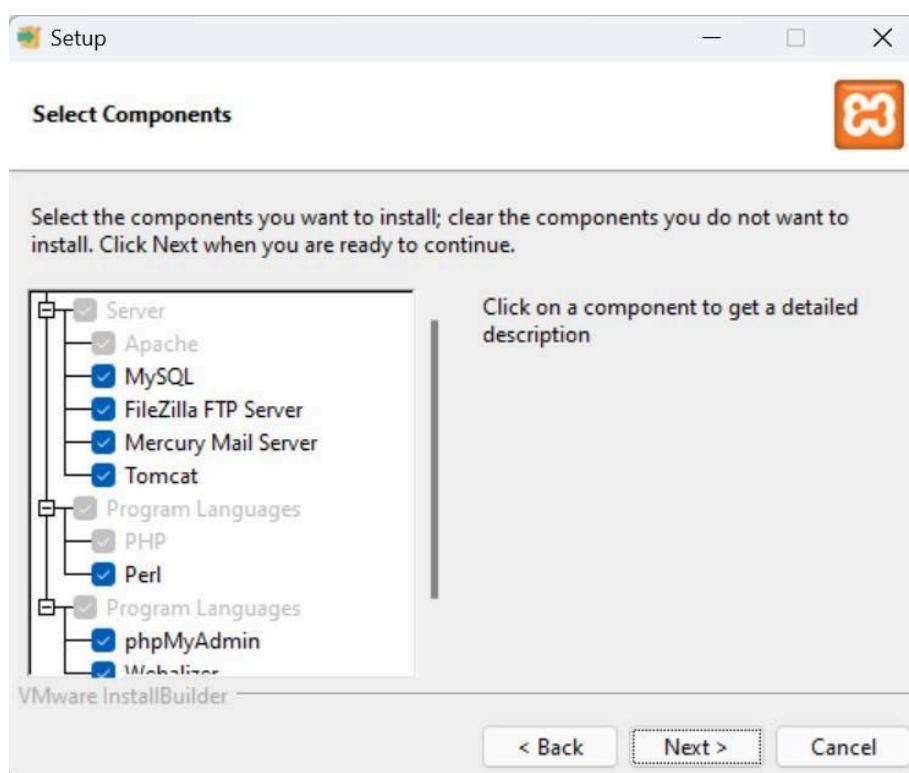
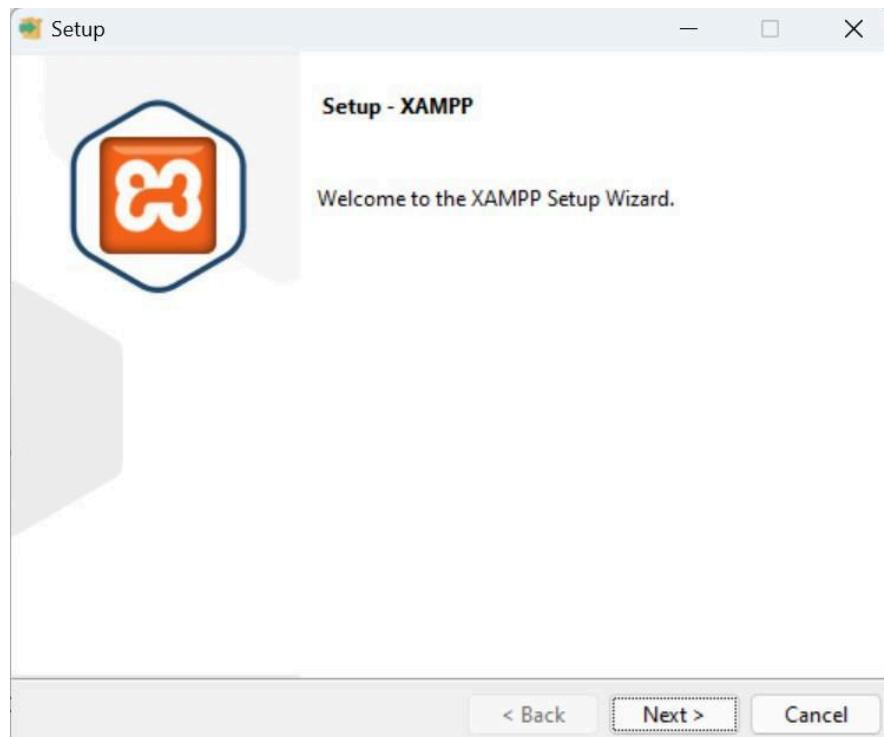
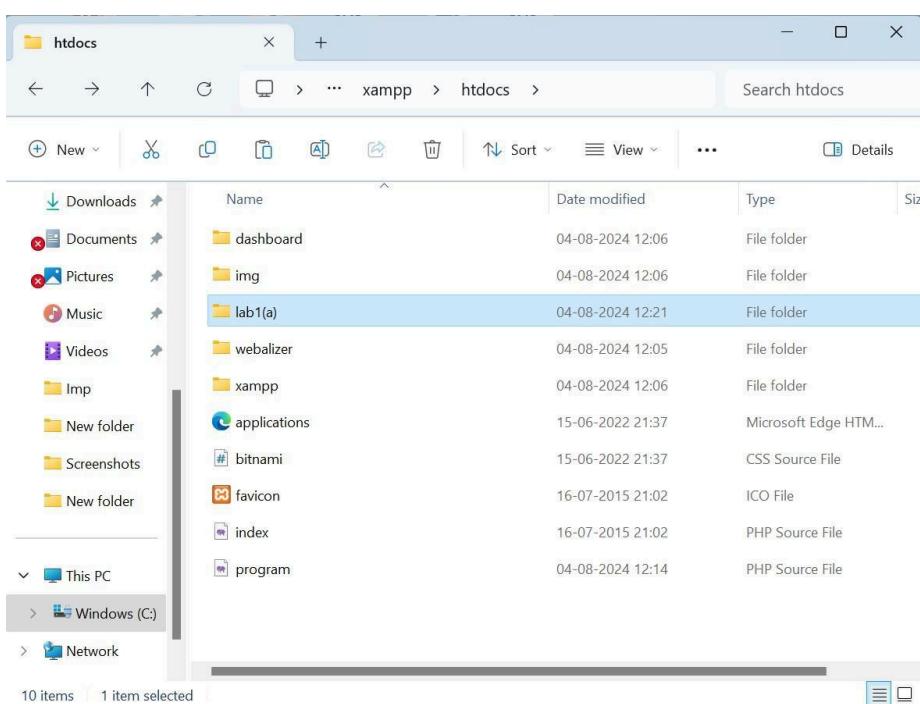
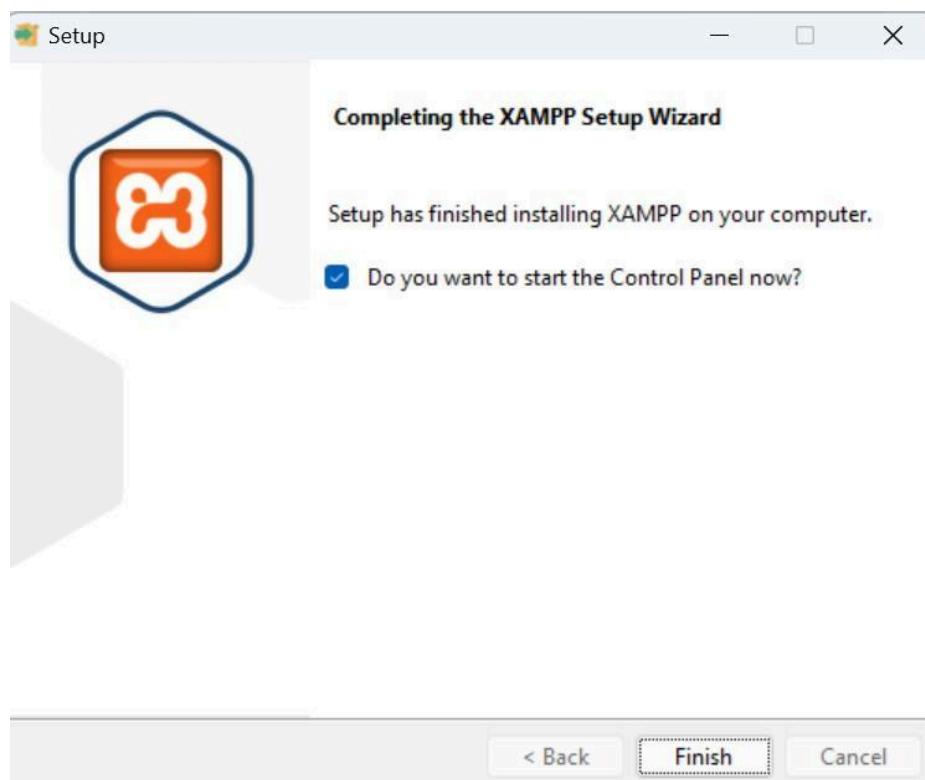
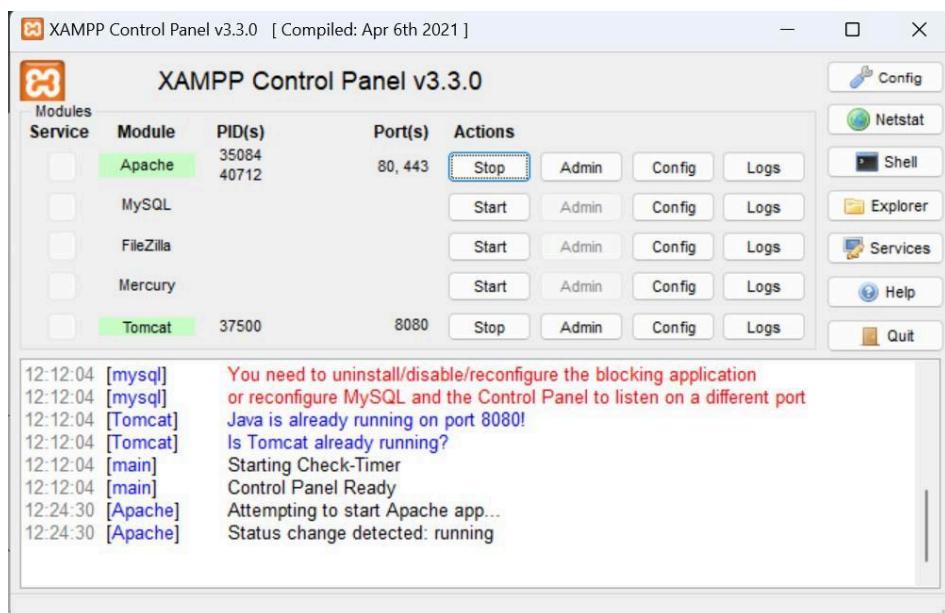
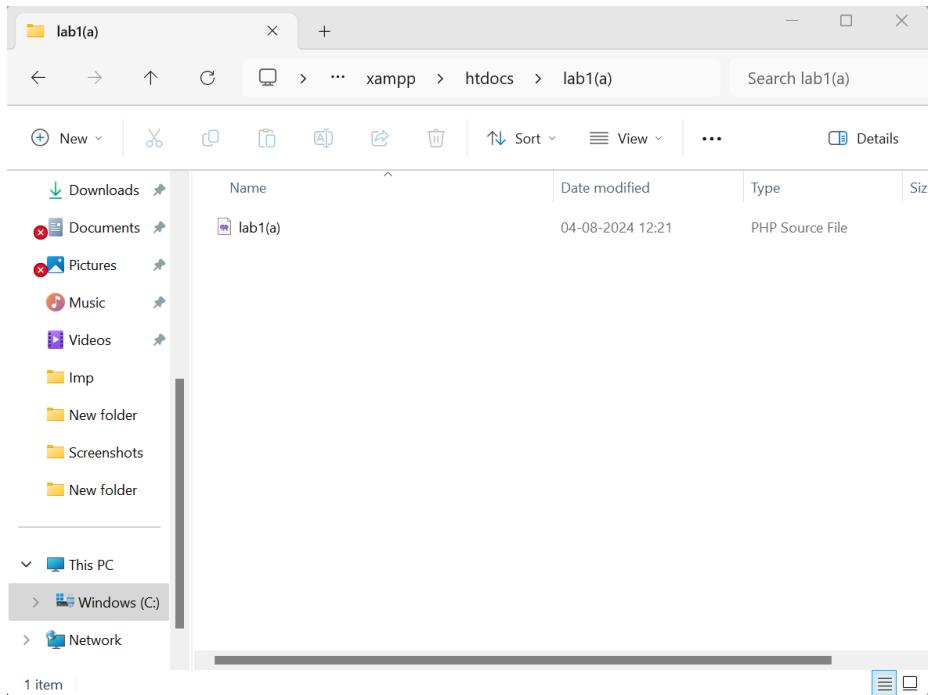


Aim: To understand the benefits of Cloud Infrastructure and Setup AWS Cloud9 IDE, Launch AWS Cloud9 IDE and Perform Collaboration Demonstration.







The screenshot shows a web browser window with the URL `localhost/lab1(a)/` in the address bar. The page title is "Index of /lab1(a)". Below the title is a table with the following columns: Name, Last modified, Size, and Description. The table contains two rows:

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
Parent Directory	-	-	
lab1(a).php	2024-08-04 12:21	31	

At the bottom of the page, the server information is displayed: `Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.0.30 Server at localhost Port 80`.

The screenshot shows a web browser window with the URL `localhost/lab1(a)/lab1(a).php` in the address bar. The page content is a single line of text: "Hello World".

The screenshot shows the Amazon S3 service landing page. The top navigation bar includes the AWS logo, Services, a search bar, and account information: `voclabs/user3385469=KULKARNI_ANISH_AMIT @ 6635-3922-7562`. The main content area features the heading "Amazon S3" and the subtext "Store and retrieve any amount of data from anywhere". A call-to-action button labeled "Create a bucket" is visible. At the bottom, there is a "Pricing" link.

Amazon S3 > Buckets > Create bucket

Create bucket Info

Buckets are containers for data stored in S3.

General configuration

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

Bucket type Info

- General purpose

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

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

Bucket name Info

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

Copy settings from existing bucket - *optional*
Only the bucket settings in the following configuration are copied.

[Choose bucket](#)
Format: s3://bucket/prefix

Object Ownership Info

Edit static website hosting Info

Static website hosting

Use this bucket to host a website or redirect requests. [Learn more](#)

Static website hosting

- Disable
- Enable

Hosting type

- Host a static website

Use the bucket endpoint as the web address. [Learn more](#)
- Redirect requests for an object

Redirect requests to another bucket or domain. [Learn more](#)

ⓘ For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see [Using Amazon S3 Block Public Access](#)

Index document
Specify the home or default page of the website.

Error document - *optional*
This is returned when an error occurs.

Static website hosting

Use this bucket to host a website or redirect requests. [Learn more](#)

Static website hosting
Enabled

Hosting type
Bucket hosting

Bucket website endpoint
When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket. [Learn more](#)

<http://www.lab1a.com.s3-website-us-east-1.amazonaws.com>

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 (2 Total, 2.7 KB)			
All files and folders in this table will be uploaded.			
<input type="button" value="Remove"/> <input type="button" value="Add files"/> <input type="button" value="Add folder"/>			
<input type="text" value="Find by name"/> < 1 >			
Name	Folder	Type	
error.html	-	text/html	
lab1.html	-	text/html	

Upload succeeded
View details below.

Upload: status [Close](#)

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

Summary		
Destination s3://www.lab1a.com	Succeeded 2 files, 2.7 KB (100.00%)	Failed 0 files, 0 B (0%)
Files and folders	Configuration	

Files and folders (2 Total, 2.7 KB)						
<input type="text" value="Find by name"/> < 1 >						
Name	Folder	Type	Size	Status	Error	
error.html	-	text/html	150.0 B	Succeeded	-	
lab1.html	-	text/html	2.6 KB	Succeeded	-	

