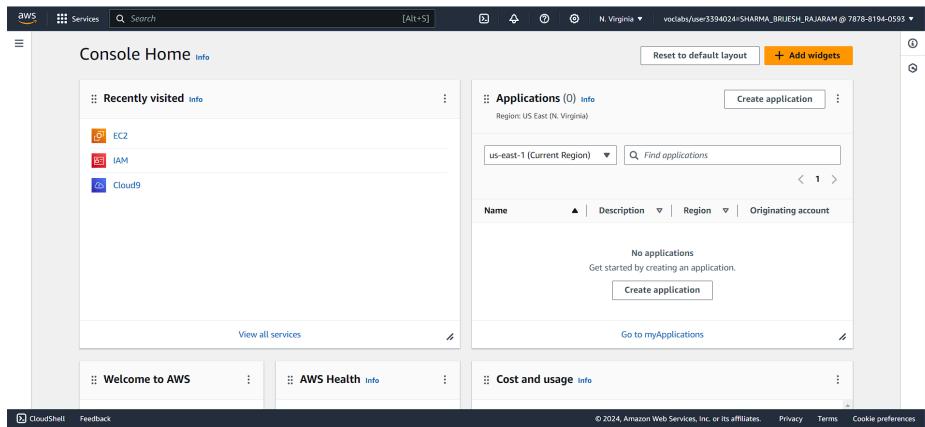


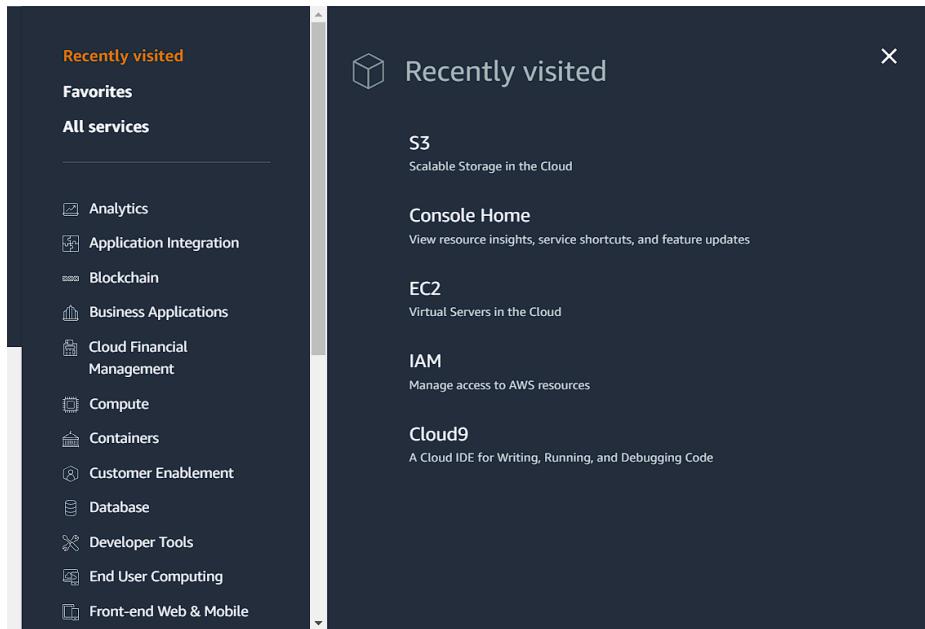
Hosting a static website on Amazon Web Services (S3)

1) Open the AWS console home



The screenshot shows the AWS Console Home page. On the left, there's a sidebar with 'Recently visited' (EC2, IAM, Cloud9), 'Welcome to AWS', 'AWS Health', and 'Cost and usage'. The main area has a header for 'Applications (0)' with a 'Create application' button. It shows a message: 'No applications. Get started by creating an application.' Below this is another 'Create application' button and a link to 'Go to myApplications'. The bottom of the page includes standard AWS footer links like 'cloudShell', 'Feedback', and copyright information.

2) Navigate to the S3 to host the website



The screenshot shows the AWS Services menu. On the left, under 'All services', various services are listed: Analytics, Application Integration, Blockchain, Business Applications, Cloud Financial Management, Compute, Containers, Customer Enablement, Database, Developer Tools, End User Computing, and Front-end Web & Mobile. On the right, a 'Recently visited' panel is open, showing the 'S3' service with the sub-section 'Console Home' selected. Other items in the panel include EC2, IAM, and Cloud9. The top of the page has a dark header with the AWS logo and a search bar.

3) On S3, click on create bucket

Create a bucket

Every object in S3 is stored in a bucket. To upload files and folders to S3, you'll need to create a bucket where the objects will be stored.

[Create bucket](#)

- 4) Click on Bucket type as General Purpose and name the bucket.

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

Bucket type | [Info](#)

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

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

Bucket name | [Info](#)

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

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

Format: s3://bucket/prefix

- 5) Keep the default settings intact, checking for bucket versioning as disable and bucket key enabled.

Object Ownership [Info](#)

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

ACLs disabled (recommended)

All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

ACLs enabled

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

Object Ownership
Bucket owner enforced

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your application 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 case. [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.

Bucket Versioning

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

Bucket Versioning
 Disable
 Enable

Tags - optional (0)

You can use bucket tags to track storage costs and organize buckets. [Learn more](#)

No tags associated with this bucket.
[Add tag](#)

Default encryption [Info](#)

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 (DSS-E-KMS)
Secure your objects with two separate layers of encryption. For details on pricing, see [DSS-E-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 DSS-E-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

- 6) After successfully creating the bucket, click on bucket name to change the settings to host the website.

The screenshot shows the AWS S3 Buckets page. At the top, a green banner indicates "Successfully created bucket 'brijeshkabucket'". Below the banner, there's an "Account snapshot" section with a link to "View details". The main area shows two tabs: "General purpose buckets" (selected) and "Directory buckets". Under "General purpose buckets", there's a table with one row for "brijeshkabucket". The table columns include Name, AWS Region, IAM Access Analyzer, and Creation date. The bucket name is "brijeshkabucket", the region is "All AWS Regions", the creation date is "August 7, 2024, 20:38:57 (UTC+05:30)", and there are buttons for "Copy ARN", "Empty", "Delete", and "Create bucket".

- 7) Go on Permissions tab and check for Block public access

The screenshot shows the "brijeshkabucket" page under the "Objects" tab. The top navigation bar includes "Objects" (selected), "Properties", "Permissions", "Metrics", "Management", and "Access Points". The main content area shows a table with one row labeled "No objects". It includes columns for Name, Type, Last modified, Size, and Storage class. A "Upload" button is visible at the bottom. The "Permissions" tab is also visible in the navigation bar.

- 8) Block public access is default on, we need to uncheck it to ensure the hosted website is public.

The screenshot shows the "Block public access (bucket settings)" page. It explains that public access is granted through ACLs, bucket policies, access point policies, or all. It recommends turning on "Block all public access" to ensure no public access. A note states that these settings apply only to the current bucket and its access points. There are two sections: "Block all public access" (with "On" checked) and "Individual Block Public Access settings for this bucket". An "Edit" button is located in the top right corner.

- 9) Now the block public access option is unchecked and hence the website can be hosted successfully.

Block public access (bucket settings)

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 all your S3 buckets and 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 your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

Block all public access

Off

Individual Block Public Access settings for this bucket

- 10) Now, navigate to the edit bucket policy in Properties tab to provide access to the services.

Amazon S3 > Buckets > brijeshkabucket > Edit bucket policy

Edit bucket policy [Info](#)

Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. [Learn more](#)

Bucket ARN

arn:aws:s3:::brijeshkabucket

Policy

```
1 |
```

Edit statement

Select a statement

Select an existing statement in the policy or add a new statement.

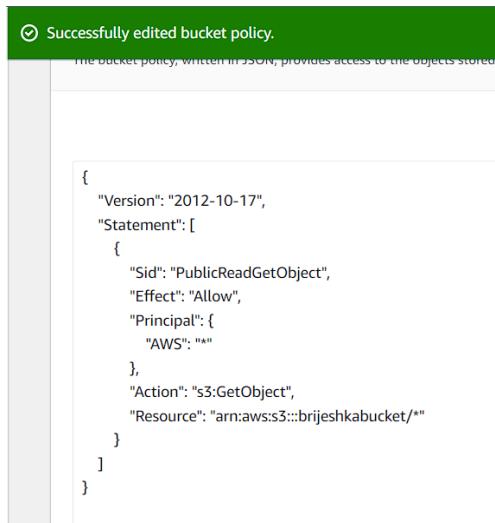
- 11) Fill the following policy in the empty policy space. Ensure that you change the name of the bucket in Resource with the name of your bucket.

Policy

```

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

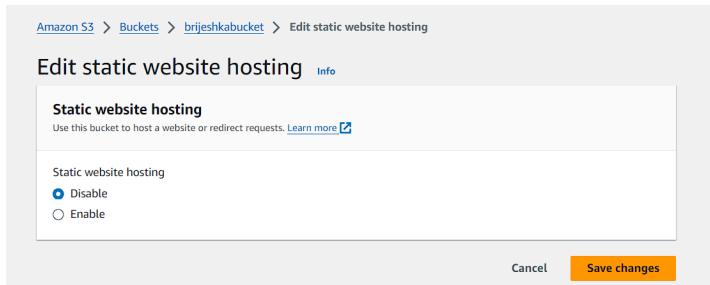
12) After saving the changes, you will see a message.



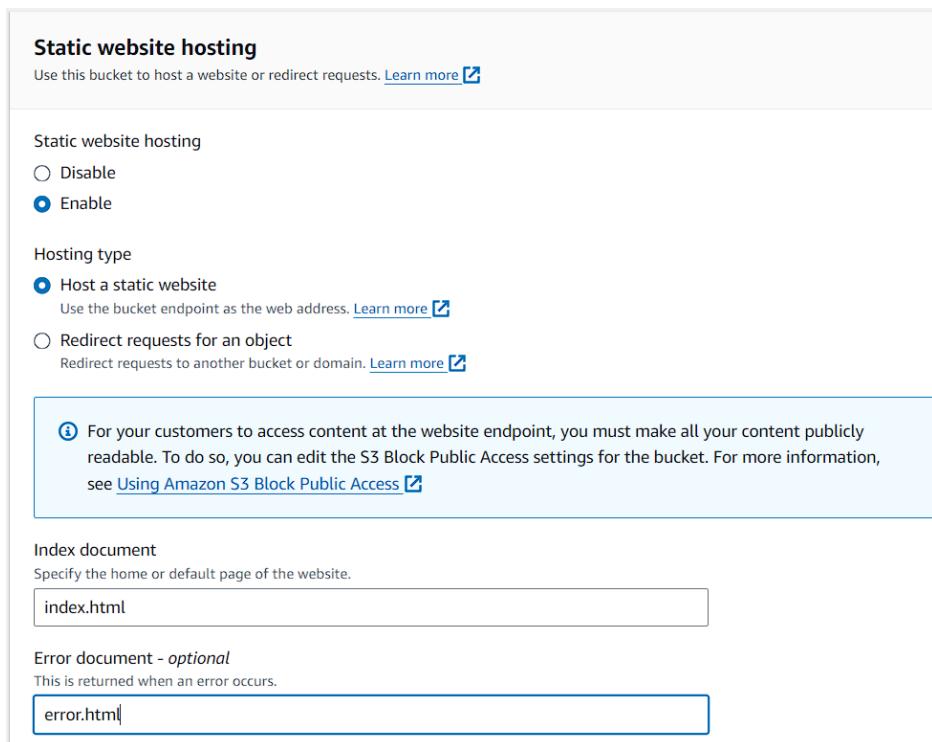
The screenshot shows a green success message at the top: "Successfully edited bucket policy." Below it is a JSON representation of the bucket policy:

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

13) Go to the edit static website hosting in the properties tab to use bucket to host websites.



14) Check the options as shown below, and add the names of the file.



The screenshot shows the "Static website hosting" configuration dialog. It includes sections for "Static website hosting" (with "Enable" selected), "Hosting type" (with "Host a static website" selected), and "Index document" (set to "index.html"). A note at the bottom explains that content must be publicly readable. There is also an "Error document - optional" field set to "error.html".

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

Static website hosting
 Disable
 Enable

Hosting type
 Host a static website
Use the bucket endpoint as the web address. [Learn more](#)
 Redirect requests for an object
Redirect requests to another bucket or domain. [Learn more](#)

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

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

- 15) Navigate to the Upload section and upload the documents with the name as mentioned in the previous section.

The screenshot shows the 'Upload' section of the AWS S3 console. At the top, there's a breadcrumb navigation: Amazon S3 > Buckets > brijeshkabucket > Upload. Below the navigation is a title 'Upload' with a 'Info' link. A sub-instruction says: 'Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more' with a link icon. A large dashed box area is labeled 'Drag and drop files and folders you want to upload here, or choose Add files or Add folder.' Below this is a table titled 'Files and folders (0)' with a note 'All files and folders in this table will be uploaded.' It includes a search bar 'Find by name' and buttons for 'Remove', 'Add files', and 'Add folder'. The table has columns for 'Name', 'Folder', and 'Type'. A message 'No files or folders' is displayed, followed by the sub-instruction 'You have not chosen any files or folders to upload.'

- 16) Uploaded files will be visible after successful upload.

The screenshot shows the 'Files and folders' list in the AWS S3 console. The title is 'Files and folders (2 Total, 906.0 B)'. A note says 'All files and folders in this table will be uploaded.' Below is a search bar 'Find by name' and buttons for 'Remove', 'Add files', and 'Add folder'. The table has columns for 'Name', 'Folder', and 'Type'. Two files are listed: 'error.html' and 'index.html', both of which are 'text/html' type files.

- 17) Get the link for the hosted website in the properties tab at the bottom.

The screenshot shows the 'Static website hosting' properties tab in the AWS S3 console. It includes sections for 'Static website hosting' (with a note 'Use this bucket to host a website or redirect requests. Learn more'), 'Hosting type' (set to 'Bucket hosting'), and 'Bucket website endpoint' (with a note 'When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket. Learn more'). The endpoint URL is listed as <http://brijeshkabucket.s3-website-us-east-1.amazonaws.com>.

18) The hosted website using AWS S3.

VESIT_Batch6_Koma... VESIT-Ganit App Int... Home IoE Theory- D208-V...

Hello

My first AWS Deployment

Made with ❤

19) To terminate the S3 bucket, first empty the bucket by selecting the files and clicking on Empty.

General purpose buckets (1) [Info](#) All AWS Regions

Buckets are containers for data stored in S3.

Find buckets by name

Name AWS Region IAM Access Analyzer Creation date

brijeshkabucket	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 7, 2024, 20:38:57 (UTC+05:30)
-----------------	---------------------------------	---	--------------------------------------

Successfully emptied bucket "brijeshkabucket"
View details below. If you want to delete this bucket, use the [delete bucket configuration](#).

Empty bucket: status

The details below are no longer available after you navigate away from this page.

Summary

Source s3://brijeshkabucket	Successfully deleted 2 objects, 906.0 B	Failed to delete 0 objects
--	--	-------------------------------

Failed to delete (0)

Find objects by name

Name	Prefix	Version ID	Type	Last modified	Size	Error
No failed object deletions						

20) Then navigate to the Delete bucket option and enter the name of the bucket and delete the bucket.

The screenshot shows the 'Delete bucket' confirmation dialog. At the top, the breadcrumb navigation is: Amazon S3 > Buckets > brijeshkabucket > Delete bucket. Below this is the title 'Delete bucket' with an 'Info' link. A warning box contains the following text:

⚠ • Deleting a bucket cannot be undone.
• Bucket names are unique. If you delete a bucket, another AWS user can use the name.
• If this bucket is used with a Multi-Region Access Point in an external account, initiate failover before deleting the bucket.
• If this bucket is used with an access point in an external account, the requests made through those access points will fail after you delete this bucket.
• This bucket is configured to host a static website. We recommend that you clean up the Route 53 hosted zone settings that are related to the bucket.

[Learn more](#)

The main area is titled 'Delete bucket "brijeshkabucket"?' with the instruction 'To confirm deletion, enter the name of the bucket in the text input field.' A text input field contains the value 'brijeshkabucket'. At the bottom right are 'Cancel' and 'Delete bucket' buttons.

21) After deleting the bucket, a message will appear.

The screenshot shows the Amazon S3 home page. A green header bar displays the message 'Successfully deleted bucket "brijeshkabucket"'. The main content area features the heading 'Amazon S3' and the subtext 'Store and retrieve any amount of data from anywhere'. It includes a brief description of what S3 is and a 'Create a bucket' button. On the left, there's a 'How it works' section with a video thumbnail and a 'Copy link' button. On the right, there's a 'Pricing' section with information about no minimum fees and a link to the Simple Monthly Calculator. The overall theme is dark with blue and white text.

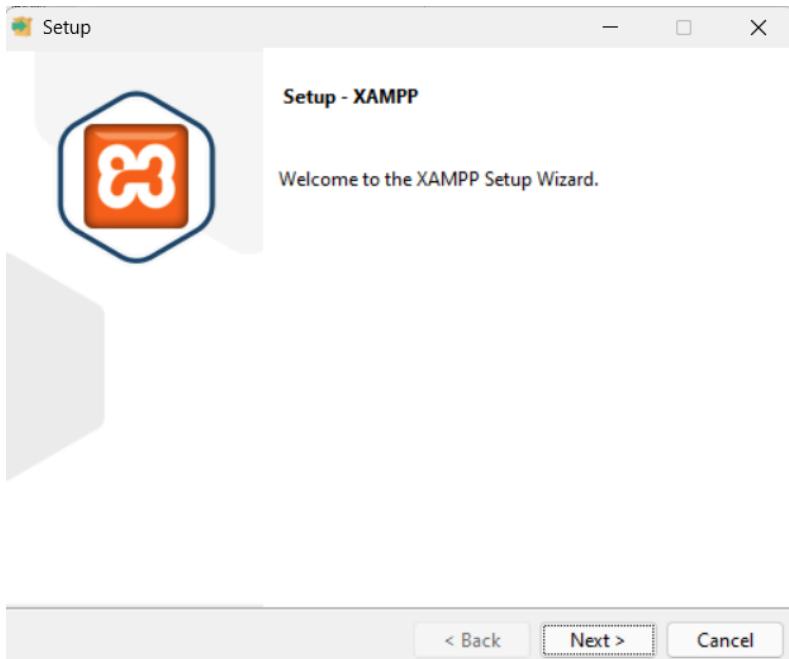
XAMPP Hosting

- 1) Search for XAMPP download and navigate to the xampp official website and click on download as per your system.

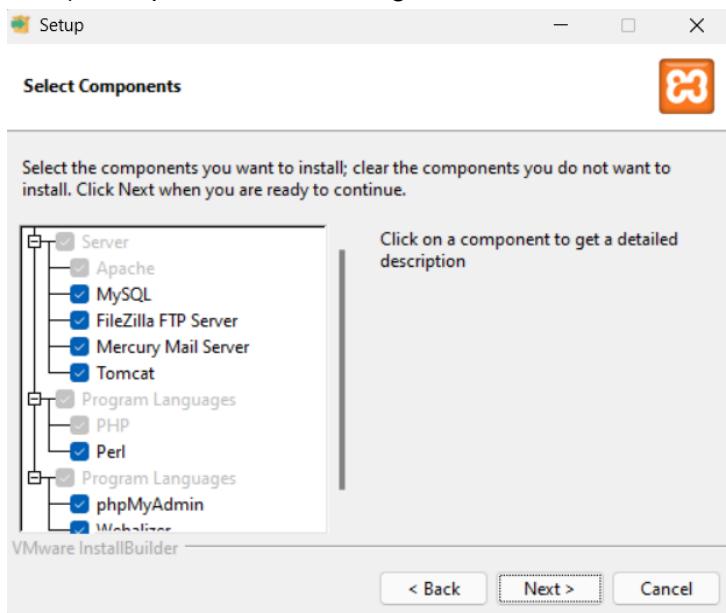
The screenshot shows the Apache Friends website's download section. At the top, there's a navigation bar with links for Apache Friends, Download, Hosting, Community, and About. A search bar and a language selector (EN) are also present. The main heading is "Download". Below it, a sub-section title "XAMPP" is displayed. A text block explains that XAMPP is an easy-to-install Apache distribution containing MariaDB, PHP, and Perl. It includes a table showing three versions of XAMPP for Windows: 8.0.30, 8.1.25, and 8.2.12, along with their checksums (md5 and sha1) and download links (64-bit). To the right, a "Documentation/FAQs" sidebar provides links to forums and Stack Overflow. At the bottom of the main content area, there are links for "Requirements" and "More Downloads". A note at the bottom states that Windows XP or 2003 are not supported and directs users to a link for compatible versions.

Version	Checksum	Size
8.0.30 / PHP 8.0.30	What's Included? md5 sha1	Download (64 bit) 144 Mb
8.1.25 / PHP 8.1.25	What's Included? md5 sha1	Download (64 bit) 148 Mb
8.2.12 / PHP 8.2.12	What's Included? md5 sha1	Download (64 bit) 149 Mb

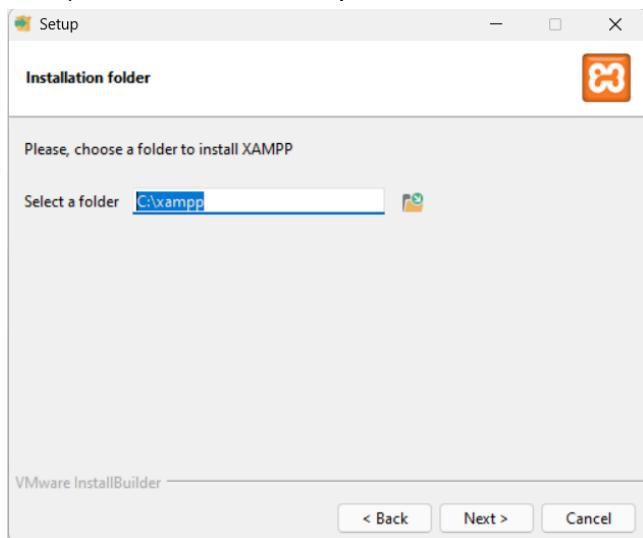
- 2) Xampp installer will be installed in the system.
- 3) Window asking to give permission will appear. Click pn 'Yes'.
- 4) A window will appear for setup. Click on 'Next'.



5) Keep the default settings and click on Next.



6) Choose the folder path.



7) Then the XAMPP installation will be done.

8) Locate the folder and then locate the 'htdocs' folder in the xampp folder.

anonymous	01-08-2024 21:37	File folder
apache	01-08-2024 21:38	File folder
cgi-bin	01-08-2024 21:45	File folder
contrib	01-08-2024 21:37	File folder
FileZillaFTP	01-08-2024 21:45	File folder
htdocs	01-08-2024 21:51	File folder
img	01-08-2024 21:37	File folder
install	01-08-2024 21:45	File folder

9) Create a test.php file in the htdocs folder and write php code.



```
<html>
<head>
    <title>First PHP Program</title>
</head>
<body>
    <center>
        <?php
echo "PHP website hosted using Xampp";
?>

        <font style="font-size:x-large;font-family:'Segoe UI', Tahoma, Geneva, Verdana, sans-serif"><h1>Hello</h1></font>
        <font color="blue" style="font-family:Georgia, 'Times New Roman', Times, serif;"><h2>My first Xampp Deployment</h2></font>
        <h3>Made with <font style="color: red;">&#10084;</font></h3>
    </center>
</center>
</body>
</html>

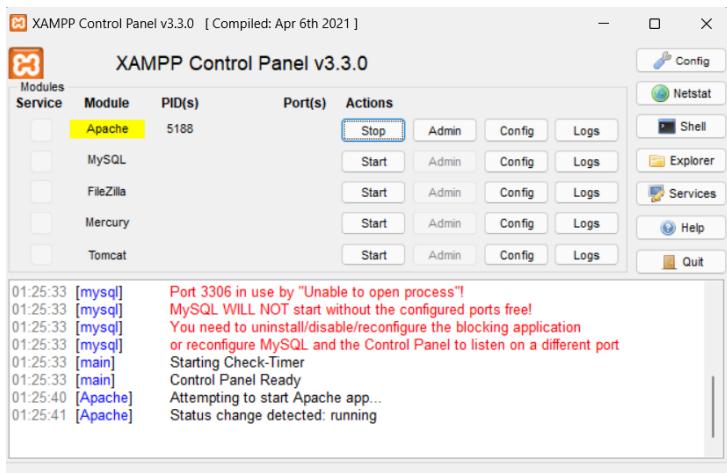
<!-- store in this file whatever to be displayed and then localhost/MicroNG/file_name --&gt;</pre>
```

Ln 1, Col 1 | 575 characters | 100% | Windows (CRL) | UTF-8

10) Now go to the xampp control panel in the xampp folder.

 xampp_start	30-03-2013 17:59	Application	116 KB
 xampp_stop	30-03-2013 17:59	Application	116 KB
 xampp-control	06-04-2021 17:08	Application	3,290 KB
 xampp-control	01-08-2024 21:45	Configuration setti...	1 KB

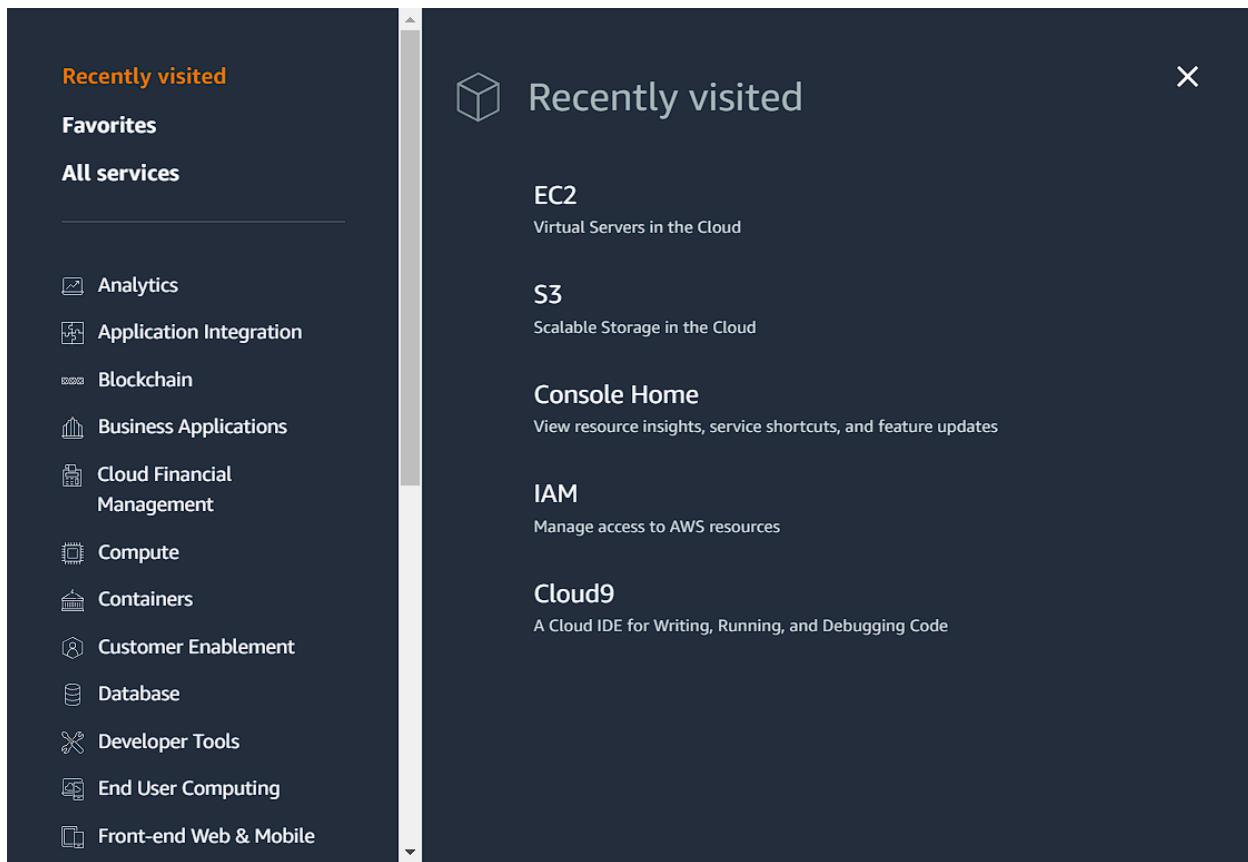
11) Start the Apache server.



12) After strating the service, got to “localhost/file_name”, then the output window will appear.



EC2 Instance



Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

[Launch instance](#)

[Migrate a server](#)

Note: Your instances will launch in the US East (N. Virginia) Region

Launch an instance Info

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

Name and tags Info

Name

[Add additional tags](#)

▼ Application and OS Images (Amazon Machine Image) Info

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



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

[Recents](#)[Quick Start](#)[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

ami-04a81a99f5ec58529 (64-bit (x86)) / ami-0c14ff330901e49ff (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Ubuntu Server 24.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Architecture

64-bit (x86)

AMI ID

ami-04a81a99f5ec58529

Verified provider

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

Instance type

t2.micro

Free tier eligible

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

All generations

[Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

▼ Key pair (login) [Info](#)

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

Key pair name - *required*

vockey

 [Create new key pair](#)

▼ Network settings [Info](#)

Edit

Network | [Info](#)

vpc-0531204c9e29f6332

Subnet | [Info](#)

No preference (Default subnet in any availability zone)

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

Enable

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

Firewall (security groups) | [Info](#)

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

Create security group

Select existing security group

We'll create a new security group called '**launch-wizard-2**' with the following rules:

Allow SSH traffic from
Helps you connect to your instance

Anywhere
0.0.0.0/0

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

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

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

X

▼ Configure storage [Info](#)

[Advanced](#)

1x GiB ▾ Root volume (Not encrypted)

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

[Add new volume](#)

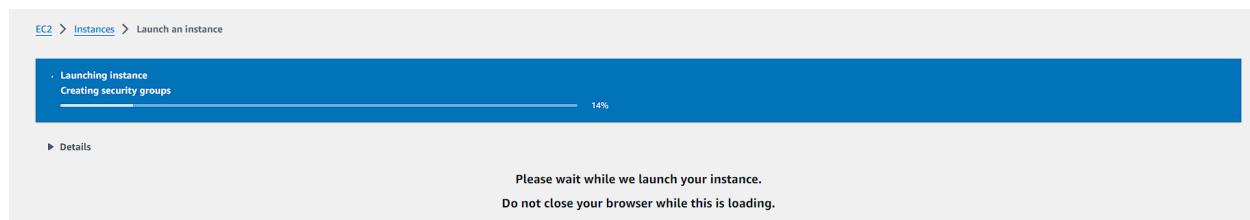
The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance

 Click refresh to view backup information



The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems

[Edit](#)

Connect to instance Info

Connect to your instance i-033e8bf30d3d5ed91 (AWS Test Server) using any of these options

[EC2 Instance Connect](#)

[Session Manager](#)

[SSH client](#)

[EC2 serial console](#)



Port 22 (SSH) is open to all IPv4 addresses

Port 22 (SSH) is currently open to all IPv4 addresses, indicated by **0.0.0.0/0** in the inbound rule in [your security group](#). For increased security, consider restricting access to only the EC2 Instance Connect service IP addresses for your Region: 18.206.107.24/29. [Learn more](#).

Instance ID

[i-033e8bf30d3d5ed91 \(AWS Test Server\)](#)

Connection Type

[Connect using EC2 Instance Connect](#)

Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.

[Connect using EC2 Instance Connect Endpoint](#)

Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address

[35.171.89.89](#)

Username

Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, `ubuntu`.

`ubuntu`



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



You have insufficient IAM permissions to connect to an instance using EC2 Instance Connect

To connect to an instance via EC2 Instance Connect, you must have an attached IAM policy that grants the following permissions:

- `ec2-instance-connect:SendSSHPublicKey`
- `ec2:DescribeInstances`

Consider restricting access to specific EC2 instances using `ec2:osuser` condition, or specific resource tag. Visit [IAM Console](#) to verify if you have above permissions.

For more information about IAM policy examples, see [Grant IAM permissions for EC2 Instance Connect](#).

[Cancel](#)

[Connect](#)

AWS | Services | Search [Alt+S] | N. Virginia | vclabs/user3394024=SHARMA_BRIJESH_RAJARAM @ 7878-8194-05

