline is run.

Superseded

[Developer Tools](#) > [Connections](#) > [Create connection](#)

Connect to GitHub

GitHub connection settings Info

Connection name

GitHub Apps

GitHub Apps create a link for your connection with GitHub. Install a new app and save this connection.



or

[Install a new app](#)

► Tags - optional

[Connect](#)

Install AWS Connector for GitHub

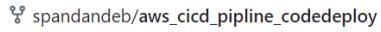
Install on your personal account spandandeb 

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).

 Selected 1 repository.

with these permissions:

 **Read access to issues and metadata**

Edit inbound rules Info

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

Security group rule ID	Type <small>Info</small>	Protocol <small>Info</small>	Port range	Source <small>Info</small>	Description - optional <small>Info</small>
<small>Info</small>					
sgr-0a518631208bbdfc1	All traffic	All	All	Custom	<input type="text"/> 
-	HTTP	TCP	80	Any...	<input type="text"/> 
-	SSH	TCP	22	Any...	<input type="text"/> 

SD-CICD-PIPELINE

Notify ▾ Edit Stop execution Clone pipeline Release change

Pipeline type: V2 Execution mode: QUEUED

Source Succeeded

Pipeline execution ID: [451062c7-8807-4c6f-b07e-2f079c47ec60](#)

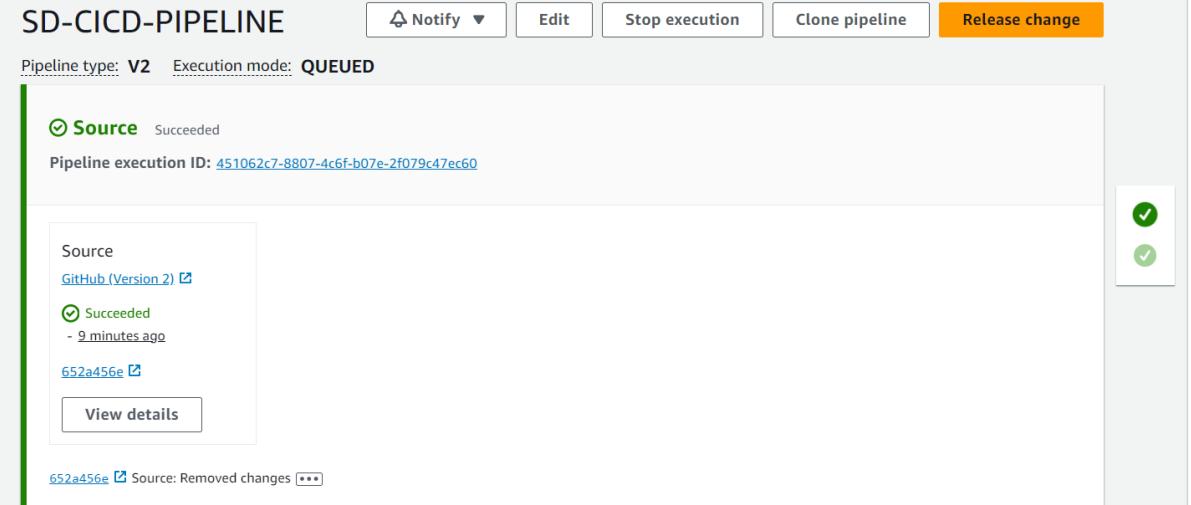
Source GitHub (Version 2)

Succeeded - 9 minutes ago

[652a456e](#)

[View details](#)

[652a456e](#) Source: Removed changes



C Not secure 13.55.205.25

YouTube Maps Translate Axis Omni Channel



Congratulations!!!

Welcome to AAR SOURCE| Ram Hemareddy

This application was deployed using AWS CodeDeploy.

For next steps, read the [AWS CodeDeploy Documentation](#).

EXPERIMENT NO 3

Create 3 EC2 instances

Name
master

Add additional tags

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

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

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

Recents Quick Start

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

▼ Summary

Number of instances | [Info](#)

1|

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd64...[read more](#)
ami-0e86e20dae9224db8

Virtual server type (instance type)
t2.medium

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 30 GiB

[Cancel](#) [Launch instance](#)

Instance state = running		Clear filters	<	1	>		
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
1	master	i-03de8ab0e6260426b	Running	t2.medium	Initializing	View alarms	us-east-1a
2	worker-1	i-0140746dec114251e	Running	t2.medium	Initializing	View alarms	us-east-1a
3	worker-2	i-04df968130e67b5f4	Running	t2.medium	Initializing	View alarms	us-east-1a

CONNECT TO INSTANCE Info

Connect to your instance i-03de8ab0e6260426b (master) using any of these options

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

Instance ID

[i-03de8ab0e6260426b \(master\)](#)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is `advd.pem`
3. Run this command, if necessary, to ensure your key is not publicly viewable.
 `chmod 400 "advd.pem"`
4. Connect to your instance using its Public DNS:
 `ec2-52-86-6-228.compute-1.amazonaws.com`

Example:

`ssh -i "advd.pem" ubuntu@ec2-52-86-6-228.compute-1.amazonaws.com`

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

Cancel

```

u@ip-172-31-31-226: ~
Permanently added 'ec2-54-82-231-139.
c2-54-82-231-139.compute-1.amazonaws.com' (RSA) to the list of known hosts.

TOP-9JIMM8I3 MINGW64 ~
nloads/

TOP-9JIMM8I3 MINGW64 ~/Downloads
$ ssh -i "advd.pem" ubuntu@ec2-54-82-231-139.c
to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1
entation: https://help.ubuntu.com
ement: https://landscape.canonical
rt: https://ubuntu.com/pro
information as of Sun Sep 15 07:27:53
load: 0.0 Processes:
of /: 5.4% of 28.02GB Users logged
usage: 5% IPv4 address
sage: 0% Expanded Security Maintenance for Applications
Security Maintenance for Applications 0 updates can be applied immediately.

ubuntu@ip-172-31-26-46: ~
HOME@LAPTOP-9JIMM8I3 MINGW64 ~
$ cd Downloads/
HOME@LAPTOP-9JIMM8I3 MINGW64 ~/Downloads
$ ssh -i "advd.pem" ubuntu@ec2-54-225-49-46
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.
* Documentation: https://help.ubuntu.co
* Management: https://landscape.canon
* Support: https://ubuntu.com/pro
System information as of Sun Sep 15 07:3
System load: 0.0 Process
Usage of /: 5.4% of 28.02GB Users 1
Memory usage: 5% IPV4 ad
Swap usage: 0%
Expanded Security Maintenance for Applications
0 updates can be applied immediately.

ubuntu@ip-172-31-23-85: ~
HOME@LAPTOP-9JIMM8I3 MINGW64 ~
$ cd Downloads/
HOME@LAPTOP-9JIMM8I3 MINGW64 ~/Downloads
$ ssh -i "advd.pem" ubuntu@ec2-52-86-6-228.compute-1.amazonaws.com
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

System information as of Sun Sep 15 07:33:16 UTC 2024
System load: 0.0 Processes: 114
Usage of /: 5.4% of 28.02GB Users logged in: 0
Memory usage: 5% IPv4 address for enX0: 172.31.23.85
Swap usage: 0%
Expanded Security Maintenance for Applications is not enabled.

```

Prepare nodes

Use sudo apt-get update && sudo apt-get upgrade -y to update the package manager

```
ubuntu@ip-172-31-31-226:~$ sudo apt-get update && sudo apt-get upgrade -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [351 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [77.3 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4416 B]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [267 kB]
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [111 kB]
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [8632 B]
```

Disable Swap

```
sudo swapoff -a
```

```
sudo sed -i '/ swap / s/^/#/' /etc/fstab
```

```
ubuntu@ip-172-31-31-226:~$ sudo swapoff -a
sudo sed -i '/ swap / s/^/#/' /etc/fstab
```

Load necessary kernel modules

```
cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
```

```
overlay
```

```
br_netfilter
```

```
EOF
```

```
sudo modprobe overlay
```

```
sudo modprobe br_netfilter
```

```
ubuntu@ip-172-31-23-85:~$ cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
overlay
br_netfilter
EOF
sudo modprobe overlay
sudo modprobe br_netfilter
overlay
br_netfilter
ubuntu@ip-172-31-23-85:~$ |
```

Configure systemctl settings for Kuber

```
cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf  
net.bridge.bridge-nf-call-ip6tables = 1  
net.bridge.bridge-nf-call-iptables = 1  
EOF
```

```
sudo sysctl -- system
```

```
ubuntu@ip-172-31-23-85:~$ cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf  
net.bridge.bridge-nf-call-ip6tables = 1  
net.bridge.bridge-nf-call-iptables = 1  
EOF  
sudo sysctl --system  
net.bridge.bridge-nf-call-ip6tables = 1  
net.bridge.bridge-nf-call-iptables = 1  
* Applying /usr/lib/sysctl.d/10-apparmor.conf ...  
* Applying /etc/sysctl.d/10-console-messages.conf ...  
* Applying /etc/sysctl.d/10-ipv6-privacy.conf ...  
* Applying /etc/sysctl.d/10-kernel-hardening.conf ...  
* Applying /etc/sysctl.d/10-magic-sysrq.conf ...  
* Applying /etc/sysctl.d/10-map-count.conf ...  
* Applying /etc/sysctl.d/10-network-security.conf ...  
* Applying /etc/sysctl.d/10-ptrace.conf ...  
* Applying /etc/sysctl.d/10-zero-page.conf ...  
* Applying /etc/sysctl.d/50-cloudimg-settings.conf ...  
* Applying /usr/lib/sysctl.d/50-pid-max.conf ...
```

Installing Docker

```
sudo apt-get update  
sudo apt-get install -y apt-transport-https ca-certificates  
curl software-properties-common curl -fsSL  
https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -  
sudo add-apt-repository "deb [arch=amd64]  
https://download.docker.com/linux/ubuntu  
$(lsb_release -cs) stable"  
sudo apt-get update  
sudo apt-get install -y docker-ce docker-ce-cli containerd.io
```

```

ubuntu@ip-172-31-23-85:~$ sudo apt-get update
sudo apt-get install -y apt-transport-https ca-certificates
curl software-properties-common curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable"
sudo apt-get update
sudo apt-get install -y docker-ce docker-ce-cli containerd.io
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Fetched 126 kB in 0s (358 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203).
ca-certificates set to manually installed.
The following NEW packages will be installed:
  apt-transport-https

```

Configure Docker For Kubernetes

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

```
{
  "exec-opts": ["native.cgroupdriver=systemd"], "log-driver": "json-file",
  "log-opt": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
```

EOF

sudo systemctl restart docker

```

ubuntu@ip-172-31-23-85:~$ cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"], "log-driver": "json-file",
  "log-opt": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
EOF
sudo systemctl restart docker

{
  "exec-opts": ["native.cgroupdriver=systemd"], "log-driver": "json-file",
  "log-opt": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
ubuntu@ip-172-31-23-85:~$ |

```

Add Kubernetes APT repository

```
sudo curl -fsSL /usr/share/keyrings/kubernetes-archive-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg
echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg]
https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee
/etc/apt/sources.list.d/kubernetes.list
```

```
ubuntu@ip-172-31-27-12:~$ curl -fsSL https://pkgs.k8s.io/core:/stable:v1.29/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-archive-keyring.gpg
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-archive-keyring.gpg] https://pkgs.k8s.io/core:/stable:v1.29/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list
sudo apt-get update -y
sudo apt-get install -y kubelet="1.29.0-*" kubectl="1.29.0-*" kubeadm="1.29.0-*"
sudo apt-get update -y
sudo apt-get install -y jq
deb [signed-by=/etc/apt/keyrings/kubernetes-archive-keyring.gpg] https://pkgs.k8s.io/core:/stable:v1.29/deb/ /
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:5 https://download.docker.com/linux/ubuntu jammy InRelease
Hit:6 https://prod-cdn.packages.k8s.io/repositories/isy:/kubernetes:/addons:/cri-o:/prerelease:/main/deb InRelease
Get:7 https://prod-cdn.packages.k8s.io/repositories/isy:/kubernetes:/core:/stable:v1.29/deb InRelease [1189 B]
Get:8 https://prod-cdn.packages.k8s.io/repositories/isy:/kubernetes:/core:/stable:v1.29/deb Packages [14.0 kB]
Fetched 15.1 kB in 1s (20.2 kB/s)
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/jammy/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the
y(8) for details.
Reading package lists... Done
Building dependency tree... Done
```

Install kubelet, kubeadm, and kubectl:

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

```
ubuntu@ip-172-31-27-12:~$ sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  ebttables socat
Use 'sudo apt autoremove' to remove them.
The following packages will be upgraded:
  kubeadm kubectl kubelet
3 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
Need to get 40.7 MB of archives.
After this operation, 1758 kB of additional disk space will be used.
Get:1 https://prod-cdn.packages.k8s.io/repositories/isy:/kubernetes:/core:/stable:v1
```

Initialize the Kubernetes cluster on master node

```
sudo kubeadm init --pod-network-cidr=10.244.0.0/16
sudo kubeadm init --cri-socket unix:///var/run/crio/crio.sock
```

```
ubuntu@ip-172-31-27-12:~$ nano kubeadm-config.yaml
ubuntu@ip-172-31-27-12:~$ sudo kubeadm init --config kubeadm-config.yaml
W0915 10:18:14.121956    23054 initconfiguration.go:312] error unmarshaling configuration schema.GroupVersionKind
configuration": strict decoding error: unknown field "criSocket"
Found multiple CRI endpoints on the host. Please define which one do you wish to use by setting the 'criSocket' field in your configuration file.
To see the stack trace of this error execute with --v=5 or higher
ubuntu@ip-172-31-27-12:~$ sudo kubeadm init --cri-socket unix:///var/run/crio/crio.sock
I0915 10:19:09.235316    23090 version.go:256] remote version is much newer: v1.31.0; falling back to: stable
[init] Using Kubernetes version: v1.29.8
[preflight] Running pre-flight checks
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
```

To setup kubectl on master node

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

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

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

```
ubuntu@ip-172-31-27-12:~$ 
  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

Install pod network add-on

```
kubectl apply -f
```

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

```
ubuntu@ip-172-31-27-12:~$ 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
ubuntu@ip-172-31-27-12: ~ % client lost send disconnect: Connection reset by peer
```

Join Worker Nodes

```
sudo kubeadm join 172.31.27.12:6443 --token zerrkd.zyzasxm2e69grelo \
--discovery-token-ca-cert-hash
sha256:c7e56b4e0a3eaf45f6c55c19e338a0e68ad04a2cab623808ddebcb74e7b
bcfd \
--cri-socket unix:///var/run/crio/crio.sock --v=5
```

```
ubuntu@ip-172-31-30-123:~$ sudo kubeadm join 172.31.27.12:6443 --token zerrkd.zyzasxm2e69grelo \
--discovery-token-ca-cert-hash sha256:c7e56b4e0a3eaf45f6c55c19e338a0e68ad04a2cab623808ddebcb74e7bbced \
--cri-socket unix:///var/run/crio/crio.sock --v=5
I0915 10:57:30.867146 17284 join.go:413] [preflight] found NodeName empty; using OS hostname as NodeName
[preflight] Running pre-flight checks
I0915 10:57:30.867225 17284 preflight.go:93] [preflight] Running general checks
I0915 10:57:30.867256 17284 checks.go:280] validating the existence of file /etc/kubernetes/kubelet.conf
I0915 10:57:30.867262 17284 checks.go:280] validating the existence of file /etc/kubernetes/bootstrap-kubelet.conf
I0915 10:57:30.867269 17284 checks.go:104] validating the container runtime
I0915 10:57:30.885423 17284 checks.go:639] validating whether swap is enabled or not
I0915 10:57:30.885480 17284 checks.go:370] validating the presence of executable crictl
I0915 10:57:30.885597 17284 checks.go:370] validating the presence of executable conntrack
I0915 10:57:30.885622 17284 checks.go:370] validating the presence of executable ip
I0915 10:57:30.886766 17284 checks.go:370] validating the presence of executable iptables
I0915 10:57:30.886801 17284 checks.go:370] validating the presence of executable mount
I0915 10:57:30.886900 17284 checks.go:370] validating the presence of executable nsenter
I0915 10:57:30.886925 17284 checks.go:370] validating the presence of executable ebtables
I0915 10:57:30.886973 17284 checks.go:370] validating the presence of executable ethtool
I0915 10:57:30.887017 17284 checks.go:370] validating the presence of executable socat
I0915 10:57:30.887117 17284 checks.go:370] validating the presence of executable tc
I0915 10:57:30.887131 17284 checks.go:370] validating the presence of executable touch
I0915 10:57:30.887147 17284 checks.go:516] running all checks
I0915 10:57:30.887156 17284 checks.go:401] checking whether the given node name is valid and reachable using net.LookupLocal
```

Verify the cluster

```
ubuntu@ip-172-31-27-12:~$ kubectl get nodes
NAME           STATUS    ROLES      AGE     VERSION
ip-172-31-27-12   Ready    control-plane   38m    v1.29.9
ip-172-31-27-123  Ready    <none>        59s    v1.29.9
ip-172-31-30-123  Ready    <none>        45s    v1.29.9
ubuntu@ip-172-31-27-12:~$ |
```

EXPERIMENT NO 4

NAME-SPANDAN DEB

CLASS-D15A

ROLL NO-13

Running the application on cluster

```
kubectl create deployment nginx --image=nginx
```

```
ubuntu@ip-172-31-27-12:~$ kubectl create deployment nginx --image=nginx
deployment.apps/nginx created
ubuntu@ip-172-31-27-12:~$ |
```

Verify the deployments

```
kubectl get deployments
```

```
ubuntu@ip-172-31-27-12:~$ kubectl get deployments
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
nginx         0/1     1            0           89s
nginx-deployment 0/2     2            0           29h
ubuntu@ip-172-31-27-12:~$ |
```

Run the following command to create a service named nginx that will expose the app publicly.

