

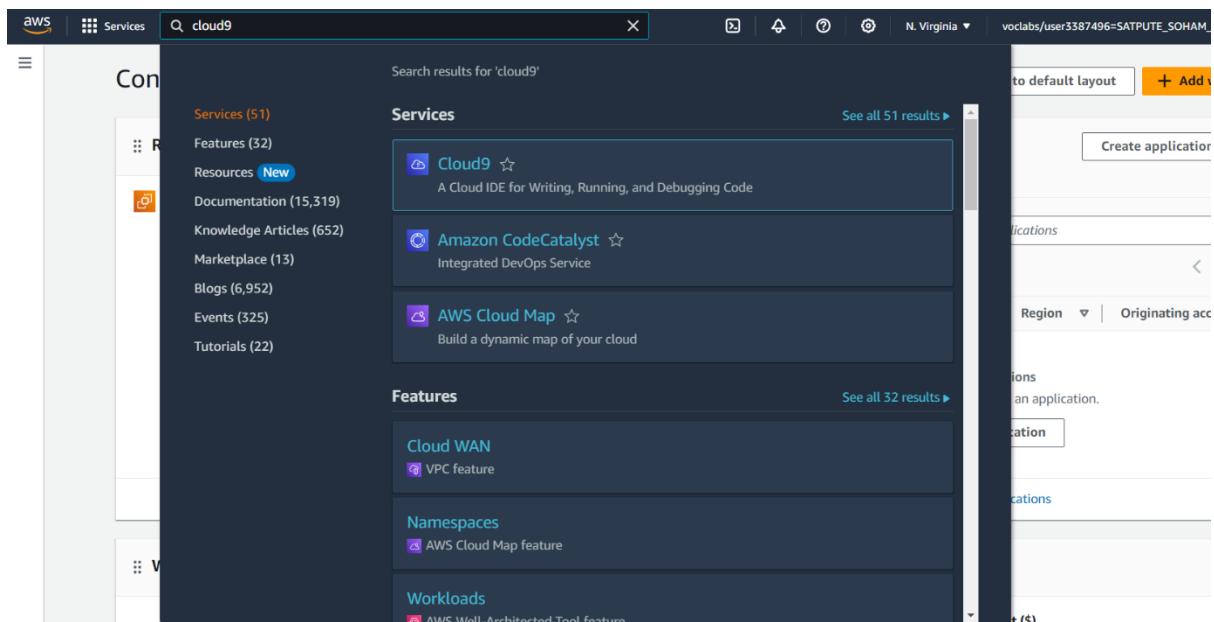
EXPERIMENT NO.1

Name: Soham Satpute

Class:D15A Roll No:52

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

1. Hosting using Cloud 9:



Click to go back, hold to see history

Services Search [Alt+S] N. Virginia vocabs/user3387496=SATPUTE_SOHAM_S

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

AWS Cloud9 > Environments > Create environment

Create environment [Info](#)

Details

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

Description – *optional* Limit 200 characters.

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

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

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

aws Services Search [Alt+S] N. Virginia vocabs/user3387496=SATPUTE_SOHAM_SUDHIR @ 2292-1357-6924

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 needs.

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](#)

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. [Find out 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. [Find out more](#)

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

Network settings [Info](#)

Connection
How your environment is accessed.

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

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

VPC settings [Info](#)
Amazon Virtual Private Cloud (VPC)
The VPC that your environment will access. To allow the AWS Cloud9 environment to connect to its EC2 instance, attach an internet gateway (IGW) to your VPC. [Create new VPC](#)
vpc-03d1ce76af665f00f
Name –

Subnet
Used to setup your VPC configuration. To use a private subnet, select AWS Systems Manager (SSM) as the connection type. [Create new subnet](#)
No preference
Uses default subnet in any Availability Zone

Tags – optional [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.

AWS Cloud9

Successfully created DEMO 1. To get the most out of your environment, 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. [Find out more](#)

Environments (1)

Name	Cloud9 IDE	Environment type	Connection	Permission	Owner ARN
DEMO 1	Open	EC2 instance	Secure Shell (SSH)	Owner	arn:aws:sts::229213576924:assumed-role/voclabs/user3387496=SATPUTE_SOHAM_SUDHIR

[CloudShell](#) [Feedback](#)

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

Q iam X

Search results for 'iam'

Services (11) Features (24) Resources New Documentation (59,469) Knowledge Articles (459) Marketplace (864) Blogs (1,846) Events (12) Tutorials (1)

Services See all 11 results ▶

IAM ☆ Manage access to AWS resources

IAM Identity Center ☆ Manage workforce user access to multiple AWS accounts and cloud applications

Resource Access Manager ☆ Share AWS resources with other accounts or AWS Organizations

Features See all 24 results ▶

Groups IAM feature

Roles IAM feature

Roles Anywhere

This screenshot shows the AWS search results for the query 'iam'. The left sidebar contains links for Services (11), Features (24), Resources (New), Documentation (59,469), Knowledge Articles (459), Marketplace (864), Blogs (1,846), Events (12), and Tutorials (1). The main content area is divided into 'Services' and 'Features'. Under 'Services', there are three items: 'IAM' (Manage access to AWS resources), 'IAM Identity Center' (Manage workforce user access to multiple AWS accounts and cloud applications), and 'Resource Access Manager' (Share AWS resources with other accounts or AWS Organizations). Under 'Features', there are three items: 'Groups' (IAM feature), 'Roles' (IAM feature), and 'Roles Anywhere'. Each item has a small icon and a star rating.

AWS Console Home Services Search Global v vocabs/user3387496=SATPUTE_SOHAM_SUDHIR @ 2292-1357-6924 ▾

Identity and Access Management (IAM)

IAM > Users

Users (0) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

Create user

No resources to display

IAM > Users > Create user

Step 1 Specify user details

Step 2 Set permissions

Step 3 Review and create

Step 4 Retrieve password

User details

User name: Soham

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

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

Console password

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

Custom password
Enter a custom password for the user.

- Must be at least 8 characters long
- Must include at least three of the following mix of character types: uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), and symbols ! @ # \$ % ^ & * () _ + - (Hyphen) = [{ }] !

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.

Review and create

Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

User details

User name Soham	Console password type Custom password	Require password reset Yes
--------------------	--	-------------------------------

Permissions summary

Name	Type	Used as
IAMUserChangePassword	AWS managed	Permissions policy

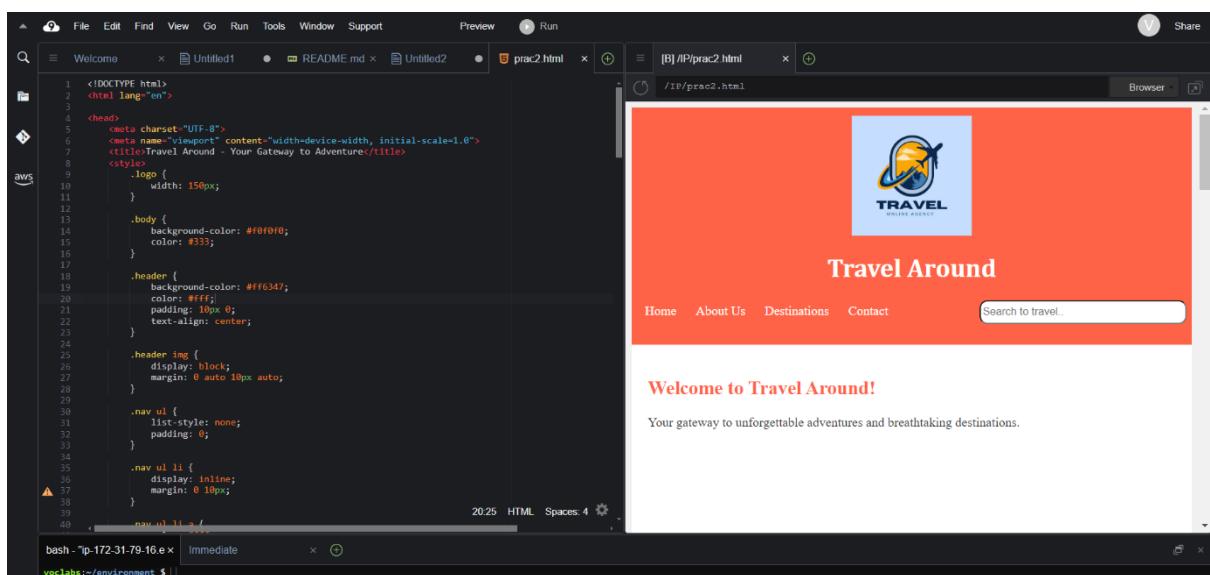
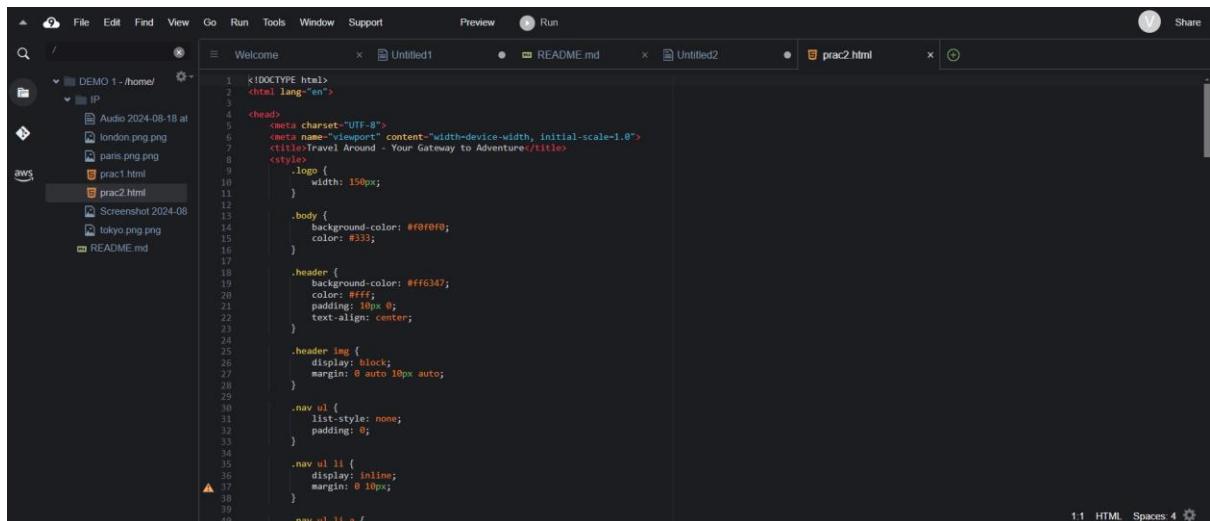
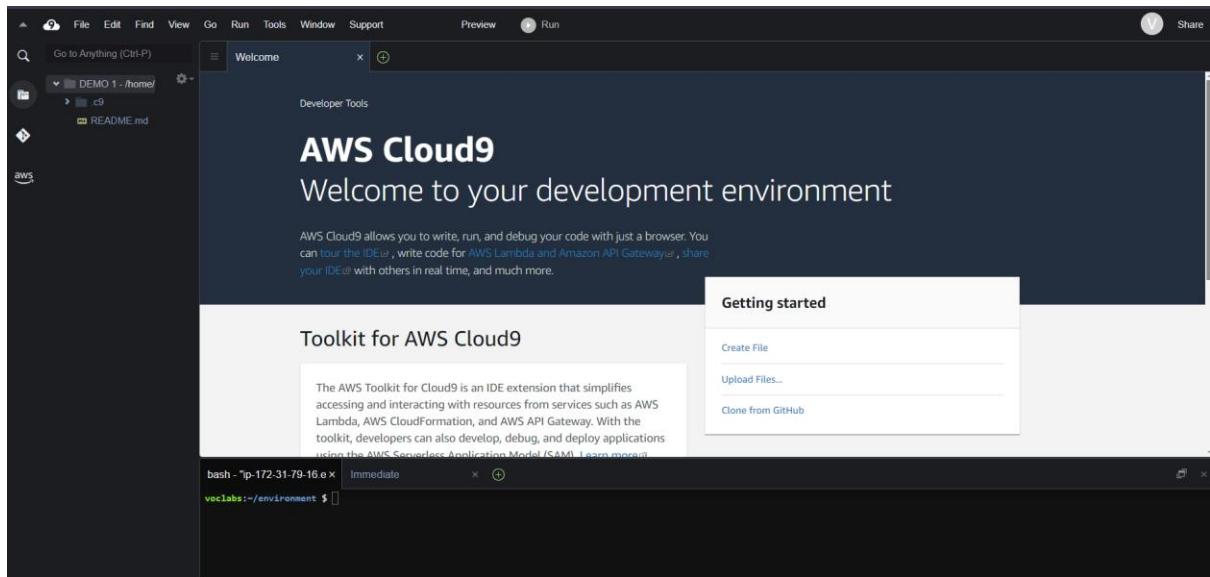
Tags - optional

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

Add new tag

<https://us-east-1.console.aws.amazon.com/console/home?region=us-east-1> © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



2. Hosting using EC2 INSTANCE:

The screenshot shows two overlapping AWS interface windows. The top window is the 'EC2 Dashboard' under the 'Services' tab, showing resource counts for Instances (running: 3), Auto Scaling Groups (0), Capacity Reservations (0), Dedicated Hosts (0), Elastic IPs (0), Instances (3), Key pairs (1), Load balancers (0), Placement groups (0), Security groups (3), Snapshots (0), and Volumes (3). It also displays 'Account attributes' like the Default VPC (vpc-03d1ce76af665f00f) and 'Explore AWS' sections for Best Price-Performance and Spot Instances. The bottom window is the 'Launch instance' wizard, starting with the 'Name and tags' step where 'Sohamm' is entered. It then moves to the 'Application and OS Images (Amazon Machine Image)' step, where a search bar is shown. A modal box provides information about the Free tier, stating: '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'. The final step shows summary details and a 'Launch instance' button.

AWS Console Home

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

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

Recents Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Linux

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

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type

Free tier eligible

Description

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

Architecture 64-bit (x86)

AMI ID ami-0e86e20dae9224db8

Verified provider

Software Image (AMI)

Canonical, Ubuntu, 24.04, amd6...read more

ami-0e86e20dae9224db8

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 IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel Launch instance Review commands

Services Search

Instance type Info | Get advice

Instance type

t2.micro

Family: t2 1 vCPU 1 GiB Memory Current generation: true

On-Demand Windows base pricing: 0.0162 USD per Hour

On-Demand SUSE base pricing: 0.0116 USD per Hour

On-Demand RHEL base pricing: 0.026 USD per Hour

On-Demand Linux base pricing: 0.0116 USD per Hour

Free tier eligible

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

Key pair (login) Info

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

Key pair name - required

vockey

Create new key pair

vockey

Network settings Info

Network Info

vpc-03d1ce76af665f00f

Cancel Launch instance Review commands

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 IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Services Search

Network settings Info

Network Info

vpc-03d1ce76af665f00f

Subnet Info

No preference (Default subnet in any availability zone)

Auto-assign public IP Info

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) Info

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

Create security group Select existing security group

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

Allow SSH traffic from Anywhere 0.0.0.0/0

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

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

Cancel Launch instance Review commands

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 IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

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

1x GiB [▼](#) Root volume (Not encrypted)

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

Add new volume

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

Click refresh to view backup information [Edit](#)

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

0 x File systems [Edit](#)

Success Successfully initiated launch of instance (i-013988963664b36cd)

[Launch log](#)

Next Steps

What would you like to do next with this instance, for example "create alarm" or "create backup"

[Create billing and free tier usage alerts](#) [Connect to your instance](#) [Connect an RDS database](#) [Create EBS snapshot policy](#)

To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds.

Once your instance is running, log into it from your local computer.

[Connect to instance](#) [Learn more](#)

Configure the connection between an EC2 instance and a database to allow traffic flow between them.

[Connect an RDS database](#) [Create a new RDS database](#) [Learn more](#)

```
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

System information as of Fri Aug 23 13:49:20 UTC 2024

System load: 0.53 Processes: 105
Usage of /: 22.8% of 6.71GB Users logged in: 0
Memory usage: 21% IPv4 address for enX0: 172.31.93.140
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.
```

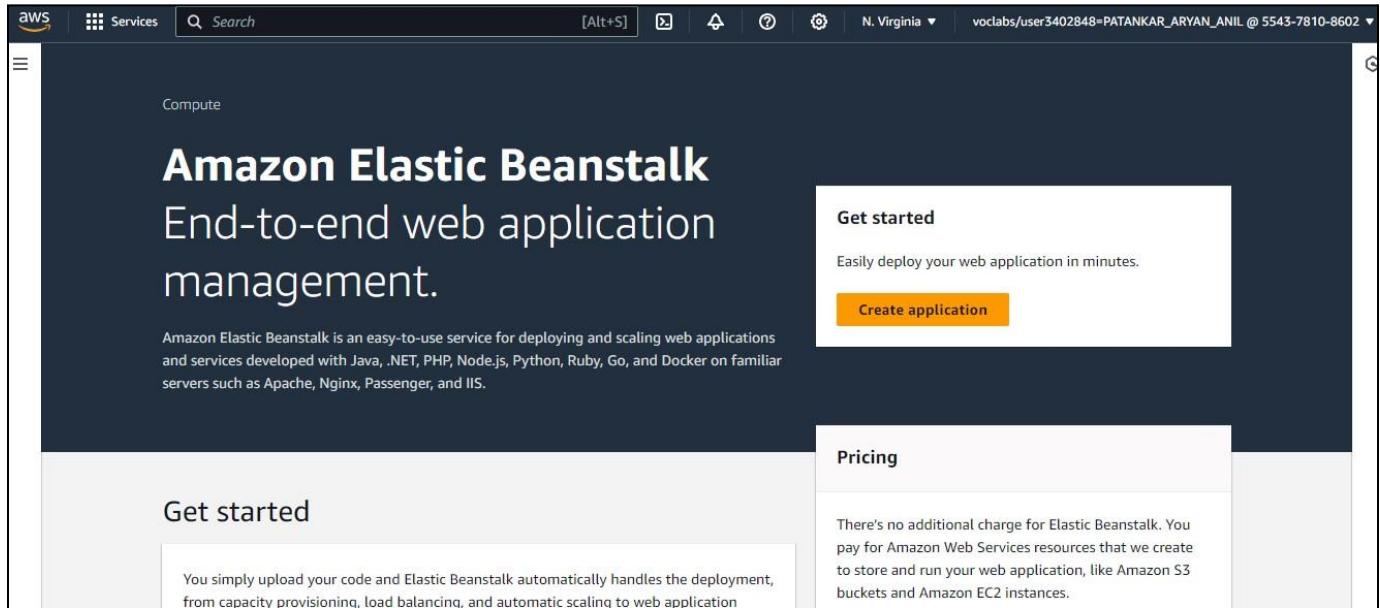
ADVANCE DEVOPS EXPERIMENT NO.2

Name: Soham Satpute

Class:D15A

Roll No:52

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.



Step 1: Search for **Elastic Beanstalk** in the search bar next to services section and you would see the following page.

Step 2: Create a new application and proceed with the following settings

Configure environment [Info](#)

Environment tier [Info](#)

Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.

Web server environment
Run a website, web application, or web API that serves HTTP requests. [Learn more](#)

Worker environment
Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. [Learn more](#)

Application information [Info](#)

Application name

Aryan27

Maximum length of 100 characters.

► Application tags (optional)

Platform [Info](#)

Platform type

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

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

Platform

PHP

Platform branch

PHP 8.3 running on 64bit Amazon Linux 2023

Platform version

4.3.2 (Recommended)

Application code Info

Sample application

Existing version

Application versions that you have uploaded.

Upload your code

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

Presets Info

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

Configuration presets

Single instance (free tier eligible)

Single instance (using spot instance)

High availability

High availability (using spot and on-demand instances)

Custom configuration

Cancel

Next

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

Privacy

Term

Step 3: Create a new service role as given below, if an existing service role with the same name does not exist. Proceed with the steps given below.

