

Exp 1a : Static Hosting

a) Hosting of a PHP file on Local virtual machine using Xampp



b) Static hosting using AWS S3 bucket

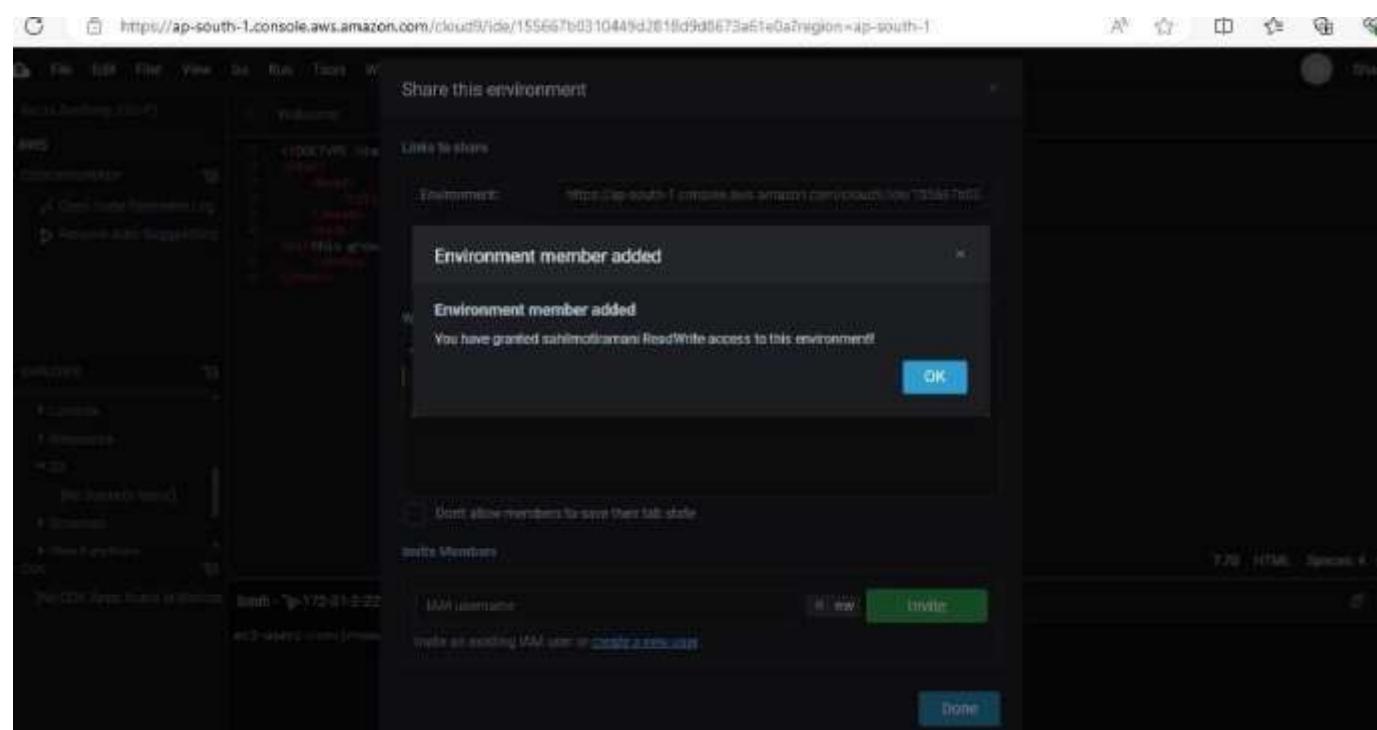
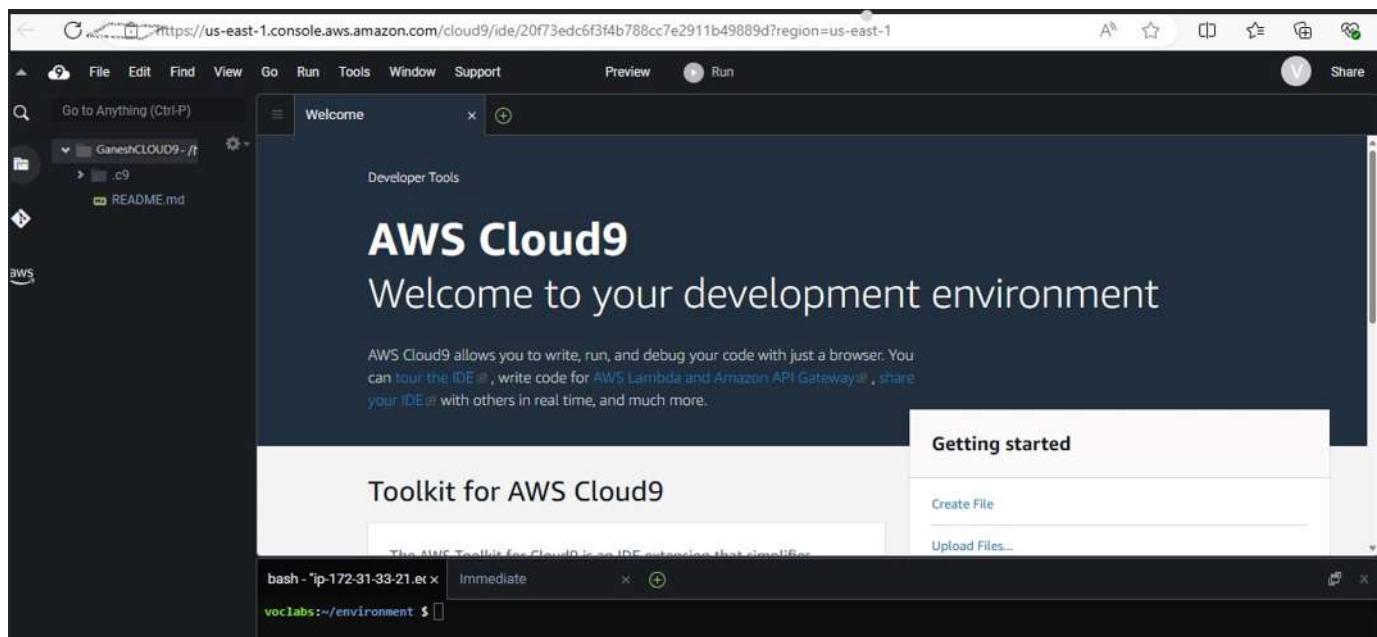
The screenshot shows the AWS S3 Buckets page. At the top, a green banner indicates "Successfully created bucket 'myawsbucket31012004'". Below the banner, there's a section for "Account snapshot - updated every 24 hours" with a "View details" button. The main area displays "General purpose buckets" and "Directory buckets". A table lists one bucket:

Name	AWS Region	IAM Access Analyzer	Creation date
myawsbucket31012004	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 14, 2024, 04:10:14 (UTC +05:30)

In the bottom left corner of the browser window, the text "Hello from ganesh" is displayed.

Experiment No. 1B

CLOUD 9 Environment:



IAM user and User Group:



IAM > Users

Users (1) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

<input type="checkbox"/>	User name	Path	Group:	Last activity	MFA	P
<input type="checkbox"/>	user1	/	0	-	-	-

Step 1: Specify user details

Step 2: Set permissions

Step 3: Review and create

Step 4: Retrieve password

Review and create

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

User details		
User name	Ganesh	Console password type: Custom password
		Require password reset: No

Permissions summary

Name	Type	Used as
AdvanceDevOps_15	Group	Permissions group
AdvanceDevOps_16	Group	Permissions group
AdvDevOpsLab_17	Group	Permissions group

Experiment No 2

Step 1: Choose pipeline settings

Pipeline settings

Pipeline name

ganeshpipeline

Pipeline type

V2

Execution mode

QUEUED

Artifact location

codepipeline-us-east-1-602624750893

Service role name

AWSCodePipelineServiceRole-us-east-1-ganeshpipeline

Source Succeeded

Pipeline execution ID: a23b429a-7039-444c-8386-b03ba43a425f

Source

GitHub (Version 2)



Succeeded - 8 minutes ago

da19c44a

da19c44a Source: Update README.md

Disable transition

Deploy Succeeded

Pipeline execution ID: a23b429a-7039-444c-8386-b03ba43a425f

Deploy

AWS Elastic Beanstalk



Succeeded - 7 minutes ago

da19c44a Source: Update README.md

Congratulations!

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

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

Advanced DevOps Lab

Experiment:3

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

Steps:

1. Create 3 EC2 Ubuntu Instances on AWS.

Instances (1/3) Info											
Find Instance by attribute or tag (case-sensitive)		All states ▼									
Name ▼	Instance ID	Instance state ▼	Instance type ▼	Status check	Alarm status	Availability Zone ▼	Public IPv4 DNS	Public IPv4 ... ▼	Elastic IP	IP	Actions ▼
<input type="checkbox"/> worker-2	i-0565519f0e4fc049	Running View Logs	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-44-203-151-154.co...	44.203.151.154	-	-	Actions
<input checked="" type="checkbox"/> Master	i-06d8202bcbf192d3b	Terminated View Logs	t2.micro	-	View alarms +	us-east-1a	-	-	-	-	Actions
<input type="checkbox"/> worker-1	i-0da69d1cb88393d8b	Running View Logs	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-54-164-90-206.co...	54.164.90.206	-	-	Actions

(Name 1 as Master, the other 2 as worker-1 and worker-2)

- 2. Edit the Security Group Inbound Rules to allow SSH



3. SSH into all 3 machines ssh -i <keyname>.pem

ubuntu@<public_ip_address>

```

The authenticity of host 'ec2-44-203-151-154.compute-1.amazonaws.com (44.203.151.154)' can't be established.
ED25519 key fingerprint is SHA256:4cidwEvWyoqWE0gGMsqDMjX2SlxkZVTUTibDzMdlc.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-203-151-154.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

[ec2-user@ip-172-31-95-91 ~]$ ssh -i "newkey.pem" ec2-user@ec2-54-164-90-206.compute-1.amazonaws.com
Warning: Identity file newkey.pem not accessible: No such file or directory.
The authenticity of host 'ec2-54-164-90-206.compute-1.amazonaws.com (172.31.88.50)' can't be established.
ED25519 key fingerprint is SHA256:RRoSz1NvNq9JLCAhDKUn6F1RCRul+VtNbkjV0SM/I.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-164-90-206.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
[ec2-user@ec2-54-164-90-206.compute-1.amazonaws.com: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[ec2-user@ip-172-31-95-91 ~]$ 

```

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

`sudo yum install docker -y`

```
[ec2-user@ip-172-31-92-18 ~]$ sudo yum install docker -y
Last metadata expiration check: 0:09:56 ago on Wed Sep 11 15:19:39 2024.
Dependencies resolved.
```

Package	Architecture
Installing:	
docker	x86_64
Installing dependencies:	
containerd	x86_64
iptables-libs	x86_64
iptables-nft	x86_64
libcgroup	x86_64
libnetfilter_conntrack	x86_64
libnfnetlink	x86_64
libnftnl	x86_64
pigz	x86_64
runc	x86_64

Transaction Summary

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

- `cd /etc/docker`
- `cat <<EOF | sudo tee /etc/docker/daemon.json`
{
 "exec-opts":
 ["native.cgroupdriver=systemd"], "log-driver":
 "json-file",

```
"log-opt": {  
    "max-size": "100m"  
},  
    "storage-driver": "overlay2"  
}  
EOF
```

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

Install Kubernetes on all 3 machines

SELinux needs to be disable before configuring kubelet

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

```
[ec2-user@ip-172-31-81-63 docker]$ sudo setenforce 0  
[ec2-user@ip-172-31-81-63 docker]$ sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
```

Add kubernetes repository (paste in terminal)

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

Type following commands:

- sudo yum update

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

```
[ec2-user@ip-172-31-81-63 docker]$ sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Last metadata expiration check: 0:01:34 ago on Wed Sep 11 15:39:05 2024.
Dependencies resolved.
=====
Package                                Architecture      Version
=====
Installing:
kubeadm                               x86_64          1.30.4-150500.1.1
kubectl                               x86_64          1.30.4-150500.1.1
kubelet                               x86_64          1.30.4-150500.1.1
Installing dependencies:
comtrack-tools                         x86_64          1.4.6-2.amzn2023.0.2
cri-tools                             x86_64          1.30.1-150500.1.1
kubernetes-cni                         x86_64          1.4.0-150500.1.1
libnetfilter_cthelper                  x86_64          1.0.0-21.amzn2023.0.2
libnetfilter_cttimeout                 x86_64          1.0.0-19.amzn2023.0.2
libnetfilter_queue                      x86_64          1.0.5-2.amzn2023.0.2
socat                                x86_64          1.7.4.2-1.amzn2023.0.2
=====
Transaction Summary
=====
Install 10 Packages
```

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

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

1. Perform this ONLY on the Master machine

Initialize kubernetes by typing below command

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

```
[addons] Applied essential addon: kube-proxy

Your Kubernetes control-plane has initialized successfully!

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

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

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

export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/

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

kubeadm join 172.31.81.63:6443 --token zh5jbb.a6ty3eujzc51d15d \
    --discovery-token-ca-cert-hash sha256:0822f656bf52a17a2b6686c123f811306f41495ca650a0aed9bf6cd2d2f6f8c5
[ec2-user@ip-172-31-81-63 docker]$ mkdir -p $HOME/.kube
[ec2-user@ip-172-31-81-63 docker]$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
[ec2-user@ip-172-31-81-63 docker]$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
[ec2-user@ip-172-31-81-63 docker]$
```

Copy the mkdir and chown commands from the top and execute them

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

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

My personal join key:

```
kubeadm join 172.31.92.157:6443 --token x4sw6q.sbckmh5gkoubquv \
    --discovery-token-ca-cert-hash
sha256:24c005691fcab2260667ee43384d46af4b2b27401e82c01550798a0d8f98950
```

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

D15C

4

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

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

Create 3 EC2 Ubuntu Instances on AWS.

Login to your AWS console.Go to services and in that search for EC2 and create 3 EC2 Ubuntu Instances as master 1 ,node1 and node 2.While making an instance make sure to select Amazon Linux and in linux type instead of default t2 .micro select t2.medium.

The screenshot shows the 'Name and tags' step of the AWS EC2 instance creation wizard. It includes fields for 'Name' (set to 'master1') and 'Add additional tags'. Below this is the 'Application and OS Images (Amazon Machine Image)' section, which contains a search bar and a grid of AMI icons for Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE Linux. A 'Quick Start' tab is selected. The 'Instance type' section shows 't2.medium' selected, with detailed pricing information for On-Demand instances of various families. The 'Key pair (login)' section shows 'vokey' selected as the key pair name. A note at the bottom states: 'You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.'

Name and tags [Info](#)

Name
master1 [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

Recent [Quick Start](#)

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

▼ Instance type [Info](#) | [Get advice](#)

Instance type
t2.medium

Family: t2 2 vCPU 4 GiB Memory Current generation: true

On-Demand Linux base pricing: 0.0464 USD per Hour

On-Demand RHEL base pricing: 0.0752 USD per Hour

On-Demand Windows base pricing: 0.0644 USD per Hour

On-Demand SUSE base pricing: 0.1464 USD per Hour

All generations

[Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

▼ Key pair (login) [Info](#)

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

Key pair name - required
vokey [Create new key pair](#)

Setting SSH for establishing connections

```
$ ssh -i "test.pem" ec2-user@ec2-3-88-211-185.compute-1.amazonaws.com
The authenticity of host 'ec2-3-88-211-185.compute-1.amazonaws.com (3.88.211.185)' can't be established.
ED25519 key fingerprint is SHA256:lvde44+eLezx0A07MTKiy+0c1Eh+1PRO8C28aKA2yE0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-88-211-185.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

[ec2-user@ip-172-31-25-142 ~]$ |
```

INSTALLATION OF DOCKER

For installing docker we use the following steps:

STEP 1: In node 1 EC2 instance install docker and repeat the same step for master and node2

Syntax: yum install docker -y

```
[root@ip-172-31-21-176 ec2-user]# yum install docker -y
Last metadata expiration check: 0:21:25 ago on Fri Sep 13 17:05:55 2024.
Dependencies resolved.
=====
| Package           | Architecture | Version      | Repository | Size   |
|:-----|:-----|:-----|:-----|:-----|
| Installing:
|   docker          | x86_64       | 25.0.6-1.amzn2023.0.2 | amazonlinux | 44 M  |
| Installing dependencies:
|   containerd      | x86_64       | 1.7.20-1.amzn2023.0.1 | amazonlinux | 35 M  |
|   iptables-libc   | x86_64       | 1.8.8-3.amzn2023.0.2 | amazonlinux | 401 k |
|   iptables-nft    | x86_64       | 1.8.8-3.amzn2023.0.2 | amazonlinux | 183 k |
|   libcgroup        | x86_64       | 3.0-1.amzn2023.0.1   | amazonlinux | 75 k  |
|   libnethelper_conntrack | x86_64       | 1.0.8-2.amzn2023.0.2 | amazonlinux | 58 k  |
|   libnfnetlink     | x86_64       | 1.0.1-19.amzn2023.0.2 | amazonlinux | 30 k  |
|   libnfnetlink     | x86_64       | 1.2.2-2.amzn2023.0.2 | amazonlinux | 84 k  |
|   pigz             | x86_64       | 2.5-1.amzn2023.0.3   | amazonlinux | 83 k  |
|   runc             | x86_64       | 1.1.13-1.amzn2023.0.1| amazonlinux | 3.2 M |
=====
Transaction Summary
=====
Install 10 Packages
```

i-0defb5859fc2b0488 (node1)

PublicIPs: 54.157.60.252 PrivateIPs: 172.31.21.176

STEP 2: After the installation of docker is successfully completed in all the three instances start the docker by the syntax given below:

Syntax : systemctl start docker.

Start the docker in master and node2 too .

```

Running scriptlet: docker-25.0.6-1.amzn2023.0.2.x86_64
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.

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

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

Complete!
[root@ip-172-31-21-176 ec2-user]# systemctl start docker
[root@ip-172-31-21-176 ec2-user]# [x]

i-0defb5859fc2b0488 (node1)
PublicIPs: 54.157.60.252 PrivateIPs: 172.31.21.176

```

INSTALLATION OF KUBERNETES

After installing and starting the docker in all the three instances ,now lets install kubernetes for the installation we use the following steps:

STEP 1: Set SELinux to permissive mode:

Syntax: `sudo setenforce 0`

```

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

```

```

[root@ip-172-31-25-172 docker]# sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config

```

STEP 2: Add the Kubernetes yum repository. The exclude parameter in the repository definition ensures that the packages related to Kubernetes are not upgraded upon running yum update as there's a special procedure that must be followed for upgrading Kubernetes

```
[root@ip-172-31-21-176 ec2-user]# sudo su
[root@ip-172-31-21-176 ec2-user]# yum repolist
repo id                                repo name
amazonlinux                             Amazon Linux 2023 repository
kernel-livepatch                         Amazon Linux 2023 Kernel Livepatch repository
[root@ip-172-31-21-176 ec2-user]# cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/repo/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
EOF
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/repo/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
[root@ip-172-31-21-176 ec2-user]# 
```

i-0defb5859fc2b0488 (node1)
PublicIPs: 54.157.60.252 PrivateIPs: 172.31.21.176

STEP 3:Install kubelet, kubeadm and kubectl:

Syntax:`sudo yum install -y kubelet kubeadm kubectl
--disables excludes=kubernetes`

```
Last login: Fri Sep 13 17:58:28 2024 from 18.206.107.27
[ec2-user@ip-172-31-21-176 ~]$ sudo su
[root@ip-172-31-21-176 ec2-user]# sudo yum install -y kubelet kubeadm kubectl --disables excludes=kubernetes
Dependencies resolved.
60 kB/s | 9.4 kB 00:00
=====
Package           Architecture Version       Repository      Size
=====
Installing:
kubeadm          x86_64     1.31.1-150500.1.1   kubernetes    11 M
kubectl          x86_64     1.31.1-150500.1.1   kubernetes    11 M
kubelet          x86_64     1.31.1-150500.1.1   kubernetes    15 M
Installing dependencies:
cni-tools         x86_64     1.4.6-2.amzn2023.0.2  amazonlinux   208 k
cri-tools         x86_64     1.31.1-150500.1.1   kubernetes    6.9 M
kubernetes-cni   x86_64     1.5.1-150500.1.1   kubernetes    7.1 M
libnetfilter_cthelper x86_64   1.0.0-21.amzn2023.0.2  amazonlinux   24 k
libnetfilter_cttimeout x86_64   1.0.0-19.amzn2023.0.2  amazonlinux   24 k
libnetfilter_queue x86_64   1.0.5-2.amzn2023.0.2   amazonlinux   30 k
=====
Transaction Summary
=====
Install 9 Packages
```

STEP 4:Enable the kubelet service before running kubeadm:

Syntax:`sudo systemctl enable --now kubelet`

```
[root@ip-172-31-21-176 ec2-user]# sudo systemctl enable --now kubelet
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /usr/lib/systemd/system/kubelet.service.
[root@ip-172-31-21-176 ec2-user]# 
```

i-0defb5859fc2b0488 (node1)
PublicIPs: 54.157.60.252 PrivateIPs: 172.31.21.176

STEP 5:It can be seen from the repolist command which lists all the repository we can see that kubernetes is installed repeat all these steps on master1 and node2.

```
[root@ip-172-31-21-176 ec2-user]# yum repolist
repo id                                repo name
amazonlinux                             Amazon Linux 2023 repository
kernel-livepatch                         Amazon Linux 2023 Kernel Livepatch repository
kubernetes                               Kubernetes
[root@ip-172-31-21-176 ec2-user]# [REDACTED]
i-0defb5859fc2b0488 (node1)
PublicIPs: 54.157.60.252  PrivateIPs: 172.31.21.176
```

STEP 6 :This command disable swap space and configure the system to use iptables for bridged network traffic, then apply these settings.

