

Advance Devops
Hitesh Rohra
D15A - 47

Experiment No 1

Screenshots for hosting webpage on EC2 instance :-

```

AWS Services Search [Alt+S] N. Virginia vocabs/user3387419=TALREJA_JAI_@ 5672-7063-6093 ▾

Scanning candidates...
scanning linux images...

Pending kernel upgrade!
Running kernel version:
  6.8.0-1009-aws
Diagnostics:
  The currently running kernel version is not the expected kernel version 6.8.0-1013-aws.

Restarting the system to load the new kernel will not be handled automatically, so you should consider rebooting.

Restarting services...
systemctl restart multipathd.service polkit.service rsyslog.service udisks2.service

Service restarts being deferred:
systemctl restart ModemManager.service
/etc/needrestart/restart.d/dbus.service
systemctl restart networkd-dispatcher.service
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service

No containers need to be restarted.

User sessions running outdated binaries:
ubuntu @ session #1: apt[1706], bash[1374], sshd[857]
ubuntu @ user manager service: systemd[1204]

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-42-7:~# 

```

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

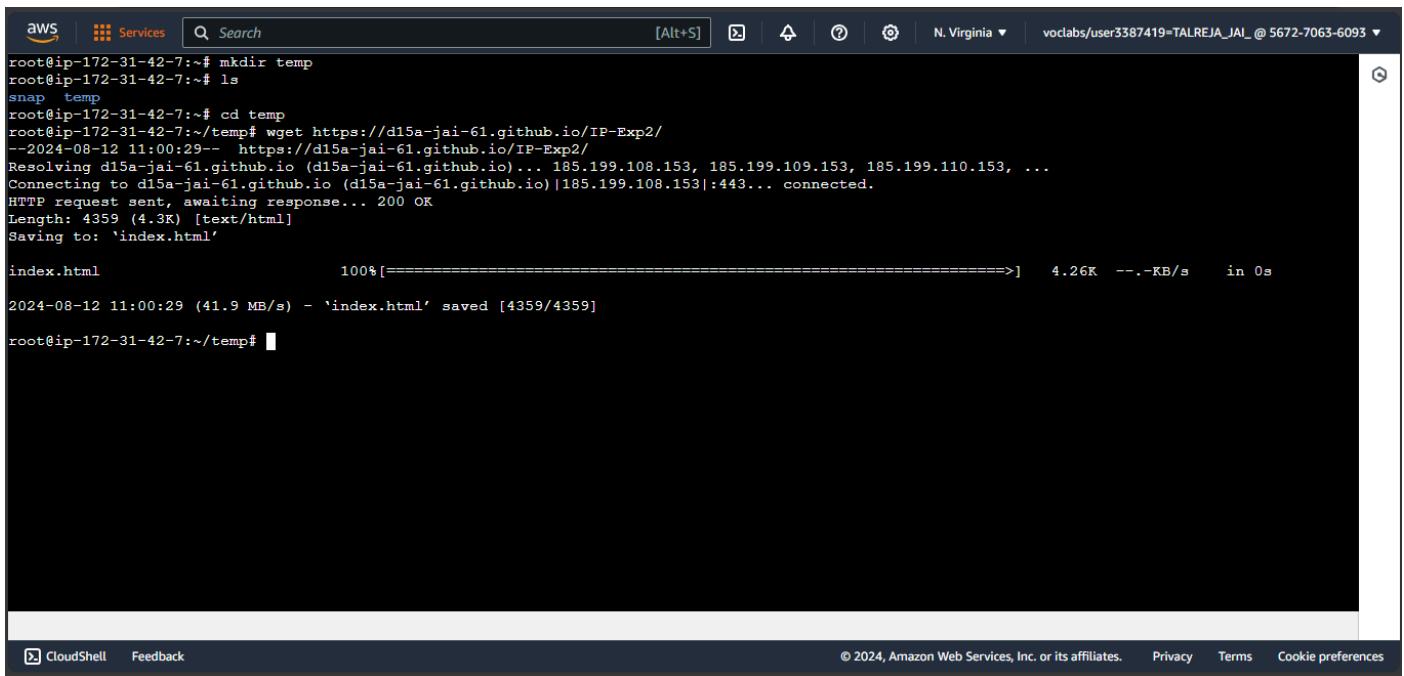
```

root@ip-172-31-42-7:~# apt install apache2
E: Package 'yum' has no installation candidate
root@ip-172-31-42-7:~# apt install apache2

root@ip-172-31-42-7:~# apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-db-sqlite3 libaprutil1-ldap libaprutil1t64 liblua5.4-0 ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-db-sqlite3 libaprutil1-ldap libaprutil1t64 liblua5.4-0 ssl-cert
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.
Need to get 2083 kB of archives.
After this operation, 8094 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libapr1t64 amd64 1.7.2-3.1build2 [107 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1t64 amd64 1.6.3-1.1ubuntu7 [91.9 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-db-sqlite3 amd64 1.6.3-1.1ubuntu7 [11.2 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-ldap amd64 1.6.3-1.1ubuntu7 [9116 B]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblua5.4-0 amd64 5.4.6-3build2 [166 kB]
Progress: [ 98% ] [##########################################]
..] :7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-data all 2.4.58-1ubuntu8.4 [163 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-utils amd64 2.4.58-1ubuntu8.4 [97.1 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2 amd64 2.4.58-1ubuntu8.4 [90.2 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 ssl-cert all 1.1.2ubuntu1 [17.8 kB]
Fetched 2083 kB in 0s (25.5 MB/s)
Preconfiguring packages ...
Scanning processes...
root@ip-172-31-42-7:~# 

```

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



AWS CloudShell terminal window showing the execution of a wget command to download index.html from a GitHub repository. The terminal interface includes tabs for Services, Search, and various system icons. The wget progress bar shows 100% completion at 4.26K/s.

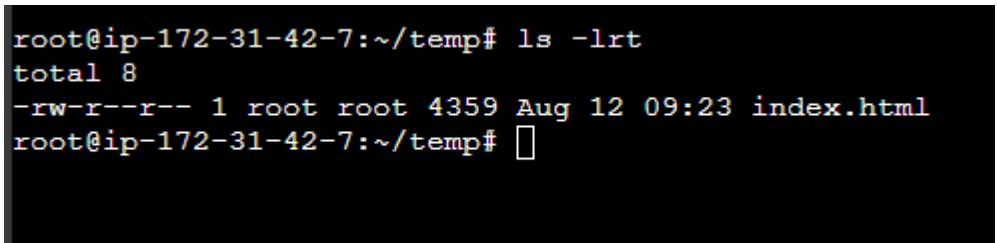
```
root@ip-172-31-42-7:~# mkdir temp
root@ip-172-31-42-7:~# ls
snap temp
root@ip-172-31-42-7:~# cd temp
root@ip-172-31-42-7:~/temp# wget https://d15a-jai-61.github.io/IP-Exp2/
--2024-08-12 11:00:29-- https://d15a-jai-61.github.io/IP-Exp2/
Resolving d15a-jai-61.github.io (d15a-jai-61.github.io)... 185.199.108.153, 185.199.109.153, 185.199.110.153, ...
Connecting to d15a-jai-61.github.io (d15a-jai-61.github.io)|185.199.108.153|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4359 (4.3K) [text/html]
Saving to: 'index.html'

index.html          100%[=====] 4.26K --.-KB/s   in 0s

2024-08-12 11:00:29 (41.9 MB/s) - 'index.html' saved [4359/4359]

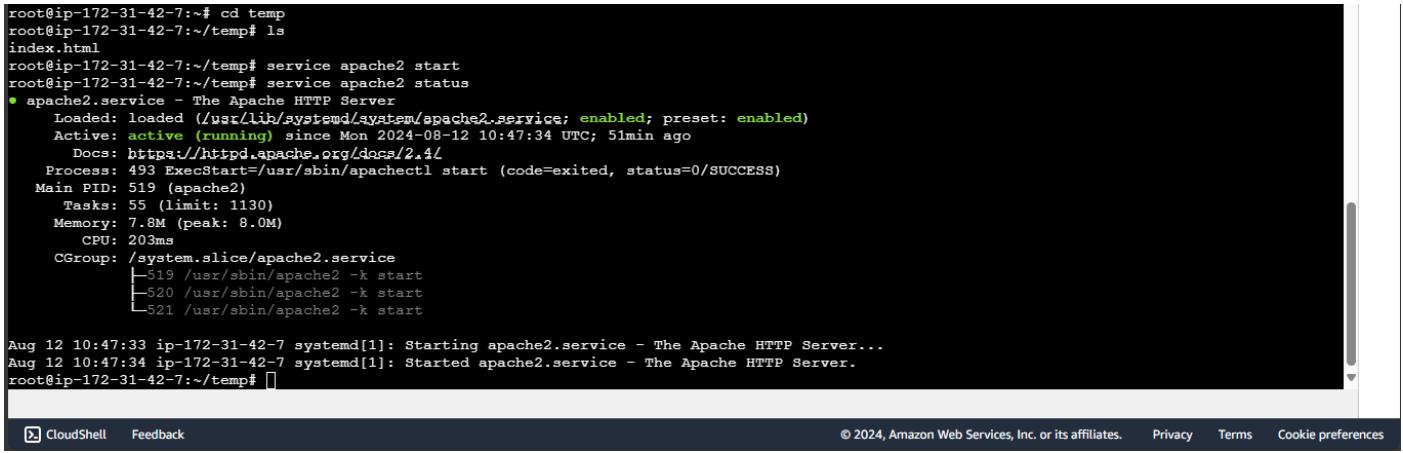
root@ip-172-31-42-7:~/temp#
```

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



AWS CloudShell terminal window showing the output of the ls -lrt command, which lists index.html as the most recent file.

```
root@ip-172-31-42-7:~/temp# ls -lrt
total 8
-rw-r--r-- 1 root root 4359 Aug 12 09:23 index.html
root@ip-172-31-42-7:~/temp#
```



AWS CloudShell terminal window showing the start and status of the Apache2 service using systemctl commands. It also displays logs from the Apache server starting up.

```
root@ip-172-31-42-7:~# cd temp
root@ip-172-31-42-7:~/temp# ls
index.html
root@ip-172-31-42-7:~/temp# service apache2 start
root@ip-172-31-42-7:~/temp# service apache2 status
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Mon 2024-08-12 10:47:34 UTC; 51min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 493 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 519 (apache2)
    Tasks: 55 (limit: 1130)
   Memory: 7.8M (peak: 8.0M)
      CPU: 203ms
     CGroup: /system.slice/apache2.service
             └─519 /usr/sbin/apache2 -k start
                  ├─520 /usr/sbin/apache2 -k start
                  ├─521 /usr/sbin/apache2 -k start

Aug 12 10:47:33 ip-172-31-42-7 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Aug 12 10:47:34 ip-172-31-42-7 systemd[1]: Started apache2.service - The Apache HTTP Server.
root@ip-172-31-42-7:~/temp#
```

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

AWS Services Search [Alt+S] N. Virginia vodlabs/user3387419=TALREJA_JAI_@ 5672-7063-6093 ▾

EC2 > Security Groups > sg-0234926154e2078d6 > Edit inbound rules

Edit inbound rules Info

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

| Inbound rules <small>Info</small> | Type <small>Info</small> | Protocol <small>Info</small> | Port range <small>Info</small> | Source <small>Info</small> | Description - optional <small>Info</small> |
|-----------------------------------|--------------------------|------------------------------|--------------------------------|----------------------------|--|
| sgr-0a05c7d3f2d659b92 | SSH | TCP | 22 | Cust... ▾ | <input type="text"/> Q 0.0.0.0/0 X |
| sgr-00956191191b304fc | HTTP | TCP | 80 | Cust... ▾ | <input type="text"/> http 0.0.0.0/0 X |

Add rule

⚠️ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

InPrivate Launch AWS Academy Lesson Instances | EC2 | us-east-1 EC2 Instance Connect | us how to host website on Petco company profile + - A ⚡ ⚡ ...

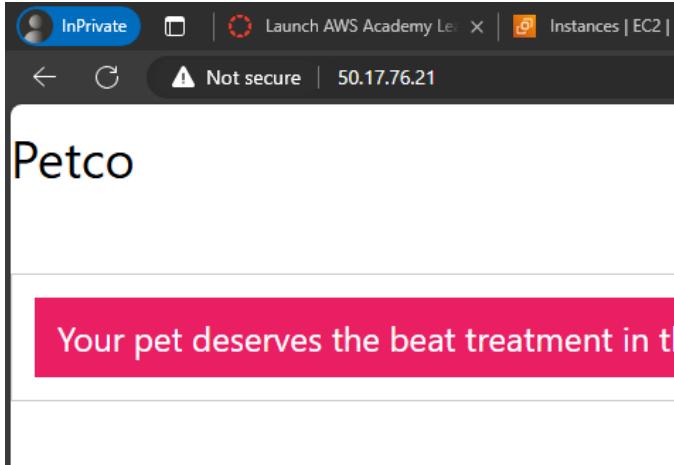
Not secure | 50.17.76.21

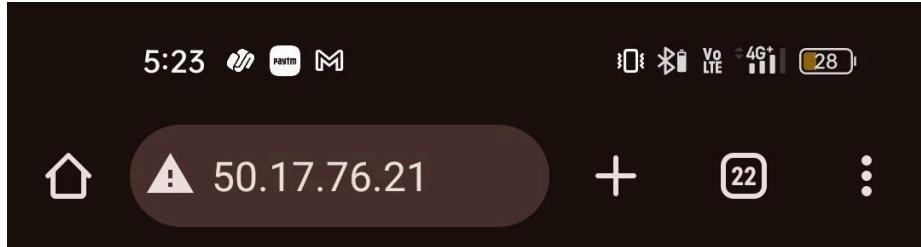
Petco

Your pet deserves the beat treatment in the world !

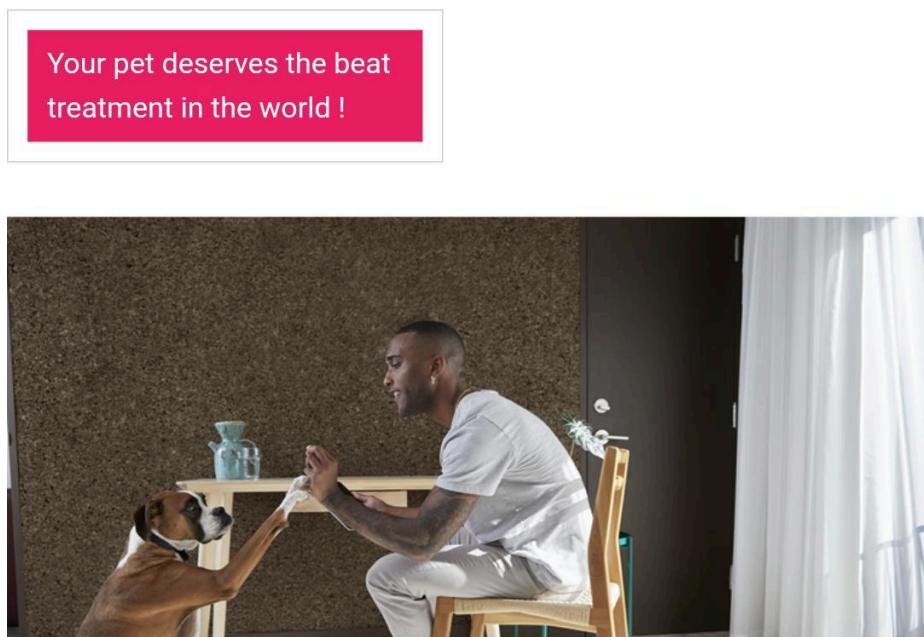


17:19 ENG IN 12-08-2024





Petco



About us

Petco is a category-defining health and wellness company focused on improving the lives of pets, pet parents and our own Petco partners. Since our founding in 1965, we've been trailblazing new standards in pet care, delivering comprehensive wellness solutions through our products and services, and creating communities that deepen the pet-pet parent bond.

We employ more than 29,000 partners nationwide and operate more than 1,500 Petco locations across the U.S., Mexico and Puerto Rico — including a growing network of more than 200 in-store veterinary hospitals — and offer a complete online resource for pet health and wellness at petco.com and on the Petco app.

In tandem with Petco Love, an independent nonprofit organization, we have helped find homes for more than 7 million animals through in-store adoption events.

Working at Petco



Experiment No 2

Screenshots for setting up and deploying Elastic Beanstalk application :-

Configure environment

Environment tier

- Web server environment
- Worker environment

Application information

Application name: WebServer

Step 1: Configure environment

Step 2: Configure service access

Step 3 - optional: Set up networking, database, and tags

Step 4 - optional: Configure instance traffic and scaling

Step 5 - optional: Configure updates, monitoring, and logging

Step 6: Review

Platform type

Managed platform

Platforms published and maintained by Amazon Elastic Beanstalk. [Learn more](#)

Custom platform

Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform

Node.js

Platform branch

Node.js 20 running on 64bit Amazon Linux 2023

Platform version

6.2.0 (Recommended)

Application code

Sample application

Existing version

Application versions that you have uploaded.

Upload your code

Upload a source bundle from your computer or copy one from Amazon S3.

Presets Info

Start from a preset that matches your use case or choose custom configuration to unset recommended values and use the service's default values.

Configuration presets

- Single instance (free tier eligible)
- Single instance (using spot instance)
- High availability
- High availability (using spot and on-demand instances)
- Custom configuration

Cancel**Next****Service access**

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

Service role

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

Existing service roles

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

▼


EC2 key pair

Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#) 

▼


EC2 instance profile

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

▼


[View permission details](#)**Cancel****[Skip to review](#)****[Previous](#)****Next**

Set up networking, database, and tags - optional Info

Virtual Private Cloud (VPC)

VPC

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

vpc-0b4df95bddc923aea | (172.31.0.0/16)

-

vpc-0b4df95bddc923aea | (172.31.0.0/16) ✓

Instance settings

Choose a subnet in each AZ for the instances that run your application. To avoid exposing your instances to the Internet, run your instances in private subnets and load balancer in public subnets. To run your load balancer and instances in the same public subnets, assign public IP addresses to the instances. [Learn more](#)

Public IP address

Assign a public IP address to the Amazon EC2 instances in your environment.

Activated

Instance subnets

Instance settings

Choose a subnet in each AZ for the instances that run your application. To avoid exposing your instances to the Internet, run your instances in private subnets and load balancer in public subnets. To run your load balancer and instances in the same public subnets, assign public IP addresses to the instances. [Learn more](#)

Public IP address

Assign a public IP address to the Amazon EC2 instances in your environment.

Activated

Instance subnets

Filter instance subnets

| | Availability Zone | Subnet | CIDR | Name |
|-------------------------------------|-------------------|----------------------|----------------|------|
| <input checked="" type="checkbox"/> | us-east-1e | subnet-0098f3bf4... | 172.31.48.0/20 | |
| <input type="checkbox"/> | us-east-1f | subnet-03b751cf5... | 172.31.64.0/20 | |
| <input checked="" type="checkbox"/> | us-east-1a | subnet-05ff6de98... | 172.31.16.0/20 | |
| <input type="checkbox"/> | us-east-1c | subnet-089dbc2d1... | 172.31.0.0/20 | |
| <input type="checkbox"/> | us-east-1b | subnet-0d4591b4f... | 172.31.32.0/20 | |
| <input type="checkbox"/> | us-east-1d | subnet-0f341fff62... | 172.31.80.0/20 | |

| | Subnet ID | Range |
|--------------------------|------------|-------------------------------------|
| <input type="checkbox"/> | us-east-1f | subnet-03b751cf5... 172.31.64.0/20 |
| <input type="checkbox"/> | us-east-1a | subnet-05ff6de98... 172.31.16.0/20 |
| <input type="checkbox"/> | us-east-1c | subnet-089dbc2d1... 172.31.0.0/20 |
| <input type="checkbox"/> | us-east-1b | subnet-0d4591b4f... 172.31.32.0/20 |
| <input type="checkbox"/> | us-east-1d | subnet-0f341fff62... 172.31.80.0/20 |

Enable database

Restore a snapshot - *optional*

Restore an existing snapshot from a previously used database.

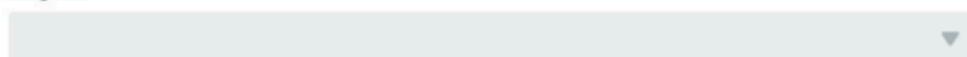
Snapshot

None

Database settings

Choose an engine and instance type for your environment's database.

Engine



Engine version

Tags

Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the resource and is case-sensitive. [Learn more](#) 

No tags associated with the resource.

[Add new tag](#)

You can add 50 more tags.

[Cancel](#)

[Skip to review](#)

[Previous](#)

[Next](#)

Configure instance traffic and scaling - optional Info

▼ Instances Info

Configure the Amazon EC2 instances that run your application.

Root volume (boot device)

Root volume type

General Purpose (SSD) 

(Container default)

Magnetic

hed to each instance.

General Purpose (SSD)  GB

General Purpose 3(SSD)

General Purpose (SSD)

Provisioned IOPS (SSD)

ded IOPS (SSD) volume.

 IOPS

Throughput

The desired throughput to provision for the Amazon EBS root volume attached to your environment's EC2 instance

125



MiB/s

Root volume type

General Purpose (SSD) 

Size

The number of gigabytes of the root volume attached to each instance.

8



GB

IOPS

Input/output operations per second for a provisioned IOPS (SSD) volume.



IOPS

Throughput

The desired throughput to provision for the Amazon EBS root volume attached to your environment's EC2 instance

125



MiB/s

Amazon CloudWatch monitoring

The time interval between when metrics are reported from the EC2 instances

Monitoring interval

5 minute 

Instance metadata service (IMDS)

Your environment's platform supports both IMDSv1 and IMDSv2. To enforce IMDSv2, deactivate IMDSv1. [Learn more](#)

IMDSv1

With the current setting, the environment enables only IMDSv2.

Deactivated

EC2 security groups

Select security groups to control traffic.

EC2 security groups (3)

C

Filter security groups

| <input type="checkbox"/> | Group name | ▲ | Group ID | ▼ | Name | ▼ |
|-------------------------------------|-----------------|---|----------------------|---|------|---|
| <input checked="" type="checkbox"/> | default | | sg-0b6f2684b3eea9987 | | | |
| <input type="checkbox"/> | launch-wizard-1 | | sg-0234926154e2078d6 | | | |
| <input type="checkbox"/> | launch-wizard-2 | | sg-0f76834772eb69439 | | | |

▼ Capacity [Info](#)

Configure the compute capacity of your environment and auto scaling settings to optimize the number of instances used.

Auto scaling group

Environment type

Select a single-instance or load-balanced environment. You can develop and test an application in a single-instance environment to save costs and then upgrade to a load-balanced environment when the application is ready for production. [Learn more](#)

Single instance



Instances

1



Min

1



Max

Fleet composition

Spot instances are launched at the lowest available price. [Learn more](#)

On-Demand instance

Spot instance

Maximum spot price

The maximum price per instance-hour, in USD, that you're willing to pay for a Spot Instance. Setting a custom price limits your chances to fulfill your target capacity using Spot instances.

Default

Set your maximum price

Instance types

Add instance types for your fleet. Change the order that the instances are in to set the preferred launch order. This only affects On-Demand instances. We recommend you include at least two instance types. [Learn more](#)

Choose x86 instance types ▾

t2.small X

AMI ID

Elastic Beanstalk selects a default Amazon Machine Image (AMI) for your environment based on the Region, platform version, and processor architecture that you choose. [Learn more](#)

ami-0b4a9cc2fba693a25

Availability Zones

Number of Availability Zones (AZs) to use.

Any ▾

Placement

Specify Availability Zones (AZs) to use.

Choose Availability Zones (AZs) ▾

Scaling cooldown

360 ▾ seconds

Cancel

Skip to review

Previous

Next

Root volume (boot device)**Root volume type**

General Purpose (SSD) ▾

Size

The number of gigabytes of the root volume attached to each instance.

8 ▾ GB

✖ Size must be between 10 and 16384.

Root volume type

General Purpose (SSD) ▾

Size

The number of gigabytes of the root volume attached to each instance.

16 ▾ GB

Configure updates, monitoring, and logging - optional Info

▼ Monitoring Info

Health reporting

Enhanced health reporting provides free real-time application and operating system monitoring of the instances and other resources in your environment. The **EnvironmentHealth** custom metric is provided free with enhanced health reporting. Additional charges apply for each custom metric. For more information, see [Amazon CloudWatch Pricing](#)

System

- Basic
- Enhanced

Health event streaming to CloudWatch Logs

Configure Elastic Beanstalk to stream environment health events to CloudWatch Logs. You can set the retention up to a maximum of ten years and configure Elastic Beanstalk to delete the logs when you terminate your environment.

Log streaming

- Activated (standard CloudWatch charges apply.)

Retention

7

Lifecycle

▼ Managed platform updates Info

Activate managed platform updates to apply platform updates automatically during a weekly maintenance window that you choose. Your application stays available during the update process.

Managed updates

- Activated

Weekly update window

Tuesday at : UTC

Update level

Minor and patch

Instance replacement

If enabled, an instance replacement will be scheduled if no other updates are available.

- Activated

Activated

Instance log streaming to CloudWatch logs

Configure the instances in your environment to stream logs to CloudWatch logs. You can set the retention to up to 10 years and configure Elastic Beanstalk to delete the logs when you terminate your environment. [Learn more](#)

Log streaming

(standard CloudWatch charges apply.)

Activated

Retention

7



Lifecycle

Keep logs after terminating envir...



Environment properties

The following properties are passed in the application as environment properties. [Learn more](#)

No environment properties have been configured.

[Add environment property](#)

[Cancel](#)

[Previous](#)

[Next](#)

Elastic Beanstalk application created, review screen before finalization :-

Review [Info](#)

Step 1: Configure environment [Edit](#)

Environment information

| | |
|---|--------------------|
| Environment tier | Application name |
| Web server environment | D15A-Jai-61 |
| Environment name | Application code |
| D15A-Jai-61-env | Sample application |
| Platform | |
| arn:aws:elasticbeanstalk:us-east-1::platform/Node.js 20 | |
| running on 64bit Amazon Linux 2023/6.2.0 | |

Step 2: Configure service access [Edit](#)

Service access [Info](#)

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

Step 2: Configure service access [Edit](#)

Service access [Info](#)

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

No options configured

Step 3: Set up networking, database, and tags [Edit](#)

Networking, database, and tags [Info](#)

Configure VPC settings, and subnets for your environment's EC2 instances and load balancer. Set up an Amazon RDS database that's integrated with your environment.

