

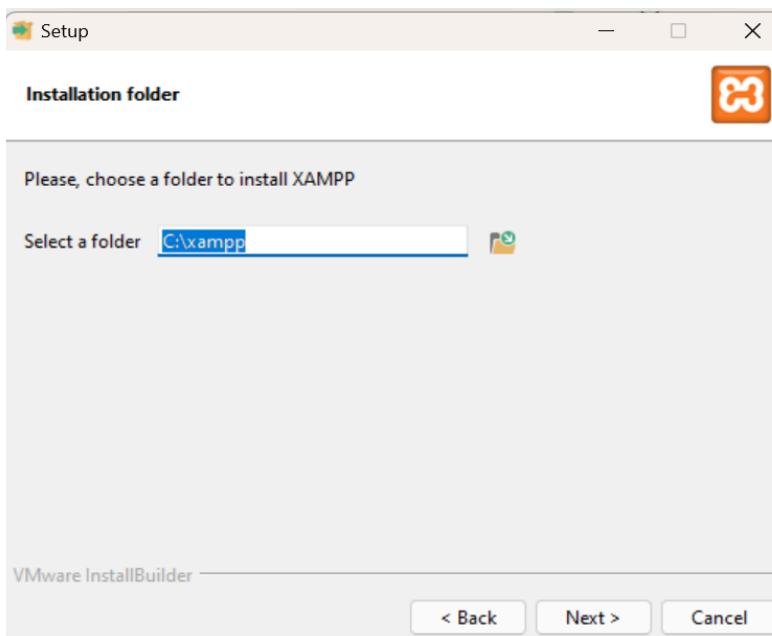
Experiment 1A

Hosting a static site using xampp

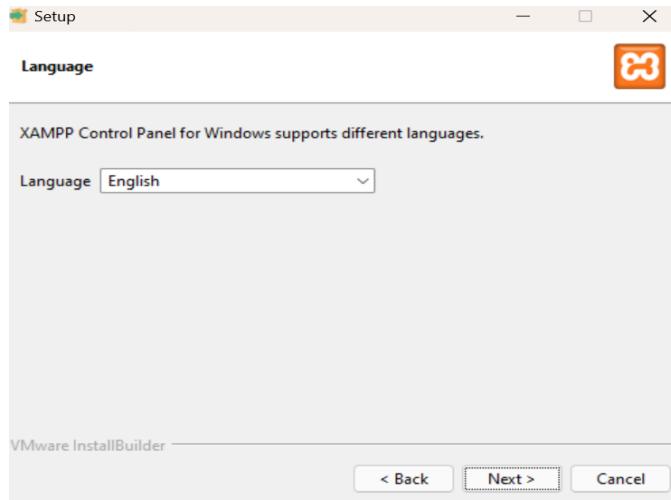
1. Download and open the xampp application and click on zip and extract it then click it.



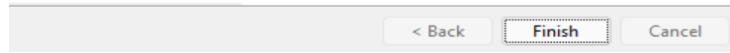
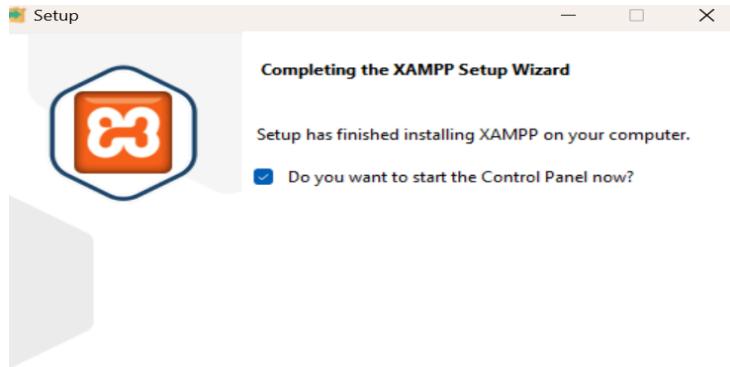
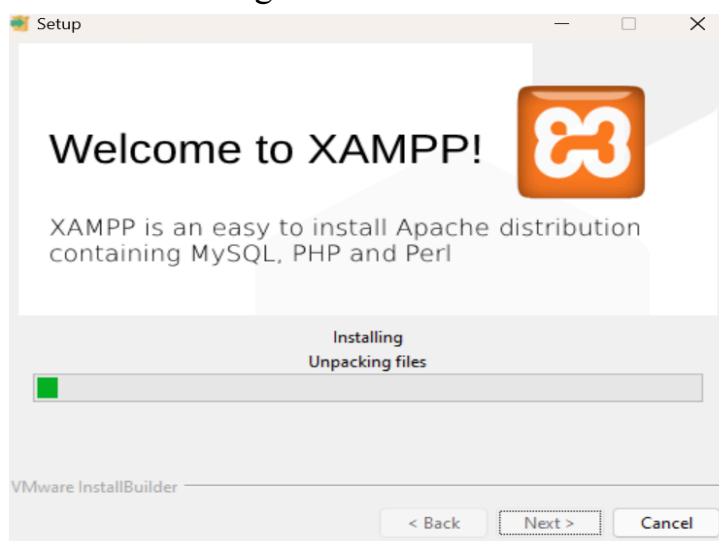
2. Then select the c directory.



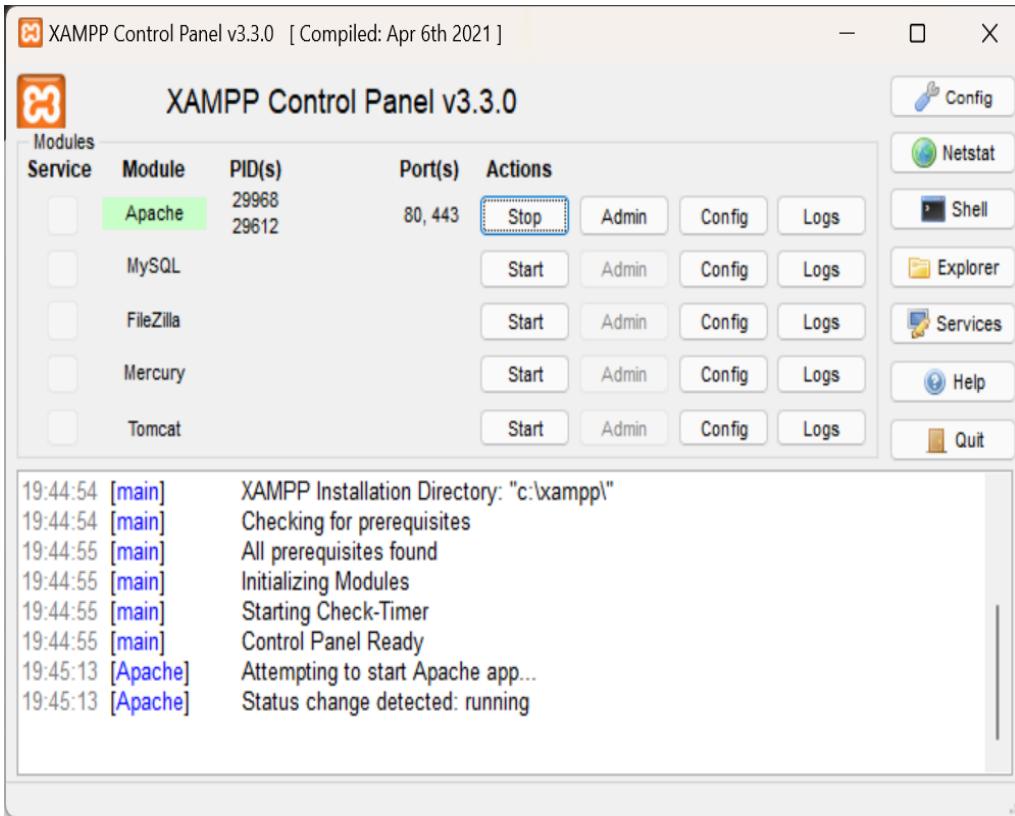
3. Then select language as english.



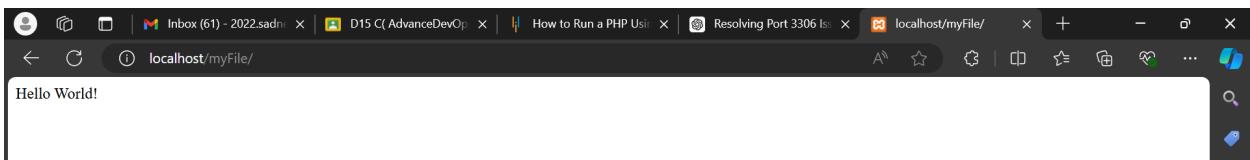
Then install and get started.



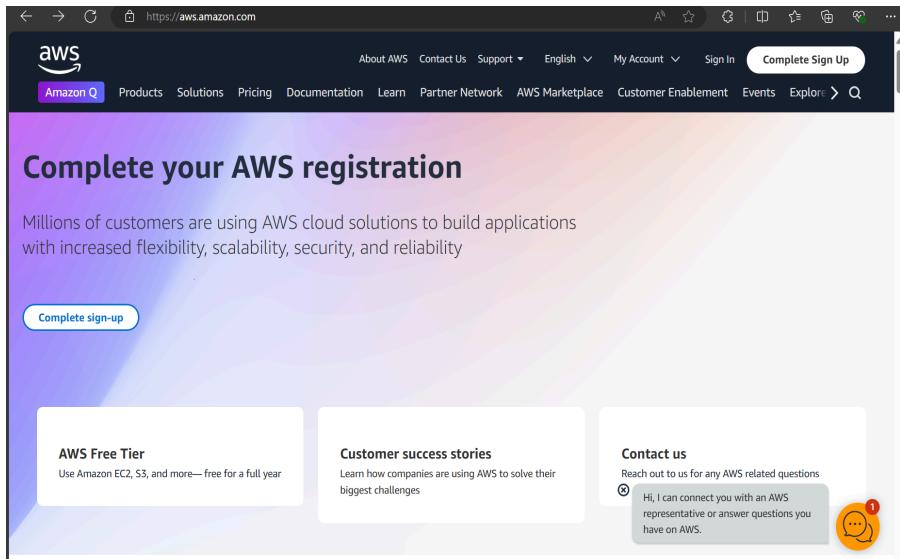
Then the final control panel opens.



Then I created a file myFile inside htdocs folder and uploaded a sample file.and then opened <https://localhost/myFile/in.php>



Hosting a static site using S3 bucket:

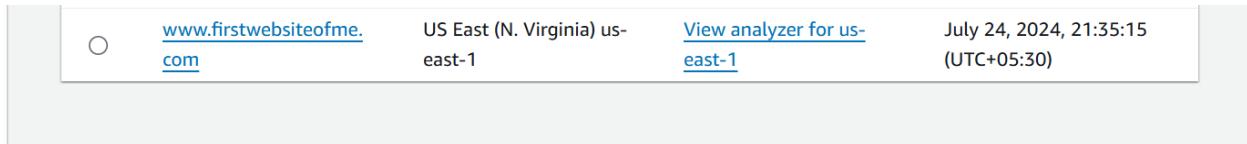


1. I clicked on create bucket.

A screenshot of a web browser showing the "Amazon S3" get-started page for the "eu-north-1" region at https://eu-north-1.console.aws.amazon.com/s3/get-started?region=eu-north-1. The main content area features the heading "Amazon S3" and the sub-headline "Store and retrieve any amount of data from anywhere". Below this, there is a section titled "How it works" with a video thumbnail titled "Introduction to Amazon S3" and a "Copy link" button. To the right, there is a "Create a bucket" box with the text: "Every object in S3 is stored in a bucket. To upload files and folders to S3, you'll need to create a bucket where the objects will be stored." and a large orange "Create bucket" button. On the left, there is a sidebar with links like "Storage", "CloudShell", and "Feedback". At the bottom, there is a footer with links for "Pricing", "AWS Simple Monthly Calculator", and "Cookie preferences".

A screenshot of a web browser showing the "Amazon S3" get-started page for the "us-east-1" region at https://us-east-1.console.aws.amazon.com/s3/get-started?region=us-east-1. The layout is identical to the EU version, featuring the "Amazon S3" heading, "Store and retrieve any amount of data from anywhere" sub-headline, "How it works" section with a video thumbnail, and the "Create a bucket" box. The sidebar on the left includes "Buckets", "Access Grants", "Access Points", "Object Lambda Access Points", "Multi-Region Access Points", "Batch Operations", and "IAM Access Analyzer for S3". At the bottom, there is a footer with links for "Storage", "CloudShell", and "Feedback".

3. Then I created a bucket named www.firstwebsiteofme.com



I clicked on the bucket.

The screenshot shows the AWS S3 console for the 'www.firstwebsiteofme.com' bucket. The 'Objects' tab is selected, showing a single object named 'index.html' which is an HTML file (Type: html, Size: 62.0 B, Storage class: Standard). The 'Actions' bar includes buttons for Copy S3 URI, Copy URL, Download, Open, and Delete, with 'Upload' being the active button. The left sidebar shows various AWS services like IAM Access Analyzer and Storage Lens.

4. Then I also S3 webhosting-bucket policy with this code .I entered in the permission section there is an option edit bucket policy. Where I uploaded the code and saved it.

The screenshot shows the 'Edit bucket policy' page. The policy is defined in JSON:

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

The right side of the screen has a 'Select a statement' dropdown and a '+ Add new statement' button.

4. Then i clicked on index.html file and opened its properties section.

The screenshot shows the AWS S3 console interface. On the left, there's a sidebar with various services like Buckets, Access Grants, and Storage Lens. The main area shows the object details for 'index.html' in the 'www.firstwebsiteofme.com' bucket. The 'Properties' tab is selected, displaying information such as Owner (2022.sadneya.samant), AWS Region (US East (N. Virginia) us-east-1), Last modified (July 24, 2024, 21:35:41 (UTC+05:30)), Size (62.0 B), Type (html), and Key (index.html). It also shows the S3 URI (s3://www.firstwebsiteofme.com/index.html), Amazon Resource Name (ARN) (arn:aws:s3:::www.firstwebsiteofme.com/index.html), Entity tag (Etag) (18f77726d0973067f7329656daaccd8), and Object URL (<https://s3.amazonaws.com/www.firstwebsiteofme.com/index.html>). At the bottom, there are links for CloudShell, Feedback, and a footer with copyright information.

4. Then I clicked on

<https://s3.amazonaws.com/www.firstwebsiteofme.com/index.html> which given me the final output.

The screenshot shows a web browser window with the URL <https://s3.amazonaws.com/www.firstwebsiteofme.com/index.html>. The page content is "Hello World!" and below it, a small note says "This is just a sample page". The browser interface includes a back button, forward button, search bar, and a vertical toolbar on the right.

Experiment 1B

Creation of Cloud9 Environment

1. Go to the amazon console page.

The screenshot shows the AWS Management Console search results for 'cloud 9'. The search bar at the top contains 'cloud 9'. On the left, there's a sidebar for 'Amazon S3' with options like Buckets, Access Grants, and Object Lambda Access. The main search results are under 'Services' (58) and 'Features' (89). Under 'Services', the first result is 'Cloud9' with a star icon, described as 'A Cloud IDE for Writing, Running, and Debugging Code'. Other services listed include Amazon CodeCatalyst, AWS Cloud Map, and AWS Deadline Cloud. Under 'Features', the first result is 'Cloud WAN' with a star icon, described as 'VPC feature'. Other features listed include Namespaces and AWS Cloud Map feature.

2. click on create environment.

The screenshot shows the AWS Cloud9 control panel. The title is 'AWS Cloud9' with the subtitle 'A cloud IDE for writing, running, and debugging code'. Below the title, there's a paragraph about AWS Cloud9 allowing you to write, run, and debug your code with just a browser. To the right, there's a callout box titled 'New AWS Cloud9 environment' with a large orange 'Create environment' button. At the bottom, there are two sections: 'How it works' and 'Getting started'. The 'How it works' section has a sub-section about creating an AWS Cloud9 development environment on a new Amazon EC2 instance. The 'Getting started' section has links to 'Before you start' and 'Create an environment'.

3. Give the name to your environment. Here I have given the name as sadneya_46. keep the remaining things by default.

The screenshot shows the 'Create environment' wizard in the AWS Cloud9 interface. The 'Details' step is selected. The 'Name' field contains 'sadneya_46'. The 'Environment type' section shows 'New EC2 instance' selected, with a note that Cloud9 creates an EC2 instance in your account. There are two options: 'New EC2 instance' (selected) and 'Existing compute'. The 'New EC2 instance' section includes a sub-section for 'Instance type' with 't2.micro' selected. Other options include 't3.small' and 'm5.large'. It also includes sections for 'Platform' (Amazon Linux 2023), 'Timeout' (30 minutes), and 'Additional instance types' (which is currently empty). The bottom navigation bar shows 'New EC2 instance' is the active step.

Create environment Info

Details

Name
sadneya_46

Description - *optional*

Environment type Info

Determines what the Cloud9 IDE will run on.

New EC2 instance
Cloud9 creates an EC2 instance in your account. The configuration of your EC2 instance cannot be changed by Cloud9 after creation.

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

New EC2 instance

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

New EC2 instance

Instance type Info

The memory and CPU of the EC2 instance that will be created for Cloud9 to run on.

t2.micro (1 GiB RAM + 1 vCPU)
Free-tier eligible. Ideal for educational users and exploration.

t3.small (2 GiB RAM + 2 vCPU)
Recommended for small web projects.

m5.large (8 GiB RAM + 2 vCPU)
Recommended for production and most general-purpose development.

Additional instance types
Explore additional instances to fit your need.

Platform Info

This will be installed on your EC2 instance. We recommend Amazon Linux 2023.

Amazon Linux 2023

Timeout

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

30 minutes

The screenshot shows the AWS Cloud9 configuration interface. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, 'Search' bar, and account information ('N. Virginia' and 'voclabs/user3404102=SAMANT_SADNEYA_SADANAND @ 4250-0137-5268').

Network settings

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

Tags - optional: 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.

No tags associated with the resource.

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

The following IAM resources will be created in your account

VPC settings

Amazon Virtual Private Cloud (VPC): The VPC that your environment will access. To allow the AWS Cloud9 environment to connect to its EC2 instance, attach an internet gateway (IGW) to your VPC. [Create new VPC](#).

vpc-051bba342b3626898
Name -

Subnet: Used to setup your VPC configuration. To use a private subnet, select AWS Systems Manager (SSM) as the connection type. [Create new subnet](#).

No preference
Uses default subnet in any Availability Zone

Tags - optional: 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.

No tags associated with the resource.

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

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](#)

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

Amazon Virtual Private Cloud (VPC)
The VPC that your environment will access. To allow the AWS Cloud9 environment to connect to its EC2 instance, attach an internet gateway (IGW) to your VPC. [Create new VPC](#)

vpc-051bba342b3626898
Name -

Subnet
Used to setup your VPC configuration. To use a private subnet, select AWS Systems Manager (SSM) as the connection type. [Create new subnet](#)

No preference
Uses default subnet in any Availability Zone

Thus, the environment was created.

AWS Cloud9

Creating sadneya_46. This can take several minutes. While you wait, see [Best practices for using AWS Cloud9](#)

For capabilities similar to AWS Cloud9, explore AWS Toolkits in your own IDE and AWS CloudShell in the AWS Management Console. [Learn more](#)

AWS Cloud9 > Environments

Name	Cloud9 IDE	Environment type	Connection	Permission	Owner ARN
sadneya_46	Open	EC2 instance	Secure Shell (SSH)	Owner	arn:aws:sts::42500137526 role/voclabs/user3404102=SAMANT_SADNEYA_SADANAND@4250-0137-5268

Creation of I am user

1. Go to I am.

2. Select Users here.
3. Then Click on create user.

The screenshot shows the AWS Identity and Access Management (IAM) service interface. The left sidebar is titled 'Identity and Access Management (IAM)' and contains the following navigation items:

- Dashboard
- Access management
 - User groups
 - Users** (selected)
 - Roles
 - Policies
 - Identity providers
 - Account settings
- Access reports
 - Access Analyzer
 - External access
 - Unused access
 - Analyzer settings

The main content area is titled 'Users (0) Info' and displays the message: 'An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.' A search bar and a table header ('User name') are visible, but the table body shows 'No resources to display'.

4. Give I AM user name. Here I have given it as sadneya_46. Select custom password and enter your password.

This screenshot shows the 'Specify user details' step of the IAM User creation wizard. On the left, a vertical navigation bar lists four steps: Step 1 (Specify user details, selected), Step 2 (Set permissions), Step 3 (Review and create), and Step 4 (Retrieve password).

The main form is titled 'Specify user details' and contains the following fields:

- User details** section:
 - User name:** sadneya_46
 - Provide user access to the AWS Management Console - optional:** A checked checkbox with a note: 'If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center.'
- Console password** section:
 - Autogenerated password:** An unchecked radio button with a note: 'You can view the password after you create the user.'
 - Custom password:** A checked radio button with a note: 'Enter a custom password for the user.' Below this is a password input field containing '*****'.
 - Show password:** An unchecked checkbox.
- Users must create a new password at next sign-in - Recommended:** A checked checkbox with a note: 'Users automatically get the IAMUserChangePassword policy to allow them to change their own password.'
- Generate programmatic access:** A note: '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.' It includes a 'Learn more' link.

At the bottom right are 'Cancel' and 'Next' buttons.

5. Select Add user to group. And click next.

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](#)

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.

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

Set permissions boundary - optional

Cancel Previous Next

6. Also select permissions. Here I have selected AWSCloud9EnvironmentMember for cloud9 Environment. then click on next.

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](#)

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.

Permissions policies (1/1227)

Choose one or more policies to attach to your new user.

Policy name	Type	Attached entities
AWSCloud9Administrator	AWS managed	0
AWSCloud9EnvironmentMember	AWS managed	0
AWSCloud9ServiceRolePolicy	AWS managed	1
AWSCloud9SSMInstanceProfile	AWS managed	0
AWSCloud9User	AWS managed	0

To add user, we need to create

7. To create a user group, give it a name as WebAppUser.

Create user group

User group name
Enter a meaningful name to identify this group.
WebAppUser
Maximum 128 characters. Use alphanumeric and '+,-,_-' characters.

Permissions policies (1/951)

Policy name	Type	Description
<input checked="" type="checkbox"/> AdministratorAccess	AWS managed	Provides full access to AWS services
<input type="checkbox"/> AdministratorAcc...	AWS managed	Grants account administrative perm
<input type="checkbox"/> AdministratorAcce...	AWS managed	Grants account administrative perm
<input type="checkbox"/> AlexaForBusinessD...	AWS managed	Provide device setup access to Alex
<input type="checkbox"/> AlexaForBusinessF...	AWS managed	Grants full access to AlexaForBusin
<input type="checkbox"/> AlexaForBusinessG...	AWS managed	Provide gateway execution access t

Cancel Create user group

8. Thus, WebAppUser get created successfully.

WebAppUsers user group created.

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.

User groups (1)

Group name	Users	Attached policies	Created
WebAppUsers	0	-	2024-08-02 (Now)

Set permissions boundary - optional

Cancel Previous Next

The screenshot shows the AWS IAM User creation interface. The user has just completed Step 3: Review and create. The main panel displays the User details and Permissions summary. The User details section shows the User name as "sadneya_46", Console password type as "Custom password", and Require password reset as "Yes". The Permissions summary table lists a single permission: "IAMUserChangePassword" (AWS managed, Permissions policy). Below this, there is a section for Tags - optional, which is currently empty. At the bottom right, there are "Cancel", "Previous", and "Create user" buttons.

The screenshot shows the AWS IAM User Groups page. The sidebar navigation is under the "Identity and Access Management (IAM)" section, specifically under "Access management" and "User groups". The main content area shows a table titled "User groups (1) Info". The table has one entry: "WebAppUsers". The row for "WebAppUsers" shows 0 users and "Not defined" for permissions. It was created 4 minutes ago. The table includes columns for Group name, Users, Permissions, and Creation time. There are "Search", "Delete", and "Create group" buttons at the top of the table area.

The screenshot shows the AWS IAM console. On the left, the navigation pane is open with 'Identity and Access Management (IAM)' selected. Under 'Access management', 'User groups' is selected, showing 'WebAppUsers'. The main content area displays the 'WebAppUsers' group details. The 'Summary' section shows the user group name 'WebAppUsers', creation time 'August 02, 2024, 19:25 (UTC+05:30)', and ARN 'arn:aws:iam::851725480355:group/WebAppUsers'. Below this, there are tabs for 'Users', 'Permissions', and 'Access Advisor'. The 'Users' tab is active, showing 'Users in this group (0)'. A search bar and a table header with columns 'User name', 'Groups', 'Last activity', and 'Creation time' are visible. The message 'No resources to display' is shown below the table.

9. Then click on next and thus the user gets successfully created.

The screenshot shows the AWS IAM 'Create user' process at Step 4: 'Retrieve password'. A green success banner at the top states 'User created successfully' and provides instructions to view and download the user's password or email sign-in instructions. The main content area shows the 'Console sign-in details' section, which includes the 'Console sign-in URL' (https://851725480355.signin.aws.amazon.com/console), 'User name' ('sadneya_46'), and 'Console password' (redacted). Buttons for 'Cancel', 'Download .csv file', and 'Return to users list' are at the bottom.

10. This is a summary of user created.

The screenshot shows the AWS Identity and Access Management (IAM) service interface. On the left, a sidebar menu is open under the 'Identity and Access Management (IAM)' section, showing various navigation options like Dashboard, Access management, Access reports, and Service control policies. The main content area displays the details for a user named 'sadneya_46'. The 'Summary' tab is selected, showing the ARN (arn:aws:iam::851725480355:user/sadneya_46), which is highlighted with a yellow warning icon indicating it is enabled without MFA. Other summary details include the creation date (August 02, 2024, 19:28 (UTC+05:30)) and the last console sign-in (Never). Below the summary, the 'Security credentials' tab is selected, showing one access key (Access key 1) with a 'Create access key' link. The 'Permissions policies (1)' section lists a single policy named 'IAMUserChangePassword', which is an AWS managed policy attached directly to the user. A search bar and filter options are also present in this section.

Experiment No: 2

Step1:- Creation of role:-

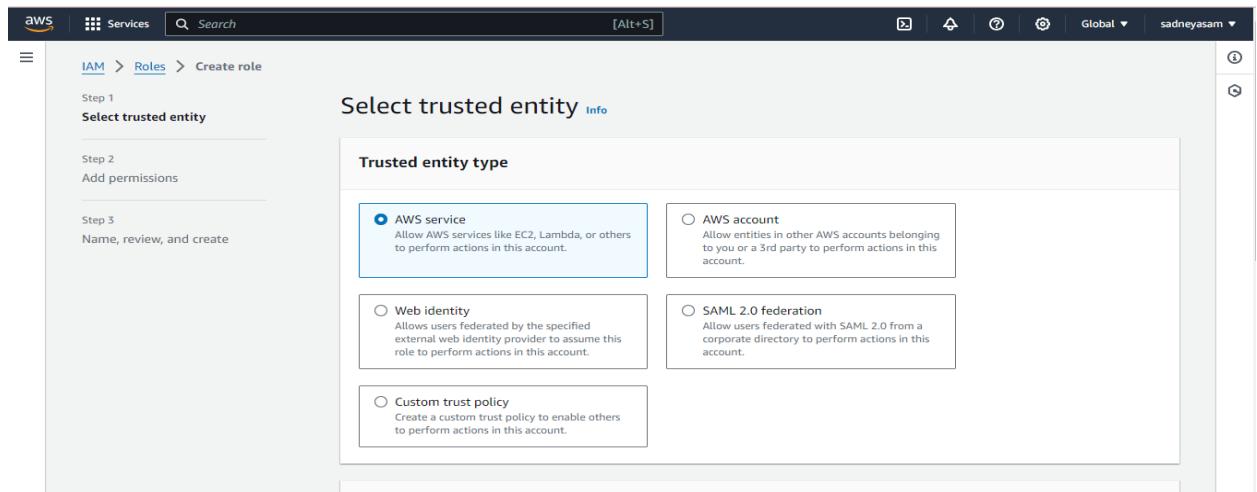
1. Login to your AWS account and search for IAM

The screenshot shows the AWS search interface with the search term 'IAM' entered. The results are categorized into 'Services' and 'Features'. Under 'Services', the top result is 'IAM' with the description 'Manage access to AWS resources'. Other services listed include 'IAM Identity Center', 'AWS App Mesh', and 'Amazon Machine Learning'. Under 'Features', the top result is 'Amazon Pinpoint Campaign Orchestration'. Other features listed include 'Namespaces' and 'AMIs'.

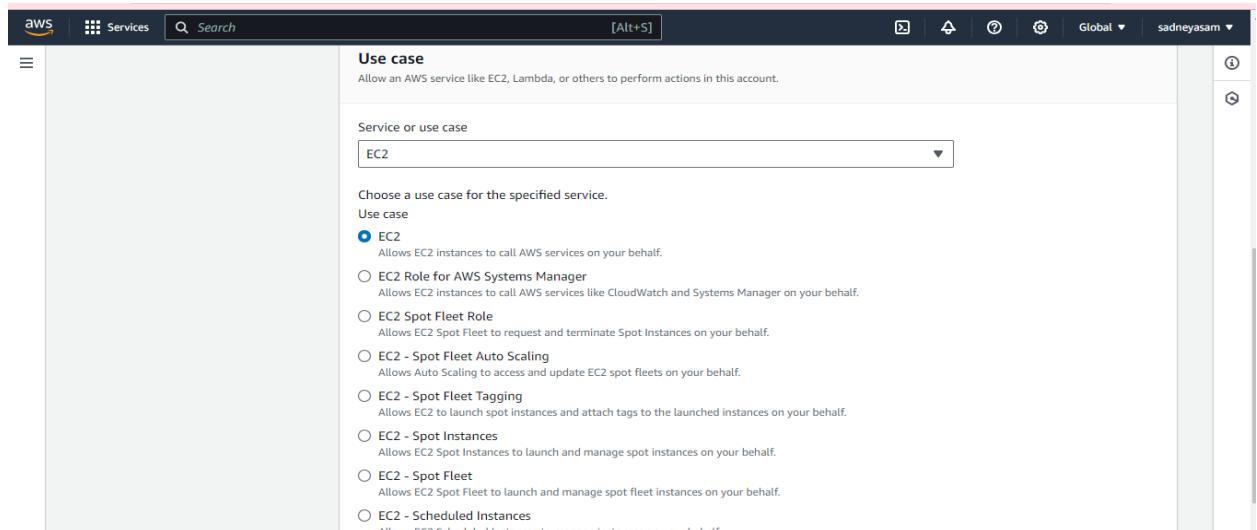
2. Then go into the role section and click on create role.

The screenshot shows the 'Roles' page in the AWS IAM console. The left sidebar shows navigation options like 'Identity and Access Management (IAM)', 'Dashboard', 'Access management', 'Access reports', and 'Related consoles'. The main area displays a table of existing roles, each with a checkbox, a 'Role name' column (listing names like 'aws-elasticbeanstalk-ec2-role', 'aws-elasticbeanstalk-service-role', etc.), a 'Trusted entities' column (listing AWS services), and a 'Last activity' column (showing times like '17 minutes ago', '20 minutes ago', etc.). A 'Create role' button is visible at the top right of the table area.

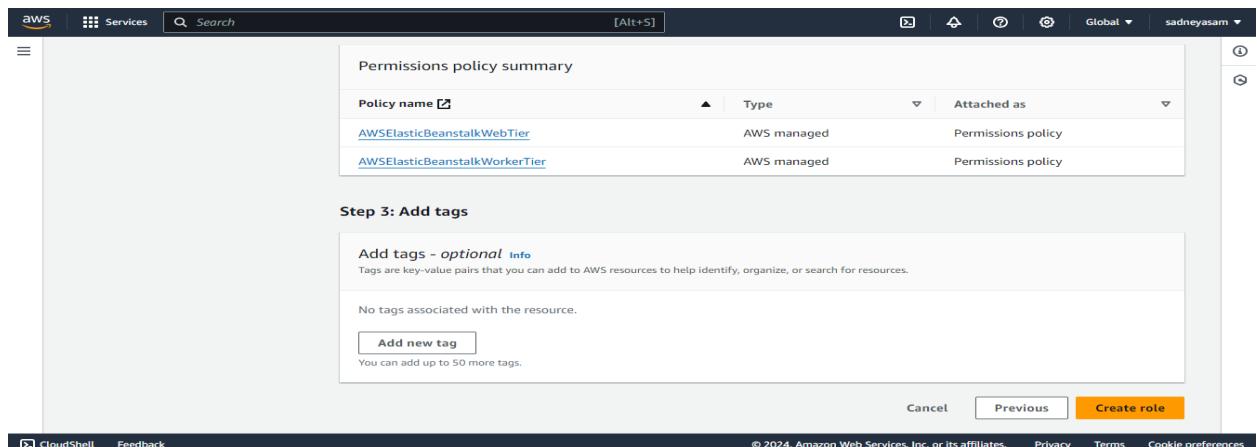
3. Then select a trusted entity as AWS service.