```

Usage of /: 22.7% of 6.71GB Users logged in: 0
Memory usage: 20% IPv4 address for enx0: 172.31.38.111
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 list of available updates is more than a week old.
To check for new updates run: sudo apt update

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-172-31-38-111:~$ i-033e8bf30d3d5ed91 (AWS Test Server)
PublicIPs: 35.171.89.89 PrivateIPs: 172.31.38.111

```

```

ubuntu@ip-172-31-38-111:~$ ping www.google.com
PING www.google.com (142.251.167.106) 56(84) bytes of data.
64 bytes from ww-in-f106.1e100.net (142.251.167.106): icmp_seq=1 ttl=58 time=2.22 ms
64 bytes from ww-in-f106.1e100.net (142.251.167.106): icmp_seq=2 ttl=58 time=2.24 ms
64 bytes from ww-in-f106.1e100.net (142.251.167.106): icmp_seq=3 ttl=58 time=2.26 ms
64 bytes from ww-in-f106.1e100.net (142.251.167.106): icmp_seq=4 ttl=58 time=2.38 ms
64 bytes from ww-in-f106.1e100.net (142.251.167.106): icmp_seq=5 ttl=58 time=2.31 ms
64 bytes from ww-in-f106.1e100.net (142.251.167.106): icmp_seq=6 ttl=58 time=2.26 ms
^C
--- www.google.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5007ms
rtt min/avg/max/mdev = 2.220/2.276/2.376/0.052 ms
ubuntu@ip-172-31-38-111:~$ vmstat
procs -----memory----- ---swap-- -----io---- -system-- -----cpu-----
  r   b   swpd   free   buff   cache   si   so   bi   bo   in   cs   us   sy   id   wa   st   gu
  2   0     0 504660 18052 288972     0     0   381   298   141     1   2   1 95   1   2   0
ubuntu@ip-172-31-38-111:~$ df
Filesystem      1K-blocks      Used Available Use% Mounted on
/dev/root        7034376  1609612    5408380  23% /
tmpfs            490212       0    490212   0% /dev/shm
tmpfs            196088      868    195220   1% /run
tmpfs              5120       0      5120   0% /run/lock
/dev/xvda16       901520    76972    761420  10% /boot
/dev/xvda15       106832     6246   100586   6% /boot/efi
tmpfs             98040       12    98028   1% /run/user/1000
ubuntu@ip-172-31-38-111:~$ mkdir test
ubuntu@ip-172-31-38-111:~$ touch e.txt
ubuntu@ip-172-31-38-111:~$ ls
e.txt  test
ubuntu@ip-172-31-38-111:~$ history
  1  ping
  2  ping www.google.com
  3  vmstat
  4  df
  5  mkdir test
  6  touch e.txt
  7  ls
  8  history

```

Instances (1) [Info](#)

Find Instance by attribute or tag (case-sensitive)

All states ▾

Instance ID = i-033e8bf30d3d5ed91 X Clear filters

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
AWS Test Server	i-033e8bf30d3d5ed91	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	ec2-35-

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

Find Instance by attribute or tag (case-sensitive)

All states ▾

Instance ID = i-033e8bf30d3d5ed91 X Clear filters

Name	Instance ID	Instance state	Instance type	Status	Alarm	Availability Zone	Public IP
AWS Test Server	i-033e8bf30d3d5ed91	Running	t2.micro	Green	View alarms +	us-east-1b	ec2-35-

Terminate instance?

⚠ On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated. Storage on any local drives will be lost.

Are you sure you want to terminate these instances?

Instance ID	Termination protection
i-033e8bf30d3d5ed91 (AWS Test Server)	<input checked="" type="checkbox"/> Disabled

To confirm that you want to terminate the instances, choose the terminate button below. Instances with termination protection enabled will not be terminated. Terminating the instance cannot be undone.

[Cancel](#) [Terminate](#)

Instances (1) [Info](#)

Find Instance by attribute or tag (case-sensitive)

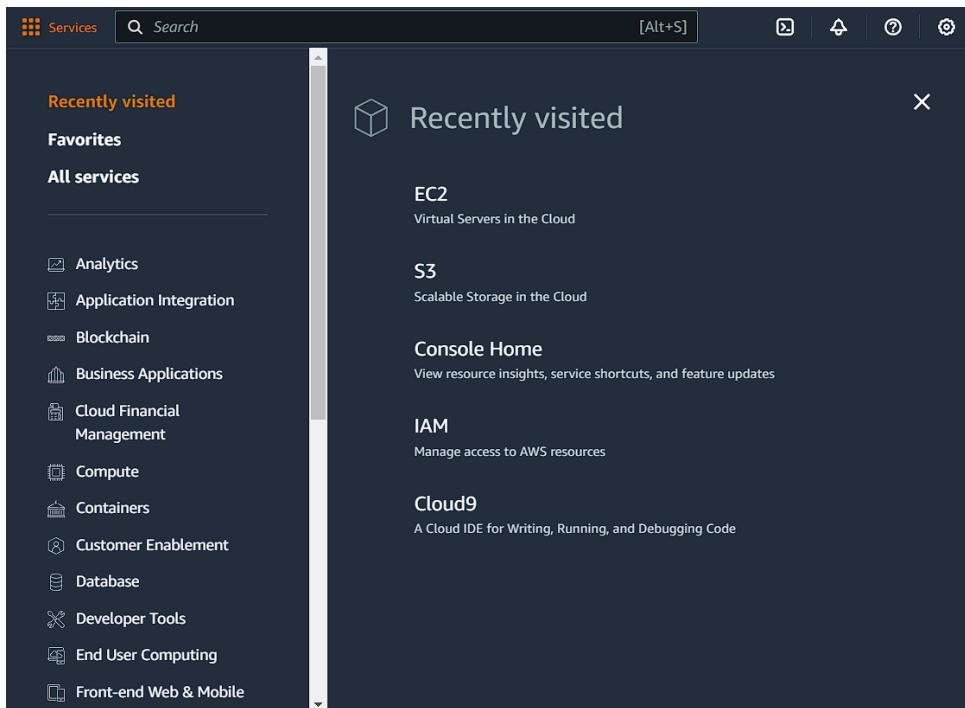
All states ▾

Instance ID = i-033e8bf30d3d5ed91 X Clear filters

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
AWS Test Server	i-033e8bf30d3d5ed91	Terminated	t2.micro	-	View alarms +	us-east-1b	-

Cloud 9 - IDE

- 1) Navigate to developer tools -> Cloud9 and start creating Cloud9 environment.



- 2) Click on Create Environment and start creating the environment

A screenshot of the AWS Cloud9 landing page. The page title is 'AWS Cloud9' and the subtitle is 'A cloud IDE for writing, running, and debugging code'. Below the title, there is a brief description of what AWS Cloud9 is and how it works. On the right side, there is a call-to-action box with the heading 'New AWS Cloud9 environment' and a large orange 'Create environment' button. At the bottom of the page, there are two sections: 'How it works' and 'Getting started'.

3) Name the environment and select new EC2 instance.

Details

Name

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

Description - *optional*

Limit 200 characters.

Environment type [Info](#)
Determines what the Cloud9 IDE will run on.

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

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

4) Keep the options default and proceed

New EC2 instance

Instance type [Info](#)
The memory and CPU of the EC2 instance that will be created for Cloud9 to run on.

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

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

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

Additional instance types
Explore additional instances to fit your need.

Platform [Info](#)
This will be installed on your EC2 instance. We recommend Amazon Linux 2023.

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

Network settings [Info](#)

Connection
How your environment is accessed.

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

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

► VPC settings [Info](#)

5) Environment created successfully.

The screenshot shows a dual-monitor setup. The top monitor displays the AWS Management Console with the Cloud9 service selected. A modal window titled "Creating D15C48" is open, indicating the process is taking several minutes. The main Cloud9 interface shows a single environment named "D15C48" listed in the "Environments" table. The bottom monitor displays the AWS Cloud9 IDE itself. The interface includes a navigation bar with File, Edit, Find, View, Go, Run, Tools, Window, Support, and Preview. The main area shows a "Welcome" screen with the title "AWS Cloud9" and the sub-header "Welcome to your development environment". Below this is a "Toolkit for AWS Cloud9" section, which provides a brief overview of the toolkit's features. At the bottom of the IDE, there is a terminal window showing a bash session with the command "vocabs:~/environment \$". The taskbar at the bottom of both monitors shows various application icons and system status indicators like battery level and network connection.

6) Create user using the IAM.

The screenshot shows the AWS IAM 'Users' page. At the top, there is a header with 'Users (0) info'. Below it, a note says 'An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.' A search bar labeled 'Search' is followed by navigation icons: back, forward, and refresh. A large orange button labeled 'Create user' is prominently displayed. Below these, there is a table header with columns: 'User name', 'Path', 'Group', 'Last activity', 'MFA', 'Password age', and 'Console last sign-in'. The main content area below the table header displays the message 'No resources to display'.

7) Add the username

The screenshot shows the 'Specify user details' step of the IAM user creation wizard. The title is 'Specify user details'. Under 'User details', the 'User name' field is filled with 'Brij@aws'. A note below it states: 'The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = . @ _ - (hyphen)' and includes a link to 'Learn more'. There is an optional checkbox 'Provide user access to the AWS Management Console - optional' with a note: 'If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center.' A blue callout box contains the note: 'If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keypairs, you can generate them after you create this IAM user. [Learn more](#)'.

8) Add the remaining user details and provide access to the AWS Management Console

The screenshot shows the 'Add remaining user details' step of the IAM user creation wizard. The title is 'User details'. The 'User name' field is filled with 'Brij@aws'. The 'Provide user access to the AWS Management Console - optional' checkbox is checked. The 'Console password' section shows 'Autogenerated password' selected, with a note: 'You can view the password after you create the user.' and 'Custom password' selected with a note: 'Enter a custom password for the user.' A password field contains '*****'. A note below it specifies: 'Must be at least 8 characters long' and 'Must include at least three of the following mix of character types: uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), and symbols ! @ # \$ % ^ & * () _ + - (hyphen) = [] { } ; , . / ? '.

Checkboxes for 'Show password' and 'Users must create a new password at next sign-in - Recommended' are present. A note for the latter states: 'Users automatically get the IAMUserChangePassword policy to allow them to change their own password.'

A blue callout box contains the note: 'If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keypairs, you can generate them after you create this IAM user. [Learn more](#)'.

9) User created successfully and can be added to user groups.

Set permissions

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

Permissions options

Add user to group

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

Copy permissions

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

Attach policies directly

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



Get started with groups

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

[Create group](#)

► Set permissions boundary - *optional*

[Cancel](#)

[Previous](#)

[Next](#)

10) User credentials can be downloaded.

Retrieve password

You can view and download the user's password below or email users instructions for signing in to the AWS Management Console. This is the only time you can view and download this password.

Console sign-in details

[Email sign-in instructions](#)

Console sign-in URL

<https://017820672175.signin.aws.amazon.com/console>

User name

Brij@aws

Console password

***** [Show](#)

[Cancel](#)

[Download .csv file](#)

[Return to users list](#)

11) Add user to group if group exists else create a new group.

AWSGroup1 user group created.

X

[IAM](#) > [Users](#) > Create user

Step 1

[Specify user details](#)

Step 2

[Set permissions](#)

Step 3

[Review and create](#)

Step 4

[Retrieve password](#)

Set permissions

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

Permissions options

Add user to group

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

Copy permissions

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

Attach policies directly

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

12) Write the user group name and proceed.

The screenshot shows the 'User groups (1)' page in the AWS IAM console. At the top right are 'Create group' and 'Cancel' buttons. Below is a search bar and a table with columns: Group name, Users, Attached policies, and Created. A single row is shown for 'AWSGroup1'. At the bottom is a note about setting a permissions boundary and navigation buttons 'Previous' and 'Next'.

13) Click on *group_name*.

The screenshot shows the 'User groups (1) Info' page. It displays a table with columns: Group name, Users, Permissions, and Creation time. The 'AWSGroup1' row shows 0 users, 'Not defined' permissions, and was created 4 minutes ago. Buttons for 'Delete' and 'Create group' are at the top right.

14) Go to Add permissions and click on Add Permissions

The screenshot shows the 'AWSGroup1 Info' page. The 'Permissions' tab is selected. It includes sections for 'Summary' (with ARN), 'Users', 'Permissions', and 'Access Advisor'. Under 'Permissions', there's a table for 'Permissions policies (0) Info' with columns: Policy name, Type, and Attached entities. A note says 'No resources to display'.

15) On attach policies, select AWSCloud9EnvironmentMember and click on Attach policies.

The screenshot shows the 'Attach permission policies' dialog for the user group 'AWSGroup1'. At the top, it says 'Attach permission policies to AWSGroup1'. Below that is a section titled 'Current permissions policies (0)'. Underneath is a heading 'Other permission policies (1/945)' with a note: 'You can attach up to 10 managed policies to this user group. All of the users in this group inherit the attached permissions.' A search bar and filter dropdown are present. A table lists four policies:

Policy name	Type	Used as	Description
AWSCloud9Administrator	AWS managed	None	Provides administrator access to AWS Clo...
<input checked="" type="checkbox"/> AWSCloud9EnvironmentMember	AWS managed	None	Provides the ability to be invited into AW...
AWSCloud9SSMInstanceProfile	AWS managed	None	This policy will be used to attach a role o...
AWSCloud9User	AWS managed	None	Provides permission to create AWS Cloud...

At the bottom right are 'Cancel' and 'Attach policies' buttons, with 'Attach policies' being highlighted.

16) User group is created successfully.

The screenshot shows the 'AWSGroup1' details page. At the top, a green bar indicates 'Policies attached to this user group.' The main area shows the user group summary: 'User group name: AWSGroup1', 'Creation time: August 09, 2024, 00:07 (UTC+05:30)', and 'ARN: arn:aws:iam::017820672175:group/AWSGroup1'. Below this, tabs for 'Users', 'Permissions' (which is selected), and 'Access Advisor' are visible. The 'Permissions' tab displays the attached policy:

Permissions policies (1)

You can attach up to 10 managed policies.

Policy name	Type	Attached entities
<input type="checkbox"/> AWSCloud9EnvironmentMember	AWS managed	1

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) Open the aws console and then search Elastic Beanstalk

The screenshot shows the Amazon Elastic Beanstalk landing page. At the top, there's a navigation bar with the AWS logo, a search bar, and account information for 'N. Virginia' and 'Brijesh'. Below the navigation, the page title 'Amazon Elastic Beanstalk' is displayed with the subtitle 'End-to-end web application management.' A 'Get started' section contains a button labeled 'Create application'. Another section titled 'Pricing' explains that there's no additional charge for Elastic Beanstalk. On the left, there are sections for 'Get started', 'Benefits and features' (listing 'Easy to get started' and 'Complete resource control'), and links for 'Getting started' and 'Launch a web application'. The main content area has a dark background with white text.

2) Click on create application and configure the environment by adding your application name.

The screenshot shows the 'Configure environment' step of the application creation wizard. On the left, a sidebar lists steps: Step 1 (Configure environment), Step 2 (Configure service access), Step 3 - optional (Set up networking, database, and tags), Step 4 - optional (Configure instance traffic and scaling), Step 5 - optional (Configure updates, monitoring, and logging), and Step 6 (Review). The main content area is titled 'Configure environment' with a sub-section 'Environment tier'. It shows two options: 'Web server environment' (selected) and 'Worker environment'. The 'Web server environment' option is described as running a website, web application, or web API that serves HTTP requests. The 'Worker environment' option is described as running a worker application that processes long-running workloads on demand or performs tasks on a schedule. Below this is another sub-section 'Application information' where the 'Application name' is set to 'myawsbean'. There's also a section for 'Application tags (optional)'.

3) Select the environment as PHP and other options as default and click on next.

Environment information [Info](#)

Choose the name, subdomain and description for your environment. These cannot be changed later.

Environment name

Must be from 4 to 40 characters in length. The name can contain only letters, numbers, and hyphens. It can't start or end with a hyphen. This name must be unique within a region in your account.

Domain
 .us-east-1.elasticbeanstalk.com [Check availability](#)

Environment description

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

Platform branch

Platform version

Application code [Info](#)

Sample application
 Existing version
Application versions that you have uploaded.
 Upload your code
Upload a source bundle from your computer or copy one from Amazon S3.

Presets [Info](#)

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

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

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

- 4) After clicking on Next for creating Elastic Beanstalk, we need key-pair that will be required for deployment. Go to EC2 Instance and click on Key Pairs.

The screenshot shows the AWS EC2 Instances dashboard. On the left, there's a summary of resources: Instances (running) 0, Auto Scaling Groups 0, Dedicated Hosts 0, Elastic IPs 0, Instances 0, Key pairs 0, Load balancers 0, Placement groups 0, Security groups 1, Snapshots 0, and Volumes 0. Below this is a 'Launch instance' section with a 'Launch instance' button and a 'Migrate a server' link. A note says instances will launch in the US East (N. Virginia) Region. There are sections for 'Instance alarms' (0 in alarm, 0 OK, 0 insufficient data), 'Instances in alarm', and 'Scheduled events'. On the right, the 'Service health' section shows 'AWS Health Dashboard' and indicates the service is operating normally. Below it is the 'EC2 Free Tier' info, which shows 0 offers in use, end-of-month forecast, and exceeds free tier. It also links to view global EC2 resources and all AWS Free Tier offers. Further down are 'Account attributes' (Default VPC vpc-07b4b6571f0c46e01, Settings for Data protection and security, Zones, EC2 Serial Console, Default credit specification, EC2 console preferences), and an 'Additional information' section.

- 5) Then click on Create key pair

The screenshot shows the AWS Key Pairs list page. At the top, there's a search bar for 'Find Key Pair by attribute or tag'. Below it is a table header with columns: Name, Type, Created, Fingerprint, and ID. The table body is empty, showing 'No key pairs to display'. At the bottom right are 'Actions' and a 'Create key pair' button.

- 6) Input the name of the key-pair and select pem as file format and click on Create key pair.

The screenshot shows the 'Create key pair' wizard. Step 1: Key pair. It asks for a name ('the_key') and key pair type ('RSA'). It also shows private key file format options: '.pem' (selected) and '.ppk'. There are optional tags and a note about adding up to 50 more tags. At the bottom are 'Cancel' and 'Create key pair' buttons.

- 7) After creating key pair, open new tab and go to IAM to create a role that will be used to build Codepipeline. Click on Create role.

The screenshot shows the AWS IAM Roles page. At the top, there is a search bar and a header with the number of roles (2), an 'Info' link, and buttons for 'Delete' and 'Create role'. Below the header, there is a table with columns for 'Role name', 'Trusted entities', and 'Last activity'. Two roles are listed:

Role name	Trusted entities	Last activity
AWSServiceRoleForSupport	AWS Service: support (Service-Linker)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service)	-

- 8) Select AWS service as Trusted Entity type and EC2 as service.

The screenshot shows the 'Create New Role' wizard, Step 1: Set the Trusted entity type. It has two main sections: 'Trusted entity type' and 'Use case'.

Trusted entity type:

- AWS service: Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account: Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- Web identity: Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- SAML 2.0 federation: Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- Custom trust policy: Create a custom trust policy to enable others to perform actions in this account.

Use case:

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case:

EC2

Choose a use case for the specified service.

Use case:

- EC2: Allows EC2 instances to call AWS services on your behalf.
- EC2 Role for AWS Systems Manager: Allows EC2 instances to call AWS services like CloudWatch and Systems Manager on your behalf.
- EC2 Spot Fleet Role: Allows EC2 Spot Fleet to request and terminate Spot Instances on your behalf.
- EC2 - Spot Fleet Auto Scaling: Allows Auto Scaling to access and update EC2 spot fleets on your behalf.
- EC2 - Spot Fleet Tagging: Allows EC2 to launch spot instances and attach tags to the launched instances on your behalf.

- 9) Choose AdministratorAccess-AWSElasticBeanstalk as Policy and click on Next.

The screenshot shows the 'Add permissions' step of the wizard. It displays a list of available policies under 'Permissions policies (1/946)'.

Permissions policies (1/946)

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

Filter by Type: All types

Policy name	Type	Description
<input type="checkbox"/> AdministratorAccess	AWS managed - job function	Provides full access to AWS services an...
<input type="checkbox"/> AdministratorAccess-Amplify	AWS managed	Grants account administrative permis...
<input checked="" type="checkbox"/> AdministratorAccess-AWSElasticBeanstalk	AWS managed	Grants account administrative permis...
<input type="checkbox"/> AlexaForBusinessDeviceSetup	AWS managed	Provide device setup access to AlexaFo...
<input type="checkbox"/> AlexaForBusinessFullAccess	AWS managed	Grants full access to AlexaForBusiness ...
<input type="checkbox"/> AlexaForBusinessGatewayExecution	AWS managed	Provide gateway execution access to A...
<input type="checkbox"/> AlexaForBusinessLifesizeDelegatedAccessPolicy	AWS managed	Provide access to Lifesize AVS devices
<input type="checkbox"/> AlexaForBusinessPolyDelegatedAccessPolicy	AWS managed	Provide access to Poly AVS devices

10) Name the role and keep other as default.

The screenshot shows the 'Role details' section of the AWS IAM console. It includes fields for 'Role name' (set to 'new-user'), 'Description' (set to 'Allows EC2 instances to call AWS services on your behalf.'), and a 'Trust policy' code editor containing a JSON-based policy definition:

