

### Expt 1a]Website Hosted Locally on XAMPP

Step1: To download the latest version of XAMPP (8.2.12 / PHP 8.2.12) for Windows, visit the official XAMPP website, navigate to the download section, and select the appropriate version for your system.

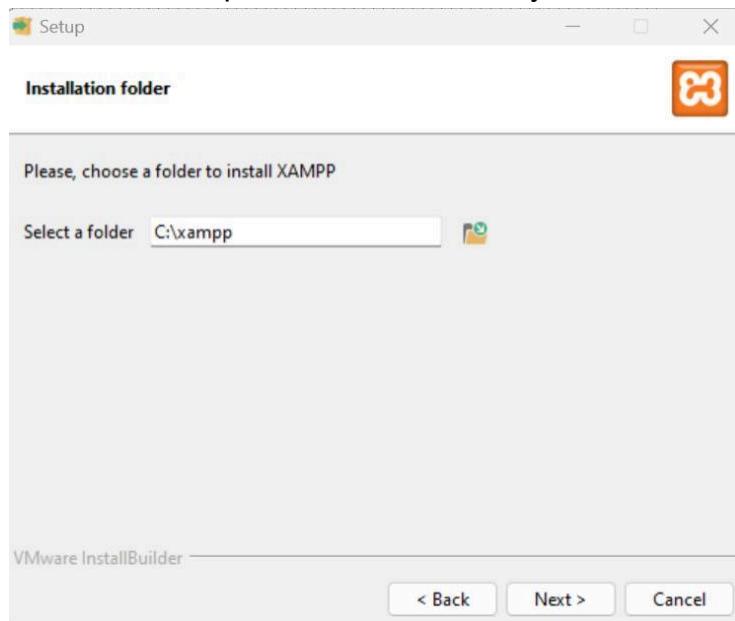
<https://www.apachefriends.org/download.html>

The screenshot shows the Apache Friends XAMPP Download page. At the top, there's a navigation bar with links for Apache Friends, Download, Hosting, Community, and About. A search bar and a language selection (EN) are also present. The main content area is titled "Download". It features two sections: "XAMPP for Windows" and "XAMPP for Linux". The Windows section lists three versions: 8.0.30 / PHP 8.0.30, 8.1.25 / PHP 8.1.25, and 8.2.12 / PHP 8.2.12. Each entry includes a "What's Included?" link, checksums (md5, sha1), download links (Download (64 bit)), and file sizes (144 Mb, 148 Mb, 149 Mb). The Linux section has a similar layout. To the right, there's a sidebar titled "Documentation/FAQs" with links to Linux, Windows, and OS X FAQs. Below the main content, there's a note about Windows XP and 2003 support.

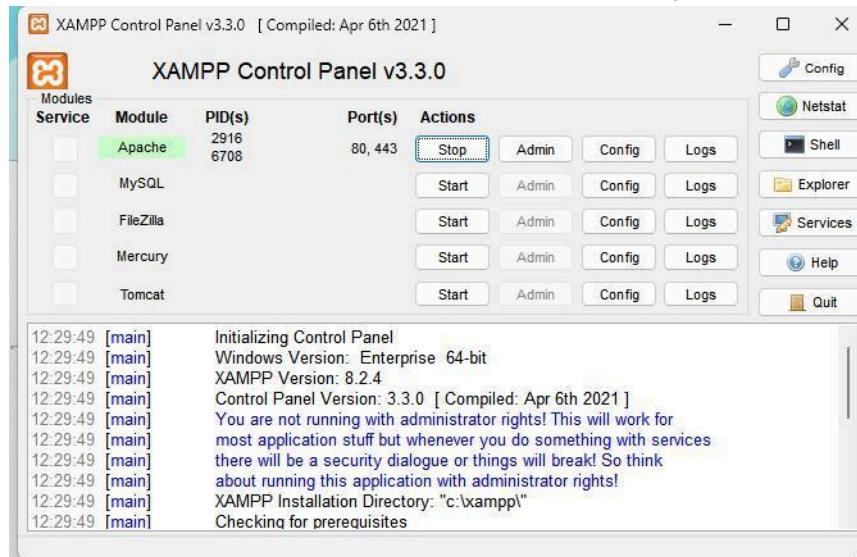
Step 2: XAMPP offers various components during installation, including Apache (web server), MySQL (database management), PHP (server-side scripting language), and more. These components are essential for running a local development environment. You can choose to install only the components you require for your specific project.

The screenshot shows the XAMPP Setup window with the title "Select Components". It displays a tree view of available components under "VMware InstallBuilder". The "Server" category is expanded, showing Apache, MySQL, FileZilla FTP Server, Mercury Mail Server, and Tomcat, all of which have checkboxes checked. The "Program Languages" category is also expanded, showing PHP and Perl, both with checked checkboxes. Under "Program Languages", there are sub-categories for "phpMyAdmin" and "Webalizer", each with a checked checkbox. On the right side, there's a large orange icon of a person with arms raised. Below the tree view, a message says "Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue." A note also says "Click on a component to get a detailed description". At the bottom, there are buttons for "< Back", "Next >", and "Cancel".

Step 3: Choose the destination folder where you want to install XAMPP and store your local files. This path will be where all your web server files and configurations are saved.

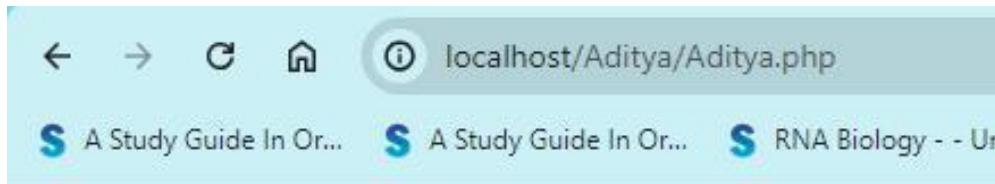


Step 4: After successfully downloading XAMPP, open the XAMPP Control Panel, and under the "Module" section, click "Start" next to Apache to activate your local server.



Step 5: Go to the following directory and save your files there so that they can be locally hosted

Step 6: Open your web browser and navigate to 'localhost', followed by your folder name (e.g., 'localhost/your-folder-name'). You will see a list of available PHP files—select one to run it, and it will be hosted successfully.



# Welcome

I am Aditya Raorane

## 1a] Website Hosted Remotely on AWS S3 Bucket

Step 1: In your AWS Academy account, navigate to the "Services" search bar, type "S3" and open it to access the Amazon Simple Storage Service.

Select your cookie preferences  
We use essential cookies and similar tools that are necessary to provide our site and services. We use performance cookies to collect anonymous statistics, so we can understand how customers use our site and make improvements. Essential cookies cannot be deactivated, but you can choose "Customize" or "Decline" to decline performance cookies.  
If you agree, AWS and approved third parties will also use cookies to provide useful site features, remember your preferences, and display relevant content, including relevant advertising. To accept or decline all non-essential cookies, choose "Accept" or "Decline." To make more detailed choices, choose "Customize."  
Accept  
Decline  
Customize

Step 2: In AWS S3, a bucket is a container for storing objects, with globally unique names and customizable configurations for data management and access control. Click on "Create Bucket" to create one.

The screenshot shows the AWS S3 service page. On the left, there's a sidebar with options like Buckets, Access Grants, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, IAM Access Analyzer for S3, Block Public Access settings for this account, and Storage Lens (which is expanded to show Dashboards, Storage Lens groups, and AWS Organizations settings). The main area displays an 'Account snapshot - updated every 24 hours' section with a link to 'All AWS Regions'. Below it, under 'General purpose buckets', there's a table with one entry:

Name	AWS Region	IAM Access Analyzer	Creation date
<a href="#">elasticbeanstalk-us-east-1-090530457694</a>	US East (N. Virginia) us-east-1	<a href="#">View analyzer for us-east-1</a>	August 5, 2024, 14:00:29 (UTC+05:30)

**Step 3:** Under "Bucket Type," select **General Purpose** because creating a bucket sets up a new storage container with a unique name and configurations. If you select **Directory**, it will organize files within an existing bucket, making it easier to manage and retrieve specific files, but it won't create a new storage container.

The screenshot shows the 'Create bucket' wizard. It starts with a 'General configuration' step. The 'AWS Region' is set to 'US East (N. Virginia) us-east-1'. The 'Bucket type' section shows two options: 'General purpose' (selected) and 'Directory - New'. The 'Bucket name' field contains 'www.raorane.com'. Below the form, a note says 'Bucket name must be unique within the global namespace and follow the bucket naming rules.' A link 'See rules for bucket naming' is provided. At the bottom, there's a note about 'Copy settings from existing bucket - optional'.

Step 4: Under "Object Ownership," select ACLs disabled to ensure that all objects are owned by the bucket owner, providing simplified permissions management and improved security.

Object Ownership [Info](#)

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

**ACLs disabled (recommended)**  
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

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

Object Ownership  
Bucket owner enforced

**Block Public Access settings for this bucket**

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

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

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

[CloudShell](#) [Feedback](#) © 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Step 5: Under "Object Lock," I chose **Disable** to allow unrestricted deletion and modification of objects within the bucket. And finally click on "Create Bucket".

Advanced settings

**Object Lock**  
Store objects using a write-once-read-many (WORM) model to help you prevent objects from being deleted or overwritten for a fixed amount of time or indefinitely. Object Lock works only in versioned buckets. [Learn more](#)

**Disable**

**Enable**  
Permanently allows objects in this bucket to be locked. Additional Object Lock configuration is required in bucket details after bucket creation to protect objects in this bucket from being deleted or overwritten.

**Object Lock works only in versioned buckets. Enabling Object Lock automatically enables Versioning.**

**After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.**

[Cancel](#) [Create bucket](#)

Step 6: Finally, once the bucket is created, click on the bucket link to view its properties and configuration details.

The screenshot shows two consecutive screenshots of the AWS S3 console.

**Screenshot 1: Bucket Creation Confirmation**

A green success message at the top states: "Successfully created bucket 'www.raorane.com'". It includes a "View details" button and a "Create bucket" button. Below this, the "General purpose buckets" section lists two buckets:

Name	AWS Region	IAM Access Analyzer	Creation date
elasticbeanstalk-us-east-1-090530457694	US East (N. Virginia) us-east-1	<a href="#">View analyzer for us-east-1</a>	August 5, 2024, 14:00:29 (UTC+05:30)
www.raorane.com	US East (N. Virginia) us-east-1	<a href="#">View analyzer for us-east-1</a>	August 12, 2024, 15:12:44 (UTC+05:30)

**Screenshot 2: Bucket Properties**

The URL is https://us-east-1.console.aws.amazon.com/s3/buckets/www.raorane.com?region=us-east-1&bucketType=general. The page shows the "Properties" tab selected under "Objects".

**Bucket overview**

AWS Region US East (N. Virginia) us-east-1	Amazon Resource Name (ARN) <a href="#">arn:aws:s3:::www.raorane.com</a>	Creation date August 12, 2024, 15:12:44 (UTC+05:30)
---	--	--

**Bucket Versioning**

Bucket Versioning is disabled. There is an "Edit" button available.

Step 7: Search for Static Website Hosting; it is disabled by default but needs to be enabled to serve static web content from your bucket. Enabling Static Website Hosting allows your S3 bucket to serve static content, such as HTML files. You must specify an index document (e.g., index.html) and optionally an error document. This configuration makes your bucket accessible via a web URL, hosting your static website directly

The screenshot shows the AWS S3 Bucket Properties page. In the 'Static website hosting' section, the status is set to 'Disabled'. An 'Edit' button is visible to the right of the status field.

Step 8: Enable static website hosting option and select host type as a “static website”

The screenshot shows the 'Edit static website hosting' configuration page. Under 'Hosting type', the 'Host a static website' option is selected. A note at the bottom states: '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.'

Step 9: Specify the index document (e.g., index.html) and the error document (e.g., error.html) along with the desired HTTP error code to display custom error messages.

The screenshot shows the AWS S3 Website Configuration interface for the 'www.raorane.com' bucket. At the top, there's a note about making content public readable via S3 Block Public Access settings. Below it, the 'Index document' field is set to 'aditya.html'. The 'Error document - optional' field is set to '404.html'. Under 'Redirection rules - optional', there is one rule listed with the ID '1'. At the bottom, there are links for CloudShell, Feedback, and cookie preferences.

Step 10: Under the “Objects” section, you will see the page to upload your HTML files. Click on the “Upload” button.

The screenshot shows the AWS S3 Objects page for the 'www.raorane.com' bucket. The 'Objects' tab is selected. At the top, there are buttons for Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, and Upload. A search bar for 'Find objects by prefix' is also present. The main area displays a message stating 'No objects' and 'You don't have any objects in this bucket.' There is a prominent 'Upload' button at the bottom. The navigation bar at the top includes links for CloudShell, Feedback, and cookie preferences.

Step 11: Add your file(s) and click the **Upload** button. You will see a confirmation indicating that the file was successfully uploaded.

The screenshot shows two consecutive screenshots of the AWS S3 console interface.

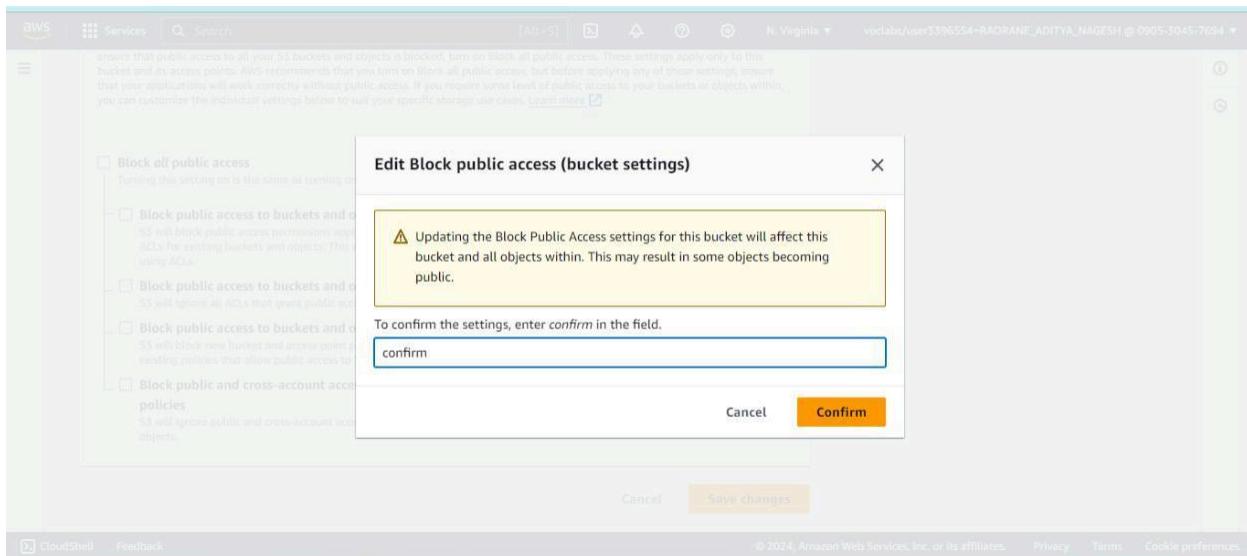
**Screenshot 1 (Top):** The "Upload" page. A message at the top says: "Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#)". Below is a large dashed blue box with the placeholder text "Drag and drop files and folders you want to upload here, or choose Add files or Add folder.". Underneath is a table titled "Files and folders (1 Total, 4.2 KB)". It contains one item: "aditya.html" (text/html, 4.2 KB). There are "Remove", "Add files", and "Add folder" buttons above the table. A search bar and pagination controls (< 1 >) are also present.

**Screenshot 2 (Bottom):** The "Upload succeeded" confirmation page. The title bar says "Upload succeeded". Below it, a message says "View details below." A summary table shows the upload results:

Destination	Succeeded	Failed
s3://www.raorane.com	1 file, 4.2 KB (100.00%)	0 files, 0 B (0%)

Below the summary is a "Files and folders" section with a table showing the uploaded file "aditya.html" (text/html, 4.2 KB, Status: Succeeded).

Step 12: Now, if you click on the link, it will show an error 403 Forbidden message due to block policies. To resolve this, you need to configure the bucket's public access settings by unchecking all block public access options.



Step 13: Scroll down to the **Bucket Policy** section and paste the policy from the following link:

<https://gist.github.com/Savjee/b4b3a21d143a30e7dc07>

To configure access permissions:

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

The screenshot shows the AWS S3 Policy editor interface. On the left, there's a sidebar with various AWS services like Access Grants, Object Lambda Access Points, and IAM Access Analyzer. The main area displays a JSON policy document:

```

1  {
2    "Version": "2012-10-17",
3    "Statement": [
4      {
5        "Sid": "PublicReadGetObject",
6        "Effect": "Allow",
7        "Principal": "*",
8        "Action": "s3:GetObject",
9        "Resource": "arn:aws:s3:::www.raorane.com/*"
10      }
11    ]
12  }
13 }
14 }

```

To the right of the policy, there's a panel titled "Edit statement PublicReadGetObject" with a "Remove" button. Below it, there's a "Add actions" section with a search bar and a list of available services: S3, AMP, API Gateway, API Gateway V2, and ASG.

**Step 14:** Finally, you will see a confirmation about the policies. The website is now ready to run—click on the provided link to view the live website.

The screenshot shows the AWS S3 bucket creation confirmation. It says "Successfully created bucket 'www.raorane.com'". Below it, there's a summary of the bucket settings and a "View details" button.

Below this, the S3 buckets page is shown. It lists a single bucket named "elasticbeanstalk-us-east-1-090530457604" with the status "General purpose buckets (2) info All AWS Regions". There are buttons for "Create bucket", "Copy ARN", "Empty", and "Delete".

At the bottom, a browser window is open to the URL <http://us-east-1.console.aws.amazon.com/s3/#/>. The page title is "Shop with Sale". It features a navigation bar with "HOME", "PRODUCTS", "ABOUT US", and "CONTACT". A pink banner at the top says "Summer Sale! Up to 50% Off on Select Items". Below it, three products are displayed: "Product 1" (Elegant Watch), "Product 2" (Stylish Sunglasses), and "Product 3" (Leather Handbag). At the bottom, a footer note says "© 2024 Shop with Sale. All rights reserved."

### **1b]Cloud9 IDE Collaborative Setup**

**Step 1:**In your AWS Academy account, navigate to the "Services" search bar, type "Cloud9" and open it to access the Cloud9 IDE.

Search results for 'cloud9'

**Services** See all 51 results ▶

- Cloud9** ★  
A Cloud IDE for Writing, Running, and Debugging Code
- Amazon CodeCatalyst** ★  
Integrated DevOps Service
- AWS Cloud Map** ★  
Build a dynamic map of your cloud
- AWS Deadline Cloud** ★  
Simplified render management

**Step 2:**To set up an environment in AWS Cloud9, create a **new Cloud9 environment** in the AWS Management Console, choosing your preferred instance type and VPC settings. Select “EC2 instance” in the environment type.

AWS Cloud9 > Environments > Create environment

Create environment Info

**Details**

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

Description - optional  
  
Limit 200 characters.

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

**New EC2 instance**  
Cloud9 creates an EC2 instance in your account. The

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

**Instance type**

- 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-sized web projects.
- m5.large (8 GiB RAM + 2 vCPU)**  
Recommended for production and general-purpose development.
- Other instance type**  
Select an instance type.

t3.nano

**Platform**

- Amazon Linux 2 (recommended)
- Amazon Linux AMI
- Ubuntu Server 10.04 LTS

**Cost-saving setting**

Choose a predetermined amount of time to auto-hibernate your environment and prevent unnecessary charges. We recommend a hibernation setting of half an hour of no activity to maximize savings.

After 30 minutes (default)

**IAM role**

AWS Cloud9 creates a service-linked role for you. This allows AWS Cloud9 to call other AWS services on your behalf. You can delete the role from the AWS IAM console once you no longer have any AWS Cloud9 environments. [Learn more](#)

AWSServiceRoleForAWSCloud9

▶ Network settings (advanced)

No tags associated with the resource.

Add new tag

You can add 60 more tags.

Cancel Previous step Next step

---

**Timeout**

How long Cloud9 can be inactive (no user input) before auto-hibernating. This helps prevent unnecessary charges.

30 minutes

---

**Network settings** [Info](#)

**Connection**

How your environment is accessed.

- AWS Systems Manager (SSM)  
Accesses environment via SSM without opening inbound ports (no ingress).
- Secure Shell (SSH)  
Accesses environment directly via SSH, opens inbound ports.

▶ VPC settings [Info](#)

▶ Tags - optional [Info](#)

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

**i** The following IAM resources will be created in your account

- **AWSServiceRoleForAWSCloud9** - AWS Cloud9 creates a service-linked role for you. This allows AWS Cloud9 to call other AWS services on your behalf. You can delete the role from the AWS IAM console once you no longer have any AWS Cloud9 environments. [Learn more](#)

Step 3: Finally the environment with EC2 instances is created.

The screenshot shows the AWS Cloud9 Environments page. At the top, there is a breadcrumb navigation: AWS Cloud9 > Environments. Below the header, there is a table titled "Environments (1)". The table has columns: Name, Cloud9 IDE, Environment type, Connection, Permission, and Owner ARN. One row is present in the table, corresponding to the environment "Aditya Raorane".

Name	Cloud9 IDE	Environment type	Connection	Permission	Owner ARN
Aditya Raorane	Open	EC2 instance	Secure Shell (SSH)	Owner	arn:aws:sts::090530457694:assumed-role/voclabs/user3396554=RAORANE_ADITYA_NAGESH

The screenshot shows the details page for the environment "Aditya Raorane". At the top, there is a title "Aditya Raorane" and two buttons: "Delete" and "Open in Cloud9". Below the title, there is a section titled "Details" with an "Edit" button. The details are listed in a table:

Name Aditya Raorane	Owner ARN arn:aws:sts::090530457694:assumed-role/voclabs/user3396554=RAORANE_ADITYA_NAGESH	Status Creating
Description First experiment	Number of members 1	Lifecycle status Creating
Environment type EC2 instance		

Below the "Details" section, there are three tabs: "EC2 instance" (selected), "Network settings", and "Tags". The "EC2 instance" tab displays the following information:

ARN arn:aws:cloud9:us-east-1:090530457694:environment:dcf40a1e1d824b2d9f0d4c0779f5a107	Instance type t2.micro (1 GiB RAM + 1 vCPU)
Platform Amazon Linux 2023	Storage EBS only

At the top right of the "EC2 instance" tab, there is a "Manage EC2 instance" button.

Step 4: Configure a username and password for a user and assign the role.

### Specify user details

**User details**

User name  
AdityaRaorane

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ \_ - (hyphen)

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

Console password

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

Custom password  
Enter a custom password for the user.  
AdityaRaorane@44

Show password

Users must create a new password at next sign-in - Recommended  
Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keypairs, you can generate them after you create this IAM user. [Learn more](#)

[Cancel](#) [Next](#)

---

**Timeout**  
How long Cloud9 can be inactive (no user input) before auto-hibernating. This helps prevent unnecessary charges.  
30 minutes

**Network settings** [Info](#)

**Connection**  
How your environment is accessed.

AWS Systems Manager (SSM)  
Accesses environment via SSM without opening inbound ports (no ingress).

Secure Shell (SSH)  
Accesses environment directly via SSH, opens inbound ports.

► **VPC settings** [Info](#)

► **Tags - optional** [Info](#)  
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

**i** The following IAM resources will be created in your account

- **AWSServiceRoleForAWSCloud9** - AWS Cloud9 creates a service-linked role for you. This allows AWS Cloud9 to call other AWS services on your behalf. You can delete the role from the AWS IAM console once you no longer have any AWS Cloud9 environments. [Learn more](#)
- **AWSCloud9SSMAccessRole** and **AWSCloud9SSMInstanceProfile** - A service role and an instance profile are automatically created if Cloud9 accesses its EC2 instance through AWS Systems Manager. If your environments no longer require EC2 instances that block incoming traffic, you can delete these

[CloudShell](#) [Feedback](#) © 2024, Amazon Web Services, Inc. or its affiliates.

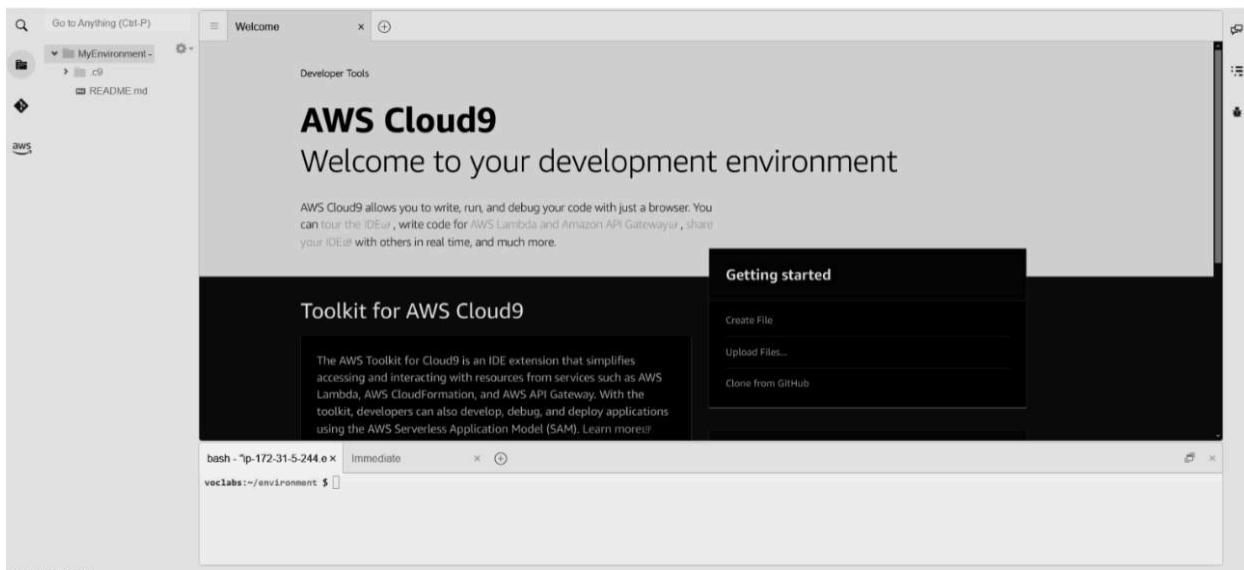
The screenshot shows the AWS Cloud9 IAM resource creation interface. At the top, there's a section for 'Tags - optional' with a note about tags being optional labels for AWS resources. Below this, a box informs the user that IAM resources will be created in their account, listing two specific roles: 'AWSServiceRoleForAWSCloud9' and 'AWSCloud9SSMAccessRole'.

At the bottom right, there are 'Cancel' and 'Create' buttons. Two error messages are displayed in red boxes:

- "There was an error creating the IAM resources needed for SSM connection."
- "You don't have the permission required to perform this operation. Ask your administrator to give you permissions."

Note: If you're unable to create a user with your AWS Academy account, it likely stems from limited permissions or role restrictions imposed to prevent unauthorized actions or costs. These accounts often have constraints tailored for educational purposes, so you may need to use a personal account or request assistance from your instructor.

## Step 5: Open Cloud9 IDE.



Step 6: Open AWS IAM service. Configure a user and a group.

Specify user details

User details

User name

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = . @ \_ - (hyphen)

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

Console password

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

Custom password  
Enter a custom password for the user.

Show password

Users must create a new password at next sign-in - Recommended  
Users automatically get the IAMUserChangePassword [policy](#) to allow them to change their own password.