4. Select use case as EC2.



5. Select permissions as AWS Elastic Beanstalk Web Tier and AWS elastic Beanstalk worker tier.



6. Give a name to Role. Here I have given my role name as aws -elasticbeanstalk -ec2 - role.

The screenshot shows the 'Create role' wizard in the AWS IAM console. The current step is 'Step 3: Name, review, and create'. The 'Role details' section contains the following fields:

- Role name:** aws-elasticbeanstalk-ec2-role
- Description:** Allows EC2 instances to call AWS services on your behalf.

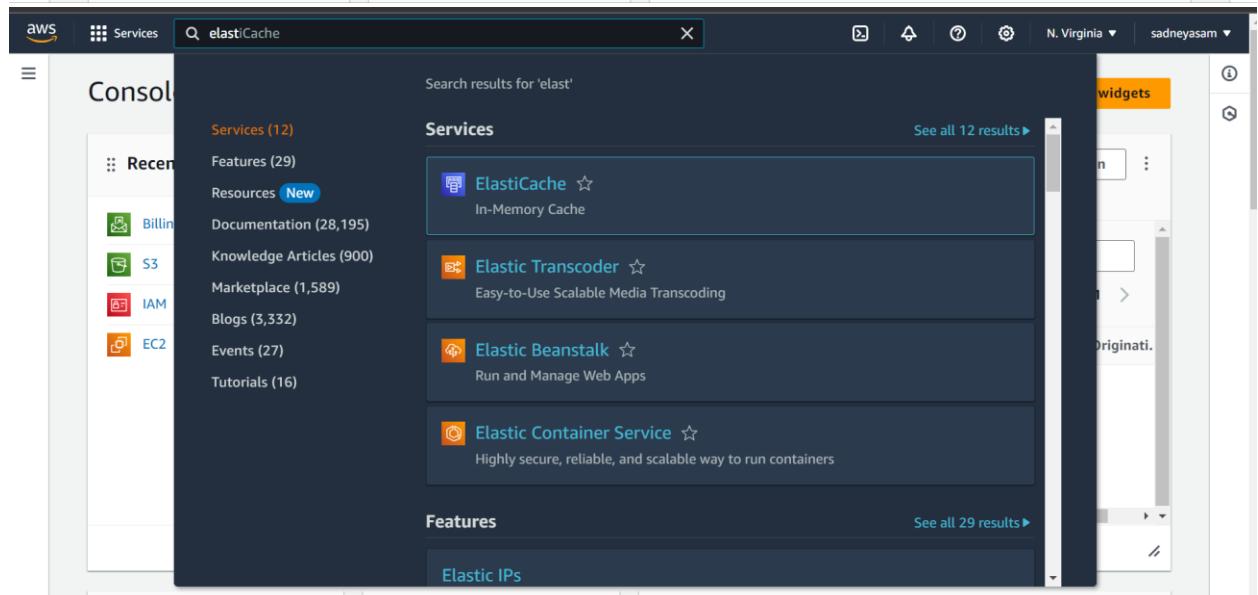
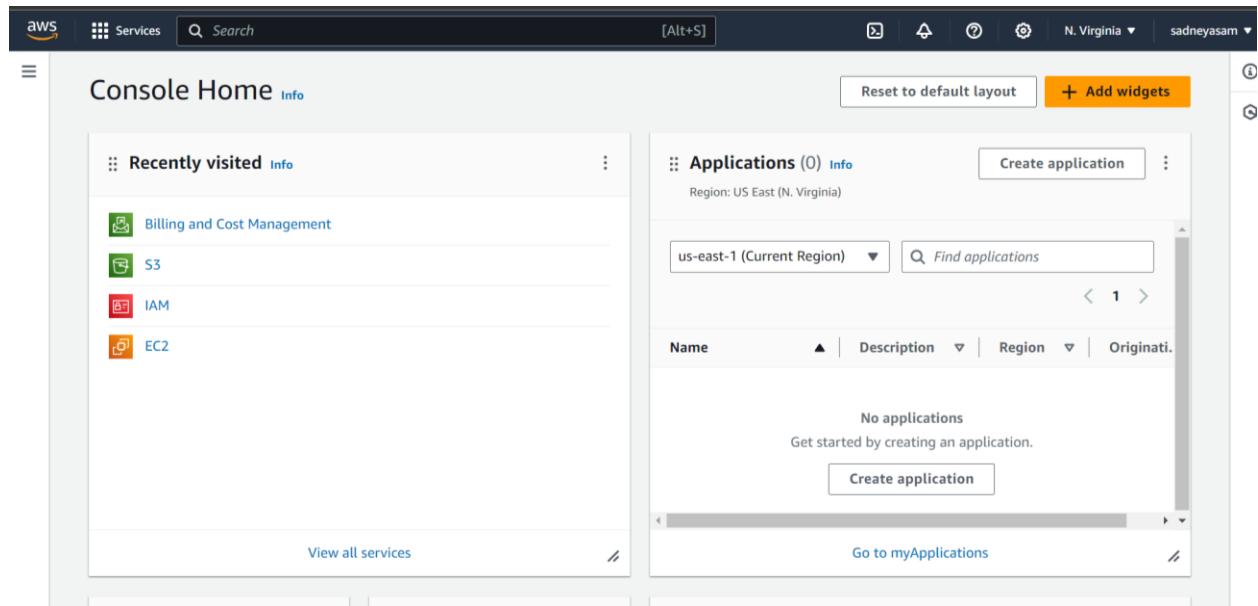
Below the role details, there is a summary of the selected trusted entities and a link to 'Trust policy'.

7. Then the role gets created.

The screenshot shows the details of the 'aws-elasticbeanstalk-ec2-role' in the AWS IAM console. The role was created on August 09, 2024, at 09:33 (UTC+05:30). The ARN is arn:aws:iam::851725480355:role/aws-elasticbeanstalk-ec2-role and the instance profile ARN is arn:aws:iam::851725480355:instance-profile/aws-elasticbeanstalk-ec2-role. The role has a maximum session duration of 1 hour. The 'Permissions' tab is selected, showing 2 managed policies attached to the role. Other tabs include 'Trust relationships', 'Tags', 'Access Advisor', and 'Revoke sessions'.

Step2:- Creation Elastic Beanstalk Environment

1. search for Elastic Beanstalk in the search box.



2. Open up Elastic Beanstalk and name your web app. (here I have given name sadneya123)

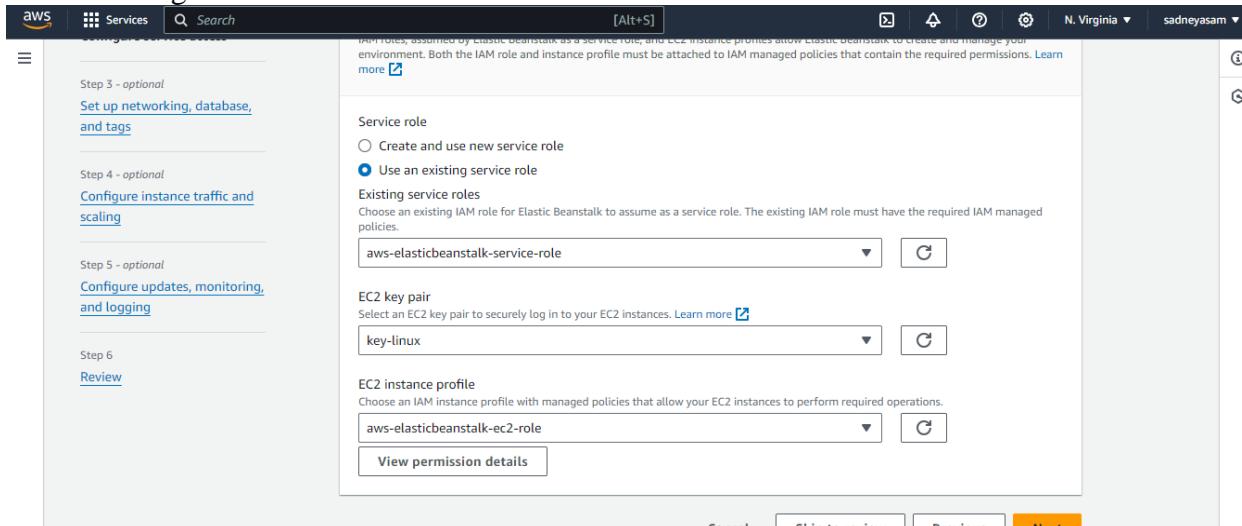
The screenshot shows the 'Configure environment' step of the Amazon Elastic Beanstalk setup wizard. On the left, a sidebar lists optional steps: Step 1 (Configure environment), Step 2 (Configure service access), Step 3 (optional: Set up networking, database, and tags), Step 4 (optional: Configure instance traffic and scaling), Step 5 (optional: Configure updates, monitoring, and logging), and Step 6 (Review). The main panel is titled 'Configure environment' and contains two sections: 'Environment tier' and 'Application information'. In 'Environment tier', 'Web server environment' is selected. In 'Application information', the 'Application name' is set to 'sadneya123'.

3. Select platform as PHP.

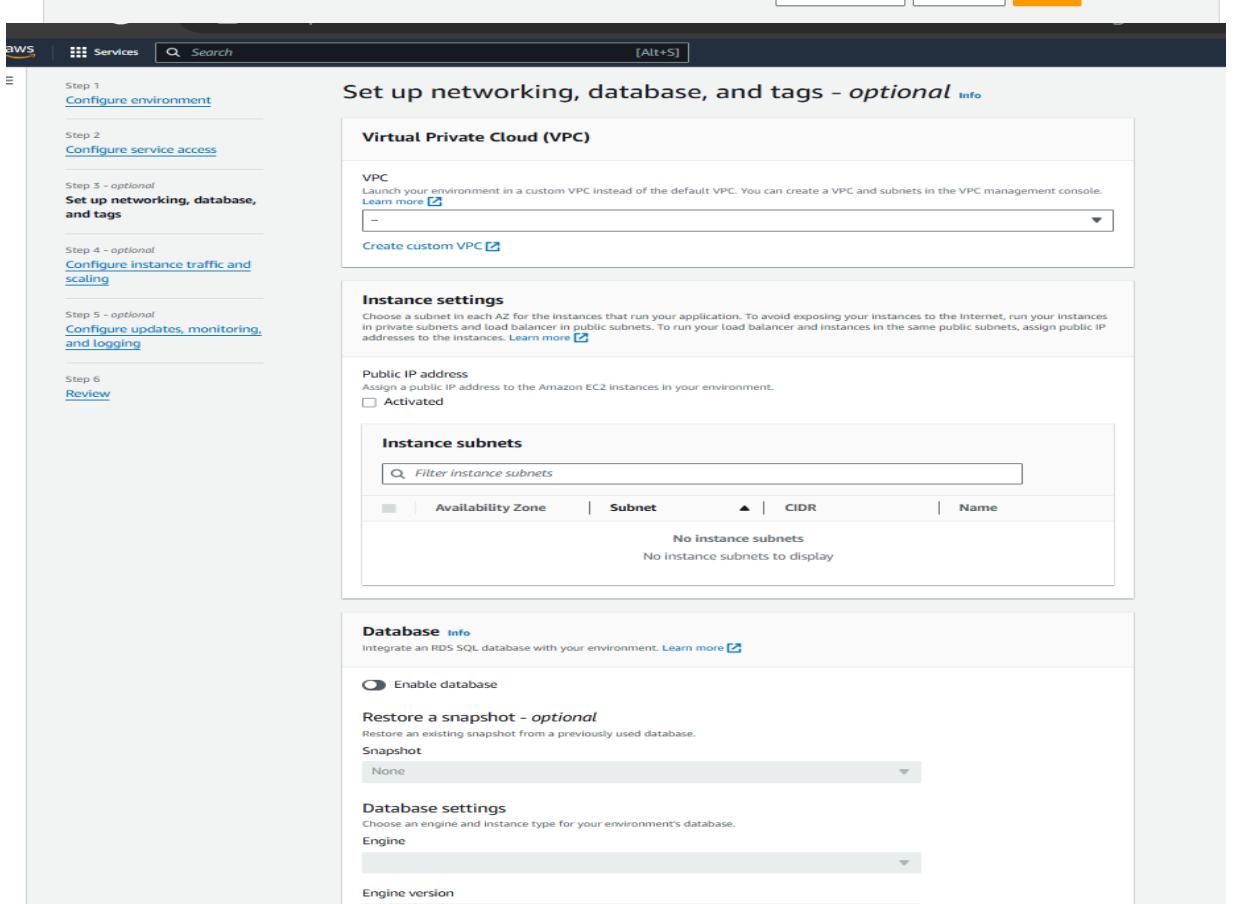
The screenshot shows the 'Platform' configuration step. It includes fields for 'Platform type' (selected: 'Managed platform'), 'Platform' (selected: 'PHP'), 'Platform branch' (selected: 'PHP 8.3 running on 64bit Amazon Linux 2023'), and 'Platform version' (selected: '4.3.1 (Recommended)'). Below these, the 'Application code' section shows 'Sample application' selected. The 'Presets' section shows 'Single instance (free tier eligible)' selected.

4. After clicking on next u need to select the use existing role. Then you will see the existing role select it like here it is aws-elasticbeanstalk-service-role. Which we created

in 1st part.Select role,then select key you have created then profile will be automatically selected according to role.then click on create application by keeping all the remaining settings as it is.



The screenshot shows the AWS Elastic Beanstalk configuration interface. On the left, a sidebar lists steps: 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 "Step 3 - optional: Set up networking, database, and tags". It includes sections for "Service role" (radio buttons for "Create and use new service role" and "Use an existing service role" (selected)), "Existing service roles" (dropdown menu showing "aws-elasticbeanstalk-service-role"), "EC2 key pair" (dropdown menu showing "key-linux"), and "EC2 instance profile" (dropdown menu showing "aws-elasticbeanstalk-ec2-role"). A "View permission details" button is also present. At the bottom are "Cancel", "Skip to review", "Previous", and a yellow "Next" button.



The screenshot shows the continuation of the AWS Elastic Beanstalk configuration. The sidebar remains the same. The main panel is titled "Set up networking, database, and tags - optional". It includes sections for "Virtual Private Cloud (VPC)" (with a note about launching in a custom VPC instead of default), "Create custom VPC" button, "Instance settings" (with a note about subnet placement), "Public IP address" (checkbox "Activated" is checked), "Instance subnets" (table showing "No instance subnets" and "No instance subnets to display"), "Database" (with "Enable database" radio button selected), "Restore a snapshot" (with "Snapshot" dropdown showing "None"), "Database settings" (with "Engine" dropdown and "Engine version" dropdown), and "Info" (with a note about integrating with RDS). At the bottom are "Cancel", "Skip to review", "Previous", and a yellow "Next" button.

Configure instance traffic and scaling - optional

- Instances info**: Configure the Amazon EC2 instances that run your application.
- Root volume (boot device)**
 - Root volume type**: Container default
 - Size**: 8 GB
 - IOPS**: 100 IOPS
 - Throughput**: 125 MB/s
- Amazon CloudWatch monitoring**: The time interval between when metrics are reported from the EC2 instances.
- Monitoring interval**: 5 minute
- Instance metadata service (IMDS)**: Your environment's platform supports both IMDSv1 and IMDSv2. To enforce IMDSv2, deactivate IMDSv1. Learn more [\[?\]](#)
 - IMDSv1**: With the current setting, the environment enables only IMDSv1.
 - Deactivated**
- EC2 security groups**: Select security group to control traffic.

Configure updates, monitoring, and logging - optional

- Monitoring info**
 - Health reporting**: Enhanced health reporting provides free real-time application and operating system monitoring of the instances and other resources in your environment. This EnhancedHealth custom metric is provided free with enhanced health reporting. Additional charges apply for each custom metric. For more information, see [Amazon CloudWatch Pricing](#) [\[?\]](#)
 - System**: Basic (radio button)
 - Enhanced** (radio button selected)
 - CloudWatch Custom Metrics - Instance**: Choose metrics
 - CloudWatch Custom Metrics - Environment**: Choose metrics
- Health event streaming to CloudWatch Logs**: Configure Elastic Load Balancing to stream environment health events to CloudWatch Logs. You can set the retention up to a maximum of ten years and configure Elastic Load Balancing to delete the logs when you terminate your environment.
 - Log streaming**: Activated (standard CloudWatch charges apply)
 - Retention**: 7 days
 - Lifecycle**: Keep logs after terminating environment
- Managed platform updates info**: Activate managed platform updates to apply platform updates automatically during a weekly maintenance window that you choose.
 - Managed updates**: Activated (radio button selected)
 - Weekly update window**: Sunday at 00 UTC
 - Update level**: Minor and patch
 - Instance replacement**: If enabled, an instance replacement will be scheduled if no other updates are available.
 - Activated**

Keep Set up networking, database, and tags ,Configure instance traffic and scaling,Configure updates, monitoring, and logging all these default.

- Beanstalk creates a sample environment for you to deploy your application. By default, it creates an EC2 instance, a security group, an Auto Scaling group, an Amazon S3 Bucket, Amazon CloudWatch alarms and a domain name for your Application.

The screenshot shows the AWS Elastic Beanstalk console. On the left, there's a sidebar with 'Applications', 'Environments', and 'Change history'. Under 'Application: sadneya123', it shows 'Application versions' and 'Saved configurations'. Under 'Environment: Sadneya123-env', it lists 'Go to environment', 'Configuration', 'Events', 'Health', 'Logs', and 'Monitoring'. The main area displays the 'Environment overview' with the message 'Environment successfully launched.' It includes sections for 'Health' (Ok), 'Environment ID' (e-vw23gecggs), 'Domain' (Sadneya123-env.eba-6w7emmur.us-east-1.elasticbeanstalk.com), 'Application name' (sadneya123), and 'Platform' (Node.js 20 running on 64bit Amazon Linux 2023/6.1.8). The 'Platform state' is 'Supported'. At the bottom, there are tabs for 'Events', 'Health', 'Logs', 'Monitoring', 'Alarms', 'Managed updates', and 'Tags'. A 'CloudShell' button is at the bottom left, and a footer at the bottom right.

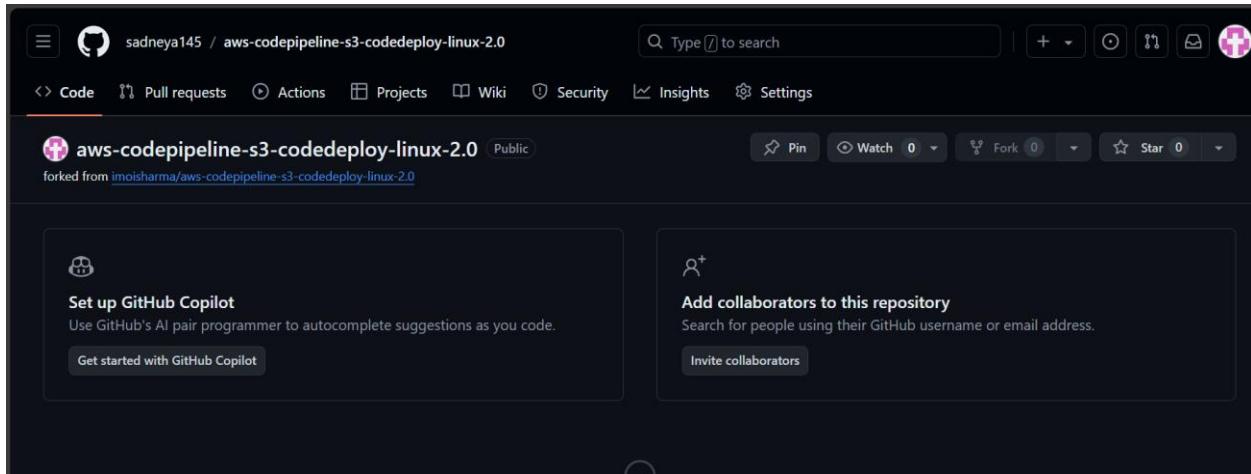
Step 3: Get a copy of your sample code

The screenshot shows the GitHub fork creation interface. The URL is 'imoisharma / aws-codepipeline-s3-codedeploy-linux-2.0'. The top navigation bar includes 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Security', and 'Insights'. The main section is titled 'Create a new fork'. It shows the 'Owner' as 'sadneya145' and the 'Repository name' as 'aws-codepipeline-s3-codedeploy-linux-2.0'. A note says 'aws-codepipeline-s3-codedeploy-linux-2.0 is available.'. Below this, there's a 'Description (optional)' field with the placeholder 'Use this sample when creating a simple pipeline in AWS CodePipeline while following the Simple Pipeline Walkthrough.' A checkbox is checked with the label 'Copy the master branch only', and a note below it says 'Contribute back to imoisharma/aws-codepipeline-s3-codedeploy-linux-2.0 by adding your own branch.' A note at the bottom says 'You are creating a fork in your personal account.' A green 'Create fork' button is at the bottom right.

In this step, we will get the sample code from this GitHub Repository to later host it. The pipeline takes code from the source and then performs actions on it.

For this experiment, as a source, we will use this forked GitHub repository. We can alternatively also use Amazon S3 and AWS CodeCommit.

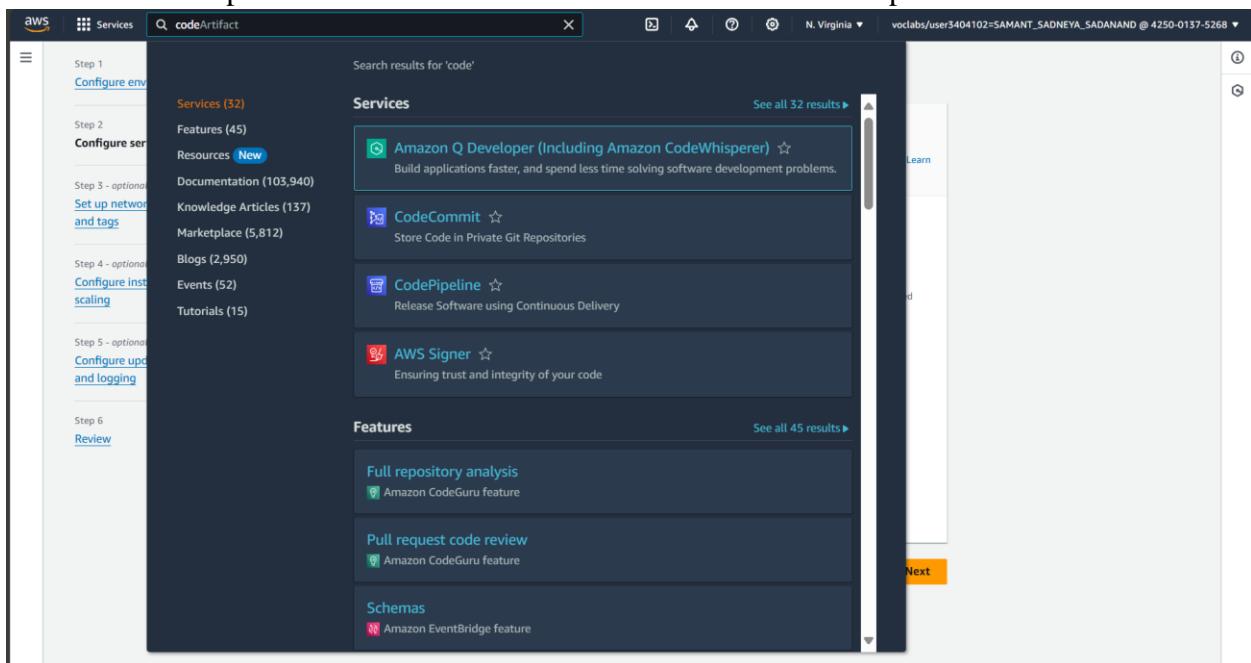
Go to the repository shared above and simply fork it.

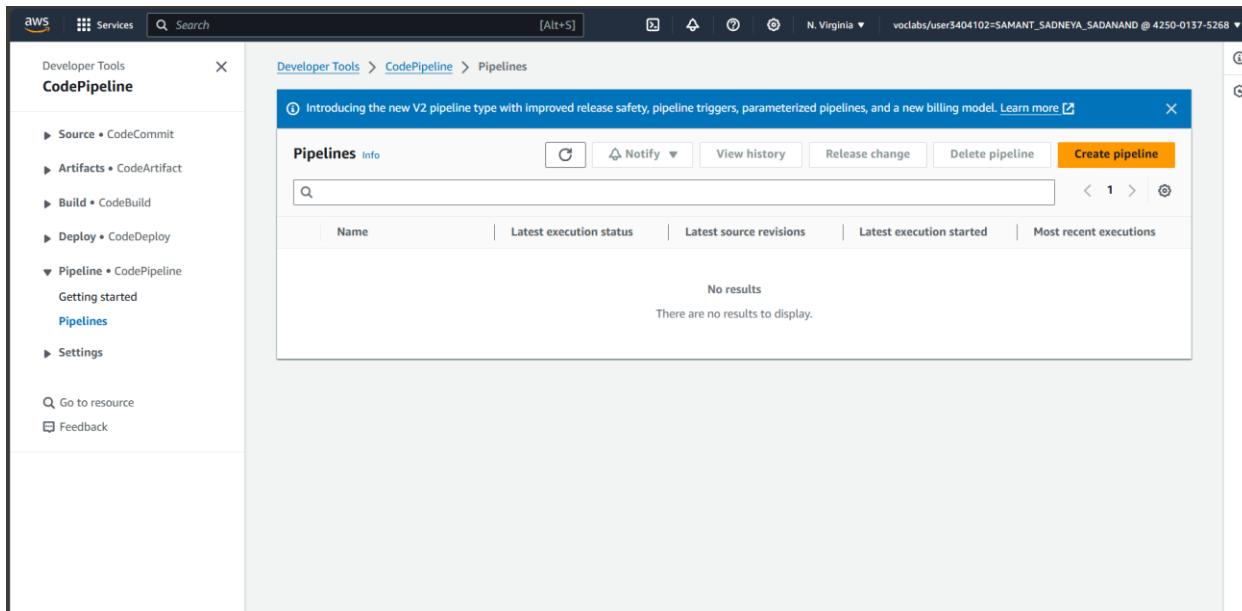


Step 4: Creating a CodePipeline

In this step, we'll create a simple pipeline that has its source and deployment information. In this case, however, we will skip the build stage where you get to plug in our preferred build provider.

1. Search CodePipeline in the search bar and click on create a new Pipeline.





2. Give a name to your pipeline. Here I have given name as sadneya_46

Choose pipeline settings Step 1 of 5

Pipeline settings

Pipeline name
Enter the pipeline name. You cannot edit the pipeline name after it is created.
sadneya46
No more than 100 characters

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

Execution mode
Choose the execution mode for your pipeline. This determines how the pipeline is run.
 Superseded
A more recent execution can overtake an older one. This is the default.
 Queued (Pipeline type V2 required)
Executions are processed one by one in the order that they are queued.
 Parallel (Pipeline type V2 required)
Executions don't wait for other runs to complete before starting or finishing.

Service role

New service role
Create a service role in your account

Existing service role
Choose an existing service role from your account

3. In the source stage, choose GitHub v2 as the provider, then connect your GitHub account to AWS by creating a connection. You'd need your GitHub credentials and then you'd need to authorize and install AWS on the forked GitHub Repository.

The screenshot shows the 'Create connection' page for AWS CodePipeline. The connection name is set to 'pipeline'. Under 'GitHub Apps', there is a search bar containing '53565526' and a button labeled 'Install a new app'. A blue banner at the top states: 'Beginning July 1, 2024, the console will create connections with codeconnections in the resource ARN. Resources with both service prefixes will continue to display in the console. [Learn more](#)'.

The screenshot shows the GitHub sign-in page. It displays the GitHub logo and the text: 'Sign in to GitHub to continue to AWS CodePipeline (N. Virginia)'. Below this, there is a 'Username or email address' field containing 'sadneyasam05@gmail.com', a 'Password' field with masked input, and a 'Sign in' button. At the bottom, there are links for 'Sign in with a passkey' and 'New to GitHub? Create an account'.

The screenshot shows the AWS CodePipeline interface. On the left, a sidebar lists steps: Step 3 (Add build stage), Step 4 (Add deploy stage), Step 5 (Review). The main area is titled "Source provider" and says "This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details." A dropdown menu shows "GitHub (Version 2)". Below it, a box says "New GitHub version 2 (app-based) action" with a link to learn more. A "Connection" section shows a search bar with "arn:aws:codeconnections:us-east-1:851725480355:connection/07d89a9a-42" and a "Connect to GitHub" button. A green box indicates "Ready to connect" with the message "Your GitHub connection is ready for use". Under "Repository name", a search bar contains "sadneya145/aws-codepipeline-s3-codedeploy-linux-2.0". A note says "You can type or paste the group path to any project that the provided credentials can access. Use the format 'group/subgroup/project'." At the bottom, there's a "Default branch" search bar with "master". The footer includes CloudShell, Feedback, and links to 2024 AWS terms and cookie preferences.

4. select the forked repository then select the master branch.

The screenshot shows the AWS CodePipeline interface. The main area is titled "Repository name" and says "Choose a repository in your GitHub account." A search bar contains "sadneya145/aws-codepipeline-s3-codedeploy-linux-2.0". A note says "You can type or paste the group path to any project that the provided credentials can access. Use the format 'group/subgroup/project'." Below it, a "Default branch" section says "Default branch will be used only when pipeline execution starts from a different source or manually started." A search bar contains "master". Under "Output artifact format", there are two options: "CodePipeline default" (selected) and "Full clone". The "CodePipeline default" box says "AWS CodePipeline uses the default zip format for artifacts in the pipeline. Does not include Git metadata about the repository." The "Full clone" box says "AWS CodePipeline passes metadata about the repository that allows subsequent actions to do a full Git clone. Only supported for AWS CodeBuild actions." The footer includes CloudShell, Feedback, and links to 2024 AWS terms and cookie preferences.

5. Then select trigger type none.

The screenshot shows the AWS CodePipeline interface. The main area is titled "Trigger" and says "Trigger type" with the sub-instruction "Choose the trigger type that starts your pipeline." Three radio buttons are shown: "No filter" (selected), "Specify filter", and "Do not detect changes". The "No filter" button has the description "Starts your pipeline on any push and clones the HEAD." The "Specify filter" button has the description "Starts your pipeline on a specific filter and clones the exact commit. Pipeline type V2 is required." The "Do not detect changes" button has the description "Don't automatically trigger the pipeline." The footer includes CloudShell, Feedback, and links to 2024 AWS terms and cookie preferences.

After that, click Continue and skip the build stage. Proceed to the Deployment stage.

Step 5: Deployment

1. Choose Beanstalk as the Deploy Provider, same region as the Bucket and Beanstalk, name and environment name.

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Add deploy stage Info

Step 4 of 5

You cannot skip this stage
Pipelines must have at least two stages. Your second stage must be either a build or deployment stage. Choose a provider for either the build stage or deployment stage.

Deploy

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

AWS Elastic Beanstalk

Region
US East (N. Virginia)

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

No more than 100 characters

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

sadneya123

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

Sadneya123-env-1

Configure automatic rollback on stage failure

Cancel Previous Next

2.Click Next, Review and create the pipeline.

The screenshot shows the 'Review' step of creating a new pipeline. The pipeline name is 'sadneya46', type is 'V2', execution mode is 'QUEUED', artifact location is 'codepipeline-us-east-1-204862929919', and service role name is 'arn:aws:iam::851725480355:role/service-role/AWSCodePipelineServiceRole-us-east-1-sadneya_46'. There are no variables defined at the pipeline level.

The screenshot shows the 'Step 2: Add source stage' configuration. It uses a GitHub (Version 2) source action provider with the following settings: OutputArtifactFormat 'CODE_ZIP', DetectChanges 'true', ConnectionArn 'arn:aws:codeconnections:us-east-1:851725480355:connection/800ab011-6749-4e3f-8a54-ba529c85155b', FullRepositoryId 'sadneya145/aws-codepipeline-s3-codedeploy-linux-2.0', and Default branch 'master'. The trigger configuration section indicates no additional triggers have been added.

Step 2: Add source stage

Source action provider

Source action provider
GitHub (Version 2)
OutputArtifactFormat
CODE_ZIP
DetectChanges
true
ConnectionArn
arn:aws:codeconnections:us-east-1:851725480355:connection/800ab011-6749-4e3f-8a54-ba529c85155b
FullRepositoryId
sadneya145/aws-codepipeline-s3-codedeploy-linux-2.0
Default branch
master

Trigger configuration
You can add additional pipeline triggers after the pipeline is created.

Trigger type
No filter

Step 3: Add build stage

Build action provider

Build stage
No build

Step 4: Add deploy stage

Deploy action provider

Deploy action provider
AWS Elastic Beanstalk

ApplicationName
sadneya123

EnvironmentName
Sadneya123-env

Configure automatic rollback on stage failure
Disabled

3. Then it will give you this result on screen. i.e. deployed successfully.

Congratulations! The pipeline sadneya46 has been created.

Developer Tools > CodePipeline > Pipelines > sadneya46

sadneya46

Pipeline type: V2 Execution mode: QUEUED

Source Succeeded Pipeline execution ID: 1874b523-d277-41ca-a269-c21357c7daa1

Source GitHub (Version 2) Succeeded - 1 minute ago 21d92e3c View details

21d92e3c Source: Update index.html

Disable transition

Deploy Succeeded Pipeline execution ID: 1874b523-d277-41ca-a269-c21357c7daa1

Deploy AWS Elastic Beanstalk Succeeded - Just now View details

21d92e3c Source: Update index.html

Start rollback

4. In a few minutes the website will get hosted successfully. Then click on the url present over the environment created on Elastic Benstalk.

