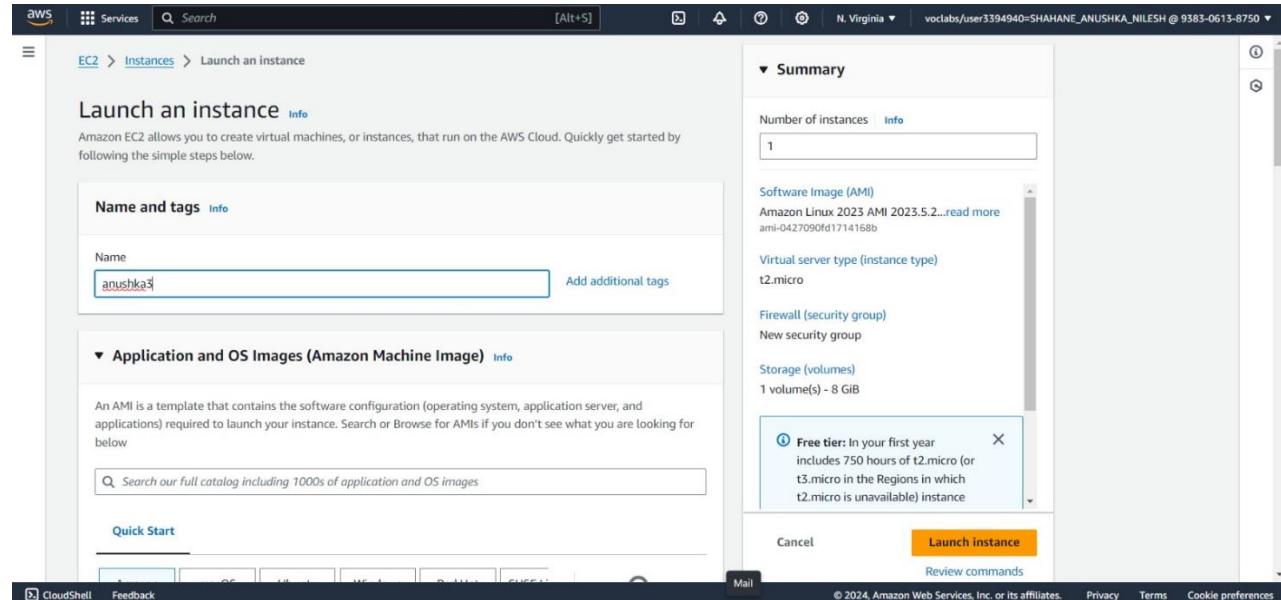


ADVANCE DEVOPS EXPERIMENT 1

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



A screenshot of the "Launch an instance" wizard, step 1: "Name and tags". It shows a "Name" input field containing "Anushka" and a "Add additional tags" button. The background shows the "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." message.

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

Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Li  [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 AMI ID
64-bit (x86) ▼ ami-04a81a99f5ec58529 Verified provider

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

1x GiB ▼ Root volume (Not encrypted)

i 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

i 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](#)

Anushka Shahane D15A 55

The screenshot shows the AWS EC2 Instances Launch an instance success page. At the top, there is a green success banner with the message "Successfully initiated launch of instance (i-0df3904aed5f9e9d9)". Below the banner, there is a "Launch log" link. A "Next Steps" section follows, containing a search bar and a numbered navigation bar (1-6). The steps are:

- Create billing and free tier usage alerts
- Connect to your instance
- Connect an RDS database
- Create EBS snapshot policy

Each step has associated links: "Create billing alerts", "Connect to instance", "Connect an RDS database", and "Create EBS snapshot policy".

This screenshot shows the same EC2 Instances Launch an instance success page, but it includes a detailed log of the launch process. The log shows the following steps and their status:

Action	Status
Initializing requests	Succeeded
Creating security groups	Succeeded
Creating security group rules	Succeeded
Launch initiation	Succeeded

The rest of the page is identical to the first screenshot, featuring the "Next Steps" section with its respective links.

Anushka Shahane D15A 55

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like EC2 Dashboard, EC2 Global View, Events, Console-to-Code, Instances (selected), Instances Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, and Images. The main content area has a header 'Instances (1) Info' with filters for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, Public IPv4 address, and Elastic IP. A search bar says 'Find Instance by attribute or tag (case-sensitive)' and a dropdown says 'All states'. Below the header is a table with one row for 'Anushka'. The table columns are Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, Public IPv4 address, and Elastic IP. The instance 'Anushka' is listed with the following details: i-0df3904aed5f9e9d9, Running, t2.micro, Initializing, us-east-1b, ec2-54-197-204-120.compute-1.amazonaws.com, 54.197.204.120, and -.

This screenshot shows the detailed view of the instance 'i-0df3904aed5f9e9d9' (Anushka). The top part is identical to the previous screenshot. Below it, the instance summary section is expanded. It contains the following details:

Item	Value
Instance ID	i-0df3904aed5f9e9d9 (Anushka)
IPv6 address	-
Hostname type	IP name: ip-172-31-42-176.ec2.internal
Answer private resource DNS name	IPv4 (A)
Auto-assigned IP address	
Public IPv4 address	54.197.204.120 open address
Instance state	Running
Private IP DNS name (IPv4 only)	ip-172-31-42-176.ec2.internal
Instance type	t2.micro
VPC ID	
Private IPv4 addresses	172.31.42.176
Public IPv4 DNS	ec2-54-197-204-120.compute-1.amazonaws.com open address
Elastic IP addresses	-
AWS Compute Optimizer finding	

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

```
ubuntu@ip-172-31-42-176:~$ ls
ubuntu@ip-172-31-42-176:~$ echo "hello"
hello
ubuntu@ip-172-31-42-176:~$ cat > myfile.txt
This is Advance devops lab
^C
ubuntu@ip-172-31-42-176:~$ cat myfile
cat: myfile: No such file or directory
ubuntu@ip-172-31-42-176:~$ cat myfile.txt
This is Advance devops lab
ubuntu@ip-172-31-42-176:~$ █
```

Hosting a static website using EC2 instance:

```
*** System restart required ***
Pending kernel upgrade!
Running kernel version:
  6.8.0-1009-aws
Diagnostics:
  The currently running kernel version is not the expected kernel version 6.8.0-1012-aws.
Last login: Tue Jul 30 08:37:47 2024 from 18.206.107.28
ubuntu@ip-172-31-41-78:~$ sudo su
root@ip-172-31-41-78:/home/ubuntu$ apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.58-1ubuntu8.4).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@ip-172-31-41-78:/home/ubuntu$ systemctl
```

i-0104434d25a50dc8d (anushka1)

PublicIPs: 18.215.241.79 PrivateIPs: 172.31.41.78

```
[ 12917 /usr/sbin/apache2 -k start
[ 12919 /usr/sbin/apache2 -k start
[ 12921 /usr/sbin/apache2 -k start
```

```
Jul 30 08:44:17 ip-172-31-41-78 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Jul 30 08:44:17 ip-172-31-41-78 systemd[1]: Started apache2.service - The Apache HTTP Server.
root@ip-172-31-41-78:/home/ubuntu$ cd/var/www/html/
bash: cd/var/www/html/: No such file or directory
root@ip-172-31-41-78:/home/ubuntu$ cd /var/www/html/
root@ip-172-31-41-78:/var/www/html$ /var/www/html$ 
bash: /var/www/html$: No such file or directory
root@ip-172-31-41-78:/var/www/html$ 
```

i-0104434d25a50dc8d (anushka1)

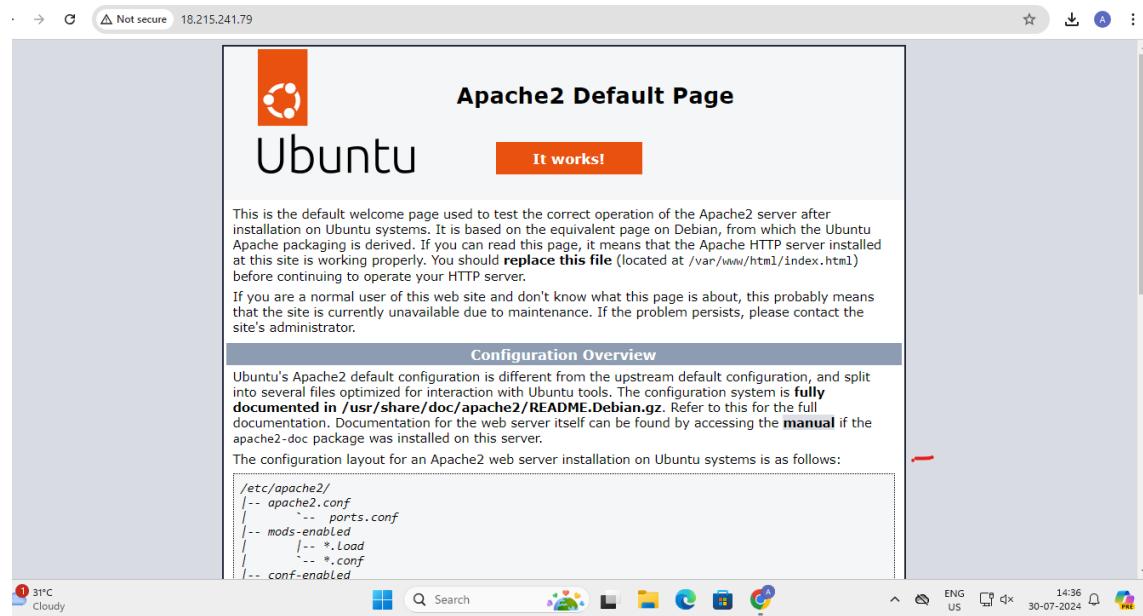
PublicIPs: 18.215.241.79 PrivateIPs: 172.31.41.78

```
command 'systemctl' from deb systemctl (1.4.4181-1.1)
Try: apt install <deb name>
root@ip-172-31-41-78:/home/ubuntu# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Tue 2024-07-30 08:44:17 UTC; 12min ago
       Docs: http://httpd.apache.org/docs/2.4/
         Main PID: 12917 (apache2)
            Tasks: 55 (limit: 1130)
           Memory: 5.3M (peak: 5.4M)
             CPU: 74ms
            CGroup: /system.slice/apache2.service
                      └─12917 /usr/sbin/apache2 -k start
                         ├─12919 /usr/sbin/apache2 -k start
                         ├─12921 /usr/sbin/apache2 -k start
                         └─12921 /usr/sbin/apache2 -k start

Jul 30 08:44:17 ip-172-31-41-78 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Jul 30 08:44:17 ip-172-31-41-78 systemd[1]: Started apache2.service - The Apache HTTP Server.
root@ip-172-31-41-78:/home/ubuntu#
```

i-0104434d25a50dc8d (anushka1)

PublicIPs: 18.215.241.79 PrivateIPs: 172.31.41.78



Hosting using S3 bucket :

Amazon S3 > Buckets > Create bucket

Create bucket Info

Buckets are containers for data stored in S3.

General configuration

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

Bucket type Info

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

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

Bucket name Info
test-123-anushka

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

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 (DSSE-KMS)
Secure your objects with two separate layers of encryption. For details on pricing, see DSSE-KMS pricing on the Storage tab of the [Amazon S3 pricing page](#). [?]

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

Disable
 Enable

► Advanced settings

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

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

Anushka Shahane D15A 55

The screenshot shows the AWS S3 Buckets page. At the top, a green banner displays a success message: "Successfully created bucket 'test-123-anushka'". Below the banner, it says "To upload files and folders, or to configure additional bucket settings, choose View details." On the left, there's a breadcrumb navigation: "Amazon S3 > Buckets". A callout box labeled "Account snapshot - updated every 24 hours" provides information about storage usage and activity trends. On the right, there are buttons for "View Storage Lens dashboard", "Create bucket", and other actions like Copy ARN, Empty, and Delete. The main area is divided into "General purpose buckets" and "Directory buckets". Under "General purpose buckets", there is a table with one row for "test-123-anushka". The table columns include Name, AWS Region, IAM Access Analyzer, and Creation date.

Name	AWS Region	IAM Access Analyzer	Creation date
test-123-anushka	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 11, 2024, 19:49:09 (UTC+05:30)

The screenshot shows the AWS S3 Upload summary page. At the top, a green banner indicates "Upload succeeded" and "View details below." A note states that the information will no longer be available after navigating away. The main section is titled "Summary" and shows the destination as "s3://test-123-anushka". It lists "Succeeded" (1 file, 0 B (0%)) and "Failed" (0 files, 0 B (0%)). Below this, there are tabs for "Files and folders" and "Configuration". The "Files and folders" tab shows a table with one row for "Test.txt". The table columns include Name, Folder, Type, Size, Status, and Error.

Name	Folder	Type	Size	Status	Error
Test.txt	-	text/plain	0 B	Succeeded	-

The screenshot shows the AWS S3 Object overview page for "Test.txt" in the "test-123-anushka" bucket. The left sidebar includes links for Buckets, Storage Lens, and Feature spotlight. The main content shows the object key "Test.txt" and its properties. The "Properties" tab is selected, displaying details such as Owner (avslabsc0w4201793t1653663267), AWS Region (US East (N. Virginia) us-east-1), Last modified (August 11, 2024, 19:58:50 (UTC+05:30)), Size (0 B), Type (txt), and Key (Test.txt). To the right, there are sections for "Object overview" (listing S3 URI, ARN, Entity tag, and Object URL) and "Object actions" (Copy S3 URI, Download, Open, and Object actions dropdown).

Anushka Shahane D15A 55

Successfully edited bucket policy.

Amazon S3 > Buckets > test-123-anushka

test-123-anushka Info

Objects Properties Permissions Metrics Management Access Points

Permissions overview

Access finding

Access findings are provided by IAM external access analyzers. Learn more about How IAM analyzer findings work [\[?\]](#)

[View analyzer for us-east-1](#)

Block public access (bucket settings)

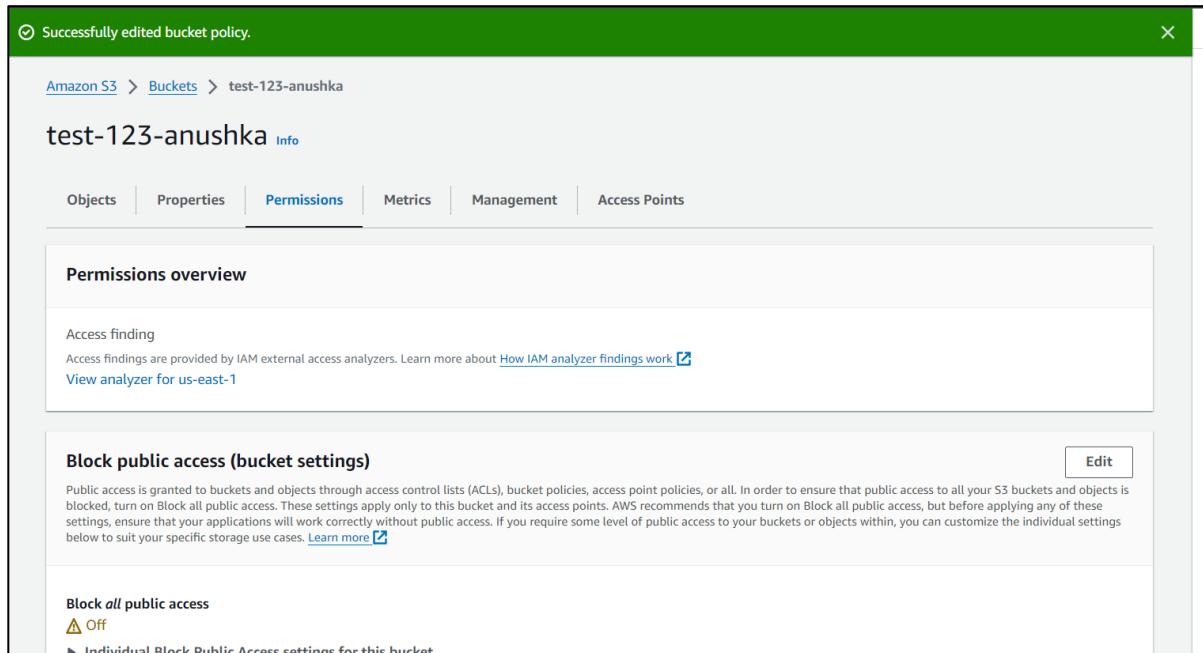
Edit

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

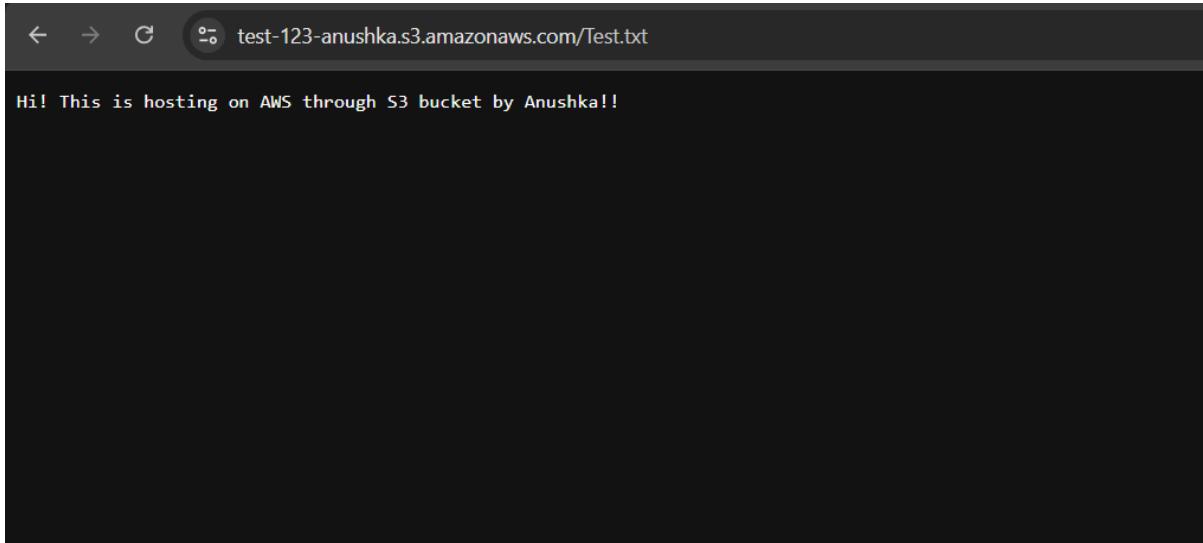
⚠ Off

[Individual Block Public Access settings for this bucket](#)

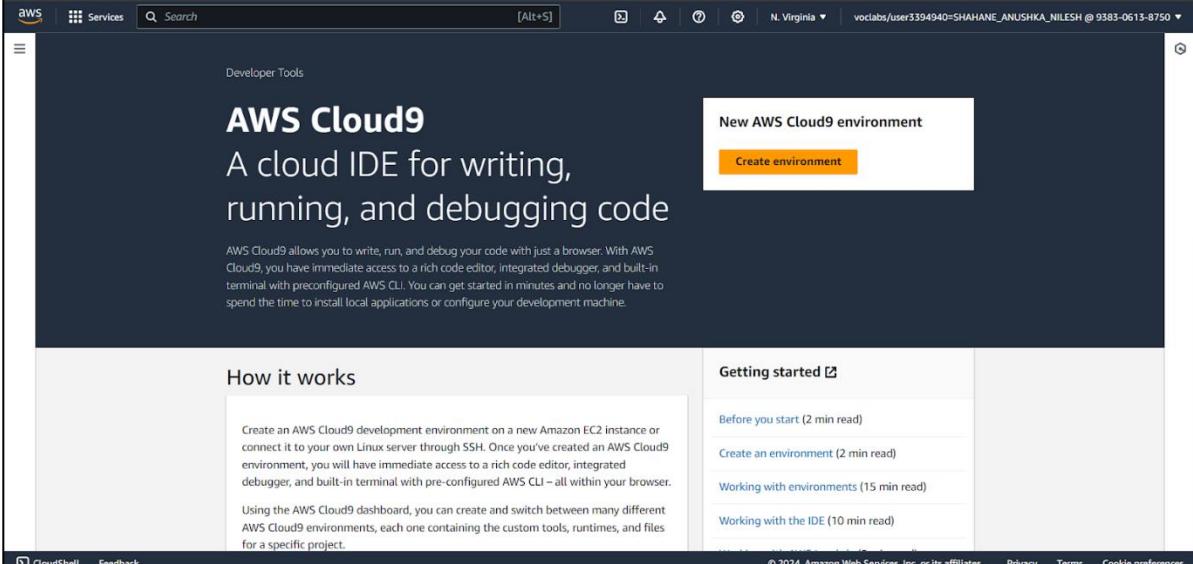


← → ⌂ test-123-anushka.s3.amazonaws.com/Test.txt

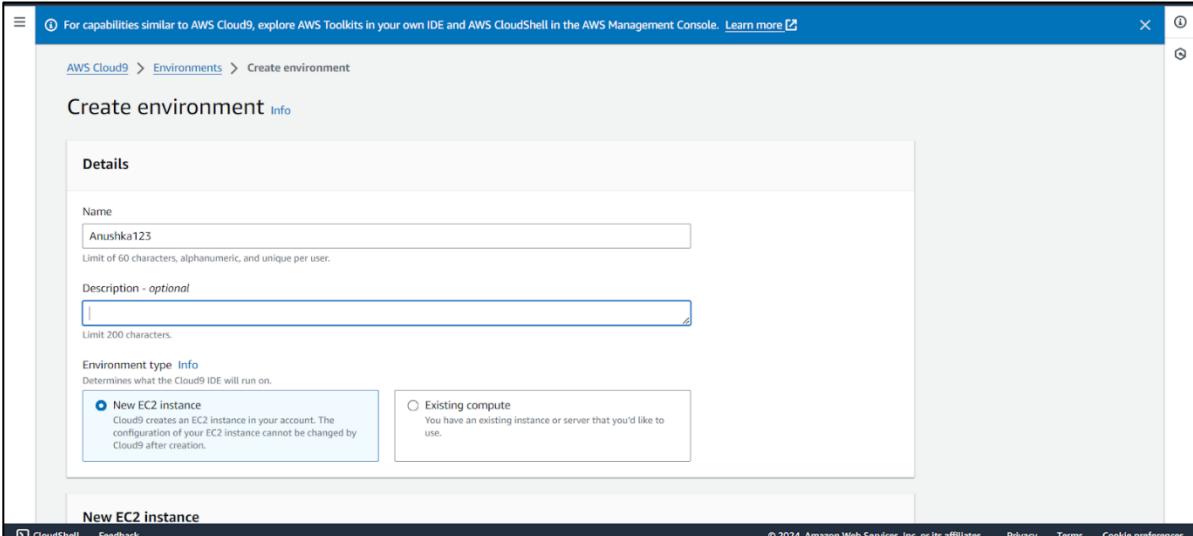
Hi! This is hosting on AWS through S3 bucket by Anushka!!



Hosting using Cloud 9 :



The screenshot shows the AWS Cloud9 homepage. At the top right, there is a call-to-action button labeled "Create environment". Below this, there is a section titled "How it works" which describes the service's purpose: "AWS Cloud9 allows you to write, run, and debug your code with just a browser. With AWS Cloud9, you have immediate access to a rich code editor, integrated debugger, and built-in terminal with preconfigured AWS CLI. You can get started in minutes and no longer have to spend the time to install local applications or configure your development machine." On the right side, there is a sidebar titled "Getting started" with links to "Before you start", "Create an environment", "Working with environments", and "Working with the IDE".



The screenshot shows the "Create environment" form in the AWS Cloud9 console. The "Details" section is active, showing fields for "Name" (containing "Anushka123") and "Description - optional". Under "Environment type", the "New EC2 instance" option is selected, with a sub-information box explaining that Cloud9 creates an EC2 instance in the user's account. There is also an "Existing compute" option. At the bottom left, there is a "New EC2 instance" button.

New EC2 instance

Instance type Info

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

t2.micro (1 GiB RAM + 1 vCPU)

Free-tier eligible. Ideal for educational users and exploration.

t3.small (2 GiB RAM + 2 vCPU)

Recommended for small web projects.

m5.large (8 GiB RAM + 2 vCPU)

Recommended for production and most general-purpose development.

Additional instance types

Explore additional instances to fit your need.

Platform Info

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

Amazon Linux 2023

Timeout

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

30 minutes

Network settings Info

Connection

How your environment is accessed.

AWS Systems Manager (SSM)

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

Secure Shell (SSH)

Accesses environment directly via SSH, opens inbound ports.

► VPC settings Info

► Tags - optional Info

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



The following IAM resources will be created in your account

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

Cancel

Create

Console password

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

Custom password
Enter a custom password for the user.

 Show password

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

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

Cancel **Next**

Set permissions

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

Permissions options

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

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

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

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

Create group

▼ Set permissions boundary - optional

Set a permissions boundary to control the maximum permissions for this user. Use this advanced feature used to delegate permission management to others. [Learn more](#)

Use a permissions boundary to control the maximum permissions
You can select one of the existing permissions policies to define the boundary.

Cancel **Previous** **Next**

Create user group

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

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

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

Permissions policies (946)

Filter by Type

<input type="checkbox"/>	Policy name	Type	Use...	Description
<input type="checkbox"/>	AdministratorAccess	AWS managed ...	None	Provides full
<input type="checkbox"/>	AdministratorAcce...	AWS managed	None	Grants accou
<input type="checkbox"/>	AdministratorAcce...	AWS managed	None	Grants accou
<input type="checkbox"/>	AlexaForBusinessD...	AWS managed	None	Provide devi
<input type="checkbox"/>	AlexaForBusinessF...	AWS managed	None	Grants full a

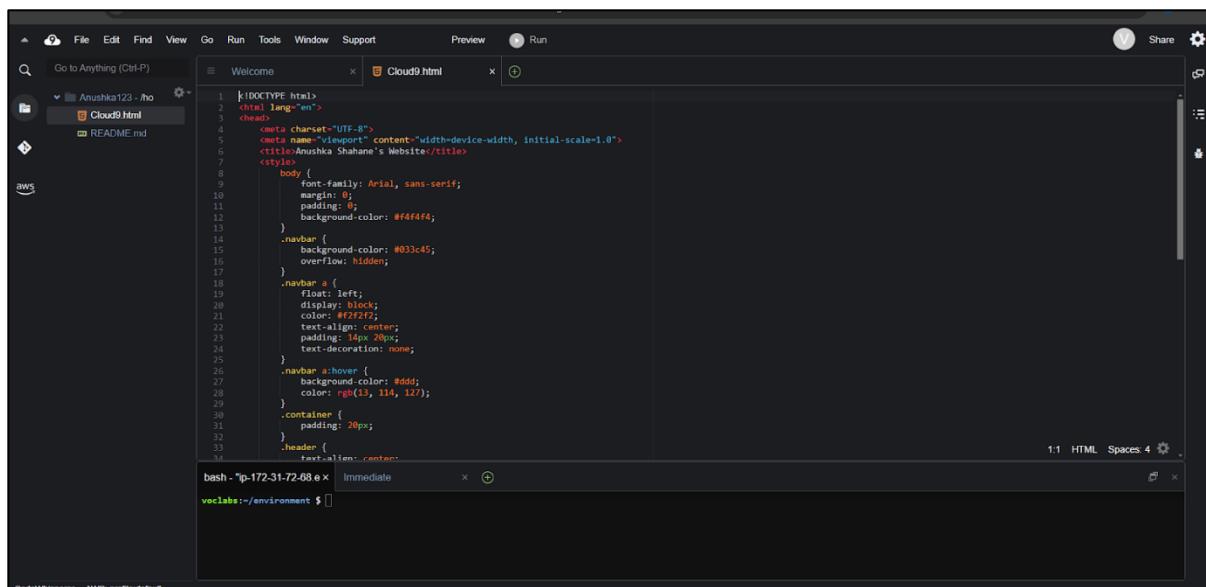
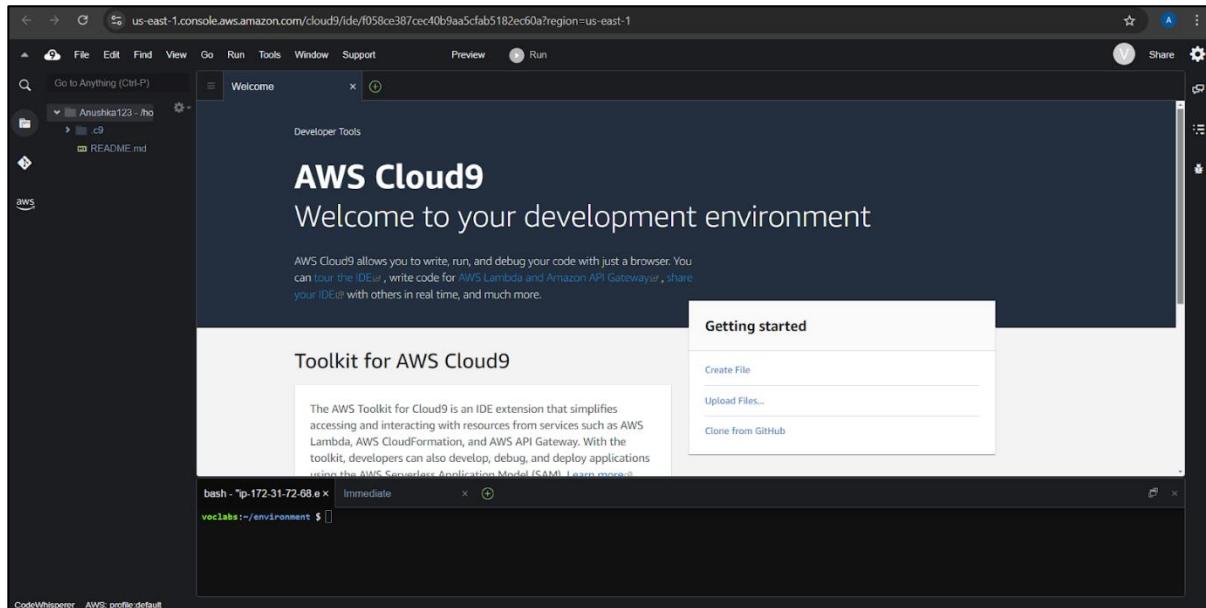
Creating Anushka123. This can take several minutes. While you wait, see [Best practices for using AWS Cloud9](#)

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

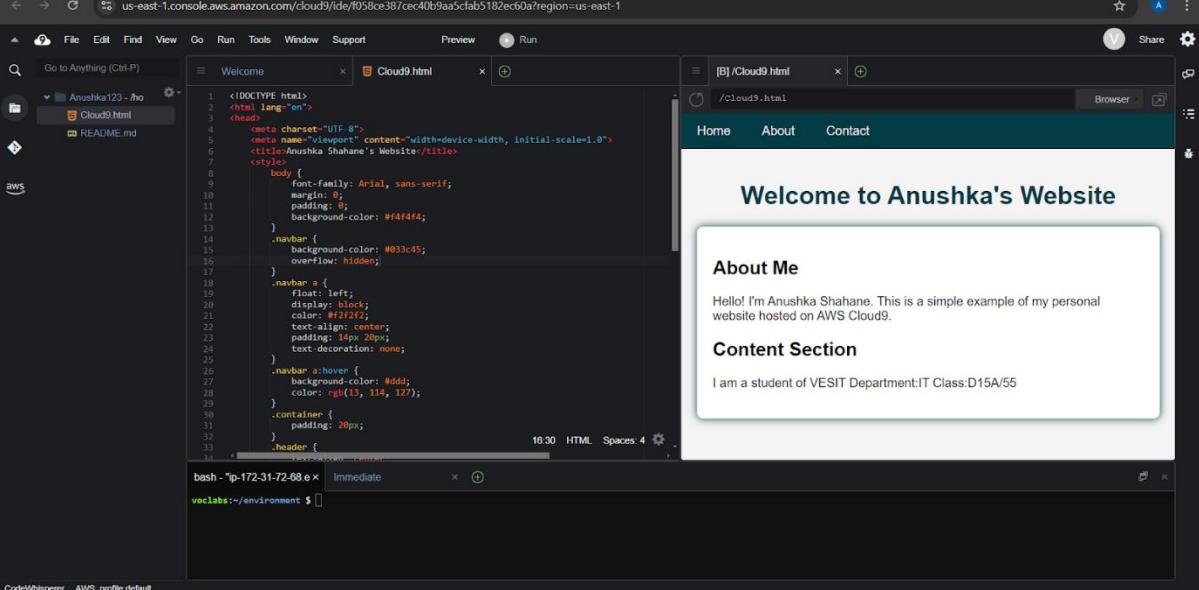
AWS Cloud9 > Environments

Environments (1)

Name	Cloud9 IDE	Environment type	Connection	Permissio	Owner ARN
Anushka123	Open	EC2 instance	Secure Shell (SSH)	Owner	arn:aws:sts::938306138750:assumed-role/voclabs/user3394940=SHAHANE_ANUSHKA_NILESH