Syntax:`sudo swapoff -a`

```
echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf
sudo sysctl -p
[root@ip-172-31-16-56 ec2-user]# sudo swapoff -a
echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf
sudo sysctl -p
net.bridge.bridge-nf-call-iptables=1
```

STEP 7: Initialize Kubernetes in master instance .

Syntax: `kubeadm init`

```
[root@ip-172-31-16-56 ec2-user]# kubeadm init
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
  [WARNING FileExisting-socat]: socat not found in system path
  [WARNING FileExisting-tc]: tc not found in system path
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
W0913 18:58:26.902514 34809 checks.go:846] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent
with that used by kubeadm. It is recommended to use "registry.k8s.io/pause:3.10" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-16-56.ec2.internal kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 172.31.16.56]
[certs] Generating "apiserver-kubelet-client" certificate and key
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
W0913 18:58:26.902514 34809 checks.go:846] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent
with that used by kubeadm. It is recommended to use "registry.k8s.io/pause:3.10" as the CRI sandbox image.
[certs] 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-172-31-16-56.ec2.internal localhost] and IPs [172.31.16.56 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-16-56.ec2.internal localhost] and IPs [172.31.16.56 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
```

i-0ddff50a232db19957 (master1)

PublicIPs: 3.88.204.138 PrivateIPs: 172.31.16.56

```
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 172.31.16.56:6443 --token oghyi3.fnspdro8pevgr0d5 \
--discovery-token-ca-cert-hash sha256:ec71ffc0d9fd79263fb8909d938da8d29e5f15a21ab5e0a17ec93514e8c4ecb8
```

Use the mkdir and chown commands shown above

```
[root@ip-172-31-16-56 ec2-user]# mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

Add a common networking plugin called flannel

Syntax: kubectl apply -f

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

```
[root@ip-172-31-16-56 ~]# kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
```

STEP 8: Apply deployment of nginx server using the following command.

Syntax:

```
kubectl apply -f https://k8s.io/examples/application/deployment.yaml
[root@ip-172-31-16-56 ~]# kubectl apply -f https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
```

Check whether the pods is created or not by the following command

Syntax: kubectl get pods

```
[root@ip-172-31-16-56 ~]# kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-gw8v8   0/1    Pending   0          82s
nginx-deployment-d556bf558-rfk9n   0/1    Pending   0          82s
```

Kubectl describe pod nginx command describe the pods.

```
[root@ip-172-31-16-56 ~]# kubectl describe pod nginx
Name:           nginx-deployment-d556bf558-gw8v8
Namespace:      default
Priority:       0
Service Account: default
Node:           <none>
Labels:          app=nginx
                  pod-template-hash=d556bf558
Annotations:    <none>
Status:         Pending
IP:
IPs:            <none>
Controlled By: ReplicaSet/nginx-deployment-d556bf558
Containers:
  nginx:
    Image:        nginx:1.14.2
    Port:         80/TCP
    Host Port:   0/TCP
    Environment: <none>
    Mounts:
```

```
Conditions:
  Type      Status
  PodScheduled  False
Volumes:
  kube-api-access-f9k9s:
    Type:      PersistentVolumeToken
    Status:    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:      kube-root-ca.crt
    ConfigMapOptional:  <nil>
    DownwardAPI:       true
    QoS Class:        BestEffort
    Node-Selectors:   <none>
    Tolerations:     node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                      node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type      Reason     Age   From           Message
  ----      ----     --   --            --
  Warning   FailedScheduling 114s  default-scheduler 0/1 nodes are available: 1 node(s) had untolerated taint {node-role.kubernetes.io/control-plane}: 0/1 nodes are available: 1 Preemption is not helpful for scheduling.
  Warning   FailedScheduling 3m18s default-scheduler 0/1 nodes are available: 1 node(s) had untolerated taint {node-role.kubernetes.io/control-plane}: 0/1 nodes are available: 1 Preemption is not helpful for scheduling.
```

```
[ec2-user@ip-172-31-26-174 ~]$ kubectl taint nodes --all node-role.kubernetes.io/control-plane-
node/ip-172-31-26-174.ec2.internal untainted
```

STEP 9:Check whether the pods are running or not.

```
[ec2-user@ip-172-31-26-174 ~]$ kubectl get pods
NAME    READY    STATUS    RESTARTS   AGE
nginx  1/1     Running   1 (6s ago)  90s
```

STEP 10:Mention the port that you want to host on

Syntax:port-forward nginx 8081:80

```
[ec2-user@ip-172-31-26-174 ~]$ kubectl port-forward nginx 8081:80
Forwarding from 127.0.0.1:8081 -> 80
Forwarding from [::1]:8081 -> 80
```

STEP 11:Then verify your deployment

Conclusion :In this experiment we have created 3 EC2 instances. Setting SSH for establishing connections in that we have installed and started docker and kubernetes ,initialising kubernetes we use the mkdir and chown commands that we get by initializing the kubertenes then we add a common networking plugin called flannel then we apply deployment to nginx server we describe the pods and we check the status of the pods we mention the port that we want to host on and at the end we very the deployment of the kubernetes application by performing the following steps we learned to deploy the our Kubernetes Application.

Name: Ganesh Gupta

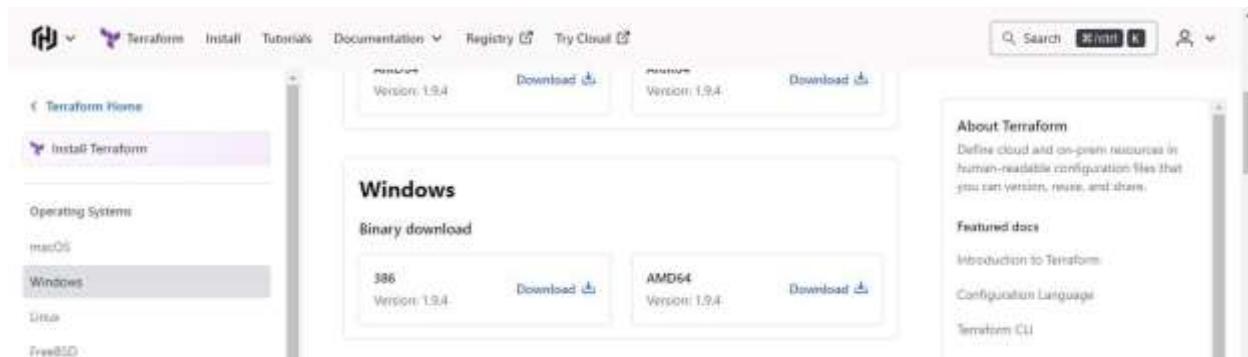
D15C

Roll No: 15

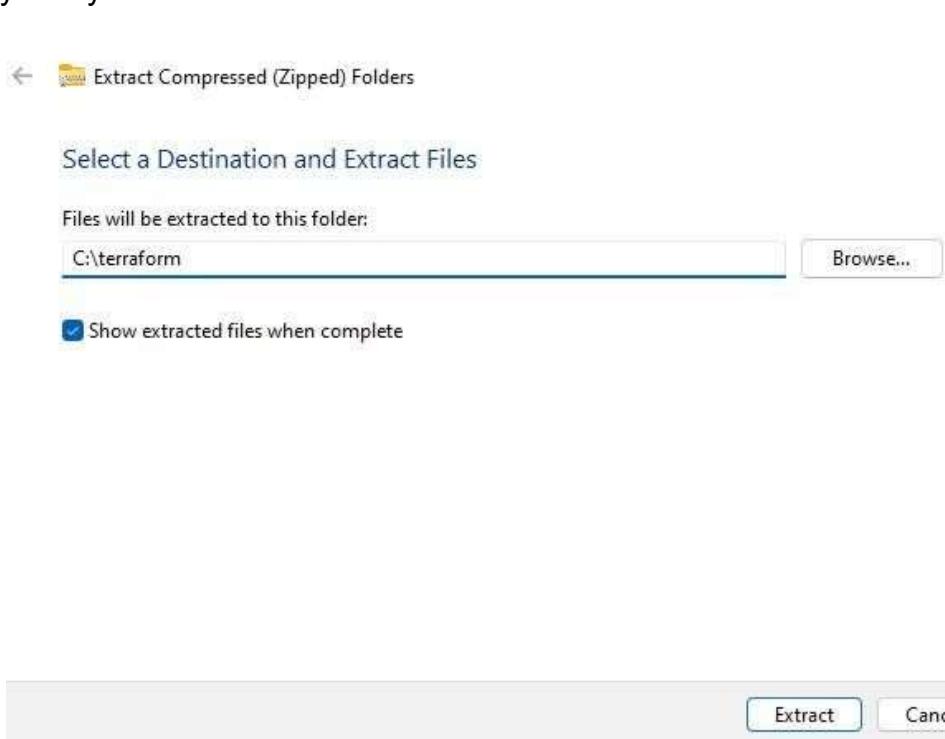
Experiment No. 5

Installation of Terraform

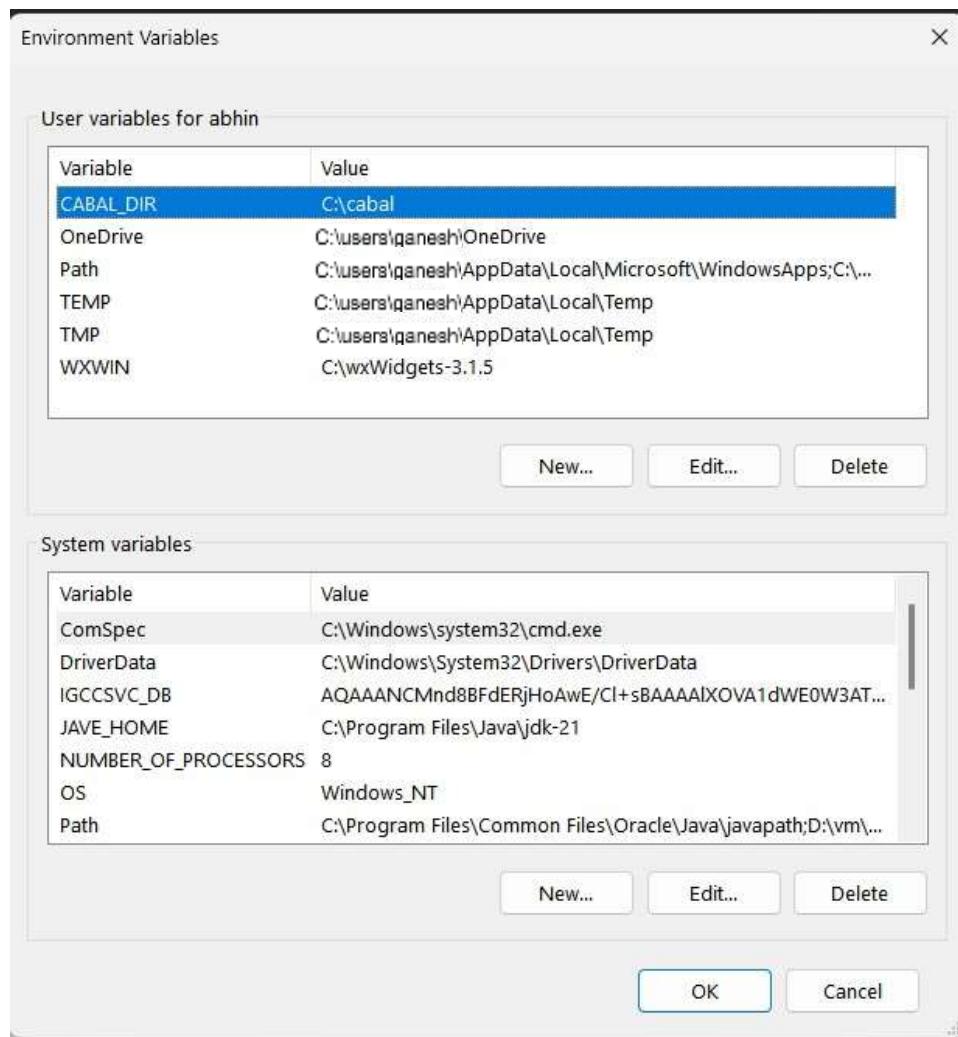
Step 1: To install Terraform, visit the official Terraform website mentioned below, go to the Downloads section, select Windows, and download the 64-bit version for your system. website: <https://www.terraform.io/downloads.html>



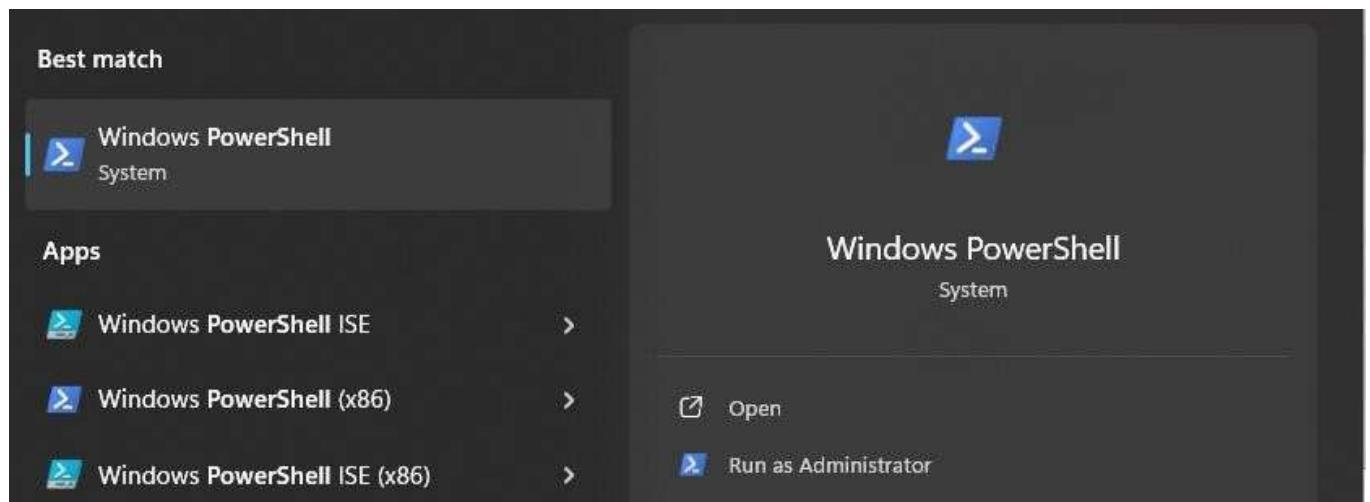
Step 2: Extract the downloaded 'Terraform.exe' file to the 'C:\Terraform' directory on your system.



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



Step 4: Run the windows powershell as administrator.



Step 5: Run “terraform” to verify its functionality. If you encounter any errors, double-check or update the Terraform path in your environment variables.

```
Administrator: Windows Pow X + 
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

PS C:\Users\abhin> 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
```

Creating docker image using terraform

Step 1: Download and install Docker Desktop by visiting <https://www.docker.com>. Run the installer and follow the prompts to complete the installation, then verify by launching Docker Desktop or using the `docker --version` command.

```
C:\Users\Admin>docker

Usage: docker [OPTIONS] COMMAND
A self-sufficient runtime for containers

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

Management Commands:
  builder  Manage builds
  buildx* Docker Buildx
  compose* Docker Compose
  container Manage containers
  context   Manage contexts
  debug*   Get a shell into any image or container
  desktop* Docker Desktop commands (Alpha)
  dev*     Docker Dev Environments
  extension* Manages Docker extensions
  feedback* Provide feedback, right in your terminal!
  image    Manage images
```

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

Step 2: Now, create a folder named ‘Terraform Scripts’ in which we save our different types of scripts which will be further used in this experiment.

Step 3: First, create a new folder named `Docker` inside the `TerraformScripts` folder. Then, open Notepad and create a new file named `docker.tf` within the `Docker` folder. Write the following contents into the `docker.tf` file to create an Ubuntu Linux container. Save the file when done.

Script:

```
terraform { required_providers {  
    docker = { source =  
        "kreuzwerker/docker" version =  
        "2.21.0"  
    }  
}  
  
provider "docker" {  
    host = "npipe:///./pipe/docker_engine"  
}  
  
# Pulls the image resource  
"docker_image" "ubuntu" {  
    name = "ubuntu:latest"  
}  
  
# Create a container resource  
"docker_container" "foo" {  
    image = docker_image.ubuntu.image_id  
    name      = "foo"  
    command = ["sleep", "infinity"]  
}
```

This Terraform script configures the Docker provider to communicate with the Docker Engine using a Windows named pipe.

It pulls the latest Ubuntu image from Docker Hub and creates a container named "foo."

The container runs the `sleep infinity` command, which keeps it active indefinitely.

This setup is useful for scenarios where the container needs to remain running continuously.

```

|terraformer {
  required_providers {
    docker = {
      source  = "kreuzwerker/docker"
      version = "2.21.0"
    }
  }
}

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

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

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

```

Step 4: Execute the `terraform init` command to initialize the working directory, download the necessary provider plugins, and set up the backend for managing Terraform state.

```

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

```

Terraform has been successfully initialized!

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

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

```
PS C:\Users\Admin\Desktop\Terraformscripts\docker> |
```

Step 5: Run `terraform plan` to preview the actions Terraform will take to reach the desired state defined in your configuration, including creating, modifying, or deleting resources.

```
PS C:\Users\Admin\Desktop\Terraformscripts\docker> terraform plan

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

Terraform will perform the following actions:

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

    + read_only      = false
    + remove_volumes = true
    + restart         = "no"
    + rm              = false
    + runtime         = (known after apply)
    + security_opts  = (known after apply)
    + shm_size        = (known after apply)
    + start           = true
    + 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.

Note: You didn't use the `-out` option to save this plan, so Terraform can't guarantee to take exactly these actions if you run `"terraform apply"` now.