```
kubectl expose deploy nginx --port 80 --target-port 80 --type NodePort
```

```
ubuntu@ip-172-31-27-12:~$ kubectl expose deploy nginx --port 80 --target-port 80 --type NodePort
service/nginx exposed
ubuntu@ip-172-31-27-12:~$ |
```

See the summary of services and port exposed

Kubectl get services

```
ubuntu@ip-172-31-27-12:~$ kubectl get services
NAME         TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)      AGE
kubernetes   ClusterIP 10.96.0.1    <none>        443/TCP     32h
nginx        NodePort   10.111.98.122 <none>        80:30146/TCP 52s
nginx-service LoadBalancer 10.107.160.54  <pending>    80:31024/TCP 29h
ubuntu@ip-172-31-27-12:~$ |
```

Add the port which is displayed i.e 31024 in the inbound rules of the security group

Inbound rules (6)								Manage tags	Edit inbound rules
	Name	Security group rule...	IP version	Type	Protocol	Port range			
]	-	sgr-0c60cb6e6d9413cc5	IPv4	All ICMP - IPv4	ICMP	All			
]	-	sgr-00f14d1398788277f	IPv4	Custom TCP	TCP	6443			
]	-	sgr-086554458ae3f0b5e	IPv4	Custom TCP	TCP	10250			
]	-	sgr-08af0fc1d5ed94236	IPv4	Custom TCP	TCP	10255			
]	-	sgr-0ee6a4fa2123f3aee	IPv4	SSH	TCP	22			
]	-	sgr-0a5f2641855bc7407	IPv4	Custom TCP	TCP	31024			

Verify that nginx page is accessible on all nodes using curl command

Sudo su

Curl worker:31024

```

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee
/etc/apt/trusted.gpg.d/docker.gpg > /dev/null
sudo add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable"

```

```

last login: Mon Sep 10 19:00:51 2024 from 12.111.100.77
ubuntu@ip-172-31-27-12:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee
/etc/apt/trusted.gpg.d/docker.gpg > /dev/null
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable"
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
-----BEGIN PGP PUBLIC KEY BLOCK-----
mQINBFit2ioBEAdhWpZ8/wvZ6hUTiXowQHXMAlaFHcPH9hAtr4F1y2+OYdbtMuth
lqqwp028AqyY+PRfVmSYMbjuQuu5byyKR01BbqYhus3jtqQmljz/bJvXqnmiVXh
38Uula+z077PxxyQhu5BbqntTPQMfiyqEiU+BKbq2WmANUKQf+1AmZY/IruoXbnq
L4C1+gJ8vfmXQt99npCaxEjaNRVYf0s8cixNzHUyb6emj1ANYEv1Zzeqo7XK17
UrWV5inawTSzWNvtjEjj4nJL8NsLwscpLPQUhTQ+7BbQXAwAmeHCUTQIVvvwXqwON
cmh4hgeQscQHygOjjDVfoY5MucvglbigCqfqzAHw9jxmRL4gbMZj+b1xoeFeth
ku4bIQN1X5P07fFNwzIgaRL5Z4POXDZT1Q/E158j9kp4bnWRCJW01ya+f8ocodo
VZZ+Doi+fy4D5ZGrL4XEcIQP/Lv5uFyf+kqt1/94VFYV30leAv8w92KdqDkhTcTD

```

```

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

```

```

ubuntu@ip-172-31-27-12:~$ sudo apt-get update
sudo apt-get install -y docker-ce
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:5 https://download.docker.com/linux/ubuntu jammy InRelease
Hit:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/addons:/cri-o:/prerelease:/main/deb InRelease
Ign:7 https://packages.cloud.google.com/apt kubernetes-focal InRelease
Err:8 https://packages.cloud.google.com/apt kubernetes-focal Release
  404  Not Found [IP: 172.253.122.100 443]
Reading package lists... Done
E: The repository 'https://apt.kubernetes.io kubernetes-focal Release' does not have a Release file.

```

```

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

```

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
ubuntu@ip-172-31-27-12:~$ sudo mkdir -p /etc/docker  
cat <<EOF | sudo tee /etc/docker/daemon.json  
{  
  "exec-opts": ["native.cgroupdriver=systemd"]  
}  
EOF  
{  
  "exec-opts": ["native.cgroupdriver=systemd"]  
}
```

```
sudo systemctl enable docker
```

```
sudo systemctl daemon-reload
```

```
sudo systemctl restart docker
```

```
ubuntu@ip-172-31-27-12:~$ sudo systemctl enable docker  
sudo systemctl daemon-reload  
sudo systemctl restart docker  
Synchronizing state of docker.service with SysV service script with /lib/systemd/systemd-sysv-install.  
Executing: /lib/systemd/systemd-sysv-install enable docker  
+ /etc/init.d/docker enable
```

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

```
/etc/apt/keyrings/kubernetes-apt-keyring.gpg
```

```
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
```

```
https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /' | sudo tee  
/etc/apt/sources.list.d/kubernetes.list
```

```
ubuntu@ip-172-31-27-12:~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o  
/etc/apt/keyrings/kubernetes-apt-keyring.gpg  
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]  
https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list  
missing argument for option "-o"  
curl: (23) Failed writing body  
-bash: /etc/apt/keyrings/kubernetes-apt-keyring.gpg: Permission denied  
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]  
https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /  
ubuntu@ip-172-31-27-12:~$ |
```

```
sudo apt-get update
```

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

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

```
ubuntu@ip-172-31-27-12:~$ sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 https://download.docker.com/linux/ubuntu jammy InRelease
Hit:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/addons:/cri-o:/prerelease:/
Get:7 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Ign:6 https://packages.cloud.google.com/apt kubernetes-xenial InRelease
Err:8 https://packages.cloud.google.com/apt kubernetes-xenial Release
  404  Not Found [IP: 172.253.63.102 443]
Reading package lists... Done
E: The repository 'https://apt.kubernetes.io/kubernetes-xenial Release' does not have a Release
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
W: https://download.docker.com/linux/ubuntu/dists/jammy/InRelease: Key is stored in legacy trust
y(8) for details.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
kubeadm is already the newest version (1.29.9-1.1).
kubectl is already the newest version (1.29.9-1.1).
kubelet is already the newest version (1.29.9-1.1).
The following packages were automatically installed and are no longer required:
  ebttables socat
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 24 not upgraded.
kubelet was already set on hold.
kubeadm was already set on hold.
kubectl was already set on hold.
```

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

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

```
ubuntu@ip-172-31-27-12:~$ sudo systemctl enable --now kubelet
sudo kubeadm init --pod-network-cidr=10.244.0.0/16
Found multiple CRI endpoints on the host. Please define which one do you wish to use by setting the 'criSocket' field in the kubeadm config
ur
ainerd/containerd.sock, unix:///var/run/crio/crio.sock
To see the stack trace of this error execute with --v=5 or higher
```

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

```
ubuntu@ip-172-31-27-12:~$ sudo apt-get install -y containerd
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-ce-rootless-extras ebttables libltdl7 libslirp0 pigz slirp4netns socat
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  runc
The following packages will be REMOVED:
  containerd.io docker-ce
The following NEW packages will be installed:
  containerd runc
0 upgraded, 2 newly installed, 2 to remove and 24 not upgraded.
Need to get 46.2 MB of archives.
After this operation, 56.8 MB disk space will be freed.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 runc amd64 1.1.12-
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 containerd amd64 1
Fetched 46.2 MB in 1s (69.4 MB/s)
(Reading database ... 96015 files and directories currently installed.)
Removing docker-ce (5:27.3.0-1~ubuntu.22.04~jammy) ...
Removing containerd.io (1.7.22-1) ...
Selecting previously unselected package runc.
```

```
sudo mkdir -p /etc/containerd
```

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

```
ubuntu@ip-172-31-27-12:~$ sudo mkdir -p /etc/containerd
sudo containerd config default | sudo tee /etc/containerd/config.toml
disabled_plugins = []
imports = []
oom_score = 0
plugin_dir = ""
required_plugins = []
root = "/var/lib/containerd"
state = "/run/containerd"
temp = ""
version = 2

[cgroup]
  path = ""

[debug]
  address = ""
  format = ""
  gid = 0
  level = ""
  uid = 0

[grpc]
  address = "/run/containerd/containerd.sock"
  gid = 0
  max_recv_message_size = 16777216
  max_send_message_size = 16777216
  tcp_address = ""
  tcp_tls_ca = ""
  tcp_tls_cert = ""
  tcp_tls_key = ""
  uid = 0
```

```
sudo systemctl restart containerd
```

```
sudo systemctl enable containerd
```

```
sudo systemctl status containerd
```

```
ubuntu@ip-172-31-27-12:~$ sudo systemctl restart containerd
sudo systemctl enable containerd
sudo systemctl status containerd
● containerd.service - containerd container runtime
   Loaded: loaded (/lib/systemd/system/containerd.service; enabled; vendor pr
   Active: active (running) since Fri 2024-09-20 16:00:05 UTC; 307ms ago
     Docs: https://containerd.io
   Main PID: 10516 (containerd)
      Tasks: 8
     Memory: 13.4M
        CPU: 73ms
      CGroup: /system.slice/containerd.service
              └─10516 /usr/bin/containerd

Sep 20 16:00:05 ip-172-31-27-12 containerd[10516]: time="2024-09-20T16:00:05.84
Sep 20 16:00:05 ip-172-31-27-12 systemd[1]: Started containerd container runtim
Sep 20 16:00:05 ip-172-31-27-12 containerd[10516]: time="2024-09-20T16:00:05.85
ubuntu@ip-172-31-27-12:~$ |
```

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

```
ubuntu@ip-172-31-27-12:~$ sudo apt-get install -y socat
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
socat is already the newest version (1.7.4.1-3ubuntu4).
socat set to manually installed.
The following packages were automatically installed and are no longer required:
  docker-ce-rootless-extras ebttables libltdl7 libslirp0 pigz slirp4netns
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 24 not upgraded.
```

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

```

ubuntu@ip-172-31-27-12:~$ sudo kubeadm init --config=kubeadm-config.yaml
[0920 16:15:56.108107    12904 version.go:256] remote version is much newer: v1.31.0; falling back to: s
[init] Using Kubernetes version: v1.29.9
[preflight] Running pre-flight checks
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
[0920 16:16:06.427760    12904 checks.go:835] detected that the sandbox image "registry.k8s.io/pause:3.8"
It is recommended that using "registry.k8s.io/pause:3.9" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-27-12 kubernetes kubernetes.default k
10.96.0.1 172.31.27.12]
[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-172-31-27-12 localhost] and IPs [172.31.27
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-27-12 localhost] and IPs [172.31.27.1
[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"

```

```

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 172.31.27.12:6443 --token w6zqha.8nzpwonpcdt32np \
    --discovery-token-ca-cert-hash sha256:95e6d5a4f6746b63757957a765fec6f3d41c3d99b6964aeae0de2103e5c16fd9
ubuntu@ip-172-31-27-12:~$ |

```

`mkdir -p $HOME/.kube`

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

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

```

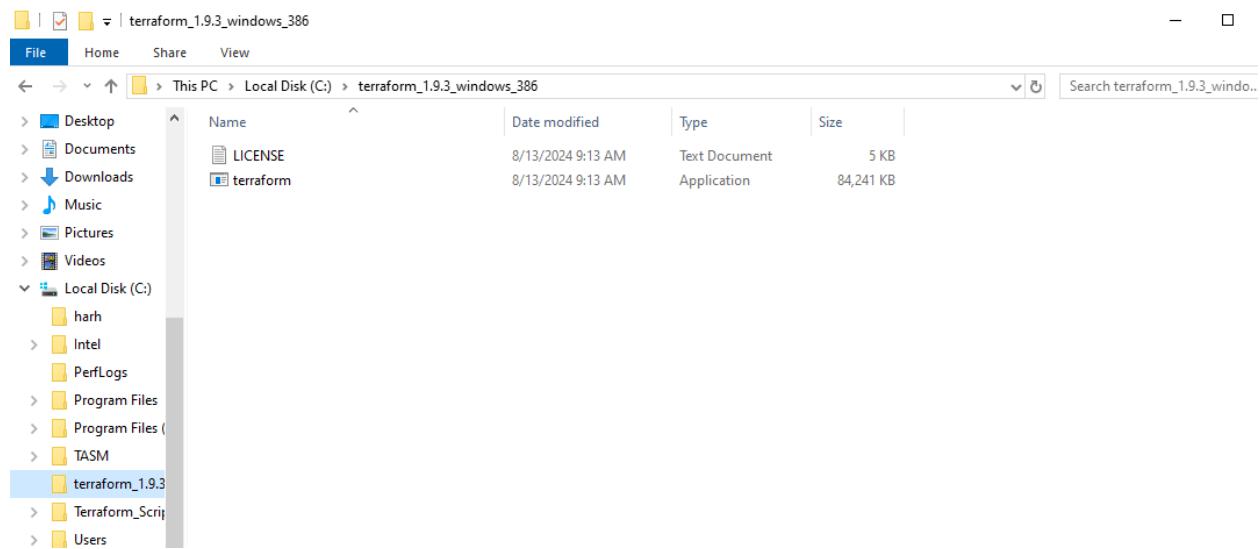
ubuntu@ip-172-31-27-12:~$ mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
cp: overwrite '/home/ubuntu/.kube/config'? y
ubuntu@ip-172-31-27-12:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@ip-172-31-27-12:~$ 

```


EXPERIMENT NO 5

NAME-SPANDAN DEB
CLASS-D15A
ROLL NO -13

The screenshot shows the Terraform website's download page. On the left, a sidebar lists operating systems: macOS, Windows (selected), Linux, FreeBSD, OpenBSD, Solaris, and Release information. Below this is a 'Next steps' section. The main content area is divided into 'Windows' and 'Linux' sections. The Windows section has 'Binary download' for 386 and AMD64 architectures, each with a 'Download' button. The Linux section has 'Package manager' tabs for Ubuntu/Debian, CentOS/RHEL, Fedora, Amazon Linux, and Homebrew. A terminal window shows the command to install Terraform via apt. A sidebar on the right contains links for 'About Terraform', 'Featured docs' (Introduction to Terraform, Configuration Language, Terraform CLI, HCP Terraform, Provider Use), and an 'HCP Terraform' section.



Environment Variables

X

User variables for Student

Variable	Value
OneDrive	C:\Users\Student.VESIT512-22\OneDrive
PATH	C:\terraform_1.9.3_windows_386
TEMP	C:\Users\Student.VESIT512-22\AppData\Local\Temp
TMP	C:\Users\Student.VESIT512-22\AppData\Local\Temp

New...

Edit...

Delete

System variables

Variable	Value
ComSpec	C:\WINDOWS\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
Path	C:\Program Files\Common Files\Oracle\Java\javapath;C:\Program ...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64

New...

Edit...

Delete

OK

Cancel

Allow local PowerShell scripts to run

```
PS C:\Users\Student.VESIT512-22>
PS C:\Users\Student.VESIT512-22> 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:\Users\Student.VESIT512-22>
```

EXPERIMENT NO 6

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

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

Management Commands:
  builder        Manage builds
  buildx*        Docker Buildx
  checkpoint    Manage checkpoints
  compose*       Docker Compose
  container     Manage containers
  context        Manage contexts
  debug*         Get a shell into any image or container
  desktop*      Docker Desktop commands (Alpha)
  dev*          Docker Dev Environments
  extension*    Manages Docker extensions
  feedback*     Provide feedback, right in your terminal!
  image          Manage images
  init*          Creates Docker-related starter files for your project
  manifest       Manage Docker image manifests and manifest lists
  network        Manage networks
  plugin         Manage plugins
  ...