Environment update successfully completed.

Elastic Beanstalk > Environments > Sadneya123-env-1

Sadneya123-env-1 Info

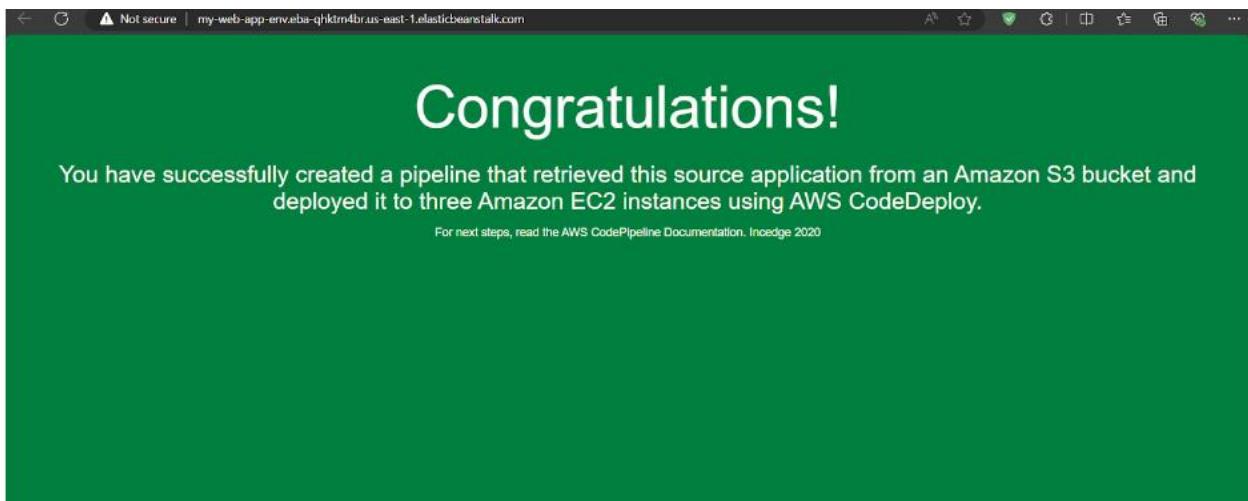
Actions Upload and deploy

Environment overview

Health	Environment ID
⚠ Warning - View causes	e-wax39sqfbq
Domain	Application name
Sadneya123-env-1.eba-6w7emmr.us-east-1.elasticbeanstalk.com	sadneya123

Platform

Platform	Platform state
PHP 8.3 running on 64bit Amazon Linux 2023/4.3.1	Supported
Running version	
code-pipeline-1723294146916-21d92e3c605bb7139abf0640332acc6	
205711000	



If you can see this, that means that you successfully created an automated software using CodePipeline.

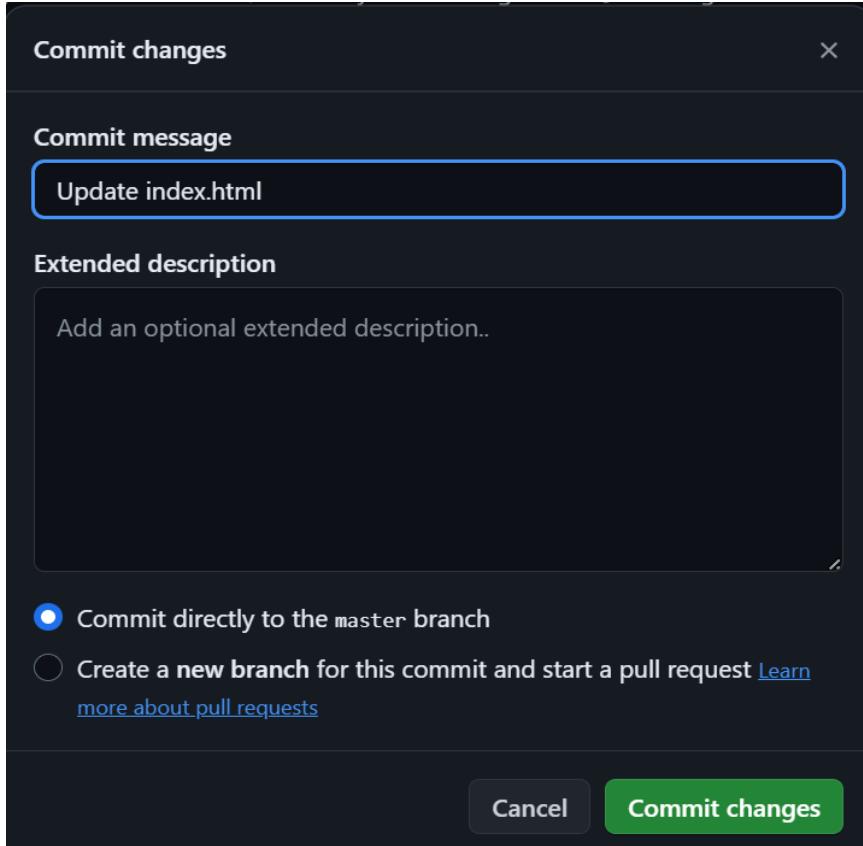
Step 6: Committing changes to update app

1. In this we make some changes in the file. Open github.com then open the forked repository. Then update the changes in the index.html file and finally commit those changes.

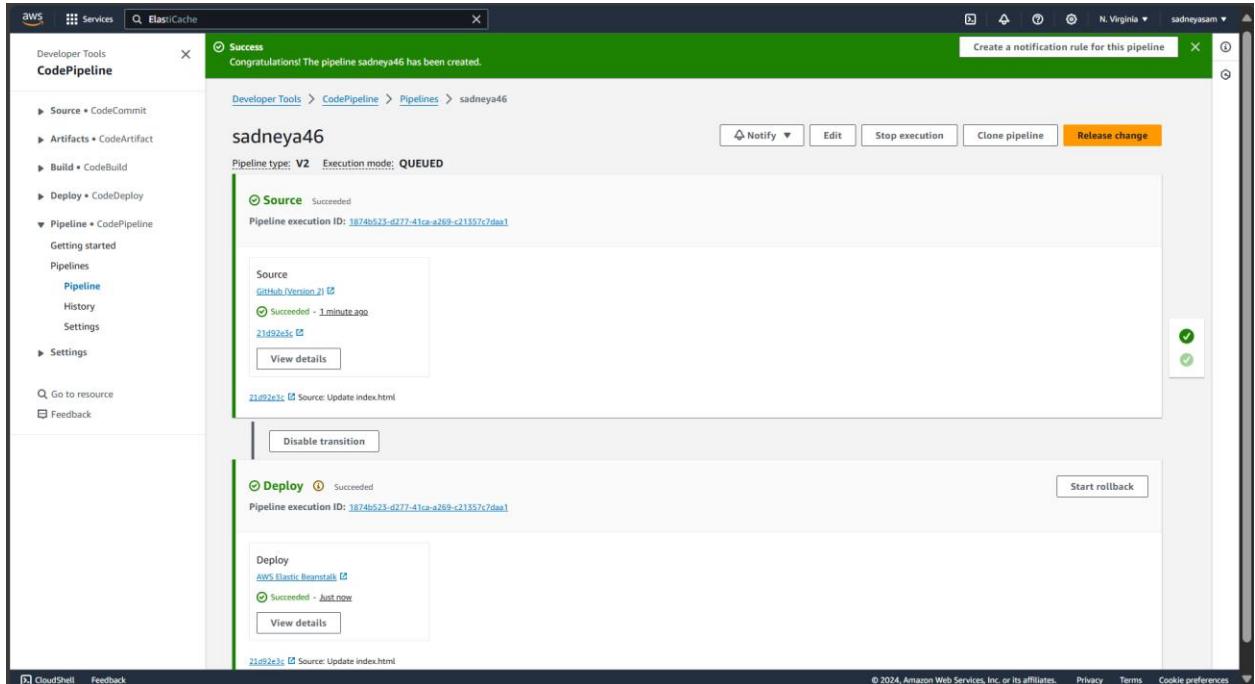
A screenshot of a GitHub repository page. The URL in the address bar is https://github.com/sadneya145/aws-codedepipeline-s3-codedeploy-linux-2.0. The repository is a fork of imoisharma/aws-codedepipeline-s3-codedeploy-linux-2.0. The main content shows a list of commits in the master branch. The commits are:

Author	Commit Message	Date
imoisharma	Update README.md	8fd5da5 · 3 years ago
	.github	Adding template
	dist	Added dist folder
	scripts	s3 setup and s3 set cache control scripts
	CODE_OF_CONDUCT.md	Adding CONTRIBUTING/CoC
	CONTRIBUTING.md	Adding CONTRIBUTING/CoC
	LICENSE	Added AWS CodePipeline Sample
	README.md	Update README.md
	app-specification.yml	Create app-spec config file
	appspec.yml	Update appspec.yml
	index.html	Update index.html

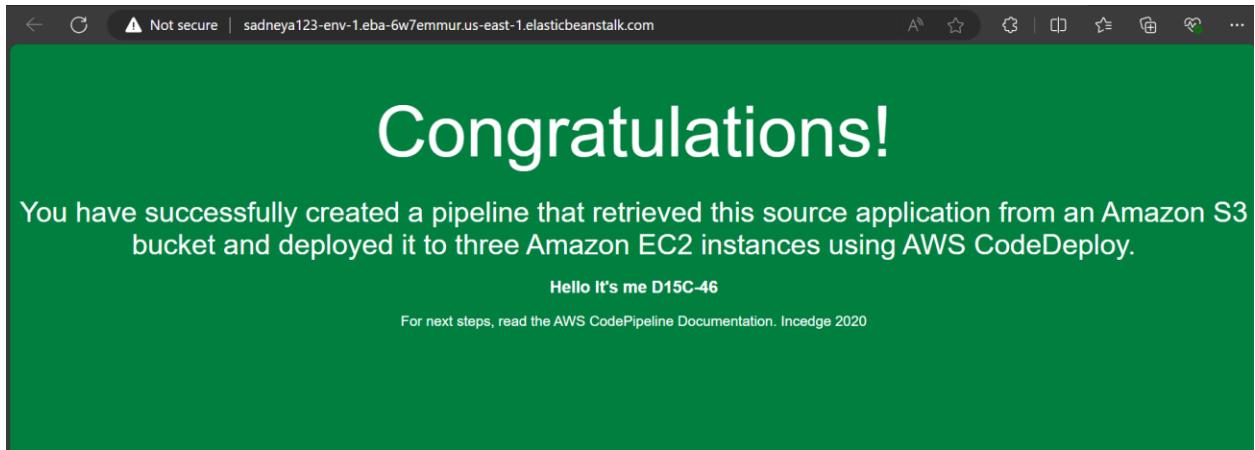
The right sidebar contains sections for About, Releases, Packages, and Languages.



2. Then again start the deployment of the pipeline.



3. Check the changes in the website , here I have added a message in h3 tag.



EXPERIMENT NO. 3

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

Procedure:

1. Creation Of Instance

The screenshot shows the AWS EC2 Dashboard. On the left, a sidebar lists various EC2-related services like Instances, Images, and Capacity. The main panel displays a summary of resources: 1 running instance, 1 Auto Scaling Group, 0 capacity reservations, 0 dedicated hosts, 0 elastic IPs, 0 instances, 0 key pairs, 0 load balancers, 0 placement groups, 0 security groups, 0 snapshots, 0 service health items, and 4 volumes. Below this, there's a section to 'Launch instance' with a prominent orange 'Launch instance' button. A note says instances will launch in the US East (N. Virginia) Region. To the right, a 'Service health' section shows 'AWS Health Dashboard' and indicates that the service is operating normally.

Search EC-2 instance. Then create three EC-2 instances and choose Amazon Linux as OS and also allow ssh traffic from anywhere.

This screenshot shows the 'Launch an instance' wizard. It starts with a 'Name and tags' step where 'kuber-master' is entered as the name. Next is the 'Application and OS Images (Amazon Machine Image)' step, which lists various AMIs including Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE. A search bar allows finding specific AMIs. The final step shown is the 'Amazon Machine Image (AMI)' details, which lists 'Amazon Linux 2023 AMI' with the identifier 'ami-0182f573e6f6f9c85'. It notes the AMI is 64-bit (x86), uefi-preferred, and includes ENA support. The status is 'Free tier eligible'.

Description

Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Architecture	Boot mode	AMI ID	Verified provider
64-bit (x86)	uefi-preferred	ami-0182f373e66fb9c85	

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

Instance type

t3.medium
 Family: t3 2 vCPU 4 GiB Memory Current generation: true
 On-Demand SUSE base pricing: 0.0979 USD per Hour
 On-Demand Windows base pricing: 0.06 USD per Hour
 On-Demand Linux base pricing: 0.0416 USD per Hour
 On-Demand RHEL base pricing: 0.0704 USD per Hour

All generations [Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

Key pair (login) [Info](#)

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

Key pair name - required

server [Create new key pair](#)

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

Network [Info](#)
 vpc-051bba342b3626898

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

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

Additional charges apply when outside of free tier allowance

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

Create security group Select existing security group

We'll create a new security group called 'launch-wizard-31' with the following rules:

Allow SSH traffic from Anywhere
 Helps you connect to your instance

Allow HTTPS traffic from the internet
 To set up an endpoint, for example when creating a web server

Allow HTTP traffic from the internet
 To set up an endpoint, for example when creating a web server

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

Configure storage [Info](#) [Advanced](#)

1x 8 GiB gp3 Root volume (Not encrypted)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

Summary

Number of instances [Info](#)
 1

Software Image (AMI)
 Amazon Linux 2023 AMI 2023.5.2...[read more](#)
 ami-0182f373e66fb9c85

Virtual server type (instance type)
 t3.medium

Firewall (security group)
 New security group

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

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

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

To efficiently run kubernetes cluster select instance type of at least t3.medium as kubernetes recommends at least 2 vCPU to run smoothly on it.

	Name 	Instance ID	Instance state	 
<input type="checkbox"/>	kube-master	i-00aa79ac09d7462c0	 Running	 
<input type="checkbox"/>	kube-worker1	i-0bab86cd3fbfcba0a	 Running	 
<input type="checkbox"/>	kube-worker2	i-00dcfd302ffd80dda	 Running	 

- Then for making connection through SSH into all 3 machines each in separate terminal
Use this following command:

ssh -i <keyname>.pem ubuntu@<public_ip_address> where keyname is name of the key you created here i created key server.pem and use public IP address.(I have entered this command on git bash where i entered in downloads where server.pem is stored then as the key is not accessible hence we need to change its mode using chmod 400 "key name.pem". Then use the given command for making connections).

```
Sadneya@DESKTOP-IEPNL3D MINGW64 ~ (master)
$ cd downloads

Sadneya@DESKTOP-IEPNL3D MINGW64 ~/downloads (master)
$ chmod 400 "server.pem"

Sadneya@DESKTOP-IEPNL3D MINGW64 ~/downloads (master)
$ ssh -i "server.pem" ec2-user@ec2-54-174-206-93.compute-1.amazonaws.com
The authenticity of host 'ec2-54-174-206-93.compute-1.amazonaws.com (54.174.206.93)' can't be established.
ED25519 key fingerprint is SHA256:T+tsGyI15gAvUvjeAZ7GjDIWXHOaI4EPF5g5oICrkoQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-174-206-93.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

      #_
      ~\_\ #####
      ~~ \#####\
      ~~   \###|
      ~~     \#/ ,__->
      ~~       V~' ,-->
      ~~~      /`-
      ~~ .-.
      ~~ /`-/
      _/m/`-
```

2. Installation Of Docker on three machines

- For installation of Docker into all three machines run the following command:
sudo yum install docker -y

```
[ec2-user@ip-172-31-87-114 ~]$ sudo yum install docker -y
Last metadata expiration check: 0:06:20 ago on Fri Sep 13 03:20:22 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-libs     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
  libnftnetlink     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

Downloading Packages:

```
(1/10): iptables-libs-1.8.8-3.amzn2023.0.2.x86_64 3.6 MB/s | 401 kB     00:00
(2/10): iptables-nft-1.8.8-3.amzn2023.0.2.x86_64 4.6 MB/s | 183 kB     00:00
```

Installed:

```
  containerd-1.7.20-1.amzn2023.0.1.x86_64
  docker-25.0.6-1.amzn2023.0.2.x86_64
  iptables-libs-1.8.8-3.amzn2023.0.2.x86_64
  iptables-nft-1.8.8-3.amzn2023.0.2.x86_64
  libcgroup-3.0-1.amzn2023.0.1.x86_64
  libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
  libnftnetlink-1.0.1-19.amzn2023.0.2.x86_64
  libnftnl-1.2.2-2.amzn2023.0.2.x86_64
  pigz-2.5-1.amzn2023.0.3.x86_64
  runc-1.1.13-1.amzn2023.0.1.x86_64
```

Complete!

- Then, configure cgroup in a daemon.json file by using following commands
cd /etc/docker

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

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

```
"storage-driver": "overlay2"
}
EOF
[ec2-user@ip-172-31-87-114 ~]$ cd /etc/docker
[ec2-user@ip-172-31-87-114 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"
}
```

- Then after this run the following command to enable and start docker and also to load the daemon.json file.
sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker

```
[ec2-user@ip-172-31-80-126 docker]$ sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
```

- Then check the version of docker installed.
docker -v

```
[ec2-user@ip-172-31-80-126 docker]$ docker -v
Docker version 25.0.5, build 5dc9bcc
```

3. Installation Of Kubernetes on three machines

- SELinux needs to be disable before configuring kubelet thus run the following command
sudo setenforce 0

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

```
[ec2-user@ip-172-31-80-126 docker]$ sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
```

- Here We are adding kubernetes using the repository whose command is given below.

```
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/repomd.xml.key
```

```
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
```

```
EOF
```

```
[ec2-user@ip-172-31-80-126 docker]$ 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/repomd.xml.key
```

```
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
```

```
EOF
```

```
[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/repomd.xml.key
```

```
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
```

- After that Run following command to make the updation and also to install kubelet ,kubeadm, kubectl: sudo yum update

```
[ec2-user@ip-172-31-80-126 docker]$ sudo yum update
```

```
Kubernetes
```

```
Dependencies resolved.
```

```
Nothing to do.
```

```
Complete!
```

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

```
[ec2-user@ip-172-31-80-126 docker]$ sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Last metadata expiration check: 0:00:10 ago on Fri Sep 13 10:31:17 2024.
Dependencies resolved.

=====
| Package           | Architecture | Version      | Repository | Size
|=====
| Installing:
|   kubeadm          | x86_64       | 1.30.5-150500.1.1 | kubernetes | 10 M
|   kubectl          | x86_64       | 1.30.5-150500.1.1 | kubernetes | 10 M
|   kubelet           | x86_64       | 1.30.5-150500.1.1 | kubernetes | 17 M
| Installing dependencies:
|   conntrack-tools  | x86_64       | 1.4.6-2.amzn2023.0.2 | amazonlinux | 208 k
|   cri-tools         | x86_64       | 1.30.1-150500.1.1 | kubernetes | 8.6 M
|   kubernetes-cni   | x86_64       | 1.4.0-150500.1.1 | kubernetes | 6.7 M
|   libnetfilter-cthelper | x86_64       | 1.0.0-21.amzn2023.0.2 | amazonlinux | 24 k
|   libnetfilter_cttimeout | x86_64       | 1.0.0-19.amzn2023.0.2 | amazonlinux | 24 k
|   libnetfilter_queue | x86_64       | 1.0.5-2.amzn2023.0.2 | amazonlinux | 30 k
| Transaction Summary
|=====
| Install 9 Packages
|=====
| Total          | 64 MB/s | 53 MB | 00:00
| 20 kB/s | 1.7 kB | 00:00
| Importing GPG key 0x9A296436:
|   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/:v1.30/rpm/repodata/repomd.xml.key
| Key imported successfully
| Running transaction check
| Transaction check succeeded.
| Running transaction test
| Transaction test succeeded.
| Running transaction
|   Preparing : 1/1
|   Installing : kubelet-cni-1.4.0-150500.1.1.x86_64
|   Installing : cri-tools-1.30.1-150500.1.1.x86_64
|   Installing : libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64
|   Installing : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64
|   Installing : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64
|   Installing : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64
|   Running scriptlet: conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64
|   Installing : kubelet-1.30.5-150500.1.1.x86_64
|   Running scriptlet: kubelet-1.30.5-150500.1.1.x86_64
|   Installing : kubeadm-1.30.5-150500.1.1.x86_64
|   Installing : kubectl-1.30.5-150500.1.1.x86_64
|   Running scriptlet: kubectl-1.30.5-150500.1.1.x86_64
|   Verifying   : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64
|   Verifying   : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64
|   Verifying   : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64
|   Verifying   : libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64
|   Verifying   : cri-tools-1.30.1-150500.1.1.x86_64
|   Verifying   : kubeadm-1.30.5-150500.1.1.x86_64
|   Verifying   : kubectl-1.30.5-150500.1.1.x86_64
|   Verifying   : kubelet-1.30.5-150500.1.1.x86_64
|   Verifying   : kubernetes-cni-1.4.0-150500.1.1.x86_64
| Installed:
|   conntrack-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
|   kubelet-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_cttimeout-1.0.0-19.amzn2023.0.2.x86_64 libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64
| Complete!
```

- After installing Kubernetes, we need to configure internet options to allow bridging.
 - sudo swapoff -a
 - echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf
 - sudo sysctl -p

```
[ec2-user@ip-172-31-80-126 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
```

4. Perform this ONLY on the Master machine

- Initialize kubernetes by typing below command

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

```
[ec2-user@ip-172-31-80-126 docker]$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
I0913 10:32:44.629146    26680 version.go:256] remote version is much newer: v1.31.0; falling back to: stable-1.30
[init] Using Kubernetes version: v1.30.4
[preflight] Running pre-flight checks

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 172.31.80.126:6443 --token jhtgwo.4qv2vtxrcf6nvgpk \
  --discovery-token-ca-cert-hash sha256:766e48546942419274bcd18c370d2492f6e49dac9f98890804362194690f0f4a
```

- So after initialization you will get token at the end for joining master and worker. Like here I got this :(save this token as it is required later.Then you can join any number of worker nodes by running the following on each as root.)

```
kubeadm join 172.31.80.126:6443 --token jhtgwo.4qv2vtxrcf6nvgpk\
--discovery-token-ca-cert-hash
sha256:766e48546942419274bcd18c370d2492f6e49dac9f98890804362194690f0f4a
```

- Also,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-172-31-80-126 docker]$ mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

- Then, add a common networking plugin called flannel file as mentioned in the code.
kubectl apply -f
<https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml>

```
[ec2-user@ip-172-31-80-126 docker]$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yaml
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
```

- Now to Check the created pod use this command
kubectl get pods

5. Perform this ONLY on the worker machines

Paste the below command on all 2 worker machines

- sudo yum install iproute-tc -y
- sudo systemctl enable kubelet
- sudo systemctl restart kubelet

Now use this

```
kubeadm join 172.31.80.126:6443 --token jhtgwo.4qv2vtxrcf6nvgpk\
--discovery-token-ca-cert-hash
sha256:766e48546942419274bcd18c370d2492f6e49dac9f98890804362194690f0f4a
```

(Optional To check the status of pods executed these commands:

Kubectl get pods -n kube-system :gives status of all pods

Kubectl get daemonstat -n kube-system: gives status of pod named daemonstat

```
[ec2-user@ip-172-31-87-114 docker]$ kubectl get pods -n kube-system
NAME                               READY   STATUS    RESTARTS   AGE
coredns-55cb58b774-fx12f           1/1    Running   0          100s
coredns-55cb58b774-xn14v           1/1    Running   0          100s
etcd-ip-172-31-87-114.ec2.internal 1/1    Running   1 (2m45s ago) 75s
kube-apiserver-ip-172-31-87-114.ec2.internal 1/1    Running   1 (2m15s ago) 2m11s
kube-controller-manager-ip-172-31-87-114.ec2.internal 0/1    CrashLoopBackOff 1 (8s ago) 70s
kube-proxy-4dv8m                   1/1    Running   2 (26s ago) 100s
kube-scheduler-ip-172-31-87-114.ec2.internal 1/1    Running   1 (2m45s ago) 76s
[ec2-user@ip-172-31-87-114 docker]$ kubectl get daemonset -n kube-system
NAME        DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
kube-proxy   1         1         1       1           1           kubernetes.io/os=linux 3m
```

)

Now to see whether master and workers get connected successfully or not run **kubectl get nodes** command on master machine

```
[ec2-user@ip-172-31-87-114 docker]$ kubectl get nodes
NAME           STATUS   ROLES      AGE   VERSION
ip-172-31-87-114.ec2.internal  Ready   control-plane  3m21s   v1.30.5
```

Conclusion: In these EC-2 instance created successfully on AWS Linux. Then I installed docker ,kuberneted and then kubelet ,kubeadm, kubectl.Then on Master machine ,I initailized the kubernetes which given me the token which will be used for connection of master and workers.then on slave I installed iproute and enabled and restarted kubelet then i enter the token which i got from master but there was an issue in joint.that is why on output i just got of only one pc mater on performing command kubectl get nodes.

Experiment 4

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

Procedure:

1. Creation Of EC-2 instance

- Create an EC2 AWS Linux instance on AWS .also edit the Security Group Inbound Rules to allow SSH. then select the t2.micro instance type

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with navigation links like 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Console-to-Code', 'Instances', 'Images', and 'AWS Health Dashboard'. The main area is titled 'Resources' and displays various metrics: Instances (running) 1, Auto Scaling Groups 1, Capacity Reservations 0, Dedicated Hosts 0, Elastic IPs 0, Instances 0, Key pairs 0, Load balancers 0, Placement groups 0, Security groups 0, Snapshots 0, and Volumes 4. Below this, the 'Launch instance' section is visible, featuring a large orange 'Launch instance' button. The 'Service health' section shows the region as 'US East (N. Virginia)' and a status message: 'This service is operating normally.' At the bottom, the 'Launch an instance' wizard is open, showing the 'Name and tags' step where 'kuber' is entered. It also shows the 'Application and OS Images (Amazon Machine Image)' step, where the 'Amazon Linux 2023 AMI' is selected. The AMI details show it's 'Free tier eligible' and includes information about virtualization, ENA support, and root device type.

Description
Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Architecture	Boot mode	AMI ID
64-bit (x86)	uefi-preferred	ami-0182f373e66f89c85

Verified provider

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

Instance type
t2.medium
Family: t2 2 vCPU 4 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0464 USD per Hour
On-Demand RHEL base pricing: 0.0752 USD per Hour
On-Demand Windows base pricing: 0.0644 USD per Hour
On-Demand SUSE base pricing: 0.1464 USD per Hour

Additional costs apply for AMIs with pre-installed software

Key pair (login) [Info](#)

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

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

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

Network [Info](#)
vpc-051bba342b3626898

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

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

Additional charges apply when outside of **free tier allowance**

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

Create security group Select existing security group

We'll create a new security group called 'launch-wizard-35' with the following rules:

Allow SSH traffic from Anywhere 0.0.0.0/0
 Allow HTTPS traffic from the internet To set up an endpoint, for example when creating a web server
 Allow HTTP traffic from the internet To set up an endpoint, for example when creating a web server

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

Summary

Number of instances [Info](#)
1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.5.2...[read more](#)
ami-0182f373e66f89c85

Virtual server type (instance type)
t2.medium

Firewall (security group)
New security group

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

Free tier: In your first year includes [X](#)
750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GiB of bandwidth to the internet.

Launch instance [Review commands](#)

Instances (1) [Info](#)

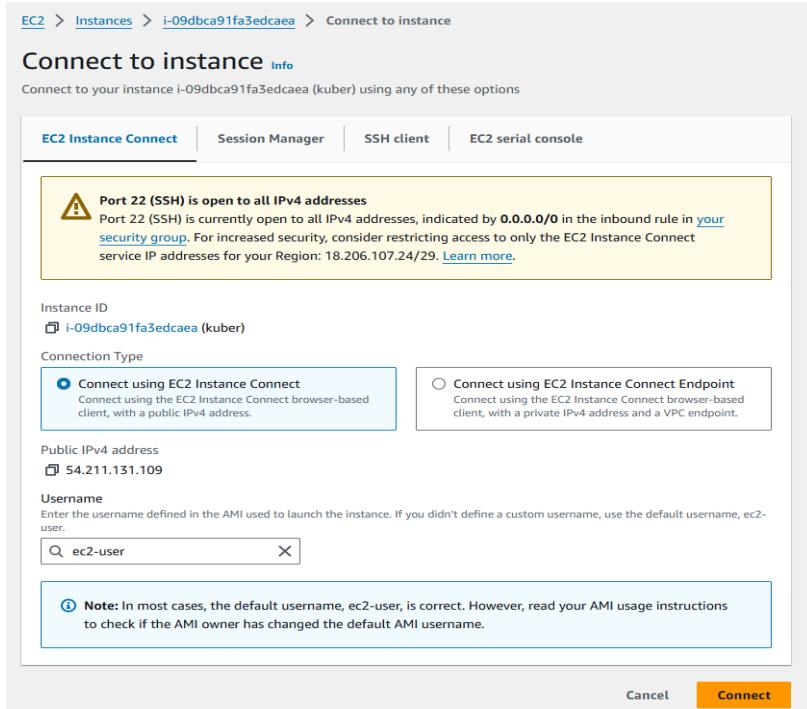
Last updated less than a minute ago [C](#) Connect Instance state Actions [Launch instances](#)

Find Instance by attribute or tag (case-sensitive) All states

Instance ID = i-09dbca91fa3edcaeaa X Clear filters

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
kuber	i-09dbca91fa3edcaeaa	Running View alarms	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-54-211-131-109.co...	54.211.131.109	-

- Thus Kuber named -instance gets created. Then click on Id of that instance then click on connect button you will see this:



- Then go into SSH client where you will get this command
Chmod 400 “keyname.pem”
ssh -i <keyname>.pem ubuntu@<public_ip_address> copy it and then connect it and run the following command for establishing connection.(I have entered this command on git bash where i entered in downloads where server.pem is stored then as the key is not accessible hence we need to change its mode using chmod 400 “key name.pem”. Then use the given command for making connections).

```

Sadneya@DESKTOP-IEPNL3D MINGW64 ~ (master)
$ cd downloads

Sadneya@DESKTOP-IEPNL3D MINGW64 ~/downloads (master)
$ chmod 400 "server.pem"

Sadneya@DESKTOP-IEPNL3D MINGW64 ~/downloads (master)
$ ssh -i "server.pem" ec2-user@ec2-54-196-176-21.compute-1.amazonaws.com
The authenticity of host 'ec2-54-196-176-21.compute-1.amazonaws.com (54.196.176.21)' can't be established.
ED25519 key fingerprint is SHA256:1E4CbWC3A0Kdn1J+99jTmUzur4joKQmThQyRcwIJzUU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-196-176-21.compute-1.amazonaws.com' (ED25519)
to the list of known hosts.