Anushka Shahane D15A 55



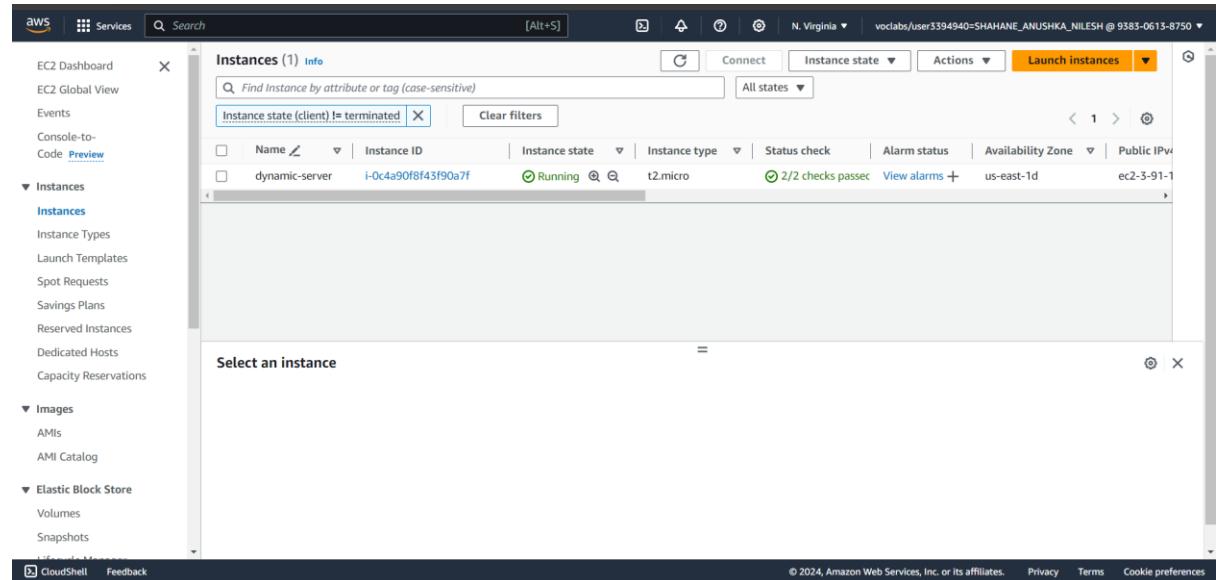
The screenshot shows the AWS Cloud9 IDE interface. On the left, there's a file tree with 'Anushka123 - htm' containing 'Cloud9.html' and 'README.md'. The main area has two tabs: 'Welcome' (HTML code) and 'Cloud9.html' (Preview). The preview shows a simple website with a header 'Welcome to Anushka's Website', a 'About Me' section with the message 'Hello! I'm Anushka Shahane. This is a simple example of my personal website hosted on AWS Cloud9.', and a 'Content Section' with the message 'I am a student of VESIT Department:IT Class:D15A/55'. Below the preview is a terminal window showing a bash session on an EC2 instance.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Anushka Shahane's Website</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            background-color: #f4f4f4;
        }
        .navbar {
            background-color: #033c45;
            overflow: hidden;
        }
        .navbar a {
            float: left;
            display: block;
            color: #fff;
            text-align: center;
            padding: 14px 20px;
            text-decoration: none;
        }
        .navbar a:hover {
            background-color: #ddd;
            color: #033c45;
        }
        .container {
            padding: 20px;
        }
        .header {
            background-color: #033c45;
            color: white;
            padding: 10px;
        }
    </style>
</head>
<body>
    <div class="header">
        Welcome to Anushka's Website
    </div>
    <div class="container">
        <div style="border: 1px solid #ccc; padding: 10px; border-radius: 10px; margin-bottom: 10px; background-color: #fff; position: relative; height: 100px; width: 100%;">
            <div style="position: absolute; top: 0; left: 0; width: 100%; height: 100%; background-color: #fff; border-radius: 10px; z-index: 1; opacity: 0.9; transition: all 0.3s ease; transform: scale(1);></div>
            <div style="position: absolute; top: 0; left: 0; width: 100%; height: 100%; background-color: #033c45; border-radius: 10px; z-index: 2; opacity: 0.1; transition: all 0.3s ease; transform: scale(1);></div>
        </div>
        <div style="border: 1px solid #ccc; padding: 10px; border-radius: 10px; background-color: #fff; position: relative; height: 100px; width: 100%;">
            <div style="position: absolute; top: 0; left: 0; width: 100%; height: 100%; background-color: #fff; border-radius: 10px; z-index: 1; opacity: 0.9; transition: all 0.3s ease; transform: scale(1);></div>
            <div style="position: absolute; top: 0; left: 0; width: 100%; height: 100%; background-color: #033c45; border-radius: 10px; z-index: 2; opacity: 0.1; transition: all 0.3s ease; transform: scale(1);></div>
        </div>
        <div style="border: 1px solid #ccc; padding: 10px; border-radius: 10px; background-color: #fff; position: relative; height: 100px; width: 100%;">
            <div style="position: absolute; top: 0; left: 0; width: 100%; height: 100%; background-color: #fff; border-radius: 10px; z-index: 1; opacity: 0.9; transition: all 0.3s ease; transform: scale(1);></div>
            <div style="position: absolute; top: 0; left: 0; width: 100%; height: 100%; background-color: #033c45; border-radius: 10px; z-index: 2; opacity: 0.1; transition: all 0.3s ease; transform: scale(1);></div>
        </div>
    </div>
    <div style="text-align: center; margin-top: 20px; font-size: 0.8em; font-weight: bold; color: #033c45; letter-spacing: 1px; text-decoration: underline;">About MeContent Section

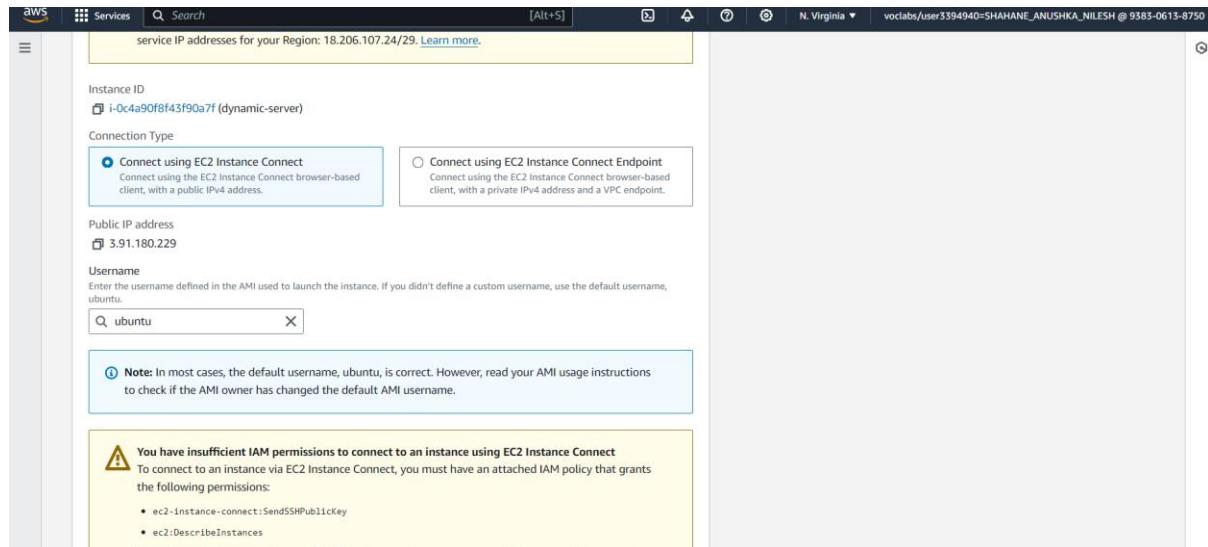
bash - "ip-172-31-72-68.e" Immediate vclabs:~/environment $


```

Dynamic Hosting using EC2 instance :



Anushka Shahane D15A 55



```
ubuntu@ip-172-31-83-228:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-83-228:~$ mkdir Anushka
ubuntu@ip-172-31-83-228:~$ cd Anushka
ubuntu@ip-172-31-83-228:~/Anushka$ git clone https://github.com/Anushka3204/Dynamic_hosting_EC2.git
Cloning into 'Dynamic_hosting_EC2'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), 8.39 KiB | 1.68 MiB/s, done.
ubuntu@ip-172-31-83-228:~/Anushka$ ls
Dynamic_hosting_EC2
ubuntu@ip-172-31-83-228:~/Anushka$ cd^C
ubuntu@ip-172-31-83-228:~/Anushka$ cd Dynamic_hosting_EC2
ubuntu@ip-172-31-83-228:~/Anushka/Dynamic_hosting_EC2$ ls
index.js package-lock.json package.json
```

```
NO VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-83-228:~/Anushka/Dynamic_hosting_EC2$ npm i
added 64 packages, and audited 65 packages in 3s

12 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
ubuntu@ip-172-31-83-228:~/Anushka/Dynamic_hosting_EC2$ 
```

```
Dynamic_hosting_EC2  Dynamic_hosting_ec2  Hosting_dynamic_ec2  package-lock.json
ubuntu@ip-172-31-83-228:~/Anushka$ cd Hosting_dynamic_ec2
ubuntu@ip-172-31-83-228:~/Anushka/Hosting_dynamic_ec2$ ls
index.js package-lock.json package.json
ubuntu@ip-172-31-83-228:~/Anushka/Hosting_dynamic_ec2$ npm i
added 64 packages, and audited 65 packages in 1s

12 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
ubuntu@ip-172-31-83-228:~/Anushka/Hosting_dynamic_ec2$ npm start
> server@1.0.0 start
> node index.js

Server is running on port 3000
```

Anushka Shahane D15A 55

The screenshot shows the AWS EC2 Security Groups interface. On the left, a navigation sidebar lists various EC2-related services and resources. The main panel displays a table of security groups, with one row selected. The selected row shows the security group ID (sg-0e9c7347eda0b5a19), name (launch-wizard-3), and VPC ID (vpc-0dbcbbe5ce8b91c8d). Below the table, tabs for 'Details', 'Inbound rules' (which is selected), 'Outbound rules', and 'Tags' are visible. The 'Inbound rules' section contains a table with two entries:

Name	Security group rule...	IP version	Type	Protocol	Port
-	sgr-0719276f3e5dc89d2	IPv4	HTTP	TCP	80
-	sgr-02b44d0276b699...	IPv4	SSH	TCP	22



ADVANCE DEVOPS EXPERIMENT NO. 2

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

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. Below the header, the main title "Amazon Elastic Beanstalk" is displayed in large bold letters, followed by the subtitle "End-to-end web application management". A "Get started" section contains a "Create application" button. Another section titled "Pricing" states that there's no additional charge for Elastic Beanstalk. At the bottom, there's a "Get started" button and some descriptive text about the service's automatic deployment features. The footer includes links for CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

The screenshot shows the "Configure environment" step in the AWS Elastic Beanstalk setup 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" and contains two sections: "Environment tier" and "Application information". Under "Environment tier", it says "Web server environment" is selected. Under "Application information", the "Application name" field is filled with "Anushka123". The footer includes links for CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

The screenshot shows the 'Platform Info' configuration page for AWS Elastic Beanstalk. The 'Platform type' section is set to 'Managed platform' (selected radio button). Below it, there is a link to 'Platforms published and maintained by Amazon Elastic Beanstalk'. The 'Custom platform' option is also present but disabled. The 'Platform' dropdown is set to 'PHP'. The 'Platform branch' dropdown is set to 'PHP 8.3 running on 64bit Amazon Linux 2023'. The 'Platform version' dropdown is set to '4.3.2 (Recommended)'. At the bottom of the page, there are links for 'CloudShell', 'Feedback', and copyright information: '© 2024, Amazon Web Services, Inc. or its affiliates.' followed by 'Privacy', 'Terms', and 'Cookie preferences'.

The screenshot shows the 'Application code' configuration page for AWS Elastic Beanstalk. The 'Sample application' option is selected (radio button). Below it, there is a link to 'Existing version' which leads to a list of uploaded application versions. There is also an option to 'Upload your code' by uploading a source bundle from a computer or copying one from Amazon S3. The 'Presets' section is visible below, showing configuration presets: 'Single instance (free tier eligible)' (selected radio button), 'Single instance (using spot instance)', 'High availability', 'High availability (using spot and on-demand instances)', and 'Custom configuration'. At the bottom right, there are 'Cancel' and 'Next' buttons. The bottom navigation bar includes 'CloudShell', 'Feedback', and copyright information: '© 2024, Amazon Web Services, Inc. or its affiliates.' followed by 'Privacy', 'Terms', and 'Cookie preferences'.

Anushka Shahane D15A 55

How would you rate your experience with this service console? ☆ ☆ ☆ ☆ ☆

aws Services Search [Alt+S] N. Virginia v vocabs/user3394940=SHAHANE_ANUSHKA_NILESH @ 9383-0613-8750 ▾

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

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

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

Step 6
[Review](#)

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

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

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

[View permission details](#)

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

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

aws Services Search [Alt+S] N. Virginia v vocabs/user3394940=SHAHANE_ANUSHKA_NILESH @ 9383-0613-8750 ▾

Step 1
[Configure environment](#)

Step 2
[Configure service access](#)

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

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

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

Step 6
[Review](#)

Configure instance traffic and scaling - *optional* Info

▼ Instances Info
Configure the Amazon EC2 instances that run your application.

Root volume (boot device)

Root volume type

Size
The number of gigabytes of the root volume attached to each instance.
8 GB

IOPS
Input/output operations per second for a provisioned IOPS (SSD) volume.
100 IOPS

Throughput
The desired throughput to provision for the Amazon EBS root volume attached to your environment's EC2 instances.

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

Anushka Shahane D15A 55

The screenshot shows the AWS Management Console interface for managing EC2 security groups. The top navigation bar includes the AWS logo, a 'Services' dropdown, a search bar, and account information ('N. Virginia' and 'voclabs/user3394940=SHAHANE_ANUSHKA_NILESH @ 9383-0613-8750'). The main content area is titled 'EC2 security groups' and displays a list of six security groups. A search bar at the top of the list allows filtering by group name. The columns in the table are 'Group name', 'Group ID', and 'Name'. The 'launch-wizard-5' group is selected, indicated by a blue border around its row.

Group name	Group ID	Name
default	sg-0d3cb5afc1b7cbb08	
launch-wizard-1	sg-0860b2782be688343	
launch-wizard-2	sg-0660e70bfb96919c1	
launch-wizard-3	sg-0e9c7347eda0b5a19	
launch-wizard-4	sg-089eb269faf05e160	
launch-wizard-5	sg-0c3feed1848240d94	

The screenshot shows the 'Review' step of the AWS Elastic Beanstalk 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), and Step 5 (optional: Configure updates, monitoring, and logging). The main content area is titled 'Step 1: Configure environment' and contains a 'Review' section with 'Edit' and 'Info' buttons. It displays environment information: Environment tier (Web server environment), Application name (Anushka123), Environment name (Anushka123-env), Application code (Sample application), and Platform (arm:aws:elasticbeanstalk:us-east-1::platform/PHP 8.3 running on 64bit Amazon Linux 2023/4.3.2).

Anushka Shahane D15A 55

AWS CloudWatch Metrics Configuration			
	On-demand above base	Capacity rebalancing	Scaling cooldown
0	Deactivated	360	
Processor type	Instance types	AMI ID	
x86_64	t3.micro,t3.small	ami-01666c45687a3fe87	
Availability Zones	Metric	Statistic	
Any	NetworkOut	Average	
Unit	Period	Breach duration	
Bytes	5	5	
Upper threshold	Scale up increment	Lower threshold	
6000000	1	2000000	
Scale down increment			
-1			
Load balancer			
Load balancer visibility	Load balancer type		
public	application		

AWS CloudWatch Metrics Configuration															
	Off	–	60												
Memory limit	Zlib output compression	Proxy server													
256M	Off	nginx													
Logs retention	Rotate logs	Update level													
7	Deactivated	minor													
X-Ray enabled															
Deactivated															
Environment properties															
<table border="1"><thead><tr><th>Key</th><th>▲</th><th>Value</th><th>▼</th></tr></thead><tbody><tr><td colspan="4">No environment properties</td></tr><tr><td colspan="4">There are no environment properties defined</td></tr></tbody></table>				Key	▲	Value	▼	No environment properties				There are no environment properties defined			
Key	▲	Value	▼												
No environment properties															
There are no environment properties defined															
Cancel Previous Submit															

Anushka Shahane D15A 55

The screenshot shows the AWS CodePipeline interface. The top navigation bar includes the AWS logo, a services menu, a search bar, and account information for 'N. Virginia' and 'voclabs/user3394940=SHAHANE_ANUSHKA_NILESH @ 9383-0613-8750'. The main content area is titled 'Choose pipeline settings' with a sub-section 'Step 1 of 5'. On the left, a sidebar lists steps: Step 1 (Choose pipeline settings), Step 2 (Add source stage), Step 3 (Add build stage), Step 4 (Add deploy stage), and Step 5 (Review). The central panel is titled 'Pipeline settings' and contains fields for 'Pipeline name' (with placeholder 'Enter the pipeline name. You cannot edit the pipeline name after it is created.') and 'Pipeline type' (with a note: '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.'). Below these are sections for 'Execution mode' (radio buttons for 'Superseded', 'Queued (Pipeline type V2 required)', and 'Parallel (Pipeline type V2 required)'), and 'Pipeline triggers' (with a note: 'Triggers don't wait for other runs to complete before starting or finishing'). The bottom of the screen shows standard AWS footer links: CloudShell, Feedback, © 2024, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

This screenshot is identical to the one above, but the 'Pipeline name' field has been populated with the value 'Pipeline_Anushka'. All other elements, including the sidebar steps, the pipeline settings panel, and the execution mode section, remain the same.

Anushka Shahane D15A 55

The screenshot shows the 'Add source stage' step in the AWS CodePipeline pipeline creation wizard. The left sidebar lists steps: Step 1 (Choose pipeline settings), Step 2 (Add source stage), Step 3 (Add build stage), Step 4 (Add deploy stage), Step 5 (Review). The main panel is titled 'Source' and contains a 'Source provider' dropdown set to 'GitHub (Version 1)'. A success message box says 'You have successfully configured the action with the provider.' Below it, a warning box states: 'The GitHub (Version 1) action is not recommended. The selected action uses OAuth apps to access your GitHub repository. This is no longer the recommended method. Instead, choose the GitHub (Version 2) action to access your repository by creating a connection. Connections use GitHub Apps to manage authentication and can be shared with other resources. Learn more'.

The screenshot shows the 'Add deploy stage' step in the AWS CodePipeline pipeline creation wizard. The left sidebar lists steps: Step 1 (Choose pipeline settings), Step 2 (Add source stage), Step 3 (Add build stage), Step 4 (Add deploy stage), Step 5 (Review). The main panel is titled 'Deploy' and contains a 'Deploy provider' dropdown set to 'AWS Elastic Beanstalk'. It includes fields for 'Region' (US East (N. Virginia)), 'Input artifacts' (SourceArtifact), 'Application name' (Anushka123), and 'Environment name' (Anushka123-env). A checkbox for 'Configure automatic rollback on stage failure' is present at the bottom.

Variables		
Name	Default value	Description
No variables		
No variables defined at the pipeline level in this pipeline.		

Step 2: Add source stage

Source action provider

Source action provider

GitHub (Version 1)

PollForSourceChanges

true

Repo

ChillNGrill

Owner

Anushka3204

Branch

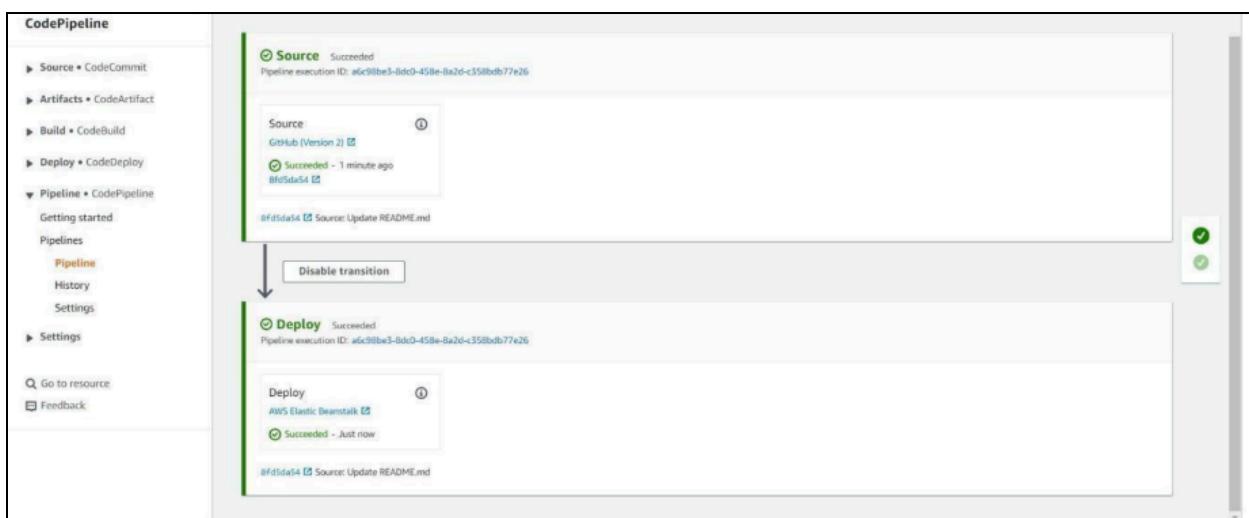
Step 3: Add build stage

Build action provider
Build stage
No build

Step 4: Add deploy stage

Deploy action provider
AWS Elastic Beanstalk
ApplicationName
Anushka123
EnvironmentName
Anushka123-env
Configure automatic rollback on stage failure
Disabled

Cancel Previous **Create pipeline**



The screenshot shows a web browser window with the URL Anushka123-env.eba.qasjmdxx.us-east-1.elasticbeanstalk.com. The page title is "ABC Consulting". The "About Us" section contains a brief welcome message and contact information: Address (456 IT Park, Andheri East, Mumbai, Maharashtra, India) and Contact Information (Email: info@abcconsulting.com | Phone: (022) 1234-5678). The "Services" section lists "Software Development: Custom software solutions tailored to your business needs".

Using S3 bucket :

The screenshot shows the "Create bucket" wizard in the AWS S3 console. The current step is "General configuration". It shows the following settings:

- AWS Region: US East (N. Virginia) us-east-1
- Bucket type: General purpose (selected)
- Bucket name: test-123-anushka
- Copy settings from existing bucket - optional: Only the bucket settings in the following configuration are copied.

Below the configuration, it says "Format: s3://bucket/prefix".

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 (DSSE-KMS)
Secure your objects with two separate layers of encryption. For details on pricing, see [DSSE-KMS pricing](#) on the [Storage](#) tab of the [Amazon S3 pricing page](#).

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

Disable
 Enable

► Advanced settings

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

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

<p>Success Successfully created bucket "test-123-anushka"</p> <p>To upload files and folders, or to configure additional bucket settings, choose View details.</p>									
Amazon S3	Buckets								
► Account snapshot - updated every 24 hours All AWS Regions	View Storage Lens dashboard								
Storage lens provides visibility into storage usage and activity trends. Learn more									
General purpose buckets Directory buckets									
General purpose buckets (1) Info All AWS Regions	Create bucket								
Buckets are containers for data stored in S3.									
<input type="text"/> Find buckets by name	C Copy ARN Empty Delete								
<table border="1"><thead><tr><th>Name</th><th>AWS Region</th><th>IAM Access Analyzer</th><th>Creation date</th></tr></thead><tbody><tr><td>test-123-anushka</td><td>US East (N. Virginia) us-east-1</td><td>View analyzer for us-east-1</td><td>August 11, 2024, 19:49:09 (UTC+05:30)</td></tr></tbody></table>	Name	AWS Region	IAM Access Analyzer	Creation date	test-123-anushka	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 11, 2024, 19:49:09 (UTC+05:30)	
Name	AWS Region	IAM Access Analyzer	Creation date						
test-123-anushka	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 11, 2024, 19:49:09 (UTC+05:30)						

Anushka Shahane D15A 55

Upload succeeded
View details below.

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

Summary

Destination	Succeeded	Failed
s3://test-123-anushka	1 file, 0 B (0%)	0 files, 0 B (0%)

[Files and folders](#) [Configuration](#)

Files and folders (1 Total, 0 B)

Name	Folder	Type	Size	Status	Error
Test.txt	-	text/plain	0 B	Succeeded	-

Amazon S3

Buckets

- Access Grants
- Access Points
- Object Lambda Access Points
- Multi-Region Access Points
- Batch Operations
- IAM Access Analyzer for S3

Block Public Access settings for this account

Storage Lens

- Dashboards
- Storage Lens groups
- AWS Organizations settings

Feature spotlight

AWS Marketplace for S3

Amazon S3 > Buckets > test-123-anushka > Test.txt

Test.txt

Properties | Permissions | Versions

Object overview

Owner	s3://test-123-anushka/Test.txt
AWS Region	Amazon Resource Name (ARN)
Last modified	us-east-1
Size	d41d8cd98f00b204e9800998ecf8427e
Type	Entity tag (Etag)
Key	https://test-123-anushka.s3.amazonaws.com/Test.txt

[Copy S3 URI](#) [Download](#) [Open](#) [Object actions](#)

Anushka Shahane D15A 55

Successfully edited bucket policy.

Amazon S3 > Buckets > test-123-anushka

test-123-anushka Info

Objects Properties Permissions Metrics Management Access Points

Permissions overview

Access finding

Access findings are provided by IAM external access analyzers. Learn more about [How IAM analyzer findings work](#) ?
[View analyzer for us-east-1](#)

Block public access (bucket settings)

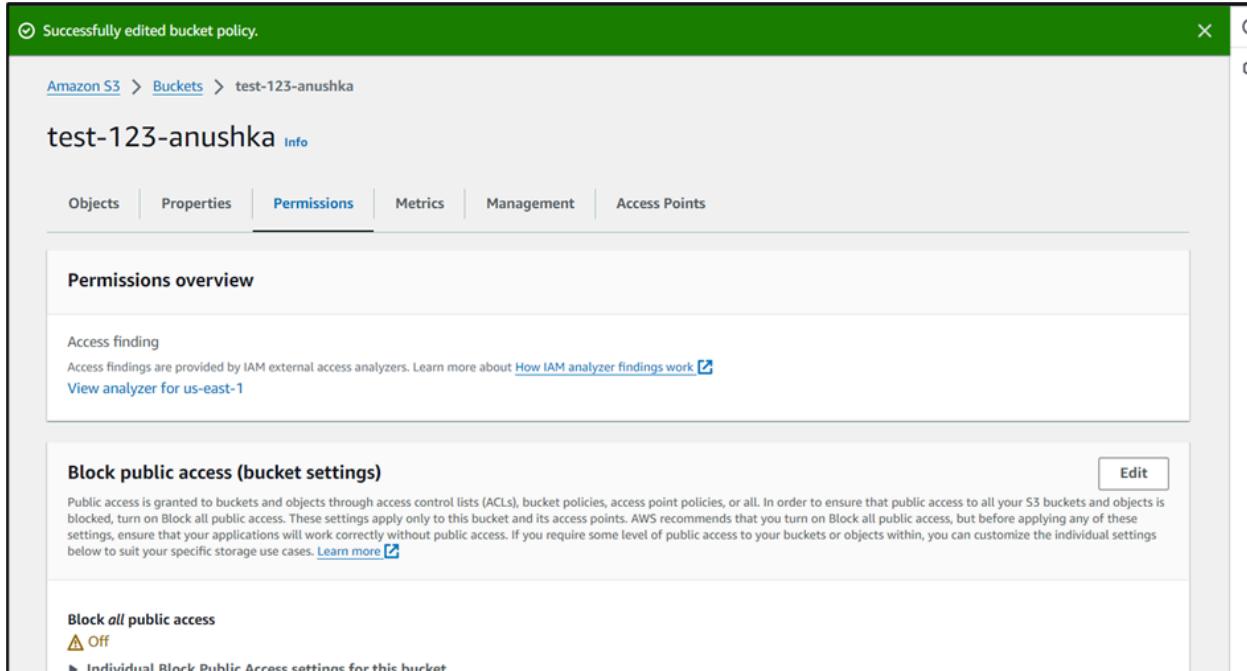
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access 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

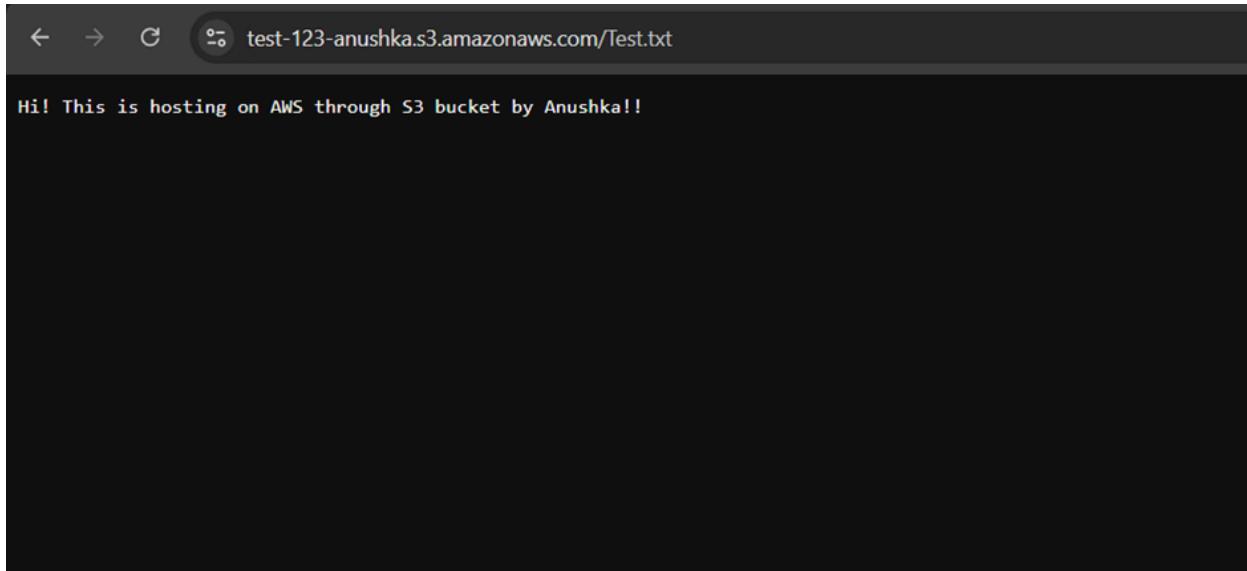
[Individual Block Public Access settings for this bucket](#)

Edit



← → ⌂ test-123-anushka.s3.amazonaws.com/Test.txt

Hi! This is hosting on AWS through S3 bucket by Anushka!!



Using EC2 instance :

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like EC2 Dashboard, EC2 Global View, Events, Console-to-Code (Preview), Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes, Snapshots). The main content area is titled "Instances (1) Info". It shows a table with one row for the instance "dynamic-server" (Instance ID: i-0c4a90f8f43f90a7f, State: Running, Type: t2.micro, Status: 2/2 checks passed, Availability Zone: us-east-1d, Public IP: ec2-3-91-1). Below the table is a "Select an instance" dropdown menu.

The screenshot shows the "Connect instance" configuration page for the instance "dynamic-server" (i-0c4a90f8f43f90a7f). It includes fields for "Instance ID" (i-0c4a90f8f43f90a7f (dynamic-server)), "Connection Type" (selected "Connect using EC2 Instance Connect"), "Public IP address" (3.91.180.229), "Username" (ubuntu), and a note about the default username. A warning message at the bottom states: "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".

Anushka Shahane D15A 55

```
Dynamic_hosting_EC2 Dynamic_hosting_ec2 Hosting_dynamic_ec2 package-lock.json
ubuntu@ip-172-31-83-228:~/Anushka$ cd Hosting_dynamic_ec2
ubuntu@ip-172-31-83-228:~/Anushka/Hosting_dynamic_ec2$ ls
index.js package-lock.json package.json
ubuntu@ip-172-31-83-228:~/Anushka/Hosting_dynamic_ec2$ npm i

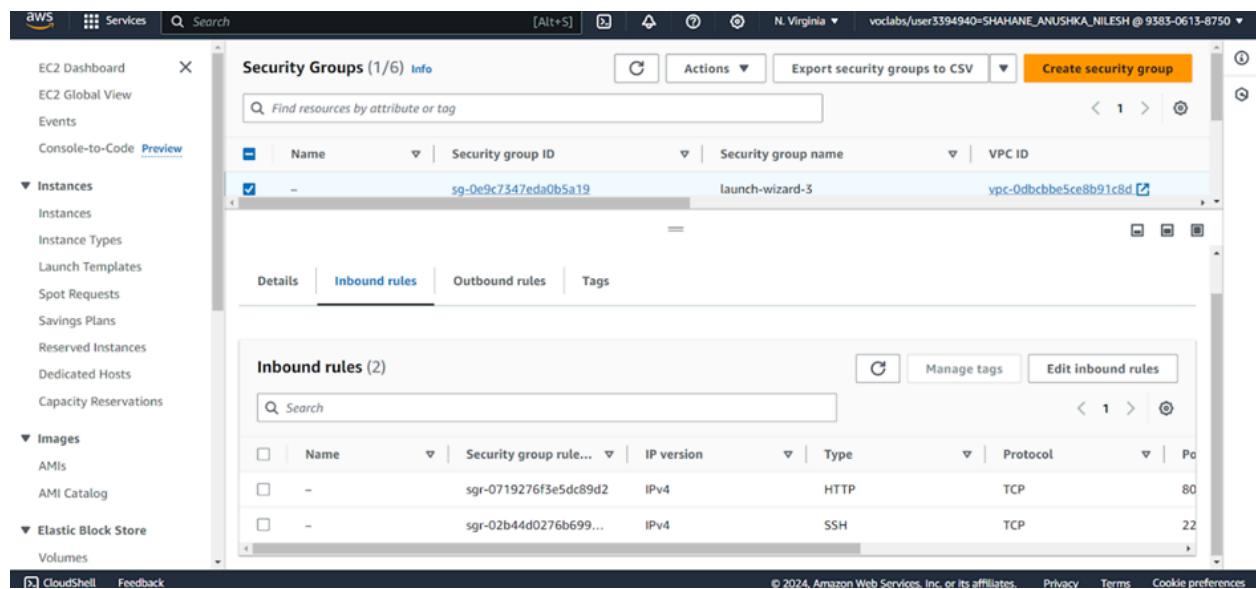
added 64 packages, and audited 65 packages in 1s

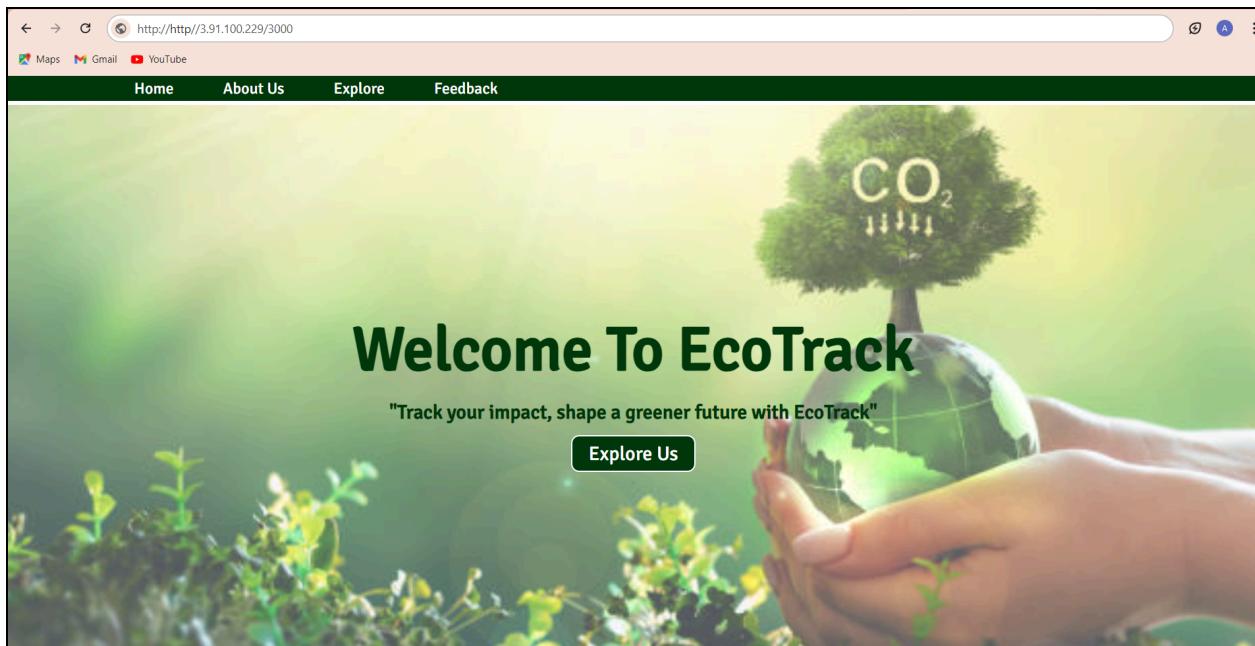
12 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
ubuntu@ip-172-31-83-228:~/Anushka/Hosting_dynamic_ec2$ npm start