PS C:\Users\Admin\Desktop\Terraformscripts\docker> |

Step 6: Execute “**terraform apply**” to apply the configuration, which will automatically create and run the Ubuntu container based on our configuration.

```
PS C:\Users\Admin\Desktop\Terraformscripts\docker> terraform apply

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

Terraform will perform the following actions:

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

+ read_only      = false
+ remove_volumes = true
+ restart        = "no"
+ rm             = false
+ runtime         = (known after apply)
+ security_opts  = (known after apply)
+ shm_size        = (known after apply)
+ start          = true
+ stdin_open     = false
+ stop_signal    = (known after apply)
+ stop_timeout   = (known after apply)
+ tty             = false

+ healthcheck (known after apply)

+ labels (known after apply)
}

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

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

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

Enter a value: yes

docker_image.ubuntu: Creating...
docker_image.ubuntu: Still creating... [10s elapsed]
docker_image.ubuntu: Still creating... [20s elapsed]
docker_image.ubuntu: Still creating... [30s elapsed]
docker_image.ubuntu: Creation complete after 37s [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Creating...
docker_container.foo: Creation complete after 2s [id=76c6390ec277dc11f709997c4eb636ee8b45f90723d0f6ec07d85caeda18ead9]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
PS C:\Users\Admin\Desktop\Terraformscripts\docker>
```

Step 7: The command `docker images` lists all Docker images stored locally on your system, showing details like repository names, tags, image IDs, and creation dates.

Docker images, Before Executing Apply step:

PS C:\Users\Admin\Desktop\Terraformscripts\docker> docker images				
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE

Docker images, After Executing Apply step:

PS C:\Users\Admin\Desktop\Terraformscripts\docker> docker images				
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ubuntu	latest	edbfe74c41f8	3 weeks ago	78.1MB

Step 8: Execute Terraform destroy to delete the configuration, which will automatically delete the Ubuntu Container.

```
PS C:\Users\Admin\Desktop\Terraformscripts\docker> terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Refreshing state... [id=76c6390ec277dc11f709997c4eb636ee8b45f90723d0f6ec07d85caeda18ead9]

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

Terraform will perform the following actions:

# docker_container.foo will be destroyed
- resource "docker_container" "foo" {
  - attach          = false -> null
  - command        = [
    - "sleep",
    - "infinity",
  ] -> null
  - cpu_shares     = 0 -> null
  - dns            = [] -> null
  - dns_opts       = [] -> null
  - dns_search     = [] -> null
  - entrypoint     = [] -> null
  - env            = [] -> null
  - gateway        = "172.17.0.1" -> null
  - group_add      = [] -> null
  - hostname       = "76c6390ec277" -> null
  - id             = "76c6390ec277dc11f709997c4eb636ee8b45f90723d0f6ec07d85caeda18ead9" -> null
  - image          = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> null
  - init           = false -> null
  - ip_address     = "172.17.0.2" -> null
  - ip_prefix_length = 16 -> null
  - ipc_node       = "private" -> null
  - links          = [] -> null
  - log_driver     = "json-file" -> null
  - log_opts        = {} -> null
  - logs           = false -> null
  - max_retry_count = 0 -> null
  - memory          = 0 -> null
  - memory_swap     = 0 -> null
  - must_run        = true -> null
  - name            = "foo" -> null
  - network_data    = [
    - {
      - gateway        = "172.17.0.1"
      - global_ipv6_prefix_length = 0
      - ip_address     = "172.17.0.2"
      - ip_prefix_length = 16
      - network_name   = "bridge"
    # (2 unchanged attributes hidden)
  ]
}
```

```

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

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

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

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

Enter a value: yes

docker_container.foo: Destroying... [id=76c6390ec277dc11f709997c4eb636ee8b45f90723d0f6ec07d85caeda18ead9]
docker_container.foo: Destruction complete after 1s
docker_image.ubuntu: Destroying... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_image.ubuntu: Destruction complete after 1s

Destroy complete! Resources: 2 destroyed.
PS C:\Users\Admin\Desktop\Terraformsscripts\docker> |

```

Step 9: Docker images After Executing Destroy step

```

PS C:\Users\Admin\Desktop\Terraformsscripts\docker> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE

```

Adv DevOps Practical 7

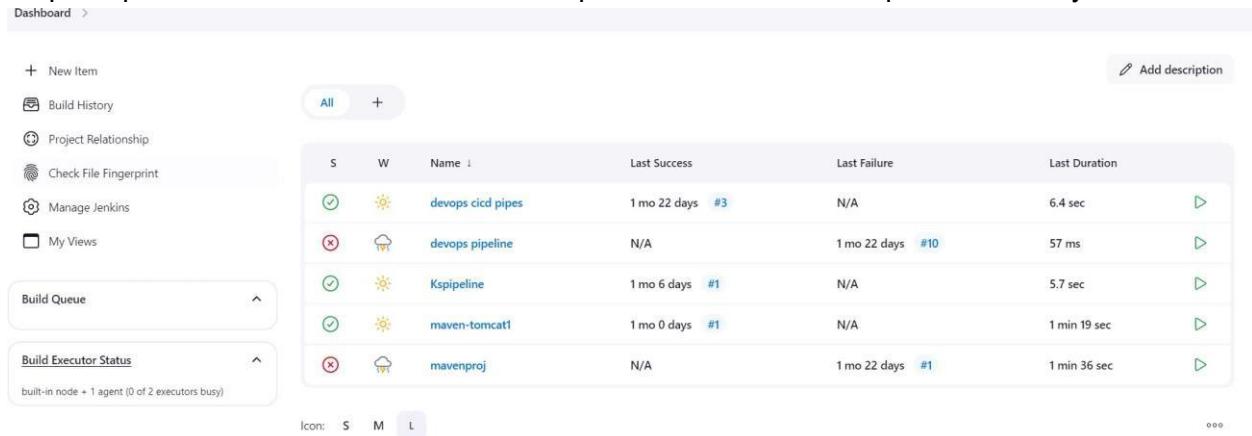
Aim: To understand Static Analysis SAST process and learn to integrate Jenkins SAST to SonarQube/GitLab.

Integrating Jenkins with SonarQube:

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

Steps to integrate Jenkins with SonarQube

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



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

- Left Sidebar:** Contains links for "New Item", "Build History", "Project Relationship", "Check File Fingerprint", "Manage Jenkins", and "My Views". It also shows sections for "Build Queue" and "Build Executor Status".
- Top Bar:** Includes a "Dashboard" link, a search bar, and a "Add description" button.
- Main Content Area:** A table listing Jenkins projects. The columns are: Status (S), Warning (W), Name, Last Success, Last Failure, and Last Duration.

S	W	Name	Last Success	Last Failure	Last Duration
Green	Sunny	devops cicd pipes	1 mo 22 days #3	N/A	6.4 sec
Red	Cloudy	devops pipeline	N/A	1 mo 22 days #10	57 ms
Green	Sunny	Kspipeline	1 mo 6 days #1	N/A	5.7 sec
Green	Sunny	maven-tomcat1	1 mo 0 days #1	N/A	1 min 19 sec
Red	Cloudy	mavenproj	N/A	1 mo 22 days #1	1 min 36 sec

At the bottom, there are icons for "Icon: S M L" and a "More" button.

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

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

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

```
PS C:\Users\91773\Desktop\College Resources\Exp7 adv devops> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
77e678cded2e-f5f989912d3d9e6991dd548eac03faa1eed68dd906614be53acc
PS C:\Users\91773\Desktop\College Resources\Exp7 adv devops>
```

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



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

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

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

1 of 2

Create a local project

Project display name *

exp7



Project key *

exp7



Main branch name *

main

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

Cancel

Next

Setup the project and come back to Jenkins Dashboard.

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

The screenshot shows the Jenkins 'Manage Jenkins > Plugins' page. A search bar at the top contains the text 'sonar'. Below the search bar, there are three tabs: 'Updates' (with 30 items), 'Available plugins' (selected), 'Installed plugins', and 'Advanced settings'. The search results list three items:

- SonarQube Scanner 2.17.2**: Released 7 months ago. Description: This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.
- Sonar Quality Gates 315.v1ff12b_e81a_3a_4**: Released 29 days ago. Description: Library plugins (for use by other plugins) analysis Other Post-Build Actions. Fails the build whenever the Quality Gates criteria in the Sonar 5.6+ analysis aren't met (the project Quality Gates status is different than "Passed")
- Quality Gates 2.5**: Fails the build whenever the Quality Gates criteria in the Sonar analysis aren't met (the project Quality Gates status is different than "Passed")

- 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.

In SonarQube installations: Under **Name** add <project name of sonarqube> for me **sahilexp7** In

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

The screenshot shows the 'SonarQube servers' configuration page. It includes sections for 'Environment variables' (checked), 'SonarQube installations' (list of installations), 'Name' (set to 'exp7'), 'Server URL' (Default is http://localhost:9000, set to http://localhost:9000), 'Server authentication token' (dropdown menu with '- none -' selected, plus an '+ Add' button), and an 'Advanced' section.

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

Dashboard > Manage Jenkins > Tools

The screenshot shows the Jenkins 'Tools' configuration page. It includes sections for:

- Gradle installations:** Includes an 'Add Gradle' button.
- SonarScanner for MSBuild installations:** Includes an 'Add SonarScanner for MSBuild' button.
- SonarQube Scanner installations:** Includes an 'Add SonarQube Scanner' button.
- Ant installations:** Includes an 'Add Ant' button.

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

The screenshot shows the configuration dialog for the SonarQube Scanner tool. It includes fields for:

- Name:** sonarqube_exp7
- Install automatically:** A checked checkbox.
- Install from Maven Central:** A section containing:
 - Version:** SonarQube Scanner 6.2.0.4584
 - Add Installer:** A dropdown menu.

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

New Item

Enter an item name

Select an item type



Freestyle project

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



Maven project

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



Pipeline

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



Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple

OK

9. Choose this GitHub repository in Source Code Management.

https://github.com/shazforiot/MSBuild_firstproject.git

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

Dashboard > exp7 > Configuration

Source Code Management

None

Git [?](#)

Repositories [?](#)

Repository URL [?](#)

Credentials [?](#)

- none -

[+ Add](#)

[Advanced](#)

[Add Repository](#)

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

The screenshot shows the Jenkins 'Configuration' screen for a project named 'exp7'. A dropdown menu is open under 'Add build step' containing various options like 'Execute SonarQube Scanner', 'Execute Windows batch command', etc. Below this, the 'Post-build Actions' section is visible with an 'Add post-build action' dropdown. The main configuration area is expanded to show the 'Execute SonarQube Scanner' step. It includes fields for 'JDK' (set to 'JDK 17'), 'Path to project properties' (empty), 'Analysis properties' (containing project-specific configuration), and 'Additional arguments' (empty). At the bottom are 'Save' and 'Apply' buttons.

Dashboard > exp7 > Configuration

Filter

- Execute SonarQube Scanner
- Execute Windows batch command
- Execute shell
- Invoke Ant
- Invoke Gradle script
- Invoke top-level Maven targets
- Run with timeout
- Set build status to "pending" on GitHub commit
- SonarScanner for MSBuild - Begin Analysis
- SonarScanner for MSBuild - End Analysis

Add build step ^

Post-build Actions

Add post-build action ▾

Save Apply

Execute SonarQube Scanner

JDK ?
JDK to be used for this SonarQube analysis
JDK 17

Path to project properties ?
[Empty input field]

Analysis properties ?
sonar.projectKey=ks_exp7
sonar.projectName=ks_exp7
sonar.projectVersion=1.0
sonar.sources=C:/ProgramData/Jenkins/jenkins/workspace/ks_exp7
sonar.host.url=http://localhost:9000
sonar.login=admin
sonar.password=kshitij24

Additional arguments ?
[Empty input field]

Dashboard > ks_exp7 >

Status

- </> Changes
- Workspace
- Build Now
- Configure
- Delete Project
- SonarQube
- Rename

ks_exp7

SonarQube

Permalinks

- Last build (#7), 4 min 55 sec ago
- Last stable build (#7), 4 min 55 sec ago
- Last successful build (#7), 4 min 55 sec ago
- Last failed build (#6), 17 min ago
- Last unsuccessful build (#6), 17 min ago
- Last completed build (#7), 4 min 55 sec ago

Build History

trend ▾

Filter... /

#7 Sep 25, 2024, 3:09 PM

Console Output

Started by user Kshitij Hundre
Running as SYSTEM
Building on the built-in node in workspace C:\ProgramData\Jenkins\.jenkins\workspace\ks_exp7
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\ks_exp7\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/shazforiot/MSBuild_firstproject.git # timeout=10
Fetching upstream changes from https://github.com/shazforiot/MSBuild_firstproject.git
> git.exe --version # timeout=10
> git --version # 'git version 2.46.0.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/shazforiot/MSBuild_firstproject.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision f2bc042c04c6e72427c380bcace6d6fee7b49adf (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f f2bc042c04c6e72427c380bcace6d6fee7b49adf # timeout=10
Commit message: "updated"
> git.exe rev-list --no-walk f2bc042c04c6e72427c380bcace6d6fee7b49adf # timeout=10
[ks_exp7] \$ C:\ProgramData\Jenkins\.jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube1_exp7\bin\sonar-scanner.bat -Dsonar.host.url=http://localhost:9000 -Dsonar.projectKey=ks_exp7 -Dsonar.projectName=ks_exp7 -Dsonar.host.url=http://localhost:9000 -Dsonar.login=admin -Dsonar.projectVersion=1.0 -Dsonar.sources=C:\ProgramData\Jenkins\.jenkins\workspace\ks_exp7 -Dsonar.password=kshitij24 -Dsonar.projectBaseDir=C:\ProgramData\Jenkins\.jenkins\workspace\ks_exp7
15:09:08.473 WARN Property 'sonar.host.url' with value 'http://localhost:9000' is overridden with value 'http://localhost:9000'

Download **Copy** **View as plain**

11. Go to http://localhost:9000/<user_name>/permissions and allow Execute Permissions to

the Admin user.

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

4 of 4 shown

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

The screenshot shows the SonarQube interface for a project named 'main'. At the top, there's a navigation bar with tabs like Overview, Issues, Security Hotspots, Measures, Code, and Activity. On the right, there are Project Settings and Project Information dropdowns. The main content area displays the project name 'main' and its version 'Version 1.0'. A prominent green checkmark icon indicates that the 'Quality Gate' has passed. A yellow warning box states: 'The last analysis has warnings. See details'. Below this, there are three performance cards: 'Security' (0 Open issues), 'Reliability' (0 Open issues), and 'Maintainability' (0 Open issues). Each card has a green 'A' icon. The overall status is 'Passed'.

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

Conclusion:

In this project, we integrated Jenkins with SonarQube for automated static application security testing (SAST). We set up SonarQube using Docker, configured Jenkins with the necessary plugins and authentication, and linked it to a GitHub repository. The SonarQube scanner was added as a build step, enabling continuous code analysis for vulnerabilities, code smells, and quality issues, ensuring automated reporting and continuous code quality improvement.

Expt No. 08 Advanced DevOps Lab

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

Theory:

What is SAST?

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

What problems does SAST solve?

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

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

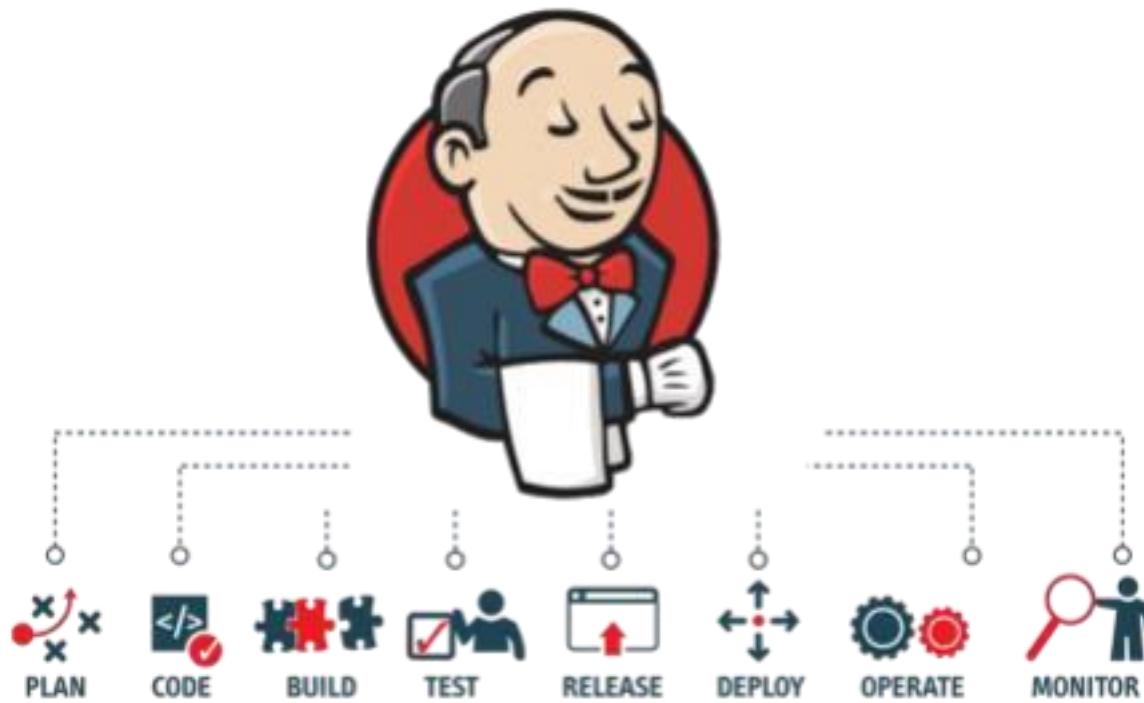
It's important to note that SAST tools must be run on the application on a regular basis, such as during daily/monthly builds, every time code is checked in, or during a code release.

Why is SAST important?

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

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

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



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

What is SonarQube?

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

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

Benefits of SonarQube

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

Integrating Jenkins with SonarQube:

Prerequisites:

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

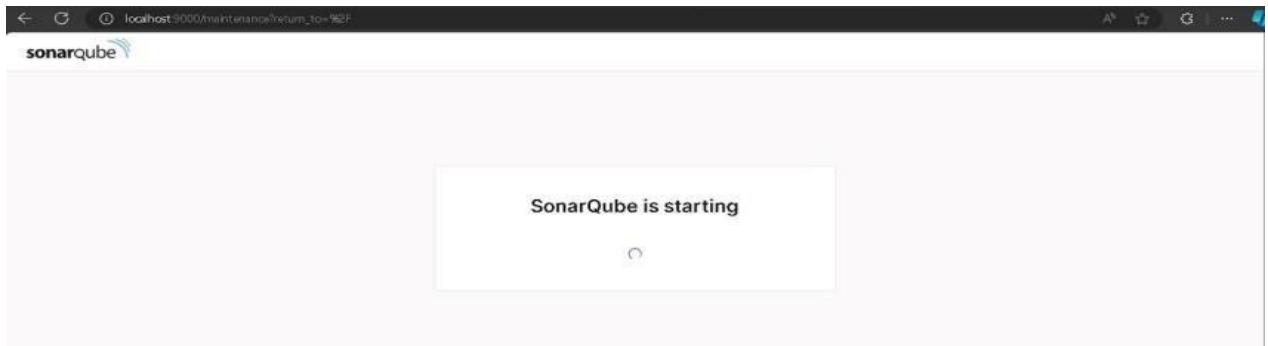
Steps to create a Jenkins CI/CD Pipeline and use SonarQube to perform SAST

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

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

```
PS C:\Users\91773\Desktop\College Resources\Advdevops Exp8> docker run -d --name sonarqube2 -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest  
71fc67f0b15baa5be5bdcd66966938e18682683d020beadcbc909dd027cfe7a  
PS C:\Users\91773\Desktop\College Resources\Advdevops Exp8>
```

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



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



Log in to SonarQube

Login *

admin

Password *

Go back Log in

5. 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](#)

Setup the project and come back to Jenkins Dashboard.

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

New Item

Enter an item name

KsSonarQube

Select an item type



Freestyle project

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



Maven project

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



Pipeline

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



Multi-configuration project

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

OK

7. Under Pipeline Script, enter the following - node {

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

Configure

General

Advanced Project Options

Pipeline

Pipeline

Definition Pipeline script

```

1 node {
2   stage('Cloning the GitHub Repo') {
3     git 'https://github.com/shazforiot/GOL.git'
4   }
5 
6 stage('SonarQube Analysis') {
7   withSonarQubeEnv('expB') {
8     bat """
9       <!--Program Files\Sonar Scanner\sonar-scanner-6.2.0.4584-windows-x64\bin\sonar-scanner.bat-->
10      -Dsonar.login=shazforiot
11      -Dsonar.password=kshitij24
12      -Dsonar.projectKey=sonarqube-test
13      -Dsonar.exclusions=vendor/**,resources/**,**/*.java
14      -Dsonar.host.url=http://127.0.0.1:9000/
15      """
16   }
17 }

```

Use Groovy Sandbox

Pipeline Syntax

Save **Apply**

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

8. Run The Build.

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

Status

KsSonarQube

</> Changes

▷ Build Now

Configure

Delete Pipeline

Full Stage View

SonarQube

Stages

Rename

Pipeline Syntax

Build History

trend

Filter...