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

Cancel Next

---

IAM > Users > Create user

Step 1  
Specify user details

Step 2  
**Set permissions**

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

Step 3  
Review and create

**Permissions options**

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

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

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

▶ Set permissions boundary - *optional*

Cancel Previous Next

### 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.

myweb-app-group

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

#### Permissions policies (947)

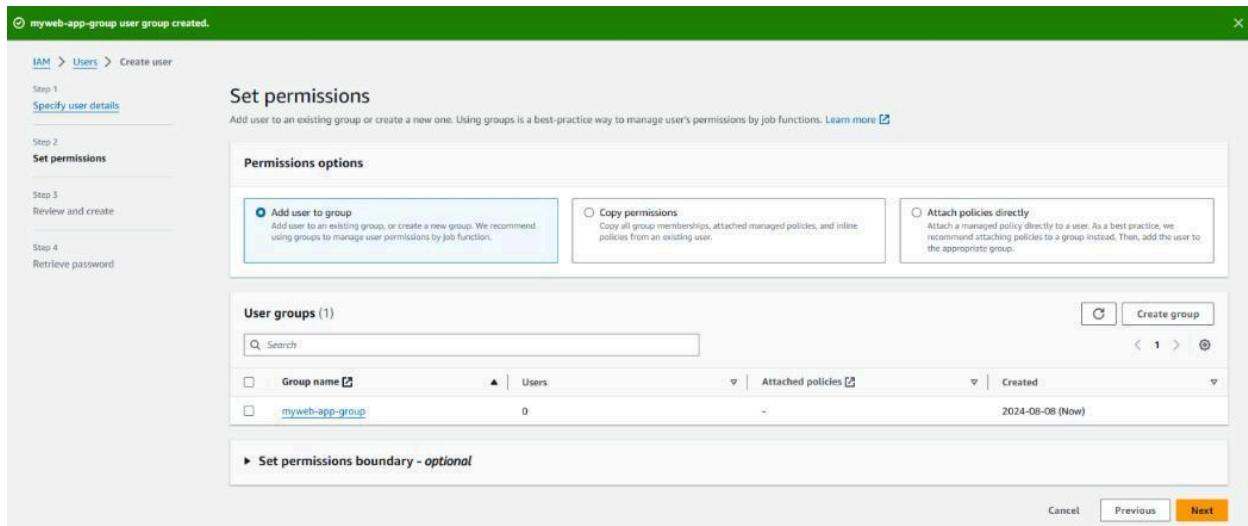
Filter by Type

<input type="checkbox"/>	Policy name	Type	Use...	Description
<input type="checkbox"/>	<a href="#">AdministratorAccess</a>	AWS managed ...	None	Provides full access to AWS services
<input type="checkbox"/>	<a href="#">AdministratorAcc...</a>	AWS managed	None	Grants account administrative perm
<input type="checkbox"/>	<a href="#">AdministratorAcc...</a>	AWS managed	None	Grants account administrative perm
<input type="checkbox"/>	<a href="#">AlexaForBusinessD...</a>	AWS managed	None	Provide device setup access to Alex
<input type="checkbox"/>	<a href="#">AlexaForBusinessF...</a>	AWS managed	None	Grants full access to AlexaForBusin
<input type="checkbox"/>	<a href="#">AlexaForBusinessG...</a>	AWS managed	None	Provide gateway execution access t
<input type="checkbox"/>	<a href="#">AlexaForBusinessLi...</a>	AWS managed	None	Provide access to Lifesize AVS devic
<input type="checkbox"/>	<a href="#">AlexaForBusinessP...</a>	AWS managed	None	Provide access to Poly AVS devices
<input type="checkbox"/>	<a href="#">AlexaForBusinessR...</a>	AWS managed	None	Provide read only access to AlexaFo
<input type="checkbox"/>	<a href="#">AmazonAPICreate...</a>	AWS managed	None	Provides full access to create/edit/c
<input type="checkbox"/>	<a href="#">AmazonAPICreate...</a>	AWS managed	None	Provides full access to invoke API c

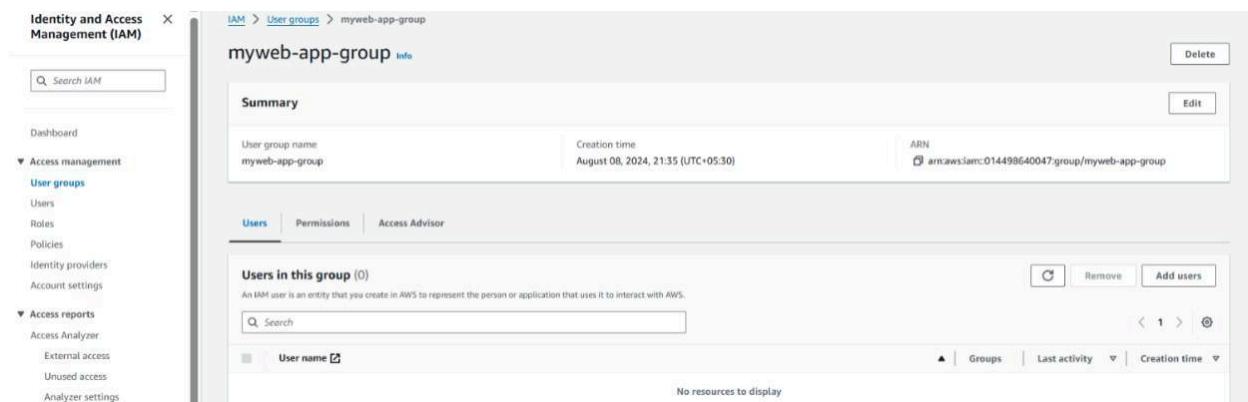
[Create policy](#)

[Cancel](#) [Create user group](#)

Step 7: Once the user group is created click on the server link created next to the user.

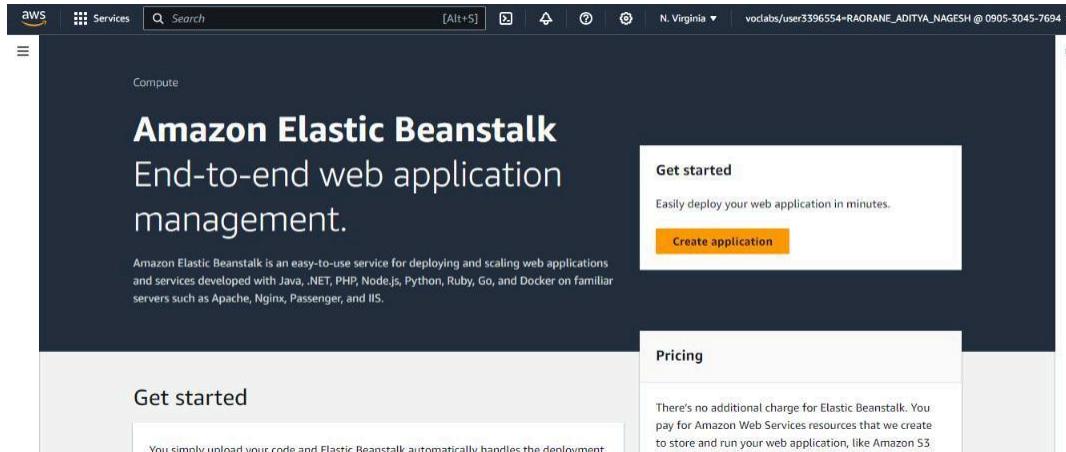


Step 8: Finally search for “AWSCloud9EnvironmentMember” policy and attach it.



## **Expt 2]Application Deployment with AWS Elastic Beanstalk and AWS CodePipeline**

Step 1: To open AWS Elastic Beanstalk, log in to the AWS Management Console, navigate to Elastic Beanstalk under "Compute," and click "Create Application" to start a new project

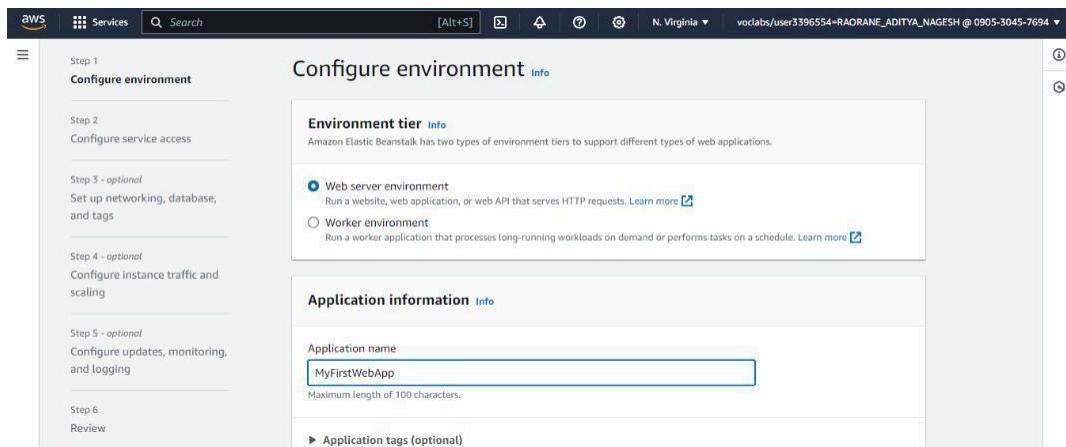


2] After clicking "Create Application" in Elastic Beanstalk, you'll choose between two environment tiers:

**Web Server Environment:** Handles HTTP requests for web applications with built-in load balancing, auto-scaling, and a pre-configured web server.

**Worker Environment:** Manages background tasks asynchronously, processing jobs from an SQS queue without direct user interaction.

Click on "Web server Environment".



Step 3: Under Platform, you can choose between Managed Platform and Custom Platform.

Selected Managed Platform and configured it with PHP for ease of use and automatic updates. **Managed Platform:** AWS manages and updates the environment, providing pre-configured platforms like PHP, making deployment easier and maintenance automated.

**Custom Platform:** Allows you to create and manage your own platform, giving full control over the environment configuration and software stack.

**Platform Info**

Platform type

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

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

Platform

PHP

Platform branch

PHP 8.3 running on 64bit Amazon Linux 2023

Platform version

4.3.1 (Recommended)

**Step 4:Presets** offer different deployment configurations to match your needs.

**Single Instance (Free Tier Eligible)** runs your application on one EC2 instance, suitable for development or low-traffic scenarios and eligible for the AWS Free Tier.

**Single Instance** deploys on one EC2 instance but without Free Tier limitations, ideal for small-scale projects.

**High Availability** uses multiple instances and a load balancer to ensure fault tolerance and reliability for production environments.

**Using Spot Instances** leverages unused EC2 capacity at reduced costs, suitable for applications that can handle interruptions and require cost savings.

**Application code Info**

Sample application

Existing version  
Application versions that you have uploaded.

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

**Presets Info**

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

Configuration presets

Single instance (free tier eligible)

Single instance (using spot instance)

High availability

High availability (using spot and on-demand instances)

Custom configuration

Cancel **Next**

**Step 5:** Define your application's service configuration, including settings for load balancing, auto-scaling, and health checks. Each EC2 instance can be accessed using its corresponding key pair. Choosing an EC2 key pair as a "vockey" (or a "virtual key") is typically done to enable secure access to your EC2 instances; allowing you to securely connect to your instances using SSH (Linux), ensuring that only authorized users can access the server.

The screenshot shows the AWS Elastic Beanstalk configuration interface. The left sidebar lists steps: Step 2 (Configure service access), Step 3 (optional: Set up networking, database, and tags), Step 4 (optional: Configure instance traffic and scaling), Step 5 (optional: Configure updates, monitoring, and logging), and Step 6 (Review). The main panel is titled 'Configure service access' and contains the following fields:

- Service role:** A dropdown menu showing 'Create and use new service role' and 'Use an existing service role'. The 'Use an existing service role' option is selected and set to 'LabRole'.
- EC2 key pair:** A dropdown menu showing 'Select an EC2 key pair to securely log in to your EC2 instances'. The value is 'vockey'.
- EC2 instance profile:** A dropdown menu showing 'Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations'. The value is 'LabInstanceProfile'.

**Step 6:** Configure VPC, subnets, and security groups for network access. Set up your environment's database if needed and apply tags for resource management.

The screenshot shows the AWS Elastic Beanstalk configuration interface. The left sidebar lists steps: Step 1 (Configure environment), Step 2 (Configure service access), Step 3 (optional: Set up networking, database, and tags), Step 4 (optional: Configure instance traffic and scaling), Step 5 (optional: Configure updates, monitoring, and logging), and Step 6 (Review). The main panel is titled 'Set up networking, database, and tags - optional' and contains the following sections:

- Virtual Private Cloud (VPC):** A section for setting up a custom VPC. It shows a dropdown menu with the value 'vpc-0848777a13d544a21 | (172.31.0.0/16)' and a link to 'Create custom VPC'.
- Instance settings:** A section for configuring instance subnet settings. It includes a note about avoiding public exposure and a checkbox for 'Activated' which is checked.
- Instance subnets:** A section for managing instance subnets, currently showing a single entry.

**Step 7:** Adjust load balancer settings to manage incoming traffic and configure auto-scaling rules to handle changes in traffic volume. The **root volume type** is the primary storage for the OS, and selecting **General Purpose (SSD)** offers balanced performance and durability for most applications.

The screenshot shows the AWS Elastic Beanstalk configuration interface. On the left, a sidebar lists steps from 1 to 6. Step 4 is selected, titled "Configure instance traffic and scaling". The main content area is titled "Configure instance traffic and scaling - optional". It contains a section for "Instances" where the "Root volume type" is set to "General Purpose (SSD)" and the "Size" is 8 GB. Below this, there's a "Throughput" section with a slider set at 125 MiB/s. At the bottom right, there are links for "CloudShell", "Feedback", and copyright information.

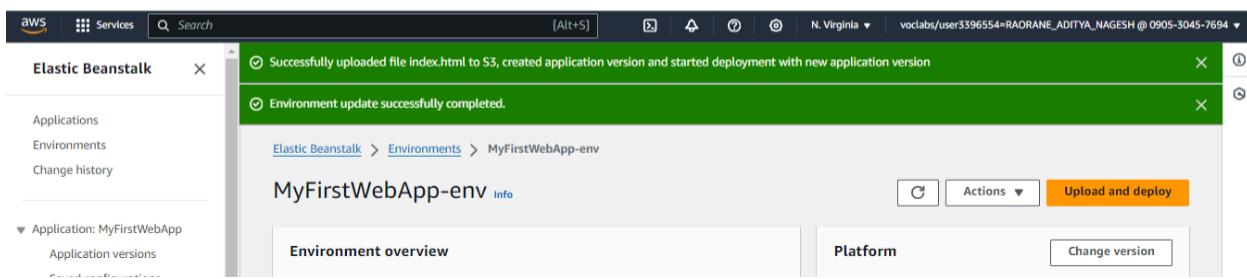
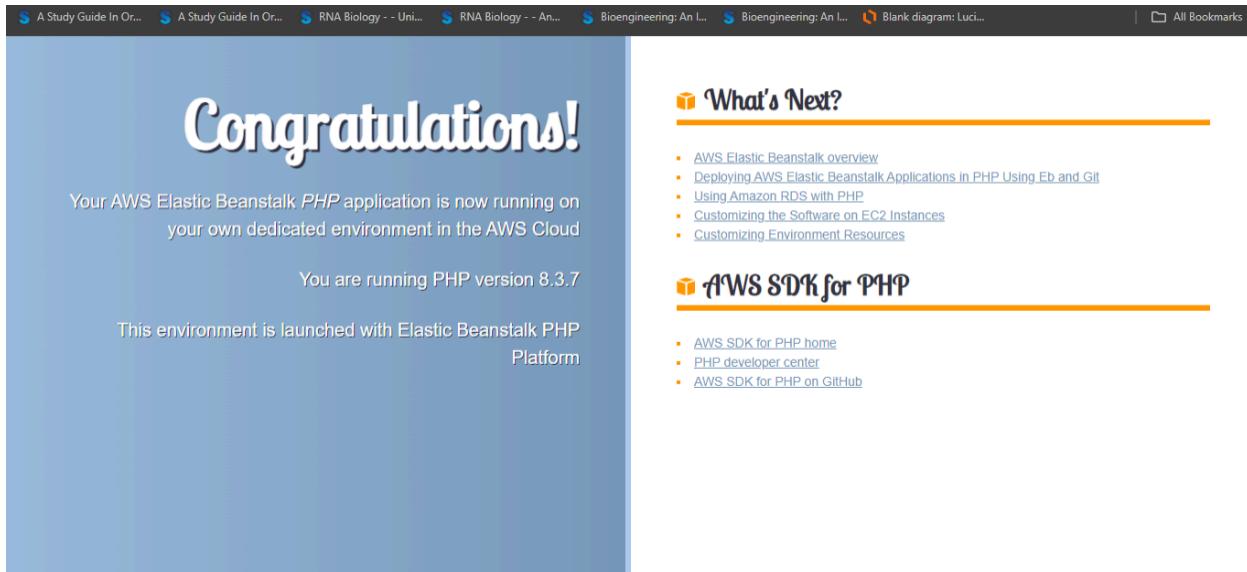
**Step 8:** Set up deployment policies for updates, enable monitoring for performance metrics, and configure logging to capture and analyze application logs.

The screenshot shows the AWS Elastic Beanstalk configuration interface. Step 5 is selected, titled "Configure updates, monitoring, and logging". The main content area is titled "Configure updates, monitoring, and logging - optional". It has a "Monitoring" section with "Health reporting" enabled (Basic) and a "Health event streaming to CloudWatch Logs" section. There are also sections for "Log streaming" (activated), "Retention" (7 days), and "Lifecycle". The footer includes links for "CloudShell", "Feedback", and copyright information.

Step 9: This creates our environment .We can now check our sample application uploaded if our environment is successfully deployed.

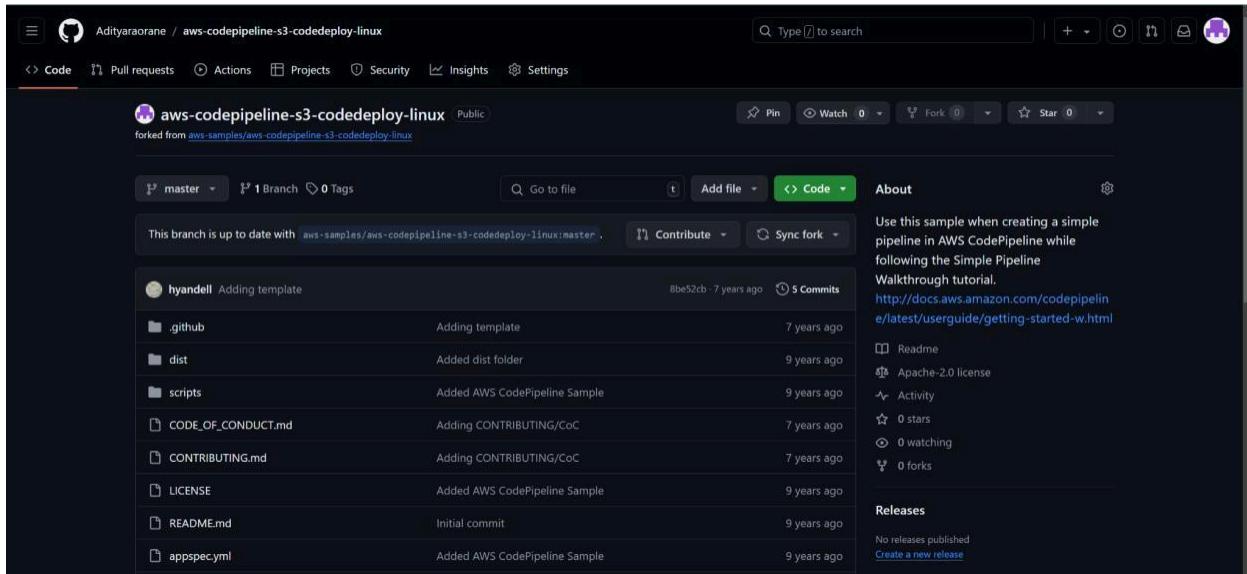
The screenshot shows the AWS Elastic Beanstalk console for the environment 'MyFirstWebApp-env'. The left sidebar shows the application 'MyFirstWebApp' and its environment 'MyFirstWebApp-env'. The main area displays the 'Environment overview' and the 'Events' tab. The 'Events' tab shows four INFO-level events from August 5, 2024, at 14:20:47 UTC+5:30, detailing the creation of security groups, SNS notification topics, and storage buckets, along with the start of the environment creation.

Time	Type	Details
August 5, 2024 14:20:47 (UTC+5:30)	INFO	Created security group named: sg-027c14bb95036c61
August 5, 2024 14:20:29 (UTC+5:30)	INFO	Created SNS Notification Topic. ARN: arn:aws:sns:us-east-1:090530457694:ElasticBeanstalkNotifications-Environment-MyFirstWebApp-env
August 5, 2024 14:20:20 (UTC+5:30)	INFO	Using elasticbeanstalk-us-east-1-090530457694 as Amazon S3 storage bucket for environment data.
August 5, 2024 14:20:19 (UTC+5:30)	INFO	createEnvironment is starting.



**Step 10:** Go to the github link below. This is a github with a sample code for deploying a file on AWSCodePipeline. Fork this repository into your personal github.

<https://github.com/aws-samples/aws-codepipeline-s3-codedeploy-linux>



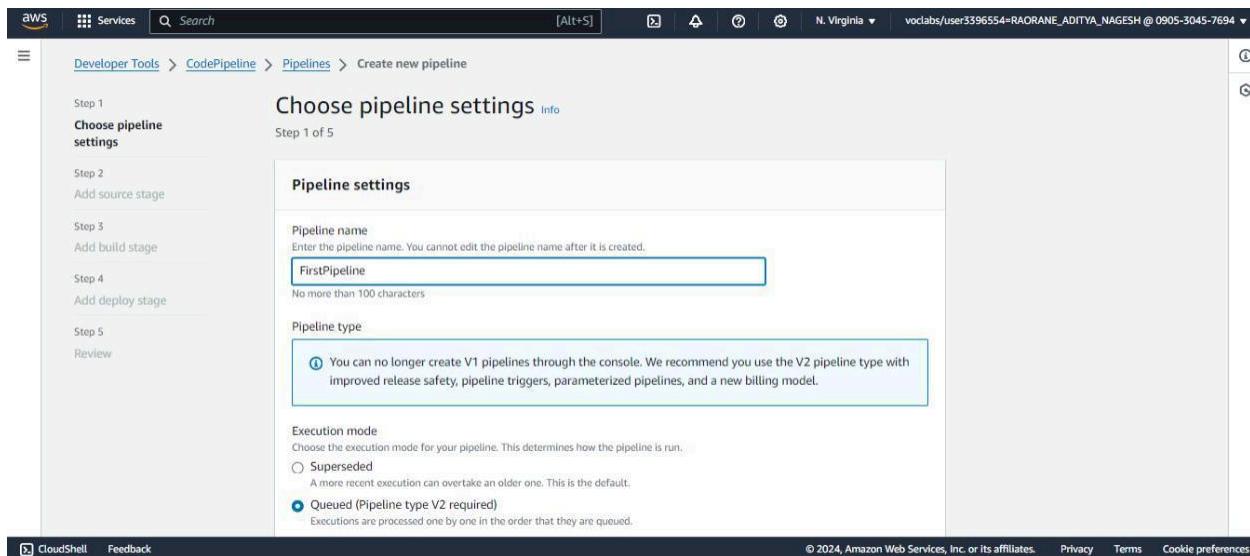
Step 11: Search **CodePipeline** in the “services” tab and click on it.

Step 12: After navigation to CodePipeline under “Developer Tools,” and click “Create pipeline” to start setting up your CI/CD.

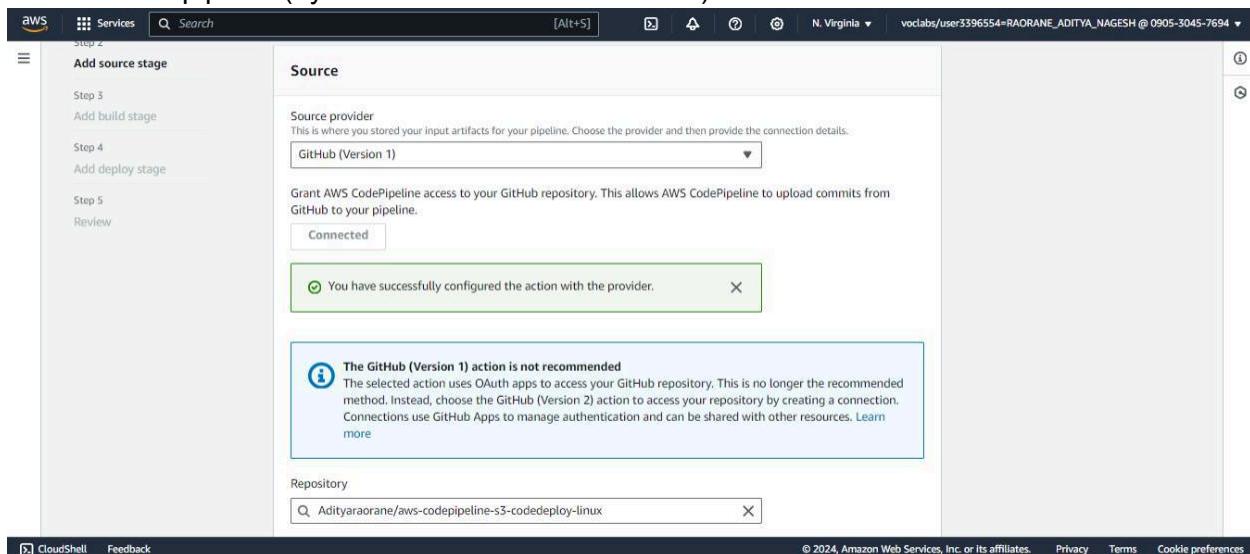
Step 13: Give a name to your Pipeline. A new service role would be created with the name of the pipeline. The **pipeline type** options are:

- **Suspended:** Pauses the pipeline, stopping all execution of actions and deployments until resumed.
- **Queued:** Holds pipeline executions in a queue, executing them sequentially. This is useful for managing limited resources and preventing overload.
- **Parallel:** Runs multiple pipeline executions concurrently, allowing for faster processing but potentially higher resource usage.

Selecting **Queued** helps manage resource allocation by ensuring that pipelines run one at a time, avoiding conflicts and optimizing resource usage. Finally, click on “**Next**” at the bottom.



**Step 14:** In the Source stage of AWS CodePipeline, select GitHub(Version 1) as your source provider, then connect your GitHub account and choose the repository and branch you want to use for the pipeline(by default the branch is **master**). Then click on “**Next**”.



**Step 15:** Set the Trigger type as no filter. This would allow it to the website to update as soon as some change is made in the github.

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

**Output artifact format**  
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.

**Trigger**

**Trigger type**  
Choose the trigger type that starts your pipeline.

**No filter**  
Starts your pipeline on any push and clones the HEAD.

**Specify filter**  
Starts your pipeline on a specific filter and clones the exact commit. Pipeline type V2 is required.

**Do not detect changes**  
Don't automatically trigger the pipeline.

**Info:** You can add additional sources and triggers by editing the pipeline after it is created.

Cancel Previous Next

Step 16: Skip the build stage and go to Deploy. Select the deploy provider as AWS Elastic Beanstalk and Input Artifact as SourceArtifact. The application name would be the name of your Elastic Beanstalk. Then click on next.

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1 Choose pipeline settings Step 3 of 5

Step 2 Add source stage

Step 3 Add build stage

Step 4 Add deploy stage

Step 5 Review

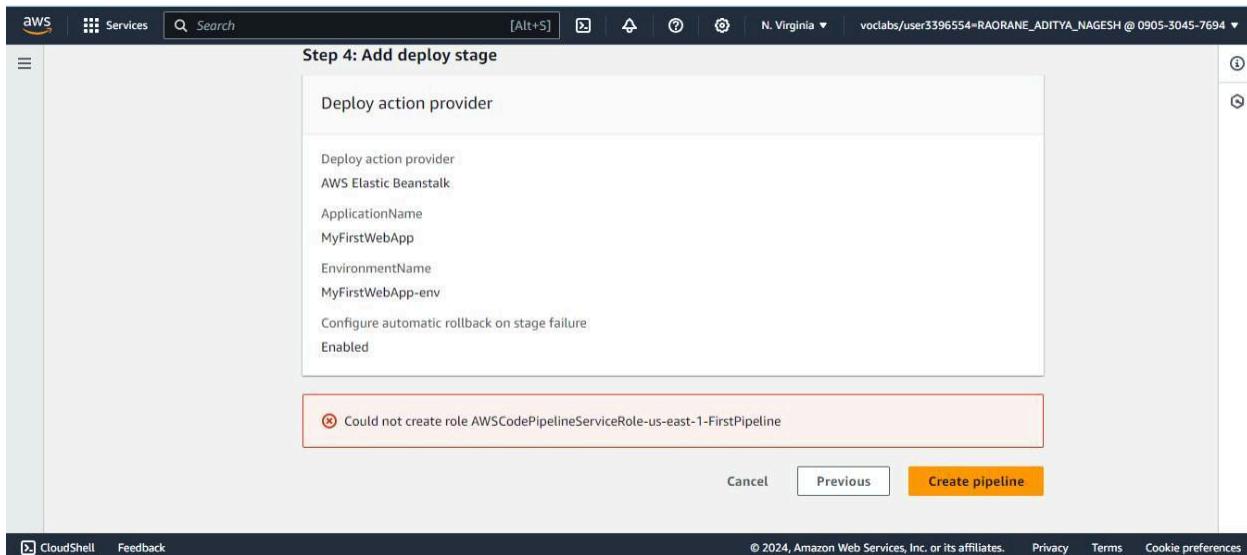
**Add build stage** Info

**Build - optional**

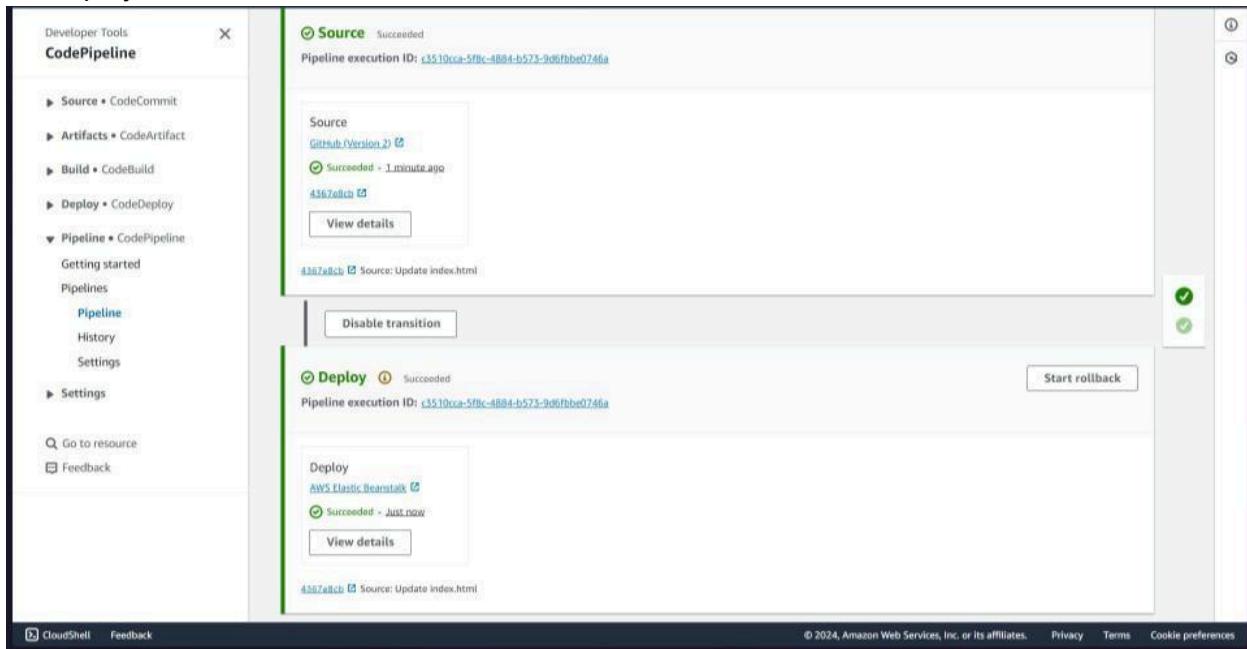
**Build provider**  
This is the tool of your build project. Provide build artifact details like operating system, build spec file, and output file names.

Cancel Previous Skip build stage Next

Step 17: The error occurred during the deployment stage because AWS Academy accounts often have **restricted IAM permissions**, preventing the creation of roles needed for the pipeline. This limitation can cause issues when configuring pipelines that require specific roles for execution. To resolve the error, deploy the pipeline using your personal AWS account, which typically has broader permissions and can create the necessary roles for successful deployment.



