

Experiment 1

Exp 1a: Prerequisites

Xampp

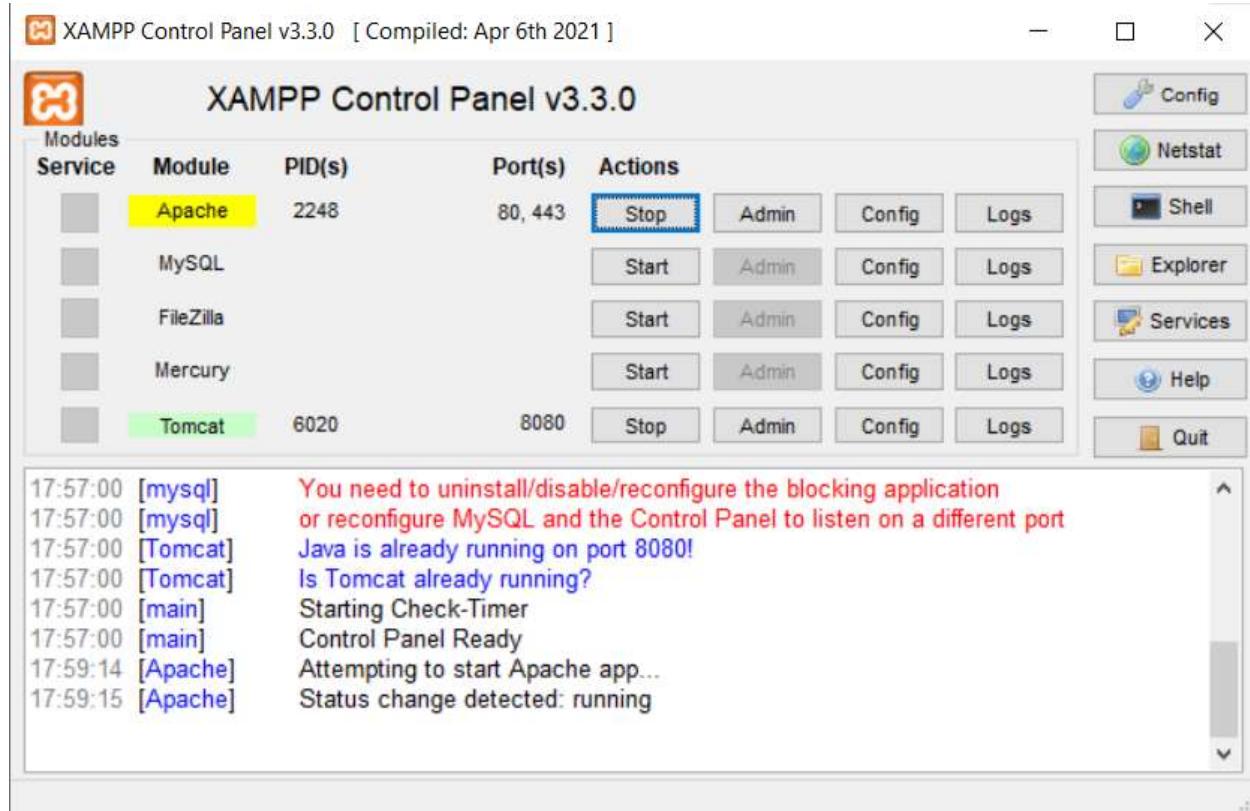
1. Go to the official website of Xampp. <https://www.apachefriends.org/download.html>. Select the suitable version and complete the installation.

The screenshot shows the Apache Friends download page. At the top, there are navigation links: Apache Friends, Download, Hosting, Community, About, Search, and EN. Below the header, a large "Download" button is visible. To its right, a "Documentation/FAQs" box contains text about the lack of a manual and links to forums and Stack Overflow. The main content area displays a table for XAMPP for Windows versions 8.0.30, 8.1.25, and 8.2.12. The table includes columns for Version, Checksum (md5, sha1), and Size (144 Mb, 148 Mb, 149 Mb). Each row has a "Download (64 bit)" button. Below the table, there are links for Requirements and More Downloads, and a note about unsupported platforms (Windows XP or 2003).

2. Once installation is complete, open the Xampp control panel. To host a php project locally we require to start the apache server.

The screenshot shows the XAMPP Control Panel v3.3.0 window. The title bar reads "XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]". The main interface lists services: Apache, MySQL, FileZilla, Mercury, and Tomcat. The Tomcat service is highlighted with a green background. The Apache service is listed under the "Service" column. The "Actions" column for Apache contains buttons for Start, Admin, Config, and Logs. The MySQL service also has similar action buttons. On the right side of the panel, there are icons for Config, Netstat, Shell, Explorer, Services, Help, and Quit. A status message at the bottom left indicates that MySQL is unable to start due to port 3306 being used by another process. The bottom right corner shows a "Control Panel Ready" message.

3. Click on start in front of apache and wait for the server to start.



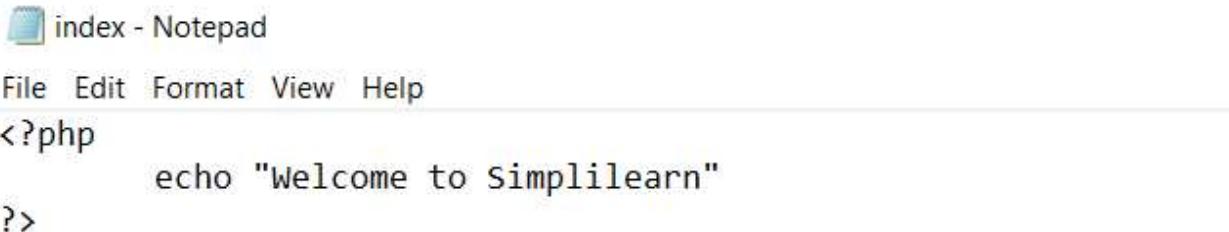
4. We can now set up the php project. In the xampp folder in C drive there will be a folder named htdocs. Every project that we want to host locally should be present in htdocs.

| This PC > Local Disk (C:) > xampp > | | | |
|-------------------------------------|------------------|-------------|------|
| Name | Date modified | Type | Size |
| anonymous | 01-08-2024 21:02 | File folder | |
| apache | 01-08-2024 21:02 | File folder | |
| cgi-bin | 01-08-2024 21:09 | File folder | |
| contrib | 01-08-2024 21:02 | File folder | |
| FileZillaFTP | 01-08-2024 21:09 | File folder | |
| htdocs | 01-08-2024 21:18 | File folder | |
| img | 01-08-2024 21:02 | File folder | |
| install | 01-08-2024 21:09 | File folder | |
| licenses | 01-08-2024 21:02 | File folder | |
| locale | 01-08-2024 21:02 | File folder | |
| mailoutput | 01-08-2024 21:02 | File folder | |
| mailtodisk | 01-08-2024 21:02 | File folder | |
| MercuryMail | 01-08-2024 21:09 | File folder | |
| mysql | 01-08-2024 21:03 | File folder | |
| perl | 01-08-2024 21:05 | File folder | |

5. We will create a project named demo inside htdocs. This demo folder will contain our php source code.

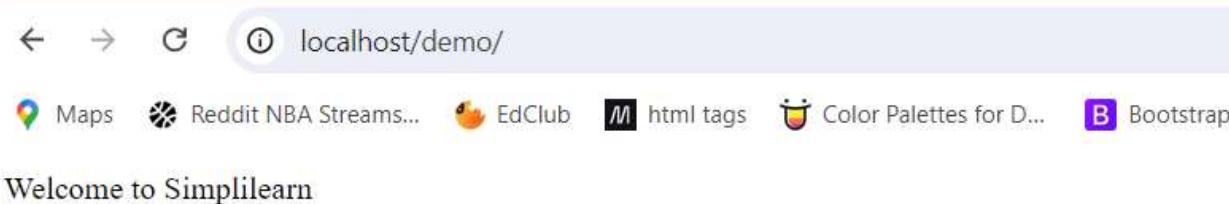
| This PC > Local Disk (C:) > xampp > htdocs | | | |
|--|------------------|-----------------------|-------|
| Name | Date modified | Type | Size |
| dashboard | 01-08-2024 21:02 | File folder | |
| demo | 01-08-2024 21:19 | File folder | |
| img | 01-08-2024 21:02 | File folder | |
| webalizer | 01-08-2024 21:02 | File folder | |
| xampp | 01-08-2024 21:02 | File folder | |
| applications | 15-06-2022 21:37 | Chrome HTML Do... | 4 KB |
| bitnami | 15-06-2022 21:37 | Cascading Style Sh... | 1 KB |
| favicon | 16-07-2015 21:02 | Icon | 31 KB |
| index | 16-07-2015 21:02 | PHP Source File | 1 KB |
| program | 01-08-2024 21:17 | PHP Source File | 1 KB |

6. Open the notepad and write a simple php script. Save this file inside the demo folder.



```
<?php
    echo "Welcome to Simplilearn"
?>
```

7. Go to your web browser and type “localhost/project_folder_name”, “localhost/demo” in this case. And we can see that our php script has been hosted locally using Xampp.



S3

1. Go to the AWS academy lab and search for s3 service.

The screenshot shows the AWS Academy lab interface. A search bar at the top contains the query 's3'. Below it, a sidebar lists categories like Services, Features, Resources, Documentation, Knowledge Articles, Marketplace, Blogs, Events, and Tutorials. The main content area displays search results for 's3' under the 'Services' category. The first result is 'S3' (Scalable Storage in the Cloud), which is highlighted with a blue box. Other results include 'S3 Glacier' (Archive Storage in the Cloud) and 'AWS Snow Family' (Large Scale Data Transport). Below these, there are sections for 'Features' (Imports from S3, Feature spotlight) and 'Top features' (Buckets, Storage Lens dashboards, Batch Operations, S3 Express One Zone, S3 Access Grants). On the right side, there's a sidebar with options to 'Create application' and 'Add widgets', along with account and region selection dropdowns.

2. In the s3 service select the create bucket option.

The screenshot shows the Amazon S3 service page. The main heading is 'Amazon S3' with the subtext 'Store and retrieve any amount of data from anywhere'. Below this, a paragraph explains that Amazon S3 is an object storage service. To the right, a large call-to-action button labeled 'Create a bucket' is highlighted with a yellow box. To the left, a section titled 'How it works' includes a video thumbnail titled 'Introduction to Amazon S3' with a play button icon. To the right, there are sections for 'Pricing' (which states there are no minimum fees), 'Estimate your monthly bill using the AWS Simple Monthly Calculator', and 'View pricing details'. At the bottom, there are links for 'Documentation' and 'Feedback'.

3. Select the name for bucket. Enter other details. Uncheck the block all public access option. Checking this box will give the 403 error when we try to access the website.

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.

⚠️ Turning off block all public access might result in this bucket and the objects within becoming public
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

4. Use default setting for other options. Once configuration is completed, click on create bucket.

Successfully created bucket "www.d15c49.com"
To upload files and folders, or to configure additional bucket settings, choose [View details](#).

Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type [Info](#)
 Server-side encryption with Amazon S3 managed keys (SSE-S3)
 Server-side encryption with AWS Key Management Service keys (SSE-KMS)
 Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)
 Secure your objects with two separate layers of encryption. For details on pricing, see [DSSE-KMS pricing](#) on the Storage tab of the [Amazon S3 pricing page](#).

Bucket Key
 Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. [Learn more](#)
 Disable
 Enable

Advanced settings

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

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

5. You can see the buckets created in General purpose buckets.

| General purpose buckets (1) Info All AWS Regions | | | |
|--|---------------------------------|---|--------------------------------------|
| Buckets are containers for data stored in S3. | | | |
| <input type="text" value="Find buckets by name"/> ✖ 1 ➡ ⓘ | | | |
| Name | AWS Region | IAM Access Analyzer | Creation date |
| www.d15c49.com | US East (N. Virginia) us-east-1 | View analyzer for us-east-1 | August 8, 2024, 18:53:28 (UTC+05:30) |

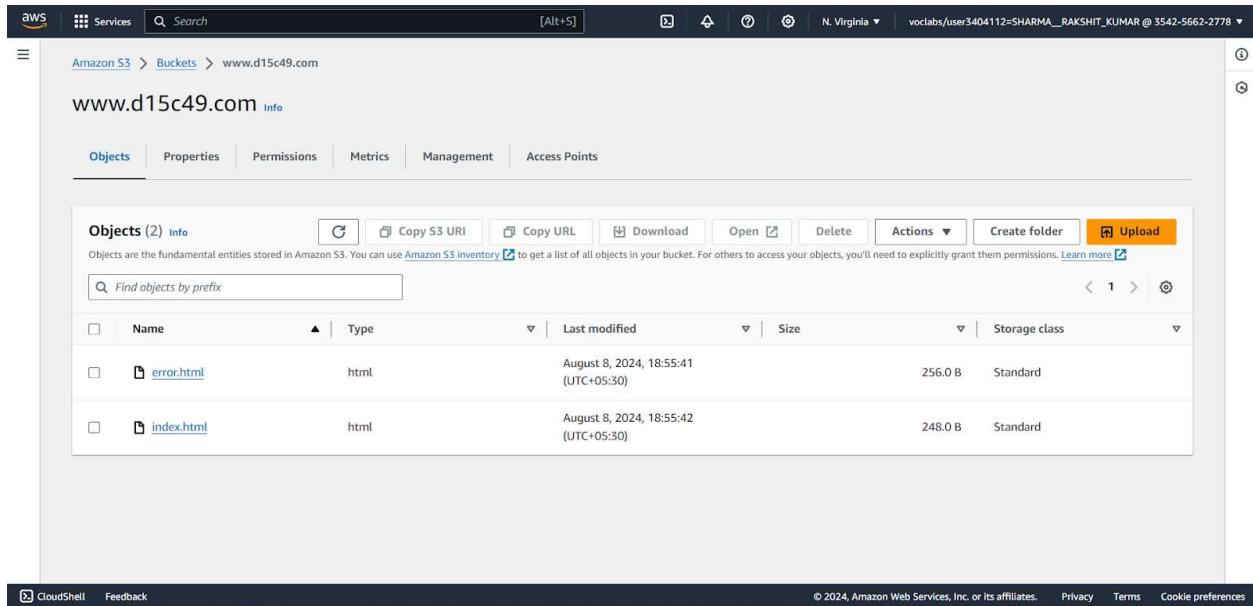
6. Open the bucket, now we will upload files in our bucket. Select the upload button

The screenshot shows the AWS S3 console interface. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, 'Search' bar, and account information: 'N. Virginia' and 'voclabs/user3404112=SHARMA_RAKSHIT_KUMAR @ 3542-5662-2778'. Below the navigation is a breadcrumb trail: 'Amazon S3 > Buckets > www.d15c49.com'. The main area is titled 'www.d15c49.com info'. A sub-header 'Objects (0) Info' is present. A toolbar at the top of this section includes 'Copy', 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Actions', 'Create folder', and the 'Upload' button, which is highlighted with a yellow border. Below the toolbar is a search bar labeled 'Find objects by prefix'. A table header row shows columns for 'Name', 'Type', 'Last modified', 'Size', and 'Storage class'. A message 'No objects' indicates there are no files in the bucket. At the bottom of the table area is a large orange 'Upload' button. The footer of the page includes links for 'CloudShell', 'Feedback', and copyright information: '© 2024, Amazon Web Services, Inc. or its affiliates.' and 'Privacy Terms Cookie preferences'.

7. Select the files you want to upload in the bucket. Click on the upload button to upload files.

The screenshot shows the 'Upload' dialog box from the AWS S3 console. At the top, it says 'Files and folders (2 Total, 504.0 B)' and 'All files and folders in this table will be uploaded.' Below this is a search bar 'Find by name' and a table with two items: 'index.html' (text/html). The table has columns for 'Name', 'Folder', and 'Type'. Underneath the table is a 'Destination' section with 'Destination' set to 's3://www.d15c49.com'. It includes a 'Destination details' panel with the note 'Bucket settings that impact new objects stored in the specified destination.' At the bottom of the dialog are 'Cancel' and 'Upload' buttons, with 'Upload' being orange. The footer of the dialog includes 'CloudShell', 'Feedback', and copyright information: '© 2024, Amazon Web Services, Inc. or its affiliates.' and 'Privacy Terms Cookie preferences'.

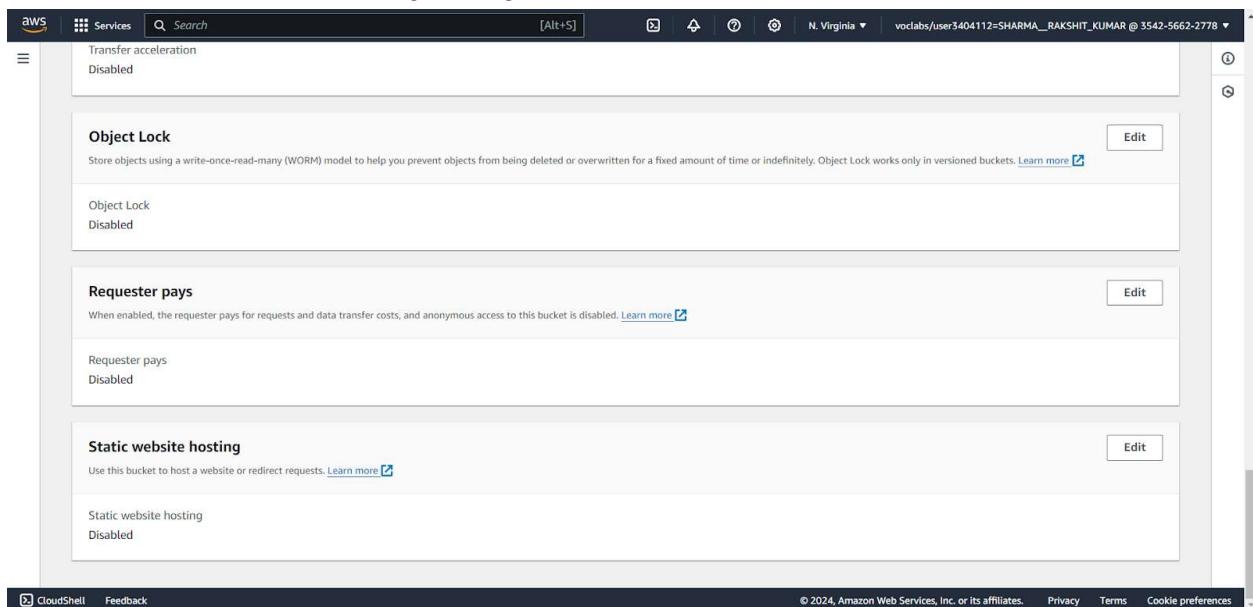
8. You can see all the files uploaded in the bucket. Now go to the properties section to enable the static hosting option for the bucket.



The screenshot shows the AWS S3 console interface. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, search bar, and account information ('N. Virginia' and 'vocabs/user3404112=SHARMA_RAKSHIT_KUMAR @ 3542-5662-2778'). Below the navigation is a breadcrumb trail: 'Amazon S3 > Buckets > www.d15c49.com'. The main area is titled 'www.d15c49.com Info' with tabs for 'Objects', 'Properties', 'Permissions', 'Metrics', 'Management', and 'Access Points'. The 'Objects' tab is selected, showing a list of 2 items:

| Name | Type | Last modified | Size | Storage class |
|------------|------|--------------------------------------|---------|---------------|
| error.html | html | August 8, 2024, 18:55:41 (UTC+05:30) | 256.0 B | Standard |
| index.html | html | August 8, 2024, 18:55:42 (UTC+05:30) | 248.0 B | Standard |

9. Edit the static website hosting setting.



The screenshot shows the 'Properties' section of the AWS S3 console for the 'www.d15c49.com' bucket. On the left, there's a sidebar with 'Transfer acceleration' set to 'Disabled'. The main area has several sections:

- Object Lock**: Describes Object Lock using a WORM model. Status: 'Object Lock' is 'Disabled'.
- Requester pays**: Describes requester pays for requests and data transfer costs. Status: 'Requester pays' is 'Disabled'.
- Static website hosting**: Allows hosting a website or redirecting requests. Status: 'Static website hosting' is 'Disabled'.

At the bottom, there are 'Edit' buttons for each of these settings.

10. Enable the static website hosting, select the index files and error files(optional). These will be the files you have uploaded in the bucket

The screenshot shows the 'Static website hosting' configuration for a bucket. The 'Enable' radio button is selected under 'Static website hosting'. Under 'Hosting type', the 'Host a static website' radio button is selected, with a note explaining that the bucket endpoint can be used as the web address. Below this, the 'Index document' field contains 'index.html' and the 'Error document - optional' field contains 'error.html'. A note at the bottom states that for public access, content must be publicly readable. The URL in the browser bar is <https://docs.aws.amazon.com/console/s3/hostingstaticwebsite>.

11. Now we need to change the bucket policy in permissions. Ensure that block public access is off. Edit the bucket policy.

The screenshot shows the 'Bucket policy' settings for a bucket. The 'Block all public access' setting is set to 'Off'. The 'Bucket policy' section displays a note stating that no policy is present. The URL in the browser bar is <https://docs.aws.amazon.com/console/s3/hostingstaticwebsite>.

12. You can see examples of bucket policy. Use the appropriate bucket policy and enter your bucket name in it.

The screenshot shows the 'Edit bucket policy' page in the AWS Management Console. The policy is defined as follows:

```

1  2 ▼ {
2  3   "Version": "2012-10-17",
2  4   "Statement": [
2  5     {
2  6       "Sid": "PublicReadGetObject",
2  7       "Effect": "Allow",
2  8       "Principal": "*",
2  9       "AWS": "*",
2 10      },
2 11      {
2 12        "Action": "s3:GetObject",
2 13        "Resource": "arn:aws:s3:::www.d15c49.com/*"
2 14      }
2 15    }
  
```

On the right side, there are buttons for 'Edit statement', 'Remove', 'Add actions', 'Choose a service', and dropdown menus for 'Included' (S3) and 'Available' (AMP). The bottom of the screen shows standard AWS navigation links like CloudShell, Feedback, and a copyright notice.

13. Once the bucket policy is added, you can see the link for website in static website hosting.

The screenshot shows the 'Static website hosting' configuration page. It indicates that the static website hosting is 'Enabled'. Under 'Hosting type', it shows 'Bucket hosting'. The 'Bucket website endpoint' section displays the generated URL: <http://www.d15c49.com.s3-website-us-east-1.amazonaws.com>.