The screenshot shows the 'Bucket policy' section of an AWS S3 bucket's properties. It displays a JSON-based policy document:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::www.lab1a.com/*"
    }
  ]
}
```

At the top right, there are 'Edit' and 'Delete' buttons. A 'Copy' button is located at the top right of the policy text area.

The screenshot shows a web browser window with the URL www.lab1a.com.s3-website-us-east-1.amazonaws.com. The page content is:

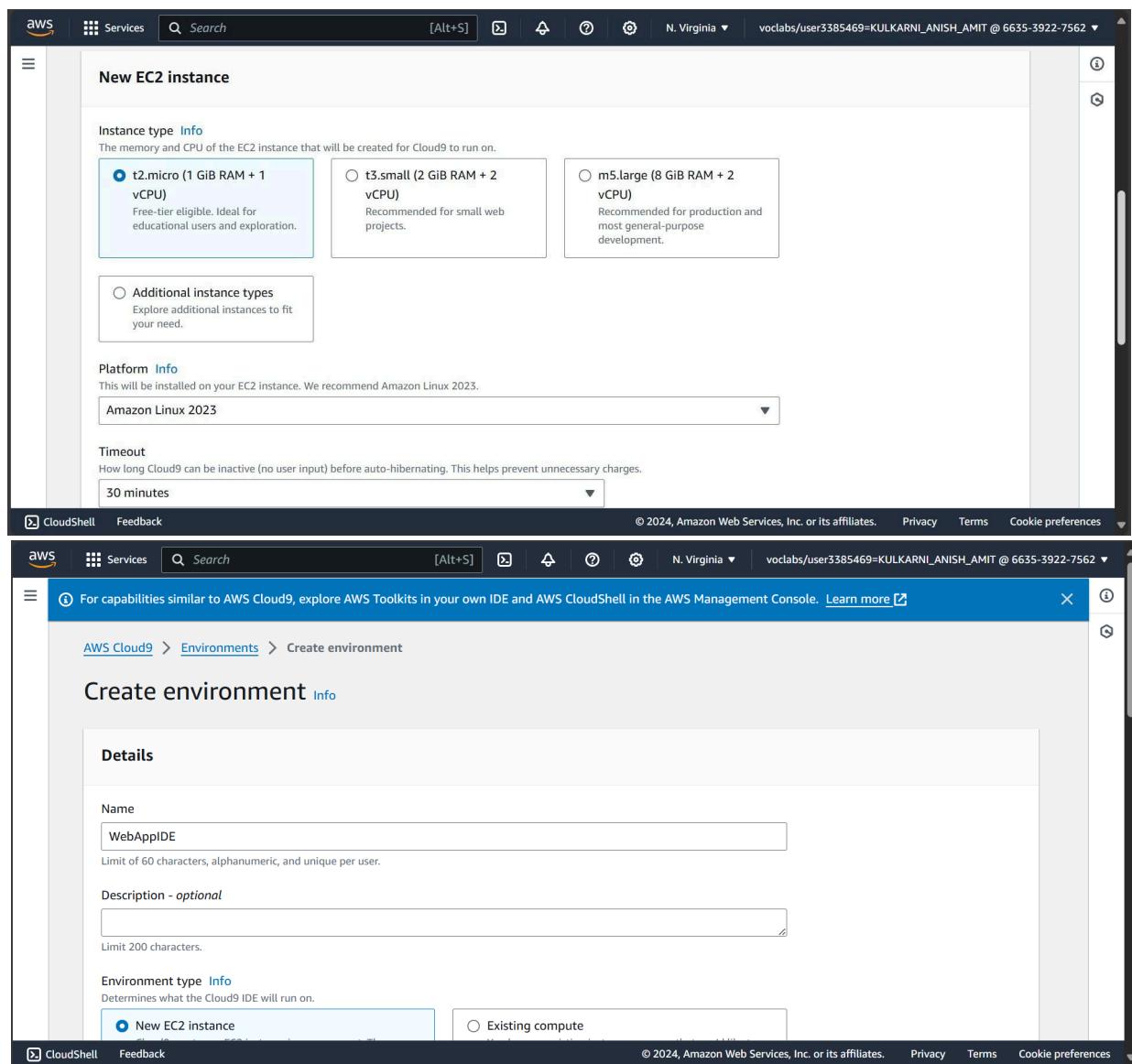
Hello World!

This is a lab1(a) website

The screenshot shows a web browser window with the URL www.lab1a.com.s3-website-us-east-1.amazonaws.com/lab2.html. The page content is:

Error Message

Aim: To understand the benefits of Cloud Infrastructure and Setup AWS Cloud9 IDE, Launch AWS Cloud9 IDE and Perform Collaboration Demonstration.



The screenshot shows two sequential steps in the AWS Cloud9 'Create environment' wizard:

Step 1: New EC2 instance

- Instance type Info:** The memory and CPU of the EC2 instance that will be created for Cloud9 to run on.
 - t2.micro (1 GiB RAM + 1 vCPU)
Free-tier eligible. Ideal for educational users and exploration.
 - t3.small (2 GiB RAM + 2 vCPU)
Recommended for small web projects.
 - m5.large (8 GiB RAM + 2 vCPU)
Recommended for production and most general-purpose development.
- Additional instance types:** Explore additional instances to fit your need.
- Platform Info:** This will be installed on your EC2 instance. We recommend Amazon Linux 2023.
 - Amazon Linux 2023
- Timeout:** How long Cloud9 can be inactive (no user input) before auto-hibernating. This helps prevent unnecessary charges.
 - 30 minutes

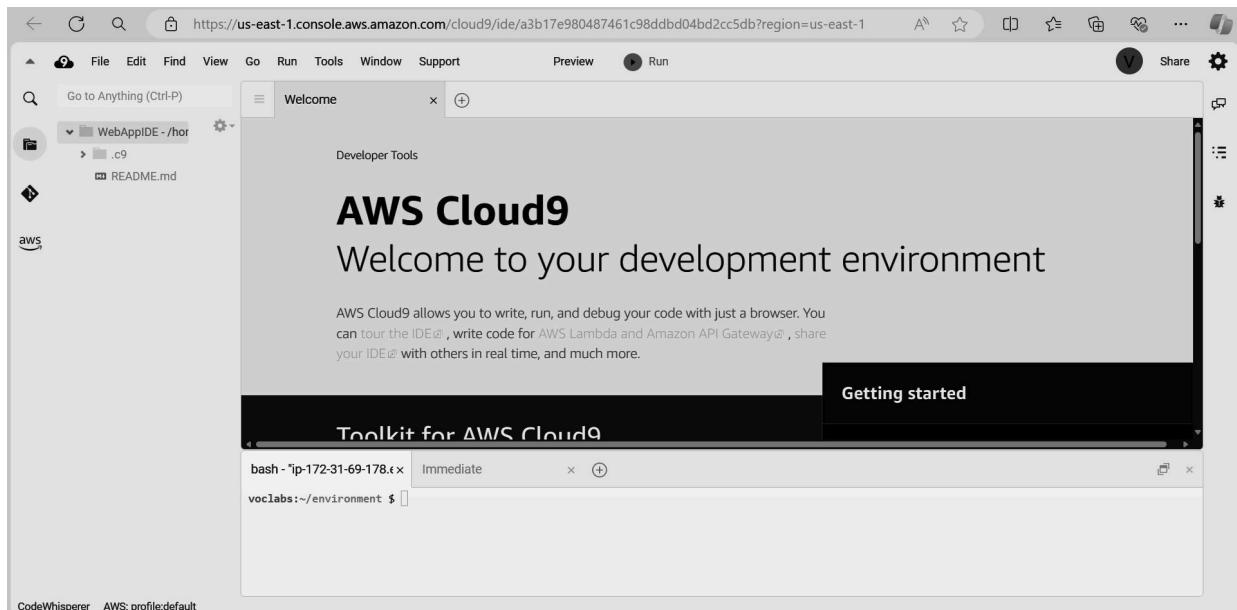
Step 2: Create environment

Details

- Name:** WebAppIDE
Limit of 60 characters, alphanumeric, and unique per user.
- Description - optional:** (Empty field)
- Environment type Info:** Determines what the Cloud9 IDE will run on.
 - New EC2 instance
 - Existing compute

The screenshot shows the 'Network settings' configuration page for an AWS Cloud9 environment. At the top, there's a 'Connection' section with two options: 'AWS Systems Manager (SSM)' and 'Secure Shell (SSH)'. 'Secure Shell (SSH)' is selected, indicated by a blue circle. Below this, there's a 'VPC settings' link. A 'Tags - optional' section follows, explaining what tags are and how they can be used. At the bottom, a note informs the user that IAM resources will be created in their account, specifically a service-linked role named 'AWSServiceRoleForAWSCloud9'. The page includes standard AWS navigation elements like CloudShell and Feedback links at the bottom.

The screenshot shows the 'Environments' list page in the AWS Cloud9 interface. A green success message at the top states 'Successfully created WebAppIDE. To get the most out of your environment, see Best practices for using AWS Cloud9'. Below this, a blue info message encourages exploring AWS Toolkits in your own IDE and AWS CloudShell in the AWS Management Console. The main area displays a table of environments. The first environment listed is 'WebAppIDE', which is currently 'Open'. It is an 'EC2 instance' type, connected via 'Secure Shell (SSH)', and owned by the user. The table has columns for Name, Cloud9 IDE, Environment type, Connection, Permission, and Owner ARN. The 'Create environment' button is visible at the top right of the table area. The page also includes standard AWS navigation elements like CloudShell and Feedback links at the bottom.



Screenshot of the AWS IAM 'Create user' wizard - Step 2: Set permissions.

The left sidebar shows steps: Step 1 (Specify user details), Step 2 (Set permissions), and Step 3 (Review and create). The main area is titled 'Set permissions' with the sub-section 'Permissions options'. It contains three radio button options:

- Add user to group: Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
- Copy permissions: Copy all group memberships, attached managed policies, and inline policies from an existing user.
- Attach policies directly: Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

A callout box 'Get started with groups' provides instructions: Create a group and select policies to attach to the group. We recommend using groups to manage user permissions by job function, AWS service access, or custom permissions. A 'Create group' button is shown.

Create user group

Create a user group and select policies to attach to the group. We recommend using groups to manage user permissions by job function, AWS service access, or custom permissions. [Learn more](#)

User group name

Enter a meaningful name to identify this group.

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

Permissions policies (947)


[Create policy](#)

Filter by Type

All ty... ▾

< 1 2 3 4 5 6 7 ... 48 > ⚙

<input type="checkbox"/>	Policy name	Type	Use...	Description
<input type="checkbox"/>	AdministratorAccess	AWS managed ...	None	Provides full access to AWS services
<input type="checkbox"/>	AdministratorAcc...	AWS managed	None	Grants account administrative perm

[Cancel](#)

[Create user group](#)

REVIEW AND CREATE

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

Step 2
Set permissions

Step 3
Review and create

User name	Console password type	Require password reset
user1	None	No

Permissions summary

Name	Type	Used as
No resources		

Tags - optional

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

Add new tag
You can add up to 50 more tags.

Cancel Previous Create user



IAM > Users

Users (1) Info

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

User name	Path	Group	Last activity	MFA
user1	/	0	-	-

User groups (1) Info

A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.

Group name	Users	Permissions	Creation time
group1	⚠ 0	Pending	5 minutes ago

Identity and Access Management (IAM)

[Search IAM](#)

- Dashboard
- Access management
 - User groups
 - Users
 - Roles
 - Policies
 - Identity providers
 - Account settings
- Access reports

group1 [Info](#) [Delete](#)

Summary [Edit](#)

User group name group1	Creation time August 04, 2024, 16:59 (UTC+05:30)	ARN arn:aws:iam::010928206130:group/group1
---------------------------	---	---

Users [Permissions](#) Access Advisor

Permissions policies (0) [Info](#) [Simulate](#) [Remove](#) [Add permissions](#)

You can attach up to 10 managed policies.

Other permission policies (1/945) [Filter](#)

You can attach up to 10 managed policies to this user group. All of the users in this group inherit the attached permissions.

Filter by Type [X](#) All types 4 matches [<](#) [1](#) [>](#) [@](#)

<input type="checkbox"/>	Policy name	Type	Used as	Description
<input type="checkbox"/>	AWSCloud9Administ...	AWS managed	None	Provides administrator access to AWS ...
<input checked="" type="checkbox"/>	AWSCloud9Environ...	AWS managed	None	Provides the ability to be invited into A...
<input type="checkbox"/>	AWSCloud9SSMInsta...	AWS managed	None	This policy will be used to attach a rol...
<input type="checkbox"/>	AWSCloud9User	AWS managed	None	Provides permission to create AWS Clo...

Exp 2

Aim: To Build Your Application using AWS CodeBuild and Deploy on S3 / SEBS using AWS CodePipeline, deploy Sample Application on EC2 instance using AWS CodeDeploy.

Configure environment Info

Environment tier Info

Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.

Web server environment

Run a website, web application, or web API that serves HTTP requests. [Learn more](#)

Worker environment

Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. [Learn more](#)

Application information Info

Application name

Maximum length of 100 characters.

► Application tags (optional)

Service access

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

Service role

Create and use new service role

Use an existing service role

Existing service roles

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



EC2 key pair

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



EC2 instance profile

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



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-0da5a3d9b481e2d7b | (172.31.0.0/16)



[Create custom VPC](#)

Review Info

Step 1: Configure environment

[Edit](#)

Environment information

Environment tier

Web server environment

Application name

Application1

Environment name

Application1-env

Application code

Sample application

Platform

arn:aws:elasticbeanstalk:us-east-1::platform/PHP 8.3
running on 64bit Amazon Linux 2023/4.3.1

Step 2: Configure service access

[Edit](#)

Service access Info

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

Service role

arn:aws:iam::011528263337:role/service-role/AWSCloud9SSMAccessRole

EC2 instance profile

AWSCloud9SSMInstanceProfile

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

⌚ Environment successfully launched.

Elastic Beanstalk > Environments > Application1-env

Application1-env Info

Actions ▼ **Upload and deploy**

Environment overview	Platform <small>Change version</small>
Health ⚠ Warning	Platform PHP 8.3 running on 64bit Amazon Linux 2023/4.3.1
Domain Application1-env.eba-gepbcwby.us-east-1.elasticbeanstalk.com <small>Copy</small>	Running version -
Application name Application1	Platform state _supported_

Choose pipeline settings Info

Step 1 of 5

Pipeline settings

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

pipeline1
No more than 100 characters

Pipeline type

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

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

Superseded
A more recent execution can overtake an older one. This is the default.
 Queued (Pipeline type V2 required)

Service role

New service role
Create a service role in your account

Existing service role
Choose an existing service role from your account

Role name

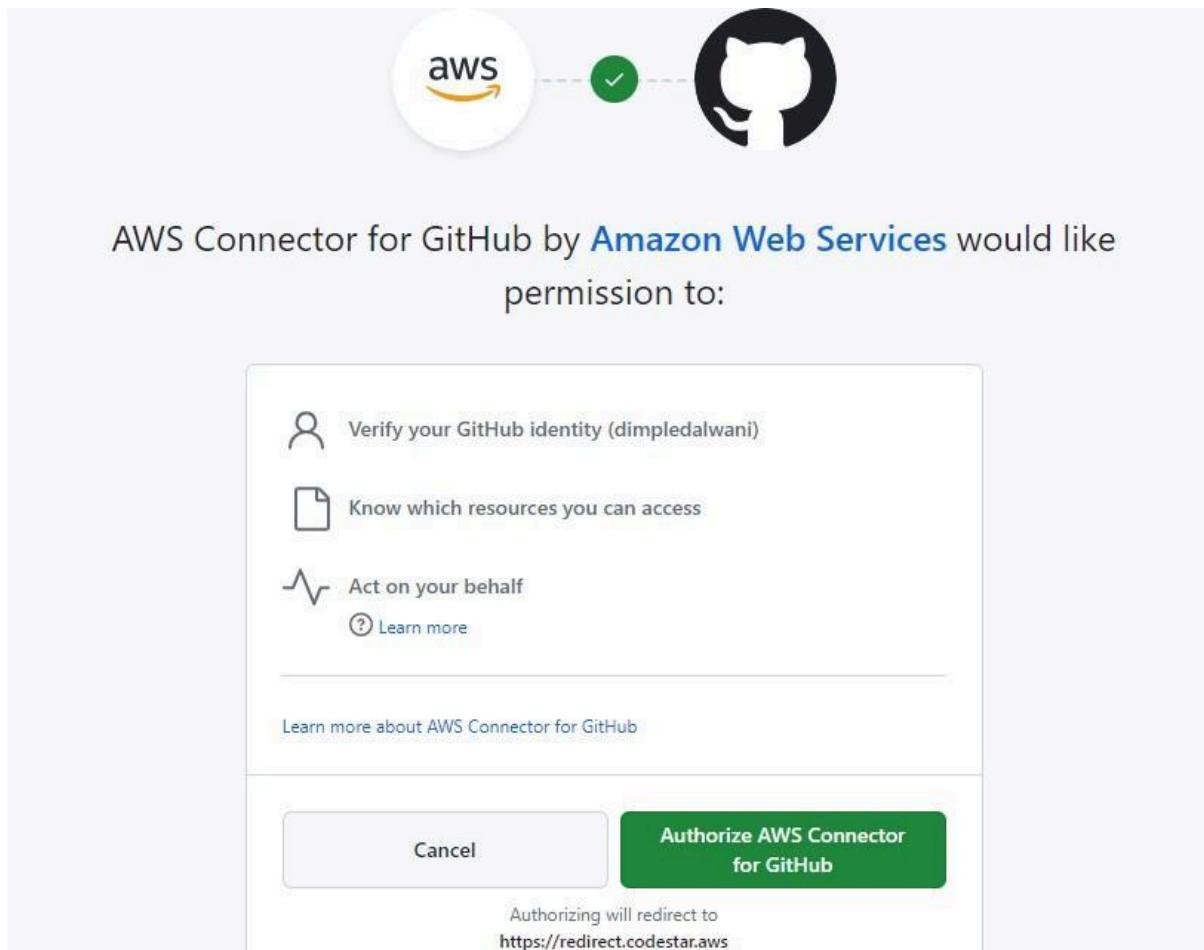
AWSCodePipelineServiceRole-us-east-1-pipeline1
Type your service role name
 Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

Variables

You can add variables at the pipeline level. You can choose to assign the value when you start the pipeline. Choosing this option requires pipeline type V2. [Learn more](#)

No variables defined at the pipeline level in this pipeline.

[Add variable](#)
You can add up to 50 variables.



Install AWS Connector for GitHub

Install on your personal account dimpledalwani 

for these repositories:

All repositories

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

Only select repositories

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

with these permissions:

- Read access to issues and metadata**
- Read and write access to administration, code, commit statuses, pull requests, and repository hooks**

Install

[Cancel](#)

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

arn:aws:codeconnections:us-east-1:011528263337:connection/be7ab482-33 X | or | [Connect to GitHub](#)

Ready to connect
Your GitHub connection is ready for use.

Repository name
Choose a repository in your GitHub account.
 X

You can type or paste the group path to any project that the provided credentials can access. Use the format 'group/subgroup/project'.

Default branch
Default branch will be used only when pipeline execution starts from a different source or manually started.
 X

master
Choose the output artifact format.

CodePipeline default
AWS CodePipeline uses the default zip format for artifacts in the pipeline. Does not include Git metadata about the repository.

Full clone
AWS CodePipeline passes metadata about the repository that allows subsequent actions to do a full Git clone. Only supported for AWS CodeBuild actions.

Skip build stage X

Your pipeline will not include a build stage. Are you sure you want to skip this stage?

Cancel Skip

Deploy

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

AWS Elastic Beanstalk ▾

Region
US East (N. Virginia) ▾

Input artifacts
Choose an input artifact for this action. [Learn more](#)

SourceArtifact ▾
No more than 100 characters

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

Application1 X

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

Application1-env X

⌚ Success
Congratulations! The pipeline pipeline1 has been created.

Create a notification rule for this pipeline X

Developer Tools > [CodePipeline](#) > [Pipelines](#) > pipeline1

pipeline1

Pipeline type: V2 Execution mode: QUEUED

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

<p>Source</p> <p>GitHub (Version 2)</p> <p>⌚ Succeeded - Just now</p> <p>8be52cba</p> <p>View details</p>	<p></p> <p></p>
---	---

Deploy ⓘ In progress

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

Deploy

[AWS Elastic Beanstalk](#) ↗

⌚ In progress - Just now

[View details](#)

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

pipeline1

Pipeline type: V2 Execution mode: QUEUED

Source Succeeded

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

Source

[GitHub \(Version 2\)](#) ↗

⌚ Succeeded - 1 minute ago

[8be52cba](#) ↗

[View details](#)

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

Deploy ⓘ Succeeded

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

Start rollback

Deploy

[AWS Elastic Beanstalk](#) ↗

⌚ Succeeded - Just now

[View details](#)

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

Applications (1) Info		Actions Create application	
<input type="text"/> Filter results matching the display value			
Application name	Environments	Date created	Last modified
Application1	Application1-env	August 9, 2024 20:11:10 (...)	August 9, 2024 20:11:10 (...)



Name: Aditya Ahuja

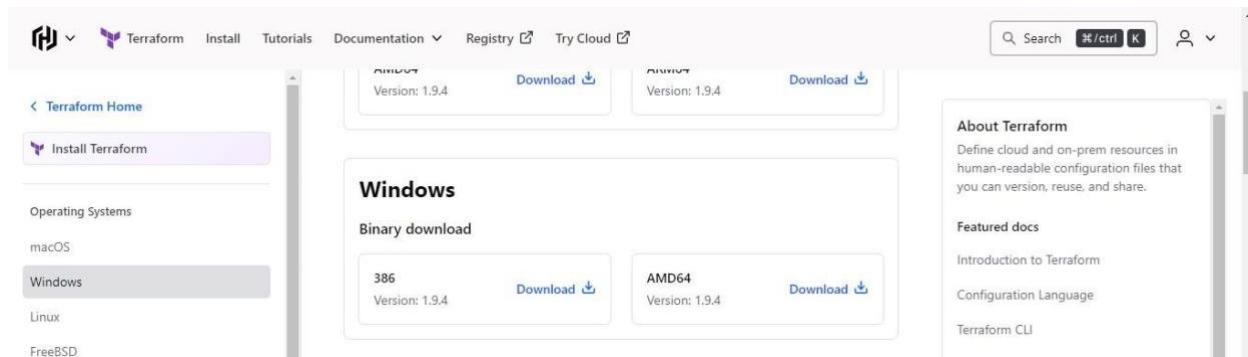
D15C

Roll No: 02

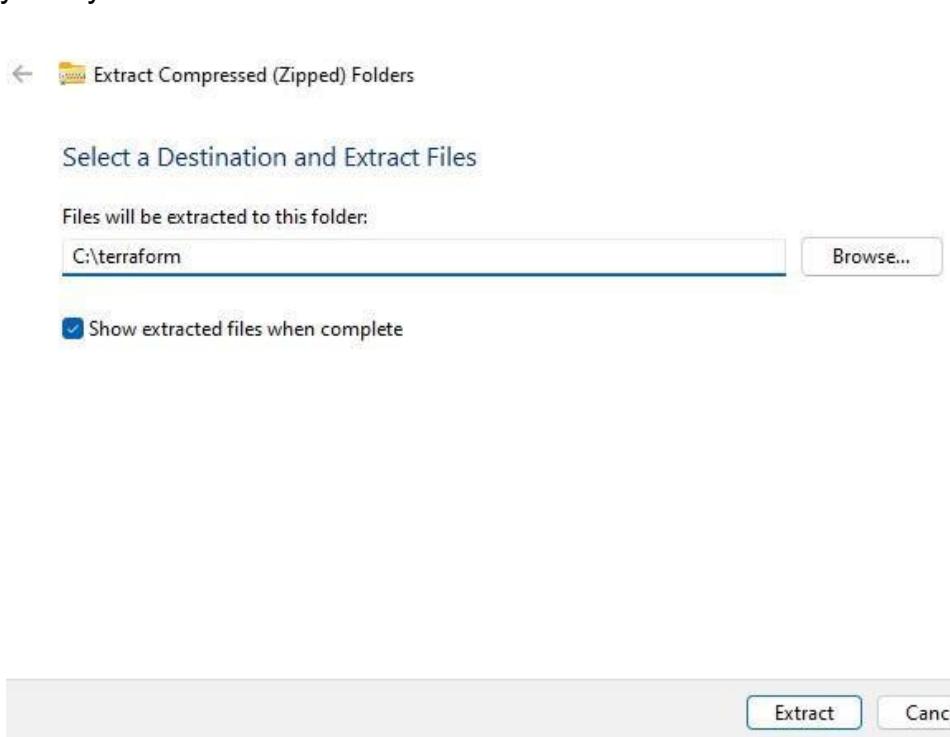
Experiment No. 5

Installation of Terraform

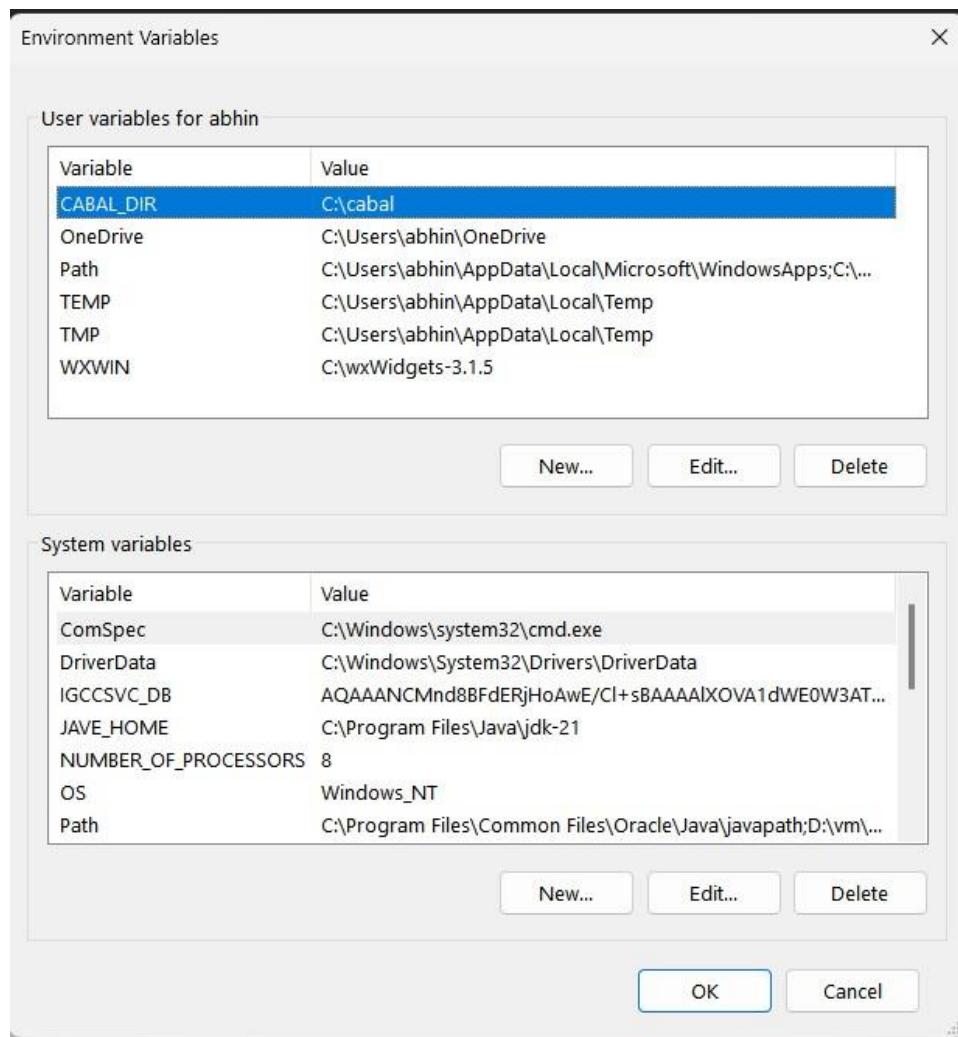
Step 1: To install Terraform, visit the official Terraform website mentioned below, go to the Downloads section, select Windows, and download the 64-bit version for your system. website:<https://www.terraform.io/downloads.html>



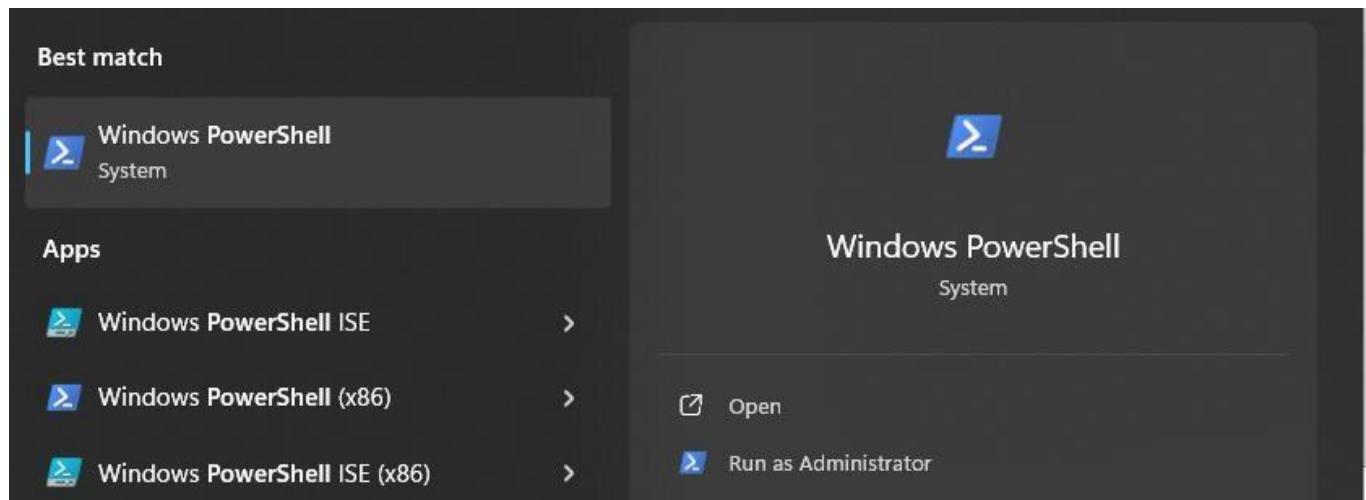
Step 2: Extract the downloaded 'Terraform.exe' file to the 'C:\Terraform' directory on your system.



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



Step 4: Run the windows powershell as administrator.



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

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

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

PS C:\Users\abhin> terraform
Usage: terraform [global options] <subcommand> [args]

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

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

All other commands:
  console    Try Terraform expressions at an interactive command prompt
  fmt        Reformat your configuration in the standard style
  force-unlock Release a stuck lock on the current workspace
  get        Install or upgrade remote Terraform modules
  graph      Generate a Graphviz graph of the steps in an operation
  import     Associate existing infrastructure with a Terraform resource
  login      Obtain and save credentials for a remote host
  logout     Remove locally-stored credentials for a remote host
  metadata   Metadata related commands
  output     Show output values from your root module
  providers  Show the providers required for this configuration
  refresh    Update the state to match remote systems
  show       Show the current state or a saved plan
  state      Advanced state management
  taint      Mark a resource instance as not fully functional
```

EXPERIMENT No.4

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

Steps:

1. Select Amazon linux as OS image or use any but change the setting acc to it:

The screenshot shows the AWS Marketplace interface for searching Application and OS Images (Amazon Machine Images). A search bar at the top contains the query "AMZ - nginx". Below the search bar, a section titled "Application and OS Images (Amazon Machine Image)" includes a link to "Info". A descriptive text explains what an AMI is: "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". Below this, there is a search bar with placeholder text "Search our full catalog including 1000s of application and OS images". The main content area displays a grid of recent and quick-start AMI options. The "Recent" section includes "Amazon Linux" (with the AWS logo), "macOS (Mac)", "Ubuntu", "Windows", "Red Hat", and "SUSE LI". The "Quick Start" section includes "Amazon Machine Image (AMI)". To the right, a "Browse more AMIs" button with a magnifying glass icon is shown, along with the text "Including AMIs from AWS, Marketplace and the Community".