11]Once the source and deploy stages are successfully completed in AWS CodePipeline, you will see a message indicating "Review" to verify and confirm the pipeline setup before finalizing the deployment.



12]This will successfully show the sample website hosted.

# Congratulations!

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

For next steps, read the AWS CodePipeline Documentation.

**Note:** We can make some changes to the index.html file in the github. Once the changes are committed, when the website is refreshed, the changes can be seen.

## Experiment 03

**Aim:** To understand the Kubernetes Cluster Architecture, install and spin up a Kubernetes Cluster on Linux Machines/Cloud.

1] Create 3 EC2 instances with all running on Amazon Linux as OS with inbound SSH allowed. To efficiently run the machines it is advised to select the instance type as t2.medium which comes with 4 GiB of memory and 2 vCPU's.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IP
MyFirstWebApp...	i-0711ca634e9c1b607	Running	t2.small	2/2 checks passed	View alarms +	us-east-1e	ec2-54-158-219-89.co...	54.158.2
slave2	i-083e84e34cd3b9017	Running	t2.micro	Initializing	View alarms +	us-east-1b	ec2-18-212-120-149.co...	18.212.1
master	i-027085adeca04aa30	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	ec2-54-89-85-90.comp...	54.89.85
slave1	i-0e9f39c54022795b5	Running	t2.micro	Initializing	View alarms +	us-east-1b	ec2-54-198-212-237.co...	54.198.2

2] Update the inbound rules for the security groups on all three machines to allow TCP traffic on port 6443 from source 0.0.0.0/0.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0ab9d64ab3685970b	SSH	TCP	22	Custom	0.0.0.0/0
-	Custom TCP	TCP	6443	Anywhere	0.0.0.0/0

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

Cancel Preview changes Save rules

3] SSH into all 3 machines each in a separate terminal.

```
master          x + - x
Microsoft Windows [Version 10.0.22621.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\adity>cd Downloads

C:\Users\adity>Downloads>ssh -i "aditya.pem" ec2-user@ec2-35-175-239-211.compute-1.amazonaws.com
The authenticity of host 'ec2-35-175-239-211.compute-1.amazonaws.com' (35.175.239.211) can't be established.
ED25519 key fingerprint is SHA256:WDE1HWu02mnikn5MGmFWZPLcUzK2mtGfQ8B+bY/KIg.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-35-175-239-211.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

#_ _ _ _ _ Amazon Linux 2023
#_ _ _ _ _ #####\_
#_ _ _ _ _ \###\_
#_ _ _ _ _ /#\_
#_ _ _ _ V\---> https://aws.amazon.com/linux/amazon-linux-2023
#_ _ _ _ / \
#_ _ _ _ / \
#_ _ _ _ / \
#_ _ _ _ / \
[ec2-user@ip-172-31-34-60 ~]$


slave          x + - x
C:\Users\adity\Downloads>ssh -i "aditya.pem" ec2-user@ec2-54-152-106-104.compute-1.amazonaws.com
The authenticity of host 'ec2-54-152-106-104.compute-1.amazonaws.com' (54.152.106.104) can't be established.
ED25519 key fingerprint is SHA256:z7dnZ3c8gcKjVn84c8LLLZD+hw3kHTJhsBesoGNYg.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-152-106-104.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

#_ _ _ _ _ Amazon Linux 2023
#_ _ _ _ _ #####\_
#_ _ _ _ _ \###\_
#_ _ _ _ _ /#\_
#_ _ _ _ V\---> https://aws.amazon.com/linux/amazon-linux-2023
#_ _ _ _ / \
#_ _ _ _ / \
#_ _ _ _ / \
#_ _ _ _ / \
[ec2-user@ip-172-31-36-224 ~]$ |


slave2         x + - x
C:\Users\adity\Downloads>ssh -i "aditya.pem" ec2-user@ec2-18-206-215-250.compute-1.amazonaws.com
The authenticity of host 'ec2-18-206-215-250.compute-1.amazonaws.com' (18.206.215.256) can't be established.
ED25519 key fingerprint is SHA256:oyKLdnPGNg0q6az0lgZFh1KwG4Wco5dNALpNA6RFnw.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-206-215-250.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

#_ _ _ _ _ Amazon Linux 2023
#_ _ _ _ _ #####\_
#_ _ _ _ _ \###\_
#_ _ _ _ _ /#\_
#_ _ _ _ V\---> https://aws.amazon.com/linux/amazon-linux-2023
#_ _ _ _ / \
#_ _ _ _ / \
#_ _ _ _ / \
#_ _ _ _ / \
[ec2-user@ip-172-31-35-52 ~]$ |
```

4] To install Docker, use the following command:

```
sudo yum install docker -y
```

```
[tec2-user@ip-172-31-34-60 ~]$ sudo yum install docker -y
Last metadata expiration check: 0:04:50 ago on Tue Sep 17 18:44:28 2024.
Dependencies resolved.
=====
Package           Arch      Version       Repository   Size
=====
Installing:
  docker          x86_64    25.0.6-1.amzn2023.0.2  amazonlinux   44 M
Installing dependencies:
  containerd      x86_64    1.7.20-1.amzn2023.0.1  amazonlinux  35 M
  iptables-libc
  iptables-nft
  libcgroup
  libnetfilter_conntrack
  libnfnetworklink
  libnftnl
  pigz
  runc
=====
Transaction Summary
=====
Install 10 Packages

Total download size: 84 M
Installed size: 317 M
Downloaded Packages:
(1/10): iptables-libc-1.8.8-3.amzn2023.0.2.x86_64.rpm 4.0 MB/s | 401 kB 00:00
(2/10): iptables-nft-1.8.8-3.amzn2023.0.2.x86_64.rpm 4.7 MB/s | 183 kB 00:00
(3/10): libcgroup-3.0-1.amzn2023.0.1.x86_64.rpm 3.4 kB/s | 75 kB 00:00
(4/10): libnetfilter_conntrack-1.0-8-2.amzn2023.0.2.x86_64.rpm 2.6 kB/s | 58 kB 00:00
(5/10): libnfnetworklink-1.0-19.amzn2023.0.2.x86_64.rpm 1.2 kB/s | 30 kB 00:00
(6/10): libnftnl-1.2.2-2.amzn2023.0.2.x86_64.rpm 2.2 kB/s | 84 kB 00:00
(7/10): pigz-2.5-1.amzn2023.0.3.x86_64.rpm 762 kB/s | 83 kB 00:00
(8/10): runc-1.1.13-1.amzn2023.0.1.x86_64.rpm 25 kB/s | 3.2 MB 00:00
(9/10): containerd-1.7.20-1.amzn2023.0.1.x86_64.rpm 35 kB/s | 35 kB 00:01
(10/10): docker-25.0.6-1.amzn2023.0.2.x86_64.rpm 33 kB/s | 44 MB 00:01
=====
Total                                         60 MB/s | 84 MB 00:01
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing : 1/1
  Installing : runc-1.1.13-1.amzn2023.0.1.x86_64 1/10
=====
[slave1] Verifying : libcgroup-3.0-1.amzn2023.0.1.x86_64 5/10
[slave1] Verifying : libnetfilter_conntrack-1.0-8-2.amzn2023.0.2.x86_64 6/10
[slave1] Verifying : libnfnetworklink-1.0-19.amzn2023.0.2.x86_64 7/10
[slave1] Verifying : libnftnl-1.2.2-2.amzn2023.0.2.x86_64 8/10
[slave1] Verifying : pigz-2.5-1.amzn2023.0.3.x86_64 9/10
[slave1] Verifying : runc-1.1.13-1.amzn2023.0.1.x86_64 10/10
=====
Installed:
  containerd-1.7.20-1.amzn2023.0.1.x86_64
  docker-25.0.6-1.amzn2023.0.2.x86_64
  iptables-libc-1.8.8-3.amzn2023.0.2.x86_64
  iptables-nft-1.8.8-3.amzn2023.0.2.x86_64
  libcgroup-3.0-1.amzn2023.0.1.x86_64
  libnetfilter_conntrack-1.0-8-2.amzn2023.0.2.x86_64
  libnfnetworklink-1.0-19.amzn2023.0.2.x86_64
  libnftnl-1.2.2-2.amzn2023.0.2.x86_64
  pigz-2.5-1.amzn2023.0.3.x86_64
  runc-1.1.13-1.amzn2023.0.1.x86_64
=====
[slave2] Completel [tec2-user@ip-172-31-36-224 ~]$ |
[slave2] Verifying : libcgroup-3.0-1.amzn2023.0.1.x86_64 5/10
[slave2] Verifying : libnetfilter_conntrack-1.0-8-2.amzn2023.0.2.x86_64 6/10
[slave2] Verifying : libnfnetworklink-1.0-19.amzn2023.0.2.x86_64 7/10
[slave2] Verifying : libnftnl-1.2.2-2.amzn2023.0.2.x86_64 8/10
[slave2] Verifying : pigz-2.5-1.amzn2023.0.3.x86_64 9/10
[slave2] Verifying : runc-1.1.13-1.amzn2023.0.1.x86_64 10/10
=====
Installed:
  containerd-1.7.20-1.amzn2023.0.1.x86_64
  docker-25.0.6-1.amzn2023.0.2.x86_64
  iptables-libc-1.8.8-3.amzn2023.0.2.x86_64
  iptables-nft-1.8.8-3.amzn2023.0.2.x86_64
  libcgroup-3.0-1.amzn2023.0.1.x86_64
  libnetfilter_conntrack-1.0-8-2.amzn2023.0.2.x86_64
  libnfnetworklink-1.0-19.amzn2023.0.2.x86_64
  libnftnl-1.2.2-2.amzn2023.0.2.x86_64
  pigz-2.5-1.amzn2023.0.3.x86_64
  runc-1.1.13-1.amzn2023.0.1.x86_64
=====
[slave2] Completel [tec2-user@ip-172-31-35-52 ~]$ |
```

5] Then, configure cgroup in a daemon.json file by using the following commands. This allows kubernetes to manage host more efficiently

- cd /etc/docker

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

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

EOF

The image shows three terminal windows labeled 'master', 'slave1', and 'slave2'. Each window displays the command 'cat <<EOF | sudo tee /etc/docker/daemon.json' followed by the JSON configuration provided above. The output in each window shows the command being run and the configuration being written to the file.

```
[ec2-user@ip-172-31-34-60 docker]$ cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
EOF
[ec2-user@ip-172-31-34-60 docker]$
```

```
[ec2-user@ip-172-31-36-224 ~]$ cd /etc/docker
[ec2-user@ip-172-31-36-224 docker]$ cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
EOF
[ec2-user@ip-172-31-36-224 docker]$
```

```
[ec2-user@ip-172-31-35-52 ~]$ cd /etc/docker
[ec2-user@ip-172-31-35-52 docker]$ cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
EOF
[ec2-user@ip-172-31-35-52 docker]$
```

6] After configuring restart docker service service :

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

```

master      x + v
[ec2-user@ip-172-31-34-60 docker]$ cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
EOF
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
[ec2-user@ip-172-31-34-60 docker]$ sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker
docker -v
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Docker version 25.0.5, build 5dc9bcc
[ec2-user@ip-172-31-34-60 docker]$ |
```

```

slave1      x + v
"max-size": "100m"
},
"storage-driver": "overlay2"
}
EOF
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
[ec2-user@ip-172-31-36-224 docker]$ sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker
docker -v
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Docker version 25.0.5, build 5dc9bcc
[ec2-user@ip-172-31-36-224 docker]$ |
```

```

slave2      x + v
"max-size": "100m"
},
"storage-driver": "overlay2"
}
EOF
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
[ec2-user@ip-172-31-35-52 docker]$ sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker
docker -v
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Docker version 25.0.5, build 5dc9bcc
[ec2-user@ip-172-31-35-52 docker]$ |
```

7] SELinux needs to be disabled before configuring kubelet to avoid interference with kubernetes api server

- sudo setenforce 0
  - sudo sed -i 's/^SELINUX=enforcing\$/SELINUX=permissive/' /etc/selinux/config
- Add kubernetes repository (paste in terminal)
- ```

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

```

8] Type following commands to install set of kubernetes packages:

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

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

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

- sudo sysctl -p

```

master                         slave1                         slave2
Transaction check succeeded.   Verifying : kubelet-1.30.5-150500.1.1.x86_64      8/9
Running transaction test      Verifying : kubernetes-cni-1.4.0-150500.1.1.x86_64      9/9
Transaction test succeeded.   Installed: contrack-tools-1.4.6-2.amzn2023.0.2.x86_64
Running transaction           cri-tools-1.30.1-150500.1.1.x86_64      1/9
transaction.                 libnetfilter_ctimeout-1.0.5-2.amzn2023.0.2.x86_64      2/9
Preparing : kubernetes-cni-1.4.0-150500.1.1.x86_64      1/1
Installing : cri-tools-1.30.1-150500.1.1.x86_64      1/9
Installing : libnetfilter_ctimeout-1.0.5-2.amzn2023.0.2.x86_64      2/9
Installing : libnetfilter_ctimeout-1.0.0-19.amzn2023.0.2.x86_64      3/9
Installing : libnetfilter_ctimeout-1.0.0-21.amzn2023.0.2.x86_64      4/9
Installing : libnetfilter_ctimeout-1.0.0-21.amzn2023.0.2.x86_64      5/9
Running scriptlet: contrack-tools-1.4.6-2.amzn2023.0.2.x86_64      6/9
Installing : kubelet-1.30.5-150500.1.1.x86_64      6/9
Running scriptlet: kubelet-1.30.5-150500.1.1.x86_64      7/9
Installing : kubelet-1.30.5-150500.1.1.x86_64      8/9
Running scriptlet: kubelet-1.30.5-150500.1.1.x86_64      9/9
Verifying   : kubelet-1.30.5-150500.1.1.x86_64      1/9
Verifying   : libnetfilter_ctimeout-1.0.0-19.amzn2023.0.2.x86_64      2/9
Verifying   : libnetfilter_ctimeout-1.0.0-19.amzn2023.0.2.x86_64      3/9
Verifying   : libnetfilter_ctimeout-1.0.5-2.amzn2023.0.2.x86_64      4/9
Verifying   : cri-tools-1.30.1-150500.1.1.x86_64      5/9
Verifying   : kubelet-1.30.5-150500.1.1.x86_64      6/9
Verifying   : kubelet-1.30.5-150500.1.1.x86_64      7/9
Verifying   : kubernetes-cni-1.4.0-150500.1.1.x86_64      8/9
Verifying   : kubernetes-cni-1.4.0-150500.1.1.x86_64      9/9

Installed:
contrack-tools-1.4.6-2.amzn2023.0.2.x86_64
cri-tools-1.30.1-150500.1.1.x86_64
kubeadm-1.30.5-150500.1.1.x86_64
kubectl-1.30.5-150500.1.1.x86_64
kubelet-1.30.5-150500.1.1.x86_64
kubernetes-cni-1.4.0-150500.1.1.x86_64
libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64
libnetfilter_ctimeout-1.0.0-19.amzn2023.0.2.x86_64
libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64

Complete!
[ec2-user@ip-172-31-36-224 docker]$ 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
[ec2-user@ip-172-31-36-224 docker]$ |
```

```

slave1                         slave2
Verifying : kubelet-1.30.5-150500.1.1.x86_64      8/9
Verifying : kubernetes-cni-1.4.0-150500.1.1.x86_64      9/9
Installed:
contrack-tools-1.4.6-2.amzn2023.0.2.x86_64
cri-tools-1.30.1-150500.1.1.x86_64
kubeadm-1.30.5-150500.1.1.x86_64
kubectl-1.30.5-150500.1.1.x86_64
kubelet-1.30.5-150500.1.1.x86_64
kubernetes-cni-1.4.0-150500.1.1.x86_64
libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64
libnetfilter_ctimeout-1.0.0-19.amzn2023.0.2.x86_64
libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64

Complete!
[ec2-user@ip-172-31-36-224 docker]$ 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
[ec2-user@ip-172-31-36-224 docker]$ |
```

```

slave2
Verifying : kubelet-1.30.5-150500.1.1.x86_64      8/9
Verifying : kubernetes-cni-1.4.0-150500.1.1.x86_64      9/9
Installed:
contrack-tools-1.4.6-2.amzn2023.0.2.x86_64
cri-tools-1.30.1-150500.1.1.x86_64
kubeadm-1.30.5-150500.1.1.x86_64
kubectl-1.30.5-150500.1.1.x86_64
kubelet-1.30.5-150500.1.1.x86_64
kubernetes-cni-1.4.0-150500.1.1.x86_64
libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64
libnetfilter_ctimeout-1.0.0-19.amzn2023.0.2.x86_64
libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64

Complete!
[ec2-user@ip-172-31-35-52 docker]$ 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
[ec2-user@ip-172-31-35-52 docker]$ |
```

### 10] Perform this ONLY on the MASTER

**machine:-a) Initialize Kubernetes By Typing Below**

Command:

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

```
[ec2-user@ip-172-31-34-60 docker]$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=all
I0917 18:58:49.232613    27391 version.go:256] remote version is much newer: v1.31.0; falling back to: stable-1.30
[init] Using Kubernetes version: v1.30.5
[preflight] Running pre-flight checks
    [WARNING NumCPU]: the number of available CPUs 1 is less than the required 2
    [WARNING Mem]: the system RAM (949 MB) is less than the minimum 1700 MB
    [WARNING FileExisting-socat]: socat not found in system path
    [WARNING FileExisting-tc]: tc not found in system path
    [WARNING Service-Kubelet]: kubelet service is not enabled, please run 'systemctl enable kubelet.service'
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
W0917 18:58:49.564960    27391 checks.go:844] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent with that used by kubeadm. It is recommended to use "registry.k8s.io/pause:3.9" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-34-60.ec2.internal kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 172.31.34.60]
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key
[certs] Generating "etcd/ca" certificate and key
[certs] Generating "etcd/server" certificate and key
[certs] etcd/server serving cert is signed for DNS names [ip-172-31-34-60.ec2.internal localhost] and IPs [172.31.34.60 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-34-60.ec2.internal localhost] and IPs [172.31.34.60 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
[certs] Generating "sa" key and public key
[kubeconfig] Using kubeconfig folder "/etc/kubernetes"
[kubeconfig] Writing "admin.conf" kubeconfig file
[kubeconfig] Writing "super-admin.conf" kubeconfig file
[kubeconfig] Writing "kubelet.conf" kubeconfig file
[kubeconfig] Writing "controller-manager.conf" kubeconfig file
[kubeconfig] Writing "scheduler.conf" kubeconfig file
```

b) Copy the mkdir and chown commands from 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

```
[mark-control-plane] Marking the node ip-172-31-34-60.ec2.internal as control-plane by adding the labels: [node-role.kubernetes.io/control-plane node.kubernetes.io/exclude-from-external-load-balancers]
[mark-control-plane] Marking the node ip-172-31-34-60.ec2.internal as control-plane by adding the taints [node-role.kubernetes.io/control-plane:NoSchedule]
[bootstrap-token] Using token: f70o2w.perwilygbt7qu724
[bootstrap-token] Configuring bootstrap tokens, cluster-info ConfigMap, RBAC Roles
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to get nodes
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to post CSRs in order for nodes to get long term certificate credentials
[bootstrap-token] Configured RBAC rules to allow the csrapprover controller automatically approve CSRs from a Node Bootstrap Token
[bootstrap-token] Configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
[bootstrap-token] Creating the "cluster-info" ConfigMap in the "kube-public" namespace
[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable kubelet client certificate and key
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy

Your Kubernetes control-plane has initialized successfully!

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 172.31.34.60:6443 --token f70o2w.perwilygbt7qu724 \
    --discovery-token-ca-cert-hash sha256:df9aa1a237965f949c055717ba438799f890a8647b8cd10386fa7ed864e59096
[ec2-user@ip-172-31-34-60 docker]$ mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
[ec2-user@ip-172-31-34-60 docker]$
```

c)Copy the Kubernetes join command from your output to the clipboard

- kubeadm join 172.31.34.60:6443 --token  
f70o2w.perwilygbt7qu724 \ --discovery-token-ca-cert-hash  
sha256:df9aa1a237965f949c055717ba438799f890a8647b8cd10386fa7ed  
864e59096

d)Then, add a common networking plugin called flammel file as mentioned code.

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

```
[ec2-user@ip-172-31-34-60 docker]$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
[ec2-user@ip-172-31-34-60 docker]$ |
```

e)Check the created node using this command

- kubectl get nodes

```
[ec2-user@ip-172-31-34-60 docker]$ kubectl get nodes
NAME                  STATUS   ROLES      AGE      VERSION
ip-172-31-34-60.ec2.internal   Ready   control-plane   5m41s   v1.30.5
```

**11] Perform this ONLY on the WORKER machines:-a)Paste**

the below command on all 2 worker machines

- sudo yum install iproute-tc socat-y (necessary packages required by kubernetes )
- sudo systemctl enable kubelet
- sudo systemctl restart kubelet

```
slave1
[ec2-user@ip-172-31-36-224 docker]$ 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
[ec2-user@ip-172-31-36-224 docker]$ sudo yum install -y iproute-tc socat # Install necessary packages required by Kubernetes
sudo systemctl enable kubelet      # Enable kubelet to start on boot
sudo systemctl restart kubelet     # Restart the kubelet service
Last metadata expiration check: 0:07:42 ago on Tue Sep 17 18:55:36 2024.
Dependencies resolved.
=====
 Package          Architecture Version      Repository      Size
=====
Installing:
 iproute-tc      x86_64        5.10.0-2.amzn2023.0.5      amazonlinux    455 k
 socat           x86_64        1.7.4.2-1.amzn2023.0.2      amazonlinux    303 k
Transaction Summary
=====
Install 2 Packages

slave2
Total   5.2 MB/s | 758 kB   00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing :   1/1
Installing : socat-1.7.4.2-1.amzn2023.0.2.x86_64          1/2
Installing : iproute-tc-5.10.0-2.amzn2023.0.5.x86_64       2/2
Running scriptlet: iproute-tc-5.10.0-2.amzn2023.0.5.x86_64 2/2
Verifying   : iproute-tc-5.10.0-2.amzn2023.0.5.x86_64       1/2
Verifying   : socat-1.7.4.2-1.amzn2023.0.2.x86_64         2/2
Installed:
 iproute-tc-5.10.0-2.amzn2023.0.5.x86_64      socat-1.7.4.2-1.amzn2023.0.2.x86_64

Complete!
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /usr/lib/systemd/system/kubelet.service.
[ec2-user@ip-172-31-35-52 docker]$ |
```

b)kubeadm join 172.31.34.60:6443 --token f70o2w.perwi1ygbt7qu724 \  
 --discovery-token-ca-cert-hash  
 sha256:df9aa1a237965f949c055717ba438799f890a8647b8cd10386fa7ed  
 864e59096

```
slave1
[ec2-user@ip-172-31-36-224 docker]$ sudo kubeadm join 172.31.34.60:6443 --token f70o2w.perwi1ygbt7qu724 \  

--discovery-token-ca-cert-hash sha256:df9aa1a237965f949c055717ba438799f890a8647b8cd10386fa7ed  

864e59096 --v=5  

I0917 19:04:45.857329 28291 join.go:417] [preflight] found NodeName empty; using OS host  

name as NodeName  

I0917 19:04:45.858112 28291 initconfiguration.go:122] detected and using CRI socket:  

unix:///var/run/containerd/containerd.sock  

[preflight] Running pre-flight checks  

I0917 19:04:45.858444 28291 preflight.go:93] [preflight] Running general checks  

I0917 19:04:45.858592 28291 checks.go:278] validating the existence of file /etc/kube  

rnetes/kubelet.conf  

I0917 19:04:45.858911 28291 checks.go:278] validating the existence of file /etc/kube  

rnetes/bootstrap-kubelet.conf  

I0917 19:04:45.859018 28291 checks.go:102] validating the container runtime  

I0917 19:04:45.910166 28291 checks.go:637] validating whether swap is enabled or not  

I0917 19:04:45.910447 28291 checks.go:368] validating the presence of executable cric  

tl  

I0917 19:04:45.910556 28291 checks.go:368] validating the presence of executable conn  

track  

I0917 19:04:45.910709 28291 checks.go:368] validating the presence of executable ip  
  

slave2
Node in the cluster with name "ip-172-31-35-52.ec2.internal" and status "Ready"  

I0917 19:05:00.951572 28278 kubelet.go:175] [kubelet-start] Stopping the kubelet  

[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"  

[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/k  

ubeadm-flags.env"  

[kubelet-start] Starting the kubelet  

[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This c  

an take up to 4m0s  

[kubelet-check] The kubelet is healthy after 1.00439379s  

[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap  

I0917 19:06:45.405069 28278 kubelet.go:242] [kubelet-start] preserving the crisocket  

information for the node  

I0917 19:06:45.405225 28278 patchnode.go:31] [patchnode] Uploading the CRI Socket inf  

ormation "unix:///var/run/containerd/containerd.sock" to the Node API object "ip-172-31  

-35-52.ec2.internal" as an annotation  

I0917 19:06:45.405575 28278 cert_rotation.go:137] Starting client certificate rotatio  

n controller  
  

This node has joined the cluster:  

* Certificate signing request was sent to apiserver and a response was received.  

* The Kubelet was informed of the new secure connection details.
```

- c) With the help of command the worker nodes are connected to the master node and are ready to do tasks assigned by the master node.

Now we can see in the master/control node of kubernetes that worker nodes are connected by typing **watch kubectl get nodes** in the **master** node instance.

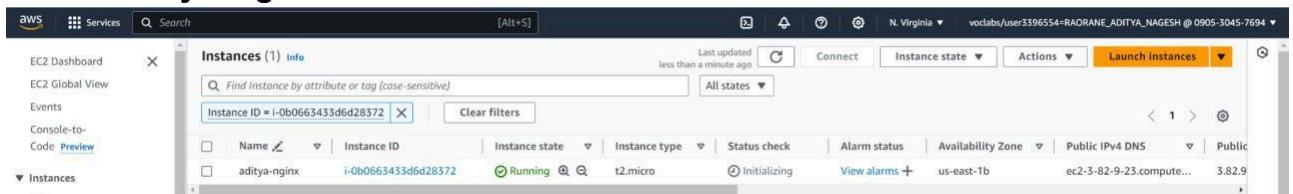
```
[ec2-user@ip-172-31-34-60 docker]$ kubectl get nodes
NAME                  STATUS   ROLES      AGE      VERSION
ip-172-31-34-60.ec2.internal   Ready    control-plane   9m18s   v1.30.5
ip-172-31-35-52.ec2.internal   Ready    <none>       112s    v1.30.5
ip-172-31-36-224.ec2.internal  Ready    <none>       110s    v1.30.5
[ec2-user@ip-172-31-34-60 docker]$
```

**Conclusion:** We commenced the installation and configuration of the essential Kubernetes packages. While some were accessible via the default Linux repositories, others required the addition of external repositories for proper installation. During the setup, we encountered an issue where the nodes were tainted, resulting in the Kubernetes API server crashing. This was mitigated by removing the taints from the nodes. Additionally, SELinux was disabled to prevent potential conflicts with Kubernetes operations. In conclusion, we successfully integrated the worker nodes with the Kubernetes master node, achieving a fully operational cluster setup.

## Experiment 04

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

1. Select **Amazon linux as OS** image and create an AWS **EC2 instance** named **aditya-nignx**.



2. Make ssh connection in terminal

```
C:\Users\adity\Downloads>ssh -i "aditya.pem" ec2-user@ec2-3-82-9-23.compute-1.amazonaws.com
The authenticity of host 'ec2-3-82-9-23.compute-1.amazonaws.com (3.82.9.23)' can't be established.
ED25519 key fingerprint is SHA256:DwfV9Db7wOQNID7+1XZ6AdAQt8EEo39x5Fjlg9DX7dE.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-82-9-23.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

  _#_
 /_###_      Amazon Linux 2023
 \_#####\
  \###|
   \|/--- https://aws.amazon.com/linux/amazon-linux-2023
    \~'`->
     ``_/
      _/_/
     _/`_/
 [ec2-user@ip-172-31-41-14 ~]$ |
```

### 3. Install Docker

- sudo dnf update -y && sudo dnf install -y docker && sudo systemctl enable docker && sudo systemctl start docker && sudo docker run hello-world

```
[ec2-user@ip-172-31-41-14 ~]$ sudo dnf update -y && sudo dnf install -y docker && sudo systemctl enable docker && sudo systemctl start docker && sudo docker run hello-world
Amazon Linux 2023 Kernel Livepatch repository
Dependencies resolved.
Nothing to do.
Complete!
Last metadata expiration check: 0:00:01 ago on Wed Sep 18 15:59:52 2024.
Dependencies resolved.
=====
Package           Architecture   Version      Repository    Size
=====
Installing:
  docker          x86_64        25.0.6-1.amzn2023.0.2  amazonlinux  44 M
Installing dependencies:
  containerd      x86_64        1.7.20-1.amzn2023.0.1  amazonlinux  35 M
  iptables        x86_64        1.8.8-3.amzn2023.0.2  amazonlinux  401 k
  iptables-nft    x86_64        1.8.8-3.amzn2023.0.2  amazonlinux  183 k
  libcgroup       x86_64        3.0-1.amzn2023.0.1   amazonlinux  75 k
  libnetfilter_conntrack x86_64        1.0.8-2.amzn2023.0.2  amazonlinux  58 k
  libnftnl        x86_64        1.0.1-19.amzn2023.0.2  amazonlinux  30 k
  libnftnl        x86_64        1.2.2-2.amzn2023.0.2  amazonlinux  84 k
  pigz            x86_64        2.5-1.amzn2023.0.3   amazonlinux  83 k
  runc            x86_64        1.1.13-1.amzn2023.0.1  amazonlinux  3.2 M
=====
Transaction Summary
=====
Install  10 Packages

Total download size: 84 M
Installed size: 317 M
```

```
[ec2-user@ip-172-31-41-14 ~]$ 
Complete!
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: 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.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
[ec2-user@ip-172-31-41-14 ~]$ |
```

Then, configure cgroup in a daemon.json file. This allows kubernetes to manage host more efficiently.