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

```

2. Installation of Docker

- For installation of Docker into the machines run the following command:
sudo yum install docker -y

```
[ec2-user@ip-172-31-26-174 ~]$ sudo yum install docker -y
Last metadata expiration check: 0:05:13 ago on Fri Sep 13 13:17:25 2024.
Dependencies resolved.
=====
| Package           | Architecture | Version      | Repository |
|=====             |=====         |=====        |=====       |
| Installing:      |              |              |            |
|   docker          | x86_64       | 25.0.6-1.amzn2023.0.2 | amazonlinux |
| Installing dependencies: |
|   containerd     | x86_64       | 1.7.20-1.amzn2023.0.1 | amazonlinux |
|   iptables-libs  | x86_64       | 1.8.8-3.amzn2023.0.2 | amazonlinux |
|   iptables-nft   | x86_64       | 1.8.8-3.amzn2023.0.2 | amazonlinux |
|   libcgroup      | x86_64       | 3.0-1.amzn2023.0.1   | amazonlinux |
|   libnetfilter_conntrack | x86_64 | 1.0.8-2.amzn2023.0.2 | amazonlinux |
|   libnftnlink    | x86_64       | 1.0.1-19.amzn2023.0.2 | amazonlinux |
|   libnftnl       | x86_64       | 1.2.2-2.amzn2023.0.2 | amazonlinux |
|   pigz           | x86_64       | 2.5-1.amzn2023.0.3   | amazonlinux |
|   runc           | x86_64       | 1.1.13-1.amzn2023.0.1 | amazonlinux |
| Transaction Summary |
|=====             |=====         |=====        |=====       |
| Total             |              |              |            |
| Running transaction check.
| Transaction check succeeded.
| Running transaction test
| Transaction test succeeded.
| Running transaction
|   Preparing      :
|   Installing     : runc-1.1.13-1.amzn2023.0.1.x86_64
|   Installing     : containerd-1.7.20-1.amzn2023.0.1.x86_64
|   Running scriptlet: containerd-1.7.20-1.amzn2023.0.1.x86_64
|   Installing     : pigz-2.5-1.amzn2023.0.3.x86_64
|   Installing     : libnftnl-1.2.2-2.amzn2023.0.2.x86_64
|   Installing     : libnftnlink-1.0.1-19.amzn2023.0.2.x86_64
|   Installing     : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
|   Installing     : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64
|   Installing     : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64
|   Running scriptlet: iptables-libs-1.8.8-3.amzn2023.0.2.x86_64
|   Installing     : libcgroup-3.0-1.amzn2023.0.1.x86_64
|   Running scriptlet: docker-25.0.6-1.amzn2023.0.2.x86_64
|   Installing     : docker-25.0.6-1.amzn2023.0.2.x86_64
|   Running scriptlet: docker-25.0.6-1.amzn2023.0.2.x86_64
| Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
| Verifying       : containerd-1.7.20-1.amzn2023.0.1.x86_64
| Verifying       : docker-25.0.6-1.amzn2023.0.2.x86_64
| Verifying       : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64
| Verifying       : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64
| Verifying       : libcgroup-3.0-1.amzn2023.0.1.x86_64
| Verifying       : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
| Verifying       : libnftnlink-1.0.1-19.amzn2023.0.2.x86_64
| Verifying       : libnftnl-1.2.2-2.amzn2023.0.2.x86_64
| Verifying       : pigz-2.5-1.amzn2023.0.3.x86_64
| Verifying       : runc-1.1.13-1.amzn2023.0.1.x86_64
| Installed:
|   containerd-1.7.20-1.amzn2023.0.1.x86_64           docker-25.0.6-1.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
|   libnftnlink-1.0.1-19.amzn2023.0.2.x86_64         libnftnl-1.2.2-2.amzn2023.0.2.x86_64
|   runc-1.1.13-1.amzn2023.0.1.x86_64                 pigz-2.5-1.amzn2023.0.3.x86_64
| Completions: 100%
```

- Then, configure cgroup in a daemon.json file by using following commands
cd /etc/docker

```
cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opt": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
EOF
```

```
[ec2-user@ip-172-31-26-174 ~]$ cd /etc/docker
[ec2-user@ip-172-31-26-174 docker]$ cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opt": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
EOF
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opt": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
```

- Then after this run the following command to enable and start docker and also to load the daemon.json file.

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

```
[ec2-user@ip-172-31-26-174 docker]$ sudo systemctl enable docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
[ec2-user@ip-172-31-26-174 docker]$ sudo systemctl daemon-reload
[ec2-user@ip-172-31-26-174 docker]$ sudo systemctl restart docker
[ec2-user@ip-172-31-26-174 docker]$ docker -v
Docker version 25.0.5, build 5dc9bcc
```

- docker -v

```
[ec2-user@ip-172-31-80-126 docker]$ docker -v
Docker version 25.0.5, build 5dc9bcc
```

3. Then Install Kubernetes with the following command.

- SELinux needs to be disable before configuring kubelet thus run the following command

```
sudo setenforce 0
```

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

```
[ec2-user@ip-172-31-26-174 docker]$ sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
[ec2-user@ip-172-31-26-174 docker]$ cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
```

- Here We are adding kubernetes using the repository whose command is given below.

```
cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
```

```
[kubernetes]
```

```
name=Kubernetes
```

```
baseurl=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/
```

```
enabled=1
```

```
gpgcheck=1
```

```
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/repo/repodata/repomd.xml.key
```

```
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
```

```
EOF
```

```
[ec2-user@ip-172-31-26-174 docker]$ sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
[ec2-user@ip-172-31-26-174 docker]$ 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/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
EOF
[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/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
```

- After that Run following command to make the updation and also to install kubelet ,kubeadm, kubectl:
`sudo yum update`

```
[ec2-user@ip-172-31-80-126 docker]$ sudo yum update
Dependencies resolved.
Nothing to do.
Complete!
```

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

```
[ec2-user@ip-172-31-80-126 docker]$ sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Last metadata expiration check: 0:00:10 ago on Fri Sep 13 10:31:17 2024.
Dependencies resolved.
=====
Transaction Summary
=====
Install 9 Packages

Total
Kubernetes
Importing GPG key 0x9A296436:
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:/v1.30/rpm/repo/repomd.xml.key
Key imported successfully
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing : 1/1
  Installing : kubelet-1.30.5-150500.1.1.x86_64 1/9
  Installing : cri-tools-1.30.1-150500.1.1.x86_64 2/9
  Installing : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64 3/9
  Installing : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64 4/9
  Installing : libnetfilter_ctqueue-1.0.5-2.amzn2023.0.2.x86_64 5/9
  Installing : comtrack-tools-1.4.6-2.amzn2023.0.2.x86_64 6/9
  Running scriptlet: comtrack-tools-1.4.6-2.amzn2023.0.2.x86_64 6/9
  Installing : kubelet-1.30.5-150500.1.1.x86_64 7/9
  Running scriptlet: kubelet-1.30.5-150500.1.1.x86_64 7/9
  Installing : kubeadm-1.30.5-150500.1.1.x86_64 8/9
  Installing : kubelet-1.30.5-150500.1.1.x86_64 9/9
  Running scriptlet: kubelet-1.30.5-150500.1.1.x86_64 9/9
  Verifying : comtrack-tools-1.4.6-2.amzn2023.0.2.x86_64 1/9
  Verifying : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64 2/9
  Verifying : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64 3/9
  Verifying : libnetfilter_ctqueue-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 : kubeadm-1.30.5-150500.1.1.x86_64 8/9
  Verifying : kubelet-1.30.5-150500.1.1.x86_64 9/9
  Installed:
    comtrack-tools-1.4.6-2.amzn2023.0.2.x86_64
    kubelet-1.30.5-150500.1.1.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_ctqueue-1.0.5-2.amzn2023.0.2.x86_64
    kubeadm-1.30.5-150500.1.1.x86_64
    kubelet-1.30.5-150500.1.1.x86_64
    libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64
    libnetfilter_ctqueue-1.0.5-2.amzn2023.0.2.x86_64
  Complete!
```

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

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

```
[ec2-user@ip-172-31-26-174 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-ip6tables = 1
```

4. Initialize the Kubecluster

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

```
[ec2-user@ip-172-31-80-126 docker]$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16  
I0913 10:32:44.629146 26680 version.go:256] remote version is much newer: v1.31.0; falling back to: stable-1.30  
[init] Using Kubernetes version: v1.30.4  
[preflight] Running pre-flight checks
```

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 172.31.26.174:6443 --token pv0yyi.xh1lqhclfjr50pt8 \
    --discovery-token-ca-cert-hash sha256:8293b2f6d29de466bd859007f5adbcdb3a
ecb0c446ba09033d32a5846b3d434f
```

- copy the token and save for future use .
kubeadm join 172.31.26.174:6443 --token pv0yyi.xhllqhclfjr50pt8
\\--discovery-token-ca-cert-hash
sha256:8293b2f6d29de466bd859007f5adbcd3aecb0c446ba09033d32a5846b3d434f
 - 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-172-31-80-126 docker]$ ls -l  
[ec2-user@ip-172-31-80-126 docker]$ mkdir -p $HOME/.kube  
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config  
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

- 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-f  
lannel.yaml
```

```
[ec2-user@ip-172-31-26-174 docker]$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flanne1.yaml  
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
```

5. Now that the cluster is up and running, we can deploy our nginx server on this cluster. Apply deployment using this following command:

```
kubectl apply -f https://k8s.io/examples/pods/simple-pod.yaml
```

```
[ec2-user@ip-172-31-26-174 docker]$ kubectl apply -f https://k8s.io/examples/pods/s  
imple-pod.yaml  
pod/nginx created
```

Then use **kubectl get pods** to check whether the pod gets created or not.

```
[ec2-user@ip-172-31-26-174 docker]$ kubectl get pods  
NAME      READY   STATUS    RESTARTS   AGE  
nginx    0/1     Pending   0          12s
```

To convert state from pending to running use following command:

kubectl describe pod nginx This command will help to describe the pods it gives reason for failure as it shows the untolerated taints which need to be untainted.

- kubectl describe pod nginx

```
[ec2-user@ip-172-31-26-174 docker]$ kubectl describe pod nginx  
Name:           nginx  
Namespace:      default  
Priority:       0  
Service Account: default  
Node:           <none>  
Labels:          <none>  
Annotations:    <none>  
Status:         Pending  
IP:  
IPs:  
Containers:  
  nginx:  
    Image:        nginx:1.14.2  
    Port:         80/TCP  
    Host Port:   0/TCP  
    Environment: <none>  
    Mounts:  
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-k4lj6 (ro)
```

```

Conditions:
  Type        Status
  PodScheduled  False
Volumes:
  kube-api-access-k4lj6:
    Type:           Projected (a volume that contains injected data from m
                    ultiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:      kube-root-ca.crt
    ConfigMapOptional:  <nil>
    DownwardAPI:       true
  QoS Class:        BestEffort
  Node-Selectors:   <none>
  Tolerations:     node.kubernetes.io/not-ready:NoExecute op=Exists for 3
                    00s
                    node.kubernetes.io/unreachable:NoExecute op=Exists for
                    300s
  Events:
    Type     Reason          Age   From            Message
    ----     ----          ----  ----            -----
    Warning  FailedScheduling 7s    default-scheduler  0/1 nodes are available: 1 no
de(s) had untolerated taint {node-role.kubernetes.io/control-plane: }. preemption:
0/1 nodes are available: 1 Preemption is not helpful for scheduling.

```

- kubectl taint nodes --all node-role.kubernetes.io/control-plane-

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

6. Now check pod status is is running perform **kubectl get pods** this command.

```
[ec2-user@ip-172-31-28-70 docker]$ kubectl get pods
NAME    READY  STATUS        RESTARTS   AGE
nginx  0/1   ContainerCreating  0          39s
[ec2-user@ip-172-31-28-70 docker]$ kubectl get pods
NAME    READY  STATUS        RESTARTS   AGE
nginx  1/1   Running  1 (45s ago)  70s
```

7. Lastly, mention the port you want to host. Here i have used localhost 8081 then check it.

kubectl port-forward nginx 8081:80

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

8. Verify your deployment

Open up a new terminal and ssh to your EC2 instance.

Then, use this curl command to check if the Nginx server is running.

curl --head http://127.0.0.1:8081

If the response is 200 OK and you can see the Nginx server name, your deployment was successful. We have successfully deployed our Nginx server on our EC2 instance.

Conclusion: Firstly I created an EC2 AWS Linux instance successfully.then installed docker and

kubernetes successfully.then initialized kubernetes which given me token and chown and mkdir command. Then I execute mkdir and chown the command successfully. Then I installed a networking plugin called flannel successfully. Then I tried to deploy nginx which initially gave an error. Then I deployed (simple-pod.yml) nginx successfully and also checked by using the get pods command.then hosted it on localhost 8081 ie <http://localhost:8081> successfully.

Experiment No 5

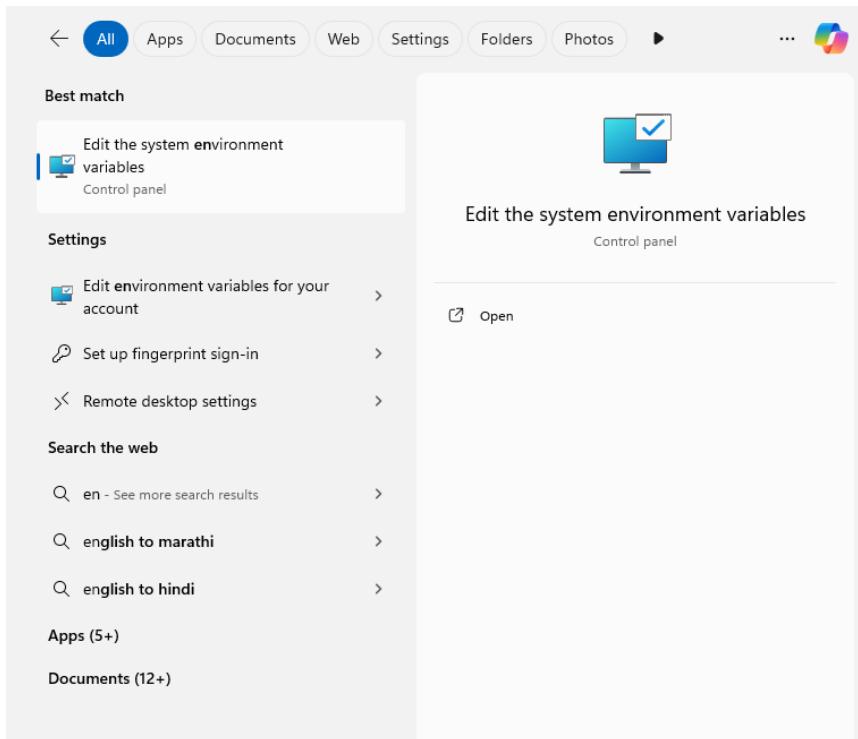
- To install , First download Terraform Client Utility for windows from terraforms official website <https://developer.hashicorp.com/terraform/install> then select the operating system on which you want to download.

The screenshot shows the Terraform download page. On the left, there's a sidebar with 'Operating Systems' options: macOS, Windows (selected), Linux, FreeBSD, OpenBSD, and Solaris. The main content area has two sections: 'Windows' and 'Linux'. Under 'Windows', there are 'Binary download' sections for '386' (Version: 1.9.4) and 'AMD64' (Version: 1.9.4). The 'AMD64' link is highlighted with a blue border. Under 'Linux', there's a 'Package manager' section with links for Ubuntu/Debian, CentOS/RHEL, Fedora, Amazon Linux, and Homebrew. A cookie consent banner at the bottom says 'We use cookies & other similar technology to collect data to improve your experience on our site, as described in our [Privacy Policy](#) and [Cookie Policy](#)' with 'ACCEPT' and 'Manage Preferences' buttons.

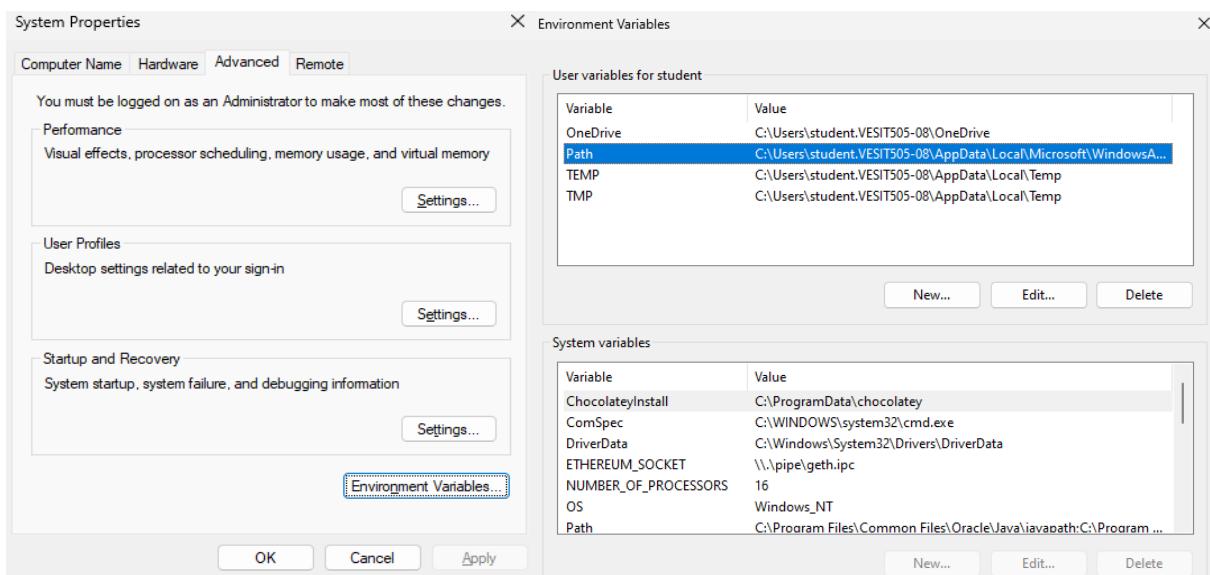
- Extract the file from downloads. Then click on terraform Application.

The screenshot shows a Windows File Explorer window. The path is 'Downloads > terraform_1.9.4_windows_amd64'. The 'terraform' file is selected and highlighted with a blue border. The 'LICENSE' file is also visible. The left sidebar shows standard folder icons for Home, Gallery, OneDrive, Desktop, Downloads, Documents, Pictures, Music, Videos, and a few others like 'HTML netflix pa' and 'akshad'.

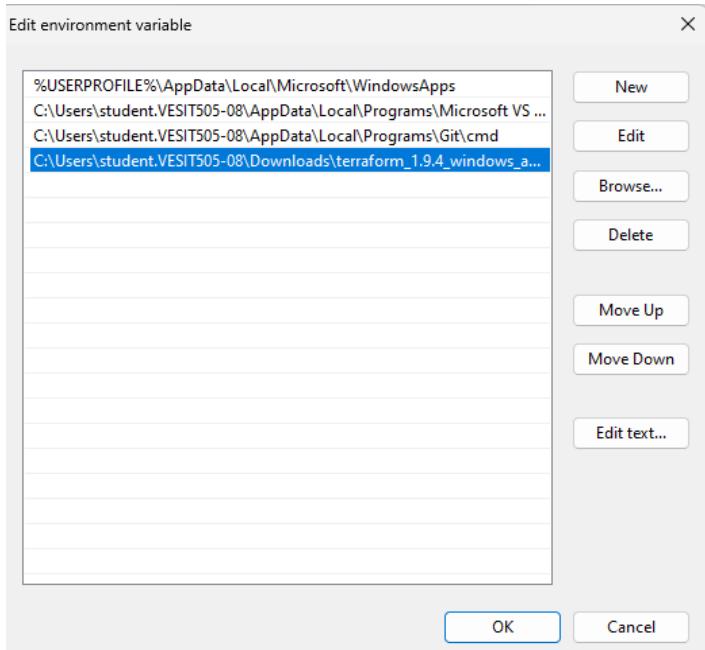
3. Then search for Edit the system environment variables.



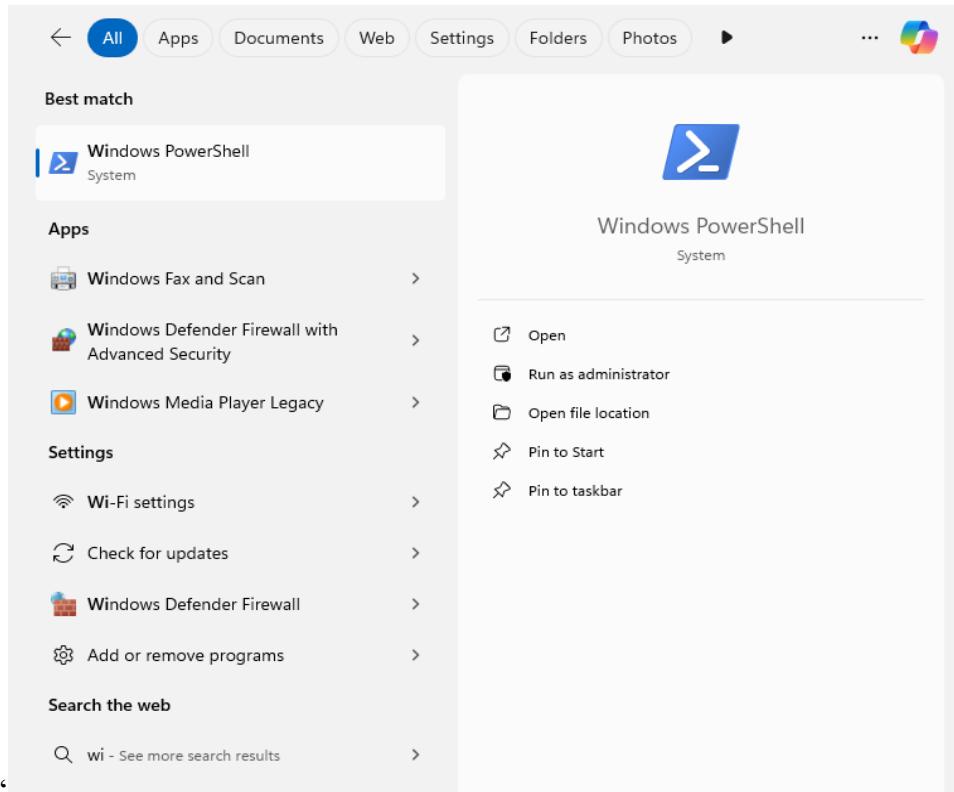
4. Then click on Environment Variables and then click on PATH.



5. Then add the path of where you have stored the Terraform. Then click on ok.



6. Then search for windows powershell and open it as an administrator.



7. Then type terraform which will give information about commands ie. main commands and all other commands.

```
Windows PowerShell x + v
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

PS C:\Users\student.VESIT505-08> 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

  untaint   Remove the 'tainted' state from a resource instance
  version   Show the current Terraform version
  workspace Workspace management

Global options (use these before the subcommand, if any):
  -chdir=DIR  Switch to a different working directory before executing the
              given subcommand.
  -help       Show this help output, or the help for a specified subcommand.
  -version    An alias for the "version" subcommand.
PS C:\Users\student.VESIT505-08> |
```

EXPERIMENT NO:6**A. Creating docker image using terraform**

1) Download and Install Docker Desktop from <https://www.docker.com/>

Step 1: Check the DOCKER Functionality.



```
Command Prompt      X + ▾
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sadneya>docker

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

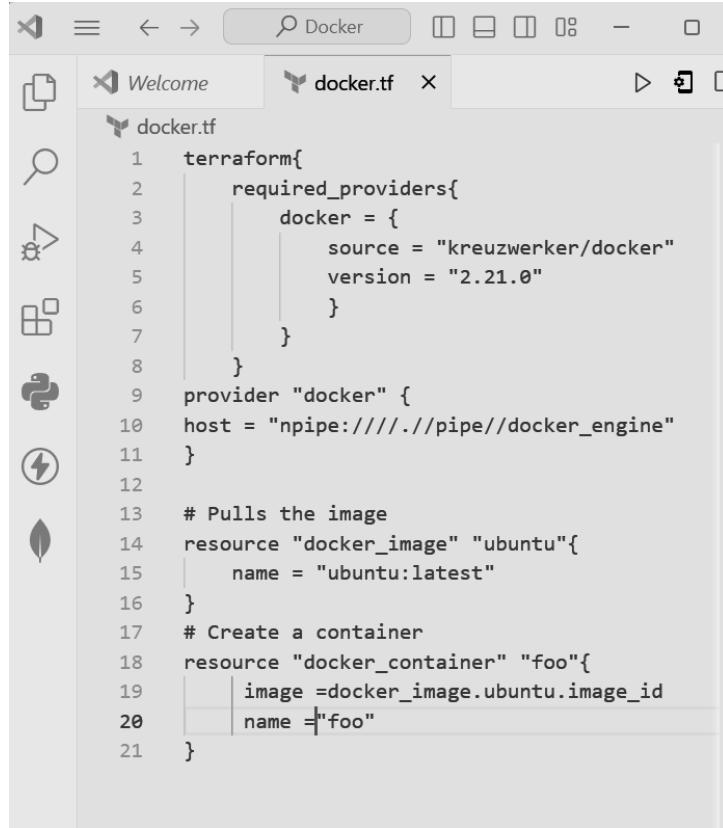
Common Commands:
  run      Create and run a new container from an image
  exec    Execute a command in a running container
  ps       List containers
  build   Build an image from a Dockerfile
  pull    Download an image from a registry
  push    Upload an image to a registry
  images  List images
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

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

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

Now, create a folder and give it name as ‘Terraform Scripts’ in which we save our different types which will be further used

Step 2: Firstly create a new folder named ‘Docker’ in the ‘TerraformScripts’ folder. Then create a new docker.tf file in VS code and write the following script.



```

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

```

Step 3: Execute Terraform Init command to initialize the resources

```

C:\Users\Sadneya>cd TerraformScripts
C:\Users\Sadneya\TerraformScripts>cd Docker
C:\Users\Sadneya\TerraformScripts\Docke>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 4: Execute Terraform plan to see the available resources

```

  Command Prompt      + 
C:\Users\Sadneya\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)
    + tty             = false
    + healthcheck     = (known after apply)
    + labels          = (known after apply)
}

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

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

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

C:\Users\Sadneya\TerraformScripts\Docker>

```

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

```
Command Prompt + v

C:\Users\Sadneya\TerraformScripts\Docker>terraform apply
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:
+ create

Terraform will perform the following actions:

# docker_container.foo will be created
+ resource "docker_container" "foo" {
    + attach                = false
    + bridge                 = (known after apply)
    + command               = [
        + "sleep",
        + "3600",
    ]
    + container_logs         = (known after apply)
    + entrypoint             = (known after apply)
    + env                    = (known after apply)
    + exit_code              = (known after apply)
    + gateway                = (known after apply)
    + hostname               = (known after apply)
    + id                     = (known after apply)
    + image                  = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a"
    + init                   = (known after apply)
    + ip_address              = (known after apply)
    + ip_prefix_length        = (known after apply)
    + ipc_mode                = (known after apply)
    + log_driver              = (known after apply)
    + logs                   = false
    + must_run                = true
    + name                   = "foo"
    + network_data            = (known after apply)
    + read_only                = false
    + remove_volumes          = true

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

    + healthcheck (known after apply)
    + labels (known after apply)
}

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

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

Enter a value: yes

docker_container.foo: Creating...
docker_container.foo: Creation complete after 1s [id=292de3d48673e0e7619d1826bdfcf93e34bfa8a9696e83b5f404b8ffff82309]

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

C:\Users\Sadneya\TerraformScripts\Docker>
```

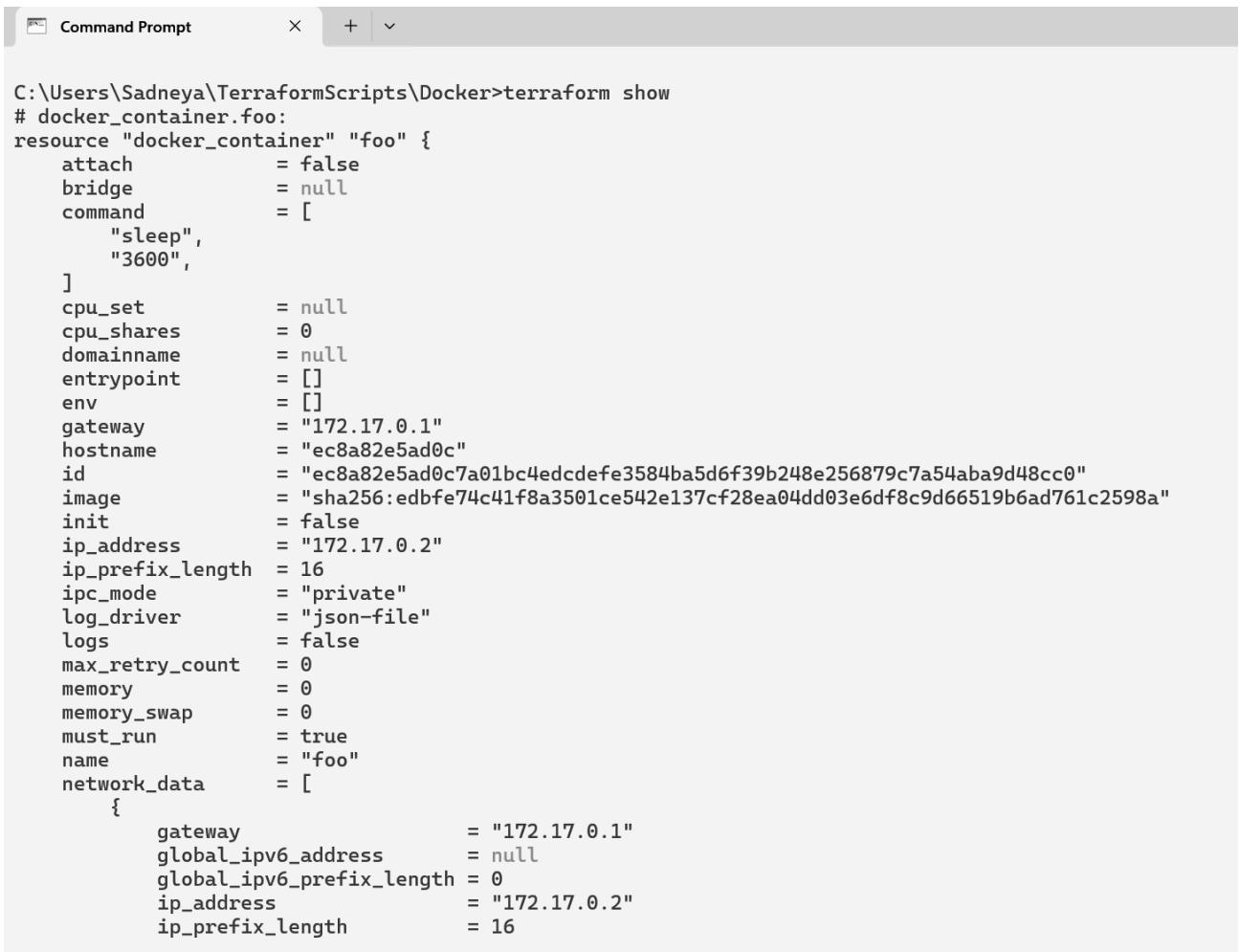
Docker images, Before Executing Apply step:

```
C:\Users\Sadneya\TerraformScripts\Docker>docker images
REPOSITORY      TAG          IMAGE ID      CREATED       SIZE
C:\Users\Sadneya\TerraformScripts\Docker>
```

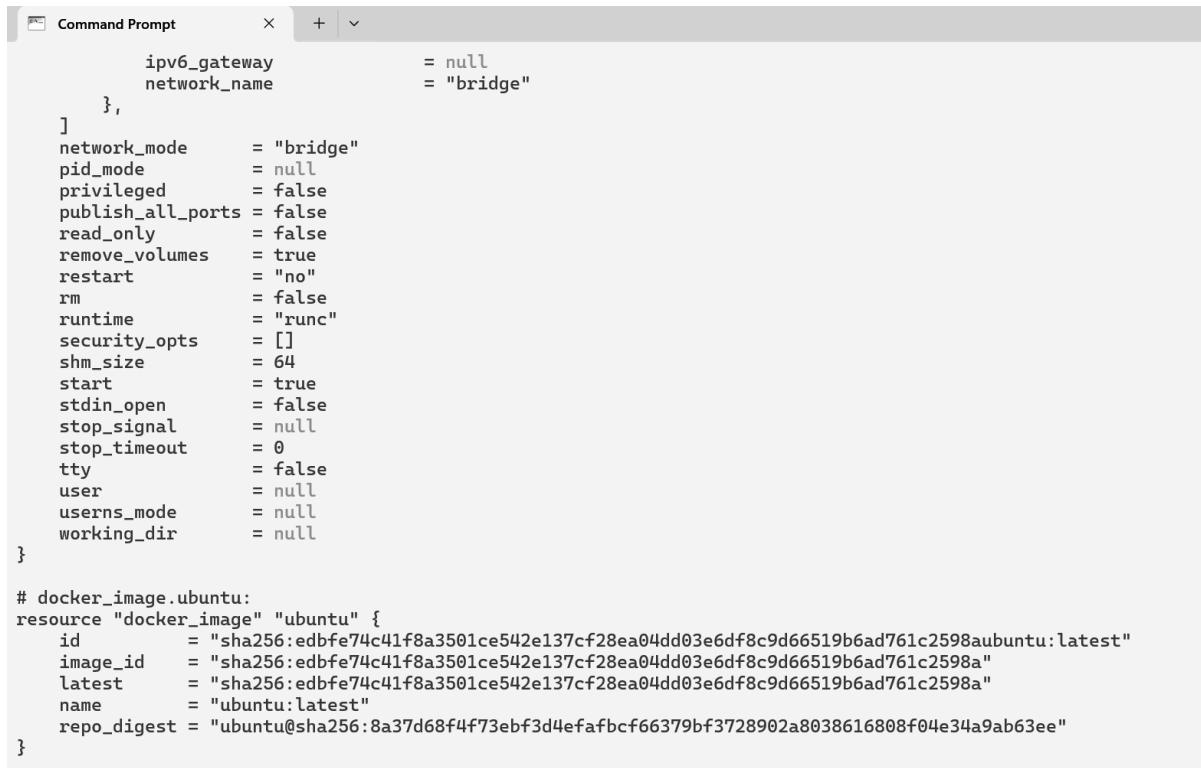
Docker images, After Executing Apply step:

```
C:\Users\Sadneya\TerraformScripts\Docker>docker images
REPOSITORY      TAG          IMAGE ID      CREATED       SIZE
ubuntu          latest       edbfe74c41f8  3 weeks ago  78.1MB
```

Step 6: terraform show it will Show the state file in a human-readable format.