2. Make ssh connection in terminal or in browser:

```
quantum@machine ~/Downloads ➤ ssh -i "ec2-ubuntu.pem" ec2-user@ec2-54-162-208-25.compute-1.amazonaws.com
The authenticity of host 'ec2-54-162-208-25.compute-1.amazonaws.com (54.162.208.25)' can't be established.
ED25519 key fingerprint is SHA256:tBrdxB+9Hn3KWL0YZWmzh1wg/R4s+e7QDAMBJPv/E.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-162-208-25.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

# 
~\_\_ #####_ Amazon Linux 2
~\_\_ \#####\_
~\_\_ \###| AL2 End of Life is 2025-06-30.
~\_\_ \#/ --- 
~\_\_ \v\` '-'>
~\_\_ / A newer version of Amazon Linux is available!
~\_\_ . / Amazon Linux 2023, GA and supported until 2028-05-15.
~\_\_ /m/ https://aws.amazon.com/linux/amazon-linux-2023/
[ec2-user@ip-172-31-31-63 ~]$ sudo yum update
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
No packages marked for update
[ec2-user@ip-172-31-31-63 ~]$ sudo yum upgrade
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
No packages marked for update
[ec2-user@ip-172-31-31-63 ~]$
```

3. Install Docker

Use below commands and enable and test docker;

```
sudo dnf update
sudo dnf install docker
sudo systemctl enable docker
sudo systemctl start docker
```

```
[ec2-user@ip-172-31-24-190 ~]$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
clec31eb5944: Pull complete
Digest: sha256:91fb4b041da273d5a3273b6d587d62d518300a6ad268b28628f74997b93171b2
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.
```

4. Install Kubernetes

Install CNI plugins :

```
CNI_PLUGINS_VERSION="v1.3.0"
ARCH="amd64"
DEST="/opt/cni/bin"
sudo mkdir -p "$DEST"
curl -L
"https://github.com/containernetworking/plugins/releases/download/${CNI_PLUGINS_
VERSION}/cni-plugins-linux-${ARCH}-${CNI_PLUGINS_VERSION}.tgz" | sudo tar -C
"$DEST" -xz
```

Define the directory to download command files:

```
DOWNLOAD_DIR="/usr/local/bin"
sudo mkdir -p "$DOWNLOAD_DIR"
```

Optionally install crictl (required for interaction with the Container Runtime Interface (CRI), optional for kubeadm):

```
CRICCTL_VERSION="v1.31.0"
ARCH="amd64"
curl -L
"https://github.com/kubernetes-sigs/cri-tools/releases/download/${CRICCTL_VERSION}/
crictl-${CRICCTL_VERSION}-linux-${ARCH}.tar.gz" | sudo tar -C $DOWNLOAD_DIR -xz
```

Install kubeadm, kubelet and add a kubelet systemd service:

```
RELEASE=$(curl -sSL https://dl.k8s.io/release/stable.txt)
ARCH="amd64"
cd $DOWNLOAD_DIR
sudo curl -L --remote-name-all
https://dl.k8s.io/release/${RELEASE}/bin/linux/${ARCH}/{kubeadm,kubelet}
sudo chmod +x {kubeadm,kubelet}
```

```
RELEASE_VERSION="v0.16.2"
curl -sSL
"https://raw.githubusercontent.com/kubernetes/release/${RELEASE_VERSION}/cmd/kr
el/templates/latest/kubelet/kubelet.service" | sed "s:/usr/bin:${DOWNLOAD_DIR}:g" |
sudo tee /usr/lib/systemd/system/kubelet.service
sudo mkdir -p /usr/lib/systemd/system/kubelet.service.d
curl -sSL
"https://raw.githubusercontent.com/kubernetes/release/${RELEASE_VERSION}/cmd/kr
el/templates/latest/kubeadm/10-kubeadm.conf" | sed "s:/usr/bin:${DOWNLOAD_DIR}:g" |
sudo tee /usr/lib/systemd/system/kubelet.service.d/10-kubeadm.conf
```

Now we need to install kubectl

Set up repository:

```
cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/repo/repodata/repomd.xml.key
EOF
```

```
sudo yum install -y kubectl
```

```
ec2-user@ip-172-31-24-190 ~ $ kubectl version
Client Version: v1.31.1
Kustomize Version: v5.4.2
```

We have installed successfully installed kubernetes

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

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

```
[root@ip-172-31-24-190 bin]# sudo swapoff -a
echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf
sudo sysctl -p
net.bridge.bridge-nf-call-iptables=1
net.bridge.bridge-nf-call-iptables = 1
[root@ip-172-31-24-190 bin]#
```

Disable SELINUX

Type **sudo nano /etc/selinux/config** and set the value of **SELINUX=disabled** instead of **SELINUX=permissive**

Save the file by pressing **ctrl+o** then press enter then press **ctrl+x**

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - No SELinux policy is loaded.
# See also:
# https://docs.fedoraproject.org/en-US/quick-docs/getting-started-with-selinux/#getting-started-with-selinux-selinux-states-and-modes
#
# NOTE: In earlier Fedora kernel builds, SELINUX=disabled would also
# fully disable SELinux during boot. If you need a system with SELinux
# fully disabled instead of SELinux running with no policy loaded, you
# need to pass selinux=0 to the kernel command line. You can use grubby
# to persistently set the bootloader to boot with selinux=0:
#
#     grubby --update-kernel ALL --args selinux=0
#
# To revert back to SELinux enabled:
#
#     grubby --update-kernel ALL --remove-args selinux
#
SELINUX=disabled
# SELINUXTYPE= can take one of these three values:
#       targeted - Targeted processes are protected,
#       minimum - Modification of targeted policy. Only selected processes are protected.
#       mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Then reboot the system using **sudo reboot**

After rebooting we need to make ssh connection with machine after it gets disconnected

Now if we type command **sestatus**, then it show disabled

```
ec2-user@ip-172-31-24-190 ~ $ sestatus
SELinux status:                 disabled
```

5. Initialize the Kubecluster

Install packages socat and iproute-tc and conntrack to avoid prelight errors

```
sudo dnf install socat iproute-tc conntrack-tools -y
```

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

```
Your Kubernetes control-plane has initialized successfully!

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

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

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

export KUBECONFIG=/etc/kubernetes/admin.conf

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

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

kubeadm join 172.31.24.190:6443 --token xsbsq1.6ro11sawnvttbsvu \
  --discovery-token-ca-cert-hash sha256:10d2b67f4f4749b51854065a554c74e6a956e4782d9ab4bb79b8591648b3edef
ec2-user@ip-172-31-24-190 ~ $ kubectl get nodes
```

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

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

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

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

```
sudo systemctl restart kubelet
```

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

```
kubectl apply -f
```

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

```
  clusterrole.rbac.authorization.k8s.io/flannel created
  clusterrolebinding.rbac.authorization.k8s.io/flannel created
  serviceaccount/flannel created
  configmap/kube-flannel-cfg created
  daemonset.apps/kube-flannel-ds created
  ec2-user@ip-172-31-24-190 ~ $ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
  namespace/kube-flannel created
```

Now type **kubectl get nodes**

```
The connection to the server 172.31.24.190:6443 was refused - did you specify the right host or port?
ec2-user@ip-172-31-24-190 ~ $ kubectl get nodes
The connection to the server 172.31.24.190:6443 was refused - did you specify the right host or port?
ec2-user@ip-172-31-24-190 ~ $ kubectl get nodes
^[[AError from server (Forbidden): nodes is forbidden: User "kubernetes-admin" cannot list resource "nodes" in
ec2-user@ip-172-31-24-190 ~ $ kubectl get nodes
NAME STATUS ROLES AGE VERSION
ip-172-31-24-190.ec2.internal Ready control-plane 34m v1.31.0
ec2-user@ip-172-31-24-190 ~ $ kubectl get nodes
NAME STATUS ROLES AGE VERSION
ip-172-31-24-190.ec2.internal Ready control-plane 34m v1.31.0
ec2-user@ip-172-31-24-190 ~ $
```

Now that the cluster is up and running, we can deploy our nginx server on this cluster.

Apply this deployment file using this command to create a deployment

```
ec2-user@ip-172-31-24-190 ~ $ kubectl apply -f https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
```

Use 'kubectl get pods' to verify if the deployment was properly created and the pod is working correctly.

```
ec2-user@ip-172-31-24-190 ~ $ kubectl get pods
NAME READY STATUS RESTARTS AGE
nginx-deployment-d556bf558-mwd8p 0/1 Pending 0 7s
nginx-deployment-d556bf558-zc25s 0/1 Pending 0 7s
```

As we can see our pods are in pending state

On checking logs to we came to know the pods are in tainted state (using command

kubectl describe pod nginx-deployment-d556bf558-mwd8p)

```
Events:
Type Reason Age From Message
---- ---- -- -- -----
Warning FailedScheduling 56s default-scheduler 0/1 nodes are available: 1 node(s) had untolerated taint {node-role.kubernetes.io/control-plane: }. preemption: 0/1 nodes are available:
Reason: no available nodes
```

To make pods untainted

Type kubectl get nodes to see name of node

```
ec2-user@ip-172-31-24-190 ~ $ kubectl get nodes
NAME STATUS ROLES AGE VERSION
ip-172-31-24-190.ec2.internal Ready control-plane 43m v1.31.0
ec2-user@ip-172-31-24-190 ~ $
```

Copy the name of the node (ip-172-31-24-190.ec2.internal)

Then type command **kubectl taint nodes <NODE_NAME> - -all**

In my case **kubectl taint nodes ip-172-31-24-190.ec2.internal node-role.kubernetes.io/control-plane-**

```
ec2-user@ip-172-31-24-190 ~ $ kubectl taint nodes ip-172-31-24-190.ec2.internal node-role.kubernetes.io/control-plane-
node/ip-172-31-24-190.ec2.internal untainted
```

After executing above command, check again status of pods if still pending then restart kubelet wait for 1-2 minutes and check again

```
ec2-user@ip-172-31-24-190 ~ $ kubectl get pods
NAME                  READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-mwd8p   1/1     Running   2 (73s ago)   12m
nginx-deployment-d556bf558-zc25s   1/1     Running   2 (73s ago)   12m
```

As we can see our pods are running

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

kubectl port-forward <POD_NAME> 8080:80

In my case : **kubectl port-forward nginx-deployment-d556bf558-mwd8p 8080:80**

Note: if you are getting connection refused error then restart kubelet

```
ec2-user@ip-172-31-24-190 ~ $ kubectl port-forward nginx-deployment-d556bf558-mwd8p 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
```

As port forwarding is active so we cannot type other commands.

Open new terminal window and make ssh connection to same machine OR we can open instance of same machine in new browser tab

And type command **curl --head <http://127.0.0.1:8080>**

```
ec2-user@ip-172-31-24-190 ~ $ curl --head http://127.0.0.1:8080
HTTP/1.1 200 OK
Server: nginx/1.14.2
Date: Sat, 14 Sep 2024 06:54:21 GMT
Content-Type: text/html
Content-Length: 612
Last-Modified: Tue, 04 Dec 2018 14:44:49 GMT
Connection: keep-alive
ETag: "5c0692e1-264"
Accept-Ranges: bytes

ec2-user@ip-172-31-24-190 ~ $ █
3 1:ec2-user@ip-172-31-24-190:~# - 2:ec2-user@ip-172-31-24-190:~* 3:~/Downloads
```

Response status 200 (OK) indicates that our nginx server is running successfully on kubernetes

Conclusion: We started with the installation and configuration of Docker and Kubernetes. Initially, the Kubernetes API server encountered some problems, but these were fixed by restarting the kubelet service. The pods we created were not running because the nodes had taints, so we had to remove those taints. After addressing all the issues, the NGINX server pods were successfully deployed and made accessible through port forwarding. The NGINX server can now be accessed from different terminals or by running the port-forward command in the background using an '&' at the end of the command.

Name: Aditya Ahuja

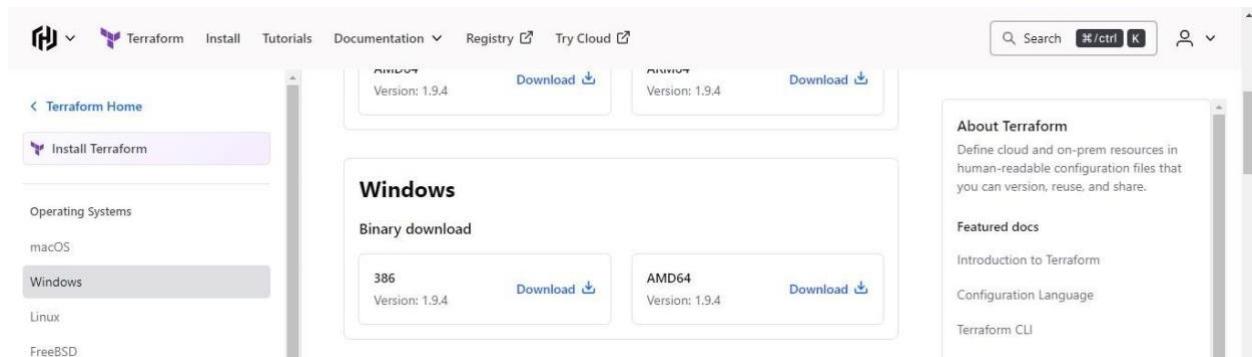
D15C

Roll No: 02

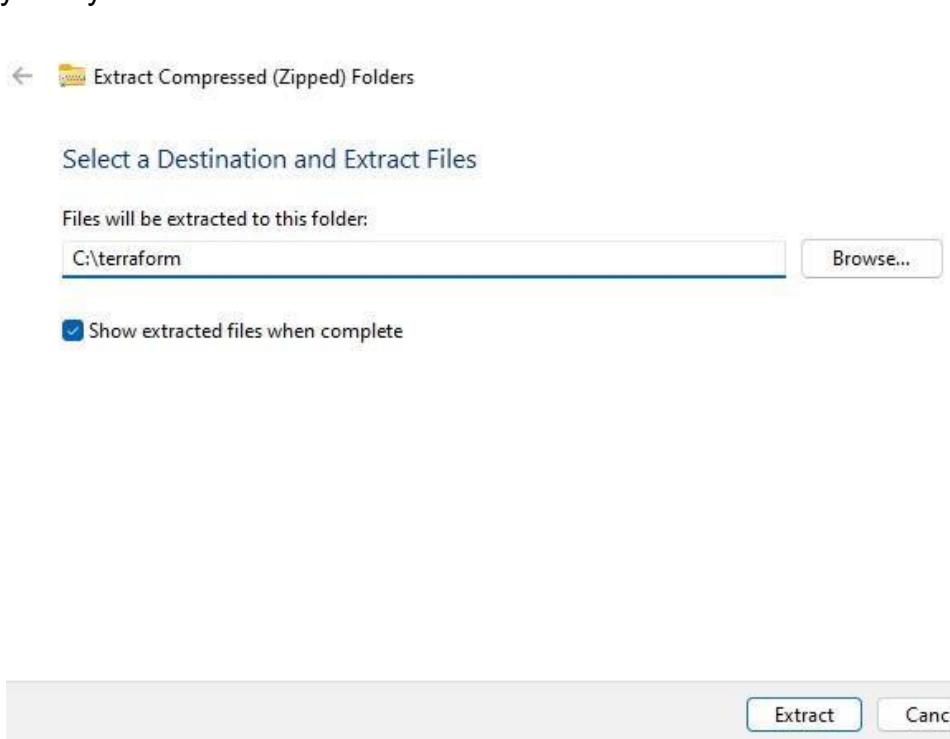
Experiment No. 5

Installation of Terraform

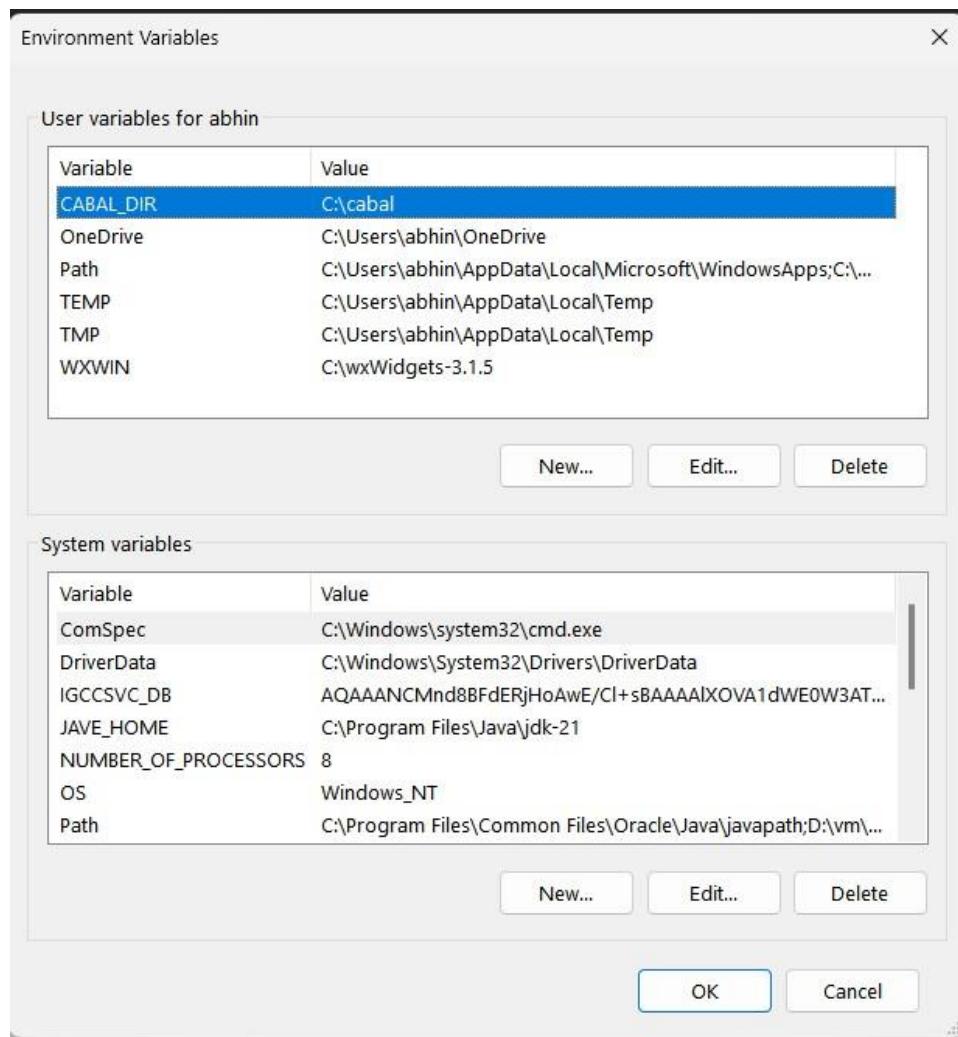
Step 1: To install Terraform, visit the official Terraform website mentioned below, go to the Downloads section, select Windows, and download the 64-bit version for your system. website: <https://www.terraform.io/downloads.html>



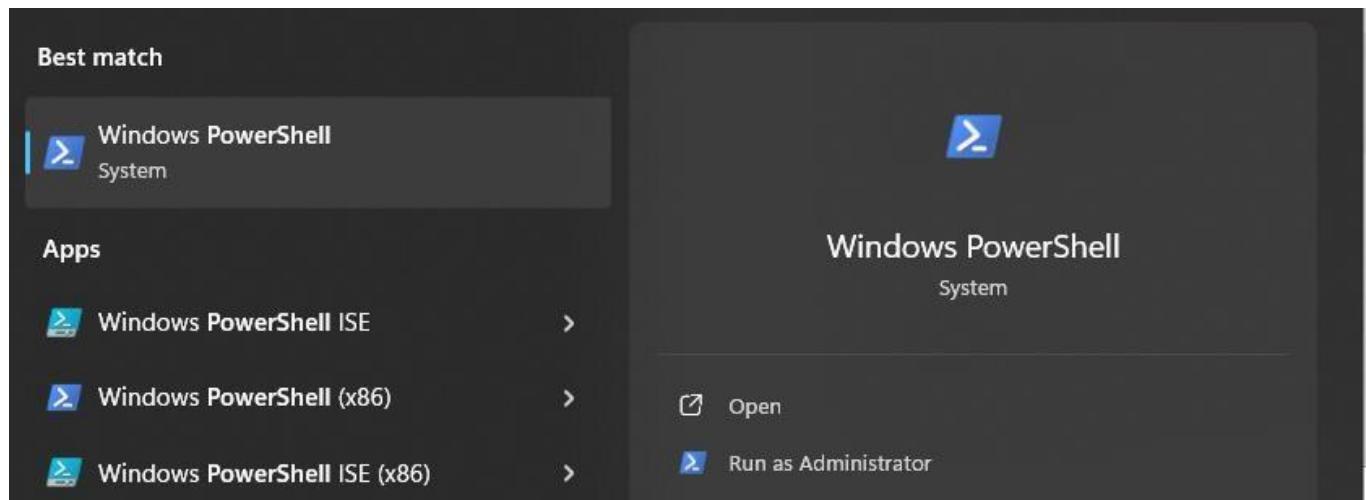
Step 2: Extract the downloaded 'Terraform.exe' file to the 'C:\Terraform' directory on your system.



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



Step 4: Run the windows powershell as administrator.



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

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

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

PS C:\Users\abhin> terraform
Usage: terraform [global options] <subcommand> [args]

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

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

All other commands:
  console    Try Terraform expressions at an interactive command prompt
  fmt        Reformat your configuration in the standard style
  force-unlock Release a stuck lock on the current workspace
  get        Install or upgrade remote Terraform modules
  graph      Generate a Graphviz graph of the steps in an operation
  import     Associate existing infrastructure with a Terraform resource
  login      Obtain and save credentials for a remote host
  logout     Remove locally-stored credentials for a remote host
  metadata   Metadata related commands
  output     Show output values from your root module
  providers  Show the providers required for this configuration
  refresh    Update the state to match remote systems
  show       Show the current state or a saved plan
  state      Advanced state management
  taint      Mark a resource instance as not fully functional
```

Experiment No. 6

- Creating a docker image using terraform

```
C:\Users\abhinav>docker --version
Docker version 26.1.3, build b72abbb

C:\Users\abhinav>docker

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

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

Management Commands:
  builder  Manage builds
  buildx*  Docker Buildx
  compose*  Docker Compose
  container  Manage containers
  context   Manage contexts
```

- Create a folder named ‘Terraform Scripts’ in which we save our different types of scripts which will be further used in this experiment.



- Create a new folder named ‘Docker’ in the ‘TerraformScripts’ folder. Then create a new docker.tf file using a text editor and write the following contents into it to create a Ubuntu Linux container.

A screenshot of a Windows File Explorer window. The path 'Temporary Storage (D:) > Terraform_Scripts > Docker' is visible. Inside the 'Docker' folder, there is a single file icon for 'docker.tf'. The details pane shows the file was modified on '8/22/2024 4:12 PM' and is a 'TF File' with a size of '1 KB'.

Name	Date modified	Type	Size
docker.tf	8/22/2024 4:12 PM	TF File	1 KB

 docker.tf - Notepad

```
File Edit Format View Help
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"
}
```

4. Execute Terraform Init command to initialize the resources

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

Terraform has been successfully initialized!

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

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

D:\Terraform_Scripts\Docker>
```

5. Execute Terraform plan to see the available resources

```
D:\Terraform_Scripts\Docker>terraform plan

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

Terraform will perform the following actions:

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

6. 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”

```
D:\Terraform_Scripts\Dockers>terraform apply

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

Terraform will perform the following actions:

# docker_container.foo will be created
+ resource "docker_container" "foo" {
  + attach           = false
  + bridge           = (known after apply)
  + command          = (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
  + stdio_open          = false
  + stop_signal          = (known after apply)
  + stop_timeout         = (known after apply)
  + tty                 = false

  + healthcheck (known after apply)

  + labels (known after apply)
}

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

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

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

Enter a value: yes

docker image.ubuntu: Creating...
docker image.ubuntu: Still creating... [19s elapsed] docker image.ubuntu: Still creating... (20s elapsed) docker image.ubuntu: Still creating... [30s elapsed]
docker image.ubuntu: Creation complete after 30s [id=sha256:263966596d42ad38ae9914716692777ba9ff8779a62ad93a74fe82e3e1f
ubuntu:latest] docker_container.foo: Creating...
```

7. Check Docker images, Before and After Executing Apply step

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
mcr.microsoft.com/dotnet/framework/aspnet	4.8-windowsservercore-ltsc2022	0b1ef1176a57	6 weeks ago	5.43GB
mcr.microsoft.com/dotnet/framework/sdk	4.8-windowsservercore-ltsc2022	c3f8c2735565	6 weeks ago	9.04GB
mcr.microsoft.com/dotnet/framework/runtime	4.8-windowsservercore-ltsc2022	e69ea8a5ec1b	6 weeks ago	5.1GB
mcr.microsoft.com/windows/servercore	ltsc2022	e60f47e635b7	7 weeks ago	4.84GB
mcr.microsoft.com/windows/nanoserver	ltsc2022	f0ca29645006	7 weeks ago	292MB

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
mcr.microsoft.com/dotnet/framework/aspnet	4.8-windowsservercore-ltsc2022	0b1ef1176a57	6 weeks ago	5.43GB
mcr.microsoft.com/dotnet/framework/sdk	4.8-windowsservercore-ltsc2022	c3f8c2735565	6 weeks ago	9.04GB
mcr.microsoft.com/dotnet/framework/runtime	4.8-windowsservercore-ltsc2022	e69ea8a5ec1b	6 weeks ago	5.1GB
mcr.microsoft.com/windows/servercore	ltsc2022	e60f47e635b7	7 weeks ago	4.84GB
mcr.microsoft.com/windows/nanoserver	ltsc2022	f0ca29645006	7 weeks ago	292MB
ubuntu	Latest	2dc39ba859dc	2 minutes ago	77.8MB

```
D:\Terraform_Scripts\Dockers>docker images
docker_image.ubuntu: Refreshing state... [id:sha256:2dc29b50cd2d30101475692777ba087762d92de0221fubuntu: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      = "sha256:2dc39b859dc2ade8ae30147b5692777ba9ff9779a62ad93a78de82e3elfubuntu:latest" -> null
  - image_id = "sha256:2dc39b859dc2ade8ae30147b5692777ba9ff9779a62ad93a78de82e3elf" -> null
  - Latest   = "sha256:2dc39b859dc2ade8ae30147b5692777ba9ff9779a62ad93a78de82e3elf" -> null
  - name     = "ubuntu:latest" -> null
  - repo digest = "ubuntu@sha256:204a3d7bb4d7723452be3923b06cd7043704030041c83c#7856c1" -> null
}