#9 Sep 25 20:49 No Changes

Average stage times:
(Average full run time: ~8min 36s)

Cloning the GitHub Repo	SonarQube Analysis
2s	1min 44s
2s	8min 33s
4s	835ms failed
2s	3s failed
2s	3s failed

Stage View

 Status

 Changes

 Console Output

 View as plain text

 Edit Build Information

 Delete build '#9'

 Timings

 Git Build Data

 Pipeline Overview

 Pipeline Console

 Replay

 Pipeline Steps

 Workspaces

 Previous Build

Console Output

Skipping 4,247 KB. [Full Log](#)

```
20:56:15.267 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 798. Keep only the first 100 references.
20:56:15.267 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 810. Keep only the first 100 references.
20:56:15.267 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 823. Keep only the first 100 references.
20:56:15.267 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 844. Keep only the first 100 references.
20:56:15.267 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 509. Keep only the first 100 references.
20:56:15.267 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 1065. Keep only the first 100 references.
20:56:15.267 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 776. Keep only the first 100 references.
20:56:15.267 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 778. Keep only the first 100 references.
20:56:15.267 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 530. Keep only the first 100 references.
20:56:15.267 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 648. Keep only the first 100 references.
20:56:15.267 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 798. Keep only the first 100 references.
20:56:15.267 WARN Too many duplication references on file gameoflife-
```

for block at line 17. Keep only the first 100 references.

```
20:56:18.455 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html
for block at line 296. Keep only the first 100 references.
```

```
20:56:18.455 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html
for block at line 75. Keep only the first 100 references.
```

```
20:56:18.456 INFO CPD Executor CPD calculation finished (done) | time=107093ms
```

```
20:56:18.490 INFO SCM revision ID 'ba799ba7e1b576f04a4612322b0412c5e6e1e5e4'
```

```
20:57:50.106 INFO Analysis report generated in 3149ms, dir size=127.2 MB
```

```
20:57:56.943 INFO Analysis report compressed in 6828ms, zip size=29.6 MB
```

```
20:57:58.685 INFO Analysis report uploaded in 1732ms
```

```
20:57:58.688 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://127.0.0.1:9000/dashboard?id=sonarqube-test
```

```
20:57:58.688 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
```

```
20:57:58.688 INFO More about the report processing at http://127.0.0.1:9000/api/ce/task?id=18847db4-4f06-4766-9ad4-ee006448353c
```

```
20:58:06.225 INFO Analysis total time: 8:22.672 s
```

```
20:58:06.231 INFO SonarScanner Engine completed successfully
```

```
20:58:06.824 INFO EXECUTION SUCCESS
```

```
20:58:06.857 INFO Total time: 8:31.713s
```

```
[Pipeline] }
```

```
[Pipeline] // withSonarQubeEnv
```

```
[Pipeline] }
```

```
[Pipeline] // stage
```

```
[Pipeline] }
```

```
[Pipeline] // node
```

```
[Pipeline] End of Pipeline
```

```
Finished: SUCCESS
```

10. After that, check the project in SonarQube.

sonarqube-test / main ?

Overview Issues Security Hotspots Measures Code Activity Project Settings

main

Quality Gate: Passed

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

Last analysis 16 minutes ago

New Code Overall Code

Security: 0 Open issues (A) | Reliability: 68k Open issues (C) | Maintainability: 164k Open issues (A)

Accepted issues: 0 | Coverage: 50.6% (On 759k lines)

Security Hotspots: 3 (E)

Under different tabs, check all different issues with the code. 11.

Bugs

Responsibility

Add to selection Ctrl + click

Software Quality

- Security: 0
- Reliability: 33k
- Maintainability: 0

Severity

Type

- Bug: 33k
- Vulnerability: 0
- Code Smell: 164k

Scope

Status

Bulk Change

Select issues ▾ Navigate to issue ▾ 32,896 issues 1369d effort

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

- Insert a <!DOCTYPE> declaration to before this <html> tag. (Consistency, user-experience) L1 • 5min effort • 4 years ago • Bug • Major
- Insert a <!DOCTYPE> declaration to before this <html> tag. (Consistency, user-experience) L1 • 5min effort • 4 years ago • Bug • Major
- Insert a <!DOCTYPE> declaration to before this <html> tag. (Consistency, user-experience) L1 • 5min effort • 4 years ago • Bug • Major

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

gameoflife-core/build/reports/tests/alltests-errors.html

Code Smells

Add to selection Ctrl + click

> Software Quality

> Severity ?

> Type

- Bug 33k
- Vulnerability 0
- Code Smell 164k

Add to selection Ctrl + click

> Scope

> Status

> Security Category

Bulk Change Select issues Navigate to issue 163,766 issues 1705d effort gameoflife-core/build/reports/tests/all-tests.html

Remove this deprecated "width" attribute. Consistency html5 obsolete +

Maintainability Open Not assigned L9 ~ 5min effort ~ 4 years ago ⚡ Code Smell ⚡ Major

Remove this deprecated "align" attribute. Consistency html5 obsolete +

Maintainability Open Not assigned L11 ~ 5min effort ~ 4 years ago ⚡ Code Smell ⚡ Major

Remove this deprecated "align" attribute. Consistency html5 obsolete +

Maintainability Open Not assigned L12 ~ 5min effort ~ 4 years ago ⚡ Code Smell ⚡ Major

Remove this deprecated "size" attribute. Consistency html5 obsolete +

Maintainability Open Not assigned

Intentional issues

Issues in new code

> Clean Code Attribute

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

Add to selection Ctrl + click

> Software Quality

> Severity ?

> Type

- Bug 14k
- Vulnerability 0
- Code Smell 268

Bulk Change Select issues Navigate to issue 13,887 issues 59d effort gameoflife-acceptance-tests/Dockerfile

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

Maintainability Open Not assigned L1 ~ 5min effort ~ 4 years ago ⚡ Code Smell ⚡ Major

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

Maintainability Open Not assigned L12 ~ 5min effort ~ 4 years ago ⚡ Code Smell ⚡ Major

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

Maintainability Open Not assigned L12 ~ 5min effort ~ 4 years ago ⚡ Code Smell ⚡ Major

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

Maintainability Open Not assigned

Reliabilities issue

Issues in new code

> Clean Code Attribute

- Consistency 54k
- Intentionality 14k
- Adaptability 0
- Responsibility 0

Add to selection Ctrl + click

> Software Quality

- Security 0
- Reliability 54k
- Maintainability 164k

Add to selection Ctrl + click

> Severity ?

> Type

Bulk Change Select issues Navigate to issue 53,752 issues 1587d effort gameoflife-core/build/reports/tests/all-tests.html

Insert a <!DOCTYPE> declaration to before this <html> tag. Consistency user-experience +

Maintainability Reliability Open Not assigned L1 ~ 5min effort ~ 4 years ago ⚡ Bug ⚡ Major

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

Maintainability Reliability Open Not assigned L29 ~ 5min effort ~ 4 years ago ⚡ Code Smell ⚡ Minor

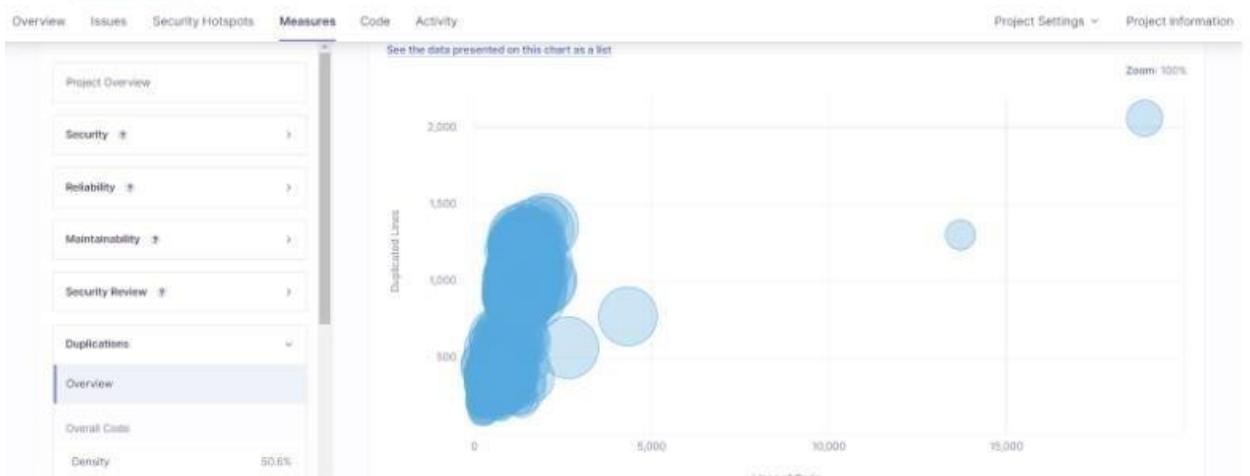
Anchors must have content and the content must be accessible by a screen reader. Consistency accessibility +

Maintainability Reliability Open Not assigned L38 ~ 5min effort ~ 4 years ago ⚡ Code Smell ⚡ Minor

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

Maintainability Reliability Open Not assigned

Duplicates



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

Conclusion:

In this experiment, we integrated Jenkins with SonarQube to enable automated code quality checks within our CI/CD pipeline. We started by deploying SonarQube using Docker, setting up a project, and configuring it to analyze code quality. Next, we configured Jenkins by installing the SonarQube Scanner plugin, adding SonarQube server details, and setting up the scanner tool. We then developed a Jenkins pipeline to automate the process of cloning a GitHub repository and running SonarQube analysis on the code. This integration helps ensure continuous monitoring of code quality, detecting issues such as bugs, code smells, and security vulnerabilities throughout the development process.

Adv DevOps Exp 09

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

Theory:

What is Nagios?

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

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

Why We Need Nagios tool?

Here are the important reasons to use Nagios monitoring tool:

- Detects all types of network or server issues
- Helps you to find the root cause of the problem which allows you to get the permanent solution to the problem
- Active monitoring of your entire infrastructure and business processes
- Allows you to monitor and troubleshoot server performance issues
- Helps you to plan for infrastructure upgrades before outdated systems create failures
- You can maintain the security and availability of the service
- Automatically fix problems in a panic situation

Features of Nagios

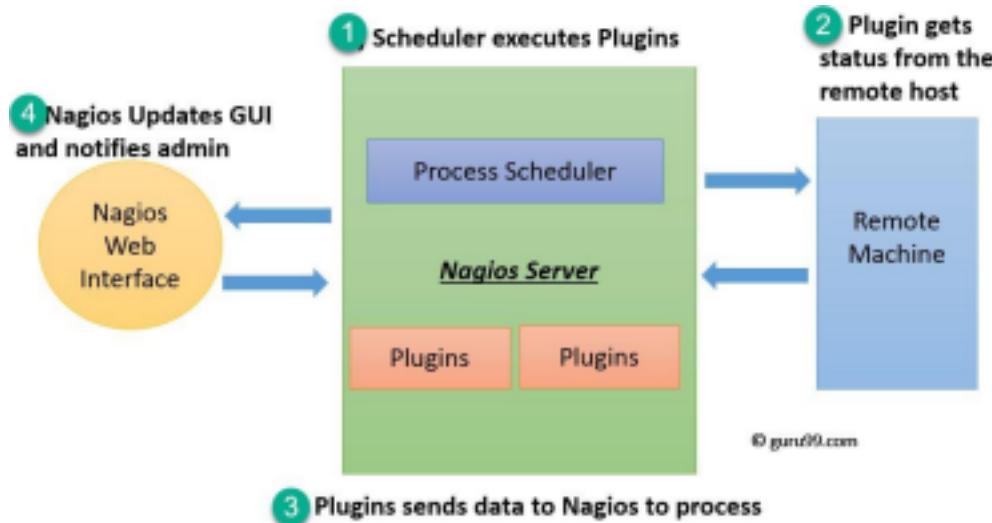
Following are the important features of Nagios monitoring tool:

- Relatively scalable, Manageable, and Secure
- Good log and database system
- Informative and attractive web interfaces
- Automatically send alerts if condition changes
- If the services are running fine, then there is no need to do check that host is alive
- Helps you to detect network errors or server crashes
- You can troubleshoot the performance issues of the server.
- The issues, if any, can be fixed automatically as they are identified during the monitoring process
- You can monitor the entire business process and IT infrastructure with a single pass
- The product's architecture is easy to write new plugins in the language of your choice
- Nagios allows you to read its configuration from an entire directory which helps you to decide how to define individual files
- Utilizes topology to determine dependencies
- Monitor network services like HTTP, SMTP, HTTP, SNMP, FTP, SSH, POP, etc.
- Helps you to define network host hierarchy using parent hosts
- Ability to define event handlers that runs during service or host events for proactive

- problem resolution
- Support for implementing redundant monitoring hosts

Nagios Architecture

Nagios is a client-server architecture. Usually, on a network, a Nagios server is running on a host, and plugins are running on all the remote hosts which should be monitored.



1. The scheduler is a component of the server part of Nagios. It sends a signal to execute the plugins at the remote host.
2. The plugin gets the status from the remote host
3. The plugin sends the data to the process scheduler
4. The process scheduler updates the GUI and notifications are sent to admins.

Step 1: Create a security group with the required configurations

I have created a new security group with a name 'newsecurity'

EC2 > Security Groups > Create security group

Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, you must provide a name and optional description. You can also specify the VPC that the security group belongs to.

Basic details

Security group name Info

Name cannot be edited after creation.

Description Info

VPC Info

I have modified the INBOUND RULES as follows

Inbound rules Info

Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>	Delete
HTTP	TCP	80	Anywhere	:/0 X	<input type="button" value="Delete"/>
HTTPS	TCP	443	Anywhere	0.0.0.0/0 X	<input type="button" value="Delete"/>
SSH	TCP	22	Anywhere	0.0.0.0/0 X	<input type="button" value="Delete"/>
All ICMP - IPv6	IPv6 ICMP	All	Anywhere	:/0 X	<input type="button" value="Delete"/>
All ICMP - IPv4	ICMP	All	Anywhere	0.0.0.0/0 X	<input type="button" value="Delete"/>
All traffic	All	All	Anywhere	0.0.0.0/0 X	<input type="button" value="Delete"/>
Custom TCP	TCP	5666	Anywhere	0.0.0.0/0 X	<input type="button" value="Delete"/>

Step 2: Create ec2 instance

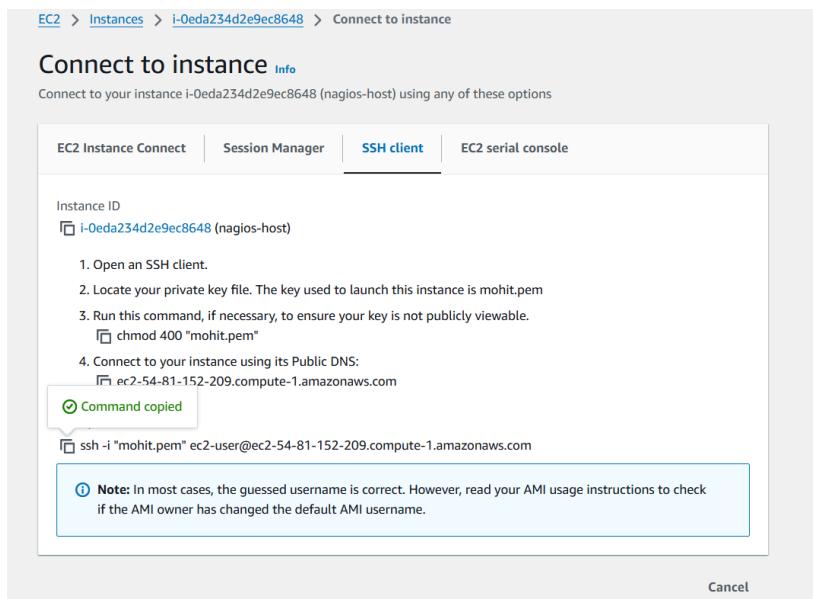
Name it as nagios-host. Select instance type as amazon-linux and choose the already created key pair and security group

The screenshot shows the AWS Lambda console interface. A modal window is open for creating a new function. The 'Function name' field contains 'HelloWorld'. Under 'Runtime', 'Node.js 12.x' is selected. The 'Handler' field shows 'index.handler'. In the 'Code' section, 'Upload a ZIP file' is selected, and a file named 'lambda_function.zip' is uploaded. The 'Configure' tab is selected. On the left, there's a sidebar with 'Lambda' and 'AWS Lambda'.

The screenshot shows the AWS Lambda console interface. A modal window is open for creating a new function. The 'Function name' field contains 'HelloWorld'. Under 'Runtime', 'Node.js 12.x' is selected. The 'Handler' field shows 'index.handler'. In the 'Code' section, 'Upload a ZIP file' is selected, and a file named 'lambda_function.zip' is uploaded. The 'Configure' tab is selected. On the left, there's a sidebar with 'Lambda' and 'AWS Lambda'.

The screenshot shows the AWS Lambda console interface. A modal window is open for creating a new function. The 'Function name' field contains 'HelloWorld'. Under 'Runtime', 'Node.js 12.x' is selected. The 'Handler' field shows 'index.handler'. In the 'Code' section, 'Upload a ZIP file' is selected, and a file named 'lambda_function.zip' is uploaded. The 'Configure' tab is selected. On the left, there's a sidebar with 'Lambda' and 'AWS Lambda'.

Copy the given ssh command, as we will require it for logging into our nagios-host instance from our windows powershell



Step 3: Open an administrative powershell and remotely login using the above mentioned ssh command

```

> ec2-user@ip-172-31-92-249:~ 
Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.

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

PS C:\WINDOWS\system32> cd C:\Users\Del1\Downloads
PS C:\Users\Del1\Downloads> ssh -i "mohit.pem" ec2-user@ec2-54-81-152-209.compute-1.amazonaws.com
, # ~\_ ##### Amazon Linux 2023
~~ \#####\ ~\###| ~~ \#/#/ https://aws.amazon.com/linux/amazon-linux-2023
~~ \#/ .-'-> ~~ V~' '-'>
~~ / ~~ / 
~~ / .-' / 
~~ / / 
Last login: Mon Sep 30 09:25:13 2024 from 125.99.93.18
, # ~\_ ##### Amazon Linux 2023
~~ \#####\ ~\###| ~~ \#/#/ https://aws.amazon.com/linux/amazon-linux-2023
~~ \#/ .-'-> ~~ V~' '-'>
~~ / ~~ / 
~~ / .-' / 
~~ / / 
Last login: Mon Sep 30 09:25:13 2024 from 125.99.93.18
[ec2-user@ip-172-31-92-249 ~]$ sudo yum update
Last metadata expiration check: 0:13:13 ago on Mon Sep 30 09:23:03 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-92-249 ~]$ sudo yum install httpd php
Last metadata expiration check: 0:13:23 ago on Mon Sep 30 09:23:03 2024.
Package httpd-2.4.62-1.amzn2023.x86_64 is already installed.
Package php8.3-8.3.10-1.amzn2023.0.1.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!