Service access

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

Service role

- Create and use new service role
 Use an existing service role

Service role name

Enter the name for an IAM role that Elastic Beanstalk will create to assume as a service role. Beanstalk will attach the required managed policies to it.

aws-elasticbeanstalk-service-role

[View permission details](#)

EC2 key pair

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

vockey



EC2 instance profile

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

EMR_EC2_DefaultRole



[View permission details](#)

▼ Instances Info

Configure the Amazon EC2 instances that run your application.

Root volume (boot device)

Root volume type

(Container default) ▾

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 instance

125 MiB/s

AWS Services Search [Alt+S] Stockholm AryanPatankar

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

▼ Monitoring Info

Health reporting
Enhanced health reporting provides free real-time application and operating system monitoring of the instances and other resources in your environment. The EnvironmentHealth custom metric is provided free with enhanced health reporting. Additional charges apply for each custom metric. For more information, see [Amazon CloudWatch Pricing](#).

System
 Basic
 Enhanced

Health event streaming to CloudWatch Logs
Configure Elastic Beanstalk to stream environment health events to CloudWatch Logs. You can set the retention up to a maximum of ten years and configure Elastic Beanstalk to delete the logs when you terminate your environment.

Log streaming
 Activated (standard CloudWatch charges apply.)

Retention
7

Step 4: Review each step along with the selected options and verify that the correct options have been chosen.

Review Info

Step 1: Configure environment

Environment information	
Environment tier	Application name
Web server environment	Aryan27
Environment name	Application code
Aryan27-env	Sample application
Platform	
arn:aws:elasticbeanstalk:us-east-1::platform/PHP 8.3 running on 64bit Amazon Linux 2023/4.3.2	

[Edit](#)

Step 2: Configure service access

Edit

Service access Info

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

Service role	EC2 instance profile
am:aws:iam::405894863107:role/service-role/aws-elasticbeanstalk-service-role	aws-elasticbeanstalk-ec2-role

Step 3: Set up networking, database, and tags

Edit

Networking, database, and tags Info

Configure VPC settings, and subnets for your environment's EC2 instances and load balancer. Set up an Amazon RDS database that's integrated with your environment.

Network

VPC	Public IP address	Instance subnets
vpc-0bf7d7d872a737f13	false	subnet-035fe38d8d742329e,subnet-0a7c9c6dedec1325d

Step 5: Configure updates, monitoring, and logging

Edit

Updates, monitoring, and logging Info

Define when and how Elastic Beanstalk deploys changes to your environment. Manage your application's monitoring and logging settings, instances, and other environment resources.

Monitoring

System	Cloudwatch custom metrics - instance	Cloudwatch custom metrics - environment
enhanced	—	—
Log streaming	Retention	Lifecycle
Deactivated	7	false

Updates

Managed updates	Deployment batch size	Deployment batch size type
Activated	100	Percentage

Platform software

Lifecycle	Log streaming	Allow URL fopen
false	Deactivated	On
Display errors	Document root	Max execution time
Off	-	60
Memory limit	Zlib output compression	Proxy server
256M	Off	nginx
Logs retention	Rotate logs	Update level
7	Deactivated	minor
X-Ray enabled		

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

Key	▲	Value	▼
No environment properties			
There are no environment properties defined			

[Cancel](#)[Previous](#)[Submit](#)

Step 5: After clicking on the submit button, you would notice that the Elastic Beanstalk environment is being created and it may take some time for the environment to load completely.

The screenshot shows the AWS Elastic Beanstalk console. On the left, a sidebar lists 'Applications', 'Environments', and 'Change history'. Under 'Environments', 'Application: Aryan27' is selected, showing 'Application versions' and 'Saved configurations'. Below this, 'Environment: Aryan27-env' is selected, with options like 'Go to environment', 'Configuration', 'Events', 'Health', 'Logs', 'Monitoring', 'Alarms', 'Managed updates', and 'Tags'. The main content area displays the 'Aryan27-env' environment overview. It includes sections for 'Environment overview' (Health: Unknown, Environment ID: e-83gzhjzgmq, Domain: -, Application name: Aryan27) and 'Platform' (Platform: PHP 8.3 running on 64bit Amazon Linux 2023/4.3.2, Running version: -, Platform state: Supported). At the bottom, tabs for 'Events', 'Health', 'Logs', 'Monitoring', 'Alarms', 'Managed updates', and 'Tags' are visible, along with an 'Events (2)' section. The top navigation bar shows 'Elastic Beanstalk is launching your environment. This will take a few minutes.' and the AWS logo.

The screenshot shows the 'Trusted entity type' section of an AWS IAM role or policy configuration. It lists five options: 'AWS service' (selected), 'AWS account', 'Web identity', 'SAML 2.0 federation', and 'Custom trust policy'. Each option has a brief description. Below this, a 'Use case' section states 'Allow an AWS service like EC2, Lambda, or others to perform actions in this account.' A 'Service or use case' dropdown menu at the bottom contains the value 'EC2'.

Step 6: Meanwhile, if a role is already not defined, then you need to create a new role for the elastic beanstalk and ensure there is a blue checkmark on the following three permissions given below.

Role Name	Type	Description
AWSElasticBeanstalkCustomPlatformforEC2Role	AWS managed	Provide the instance
AWSElasticBeanstalkEnhancedHealth	AWS managed	AWS Elastic Beanstal
AWSElasticBeanstalkManagedUpdatesCustomerR...	AWS managed	This policy is for the
<input checked="" type="checkbox"/> AWSElasticBeanstalkMulticontainerDocker	AWS managed	Provide the instance
AWSElasticBeanstalkReadOnly	AWS managed	Grants read-only per
AWSElasticBeanstalkRoleCore	AWS managed	AWSElasticBeanstalk
AWSElasticBeanstalkRoleCWL	AWS managed	(Elastic Beanstalk op
AWSElasticBeanstalkRoleECS	AWS managed	(Elastic Beanstalk op
AWSElasticBeanstalkRoleRDS	AWS managed	(Elastic Beanstalk op
AWSElasticBeanstalkRoleSNS	AWS managed	(Elastic Beanstalk op
AWSElasticBeanstalkRoleWorkerTier	AWS managed	(Elastic Beanstalk op
<input checked="" type="checkbox"/> AWSElasticBeanstalkWebTier	AWS managed	Provide the instance
<input checked="" type="checkbox"/> AWSElasticBeanstalkWorkerTier	AWS managed	Provide the instance

Step 7: Enter a role name and proceed. You would notice the role being successfully created after some time.

Step 2
Add permissions
Step 3
Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.

Maximum 64 characters. Use alphanumeric and '+,-,@,_' characters.

Description
Add a short explanation for this role.

Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+=,. @-/[\{\}!#\$%^&*();:_`~`]

Step 1: Select trusted entities Edit

The screenshot shows the AWS IAM Roles page. A green banner at the top indicates that a role named "aws-elastic-beanstalk-ec2-role" has been created. The main table lists four roles:

Role name	Trusted entities	Last activity
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service)	-
AWSServiceRoleForSupport	AWS Service: support (Service-Linker)	-
aws-elasticbeanstalk-service-role	AWS Service: elasticbeanstalk	-
aws-elastic-beanstalk-ec2-role	AWS Service: ec2	-

Step 8(optional): Search for CloudFormation as it helps you to manage AWS resources in a text file or a template.

The screenshot shows the AWS CloudFormation search results page. The search term "cloud formation" is entered in the search bar. The results section displays three services:

- CloudFormation** ☆ Create and Manage Resources with Templates
- Application Composer** ☆ Visually design and build modern applications quickly
- Athena** ☆ Serverless interactive analytics service

Here, the stacks option given below is a collection of AWS resources.

The screenshot shows the AWS CloudFormation Stacks page. It displays a single stack named "awseb-e-rh8w3tywxk-stack" with the status "CREATE_COMPLETE". The stack was created on 2024-08-21 14:36:08 UTC+0530. The description indicates it is an AWS Elastic Beanstalk environment (Name: 'Aryan27-env-1' Id: 'e-rh8w3tywxk').

The screenshot shows the AWS CloudFormation console. On the left, the navigation pane includes 'Stack details', 'Drifts', 'StackSets', 'Exports', 'Application Composer', 'laC generator', 'Registry', 'Public extensions', 'Activated extensions', 'Publisher', and 'Spotlight'. The main area displays a stack named 'awseb-e-rh8w3tywxk-stack' with a status of 'CREATE_COMPLETE' at '2024-08-21 14:36:08 UTC+0530'. The 'Template' tab is selected, showing the following JSON code:

```
{
  "Outputs": {},
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "InstanceTypeFamily": {
      "NoEcho": "true",
      "Type": "String",
      "Description": "WebServer EC2 instance type family"
    },
    "LogPublicationControl": {
      "NoEcho": "true",
      "Type": "String",
      "Description": "If true customer service logs will be published to S3."
    },
    "AllowedValues": [
      "true",
      "false"
    ]
  }
}
```

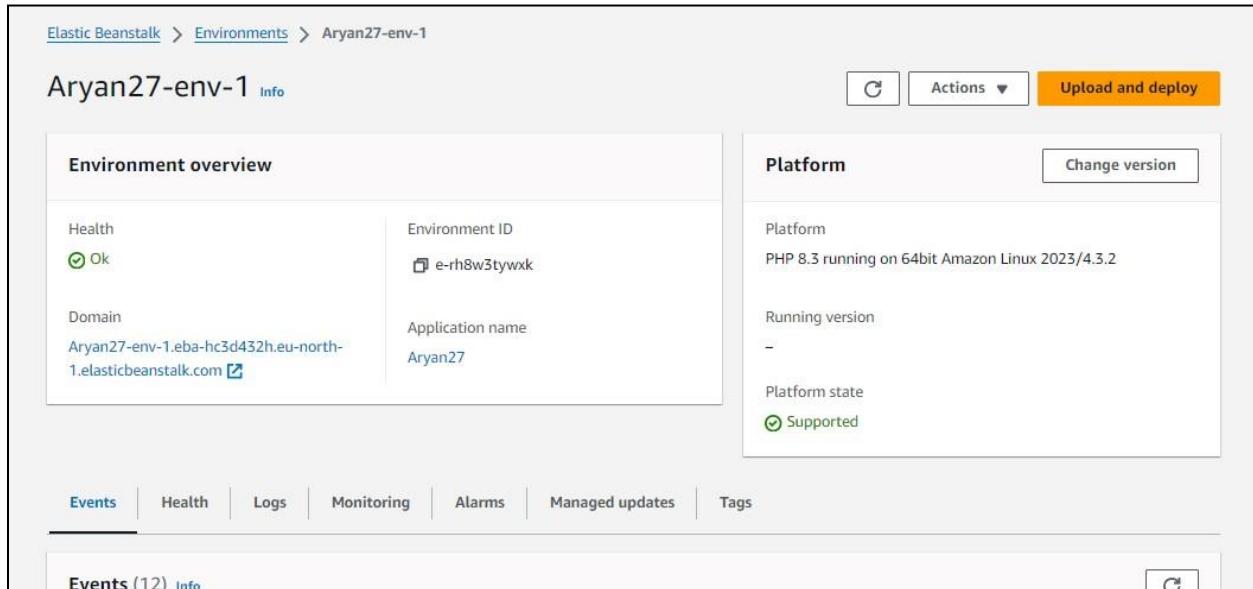
The screenshot shows the AWS CloudFormation Application Composer in console mode. The top bar includes 'CloudFormation console mode' and 'Menu'. A message box says 'Looking for CloudFormation Designer? Application Composer in CloudFormation console mode is an improvement from CloudFormation Designer. Please provide your feedback.' with a 'Go to Designer' button. The main area is a canvas with several components: 'AWSEBAutoScalingLaunchConfiguration', 'AWSEBInstanceLaunchWaitHandle', 'AWSEBEIP', 'AWSEBBeanstalkMetadata', and 'AWSEBInstanceLaunchWaitCondition'. Components are connected by dashed lines, indicating dependencies or relationships between them.

Step 9: Now, we search for EC2 in the services section and we notice that an instance of the Elastic Beanstalk app has already been created and it is running.

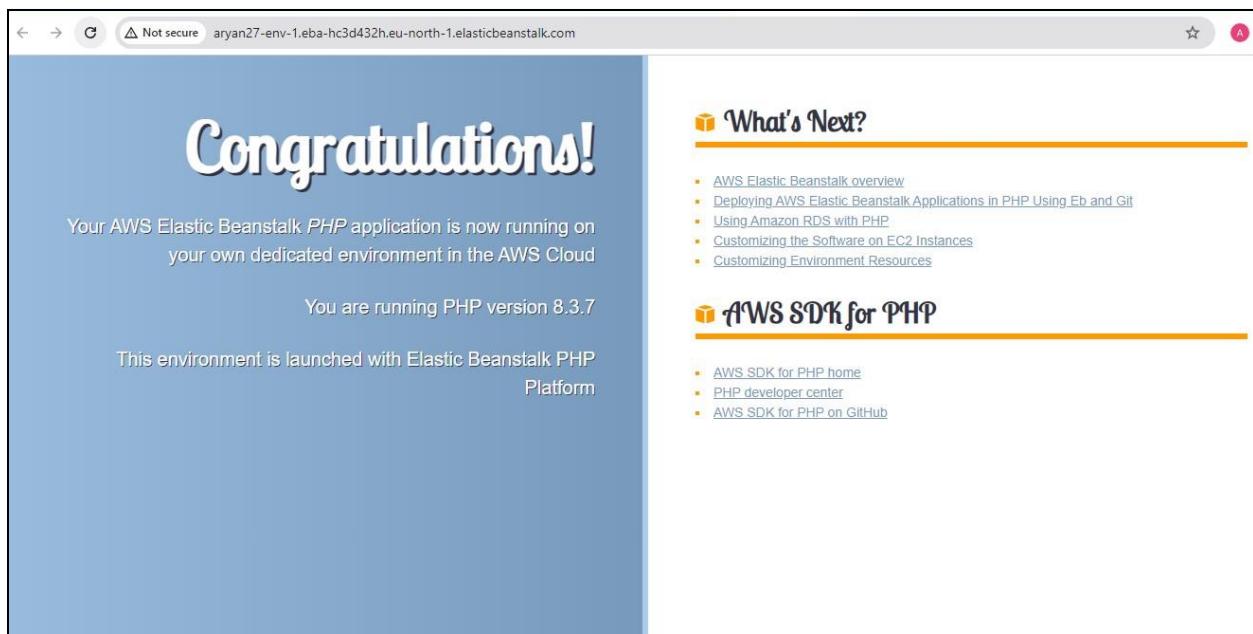
The screenshot shows the AWS EC2 Instances page. The top bar includes 'Instances (1) Info', 'Last updated less than a minute ago', 'Connect', 'Actions', and 'Launch instances'. A search bar says 'Find Instance by attribute or tag (case-sensitive)' with 'All states' dropdown. A filter bar says 'Instance state = running'. The main table lists one instance:

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Publ...
<input type="checkbox"/>	Aryan27-env-1	i-0c23baf27044b934d	Running	t3.micro	2/2 checks passed	View alarms	eu-north-1c	ec2-1...

Step 10: Click on the domain link given below, after which we are redirected to a Congratulations page implying that our sample PHP application has been successfully hosted.



The screenshot shows the AWS Elastic Beanstalk Environment Overview page for 'Aryan27-env-1'. The 'Events' tab is selected, showing 12 events. The environment status is 'Ok'. The platform is listed as 'PHP 8.3 running on 64bit Amazon Linux 2023/4.3.2'. The application name is 'Aryan27'. The domain is 'Aryan27-env-1.eba-hc3d432h.eu-north-1.elasticbeanstalk.com'. The 'Platform state' is 'Supported'.



The screenshot shows the 'Congratulations!' page from the AWS Elastic Beanstalk environment. It states: 'Your AWS Elastic Beanstalk PHP application is now running on your own dedicated environment in the AWS Cloud'. Below it says 'You are running PHP version 8.3.7'. To the right, there are sections for 'What's Next?' and 'AWS SDK for PHP' with links to various AWS resources.

Step 11: Now, we will be deploying our website using CodePipeline, so follow all the steps given below and proceed.

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

No more than 100 characters

Pipeline type

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

Execution mode
Choose the execution mode for your pipeline. This determines how the pipeline is run.

Superseded
A more recent execution can overtake an older one. This is the default.

Queued (Pipeline type V2 required)
Executions are processed one by one in the order that they are queued.

Parallel (Pipeline type V2 required)
Executions don't wait for other runs to complete before starting or finishing.

Service role

Step 12: In the source stage, choose GitHub v2 as the provider, then connect your GitHub account to AWS by creating a connection. You'd need your GitHub credentials and then you'd need to authorize and install AWS on the forked GitHub Repository.

Step 2 of 5

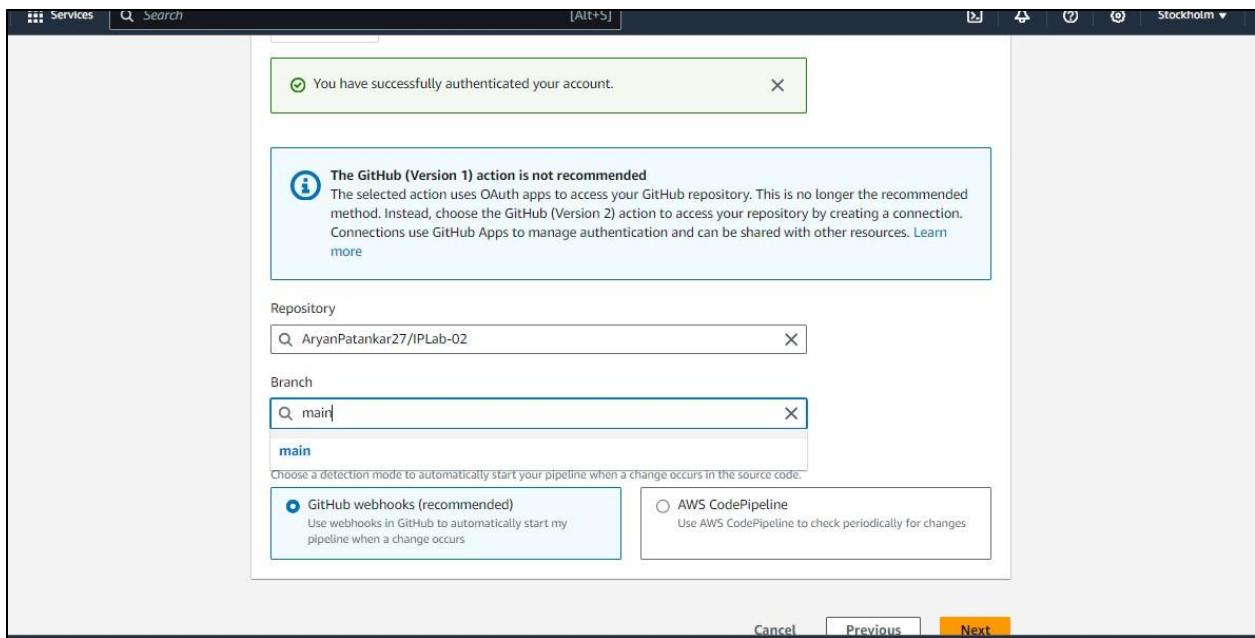
Source

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

Grant AWS CodePipeline access to your GitHub repository. This allows AWS CodePipeline to upload commits from GitHub to your pipeline.