```

1- {
2-     "Version": "2012-10-17",
3-     "Statement": [
4-         {
5-             "Effect": "Allow",
6-             "Action": [
7-                 "sts:AssumeRole"
8-             ],
9-             "Principal": [
10-                 "Service": [
11-                     "ec2.amazonaws.com"
12-                 ]
13-             ]
14-         }
15-     ]
16- }

```

11) The role is created successfully.

The screenshot shows the 'Roles (3) Info' section of the AWS IAM console. It lists three roles: 'AWSServiceRoleForSupport', 'AWSServiceRoleForTrustedAdvisor', and 'new-user'. The 'new-user' role was just created and is listed under the 'Last activity' column.

12) Now move to the tab where Elastic Beanstalk was opened and from the drop down menu select the newly created key pair and instance profile. Now let everything be default.

The screenshot shows the 'Configure service access' step of the AWS Elastic Beanstalk setup wizard. It includes sections for 'Service access', 'Service role', 'EC2 key pair', and 'EC2 instance profile'. The 'Service role' section shows 'Create and use new service role' selected, with a service role name 'aws-elasticbeanstalk-service-role'. The 'EC2 key pair' section shows 'new-key' selected. The 'EC2 instance profile' section shows 'new-user' selected. Navigation buttons at the bottom include 'Cancel', 'Skip to review', 'Previous', and a highlighted 'Next' button.

Set up networking, database, and tags - optional Info

Virtual Private Cloud (VPC)

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

-

[Create custom VPC](#)

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

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

Activated

Instance subnets

Filter instance subnets

	Availability Zone	Subnet	CIDR	Name
No instance subnets No instance subnets to display				

13) Review the changes and click on Create.

Review Info

Step 1: Configure environment [Edit](#)

Environment information

Environment tier	Application name
Web server environment	myawsbean
Environment name	Application code
Myawsbean-env	Sample application
Platform	
arn:aws:elasticbeanstalk:us-east-1::platform/PHP 8.3 running on 64bit Amazon Linux 2023/4.3.2	

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

Service access Info

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

Service role	EC2 key pair	EC2 instance profile
arn:aws:iam::017820672175:role/service-role/aws-elasticbeanstalk-service-role	the_key	new-user

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

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

The screenshot shows the AWS Elastic Beanstalk Environment Overview page for 'Myawsbean-env'. At the top, a green banner indicates 'Environment successfully launched.' Below the banner, the navigation path is 'Elastic Beanstalk > Environments > Myawsbean-env'. The main content is divided into two sections: 'Environment overview' and 'Platform'. The 'Environment overview' section shows 'Health' as 'Ok', 'Domain' as 'Myawsbean-env.eba-sp3sdamg.us-east-1.elasticbeanstalk.com', and 'Application name' as 'myawsbean'. The 'Platform' section shows 'Platform' as 'PHP 8.3 running on 64bit Amazon Linux 2023/4.3.2', 'Running version' as '—', and 'Platform state' as 'Supported'.

15) Now, we need to make a CodePipeline. Go to CodePipeline and click on Create Pipeline.

The screenshot shows the AWS CodePipeline Pipelines page. The navigation path is 'Developer Tools > CodePipeline > Pipelines'. A modal window titled 'Introducing the new V2 pipeline type with improved release safety, pipeline triggers, parameterized pipelines, and a new billing model. Learn more' is open. The main table has a single row with the following columns: 'Name', 'Latest execution status', 'Latest source revisions', 'Latest execution started', and 'Most recent executions'. The table displays 'No results' and the message 'There are no results to display.'

16) Name the pipeline and select the service role as below and click on Next.

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](#)

- 17) In the source stage select Github v2 as the provider and then connect your github connect so that the pipeline can access the forked source code. Name the connection.

[Developer Tools](#) > [Connections](#) > [Create connection](#)

Create a connection Info

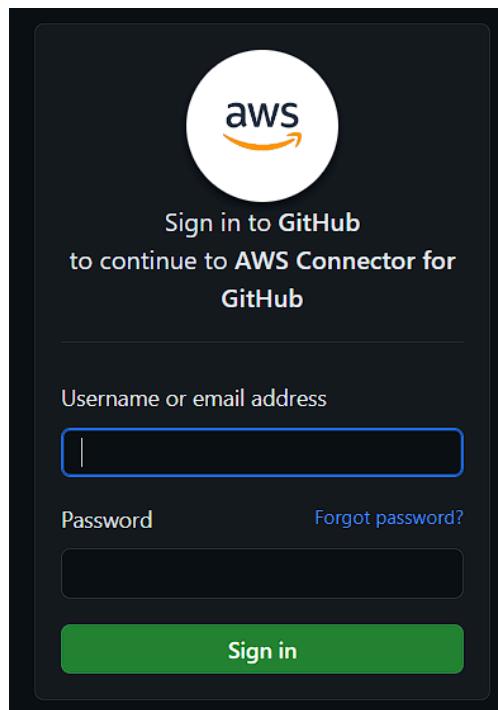
Create GitHub App connection Info

Connection name

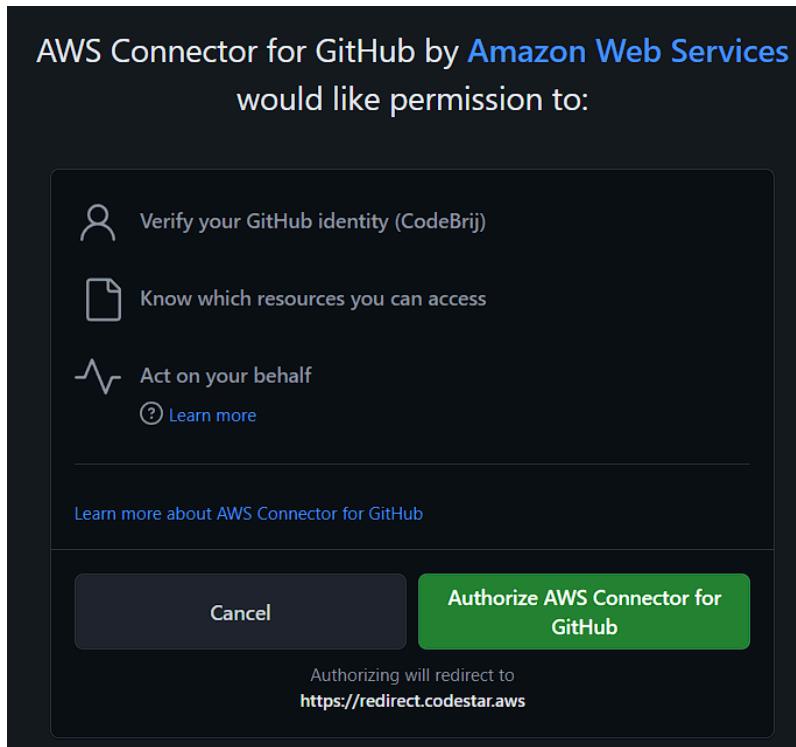
▶ Tags - *optional*

Connect to GitHub

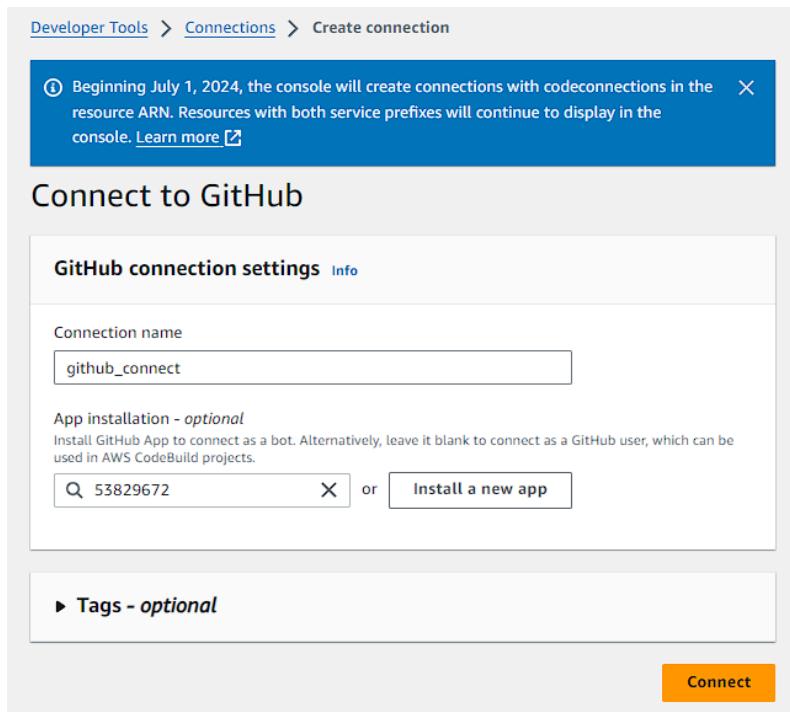
- 18) Signin to GitHub to connect with AWS.



19) Authorize AWS Connector for GitHub.



20) We need to install the GitHub connector.



21) Now, select the repository and the branch to be deployed.

Source

Source provider
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.

GitHub (Version 2) ▾

New GitHub version 2 (app-based) action
To add a GitHub version 2 action in CodePipeline, you create a connection, which uses GitHub Apps to access your repository. Use the options below to choose an existing connection or create a new one. [Learn more](#)

Connection
Choose an existing connection that you have already configured, or create a new one and then return to this task.

Q arn:aws:codeconnections:us-east-1:017820672175:connection/ca9502e9-40... X or [Connect to GitHub](#)

Ready to connect
Your GitHub connection is ready for use.

Repository name
Choose a repository in your GitHub account.

Q CodeBrij/aws-codepipeline-s3-codedeploy-linux-2.0 X

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

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

Q master X

Output artifact format
Choose the output artifact format.

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

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

22) Select No filter in Trigger.

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.

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

23) In deploy stage add application name as environment name. Then review the settings and click on Create pipeline.

Add deploy stage [Info](#)

Step 4 of 5

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

Deploy

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

AWS Elastic Beanstalk

Region
US East (N. Virginia)

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

No more than 100 characters

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

Q myawsbean X

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

Q Myawsbean-env X

Configure automatic rollback on stage failure

24) The pipeline is ready and the provided repo is deployed successfully.

Success
Congratulations! The pipeline mypipeline has been created.

Create a notification rule for this pipeline X

Developer Tools > CodePipeline > Pipelines > mypipeline

mypipeline

Pipeline type: V2 Execution mode: QUEUED

Source In progress
Pipeline execution ID: #07689e-bbc4-4c49-a188-366bca8f99ef

Source
 In progress - Just now
View details

Deploy Didn't run

Start rollback

Deploy
AWS Elastic Beanstalk
 Didn't Run
No executions yet

25) Go to Elastic Beanstalk and from Domain open the hosted site.

The screenshot shows the AWS Elastic Beanstalk console with the path 'Elastic Beanstalk > Applications > myawsbean'. The main view displays the 'Application myawsbean environments (1)' section. A single environment named 'Myawsbean-env' is listed, showing 'Ok' status, created on August 15, 2024 at 22:12:50, running version 'Myawsbean-env.eba-sp3sdam...', platform 'code-pipeline-172374181...', and PHP 8.3 running on 64bit. The 'Actions' dropdown menu includes options like 'Create new environment', 'Edit environment', 'Delete environment', and 'Switch environment'.

26) Hosted site from the github repository.

The screenshot shows a web browser window with the URL 'myawsbean-env.eba-sp3sdam.us-east-1.elasticbeanstalk.com'. The page has a green header with the text 'Congratulations!'. Below it, a message states: '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.' At the bottom, there is a link: 'For next steps, read the AWS CodePipeline Documentation. Incedege 2020'.

27) Make some changes in the index.html and reload.

The screenshot shows the 'Commit changes' dialog box. It contains a 'Commit message' field with the text 'Update index.html'. Below it is an 'Extended description' field with the placeholder 'Add an optional extended description..'. At the bottom, there are two radio button options: 'Commit directly to the master branch' (selected) and 'Create a new branch for this commit and start a pull request [Learn more about pull requests](#)'. There are also 'Cancel' and 'Commit changes' buttons.

The screenshot shows a web browser window with the same URL as the previous screenshot. The page now displays a personalized message: 'Congratulations Brijesh Sharma!' followed by 'Roll No. 48'. Below this, the same deployment confirmation message is shown: '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.' At the bottom, there is a link: 'For next steps, read the AWS CodePipeline Documentation. Incedege 2020'.

Experiment - 3

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

- 1) Create 3 EC-2 instances with all running on Amazon Linux as OS with inbound SSH allowed.
- 2) To efficiently run kubernetes cluster, select instance type of at least t2.medium as kubernetes recommends at least 2 vCPU to run smoothly

Launch an instance

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

Name and tags

Name: worker

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 are looking for below.

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

Recent AMIs: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux Enterprise Server

Browse more AMIs

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI (ami-0182f573c66fb98:85) (64-bit (x86), uefi-preferred) / ami-0b947c5d5516fa06e (64-bit (Arm), uefi)

Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

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

Key pair (login)

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

Key pair name - required: brijesh

Create new key pair

Network settings

Network: vpc-051204c9e29f6352

Subnet: No preference (Default subnet in any availability zone)

Auto-assign public IP: Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups): Create security group

We'll create a new security group called 'launch-wizard-8' with the following rules:

Allow SSH traffic from: Anywhere

Helps you connect to your instance: 0.0.0.0/0

Summary

Number of instances: 3

Software Image (AMI): Amazon Linux 2023 AMI 2023.5.2...read more

Virtual server type (instance type): t2.medium

Firewall (security group): New security group

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

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

Cancel Launch instance Review commands

3) Three instance are ready - master, worker1, and worker2.

Instances (1/9) Info		Last updated less than a minute ago	C	Connect	Instance state ▾	Actions ▾	Launch instances ▾	?
Find Instance by attribute or tag (case-sensitive)								All states ▾
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP	
master	i-076c3bb1d7c953b83	Running ⓘ ⓘ	t2.medium	Initializing	View alarms +	us-east-1a	ec2-34-182-212-92-39.compute-1.amazonaws.com	
worker1	i-0e6c44ebfb4ca3afa	Running ⓘ ⓘ	t2.medium	Initializing	View alarms +	us-east-1a	ec2-18-234-44-235.compute-1.amazonaws.com	
worker2	i-0d83b162ba0948d34	Running ⓘ ⓘ	t2.medium	Initializing	View alarms +	us-east-1a	ec2-18-234-44-235.compute-1.amazonaws.com	

4) Connect the instances to the local terminal using the SSH client.

```

[ec2-user@ip-172-31-18-4 ~]$ ssh -i "brijesh.pem" ec2-user@ec2-18-212-92-39.compute-1.amazonaws.com
[ec2-user@ip-172-31-26-194 ~]$ ssh -i "brijesh.pem" ec2-user@ec2-18-212-92-39.compute-1.amazonaws.com
[ec2-user@ip-172-31-25-225 ~]$ ssh -i "brijesh.pem" ec2-user@ec2-18-234-44-235.compute-1.amazonaws.com

```

5) Run the following commands on all the machines.

Install Docker

a) `sudo yum install docker -y`

```

[ec2-user@ip-172-31-25-225 ~]$ sudo yum install docker -y
Last metadata expiration check: 0:06:08 ago on Thu Sep 12 13:45:25 2024.
Dependencies resolved.
=====
Package           Architecture Version       Repository   Size
=====
Installing:
docker            x86_64      25.0.6-1.amzn2023.0.2   amazonlinux  44 M
Installing dependencies:
containerd         x86_64      1.7.20-1.amzn2023.0.1   amazonlinux  35 M
iptables-libs     x86_64      1.8.8-3.amzn2023.0.2   amazonlinux  401 k
iptables-nft      x86_64      1.8.8-3.amzn2023.0.2   amazonlinux  183 k
libcgroup          x86_64      3.0-1.amzn2023.0.1    amazonlinux  75 k
libnetfilter_conntrack x86_64      1.0.8-2.amzn2023.0.2   amazonlinux  58 k
libnfnetlink       x86_64      1.0.1-19.amzn2023.0.2  amazonlinux  30 k
libnftnl           x86_64      1.2.2-2.amzn2023.0.2  amazonlinux  84 k
pigz              x86_64      2.5-1.amzn2023.0.3    amazonlinux  83 k
runc              x86_64      1.1.13-1.amzn2023.0.1  amazonlinux  3.2 M
=====
Transaction Summary
=====
Install 10 Packages

Total download size: 84 M
Installed size: 317 M
Downloading Packages:
(1/10): iptables-libs-1.8.8-3.amzn2023.0.2.x86_64.rpm          3.6 MB/s | 401 kB     00:00

```

b) Then, configure cgroup in a daemon.json file by using following commands. This allows kubernetes to manage host more efficiently -

- `cd /etc/docker`

Run the scripts below -

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

```
[ec2-user@ip-172-31-18-9 ~]$ cd /etc/docker
[ec2-user@ip-172-31-18-9 docker]$ █
```

```
[ec2-user@ip-172-31-18-9 docker]$ cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opt": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
EOF
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opt": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
[ec2-user@ip-172-31-18-9 docker]$ █
```

c) After configuring restart docker service service :

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

```
[ec2-user@ip-172-31-18-9 docker]$ sudo systemctl enable docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
[ec2-user@ip-172-31-18-9 docker]$ █
```

```
[ec2-user@ip-172-31-18-9 docker]$ sudo systemctl daemon-reload
sudo systemctl restart docker
docker -v
Docker version 25.0.5, build 5dc9bcc
[ec2-user@ip-172-31-18-9 docker]$ █
```

Install Kubernetes

- a) SELinux needs to be disabled before configuring kubelet to avoid interference with kubernetes api server
- sudo setenforce 0
 - sudo sed -i 's/^SELINUX=enforcing\$/SELINUX=permissive/' /etc/selinux/config
 - Add kubernetes repository (paste in terminal)

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

```
[ec2-user@ip-172-31-18-9 docker]$ sudo setenforce 0
[ec2-user@ip-172-31-18-9 docker]$ sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
[ec2-user@ip-172-31-18-9 docker]$ cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/repo/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
EOF
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/repo/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
[ec2-user@ip-172-31-18-9 docker]$
```

- b) sudo yum update

```
[ec2-user@ip-172-31-25-225 docker]$ sudo yum update
Kubernetes
Dependencies resolved.
Nothing to do.
Complete!
```

c) sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes

```
[ec2-user@ip-172-31-18-9 docker]$ sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Last metadata expiration check: 0:00:54 ago on Thu Sep 12 14:50:28 2024.
Dependencies resolved.
=====
Package           Architecture      Version       Repository   Size
=====
Installing:
kubeadm          x86_64          1.30.5-150500.1.1   kubernetes   10 M
kubectl          x86_64          1.30.5-150500.1.1   kubernetes   10 M
kubelet           x86_64          1.30.5-150500.1.1   kubernetes   17 M
=====
Installing dependencies:
  conntrack-tools    x86_64          1.4.6-2.amzn2023.0.2   amazonlinux  208 k
  cri-tools          x86_64          1.30.1-150500.1.1   kubernetes   8.6 M
  kubernetes-cni     x86_64          1.4.0-150500.1.1   kubernetes   6.7 M
  libnetfilter_cthelper x86_64          1.0.0-21.amzn2023.0.2   amazonlinux  24 k
  libnetfilter_cttimeout x86_64          1.0.0-19.amzn2023.0.2   amazonlinux  24 k
  libnetfilter_queue  x86_64          1.0.5-2.amzn2023.0.2   amazonlinux  30 k
=====
Transaction Summary
=====
Install 9 Packages

Total download size: 53 M
Installed size: 292 M
Downloading Packages:
(1/9): conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64.rpm           3.1 MB/s | 208 kB     00:00
(2/9): libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64.rpm        332 kB/s | 24 kB    00:00
```

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

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

```
[ec2-user@ip-172-31-25-225 docker]$ sudo swapoff -a
[ec2-user@ip-172-31-25-225 docker]$ echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf
net.bridge.bridge-nf-call-iptables=1
[ec2-user@ip-172-31-25-225 docker]$ sudo sysctl -p
net.bridge.bridge-nf-call-iptables = 1
[ec2-user@ip-172-31-25-225 docker]$
```

6) To perform only on Master machine

a) Initialize kubernetes by typing below command

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

```
net.bridge.bridge-nf-call-iptables=1
[ec2-user@ip-172-31-18-9 docker]$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=all
I0912 14:55:34.563710 30027 version.go:256] remote version is much newer: v1.31.0; falling back to: stable-1.30
[init] Using Kubernetes version: v1.30.4
[preflight] Running pre-flight checks
  [WARNING FileExisting-socat]: socat not found in system path
  [WARNING FileMissing-tc]: tc not found in system path
  [WARNING Service-Kubelet]: kubelet service is not enabled, please run 'systemctl enable kubelet.service'
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
W0912 14:55:34.796553 30027 checks.go:844] detected that the sandbox image "registry.k8s.io/pause:3.9" of the container runtime is inconsistent with that used by kubeadm. It is recommended to use "registry.k8s.io/pause:3.9" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-18-9.ec2.internal kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 172.31.18.9]
[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
```

b) Copy this join link

```
kubeadm join 172.31.22.128:6443 --token 2nzclk.1ek0i93tsqnednb9 \
```

```
--discovery-token-ca-cert-hash
```

```
sha256:e7c55b0579b7e928431704c459e9c9c521c4af034e3d346f3418e1afc672928d
```

c) Run the below command -

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

- d) Then, add a common networking plugin called flammel file as mentioned in the code.
 kubectl apply -f

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

```
[ec2-user@ip-172-31-22-128 docker]$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
[ec2-user@ip-172-31-22-128 docker]$ kubectl get pods
No resources found in default namespace.
```

- e) Check the created pod using this command
- kubectl get pods

7) To perform on both worker machine -

- a) sudo yum install iproute-tc socat -y

```
[ec2-user@ip-172-31-27-40 ~]$ sudo yum install iproute-tc socat -y
Last metadata expiration check: 0:11:39 ago on Sat Sep 14 12:44:32 2024.
Dependencies resolved.
=====
                                               Package           Architecture
=====
=Installing:
iproute-tc      x86_64      5.10.0-2.amzn2023.0.5      amazonlinux      455 k
socat          x86_64      1.7.4.2-1.amzn2023.0.2      amazonlinux      303 k

Transaction Summary
=====
Install 2 Packages

Total download size: 758 k
Installed size: 2.0 M
Downloading Packages:
(1/2): socat-1.7.4.2-1.amzn2023.0.2.x86_64.rpm      4.5 MB/s | 303 kB   00:00
(2/2): iproute-tc-5.10.0-2.amzn2023.0.5.x86_64.rpm    6.1 MB/s | 455 kB   00:00
                                                               Total

Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing :                                                               1/1
Installing : socat-1.7.4.2-1.amzn2023.0.2.x86_64          1/2
Installing : iproute-tc-5.10.0-2.amzn2023.0.5.x86_64        2/2
Running scriptlet: iproute-tc-5.10.0-2.amzn2023.0.5.x86_64  2/2
Verifying  : iproute-tc-5.10.0-2.amzn2023.0.5.x86_64        1/2
Verifying  : socat-1.7.4.2-1.amzn2023.0.2.x86_64          2/2

Installed:
  iproute-tc-5.10.0-2.amzn2023.0.5.x86_64      socat-1.7.4.2-1.amzn2023.0.2.x86_64

Complete!
```

- b) sudo systemctl enable kubelet

```
[ec2-user@ip-172-31-17-38 ~]$ sudo systemctl enable kubelet
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /u
sr/lib/systemd/system/kubelet.service.
```

- c) sudo systemctl restart kubelet
- d) kubeadm join 172.31.22.128:6443 --token 2nzclk.1ek0i93tsqnednb9 \
--discovery-token-ca-cert-hash
sha256:e7c55b0579b7e928431704c459e9c9c521c4af034e3d346f3418e1afc672928d

```
[ec2-user@ip-172-31-17-38 ~]$ sudo kubeadm join 172.31.22.128:6443 --token 2nzclk.1ek0i93tsqnednb9 --discovery-token-ca-cert-hash sha256:e7c55b0579b7e928431704c459e9c9c521c4af034e3d346f3418e1afc672928d
[preflight] Running pre-flight checks
error execution phase preflight: couldn't validate the identity of the API Server: failed to request the cluster-info ConfigMap: Get "https://172.31.22.128:6443/api/v1/namespaces/kube-public/configmaps/cluster-info?timeout=10s": context deadline exceeded
To see the stack trace of this error execute with --v=5 or higher
```

If it gives error or refusing connection, restart the master server using sudo systemctl restart kubelet and try to connect.

With the help of command the worker nodes are connected master node and is ready to do task assigned by master node.

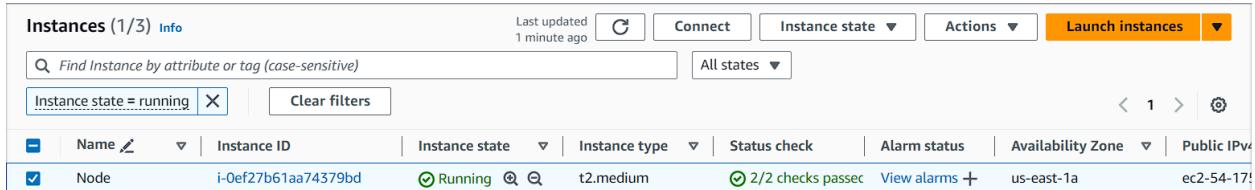
Conclusion -

In this experiment, we connected master nodes from Kubernetes to the worker nodes successfully First, we created the instances and connected with Kubernetes, while installing and configuring, there were some packages, that were needed to be installed separately. Even while connecting the nodes, error occurs and hence, system needs to be restarted and connected properly..

Experiment - 4

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

- 1) Create an EC2 instance and allow SSH traffic to connect.



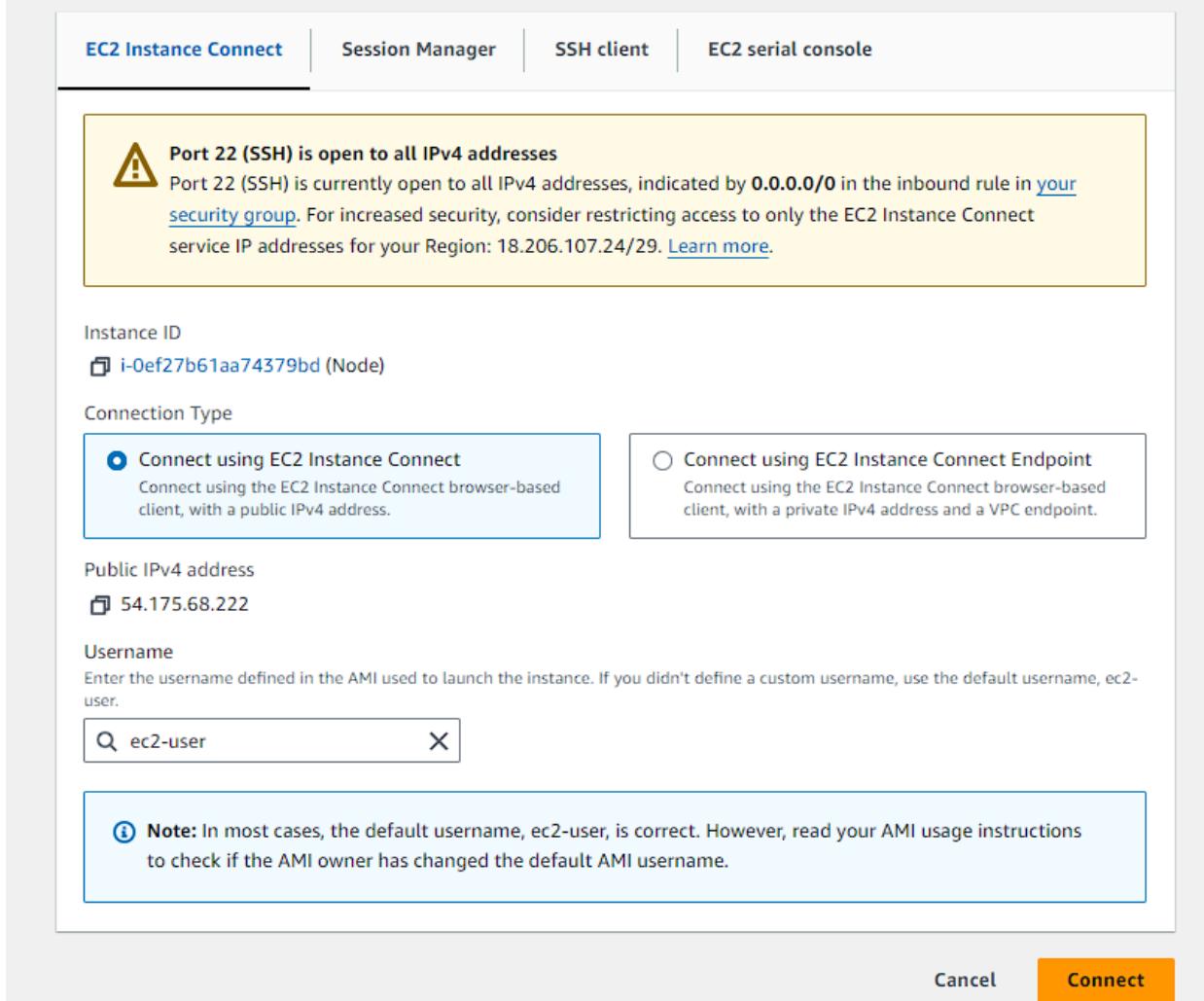
The screenshot shows the AWS EC2 Instances page. It displays one instance named "Node" with the ID i-0ef27b61aa74379bd. The instance is running, has a status check of 2/2 checks passed, and is located in the us-east-1a availability zone. The public IP is listed as ec2-54-17...

- 2) Select the instance and click on Connect and go to Connect.

EC2 > Instances > i-0ef27b61aa74379bd > Connect to instance

Connect to instance Info

Connect to your instance i-0ef27b61aa74379bd (Node) using any of these options



The screenshot shows the "Connect to instance" dialog for the EC2 instance i-0ef27b61aa74379bd (Node). The dialog includes tabs for "EC2 Instance Connect", "Session Manager", "SSH client", and "EC2 serial console". A warning message states: "Port 22 (SSH) is open to all IPv4 addresses. Port 22 (SSH) is currently open to all IPv4 addresses, indicated by 0.0.0.0/0 in the inbound rule in your security group. For increased security, consider restricting access to only the EC2 Instance Connect service IP addresses for your Region: 18.206.107.24/29." Below this, connection settings are shown: "Instance ID" (i-0ef27b61aa74379bd), "Connection Type" (selected "Connect using EC2 Instance Connect"), "Public IPv4 address" (54.175.68.222), and "Username" (ec2-user). A note at the bottom says: "Note: In most cases, the default username, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username." At the bottom right are "Cancel" and "Connect" buttons.

- 3) EC2 server is connected.

```
'      #
~\_ #####
~~ \####\ Amazon Linux 2023
~~ \|##| https://aws.amazon.com/linux/amazon-linux-2023
~~ \#/ \_>
~~~ / 
~~ ._. / 
~/` /` 
>Last login: Sat Sep 14 13:21:24 2024 from 171.48.84.95
[ec2-user@ip-172-31-22-128 ~]$
```

- 4) To install docker run the following command:

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

```
[ec2-user@ip-172-31-22-128 ~]$ sudo yum install docker -y
Last metadata expiration check: 2:24:08 ago on Sat Sep 14 12:44:19 2024.
Package docker-25.0.6-1.amzn2023.0.2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

- 5) Configure cgroup in daemon.json file using the following commands

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

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

```
[ec2-user@ip-172-31-30-107 ~]$ cd /etc/docker
[ec2-user@ip-172-31-30-107 docker]$ cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opt": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
EOF
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opt": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
```

- 6) Run the following command after this:

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

```
[ec2-user@ip-172-31-30-107 docker]$ sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
```

- 7) Check for installation of docker by docker -v command.

```
[ec2-user@ip-172-31-30-107 docker]$ docker -v
Docker version 25.0.5, build 5dc9bcc
```

Install Kubernetes

- 8) Disable SELinux before configuring kubelet

- sudo setenforce 0
- sudo sed -i 's/^SELINUX=enforcing\$/SELINUX=permissive/' /etc/selinux/config

```
[ec2-user@ip-172-31-30-107 docker]$ sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
```

- 9) Add kubernetes repository

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

```
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/repo/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
EOF
```

10) Run the commands to update and install kubernetes packages

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

```
[ec2-user@ip-172-31-31-241 docker]$ sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Last metadata expiration check: 0:00:46 ago on Sat Sep 14 16:39:07 2024.
Dependencies resolved.
=====
Package           Architecture      Version       Repository     Size
=====
Installing:
kubeadm          x86_64          1.31.1-150500.1.1   kubernetes    11 M
kubectl          x86_64          1.31.1-150500.1.1   kubernetes    11 M
kubelet          x86_64          1.31.1-150500.1.1   kubernetes    15 M
=====
Installing dependencies:
comtrack-tools   x86_64          1.4.6-2.amzn2023.0.2  amazonlinux  208 k
cri-tools        x86_64          1.31.1-150500.1.1   kubernetes    6.9 M
kubernetes-cni   x86_64          1.5.1-150500.1.1   kubernetes    7.1 M
libnetfilter_cthelper x86_64        1.0.0-21.amzn2023.0.2  amazonlinux  24 k
libnetfilter_cttimeout x86_64        1.0.0-19.amzn2023.0.2  amazonlinux  24 k
libnetfilter_queue x86_64        1.0.5-2.amzn2023.0.2  amazonlinux  30 k
=====
Transaction Summary
=====
Install 9 Packages

Total download size: 51 M
Installed size: 269 M
Downloading Packages:
1/9: libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64.rpm 412 kB/s | 24 kB 00:00
```

11) Configure internet options to allow bridging

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

```
[ec2-user@ip-172-31-31-241 docker]$ sudo swapoff -a
echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf
sudo sysctl -p
net.bridge.bridge-nf-call-iptables=1
net.bridge.bridge-nf-call-iptables = 1
```

12) Initialize the kubernetes cluster

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

```
[ec2-user@ip-172-31-31-241 docker]$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
  [WARNING FileExisting-socat]: socat not found in system path
  [WARNING FileExisting-tc]: tc not found in system path
  [WARNING Service-Kubelet]: kubelet service is not enabled, please run 'systemctl enable kubelet.service'
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
W0914 16:43:42.619918 27909 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 certificate-dir 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-172-31-31-241.ec2.internal kubernetes.kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 172.31.31.241]
[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-172-31-31-241.ec2.internal localhost] and IPs [172.31.31.241 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-31-241.ec2.internal localhost] and IPs [172.31.31.241 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
```

13) Copy the Join Command-

```
Then you can join any number of worker nodes by running the following on each as root:
kubeadm join 172.31.31.241:6443 --token x6zuwa.bafjhtm4fqxp4yi8 \
--discovery-token-ca-cert-hash sha256:ad0a2b0eb8318975f42be705f750f923adc01bd102ebd7d93401df81429ef5a5
```

```
kubeadm join 172.31.31.241:6443 --token x6zuwa.bafjhtm4fqxp4yi8 \
--discovery-token-ca-cert-hash
sha256:ad0a2b0eb8318975f42be705f750f923adc01bd102ebd7d93401df81429ef5a5
```

14) Run the following commands

```
[ec2-user@ip-172-31-31-241 docker]$ mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
cp: overwrite '/home/ec2-user/.kube/config'? yes
```

15) Add a common network plugin called Flannel as mentioned in the code below:

```
kubectl apply -f
```

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

```
c2-user@ip-172-31-20-245 dockekubectl 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
c2-user@ip-172-31-20-245 docker]$
```

16) Check for creation of pods

```
[ec2-user@ip-172-31-20-245 ~]$ kubectl get pods
NAME      READY     STATUS    RESTARTS   AGE
nginx     0/1      Pending    0          80s
```

17) To change the state from pending to running, use the following command kubectl

```
describe pod nginx
```

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

```

Containers:
  nginx:
    Image:      nginx:1.14.2
    Port:       80/TCP
    Host Port:  0/TCP
    Environment: <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-dmnncs (ro)
Conditions:
  Type        Status
  PodScheduled  False
Volumes:
  kube-api-access-dmnncs:
    Type:           Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:   kube-root-ca.crt
    ConfigMapOptional: <nil>
    DownwardAPI:     true
    QoS Class:       BestEffort
    Node-Selectors:  <none>
    Tolerations:    node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                    node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type     Reason          Age   From            Message
  ----   -----          --   --              --
  Warning FailedScheduling 2m47s default-scheduler 0/1 nodes are available: 1 node(s) had untolerated taint {node-role.kubernetes.io/control-plane: }. preemption: 0/1 nodes are available: 1 Preemption is not helpful for scheduling.
[ec2-user@ip-172-31-20-245 docker]$ kubectl taint nodes ip-172-31-20-245.ec2.internal node-role.kubernetes.io/control-plane-
node/ip-172-31-20-245.ec2.internal untainted

```

18) Check the status of pods.