```

And then run these commands
sudo yum update
sudo yum install httpd php

```
[ec2-user@ip-172-31-41-160 ~]$ sudo yum update
Last metadata expiration check: 0:01:37 ago on Wed Oct 2 12:28:33 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-41-160 ~]$ sudo yum install httpd php
Last metadata expiration check: 0:01:45 ago on Wed Oct 2 12:28:33 2024.
Dependencies resolved.

=====
Package           Architecture      Version       Repository   Size
=====
Installing:
httpd            x86_64          2.4.62-1.amzn2023
php8_3           x86_64          8.3.10-1.amzn2023.0.1

Installing dependencies:
apr              x86_64          1.7.2-2.amzn2023.0.2
apr-util         x86_64          1.6.3-1.amzn2023.0.1
generic-logs-httd noarch        18.0.0-12.amzn2023.0.3
httpd-core       x86_64          2.4.62-1.amzn2023
httpd-filesystem noarch        2.4.62-1.amzn2023
httpd-tools      x86_64          2.4.62-1.amzn2023
libcurl          x86_64          1.0.19-4.amzn2023.0.2
libcurl-ssl     x86_64          1.0.19-5.amzn2023
libcurl-openssl x86_64          1.0.19-4.amzn2023
libcurl-zlib    x86_64          1.0.19-5.amzn2023
libxml2          x86_64          2.1.49-3.amzn2023.0.3
mailing          noarch        2.1.24-0.1.amzn2023.0.4
nginx-filesystem noarch        2.1.24-0.1.amzn2023.0.4
php8_3-3-libs   x86_64          8.3.10-1.amzn2023.0.1
php8_3-common   x86_64          8.3.10-1.amzn2023.0.1
php8_3-process  x86_64          8.3.10-1.amzn2023.0.1
php8_3-xml     x86_64          8.3.10-1.amzn2023.0.1

Installing weak dependencies:
apr-util-openssl x86_64          1.6.3-1.amzn2023.0.1
mod_ssl          x86_64          2.5.22-1.amzn2023.0.3
mod_wsgi         x86_64          2.4.62-1.amzn2023
php8_3-3-fpm    x86_64          8.3.10-1.amzn2023.0.1
php8_3-3-mbstring x86_64          8.3.10-1.amzn2023.0.1
php8_3-3-opcache x86_64          8.3.10-1.amzn2023.0.1
php8_3-3-pdo    x86_64          8.3.10-1.amzn2023.0.1
php8_3-3-sodium x86_64          8.3.10-1.amzn2023.0.1

Transaction Summary
Install 25 Packages
```

sudo yum install gcc glibc glibc-common

```
[ec2-user@ip-172-31-41-160 ~]$ sudo yum install gcc glibc glibc-common
Last metadata expiration check: 0:02:02 ago on Wed Oct 2 12:28:33 2024.
Package glibc-2.34-52.amzn2023.0.11.x86_64 is already installed.
Package glibc-common-2.34-52.amzn2023.0.11.x86_64 is already installed.
Dependencies resolved.

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

Transaction Summary
Install 13 Packages

Total download size: 52 M
Installed size: 168 M
Is this ok [y/N]: y
Downloading Packages:
(1/13): annobin-docs-10.93-1.amzn2023.0.1.noarch.rpm                                852 kB/s |  92 kB  00:00
(2/13): annobin-plugin-gcc-10.93-1.amzn2023.0.1.x86_64.rpm                          6.5 MB/s | 887 kB  00:00
(3/13): gc-8.0.4-5.amzn2023.0.2.x86_64.rpm                                         2.3 MB/s | 105 kB  00:00
(4/13): glibc-devel-2.34-52.amzn2023.0.11.x86_64.rpm                           1.1 MB/s | 27 kB  00:00
(5/13): cpp-11.4.1-2.amzn2023.0.2.x86_64.rpm                                     32 MB/s | 10 MB  00:00
(6/13): glibc-headers-x86-2.34-52.amzn2023.0.11.noarch.rpm                      2.9 MB/s | 427 kB  00:00
(7/13): kernel-headers-6.1.109-118.189.amzn2023.x86_64.rpm                         16 MB/s | 1.4 MB  00:00
(8/13): libmpc-1.2.1-2.amzn2023.0.2.x86_64.rpm                               2.1 MB/s | 62 kB  00:00
(9/13): guile22-2.2.7-2.amzn2023.0.3.x86_64.rpm                            27 MB/s | 6.4 MB  00:00
(10/13): libtool-ltdl-2.4.7-1.amzn2023.0.3.x86_64.rpm                         322 kB/s | 38 kB  00:00
(11/13): libcrypt-devel-4.4.33-7.amzn2023.x86_64.rpm                         1.4 MB/s | 32 kB  00:00
```

sudo yum install gd gd-devel

Package	Architecture	Version	Repository	Size
gd	x86_64	2.3.3-3.amzn2023.0.3	amazonlinux	139 k
gd-devel	x86_64	2.3.3-5.amzn2023.0.3	amazonlinux	38 k
Installing:				
gd	x86_64	1.0.9-4.amzn2023.0.2	amazonlinux	314 k
gd-devel	x86_64	1.0.9-4.amzn2023.0.2	amazonlinux	31 k
brotli	x86_64	1.0.8-6.amzn2023.0.2	amazonlinux	214 k
bzip2-devel	x86_64	1.17.6-2.amzn2023.0.1	amazonlinux	684 k
cairo	x86_64	3.22.2-1.amzn2023.0.4	amazonlinux	16 k
caja-filesystem	x86_64	2.13.94-2.amzn2023.0.2	amazonlinux	273 k
fontconfig	x86_64	2.13.94-2.amzn2023.0.2	amazonlinux	128 k
fontconfig-devel	x86_64	1.12.0.5-12.amzn2023.0.2	amazonlinux	9.5 k
fonts-filesystem	x86_64	2.13.2-5.amzn2023.0.1	amazonlinux	423 k
freetype	x86_64	2.13.2-5.amzn2023.0.1	amazonlinux	912 k
freetype-devel	x86_64	2.74.7-689.amzn2023.0.2	amazonlinux	486 k
glib2-devel	x86_64	2.99.01296.2.amzn2023.0.2	amazonlinux	15 k
google-noto-fonts-common	x86_64	2.02.01206-2.amzn2023.0.2	amazonlinux	492 k
google-noto-sans-vf-fonts	x86_64	1.3.14-7.amzn2023.0.2	amazonlinux	97 k
grayscale2	x86_64	1.3.14-7.amzn2023.0.2	amazonlinux	21 k
grayscale2-devel	x86_64	7.0.0-2.amzn2023.0.1	amazonlinux	868 k
harfbuzz	x86_64	7.0.0-2.amzn2023.0.1	amazonlinux	404 k
harfbuzz-devel	x86_64	7.0.0-2.amzn2023.0.1	amazonlinux	18 k
harfbuzz-icu	x86_64	2.1.21.amzn2023.0.2	amazonlinux	54 k
jbigkit-libs	x86_64	3.0-21.amzn2023.0.4	amazonlinux	10 k
langpacks-core-font-en	x86_64	1.0.19-6.amzn2023.0.2	amazonlinux	71 k
libICE	x86_64	1.2.3-8.amzn2023.0.2	amazonlinux	42 k
libSM	x86_64	1.1.7.2-3.amzn2023.0.4	amazonlinux	657 k
libX11	x86_64	1.1.7.2-3.amzn2023.0.4	amazonlinux	152 k
libX11-common	x86_64	1.1.7.2-3.amzn2023.0.4	amazonlinux	939 k
libX11-devel	x86_64	1.1.7.2-3.amzn2023.0.4	amazonlinux	12 k
libX11-xcb	x86_64	1.0.9-6.amzn2023.0.2	amazonlinux	31 k
libXau	x86_64	1.0.9-6.amzn2023.0.2	amazonlinux	14 k
libXau-devel	x86_64	1.0.9-6.amzn2023.0.2	amazonlinux	41 k
libXext	x86_64	3.5.15-2.amzn2023.0.3	amazonlinux	65 k
libXpm	x86_64	3.5.15-2.amzn2023.0.3	amazonlinux	59 k
libXpm-devel	x86_64	0.9.10-14.amzn2023.0.2	amazonlinux	28 k
libXrender	x86_64	1.2.0-4.amzn2023.0.2	amazonlinux	181 k
libXt	x86_64	2.37.4-1.amzn2023.0.4	amazonlinux	15 k

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-172-31-41-160 ~]$ sudo adduser -m nagios
[ec2-user@ip-172-31-41-160 ~]$ sudo passwd nagios
Changing password for user nagios.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-41-160 ~]$
```

Create a new user group & create a new directory for Nagios downloads using the following commands

sudo groupadd nagcmd
sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
mkdir ~/downloads
cd ~/downloads

Use **wget** to download the source zip files.

In this step, we are downloading, the latest version of nagios and the necessary plugins required to carry out the tasks of setting up a nagios server

wget <https://sourceforge.net/projects/nagios/files/latest/download>

```
[ec2-user@ip-172-31-41-160:~/downloads/nagios-4.5.5]
[ec2-user@ip-172-31-41-160 downloads]$ wget https://sourceforge.net/projects/nagios/files/latest/download
--2024-10-02 12:34:21-- https://sourceforge.net/projects/nagios/files/latest/download
Resolving sourceforge.net (sourceforge.net)... 172.64.150.145, 104.18.37.111, 2606:4700:4400::6812:256f, ...
Connecting to sourceforge.net (sourceforge.net)|172.64.150.145|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://downloads.sourceforge.net/project/nagios/nagios-4.x/nagios-4.5.5/nagios-4.5.5.tar.gz?ts=gAAAAABm_T3NmNSZPzP-6la2Tltvo0GCG7VVV7QGVH08n3tC240ehfMw7vhCoKbGHg2iIRxbmfugI10LccNfxtaoixg3jzKg3w3%3Duse_mirror=phoenixnapkr=[following]
--2024-10-02 12:34:21-- https://downloads.sourceforge.net/project/nagios/nagios-4.x/nagios-4.5.5/nagios-4.5.5.tar.gz?ts=gAAAAABm_T3NmNSZPzP-6la2Tltvo0GCG7VVV7QGVH08n3tC240ehfMw7vhCoKbGHg2iIRxbmfugI10LccNfxtaoixg3jzKg3w3%3Duse_mirror=phoenixnapkr=
Resolving downloads.sourceforge.net (downloads.sourceforge.net)|204.68.111.105|:443... connected.
Connecting to downloads.sourceforge.net (downloads.sourceforge.net)|204.68.111.105|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2065473 (2.0M) [application/x-gzip]
Saving to: 'download'

download                                         100%[=====] 1.97M 4.23MB/s in 0.5s

2024-10-02 12:34:22 (4.23 MB/s) - 'download' saved [2065473/2065473]
```

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

```
[ec2-user@ip-172-31-41-160:~/downloads/nagios-4.5.5]
[ec2-user@ip-172-31-41-160 downloads]$ wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
--2024-10-02 12:34:46-- https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251
Connecting to nagios-plugins.org (nagios-plugins.org)|45.56.123.251|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2753049 (2.0M) [application/x-gzip]
Saving to: 'nagios-plugins-2.4.11.tar.gz'

nagios-plugins-2.4.11.tar.gz 100%[=====] 2.62M 7.48MB/s in 0.4s

2024-10-02 12:34:46 (7.48 MB/s) - 'nagios-plugins-2.4.11.tar.gz' saved [2753049/2753049]
```

```
[ec2-user@ip-172-31-92-249:~/downloads]
[ec2-user@ip-172-31-92-249 ~]$ cd ~/downloads
[ec2-user@ip-172-31-92-249 downloads]$ wget https://sourceforge.net/projects/nagios/files/latest/download
--2024-09-30 09:54:56-- https://sourceforge.net/projects/nagios/files/latest/download
Resolving sourceforge.net (sourceforge.net)... 172.64.150.145, 104.18.37.111, 2606:4700:4400::6812:256f, ...
Connecting to sourceforge.net (sourceforge.net)|172.64.150.145|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://downloads.sourceforge.net/project/nagios/nagios-4.x/nagios-4.5.5/nagios-4.5.5.tar.gz?ts=gAAAAABm-nVw9RdAvnMaShLf3gu4RXTSVxrtZ6fGxJvhVAOzpB1bPgbyzLMcDDAALgtEC1p0Kr0cgJNJ23bKktan1cJ0Vfkpg3%3Duse_mirror=netaactuate&r=[following]
--2024-09-30 09:54:56-- https://downloads.sourceforge.net/project/nagios/nagios-4.x/nagios-4.5.5/nagios-4.5.5.tar.gz?ts=gAAAAABm-nVw9RdAvnMaShLf3gu4RXTSVxrtZ6fGxJvhVAOzpB1bPgbyzLMcDDAALgtECl0OKrcpgJNj23bKktan1cJ0Vfkpg3%3Duse_mirror=netaactuate&r=
Resolving downloads.sourceforge.net (downloads.sourceforge.net)|204.68.111.105|:443... connected.
Connecting to downloads.sourceforge.net (downloads.sourceforge.net)|204.68.111.105|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2065473 (2.0M) [application/x-gzip]
Saving to: 'download'

download                                         100%[=====] 1.97M ---KB/s in 0.07s

2024-09-30 09:54:57 (29.8 MB/s) - 'download' saved [2065473/2065473]

[ec2-user@ip-172-31-92-249 downloads]$ wget https://nagios-plugins.org/download/nagios-plugins-2.4.9.tar.gz
--2024-09-30 09:56:53-- https://nagios-plugins.org/download/nagios-plugins-2.4.9.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251
Connecting to nagios-plugins.org (nagios-plugins.org)|45.56.123.251|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2754403 (2.0M) [application/x-gzip]
Saving to: 'nagios-plugins-2.4.9.tar.gz'

nagios-plugins-2.4.9.tar.gz 100%[=====] 2.63M 7.54MB/s in 0.3s

2024-09-30 09:56:54 (7.54 MB/s) - 'nagios-plugins-2.4.9.tar.gz' saved [2754403/2754403]
```

Now, we run the next command in the following manner

tar zxvf <nagios-4.5.5 version> (for me it has gotten saved as 'download')

So i wrote tar zxvf download

```
[ec2-user@ip-172-31-41-160:~/downloads]$ tar zxvf download
nagios-4.5.5/
nagios-4.5.5/.github/
nagios-4.5.5/.github/workflows/
nagios-4.5.5/.github/workflows/test.yml
nagios-4.5.5/.gitignore
nagios-4.5.5/CONTRIBUTING.md
nagios-4.5.5/Changelog
nagios-4.5.5/INSTALLING
nagios-4.5.5/LEGAL
nagios-4.5.5/LICENSE
nagios-4.5.5/Makefile.in
nagios-4.5.5/README.md
nagios-4.5.5/THANKS
nagios-4.5.5/UPGRADING
nagios-4.5.5/autocal.m4
nagios-4.5.5/autocom-macros/
nagios-4.5.5/autocom-macros/.gitignore
nagios-4.5.5/autocom-macros/CHANGELOG.md
nagios-4.5.5/autocom-macros/LICENSE
nagios-4.5.5/autocom-macros/LICENSE.md
nagios-4.5.5/autocom-macros/README.md
nagios-4.5.5/autocom-macros/add_group_user
nagios-4.5.5/autocom-macros/ax_nagios_get_distrib
nagios-4.5.5/autocom-macros/ax_nagios_get_files
nagios-4.5.5/autocom-macros/ax_nagios_get_inetd
nagios-4.5.5/autocom-macros/ax_nagios_get_init
nagios-4.5.5/autocom-macros/ax_nagios_get_os
nagios-4.5.5/autocom-macros/ax_nagios_get_paths
nagios-4.5.5/autocom-macros/ax_nagios_get_ssl
nagios-4.5.5/base/
nagios-4.5.5/base/.gitignore
nagios-4.5.5/base/Makefile.in
nagios-4.5.5/base/broker.c
nagios-4.5.5/base/checks.c
nagios-4.5.5/base/commands.c
nagios-4.5.5/base/config.c
nagios-4.5.5/base/events.c
nagios-4.5.5/base/flapping.c
nagios-4.5.5/base/logging.c
nagios-4.5.5/base/nagios.c
nagios-4.5.5/base/nagiosstats.c
nagios-4.5.5/base/nemodus.c
nagios-4.5.5/base/nerc.c
```

After which we are supposed to **change our directory** over there

For eg. **cd nagios-4.5.5...** depending on the version that we have downloaded

Next, Run this command (make sure that you are working inside nagios-4.x.x directory)

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

```
[ec2-user@ip-172-31-41-160:~/downloads/nagios-4.5.5]$ cd nagios-4.5.5
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$ ./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking for suffix of object files... o
checking whether the compiler supports GNU C... yes
checking whether gcc accepts -g... yes
checking for gcc option to enable C11 features... none needed
checking whether make sets ${MAKE}... yes
checking whether ln -s works... yes
checking for strip... /usr/bin/strip
checking for sys/wait.h that is POSIX.1 compatible... yes
checking for stdio.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for strings.h... yes
checking for sys/stat.h... yes
checking for sys/types.h... yes
checking for unistd.h... yes
checking for arpa/inet.h... yes
checking for ctype.h... yes
checking for dirent.h... yes
checking for errno.h... yes
checking for fcntl.h... yes
checking for getopt.h... yes
checking for grp.h... yes
checking for libgen.h... yes
checking for limits.h... yes
checking for math.h... yes
checking for netdb.h... yes
checking for netinet/in.h... yes
checking for pwd.h... yes
checking for regex.h... yes
checking for signal.h... yes
checking for socket.h... no
checking for stdarg.h... yes
```

```
checking for strerror... yes
checking for strtoul... yes
checking for unsetenv... yes
checking for type of socket size... size_t
checking for Kerberos include files... configure: WARNING: could not find include files
checking for pkg-config... pkg-config
checking for SSL headers... configure: error: Cannot find ssl headers
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$
```

After running this command, we get an **error related to ssl header being absent**

For that purpose, we are to run the following command.

sudo yum install openssl-devel (for ssl header)

```
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$ sudo yum install openssl-devel
Last metadata expiration check: 0:12:11 ago on Wed Oct 2 12:28:33 2024.
Dependencies resolved.