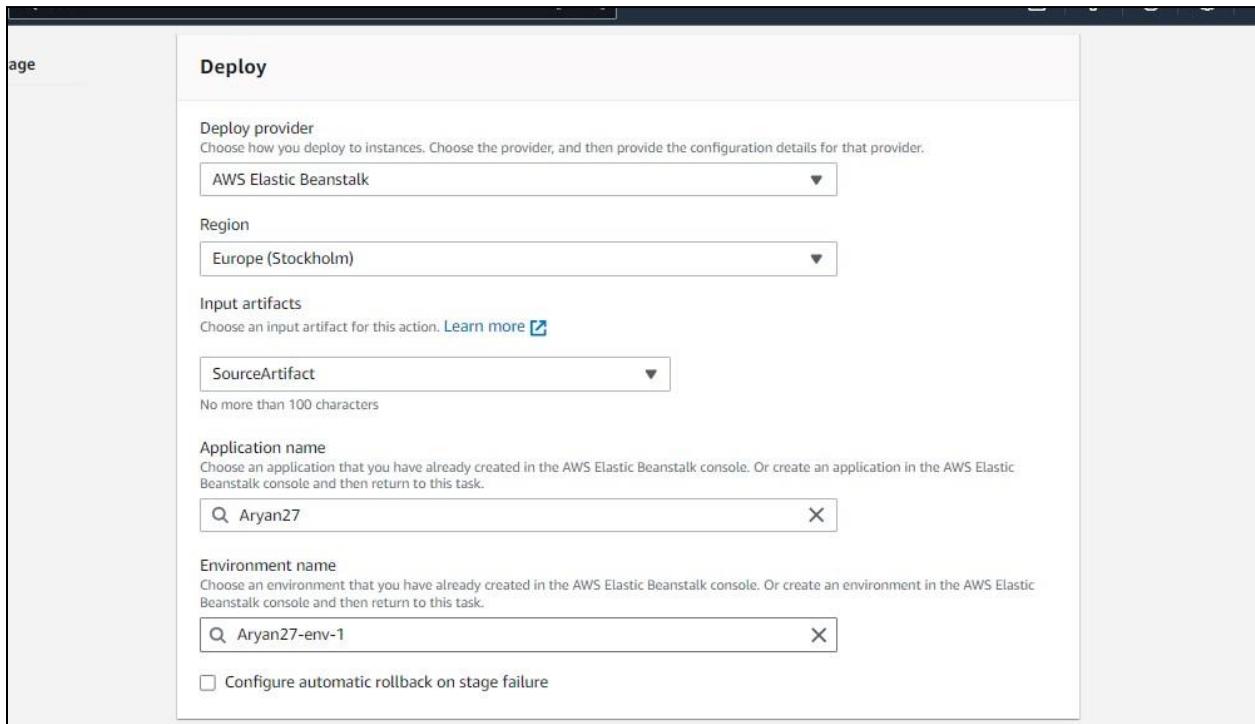
i You have successfully configured the action with the provider. X

i **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](#)



Then, simply choose this forked repository and the branch which you will be able to find in the search box. After that, click Continue and skip the build stage. Proceed to the Deployment stage.

Step 13: Choose Beanstalk as the Deploy Provider, same region as the Bucket and Beanstalk, name and environment name. Click Next, Review and create the pipeline.



Step 4: Add deploy stage

Deploy action provider

Deploy action provider
AWS Elastic Beanstalk

ApplicationName
Aryan27

EnvironmentName
Aryan27-env

Configure automatic rollback on stage failure
Disabled

[Cancel](#) [Previous](#) [Create pipeline](#)

Step 14: Review all the selected steps once.

Review Info

Step 5 of 5

Step 1: Choose pipeline settings

Pipeline settings

Pipeline name
Pipeline_Aryan

Pipeline type
V2

Execution mode
QUEUED

Artifact location
A new Amazon S3 bucket will be created as the default artifact store for your pipeline

Service role name
AWSCodePipelineServiceRole-eu-north-1-Pipeline_Aryan

Step 2: Add source stage

Source action provider

Source action provider

GitHub (Version 1)

PollForSourceChanges

false

Repo

IPLab-02

Owner

AryanPatankar27

Branch

main

Step 3: Add build stage

Build action provider

Build stage

No build

Step 4: Add deploy stage

Deploy action provider

Deploy action provider

AWS Elastic Beanstalk

ApplicationName

Aryan27

EnvironmentName

Aryan27-env-1

Configure automatic rollback on stage failure

Disabled

[Cancel](#)[Previous](#)[Create pipeline](#)

Step 15: In a few minutes, we will have our pipeline created. Once we have the success message on the

Deploy part, we can go ahead and check our URL provided in the EBS environment.

The screenshot shows the AWS CodePipeline console for a pipeline named "Pipeline_Aryan". The pipeline type is V2 and the execution mode is QUEUED. The execution ID is 5622b57f-b111-4e0e-9bb3-05f6fe3e98d8. The "Source" stage is listed as succeeded, with a GitHub commit history showing a successful build from 1 minute ago. A "View details" button is available. The "Deploy" stage is also listed as succeeded. A "Start rollback" button is located in the Deploy stage area. The overall status is green, indicating success.

This is the sample website we just created.

The screenshot shows a web browser window displaying the Amazon homepage. The URL in the address bar is Not secure aryan27-env-1.eba-hc3d432h.eu-north-1.elasticbeanstalk.com. The page features the Amazon logo and navigation links for Services, About Us, and Contact. Below the header, there is a section titled "Our Services" featuring the Amazon Online Retail and Prime Video logos. A descriptive text block states: "Amazon offers a vast selection of products, ranging from electronics to groceries. Our user-friendly platform makes shopping easy and convenient, ensuring that you find exactly what you need with just a few clicks." At the bottom of the page, there is a link for "Amazon Prime Membership".

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

Using S3 Bucket

Step 16: Now, we will be deploying our website using the S3 bucket. So proceed with the options as given below.

AWS Region
Europe (Stockholm) eu-north-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](#)
aryan2711

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

Object Ownership | [Info](#)

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

Object Ownership | [Info](#)

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

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

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

Object Ownership
Bucket owner enforced

Block Public Access settings for this bucket

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

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

Block public access to buckets and objects granted through new access control lists (ACLs)
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access

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](#)

⌚ Successfully created bucket "aryan2711"
To upload files and folders, or to configure additional bucket settings, choose [View details](#).

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

General purpose buckets		Directory buckets
General purpose buckets (3) Info All AWS Regions		
Buckets are containers for data stored in S3.		
<input type="text" value="Find buckets by name"/> < 1 > ⚙		
Name	AWS Region	IAM Access Analyzer
<input type="radio"/> aryan2711	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1
<input type="radio"/> codepipeline-eu-north-1-365572256475	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1
<input type="radio"/> elasticbeanstalk-eu-north-1-405894863107	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1

Step 17: Upload all the files that you want on your website that is to be hosted.

Upload [Info](#)

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](#)

Drag and drop files and folders you want to upload here, or choose **Add files** or **Add folder**.

Files and folders (11 Total, 545.5 KB)			
All files and folders in this table will be uploaded.			
<input type="text"/> Find by name			
<input checked="" type="checkbox"/>	Name	Folder	Type
<input checked="" type="checkbox"/>	download (1).jfif	IPLab-02-main/	image/jpeg
<input checked="" type="checkbox"/>	download (1).png	IPLab-02-main/	image/png
<input checked="" type="checkbox"/>	download (2).jfif	IPLab-02-main/	image/jpeg
<input checked="" type="checkbox"/>	download (2).png	IPLab-02-main/	image/png
<input checked="" type="checkbox"/>	download (3).png	IPLab-02-main/	image/png
<input checked="" type="checkbox"/>	download (4).png	IPLab-02-main/	image/png
<input checked="" type="checkbox"/>	download.png	IPLab-02-main/	image/png
<input checked="" type="checkbox"/>	index.html	IPLab-02-main/	text/html
<input checked="" type="checkbox"/>	introduction.mp3	IPLab-02-main/	audio/mpeg
<input checked="" type="checkbox"/>	promotional-video.mp4	IPLab-02-main/	video/mp4

☰ **Upload succeeded**
View details below.

[Configuration](#)

Files and folders (11 Total, 545.5 KB)						
<input type="text"/> Find by name						
Name	Folder	Type	Size	Status	Error	
download (1...)	IPLab-02-main/	image/jpeg	8.4 KB	✓ Succeeded	-	
download (1...)	IPLab-02-main/	image/png	3.3 KB	✓ Succeeded	-	
download (2...)	IPLab-02-main/	image/jpeg	5.5 KB	✓ Succeeded	-	
download (2...)	IPLab-02-main/	image/png	6.0 KB	✓ Succeeded	-	
download (3...)	IPLab-02-main/	image/png	6.1 KB	✓ Succeeded	-	
download (4...)	IPLab-02-main/	image/png	2.4 KB	✓ Succeeded	-	
download.pn...	IPLab-02-main/	image/png	4.7 KB	✓ Succeeded	-	
index.html	IPLab-02-main/	text/html	6.1 KB	✓ Succeeded	-	
introduction....	IPLab-02-main/	audio/mpeg	158.0 KB	✓ Succeeded	-	
promotional...	IPLab-02-main/	video/mp4	341.4 KB	✓ Succeeded	-	

Step 18: Here, if the upload of files is successful you would get the following page, meaning your website has been successfully hosted using the S3 bucket.

← → ⌂ Not secure aryan27-env-1.eba-hc3d432h.eu-north-1.elasticbeanstalk.com ☆ ⓘ

Amazon

amazon

Services About Us Contact

Our Services

amazon
Online Retail

Amazon offers a vast selection of products, ranging from electronics to groceries. Our user-friendly platform makes shopping easy and convenient, ensuring that you find exactly what you need with just a few clicks.

prime video

Amazon Prime Membership

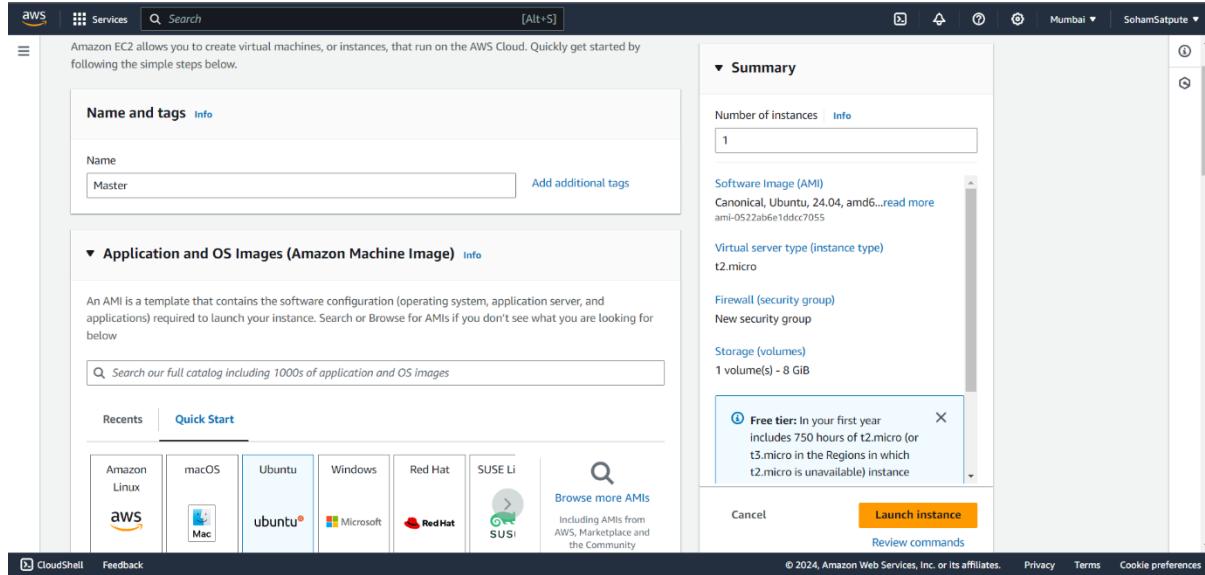
ADVANCE DEVOPS EXP-3

Name:Soham Satpute

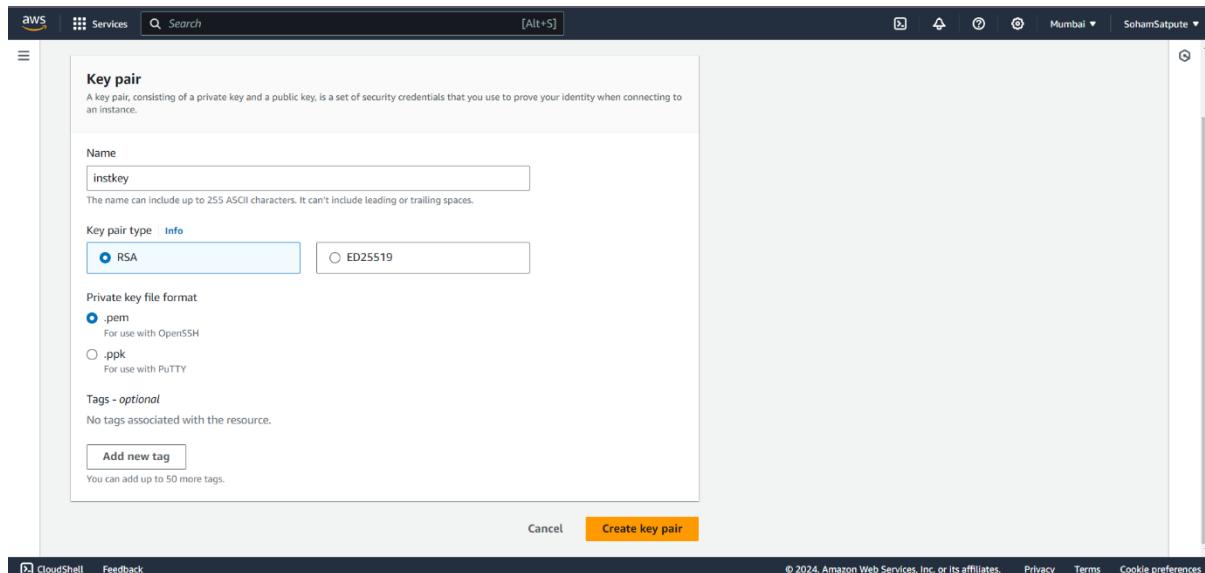
Roll No:52

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

Step 1: Create 2 Security Groups for Master and Nodes and add the following inbound rules in those groups:



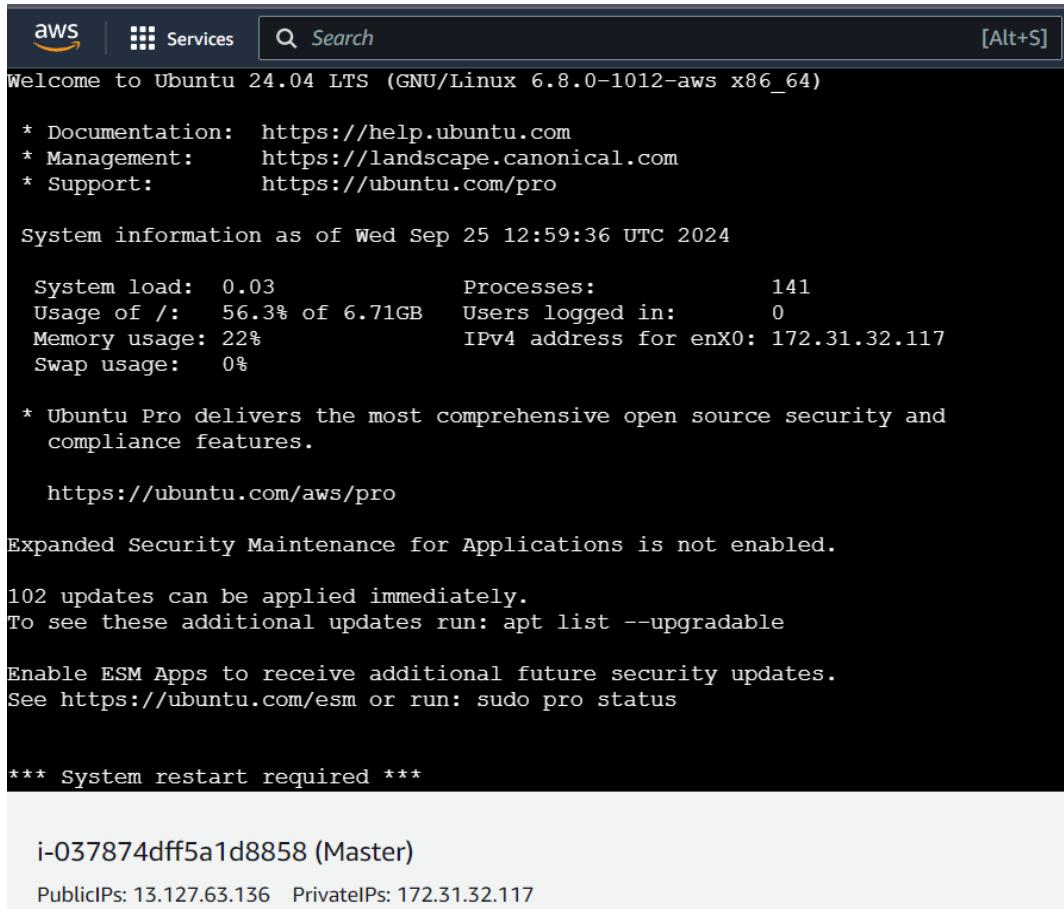
Generate a key pair for the same:



<input type="checkbox"/>	node 2	i-05c78ee26bbc9b179	.Pending	Q	Q	t2.medium	-	View alarms +	ap-south-1a	ec2-13-20
<input type="checkbox"/>	node 1	i-0b1270d945da2029e	Running	Q	Q	t2.medium	Initializing	View alarms +	ap-south-1a	ec2-13-23
<input type="checkbox"/>	Master	i-0f13653cf3d300e3	Running	Q	Q	t2.medium	2/2 checks passed	View alarms +	ap-south-1a	ec2-13-20

Step2:

Open Master and node on EC2 terminal:



```

aws | Services | Search [Alt+S]
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

System information as of Wed Sep 25 12:59:36 UTC 2024

System load: 0.03          Processes:           141
Usage of /: 56.3% of 6.71GB Users logged in:      0
Memory usage: 22%          IPv4 address for enx0: 172.31.32.117
Swap usage:  0%

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

  https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

102 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

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

*** System restart required ***

i-037874dff5a1d8858 (Master)
Public IPs: 13.127.63.136  Private IPs: 172.31.32.117

```

The screenshot shows a terminal window within the AWS Systems Manager Session Manager interface. The session is connected to an Ubuntu instance. The terminal displays various system statistics and security-related messages.

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

System information as of Wed Sep 25 13:00:16 UTC 2024

System load: 0.0      Processes: 128
Usage of /: 47.9% of 6.71GB  Users logged in: 0
Memory usage: 12%          IPv4 address for enX0: 172.31.33.216
Swap usage: 0%

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

102 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

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

*** System restart required ***
Last login: Wed Sep 25 05:20:50 2024 from 13.233.177.3
ubuntu@ip-172-31-33-216:~$
```

i-0ebf8336b19fd0d8f (node1)
PublicIPs: 13.201.135.29 PrivateIPs: 172.31.33.216

Step 3:

Install Docker

The screenshot shows a terminal window within the AWS Systems Manager Session Manager interface. The session is connected to an Ubuntu instance. The terminal displays the output of the `sudo apt-get update` command.