```
[ec2-user@ip-172-31-20-245 docker]$ kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
nginx    1/1     Running   0          4m3s
```

```
[ec2-user@ip-172-31-20-245 docker]$ kubectl port-forward nginx 8081:80
Forwarding from 127.0.0.1:8081 -> 80
Forwarding from [::1]:8081 -> 80
```

Conclusion:

We successfully configured Kubernetes environment on Amazon Linux EC2 instance. We installed Docker and adjusted its settings to use systemd for cgroup management. Then, we installed Kubernetes. We disabled SELinux, and added the Kubernetes repository. Then we installed the necessary components. Then added a network plugin and deployed on Nginx server. We also addressed issues related to pod scheduling and port forwarding, ensuring the Nginx pod.

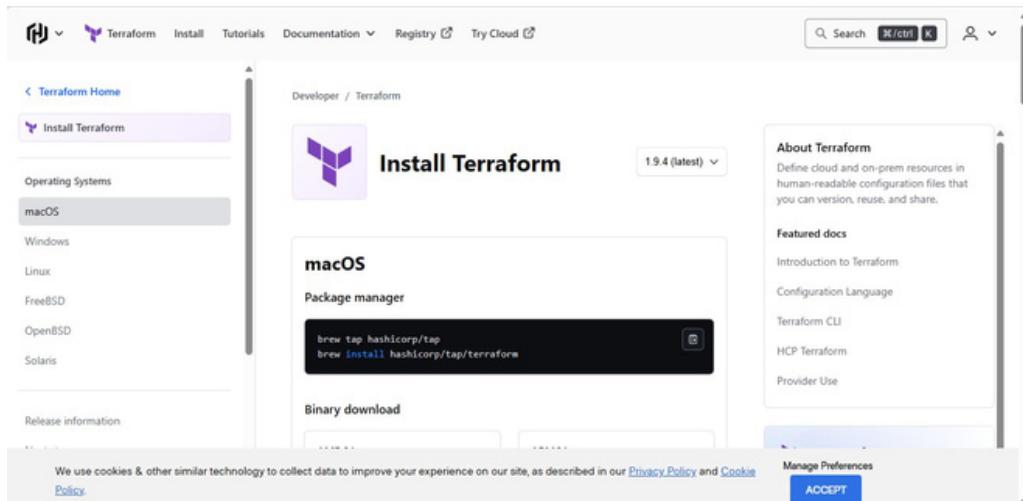
Experiment No: 5

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

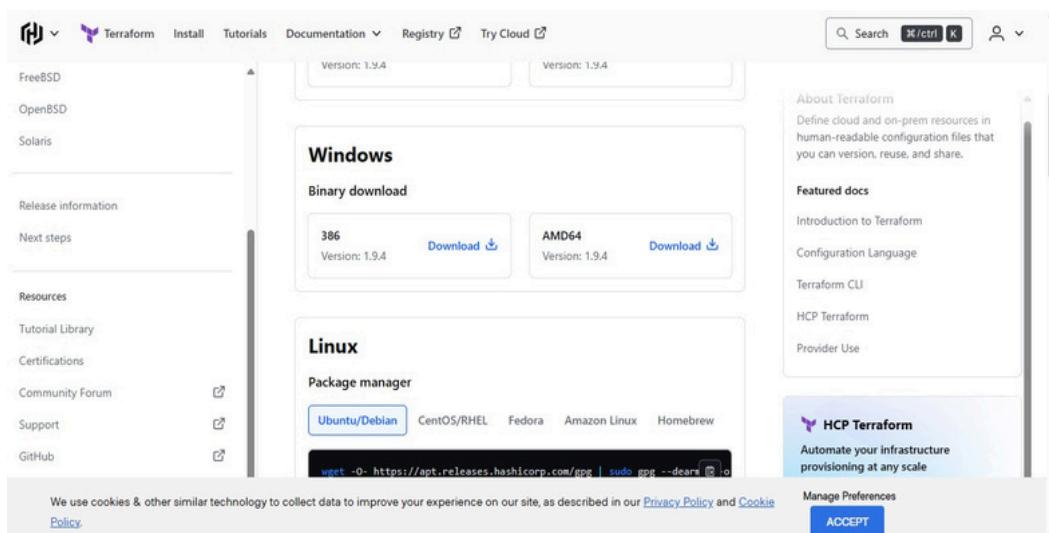
Installation and Configuration Steps of Terraform in Windows:

Step 1: Download terraform

- 1) To install Terraform, First Download the Terraform Cli Utility for windows from terraforms official website <https://www.terraform.io/downloads.html>

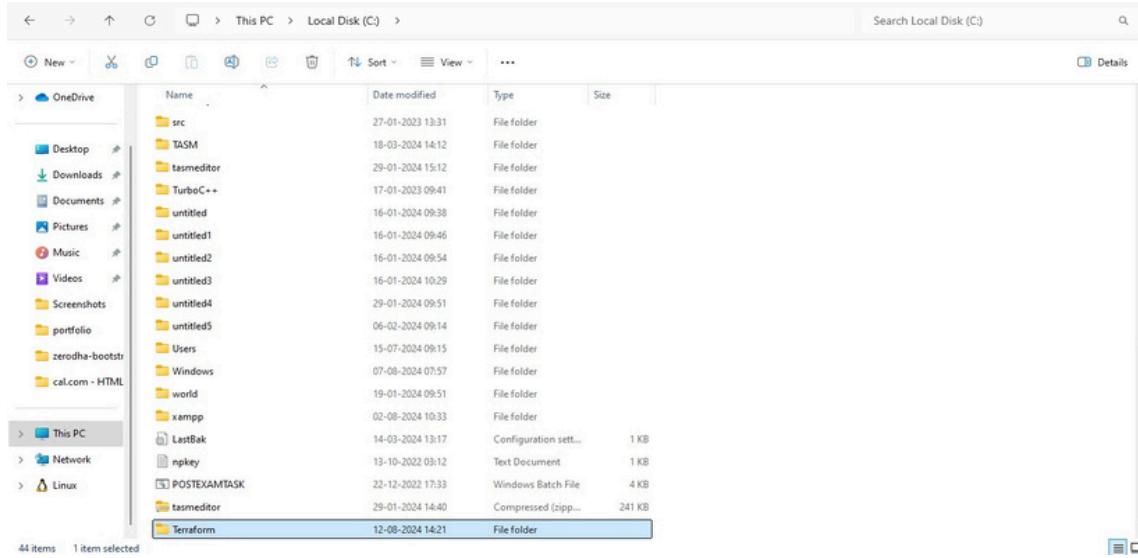


- 2) Select the Operating System Windows followed by either 32 bit or 64 bit based on your OS type.

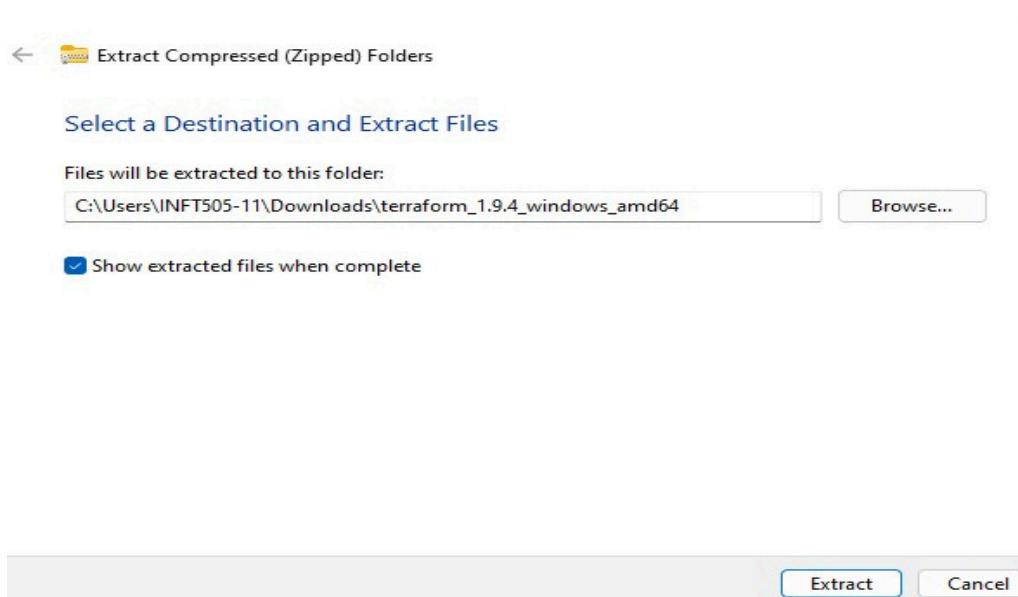


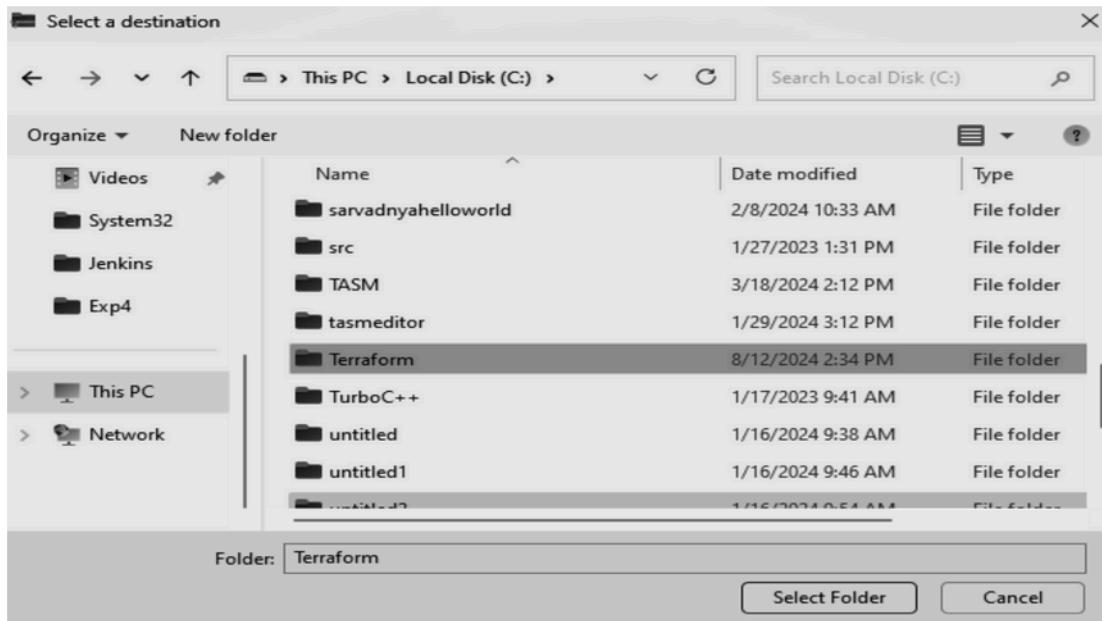
Step 2: Extract the downloaded setup file Terraform.exe in C:\Terraform Directory

1) Go to file manager -> Click on This PC -> move to C drive and create a new folder for example named terraform

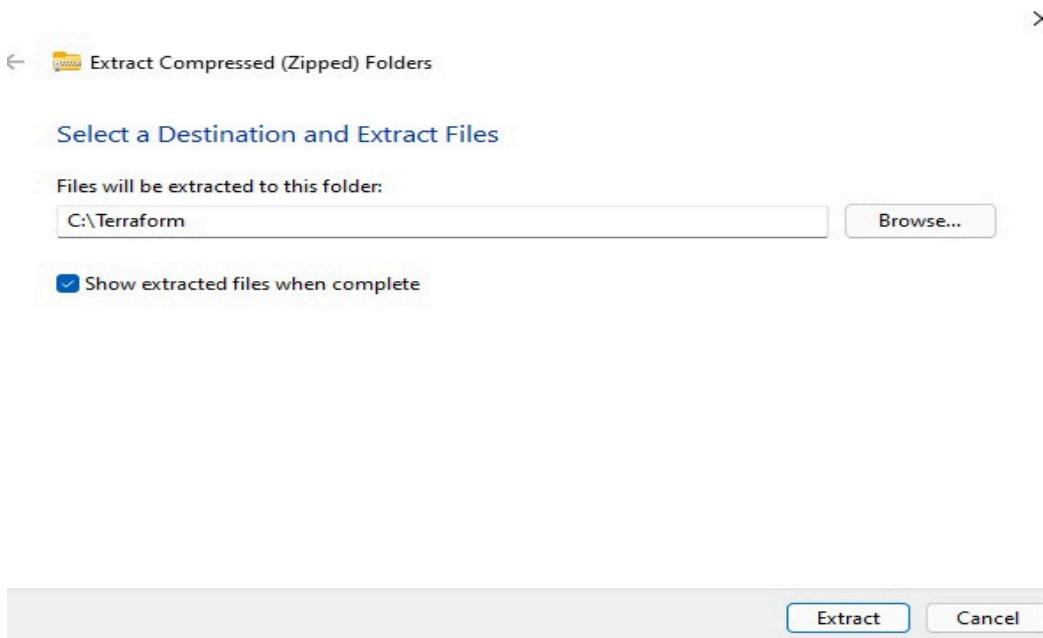


2) Got to location where main file is downloaded , right click on the file name and click on extract all while clicking on extract all it displays the location of the file to be saved , after extraction initially the destination of file to be extracted is not the folder which we have created earlier in previous steps , so to get the exact folder click on Browse and select it.

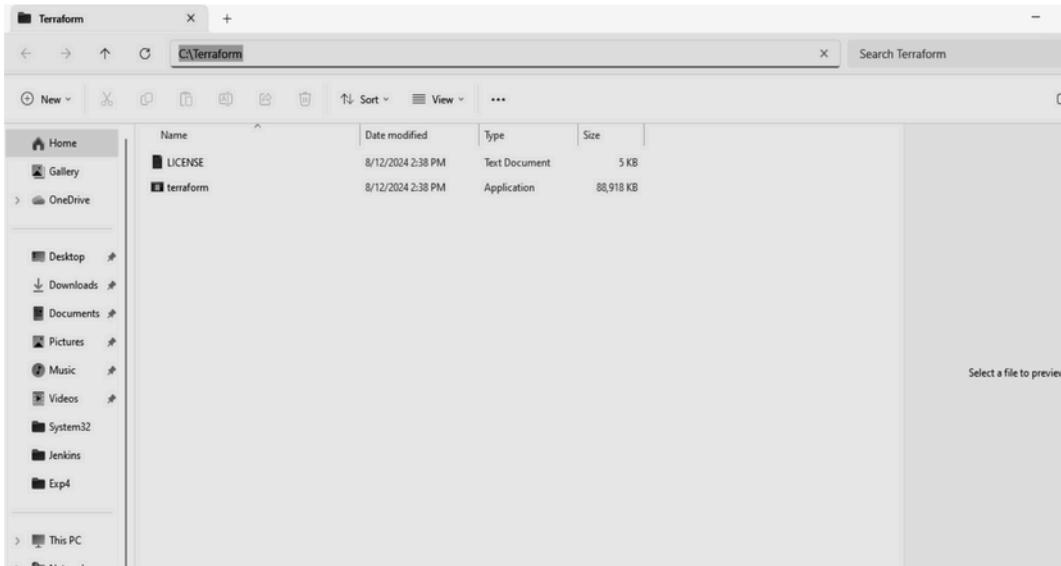




3) Click on extract Button.

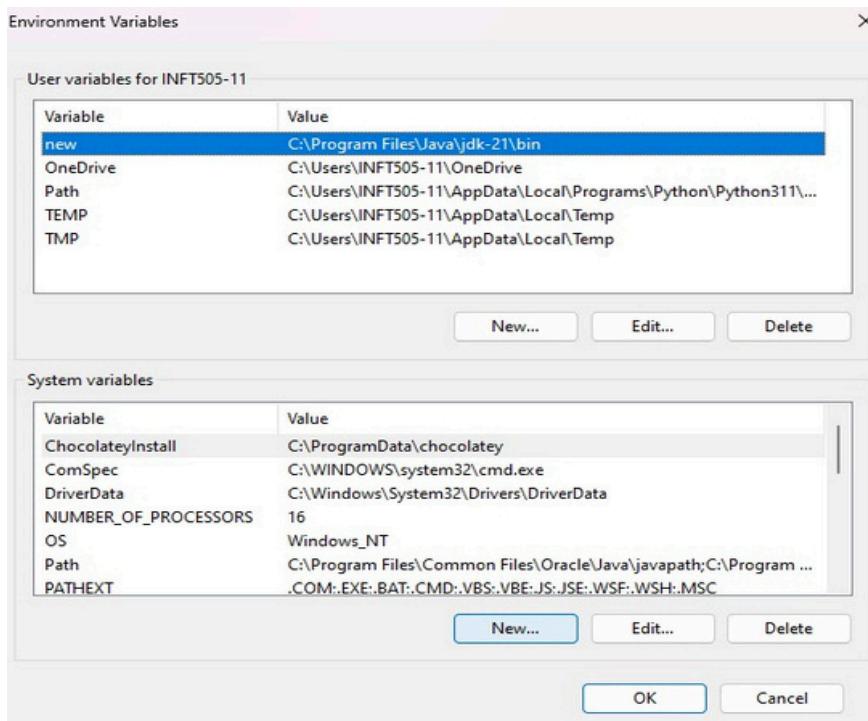


4) After extracting it will redirect to file manager copy the path which will be used for the further steps

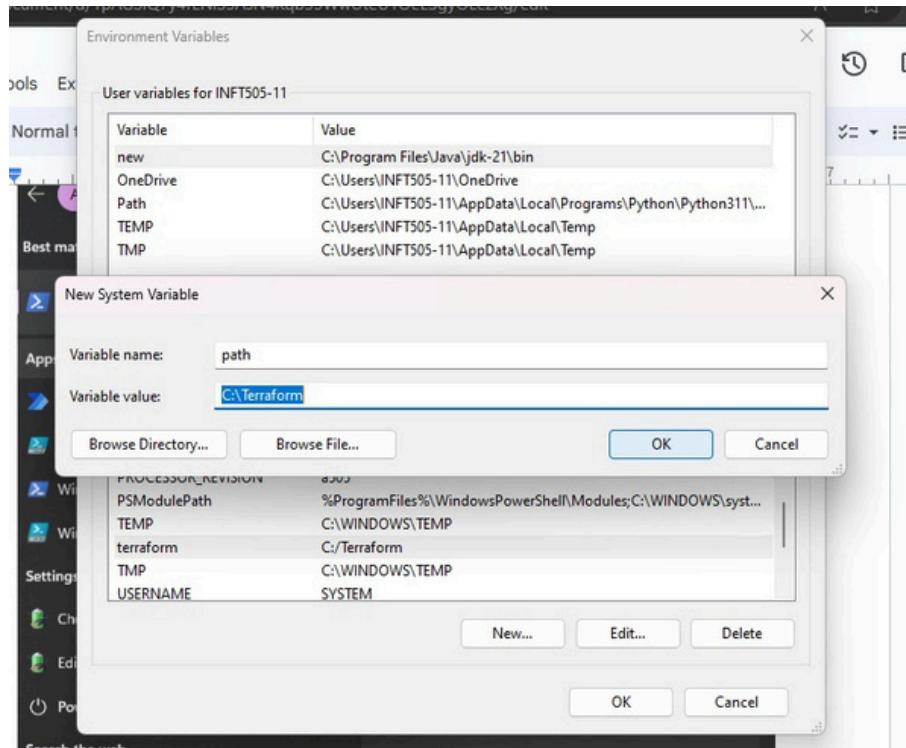


Step 3: Set the System path for Terraform in Environment Variables

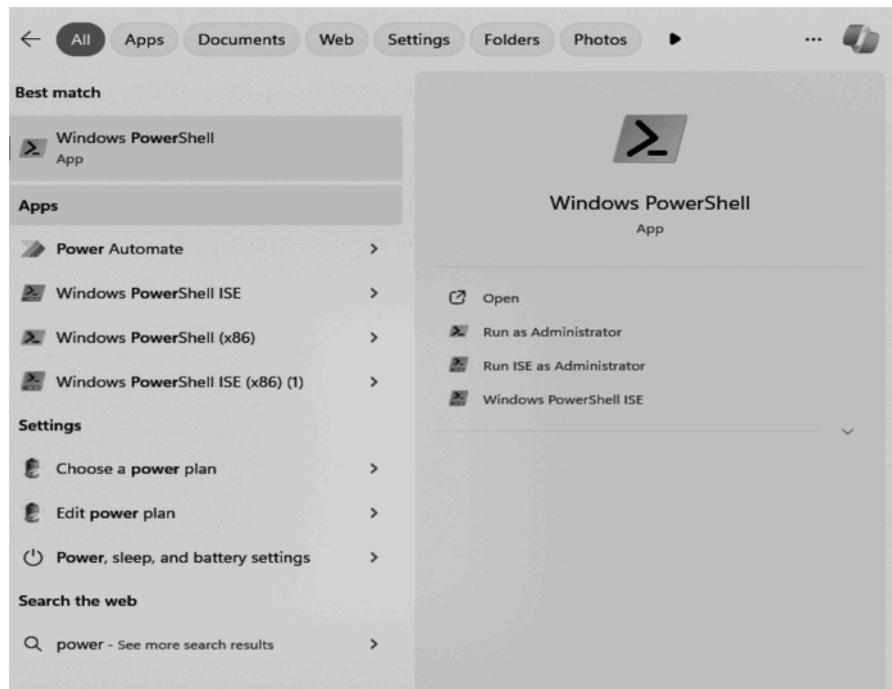
1) Search Environment Variable from search area in the Taskbar then Click on New under System Variables



2) Set the Variable name as path and paste the location of the folder which we copied in previous step after extraction process as the variable value .Click on Ok



Step 4: Open PowerShell with Admin Access



Step 5 : Open Terraform in PowerShell and check its functionality

```
PS C:\Users\INFT505-11> 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
```

If the case of any Errors, then please recheck or set the path of Terraform in Environment variable again.

Aim : Exp 6 To Build, change, and destroy AWS / GCP /Microsoft Azure/ DigitalOcean infrastructure Using Terraform.(S3 bucket or Docker)

- 1) Check the docker version and functionality if its not downloaded you can download it from <https://www.docker.com/>
- 2) Use `docker --version` command to check for the version of docker.

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

```
PS C:\Users\hp> 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
```

- 3) Create a folder named '**Terraform Scripts**' in which we save our different types of scripts which will be further used in this experiment.
- 4) Create a new docker.tf file using an IDE and write the following contents into it to create a Ubuntu Linux container.

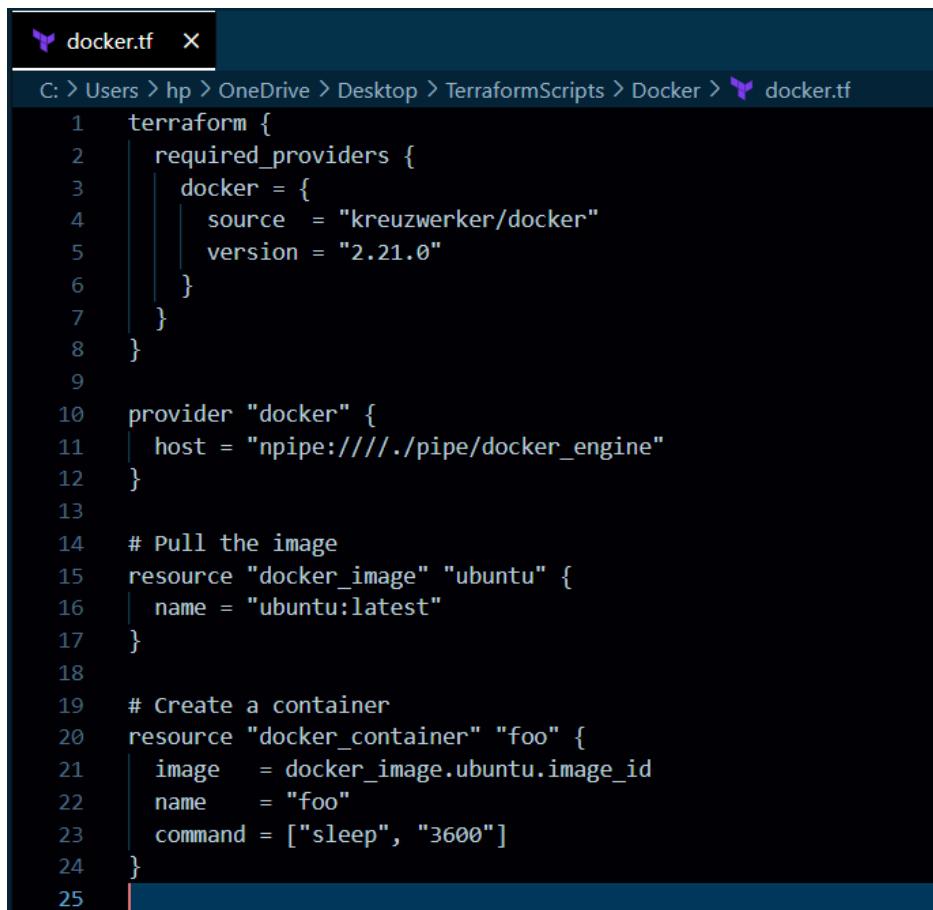
```
terraform {
  required_providers {
    docker = {
      source = "kreuzwerker/docker"
      version = "2.21.0"
    }
  }
}
```

```
}
```

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

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

# Create a container
resource "docker_container" "foo" {
  image  = docker_image.ubuntu.image_id
  name   = "foo"
  command = ["sleep", "3600"]
}
```



The screenshot shows a code editor window with the file name 'docker.tf' at the top. The file path is C:\Users\hp\OneDrive\Desktop\TerraformScripts\Docker\docker.tf. The code itself is a Terraform configuration file:

```
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 }
25 }
```

- 5) Execute *terraform init* command to initialize the resources (Make sure you are in the Docker directory before executing the command)

```
PS C:\Users\hp\OneDrive\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.
```

- 6) Execute *terraform plan* to see the available resources if the build fails, there may be a chance that the docker engine is not running. Go to Docker Desktop and start the engine.

```
PS C:\Users\hp\OneDrive\Desktop\TerraformScripts\Docker> terraform plan
Planning failed. Terraform encountered an error while generating this plan.

Error: Error pinging Docker server: error during connect: This error may indicate that the docker daemon is not running.: Get "http://%2F%2F.%2Fpipe%2Fdocker_engine/_ping": open //./pipe/docker_engine: The system cannot find the file specified.

with provider["registry.terraform.io/kreuzwerker/docker"],
on docker.tf line 10, in provider "docker":
  10: provider "docker" {
```

- 7) Again, run the terraform plan.

```
PS C:\Users\hp\OneDrive\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)
```

- 8) Execute *terraform apply* to apply the configuration, which will automatically create and run the Ubuntu Linux container based on our configuration.

```
PS C:\Users\hp\OneDrive\Desktop\TerraformScripts\Docker> terraform apply

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

Terraform will perform the following actions:

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

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

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

Enter a value: yes

docker_image.ubuntu: Creating...
docker_image.ubuntu: Still creating... [10s elapsed]
docker_image.ubuntu: Still creating... [20s elapsed]
docker_image.ubuntu: Still creating... [30s elapsed]
docker_image.ubuntu: Still creating... [40s elapsed]
docker_image.ubuntu: Still creating... [50s elapsed]
docker_image.ubuntu: Still creating... [1m0s elapsed]
docker_image.ubuntu: Still creating... [1m10s elapsed]
docker_image.ubuntu: Creation complete after 1m11s [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Creating...
docker_container.foo: Still creating... [10s elapsed]
docker_container.foo: Still creating... [20s elapsed]
docker_container.foo: Still creating... [30s elapsed]
docker_container.foo: Creation complete after 34s [id=0461fcf8f00556f61c080a147e0d0b33687fc09868abba5c3e8205f6f6b0331]
```

- 9) Docker images before executing this command

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

Docker images after executing this command

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

- 10) Execute *terraform destroy* to delete the configuration, which will automatically delete the Ubuntu Container.

```
PS C:\Users\hp\OneDrive\Desktop\TerraformScripts\Docker> terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbf74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Refreshing state... [id=0461fcf8f00556f61c080a147e0d0b33687fc09868abba5c3e8205f6f6b0331]

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                 = "0461fcf8f00556f61c080a147e0d0b33687fc09868abba5c3e8205f6f6b0331" -> null
    - id                       = "0461fcf8f00556f61c080a147e0d0b33687fc09868abba5c3e8205f6f6b0331" -> null
    - image                     = "sha256:edbf74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> 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
}

# 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=0461fcf8f00556f61c080a147e0d0b33687fc09868abba5c3e8205f6f6b0331]
docker_container.foo: Still destroying... [id=0461fcf8f00556f61c080a147e0d0b33687fc09868abba5c3e8205f6f6b0331, 10s elapsed]
docker_container.foo: Still destroying... [id=0461fcf8f00556f61c080a147e0d0b33687fc09868abba5c3e8205f6f6b0331, 20s elapsed]
docker_container.foo: Still destroying... [id=0461fcf8f00556f61c080a147e0d0b33687fc09868abba5c3e8205f6f6b0331, 30s elapsed]
docker_container.foo: Still destroying... [id=0461fcf8f00556f61c080a147e0d0b33687fc09868abba5c3e8205f6f6b0331, 40s elapsed]
docker_container.foo: Destruction complete after 44s
docker_image.ubuntu: Destroying... [id=sha256:edbf74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_image.ubuntu: Destruction complete after 7s

Destroy complete! Resources: 2 destroyed.
```

- 11) Docker images after the destroy command execution

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

Experiment 7

Aim:

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

Integrating Jenkins with SonarQube:

Prerequisites:

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

Steps to integrate Jenkins with SonarQube

Prerequisites: Make sure you have docker and jenkins installed.

Run **docker -v** to check the docker installation.