  * A command with a star is experimental and subject to change.
```

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

The screenshot shows the Visual Studio Code interface. On the left, the 'EXPLORER' sidebar is open, displaying a tree view of a 'TERRAFORM' directory. Under 'Docker', there are files: '.terraform', '.terraform.lock.hcl', 'docker.tf' (which is selected), '{} terraform.tfstate', and 'terraform.tfstate.bac...'. In the main area, a code editor tab titled 'Welcome' is open, showing the contents of 'docker.tf'. The code defines a Terraform configuration for a Docker container named 'foo':

```
Docker > docker.tf > resource "docker_container" "foo"
1  terraform {
2
3    required_providers {
4      docker = {
5        source = "kreuzwerker/docker"
6        version = "2.21.0"
7      }
8    }
9  }
10 provider "docker" {
11   host = "npipe://./pipe/docker_engine"
12 }
13 # Pulls the image
14 resource "docker_image" "ubuntu"{
15   name = "ubuntu:latest"
16 }
17 # Create a container
18 resource "docker_container" "foo"{
19   image = docker_image.ubuntu.image_id
20   name = "foo"
21   tty = true
22 }
```

```
PS C:\Users\HOME\OneDrive\Desktop\Sem Files\sem 5\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.
```

```
PS C:\Users\HOME\OneDrive\Desktop\Sem Files\sem 5\Terraform Scripts\Docker> terraform plan

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

Terraform will perform the following actions:

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

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

    + healthcheck (known after apply)

    + labels (known after apply)
}

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

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

```
PS C:\Users\HOME\OneDrive\Desktop\Sem Files\sem 5\Terraform Scripts\Docker> terraform apply
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubur

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

Terraform will perform the following actions:

# docker_container.foo will be created
+ resource "docker_container" "foo" {
  + attach          = false
  + bridge          = (known after apply)
  + command         = (known after apply)
  + container_logs = (known after apply)
  + entrypoint      = (known after apply)
  + env              = (known after apply)
  + exit_code        = (known after apply)
  + gateway          = (known after apply)
  + hostname         = (known after apply)
  + id               = (known after apply)
  + image             = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a"
  + 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"

  + labels (known after apply)
}

Plan: 1 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_container.foo: Creating...

Enter a value: yes

docker_container.foo: Creating...
  Enter a value: yes

docker_container.foo: Creating...

docker_container.foo: Creating...
docker_container.foo: Creating...
docker_container.foo: Creation complete after 1s [id=49376784d30a5434da697a13c16793bd7e7b89cae005d54167cedfe80cb03e09]

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

```
PS C:\Users\HOME\OneDrive\Desktop\Sem Files\sem 5\Terraform Scripts\Docker> terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c]
docker_container.foo: Refreshing state... [id=49376784d30a5434da697a13c16793bd7e7b89cae005d54167cedfe80cb03e09]

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

Terraform will perform the following actions:

# docker_container.foo will be destroyed
- resource "docker_container" "foo" {
    - attach          = false -> null
    - command        = [
        - "/bin/bash",
    ] -> null
    - cpu_shares     = 0 -> null
    - dns            = [] -> null
    - dns_opts       = [] -> null
    - dns_search     = [] -> null
    - entrypoint     = [] -> null
    - env            = [] -> null
    - gateway        = "172.17.0.1" -> null
    - group_add      = [] -> null
}
```

```
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=49376784d30a5434da697a13c16793bd7e7b89cae005d54167cedfe80cb03e09]
docker_container.foo: Destruction complete after 0s
docker_image.ubuntu: Destroying... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c]
docker_image.ubuntu: Destruction complete after 1s
```

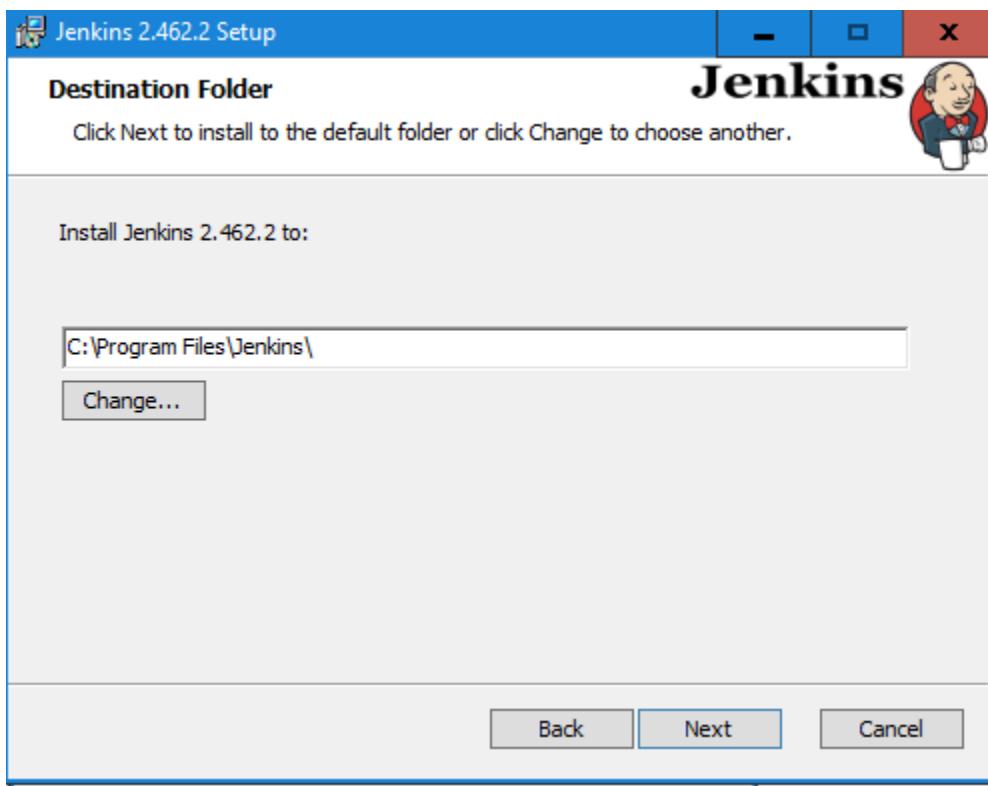
```
Destroy complete! Resources: 2 destroyed.
```

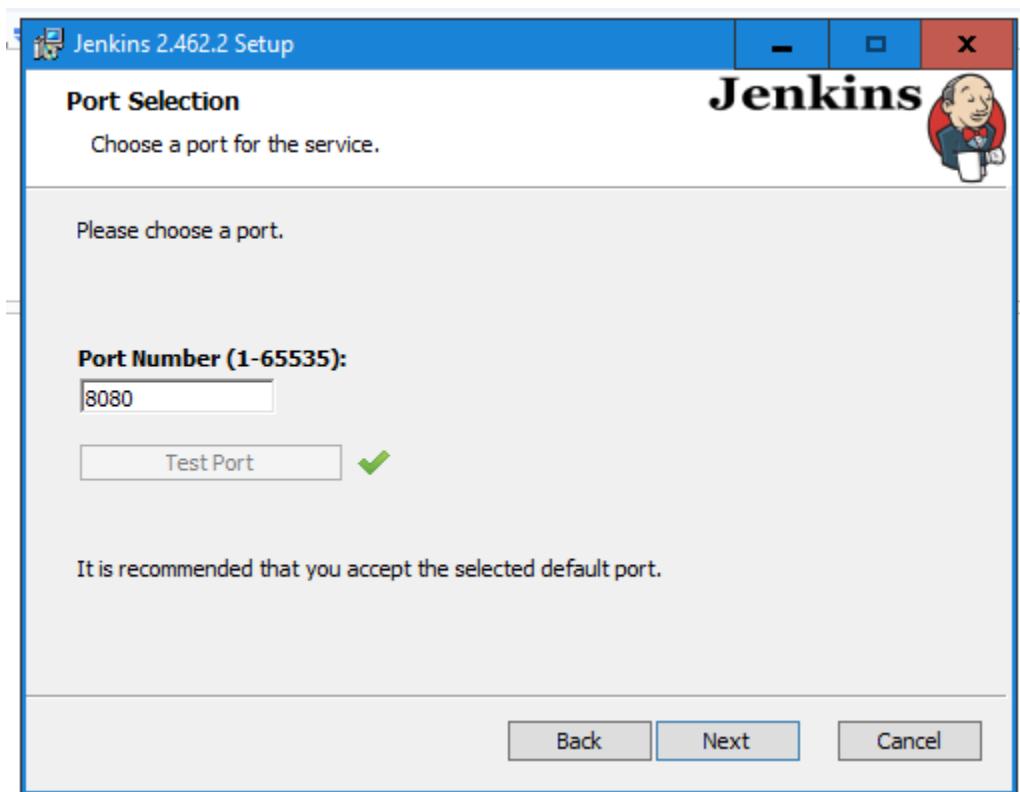
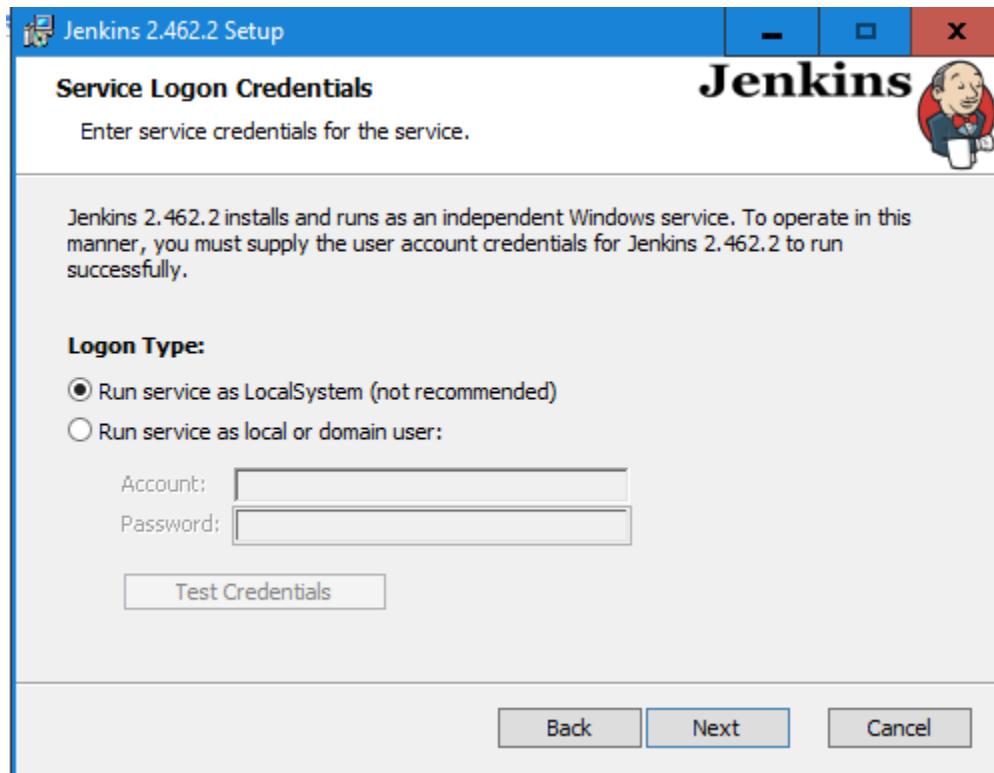
EXPERIMENT NO 7

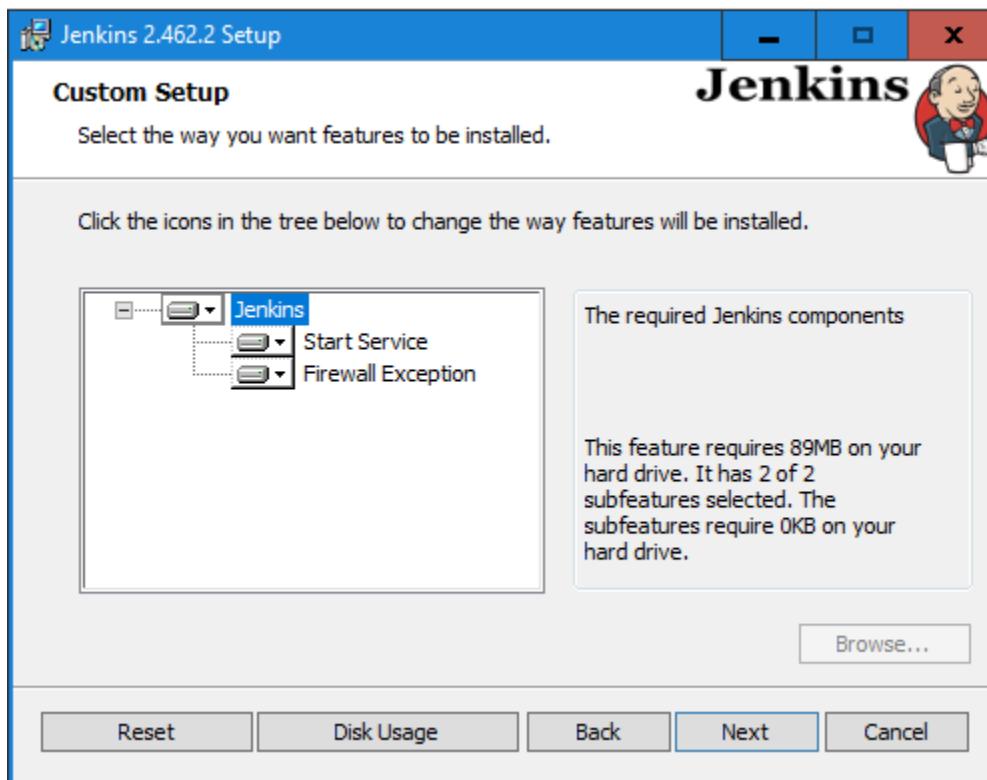
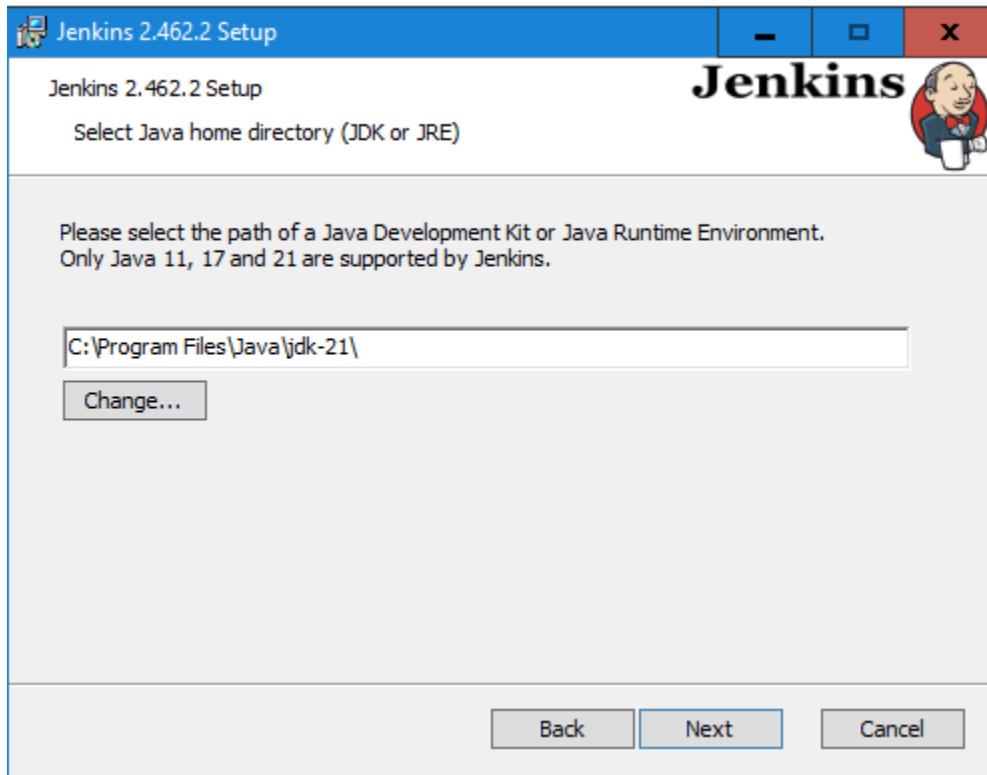
Aim-To understand Static analysis SAST process and learn to integrate jenkins SAST to SonarQube/GitLab.

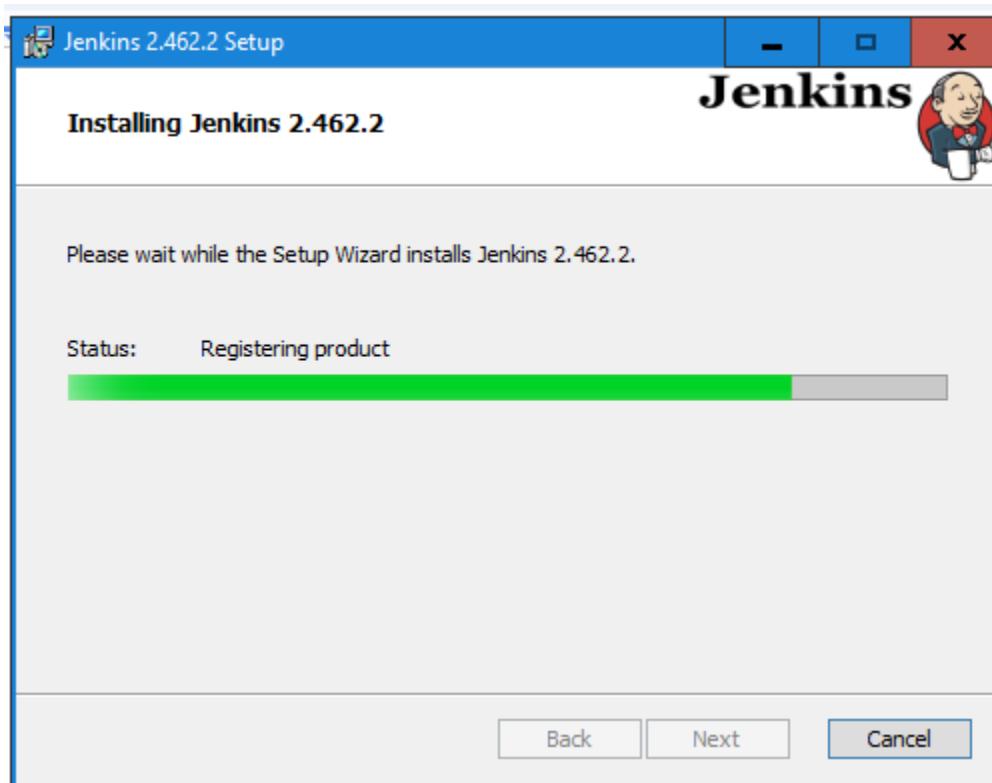
Jenkins Installation











Getting Started

Getting Started

✓ Folders	✓ OWASP Markup Formatter	⌚ Build Timeout	⌚ Credentials Binding	** Ionicons API Folders OWASP Markup Formatter ** ASM API ** JSON Path API
⌚ Timestamper	⌚ Workspace Cleanup	⌚ Ant	⌚ Gradle	
⌚ Pipeline	⌚ GitHub Branch Source	⌚ Pipeline: GitHub Groovy Libraries	⌚ Pipeline Graph View	
⌚ Git	⌚ SSH Build Agents	⌚ Matrix Authorization Strategy	⌚ PAM Authentication	
⌚ LDAP	⌚ Email Extension	⌚ Mailer	⌚ Dark Theme	

** - required dependency

Jenkins 2.462.2

 Jenkins

Dashboard >

+ New Item Add description

Build History Manage Jenkins My Views

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Build Queue Create a job

No builds in the queue.

Build Executor Status Set up a distributed build

1 Idle 2 Idle

Set up an agent Configure a cloud

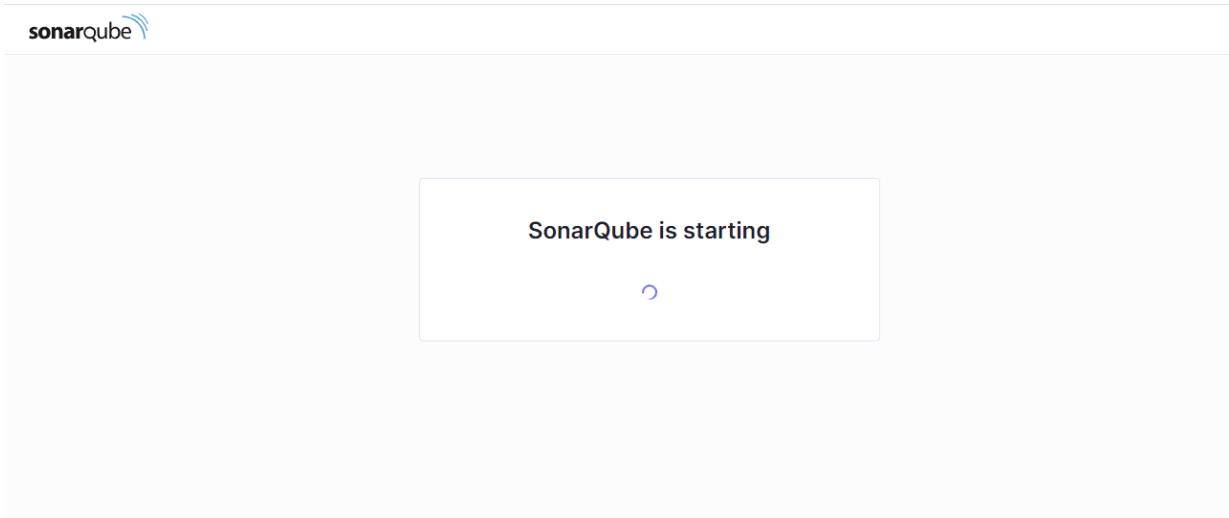
Learn more about distributed builds

Install SonarQube