14. On clicking the link you will be redirected to the hosted website.

A screenshot of a web browser window. The address bar shows the URL: [d15c49.com.s3-website-us-east-1.amazonaws.com](http://www.d15c49.com.s3-website-us-east-1.amazonaws.com). Below the address bar, there are several browser extensions or toolbars visible.

Hello this is my website

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

1. Open the AWS account and search for Cloud9. Click on create environment.

The screenshot shows the AWS Cloud9 landing page. At the top, there's a navigation bar with the AWS logo, 'Services', a search bar, and a user profile. Below the header, the title 'AWS Cloud9' is prominently displayed, followed by the subtitle 'A cloud IDE for writing, running, and debugging code'. A brief description explains that AWS Cloud9 allows you to write, run, and debug your code with just a browser. It highlights immediate access to a rich code editor, integrated debugger, and built-in terminal with preconfigured AWS CLI. On the right side, a large button labeled 'Create environment' is highlighted with a yellow box. Below the main title, there's a section titled 'How it works' containing a box of text and a 'Getting started' sidebar with several links. At the bottom, there are links for 'CloudShell', 'Feedback', and copyright information.

2. Enter the name and other required configuration for creating an environment. In network settings, using the AWS system manager gives an error while creating the environment. It states there was an error creating IAM resources needed for SSM.

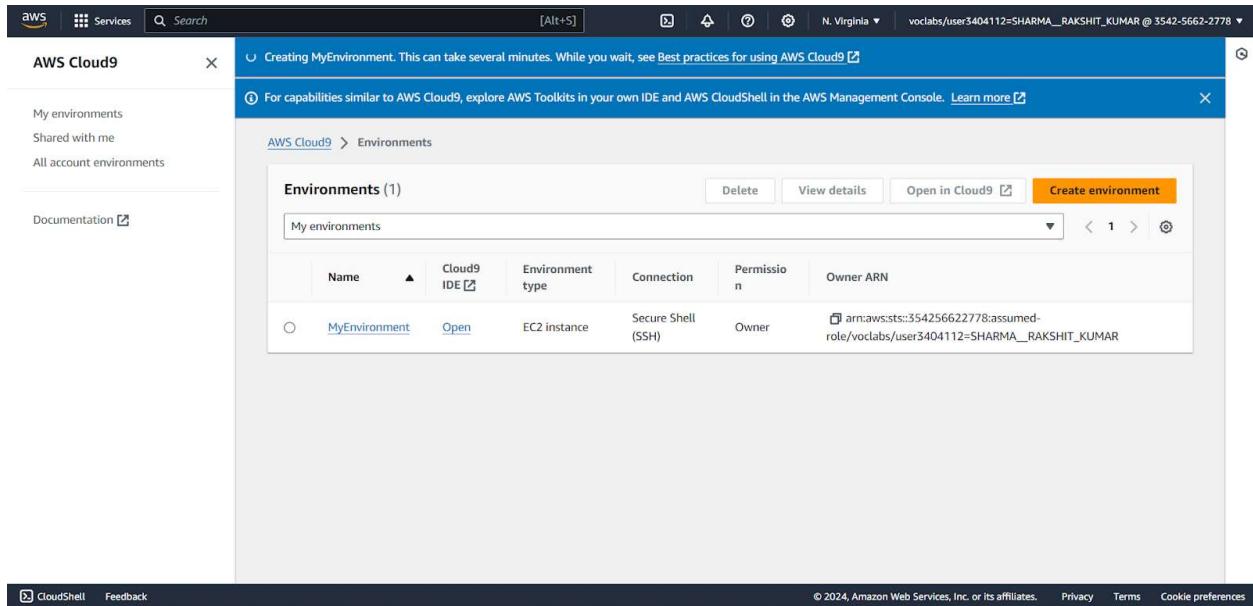
The screenshot shows the 'Create environment' configuration page. At the top, there's a 'Timeout' setting set to '30 minutes'. Below that is a 'Network settings' section with two options: 'AWS Systems Manager (SSM)' (selected) and 'Secure Shell (SSH)'. There are also sections for 'VPC settings' and 'Tags - optional'. At the bottom, a callout box provides information about IAM resources being created: 'AWS Service Role for AWS Cloud9', 'AWS Cloud9 SSM Access Role', and 'AWS Cloud9 SSM Instance Profile'. The page includes standard AWS footer links for 'CloudShell', 'Feedback', and copyright information.

The screenshot shows the AWS Cloud9 VPC settings page. At the top, there's a header with the AWS logo, a search bar, and navigation links. Below the header, there's a section titled "Tags - optional" with a note about tags being optional labels for AWS resources. A callout box highlights "The following IAM resources will be created in your account" with a list of three items: "AWSServiceRoleForAWSCloud9", "AWSCloud9SSMAccessRole", and "AWSCloud9SSMInstanceProfile". At the bottom right of the main content area are "Cancel" and "Create" buttons. Below the main content, there are three error messages in red-bordered boxes: 1. "There was an error creating the IAM resources needed for SSM connection." 2. "You don't have the permission required to perform this operation. Ask your administrator to give you permissions." 3. "User: arn:aws:sts::354256622778:assumed-role/voclabs/user3404112=SHARMA_RAKSHIT_KUMAR is not authorized to perform: iam:CreateRole on resource: arn:aws:iam::354256622778:role/service-role/AWSCloud9SSMAccessRole because no identity-based policy allows the iam:CreateRole action".

3. Use the Secure Shell option in Network settings.

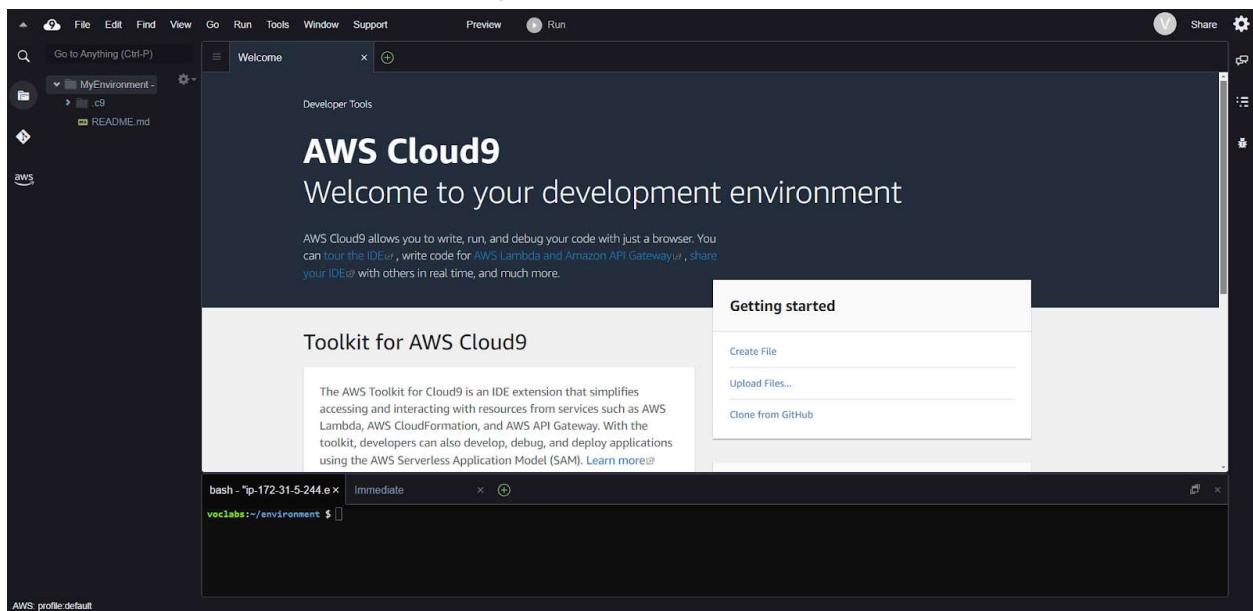
The screenshot shows the AWS Cloud9 Network settings page. At the top, there's a header with the AWS logo, a search bar, and navigation links. Below the header, there's a "Timeout" section with a dropdown menu set to "30 minutes". Under "Network settings", there's a "Connection" section where the "Secure Shell (SSH)" option is selected, indicated by a blue circle and the text "Accesses environment directly via SSH, opens inbound ports.". Below this, there's a "VPC settings" section with a link to "Info". A callout box highlights "The following IAM resources will be created in your account" with a list of three items: "AWSServiceRoleForAWSCloud9", "AWSCloud9SSMAccessRole", and "AWSCloud9SSMInstanceProfile". At the bottom right of the main content area are "Cancel" and "Create" buttons. Below the main content, there are two error messages in red-bordered boxes: 1. "How long Cloud9 can be inactive (no user input) before auto-hibernating. This helps prevent unnecessary charges." 2. "User: arn:aws:sts::354256622778:assumed-role/voclabs/user3404112=SHARMA_RAKSHIT_KUMAR is not authorized to perform: iam:CreateRole on resource: arn:aws:iam::354256622778:role/service-role/AWSCloud9SSMAccessRole because no identity-based policy allows the iam:CreateRole action".

4. Once the configuration is complete, click on create environment to create a Cloud9 environment.



The screenshot shows the AWS Cloud9 Environments page. On the left, there's a sidebar with 'AWS Cloud9' and links for 'My environments', 'Shared with me', and 'All account environments'. Below that is a 'Documentation' link. The main content area has a header 'Creating MyEnvironment. This can take several minutes. While you wait, see Best practices for using AWS Cloud9'. It includes a note about AWS Toolkits and a 'Learn more' link. Below this is a table titled 'Environments (1)'. The table has columns: Name, Cloud9 IDE, Environment type, Connection, Permission, and Owner ARN. A single row is shown for 'MyEnvironment', which is an 'Open' EC2 instance connected via Secure Shell (SSH), owned by the user, and has the ARN: arn:aws:sts::354256622778:assumed-role/voclabs/user3404112=SHARMA_RAKSHIT_KUMAR. There are buttons for 'Delete', 'View details', 'Open in Cloud9', and 'Create environment' (which is highlighted in orange). At the bottom of the page are links for 'CloudShell', 'Feedback', and copyright information: '© 2024, Amazon Web Services, Inc. or its affiliates.' and 'Privacy Terms Cookie preferences'.

5. Click on the environment name to open the created Cloud9 Environment.



The screenshot shows the AWS Cloud9 IDE interface. On the left is a sidebar with a file tree showing 'MyEnvironment' (with 'c9' and 'README.md') and an 'aws' icon. The main area has a title 'AWS Cloud9' and subtitle 'Welcome to your development environment'. Below that is a 'Developer Tools' section. In the center, there's a 'Toolkit for AWS Cloud9' panel with a sub-section 'Getting started' containing 'Create File', 'Upload Files...', and 'Clone from GitHub' options. At the bottom is a terminal window with a bash prompt: 'bash -> ip-172-31-5-244.e x | Immediate < >'. The terminal shows the command 'voclabs:~/environment \$'. The status bar at the bottom left says 'AWS profile default'.

6. Open the aws account and search for IAM service.

The screenshot shows the AWS IAM Dashboard. On the left, a sidebar lists navigation options like Dashboard, Access management, and Access reports. The main area displays 'Security recommendations' with a warning about adding MFA for the root user and a note about root user access keys. It also shows 'IAM resources' with counts of 0 for User groups, 0 for Users, 2 for Roles, 0 for Policies, and 0 for Identity providers. A 'What's new' section is present. On the right, there's a sidebar for the 'AWS Account' (Account ID: 975050293750) and a 'Quick Links' section for managing security credentials.

7. Go to the users tab. Click on create user to create a new user.

The screenshot shows the 'Users' page in the AWS IAM service. The sidebar is identical to the previous dashboard. The main area shows a table titled 'Users (0)' with one row header: 'User name'. A prominent orange 'Create user' button is located at the top right of the table area. A note below the table states 'No resources to display'.

8. Give the user name, select the “provide user access” checkbox. Also select the option “I want to create an IAM user”. Otherwise we will have to enable the Identity center and specify a user there.

9. Next click on add user to group. If you do not have a existing group, select create group.

10. Give the group name and policies if required, and create a group.

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

User group name
Enter a meaningful name to identify this group.

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

Permissions policies (947)

Filter by Type

| <input type="checkbox"/> Policy name | Type | Use... | Description |
|---|-----------------|--------|--------------------------------------|
| <input type="checkbox"/> AdministratorAccess | AWS managed ... | None | Provides full access to AWS services |
| <input type="checkbox"/> AdministratorAcce... | AWS managed | None | Grants account administrative perm |
| <input type="checkbox"/> AdministratorAcce... | AWS managed | None | Grants account administrative perm |
| <input type="checkbox"/> AlexaForBusinessD... | AWS managed | None | Provide device setup access to Alex |
| <input type="checkbox"/> AlexaForBusinessF... | AWS managed | None | Grants full access to AlexaForBusin |
| <input type="checkbox"/> AlexaForBusinessG... | AWS managed | None | Provide gateway execution access t |

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

11. Once the group is created, select the group in which the user should be added.

D15C user group created.

Step 2 Set permissions

Step 3 Review and create

Step 4 Retrieve password

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 (1/1)

| <input checked="" type="checkbox"/> Group name | Users | Attached policies | Created |
|--|-------|-------------------|------------------|
| <input checked="" type="checkbox"/> D15C | 0 | - | 2024-08-08 (Now) |

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

12. Recheck all the configuration and details of the user and click on create user.

The screenshot shows the AWS IAM 'Create New User' wizard at Step 3: Review and create. The user details are as follows:

| | | | |
|-----------|---------|------------------------|-----------------|
| User name | Rakshit | Console password type | Custom password |
| | | Require password reset | Yes |

The Permissions summary table shows:

| Name | Type | Used as |
|-----------------------|-------------|--------------------|
| D15C | Group | Permissions group |
| IAMUserChangePassword | AWS managed | Permissions policy |

The Tags section is empty, indicating no tags have been added.

At the bottom right, there are 'Cancel', 'Previous', and a prominent orange 'Create user' button.

The screenshot shows the AWS IAM 'Create New User' wizard at Step 4: Retrieve password. It displays the retrieved password details:

Console sign-in URL: <https://975050293750.signin.aws.amazon.com/console>

User name: Rakshit

Console password: [REDACTED] [Show](#)

At the bottom right, there are 'Email sign-in instructions' (button), 'Cancel', 'Download .csv file', and a prominent orange 'Return to users list' button.

The screenshot shows the AWS IAM 'Create New User' wizard completed successfully. A green banner at the top says 'User created successfully'. The message below states: 'You can view and download the user's password and email instructions for signing in to the AWS Management Console.' There is a 'View user' button on the right.

13. After creation of the user, go to the user groups tab and select the group in which the user has been added. Navigate to the permissions tab. Click on add permissions and attach policy.

The screenshot shows the AWS IAM User Groups page. On the left, a sidebar menu includes 'Identity and Access Management (IAM)', 'Access management' (with 'User groups' selected), and 'Access reports'. The main content area displays 'D15C Info' with a 'Summary' section showing the user group name 'D15C', creation time 'August 08, 2024, 22:18 (UTC+05:30)', and ARN 'arn:aws:iam::975050293750:group/D15C'. Below this are tabs for 'Users (1)', 'Permissions', and 'Access Advisor'. The 'Users' tab shows a table with one user, 'Rakshit', listed under 'User name'. The 'Permissions' tab is currently selected. At the bottom, there are links for 'CloudShell' and 'Feedback'.

14. Search for the “AWSCloud9EnvironmentMember” policy and attach it.

The screenshot shows the 'Add permissions' page for the D15C user group. The title is 'Attach permission policies to D15C'. A section titled 'Current permissions policies (0)' is shown. Below it, a search bar and filter options ('Filter by Type') are used to search for the 'AWSCloud9EnvironmentMember' policy. The search results table lists four policies: 'AWSCloud9Administrator', 'AWSCloud9EnvironmentMember' (which is checked), 'AWSCloud9SSMInstanceProfile', and 'AWSCloud9User'. The 'AWSCloud9EnvironmentMember' row is highlighted. At the bottom right are 'Cancel' and 'Attach policies' buttons.

Experiment 2

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

1. Login to your AWS account and search for Elastic Beanstalk.

The screenshot shows the AWS Elastic Beanstalk landing page. At the top, there's a navigation bar with the AWS logo, a search bar, and a 'Get started' button. Below the header, the service name 'Amazon Elastic Beanstalk' is prominently displayed with the tagline 'End-to-end web application management.' A detailed description follows, mentioning supported languages and technologies. To the right, a 'Get started' box contains a 'Create application' button. Further down, a 'Pricing' section states that there's no additional charge. The main content area features a 'Get started' section with a detailed description of the deployment process. At the bottom, there's a 'Benefits and features' section and a 'Getting started' link. The footer includes standard AWS links for CloudShell, Feedback, and legal information.

2. Click on create application. Enter your application names and other basic details.

The screenshot shows the 'Create application' wizard. Step 3 is completed, showing options for networking, database, and tags. Step 4 is optional, for configuring instance traffic and scaling. Step 5 is optional, for updates, monitoring, and logging. Step 6 is the review step. In the application information section, the application name is set to 'MyWebsite'. In the environment information section, the environment name is 'MyWebsite-env'. The domain field is left blank with a placeholder 'Leave blank for autogenerated value'. The bottom navigation bar includes CloudShell, Feedback, and links to Privacy, Terms, and Cookie preferences.

3. In the platform select PHP among other options. Platform branch and platform version will be entered automatically.

Platform Info

Platform type

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

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

Platform

PHP

Platform branch

PHP 8.3 running on 64bit Amazon Linux 2023

Platform version

4.3.2 (Recommended)

4. Keep the other setting to default and click on next. In service access click on “Use an existing service role”.

Configure service access [Info](#)

Step 1
[Configure environment](#)

Step 2
Configure service access

Step 3 - optional
[Set up networking, database, and tags](#)

Step 4 - optional
[Configure instance traffic and scaling](#)

Step 5 - optional
[Configure updates, monitoring, and logging](#)

Step 6
[Review](#)

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

Service role

Create and use new service role
 Use an existing service role
Existing service roles
Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.
aws-elasticbeanstalk-service-role

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

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

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

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

5. Go to EC2 service and click on Key pair to create a new key pair. Give the key pair a name and select the type as RSA. For private key file format select .pem.

Key pair
A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name

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

Key pair type [Info](#)
 RSA ED25519

Private key file format
 .pem For use with OpenSSH
 .ppk For use with PuTTY

Tags - optional
No tags associated with the resource.
[Add new tag](#)
You can add up to 50 more tags.

[Cancel](#) [Create key pair](#)

4. Come back to Elastic Beanstalk configuration. Select the newly created key pair from the dropdown menu. Also select the EC2 instance profile. Click on next.

Configure service access [Info](#)

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

Service role
 Create and use new service role
 Use an existing service role
Existing service roles
Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.

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

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

[View permission details](#)

[Cancel](#) [Skip to review](#) [Previous](#) [Next](#)

5. Skip to review. Review all the configurations and click on submit. Wait for the “Environment successfully launched” message.

The screenshot shows two stacked screenshots of the AWS Elastic Beanstalk console. The top screenshot displays the 'Platform software' and 'Environment properties' sections. The 'Platform software' section includes settings like Lifecycle (false), Log streaming (Deactivated), and Max execution time (60). The 'Environment properties' section shows a table with one row: 'No environment properties' (Key: No environment properties, Value: There are no environment properties defined). The bottom screenshot shows a success message: 'Environment successfully launched.' It details the environment configuration: Application: Mywebsite, Environment: MyWebsite-env, Health: Pending, Domain: MyWebsite-env.eba-upbp4pj.us-east-1.elasticbeanstalk.com, Environment ID: e-xipkejhr8, Application name: Mywebsite, Platform: PHP 8.3 running on 64bit Amazon Linux 2023/4.3.2, Running version: -, Platform state: Supported. Below this, there's an 'Events' section showing 10 events.

6. Create a github repository with the source code to be deployed. Here I have forked an existing repository.

Rakshit23665 / aws-codepipeline-s3-codedeploy-linux-2.0

Code Pull requests Actions Projects Wiki Security Insights Settings

Type to search

aws-codepipeline-s3-codedeploy-linux-2.0 (Public)

forked from imoisharma/aws-codepipeline-s3-codedeploy-linux-2.0

Pin Watch Fork Star

Code About

This branch is up to date with imoisharma/aws-codepipeline-s3-codedeploy-linux-2.0:master.

Contribute Sync fork

imoisharma Update README.md 8fd5da5 · 3 years ago 20 Commits

.github Adding template 7 years ago

dist Added dist folder 9 years ago

scripts s3 setup and s3 set cache control scripts 3 years ago

7. Go to CodePipeline service and click on create pipeline.

aws Services Search [Alt+S]

Developer Tools CodePipeline

Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Deploy • CodeDeploy

Pipeline • CodePipeline
Getting started
Pipelines
Settings

Go to resource Feedback

Successfully deleted
Pipeline: joshi2611 has been deleted.

Introducing the new V2 pipeline type with improved release safety, pipeline triggers, parameterized pipelines, and a new billing model. Learn more

Pipelines Info

Name Latest execution status Latest source revisions Latest execution started Most recent executions

No results
There are no results to display.

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

8. Give the pipeline a name. Role name will be generated automatically based on pipeline name

Choose pipeline settings Info

Step 1 of 5

Pipeline settings

Pipeline name

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

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.

Role name

Type your service role name:

- Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

Variables

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

No variables defined at the pipeline level in this pipeline.

[Add variable](#)

You can add up to 50 variables.

- ⓘ The first pipeline execution will fail if variables have no default values.

[▶ Advanced settings](#)

Cancel

Next

9. Select Github(Version 2) as a source provider. Click on connect to github to create a new connection if you don't have one.

The screenshot shows the 'Add source stage' configuration screen in AWS CodePipeline. The left sidebar lists steps: Step 1 (Choose pipeline settings), Step 2 (Add source stage, currently selected), Step 3 (Add build stage), Step 4 (Add deploy stage), and Step 5 (Review). The main panel is titled 'Source' and contains the following fields:

- Source provider:** GitHub (Version 2) is selected from a dropdown menu.
- New GitHub version 2 (app-based) action:** A callout box provides information about using GitHub Apps to access repositories.
- Connection:** A search bar and a 'Connect to GitHub' button.
- Repository name:** A search bar for selecting a repository in the GitHub account.
- Default branch:** A search bar for specifying the default branch.
- Output artifact format:** A dropdown menu showing 'CodePipeline default' and 'Full clone'.