Plan: to add, 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:sha256:2de99b59cd42ade83814765692777ba5ff8779a62ad93ad62e3e1fubuntu:latest]
docker image.ubuntu: Destruction complete after 1s

Destroy complete! Resources: 1 destroyed.
```

Adv DevOps Practical 7

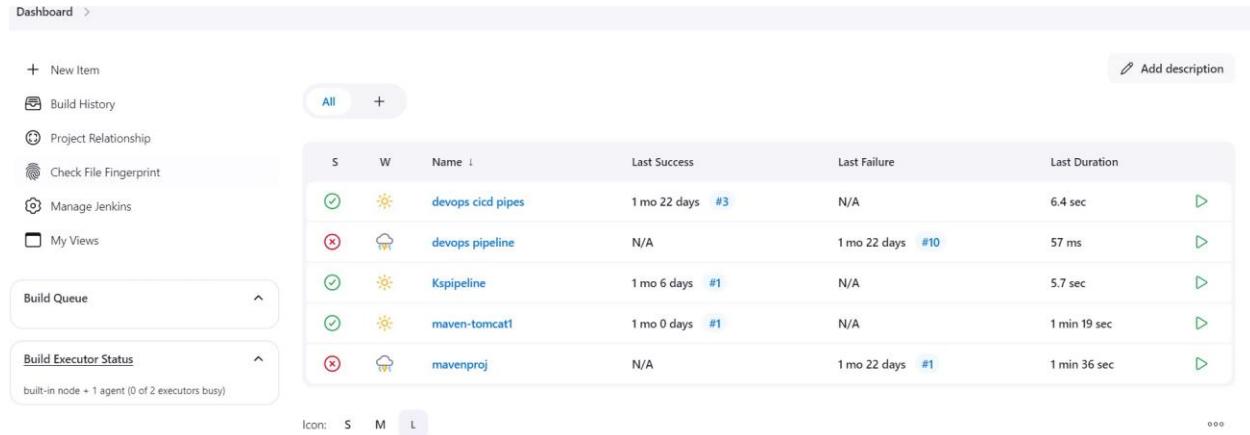
Aim: To understand Static Analysis SAST process and learn to integrate Jenkins SAST to SonarQube/GitLab.

Integrating Jenkins with SonarQube:

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

Steps to integrate Jenkins with SonarQube

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



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

- Left Sidebar:** Includes links for "New Item", "Build History", "Project Relationship", "Check File Fingerprint", "Manage Jenkins", and "My Views".
- Top Bar:** Buttons for "All" and "+" to add new items, and a "Add description" button.
- Table View:** A table listing five Jenkins projects:

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀️	devops cicd pipes	1 mo 22 days #3	N/A	6.4 sec
✗	🌧️	devops pipeline	N/A	1 mo 22 days #10	57 ms
✓	☀️	Kspipeline	1 mo 6 days #1	N/A	5.7 sec
✓	☀️	maven-tomcat1	1 mo 0 days #1	N/A	1 min 19 sec
✗	🌧️	mavenproj	N/A	1 mo 22 days #1	1 min 36 sec
- Bottom Navigation:** Icons for "S", "M", and "L" (Large) and a "000" link.

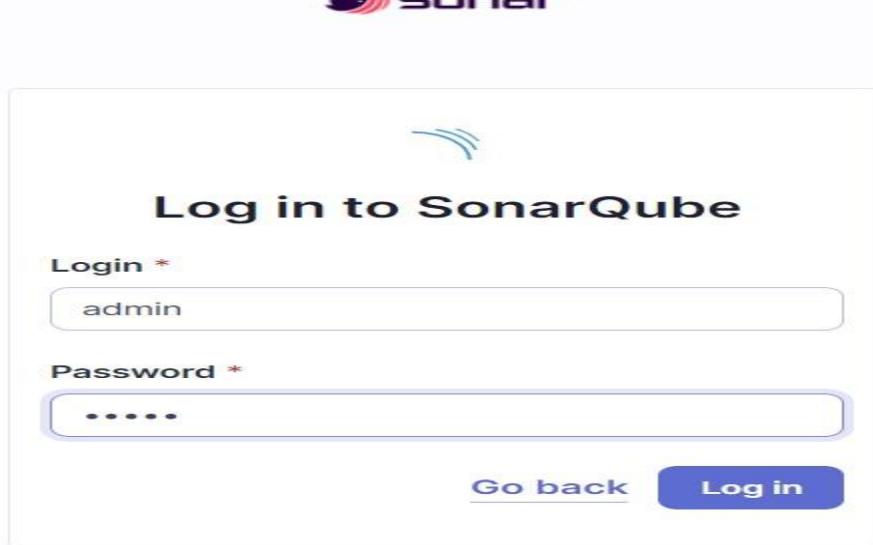
2. Run SonarQube in a Docker container using this command -

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

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

```
PS C:\Users\91773\Desktop\College Resources\Exp7 adv devops> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
77e678cded2ef5f989912d3d9e6991dd548eac03faa1eed68dd906614be53acc
PS C:\Users\91773\Desktop\College Resources\Exp7 adv devops>
```

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



- Login to SonarQube using username admin and password admin.

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

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

1 of 2

Create a local project

Project display name *

exp7



Project key *

exp7



Main branch name *

main

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

[Cancel](#)

[Next](#)

Setup the project and come back to Jenkins Dashboard.

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

The screenshot shows the Jenkins 'Manage Jenkins' interface under the 'Plugins' section. A search bar at the top right contains the text 'sonar'. Below it, a table lists three Jenkins plugins related to SonarQube:

- SonarQube Scanner 2.17.2**: Released 7 months ago. Description: This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.
- Sonar Quality Gates 315.vf1f12b_e81a_3a_4**: Released 29 days ago. Description: Library plugins (for use by other plugins) analysis Other Post-Build Actions. Fails the build whenever the Quality Gates criteria in the Sonar 5.6+ analysis aren't met (the project Quality Gates status is different than "Passed")
- Quality Gates 2.5**: Fails the build whenever the Quality Gates criteria in the Sonar analysis aren't met (the project Quality Gates status is different than "Passed")

- Under Jenkins 'Manage Jenkins' then go to 'system', scroll and look for **SonarQube Servers** and enter the details.

Enter the Server Authentication token if needed.

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

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

The screenshot shows the 'SonarQube servers' configuration page. It includes fields for:

- SonarQube servers**: A checkbox for injecting SonarQube server configuration as environment variables.
- Environment variables**: A checked checkbox.
- SonarQube installations**: A list of installations with one entry: 'List of SonarQube installations'.
- Name**: A text input field containing 'exp7'.
- Server URL**: A text input field containing 'http://localhost:9000'. A note says 'Default is http://localhost:9000'.
- Server authentication token**: A dropdown menu currently set to '- none -'. A note says 'SonarQube authentication token. Mandatory when anonymous access is disabled.' A '+ Add' button is available.
- Advanced**: A dropdown menu.

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

Dashboard > Manage Jenkins > Tools

Dashboard > Manage Jenkins > Tools

Add Git ▾

Gradle installations

Add Gradle

SonarScanner for MSBuild installations

Add SonarScanner for MSBuild

SonarQube Scanner installations

Add SonarQube Scanner

Ant installations

Add Ant

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

SonarQube Scanner

Name
sonarqube_exp7

Install automatically ?

Install from Maven Central

Version
SonarQube Scanner 6.2.0.4584

Add Installer ▾

Add SonarQube Scanner

Ant installations

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

New Item

Enter an item name

Select an item type



Freestyle project

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



Maven project

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



Pipeline

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



Multi-configuration project

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

OK

9. Choose this GitHub repository in Source Code Management.

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

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

The screenshot shows the Jenkins configuration interface for a new item. Under 'Source Code Management', the 'Git' option is selected. A repository is configured with the URL https://github.com/shazforiot/MSBuild_firstproject.git. The 'Credentials' dropdown is set to '- none -'. There is an 'Advanced' section with a dropdown menu. At the bottom, there is a button labeled 'Add Repository'.

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

Dashboard > exp7 > Configuration

Filter

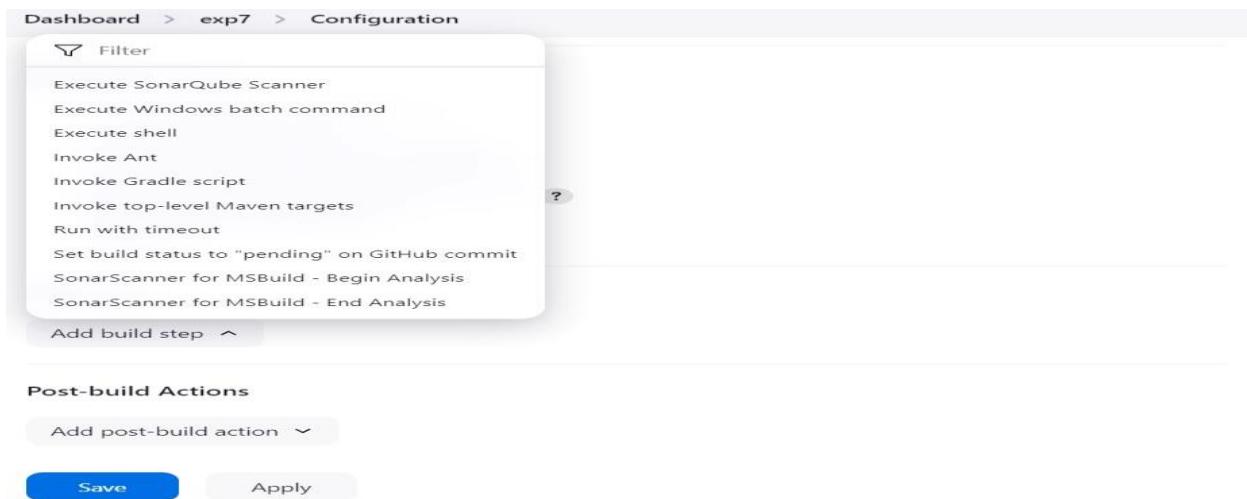
- Execute SonarQube Scanner
- Execute Windows batch command
- Execute shell
- Invoke Ant
- Invoke Gradle script
- Invoke top-level Maven targets
- Run with timeout
- Set build status to "pending" on GitHub commit
- SonarScanner for MSBuild - Begin Analysis
- SonarScanner for MSBuild - End Analysis

Add build step ^

Post-build Actions

Add post-build action ▾

Save **Apply**



≡ Execute SonarQube Scanner

JDK ?

JDK to be used for this SonarQube analysis

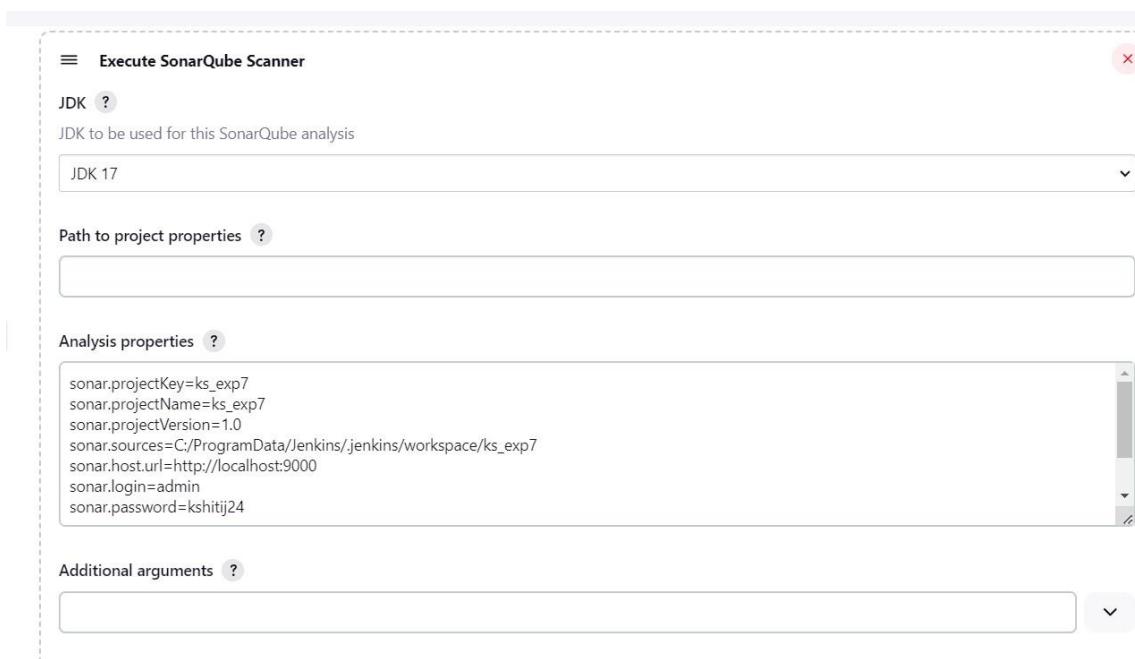
JDK 17

Path to project properties ?

Analysis properties ?

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

Additional arguments ?



Dashboard > ks_exp7 >

Status **ks_exp7**

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

SonarQube

Permalinks

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

Build History **trend** **#7** Sep 25, 2024, 3:09 PM

Console Output

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

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

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

4 of 4 shown

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

The screenshot shows the SonarQube interface for the 'main' project. At the top, there's a navigation bar with tabs for Overview, Issues, Security Hotspots, Measures, Code, and Activity. On the right, there are Project Settings and Project Information dropdowns. The main content area is titled 'main'. It features a large green 'Passed' button with a checkmark, indicating the quality gate status. Below it, a yellow warning box says 'The last analysis has warnings. See details'. Underneath, there are four cards: 'New Code' (Overall Code selected), 'Security' (0 Open issues, A grade), 'Reliability' (0 Open issues, A grade), and 'Maintainability' (0 Open issues, A grade). Each card has a horizontal bar with segments labeled '0 H', '0 M', and '0 L'.

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

Conclusion:

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

Expt No. 08 Advanced DevOps Lab

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

Theory:

What is SAST?

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

What problems does SAST solve?

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

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

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

Why is SAST important?

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

What is a CI/CD Pipeline?

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

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



Now, consider the CI/CD pipeline as the backbone of the DevOps approach. This Pipeline is responsible for building codes, running tests, and deploying new software versions. The Pipeline

executes the job in a defined manner by first coding it and then structuring it inside several blocks that may include several steps or tasks.

What is SonarQube?

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

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

Benefits of SonarQube

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

Integrating Jenkins with SonarQube:

Prerequisites:

- Jenkins installed

- Docker Installed (for SonarQube)

- SonarQube Docker Image

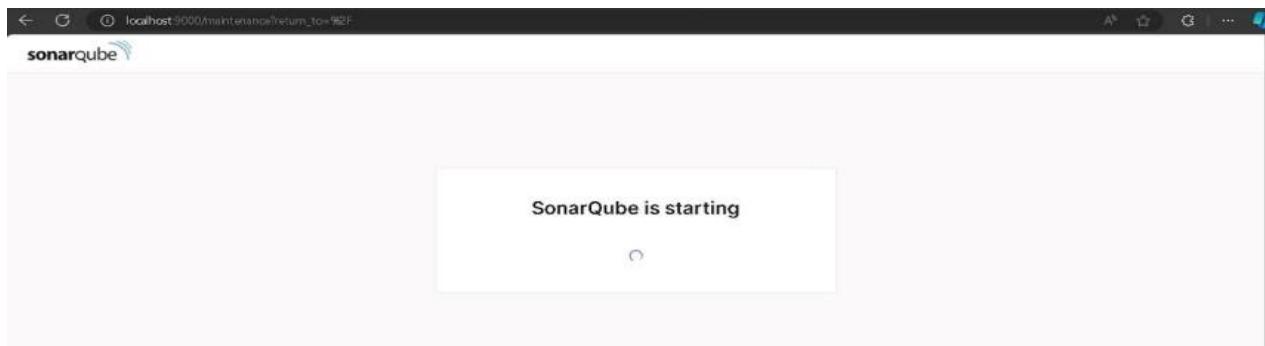
Steps to create a Jenkins CI/CD Pipeline and use SonarQube to perform SAST

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

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

```
| PS C:\Users\91773\Desktop\College Resources\Advdevops Exp8> docker run -d --name sonarqube2 -e SONAR_ES_BOOTSTRAP_CHECKS  
_DISABLE=true -p 9000:9000 sonarqube:latest  
71fc67f0b15baa5be5bcd66966938e18682683d020beadcbc909dd027cfe7a  
PS C:\Users\91773\Desktop\College Resources\Advdevops Exp8>
```

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



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

The form contains the following fields:

- Login ***: admin
- Password ***: *****
- Buttons**: Go back, Log in

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

1 of 2

Create a local project

Project display name *

Project key *

Main branch name *

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

Setup the project and come back to Jenkins Dashboard.

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

New Item

Enter an item name

Select an item type



Freestyle project

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



Maven project

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



Pipeline

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



Multi-configuration project

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

7. Under Pipeline Script, enter the following -

```
node {
```

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

Configure

The screenshot shows a Jenkins Pipeline configuration page. The left sidebar has 'General' and 'Advanced Project Options' selected. The main area is titled 'Pipeline' and contains a 'Definition' section with a 'Pipeline script' tab selected. The script content is as follows:

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

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

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

8. Run The Build.
 9. Check the console output once the build is complete.

Status

- </> Changes
- ▷ Build Now
- ⚙ Configure
- trash Delete Pipeline
- 🔍 Full Stage View
- SonarQube
- /Stages
- ✍ Rename
- ⓘ Pipeline Syntax

KsSonarQube

Stage View

		Cloning the GitHub Repo	SonarQube Analysis
Average stage times:	2s	1min 44s	
(Average full run time: ~8min 36s)			
#9 Sep 25 20:49 No Changes	2s	8min 33s	
#8 Sep 25 20:44 No Changes	4s	835ms failed	
#7 Sep 25 20:42 No Changes	2s	3s failed	
#6 Sep 25 20:31 No Changes	2s	3s failed	

Build History

trend ▾

Filter... /

#9 Sep 25, 2024, 8:49 PM

Status

- </> Changes
- Console Output
- View as plain text
- Edit Build Information
- trash Delete build '#9'
- ⌚ Timings
- Git Build Data
- Pipeline Overview
- Pipeline Console
- Replay
- Pipeline Steps
- Workspaces
- ← Previous Build

Console Output

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

```

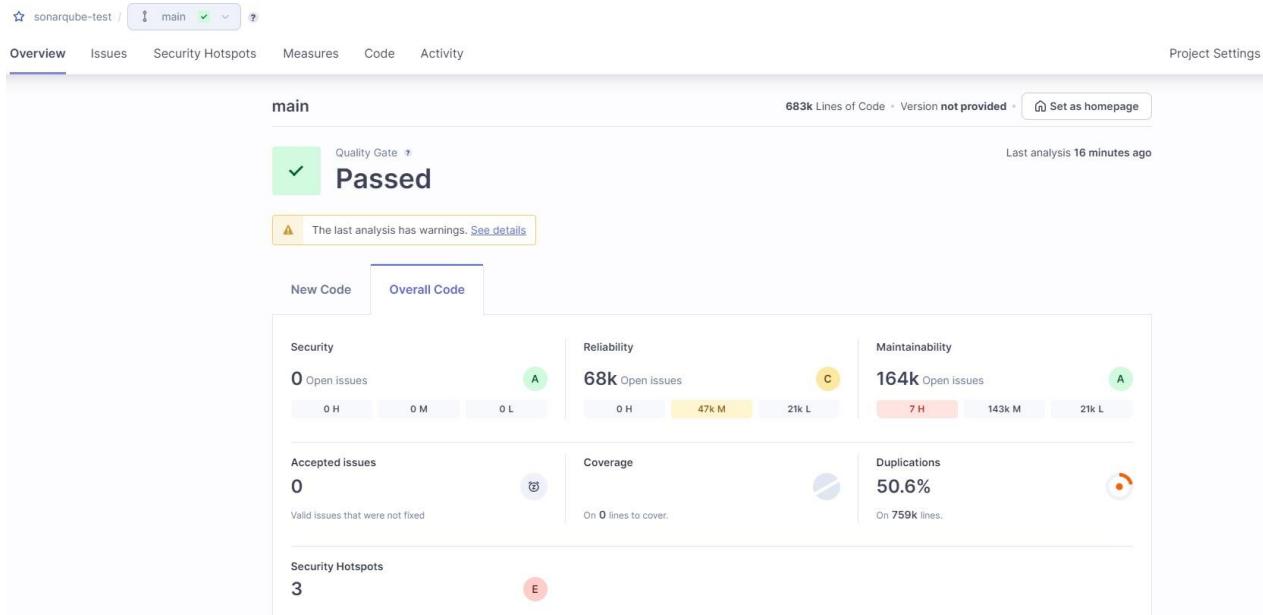
20:56:15.267 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 798. Keep only the first 100 references.
20:56:15.267 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPane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

```

Kshitij Hundre > My Views > All > KsSonarQube > #9

```
for block at line 17. Keep only the first 100 references.
20:56:18.455 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html
for block at line 296. Keep only the first 100 references.
20:56:18.455 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html
for block at line 100. Keep only the first 100 references.
20:56:18.456 INFO CPD Executor CPD calculation finished (done) | time=107093ms
20:56:18.490 INFO SCM revision ID 'ba799ba7e1b576f04a4612322b0412c5e6e1e5e4'
20:57:50.106 INFO Analysis report generated in 3149ms, dir size=127.2 MB
20:57:56.943 INFO Analysis report compressed in 6828ms, zip size=29.6 MB
20:57:58.685 INFO Analysis report uploaded in 1732ms
20:57:58.688 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://127.0.0.1:9000/dashboard?id=sonarqube-test
20:57:58.688 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
20:57:58.688 INFO More about the report processing at http://127.0.0.1:9000/api/ce/task?id=18847db4-4f06-4766-9ad4-ee006448353c
20:58:06.225 INFO Analysis total time: 8:22.672 s
20:58:06.231 INFO SonarScanner Engine completed successfully
20:58:06.824 INFO EXECUTION SUCCESS
20:58:06.857 INFO Total time: 8:31.713s
[Pipeline]
[Pipeline] // withSonarQubeEnv
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

10. After that, check the project in SonarQube.



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

Bugs

The screenshot shows the SonarQube Issues page with the following filters applied:

- Responsibility**: Add to selection Ctrl + click
- Software Quality** (selected): Security (0), Reliability (33k), Maintainability (0)
- Severity**: Add to selection Ctrl + click
- Type**: Bug (33k) (selected), Vulnerability (0), Code Smell (164k)
- Scope**: Add to selection Ctrl + click
- Status**: Add to selection Ctrl + click

The page displays three reports:

- gameoflife-core/build/reports/tests/all-tests.html**: 1 issue: "Insert a <!DOCTYPE> declaration to before this <html> tag." (Reliability, Consistency, user-experience)
- gameoflife-core/build/reports/tests/allclasses-frame.html**: 1 issue: "Insert a <!DOCTYPE> declaration to before this <html> tag." (Reliability, Consistency, user-experience)
- gameoflife-core/build/reports/tests/alltests-errors.html**: 1 issue: "Insert a <!DOCTYPE> declaration to before this <html> tag." (Reliability, Consistency, user-experience)

Total: 32,896 issues | 1369d effort

Code Smells

The screenshot shows the 'Code Smells' section of a software tool. On the left, a sidebar lists categories: Software Quality, Severity, Type (Bug, Vulnerability, Code Smell), Scope, Status, and Security Category. The 'Code Smell' category is selected, highlighted in blue. On the right, a main pane displays a list of specific code smell issues. At the top of the list is an issue titled 'Remove this deprecated "width" attribute.' with a 'Consistency' tag. Below it are three more issues, each with a 'Consistency' tag and a note indicating they are 'html5 obsolete'. The total count of 163,766 issues and effort of 1705d are displayed at the top right.