Network

| | | |
|-----------------------|-------------------|---|
| VPC | Public IP address | Instance subnets |
| vpc-0b4df95bddc923aea | true | subnet-0098f3bf430c41040,subnet-05ff6de9850aa5577 |

Tags

Step 4: Configure instance traffic and scaling**Edit****Instance traffic and scaling** [Info](#)

Customize the capacity and scaling for your environment's instances. Select security groups to control instance traffic. Configure the software that runs on your environment's instances by setting platform-specific options.

Instances

| | | |
|----------------------------|---------------|-------------|
| Root volume type | Instance size | IMDSv1 |
| gp2 | 16 | Deactivated |
| EC2 Security Groups | | |
| sg-0b6f2684b3eea9987 | | |

Capacity

| | | |
|----------------------|----------------------|-----------------------|
| Environment type | Fleet composition | On-demand base |
| Single instance | On-Demand instance | 0 |
| On-demand above base | Capacity rebalancing | Scaling cooldown |
| 0 | Deactivated | 360 |
| Processor type | Instance types | AMI ID |
| x86_64 | t2.small | ami-0b4a9cc2fba693a25 |

Step 5: Configure updates, monitoring, and logging**Edit****Updates, monitoring, and logging** [Info](#)

Define when and how Elastic Beanstalk deploys changes to your environment. Manage your application's monitoring and logging settings, instances, and other environment resources.

Monitoring

| | | |
|---------------|--------------------------------------|---|
| System | Cloudwatch custom metrics - instance | Cloudwatch custom metrics - environment |
| basic | — | — |
| Log streaming | Retention | Lifecycle |
| Deactivated | 7 | false |

Updates

| | | |
|---------------------|-----------------------|----------------------------|
| Managed updates | Deployment batch size | Deployment batch size type |
| Deactivated | 100 | Percentage |
| Command timeout | Deployment policy | Health threshold |
| 600 | AllAtOnce | Ok |
| Ignore health check | | Instance replacement |

| Ignore health check | Instance replacement | | | | | | | |
|--|----------------------|--------------|-----|-------|---------------------------|--|---|--|
| false | false | | | | | | | |
| Platform software | | | | | | | | |
| Lifecycle | Log streaming | Proxy server | | | | | | |
| false | Deactivated | nginx | | | | | | |
| Logs retention | Rotate logs | Update level | | | | | | |
| 7 | Deactivated | minor | | | | | | |
| X-Ray enabled | | | | | | | | |
| Deactivated | | | | | | | | |
| Environment properties | | | | | | | | |
| <table border="1"> <thead> <tr> <th>Key</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td colspan="2" style="text-align: center;">No environment properties</td> </tr> <tr> <td colspan="2" style="text-align: center;">There are no environment properties defined</td> </tr> </tbody> </table> | | | Key | Value | No environment properties | | There are no environment properties defined | |
| Key | Value | | | | | | | |
| No environment properties | | | | | | | | |
| There are no environment properties defined | | | | | | | | |
| <input type="button" value="Cancel"/> <input type="button" value="Previous"/> <input type="button" value="Submit"/> | | | | | | | | |

WebServerPipe

Pipeline type: V2 Execution mode: QUEUED

Source Succeeded
Pipeline execution ID: [Be2460af-df87-463f-925d-7d5d2ee71640](#)

Source
[GitHub/Version_21](#)
Succeeded - 1 minute ago
Blfsdssd

[#EditsData](#) Source: Update README.md

Deploy Succeeded
Pipeline execution ID: [Be2460af-df87-463f-925d-7d5d2ee71640](#)

Deploy
[AWS Elastic Beanstalk](#)
Succeeded - Just now

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Elastic Beanstalk

Applications Environments Change history

Application: WebServer

- Application versions
- Saved configurations

Recent environments

- WebServer-env
- WebApp02-env
- SupraApp-env-1
- MyFirstApp-env

Application WebServer environments (1) [Info](#)

Actions [Create new environment](#)

Filter environments

| Environment name | Health | Date created | Domain | Running vers |
|------------------|--------|------------------------|-------------------------------|---------------|
| WebServer-env | Green | August 17, 2024 22:... | WebServer-env.eba-227p9xyx... | code-pipeline |

https://ap-south-1.console.aws.amazon.com/elasticbeanstalk/home?region=ap-south-1# © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Environment successfully launched.

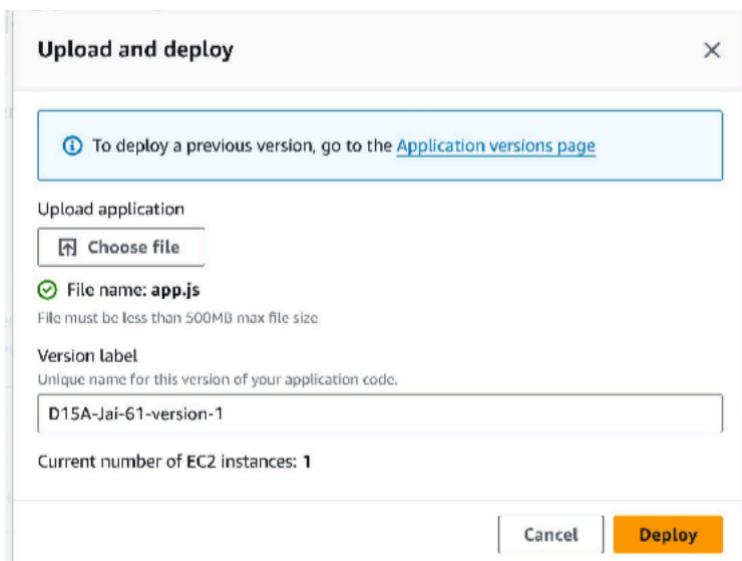
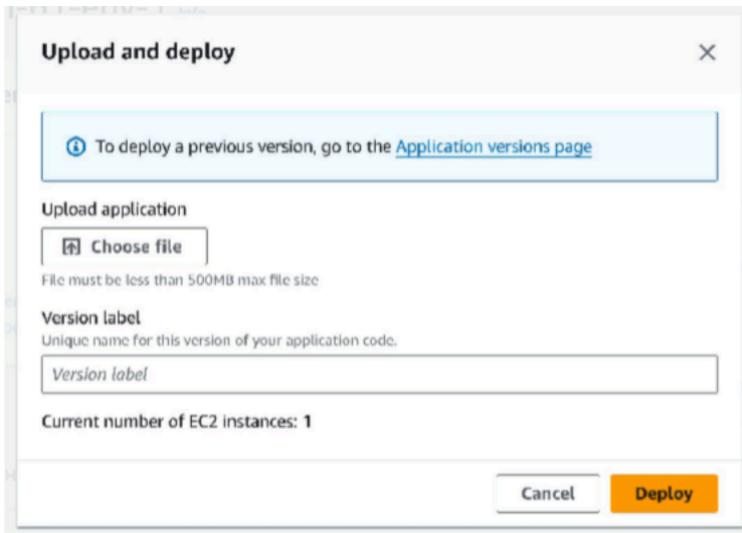
Events Health Logs Monitoring Alarms Managed updates Tags

Events (7) [Info](#)

Filter events by text, property or value

| Time | Type | Details |
|-------------------------------------|------|---|
| August 20, 2024 20:01:18 (UTC+5:30) | INFO | Successfully launched environment: D15A-Jai-61-env-1 |
| August 20, 2024 20:00:14 (UTC+5:30) | INFO | Instance deployment completed successfully. |
| August 20, 2024 19:59:01 (UTC+5:30) | INFO | Waiting for EC2 instances to launch. This may take a few minutes. |
| August 20, 2024 19:57:58 (UTC+5:30) | INFO | Created EIP: 23.21.64.185 |
| August 20, 2024 19:57:43 (UTC+5:30) | INFO | Created security group named: sg-003edb017065a12ed |
| August 20, 2024 19:57:22 (UTC+5:30) | INFO | Using elasticbeanstalk-us-east-1-567270636093 as Amazon S3 storage bucket for environment data. |
| August 20, 2024 19:57:21 (UTC+5:30) | INFO | createEnvironment is starting. |

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



The screenshot shows the deployment confirmation page. A green banner at the top says 'Congratulations' and notes that the first AWS Elastic Beanstalk Node.js application is now running. The dark sidebar on the right lists 'What's Next?' with links to various AWS Elastic Beanstalk documentation pages.

What's Next?

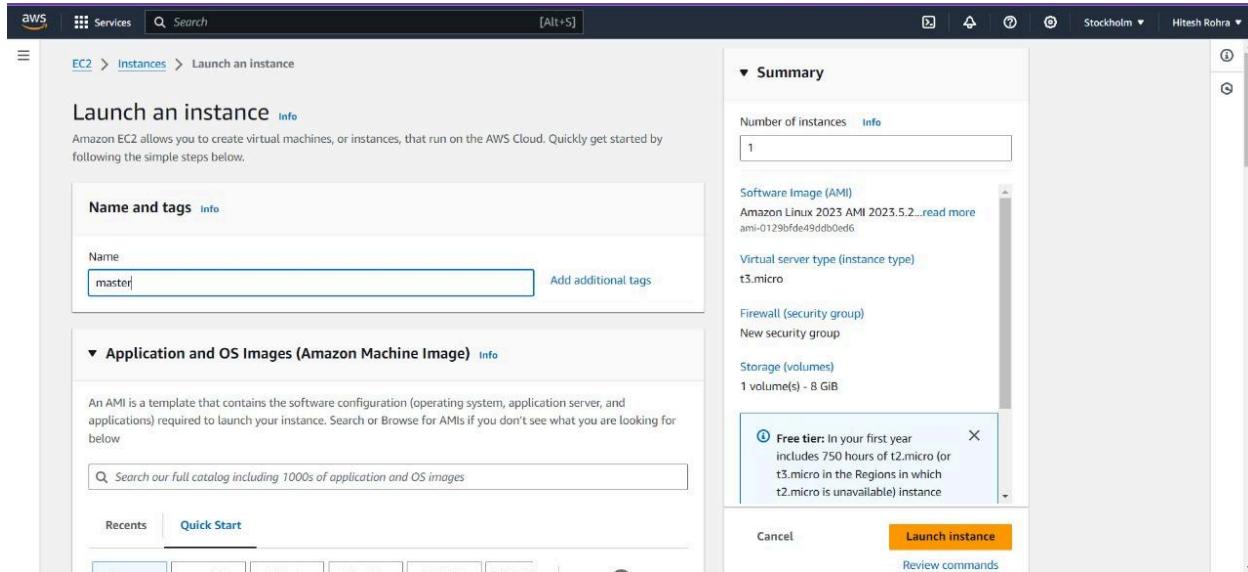
- [AWS Elastic Beanstalk overview](#)
- [AWS Elastic Beanstalk controls](#)
- [Deploying an Express Application to AWS Elastic Beanstalk](#)
- [Deploying an Express application with clustering to Elastic Beanstalk](#)
- [Customizing and Configuring a Node.js Container](#)
- [Working with Logs](#)

Experiment No 3

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

Step 1:Prerequisites

1.1 Create 3 EC2 instances,one for the master node and two for the worker nodes.



```
ubuntu@ip-172-31-31-60:~$ $ sudo apt-get update
$: command not found
ubuntu@ip-172-31-31-60:~$ sudo apt-get update
Hit:1 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:9 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:10 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:11 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:12 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:13 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [530 kB]
Get:14 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [128 kB]
Get:15 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [89564 B]
Get:16 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [374 kB]
Get:17 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [154 kB]
Get:18 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [45.0 kB]
Get:19 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [14.6 kB]
Get:20 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [353 kB]
Get:21 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [68.1 kB]
Get:22 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 c-n-f Metadata [424 B]
```

1.2 Proceed with the following settings and create a new key pair as follows(use the same key pair for all the three nodes)

Salesforce

Search [Alt+S]

Name and tags [Info](#)

Name Add additional tags

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

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

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

Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Linux [Browse more AMIs](#)

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
ami-0522ab6e1ddcc7055 (64-bit (x86)) / ami-0000791bad666add5 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Number of instances [Info](#)
1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd64... [read more](#)
ami-0522ab6e1ddcc7055

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

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

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IP address usage per month

Cancel [Launch instance](#)

AWS CloudFormation Stack: master

Stack ID: arn:aws:cloudformation:us-east-1:123456789012:stack/master/1234567890123456

Created: 2023-09-01T12:00:00Z

Last Updated: 2023-09-01T12:00:00Z

Outputs:

- Output 1: Value: master, Description: The name of the CloudFormation stack.

Resources:

- Resource Type: AWS::CloudFormation::Stack
- Logical ID: master
- Physical ID: arn:aws:cloudformation:us-east-1:123456789012:resource/arn:aws:cloudformation:us-east-1:123456789012:stack/master/1234567890123456
- Creation Time: 2023-09-01T12:00:00Z
- Last Update Time: 2023-09-01T12:00:00Z
- Deletion Time: N/A
- Deletion Policy: Retain
- Update Policy: Retain
- Depends On: N/A
- Properties:
 - Stack Name: master
 - Template Body:

```
version: '2010-09-09'
resources:
  Master:
    Type: AWS::CloudFormation::Interface
    Properties:
      ParameterGroups:
        - Label: 'Master Parameters'
          Parameters:
            - Name: MasterParameter
              Type: String
              Description: 'The name of the CloudFormation stack.'
```

Create key pair

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

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

Key pair type

RSA RSA encrypted private and public key pair

ED25519 ED25519 encrypted private and public key pair

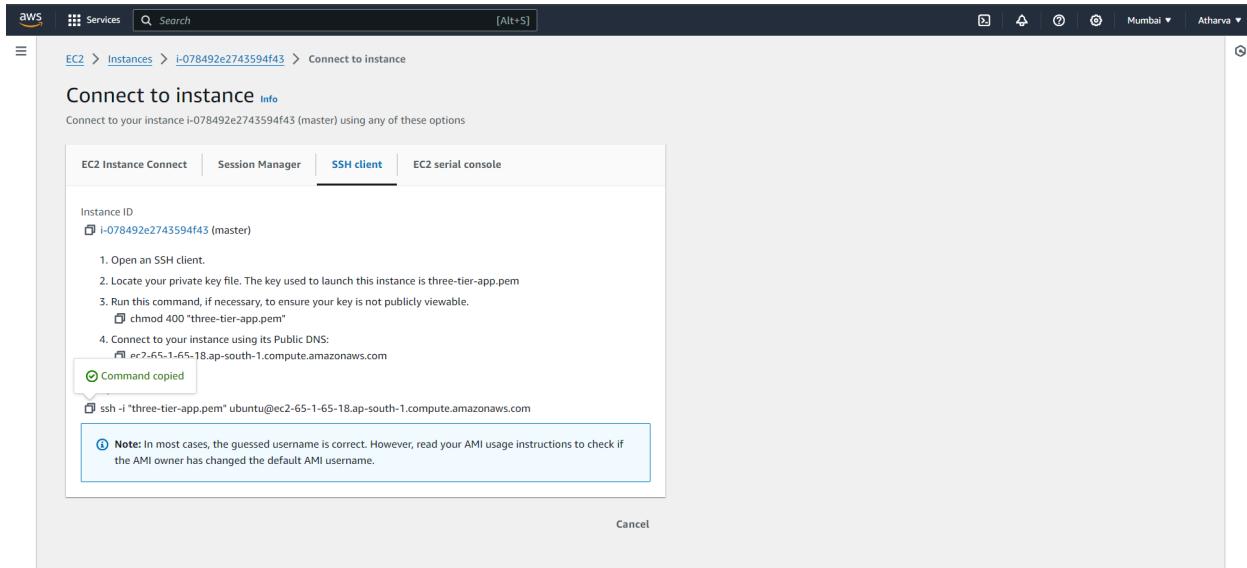
Private key file format

.pem For use with OpenSSH

.ppk For use with PuTTY

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

Cancel [Create key pair](#)



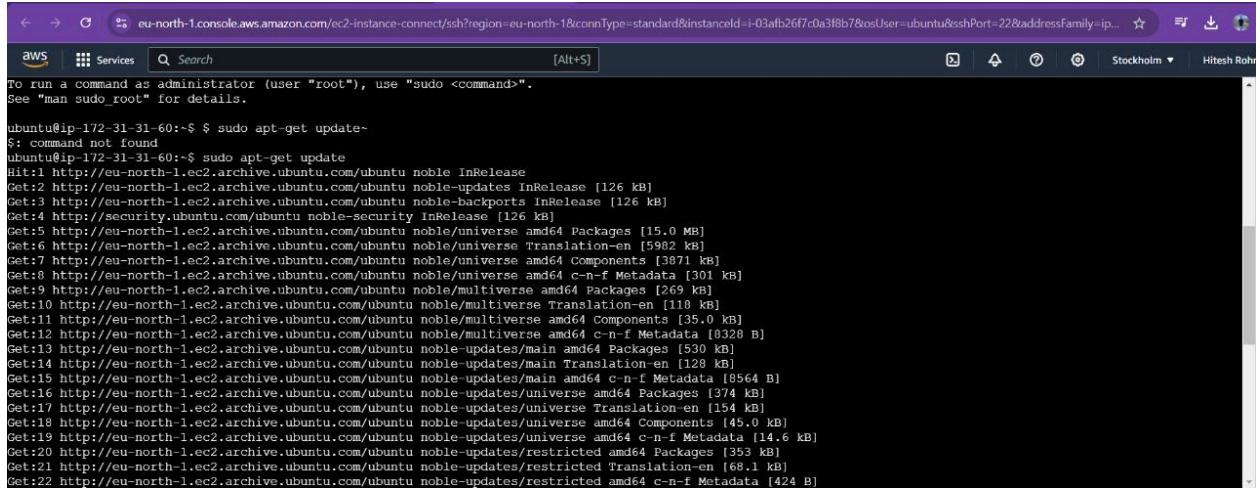
1.3 Add port 6443 in each security group

The screenshot shows the 'Inbound rules' section of the AWS Security Groups page. It lists two rules:

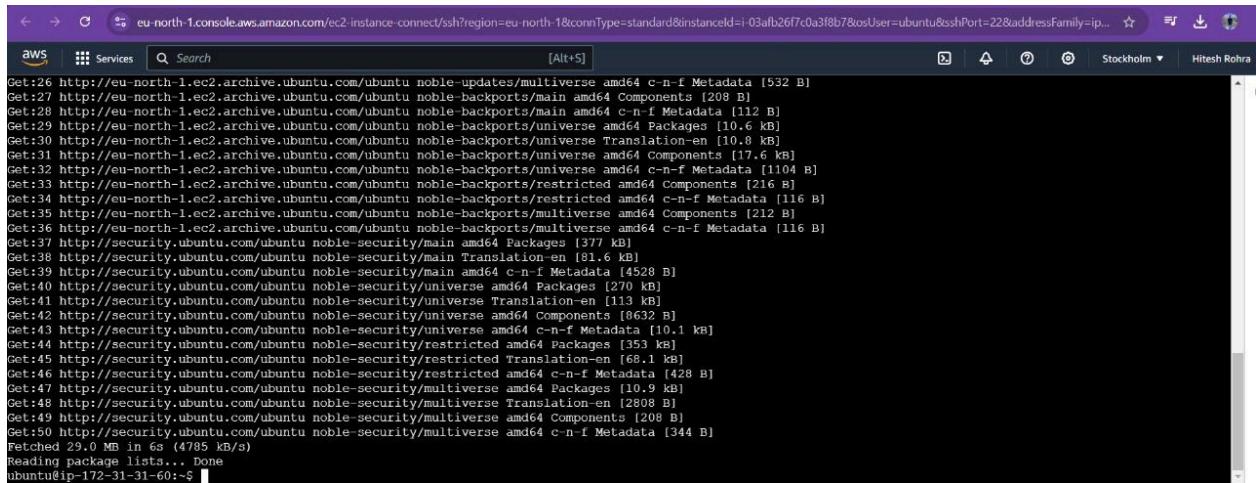
| Security group rule ID | Type | Protocol | Port range | Source | Description - optional |
|------------------------|------------|----------|------------|--------|------------------------|
| sgr-0cf20b6a9f8501fc6 | Custom TCP | TCP | 6443 | Custom | 0.0.0.0/0 |
| sgr-0e02c88e6fce1b710 | SSH | TCP | 22 | Custom | 0.0.0.0/0 |

At the bottom left is a 'Add rule' button.

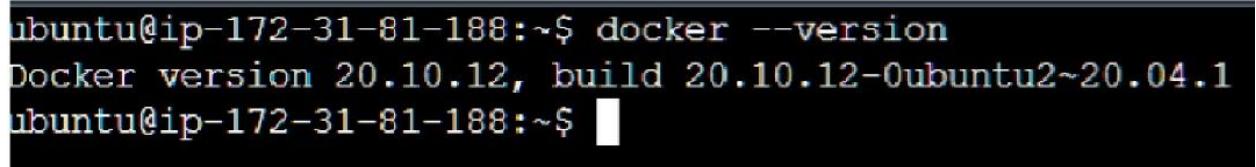
1.4 After the instances have been created, copy the text given in the example part of each of the three instances into git bash.



```
ubuntu@ip-172-31-60:~$ sudo apt-get update
$: command not found
ubuntu@ip-172-31-60:~$ sudo apt-get update
Hit:1 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu/noble/universe amd64 Packages [15.0 MB]
Get:6 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu/noble/universe Translation-en [5982 kB]
Get:7 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu/noble/universe amd64 Components [3871 kB]
Get:8 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu/noble/universe amd64 c-n-f Metadata [301 kB]
Get:9 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu/noble/multiverse amd64 Packages [269 kB]
Get:10 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu/noble/multiverse Translation-en [118 kB]
Get:11 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu/noble/multiverse amd64 Components [35.0 kB]
Get:12 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu/noble/multiverse amd64 c-n-f Metadata [9328 B]
Get:13 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu/noble-updates/main amd64 Packages [530 kB]
Get:14 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu/noble-updates/main Translation-en [128 kB]
Get:15 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu/noble-updates/main amd64 c-n-f Metadata [8564 B]
Get:16 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu/noble-updates/universe amd64 Packages [374 kB]
Get:17 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu/noble-updates/universe Translation-en [154 kB]
Get:18 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu/noble-updates/universe amd64 Components [45.0 kB]
Get:19 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu/noble-updates/universe amd64 c-n-f Metadata [14.6 kB]
Get:20 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu/noble-updates/restricted amd64 Packages [353 kB]
Get:21 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu/noble-updates/restricted Translation-en [68.1 kB]
Get:22 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu/noble-updates/restricted amd64 c-n-f Metadata [424 B]
```



```
Get:26 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [532 B]
Get:27 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]
Get:28 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 B]
Get:29 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [10.6 kB]
Get:30 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [10.8 kB]
Get:31 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [17.6 kB]
Get:32 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1104 B]
Get:33 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:34 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B]
Get:35 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu/noble-backports/multiverse amd64 c-n-f Metadata [212 B]
Get:36 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu/noble-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:37 http://security.ubuntu.com/ubuntu/noble-security/main amd64 Packages [377 kB]
Get:38 http://security.ubuntu.com/ubuntu/noble-security/main Translation-en [81.6 kB]
Get:39 http://security.ubuntu.com/ubuntu/noble-security/main amd64 c-n-f Metadata [4528 B]
Get:40 http://security.ubuntu.com/ubuntu/noble-security/universe amd64 Packages [270 kB]
Get:41 http://security.ubuntu.com/ubuntu/noble-security/universe Translation-en [113 kB]
Get:42 http://security.ubuntu.com/ubuntu/noble-security/universe amd64 Components [8632 B]
Get:43 http://security.ubuntu.com/ubuntu/noble-security/universe amd64 c-n-f Metadata [10.1 kB]
Get:44 http://security.ubuntu.com/ubuntu/noble-security/restricted amd64 Packages [353 kB]
Get:45 http://security.ubuntu.com/ubuntu/noble-security/restricted Translation-en [68.1 kB]
Get:46 http://security.ubuntu.com/ubuntu/noble-security/restricted amd64 c-n-f Metadata [428 B]
Get:47 http://security.ubuntu.com/ubuntu/noble-security/multiverse amd64 Packages [10.9 kB]
Get:48 http://security.ubuntu.com/ubuntu/noble-security/multiverse Translation-en [2808 B]
Get:49 http://security.ubuntu.com/ubuntu/noble-security/multiverse amd64 Components [208 B]
Get:50 http://security.ubuntu.com/ubuntu/noble-security/multiverse amd64 c-n-f Metadata [344 B]
Fetched 29.0 MB in 6s (4785 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-60:~$
```



```
ubuntu@ip-172-31-81-188:~$ docker --version
Docker version 20.10.12, build 20.10.12-0ubuntu2~20.04.1
ubuntu@ip-172-31-81-188:~$
```

Step 2: Run the following commands on both the master and worker nodes to prepare them for kubeadm.

```
# disable swap
sudo swapoff -a

# Create the .conf file to load the modules at bootup
cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
```

```
overlay
br_netfilter
EOF

sudo modprobe overlay
sudo modprobe br_netfilter

# sysctl params required by setup, params persist across reboots
cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-iptables = 1
net.bridge.bridge-nf-call-ip6tables = 1
net.ipv4.ip_forward = 1
EOF

# Apply sysctl params without reboot
sudo sysctl --system

## Install CRI-O Runtime
sudo apt-get update -y
sudo apt-get install -y software-properties-common curl apt-transport-https
ca-certificates gpg

sudo curl -fsSL https://pkgs.k8s.io/addons:/cri-o:/prerelease:/main/deb/Release.key |
sudo gpg --dearmor -o /etc/apt/keyrings/cri-o-apt-keyring.gpg
echo "deb [signed-by=/etc/apt/keyrings/cri-o-apt-keyring.gpg]
https://pkgs.k8s.io/addons:/cri-o:/prerelease:/main/deb/ /" | sudo tee
/etc/apt/sources.list.d/cri-o.list

sudo apt-get update -y
sudo apt-get install -y cri-o

sudo systemctl daemon-reload
sudo systemctl enable crio --now
sudo systemctl start crio.service

echo "CRI runtime installed successfully"

# Add Kubernetes APT repository and install required packages
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.29/deb/Release.key | sudo gpg --dearmor
-o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
```

```
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
https://pkgs.k8s.io/core:/stable:/v1.29/deb/ /' | sudo tee
/etc/apt/sources.list.d/kubernetes.list
```

```
sudo apt-get update -y
sudo apt-get install -y kubelet="1.29.0-*" kubectl="1.29.0-*" kubeadm="1.29.0-*"
sudo apt-get update -y
sudo apt-get install -y jq
```