At the bottom of the page, there are links for CloudShell, Feedback, and navigation icons. The footer includes copyright information for Amazon Web Services and links for Privacy, Terms, and Cookie preferences.

10. Give the connection a name and click on Install a new app. After this click on install. Once installation is complete click on connect to establish a connection.

The screenshot shows two overlapping dialogs from the AWS Developer Tools interface.

Top Dialog (AWS):

- Header: "Developer Tools > ... > Create connection".
- Message: "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](#)".
- Section: "Connect to GitHub".
- Sub-section: "GitHub connection settings".
- Input: "Connection name" field containing "sample".
- Section: "GitHub Apps".
 - Description: "GitHub Apps create a link for your connection with GitHub. Install a new app and save this connection."
 - Input: Search bar with placeholder "Q 53777842" and "Install a new app" button.
- Footer: "Feedback", "Privacy", "Terms", "Cookie preferences".

Bottom Dialog (GitHub):

- Header: "Install on your personal account RakshitSharma".
- Text: "for these repositories:"
- Radio button: "All repositories" (selected).
 - Description: "This applies to all current and future repositories owned by the resource owner. Also includes public repositories (read-only)."
- Radio button: "Only select repositories".
 - Description: "Select at least one repository. Also includes public repositories (read-only)."
- Text: "with these permissions:"
- Checklist:
 - ✓ Read access to issues and metadata
 - ✓ Read and write access to administration, code, commit statuses, pull requests, and repository hooks
- Buttons: "Install" (green) and "Cancel".
- Text: "Next: you'll be directed to the GitHub App's site to complete setup."

11. Once the connection is established, you will get a success message. Select the repository containing the source code. Also select the branch(usually master). Be sure to select the no filter option in Trigger section. Click on next.

The screenshot shows the AWS CodePipeline interface for setting up a GitHub connection. At the top, there's a banner for 'New GitHub version 2 (app-based) action'. Below it, under 'Connection', a connection named 'arn:aws:codeconnections:us-east-1:975050293750:connection/69e74936-36' is selected, with a 'Connect to GitHub' button. A green box indicates the connection is 'Ready to connect'. Under 'Repository name', 'Rakshit25665/aws-codepipeline-s3-codedeploy-linux-2.0' is selected. Under 'Default branch', 'master' is selected. Under 'Output artifact format', 'CodePipeline default' is selected. At the bottom, the 'Trigger' section is shown with the title 'Trigger type'. It lists three options: 'No filter' (selected), 'Specify filter', and 'Do not detect changes'. The 'No filter' option is described as starting the pipeline on any push and cloning the HEAD.

Trigger

Trigger type
Choose the trigger type that starts your pipeline.

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

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

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

12. Skip the build stage and directly go to deploy stage. Select Elastic Beanstalk as Deploy provider. Select the Elastic Beanstalk application name that we created earlier. Click on next once done.

The screenshot shows the 'Deploy' stage configuration screen. It includes fields for 'Deploy provider' (set to 'AWS Elastic Beanstalk'), 'Region' (set to 'US East (N. Virginia)'), 'Input artifacts' (a dropdown menu), 'Application name' (set to 'Mywebsite'), 'Environment name' (set to 'MyWebsite-env'), and a checkbox for 'Configure automatic rollback on stage failure'. Navigation buttons 'Cancel', 'Previous', and 'Next' are at the bottom.

13. Review the configurations made and click on create pipeline.

The screenshot shows the 'Step 3: Add build stage' and 'Step 4: Add deploy stage' configuration screens. Step 3 has a 'Build action provider' section. Step 4 has sections for 'Deploy action provider' (set to 'AWS Elastic Beanstalk'), 'ApplicationName' (set to 'Mywebsite'), 'EnvironmentName' (set to 'MyWebsite-env'), and a 'Configure automatic rollback on stage failure' checkbox (set to 'Disabled'). Navigation buttons 'Cancel', 'Previous', and 'Create pipeline' (highlighted in orange) are at the bottom.

14. Once the pipeline is created you can go to the environments page(Elastic Beanstalk). The website is hosted on the link under domain column. Click on the link to go to the hosted website.

The screenshot shows two windows side-by-side. The left window is the AWS Elastic Beanstalk 'Environments' page, displaying a single environment named 'MyWebsite-env'. The right window is a web browser showing the deployed application at the URL 'mywebsite-env.eba-upbp4pj.us-east-1.elasticbeanstalk.com'. The browser title bar says 'Environments | Elastic Beanstalk'.

AWS Elastic Beanstalk Environments Page:

| Environment name | Health | Application | Platform | Domain | Running v... | Tier name | Date cr... |
|------------------|---------|-------------|------------------|------------------------------|------------------|-----------|------------|
| MyWebsite-env | No Data | Mywebsite | PHP 8.3 runni... | MyWebsite-env.eba-upbp4pj... | code-pipeline... | WebServer | August |

Browser View of Deployed Website:

Rakshit Sharma d15c

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

For next steps, read the AWS CodePipeline Documentation. Incedge 2020

Experiment 3

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

Step 1: Log in to your AWS Academy/personal account and launch 3 new Ec2 Instances. Select Ubuntu as AMI and t2.medium as Instance Type and create a key of type RSA with .pem extension and move the downloaded key to the new folder.We can use 3 Different keys or 1 common key also. Note: A minimum of 2 CPUs are required so Please select t2.medium and do not forget to stop the instance after the experiment because it is not available in the free tier.

Master:

The screenshot shows the AWS CloudFormation console interface. On the left, there's a navigation pane with 'AWS Lambda' selected. In the center, there's a large 'Create New Stack' button with the text 'Create New Stack'. Below it, there's a section titled 'Create a stack from template' with a 'Create New Stack' button. On the right, there's a 'Template' section with a 'Create New Template' button and a 'Upload Existing Template' section with a 'Choose File' button. At the bottom, there's a 'Next Step' button with the text 'Create stack'.

Worker:

Name: node

Application and OS Images (Amazon Machine Image)

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're looking for below.

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

Recents | Quick Start

Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux

Browse more AMIs Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type

Last updated less than a minute ago

Connect

Instance state ▾

Actions ▾

Launch instances ▾

Security group rule 4 (TCP, 10250, 0.0.0.0/0)

Type: Custom TCP, Protocol: TCP, Port range: 10250

Source type: Custom, Source: 0.0.0.0/0

Description - optional: e.g. SSH for admin desktop

Security group rule 5 (TCP, 30000-32767, 0.0.0.0/0)

Type: Custom TCP, Protocol: TCP, Port range: 30000-32767

Source type: Custom, Source: 0.0.0.0/0

Description - optional: e.g. SSH for admin desktop

Security group rule 6 (TCP, 0-65535, 0-65535)

Type: All TCP, Protocol: TCP, Port range: 0-65535

Step 2: After creating the instances click on Connect & connect all 3 instances and navigate to SSH Client.

Instances (4) Info Last updated less than a minute ago

Find Instance by attribute or tag (case-sensitive)

All states ▾

| <input type="checkbox"/> Name | Instance ID | Instance state | Instance type | Status check | Alarm status |
|--------------------------------------|---------------------|-------------------------|---------------|--------------------------------|-------------------------------|
| <input type="checkbox"/> nagios-host | i-06c955d252d68c106 | Terminated | t2.micro | ... | View alarms + |
| <input type="checkbox"/> master | i-03753a87170da35bc | Running | t2.medium | 2/2 checks passed | View alarms + |
| <input type="checkbox"/> node | i-0b9d7d0a41c70f7f0 | Running | t2.medium | Initializing | View alarms + |
| <input type="checkbox"/> node | i-0c4a87fdff3eb59e0 | Running | t2.medium | Initializing | View alarms + |

Connect to instance Info

Connect to your instance i-0a9a7aa3d967ebcd8 (node2) using any of these options

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

Instance ID: i-0a9a7aa3d967ebcd8 (node2)

- Open an SSH client.
- Locate your private key file. The key used to launch this instance is Rakshit.pem
- Run this command, if necessary, to ensure your key is not publicly viewable.
chmod 400 "Rakshit.pem"
- Connect to your instance using its Public DNS:
ec2-54-197-87-163.compute-1.amazonaws.com

Example:

```
ssh -i "Rakshit.pem" ubuntu@ec2-54-197-87-163.compute-1.amazonaws.com
```

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

Step 3: Now open the folder in the terminal 3 times for Master, Node1& Node 2 where our .pem key is stored and paste the Example command (starting with ssh -i) in the terminal.(ssh -i "Master_Ec2_Key.pem" ubuntu@ec2-54-196-129-215.compute-1.amazonaws.com) Master:

```
Expanded Security Maintenance for Applications is not enabled.
```

```
0 updates can be applied immediately.
```

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

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

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

```
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.
```

```
ubuntu@ip-172-31-81-239:~$ |
```

Step 4: Run on Master,Node 1, and Node 2 the below commands to install and setup Docker in Master, Node1, and Node2.

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add - curl -fsSL  
https://download.docker.com/linux/ubuntu/gpg | sudo tee /etc/apt/trusted.gpg.d/docker.gpg >  
/dev/null sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu  
$(lsb_release -cs) stable"
```

```
ubuntu@ip-172-31-81-239:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -  
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee  
/etc/apt/trusted.gpg.d/docker.gpg > /dev/null  
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu  
$(lsb_release -cs) stable"  
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).  
OK  
-----BEGIN PGP PUBLIC KEY BLOCK-----  
  
mQINBFit2ioBEADhWpZ8/wvZ6hUTiXOwQHXMAlaFHcPH9hAtr4F1y2+0YdbtMuth  
lqwp028Aqy+PRfVmSYMbjuQuu5byyKR01BbqYhuS3jtqQmljZ/bJvXqnmivXh  
38UuLa+z077PxxyQhu5BbqntTPQMfiyqEiU+BKbq2WmANUKQf+1AmZY/IruOXbnq  
L4C1+gJ8vfmQt99npCaxEjaNRVYf0S8QcixNzHUYnb6emjlAnyEVlZzeqo7XkL7  
UrWV5inawTSzwNvtjEjj4nJL8NsLwscpLPQUhTQ+7BbQXAwAmeHCUTQIVvvXqgw0N  
cmhh4HgeQscQHYg0JJjDVfoY5MucvglbIgCqfzAHW9jxmRL4qbMZj+b1XoePEtht  
ku4bIQN1X5P07fNWzIgaRL5Z4POXDDZTLIQ/E158j9kp4bnWRCJW0lya+f8ocodo  
vZZ+Doi+fy4D5ZGrL4XEcIQP/Lv5ufyf+kQtL/94VFYVJ0LeAv8W92KdgDkhTcTD  
G7c0tIkVEKNUq48b3aQ64NOZQW7FVjfoKwEZd0qPE72Pa45jrZzvUFxSpdiNk2tZ  
XYukHjlxxEgBdC/J3cMMNRE1F4NCA3ApfV1Y7/hTe0nmDuDYwr9/obA8t016Yljj  
q5rdkywPf4JF8mXUW5eCN1vAFHxe9ZLembhBtQmGxXnw9M+z6hWwc6ahmwARAQAB  
+tEh2NrZXtallmVs7WEzzSAo00UigZGVikSA8ZG9ia2Vv0GRvY2t1ci5ib2o+i0T3
```

```
sudo apt-get update
```

```
sudo apt-get install -y docker-ce
```

```
ubuntu@ip-172-31-81-239:~$ sudo apt-get update
sudo apt-get install -y docker-ce
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://download.docker.com/linux/ubuntu noble InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  containerd.io docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0
  pigz slirp4netns
Suggested packages:
```

```
sudo mkdir -p /etc/docker
cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"]
}
EOF
```

```
ubuntu@ip-172-31-81-239:~$ sudo mkdir -p /etc/docker
cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"]
}
EOF
{
  "exec-opts": ["native.cgroupdriver=systemd"]
}
```

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

```
ubuntu@ip-172-31-81-239:~$ sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker
Synchronizing state of docker.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable docker
```

Step 5: Run the below command to install Kubernets.

```
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o
/etc/apt/keyrings/kubernetes-apt-keyring.gpg echo 'deb
[signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
```

```
https://pkgs.k8s.io/core:/stable:/v1.31/deb/ | sudo tee /etc/apt/sources.list.d/kubernetes.list
```

```
ubuntu@ip-172-31-81-239:~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
https://pkgs.k8s.io/core:/stable:/v1.31/deb/ | sudo tee /etc/apt/sources.list.d/kubernetes.list
gpg: missing argument for option "--o"
-bash: /etc/apt/keyrings/kubernetes-apt-keyring.gpg: No such file or directory
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /'
```

```
sudo apt-get update
```

```
sudo apt-get install -y kubelet kubeadm kubectl
```

```
sudo apt-mark hold kubelet kubeadm kubectl
```

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

```
sudo apt-get install -y containerd
```

```
ubuntu@ip-172-31-81-239:~$ sudo apt-get install -y apt-transport-https ca-certificates curl
curl -fsSL https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -
echo "deb https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list
sudo apt-get update
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203).
ca-certificates set to manually installed.
curl is already the newest version (8.5.0-2ubuntu10.4).
curl set to manually installed.
The following NEW packages will be installed:
  apt-transport-https
0 upgraded, 1 newly installed, 0 to remove and 6 not upgraded.
Need to get 3974 B of archives.
After this operation, 35.8 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 apt-transport-https all 2.7.14build2 [3974 B]
Fetched 3974 B in 0s (0 B/s)
```

```
sudo mkdir -p /etc/containerd
```

```
sudo containerd config default | sudo tee /etc/containerd/config.toml
```

```
ubuntu@ip-172-31-81-239:~$ sudo mkdir -p /etc/containerd
sudo containerd config default | sudo tee /etc/containerd/config.toml
disabled_plugins = []
imports = []
oom_score = 0
plugin_dir = ""
required_plugins = []
root = "/var/lib/containerd"
state = "/run/containerd"
temp = ""
version = 2

[cgroup]
  path = ""

[debug]
  address = ""
  format = ""
  gid = 0
  level = ""
  uid = 0
```

```
sudo systemctl restart containerd
```

```
sudo systemctl enable containerd
```

sudo systemctl status containerd

```
ubuntu@ip-172-31-81-239:~$ sudo systemctl restart containerd
sudo systemctl enable containerd
sudo systemctl status containerd
● containerd.service - containerd container runtime
   Loaded: loaded (/usr/lib/systemd/system/containerd.service; enabled; preset: enabled)
   Active: active (running) since Thu 2024-10-03 12:03:56 UTC; 187ms ago
     Docs: https://containerd.io
     Main PID: 4388 (containerd)
       Tasks: 8
      Memory: 13.8M (peak: 14.1M)
        CPU: 47ms
       CGroup: /system.slice/containerd.service
               └─4388 /usr/bin/containerd

Oct 03 12:03:56 ip-172-31-81-239 containerd[4388]: time="2024-10-03T12:03:56.810259144Z" level=info msg="Start subscriber"
Oct 03 12:03:56 ip-172-31-81-239 containerd[4388]: time="2024-10-03T12:03:56.810436084Z" level=info msg="Start recoverer"
Oct 03 12:03:56 ip-172-31-81-239 containerd[4388]: time="2024-10-03T12:03:56.810483329Z" level=info msg="serving... addr"
Oct 03 12:03:56 ip-172-31-81-239 containerd[4388]: time="2024-10-03T12:03:56.810489545Z" level=info msg="Start event mo"
Oct 03 12:03:56 ip-172-31-81-239 containerd[4388]: time="2024-10-03T12:03:56.810583963Z" level=info msg="Start snapshot"
Oct 03 12:03:56 ip-172-31-81-239 containerd[4388]: time="2024-10-03T12:03:56.810599063Z" level=info msg="Start cni netw"
Oct 03 12:03:56 ip-172-31-81-239 containerd[4388]: time="2024-10-03T12:03:56.810605397Z" level=info msg="Start streamin"
Oct 03 12:03:56 ip-172-31-81-239 containerd[4388]: time="2024-10-03T12:03:56.810635592Z" level=info msg=serving... addr
Oct 03 12:03:56 ip-172-31-81-239 systemd[1]: Started containerd.service - containerd container runtime.
Oct 03 12:03:56 ip-172-31-81-239 containerd[4388]: time="2024-10-03T12:03:56.813064087Z" level=info msg="containerd suc
```

sudo apt-get install -y socat

```
ubuntu@ip-172-31-81-239:~$ sudo apt-get install -y socat
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz
  slirp4netns
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  socat
0 upgraded, 1 newly installed, 0 to remove and 6 not upgraded.
Need to get 374 kB of archives.
After this operation, 1649 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 socat amd64 1.8.0.0-4build3 [374 kB]
Fetched 374 kB in 0s (12.2 MB/s)
Selecting previously unselected package socat.
(Reading database ... 68148 files and directories currently installed.)
Preparing to unpack .../socat_1.8.0.0-4build3_amd64.deb ...
Unpacking socat (1.8.0.0-4build3) ...
S: /var/cache/apt/archives/socat_1.8.0.0-4build3_amd64.deb
```

Step 6: Initialize the Kubecluster .Now Perform this Command only for Master.

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

```
ubuntu@ip-172-31-82-119:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[kinit] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
W1003 12:37:31.290835 10158 checks.go:846] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent with that used by kubeadm. It is recommended to use "registry.k8s.io/pause:3.10" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-82-119 kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 172.31.82.119]
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key
[certs] Generating "etcd/ca" certificate and key
[certs] Generating "etcd/server" certificate and key
[certs] etcd/server serving cert is signed for DNS names [ip-172-31-82-119 localhost] and IPs [172.31.82.119 127.0.0.1 :1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-82-119 localhost] and IPs [172.31.82.119 127.0.0.1 :1]
```

Run this command on master and also copy and save the Join command from above.

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

```
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 172.31.82.119:6443 --token h4iisg.g9tdfmc88m9toefp \  
    --discovery-token-ca-cert-hash sha256:5b3c7cfdf8115a26f1e73b752820d2b27b84ed476b344aaa3bd1a90e4b1f2105  
ubuntu@ip-172-31-82-119:~$ mkdir -p $HOME/.kube  
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config  
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

Step 7: Now Run the command kubectl get nodes to see the nodes before executing Join command on nodes.

```
ubuntu@ip-172-31-82-119:~$ kubectl get nodes  
NAME           STATUS   ROLES      AGE   VERSION  
ip-172-31-82-119   NotReady   control-plane   82s   v1.31.1
```

Step 8: Now Run the following command on Node 1 and Node 2 to Join to master.

```
sudo kubeadm join 172.31.82.119:6443 --token h4iisg.g9tdfmc88m9toefp \  
    --discovery-token-ca-cert-hash  
sha256:5b3c7cfdf8115a26f1e73b752820d2b27b84ed476b344aaa3bd1a90e4b1f2105  
ubuntu@ip-172-31-84-169:~$ sudo kubeadm join 172.31.82.119:6443 --token h4iisg.g9tdfmc88m9toefp \  
    --discovery-token-ca-cert-hash sha256:5b3c7cfdf8115a26f1e73b752820d2b27b84ed476b344aaa3bd1a90e4b1f2105  
[preflight] Running pre-flight checks  
[preflight] Reading configuration from the cluster...  
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'  
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"  
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"  
[kubelet-start] Starting the kubelet  
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s  
[kubelet-check] The kubelet is healthy after 501.315929ms  
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap  
  
This node has joined the cluster:  
* Certificate signing request was sent to apiserver and a response was received.  
* The Kubelet was informed of the new secure connection details.  
  
Run 'kubectl get nodes' on the control-plane to see this node join the cluster.
```

Step 9: Now Run the command kubectl get nodes to see the nodes after executing Join command on nodes.

| NAME | STATUS | ROLES | AGE | VERSION |
|------------------|----------|---------------|-------|---------|
| ip-172-31-82-119 | NotReady | control-plane | 2m56s | v1.31.1 |
| ip-172-31-84-169 | NotReady | <none> | 43s | v1.31.1 |
| ip-172-31-87-189 | NotReady | <none> | 39s | v1.31.1 |

Step 10: Since Status is NotReady we have to add a network plugin. And also we have to give the name to the nodes.

```
kubectl apply -f https://docs.projectcalico.org/manifests/calico.yaml
```

```
ubuntu@ip-172-31-82-119:~$ kubectl apply -f https://docs.projectcalico.org/manifests/calico.yaml
poddisruptionbudget.policy/calico-kube-controllers created
serviceaccount/calico-kube-controllers created
serviceaccount/calico-node created
configmap/calico-config created
customresourcedefinition.apiextensions.k8s.io/bgpconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/bgppeers.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/blockaffinities.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/caliconodestatuses.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/clusterinformations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/felixconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworksets.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/hostendpoints.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamblocks.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamconfigs.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamhandles.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ippools.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipservations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/kubecontrollersconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networksets.crd.projectcalico.org created
clusterrole.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrole.rbac.authorization.k8s.io/calico-node created
```

sudo systemctl status kubelet

```
ubuntu@ip-172-31-82-119:~$ sudo systemctl status kubelet
● kubelet.service - kubelet: The Kubernetes Node Agent
   Loaded: loaded (/usr/lib/systemd/system/kubelet.service; enabled; preset: enabled)
   Drop-In: /usr/lib/systemd/system/kubelet.service.d
             └─10-kubeadm.conf
     Active: active (running) since Thu 2024-10-03 12:37:52 UTC; 4min 4s ago
       Docs: https://kubernetes.io/docs/
     Main PID: 10849 (kubelet)
        Tasks: 10 (Limit: 4676)
      Memory: 32.3M (peak: 33.0M)
        CPU: 5.030s
       CGroup: /system.slice/kubelet.service
                 └─10849 /usr/bin/kubelet --bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kubelet.conf --kubeconfig=/etc/kubeconfig

Oct 03 12:41:52 ip-172-31-82-119 kubelet[10849]: E1003 12:41:52.678397 10849 kuberuntime_container.go:851] "Kill container"
Oct 03 12:41:52 ip-172-31-82-119 kubelet[10849]:               rpc error: code = Unknown desc = failed to kill container "233"
Oct 03 12:41:52 ip-172-31-82-119 kubelet[10849]:               : unknown
Oct 03 12:41:52 ip-172-31-82-119 kubelet[10849]: > pod="kube-system/etcd-ip-172-31-82-119" podUID="201366195a044a56173"
Oct 03 12:41:52 ip-172-31-82-119 kubelet[10849]: E1003 12:41:52.688038 10849 log.go:32] "StopPodSandbox from runtime"
Oct 03 12:41:52 ip-172-31-82-119 kubelet[10849]:               rpc error: code = Unknown desc = failed to stop container "233"
Oct 03 12:41:52 ip-172-31-82-119 kubelet[10849]:               : unknown
```

Now Run command kubectl get nodes -o wide we can see Status is ready.

| NAME | STATUS | ROLES | AGE | VERSION | INTERNAL-IP | EXTERNAL-IP | OS-IMAGE | KERNEL- |
|------------------|---------------------|-------------------|-------|---------|---------------|-------------|--------------------|---------|
| VERSION | COLUMN | CONTAINER-RUNTIME | | | | | | |
| ip-172-31-82-119 | Ready | control-plane | 4m32s | v1.31.1 | 172.31.82.119 | <none> | Ubuntu 24.04.1 LTS | 6.8.0-1 |
| 016-aws | containerd://1.7.12 | | | | | | | |
| ip-172-31-84-169 | Ready | <none> | 2m19s | v1.31.1 | 172.31.84.169 | <none> | Ubuntu 24.04.1 LTS | 6.8.0-1 |
| 016-aws | containerd://1.7.12 | | | | | | | |
| ip-172-31-87-189 | Ready | <none> | 2m15s | v1.31.1 | 172.31.87.189 | <none> | Ubuntu 24.04.1 LTS | 6.8.0-1 |
| 016-aws | containerd://1.7.12 | | | | | | | |

Now to Rename run this command kubectl label node ip-172-31-18-135

kubernetes.io/role=worker Rename to Node 1:kubectl label node ip-172-31-28-117

kubernetes.io/role=Node1 Rename to Node 2:kubectl label node ip-172-31-18-135

kubernetes.io/role=Node2

Step 11: Run command kubectl get nodes -o wide . And Hence we can see we have Successfully connected Node 1 and Node 2 to the Master.

| NAME | STATUS | ROLES | AGE | VERSION |
|------------------|--------|---------------|-------|---------|
| ip-172-31-82-119 | Ready | control-plane | 6m52s | v1.31.1 |
| ip-172-31-84-169 | Ready | Node1 | 4m39s | v1.31.1 |
| ip-172-31-87-189 | Ready | Node2 | 4m35s | v1.31.1 |

Conclusion:

In this experiment, we successfully set up a Kubernetes cluster on AWS EC2 instances. We began by provisioning three EC2 instances—Master, Node1, and Node2—with appropriate security rules to facilitate communication within the cluster. After securely SSHing into each instance, we installed Docker as the container runtime on all machines. We proceeded by installing Kubernetes on each instance and initializing the cluster from the master node. Following cluster initialization, the worker nodes were connected to the master using the kubeadm join command, effectively forming a cohesive Kubernetes cluster. Additionally, we implemented a network plugin to ensure proper communication between nodes, allowing the nodes' status to transition to "Ready."