```
PS Select Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS 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
762bedf4b1b7: Pull complete
95f9bd9906fa: Pull complete
a32d681e6b99: Pull complete
aabdd0a18314: Pull complete
5161e45ecd8d: Pull complete
aeb0020dfa06: Pull complete
01548d361aea: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:bb444c58c1e04d8a147a3bb12af941c57e0100a5b21d10e599384d59bed36c86
Status: Downloaded newer image for sonarqube:latest
93034b606e8bffe7ada35eddbe56af8e46b55dfd72c2a546a601dedd07016b72
PS C:\Users\admin>
```

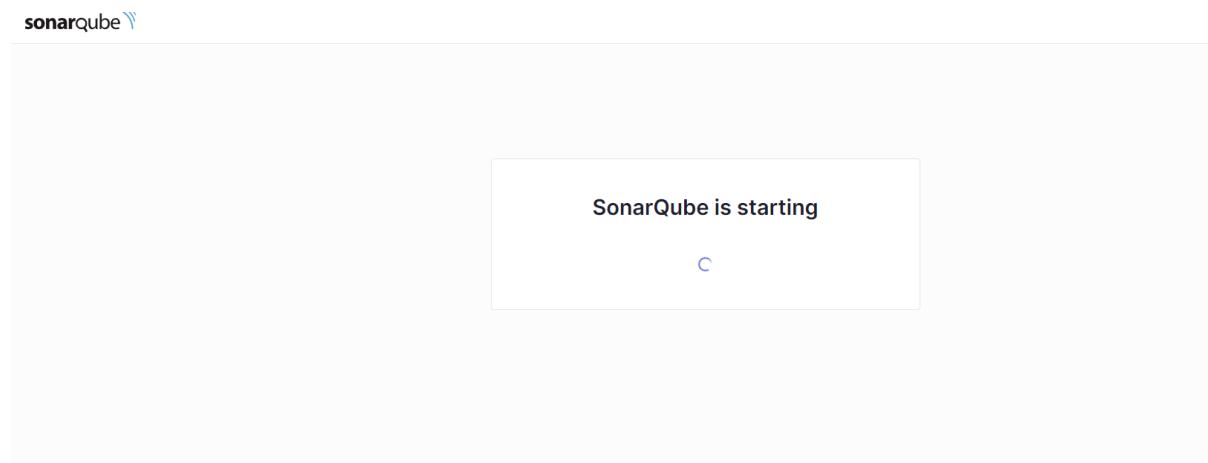


EXPERIMENT NO 8

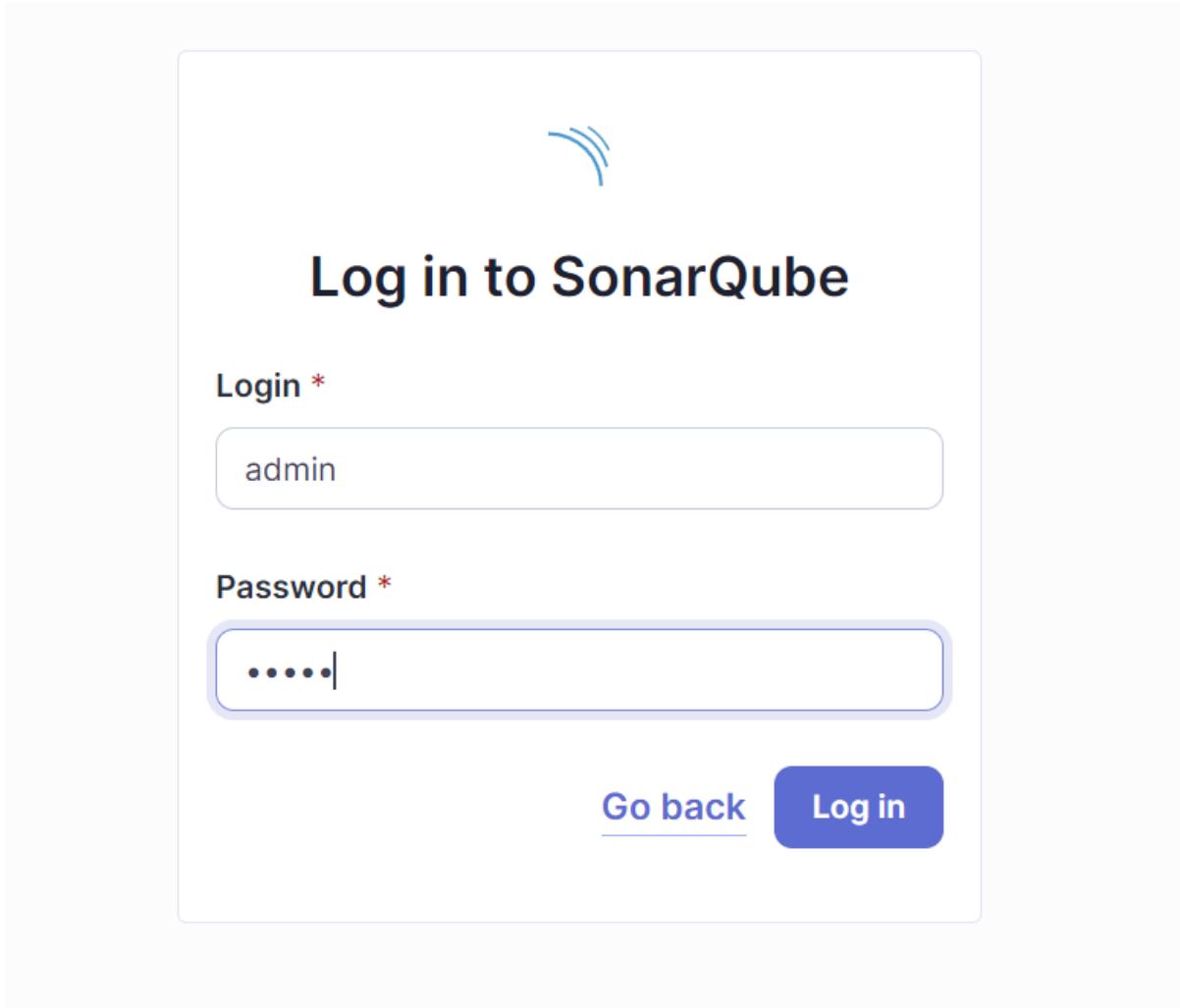
Run SonarQube in a Docker container using this command – docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest

```
PS C:\Users\HOME> 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
762bedf4b1b7: Pull complete
95f9bd9906fa: Pull complete
a32d681e6b99: Pull complete
aabdd0a18314: Pull complete
5161e45ecd8d: Pull complete
aeb0020dfa06: Pull complete
01548d361aea: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:bb444c58c1e04d8a147a3bb12af941c57e0100a5b21d10e599384d59bed36c86
Status: Downloaded newer image for sonarqube:latest
60e160dc38e9d08738e29dfab68329dbff2aa0ffe9db57fec825ee41a9d1608c
PS C:\Users\HOME>
```

At localhost:9000



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.



How do you want to create your project?

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

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

Import from Azure DevOps	Setup
Import from Bitbucket Cloud	Setup
Import from Bitbucket Server	Setup
Import from GitHub	Setup
Import from GitLab	Setup

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

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

1 of 2

Create a local project

Project display name *

sonarqube-test



Project key *

sonarqube-test



Main branch name *

main

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

[Cancel](#)

[Next](#)

2 of 2

Set up project for Clean as You Code

X

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

Choose the baseline for new code for this project

Use the global setting

Previous version

Any code that has changed since the previous version is considered new code.

Recommended for projects following regular versions or releases.

Define a specific setting for this project

Previous version

Any code that has changed since the previous version is considered new code.

Recommended for projects following regular versions or releases.

Number of days

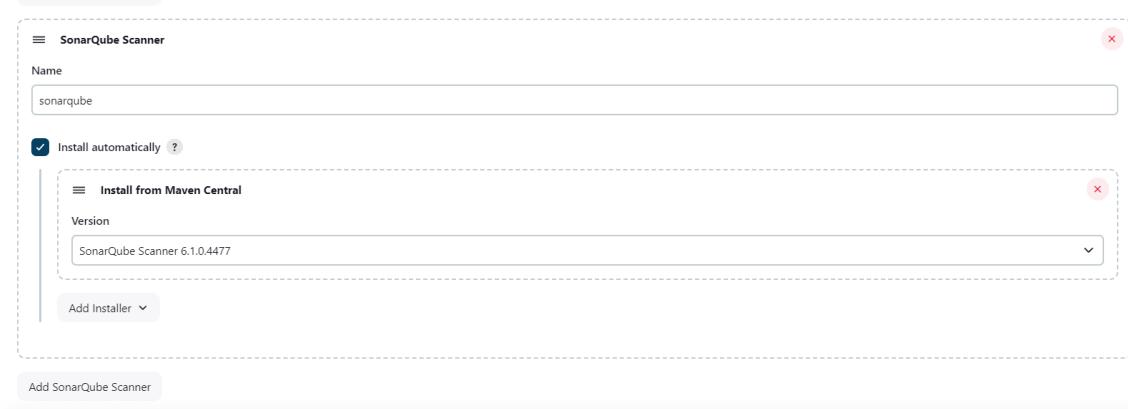
Go to Manage Jenkins and search for SonarQube Scanner for Jenkins and install it.



Under Jenkins 'Manage Jenkins' then go to 'system', scroll and look for SonarQube Servers and enter the details. Enter the Server Authentication token if needed.

A screenshot of the Jenkins System Configuration page. At the top, there is a note: "If checked, job administrators will be able to inject a sonarqube server configuration as environment variables in the build." Below this, there is a checkbox labeled "Environment variables" which is checked. The main section is titled "SonarQube installations" and contains a table with one row. The row has a "Name" column containing "sonarqube", a "Server URL" column containing "Default is http://localhost:9000" and "http://localhost:9000" in the input field, and a "Server authentication token" column containing "- none -" and "+ Add" button. There is also an "Advanced" dropdown menu.

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



Generate a token

Analyze your project

We initialized your project on SonarQube, now it's up to you to launch analyses!

1 Provide a token

[Generate a project token](#)

[Use existing token](#)

Token name [?](#)

Expires in

Analyze "sonarqube-test"

90 days

[Generate](#)

Please note that this token will only allow you to analyze the current project. If you want to use the same token to analyze multiple projects, you need to generate a global token in your [user account](#). See the [documentation](#) for more information.

The token is used to identify you when an analysis is performed. If it has been compromised, you can revoke it at any point in time in your [user account](#).

Analyze your project

We initialized your project on SonarQube, now it's up to you to launch analyses!

1 Provide a token

Analyze "sonarqube-test": sqp_def32b923cf4d6ba1a713fbf2980a90a95752daf 

The token is used to identify you when an analysis is performed. If it has been compromised, you can revoke it at any point in time in your [user account](#).

[Continue](#)

2 Run analysis on your project

Click on global under the domains part of Stores scoped to Jenkins section. Further click on add credentials. Proceed with the following details. Make sure to copy the token generated earlier in sonarqube and give any suitable name as the ID.

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

New credentials

Kind: Secret text

Scope: Global (Jenkins, nodes, items, all child items, etc)

Secret:

ID: sonarqube_practical
Unacceptable characters

Description:

[Create](#)

SonarQube installations

List of SonarQube installations

Name	<input type="text" value="sonarqube"/>
Server URL	Default is http://localhost:9000
	<input type="text" value="http://localhost:9000"/>
Server authentication token	SonarQube authentication token. Mandatory when anonymous access is disabled.
	<input type="text" value="sonarqube_practical"/>
+ Add ▾	
Advanced ▾	
Save	Apply

After configuration, create a New Item → choose a pipeline project Under Pipeline script, enter the following:

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

Dashboard > sonarqube8 > Configuration

Definition

Pipeline script

try sample Pipeline... ▾

Configure

General

Advanced Project Options

Pipeline

Script ?

```

1 - node {
2 -   stage('Cloning the GitHub Repo') {
3 -     git "https://github.com/shazforiot/GOL.git"
4 -   }
5 -   stage("SonarQube analysis") {
6 -     withSonarQubeEnv('sonarqube') {
7 -       sh "FRATH_TO SONARQUBE_FOLDER/bin/sonar-scanner \
8 -         -D sonar.login=<SonarQube_USERNAME> \
9 -         -D sonar.password=<SonarQube_PASSWORD> \
10 -        -D sonar.projectKey=<Project_KEY> \
11 -        -D sonar.exclusions=vendor/**,resources/**,*/*.java \
12 -        -D sonar.host.url=http://127.0.0.1:9000/"
13 -     }
14 -   }
15 - }

```

Use Groovy Sandbox ?

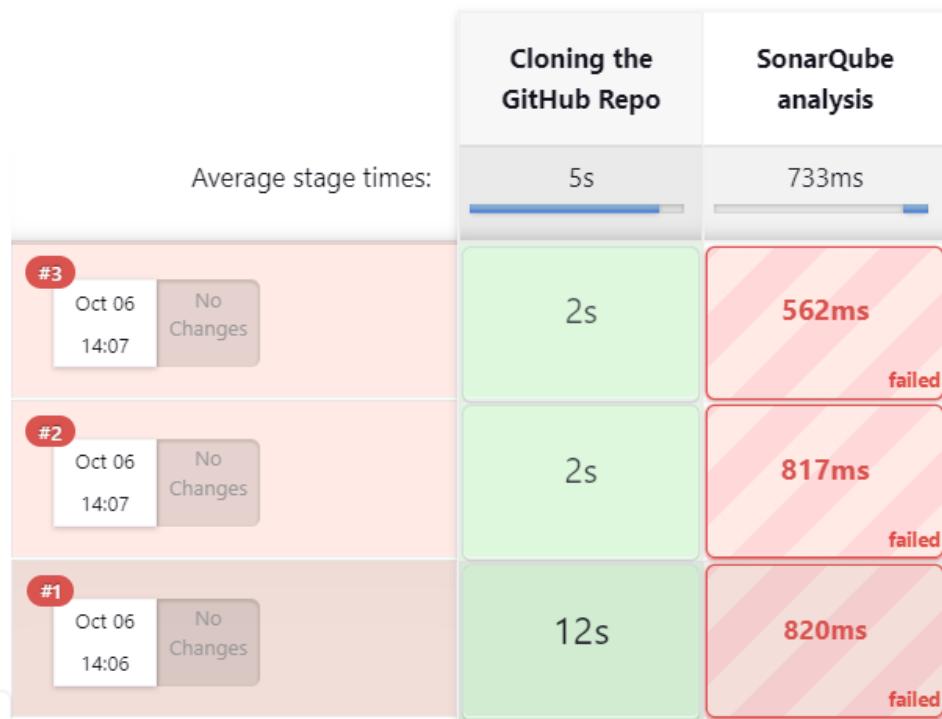
Pipeline Syntax

Save Apply

Build Project

sonarqube8

Stage View

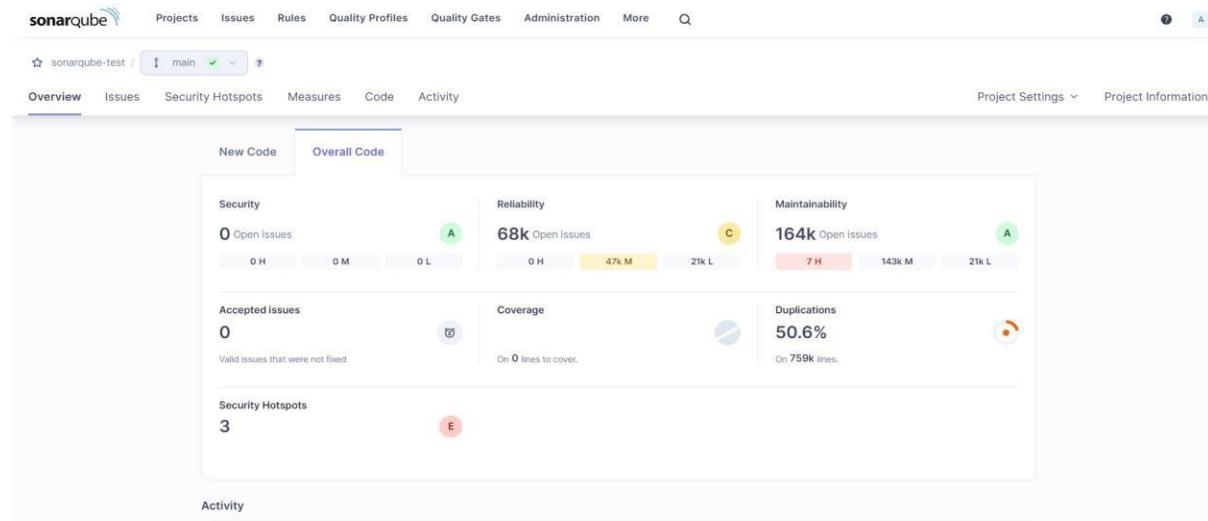


Permalinks

Console Output