```
ubuntu@worker-node:~$ sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:13 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [530 kB]
Get:14 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [128 kB]
```

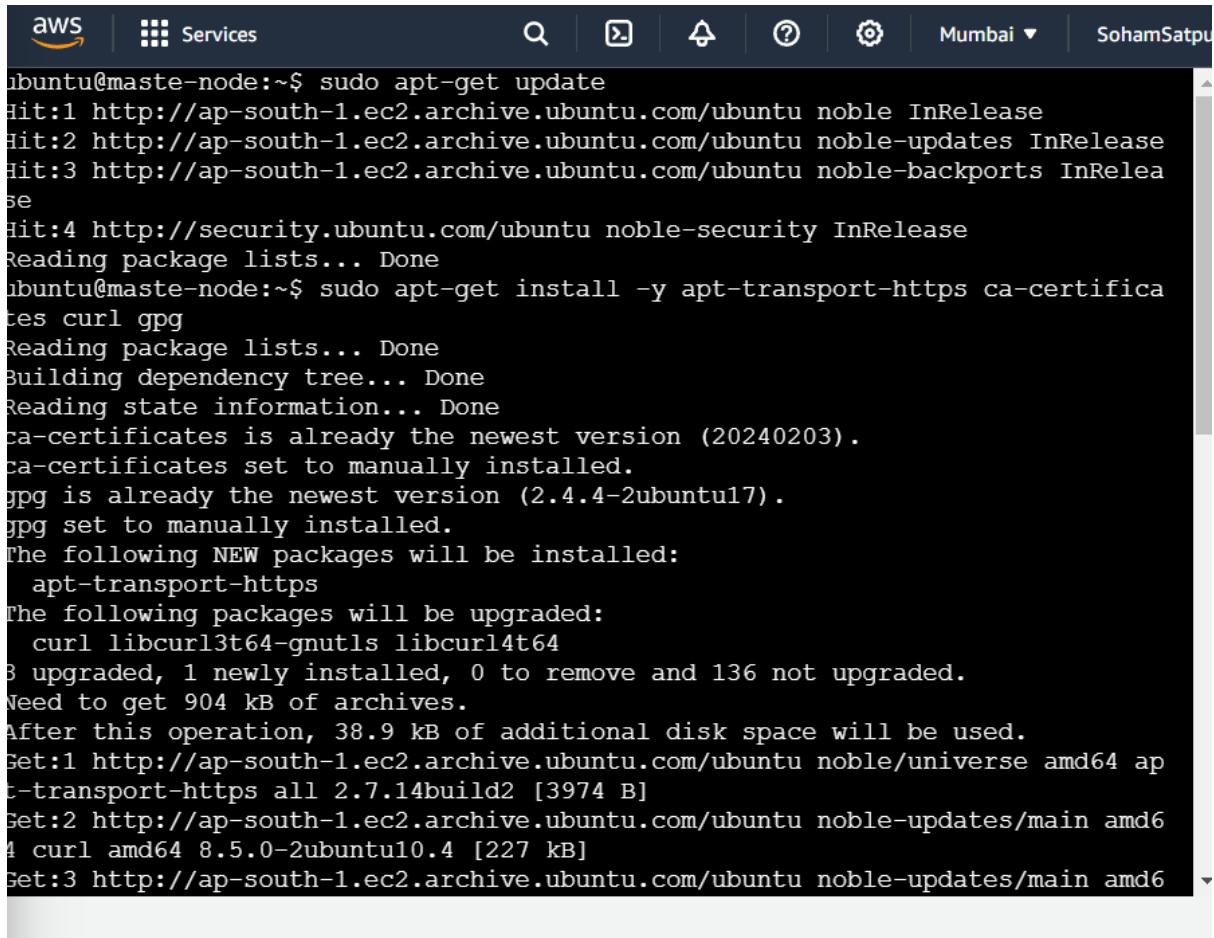
```
aws | Services | Q | X | ⓘ | ⓘ | Mumbai | SohamSatpute | 
ubuntu@maste-node:~$ sudo apt-get install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap
  docker-buildx docker-compose-v2 docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz runc
  ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 139 not upgraded.
Need to get 76.8 MB of archives.
After this operation, 289 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 pi
gz amd64 2.8-1 [65.6 kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 bridge
-utils amd64 1.7.1-1ubuntu2 [33.9 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd6
4 runc amd64 1.1.12-0ubuntu3.1 [8599 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64
  bridge-utils 2.8-1 [10.2 kB]
Fetched 76.8 MB in 0s (10.2 kB/s)
debconf: delaying configuration of package bridge-utils until user logs in
debconf: delaying configuration of package runc until user logs in
debconf: delaying configuration of package containerd until user logs in
debconf: delaying configuration of package dns-root-data until user logs in
debconf: delaying configuration of package dnsmasq-base until user logs in
debconf: delaying configuration of package pigz until user logs in
debconf: delaying configuration of package ubuntu-fan until user logs in
debconf: delaying configuration of package ifupdown until user logs in
debconf: delaying configuration of package aufs-tools until user logs in
debconf: delaying configuration of package cgroupfs-mount until user logs in
debconf: delaying configuration of package cgroup-lite until user logs in
debconf: delaying configuration of package debootstrap until user logs in
debconf: delaying configuration of package docker-buildx until user logs in
debconf: delaying configuration of package docker-compose-v2 until user logs in
debconf: delaying configuration of package docker-doc until user logs in
debconf: delaying configuration of package rinse until user logs in
debconf: delaying configuration of package zfs-fuse until user logs in
debconf: delaying configuration of package zfsutils until user logs in
```

```
aws | Services | Q | X | ⓘ | ⓘ | Mumbai | SohamSatpute | 
ubuntu@maste-node:~$ sudo systemctl enable docker
ubuntu@maste-node:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset>
   Active: active (running) since Tue 2024-09-24 17:27:29 UTC; 2min 12s ago
     TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
       Main PID: 2734 (dockerd)
         Tasks: 9
        Memory: 24.3M (peak: 24.5M)
          CPU: 221ms
        CGroup: /system.slice/docker.service
                  └─2734 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/c>

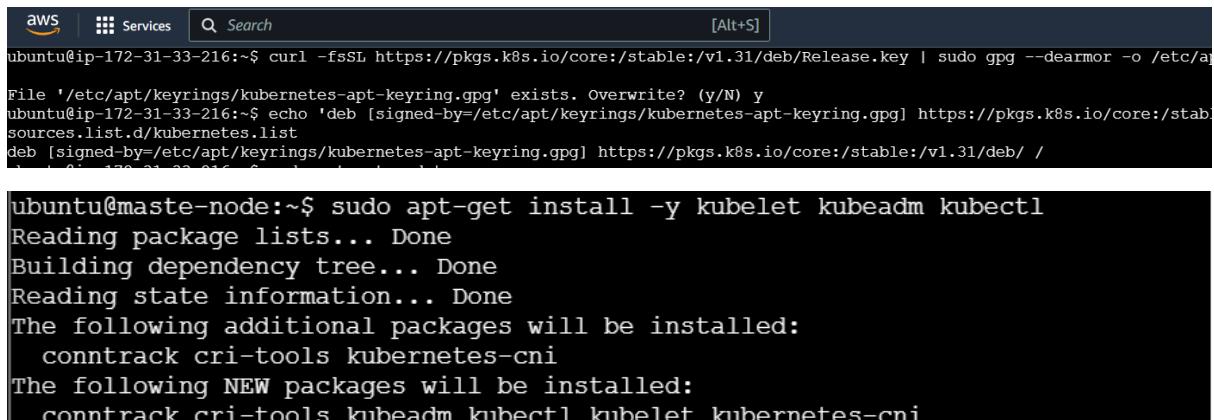
Sep 24 17:27:29 maste-node systemd[1]: Starting docker.service - Docker Appl>
Sep 24 17:27:29 maste-node dockerd[2734]: time="2024-09-24T17:27:29.199311437>
Sep 24 17:27:29 maste-node dockerd[2734]: time="2024-09-24T17:27:29.200465617>
Sep 24 17:27:29 maste-node dockerd[2734]: time="2024-09-24T17:27:29.636490377>
Sep 24 17:27:29 maste-node dockerd[2734]: time="2024-09-24T17:27:29.862800177>
Sep 24 17:27:29 maste-node dockerd[2734]: time="2024-09-24T17:27:29.880009307>
Sep 24 17:27:29 maste-node dockerd[2734]: time="2024-09-24T17:27:29.880104217>
Sep 24 17:27:29 maste-node dockerd[2734]: time="2024-09-24T17:27:29.927005647>
Sep 24 17:27:29 maste-node systemd[1]: Started docker.service - Docker Appl>
lines 1-21/21 (END)
```

Step 4:

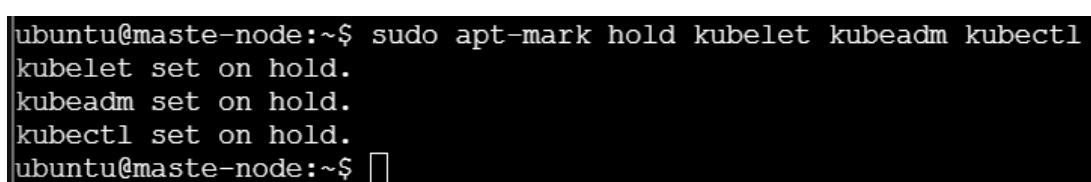
Install kubeadm, kubelet, kubectl:



```
ubuntu@maste-node:~$ sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
ubuntu@maste-node:~$ sudo apt-get install -y apt-transport-https ca-certificates curl gpg
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203).
ca-certificates set to manually installed.
gpg is already the newest version (2.4.4-2ubuntu17).
gpg set to manually installed.
The following NEW packages will be installed:
  apt-transport-https
The following packages will be upgraded:
  curl libcurl3t64-gnutls libcurl4t64
3 upgraded, 1 newly installed, 0 to remove and 136 not upgraded.
Need to get 904 kB of archives.
After this operation, 38.9 kB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 ap
t-transport-https all 2.7.14build2 [3974 B]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd6
4 curl amd64 8.5.0-2ubuntu10.4 [227 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd6
```



```
ubuntu@ip-172-31-33-216:~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/ap
file '/etc/apt/keyrings/kubernetes-apt-keyring.gpg' exists. Overwrite? (y/N) y
ubuntu@ip-172-31-33-216:~$ echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stabl
sources.list.d/kubernetes.list
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ '
ubuntu@maste-node:~$ sudo apt-get install -y kubelet kubeadm kubectl
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
```



```
ubuntu@maste-node:~$ sudo apt-mark hold kubelet kubeadm kubectl
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
ubuntu@maste-node:~$ 
```

Step5:

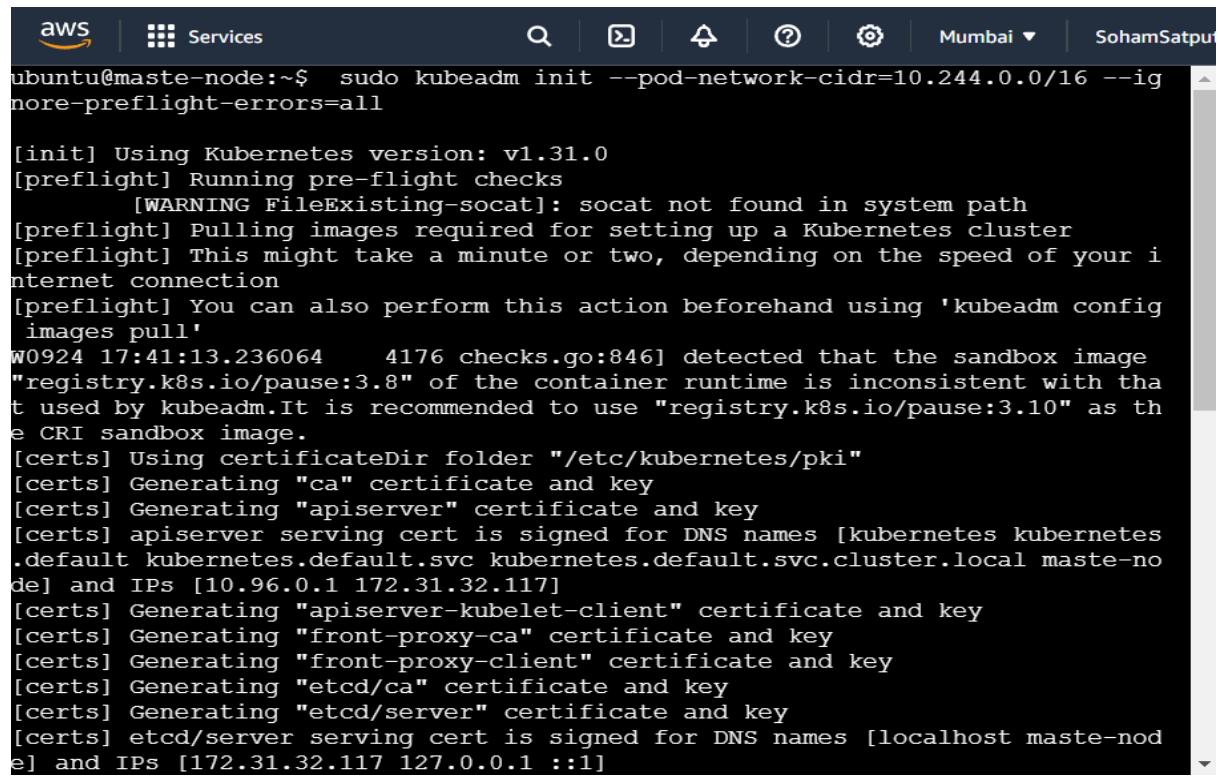
Disable Swap (Kubernetes requires swap to be off):

```
ubuntu@maste-node:~$ sudo swapoff -a  
ubuntu@maste-node:~$ █
```

Step 6:

Initialize the Kubernetes Cluster on Master Node On the master node:

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



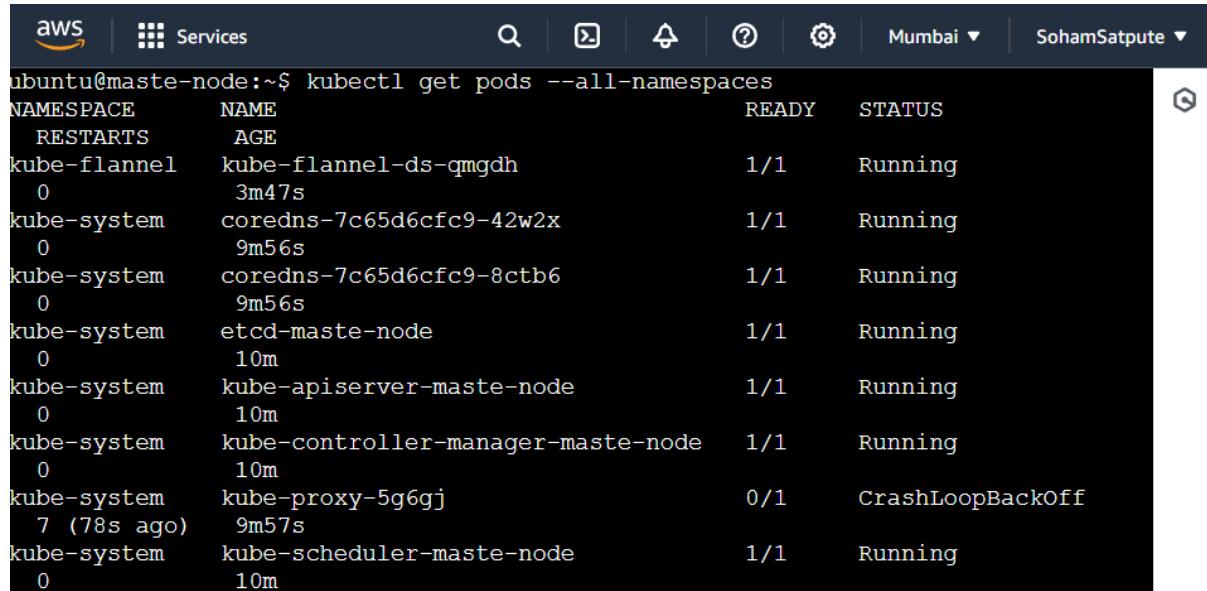
The screenshot shows a terminal window titled "Services" with the AWS logo. The session is named "SohamSatputra" and is located in "Mumbai". The terminal output is as follows:

```
ubuntu@maste-node:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=all  
  
[init] Using Kubernetes version: v1.31.0  
[preflight] Running pre-flight checks  
    [WARNING FileExisting-socat]: socat not found in system path  
[preflight] Pulling images required for setting up a Kubernetes cluster  
[preflight] This might take a minute or two, depending on the speed of your internet connection  
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'  
W0924 17:41:13.236064    4176 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 [kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local master-node] and IPs [10.96.0.1 172.31.32.117]  
[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 [localhost master-node] and IPs [172.31.32.117 127.0.0.1 ::1]
```

Set up kubectl on the master node:

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

```
kubeadm join 172.31.32.117:6443 --token t2jppj2.rauz0s7fimwpdo4a --discovery-token-ca-cert-hash sha256:1fa6fa408d342aef675f4bab47ff1c02da288b9cf34dd1cff8161d84586cd50b
```



The screenshot shows the AWS CloudShell interface with the AWS logo and Services navigation bar. The user is in the 'Mumbai' region and signed in as 'SohamSatpute'. The terminal window displays the output of the command 'kubectl get pods --all-namespaces'. The output lists various Kubernetes components across different namespaces, including kube-flannel, kube-system, and master nodes, all in a 'Running' state.

NAMESPACE	NAME	READY	STATUS
RESTARTS	AGE		
kube-flannel	kube-flannel-ds-qmgdh	1/1	Running
0	3m47s		
kube-system	coredns-7c65d6cfc9-42w2x	1/1	Running
0	9m56s		
kube-system	coredns-7c65d6cfc9-8ctb6	1/1	Running
0	9m56s		
kube-system	etcd-maste-node	1/1	Running
0	10m		
kube-system	kube-apiserver-maste-node	1/1	Running
0	10m		
kube-system	kube-controller-manager-maste-node	1/1	Running
0	10m		
kube-system	kube-proxy-5g6gj	0/1	CrashLoopBackOff
7 (78s ago)	9m57s		
kube-system	kube-scheduler-maste-node	1/1	Running
0	10m		

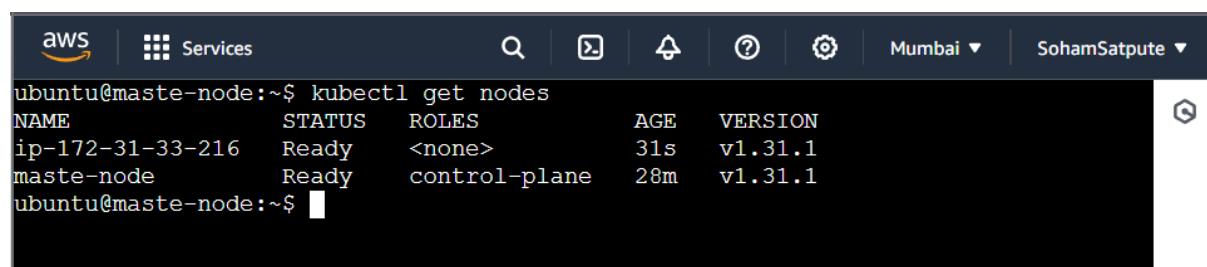
Step 7:

Join Worker Nodes to the Cluster On the worker nodes, run the command provided by the master node during initialization:

```
ubuntu@maste-node:~$ sudo kubeadm join 172.31.32.117:6443 --token t2jppj2.rauz0s7fimwpdo4a --discovery-token-ca-cert-hash sha256:288b9cf34dd1cff8161d84586cd50b
```

Step 8:

Verify the Cluster Once the worker node joins, check the status on the master node



The screenshot shows the AWS CloudShell interface with the AWS logo and Services navigation bar. The user is in the 'Mumbai' region and signed in as 'SohamSatpute'. The terminal window displays the output of the command 'kubectl get nodes'. It shows two nodes: 'ip-172-31-33-216' and 'maste-node', both in a 'Ready' state and assigned to the 'control-plane' role. The version of the kernel is listed as v1.31.1.

NAME	STATUS	ROLES	AGE	VERSION
ip-172-31-33-216	Ready	<none>	31s	v1.31.1
maste-node	Ready	control-plane	28m	v1.31.1

ADVANCE DEVOPS EXP 4

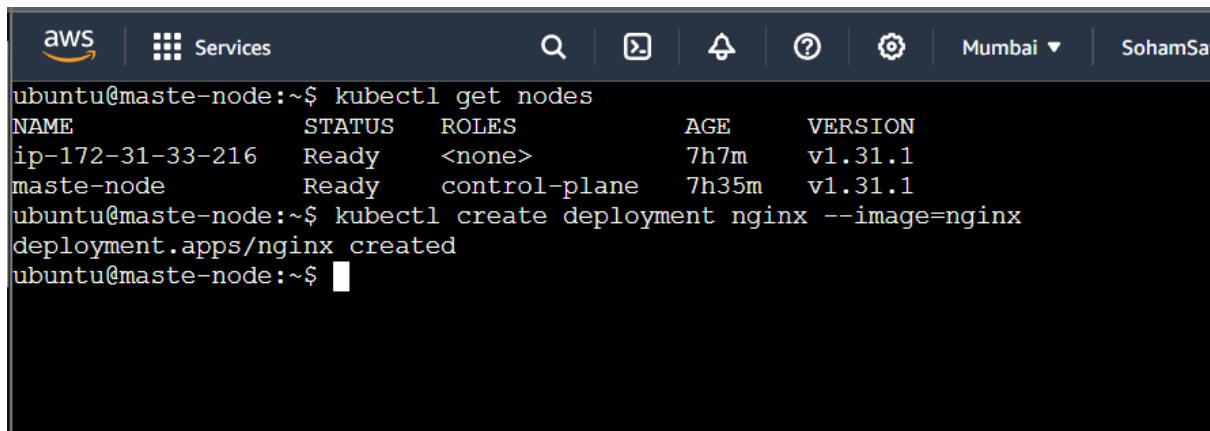
Name :- Soham Satpute

Roll no :- 52

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

Step 1: As 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.

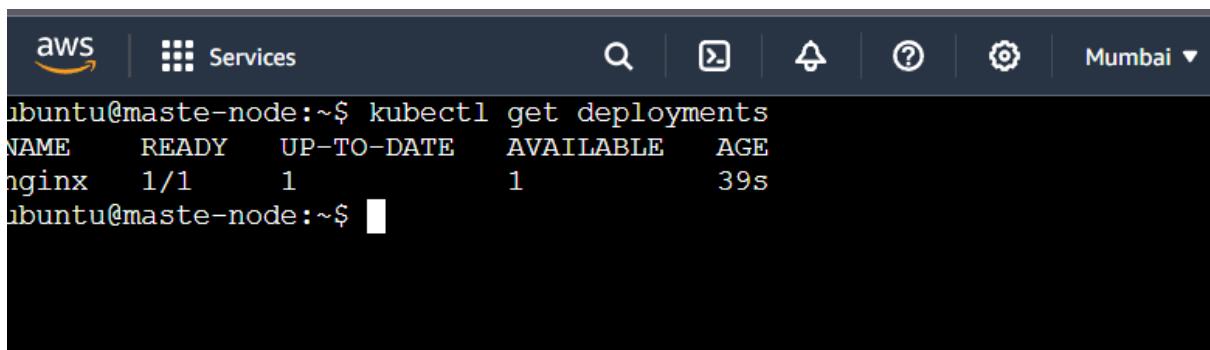
```
$ kubectl create deployment nginx --image=nginx
```



The screenshot shows a terminal window within the AWS CloudShell interface. The terminal title is "ubuntu@maste-node:~\$". The user has run the command "kubectl get nodes" which lists two nodes: "ip-172-31-33-216" and "maste-node", both in a "Ready" state. The user then runs "kubectl create deployment nginx --image=nginx" which creates a deployment named "nginx" in the "deployment.apps" namespace. The terminal ends with "ubuntu@maste-node:~\$".

Step 2: Verify the deployment using the command:

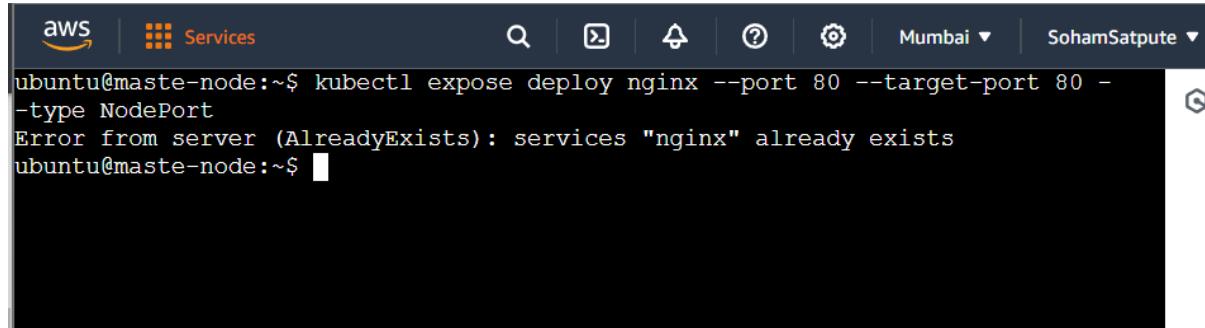
```
$ kubectl get deployments
```



The screenshot shows a terminal window within the AWS CloudShell interface. The terminal title is "ubuntu@maste-node:~\$". The user has run the command "kubectl get deployments" which lists a single deployment named "nginx". The deployment is in a "READY" state with 1/1 pods, "UP-TO-DATE" with 1 pod, and "AVAILABLE" with 1 pod. It was created 39 seconds ago. The terminal ends with "ubuntu@maste-node:~\$".

Step 3: Next, run the following command to create a service named nginx that will expose the app publicly. It will do so through a NodePort, a scheme that will make the pod accessible through an arbitrary port opened on each node of the cluster with this service-type, Kubernetes will assign this service on ports on the **30000+** range.

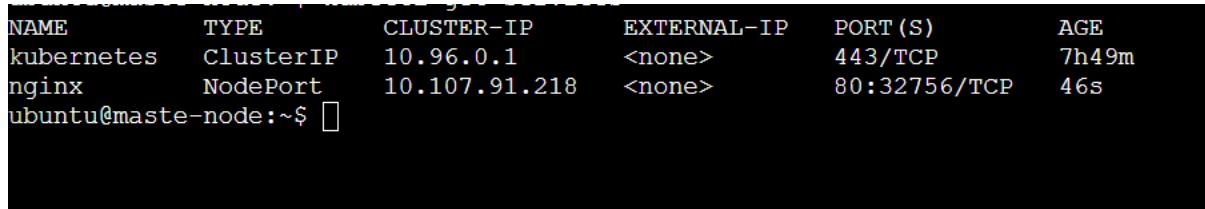
```
$kubectl expose deploy nginx --port 80 --target-port 80 --type NodePort
```



A screenshot of a terminal window titled "aws Services" with a user "SohamSatpute" and location "Mumbai". The terminal shows the command \$kubectl expose deploy nginx --port 80 --target-port 80 --type NodePort being run, followed by an error message: "Error from server (AlreadyExists): services \"nginx\" already exists".

Step 4: Run this command to see a summary of the service and the ports exposed.

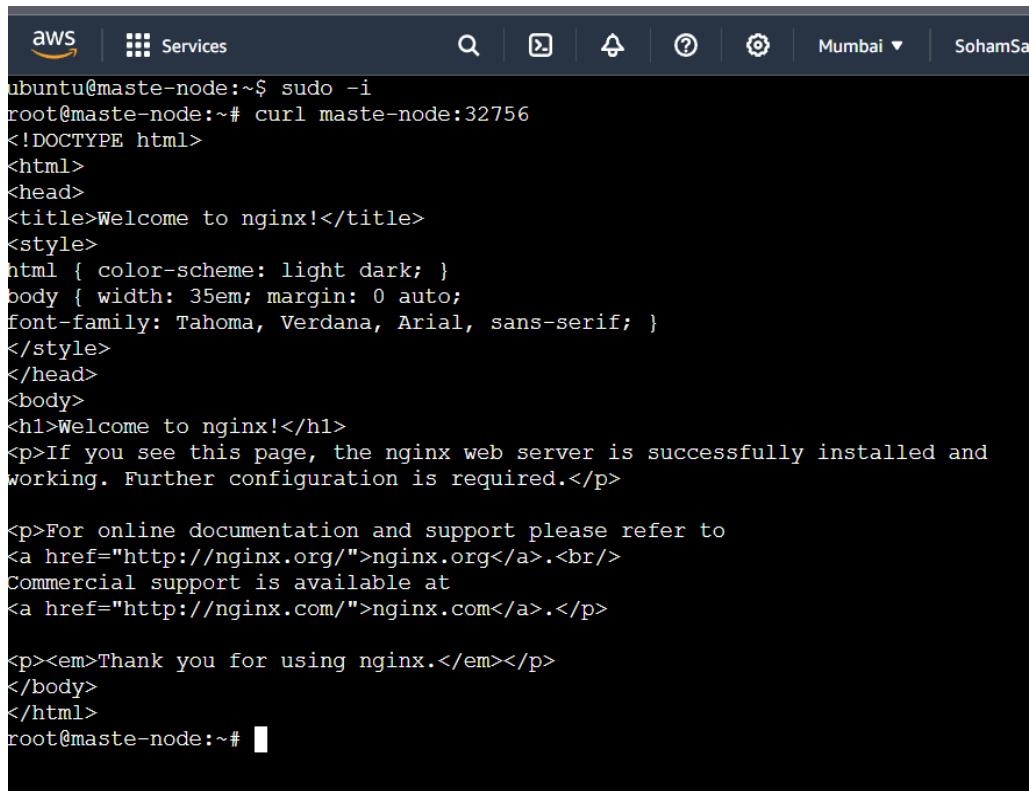
```
$kubectl get services
```



NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	7h49m
nginx	NodePort	10.107.91.218	<none>	80:32756/TCP	46s

Step 5: Add the port which is displayed i.e. 32756(in our case) in the inbound rules of the security group.

Step 6: Now you can verify that the Nginx page is reachable on all nodes using the curl command. As you can see, the “WELCOME TO NGINX!” page can be reached.

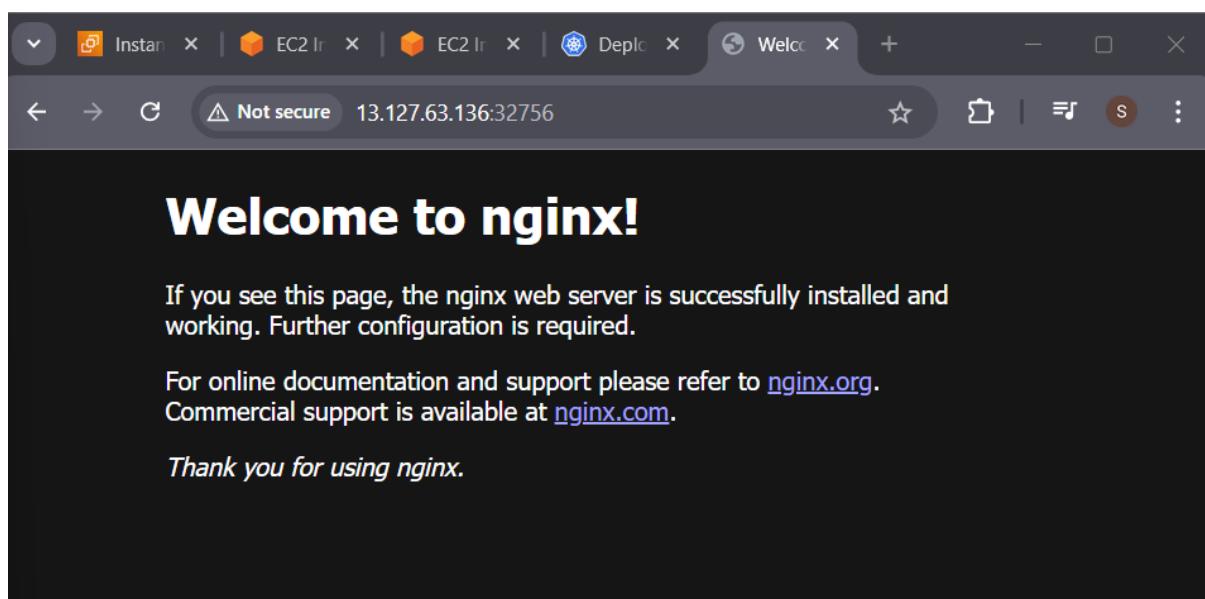


```
ubuntu@maste-node:~$ sudo -i
root@maste-node:~# curl maste-node:32756
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
<p><em>Thank you for using nginx.</em></p>
</body>
</html>
root@maste-node:~#
```

Step 7: To test that everything is working, visit http://worker_1_ip:nginx_port or http://worker_2_ip:nginx_port

through a browser on your local machine. You will see Nginx’s familiar welcome page.

<http://13.127.63.136:32756/>



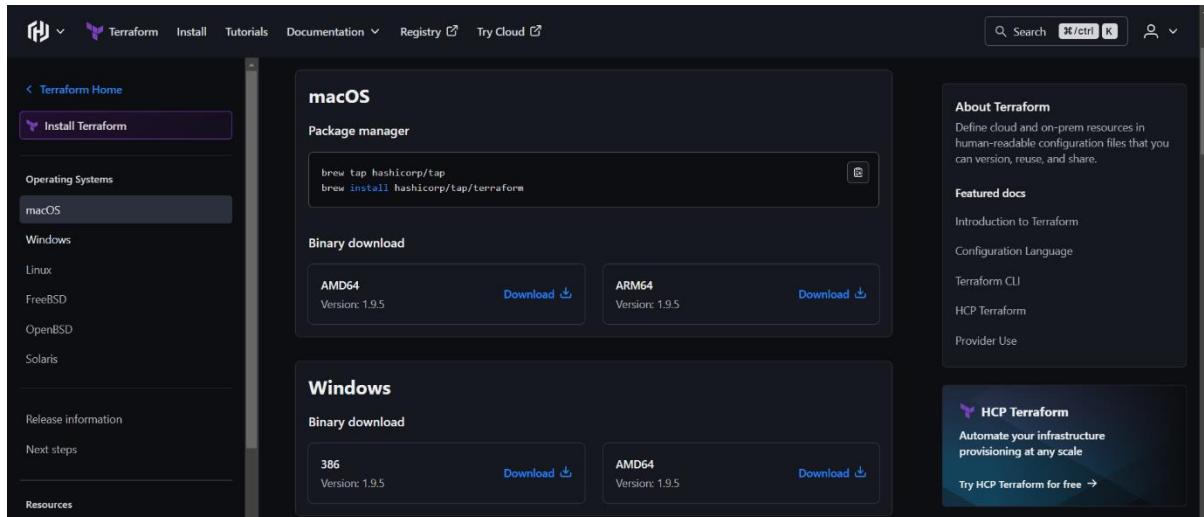
Experiment no 5

Name :Soham Satpute

Roll no :- 52/D15A

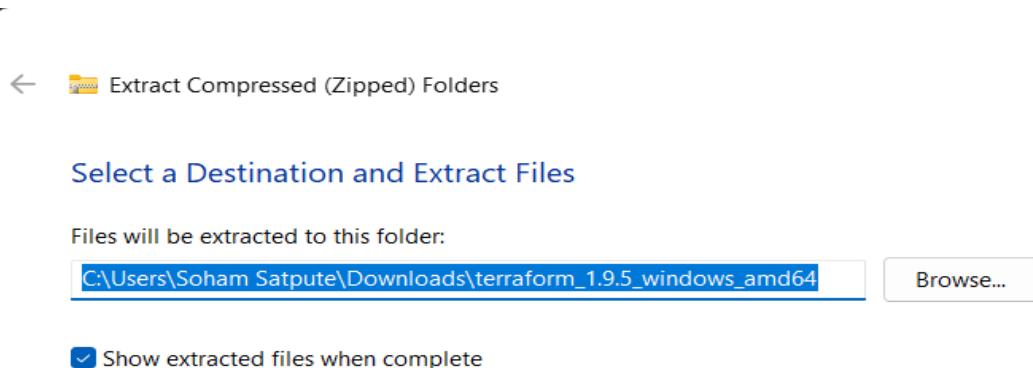
Aim :-Installation and Configuration of Terraform in Windows

Installation for Windows :-



Step1: In your Downloads, right-click on the downloaded Terraform binary file and select "**Extract All**" .

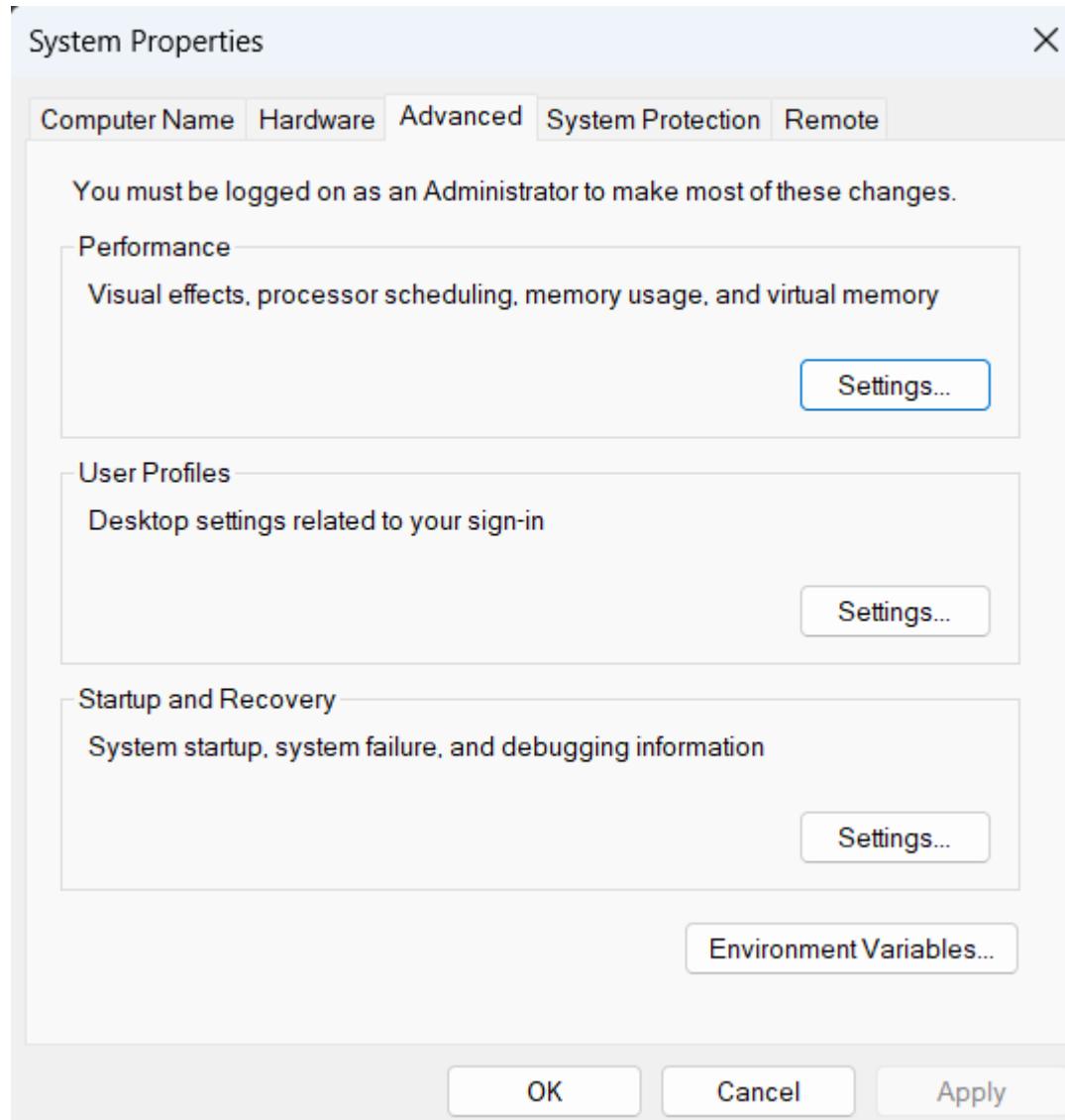
The following window will pop up:



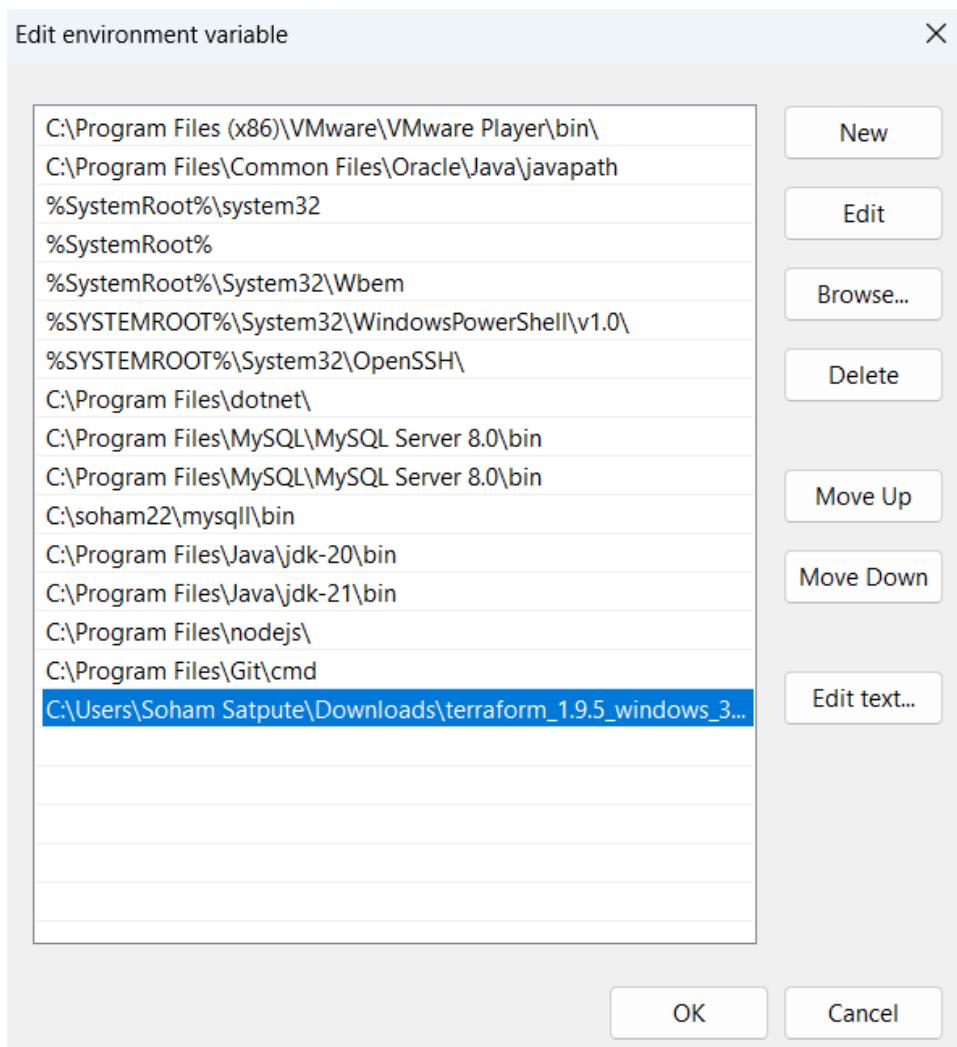
Step 2 :

Set Terraform Path to System Environment Variables

Click on the “**Environment variables**” in the system properties:



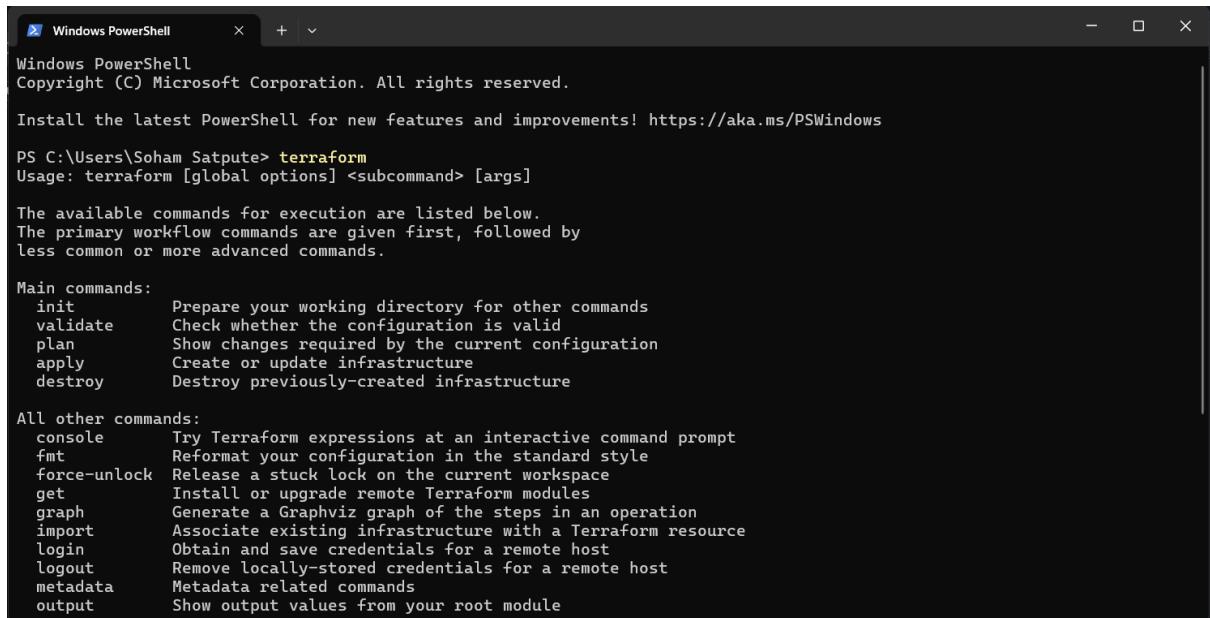
Step 3: Set the path



Step 4: Navigate to the folder path C:\terraform in a new command prompt window and type the terraform -version to verify the installed version.

The screenshot shows a Microsoft Windows Command Prompt window. The title bar says 'Command Prompt'. The window displays the following text:
Microsoft Windows [Version 10.0.22621.4037]
(c) Microsoft Corporation. All rights reserved.
C:\Users\Soham Satpute>terraform -version
Terraform v1.9.5
on windows_386
C:\Users\Soham Satpute>

Step 5: To see more Terraform commands, you can simply type `terraform` in the terminal.