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

```
{  
  "exec-opts":  
    ["native.cgroupdriver=systemd"] } EOF
```

```
[ec2-user@ip-172-31-41-14 docker]$ sudo tee /etc/docker/daemon.json <<EOF  
{  
  "exec-opts": ["native.cgroupdriver=systemd"]  
}  
EOF  
{  
  "exec-opts": ["native.cgroupdriver=systemd"]  
}  
[ec2-user@ip-172-31-41-14 docker]$ |
```

- sudo systemctl daemon-reload
- sudo systemctl restart docker

```
[ec2-user@ip-172-31-41-14 docker]$ sudo systemctl daemon-reload  
sudo systemctl restart docker  
[ec2-user@ip-172-31-41-14 docker]$ |
```

## 4. Install Kubernetes

**Note:** I'm directly installing binary package you may install from package repository of your distribution

**Install CNI plugins (required for most pod network):**

```
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
```

```
[ec2-user@ip-172-31-41-14 docker]$ 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-${ARC
H}-${CNI_PLUGINS_VERSION}.tgz" | sudo tar -C "$DEST" -xz
% Total    % Received % Xferd  Average Speed   Time     Time      Current
          Dload  Upload Total Spent   Left Speed
0       0     0     0       0      0      0 --:--:-- --:--:-- --:--:-- 0
100 43.2M  100 43.2M     0      0  42.1M    0  0:00:01  0:00:01 --:--:-- 52.2M
[ec2-user@ip-172-31-41-14 docker]$ |
```

**Defne the directory to download command files:**

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

```
[ec2-user@ip-172-31-41-14 docker]$ 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):**

```
CRICTL_VERSION="v1.31.0"
```

```
ARCH="amd64"
```

```
curl -L
```

```
"https://github.com/kubernetes-sigs/cri-tools/releases/download/${CRICTL_VERSION}/crictl-${CRICTL_VERSION}-linux-${ARCH}.tar.gz" | sudo tar -C $DOWNLOAD_DIR -xz
```

```
[ec2-user@ip-172-31-41-14 docker]$ CRICTL_VERSION="v1.31.0" ARCH="amd64"
curl -L "https://github.com/kubernetes-sigs/cri-tools/releases/download/${CRICTL_VERSION}/crictl-${CRICTL_VERSION}-linux-${ARCH}.tar.gz" | sudo tar -C "$DOWNLOAD_DIR" -xz
% Total    % Received % Xferd  Average Speed   Time     Time      Current
          Dload  Upload Total Spent   Left Speed
  0       0     0      0      0      0 --:--:-- --:--:-- --:--:-- 0
100 17.5M  100 17.5M    0      0  31.3M      0 --:--:-- --:--:-- 66.4M
```

**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}
```

```
[ec2-user@ip-172-31-41-14 docker]$ 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}
% Total    % Received % Xferd  Average Speed   Time     Time      Current
          Dload  Upload Total Spent   Left Speed
100 138  100 138    0      0  1643      0 --:--:-- --:--:-- 1662
100 55.5M  100 55.5M    0      0  35.9M      0 0:00:01 0:00:01 --:--:-- 26.1M
% Total    % Received % Xferd  Average Speed   Time     Time      Current
          Dload  Upload Total Spent   Left Speed
100 138  100 138    0      0  3349      0 --:--:-- --:--:-- 3365
100 73.3M  100 73.3M    0      0  89.4M      0 --:--:-- --:--:-- 101M
```

```
RELEASE_VERSION="v0.16.2"
```

```
curl -sSL
```

```
"https://raw.githubusercontent.com/kubernetes/release/${RELEASE_VERSION}/cmd/kre
```

```
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/kre
```

```
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
```

```
[ec2-user@ip-172-31-41-14 bin]$ RELEASE_VERSION="v0.16.2"
curl -sSL "https://raw.githubusercontent.com/kubernetes/release/${RELEASE_VERSION}/cmd/krel/templates/latest/kubelet/kubelet.service" | sed "s:/usr/bin:${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/${RELEASE_VERSION}/cmd/krel/templates/latest/kubeadm/10-kubeadm.conf" | sed "s:/usr/bin:${DOWNLOAD_DIR}:g" | sudo tee /usr/lib/systemd/system/kubelet.service.d/10-kubeadm.conf
[Unit]
Description=kubelet: The Kubernetes Node Agent
Documentation=https://kubernetes.io/docs/
Wants=network-online.target
After=network-online.target

[Service]
ExecStart=/usr/local/bin/kubelet
Restart=always
StartLimitInterval=0
RestartSec=10

[Install]
WantedBy=multi-user.target

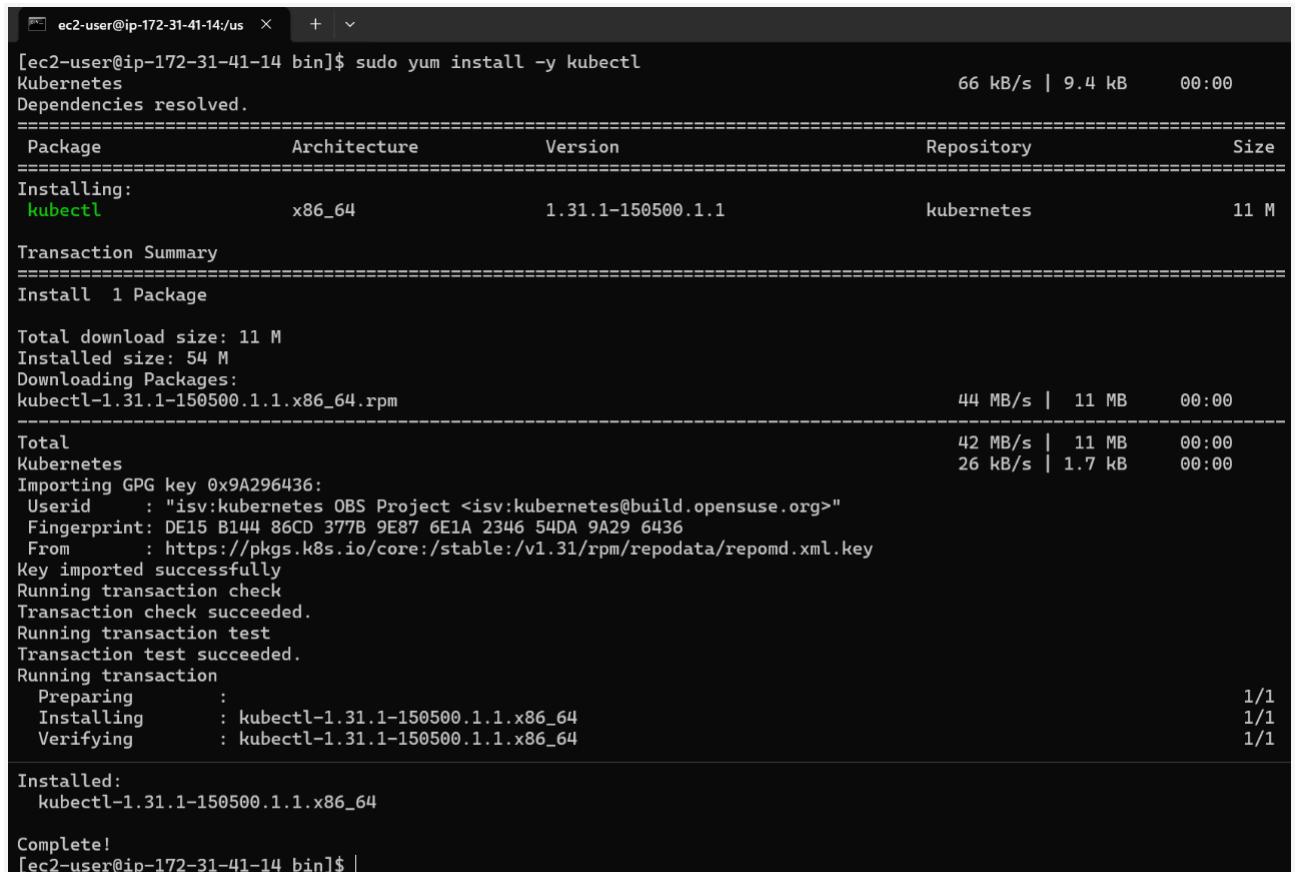
# Note: This dropin only works with kubeadm and kubelet v1.11+
[Service]
Environment="KUBELET_KUBECONFIG_ARGS=--bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kubelet.conf --kubeconfig=/etc/kubernetes/kubelet.conf"
Environment="KUBELET_CONFIG_ARGS=--config=/var/lib/kubelet/config.yaml"
# This is a file that "kubeadm init" and "kubeadm join" generates at runtime, populating the KUBELET_KUBEADM_ARGS variable dynamically
EnvironmentFile=/var/lib/kubelet/kubeadm-flags.env
# This is a file that the user can use for overrides of the kubelet args as a last resort. Preferably, the user should use
# the .NodeRegistration.KubeletExtraArgs object in the configuration files instead. KUBELET_EXTRA_ARGS should be sourced
# from this file.
EnvironmentFile=/etc/sysconfig/kubelet
ExecStart=
ExecStart=/usr/local/bin/kubelet $KUBELET_KUBECONFIG_ARGS $KUBELET_CONFIG_ARGS $KUBELET_KUBEADM_ARGS $KUBELET_EXTRA_ARGS
[ec2-user@ip-172-31-41-14 bin]$ |
```

## 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
```

```
[ec2-user@ip-172-31-41-14 bin]$ 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
[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
[ec2-user@ip-172-31-41-14 bin]$ |
sudo yum install -y kubectl
```



```
ec2-user@ip-172-31-41-14: ~ + ~
[ec2-user@ip-172-31-41-14 bin]$ sudo yum install -y kubectl
Kubernetes
Dependencies resolved.
=====
 Package           Architecture      Version       Repository      Size
=====
 Installing:
  kubectl          x86_64          1.31.1-150500.1.1   kubernetes    11 M
Transaction Summary
=====
 Install 1 Package

Total download size: 11 M
Installed size: 54 M
Downloading Packages:
kubectl-1.31.1-150500.1.1.x86_64.rpm
=====
Total
Kubernetes
Importing GPG key 0xA296436:
  Userid : "isv:kubernetes OBS Project <isv:kubernetes@build.opensuse.org>"
  Fingerprint: DE15 B144 86CD 377B 9E87 6E1A 2346 54DA 9A29 6436
  From   : https://pkgs.k8s.io/core:/stable:/v1.31/rpm/repo/repodata/repomd.xml.key
Key imported successfully
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing   :
  Installing  : kubectl-1.31.1-150500.1.1.x86_64
  Verifying   : kubectl-1.31.1-150500.1.1.x86_64
=====
Installed:
  kubectl-1.31.1-150500.1.1.x86_64
Complete!
[ec2-user@ip-172-31-41-14 bin]$ |
```

```
[ec2-user@ip-172-31-41-14 bin]$ 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.

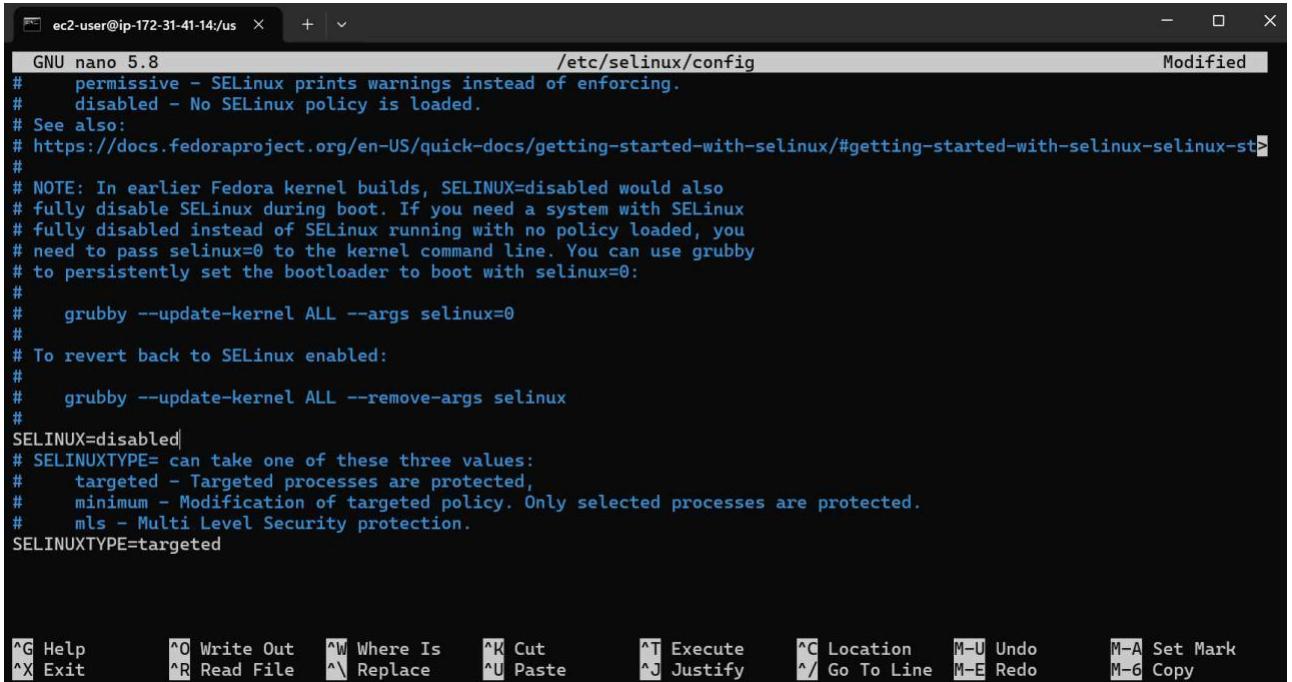
```
sudo swapoff -a
&& echo
"net.bridge.bridge-nf-call-iptables
s=1" | sudo tee
-a
/etc/sysctl.conf
&& sudo sysctl -p
```

```
[ec2-user@ip-172-31-41-14 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
[ec2-user@ip-172-31-41-14 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**



```
ec2-user@ip-172-31-41-14:~ % nano /etc/selinux/config
GNU nano 5.8                               /etc/selinux/config                         Modified
#           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-st
#
# 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

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

Then reboot the system using **sudo reboot**

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

```
[ec2-user@ip-172-31-41-14 bin]$ sudo nano /etc/selinux/config
[ec2-user@ip-172-31-41-14 bin]$ sudo reboot

Broadcast message from root@localhost on pts/1 (Wed 2024-09-18 16:15:38 UTC):
The system will reboot now!

[ec2-user@ip-172-31-41-14 bin]$ Connection to ec2-3-82-9-23.compute-1.amazonaws.com closed by remote host.
Connection to ec2-3-82-9-23.compute-1.amazonaws.com closed.

C:\Users\aditya\Downloads>ssh -i "aditya.pem" ec2-user@ec2-3-82-9-23.compute-1.amazonaws.com
      _#
     ~\_ #####_      Amazon Linux 2023
     ~~ \_\#####\_
     ~~   \###|
     ~~     \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023
     ~~       \~' _-->
     ~~~        /
     ~~~ ._. _/
     ~~~ /_/
     ~~~ /m/
Last login: Wed Sep 18 15:59:22 2024 from 202.179.85.199
[ec2-user@ip-172-31-41-14 ~]$ |
```

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

```
[ec2-user@ip-172-31-41-14 ~]$ sestatus
SELinux status:                      disabled
[ec2-user@ip-172-31-41-14 ~]$ |
```

## 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**

```
ec2-user@ip-172-31-41-14:~ + v
[ec2-user@ip-172-31-41-14 ~]$ sudo dnf install socat iproute-tc conntrack-tools -y
Last metadata expiration check: 0:06:41 ago on Wed Sep 18 16:10:55 2024.
Dependencies resolved.
=====
  Package           Architecture   Version      Repository    Size
=====
Installing:
  conntrack-tools      x86_64        1.4.6-2.amzn2023.0.2      amazonlinux  208 k
  iproute-tc          x86_64        5.10.0-2.amzn2023.0.5      amazonlinux  455 k
  socat              x86_64        1.7.4.2-1.amzn2023.0.2      amazonlinux  303 k
Installing dependencies:
  libnetfilter_cthelper x86_64        1.0.0-21.amzn2023.0.2      amazonlinux  24 k
  libnetfilter_cttimeout x86_64       1.0.0-19.amzn2023.0.2      amazonlinux  24 k
  libnetfilter_queue   x86_64        1.0.5-2.amzn2023.0.2      amazonlinux  30 k
=====
Transaction Summary
=====
Install 6 Packages

Total download size: 1.0 M
Installed size: 2.8 M
Downloading Packages:
(1/6): libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64.rpm      461 kB/s | 24 kB   00:00
(2/6): conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64.rpm            3.0 MB/s | 208 kB  00:00
(3/6): libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64.rpm     1.0 MB/s | 24 kB   00:00
(4/6): libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64.rpm         1.2 MB/s | 30 kB   00:00
(5/6): socat-1.7.4.2-1.amzn2023.0.2.x86_64.rpm                  5.9 MB/s | 303 kB  00:00
(6/6): iproute-tc-5.10.0-2.amzn2023.0.5.x86_64.rpm             3.1 MB/s | 455 kB  00:00
=====
Total   5.3 MB/s | 1.0 MB  00:00

Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing           :
  Installing         : libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64      1/1
  Installing         : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64    1/6
  Installing         : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64    2/6
  Installing         : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64        3/6
  Running scriptlet: conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64        4/6
  Installing         : socat-1.7.4.2-1.amzn2023.0.2.x86_64          5/6
  Installing         : iproute-tc-5.10.0-2.amzn2023.0.5.x86_64        6/6
  Running scriptlet: iproute-tc-5.10.0-2.amzn2023.0.5.x86_64        6/6
  Verifying          : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64        1/6
  Verifying          : iproute-tc-5.10.0-2.amzn2023.0.5.x86_64        2/6
  Verifying          : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64    3/6
  Verifying          : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64    4/6
  Verifying          : libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64        5/6
  Verifying          : socat-1.7.4.2-1.amzn2023.0.2.x86_64          6/6
=====
Installed:
  conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64                      iproute-tc-5.10.0-2.amzn2023.0.5.x86_64
  libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64                 libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64
  libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64                   socat-1.7.4.2-1.amzn2023.0.2.x86_64
=====
Complete!
```

- sudo kubeadm init --pod-network-cidr=10.244.0.0/16  
--ignore-preflight-errors=NumCPU,Mem

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

```
[ec2-user@ip-172-31-41-14 ~]$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=NumCPU,Mem
[kubelet] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
    [WARNING NumCPU]: the number of available CPUs 1 is less than the required 2
    [WARNING Mem]: the system RAM (949 MB) is less than the minimum 1700 MB
    [WARNING Service-Kubelet]: kubelet service is not enabled, please run 'systemctl enable kubelet.service'
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
W0918 16:19:10.818991    3251 checks.go:846] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent with that used by kubeadm. It is recommended to use "registry.k8s.io/pause:3.10" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-41-14.ec2.internal kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 172.31.41.14]
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key
[certs] Generating "etcd/ca" certificate and key
[certs] Generating "etcd/server" certificate and key
[certs] etcd/server serving cert is signed for DNS names [ip-172-31-41-14.ec2.internal localhost] and IPs [172.31.41.14 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-41-14.ec2.internal localhost] and IPs [172.31.41.14 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
[certs] Generating "sa" key and public key
```

- sudo systemctl restart kubelet

```
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 172.31.41.14:6443 --token sji44d.lmoyle2o2vxqpty1 \
    --discovery-token-ca-cert-hash sha256:e6ea5719242f99612e2a1f595c714c8a123691c05c912e18f6112e90cb67c035
[ec2-user@ip-172-31-41-14 ~]$ mkdir -p $HOME/.kube
[ec2-user@ip-172-31-41-14 ~]$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
[ec2-user@ip-172-31-41-14 ~]$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
[ec2-user@ip-172-31-41-14 ~]$ 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>**

```
[ec2-user@ip-172-31-41-14 ~]$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
[ec2-user@ip-172-31-41-14 ~]$ |
```

Now type **kubectl get nodes**

| NAME                         | STATUS | ROLES         | AGE   | VERSION |
|------------------------------|--------|---------------|-------|---------|
| ip-172-31-41-14.ec2.internal | Ready  | control-plane | 3m13s | v1.31.1 |

**Note: If any time of get error of connection refused just restart the kubelet service (sudo systemctl restart kubelet)**

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

Apply this deployment file using this command to create a deployment

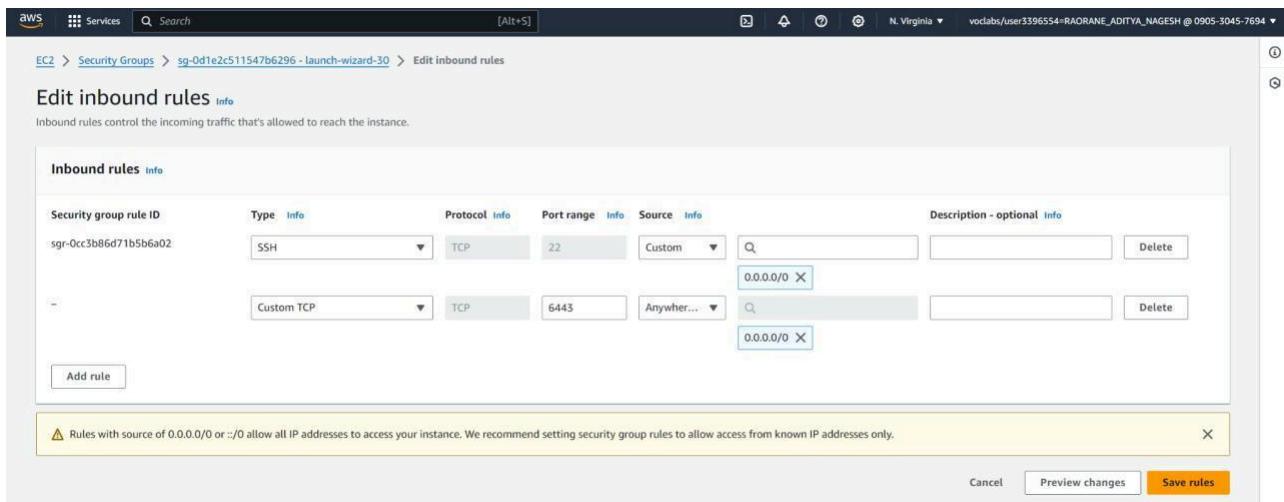
- **kubectl apply -f https://k8s.io/examples/application/deployment.yaml**

```
[ec2-user@ip-172-31-41-14 ~]$ kubectl apply -f https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
[ec2-user@ip-172-31-41-14 ~]$ |
```

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

```
[ec2-user@ip-172-31-41-14 ~]$ kubectl get pods
The connection to the server 172.31.41.14:6443 was refused - did you specify the right host or port?
[ec2-user@ip-172-31-41-14 ~]$ |
```

Add an inbound rule under SSH security groups which will allow the traffic for a **custom TCP port** with port number **6443** with setting the source as **anywhere from IPv4**.



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-d556bf558-7zthh**)

To make pods untainted

Type **kubectl get nodes** to see the name of the node.

Then type command **kubectl taint nodes <NODE\_NAME> - -all**

In my case it is as follows:

**kubectl taint nodes ip-172-31-41-14.ec2.internal node-role.kubernetes.io/control-plane-**

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

```
[ec2-user@ip-172-31-41-14 ~]$ kubectl taint nodes ip-172-31-41-14.ec2.internal node-role.kubernetes.io/control-plane-
node/ip-172-31-41-14.ec2.internal untainted
```

```
[ec2-user@ip-172-31-41-14 ~]$ kubectl get pods
NAME                      READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-7zthh   1/1     Running   2 (27s ago)   60s
nginx-deployment-d556bf558-vdldn   1/1     Running   2 (27s ago)   60s
[ec2-user@ip-172-31-41-14 ~]$ |
```

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-d556bf558-7zthh 8080:80**

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

```
[ec2-user@ip-172-31-41-14 ~]$ kubectl port-forward nginx-deployment-d556bf558-7zthh 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 And type command **curl --head <http://127.0.0.1:8080>**

```
[ec2-user@ip-172-31-41-14 ~]$ curl --head http://127.0.0.1:8080
HTTP/1.1 200 OK
Server: nginx/1.14.2
Date: Wed, 18 Sep 2024 16:33:35 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-172-31-41-14 ~]$ |
```

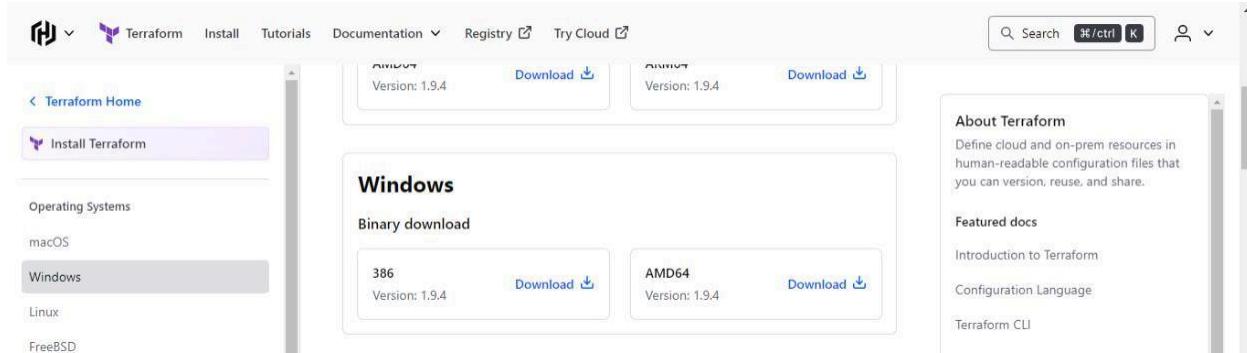
Response status 200 (OK) indicates that our nginx server is running successfully on kubernetes

**Conclusion:** We began by installing and configuring Docker and Kubernetes, encountering some initial issues with the Kubernetes API server, which were resolved by restarting the `kubelet` service. The pods didn't start at first due to taints on the nodes, which we removed to allow normal pod scheduling. After resolving these errors, we successfully deployed Nginx server pods and configured them to be accessible via port forwarding. Additionally, we configured the SSH security group by adding an inbound rule to permit traffic on TCP port 6443 from any IPv4 address.