> server@1.0.0 start
> node index.js

Server is running on port 3000
```





Exp 3 Advance Devops

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

Theory:

Container-based microservices architectures have revolutionized how development and operations teams test and deploy modern software. Containers allow companies to scale and deploy applications more efficiently, but they also introduce new challenges, adding complexity by creating a whole new infrastructure ecosystem.

Today, both large and small software companies are deploying thousands of container instances daily.

Managing this level of complexity at scale requires advanced tools. Enter Kubernetes. Originally developed by Google, Kubernetes is an open-source container orchestration platform designed to automate the deployment, scaling, and management of containerized applications. Kubernetes has quickly become the de facto standard for container orchestration and is the flagship project of the Cloud Native Computing Foundation (CNCF), supported by major players like Google, AWS, Microsoft, IBM, Intel, Cisco, and Red Hat.

Kubernetes simplifies the deployment and operation of applications in a microservice architecture by providing an abstraction layer over a group of hosts. This allows development teams to deploy their

applications while Kubernetes takes care of key tasks, including:

- Managing resource consumption by applications or teams
- Distributing application load evenly across the infrastructure
- Automatically load balancing requests across multiple instances of an application
- Monitoring resource usage to prevent applications from exceeding resource limits and automatically restarting them if needed
- Moving application instances between hosts when resources are low or if a host fails
- Automatically utilizing additional resources when new hosts are added to the cluster
- Facilitating canary deployments and rollbacks with ease

Necessary Requirements:

- EC2 Instance: The experiment required launching a t2.medium EC2 instance with 2 CPUs, as Kubernetes demands sufficient resources for effective functioning.
- Minimum Requirements:
 - Instance Type: t2.medium
 - CPUs: 2
 - Memory: Adequate for container orchestration.

This ensured that the Kubernetes cluster had the necessary resources to function smoothly

Prerequisites :

Create 2 Security Groups for Master and Nodes and add the following rules inbound rules in those Groups.

master

The screenshot shows the 'Create security group' wizard in the AWS EC2 console. In the 'Basic details' section, the 'Security group name' is set to 'Master'. The 'Description' is 'master_group'. Under 'VPC Info', the VPC is selected as 'vpc-0f1ae6b32b6900863'. A note at the bottom states: 'A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.'

The screenshot shows the 'Inbound rules' section for the 'Master' security group. It lists nine rules:

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	Anywhere...	0.0.0.0/0
All traffic	All	All	Anywhere...	0.0.0.0/0
Custom TCP	TCP	6443	Anywhere...	0.0.0.0/0
Custom TCP	TCP	10251	Anywhere...	0.0.0.0/0
Custom TCP	TCP	10250	Anywhere...	0.0.0.0/0
All TCP	TCP	0 - 65535	Anywhere...	0.0.0.0/0
Custom TCP	TCP	10252	Anywhere...	0.0.0.0/0
SSH	TCP	22	Anywhere...	0.0.0.0/0

Node

The screenshot shows the 'Inbound rules' section of an AWS Security Group configuration. It lists seven rules:

Type	Protocol	Port range	Source	Description - optional	Action
All traffic	All	All	Anywhere... ▾	Q. 0.0.0.0/0	0.0.0.0/0 X Delete
SSH	TCP	22	Anywhere... ▾	Q. 0.0.0.0/0	0.0.0.0/0 X Delete
Custom TCP	TCP	10250	Anywhere... ▾	Q. 0.0.0.0/0	0.0.0.0/0 X Delete
All TCP	TCP	0 - 65535	Anywhere... ▾	Q. 0.0.0.0/0	0.0.0.0/0 X Delete
Custom TCP	TCP	30000 - 32767	Anywhere... ▾	Q. 0.0.0.0/0	0.0.0.0/0 X Delete
HTTP	TCP	80	Anywhere... ▾	Q. 0.0.0.0/0	0.0.0.0/0 X Delete

At the bottom left is a 'Add rule' button.

Step 1: Log in to your AWS Academy/personal account and launch 3 new Ec2 Instances. Select Ubuntu as AMI and t2.medium as Instance Type and create a key of type RSA with .pem extension and move the downloaded key to the new folder. We can use 3 Different keys or 1 common key also.

Note: A minimum of 2 CPUs are required so Please select t2.medium and do not forget to stop the

instance after the experiment because it is not available in the free tier.

Also Select Security groups from existing.

Master:

Anushka Shahane D15A 55

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

Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Linux Browse more AMIs Including AMIs from AWS, Marketplace and the Community

Summary

Number of instances [Info](#)
1

Virtual server type (instance type)
t2.micro

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 GiB of snapshots, and 100 GiB of bandwidth to the internet.

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

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
Master_ec2_key [Create new key pair](#)

Network settings [Info](#)

Network [Info](#)

Summary

Number of instances [Info](#)
1

Virtual server type (instance type)
t2.micro

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 GiB of snapshots, and 100 GiB of bandwidth to the internet.

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
Master sg-09027e8860f8edb6c X
VPC: vpc-0f1ae6b32b6900863

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

Network settings [Info](#)

Network [Info](#)
vpc-0f1ae6b32b6900863

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

Auto-assign public IP [Info](#)

Enable

Configure storage [Info](#)

Advanced

1x 8 GiB gp3 Root volume (Not encrypted)

Summary

Number of instances [Info](#)
1

Virtual server type (instance type)
t2.micro

Firewall (security group)
Master

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 GiB of snapshots, and 100 GiB of bandwidth to the internet.

Cancel [Launch instance](#) Review commands

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

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

[Amazon Linux](#) [macOS](#) [Ubuntu](#) [Windows](#) [Red Hat](#) [SUSE Li](#)

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

Amazon Machine Image (AMI)

[aws](#) [Services](#) [Q Search](#) [\[Alt+S\]](#)

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

Network [Info](#)
vpc-0f1ae6b32b6900863

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

Node sg-0f7d588b9ab24195d X
VPC vpc-0f1ae6b32b6900863

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

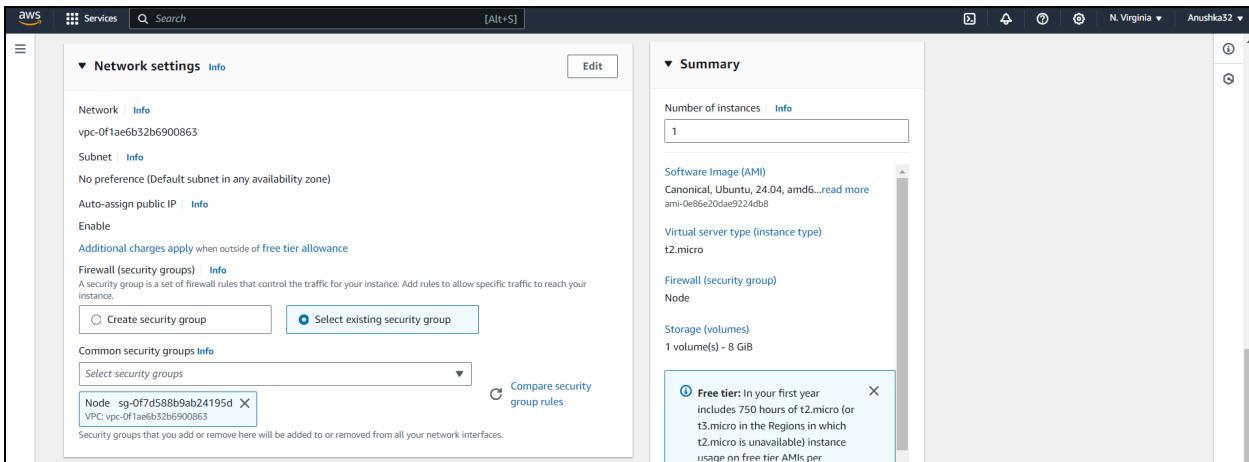
Software image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-0e66e20dae9224db8

Virtual server type (instance type)
t2.micro

Firewall (security group)
Node

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 year



Anushka Shahane D15A 55

Do Same for 2 Nodes and use security groups of Node for that.

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

Instances (3) Info											
		Name A		Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IP
<input type="checkbox"/>	Master	i-04c91a517080179c3	Running			t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-34-228-241-138.compute-1.amazonaws.com	34.228.241.138
<input type="checkbox"/>	node2	i-0464a7a65a0fcea07	Running			t2.micro	Initializing	View alarms +	us-east-1a	ec2-54-86-236-178.compute-1.amazonaws.com	54.86.236.178
<input type="checkbox"/>	node1	i-0e39bc125d07dcde	Running			t2.micro	Initializing	View alarms +	us-east-1a	ec2-107-23-240-96.compute-1.amazonaws.com	107.23.240.96

EC2 > Instances > i-04c91a517080179c3 > Connect to instance

Connect to instance [Info](#)

Connect to your instance i-04c91a517080179c3 (Master) using any of these options

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

Instance ID [i-04c91a517080179c3 \(Master\)](#)

- Open an SSH client.
- Locate your private key file. The key used to launch this instance is Master_ec2_key.pem
- Run this command, if necessary, to ensure your key is not publicly viewable.
`chmod 400 "Master_ec2_key.pem"`
- Connect to your instance using its Public DNS:
`ec2-34-228-241-138.compute-1.amazonaws.com`

Example:
`ssh -i "Master_ec2_key.pem" ubuntu@ec2-34-228-241-138.compute-1.amazonaws.com`

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

Cancel

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

Master:

Anushka Shahane D15A 55

```
ubuntu@ip-172-31-40-255: ~ + v
System information as of Tue Sep 24 18:53:14 UTC 2024

System load: 0.0          Processes: 104
Usage of /: 22.7% of 6.71GB  Users logged in: 0
Memory usage: 19%          IPv4 address for enX0: 172.31.40.255
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-172-31-40-255:~$ |
```

```
YcMmhD9mRiPpQn6Ya2w3e3B8zfIVKipbMBnke/ytZ9M7qHmDCcjoisMmwEXN3wKYI
mD9VHOnsL/CG1rU9IsW1jtB5g1YxuBA7M/m36XN6x2u+NtNMDB9P56yc4gfsZVES
KA9v+yY2/l45l8d/WUkUi0YXomn6hyBG17JrBLq0CX37GEYP609rrKipfz73Xf07
JIGzOKZlljb/D9RX/g7nRbcN+3EtH7xnk+T/50euEkw8SMUg147sJTcpQmv6UzZ
cM4JgL0HBHVCojV4C/pLElwMddALOfeYQzTif6sMRPf+3DSj8frbInjChC3y0Ly0
6br92KFom17EiJ2ACoeq7UPhi2ooUYbwPxh5ytdehJkoo+sN7RTWua6P2WSmon5
U888cylXC0+ADFdglX9K2zrDVYUG1vo8CX0vzxFBaHwN6Px26fhIT1/hYUHQRIz
VNNDcyQmXqkOnZvvoMFz/Q0s9BhFJ/uU6AgqbIZE/hm1spfgvtsD1frZfygXJ9f
irP+MSAI80xHSf91qSRZ0j4Pl3ZJNbq4yYxv0b1pkMqeGdjdcYhLU+lZ4wbQmpCk
SVE2prLureigXtmZfkqevRz7FrIZiu9ky8wnCAPwC7/zmS18rgP/17b0tL4/iIz
QhAAoAMVVrGyJivSkjhSGx1uCojsWfsTAml1P7jsruIL61zMUVE2aM3Pmj5G+W
9AcZ58Em+1WsVnAXdUR//bMmhyr8wL/G1Y01V3JEJTRdxsSxdYa4deGBBY/Adpsw
24jxh0JR+LsJpqIueb999+R8euUhRH9eFO7DRu6weatUJ6suupoDTRW/tr/4yGqe
dkxV3qQhNLsNaAzqW/1nA3iUB4k7kCaKzXhdhDbClf9P37qaRW467BLCVO/coL3y
Vm50wdwrNtKpMBh3ZpbBluJvg19mXtyB0MJ3v8RZeDzFiG8HdCtg9RvIt/AIfoHR
H3s+U79NT6i0KPzLImDfs8T7RlpuyMc4Ufs8gggyg9v3Ae6cN3eQyxck3w0cbBwsh
/nQNfsA6uu+9H7NhbehBhYnpNzyrHzCmzyXkauwRAqoCbgCNyktRwsur9gS41TQ
M8ssD1jFheOJf3h0DnkKU+HKjvMR01DK7zdmLdNzA1cvtZH/nCC9KPj1z8QC47S
xx+dTZsX40NAhwbs/LN3PoKtn8LPjY9NP9uDWI+TWYqus2U+KHDxBDlsgozDbs/0
jCxcpDzNmXpWQHETHU76490XHP7UeNST1mCUCH5qdank0V1iejF6/CFTFU4Mfcrg
YT90qFF93M3v01BbxP+EIY2/9tiIPbrd
=0YYh
-----END PGP PUBLIC KEY BLOCK-----
-bash: /etc/apt/trusted.gpg.d/docker.gpg: No such file or directory
Repository: 'deb [arch=amd64] https://download.docker.com/linux/ubuntu noble stable'
Description:
Archive for codename: noble components: stable
More info: https://download.docker.com/linux/ubuntu
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.|
```

Anushka Shahane D15A 55

```
ubuntu@ip-172-31-40-255: ~ + - × 
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [532 B]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 B]
Get:31 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [10.6 kB]
Get:32 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [10.8 kB]
Get:33 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [17.6 kB]
Get:34 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1104 B]
Get:35 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:36 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B]
Get:37 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:38 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:39 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [377 kB]
Get:40 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [81.6 kB]
Get:41 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4528 B]
Get:42 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [270 kB]
Get:43 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [113 kB]
Get:44 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [8632 B]
Get:45 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [10.1 kB]
Get:46 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [353 kB]
Get:47 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [68.1 kB]
Get:48 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 c-n-f Metadata [428 B]
Get:49 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [10.9 kB]
Get:50 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [2808 B]
Get:51 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:52 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [344 B]
Fetched 29.1 MB in 6s (4624 kB/s)
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
ubuntu@ip-172-31-40-255:~$ |
d not get lock /var/lib/apt/lists/lock. It is held by process 2918 (apt-get)
```

sudo apt-get update

sudo apt-get install -y kubelet kubeadm kubectl

sudo apt-mark hold kubelet kubeadm kubectl

```
ubuntu@ip-172-31-40-255:~$ sudo apt-get update
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  containerd.io docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0
  pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | coroupe-lite
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0
  libslirp0 pigz slirp4netns
0 upgraded, 10 newly installed, 0 to remove and 139 not upgraded.
Need to get 123 MB of archives.
After this operation, 442 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu/noble/main amd64 libltdl7 amd64 2.4.7-7build1 [40.3 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu/noble/main amd64 libslirp0 amd64 4.7.0-1ubuntu3 [63.8 kB]
```

Anushka Shahane D15A 55

```
Running kernel seems to be up-to-date.  
No services need to be restarted.  
No containers need to be restarted.  
No user sessions are running outdated binaries.  
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
ubuntu@ip-172-31-40-255:~$ |
```

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

```
}
```

```
ubuntu@ip-172-31-40-255:~$ sudo systemctl enable docker  
sudo systemctl daemon-reload  
sudo systemctl restart docker  
Synchronizing state of docker.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.  
Executing: /usr/lib/systemd/systemd-sysv-install enable docker  
ubuntu@ip-172-31-40-255:~$ |
```

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

Node1, and Node2.

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

```
Executing: /usr/lib/systemd/systemd-sysv-install enable docker  
ubuntu@ip-172-31-40-255:~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o  
/etc/apt/keyrings/kubernetes-apt-keyring.gpg  
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]  
https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list  
gpg: missing argument for option "--"  
-bash: /etc/apt/keyrings/kubernetes-apt-keyring.gpg: No such file or directory  
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]  
https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /  
ubuntu@ip-172-31-40-255:~$ |
```

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

Error

```
https://pkgs.kos.io/core./stable./v1.31/deb/  
ubuntu@ip-172-31-40-255:~$ sudo apt-get update  
sudo apt-get install -y kubelet kubeadm kubectl  
sudo apt-mark hold kubelet kubeadm kubectl  
E: Malformed entry 1 in list file /etc/apt/sources.list.d/kubernetes.list (URI)  
E: The list of sources could not be read.  
E: Malformed entry 1 in list file /etc/apt/sources.list.d/kubernetes.list (URI)  
E: The list of sources could not be read.  
E: Malformed entry 1 in list file /etc/apt/sources.list.d/kubernetes.list (URI)  
E: The list of sources could not be read.  
ubuntu@ip-172-31-40-255:~$ |
```

To solve

```
ubuntu@ip-172-31-40-255:~$ sudo mkdir -p /etc/apt/keyrings  
ubuntu@ip-172-31-40-255:~$ sudo apt-get update  
sudo apt-get install -y kubelet kubeadm kubectl  
sudo apt-mark hold kubelet kubeadm kubectl  
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease  
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]  
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease  
Hit:4 https://download.docker.com/linux/ubuntu noble InRelease  
Hit:6 http://security.ubuntu.com/ubuntu noble-security InRelease  
Get:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb InRelease [1186 B]  
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [8564 B]  
Get:8 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb Packages [4865 B]  
Fetched 141 kB in 1s (141 kB/s)  
Reading package lists... Done  
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  conntrack cri-tools kubernetes-cni  
The following NEW packages will be installed:  
  Conntrack cri-tools kubeadm kubectl kubelet kubernetes-cni  
0 upgraded, 6 newly installed, 0 to remove and 139 not upgraded.  
Need to get 87.4 MB of archives.  
After this operation, 314 MB of additional disk space will be used.  
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 conntrack amd64 1:1.4.8-1ubuntu1 [37.9 kB]  
Get:2 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb cri-tools 1.31.1-1.1 [15.7 MB]  
Get:3 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb kubeadm 1.31.1-1.1 [11.4 MB]  
Get:4 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb kubectl 1.31.1-1.1 [11.2 MB]  
Get:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb kubernetes-cni 1.5.1-1.1 [33.9 MB]  
79% [5 kubernetes-cni 33.9 MB/33.9 MB 100%]
```

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.

kubelet set on hold.

kubeadm set on hold.

kubectl set on hold.

```
ubuntu@ip-172-31-40-255:~$ |
```

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

```
ubuntu@ip-172-31-40-255:~$ sudo mkdir -p /etc/docker
cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"]
}
EOF
{
  "exec-opts": ["native.cgroupdriver=systemd"]
}
ubuntu@ip-172-31-40-255:~$ |
```

Step 5: Run the below command to install Kubernets.

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

```
ubuntu@ip-172-31-40-255:~$ sudo systemctl enable --now kubelet
sudo apt-get install -y containerd
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz slirp4netns
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  runc
The following packages will be REMOVED:
  containerd.io docker-ce
The following NEW packages will be installed:
  containerd runc
0 upgraded, 2 newly installed, 2 to remove and 139 not upgraded.
Need to get 47.2 MB of archives.
```

```
sudo mkdir -p /etc/containerd
sudo containerd config default | sudo tee /etc/containerd/config.toml
```

Anushka Shahane D15A 55

```
ubuntu@ip-172-31-40-255:~$ sudo mkdir -p /etc/containerd
sudo containerd config default | sudo tee /etc/containerd/config.toml
disabled_plugins = []
imports = []
oom_score = 0
plugin_dir = ""
required_plugins = []
root = "/var/lib/containerd"
state = "/run/containerd"
temp = ""
version = 2

[cgroup]
path = ""

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

[grpc]
address = "/run/containerd/containerd.sock"
gid = 0
max_recv_message_size = 16777216
max_send_message_size = 16777216
tcp_address = ""
tcp_tls_ca = ""
tcp_tls_cert = ""
tcp_tls_key = ""
```

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

```
uid = 0
ubuntu@ip-172-31-40-255:~$ sudo systemctl restart containerd
sudo systemctl enable containerd
sudo systemctl status containerd
● containerd.service - containerd container runtime
   Loaded: loaded (/usr/lib/systemd/system/containerd.service; enabled; preset: enabled)
     Active: active (running) since Tue 2024-09-24 19:23:12 UTC; 420ms ago
       Docs: https://containerd.io
      Main PID: 8629 (containerd)
         Tasks: 7
        Memory: 15.8M (peak: 16.0M)
          CPU: 82ms
         CGroup: /system.slice/containerd.service
             └─8629 /usr/bin/containerd

Sep 24 19:23:12 ip-172-31-40-255 containerd[8629]: time="2024-09-24T19:23:12.238510263Z" level=info msg="serving..." address=/run/containerd/containerd.sock
Sep 24 19:23:12 ip-172-31-40-255 containerd[8629]: time="2024-09-24T19:23:12.238584725Z" level=info msg="serving..." address=/run/containerd/containerd.sock
Sep 24 19:23:12 ip-172-31-40-255 containerd[8629]: time="2024-09-24T19:23:12.240940850Z" level=info msg="Start subscribing containerd event"
Sep 24 19:23:12 ip-172-31-40-255 containerd[8629]: time="2024-09-24T19:23:12.240176953Z" level=info msg="Start recovering state"
Sep 24 19:23:12 ip-172-31-40-255 containerd[8629]: time="2024-09-24T19:23:12.240253922Z" level=info msg="Start event monitor"
Sep 24 19:23:12 ip-172-31-40-255 containerd[8629]: time="2024-09-24T19:23:12.240273184Z" level=info msg="Start snapshots syncer"
Sep 24 19:23:12 ip-172-31-40-255 containerd[8629]: time="2024-09-24T19:23:12.240286270Z" level=info msg="Start cni network conf syncer for default"
Sep 24 19:23:12 ip-172-31-40-255 containerd[8629]: time="2024-09-24T19:23:12.240299291Z" level=info msg="Start streaming server"
Sep 24 19:23:12 ip-172-31-40-255 systemd[1]: Started containerd.service - containerd container runtime.
Sep 24 19:23:12 ip-172-31-40-255 containerd[8629]: time="2024-09-24T19:23:12.244849418Z" level=info msg="containerd successfully booted in 0.070201s"
ubuntu@ip-172-31-40-255:~$ |
```

```
sudo apt-get install -y socat
```

Anushka Shahane D15A 55

```
ubuntu@ip-172-31-40-255:~$ sudo apt-get install -y socat
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz slirp4
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  socat
0 upgraded, 1 newly installed, 0 to remove and 139 not upgraded.
Need to get 374 kB of archives.
After this operation, 1649 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 socat amd64 1.8.0.0-4build3 [374 kB]
Fetched 374 kB in 0s (11.6 MB/s)
Selecting previously unselected package socat.
(Reading database ... 68108 files and directories currently installed.)
Preparing to unpack .../socat_1.8.0.0-4build3_amd64.deb ...
Unpacking socat (1.8.0.0-4build3) ...
Setting up socat (1.8.0.0-4build3) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...
Running kernel seems to be up-to-date.
```

```
Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-40-255:~$ |
```

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

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

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-40-255:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
error execution phase preflight: [preflight] Some fatal errors occurred:
    [ERROR NumCPU]: the number of available CPUs 1 is less than the required 2
    [ERROR Mem]: the system RAM (957 MB) is less than the minimum 1700 MB
[preflight] If you know what you are doing, you can make a check non-fatal with '--ignore-preflight-errors=...'
To see the stack trace of this error execute with --v=5 or higher
ubuntu@ip-172-31-40-255:~$ |
```

Anushka Shahane D15A 55

```
ubuntu@ip-172-31-40-255:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=NumCPU --ignore-preflight-errors=Mem
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
  [WARNING NumCPU]: the number of available CPUs 1 is less than the required 2
  [WARNING Mem]: the system RAM (957 MB) is less than the minimum 1700 MB
[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'
W0924 19:29:57.933799    9246 checks.go:846] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent with that
used by kubeadm. It is recommended to use "registry.k8s.io/pause:3.10" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-40-255 kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster
.local] and IPs [10.96.0.1 172.31.40.255]
[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-172-31-40-255 localhost] and IPs [172.31.40.255 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-40-255 localhost] and IPs [172.31.40.255 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
[certs] Generating "sa" key and public key
```

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

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

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

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

```
export KUBECONFIG=/etc/kubernetes/admin.conf
```

```
You should now deploy a pod network to the cluster.
```

```
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
https://kubernetes.io/docs/concepts/cluster-administration/addons/
```

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

```
kubeadm join 172.31.40.255:6443 --token l2izt1.mt5iy3g70oyhjft7 \
  --discovery-token-ca-cert-hash sha256:39c290262a4af785e7629a945f25514226b3f65234f280fe02b033f0f9924cf
ubuntu@ip-172-31-40-255:~$ |
```

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

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

```
--discovery-token-ca-cert-hash sha256:39c290262a4af785e7629a945f25514226b3f65234
ubuntu@ip-172-31-40-255:~$ mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@ip-172-31-40-255:~$ |
```

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

```
ubuntu@ip-172-31-40-255:~$ kubectl get nodes
NAME           STATUS      ROLES   AGE      VERSION
ip-172-31-40-255   NotReady   control-plane   113s   v1.31.1
ubuntu@ip-172-31-40-255:~$ |
```

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

Anushka Shahane D15A 55

```
sudo kubeadm join 172.31.27.176:6443 --token ttay2x.n0squeukjai8sgfg3 \
--discovery-token-ca-cert-hash
sha256:d6fc5fb7e984c83e2807780047fec6c4f2acfe9da9184ecc028d77157608fbb6
Node 1:
```

```
ubuntu@ip-172-31-40-255:~$ sudo kubeadm join 172.31.40.255:6443 --token l2izt1.mt5iy3g7o0yhjft7 --discovery-token-ca-cer
t-hash sha256:39c290262a4af785e7629a945f25514226b3f65234f280fe02b033f0f9924cf
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 1.509666981s
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.
```

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

```
Last login: Wed Sep 25 03:09:14 2024 from 152.58.42.1
ubuntu@ip-172-31-37-88:~$ sudo kubeadm join 172.31.40.255:6443 --token zo9fea.16bddwnc11vqlqso \
--discovery-token-ca-cert-hash sha256:a92bc7fadcc6f973441bb6c3278fdb5f33e67ed5c9dc5d7b83aa2aaf2b56b0510
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 501.636003ms
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

ubuntu@ip-172-31-37-88:~$ client_loop: send disconnect: Connection reset
C:\Users\Admin\Desktop\Node1_key>
```

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

```
Last login: Wed Sep 25 03:10:01 2024 from 152.58.42.1
ubuntu@ip-172-31-40-255:~$ kubectl get nodes -o wide
NAME           STATUS    ROLES                  AGE   VERSION
ip-172-31-37-88  Ready    <none>                45m   v1.31.1
ip-172-31-40-255  Ready    node1,control-plane  48m   v1.31.1
ubuntu@ip-172-31-40-255:~$ |
```

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

Or run kubectl get nodes

```
Last login: Wed Sep 25 03:10:01 2024 from 152.58.42.1
ubuntu@ip-172-31-40-255:~$ kubectl get nodes -o wide
NAME           STATUS    ROLES                  AGE   VERSION
ip-172-31-37-88  Ready    <none>                45m   v1.31.1
ip-172-31-40-255  Ready    node1,control-plane  48m   v1.31.1
ubuntu@ip-172-31-40-255:~$ |
```

Advance Devops Experiment 4

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

Application.

Theory: What is kubectl?

kubectl is the command-line tool for interacting with Kubernetes clusters. It allows you to manage Kubernetes resources by creating, updating, and deleting pods, deployments, services, and more.

Prerequisites

A Kubernetes cluster running either locally (e.g., with Minikube, Kind, or Docker Desktop) or remotely (cloud-based, such as Google Kubernetes Engine (GKE), Amazon Elastic Kubernetes Service (EKS), or Azure Kubernetes Service (AKS)).

kubectl installed on your local machine to interact with the cluster.

Step 1:

Go to AWS Academia in services select EC2 and create 3 instance with instance type t2.medium and names as node1, node2 and master

The screenshot shows the AWS EC2 'Launch an instance' wizard. The process is at the 'Summary' step, where the user has specified 1 instance. The configuration includes:

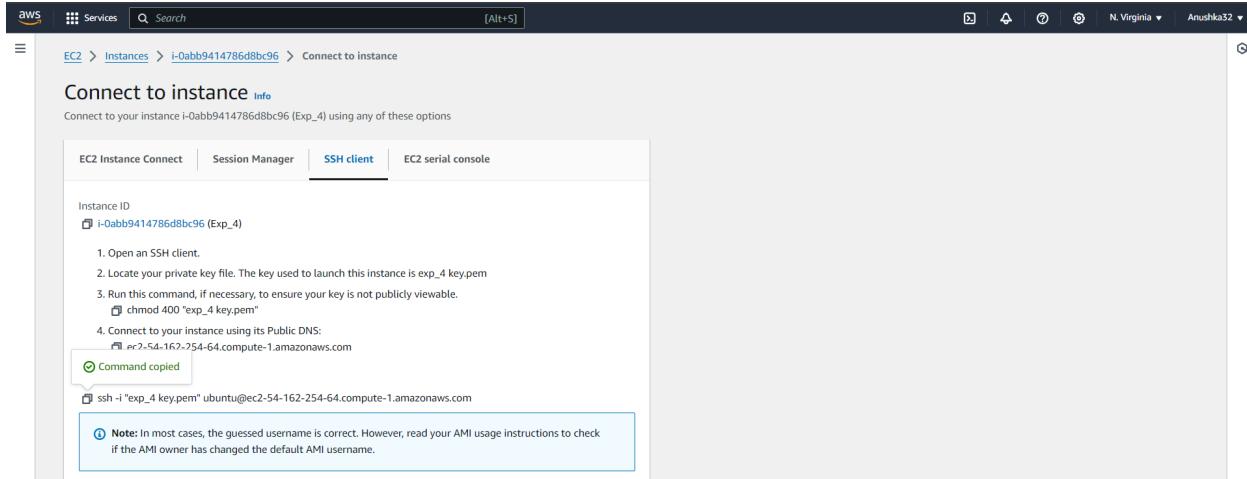
- Software Image (AMI):** Canonical, Ubuntu, 24.04, amd64 (ami-0e86e20dae9224db8)
- Virtual server type (instance type):** t2.medium
- Firewall (security group):** New security group
- Storage (volumes):** 1 volume(s) - 8 GiB

A callout box highlights the **Free tier:** In your first year, it 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 IOPS, and 1 TB of Amazon S3 storage.

At the bottom right of the summary step, there are 'Cancel', 'Launch instance', and 'Review commands' buttons. A green success message at the bottom indicates the launch was successful: "Success Successfully initiated launch of instance (i-0abb9414786d8b96)".

Step 2: Create a new key pair and name it as myKey1 and download as .pem file.