```
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\Soham Satpute> 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
```

Experiment no 6

Name: Soham Satpute

Roll_No :52

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 Prerequisite:

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

Step 1:

Check Docker functionality:

```
c:\Users\Soham Satpute> docker
usage: docker [OPTIONS] COMMAND
A self-sufficient runtime for containers

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

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

Check for the docker version with the following command:

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

```
C:\Users\Soham Satpute>
```

Now, 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: Firstly create a new folder named ‘Docker’ in the ‘TerraformScripts’ folder. Then create a new docker.tf file using Atom editor and write the following contents into it to create a Ubuntu Linux container.

Script:

```
terraform {  
  
required_providers {  
  
docker = {  
  
source = "kreuzwerker/docker"  
version = "2.25.0"  
  
}  
}  
}  
  
provider "docker" {  
  
host = "npipe:///./pipe/docker_engine"  
}  
  
resource "docker_image" "ubuntu" {  
name = "ubuntu:latest"  
}  
  
resource "docker_container" "foo" {
```

```

image = docker_image.ubuntu.image_id
name = "foo"
command = ["sleep", "3600"]
}

```

```

1  terraform {
2    required_providers {
3      docker = {
4        source  = "kreuzwerker/docker"
5        version = "2.25.0"
6      }
7    }
8  }
9
10 provider "docker" {
11   host = "npipe://./pipe/docker_engine"
12 }
13
14 resource "docker_image" "ubuntu" {
15   name = "ubuntu:latest"
16 }
17
18 resource "docker_container" "foo" {
19   image     = docker_image.ubuntu.image_id
20   name      = "foo"
21   command   = ["sleep", "3600"]
22 }
23

```

Step 3:

Execute Terraform Init command to initialize the resources:

```

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

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

PS C:\soham22\Terraform Scripts\Docker> terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of kreuzwerker/docker from the dependency lock file
- Using previously-installed kreuzwerker/docker v2.25.0

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.
PS C:\soham22\Terraform Scripts>Docker>

```

Step 4:

Execute Terraform plan to see the available resources:

```
PS C:\soham22\Terraform Scripts\Docker> terraform plan
docker_image.ubuntu: Refreshing state... [id=sha256:b1e9cef3f2977f8bdd19eb9ae04f83b315f80fe4f5c5651fedf41482c12432f7ubuntu:latest]
docker_container.foo: Refreshing state... [id=0200e3c246ed3069424b55d2bb9803da5442e04190c918629f4ee6a8879218fa]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
-/+ destroy and then create replacement

Terraform will perform the following actions:

# docker_container.foo must be replaced
-/+ resource "docker_container" "foo" {
    + bridge
    + container_logs
    - cpu_shares
    - dns
    - dns_opts
    - dns_search
    + entrypoint
    - env
    + exit_code
    - gateway
    - group_add
    - hostname
    - id
-> (known after apply)
    - init
    - ip_address

```

Step 5 :

Type terraform apply to apply changes:

```
PS C:\soham22\Terraform Scripts\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
    + bridge
    + command
        + "sleep",
        + "3600",
    ]
    + container_logs
    + container_read_refresh_timeout_milliseconds = 15000
    + entrypoint
    + env
    + exit_code
    + gateway
    + hostname
    + id
    + image
    + init
    + ip_address

```

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

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

Enter a value: yes

docker_image.ubuntu: Creating...
docker_image.ubuntu: Still creating... [10s elapsed]
docker_image.ubuntu: Still creating... [20s elapsed]
docker_image.ubuntu: Still creating... [30s elapsed]
```

Docker images , Before Executing Apply

```
PS C:\soham22\Terraform Scripts\Docker> docker images
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
```

Docker images , After Executing Apply

```
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
PS C:\soham22\Terraform Scripts\Docker> docker images
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
ubuntu          latest       b1e9cef3f297  3 weeks ago   78.1MB
nginx           latest       39286ab8a5e1  5 weeks ago   188MB
hello-world     latest       d2c94e258dcb  16 months ago  13.3kB
PS C:\soham22\Terraform Scripts\Docker>
```

Step 6:

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

```
PS C:\soham22\Terraform Scripts\Docker> terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:b1e9cef3f297f8bdd19eb9ae04f83b315f80fe4f5c5651fedf41482c12432f7ubuntu:latest]
docker_container.foo: Refreshing state... [id=877e2e0bb5cf00634ddc5cde0f73121bc963959cb5c8f1ddf827471f515e4d5]

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
    - container_read_refresh_timeout_milliseconds = 15000 -> 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                            = "877e2e0bb5cf" -> null
    - id                                  = "877e2e0bb5cf00634ddc5cde0f73121bc963959cb5c8f1ddf827471f515e4d5" -> null
    - image                               = "sha256:b1e9cef3f297f8bdd19eb9ae04f83b315f80fe4f5c5651fedf41482c12432f7" -> null
}
```

Docker images After Executing Destroy step:

```
Destroy complete! Resources: 2 destroyed.
PS C:\soham22\Terraform Scripts\Docker> docker images
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
nginx           latest       39286ab8a5e1  5 weeks ago   188MB
hello-world     latest       d2c94e258dcb  16 months ago  13.3kB
```

ADVANCE DEVOPS EXP 7

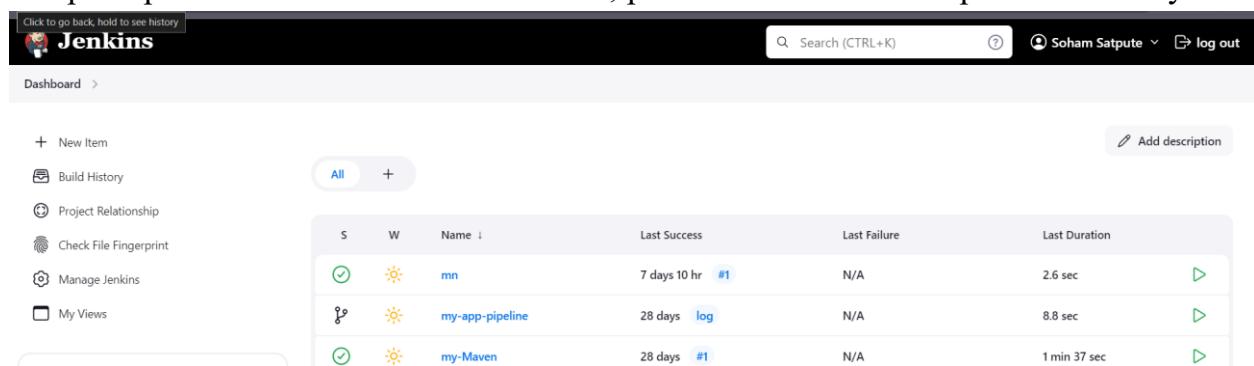
Name: Soham Satpute
Class:D15A
Roll No:52

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

Jenkins integration with SonarQube:

Steps:

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



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

- Header:** Click to go back, hold to see history, Jenkins logo, Search (CTRL+K), Soham Satpute, log out.
- Breadcrumbs:** Dashboard >
- Left sidebar:** + New Item, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins, My Views.
- Toolbar:** All, +, Add description.
- Table:** A list of projects with columns: S, W, Name, Last Success, Last Failure, Last Duration.

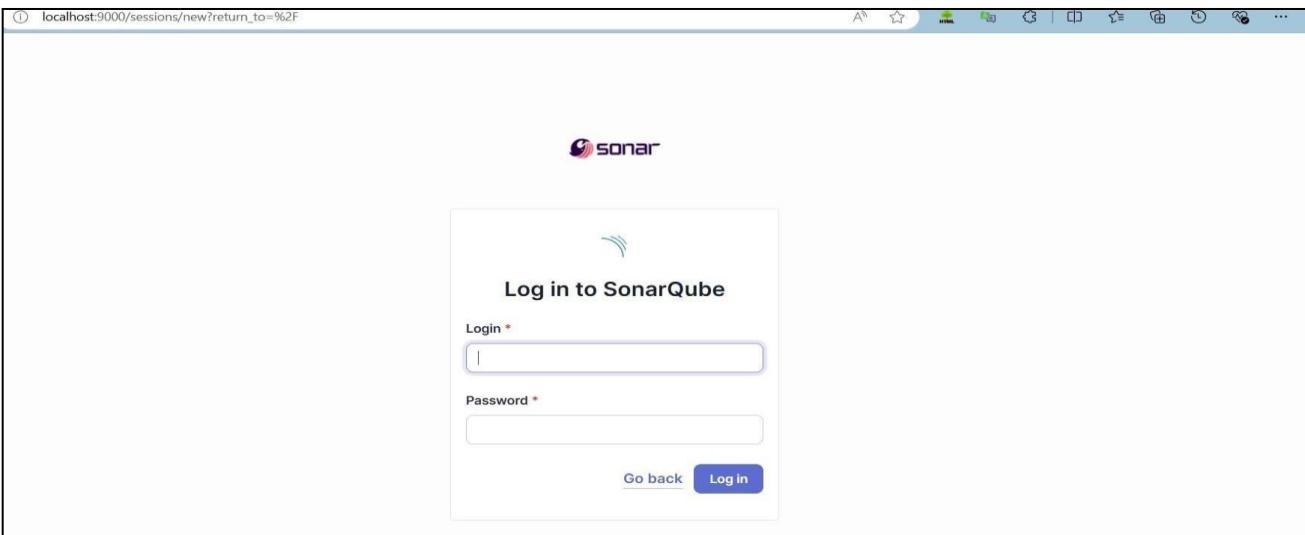
S	W	Name	Last Success	Last Failure	Last Duration
Green checkmark	Sun icon	mn	7 days 10 hr #1	N/A	2.6 sec
Yellow question mark	Sun icon	my-app-pipeline	28 days log	N/A	8.8 sec
Green checkmark	Sun icon	my-Maven	28 days #1	N/A	1 min 37 sec

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

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

```
PS C:\Windows\system32> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest  
Unable to find image 'sonarqube:latest' locally  
latest: Pulling from library/sonarqube  
7478e0ac0f23: Pull complete  
90a925ab929a: Pull complete  
7d9a34308537: Pull complete  
80338217a4ab: Pull complete  
1afdf5c7e184: Pull complete  
7b87d6fa783d: Pull complete  
bd819c9b5ead: Pull complete  
4f4fb700ef54: Pull complete  
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde  
Status: Downloaded newer image for sonarqube:latest  
5ab3928e5e27607e3661d129731e4e600a9019574c7dc2767aa9b3bfdaa941be
```

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

The screenshot shows the SonarQube interface for creating a local project. The top navigation bar includes tabs for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, More, and a search icon. The main content area displays the second step of a two-step process: "Set up project for Clean as You Code". A sub-instruction states: "The new code definition sets which part of your code will be considered new code. This helps you focus attention on to follow the Clean as You Code methodology. Learn more: [Defining New Code](#)". Below this, a section titled "Choose the baseline for new code for this project" contains two options: "Use the global setting" (selected) and "Define a specific setting for this project". The "Use the global setting" option is described as "Any code that has changed since the previous version is considered new code. Recommended for projects following regular versions or releases." The "Define a specific setting for this project" option is currently unselected.

Setup the project and come back to Jenkins Dashboard.

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

The screenshot shows the Jenkins Plugins page. The left sidebar lists "Updates" (25), "Available plugins" (selected), "Installed plugins", and "Advanced settings". The main content area shows a search bar with "sonard" and an "Install" button. A table lists the "SonarQube Scanner" plugin, version 2.17.2, released 6 months and 29 days ago. The plugin description states: "This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality." The "Available plugins" tab is highlighted.

The screenshot shows the Jenkins Plugins page with the "Download progress" tab selected. The left sidebar includes "Updates" (25), "Available plugins" (selected), "Installed plugins", and "Advanced settings". The main content area displays the "Download progress" section, which includes a "Preparation" list with three items: "Checking internet connectivity", "Checking update center connectivity", and "Success". It also shows the "SonarQube Scanner" plugin with a green checkmark and the status "Success". Below this, there are links to "Go back to the top page" and "Restart Jenkins when installation is complete and no jobs are running".

6.Under Jenkins ‘Manage Jenkins’ then go to ‘system’, scroll and look for **SonarQube Servers** and enter the details.

Enter the Server Authentication token if needed.

In SonarQube installations: Under **Name** add <project name of sonarqube>,here we have named it as **adv_devops_7_sonarqube** In **Server URL** Default is <http://localhost:9000>

The screenshot shows the Jenkins configuration interface for SonarQube installations. The navigation path is: Dashboard > Manage Jenkins > System > Environment variables. A sub-section titled "SonarQube installations" is selected. The configuration form includes fields for "Name" (set to "sonarqube"), "Server URL" (set to "http://localhost:9000"), and "Server authentication token" (set to "- none -"). There is also an "Advanced" dropdown menu. At the bottom are "Save" and "Apply" buttons.

Dashboard > Manage Jenkins > System >

Environment variables

SonarQube installations

List of SonarQube installations

Name

sonarqube

Server URL

Default is http://localhost:9000

http://localhost:9000

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

- none -

+ Add ▾

Advanced ▾

Save Apply

7. Search for SonarQube Scanner under Global Tool Configuration.

Choose the latest configuration and choose Install automatically.

Dashboard > Manage Jenkins > Tools

The screenshot shows the Jenkins 'Tools' configuration page. At the top, there is a breadcrumb navigation: Dashboard > Manage Jenkins > Tools. Below the navigation, the title is 'SonarQube Scanner installations ^' with an 'Edited' status indicator. A large 'Add SonarQube Scanner' button is visible. The main configuration section is titled 'SonarQube Scanner'. It includes a 'Name' field containing 'SonarQube' and a checked 'Install automatically' checkbox. A sub-section titled 'Install from Maven Central' shows the selected version 'SonarQube Scanner 6.2.0.4584'. There is also an 'Add Installer' dropdown menu. At the bottom of the configuration area, there are 'Save' and 'Apply' buttons.

SonarQube Scanner installations ^ Edited

Add SonarQube Scanner

SonarQube Scanner

Name
SonarQube

Install automatically ?

Install from Maven Central

Version
SonarQube Scanner 6.2.0.4584

Add Installer ▾

Add SonarQube Scanner

Save Apply

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

Dashboard > All > New Item

New Item

Enter an item name
SonarQube

Select an item type

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

OK

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, just to test the integration.

The screenshot shows the 'Source Code Management' configuration screen. The 'Git' option is selected. In the 'Repositories' section, a single repository is listed with its URL set to https://github.com/shazforiot/MSBuild_firstproject.git. The 'Credentials' dropdown is set to '- none -'. There is also an 'Advanced' section and a 'Add Repository' button at the bottom.

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

The screenshot shows the 'Configure' screen with the 'Build Environment' tab selected. Under 'Build Steps', the 'Execute SonarQube Scanner' step is highlighted. The 'Post-build Actions' section is visible at the bottom.

JDK ?

JDK to be used for this SonarQube analysis

(Inherit From Job)

Path to project properties ?**Analysis properties** ?