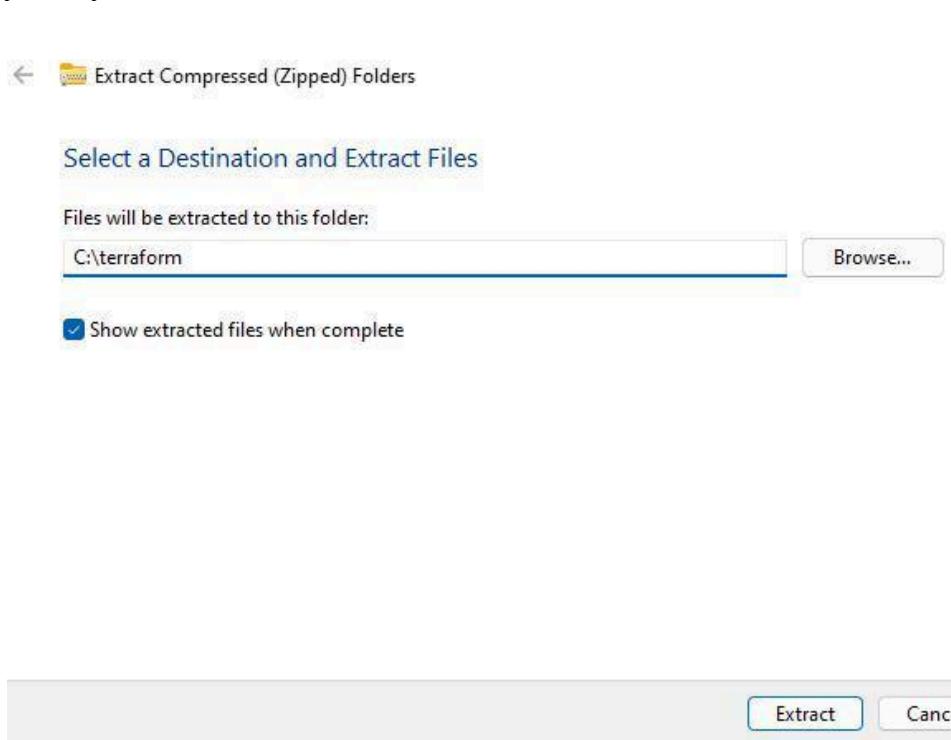
## Experiment 05: Installation of Terraform

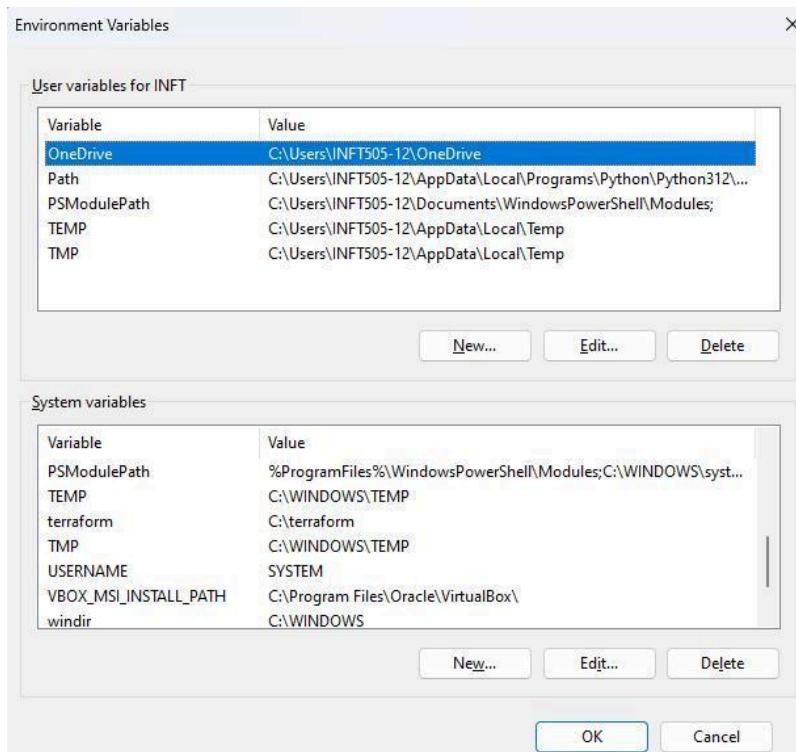
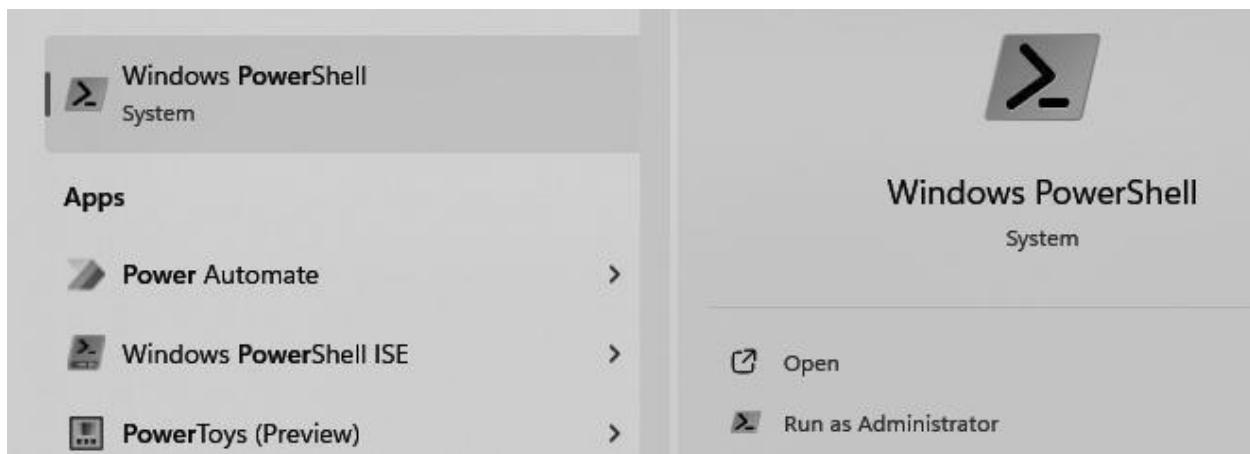
**Step 1:** To install Terraform, visit the official Terraform website mentioned below, go to the Downloads section, select Windows, and download the 64-bit version for your system.

website: <https://www.terraform.io/downloads.html>

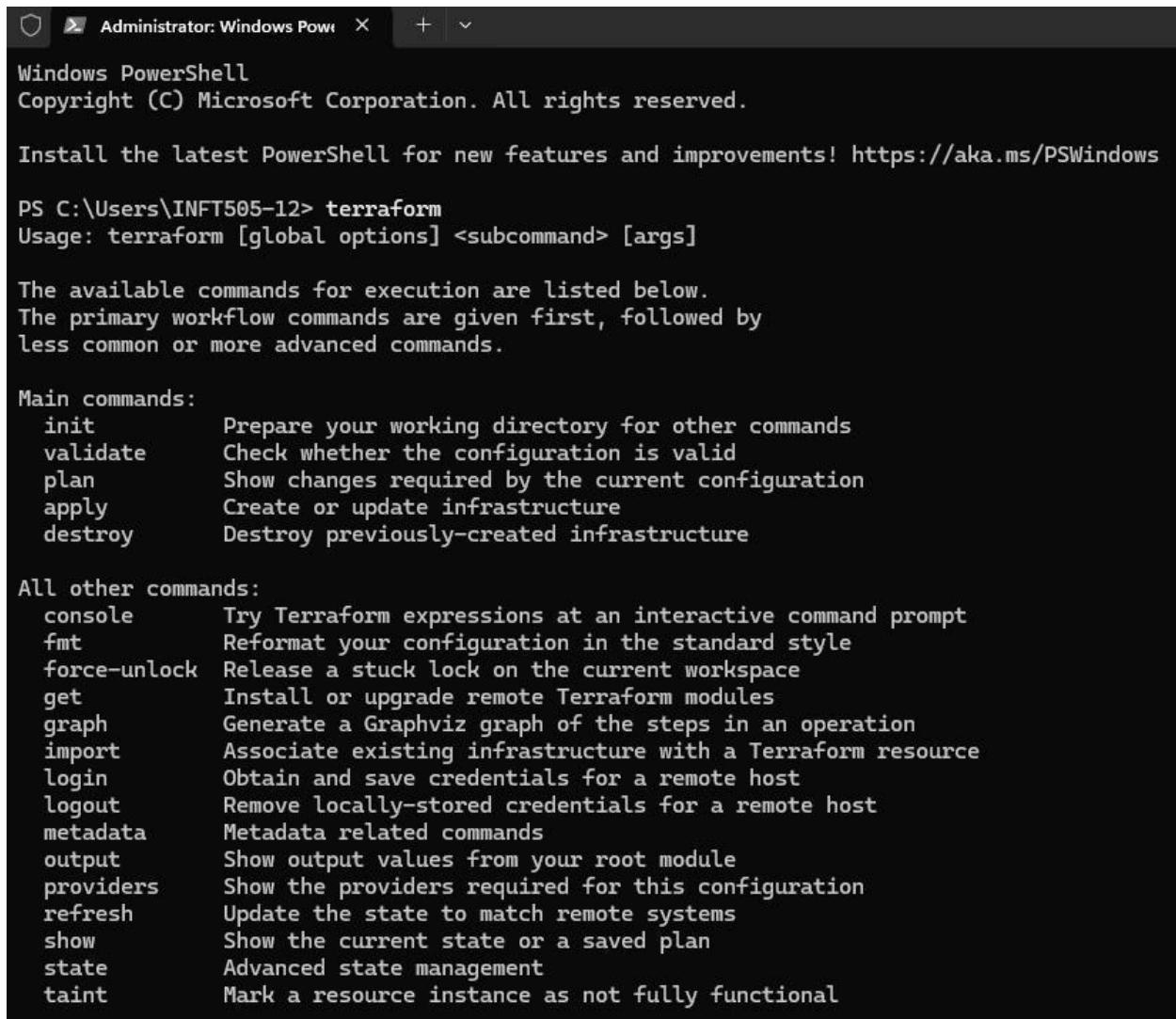


**Step 2:** Extract the downloaded 'Terraform.exe' file to the 'C:\Terraform' directory on your system.



**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.



A screenshot of a Windows PowerShell window titled "Administrator: Windows Powe". The window shows the Terraform help output. It starts with the PowerShell welcome message, followed by the command "PS C:\Users\INFT505-12> terraform", the usage information "Usage: terraform [global options] <subcommand> [args]", and a list of available commands categorized into "Main commands:" and "All other commands:". The "Main commands:" section includes init, validate, plan, apply, and destroy. The "All other commands:" section includes console, fmt, force-unlock, get, graph, import, login, logout, metadata, output, providers, refresh, show, state, and taint.

```
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\INFT505-12> 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 06 :Creating docker image using terraform

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

```
Microsoft Windows [Version 10.0.22621.3880]
(c) Microsoft Corporation. All rights reserved.

C:\Users\adity>docker

Usage: docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

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

Management Commands:
  builder   Manage builds
  buildx*   Docker Buildx
  compose*  Docker Compose
  container Manage containers
  context   Manage contexts
  debug*    Get a shell into any image or container
  desktop*  Docker Desktop commands (Alpha)
  dev*      Docker Dev Environments
  extension* Manages Docker extensions
  feedback* Provide feedback, right in your terminal!
  image     Manage images
  init*    Creates Docker-related starter files for your project
  manifest  Manage Docker image manifests and manifest lists
  network   Manage networks
  plugin    Manage plugins
  sbom*    View the packaged-based Software Bill Of Materials (SBOM) for an image
  scout*   Docker Scout
  system    Manage Docker
  trust     Manage trust on Docker images
  volume   Manage volumes
```

```
Microsoft Windows [Version 10.0.22621.3880]
(c) Microsoft Corporation. All rights reserved.
```

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

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

```
C:\Users\adity\Desktop\Project 1>mkdir TerraformScripts
```

```
C:\Users\adity\Desktop\Project 1>cd TerraformScripts
```

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

```
C:\Users\adity\Desktop\Project 1\TerraformScripts>mkdir Docker
```

```
C:\Users\adity\Desktop\Project 1\TerraformScripts>cd Docker
```

```
C:\Users\adity\Desktop\Project 1\TerraformScripts\ Docker>notepad docker.tf
```

Script:

```
terraform {
  required_providers {
    docker = {
      source = "kreuzwerker/docker"
      version = "2.21.0"
    }
  }
}

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

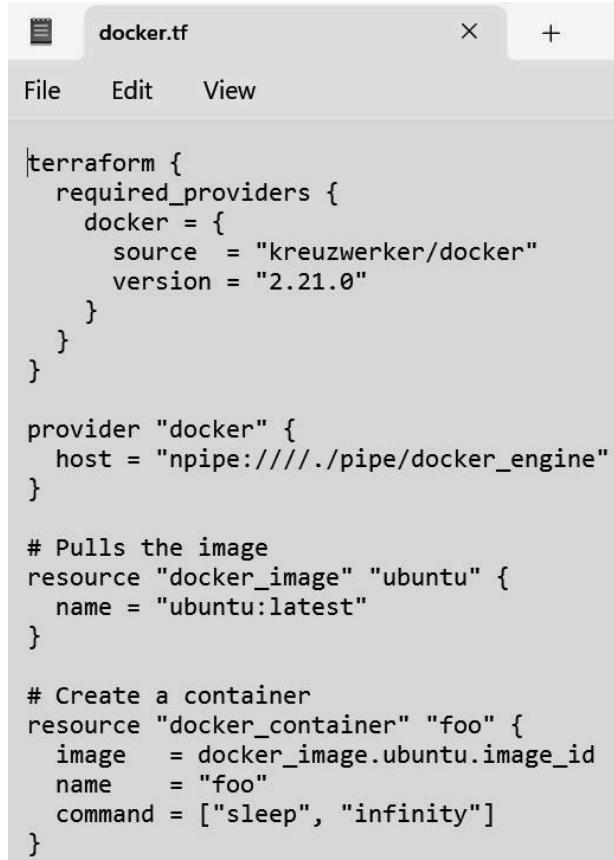
# Pulls the image
resource "docker_image" "ubuntu" {
  name = "ubuntu:latest"
}

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

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

It pulls the latest Ubuntu image from Docker Hub and creates a container named "foo." The container runs the `sleep infinity` command, which keeps it active indefinitely.

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



```

terraform {
  required_providers {
    docker = {
      source  = "kreuzwerker/docker"
      version = "2.21.0"
    }
  }
}

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

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

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

```

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

```

C:\Users\adity\Desktop\Project 1\TerraformScripts\Docker>terraform init
Initializing the backend...
Initializing provider plugins...
- Finding kreuzwerker/docker versions matching "2.21.0"...
- Installing kreuzwerker/docker v2.21.0...
- Installed kreuzwerker/docker v2.21.0 (self-signed, key ID BD080C4571C6104C)
  Partner and community providers are signed by their developers.
  If you'd like to know more about provider signing, you can read about it here:
  https://www.terraform.io/docs/cli/plugins/signing.html
  Terraform has created a lock file .terraform.lock.hcl to record the provider
  selections it made above. Include this file in your version control repository
  so that Terraform can guarantee to make the same selections by default when
  you run "terraform init" in the future.

```

Terraform has been successfully initialized!

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

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

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

```
C:\Users\aditya\Desktop\Project 1\TerraformScripts\Docker>terraform plan
```

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

```
+ create
```

```
Terraform will perform the following actions:
```

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

    + healthcheck (known after apply)

    + labels (known after apply)
}
```

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

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

---

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

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

```
C:\Users\adity\Desktop\Project 1\TerraformScripts\Docker>terraform apply

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

Terraform will perform the following actions:

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

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

Enter a value: yes

docker_image.ubuntu: Creating...
docker_image.ubuntu: Creation complete after 8s [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Creating...
```

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

Docker images, Before Executing Apply step:

```
C:\Users\adity\Desktop\Project 1\TerraformScripts\Docker>docker images
REPOSITORY      TAG          IMAGE ID      CREATED       SIZE
```

Docker images, After Executing Apply step:

```
C:\Users\adity\Desktop\Project 1\TerraformScripts\Docker>docker images
REPOSITORY      TAG          IMAGE ID      CREATED       SIZE
ubuntu          latest        edbfe74c41f8   2 weeks ago  78.1MB
```

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

```
C:\Users\adity\Desktop\Project 1\TerraformScripts\Docker>terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]

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

Terraform will perform the following actions:

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

Plan: 0 to add, 0 to change, 1 to destroy.

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

Enter a value: yes

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

Destroy complete! Resources: 1 destroyed.
```

Step 9: Docker images After Executing Destroy step

```
C:\Users\adity\Desktop\Project 1\TerraformScripts\Docker>docker images
REPOSITORY      TAG          IMAGE ID      CREATED       SIZE
```

## Experiment 07

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

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

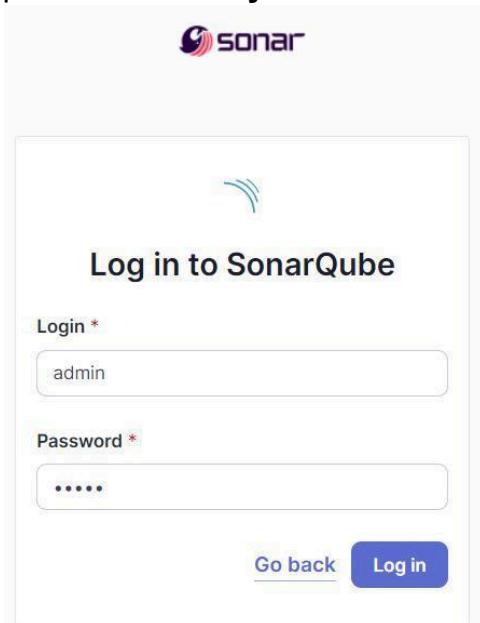
| S | W | Name           | Last Success                    | Last Failure                   | Last Duration |
|---|---|----------------|---------------------------------|--------------------------------|---------------|
|   |   | My_First_Maven | 23 days <a href="#">#2</a>      | 23 days <a href="#">#1</a>     | 20 sec        |
|   |   | MyPipeline_1   | 28 days <a href="#">#1</a>      | N/A                            | 9.2 sec       |
|   |   | Pipeline_01    | 1 mo 15 days <a href="#">#3</a> | N/A                            | 9.9 sec       |
|   |   | WebTestDriver  | 1 day 16 hr <a href="#">#5</a>  | 1 day 16 hr <a href="#">#4</a> | 13 sec        |

2. Run SonarQube in a Docker container using this command :  
 a] docker -v  
 b] docker pull sonarqube  
 c] docker run -d --name sonarqube -e SONAR\_ES\_BOOTSTRAP\_CHECKS\_DISABLE=true -p 9000:9000 sonarqube:latest

```
C:\Users\adity>docker -v
Docker version 27.0.3, build 7d4bcd8

C:\Users\adity>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
4a6e73f4472de892b1ddead1abe77372a85a7b09408cce3a0abd37c5ab6b49a4
```

3. Once the container is up and running, you can check the status of SonarQube at **localhost port 9000**. The login id is “**admin**” and the password is “**aditya**”.



4. Create a local project in SonarQube with the name **sonarqube**

1 of 2

Create a local project

Project display name \*

Project key \*

Main branch name \*

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

[Cancel](#)

[Next](#)

2 of 2

### Set up project for Clean as You Code

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

Choose the baseline for new code for this project

Use the global setting

Previous version  
Any code that has changed since the previous version is considered new code.  
Recommended for projects following regular versions or releases.

Define a specific setting for this project

Previous version  
Any code that has changed since the previous version is considered new code.  
Recommended for projects following regular versions or releases.

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

Reference branch  
Choose a branch as the baseline for the new code.  
Recommended for projects using feature branches.

[Back](#) [Create project](#)

5. Setup the project and come back to Jenkins Dashboard. Go to **Manage Jenkins → Plugins** and search for **SonarQube Scanner** in **Available Plugins** and install it.



6. Under '**Manage Jenkins → System**', look for **SonarQube Servers** and enter these details.

Name : sonarqube

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

The screenshot shows the Jenkins System configuration page under 'SonarQube servers'. It has sections for 'Environment variables' (checked), 'SonarQube installations' (list of installations), 'Name' (input field containing 'sonarqube'), 'Server URL' (input field containing 'http://localhost:9000'), 'Server authentication token' (input field containing '- none -'), and an 'Advanced' dropdown. At the bottom are 'Save' and 'Apply' buttons.

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

**Manage Jenkins → Tools → SonarQube Scanner Installation**

The screenshot shows the Jenkins 'Tools' configuration page. Under 'SonarQube Scanner installations', there is a configuration named 'sonarqube'. The 'Install automatically' checkbox is checked. The 'Install from Maven Central' section is expanded, showing the version 'SonarQube Scanner 6.2.0.4584'. There is also a 'Save' button at the bottom.

8. After the configuration, create a **New Item** in Jenkins, choose a

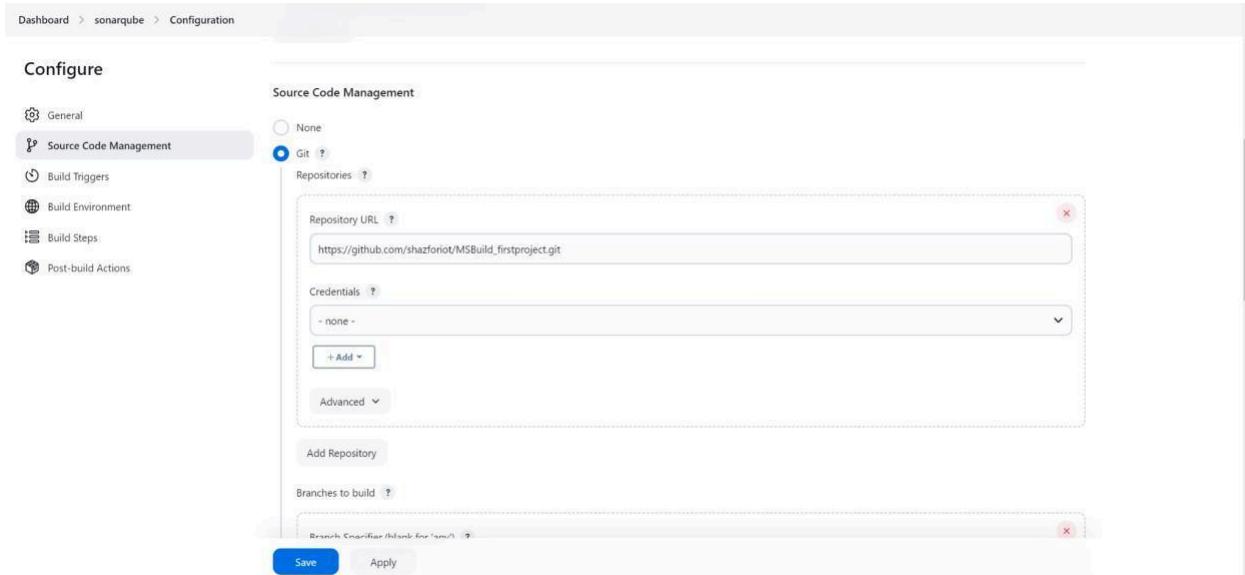
**freestyle project named sonarqube.**

The screenshot shows the Jenkins 'New Item' dialog. The item name 'sonarqube' is entered. The 'Freestyle project' option is selected, with a tooltip explaining it's a classic, general-purpose job type. Other options like 'Maven project', 'Pipeline', 'Multi-configuration project', and 'Folder' are listed below. At the bottom, there is an 'OK' button.

9. Choose this GitHub repository in **Source Code**

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

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



10. Under **Build-> Execute SonarQube Scanner**, enter these **Analysis Properties**.

Mention the SonarQube Project Key, Login, Password, Source path and Host URL.

sonar.projectKey=sonarqube

sonar.login=admin

sonar.password=aditya

sonar.sources=.

sonar.host.url=http://localhost:9000

The screenshot shows the SonarQube Configuration page with the 'Build Steps' tab selected. A modal window titled 'Execute SonarQube Scanner' is open, displaying configuration options for the scanner. The 'JDK' dropdown is set to '(Inherit From Job)'. The 'Path to project properties' field contains the path to project properties. The 'Analysis properties' section contains the following configuration:

```

sonar.projectKey=sonarqube
sonar.login=admin
sonar.host.url=http://localhost:9000
sonar.sources=.

```

The 'Additional arguments' and 'JVM Options' fields are empty. At the bottom of the modal, there are 'Save' and 'Apply' buttons.

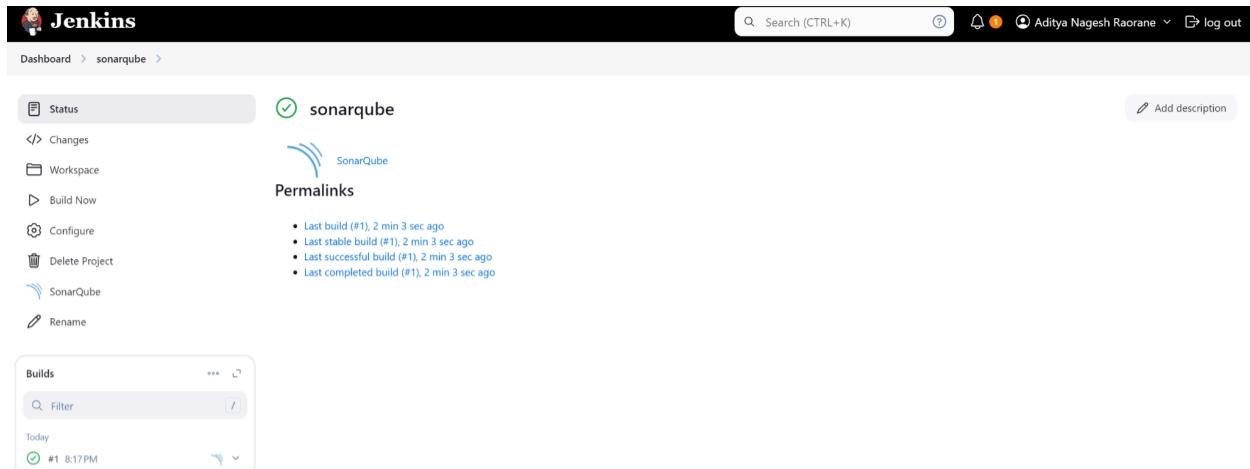
11. Go to <http://localhost:9000/admin/permissions> and allow Execute Permissions to the Admin user.

The screenshot shows the SonarQube Administration interface with the 'Global Permissions' section selected. The 'Security' tab is active. The page lists global permissions for groups: 'sonar-administrators', 'sonar-users', and 'Anyone'. The 'sonar-administrators' group has checked boxes for 'Administer System', 'Administer Quality Gates', and 'Create Projects'. The 'sonar-users' group has checked boxes for 'Administer Quality Profiles' and 'Create Projects'. The 'Anyone' group has checked boxes for 'Administer Quality Profiles' and 'Create Projects'. There is a note about the 'Anyone' group being deprecated.

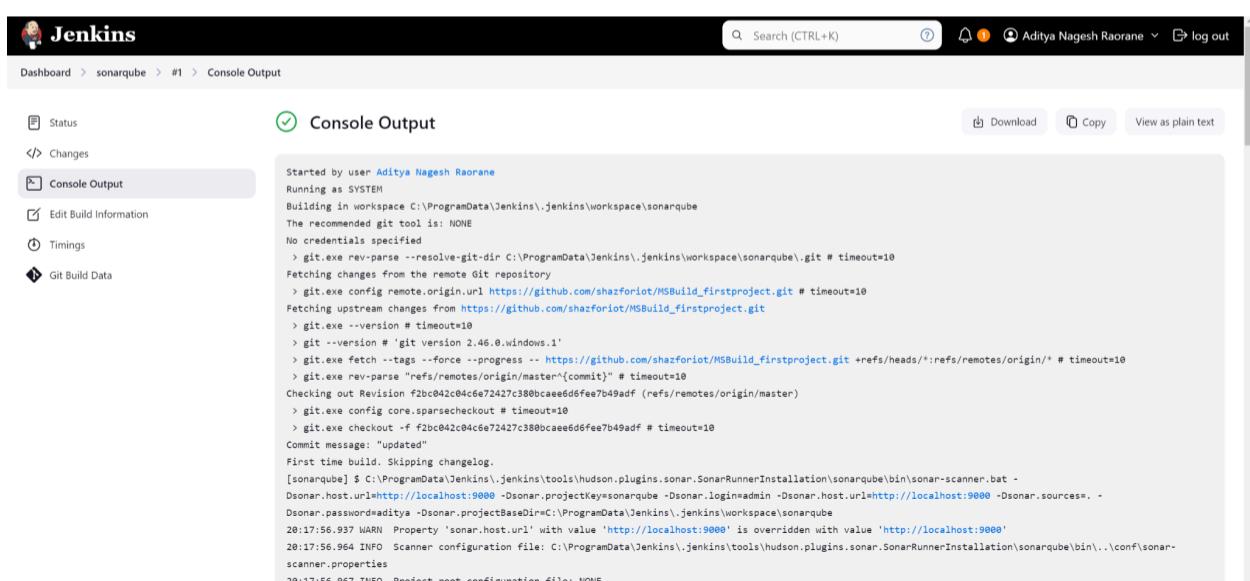
|                      | Administer System                   | Administer Quality Gates                                        | Execute Analysis                    | Create Projects                     |
|----------------------|-------------------------------------|-----------------------------------------------------------------|-------------------------------------|-------------------------------------|
| sonar-administrators | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/>                             | <input type="checkbox"/>            | <input checked="" type="checkbox"/> |
| sonar-users          | <input type="checkbox"/>            | <input type="checkbox"/><br><input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Anyone DEPRECATED    | <input type="checkbox"/>            | <input type="checkbox"/><br><input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/>            |
| Administrator admin  | <input checked="" type="checkbox"/> | <input type="checkbox"/><br><input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

4 of 4 shown

## 12. Run The Build and check the console output.



The screenshot shows the Jenkins dashboard for the 'sonarqube' project. The status is green with a checkmark icon. The sidebar includes options like Status, Changes, Workspace, Build Now, Configure, Delete Project, SonarQube, and Rename. A 'Builds' section shows a single build (#1) from 8:17PM today. The main content area displays the SonarQube logo and permalinks.

The screenshot shows the Jenkins console output for build #1. The status is green with a checkmark icon. The sidebar includes options like Status, Changes, Console Output, Edit Build Information, Timings, and Git Build Data. The main content area shows the command-line output of the build process, which includes cloning a GitHub repository and running a SonarScanner batch file. The output ends with a success message: "Project root configuration file: NONE".



The screenshot shows the Jenkins console output for a SonarQube analysis. The log output is as follows:

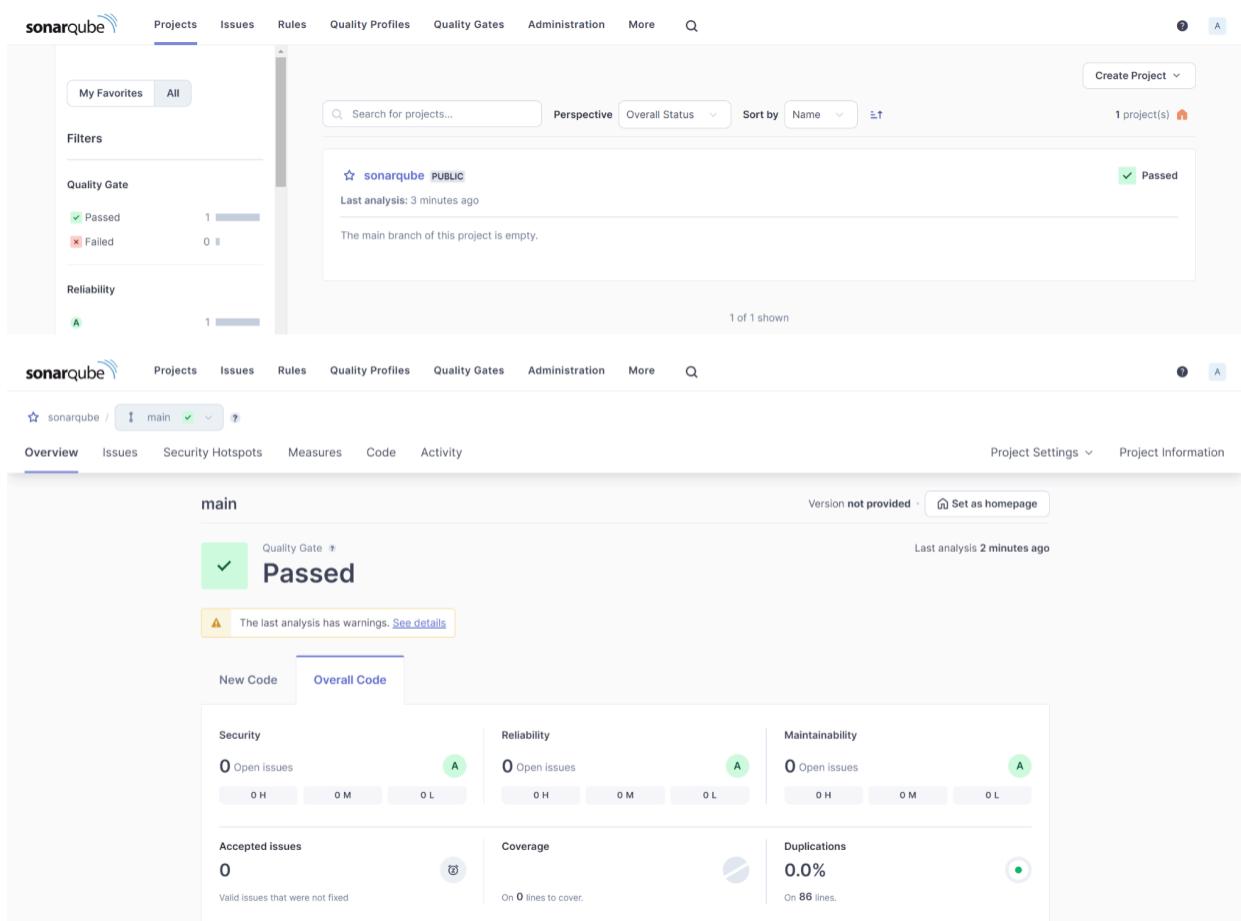
```