Open command prompt run the following command



Step 3: Now open the folder in the terminal where our .pem key is stored and paste the Example command (starting with ssh -i) in the terminal.([ssh -i "Master_Ec2_Key.pem" ubuntu@ec2-54-196-129-215.compute-1.amazonaws.com](ssh -i \))

```
System information as of Tue Sep 24 21:28:51 UTC 2024

System load: 0.08      Processes:          117
Usage of /: 22.8% of 6.71GB  Users logged in:    0
Memory usage: 5%           IPv4 address for enX0: 172.31.86.116
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-172-31-86-116:~$ |
```

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

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

```
ubuntu@ip-172-31-86-116:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -  
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee  
/etc/apt/trusted.gpg.d/docker.gpg > /dev/null  
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu  
$(lsb_release -cs) stable"  
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).  
OK  
-----BEGIN PGP PUBLIC KEY BLOCK-----  
  
mQINBFit2ioBEADhWpZ8/wvZ6hUTiX0wQHXMAlaFHcPH9hAtr4F1y2+OYdbtMuth  
lqqwp028AqyY+PRFVmTSYMbjuQuu5byyKR01BbqYhuS3jtqQmljZ/bJvXqnmiVXh  
38Uula+z077PxxyQhu5BbqntTPQMfiyqEiu+BKbq2WmANUKQf+iAmZY/IruOXbnq  
L4C1+gJ8vfmxQt99npCaxEjaNRVYfOS8QcixNzHUYNb6emjlANYEvLZzeqo7XKL7  
UrwV5inawTSzWNvtjEjj4nJL8NsLwscplPQUhTQ+7BbQXAwAmeHCUTQIvvWXqw0N  
cmhh4HgeQscQHYg0JJjDVfoY5MucvglbIgCafzAHW9jxmRL4qbMZj+b1XoePEht  
ku4bIQN1X5P07fNWz1gaRL5Z4POXDDZTLIQ/EI58j9kp4bnWRCJW0lya+f8ocodo  
vZz+Doi+fy4D5ZGrL4XEcIQP/Lv5uFyf+kQtL/94VFYYJ0leAv8W92KdgDkhTCD  
G7c0tIkVEKNUq48b3aQ64NOZQW7fVjfoKwEZd0qPE72Pa45jrZzvUFxSpdiNk2tZ  
XYukHjlxxEgBdC/J3cMMNRE1F4NCA3ApfvV1Y7/hTeOnmDuDYwr9/obA8t016Yljj  
q5rdkywPf4JF8mXUW5eCn1vAFHxeg9ZlwmhBtQmGxXnw9M+Z6hWwca6ahmwARAQAB  
tCtEb2NrZXIGUmVsZWFrZSAoQ0UgZGVikSA8ZG9ja2VyQGRvY2tlci5jb20+iQI3  
BBMBCgAhB0JYrefAAhsyB0sJCAcDBRUkC0qlBRYCAwEAh4BAheAAAoJEI2BqDw0
```

sudo apt-get update

sudo apt-get install -y docker-ce

```
/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.  
ubuntu@ip-172-31-86-116:~$ sudo apt-get update  
sudo apt-get install -y docker-ce  
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease  
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease  
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease  
Hit:4 https://download.docker.com/linux/ubuntu noble InRelease  
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease  
Reading package lists... Done  
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  containerd.io docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0  
  pigz slirp4netns  
Suggested packages:  
  aufs-tools cgroupfs-mount | cgroup-lite  
The following NEW packages will be installed:  
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7  
  libslirp0 pigz slirp4netns  
0 upgraded, 10 newly installed, 0 to remove and 139 not upgraded.  
Need to get 123 MB of archives.  
After this operation, 442 MB of additional disk space will be used.  
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]  
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libltdl7 amd64 2.4.7-7build1 [40.3 kB]
```

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

```
ubuntu@ip-172-31-86-116:~$ sudo mkdir -p /etc/docker  
cat <<EOF | sudo tee /etc/docker/daemon.json  
{  
  "exec-opts": ["native.cgroupdriver=systemd"]  
}  
EOF  
{  
  "exec-opts": ["native.cgroupdriver=systemd"]  
}  
ubuntu@ip-172-31-86-116:~$ sudo systemctl enable docker  
sudo systemctl daemon-reload  
sudo systemctl restart docker  
Synchronizing state of docker.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.  
Executing: /usr/lib/systemd/systemd-sysv-install enable docker  
ubuntu@ip-172-31-86-116:~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o  
/etc/apt/keyrings/kubernetes-apt-keyring.gpg  
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]  
https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list  
gpg: missing argument for option "--o"  
-bash: /etc/apt/keyrings/kubernetes-apt-keyring.gpg: No such file or directory  
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]  
https://pkgs.k8s.io/core:/stable:/v1.31/deb/  
ubuntu@ip-172-31-86-116:~$ sudo apt-get update  
sudo apt-get install -y kubelet kubeadm kubectl  
sudo apt-mark hold kubelet kubeadm kubectl  
E: Malformed entry 1 in list file /etc/apt/sources.list.d/kubernetes.list (URI)
```

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

```
ubuntu@ip-172-31-86-116:~$ sudo systemctl enable docker  
sudo systemctl daemon-reload  
sudo systemctl restart docker  
Synchronizing state of docker.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.  
Executing: /usr/lib/systemd/systemd-sysv-install enable docker  
ubuntu@ip-172-31-86-116:~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o  
/etc/apt/keyrings/kubernetes-apt-keyring.gpg  
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]  
https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list  
gpg: missing argument for option "--o"  
-bash: /etc/apt/keyrings/kubernetes-apt-keyring.gpg: No such file or directory  
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]  
https://pkgs.k8s.io/core:/stable:/v1.31/deb/  
ubuntu@ip-172-31-86-116:~$ sudo apt-get update  
sudo apt-get install -y kubelet kubeadm kubectl  
sudo apt-mark hold kubelet kubeadm kubectl  
E: Malformed entry 1 in list file /etc/apt/sources.list.d/kubernetes.list (URI)  
E: The list of sources could not be read.  
E: Malformed entry 1 in list file /etc/apt/sources.list.d/kubernetes.list (URI)  
E: The list of sources could not be read.  
E: Malformed entry 1 in list file /etc/apt/sources.list.d/kubernetes.list (URI)  
E: The list of sources could not be read.  
ubuntu@ip-172-31-86-116:~$  
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o  
/etc/apt/keyrings/kubernetes-apt-keyring.gpg  
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
```

Step 5: Run the below command to install Kubernets.

```
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor  
-o  
/etc/apt/keyrings/kubernetes-apt-keyring.gpg  
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]  
https://pkgs.k8s.io/core:/stable:/v1.31/deb/ ' | sudo tee  
/etc/apt/sources.list.d/kubernetes.list  
sudo apt-get update  
sudo apt-get install -y kubelet kubeadm kubectl  
sudo apt-mark hold kubelet kubeadm kubectl
```

```
ubuntu@ip-172-31-86-116:~$ sudo mkdir -p /etc/apt/keyrings  
ubuntu@ip-172-31-86-116:~$ sudo apt-get update  
sudo apt-get install -y kubelet kubeadm kubectl  
sudo apt-mark hold kubelet kubeadm kubectl  
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease  
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease  
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease  
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease  
Hit:5 https://download.docker.com/linux/ubuntu noble InRelease  
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb InRelease [1186 B]  
Get:7 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb Packages [4865 B]  
Fetched 6051 B in 1s (11.3 kB/s)  
Reading package lists... Done  
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  conntrack cri-tools kubernetes-cni  
The following NEW packages will be installed:  
  conntrack cri-tools kubeadm kubectl kubernetes-cni  
0 upgraded, 6 newly installed, 0 to remove and 139 not upgraded.  
Need to get 87.4 MB of archives.  
After this operation, 314 MB of additional disk space will be used.  
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 conntrack amd64 1:1.4.8-1ubuntu1 [37.9 kB]  
Get:2 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb cri-tools 1.31.1-1.1 [15.7 MB]  
Get:3 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb kubeadm 1.31.1-1.1 [11.4 M]
```

```
ubuntu@ip-172-31-86-116:~$ kubectl describe pod nginx-deployment-d556bf558-dmdsr  
Name:           nginx-deployment-d556bf558-dmdsr  
Namespace:      default  
Priority:       0  
Service Account: default  
Node:           <none>  
Labels:          app=nginx  
                 pod-template-hash=d556bf558  
Annotations:    <none>  
Status:         Pending  
IP:               
IPs:            <none>  
Controlled By: ReplicaSet/nginx-deployment-d556bf558
```

Anushka Shahane D15A 55

```
ubuntu@ip-172-31-86-116:~$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
error: flag needs an argument: 'f' in -f
See 'kubectl apply --help' for usage.
-bash: https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml: No such file or directory
ubuntu@ip-172-31-86-116:~$ 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
ubuntu@ip-172-31-86-116:~$ kubectl apply -f https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
ubuntu@ip-172-31-86-116:~$ kubectl get pods
NAME           READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-dmdsr   0/1     Pending   0          10s
nginx-deployment-d556bf558-kf9l4   0/1     Pending   0          10s
ubuntu@ip-172-31-86-116:~$ POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath=".items[0].metadata.name")
kubectl port-forward $POD_NAME 8081:80
error: unable to forward port because pod is not running. Current status=Pending
ubuntu@ip-172-31-86-116:~$ kubectl taint nodes --all node-role.kubernetes.io/control-plane-node/ip-172-31-20-171 untainted
error: at least one taint update is required
ubuntu@ip-172-31-86-116:~$ kubectl get nodes
NAME            STATUS   ROLES      AGE      VERSION
ip-172-31-86-116   Ready    control-plane   4m15s   v1.31.1
ubuntu@ip-172-31-86-116:~$ |
```

```
sudo systemctl enable --now kubelet
sudo kubeadm init --pod-network-cidr=10.244.0.0/16
```

```
ubuntu@ip-172-31-86-116:~$ sudo apt-get install -y socat
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz
  slirp4netns
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  socat
0 upgraded, 1 newly installed, 0 to remove and 139 not upgraded.
Need to get 374 kB of archives.
After this operation, 1649 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 socat amd64 1.8.0.0-4build3 [374 kB]
Fetched 374 kB in 0s (14.0 MB/s)
Selecting previously unselected package socat.
(Reading database ... 68108 files and directories currently installed.)
Preparing to unpack .../socat_1.8.0.0-4build3_amd64.deb ...
Unpacking socat (1.8.0.0-4build3) ...
Setting up socat (1.8.0.0-4build3) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.
```

```
sudo mkdir -p /etc/containerd
sudo containerd config default | sudo tee /etc/containerd/config.toml
```

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

```
uid = 0
ubuntu@ip-172-31-86-116:~$ sudo systemctl restart containerd
sudo systemctl enable containerd
sudo systemctl status containerd
● containerd.service - containerd container runtime
   Loaded: loaded (/usr/lib/systemd/system/containerd.service; enabled; preset: enabled)
     Active: active (running) since Tue 2024-09-24 21:37:08 UTC; 220ms ago
       Docs: https://containerd.io
   Main PID: 4764 (containerd)
      Tasks: 7
     Memory: 13.4M (peak: 14.0M)
        CPU: 55ms
      CGroup: /system.slice/containerd.service
              └─4764 /usr/bin/containerd

Sep 24 21:37:08 ip-172-31-86-116 containerd[4764]: time="2024-09-24T21:37:08.940185812Z" level=info msg="Start subscrib>
Sep 24 21:37:08 ip-172-31-86-116 containerd[4764]: time="2024-09-24T21:37:08.940228819Z" level=info msg="Start recoveri>
Sep 24 21:37:08 ip-172-31-86-116 containerd[4764]: time="2024-09-24T21:37:08.940274910Z" level=info msg="Start event mo>
Sep 24 21:37:08 ip-172-31-86-116 containerd[4764]: time="2024-09-24T21:37:08.940284757Z" level=info msg="Start snapshot>
Sep 24 21:37:08 ip-172-31-86-116 containerd[4764]: time="2024-09-24T21:37:08.940292060Z" level=info msg="Start cni netw>
Sep 24 21:37:08 ip-172-31-86-116 containerd[4764]: time="2024-09-24T21:37:08.940298061Z" level=info msg="Start streamin>
Sep 24 21:37:08 ip-172-31-86-116 containerd[4764]: time="2024-09-24T21:37:08.940298336Z" level=info msg=serving... addr>
Sep 24 21:37:08 ip-172-31-86-116 containerd[4764]: time="2024-09-24T21:37:08.941160760Z" level=info msg=serving... addr>
Sep 24 21:37:08 ip-172-31-86-116 systemd[1]: Started containerd.service - containerd container runtime.
Sep 24 21:37:08 ip-172-31-86-116 containerd[4764]: time="2024-09-24T21:37:08.941884864Z" level=info msg="containerd suc>
```

Step 6: Initialize the Kubecluster

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

```
ubuntu@ip-172-31-86-116:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
W0924 21:37:32.938984    4937 checks.go:846] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent with that used by kubeadm. It is recommended to use "registry.k8s.io/pause:3.10" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-86-116 kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 172.31.86.116]
[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-172-31-86-116 localhost] and IPs [172.31.86.116 127.0.0.1 :1]
[certs] Generating "etcd/peer" certificate and key
```

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

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

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

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

Anushka Shahane D15A 55

```
ubuntu@ip-172-31-86-116:~$ mkdir -p $HOME/.kube
    sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
    sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@ip-172-31-86-116:~$ kubectl apply -f
https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
error: flag needs an argument: 'f' in -f
See 'kubectl apply --help' for usage.
-bash: https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml: No such file or directory
ubuntu@ip-172-31-86-116:~$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-f
lannel.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
ubuntu@ip-172-31-86-116:~$ kubectl apply -f https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
ubuntu@ip-172-31-86-116:~$ kubectl get pods
NAME           READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-dmdsr  0/1     Pending   0          10s
nginx-deployment-d556bf558-kf9l4  0/1     Pending   0          10s
ubuntu@ip-172-31-86-116:~$ POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath=".items[0].metadata.name")
POD_NAME=$
```

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

kubectl apply -f

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

```
ubuntu@ip-172-31-86-116:~$ kubectl apply -f
https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
error: flag needs an argument: 'f' in -f
See 'kubectl apply --help' for usage.
-bash: https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml: No such file or directory
ubuntu@ip-172-31-86-116:~$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-f
lannel.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
ubuntu@ip-172-31-86-116:~$ kubectl apply -f https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
ubuntu@ip-172-31-86-116:~$ kubectl get pods
NAME           READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-dmdsr  0/1     Pending   0          10s
nginx-deployment-d556bf558-kf9l4  0/1     Pending   0          10s
ubuntu@ip-172-31-86-116:~$ POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath=".items[0].metadata.name")
kubectl port-forward $POD_NAME 8081:80
error: unable to forward port because pod is not running. Current status=Pending
ubuntu@ip-172-31-86-116:~$ kubectl taint nodes --all node-role.kubernetes.io/control-plane-node/ip-172-31-20-171 untaint
ed
error: at least one taint update is required
ubuntu@ip-172-31-86-116:~$ kubectl get nodes
NAME           STATUS   ROLES      AGE      VERSION
ip-172-31-86-116 Ready   control-plane  4m15s   v1.31.1
ubuntu@ip-172-31-86-116:~$ |
```

Step 7: Now that the cluster is up and running, we can deploy our nginx server on this cluster. Apply this deployment file using this command to create a deployment
kubectl apply -f <https://k8s.io/examples/application/deployment.yaml>

Anushka Shahane D15A 55

```
ubuntu@ip-172-31-86-116:~$ POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath=".items[0].metadata.name")  
kubectl port-forward $POD_NAME 8081:80  
error: unable to forward port because pod is not running. Current status=Pending  
ubuntu@ip-172-31-86-116:~$ kubectl taint nodes --all node-role.kubernetes.io/control-plane-node/ip-172-31-20-171 untainted  
error: at least one taint update is required  
ubuntu@ip-172-31-86-116:~$ kubectl get nodes  
NAME STATUS ROLES AGE VERSION  
ip-172-31-86-116 Ready control-plane 4m15s v1.31.1  
ubuntu@ip-172-31-86-116:~$ kubectl get pods  
NAME READY STATUS RESTARTS AGE  
nginx-deployment-d556bf558-dmdsr 0/1 Pending 0 2m52s  
nginx-deployment-d556bf558-kf9l4 0/1 Pending 0 2m52s  
ubuntu@ip-172-31-86-116:~$ kubectl get pods  
NAME READY STATUS RESTARTS AGE  
  
heduling.  
ubuntu@ip-172-31-86-116:~$ kubectl taint nodes --all node-role.kubernetes.io/control-plane-node/ip-172-31-86-116 untainted  
ubuntu@ip-172-31-86-116:~$ kubectl get pods  
NAME READY STATUS RESTARTS AGE  
nginx-deployment-d556bf558-dmdsr 1/1 Running 0 6m29s  
nginx-deployment-d556bf558-kf9l4 1/1 Running 0 6m29s  
ubuntu@ip-172-31-86-116:~$ |
```

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

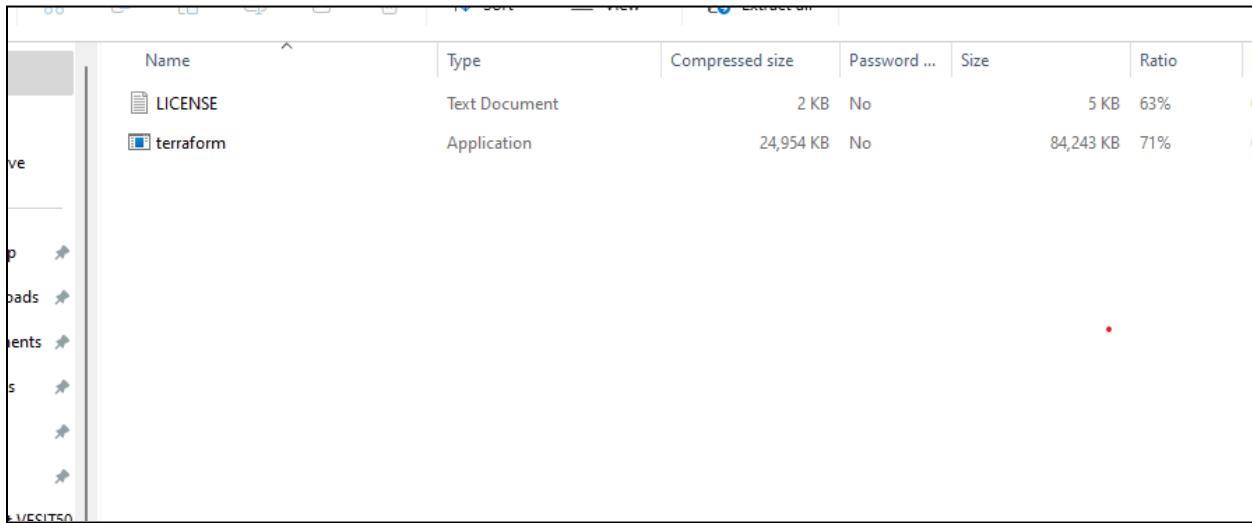
```
nginx-deployment-d556bf558-dmdsr 1/1 Running 0 6m29s  
ubuntu@ip-172-31-86-116:~$ POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath=".items[0].metadata.name")  
kubectl port-forward $POD_NAME 8080:80  
Forwarding from 127.0.0.1:8080 -> 80  
Forwarding from [::1]:8080 -> 80  
|
```

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

```
Last login: Tue Sep 24 21:28:52 2024 from 103.88.83.126  
ubuntu@ip-172-31-86-116:~$ curl --head http://127.0.0.1:8080  
HTTP/1.1 200 OK  
Server: nginx/1.14.2  
Date: Tue, 24 Sep 2024 21:48:41 GMT  
Content-Type: text/html  
Content-Length: 612  
Last-Modified: Tue, 04 Dec 2018 14:44:49 GMT  
Connection: keep-alive  
ETag: "5c0692e1-264"  
Accept-Ranges: bytes  
ubuntu@ip-172-31-86-116:~$ |
```

Experiment no. 5

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



About Terraform
Define cloud and on-prem resources in human-readable configuration files that you can version, reuse, and share.

Featured docs

- Introduction to Terraform
- Configuration Language
- Terraform CLI
- HCP Terraform
- Provider Use

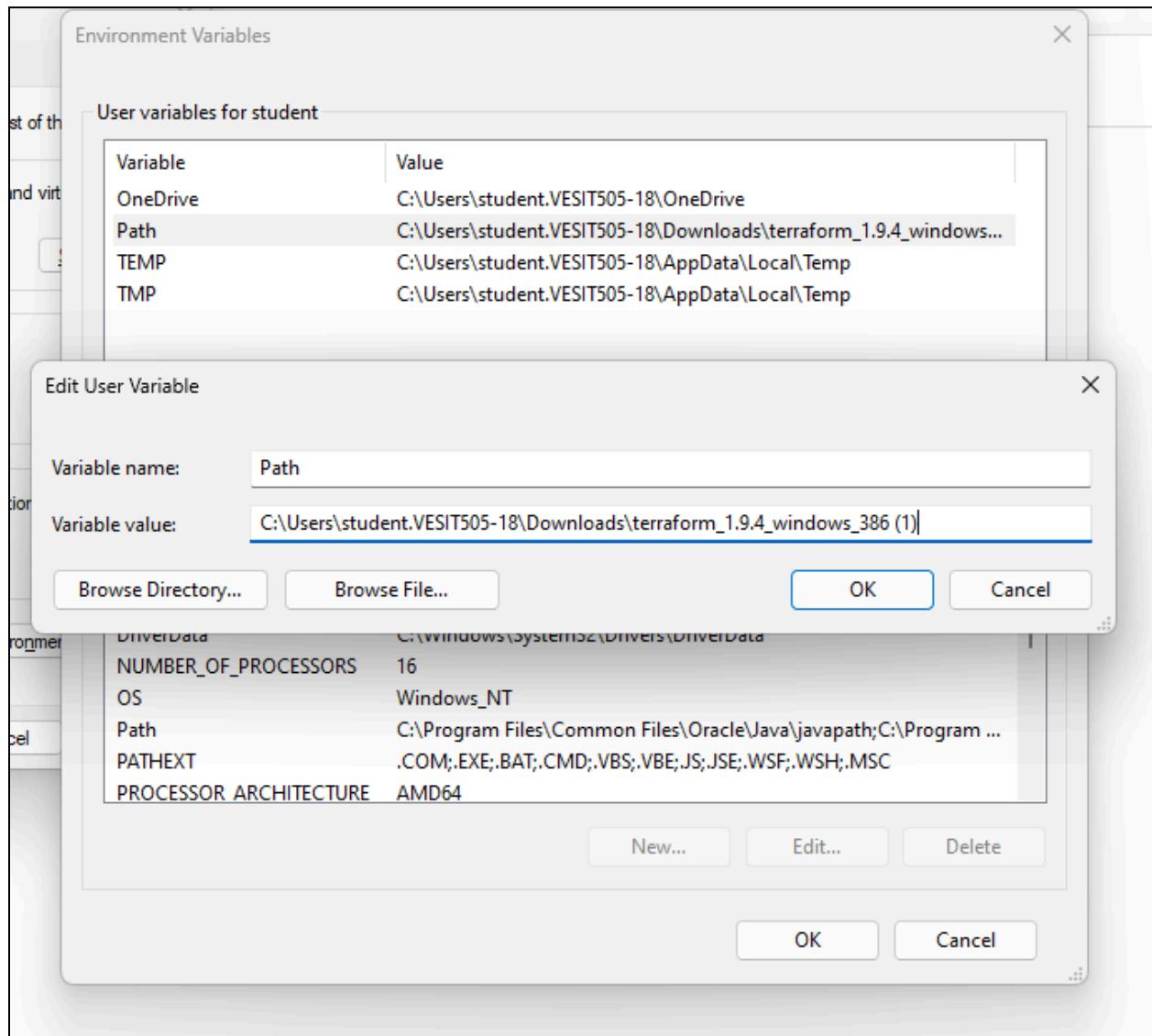
Windows
Binary download

386 Version: 1.9.4	Download	AMD64 Version: 1.9.4	Download
-----------------------	--------------------------	-------------------------	--------------------------

Linux
Package manager

Ubuntu/Debian	CentOS/RHEL	Fedora	Amazon Linux	Homebrew
-------------------------------	-----------------------------	------------------------	------------------------------	--------------------------

HCP Terraform
Automate your infrastructure



```
C:\Users\student.VESIT505-18>terraform
Usage: terraform [global options] <subcommand> [args]

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

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

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

```
C:\Users\student.VESIT505-18>terraform --version
Terraform v1.9.4
on windows_386
```

```
C:\Users\student.VESIT505-18>
```

ADVANCE DEVOPS EXPERIMENT 6

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

Part A: Creating docker image using terraform

Step 1:Check Docker functionality

```
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\student>docker

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

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

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

Check for the docker version with the following command.

```
C:\Users\student>docker --version
Docker version 27.1.1, build 6312585

C:\Users\student>
```

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

Step 2:

Creating a new folder named ‘Docker’ in the ‘TerraformScripts’ folder.

Creating a new docker.tf file using Atom editor and write the following contents into.

This will create a Ubuntu Linux container

```
vim docker.tf  X
"vim docker.tf
1  terraform {
2    required_providers {
3      docker = {
4        source  = "kreuzwerker/docker"
5        version = "2.21.0"
6      }
7    }
8  }
9
10 provider "docker" {
11   host = "npipe:///./pipe/docker_engine"
12 }
13
14 # Pull the image
15 resource "docker_image" "ubuntu" {
16   name = "ubuntu:latest"
17 }
18
19 # Create a container
20 resource "docker_container" "foo" {
21   image = docker_image.ubuntu.image_id
22   name  = "foo"
23   command = ["sleep", "3600"]
24 }
25 |
```

Step 3: Execute Terraform Init command to initialize the resources

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

Terraform has been successfully initialized!

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

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

Step 4: Execute Terraform plan to see the available resources

```
● PS C:\Users\Admin\TerraformScripts\ Docker> terraform plan
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the fol
+ 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
    + remove_volumes = true
    + restart       = "no"
    + rm            = false
}
```

```
+ runtime      = (known after apply)
+ security_opts = (known after apply)
+ shm_size     = (known after apply)
+ start        = true
+ stdin_open   = false
+ stop_signal  = (known after apply)
+ stop_timeout = (known after apply)
+ tty          = false

+ healthcheck (known after apply)

+ labels (known after apply)
}

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

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

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

```
● PS C:\Users\Admin\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
}
```

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

+ healthcheck (known after apply)

+ labels (known after apply)
}

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

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

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

Enter a value: yes
```

```
docker_image.ubuntu: Creating...
docker_image.ubuntu: Creation complete after 9s [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Creating...
docker_container.foo: Creation complete after 2s [id=01adf07e5918931fee9b90073726a03671037923dd92032ce0e15bbb764a6f24]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
```

Before Executing Apply step:

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

After Executing Apply step:

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

Step 6: Execute Terraform destroy to delete the configuration, which will automatically delete the Ubuntu Container.

```
● PS C:\Users\Admin\TerraformScripts\ Docker> terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Refreshing state... [id=01adf07e5918931fee9b90073726a03671037923dd92032ce0e15bbb764a6f24]

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         = "01adf07e5918931fee9b90073726a03671037923dd92032ce0e15bbb764a6f24" -> null
  - id               = "01adf07e5918931fee9b90073726a03671037923dd92032ce0e15bbb764a6f24" -> null
  - image             = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> null
  - init              = false -> null
  - ip_address        = "172.17.0.2" -> null
  - ip_prefix_length = 16 -> null
  - ipc_mode          = "private" -> null
  - links             = [] -> null
  - log_driver         = "json-file" -> null
  - log_opts           = {} -> null
  - logs              = false -> null
  - max_retry_count   = 0 -> null
}
```

```
- memory           = 0 -> null
- memory_swap     = 0 -> null
- must_run         = true -> null
- name             = "foo" -> null
- network_data     = [
  - {
    - gateway          = "172.17.0.1"
    - global_ipv6_prefix_length = 0
    - ip_address        = "172.17.0.2"
    - ip_prefix_length  = 16
    - network_name      = "bridge"
    # (2 unchanged attributes hidden)
  },
  ] -> null
- network_mode     = "default" -> null
- privileged        = false -> null
- publish_all_ports = false -> null
- read_only         = false -> null
- remove_volumes   = true -> null
- restart           = "no" -> null
- rm                = false -> null
- runtime            = "runc" -> null
- security_opts     = [] -> null
- shm_size          = 64 -> null
- start              = true -> null
- stdin_open         = false -> null
- stop_timeout       = 0 -> null
- storage_opts       = {} -> null
- sysctls            = {} -> null
- tmpfs              = {} -> null
- tty                = false -> null
# (8 unchanged attributes hidden)
}
```

```
# docker_image.ubuntu will be destroyed
- resource "docker_image" "ubuntu" {
  - id      = "sha256:edbe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest" -> null
  - image_id = "sha256:edbe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> null
  - latest   = "sha256:edbe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> 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=01adf07e5918931fee9b90073726a03671037923dd92032ce0e15bbb764a6f24]
docker_container.foo: Destruction complete after 0s
docker_image.ubuntu: Destroying... [id=sha256:edbe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_image.ubuntu: Destruction complete after 1s

Destroy complete! Resources: 2 destroyed.
```

Docker images After Executing Destroy step

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

ADVANCE DEVOPS EXPERIMENT 7

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

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

What problems does SAST solve?

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

SAST tools give developers real-time feedback as they code, helping them fix issues before they pass the code to the next phase of the SDLC. This prevents security-related issues from being considered an afterthought.

SAST tools also provide graphical representations of the issues found, from source to sink. These help you navigate the code easier. Some tools point out the exact location of vulnerabilities and highlight the risky code. Tools can also provide in-depth guidance on how to fix issues and the best place in the code to fix them, without requiring deep security domain expertise.

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

Why is SAST important?

Developers dramatically outnumber security staff. It can be challenging for an organization to find the resources to perform code reviews on even a fraction of its applications. A key strength of SAST tools is the ability to analyze 100% of the

codebase. Additionally, they are much faster than manual secure code reviews performed by humans. These tools can scan millions of lines of code in a matter of minutes. SAST tools automatically identify critical vulnerabilities—such as buffer overflows, SQL injection, cross-site scripting, and others—with high confidence. Thus, integrating static analysis into the SDLC can yield dramatic results in the overall quality of the code developed.

What are the key steps to run SAST effectively?

There are six simple steps needed to perform SAST efficiently in organizations that have a very large number of applications built with different languages, frameworks, and platforms.

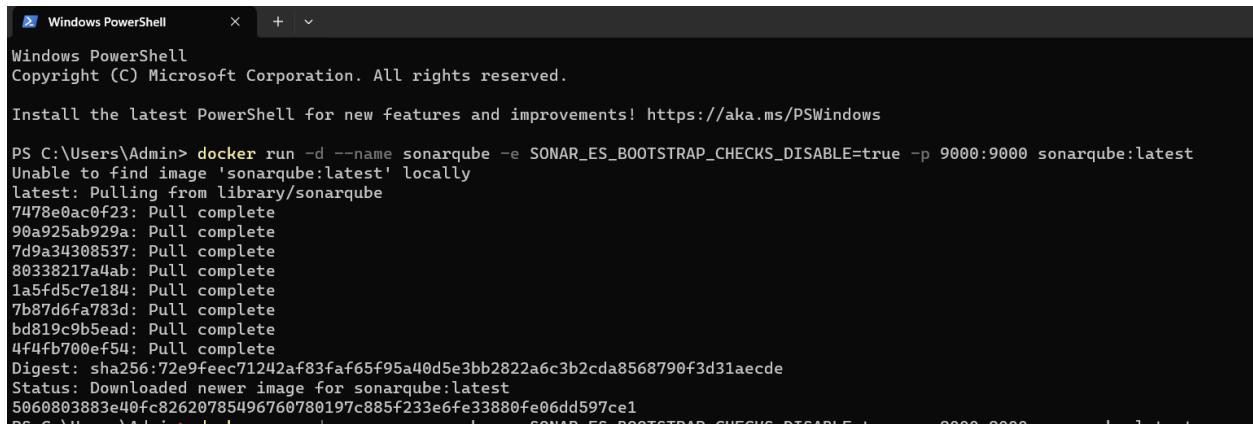
1. **Finalize the tool.** Select a static analysis tool that can perform code reviews of applications written in the programming languages you use. The tool should also be able to comprehend the underlying framework used by your software.
2. **Create the scanning infrastructure, and deploy the tool.** This step involves handling the licensing requirements, setting up access control and authorization, and procuring the resources required (e.g., servers and databases) to deploy the tool.
3. **Customize the tool.** Fine-tune the tool to suit the needs of the organization. For example, you might configure it to reduce false positives or find additional security vulnerabilities by writing new rules or updating existing ones. Integrate the tool into the build environment, create dashboards for tracking scan results, and build custom reports.
4. **Prioritize and onboard applications.** Once the tool is ready, onboard your applications. If you have a large number of applications, prioritize the high-risk applications to scan first. Eventually, all your applications should be onboarded and scanned regularly, with application scans synced with release cycles, daily or monthly builds, or code check-ins.
5. **Analyze scan results.** This step involves triaging the results of the scan to remove false positives. Once the set of issues is finalized, they should be tracked and provided to the deployment teams for proper and timely remediation.

6. Provide governance and training

Proper governance ensures that your development teams are employing the scanning tools properly. The software security touchpoints should be present within the SDLC. SAST should be incorporated as part of your application development and deployment process.