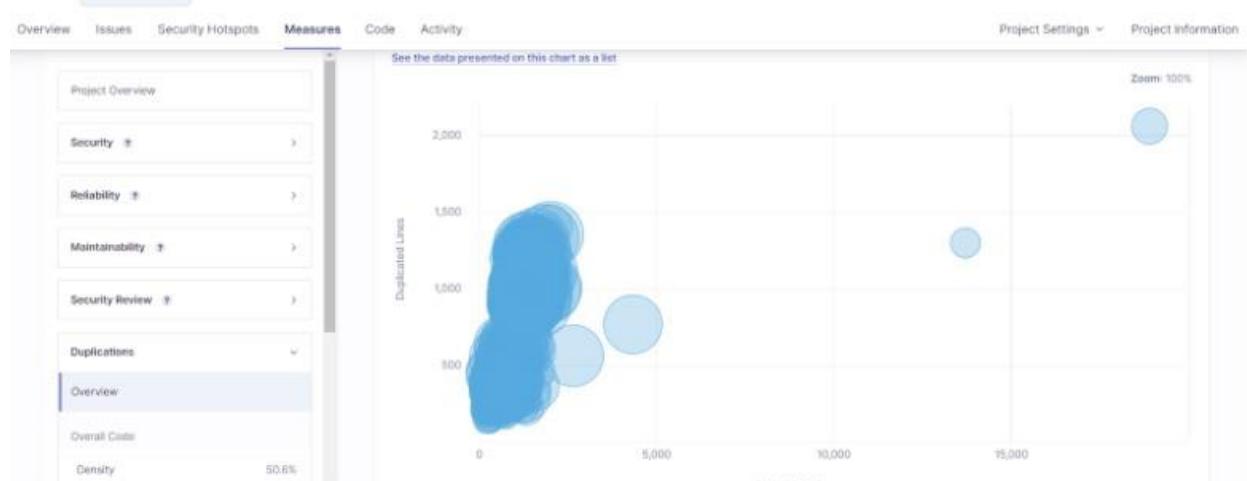
Intentional issues

The screenshot shows the 'Intentional issues' section. The sidebar on the left lists Clean Code Attribute (Consistency, Intentionality, Adaptability, Responsibility), Software Quality, Severity, and Type (Bug, Vulnerability, Code Smell). The 'Intentionality' category is selected. The main pane lists several intentional issues under the file 'gameoflife-acceptance-tests/Dockerfile'. Each issue is associated with an 'Intentionality' tag and a note indicating it is 'No tags'. The total count of 13,887 issues and effort of 59d are displayed at the top right.

Reliabilities issue

The screenshot shows the SonarQube interface for a project named "gameoflife-core/build/reports/tests/all-tests.html". At the top right, it displays "53,752 issues" and "1587d effort". On the left, there's a sidebar titled "Issues in new code" with sections for "Clean Code Attribute" and "Software Quality". Under "Clean Code Attribute", "Consistency" is highlighted with 54k issues. Under "Software Quality", "Reliability" is highlighted with 54k issues. Below these are sections for "Severity" and "Type". On the right, several issues are listed under the "Reliability" category, each with a checkbox, a title, a severity level (e.g., "Reliability (3)"), and a detailed description. The first issue is "Insert a <!DOCTYPE> declaration to before this <html> tag.", which is a "user-experience" issue assigned to "Consistency". Other issues include "Anchors must have content and the content must be accessible by a screen reader." under "accessibility" and "Maintainability" categories.

Duplicates



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

Conclusion:

In this experiment, we integrated Jenkins with SonarQube to enable automated code quality checks within our CI/CD pipeline. We started by deploying SonarQube using Docker, setting up

a project, and configuring it to analyze code quality. Next, we configured Jenkins by installing the SonarQube Scanner plugin, adding SonarQube server details, and setting up the scanner tool. We then developed a Jenkins pipeline to automate the process of cloning a GitHub repository and running SonarQube analysis on the code. This integration helps ensure continuous monitoring of code quality, detecting issues such as bugs, code smells, and security vulnerabilities throughout the development process.

Adv DevOps Exp 09

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

Theory:

What is Nagios?

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

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

Why We Need Nagios tool?

Here are the important reasons to use Nagios monitoring tool:

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

Features of Nagios

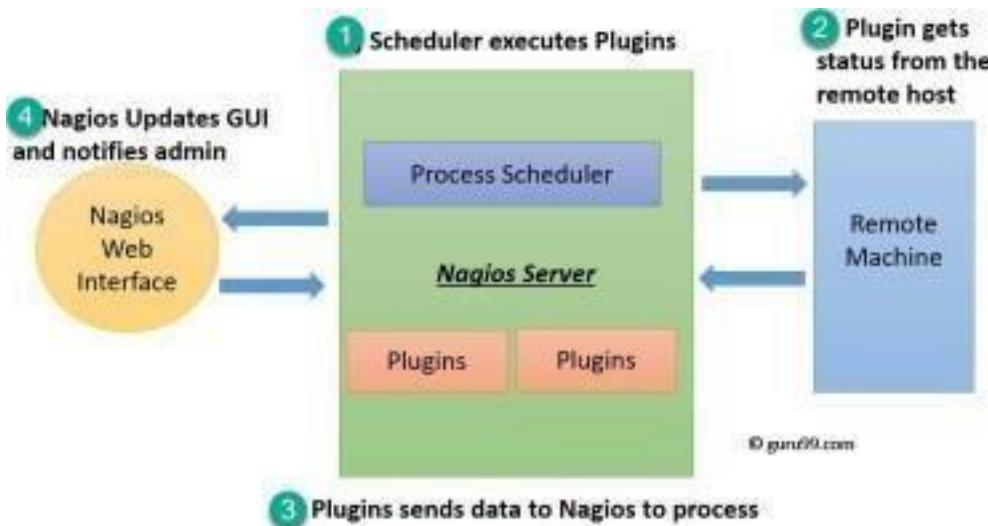
Following are the important features of Nagios monitoring tool:

- Relatively scalable, Manageable, and Secure
- Good log and database system
- Informative and attractive web interfaces
- Automatically send alerts if condition changes
- If the services are running fine, then there is no need to do check that host is alive
- Helps you to detect network errors or server crashes ● You can troubleshoot the performance issues of the server.
- The issues, if any, can be fixed automatically as they are identified during the monitoring process
- You can monitor the entire business process and IT infrastructure with a single pass

- The product's architecture is easy to write new plugins in the language of your choice
- Nagios allows you to read its configuration from an entire directory which helps you to decide how to define individual files
- Utilizes topology to determine dependencies
- Monitor network services like HTTP, SMTP, HTTP, SNMP, FTP, SSH, POP, etc.
- Helps you to define network host hierarchy using parent hosts
- Ability to define event handlers that runs during service or host events for proactive problem resolution
- Support for implementing redundant monitoring hosts

Nagios Architecture

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



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

Step 1: Create a security group with the required configurations I have created a new security group with a name 'newsecurity'

The screenshot shows the 'Create security group' wizard. In the 'Basic details' section, the 'Security group name' is set to 'newsecurity'. The 'Description' is 'made for exp09'. The 'VPC' dropdown is set to 'vpc-0aa3db8937df8678b'.

I have modified the INBOUND RULES as follows

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	Anywh...	/0 X
HTTPS	TCP	443	Anywh...	0.0.0.0/0 X
SSH	TCP	22	Anywh...	0.0.0.0/0 X
All ICMP - IPv6	IPv6 ICMP	All	Anywh...	/0 X
All ICMP - IPv4	ICMP	All	Anywh...	0.0.0.0/0 X
All traffic	All	All	Anywh...	0.0.0.0/0 X
Custom TCP	TCP	5666	Anywh...	0.0.0.0/0 X

Step 2: Create ec2 instance

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

Name and tags [Info](#)

Name
nagios-host [Add additional tags](#)

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

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

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

Recents [Quick Start](#)

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

Amazon Machine Images (AMI)

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*
abhinav [▼](#) [Create new key pair](#)

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

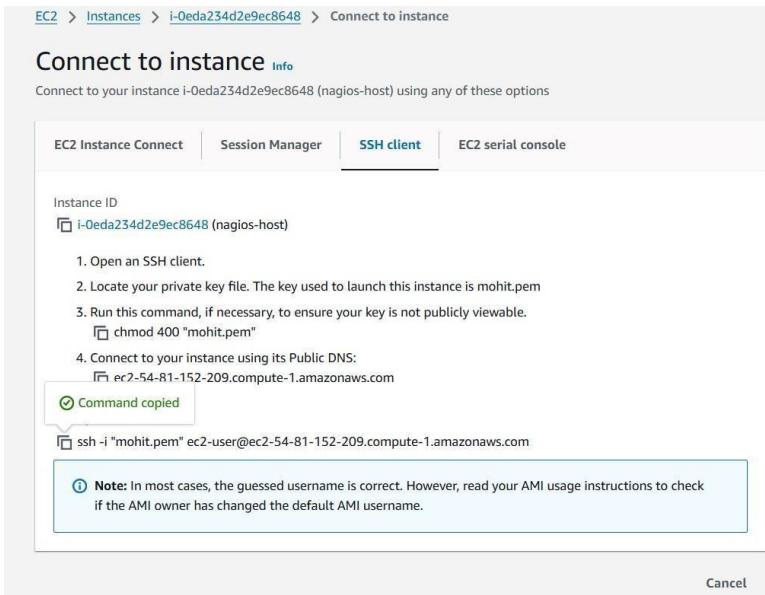
Common security groups [Info](#)

Select security groups [▼](#) [Compare security group rules](#)

newsecurity sg-05d7468fe3a2f7a8e X
VPC: vpc-0aa3db8937df8678b

Security groups that you add or remove here will be added to or removed from all your network interfaces.

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



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

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

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

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

And then run these commands

sudo yum update sudo yum install httpd php

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

Transaction Summary
Install 25 Packages

-----
```

Package	Architecture	Version	Repository	Size
Installing:				
httpd	x86_64	2.4.62.1.amzn2023	amazonlinux	48 k
php8_3	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux	10 k
Installing dependencies:				
apr	x86_64	1.7.2-2.amzn2023.0.2	amazonlinux	129 k
apr-util	x86_64	1.6.3-1.amzn2023.0.1	amazonlinux	98 k
gengetopt	x86_64	1.0.10-1.amzn2023.0.3	amazonlinux	19 k
httpd	x86_64	2.4.62.1.amzn2023	amazonlinux	1.4 M
httpd-filesystem	x86_64	2.4.62.1.amzn2023	amazonlinux	14 k
httpd-tools	x86_64	2.4.62.1.amzn2023	amazonlinux	81 k
libbrotli	x86_64	1.0.9-4.amzn2023.0.2	amazonlinux	315 k
libcurl	x86_64	1.0.19-4.amzn2023.0.1	amazonlinux	176 k
libedit	x86_64	1.11-1.amzn2023.0.2	amazonlinux	241 k
mod_wap	x86_64	2.1.49-3.amzn2023.0.3	amazonlinux	33 k
nginx-filesystem	x86_64	1:1.24.0-1.amzn2023.0.4	amazonlinux	9.8 k
php8_3-cli	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux	3.7 M
php8_3-common	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux	737 k
php8_3-process	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux	45 k
php8_3-zts	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux	154 k
Installing weak dependencies:				
apr-util-openssl	x86_64	1.6.3-1.amzn2023.0.1	amazonlinux	17 k
mod_http2	x86_64	2.0.27-1.amzn2023.0.3	amazonlinux	166 k
mod_imap	x86_64	2.4.62.1.amzn2023.0.1	amazonlinux	61 k
php8_3-fpm	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux	13.9 M
php8_3-mbstring	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux	528 k
php8_3-pcre	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux	379 k
php8_3-pdo	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux	89 k
php8_3-sodium	x86_64	8.3.10-1.amzn2023.0.1	amazonlinux	41 k

Transaction Summary

Install 25 Packages

sudo yum install gcc glibc glibc-common

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

-----
```

Package	Architecture	Version	Repository	Size
Installing:				
gcc	x86_64	11.4.1-2.amzn2023.0.2	amazonlinux	32 M
Installing dependencies:				
annobin-docs	noarch	10.93-1.amzn2023.0.1	amazonlinux	92 k
annobin-plugin-gcc	x86_64	10.93-1.amzn2023.0.1	amazonlinux	887 k
cpp	x86_64	11.4.1-2.amzn2023.0.2	amazonlinux	10 M
gc	x86_64	8.0.4-5.amzn2023.0.2	amazonlinux	105 k
glibc-devel	x86_64	2.34.52.amzn2023.0.11	amazonlinux	27 k
glibc-headers-x86	noarch	2.34.52.amzn2023.0.11	amazonlinux	427 k
guile22	x86_64	2.2.7-2.amzn2023.0.3	amazonlinux	6.4 M
kernel-headers	x86_64	6.1.109-118.189.amzn2023	amazonlinux	1.4 M
libmpc	x86_64	1.2.1-2.amzn2023.0.2	amazonlinux	62 k
libtool-ltdl	x86_64	2.4.7-1.amzn2023.0.3	amazonlinux	38 k
libcrypt-devel	x86_64	4.4.33-7.amzn2023	amazonlinux	32 k
make	x86_64	1:4.3-5.amzn2023.0.2	amazonlinux	534 k

Transaction Summary

Install 13 Packages

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

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

sudo yum install gd gd-devel

Package	Architecture	Version	Repository	Size
Installing:				
gd	x86_64	2.3.3-5.amzn2023.0.3	amazonlinux	139 k
gd-devel	x86_64	2.3.3-5.amzn2023.0.3	amazonlinux	38 k
Installing dependencies:				
brotli	x86_64	1.0.9-4.amzn2023.0.2	amazonlinux	314 k
brotli-devel	x86_64	1.0.9-4.amzn2023.0.2	amazonlinux	31 k
bzip2-devel	x86_64	1.0.8-6.amzn2023.0.2	amazonlinux	214 k
cairo	x86_64	1.17.6-2.amzn2023.0.1	amazonlinux	684 k
cmake-filesystem	x86_64	3.22.2-1.amzn2023.0.4	amazonlinux	16 k
fontconfig	x86_64	2.13.94-2.amzn2023.0.2	amazonlinux	273 k
fontconfig-devel	x86_64	2.13.94-2.amzn2023.0.2	amazonlinux	128 k
fonts-filesystem	noarch	1:12.0.5-12.amzn2023.0.2	amazonlinux	9.5 k
freetype	x86_64	2.13.2-5.amzn2023.0.1	amazonlinux	423 k
freetype-devel	x86_64	2.13.2-5.amzn2023.0.1	amazonlinux	912 k
glib2-devel	x86_64	2.74.7-689.amzn2023.0.2	amazonlinux	486 k
google-noto-fonts--common	noarch	20301206-2.amzn2023.0.2	amazonlinux	15 k
google-noto-sans-vf-fonts	noarch	20201206-2.amzn2023.0.2	amazonlinux	492 k
graphite2	x86_64	1.3.14-7.amzn2023.0.2	amazonlinux	97 k
graphite2-devel	x86_64	1.3.14-7.amzn2023.0.2	amazonlinux	21 k
harfbuzz	x86_64	7.0.0-2.amzn2023.0.1	amazonlinux	868 k
harfbuzz-devel	x86_64	7.0.0-2.amzn2023.0.1	amazonlinux	404 k
harfbuzz-icu	x86_64	7.0.0-2.amzn2023.0.1	amazonlinux	18 k
jbigkit-libs	x86_64	2.1-21.amzn2023.0.2	amazonlinux	54 k
langpacks-core-font-en	noarch	3.0-21.amzn2023.0.4	amazonlinux	10 k
libICE	x86_64	1.0.10-6.amzn2023.0.2	amazonlinux	71 k
libSM	x86_64	1.2.3-8.amzn2023.0.2	amazonlinux	42 k
libX11	x86_64	1.7.2-2.amzn2023.0.4	amazonlinux	657 k
libX11-common	noarch	1.7.2-3.amzn2023.0.4	amazonlinux	152 k
libX11-devel	x86_64	1.7.2-3.amzn2023.0.4	amazonlinux	939 k
libX11-xcb	x86_64	1.7.2-3.amzn2023.0.4	amazonlinux	12 k
libXau	x86_64	1.0.9-6.amzn2023.0.2	amazonlinux	31 k
libXau-devel	x86_64	1.0.9-6.amzn2023.0.2	amazonlinux	14 k
libXext	x86_64	1.3.4-6.amzn2023.0.2	amazonlinux	41 k
libXpm	x86_64	3.5.15-2.amzn2023.0.3	amazonlinux	65 k
libXpm-devel	x86_64	3.5.15-2.amzn2023.0.3	amazonlinux	59 k
libXrender	x86_64	0.9.10-14.amzn2023.0.2	amazonlinux	28 k
libXt	x86_64	1.2.0-6.amzn2023.0.2	amazonlinux	181 k
libblkid-devel	x86_64	2.37.4-1.amzn2023.0.4	amazonlinux	15 k

Create a new Nagios User with its password. You'll have to enter the password twice for confirmation. **sudo adduser -m nagios sudo passwd nagios**

```
[ec2-user@ip-172-31-41-160 ~]$ sudo adduser -m nagios
[ec2-user@ip-172-31-41-160 ~]$ sudo passwd nagios
Changing password for user nagios.
```

New password:

Retype new password:

passwd: all authentication tokens updated successfully.

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

Create a new user group & create a new directory for Nagios downloads using the following commands **sudo groupadd nagcmd sudo usermod -a -G nagcmd nagios sudo usermod -a -G nagcmd apache mkdir ~/downloads cd ~/downloads**

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

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

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

```
[ec2-user@ip-172-31-41-160:~/downloads/nagios-4.5.5]
[ec2-user@ip-172-31-41-160 downloads]$ wget https://sourceforge.net/projects/nagios/files/latest/download
--2024-10-02 12:34:21- https://sourceforge.net/projects/nagios/files/latest/download
resolving sourceforge.net (sourceforge.net)... 172.64.150.145, 104.18.37.111, 2606:4700:4400::6812:256f, ...
Connecting to sourceforge.net (sourceforge.net)|172.64.150.145|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://downloads.sourceforge.net/project/nagios/nagios-4.x/nagios-4.5.5/nagios-4.5.5.tar.gz?ts=gAAAAABm_T3NMNSZPzP-6la2Tltvo06CG7VV7QGVH08n3tC24QehfMw7vHcoKbGhg2iIRxbmfugII0LccN
rxta0Ixg3j2KGwA30K3&use_mirror=phoenixnap&r=
--2024-10-02 12:34:21- https://downloads.sourceforge.net/project/nagios/nagios-4.x/nagios-4.5.5/nagios-4.5.5.tar.gz?ts=gAAAAABm_T3NMNSZPzP-6la2Tltvo06CG7VV7QGVH08n3tC24QehfMw7vHcoKbGhg2i
rx0m0fug10LccNrx0aIxg3j2KGwA30K3&use_mirror=phoenixnap&r=
Resolving downloads.sourceforge.net (downloads.sourceforge.net)... 204.68.111.105
Connecting to downloads.sourceforge.net (downloads.sourceforge.net)|204.68.111.105|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2065473 (2.0M) [application/x-gzip]
Saving to: 'download'

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

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

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

```
[ec2-user@ip-172-31-41-160:~/downloads/nagios-4.5.5]
[ec2-user@ip-172-31-41-160 downloads]$ wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
--2024-10-02 12:34:46- https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251
Connecting to nagios-plugins.org (nagios-plugins.org)|45.56.123.251|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2753049 (2.6M) [application/x-gzip]
Saving to: 'nagios-plugins-2.4.11.tar.gz'

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

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

[ec2-user@ip-172-31-92-249:~/downloads]
[ec2-user@ip-172-31-92-249 ~]$ cd ~/downloads
[ec2-user@ip-172-31-92-249 downloads]$ wget https://sourceforge.net/projects/nagios/files/latest/download
--2024-09-30 09:54:56- https://sourceforge.net/projects/nagios/files/latest/download
Resolving sourceforge.net (sourceforge.net)... 172.64.150.145, 104.18.37.111, 2606:4700:4400::6812:256f, ...
Connecting to sourceforge.net (sourceforge.net)|172.64.150.145|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://downloads.sourceforge.net/project/nagios/nagios-4.x/nagios-4.5.5/nagios-4.5.5.tar.gz?ts=gAAAAABm-nVw9RdAvnMaShLf3gu4RXT5VxrTz6fGxJvhVa0zPBIbPgbyzLMcDDAALgtEc1pKr9cgJNj2b
kKtar1iCj0Tfkgx30K3&use_mirror=netactuate&r=
--2024-09-30 09:54:56- https://downloads.sourceforge.net/project/nagios/nagios-4.x/nagios-4.5.5/nagios-4.5.5.tar.gz?ts=gAAAAABm-nVw9RdAvnMaShLf3gu4RXT5VxrTz6fGxJvhVa0zPBIbPgbyzLMcDDAALgtE
C1pOKr0cgJNj2bKtar1iCj0Tfkgx30K3&use_mirror=netactuate&r=
Resolving downloads.sourceforge.net (downloads.sourceforge.net)... 204.68.111.105
Connecting to downloads.sourceforge.net (downloads.sourceforge.net)|204.68.111.105|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://netactuate.dl.sourceforge.net/project/nagios/nagios-4.x/nagios-4.5.5/nagios-4.5.5.tar.gz?ts=gAAAAABm-nVw9RdAvnMaShLf3gu4RXT5VxrTz6fGxJvhVa0zPBIbPgbyzLMcDDAALgtE
--2024-09-30 09:54:57- https://netactuate.dl.sourceforge.net/project/nagios/nagios-4.x/nagios-4.5.5/nagios-4.5.5.tar.gz?ts=gAAAAABm-nVw9RdAvnMaShLf3gu4RXT5VxrTz6fGxJvhVa0zPBIbPgbyzLMcDDAALgtE
resolving netactuate.dl.sourceforge.net (netactuate.dl.sourceforge.net)... 104.225.3.66
Connecting to netactuate.dl.sourceforge.net (netactuate.dl.sourceforge.net)|104.225.3.66|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2065473 (2.0M) [application/x-gzip]
Saving to: 'download'

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

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

[ec2-user@ip-172-31-92-249 downloads]$ wget https://nagios-plugins.org/download/nagios-plugins-2.4.9.tar.gz
--2024-09-30 09:56:53- https://nagios-plugins.org/download/nagios-plugins-2.4.9.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251
Connecting to nagios-plugins.org (nagios-plugins.org)|45.56.123.251|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2754403 (2.6M) [application/x-gzip]
Saving to: 'nagios-plugins-2.4.9.tar.gz'

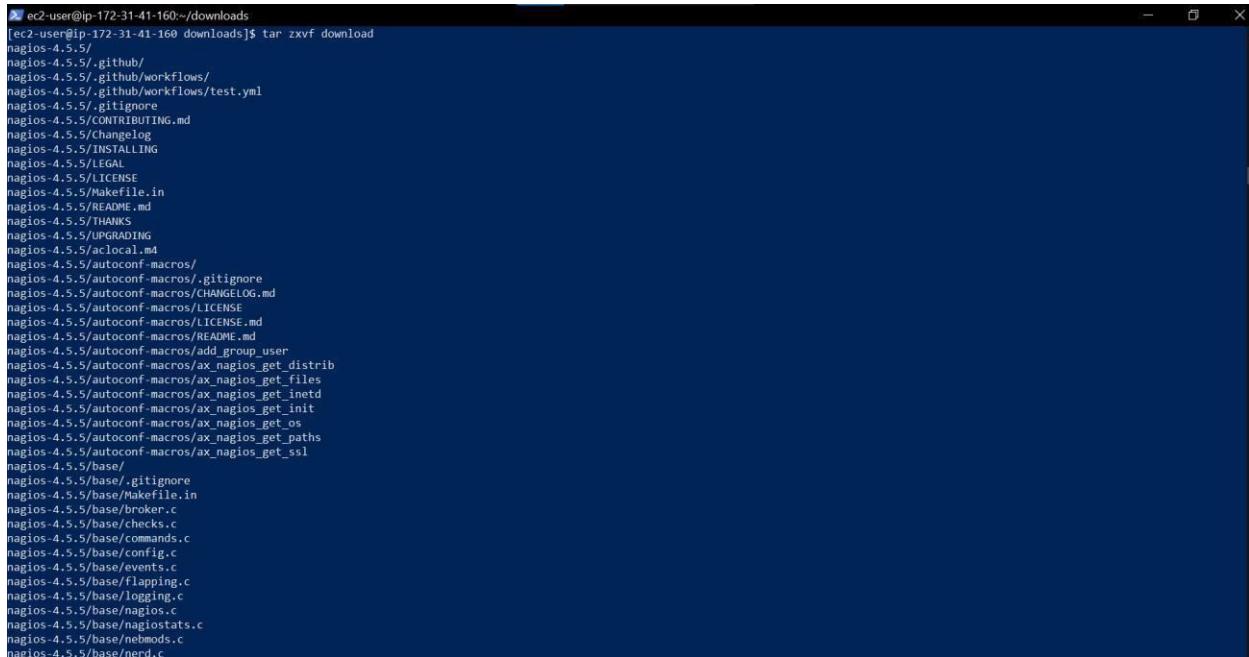
nagios-plugins-2.4.9.tar.gz 100%[=====] 2.63M 7.54MB/s in 0.3s