-----  
 Package           Architecture      Version       Repository  Size  
-----  
Installing:  
openssl-devel.x86_64          1:3.0.8-1.amzn2023.0.14        amazonlinux   3.0 M  
  
Transaction Summary  
-----  
Install 1 Package  
  
Total download size: 3.0 M  
Installed size: 4.7 M  
Is this ok [y/N]: y  
Downloading Packages:  
openssl-devel-3.0.8-1.amzn2023.0.14.x86_64.rpm          26 MB/s | 3.0 MB  00:00  
  
total  
Running transaction check  
Transaction check succeeded.  
Running transaction test  
Transaction test succeeded.  
Running transaction  
  Preparing : 1/1  
  Installing : openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64 1/1  
  Running scriptlet: openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64 1/1  
  Verifying  : openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64 1/1  
  
Installed:  
openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64  
  
Complete!
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$
```

Now, Re-run `./configure --with-command-group=nagcmd`

After this, run **make all** command

```

ec2-user@ip-172-31-41-160:~/downloads/nagios-4.5.5
on doing this. Pay particular attention to the docs on
object configuration files, as they determine what/how
things get monitored!
make install-webconf
  This installs the Apache config file for the Nagios
  web interface
make install-exfoliation
  This installs the Exfoliation theme for the Nagios
  web interface
make install-classicui
  This installs the classic theme for the Nagios
  web interface

*** Support Notes *****
If you have questions about configuring or running Nagios,
please make sure that you:
  - Look at the sample config files
  - Read the documentation on the Nagios Library at:
    https://library.nagios.com

before you post a question to one of the mailing lists.
Also make sure to include pertinent information that could
help others help you. This might include:
  - What version of Nagios you are using
  - What version of the plugins you are using
  - Relevant snippets from your config files
  - Relevant error messages from the Nagios log file

For more information on obtaining support for Nagios, visit:
  https://support.nagios.com

*****
Enjoy.

```

Run the following set of commands to ensure that
sudo make install

```

ec2-user@ip-172-31-41-160:~/downloads/nagios-4.5.5
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$ sudo make install
cd ./base && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagios /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagiostats /usr/local/nagios/bin
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/base'
cd ./cgi && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
make install-basic
make[2]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/sbin
for file in *.cgi; do \
    /usr/bin/install -c -s -m 775 -o nagios -g nagios $file /usr/local/nagios/sbin; \
done
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
cd ./html && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/html'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/media
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/stylesheets
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/contexthelp
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/docs
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/docs/images
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/js
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/images
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/images/logos
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/includes
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/ssi
/usr/bin/install -c -m 664 -o nagios -g nagios ./robots.txt /usr/local/nagios/share
/usr/bin/install -c -m 664 -o nagios -g nagios ./jsonquery.html /usr/local/nagios/share
rm -f /usr/local/nagios/share/index.html
rm -f /usr/local/nagios/share/main.html
rm -f /usr/local/nagios/share/side.html
rm -f /usr/local/nagios/share/map.html
rm -f /usr/local/nagios/share/rss/*
rm -f /usr/local/nagios/share/graph-header.html
rm -f /usr/local/nagios/share/histogram.html
rm -f /usr/local/nagios/share/histogram-form.html
rm -f /usr/local/nagios/share/histogram-graph.html
rm -f /usr/local/nagios/share/histogram-links.html
rm -f /usr/local/nagios/share/infobox.html
rm -f /usr/local/nagios/share/map.php

```

sudo make install-init

```
[ec2-user@ip-172-31-41-160:~/downloads/nagios-4.5.5]
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$ sudo make install-init
/usr/bin/install -c -m 755 -d -o root -g root /lib/systemd/system
/usr/bin/install -c -m 755 -o root -g root startup/default-service /lib/systemd/system/nagios.service
```

sudo make install-config

```
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$ sudo make install-config
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc/objects
/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.cgi /usr/local/nagios/etc/cgi.cgi
/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/templates.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/commands.cfg /usr/local/nagios/etc/objects/commands.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/contacts.cfg /usr/local/nagios/etc/objects/contacts.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/timeperiods.cfg /usr/local/nagios/etc/objects/timeperiods.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/localhost.cfg /usr/local/nagios/etc/objects/localhost.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/windows.cfg /usr/local/nagios/etc/objects/windows.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/printer.cfg /usr/local/nagios/etc/objects/printer.cfg
/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.
```

sudo make install-webconf

```
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf
if [ 0 -eq 1 ]; then \
    ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi

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

[ec2-user@ip-172-31-41-160 nagios-4.5.5]$
```

Next, we are supposed to create a nagiosadmin account for nagios login along with password. Specify the password twice.

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

```
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$
```

Restart Apache

sudo service httpd restart

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

cd ~/downloads

tar zxvf nagios-plugins-2.4.11.tar.gz

```
[ec2-user@ip-172-31-41-160:~/downloads]
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$ cd ~/downloads
[ec2-user@ip-172-31-41-160 downloads]$ tar zxvf nagios-plugins-2.4.11.tar.gz
nagios-plugins-2.4.11/
nagios-plugins-2.4.11/build-aux/
nagios-plugins-2.4.11/build-aux/compile
nagios-plugins-2.4.11/build-aux/config.guess
nagios-plugins-2.4.11/build-aux/config.rpath
nagios-plugins-2.4.11/build-aux/config.sub
nagios-plugins-2.4.11/build-aux/install-sh
nagios-plugins-2.4.11/build-aux/ltdmain.sh
nagios-plugins-2.4.11/build-aux/missing
nagios-plugins-2.4.11/build-aux/mkinstalldirs
nagios-plugins-2.4.11/build-aux/depcomp
nagios-plugins-2.4.11/build-aux/snippet/
nagios-plugins-2.4.11/build-aux/snippet/_Noreturn.h
nagios-plugins-2.4.11/build-aux/snippet/arg-nonnull.h
nagios-plugins-2.4.11/build-aux/snippet/c++defs.h
nagios-plugins-2.4.11/build-aux/snippet/warn-on-use.h
nagios-plugins-2.4.11/build-aux/test-driver
```

Compile and install plugins

cd nagios-plugins-2.4.11

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

Run the following command:

sudo chkconfig --add nagios

On running the above command

```
[ec2-user@ip-172-31-41-160 nagios-plugins-2.4.11]$ sudo chkconfig --add nagios
error reading information on service nagios: No such file or directory
```

If this is the output that one is getting, then it means that the init script is missing...

We can check this by running ls /etc/init.d/

```
[ec2-user@ip-172-31-92-249 nagios-plugins-2.4.9]$ ls /etc/init.d/
README  functions
[ec2-user@ip-172-31-92-249 nagios-plugins-2.4.9]$
```

With ls command, we must see a file named nagios, which i was not able to see

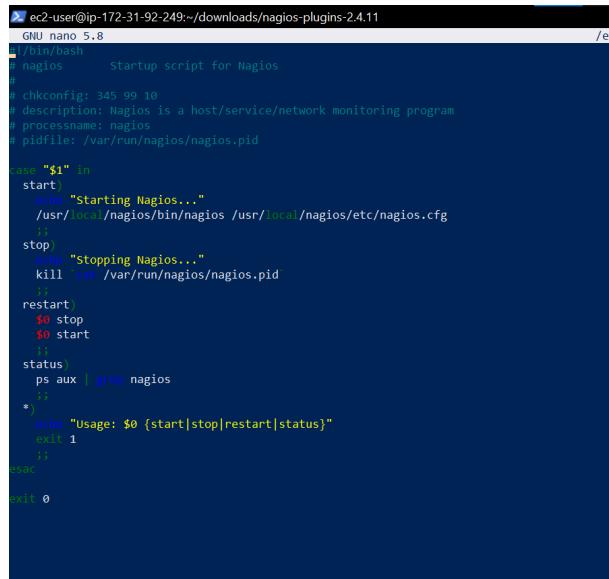
If the Init Script is Missing i.e If you don't see the nagios script in /etc/init.d/, you can create it manually. Here's how:

Run the following command:

sudo nano /etc/init.d/nagios

Within this file, paste the following script

```
#!/bin/bash
# nagios      Startup script for Nagios
#
# chkconfig: 345 99 10
# description: Nagios is a host/service/network monitoring program
# processname: nagios
# pidfile: /var/run/nagios/nagios.pid
case "$1" in
    start)
        echo "Starting Nagios..."
        /usr/local/nagios/bin/nagios /usr/local/nagios/etc/nagios.cfg
        ;;
    stop)
        echo "Stopping Nagios..."
        kill `cat /var/run/nagios/nagios.pid`
        ;;
    restart)
        $0 stop
        $0 start
        ;;
    status)
        ps aux | grep nagios
        ;;
    *)
        echo "Usage: $0 {start|stop|restart|status}"
        exit 1
        ;;
esac
exit 0
```



The screenshot shows a terminal window titled 'ec2-user@ip-172-31-92-249:~/downloads/nagios-plugins-2.4.11'. The window displays the script content from the previous code block. The script is a startup script for Nagios, using the nano 5.8 editor. The code includes comments explaining the purpose of each section: starting, stopping, restarting, and checking the status of the Nagios service. It also handles usage errors and exits with a status of 0 at the end.

Make the Script Executable: After saving the file, run the following command to make it executable:

sudo chmod +x /etc/init.d/nagios

Run sudo chkconfig --add nagios again

And then run sudo chkconfig nagios on

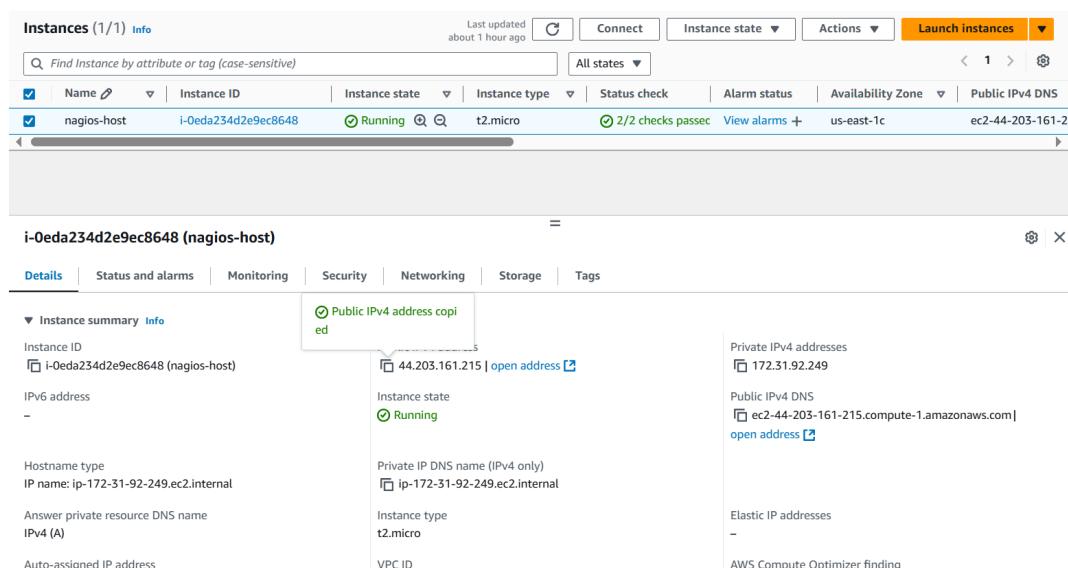
```
[ec2-user@ip-172-31-41-160 nagios-plugins-2.4.11]$ sudo nano /etc/init.d/nagios
[ec2-user@ip-172-31-41-160 nagios-plugins-2.4.11]$ sudo chmod +x /etc/init.d/nagios
[ec2-user@ip-172-31-41-160 nagios-plugins-2.4.11]$ sudo chkconfig --add nagios
[ec2-user@ip-172-31-41-160 nagios-plugins-2.4.11]$ sudo chkconfig nagios on
Note: Forwarding request to 'systemctl enable nagios.service'.
Synchronizing state of nagios.service with sysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable nagios
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service → /usr/lib/systemd/system/nagios.service.
[ec2-user@ip-172-31-41-160 nagios-plugins-2.4.11]$
```

sudo service nagios start

```
[ec2-user@ip-172-31-92-249 nagios-plugins-2.4.11]$ sudo service nagios start
Starting Nagios...
Nagios Core 4.5.5
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2024-09-17
License: GPL

Website: https://www.nagios.org
Nagios 4.5.5 starting... (PID=72261)
Local time is Tue Oct 01 20:59:58 UTC 2024.
wproc: Successfully registered manager as @wproc with query handler
wproc: Registry request: name=Core Worker 72265;pid=72265
wproc: Registry request: name=Core Worker 72264;pid=72264
wproc: Registry request: name=Core Worker 72263;pid=72263
wproc: Registry request: name=Core Worker 72262;pid=72262
Successfully launched command file worker with pid 72266
wproc: NOTIFY job 4 from worker Core Worker 72262 is a non-check helper but exited with return code 127
wproc: host=localhost; service=Swap Usage; contact=nagiosadmin
wproc: early_timeout=0; exited_ok=1; wait_status=32512; error_code=0;
wproc: stderr line 01: /bin/sh: line 1: /bin/mail: No such file or directory
wproc: stderr line 02: /usr/bin/printf: write error: Broken pipe
```

Get your public IPv4 address from your instance. We will require it for connecting to our nginx server



Browse for this url: http://<your_public_ip_address>/nagios

The browser may ask you for your nagios credentials which set in the earlier steps

The username is nagiosadmin and enter the password that you set earlier

The screenshot shows the Nagios Core web interface. The top header displays the URL as 'Not secure | 34.229.45.75/nagios/' and the title 'Nagios® Core™'. A green checkmark icon indicates 'Process running with PID 62668'. The left sidebar contains a navigation menu with sections like 'General' (Home, Documentation), 'Current Status' (Tactical Overview, Map, Hosts, Services, Host Groups, Summary, Grid), 'Service Groups' (Summary, Grid), 'Problems' (Services, Hosts, Network Outages), 'Reports' (Availability, Trends, Alerts, History, Summary, Histogram, Notifications, Event Log), and a 'Quick Search' bar. The main content area features a 'Get Started' section with a bulleted list: Start monitoring your infrastructure, Change the look and feel of Nagios, Extend Nagios with hundreds of addons, Get support, Get training, and Get certified. It also includes 'Latest News' and 'Don't Miss...' sections. On the right, there's a 'Quick Links' sidebar with links to Nagios Library, Nagios Labs, Nagios Exchange, Nagios Support, Nagios.com, and Nagios.org. A vertical 'Page Tour' button is located on the far right.

Conclusion:

In this experiment, we successfully installed and configured Nagios Core on an Amazon Linux EC2 instance, showcasing its role in continuous monitoring within a DevOps environment. We learned about user management and service configuration, emphasizing Nagios's ability to monitor systems and networks effectively. This experience laid the groundwork for enhancing infrastructure reliability and integrating advanced monitoring strategies in future projects.

Adv DevOps Exp 10

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

Monitoring Using Nagios:

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

Run this command **sudo systemctl status**

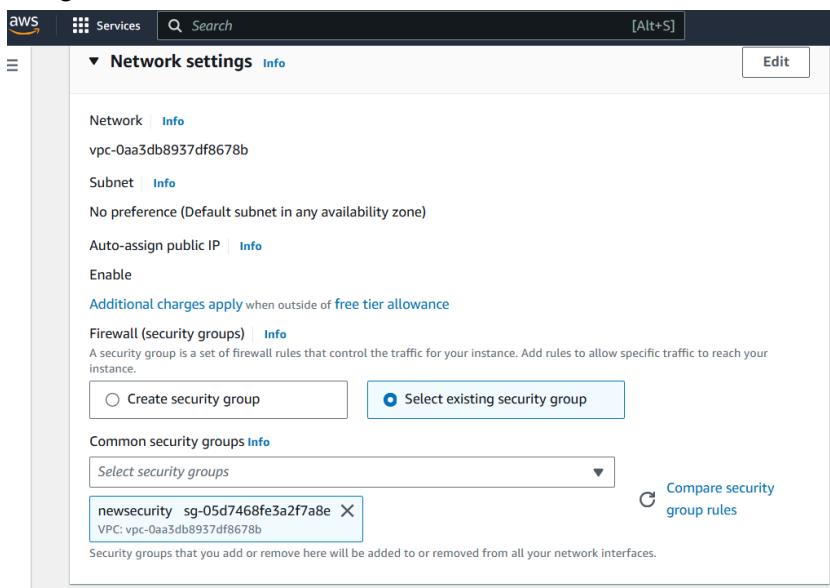
```
ec2-user@ip-172-31-41-160:~$ /downloads/nagios-plugins-2.4.11
[ec2-user@ip-172-31-41-160 nagios-plugins-2.4.11]$ sudo systemctl status
● ip-172-31-41-160.ec2.internal
    State: running
      Units: 296 loaded (incl. loaded aliases)
        Jobs: 0 queued
       Failed: 0 units
     Since: Wed 2024-10-02 12:28:05 UTC; 33min ago
    Systemd: 252.23-2.amzn2023
   CGroup: /
           └─init.scope
             ├─1 /usr/lib/systemd/systemd --switched-root --system --deserialize=32
             ├─system.slice
             ├─acpid.service
             ├─amazon-ssm-agent.service
             ├─atd.service
             ├─auditd.service
             ├─chronynd.service
             ├─dbus-broker.service
             ├─gssproxy.service
             ├─httpd.service
             ├─libstoragemgmt.service
             └─1940 /usr/bin/lsm -d

1938 /usr/bin/systemd-inhibit --what=handle-suspend-key:handle-hibernate-key --who=noah "--why=acpid instead" --mode=block /usr/sbin/acpid -f
2059 /usr/sbin/acpid -f
2141 /usr/bin/amazon-ssm-agent
2152 /usr/sbin/atd -f
1768 /sbin/auditd
2175 /usr/sbin/chronynd -F 2
1946 /usr/bin/dbus-broker-launch --scope system --audit
1954 dbus-broker --log 4 --controller 9 --machine-id ec2e4d759a3e2f6fe850b14e4cdacabe --max-bytes 536870912 --max-fds 4096 --max-matches 16384 --audit
1959 /usr/sbin/gssproxy -D
49553 /usr/sbin/httpd -DFOREGROUND
49555 /usr/sbin/httpd -DFOREGROUND
49556 /usr/sbin/httpd -DFOREGROUND
49557 /usr/sbin/httpd -DFOREGROUND
49558 /usr/sbin/httpd -DFOREGROUND
62800 /usr/sbin/httpd -DFOREGROUND
```

Step 2: Before we begin,

To monitor a Linux machine, create an **Ubuntu 20.04 server** EC2 Instance in AWS.

Provide it with the **same security group** as the Nagios Host and name it 'nagios-client' alongside the host.



Key pair (login) Info

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

Key pair name - *required*

mohit ▼ ⟳ Create new key pair

The screenshot shows the AWS EC2 Instances page. It displays two instances: 'nagios-host' and 'nagios-client', both of which are running. The 'Instances' section has a search bar and filters for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 DNS. The 'Actions' dropdown menu includes options like 'Launch instances'. The left sidebar shows navigation links for EC2 Dashboard, EC2 Global View, Events, Console-to-Code, Instances (selected), Instance Types, Launch Templates, and Spot Requests.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
nagios-host	i-03facef442a77494d	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-34-229-45-75
nagios-client	i-0b934b61f21351c1b	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-54-172-92-221

Step 3: TO BE DONE IN THE Nagios-host TERMINAL

In the nagios-host terminal, run this command

ps -ef | grep nagios

```
[ec2-user@ip-172-31-41-160 nagios-plugins-2.4.11]$ ps -ef | grep nagios
ec2-user 63115 2315 0 13:03 pts/0 00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-41-160 nagios-plugins-2.4.11]$ ■
```

To become a root user, run '**sudo su**' and make two directories using the following commands. If one is running these commands in windows powershell, make sure that he/she copies it line by line as powershell might make an error while interpreting multiple lines

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

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

```
[ec2-user@ip-172-31-92-249 ~]$ sudo su
[root@ip-172-31-92-249 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts
[root@ip-172-31-92-249 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-92-249 ec2-user]#
```

Copy the sample localhost.cfg file to linuxhost folder. Use the following mentioned command to achieve it

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

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

Open linuxserver.cfg using nano and make the following changes. This is a conf type file in which we will have to modify the configurations in way which will help us specify the hosts and clients to be monitored

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

Changes to be made:

1. Change the hostname to linux-server (EVERYWHERE ON THE FILE)
2. Change address to the public IP address of your LINUX CLIENT.
3. Change hostgroup_name under hostgroup to linux-servers1

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

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

    host_name     linux-server
    alias         localhost
    address       54.172.92.226
}

#####
# Define an optional hostgroup for Linux machines

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

IMP: Everywhere else on the file, change the hostname to linux-server instead of localhost.

Open the Nagios Config file and add the following line

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

Add the following line in the file and save

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

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

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

# Definitions for monitoring the local (Linux) host
cfg_file=/usr/local/nagios/etc/objects/localhost.cfg
cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/
# Definitions for monitoring a Windows machine
#cfg_file=/usr/local/nagios/etc/objects/windows.cfg
```

Verify the configuration files by running the following command

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

```
[root@ip-172-31-41-160 nagios-plugins-2.4.11]# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

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

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

Running pre-flight check on configuration data...

Checking objects...
  Checked 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-172-31-41-160 nagios-plugins-2.4.11]#
```

You are good to go if there are no errors.