Run

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

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

- Header:** Jenkins logo, Search (CTRL+K), Help icon, Sahil Motiramani (logged in), Log Out.
- Left Sidebar:**
 - Dashboard >
 - + New Item
 - Build History
 - Manage Jenkins
 - My Views
- Build Queue:** A card stating "No builds in the queue."
- Build Executor Status:** A card showing:

Icon	Status	Name	Last Success	Last Failure	Last Duration
Green circle	Idle	sahil 7	24 days #2	N/A	96 ms
Green circle	Idle	Sahil exp6	24 days #3	N/A	1 sec
Blue circle	Idle	SahilExp6	N/A	N/A	N/A
Red circle	Idle	sahiljob	N/A	24 days #1	1.5 sec
- Bottom Navigation:** Icons for S (Server), M (Master), L (Label).

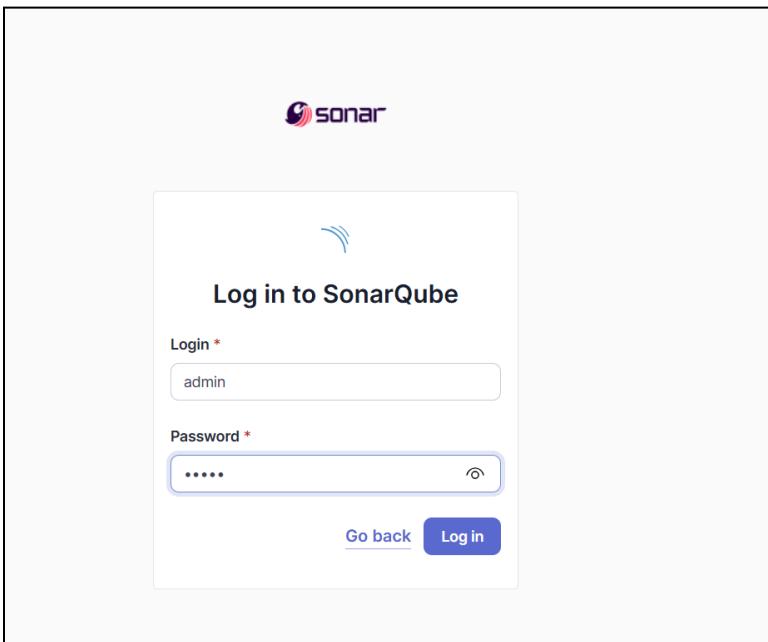
- 2) Run SonarQube in a Docker container using this command -

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

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

```
C:\Users\Lenovo>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
47f6db8dbf2ed99dbe304bc0ebdf47b9d4144c4e4add42055ba44ce231058272
```

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



- 4) Login to SonarQube using username - *admin* and password - *admin*.
(do change the password as you cannot use the default one)

Update your password

⚠ This account should not use the default password.

Enter a new password
All fields marked with * are required

Old Password *

New Password *

Confirm Password *

Update

The screenshot shows the 'Create a local project' wizard in SonarQube. At the top, there's a navigation bar with links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, More, and a search icon. Below the navigation, the title 'How do you want to create your project?' is displayed. A note asks if the user wants to benefit from SonarQube's features like repository import and Pull Request decoration, and suggests creating a project from a favorite DevOps platform. It then asks the user to set up a DevOps platform configuration. Five options are shown: 'Import from Azure DevOps' (Setup), 'Import from Bitbucket Cloud' (Setup), 'Import from Bitbucket Server' (Setup), 'Import from GitHub' (Setup), and 'Import from GitLab' (Setup). Below these, a note says 'Are you just testing or have an advanced use-case? Create a local project.' followed by a button labeled 'Create a local project'.

- 5) Create a manual project in SonarQube with the name sonarqube
(Click on create local project)

1 of 2

Create a local project

Project display name *

exp7



Project key *

exp7



Main branch name *

main

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

[Cancel](#)

[Next](#)

6) Setup the project and come back to Jenkins Dashboard.

Choose the baseline for new code for this project

Use the global setting

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

Define a specific setting for this project

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

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

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

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

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

Plugins

Available plugins

Install	Name	Released
<input checked="" type="checkbox"/>	SonarQube Scanner 2.17.2	7 mo 8 days ago
<input type="checkbox"/>	Sonar Gerrit 388.v9b_ftcb_e42306	3 mo 22 days ago
<input type="checkbox"/>	SonarQube Generic Coverage 1.0 TODO	5 yr 1 mo ago

- 8) Under Jenkins 'Manage Jenkins' then go to 'system', scroll and look for **SonarQube Servers** and click on **add SonarQube** and then enter the details.
 Enter the Server Authentication token if needed.(I didn't do it)
 In SonarQube installations: Under **Name** add <project name of sonarqube> for me its sonarqube_exp7
 In **Server URL** Default is <http://localhost:9000>

SonarQube servers

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

 Environment variables**SonarQube installations**

List of SonarQube installations

Name	sonarqube_exp7
Server URL	Default is http://localhost:9000 http://localhost:9000
Server authentication token	SonarQube authentication token. Mandatory when anonymous access is disabled. - none - + Add ▾
Advanced ▾	

Add SonarQube

ed

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

Dashboard > Manage Jenkins > Tools

The screenshot shows the Jenkins 'Tools' configuration page. At the top, there's a breadcrumb navigation: Dashboard > Manage Jenkins > Tools. Below this, there are five main sections, each with an 'Add [Tool Name]' button:

- Gradle installations**
- SonarScanner for MSBuild installations**
- SonarQube Scanner installations**
- Ant installations**
- Maven installations**

Under the Maven installations section, there's a dropdown menu labeled 'Maven installations' with a downward arrow, and a status message 'Edited'. At the bottom of the page are two buttons: a blue 'Save' button and a grey 'Apply' button.

Click on **Add SonarQube Scanner**.

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

SonarScanner for MSBuild installations

Add SonarScanner for MSBuild

SonarQube Scanner installations

Add SonarQube Scanner

≡ SonarQube Scanner

Name

sonarqube_scanner_exp7

Install automatically ?

≡ Install from Maven Central

Version

SonarQube Scanner 6.2.0.4584

Add Installer ▾

Add SonarQube Scanner

 Saved

10) After the configuration, create a New Item in Jenkins, choose a freestyle project.

Dashboard > All >

Enter an item name

exp7

» Required field



Freestyle project

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

- 11) Choose this GitHub repository in Source Code Management.

https://github.com/shazforiot/MSBuild_firstproject

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

The screenshot shows the 'Configure' screen for a CircleCI project named 'exp7'. Under the 'Source Code Management' tab, the 'Git' option is selected. The 'Repository URL' field contains the URL https://github.com/shazforiot/MSBuild_firstproject.git. The 'Credentials' dropdown is set to '- none -'. There is an 'Advanced' button and a 'Save' button at the bottom.

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

The screenshot shows the 'Configure' screen for a CircleCI project named 'exp7'. Under the 'Build Environment' tab, a dropdown menu is open, listing various build steps. The 'Execute SonarQube Scanner' option is highlighted. Other options include 'Execute Windows batch command', 'Execute shell', 'Invoke Ant', 'Invoke Gradle script', 'Invoke top-level Maven targets', 'Run with timeout', 'Set build status to "pending" on GitHub commit', 'SonarScanner for MSBuild - Begin Analysis', and 'SonarScanner for MSBuild - End Analysis'. Below the dropdown, there is a 'Post-build Actions' section with a 'Save' and 'Apply' button.

Following window will open -

Dashboard > exp7 > Configuration

Configure

Build Steps

Execute SonarQube Scanner

JDK ? JDK to be used for this SonarQube analysis
(Inherit From Job)

Path to project properties ?

Analysis properties ?

Additional arguments ?

JVM Options ?

Add build step ▾

Open sonarQube again and go to Project Information appearing in the right side. Click on it and you can copy the project key from About the Project Section.

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration More Q

exp7 / main ▾

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

About this Project

Quality Gate used
(Default) Sonar way

Project Key ?
exp7

Visibility
Public

Description
No description added for this project.

Tags
No tags ▾

Notifications

A notification is never sent to the author of the event.

Send me an email for:

- Background tasks in failure
- Changes in issues/hotspots assigned to me
- Quality gate changes
- Issues resolved as false positive or accepted
- New issues
- My new issues

Badges

Use this key in place of <projectKey> in the following code

```
sonar.projectKey=<projectKey>
sonar.login =admin
sonar.password =<yourpassword for sonar qube>
sonar.host.url =http://localhost:9000
sonar.sources =
```

(Inherit From Job)

Path to project properties ?

Analysis properties ?

```
sonar.projectKey=exp7
sonar.login=admin
sonar.password=2923
sonar.host.url=http://localhost:9000
sonar.sources=
```

Additional arguments ?

Apply and save.

13) Go to sonarQube and go to administration → Security (dropdown) → Global Permissions.

See the administrator below and check the boxes as checked below..

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

4 of 4 shown

12. Go to jenkins and click build:

Dashboard > exp7 >

Status: exp7

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

Build History: trend ▾

Filter... /

#1 Sep 25, 2024, 11:37 AM

Atom feed for all Atom feed for failures

The screenshot shows the Jenkins console output for build #5. The left sidebar includes links for Status, Changes, Console Output (which is selected), View as plain text, Edit Build Information, Delete build '#5', Timings, Git Build Data, and Previous Build. The main area displays the following log output:

```

Started by user Shubham Jha
Running as SYSTEM
Building on the built-in node in workspace C:\ProgramData\Jenkins\jenkins\workspace\exp7
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\jenkins\workspace\exp7\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/shazforiot/MSBuild_FirstProject # timeout=10
Fetching upstream changes from https://github.com/shazforiot/MSBuild_FirstProject
> git.exe --version # timeout=10
> git --version # git version 2.45.0.windows.1'
> git.exe fetch --tags --progress -- https://github.com/shazforiot/MSBuild_FirstProject +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse --refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision f2bc042c046e72427c30bcae6d6fee7b49adfc (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f f2bc042c046e72427c30bcae6d6fee7b49adfc # timeout=10
Commit message: "updated"
> git.exe rev-list --no-walk f2bc042c046e72427c30bcae6d6fee7b49adfc # timeout=10
[exp7] $ C:\ProgramData\Jenkins\jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube_scanner_exp7\bin\sonar-scanner.bat -Dsonar.host.url=http://localhost:9000 -Dsonar.projectKey=exp7 -Dsonar.login=admin -Dsonar.host.url=http://localhost:9000 -Dsonar.sources=. -Dsonar.password=2923 -Dsonar.projectBaseDir=c:\ProgramData\Jenkins\jenkins\workspace\exp7
13:47:35,366 WARN Property 'sonar.host.url' with value 'http://localhost:9000' is overridden with value 'http://localhost:9000'
13:47:35,382 INFO Scanner configuration file: C:\ProgramData\Jenkins\jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube_scanner_exp7\bin..\conf\sonar-scanner.properties
13:47:35,403 INFO Project root configuration file: NONE
13:47:35,403 INFO SonarScanner CLI 6.2.0.4584
13:47:35,403 INFO Java 21.0.4 Oracle Corporation (64-bit)
13:47:35,413 INFO Windows 11 10.0 amd64
13:47:35,429 INFO User cache: C:\Windows\system32\config\systemprofile\.sonar\cache
13:47:37,015 INFO JRE provisioning: os[windows], arch[amd64]
13:47:47,097 INFO Communicating with SonarQube server 10.6.0.92116
13:47:47,665 INFO Starting SonarScanner Engine...

```

Conclusion:

In this project, we successfully integrated Jenkins with SonarQube to establish a robust automated static application security testing (SAST) pipeline. The setup involved deploying SonarQube using Docker, ensuring smooth container orchestration and efficient resource management. A key component was configuring Jenkins with the appropriate SonarQube plugins, authentication mechanisms, and linking it to a GitHub repository for continuous integration.

One of the challenges was configuring Docker on the Jenkins environment, which required resolving networking issues between the Docker containers and ensuring that the SonarQube server was reachable from Jenkins. Additionally, setting up secure authentication between Jenkins and SonarQube involved troubleshooting token-based authentication and resolving environment path issues, particularly with the **JAVA_HOME** setup for the SonarQube scanner.

After overcoming these obstacles, I integrated the SonarQube scanner as a build step, allowing for continuous code analysis. This setup provided automated detection of code vulnerabilities, code smells, and quality issues. It helped ensure that any new commits triggered immediate analysis, generating detailed reports and promoting continuous improvement in code quality.

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

Theory:

What is SAST?

Static Application Security Testing (SAST) is a methodology that analyzes source code to identify security vulnerabilities before compilation. It is often referred to as white box testing and helps developers detect issues early in the software development lifecycle (SDLC).

Problems SAST Solves

1. **Early Detection:** Identifies vulnerabilities in the initial development stages, reducing later risks.
2. **Real-Time Feedback:** Provides immediate insights, allowing developers to fix issues before moving forward.
3. **Code Navigation:** Offers visual representations of vulnerabilities for easier code understanding.
4. **Guidance on Fixes:** Suggests specific remediation steps without requiring deep security expertise.
5. **Comprehensive Coverage:** Analyzes the entire codebase quickly, outperforming manual reviews.
6. **Regular Scanning:** Ensures continuous security assessment through scheduled scans during builds or releases.

Importance of SAST

- **Resource Efficiency:** Automates code reviews, addressing the resource gap between developers and security staff.
- **Speed:** Processes millions of lines of code in minutes, identifying critical vulnerabilities.
- **Proactive Security:** Integrates security into the development process, preventing vulnerabilities from being overlooked.

What is a CI/CD Pipeline?

A **CI/CD Pipeline** refers to Continuous Integration and Continuous Delivery, automating software development tasks. It includes stages such as coding, building, testing, and deploying, ensuring each step is completed sequentially for efficient releases.

What is SonarQube?

SonarQube is an open-source platform for continuous code quality inspection. It performs static code analysis to generate reports on bugs, vulnerabilities, and code duplications across various programming languages.

Benefits of SonarQube

- **Sustainability:** Optimizes application lifecycle by reducing complexity and vulnerabilities.
- **Increased Productivity:** Minimizes maintenance efforts and costs.
- **Quality Control:** Integrates code quality checks into development.
- **Error Detection:** Alerts developers to fix issues before release.
- **Scalability:** Supports multiple projects without restrictions.
- **Skill Enhancement:** Provides regular feedback to improve developer skills.

Prerequisites:

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

Download The SonarQube CLI according to your system :

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

The screenshot shows the SonarScanner CLI page on the SonarQube documentation site. The top navigation bar includes 'sonarqube' and 'Docs 10.6'. The main content area features a title 'SonarScanner CLI' with tabs for 'SonarScanner' and 'Issue Tracker'. A release note for version 6.2 is displayed, stating: 'Support PKCS12 truststore generated with OpenSSL' and 'Download scanner for: Linux x64, Linux AArch64, Windows x64, macOS x64, macOS AArch64, Docker Any (Requires a pre-installed JVM)'. Below this is a 'Release notes' section. A note at the bottom left says: 'The SonarScanners run on code that is checked out. See Verifying the code checkout step of your build.' On the right side, there's a sidebar titled 'On this page' with links to various SonarScanner-related topics like 'Configuring your project', 'Running SonarScanner CLI from the zip file', etc.

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

The screenshot shows the Jenkins dashboard with the following details:

- Build Queue:** No builds in the queue.
- Build Executor Status:**
 - Built-In Node: 1 Idle, 2 Idle
 - Bhushan_Ex_7_Node: (offline)
 - My_Node: (offline)
- Jobs Overview:**

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀️	Bhushan_EXP_6_Pipeline	24 days #1	N/A	11 sec
✓	☀️	BhushanKor	24 days #5	N/A	1.5 sec
✓	☀️	Devops	1 mo 13 days #7	N/A	15 sec
...	☀️	Exp_6_Job	N/A	N/A	N/A
✗	☁️	Exp_6_Job_Maven	N/A	23 days #6	21 sec
✗	☁️	test	N/A	24 days #2	0.25 sec
✓	☁️	Test2	24 days #3	24 days #2	0.27 sec

Step 2: Run SonarQube in a Docker container using this command

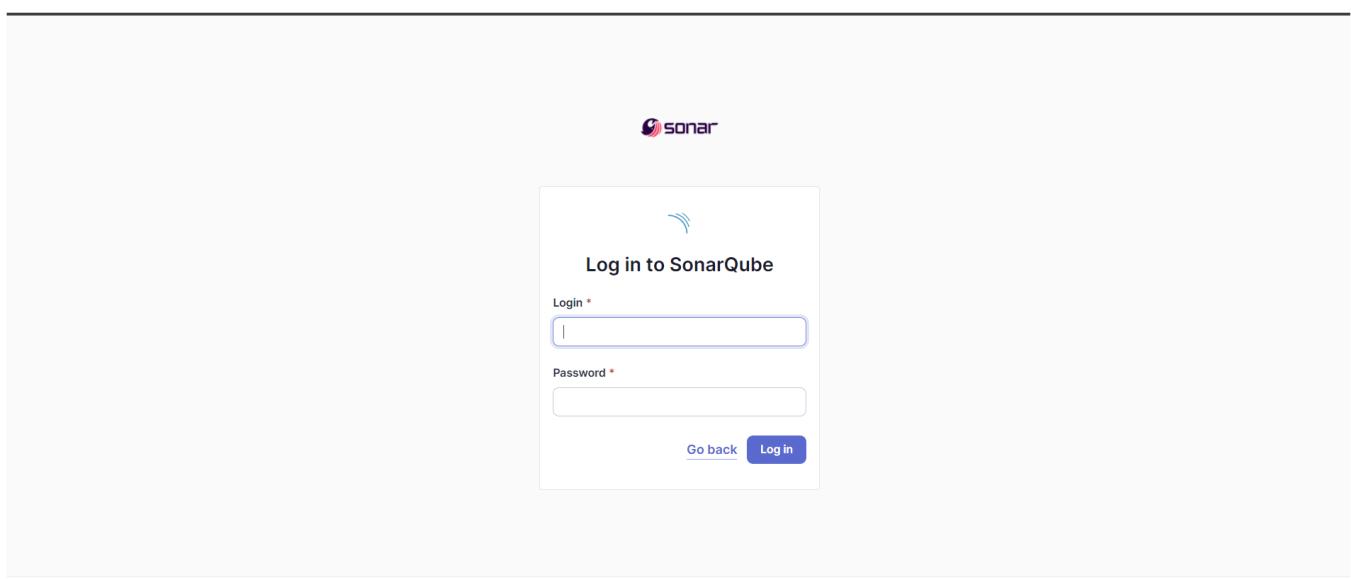
```
docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
```

```
Microsoft Windows [Version 10.0.22621.4169]
(c) Microsoft Corporation. All rights reserved.

C:\Users\INFT505-11>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:late
st
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
d72b183b1866cea7ecdb976a63dfe521172c307eb45eace7b769f726f0bbf989

C:\Users\INFT505-11>
```

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



Start 4: Login to SonarQube using username admin and password admin.

A screenshot of the SonarQube interface showing the "How do you want to create your project?" section. At the top, there is a navigation bar with tabs for "Projects" (which is selected), "Issues", "Rules", "Quality Profiles", "Quality Gates", "Administration", "More", and a search icon. Below the navigation bar, there is a heading "How do you want to create your project?". It asks if the user wants to benefit from all of SonarQube's features and creates their project from their favorite DevOps platform. It then asks if the user needs to set up a DevOps platform configuration. There are six buttons arranged in a grid: "Import from Azure DevOps" (Setup), "Import from Bitbucket Cloud" (Setup), "Import from Bitbucket Server" (Setup), "Import from GitHub" (Setup), "Import from GitLab" (Setup), and a "Create a local project" button. At the bottom, there is a warning message in a yellow box: "⚠️ Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine." The footer contains the same information as the previous screenshot: "SonarQube™ technology is powered by SonarSource SA", links for "Community Edition v10.6 (92116) ACTIVE", "LGPL v3", "Community", "Documentation", "Plugins", and "Web API".

Step 5: Create a manual project in SonarQube with any Name

1 of 2

Create a local project

Project display name *

 ✓

Project key *

 ✓

Main branch name *

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

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

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

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration More Q

Set up project for Clean as You Code

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

Choose the baseline for new code for this project

Use the global setting

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

Define a specific setting for this project

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

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

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

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

Step 6: Setup the project and come back to Jenkins Dashboard.
Go to Manage Jenkins and search for SonarQube Scanner for Jenkins and install it.

The screenshot shows the Jenkins Plugins page. A search bar at the top contains the text "sonar". Below the search bar, there are tabs for "Updates" (17), "Available plugins", "Installed plugins" (selected), and "Advanced settings". A search result for "SonarQube Scanner for Jenkins 2.17.2" is displayed, with a description stating: "This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality." To the right of the description is a "Enabled" switch, which is turned on, and a red "Uninstall" button.

Step 7: Under Jenkins 'Configure System', look for SonarQube Servers and enter the details. Enter the Server Authentication token if needed.

The screenshot shows the Jenkins System configuration page under "SonarQube servers". It includes fields for "Name" (Bhushan's Server), "Server URL" (http://localhost:9000), and "Server authentication token" (a dropdown menu showing "- none -"). There is also an "Advanced" button. At the bottom are "Save" and "Apply" buttons.

Step 8: Search for SonarQube Scanner under Global Tool Configuration. Choose the latest configuration and choose Install automatically.

The screenshot shows the Jenkins Tools configuration page under "SonarQube Scanner installations". It includes fields for "Name" (Bhushan's Scanner) and a checkbox for "Install automatically" (which is checked). Below this is a section for "Install from Maven Central" with a dropdown for "Version" (SonarQube Scanner 6.1.0.4477) and an "Add Installer" button. At the bottom are "Add SonarQube Scanner" and "Ant installations" buttons. A green success message "Saved" is visible at the bottom left.

Step 9: Now click on the new item and select the pipeline project and give name.

Step 10: Under the scripts add the following script

```
node {
    stage('Cloning the GitHub Repo'){
        git 'https://github.com/shazforiot/GOL.git'
    }
    stage('SonarQube analysis') {
        withSonarQubeEnv('sonarqube') {
            bat "<your bin file location of SonarQube CLI>"+
                "-D sonar.login=<your user name>"+
                "-D sonar.password=<your password>"+
                "-D sonar.projectkey=<your projectkey>"+
                "-D sonar.exclusions=vendor/**,resources/**,*/*.java "+
                "-D sonar.host.url=http://localhost:9000"+"
        }
    }
}
```

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

Step 11: Go back to jenkins. Go to the job you had just built and click on Build Now.

The screenshot shows the Jenkins dashboard for the 'Bhushan's Pipeline' job. The left sidebar contains links for Status, Changes, Workspace, Build Now, Configure, Delete Project, SonarQube, and Rename. The main area displays the pipeline status with a green checkmark and the name 'Bhushan's Pipeline'. It includes a SonarQube icon and a 'Permalinks' section with a list of recent builds. A 'Build History' card shows the most recent build (#2) from Sep 19, 2024, at 9:01PM.

Step 12: Check the console output.

The screenshot shows the Jenkins console output for build #11 of the 'Bhushan's Pipeline' job. The left sidebar shows links for Status, Changes, Console Output (which is selected), Edit Build Information, Timings, Git Build Data, Pipeline Overview, Pipeline Console, Thread Dump, Pause/resume, Replay, Pipeline Steps, and Workspaces. The main area displays the console log, which shows the start of the pipeline, cloning of the GitHub repository, and fetching changes from the remote Git repository.

```
Started by user Bhushan
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\Bhushan's Pipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] {
(Cloning the GitHub Repo)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\Bhushan's Pipeline\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/shazforiot/GOL.git # timeout=10
Fetching upstream changes from https://github.com/shazforiot/GOL.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/GOL.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision ba799ba7e1b576f04a461232b0412c5e6e1e5e4 (refs/remotes/origin/master)
> git.exe config core.sparseCheckout # timeout=10
> git.exe checkout -f ba799ba7e1b576f04a461232b0412c5e6e1e5e4 # timeout=10
```

Step 13: Once the build is complete, go back to SonarQube and check the project linked.

The screenshot shows the SonarQube interface with the 'Projects' tab selected. A sidebar on the left contains filters for Quality Gate (Passed: 1, Failed: 0), Reliability (A: 0, B: 0, C: 1, D: 0, E: 0), and Security (A: 1, B: 0). The main panel displays the project 'Bhushan's Pipeline' with a status of 'PUBLIC'. It shows the last analysis was 12 minutes ago, with 683k Lines of Code, HTML, XML, etc. The dashboard includes metrics for Security (0 issues), Reliability (68k issues), Maintainability (164k issues), Coverage (0.0%), and Duplications (50.6%). A note at the bottom states: 'Embedded database should be used for evaluation purposes only. The embedded database will not scale. It will not support connections to a remote instance of SonarQube, and there is no support for migrating your data out of it into a different database vendor.'

The screenshot shows the 'main' project page for 'Bhushan's Pipeline'. The top navigation bar includes tabs for Overview, Issues, Security Hotspots, Measures, Code, and Activity. The 'Project Settings' and 'Project Information' dropdowns are also visible. The main content area shows the 'Passed' quality gate status. Below this, the 'Overall Code' section displays various metrics: Security (0 Open Issues, 0 H, 0 M, 0 L), Reliability (68k Open Issues, 0 H, 47k M, 21k L), Maintainability (164k Open issues, 7 H, 143k M, 21k L), Accepted issues (0), Coverage (0 lines to cover), and Duplications (50.6% on 759k lines). A note at the bottom of the page says: 'On 0 lines to cover.'

Click on Issues and see the different Issues.

Codesmell:

The screenshot shows the SonarQube interface for the 'Issues' tab of the 'main' project. On the left, a sidebar displays software quality metrics: Security (0), Reliability (21k), and Maintainability (164k). Under the 'Type' section, 'Code Smell' is selected, showing 164k issues. The main panel lists three specific code smell findings:

- Use a specific version tag for the image. (Intentionality: Maintainability, Status: Not assigned)
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. (Intentionality: Maintainability, Status: Not assigned)
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. (Intentionality: Maintainability, Status: Not assigned)

At the bottom of the page, a warning message reads: "localhost:9000/security_hotspots?id=Bhushan's%20Pipeline%20for%20evaluation%20purposes%20only".

Bugs:

The screenshot shows the SonarQube interface for the 'Issues' tab of the 'main' project. On the left, a sidebar displays software quality metrics: Security (0), Reliability (47k), and Maintainability (0). Under the 'Type' section, 'Bug' is selected, showing 47k issues. The main panel lists three specific bug findings:

- Insert a <!DOCTYPE> declaration to before this <html> tag. (Consistency, Reliability, Status: Not assigned)
- Add "lang" and/or "xml:lang" attributes to this "chtml" element. (Reliability, Status: Not assigned)
- Add "<th>" headers to this "<table>". (Reliability, Status: Not assigned)

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

Click on Security hotspots and see the different Security hotspots.