```
Started by user Spandan Deb
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\sonarqube8
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Cloning the GitHub Repo)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\sonarqube8\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/shazforiot/GOL.git # timeout=10
Fetching upstream changes from https://github.com/shazforiot/GOL.git
> git.exe --version # timeout=10
> git --version # 'git version 2.43.0.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/shazforiot/GOL.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision ba799ba7e1b576f04a4612322b0412c5e6e1e5e4 (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f ba799ba7e1b576f04a4612322b0412c5e6e1e5e4 # timeout=10
> git.exe branch -a -v --no-abbrev # timeout=10
> git.exe branch -D master # timeout=10
> git.exe checkout -b master ba799ba7e1b576f04a4612322b0412c5e6e1e5e4 # timeout=10
Commit message: "Update Jenkinsfile"
> git.exe rev-list --no-walk ba799ba7e1b576f04a4612322b0412c5e6e1e5e4 # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (SonarQube analysis)
```

Project in Sonarqube



Code Problems

Consistency

The screenshot shows the SonarQube interface for the project 'sonarqube-test'. The 'Issues' tab is selected. On the left, there's a sidebar with filters and a 'Clean Code Attribute' section where 'Consistency' is highlighted. The main panel displays three issues under the file 'gameoflife-core/build/reports/tests/all-tests.html':

- Insert a <!DOCTYPE> declaration to before this <html> tag. (Reliability) Consistency user-experience
- Remove this deprecated "width" attribute. (Maintainability) Consistency html5 obsolete
- Remove this deprecated "align" attribute. (Maintainability) Consistency html5 obsolete

At the top right, it says '196,662 issues' and '3075d effort'.

Intentionality

The screenshot shows the SonarQube interface for the project 'sonarqube-test'. The 'Issues' tab is selected. On the left, there's a sidebar with filters and a 'Clean Code Attribute' section where 'Intentionality' is highlighted. The main panel displays three issues under the file 'gameoflife-acceptance-tests/Dockerfile':

- Use a specific version tag for the image. (Maintainability) Intentionality No tags
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. (Maintainability) Intentionality No tags
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. (Maintainability) Intentionality No tags

At the top right, it says '13,887 issues' and '59d effort'.

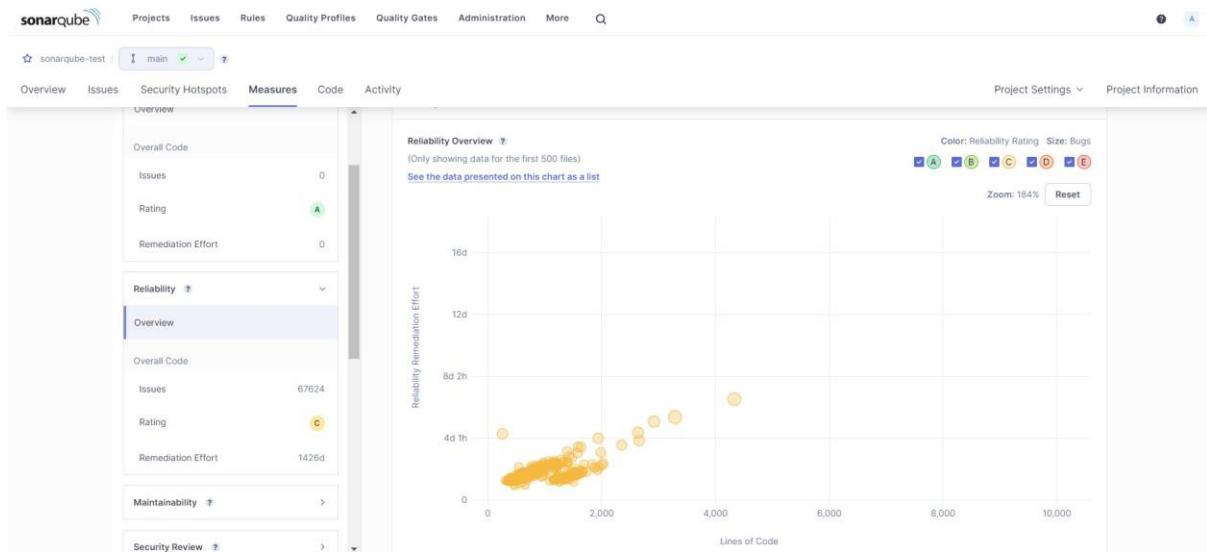
Bugs

The screenshot shows the SonarQube interface for the project 'sonarqube-test / main'. The 'Issues' tab is selected. On the left, there's a sidebar with sections for 'Clean Code Attribute' (Consistency: 54k, Intentionality: 14k) and 'Software Quality' (Security: 0, Reliability: 14k, Maintainability: 15). The main area displays two code snippets from 'gameoflife-core/build/reports/tests/all-tests.html'. The first snippet has an issue titled 'Add "lang" and/or "xml:lang" attributes to this "<html>" element' with a reliability score of 1. The second snippet has an issue titled 'Add "<th>" headers to this "<table>".' with a reliability score of 1. Both issues are marked as 'Open' and 'Not assigned'. Project statistics at the top right show 13,872 issues and 59d effort.

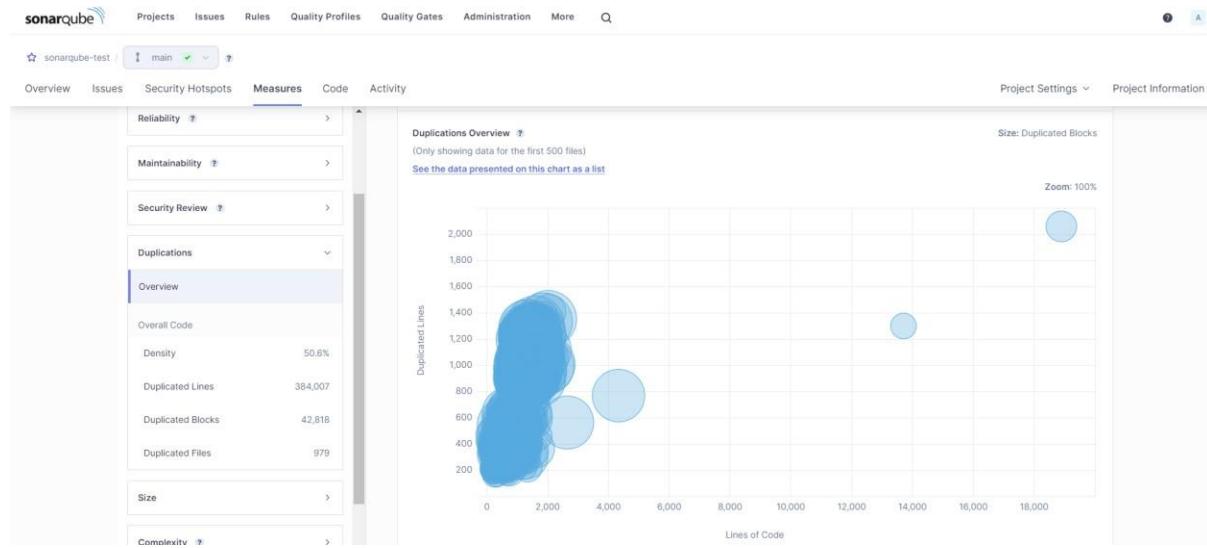
Code Smells

The screenshot shows the SonarQube interface for the project 'sonarqube-test / main'. The 'Issues' tab is selected. The sidebar shows 'Clean Code Attribute' (Consistency: 164k, Intentionality: 15) and 'Software Quality' (Security: 0, Reliability: 14k, Maintainability: 15). The main area displays three code snippets from 'gameoflife-acceptance-tests/Dockerfile'. The first snippet has an issue titled 'Use a specific version tag for the image.' with a maintainability score of 1. The second and third snippets both have issues titled 'Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.' with a maintainability score of 1. All issues are marked as 'Open' and 'Not assigned'. Project statistics at the top right show 15 issues and 44min effort.

Reliability



Duplications



Cyclomatic Complexities

The screenshot shows the SonarQube interface for a project named "sonarqube-test". The main navigation bar includes links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, More, and a search bar. Below the navigation, there are tabs for Overview, Issues, Security Hotspots, Measures (which is selected), Code, and Activity. On the right side, there are Project Settings and Project Information dropdowns.

The left sidebar contains sections for Security Review, Duplications, Overall Code, Size, and Complexity. The Complexity section is expanded, showing a table with one row: Cyclomatic Complexity 1,112.

The main content area displays the "sonarqube-test" project with a title "Cyclomatic Complexity 1,112 See history". It lists six files with their respective cyclomatic complexities:

File	Cyclomatic Complexity
gameoflife-acceptance-tests	—
gameoflife-build	—
gameoflife-core	18
gameoflife-deploy	—
gameoflife-web	1,094
pom.xml	—

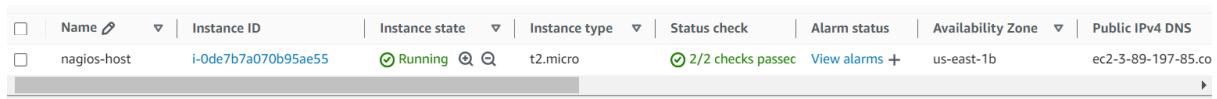
At the bottom of the main content area, it says "6 of 6 shown".

EXPERIMENT NO 9

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

Steps:

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



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

Inbound rules (7)						
Security group rule...	IP version	Type	Protocol	Port range	Source	
sgr-0ddb6805308656...	IPv4	SSH	TCP	22	0.0.0.0/0	
sgr-0497fc74d8e428588	IPv6	All ICMP - IPv6	IPv6 ICMP	All	::/0	
sgr-0f0f1647db14c0518	IPv4	All ICMP - IPv4	ICMP	All	0.0.0.0/0	
sgr-0ab870acf84613761	IPv4	All traffic	All	All	0.0.0.0/0	
sgr-0c70a5fdc2094ca48	IPv4	Custom TCP	TCP	5666	0.0.0.0/0	
sgr-09d4a2a77e90499f4	IPv4	HTTPS	TCP	443	0.0.0.0/0	
sgr-008c76246827cb4...	IPv6	HTTP	TCP	80	::/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.

```
'#_
~\_ #####
~~\_#####
~~ \|##|
~~ \#/ V~'__->
~~ .- / \
~~ /m/ [ec2-user@ip-172-31-41-29 ~]$' data-bbox="117 737 875 909"/>

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


```

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

```
sudo yum update
```

```
sudo yum install httpd php
```

```
sudo yum install gcc glibc glibc-common
```

```
sudo yum install gd gd-devel
```

```
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
  libX11-1.7.2-3.amzn2023.0.4.x86_64
  libX11-xcb-1.7.2-3.amzn2023.0.4.x86_64
  libXext-1.3.4-6.amzn2023.0.2.x86_64
  libXrender-0.9.10-14.amzn2023.0.2.x86_64
  libfffi-devel-3.4.4-1.amzn2023.0.1.x86_64
  libjpeg-turbo-2.1.4-2.amzn2023.0.5.x86_64
  libpng-2.1.6.37-10.amzn2023.0.6.x86_64
  libsep01-devel-3.4-3.amzn2023.0.3.x86_64
  libwebp-1.2.4-1.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-3.40.1-2.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-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-common-1.7.2-3.amzn2023.0.4.noarch
libXau-1.0.9-6.amzn2023.0.2.x86_64
libXpm-3.5.15-2.amzn2023.0.3.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-2.1.6.37-10.amzn2023.0.6.x86_64
libtiff-4.4.0-4.amzn2023.0.18.x86_64
libwebp-devel-1.2.4-1.amzn2023.0.6.x86_64
libxml2-devel-2.10.4-1.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-devel-1.0.9-6.amzn2023.0.2.x86_64
libXpm-devel-3.5.15-2.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-4.4.0-4.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
```

```
[ec2-user@ip-172-31-41-29 ~]$ sudo adduser -m nagios
sudo passwd nagios
Changing password for user nagios.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-41-29 ~]$ █
```

6. Create a new user group

```
sudo groupadd nagcmd
```

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

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

```
[ec2-user@ip-172-31-41-29 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-41-29 ~]$ sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
[ec2-user@ip-172-31-41-29 ~]$ █
```

8. Create a new directory for Nagios downloads

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

9. Use wget to download the source zip files.

```
wget
http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.0.8.tar.gz
```

```
wgethttp://nagiosplugins.org/download/nagiosplugin-2.0.3.tar.gz
```

```
[ec2-user@ip-172-31-41-29 downloads]$ wget
http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.0.8.tar.gz
wget: missing URL
Usage: wget [OPTION]... [URL]...
Try 'wget --help' for more options.
-Bash: http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.0.8.tar.: No such file or directory
--2024-10-08 13:55:55- http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251
Connecting to nagios-plugins.org (nagios-plugins.org)|45.56.123.251|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2659772 (2.3M) [application/x-gzip]
Saving to: 'nagios-plugins-2.0.3.tar.gz'

nagios-plugins-2.0.3.tar.gz          100%[=====]  2.54M  8.31MB/s   in 0.3s
2024-10-08 13:55:56 (8.31 MB/s) - 'nagios-plugins-2.0.3.tar.gz' saved [2659772/2659772]
```

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

```
tar zxvf nagios-4.0.8.tar.gz
```

```
nagios-4.0.8/xdata/Makefile.in
nagios-4.0.8/xdata/xcddefault.c
nagios-4.0.8/xdata/xcddefault.h
nagios-4.0.8/xdata/xodtemplate.c
nagios-4.0.8/xdata/xodtemplate.h
nagios-4.0.8/xdata/xpddefault.c
nagios-4.0.8/xdata/xpddefault.h
nagios-4.0.8/xdata/xrddefault.c
nagios-4.0.8/xdata/xrddefault.h
nagios-4.0.8/xdata/xsddefault.c
nagios-4.0.8/xdata/xsddefault.h
```

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

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

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

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

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

12. Compile the source code.

```
make all
```

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

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

```
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/nagios.cfg /usr/local/nagios/etc/nagios.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/cgi.cfg /usr/local/nagios/etc/cgi.cfg
/usr/bin/install -c -b -m 660 -o nagios -g nagios sample-config/resource.cfg /usr/local/nagios/etc/resource.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/templates.cfg /usr/local/nagios/etc/objects/template
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/commands.cfg /usr/local/nagios/etc/objects/commands.
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/contacts.cfg /usr/local/nagios/etc/objects/contacts.
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/timeperiods.cfg /usr/local/nagios/etc/objects/timepe
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/localhost.cfg /usr/local/nagios/etc/objects/localhos
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/windows.cfg /usr/local/nagios/etc/objects/windows.cf
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/printer.cfg /usr/local/nagios/etc/objects/printer.cf
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/switch.cfg /usr/local/nagios/etc/objects/switch.cfg

*** Config files installed ***

Remember, these are *SAMPLE* config files. You'll need to read
the documentation for more information on how to actually define
services, hosts, etc. to fit your particular needs.

/usr/bin/install -c -m 775 -o nagios -g nagcmd -d /usr/local/nagios/var/rw
chmod g+s /usr/local/nagios/var/rw

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

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

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

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

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

define contact{
    contact_name          nagiosadmin      ; Short name of user
    use                   generic-contact   ; Inherit default values from generic-contact template (defined above)
    alias                Nagios Admin     ; Full name of user

    email                spandan.deb04@gmail.com ; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS *****

}
```

^G Help ^C Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark M-J To Brac
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo M-C Copy ^Q Where Wa

15. Configure the web interface.