Restart the nagios service

service nagios restart

And by running sudo systemctl status nagios, we can again check whether our server is running or not

```

root@ip-172-31-41-160:/tmp/nagios-plugins-2.4.11]
[root@ip-172-31-41-160 nagios-plugins-2.4.11]# sudo systemctl restart nagios
[root@ip-172-31-41-160 nagios-plugins-2.4.11]# sudo systemctl status nagios
● nagios.service - Nagios Core 5
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
     Active: active (running) since Wed 2024-10-02 13:20:17 UTC; 7s ago
       Docs: https://www.nagios.org/documentation
   Process: 78776 ExecStart=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
  Process: 78777 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 78778 (nagios)
   Tasks: 6 (limit: 1112)
      Memory: 4.0M
        CPU: 24ms
      CGroup: /system.slice/nagios.service
          └─78778 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 02 13:20:17 ip-172-31-41-160.ec2.internal nagios[78778]: qh: echo service query handler registered
Oct 02 13:20:17 ip-172-31-41-160.ec2.internal nagios[78778]: qh: help for the query handler registered
Oct 02 13:20:17 ip-172-31-41-160.ec2.internal nagios[78778]: wpoc: Successfully registered Nagios @proc with query handler
Oct 02 13:20:17 ip-172-31-41-160.ec2.internal nagios[78778]: wpoc: Registry request: name=Core Worker 78782;pid=78782
Oct 02 13:20:17 ip-172-31-41-160.ec2.internal nagios[78778]: wpoc: Registry request: name=Core Worker 78781;pid=78781
Oct 02 13:20:17 ip-172-31-41-160.ec2.internal nagios[78778]: wpoc: Registry request: name=Core Worker 78780;pid=78780
Oct 02 13:20:17 ip-172-31-41-160.ec2.internal nagios[78778]: wpoc: Registry request: name=Core Worker 78779;pid=78779
Oct 02 13:20:17 ip-172-31-41-160.ec2.internal nagios[78778]: Successfully launched command file worker with pid 78783
Oct 02 13:20:21 ip-172-31-41-160.ec2.internal nagios[78778]: HOST ALERT: linux-server;UP;SOFT;1;PING OK - Packet loss = 0%, RTA = 0.93 ms
Oct 02 13:20:24 ip-172-31-41-160.ec2.internal nagios[78778]: SERVICE ALERT: localhost;HTTP;WARNING;HARD;4;HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.0
[root@ip-172-31-41-160 nagios-plugins-2.4.11]# sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
     Drop-In: /usr/lib/systemd/system/httpd.service.d
       └─php-fpm.conf
     Active: active (running) since Wed 2024-10-02 12:47:56 UTC; 33min ago
       Docs: man:httpd.service(8)
   Main PID: 49553 (httpd)
     Status: "Total requests: 26; Idle/Busy workers 100/0;Requests/sec: 0.0129; Bytes served/sec: 94 B/sec"
     Tasks: 230 (limit: 1112)
        Memory: 1.7M
          CPU: 1.416s
        CGroup: /system.slice/httpd.service
            ├─49553 /usr/sbin/httpd -DFOREGROUND

```

Step 4: TO BE DONE IN THE Nagios-client TERMINAL

Now it is time to switch to the client machine.

SSH into the machine or simply use the EC2 Instance Connect feature.

```

PS C:\WINDOWS\system32> cd C:\Users\DEll\Downloads
PS C:\Users\DEll\Downloads> ssh -i "mohit.pem" ubuntu@ec2-54-172-92-226.compute-1.amazonaws.com
The authenticity of host 'ec2-54-172-92-226.compute-1.amazonaws.com (54.172.92.226)' can't be established.
ECDSA key fingerprint is SHA256:e/WkFQRuHSpjQ5hDMA0dku8msNHETN9SAgzEy53E.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-172-92-226.compute-1.amazonaws.com,54.172.92.226' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1016-aws x86_64)

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

System information as of Wed Oct  2 13:26:11 UTC 2024

System load:  0.0           Processes:      104
Usage of /:   22.8% of  6.71GB  Users logged in:    0
Memory usage: 20%           IPv4 address for enx0: 172.31.36.100
Swap usage:   0%

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

  https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

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

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

```

Make a package index update and install gcc, nagios-nrpe-server and the plugins. Run the following commands to achieve the same.

sudo apt update -y

sudo apt install gcc -y

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

Open nrpe.cfg file to make changes.

sudo nano /etc/nagios/nrpe.cfg

Under allowed_hosts, add your nagios host IP address like so

```
ubuntu@ip-172-31-36-100: ~
GNU nano 7.2

#
# Note: The daemon only does rudimentary checking of the client's IP
# address. I would highly recommend adding entries in your /etc/hosts.allow
# file to allow only the specified host to connect to the port
# you are running this daemon on.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd

allowed_hosts=127.0.0.1,34.229.45.75

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

Now restart the NRPE server by this command.

sudo systemctl restart nagios-nrpe-server

```
ubuntu@ip-172-31-36-100: ~$ sudo systemctl restart nagios-nrpe-server
ubuntu@ip-172-31-36-100: ~$
```

Run the following command in the Nagios-host terminal

sudo systemctl status nagios

```
[root@ip-172-31-41-160 nagios-plugins-2.4.11]# sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
   Active: active (running) since Wed 2024-10-02 13:20:17 UTC; 15min ago
     Docs: https://www.nagios.org/documentation
 Main PID: 78778 (nagios)
   Tasks: 6 (limit: 1112)
    Memory: 4.3M
       CPU: 403ms
      CGroup: /system.slice/nagios.service
              └─78778 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 02 13:22:54 ip-172-31-41-160.ec2.internal nagios[78778]: SERVICE NOTIFICATION: nagiosadmin;localhost;Swap Usage;CRITICAL;notify-service-by-email;SWAP CRITICAL - 0% free (0 MB out of 0 MB)
Oct 02 13:22:54 ip-172-31-41-160.ec2.internal nagios[78778]: wproc: NOTIFY job 3 from worker Core Worker 78782 is a non-check helper but exited with return code 127
Oct 02 13:22:54 ip-172-31-41-160.ec2.internal nagios[78778]: wproc: host=localhost; service=Swap Usage; contact=nagiosadmin
Oct 02 13:22:54 ip-172-31-41-160.ec2.internal nagios[78778]: wproc: early timeout=0; exited_ok=1; wait_status=2512; error_code=0;
Oct 02 13:22:54 ip-172-31-41-160.ec2.internal nagios[78778]: wproc: stderr line 01: /bin/sh: line 1: /bin/mail: No such file or directory
Oct 02 13:22:54 ip-172-31-41-160.ec2.internal nagios[78778]: wproc: stderr line 02: /usr/bin/printf: write error: Broken pipe
Oct 02 13:23:13 ip-172-31-41-160.ec2.internal nagios[78778]: SERVICE ALERT: linux-server;Total Processes;OK;HARD;1;PROCS OK: 37 processes with STATE = RSZDT
Oct 02 13:23:50 ip-172-31-41-160.ec2.internal nagios[78778]: SERVICE ALERT: linux-server;Current Load;OK;HARD;1;OK - load average: 0.01, 0.07, 0.04
Oct 02 13:24:28 ip-172-31-41-160.ec2.internal nagios[78778]: SERVICE ALERT: linux-server;Current Users;OK;HARD;1;USERS OK - 2 users currently logged in
Oct 02 13:24:46 ip-172-31-41-160.ec2.internal nagios[78778]: SERVICE ALERT: localhost;Current Users;OK;HARD;1;USERS OK - 2 users currently logged in
Lines 1-26/26 (END)
```

Step 5: Visiting your nagios server using your nagios-host ip address

Open up your browser and look for http://<public_ip_address_of_nagios-host>/nagios

The screenshot shows the Nagios Core 4.5.5 dashboard. At the top right, it displays "Nagios® Core™ Version 4.5.5" and the date "September 17, 2024". A green checkmark indicates "Daemon running with PID 78778". The left sidebar contains a navigation menu with sections like General, Current Status, Service Groups, Reports, and Problems. The "Current Status" section is expanded, showing links for Hosts, Services, Host Groups, and Service Groups. The main content area includes a "Get Started" box with bullet points about monitoring, a "Quick Links" box with links to Nagios documentation and support, and two empty boxes for "Latest News" and "Don't Miss...".

Click on Hosts.

The screenshot shows the "Host Status Details For All Host Groups" table. It lists two hosts: "linux-server" and "localhost", both marked as "UP". The table includes columns for Host, Status, Last Check, Duration, and Status Information. The "Status Information" column for both hosts indicates "PING OK - Packet loss = 0%, RTA = 0.84 ms" and "PING OK - Packet loss = 0%, RTA = 0.04 ms" respectively. The left sidebar shows the "Current Status" menu selected, and the right sidebar has a "Page Tour" button.

Host	Status	Last Check	Duration	Status Information
linux-server	UP	10-02-2024 13:40:17	0d 0h 20m 18s	PING OK - Packet loss = 0%, RTA = 0.84 ms
localhost	UP	10-02-2024 13:40:09	0d 0h 20m 26s	PING OK - Packet loss = 0%, RTA = 0.04 ms

Click on linux-server to view host information

The screenshot shows the Nagios web interface for a host named 'localhost' (linux-server). The main content area displays 'Host Information' and 'Host State Information'. In 'Host State Information', the host status is 'UP' (green) with a duration of 0d 0h 20m 39s. The 'Active Checks' section shows all checks as 'ENABLED'. On the right, there's a 'Host Commands' panel with various options like 'Locate host on map' and 'Disable active checks of this host'. At the bottom, there are 'Host Comments' sections and a system tray with the date and time.

We can even navigate to the services section, which explicitly mentions the status, duration, checks, information about the numerous services present on our hosts

The screenshot shows the Nagios web interface for the 'Services' section. It displays 'Current Network Status' and 'Service Status Details For All Hosts'. The 'Service Status Details' table lists various services for 'localhost' and 'linux-server'. For 'localhost', services include Current Load (OK), Current Users (OK), HTTP (CRITICAL - Connection refused), PING (OK), SSH (OK), Swap Usage (CRITICAL - Swap is either disabled, not present, or of zero size), Total Processes (OK), Current Load (OK), Current Users (OK), HTTP (WARNING - Forbidden), PING (OK), Root Partition (OK), SSH (OK), Swap Usage (CRITICAL - Swap is either disabled, not present, or of zero size), and Total Processes (OK). For 'linux-server', services listed are Current Load (OK), Current Users (OK), HTTP (CRITICAL - Connection refused), PING (OK), Root Partition (OK), SSH (OK), Swap Usage (CRITICAL - Swap is either disabled, not present, or of zero size), Total Processes (OK), Current Load (OK), Current Users (OK), HTTP (WARNING - Forbidden), PING (OK), Root Partition (OK), SSH (OK), Swap Usage (CRITICAL - Swap is either disabled, not present, or of zero size), and Total Processes (OK). The table includes columns for Host, Service, Status, Last Check, Duration, Attempt, and Status Information.

Conclusion: In conclusion, the experiment focused on monitoring ports, services, and a Linux server using Nagios. Through the step-by-step process, we successfully configured Nagios to monitor essential network services on the Linux server. By setting up both the Nagios host and client, we were able to track system performance, ensure service availability, and monitor key metrics like CPU and memory usage.

Experiment 11

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

Theory:

AWS Lambda

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

Lambda Workflow

- **Create a Function:** Write the function code and define its handler (entry point). You can use the AWS Console, CLI, or upload a deployment package.
- **Set Event Sources:** Define how the function is triggered (e.g., when an object is uploaded to S3 or a DynamoDB table is updated).
- **Execution:** When triggered, Lambda runs your function, executes the logic, and automatically scales to handle the incoming event volume.
- **Scaling and Concurrency:** Lambda scales automatically by launching more instances of the function to handle simultaneous invocations. There are also options for configuring **reserved concurrency** to manage traffic.
- **Monitoring and Logging:** Lambda integrates with Amazon CloudWatch for logging and monitoring. Logs for each invocation are sent to CloudWatch, allowing you to track performance and troubleshoot errors.

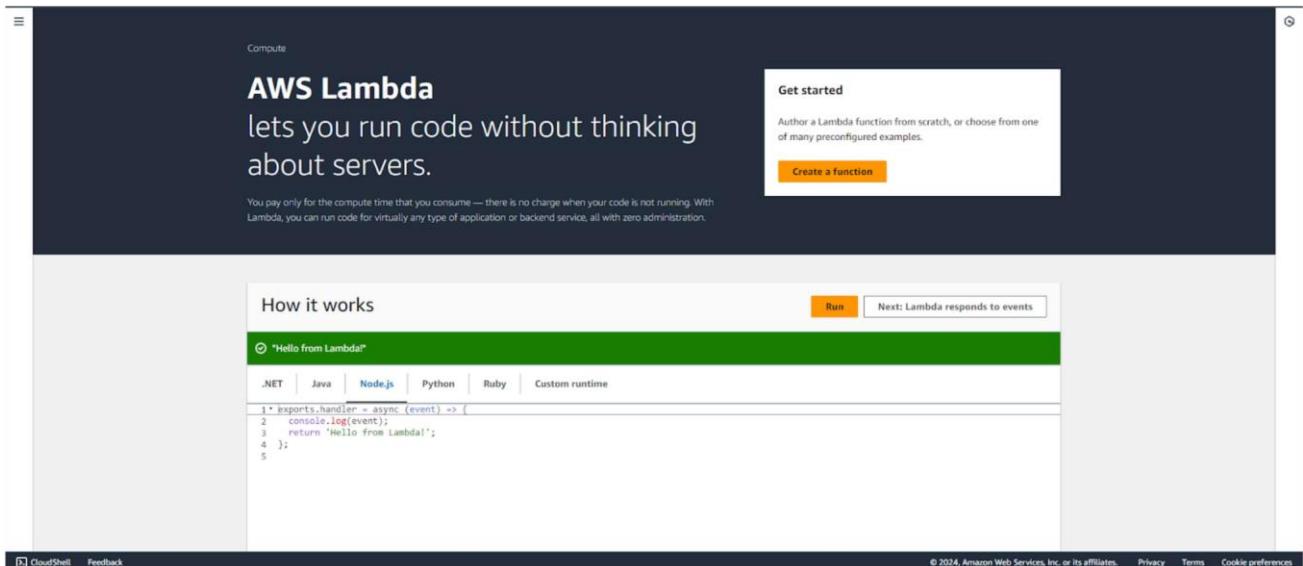
AWS Lambda Functions

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

Prerequisites: AWS Personal/Academy Account

Steps To create the lambda function:

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



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

Lambda > Functions > Create function

Create function info

Choose one of the following options to create your function.

- Author from scratch
Start with a simple Hello World example.
- Use a blueprint
Build a Lambda application from sample code and configuration presets for common use cases.
- Container image
Select a container image to deploy for your function.
- Browse serverless app repository
Deploy a sample Lambda application from the AWS Serverless Application Repository.

Basic information

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

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

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

Architecture Info
Choose the instruction set architecture you want for your function code.
 x86_64
 arm64

Name : Ganesh Gupta

RollNo./Div:13/D15C

Python 3.12

Architecture Info
Choose the instruction set architecture you want for your function code.
 x86_64
 arm64

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

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

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

Lambda will create an execution role named Bhushan_Lamda-role-pbjr1991, with permission to upload logs to Amazon CloudWatch Logs.

► Advanced settings

Cancel Create function

CloudShell Feedback

Lambda > Functions > myLambda

myLambda

Throttle Copy ARN Actions ▾

▼ Function overview [Info](#)

Diagram Template

 myLambda

Layers (0)

+ Add trigger + Add destination

Description

Last modified 4 minutes ago

Function ARN arn:aws:lambda:eu-north-1:860015268757:function:myLambda

Function URL [Info](#)

Code Test Monitor Configuration Aliases Versions

Code source [Info](#)

File Edit Find View Go Tools Window Test Deploy

Upload from ▾

Environment

Go to Anything (Ctrl+P)

lambda_function Environment Vari

```
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
```

Name : Ganesh Gupta

RollNo./Div:13/D15C

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

The screenshot shows the AWS Lambda function configuration interface. The top navigation bar includes tabs for Code, Test, Monitor, Configuration (which is highlighted in blue), Aliases, and Versions. On the left, a sidebar lists General configuration, Triggers, Permissions, Destinations, Function URL, Environment variables, Tags, VPC, RDS databases, and other options. The main content area is titled "General configuration" and contains fields for Description (empty), Memory (128 MB), Timeout (0 min 3 sec), SnapStart (None), and Ephemeral storage (512 MB). An "Edit" button is located in the top right corner of this panel.

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 "Edit basic settings" dialog box. It includes fields for Description (set to "Basic Settings"), Memory (128 MB), Ephemeral storage (512 MB), SnapStart (None), and Timeout (0 min 1 sec). It also includes execution role options: "Use an existing role" (selected) and "Create a new role from AWS policy templates".

Name : Ganesh Gupta

RollNo./Div:13/D15C

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

Test event [Info](#)

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

Test event action

Create new event Edit saved event

Event name

MyEvent

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

Event sharing settings

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

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

Template - optional

hello-world

Event JSON