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

```
ubuntu@ip-172-31-46-220:~$ # disable swap
sudo swapoff -a

# Create the .conf file to load the modules at bootup
cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
overlay
br_netfilter
EOF

sudo modprobe overlay
sudo modprobe br_netfilter

# sysctl params required by setup, params persist across reboots
cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-iptables = 1
net.bridge.bridge-nf-call-ip6tables = 1
net.ipv4.ip_forward = 1
EOF

# Apply sysctl params without reboot
sudo sysctl --system

## Install CRIS Runtime
sudo apt-get update -y
sudo apt-get install -y software-properties-common curl apt-transport-https ca-certificates gpg

sudo curl -fsSL https://pkgs.k8s.io/addons/:cri-o:/prerelease:/main/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/cri-o-apt-keyring.gpg
echo "deb [signed-by=/etc/apt/keyrings/cri-o-apt-keyring.gpg] https://pkgs.k8s.io/addons/:cri-o:/prerelease:/main/deb/ /" | sudo systemctl start kubeletkubelet29.0-*" kubectl="1.29.0-*" kubeadm="1.29.0-*"k8s.io/co
re/stable:/v1.29/deb/ /" | sudo
overlay
br_netfilter
net.bridge.bridge-nf-call-iptables = 1
net.bridge.bridge-nf-call-ip6tables = 1
net.ipv4.ip_forward = 1
* Applying /etc/lib/systemd/d/10-appArmor.conf ...
* Applying /etc/sysctl.d/40-console-messages.conf ...
* Applying /etc/sysctl.d/40-ipv6-privacy.conf ...
* Applying /etc/sysctl.d/40-kernel-hardening.conf ...
* Applying /etc/sysctl.d/40-magic-sysrq.conf ...
* Applying /etc/sysctl.d/40-map-count.conf ...
* Applying /etc/sysctl.d/40-network-security.conf ...
* Applying /etc/sysctl.d/40-pid-max.conf ...
* Applying /etc/sysctl.d/40-ppc64.conf ...
* Applying /etc/sysctl.d/50-cloudimg-settings.conf ...
* Applying /usr/lib/sysctl.d/50-pid-max.conf ...
* Applying /etc/lib/sysctl.d/99-cloudimg-ipv6.conf ...
* Applying /usr/lib/sysctl.d/99-protect-links.conf ...
* Applying /etc/sysctl.d/99-sysctl.conf ...
```

Step3: Run the above command only on master node

```
sudo kubeadm config images pull
```

```
sudo kubeadm init
```

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

```
# Network Plugin = calico
```

```
kubectl apply -f
```

```
https://raw.githubusercontent.com/projectcalico/calico/v3.26.0/manifests/calico.yaml
```

```
kubeadm token create --print-join-command
```

```
ubuntu@ip-172-31-46-220:~$ sudo kubeadm config images pull
sudo kubeadm init
mkdir -p "$HOME"/.kube
sudo cp -i /etc/kubernetes/admin.conf "$HOME"/.kube/config
sudo chown "$(id -u)":"$(id -g)" "$HOME"/.kube/config

# Network Plugin = calico
kubectl apply -f https://raw.githubusercontent.com/projectcalico/calico/v3.26.0/manifests/calico.yaml

kubeadm token create --print-join-command
I0921 11:12:21.776389 3963 remote version is much newer: v1.31.0; falling back to: stable-1.29
[config/images] Pulled image: k8s.io/kube-controller-manager:v1.29.9
[config/images] Pulled registry: k8s.io/kube-scheduler:v1.29.9
[config/images] Pulled registry: k8s.io/kube-proxy:v1.29.9
[config/images] Pulled registry: k8s.io/coredns/coredns:v1.11.1
[config/images] Pulled registry: k8s.io/pause:3.9
[config/images] Pulled registry: k8s.io/etc2:3.5.18-8
I0921 11:12:40.995686 4384 remote version is much newer: v1.31.0; falling back to: stable-1.29
[init] Kubernetes v1.29.9
[preflight] Running pre-flight checks
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
W0921 11:12:41.763011 4384 checks.go:835 detected that the sandbox image "registry.k8s.io/pause:3.10" of the container runtime is inconsistent with that used by kubeadm. It is recommended that using "regis
try.k8s.io/pause:3.9" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "etcd/ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver service cert is signed for DNS names [ip-172-31-46-220 kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 172.31.46.220]
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key
[certs] Generating "etcd/ca" certificate and key
[certs] Generating "etcd/server" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-46-220 localhost] and IPs [172.31.46.220 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-46-220 localhost] and IPs [172.31.46.220 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
[certs] Generating "sa" key and public key
[kubeconfig] Using kubeconfig folder "/etc/kubernetes"
[kubeconfig] Writing "admin.conf" kubeconfig file
[kubeconfig] Writing "super-admin.conf" kubeconfig file
[kubeconfig] Writing "controller-manager.conf" kubeconfig file
[kubeconfig] Writing "scheduler.conf" kubeconfig file
[etcd] Creating static Pod manifest for local etcd in "/etc/kubernetes/manifests"
```

You will get kubeadm token, Copy it.

Step 4: Run the above command only on worker nodes

```
sudo kubeadm reset pre-flight checks
```

```
sudo your-token --v=5
```

```
ubuntu@ip-172-31-36-212:~$ sudo kubeadm reset pre-flight checks
W0921 11:14:17.713669 3933 preflight.go:56] [reset] WARNING: Changes made to this host by 'kubeadm init' or 'kubeadm join' will be reverted.
[reset] Are you sure you want to proceed? [y/N]: yes
[preflight] Running pre-flight checks
W0921 11:14:28.535200 3933 removeetcdmember.go:106] [reset] No kubeadm config, using etcd pod spec to get data directory
[reset] Deleted contents of the etcd data directory: /var/lib/etcd
[reset] Stopping the kubelet service
[reset] Deleting mounted directories in "/var/lib/kubelet"
[reset] Deleting contents of directories: /etc/kubernetes/manifests /var/lib/kubelet /etc/kubernetes/pki
[reset] Deleting files: /etc/kubernetes/admin.conf /etc/kubernetes/super-admin.conf /etc/kubernetes/kubelet.conf /etc/kubernetes/bootstrap-kubelet.conf /etc/kubernetes/controller-manager.conf /etc/kubernetes/scheduler.conf
The reset process does not clean CNI configuration. To do so, you must remove /etc/cni/net.d
The reset process does not reset or clean up iptables rules or IPVS tables.
If you wish to reset iptables, you must do so manually by using the "iptables" command.
If your cluster was setup to utilize IPVS, run ipvsadm --clear (or similar)
to reset your system's IPVS tables.
The reset process does not clean your kubeconfig files and you must remove them manually.
```

```

ubuntu@ip-172-31-36-212:~$ sudo kubeadm join 172.31.46.220:6443 --token k4psyh.ns1g1yett9he59kd4 --discovery-token-ca-cert-hash sha256:80e7e9abf8f31f0333a9d2f7a680d6bd961267ae45cf71bada8de069d4a292e --v=5
I0921 11:28:31.063878 4097 join.go:413] [preflight] found NodeName empty; using OS hostname as NodeName
I0921 11:28:31.063885 4097 initConfiguration.go:122] detected and using CRI socket: unix:///var/run/crio/crio.sock
[preflight] Running pre-flight checks
I0921 11:28:31.064142 4097 preFlight.go:93] [preflight] Running general checks
I0921 11:28:31.064183 4097 checks.go:280] validating the existence of file /etc/kubernetes/kubelet.conf
I0921 11:28:31.064287 4097 checks.go:280] validating the existence of file /etc/kubernetes/bootstrap-kubelet.conf
I0921 11:28:31.064219 4097 checks.go:184] validating the container runtime
I0921 11:28:31.089669 4097 checks.go:639] validating whether swap is enabled or not
I0921 11:28:31.089763 4097 checks.go:280] validating the presence of executable crictl
I0921 11:28:31.089799 4097 checks.go:378] validating the presence of executable cointtrack
I0921 11:28:31.089810 4097 checks.go:378] validating the presence of executable cni
I0921 11:28:31.089818 4097 checks.go:370] validating the presence of executable iptables
I0921 11:28:31.089870 4097 checks.go:370] validating the presence of executable mount
I0921 11:28:31.089897 4097 checks.go:370] validating the presence of executable nsenter
I0921 11:28:31.089919 4097 checks.go:370] validating the presence of executable ebttables
I0921 11:28:31.089954 4097 checks.go:370] validating the presence of executable ethtool
I0921 11:28:31.089977 4097 checks.go:370] validating the presence of executable socat
I0921 11:28:31.089996 4097 checks.go:370] validating the presence of executable tc
I0921 11:28:31.089911 4097 checks.go:370] validating the presence of executable touch
I0921 11:28:31.089953 4097 checks.go:370] validating the presence of executable curl
I0921 11:28:31.183935 4097 checks.go:401] checking whether the given node name is valid and reachable using net.LookupHost
I0921 11:28:31.185638 4097 checks.go:685] validating Kubelet version
I0921 11:28:31.162593 4097 checks.go:130] validating if the "kubelet" service is enabled and active
I0921 11:28:31.176512 4097 checks.go:283] validating availability of port 10259
I0921 11:28:31.176737 4097 checks.go:280] validating the existence of file /etc/kubernetes/pki/ca.crt
I0921 11:28:31.176765 4097 checks.go:430] validating if the connectivity type is via proxy or direct
I0921 11:28:31.176893 4097 checks.go:329] validating the contents of file /proc/sys/net/bridge/bridge-nf-call-iptables
I0921 11:28:31.176909 4097 checks.go:329] validating the contents of file /proc/sys/net/ipv4/ip_forward
I0921 11:28:31.176983 4097 checks.go:521] [preflight] Discovering cluster info
I0921 11:28:31.176987 4097 token.go:481] [discovery] Created cluster-info discovery client, requesting info from "172.31.46.220:6443"
I0921 11:28:31.187676 4097 token.go:118] [discovery] Requesting info from "172.31.46.220:6443" again to validate TLS against the pinned public key
I0921 11:28:31.194531 4097 token.go:135] [discovery] Cluster info signature and contents are valid and TLS certificate validates against pinned roots, will use API Server "172.31.46.220:6443"
I0921 11:28:31.194608 4097 discovery.go:52] [discovery] Using provided TLSBootstrapToken as authentication credentials for the join process
I0921 11:28:31.194622 4097 join.go:546] [preflight] Fetching init configuration
I0921 11:28:31.194629 4097 join.go:592] [preflight] Retrieving KubeConfig objects
[preflight] Retrieving configuration for the cluster
[preflight] If you look at this config file with 'kubectl -n kube-system get cm kubeade-config -o yaml'
I0921 11:28:31.281989 4097 kubeProxy.go:58] attempting to download the KubeProxyConfiguration from ConfigMap "kube-proxy"
I0921 11:28:31.285146 4097 kubelet.go:74] attempting to download the KubeletConfiguration from ConfigMap "kubelet-config"
I0921 11:28:31.288379 4097 initConfiguration.go:114] skip CRI socket detection, fill with the default CRI socket unix:///var/run/containerd/containerd.sock
I0921 11:28:31.288595 4097 interface.go:432] Looking for default routes with IPv4 addresses
I0921 11:28:31.288617 4097 interface.go:437] Default route transits interface "enX0"
I0921 11:28:31.288751 4097 interface.go:289] Interface enX0 is up
I0921 11:28:31.288839 4097 interface.go:259] Interface enX0 has 2 addresses : [172.31.36.212/20 fe80::75:41ff:fea5:aefb1/64].
I0921 11:28:31.288840 4097 interface.go:240] Checking addr: 172.31.36.212/20
I0921 11:28:31.288829 4097 interface.go:231] IP found 172.31.36.212
I0921 11:28:31.288848 4097 interface.go:263] Found valid IPv4 address 172.31.36.212 for interface "enX0".
I0921 11:28:31.288849 4097 interface.go:443] Found active IP 172.31.36.212
I0921 11:28:31.215082 4097 preflight.go:184] [preflight] Running configuration dependant checks
I0921 11:28:31.215028 4097 controlplaneprepare.go:225] [download-certs] Skipping certs download

```

Step5: Run the given command to verify cluster creation

kubectl get nodes

| ubuntu@ip-172-31-46-220:~\$ kubectl get nodes | | | | |
|---|--------|---------------|-----|---------|
| NAME | STATUS | ROLES | AGE | VERSION |
| ip-172-31-36-212 | Ready | <none> | 47s | v1.29.0 |
| ip-172-31-46-220 | Ready | control-plane | 16m | v1.29.0 |
| ip-172-31-47-26 | Ready | <none> | 29s | v1.29.0 |

Experiment No 4

Step 1: Deploying Your Application on Kubernetes**1.1 Set up Kubernetes Cluster**

1. If you haven't already set up a Kubernetes cluster (e.g., with kubeadm), use minikube or any managed Kubernetes service (like EKS, GKE, etc.) to get a cluster running.

2. Once your cluster is ready, verify the nodes:

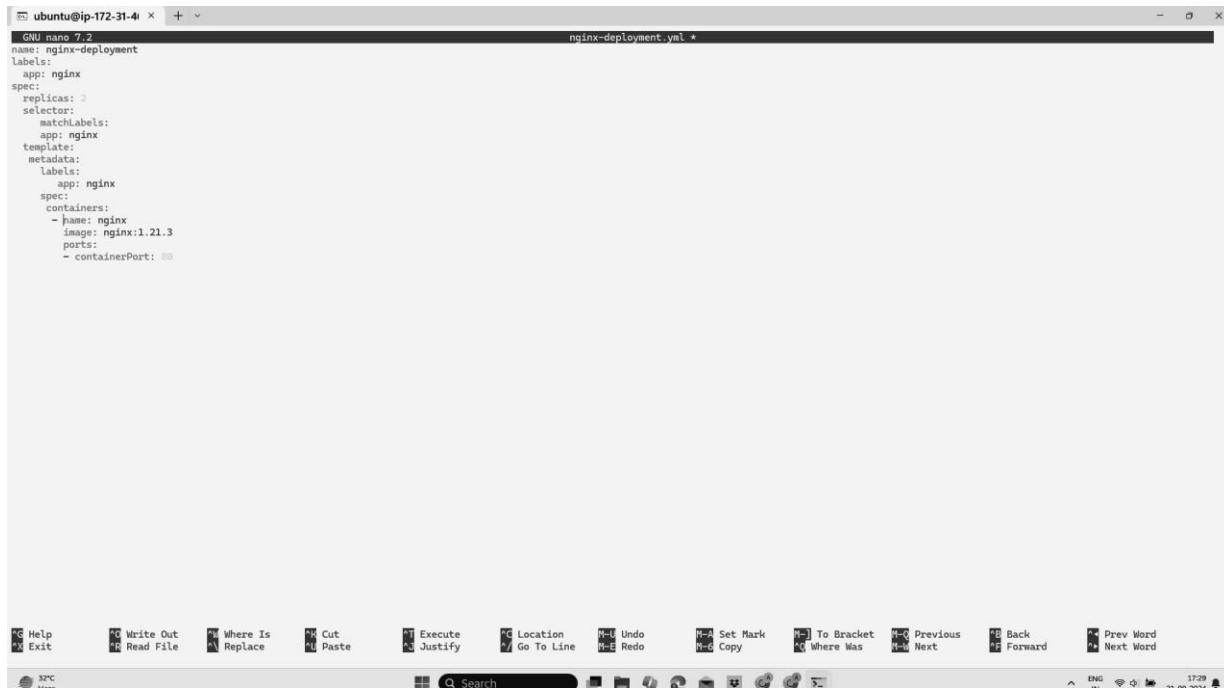
```
kubectl get nodes
```

```
ubuntu@ip-172-31-46-220:~$ kubectl get nodes
NAME           STATUS    ROLES      AGE     VERSION
ip-172-31-36-212   Ready    <none>    47s    v1.29.0
ip-172-31-46-220   Ready    control-plane   16m    v1.29.0
ip-172-31-47-26   Ready    <none>    29s    v1.29.0
```

Step 2: Create the Deployment YAML file

a) Create the YAML file: Use a text editor to create a file named nginx-deployment.yaml

Add the Deployment Configuration: Copy and paste the following YAML content into the file. Save and exit the editor (Press Ctrl+X, then Y, and Enter).



```
ubuntu@ip-172-31-4i ~ % kubectl get nodes
NAME           STATUS    ROLES      AGE     VERSION
ip-172-31-36-212   Ready    <none>    47s    v1.29.0
ip-172-31-46-220   Ready    control-plane   16m    v1.29.0
ip-172-31-47-26   Ready    <none>    29s    v1.29.0
```

```
GNU nano 7.2                                         nginx-deployment.yaml *
name: nginx-deployment
labels:
  app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
      spec:
        containers:
        - name: nginx
          image: nginx:1.21.3
          ports:
          - containerPort: 80
```

Step 3:Create the Service YAML File

a) Create the YAML File: Create another file named nginx-service.yaml Add the Service Configuration: Copy and paste the following YAML content into the file given below



```
GNU nano 7.2
apiVersion: v1
kind: Service
metadata:
  name: nginx-server
spec:
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 80
      targetport: 80
  type: loadbalancer
```

Step 4:Apply the YAML Files a) Deploy the Application: Use kubectl to create the Deployment and Service from the YAML files. Verify the Deployment: Check the status of your Deployment, Pods and Services. Describe the deployment(Extra)

```
ubuntu@ip-172-31-46-220:~$ kubectl apply -f nginx-deployment.yaml
deployment.apps/nginx-deployment created
```

```
ubuntu@ip-172-31-46-220:~$ kubectl apply -f nginx-service.yaml
service/nginx-server created
```

Step 5:Ensure Service is Running 6.1 Verify Service: Run the following command to check the services running in your cluster: Kubectl get deployment Kubectl get pods kubectl get service

```
error: the server doesn't have a resource type "deployments"
ubuntu@ip-172-31-46-220:~$ kubectl get deployments
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment  3/3     3           3           7m27s
```

```
ubuntu@ip-172-31-46-220:~$ kubectl get services
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
kubernetes  ClusterIP  10.96.0.1      <none>        443/TCP      85m
nginx-server  LoadBalancer  10.111.218.213  <pending>      80:30798/TCP  110s
```

Step 6: Forward the Service Port to Your Local Machine kubectl port-forward allows you to forward a port from your local machine to a port on a service running in the Kubernetes cluster.

1. Forward the Service Port: Use the following command to forward a local port to the service's target port. kubectl port-forward service/ :

This command will forward local port 8080 on your machine to port 80 of the service nginx-service running inside the cluster.

```
ubuntu@ip-172-31-46-220:~$ kubectl describe deployments
Name:           nginx-deployment
Namespace:      default
CreationTimestamp: Sat, 21 Sep 2024 12:30:54 +0000
Labels:          app=nginx
Annotations:    deployment.kubernetes.io/revision: 1
Selector:        app=nginx
Replicas:       3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType:   RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=nginx
  Containers:
    nginx:
      Image:      nginx:1.16
      Port:       80/TCP
      Host Port:  0/TCP
      Environment: <none>
      Mounts:     <none>
      Volumes:    <none>
  Conditions:
    Type     Status  Reason
    ----  -----
    Available  True    MinimumReplicasAvailable
    Progressing  True    NewReplicaSetAvailable
  OldReplicaSets: <none>
  NewReplicaSet:  nginx-deployment-854bc88786 (3/3 replicas created)
Events:
  Type     Reason          Age   From            Message
  ----  -----  --  --  -----
  Normal  ScalingReplicaSet 11m  deployment-controller  Scaled up replica set nginx-deployment-854bc88786 to 3
```

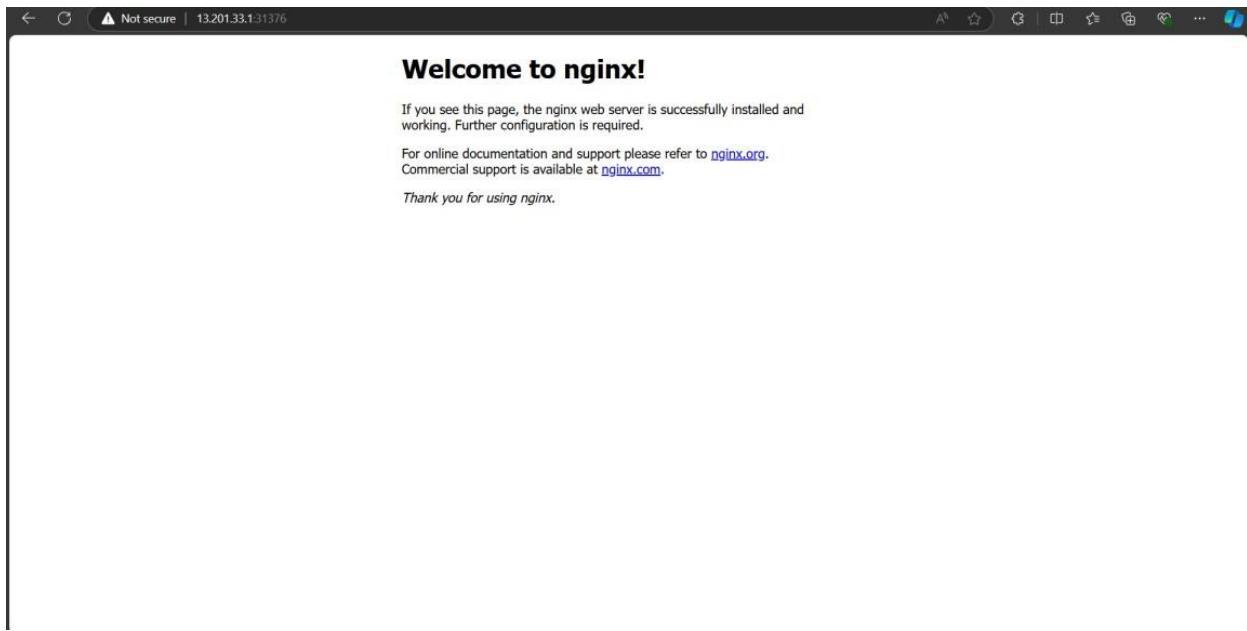
2. This means port forwarding is now active, and any traffic to localhost:8080 will be routed to the nginx-service on port 80.

```
ubuntu@ip-172-31-46-220:~$ kubectl port-forward service/nginx-server 8080:80
```

Step 7:

Access the Application Locally

1. Open a Web Browser: Now open your web browser and go to the following URL:
<http://localhost:8080> You should see the application (in this case, Nginx) that you have deployed running in the Kubernetes cluster, served locally via port 8080. In case the port 8080 is unavailable, try using a different port like 8081



Experiment No 5

Advance devops Exp:5

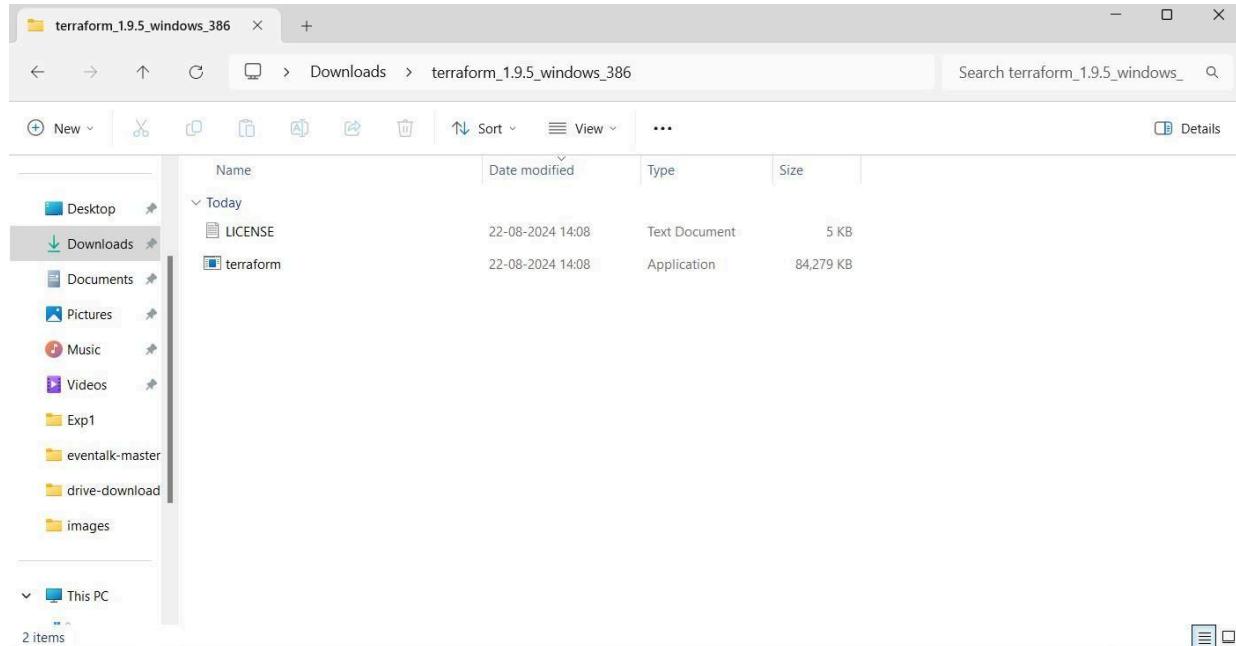
Hitesh Rohra

D15 A-47

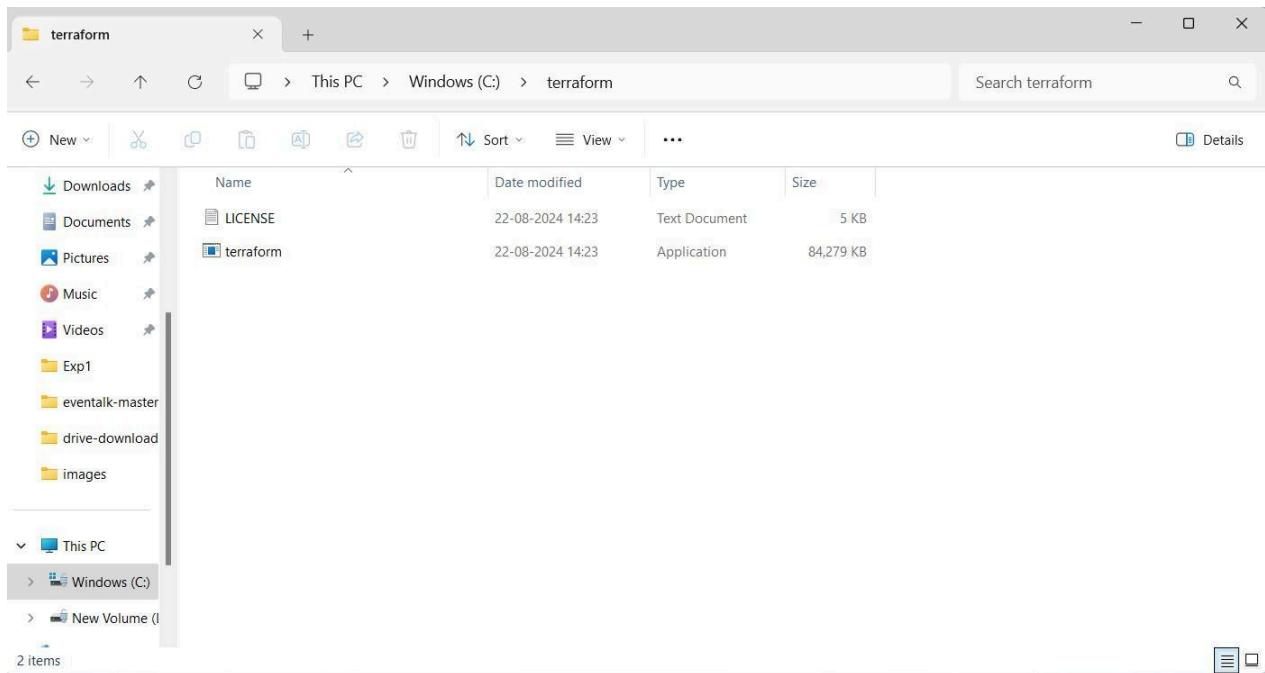
Aim: To understand terraform lifecycle, core concepts/terminologies and install it on a linux machine and windows

Step1: Download Terraform from the official website

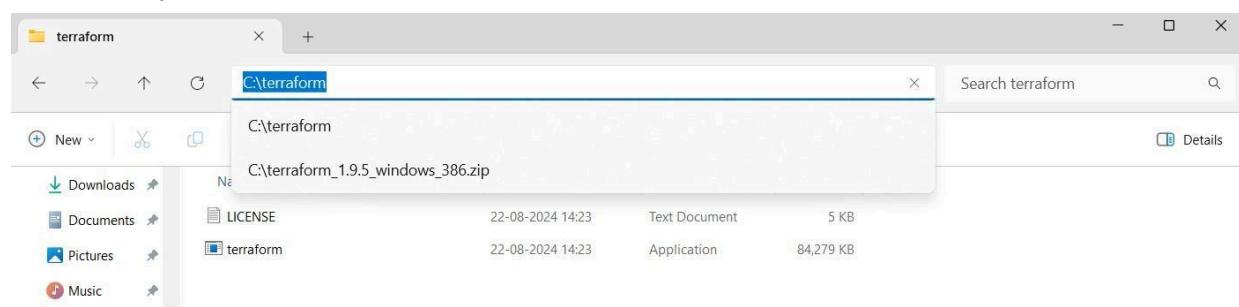
The screenshot shows the Terraform official website at developer.hashicorp.com/terraform/install#windows. The left sidebar has a 'Install Terraform' section selected. Under 'macOS', there's a 'Package manager' section with a terminal command: `brew tap hashicorp/tap` and `brew install hashicorp/tap/terraform`. Below it is a 'Binary download' section with links for 'AMD64' and 'ARM64' versions. The right sidebar contains sections for 'About Terraform', 'Featured docs' (Introduction to Terraform, Configuration Language, Terraform CLI, HCP Terraform, Provider Use), and an 'HCP Terraform' section.