Steps to integrate Jenkins with SonarQube

1. Open up Jenkins Dashboard on localhost, port 8080 .
2. Run SonarQube in a Docker container using this command -



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

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

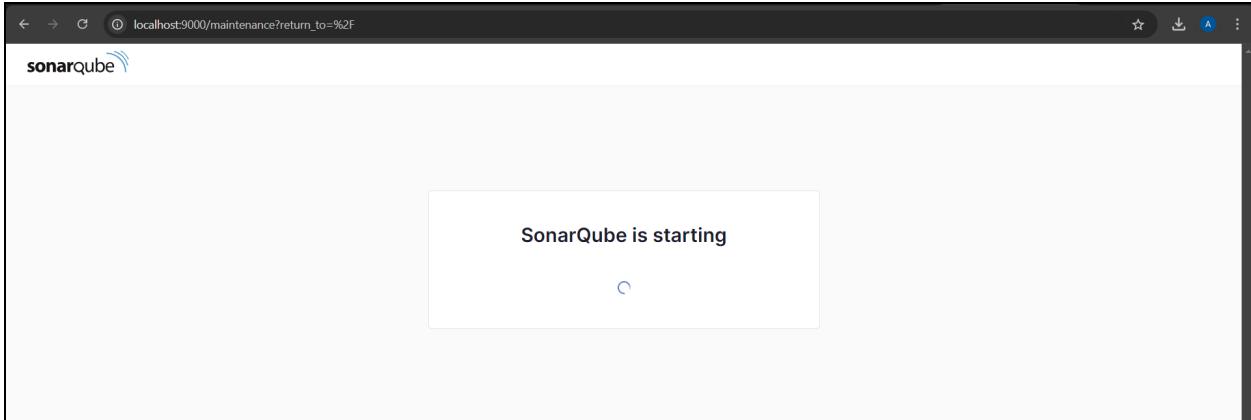
PS C:\Users\Admin> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
5060803883e40fc82620785496760780197c885f233e6fe33880fe06dd597ce1
```

Warning: run below command only once

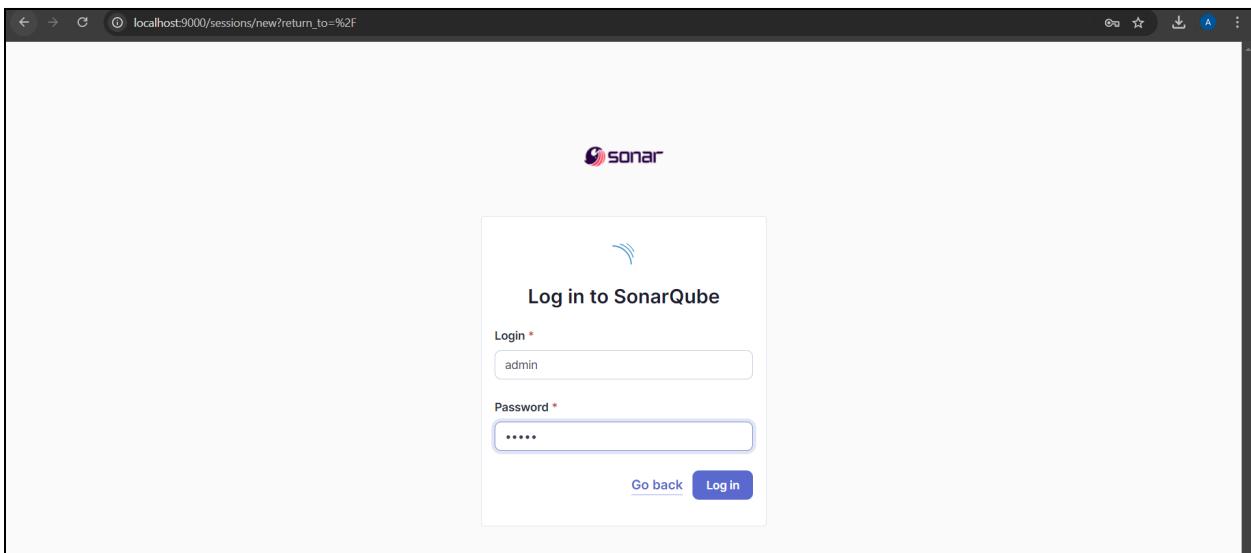
```
docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
```

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

Anushka Shahane D15A 55



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



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

The screenshot shows the SonarQube interface for setting up a new code definition. At the top, there are navigation links: Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, More, and a search bar. Below the header, it says "2 of 2" and "Set up project for Clean as You Code". A note explains that this sets which part of the code will be considered new code, helping to focus on recent changes. It links to "Defining New Code". The main section is titled "Choose the baseline for new code for this project". There are two options: "Use the global setting" (selected) and "Define a specific setting for this project". Under "Define a specific setting for this project", there are three choices: "Previous version" (selected), "Number of days", and "Previous version" again (disabled). The "Previous version" choice is described as any code changed since the previous version being new code, recommended for regular versions or releases.

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 dashboard. At the top, there's a navigation bar with links for "Dashboard", "Search (CTRL+K)", notifications (2), user "Anushka Shahane", and "log out". The main area has a sidebar with links: "+ New Item", "Build History" (selected), "Project Relationship", "Check File Fingerprint", "Manage Jenkins", and "My Views". Below the sidebar is a "Build Queue" section stating "No builds in the queue." To the right is a "Build Executor Status" section showing "built-in node + 2 agents (0 of 2 executors busy)". The central part of the dashboard displays a table of build history for projects: DevOps Pipeline, maven-project, maven_project2, and pipeline2. The table columns include Status (S), Workstation (W), Name, Last Success, Last Failure, and Last Duration.

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀️	DevOps Pipeline	1 mo 24 days #4	N/A	12 sec
✓	☁️	maven-project	1 mo 17 days #4	1 mo 17 days #3	21 sec
✗	🌧️	maven_project2	N/A	28 days #2	21 ms
✓	☀️	pipeline2	28 days #1	N/A	7.9 sec

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

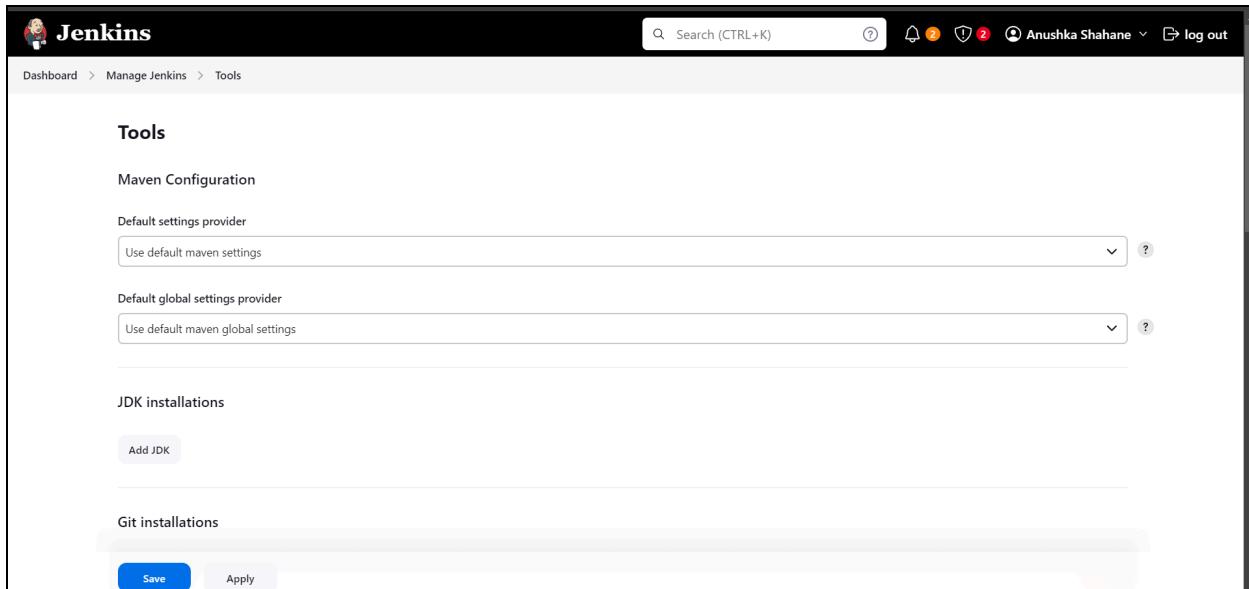
Anushka Shahane D15A 55

The screenshot shows the Jenkins Manage Jenkins interface. On the left, there's a sidebar with links like 'New Item', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins' (which is selected), and 'My Views'. Below this are sections for 'Build Queue' (empty) and 'Build Executor Status' (one build-in node + 2 agents). The main area is titled 'Manage Jenkins' and contains several notifications: one about a Jenkins data directory being almost full, another about a new version (2.462.2) available for download, and a warning about security vulnerabilities in Jenkins 2.452.3. At the bottom, there's a 'System Configuration' section with tabs for 'System', 'Tools', and 'Plugins'.

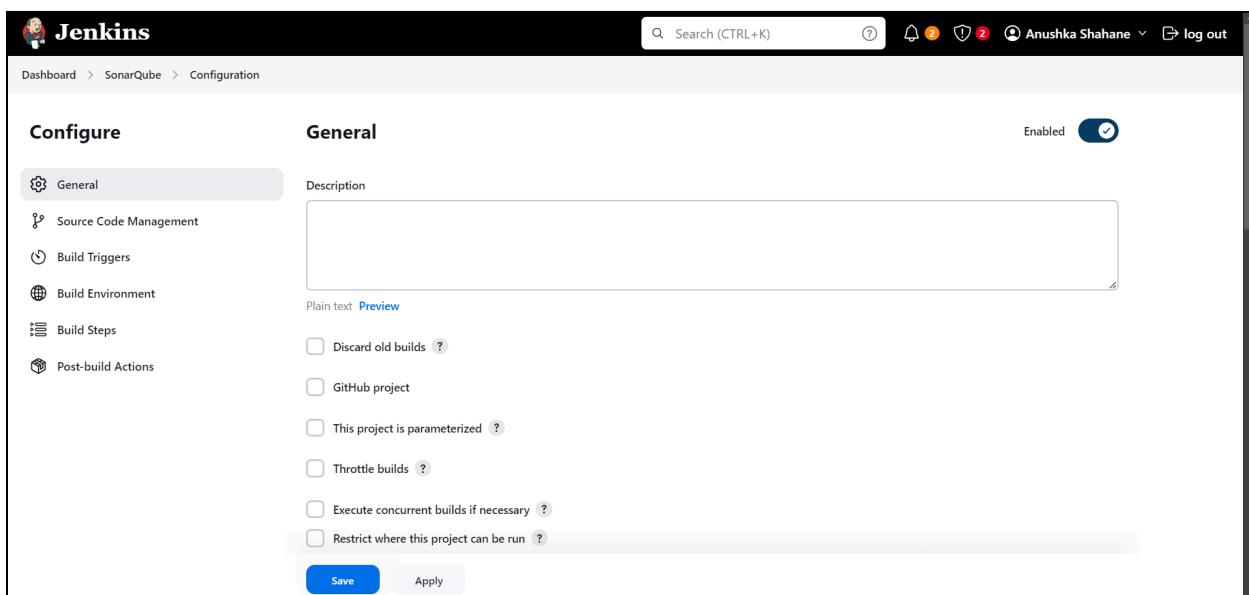
The screenshot shows the Jenkins Plugins page. The left sidebar has links for 'Updates' (29), 'Available plugins', 'Installed plugins', 'Advanced settings', and 'Download progress' (selected). The main content area is titled 'Download progress' and shows the 'Preparation' phase with steps: 'Checking internet connectivity', 'Checking update center connectivity', and 'Success'. It also shows the 'SonarQube Scanner' plugin being loaded, with 'Loading plugin extensions' and 'Success' status. There are two buttons at the bottom: a blue link to 'Go back to the top page' and a grey button to 'Restart Jenkins when installation is complete and no jobs are running'.

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

Anushka Shahane D15A 55



The screenshot shows the Jenkins 'Tools' configuration page. At the top, there are dropdown menus for 'Default settings provider' (set to 'Use default maven settings') and 'Default global settings provider' (set to 'Use default maven global settings'). Below these are sections for 'JDK installations' (with a 'Add JDK' button) and 'Git installations'. At the bottom are 'Save' and 'Apply' buttons.

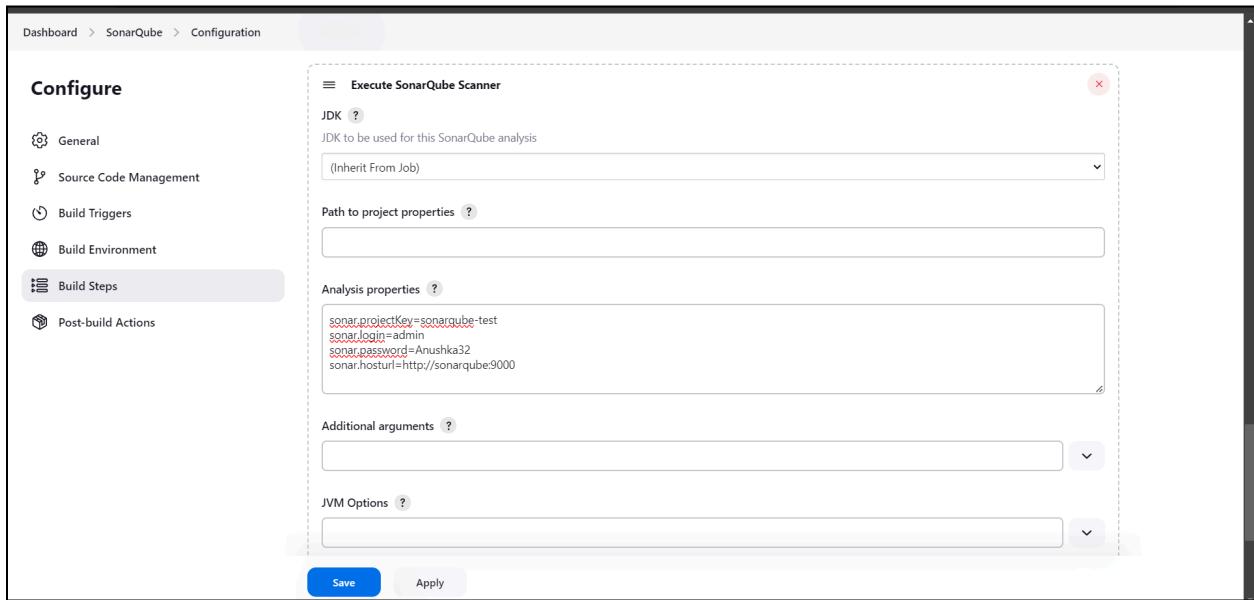


The screenshot shows the Jenkins 'SonarQube Configuration' page under the 'Configure' tab. The 'General' section is selected. It includes a 'Description' field, a 'Plain text' preview area, and several build options: 'Discard old builds', 'GitHub project', 'This project is parameterized', 'Throttle builds', 'Execute concurrent builds if necessary', and 'Restrict where this project can be run'. A 'Save' and 'Apply' button are at the bottom. The 'Enabled' toggle switch is turned on.

9. Choose this GitHub repository in Source Code Management.

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

It is a sample hello-world project with no vulnerabilities and issues,



10. Under Build-> Execute SonarQube Scanner, enter these Analysis properties. Mention the SonarQube Project Key, Login, Password, Source path and Host URL.
11. Go to http://localhost:9000/<user_name>/permissions and allow Execute Permissions to the Admin user.

Anushka Shahane D15A 55

The screenshot shows the SonarQube administration interface for global permissions. At the top, there are tabs for Projects, Issues, Rules, Quality Profiles (which is selected), Quality Gates, Administration, and More. A search bar is also present. Below the tabs, there's a navigation bar with Configuration, Security (selected), Projects, System, and Marketplace. The main section is titled "Global Permissions" with a sub-instruction: "Grant and revoke permissions to make changes at the global level. These permissions include editing Quality Profiles, executing analysis, and performing global system administration." There are three tabs: All, Users, and Groups, with "All" selected. A search bar "Search for users or groups..." is available. Below this, four permission categories are listed: "Administer System", "Administer", "Execute Analysis", and "Create". Under "Administer System", the "Administrator" role "admin" is assigned. Under "Execute Analysis", the "Quality Gates" and "Quality Profiles" checkboxes are checked, while "Projects" is unchecked. 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 upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine." At the bottom right, it says "Community Edition v10.6 (92116) ACTIVE" and provides links for LGPL v3, Community, Documentation, Plugins, and Web API.

Check the console output.

The screenshot shows the Jenkins console output for a build. The left sidebar includes links for Dashboard, Status, Changes, Console Output (which is selected), View as plain text, Edit Build Information, Timings, Git Build Data, and Previous Build. The main area is titled "Console Output" and shows the command-line output of the build process. The output starts with "Started by user Anushka Shahane" and continues with various git commands for cloning and fetching from a GitHub repository. It also shows the configuration of the SonarScanner and the execution of the scanner. The Jenkins interface includes a search bar, user profile, and log out link at the top right.

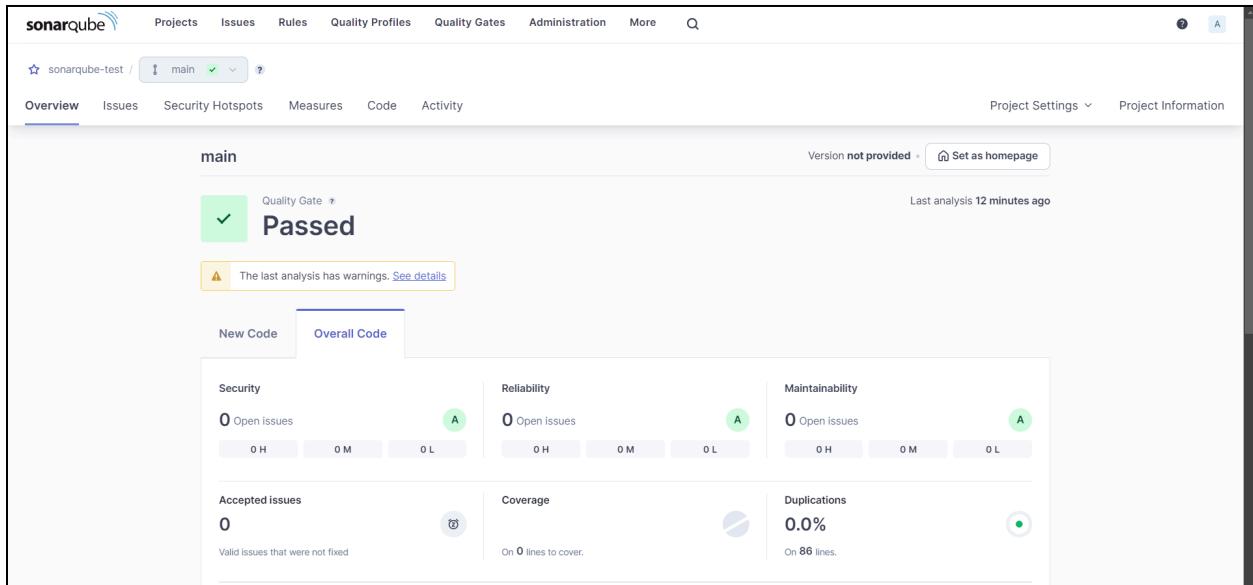
```

Dashboard > SonarQube > #16 > Console Output

00:09:43.074 INFO 14 source files to be analyzed
00:09:43.536 INFO 14/14 source files have been analyzed
00:09:43.538 INFO Sensor TextAndSecretsSensor [text] (done) | time=681ms
00:09:43.557 INFO ----- Run sensors on project
00:09:43.938 INFO Sensor C# [csharp]
00:09:43.936 WARNING Your project contains C# files which cannot be analyzed with the scanner you are using. To analyze C# or VB.NET, you must use the SonarScanner for .NET 5.x or higher, see https://redirect.sonarsource.com/doc/install-configure-scanner-msbuild.html
00:09:43.938 INFO Sensor C# [csharp] (done) | time=ms
00:09:43.939 INFO Sensor Analysis Warnings import [csharp]
00:09:43.941 INFO Sensor Analysis Warnings import [csharp] (done) | time=2ms
00:09:43.942 INFO Sensor C# File Caching Sensor [csharp]
00:09:43.951 WARNING Incremental PR analysis: Could not determine common base path, cache will not be computed. Consider setting 'sonar.projectBaseDir' property.
00:09:43.952 INFO Sensor C# File Caching Sensor [csharp] (done) | time=9ms
00:09:43.953 INFO Sensor Zero Coverage Sensor
00:09:44.036 INFO Sensor Zero Coverage Sensor (done) | time=84ms
00:09:44.042 INFO SCM Publisher SCM provider for this project is: git
00:09:44.043 INFO SCM Publisher 4 source files to be analyzed
00:09:45.208 INFO SCM Publisher 4/4 source files have been analyzed (done) | time=1146ms
00:09:45.208 INFO CPD Executor Calculating CPD for 0 files
00:09:45.233 INFO CPD Executor CPD calculation finished (done) | time=1ms
00:09:45.252 INFO SCM revision ID: 'f2bc042c04ce72427c380bcbee6d6fee7b49adf'
00:09:46.446 INFO Analysis report generated in 845ms, dir size=201.8 kB
00:09:46.554 INFO Analysis report compressed in 77ms, zip size=22.1 kB
00:09:48.616 INFO Analysis report uploaded in 205ms
00:09:48.633 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube-test
00:09:48.634 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
00:09:48.634 INFO More about the report processing at http://localhost:9000/api/ce/task?id=56715393-077a-4755-994a-bace670afb66
00:09:48.685 INFO Analysis total time: 1:25.553 s
00:09:48.717 INFO SonarScanner Engine completed successfully
00:09:48.900 INFO EXECUTION SUCCESS
00:09:49.053 INFO Total time: 1:52.816s
Finished: SUCCESS

```

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



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

Anushka Shahane D15A 55

EXPERIMENT 8 - ADVANCE DEVOPS

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 /

Theory:

What is SAST?

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

What problems does SAST solve?

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

without breaking builds or passing on vulnerabilities to the final release of the application.

SAST tools give developers real-time feedback as they code, helping them fix issues before they

pass the code to the next phase of the SDLC. This prevents security-related issues from being

considered an afterthought. SAST tools also provide graphical representations of the issues

found, from source to sink. These help you navigate the code easier. Some tools point out the

exact location of vulnerabilities and highlight the risky code. Tools can also provide in-depth

guidance on how to fix issues and the best place in the code to fix them, without requiring deep

security domain expertise.

It's important to note that SAST tools must be run on the application on a regular basis, such as

during daily/monthly builds, every time code is checked in, or during a code release.

Why is SAST important?

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

What is a CI/CD Pipeline?

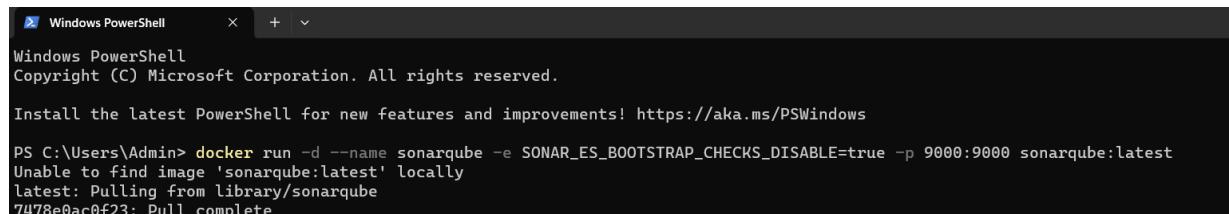
What is SonarQube

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

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

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

1. Open up Jenkins Dashboard on localhost, port 8080 or whichever port it is at for you.
2. Run SonarQube in a Docker container using this command -



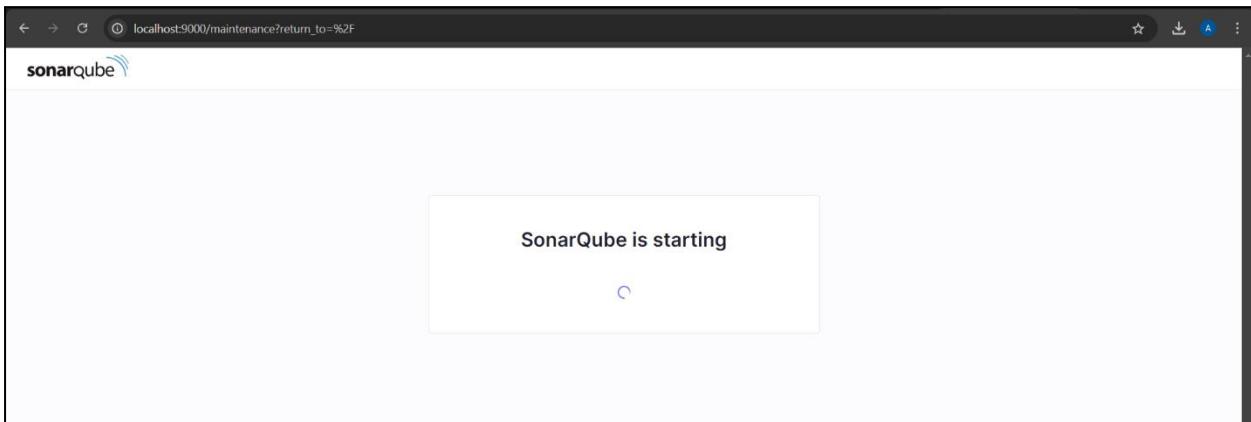
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

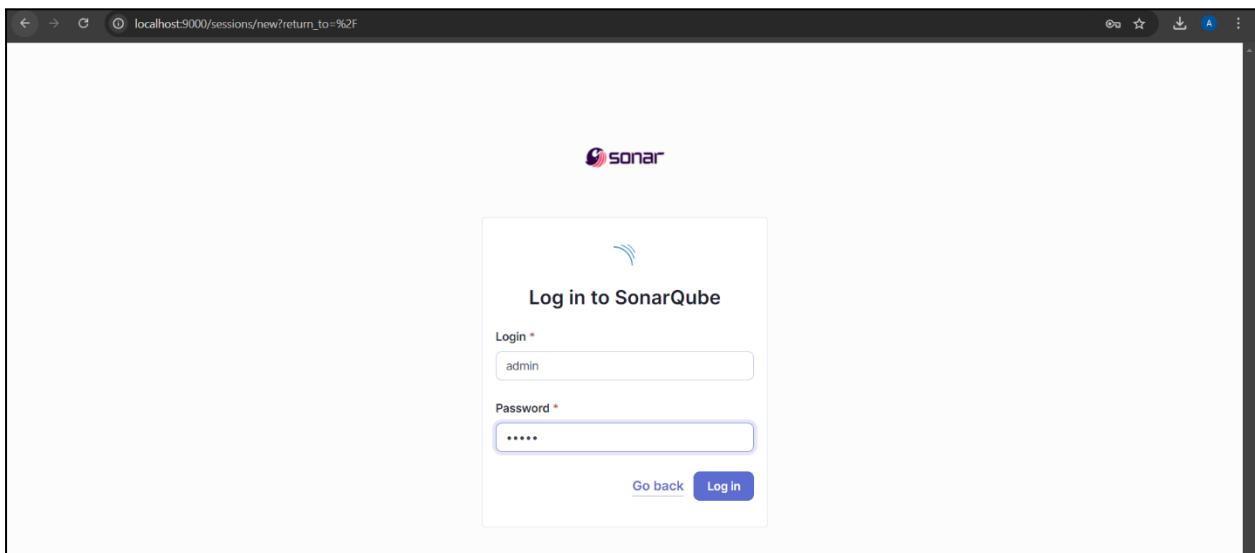
PS C:\Users\Admin> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
```

Anushka Shahane D15A 55

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.



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

The screenshot shows the SonarQube interface for setting up a new code definition. At the top, there's a navigation bar with links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, More, and a search bar. Below the header, it says "2 of 2" and "Set up project for Clean as You Code". A note explains that this helps focus on recent changes. There are two main options for defining new code: "Use the global setting" (selected) and "Define a specific setting for this project". Under the specific setting, there are two choices: "Previous version" (selected) and "Number of days". Both options have explanatory text below them.

The screenshot shows the Jenkins interface for creating a new job. The title bar says "Jenkins". The main area has a search bar and a "Dashboard > All" link. A "Required field" message is displayed above a text input field where "sonarqube-test" is typed. Below the input field, there are five project types listed with their descriptions: "Freestyle project", "Maven project", "Pipeline", "Multi-configuration project", and "Folder". Each type has a corresponding icon and a brief description. At the bottom of the dialog, there are "OK" and "Cancel Pipeline" buttons.

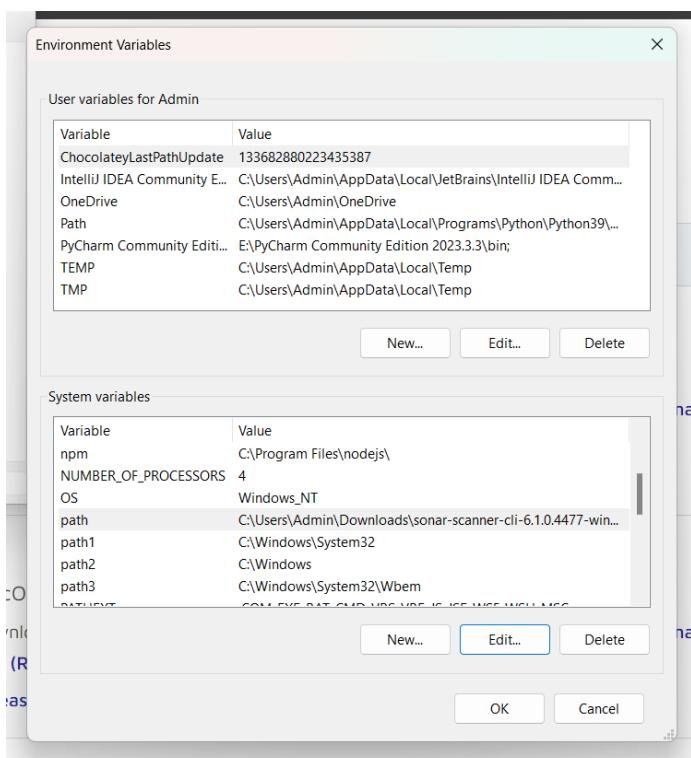
Step 6 : Go to [download sonarscanner](#) to download sonar scanner

The screenshot shows the SonarScanner CLI page on the SonarQube website. The left sidebar includes links for SonarQube analysis overview, Project analysis setup, Scanners, Scanner environment, SonarScanner CLI, SonarQube extension for Azure DevOps, SonarQube extension for Jenkins, SonarScanner for .NET, SonarScanner for Maven, SonarScanner for Gradle, SonarScanner for NPM, SonarScanner for Ant (Deprecated), SonarScanner for Python (Beta), and Analysis parameters. The main content area displays three releases of the SonarScanner CLI:

- 6.2** (2024-09-17): Support PKCS12 truststore generated with OpenSSL. Download scanner for: Linux x64, Linux AArch64, Windows x64, macOS x64, macOS AArch64, Docker, Any (Requires a pre-installed JVM). Release notes.
- 6.1** (2024-06-27): macOS and Linux AArch64 distributions. Download scanner for: Linux x64, Linux AArch64, Windows x64, macOS x64, macOS AArch64, Docker, Any (Requires a pre-installed JVM). Release notes.
- 6.0** (2024-06-04): New bootstrapping mechanism and JRE provisioning with SonarQube 10.6+ and SonarCloud. Download scanner for: Linux x64, Windows x64, macOS x64, Docker, Any (Requires a pre-installed JVM).

A red box highlights the "Windows x64" link under the 6.1 release.

Step 7: After the download is complete, extract the file and copy the path to bin folder
Go to environment variables, system variables and click on path
Add a new path, paste the path copied earlier.



Step 8 : Save the pipeline and build it.

Pipeline

Definition

Pipeline script

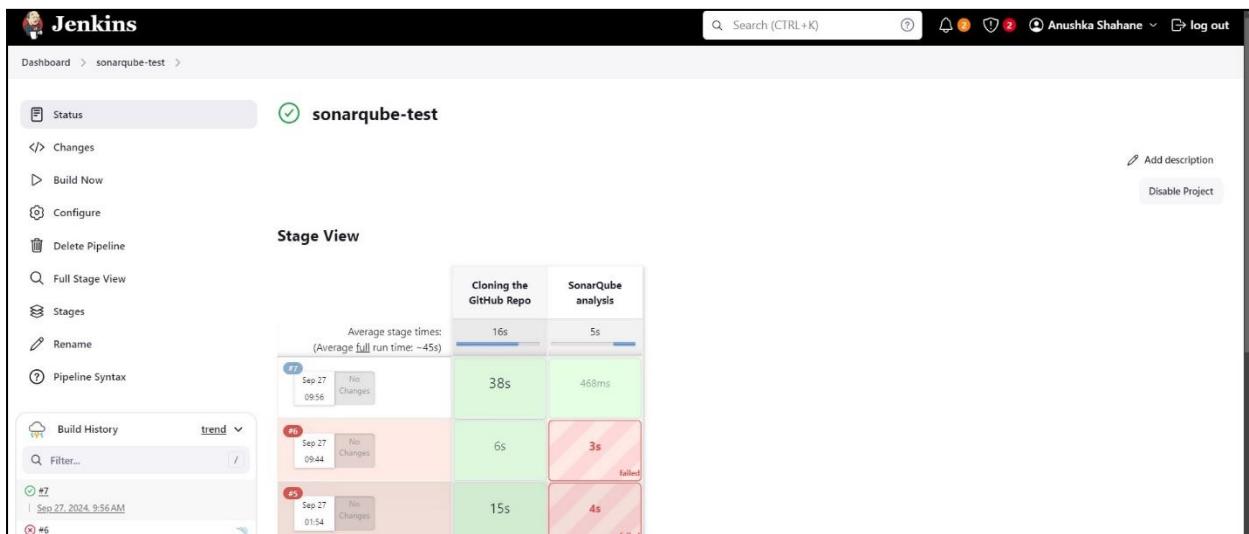
```
1 node {  
2   stage('Cloning the GitHub Repo') {  
3     git 'https://github.com/shazforiot/GOL.git'  
4   }  
5  
6   stage('SonarQube analysis') {  
7     withSonarQubeEnv('sonarqube') {  
8       bat """  
9         C:/Users/Admin/Downloads/sonar-scanner-cli-6.1.0.4477-windows-x64/sonar-scanner-6.1.0.4477-windows-x64/bin/sonar-scanner.bat ^  
10        -D sonar.login=admin ^  
11        -D sonar.password=Anushka32 ^  
12        -D sonar.projectKey=sonarqube-test ^  
13        -D sonar.exclusions=vendor/**,resources/**,**/*.java ^  
14        -D sonar.host.url=http://localhost:9000/  
15      """  
16    }  
17  }  
18}
```

Use Groovy Sandbox ?