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

Now, we run the next command in the following manner tar zxvf

<nagios4.5.5 version> (for me it has gotten saved as 'download') So i wrote

tar zxvf download



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

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

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

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

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

```
[ec2-user@ip-172-31-41-160:~/downloads/nagios-4.5.5]$ cd nagios-4.5.5
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$ ./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64_pc-linux-gnu
checking host system type... x86_64_pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking 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
checking for strings.h... yes
checking for sys/stat.h... yes
checking for sys/types.h... yes
checking for unistd.h... yes
checking for arpa/inet.h... yes
checking for ctype.h... yes
Checking for dirent.h... yes
Checking for errno.h... yes
Checking for fcntl.h... yes
Checking for getopt.h... yes
Checking for grp.h... yes
Checking for libgen.h... yes
Checking for limits.h... yes
Checking for math.h... yes
Checking for netdb.h... yes
Checking for netinet/in.h... yes
Checking for pwd.h... yes
Checking for regex.h... yes
Checking for signal.h... yes
Checking for socket.h... no
Checking for stdarg.h... yes

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

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

For that purpose, we are to run the following command. **`sudo yum install openssl-devel`** (for ssl header)

```
[ec2-user@ip-172-31-41-160 nagios-4.5.5]$ sudo yum install openssl-devel
Last metadata expiration check: 0:12:11 ago on Wed Oct  2 12:28:33 2024.
Dependencies resolved.
=====
Package           Architecture      Version       Repository   Size
=====
Installing:
openssl-devel    x86_64          1:3.0.8-1.amzn2023.0.14   amazonlinux  3.0 M

Transaction Summary
=====
Install 1 Package

Total download size: 3.0 M
Installed size: 4.7 M
Is this ok [y/N]: y
Downloading Packages:
openssl-devel-3.0.8-1.amzn2023.0.14.x86_64.rpm          26 MB/s | 3.0 MB  00:00
Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing           :                                                 1/1
  Installing openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64 : 1/1
  Running scriptlet: openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64 : 1/1
  Verifying           : openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64 : 1/1

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

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

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

After this, run **make all** command

```
[ec2-user@ip-172-31-41-160:~/downloads/nagios-4.5.5]$ make all
cd ./hadoop && make

make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
gcc -Wall -I... -I... -I... -I.../include -I.../include -I... -g -O2 -DHAVE_CONFIG_H -DSNSCORE -c -o nagios.o ./nagios.c
gcc -Wall -I... -I... -I... -I.../include -I.../include -I... -g -O2 -DHAVE_CONFIG_H -DSNSCORE -c -o broker.o broker.c
gcc -Wall -I... -I... -I... -I.../include -I.../include -I... -g -O2 -DHAVE_CONFIG_H -DSNSCORE -c -o nebmods.o nebmods.c
gcc -Wall -I... -I... -I... -I.../include -I.../include -I... -g -O2 -DHAVE_CONFIG_H -DSNSCORE -c -o ../common/shared.o ./common/shared.c
gcc -Wall -I... -I... -I... -I.../include -I.../include -I... -g -O2 -DHAVE_CONFIG_H -DSNSCORE -c -o query-handler.o query-handler.c
gcc -Wall -I... -I... -I... -I.../include -I.../include -I... -g -O2 -DHAVE_CONFIG_H -DSNSCORE -c -o workers.o workers.c
In function `get_worker_list',
  inlined from `get_worker' at workers.c:277:12:
workers.c:253:17: warning: %s' direct argument is null [Wformat-overflow=]
  253 |     log_debug_info(DEBUG_CHECKS, 1, "Found specialized worker(s) for '%s'", (slash && *slash != '/') ? slash : cmd_name);
                                                               ^~~~~~
gcc -Wall -I... -I... -I.../lib -I.../include -I.../include -I... -g -O2 -DHAVE_CONFIG_H -DSNSCORE -c -o checks.o checks.c
gcc -Wall -I... -I... -I.../lib -I.../include -I.../include -I... -g -O2 -DHAVE_CONFIG_H -DSNSCORE -c -o config.o config.c
gcc -Wall -I... -I... -I.../lib -I.../include -I.../include -I... -g -O2 -DHAVE_CONFIG_H -DSNSCORE -c -o commands.o commands.c
gcc -Wall -I... -I... -I.../lib -I.../include -I.../include -I... -g -O2 -DHAVE_CONFIG_H -DSNSCORE -c -o events.o events.c
gcc -Wall -I... -I... -I.../lib -I.../include -I.../include -I... -g -O2 -DHAVE_CONFIG_H -DSNSCORE -c -o flapping.o flapping.c
gcc -Wall -I... -I... -I.../lib -I.../include -I.../include -I... -g -O2 -DHAVE_CONFIG_H -DSNSCORE -c -o logging.o logging.c
gcc -Wall -I... -I... -I.../lib -I.../include -I.../include -I... -g -O2 -DHAVE_CONFIG_H -DSNSCORE -c -o macros-base.o ./common/macros.c
gcc -Wall -I... -I... -I.../lib -I.../include -I.../include -I... -g -O2 -DHAVE_CONFIG_H -DSNSCORE -c -o netutils.o netutils.c
gcc -Wall -I... -I... -I.../lib -I.../include -I.../include -I... -g -O2 -DHAVE_CONFIG_H -DSNSCORE -c -o notifications.o notifications.c
gcc -Wall -I... -I... -I.../lib -I.../include -I.../include -I... -g -O2 -DHAVE_CONFIG_H -DSNSCORE -c -o sehandles.o sehandles.c
gcc -Wall -I... -I... -I.../lib -I.../include -I.../include -I... -g -O2 -DHAVE_CONFIG_H -DSNSCORE -c -o utils.o utils.c
gcc -Wall -I... -I... -I.../lib -I.../include -I.../include -I... -g -O2 -DHAVE_CONFIG_H -DSNSCORE -c -o retention-base.o ./retention.c
gcc -Wall -I... -I... -I.../lib -I.../include -I.../include -I... -g -O2 -DHAVE_CONFIG_H -DSNSCORE -c -o xretention-base.o ./xdata/xrddefault.c
gcc -Wall -I... -I... -I.../lib -I.../include -I.../include -I... -g -O2 -DHAVE_CONFIG_H -DSNSCORE -c -o comments-base.o ./common/comments.c
gcc -Wall -I... -I... -I.../lib -I.../include -I.../include -I... -g -O2 -DHAVE_CONFIG_H -DSNSCORE -c -o xcomments-base.o ./xdata/xcddefault.c
gcc -Wall -I... -I... -I.../lib -I.../include -I.../include -I... -g -O2 -DHAVE_CONFIG_H -DSNSCORE -c -o objects-base.o ./common/objects.c
gcc -Wall -I... -I... -I.../lib -I.../include -I.../include -I... -g -O2 -DHAVE_CONFIG_H -DSNSCORE -c -o xobjects-base.o ./xdata/xoddefault.c
gcc -Wall -I... -I... -I.../lib -I.../include -I.../include -I... -g -O2 -DHAVE_CONFIG_H -DSNSCORE -c -o xstatusdata-base.o ./common/xstatusdata.c
gcc -Wall -I... -I... -I.../lib -I.../include -I.../include -I... -g -O2 -DHAVE_CONFIG_H -DSNSCORE -c -o xstatusdata-base.o ./xdata/xstatusdata.c
gcc -Wall -I... -I... -I.../lib -I.../include -I.../include -I... -g -O2 -DHAVE_CONFIG_H -DSNSCORE -c -o perfdata-base.o ./perfdata.c
gcc -Wall -I... -I... -I.../lib -I.../include -I.../include -I... -g -O2 -DHAVE_CONFIG_H -DSNSCORE -c -o xperfdata-base.o ./xdata/xpddefault.c
gcc -Wall -I... -I... -I.../lib -I.../include -I.../include -I... -g -O2 -DHAVE_CONFIG_H -DSNSCORE -c -o downtime-base.o ./common/downtime.c
make -C ./lib

make[2]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/lib'
gcc -Wall -g -O2 -I... -I.../include -DHAVE_CONFIG_H -c queue.c -o queue.o
gcc -Wall -g -O2 -I... -I.../include -DHAVE_CONFIG_H -c kvvec.c -o kvvec.o
gcc -Wall -g -O2 -I... -I.../include -DHAVE_CONFIG_H -c iocache.c -o iocache.o
gcc -Wall -g -O2 -I... -I.../include -DHAVE_CONFIG_H -c iobroker.c -o iobroker.o
gcc -Wall -g -O2 -I... -I.../include -DHAVE_CONFIG_H -c bitmap.c -o bitmap.o
gcc -Wall -g -O2 -I... -I.../include -DHAVE_CONFIG_H -c dkhash.c -o dkhash.o

[ec2-user@ip-172-31-41-160:~/downloads/nagios-4.5.5]$ make install-webconf
  on doing this. Pay particular attention to the docs on
  object configuration files, as they determine what/how
  things get monitored

  make install-webconf
    - This installs the Apache config file for the Nagios
      web interface

  make install-exfoliation
    - This installs the Exfoliation theme for the Nagios
      web interface

  make install-classicui
    - This installs the classic theme for the Nagios
      web interface

*** Support Notes ***

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

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

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

*****
```

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

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

sudo make install-init

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

sudo make install-config

```
ec2-user@ip-172-31-41-160:~/downloads/nagios-4.5.5$ sudo make install-config
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc/objects
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/nagios.cfg /usr/local/nagios/etc/nagios.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/cgi.cgi /usr/local/nagios/etc/cgi.cgi
/usr/bin/install -c -b -m 660 -o nagios -g nagios sample-config/resource.cfg /usr/local/nagios/etc/resource.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/templates.cfg /usr/local/nagios/etc/objects/templates.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/commands.cfg /usr/local/nagios/etc/objects/commands.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/contacts.cfg /usr/local/nagios/etc/objects/contacts.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/timeperiods.cfg /usr/local/nagios/etc/objects/timeperiods.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/localhost.cfg /usr/local/nagios/etc/objects/localhost.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/windows.cfg /usr/local/nagios/etc/objects/windows.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/printer.cfg /usr/local/nagios/etc/objects/printer.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/switch.cfg /usr/local/nagios/etc/objects/switch.cfg

*** Config files installed ***

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

sudo make install-webconf

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

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

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

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

Specify the password twice. **sudo htpasswd -c**

/usr/local/nagios/etc/htpasswd.users nagiosadmin

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

Restart Apache **sudo service httpd**

restart

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

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

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

Compile and install plugins **cd nagios-plugins-2.4.11**
./configure --with-nagios-user=nagios --with-
nagios-group=nagios

Run the following command: **sudo chkconfig --add nagios** On running the above command

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

If this is the output that one is getting, then it means that the init script is missing... We can check this by running **ls /etc/init.d/**

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

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

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

Run the following command: **sudo nano /etc/init.d/nagios**

Within this file, paste the following script

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

```
$0 start
;;
status)
ps aux | grep nagios
;;
*)
echo "Usage: $0 {start|stop|restart|status}"
exit 1
;;
esac exit
0
```

```
ec2-user@ip-172-31-92-249:~/downloads/nagios-plugins-2.4.11
GNU nano 5.8
#!/bin/bash
# nagios      Startup script for Nagios
#
# chkconfig: 345 99 10
# description: Nagios is a host/service/network monitoring program
# processname: nagios
# pidfile: /var/run/nagios/nagios.pid
#
case "$1" in
    start)
        "Starting Nagios..."
        /usr/local/nagios/bin/nagios /usr/local/nagios/etc/nagios.cfg
        ;;
    stop)
        "Stopping Nagios..."
        kill -9 `cat /var/run/nagios/nagios.pid`
        ;;
    restart)
        $0 stop
        $0 start
        ;;
    status)
        ps aux | grep nagios
        ;;
    *)
        echo "Usage: $0 {start|stop|restart|status}"
        exit 1
        ;;
esac

exit 0
```

Make the Script Executable: After saving the file, run the following command to make it executable: **sudo chmod +x /etc/init.d/nagios**

Run **sudo chkconfig --add nagios** again And then run **sudo chkconfig nagios on**

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

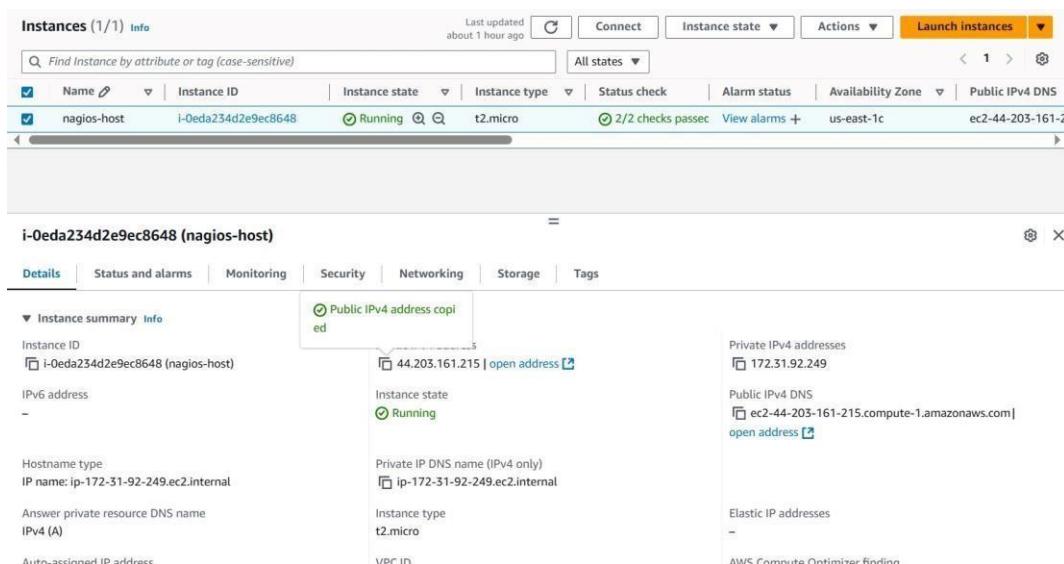
sudo service nagios start

```
[ec2-user@ip-172-31-92-249 nagios-plugins-2.4.11]$ sudo service nagios start
Starting Nagios...

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

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

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



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

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

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

The screenshot shows the Nagios Core 4.5.5 web interface. At the top, there's a header bar with a 'Not secure' warning and the URL '34.229.45.75/nagios/'. Below the header is the Nagios logo and a message 'Process running with PID 62668'. The main content area has several sections:

- General**: Home, Documentation.
- Current Status**: Tactical Overview, Map, Hosts, Services, Host Groups (Summary, Grid), Service Groups (Summary, Grid), Problems (Services (Unhandled), Hosts (Unhandled), Network Outages), Quick Search.
- Reports**: Availability, Trends, Alerts (History, Summary, Histogram, Notifications, Event Log).

Get Started section lists:

- 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 section lists:

- 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 and **Don't Miss...** sections are also present.

Conclusion:

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

Adv DevOps Exp 10

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

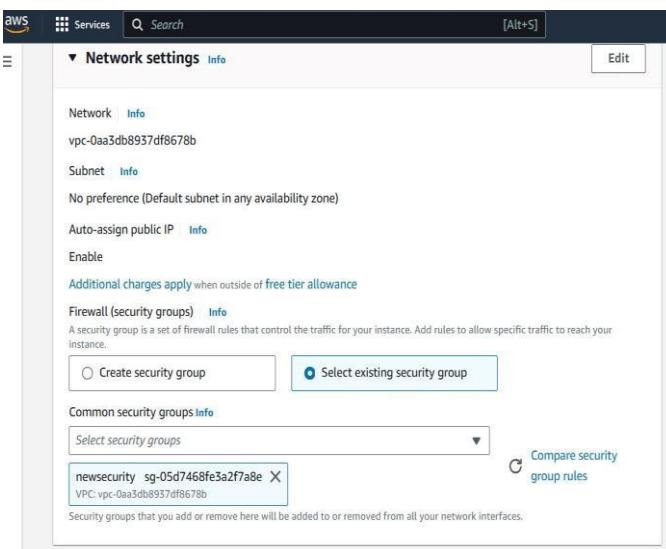
Monitoring Using Nagios:

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

```
ec2-user@ip-172-31-41-160:~/downloads/nagios-plugins-2.4.11$ sudo systemctl status
● ip-172-31-41-160.ec2.internal
    State: running
      Units: 296 loaded (incl. loaded aliases)
        Jobs: 0 queued
       Failed: 0 units
     Since: Wed 2024-10-02 12:28:05 UTC; 33min ago
    Systemd: 252.23-2.amzn2023
   CGroup: /
           ├─init.scope
           └─1 /usr/lib/systemd/systemd --switched-root --system --deserialize=32
   system.slice
   ├─acpid.service
   │   ├─1938 /usr/bin/systemd-inhibit --what=handle-suspend-key:handle-hibernate-key --who=noah "--why=acpid instead" --mode=block /usr/sbin/acpid -f
   │   └─2059 /usr/sbin/acpid -f
   ├─amazon-ssm-agent.service
   │   └─2141 /usr/bin/amazon-ssm-agent
   ├─atd.service
   │   └─2152 /usr/sbin/atd -f
   ├─auditd.service
   │   └─1768 /sbin/auditd
   ├─chronyd.service
   │   └─2175 /usr/sbin/chronyd -F 2
   ├─dbus-broker.service
   │   └─1944 /usr/bin/dbus-broker-launch --scope system --audit
   │       └─1944 dbus-broker --log 4 --controller 9 --machine-id ec2e4d759a3e2f6fe850b14e4cdacabe --max-bytes 536870912 --max-fds 4096 --max-matches 16384 --audit
   ├─gssproxy.service
   │   └─1959 /usr/sbin/gssproxy -D
   ├─httpd.service
   │   ├─49553 /usr/sbin/httpd -DFOREGROUND
   │   ├─49555 /usr/sbin/httpd -DFOREGROUND
   │   ├─49556 /usr/sbin/httpd -DFOREGROUND
   │   ├─49557 /usr/sbin/httpd -DFOREGROUND
   │   ├─49558 /usr/sbin/httpd -DFOREGROUND
   │   └─62800 /usr/sbin/httpd -DFOREGROUND
   └─libstoragemgmt.service
       └─1940 /usr/bin/lsmmd -d
```

Step 2: Before we begin,

To monitor a Linux machine, create an **Ubuntu 20.04 server** EC2 Instance in AWS. Provide it with the **same security group** as the Nagios Host and name it 'nagios-client' alongside the host.



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*

abhinav

Create new key pair

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
nagios-host	i-03facef442a77494d	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-34-229-45-75
nagios-client	i-0b934b61f21351c1b	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-54-172-92-22

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

In the nagios-host terminal, run this command **ps -ef | grep nagios**

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

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

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

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

Copy the sample localhost.cfg file to linuxhost folder. Use the following mentioned command to achieve it **cp /usr/local/nagios/etc/objects/localhost.cfg**

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

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

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

Changes to be made:

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

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

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

    host_name     linux-server
    alias         localhost
    address       54.172.92.226
}

#####

# Define an optional hostgroup for Linux machines

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

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

Open the Nagios Config file and add the following line **nano**

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

Add the following line in the file and save **cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/**

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

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

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

Verify the configuration files by running the following command

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

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

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

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

Running pre-flight check on configuration data...

Checking objects...
  Checked 16 services.
  Checked 2 hosts.
  Checked 2 host groups.
  Checked 0 service groups.
  Checked 1 contacts.
  Checked 1 contact groups.
  Checked 24 commands.
  Checked 5 time periods.
  Checked 0 host escalations.
  Checked 0 service escalations.
Checking for circular paths...
  Checked 2 hosts
  Checked 0 service dependencies
  Checked 0 host dependencies
  Checked 5 timperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
[root@ip-172-31-41-160 nagios-plugins-2.4.11]#
```

You are good to go if there are no errors.

Restart the nagios service service nagios

restart

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

```
[root@ip-172-31-41-160/tmp:nagios-plugins-2.4.11]
[root@ip-172-31-41-160 nagios-plugins-2.4.11]# sudo systemctl restart nagios
[root@ip-172-31-41-160 nagios-plugins-2.4.11]# sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
   Active: active (running) since Wed 2024-10-02 13:20:17 UTC; 7s ago
     Docs: https://www.nagios.org/documentation
  Process: 78777 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
  Process: 78777 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 78778 (nagios)
   Tasks: 6 (limit: 1112)
    Memory: 4.0M
      CPU: 2.4ms
     CGroup: /system.slice/nagios.service
             └─78778 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 02 13:20:17 ip-172-31-41-160.ec2.internal nagios[78778]: qh: echo service query handler registered
Oct 02 13:20:17 ip-172-31-41-160.ec2.internal nagios[78778]: qh: help for the query handler registered
Oct 02 13:20:17 ip-172-31-41-160.ec2.internal nagios[78778]: wproc: Successfully registered manager as @wproc with query handler
Oct 02 13:20:17 ip-172-31-41-160.ec2.internal nagios[78778]: wproc: Registry request: name=Core Worker 78782;pid=78782
Oct 02 13:20:17 ip-172-31-41-160.ec2.internal nagios[78778]: wproc: Registry request: name=Core worker 78781;pid=78781
Oct 02 13:20:17 ip-172-31-41-160.ec2.internal nagios[78778]: wproc: Registry request: name=Core Worker 78780;pid=78780
Oct 02 13:20:17 ip-172-31-41-160.ec2.internal nagios[78778]: wproc: Registry request: name=Core worker 78779;pid=78779
Oct 02 13:20:17 ip-172-31-41-160.ec2.internal nagios[78778]: Successfully launched command file worker with pid 78783
Oct 02 13:20:21 ip-172-31-41-160.ec2.internal nagios[78778]: HOST ALERT: linux-server|UP;SOFT|1;PING OK - Packet loss = 0%, RTA = 0.93 ms
Oct 02 13:20:24 ip-172-31-41-160.ec2.internal nagios[78778]: SERVICE ALERT: localhost|HTTP;WARNING;HARD|4;HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.000 seconds
[root@ip-172-31-41-160 nagios-plugins-2.4.11]# sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Drop-In: /usr/lib/systemd/system/httpd.service.d
             └─49553 (httpd)
     Active: active (running) since Wed 2024-10-02 12:47:56 UTC; 33min ago
       Docs: man:httpd.service(8)
     Main PID: 49553 (httpd)
        Status: "Total requests: 26; Idle/Busy workers 100/0;Requests/sec: 0.0129; Bytes served/sec: 94 B/sec"
        Tasks: 230 (limit: 1112)
      Memory: 21.7M
        CPU: 1.416s
       CGroup: /system.slice/httpd.service
               └─49553 /usr/sbin/httpd -DFOREGROUND
```

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

Now it is time to switch to the client machine.

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

```
PS C:\WINDOWS\system32> cd C:\Users\DELL\Downloads
PS C:\Users\DELL\Downloads> ssh -i "mohit.pem" ubuntu@ec2-54-172-92-226.compute-1.amazonaws.com
The authenticity of host 'ec2-54-172-92-226.compute-1.amazonaws.com (54.172.92.226)' can't be established.
ECDSA key fingerprint is SHA256:e/WkFQRuHSqPjqQ5hDMAoAdku8msNhETN9SAgzEy53E.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-172-92-226.compute-1.amazonaws.com,54.172.92.226' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1016-aws x86_64)