```
sonar.projectKey=Sonargube-test  
sonar.sources=.  
sonar.host.url=http://localhost:9000  
sonar.login=admin  
sonar.password=
```

Additional arguments ?**JVM Options** ?

Add build step ▾

Save

Apply

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

The screenshot shows the SonarQube Administration interface, specifically the Groups section. It lists several groups with their permissions:

- sonar-administrators**: System administrators. Has checked boxes for "Administer System", "Administrator", and "Execute Analysis". Has checked checkboxes for "Quality Gates" and "Quality Profiles". Has an unchecked checkbox for "Projects".
- sonar-users**: Every authenticated user automatically belongs to this group. Has an unchecked checkbox for "Administer System". Has checked checkboxes for "Quality Gates" and "Quality Profiles". Has checked checkboxes for "Administrator" and "Execute Analysis". Has checked checkboxes for "Projects".
- Anyone DEPRECATED**: Anybody who browses the application belongs to this group. If authentication is not enforced, assigned permissions also apply to non-authenticated users. Has an unchecked checkbox for "Administer System". Has checked checkboxes for "Quality Gates" and "Quality Profiles". Has an unchecked checkbox for "Administrator". Has an unchecked checkbox for "Execute Analysis". Has an unchecked checkbox for "Projects".
- Administrator admin**: Has checked checkboxes for "Administer System", "Administrator", and "Execute Analysis". Has checked checkboxes for "Quality Gates" and "Quality Profiles". Has an unchecked checkbox for "Projects".

At the bottom, it says "4 of 4 shown".

12. Check the console Output

The screenshot shows the Jenkins Console Output page for a build named "SonarQube". The left sidebar shows build steps like Status, Changes, Console Output (which is selected), Edit Build Information, Delete build #3, Timings, Git Build Data, and Previous build. The main area displays the build log:

```

Started by user Soham Satpute
Running as SYSTEM
Building on the built-in node in workspace C:\ProgramData\Jenkins\jenkins\workspace\SonarQube
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\jenkins\workspace\SonarQube\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/shazforiot/MSBuild_FirstProject # timeout=10
Fetching upstream changes from https://github.com/shazforiot/MSBuild_FirstProject
> git.exe --version # timeout=10
> git --version # 'git version 2.46.0.windows.1'
> git --version # 'git version 2.46.0.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/shazforiot/MSBuild_FirstProject +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse --refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision f2bc042c04ce672427c380bcae6d6fee7b49adf (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f f2bc042c04ce672427c380bcae6d6fee7b49adf # timeout=10
Commit message: "updated"
> git.exe rev-list --no-walk f2bc042c04ce672427c380bcae6d6fee7b49adf # timeout=10
Injecting SonarQube environment variables using the configuration: sonarqube
[SonarQube] $ C:\ProgramData\Jenkins\jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\SonarQube\bin\sonar-scanner.bat -
-Dsonar.host.url=http://localhost:9000 -Dsonar.projectKey=SonarQube-test -Dsonar.host.url=http://localhost:9000 -Dsonar.login=admin -Dsonar.sources=.
-Dsonar.password=Donon@33 -Dsonar.projectBaseDir=C:\ProgramData\Jenkins\jenkins\workspace\SonarQube
15:15:05.681 WARN Property 'sonar.host.url' with value 'http://localhost:9000' is overridden with value 'http://localhost:9000'
15:15:05.755 INFO Scanner configuration file:

```

```

15:15:34.677 INFO Sensor Analysis Warnings import [csharp]
15:15:34.683 INFO Sensor Analysis Warnings import [csharp] (done) | time=0ms
15:15:34.683 INFO Sensor C# File Caching Sensor [csharp]
15:15:34.683 WARN Incremental PR analysis: Could not determine common base path, cache will not be computed. Consider setting 'sonar.projectBaseDir' property.
15:15:34.683 INFO Sensor C# File Caching Sensor [csharp] (done) | time=6ms
15:15:34.683 INFO Sensor Zero Coverage Sensor
15:15:34.686 INFO Sensor Zero Coverage Sensor (done) | time=3ms
15:15:34.695 INFO SCM Publisher SCM provider for this project is: git
15:15:34.696 INFO SCM Publisher 4 source files to be analyzed
15:15:35.179 INFO SCM Publisher 4/4 source files have been analyzed (done) | time=483ms
15:15:35.182 INFO CPD Executor Calculating CPD for 0 files
15:15:35.183 INFO CPD Executor CPD calculation finished (done) | time=0ms
15:15:35.189 INFO SCM revision ID 'f2bc042x04c6e72427c380bcae6d6fee7b49adf'
15:15:35.436 INFO Analysis report generated in 103ms, dir size=201.0 kB
15:15:35.471 INFO Analysis report compressed in 30ms, zip size=22.3 kB
15:15:35.678 INFO Analysis report uploaded in 204ms
15:15:35.681 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=Sonarqube-test
15:15:35.681 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
15:15:35.682 INFO More about the report processing at http://localhost:9000/api/ce/task?id=7cbcdff-b0aa-482a-8fa3-fbd1b093cbe6
15:15:35.705 INFO Analysis total time: 24.743 s
15:15:35.708 INFO SonarScanner Engine completed successfully
15:15:35.766 INFO EXECUTION SUCCESS
15:15:35.767 INFO Total time: 30.074s
Finished: SUCCESS

```

REST API Jenkins 2.462.2



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

Sonarqube

Projects Issues Rules Quality Profiles Quality Gates Administration More Q

Sonarqube-test / main ?

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

main

Version not provided - Set as homepage

Quality Gate: **Passed** Last analysis 2 minutes ago

The last analysis has warnings. See details

New Code Overall Code

Security	Reliability	Maintainability
0 Open issues (A)	0 Open issues (A)	0 Open issues (A)
0 H 0 M 0 L	0 H 0 M 0 L	0 H 0 M 0 L

Accepted issues	Coverage	Duplications
0	(S)	0.0%

81°F Cloudy 15:17 26-09-2024

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

ADVANCE DEVOPS EXP 8

Name:Soham Satpute

Class:D15A Roll

No:52

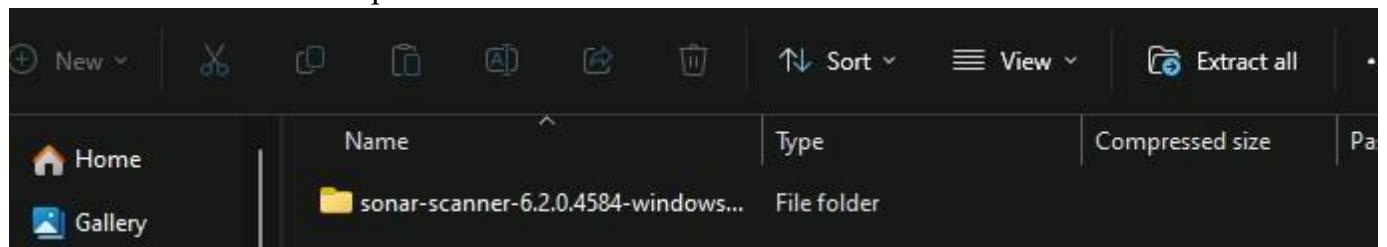
Aim: Create a Jenkins CICD Pipeline with SonarQube / GitLab Integration to perform a static analysis of the code to detect bugs, code smells, and security vulnerabilities on a sample Web / Java / Python application.

Step 1: Download sonar scanner <https://docs.sonarsource.com/sonarqube/latest/analyzing-source-code/scanners/sonarscan>

The screenshot shows a web browser displaying the SonarScanner CLI documentation. The URL in the address bar is <https://docs.sonarsource.com/sonarqube/latest/analyzing-source-code/scanners/sonarscan>. The page title is "SonarScanner CLI". On the left, there's a sidebar with navigation links like "Homepage", "Try out SonarQube", "Server installation and setup", "Analyzing source code" (which is expanded to show "SonarQube analysis overview" and "Project analysis setup"), "Scanners" (which is expanded to show "Scanner environment", "SonarScanner CLI", "SonarQube extension for Azure DevOps", "SonarQube extension for Jenkins", "SonarScanner for .NET", and "SonarScanner for Maven"), and "Docs 10.6". The main content area features a card for version 6.1, released on 2024-06-27, which supports macOS and Linux AArch64 distributions. It provides download links for Linux x64, Linux AArch64, Windows x64, macOS x64, macOS AArch64, Docker Any (Requires a pre-installed JVM), and Release notes. Below the card, there are two paragraphs of text about the SonarScanner CLI.

ner/ Visit this link and download the sonarqube scanner CLI.

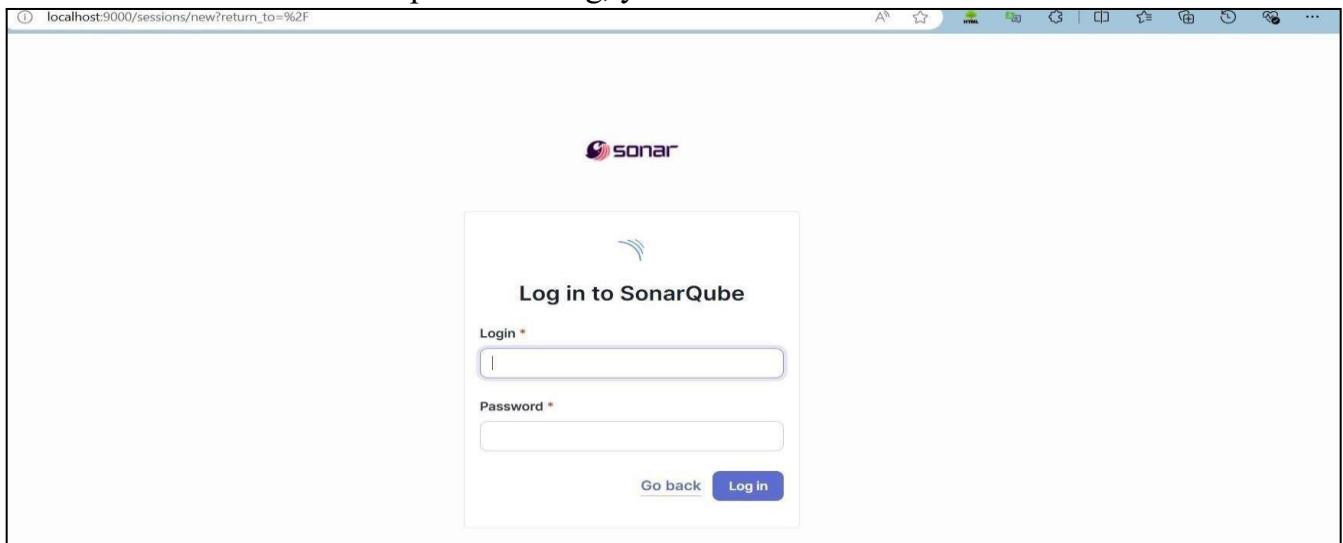
Extract the downloaded zip file in a folder.



1. Install sonarqube image Command: **docker pull sonarqube**

```
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindc
PS C:\Users\Soham Satpute> docker pull sonarqube
Using default tag: latest
latest: Pulling from library/sonarqube
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Image is up to date for sonarqube:latest
docker.io/library/sonarqube:latest
```

2. Once the container is up and running, you can check the status of



SonarQube at localhost port 9000.

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

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration More Q

How do you want to create your project?

Do you want to benefit from all of SonarQube's features (like repository import and Pull Request decoration)? Create your project from your favorite DevOps platform.

First, you need to set up a DevOps platform configuration.

Import from Azure DevOps Setup Import from Bitbucket Cloud Setup Import from Bitbucket Server Setup

Import from GitHub Setup Import from GitLab Setup

Are you just testing or have an advanced use-case? Create a local project.

Create a local project

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

1 of 2

Create a local project

Project display name *

Sonarqube-test



Project key *

Sonarqube-test



Main branch name *

main

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

2 of 2

Set up project for Clean as You Code

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

Choose the baseline for new code for this project

Use the global setting

Previous version

Any code that has changed since the previous version is considered new code.

Recommended for projects following regular versions or releases.

Define a specific setting for this project

Previous version

Any code that has changed since the previous version is considered new code.

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

The screenshot shows the Jenkins dashboard with the following details:

- Left sidebar:** Includes links for New Item, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins, and My Views.
- Top right:** Search bar, user info (Soham Satpute), and log out button.
- Main area:** A table listing three projects:

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀️	mn	7 days 10 hr #1	N/A	2.6 sec
⌚	☀️	my-app-pipeline	28 days log	N/A	8.8 sec
✓	☀️	my-Maven	28 days #1	N/A	1 min 37 sec

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

The screenshot shows the Jenkins Manage Jenkins > Plugins page with the following details:

- Left sidebar:** Updates (25), Available plugins (selected), Installed plugins, Advanced settings.
- Search bar:** sonarq
- Available plugins section:**

Install	Name	Released
<input type="checkbox"/>	SonarQube Scanner 2.17.2	6 mo 29 days ago
	External Site/Tool Integrations	
	Build Reports	

This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.

The screenshot shows the Jenkins Manage Jenkins > Plugins page with the following details:

- Left sidebar:** Updates (25), Available plugins (selected), Installed plugins, Advanced settings, Download progress (selected).
- Right panel:**

Download progress

Preparation

 - Checking internet connectivity
 - Checking update center connectivity
 - Success

SonarQube Scanner ✓ Success

Loading plugin extensions ✓ Success

[Go back to the top page](#)
(you can start using the installed plugins right away)

[Restart Jenkins when installation is complete and no jobs are running](#)

7.Under Jenkins ‘Manage Jenkins’ then go to ‘system’, scroll and look for **SonarQube Servers** and enter the details.

Enter the Server Authentication token if needed.

In SonarQube installations: Under **Name** add <project name of sonarqube> for me **adv_devops_7_sonarqube**

In **Server URL** Default is <http://localhost:9000>

8. Search for SonarQube Scanner under Global Tool Configuration.

Name

Server URL

Default is <http://localhost:9000>

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

- none -

+ Add ▾

Advanced ▾

Choose the latest configuration and choose Install automatically.

Dashboard > Manage Jenkins > Tools

Dashboard > Manage Jenkins > Tools

Add Git ▾

Gradle installations

Add Gradle

SonarScanner for MSBuild installations

Add SonarScanner for MSBuild

SonarQube Scanner installations

Add SonarQube Scanner

Ant installations

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

SonarQube Scanner installations ^ Edited

Add SonarQube Scanner

SonarQube Scanner

Name
SonarQube

Install automatically ?

Install from Maven Central

Version
SonarQube Scanner 6.2.0.4584

Add Installer ▾

Add SonarQube Scanner

Save **Apply**

9. After configuration, create a New Item → choose a pipeline project.

Dashboard > All > New Item

New Item

Enter an item name
AdDevops-8

Select an item type

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

OK

10. Under Pipeline script, enter the following:

```

node {

stage('Cloning the GitHub Repo') { git
  'https://github.com/shazforiot/GOL.git'
} stage('SonarQube

analysis') {

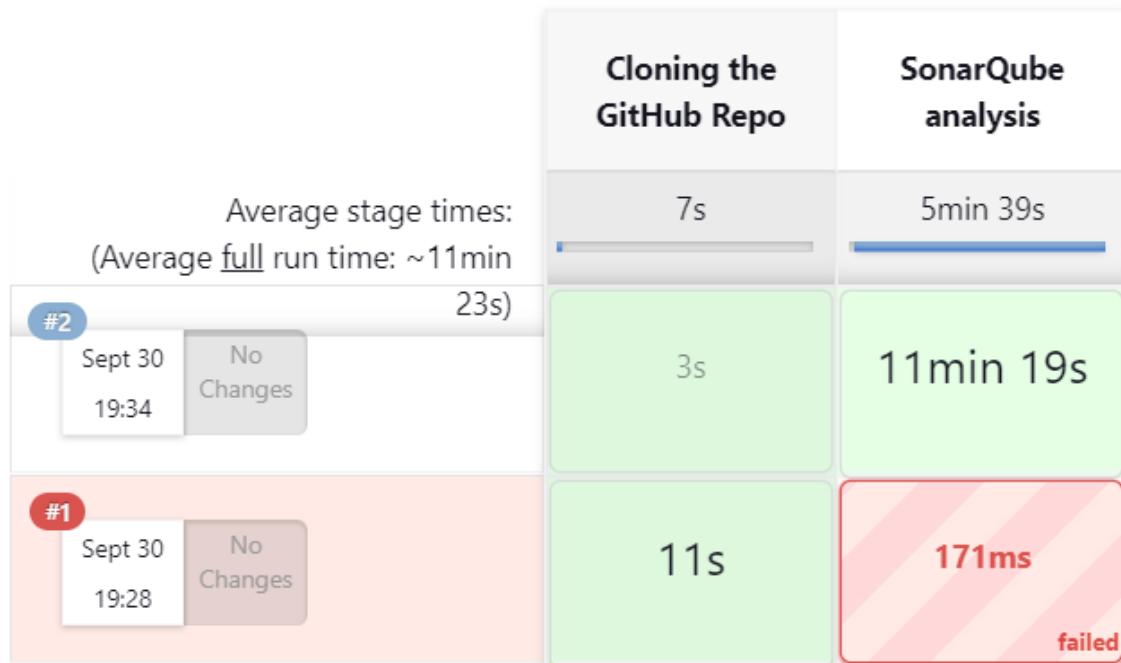
withSonarQubeEnv('<Name_of_SonarQube_environment_on_Jenk
i ns>') { sh """
<PATH_TO SONARQUBE_SCANNER_FOLDER>/bin/sonar-scanner \
-D sonar.login=<SonarQube_USERNAME> \
-D sonar.password=<SonarQube_PASSWORD> \
-D sonar.projectKey=<Project_KEY> \
-D sonar.exclusions=vendor/**,resources/**,*/*.java \
-D sonar.host.url=<SonarQube_URL>(default: http://localhost:9000/)
"""