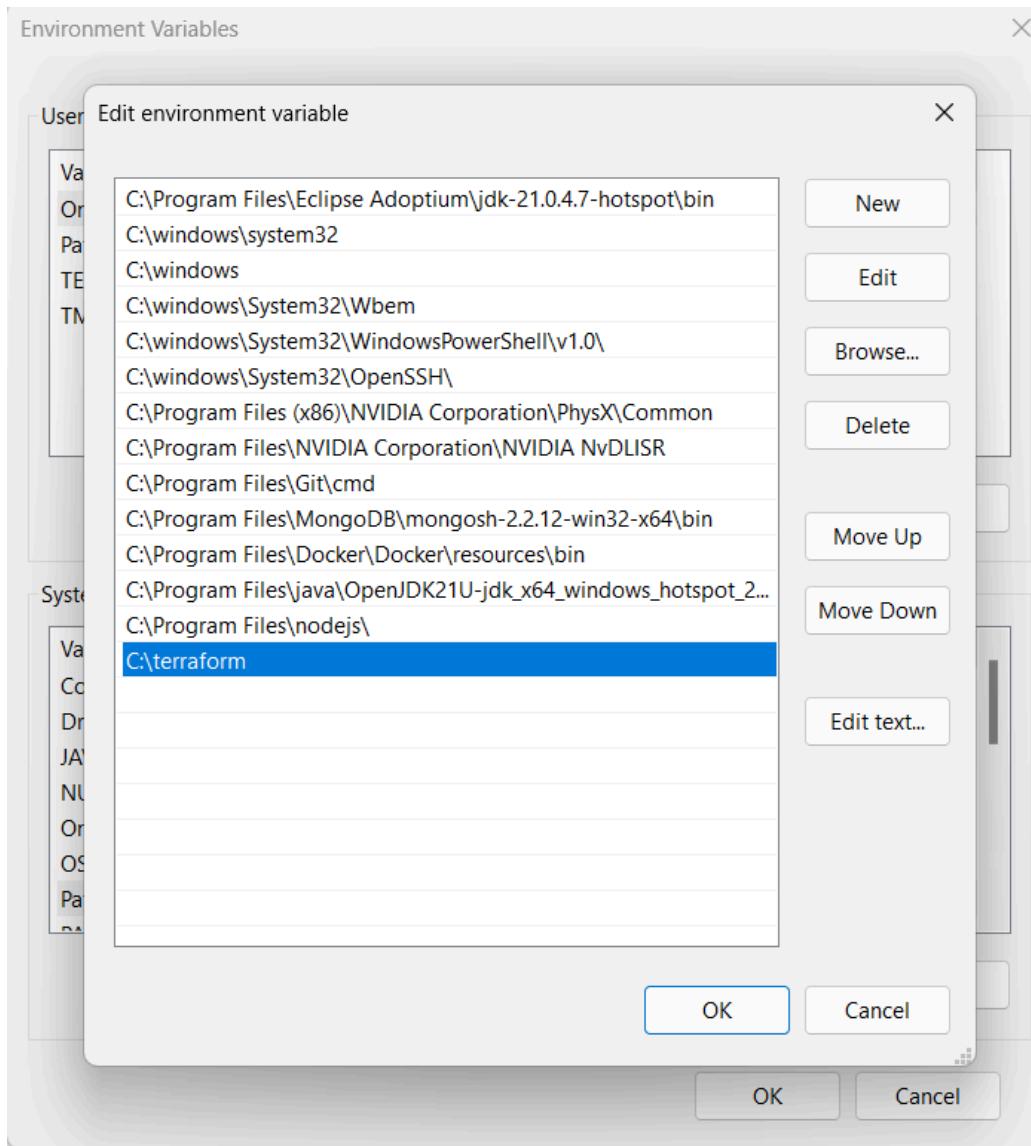
Step 2: Copy and extract Terraform from the downloads and paste it in the C drive



Step 3: Copy the file path to paste in the environment variables



Step 4: Set the environment variables for terraform



Step 5: Check whether the terraform is installed

```
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\navan>terraform --version
Terraform v1.9.5
on windows_386
```

Experiment No 6

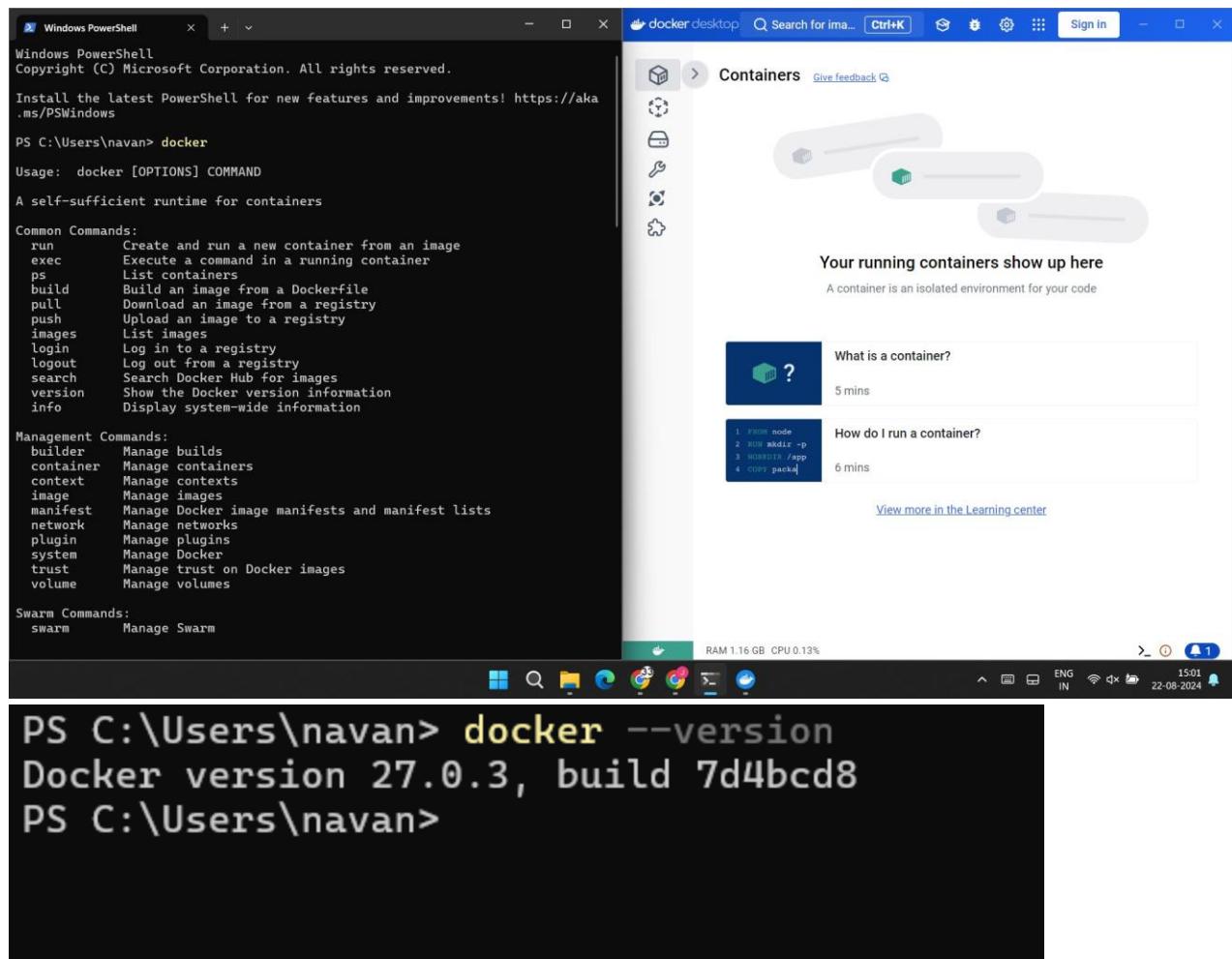
Advance devops Exp:5

Hitesh Rohra

D15 A-47

Aim: Creating docker image using Terraform

Step 1: Install docker Desktop after installation check the functionality

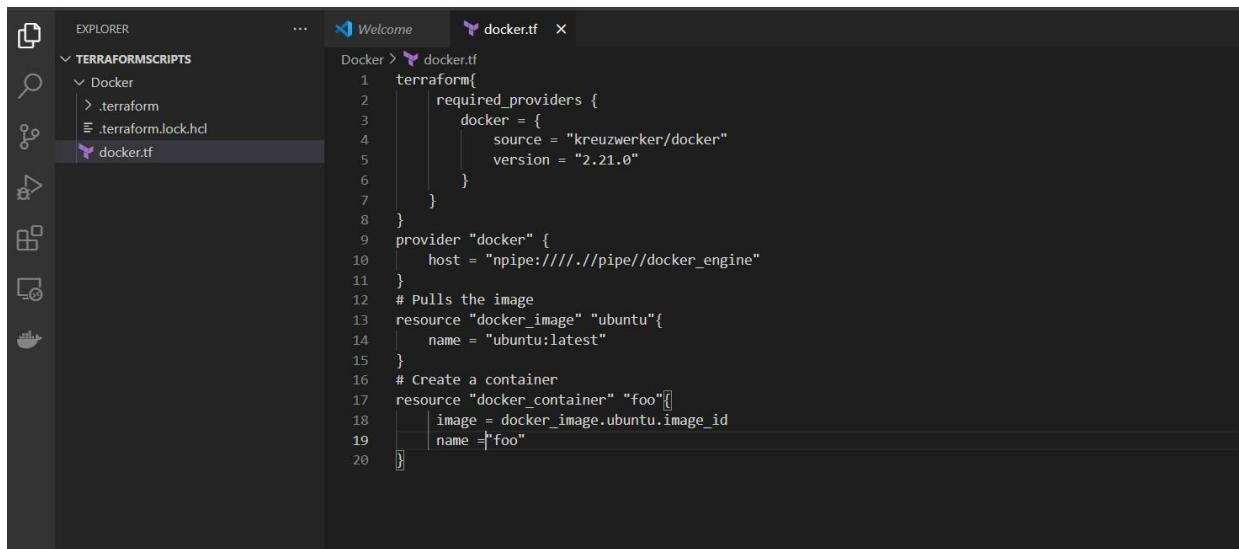


Now, create a folder named 'Terraform Scripts' in which we save our different types of scripts which will be further used in this experiment.

Step 2: Firstly create a new folder named 'Docker' in the 'TerraformScripts' folder. Then create a new docker.tf file using Atom editor and write the following contents into it to create a Ubuntu Linux container.

Script:

```
terraform{
    required_providers {
        docker = {
            source = "kreuzwerker/docker"
            version = "2.21.0"
        }
    }
}
provider "docker" {
    host = "npipe:///./pipe//docker_engine"
}
# 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"
}
```



Step 3: Execute terraform init command to initialize the resources

```
C:\Users\navan\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.

C:\Users\navan\Desktop\TerraformScripts\Docker>
```

Step 4: Execute Terraform plan to see the available resources

```
C:\Users\navan\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         = (known after apply)
    + container_logs = (known after apply)
    + entrypoint      = (known after apply)
    + env             = (known after apply)
    + exit_code       = (known after apply)
    + gateway         = (known after apply)
    + hostname        = (known after apply)
    + id              = (known after apply)
    + image           = (known after apply)
    + init            = (known after apply)
    + ip_address      = (known after apply)
    + ip_prefix_length = (known after apply)
    + ipc_mode        = (known after apply)
    + log_driver      = (known after apply)
    + logs            = false
    + must_run        = true
    + name            = "foo"
    + network_data   = (known after apply)
    + read_only       = false
    + remove_volumes = true
    + restart         = "no"
    + rm              = false
    + runtime         = (known after apply)
```

```

+ security_opts    = (known after apply)
+ shm_size         = (known after apply)
+ start            = true
+ 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.

C:\Users\navan\Desktop\TerraformScripts\Docker>

```

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

```

}
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: Creation complete after 10s [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Creating...

```

Docker images,before Executing Apply step:

```

C:\Users\navan\Desktop\TerraformScripts\Docker>docker images
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
ubuntu          latest        edbfe74c41f8  2 weeks ago   78.1MB
node            20-alpine    e2997a3fdff8  4 weeks ago   133MB

```

```
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28e  
3e6df8c9d66519b6ad761c2598aubuntu:latest]
```

Note: Objects have changed outside of Terraform

Terraform detected the following changes made outside of Terraform since the last "terraform apply" which may have affected this plan:

```
# docker_image.ubuntu has been deleted
- resource "docker_image" "ubuntu" {
    id          = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6  
2598aubuntu:latest"
    - image_id   = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6  
2598a" -> null
    name        = "ubuntu:latest"
    # (2 unchanged attributes hidden)
}
```

Unless you have made equivalent changes to your configuration, or ignored the relevant attributes using ignore_changes, the following plan may include actions to undo or respond to these changes.

Step 6: Execute Terraform destroy to delete the configuration, which will automatically delete the ubuntu container.

```
C:\Users\navan\Desktop\TerraformScripts\Docker>terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]

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_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, 1 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:
```

```
C:\Windows\System32\cmd.exe > terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
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_image.ubuntu will be destroyed
- resource "docker_image" "ubuntu" {
  - id          = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> 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, 1 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_image.ubuntu: Destroying... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_image.ubuntu: Destruction complete after 0s

Destroy complete! Resources: 1 destroyed.
```

Docker images after executing destroy step

```
C:\Users\navan\Desktop\TerraformScripts\Docker>docker images
REPOSITORY      TAG          IMAGE ID      CREATED       SIZE
node            20-alpine    e2997a3fdff8   5 weeks ago   133MB
```

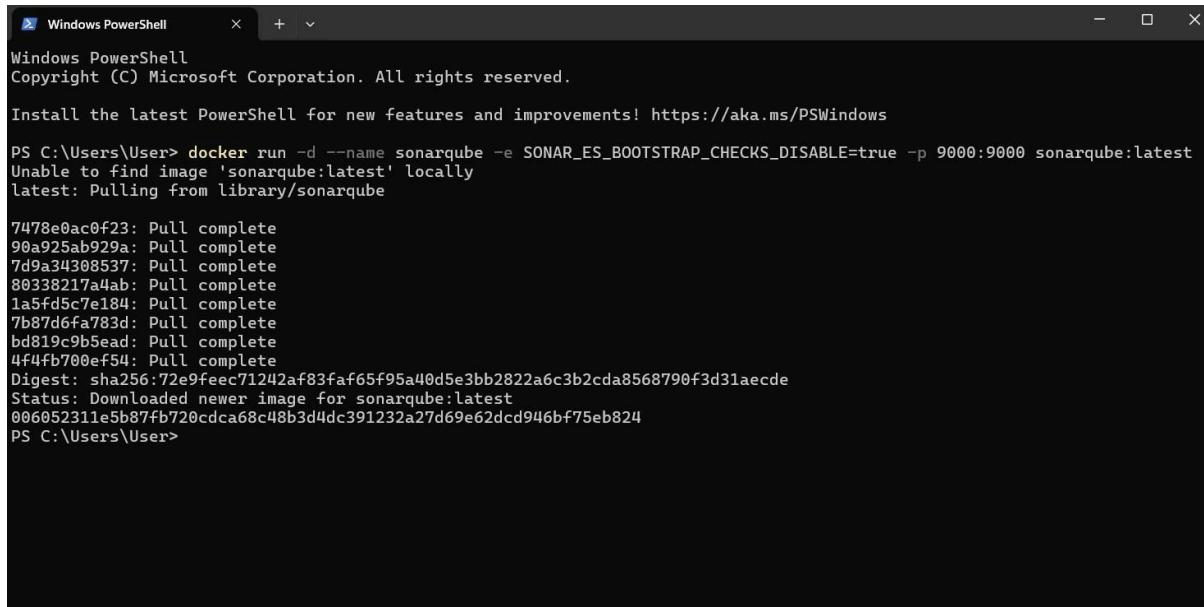
Experiment No 7

EXPERIMENT 7

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

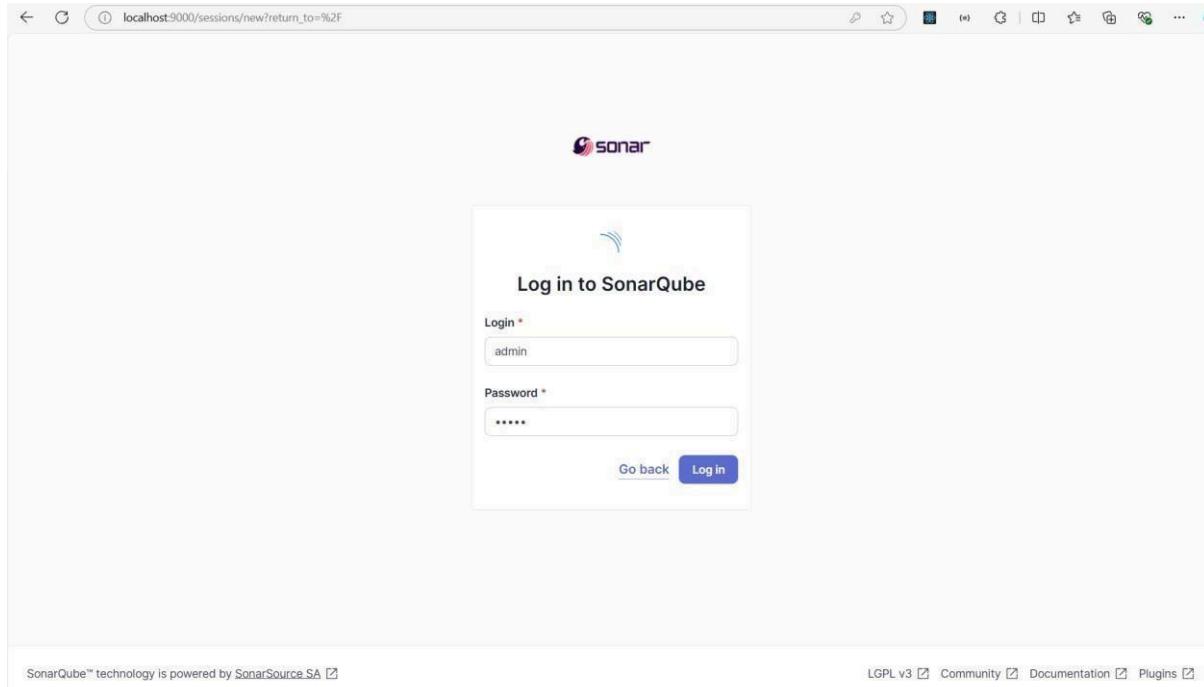
Step 1: Open Windows PowerShell and run the following command –

```
docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000  
sonarqube:latest WARNING: Run the following command only once
```



```
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\User> 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  
006052311e5b87fb720cdca68c48b3d4dc391232a27d69e62dc946bf75eb824  
PS C:\Users\User>
```

Step 2: Visit <http://localhost:9000/> to open SonarQube. Login with username: admin and password: admin.



Step 3: Click on create a local project and name the project as sonarqube-test and key as sonarqube-test and click on the next button. In the next step select the “Use the global setting” option and click on create project.

localhost:9000/projects/create

sonarqube

Projects Issues Rules Quality Profiles Quality Gates Administration More Q

How do you want to create your project?

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

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

Import from Azure DevOps Setup

Import from Bitbucket Cloud Setup

Import from Bitbucket Server Setup

Import from GitHub Setup

Import from GitLab Setup

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

Create a local project

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

SonarQube™ technology is powered by [SonarSource SA](#) Community Edition v10.6 (92116) ACTIVE Documentation Web API

sonarqube

Projects Issues Rules Quality Profiles Quality Gates Administration More Q

1 of 2

Create a local project

Project display name *

Project key *

Main branch name *

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

Cancel Next

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

SonarQube™ technology is powered by [SonarSource SA](#) Community Edition v10.6 (92116) ACTIVE Documentation Web API

2 of 2

Set up project for Clean as You Code

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

Choose the baseline for new code for this project

Use the global setting

Previous version

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

Recommended for projects following regular versions or releases.

Define a specific setting for this project

Previous version

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

Recommended for projects following regular versions or releases.

Number of days

Any code that has changed in the last x days is considered new code. If no action is taken on a new issue after x days, this issue will become part of the overall code.

Recommended for projects following continuous delivery.

Reference branch

Choose a branch as the baseline for the new code.

Step 1 Jenkins

av... Dashboard > Manage Jenkins
plus

+ New Item Manage Jenkins Search settings /

Project Relationship Check File Fingerprint Manage Jenkins My Views

Build Queue Build Executor Status

Built-In Node 1 Idle 2 Idle Slave1 (offline)

System Configuration

Nodes Tools Plugins Appearance

Clouds

localhost:8080/manage/pluginManager

gins and select
ew the installed

Jenkins

Dashboard > Manage Jenkins > Plugins

Plugins

Search: sona

Updates Available plugins Installed plugins Advanced settings

Name: SonarQube Scanner for Jenkins 2.17.2
This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.
Report an issue with this plugin

Enabled

REST API Jenkins 2.452.3

St Jenkins

SC Dashboard > Manage Jenkins

New Item Build History Project Relationship Check File Fingerprint Manage Jenkins My Views

+ New Item

Manage Jenkins

Search settings

New version of Jenkins (2.462.2) is available for download (changelog). Or Upgrade Automatically

Building on the built-in node can be a security issue. You should set the number of executors on the built-in node to 0. See the documentation. Manage Dismiss

Warnings have been published for the following currently installed components:

Jenkins 2.452.3 core and libraries: Multiple security vulnerabilities in Jenkins 2.470 and earlier, LTS 2.452.3 and earlier A fix for this issue is available. Update Jenkins now.

Configure which of these warnings are shown

System Configuration

Build Queue Build Executor Status

Built-in Node: 1 Idle, 2 Idle, Slave1 (offline)

Nodes: Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Tools: Configure tools, their locations and automatic installers.

Clouds: Add, remove, and configure cloud instances to provision agents on-demand.

Plugins: 29 Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

Appearance: Configure the look and feel of Jenkins

localhost:8080/manage/configure

SonarQube servers

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

Environment variables

SonarQube installations

List of SonarQube installations

Name

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

Server authentication token
SonarQube authentication token. Mandatory when anonymous access is disabled.

[Advanced](#)

Jenkins

Dashboard > Manage Jenkins

sa

+ New Item

Manage Jenkins

Search (CTRL+K)

Anuprita Mhapankar

log out

to SonarQube
y then click on

 Project Relationship Check File Fingerprint Manage Jenkins My Views Build Queue Build Executor Status Built-In Node

1 Idle

2 Idle

 Slave1

^

System Configuration**System**

Configure global settings and paths.

**Tools**

Configure tools, their locations and automatic installers.