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

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

 System load: 0.0          Processes:      104
 Usage of /: 22.8% of 6.71GB  Users logged in:   0
 Memory usage: 20%
 Swap usage:  0%
* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

 https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

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

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

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

Name:Aditya Ahuja

Div: D15C

Roll No: 02

install -y nagios-nrpe-server nagios-plugins

```
ubuntu@ip-172-31-36-100:~$ sudo apt update -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [300 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [83.1 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4576 B]
Get:9 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [126 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [116 kB]
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [8632 kB]
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [10.4 kB]
Get:14 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [10.9 kB]
Get:15 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [2888 B]
Get:16 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 kB]
Get:17 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [344 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [260 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [10.8 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [9328 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [535 kB]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [130 kB]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [6762 kB]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [380 kB]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [157 kB]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [45.0 kB]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [14.9 kB]
Get:31 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [14.4 kB]
Get:32 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [1608 kB]
Get:33 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [12.2 kB]
Get:34 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [532 kB]
Get:35 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [388 kB]
Get:36 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 kB]
Get:37 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [10.6 kB]
Get:38 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [10.8 kB]
Get:39 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [17.6 kB]
Get:40 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1104 kB]
Get:41 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 kB]
Get:42 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 kB]
Get:43 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 kB]
```

```
ubuntu@ip-172-31-36-100:~$ sudo apt install gcc -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86_64-linux-gnu cpp cpp-13 cpp-13-x86_64-linux-gnu cpp-x86_64-linux-gnu fontconfig-config fonts-dejavu-core fonts-dejavu-mono gcc-13 gcc-13-base
  gcc-13-x86_64-linux-gnu gcc-x86_64-linux-gnu libaios3 libaiotecni1 libbinbufs1 libbc-dev liblbc1-dev liblbc1-nobfd0 liblbcf0 liblde265-0
  libldeflate libfontconfig1 liblbcg-13-dev liblbgd3 liblgbomp1 liblgbpfgn liblgbf-plugin-aomdec liblgbf-plugin-aomenc liblgbf-plugin-lbde265 liblhwasano liblis123 liblmtm libjbjig0
  libjpeg-turbo libljpeg liblberc3 liblbgpc3 liblquadmath0 liblframe1 liblsharpyuv libliff6 libltsam2 liblwsan1 liblwebp7 liblwpdm0 linux-libc-dev manpages-dev rpcsvc-proto
  liblhelpf-tools liblhelpf-plugin-fmpqdec liblhelpf-plugin-jmpdec liblhelpf-plugin-jkdec liblhelpf-plugin-jkenc liblhelpf-plugin-ravie
  liblhelpf-plugin-svtenc
The following NEW packages will be installed:
  binutils binutils-common binutils-x86_64-linux-gnu cpp cpp-13 cpp-13-x86_64-linux-gnu fontconfig-config fonts-dejavu-core fonts-dejavu-mono gcc-13 gcc-13-base
  gcc-13-x86_64-linux-gnu gcc-x86_64-linux-gnu libaios3 libaiotecni1 libbinbufs1 libbc-dev liblbc1-dev liblbc1-nobfd0 liblbcf0 liblde265-0
  libldeflate libfontconfig1 liblbcg-13-dev liblbgd3 liblgbomp1 liblgbpfgn liblgbf-plugin-aomdec liblgbf-plugin-aomenc liblgbf-plugin-lbde265 liblhwasano liblis123 liblmtm libjbjig0
  libjpeg-turbo libljpeg liblberc3 liblbgpc3 liblquadmath0 liblframe1 liblsharpyuv libliff6 libltsam2 liblwsan1 liblwebp7 liblwpdm4 linux-libc-dev manpages-dev rpcsvc-proto
0 upgraded, 57 newly installed, 0 to remove and 0 not upgraded.
Need to get 62.8 kB of archives.
After this operation, 222 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 binutils-common amd64 2.42-4ubuntu2 [239 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libbsframe1 amd64 2.42-4ubuntu2 [14.8 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblbinbufs1 amd64 2.42-4ubuntu2 [572 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblbc1-dev liblbc1-nobfd0 amd64 2.42-4ubuntu2 [97.1 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblbc1-nobfd0 liblbcf0 liblde265-0 [50.9 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblbcg-13-dev liblbgd3 liblgbomp1 liblgbpfgn liblgbf-plugin-aomdec liblgbf-plugin-aomenc liblgbf-plugin-lbde265 liblhwasano liblis123 liblmtm libjbjig0
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblbinbufs1-x86_64-linux-gnu amd64 2.42-4ubuntu2 [2469 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblbc1-base amd64 13.2.0-23ubuntu2 [18.0 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblbc1-base amd64 13.2.0-23ubuntu2 [49.0 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblbinbufs1 amd64 0.26-3ubunti [680 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblbgpc3 amd64 1.3.1-1ubunti [54.5 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblbc1-x86_64-linux-gnu amd64 13.2.0-23ubuntu4 [11.2 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblbc1-x86_64-linux-gnu amd64 13.2.0-23ubuntu4 [5326 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblbc1-x86_64-linux-gnu amd64 13.2.0-23ubuntu4 [5326 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblbc1-x86_64-linux-gnu amd64 13.2.0-23ubuntu4 [5326 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 fonts-dejavu-mono all 2.37.8 [502 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 fonts-dejavu-core all 2.37.8 [835 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 fontconfig-config amd64 2.15.0-1.ubuntu2 [37.3 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblbc1-0 amd64 14-20240412-0ubuntu1 [47.7 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblbgomp1 amd64 14-20240412-0ubuntu1 [47.7 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblhtml1 amd64 14-20240412-0ubuntu1 [28.9 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblwsan1 amd64 14-20240412-0ubuntu1 [28.9 kB]
```

```
ubuntu@ip-172-31-36-100:~$ sudo apt install -y nagios-nrpe-server nagios-plugins
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'nagios-plugins' instead of 'nagios-plugins'
The following additional packages will be installed:
  libnavah1-client3 libnavah1-common-data libnavah1-commons liblbcups2t64 liblbd1t64 liblbd2 libmysqlclient21 libnet-snmp-perl libpq5 libraddr11 libsnmpclient0 libsnmp-base libsnmp40t64
  liblalloc2 liblalloc2 liblweventt64 liblwparsr1 liblwparsr1 monitoring-plugins-basic monitoring-plugins-common monitoring-plugins-mysql python3-gpg python3-ldb
  python3-markdown python3-samba python3-talloc python3-tdb rcpbind samba-common-bin samba-dbus modules-samba-libs smclient smp
Suggested packages:
  cups-ppd liblhelpf-des-perl liblhelpf-hmac-perl liblio-socket-inetd perl snmp mibs-downloader icinga2 nagios-plugins-contrib fping postfix | sendmail-bin | exim4-daemon-heavy
  | exim4-daemon-light qstat xinetd | inetd python3-dnsproxy python3-libs
The following NEW packages will be installed:
  libnavah1-client3 libnavah1-common-data libnavah1-commons liblbcups2t64 liblbd1t64 liblbd2 libmysqlclient21 libnet-snmp-perl libpq5 libraddr11 libsnmpclient0 libsnmp-base libsnmp40t64
  liblalloc2 liblalloc2 liblweventt64 liblwparsr1 liblwparsr1 monitoring-plugins-basic monitoring-plugins-common monitoring-plugins-mysql python3-gpg python3-ldb
  python3-markdown python3-samba python3-talloc python3-tdb rcpbind samba-common-bin samba-dbus modules-samba-libs smclient smp
0 upgraded, 22 newly installed, 0 to remove and 0 not upgraded.
Need to get 16.1 MB of archives.
After this operation, 72.0 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 nagios-nrpe-server amd64 4.1.0-1ubuntu3 [356 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 nagios-nrpe-server amd64 4.1.0-1ubuntu3 [4.5 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libnavah1-client3 amd64 0.8.1-1ubuntu2 [29.7 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libnavah1-common3 amd64 0.8.1-1ubuntu2 [23.3 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libnavah1-common3 amd64 0.8-1ubuntu6 [26.8 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 liblbcups2t64 amd64 2.4.7-1.2ubuntu7.3 [272 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblbd1t64 amd64 0.9-0.6-1ubunti [25.7 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblbd1t64 amd64 0.9-0.6-1ubunti [25.7 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblbd1t64 1.4.10-1ubunti [46.8 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblweventt64 amd64 0.16.1-2ubunti [42.6 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblbd2 amd64 2.2.8.0+sam3a9.19.5+dfsg-1ubuntu9 [187 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 mysql-common all 5.8+1.1.ubunti [6748 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblweventt64 amd64 0.16.1-2ubunti [39.3ubuntu0.24.2 [1254 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblbc1-0 amd64 14-20240412-0ubuntu1 [47.7 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libpq5 amd64 16.4-ubuntu24.04.2 [141 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblweventt64 amd64 1.2.21-1ubunti [40.5 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblbd1t64 amd64 2.4.19.5+dfsg-1ubuntu9 [76.6 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 samba-libs amd64 2.14.1-1dfsg-1ubuntu9 [2017 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblbc1-0 amd64 14-20240412-0ubuntu1 [25.7 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libsnmp-base all 5.9.4+dfsg.1.ubuntu3 [206 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libsnmp40t64 amd64 5.9.4+dfsg.1.ubuntu3 [1066 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 liblwparsr1 amd64 0.9.7+dfsg-2ubunti [35.8 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 python3-gpg amd64 1.18.0-1ubuntu1 [209 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 python3-ldb amd64 2.2.8.0+sam3a9.19.5+dfsg-4ubuntu9 [41.8 kB]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu/nagios-markdown all 3.5.2-1 [72.0 kB]
```

Open nrpe.cfg file to make changes. **sudo**

nano /etc/nagios/nrpe.cfg

Under allowed_hosts, add your nagios host IP address like so

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

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

allowed_hosts=127.0.0.1,34.229.45.75

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

Now restart the NRPE server by this command. **sudo**

systemctl restart nagios-nrpe-server

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

Run the following command in the Nagios-host terminal **sudo**

systemctl status nagios

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

Oct 02 13:22:54 ip-172-31-41-160.ec2.internal nagios[78778]: SERVICE NOTIFICATION: nagiosadmin@localhost:Swap Usage;CRITICAL;notify-service-by-email;SWAP CRITICAL - 0 free (0 MB out of 0)
Oct 02 13:22:54 ip-172-31-41-160.ec2.internal nagios[78778]: wproc: NOTIFY job 3 from worker Core Worker 78782 is a non-check helper but exited with return code 127
Oct 02 13:22:54 ip-172-31-41-160.ec2.internal nagios[78778]: wproc: host=localhost; service=Swap Usage; contact=nagiosadmin
Oct 02 13:22:54 ip-172-31-41-160.ec2.internal nagios[78778]: wproc: early_timeout=0; exited_ok=1; wait_status=32512; error_code=0;
Oct 02 13:22:54 ip-172-31-41-160.ec2.internal nagios[78778]: wproc: stderr_line 01: /bin/sh: line 1: /bin/mail: No such file or directory
Oct 02 13:22:54 ip-172-31-41-160.ec2.internal nagios[78778]: wproc: stderr_line 02: /usr/bin/printf: write error: Broken pipe
Oct 02 13:23:13 ip-172-31-41-160.ec2.internal nagios[78778]: SERVICE ALERT: linux-server;Total Processes;OK;HARD;1;PROCS OK: 37 processes with STATE = R/SZDT
Oct 02 13:23:50 ip-172-31-41-160.ec2.internal nagios[78778]: SERVICE ALERT: linux-server;Current Load;OK;HARD;1;OK - load average: 0.01, 0.07, 0.04
Oct 02 13:24:28 ip-172-31-41-160.ec2.internal nagios[78778]: SERVICE ALERT: linux-server;Current Users;OK;HARD;1;USERS OK - 2 users currently logged in
Oct 02 13:24:46 ip-172-31-41-160.ec2.internal nagios[78778]: SERVICE ALERT: localhost;Current Users;OK;HARD;1;USERS OK - 2 users currently logged in
lines 1-26/26 (END).
```

Step 5: Visiting your nagios server using your nagios-host ip address Open up your browser and look for http://<public_ip_address_of_nagios-host>/nagios

The screenshot shows the Nagios Core 4.5.5 dashboard. At the top right, it says "Daemon running with PID 78778". Below that, the version "Nagios® Core™ Version 4.5.5" and the date "September 17, 2024" are displayed. A "Check for updates" link is also present. The left sidebar contains links for General, Current Status (Tactical Overview, Map, Hosts, Services, Host Groups, Service Groups, Problems), Reports (Availability, Trends, Alerts, History, Summary, Histogram, Notifications, Event Log), and a Quick Search bar. The main content area includes sections for "Get Started" (with bullet points about monitoring, changing look, extending with addons, support, training, and certification), "Latest News" (empty), "Don't Miss..." (empty), and "Quick Links" (links to Nagios Library, Labs, Exchange, Support, and the official website).

Click on Hosts.

This screenshot shows the Nagios Core 4.5.5 dashboard with a focus on host status. The "Current Network Status" section indicates "Last Updated: Wed Oct 2 13:40:35 UTC 2024" and "Updated every 90 seconds". It shows "Nagios® Core™ 4.5.5 - www.nagios.org" and "Logged in as nagiosadmin". The "Host Status Totals" table shows 2 Up, 0 Down, 0 Unreachable, and 0 Pending hosts. The "Service Status Totals" table shows 12 Ok, 1 Warning, 0 Unknown, 3 Critical, and 0 Pending services. Below these, the "Host Status Details For All Host Groups" table lists two hosts: "linux-server" and "localhost", both marked as UP. The "Status Information" column for "linux-server" shows "PING OK - Packet loss = 0%, RTA = 0.84 ms" and for "localhost" shows "PING OK - Packet loss = 0%, RTA = 0.04 ms". A "Limit Results: 100" dropdown is visible. A "Page Tour" button is located on the right side of the page.

Click on linux-server to view host information

Host Information

- Last Updated: Wed Oct 2 13:40:56 UTC 2024
- Updated every 90 seconds
- Nagios® Core™ 4.5.5 - www.nagios.org
- Logged in as nagiosadmin

Host: localhost
(linux-server)

Member of: No hostgroups

Address: 54.172.92.226

Host State Information

Host Status:	UP (for 0d 0h 20m 39s)
Status Information:	PING OK - Packet loss = 0%, RTA = 0.84 ms
Performance Data:	rta=0.838000ms;3000.000000;5000.000000;0.000000 p=0%;80;100;0
Current Attempt:	1/10 (HARD state)
Last Check Time:	10-02-2024 13:40:17
Check Type:	ACTIVE
Check Latency / Duration:	0.000 / 4.121 seconds
Next Scheduled Active Check:	10-02-2024 13:45:17
Last State Change:	10-02-2024 13:20:17
Last Notification:	N/A (notification 0)
Is This Host Flapping?	NO (0.00% state change)
In Scheduled Downtime?	NO
Last Update:	10-02-2024 13:40:46 (0d 0h 0m 10s ago)

Active Checks: ENABLED

Passive Checks: ENABLED

Obsessing: ENABLED

Notifications: ENABLED

Event Handler: ENABLED

Flap Detection: ENABLED

Host Commands

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

Host Comments

Add a new comment | Delete all comments

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

Current Network Status

- Last Updated: Wed Oct 2 13:42:09 UTC 2024
- Updated every 90 seconds
- Nagios® Core™ 4.5.5 - www.nagios.org
- Logged in as nagiosadmin

Host Status Totals

Up	Down	Unreachable	Pending
2	0	0	0

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
12	1	0	3	0

Service Status Details For All Hosts

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	10-02-2024 13:38:50	0d 0h 18m 19s	1/4	OK - load average: 0.00, 0.00, 0.00
	Current Users	OK	10-02-2024 13:39:28	0d 0h 17m 41s	1/4	USERS OK - 3 users currently logged in
	HTTP	CRITICAL	10-02-2024 13:40:05	0d 0h 27m 4s	4/4	connect to address 54.172.92.226 and port 80: Connection refused
	PING	OK	10-02-2024 13:40:43	0d 0h 21m 26s	1/4	PING OK - Packet loss = 0%, RTA = 1.05 ms
	Root Partition	OK	10-02-2024 13:41:12	0d 0h 20m 49s	1/4	DISK OK - free space: / 6122 MB (75.43% inode=98%).
	SSH	OK	10-02-2024 13:41:55	0d 0h 20m 11s	1/4	SSH OK - OpenSSH_9.6p1 Ubuntu-3ubuntu13.5 (protocol 2.0)
	Swap Usage	CRITICAL	10-02-2024 13:37:35	0d 0h 24m 34s	4/4	SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size.
	Total Processes	OK	10-02-2024 13:38:13	0d 0h 18m 56s	1/4	PROCS OK: 38 processes with STATE = RSZDT
	Current Load	OK	10-02-2024 13:40:00	0d 0h 22m 0s	1/4	OK - load average: 0.00, 0.00, 0.00
	Current Users	OK	10-02-2024 13:39:46	0d 0h 17m 23s	1/4	USERS OK - 3 users currently logged in
linux-server	HTTP	WARNING	10-02-2024 13:40:24	0d 0h 21m 45s	4/4	HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.001 second response time
	PING	OK	10-02-2024 13:41:01	0d 0h 21m 8s	1/4	PING OK - Packet loss = 0%, RTA = 0.04 ms
	Root Partition	OK	10-02-2024 13:41:39	0d 0h 20m 30s	1/4	DISK OK - free space: / 6122 MB (75.43% inode=98%).
	SSH	OK	10-02-2024 13:37:16	0d 0h 19m 53s	1/4	SSH OK - OpenSSH_8.7 protocol 2.0
	Swap Usage	CRITICAL	10-02-2024 13:37:54	0d 0h 24m 15s	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	10-02-2024 13:42:01	0d 0h 20m 8s	1/4	PROCS OK: 38 processes with STATE = RSZDT

Results 1 - 16 of 16 Matching Services

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

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

Prerequisites:

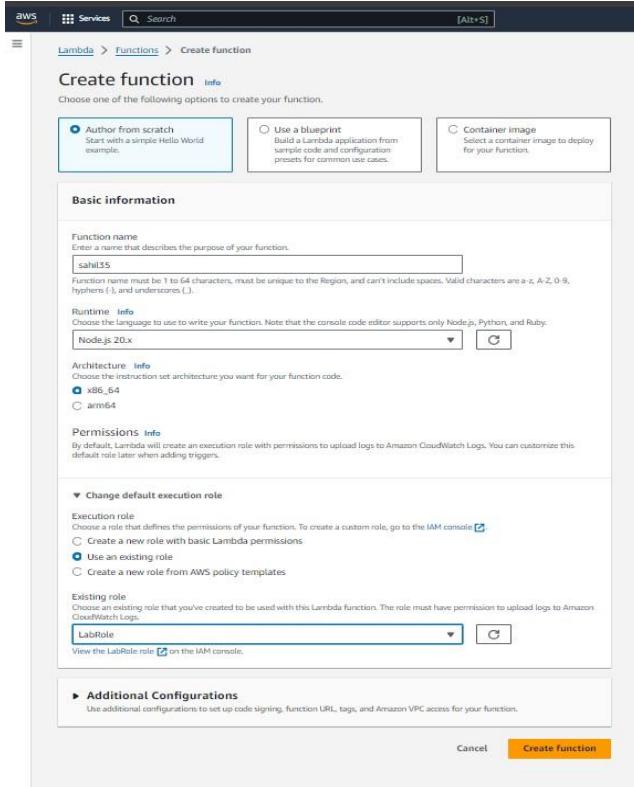
- 1) AWS account (academy recommended)

Step 1: Set up AWS Lambda Function

- 1) Search for Lambda in the services tab. Click on it once found.

- 2) Click on create functions.

- 3) Give a name to your Lambda function. Select the runtime as Node.js 20.x (You can also use python). Select the architecture as x86_64. Set the default execution role as LabRole if you are doing this on your academy account. (Use an existing role → LabRole)



- 4) Once the function is created, click on the name of the function (myLambda27 in my case).

Functions (7)					
<input type="text"/> Filter by tags and attributes or search by keyword					
	Function name	Description	Package type	Runtime	Last modified
<input type="checkbox"/>	myLambda27	-	Zip	Node.js 20.x	5 days ago
<input type="checkbox"/>	RoleCreationFunction	Create SLR if absent	Zip	Python 3.8	2 months ago
<input type="checkbox"/>	ModLabRole	updates LabRole to allow it to assume itself	Zip	Python 3.8	2 months ago
<input type="checkbox"/>	myLambda27_12	-	Zip	Python 3.12	5 days ago
<input type="checkbox"/>	MainMonitoringFunction	-	Zip	Python 3.8	2 months ago
<input type="checkbox"/>	RedshiftEventSubscription	Create Redshift event subscription to SNS Topic.	Zip	Python 3.8	2 months ago
<input type="checkbox"/>	RedshiftOverwatch	Deletes Redshift Cluster if the count is more than 2.	Zip	Python 3.8	2 months ago

- 5) This is the dashboard of our lambda function.

Name:Aditya Ahuja

Div:D15C

Roll no:02

Successfully created the function sahil35. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Lambda > Functions > sahil35

sahil35

Function overview [Info](#)

Description
-

Last modified
10 seconds ago

Function ARN
[arn:aws:lambda:us-east-1:767378259677:function:sahil35](#)

Function URL [Info](#)
-

[Throttle](#) [Copy ARN](#) [Actions ▾](#)

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

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

 sahil35
Layers (0)

+ Add trigger + Add destination

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

Code source [Info](#)

File Edit Find View Go Tools Window Test Deploy

Go to Anything (Ctrl-P) index.mjs Environment Var

```
index.mjs
1 export const handler = async (event) => {
2   // TODO implement
3   const response = {
4     statusCode: 200,
5     body: JSON.stringify('Hello from Lambda!'),
6   };
7   return response;
8 }
```

Upload from ▾

Code properties [Info](#)

Package size SHA256 hash Last modified

6) This function has the following default code, which is used to print “Hello form Lambda!”.

Successfully created the function sahil35. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Code source [Info](#)

File Edit Find View Go Tools Window Test Deploy

Go to Anything (Ctrl-P) index.mjs Environment Var

```
index.mjs
1 export const handler = async (event) => {
2   // TODO implement
3   const response = {
4     statusCode: 200,
5     body: JSON.stringify('Hello from Lambda!'),
6   };
7   return response;
8 }
```

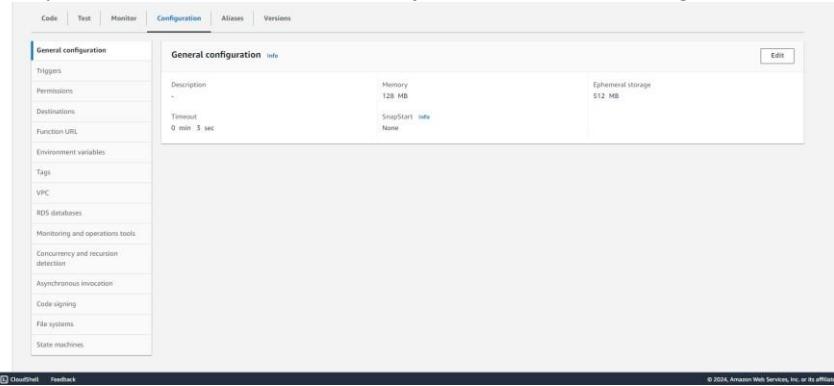
Upload from ▾

Code properties [Info](#)

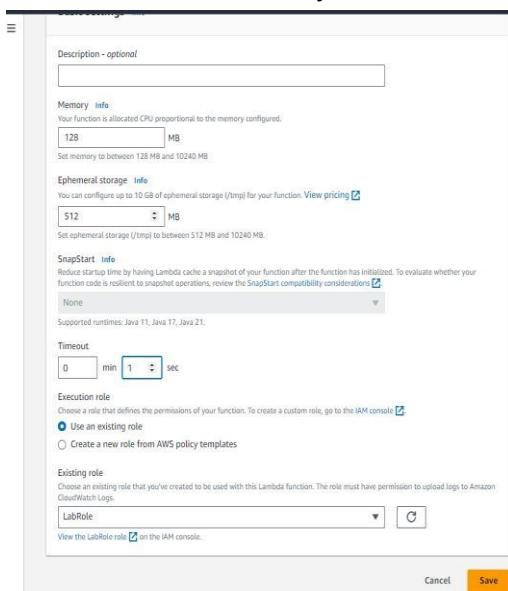
Package size SHA256 hash Last modified

Step 3: Set up configurations and test events

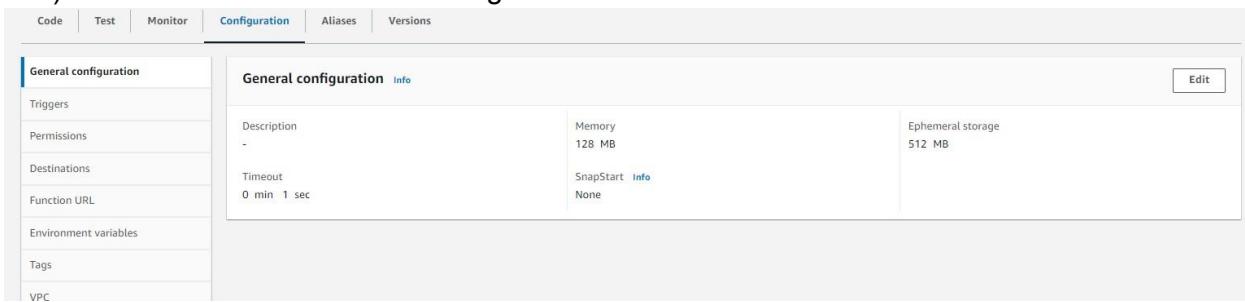
- 1) Just above the test code, you would find Configuration, click on it. Then click on Edit.



- 2) Here, change the Timeout to 1 sec. This is the time for which the function can be running before it is forcibly terminated.



- 3) We can see the executed changes.



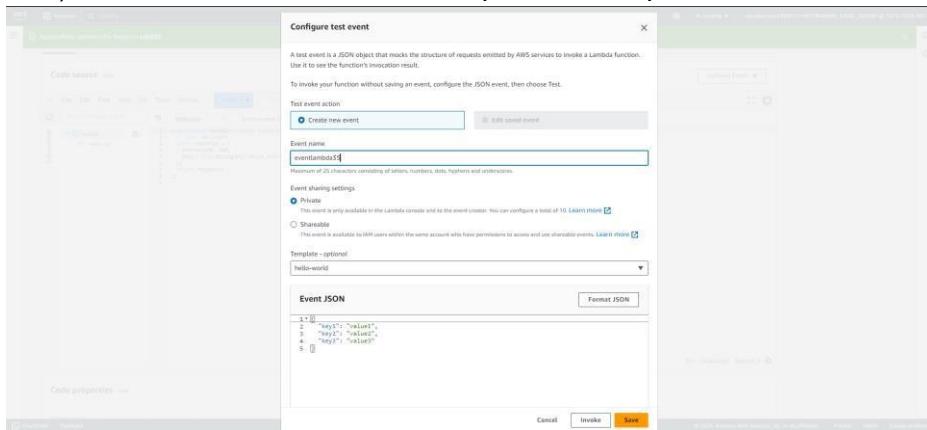
- 4) Switch back to the code tab. Click on the dropdown arrow near test. Then select configure test event.