Dashboard > sonarqube > #1 > Console Output
20:18:52.472 WARN Your project contains C# files which cannot be analyzed with the scanner you are using. To analyze C# or VB.NET, you must use the SonarScanner for .NET 5.x or higher, see https://redirect.sonarsource.com/doc/install-configure-scanner-msbuild.html
20:18:52.473 INFO Sensor C# [csharp] (done) | time=2ms
20:18:52.474 INFO Sensor Analysis Warnings import [csharp]
20:18:52.478 INFO Sensor Analysis Warnings import [csharp] (done) | time=4ms
20:18:52.479 INFO Sensor C# File Caching Sensor [csharp]
20:18:52.482 WARN Incremental PR analysis: Could not determine common base path, cache will not be computed. Consider setting 'sonar.projectBaseDir' property.
20:18:52.482 INFO Sensor C# File Caching Sensor [csharp] (done) | time=4ms
20:18:52.483 INFO Sensor Zero Coverage Sensor
20:18:52.510 INFO Sensor Zero Coverage Sensor (done) | time=28ms
20:18:52.515 INFO SCM Publisher SCM provider for this project is: git
20:18:52.518 INFO SCM Publisher 4 source files to be analyzed
20:18:53.806 INFO SCM Publisher 4/4 source files have been analyzed (done) | time=1286ms
20:18:53.810 INFO CPD Executor Calculating CPD for 0 files
20:18:53.811 INFO CPD Executor CPD calculation finished (done) | time=0ms
20:18:53.822 INFO SCM revision ID 'f20cb42c84c6e72427c3800cae6d6fe7b549ad'
20:18:54.975 INFO Analysis report generated in 240ms, dir size=201.0 kB
20:18:55.237 INFO Analysis report compressed in 114ms, zip size=22.4 kB
20:18:55.614 INFO Analysis report uploaded in 374ms
20:18:55.618 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube
20:18:55.621 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
20:18:55.622 INFO More about the report processing at http://localhost:9000/api/ce/task?id=a2e28c04-ce64-4689-8023-5b03ea519fc9
20:18:55.653 INFO Analysis total time: 39.158 s
20:18:55.658 INFO SonarScanner Engine completed successfully
20:18:55.741 INFO EXECUTION SUCCESS
20:18:55.743 INFO Total time: 58.785s
Finished: SUCCESS

```

REST API Jenkins 2.473

### 13. Once the build is complete, check the project in SonarQube.



The screenshots show the SonarQube interface. The top screenshot displays the main project page for 'sonarqube PUBLIC'. It shows a single project with a status of 'Passed'. The bottom screenshot shows the detailed analysis for the 'main' branch. The analysis is marked as 'Passed' with a green checkmark. Key metrics shown include 0 open issues, 0 accepted issues, 0.0% duplication, and 100% coverage.

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

### **Conclusion:**

In this experiment, we have understood the importance of SAST and have successfully integrated Jenkins with SonarQube for Static Analysis and Code Testing.

## Experiment 08

**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.

1. Open up Jenkins Dashboard on localhost:8080.

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

- Header:** Jenkins logo, search bar (Search (CTRL+K)), notifications, user info (Aditya Nagesh Raorane), and log out button.
- Left Sidebar:**
  - New Item
  - Build History
  - Project Relationship
  - Check File Fingerprint
  - Manage Jenkins
  - My Views
- Build Queue:** A dropdown menu showing "No builds in the queue."
- Build Executor Status:** A dropdown menu showing "0/2".
- Main Content:** A table listing Jenkins projects:
 

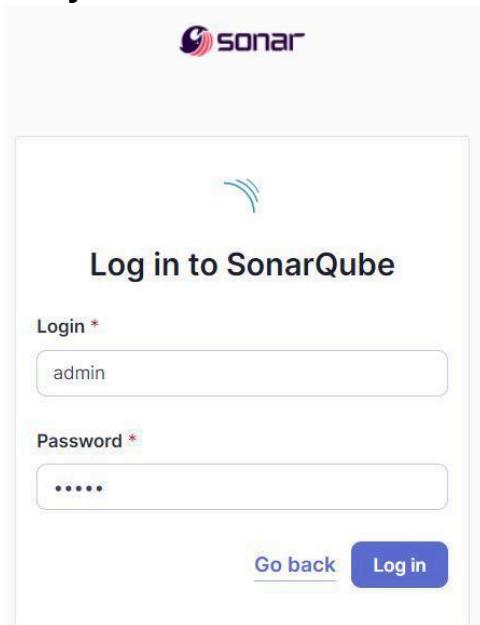
| S | W | Name           | Last Success    | Last Failure   | Last Duration |
|---|---|----------------|-----------------|----------------|---------------|
|   |   | My_First_Maven | 23 days #2      | 23 days #1     | 20 sec        |
|   |   | MyPipeline1    | 28 days #1      | N/A            | 9.2 sec       |
|   |   | Pipeline_01    | 1 mo 15 days #3 | N/A            | 9.9 sec       |
|   |   | sonarqube      | 43 min #1       | N/A            | 1 min 2 sec   |
|   |   | WebTestDriver  | 1 day 18 hr #5  | 1 day 18 hr #4 | 13 sec        |
- Bottom:** Icon selection buttons (S, M, L) and a page number (006).

2. Run SonarQube in a Docker container using this command:
  - a] docker -v
  - b] docker pull sonarqube
  - c] docker run -d --name sonarqube -e SONAR\_ES\_BOOTSTRAP\_CHECKS\_DISABLE=true -p 9000:9000 sonarqube:latest

```
C:\Users\adity>docker -v
Docker version 27.0.3, build 7d4bcd8

C:\Users\adity>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
4a6e73f4472de892b1ddead1abe77372a85a7b09408cce3a0abd37c5ab6b49a4
```

3. Once the container is up and running, you can check the status of SonarQube at **localhost port 9000**. The login id is “**admin**” and the password is “**aditya**”.



#### 4. Create a local project in SonarQube with the name **sonarqube-test**.

1 of 2

#### Create a local project

Project display name \*

 •

Project key \*

 •

Main branch name \*

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

[Cancel](#) [Next](#)

2 of 2

#### Set up project for Clean as You Code

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

Choose the baseline for new code for this project

Use the global setting

Previous version  
Any code that has changed since the previous version is considered new code.  
Recommended for projects following regular versions or releases.

Define a specific setting for this project

Previous version  
Any code that has changed since the previous version is considered new code.  
Recommended for projects following regular versions or releases.

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

Reference branch  
Choose a branch as the baseline for the new code.  
Recommended for projects using feature branches.

[Back](#) [Create project](#)

Setup the project and come back to Jenkins Dashboard.

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

New Item

Enter an item name  
sonarqube-test

Select an item type

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

OK

## 7. Under **Pipeline Script**, enter the following -

```

node {
    stage('Cloning the GitHub Repo') {
        git 'https://github.com/shazforiot/GOL.git'
    }
    stage('SonarQube analysis') {
        withSonarQubeEnv('sonarqube') {
            bat
                "C:\\\\Users\\\\adity\\\\Downloads\\\\sonar-scanner-cli-6.1.0.4477-windows-x64\\\\sonar-scanner-6.1.0.4477-windows-x64\\\\bin\\\\sonar-scanner.bat \\
                    -D sonar.login=<YOUR ID> \\
                    -D sonar.password=<YOUR PASSWORD> \\
                    -D sonar.projectKey=<YOUR PROJECT KEY> \\
                    -D sonar.exclusions=vendor/**,resources/**, **/*.java \\
                    -D sonar.host.url=http://localhost:9000/ "
        }
    }
}

```

The screenshot shows the Jenkins Pipeline configuration page for a project named 'sonarqube-test'. The pipeline script is defined as follows:

```
1 * node {
2   stage('Cloning the GitHub Repo') {
3     git 'https://github.com/shafiqrao/GOI.git'
4   }
5   stage('SonarQube analysis') {
6     withSonarQubeEnv('sonarqube') {
7       bat "C:\\Users\\Aditya\\Downloads\\sonar-scanner-cli-0.1.0.4477-windows-x64\\sonar-scanner-0.1.0.4477-windows-x64\\bin\\sonar-scanner.bat"
8       -D sonar.login=admin
9       -D sonar.password=aditya
10      -D sonar.projectKey=sonarqube-test
11      -D sonar.exclusions.vendor/**,resources/**,*/*.java
12      -D sonar.host.url=http://localhost:9000/
13    }
14  }
15 }
```

Below the script, there is a checkbox labeled 'Use Groovy Sandbox' which is checked. At the bottom of the page 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.

The screenshot shows the Jenkins dashboard for the 'sonarqube-test' project. The status is 'Status' (green), and there is a 'Build Now' button.

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

The screenshot shows the Jenkins Pipeline interface for a project named "sonarqube-test". On the left, there's a sidebar with various pipeline management options like Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, SonarQube, Stages, Rename, and Pipeline Syntax. The main area is titled "Stage View" and displays a summary of the build process. It includes a timeline bar with two segments: "Cloning the GitHub Repo" (1s) and "SonarQube analysis" (14min 37s). Below the timeline, a card for build #1 shows it was run on Sep 17 at 21:26 with "No Changes". A note indicates an average stage time of 1s and a full run time of ~14min 39s. To the right, a section titled "Permalinks" lists four recent builds: Last build (#1), Last stable build (#1), Last successful build (#1), and Last completed build (#1), all from 15 minutes ago. At the bottom left, a "Builds" section shows a single build entry for "#1 9:26PM".

The screenshot shows the Jenkins Pipeline interface for the same project. The sidebar on the left includes options like Status, Changes, Console Output (which is selected), View as plain text, Edit Build Information, Timings, Git Build Data, Pipeline Overview, Pipeline Console, Thread Dump, Pause/resume, Replay, Pipeline Steps, and Workspaces. The main area is titled "Console Output" and displays the log of the build process. The log starts by showing the pipeline being started by user Aditya Nagesh Raorane, then cloning the GitHub repository, and finally fetching upstream changes from a specific GitHub URL. The log concludes with a commit message: "Update Jenkinsfile".

```

Dashboard > sonarqube-test > #1
line 41. Keep only the first 100 references.
21:37:59.929 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at
line 17. Keep only the first 100 references.
21:37:59.929 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.
21:37:59.929 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at
line 75. Keep only the first 100 references.
21:37:59.938 INFO CPD Executor CPD calculation finished (done) | time=153336ms
21:37:59.955 INFO SCM revision ID 'ba799ba7e1b576f04a461232b0412c5ed6e1e5e4'
21:40:14.276 INFO Analysis report generated in 5151ms, dir size=127.2 MB
21:40:35.678 INFO Analysis report compressed in 21388ms, zip size=29.6 MB
21:40:36.170 INFO Analysis report uploaded in 492ms
21:40:36.173 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube-test
21:40:36.173 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
21:40:36.173 INFO More about the report processing at http://localhost:9000/api/ce/task?id=99fcde5-df4e-4f9f-8688-3169741e0856
21:40:53.466 INFO Analysis total time: 14:32.336 s
21:40:53.468 INFO SonarScanner Engine completed successfully
21:40:54.148 INFO EXECUTION SUCCESS
21:40:54.150 INFO Total time: 14:35.645s
[Pipeline]
[Pipeline] // withSonarQubeEnv
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

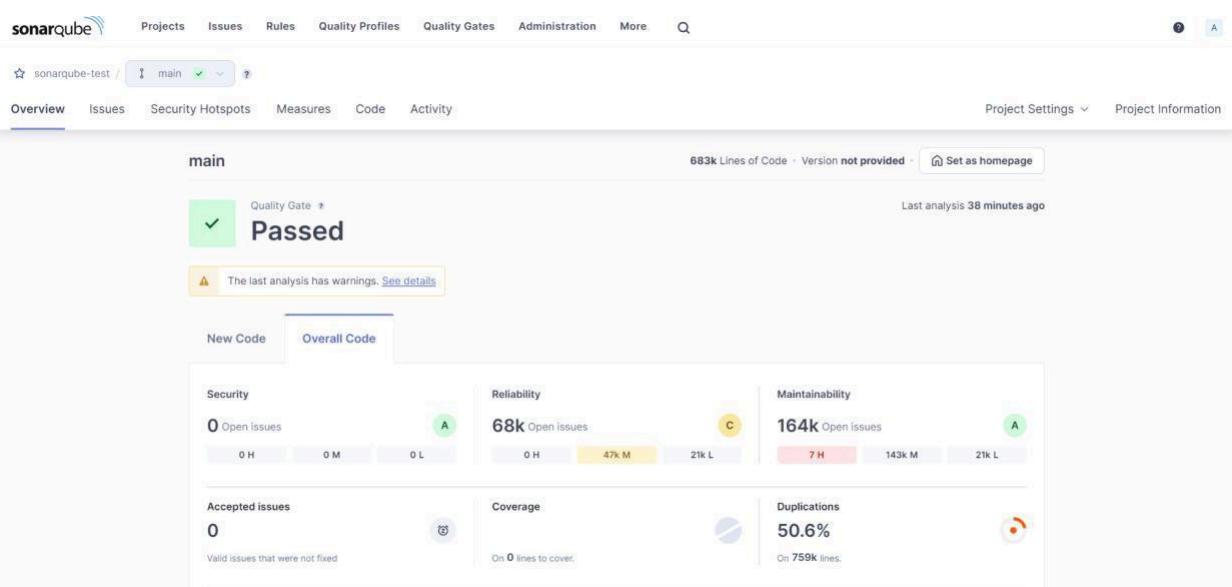
REST API Jenkins 2.473

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

The screenshot shows the SonarQube web interface. 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, there's a sidebar with filters for Quality Gate (Passed: 2, Failed: 0), Reliability (A: 1, B: 0, C: 1, D: 0, E: 0), and Security (A: 2). The main area displays two projects:

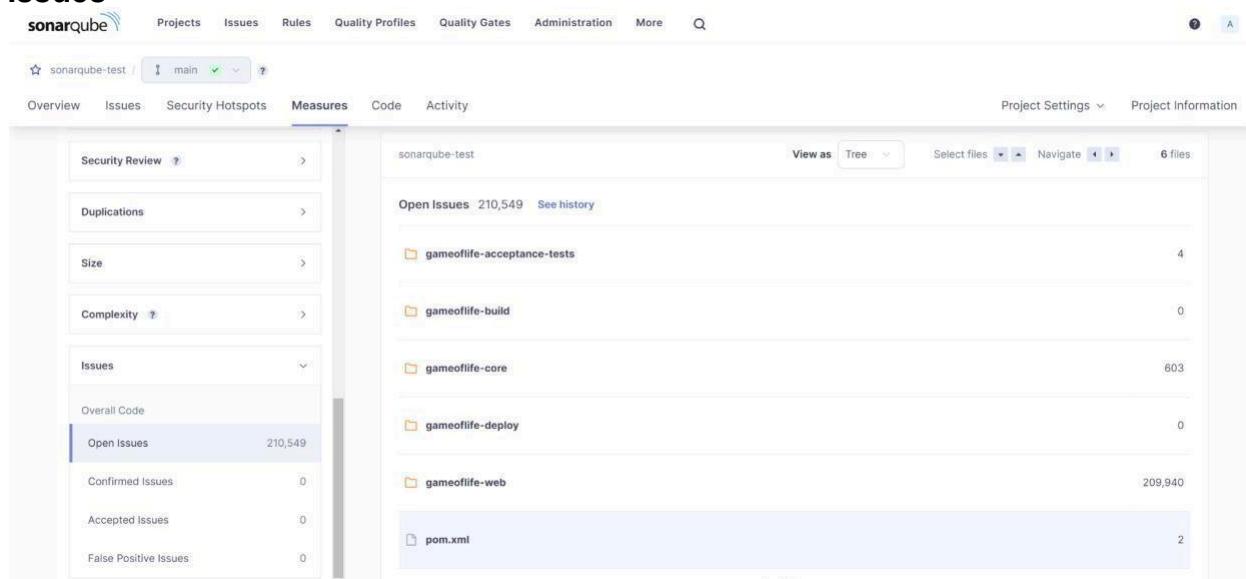
- sonarqube PUBLIC**: Last analysis: 1 hour ago. Status: Passed. Message: The main branch of this project is empty.
- sonarqube-test PUBLIC**: Last analysis: 16 minutes ago - 683k Lines of Code - HTML, XML, ... Status: Passed. Metrics: Security (A: 0), Reliability (C: 68k), Maintainability (A: 164k), Hotspots Reviewed (E: 0.0%), Coverage (—), Duplications (50.6%).

At the bottom right, it says "2 of 2 shown".



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

## 11. Code Problems - Open Issues



## Consistency

The screenshot shows the SonarQube Issues page for the 'gameoflife-core' project. The 'Issues' tab is selected. On the left, the 'Clean Code Attribute' filter is expanded, showing 'Consistency' selected with 197k issues. The main panel displays several consistency-related code smells:

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

At the bottom, a warning message reads: "Embedded database should be used for evaluation purposes only".

## Intentionality

The screenshot shows the SonarQube Issues page for the 'gameoflife-acceptance-tests' project. The 'Issues' tab is selected. The 'Clean Code Attribute' filter is expanded, showing 'Intentionality' selected with 14k issues. The main panel displays several intentionality-related code smells:

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

At the bottom, a warning message reads: "Embedded database should be used for evaluation purposes only".

## Code Smells

**sonarqube** Projects Issues Rules Quality Profiles Quality Gates Administration More Q

star sonarqube-test / main ?

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

Severity: High: 0, Medium: 0, Low: 253

Type: Bug: 14K, Vulnerability: 0, Code Smell: 253

Scope, Status, Security Category

Bulk Change Select issues Navigate to issue 253 issues 2d 5h effort

gameoflife-web/tools/meter/printable\_docs/building.html

Add an "alt" attribute to this image. Reliability: Open Not assigned L29 - 5min effort - 4 years ago - Code Smell - Minor Intentionality: accessibility wcag2-a

gameoflife-web/tools/jmeter/printable\_docs/changes.html

Add an "alt" attribute to this image. Reliability: Open Not assigned L31 - 5min effort - 4 years ago - Code Smell - Minor Intentionality: accessibility wcag2-a

gameoflife-web/tools/jmeter/printable\_docs/changes\_history.html

Add an "alt" attribute to this image. Intentionality: accessibility wcag2-a

Embedded database should be used for evaluation purposes only

This embedded database will not persist. It will not connect automatically to remote instances of SonarQube, and there is no support for migrating your data out of it into a different database system.

## Bugs

**sonarqube** Projects Issues Rules Quality Profiles Quality Gates Administration More Q

star sonarqube-test / main ?

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

Severity: High: 0, Medium: 14K, Low: 0

Type: Bug: 14K, Vulnerability: 0, Code Smell: 253

Scope, Status, Security Category

Bulk Change Select issues Navigate to issue 13,619 issues 56d effort

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

Add "lang" and/or "xml:lang" attributes to this "<html>" element. Reliability: Open Not assigned L1 - 2min effort - 4 years ago - Bug - Major Intentionality: accessibility wcag2-a

Add "<th>" headers to this "<table>". Reliability: Open Not assigned L1 - 2min effort - 4 years ago - Bug - Major Intentionality: accessibility wcag2-a

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

Add "lang" and/or "xml:lang" attributes to this "<html>" element. Reliability: Open Not assigned L9 - 2min effort - 4 years ago - Bug - Major Intentionality: accessibility wcag2-a

Embedded database should be used for evaluation purposes only

## Reliability

**Issues** tab selected. Project: sonarqube-test / main. 13,619 issues, 56d effort.

- Clean Code Attribute**: Intentionality (14k)
- Software Quality**: Reliability (14k)
- Severity**: Medium (14k)

Issue details for gameoflife-core/build/reports/tests/all-tests.html:

- Add "lang" and/or "xml:lang" attributes to this "<html>" element (Intentionality, Reliability)
- Add "<th>" headers to this "<table>" (Intentionality, Reliability)

Message: Embedded database should be used for evaluation purposes only

## Duplicates

**Measures** tab selected. Project: sonarqube-test / main. 1,147 files.

Duplicated Lines (%): 50.6% (See history)

|                         | Duplicated Lines (%) | Duplicated Lines |
|-------------------------|----------------------|------------------|
| ReportCellRenderer.html | 92.4%                | 1,282            |
| RightAlignRenderer.html | 92.4%                | 1,198            |
| JMeterCellRenderer.html | 92.1%                | 1,281            |
| FunctionHelper.html     | 89.5%                | 1,070            |
| AjpSamplerGui.html      | 89.0%                | 1,219            |

## Security Hotspot

The screenshot shows the SonarQube interface for a project named "sonarqube-test". The "Security Hotspots" tab is selected. A prominent red circle indicates 0.0% Security Hotspots Reviewed. A single critical hotspot is listed:

- Review priority:** Medium (orange)
- Category:** Permission
- Assignee:** Not assigned
- Description:** The tomcat image runs with root as the default user. Make sure it is safe here.
- Status:** To review
- Code Snippet:**

```
FROM tomcat:8-jre8
EXPOSE 8080
CMD ["catalina.sh", "run"]
```

## Cyclomatic Complexity

The screenshot shows the SonarQube interface for a project named "sonarqube-test". The "Measures" tab is selected. On the left, a sidebar lists various metrics, with "Cyclomatic Complexity" highlighted. The main panel displays the following information:

- Cyclomatic Complexity:** 1,112 (highlighted in blue)
- View as:** Tree
- Select files:** pom.xml
- Files:** 6 files
- Complexity by file:**
  - gameoflife-acceptance-tests: 18
  - gameoflife-build: 18
  - gameoflife-core: 18
  - gameoflife-deploy: 18
  - gameoflife-web: 1,094
  - pom.xml: 18

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.

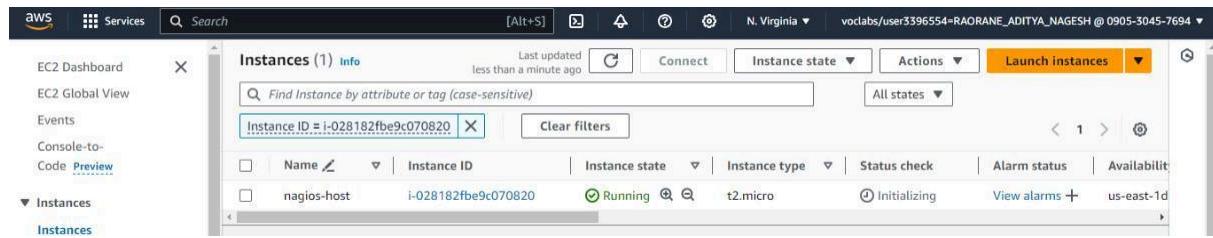
### Conclusion:

In this experiment, we performed a static analysis of the code to detect bugs, code smells, and security vulnerabilities on our sample Java application.

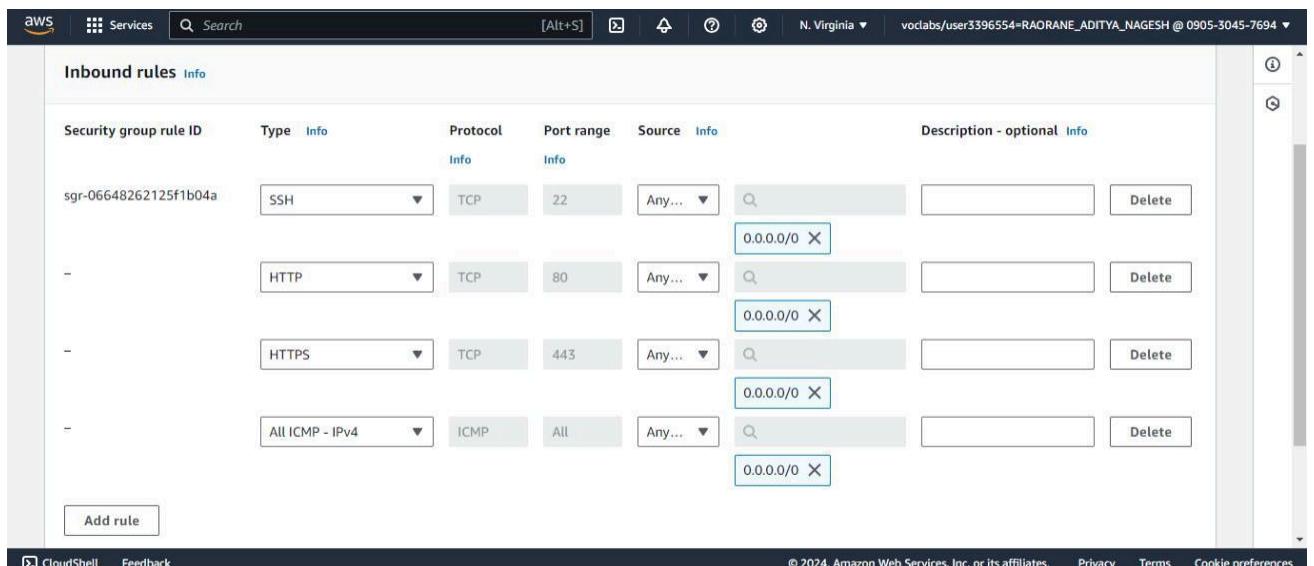
## Experiment 09

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

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



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



### 3. SSH into Your EC2 instance.

```
C:\Users\INFT505-12\Downloads>ssh -i "aditya.pem" ec2-user@ec2-34-207-192-114.compute-1.amazonaws.com
'~\_\ ####_          Amazon Linux 2023
~~ \_\#####\
~~ \|##|
~~ \|#/ __ https://aws.amazon.com/linux/amazon-linux-2023
~~ \|~' '->
~~ /`_
~~ /`_/
~~ /`_/
~/m/`_
[ec2-user@ip-172-31-84-15 ~]$ |
```

### 4. Update the package indices and install the following packages

```
sudo yum update
sudo yum install httpd php
sudo yum install gcc glibc glibc-common
sudo yum install gd gd-devel
```

```
[ec2-user@ip-172-31-84-15 ~]$ sudo yum update -y
sudo yum install httpd php -y
sudo yum install gcc glibc glibc-common -y
sudo yum install gd gd-devel -y
Last metadata expiration check: 0:04:01 ago on Mon Sep 23 08:24:16 2024.
Dependencies resolved.
Nothing to do.
Complete!
Last metadata expiration check: 0:04:01 ago on Mon Sep 23 08:24:16 2024.
Dependencies resolved.

=====
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
generic-logos-httd noarch      18.0.0-12.amzn2023.0.3 amazonlinux 19 k
httpd-core       x86_64       2.4.62-1.amzn2023 amazonlinux 1.4 M
httpd-filesystem noarch      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
libsodium         x86_64       1.0.19-4.amzn2023 amazonlinux 176 k
libxml2          x86_64       1.1.34-5.amzn2023.0.2 amazonlinux 241 k
mailcap          noarch      2.1.49-3.amzn2023.0.3 amazonlinux 33 k
nginx-filesystem noarch      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-xml       x86_64       8.3.10-1.amzn2023.0.1 amazonlinux 184 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
```

```

Installed:
  brotli-1.0.9-4.amzn2023.0.2.x86_64
  bzip2-devel-1.0.8-6.amzn2023.0.2.x86_64
  cmake-fs-3.22.2-1.amzn2023.0.4.x86_64
  fontconfig-devel-2.13.94-2.amzn2023.0.2.x86_64
  freetype-2.13.2-5.amzn2023.0.1.x86_64
  gd-2.3.3-5.amzn2023.0.3.x86_64
  glib2-devel-2.74.7-689.amzn2023.0.2.x86_64
  google-noto-sans-vf-fonts-20201206-2.amzn2023.0.2.noarch
  graphite2-devel-1.3.14-7.amzn2023.0.2.x86_64
  harfbuzz-devel-7.0.0-2.amzn2023.0.1.x86_64
  jbigkit-libs-2.1-21.amzn2023.0.2.x86_64
  libICE-1.0.10-6.amzn2023.0.2.x86_64
  libX11-1.7.2-3.amzn2023.0.4.x86_64
  libX11-devel-1.7.2-3.amzn2023.0.4.x86_64
  libXau-1.0.9-6.amzn2023.0.2.x86_64
  libXext-1.3.4-6.amzn2023.0.2.x86_64
  libXpm-devel-3.5.15-2.amzn2023.0.3.x86_64
  libXt-1.2.0-4.amzn2023.0.2.x86_64
  libffi-devel-3.4.4-1.amzn2023.0.1.x86_64
  libicu-devel-67.1-7.amzn2023.0.3.x86_64
  libjpeg-turbo-devel-2.1.4-2.amzn2023.0.5.x86_64
  libpng-2.1.6.37-10.amzn2023.0.6.x86_64
  libselinux-devel-3.4-5.amzn2023.0.2.x86_64
  libtiff-4.4.0-4.amzn2023.0.18.x86_64
  libweebp-1.2.4-1.amzn2023.0.6.x86_64
  libxcb-1.13.1-7.amzn2023.0.2.x86_64
  libxml2-devel-2.10.4-1.amzn2023.0.6.x86_64
  pcre2-utf16-10.40-1.amzn2023.0.3.x86_64
  pixman-0.40.0-3.amzn2023.0.3.x86_64
  xml-common-0.6.3-56.amzn2023.0.2.noarch
  xz-devel-5.2.5-9.amzn2023.0.2.x86_64

Complete!
[ec2-user@ip-172-31-84-15 ~]$
```

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

```

sudo adduser -m nagios
sudo passwd nagios
```

```

[ec2-user@ip-172-31-84-15 ~]$ sudo passwd nagios
Changing password for user nagios.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-84-15 ~]$ |
```

6. Create a new user group

```
sudo groupadd nagcmd
```

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

```

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

8. Create a new directory for Nagios downloads

```

mkdir ~/downloads
cd ~/downloads
```

```
[ec2-user@ip-172-31-84-15 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-84-15 ~]$ sudo usermod -aG nagcmd nagios
sudo usermod -aG nagcmd apache
[ec2-user@ip-172-31-84-15 ~]$ mkdir ~/downloads
cd ~/downloads
[ec2-user@ip-172-31-84-15 downloads]$ |
```

9. Use wget to download the source zip files.

wget <https://go.nagios.org/l/975333/2024-09-17/6kqcx>

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