```
1: [{}]
2:   "key1": "value1",
3:   "key2": "value2",
```

Format JSON

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

Code Test Monitor Configuration Aliases Versions

Code source [Info](#)

Upload from ▾

File Edit Find View Go Tools Window

Test Deploy

Configure test event Ctrl-Shift-C

• (unsaved) test event

Private saved events

MyEvent

Environment

myLambda /

lambda_function.py

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

The screenshot shows the AWS Lambda console interface. At the top, a green banner indicates that the test event 'MyEvent' was successfully saved. Below this, the 'Code source' tab is selected. In the center, there's a 'Test' dropdown menu with 'Execution result' highlighted. The results show a successful deployment with a status code of 200 and a body of 'Hello from Lambda!'. The function logs provide detailed information about the request and response.

Step 5: You can edit your lambda function code. I have changed the code to display the new String.

This screenshot shows the AWS Lambda code editor for the 'lambda_function' function. The code is written in Python and defines a handler function 'lambda_handler'. The 'body' field of the response is now set to 'Hello from D15C-12,15,22!' instead of the previous 'Hello from Lambda!'.

Step 6: Now click on the test and observe the output. We can see the status code 200 and your string output and function logs. On successful deployment.

This screenshot shows the AWS Lambda execution results after redeployment. The test event 'MyEvent' is run again, and the response body is now correctly displayed as 'Hello from D15C-12,15,22!'. The function logs show the deployment details and the successful execution.

Conclusion:

In this experiment, we successfully implemented an AWS Lambda function, covering all the key steps involved. Starting with the function's setup in Python, we configured essential settings such as adjusting the timeout to 1 second. A test event was then created, followed by deploying the function and verifying its output. We also made code modifications to the Lambda function, redeployed it, and observed the real-time effects of these changes. This hands-on experience highlighted the ease and adaptability of AWS Lambda for building serverless applications, enabling developers to concentrate on writing code while AWS handles infrastructure and scalability.

Experiment 12

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

Theory:

AWS Lambda and S3 Integration:

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

Workflow:

1. Create an S3 Bucket:

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

2. Create the Lambda Function:

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

3. Set Up Permissions:

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

4. Configure S3 Trigger:

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

5. Test the Setup:

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

Prerequisites: AWS Personal Account

Steps To create the lambda function:

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

Name	AWS Region	IAM Access Analyzer	Creation date
codepipeline-eu-north-1-823007647292	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1	August 8, 2024, 23:54:38 (UTC+05:30)
codepipeline-us-east-1-934567252759	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 11, 2024, 22:46:59 (UTC+05:30)
elasticbeanstalk-eu-north-1-010928205712	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1	August 9, 2024, 00:03:29 (UTC+05:30)
elasticbeanstalk-us-east-1-010928205712	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 11, 2024, 20:15:18 (UTC+05:30)

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

General configuration

AWS Region
US East (N. Virginia) us-east-1

Bucket type [Info](#)

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

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

Bucket name [Info](#)
wearekcs

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

Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied.
[Choose bucket](#)

Format: s3://bucket/prefix

Object Ownership Info

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

 ACLs disabled (recommended)

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

 ACLs enabled

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

Object Ownership

Bucket owner enforced

Block Public Access settings for this bucket

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

 Block all public access

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

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

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

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

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

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

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

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

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

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

Amazon S3 > Buckets

Account snapshot - updated every 24 hours [Edit filters](#) [View Storage Lam dashboard](#)

Storage info provided reflects three storage classes and is subject to change. Learn more

[General purpose buckets](#) [Directory buckets](#)

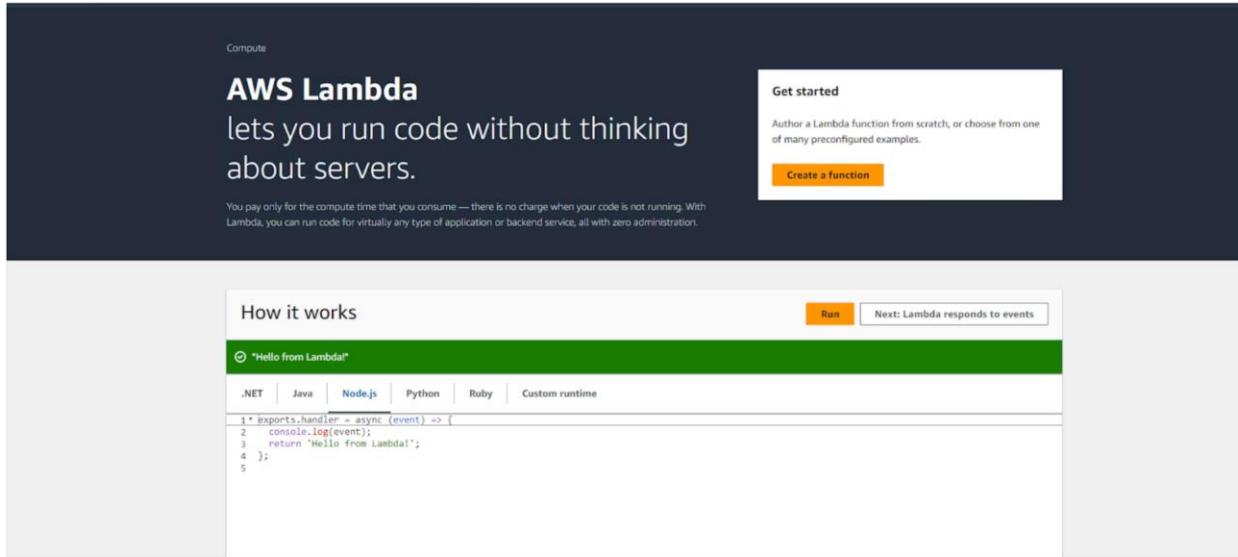
General purpose buckets (1) (1) [AWS Region](#)

Bucket last modified: 30 days ago (11:40 AM UTC+05:30)

Find buckets by name:

Name	AWS Region	Object storage class	Last modified	Creation date
awsbucket	US East (N. Virginia) (selected)	Standard	November 1, 2024, 11:40:40 (UTC+05:30)	November 1, 2024, 11:40:40 (UTC+05:30)

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



Step 4: Now Give a name to your Lambda function, Select the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby. So will select Python 3.12

, Architecture as x86, and Execution role to Create a new role with basic Lambda permissions.

The screenshot shows the 'Create function' wizard in the AWS Lambda console. The top navigation bar shows 'Lambda > Functions > Create function'. The main title is 'Create function' with an 'Info' link. Below the title, a note says 'Choose one of the following options to create your function.' There are four options:

- Author from scratch**: Start with a simple Hello World example.
- Use a blueprint**: Build a Lambda application from sample code and configuration presets for common use cases.
- Container image**: Select a container image to deploy for your function.
- Browse serverless app repository**: Deploy a sample Lambda application from the AWS Serverless Application Repository.

The 'Basic information' step is active. It includes fields for 'Function name' (set to 'MyLambda'), 'Runtime' (set to 'Python 3.12'), and 'Architecture' (set to 'x86_64').

Name: Ganesh Gupta

DIV/Roll no.:D15C/13

Python 3.12

Architecture [Info](#)
Choose the instruction set architecture you want for your function code.
 x86_64
 arm64

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

▼ Change default execution role

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

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

Lambda will create an execution role named Bhushan_Lambda-role-pbjr1991, with permission to upload logs to Amazon CloudWatch Logs.

► Advanced settings

[Cancel](#) [Create function](#)

[CloudShell](#) [Feedback](#)

Lambda > Functions > myLambda

myLambda

Throttle [Copy ARN](#) [Actions ▾](#)

▼ Function overview [Info](#)

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

myLambda

Layers (0)

+ Add trigger + Add destination

Description:
-

Last modified
4 minutes ago

Function ARN
 arn:aws:lambda:eu-north-1:860015268757:function:myLambda

Function URL: [Info](#)
-

Code Test Monitor Configuration Aliases Versions

Code source [Info](#) Upload from ▾

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

Environment ☰

Go to Anything (Ctrl-P) ✖

lambda_function Environment Var +

myLambda / ⚙️

lambda_function.py

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

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

The screenshot shows the AWS Lambda Configuration interface. The top navigation bar includes tabs for Code, Test, Monitor, Configuration (which is highlighted in blue), Aliases, and Versions. On the left, a sidebar lists categories: General configuration, Triggers, Permissions, Destinations, Function URL, Environment variables, Tags, VPC, and RDS databases. The main content area is titled "General configuration" with an "Edit" button. It contains fields for "Description" (empty), "Memory" (128 MB), "Timeout" (0 min 3 sec), and "SnapStart" (None). A note indicates that memory is allocated CPU proportional to the memory configured.

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 "Edit basic settings" page for a Lambda function named "Bhushan_Lambda". The top navigation path is Lambda > Functions > Bhushan_Lambda > Edit basic settings. The main form is titled "Basic settings" and includes the following fields:

- Description**: An optional field containing "Basic Settings".
- Memory**: Set to 128 MB. Info: Your function is allocated CPU proportional to the memory configured.
- Ephemeral storage**: Set to 512 MB. Info: You can configure up to 10 GB of ephemeral storage (/tmp) for your function.
- SnapStart**: Set to None. Info: Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized.
- Timeout**: Set to 0 min 1 sec.
- Execution role**: Options include "Use an existing role" (selected) and "Create a new role from AWS policy templates". Info: Choose a role that defines the permissions of your function.

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

Name: Ganesh Gupta

DIV/Roll no.:D15C/13

Test event Info

Save

Test

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

MyEvent

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

10 of 10 | Page

hello-world

Event JS

Event JSON Format JSON

```
1 [ ]  
2   "key1": "value1",  
3   "key2": "value2",
```

Services Search [Alt+5]

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

Template - optional

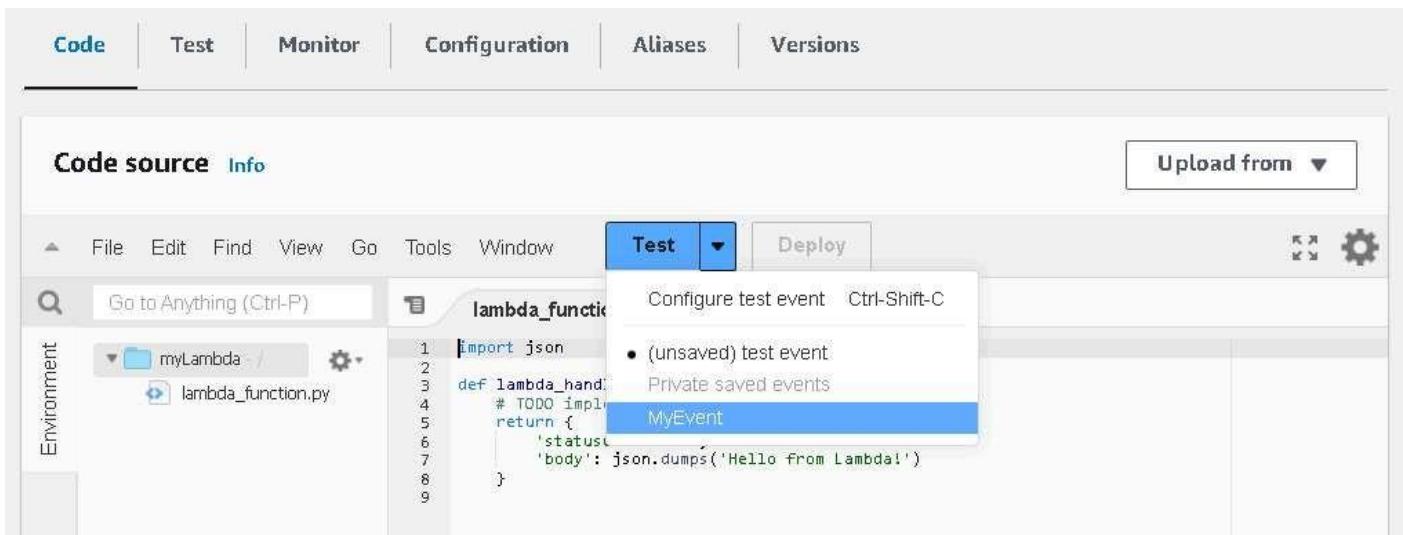
s3-put

Event JSON Format JSON

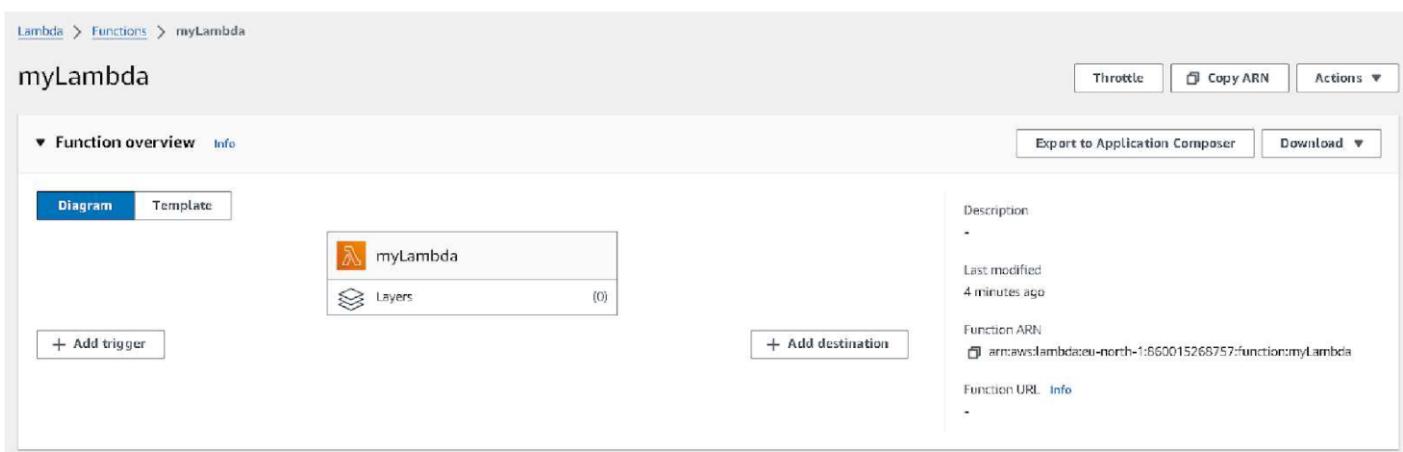
```
2 * "Records": [
3 *   {
4 *     "eventVersion": "2.0",
5 *     "eventSource": "aws:s3",
6 *     "awsRegion": "us-east-1",
7 *     "eventTime": "1970-01-01T00:00:00.000Z",
8 *     "eventName": "ObjectCreated:Put",
9 *     "userIdentity": {
10 *       "principalId": "EXAMPLE"
11 *     },
12 *     "requestParameters": {
13 *       "sourceIPAddress": "127.0.0.1"
14 *     },
15 *     "responseElements": {
16 *       "x-amz-request-id": "EXAMPLE123456789",
17 *       "x-amz-id-2": "EXAMPLE123/5678abcdefghijklmklambdaisawesome/mnopqrstuvwxyzABCDEFGH"
18 *     },
19 *     "s3": {
20 *       "s3SchemaVersion": "1.0",
21 *       "configurationId": "testConfigRule",
22 *       "bucket": {
23 *         "name": "example-bucket",
24 *         "ownerIdentity": {
25 *           "principalId": "EXAMPLE"
26 *         },
27 *         "arn": "arn:aws:s3:::example-bucket"
28 *       },
29 *       "object": {
30 *         "key": "test%2Fkey",
31 *         "size": 1024,
32 *       }
33 *     }
34 *   }
35 * ]
36 *
```

1:1 JSON Spaces: 2

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



Step 7: Now In the Lambda function click on add tigger.



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

Lambda > Add triggers

Add trigger

Trigger configuration [Info](#)

S3 [Info](#) [Edit](#) [Delete](#)

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

[X](#) [G](#)

Bucket region: us-east-1

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

All object create events [X](#)

Prefix - optional
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters. Any special characters must be URL encoded.

Suffix - optional
Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters. Any special characters must be URL encoded.

Recursive invocation

Function overview [Info](#)

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

Diagram [Template](#)

 **myLambda**

[Layers](#) (0)

[+ Add destination](#)

[+ Add trigger](#)

Description
Basic Settings

Last modified
1 hour ago

Function ARN
[arn:aws:lambda:us-east-1:1010928205712:function:Bhushan_Lambda](#)

Function URL [Info](#)

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

Configuration

- [Triggers](#) [\(1\) Info](#)
- [Integrations](#)
- [Lambda layers](#)
- [Environment variables](#)
- [File](#)
- [VPC](#)
- [AWS Outposts](#)
- [Monitoring and optimization tools](#)
- [Dependency and resource detection](#)
- [Asynchronous invocation](#)
- [CloudWatch Metrics](#)
- [Metrics](#)
- [CloudWatch Metrics Insights](#)
- [CloudWatch Metrics Insights Insights](#)
- [CloudWatch Metrics Insights Metrics](#)

Triggers (1) Info

Edge

 **s3://wwweks** [arn:aws:s3:::wwweks](#)

[Details](#)

[G](#) [Permissions](#) [Edit](#) [Delete](#) [Add trigger](#)

Name: Ganesh Gupta

DIV/Roll no.:D15C/13

Step 8: Now Write code that logs a message like “An Image has been added” when triggered.

Save the file and click on deploy.

The screenshot shows a code editor interface for a Lambda function named 'lambda_function'. The code is written in Python and defines a handler function 'lambda_handler' that prints a message to the log when an S3 object is uploaded. The code is as follows:

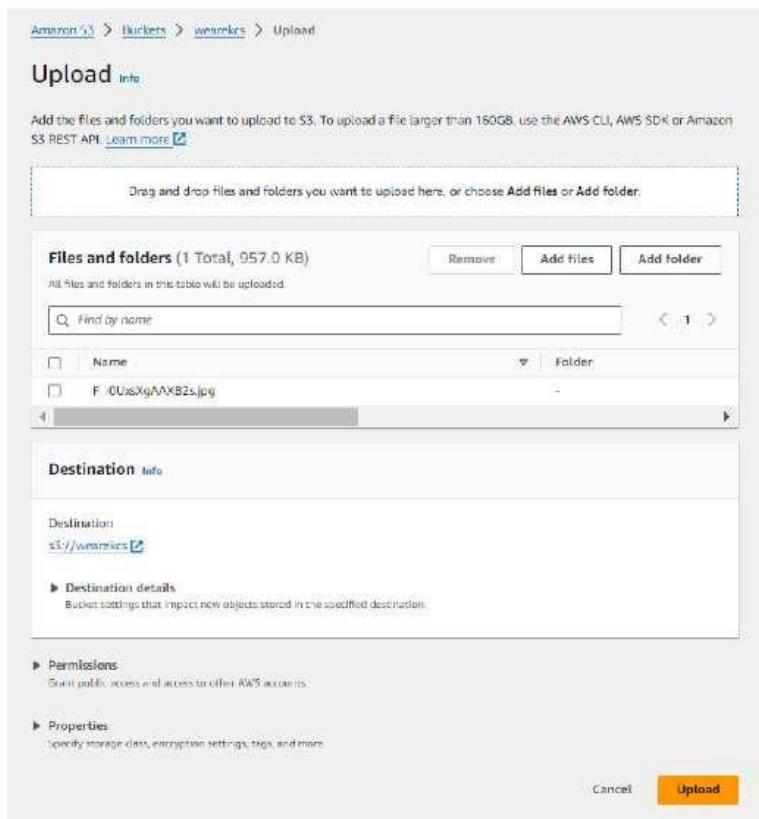
```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     bucket_name= event['Records'][0]['s3']['bucket']['name']
6     object_key= event['Records'][0]['s3']['object']['key']
7
8     print(f"An Image has been added to the bucket {bucket_name} : {object_key}")
9     return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

```

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

```

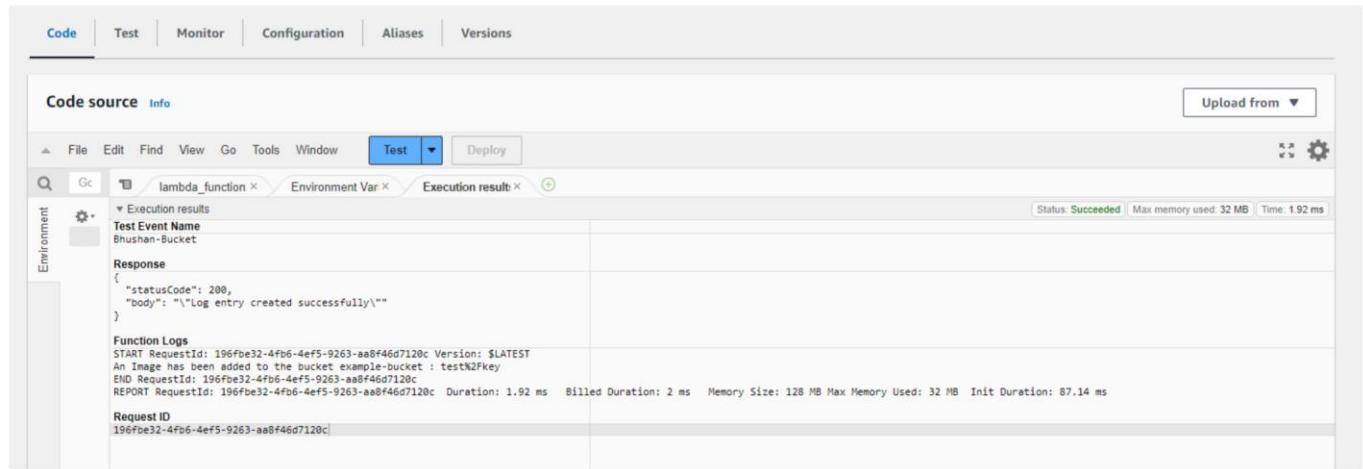
Step 9: Now upload any image to the bucket.



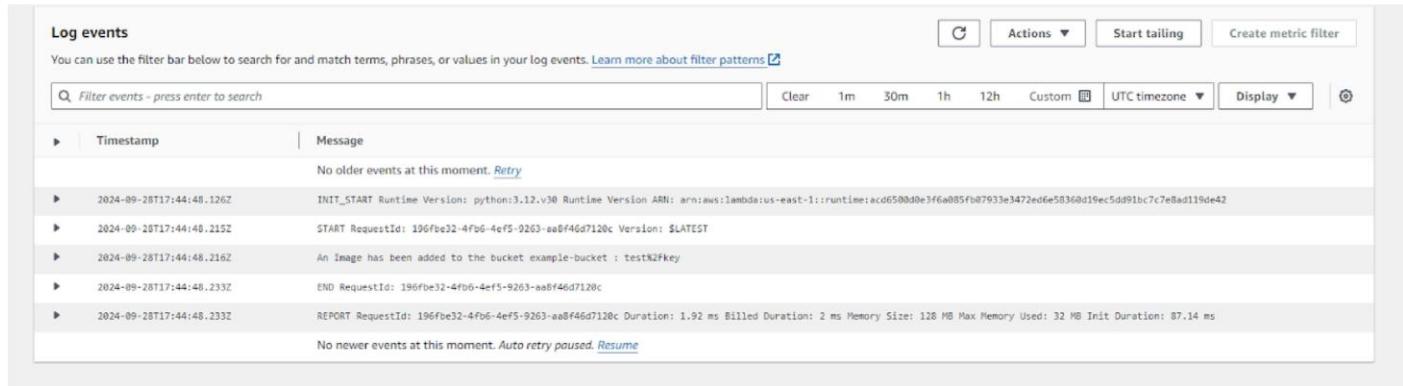
Name: Ganesh Gupta

DIV/Roll no.:D15C/13

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



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



Conclusion:

In this experiment, we successfully created an AWS Lambda function that logs a message when an image is uploaded to an S3 bucket. It is important to note that we have to select S3-put template in the event otherwise code will give an error. The function was successfully triggered by S3 object uploads, validating the functionality of Lambda's event-driven architecture. This experiment demonstrated how Lambda can efficiently respond to S3 events and how to troubleshoot common issues with event structure.