```

Code source Info
File Edit Find View Go Tools Window Test Deploy
Go to Anything (Ctrl-P)
Environment sahil35 index.mjs Configure test event Ctrl-Shift-C
1 export const han...
2 // TODO implement
3 const response = {
4   statusCode: 200,
5   body: JSON.stringify('Hello from Lambda!'),
6 }
7 return response;
8
9

```

- 5) Here, create a new event, keep the other options default and save the event.



- 6) Now, again click on the dropdown. This time, select the event you have created. Then, click on TEST.

The test event eventlambda35 was successfully saved.

Code source Info
File Edit Find View Go Tools Window Test Deploy
Go to Anything (Ctrl-P)
Environment sahil35 index.mjs Configure test event Ctrl-Shift-C
1 export const han...
2 // TODO implement
3 const response = {
4 statusCode: 200,
5 body: JSON.stringify('Hello from Lambda!'),
6 }
7 return response;
8
9
Private saved events
• eventlambda35

7) We can see the expected output for the sample code.

The screenshot shows the AWS Lambda Test interface. At the top, a green banner says "The test event eventlambda35 was successfully saved." Below it, the "Code source" tab is selected. In the "index.js" file, there is a single line of code: `console.log('Hello from Lambda!')`. Under the "Execution results" tab, the response is shown as JSON: `{ "statusCode": 200, "body": "\u005chello from Lambda!\u005c" }`. The "Function Log" section contains several log entries, including the request ID and duration. At the bottom, the status is listed as "Status: Succeeded | Max memory used: 62 MB | Time: 5.33 ms".

8) For a test, declare a string and call it in line 6. After making the changes click on deplyo.

The screenshot shows the AWS Lambda Test interface. A red box highlights the "Test" button, which is currently blue and labeled "Changes not deployed". The "Code source" tab is selected, showing the index.js file with the following code:

```

1 const test="sahil"
2 export const handler = async (event) => {
3   const response = {
4     statusCode: 200,
5     body: JSON.stringify(test),
6   };
7   return response;
8 }
9 
```

9) Run the test. We can see that the string we declared has come in the output.

The screenshot shows the AWS Lambda Test interface. A green banner at the top says "Successfully updated the function sahil35.". Below it, the "Code" tab is selected. The "index.js" file now contains the modified code:

```

1 const test="sahil"
2 export const handler = async (event) => {
3   const response = {
4     statusCode: 200,
5     body: JSON.stringify(test),
6   };
7   return response;
8 }
9 
```

Under the "Execution results" tab, the response is shown as JSON: `{ "statusCode": 200, "body": "\u005chsahil\u005c" }`. The "Function Log" section shows the updated log entries. At the bottom, the status is listed as "Status: Succeeded | Max memory used: 62 MB | Time: 4.06 ms".

Conclusion:

In this experiment, we effectively investigated the AWS Lambda service by developing and configuring Lambda functions using Python, Java, or Node.js. We discovered how to establish a Lambda function, tweak its settings (like modifying the timeout duration), and evaluate the function with personalized events. Throughout this experience, we noted how Lambda manages executions, including timeout handling and producing expected results according to modifications in the code.

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.

Prerequisites:

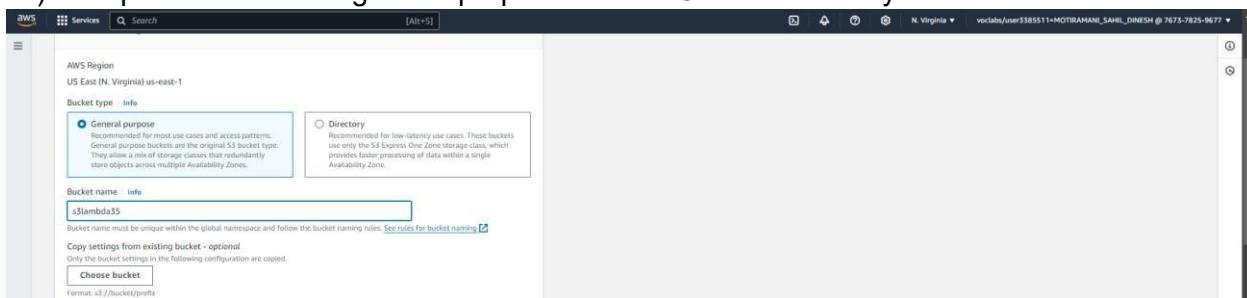
- 1) AWS account (academy preferable)
- 2) Lambda function (created in the previous experiment).

Step 1: Create a s3 bucket.

- 1) Search for S3 bucket in the services search. Then click on create bucket.



- 2) Keep the bucket as a general purpose bucket. Give a name to your bucket.



- 3) Uncheck block all public access.

Block all public access

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

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

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

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

AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

I acknowledge that the current settings might result in this bucket and the objects within becoming public.

- 4) Keeping all other options same, click on create. This would create your bucket. Now click on the name of the bucket.

Successfully created bucket "s3lambda35"

To upload files and folders, or to configure additional bucket settings, choose View details.

Amazon S3 > Buckets

▶ Account snapshot - updated every 24 hours All AWS Regions

General purpose buckets (1) All AWS Regions

Buckets are containers for data stored in S3.

Name	AWS Region	IAM Access Analyzer	Creation date
s3lambda35	US East (N. Virginia) us-east-1	<small>View analyzer for us-east-1</small>	October 6, 2024, 20:58:58 (UTC+05:30)

Copy ARN Empty Delete Create bucket

- 5) Here, click on upload, then add files. Select any image that you want to upload in the bucket and click on upload.

Amazon S3 > Buckets > s3lambda35

s3lambda35 Info

Objects (0) Info

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 Inventory [\[Learn more\]](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [\[Learn more\]](#)

Upload

Name	Type	Last modified	Size	Storage class
No objects				
Upload				

- 6) The image has been uploaded to the bucket.

The screenshot shows two windows from the AWS S3 console. The top window is titled 'Upload' and shows a file named '1.jpg' selected for upload. The bottom window is titled 'Upload succeeded' and shows a summary table with one file uploaded successfully and zero files failed. Below the summary is a detailed list of the uploaded file.

Files and folders (1 Total, 48.4 KB)	
<input type="checkbox"/> Name	1.jpg

Summary

Destination	Status
s3://s3lambda35	Succeeded 1 file, 48.4 KB (100.00%)
Failed 0 files, 0 B (0%)	

Files and folders (1 Total, 48.4 KB)

Name	Folder	Type	Size	Status	Error
1.jpg	-	image/jpeg	48.4 KB	Succeeded	-

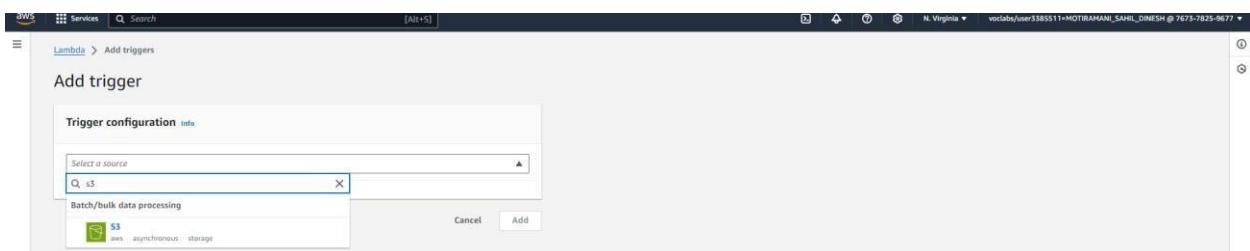
Step 2: Configure Lambda function

- 1) Go to the lambda function you had created before. (Services → Lambda → Click on name of function). Here, click on add trigger.

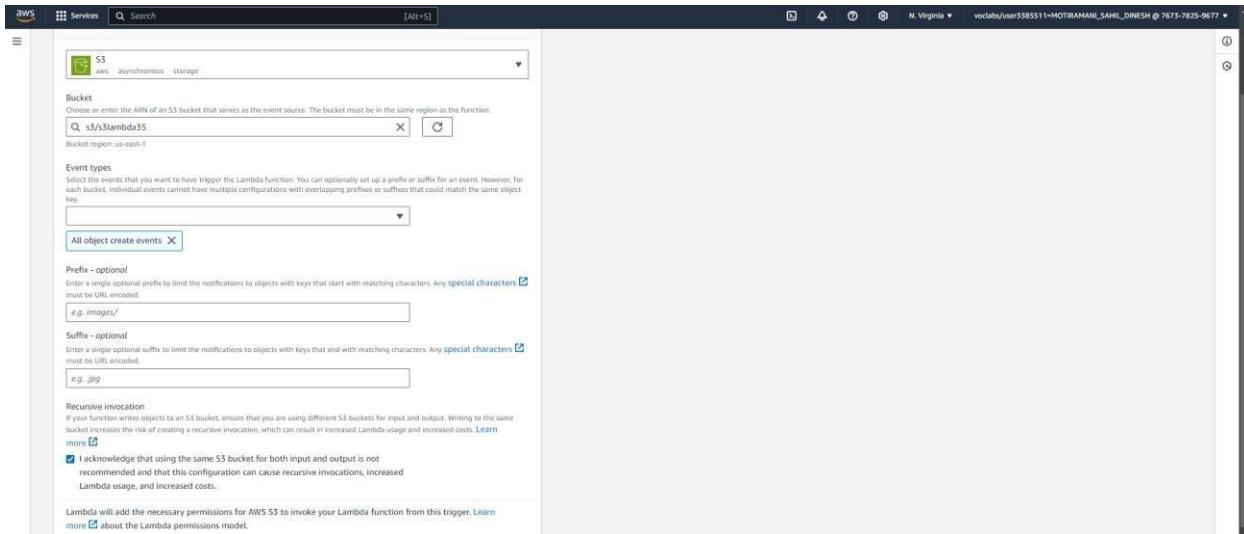
The screenshot shows the AWS Lambda console for a function named 'sahil35'. The 'General configuration' tab is selected, displaying settings such as triggers, permissions, and memory. The 'Code' tab is also visible at the bottom.

General configuration	General configuration	Code
Triggers	Description	Code
Permissions	Last modified	Test
Destinations	50 minutes ago	Monitor
Function URL	Function ARN	Configuration
Environment variables	arn:aws:lambda:us-east-1:176737825967:function:sahil35	Aliases
	Function URL	Versions

2) Under trigger configuration, search for S3 and select it.



Here, select teh S3 bucket you created for this experiment. Acknowledge the condition given by AWS. then click on Add. This will add the S3 bucket trigger to your function.



Scroll down to the code section of the function. Add the following javascript code to the code area by replacingthe existing code.

```
export const handler = async (event) => {
  (!event.Records || event.Records.length === 0) {
    console.error("No records found in the event.");
    return { statusCode: 400, body: JSON.stringify('No records
          found in the event')
  };

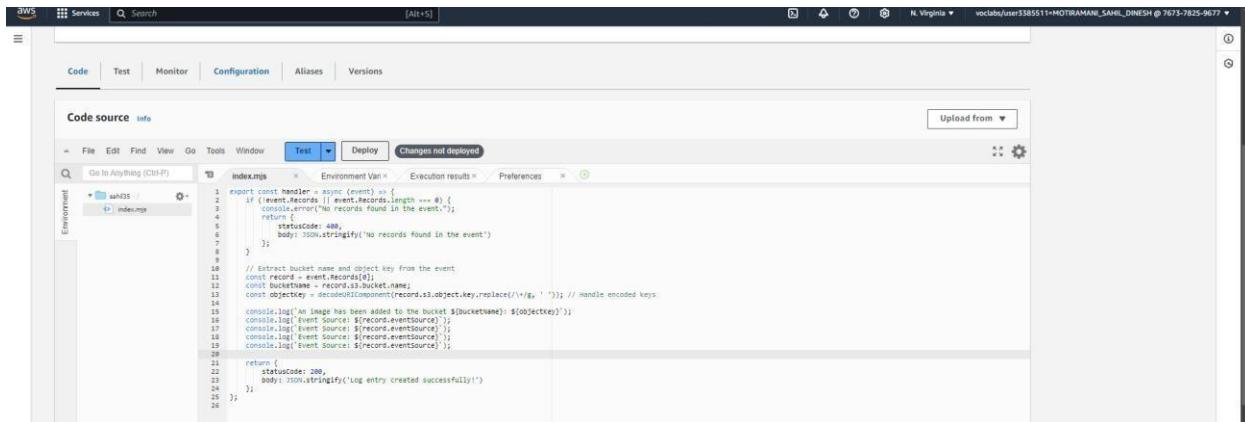
  // Extract bucket name and object key from the event
  const record = event.Records[0];
  const bucketName = record.s3.bucket.name;
  const objectKey = decodeURIComponent(record.s3.object.key.replace(/\+/g, ' '));
  // Handle encoded keys

  console.log(`An image has been added to the bucket ${bucketName}: ${objectKey}`);
  console.log(`Event Source: ${record.eventSource}`);
  console.log(`Event Source: ${record.eventSource}`);
  console.log(`Event Source: ${record.eventSource}`);
  return
}
```

```
{
  statusCode: 200, body: JSON.stringify('Log entry
  created successfully!')
};

}
```

This code checks for records in the event, extracts the bucket name and object key, logs the details, and returns a success message if an image is added to the bucket.



The screenshot shows the AWS Lambda function editor. The code source tab is selected, displaying the following JavaScript code:

```

1  export const handler = async (event) => {
2    if (!event.Records || event.Records.length === 0) {
3      throw new Error('No records found in the event.')
4    }
5    return {
6      statusCode: 400,
7      body: JSON.stringify('No records found in the event')
8    }
9  }
10 // Extract bucket name and object key from the event
11 const record = event.Records[0];
12 const bucketname = record.s3.bucket.name;
13 const objectkey = decodeURIComponent(record.s3.object.key.replace(/\+/g, ' ')); // Handle encoded keys
14
15 console.log(`An image has been added to the bucket ${bucketname}: ${objectkey}`);
16 console.log(`Event Source: ${record.eventSource}`);
17 console.log(`Event Source: ${record.eventSource}`);
18 console.log(`Event Source: ${record.eventSource}`);
19 console.log(`Event Source: ${records.eventSource}`);
20
21 return {
22   statusCode: 200,
23   body: JSON.stringify('Log entry created successfully!')
24 };
25 }
```

Now, click on the dropdown near test, then click on configure test event.

- 6) Here, select edit saved event. Select the event taht you had created before. Under Event JSON, paste the following code.

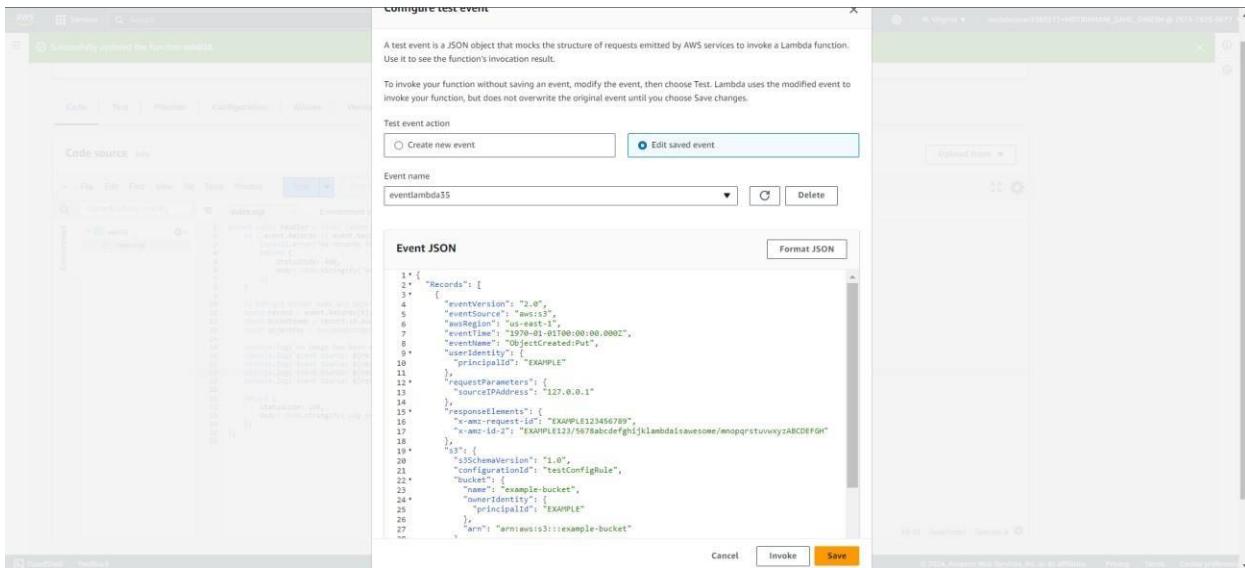
```
{
  "Records": [
    {
      "eventVersion": "2.0",
      "eventSource": "aws:s3",
      "awsRegion": "us-east-1",
      "eventTime": "1970-01-01T00:00:00.000Z",
      "eventName": "ObjectCreated:Put",
      "userIdentity": {
```

```
"principalId": "EXAMPLE"
},
"requestParameters": {
    "sourceIPAddress": "127.0.0.1"
},
"responseElements": {
    "x-amz-request-id": "EXAMPLE123456789", "x-amz-id-
2":
"EXAMPLE123/5678abcdefghijklmabaisawesome/mnopqrstuvwxyzABCDEFGHI"
},
"s3": {
    "s3SchemaVersion": "1.0",
    "configurationId": "testConfigRule",
}

"bucket": {
    "name": "example-bucket", "ownerIdentity": {
        "principalId": "EXAMPLE"
    },
    "arn": "arn:aws:s3:::example-bucket"
},
"object": {
    "key": "test%2Fkey",
    "size": 1024,
    "eTag": "0123456789abcdef0123456789abcdef", "sequencer": "0A1B2C3D4E5F678901"
}
}
```

```
}
```

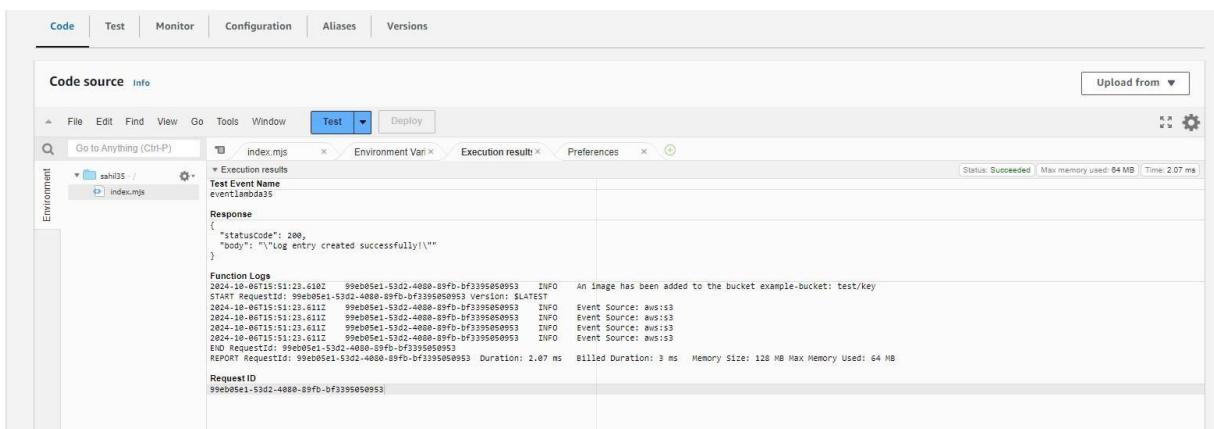
This JSON structure represents an S3 event notification triggered when an object is uploaded to an S3 bucket. It contains details about the event, including the bucket name (example-bucket), the object key (test/key), and metadata like the object's size, the event source (aws:s3), and the event time.



Save the changes. Then deploy the code changes by clicking on deploy.

7) After deploying, click on Test. The console output shows that 'an image has been added to the bucket'

The JSON response shows that the log entry was created successfully.



Step 3: Check the logs

- 1) To check the logs explicitly, search for CloudWatch on services and open it in a new tab.

The screenshot shows two separate browser tabs. The left tab is titled 'Services' and contains a search bar for 'cloud watch'. Below the search bar, there's a list of services: CloudWatch (selected), Athena, Amazon EventBridge, and S3. The right tab is titled 'Applications (0)' and shows a list of applications with a 'Create application' button at the bottom.

- 2) Here, Click on Logs → Log Groups. Select the log that has the lambda function name you just ran.

The screenshot shows the 'Log groups (4)' page in the CloudWatch service. The left sidebar has a 'Logs' section selected. The main area displays four log groups: '/aws/lambda/RedshiftEventSubscription', '/aws/lambda/RedshiftOverwatch', '/aws/lambda/RoleCreationFunction', and '/aws/lambda/sahilSS'. Each log group row includes columns for Log class, Anomaly detection, Data protection, Sensitive data controls, Retention, Metric filters, and Contributor Insights.

- 3) Here, under Log streams, select the log stream you want to check.

The screenshot shows the AWS CloudWatch console. The left sidebar is collapsed. The main area displays the 'Log group details' for the log group '/aws/lambda/sahil35'. Below this, the 'Log streams' section lists three entries:

- 2024/10/06/[\$LATEST]d475d9ade605443995f9aaaa948219e4 (Last event time: 2024-10-06 15:46:47 (UTC))
- 2024/10/06/[\$LATEST]f601fb3fec0460c95b5260cb1073672 (Last event time: 2024-10-06 14:49:37 (UTC))
- 2024/10/06/[\$LATEST]6911d11a337b49be937e17816cc40bc5 (Last event time: 2024-10-06 14:47:00 (UTC))

4) Here again, we can see that 'An image has been added to the bucket'.

The screenshot shows the AWS CloudWatch console with the 'Logs' section expanded. The main area displays the 'Log events' for the log stream '/aws/lambda/sahil35/2024/10/06/[\$LATEST]d475d9ade605443995f9aaaa948219e4'. The log entries include:

- 2024-10-06T15:46:47.632Z START RequestId: e226dc0b-1951-4ef7-9a1d-58c9a2d4a5dc Version: \$LATEST
- 2024-10-06T15:46:47.632Z END RequestId: e226dc0b-1951-4ef7-9a1d-58c9a2d4a5dc ERROR No records found in the event.
- 2024-10-06T15:46:47.637Z REPORT RequestId: e226dc0b-1951-4ef7-9a1d-58c9a2d4a5dc Duration: 47.61 ms Billed Duration: 48 ms Memory Size: 128 MB Max Memory Used: 64 MB Init Duration: 199.25 ms
- 2024-10-06T15:51:23.618Z REPORT RequestId: 99e095e1-5302-4080-89fd-bf3399e50953 INFO An image has been added to the bucket example-bucket: test/key
- 2024-10-06T15:51:23.618Z START RequestId: 99e095e1-5302-4080-89fd-bf3399e50953 Version: \$LATEST
- 2024-10-06T15:51:23.618Z END RequestId: 99e095e1-5302-4080-89fd-bf3399e50953 INFO Event Source: aus13
- 2024-10-06T15:51:23.618Z REPORT RequestId: 99e095e1-5302-4080-89fd-bf3399e50953 Duration: 2.07 ms Billed Duration: 3 ms Memory Size: 128 MB Max Memory Used: 64 kB

Conclusion:

In this experiment, we developed and deployed a Lambda function designed to respond to file uploads in an S3 bucket. The function was triggered automatically whenever a new object was added to the bucket, illustrating how AWS services can efficiently automate workflows. The Lambda function extracted and logged key details from the event, such as the bucket's name and the object's key. We tested this by uploading a sample file, and upon reviewing the logs in CloudWatch, we confirmed that the function executed successfully, capturing the upload event.

Name:Aditya Ahuja

Div :D15C

Roll no:02

This experiment demonstrated the powerful synergy between AWS Lambda and S3, enabling seamless, event-driven automation.