Experiment 4

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

Step 1: Log in to your AWS Academy/personal account and launch a new Ec2 Instance. Select Ubuntu as AMI and t2.medium as Instance Type, create a key of type RSA with .pem extension, and move the downloaded key to the new folder. Note: A minimum of 2 CPUs are required so Please select t2.medium and do not forget to stop the instance after the experiment because it is not available in the free tier.

The screenshot shows the AWS EC2 instance creation process. In the 'Name and tags' section, the instance is named 'exp4'. In the 'Application and OS Images (Amazon Machine Image)' section, the 'Ubuntu' AMI is selected. The 'Instance type' section shows 't2.medium' is chosen, along with other details like memory and pricing.

Name and tags

Name
exp4 [Add additional tags](#)

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

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

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

[Recent](#) [Quick Start](#)

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

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

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

Instance type
t2.medium

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

All generations [Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

Step 2: After creating the instance click on Connect the instance and navigate to SSH Client.

| Instances (7) Info | | Last updated less than a minute ago | | Connect | Instance state ▾ | Actions ▾ | Launch instances | ▼ |
|---|--------|--|----------------|-------------------------|-------------------|--------------|----------------------------------|---|
| <input type="text"/> Find Instance by attribute or tag (case-sensitive) | | | | | All states ▾ | < 1 > | | |
| <input type="checkbox"/> | Name | Instance ID | Instance state | Instance type | Status check | Alarm status | | |
| <input type="checkbox"/> | master | i-03753a87170da35bc | Terminated | t2.medium | - | ... | | |
| <input type="checkbox"/> | node1 | i-0b9d7d0a41c70f7f0 | Terminated | t2.medium | - | ... | | |
| <input type="checkbox"/> | node2 | i-0c4a87fdff3eb59e0 | Terminated | t2.medium | - | ... | | |
| <input type="checkbox"/> | exp4 | i-0ee5e02f7bc43c58e | Running | t2.medium | Initializing | ... | | |
| <input type="checkbox"/> | node1 | i-0e08c62c4f3bc3e7b | Running | t2.medium | 2/2 checks passed | ... | | |
| <input type="checkbox"/> | node2 | i-0a9a7aa3d967ebcd8 | Running | t2.medium | 2/2 checks passed | ... | | |
| <input type="checkbox"/> | master | i-0e47b7838cfcc23f21 | Running | t2.medium | 2/2 checks passed | ... | | |

Connect to instance [Info](#)

Connect to your instance i-0ee5e02f7bc43c58e (exp4) using any of these options

[EC2 Instance Connect](#) | [Session Manager](#) | [SSH client](#) [EC2 serial console](#)

Instance ID
 [i-0ee5e02f7bc43c58e \(exp4\)](#)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is Rakshit.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
 chmod 400 "Rakshit.pem"
4. Connect to your instance using its Public DNS:
 ec2-44-204-48-106.compute-1.amazonaws.com

Example:
 ssh -i "Rakshit.pem" ubuntu@ec2-44-204-48-106.compute-1.amazonaws.com

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

Step 3: Now open the folder in the terminal where our .pem key is stored and paste the Example command (starting with ssh -i) in the terminal.(ssh -i "Master_Ec2_Key.pem"

```
ubuntu@ec2-54-196-129-215.compute-1.amazonaws.com)
```

```
System information as of Thu Oct 3 13:05:18 UTC 2024
```

```
System load: 0.34          Processes:           118
Usage of /: 22.8% of 6.71GB  Users logged in:      0
Memory usage: 5%            IPv4 address for enX0: 172.31.86.43
Swap usage: 0%
```

```
Expanded Security Maintenance for Applications is not enabled.
```

```
0 updates can be applied immediately.
```

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

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

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

```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

```
ubuntu@ip-172-31-86-43:~$ |
```

Step 4: Run the below commands to install and setup Docker.

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add - curl -fsSL
https://download.docker.com/linux/ubuntu/gpg | sudo tee /etc/apt/trusted.gpg.d/docker.gpg >
/dev/null sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable"
```

```
ubuntu@ip-172-31-86-43:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee
/etc/apt/trusted.gpg.d/docker.gpg > /dev/null
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable"
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBFit2ioBEADhWpZ8/wvZ6hUTiX0wQHXMAlaFHcPH9hAtr4F1y2+OYdbtMuth
lqqwp028AqyY+PRfVMtSYMbjuQuu5byyKR01BbqYhuS3jtqQmljZ/bJvXqnmiVXh
38UuLa+z077PxxyQhu5BbqntTPQMfiyqEiU+BKbq2WmANUKQf+1AmZY/IruOXbnq
L4C1+gJ8vfmXQt99npCaxEjaNRVYf0S80cixNzHUYnb6emjlANyEVLZzeqo7XKl7
UrwV5inawTSzWNvtjEjj4nJL8NsLwsSCPQUhTQ+7BbQXAwAmeHCUTQIVvvWXqw0N
cmhh4HgeQscQHYg0JjjDVfoY5MucvgIbIgCqfqzAHW9jxmRL4qbMZj+b1XoePEht
ku4bIQN1X5P07fNWzlgarL5Z4POXDDZTLiQ/El58j9kp4bnWRcjW0lya+f8ocodo
vZZ+Doi+fy4D5ZGrL4XEcIQP/Lv5uFyf+kQt1/94VFYVJoleAv8W92KdgDkhTcTD
G7c0tIkVEKNUq4Bb3aQ64NOZQW7FVjfokwEZd0qPE72Pa45jrZzvUFxSpdiNk2tZ
```

```
sudo apt-get update
```

```
sudo apt-get install -y docker-ce
```

```
ubuntu@ip-172-31-86-43:~$ sudo apt-get update
sudo apt-get install -y docker-ce
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://download.docker.com/linux/ubuntu noble InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

```
sudo mkdir -p /etc/docker
cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"]
}
EOF
```

```
no virtual guests are running on this hypervisor
ubuntu@ip-172-31-86-43:~$ sudo mkdir -p /etc/docker
cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"]
}
EOF
{
  "exec-opts": ["native.cgroupdriver=systemd"]
}
```

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

```
ubuntu@ip-172-31-86-43:~$ sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker
Synchronizing state of docker.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable docker
```

Step 5: Run the below command to install Kubernets. curl -fsSL

```
https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o
/etc/apt/keyrings/kubernetes-apt-keyring.gpg echo 'deb
[signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
```

```
https://pkgs.k8s.io/core:/stable:/v1.31/deb/ | sudo tee /etc/apt/sources.list.d/kubernetes.list
```

```
ubuntu@ip-172-31-86-43:~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list
gpg: missing argument for option "-o"
-bash: /etc/apt/keyrings/kubernetes-apt-keyring.gpg: No such file or directory
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /
```

```
sudo apt-get update
```

```
sudo apt-get install -y kubelet kubeadm kubectl
```

```
sudo apt-mark hold kubelet kubeadm kubectl
```

```
ubuntu@ip-172-31-86-43:~$ sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://download.docker.com/linux/ubuntu noble InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb InRelease [1186 B]
Err:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb InRelease
  The following signatures couldn't be verified because the public key is not available: NO_PUBKEY 234654DA9A296436
Reading package lists... Done
```

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

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

```
ubuntu@ip-172-31-86-43:~$ sudo systemctl enable --now kubelet
sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
W1003 13:11:39.162569    5084 checks.go:1080] [preflight] WARNING: Couldn't create the interface used for talking to the
  container runtime: failed to create new CRI runtime service: validate service connection: validate CRI v1 runtime API f
or endpoint "unix:///var/run/containerd/containerd.sock": rpc error: code = Unimplemented desc = unknown service runtime
.v1.RuntimeService
  [WARNING FileExisting-socat]: socat not found in system path
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
```

Now We have got an error. So we have to perform some additional commands as follow.

```
sudo apt-get install -y containerd
```

```
ubuntu@ip-172-31-86-43:~$ sudo apt-get install -y containerd
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz
  slirp4netns
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  runc
The following packages will be REMOVED:
  containerd.io docker-ce
The following NEW packages will be installed:
  containerd runc
0 upgraded, 2 newly installed, 2 to remove and 6 not upgraded.
Need to get 47.2 MB of archives.
After this operation, 53.1 MB disk space will be freed.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.1.12-0ubuntu3.1 [8599 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 containerd amd64 1.7.12-0ubuntu4.1 [38.6 M]
```

```
sudo mkdir -p /etc/containerd
```

```
sudo containerd config default | sudo tee /etc/containerd/config.toml
```

```

no VM guests are running (addacted hypervisor (qemu) binaries on this host).
ubuntu@ip-172-31-86-43:~$ sudo mkdir -p /etc/containerd
sudo containerd config default | sudo tee /etc/containerd/config.toml
disabled_plugins = []
imports = []
oom_score = 0
plugin_dir = ""
required_plugins = []
root = "/var/lib/containerd"
state = "/run/containerd"
temp = ""
version = 2

[cgroup]
path = ""

[debug]
address = ""
format = ""

```

sudo systemctl restart containerd
 sudo systemctl enable containerd
 sudo systemctl status containerd

```

ubuntu@ip-172-31-86-43:~$ sudo systemctl restart containerd
sudo systemctl enable containerd
sudo systemctl status containerd
● containerd.service - containerd container runtime
   Loaded: loaded (/usr/lib/systemd/system/containerd.service; enabled; preset: enabled)
   Active: active (running) since Thu 2024-10-03 13:12:51 UTC; 276ms ago
     Docs: https://containerd.io
     Main PID: 5462 (containerd)
       Tasks: 8
      Memory: 13.5M (peak: 14.1M)
        CPU: 54ms
       CGroup: /system.slice/containerd.service
               └─5462 /usr/bin/containerd

Oct 03 13:12:51 ip-172-31-86-43 containerd[5462]: time="2024-10-03T13:12:51.543254591Z" level=info msg="serving...
Oct 03 13:12:51 ip-172-31-86-43 containerd[5462]: time="2024-10-03T13:12:51.543291344Z" level=info msg="serving...
Oct 03 13:12:51 ip-172-31-86-43 containerd[5462]: time="2024-10-03T13:12:51.543326951Z" level=info msg="Start subscribi...
Oct 03 13:12:51 ip-172-31-86-43 containerd[5462]: time="2024-10-03T13:12:51.543358179Z" level=info msg="Start recoverin...
Oct 03 13:12:51 ip-172-31-86-43 containerd[5462]: time="2024-10-03T13:12:51.543399144Z" level=info msg="Start event mon...
Oct 03 13:12:51 ip-172-31-86-43 containerd[5462]: time="2024-10-03T13:12:51.543407458Z" level=info msg="Start snapshots...
Oct 03 13:12:51 ip-172-31-86-43 containerd[5462]: time="2024-10-03T13:12:51.543414974Z" level=info msg="Start cn...
Oct 03 13:12:51 ip-172-31-86-43 containerd[5462]: time="2024-10-03T13:12:51.543422043Z" level=info msg="Start streaming...
Oct 03 13:12:51 ip-172-31-86-43 containerd[5462]: time="2024-10-03T13:12:51.543466839Z" level=info msg="containerd succ...
Oct 03 13:12:51 ip-172-31-86-43 systemd[1]: Started containerd.service - containerd container runtime.


```

sudo apt-get install -y socat

```

ubuntu@ip-172-31-86-43:~$ sudo apt-get install -y socat
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz
  slirp4netns
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  socat
0 upgraded, 1 newly installed, 0 to remove and 6 not upgraded.
Need to get 374 kB of archives.
After this operation, 1649 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/amd64 socat amd64 1.8.0.0-4build3 [374 kB]
Fetched 374 kB in 0s (13.9 MB/s)
Selecting previously unselected package socat.

```

Step 6: Initialize the Kubecluster sudo kubeadm init --pod-network-cidr=10.244.0.0/16

```
ubuntu@ip-172-31-86-43:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
W1003 13:13:28.336900      5640 checks.go:846] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent with that used by kubeadm. It is recommended to use "registry.k8s.io/pause:3.10" as the CRI sandbox image.
```

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

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

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

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

```
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 172.31.86.43:6443 --token oy5zuy.78xd696vk5gr8ip8 \
    --discovery-token-ca-cert-hash sha256:cb705f4200aa1d0a890b60cedb8802c8512842a85be9e9fd527799b09995c9ca
ubuntu@ip-172-31-86-43:~$ mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

Add a common networking plugin called flannel as mentioned in the code.

```
kubectl apply -f
```

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

```
ubuntu@ip-172-31-86-43:~$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
```

Step 7: Now that the cluster is up and running, we can deploy our nginx server on this cluster. Apply this deployment file using this command to create a deployment

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

```
ubuntu@ip-172-31-86-43:~$ kubectl apply -f https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
```

```
kubectl get pods
```

| NAME | READY | STATUS | RESTARTS | AGE |
|----------------------------------|-------|---------|----------|-----|
| nginx-deployment-d556bf558-52k84 | 0/1 | Pending | 0 | 17s |
| nginx-deployment-d556bf558-d5fv5 | 0/1 | Pending | 0 | 17s |

```
POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath=".items[0].metadata.name")  
kubectl port-forward $POD_NAME 8080:80
```

```
ubuntu@ip-172-31-86-43:~$ POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath=".items[0].metadata.name")  
kubectl port-forward $POD_NAME 8080:80  
error: unable to forward port because pod is not running. Current status=Pending
```

Note : We have faced an error as pod status is pending so make it running run below commands then again run above 2 commands.

```
kubectl taint nodes --all node-role.kubernetes.io/control-plane-node/ip-172-31-20-171 untainted  
kubectl get nodes
```

```
ubuntu@ip-172-31-86-43:~$ kubectl taint nodes ip-172-31-86-43 node-role.kubernetes.io/control-plane:NoSchedule-  
node/ip-172-31-86-43 untainted  
ubuntu@ip-172-31-86-43:~$ kubectl get nodes  
NAME           STATUS    ROLES     AGE      VERSION  
ip-172-31-86-43   Ready    control-plane   4m30s   v1.31.1
```

kubectl get pods

```
POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath=".items[0].metadata.name")  
kubectl port-forward $POD_NAME 8080:80
```

```
ubuntu@ip-172-31-86-43:~$ kubectl get pods  
NAME          READY   STATUS    RESTARTS   AGE  
nginx-deployment-d556bf558-52k84   1/1     Running   0          3m51s  
nginx-deployment-d556bf558-d5fv5   1/1     Running   0          3m51s  
ubuntu@ip-172-31-86-43:~$ POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath=".items[0].metadata.name")  
kubectl port-forward $POD_NAME 8080:80  
Forwarding from 127.0.0.1:8080 -> 80  
Forwarding from [::1]:8080 -> 80
```

Step 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.

```
Expanded Security Maintenance for Applications is not enabled.

6 updates can be applied immediately.
5 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

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

Last login: Thu Oct  3 13:05:19 2024 from 110.226.182.217
ubuntu@ip-172-31-86-43:~$ curl --head http://127.0.0.1:8080
HTTP/1.1 200 OK
Server: nginx/1.14.2
Date: Thu, 03 Oct 2024 13:19:50 GMT
Content-Type: text/html
Content-Length: 612
Last-Modified: Tue, 04 Dec 2018 14:44:49 GMT
Connection: keep-alive
ETag: "5c0692e1-264"
Accept-Ranges: bytes

ubuntu@ip-172-31-86-43:~$ |
```

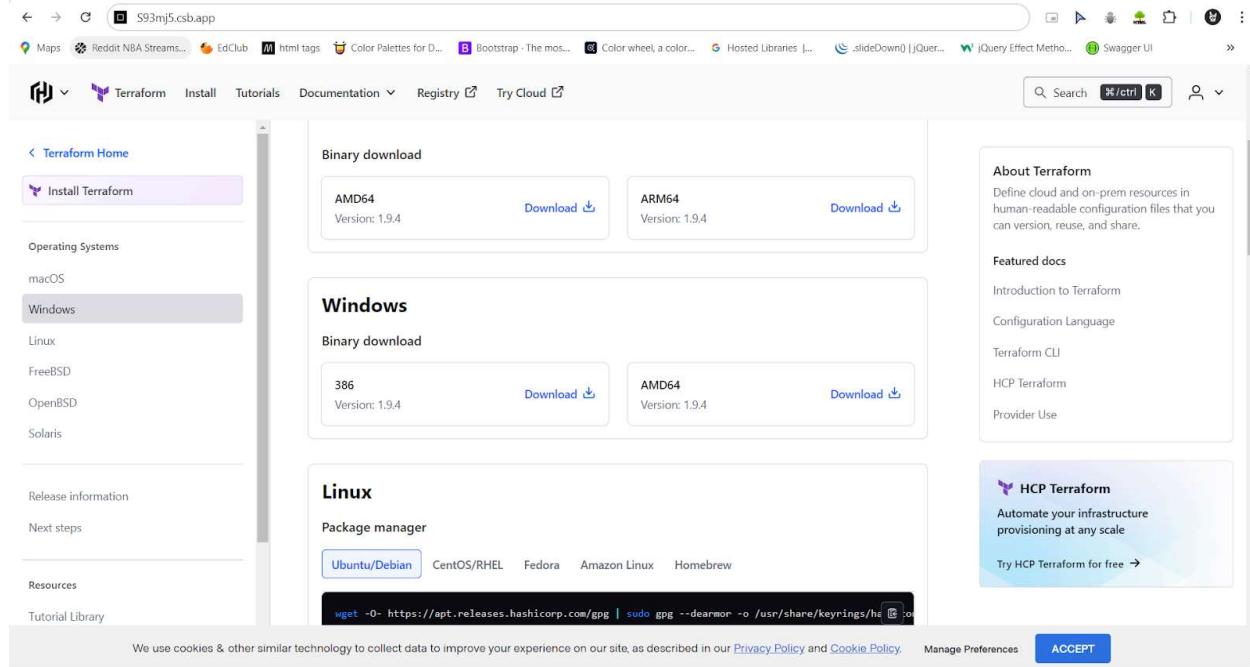
Conclusion:

In this experiment, we successfully set up a Kubernetes cluster on an AWS EC2 instance using kubectl and deployed a sample nginx application. By installing Docker and Kubernetes, configuring containerd, and initializing the cluster, we ensured smooth cluster operations. After deploying the nginx server, we verified its successful deployment by using kubectl port-forward and confirming access via curl. This process provided practical experience in managing Kubernetes clusters and deploying applications on cloud infrastructure.