Pipeline Syntax

Save Apply

Output :

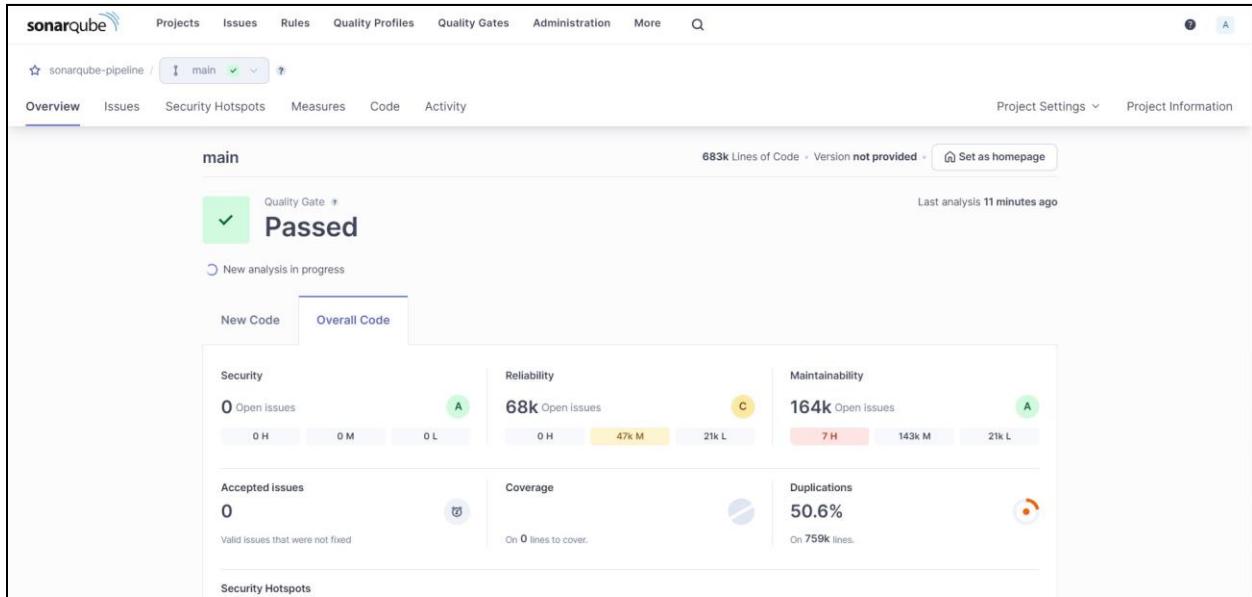


Anushka Shahane D15A 55

The screenshot shows the Jenkins interface. On the left, there's a sidebar with links like Status, Changes, Console Output (which is selected), View as plain text, Edit Build Information, Delete build #7, Timings, Git Build Data, Pipeline Overview, Pipeline Console, Replay, Pipeline Steps, and Workspaces. The main area is titled "Console Output" and shows the command-line logs for build #7. The logs indicate the build was started by user Anushka Shahane, and it runs on Jenkins in C:\ProgramData\Jenkins\jenkins\workspace\sonarqube-test. It shows the git clone process from https://github.com/shazforiot/GOL.git, fetching upstream changes, and performing a git checkout to revision ba799ba7e1b576f04a4612322b0412c5e6ele5e4. The logs conclude with a message: "Started by user Anushka Shahane [Pipeline] Start of Pipeline [Pipeline] node Running on Jenkins in C:\ProgramData\Jenkins\jenkins\workspace\sonarqube-test [Pipeline] { [Pipeline] stage [Pipeline] { (Cloning the GitHub Repo) [Pipeline] git The recommended git tool is: NONE No credentials specified > C:\Program Files\Git\bin\git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\jenkins\workspace\sonarqube-test\.git # timeout=10 Fetching changes from the remote Git repository > C:\Program Files\Git\bin\git.exe config remote.origin.url https://github.com/shazforiot/GOL.git # timeout=10 Fetching upstream changes from https://github.com/shazforiot/GOL.git > C:\Program Files\Git\bin\git.exe --version # timeout=10 > git --version # 'git' version 2.46.2.windows.1" > C:\Program Files\Git\bin\git.exe fetch --tags --force --progress -- https://github.com/shazforiot/GOL.git +refs/heads/*:refs/remotes/origin/* # timeout=10 > C:\Program Files\Git\bin\git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10 Checking out Revision ba799ba7e1b576f04a4612322b0412c5e6ele5e4 (refs/remotes/origin/master) > C:\Program Files\Git\bin\git.exe config core.sparsecheckout # timeout=10 > C:\Program Files\Git\bin\git.exe checkout -f ba799ba7e1b576f04a4612322b0412c5e6ele5e4 # timeout=10 > C:\Program Files\Git\bin\git.exe branch -a -v --no-abbrev # timeout=10

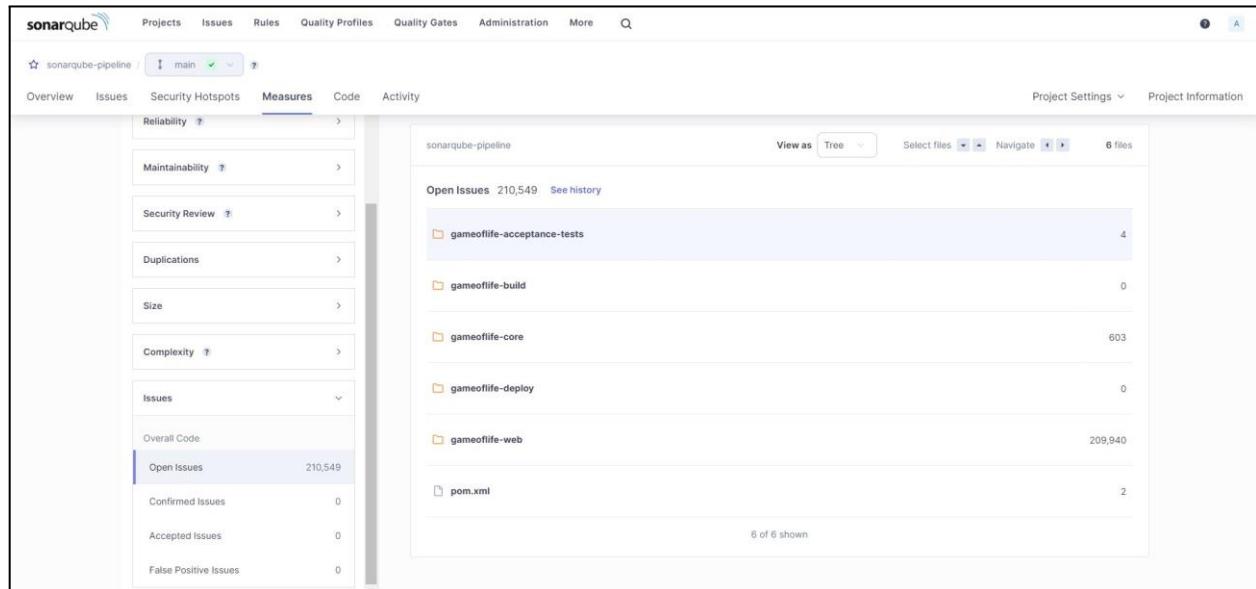
```
20:50:01.832 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube-pipeline
20:50:01.832 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
20:50:01.832 INFO More about the report processing at http://localhost:9000/api/ce/task?id=159a9d05-1f5f-4e17-bd27-3643a32a836a
20:50:12.108 INFO Analysis total time: 7:37.235 s
20:50:12.110 INFO SonarScanner Engine completed successfully
20:50:12.849 INFO EXECUTION SUCCESS
20:50:12.851 INFO Total time: 7:44.878s
[Pipeline]
[Pipeline] // withSonarQubeEnv
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Step 9 : Check the project in SonarQube



The screenshot shows the SonarQube interface for the 'sonarqube-pipeline / main' project. The top navigation bar includes 'Projects', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', 'Administration', 'More', and a search bar. Below the navigation is a breadcrumb trail: 'sonarqube-pipeline / main'. The main content area has tabs for 'Overview', 'Issues', 'Security Hotspots', 'Measures', 'Code', and 'Activity'. The 'Overview' tab is selected. It displays a large green 'Passed' badge with a checkmark. Below it, there's a note about a 'New analysis in progress'. The dashboard is divided into several sections: 'Security' (0 Open issues), 'Reliability' (68k Open issues), 'Maintainability' (164k Open issues), 'Accepted issues' (0), 'Coverage' (On 0 lines to cover), and 'Duplications' (50.6% on 759k lines). At the bottom, there's a 'Security Hotspots' section.

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



The screenshot shows the SonarQube interface for the 'sonarqube-pipeline' project, specifically the 'Measures' tab. The top navigation bar and breadcrumb trail are identical to the previous screenshot. The 'Measures' tab is selected. On the left, there's a sidebar with various metrics: Reliability, Maintainability, Security Review, Duplications, Size, Complexity, Issues, Overall Code (Open Issues: 210,549), Confirmed Issues (0), Accepted Issues (0), and False Positive Issues (0). The main content area shows a detailed list of open issues under the 'sonarqube-pipeline' project. The list includes: gameoflife-acceptance-tests (4 issues), gameoflife-build (0 issues), gameoflife-core (603 issues), gameoflife-deploy (0 issues), gameoflife-web (209,940 issues), and pom.xml (2 issues). There are buttons for 'View as Tree', 'Select files', 'Navigate', and a file count of '6 files'.

Anushka Shahane D15A 55

SonarQube Issues page for project gameoflife-core. The left sidebar shows navigation tabs: Overview, Issues (selected), Security Hotspots, Measures, Code, Activity. Project settings and information are at the top right. The main area displays a list of issues under the 'Intentionality' category. There are two sections of results from 'gameoflife-core/build/reports/tests/all-tests.html':

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

Below these, another section from 'gameoflife-core/build/reports/tests/allclasses-frame.html' shows:

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

The sidebar also includes filters for Clean Code Attribute (Consistency, Intentionality, Adaptability, Responsibility) and Software Quality (Security, Reliability, Maintainability).

SonarQube Issues page for project gameoflife-acceptance-tests. The left sidebar shows navigation tabs: Overview, Issues (selected), Security Hotspots, Measures, Code, Activity. Project settings and information are at the top right. The main area displays a list of issues under the 'Maintainability' category. There are four sections of results from 'gameoflife-acceptance-tests/Dockerfile':

- Use a specific version tag for the image.** (Maintainability)
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.** (Maintainability)
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.** (Maintainability)
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.** (Maintainability)

The sidebar also includes filters for Clean Code Attribute (Consistency, Intentionality, Adaptability, Responsibility) and Software Quality (Security, Reliability, Maintainability).

Anushka Shahane D15A 55

SonarQube Issues page for project sonarcube-pipeline. The left sidebar shows a summary of quality metrics: Security (0), Reliability (253), and Maintainability (15). The main area displays a list of 15 issues under the 'gameoflife-acceptance-tests/Dockerfile' file. Each issue is a code smell related to Dockerfiles:

- L1: Use a specific version tag for the image. (Maintainability)
- L1: Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. (Maintainability)
- L1: Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. (Maintainability)
- L1: Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. (Maintainability)

Project Settings and Bulk Change buttons are visible at the top right.

SonarQube Security Hotspots page for project sonarcube-pipeline. The left sidebar shows 3 security hotspots. The main area details a critical hotspot in the 'gameoflife-web/Dockerfile' file:

Status: To review
This security hotspot needs to be reviewed to assess whether the code poses a risk.

Review priority: Medium

Permission: The tomcat image runs with root as the default user. Make sure it is safe here.

Encryption of Sensitive Data: None shown.

Others: None shown.

The right side of the screen shows the Dockerfile content with the highlighted line: `FROM tomcat:8.0-jre8`. A note below it states: "The tomcat image runs with root as the default user. Make sure it is safe here."

Anushka Shahane D15A 55

The screenshot shows the SonarQube interface for the project "sonarqube-pipeline". The "Measures" tab is selected. On the left, a sidebar displays metrics: Density (50.6%), Duplicated Lines (384,007), Duplicated Blocks (42,808), and Duplicated Files (979). The main panel shows a table for "Duplicated Lines (%)" with 50.6% overall. It lists files with their respective duplicated lines percentages: gameoflife-acceptance-tests (0.0%), gameoflife-build (0.0%), gameoflife-core (9.6%, 374 lines), gameoflife-deploy (0.0%), gameoflife-web (50.9%, 383,633 lines), and pom.xml (0.0%).

	Duplicated Lines (%)	Duplicated Lines
gameoflife-acceptance-tests	0.0%	0
gameoflife-build	0.0%	0
gameoflife-core	9.6%	374
gameoflife-deploy	0.0%	0
gameoflife-web	50.9%	383,633
pom.xml	0.0%	0

The screenshot shows the SonarQube interface for the project "sonarqube-pipeline". The "Measures" tab is selected. On the left, a sidebar displays metrics: Density (50.6%), Duplicated Lines (384,007), Duplicated Blocks (42,808), and Duplicated Files (979). The main panel shows a table for "Cyclomatic Complexity" with 1,112 overall. It lists files with their respective complexity values: gameoflife-acceptance-tests (—), gameoflife-build (—), gameoflife-core (18), gameoflife-deploy (—), gameoflife-web (1,094), and pom.xml (—).

	Cyclomatic Complexity
gameoflife-acceptance-tests	—
gameoflife-build	—
gameoflife-core	18
gameoflife-deploy	—
gameoflife-web	1,094
pom.xml	—

Advance devops Experiment No. 9

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

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.

Features of Nagios

Following are the important features of Nagios monitoring tool:

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

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

1. Create an Amazon Linux EC2 Instance

- Name it nagios-host.

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like EC2 Dashboard, EC2 Global View, Events, Console-to-Code Preview, Instances, Instances Types, Launch Templates, and Spot Requests. The main area is titled 'Instances (1/5) Info'. It lists five instances: 'Exp_4', 'Master', 'node2', 'nagios-host', and 'node1'. The 'nagios-host' instance is selected, indicated by a checked checkbox in the first column. Its details are shown in the table: Name (nagios-host), Instance ID (i-0d0e623c6c1f8a773), Instance state (Running), Instance type (t2.micro), Status check (2/2 checks passed), Alarm status (View alarms +), Availability Zone (us-east-1c), and Public IP (ec2-35-17-). There are buttons for Connect, Actions, and Launch instances.

2. Configure Security Group

- Ensure HTTP, HTTPS, SSH, and ICMP are open from everywhere.
- Edit the inbound rules of the specified Security Group

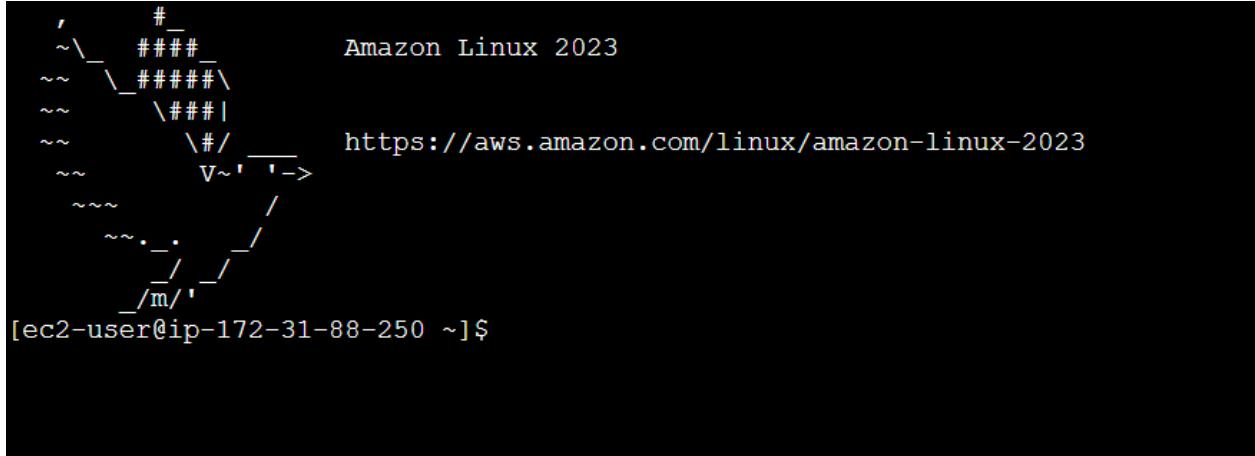
The screenshot shows the AWS Security Groups page. A security group named 'sgr-086670fe3fbbae4a68' is selected. The table lists seven inbound rules:

- HTTP (TCP, port 80, Anywhere, 0.0.0.0/0)
- All ICMP - IPv6 (IPv6 ICMP, All, Anywhere, ::/0)
- HTTPS (TCP, port 443, Anywhere, 0.0.0.0/0)
- All traffic (All, All, Anywhere, 0.0.0.0/0)
- SSH (TCP, port 22, Anywhere, 0.0.0.0/0)
- Custom TCP (TCP, port 5666, Anywhere, 0.0.0.0/0)
- All ICMP - IPv4 (ICMP, All, Anywhere, 0.0.0.0/0)

Each rule has a 'Delete' button next to it.

3. Connect to Your EC2 Instance

- SSH into your EC2 instance or use EC2 Instance Connect from the browser



The image shows the Amazon Linux 2023 logo, which is a stylized tree made of hashtags (#). To the right of the tree, the text "Amazon Linux 2023" is displayed. Below the tree, there is a URL: "https://aws.amazon.com/linux/amazon-linux-2023". At the bottom left, it says "[ec2-user@ip-172-31-88-250 ~]\$".

4. Update Package Indices and Install Required Packages

Commands -

```
sudo yum update
```

```
sudo yum install httpd php
```

```
sudo yum install gcc glibc glibc-common
```

```
sudo yum install gd gd-devel
```

Anushka Shahane D15A 55

```
① Keyboard shortcut
To tab out of the terminal window and select the next button element, press the left and right Shift keys together.

[m/`]
[ec2-user@ip-172-31-80-174 ~]$ sudo yum update
Last metadata expiration check: 0:15:44 ago on Tue Oct 1 04:26:43 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-80-174 ~]$ sudo yum install httpd php
Last metadata expiration check: 0:16:34 ago on Tue Oct 1 04:26:43 2024.
Dependencies resolved.

Package           Architecture     Version      Repository   Size
Installing:
httpd            x86_64          2.4.62-1.amzn2023
php8_3           x86_64          8.3.10-1.amzn2023.0.1

Installing dependencies:
apr              x86_64          1.7.2-2.amzn2023.0.2
apr-util         x86_64          1.6.3-1.amzn2023.0.1
generic-logos-httd
noarch
18.0.0-12.amzn2023.0.3
httpd-core       x86_64          2.4.62-1.amzn2023
httpd-filesystem noarch
2.4.62-1.amzn2023
httpd-tools      x86_64          2.4.62-1.amzn2023
libbrotli        x86_64          1.0.9-4.amzn2023.0.2
libsodium         x86_64          1.0.19-4.amzn2023
libxml2          x86_64          1.1.34-5.amzn2023.0.2
mailcap          noarch
2.1.49-3.amzn2023.0.3
nginx-filesystem noarch
1.1.24.0-1.amzn2023.0.4
php8_3-clli      x86_64          8.3.10-1.amzn2023.0.1
php8_3-common    x86_64          8.3.10-1.amzn2023.0.1

amazonlinux      48 kB
amazonlinux      10 kB
amazonlinux      129 kB
amazonlinux      98 kB
amazonlinux      19 kB
amazonlinux      1.4 MB
amazonlinux      14 kB
amazonlinux      81 kB
amazonlinux      315 kB
amazonlinux      176 kB
amazonlinux      241 kB
amazonlinux      33 kB
amazonlinux      9.8 kB
amazonlinux      3.7 MB
amazonlinux      737 kB

: ODO0dddee33dde6db (nagios-host) X
```

```
① Keyboard shortcut
To tab out of the terminal window and select the next button element, press the left and right Shift keys together.

Verifying : libxml2-1.1.34-5.amzn2023.0.2.x86_64
Verifying : mailcap-2.1.49-3.amzn2023.0.3.noarch
Verifying : mod_http-2.0.27-1.amzn2023.0.3.x86_64
Verifying : nginx-filesystem-1:2.4.0-1.amzn2023.0.4.noarch
Verifying : php8_3-8.3.10-1.amzn2023.0.1.x86_64
Verifying : php8_3-clli-8.3.10-1.amzn2023.0.1.x86_64
Verifying : php8_3-common-8.3.10-1.amzn2023.0.1.x86_64
Verifying : php8_3-fpm-8.3.10-1.amzn2023.0.1.x86_64
Verifying : php8_3-mbstring-8.3.10-1.amzn2023.0.1.x86_64
Verifying : php8_3-opcache-8.3.10-1.amzn2023.0.1.x86_64
Verifying : php8_3-pdo-8.3.10-1.amzn2023.0.1.x86_64
Verifying : php8_3-process-8.3.10-1.amzn2023.0.1.x86_64
Verifying : php8_3-sodium-8.3.10-1.amzn2023.0.1.x86_64
Verifying : php8_3-xml-8.3.10-1.amzn2023.0.1.x86_64

11/25
12/25
13/25
14/25
15/25
16/25
17/25
18/25
19/25
20/25
21/25
22/25
23/25
24/25
25/25

Installing:
apr-1.7.2-2.amzn2023.0.2.x86_64
generic-logos-httd-2.4.62-1.amzn2023.0.3.noarch
httpd-filesystem-2.4.62-1.amzn2023.noarch
libsodium-1.0.19-4.amzn2023.x86_64
mod_http-2.0.27-1.amzn2023.0.3.x86_64
php8_3-8.3.10-1.amzn2023.0.1.x86_64
php8_3-fpm-8.3.10-1.amzn2023.0.1.x86_64
php8_3-pdo-8.3.10-1.amzn2023.0.1.x86_64
php8_3-xml-8.3.10-1.amzn2023.0.1.x86_64

apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
httpd-core-2.4.62-1.amzn2023.x86_64
libbrotli-1.0.9-4.amzn2023.0.2.x86_64
mailcap-2.1.49-3.amzn2023.0.3.noarch
nginx-filesystem-1:2.4.0-1.amzn2023.0.4.noarch
php8_3-common-8.3.10-1.amzn2023.0.1.x86_64
php8_3-opcache-8.3.10-1.amzn2023.0.1.x86_64
php8_3-sodium-8.3.10-1.amzn2023.0.1.x86_64

Complete!
[ec2-user@ip-172-31-80-174 ~]$ i-0d08dddee33dde6db (nagios-host) X
```

```
① Keyboard shortcut
To tab out of the terminal window and select the next button element, press the left and right Shift keys together.

(50/62): libwebp-devel-1.2.4-1.amzn2023.0.6.x86_64.rpm
(51/62): libxml2-devel-2.10.4-4.amzn2023.0.6.x86_64.rpm
(52/62): libtiff-devel-4.4.0-4.amzn2023.0.18.x86_64.rpm
(53/62): libxcb-devel-1.13.1-1.amzn2023.0.2.x86_64.rpm
(54/62): pcre2-devel-10.40-1.amzn2023.0.3.x86_64.rpm
(55/62): pcre2-utf16-10.40-1.amzn2023.0.3.x86_64.rpm
(56/62): pcre2-utf32-10.40-1.amzn2023.0.3.x86_64.rpm
(57/62): sysprof-capture-devel-3.40.1-2.amzn2023.0.2.x86_64.rpm
(58/62): pixman-0.40.0-3.amzn2023.0.3.x86_64.rpm
(59/62): xml-common-0.6.3-56.amzn2023.0.2.noarch.rpm
(60/62): xz-devel-5.2.5-9.amzn2023.0.2.x86_64.rpm
(61/62): zlib-devel-1.2.11-33.amzn2023.0.5.x86_64.rpm
(62/62): xorg-x11proto-devel-2021.4-1.amzn2023.0.2.noarch.rpm

Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing :
Installing : zlib-devel-1.2.11-33.amzn2023.0.5.x86_64
Installing : libpng-2:1.6.37-10.amzn2023.0.6.x86_64
Installing : libwebp-1.2.4-1.amzn2023.0.6.x86_64
Installing : libjpeg-turbo-2.1.4-2.amzn2023.0.5.x86_64
Installing : cmake-filesystem-3.22.2-1.amzn2023.0.4.x86_64
Installing : libpng-devel-2:1.6.37-10.amzn2023.0.6.x86_64
Installing : xorg-x11proto-devel-2021.4-1.amzn2023.0.2.noarch
Installing : libicu-67.1-7.amzn2023.0.3.x86_64 [=====]

644 kB/s | 37 kB 00:00
14 MB/s | 500 kB 00:00
3.8 MB/s | 516 kB 00:00
15 MB/s | 1.0 MB 00:00
13 MB/s | 473 kB 00:00
5.8 MB/s | 216 kB 00:00
3.7 MB/s | 205 kB 00:00
2.7 MB/s | 60 kB 00:00
5.2 MB/s | 295 kB 00:00
1.4 MB/s | 32 kB 00:00
2.5 MB/s | 53 kB 00:00
2.1 MB/s | 45 kB 00:00
3.6 MB/s | 263 kB 00:00

24 MB/s | 23 MB 00:00
1/62
2/62
3/62
4/62
5/62
6/62
7/62
8/62

i-0d08dddee33dde6db (nagios-host) X
PublicIPs: 35.173.200.53 PrivateIPs: 172.31.80.174
```

5. Create a New Nagios User

Commands -

```
sudo adduser -m nagios
```

```
sudo passwd nagios
```

```
Last login: Mon Sep 30 17:45:34 2024 from 18.206.107.27
[ec2-user@ip-172-31-88-250 ~]$ sudo adduser -m nagios
adduser: user 'nagios' already exists
[ec2-user@ip-172-31-88-250 ~]$ sudo passwd nagios
Changing password for user nagios.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
Sorry, passwords do not match.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-88-250 ~]$ █
```

6. Create a New User Group

Commands -

```
sudo groupadd nagcmd
```

```
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-88-250 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-88-250 ~]$ sudo usermod -a -G nagcmd nagios
[ec2-user@ip-172-31-88-250 ~]$ sudo usermod -a -G nagcmd apache
[ec2-user@ip-172-31-88-250 ~]$ █
```

7. Add Users to the Group

Commands -

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

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

```
[ec2-user@ip-172-31-80-22 ~]$ sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
```

8. Create a Directory for Nagios Downloads

Commands -

```
mkdir ~/downloads
```

```
cd ~/downloads
```

```
[ec2-user@ip-172-31-88-250 ~]$ sudo usermod -a -G nagcmd nagios
[ec2-user@ip-172-31-88-250 ~]$ mkdir ~/downloads
[ec2-user@ip-172-31-88-250 ~]$ cd ~/downloads
```

9. Download Nagios and Plugins Source Files

Commands -

```
wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz
```

```
wget https://nagios-plugins.org/download/nagios-plugins-2.3.3.tar.gz
```

```
[ec2-user@ip-172-31-88-250 downloads]$ wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz
--2024-09-30 18:02:53-- https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 2600:3c00::f03c:92ff:fe7:45ce
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 11333414 (11M) [application/x-gzip]
Saving to: 'nagios-4.4.6.tar.gz'

nagios-4.4.6.tar.gz          100%[=====]  10.81M  12.3MB/s   in 0.9s

2024-09-30 18:02:54 (12.3 MB/s) -- 'nagios-4.4.6.tar.gz' saved [11333414/11333414]

[ec2-user@ip-172-31-88-250 downloads]$
```

Anushka Shahane D15A 55

```
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: 2782610 (2.7M) [application/x-gzip]
Saving to: 'nagios-plugins-2.3.3.tar.gz.3'

nagios-plugins-2.3.3.tar.gz.3          100%[=====] 2.65M 7.09MB/s in 0.4s

2024-09-30 18:03:39 (7.09 MB/s) - 'nagios-plugins-2.3.3.tar.gz.3' saved [2782610/2782610]

[ec2-user@ip-172-31-88-250 downloads]$
```

```
curl -OJ -L http://prdownloads.sourceforge.net/nagios/nagios-4.0.8.tar.gz
Resolving prdownloads.sourceforge.net (prdownloads.sourceforge.net) ... 204.68.111.105
Connecting to prdownloads.sourceforge.net (prdownloads.sourceforge.net)|204.68.111.105|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: http://downloads.sourceforge.net/project/nagios/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz [following]
--2024-10-01 04:54:16-- http://downloads.sourceforge.net/project/nagios/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz
Resolving downloads.sourceforge.net (downloads.sourceforge.net)... 204.68.111.105
Reusing existing connection to prdownloads.sourceforge.net:80.
HTTP request sent, awaiting response... 302 Found
Location: http://netactuate.dl.sourceforge.net/project/nagios/nagios-4.x/nagios-4.0.8/nagios-4.0.8.tar.gz?viafs=1 [following]
--2024-10-01 04:54:16-- http://netactuate.dl.sourceforge.net/project/nagios/nagios-4.0.8/nagios-4.0.8.tar.gz?viafs=1
Resolving netactuate.dl.sourceforge.net (netactuate.dl.sourceforge.net)... 104.225.3.66
Connecting to netactuate.dl.sourceforge.net (netactuate.dl.sourceforge.net)|104.225.3.66|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1805059 (1.7M) [application/x-gzip]
Saving to: 'nagios-4.0.8.tar.gz'

nagios-4.0.8.tar.gz          100%[=====] 1.72M ---KB/s in 0.09s

2024-10-01 04:54:17 (18.6 MB/s) - 'nagios-4.0.8.tar.gz' saved [1805059/1805059]

[ec2-user@ip-172-31-80-174 downloads]$
```

```
i-Od08dddee33dde6db (nagios-host)
PuTTY v0.7.1.10
PuTTYPC 35.173.200.53 PrivatePe 172.31.80.174

nagios-4.0.8.tar.gz          100%[=====] 1.72M ---KB/s in 0.09s

2024-10-01 04:54:17 (18.6 MB/s) - 'nagios-4.0.8.tar.gz' saved [1805059/1805059]

[ec2-user@ip-172-31-80-174 downloads]$ wget http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz
--2024-10-01 04:55:13-- http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251
Connecting to nagios-plugins.org (nagios-plugins.org)|45.56.123.251|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2659772 (2.5M) [application/x-gzip]
Saving to: 'nagios-plugins-2.0.3.tar.gz'

nagios-plugins-2.0.3.tar.gz          100%[=====] 2.54M 7.82MB/s in 0.3s

2024-10-01 04:55:13 (7.82 MB/s) - 'nagios-plugins-2.0.3.tar.gz' saved [2659772/2659772]

[ec2-user@ip-172-31-80-174 downloads]$
```

10. Extract the Nagios Source File

Commands -

```
tar zxvf nagios-4.4.6.tar.gz
```

```
cd nagios-4.4.6
```

```
[ec2-user@ip-172-31-80-174 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/LICENS
nagios-4.0.8/MANAGER
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
nagios-4.0.8/base/Makefile.in
nagios-4.0.8/base/broker.c
nagios-4.0.8/base/checks.c
nagios-4.0.8/base/commands.c
nagios-4.0.8/base/events.c
nagios-4.0.8/base/flapping.c
nagios-4.0.8/base/logging.c
```

11. Run the Configuration Script

Commands -

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

```
nagios-4.0.8 nagios-4.0.8.tar.gz nagios-plugins-2.0.3.tar.gz
[ec2-user@ip-172-31-80-174 downloads]$ cd nagios-4.0.8
[ec2-user@ip-172-31-80-174 nagios-4.0.8]$ ./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
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 make sets $(MAKE)... yes
checking for strip... /usr/bin/strip
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
checking for ANSI C header files... yes
checking whether time.h and sys/time.h may both be included... yes
checking for sys/wait.h that is POSIX.1 compatible... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
checking for stdlib.h... yes
```

12. Compile the Source Code

Commands -

```
make all
```

```
[ec2-user@ip-172-31-80-22 nagios-4.4.6]$ make all
cd ./base && make
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.4.6/base'
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nagios.o nagios.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o broker.o broker.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o netmods.o netmods.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o ../common/shared.o ../common/shared.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o query-handler.o query-handler.c
gcc -Wall -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: `bs' directive argument is null [-Wformat-overflow=]
  253 |         log_debug_info(DEBUG_CHECKS, 1, "Found specialized worker(s) for '%s'", (slash && *slash != '/') ? slash : cmd_name);
           |
           ^~~~~~
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o checks.o checks.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o config.o config.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o commands.o commands.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o events.o events.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o flapping.o flapping.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o logging.o logging.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o macros-base.o ../common/macros.c
```

```
    /usr/bin/install -c -m 775 -o nagios -g nagios $FILE /usr/local/nagios/sbin/`  
done  
/usr/bin/install: cannot stat '*.cgi': No such file or directory  
make[2]: *** [Makefile:205: install-basic] Error 1  
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-4.0.8/cgi'  
make[1]: *** [Makefile:197: install] Error 2  
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.0.8/cgi'  
make: *** [Makefile:235: install] Error 2  
[ec2-user@ip-172-31-80-174 nagios-4.0.8]$ sudo make install-init  
/usr/bin/install -c -m 755 -d -o root -g root /etc/rc.d/init.d  
/usr/bin/install -c -m 755 -o root -g root daemon-init /etc/rc.d/init.d/nagios  
  
*** Init script installed ***  
[ec2-user@ip-172-31-80-174 nagios-4.0.8]$
```

```
*** Support Notes ****  
If you have questions about configuring or running Nagios,  
please make sure that you:  
- Look at the sample config files  
- Read the documentation on the Nagios Library at:  
  https://library.nagios.com  
before you post a question to one of the mailing lists.  
Also make sure to include pertinent information that could  
help others help you. This might include:  
- What version of Nagios you are using  
- What version of the plugins you are using  
- Relevant snippets from your config files  
- Relevant error messages from the Nagios log file  
For more information on obtaining support for Nagios, visit:  
  https://support.nagios.com  
*****  
Enjoy.  
[ec2-user@ip-172-31-88-250 nagios-4.4.6]$
```

13. Install Binaries, Init Script, and Sample Config Files

Commands -

./sudo make install

sudo make install-init

sudo make install-config

sudo make install-commandmode

```
*** External command directory configured ***  
[ec2-user@ip-172-31-80-174 nagios-4.0.8]$ sudo nano /usr/local/nagios/etc/objects/contacts.cfg  
[ec2-user@ip-172-31-80-174 nagios-4.0.8]$ sudo make install-webconf  
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf  
  
*** Nagios/Apache conf file installed ***  
  
[ec2-user@ip-172-31-80-174 nagios-4.0.8]$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin  
New password:  
Re-type new password:  
Adding password for user nagiosadmin  
[ec2-user@ip-172-31-80-174 nagios-4.0.8]$ sudo service httpd restart  
Redirecting to /bin/systemctl restart httpd.service  
[ec2-user@ip-172-31-80-174 nagios-4.0.8]$ cd ~/downloads  
[ec2-user@ip-172-31-80-174 downloads]$ tar zxvf nagios-plugins-2.0.3.tar.gz  
nagios-plugins-2.0.3/  
nagios-plugins-2.0.3/perlmods/  
nagios-plugins-2.0.3/perlmods/Config-Tiny-2.14.tar.gz  
nagios-plugins-2.0.3/perlmods/parent-0.226.tar.gz  
nagios-plugins-2.0.3/perlmods/Test-Simple-0.98.tar.gz  
nagios-plugins-2.0.3/perlmods/Makefile.in  
nagios-plugins-2.0.3/perlmods/version-0.9903.tar.gz  
nagios-plugins-2.0.3/perlmods/Makefile.am  
nagios-plugins-2.0.3/perlmods/Module-Runtime-0.013.tar.gz  
nagios-plugins-2.0.3/perlmods/Module-Metadata-1.000014.tar.gz  
nagios-plugins-2.0.3/perlmods/Params-Validate-1.08.tar.gz  
nagios-plugins-2.0.3/perlmods/Class-Accessor-0.34.tar.gz
```

14. Edit the Config File to Change the Email Address

Commands -

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

- Change the email address in the contacts.cfg file to your preferred email.