```
Command Prompt      + ▾
C:\Users\Sadneya\TerraformScripts\Docker>terraform show
# docker_container.foo:
resource "docker_container" "foo" {
  attach          = false
  bridge          = null
  command         = [
    "sleep",
    "3600",
  ]
  cpu_set         = null
  cpu_shares     = 0
  domainname     = null
  entrypoint      = []
  env             = []
  gateway         = "172.17.0.1"
  hostname        = "ec8a82e5ad0c"
  id              = "ec8a82e5ad0c7a01bc4edcdefe3584ba5d6f39b248e256879c7a54aba9d48cc0"
  image           = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a"
  init            = false
  ip_address      = "172.17.0.2"
  ip_prefix_length = 16
  ipc_mode        = "private"
  log_driver       = "json-file"
  logs            = false
  max_retry_count = 0
  memory          = 0
  memory_swap     = 0
  must_run         = true
  name             = "foo"
  network_data    = [
    {
      gateway          = "172.17.0.1"
      global_ipv6_address = null
      global_ipv6_prefix_length = 0
      ip_address        = "172.17.0.2"
      ip_prefix_length   = 16
    }
  ]
}
```



```

  ipv6_gateway      = null
  network_name     = "bridge"
},
]
network_mode      = "bridge"
pid_mode         = null
privileged        = false
publish_all_ports = false
read_only          = false
remove_volumes    = true
restart           = "no"
rm                = false
runtime            = "runc"
security_opts     = []
shm_size          = 64
start              = true
stdin_open         = false
stop_signal        = null
stop_timeout       = 0
tty                = false
user               = null
userns_mode        = null
working_dir        = null
}

# docker_image.ubuntu:
resource "docker_image" "ubuntu" {
  id      = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest"
  image_id = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a"
  latest   = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a"
  name     = "ubuntu:latest"
  repo_digest = "ubuntu@sha256:8a37d68f4f73ebf3d4efafbcf66379bf3728902a8038616808f04e34a9ab63ee"
}

```

Step 7: terraform state list displays Lists out all the resources that are tracked in the current state file.

```
C:\Users\Sadneya\TerraformScripts\Docker>terraform state list
docker_container.foo
docker_image.ubuntu
```

Step 8: terraform graph Produces a graph in DOT language showing the dependencies between objects in the state file. This can then be rendered by a program called Graphviz (amongst others).

```
C:\Users\Sadneya\TerraformScripts\Docker>terraform graph
digraph G {
  rankdir = "RL";
  node [shape = rect, fontname = "sans-serif"];
  "docker_container.foo" [label="docker_container.foo"];
  "docker_image.ubuntu" [label="docker_image.ubuntu"];
  "docker_container.foo" --> "docker_image.ubuntu";
}
```

Step 9: Execute Terraform destroy to delete the configuration, which will automatically delete the Ubuntu Container.

```
C:\Users\Sadneya\TerraformScripts\Docker>terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Refreshing state... [id=292de3d48673e0e7619d1826bdfcf93e34bfa8a9696e83b5f404b8fdfff82309]

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

Terraform will perform the following actions:

# docker_container.foo will be destroyed
- resource "docker_container" "foo" {
    - attach           = false -> null
    - command         = [
        - "sleep",
        - "3600",
    ] -> null
    - cpu_shares      = 0 -> null
    - dns              = [] -> null
    - dns_opts         = [] -> null
    - dns_search       = [] -> null
    - entrypoint       = [] -> null
    - env              = [] -> null
    - gateway          = "172.17.0.1" -> null
    - group_add        = [] -> null
    - hostname         = "292de3d48673" -> null
    - id               = "292de3d48673e0e7619d1826bdfcf93e34bfa8a9696e83b5f404b8fdfff82309" -> null
    - image             = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> null
    - init              = false -> null
    - ip_address        = "172.17.0.2" -> null
    - ip_prefix_length = 16 -> null
    - ipc_mode          = "private" -> null
    - links             = [] -> null
    - log_driver         = "json-file" -> null
    - log_opts           = {} -> null
    - logs              = false -> null
        - max_retry_count   = 0 -> null
        - memory            = 0 -> null
        - memory_swap        = 0 -> null
        - must_run           = true -> null
        - name               = "foo" -> null
        - network_data        = [
            - {
                - gateway                  = "172.17.0.1"
                - global_ipv6_prefix_length = 0
                - ip_address                = "172.17.0.2"
                - ip_prefix_length          = 16
                - network_name               = "bridge"
                # (2 unchanged attributes hidden)
            },
        ] -> null
        - network_mode        = "bridge" -> null
        - privileged           = false -> null
        - publish_all_ports    = false -> null
        - read_only             = false -> null
        - remove_volumes       = true -> null
        - restart              = "no" -> null
        - rm                   = false -> null
        - runtime              = "runc" -> null
        - security_opts         = [] -> null
        - shm_size              = 64 -> null
        - start                 = true -> null
        - stdio_open             = false -> null
        - stop_timeout           = 0 -> null
        - storage_opts          = {} -> null
        - sysctls                = {} -> null
        - tmpfs                  = {} -> null
        - tty                   = false -> null
        # (8 unchanged attributes hidden)
    ]
}

# docker_image.ubuntu will be destroyed
- resource "docker_image" "ubuntu" {
```

```

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

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

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

Enter a value: yes

docker_container.foo: Destroying... [id=292de3d48673e0e7619d1826bdfcf93e34bfa8a9696e83b5f404b8fdff82309]
docker_container.foo: Destruction complete after 1s
docker_image.ubuntu: Destroying... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_image.ubuntu: Destruction complete after 0s

Destroy complete! Resources: 2 destroyed.

C:\Users\Sadneya\TerraformScripts\Docker>docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE

```

Docker images After Executing Destroy step

```

C:\Users\Sadneya\TerraformScripts\Docker>docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE

C:\Users\Sadneya\TerraformScripts\Docker>

```

EXPERIMENT NO:7

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

Procedure:

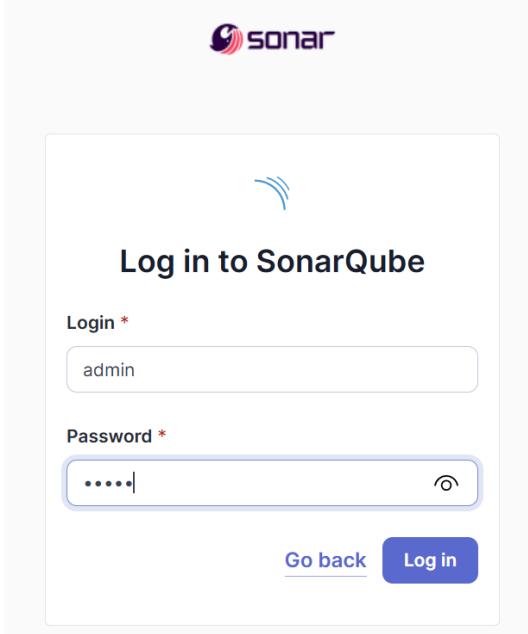
1. Creation of project on Sonarqube

1. Open the command prompt and perform this command
 (**Docker must be Installed before running this command.**)

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

```
PS C:\Users\Sadneya> 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
19948c8162691fd8be61d035e355c4fe4a6a2fd0a15237e94f75c0857ff7e2ff
```

2. Then after its successful execution, run the sonarqube at <http://localhost:9000>
3. Then Login as Username admin and password admin.



The screenshot shows the SonarQube interface for creating a new project. At the top, there are links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, More, and a search bar. Below the header, a section titled "How do you want to create your project?" lists several options:

- Import from Azure DevOps (Setup)
- Import from Bitbucket Cloud (Setup)
- Import from Bitbucket Server (Setup)
- Import from GitHub (Setup)
- Import from GitLab (Setup)

Below these, a button labeled "Create a local project" is visible. A warning message in a yellow box states: "Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine." At the bottom of the page, footer information includes "SonarQube™ technology is powered by SonarSource SA", "Community Edition v10.6 (92116) ACTIVE", "LGPL v3", "Community Documentation Plugins Web API".

4. Then create a project. Here I have given the name “sonarqube”. Keep branch name main only and then click on next.

The screenshot shows the "Create a local project" form. The "Project display name" field contains "sonarqube". The "Project key" field contains "sonarqube". The "Main branch name" field contains "main". Below the form, a note says "The name of your project's default branch [Learn More](#)". At the bottom, there are "Cancel" and "Next" buttons.

5. Then in the Setup project for clean as you code they will ask to choose the baseline for new code for this project. Choose Use the global setting. Then click on create project.

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](#)

- Then click on your account profile and select my account. Inside this click on security option.

A Administrator

[Profile](#) [Security](#) [Notifications](#) [Projects](#)

Profile

Login: admin

Groups

- sonar-administrators
- sonar-users

SCM Accounts ?

- admin

Preferences

Enable Keyboard Shortcuts

Some actions can be performed using keyboard shortcuts. If you do not want to use these shortcuts, you can disable them here (this won't disable navigation shortcuts, which include the arrows, escape, and enter keys). For a list of available keyboard shortcuts, use the question mark shortcut (hit ? on your keyboard).

- Then set the name of your token and here I have chosen the global scope of the token and then clicked on generate token, thus token created successfully.(Here I have given my token name as 'jenkins token').

Administrator

Profile Security Notifications Projects

a User Token is a replacement for the user login. This will increase the security of your installation by not letting your analysis user's password going through your network.

Generate Tokens

Name	Type	Expires in
Enter Token Name	Select Token Type	30 days
Generate		

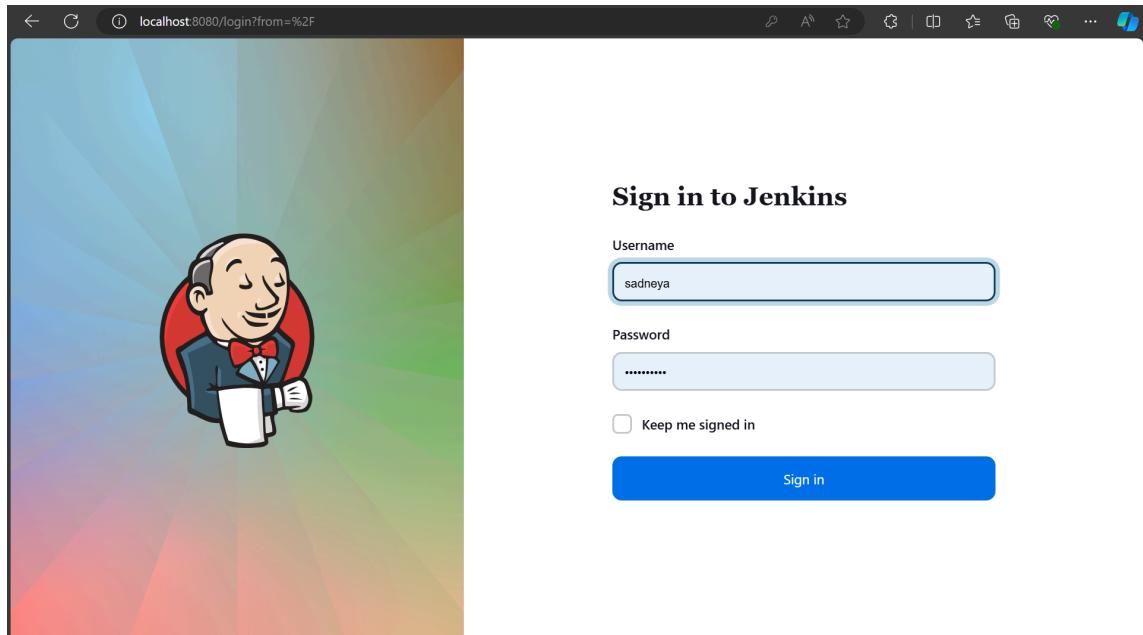
New token "Jenkins Token" has been created. Make sure you copy it now, you won't be able to see it again!

Name	Type	Project	Last use	Created	Expiration
sqa_ab7e392ec51669ad950952a9bdbd2dbac3b7498e	Global		Never	September 21, 2024	October 21, 2024
					Revoke

****Copy and paste this token as it will be used ahead in Jenkins.****

2. Creation Freestyle project on Jenkins

- Run the jenkins at <http://localhost:8080> and enter username and password and click on sign in



- Then Click on manage Jenkins.

3. Here click on Plugins.Inside Available Plugins search for SonarQube Scanner and then click on Install.Then again restart Jenkins.

4. Then go back to Manage Jenkins and click on system and search for sonarqube server.

SonarQube servers

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

Environment variables

SonarQube installations

List of SonarQube installations

Name	sonarcube local	X
Server URL	Default is http://localhost:9000 http://localhost:9000	
Server authentication token	SonarQube authentication token. Mandatory when anonymous access is disabled. - none - + Add ▾	
Advanced ▾		

5. Click on the environment variables and give name to SonarQube installation.(here I have given the name as “sonarqube local”) then give the server URL as <https://localhost:9000> then save and apply these changes.
6. Then select item type Freestyle Project and give name to your project here I have given name “advdevops project”.Then click on ok.

localhost:8080/view/all/newJob

Jenkins

Search (CTRL+K) | ? | Log out

Dashboard > All > New Item

New Item

Enter an item name
advdevops project

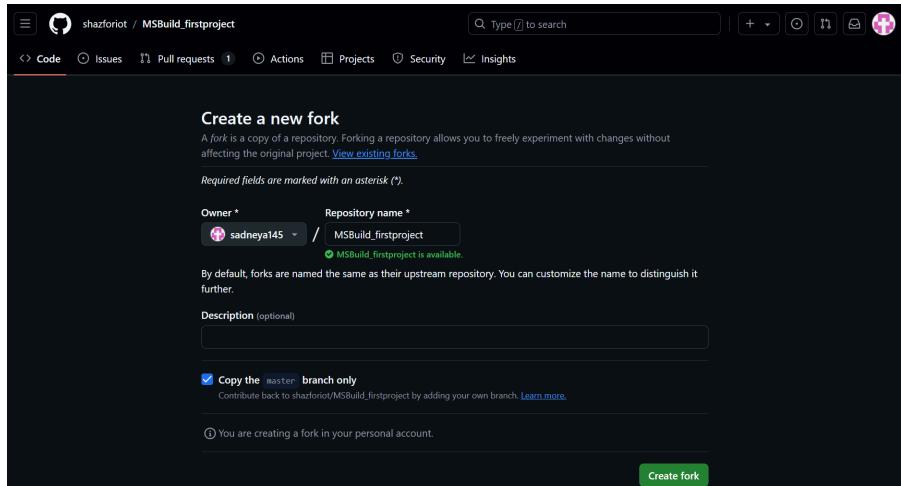
Select an item type

- 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, folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

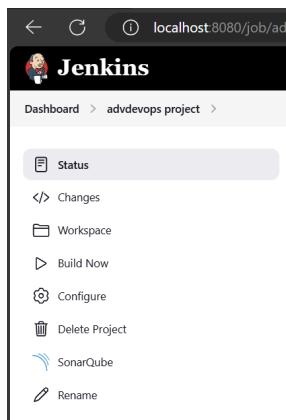
OK

7. Choose the Github repository : https://github.com/shazforiot/MSBuild_firstproject.git

It is a sample hello-world project with no vulnerabilities and issues. It is just for testing purposes.
Fork This Repository.



- Then click on your project and then go to configure.



- Then enable Git and add that github Repository URL.select */master branch.



10. Go to Sonarqube http://localhost:9000/<user_name>/permissions.

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

4 of 4 shown

Then allow execute permissions to Admin User.

Group	Description	Administer System	Administer	Execute Analysis	Create
Administrator admin		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

4 of 4 shown

11. Then again Go inside the project and click on configure (addevops project->configure). Go there in the build steps section. Inside It add the analysis properties.

```
sonar.host.url=http://localhost:9000 (Your_url)
sonar.login=sqa_0d6700917b11bb2a57f2cc2189e162f6bbf26929 (login_token that You have copied on clipboard created in step 7 of creation of project of sonarqube)
sonar.sources=/
sonar.projectKey=sonarqube(project name)
```

Then click on save and apply.

Configure

Build Steps

Execute SonarQube Scanner

JDK (Inherit From Job)

Path to project properties

```
sonar.host.url=http://localhost:9000
sonar.login=sqa_0d6700917b11bb2a57f2cc2189e162f6bbf26929
sonar.sources=.
sonar.projectKey=sonarqube
```

Additional arguments

JVM Options

Add build step ▾

Save Apply

12. Then again go into manage Jenkins and check the number of executors if it is zero then set to two or more. If it is zero then it will not build successfully and give error.

Maven Project Configuration

Global MAVEN_OPTS

Local Maven Repository (Default: ~/m2/repository)

of executors 0

Labels

Usage (Use this node as much as possible)

Quiet period 5

SCM checkout retry count 0

Restrict project naming

Save Apply

In the above image it was zero that's why my build was getting unsuccessful. Then I declared the value of # of executors to 4.

The screenshot shows the Jenkins System configuration page at localhost:8080/manage/configure. The 'System' section is selected. Under 'Maven Project Configuration', the '# of executors' field is set to '4'. Other fields like 'Global MAVEN_OPTS' and 'Local Maven Repository' are also visible.

13. Then click on the build project. Thus the build is successful. see the console output.

The screenshot shows the Jenkins dashboard for the 'advdevops project'. The 'Status' tab is active. The 'Build History' section shows a single build (#1) from September 21, 2024, at 12:40 PM. The status of this build is 'Success'. Other tabs like 'Changes', 'Workspace', and 'Configure' are also visible.

localhost:8080/job/advdevops%20project/1/console

Dashboard > advdevops project > #1 > Console Output

Status Changes Console Output Edit Build Information Delete build #1 Timings Git Build Data

Console Output

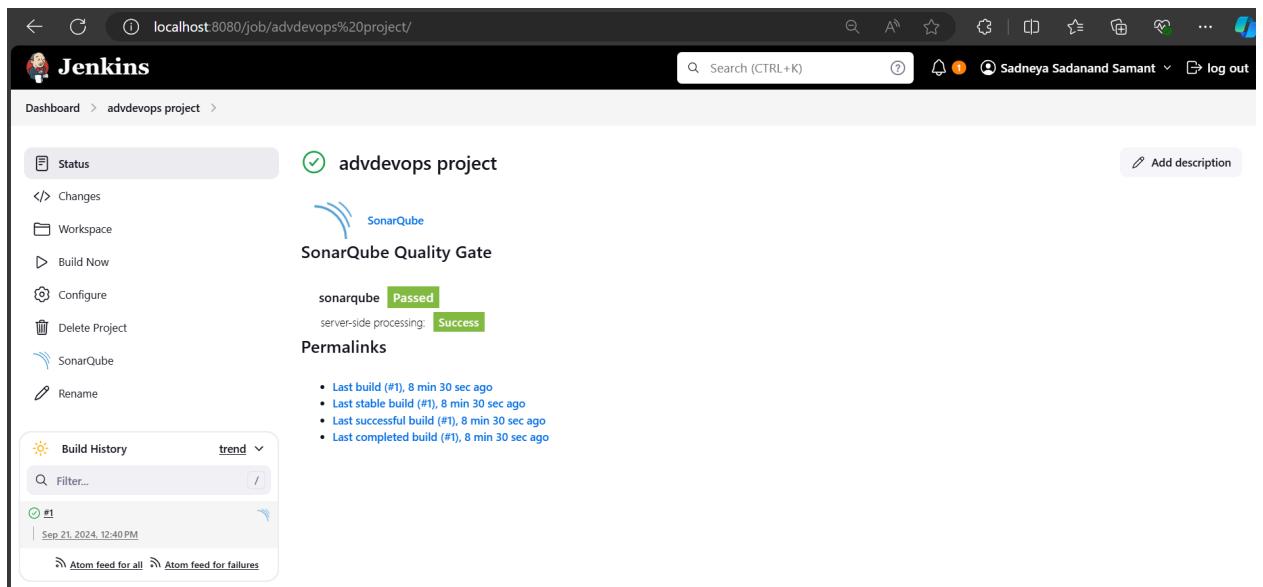
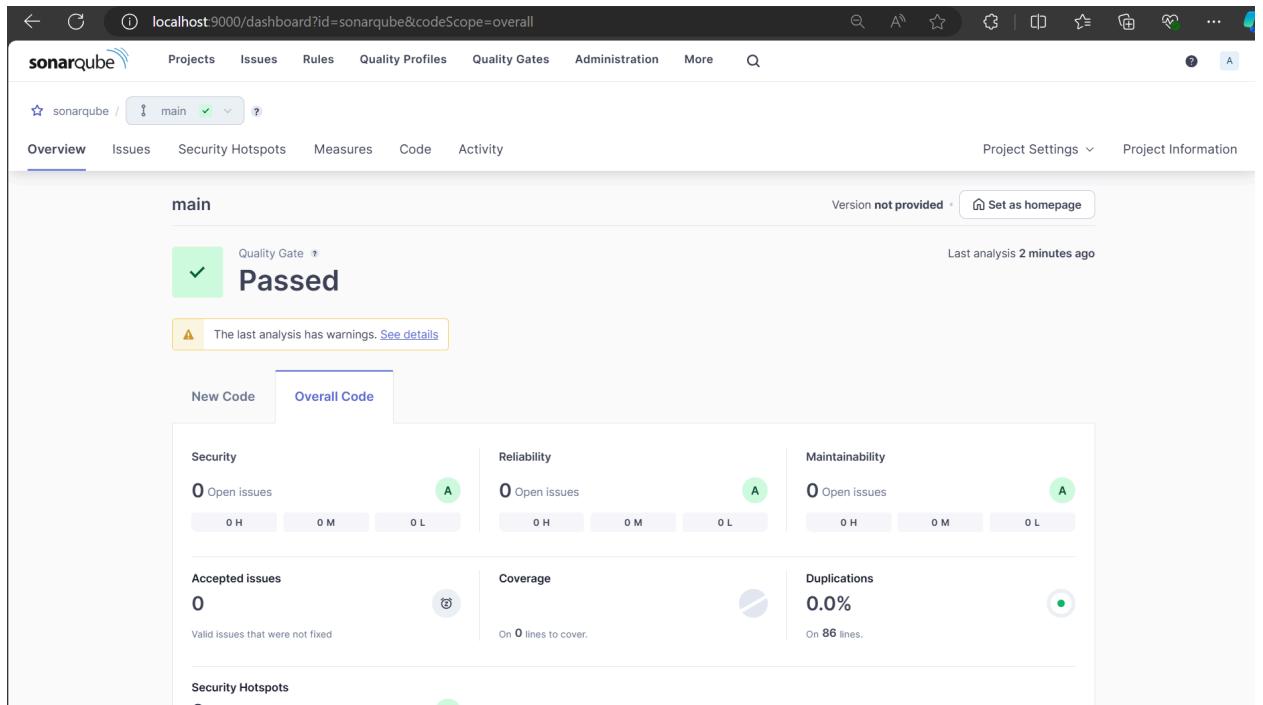
Started by user Sadneya Sadanand Samant
Running as SYSTEM
Building on the built-in node in workspace C:\ProgramData\Jenkins\jenkins\workspace\advdevops project
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/shazforiot/MSBuild_FirstProject
> git.exe init C:\ProgramData\Jenkins\jenkins\workspace\advdevops project # timeout=10
Fetching upstream changes from https://github.com/shazforiot/MSBuild_FirstProject
> git.exe --version # timeout=10
> git.exe config core.sparsecheckout # timeout=10
> git.exe config remote.origin.url https://github.com/shazforiot/MSBuild_FirstProject # timeout=10
> git.exe config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision f2bc042c04c6e72427c380bcaee6d6fee7b49adf (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f f2bc042c04c6e72427c380bcaee6d6fee7b49adf # timeout=10
Commit message: "updated"
First time build. Skipping changelog.
Unpacking https://repo1.maven.org/maven2/org/sonarsource/scanner/cli/sonar-scanner-cli/6.2.0.4584/sonar-scanner-cli-6.2.0.4584.zip to C:\ProgramData\Jenkins\jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\SonarQube_Scanner on Jenkins [advdevops project] \$ C:\ProgramData\Jenkins\jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\SonarQube_Scanner\bin\sonar-scanner.bat -Dsonar.host.url=http://localhost:9000 ***** -Dsonar.projectKey=sonarqube -Dsonar.host.url=http://localhost:9000 -Dsonar.login=sqa_767ae4fae7c246c2c3d382e75d72abe91d328d87 -Dsonar.sources=../ -Dsonar.projectBaseDir=C:\ProgramData\Jenkins\jenkins\workspace\advdevops project"
12:44:58.811 WARN Property 'sonar.host.url' with value 'http://localhost:9000' is overridden with value 'http://localhost:9000'
12:44:58.826 INFO Scanner configuration file:
C:\ProgramData\Jenkins\jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\SonarQube_Scanner\bin..\conf\sonar-scanner.properties
12:44:58.826 INFO Project root configuration file: NONE
12:45:36.578 INFO 14 source files to be analyzed
12:45:36.578 INFO 14/14 source files have been analyzed
12:45:36.594 INFO Sensor TextAndSecretsSensor [text] (done) | time=2132ms
12:45:36.605 INFO ----- Run sensors on project
12:45:36.814 INFO Sensor C# [csharp]
12:45:36.814 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>
12:45:36.814 INFO Sensor C# [csharp] (done) | time=10ms
12:45:36.814 INFO Sensor Analysis Warnings import [csharp]
12:45:36.814 INFO Sensor Analysis Warnings import [csharp] (done) | time=0ms
12:45:36.814 INFO Sensor C# File Caching Sensor [csharp]
12:45:36.814 WARN Incremental PR analysis: Could not determine common base path, cache will not be computed. Consider setting 'sonar.projectBaseDir' property.
12:45:36.814 INFO Sensor C# File Caching Sensor [csharp] (done) | time=0ms
12:45:36.814 INFO Sensor Zero Coverage Sensor
12:45:36.846 INFO Sensor Zero Coverage Sensor (done) | time=32ms
12:45:36.846 INFO SCM Publisher SCM provider for this project is: git
12:45:36.861 INFO SCM Publisher 4 source files to be analyzed
12:45:39.311 INFO SCM Publisher 4/4 source files have been analyzed (done) | time=2443ms
12:45:39.311 INFO CPD Executor Calculating CPD for 0 files
12:45:39.420 INFO CPD Executor CPD calculation finished (done) | time=0ms
12:45:39.436 INFO SCM revision ID 'f2bc042c04c6e72427c380bcaee6d6fee7b49adf'
12:45:39.782 INFO Analysis report generated in 267ms, dir size=199.9 kB
12:45:39.892 INFO Analysis report compressed in 110ms, zip size=22.4 kB
12:45:42.561 INFO Analysis report uploaded in 2661ms
12:45:42.563 INFO ANALYSIS SUCCESSFUL, you can find the results at: <http://localhost:9000/dashboard?id=sonarqube>
12:45:42.563 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
12:45:42.564 INFO More about the report processing at <http://localhost:9000/api/ce/task?id=b041298d-55f9-4005-844f-c6fce8ccc151>
12:45:42.580 INFO Analysis total time: 28.328 s
12:45:42.583 INFO SonarScanner Engine completed successfully
12:45:42.637 INFO EXECUTION SUCCESS
12:45:42.637 INFO Total time: 43.826s
Finished: SUCCESS

14. Then go to SonarQube. Then go inside the project that you created. where it will show output passed.

Name:Sadneya Sadanand Samant

Roll No:46

Advdevops_07



Thus Project Build successfully.

Conclusion: Here we created the sonarqube project locally successfully. Then created a freestyle project on jenkins and installed a plugin named sonarqube scanner. Then we build that project by firstly adding github repository and mentioning Analysis properties which is nothing but name ,key, token. Then after saving and applying the changes, the build executed successfully. Then we also checked that sonarqube given response about passing of analysis. Thus Build is done successfully.

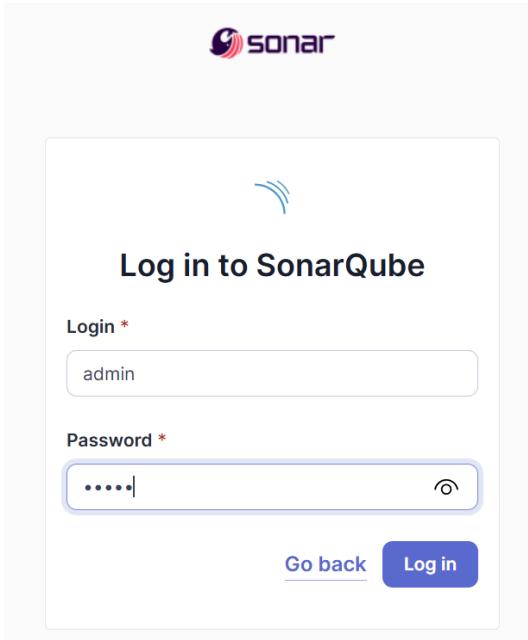
Aim: Create a Jenkins CICD Pipeline with SonarQube / GitLab Integration to perform a static analysis of the code to detect bugs, code smells, and security vulnerabilities on a sample Web / Java / Python application.

1. Creation of project on Sonarqube

1. Open the command prompt and perform this command
(Docker must be Installed before running this command.**)**
docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest

```
PS C:\Users\Sadneya> 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
19948c8162691fd8be61d035e355c4fe4a6a2fd0a15237e94f75c0857ff7e2ff
```

2. Then after its successful execution, run the sonarqube at <http://localhost:9000>
3. Then Login as Username admin and password admin.



The screenshot shows the SonarQube interface at localhost:9000/projects/create. The top navigation bar includes links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, More, and a search bar. Below the navigation is a section titled "How do you want to create your project?". It asks if the user wants to benefit from SonarQube's features like repository import and Pull Request decoration. It then prompts the user to set up a DevOps platform configuration. There are six buttons for importing from Azure DevOps, Bitbucket Cloud, Bitbucket Server, GitHub, GitLab, and a "Create a local project" button. A warning message in a yellow box states: "Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine." At the bottom, it says SonarQube™ technology is powered by SonarSource SA, and provides links for Community Edition v10.6 (92116) ACTIVE, LGPL v3, Community, Documentation, Plugins, and Web API.

- Then create a project. Here I have given the name as “sonar46”. Keep branch name main only and then click on next.

The screenshot shows the "Create a local project" step in the SonarQube setup process. It is the first of two steps, indicated by "1 of 2" at the top left. The form fields are: "Project display name" (sonar46_), "Project key" (sonar46_), and "Main branch name" (main). Below the form is a note: "The name of your project's default branch [Learn More](#)". At the bottom are "Cancel" and "Next" buttons.

- Then in the Setup project for clean as you code they will ask to choose the baseline for new code for this project. Choose Use the global setting.Then click on create project.

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](#)

- Then click on your account profile and select my account.Inside this click on security option.

A Administrator

Profile Security Notifications Projects

Profile

Login: admin

Groups

sonar-administrators
sonar-users

SCM Accounts ?

admin

Preferences

Enable Keyboard Shortcuts

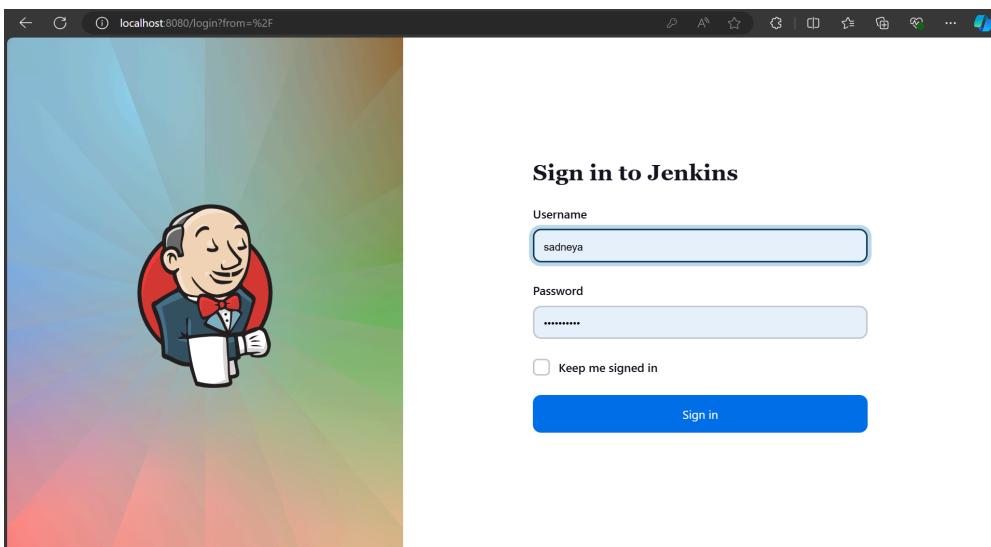
Some actions can be performed using keyboard shortcuts. If you do not want to use these shortcuts, you can disable them here (this won't disable navigation shortcuts, which include the arrows, escape, and enter keys). For a list of available keyboard shortcuts, use the question mark shortcut (hit [?](#) on your keyboard).

- Then set the name of your token and here I have chosen the global scope of the token and then clicked on generate token, thus token created successfully.(Here I have given my token name as 'jenkins token').

<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> ✔ New token "Jenkins Token" has been created. Make sure you copy it now, you won't be able to see it again! </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Name</th><th>Type</th><th>Project</th><th>Last use</th><th>Created</th><th>Expiration</th></tr> </thead> <tbody> <tr> <td>Jenkins Token</td><td>Global</td><td></td><td>Never</td><td>September 21, 2024</td><td>October 21, 2024</td></tr> </tbody> </table>						Name	Type	Project	Last use	Created	Expiration	Jenkins Token	Global		Never	September 21, 2024	October 21, 2024
Name	Type	Project	Last use	Created	Expiration												
Jenkins Token	Global		Never	September 21, 2024	October 21, 2024												
Revoke																	

2. Creation and building of jenkins pipeline

- Run the jenkins at <http://localhost:8080> and enter username and password and click on sign in



- Then create a pipeline item type in jenkins.

New Item

Enter an item name
advpipeline

Select an item type

- 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.
- Freestyle project
- Maven project
- Multi-configuration project
- Folder
- Multibranch Pipeline

OK

3. Go to [SonarScanner CLI \(sonarsource.com\)](https://sonarsource.com/sonar-scanner-cli) this website and download the latest version of sonar scanner on the computer.
4. Then select the machine here I am using windows x64

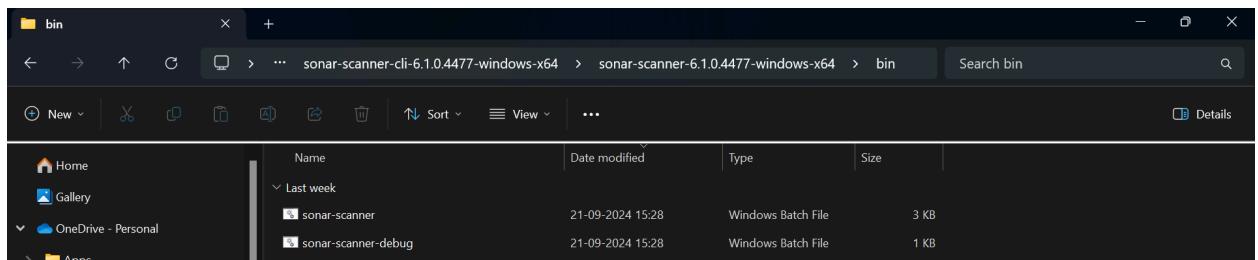
[Latest](#) | Analyzing source code | Scanners | SonarScanner CLI

SonarScanner CLI

The screenshot shows the official SonarScanner CLI release page. At the top, there are navigation links for 'SonarScanner' and 'Issue Tracker', and a 'Show more' dropdown. Below this, the version '6.2' is displayed next to the release date '2024-09-17'. A brief description states 'Support PKCS12 truststore generated with OpenSSL'. Below the description are download links for 'Linux x64', 'Linux AArch64', 'Windows x64', 'macOS x64', 'macOS AArch64', 'Docker', and 'Any (Requires a pre-installed JVM)'. At the bottom of the card is a 'Release notes' link.

Once it is downloaded, copy its path as it is required in the pipeline script.

C:\Users\Sadneya\Downloads\sonar-scanner-cli-6.1.0.4477-windows-x64\sonar-scanner-6.1.0.4477-windows-x64\bin\sonar-scanner.bat



5. Inside the pipeline script section write the following script.