The screenshot shows the SonarQube interface for the project "Bhushan's Pipeline". The "Security Hotspots" tab is selected. A single hotspot is listed, indicating 0.0% security hotspots reviewed. The hotspot details show a warning message: "The tomcat image runs with root as the default user. Make sure it is safe here." It includes a "Review" button and tabs for "Where is the risk?", "What's the risk?", "Assess the risk", "How can I fix it?", and "Activity". Below the hotspot, a snippet of a Dockerfile is shown with the problematic line highlighted:

```

FROM tomcat:8-jre8
RUN rm -rf /usr/local/tomcat/webapps/*
COPY target/gameoflife.war /usr/local/tomcat/webapps/ROOT.war
EXPOSE 8080
CMD ["catalina.sh", "run"]

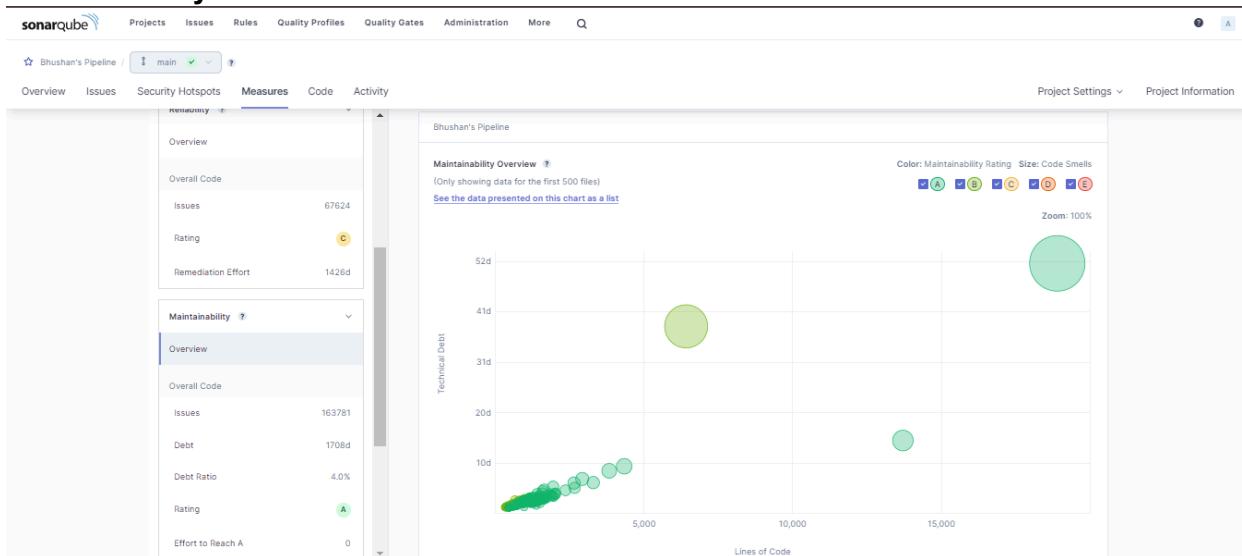
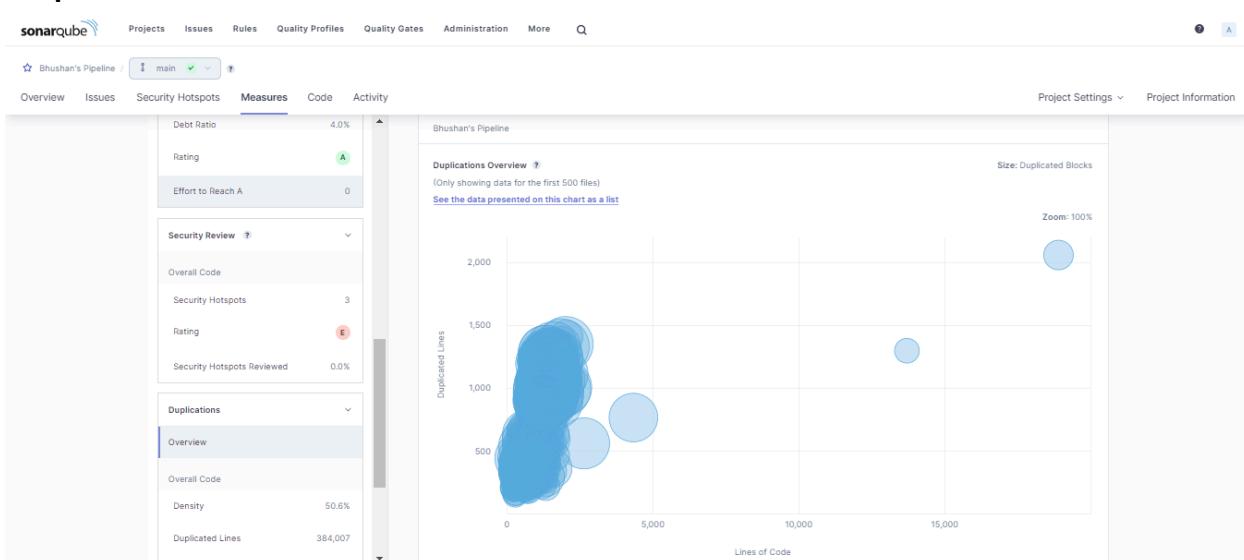
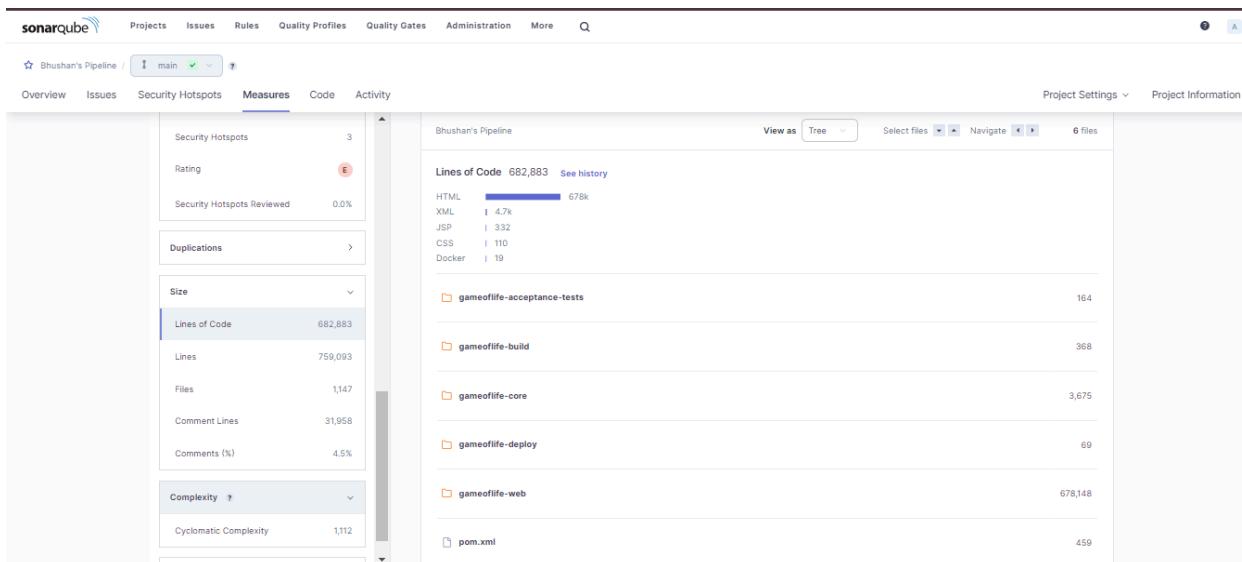
```

Click on Measures and see the Measures in the form of Graphs.
Security:

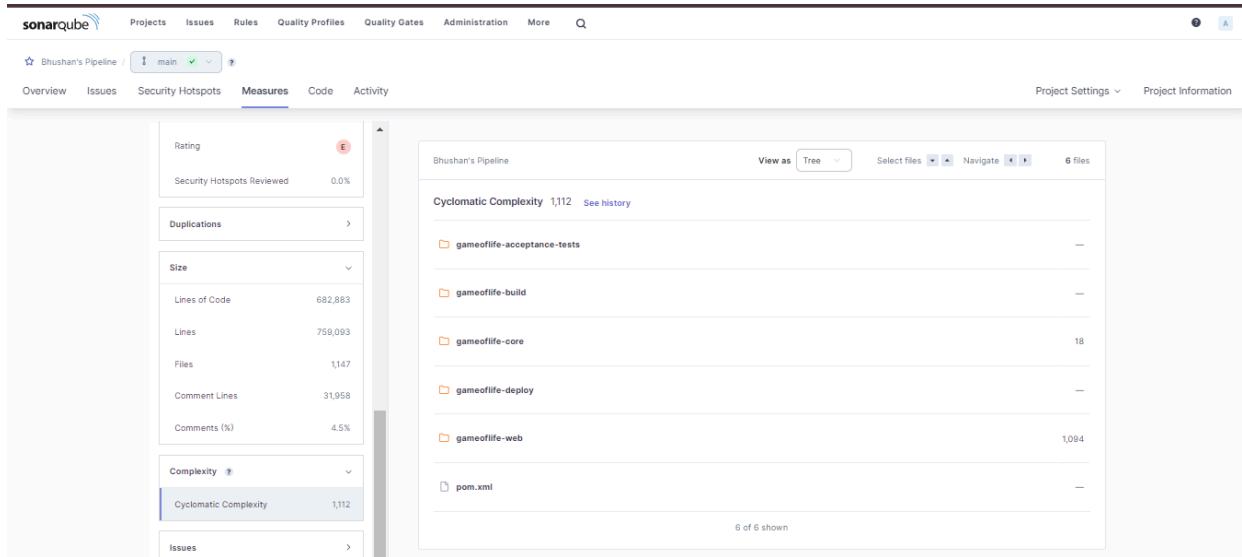
The screenshot shows the SonarQube interface for the project "Bhushan's Pipeline". The "Measures" tab is selected. On the left, a sidebar provides an overview of security measures like Overall Code Issues, Rating, and Remediation Effort. The main area displays a bubble chart titled "Security Overview" showing the relationship between Lines of Code (X-axis) and Security Remediation Effort (Y-axis). The bubbles are colored by security rating (A-E) and sized by vulnerabilities.

Reliability:

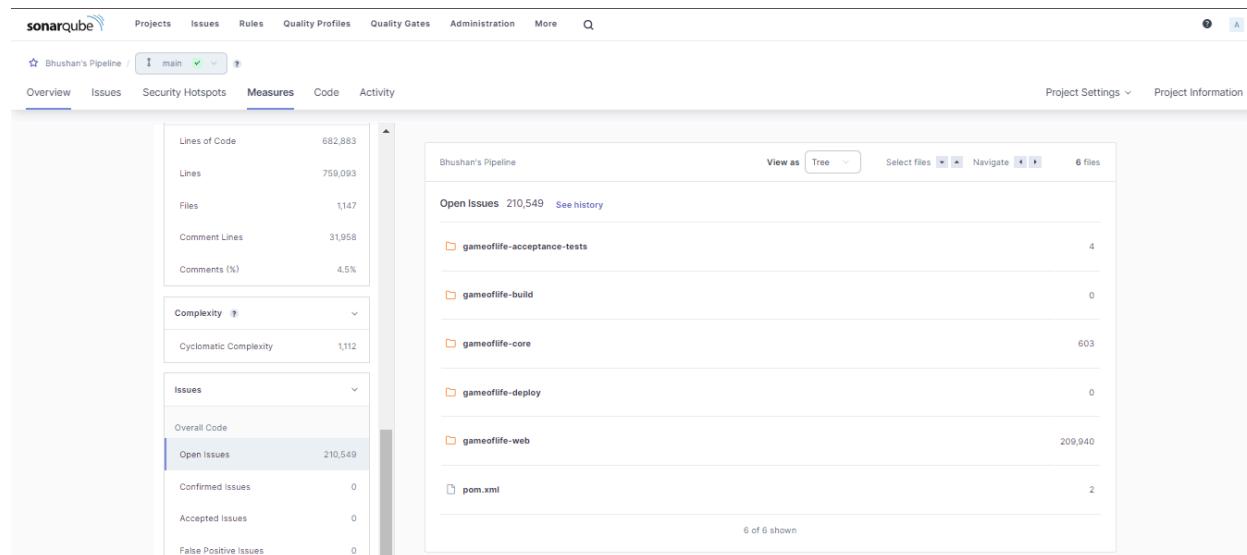
The screenshot shows the SonarQube interface for the project "Bhushan's Pipeline". The "Measures" tab is selected. On the left, a sidebar provides an overview of reliability measures like Overall Code Issues, Rating, and Remediation Effort. The main area displays a bubble chart titled "Reliability Overview" showing the relationship between Lines of Code (X-axis) and Reliability Remediation Effort (Y-axis). The bubbles are colored by reliability rating (A-E) and sized by bugs.

Maintainability:**Duplication:****Sizes:**

Complexity:



Issues:



Conclusion: In this experiment, we performed static analysis of a code using Jenkins CI/CD Pipeline with SonarQube analysis. A pipeline project is to be created which is given a pipeline script. This script contains all the information needed for the project to run the SonarQube analysis. After the necessary configurations are made on Jenkins, the Jenkins project is built. The code provided in this experiment contains lots of errors, bugs, duplications which can be checked on the SonarQube project linked with this build. It streamlines the process of detecting errors in the code.

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.

Theory:

What is Nagios?

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

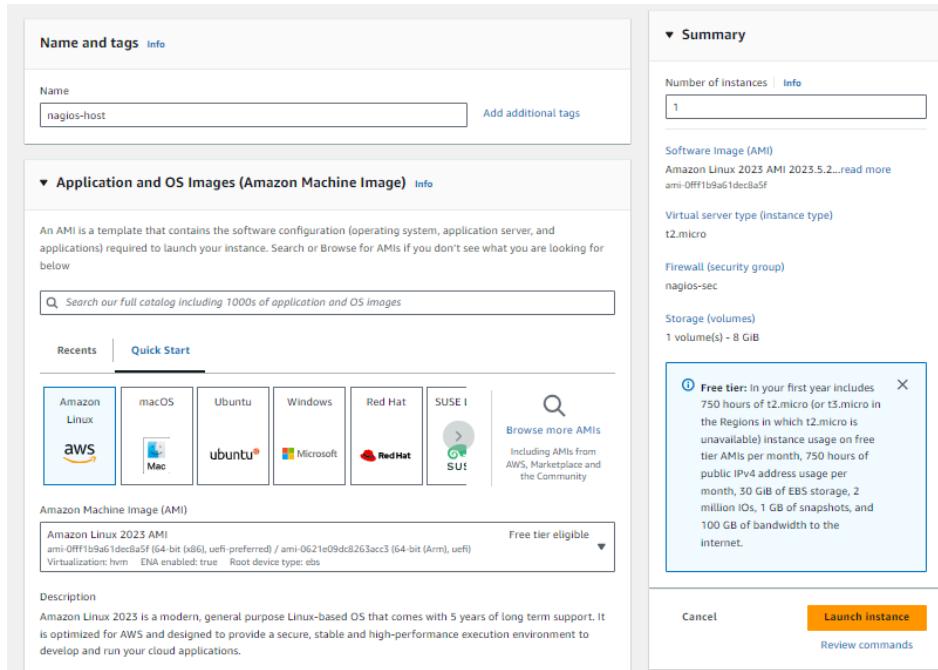
Nagios is used for continuous monitoring of systems, applications, service and business processes in a DevOps culture

Installation of Nagios

Prerequisites: AWS Free Tier

Steps:

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



▼ Key pair (login) [Info](#)

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

Key pair name - required

[Create new key pair](#)

▼ Network settings [Info](#)

Network [Info](#)
vpc-0531204c9e29f6332

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

Auto-assign public IP [Info](#)
Enable
Additional charges apply when outside of free tier allowance

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

Create security group Select existing security group

Common security groups [Info](#)
Select security groups
nagios-sec sg-0641cf06e5063ce27 X
VPC: vpc-0531204c9e29f6332

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

▼ Summary

Number of instances [Info](#)
1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.5.2...[read more](#)
ami-0fff1b9a61dec8a5f

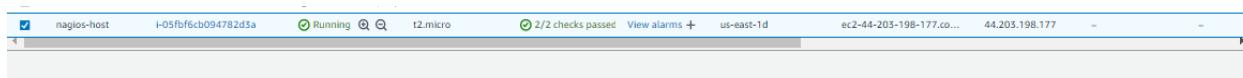
Virtual server type (instance type)
t2.micro

Firewall (security group)
nagios-sec

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

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

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



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

Inbound rules (7)

<input type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
<input type="checkbox"/>	-	sgr-0fb465fa81870fb	IPv4	Custom TCP	TCP	5666	0.0.0.0/0	-
<input type="checkbox"/>	-	sgr-03b98ec2d7ebffffbe	IPv4	HTTPS	TCP	443	0.0.0.0/0	-
<input type="checkbox"/>	-	sgr-0662407c877a008...	IPv4	All ICMP - IPv6	IPv6 ICMP	All	0.0.0.0/0	-
<input type="checkbox"/>	-	sgr-0e4392fd6c7375227	IPv4	All ICMP - IPv4	ICMP	All	0.0.0.0/0	-
<input type="checkbox"/>	-	sgr-0c07df76a67494d56	IPv4	SSH	TCP	22	0.0.0.0/0	-
<input type="checkbox"/>	-	sgr-0a9bd50af59255429	IPv4	All traffic	All	All	0.0.0.0/0	-
<input type="checkbox"/>	-	sgr-0272c1b9ca6fb1b73	IPv4	HTTP	TCP	80	0.0.0.0/0	-

You have to edit the inbound rules of the specified Security Group for this.

3. SSH into Your EC2 instance or simply use EC2 Instance Connect from the browser.

EC2 > Instances > [I-095b567c2f5ae6a91](#) > Connect to instance

Connect to instance [Info](#)

Connect to your instance I-095b567c2f5ae6a91 (nagios-host) using any of these options

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

⚠️ All ports are open to all IPv4 addresses in your security group

All ports are currently open to all IPv4 addresses, indicated by All and 0.0.0.0/0 in the inbound rule in [your security group](#). For increased security, consider restricting access to only the EC2 Instance Connect service IP addresses for your Region: 18.206.107.24/29. [Learn more.](#)

Instance ID I-095b567c2f5ae6a91 (nagios-host)	Connection Type
<input checked="" type="radio"/> Connect using EC2 Instance Connect Connect using the EC2 Instance Connect browser-based client, with a public IPv4 or IPv6 address.	
<input type="radio"/> Connect using EC2 Instance Connect Endpoint Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.	
<input checked="" type="radio"/> Public IPv4 address 54.162.128.119	
<input type="radio"/> IPv6 address	
Username	
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ec2-user. <input type="text" value="ec2-user"/> X	
<p>Note: In most cases, the default username, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.</p>	

[Cancel](#) | [Connect](#)

```
[ec2-user@ip-172-31-34-108 ~]$ sudo yum update
Last metadata expiration check: 0:04:33 ago on Fri Oct  4 04:35:02 2024.
Dependencies resolved.
Nothing to do.
Complete!
```

sudo yum update

```
[ec2-user@ip-172-31-91-91 ~]$  
sudo yum update  
Last metadata expiration check: 0:19:03 ago on Sun Sep 29 06:56:15 2024.  
Dependencies resolved.  
Nothing to do.  
Complete!  
[ec2-user@ip-172-31-91-91 ~]$ |
```

sudo yum install httpd php

```
[ec2-user@ip-172-31-91-91 ~]$ sudo yum install httpd php
Last metadata expiration check: 0:19:29 ago on Sun Sep 29 06:56:15 2024.
Dependencies resolved:
=====
| Package           | Architecture | Version      | Repository | Size   |
=====
Installing:
| httpd            | x86_64       | 2.4.62-1.amzn2023 | amazonlinux | 48 k   |
| httpd-tools      | x86_64       | 8.3.10-1.amzn2023.0.1 | amazonlinux | 10 k   |
=====
Installing dependencies:
| apr              | x86_64       | 1.7.2-2.amzn2023.0.2 | amazonlinux | 129 k  |
| apr-util         | x86_64       | 1.6.3-1.amzn2023.0.1 | amazonlinux | 98 k   |
| generic-logos-httdp | noarch      | 18.0.0-0.12.amzn2023.0.3 | amazonlinux | 19 k   |
| httpd-core       | x86_64       | 2.4.62-1.amzn2023 | amazonlinux | 1.4 M   |
| httpd-filesystem | noarch      | 2.7.1-1.amzn2023 | amazonlinux | 11 k   |
| httpd-tools      | x86_64       | 2.1.4-2.1.amzn2023 | amazonlinux | 81 k   |
| libaprutil        | x86_64       | 1.8.9-4.amzn2023.0.2 | amazonlinux | 315 k  |
| libxml            | x86_64       | 1.8.19-4.amzn2023 | amazonlinux | 176 k  |
| libxml2           | x86_64       | 1.1.34-5.amzn2023.0.2 | amazonlinux | 241 k  |
| mailcap          | noarch      | 2.1.49-3.amzn2023.0.3 | amazonlinux | 33 k   |
| nginx-filesystem | noarch      | 1.12.4-0.1.amzn2023.0.4 | amazonlinux | 9.8 k   |
| php8-3-cgi       | x86_64       | 8.0.29-1.amzn2023.0.1 | amazonlinux | 3.7 M   |
| php8-3-common    | x86_64       | 8.3.10-1.amzn2023.0.1 | amazonlinux | 737 k  |
| php8-3-process   | x86_64       | 8.3.10-1.amzn2023.0.1 | amazonlinux | 45 k   |
| php8-3-xml       | x86_64       | 8.3.10-1.amzn2023.0.1 | amazonlinux | 154 k  |
=====
Installing weak dependencies:
| apr-util-openssl | x86_64       | 1.6.3-1.amzn2023.0.1 | amazonlinux | 17 k   |
| mod_http2        | x86_64       | 2.0.27-1.amzn2023.0.3 | amazonlinux | 160 k  |
| mod_perl         | x86_64       | 2.0.10-1.amzn2023.0.1 | amazonlinux | 61 k   |
| php8-3-fpm       | x86_64       | 8.3.10-1.amzn2023.0.1 | amazonlinux | 1.9 M   |
| php8-3-mbstring  | x86_64       | 8.3.10-1.amzn2023.0.1 | amazonlinux | 528 k  |
| php8-3-ocache    | x86_64       | 8.3.10-1.amzn2023.0.1 | amazonlinux | 379 k  |
| php8-3-pdo       | x86_64       | 8.3.10-1.amzn2023.0.1 | amazonlinux | 89 k   |
| php8-3-sodium    | x86_64       | 8.3.10-1.amzn2023.0.1 | amazonlinux | 41 k   |

```

```
Total                                         22 MB/s | 10 MB   00:00
Running transaction check.
Transaction check succeeded.
Running transaction test.
Transaction test succeeded.
Running transaction.
Preparing :
  Installing : php8.3-common-8.3.10-1.amzn2023.0.1.x86_64           1/1
  Installing : apr-1.7.2-2.amzn2023.0.2.x86_64                      1/25
  Installing : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64          2/25
  Installing : apr-util-1.6.3-1.amzn2023.0.1.x86_64                  3/25
  Installing : mailcap-2.1.49-3.amzn2023.0.3.noarch                 4/25
  Installing : httpd-filesystem-2.4.62-1.amzn2023.noarch            5/25
Running scriptlet: httpd-filesystem-2.4.62-1.amzn2023.noarch             6/25
```

sudo yum install gcc glibc glibc-common

```
[ec2-user@ip-172-31-34-108 ~]$ sudo yum install gcc glibc glibc-common
Last metadata expiration check: 0:06:01 ago on Fri Oct  4 04:35:02 2024.
Package glibc-2.34-52.amzn2023.0.11.x86_64 is already installed.
Package glibc-common-2.34-52.amzn2023.0.11.x86_64 is already installed.
Dependencies resolved.
=====
 Package          Arch    Version        Repository      Size
=====
Installing:
  gcc              x86_64  11.4.1-2.amzn2023.0.2      amazonlinux   32 M
Installing dependencies:
  annobin-docs     noarch  10.93-1.amzn2023.0.1      amazonlinux   92 k
  annobin-plugin-gcc x86_64  10.93-1.amzn2023.0.1      amazonlinux  887 k
  cpp              x86_64  11.4.1-2.amzn2023.0.2      amazonlinux   10 M
  gc               x86_64  8.0.4-5.amzn2023.0.2      amazonlinux  105 k
  glibc-devel      x86_64  2.34-52.amzn2023.0.11     amazonlinux   27 k
  glibc-headers-x86 noarch  2.34-52.amzn2023.0.11     amazonlinux  427 k
  guile22         x86_64  2.2.7-2.amzn2023.0.3      amazonlinux   6.4 M
  kernel-headers   x86_64  6.1.109-118.189.amzn2023 amazonlinux  1.4 M
```

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

Complete!
```

sudo yum install gd gd-devel

```
[ec2-user@ip-172-31-34-108 ~]$ sudo yum install gd gd-devel
Last metadata expiration check: 0:06:19 ago on Fri Oct 4 04:35:02 2024.
Dependencies resolved.
=====
 Package          Arch    Version           Repository  Size
=====
 Installing:
  gd                  x86_64  2.3.3-5.amzn2023.0.3      amazonlinux 139 k
  gd-devel            x86_64  2.3.3-5.amzn2023.0.3      amazonlinux 38 k
 Installing dependencies:
  brotli              x86_64  1.0.9-4.amzn2023.0.2      amazonlinux 314 k
  brotli-devel        x86_64  1.0.9-4.amzn2023.0.2      amazonlinux 31 k
  bzip2-devel          x86_64  1.0.8-6.amzn2023.0.2      amazonlinux 214 k
  cairo                x86_64  1.17.6-2.amzn2023.0.1      amazonlinux 684 k
  cmake-filesystem     x86_64  3.22.2-1.amzn2023.0.4      amazonlinux 16 k
  fontconfig           x86_64  2.13.94-2.amzn2023.0.2      amazonlinux 273 k
  fontconfig-devel     x86_64  2.13.94-2.amzn2023.0.2      amazonlinux 128 k
  fonts-filesystem     noarch  1:2.0.5-12.amzn2023.0.2      amazonlinux 9.5 k
  freetype              x86_64  2.13.2-5.amzn2023.0.1      amazonlinux 423 k
  freetype-devel       x86_64  2.13.2-5.amzn2023.0.1      amazonlinux 912 k
  glib2-devel           x86_64  2.74.7-689.amzn2023.0.2      amazonlinux 486 k
  google-noto-fonts-common noarch  20201206-2.amzn2023.0.2      amazonlinux 15 k
  google-noto-sans-vf-fonts
                        noarch  20201206-2.amzn2023.0.2      amazonlinux 492 k
  graphite2             x86_64  1.3.14-7.amzn2023.0.2      amazonlinux 97 k
  graphite2-devel       x86_64  1.3.14-7.amzn2023.0.2      amazonlinux 21 k
  harfbuzz              x86_64  7.0.0-2.amzn2023.0.1      amazonlinux 868 k
  harfbuzz-devel        x86_64  7.0.0-2.amzn2023.0.1      amazonlinux 404 k
  harfbuzz-icu          x86_64  7.0.0-2.amzn2023.0.1      amazonlinux 18 k
  jbigkit-libs           x86_64  2.1-21.amzn2023.0.2      amazonlinux 54 k
  langpacks-core-font-en noarch  3.0-21.amzn2023.0.4      amazonlinux 10 k
  libICE                 x86_64  1.0.10-6.amzn2023.0.2      amazonlinux 71 k
  libSM                 x86_64  1.2.3-8.amzn2023.0.2      amazonlinux 42 k
  libX11                 x86_64  1.7.2-2.amzn2023.0.4      amazonlinux 657 k
=====
 Installed:
  brotli-1.0.9-4.amzn2023.0.2.x86_64
  cairo-1.17.6-2.amzn2023.0.1.x86_64
  fontconfig-devel-2.13.94-2.amzn2023.0.2.x86_64
  freetype-devel-2.13.2-5.amzn2023.0.1.x86_64
  glib2-devel-2.74.7-689.amzn2023.0.2.x86_64
  graphite2-1.3.14-7.amzn2023.0.2.x86_64
  harfbuzz-devel-7.0.0-2.amzn2023.0.1.x86_64
  langpacks-core-font-en-3.0-21.amzn2023.0.4.noarch
  libX11-1.7.2-3.amzn2023.0.4.x86_64
  libX11-xcb-1.7.2-3.amzn2023.0.4.x86_64
  libXext-1.3.4-6.amzn2023.0.2.x86_64
  libXrender-0.9.10-14.amzn2023.0.2.x86_64
  libffi-devel-3.4.4-1.amzn2023.0.1.x86_64
  libjpeg-turbo-2.1.4-2.amzn2023.0.5.x86_64
  libpng-2.1.6.37-10.amzn2023.0.6.x86_64
  libsep0-devel-3.4-3.amzn2023.0.3.x86_64
  libxcb-1.2.4-1.amzn2023.0.6.x86_64
  libxcb-devel-1.13.1-7.amzn2023.0.2.x86_64
  pcre2-utf16-10.40-1.amzn2023.0.3.x86_64
  sysprof-capture-devel-3.40.1-2.amzn2023.0.2.x86_64
  xz-devel-5.2.5-9.amzn2023.0.2.x86_64
=====
 bzip2-devel-1.0.8-6.amzn2023.0.2.x86_64
  fontconfig-2.13.94-2.amzn2023.0.2.x86_64
  freetype-2.13.2-5.amzn2023.0.1.x86_64
  gd-devel-2.3.3-5.amzn2023.0.3.x86_64
  google-noto-sans-vf-fonts-20201206-2.amzn2023.0.2.noarch
  graphite2-devel-1.3.14-7.amzn2023.0.2.x86_64
  harfbuzz-7.0.0-2.amzn2023.0.1.x86_64
  jbigkit-libs-2.1-21.amzn2023.0.2.x86_64
  libICE-1.0.10-6.amzn2023.0.2.x86_64
  libXau-1.0.9-6.amzn2023.0.2.x86_64
  libXau-common-1.7.2-3.amzn2023.0.4.noarch
  libXau-1.0.9-6.amzn2023.0.2.x86_64
  libXpm-3.5.15-2.amzn2023.0.3.x86_64
  libXt-1.2.0-4.amzn2023.0.2.x86_64
  libXt-1.2.0-4.amzn2023.0.2.x86_64
  libicu-67.1-7.amzn2023.0.3.x86_64
  libjpeg-turbo-devel-2.1.4-2.amzn2023.0.5.x86_64
  libpng-devel-2.1.6.37-10.amzn2023.0.6.x86_64
  libtiff-4.0.0-4.amzn2023.0.18.x86_64
  libwebp-devel-1.2.4-1.amzn2023.0.6.x86_64
  libxml2-devel-2.10.4-4.amzn2023.0.6.x86_64
  pcre2-utf32-10.40-1.amzn2023.0.3.x86_64
  xml-common-0.6.3-56.amzn2023.0.2.noarch
  zlib-devel-1.2.11-33.amzn2023.0.5.x86_64
=====
 pixman-0.40.0-3.amzn2023.0.3.x86_64
  xorg-x11proto-devel-2021.4-1.amzn2023.0.2.noarch
=====
 Complete!
```

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

sudo adduser -m nagios

sudo passwd nagios

(password : *hello123*)

```
[ec2-user@ip-172-31-34-108 ~]$ sudo adduser -m nagios
sudo passwd nagios
Changing password for user nagios.
New password:
[ec2-user@ip-172-31-34-108 ~]$ sudo passwd nagios
Changing password for user nagios.
New password:
BAD PASSWORD: The password fails the dictionary check - it is based on a dictionary word
Retype new password:
passwd: all authentication tokens updated successfully.
```

6. Create a new user group

sudo groupadd nagcmd

```
[ec2-user@ip-172-31-34-108 ~]$ sudo groupadd nagcmd
```

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

sudo usermod -a -G nagcmd nagios

sudo usermod -a -G nagcmd apache

```
[ec2-user@ip-172-31-34-108 ~]$ sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
```

8. Create a new directory for Nagios downloads

mkdir ~/downloads

cd ~/downloads

```
[ec2-user@ip-172-31-34-108 ~]$ mkdir ~/downloads
cd ~/downloads
```

9. Use wget to download the source zip files.

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

```
[ec2-user@ip-172-31-34-108 downloads]$ wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
--2024-10-04 04:42:39--  https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 2600:3c00::f03c:92ff:fef7:45ce
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2065473 (2.0M) [application/x-gzip]
Saving to: 'nagios-4.5.5.tar.gz'

nagios-4.5.5.tar.g 100%[=====] 1.97M 5.31MB/s in 0.4s

2024-10-04 04:42:39 (5.31 MB/s) - 'nagios-4.5.5.tar.gz' saved [2065473/2065473]
```

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

```
[ec2-user@ip-172-31-34-108 downloads]$ wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
--2024-10-04 04:42:47-- 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-2.4 100%[=====] 2.62M 5.79MB/s in 0.5s

2024-10-04 04:42:48 (5.79 MB/s) - 'nagios-plugins-2.4.11.tar.gz' saved [2753049/2753049]

[ec2-user@ip-172-31-34-108 downloads]$ tar zxvf nagios-4.5.5.tar.gz
nagios-4.5.5/
nagios-4.5.5/.github/
nagios-4.5.5/.github/workflows/
nagios-4.5.5/.github/workflows/test.yml
nagios-4.5.5/.gitignore
nagios-4.5.5/CONTRIBUTING.md
```

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

tar zxvf nagios-4.5.5.tar.gz