```
GNU nano 5.8                               /usr/local/nagios/etc/objects/contacts.cfg                         Modified | #####
#####
# 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             anushka3204@gmail.com  ; <>***** CHANGE THIS TO YOUR EMAIL ADDRESS *****

}

#####
# CONTACT GROUPS
#
^G Help      ^O Write Out   ^W Where Is   ^R Cut           ^T Execute   ^C Location   M-U Undo   M-A Set Mark   M-J To Bracket   M-Q Previous
^X Exit      ^R Read File   ^\ Replace    ^U Paste        ^J Justify   ^Y Go To Line M-E Redo   M-G Copy      ^Q Where Was    M-W Next
```

15. Configure the Web Interface

Commands -

```
sudo make install-webconf
```

```
[ec2-user@ip-172-31-88-250 nagios-4.4.6]$ sudo nano /usr/local/nagios/etc/objects/contacts.cfg
[ec2-user@ip-172-31-88-250 nagios-4.4.6]$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf
if [ 0 -eq 1 ]; then \
    ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi

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

[ec2-user@ip-172-31-88-250 nagios-4.4.6]$
```

16. Create a Nagios Admin Account

Commands -

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

- You will be prompted to enter and confirm the password for the nagiosadmin user

Anushka Shahane D15A 55

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

[ec2-user@ip-172-31-88-250 nagios-4.4.6]$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
[ec2-user@ip-172-31-88-250 nagios-4.4.6]$ █

nagios-plugins-2.3.3/pkg/fedora/requires
nagios-plugins-2.3.3/pkg/solaris/
nagios-plugins-2.3.3/pkg/solaris/preinstall
nagios-plugins-2.3.3/pkg/solaris/solpkg
nagios-plugins-2.3.3/pkg/solaris/pkginfo.in
nagios-plugins-2.3.3/pkg/solaris/pkginfo
nagios-plugins-2.3.3/pkg/redhat/
nagios-plugins-2.3.3/pkg/redhat/requirements
[ec2-user@ip-172-31-88-250 downloads]$ cd nagios-plugins-2.3.3
[ec2-user@ip-172-31-88-250 nagios-plugins-2.3.3]$ █
```

```
make install /usr/bin/install -c -o nagios -g nagios check_nc /usr/local/nagios/libexec/check_nc
make install-exec-hook
make[3]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.3.3/plugins'
cd /usr/local/nagios/libexec && \
for i in check_ftp check_imap check_ntp check_pop check_udp check_clamd ; do rm -f $i; ln -s check_tcp $i ; done ;\
if [ -x check_ldap ] ; then rm -f check_ldaps ; ln -s check_ldap check_ldaps ; fi
make[3]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.3.3/plugins'
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.3.3/plugins'
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.3.3/plugins'
Making install in plugins-scripts
make[1]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.3.3/plugins-scripts'
make[2]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.3.3/plugins-scripts'
test -z "/usr/local/nagios/libexec" || /usr/bin/mkdir -p "/usr/local/nagios/libexec"
/usr/bin/install -c -o nagios -g nagios check_breeze check_disk_smb check_flexlm check_ircd check_log check_rpc check_sensors check_wave check_ifs
status check_ifoperstatus check_mailq check_file_age check_ssl_validity utils.sh utils.pm '/usr/local/nagios/libexec'
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.3.3/plugins-scripts'
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.3.3/plugins-scripts'
```

17. Restart Apache

Commands -

```
sudo systemctl restart httpd
```

```
[ec2-user@ip-172-31-80-22 nagios-4.4.6]$ sudo systemctl restart httpd
[ec2-user@ip-172-31-80-22 nagios-4.4.6]$ █
```

18. Extract the Plugins Source File

Commands -

```
cd ~/downloads
```

```
tar zxvf nagios-plugins-2.3.3.tar.gz
```

Anushka Shahane D15A 55

```
cd nagios-plugins-2.3.3
```

```
[ec2-user@ip-172-31-80-22 nagios-4.4.6]$ sudo systemctl restart httpd
[ec2-user@ip-172-31-80-22 nagios-4.4.6]$ cd ~/downloads
tar zxfv nagios-plugins-2.3.3.tar.gz
cd nagios-plugins-2.3.3
nagios-plugins-2.3.3/
nagios-plugins-2.3.3/perlmods/
nagios-plugins-2.3.3/perlmods/Config-Tiny-2.14.tar.gz
nagios-plugins-2.3.3/perlmods/parent-0.226.tar.gz
nagios-plugins-2.3.3/perlmods/Test-Simple-0.98.tar.gz
nagios-plugins-2.3.3/perlmods/Makefile.in
nagios-plugins-2.3.3/perlmods/version-0.9903.tar.gz
nagios-plugins-2.3.3/perlmods/Makefile.am
nagios-plugins-2.3.3/perlmods/Module-Runtime-0.013.tar.gz
nagios-plugins-2.3.3/perlmods/Module-Metadata-1.000014.tar.gz
nagios-plugins-2.3.3/perlmods/Params-Validate-1.08.tar.gz
nagios-plugins-2.3.3/perlmods/Class-Accessor-0.34.tar.gz
nagios-plugins-2.3.3/perlmods/Try-Tiny-0.18.tar.gz
nagios-plugins-2.3.3/perlmods/Module-Implementation-0.07.tar.gz
nagios-plugins-2.3.3/perlmods/Makefile
nagios-plugins-2.3.3/perlmods/Perl-OSType-1.003.tar.gz
nagios-plugins-2.3.3/perlmods/install_order
nagios-plugins-2.3.3/perlmods/Nagios-Plugin-0.36.tar.gz
nagios-plugins-2.3.3/perlmods/Math-Calc-Units-1.07.tar.gz
nagios-plugins-2.3.3/perlmods/Module-Build-0.4007.tar.gz
nagios-plugins-2.3.3/ABOUT-NLS
nagios-plugins-2.3.3/configure.ac
nagios-plugins-2.3.3/Makefile.in
```

```
[ec2-user@ip-172-31-80-22 nagios-4.4.6]$ make
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.0.3/po'
make[1]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.0.3'
make[2]: Entering directory '/home/ec2-user/downloads/nagios-plugins-2.0.3'
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-2.0.3'
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-plugins-2.0.3'
[ec2-user@ip-172-31-80-174 nagios-plugins-2.0.3]$ sudo chkconfig --add nagios
[ec2-user@ip-172-31-80-174 nagios-plugins-2.0.3]$ sudo chkconfig nagios on
[ec2-user@ip-172-31-80-174 nagios-plugins-2.0.3]$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.0.8
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 08-12-2014
License: GPL

Website: http://www.nagios.org
Reading configuration data...
Error in configuration file '/usr/local/nagios/etc/nagios.cfg' - Line 452 (Check result path '/usr/local/nagios/var/spool/checkresults' is not a valid directory)
  Error processing main config file!

[ec2-user@ip-172-31-80-174 nagios-plugins-2.0.3]$
```

Anushka Shahane D15A 55

```
Read object config files okay...
Running pre-flight check on configuration data...

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

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
[ec2-user@ip-172-31-80-174 nagios-plugins-2.0.3]$
```

If Error

```
sudo mkdir -p /usr/local/nagios/var/spool/checkresults
sudo chown -R nagios:nagcmd /usr/local/nagios/var
```

```
Things look okay - No serious problems were detected during the pre-flight check
[ec2-user@ip-172-31-80-174 nagios-plugins-2.0.3]$ sudo service nagios start
Reloading systemd:                                [  OK  ]
Starting nagios (via systemctl):                  [  OK  ]
[ec2-user@ip-172-31-80-174 nagios-plugins-2.0.3]$
```

19. Compile and Install Plugins

Commands -

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

```
make
```

```
sudo make install
```

Anushka Shahane D15A 55

```
[ec2-user@ip-172-31-80-22 nagios-plugins-2.3.3]$ ./configure --with-nagios-user=nagios --with-nagios-group=nagios
make
sudo make install
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 to disable maintainer-specific portions of Makefiles... yes
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
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 for style of include used by make... GNU
checking dependency style of gcc... gcc3
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep...

```

20. Start Nagios

Commands -

```
sudo chkconfig --add nagios
sudo chkconfig nagios on
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
sudo systemctl start nagios
```

```
[ec2-user@ip-172-31-80-22 nagios-plugins-2.3.3]$ sudo chkconfig --add nagios
sudo chkconfig nagios on
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
sudo systemctl start nagios
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.

Nagios Core 4.4.6
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2020-04-28
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.
```

21. Check the Status of Nagios

Commands -

```
sudo systemctl status nagios
```

```
Things look okay - No serious problems were detected during the pre-flight check
[ec2-user@ip-172-31-80-174 nagios-plugins-2.0.3]$ sudo service nagios start
Reloading systemd: [ OK ]
Starting nagios (via systemctl): [ OK ]
[ec2-user@ip-172-31-80-174 nagios-plugins-2.0.3]$ sudo systemctl status nagios
● nagios.service - LSB: Starts and stops the Nagios monitoring server
  Loaded: loaded (/etc/rc.d/init.d/nagios; generated)
  Active: active (running) since Tue 2024-10-01 05:25:27 UTC; 38s ago
    Docs: man:systemd-sysv-generator(8)
  Process: 67937 ExecStart=/etc/rc.d/init.d/nagios start (code=exited, status=0/SUCCESS)
  Tasks: 6 (limit: 1112)
  Memory: 2.1M
  CPU: 49ms
  CGroup: /system.slice/nagios.service
          ├─67959 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
          ├─67961 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
          ├─67962 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
          ├─67963 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
          ├─67964 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
          └─67965 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

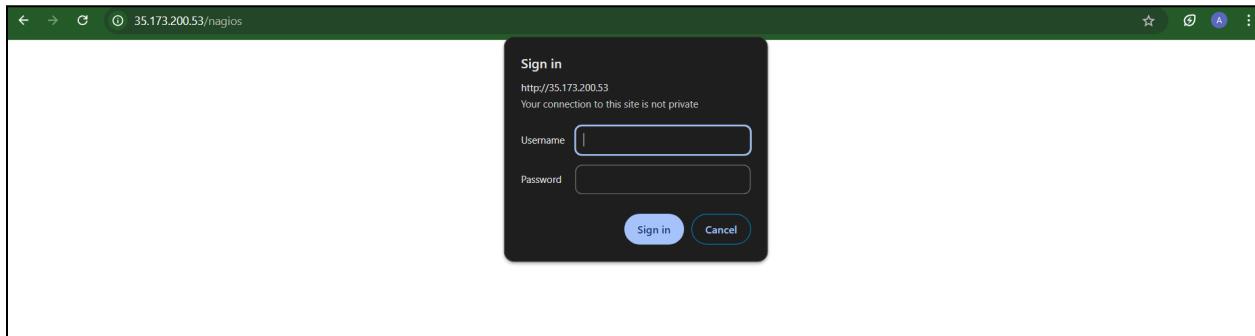
Oct 01 05:25:27 ip-172-31-80-174.ec2.internal nagios[67959]: nerd: Channel hostchecks registered successfully
Oct 01 05:25:27 ip-172-31-80-174.ec2.internal nagios[67959]: nerd: Channel servicechecks registered successfully
Oct 01 05:25:27 ip-172-31-80-174.ec2.internal nagios[67959]: nerd: Channel opathchecks registered successfully
Oct 01 05:25:27 ip-172-31-80-174.ec2.internal nagios[67959]: nerd: Fully initialized and ready to rock!
```

```
Starting nagios (via systemctl): [ OK ]
[ec2-user@ip-172-31-80-174 nagios-plugins-2.0.3]$ sudo systemctl status nagios
● nagios.service - LSB: Starts and stops the Nagios monitoring server
  Loaded: loaded (/etc/rc.d/init.d/nagios; generated)
  Active: active (running) since Tue 2024-10-01 05:25:27 UTC; 38s ago
    Docs: man:systemd-sysv-generator(8)
  Process: 67937 ExecStart=/etc/rc.d/init.d/nagios start (code=exited, status=0/SUCCESS)
  Tasks: 6 (limit: 1112)
  Memory: 2.1M
  CPU: 49ms
  CGroup: /system.slice/nagios.service
          ├─67959 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
          ├─67961 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
          ├─67962 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
          ├─67963 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
          ├─67964 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
          └─67965 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 01 05:25:27 ip-172-31-80-174.ec2.internal nagios[67959]: nerd: Channel hostchecks registered successfully
Oct 01 05:25:27 ip-172-31-80-174.ec2.internal nagios[67959]: nerd: Channel servicechecks registered successfully
Oct 01 05:25:27 ip-172-31-80-174.ec2.internal nagios[67959]: nerd: Channel opathchecks registered successfully
Oct 01 05:25:27 ip-172-31-80-174.ec2.internal nagios[67959]: nerd: Fully initialized and ready to rock!
Oct 01 05:25:27 ip-172-31-80-174.ec2.internal nagios[67959]: wproc: Successfully registered manager as @wproc with query handler
Oct 01 05:25:27 ip-172-31-80-174.ec2.internal nagios[67959]: wproc: Registry request: name=Core Worker 67964;pid=67964
Oct 01 05:25:27 ip-172-31-80-174.ec2.internal nagios[67959]: wproc: Registry request: name=Core Worker 67962;pid=67962
Oct 01 05:25:27 ip-172-31-80-174.ec2.internal nagios[67959]: wproc: Registry request: name=Core Worker 67963;pid=67963
Oct 01 05:25:27 ip-172-31-80-174.ec2.internal nagios[67959]: wproc: Registry request: name=Core Worker 67961;pid=67961
Oct 01 05:25:27 ip-172-31-80-174.ec2.internal nagios[67959]: Successfully launched command file worker with pid 67965
[ec2-user@ip-172-31-80-174 nagios-plugins-2.0.3]$
```

22. Access Nagios Web Interface

- Copy the Public IP address of your EC2 instance.
- Open your browser and navigate to http://<your_public_ip_address>/nagios.
- Enter the username nagiosadmin and the password you set in Step 16.



ADVANCE DEVOPS EXPERIMENT 10

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

Steps:

Prerequisites: AWS Free Tier, Nagios Server running on Amazon Linux Machine.

1. To Confirm that Nagios is running on the server side, run this sudo systemctl status nagios on the “NAGIOS HOST”

The screenshot shows a terminal window with the following text:

```
① Keyboard shortcut
To tab out of the terminal window and select the next button element, press the left and right Shift keys together.
Close permanently

Oct 01 08:19:24 ip-172-31-80-174.ec2.internal nagios[2192]: Auto-save of retention data completed successfully.
Oct 01 08:23:47 ip-172-31-80-174.ec2.internal nagios[2192]: SERVICE NOTIFICATION: nagiosadmin@localhost:Swap Usage;CRITICAL;notify-service-by-email;SWAP CRITICAL>
Oct 01 08:23:47 ip-172-31-80-174.ec2.internal nagios[2192]: wproc: NOTIFY job 32 from worker Core Worker 2198 is a non-check helper but exited with return code 1>
Oct 01 08:23:47 ip-172-31-80-174.ec2.internal nagios[2192]: wproc: host=localhost; service=Swap Usage; contact=nagiosadmin
Oct 01 08:23:47 ip-172-31-80-174.ec2.internal nagios[2192]: wproc: early_timeout=0; exited_ok=1; wait_status=32512; error_code=0;
Oct 01 08:23:47 ip-172-31-80-174.ec2.internal nagios[2192]: wproc: stderr line 01: /bin/sh: line 1: /bin/mail: No such file or directory
Oct 01 08:23:47 ip-172-31-80-174.ec2.internal nagios[2192]: wproc: stderr line 02: /usr/bin/printf: write error: Broken pipe
...skipping...
● nagios.service - Nagios Core 4.4.6
  Loaded: loaded ('/usr/lib/systemd/system/nagios.service'; enabled; preset: disabled)
  Active: active (running) since Tue 2024-10-01 07:19:25 UTC; 1h 33min ago
    Docs: https://www.nagios.org/documentation
   Tasks: 6 (limit: 1112)
  Memory: 4.6M
     CPU: 1.334s
    CGroup: /system.slice/nagios.service
           ├─2192 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           ├─2195 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
           ├─2196 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─2197 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─2198 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─2201 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 01 07:23:47 ip-172-31-80-174.ec2.internal nagios[2192]: wproc: early_timeout=0; exited_ok=1; wait_status=32512; error_code=0;
Oct 01 07:23:47 ip-172-31-80-174.ec2.internal nagios[2192]: wproc: stderr line 01: /bin/sh: line 1: /bin/mail: No such file or directory
Oct 01 07:36:17 ip-172-31-80-174.ec2.internal nagios[2192]: SERVICE ALERT: localhost:HTTP;WARNING;HARD;4:HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.00s
Oct 01 08:19:24 ip-172-31-80-174.ec2.internal nagios[2192]: Auto-save of retention data completed successfully.
Oct 01 08:23:47 ip-172-31-80-174.ec2.internal nagios[2192]: SERVICE NOTIFICATION: nagiosadmin@localhost:Swap Usage;CRITICAL;notify-service-by-email;SWAP CRITICAL>
Oct 01 08:23:47 ip-172-31-80-174.ec2.internal nagios[2192]: wproc: NOTIFY job 32 from worker Core Worker 2198 is a non-check helper but exited with return code 1>
```

You can proceed if you get this message.

2. Before we begin,

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

Anushka Shahane D15A 55

The screenshot shows the AWS Lambda console interface. A modal window is open for creating a new function named "linux-client". The "Code" tab is selected, displaying the Lambda@Edge code for the function. The "Summary" section indicates 1 instance, using the Canonical, Ubuntu, 24.04, amd64 AMI, t2.micro instance type, and a New security group. A tooltip for the Free tier is visible, stating it includes 750 hours of t2.micro (or t3.micro) in the Regions in which t2.micro is unavailable. The "Launch instance" button is highlighted.

The screenshot shows the AWS EC2 Instances page. It displays two running instances: "nagios-host" (Instance ID: i-0d08dddee33dde6db) and "linux-client" (Instance ID: i-0bdf16f8dffbb2d56f). Both instances are of the t2.micro type and are running. The "Actions" dropdown menu for the instances is open, showing options like "Stop", "Start", "Reboot", and "Delete".

The screenshot shows the AWS Security Groups page for the "nagios-host" instance. It lists several inbound rules:

- SSH (TCP, port 22, Custom, 0.0.0.0/0)
- All ICMP - IPv6 (IPv6 ICMP, All, Anywhere, ::/0)
- All ICMP - IPv4 (ICMP, All, Anywhere, 0.0.0.0/0)
- HTTP (TCP, port 80, Anywhere, 0.0.0.0/0)
- HTTPS (TCP, port 443, Anywhere, 0.0.0.0/0)
- All traffic (All, All, Anywhere, 0.0.0.0/0)
- Custom TCP (TCP, port 5666, Anywhere, 0.0.0.0/0)

An "Add rule" button is located at the bottom left of the list.

For now, leave this machine as is, and go back to your nagios HOST machine.

3. On the server, run this command

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

```
[ec2-user@ip-172-31-80-174 ~]$ ps -ef | grep nagios
lines 2-29
-bash: nagios: command not found
[ec2-user@ip-172-31-80-174 ~]$ ps -ef | grep nagios
nagios   2192      1  0 07:19 ?    00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios   2195      2192  0 07:19 ?    00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios   2196      2192  0 07:19 ?    00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios   2197      2192  0 07:19 ?    00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios   2198      2192  0 07:19 ?    00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios   2201      2192  0 07:19 ?    00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
[ec2-user 33385 33343 0 09:08 pts/1 00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-80-174 ~]$
```

4. Become a root user and create 2 folders

```
sudo su
```

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

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

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

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

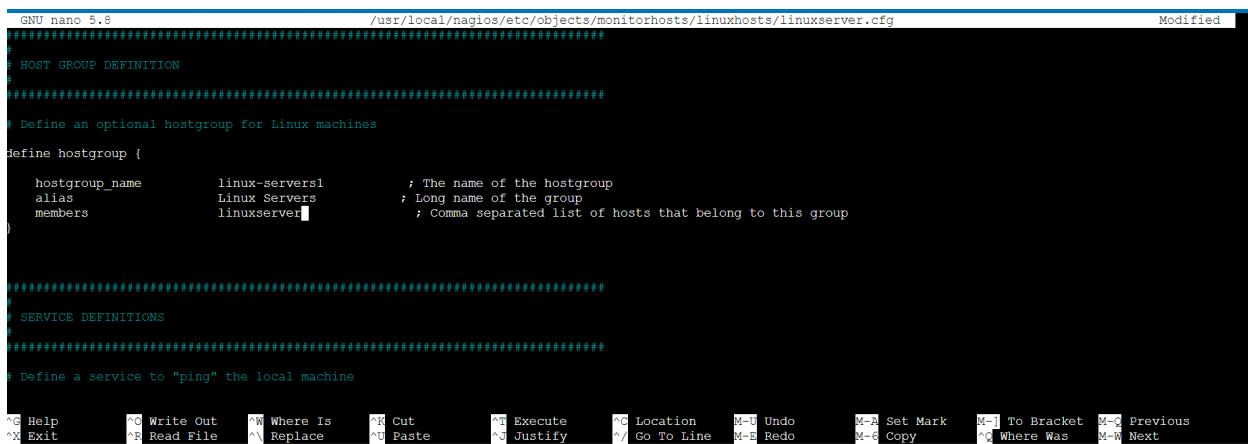
```
/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
```

```
ec2-user 33385 33343 0 09:08 pts/1 00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-80-174 ~]$ sudo su
[root@ip-172-31-80-174 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts
[root@ip-172-31-80-174 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-80-174 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg
cp: missing destination file operand after '/usr/local/nagios/etc/objects/localhost.cfg'
Try 'cp --help' for more information.
bash: /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg: No such file or directory
[root@ip-172-31-80-174 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
cp: missing destination file operand after '/usr/local/nagios/etc/objects/localhost.cfg'
Try 'cp --help' for more information.
[root@ip-172-31-80-174 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
bash: cp/usr/local/nagios/etc/objects/localhost.cfg/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg: No such file or directory
[root@ip-172-31-80-174 ec2-user]#
```

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

/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
Change the hostname to linuxserver (EVERYWHERE ON THE FILE)
Change address to the public IP address of your LINUX CLIENT.

Change hostgroup_name under hostgroup to linux-servers1



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

# HOST GROUP DEFINITION
#
# Define an optional hostgroup for Linux machines
define hostgroup {
    hostgroup_name      linux-servers1          ; The name of the hostgroup
    alias               Linux Servers           ; Long name of the group
    members              linuxserver           ; Comma separated list of hosts that belong to this group
}

# SERVICE DEFINITIONS
#
# Define a service to "ping" the local machine

^G Help      ^O Write Out   ^W Where Is   ^K Cut        ^T Execute   ^C Location   M-U Undo     M-A Set Mark M-[ To Bracket M-Q Previous
^X Exit      ^R Read File    ^V Replace    ^U Paste     ^J Justify   ^Y Go To Line M-B Redo     M-G Copy     M-Q Where Was M-W Next
```

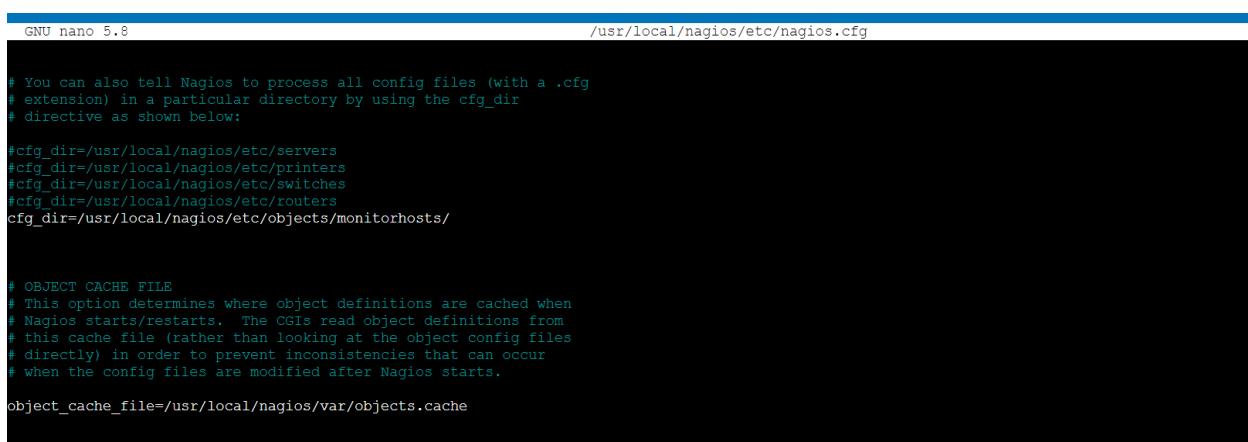
Everywhere else on the file, change the hostname to linuxserver instead of localhost.

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

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

##Add this line

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



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

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

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

# OBJECT CACHE FILE
# This option determines where object definitions are cached when
# Nagios starts/restarts. The CGIs read object definitions from
# this cache file (rather than looking at the object config files
# directly) in order to prevent inconsistencies that can occur
# when the config files are modified after Nagios starts.

object_cache_file=/usr/local/nagios/var/objects.cache
```

8. Verify the configuration files

You are good to go if there are no errors.

```
① Keyboard shortcut
To tab out of the terminal window and select the next button element, press the left and right Shift keys together.

Read object config files okay...

Running pre-flight check on configuration data...

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

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
[root@ip-172-31-80-174 ec2-user]#
```

9. Restart the nagios service

service nagios restart

Now it is time to switch to the client machine.

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

```
Things look okay - No serious problems were detected during the pre-flight check
[root@ip-172-31-80-174 ec2-user]# service nagios restart
Restarting nagios (via systemctl): [ OK ]
[root@ip-172-31-80-174 ec2-user]#
```

```
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-172-31-95-22:~$ 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 [380 kB]  
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]  
Get:8 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [83.1 kB]  
Get:9 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4576 B]  
Get:10 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [274 kB]  
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [116 kB]  
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [8632 B]  
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [10.4 kB]
```

i-0bdf16f8dff82d56f (linux-client)

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

sudo apt update -y

sudo apt install gcc -y

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

12. Open nrpe.cfg file to make changes.

sudo nano /etc/nagios/nrpe.cfg

Under allowed_hosts, add your nagios host IP address like so

13. Restart the NRPE server

sudo systemctl restart nagios-nrpe-server

```
GNU nano 7.2                               /etc/nagios/nrpe.cfg  
  
Note: The daemon only does rudimentary checking of the client's IP  
address. I would highly recommend adding entries in your /etc/hosts.allow  
file to allow only the specified host to connect to the port  
you are running this daemon on.  
  
NOTE: This option is ignored if NRPE is running under either inetd or xinetd  
  
allowed_hosts=127.0.0.1,34.227.30.243  
  
COMMAND ARGUMENT PROCESSING  
This option determines whether or not the NRPE daemon will allow clients  
to specify arguments to commands that are executed. This option only works  
if the daemon was configured with the --enable-command-args configure script  
option.  
  
*** ENABLING THIS OPTION IS A SECURITY RISK! ***  
Read the SECURITY file for information on some of the security implications  
of enabling this variable.  
  
Values: 0=do not allow arguments, 1=allow command arguments  
  
dont_blame_nrpe=0  
  
^H Help      ^W Write Out   ^A Where Is    ^X Cut      ^T Execute     ^C Location    ^U Undo      ^F Set Mark    ^G To Bracket  ^Q Previous  
^X Exit      ^R Read File   ^Y Replace   ^V Paste     ^J Justify    ^L Go to Line   ^M Redo      ^H Copy       ^I Where Was   ^K Next  
i-0bdf16f8dff82d56f (linux-client)  
PublicIPs: 3.86.147.85 PrivateIPs: 172.31.95.22
```

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

The screenshot shows the Nagios Core 4.4.6 dashboard. On the left, there's a sidebar with links for General, Home, Documentation, Current Status, Hosts, Services, Host Groups, Service Groups, Problems, Reports, and System. The 'Current Status' section is currently selected. At the top, there are two status summary boxes: 'Host Status Totals' and 'Service Status Totals'. Below them is a table titled 'Host Status Details For All Host Groups'.

Host	Status	Last Check	Duration	Status Information
linuxserver	UP	10-01-2024 09:54:32	0d 0h 15m 22s	PING OK - Packet loss = 0%, RTA = 1.36 ms
localhost	UP	10-01-2024 09:53:51	0d 4h 29m 28s	PING OK - Packet loss = 0%, RTA = 0.03 ms

Below the table, it says 'Results 1 - 2 of 2 Matching Hosts'. A 'Page Tour' link is located on the right side of the dashboard.

Advance Devops-11

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

Theory:

AWS Lambda

AWS Lambda is a serverless computing service provided by Amazon Web Services (AWS). Users of AWS Lambda create functions, self-contained applications written in one

of the supported languages and runtimes, and upload them to AWS Lambda, which executes those functions in an efficient and flexible manner. The Lambda functions can perform any kind of computing task, from serving web pages and processing streams of data to calling APIs and integrating with other AWS services.

The concept of “serverless” computing refers to not needing to maintain your own servers to run

these functions. AWS Lambda is a fully managed service that takes care of all the infrastructure

for you. And so “serverless” doesn’t mean that there are no servers involved: it just means that

the servers, the operating systems, the network layer and the rest of the infrastructure have

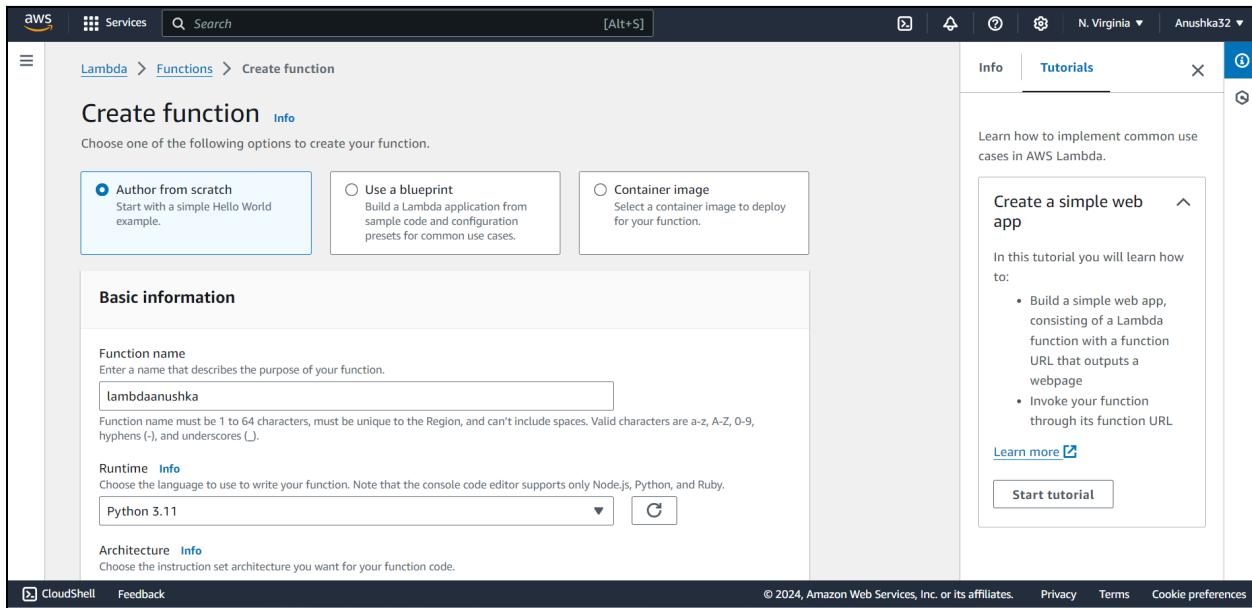
already been taken care of so that you can focus on writing application code.

Features of AWS Lambda

- AWS Lambda easily scales the infrastructure without any additional configuration. It reduces the operational work involved.
- It offers multiple options like AWS S3, CloudWatch, DynamoDB, API Gateway, Kinesis, CodeCommit, and many more to trigger an event.
- You don’t need to invest upfront. You pay only for the memory used by the lambda function and minimal cost on the number of requests hence cost-efficient.
- AWS Lambda is secure. It uses AWS IAM to define all the roles and security policies.
- It offers fault tolerance for both services running the code and the function. You do not have to worry about the application down.

Steps to create an AWS Lambda function

1. Open up the Lambda Console and click on the Create button.
Be mindful of where you create your functions since Lambda is region-dependent.



2. Choose to create a function from scratch or use a blueprint, i.e templates defined by AWS for you with all configuration presets required for the most common use cases.

Then, choose a runtime env for your function, under the dropdown, you can see all the options AWS supports, Python, Nodejs, .NET and Java being the most popular ones. After that, choose to create a new role with basic Lambda permissions if you don't have an existing one.

Anushka Shahane D15A 55

The screenshot shows the AWS Lambda function creation interface. On the left, there's a sidebar with navigation links like 'Lambda', 'Functions', and 'Logs'. The main area has tabs for 'Architecture' and 'Permissions'. Under 'Architecture', 'x86_64' is selected. Under 'Permissions', it says '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.' A note below says 'Change default execution role'. It lists three options: 'Create a new role with basic Lambda permissions' (selected), 'Use an existing role', and 'Create a new role from AWS policy templates'. A note below says 'Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in this role.' At the bottom, it says 'Lambda will create an execution role named lambdaanushka-role-rx3dhtr, with permission to upload logs to Amazon CloudWatch Logs.'

The screenshot shows the AWS Lambda function details page for 'lambdaanushka'. At the top, a green banner says 'Successfully created the function lambdaanushka. You can now change its code and configuration. To invoke your function with a test event, choose "Test".' Below this, the function name 'lambdaanushka' is displayed. The 'Function overview' tab is selected. It shows a diagram of the function, which includes a Lambda icon labeled 'lambdaanushka' and a 'Layers' section with '(0)'. There are buttons for '+ Add trigger' and '+ Add destination'. On the right, there are sections for 'Description' (empty), 'Last modified' (22 seconds ago), 'Function ARN' (arn:aws:lambda:us-east-1:010928179930:function:lambdaanushka), and 'Function URL' (Info). A sidebar on the right contains a 'Tutorials' tab with a 'Create a simple web app' section, which says 'In this tutorial you will learn how to:' followed by a bulleted list: 'Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage', 'Invoke your function through its function URL'. It also has 'Learn more' and 'Start tutorial' buttons.

The screenshot shows the AWS Lambda Functions list page. At the top, there is a search bar with the placeholder "Filter by tags and attributes or search by keyword". Below the search bar is a table header with columns: Function name, Description, Package type, Runtime, and Last modified. A "Create function" button is located in the top right corner of the table area. The table contains one row for the function "lambdaanushka", which has the description "Anushka-exp11", package type "Zip", runtime "Python 3.11", and was last modified "2 minutes ago".

Click on the Create button.

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

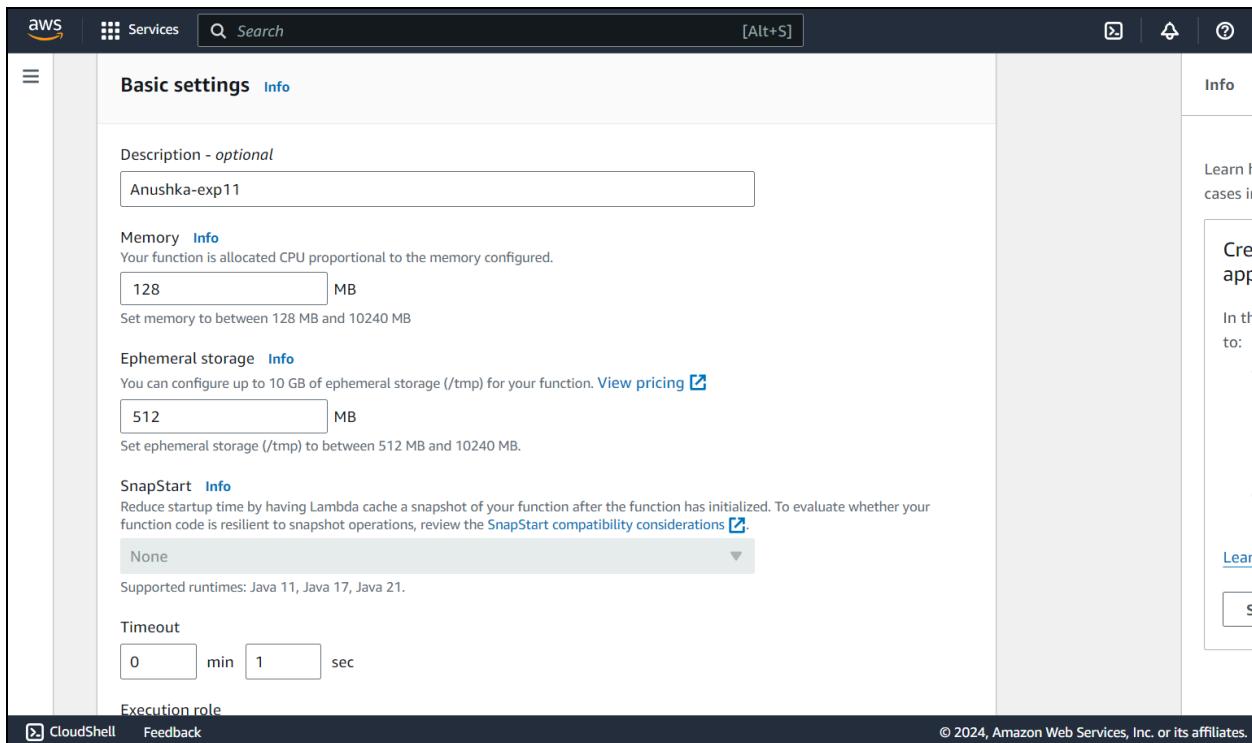
The screenshot shows the AWS Lambda function editor for the function "lambdaanushka". The code editor displays the following Python code:

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

A green success message at the top of the editor states: "Successfully created the function lambdaanushka. You can now change its code and configuration. To invoke your function with a test event, choose 'Test'." On the right side of the editor, there is a "Tutorials" panel titled "Create a simple web app". It includes a brief description and a list of objectives:

- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

There are "Learn more" and "Start tutorial" buttons at the bottom of the panel.