```
node {
    stage('Cloning the GitHub Repo') {
        git "https://github.com/shazforiot/GOL.git" (cloning this repo)
    }
    stage('SonarQube analysis') {
        withSonarQubeEnv('sonarqube') {
            bat """

```

```
"C:\Users\Sadneya\Downloads\sonar-scanner-cli-6.1.0.4477-windows-x64\sonar-scanner-6.1.0.4477-windows-x64\bin\sonar-scanner.bat" ^
-Dsonar.login=sq...fab91d70fc5db32bb83641a23decbbf04470ba2d ^ (login token)
```

Name: Sadneya Sadanand Samant

Roll No:46

Adv_devops_07

-Dsonar.projectKey=sonar46 ^ (your project name)
-Dsonar.exclusions=vendor/**,resources/**,**/*.java ^ (libraries u want to exclude)

-Dsonar.host.url=http://localhost:9000 (server URL)
""""

```
}
```

```
}
```

```
}
```

The screenshot shows the Jenkins Pipeline configuration page for a job named 'soanrpipeline'. The 'Advanced Project Options' tab is selected. The pipeline script is defined as follows:

```
node {
    stage('Cloning the GitHub Repo') {
        git "https://github.com/shazforiot/GOL.git"
    }
    stage('SonarQube analysis') {
        withSonarQubeEnv('sonarqube') {
            bat """
                C:\Users\Sadneya\Downloads\sonar-scanner-cli-6.1.0.4477-windows-x64\sonar-scanner-6.1.0.4477-windows-x64\bin\sonar-scanner.bat
                -Dsonar.login=sup_fab91d7a7c5db32bb83641a23decbf04470ba2d
                -Dsonar.projectKey=sonar46
                -Dsonar.exclusions=vendor/**,resources/**,**/*.java
                -Dsonar.host.url=http://localhost:9000
            """
        }
    }
}
```

The 'Use Groovy Sandbox' checkbox is checked. At the bottom, there are 'Save' and 'Apply' buttons.

6. Then save and apply the changes.Then click on build option.

The screenshot shows the Jenkins dashboard for the 'advdevops project'. The left sidebar includes options like Status, Changes, Workspace, Build Now, Configure, Delete Project, SonarQube, and Rename.

Then it throws an error.

Console Output

```

Started by user Sadneya Sadanand Samant
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\soanrpipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Cloning the GitHub Repo)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/shazforiot/GOL.git
> git.exe init C:\ProgramData\Jenkins\.jenkins\workspace\soanrpipeline # timeout=10
Fetching upstream changes from https://github.com/shazforiot/GOL.git
> git.exe --version # timeout=10
> git --version # 'git' version 2.43.0.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/shazforiot/GOL.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe config remote.origin.url https://github.com/shazforiot/GOL.git # timeout=10
> git.exe config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10

17:42:26.064 INFO  INFO [main] jenkins.Workspace - Provisioning: Using workspace: C:\ProgramData\Jenkins\workspace\soanrpipeline
17:42:26.066 INFO  EXECUTION FAILURE
17:42:26.066 INFO  Total time: 0.766s
17:42:26.067 ERROR Error during SonarScanner CLI execution
java.lang.IllegalStateException: Error status returned by url [http://localhost:9000/api/v2/analysis/jres?os=windows&arch=amd64]: 401
    at org.sonarsource.scanner.lib.internal.http.ServerConnection.callUrl(ServerConnection.java:182)
    at org.sonarsource.scanner.lib.internal.http.ServerConnection.callApi(ServerConnection.java:145)
    at org.sonarsource.scanner.lib.internal.http.ServerConnection.callRestApi(ServerConnection.java:123)
    at org.sonarsource.scanner.lib.internal.JavaRunnerFactory.getJreMetadata(JavaRunnerFactory.java:159)
    at org.sonarsource.scanner.lib.internal.JavaRunnerFactory.getJreFromServer(JavaRunnerFactory.java:138)
    at org.sonarsource.scanner.lib.internal.JavaRunnerFactory.createRunner(JavaRunnerFactory.java:85)
    at org.sonarsource.scanner.lib.internal.ScannerEngineLauncherFactory.createLauncher(ScannerEngineLauncherFactory.java:53)
    at org.sonarsource.scanner.lib.ScannerEngineBootstrapper.bootstrap(ScannerEngineBootstrapper.java:118)
    at org.sonarsource.scanner.cli.Main.analyze(Main.java:75)
    at org.sonarsource.scanner.cli.Main.main(Main.java:63)
17:42:26.068 ERROR
17:42:26.068 ERROR Re-run SonarScanner CLI using the -X switch to enable full debug logging.
[Pipeline] }
WARN: Unable to locate 'report-task.txt' in the workspace. Did the SonarScanner succeed?
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
ERROR: script returned exit code 1
Finished: FAILURE

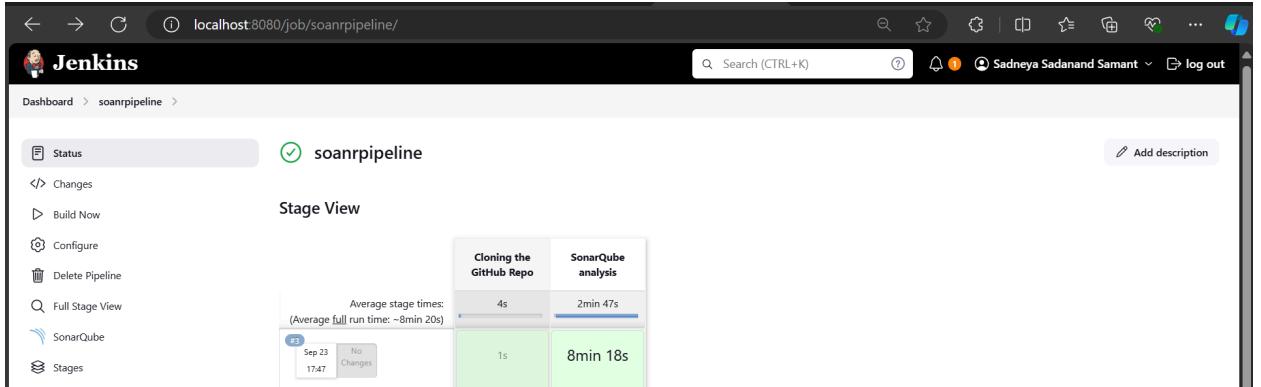
```

This error was due to failure in login and connection. So I generated the token again and replaced it in the pipeline script inside the configure of the pipeline.

ie. -Dsonar.login=(new token)

7. Then I again clicked on build ,but this time the build got executed successfully.

3. Analysis by sonarqube



8. Check the Console Output.

The screenshot shows the Jenkins Pipeline console output for build #3. The left sidebar includes options like Status, Changes, Console Output (which is selected), View as plain text, Edit Build Information, Delete build '#3', Timings, Git Build Data, Pipeline Overview, Pipeline Console, Replay, Pipeline Steps, Workspaces, and Previous Build. The main content area is titled 'Console Output' and shows the following log entries:

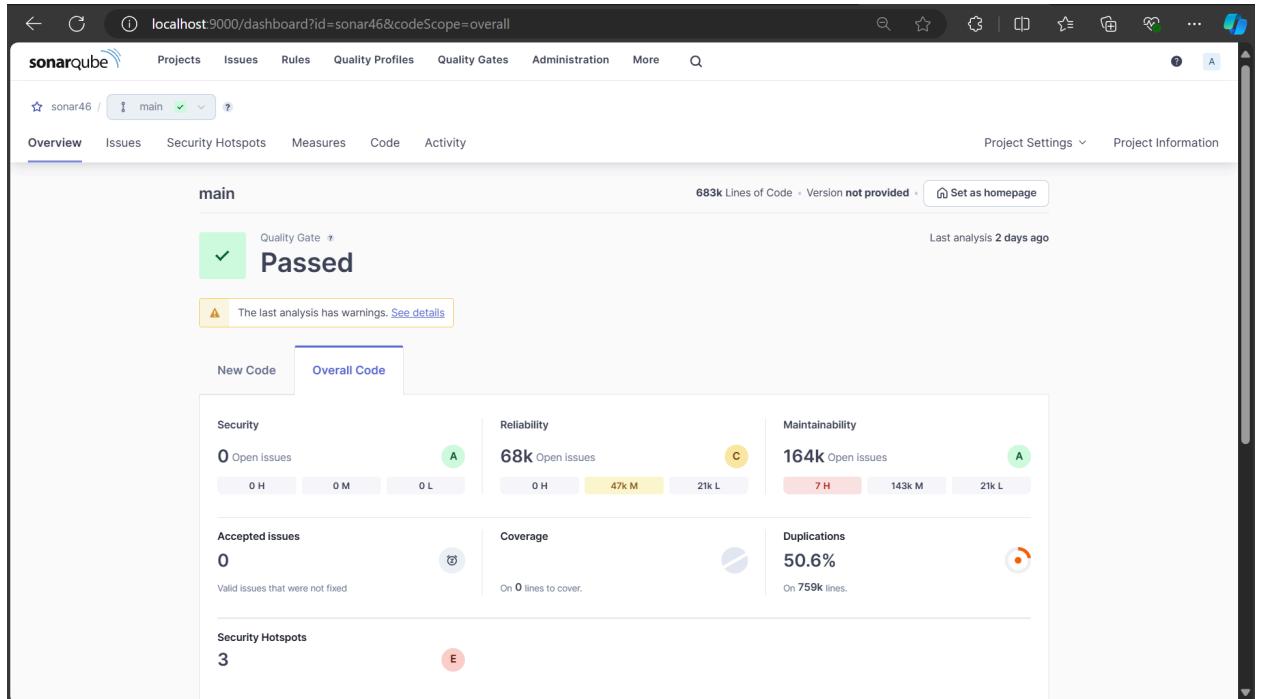
```

Skipping 4.248 KB. Full Log
17:54:55.087 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/GuiPackage.html for block at line 567. Keep only the first 100 references.
17:54:55.087 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/GuiPackage.html for block at line 568. Keep only the first 100 references.
17:54:55.087 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/GuiPackage.html for block at line 571. Keep only the first 100 references.
17:54:55.087 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/GuiPackage.html for block at line 1485. Keep only the first 100 references.
17:54:55.087 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/GuiPackage.html for block at line 1487. Keep only the first 100 references.
17:54:55.087 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/GuiPackage.html for block at line 571. Keep only the first 100 references.
17:54:55.087 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/GuiPackage.html for block at line 32. Keep only the first 100 references.
17:54:55.087 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/GuiPackage.html for block at line 567. Keep only the first 100 references.
17:54:55.087 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/GuiPackage.html for block at line 568. Keep only the first 100 references.
17:54:55.087 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/GuiPackage.html for block at line 571. Keep only the first 100 references.
17:54:55.087 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/GuiPackage.html for block at line 40. Keep only the first 100 references.

Keep only the first 100 references.
17:55:00.057 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/sampler/gui/TestActionGui.html for block at line 154. Keep only the first 100 references.
17:55:00.057 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/sampler/gui/TestActionGui.html for block at line 555. Keep only the first 100 references.
17:55:00.057 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/sampler/gui/TestActionGui.html for block at line 154. Keep only the first 100 references.
17:55:00.057 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/sampler/gui/TestActionGui.html for block at line 555. Keep only the first 100 references.
17:55:00.072 INFO CPD Executor CPD calculation finished (done) | time=150691ms
17:55:00.338 INFO SCM revision ID 'ba799ba7e1b576f04a4612322b0412c5e6e1e5e4'
17:55:08.022 INFO Analysis report generated in 4925ms, dir size=127.2 MB
17:55:27.419 INFO Analysis report compressed in 19397ms, zip size=29.5 MB
17:55:30.386 INFO Analysis report uploaded in 2962ms
17:55:30.397 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonar46
17:55:30.397 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
17:55:30.397 INFO More about the report processing at http://localhost:9000/api/ce/task?id=b0639869-14e8-402a-b7e6-7bd8be121405
17:55:46.751 INFO Analysis total time: 8:11.934 s
17:55:46.771 INFO SonarScanner Engine completed successfully
17:55:47.395 INFO EXECUTION SUCCESS
17:55:47.520 INFO Total time: 8:15.609s
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

9. Then go to SonarQube. Then go inside the project that you created. where it will show output passed.



Thus Project Build successfully.

4. Analysis by sonarqube after changing code

1. Then I made the changes in the [pom.xml](#) file to see the analysis given by sonarqube.
In these I have introduced the hardcoded credentials and an unused variable:
2. I have changed the properties in the [pom.xml](#) from

```

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.wakaleo.gameoflife</groupId>
    <artifactId>gameoflife</artifactId>
    <version>1.0-SNAPSHOT</version>
    <packaging>pom</packaging>
    <name>gameoflife</name>
    <url>https://github.com/wakaleo/game-of-life</url>
    <properties>
        <build.number>SNAPSHOT</build.number>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <easyb.version>1.4</easyb.version>
        <cobertura.version>2.6</cobertura.version>
        <!-- A workaround for a bug in PMD -->
        <sourceJdk>1.8</sourceJdk>
        <targetJdk>1.8</targetJdk>
        <github.account>wakaleo</github.account>
        <thucydides.version>1.8.4</thucydides.version>
        <jelastic.context>gameoflife</jelastic.context>
        <jelastic.environment>wakaleo</jelastic.environment>
    </properties>
    <scm>
        <connection>scm:git:git@github.com:${github.account}/game-of-life.git</connection>
        <developerConnection>scm:git:git@github.com:${github.account}/game-of-life.git</developerConnection>
    </scm>

```

To this

<properties>

```

<build.number>SNAPSHOT</build.number>
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
<easyb.version>1.4</easyb.version>
<cobertura.version>2.6</cobertura.version>
<!-- A workaround for a bug in PMD -->
<sourceJdk>1.8</sourceJdk>
<targetJdk>1.8</targetJdk>
<github.account>wakaleo</github.account>
<thucydides.version>1.8.4</thucydides.version>
<jelastic.context>gameoflife</jelastic.context>
<jelastic.environment>wakaleo</jelastic.environment>

<!-- Hardcoded username and password (Security issue) -->
<jelastic.username>admin</jelastic.username>
<jelastic.password>password123</jelastic.password>
```

```

<!-- Unused property (Maintainability issue) -->
<unused.property>ThisIsUnused</unused.property>
</properties>
```

3. Then again I build the code on jenkins.

The screenshot shows the Jenkins Pipeline interface for the 'soanrpipeline'. At the top, there's a navigation bar with a Jenkins logo and the text 'soanrpipeline'. Below it, a sidebar on the left lists pipeline management options: Status (selected), Changes, Build Now, Configure, Delete Pipeline, Full Stage View, SonarQube, and Stages. The main area is titled 'Stage View' and displays the execution details for each stage of the pipeline. The stages are listed in a grid:

Cloning the GitHub Repo	SonarQube analysis
4s	18min 57s
3s	1h 13min

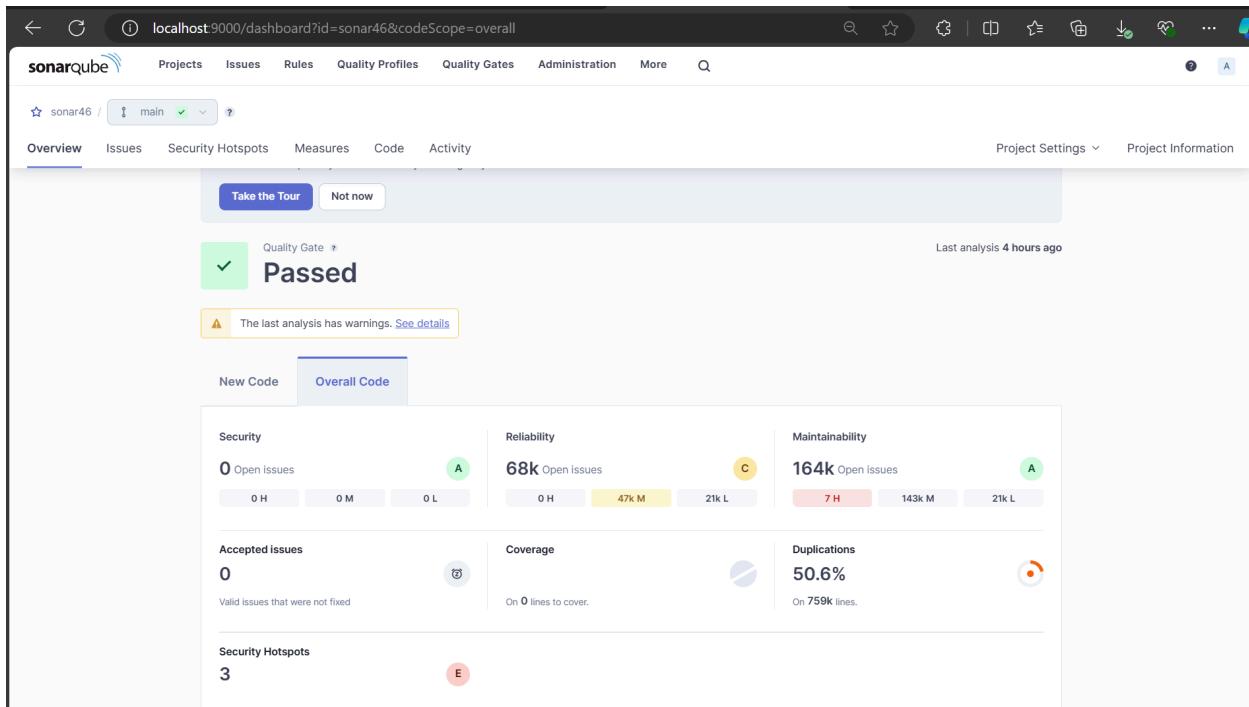
Below the stages, a summary indicates an average stage time of 38s and an average full run time of approximately 31 minutes. The pipeline ID is #5, and the commit date is Sep 30 at 15:44.

4. Then I again go to the sonarqube to see the analysis it given me output passed

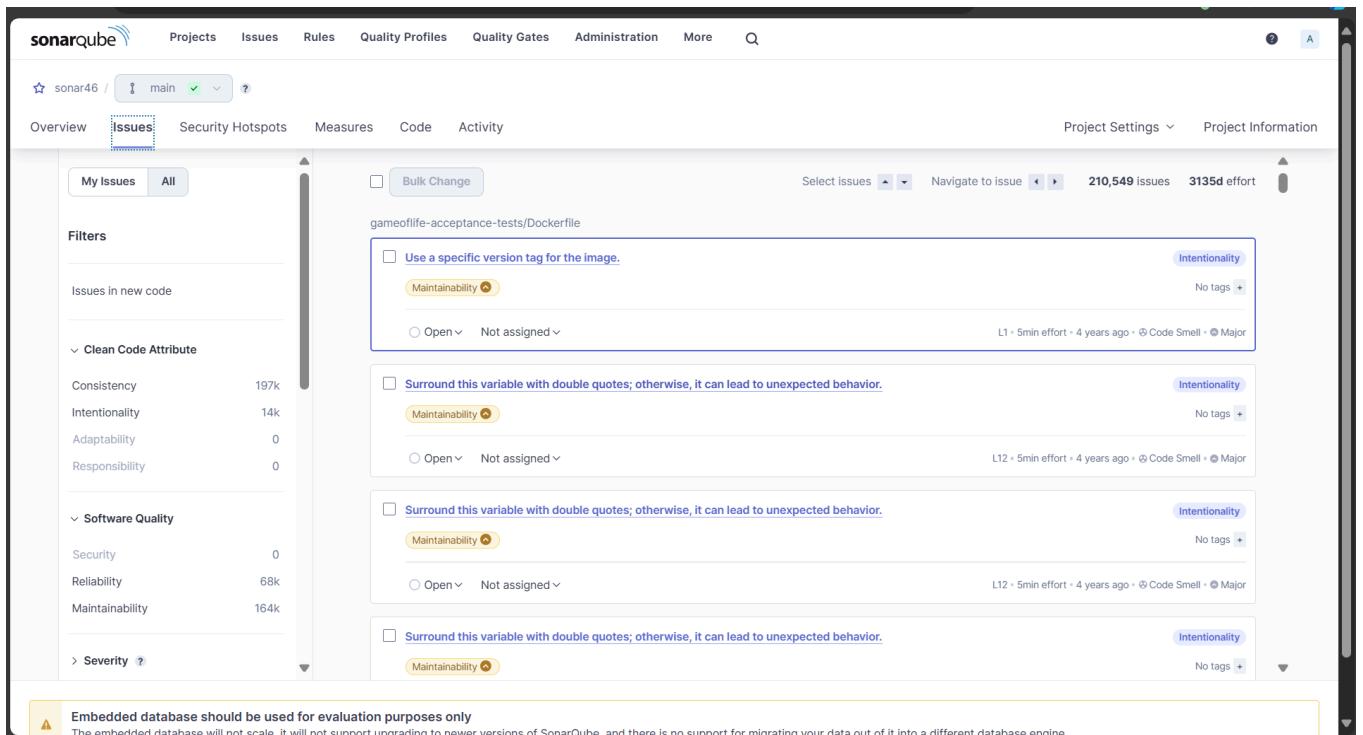
Name: Sadneya Sadanand Samant

Roll No:46

Adv_devops_07



But in issues section there are many issues after making changes.



Thus sonarqube analysis is done.

Conclusion: Here we created the “sonarpipeline” project locally successfully. Then created a pipeline on jenkins and installed a plugin named sonarqube scanner. Then we build that project by adding the script inside pipeline which consist of project name ,key, token and then github repository to clone. Then after saving and applying the changes, the build executed successfully. Initially there was an issue in connection to solve this I created the token again and it executed successfully. Then we also checked that sonarqube gives response as passed. Thus, Build is done successfully. Then again I made some changes in pom.xml and again built the code which was also done successfully and then I opened sonarqube which gave me detailed analysis of issues arised due to new changes in code.

Experiment No:9

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

Steps:

Go to AWS ACADEMY.

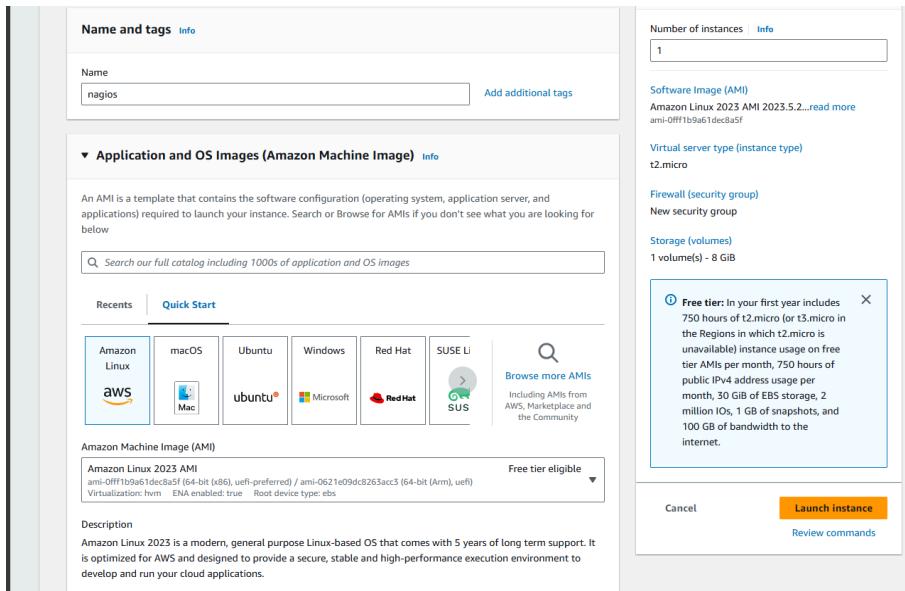
On Dashboard, go to EC-2 instance.

The screenshot shows the AWS EC2 Dashboard for the US East (N. Virginia) Region. The left sidebar includes links for EC2 Global View, Events, Console-to-Code Preview, Instances, Images, Elastic Block Store, Network & Security, and Load Balancing. The main content area displays the following information:

- Resources:** Instances (running) 0, Auto Scaling Groups 0, Capacity Reservations 0, Dedicated Hosts 0, Elastic IPs 0, Instances 0, Key pairs 2, Load balancers 0, Placement groups 0, Security groups 5, Snapshots 0, Volumes 0.
- Launch instance:** A button to "Launch instance" and a link to "Migrate a server". Note: Your instances will launch in the US East (N. Virginia) Region.
- Service health:** Shows "AWS Health Dashboard" and indicates "This service is operating normally".
- Zones:** A table of Zone names and Zone IDs:

Zone name	Zone ID
us-east-1a	use1-az1
us-east-1b	use1-az2
us-east-1c	use1-az4
us-east-1d	use1-az6
us-east-1e	use1-az3
us-east-1f	use1-az5
- EC2 Free Tier Info:** Offers for all AWS Regions. 0 EC2 free tier offers in use. End of month forecast: 0 offers forecasted to exceed free tier limit. Exceeds free tier: 0 offers exceeded and is now pay-as-you-go pricing. A link to "View all AWS Free Tier offers".
- Account attributes:** Default VPC (vpc-04a6482e2565a490), Settings (Data protection and security, Zone, EC2 Serial Console, Default credit specification, EC2 console preferences).
- Explore AWS:** A section with 10 tips to reduce AWS costs, including "Explore how to effectively manage your AWS costs without compromising on performance or capacity". A link to "Learn more".

1. Creation Of EC2 instance: Create an Amazon Linux EC2 Instance of type t2.micro in AWS and name it - nagios-host



2. Select the existing Key pair or create a new pair.

Key pair (login) Info

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

Key pair name - required

 Create new key pair

Network settings Info

Network Info

vpc-051bba342b3626898

Subnet Info

No preference (Default subnet in any availability zone)

Auto-assign public IP Info

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) Info

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

Create security group Select existing security group

Common security groups Info

Select security groups

Nagios sg-0b1355e80625c05ee X

VPC vpc-051bba342b3626898

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

3. Go back to the EC2 Dashboard there on the left pane, select the security group.

▼ Network & Security

Security Groups

Elastic IPs

Placement Groups

Key Pairs

Network Interfaces

Under Security Group, create a new security group, Give a description to it. Then edit the inbound rules of the specified Security Group for this. add HTTP at port 80, HTTPS at port 443, SSH at port 22, ICMP are open from everywhere.

sg-0b1355e80625c05ee - Nagios

Details

Security group name Nagios	Security group ID sg-0b1355e80625c05ee	Description nagios use	VPC ID vpc-051bb342b3626898
Owner 425001375268	Inbound rules count 7 Permission entries	Outbound rules count 1 Permission entry	

Inbound rules | Outbound rules | Tags

Inbound rules (7)

Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
-	sgr-086d9781937f33d...	IPv4	SSH	TCP	22	0.0.0.0/0	-
-	sgr-0537216a5b2a49...	IPv6	All ICMP - IPv6	IPv6 ICMP	All	::/0	-
-	sgr-0073f2b189e0214e0	IPv4	HTTPS	TCP	443	0.0.0.0/0	-
-	sgr-0d667517b1040d..	IPv4	Custom TCP	TCP	5666	0.0.0.0/0	-
-	sgr-0cd7176351b1f8590	IPv4	All ICMP - IPv4	ICMP	All	0.0.0.0/0	-
-	sgr-0225770a0073a7...	IPv4	All traffic	All	All	0.0.0.0/0	-
-	sgr-0368947d47bb93...	IPv4	HTTP	TCP	80	0.0.0.0/0	-

4. Making the connection: Make the connection by SSH into Your EC2 instance. the command is given by **ssh -i "key_name.pem"**
ec2-user@ec2-public_IP_address.compute-1.amazonaws.com

```
PS C:\Users\Sadneya\Downloads> ssh -i "server.pem" ec2-user@ec2-18-232-155-134.compute-1.amazonaws.com
The authenticity of host 'ec2-18-232-155-134.compute-1.amazonaws.com (18.232.155.134)' can't be established.
ED25519 key fingerprint is SHA256:6UVLjB6FbGB7A92sIEobs4886tzb5yML0ekn5Xzfrw.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-232-155-134.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

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

5. Installing the packages: Install the following packages:

1. sudo yum update

```
[ec2-user@ip-172-31-39-90 ~]$ sudo yum update
sudo yum install httpd php
sudo yum install gcc glibc glibc-common
sudo yum install gd gd-devel
Last metadata expiration check: 0:02:27 ago on Fri Oct  4 03:35:02 2024.
Dependencies resolved.
Nothing to do.
Complete!
```

2. sudo yum install httpd php
3. sudo yum install gcc glibc glibc-common
4. sudo yum install gd gd-devel

<pre>Installed: brotli-1.0.9-4.amzn2023.0.2.x86_64 bzip2-devel-1.0.8-6.amzn2023.0.2.x86_64 cmake-filesystems-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 libwebp-1.2.4-1.amzn2023.0.6.x86_64 libxcb-1.13.1-7.amzn2023.0.2.x86_64 libxml2-devel-2.10.4-1.amzn2023.0.6.x86_64 pcre2-utf16-10.40-1.amzn2023.0.3.x86_64 pixman-0.40.0-3.amzn2023.0.3.x86_64 xml-common-0.6.3-56.amzn2023.0.2.noarch xz-devel-5.2.5-9.amzn2023.0.2.x86_64</pre>	<pre>brotli-devel-1.0.9-4.amzn2023.0.2.x86_64 cairo-1.17.6-2.amzn2023.0.1.x86_64 fontconfig-2.13.94-2.amzn2023.0.2.x86_64 fonts-filesystem-1:2.0.5-12.amzn2023.0.2.noarch freetype-devel-2.13.2-5.amzn2023.0.1.x86_64 gd-devel-2.3.3-5.amzn2023.0.3.x86_64 google-noto-fonts-common-20201206-2.amzn2023.0.2.noarch graphite2-1.3.14-7.amzn2023.0.2.x86_64 harfbuzz-7.0.0-2.amzn2023.0.1.x86_64 harfbuzz-icu-7.0.0-2.amzn2023.0.1.x86_64 langpacks-core-font-en-3.0-21.amzn2023.0.4.noarch libSM-1.2.3-8.amzn2023.0.2.x86_64 libX11-common-1.7.2-3.amzn2023.0.4.noarch libX11-xcb-1.7.2-3.amzn2023.0.4.x86_64 libXau-devel-1.0.9-6.amzn2023.0.2.x86_64 libXpm-3.5.15-2.amzn2023.0.3.x86_64 libXrender-0.9.10-14.amzn2023.0.2.x86_64 libblkid-devel-2.37.4-1.amzn2023.0.4.x86_64 libicu-67.1-7.amzn2023.0.3.x86_64 libjpeg-turbo-2.1.4-2.amzn2023.0.5.x86_64 libmount-devel-2.37.4-1.amzn2023.0.4.x86_64 libpng-devel-2:1.6.37-10.amzn2023.0.6.x86_64 libsep-devel-3.4-3.amzn2023.0.3.x86_64 libtiff-devel-4.4.0-4.amzn2023.0.18.x86_64 libwebp-devel-1.2.4-1.amzn2023.0.6.x86_64 libxcb-devel-1.13.1-7.amzn2023.0.2.x86_64 pcre2-devel-10.40-1.amzn2023.0.3.x86_64 pcre2-utf32-10.40-1.amzn2023.0.3.x86_64 sysprof-capture-devel-3.40.1-2.amzn2023.0.2.x86_64 xorg-x11proto-devel-2021.4-1.amzn2023.0.2.noarch zlib-devel-1.2.11-33.amzn2023.0.5.x86_64</pre>
--	--

Complete!

6.Create a new user:Create a new User with its password by following command. Then again retype the password for confirmation.Here,I am creating a user named “nagios”.