```
ec2-user@ip-172-31-84-15 downloads]$ wget http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.0.8.tar.gz
wget http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz
--2024-09-23 08:33:27-- http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.0.8.tar.gz
Resolving prdownloads.sourceforge.net (prdownloads.sourceforge.net)... 204.68.111.105
Connecting to prdownloads.sourceforge.net (prdownloads.sourceforge.net)|204.68.111.105|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: http://downloads.sourceforge.net/project/nagios/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz [following]
--2024-09-23 08:33:27-- http://downloads.sourceforge.net/project/nagios/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz
Resolving downloads.sourceforge.net (downloads.sourceforge.net)... 204.68.111.105
Reusing existing connection to prdownloads.sourceforge.net:80.
HTTP request sent, awaiting response... 302 Found
Location: http://pilotfiber.dl.sourceforge.net/project/nagios/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz?viafs=1 [following]
--2024-09-23 08:33:27-- http://pilotfiber.dl.sourceforge.net/project/nagios/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz?viafs=1
Resolving pilotfiber.dl.sourceforge.net (pilotfiber.dl.sourceforge.net)|216.158.133.250
Connecting to pilotfiber.dl.sourceforge.net (pilotfiber.dl.sourceforge.net)|216.158.133.250|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1805059 (1.7M) [application/x-gzip]
Saving to: 'nagios-4.0.8.tar.gz'

nagios-4.0.8.tar.gz          100%[=====] 1.72M  10.8MB/s   in 0.2s
2024-09-23 08:33:28 (10.8 MB/s) - 'nagios-4.0.8.tar.gz' saved [1805059/1805059]

--2024-09-23 08:33:28-- http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251
Connecting to nagios-plugins.org (nagios-plugins.org)|45.56.123.251|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2659772 (2.5M) [application/x-gzip]
Saving to: 'nagios-plugins-2.0.3.tar.gz'

nagios-plugins-2.0.3.tar.gz    100%[=====] 2.54M  8.81MB/s   in 0.3s
2024-09-23 08:33:28 (8.81 MB/s) - 'nagios-plugins-2.0.3.tar.gz' saved [2659772/2659772]
[ec2-user@ip-172-31-84-15 downloads]$ |
```

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

tar zxvf nagios-4.5.5.tar.gz

cd nagios-4.5.5

```
[ec2-user@ip-172-31-80-195 downloads]$ cd nagios-4.5.5
[ec2-user@ip-172-31-80-195 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 string.h... (cached) yes
checking for strings.h... (cached) yes
```

```
[ec2-user@ip-172-31-80-195 nagios-4.5.5]$ sudo yum install openssl-devel
Last metadata expiration check: 0:10:42 ago on Fri Sep 27 11:19:57 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           14 MB/s | 3.0 MB  00:00
 -----
 Total   11 MB/s | 3.0 MB  00:00
 Running transaction check
 Transaction check succeeded.
 Running transaction test
 Transaction test succeeded.
 Running transaction
   Preparing           :                                1/1
   Installing         : openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64 1/1
   Running scriptlet: openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64 1/1
   Verifying          : openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64 1/1
 -----
 Installed:
   openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64
 Complete!
```

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

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

12. Compile the source code. make all

```

Command Prompt x + v
*** Configuration summary for nagios 4.5.5 2024-09-17 ***

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

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

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

[ec2-user@ip-172-31-80-195 nagios-4.5.5]$ make all
cd ./base && make
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o nagios.o ./nagios.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o broker.o broker.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o nebmods.o nebmods.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o common/shared.o ./common/shared.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o query-handler.o query-handler.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o workers.o workers.c
In function 'get_wproc_list',
  inlined from 'get_worker' at workers.c:277:12:
workers.c:253:17: warning: '%s' directive argument is null [-Wformat-overflow=]
  253 |         log_debug_info(DEBUGL_CHECKS, 1, "Found specialized worker(s) for '%s'", (slash && *slash != '/') ? slash : cmd_name);
  |         ^
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o checks.o checks.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o config.o config.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o commands.o commands.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_CONFIG_H -DSCORE -c -o events.o events.c

```

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

sudo make install sudo

make install-init sudo

make install-config

sudo make install-commandmode

```
[ec2-user@ip-172-31-80-195 nagios-4.5.5]$ sudo make install
sudo make install-init
sudo make install-config
sudo make install-commandmode
cd ./base && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagios /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagiosstats /usr/local/nagios/bin
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/base'
cd ./cgi && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
make install-basic
make[2]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/sbin
for file in *.cgi; do \
    /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/contextthelp
/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
```

#### 14. Configure the web

interface. sudo make  
install-webconf

```
[ec2-user@ip-172-31-80-195 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 ***
```

15. Create a nagiosadmin account for nagios login along with password. You'll have to specify the password twice. (adityaraorane)

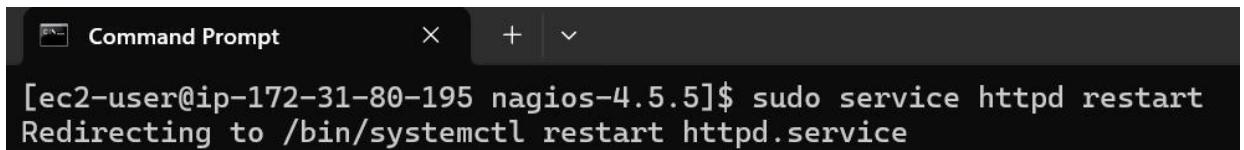
```
sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```



```
[ec2-user@ip-172-31-80-195 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
```

16. Restart Apache

```
sudo service httpd restart
```



```
[ec2-user@ip-172-31-80-195 nagios-4.5.5]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
```

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

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

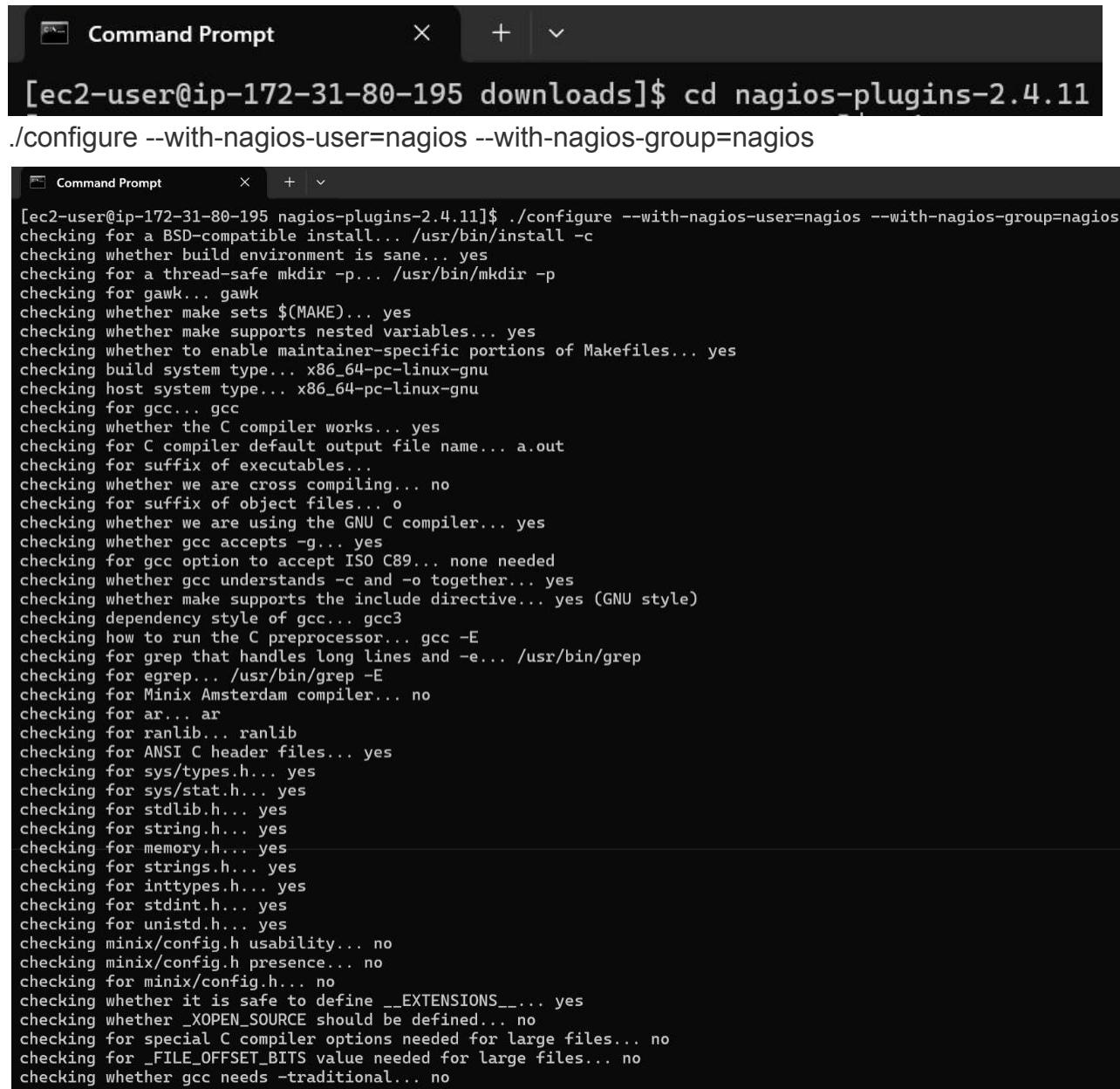


The screenshot shows a terminal window titled "Command Prompt". The user has run two commands: "ls" to list the contents of the "downloads" directory, which contains the "nagios-4.5.5" and "nagios-plugins-2.4.11.tar.gz" files; and "tar zxvf nagios-plugins-2.4.11.tar.gz" to extract the tarball into the current directory. The extracted directory structure for "nagios-plugins-2.4.11" is displayed, showing numerous sub-directories and m4 files related to the plugin compilation process.

```
[ec2-user@ip-172-31-80-195 downloads]$ ls
6kqcx nagios-4.5.5 nagios-plugins-2.4.11.tar.gz
[ec2-user@ip-172-31-80-195 downloads]$ tar zxvf nagios-plugins-2.4.11.tar.gz
nagios-plugins-2.4.11/
nagios-plugins-2.4.11/build-aux/
nagios-plugins-2.4.11/build-aux/compile
nagios-plugins-2.4.11/build-aux/config.guess
nagios-plugins-2.4.11/build-aux/config.rpath
nagios-plugins-2.4.11/build-aux/config.sub
nagios-plugins-2.4.11/build-aux/install-sh
nagios-plugins-2.4.11/build-aux/ltmain.sh
nagios-plugins-2.4.11/build-aux/missing
nagios-plugins-2.4.11/build-aux/mkinstalldirs
nagios-plugins-2.4.11/build-aux/depcomp
nagios-plugins-2.4.11/build-aux/snippet/
nagios-plugins-2.4.11/build-aux/snippet/_Noreturn.h
nagios-plugins-2.4.11/build-aux/snippet/arg-nonnull.h
nagios-plugins-2.4.11/build-aux/snippet/c++defs.h
nagios-plugins-2.4.11/build-aux/snippet/warn-on-use.h
nagios-plugins-2.4.11/build-aux/test-driver
nagios-plugins-2.4.11/config_test/
nagios-plugins-2.4.11/config_test/Makefile
nagios-plugins-2.4.11/config_test/run_tests
nagios-plugins-2.4.11/config_test/child_test.c
nagios-plugins-2.4.11/gl/
nagios-plugins-2.4.11/gl/m4/
nagios-plugins-2.4.11/gl/m4/00gnulib.m4
nagios-plugins-2.4.11/gl/m4/absolute-header.m4
nagios-plugins-2.4.11/gl/m4/alloca.m4
nagios-plugins-2.4.11/gl/m4/arpa_inet_h.m4
nagios-plugins-2.4.11/gl/m4/base64.m4
nagios-plugins-2.4.11/gl/m4/btowc.m4
nagios-plugins-2.4.11/gl/m4/codeset.m4
nagios-plugins-2.4.11/gl/m4/configmake.m4
nagios-plugins-2.4.11/gl/m4 dirname.m4
nagios-plugins-2.4.11/gl/m4 double-slash-root.m4
nagios-plugins-2.4.11/gl/m4 eealloc.m4
nagios-plugins-2.4.11/gl/m4 environ.m4
nagios-plugins-2.4.11/gl/m4 errno_h.m4
nagios-plugins-2.4.11/gl/m4 error.m4
nagios-plugins-2.4.11/gl/m4 exponentd.m4
nagios-plugins-2.4.11/gl/m4 extensions.m4
nagios-plugins-2.4.11/gl/m4 extern-inline.m4
nagios-plugins-2.4.11/gl/m4 fcntl-o.m4
nagios-plugins-2.4.11/gl/m4 float_h.m4
nagios-plugins-2.4.11/gl/m4 floorf.m4
```

## 18. Compile and install plugins

```
cd nagios-plugins-2.0.3
```



The screenshot shows a terminal window titled "Command Prompt". The command entered is `[ec2-user@ip-172-31-80-195 downloads]$ cd nagios-plugins-2.4.11`. Below this, the command `./configure --with-nagios-user=nagios --with-nagios-group=nagios` is run. The terminal then displays a series of "checking" messages as the configure script runs through various system checks:

```
[ec2-user@ip-172-31-80-195 nagios-plugins-2.4.11]$ ./configure --with-nagios-user=nagios --with-nagios-group=nagios
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether to enable maintainer-specific portions of Makefiles... yes
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether gcc understands -c and -o together... yes
checking whether make supports the include directive... yes (GNU style)
checking dependency style of gcc... gcc3
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
checking for Minix Amsterdam compiler... no
checking for ar... ar
checking for ranlib... ranlib
checking for ANSI C header files... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for memory.h... yes
checking for strings.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for unistd.h... yes
checking minix/config.h usability... no
checking minix/config.h presence... no
checking for minix/config.h... no
checking whether it is safe to define __EXTENSIONS__... yes
checking whether _XOPEN_SOURCE should be defined... no
checking for special C compiler options needed for large files... no
checking for _FILE_OFFSET_BITS value needed for large files... no
checking whether gcc needs -traditional... no
```

```
make
```

```
[ec2-user@ip-172-31-80-195 nagios-plugins-2.4.11]$ ./configure --with-nagios-user=nagios --with-nagios-group=nagios
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether to enable maintainer-specific portions of Makefiles... yes
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether gcc understands -c and -o together... yes
checking whether make supports the include directive... yes (GNU style)
checking dependency style of gcc... gcc3
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
checking for Minix Amsterdam compiler... no
checking for ar... ar
checking for ranlib... ranlib
checking for ANSI C header files... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for memory.h... yes
checking for strings.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for unistd.h... yes
checking minix/config.h usability... no
checking minix/config.h presence... no
checking for minix/config.h... no
checking whether it is safe to define __EXTENSIONS__... yes
checking whether _XOPEN_SOURCE should be defined... no
checking for special C compiler options needed for large files... no
checking for _FILE_OFFSET_BITS value needed for large files... no
checking whether gcc needs -traditional... no
```

```
[ec2-user@ip-172-31-80-195 nagios-plugins-2.4.11]$ sudo make install
all
Making install in gl
make[1]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/gl'
make  install-recursive
make[2]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/gl'
make[3]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/gl'
make[4]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/gl'
if test yes = no; then \
    case 'linux-gnu' in \
        darwin[56]*) \
            need_charset_alias=true ; \
        darwin* | cygwin* | mingw* | pw32* | cegcc*) \
            need_charset_alias=false ; \
        *) \
            need_charset_alias=true ; \
    esac ; \
else \
    need_charset_alias=false ; \
fi ; \
if $need_charset_alias; then \
    /bin/sh ../build-aux/mkinstalldirs /usr/local/nagios/lib ; \
fi ; \
if test -f /usr/local/nagios/lib/charset.alias; then \
    sed -f ref-add.sed /usr/local/nagios/lib/charset.alias > /usr/local/nagios/lib/charset.tmp ; \
    /usr/bin/install -c -o nagios -g nagios -m 644 /usr/local/nagios/lib/charset.tmp /usr/local/nagios/lib/charset.alias ; \
    rm -f /usr/local/nagios/lib/charset.tmp ; \
else \
    if $need_charset_alias; then \
        sed -f ref-add.sed charset.alias > /usr/local/nagios/lib/charset.tmp ; \
        /usr/bin/install -c -o nagios -g nagios -m 644 /usr/local/nagios/lib/charset.tmp /usr/local/nagios/lib/charset.alias ; \
        rm -f /usr/local/nagios/lib/charset.tmp ; \
    fi ; \
fi
make[4]: Nothing to be done for 'install-data-am'.
make[4]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/gl'
make[3]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/gl'
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/gl'
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/gl'
Making install in tap
make[1]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/tap'
make[2]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/tap'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/tap'
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/tap'
Making install in lib

```

## 19. Start Nagios

Add Nagios to the list of system services

```
sudo chkconfig --add nagios
sudo chkconfig nagios on
```

```
[ec2-user@ip-172-31-80-195 nagios-plugins-2.4.11]$ sudo chkconfig --add nagios
sudo chkconfig nagios on
error reading information on service nagios: No such file or directory
Note: Forwarding request to 'systemctl enable nagios.service'.
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service → /usr/lib/systemd/system/nagios.service.
```

Verify the sample configuration files

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

```
Command Prompt [ec2-user@ip-172-31-80-195 nagios-plugins-2.4.11]$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg  
Nagios Core 4.5.5  
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors  
Copyright (c) 1999-2009 Ethan Galstad  
Last Modified: 2024-09-17  
License: GPL  
  
Website: https://www.nagios.org  
Reading configuration data...  
    Read main config file okay...  
    Read object config files okay...  
  
Running pre-flight check on configuration data...  
  
Checking objects...  
    Checked 8 services.  
    Checked 1 hosts.  
    Checked 1 host groups.  
    Checked 0 service groups.  
    Checked 1 contacts.  
    Checked 1 contact groups.  
    Checked 24 commands.  
    Checked 5 time periods.  
    Checked 0 host escalations.  
    Checked 0 service escalations.  
Checking for circular paths...  
    Checked 1 hosts  
    Checked 0 service dependencies  
    Checked 0 host dependencies  
    Checked 5 timeperiods  
Checking global event handlers...  
Checking obsessive compulsive processor commands...  
Checking misc settings...  
  
Total Warnings: 0  
Total Errors: 0  
  
Things look okay - No serious problems were detected during the pre-flight check
```

sudo service nagios start

```
Command Prompt [ec2-user@ip-172-31-80-195 nagios-plugins-2.4.11]$ cd  
[ec2-user@ip-172-31-80-195 ~]$ sudo service nagios start  
Redirecting to /bin/systemctl start nagios.service
```

## 20. Check the status of Nagios

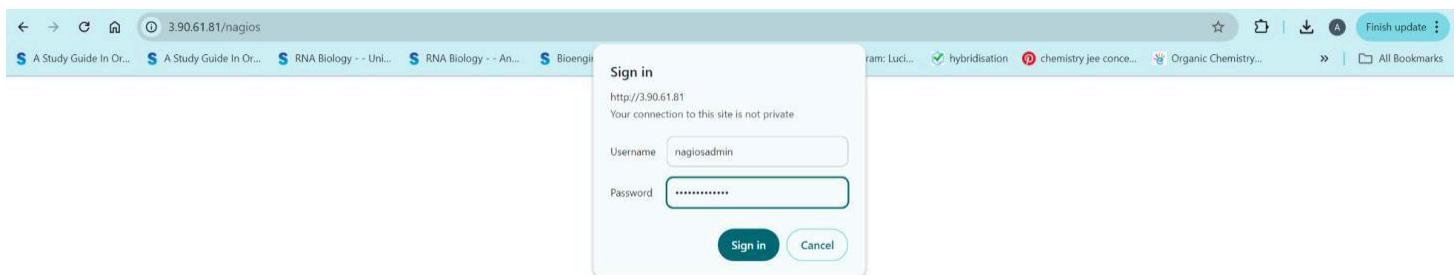
sudo systemctl status nagios

```
Command Prompt × + ▾
[ec2-user@ip-172-31-80-195 ~]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
  Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled)
  Active: active (running) since Fri 2024-09-27 11:43:06 UTC;
            Docs: https://www.nagios.org/documentation
    Process: 65083 ExecStartPre=/usr/local/nagios/bin/nagios -v >
   Process: 65084 ExecStart=/usr/local/nagios/bin/nagios -d /us>
   Main PID: 65085 (nagios)
      Tasks: 6 (limit: 1112)
     Memory: 5.7M
        CPU: 86ms
       CGroup: /system.slice/nagios.service
               └─65085 /usr/local/nagios/bin/nagios -d /usr/local/>
                  ├─65086 /usr/local/nagios/bin/nagios --worker /usr/>
                  ├─65087 /usr/local/nagios/bin/nagios --worker /usr/>
                  ├─65088 /usr/local/nagios/bin/nagios --worker /usr/>
                  ├─65089 /usr/local/nagios/bin/nagios --worker /usr/>
                  └─65090 /usr/local/nagios/bin/nagios -d /usr/local/>

Sep 27 11:43:06 ip-172-31-80-195.ec2.internal nagios[65085]: qh:>
Sep 27 11:43:06 ip-172-31-80-195.ec2.internal nagios[65085]: wpr>
Sep 27 11:43:06 ip-172-31-80-195.ec2.internal nagios[65085]: Suc>
[lines 1-28/28 (END)]
```

21. Go back to EC2 Console and copy the Public IP address of this instance. Open up your browser and look for [http://<your\\_public\\_ip\\_address>/nagios](http://<your_public_ip_address>/nagios).

<http://3.90.61.81/nagios>



Enter username as **nagiosadmin** and password as **adityaraorane** (which you set in Step 15).

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

The screenshot shows the Nagios Core dashboard. At the top right, it says "Nagios® Core™ Version 4.5.5" and "September 17, 2024". Below that, a green checkmark indicates "Daemon running with PID 65085". The left sidebar has sections for General, Current Status, Service Groups, Problems, Reports, and System. The Current Status section is expanded, showing links for Hosts, Services, Host Groups, and Network Outages. The Service Groups section shows a summary and grid view. The Problems section shows services and hosts with unhandled issues. The Reports section includes Availability, Trends, Alerts, History, Summary, Histogram, Notifications, and Event Log. The System section includes Comments, Downtime, Process Info, Performance Info, Scheduling Queue, and Configuration. The main content area has three boxes: "Get Started" with a list of monitoring steps, "Latest News" which is empty, and "Don't Miss..." which is also empty. To the right, there is a "Quick Links" box with links to Nagios Library, Labs, Exchange, Support, and the official websites. A copyright notice at the bottom states: "Copyright © 2010-2024 Nagios Core Development Team and Community Contributors. Copyright © 1999-2009 Ethan Galstad. See the THANKS file for more information on contributors." It also notes that Nagios Core is licensed under the GNU General Public License. A "Page Tour" link is visible on the far right.

**Conclusion:** We have successfully installed and configured Nagios Core, Nagios Plugins, and NRPE on a Linux machine. This enables us to effectively manage system performance and proactively address potential issues.

## Experiment 10

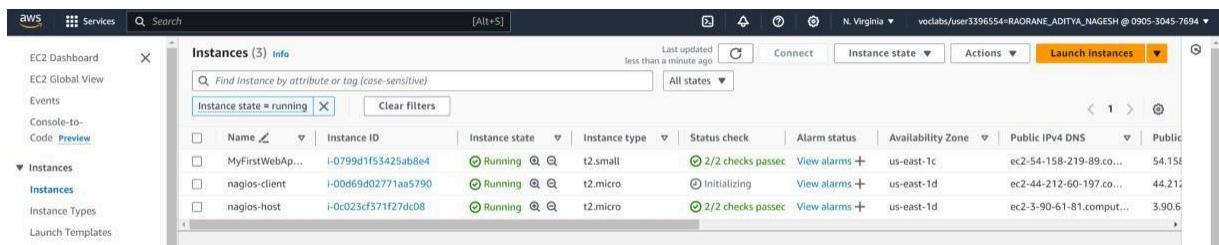
**Aim:** To perform Port, Service monitoring, Windows/Linux server monitoring using Nagios.

1. To Confirm that Nagios is running **on the server side**, run this **sudo systemctl status nagios** on the **nagios-host**.

```
[ec2-user@ip-172-31-80-195 ~]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
    Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; v
      Active: active (running) since Fri 2024-09-27 11:43:06 UTC; 10s ago
        Docs: https://www.nagios.org/documentation
    Process: 65083 ExecStartPre=/usr/local/nagios/bin/nagios -v >
    Process: 65084 ExecStart=/usr/local/nagios/bin/nagios -d /us>
    Main PID: 65085 (nagios)
      Tasks: 6 (limit: 1112)
     Memory: 5.7M
        CPU: 86ms
       CGroup: /system.slice/nagios.service
               └─65085 /usr/local/nagios/bin/nagios -d /usr/local/>
                 ├─65086 /usr/local/nagios/bin/nagios --worker /usr/>
                 ├─65087 /usr/local/nagios/bin/nagios --worker /usr/>
                 ├─65088 /usr/local/nagios/bin/nagios --worker /usr/>
                 ├─65089 /usr/local/nagios/bin/nagios --worker /usr/>
                 └─65090 /usr/local/nagios/bin/nagios -d /usr/local/>

Sep 27 11:43:06 ip-172-31-80-195.ec2.internal nagios[65085]: qh:>
Sep 27 11:43:06 ip-172-31-80-195.ec2.internal nagios[65085]: wpr>
Sep 27 11:43:06 ip-172-31-80-195.ec2.internal nagios[65085]: Suc:>
```

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



**For now, leave this machine as is, and go back to your nagios-host.**

### 3. On the server, run this

command ps -ef | grep nagios

```
[ec2-user@ip-172-31-80-195 ~]$ ps -ef | grep nagios
nagios  65085      1  0 11:43 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios  65086  65085  0 11:43 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios  65087  65085  0 11:43 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios  65088  65085  0 11:43 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios  65089  65085  0 11:43 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios  65090  65085  0 11:43 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
ec2-user 65683     2261  0 11:52 pts/0    00:00:00 grep --color=auto nagios
```

### 4. Become a root user and create 2

folders sudo su

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

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

```
[ec2-user@ip-172-31-80-195 ~]$ sudo su
mkdir /usr/local/nagios/etc/objects/monitorhosts
mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
```

### 5. Copy the sample localhost.cfg file to linuxhost folder

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

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

```
[root@ip-172-31-80-195 ec2-user]# sudo cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
[root@ip-172-31-80-195 ec2-user]# nano
/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
```

### 6. Open linuxserver.cfg using nano and make the following changes

nano

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

```
[root@ip-172-31-80-195 ec2-user]# nano /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
[root@ip-172-31-80-195 ec2-user]# nano /usr/local/nagios/etc/nagios.cfg
[root@ip-172-31-80-195 ec2-user]# nano /usr/local/nagios/etc/nagios.cfg
```

```
GNU nano 5.8                               /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg

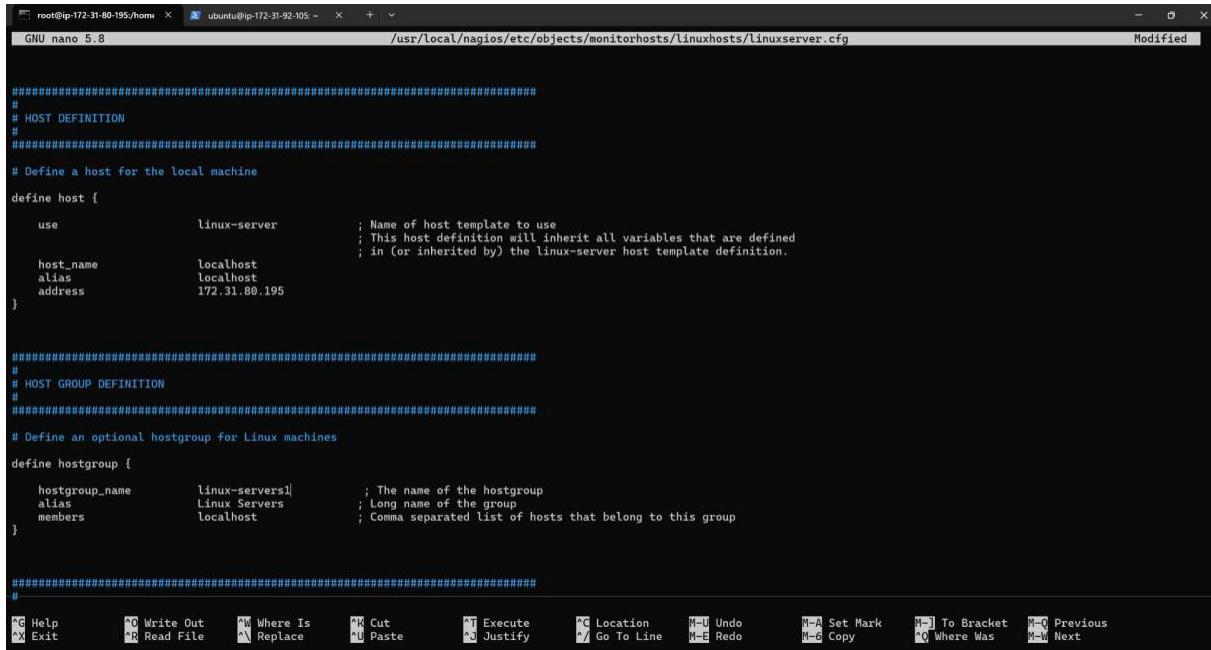
#####
# HOST DEFINITION
#
#####

# 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      localhost
    alias          localhost
    address        172.31.80.195
}

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



```

GNU nano 5.8                               /usr/local/nagios/etc/objects/monitorhosts/linuxserver.cfg
Modified

#####
# 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    localhost
    alias        localhost
    address     172.31.80.195
}

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

# Define an optional hostgroup for Linux machines

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

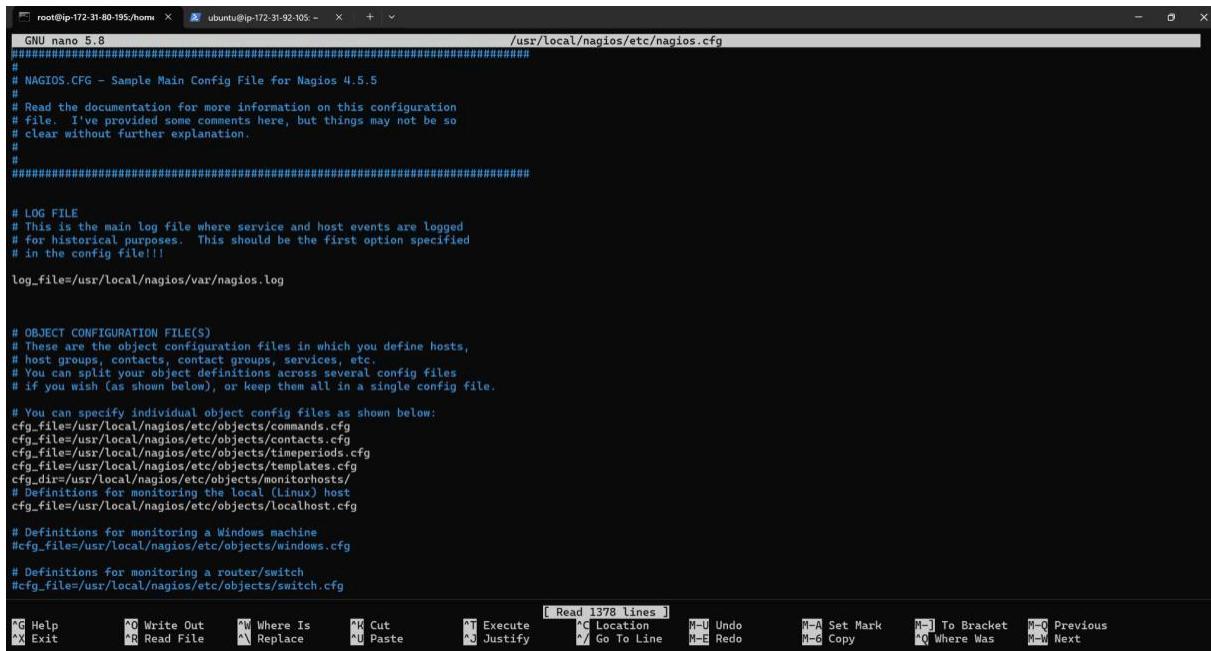
#####
#
```

File menu: Help, Write Out, Read File, Where Is, Cut, Paste, Execute, Justify, Location, Go To Line, Undo, Redo, Set Mark, To Bracket, Copy, Where Was, Previous, Next.

## 7. Open the Nagios Config file and add the following

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

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



```

GNU nano 5.8                               /usr/local/nagios/etc/nagios.cfg
Modified

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

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

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

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

# Definitions for monitoring a Windows machine
#cfg_file=/usr/local/nagios/etc/objects/windows.cfg

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

[ Read 1378 lines ]
```

File menu: Help, Write Out, Read File, Where Is, Cut, Paste, Execute, Justify, Location, Go To Line, Undo, Redo, Set Mark, To Bracket, Copy, Where Was, Previous, Next.

## 8. Verify the configuration files

```
[root@ip-172-31-80-195 ec2-user]# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
Nagios Core 4.5.5
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2024-09-17
License: GPL

Website: https://www.nagios.org
Reading configuration data...
  Read main config file okay...
  Warning: Duplicate definition found for service 'HTTP' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', starting on line 1
  92)
  Warning: Duplicate definition found for service 'SSH' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', starting on line 13
  8)
  Warning: Duplicate definition found for service 'Swap Usage' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', starting on
  line 125)
  Warning: Duplicate definition found for service 'Current Load' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', starting o
  n line 112)
  Warning: Duplicate definition found for service 'Total Processes' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', startin
  g on line 100)
  Warning: Duplicate definition found for service 'Current Users' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', starting
  on line 86)
  Warning: Duplicate definition found for service 'Root Partition' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', starting
  on line 72)
  Warning: Duplicate definition found for service 'PING' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', starting on line 5
  8)
  Read object config files okay...
Running pre-flight check on configuration data...