```
[ec2-user@ip-172-31-91-91 downloads]$ tar zxvf nagios-4.0.8.tar.gz
nagios-4.0.8/
nagios-4.0.8/.gitignore
nagios-4.0.8/Changelog
nagios-4.0.8/INSTALLING
nagios-4.0.8/LEGAL
nagios-4.0.8/LICENSE
nagios-4.0.8/Makefile.in
nagios-4.0.8/README
nagios-4.0.8/README.asciidoc
nagios-4.0.8/THANKS
nagios-4.0.8/UPGRADING
nagios-4.0.8/base/
nagios-4.0.8/base/.gitignore
```

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

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

Here we get an error

```
[ec2-user@ip-172-31-34-108 downloads]$ ./configure --with-command-group=nagcmd
-bash: ./configure: No such file or directory
```

Solution

Navigate to nagios folder in downloads

cd nagios-4.5.5

```
[ec2-user@ip-172-31-34-108 downloads]$ cd nagios-4.5.5
```

Error 2: Cannot find SSL headers.

Solution: Install openssl dev library

Steps:

sudo yum install openssl-devel

```
[ec2-user@ip-172-31-91-91 nagios-4.5.5]$ sudo yum install openssl-devel
Last metadata expiration check: 2:24:05 ago on Sun Sep 29 06:56:15 2024.
Dependencies resolved.
=====
 Package           Arch      Version            Repository      Size
 =====
 Installing:
  openssl-devel    x86_64    1:3.0.8-1.amzn2023.0.14   amazonlinux    3.0 M

Transaction Summary
=====
 Install 1 Package

Total download size: 3.0 M
Installed size: 4.7 M
Is this ok [y/N]: y
Downloading Packages:
```

Now run

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

```
[ec2-user@ip-172-31-34-108 nagios-4.5.5]$ ./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether the compiler supports GNU C... yes
checking whether gcc accepts -g... yes
checking for gcc option to enable C11 features... none needed
checking whether make sets $(MAKE)... yes
checking whether ln -s works... yes
checking for strip... /usr/bin/strip
checking for sys/wait.h that is POSIX.1 compatible... yes
```

12. Compile the source code.

make all

```
[ec2-user@ip-172-31-34-108 nagios-4.5.5]$ make all
cd ./base && make
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_
CONFIG_H -DNSCORE -c -o nagios.o ./nagios.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_
CONFIG_H -DNSCORE -c -o broker.o broker.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_
CONFIG_H -DNSCORE -c -o nebmods.o nebmods.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_
CONFIG_H -DNSCORE -c -o ..//common/shared.o ..//common/shared.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_
CONFIG_H -DNSCORE -c -o query-handler.o query-handler.c
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_
CONFIG_H -DNSCORE -c -o workers.o workers.c
In function 'get_wproc_list',
  inlined from 'get_worker' at workers.c:277:12:
workers.c:253:17: warning: '%s' directive argument is null [-Wformat-overflo
w=]
  253 |           log_debug_info(DEBUGL_CHECKS, 1, "Found specialized
worker(s) for '%s'", (slash && *slash != '/') ? slash : cmd_name);
          |
~~~~~
gcc -Wall -I.. -I.. -I./lib -I./include -I./include -I.. -g -O2 -DHAVE_
CONFIG_H -DNSCORE -c -o checks.o checks.c
```

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

sudo make install

sudo make install-init

sudo make install-config

sudo make install-commandmode

```
[ec2-user@ip-172-31-34-108 nagios-4.5.5]$ sudo make install
sudo make install-init
sudo make install-config
sudo make install-commandmode
cd ./base && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagios /usr/local/nagios/b
in
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagiosstats /usr/local/nagi
os/bin
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/base'
cd ./cgi && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
make install-basic
make[2]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/sbin
for file in *.cgi; do \
    /usr/bin/install -c -s -m 775 -o nagios -g nagios $file /usr/local/n
agios/sbin; \
done
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
cd ./html && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/html'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/me
dia
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/st
ylesheets
```

14. Edit the config file and change the email address.

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

```
[ec2-user@ip-172-31-34-108 nagios-4.5.5]$ sudo nano /usr/local/nagios/etc/ob
jects/contacts.cfg
```

```

BNF nmc 5.5
=====
CONTACTS.CFG - SAMPLE CONTACT/CONTACTGROUP DEFINITIONS
=====

# NOTES: This config file provides you with some example contact and contact
# group definitions that you can reference in host and service
# definitions.

# You don't need to keep these definitions in a separate file from your
# other object definitions. This has been done just to make things
# easier to understand.

=====
# CONTACTS
=====

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

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

}

=====
[Help   Write Out Where Is Cut Undo Read 54 lines Set Mark To Bracket Previous Back Prev Word Home End Prev Line Next Line
X Exit Replace Paste Execute Justify Location Go To Line Redo M-d Copy Where Was M-n Next Forward Next Word M-s End Next Line]

```

In this page, change email address:

```

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.brijesh.sharma@ves.ac.in ; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS *****

}

```

15. Configure the web interface.

sudo make install-webconf

```

[ec2-user@ip-172-31-34-108 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 ***

```

16. Create a nagiosadmin account for nagios login along with password. You'll have to specify the password twice.

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

```

[ec2-user@ip-172-31-34-108 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

```

Password: *hello123*

17. Restart Apache

```
sudo service httpd restart
```

```
[ec2-user@ip-172-31-34-108 nagios-4.5.5]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
```

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

```
cd ~/downloads
```

```
tar zxvf nagios-plugins-2.4.11.tar.gz
```

```
[ec2-user@ip-172-31-34-108 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
nagios-plugins-2.4.11/build-aux/snippet/arg-nonnull.h
nagios-plugins-2.4.11/build-aux/snippet/c++defs.h
nagios-plugins-2.4.11/build-aux/snippet/warn-on-use.h
nagios-plugins-2.4.11/build-aux/test-driver
nagios-plugins-2.4.11/config_test/
nagios-plugins-2.4.11/config_test/Makefile
nagios-plugins-2.4.11/config_test/run_tests
nagios-plugins-2.4.11/config_test/child_test.c
```

19. Compile and install plugins

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

```
./configure --with-nagios-user=nagios --with-nagios-group=nagios
```

```
[ec2-user@ip-172-31-34-108 downloads]$ cd nagios-plugins-2.4.11
./configure --with-nagios-user=nagios --with-nagios-group=nagios
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether to enable maintainer-specific portions of Makefiles... yes
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether gcc understands -c and -o together... yes
checking whether make supports the include directive... yes (GNU style)
```

make

sudo make install

```
[ec2-user@ip-172-31-34-108 nagios-plugins-2.4.11]$ make
sudo make install
make all-recursive
make[1]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
Making all in gl
make[2]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/
gl'
rm -f alloca.h-t alloca.h && \
{ echo '/* DO NOT EDIT! GENERATED AUTOMATICALLY! */'; \
cat ./alloca.in.h; \
} > alloca.h-t && \
mv -f alloca.h-t alloca.h
rm -f c++defs.h-t c++defs.h && \
sed -n -e '/_GL_CXXDEFS/, $p' \
< ../build-aux/snippet/c++defs.h \
> c++defs.h-t && \
mv c++defs.h-t c++defs.h
rm -f warn-on-use.h-t warn-on-use.h && \
sed -n -e '/^.ifndef/.$p' \
```

```

fi
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11/p
o'
make[1]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[2]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.4.11'

```

20. Start Nagios

Add Nagios to the list of system services

sudo chkconfig --add nagios

sudo chkconfig nagios on

```

[ec2-user@ip-172-31-34-108 nagios-plugins-2.4.11]$ sudo chkconfig --add nagi
os
sudo chkconfig nagios on
error reading information on service nagios: No such file or directory
Note: Forwarding request to 'systemctl enable nagios.service'.
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service →
/usr/lib/systemd/system/nagios.service.

```

Verify the sample configuration files

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

```

[ec2-user@ip-172-31-34-108 nagios-plugins-2.4.11]$ sudo /usr/local/nagios/bin/nagios -v /us
r/local/nagios/etc/nagios.cfg

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

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

Running pre-flight check on configuration data...

Checking objects...
  Checked 8 services.
  Checked 1 hosts.
  Checked 1 host groups.
  Checked 0 service groups.
  Checked 1 contacts.
  Checked 1 contact groups.
  Checked 24 commands.
  Checked 5 time periods.
  Checked 0 host escalations.
  Checked 0 service escalations.
Checking for circular paths...
  Checked 1 hosts
  Checked 0 service dependencies
  Checked 0 host dependencies
  Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors:  0

```

sudo service nagios start

```
[ec2-user@ip-172-31-34-108 nagios-plugins-2.4.11]$ sudo service nagios startRedirecting to  
/bin/systemctl start nagios.service
```

21. Check the status of Nagios

sudo systemctl status nagios

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

EC2 > Instances > i-085e568830b418e61

Instance summary for i-085e568830b418e61 (nagios-host) [Info](#)

Updated less than a minute ago

Instance ID i-085e568830b418e61 (nagios-host)	Public IPv4 address 54.161.62.217 [open address]	Private IPv4 addresses 172.31.34.108
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-54-161-62-217.compute-1.amazonaws.com [open address]
Hostname type IP name: ip-172-31-34-108.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-34-108.ec2.internal	Elastic IP addresses -
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. Learn more
Auto-assigned IP address 54.161.62.217 [Public IP]	VPC ID vpc-0531204c9e29f6332	Auto Scaling Group name -
IAM Role -	Subnet ID subnet-02833a98c4631f55d	Instance ARN arn:aws:ec2:us-east-1:787881940593:instance/i-085e568830b418e61
IMDSv2 Required		

23. Open up your browser and look for http://<your_public_ip_address>/nagios

Enter username as nagiosadmin and password which you set in Step 16.

Sign in

http://54.162.128.119
Your connection to this site is not private

Username:

Password:

[Cancel](#) [Sign In](#)

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

Not secure 54.161.62.217/nagios/

Nagios® Core™ Version 4.5.5

September 17, 2024

Check for updates

Get Started

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

Latest News

Don't Miss...

Quick Links

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

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

Nagios Core is licensed under the GNU General Public License and is provided AS IS WITH NO WARRANTY OF ANY KIND, INCLUDING THE WARRANTY OF DESIGN, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. Nagios, Nagios Core and the Nagios logo are trademarks, servicemarks, registered trademarks

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

Conclusion:

In this practical, we successfully installed and configured Nagios Core along with Nagios plugins and NRPE on an Amazon EC2 instance. We created a Nagios user, set up necessary permissions, and resolved common installation errors. Finally, we verified the setup by accessing the Nagios web interface, confirming that our monitoring system was fully operational.

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

Theory:

Port and Service Monitoring

Port and service monitoring in Nagios involves checking the availability and responsiveness of network services running on specific ports. This ensures that critical services (like HTTP, FTP, or SSH) are operational. Nagios uses plugins to ping the ports and verify whether services are up and responding as expected, allowing administrators to be alerted in case of outages.

Windows/Linux Server Monitoring

Windows/Linux server monitoring with Nagios entails tracking the performance and health of servers running these operating systems. It includes monitoring metrics such as CPU usage, memory consumption, disk space, and system logs. Nagios employs various plugins to gather data, enabling administrators to ensure optimal performance, identify potential issues, and maintain uptime across their server infrastructure.

Prerequisites:

AWS Academy or Personal account.

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

Monitoring Using Nagios:

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

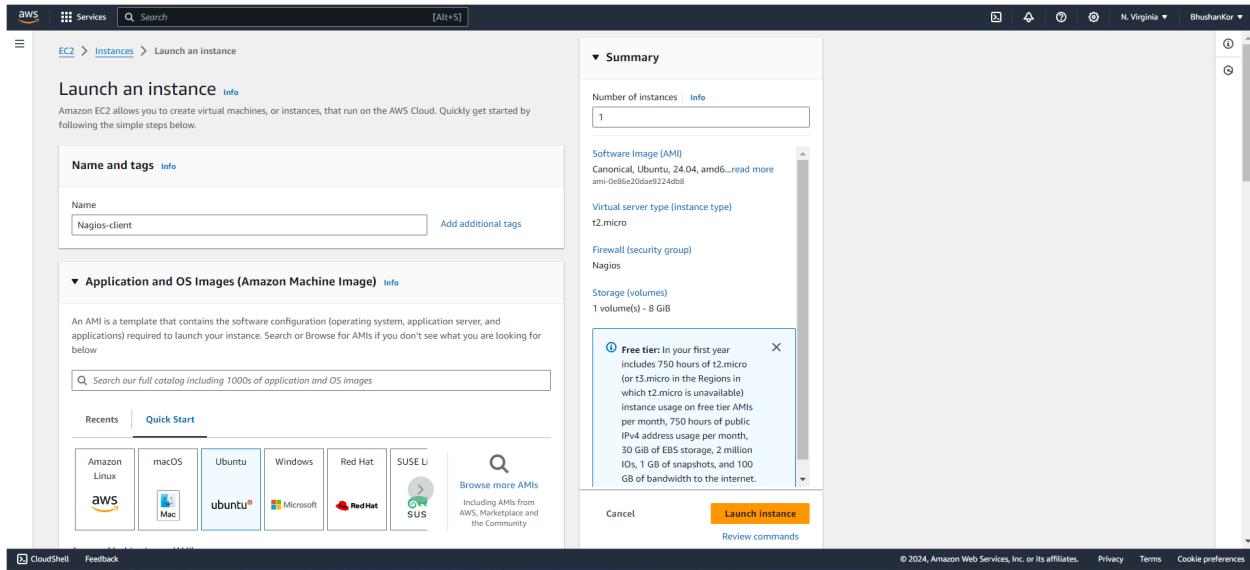
sudo systemctl status nagios

You can now proceed if you get the above message/output.

Step 2: Now Create a new EC2 instance. Name: Nagios-client, AMI: Ubuntu Instance Type: t2.micro.

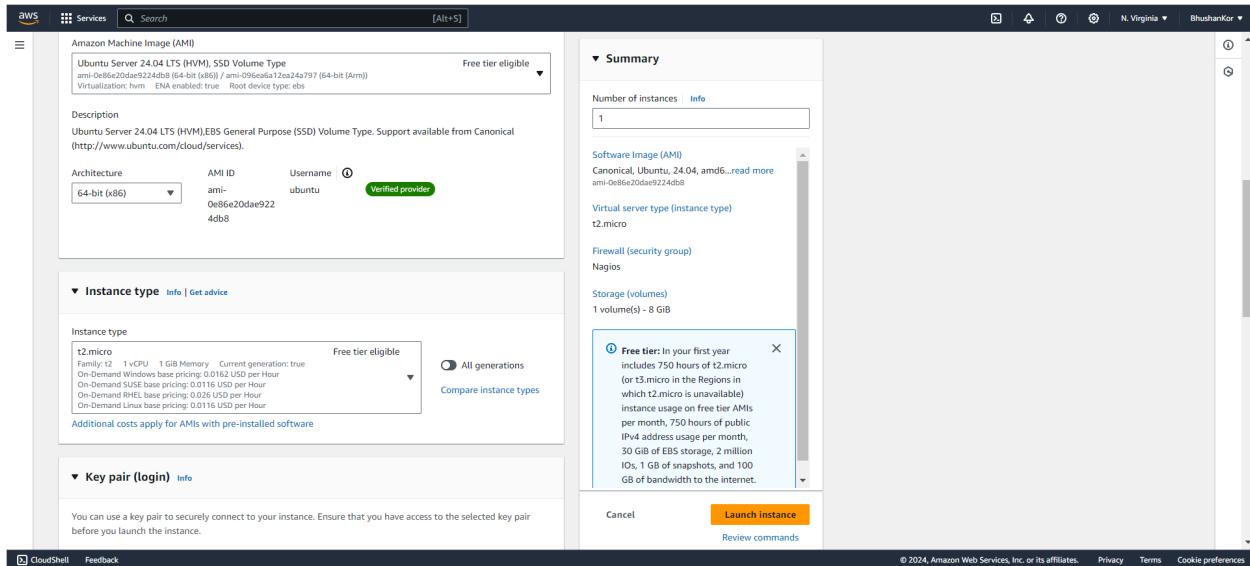
```
● nagios.service - Nagios Core 4.5.5
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
   Active: active (running) since Fri 2024-10-04 14:11:55 UTC; 28min ago
     Docs: https://www.nagios.org/documentation
 Main PID: 1998 (nagios)
    Tasks: 6 (limit: 1112)
   Memory: 6.7M
      CPU: 442ms
 CGroup: /system.slice/nagios.service
         └─1998 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
             ├─2004 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─2005 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─2006 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             ├─2007 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
             └─2008 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 04 14:11:55 ip-172-31-34-108.ec2.internal nagios[1998]: wproc: Registry request: name=Core Worker>
Oct 04 14:11:55 ip-172-31-34-108.ec2.internal nagios[1998]: Successfully launched command file worker>
Oct 04 14:13:47 ip-172-31-34-108.ec2.internal nagios[1998]: SERVICE ALERT: localhost;HTTP;CRITICAL;HTTP
Oct 04 14:16:17 ip-172-31-34-108.ec2.internal nagios[1998]: SERVICE NOTIFICATION: nagiosadmin;localhost;HTTP
Oct 04 14:16:17 ip-172-31-34-108.ec2.internal nagios[1998]: wproc: NOTIFY job 2 from worker Core Work...
Oct 04 14:16:17 ip-172-31-34-108.ec2.internal nagios[1998]: wproc: host=localhost; service=Swap Us...
Oct 04 14:16:17 ip-172-31-34-108.ec2.internal nagios[1998]: wproc: early_timeout=0; exited_ok=1; wa...
Oct 04 14:16:17 ip-172-31-34-108.ec2.internal nagios[1998]: stderr line 01: /bin/sh: line 1:>
Oct 04 14:16:17 ip-172-31-34-108.ec2.internal nagios[1998]: stderr line 02: /usr/bin/printf:>
Oct 04 14:18:47 ip-172-31-34-108.ec2.internal nagios[1998]: SERVICE ALERT: localhost;HTTP;WARNING;HTTP
lines 1-26/26 (END)
```

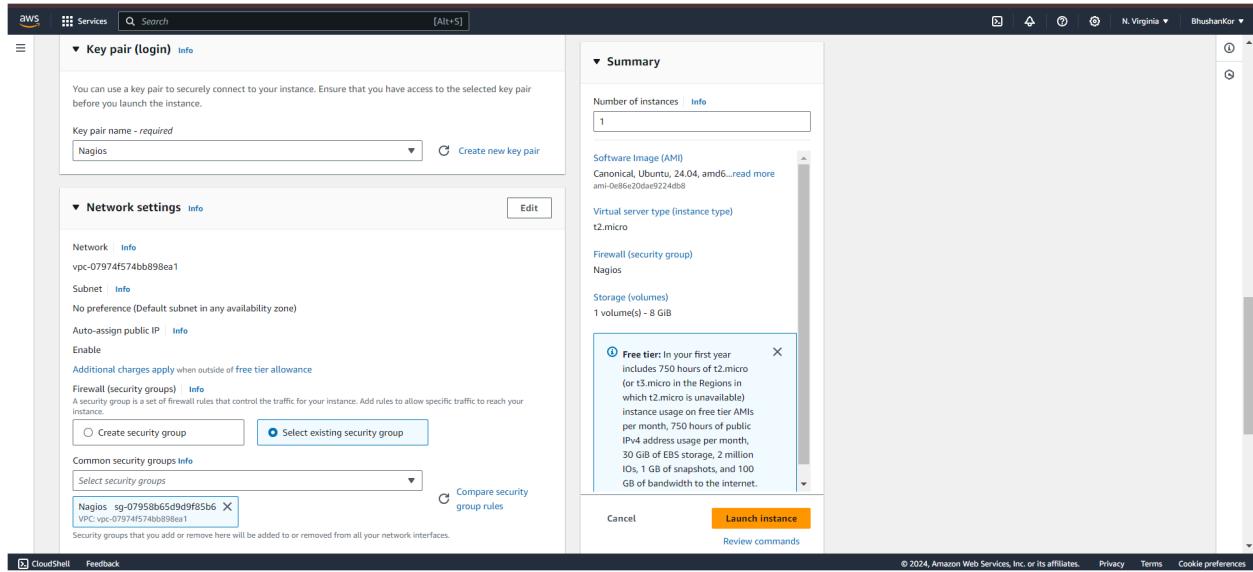


For Key pair : Click on create key and make key of type RSA with extension .pem . Key will be downloaded to your local machine.

Now select that key in key pair if you already have key with type RSA and extension .pem no need to create new key but you must have that key downloaded.



Select the Existing Security Group and select the Security Group that we have created in Experiment no 9 or the same one you have used for the Nagios server (Nagios-host).



Step 3: Now After creating the EC2 Instance click on connect and then copy the command which is given as example in the SSH Client section .

Now open the terminal in the folder where your key(RSA key with .pem) is located. and paste that copied command.

Successfully connected to the instance.

```
PS C:\Users\hp\Downloads\lab09> ssh -i "lab09.pem" ec2-user@ec2-54-172-217-167.compute-1.amazonaws.com
The authenticity of host 'ec2-54-172-217-167.compute-1.amazonaws.com (54.172.217.167)' can't be established.
ED25519 key fingerprint is SHA256:LBNeCYQvwiUAFbRZ4UavnvY9vHuBhVKS/K3DVKJ9Jn4.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-172-217-167.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

      _#
     ~\_###_      Amazon Linux 2023
     ~~\####\_
     ~~ \###|
     ~~   \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023
     ~~   V~' '-->
     ~~~   /
     ~~ ._. /_
     ~~ /'_ /_
     ~~ /m/ '
[ec2-user@ip-172-31-38-4 ~]$ |
```

Now perform all the commands on the Nagios-host till step 10

Step 4: Now on the server Nagios-host run the following command.

ps -ef | grep nagios

```
[ec2-user@ip-172-31-34-108 ~]$ ps -ef | grep nagios
nagios      1998      1  0 14:11 ?    00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios      2004      1998  0 14:11 ?    00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios      2005      1998  0 14:11 ?    00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios      2006      1998  0 14:11 ?    00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios      2007      1998  0 14:11 ?    00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios      2008      1998  0 14:11 ?    00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
root       16967     2370  0 14:20 pts/0  00:00:00 sudo systemctl status nagios
root       16970     16967  0 14:20 pts/1  00:00:00 sudo systemctl status nagios
root       16971     16970  0 14:20 pts/1  00:00:00 systemctl status nagios
root       18221     18132  0 14:40 pts/2  00:00:00 sudo systemctl status nagios
root       18223     18221  0 14:40 pts/3  00:00:00 sudo systemctl status nagios
root       18224     18223  0 14:40 pts/3  00:00:00 systemctl status nagios
ec2-user   19275     19251  0 14:59 pts/4  00:00:00 grep --color=auto nagios
```

Step 5: Now Become root user and create root directories.

sudo su

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

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

```
[ec2-user@ip-172-31-34-108 ~]$ sudo su
mkdir /usr/local/nagios/etc/objects/monitorhosts
mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-34-108 ec2-user]# |
```

```
[root@ip-172-31-34-108 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts
[root@ip-172-31-34-108 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
```

Step 6: Copy the sample localhost.cfg to linuxhost.cfg by running the following command.(**Below command should come in one line see screenshot below**)

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

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

```
[root@ip-172-31-34-108 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
[root@ip-172-31-34-108 ec2-user]# |
```

Step 7:Open linuxserver.cfg using nano and make the following changes in all positions?everywhere in file.

Change **hostname** to **linuxserver**.

Change **address** to the public IP of your Linux client.

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

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

```
define host {  
    use          linux-server      ; Name of host template to use  
    ; This host definition will inherit  
    ; from (or be inherited by) the linux-s>  
    host_name    linuxserver  
    alias        localhost  
    address     54.172.217.167  
}  
  
define hostgroup {  
    hostgroup_name  linux-servers1 ; The name of the hostgroup  
    alias           Linux Servers   ; Long name of the group  
    members         localhost       ; Comma separated list of hosts th>
```

Step 8: Now update the Nagios config file .Add the following line in the file.

Line to add : `cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/`

Run the command : `nano /usr/local/nagios/etc/nagios.cfg`

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

Step 9: Now Verify the configuration files by running the following commands.

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

```
[root@ip-172-31-34-108 ec2-user]# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

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

Website: https://www.nagios.org
Reading configuration data...
    Read main config file okay...
Warning: Duplicate definition found for service 'HTTP' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', starting on line 152)
Warning: Duplicate definition found for service 'SSH' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', starting on line 138)
Warning: Duplicate definition found for service 'Swap Usage' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', starting on line 125)
Warning: Duplicate definition found for service 'Current Load' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', starting on line 112)
Warning: Duplicate definition found for service 'Total Processes' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', starting on line 100)
Warning: Duplicate definition found for service 'Current Users' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', starting on line 86)
Warning: Duplicate definition found for service 'Root Partition' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', starting on line 72)
Warning: Duplicate definition found for service 'PING' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', starting on line 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
```

Step 10: Now restart the services of nagios by running the following command.
service nagios restart

```
[root@ip-172-31-34-108 ec2-user]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
[root@ip-172-31-34-108 ec2-user]# |
```

Step 11: Now Go to the Nagios-client ssh terminal and update and install the packages by running the following command.

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

```
ubuntu@ip-172-31-47-124:~$ 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://security.ubuntu.com/ubuntu noble-security/main Translation-en [83.9 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4704 B]
Get:9 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [277 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [117 kB]
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [8632 B]
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [10.4 kB]
Get:14 http://security.ubuntu.com/ubuntu noble-security/multiverse
```

Step 12: Open nrpe.cfg file to make changes.Under allowed_hosts, add your nagios host IP address.

```
GNU nano 7.2          /etc/nagios/nrpe.cfg
#
# NOTE: This option is ignored if NRPE is running under either inetrn>
nrpe_user=nagios

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

#
# ALLOWED HOST ADDRESSES
# This is an optional comma-delimited list of IP address or hostna>
# that are allowed to talk to the NRPE daemon. Network addresses w>
# (i.e. 192.168.1.0/24) are also supported. Hostname wildcards are>
# supported.
#
# Note: The daemon only does rudimentary checking of the client's >
# address. I would highly recommend adding entries in your /etc/h>
# 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 inetrn>
allowed_hosts=127.0.0.1,::1,54.161.62.217
```

Step 13: Now restart the NRPE server by this command.

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

```
ubuntu@ip-172-31-47-124:~$ sudo systemctl restart nagios-nrpe-server
r
```

Step 14: Now again check the status of Nagios by running this command on Nagios-host and also check httpd is active and run the command to active it.

```
sudo systemctl status nagios
```

```
[ec2-user@ip-172-31-34-108 ~]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
  Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
  Active: active (running) since Fri 2024-10-04 15:39:38 UTC; 14min ago
    Docs: https://www.nagios.org/documentation
 Process: 21775 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg >
 Process: 21776 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (co>
 Main PID: 21781 (nagios)
   Tasks: 6 (limit: 1112)
  Memory: 4.1M
     CPU: 244ms
    CGroup: /system.slice/nagios.service
            └─21781 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
              ├─21782 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─21783 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─21784 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─21785 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              └─21790 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 04 15:39:38 ip-172-31-34-108.ec2.internal nagios[21781]: Warning: Duplicate definition found >
Oct 04 15:39:38 ip-172-31-34-108.ec2.internal nagios[21781]: Warning: Duplicate definition found >
Oct 04 15:39:38 ip-172-31-34-108.ec2.internal nagios[21781]: Warning: Duplicate definition found >
Oct 04 15:39:38 ip-172-31-34-108.ec2.internal nagios[21781]: Warning: Duplicate definition found >
Oct 04 15:39:38 ip-172-31-34-108.ec2.internal nagios[21781]: Warning: Duplicate definition found >
Oct 04 15:39:38 ip-172-31-34-108.ec2.internal nagios[21781]: Warning: Duplicate definition found >
Oct 04 15:39:38 ip-172-31-34-108.ec2.internal nagios[21781]: Warning: Duplicate definition found >
Oct 04 15:39:38 ip-172-31-34-108.ec2.internal nagios[21781]: Warning: Duplicate definition found >
Oct 04 15:39:38 ip-172-31-34-108.ec2.internal nagios[21781]: Warning: Duplicate definition found >
Oct 04 15:39:38 ip-172-31-34-108.ec2.internal nagios[21781]: Warning: Duplicate definition found >
Oct 04 15:39:38 ip-172-31-34-108.ec2.internal nagios[21781]: Warning: Duplicate definition found >
Oct 04 15:39:38 ip-172-31-34-108.ec2.internal nagios[21781]: Warning: Duplicate definition found >
Oct 04 15:39:38 ip-172-31-34-108.ec2.internal nagios[21781]: Successfully launched command file w>
Oct 04 15:40:20 ip-172-31-34-108.ec2.internal nagios[21781]: HOST ALERT: linuxserver;UP;SOFT;1;PI>
Lines 1-28/28 (END)
```

sudo systemctl status httpd

```
[ec2-user@ip-172-31-34-108 ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
  Drop-In: /usr/lib/systemd/system/httpd.service.d
            └─php-fpm.conf
  Active: active (running) since Fri 2024-10-04 14:18:18 UTC; 1h 37min ago
    Docs: man:httpd.service(8)
 Main PID: 2495 (httpd)
   Status: "Total requests: 48; Idle/Busy workers 100/0;Requests/sec: 0.0082; Bytes served/sec:<
  Tasks: 230 (limit: 1112)
 Memory: 24.0M
     CPU: 3.655s
    CGroup: /system.slice/httpd.service
            └─ 2495 /usr/sbin/httpd -DFOREGROUND
              ├─ 2543 /usr/sbin/httpd -DFOREGROUND
              ├─ 2544 /usr/sbin/httpd -DFOREGROUND
              ├─ 2545 /usr/sbin/httpd -DFOREGROUND
              ├─ 2546 /usr/sbin/httpd -DFOREGROUND
              └─ 20138 /usr/sbin/httpd -DFOREGROUND

Oct 04 14:18:18 ip-172-31-34-108.ec2.internal systemd[1]: Starting httpd.service - The Apache HTTP>
Oct 04 14:18:18 ip-172-31-34-108.ec2.internal systemd[1]: Started httpd.service - The Apache HTTP>
Oct 04 14:18:18 ip-172-31-34-108.ec2.internal httpd[2495]: Server configured, listening on: port >
Lines 1-22/22 (END)
```

sudo systemctl start httpd

sudo systemctl enable httpd

Step 15: Now to check Nagios dashboard go to <http://<Nagios-host ip>/nagios> .