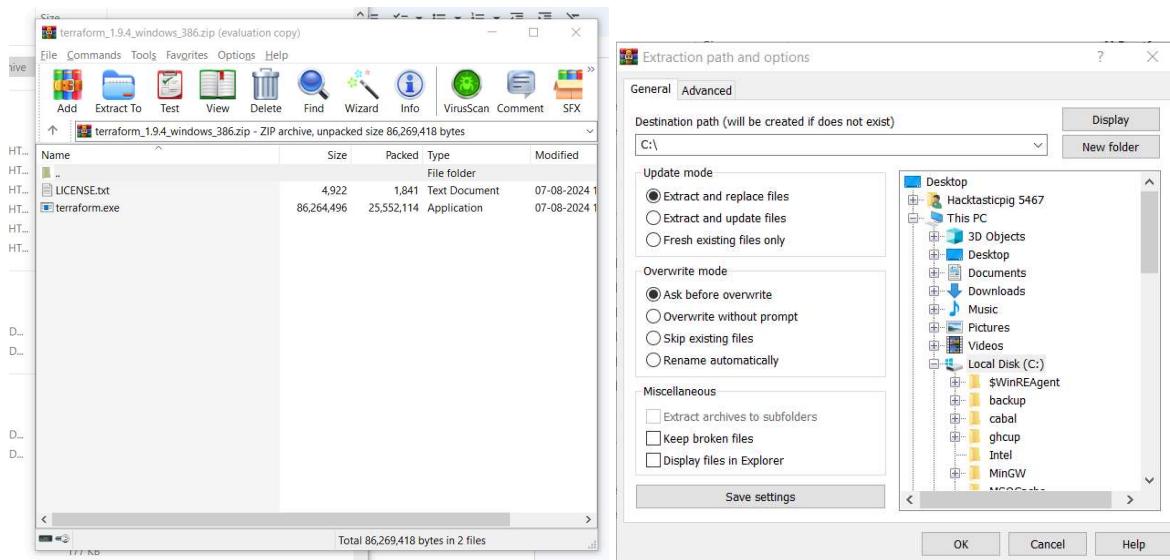
Experiment 5

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

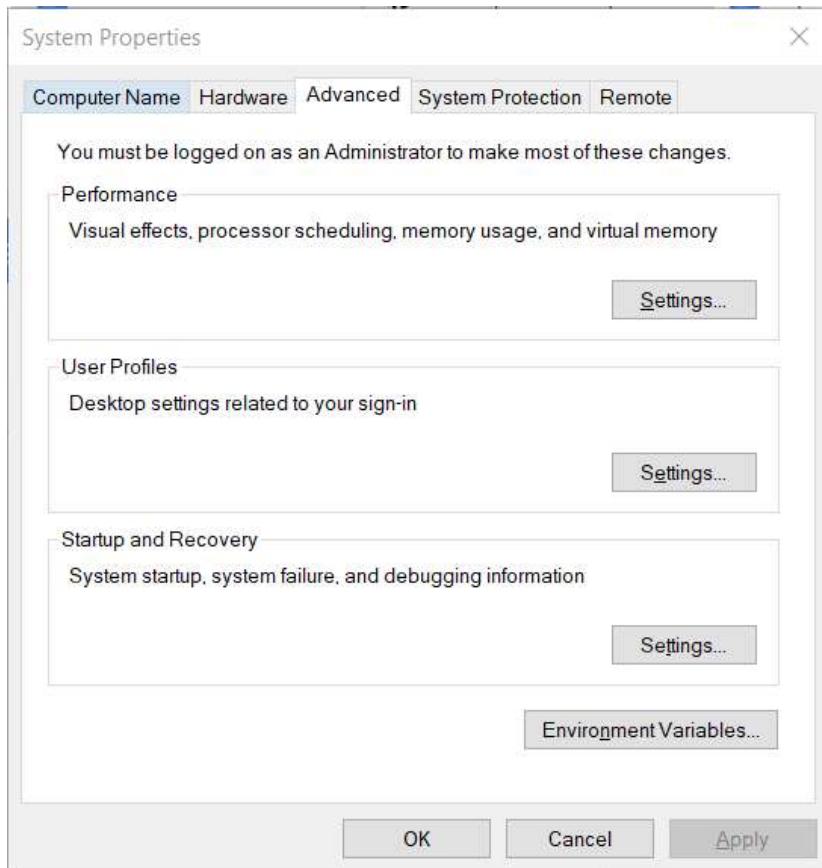
1. To install terraform, go to "<https://developer.hashicorp.com/terraform/install>". Visit the site and search for windows system download. You can either select 386 or AMD64 bit version.



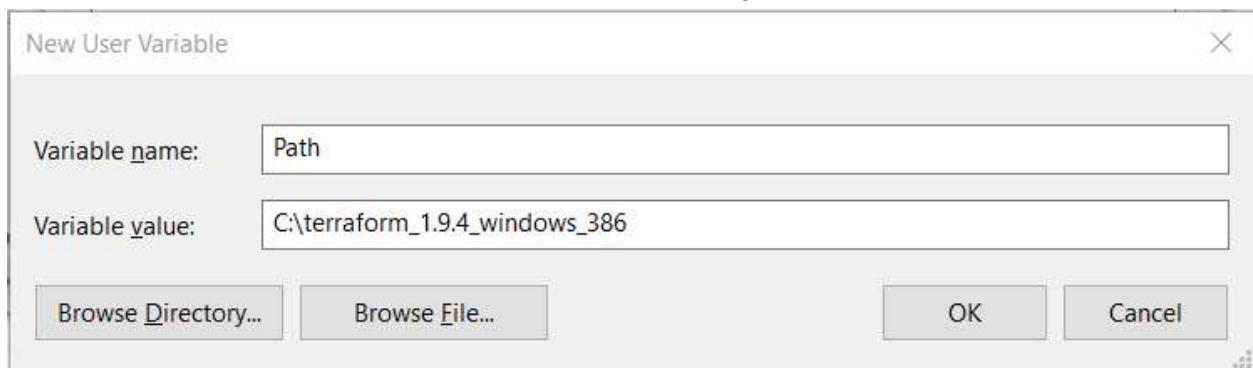
2. Extract the downloaded zip file in a suitable file in your pc.



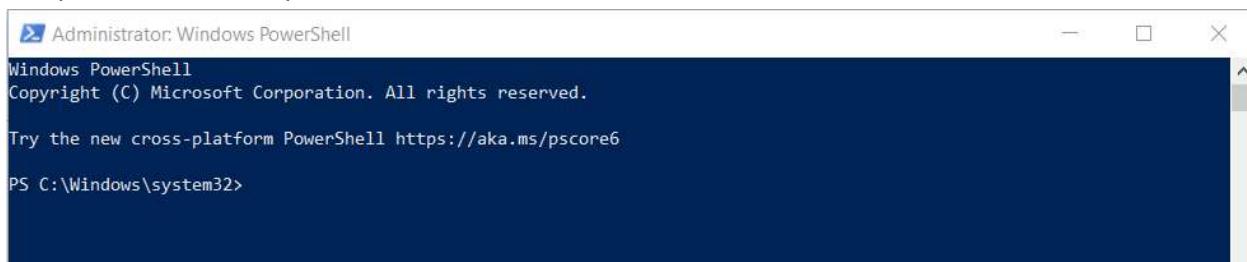
3. Copy the path where the file has been extracted. Now we need to setup the environment variable. Click on Environment variable in system properties.



4. Create a new variable with name as Path and value as the path of the file where the terraform.exe file has been extracted, and save the changes.



5. Open the windows power shell as an administrator.

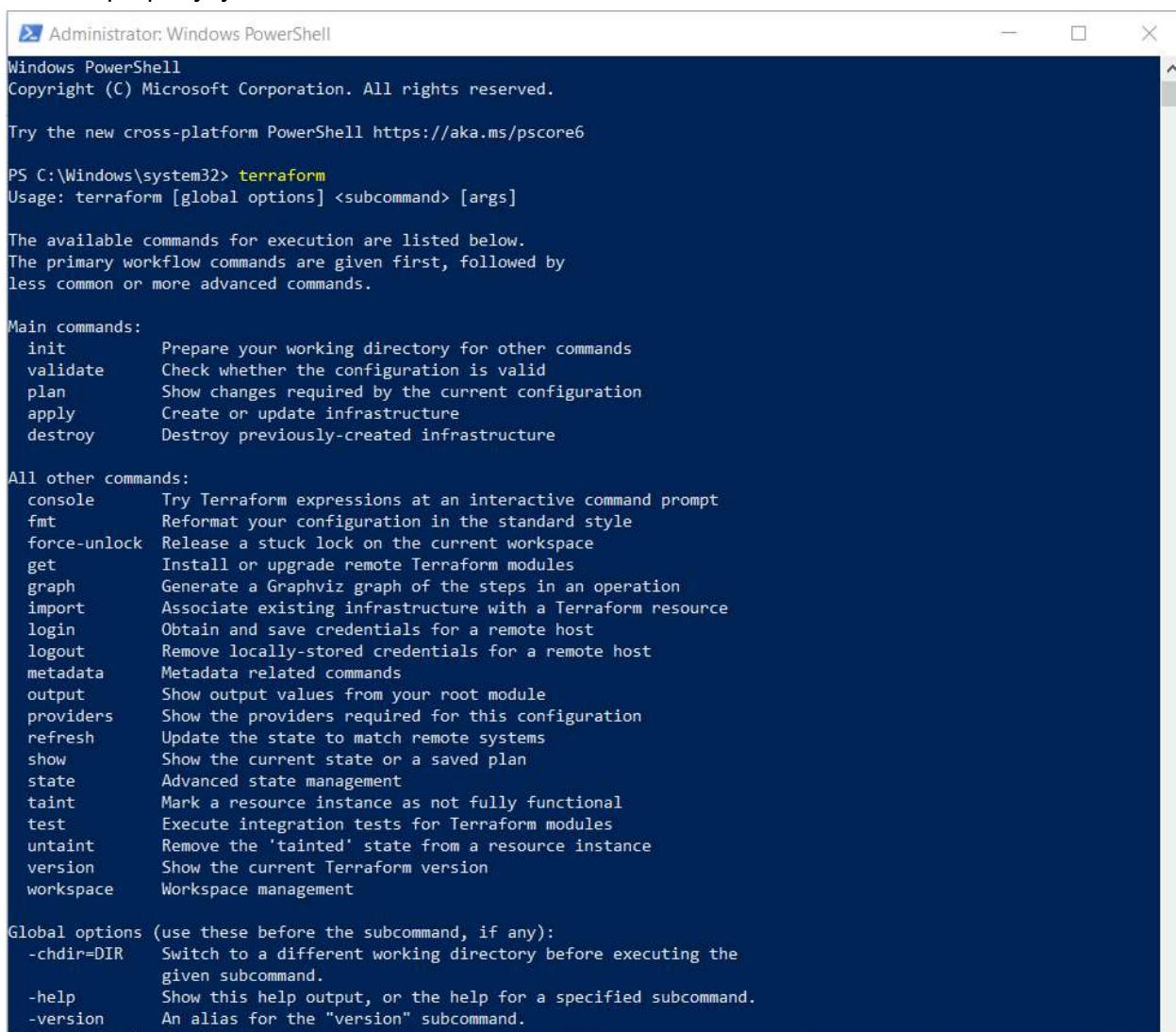


```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

PS C:\Windows\system32>
```

6. Type “terraform” in powershell. If the installation was successful and you have setup the path variable properly, you will be able to see list of available commands in terraform.



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

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

PS C:\Windows\system32> 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
  test       Execute integration tests for Terraform modules
  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.
```

Experiment 6

Aim: To Build, change, and destroy AWS / GCP /Microsoft Azure/ DigitalOcean infrastructure using Terraform. (S3 bucket or Docker)

1. Check docker installation by running the command “docker” and “docker --version”.

```
C:\Windows\system32>docker
Usage: docker [OPTIONS] COMMAND
      A self-sufficient runtime for containers

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

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

Swarm Commands:
  swarm   Manage Swarm
```

```
C:\Windows\system32>docker --version
Docker version 27.0.3, build 7d4bcd8
```

```
C:\Windows\system32>
```

2. Create a folder in your system named “terraformScripts”(Do not use terraform as name as it may recognize it as a keyword). Inside it create a subfolder docker. In this create a file docker.tf and type the following code in it.

```
docker.tf
1  terraform {
2    required_providers {
3      docker = {
4        source  = "kreuzwerker/docker"
5        version = "2.21.0"
6      }
7    }
8  }
9
10 provider "docker" {
11   host = "npipe://./pipe/docker_engine"
12 }
13
14 # Pull the image
15 resource "docker_image" "ubuntu" {
16   name = "ubuntu:latest"
17 }
18
19 # Create a container
20 resource "docker_container" "foo" {
21   image = docker_image.ubuntu.image_id
22   name  = "foo"
23   command = ["sleep", "3600"]
24 }
```

3. Run the windows powershell as administrator. Navigate to the docker folder created in the above step. Run the terraform init command.

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

Terraform has been successfully initialized!

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

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

4. Terraform plan command is used to create a execution plan and see the resources.

```
PS C:\Users\Lenovo\Desktop\TerraformScripts\Docker> terraform plan

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

Terraform will perform the following actions:

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

5. Docker image is command used to see all the images currently created in docker desktop

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

6. Terraform apply command is used to execute the steps listed in terraform plan.

```
PS C:\Users\Lenovo\Desktop\TerraformScripts\Docker> terraform apply
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubun
tu:latest]
docker_container.foo: Refreshing state... [id=4de644086273a692820f646431610cae2f095755228f61eafc2bb712b2766040]

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)
}
```

7. Terraform destroy is used to destroy all resources that are currently being managed by terraform configuration.

```
PS C:\Users\Lenovo\Desktop\TerraformScripts\Docker> terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubun
tu:latest]
docker_container.foo: Refreshing state... [id=6d649285ef1d861ad451273401bc2771fe4e0be9b78f232aa64230ca0f58d36e]

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         = "6d649285ef1d" -> null
  id               = "6d649285ef1d861ad451273401bc2771fe4e0be9b78f232aa64230ca0f58d36e" -> 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      = [
    {
  ]
```

```
# docker_image.ubuntu will be destroyed
resource "docker_image" "ubuntu" {
  id      = "sha256:edbf74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest" -> null
  image_id = "sha256:edbf74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> null
  latest   = "sha256:edbf74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> 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=6d649285ef1d861ad451273401bc2771fe4e0be9b78f232aa64230ca0f58d36e]
docker_container.foo: Destruction complete after 1s
docker_image.ubuntu: Destroying... [id=sha256:edbf74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_image.ubuntu: Destruction complete after 0s
```

Destroy complete! Resources: 2 destroyed.

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

8. Terraform validate is a command used to check for syntax and other errors in terraform configuration files.

```
PS C:\Users\Lenovo\Desktop\TerraformScripts\Docker> terraform validate
Success! The configuration is valid.
```

9. The terraform show command in Terraform is used to display information about the state of resources managed by Terraform.

```
PS C:\Users\Lenovo\Desktop\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        = "985d156486d1"
    id              = "985d156486d181dc0753e4e9eff337899b74717caf7119264c3ca8ddcb300fe9"
    image           = "sha256:edbf74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a"
    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
}
```

10. The terraform state list command is used to list all the resources currently managed by your Terraform state

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

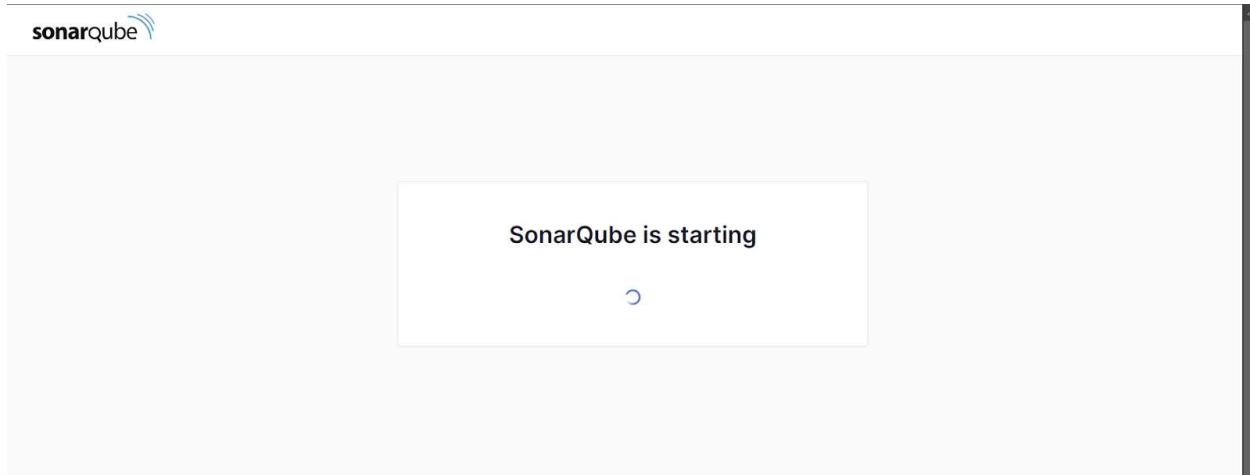
Experiment 7

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

1. Open docker desktop on your device. Then run the following command in powershell.
docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest

```
PS C:\Windows\system32> 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
5008a1b64daaff9ca9430d6426a4ceb303d964d039798555e9fb548ff7c89073
```

2. Once the command is run, you can verify running of sonarqube by checking the url
<http://localhost:9000>



3. Login with username admin and password admin
4. Create a local project in sonarqube. Give the project a name, setup the project and click on create project.

1 of 2

Create a local project

Project display name *

sonarqube-test



Project key *

sonarqube-test



Main branch name *

main

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

Cancel

Next

5. Go to jenkins, i.e <http://localhost:8080>. Go to Manage Jenkins -> Plugins -> Available Plugins. Search for SonarQube Scanner and install it.

The screenshot shows the Jenkins plugin management interface. The 'Install' tab is selected. A single plugin, 'SonarQube Scanner 2.17.2', is listed. It has a checked checkbox next to its name. Below the name are two tabs: 'External Site/Tool Integrations' and 'Build Reports'. To the right of the plugin details is a timestamp: '7 mo 11 days ago'. At the bottom of the plugin card, there is a brief description: 'This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.'

6. Search for SonarQube Scanner under Global Tool Configuration. Choose the latest configuration and choose Install automatically.
7. Create a new project. Select the type as freestyle project.

New Item

Enter an item name

advdevops7

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.



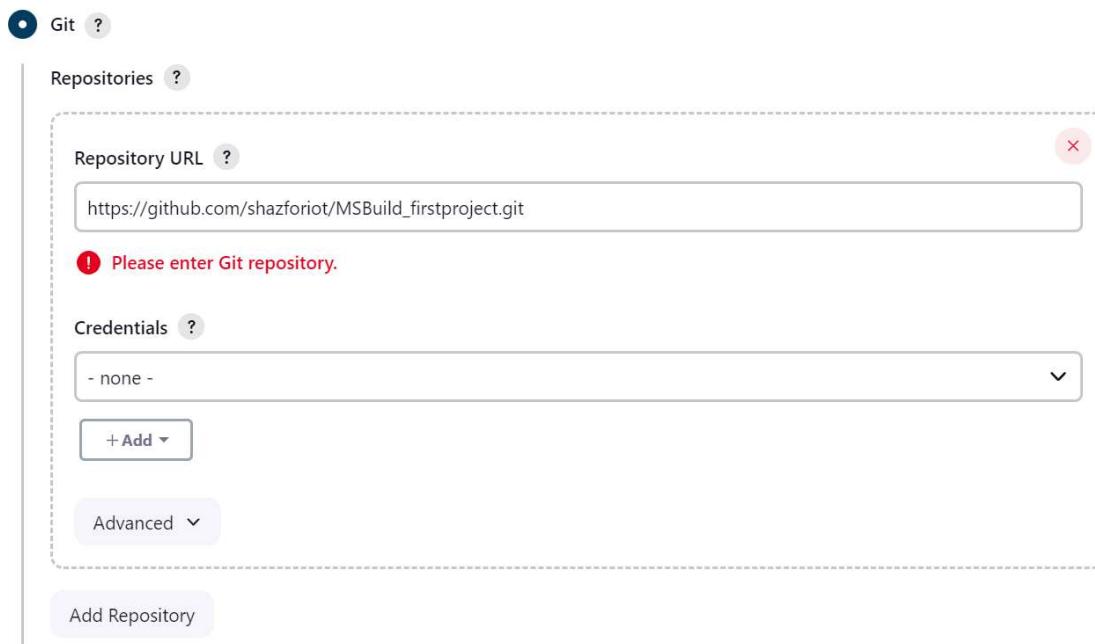
Maven project

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

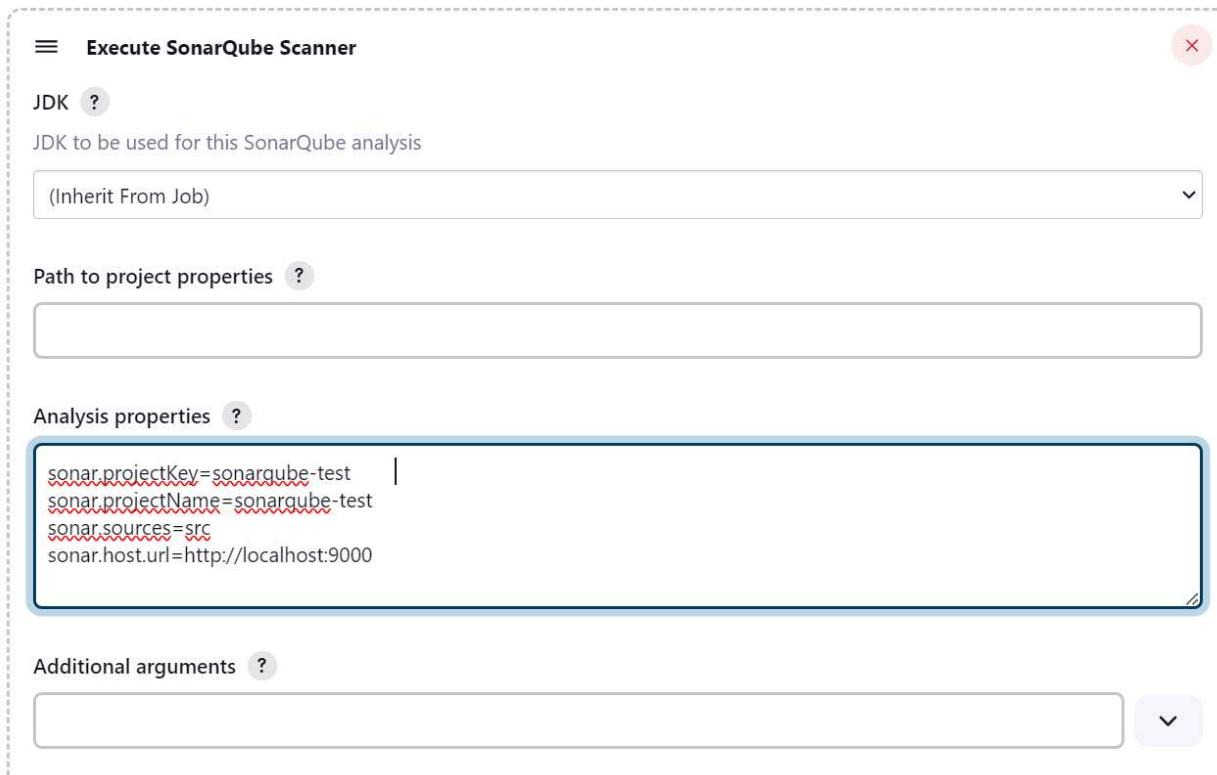
8. Select git in source code management. Enter the following as the url for repository

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

It is a simple hello world project with no vulnerabilities and issues.