**Plugins**

Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

**Nodes**

Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

**Clouds**

Add, remove, and configure cloud instances to provision agents on-demand.

**Appearance**

Configure the look and feel of Jenkins

localhost:8080/manage/configureTools

SonarScanner for MSBuild installations

[Add SonarScanner for MSBuild](#)

SonarQube Scanner installations

[Add SonarQube Scanner](#)

SonarQube Scanner

Name

sonarqubescanner

 Install automatically

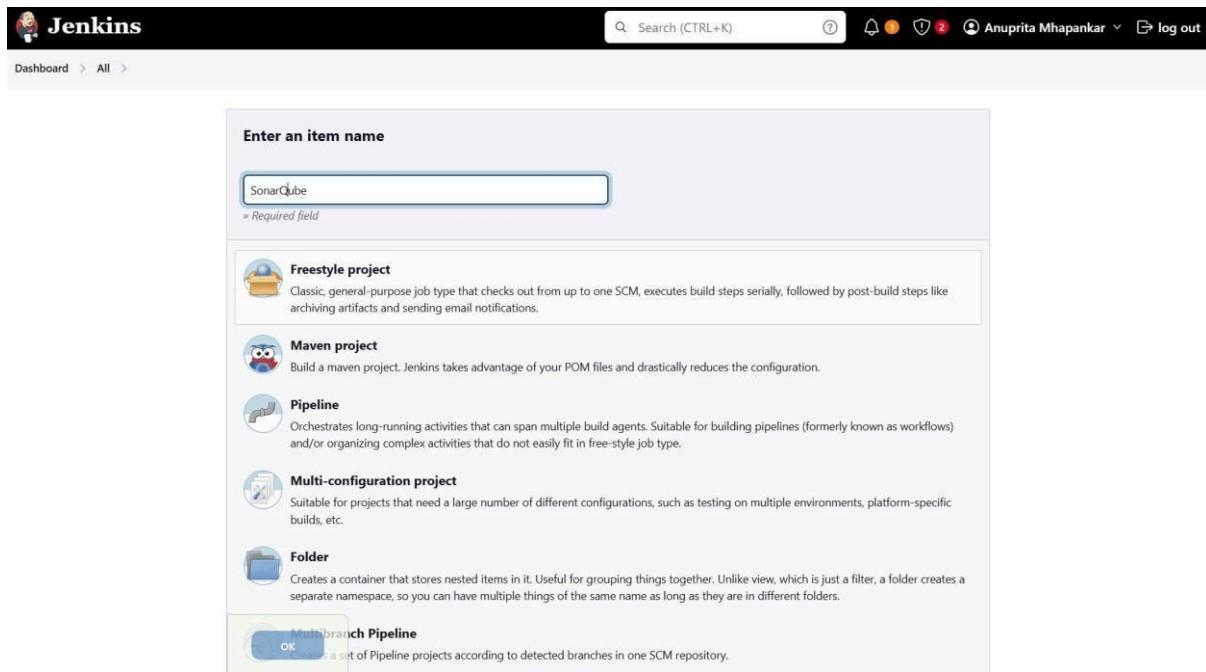
Install from Maven Central

Version

SonarQube Scanner 6.2.0.4584

[Add Installer](#)[Save](#)[Apply](#)

Step 7: Go to Jenkins dashboard and click on New Item and select Freestyle project and name it as SonarQube and then click on ok.



The screenshot shows the Jenkins dashboard with the user 'Anuprita Mhapankar' logged in. A new item is being created with the name 'SonarQube'. Below the name input, there is a list of job types:

- Freestyle project**: Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
- Maven project**: Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.
- Pipeline**: Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
- Multi-configuration project**: Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
- Folder**: Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
- Multi-branch Pipeline**: OK

Step 8: For configuration, Select git and paste the following git repository in the repository url.

https://github.com/shazforiot/MSBuild_firstproject

This is a simple Hello world project

Configure

Source Code Management

None

Git [?](#)

Repositories [?](#)

Repository URL [?](#)
https://github.com/shazforiot/MSBuild_firstproject.git

Credentials [?](#)
- none -

+ Add [▼](#)

Advanced [▼](#)

Add Repository

Branches to build [?](#)

Branch Specifier (blank for 'any') [?](#)
*/master

Save **Apply**

Step 9: Under the Build steps select “Execute SonarQube Scanner” option and under Analysis Properties write the following -

sonar.projectKey=sonarqube-test

sonar.login=admin

sonar.password=sonarqube

sonar.hosturl=http://sonarqube:900

0 Then click on the save button.

Configure

Build Steps

General

Source Code Management

Build Triggers

Build Environment

Build Steps [▼](#)

Post-build Actions

Execute SonarQube Scanner

JDK [?](#)
(Inherit From Job)

Path to project properties [?](#)

Analysis properties [?](#)
sonar.projectKey=sonarqube-test
sonar.login=admin
sonar.password=sonarqube
sonar.hosturl=http://sonarqube:9000

Additional arguments [?](#)

JVM Options [?](#)

Save **Apply**

Jenkins

Dashboard > SonarQube >

Status

</> Changes

Workspace

Build Now

Configure

Delete Project

SonarQube

Rename

Permalinks

Build History trend

No builds

Atom feed for all Atom feed for failures

Add description

Disable Project

REST API Jenkins 2.452.3

Step 10: Visit <http://localhost:9000/admin/permissions> and select the Users tab and for Administrator select the checkbox Execute Analysis.

sonarqube

Projects Issues Rules Quality Profiles Quality Gates Administration More Q

Administration

Configuration Security Projects System Marketplace

Global Permissions

Grant and revoke permissions to make changes at the global level. These permissions include editing Quality Profiles, executing analysis, and performing global system administration.

All Users Groups Search for users or groups...

Administrator System ? Administer ? Execute Analysis ? Create ?

| Administrator | admin | <input type="checkbox"/> | <input type="checkbox"/> Quality Gates | <input checked="" type="checkbox"/> Quality Profiles | <input type="checkbox"/> Projects |
|---------------|---------------------|--------------------------|--|--|-----------------------------------|
| A | Administrator admin | <input type="checkbox"/> | <input type="checkbox"/> Quality Gates | <input checked="" type="checkbox"/> Quality Profiles | <input type="checkbox"/> Projects |

1 of 1 shown

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

SonarQube™ technology is powered by SonarSource SA Community Edition v10.6 (92116) ACTIVE GPL v3 Community Documentation Plugins Web API

Step 11: Now, come back to Jenkins and click on Build Now. The build is success.

Jenkins

Dashboard > SonarQube > #4 > Console Output

Console Output

Started by user Anuprita Mhapankar
 Running as SYSTEM
 Building on the built-in node in workspace C:\ProgramData\Jenkins\jenkins\workspace\SonarQube
 The recommended git tool is: NONE
 No credentials specified
 > git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\jenkins\workspace\SonarQube\.git # timeout=10
 Fetching changes from the remote Git repository
 > git.exe config remote.origin.url https://github.com/shazforiot/MSBuild_firstproject.git # timeout=10
 Fetching upstream changes from https://github.com/shazforiot/MSBuild_firstproject.git
 > git.exe --version # timeout=10
 > git --version # 'git' version 2.41.0.windows.3'
 > 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 f2bc042c04c6e72427c380bcae6d6fee7b49adf (refs/remotes/origin/master)
 > git.exe config core.sparsecheckout # timeout=10
 > git.exe checkout -f f2bc042c04c6e72427c380bcae6d6fee7b49adf # timeout=10
 Commit message: "updated"
 > git.exe rev-list --no-walk f2bc042c04c6e72427c380bcae6d6fee7b49adf # timeout=10
 [SonarQube] \$ C:\ProgramData\Jenkins\jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqubescanner\bin\sonar-scanner.bat -Dsonar.host.url=http://localhost:9000 -Dsonar.projectKey=sonarqube-test -Dsonar.login=admin -Dsonar.hostUrl=http://sonarqube:9000 -Dsonar.password=sonarqube -Dsonar.projectBaseDir=C:\ProgramData\Jenkins\jenkins\workspace\SonarQube
 18:40:04.147 INFO Scanner configuration file:
 C:\ProgramData\Jenkins\jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqubescanner\bin..\conf\sonar-scanner.properties
 18:40:04.152 INFO Project root configuration file: NONE
 18:40:04.175 INFO SonarScanner CLI 6.2.0.4584
 18:40:04.177 INFO Java 21.0.4 Eclipse Adoptium (64-bit)
 18:40:04.184 INFO Windows 11 10.0 amd64

 18:40:41.286 INFO ----- Run sensors on project
 18:40:41.484 INFO Sensor C# [csharp]
 18:40:41.485 WARN Your project contains C# files which cannot be analyzed with the scanner you are using. To analyze C# or VB.NET, you must use the SonarScanner for .NET 5.x or higher, see <https://redirect.sonarsource.com/doc/install-configure-scanner-msbuild.html>
 18:40:41.485 INFO Sensor C# [csharp] (done) | time=2ms
 18:40:41.486 INFO Sensor Analysis Warnings import [csharp]
 18:40:41.488 INFO Sensor Analysis Warnings import [csharp] (done) | time=2ms
 18:40:41.488 INFO Sensor C# File Caching Sensor [csharp]
 18:40:41.489 WARN Incremental PR analysis: Could not determine common base path, cache will not be computed. Consider setting 'sonar.projectBaseDir' property.
 18:40:41.490 INFO Sensor C# File Caching Sensor [csharp] (done) | time=1ms
 18:40:41.491 INFO Sensor Zero Coverage Sensor
 18:40:41.508 INFO Sensor Zero Coverage Sensor (done) | time=19ms
 18:40:41.514 INFO SCM Publisher SCM provider for this project is: git
 18:40:41.517 INFO SCM Publisher 4 source files to be analyzed
 18:40:42.309 INFO SCM Publisher 4/4 source files have been analyzed (done) | time=791ms
 18:40:42.317 INFO CPD Executor Calculating CPD for 0 files
 18:40:42.318 INFO CPD Executor CPD calculation finished (done) | time=0ms
 18:40:42.326 INFO SCM revision ID 'f2bc042c04c6e72427c380bcae6d6fee7b49adf'
 18:40:42.522 INFO Analysis report generated in 181ms, dir size=201.1 kB
 18:40:42.588 INFO Analysis report compressed in 63ms, zip size=22.3 kB
 18:40:42.876 INFO Analysis report uploaded in 283ms
 18:40:42.880 INFO ANALYSIS SUCCESSFUL, you can find the results at: <http://localhost:9000/dashboard?id=sonarqube-test>
 18:40:42.881 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
 18:40:42.882 INFO More about the report processing at <http://localhost:9000/api/ce/task?id=d10eb30d-ebdd-4bb2-b564-0aa4ea71b0f2>
 18:40:42.916 INFO Analysis total time: 25.189 s
 18:40:42.926 INFO SonarScanner Engine completed successfully
 18:40:43.027 INFO EXECUTION SUCCESS
 18:40:43.029 INFO Total time: 38.885s
 Finished: SUCCESS

REST API Jenkins 2.452.3

Step 12: Visit the following URL to see the result - <http://localhost:9000/dashboard?id=sonarqube-test&codeScope=overall>

localhost:9000/dashboard?id=sonarqube-test&codeScope=overall

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration More ? A

sonarqube-test / main ?

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

main Version not provided · Set as homepage

Quality Gate **Passed** Last analysis 14 minutes ago

The last analysis has warnings. See details

New Code Overall Code

Security Reliability Maintainability

| Open issues | A |
|-------------|---|
| 0 H | |
| 0 M | |
| 0 L | |

| Open issues | A |
|-------------|---|
| 0 H | |
| 0 M | |
| 0 L | |

| Open issues | A |
|-------------|---|
| 0 H | |
| 0 M | |
| 0 L | |

Accepted issues Coverage Duplications

| Valid issues that were not fixed | Coverage | Duplications |
|----------------------------------|----------------------|----------------------|
| 0 | On 0 lines to cover. | 0.0% On 86 lines. |

Security Hotspots

This screenshot shows the SonarQube dashboard for the 'main' branch of the 'sonarqube-test' project. The top navigation bar includes links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, More, and a search bar. The dashboard header displays the project name and branch, along with a 'Quality Gate' status of 'Passed' (indicated by a green checkmark icon). A note below the status says 'The last analysis has warnings. See details'. The main content area is divided into several sections: Security, Reliability, Maintainability, Accepted issues, Coverage, and Duplications. Each section contains numerical values (e.g., 0 open issues, 0 accepted issues) and a corresponding badge with a letter grade (A). Below these sections are tabs for 'New Code' and 'Overall Code'. The 'Coverage' section notes that there are 0 lines to cover. The 'Duplications' section indicates 0.0% duplication on 86 lines. A 'Security Hotspots' section is partially visible at the bottom. The overall interface is clean and modern, using a light blue and white color scheme.

Experiment No 8

D15-A

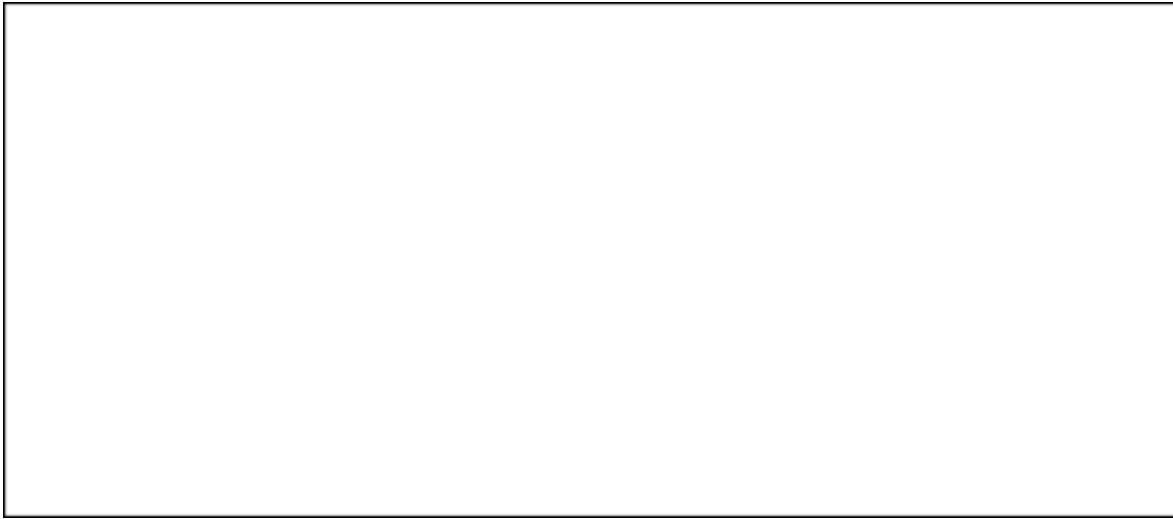
Hitesh Rohra - 47

Advance-Devops Experiment no:8

Step 1: Open Windows PowerShell and run the following command – docker run -d --name sonarqube-test1 -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest

```
erShell does not load commands from the current location by default. If you trust this command, instead type: ".\sonar-s
canner.bat". See "get-help about_Command_Precedence" for more details.
PS C:\sonar-scanner\sonar-scanner-6.2.0.4584-windows-x64\bin> .\sonar-scanner.bat
11:02:02.120 INFO Scanner configuration file: C:\sonar-scanner\sonar-scanner-6.2.0.4584-windows-x64\bin..\conf\sonar-s
canner.properties
11:02:02.124 INFO Project root configuration file: NONE
11:02:02.140 INFO SonarScanner CLI 6.2.0.4584
11:02:02.142 INFO Java 17.0.12 Eclipse Adoptium (64-bit)
11:02:02.142 INFO Windows 11 10.0 amd64
11:02:02.160 INFO User cache: C:\Users\navan\.sonar\cache
11:02:02.644 INFO JRE provisioning: os[windows], arch[amd64]
11:02:06.241 INFO EXECUTION FAILURE
11:02:06.243 INFO Total time: 4.126s
11:02:06.244 ERROR Error during SonarScanner CLI execution
java.lang.IllegalStateException: Error status returned by url [https://api.sonarcloud.io/analysis/jres?os=windows&arch=a
md64]: 401
        at org.sonarsource.scanner.lib.internal.http.ServerConnection.callUrl(ServerConnection.java:182)
        at org.sonarsource.scanner.lib.internal.http.ServerConnection.callApi(ServerConnection.java:145)
        at org.sonarsource.scanner.lib.internal.http.ServerConnection.callRestApi(ServerConnection.java:123)
        at org.sonarsource.scanner.lib.internal.JavaRunnerFactory.getJreMetadata(JavaRunnerFactory.java:159)
        at org.sonarsource.scanner.lib.internal.JavaRunnerFactory.getJreFromServer(JavaRunnerFactory.java:138)
        at org.sonarsource.scanner.lib.internal.JavaRunnerFactory.createRunner(JavaRunnerFactory.java:85)
        at org.sonarsource.scanner.lib.internal.ScannerEngineLauncherFactory.createLauncher(ScannerEngineLauncherFactory
.java:53)
        at org.sonarsource.scanner.lib.ScannerEngineBootstrapper.bootstrap(ScannerEngineBootstrapper.java:118)
        at org.sonarsource.scanner.cli.Main.analyze(Main.java:75)
        at org.sonarsource.scanner.cli.Main.main(Main.java:63)
11:02:06.246 ERROR
11:02:06.246 ERROR Re-run SonarScanner CLI using the -X switch to enable full debug logging.
PS C:\sonar-scanner\sonar-scanner-6.2.0.4584-windows-x64\bin> |
```

- Login to SonarQube using username admin and password admin.
- Create a manual project in SonarQube with the name sonarqube-test1



Dashboard > All >

Enter an item name

> Required field

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

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

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

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

OK Create a folder

A container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a

Step2: go to the jenkins and create new item select pipeline:

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/"  
        }  
    }  
}
```

}

The screenshot shows the SonarQube Pipeline Configuration page. At the top, there is a large, empty rectangular area. Below it, the URL is: Dashboard > SonarQube Pipeline > Configuration. The page title is "Pipeline". The left sidebar has sections: General, Advanced Project Options, and Pipeline (which is selected). The main content area has tabs: Definition (selected), Script (with a question mark icon), and Pipeline Syntax. The "Script" tab contains the following Groovy code:

```
1 < node {
2   stage('Cloning the Github Repo') {
3     git 'https://github.com/nazfonit/MSBuildFirstProject.git'
4   }
5   stage('SonarQube analysis') {
6     withSonarQubeEnv('sonarqube-test1') {
7       bat "C:/sonar-scanner/sonar-scanner-6.2.0.4584-windows-x64/bin/sonar-scanner.bat \
8           -D sonar.login=admin \
9           -D sonar.password=d23 \
10          -D sonar.projectKey=sonarqube-test1 \
11          -D sonar.exclusions=wwwdler/**,wwwunit/**/*_*/*.java \
12          -D sonar.host.url=http://localhost:9000"
13     }
14   }
15 }
```

Below the script, there is a checkbox labeled "Use Groovy Sandbox" which is checked. At the bottom, there are "Save" and "Apply" buttons.

Save the changes and go to build now:



Dashboard > SonarQube Pipeline >

SonarQube Pipeline

Status: Success

Changes: 0

Build Now

Configure

Delete Pipeline

Move

Full Stage View

SonarQube

Stages

Rename

Pipeline Syntax

Build History

trend

Filter...

Average stage times:
(Average full run time: ~34s)

Stage View

| Cloning the GitHub Repo | SonarQube Success analysis |
|-------------------------|---|
| 18s | Logs |
| 3s | 30s |
| 2s | 133ms failed |
| 9s | 314ms failed |

Add description

Disable Project

The screenshot shows the SonarQube Pipeline interface. On the left, there's a sidebar with options like Status, Changes, Build Now, Configure, Delete Pipeline, Move, Full Stage View, SonarQube, Stages, Rename, Pipeline Syntax, Build History, and a Filter dropdown. The main area is titled 'SonarQube Pipeline' with a green success status icon. It displays a 'Stage View' grid with three columns: 'Cloning the GitHub Repo', 'SonarQube Success analysis', and a third column where logs are available. Below the grid, it says 'Average stage times: (Average full run time: ~34s)'. The build history shows three runs: Run 27 (Sep 27, 11:11, No Changes), Run 26 (Sep 27, 11:16, No Changes), and Run 25 (Sep 27, 11:11, No Changes). The first two runs completed successfully, while the third failed. The failed runs show execution times of 3s, 2s, 9s, 30s, 133ms, and 314ms respectively. Buttons for 'Add description' and 'Disable Project' are at the top right.

Console output:

Console Output

```
Started by user jai navani
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\SonarQube Pipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Cloning the GitHub Repo)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\SonarQube Pipeline\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/shazforiot/MSBuild_firstproject.git # timcout=10
Fetching upstream changes from https://github.com/shazforiot/MSBuild_firstproject.git
> git.exe --version # timeout=10
> git --version # 'git version 2.45.2.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/shazforiot/MSBuild_firstproject.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision f2bc042c04c6e72427c380bcae6d6fee7b49adf (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f f2bc042c04c6e72427c380bcae6d6fee7b49adf # timeout=10
> git.exe branch -a -v --no-abbrev # timeout=10
> git.exe branch -D master # timeout=10
> git ...
```

```
11:22:18.236 INFO Sensor C# File Caching Sensor [csharp]
11:22:18.237 WARN Incremental PR analysis: Could not determine common base path, cache will not be computed. Consider setting 'sonar.projectBaseDir' property.
11:22:18.237 INFO Sensor C# File Caching Sensor [csharp] (done) | time=1ms
11:22:18.237 INFO Sensor Zero Coverage Sensor
11:22:18.251 INFO Sensor Zero Coverage Sensor (done) | time=14ms
11:22:18.256 INFO SCM Publisher SCM provider for this project is: git
11:22:18.257 INFO SCM Publisher 4 source files to be analyzed
11:22:18.789 INFO SCM Publisher 4/4 source files have been analyzed (done) | time=531ms
11:22:18.793 INFO CPD Executor Calculating CPD for 0 files
11:22:18.795 INFO CPD Executor CPD calculation finished (done) | time=0ms
11:22:18.810 INFO SCM revision ID 'f2bc042c04c6e72427c380bc4ee6d6fee7b49adf'
11:22:19.074 INFO Analysis report generated in 134ms, dir size=201.0 kB
11:22:19.137 INFO Analysis report compressed in 45ms, zip size=22.5 kB
11:22:19.351 INFO Analysis report uploaded in 212ms
11:22:19.353 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube-test1
11:22:19.354 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
11:22:19.354 INFO More about the report processing at http://localhost:9000/api/ce/task?id=9/1ae2f2-4e0f-49a1-88c4-ad4a6cceddff8
11:22:19.366 INFO Analysis total time: 24.819 s
11:22:19.368 INFO SonarScanner Engine completed successfully
11:22:19.454 INFO EXECUTION SUCCESS
11:22:19.455 INFO Total time: 29.696s
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```



sonarcube

Projects Issues Rules Quality Profiles Quality Gates Administration More Q

star sonarcube-test1 / main ✓ ?

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

main Version not provided Set as homepage

Quality Gate Last analysis 12 minutes ago

Passed

The last analysis has warnings. See details

New Code Overall Code

| Security | Reliability | Maintainability |
|------------------------------|------------------------------|------------------------------|
| 0 Open issues 0 H 0 M 0 L | 0 Open issues 0 H 0 M 0 L | 0 Open issues 0 H 0 M 0 L |
| Accepted issues 0 | Coverage ⌚ | Duplications 0.0% |

Experiment No 9

Screenshots for Experiment 9

Launch an ec2 instance

Give name use the default OS

EC2 > Instances > Launch an instance

Launch an instance Info

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

Name and tags Info

Name
nagios_host_exp_9kcs [Add additional tags](#)

▼ Application and OS Images (Amazon Machine Image) Info

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

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

Recents [Quick Start](#)

Amazon Linux 
macOS 
Ubuntu 
Windows 
Red Hat 

S >  [Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Hitesh Rohra 47 D15A
Make a key pair and use it.

▼ Key pair (login) [Info](#)

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

Key pair name - *required*

nagios_exp_9 ▼ C [Create new key pair](#)

▼ Network settings [Info](#) Edit

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

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

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

[Additional charges apply](#) when outside of free tier allowance

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

Create security group ○ Select existing security group

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

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

Allow HTTPS traffic from the internet

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



go to security groups:

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with various EC2-related options like Dashboard, Global View, Events, Instances, Images, Elastic Block Store, Network & Security, and Security Groups. The 'Instances' section is expanded, showing 'Instances' and 'Launch Templates'. Under 'Instances', three instances are listed: 'Master' (i-Oab175e9c60cc3a23), 'node-1' (i-08ad30b7114767ca2), and 'node-2' (i-03c70d364fb762af5). A search bar at the top right says 'Find Instance by attribute or tag (case-sensitive)'. Below the instances, a button says 'Select an instance'.

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

The screenshot shows the AWS Security Groups page. On the left, there's a sidebar with 'view' and 'New' buttons. The main area shows a table of security groups with columns: Name, Security group ID, Security group name, and VPC ID. There are 9 entries listed:

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

now click on edit inbound rules

The screenshot shows the 'Inbound rules' tab selected in a dashboard. At the top, there are tabs for 'Inbound rules' (which is active), 'Outbound rules', and 'Tags'. Below the tabs, a search bar and a button to manage tags are visible. A large 'Edit inbound rules' button is prominently displayed. The main area shows a table titled 'Inbound rules (1)'. The table has columns for Name, Security group rule..., IP version, Type, and Protocol. One row is listed: 'sgr-0d6a171458e586b3e' (Name), 'SSH' (Security group rule...), 'IPv4' (IP version), 'SSH' (Type), and 'TCP' (Protocol). There are checkboxes next to each row.

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

The screenshot shows a detailed view of the 'Inbound rules' configuration. It lists several security group rules with their details: 1. Rule ID: sgr-0d6a171458e586b3e, Type: SSH, Protocol: TCP, Port range: 22, Source: Custom (0.0.0.0/0). 2. Rule ID: -, Type: HTTP, Protocol: TCP, Port range: 80, Source: Anywhere (0.0.0.0/0). 3. Rule ID: -, Type: All ICMP - IPv6, Protocol: IPv6 ICMP, Port range: All, Source: Anywhere (0.0.0.0/0). 4. Rule ID: -, Type: HTTPS, Protocol: TCP, Port range: 443, Source: Anywhere (0.0.0.0/0). 5. Rule ID: -, Type: All traffic, Protocol: All, Port range: All, Source: Anywhere (0.0.0.0/0). 6. Rule ID: -, Type: Custom TCP, Protocol: TCP, Port range: 5666, Source: Anywhere (0.0.0.0/0). 7. Rule ID: -, Type: All ICMP - IPv4, Protocol: ICMP, Port range: All, Source: Anywhere (0.0.0.0/0). At the bottom left, there is a 'Add rule' button.

then click on save rules.