```
sudo adduser -m nagios
```

```
sudo passwd nagios
```

```
[ec2-user@ip-172-31-39-90 ~]$ sudo adduser -m nagios
sudo passwd nagios
Changing password for user nagios.
New password:
```

```
Retype new password:
```

```
passwd: all authentication tokens updated successfully.
```

7. **Create a new User group:** Create a new User group by following command.here I am creating a User group named “nagcmd”.

```
sudo groupadd nagcmd
```

```
[ec2-user@ip-172-31-39-90 ~]$ sudo groupadd nagcmd
```

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

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

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

```
[ec2-user@ip-172-31-39-90 ~]$ sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
```

9. Create a new directory for Nagios downloads

```
mkdir ~/downloads
```

```
cd ~/downloads
```

```
[ec2-user@ip-172-31-39-90 ~]$ mkdir ~/downloads
cd ~/downloads
```

10. Use wget to download the nagios which is a source zip file by following command.

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

```
[ec2-user@ip-172-31-39-90 downloads]$ wget https://go.nagios.org/l/975333/2024-09-17/6kqcx
--2024-10-04 03:45:50-- https://go.nagios.org/l/975333/2024-09-17/6kqcx
Resolving go.nagios.org (go.nagios.org)... 3.92.120.28, 34.237.219.119, 52.54.96.194, ...
Connecting to go.nagios.org (go.nagios.org)|3.92.120.28|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: http://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagios.org&utm_content=Download+Form&utm_campaign=Core+4.5.5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81d8 [following]
--2024-10-04 03:45:51-- http://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagios.org&utm_content=Download+Form&utm_campaign=Core+4.5.5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 2600:3c00::f03c:92ff:fef7:45ce
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagios.org&utm_content=Download+Form&utm_campaign=Core+4.5.5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81d8 [following]
--2024-10-04 03:45:51-- https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagios.org&utm_content=Download+Form&utm_campaign=Core+4.5.5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2065473 (2.0M) [application/x-gzip]
Saving to: '6kqcx'

6kqcx                                         100%[=====] 1.97M 7.32MB/s    i
2024-10-04 03:45:51 (7.32 MB/s) - '6kqcx' saved [2065473/2065473]

[ec2-user@ip-172-31-39-90 downloads]$ |
```

11. Then install nagios plugin by following command.

Wget <http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz>

```
[ec2-user@ip-172-31-39-90 downloads]$ wget http://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
--2024-10-04 03:47:43-- http://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251
Connecting to nagios-plugins.org (nagios-plugins.org)|45.56.123.251|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2753049 (2.6M) [application/x-gzip]
Saving to: 'nagios-plugins-2.4.11.tar.gz'

nagios-plugins-2.4.11.tar.gz   100%[=====] 2.62M
2024-10-04 03:47:44 (6.85 MB/s) - 'nagios-plugins-2.4.11.tar.gz' saved [2753049/2753049]
```

12. Use tar to unzip the file and then go inside that directory.

tar zxvf 6kqcx

```
[ec2-user@ip-172-31-39-90 downloads]$ tar zxvf 6kqcx
nagios-4.5.5/
nagios-4.5.5/.github/
nagios-4.5.5/.github/workflows/
nagios-4.5.5/.github/workflows/test.yml
nagios-4.5.5/.gitignore
nagios-4.5.5/CONTRIBUTING.md
nagios-4.5.5/Changelog
nagios-4.5.5/INSTALLING
nagios-4.5.5/LEGAL
nagios-4.5.5/LICENSE
nagios-4.5.5/Makefile.in
nagios-4.5.5/README.md
nagios-4.5.5/THANKS
nagios-4.5.5/UPGRADING
nagios-4.5.5/aclocal.m4
nagios-4.5.5/autoconf-macros/
nagios-4.5.5/autoconf-macros/.gitignore
nagios-4.5.5/autoconf-macros/CHANGELOG.md
nagios-4.5.5/autoconf-macros/LICENSE
```

cd nagios-4.5.5

```
[ec2-user@ip-172-31-39-90 downloads]$ cd nagios-4.5.5
```

13. Run the configuration script with the same group name you previously created.
./configure --with-command-group=nagcmd

```
[ec2-user@ip-172-31-39-90 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
```

This given error as can't find ssl headers

```
checking for Kerberos include files... configure: WARNING: could not find include files
checking for pkg-config... pkg-config
checking for SSL headers... configure: error: Cannot find ssl headers
```

14. For this install the following packages to solve the above error.

Sudo yum install openssl-devel

```
[ec2-user@ip-172-31-39-90 nagios-4.5.5]$ sudo yum install openssl-devel
Last metadata expiration check: 0:16:59 ago on Fri Oct  4 03:35:02 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           19 MB/s | 3.0 MB   00:00
Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing :                                         1/1
  Installing : openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64 1/1
  Running scriptlet: openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64 1/1
  Verifying   : openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64 1/1

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

Complete!
```

15.The again Run the configuration script with the same group name you previously created. Now the error has been removed.

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

```
[ec2-user@ip-172-31-39-90 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
```

```

Creating sample config files in sample-config/ ...

*** 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.

```

16. Then Install the make (binaries), then initialize the script and sample config files. Lastly, set permissions on the external command directory.

1. sudo make install
2. sudo make install-init
3. sudo make install-config
4. sudo make install-commandmode

```

[ec2-user@ip-172-31-39-90 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'

```

```
*** Main program, CGIs and HTML files installed ***

You can continue with installing Nagios as follows (type 'make'
without any arguments for a list of all possible options):

make install-init
  - This installs the init script in /lib/systemd/system

make install-commandmode
  - This installs and configures permissions on the
    directory for holding the external command file

make install-config
  - This installs sample config files in /usr/local/nagios/etc
```

```
*** Config files installed ***
```

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

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

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

17.Then Edit the configuration file and change the email address.

```
sudo nano /usr/local/nagios/etc/objects
```

```
cd nagios-plugins-2.4.11
```

```
[ec2-user@ip-172-31-39-90 nagios-plugins-2.4.11]$ ./configure --with-nagios-user=nagios
--with-nagios-group=nagios
```

```
[ec2-user@ip-172-31-39-90 downloads]$ cd nagios-plugins-2.4.11
[ec2-user@ip-172-31-39-90 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
```

18. Configure the web interface by following command.

sudo make install-webconf

```
[ec2-user@ip-172-31-39-90 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 [ $? -eq 1 ]; then \
    ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi
*** Nagios/Apache conf file installed ***
```

19. Create a “nagiosadmin” account for “nagios” login along with password. You’ll have to specify the password and then again retype the password.

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

```
[ec2-user@ip-172-31-39-90 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
```

20. Then again restart apache by following command: sudo service httpd restart

```
[ec2-user@ip-172-31-39-90 nagios-4.5.5]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
```

21. Then again go back to the downloads folder and unzip the plugins zip file.

cd ~/downloads

tar zxvf nagios-plugins-2.0.3.tar.gz

```
[ec2-user@ip-172-31-39-90 nagios-4.5.5]$ cd ~/downloads
[ec2-user@ip-172-31-39-90 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/
```

22. To Start Nagios. Firstly, Add Nagios to the list of system services

sudo chkconfig --add nagios

This given error as it can't find the directory.

```
[ec2-user@ip-172-31-39-90 nagios-plugins-2.4.11]$ sudo chkconfig --add nagios  
error reading information on service nagios: No such file or directory
```

23. Firstly check the configuration by this command

```
sudo chkconfig nagios on
```

24. Then Verify the sample configuration files by following command.

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

```
ec2-user@ip-172-31-39-90:~ $ 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
```

This command executed successfully with no errors and no warnings.

25. **Starting Nagios:** Start Nagios by following command

```
sudo service nagios start
```

```
[ec2-user@ip-172-31-39-90 ~]$ sudo service nagios start  
Redirecting to /bin/systemctl start nagios.service
```

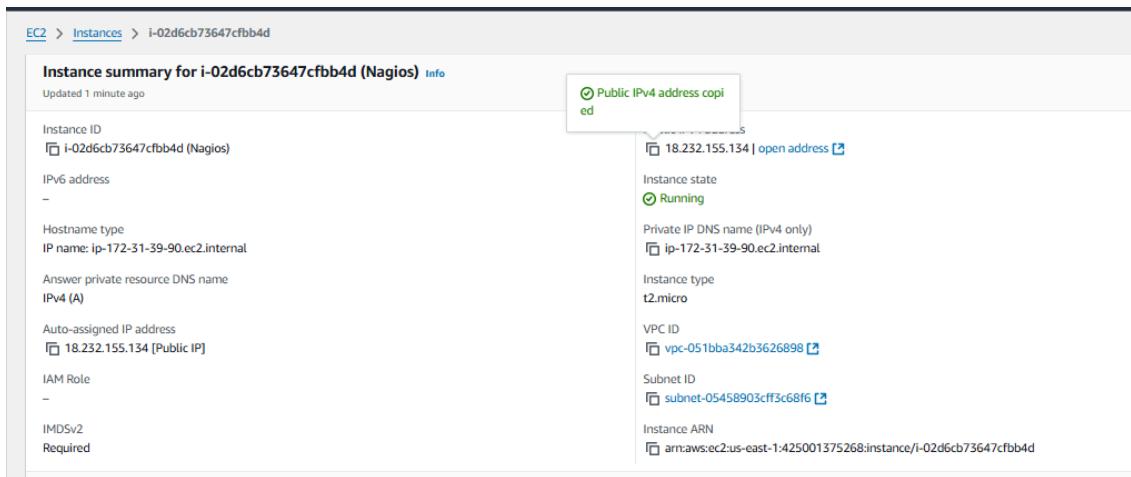
26. **Checking the status:** Then Check the status of Nagios by following command

```
sudo systemctl status nagios
```

```
[ec2-user@ip-172-31-39-90 ~]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; disabled; preset: disabled)
   Active: active (running) since Fri 2024-10-04 04:14:29 UTC; 1min 41s ago
     Docs: https://www.nagios.org/documentation
   Process: 75298 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Process: 75299 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 75300 (nagios)
   Tasks: 6 (limit: 1112)
  Memory: 5.6M
    CPU: 89ms
   CGroup: /system.slice/nagios.service
           └─75300 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─75301 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─75302 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─75303 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─75304 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─75305 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

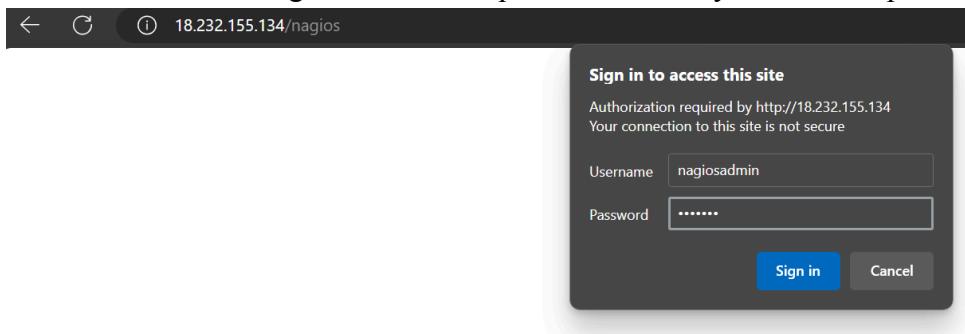
Oct 04 04:14:29 ip-172-31-39-90.ec2.internal nagios[75300]: wproc: Successfully registered manager as @wproc with query handler
Oct 04 04:14:29 ip-172-31-39-90.ec2.internal nagios[75300]: wproc: Registry request: name=Core Worker 75301;pid=75301
Oct 04 04:14:29 ip-172-31-39-90.ec2.internal nagios[75300]: wproc: Registry request: name=Core Worker 75302;pid=75302
Oct 04 04:14:29 ip-172-31-39-90.ec2.internal nagios[75300]: wproc: Registry request: name=Core Worker 75304;pid=75304
Oct 04 04:14:29 ip-172-31-39-90.ec2.internal nagios[75300]: wproc: Registry request: name=Core Worker 75303;pid=75303
Oct 04 04:14:29 ip-172-31-39-90.ec2.internal nagios[75300]: Successfully launched command file worker with pid 75305
Oct 04 04:14:29 ip-172-31-39-90.ec2.internal nagios[75300]: HOST ALERT: localhost;DOWN;SOFT;1;(No output on stdout) stderr: execvp(>
Oct 04 04:15:06 ip-172-31-39-90.ec2.internal nagios[75300]: SERVICE ALERT: localhost;Current Load;CRITICAL;HARD;1;(No output on stdo>
Oct 04 04:15:29 ip-172-31-39-90.ec2.internal nagios[75300]: HOST ALERT: localhost;DOWN;SOFT;2;(No output on stdout) stderr: execvp(>
Oct 04 04:15:44 ip-172-31-39-90.ec2.internal nagios[75300]: SERVICE ALERT: localhost;Current Users;CRITICAL;HARD;1;(No output on std>
[ec2-user@ip-172-31-39-90 ~]$ |
```

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



28. Open up your browser and look for **http://<your_public_ip_address>/nagios** (**http://18.232.155.134/nagios**)

Enter username as “nagiosadmin” and password which you set in Step 19.



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

The screenshot shows the Nagios Core 4.5.5 dashboard. On the left is a vertical navigation menu with sections: General (Home, Documentation), Current Status (Tactical Overview, Map, Hosts, Services, Host Groups, Grid), Service Groups (Summary, Grid), Problems (Services, (Unhandled), Hosts (Unhandled), Network Outages), Reports (Availability, Trends, Alerts, History, Summary, Histogram, Notifications, Event Log), and System (Comments, Downtime). The main content area features the Nagios Core logo and a green checkmark indicating "Daemon running with PID 75300". It displays the version "Nagios® Core™ Version 4.5.5" and the date "September 17, 2024". Below this are four boxes: "Get Started" (with a bulleted list of monitoring steps), "Latest News" (empty), "Don't Miss..." (empty), and "Quick Links" (with a list of external links like Nagios Library, Labs, Exchange, Support, and the official website).

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

Conclusion: Here,I have created an EC2 instance of t2.micro successfully.For EC2 instance I have created a security group nagios where I have added inbound rules.Then I have created user name “nagios” and user group “nagcmd”. Then I installed nagios and also nagios plugin successfully.Then I configured it but gave an error about can’t find ssl headers.Then I installed the required packages.Then it configured successfully.Then configured the web interfaces.Then I have started the nagios on the <http://18.232.155.134/nagios>. It given the final dashboard of nagios successfully

Experiment No:10

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

Prerequisites:

AWS Academy or Personal account.

Nagios Server running on Amazon Linux Machine. (Refer Experiment No 9)

Monitoring Using Nagios:

- Firstly, Confirm nagios Host is running or not by checking its status by following command.

sudo systemctl status nagios

```
[ec2-user@ip-172-31-39-90 ~]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; disabled; preset: disabled)
   Active: active (running) since Fri 2024-10-04 04:14:29 UTC; 9min ago
     Docs: https://www.nagios.org/documentation
 Process: 75298 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Process: 75299 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
Main PID: 75300 (nagios)
   Tasks: 6 (limit: 1112)
  Memory: 5.6M
    CPU: 164ms
   CGroup: /system.slice/nagios.service
           ├─75300 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           ├─75301 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─75302 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─75303 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─75304 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─75305 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 04 04:17:36 ip-172-31-39-90.ec2.internal nagios[75300]: SERVICE ALERT: localhost;Root Partition;CRITICAL;HARD;1;(No output on std>
Oct 04 04:18:14 ip-172-31-39-90.ec2.internal nagios[75300]: SERVICE ALERT: localhost;SSH;CRITICAL;HARD;1;(No output on stdout) stderr:>
Oct 04 04:18:29 ip-172-31-39-90.ec2.internal nagios[75300]: HOST ALERT: localhost;DOWN;SOFT;6;(No output on stdout) stderr: execvp(/>
Oct 04 04:18:51 ip-172-31-39-90.ec2.internal nagios[75300]: SERVICE ALERT: localhost;Swap Usage;CRITICAL;HARD;1;(No output on stdout)>
Oct 04 04:19:29 ip-172-31-39-90.ec2.internal nagios[75300]: SERVICE ALERT: localhost;Total Processes;CRITICAL;HARD;1;(No output on s>
Oct 04 04:19:29 ip-172-31-39-90.ec2.internal nagios[75300]: HOST ALERT: localhost;DOWN;SOFT;6;(No output on stdout) stderr: execvp(/>
Oct 04 04:20:29 ip-172-31-39-90.ec2.internal nagios[75300]: HOST ALERT: localhost;DOWN;SOFT;7;(No output on stdout) stderr: execvp(/>
Oct 04 04:21:29 ip-172-31-39-90.ec2.internal nagios[75300]: HOST ALERT: localhost;DOWN;SOFT;8;(No output on stdout) stderr: execvp(/>
Oct 04 04:22:29 ip-172-31-39-90.ec2.internal nagios[75300]: HOST ALERT: localhost;DOWN;SOFT;9;(No output on stdout) stderr: execvp(/>
Oct 04 04:23:29 ip-172-31-39-90.ec2.internal nagios[75300]: HOST ALERT: localhost;DOWN;HARD;10;(No output on stdout) stderr: execvp(>
Oct 04 04:23:29 ip-172-31-39-90.ec2.internal nagios[75300]: HOST ALERT: localhost;DOWN;HARD;10;(No output on stdout) stderr: execvp(>
lines 1-28/28 (END)
```

You can now proceed ahead if you get the above message/output. If not then again create a instance (Refer Experiment No 9)

- Creation an EC-2 Instance:** Create an EC-2 Instance of t2.micro type on ubuntu in AWS.

Name and tags [Info](#)

Name
Nagios-client [Add additional tags](#)

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

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

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

Recents [Quick Start](#)

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

Including AMIs from AWS, Marketplace and the Community

3. Then select the key pair that you have created and used in “nagios-host” EC-2 instance.

The screenshot shows the 'Instance type' section of the AWS instance creation wizard. It lists the 't2.micro' instance type as 'Free tier eligible'. Below the list, there's a note: 'Additional costs apply for AMIs with pre-installed software'. To the right, there are buttons for 'All generations' and 'Compare instance types'.

Instance type

t2.micro Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand RHEL base pricing: 0.026 USD per Hour
On-Demand Linux base pricing: 0.0116 USD per Hour

All generations Compare instance types

Additional costs apply for AMIs with pre-installed software

Key pair (login)

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

Key pair name - required

server Create new key pair

4. Then select the security group that we have created in nagios host.here I have given name to security groups as Nagios.

The screenshot shows the 'Network settings' section of the AWS instance creation wizard. It includes fields for 'Network' (vpc-051bba342b3626898), 'Subnet' (No preference (Default subnet in any availability zone)), and 'Auto-assign public IP' (Enable). There's a note about additional charges for public IPs outside the free tier allowance. The 'Firewall (security groups)' section indicates that a security group controls traffic to the instance. It shows a list of common security groups, with 'Nagios' selected. A note at the bottom says: 'Security groups that you add or remove here will be added to or removed from all your network interfaces.'

Network

vpc-051bba342b3626898

Subnet

No preference (Default subnet in any availability zone)

Auto-assign public IP

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups)

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

Create security group Select existing security group

Common security groups

Select security groups

Nagios sg-0b1355e80625c05ee X

VPC: vpc-051bba342b3626898

Compare security group rules

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

5. **Making the connection:** Then after instance get created successfully then click on connect then go to SSH client section there you will see connection command **ssh -i “key_pair.pem” ubuntu@ec2-Public_ip.compute-1.amazonaws.com**

The screenshot shows the AWS EC2 Instance Connect interface. At the top, there are four tabs: EC2 Instance Connect, Session Manager, SSH client (which is selected), and EC2 serial console. Below the tabs, under 'Instance ID', it shows 'i-06e099f1c55d0ec8d (Nagios-client)'. A numbered list of steps is provided:

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is server.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
 chmod 400 "server.pem"
4. Connect to your instance using its Public DNS:
 ec2-54-157-1-59.compute-1.amazonaws.com

Below the steps, there is an 'Example:' section with a checkbox next to the command:

ssh -i "server.pem" ubuntu@ec2-54-157-1-59.compute-1.amazonaws.com

Copy paste this command on command prompt.

```
PS C:\Users\Sadneya\downloads> ssh -i "server.pem" ec2-user@ec2-3-81-91-101.compute-1.amazonaws.com
The authenticity of host 'ec2-3-81-91-101.compute-1.amazonaws.com (3.81.91.101)' can't be established.
ED25519 key fingerprint is SHA256:6UVLjB6FbGB7A92sIEobs4886tzb5yML0ekn5Xzfrw.
This host key is known by the following other names/addresses:
  C:\Users\Sadneya/.ssh/known_hosts:68: ec2-18-232-155-134.compute-1.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-81-91-101.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

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

Last login: Fri Oct  4 03:36:10 2024 from 125.99.93.18
```

Now perform Following commands on Nagios-host

6. Now on the server Nagios-host run the following command.

ps -ef | grep nagios

```
[ec2-user@ip-172-31-39-90 ~]$ ps -ef | grep nagios
ec2-user  2377  2350  0 09:15 pts/0    00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-39-90 ~]$
```

7. Now become the root user by sudo su and create the directories.

sudo su

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

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

```
[ec2-user@ip-172-31-39-90 ~]$ sudo su
[root@ip-172-31-39-90 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts
[root@ip-172-31-39-90 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-39-90 ec2-user]#
```

8. Copy the localhost.cfg file to linuxserver.cfg by following command.

```
cp /usr/local/nagios/etc/objects/localhost.cfg  
/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
```

```
[root@ip-172-31-39-90 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
```

9. Open the linuxserver.cfg by nano command and then make following changes.

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

```
[root@ip-172-31-39-90 ec2-user]# nano /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg  
[root@ip-172-31-39-90 ec2-user]#
```

Now change the hostname to **linuxserver**. Then Change the address to the public IP of your Linux client.

Now Set hostgroup_name to **linux-servers1**.

```
GNU nano 5.8          /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg      Modified
#####
#
# 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     linuxserver
    alias         localhost
    address       98.83.6.103
}

#####
#
# 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
}

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

Make the hostname changes throughout in Host,Host Group,HTTP,SSH.

10. Now update the Nagios.config file add the following line in it.

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

Then execute the following command : **nano /usr/local/nagios/etc/nagios.cfg**

```

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

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

```

11. Now verify the configuration by the following command.

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

```

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

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

Running pre-flight check on configuration data...

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

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check

```

12. Now restart the services of nagios by running the following command.

```
service nagios restart
```

```
[root@ip-172-31-39-90 ec2-user]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
```

****Now perform Following commands on Nagios-host****

13. Firstly update the package by following command.

```
sudo apt update -y
```

```
ubuntu@ip-172-31-42-255:~$ sudo apt update -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [382 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [83.9 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4704 B]
Get:9 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [277 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [117 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [8632 B]
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [10.4 kB]
Get:13 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [10.9 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:15 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [2808 B]
Get:16 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:17 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [344 B]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]

Get:34 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [532 B]
Get:35 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]
Get:36 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 B]
Get:37 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [10.6 kB]
Get:38 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [10.8 kB]
Get:39 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [17.6 kB]
Get:40 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1104 B]
Get:41 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:42 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B]
Get:43 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:44 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 B]
Fetched 28.2 MB in 6s (4883 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
6 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

14. Now,Install the required packages by executing following commands.

sudo apt install gcc -y

```
ubuntu@ip-172-31-42-255:~$ sudo apt install gcc -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
binutils binutils-common binutils-x86-64-linux-gnu cpp cpp-13 cpp-13-x86-64-linux-gnu cpp-x86-64-linux-gnu fontconfig-config
fonts-dejavu-core fonts-dejavu-mono gcc-13 gcc-13-base gcc-13-x86-64-linux-gnu gcc-x86-64-linux-gnu libaom3 libasan8 libatomic1
libbinutils libc-dev-bin libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libde265-0 libdeflate0
libfontconfig1 libgcc-13-dev libgd3 libgomp1 libgprofng0 libheif-plugin-aomdec libheif-plugin-aomenc libheif-plugin-libde265
libheif1 libhwasan0 libisl23 libitm1 libjbig0 libjpeg-turbo8 libjpeg8 liblerc4 liblsan0 libmpc3 libquadmath0 libsframe1
libsharpyuv0 libtiff6 libtsan2 libubsan1 libwebp7 libxpm4 linux-libc-dev manpages-dev rpcsvc-proto
Suggested packages:
binutils-doc gprofng-gui cpp-doc gcc-13-locales cpp-13-doc gcc-multilib make autoconf automake libtool flex bison gcc-doc
gcc-13-multilib gcc-13-doc gdb-x86-64-linux-gnu glibc-doc libgd-tools libheif-plugin-x265 libheif-plugin-ffmpegdec
libheif-plugin-jpegdec libheif-plugin-jpegenc libheif-plugin-j2kdec libheif-plugin-j2kenc libheif-plugin-rav1e
libheif-plugin-svtenc
The following NEW packages will be installed:
binutils binutils-common binutils-x86-64-linux-gnu cpp cpp-13 cpp-13-x86-64-linux-gnu cpp-x86-64-linux-gnu fontconfig-config
fonts-dejavu-core fonts-dejavu-mono gcc gcc-13 gcc-13-base gcc-13-x86-64-linux-gnu gcc-x86-64-linux-gnu libaom3 libasan8
libatomic1 libbinutils libc-dev-bin libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libde265-0 libdeflate0
libfontconfig1 libgcc-13-dev libgd3 libgomp1 libgprofng0 libheif-plugin-aomdec libheif-plugin-aomenc libheif-plugin-libde265
libheif1 libhwasan0 libisl23 libitm1 libjbig0 libjpeg-turbo8 libjpeg8 liblerc4 liblsan0 libmpc3 libquadmath0 libsframe1
libsharpyuv0 libtiff6 libtsan2 libubsan1 libwebp7 libxpm4 linux-libc-dev manpages-dev rpcsvc-proto
0 upgraded, 57 newly installed, 0 to remove and 6 not upgraded.
Need to get 62.8 MB of archives.
After this operation, 222 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 binutils-common amd64 2.42-4ubuntu2 [239 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libbsframe1 amd64 2.42-4ubuntu2 [14.8 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libbinutils amd64 2.42-4ubuntu2 [572 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libctf-nobfd0 amd64 2.42-4ubuntu2 [97.1 kB]
Setting up gcc (4:13.2.0-7ubuntu1) ...
Setting up libheif-plugin-aomdec:amd64 (1.17.6-1ubuntu4) ...
Setting up libheif1:amd64 (1.17.6-1ubuntu4) ...
Setting up libheif-plugin-libde265:amd64 (1.17.6-1ubuntu4) ...
Setting up libheif-plugin-aomenc:amd64 (1.17.6-1ubuntu4) ...
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for sgml-base (1.31) ...
Setting up libfontconfig1:amd64 (2.15.0-1.1ubuntu2) ...
Setting up libgd3:amd64 (2.3.3-9ubuntu5) ...
Setting up libc-devtools (2.39-0ubuntu8.3) ...
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-42-255:~$ |
```

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

```
ubuntu@ip-172-31-42-255:~$ sudo apt install -y nagios-nrpe-server nagios-plugins
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'monitoring-plugins' instead of 'nagios-plugins'
The following additional packages will be installed:
  libavahi-client3 libavahi-common-data libavahi-common3 libcurl2t64 libdbdilt64 liblbdb2 libmysqlclient21 libnet-snmp-perl libpq5
  libradcli4 libsmcclient0 libsnmp-base libsnmp40t64 libtalloc2 libtdb1 libtevent0t64 liburiparser1 libwbcclient0
  monitoring-plugins-basic monitoring-plugins-common monitoring-plugins-standard mysql-common python3-gpg python3-ldb
  python3-markdown python3-samba python3-talloc python3-tdb rpcbind samba-common samba-common-bin samba-dsdb-modules samba-libs
  smcclient snmp
Suggested packages:
  cups-common libcrypt-des-perl libdigest-hmac-perl libio-socket-inet6-perl snmp-mibs-downloader icinga2 nagios-plugins-contrib
  fping postfix | sendmail-bin | exim4-daemon-heavy | exim4-daemon-light qstat xinetd | inetd python-markdown-doc heimdal-clients
  python3-dnspython cifs-utils
The following NEW packages will be installed:
  libavahi-client3 libavahi-common-data libavahi-common3 libcurl2t64 libdbdilt64 liblbdb2 libmysqlclient21 libnet-snmp-perl libpq5
  libradcli4 libsmcclient0 libsnmp-base libsnmp40t64 libtalloc2 libtdb1 libtevent0t64 liburiparser1 libwbcclient0 monitoring-plugins
  monitoring-plugins-basic monitoring-plugins-common monitoring-plugins-standard mysql-common nagios-nrpe-server python3-gpg
  python3-ldb python3-markdown python3-samba python3-talloc python3-tdb rpcbind samba-common samba-common-bin samba-dsdb-modules
  samba-libs smcclient snmp
0 upgraded, 37 newly installed, 0 to remove and 6 not upgraded.
Need to get 16.1 MB of archives.
After this operation, 72.0 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 nagios-nrpe-server amd64 4.1.0-1ubuntu3 [356 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 rpcbind amd64 1.2.6-7ubuntu2 [46.5 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libavahi-common-data amd64 0.8-13ubuntu6 [29.7 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libavahi-common3 amd64 0.8-13ubuntu6 [23.3 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libavahi-client3 amd64 0.8-13ubuntu6 [26.8 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libcurl2t64 amd64 2.4.7-1.2ubuntu7.3 [272 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libdbdilt64 amd64 0.9.0-6.1build1 [25.7 kB]

Creating config file /etc/nagios-plugins/config/netware.cfg with new version
Creating config file /etc/nagios-plugins/config/nt.cfg with new version
Creating config file /etc/nagios-plugins/config/pgsql.cfg with new version
Creating config file /etc/nagios-plugins/config/radius.cfg with new version
Creating config file /etc/nagios-plugins/config/rpc-nfs.cfg with new version
Creating config file /etc/nagios-plugins/config/snmp.cfg with new version
Setting up monitoring-plugins (2.3.5-1ubuntu3) ...
Setting up liblbdb2:amd64 (2:2.8.0+samba4.19.5+dfsg-4ubuntu9) ...
Setting up libavahi-client3:amd64 (0.8-13ubuntu6) ...
Setting up samba-libs:amd64 (2:4.19.5+dfsg-4ubuntu9) ...
Setting up python3-ldb (2:2.8.0+samba4.19.5+dfsg-4ubuntu9) ...
Setting up samba-dsdb-modules:amd64 (2:4.19.5+dfsg-4ubuntu9) ...
Setting up libsmcclient0:amd64 (2:4.19.5+dfsg-4ubuntu9) ...
Setting up libcurl2t64:amd64 (2.4.7-1.2ubuntu7.3) ...
Setting up python3-samba (2:4.19.5+dfsg-4ubuntu9) ...
Setting up smcclient (2:4.19.5+dfsg-4ubuntu9) ...
Setting up samba-common-bin (2:4.19.5+dfsg-4ubuntu9) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-42-255:~$ |
```

15. Now open the nrpe.cfg file. Scroll down the page and find allowed_hosts. Go there, and add your nagios host IP address.

sudo nano /etc/nagios/nrpe.cfg

```

GNU nano 7.2
nrpe_user=nagios

# NRPE GROUP
# This determines the effective group that the NRPE daemon should run as.
# You can either supply a group name or a GID.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd
nrpe_group=nagios

# ALLOWED HOST ADDRESSES
# This is an optional comma-delimited list of IP address or hostnames
# that are allowed to talk to the NRPE daemon. Network addresses with a bit mask
# (i.e. 192.168.1.0/24) are also supported. Hostname wildcards are not currently
# supported.
#
# Note: The daemon only does rudimentary checking of the client's IP
# address. I would highly recommend adding entries in your /etc/hosts.allow
# file to allow only the specified host to connect to the port
# you are running this daemon on.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd
allowed_hosts=127.0.0.1,::1,3.81.91.101

# COMMAND ARGUMENT PROCESSING
# This option determines whether or not the NRPE daemon will allow clients

```

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

16. Now restart the NRPE server

sudo systemctl restart nagios-nrpe-server

```
ubuntu@ip-172-31-42-255:~$ sudo systemctl restart nagios-nrpe-server
```

17. Now again check the nagios-host status

sudo systemctl status nagios

Now check the status is active or not