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

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 'Code source' tab of the Lambda function editor. The code editor displays a Python file named 'lambda_function.py' with the following content:

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

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

Configure test event

A test event is a JSON object that mocks the structure of requests emitted by AWS services to invoke a Lambda function. Use it to see the function's invocation result.

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

Cancel **Invoke** **Save**

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

The screenshot shows the AWS Lambda function editor interface. At the top, there are tabs for File, Edit, Find, View, Go, Tools, Window, Test, Deploy, and a status message "Changes not deployed". Below the tabs is a search bar labeled "Go to Anything (Ctrl-P)". The main area displays a file tree under "lambdaanushka /" with "lambda_function.py" selected. The code editor shows the following Python code:

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

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

The screenshot shows the AWS Lambda function details page for "lambdaanushka". The top navigation bar includes tabs for Code, Test, Monitor, Configuration, Aliases, and Versions. The "Test" tab is active. The main content area is titled "Code source" and shows the same Python code as the previous screenshot. Below the code editor, there is a "Test Event Name" input field containing "myevent". The "Execution result" section shows a green status bar with "Status: Succeeded", "Max memory used: 33 MB", and "Time: 2.37 ms". The "Response" field displays the JSON output: {"statusCode": 200, "body": "Hello! This is Anushka\""}". The "Function Logs" section contains log entries for the execution, including RequestId, Version, Duration, and Billed Duration. The "Request ID" field shows the value "480ce1b2-2f2c-48f7-8cf1-eb5a621f08ed".

Conclusion:

AWS Lambda is a serverless computing service that allows you to run code without managing servers, making it highly scalable, cost-effective, and easy to use. It automatically manages the compute resources, executes your code in response to specific events such as API calls, file uploads, or database updates, and scales based on the demand.

Advance Devops-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:

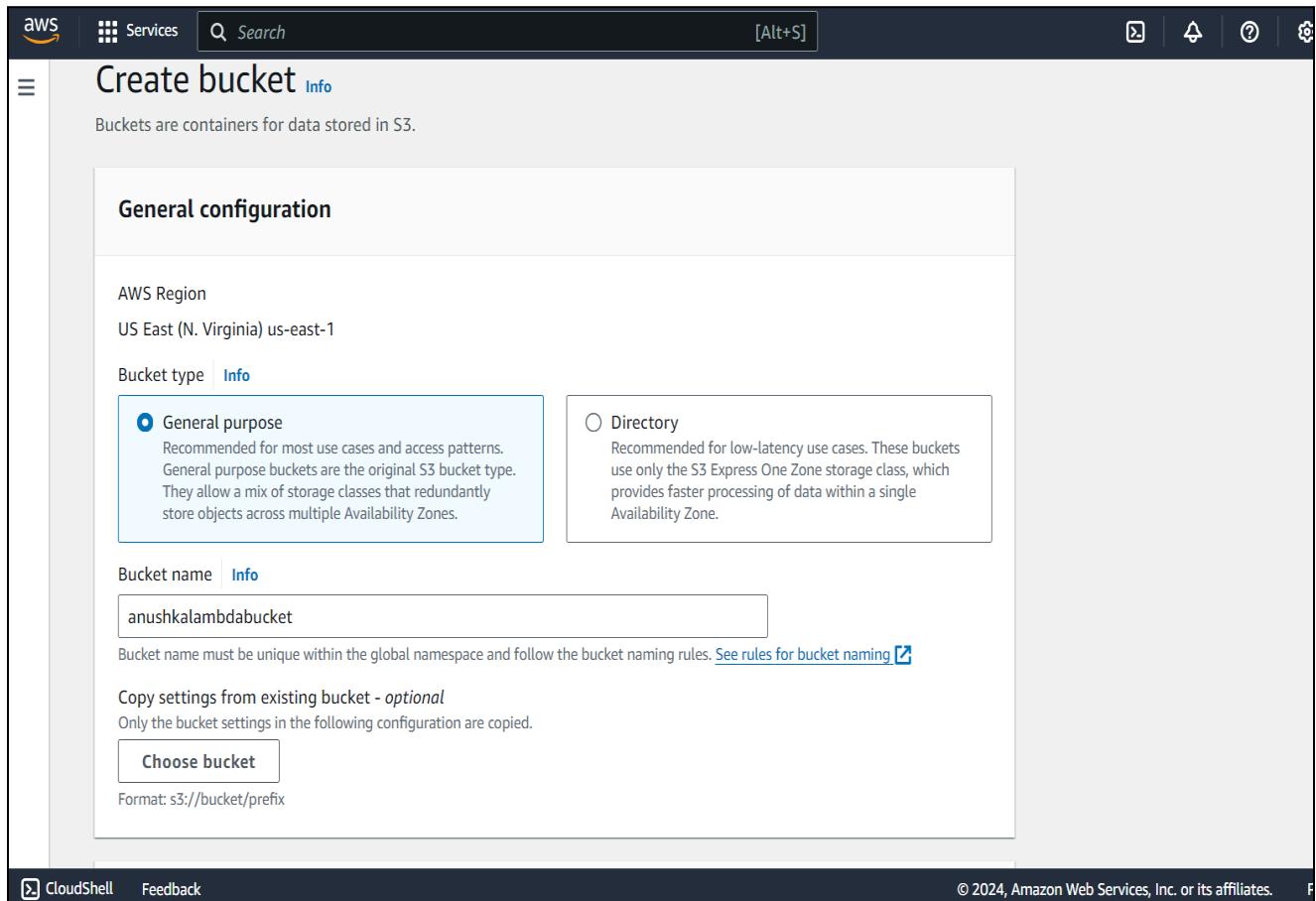
Link the S3 bucket to the Lambda function by setting up a trigger. Specify that the function should be triggered when an object is created in the bucket (e.g., when an image is uploaded).

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.

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.

The screenshot shows the AWS S3 Buckets page. At the top, a green banner displays a success message: "Successfully created bucket 'anushkalambdabucket'". Below the banner, there's an "Account snapshot - updated every 24 hours" section with a "View Storage Lens dashboard" button. The main area shows two tabs: "General purpose buckets" (selected) and "Directory buckets". Under "General purpose buckets", there's a table with one item:

Name	AWS Region	IAM Access Analyzer	Creation date
anushkalambdabucket	US East (N. Virginia) us-east-1	View analyzer for us-east-1	October 7, 2024, 00:40:49 (UTC+05:30)

Actions for the bucket include "Copy ARN", "Empty", "Delete", and "Create bucket".

The screenshot shows the AWS Lambda "Create function" wizard. The top navigation bar includes the AWS logo, Services, Search, and a [Alt+S] key shortcut. The breadcrumb path is Lambda > Functions > Create function.

Create function Info

Choose one of the following options to create your function.

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

Basic information

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

Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

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

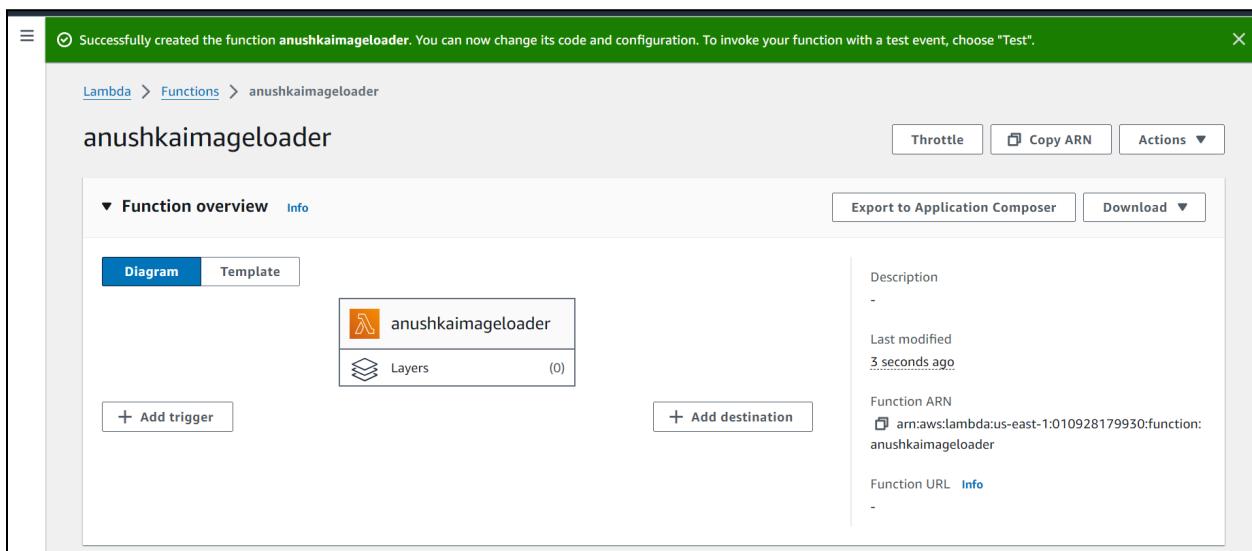
Architecture Info
Choose the instruction set architecture you want for your function code.
 x86_64

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates.

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.

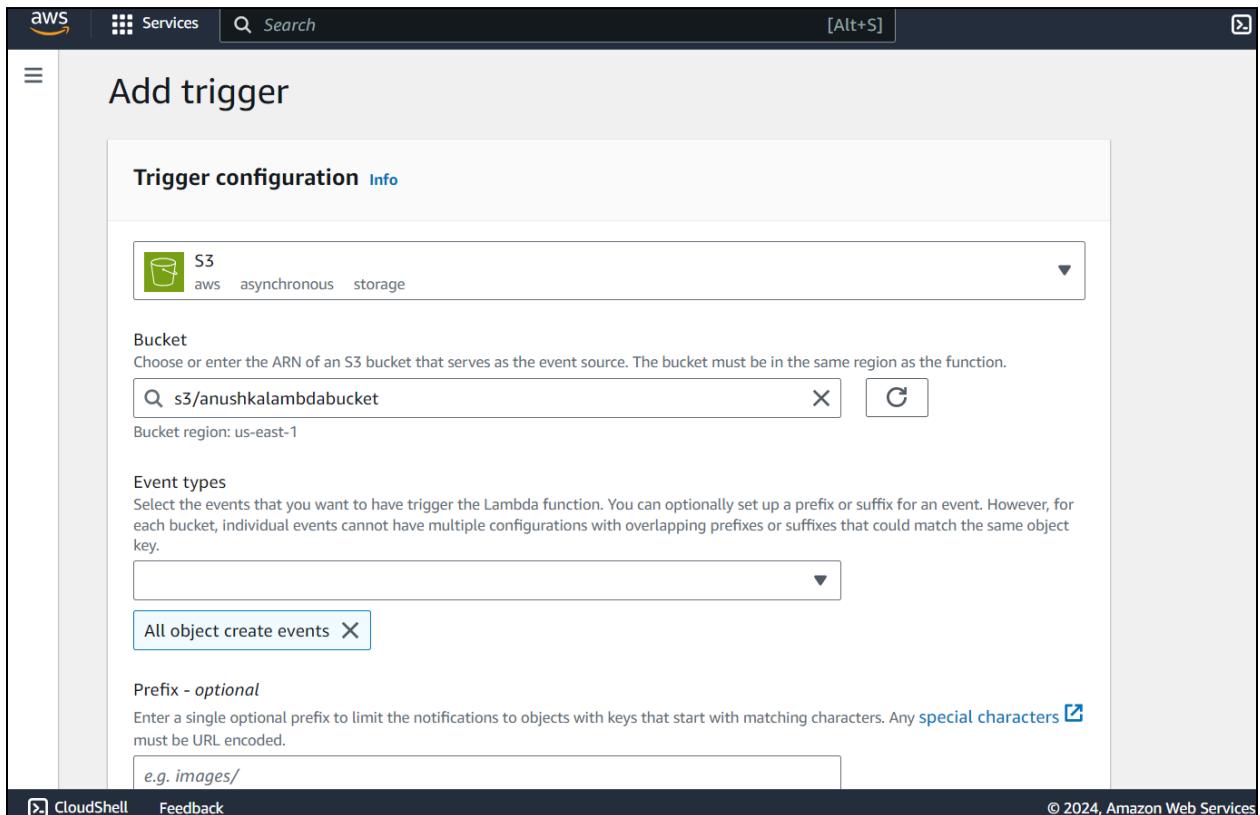


The screenshot shows the AWS Lambda Code Source editor. The title bar includes "File", "Edit", "Find", "View", "Go", "Tools", "Window", "Test", "Deploy" (which is currently selected), and "Changes not deployed". The search bar says "Go to Anything (Ctrl-P)". The sidebar shows an "Environment" section and a file tree with "anushkaimageloader" expanded, showing "lambda_function.py". The main area displays the Python code for the lambda function:

```
1 import json
2
3 def lambda_handler(event, context):
4     # Extract bucket name and object key from the event
5     bucket_name = event['Records'][0]['s3']['bucket']['name']
6     object_key = event['Records'][0]['s3']['object']['key']
7
8     # Log a message
9     print(f"An Image has been added to the bucket: {bucket_name}, with object key: {object_key}")
10
11    return {
12        'statusCode': 200,
13        'body': json.dumps("Log entry created successfully")
14    }
15
```

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).



Anushka Shahane D15A 55

Lambda > Functions > anushkaimageloader

anushkaimageloader

The trigger anushkalambdabucket was successfully added to function anushkaimageloader. The function is now receiving events from the trigger.

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

Diagram [Template](#)

anushkaimageloader

S3

[+ Add destination](#)

[+ Add trigger](#)

Description
-

Last modified
5 minutes ago

Function ARN
[arn:aws:lambda:us-east-1:010928179930:function:anushkaimageloader](#)

Function URL [Info](#)
-

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

General configuration

Triggers

Permissions

Destinations

Function URL

Environment variables

Tags

VPC

RDS databases

Monitoring and operations tools

Concurrency and recursion detection

Triggers (1) [Info](#)

Trigger

S3: anushkalambdabucket
arn:aws:s3:::anushkalambdabucket

[► Details](#)

Anushka Shahane D15A 55

The screenshot shows the AWS Lambda function policy configuration page. At the top, there are two managed policies listed:

- Managed policy AWSLambdaBasicExecutionRole-9ecf207b-9f4b-4ac0-b11b-8f0a2b5eb705, statement 0
- Managed policy AWSLambdaBasicExecutionRole-9ecf207b-9f4b-4ac0-b11b-8f0a2b5eb705, statement 1

Below this, the "Resource-based policy statements" section shows one statement:

Statement ID	Principal	PrincipalOrgID	Conditions	Action
lambda-de656519-7...	s3.amazonaws.com	-	StringEquals, ArnLike	lambda:InvokeFunction

At the bottom, there is an "Auditing and compliance" section with a note about AWS CloudTrail.

The screenshot shows the Amazon S3 bucket management page for the bucket "anushkalamdbabucket". The left sidebar includes options like Buckets, Access Grants, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, IAM Access Analyzer for S3, and Storage Lens.

The main content area shows the bucket details and an "Objects (0)" section. The "Upload" button is highlighted in orange. A note states: "Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)".

The "Objects (0)" table has columns: Name, Type, Last modified, Size, and Storage class. It displays the message: "No objects" and "You don't have any objects in this bucket."

The screenshot shows the AWS S3 'Upload' interface. At the top, there's a search bar and a 'Search' button. Below it, a large central area is titled '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](#)'.

In the center, there's a dashed blue box with the text '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 (1 Total, 321.6 KB)'. It contains one item: 'desk.jpg' (image/jpeg). There are 'Remove', 'Add files', and 'Add folder' buttons above the table. A search bar labeled 'Find by name' is also present.

Below the table is a section titled 'Destination' with an 'Info' link. It shows the destination as 's3://anushkalambdabucket'. At the bottom, there are links for 'CloudShell' and 'Feedback', and a copyright notice: '© 2024, Amazon Web Services, Inc. or its affiliates.'

The screenshot shows the AWS S3 'Upload' results interface. At the top, a green banner displays a success message: 'Upload succeeded' with a circular icon and 'View details below.' Below this is a 'Summary' section with a table:

Destination	Succeeded	Failed
s3://anushkalambdabucket	1 file, 321.6 KB (100.00%)	0 files, 0 B (0%)

Below the summary is a navigation bar with 'Files and folders' and 'Configuration' tabs, where 'Files and folders' is selected. This leads to a table titled 'Files and folders (1 Total, 321.6 KB)'. It lists the single file 'desk.jpg' with its details: Name (desk.jpg), Folder (-), Type (image/jpeg), Size (321.6 KB), Status (Succeeded), and Error (-).

The screenshot shows the AWS CloudWatch Log Groups interface. The URL is [CloudWatch > Log groups > /aws/lambda/anushkaimageloader](#). The main title is **/aws/lambda/anushkaimageloader**. Below it, there's a section titled **Log group details** with the following information:

Log class	Info	Stored bytes	KMS key ID
Standard	-	-	-
ARN	arn:aws:logs:us-east-1:010928179930:log-group:/aws/lambda/anushkaimageloader:*	Metric filters	Anomaly detection
		0	Configure
Creation time	Subscription filters	Data protection	Sensitive data count
2 minutes ago	0	-	-
Retention	Contributor Insights rules		
Never expire	-		

Below the details, there are tabs for **Log streams**, **Tags**, **Anomaly detection**, **Metric filters**, **Subscription filters**, **Contributor Insights**, and **Data protection**.

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.

The screenshot shows the AWS CloudWatch Log Events interface. The URL is [CloudWatch > Log groups > /aws/lambda/anushkaimageloader > 2024/10/06/\[LATEST\]7afb4f306f8a47c1bbe0fcac08dca6cf](#). The title is **Log events**. The interface includes a filter bar with a search input, time range (Clear, 1m, 30m, 1h, 12h, Custom, UTC timezone), and display settings. The log events table has columns for **Timestamp** and **Message**.

Timestamp	Message
2024-10-06T19:25:48.670Z	INIT_START Runtime Version: python:3.11.v44 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:b1c790bce6ec3c3a14a715f557a25d2daffc...
2024-10-06T19:25:48.746Z	START RequestId: b00e2348-6c9a-4d53-8479-3653bdd4c9af Version: \$LATEST
2024-10-06T19:25:48.746Z	An Image has been added to the bucket: anushkalambdabucket, with object key: desk.jpg
2024-10-06T19:25:48.748Z	END RequestId: b00e2348-6c9a-4d53-8479-3653bdd4c9af
2024-10-06T19:25:48.748Z	REPORT RequestId: b00e2348-6c9a-4d53-8479-3653bdd4c9af Duration: 2.16 ms Billed Duration: 3 ms Memory Size: 128 MB Max Memory Used: 33 ...

No newer events at this moment. [Auto retry paused.](#) [Resume](#)

This screenshot shows a portion of the log events table from the previous screenshot. It highlights the event where an image was uploaded:

2024-10-06T19:25:48.746Z	An Image has been added to the bucket: anushkalambdabucket, with object key: desk.jpg
2024-10-06T19:25:48.748Z	END RequestId: b00e2348-6c9a-4d53-8479-3653bdd4c9af

Conclusion:

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

ASSIGNMENT NO. 1

Advance Devops

Q1. Use S3 bucket and host video streaming

Ans: To host video streaming using Amazon S3 bucket you can create a scalable and cost-effective solution by utilizing S3 for video storage and AWS CloudFront as a Content Delivery Network (CDN) to efficiently stream video to users.

Steps to Host Video Streaming with S3 bucket:

(1) Create an S3 bucket

Store video files (MP4, WebM etc) in an S3 bucket. Set appropriate permissions to allow access to videos via CloudFront.

(2) Upload Videos:

Upload your videos to the S3 bucket. Ensure the files are publicly accessible if necessary, or restrict access based on user roles.

(3) Enable S3 Static Website Hosting (Optional):

If you want users to access a website along with streaming content, enable static website hosting for your S3 bucket.

(4) Configure AWS CloudFront:

Create a CloudFront distribution with the S3 bucket as the origin. CloudFront caches your video content at edge locations worldwide.

reducing latency for users.

(5) Setup permissions:

Set bucket policies or IAM roles to secure your S3 bucket from unauthorized access, ensuring only CloudFront can fetch content.

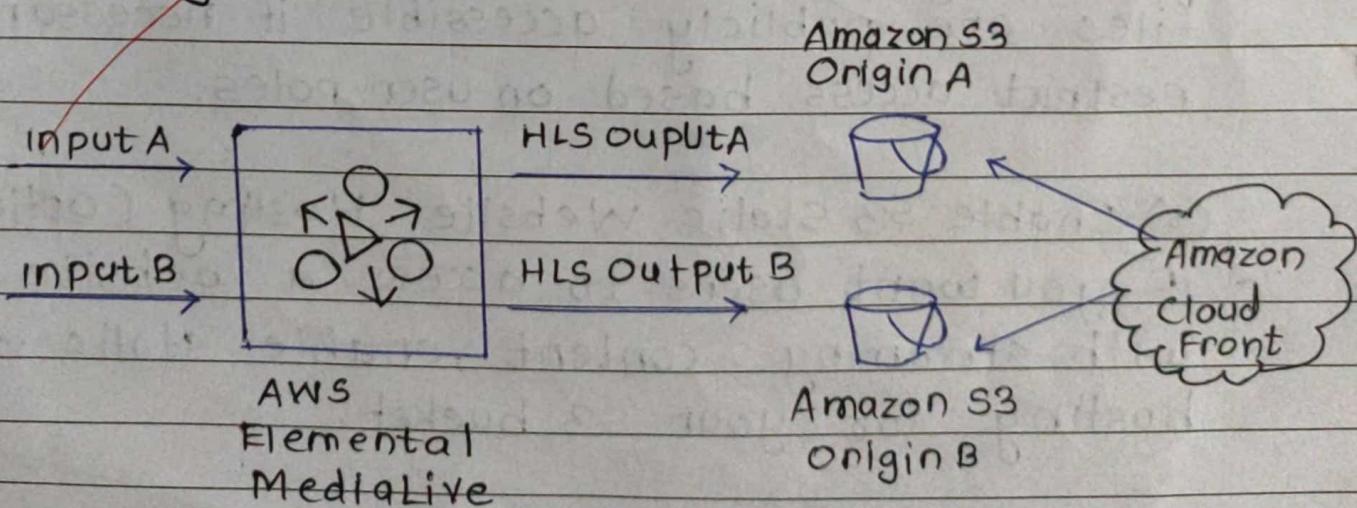
(6) Streaming Optimization:

You can transcode videos using AWS Elastic Transcoder to support multiple resolutions & formats.

Use chunk-based streaming to provide smooth streaming to users with varying bandwidth.

(7) Custom Domain and SSL (Optional)

Integrate a custom domain for your CloudFront distribution and use an SSL certificate for secure streaming.



Q(2) Discuss BMW and Hotstar case studies using AWS.

SOLN BMW Case study using AWS

Ans: Background: BMW is a leading automotive manufacturer known for its luxury vehicles, while Hotstar (now known as Disney+ Hotstar) is a popular streaming platform in India, offering a variety of content including movies, TV shows and live sports. Both companies have utilized AWS to drive innovation, improve customer experiences & optimize their operations.

BMW's use of AWS

(1) Connected Vehicles and Data Analytics

BMW has been at the forefront of integrating technology into their vehicles. By leveraging AWS, they can collect and analyze vast amounts of data from their connected cars. This includes:

- ① Vehicle performance data

② Driver Behaviour

Benefits:

- ① Predictive maintenance: AWS enables BMW to use machine learning algorithms to predict when a vehicle needs servicing, reducing downtime and improving customer satisfaction.

(2) Scalability and Cost Management

BMW utilizes AWS scalable infrastructure to

handle varying workloads, especially during product launches or events.

Benefits:

- ① Cost efficiency: BMW can scale resources up or down based on demand, ensuring they only pay for what they use.
- ② Global Reach: AWS's global infrastructure allows BMW to deploy applications closer to their customers, reducing latency and improving service delivery.

Hotstar's Use of AWS

① Content Delivery and Streaming Services

Hotstar relies heavily on AWS to manage its massive content library and delivery of high quality streaming experiences to millions of users.

② AWS CloudFront: Hotstar uses AWS's Content Delivery Network (CDN) to deliver content quickly and efficiently ensuring minimal buffering and downtime during peak times such as major sports events.

Benefits:

- Scalability: During events like the IPL user traffic can spike dramatically. AWS allows Hotstar to scale resources dynamically

to handle these spikes without compromising performance.

- Global Content Reach: AWS enables Hotstar to distribute content across multiple regions, ensuring that users worldwide can access their services seamlessly.

(2) Data Analytics for User Engagement: Hotstar leverages AWS data analytics tools to gather insights about user behaviour, content performances and viewing patterns.

- AWS Redshift and Athena: These services help Hotstar analyze large datasets to improve content recommendations and user engagement strategies.

Benefits:

- ◎ Personalized Content Recommendations
- ◎ Targeted Advertising.

challenges and solutions:

while both BMW and Hotstar have seen significant benefits from using AWS they also face challenges:

① Data Security and Compliance: Protecting user data

② Cost management: As usage scales, manage

costs... can become challenging

Conclusion:

The integration of AWS into BMW and Hotstar operations demonstrates how cloud computing can drive innovation, improve customer experiences and enhance operational efficiency.

Q(3) why Kubernetes and advantages and disadvantages of Kubernetes. Explain how adidas uses Kubernetes.

Soln: Kubernetes is an open-source container orchestration platform designed to automate deploying, scaling, and managing containerized applications. It abstracts the underlying infrastructure, allowing developers to focus on writing code rather than managing servers.

Advantages of Kubernetes:

- (1) Scalability: Kubernetes can automatically scale applications up or down based on demand, ensuring optimal resource use.
- (2) High Availability: It provides mechanisms for self-handling, meaning if a container fails, Kubernetes can automatically restart or replace it.

(3) Load Balancing : Kubernetes can distribute network traffic evenly across containers, improving application performance and reliability.

(4) Declarative Configuration: Users can define the desired state of applications using YAML files making deployments reproducible and version-controlled.

(5) Portability : Kubernetes can run on any cloud or on-premises infrastructure making it easier to move applications between environments.

Disadvantages of Kubernetes:

① Complexity: The learning curve can be steep due to its many components and abstractions, requiring more expertise to manage effectively.

② Overhead: Running Kubernetes introduces additional resource overhead, which might not be justified for smaller applications

③ Configuration Management: Managing configuration can become complicated, in larger environments.

④ Debugging Challenges: Debugging issues in a distributed system can be more difficult

than in traditional, monolithic architecture.

How Adidas uses Kubernetes:

Adidas has adopted Kubernetes to enhance its development and operational efficiencies. Here's how they utilize it:

(a) Microservices Architecture: Adidas leverages Kubernetes to manage its microservice, enabling them to develop, deploy and scale services independently.

(b) Devops Practices: Kubernetes supports Adidas devops practices allowing for continuous integration and continuous delivery (CI/CD) pipelines. This facilitates frequent updates and rapid iteration on their applications.

(c) Resource Optimization: By using Kubernetes, Adidas can efficiently manage cloud resources, reducing costs while ensuring that applications are responsive to user demand, especially during peak shopping seasons.

(d) Enhanced Collaboration: Kubernetes fosters collaboration between development and operations teams by standardizing deployment processes and improving visibility into application performances.

e) Global scaling: with a worldwide presence Adidas uses Kubernetes to deploy applications across multiple regions seamlessly ensuring consistent performances for customers everywhere.

Q4) What are Nagios and explain how Nagios are used in e-services

Soln: Nagios is an open-source monitoring system that enables organisations to monitor their IT infrastructure, including servers, networks and applications. It provides real-time monitoring and alerting capabilities to ensure that critical systems are running smoothly and any issues are promptly addressed.

Key features of Nagios:

- ① Monitoring: Tracks network services (HTTP, SMTP, POP3, FTP etc), server resources (CPU, memory, disk usage) and application performance.
- ② Alerts: Sends notifications (via email, SMS, or other methods) when thresholds are breached or services are down.
- ③ Reporting: Provides historical reports for performance analysis and capacity planning.
- ④ Customization: Offers plugins to expand functionality and supports integration with third party

systems for automation.

⑤ Scalability: Capable of monitoring large-scale environments.

Use of Nagios in E services:

In E services such as web-based systems or applications, Nagios is used to ensure the availability and performance of critical services. It can monitor the health of web servers, databases, API's and other infrastructure components detecting issues like high response times, system downtimes, or overloads.

By doing this, Nagios helps in:

① Minimizing Downtime: Real-time monitoring helps detect and rectify service failures promptly ensuring continuous service availability.

② Proactive Issue Resolution: It alerts administrators before performance degradation affects the user experience enabling them to act preemptively.

③ Service-Level Agreement (SLA) Compliance: By tracking uptime and service quality, Nagios assists businesses in meeting their service-level commitments.

④ Security Monitoring: Detects abnormal behaviour or unauthorized access attempts by monitoring logs and user activities.

Overall, Nagios is widely used in E services to ensure seamless operation and a consistent user experience.

(or)
55%

ADVANCE DEVOPS ASSIGNMENT 2

Q1)

Soln:

Create a REST API with the Serverless Framework
The Serverless Framework helps you deploy applications to cloud providers like AWS using a simplified configuration.

(1) Install Serverless framework: Ensure you have Node.js installed, then install the Serverless Framework using npm
`npm install -g serverless`

(2) Set up AWS credentials: Serverless uses AWS Lambda and API Gateway so configure your AWS Console
`aws configure`

Add your AWS Access Key, Secret Key, Region etc.

(3) Create a New Service: Create a new project using a Node.js template.
`serverless create --template aws-nodejs --path rest-api`

This creates a basic Serverless service with a structure including `serverless.yml` and `handler.js` file for code.

(4) Define the REST API in serverless.yml:
Edit the serverless.yml to define the REST API endpoints.
For eg : service: rest-api-service
provider:
 name: aws
 runtime: nodejs14.x
functions:
 hello:
 handler: handler.hello
 events:
 - http:
 path: hello
 method: get

The handler.js file would contain the logic

(5) Deploy the Service: Deploy the API to Aws Lambda and API gateway using: serverless deploy

This deploys infrastructure and you'll get a URL to access API

(6) Testing: Use the URL provided to access your REST API . You can test it with tools like Postman or simply via your browser.

x(2)

case study for Sonarqube.

SOLN:

SonarQube helps to automatically review your code for bugs, vulnerabilities, and code smells. The steps below cover Java, Python and Node.js analysis.

(1) Create a SonarQube Profile:

Go to SonarQube Cloud and create an account. You can link this account to your Github profile to analyze code directly from repositories.

Create a project in SonarCloud and connect it with your Github repository.

(2) Analyze Code on SonarCloud:

For your Github repository, configure it with SonarCloud. This can be done using CI pipelines (eg. Github Actions) or manually.

Example of Github Actions YAML for SonarCloud

```
name: SonarCloud
```

```
on:
```

```
  push:
```

```
    branches:
```

```
      - main
```

jobs:

sonarcloud:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v2

- name: SonarCloud Scan

- uses: sonarsource/sonarcloud-github-action@v1.4

with:

args: >

- Dsonar.projectKey=my-project-key

- Dsonar.organization=my-org

- Dsonar.host.url=https://sonarcloud.io

③ SonarLint Setup in Java IDE:

Install SonarLint in IntelliJ IDEA or

Eclipse from the plugin market place.

Configure it to link with your Sonar Cloud account for continuous quality checks.

Once installed SonarLint will analyze your Java code locally for issues.

(4) Python Project Analysis:

① Create a sonar-project-properties file in the root of your python project.

sonar.projectKey = my-python-project
sonar.organization = my-org
sonar.sources = .

Run the analysis using SonarQube scanner:
sonar-scanner

(5) Nodejs Project Analysis:

For a Node.js project the steps are similar to Python. Configure a sonar-project-properties file and scan the project using sonar-scanner.

Example sonar-project-properties

sonar.projectKey = my-nodejs-project
sonar.sources = .

sonar.exclusions = node_modules/**, **/*.test.js

Run sonar-scanner to analyze your Node.js project.

Analyze Results :

Just like with Python, the results of the analysis will be uploaded to Sonar Cloud. The report can be accessed from the dashboard which will highlight issues such as missing semicolons, unused variables and more.

Key features :

- SonarCloud allows you to integrate with Github easily and provides a cloud based dashboard for viewing the quality of your projects.
- SonarLint helps developers fix issues in real time as they write code, making it easier to catch issues before they are committed.
- For Python and Node.js, the sonar-project.properties file is essential for configuring the analysis and the sonar-scanner tool helps run the analysis locally.

Q3)

At a large organization, your centralized operations team get many repetitive infrastructure requests. You can use Terraform to build a "self-serve" infrastructure model that lets product teams manage their own infrastructure independently. You can create and use Terraform modules that codify the standards for deploying and managing services in your organization, allowing teams to efficiently deploy services in compliance with your organization's practices. Terraform Cloud can also integrate with ticketing systems like ServiceNow to automatically generate new infrastructure requests.

Soln: The goal of this task is to use Terraform to create a reusable infrastructure model enabling teams to deploy and manage their own resources without involving central operations.

① Understand the self serve infrastructure Model:

In large organizations product teams often request infrastructure resources from a central ops team. This can be repetitive and time-consuming.

By using Terraform you can create reusable modules that product teams can use independently to deploy their own infrastructure based on organization standards.

(2) Creating Terraform Modules:

modules allow you to reuse Terraform code. Create a module that defines a common infrastructure component, such as an EC2 instance.

Teams can use this module in Terraform configurations:

```
module "ec2" {  
    source = ".\ec2-instance"  
    ami-id = "ami-0abcd1234"  
    instance-name = "team-app-instance"
```

③ Automating with Terraform Cloud and Ticketing Systems:

Terraform Cloud allows you to collaborate on infrastructure deployments. It can be integrated with a ticketing system like ServiceNow to automate infrastructure requests.

This integration ensures that infrastructure is deployed in compliance with the organization security and governance standards.

Key Tools to Use:

- ① Serverless Framework: for deploying REST API's
- ② SonarQube/SonarCloud: for code quality analysis.
- ③ SonarLint in IntelliJ/Eclipse for on-the-fly Java Analysis.
- ④ Terraform: for infrastructure automation and self-service infrastructure.

✓