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

sg-06ac4c5a9779ecaf9 - launch-wizard-5

Details

| | | | |
|--|---|---|---------------------------------|
| Security group name launch-wizard-5 | Security group ID sg-06ac4c5a9779ecaf9 | Description Launch-wizard-5 created 2024-09-28T03:55:31.506Z | VPC ID vpc-07b0966cbfba88ee3 |
| Owner 209322483715 | Inbound rules count 7 | Outbound rules count 1 Permission entry | |

Inbound rules (7)

| Name | Security group rule... | IP version | Type | Protocol | Port range | Source | Description |
|------|-------------------------|------------|-----------------|-----------|------------|-----------|-------------|
| - | sgr-0576790f6a0b600b | IPv6 | All ICMP - IPv6 | IPv6 ICMP | All | ::/0 | - |
| - | sgr-066e171456586... | IPv4 | SSH | TCP | 22 | 0.0.0.0/0 | - |
| - | sgr-0b17ca9ed06e3b5... | IPv4 | HTTPS | TCP | 443 | 0.0.0.0/0 | - |
| - | sgr-0d3d582940a2ebaf0 | IPv4 | Custom TCP | TCP | 5666 | 0.0.0.0/0 | - |
| - | sgr-0e782e6d47b7b44f5 | IPv4 | All ICMP - IPv4 | ICMP | All | 0.0.0.0/0 | - |
| - | sgr-00bbdd4767cde375... | IPv6 | HTTP | TCP | 80 | ::/0 | - |
| - | sgr-0c81dac37a4a6020e | IPv4 | All traffic | All | All | 0.0.0.0/0 | - |

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

Instances (1/4) [Info](#)

| Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone | Public IPv4 DNS | Public IPv4 | Elastic IP | IPv6 IPs |
|-----------------------------|-----------------------|----------------|---------------|-------------------|-------------------------------|-------------------|--------------------------|----------------|------------|----------|
| Master | i-0ab175e9c60cc3a23 | Running | t2.micro | 2/2 checks passed | View alarms + | us-east-1b | ec2-54-165-203-193.co... | 54.165.203.193 | - | - |
| node-1 | i-08ad30b7114767ca2 | Running | t2.micro | 2/2 checks passed | View alarms + | us-east-1b | ec2-52-23-200-179.co... | 52.23.200.179 | - | - |
| node-2 | i-03c70d364fb762af5 | Running | t2.micro | 2/2 checks passed | View alarms + | us-east-1b | ec2-3-85-164-72.comp... | 3.85.164.72 | - | - |
| nagios_host_exp_9kcs | i-0820376be204a7fc... | Running | t2.micro | 2/2 checks passed | View alarms + | us-east-1b | ec2-54-205-31-174.co... | 54.205.31.174 | - | - |

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

EC2 > Instances > i-0820376be204a7fc... > Connect to instance

Connect to instance [Info](#)

Connect to your instance i-0820376be204a7fc (nagios_host_exp_9kcs) using any of these options

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

Instance ID
[i-0820376be204a7fc \(nagios_host_exp_9kcs\)](#)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is `nagios_exp_9.pem`
3. Run this command, if necessary, to ensure your key is not publicly viewable.
 `chmod 400 "nagios_exp_9.pem"`
4. Connect to your instance using its Public DNS:
 `ec2-54-205-31-174.compute-1.amazonaws.com`

Example:
`ssh -i "nagios_exp_9.pem" ec2-user@ec2-54-205-31-174.compute-1.amazonaws.com`

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

Hitesh Rohra 47 D15A

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

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

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

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

now paste the following commands in your connected terminal:

```
sudo yum update
```

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

```
sudo yum install httpd php
```

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

(type y when prompted)

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

Hitesh Rohra 47 D15A

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

Transaction Summary
=====
Install 13 Packages

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

sudo yum install gd gd-devel

```
google-noto-sans-vf-fonts-20201206-2.amzn2023.0.2.noarch
graphite2-devel-1.3.14-7.amzn2023.0.2.x86_64
harfbuzz-devel-7.0.0-2.amzn2023.0.1.x86_64
jbigkit-libs-2.1-21.amzn2023.0.2.x86_64
libICE-1.0.10-6.amzn2023.0.2.x86_64
libX11-1.7.2-3.amzn2023.0.4.x86_64
libX11-devel-1.7.2-3.amzn2023.0.4.x86_64
libXau-1.0.9-6.amzn2023.0.2.x86_64
libXext-1.3.4-6.amzn2023.0.2.x86_64
libXpm-devel-3.5.15-2.amzn2023.0.3.x86_64
libXt-1.2.0-4.amzn2023.0.2.x86_64
libffi-devel-3.4.4-1.amzn2023.0.1.x86_64
libicu-devel-67.1-7.amzn2023.0.3.x86_64
libjpeg-turbo-devel-2.1.4-2.amzn2023.0.5.x86_64
libpng-2.1.6.37-10.amzn2023.0.6.x86_64
libselinux-devel-3.4-5.amzn2023.0.2.x86_64
libtiff-4.4.0-4.amzn2023.0.18.x86_64
libwebp-1.2.4-1.amzn2023.0.6.x86_64
libxcb-1.13.1-7.amzn2023.0.2.x86_64
libxml2-devel-2.10.4-1.amzn2023.0.6.x86_64
pcre2-utf16-10.40-1.amzn2023.0.3.x86_64
pixman-0.40.0-3.amzn2023.0.3.x86_64
xml-common-0.6.3-56.amzn2023.0.2.noarch
xz-devel-5.2.5-9.amzn2023.0.2.x86_64

graphite2-1.3.14-7.amzn2023.0.2.x86_64
harfbuzz-7.0.0-2.amzn2023.0.1.x86_64
harfbuzz-icu-7.0.0-2.amzn2023.0.1.x86_64
langpacks-core-font-en-3.0-21.amzn2023.0.4.noarch
libSM-1.2.3-8.amzn2023.0.2.x86_64
libX11-common-1.7.2-3.amzn2023.0.4.noarch
libX11-xcb-1.7.2-3.amzn2023.0.4.x86_64
libXau-devel-1.0.9-6.amzn2023.0.2.x86_64
libXpm-3.5.15-2.amzn2023.0.3.x86_64
libXrender-0.9.10-14.amzn2023.0.2.x86_64
libblkid-devel-2.37.4-1.amzn2023.0.4.x86_64
libicu-67.1-7.amzn2023.0.3.x86_64
libjpeg-turbo-2.1.4-2.amzn2023.0.5.x86_64
libmount-devel-2.37.4-1.amzn2023.0.4.x86_64
libpng-devel-2.1.6.37-10.amzn2023.0.6.x86_64
libsep0-devel-3.4-3.amzn2023.0.3.x86_64
libtiff-devel-4.4.0-4.amzn2023.0.18.x86_64
libwebp-devel-1.2.4-1.amzn2023.0.6.x86_64
libxcb-devel-1.13.1-7.amzn2023.0.2.x86_64
pcre2-devel-10.40-1.amzn2023.0.3.x86_64
pcre2-utf32-10.40-1.amzn2023.0.3.x86_64
sysprof-capture-devel-3.40.1-2.amzn2023.0.2.x86_64
xorg-x11proto-devel-2021.4-1.amzn2023.0.2.noarch
zlib-devel-1.2.11-33.amzn2023.0.5.x86_64

complete!
root@user@ip-172-21-80-127:~$ |
```

sudo adduser -m nagios

sudo passwd nagios

Hitesh Rohra 47 D15A

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

(add a password here)

sudo groupadd nagcmd

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

sudo usermod -a -G nagcmd nagios

sudo usermod -a -G nagcmd apache

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

mkdir ~/downloads

cd ~/downloads

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

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

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

nagios-4.5.5.tar.gz          100%[=====]  1.97M  5.30MB/s   in 0.4s
2024-09-28 06:27:52 (5.30 MB/s) - 'nagios-4.5.5.tar.gz' saved [2065473/2065473]
[ec2-user@ip-172-31-80-137 downloads]$ |
```

Hitesh Rohra 47 D15A

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

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

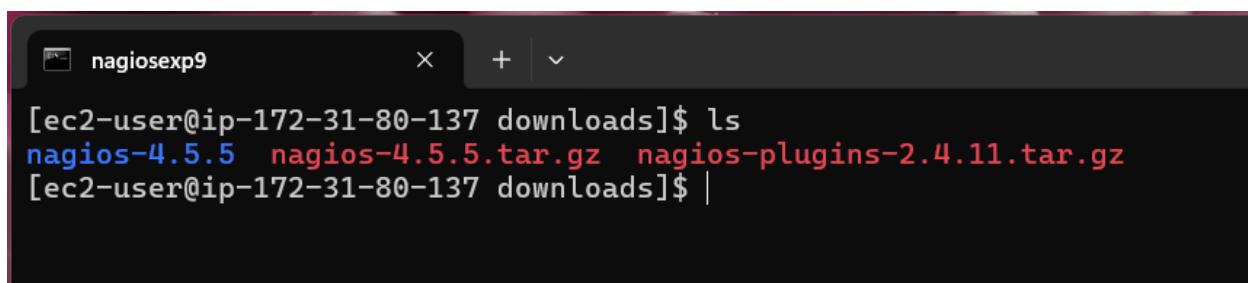
nagios-plugins-2.4.11.tar.gz 100%[=====] 2.62M 5.90MB/s in 0.4s

2024-09-28 06:28:15 (5.90 MB/s) - 'nagios-plugins-2.4.11.tar.gz' saved [2753049/2753049]

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

tar zxvf nagios-4.5.5.tar.gz

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



The screenshot shows a terminal window titled "nagiosexp9". The command "ls" is run, displaying three files: "nagios-4.5.5", "nagios-4.5.5.tar.gz", and "nagios-plugins-2.4.11.tar.gz".

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

cd nagios-4.5.5

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

sudo yum install openssl-devel

Hitesh Rohra 47 D15A

```
[ec2-user@ip-172-31-80-137 nagios-4.5.5]$ sudo yum install openssl-devel
Last metadata expiration check: 2:31:25 ago on Sat Sep 28 03:59:04 2024.
Dependencies resolved.
=====
 Package           Architecture      Version       Repository      Size
=====
Installing:
openssl-devel      x86_64          1:3.0.8-1.amzn2023.0.14      amazonlinux   3.0 M
Transaction Summary
=====
Install 1 Package
Total download size: 3.0 M
Installed size: 4.7 M
Is this ok [y/N]: y
```

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

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

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

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

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

make all

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

For more information on obtaining support for Nagios, visit:

  https://support.nagios.com

*****
Enjoy.
```

sudo make install

Hitesh Rohra 47 D15A

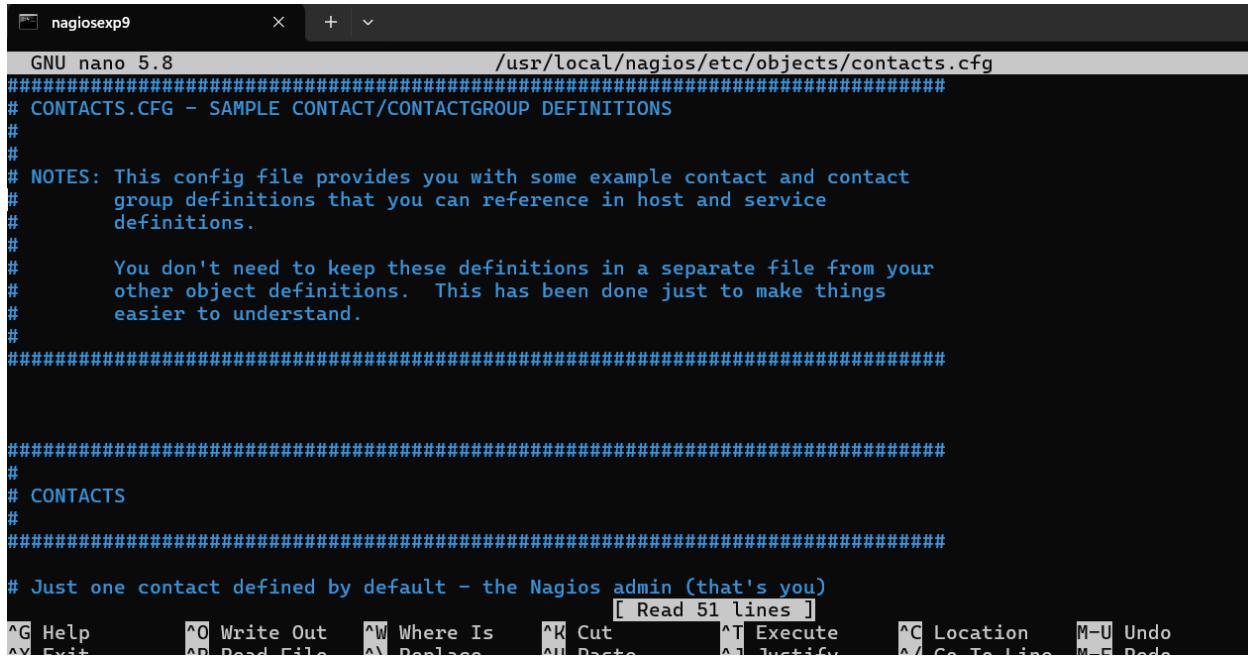
sudo make install-init

sudo make install-config

sudo make install-commandmode

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

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



```
nagiosexp9      x  +  v
GNU nano 5.8                               /usr/local/nagios/etc/objects/contacts.cfg
#####
# CONTACTS.CFG - SAMPLE CONTACT/CONTACTGROUP DEFINITIONS
#
#
# NOTES: This config file provides you with some example contact and contact
# group definitions that you can reference in host and service
# definitions.
#
# You don't need to keep these definitions in a separate file from your
# other object definitions. This has been done just to make things
# easier to understand.
#
#####
#
# CONTACTS
#
#
# Just one contact defined by default - the Nagios admin (that's you)
[ Read 51 lines ]
^G Help      ^O Write Out    ^W Where Is     ^K Cut        ^T Execute    ^C Location   M-U Undo
^V Edit      ^R Read File    ^L Paste      ^U Paste      ^Y Paste      ^I Go To Line M-E Redo
```

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

```
# This contact definition inherits a lot of default values from the
# 'generic-contact' template which is defined elsewhere.

define contact {

    contact_name      nagiosadmin      ; Short name of user
    use               generic-contact   ; Inherit default values from generic-contact template (defined above)
    alias             Nagios Admin    ; Full name of user
    email            nagios@localhost ; <***** CHANGE THIS TO YOUR EMAIL ADDRESS *****

}

#####
#
```

Hitesh Rohra 47 D15A

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

define contact {

    contact_name      nagiosadmin          ; Short name of user
    use               generic-contact       ; Inherit default values from generic-contact template (define
    alias             Nagios Admin        ; Full name of user
    email             2022.shubham.jha@ves.ac.in| ; <***** CHANGE THIS TO YOUR EMAIL ADDRESS *****

}

#####
#
# CONTACT GROUPS

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

press Ctrl+O and then enter.

then press Ctrl +X

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

*** External command directory configured ***

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

sudo make install-webconf

```
[ec2-user@ip-172-31-80-137 nagios-4.5.5]$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf
if [ 0 -eq 1 ]; then \
    ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi

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

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

Adding password for nagios admin

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

```
[ec2-user@ip-172-31-80-137 nagios-4.5.5]$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
[ec2-user@ip-172-31-80-137 nagios-4.5.5]$ |
```

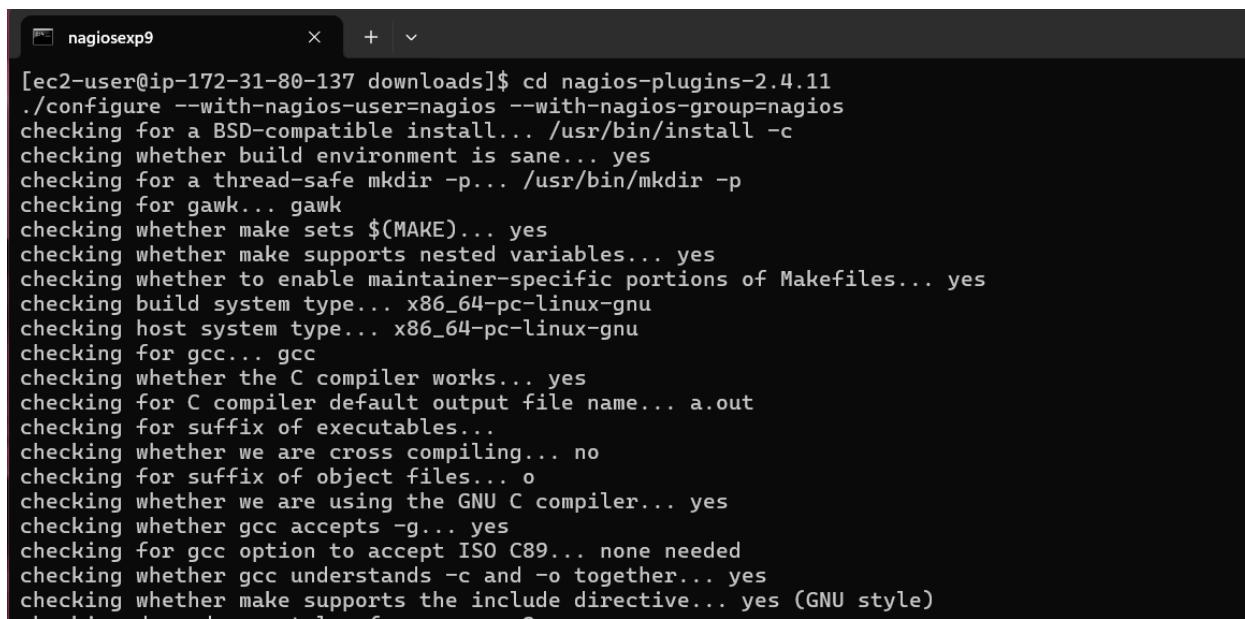
sudo service httpd restart

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

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

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

```
cd nagios-plugins-2.4.11  
.configure --with-nagios-user=nagios --with-nagios-group=nagios
```



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

make

sudo make install

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

sudo chkconfig --add nagios

sudo chkconfig nagios on

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

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

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

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

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

Running pre-flight check on configuration data...

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

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

```
[ec2-user@ip-172-31-80-137 nagios-plugins-2.4.11]$ sudo service nagios start
Redirecting to /bin/systemctl start nagios.service
[ec2-user@ip-172-31-80-137 nagios-plugins-2.4.11]$ |
```

sudo systemctl status nagios

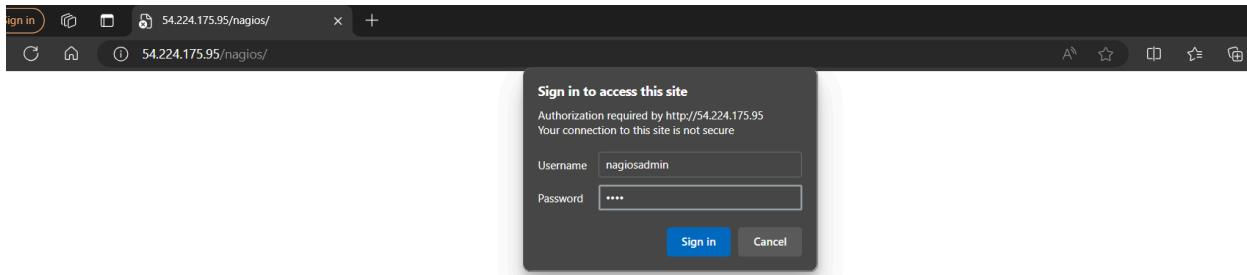
```
[ec2-user@ip-172-31-80-137 nagios-plugins-2.4.11]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
   Active: active (running) since Sat 2024-09-28 07:40:16 UTC; 35s ago
     Docs: https://www.nagios.org/documentation
 Process: 71009 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Process: 71010 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 71011 (nagios)
   Tasks: 6 (limit: 1112)
    Memory: 5.6M
      CPU: 82ms
     CGroup: /system.slice/nagios.service
             └─71011 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
                  ├─71012 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                  ├─71013 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                  ├─71014 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                  ├─71015 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                  └─71016 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

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

The screenshot shows the AWS EC2 Instances page. The instance summary for 'i-0820376be204a7fcb (nagios host exp_9kcs)' is displayed. Key details include:

- Instance ID:** i-0820376be204a7fcb (nagios_host_exp_9kcs)
- IPv4 address:** 54.224.175.95 (highlighted with a green border)
- Instance state:** Running
- Private IP DNS name (IPv4 only):** ip-172-31-80-137.ec2.internal
- Instance type:** t2.micro
- VPC ID:** vpc-07b6966cfba88ee5
- Subnet ID:** subnet-029be9bc13a1f9f65
- Instance ARN:** arn:aws:ec2:us-east-1:209322483715:instance/i-0820376be204a7fcb

A tooltip 'Public IPv4 address copied' is shown over the public IP address field.



Nagios Core™

✓ Daemon running with PID 3152

Nagios® Core™
Version 4.5.5
September 17, 2024
Check for updates

Get Started

- Start monitoring your infrastructure
- Change the look and feel of Nagios
- Extend Nagios with hundreds of addons
- Get support
- Get training
- Get certified

Quick Links

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

Latest News

Don't Miss...

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

General

- Home
- Documentation

Current Status

- Tactical Overview
- Map
- Hosts
- Services
- Host Groups
- Summary
- Grid

Service Groups

- Summary
- Grid

Problems

- Services (Unhandled)
- Hosts (Unhandled)
- Network Outages

Reports

- Availability
- Trends
- Alerts
- History
- Summary
- Histogram
- Notifications
- Event Log

System

- Comments
- Downtime
- Process Info
- Performance Info

Experiment No 10

Screenshots for Experiment 10

```
● nagios.service - Nagios Core 4.4.14
   Loaded: loaded (/lib/systemd/system/nagios.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2023-09-30 08:54:01 UTC; 20s ago
     Docs: https://www.nagios.org/documentation
 Process: 55285 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Process: 55286 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 55287 (nagios)
   Tasks: 6 (limit: 1141)
  Memory: 5.3M
    CPU: 252ms
   CGroup: /system.slice/nagios.service
           ├─55287 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           ├─55288 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─55289 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─55290 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─55291 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─55292 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 30 08:54:01 ip-172-31-44-151 nagios[55287]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfully initialized
lines 1-19]
```

Monitoring a Linux machine, create an Ubuntu 20.04 server EC2 Instance in AWS.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with 'New EC2 Experience' and a 'Tell us what you think' link. Under 'Instances', it lists three instances: 'cutenagios_se...', 'cutenagios_cli...', and 'cutenagios_se...'. Each instance is shown with its Name, Instance ID, Instance state (Running), Instance type (t2.micro), Status check (green), Alarm status (No alarms), and Availability zone (ap-south-1). A 'Launch instances' button is at the top right.

| Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability |
|-------------------|---------------------|----------------|---------------|--------------|--------------|--------------|
| cutenagios_se... | i-09d6b0d2e181a7287 | Running | t2.micro | green | No alarms | ap-south-1 |
| cutenagios_cli... | i-0e36968400dac0991 | Running | t2.micro | green | No alarms | ap-south-1 |
| cutenagios_se... | i-03a3e79fc5ab0a056 | Stopped | t2.micro | grey | No alarms | ap-south-1 |

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

```
*** System restart required ***
Last login: Sat Sep 30 08:31:30 2023 from 13.233.177.3
ubuntu@ip-172-31-44-151:~$ sudo apt install gcc -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gcc is already the newest version (4:11.2.0-1ubuntu1).
gcc set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
ubuntu@ip-172-31-44-151:~$ 

root@ip-172-31-44-151:/home/ubuntu# sudo apt install nagios-nrpe-server nagios-plugins
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'monitoring-plugins' instead of 'nagios-plugins'
monitoring-plugins is already the newest version (2.3.1-1ubuntu4).
nagios-nrpe-server is already the newest version (4.0.3-1ubuntu2).
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.

Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Fetched 229 kB in 1s (290 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
2 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@ip-172-31-44-151:/home/ubuntu# sudo apt install gcc -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gcc is already the newest version (4:11.2.0-1ubuntu1).
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
root@ip-172-31-44-151:/home/ubuntu# sudo apt install -y nagios-nrpe-server nagios-plugins
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'monitoring-plugins' instead of 'nagios-plugins'
monitoring-plugins is already the newest version (2.3.1-1ubuntu4).
nagios-nrpe-server is already the newest version (4.0.3-1ubuntu2).
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
```

sudo nano /etc/nagios/nrpe.cfg

```
GNU nano 6.2                               /etc/nagios/nrpe.cfg
// SERVER ADDRESS
// Address that nrpe should bind to in case there are more than one interface
// and you do not want nrpe to bind on all interfaces.
// NOTE: This option is ignored if NRPE is running under either inetd or xinetd
server_address=127.0.0.1

// LISTEN QUEUE SIZE
// Listen queue size (backlog) for serving incoming connections.
// You may want to increase this value under high load.
listen_queue_size=5

^G Help      ^C Write Out    ^W Where Is    ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File    ^\ Replace     ^U Paste      ^J Justify    ^/ Go To Line M-E Redo
                                         M-A Set Mark M-6 Copy

^Z 0-7600040001 (standard input) 1
```

```
GNU nano 6.2                               /etc/nagios/nrpe.cfg *
95 # that are allowed to talk to the NRPE daemon. Network addresses with a bit mask
96 # (i.e. 192.168.1.0/24) are also supported. Hostname wildcards are not currently
97 # supported.
98 #
99 # Note: The daemon only does rudimentary checking of the client's IP
100 # address. I would highly recommend adding entries in your /etc/hosts.allow
101 # file to allow only the specified host to connect to the port
102 # you are running this daemon on.
103 #
104 # NOTE: This option is ignored if NRPE is running under either inetd or xinetd
105 #
106 allowed_hosts=127.0.0.1,::1,13.235.0.144
107 server_address=0.0.0.0
108
109
110

^G Help      ^C Write Out    ^W Where Is    ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File    ^\ Replace     ^U Paste      ^J Justify    ^/ Go To Line M-E Redo
                                         M-A Set Mark M-6 Copy
```

sudo systemctl restart nagios-nrpe-server

```
Restarting services...
Service restarts being deferred:
/etc/needrestart/restart.d/dbus.service
systemctl restart getty@tty1.service
systemctl restart networkd-dispatcher.service
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service
systemctl restart user@1000.service

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-41-41:/home/ubuntu# sudo nano /etc/nagios/nrpe.cfg
root@ip-172-31-41-41:/home/ubuntu# sudo nano /etc/nagios/nrpe.cfg
root@ip-172-31-41-41:/home/ubuntu# sudo systemctl restart nagios-nrpe-server
root@ip-172-31-41-41:/home/ubuntu# sudo systemctl status nagios-nrpe-server
● nagios-nrpe-server.service - Nagios Remote Plugin Executor
```