9. In build steps select Execute SonarQube Scanner. In analysis properties give the details about the sonarqube project.



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



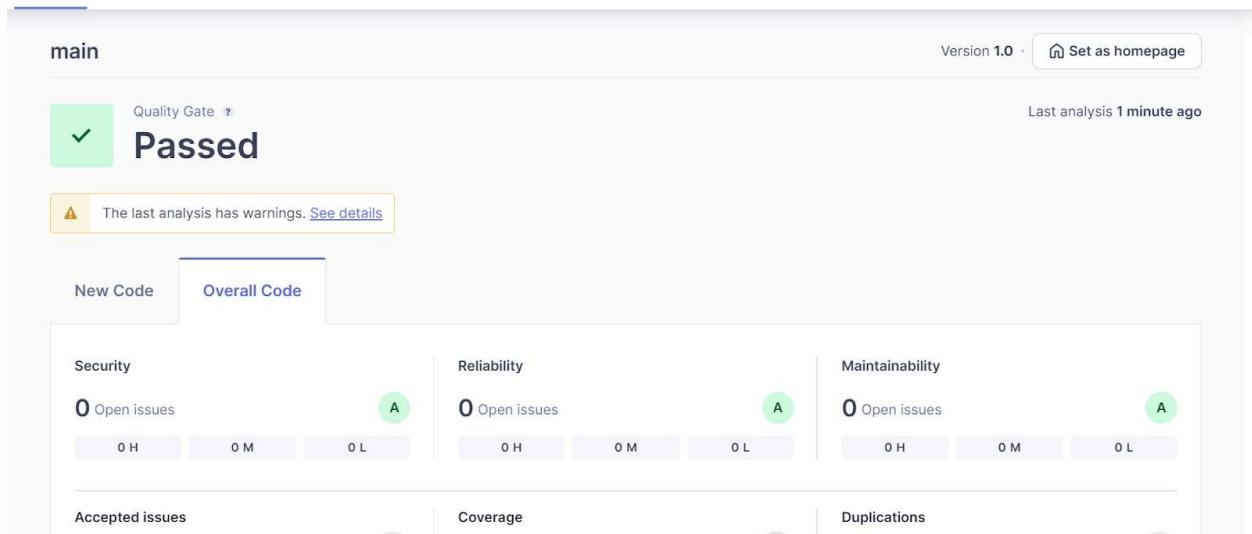
11. Once everything is set up, Click on Build now to build the freestyle project.

Console Output

Started by user [Rakshit Kumar Sharma](#)
 Running as SYSTEM
 Building on the built-in node in workspace C:\ProgramData\Jenkins\.jenkins\workspace\advdevops7
 The recommended git tool is: NONE
 No credentials specified
 > git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\advdevops7\.git # timeout=10
 Fetching changes from the remote Git repository
 > git.exe config remote.origin.url https://github.com/shazforiot/MSBuild_firstproject.git # timeout=10
 Fetching upstream changes from https://github.com/shazforiot/MSBuild_firstproject.git
 > git.exe --version # timeout=10
 > git --version # 'git version 2.46.0.windows.1'
 > git.exe fetch --tags --force --progress -- https://github.com/shazforiot/MSBuild_firstproject.git
 +refs/heads/*:refs/remotes/origin/* # timeout=10

Check the console output to see the successful build of the project.

12. Once Build is complete, check the sonarqube project. You will see a passed message, this indicated the code did not have any issues while building.



Conclusion:

In this experiment, we successfully set up a static application security testing (SAST) pipeline by integrating SonarQube with Jenkins. We initiated a SonarQube server using Docker, allowing for local access and code analysis. A new project was created in SonarQube to facilitate static code evaluation. We configured Jenkins by installing the SonarQube Scanner plugin and created a Freestyle project linked to a sample GitHub repository. By executing the SonarQube analysis within the Jenkins build steps, we provide

Experiment 8

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

1. Open up the jenkins dashboard. Usually on localhost: 8080
2. Run SonarQube in a Docker container using this command -

```
PS C:\Windows\system32> 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  
5008a1b64daaff9ca9430d6426a4ceb303d964d039798555e9fb548ff7c89073
```

3. Once the container is running, you can check it on localhost:9000. If you have already made a container named sonarqube, run the following command instead.

```
PS C:\Windows\system32> docker start sonarqube  
>>  
sonarqube
```

4. Login to sonarqube with username admin, password admin, or if you have setup any other credentials use that.
5. Create a local project. Give it a name(sonarqube-test2). Configure the project and click on create.

Create a local project

Project display name *

Project key *

Main branch name *

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

CancelNext

6. Now we need to install the SonarQube CLI.

Go to following

link:<https://docs.sonarsource.com/sonarqube/latest/analyzing-source-code/scanners/sonarscanner/>

Download the suitable version compatible with your device.

Extract the zip file downloaded. Inside the folder go to bin/sonar-scanner, copy the path of this file.

7. Create a new project in Jenkins. Give it a name and select the project type as pipeline in Jenkins.

New Item

Enter an item name

Select an item type



Freestyle project

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



Maven project

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



Pipeline

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



Multi-configuration project

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

8. Inside the pipeline script, give the following script. Use the path copied in earlier step instead of Path to sonarqube folder.

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

Click on save to create the pipeline.

Pipeline

Definition

Pipeline script

Script

```
1 ▼ node {  
2 ▼   stage('Cloning the GitHub Repo') {  
3     |   git 'https://github.com/shazforiot/GOL.git'  
4   }  
5  
6 ▼   stage('SonarQube analysis') {  
7     |   // Performing SonarQube analysis  
8     |   withSonarQubeEnv('sonarqube') {  
9       |       sh '''  
10      |       sonar-scanner \  
11      |       -Dsonar.login=admin \  
12      |       -Dsonar.password=rakshit \  
13      |       -Dsonar.projectKey=sonarqube-test2|  
14      |       -Dsonar.projectVersion=1.0 \  
15      |       -Dsonar.sources=src \  
16      |       -Dsonar.exclusions=vendor/**,resources/**,**/*.java \  
17      |       -Dsonar.host.url=http://127.0.0.1:9000
```

try sample Pipeline...

Use Groovy Sandbox

Pipeline Syntax

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

9. Run the build.

 Status

 Changes

 Build Now

Stage View



10. Check the console output

✓ Console Output

[Download](#)[Copy](#)[View as plain text](#)

Skiping 4,249 KB. [Full Log](#)

```
09:27:10.137 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/gui/ReportMainFrame.WindowHappenings.html for block at line 212. Keep
only the first 100 references.
09:27:10.137 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/gui/ReportMainFrame.WindowHappenings.html for block at line 215. Keep
only the first 100 references.
09:27:10.137 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/gui/ReportMainFrame.WindowHappenings.html for block at line 212. Keep
only the first 100 references.
09:27:10.137 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/gui/ReportMainFrame.WindowHappenings.html for block at line 296. Keep
only the first 100 references.
09:27:10.137 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/gui/ReportMainFrame.WindowHappenings.html for block at line 17. Keep
only the first 100 references.
09:27:10.137 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/gui/ReportMainFrame.WindowHappenings.html for block at line 212. Keep
only the first 100 references.
09:27:10.137 WARN Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/gui/ReportMainFrame.WindowHappenings.html for block at line 215. Keep

09:27:15.430 INFO CPU Executor CPU caulation finished (done) | time=28/056ms
09:27:15.451 INFO SCM revision ID 'ba799ba7e1b576f04a4612322b0412c5e6e1e5e4'
09:30:22.050 INFO Analysis report generated in 138747ms, dir size=127.2 MB
09:33:36.168 INFO Analysis report compressed in 194118ms, zip size=29.6 MB
09:33:36.981 INFO Analysis report uploaded in 813ms
09:33:36.982 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://127.0.0.1:9000/dashboard?id=sonarqube-test2
09:33:36.982 INFO Note that you will be able to access the updated dashboard once the server has processed the
submitted analysis report
09:33:36.982 INFO More about the report processing at http://127.0.0.1:9000/api/ce/task?id=0fd50499-d2c7-460f-a5c0-06a3b9908b8b
09:33:43.711 INFO Analysis total time: 15:38.015 s
09:33:43.729 INFO SonarScanner Engine completed successfully
09:33:44.597 INFO EXECUTION SUCCESS
09:33:44.597 INFO Total time: 15:41.397s
[Pipeline]
[Pipeline] // withSonarQubeEnv
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

11. Since the build is successful, now check the project in sonarqube.

The screenshot shows the SonarQube main dashboard for the project 'sonarqube-test2'. The dashboard is in a 'Passed' state. Key metrics displayed include:

- Security:** 0 Open issues
- Reliability:** 68k Open issues
- Maintainability:** 164k Open issues

The 'Overall Code' tab is active. A warning message at the top states: "The last analysis has warnings. See details".

You will be able to see different issues with the code under different tabs listed.

12. Code problems:

Reliability:

The screenshot shows three code problems related to Reliability:

- Add "lang" and/or "xml:lang" attributes to this "<html>" element.** Priority: Reliability (A). Status: Open. Last updated: 4 years ago. Impact: Major.
- Insert a <!DOCTYPE> declaration to before this <html> tag.** Priority: Reliability (A). Status: Open. Last updated: 4 years ago. Impact: Major.
- Add "<th>" headers to this "<table>".** Priority: Reliability (A). Status: Open. Last updated: 4 years ago. Impact: Major.

Maintainability:

The screenshot shows the SonarQube interface for the project 'gameoflife-acceptance-tests/Dockerfile'. At the top, there are buttons for 'Bulk Change', 'Select issues' (with dropdown arrows), 'Navigate to issue' (with left and right arrows), and statistics: '163,781 issues' and '1705d effort'. Below this, a search bar contains the project name. Two issues are listed in a card format:

- Issue 1:** Use a specific version tag for the image. (Intentionality) - Maintainability (No tags). Status: Open. Created: L1 5min effort 4 years ago. Type: Code Smell. Priority: Major.
- Issue 2:** Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. (Intentionality) - Maintainability (No tags). Status: Open. Created: L12 5min effort 4 years ago. Type: Code Smell. Priority: Major.

Duplications:

The screenshot shows the SonarQube interface for the project 'sonarqube-test2'. At the top, it displays 'View as List' (with a dropdown arrow), 'Select files' (with dropdown arrows), 'Navigate' (with left and right arrows), and '1,147 files'. Below this, a summary shows 'Duplicated Lines (%) 50.6%' and a link to 'See history'. A table lists the top duplicated files:

| | Duplicated Lines (%) | Duplicated Lines |
|---|----------------------|------------------|
| gameoflife-web/tools/jmeter/docs/api/org/apache/jm.../ReportCellRenderer.html | 92.4% | 1,282 |
| gameoflife-web/tools/jmeter/docs/api/org/apache/jo.../RightAlignRenderer.html | 92.4% | 1,198 |
| gameoflife-web/tools/jmeter/docs/api/org/apache/jm.../JMeterCellRenderer.html | 92.1% | 1,281 |

Bug:

The screenshot shows the SonarQube dashboard for the 'gameoflife-core' project. At the top, there are navigation buttons for 'Bulk Change', 'Select issues', 'Navigate to issue', and statistics: '46,515 issues' and '1426d effort'. Below this, the project name 'gameoflife-core/build/reports/tests/all-tests.html' is displayed. Two code smell issues are listed:

- Add "lang" and/or "xml:lang" attributes to this "<html>" element.** (Intentionality: Reliability) - Status: Open, Not assigned. Effort: L1, 2min effort, 4 years ago. Type: Bug, Major.
- Insert a <!DOCTYPE> declaration to before this <html> tag.** (Consistency: Reliability) - Status: Open, Not assigned. Effort: L1, 5min effort, 4 years ago. Type: Bug, Major.

Code smells:

The screenshot shows the SonarQube dashboard for the 'gameoflife-acceptance-tests' project. At the top, there are navigation buttons for 'Bulk Change', 'Select issues', 'Navigate to issue', and statistics: '164,034 issues' and '1708d effort'. Below this, the project name 'gameoflife-acceptance-tests/Dockerfile' is displayed. Two code smell issues are listed:

- Use a specific version tag for the image.** (Intentionality: Maintainability) - Status: Open, Not assigned. Effort: L1, 5min effort, 4 years ago. Type: Code Smell, Major.
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.** (Intentionality: Maintainability) - Status: Open, Not assigned. Effort: L12, 5min effort, 4 years ago. Type: Code Smell, Major.

And various other issues or problems can be seen in the sonarqube project dashboard.

Conclusion:

In this experiment, a Jenkins CI/CD pipeline was successfully integrated with SonarQube to perform static code analysis on a sample application. SonarQube, running in a Docker

container, analyzed the code for issues like bugs, code smells, and security vulnerabilities. The pipeline was automated by cloning the source code from a GitHub repository and using the SonarQube Scanner to detect potential code problems. After a successful Jenkins build, the SonarQube project dashboard provided information about code's reliability, maintainability, duplications, and detected bugs, offering feedback that help improve the code quality.

Experiment 9

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

- Firstly, go to the EC2 section in the Aws academy lab. Go to the security group and click on create a new security group. Add the following inbound rules in your security group and give it an appropriate name.

| Inbound rules Info | | | | | | |
|------------------------------------|---------------------------|-------------------------------|---------------------------------|-----------------------------|---|------------------------|
| Security group rule ID | Type Info | Protocol Info | Port range Info | Source Info | Description - optional Info | |
| sgr-045Saad39f9e4f0dd | Custom TCP | TCP | 5666 | Custom | Q. 0.0.0.0/0 | Delete |
| sgr-09942a64a610c4a51 | All traffic | All | All | Custom | Q. 0.0.0.0/0 | Delete |
| sgr-04fd581f3a2c685b7 | All ICMP - IPv4 | ICMP | All | Custom | Q. 0.0.0.0/0 | Delete |
| sgr-0c58c654d3393939a | SSH | TCP | 22 | Custom | Q. 0.0.0.0/0 | Delete |
| sgr-05c2dc0327e5303c7 | All ICMP - IPv6 | IPv6 ICMP | All | Custom | Q. -/0 | Delete |
| sgr-01f2fb5a613ff506 | HTTP | TCP | 80 | Custom | Q. -/0 | Delete |
| sgr-0ac3b5859ba38d7b1 | HTTPS | TCP | 443 | Custom | Q. 0.0.0.0/0 | Delete |

- Now go to the EC2 instance dashboard and click on Launch Instance. Name your instance as nagios-host, select Amazon Linux as the instance type. Create a new key pair and download the corresponding .pem file in your pc. Also choose the existing security group which we created in the previous step.

Name and tags [Info](#)

| | | |
|------|--|-------------------------------------|
| Name | <input type="text" value="nagios-host"/> | Add additional tags |
|------|--|-------------------------------------|

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

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

| | | |
|-------------------------|-----------------------------|--|
| Recents | Quick Start | |
|-------------------------|-----------------------------|--|

| | | | | | |
|---|--|---|--|--|--|
| Amazon Linux  | macOS  | Ubuntu  | Windows  | Red Hat  | ! ? |
|---|--|---|--|--|--|

Q
[Browse more AMIs](#)

Including AMIs from AWS, Marketplace and the Community

3. Now click on the instance id in the dashboard and click on connect. Go to the ssh client tab and copy the example command. Locate the folder in which your .pem file was downloaded and open it in terminal. Run the copied command in the terminal to connect to your ec2 instance.

```

,      #
~\  ###
~~ \#####
~~ \|##|
~~  \|#/
~~   V~'__->
~~~   /
~~-. /_/
~/m/
[ec2-user@ip-172-31-41-38 ~] $

```

Amazon Linux 2023
<https://aws.amazon.com/linux/amazon-linux-2023>

4. Run the commands:

```

sudo adduser -m nagios
sudo passwd nagios

```

This creates a user named 'nagios', ensures it has a home directory and sets up a password for it.

```

[ec2-user@ip-172-31-42-56 ~]$ sudo adduser -m nagios
sudo passwd nagios
Changing password for user nagios.
New password:
BAD PASSWORD: The password fails the dictionary check - it is too simplistic/systematic
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-42-56 ~]$ |

```

5. Create a user group named 'nagcmd' to execute nagios commands.

```
sudo groupadd nagcmd
sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
Add users apache and nagios to this user group.
```

```
[ec2-user@ip-172-31-42-56 ~]$ sudo groupdel nagcmd
[ec2-user@ip-172-31-42-56 ~]$ sudo groupadd nagcmd
sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
[ec2-user@ip-172-31-42-56 ~]$ |
```

6. Cd into the downloads folder, make one if there is no such directory and run the following command,
cd ~/downloads
wget <https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz>
wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz

Run the last command to install the latest version of nagios

```
[ec2-user@ip-172-31-42-56 downloads]$ wget https://go.nagios.org/l/975333/2024-09-17/6kqcx
--2024-10-08 01:20:02--  https://go.nagios.org/l/975333/2024-09-17/6kqcx
Resolving go.nagios.org (go.nagios.org)... 3.215.172.219, 3.92.120.28, 18.208.125.13, ...
Connecting to go.nagios.org (go.nagios.org)|3.215.172.219|: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=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969-f2a75b0e2254439d4a81d8 [following]
--2024-10-08 01:20:02--  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=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969-f2a75b0e2254439d4a81d8
```

```
[ec2-user@ip-172-31-42-56 downloads]$ wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
--2024-10-08 01:20:28--  https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251
Connecting to nagios-plugins.org (nagios-plugins.org)|45.56.123.251|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2753049 (2.6M) [application/x-gzip]
Saving to: 'nagios-plugins-2.4.11.tar.gz'

nagios-plugins-2.4.11.tar.gz 100%[=====] 2.62M 6.29MB/s in 0.4s
2024-10-08 01:20:28 (6.29 MB/s) - 'nagios-plugins-2.4.11.tar.gz' saved [2753049/2753049]

[ec2-user@ip-172-31-42-56 downloads]$ |
```

7. tar zxvf nagios-4.5.5.tar.gz This extracts the nagios-core files into the same directory using the tar command.

```
[ec2-user@ip-172-31-42-56 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  
nagios-4.5.5/autoconf-macros/LICENSE.md  
nagios-4.5.5/autoconf-macros/README.md  
nagios-4.5.5/autoconf-macros/add_group_user
```

8. Cd into the nagios folder and run the following command.

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

This command ensures that Nagios uses a specific group (in this case, nagcmd) for executing external commands.

```
[ec2-user@ip-172-31-42-56 downloads]$ cd nagios-4.5.5
[ec2-user@ip-172-31-42-56 nagios-4.5.5]$ |
```

```
[ec2-user@ip-172-31-42-56 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 -a... yes

checking for Kerberos include files... configure: WARNING: could not find include files
checking for pkg-config... pkg-config
checking for SSL headers... configure: error: Cannot find ssl headers
[ec2-user@ip-172-31-42-56 nagios-4.5.5]$ |
```

Error occurs which says that ssl headers cannot be found. To fix the above error, run the ‘sudo yum install openssl-devel’ command.

Then, run the ‘./configure --with-command-group=nagcmd’ command again.

```
[ec2-user@ip-172-31-42-56 nagios-4.5.5]$ sudo yum install openssl-devel
Last metadata expiration check: 0:11:54 ago on Tue Oct  8 01:14:05 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
[ec2-user@ip-172-31-42-56 nagios-4.5.5]$ |
```

- Now we need to compile all the components of this software as given in the instructions.

Run the command

make all,

After that run the following commands:

sudo make install

sudo make install-init

sudo make install-config

sudo make install-commandmode

```
[ec2-user@ip-172-31-42-56 nagios-4.5.5]$ sudo make install
cd ./base && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagios /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagiostats /usr/local/nagios/bin
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/base'
cd ./cgi && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
make install-basic
make[2]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/sbin
[ec2-user@ip-172-31-42-56 nagios-4.5.5]$ sudo make install-init
/usr/bin/install -c -m 755 -d -o root -g root /lib/systemd/system
/usr/bin/install -c -m 755 -o root -g root startup-default-service /lib/systemd/system/nagios.service
[ec2-user@ip-172-31-42-56 nagios-4.5.5]$ sudo make install-config
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc/objects
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/nagios.cfg /usr/local/nagios/etc/nagios.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/cgi.cfg /usr/local/nagios/etc/cgi.cfg
/usr/bin/install -c -b -m 660 -o nagios -g nagios sample-config/resource.cfg /usr/local/nagios/etc/resource.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/templates.cfg /usr/local/nagios
```

10. We need to update the email linked with this server to our email for it to send notifications.

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

```
GNU nano 5.8                               /usr/local/nagios/etc/objects/contacts.cfg                         Modified
# Just one contact defined by default - the Nagios admin (that's you)
# This contact definition inherits a lot of default values from the
# 'generic-contact' template which is defined elsewhere.

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

}

#####
# CONTACT GROUPS
#
#####

# We only have one contact in this simple configuration file, so there is
# no need to create more than one contact group.

define contactgroup {
    contactgroup_name   admins
}
```

11. sudo make install-webconf

This installs the necessary configuration files for the Nagios web interface.

```
[ec2-user@ip-172-31-42-56 nagios-4.5.5]$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf
if [ 0 -eq 1 ]; then \
    ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi

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

[ec2-user@ip-172-31-42-56 nagios-4.5.5]$ |
```

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

This would create a user named 'nagiosadmin' to access the nagios web interface. Save the password that you create as it will be needed later.

```
[ec2-user@ip-172-31-42-56 nagios-4.5.5]$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
[ec2-user@ip-172-31-42-56 nagios-4.5.5]$ |
```

13. cd ~/downloads

tar zxvf nagios-plugins-2.4.11.tar.gz

This will extract the files for nagios plugins.

```
[ec2-user@ip-172-31-42-56 nagios-4.5.5]$ cd ~/downloads
tar zxvf nagios-plugins-2.4.11.tar.gz
nagios-plugins-2.4.11/
nagios-plugins-2.4.11/build-aux/
nagios-plugins-2.4.11/build-aux/compile
nagios-plugins-2.4.11/build-aux/config.guess
nagios-plugins-2.4.11/build-aux/config.rpath
nagios-plugins-2.4.11/build-aux/config.sub
nagios-plugins-2.4.11/build-aux/install-sh
nagios-plugins-2.4.11/build-aux/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
```

14. Next, we must compile all components of this software according to the instructions in the Makefile. To do so, use the following commands:

make

sudo make install

15. sudo chkconfig --add nagios

sudo chkconfig nagios on

This registers the Nagios service with the system ensuring that it can manage the server status.

```
[ec2-user@ip-172-31-42-133 nagios-plugins-2.4.11]$ sudo chkconfig --add nagios
sudo chkconfig nagios on
error reading information on service nagios: No such file or directory
Note: Forwarding request to 'systemctl enable nagios.service'.
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service → /usr/lib/systemd/system/nagios.service.
[ec2-user@ip-172-31-42-133 nagios-plugins-2.4.11]$ |
```

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

Run the following command to check and verify if the configuration files are correct or not.