```
sudo make install-webconf
```

```
[ec2-user@ip-172-31-41-29 nagios-4.0.8]$ sudo make install-webconf  
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf  
*** Nagios/Apache conf file installed ***
```

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-172-31-41-29 nagios-4.0.8]$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin  
New password:  
Re-type new password:  
Adding password for user nagiosadmin
```

17. Restart Apache

```
sudo service httpd restart
```

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

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

```
nagios-plugins-2.0.3/plugins-scripts/check_sensors.sh  
nagios-plugins-2.0.3/pkg/  
nagios-plugins-2.0.3/pkg/fedora/  
nagios-plugins-2.0.3/pkg/fedora/requirements  
nagios-plugins-2.0.3/pkg/solaris/  
nagios-plugins-2.0.3/pkg/solaris/preinstall  
nagios-plugins-2.0.3/pkg/solaris/solpkg  
nagios-plugins-2.0.3/pkg/solaris/pkginfo.in  
nagios-plugins-2.0.3/pkg/solaris/pkginfo  
nagios-plugins-2.0.3/pkg/redhat/
```

19. Compile and install plugins

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

```
config.status: executing po-directories commands
config.status: creating po/POTFILES
config.status: creating po/Makefile
  --with-apt-get-command:
    --with-ping6-command: /usr/sbin/ping6 -n -U -w %d -c %d %s
    --with-ping-command: /usr/bin/ping -n -U -w %d -c %d %s
      --with-ipv6: yes
      --with-mysql: no
      --with-openssl: no
      --with-gnutls: no
  --enable-extra-opts: yes
    --with-perl: /usr/bin/perl
--enable-perl-modules: no
  --with-cgiurl: /nagios/cgi-bin
--with-trusted-path: /bin:/sbin:/usr/bin:/usr/sbin
  --enable-libtap: no
```

20. Start Nagios

Add Nagios to the list of system services

```
sudo chkconfig --add nagios
sudo chkconfig nagios on
[ec2-user@ip-172-31-41-29 nagios-plugins-2.0.3]$ sudo chkconfig --add nagios
sudo chkconfig nagios on
[ec2-user@ip-172-31-41-29 nagios-plugins-2.0.3]$ █
```

Verify the sample configuration files

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

```

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

```

If there are no errors, you can go ahead and start Nagios.

```
sudo service nagios start
```

```
[ec2-user@ip-172-31-41-29 ~]$ sudo service nagios start
Reloading systemd:                                     [ OK ]
Starting nagios (via systemctl):                     [ OK ]
[ec2-user@ip-172-31-41-29 ~]$
```

21. Check the status of Nagios

```
sudo systemctl status nagios
```

```

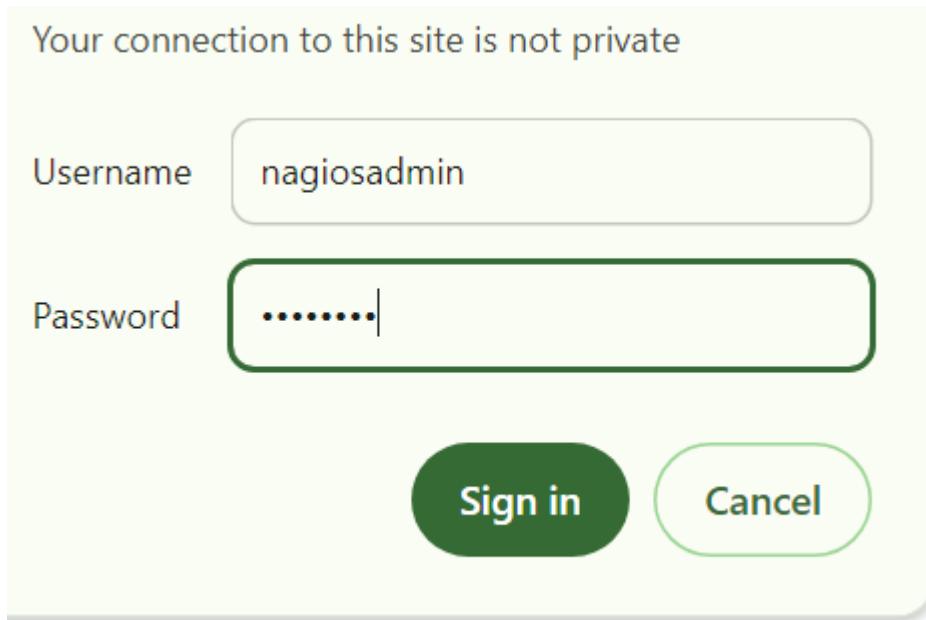
Reloading systemd:                                     [ OK ]
Starting nagios (via systemctl):                     [ OK ]
[ec2-user@ip-172-31-41-29 ~]$ sudo systemctl status nagios
● nagios.service - LSB: Starts and stops the Nagios monitoring server
  Loaded: loaded (/etc/rc.d/init.d/nagios; generated)
  Active: active (running) since Tue 2024-10-08 14:21:59 UTC; 40s ago
    Docs: man:systemd-sysv-generator(8)
  Process: 67484 ExecStart=/etc/rc.d/init.d/nagios start (code=exited, status=0/SUCCESS)
    Tasks: 6 (limit: 1112)
   Memory: 2.1M
      CPU: 40ms
     CGroup: /system.slice/nagios.service
             └─67506 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
                  ├─67508 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
                  ├─67509 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
                  ├─67510 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
                  ├─67511 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
                  └─67512 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

```

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

23. Open up your browser and look for

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



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

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

A screenshot of the Nagios Core 4.4.6 dashboard. At the top, the Nagios logo is displayed with the text "Nagios® Core™" and a note "Daemon running with PID 49199". Below the logo, it says "Nagios® Core™ Version 4.4.6 April 28, 2020 Check for updates". A blue banner at the top right says "A new version of Nagios Core is available! Visit nagios.org to download Nagios 4.5.5". On the left, there is a sidebar with links for General, Home, Documentation, Current Status, Hosts, Services, Host Groups, Service Groups, Problems, Reports, and System. The main content area features several sections: "Get Started" with a list of bullet points, "Quick Links" with a list of links, and two smaller boxes for "Latest News" and "Don't Miss...".

EXPERIMENT NO.10

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

Procedure:-

Check if the nagios service is running by executing following command

```
ubuntu@ip-172-31-89-161:~$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.4.6
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: enabled)
   Active: active (running) since Sat 2024-09-28 16:08:58 UTC; 1min 2s ago
     Docs: https://www.nagios.org/documentation
 Process: 15743 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Process: 15753 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 15764 (nagios)
   Tasks: 6 (limit: 1130)
  Memory: 2.4M (peak: 3.2M)
    CPU: 29ms
   CGroup: /system.slice/nagios.service
           ├─15764 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           ├─15765 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─15766 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─15767 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─15768 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─15769 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfully initialized
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: qh: core query handler registered
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: qh: echo service query handler registered
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: qh: help for the query handler registered
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: wproc: Successfully registered manager as @wproc with query handler
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: wproc: Registry request: name=Core Worker 15765;pid=15765
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: wproc: Registry request: name=Core Worker 15766;pid=15766
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: wproc: Registry request: name=Core Worker 15767;pid=15767
```

sudo systemctl status nagios

Now, create a new EC2 instance on AWS

Instances (2) Info		Last updated	C	Connect	Instance state	Actions	Launch instances	
		less than a minute ago			All states			
<input type="checkbox"/>	Name ↗	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	P
<input type="checkbox"/>	nagios-host	i-09e8ea019f24f4be2	Running Q Q	t2.micro	2/2 checks passed	View alarms +	us-east-1c	e
<input type="checkbox"/>	linux-client	i-0ad38836f030e3784	Running Q Q	t2.micro	Initializing	View alarms +	us-east-1c	e

Now perform the following commands on nagios-host EC2 instance.

On the server, run this command

```
ubuntu@ip-172-31-89-161:~$ ps -ef | grep nagios
nagios 15764 1 0 16:08 ? 00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios 15765 15764 0 16:08 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 15766 15764 0 16:08 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 15767 15764 0 16:08 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 15768 15764 0 16:08 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 15769 15764 0 16:08 ? 00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
ubuntu 15957 1342 0 16:13 pts/0 00:00:00 grep --color=auto nagios
ubuntu@ip-172-31-89-161:~$
```

ps -ef | grep nagios

Become a root user and create 2 folders

sudo su

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

```
ubuntu@ip-172-31-89-161:~$ sudo su  
mkdir /usr/local/nagios/etc/objects/monitorhosts  
mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts  
root@ip-172-31-89-161:/home/ubuntu#
```

Copy localhost.cfg file to the mentioned location

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

```
root@ip-172-31-89-161:/usr/local/nagios/etc/objects# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts  
cp: cannot create regular file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts': No such file or directory  
root@ip-172-31-89-161:/usr/local/nagios/etc/objects# sudo mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts  
root@ip-172-31-89-161:/usr/local/nagios/etc/objects# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts  
root@ip-172-31-89-161:/usr/local/nagios/etc/objects#
```

```
/usr/local/nagios/etc/objects/monitorhosts/linuxhosts
```

Open the nano editor for localhost.cfg file and make these changes. Add the Ip address of the linux-client for the address field.

```
nano
```

```
GNU nano 7.2                                     /usr/local/nagios/etc/objects/monitorhosts/linuxhosts  
#####  
#  
# HOST DEFINITION  
#  
#####  
  
# Define a host for the local machine  
  
define host {  
  
    use          linux-server ; Name of host template  
    ; This host definition is used as a base for other host definitions  
    ; in (or inherited by) other host definitions.  
    host_name    linuxserver  
    alias        linuxserver  
    address      52.207.253.18  
}  
  
#####  
#  
# HOST GROUP DEFINITION  
  
^G Help      ^O Write Out     ^W Where Is      ^K Cut      ^T Exit  
^X Exit      ^R Read File     ^\ Replace      ^U Paste     ^J Jump  
/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/localhost.cfg
```

Note - Here replace hostname with linuxserver

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

Add the following line to the nagios.cfg file

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

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

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

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

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

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

After making the changes in nagios.cfg file now check validate the file by typing the following command in the terminal.

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

```
License: GPL

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

Running pre-flight check on configuration data...

Checking objects...
  Checked 16 services.
  Checked 2 hosts.
  Checked 2 host groups.
  Checked 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
root@ip-172-31-89-161:/usr/local/nagios/etc/objects/monitorhosts/linuxhosts#
```

Now restart the service by using this command

```

root@ip-172-31-89-161:/usr/local/nagios/etc/objects/monitorhosts/linuxhosts# service nagios restart
root@ip-172-31-89-161:/usr/local/nagios/etc/objects/monitorhosts/linuxhosts# systemctl status nagios
● nagios.service - Nagios Core 4.4.6
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: enabled)
   Active: active (running) since Sat 2024-09-28 17:36:35 UTC; 19s ago
     Docs: https://www.nagios.org/documentation
  Process: 1870 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
  Process: 1872 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 1874 (nagios)
    Tasks: 8 (limit: 1130)
   Memory: 3.0M (peak: 3.2M)
      CPU: 24ms
     CGroupl: /system.slice/nagios.service
           |-1874 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           |-1875 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           |-1876 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           |-1877 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           |-1878 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           |-1879 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           |-1880 /usr/local/nagios/libexec/check_ping -H 52.207.253.18 -w 3000.0,80% -c 5000.0,100% -p 5
           └-1881 /usr/bin/ping -n -U -w 30 -c 5 52.207.253.18

Sep 28 17:36:35 ip-172-31-89-161 nagios[1874]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfully initialized
Sep 28 17:36:35 ip-172-31-89-161 nagios[1874]: qh: core query handler registered
Sep 28 17:36:35 ip-172-31-89-161 nagios[1874]: qh: echo service query handler registered
Sep 28 17:36:35 ip-172-31-89-161 nagios[1874]: qh: help for the query handler registered
Sep 28 17:36:35 ip-172-31-89-161 nagios[1874]: wproc: Successfully registered manager as @wproc with query handler
Sep 28 17:36:35 ip-172-31-89-161 nagios[1874]: wproc: Registry request: name=Core Worker 1875;pid=1875
lines 1-26

```

service nagios restart

Now using this command update the apt repository of ubuntu (linux-client),
install gcc, nagios-nrpe-server and nagios-plugin

sudo apt update -y

sudo apt install gcc -y

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

Now open nrpe.cfg file and add the ip address of the nagios host as shown. To open the nrpe.cfg file copy this command.

```

# supported.

#
# Note: The daemon only does rudimentary checking
# address. I would highly recommend adding entries
# file to allow only the specified host to connect
# you are running this daemon on.
#
# NOTE: This option is ignored if NRPE is running
#       as a separate process.
allowed_hosts=127.0.0.1,54.167.169.0

# COMMAND ARGUMENT PROCESSING
# This option determines whether or not the NRPE
# to specify arguments to commands that are executed
# if the daemon was configured with the --enable-command-
# option.

```

sudo nano /etc/nagios/nrpe.cfg

Now restart nrpe server by using this command
 sudo systemctl restart nagios-nrpe-server

Now, check nagios dashboard, you should see linuxserver up and running, if not

Current Network Status

Last Updated: Sat Sep 28 18:47:41 UTC 2024
 Updated every 90 seconds
 Nagios® Core™ 4.4.6 - www.nagios.org
 Logged in as nagiosadmin

Host Status Totals

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

Service Status Totals

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

Host Status Details For All Host Groups

Host	Status	Last Check	Duration	Status Information
linuxserver	UP	09-28-2024 18:45:20	0d 0h 2m 21s	PING OK - Packet loss = 68%, RTA = 0.63 ms
localhost	UP	09-28-2024 18:44:05	0d 4h 47m 45s	PING OK - Packet loss = 0%, RTA = 0.04 ms

Results 1 - 2 of 2 Matching Hosts

check security groups of the EC2 instances.

EXPERIMENT NO 11

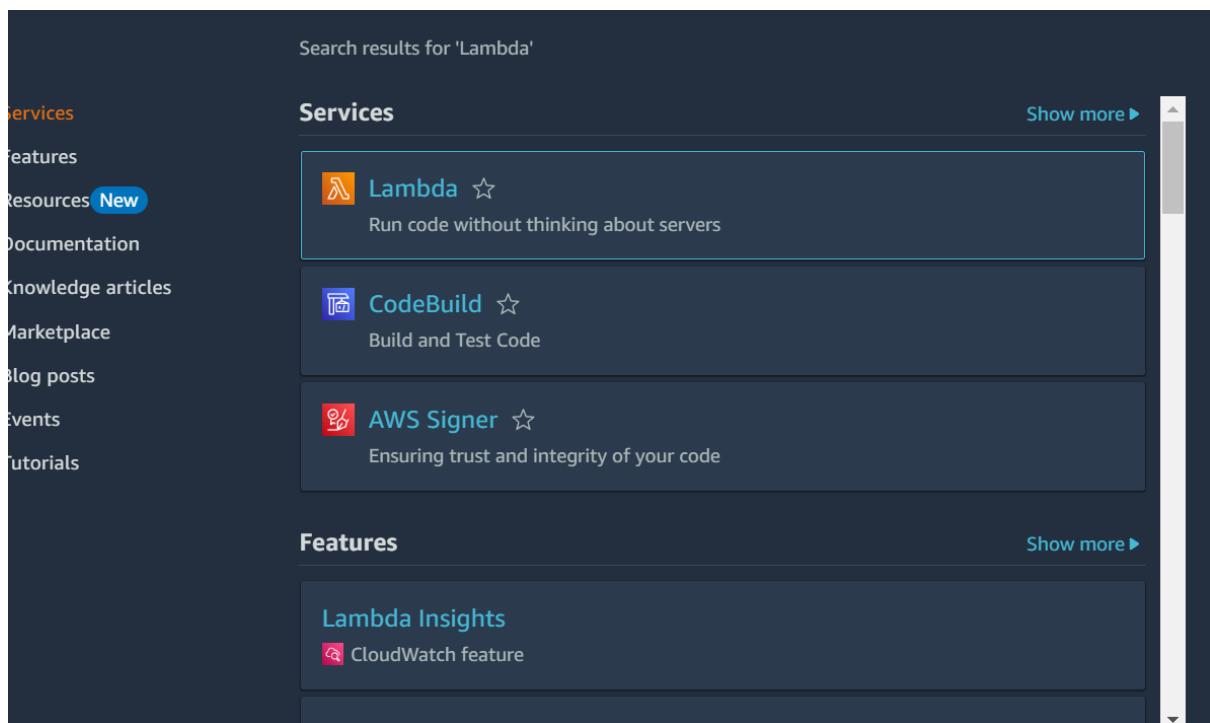
NAME-SPANDAN DEB

CLASS-D15A

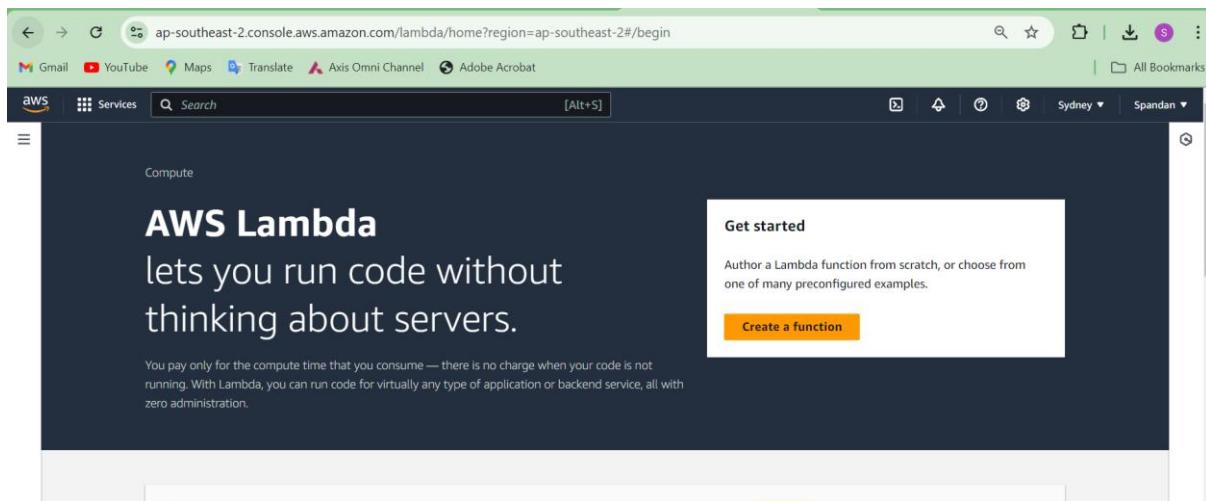
ROLL NO-13

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

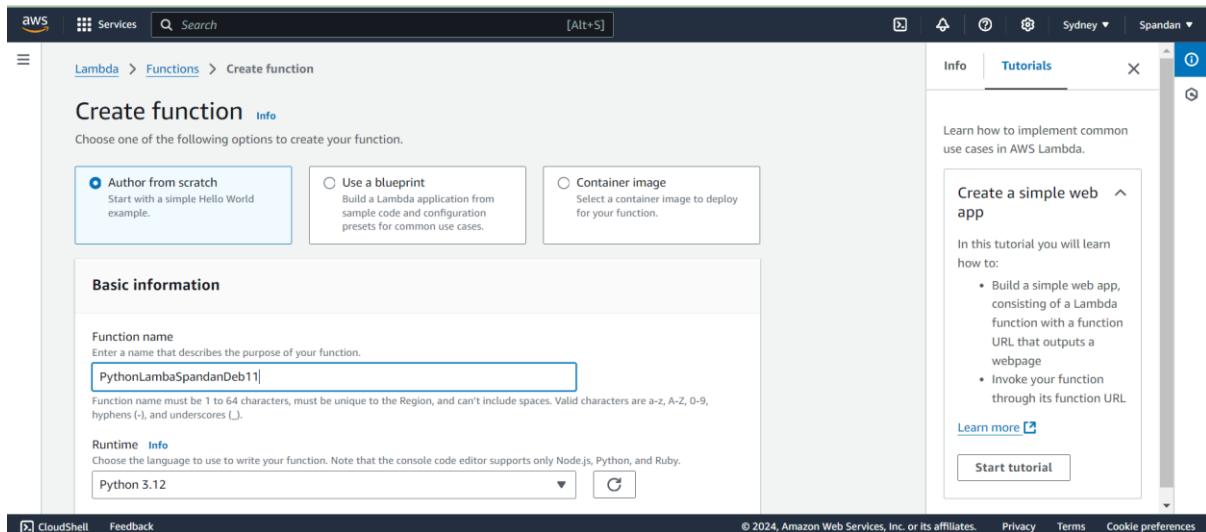
Open Lambda to create an AWS Lambda function



Open up the Lambda Console and click on the Create button



Choose to create a function from scratch or use a blueprint, i.e templates defined by AWS for you with all configuration presets required for the most common use cases. 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. After that,



Function is successfully created

The screenshot shows the AWS Lambda console. A green success message at the top states: "Successfully created the function PythonLambaSpandanDeb11. You can now change its code and configuration. To invoke your function with a test event, choose 'Test'." Below this, the function name "PythonLambaSpandanDeb11" is displayed. The "Function overview" tab is selected, showing a diagram of the function with the name "PythonLamba SpandanDeb11" and version "1". To the right, there is a "Description" field with a dash, "Last modified" at "0 seconds ago", and a "Tutorials" sidebar titled "Create a simple web app" with instructions on building a Lambda function for a webpage.