Hitesh Rohra 47 D15A

```
root@ip-172-31-41-41:/home/ubuntu# sudo systemctl status nagios-nrpe-server
● nagios-nrpe-server.service - Nagios Remote Plugin Executor
   Loaded: loaded (/lib/systemd/system/nagios-nrpe-server.service; enabled; vendor preset: enabled)
     Active: active (running) since Sat 2023-09-30 09:27:17 UTC; 6s ago
       Docs: http://www.nagios.org/documentation
   Main PID: 7349 (nrpe)
      Tasks: 1 (limit: 1141)
        Memory: 1.5M
          CPU: 9ms
        CGroup: /system.slice/nagios-nrpe-server.service
                  └─7349 /usr/sbin/nrpe -c /etc/nagios/nrpe.cfg -f

Sep 30 09:27:17 ip-172-31-41-41 systemd[1]: nagios-nrpe-server.service: Deactivated successfully.
Sep 30 09:27:17 ip-172-31-41-41 systemd[1]: Stopped Nagios Remote Plugin Executor.
Sep 30 09:27:17 ip-172-31-41-41 systemd[1]: Started Nagios Remote Plugin Executor.
Sep 30 09:27:17 ip-172-31-41-41 nrpe[7349]: Starting up daemon
Sep 30 09:27:17 ip-172-31-41-41 nrpe[7349]: Server listening on 0.0.0.0 port 5666.
Sep 30 09:27:17 ip-172-31-41-41 nrpe[7349]: Listening for connections on port 5666
Sep 30 09:27:17 ip-172-31-41-41 nrpe[7349]: Allowing connections from: 127.0.0.1,::1,13.235.0.144
root@ip-172-31-41-41:/home/ubuntu# █
```

ps -ef | grep nagios

```
root@ip-172-31-44-151:/home/ubuntu# ps -ef | grep nagios
nagios    55287      1  0 08:54 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios    55288    55287  0 08:54 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios    55289    55287  0 08:54 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios    55290    55287  0 08:54 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios    55291    55287  0 08:54 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios    55292    55287  0 08:54 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios    56327      1  0 08:58 ?        00:00:00 /usr/sbin/nrpe -c /etc/nagios/nrpe.cfg -f
root     60903   60158  0 09:32 pts/1    00:00:00 grep --color=auto nagios
root@ip-172-31-44-151:/home/ubuntu# sudo su
root@ip-172-31-44-151:/home/ubuntu# mkdir /usr/local/nagios/etc/objects/monitorhosts
root@ip-172-31-44-151:/home/ubuntu# mkdir /usr/local/nagios/etc/objects/linuxhosts
```

1.sudo su 2.mkdir /usr/local/nagios/etc/objects/monitorhosts 3.mkdir

/usr/local/nagios/etc/objects/monitorhosts/linuxhosts Copy the sample localhost.cfg file to linuxhost folder 4.cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg

```
root@ip-172-31-44-151:/home/ubuntu# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
root@ip-172-31-44-151:/home/ubuntu# nano /usr/local/nagios/etc/objects/monitorhosts/linuxserver.cfg
```

```
GNU nano 6.2                               /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg

# 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           localhost
    alias               localhost
    address             127.0.0.1
}

[...]

^G Help      ^O Write Out   ^W Where Is   ^K Cut          ^T Execute   ^C Location   M-U Undo   M-A Set Mark
^X Exit      ^R Read File   ^\ Replace    ^U Paste        ^J Justify   ^I Go To Line M-E Redo   M-G Copy
i-03a3e79fc5ab0a056 (cutenagios_server)  X
```

```
GNU nano 6.2                               /usr/local/nagios/etc/objects/monitorhosts/linuxserver.cfg *

#####
# HOST GROUP DEFINITION
#
#####

# Define an optional hostgroup for Linux machines

define hostgroup {

    hostgroup_name      linux-servers[]       ; 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
}

[...]

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

```
GNU nano 6.2                               /usr/local/nagios/etc/nagios.cfg *

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

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


# OBJECT CACHE FILE
# This option determines where object definitions are cached when
# Nagios starts/restarts. The CGIs read object definitions from
Save modified buffer?
Y Yes
N No  C Cancel
```

```
root@ip-172-31-44-151:/home/ubuntu# nano /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
root@ip-172-31-44-151:/home/ubuntu# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.4.14
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2023-08-01
License: GPL

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

Running pre-flight check on configuration data...

Checking objects...
  Checked 8 services.
  Checked 1 hosts.
  Checked 1 host groups.
```

Hitesh Rohra 47 D15A

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

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
root@ip-172-31-44-151:/home/ubuntu# nano /usr/local/nagios/etc/nagios.cfg
```

Sudo systemctl status nagios

```
● nagios.service - Nagios Core 4.4.14
   Loaded: loaded (/lib/systemd/system/nagios.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2023-09-30 08:54:01 UTC; 20s ago
     Docs: https://www.nagios.org/documentation
 Process: 55285 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
Process: 55286 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 55287 (nagios)
   Tasks: 6 (limit: 1141)
  Memory: 5.3M
    CPU: 252ms
   CGroup: /system.slice/nagios.service
           ├─55287 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           ├─55288 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─55289 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─55290 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─55291 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─55292 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 30 08:54:01 ip-172-31-44-151 nagios[55287]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfully initialized
lines 1-19]
```

The screenshot shows the Nagios web interface at the URL 3.11.245.110/nagios/. The top navigation bar includes links for Home, Documentation, and Logout. The main menu on the left has sections for General, Current Status, Reports, and System.

Current Network Status: Last updated: Sat Sep 30 18:22:09 UTC 2023. It shows 0 Down, 0 Unreachable, and 0 Pending hosts.

Host Status Totals: 0 Down, 0 Unreachable, 0 Pending hosts.

Service Status Totals: 0 Warning, 0 Unknown, 3 Critical, 0 Pending services.

Host Status Details For All Host Groups: A table showing two hosts: 'Bioserver' and 'biohost'. Both are UP with a last check of 09-30-2023 18:17:06 and a duration of 0d 0h 5m 3s. Status information for both indicates PING OK - Packet loss = 0%, RTA = 0.62 ms.

Not secure | 13.233.247.135/nagios/

Host Status Totals

| Up | Down | Unreachable | Pending |
|----|------|-------------|---------|
| 2 | 0 | 0 | 0 |

Service Status Totals

| OK | Warning | Unknown | Critical | Pending |
|----|---------|---------|----------|---------|
| 13 | 0 | 0 | 3 | 0 |

Service Status Details For All Hosts

Limit Results: 100 ▾

| Host | Service | Status | Last Check | Duration | Attempt | Status Information |
|-------------|-----------------|--------------|---------------------|---------------------|----------------|---|
| linuxserver | Current Load | OK | 10-03-2023 23:34:51 | 3d 13h 47m 10s | 1/4 | OK - load average: 0.00, 0.02, 0.00 |
| | Current Users | OK | 10-03-2023 23:35:29 | 3d 13h 46m 32s | 1/4 | USERS OK - 2 users currently logged in |
| | HTTP | CRITICAL | 10-03-2023 23:36:00 | 0d 0h 12m 5s | 4/4 | CRITICAL - Socket timeout |
| | PING | OK | 10-03-2023 23:36:44 | 0d 0h 1m 27s | 1/4 | PING OK - Packet loss = 0%, RTA = 0.60 ms |
| | Root Partition | OK | 10-03-2023 23:37:21 | 3d 13h 44m 40s | 1/4 | DISK OK - free space: / 4859 MB (62.78% inode=88%): |
| | SSH | OK | 10-03-2023 23:37:59 | 0d 0h 0m 12s | 1/4 | SSH OK - OpenSSH_8.9p1 Ubuntu-Subuntu0.1 (protocol 2.0) |
| | Swap Usage | CRITICAL | 10-03-2023 23:38:16 | 3d 13h 43m 25s | 4/4 | SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size. |
| | Total Processes | OK | 10-03-2023 23:41:14 | 3d 13h 42m 47s | 1/4 | PROCS OK: 39 processes with STATE = R/SZDT |
| | localhost | Current Load | OK | 10-03-2023 23:35:10 | 3d 14h 43m 33s | 1/4 |
| | Current Users | OK | 10-03-2023 23:35:47 | 3d 14h 42m 55s | 1/4 | USERS OK - 2 users currently logged in |
| | HTTP | OK | 10-03-2023 23:36:25 | 3d 14h 42m 18s | 1/4 | HTTP OK: HTTP/1.1 200 OK - 10945 bytes in 0.000 second response time |
| | PING | OK | 10-03-2023 23:37:02 | 3d 14h 41m 40s | 1/4 | PING OK - Packet loss = 0%, RTA = 0.40 ms |
| | Root Partition | OK | 10-03-2023 23:37:40 | 3d 14h 41m 3s | 1/4 | DISK OK - free space: / 4859 MB (62.78% inode=88%): |
| | SSH | OK | 10-03-2023 23:38:17 | 3d 14h 40m 25s | 1/4 | SSH OK - OpenSSH_8.9p1 Ubuntu-Subuntu0.4 (protocol 2.0) |
| | Swap Usage | CRITICAL | 10-03-2023 23:38:55 | 3d 14h 36m 48s | 4/4 | SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size. |
| | Total Processes | OK | 10-03-2023 23:38:24 | 3d 14h 39m 10s | 1/4 | PROCS OK: 40 processes with STATE = R/SZDT |

Results 1 - 16 of 16 Matching Services

Experiment No 11

Advanced DevOps Exp-11

Hitesh Rohra

D15A 47

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

Theory:-

AWS Lambda

AWS Lambda is a serverless computing service provided by Amazon Web Services

(AWS). Users of AWS Lambda create functions, self-contained applications written in one

of the supported languages and runtimes, and upload them to AWS Lambda, which executes those functions in an efficient and flexible manner. The Lambda functions can

perform any kind of computing task, from serving web pages and processing streams of data to call APIs and integrate with other AWS services.

The concept of “serverless” computing refers to not needing to maintain your own servers to run these functions. AWS Lambda is a fully managed service that takes care of all the infrastructure for you.

Features of AWS Lambda

- AWS Lambda easily scales the infrastructure without any additional configuration. It reduces the operational work involved.
- It offers multiple options like AWS S3, CloudWatch, DynamoDB, API Gateway, Kinesis, CodeCommit, and many more to trigger an event.
- You don't need to invest upfront. You pay only for the memory used by the lambda function and minimal cost on the number of requests hence cost-efficient.
- AWS Lambda is secure. It uses AWS IAM to define all the roles and security policies.

- It offers fault tolerance for both services running the code and the function. You do not have to worry about the application down.

Packaging Functions

Lambda functions need to be packaged and sent to AWS. This is usually a process of compressing the function and all its dependencies and uploading it to an S3 bucket. And letting AWS know that you want to use this package when a specific event takes place.

To help us with this process we use the Serverless Stack Framework (SST). We'll go over this in detail later on in this guide.

Execution Model

The container (and the resources used by it) that runs our function is managed completely by AWS. It is brought up when an event takes place and is turned off if it is not being used. If additional requests are made while the original event is being served, a new container is brought up to serve a request. This means that if we are undergoing a usage spike, the cloud provider simply creates multiple instances of the container with our function to serve those requests.

This has some interesting implications. Firstly, our functions are effectively stateless. Secondly, each request (or event) is served by a single instance of a Lambda function. This means that you are not going to be handling concurrent requests in your code. AWS brings up a container whenever there is a new request. It does make some optimizations here. It will hang on to the container for a few minutes (5 - 15 mins depending on the load) so it can respond to subsequent requests without a cold start.

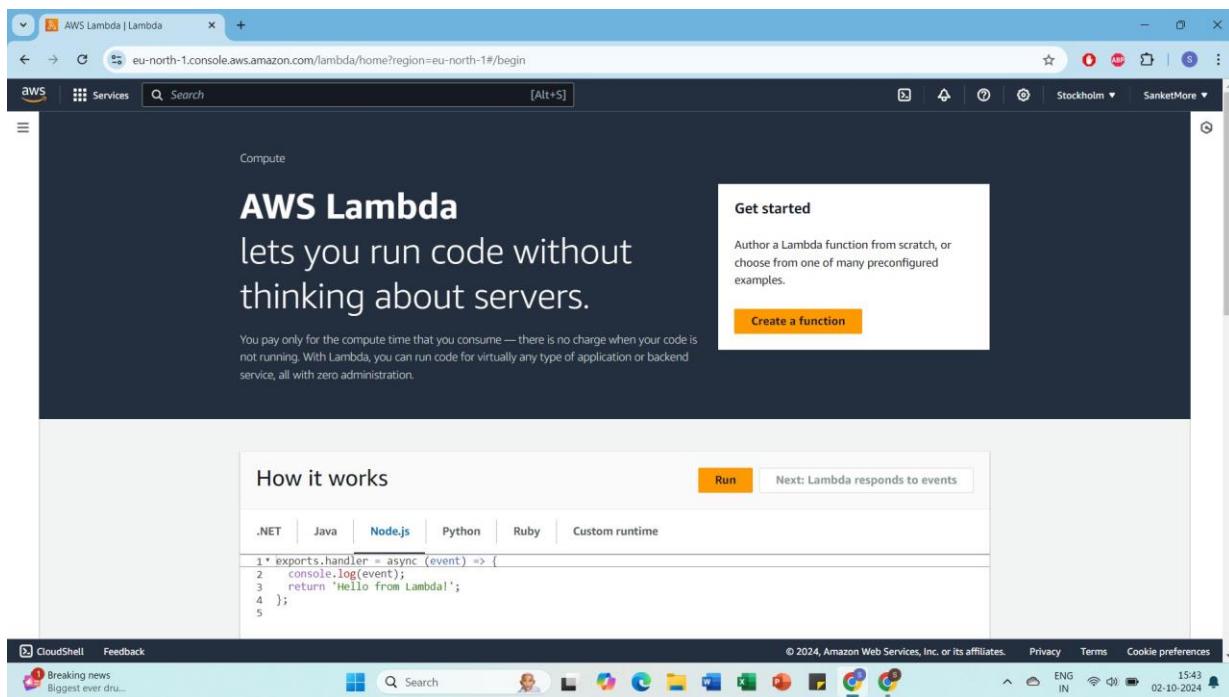
Stateless Functions

The above execution model makes Lambda functions effectively stateless. This means that every time your Lambda function is triggered by an event it is invoked in a completely new environment. You don't have access to the execution context of the previous event. However, due to the optimization noted above, the actual Lambda function is invoked only once

per container instantiation. Recall that our functions are run inside containers. So when a function is first invoked, all the code in our handler function gets executed and the handler function gets invoked. If the container is still available for subsequent requests, your function will get invoked and not the code around it.

Procedure:-

1. Open up the Lambda Console and click on the Create button. Be mindful of where you create your functions since Lambda is region-dependent.



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

Then, choose a runtime env for your function, under the dropdown, you can see all the options AWS supports, Python, Nodejs, .NET and Java being the most popular ones. After that, choose to create a new role with basic Lambda permissions if you don't have an existing one.

The screenshot shows the AWS Lambda 'Create function' wizard. The top navigation bar includes the AWS logo, 'Services' (with 'Lambda' selected), a search bar, and a keyboard shortcut '[Alt+S]'. The main title is 'Create function' with an 'Info' link. Below it, a sub-header says 'Choose one of the following options to create your function.' with 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.

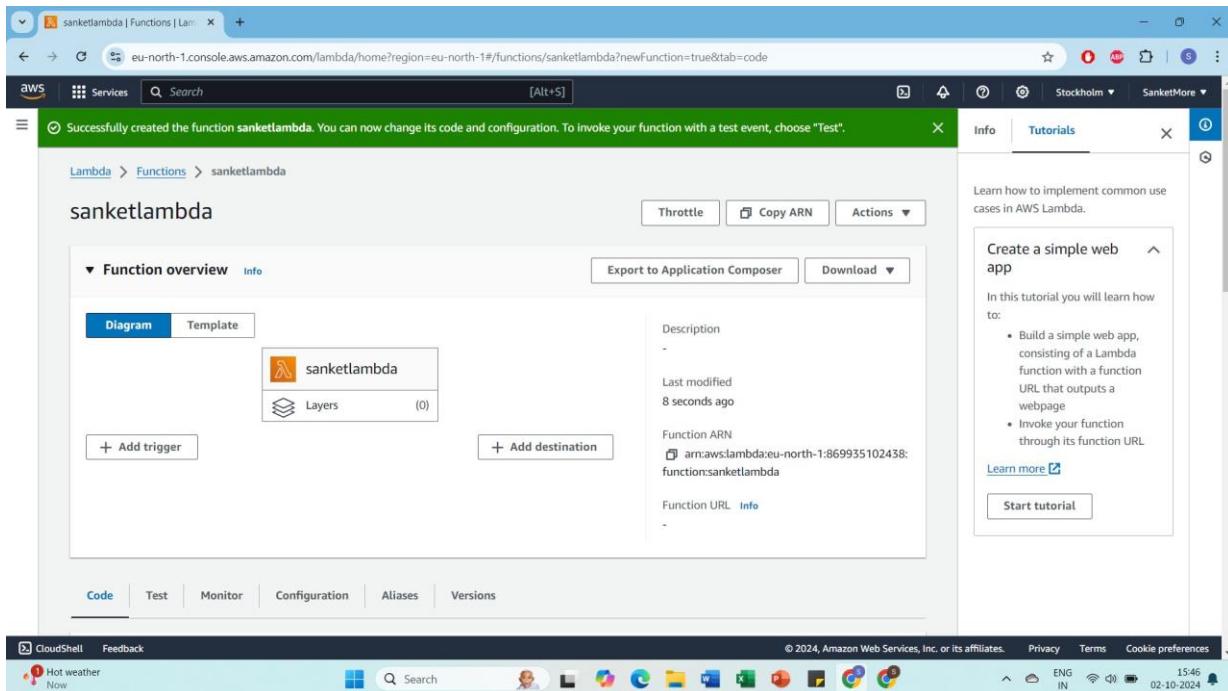
The 'Basic information' section contains fields for 'Function name' (set to 'sanketlambda'), 'Runtime' (set to 'Python 3.12'), and 'Architecture' (set to 'x86_64').

The 'Execution role' section is expanded, showing the 'Change default execution role' heading. It includes a sub-section for 'Execution role' with the note: 'Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console' with a link icon. Three options are listed:

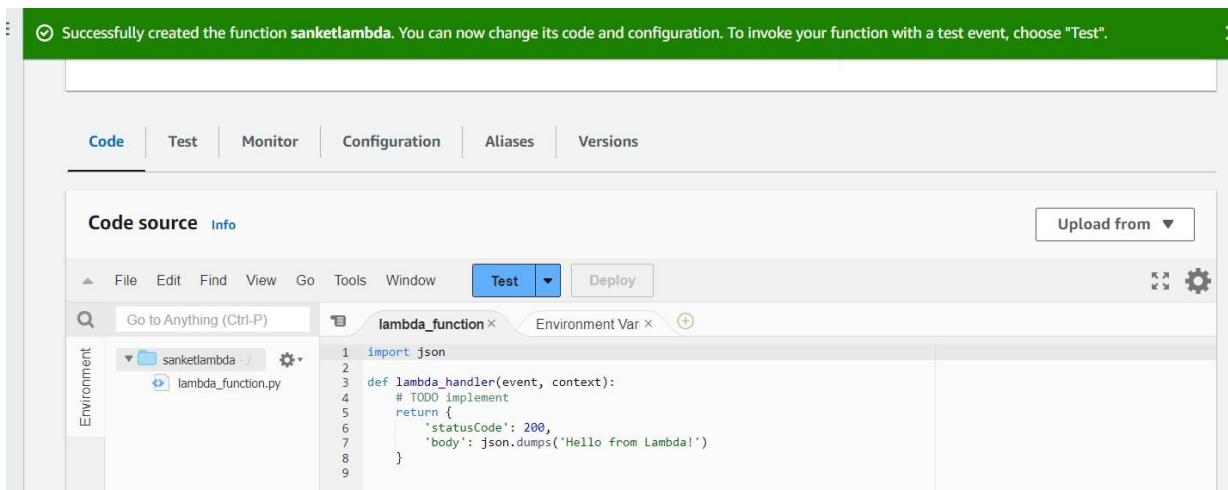
- Create a new role with basic Lambda permissions
- Use an existing role
- Create a new role from AWS policy templates

A note at the bottom of this section states: 'Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in this role.' A message at the bottom of the page says: 'Lambda will create an execution role named sanketlambda-role-aqbvjl1, with permission to upload logs to Amazon CloudWatch Logs.'

3. This process will take a while to finish and after that, you'll get a message that your function was successfully created.



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



5. To change the configuration, open up the Configuration tab and under General Configuration, choose Edit.

Here, you can enter a description and change Memory and Timeout. I've changed the Timeout period to 1 sec since that is sufficient for now.

The screenshot shows the AWS Lambda function configuration page for 'sanketlambda'. At the top, a green banner says 'Successfully created the function sanketlambda. You can now change its code and configuration. To invoke your function with a test event, choose "Test".' Below the banner, the navigation bar includes 'Code', 'Test', 'Monitor', 'Configuration' (which is highlighted in blue), 'Aliases', and 'Versions'. On the left, a sidebar menu lists 'General configuration', 'Triggers', 'Permissions', 'Destinations', and 'Function URL'. The main content area is titled 'General configuration' with an 'Edit' button. It displays the following settings:

| Description | Memory | Ephemeral storage |
|-------------|--------|-------------------|
| - | 128 MB | 512 MB |

Below this, there is a note: 'Set ephemeral storage (/tmp) to between 512 MB and 10240 MB.' Under 'SnapStart' (Info), it says: 'Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized. To evaluate whether your function code is resilient to snapshot operations, review the SnapStart compatibility considerations.' A dropdown menu is set to 'None'. Under 'Supported runtimes', it lists 'Java 11, Java 17, Java 21'. The 'Timeout' section shows '0 min 3 sec'. The 'Execution role' section has a radio button for 'Use an existing role' (which is selected) and another for 'Create a new role from AWS policy templates'. The 'Existing role' section notes: 'Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.'

The screenshot shows the AWS Lambda function configuration page for 'sanketlambda' after changes were made. The green banner at the top says 'Successfully updated the function sanketlambda.'. The rest of the interface is identical to the previous screenshot, showing the 'Configuration' tab selected, the 'Edit' button for the general configuration, and the updated timeout setting of '0 min 1 sec'.

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

The screenshot shows the 'Test event' configuration page. At the top right are 'Save' and 'Test' buttons. Below them is a note: 'To invoke your function without saving an event, configure the JSON event, then choose Test.' Under 'Test event action', 'Create new event' is selected. The 'Event name' field contains 'sanketevent'. The 'Event sharing settings' section has 'Private' selected. In the 'Template - optional' section, 'hello-world' is chosen from a dropdown. Below this, another 'Template - optional' section also has 'hello-world' selected. On the left, there's a 'Event JSON' section with a code editor containing the following JSON:

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

7. Now click on Test and you should be able to see the results.

The screenshot shows the Lambda function configuration interface. At the top, a green success message says 'The test event sanketevent was successfully saved.' Below it is a navigation bar with tabs: Code, Test, Monitor, Configuration, Aliases, and Versions. The 'Code' tab is active. The main area shows the 'Code source' tab with an 'Info' button and an 'Upload from' button. The 'Environment' sidebar lists 'sanketelambda' and 'lambda_function.py'. The 'Test' tab is selected in the toolbar. The 'Execution result' panel shows the following details:

| | | |
|-------------------|------------------------|---------------|
| Status: Succeeded | Max memory used: 32 MB | Time: 1.98 ms |
|-------------------|------------------------|---------------|

Test Event Name: (unsaved) test event

Response:

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

Function Logs:

```
START RequestId: 9308aa8a-986f-43d9-a1ce-2c182fa3cf55 Version: $LATEST
END RequestId: 9308aa8a-986f-43d9-a1ce-2c182fa3cf55
REPORT RequestId: 9308aa8a-986f-43d9-a1ce-2c182fa3cf55 Duration: 1.98 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 32 MB
```

Request ID: 9308aa8a-986f-43d9-a1ce-2c182fa3cf55

Experiment No 12

Advanced DevOps Exp-12

Hitesh Rohra

D15A 47

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.

Procedure:-

1. Create an S3 bucket of the same location as that of the Lambda function

The screenshot shows the 'Create bucket' wizard in the AWS S3 console. The 'General configuration' section is displayed. Under 'AWS Region', 'Europe (Stockholm) eu-north-1' is selected. Under 'Bucket type', 'General purpose' is selected. A note states: 'Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.' The 'Bucket name' field contains 'sanketbucket123'. A note below it says: 'Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming'.

The screenshot shows the 'sanketbucket123' bucket details page in the AWS S3 console. The 'Objects' tab is selected. The 'Actions' bar includes 'Upload' (highlighted in orange). Below the bar, a message says: 'No objects You don't have any objects in this bucket.' There is a 'Upload' button at the bottom of the list table.

The screenshot shows the 'Create function' wizard in the AWS Lambda console. At the top, there are four options for creating a 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.

Below these options, the 'Basic information' section is expanded. It includes fields for:

- Function name**: A text input field containing "sanketlambda123".
- Runtime**: A dropdown menu set to "Python 3.12".
- Architecture**: A dropdown menu set to "x86_64".