If you see total errors and warnings as 0, it means the configurations are correct.

```
[ec2-user@ip-172-31-42-133 nagios-plugins-2.4.11]$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
Nagios Core 4.5.5
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2024-09-17
License: GPL

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

Running pre-flight check on configuration data...
  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
```

17. Run the command to start nagios service

`sudo service nagios start`

```
[ec2-user@ip-172-31-42-133 nagios-plugins-2.4.11]$ sudo service nagios start
Redirecting to /bin/systemctl start nagios.service
[ec2-user@ip-172-31-42-133 nagios-plugins-2.4.11]$ |
```

18. `sudo systemctl status nagios`

This checks the status of Nagios. Ensure that it is ‘active(running)’.

```
[ec2-user@ip-172-31-42-133 nagios-plugins-2.4.11]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
   Active: active (running) since Tue 2024-10-08 04:31:34 UTC; 31s ago
     Docs: https://www.nagios.org/documentation
  Process: 64732 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 64734 (nagios)
    Tasks: 6 (limit: 1112)
   Memory: 5.6M
      CPU: 79ms
      CGroup: /system.slice/nagios.service
              ├─64734 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
              ├─64735 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─64736 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─64737 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─64738 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              └─64739 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 08 04:31:34 ip-172-31-42-133.ec2.internal nagios[64734]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successful
Oct 08 04:31:34 ip-172-31-42-133.ec2.internal nagios[64734]: qh: core query handler registered
Oct 08 04:31:34 ip-172-31-42-133.ec2.internal nagios[64734]: qh: echo service query handler registered
Oct 08 04:31:34 ip-172-31-42-133.ec2.internal nagios[64734]: qh: help for the query handler registered
Oct 08 04:31:34 ip-172-31-42-133.ec2.internal nagios[64734]: wproc: Successfully registered manager as @wproc with quer
Oct 08 04:31:34 ip-172-31-42-133.ec2.internal nagios[64734]: wproc: Registry request: name=Core Worker 64738;pid=64738
Oct 08 04:31:34 ip-172-31-42-133.ec2.internal nagios[64734]: wproc: Registry request: name=Core Worker 64737;pid=64737
Oct 08 04:31:34 ip-172-31-42-133.ec2.internal nagios[64734]: wproc: Registry request: name=Core Worker 64736;pid=64736
Oct 08 04:31:34 ip-172-31-42-133.ec2.internal nagios[64734]: wproc: Registry request: name=Core Worker 64735;pid=64739
Oct 08 04:31:34 ip-172-31-42-133.ec2.internal nagios[64734]: Successfully launched command file worker with pid 64739
[lines 1-28/28 (END)]
```

19. Again go to your ec2 instance, click on the id of your instance. On the opened page, copy the public IPV4 address of the instance.

20. In google or any other browser, enter `http://<publicipaddress>/nagios`. You will get a prompt box asking for username and password. Enter the username as `nagiosadmin` and password is the one we set earlier in step 12. If the password is correct you will be directed to the Nagios home page.

The screenshot shows the Nagios Core web interface running on a browser. The URL in the address bar is 18.207.163.135/nagios/. The page title is "Nagios® Core™". The main header includes the Nagios logo and the text "Daemon running with PID 64734". On the left, there is a navigation sidebar with sections for General, Current Status, Problems, Reports, and System. The "Current Status" section is expanded, showing links for Tactical Overview, Map, Hosts, Services, Host Groups, Service Groups, and Problems. The "Problems" section lists "Services (Unhandled)", "Hosts (Unhandled)", and "Network Outages". The "Reports" section includes Availability, Trends, Alerts, History, Summary, Histogram, Notifications, Event Log, and a "Comments" link under the System section. The main content area features a "Get Started" box with bullet points about monitoring infrastructure, changing look, extending with addons, and getting support/training/certified. It also includes "Latest News" and "Don't Miss..." boxes, both of which are currently empty. A "Quick Links" box on the right provides links to Nagios Library, Labs, Exchange, Support, and the official websites. At the bottom, there are copyright notices and a "Page Tour" link.

Conclusion:

In this experiment, we successfully installed and configured Nagios Core and its essential components on an Amazon Linux EC2 instance. This setup involved creating a user for Nagios, configuring user groups, downloading and extracting Nagios and its plugins, and addressing SSL header errors by installing necessary dependencies. The Nagios service was successfully compiled and started, allowing us to access the Nagios web interface via a browser by using the instance's public IP address. With Nagios running, we now have a fully functional monitoring system in place for real-time monitoring and notifications.

Experiment 10

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

Prerequisites: An EC2 instance(nagios-host) with a nagios server already setup. (We can use the instance created in the previous experiment).

1. Go to EC2 on your AWS academy lab. Click on Launch instance and. Give an appropriate name and select Ubuntu as the instance type. Use the same key pair and the security group which was used in previous experiment. Confirm the configurations and click on create instance.

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

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

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

Recents Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Linux

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

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type Free tier eligible

▼ Network settings [Info](#) Edit

Network [Info](#)

-

Subnet [Info](#)

-

Auto-assign public IP [Info](#)

-

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-0a6f83cec8349432a X
VPC: vpc-0f32f0f5d27c3d899

Compare security group rules

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

Instances (3) [Info](#) Last updated less than a minute ago [Connect](#) Instance state ▾ Actions ▾ Launch instances ▾

| <input type="checkbox"/> Name  ▾ | Instance ID | Instance state ▾ | Instance type ▾ | Status check | Alarm status |
|---|---------------------|--|-----------------|---|--------------|
| <input type="checkbox"/> nagios-host | i-0d78005293a497804 |  Terminated   | t2.micro | - | ... |
| <input type="checkbox"/> nagios-host | i-09c23636f09303355 |  Running   | t2.micro |  2/2 checks passed | ... |
| <input type="checkbox"/> exp10 | i-0a646bbc865f5a2dd |  Running   | t2.micro |  Initializing | ... |

You should have both the host instance and the newly created instance.

2. Now click on the instance id for the newly created instance, click on connect. Go to the ssh tab and copy the example command. Open the folder where .pem file for key pair was installed in your terminal and run the copied command. This will connect your terminal to the ec2 instance. Do this for the host instance as well.
3. To verify whether the nagios service is running or not, run the following command

```
ps -ef | grep nagios
```

Perform the following commands in the host instance until specified to do otherwise.

```
[ec2-user@ip-172-31-42-133 ~]$ ps -ef | grep nagios
nagios  64734      1  0 04:31 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios  64735  64734  0 04:31 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagio
s.qh
nagios  64736  64734  0 04:31 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagio
s.qh
nagios  64737  64734  0 04:31 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagio
s.qh
nagios  64738  64734  0 04:31 ?        00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagio
s.qh
nagios  64739  64734  0 04:31 ?        00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
root   64742      2398 0 04:32 pts/0    00:00:00 sudo systemctl status nagios
root   64744  64742  0 04:32 pts/1    00:00:00 sudo systemctl status nagios
root   64745  64744  0 04:32 pts/1    00:00:00 systemctl status nagios
ec2-user 65944  65905  0 04:51 pts/2    00:00:00 grep --color=auto nagios
```

4. sudo su

```
mkdir -p /usr/local/nagios/etc/objects/monitorhosts
mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
```

This makes you the root user and creates two folders with the above paths.

```
[ec2-user@ip-172-31-42-133 ~]$ sudo su
mkdir -p /usr/local/nagios/etc/objects/monitorhosts
mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-42-133 ec2-user]# |
```

5. Open the config file using the nano editor as we need to make some changes in the configuration.

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

Change hostname and alias from 'hostname' to 'linuxserver'.
Change address to the public ip address of the ubuntu-client instance.
Change hostgroup_name to 'linux-servers1'.
Change all the subsequent occurrences of hostname in the file from 'localhost' to 'linuxserver'.

```
GNU nano 5.8          /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg      Modified
#####
# HOST DEFINITION
#
#####

# Define a host for the local machine

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

    host_name     linuxserver
    alias         linuxserver|
    address       127.0.0.1
}

#####
#
```

6. Open the Nagios config file using the following command:

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

Then, add the following line to the config file:

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

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

# Definitions for monitoring the local (Linux) host
cfg_file=/usr/local/nagios/etc/objects/localhost.cfg

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

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

# Definitions for monitoring a network printer
#cfg_file=/usr/local/nagios/etc/objects/printer.cfg

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

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

7. To check and verify if the configurations are correct or not run the following command:

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

```
ocalhost.cfg', starting on line 58)
  Read object config files okay...

Running pre-flight check on configuration data...

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

Checking for circular paths...
  Checked 2 hosts
  Checked 0 service dependencies
  Checked 0 host dependencies
  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
```

In the end you will see “Total warning” and “total error” as 0, this confirms that the configurations is correct.

8. Now we will restart the nagios server to implement the above made changes.
service nagios restart

```
[root@ip-172-31-42-133 ec2-user]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
```

9. systemctl status nagios

Using the above command, we check the status of the nagios server and ensure that it is active (running).

```
[root@ip-172-31-42-133 ec2-user]# systemctl status nagios
● nagios.service - Nagios Core 4.5.5
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
   Active: active (running) since Tue 2024-10-08 05:04:49 UTC; 33s ago
     Docs: https://www.nagios.org/documentation
   Process: 66879 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0>
  Process: 66880 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SU>
 Main PID: 66881 (nagios)
    Tasks: 6 (limit: 1112)
   Memory: 4.0M
      CPU: 23ms
     CGroup: /system.slice/nagios.service
             └─66881 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
               ├─66882 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
               ├─66883 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
               ├─66884 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
               ├─66885 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
               └─66886 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 08 05:04:49 ip-172-31-42-133.ec2.internal nagios[66881]: wproc: Registry request: name=Core Worker 66882;pid=66882
Oct 08 05:04:49 ip-172-31-42-133.ec2.internal nagios[66881]: Warning: Duplicate definition found for service 'HTTP' on h>
Oct 08 05:04:49 ip-172-31-42-133.ec2.internal nagios[66881]: Warning: Duplicate definition found for service 'SSH' on h>
Oct 08 05:04:49 ip-172-31-42-133.ec2.internal nagios[66881]: Warning: Duplicate definition found for service 'Swap Usag>
Oct 08 05:04:49 ip-172-31-42-133.ec2.internal nagios[66881]: Warning: Duplicate definition found for service 'Current L>
Oct 08 05:04:49 ip-172-31-42-133.ec2.internal nagios[66881]: Warning: Duplicate definition found for service 'Total Pro>
Oct 08 05:04:49 ip-172-31-42-133.ec2.internal nagios[66881]: Warning: Duplicate definition found for service 'Current U>
Oct 08 05:04:49 ip-172-31-42-133.ec2.internal nagios[66881]: Warning: Duplicate definition found for service 'Root Part>
Oct 08 05:04:49 ip-172-31-42-133.ec2.internal nagios[66881]: Warning: Duplicate definition found for service 'PING' on h>
Oct 08 05:04:49 ip-172-31-42-133.ec2.internal nagios[66881]: Successfully launched command file worker with pid 66886
```

- Now open the terminal which is connected to the ubuntu instance. If not connect, follow the 2nd step in similar fashion to connect to the instance, run the following command in ubuntu instance.

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

```
ubuntu@ip-172-31-42-172:~$ sudo apt update -y
sudo apt install gcc -y
sudo apt install -y nagios-nrpe-server nagios-plugins
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [382 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
```

- Run the following command:

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

The above command opens the NRPE config file. Here, we need to add the public IP address of our host nagios-host instance to the NRPE configuration file. Under allowed_hosts, add the nagios-host public IPv4 address. The public ip address can be seen by click on the instance id of the instance in EC2 dashboard.

```

GNU nano 7.2
/etc/nagios/nrpe.cfg *
# 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=127.0.0.1,54.90.219.49

# COMMAND ARGUMENT PROCESSING

```

12. Once everything is completed, open the nagios dashboard in browser with url <http://<publicipaddress>/nagiso>. Click on the hosts and we will see that linuxserver has been added as a host

| Host | Status | Last Check | Duration | Status Information |
|-------------|--------|---------------------|---------------|---|
| linuxserver | UP | 09-29-2024 12:21:02 | 0d 0h 9m 55s | PING OK - Packet loss = 0%, RTA = 1.02 ms |
| localhost | UP | 09-29-2024 12:20:02 | 0d 0h 55m 54s | PING OK - Packet loss = 0%, RTA = 0.04 ms |

Click on 'linuxserver'. Here, we can access all information about the 'linuxserver' host.

Host Information

- Last Updated: Sun Sep 29 12:23:21 UTC 2024
- Updated every 90 seconds
- Nagios® Core™ 4.5.5 - www.nagios.org
- Logged in as nagiosadmin

Host linuxserver (linuxserver)

- Member of linux-servers1
- IP: 52.91.101.68

Host State Information

| Host Status: | UP (for 0d 0h 11m 41s) |
|------------------------------|--|
| Status Information: | PING OK - Packet loss = 0%, RTA = 1.02 ms |
| Performance Data: | rta=1.023000ms;3000.000000;5000.000000;0.000000 pl=0%;80;100;0 |
| Current Attempt: | 1/10 (HARD state) |
| Last Check Time: | 09-29-2024 12:21:02 |
| Check Type: | ACTIVE |
| Check Latency / Duration: | 0.000 / 4.122 seconds |
| Next Scheduled Active Check: | 09-29-2024 12:26:02 |
| Last State Change: | 09-29-2024 12:11:40 |
| Last Notification: | N/A (notification 0) |
| Is This Host Flapping? | NO (0.00% state change) |
| In Scheduled Downtime? | NO |
| Last Update: | 09-29-2024 12:23:19 (0d 0h 0m 2s ago) |

Host Commands

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

Host Comments

Add a new comment | Delete all comments

Click on 'Services'. Here, we can see all the services that are being monitored by 'linuxserver'.

Current Network Status

- Last Updated: Sun Sep 29 12:24:32 UTC 2024
- Updated every 90 seconds
- Nagios® Core™ 4.5.5 - www.nagios.org
- Logged in as nagiosadmin

Host Status Totals

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

Service Status Totals

| Ok | Warning | Unknown | Critical | Pending |
|----|---------|---------|----------|---------|
| 12 | 1 | 0 | 3 | 0 |

Service Status Details For All Hosts

| Host | Service | Status | Last Check | Duration | Attempts | Status Information |
|-------------|-----------------|----------|---------------------|---------------|----------|--|
| linuxserver | Current Load | OK | 09-28-2024 12:22:17 | 0d 0h 12m 15s | 1/4 | OK - load average: 0.00, 0.00, 0.00 |
| | Current Users | OK | 09-29-2024 12:22:55 | 0d 0h 11m 37s | 1/4 | USERS OK - 5 users currently logged in |
| | HTTP | CRITICAL | 09-29-2024 12:21:32 | 0d 0h 8m 0s | 4/4 | connect to address 52.91.101.68 and port 80: Connection refused |
| | PING | OK | 09-29-2024 12:24:10 | 0d 0h 10m 22s | 1/4 | PING OK - Packet loss = 0%, RTA = 0.74 ms |
| | Root Partition | OK | 09-29-2024 12:19:47 | 0d 0h 9m 45s | 1/4 | DISK OK - free space: / 6107 MB (75.24% inode=98%) |
| | SSH | OK | 09-29-2024 12:20:25 | 0d 0h 9m 7s | 1/4 | SSH OK - OpenSSH_9.8p1 Ubuntu-3ubuntu13.4 (protocol 2.0) |
| | Swap Usage | CRITICAL | 09-29-2024 12:24:02 | 0d 0h 5m 30s | 4/4 | SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size |
| | Total Processes | OK | 09-29-2024 12:21:40 | 0d 0h 7m 52s | 1/4 | PROCS OK: 43 processes with STATE = RSZDT |
| localhost | Current Load | OK | 09-29-2024 12:21:17 | 0d 0h 58m 15s | 1/4 | OK - load average: 0.00, 0.00, 0.00 |
| | Current Users | OK | 09-29-2024 12:21:56 | 0d 0h 57m 37s | 1/4 | USERS OK - 5 users currently logged in |
| | HTTP | WARNING | 09-29-2024 12:20:30 | 0d 0h 54m 1s | 4/4 | HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.001 second response time |
| | PING | OK | 09-29-2024 12:23:10 | 0d 0h 56m 22s | 1/4 | PING OK - Packet loss > 0%, RTA = 0.05 ms |
| | Root Partition | OK | 09-29-2024 12:23:47 | 0d 0h 55m 45s | 1/4 | DISK OK - free space: / 6107 MB (75.24% inode=98%) |
| | SSH | OK | 09-29-2024 12:24:25 | 0d 0h 55m 7s | 1/4 | SSH OK - OpenSSH_8.7 (protocol 2.0) |
| | Swap Usage | CRITICAL | 09-29-2024 12:23:02 | 0d 0h 51m 30s | 4/4 | SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size |
| | Total Processes | OK | 09-29-2024 12:20:40 | 0d 0h 53m 52s | 1/4 | PROCS OK: 43 processes with STATE = RSZDT |

Results 1 - 16 of 16 Matching Services

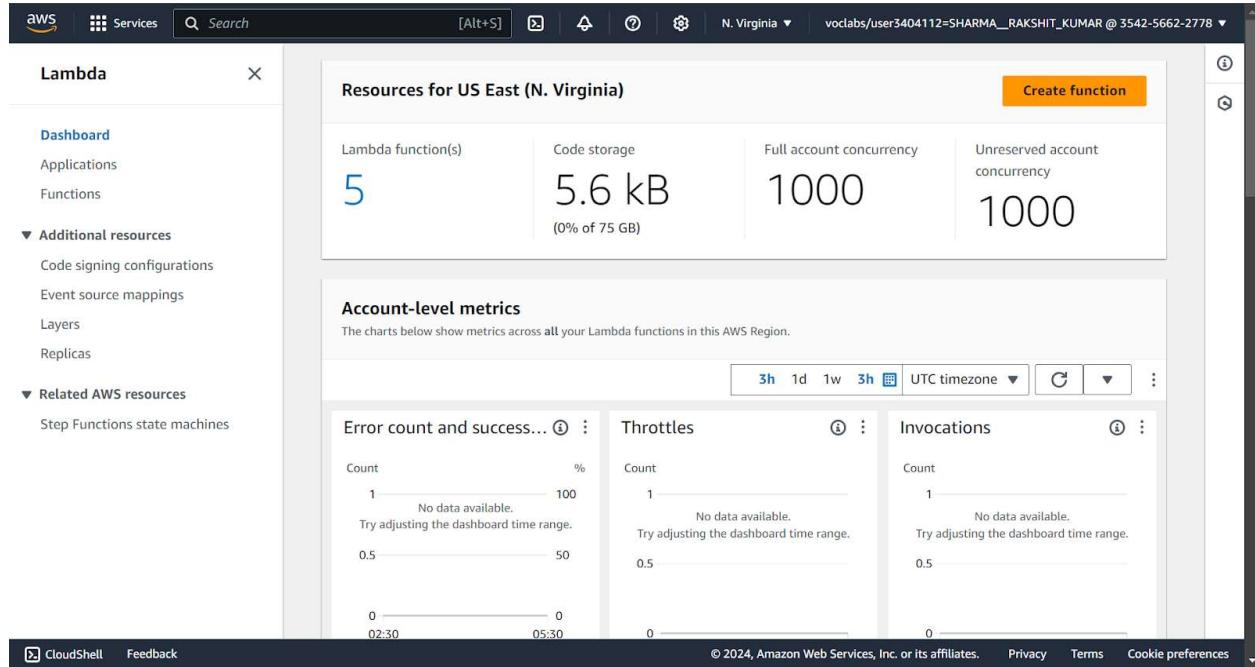
Conclusion:

In this experiment, we successfully performed port, service, and Linux server monitoring using Nagios. After setting up a new Ubuntu EC2 instance, we configured it as a monitored client by creating appropriate Nagios configuration files and modifying the host instance's settings. We installed the Nagios NRPE server and plugins on the Ubuntu instance, added the public IP of the Nagios host to the NRPE config file, and verified the changes in the Nagios dashboard.

Experiment 11

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

1. Log into your AWS academy account and search for Lambda service. Click on create function to create a new lambda function.



2. Give a name to your lambda function. Select the language you want to use to write the functions. We will use Python 3.12, Architecture x86. Select Execution role to Create a new role with basic Lambda permissions.

Create function Info

Choose one of the following options to create your function.

Author from scratch
Start with a simple Hello World example.

Use a blueprint
Build a Lambda application from sample code and configuration presets for common use cases.

Container image
Select a container image to deploy for your function.

Basic information

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

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 (_).

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

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

You will be able to see the created function in Functions tab

The screenshot shows two related interfaces for managing an AWS Lambda function named 'Rakshit_Lambda'.

Top Interface (Function Overview):

- Path:** Lambda > Functions > Rakshit_Lambda
- Title:** Rakshit_Lambda
- Actions:** Throttle, Copy ARN, Actions ▾
- Function overview:** Diagram (selected), Template
- Diagram View:** Shows the function icon 'Rakshit_Lambda' and 'Layers (0)'. Buttons: + Add trigger, + Add destination.
- Right Panel:** Description: -, Last modified: in 1 second, Function ARN: arn:aws:lambda:us-east-1:354256622778:function:Rakshit_Lambda, Function URL: Info, -.