The screenshot shows the AWS Lambda code editor. The left sidebar shows the environment with a folder named "PythonLambaSpand" containing a file "lambda_function.py". The main area displays the Python code for the 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
```

To change the configuration, open up the Configuration tab and under General Configuration, choose Edit. Here, you can enter a description and change Memory and Timeout. I've changed the Timeout period to 1 sec since that is sufficient for now

The screenshot shows the AWS Lambda Configuration page. The top navigation bar includes tabs for Code, Test, Monitor, Configuration (which is selected), Aliases, and Versions. On the left, a sidebar menu lists General configuration, Triggers, Permissions, Destinations, Function URL, Environment variables, Tags, VPC, and DNS databases. The main content area displays the General configuration settings:

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

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

This screenshot shows the detailed configuration for a specific Lambda function. The top navigation bar includes Services, Search, and a Tutorials tab. The main configuration section includes fields for Memory (128 MB), Ephemeral storage (512 MB), and SnapStart (None). Other settings shown include Timeout (0 min 1 sec) and an Execution role selection (Use an existing role). A sidebar on the right provides a tutorial titled "Create a simple web app".

You can make changes to your function inside the code editor. You can also upload a zip file of your function or upload one from an S3 bucket if needed. Press Ctrl + S to save the file and click Deploy to deploy the changes.

This screenshot shows the AWS Lambda code editor after a deployment. A green success message at the top states "Successfully updated the function PythonLambaSpandanDeb11.". The code editor displays the function code in a file named lambda_function.py:

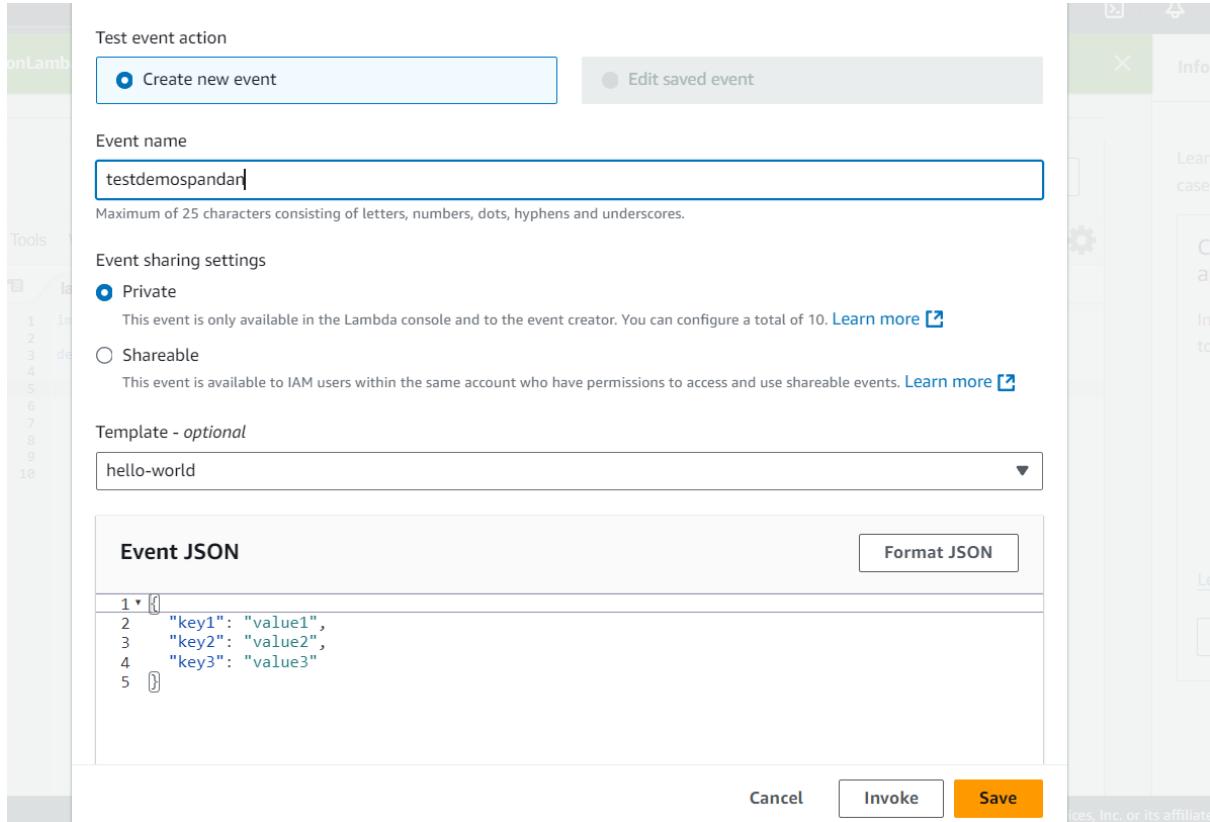
```

1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     new_string="Hi! This is Spandan here!"
6     return {
7         'statusCode': 200,
8         'body': json.dumps('Hello from Lambda!')
9     }
10

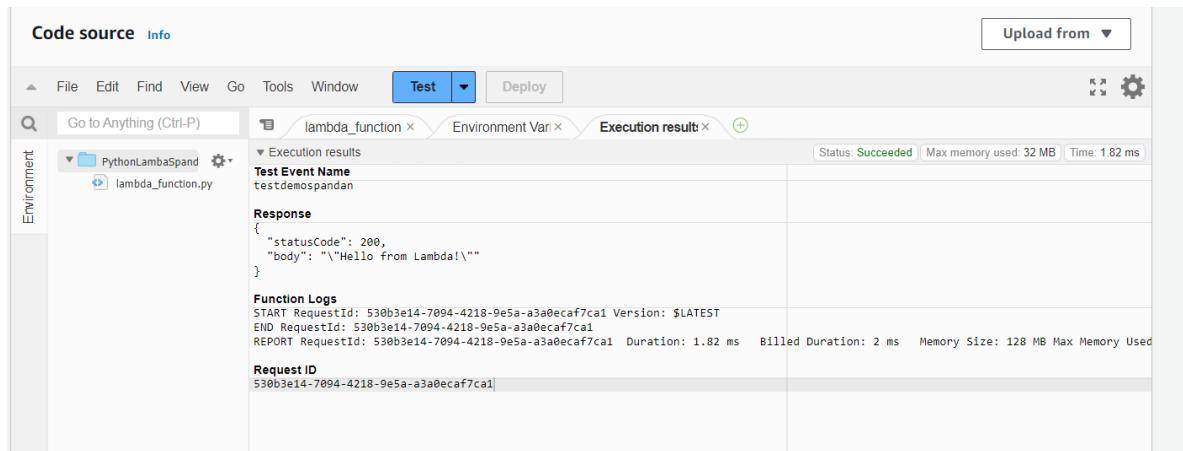
```

The editor interface includes tabs for Code, Test, Monitor, Configuration, Aliases, and Versions. A sidebar on the right provides a tutorial titled "Create a simple web app".

Click on Test and you can change the configuration, like so. If you do not have anything in the request body, it is important to specify two curly braces as valid JSON, so make sure they are there.



Click on Test and you should be able to see the results

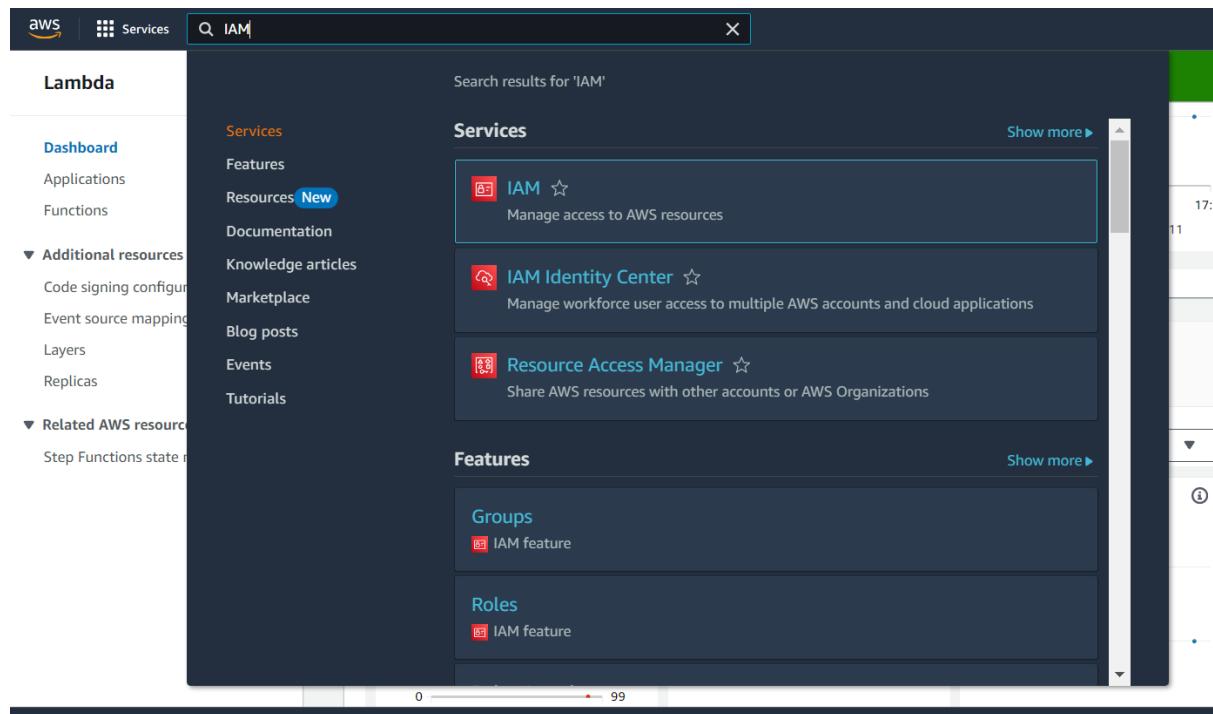


EXPERIMENT NO 12

NAME-SPANDAN DEB
CLASS-D15A
ROLL NO-13

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.

Select an IAM services



Open up the IAM Console and under Roles, choose the Role we previously created for the Python Lambda Function (You can find your role name configuration of your Lambda function).

The screenshot shows the AWS IAM Roles page. On the left, there's a navigation sidebar with options like Dashboard, Access management (Roles, Policies, Identity providers, Account settings), Access reports (Access Analyzer, External access, Unused access, Analyzer settings), and Credential report. The main area displays a table titled 'Roles (9) Info' with columns for Role name, Trusted entities, and Last activity. The roles listed are: AWSCodePipelineServiceRole-ap-southeast-2-SD-CI_CD-PIPELINE, AWSServiceRoleForAmazonSSM, AWSServiceRoleForSupport, AWSServiceRoleForTrustedAdvisor, CodeDeployRole, CodeDeployRole1, EC2CodeDeploy, PythonLambdaSpandanDeb11-role-rzmypo7i, and spandan. Each role has a corresponding AWS service listed under 'Trusted entities'.

Under Attach Policies, add S3-ReadOnly and CloudWatchFull permissions to this role.

This screenshot shows the 'Add permissions' dialog for the role 'PythonLambdaSpandanDeb11-role-rzmypo7i'. The 'Current permissions policies' section shows one policy named 'S3Read'. The 'Other permissions policies' section is filtered by 'Type' to show 'AWS managed' policies. One policy, 'AmazonS3ReadOnlyAccess', is selected and its description is visible: 'Provides read only access to all buckets via the Amazon S3 API'. At the bottom right are 'Cancel' and 'Add permissions' buttons.

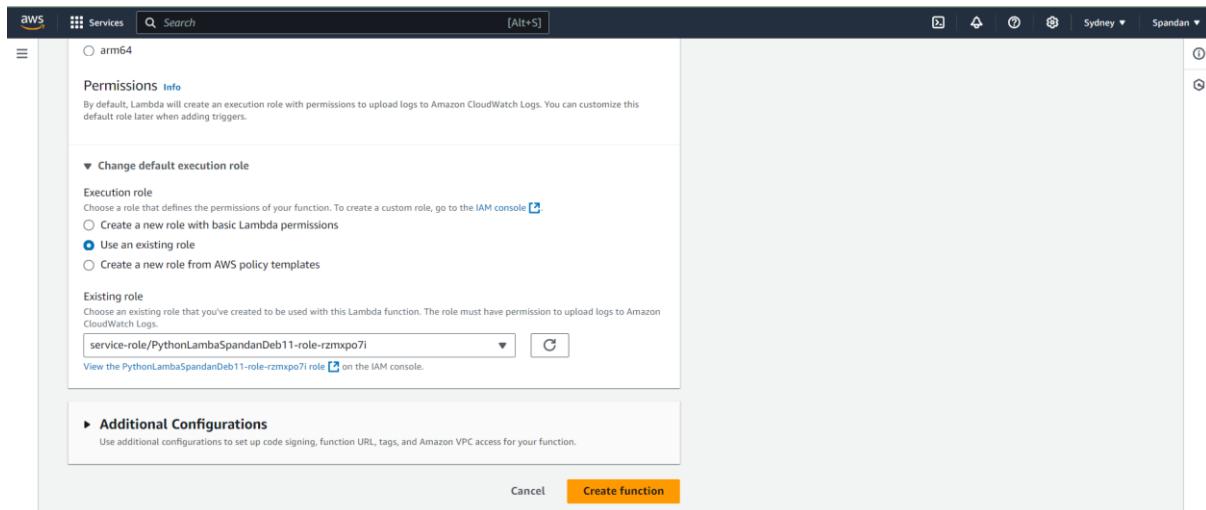
This screenshot shows the 'Add permissions' dialog after the 'CloudWatchFullAccess' policy has been attached. The 'Current permissions policies' section now shows two policies: 'S3Read' and 'CloudWatchFullAccess'. The 'Other permissions policies' section is filtered by 'Type' to show 'AWS managed' policies. The 'CloudWatchFullAccess' policy is checked and its description is visible: 'Provides full access to CloudWatch'. At the bottom right are 'Cancel' and 'Add permissions' buttons.

After successful attachment of policy you will see something like this you will be able to see the updated policies

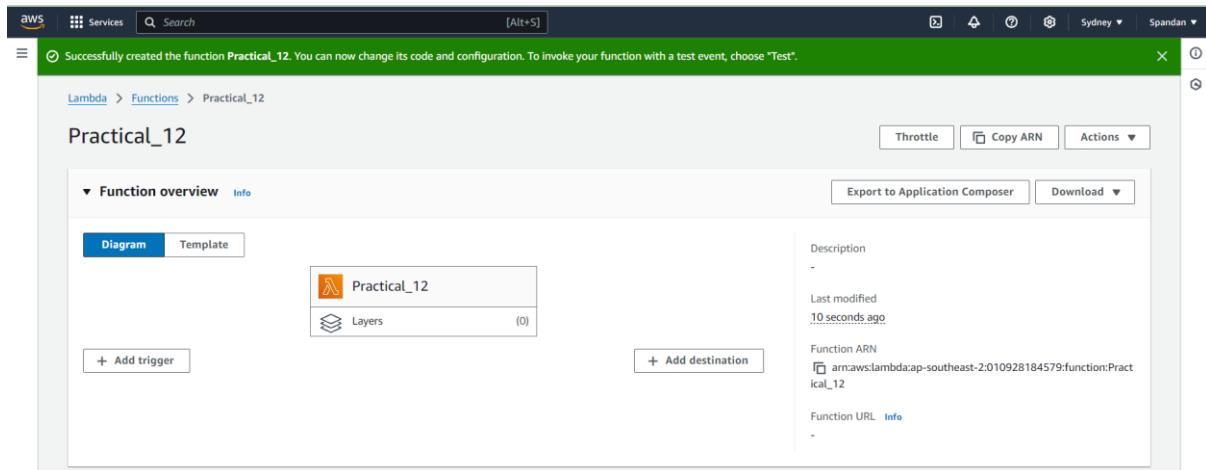
The screenshot shows the AWS IAM Permissions page. At the top, a green banner displays the message "Policy was successfully attached to role." Below the banner, there are sections for "Last activity" (26 minutes ago) and "Maximum session duration" (1 hour). A navigation bar includes tabs for "Permissions", "Trust relationships", "Tags", "Last Accessed", and "Revoke sessions". Under the "Permissions" tab, a sub-section titled "Permissions policies (3)" is shown. It includes a search bar, a filter for "All types", and a table listing three policies: "AmazonS3ReadOnlyAccess" (AWS managed), "AWSLambdaBasicExecutionRole-ed56f103..." (Customer managed), and "CloudWatchFullAccess" (AWS managed). Each policy entry shows its type, name, and the number of attached entities.

Open up AWS Lambda and create a new Python function. Under Execution Role, choose the existing role, then select the one which was previously created and to which we just added permissions.

The screenshot shows the AWS Lambda "Create function" page. The top section, "Choose one of the following options to create your function.", contains three radio button options: "Author from scratch" (selected), "Use a blueprint", and "Container image". Below this, the "Basic information" section is visible. It includes fields for "Function name" (set to "Practical_12"), "Runtime" (set to "Python 3.12"), and "Architecture" (set to "x86_64"). The "Basic information" section is currently expanded, while other sections like "Environment variables" and "Code" are collapsed.



The function is now successfully created and running



Make the following changes to the function and click on the deploy button. This code basically logs a message and logs the contents of a JSON file which is uploaded to an S3 Bucket and then deploy the code.

Successfully created the function Practical_12. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Code | Test | Monitor | Configuration | Aliases | Versions

Code source **Info**

Test | Deploy | Changes not deployed