Checking objects...
  Checked 8 services.
  Checked 2 hosts.
  Checked 2 host groups.
  Checked 0 service groups.
  Checked 1 contacts.
  Checked 1 contact groups.
  Checked 24 commands.
  Checked 5 time periods.
  Checked 0 host escalations.
  Checked 0 service escalations.
Checking for circular paths...
  Checked 2 hosts
  Checked 0 service dependencies
  Checked 0 host dependencies
```

```
[root@ip-172-31-80-195 ec2-user]# 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 Fri 2024-09-27 12:24:39 UTC; 23s ago
     Docs: https://www.nagios.org/documentation
 Process: 67569 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Process: 67568 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 67569 (nagios)
   Tasks: 6 (limit: 1112)
    Memory: 4.0M
      CPU: 21ms
      CGroup: /system.slice/nagios.service
              ├─67569 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
              ├─67570 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─67571 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─67572 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─67573 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              └─67574 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 27 12:24:39 ip-172-31-80-195.ec2.internal nagios[67569]: wproc: Registry request: name=Cores Worker 67570;pid=67570
Sep 27 12:24:39 ip-172-31-80-195.ec2.internal nagios[67569]: Warning: Duplicate definition found for service 'HTTP' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monit
Sep 27 12:24:39 ip-172-31-80-195.ec2.internal nagios[67569]: Warning: Duplicate definition found for service 'SSH' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monit
Sep 27 12:24:39 ip-172-31-80-195.ec2.internal nagios[67569]: Warning: Duplicate definition found for service 'Swap Usage' on host 'localhost' (config file '/usr/local/nagios/etc/objec
Sep 27 12:24:39 ip-172-31-80-195.ec2.internal nagios[67569]: Warning: Duplicate definition found for service 'Current Load' on host 'localhost' (config file '/usr/local/nagios/etc/objec
Sep 27 12:24:39 ip-172-31-80-195.ec2.internal nagios[67569]: Warning: Duplicate definition found for service 'Total Processes' on host 'localhost' (config file '/usr/local/nagios/etc/objec
Sep 27 12:24:39 ip-172-31-80-195.ec2.internal nagios[67569]: Warning: Duplicate definition found for service 'Current Users' on host 'localhost' (config file '/usr/local/nagios/etc/objec
Sep 27 12:24:39 ip-172-31-80-195.ec2.internal nagios[67569]: Warning: Duplicate definition found for service 'Root Partition' on host 'localhost' (config file '/usr/local/nagios/etc/objec
Sep 27 12:24:39 ip-172-31-80-195.ec2.internal nagios[67569]: Warning: Duplicate definition found for service 'PING' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monit
Sep 27 12:24:39 ip-172-31-80-195.ec2.internal nagios[67569]: Successfully launched command file worker with pid 67574
lines 1-28 (END)
```

## 9. Restart the nagios service

```
[root@ip-172-31-80-195 ec2-user]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
[root@ip-172-31-80-195 ec2-user]# 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 Fri 2024-09-27 12:24:39 UTC; 23s ago
     Docs: https://www.nagios.org/documentation
 Process: 67567 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Process: 67568 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 67569 (nagios)
   Tasks: 6 (limit: 1112)
    Memory: 4.0M
      CPU: 21ms
      CGroup: /system.slice/nagios.service
              ├─67569 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
              ├─67570 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─67571 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─67572 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─67573 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              └─67574 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 27 12:24:39 ip-172-31-80-195.ec2.internal nagios[67569]: wproc: Registry request: name=Cores Worker 67570;pid=67570
Sep 27 12:24:39 ip-172-31-80-195.ec2.internal nagios[67569]: Warning: Duplicate definition found for service 'HTTP' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monit
Sep 27 12:24:39 ip-172-31-80-195.ec2.internal nagios[67569]: Warning: Duplicate definition found for service 'SSH' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monit
Sep 27 12:24:39 ip-172-31-80-195.ec2.internal nagios[67569]: Warning: Duplicate definition found for service 'Swap Usage' on host 'localhost' (config file '/usr/local/nagios/etc/objec
Sep 27 12:24:39 ip-172-31-80-195.ec2.internal nagios[67569]: Warning: Duplicate definition found for service 'Current Load' on host 'localhost' (config file '/usr/local/nagios/etc/objec
Sep 27 12:24:39 ip-172-31-80-195.ec2.internal nagios[67569]: Warning: Duplicate definition found for service 'Total Processes' on host 'localhost' (config file '/usr/local/nagios/etc/objec
Sep 27 12:24:39 ip-172-31-80-195.ec2.internal nagios[67569]: Warning: Duplicate definition found for service 'Current Users' on host 'localhost' (config file '/usr/local/nagios/etc/objec
Sep 27 12:24:39 ip-172-31-80-195.ec2.internal nagios[67569]: Warning: Duplicate definition found for service 'Root Partition' on host 'localhost' (config file '/usr/local/nagios/etc/objec
Sep 27 12:24:39 ip-172-31-80-195.ec2.internal nagios[67569]: Warning: Duplicate definition found for service 'PING' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monit
Sep 27 12:24:39 ip-172-31-80-195.ec2.internal nagios[67569]: Successfully launched command file worker with pid 67574
lines 1-28 (END)
```

## 10. Under nagios-client ,SSH into the machine or simply use the EC2 Instance Connect feature.

```

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\adity> cd Downloads
PS C:\Users\adity\Downloads> ssh -i "ar.pem" ubuntu@ec2-44-212-60-197.compute-1.amazonaws.com
The authenticity of host 'ec2-44-212-60-197.compute-1.amazonaws.com (44.212.60.197)' can't be established.
ED25519 key fingerprint is SHA256:Lwrr9v/VJNu2G/UrjnK/N9yYvSdOCbsG7Ppk83eeKWo.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-212-60-197.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

```

11. Make a package index update and install gcc, nagios-nrpe-server and the plugins.

```

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

```

```

ubuntu@ip-172-31-92-105:~$ sudo apt update -y
sudo apt install gcc -y
sudo apt install -y nagios-nrpe-server nagios-plugins
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [380 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [535 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [130 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [8636 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [377 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [156 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [45.0 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [14.8 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [353 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [68.1 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 c-n-f Metadata [424 B]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [14.4 kB]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [3608 B]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [212 B]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [532 B]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 B]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [10.6 kB]
Get:31 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [10.8 kB]
Get:32 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [17.6 kB]
Get:33 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1104 B]
Get:34 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:35 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B]
Get:36 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:37 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:38 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [82.9 kB]
Get:39 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4560 B]
Get:40 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [272 kB]
Get:41 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [115 kB]
Get:42 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [8632 B]
Get:43 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [10.3 kB]

```

12. Open nrpe.cfg file to make changes.

```
sudo nano /etc/nagios/nrpe.cfg
```

Under allowed\_hosts, add your nagios-host IP address like so

```
GNU nano 7.2
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd
allowed_hosts=127.0.0.1,3.90.61.81
```

13. Restart the NRPE server

```
sudo systemctl restart nagios-nrpe-server
```

```
ubuntu@ip-172-31-92-105:~$ sudo nano /etc/nagios/nrpe.cfg
ubuntu@ip-172-31-92-105:~$ sudo systemctl restart nagios-nrpe-server
ubuntu@ip-172-31-92-105:~$
```

14. Now, check your nagios dashboard and you'll see a new host being added.

| Host Status Details For All Host Groups |        |                     |               |                                           |
|-----------------------------------------|--------|---------------------|---------------|-------------------------------------------|
| Host                                    | Status | Last Check          | Duration      | Status Information                        |
| Innserver                               | UP     | 09-27-2024 12:24:39 | 0d 0h 0m 32s+ | PING OK - Packet loss = 0%, RTA = 0.03 ms |
| Innhost                                 | UP     | 09-27-2024 12:07:28 | 0d 0h 42m 5s  | PING OK - Packet loss = 0%, RTA = 0.03 ms |

**Conclusion:** Thus we successfully performed port monitoring of a Linux server using Nagios. Utilizing Nagios for comprehensive monitoring of ports, services, and Windows/Linux servers enhances system reliability, improves performance, and ensures proactive management of IT infrastructure, ultimately driving operational efficiency.

## Experiment 11

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

1. Open up the Lambda Console and click on the Create button.

The screenshot shows the AWS Lambda Functions page. On the left, there's a sidebar with links like Dashboard, Applications, Functions, Additional resources, and Related AWS resources. The main area is titled 'Functions (5)' and shows a table with columns: Function name, Description, Package type, Runtime, and Last modified. The three visible functions are:

- MainMonitoringFunction: Zip, Python 3.8, 2 months ago. Description: updates LabRole to allow it to assume itself.
- ModLabRole: Zip, Python 3.8, 2 months ago. Description: Create Redshift event subscription to SNS Topic.
- RedshiftEventSubscription: Zip, Python 3.8, 2 months ago. Description: Create Redshift event subscription to SNS Topic.

2. Choose to create a function from scratch or use a blueprint, i.e templates defined by AWS for you with all configuration presets required for the most common use cases. After that, choose to create a new role with basic Lambda permissions if you don't have an existing one.

The screenshot shows the 'Create function' wizard. It starts with a choice between 'Author from scratch', 'Use a blueprint', and 'Container image'. The 'Author from scratch' option is selected. Below this, there's a 'Basic information' section where the 'Function name' is set to 'adityaraorane'. The 'Runtime' dropdown is set to 'Node.js 20.x'. At the bottom, there's a large blue 'Create' button.

Click on the *Create* button.

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

The screenshot shows the AWS Lambda console. At the top, there's a green success message: "Successfully created the function adityaraorane. You can now change its code and configuration. To invoke your function with a test event, choose 'Test'." Below this, the function name "adityaraorane" is displayed. On the left, there are tabs for "Function overview" and "Info". On the right, there are buttons for "Throttle", "Copy ARN", and "Actions". Under "Function overview", there are tabs for "Diagram" (selected) and "Template". The diagram shows a single function node labeled "adityaraorane" with a "Layers" section below it. There are buttons for "+ Add trigger" and "+ Add destination". On the far right, there are sections for "Description", "Last modified" (14 seconds ago), "Function ARN" (arn:aws:lambda:us-east-1:090530457694:function:adityaraorane), and "Function URL" (Info). A note says "Activate Windows Go to Settings to activate Windows".

The screenshot shows the AWS Lambda Code Source editor for the "adityaraorane" function. At the top, there's a green success message: "Successfully created the function adityaraorane. You can now change its code and configuration. To invoke your function with a test event, choose 'Test'." Below this, the "Code" tab is selected. The interface includes a toolbar with File, Edit, Find, View, Go, Tools, Window, a "Test" dropdown, and a "Deploy" button. On the left, there's a sidebar for "Environment" with a tree view showing "adityaraorane - /" and "index.mjs". The main area shows the code editor with "index.mjs" selected. The code is:

```
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 };
```

A "Upload from" button is located at the top right of the code editor. A note at the bottom right says "Activate Windows Go to Settings to activate Windows".

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

The screenshot shows the AWS Lambda console. The top navigation bar includes the AWS logo, Services, a search bar, and account information: N. Virginia, user3396554=RAORANE\_ADITYA\_NAGESH @ 0905-3045-7694. A green success message at the top states: "Successfully created the function adityaraorane. You can now change its code and configuration. To invoke your function with a test event, choose 'Test'." Below the message, the tabs are Code, Test, Monitor, Configuration (which is selected and highlighted in blue), Aliases, and Versions. On the left, a sidebar lists General configuration, Triggers, Permissions, Destinations, Function URL, Environment variables, Tags, VPC, and RDS databases. The main content area displays the General configuration settings:

| General configuration <a href="#">Info</a> |                                |                   |
|--------------------------------------------|--------------------------------|-------------------|
| Description                                | Memory                         | Ephemeral storage |
| -                                          | 128 MB                         | 512 MB            |
| Timeout                                    | SnapStart <a href="#">Info</a> | None              |
| 0 min 3 sec                                |                                |                   |

An "Edit" button is located in the top right corner of the configuration table. At the bottom of the page, there is a note: "Here, you can enter a description and change Memory and Timeout. I've changed the Timeout period to 5 sec since that is sufficient for now."

The screenshot shows the AWS Lambda console. The top navigation bar includes the AWS logo, Services, a search bar, and account information: N. Virginia, user3396554=RAORANE\_ADITYA\_NAGESH @ 0905-3045-7694. The URL in the address bar is Lambda > Functions > adityaraorane > Edit basic settings. The page title is "Edit basic settings". The main content area displays the Basic settings section:

| Basic settings <a href="#">Info</a>                                                                                                                                                                                                                                                                                                                                                                                                   |  |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| Description - optional<br><input type="text" value="This is my experiment 11 of advdevops lab"/>                                                                                                                                                                                                                                                                                                                                      |  |
| Memory <a href="#">Info</a><br>Your function is allocated CPU proportional to the memory configured.<br><input type="text" value="128"/> MB                                                                                                                                                                                                                                                                                           |  |
| Set memory to between 128 MB and 10240 MB.                                                                                                                                                                                                                                                                                                                                                                                            |  |
| Ephemeral storage <a href="#">Info</a><br>You can configure up to 10 GB of ephemeral storage (/tmp) for your function. <a href="#">View pricing</a><br><input type="text" value="512"/> MB                                                                                                                                                                                                                                            |  |
| Set ephemeral storage (/tmp) to between 512 MB and 10240 MB.                                                                                                                                                                                                                                                                                                                                                                          |  |
| SnapStart <a href="#">Info</a><br>Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized. To evaluate whether your function is a good candidate for SnapStart, run a test event on your function and check the CloudWatch Metrics for the function's execution. If the metrics show a significant reduction in startup time, then your function is a good candidate for SnapStart. |  |

At the bottom of the page, there is a note: "Activate Windows Go to Settings to activate Windows".

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

None

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

Timeout  
0 min 5 sec

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

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

View the LabRole role [on the IAM console](#).

Cancel [Save](#)

Activate Windows  
Go to Settings to activate Windows.

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

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

Successfully updated the function adityaraorane.

Code source [Info](#)

File Edit Find View Go Tools Window Test Deploy Changes not deployed

index.mjs

```
1 export const handler = async (event) => {
2     // TODO implement
3     const response = {
4         statusCode: 200,
5         body: JSON.stringify('Aditya Raorane welcomes you to his lambda function!'),
6     };
7     return response;
8 };
9
```

Upload from

Activate Windows  
Go to Settings to activate Windows.

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

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

The screenshot shows the AWS Lambda function configuration interface. At the top, a green banner indicates "Successfully updated the function adityaraorane.". Below this, the "Code source" tab is selected. The "Test" button is highlighted in blue. The "Execution results" section shows a successful execution with status "Succeeded", max memory used: 62 MB, and time: 7.78 ms. The "Response" field contains the JSON output: {"statusCode": 200, "body": "\"Aditya Raorane welcomes you to his lambda function!\""}.

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

The screenshot shows the AWS Lambda function configuration interface. A modal dialog titled "Configure test event" is open in the foreground. The "Test" button in the main navigation bar is also highlighted in blue. The "Execution results" section in the background shows a successful execution with status "Succeeded", max memory used: 62 MB, and time: 7.78 ms. The "Response" field contains the JSON output: {"statusCode": 200, "body": "\"Aditya Raorane welcomes you to his lambda function!\""}.

The screenshot shows the AWS Lambda console interface. At the top, there are options for 'Event sharing settings': 'Private' (selected) and 'Shareable'. Below this, a 'Template - optional' dropdown is set to 'hello-world'. The main area displays 'Event JSON' with the following code:

```

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

```

Below the JSON editor are three buttons: 'Cancel', 'Invoke' (highlighted in orange), and 'Save'. A success message at the top of the page states: 'The test event adityaraorane was successfully saved.' The Lambda function details show the 'Code source' tab selected, with 'index.mjs' as the file. The 'Execution result' section shows the response: 'statusCode: 200, body: "\'Aditya Raorane welcomes you to his lambda function!\'"' and logs indicating a successful execution.

**Conclusion:** We successfully executed the AWS Lambda function using Python. We learned how AWS Lambda allows you to run code without provisioning or managing servers, making it a cost-effective and scalable solution for serverless computing. The workflow involves uploading code, setting triggers, and letting Lambda manage the execution based on the provided conditions. AWS Lambda's event-driven architecture makes it highly efficient for handling real-time data processing and automation tasks.

## Experiment 12

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

1] Create a AWS S3 Bucket type **General Purpose** named **adityaraorane**.

The screenshot shows the AWS S3 'Create bucket' configuration page. The 'General configuration' section is visible. Under 'Bucket type', the 'General purpose' option is selected, indicated by a blue outline. The 'Bucket name' field contains 'adityaraorane'. A note at the bottom states: 'Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming.' The top navigation bar shows 'Amazon S3 > Buckets > Create bucket'.

2] Turn off the **Block Public Access** for this bucket so that it is publicly accessible to all users.

The screenshot shows the 'Block Public Access settings for this bucket' page. The 'Block all public access' checkbox is unselected. A note at the bottom states: 'Turning off block all public access might result in this bucket and the objects within becoming publicly accessible to all users.' The top navigation bar shows 'Amazon S3 > Buckets > Block Public Access'.

3] Click on the **Create** button. This will successfully create our bucket. Then click on the bucket name **adityaraorane** to open it up.

The screenshot shows the AWS S3 console. At the top, a green banner indicates that a bucket named "adityaraorane" has been successfully created. Below this, the "Buckets" page is displayed. A summary box shows an account snapshot updated every 24 hours. The main table lists "General purpose buckets" with one entry: "adityaraorane". The table includes columns for Name, AWS Region, IAM Access Analyzer, and Creation date. The "adityaraorane" entry shows "us-east-1" as the region, "View analyzer for us-east-1" for IAM, and "September 26, 2024, 12:53:12 (UTC+05:30)" for creation date. Action buttons for Copy ARN, Empty, Delete, and Create bucket are visible at the top of the table. Navigation links for "Amazon S3 > Buckets" and "View Storage Lens dashboard" are also present.

4] Now create a Lambda function named **adityaraorane**.

The screenshot shows the AWS Lambda console. A green banner at the top states that a function named "adityaraorane" has been successfully created. The main page displays the "adityaraorane" function under the "Functions" section. The "Function overview" tab is selected, showing a diagram of the function. The function name "adityaraorane" is highlighted. Other tabs include "Template" and "+ Add trigger". On the right side, detailed information is provided: "Description" (empty), "Last modified" (14 seconds ago), "Function ARN" (arn:aws:lambda:us-east-1:090530457694:function:adityaraorane), and "Function URL" (Info). A link to "Activate Windows" is also present. Action buttons for Throttle, Copy ARN, and Actions are located at the top right. Navigation links for "Lambda > Functions > adityaraorane" and "CloudShell Feedback" are at the bottom.

5] Click on **Services** and select **CloudWatch** which is an AWS service that helps you enable operational and risk auditing, governance, and compliance of your AWS account.

The screenshot shows the AWS CloudWatch Events console. At the top, there is a search bar and a navigation bar with tabs like Services, CloudShell, and Feedback. The main area displays a table for 'Event notifications (0)'. The table has columns for Name, Event types, Filters, Destination type, and Destination. A button 'Create event notification' is visible. Below the table, there is a section for 'Amazon EventBridge' with a sub-section for 'Amazon S3 event notifications'. It shows a message: 'No event notifications. Choose Create event notification to be notified when a specific event occurs.' A 'Create event notification' button is also present here. At the bottom right, there are links to 'Activate Windows' and 'Go to Settings to activate Windows.'

6] Click on **Create event notification** and select **Destination** as **Lambda function** and choose the lambda function created with the name **adityaraorane**.

The screenshot shows the AWS Lambda function creation interface. The top navigation bar includes tabs for Services, CloudShell, and Feedback. The main area is titled 'Destination' and contains a note: 'Before Amazon S3 can publish messages to a destination, you must grant the Amazon S3 principal the necessary permissions to call the relevant API to publish messages to an SNS topic, an SQS queue, or a Lambda function.' Below this, there are several options: 'Lambda function' (selected), 'SNS topic', 'SQS queue', and 'Specify Lambda function'. Under 'Specify Lambda function', there is a dropdown menu with the value 'adityaraorane'. At the bottom, there are 'Cancel' and 'Save changes' buttons. A 'Save changes' button is highlighted in orange. At the very bottom, there are links to 'Activate Windows' and 'Go to Settings to activate Windows.'

## 7] Click on Save Changes.

**Bucket overview**

AWS Region: US East (N. Virginia) us-east-1 | Amazon Resource Name (ARN): arn:aws:s3:::adityaraorane | Creation date: September 26, 2024, 12:53:12 (UTC+05:30)

**Bucket Versioning** Edit

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning: Disabled

Multi-factor authentication (MFA) delete: An additional layer of security that requires multi-factor authentication for changing Bucket Versioning settings and permanently deleting object versions. To modify MFA delete settings, use the AWS CLI, AWS SDK, or the Amazon S3 REST API. [Learn more](#)

## 8] Deploy the code and Test it to check whether the lambda function executed successfully.

**Code source** Info

**Test** Deploy

File Edit Find View Go Tools Window

Execution results Environment Var Execution result

Test Event Name: adityaraorane

Response:

```
{
  "statusCode": 200,
  "body": "\\"Lambda" function executed successfully!\\""
}
```

Function Logs

START RequestId: fa5d82d6-c569-4488-81f8-a4b2b3a9319c Version: \$LATEST
2024-09-26T07:11:13.926Z 95d42926-c569-4488-81f8-a4b2b3a9319c INFO An image has been added
END RequestId: fa5d82d6-c569-4488-81f8-a4b2b3a9319c
REPORT RequestId: fa5d82d6-c569-4488-81f8-a4b2b3a9319c Duration: 33.99 ms Billed Duration: 34 ms Memory Size: 128 MB Max Memory Used: 64 MB Init Duration: 138.48 ms

Request ID: fa5d82d6-c569-4488-81f8-a4b2b3a9319c

9] Upload a .jpg image on the S3 bucket created and click on **Upload**.

The screenshot shows the AWS S3 'Upload' interface. At the top, there's a search bar and navigation links for 'Amazon S3 > Buckets > adityaraorane > Upload'. Below this, a large blue dashed box contains the instruction 'Drag and drop files and folders you want to upload here, or choose Add files or Add folder.' To the right of this box is a table titled 'Files and folders (0)' with three buttons: 'Remove', 'Add files', and 'Add folder'. A search bar labeled 'Find by name' is positioned above the table. The table has columns for 'Name' and 'Folder'. A message at the bottom of the table says 'No files or folders' and 'You have not chosen any files or folders to upload.'

This screenshot shows the same AWS S3 'Upload' interface after a file has been added. The 'Files and folders' table now shows one item: 'adityaraorane.jpg' (1 Total, 15.6 KB). The rest of the interface remains the same, including the upload instructions and the 'Destination' section.

The screenshot shows the AWS S3 console with the destination configuration page. The destination is set to `s3://adityaraorane`. There are sections for Destination details (Bucket settings), Permissions (Grant public access and access to other AWS accounts), and Properties (Specify storage class, encryption settings, tags, and more). At the bottom right, there are **Cancel** and **Upload** buttons.

The screenshot shows the AWS S3 console with the upload status summary. It indicates **Upload succeeded** with 1 file uploaded (15.6 KB, 100.00%). The summary table includes columns for Destination, Succeeded (1 file, 15.6 KB), and Failed (0 files, 0 B). Below the summary, there are tabs for **Files and folders** and **Configuration**. At the bottom right, there is a **Close** button and a message to activate Windows.

The screenshot shows the AWS S3 console interface. At the top, there's a green success message: "Upload succeeded" with a link to "View details below." Below this, the "Summary" section shows the destination "s3://adityaraorane" and two status boxes: "Succeeded" (1 file, 15.6 KB (100.00%)) and "Failed" (0 files, 0 B (0%)). Underneath, there are tabs for "Files and folders" and "Configuration". The "Files and folders" tab is selected, displaying a table with one item: "adityaraoran..." (image/jpeg, 15.6 KB, Succeeded). A search bar and navigation icons are also present.

10] In the CloudWatch services ,in the left navigation pane select the Log groups. Our lambda function will be listed here. ([/aws/lambda/adityaraorane](#)). Click on it.

The screenshot shows the AWS CloudWatch Logs service. The left sidebar has sections for CloudWatch, Favorites and recents, Dashboards, Alarms, Logs (with Log groups selected), Metrics, X-Ray traces, Events, and Application Signals. The main content area shows the "Log groups" page with 4 entries: "/aws/lambda/RedshiftEventSubscription", "/aws/lambda/RedshiftOverwatch", "/aws/lambda/RoleCreationFunction", and "/aws/lambda/adityaraorane". Each entry includes columns for Log group, Log class, Anomaly detection, Data retention, and Retention. A "Create log group" button is at the top right. A search bar and filter options are also present. A watermark for "Activate Windows" is visible.

The screenshot shows the AWS CloudWatch Logs interface. The left sidebar is collapsed, showing sections like 'Logs' (Log groups, Log Anomalies, Live Tail, Logs Insights, Contributor Insights), 'Metrics', 'X-Ray traces', 'Events', and 'Application Signals'. The main area displays the '/aws/lambda/adityaraorane' log group details. It includes tabs for 'Actions', 'View in Logs Insights', 'Start tailing', and 'Search log group'. Below these are tabs for 'Log streams', 'Tags', 'Anomaly detection', 'Metric filters', 'Subscription filters', 'Contributor Insights', and 'Data protection'. The 'Log streams' tab is selected, showing a list of 2 entries:

| Log stream                                                            | Last event time           |
|-----------------------------------------------------------------------|---------------------------|
| <a href="#">2024/09/26/[\$LATEST]832a31bbb6bc4b469b164515de1733fa</a> | 2024-09-26 07:34:13 (UTC) |
| <a href="#">2024/09/26/[\$LATEST]41b27e8af31141eaafa1594271a35f86</a> | 2024-09-26 07:19:59 (UTC) |

At the bottom right of the main area, there is a note: 'Activate Windows' and 'Go to Settings to activate Windows.'

**Conclusion:** Integrating AWS Lambda with S3 allows for real-time, automated processing of events such as file uploads. In this example, a Lambda function is configured to log a message whenever an image is added to a specific S3 bucket. This setup demonstrates the power and flexibility of serverless computing by automating tasks without requiring manual intervention or server management. By leveraging AWS Lambda, developers can efficiently handle event-driven workflows, reduce operational overhead, and quickly deploy scalable solutions that respond to specific actions within cloud environments.