```
[root@ip-172-31-39-90 ec2-user]# sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; disabled; preset: disabled)
   Active: active (running) since Fri 2024-10-04 09:24:28 UTC; 9min ago
     Docs: https://www.nagios.org/documentation
 Process: 2725 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Process: 2727 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 2729 (nagios)
   Tasks: 6 (limit: 1112)
  Memory: 4.2M
    CPU: 114ms
   CGroup: /system.slice/nagios.service
           ├─2729 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           ├─2730 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─2731 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─2732 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─2733 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─2742 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 04 09:30:28 ip-172-31-39-90.ec2.internal nagios[2729]: HOST ALERT: linuxserver;DOWN;SOFT;7;(No output on stdout) stderr>
Oct 04 09:31:28 ip-172-31-39-90.ec2.internal nagios[2729]: HOST ALERT: linuxserver;DOWN;SOFT;8;(No output on stdout) stderr>
Oct 04 09:32:28 ip-172-31-39-90.ec2.internal nagios[2729]: HOST ALERT: linuxserver;DOWN;SOFT;9;(No output on stdout) stderr>
Oct 04 09:33:28 ip-172-31-39-90.ec2.internal nagios[2729]: HOST NOTIFICATION: nagiosadmin;linuxserver;DOWN;notify-host-by-email
Oct 04 09:33:28 ip-172-31-39-90.ec2.internal nagios[2729]: HOST ALERT: linuxserver;DOWN;HARD;10;(No output on stdout) stderr>
Oct 04 09:33:28 ip-172-31-39-90.ec2.internal nagios[2729]: wproc: NOTIFY job 6 from worker Core Worker 2732 is a non-check >
Oct 04 09:33:28 ip-172-31-39-90.ec2.internal nagios[2729]: wproc: host=linuxserver; service=(none); contact=nagiosadmin
Oct 04 09:33:28 ip-172-31-39-90.ec2.internal nagios[2729]: wproc: early_timeout=0; exited_ok=1; wait_status=32512; error_code=0
Oct 04 09:33:28 ip-172-31-39-90.ec2.internal nagios[2729]: wproc: stderr line 01: /bin/sh: line 1: /bin/mail: No such file or directory
Oct 04 09:33:28 ip-172-31-39-90.ec2.internal nagios[2729]: wproc: stderr line 02: /usr/bin/printf: write error: Broken pipe
[root@ip-172-31-39-90 ec2-user]#
```

18. Now,check the status of http

```
sudo systemctl status httpd
```

```
sudo systemctl start httpd
```

```
sudo systemctl enable httpd
```

```
[root@ip-172-31-39-90 ec2-user]# sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
   Drop-In: /usr/lib/systemd/system/httpd.service.d
             └─php-fpm.conf
   Active: active (running) since Fri 2024-10-04 09:45:14 UTC; 33min ago
     Docs: man:httpd.service(8)
   Main PID: 4146 (httpd)
      Status: "Total requests: 19; Idle/Busy workers 100/0;Requests/sec: 0.0095; Bytes served/sec: 70 B/sec"
      Tasks: 230 (limit: 1112)
     Memory: 17.3M
        CPU: 1.428s
   CGroup: /system.slice/httpd.service
           ├─4146 /usr/sbin/httpd -DFOREGROUND
           ├─4148 /usr/sbin/httpd -DFOREGROUND
           ├─4149 /usr/sbin/httpd -DFOREGROUND
           ├─4150 /usr/sbin/httpd -DFOREGROUND
           ├─4151 /usr/sbin/httpd -DFOREGROUND
           └─4533 /usr/sbin/httpd -DFOREGROUND

Oct 04 09:45:14 ip-172-31-39-90.ec2.internal systemd[1]: Stopped httpd.service - The Apache HTTP Server.
Oct 04 09:45:14 ip-172-31-39-90.ec2.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Oct 04 09:45:14 ip-172-31-39-90.ec2.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Oct 04 09:45:14 ip-172-31-39-90.ec2.internal httpd[4146]: Server configured, listening on: port 80
[root@ip-172-31-39-90 ec2-user]# sudo systemctl start httpd
[root@ip-172-31-39-90 ec2-user]# sudo systemctl enable httpd
```

19. Now to check Nagios dashboard go to <http://<Nagios-host ip>/nagios>.

Take host public IP address to add in “Nagios-host ip”.

The screenshot shows the Nagios Core 4.5.5 dashboard. At the top right, it displays "Nagios® Core™ Version 4.5.5" and the date "September 17, 2024". A green checkmark indicates "Daemon running with PID 2729". The left sidebar contains navigation links for General, Current Status, Reports, and System. The main content area includes sections for "Get Started", "Latest News", "Don't Miss...", and "Quick Links". The "Current Status" section provides an overview of host and service status totals.

Up	Down	Unreachable	Pending
2	0	0	0

Ok	Warning	Unknown	Critical	Pending
8	1	0	7	0

20. Now Click on Hosts from the left side panel there you will get the status of linuxserver and localhost. Check whether its status is up or down.If it is up then it is successful.

The screenshot shows the "Host Status Details For All Host Groups" table. It lists two hosts: "linuxserver" and "localhost", both of which are marked as "UP". The table includes columns for Host, Status, Last Check, Duration, and Status Information.

Host	Status	Last Check	Duration	Status Information
linuxserver	UP	10-04-2024 10:41:21	0d 0h 2m 33s	PING OK - Packet loss = 0%, RTA = 1.23 ms
localhost	UP	10-04-2024 10:43:13	0d 0h 3m 11s	PING OK - Packet loss = 0%, RTA = 0.06 ms

21. Now click on linuxserver. There you will get Host state information.

The screenshot shows the Nagios web interface with the URL 3.81.91.101/nagios/. The left sidebar is titled 'Current Status' and includes links for 'Tactical Overview', 'Map', 'Hosts', 'Services', 'Host Groups', 'Service Groups', 'Problems', 'Reports', 'System', and 'Comments'. The main content area displays 'Host Information' for 'localhost' (linuxserver), which is currently 'UP' (green). It provides details like last update, performance data, and check history. Below this is the 'Host State Information' panel, which lists various status metrics such as Active Checks (ENABLED), Passive Checks (ENABLED), and Notifications (ENABLED). To the right is the 'Host Commands' panel, which contains a list of actions like 'Locate host on map' and 'Disable active checks of this host'. At the bottom is the 'Host Comments' section, which is currently empty.

22. Then click on Services preset on left sidebar where you will get current network status.

The screenshot shows the Nagios web interface with the URL 3.81.91.101/nagios/. The left sidebar is titled 'Current Status' and includes links for 'Tactical Overview', 'Map', 'Hosts', 'Services', 'Host Groups', 'Service Groups', 'Problems', 'Reports', 'System', and 'Comments'. The main content area displays 'Current Network Status' for the host 'localhost'. It shows 'Host Status Totals' with 2 hosts up and 0 down, and 'Service Status Totals' with 10 services ok, 1 warning, 0 unknown, 5 critical, and 0 pending. Below this is the 'Service Status Details For All Hosts' table, which lists services for both 'linuxserver' and 'localhost'. For 'linuxserver', services include Current Load (OK), Current Users (OK), HTTP (CRITICAL), PING (OK), Root Partition (OK), SSH (CRITICAL), Swap Usage (CRITICAL), Total Processes (OK). For 'localhost', services include Current Load (OK), Current Users (OK), HTTP (WARNING), PING (CRITICAL), Root Partition (OK), SSH (OK), Swap Usage (CRITICAL), Total Processes (OK). The table includes columns for Host, Service, Status, Last Check, Duration, Attempt, and Status Information.

Conclusion: In this experiment, we created a nagios client EC-2 instance of t2.micro ubuntu instance. Then after updating hostname, hostgroup_name. Then adding the address to the public IP of your Linux client. Then verify the configuration and then check the status of nagios-host and then start and enable it then open <http://<Nagios-host ip>/nagios>. There you will get the status of linuxserver and localhost. Thus we can monitor essential network services on the Linux server.

Name:Sadneya Sadanand Samant

Roll No:46

Advdevops_10

Experiment No:11

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

Steps:

1. Go to AWS ACADEMY.

2. **Create the lambda function:**

Firstly, Search lambda, then Open lambda and then click on create function button.

The screenshot shows the AWS Lambda Functions list page. At the top, there is a header with 'Lambda > Functions'. Below the header, a search bar says 'Filter by tags and attributes or search by keyword'. There are buttons for 'Actions' and 'Create function'. The main area displays a table with the following data:

	Function name	Description	Package type	Runtime	Last modified
<input type="checkbox"/>	MainMonitoringFunction	-	Zip	Python 3.8	2 months ago
<input type="checkbox"/>	ModLabRole	updates LabRole to allow it to assume itself	Zip	Python 3.8	2 months ago
Create Redshift					

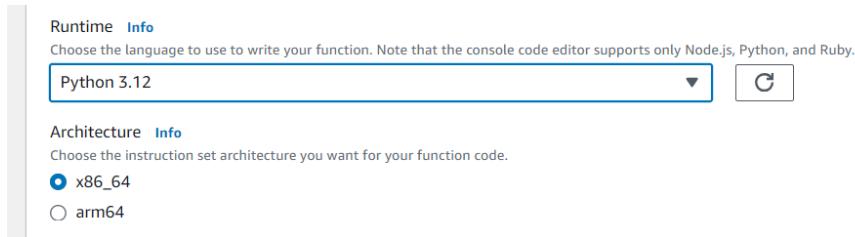
3. Now Give a name to your Lambda function,

The screenshot shows the 'Create function' wizard. At the top, there is a breadcrumb trail: 'Lambda > Functions > Create function'. Below the title, it says 'Choose one of the following options to create your function.' There are three radio button options:

- Author from scratch: Start with a simple Hello World example.
- Use a blueprint: Build a Lambda application from sample code and configuration presets for common use cases.
- Container image: Select a container image to deploy for your function.

Below the options, there is a section titled 'Basic information' with a 'Function name' field. The field contains 'anshi-lambda'. A note below the field says: 'Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).'

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



Thus the Lambda function was created successfully.

ansi-lambda

Function overview [Info](#)

Diagram **Template**

ansi-lambda (0)

+ Add trigger + Add destination

Description
-

Last modified
2 seconds ago

Function ARN
arn:aws:lambda:us-east-1:708398963195:function:ansi-lambda

Function URL [Info](#)

5. Then go to code section.

Code source [Info](#)

Test

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

Upload from

So to Edit the basic settings, go to configuration then click on edit.

General configuration [Info](#)

Edit

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

Now enter a description and change Memory and Timeout. Here, I've changed the Timeout period to 1 sec.

Name: Sadneya Sadanand Samant

Roll No: 46

AdvDevops_11

Basic settings [Info](#)

Description - optional
Basic settings

Memory [Info](#)
Your function is allocated CPU proportional to the memory configured.
128 MB
Set memory to between 128 MB and 10240 MB.

Ephemeral storage [Info](#)
You can configure up to 10 GB of ephemeral storage (/tmp) for your function. [View pricing](#)

512 MB
Set ephemeral storage (/tmp) to between 512 MB and 10240 MB.

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 1 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.

6. Now Click on the Test tab then select Create a new event, give a name to the event here i have given name as “our_event” and then select Event Sharing to private, and select hello-world template.

Test event [Info](#)

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

Test event action
 Create new event
 Edit saved event

Event name
our_event

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

Event sharing settings
 Private
This event is only available in the Lambda console and to the event creator. You can configure a total of 10. [Learn more](#)
 Shareable
This event is available to IAM users within the same account who have permissions to access and use shareable events. [Learn more](#)

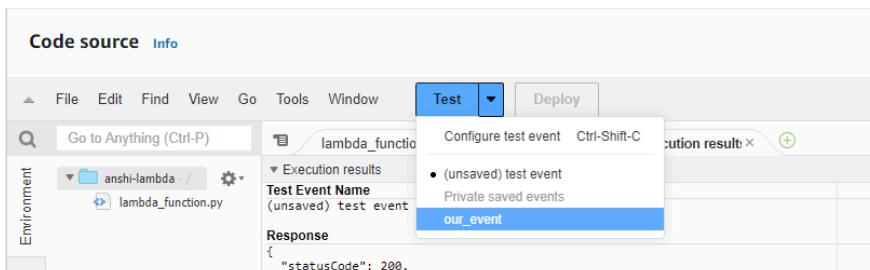
Template - optional
hello-world

Event JSON

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

[Format JSON](#)

7. **Testing & Deployment:** Now In Code section select the created event (our_event) from the dropdown of test ,then click on test .



8. Now you will see the following output.

The screenshot shows the AWS Lambda Test interface with the 'Execution results' tab selected. It displays the 'Test Event Name' as 'our_event', the 'Response' as a JSON object with 'statusCode: 200' and 'body: "Hello from Lambda!"', and the 'Function Logs' which show the request and response details. The 'Request ID' is also visible.

```

Test Event Name
our_event

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

Function Logs
START RequestId: acba54af-dad3-4830-b390-c09f019d1192 Version: $LATEST
END RequestId: acba54af-dad3-4830-b390-c09f019d1192
REPORT RequestId: acba54af-dad3-4830-b390-c09f019d1192 Duration: 1.76 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 32 MB

Request ID
acba54af-dad3-4830-b390-c09f019d1192
  
```

9. Making changes

You can edit your lambda function code. Here I have created a new string name “new_string” and assigned a string to it.

The screenshot shows the AWS Lambda Code source interface. The code editor window displays the 'lambda_function.py' file with the following content:

```

1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     new_string="This is a lambda function by Anshi and Sadneya"
6     return {
7         'statusCode': 200,
8         'body': json.dumps('Hello from Lambda!')
9     }
10
  
```

Now save it by ctrl+s and then click finally on deploy to deploy the changes.

The screenshot shows the AWS Lambda Code source interface. The code editor window displays the 'lambda_function.py' file with the following content, showing the changes made in the previous step:

```

1 import json
2
3 def lambda_handler(event, context):
4     # Implementing the custom string in the response
5     new_string = "This is a lambda function by Anshi and Sadneya"
6     return {
7         'statusCode': 200,
8         'body': json.dumps(new_string) # Returning the custom string in the response
9     }
10
  
```

10. Testing and redeploying changes Now click on the test and observe the output. Thus Output gives status code 200. Thus deployment is done successfully.

The screenshot shows the AWS Lambda Test interface. At the top, there are tabs for 'Code source' and 'Info'. Below the tabs is a menu bar with File, Edit, Find, View, Go, Tools, Window, a 'Test' button (which is highlighted in blue), and a 'Deploy' button. On the left, there's a sidebar labeled 'Environment' with a dropdown set to 'ansi-lambda /' and a gear icon. The main area has a search bar 'Go to Anything (Ctrl-P)'. Below it, there's a tree view under 'lambda_function': 'Execution results' (expanded) and 'Test Event Name' (set to 'our_event'). Under 'Response', there's a JSON snippet:

```
{  
  "statusCode": 200,  
  "body": "\"This is a lambda function by Anshi and Sadneya\""  
}
```

Below the response, 'Function Logs' show the execution details:

```
START RequestId: 501daf29-295f-46c5-b191-6fc0fc5fd69e Version: $LATEST  
END RequestId: 501daf29-295f-46c5-b191-6fc0fc5fd69e  
REPORT RequestId: 501daf29-295f-46c5-b191-6fc0fc5fd69e Duration: 1.94 ms Billed Duration: 2 ms
```

At the bottom, the 'Request ID' is listed as 501daf29-295f-46c5-b191-6fc0fc5fd69e.

Conclusion: In this experiment, we have successfully created an AWS Lambda function. After we have chosen the python language for writing the function. Then we edited the basic settings, including adjusting the timeout to 1 second. Then we tested it and finally deployed it. Thus it got deployed successfully. Additionally, we modified the Lambda function's code and redeployed it to observe the changes in real-time. This provided information about the simplicity of AWS Lambda in creating serverless application.

Experiment No: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

Steps :

1. Creating the lambda function:

Go to AWS ACADEMY. Now search and open S3 from services. Then click on create bucket.

The screenshot shows the AWS S3 console. At the top, there's a header with 'Amazon S3' and a 'View Storage Lens dashboard' button. Below the header, a banner says 'Account snapshot - updated every 24 hours' and 'All AWS Regions'. It also mentions 'Storage lens provides visibility into storage usage and activity trends' with a 'Learn more' link. The main area is titled 'General purpose buckets' with a sub-tab 'Directory buckets'. There's a search bar labeled 'Find buckets by name'. A table lists one bucket: 'elasticbeanstalk-us-east-1-708398963195' (Name), 'US East (N. Virginia) us-east-1' (AWS Region), 'View analyzer for us-east-1' (IAM Access Analyzer), and 'August 3, 2024, 22:25:17 (UTC+05:30)' (Creation date). Action buttons include 'Copy ARN', 'Empty', 'Delete', and a highlighted 'Create bucket' button.

2. Now Give a name to your Bucket. Here I have given the name as lambda_buche.

The screenshot shows the 'General configuration' step of creating a new S3 bucket. It includes fields for 'AWS Region' (set to 'US East (N. Virginia) us-east-1'), 'Bucket type' (with 'General purpose' selected), 'Bucket name' (set to 'lambda_buche'), and 'Copy settings from existing bucket - optional' (with a 'Choose bucket' button). A note at the bottom says 'Format: s3://bucket/prefix'.

3. then remove the Block public access and keep other settings default.

Block Public Access settings for this bucket

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

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

- Block public access to buckets and objects granted through new access control lists (ACLs)**
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLs)**
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies**
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

4. Thus, the S3 bucket was created successfully.

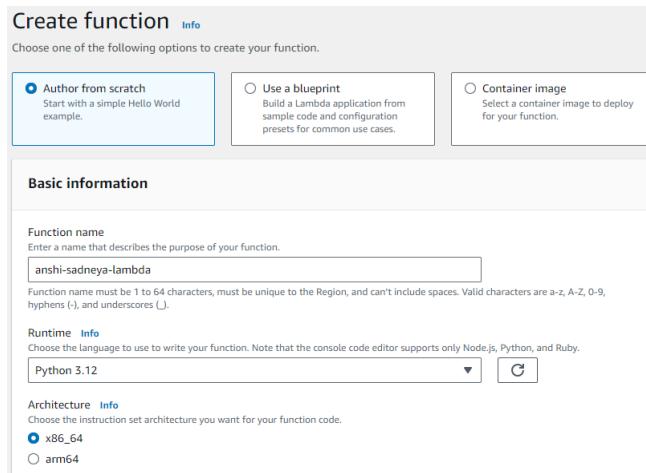
Name	AWS Region	IAM Access Analyzer	Creation date
anshi-sadneya-lambda-bucket	US East (N. Virginia) us-east-1	View analyzer for us-east-1	October 5, 2024, 11:28:16 (UTC+05:30)
elasticbeanstalk-us-east-1-708398963195	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 3, 2024, 22:25:17 (UTC+05:30)

5. Search and Open lambda console and click on create function button.

```

    exports.handler = async (event) => {
      console.log(event);
      return "Hello from Lambda!";
    };
  
```

6. Now Give a name to your Lambda function, Select the language to write your function. Here I have chosen python3.12, Architecture as x86, and Execution role to Create a new role with basic Lambda permissions. Note that the console code editor supports only Node.js, Python, and Ruby



7. Thus the Lambda function was created successfully.

8. Then Go into the code section. You will see some default code there.

```

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

```

9. To Edit the basic settings go to configuration then click on edit setting.

The screenshot shows the 'Configuration' tab selected in the top navigation bar. Under 'General configuration', the following settings are displayed:

- Description:** (empty)
- Memory:** 128 MB
- Ephemeral storage:** 512 MB
- SnapStart:** Info (None)
- Timeout:** 0 min 3 sec

10. Here, enter a description which is optional and change Memory and Timeout.
I've changed the Timeout period to 1 sec.

The 'Edit basic settings' dialog box shows the following configuration:

- Basic settings:** Info
- Description - optional:** Basic Settings
- Memory:** Info (128 MB) - Set memory to between 128 MB and 10240 MB.
- Ephemeral storage:** Info (512 MB) - Set ephemeral storage (/tmp) to between 512 MB and 10240 MB.
- SnapStart:** Info (None) - 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.
- Timeout:** 0 min 1 sec
- Execution role:**
 - Use an existing role
 - Create a new role from AWS policy templates

11. Now Click on the Test then select Create a new event, give a name to the event. Here I have given name as 'our-tester' and then select Event Sharing to private, and select s3 put template.

The 'Test event' dialog box shows the following configuration:

- Test event:** Info
- Test event action:** Create new event
- Event name:** our-tester
- Event sharing settings:**
 - Private (selected): This event is only available in the Lambda console and to the event creator. You can configure a total of 10. Learn more.
 - Shareable: This event is available to IAM users within the same account who have permissions to access and use shareable events. Learn more.
- Template - optional:** hello-world
- Event JSON:**

```

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

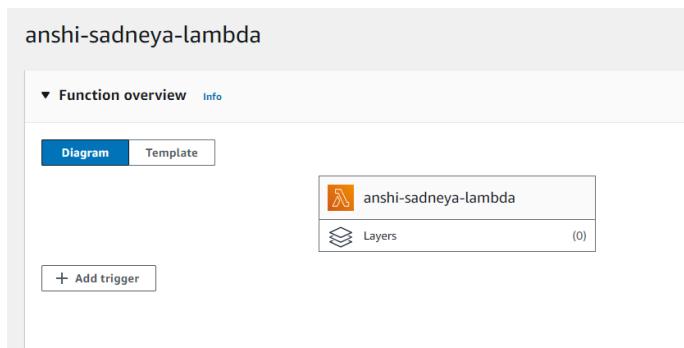
```

12. Now go to the Code section. Then click on the Test dropdown icon and select the event which we have created now('our-tester').

The screenshot shows a code editor interface with a toolbar at the top. The toolbar includes File, Edit, Find, View, Go, Tools, Window, Test (which is highlighted in blue), Deploy, and a dropdown menu. Below the toolbar is a search bar labeled "Go to Anything (Ctrl-P)". To the left, there's a sidebar titled "Environment" with a folder icon labeled "anshi-sadneya-lamb" and a file icon labeled "lambda_function.py". The main area displays a Python script:

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

13. Now go into the Lambda function and then click on add trigger.



14. Now in the Trigger information. Select the source as S3. Then select the bucket which we have created now (lambda_buche), keep other things default and also you can add prefix to image.

Lambda > Add triggers

Add trigger

Trigger configuration [Info](#)

S3 aws asynchronous storage

Bucket
Choose or enter the ARN of an S3 bucket that serves as the event source. The bucket must be in the same region as the function.
s3/ansi-sadneya-lambda-bucket

Bucket region: us-east-1

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

All object create events

Prefix - optional
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters. Any [special characters](#) must be URL encoded.
Image1

15. Thus, Trigger created successfully.

ansi-sadneya-lambda

The trigger ansi-sadneya-lambda-bucket was successfully added to function ansi-sadneya-lambda. The function is running.

Function overview [Info](#)

Diagram **Template**

ansi-sadneya-lambda
Layers (0)

S3

+ Add trigger

16. You can also check it in the configuration section.

Code | Test | Monitor | **Configuration** | Aliases | Versions

General configuration

Triggers

Permissions

Destinations

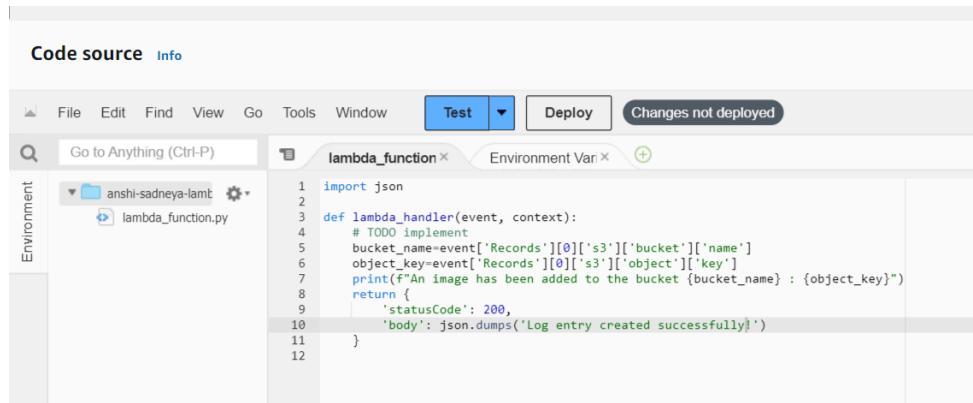
Function URL

Environment variables

Triggers (1) [Info](#)

Trigger	ARN	Action
S3: ansi-sadneya-lambda-bucket	arn:aws:s3:::ansi-sadneya-lambda-bucket	<input type="button" value="Details"/>

17. Now write a code which logs a message “Log entry created successfully” when triggered.



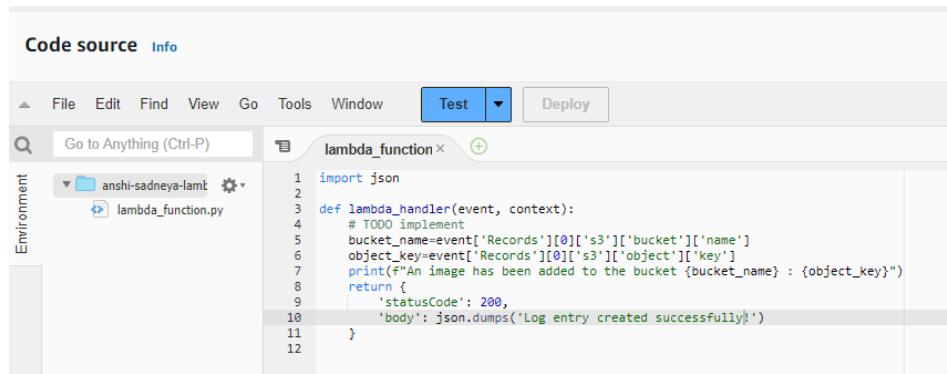
The screenshot shows the AWS Lambda function editor. The code editor window is titled "lambda_function". It contains the following Python code:

```
import json
def lambda_handler(event, context):
    # TODO implement
    bucket_name=event['Records'][0]['s3']['bucket']['name']
    object_key=event['Records'][0]['s3']['object']['key']
    print(f"An image has been added to the bucket {bucket_name} : {object_key}")
    return {
        'statusCode': 200,
        'body': json.dumps('Log entry created successfully!')
    }
```

The status bar at the bottom right of the editor window says "Changes not deployed".

Here changes are not deployed.

18. So now, Save the file by ctrl+s and then click on deploy.



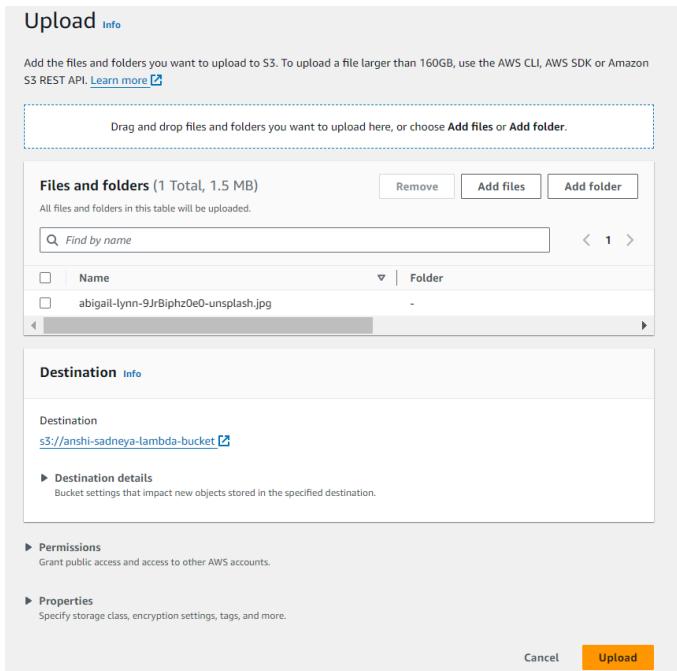
The screenshot shows the AWS Lambda function editor after deployment. The code editor window is titled "lambda_function". The code is identical to the previous screenshot. The status bar at the bottom right of the editor window now says "Deployment successful".

19. Go to S3 bucket, and there upload any image to the bucket.

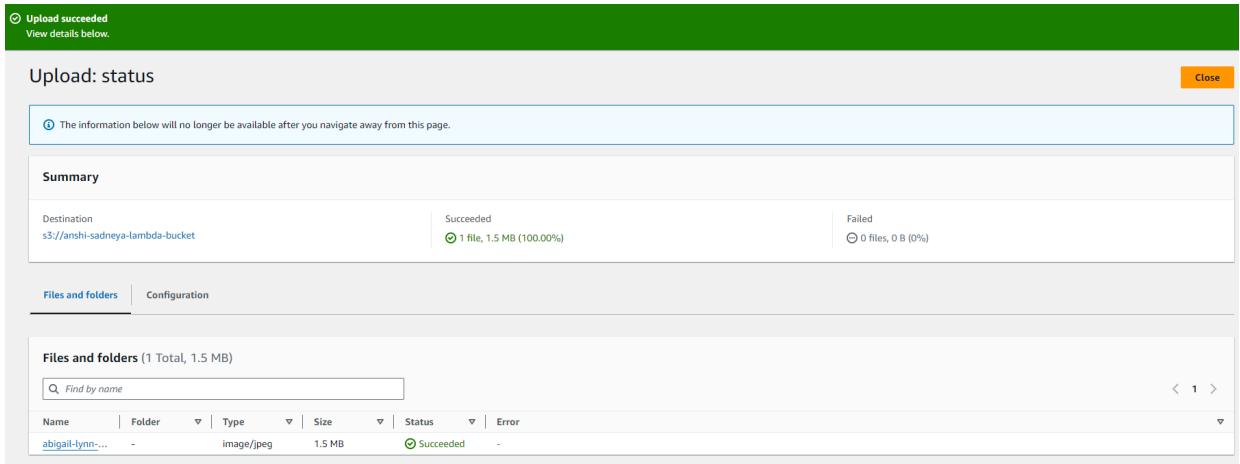
Name: Sadneya Sadanand Samant

Roll No: 46

AdvDevops_12



20. Thus image uploaded successfully.



21. Now goto lambda function. Then click on test. This will give you log about the image that we have uploaded in S3 bucket.

The screenshot shows the AWS Lambda Test console interface. At the top, there are tabs for 'Code source' and 'Info'. Below the tabs is a menu bar with File, Edit, Find, View, Go, Tools, Window, and a 'Test' dropdown. To the right of the menu is a 'Deploy' button. A search bar labeled 'Go to Anything (Ctrl-P)' is followed by a dropdown menu showing 'lambda_function' and 'Execution result'. The main area displays the 'Execution result' for a test event named 'our-tester'. The response body contains the message: "Log entry created successfully!". The function logs section shows detailed information about the log entry creation, including event details like RequestId, event time, and S3 bucket information. The status bar at the bottom indicates 'Status: Succeeded', 'Max memory used: 32 MB', and 'Time: 12.69 ms'.

In response, It gives status 200 and also the message “Log entry created successfully” and also contains function Logs.

22. Now go to cloudwatch. Then go into log groups. Inside that you will get the lambda function name that we have created click on it. Here, you will get a detailed log of events.

The screenshot shows the AWS CloudWatch Log Groups interface. The path is CloudWatch > Log groups > /aws/lambda/anshi-sadneya-lambda > 2024/10/05/[LATEST]. The log events table displays the following data:

Timestamp	Message
2024-10-05T06:45:53.704Z	INIT START Runtime Version: python:3.12.v36 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:188d9ce2e2714ff5637bd2bbe06ceb81ec3bc408a0f277dab104c14cd814b081
2024-10-05T06:45:53.778Z	START RequestId: 95f7832d-4bb2-45c3-99ae-64f848bc719e Version: \$LATEST
2024-10-05T06:45:53.778Z	Event received: {
2024-10-05T06:45:53.778Z	"key1": "value1",
2024-10-05T06:45:53.778Z	"key2": "value2",
2024-10-05T06:45:53.778Z	"key3": "value3"
2024-10-05T06:45:53.778Z	}
2024-10-05T06:45:53.780Z	END RequestId: 95f7832d-4bb2-45c3-99ae-64f848bc719e
2024-10-05T06:45:53.780Z	REPORT RequestId: 95f7832d-4bb2-45c3-99ae-64f848bc719e Duration: 1.71 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 32 MB Init Duration: 71.52 ms
2024-10-05T06:47:52.671Z	START RequestId: bb644137-1371-41cd-94cf-802cae05c1724 Version: \$LATEST
2024-10-05T06:47:52.671Z	Event received: {
2024-10-05T06:47:52.671Z	"Records": [
2024-10-05T06:47:52.671Z	{
2024-10-05T06:47:52.671Z	"eventVersion": "2.1",

Conclusion: In this, we have created lambda function successfully. Then we have also created s3 bucket successfully. Then I have edited the setting by setting timeout for 1 sec and adding a description. Then created a event name ‘our-tester’. then selected that event for test. Then deployed it. Thus deployed successfully. Then we have added a trigger in which we added a s3 bucket which we created. Then we have added a print message. Then again deployed the code after uploading an image in a s3 bucket. This given a status code 200 in response with that message which we added in code. Then in cloudwatch it given detailed logs of it.