The screenshot shows the Nagios Core 4.5.5 dashboard. On the left, there's a sidebar with links for General, Current Status, Reports, and System. The main area features the Nagios Core logo and a message that the daemon is running with PID 21781. It also displays the version (Version 4.5.5), the date (September 17, 2024), and a link to check for updates. Below this are sections for Get Started (with bullet points about monitoring, changing look, and adding addons), Latest News (empty), and Don't Miss... (empty). A Quick Links sidebar on the right lists various Nagios resources like the library, labs, and support. At the bottom, there are copyright notices and a license statement.

Now Click on Hosts from left side panel

We can see our linuxserver now click on it we can see the host information.

Current Network Status

This screenshot shows the 'Current Network Status' section of the Nagios Core 4.5.5 dashboard. The left sidebar includes links for General, Current Status, and Reports. The main content area displays 'Current Network Status' with a last update timestamp of Fri Oct 4 15:58:07 UTC 2024. It shows two hosts: 'linuxserver' (UP) and 'localhost' (UP). Below this, a table provides detailed host status totals and service status totals. A large table titled 'Host Status Details For All Host Groups' lists the two hosts with their status, last check time, duration, and status information. The table includes columns for Host, Status, Last Check, Duration, and Status Information.

Host	Status	Last Check	Duration	Status Information
linuxserver	UP	10-04-2024 15:55:16	0d 0h 17m 51s	PING OK - Packet loss = 0%, RTA = 1.31 ms
localhost	UP	10-04-2024 15:53:47	0d 11h 10m 6s	PING OK - Packet loss = 0%, RTA = 0.03 ms

Host Information

Last Updated: Fri Oct 4 15:58:45 UTC 2024
 Updated every 90 seconds
 Nagios® Core™ 4.5.5 - www.nagios.org
 Logged in as nagiosadmin

[View Status Detail For This Host](#)
[View Alert History For This Host](#)
[View Trends For This Host](#)
[View Alert Histogram For This Host](#)
[View Availability Report For This Host](#)
[View Notifications For This Host](#)

Host
localhost
 (linuxserver)

Member of
No hostgroups

34.207.171.220

Host State Information

Host Status:	UP (for 0d 0h 18m 29s)
Status Information:	PING OK - Packet loss = 0%, RTA = 1.31 ms
Performance Data:	rtt=1.314000ms;3000.000000;5000.000000;0.000000;pl=0%;90;100;0
Current Attempt:	1/10 (HARD state)
Last Check Time:	10-04-2024 15:55:16
Check Type:	ACTIVE
Check Latency / Duration:	0.002 / 4.012 seconds
Next Scheduled Active Check:	10-04-2024 16:00:16
Last State Change:	10-04-2024 15:40:16
Last Notification:	N/A (notification 0)
Is This Host Flapping?	NO (10.79% state change)
In Scheduled Downtime?	NO
Last Update:	10-04-2024 15:58:37 (0d 0h 0m 8s ago)
Active Checks:	ENABLED
Passive Checks:	ENABLED
Obsessing:	ENABLED
Notifications:	ENABLED
Event Handler:	ENABLED
Flap Detection:	ENABLED

Host Commands

- Locate host on map
- Disable active checks of this host
- Re-schedule the next check of this host
- Submit passive check result for this host
- Stop accepting passive checks for this host
- Stop obsessing over this host
- 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](#)

Entry Time	Author	Comment	Comment ID	Persistent	Type	Expires	Actions
This host has no comments associated with it.							

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

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

Theory:

AWS Lambda

A fully managed, serverless computing service where you run code without provisioning or managing servers. Lambda automatically scales your application based on the number of incoming requests or events, ensuring efficient resource utilization. You are only charged for the time your code is running, with no upfront cost, making it cost-effective for on-demand workloads.

Lambda Workflow

- **Create a Function:** Write the function code and define its handler (entry point). You can use the AWS Console, CLI, or upload a deployment package.
- **Set Event Sources:** Define how the function is triggered (e.g., when an object is uploaded to S3 or a DynamoDB table is updated).
- **Execution:** When triggered, Lambda runs your function, executes the logic, and automatically scales to handle the incoming event volume.
- **Scaling and Concurrency:** Lambda scales automatically by launching more instances of the function to handle simultaneous invocations. There are also options for configuring **reserved concurrency** to manage traffic.
- **Monitoring and Logging:** Lambda integrates with Amazon CloudWatch for logging and monitoring. Logs for each invocation are sent to CloudWatch, allowing you to track performance and troubleshoot errors.

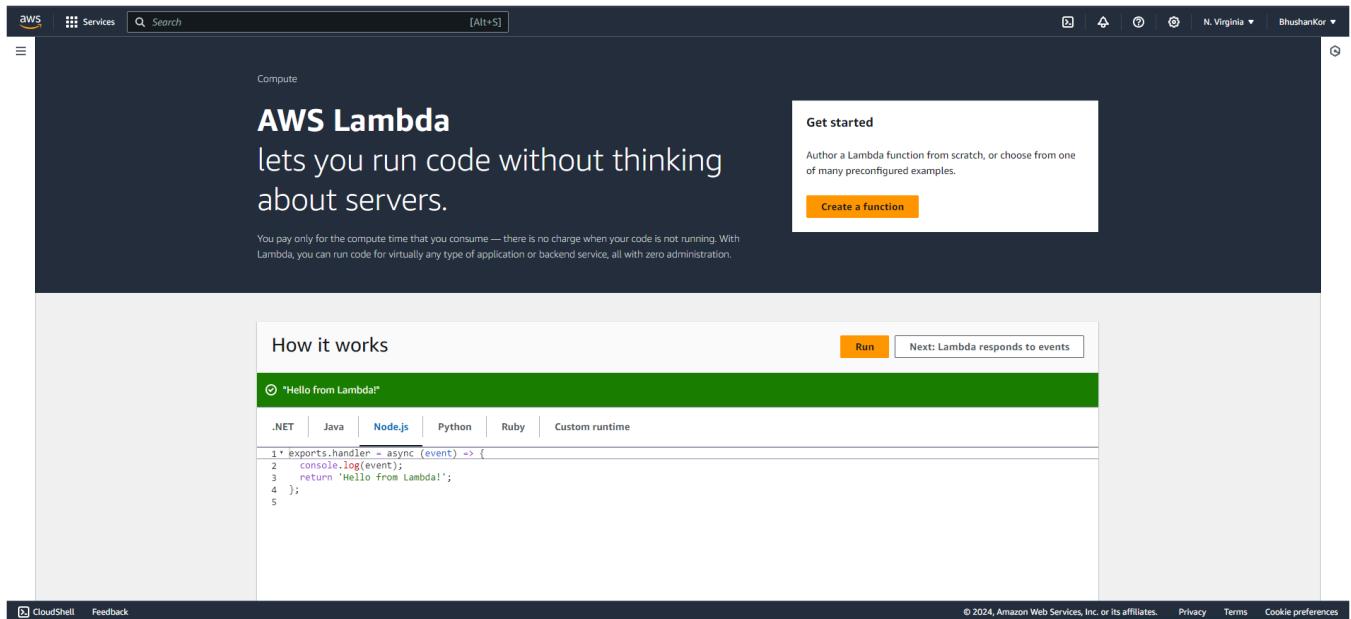
AWS Lambda Functions

- **Python:** Great for quick development with its rich standard library and support for lightweight tasks.
- **Java:** Typically used for more complex, compute-intensive tasks. While it's robust, cold start times can be higher.
- **Node.js:** Excellent for I/O-bound tasks like handling APIs or streaming data, with fast startup times and efficient memory usage.

Prerequisites: AWS Personal/Academy Account

Steps To create the lambda function:

Step 1: Login to your AWS Personal/Academy Account. Open Lambda and click on create function button.



Step 2: Now Give a name to your Lambda function, Select the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby. So will select Python 3.12, Architecture as x86, and Execution role to Create a new role with basic Lambda permissions.

The screenshot shows the 'Create function' wizard in progress. The 'Basic information' step is active. The form fields are as follows:

- Function name:** lambda_lab11
- Runtime:** Python 3.12
- Architecture:** x86_64

▼ Change default execution role

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console [IAM console](#).

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

Info Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in this role.

Lambda will create an execution role named lambda_lab11-role-6rec818x, with permission to upload logs to Amazon CloudWatch Logs.

▶ Additional Configurations
Use additional configurations to set up code signing, function URL, tags, and Amazon VPC access for your function.

Cancel **Create function**

Successfully created the function lambda_lab11. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Lambda > Functions > lambda_lab11

lambda_lab11

Throttle Copy ARN Actions ▾

▼ Function overview [Info](#)

Diagram Template

lambda_lab11

Layers (0)

+ Add trigger + Add destination

Description
-

Last modified
1 second ago

Function ARN
arn:aws:lambda:us-east-1:017820672175:function:lambda_lab11

Function URL [Info](#)
-

Export to Application Composer Download ▾

Successfully created the function lambda_lab11. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Code Test Monitor Configuration Aliases Versions

Code source [Info](#)

File Edit Find View Go Tools Window Test Deploy

Go to Anything (Ctrl-P)

Environment

lambda_lab11 /

lambda_function.py

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

Upload from ▾

1:1 Python Spaces: 4

So See or Edit the basic settings go to configuration then click on edit general setting.

The screenshot shows the AWS Lambda Configuration interface. The top navigation bar includes tabs for Code, Test, Monitor, Configuration (which is selected and underlined), Aliases, and Versions. On the left, a sidebar lists various configuration categories: Triggers, Permissions, Destinations, Function URL, Environment variables, Tags, VPC, RDS databases, Monitoring and operations tools, Concurrency and recursion detection, Asynchronous invocation, Code signing, File systems, and State machines. The main content area is titled "General configuration" and contains fields for Description (set to "-"), Memory (set to 128 MB), Ephemeral storage (set to 512 MB), Timeout (set to 0 min 3 sec), and SnapStart (set to None). An "Edit" button is located in the top right corner of this section.

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

The screenshot shows the AWS Lambda Basic settings page. It includes fields for Description (empty), Memory (128 MB), Ephemeral storage (512 MB), SnapStart (None), and Timeout (0 min 1 sec). Under Execution role, it shows "Use an existing role" selected with "Create a new role from AWS policy templates" as an option. Existing roles dropdown shows "service-role/lambda_lab11-role-6rec818x". At the bottom are "Cancel" and "Save" buttons.

Step 3: Now Click on the Test tab then select Create a new event, give a name to the event and select Event Sharing to private, and select hello-world template.

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

Test event action

Create new event Edit saved event

Event name

lab11Event

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

Event sharing settings

Private This event is only available in the Lambda console and to the event creator. You can configure a total of 10. Learn more

Shareable This event is available to IAM users within the same account who have permissions to access and use shareable events. Learn more

Template - optional

hello-world

Event JSON

```
1 [{"key1": "value1", "key2": "value2", "key3": "value3"}]
```

Format JSON

Step 4: Now In Code section select the created event from the dropdown of test then click on test . You will see the below output.

Code source

Upload from ▾

File Edit Find View Go Tools Window Test Deploy

Execution results

Test Event Name lambdaEvent

Status: Succeeded Max memory used: 32 MB Time: 2.07 ms

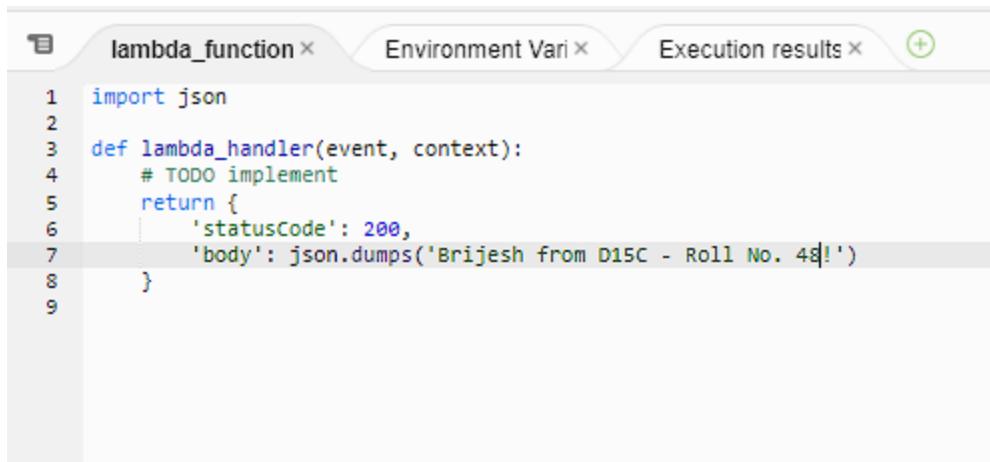
Response

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

Function Logs

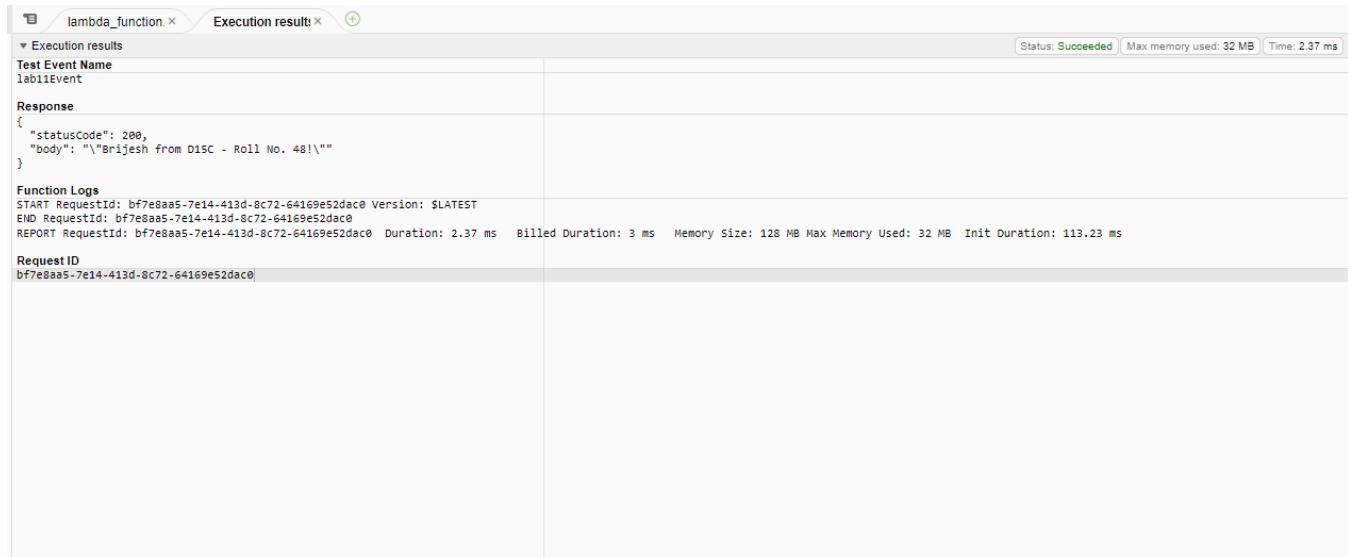
```
START RequestId: dc1eb5a-4c72-4c85-8694-d3c221c14fee Version: $LATEST
END RequestId: dc1eb5a-4c72-4c85-8694-d3c221c14fee
REPORT RequestId: dc1eb5a-4c72-4c85-8694-d3c221c14fee Duration: 2.07 ms Billed Duration: 3 ms Memory Size: 128 MB Max Memory Used: 32 MB Init Duration: 94.30 ms
RequestID
dc1eb5a-4c72-4c85-8694-d3c221c14fee
```

Step 5: You can edit your lambda function code. I have changed the code to display the new String. Now ctrl+s to save and click on deploy to deploy the changes.



```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     return {
6         'statusCode': 200,
7         'body': json.dumps('Brijesh from D15C - Roll No. 48!')}
8
9
```

Step 6: Now click on the test and observe the output. We can see the status code 200 and your string output and function logs. On successful deployment.



Execution results	Execution result
Test Event Name lambdaEvent	Status: Succeeded Max memory used: 32 MB Time: 2.37 ms
Response	{ "statusCode": 200, "body": "\"Brijesh from D15C - Roll No. 48!\"" }
Function Logs	START RequestId: bf7e8aa5-7e14-413d-8c72-64169e52dac0 Version: \$LATEST END RequestId: bf7e8aa5-7e14-413d-8c72-64169e52dac0 REPORT RequestId: bf7e8aa5-7e14-413d-8c72-64169e52dac0 Duration: 2.37 ms Billed Duration: 3 ms Memory Size: 128 MB Max Memory Used: 32 MB Init Duration: 113.23 ms
Request ID	bf7e8aa5-7e14-413d-8c72-64169e52dac0

Conclusion: In this experiment, we successfully created an AWS Lambda function and walked through its essential steps. After setting up the function with Python, we configured the basic settings, including adjusting the timeout to 1 second. We then created a test event, deployed the function, and validated the output. Additionally, we modified the Lambda function's code and redeployed it to observe the changes in real-time. This practical experience demonstrated the simplicity and flexibility of AWS Lambda in creating serverless applications, allowing you to focus on code while AWS manages the infrastructure and scaling.

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

Theory:

AWS Lambda and S3 Integration:

AWS Lambda allows you to execute code in response to various events, including those triggered by Amazon S3. When an object is added to an S3 bucket, it can trigger a Lambda function to execute, allowing for event-driven processing without managing servers.

Workflow:

1. Create an S3 Bucket:

- First, create an S3 bucket that will store the objects. This bucket will act as the trigger source for the Lambda function.

2. Create the Lambda Function:

- Set up a new Lambda function using AWS Lambda’s console. You can choose a runtime environment like Python, Node.js, or Java.
- Write code that logs a message like “An Image has been added” when triggered.

3. Set Up Permissions:

- Ensure that the Lambda function has the necessary permissions to access S3. You can do this by attaching an IAM role with policies that allow reading from the bucket and writing logs to CloudWatch.

4. Configure S3 Trigger:

- Link the S3 bucket to the Lambda function by setting up a trigger. Specify that the function should be triggered when an object is created in the bucket (e.g., when an image is uploaded).

5. Test the Setup:

- Upload an object (e.g., an image) to the S3 bucket to test the trigger. The Lambda function should execute and log the message “An Image has been added” in AWS CloudWatch Logs.

Prerequisites: AWS Personal Account

Steps To create the lambda function:

Step 1: Login to your AWS Personal account. Now open S3 from services and click on create S3 bucket.

The screenshot shows the AWS S3 service dashboard. On the left, there's a sidebar with options like Buckets, Access Grants, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, IAM Access Analyzer for S3, Block Public Access settings, Storage Lens, Dashboards, Storage Lens groups, AWS Organizations settings, Feature spotlight, and AWS Marketplace for S3. The main area is titled "Amazon S3" and shows a "General purpose buckets" section. It displays four buckets: "codepipeline-eu-north-1-823007647292", "codepipeline-us-east-1-934567252759", "elasticbeanstalk-eu-north-1-010928205712", and "elasticbeanstalk-us-east-1-010928205712". Each bucket entry includes its name, AWS Region (e.g., Europe (Stockholm) eu-north-1 or US East (N. Virginia) us-east-1), IAM Access Analyzer link, and Creation date. At the top right of the main area, there are buttons for "Create bucket", "Copy ARN", "Empty", and "Delete". Below the table, there's a search bar labeled "Find buckets by name".

Step 2: Now Give a name to the Bucket, select general purpose project and deselect the Block public access and keep other this to default.

The screenshot shows the "Create bucket" wizard. The first step, "General configuration", is active. It asks for the "Bucket name" (input field contains "lab12_bucket48") and the "AWS Region" (set to "US East (N. Virginia) us-east-1"). Under "Bucket type", the "General purpose" option is selected (radio button is checked). A tooltip explains it's recommended for most use cases and access patterns. The "Directory" option is also available but not selected. The second step, "Object Ownership", is shown below. It has two options: "ACLs disabled (recommended)" (selected, checked) and "ACLs enabled". A tooltip for "ACLs disabled" states that all objects in the bucket are owned by the current account and access is controlled by IAM policies. A tooltip for "ACLs enabled" states that objects can be owned by other accounts and access can be controlled by ACLs. At the bottom of the "Object Ownership" section, it says "Object Ownership Bucket owner enforced".

Screenshot of the AWS S3 Bucket Properties page under the 'Object Ownership' tab.

Object Ownership Info

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

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

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

Object Ownership
Bucket owner enforced

Block Public Access settings for this bucket

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

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

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

CloudShell Feedback

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

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.

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

⌚ Successfully created bucket "lab12bucket48"
To upload files and folders, or to configure additional bucket settings, choose [View details](#).

[Amazon S3](#) > Buckets

▶ Account snapshot - updated every 24 hours [All AWS Regions](#)
Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

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

General purpose buckets (2) [Info](#) All AWS Regions

Buckets are containers for data stored in S3.

Find buckets by name

Name	AWS Region
elasticbeanstalk-us-east-1-017820672175	US East (N. Virginia) us-east-1
lab12bucket48	US East (N. Virginia) us-east-1

Step 3: Open lambda console and click on create function button.

[AWS](#) [Services](#) [Search](#) [Alt+S]

N. Virginia BhushanKor ▾

Compute

AWS Lambda

lets you run code without thinking about servers.

You pay only for the compute time that you consume — there is no charge when your code is not running. With Lambda, you can run code for virtually any type of application or backend service, all with zero administration.

Get started
Author a Lambda function from scratch, or choose from one of many preconfigured examples.
[Create a function](#)

How it works

Run Next: Lambda responds to events

⌚ "Hello from Lambda!"

.NET | Java | **Node.js** | Python | Ruby | Custom runtime

```
1* exports.handler = async (event) => {
2  console.log(event);
3  return 'Hello from Lambda!';
4}
```

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

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

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 [Info](#)
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.
 [Change](#)

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

Permissions [Info](#)
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ Change default execution role

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

Successfully created the function lab12_48. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Lambda > Functions > lab12_48

lab12_48 [Throttle](#) [Copy ARN](#) [Actions ▾](#)

▼ Function overview [Info](#)

Diagram		Template	Export to Application Composer	Download ▾
 lab12_48		Layers (0)	Description Last modified 1 minute ago Function ARN arn:aws:lambda:us-east-1:017820672175:function:lab12_48 Function URL Info	
+ Add trigger			+ Add destination	

So See or Edit the basic settings go to configuration then click on edit general setting.

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 various configuration categories: Triggers, Permissions, Destinations, Function URL, Environment variables, Tags, VPC, RDS databases, Monitoring and operations tools, Concurrency and recursion detection, Asynchronous invocation, Code signing, File systems, and State machines. The main content area displays the 'General configuration' section with the following details:

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

An 'Edit' button is located in the top right corner of this section.

Edit basic settings

The 'Edit basic settings' dialog box contains the following fields:

- Basic settings [Info](#)**
- Description - optional**: A text input field.
- Memory [Info](#)**: A text input field set to 128 MB. Below it, a note states: "Your function is allocated CPU proportional to the memory configured." and "Set memory to between 128 MB and 10240 MB."
- Ephemeral storage [Info](#)**: A text input field set to 512 MB. Below it, a note states: "You can configure up to 10 GB of ephemeral storage (/tmp) for your function. [View pricing](#)" and "Set ephemeral storage (/tmp) to between 512 MB and 10240 MB."
- SnapStart [Info](#)**: A dropdown menu currently set to "None". Below it, a note states: "Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized. To evaluate whether your function code is resilient to snapshot operations, review the [SnapStart compatibility considerations](#)".
- Timeout**: A text input field showing "0 min 1 sec".
- Execution role**: A note stating "Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#)". Two radio buttons are present: "Use an existing role" (selected) and "Create a new role from AWS policy templates".
- Existing role**: A dropdown menu showing "service-role/lab12_48-role-3mhrlau0" and a "C" icon for creating a new role. Below it, a note states: "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." and "View the lab12_48-role-3mhrlau0 role [on the IAM console](#)".

At the bottom right are "Cancel" and "Save" buttons.

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.

Step 5: Now Click on the Test tab then select Create a new event, give a name to the event and select Event Sharing to private, and select s3 put template.

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

Test event action

- Create new event
- Edit saved event

Event name

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

Event JSON

```

3*   {
4    "eventVersion": "2.0",
5    "eventSource": "aws:s3",
6    "awsRegion": "us-east-1",
7    "eventTime": "1970-01-01T00:00:00Z",
8    "eventName": "ObjectCreated:Put",
9    "userIdentity": {
10      "principalId": "EXAMPLE"
11    },
12    "requestParameters": {
13      "sourceIPAddress": "127.0.0.1"
14    },
15    "responseElements": {
16      "x-amz-request-id": "EXAMPLE123456789",
17      "x-amz-id-2": "EXAMPLE123/5678abcdefghijklmabaisawesome/mnopqrstuvwxyzABCDEFGH"
18    },
19    "s3": {
20      "s3SchemaVersion": "1.0",
21      "configurationId": "testConfigRule"
22    }
  
```

Format JSON

Step 6: Now In Code section select the created event from the dropdown .

Code | Test | Monitor | Configuration | Aliases | Versions

Code source | Info

File Edit Find View Go Tools Window Test Deploy

Configure test event Ctrl-Shift-C

lambda_function

```

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     }
  
```

Step 7: Now In the Lambda function click on add tigger.Now select the source as S3 then select the bucket name from the dropdown, keep other things to default and also you can add prefix to image

lab12_48

Function overview | Info

Diagram | Template

Throttle | Copy ARN | Actions

Export to Application Composer | Download

Description

Last modified 7 minutes ago

Function ARN arn:aws:lambda:us-east-1:017820672175:function:lab12_48

Function URL | Info

+ Add trigger

+ Add destination

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/lab12bucket48

Bucket region: us-east-1

Event types

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

All object create events

Prefix - optional

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

 image

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.

 e.g. .jpg

Recursive invocation

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

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

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

Cancel

Add

Step 8: Now Write code that logs a message like “An Image has been added” when triggered. Save the file and click on deploy.

Step 9: Now upload any image to the bucket.

The image shows three screenshots illustrating the process of deploying a Lambda function and uploading an image to an S3 bucket.

Screenshot 1: Lambda Function Configuration

A screenshot of the AWS Lambda function configuration page. The code editor contains the following Python code:

```

1 import json
2
3 def lambda_handler(event, context):
4     # Extract the bucket name and object key from the S3 event
5     bucket_name = event['Records'][0]['s3']['bucket']['name']
6     object_key = event['Records'][0]['s3']['object']['key']
7
8     print(f"An image was uploaded to bucket {bucket_name}:{object_key}")
9
10    return {
11        'statusCode': 200,
12        'body': json.dumps('Log entry created successfully!')
13    }
14

```

The "Configuration" tab is selected, showing the triggers section with one S3 trigger named "S3: lab12bucket48".

Screenshot 2: S3 Bucket Upload Interface

A screenshot of the AWS S3 "Upload" interface. It shows a file named "about.jpg" being uploaded to the bucket "s3://lab12bucket48". The "Destination" field also displays "s3://lab12bucket48".

Upload: status

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

Summary		Failed
Destination s3://lab12bucket48	Succeeded 1 file, 30.4 KB (100.00%)	0 files, 0 B (0%)

Files and folders Configuration

Files and folders (1 Total, 30.4 KB)

Name	Folder	Type	Size	Status	Error
about.jpg	-	image/jpeg	30.4 KB	Succeeded	-

Step 10: Now to click on test in lambda to check whether it is giving log when image is added to S3.

Execution results

Test Event Name lab12event

Response

```
{
  "statusCode": 200,
  "body": "\"Log entry created successfully!\""
}
```

Function Logs

```
START RequestId: d4539d35-bc4d-4539-99c0-a4475259b020 Version: $LATEST
An image was uploaded to bucket example-bucket:test%2Fkey
END RequestId: d4539d35-bc4d-4539-99c0-a4475259b020
REPORT RequestId: d4539d35-bc4d-4539-99c0-a4475259b020 Duration: 2.07 ms Billed Duration: 3 ms Memory Size: 128 MB Max Memory Used: 32 MB Init Duration: 95.17 ms
```

Request ID d4539d35-bc4d-4539-99c0-a4475259b028

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

CloudWatch > Log groups > /aws/lambda/lab12_48 > 2024/10/07/[\$LATEST]f2914f94a7824006b9d673478b103e95

Log events

You can use the filter bar below to search for and match terms, phrases, or values in your log events. [Learn more about filter patterns](#)

Timestamp	Message
No older events at this moment. Retry	
2024-10-07T16:44:34.748Z	INIT_START Runtime Version: python:3.12.v36 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:188d0ca2e2214ff5637bd2bbe96ceb81ec3b0c408aef277dab104c14cd814b0881
2024-10-07T16:44:34.838Z	START RequestId: d4539d35-bc4d-4539-99c0-a4475259b020 Version: \$LATEST
2024-10-07T16:44:34.839Z	An image was uploaded to bucket example-bucket:test%2Fkey
2024-10-07T16:44:34.841Z	END RequestId: d4539d35-bc4d-4539-99c0-a4475259b020
2024-10-07T16:44:34.841Z	REPORT RequestId: d4539d35-bc4d-4539-99c0-a4475259b020 Duration: 2.07 ms Billed Duration: 3 ms Memory Size: 128 MB Max Memory Used: 32 MB Init Duration: 95.17 ms
No newer events at this moment. Auto retry paused. Resume	

Conclusion: In this experiment, we successfully created an AWS Lambda function that logs a message when an image is uploaded to an S3 bucket. It is important to note that we have to select S3-put template in event other wise code will give an error. The function was successfully triggered by S3 object uploads, validating the functionality of Lambda's event-driven architecture. This experiment demonstrated how Lambda can efficiently respond to S3 events and how to troubleshoot common issues with event structure.