2. Add roles while creating the Lambda function and give permissions for accessing the S3 bucket

The screenshot shows the 'Change default execution role' step. It includes the following sections:

- Execution role**: A note stating "Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console." Below it are three radio button options:
 - Create a new role with basic Lambda permissions
 - Use an existing role
 - Create a new role from AWS policy templates**
- Role creation note**: A callout box with the text: "Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in this role."
- Role name**: A text input field containing "sanketrole".
- Policy templates - optional**: A dropdown menu currently empty, with a note below it: "Choose one or more policy templates." Below the dropdown is a box containing "Amazon S3 object read-only permissions" with an "X" icon and the letter "S3".

The screenshot shows the AWS Lambda console interface. At the top, a green banner indicates: "Successfully created the function sanketlambda123. You can now change its code and configuration. To invoke your function with a test event, choose 'Test'." Below this, the function name "sanketlambda123" is displayed. On the right, there are buttons for "Throttle", "Copy ARN", and "Actions". The "Function overview" section is open, showing a diagram of the function. The diagram consists of a single box labeled "sanketlambda123" with a value "3" next to it, and a "Layers" section below it indicating "(0)". There are buttons for "+ Add trigger" and "+ Add destination". To the right of the diagram, there is a "Description" field containing a dash, a "Last modified" field showing "3 seconds ago", and a "Function ARN" field with the value "arn:aws:lambda:eu-north-1:869935102438:function:sanketlambda123". A "Function URL" link is also present. At the bottom of the overview section, tabs for "Code", "Test", "Monitor", "Configuration", "Aliases", and "Versions" are visible.

3. After creating the Lambda function copy a code available on the internet which allows the Lambda function to access the S3 bucket contents.

The screenshot shows the AWS Lambda function editor for a function named "TTT Lambda Function". The code editor contains the following Python script:

```
import json
import urllib.parse
import boto3

print('Loading function')

s3 = boto3.client('s3')

def lambda_handler(event, context):
    #print("Received event: " + json.dumps(event, indent=2))

    # Get the object from the event and show its content type
    bucket = event['Records'][0]['s3']['bucket']['name']
    key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key'],
                                    encoding='utf-8')
    try:
        response = s3.get_object(Bucket=bucket, Key=key)
        print("CONTENT TYPE: " + response['ContentType'])
        return response['ContentType']
    except Exception as e:
        print(e)
        print('Error getting object {} from bucket {}. Make sure they exist and your bucket is in the same region as this function.'.format(key, bucket))
        raise e
```

Successfully updated the function sanketlambda123.

Code Test Monitor Configuration Aliases Versions

Code source Info Upload from ▾

File Edit Find View Go Tools Window Test Deploy

Environment Var λ Environment

lambda_function Environment Var +

```
1 import json
2 import urllib.parse
3 import boto3
4
5 print('Loading function')
6
7 s3 = boto3.client('s3')
8
9
10 def lambda_handler(event, context):
11     #print("Received event: " + json.dumps(event, indent=2))
12
13     # Get the object from the event and show its content type
14     bucket = event['Records'][0]['s3']['bucket']['name']
15     key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key'], encoding='utf-8')
16     try:
17         response = s3.get_object(Bucket=bucket, Key=key)
18         print("CONTENT TYPE: " + response['Content-Type'])
19         return response['Content-Type']
20     except Exception as e:
21         print(e)
22         print("Error getting object {} from bucket {}. Make sure they exist and your bucket is in the same region as the Lambda function.".format(key, bucket))
23         raise e
24
```

4. Add a trigger to the Lambda function so any changes in the S3 bucket will be first visible to the user.

aws Services Search [Alt+S]

Lambda > Add triggers

Add trigger

Trigger configuration Info

Select a source

S3

Batch/bulk data processing

S3 aws asynchronous storage

Cancel Add

aws | Services Search [Alt+S]

Lambda > Add triggers

Add trigger

Trigger configuration Info

S3 aws asynchronous storage

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

Bucket region: eu-north-1

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

All object create events

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.

Recursive invocation
If your function writes objects to an S3 bucket, ensure that you are using different S3 buckets for input and output. Writing to the same bucket increases the risk of creating a recursive invocation, which can result in increased Lambda usage and increased costs. [Learn more](#)

I acknowledge that using the same S3 bucket for both input and output is not recommended and that this configuration can cause recursive invocations, increased Lambda usage, and increased costs.

Lambda will add the necessary permissions for AWS S3 to invoke your Lambda function from this trigger. [Learn more](#) about the Lambda permissions model.

Lambda > Functions > sanketlambda123

sanketlambda123

Throttle Copy ARN Actions ▾

The trigger sanketbucket123 was successfully added to function sanketlambda123. The function is now receiving events from the trigger. X

Function overview Info Export to Application Composer Download ▾

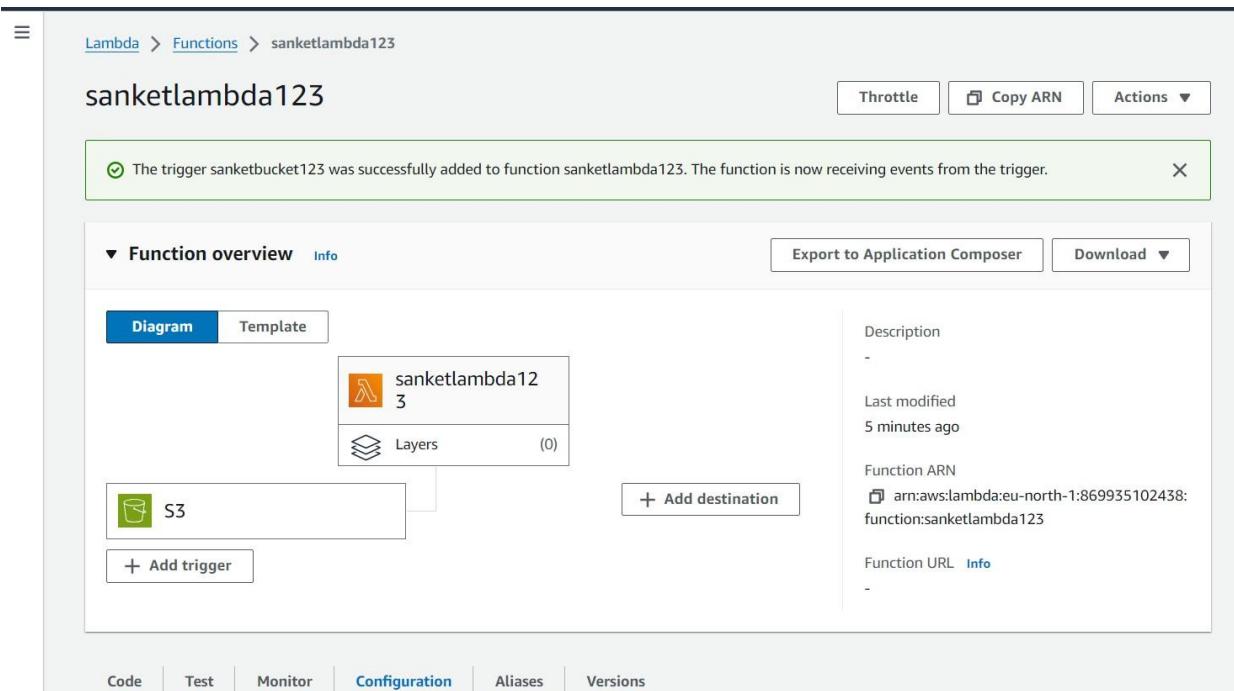
Diagram Template

sanketlambda12
3
Layers (0)

S3 + Add destination + Add trigger

Description -
Last modified 5 minutes ago
Function ARN arn:aws:lambda:eu-north-1:869935102438:function:sanketlambda123
Function URL Info -

Code Test Monitor Configuration Aliases Versions



5. In the event notification of the S3 bucket we can see that it has been connected to the Lambda function .

No data events to display.

Configure in CloudTrail

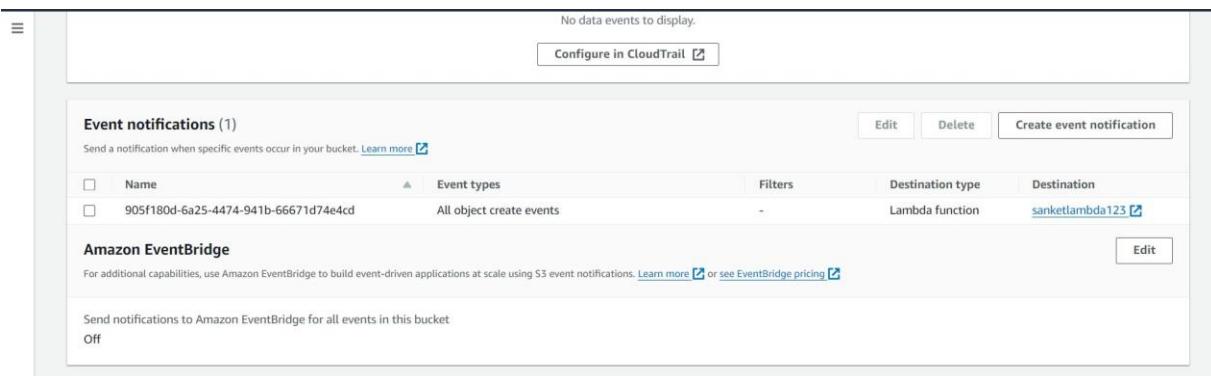
Event notifications (1)
Send a notification when specific events occur in your bucket. [Learn more](#)

Edit Delete Create event notification

| Name | Event types | Filters | Destination type | Destination |
|--------------------------------------|--------------------------|---------|------------------|-----------------|
| 905f180d-6a25-4474-941b-66671d74e4cd | All object create events | - | Lambda function | sanketlambda123 |

Amazon EventBridge
For additional capabilities, use Amazon EventBridge to build event-driven applications at scale using S3 event notifications. [Learn more](#) or see [EventBridge pricing](#)

Send notifications to Amazon EventBridge for all events in this bucket
Off



Managed policy AWSLambdaBasicExecutionRole-8a94e813-c025-4185-8c68-137a8a145ce0.statement.1

Resource-based policy document

```

1 Version: "2012-10-17",
2   "Id": "default",
3   "Statement": [
4     {
5       "Sid": "lambda-f873ff0-bb23-44ff-a3a8-08ebd4e381d2",
6       "Effect": "Allow",
7       "Principal": {
8         "Service": "s3.amazonaws.com"
9       },
10      "Action": "lambda:InvokeFunction",
11      "Resource": "arn:aws:lambda:eu-north-1:869935102438:function:sanketlambda123",
12      "Condition": {
13        "StringEquals": {
14          "AWS:SourceAccount": "869935102438"
15        },
16        "ArnLike": {
17          "AWS:SourceArn": "arn:aws:s3:::sanketbucket123"
18        }
19      }
20    }
21  ]
22 ]
23
  
```

1:1 JSON Spaces: 2

Close

6. Upload a photo to the S3 bucket

Amazon S3 > Buckets > [sanketbucket123](#) > Upload

Upload [Info](#)

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

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

| Files and folders (1 Total, 78.6 KB) | | |
|---|--------------------------|--------|
| Remove Add files Add folder | | |
| All files and folders in this table will be uploaded. | | |
| <input style="width: 20px; height: 15px; border: 1px solid #ccc; margin-right: 10px;" type="text"/> Find by name < 1 > | | |
| <input type="checkbox"/> | Name | Folder |
| <input type="checkbox"/> | sanket more photo 1.jpeg | - |
| Type image/jpeg | | |

Destination [Info](#)

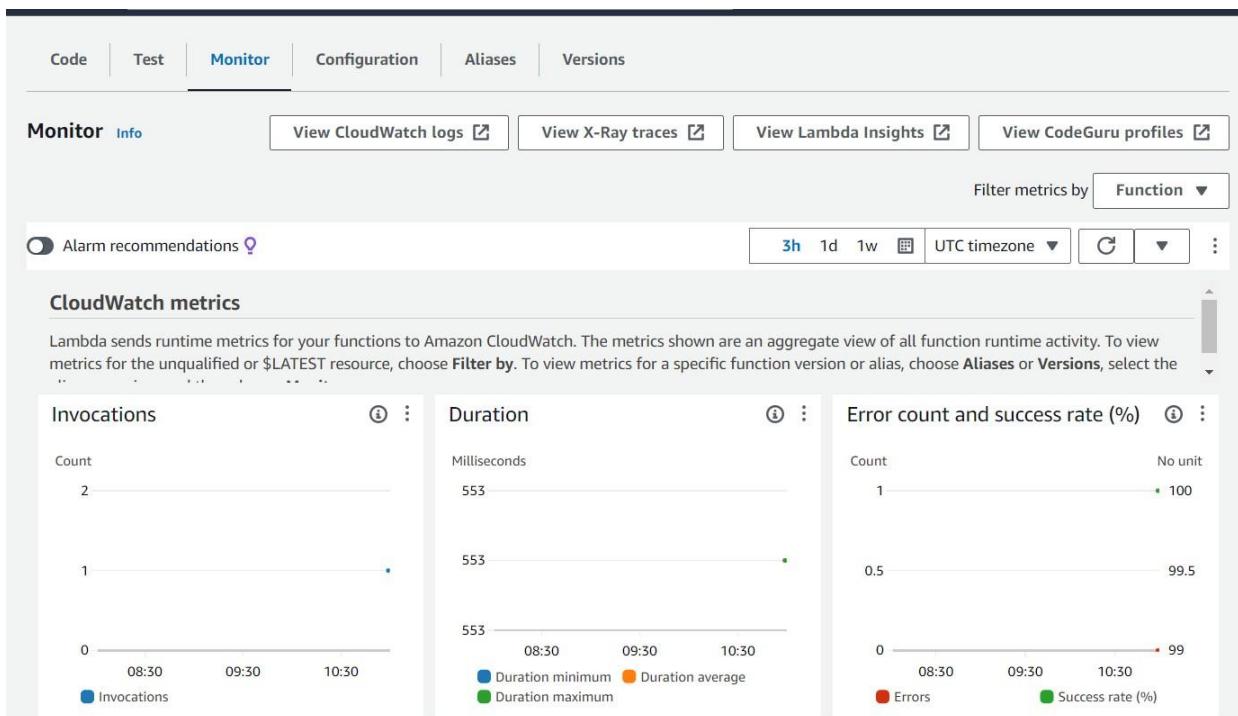
Destination

[s3://sanketbucket123](#)

The screenshot shows the AWS S3 console after a file has been uploaded. The top bar indicates "Upload succeeded". The summary section shows the destination as "s3://sanketbucket123" and the status as "Succeeded" with 1 file (78.6 KB). The "Failed" section shows 0 files (0 B). Below this, there are tabs for "Files and folders" and "Configuration", with "Files and folders" selected. The "Files and folders" section shows 1 total item, 78.6 KB, with a table listing the file name, type, size, status, and error.

| Name | Folder | Type | Size | Status | Error |
|-----------------|--------|------------|---------|-----------|-------|
| sanket more ... | - | image/jpeg | 78.6 KB | Succeeded | - |

7. Now run the function and in the cloud watch logs of AWS you can see the message printed and all the other details of the working of the Lambda function.



Screenshot of the AWS CloudWatch Log Groups interface.

CloudWatch Log group details for /aws/lambda/sanketlambda123:

| Log class | Info | Stored bytes | KMS key ID |
|---------------|---|---------------------------------|--------------------------------|
| Standard | - | - | - |
| ARN | arn:aws:log:eu-north-1:869935102438:log-group:/aws/lambda/sanketlambda123:* | Metric filters 0 | Anomaly detection Configure |
| Creation time | 3 minutes ago | Subscription filters 0 | Data protection |
| Retention | Never expire | Contributor Insights rules - | Sensitive data count - |

Log streams: [Log streams] [Tags] [Anomaly detection] [Metric filters] [Subscription filters] [Contributor Insights] [Data protection]

Screenshot of the AWS CloudWatch Log Events interface.

CloudWatch Log events for /aws/lambda/sanketlambda123 on 2024/10/02/[LATEST]:

| Timestamp | Message |
|--------------------------|--|
| 2024-10-02T10:59:36.409Z | INIT_START Runtime Version: python:3.12.v36 Runtime Version ARN: arn:aws:lambda:eu-north-1:runtime:188d9ca2e2714ff5637bd2bb... |
| 2024-10-02T10:59:36.801Z | Loading function |
| 2024-10-02T10:59:37.172Z | START RequestId: df929631-f73a-46eb-8a07-56f2f4a810c8 Version: \$LATEST |
| 2024-10-02T10:59:37.718Z | CONTENT TYPE: image/jpeg |
| 2024-10-02T10:59:37.725Z | END RequestId: df929631-f73a-46eb-8a07-56f2f4a810c8 |
| 2024-10-02T10:59:37.725Z | REPORT RequestId: df929631-f73a-46eb-8a07-56f2f4a810c8 Duration: 552.91 ms Billed Duration: 553 ms Memory Size: 128 MB Max Memory Used: 128 MB |

Filter events - press enter to search [Clear] [1m] [30m] [1h] [12h] [Custom] [UTC timezone] [Display] [More]

Assignment No 1

(04/05) Jr.

(d) Use S3 bucket and host video streaming

→ 1) log into AWS console

- Go to AWS Management console
- Enter your login credentials.

2) create an S3 bucket

- In the console search for S3 in the search bar and select S3 from the results.
- click create bucket
- Give your bucket a unique name.
- choose a region
- scroll down and uncheck Block all public access
- confirm by checking the acknowledgement box
- click create bucket.

3) upload your video file to S3

- click on your newly created bucket
- click on the upload button
- Add your video file from computer
- click upload to start the upload process.

(Q2) Discuss BMW and Hotstar case studies using AWS.

- BMW : A connected driving Experience ~~BMW~~ BMW is a leader in car technology and AWS has helped them a lot in creating connected cars
- connected cars features : BMW has developed features like remote diagnosis over the air updates and real-time traffic ~~over~~ the info AWS provides the necessary tools and infrastructure to make the features work.
 - Improving customer experience : They use AWS to enhance the customer internet with their cars for example ~~BMW~~ BMW has added voice activated assistance and personalized suggestions using services like Amazon Lex.
 - Better manufacturing : BMW uses AWS to improve its manufacturing By analyzing data from machines they can find problem early and increase efficiency
 - A streaming revolution : Hotstar is India's biggest video streaming platforms offering a huge range of movies, TV shows and sports AWS has been key of Hotstar success helping them.
 - Manage huge traffic : During big events like IPL many people watch at once AWS helps hotstar handle this stage in traffic without slowing down.

set permission for public Access.

- confirm the action by clicking make public again

Select the video URL

- After making the file public click on the video file
- you will see a URL for the video under Object URL

B.) Edit the Bucket policy

{

"version": "2012-10-17",

"statement": [

{

 "sid": "PublicReadGetObject"

 "Efect": "Allow",

 "Principal": "",

 >Action": "S3.GetObject"

 "Resource": "arn:aws:s3:::video-
 bucket /*"

}

]

}

- Quality streaming : AWS services like Amazon Kinesis ensures that users highly stream quality.
- Personalized Recommendation :- Hotstar uses AWS machine learning tools to suggest contents to users making their viewing experience more enjoyable

Key AWS services:-

- Compute : Amazon EC2
- Storage :- Amazon S3, Amazon EBS

(Q3) why Kubernetes and Advantages and disadvantages of Kubernetes. Explain How adidas uses Kubernetes

→ Kubernetes is popular because it simplifies the management of containerized applications. It automates tasks such as deployment, scaling and monitoring, making it easier for organizations to manage their applications in a cloud environment.

Advantages of Kubernetes:-

- 1) portability :- Application can be moved easily between different environment without major changes.

- 2) scalability :- Kubernetes can automatically scale apps up or down based on traffic and demand
- 3) Reliability :- It features self-healing capabilities meaning it can restart failed containers and balance workload.

Disadvantages of Kubernetes :-

- 1) complexity :- It can be complicated to setup and manage especially for those new to container
- 2) Steep learning curve :- Requires time and knowledge to fully understand and utilizes its features.
- 3) Resource Intensive :- It may require more computing resources than simpler solutions.

Adidas has adopted Kubernetes to enhance its IT infrastructure and improve its ability to respond to market needs.

(Q4) what are Nagios and Explain how Nagios are used in E - services?

→ Nagios is an open source monitoring tools that helps the organization keeps track of their IT infrastructure including servers networks and application. It provides a way to ensure that

- 2) scalability :- Kubernetes can automatically scale apps up or down based on traffic and demand
- 3) Reliability :- It features self-healing capabilities meaning it can restart failed containers and balance workload.

Disadvantages of Kubernetes :-

- 1) complexity :- It can be complicated to setup and manage especially for those new to container
- 2) Steep learning curve :- Requires time and knowledge to fully understand and utilizes its features.
- 3) Resource Intensive :- It may require more computing resources than simpler solutions.

Adidas has adopted Kubernetes to enhance its IT infrastructure and improve its ability to respond to market needs.

Q4) what are Nagios and Explain how Nagios are used in E - services?

→ Nagios is an open source monitoring tools that helps the organization keeps track of their IT infrastructure including servers networks and application. It provides a way to ensure that

Assignment No 2

Adv Devops Assignment 2

- Q Create a REST API with serverless framework
→ creating REST API with serverless framework is an efficient way to deploy serverless framework is an efficient applications that can scale automatically without managing server it serverless framework :- A powerful tool that deployment of services and serverless applications across various cloud providers such as AWS , And Google cloud
- ii) serverless architecture :- This design model allows developers to build applications without worrying about underlying infrastructure enabling focus on code & business logic
- iii) REST API :- Representational state transfer is architecture style for designing network applications

Slips for creating REST API for serverless framework

- i) Inshell serverless framework you start by installing serverless framework globally using node package manager (npm) This allow you to manage serverless apps directly from your terminal
- ii) creating a node is serverless project :- A directory is created for your project where you will initialize a serverless service (project) This service will house all your lambda function

configurations and cloud resources using the commands `serverless create`. You set up a template for AWS Node.js microservices that will eventually deploy to AWS Lambda.

iii) project structure :-

The project scaffold creates essential files like `handler.js` which contains code for Lambda function and `serverless.yml`.

iv) create a REST API Resource

In the `serverless.yml` file you define function that handles part request of HTTP

v) Deploy the service

With the `sls deploy` command `serverless` framework packages your applications, uploads necessary resources to AWS and sets up the infrastructure.

vi) Testing the API :- once deployed you can test REST API using tools like curl or postman by making POST request to generated API

vii) Storing Data in Database Dynamodb:- To store submitted candidate data you integrate AWS DynamoDB as database

viii) Adding more functionality:- Adding functional like install candidates get candidates by ID

ix) AWS IAM permissions

You need to ensure that serverless framework is given right permissions to interact with AWS resources like DynamoDB

x) Monitoring and maintenance

After deployment serverless framework provides services information like Deployed endpoints API keys, log streams

Q2)

case study for SonarQube

Creating your own profile in SonarQube for testing project Quality Use SonarQube to analyze your code Install Java IDE and analyze Java code

→ SonarQube is an open source platform used for continuous inspection of quality. It detects bug code smells and security vulnerabilities in project across programming languages

i) profile creation in SonarQube

Quality profiles in SonarQube are essential configuration that define rules applied during code analysis. Each project has a quality profile for every supported lang with default being Sonar way profile comes built

in for all languages custom profile can be created by copying or extending existing ones copying creates you can activate or deactivate rules prioritize certain rules and configure parameters to profile to specific projects.

- 2) using sonarQube to analyse github code
sonarQube is a cloud based counter part of sonarQube that triggers directly with github Bit Bucket , and github repositories To get started with sonar cloud via github signup product page and connect your github organization or personal account Once connect sonarQubecloud mirrors your git set up with each project corresponding to the github repos:- After setting up organization where each github repos a sonarcloud project Define new cloud to focus on recent changes and choose between automatic analysis or CI based analysis Automatic analysis happens directly in sonarcloud while CI based analysis integrates with your build process once the analysis results can be viewed in both sonar cloud and github including security import issue.

3) sonarlint in Java IDE:-

sonarlint is an IDE that performs on the fly code analysis as you write code. It helps developers in the developing environment such as intellic idea or Eclipse. To set it up install the sonarlint plugin, configure the connection with sonarqube or sonarcloud and select the project profile to analyse Java code in code quality, promoting clean & maintainable code from beginning.

4) Analyzing Python projects with sonarqube

Sonarqube supports python test coverage reporting but it requires third party tool like coverage part to enable and adjust your build process so that coverage tools runs before sonar scanner and ensures report file is saved in diff path. For setup you can use .travis.yml and coverage.py to configure and run test. In your .travis.yml include configuration for pytest and coverage to generate report in XML format. The build process can also be automated using GitHub Actions which install dependencies run test and involves sonarqube scan. Ensure report in XML format and place where scanner can access it.

5) Analyzing Node.js projects with SonarQube

for node.js project sonarqube can analyze Java script and typescript code. similar to the python setup you can configure sonarqube to analyze node.js project by installing the appropriate plugin and using sonar scanner to scan the projects sonarqube will check the code against industry standard rules and best practices flagging issues related to security vulnerabilities bugs and performances optimization.

Q3) At a large organization your centralized operation team may get many repeatable infrastructure request you can use form to build a self services infrastructure mode that lets product team manage their own infrastructure independently you can create and use Terraform modules that codify the standards for deploying & managing services in your organization allowing teams to efficiently deploy services

→ Implementing a self service infrastructure model using Terraform can transform how large organization manage their infrastructure independently organization can enhance efficiency

reduce benefits and ensure compliance with established needs.

The need for self service infrastructure:- In large organization centralized operators teams often face an overwhelming number repetitive request. This can lead to delay in service delivery and frustration among product teams who need to move quickly. A self service model allows team to provision and manage their infrastructure without relying on the operations team for every request.

- Benefits of using Terraform:-

1. Modularity & Reusability
2. Standardization
3. Increased Efficiency
4. Integration with ticketing systems

- Implementation steps

- i) Identify Infrastructure components.

- Begin by Identifying which components your infrastructure can be modularized like VPCs, security groups, load balanced.

2) Develop Terraform modules

- create reusable modules that define the desired configuration & resources
- Ensure each module includes input variables for customization and outputs for integration with other modules.

3) Establish Governance and Best practices

- Define guidelines for module usage version and documentation to ensure clarity and maintainability
- Encourage teams to contribute to module development and share improvements.

4) Testing and validation

- Implement a testing removable to validate functionality before development.
- Best practices for module management
- utilize the terraform registry
- leverage existing community modules from the Terraform Registry to avoid reinventing solutions and ensure adherence to best practices.