```

lambda_function.py
1 import json
2 import boto3
3 import urllib
4
5 def lambda_handler(event, context):
6     s3_client = boto3.client('s3')
7
8     bucket_name = event['Records'][0]['s3']['bucket']['name']
9     key = event['Records'][0]['s3']['object']['key']
10
11    key = urllib.parse.unquote_plus(key, encoding='utf-8')
12
13    message = f'A file has been added with key {key} to the bucket {bucket_name}'
14    print(message)
15
16    response = s3_client.get_object(Bucket=bucket_name, Key=key)
17    contents = response['Body'].read().decode()
18
19    contents = json.loads(contents)
20    print(f'These are the Contents of the File: \n{contents}')
21

```

Click on Test and choose the ‘S3 Put’ Template

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

Test event action

Create new event Edit saved event

Event name
practical

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

Event sharing settings

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

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

Template - optional
s3-put

Event JSON

```

1 [
2   {
3     "Records": [
4       {
5         "eventVersion": "2.0",
6         "eventSource": "aws:s3",
7         "awsRegion": "us-east-1",
8         "eventTime": "1970-01-01T00:00:00Z",
9         "eventName": "ObjectCreated:Put",
10        "userIdentity": {
11          "principal": "arn:aws:iam::123456789012:root"
12        }
13      }
14    ]
15  }
16]

```

Format JSON

Cancel | Invoke | Save

Open up the S3 Console and create a new bucket.

The screenshot shows the AWS S3 Buckets page. On the left, there's a sidebar with options like Buckets, Storage Lens, and IAM Access Analyzer. The main area displays an account snapshot and a list of General purpose buckets. Two buckets are listed:

Name	AWS Region	IAM Access Analyzer	Creation date
codepipeline-ap-southeast-2 53836253805	Asia Pacific (Sydney) ap-southeast-2	View analyzer for ap-southeast-2	August 11, 2024, 23:24:32 (UTC+05:30)
spandan1	Asia Pacific (Sydney) ap-southeast-2	View analyzer for ap-southeast-2	August 6, 2024, 23:12:07 (UTC+05:30)

With all general settings, create the bucket in the same region as the function.

The screenshot shows the 'Create bucket' wizard. The first step, 'General configuration', is active. It asks for the AWS Region (set to Asia Pacific (Sydney) ap-southeast-2), a Bucket name (with 'addevopsexp12' typed in), and optional copy settings from another bucket. The second step, 'Object Ownership', is partially visible below.

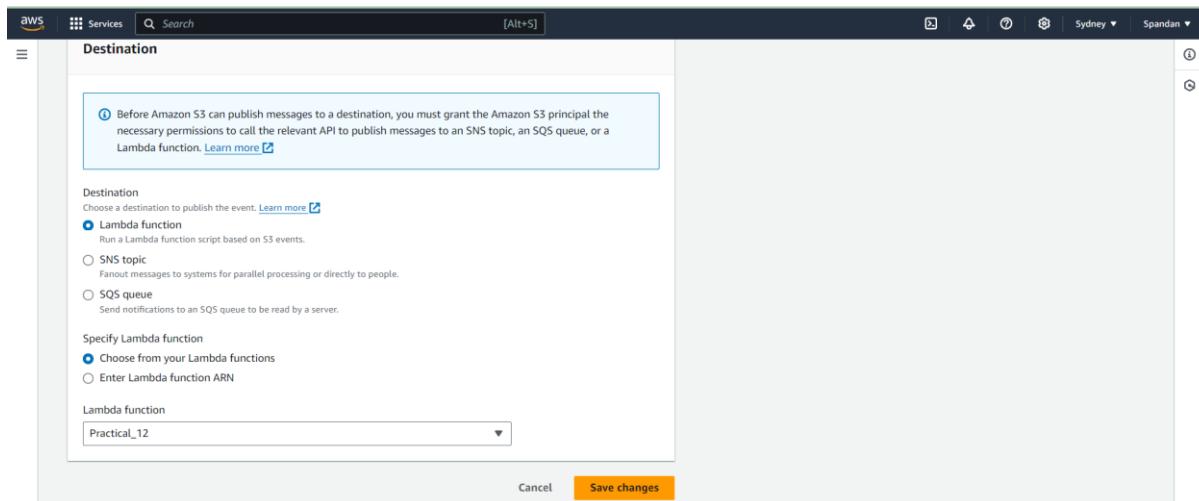
Click on the created bucket and under properties, look for events. Click on Create Event Notification

The screenshot shows the AWS CloudTrail console for an S3 bucket named "Spandan". The main area displays "Event notifications (0)" with a message: "No data events to display." A button "Configure in CloudTrail" is available. Below this, there's a table for "Amazon EventBridge" settings, which is currently off. The table columns are "Name", "Event types", "Filters", "Destination type", and "Destination". A "Create event notification" button is also present. At the bottom, there's a section for "Transfer acceleration" with a link to learn more.

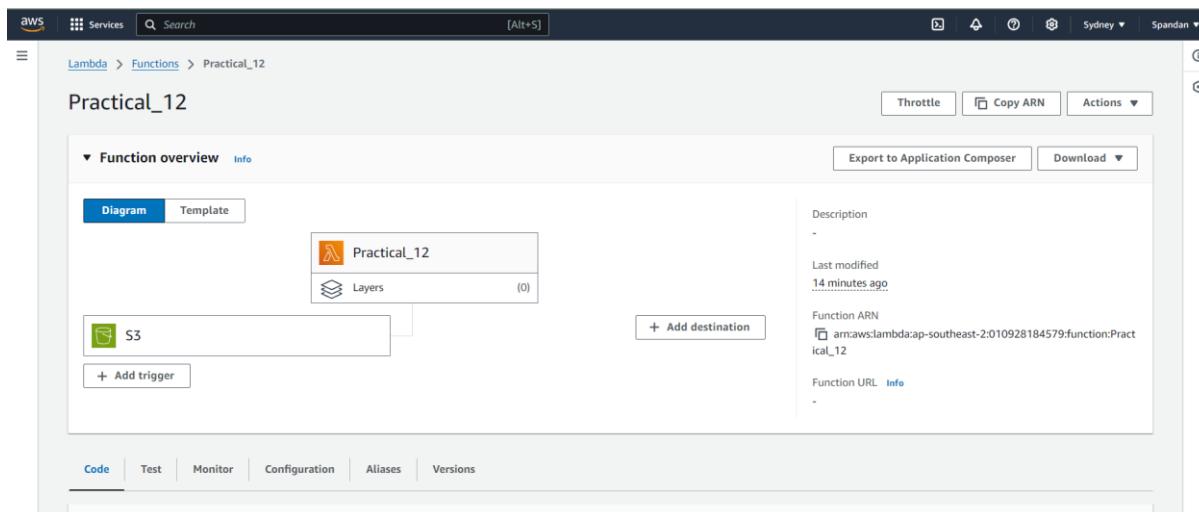
Mention an event name and check Put under event types

The screenshot shows the "Event notifications" configuration page for an S3 bucket. It includes a "Suffix - optional" field containing ".jpg". Under "Event types", it says "Specify at least one event for which you want to receive notifications. For each group, you can choose an event type for all events, or you can choose one or more individual events." The "Object creation" section lists several options, with "Put" checked and "s3:ObjectCreated:Put" selected. Other options include "Post", "s3:ObjectCreated:Post", "Copy", "s3:ObjectCreated:Copy", and "Multipart upload completed", "s3:ObjectCreated:CompleteMultipartUpload".

Choose Lambda function as destination and choose your lambda function and save the changes.



Refresh the Lambda function console and you should be able to see an S3 Trigger in the overview.



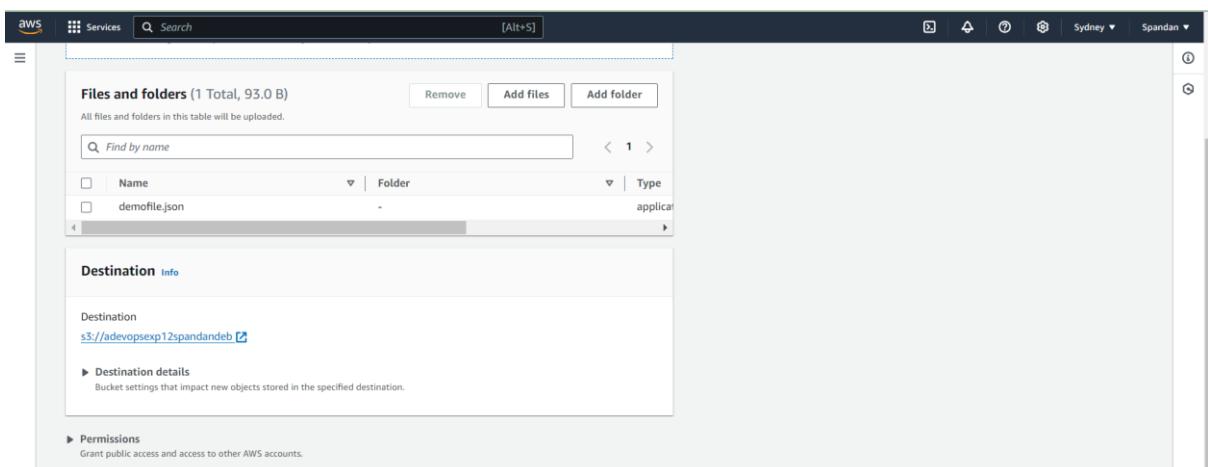
Now, create a demofile JSON file locally.

```
{} demofile.json X

{} demofile.json > ...

1   {
2     "firstname": "SPANDAN",
3     "lastname": "DEB",
4     "gender": "MALE",
5     "age": "20"
6 }
```

Go back to your S3 Bucket and click on Add Files to upload a new file. Select the demofile data file from your computer and click Upload.



After this make the necessary changes in the Test configuration file which we created it previously by replacing the Bucket Name and the ARN of Bucket.

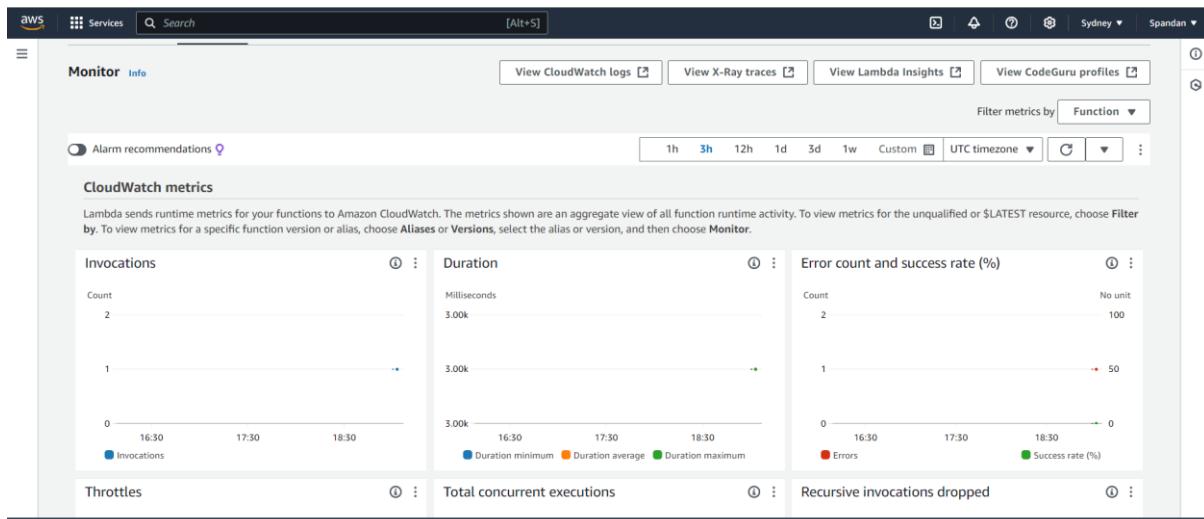
```

1 * {
2 *   "Records": [
3 *     {
4 *       "eventVersion": "2.0",
5 *       "eventSource": "aws:s3",
6 *       "eventSourceARN": "us-east-1",
7 *       "eventName": "ObjectCreated:Put",
8 *       "userIdentity": {
9 *         "principalId": "EXAMPLE"
10 *       },
11 *       "requestParameters": {
12 *         "sourceIPAddress": "127.0.0.1"
13 *       },
14 *       "responseElements": {
15 *         "x-amz-request-id": "EXAMPLE123456789",
16 *         "x-amz-id-2": "EXAMPLE123/5678abcdefghijklmnaaaaaaaaaaaaaaaaaaaaaaa"
17 *       },
18 *       "s3": {
19 *         "configurationVersion": "1.0",
20 *         "configurationId": "testConfigRule",
21 *         "bucket": {
22 *           "name": "addevopsexp12spandandeb",
23 *           "ownerIdentity": {
24 *             "principalId": "EXAMPLE"
25 *           },
26 *           "arn": "arn:aws:s3:::addevopsexp12spandandeb"
27 *         },
28 *         "object": {
29 *           "key": "test%2Fkey",
30 *         }
31 *       }
32 *     }
33 *   ]
34 * }

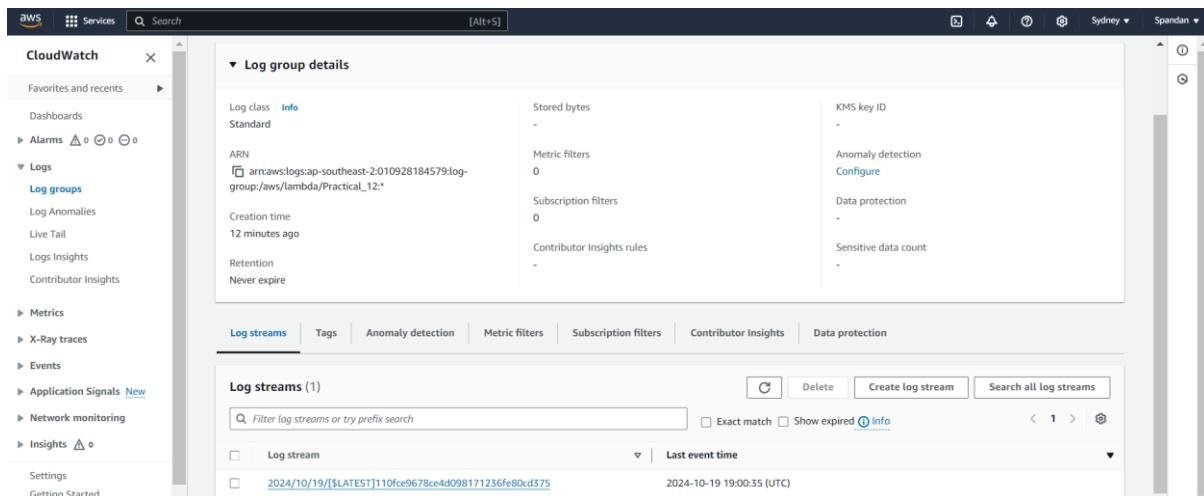
```

27:54 JSON Spaces: 2

Go back to your Lambda function , Refresh it and check the Monitor tab.



Under Log streams, click on View logs in Cloudwatch to check the Function logs.



Click on this log Stream that was created to view what was logged by your function.

The screenshot shows the AWS CloudWatch Log Events interface. On the left, there's a navigation sidebar with options like 'CloudWatch', 'Logs', 'Metrics', 'X-Ray traces', 'Events', 'Application Signals', 'Network monitoring', and 'Insights'. Under 'Logs', 'Log groups' is expanded, showing 'Log Anomalies', 'Live Tail', 'Logs Insights', and 'Contributor Insights'. The main area displays a log stream titled 'Log events' for the path '/aws/lambda/Practical_12'. The log entries are timestamped and show various Lambda runtime events such as INIT, START, REPORT, and END. One entry indicates a file was added to a bucket. The interface includes a search bar, filter buttons for time intervals (Clear, 1m, 30m, 1h, 12h, Custom), and a UTC timezone dropdown.

Timestamp	Message
2024-10-19T19:00:32.329Z	INIT_START Runtime Version: python:3.12.v36 Runtime Version ARN: arn:aws:lambda:ap-southeast-2::runtime:188d9ca2e2714ff5637bd2b0e0ceb01ec3bc408a0..
2024-10-19T19:00:32.681Z	START RequestId: de97b53a-4d55-4c21-8ee3-209c5c164bd2 Version: \$LATEST
2024-10-19T19:00:35.317Z	A file has been added with key demoFile.json to the bucket adevopsexpl2spandande..
2024-10-19T19:00:35.745Z	END RequestId: de07b53a-4d55-4c21-8ee3-209c5c164bd2
2024-10-19T19:00:35.745Z	REPORT RequestId: de07b53a-4d55-4c21-8ee3-209c5c164bd2 Duration: 3000.00 ms Billed Duration: 3000 ms Memory Size: 128 MB Max Memory Used: 82 MB In..
2024-10-19T19:01:42.836Z	INIT_REPORT Init Duration: 3000.11 ms Phase: Invoke Status: timeout
2024-10-19T19:01:42.836Z	START RequestId: de07b53a-4d55-4c21-8ee3-209c5c164bd2 Version: \$LATEST
2024-10-19T19:01:42.883Z	END RequestId: de07b53a-4d55-4c21-8ee3-209c5c164bd2
2024-10-19T19:01:42.883Z	REPORT RequestId: de07b53a-4d55-4c21-8ee3-209c5c164bd2 Duration: 3000.00 ms Billed Duration: 3000 ms Memory Size: 128 MB Max Memory Used: 69 MB St..
2024-10-19T19:03:44.002Z	INIT_REPORT Init Duration: 3000.16 ms Phase: Invoke Status: timeout
2024-10-19T19:03:44.002Z	START RequestId: de07b53a-4d55-4c21-8ee3-209c5c164bd2 Version: \$LATEST