Bottom Interface (Test Tab):

- Tab Bar:** File, Edit, Find, View, Go, Tools, Window, Test (selected), Deploy
- Search:** Go to Anything (Ctrl-P)
- Environment:** Environment Variants (lambda_function)
- Code Editor:** Shows the 'lambda_function.py' file content:

```
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
```

3. Scroll down and go to the configuration tab. In General configuration click on edit to change the configuration.

The screenshot shows the AWS Lambda Configuration page. The top navigation bar includes tabs for Code, Test, Monitor, Configuration (which is selected), Aliases, and Versions. On the left, a sidebar lists General configuration options: Triggers, Permissions, Destinations, Function URL, Environment variables, Tags, VPC, and RDS databases. The main content area displays the General configuration settings:

| General configuration | | |
|-----------------------|-----------|-------------------|
| Description | Memory | Ephemeral storage |
| - | 128 MB | 512 MB |
| Timeout | SnapStart | |
| 0 min 3 sec | None | |

An 'Edit' button is located in the top right corner of the configuration table.

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

- Now go to the test tab and select Create new event. Give the event an appropriate name, keep the sharing settings as private and template as hello world.

The screenshot shows the AWS Lambda Test event configuration page. It includes fields for Test event action (Create new event selected), Event name (RakshitEvent), and Event sharing settings (Private selected). The Template dropdown is set to hello-world. The Event JSON section contains the following JSON code:

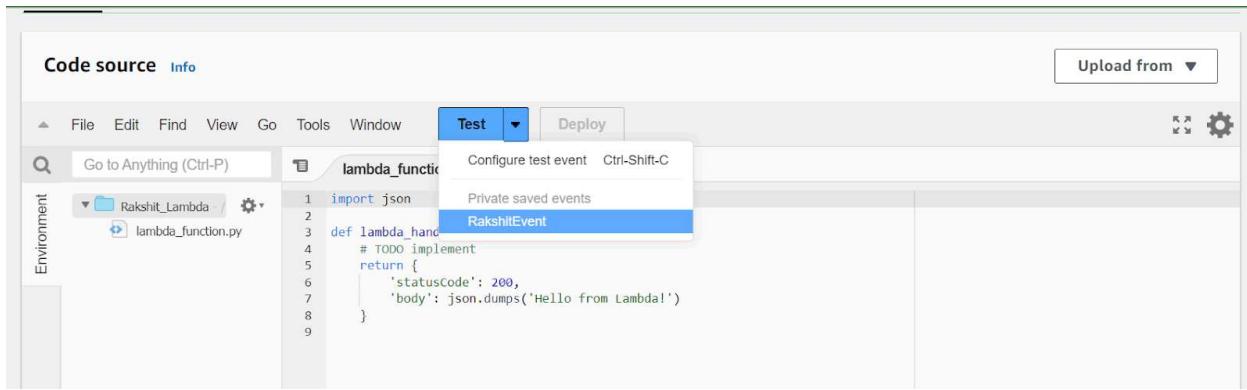
```

1 [
2   "key1": "value1",
3   "key2": "value2".

```

A 'Format JSON' button is located in the top right corner of the Event JSON editor.

- Now In Code section select the created event from the dropdown of test then click on test . Observe the output.



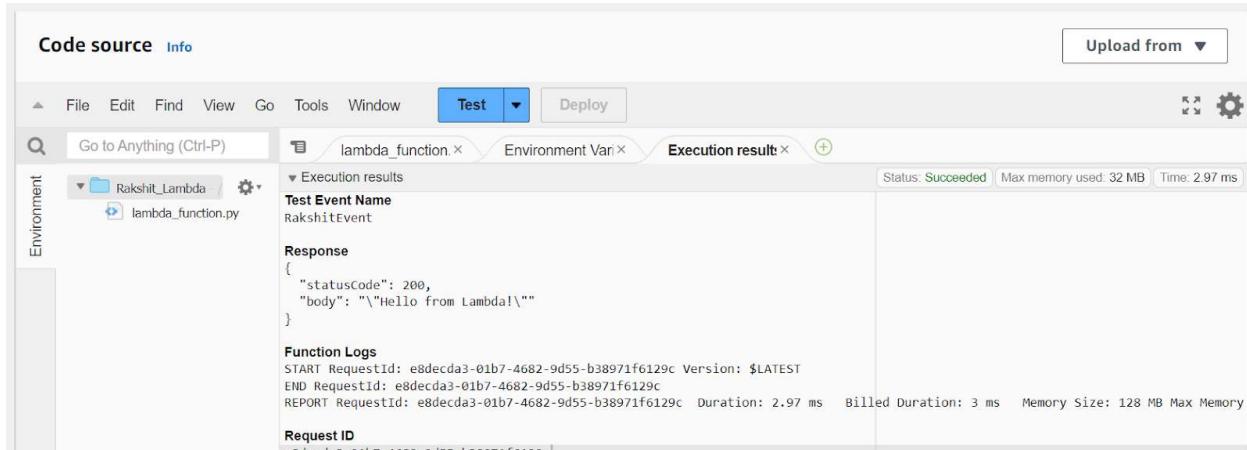
The screenshot shows the AWS Lambda Test interface. In the code editor, there is a Python script named `lambda_function.py` with the following content:

```

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     }

```

A dropdown menu is open under the `Test` button, showing options: `Configure test event`, `Ctrl-Shift-C`, `Private saved events`, and `RakshitEvent`. The `RakshitEvent` option is highlighted.



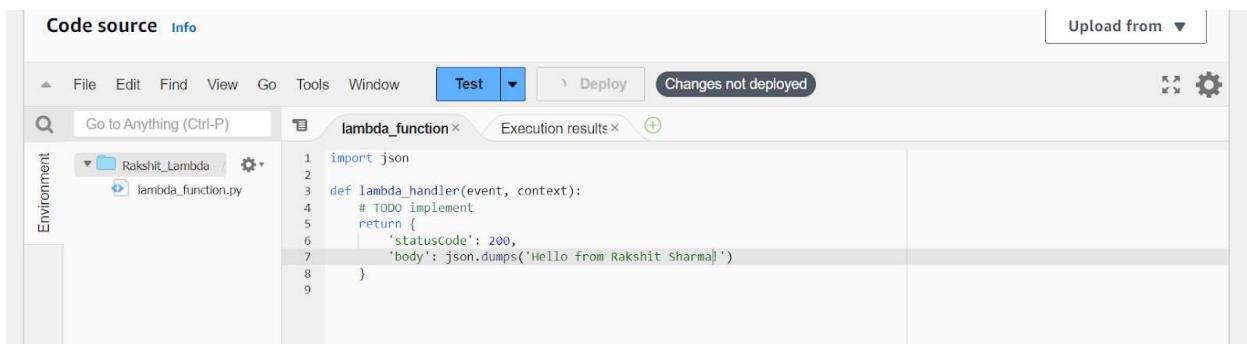
The screenshot shows the AWS Lambda Test interface after a test run. The execution results are displayed:

- Test Event Name:** RakshitEvent
- Response:**

```
{
    "statusCode": 200,
    "body": "Hello from Lambda!"
}
```
- Function Logs:**

```
START RequestId: e8decda3-01b7-4682-9d55-b38971f6129c Version: $LATEST
END RequestId: e8decda3-01b7-4682-9d55-b38971f6129c
REPORT RequestId: e8decda3-01b7-4682-9d55-b38971f6129c Duration: 2.97 ms Billed Duration: 3 ms Memory Size: 128 MB Max Memory
```
- Request ID:** e8decda3-01b7-4682-9d55-b38971f6129c

6. We can also edit the lambda function code. I have added my name in the print statement.



The screenshot shows the AWS Lambda Test interface with the following code changes made in the `lambda_function.py` file:

```

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

```

A message `Changes not deployed` is displayed in the top right corner of the interface.

To save the changes click on deploy. And again select the event and click on Test. Check the output displayed.

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 toolbar with 'File', 'Edit', 'Find', 'View', 'Go', 'Tools', 'Window', 'Test' (which is selected), and 'Deploy'. On the left, there's a sidebar titled 'Environment' with a search bar 'Go to Anything (Ctrl-P)' and a folder named 'Rakshit_Lambda' containing 'lambda_function.py'. The main area has three tabs: 'Execution results' (selected), 'Execution result' (with a status of 'Succeeded'), and 'Environment Vari'. The 'Execution results' tab displays the 'Test Event Name' as 'RakshitEvent' and the 'Response' as a JSON object:

```
{ "statusCode": 200, "body": "\"Hello from Rakshit Sharma!\""}  
Function Logs  
START RequestId: 7ad0f22c-cf58-4269-8898-1bb23c216184 Version: $LATEST  
END RequestId: 7ad0f22c-cf58-4269-8898-1bb23c216184  
REPORT RequestId: 7ad0f22c-cf58-4269-8898-1bb23c216184 Duration: 2.16 ms Billed Duration: 3 ms Memory Size: 128 MB Max Memory Request ID 7ad0f22c-cf58-4269-8898-1bb23c216184
```

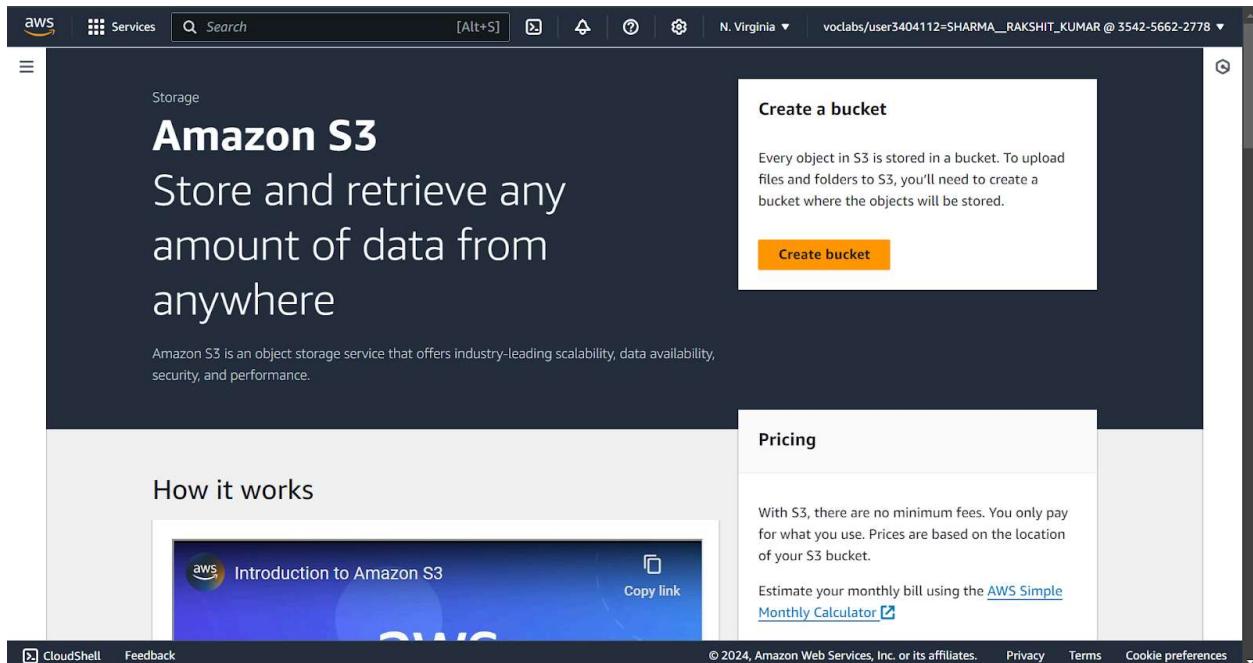
Conclusion:

In this experiment, we explored AWS Lambda, created a basic Lambda function, and tested it using Python 3.12. We walked through creating a Lambda function, configuring its settings, and testing it with a predefined event. We also demonstrated how to modify the function's code, deployed the changes, and observed the output. This experiment gave us insight into AWS Lambda's workflow, showing how serverless functions can be easily created, tested, and deployed, making it a powerful tool for event-driven applications.

Experiment 12

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

1. Login to your AWS account. Search for S3 service, click on create bucket to make a new bucket.



2. Select the bucket type as general purpose, give your bucket an appropriate name. Uncheck the block all public access. Keep rest of the settings as default and click on create bucket.

The screenshot shows the 'Create bucket' page in the AWS S3 console. The 'General configuration' section is selected. The 'Bucket name' field contains 'myawsbucket'. The 'Bucket type' dropdown is set to 'General purpose'. Other options like 'Directory' are also shown. The 'Bucket location' dropdown is set to 'US East (N. Virginia) us-east-1'. A success message at the bottom indicates the bucket 'advdevops12' was created successfully.

Create bucket Info

Buckets are containers for data stored in S3.

General configuration

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

Bucket type Info

- General purpose

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

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

Bucket name Info

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

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

Format: [s3://myawsbucket.location](#)

Success! Successfully created bucket "advdevops12"

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

[View details](#) X

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

Account snapshot - updated every 24 hours All AWS Regions

Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

[View Storage Lens dashboard](#)

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

General purpose buckets (1) Info All AWS Regions

Buckets are containers for data stored in S3.

| Name | AWS Region | IAM Access Analyzer | Creation date |
|-----------------------------|---------------------------------|---|---|
| advdevops12 | US East (N. Virginia) us-east-1 | View analyzer for us-east-1 | October 10, 2024, 15:54:28 (UTC+05:30) |

[Create bucket](#)

- Search for lambda and open its console. Click on create function to make a new function.

| Functions (6) | | | | | |
|---|--------------------------------------|---|--------------|------------|---------------|
| Last fetched 0 seconds ago | | | | | |
| Actions Create function | | | | | |
| <input type="text"/> Filter by tags and attributes or search by keyword | | | | | |
| <input type="checkbox"/> | Function name | Description | Package type | Runtime | Last modified |
| <input type="checkbox"/> | RoleCreationFunction | Create SLR if absent | Zip | Python 3.8 | 2 months ago |
| <input type="checkbox"/> | RedshiftOverwatch | Deletes Redshift Cluster if the count is more than 2. | Zip | Python 3.8 | 2 months ago |
| <input type="checkbox"/> | ModLabRole | updates LabRole to allow it to assume itself | Zip | Python 3.8 | 2 months ago |

4. Give a name to your lambda function. Select the language you want to use to write the functions. We will use Python 3.12, Architecture x86. Select Execution role to Create a new role with basic Lambda permissions.

Create function [Info](#)

Choose one of the following options to create your function.

Author from scratch
Start with a simple Hello World example.

Use a blueprint
Build a Lambda application from sample code and configuration presets for common use cases.

Container image
Select a container image to deploy for your function.

Basic information

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

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 (_).

Runtime [Info](#)
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
 [▼](#) [C](#)

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

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

Function overview [Info](#) [Export to Application Composer](#) [Download ▾](#)

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

 Rakshit_Lambda

 Layers (0)

[+ Add trigger](#) [+ Add destination](#)

Description
-

Last modified
5 hours ago

Function ARN
 arn:aws:lambda:us-east-1:354256622778:function:Rakshit_Lambda

Function URL [Info](#)
-

5. Scroll down and go to the configuration tab. In General configuration click on edit to change the configuration.

The screenshot shows the AWS Lambda Configuration page for a function named "General configuration". The "Configuration" tab is selected. On the left, a sidebar lists options: Triggers, Permissions, Destinations, Function URL, Environment variables, Tags, VPC, and RDS databases. The main area displays the "General configuration" settings:

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

Below this, there is a note: "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.

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

- Now to create a new event, go to the test tab. Create a new event, give a name to the event and select Event Sharing to private, and select s3 put template.

Test event action

Create new event Edit saved event

Event name

RakshitBucket

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

s3-put

Event JSON

```

4   "eventVersion": "2.0",
5   "eventSource": "aws:s3",
6   "awsRegion": "us-east-1",
7   "eventTime": "1970-01-01T00:00:00.000Z",
8   "eventName": "ObjectCreated:Put",
9   "userIdentity": {
10     "principalId": "EXAMPLE"
11   }

```

Format JSON

7. Now in the lambda function click on 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/advdevops12 X G

Bucket region: us-east-1

Event types
Select the events that you want to have trigger the Lambda function. You can optionally set 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 X

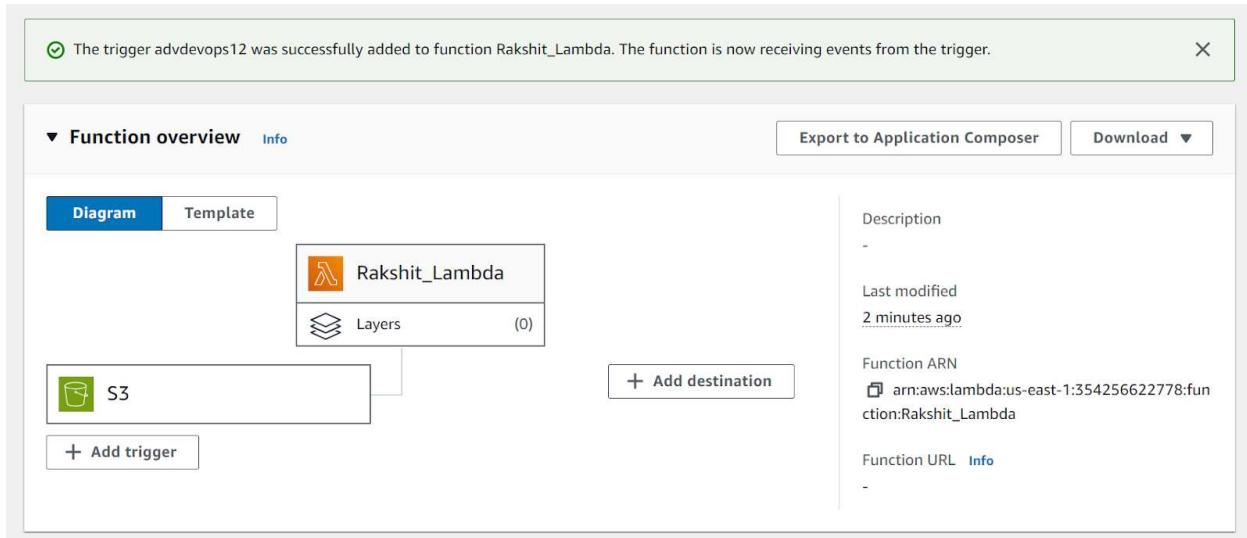
Prefix - optional
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters. Any [special characters](#) must be URL encoded.

ExperimentImage

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

.jpg

Select the source as s3, and search for the bucket that we created earlier. If you want you can add the prefix for the image. Keep the rest of the setting as default and add trigger.



- Now Write code that logs a message like "An Image has been added" when triggered. Save the file and click on deploy.

```
Code source Info Upload from ▾  
File Edit Find View Go Tools Window Test Deploy Environment Vari  
Go to Anything (Ctrl-P)  
λ lambda_function Execution results Environment Vari +  
Rakshit_Lambda λ lambda_function.py  
1 import json  
2  
3 def lambda_handler(event, context):  
4     # TODO implement  
5     bucket_name = event['Records'][0]['s3']['bucket']['name']  
6     object_key = event['Records'][0]['s3']['object']['key']  
7  
8     print(f"An image has been added {bucket_name} : {object_key}")  
9     return {  
10         'statusCode': 200,  
11         'body': json.dumps('Log entry successful.')  
12     }  
13
```

- Now we will upload an image to our bucket.

The screenshot shows the AWS S3 'Upload' interface. At the top, it says 'Amazon S3 > Buckets > advdevops12 > Upload'. Below that is a 'Upload' section with a 'Drag and drop files and folders you want to upload here, or choose Add files or Add folder.' button. A table titled 'Files and folders (1 Total, 74.8 KB)' lists one item: 'Screenshot 2024-10-10 161236.png' (image/png). There are 'Remove', 'Add files', and 'Add folder' buttons above the table. A search bar labeled 'Find by name' is also present. The 'Destination' section shows a green success message: 'Upload succeeded' with a link to 'View details below.' A note below it says 'The information below will no longer be available after you navigate away from this page.' The 'Summary' section shows the destination as 's3://advdevops12' and the results: 'Succeeded' (1 file, 74.8 KB (100.00%)) and 'Failed' (0 files, 0 B (0%)). Below this is a 'Files and folders' section with a table showing the same file details.

| Name | Folder | Type | Size | Status | Error |
|-----------------|--------|-----------|---------|-----------|-------|
| Screenshot 2... | - | image/png | 74.8 KB | Succeeded | - |

10. Now in lambda function, select the event which we created now and click on test. Check if it is logging the string that we added in our code.

The screenshot shows the AWS Lambda Test & Deploy interface. In the top navigation bar, the 'Test' button is highlighted. The main area displays the 'Execution results' tab for a function named 'lambda_function'. The 'Test Event Name' is set to 'RakshitBucket'. The 'Response' section shows a JSON object with a 'statusCode' of 200 and a 'body' of '\"Log entry successfull.''. Below this, the 'Function Logs' section provides detailed log information for a single request, including RequestId, Start and End times, Duration, Billed Duration, and Memory Size. The 'Request ID' is listed as f10c152f-7959-44f6-ae85-600b211d12a3.

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

The screenshot shows the AWS CloudWatch Log Events interface. The URL in the address bar is CloudWatch > Log groups > /aws/lambda/Rakshit_Lambda > 2024/10/10/[\$.LATEST]3c7e1ef488474437ace14e4aa7d2c688. The main area is titled 'Log events' and contains a table of log events. The columns are 'Timestamp' and 'Message'. The table shows several log entries corresponding to the Lambda function's execution, including INIT, START, REPORT, and END requests for the 'test%2Fkey' bucket. The most recent event is a REPORT request from 2024-10-10T10:44:26.041Z.

Conclusion:

In this experiment, we successfully created a Lambda function that logs "An Image has been added" when an object is uploaded to a specific S3 bucket. We started by creating an S3 bucket and setting up a Lambda function with Python 3.12. We configured a trigger for the Lambda function to monitor the S3 bucket and added code to log the message when an image is uploaded. After deploying the code and testing it by uploading an image, we verified the logs in CloudWatch. This experiment demonstrated the use of AWS Lambda for event-driven automation, specifically integrating Lambda with S3.