}
}
}

```

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

Stage View



Permalinks

- [Last build \(#1\), 5 min 59 sec ago](#)
- [Last failed build \(#1\), 5 min 59 sec ago](#)

11. Check console

Jenkins

Dashboard > SonarQube_pipeline > #

Status Changes Console Output Edit Build Information Delete build '#2' Timings Git Build Data Pipeline Overview Pipeline Console Replay Pipeline Steps Workspaces

Console Output

Skipping 4,247 KB.. [Full Log](#)

19:43:43.934 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/report/gui/LineGraphGui.html for block at line 145. Keep only the first 100 references.
19:43:43.934 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/report/gui/LineGraphGui.html for block at line 506. Keep only the first 100 references.
19:43:43.934 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/report/gui/LineGraphGui.html for block at line 759. Keep only the first 100 references.
19:43:43.934 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/report/gui/LineGraphGui.html for block at line 705. Keep only the first 100 references.
19:43:43.934 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/report/gui/LineGraphGui.html for block at line 659. Keep only the first 100 references.
19:43:43.934 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/report/gui/LineGraphGui.html for block at line 511. Keep only the first 100 references.
19:43:43.934 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/report/gui/LineGraphGui.html for block at line 578. Keep only the first 100 references.
19:43:43.934 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/report/gui/LineGraphGui.html for block at line 707. Keep only the first 100 references.
19:43:43.934 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/report/gui/LineGraphGui.html for

Download Copy View as plain text

12.Now, check the project in SonarQube:

The screenshot shows the SonarQube 'main' project overview. At the top, there's a navigation bar with tabs for Overview, Issues, Security Hotspots, Measures, Code, and Activity. On the right, there are Project Settings and Project Information options. Below the navigation, a large green 'Passed' button indicates the quality gate status. A yellow warning box says 'The last analysis has warnings. See details'. The 'New Code' section shows 'New issues: 0' and 'Accepted issues: 0'. The 'Coverage' section shows a blue circle icon. The 'Duplications' section shows a grey circle icon. The 'Security Hotspots' section shows a green circle icon with the letter 'A'. At the bottom right, it says 'Last analysis 26 minutes ago'.

13.code problems consistency:

The screenshot shows the SonarQube code problems list for the file 'gameoflife-acceptance-tests/Dockerfile'. On the left, there's a sidebar with 'My Issues' and 'All' buttons, and a 'Filters' section with dropdowns for 'Issues in new code', 'Clean Code Attribute' (Consistency: 197k, Intentionality: 14k, Adaptability: 0, Responsibility: 0), and 'Software Quality'. The main area lists three code smells with checkboxes and descriptions. Each item includes an 'Intentionality' button and a 'No tags +' button. The first item is 'Use a specific version tag for the image.' (L1 - 5min effort - 4 years ago - Code Smell - Major). The second is 'Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.' (L12 - 5min effort - 4 years ago - Code Smell - Major). The third is another 'Surround this variable with double quotes...' entry (L12 - 5min effort - 4 years ago - Code Smell - Major).

14. Intentionality:

The screenshot shows a software interface for managing code quality issues. On the left, there's a sidebar with filters for 'My Issues' and 'All'. Under 'Filters', there are sections for 'Issues in new code' and 'Clean Code Attribute'. The 'Clean Code Attribute' section is expanded, showing 'Consistency' (197k), 'Intentionality' (14k, highlighted in blue), 'Adaptability' (0), and 'Responsibility' (0). Below this is a button 'Add to selection Ctrl + click'. At the top right, there are buttons for 'Bulk Change', 'Select issues', 'Navigate to issue', and statistics: '13,887 issues' and '59d effort'. The main area displays a list of issues under 'gameoflife-acceptance-tests/Dockerfile'. The first issue is 'Use a specific version tag for the image.' (Intentionality, Maintainability, Open, Not assigned, L1 - 5min effort, 4 years ago, Code Smell, Major). The second issue is 'Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.' (Intentionality, Maintainability, Open, Not assigned, L12 - 5min effort, 4 years ago, Code Smell, Major). The third issue is another 'Surround this variable with double quotes...' entry (Intentionality, Maintainability, Open, Not assigned, L12 - 5min effort, 4 years ago, Code Smell, Major).

15. Bugs

The screenshot shows a software interface for managing bugs. On the left, there's a sidebar with filters for 'Software Quality' (Security 0, Reliability 14k, Maintainability 0) and 'Severity' (0). Below this is a section for 'Type' with 'Bug' selected (14k, highlighted in blue), followed by 'Vulnerability' (0) and 'Code Smell' (268). There's also a button 'Add to selection Ctrl + click'. At the top right, there are buttons for 'Bulk Change', 'Select issues', 'Navigate to issue', and statistics: '13,619 issues' and '56d effort'. The main area displays a list of bugs under 'gameoflife-core/build/reports/tests/all-tests.html'. The first bug is 'Add "lang" and/or "xml:lang" attributes to this "<html>" element' (Intentionality, Reliability, Open, Not assigned, L1 - 2min effort, 4 years ago, Bug, Major). The second bug is 'Add "<th>" headers to this "<table>".' (Intentionality, Reliability, Open, Not assigned, L9 - 2min effort, 4 years ago, Bug, Major). Below these is a link to 'gameoflife-core/build/reports/tests/allclasses-frame.html'.

Code smells:

Sonarqube-test / main

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

Type: 1 Bug: 14k Vulnerability: 0 Code Smell: 253

Add to selection Ctrl + click

Scope Status Security Category

gameoflife-web/tools/jmeter/printable_docs/building.html

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

Open Not assigned L29 - 5min effort - 4 years ago - Code Smell - Minor

gameoflife-web/tools/jmeter/printable_docs/changes.html

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

Open Not assigned L31 - 5min effort - 4 years ago - Code Smell - Minor

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

SonarQube™ technology is powered by SonarSource SA

Community Edition v10.6 (92116) ACTIVE LGPL v3 Community Documentation Plugins Web API

Duplications:

Sonarqube-test / main

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

Coverage Duplications Overview New Code Duplicated Lines: 0 Duplicated Blocks: 0 Overall Code Density: 50.6% Duplicated Lines: 384,007

Duplications Overview (Only showing data for the first 500 files) See the data presented on this chart as a list Size: Duplicated Blocks Zoom: 100%

Duplicated Lines

localhost:9000/component_measures/metric=Duplications&id=Sonarqube-test#

Cyclomatic Complexities:

The screenshot shows the SonarQube interface for the project "Sonarqube-test / main". The "Measures" tab is selected. On the left, a sidebar displays various code metrics: Duplicated Blocks (0), Overall Code, Density (50.6%), Duplicated Lines (384,007), Duplicated Blocks (42,799), and Duplicated Files (979). Below these are dropdown menus for "Size" and "Complexity" (selected), and a link to "Issues". The main panel shows the project structure under "Sonarqube-test":

Category	Sub-Category	Value
Cyclomatic Complexity	gameoflife-acceptance-tests	
	gameoflife-build	
	gameoflife-core	18
	gameoflife-deploy	
	gameoflife-web	1,094
	Total	1,112

At the top right, there are buttons for "View as Tree", "Select files", "Navigate", and a note about "New Code: Since September 26, 2024".

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

ADVANCE DEVOPS EXP 9

Name :- Soham Satpute

Roll_no :- 52

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

1. Create an Amazon Linux EC2 Instance

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with links like EC2 Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, and Dedicated Hosts. The main area displays a table titled 'Instances (1/6) Info' with the following data:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
Webpage-env	i-08c9a73704a6bb7e1	Running	t3.micro	3/3 checks passed	View alarms +	ap-south-1a	ec2-3-100-111-111
nagios-host	i-0c656f86538bd5d01	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1b	ec2-15-100-111-111
linux-client1	i-0af927414560f6909	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1b	ec2-13-100-111-111

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. It lists several security groups and their details. The focus is on the 'Edit inbound rules' section for the security group 'sg-0b6d663161209f32d'. The table shows the following inbound rules:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-00d59807a11e23687	All ICMP - IPv4	ICMP	All	Custom	0.0.0.0/0
sgr-0ae620ec0b187c4a7	All traffic	All	All	Custom	0.0.0.0/0
sgr-0775d4388ffe14db6	SSH	TCP	22	Custom	0.0.0.0/0
sgr-0ebaadedcb97cb60fc	HTTP	TCP	80	Custom	0.0.0.0/0
sgr-08983e0020306b273	HTTPS	TCP	443	Custom	0.0.0.0/0

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

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

The screenshot shows a terminal window within the AWS CloudShell interface. At the top, there's a blue header bar with the AWS logo, 'Services' menu, a search bar containing 'Search', and a keyboard shortcut note: 'To tab out of the terminal window and select the next button element, press the left and right Shift keys together.' On the right side of the header, there are icons for 'Mumbai' and 'SohamSatpute'. Below the header, the terminal window has a black background. It displays a command history starting with a root prompt '#', followed by several commands related to Amazon Linux 2023, including a URL for the distribution's documentation. The command history ends with '[ec2-user@ip-172-31-2-12 ~]\$'. A vertical scroll bar is visible on the right edge of the terminal window.

4. Update the package indices and install the following packages using yum sudo yum update

```
[ec2-user@ip-172-31-40-254 ~]$ sudo yum update
Last metadata expiration check: 0:01:31 ago on Wed Oct 2 05:48:47 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-40-254 ~]$ █
```

sudo yum install httpd php

```
[ec2-user@ip-172-31-40-254 ~]$ sudo yum install httpd php
Last metadata expiration check: 0:01:59 ago on Wed Oct 2 05:48:47 2024.
Dependencies resolved.
=====
 Package          Architecture Version       Repository      Size
=====
Installing:
httpd            x86_64      2.4.62-1.amzn2023   amazonlinux    48 M
php8.3           x86_64      8.3.10-1.amzn2023.0.1  amazonlinux    10 M
Installing dependencies:
apr              x86_64      1.7.2-2.amzn2023.0.2  amazonlinux    129 M
apr-util         x86_64      1.6.3-1.amzn2023.0.1  amazonlinux    98 M
generic-logos-httdp x86_64      18.0.0-12.amzn2023.0.3  amazonlinux    19 M
httpd-core       x86_64      2.4.62-1.amzn2023   amazonlinux    1.4 M
httpd-filesystem x86_64      2.4.62-1.amzn2023   amazonlinux    14 M
httpd-tools      x86_64      2.4.62-1.amzn2023   amazonlinux    81 M
libbrotli        x86_64      1.0.9-4.amzn2023.0.2  amazonlinux    315 M
libsodium        x86_64      1.0.19-4.amzn2023   amazonlinux    176 M
libxml2          x86_64      1.1.34-5.amzn2023.0.2  amazonlinux    241 M
mailcap          noarch     2.1.49-3.amzn2023.0.3  amazonlinux    33 M
nginx-filesystem noarch     1:1.24.0-1.amzn2023.0.4  amazonlinux    9.8 M
php8.3-cli       x86_64      8.3.10-1.amzn2023.0.1  amazonlinux    3.7 M
php8.3-common    x86_64      8.3.10-1.amzn2023.0.1  amazonlinux    737 M
php8.3-process   x86_64      8.3.10-1.amzn2023.0.1  amazonlinux    45 M
```

sudo yum install gcc glibc glibc-common

```
[ec2-user@ip-172-31-40-254 ~]$ sudo yum install gcc glibc glibc-common
Last metadata expiration check: 0:02:41 ago on Wed Oct 2 05:48:47 2024.
Package glibc-2.34-52.amzn2023.0.11.x86_64 is already installed.
Package glibc-common-2.34-52.amzn2023.0.11.x86_64 is already installed.
Dependencies resolved.
=====
Package           Architecture     Version          Repository      Size
=====
Installing:
  gcc             x86_64         11.4.1-2.amzn2023.0.2   amazonlinux    32 M
  glibc           noarch        10.93-1.amzn2023.0.1   amazonlinux    92 M
  glibc-common    noarch        10.93-1.amzn2023.0.1   amazonlinux    887 M
  annobin-docs    x86_64         11.4.1-2.amzn2023.0.2   amazonlinux    10 M
  annobin-plugin-gcc x86_64         11.4.1-2.amzn2023.0.2   amazonlinux    27 M
  cpp             x86_64         8.0.4-5.amzn2023.0.2   amazonlinux    105 M
  gc              x86_64         2.34-52.amzn2023.0.11  amazonlinux    427 M
  glibc-devel     x86_64         2.34-52.amzn2023.0.11  amazonlinux    6.4 M
  glibc-headers-x86 x86_64         2.2.7-2.amzn2023.0.3   amazonlinux    1.4 M
  guile22        x86_64         6.1.109-118.189.amzn2023  amazonlinux
  kernel-headers x86_64         1.2.1-2.amzn2023.0.2   amazonlinux    62 M
  libmpc          x86_64         2.4.7-1.amzn2023.0.3   amazonlinux    38 M
  libtool-ltdl    x86_64         4.4.33-7.amzn2023       amazonlinux    32 M
  libxcrypt-devel x86_64         1:4.3-5.amzn2023.0.2   amazonlinux    534 M
  make            x86_64
=====
Transaction Summary
=====

```

sudo yum install gd gd-devel

```
[ec2-user@ip-172-31-40-254 ~]$ sudo yum install gd gd-devel
Last metadata expiration check: 0:03:46 ago on Wed Oct 2 05:48:47 2024.
Dependencies resolved.
=====
Package           Architecture     Version          Repository      Size
=====
Installing:
  gd              x86_64         2.3.3-5.amzn2023.0.3   amazonlinux    139 M
  gd-devel        x86_64         2.3.3-5.amzn2023.0.3   amazonlinux    38 M
  Installing dependencies:
    brotli         x86_64         1.0.9-4.amzn2023.0.2   amazonlinux    314 M
    brotli-devel   x86_64         1.0.9-4.amzn2023.0.2   amazonlinux    31 M
    bzip2-devel    x86_64         1.0.8-6.amzn2023.0.2   amazonlinux    214 M
    cairo          x86_64         1.17.6-2.amzn2023.0.1   amazonlinux    694 M
    cmake-filesystem x86_64         3.22.2-1.amzn2023.0.4   amazonlinux    16 M
    fontconfig      x86_64         2.13.94-2.amzn2023.0.2  amazonlinux    273 M
    fontconfig-devel x86_64         2.13.94-2.amzn2023.0.2  amazonlinux    128 M
    fonts-filesystem noarch        1:2.0.5-12.amzn2023.0.2  amazonlinux    9.5 M
    freetype        x86_64         2.13.2-5.amzn2023.0.1   amazonlinux    423 M
    freetype-devel  x86_64         2.13.2-5.amzn2023.0.1   amazonlinux    912 M
    glib2-devel     x86_64         2.74.7-689.amzn2023.0.2  amazonlinux    486 M
    google-noto-fonts-common noarch        20201206-2.amzn2023.0.2  amazonlinux    15 M
    google-noto-sans-vf-fonts noarch        20201206-2.amzn2023.0.2  amazonlinux    492 M
    graphite2       x86_64         1.3.14-7.amzn2023.0.2   amazonlinux    97 M
    graphite2-devel x86_64         1.3.14-7.amzn2023.0.2   amazonlinux    21 M
    harfbuzz        x86_64         7.0.0-2.amzn2023.0.1   amazonlinux    868 M
    harfbuzz-devel  x86_64         7.0.0-2.amzn2023.0.1   amazonlinux    404 M
    harfbuzz-icu    x86_64         7.0.0-2.amzn2023.0.1   amazonlinux    18 M
  =====
```

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

sudo adduser -m nagios sudo passwd nagios

```
[ec2-user@ip-172-31-40-254 ~]$ sudo adduser -m nagios
[ec2-user@ip-172-31-40-254 ~]$ sudo passwd nagios
Changing password for user nagios.
New password:
BAD PASSWORD: The password is a palindrome
Retype new password:
Sorry, passwords do not match.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-40-254 ~]$
```

6. Create a new user group sudo groupadd nagcmd

```
[ec2-user@ip-172-31-40-254 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-40-254 ~]$
```

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

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

```
[ec2-user@ip-172-31-40-254 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-40-254 ~]$ sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
[ec2-user@ip-172-31-40-254 ~]$
```

8. Create a new directory for Nagios downloads mkdir ~/downloads cd ~/downloads

```
[ec2-user@ip-172-31-40-254 ~]$ mkdir ~/downloads  
cd ~/downloads  
[ec2-user@ip-172-31-40-254 downloads]$ █
```

9. Use wget to download the source zip files.

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

```
[ec2-user@ip-172-31-40-254 downloads]$ 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
--2024-10-02 06:15:45-- https://nagios-plugins.org/download/nagios-plugins-2.3.3.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|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2782610 (2.7M) [application/x-qzip]
Saving to: 'nagios-plugins-2.3.3.tar.gz'

nagios-plugins-2.3.3.tar.gz      0%[=====]   632.00K  3.02MB/s
nagios-plugins-2.3.3.tar.gz    23%[=====>]  2.65M  8.10MB/s  in 0.3s
nagios-plugins-2.3.3.tar.gz  100%[=====]  2.65M  8.10MB/s  in 0.3s

2024-10-02 06:15:46 (8.10 MB/s) - 'nagios-plugins-2.3.3.tar.gz' saved [2782610/2782610]

[ec2-user@ip-172-31-40-254 downloads]$
```

<https://nagios-plugins.org/download/nagios-plugins-2.3.3.tar.gz>

```
[ec2-user@ip-172-31-40-254 downloads]$ wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz
--2024-10-02 06:17:24-- 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          0%[=====]   0      0 --.-KB/s
nagios-4.4.6.tar.gz          4%[==>]    495.62K 2.40MB/s
nagios-4.4.6.tar.gz          30%[=====>]   3.26M 7.99MB/s
nagios-4.4.6.tar.gz          63%[=====>>>]  6.91M 11.0MB/s
nagios-4.4.6.tar.gz          96%[=====>>>>] 10.46M 12.6MB/s
nagios-4.4.6.tar.gz     100%[=====>>>>] 10.81M 12.9MB/s  in 0.8s

2024-10-02 06:17:25 (12.9 MB/s) - 'nagios-4.4.6.tar.gz' saved [11333414/11333414]

[ec2-user@ip-172-31-40-254 downloads]$
```

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

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

```
[ec2-user@ip-172-31-40-254 downloads]$ tar zxvf nagios-4.4.6.tar.gz
nagios-4.4.6/
nagios-4.4.6/.gitignore
nagios-4.4.6/.travis.yml
nagios-4.4.6/CONTRIBUTING.md
nagios-4.4.6/ChangeLog
nagios-4.4.6/INSTALLING
nagios-4.4.6/LEGAL
nagios-4.4.6/LICENSE
nagios-4.4.6/Makefile.in
nagios-4.4.6/README.md
nagios-4.4.6/THANKS
nagios-4.4.6/UPGRADING
nagios-4.4.6/aclocal.m4
nagios-4.4.6/autoconf-macros/
nagios-4.4.6/autoconf-macros/.gitignore
nagios-4.4.6/autoconf-macros/CHANGELOG.md
nagios-4.4.6/autoconf-macros/LICENSE
nagios-4.4.6/autoconf-macros/LICENSE.md
nagios-4.4.6/autoconf-macros/README.md
nagios-4.4.6/autoconf-macros/add_group_user
nagios-4.4.6/autoconf-macros/ax_nagios_get_distrib
nagios-4.4.6/autoconf-macros/ax_nagios_get_files
nagios-4.4.6/autoconf-macros/ax_nagios_get_inetd
nagios-4.4.6/autoconf-macros/ax_nagios_get_init
nagios-4.4.6/autoconf-macros/ax_nagios_get_os
nagios-4.4.6/autoconf-macros/ax_nagios_get_paths
nagios-4.4.6/autoconf-macros/ax_nagios_get_ssl
nagios-4.4.6/base/
nagios-4.4.6/base/.gitignore
nagios-4.4.6/base/Makefile.in
nagios-4.4.6/base/broker.c
```

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

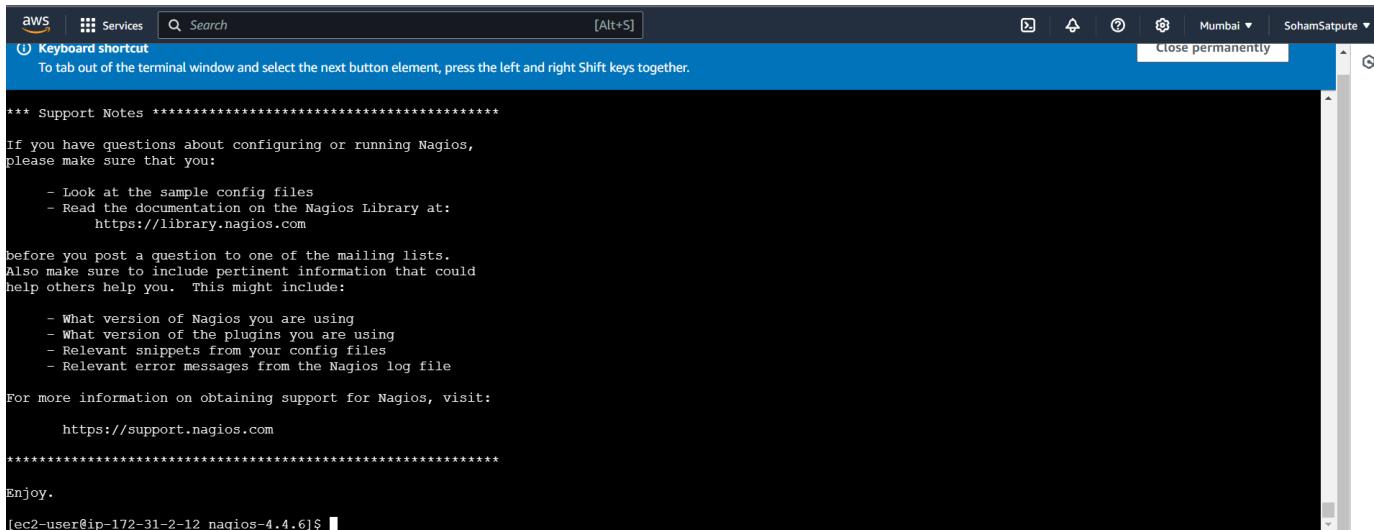
```
./configure --with-command-group=nagcmd
```

```
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$ ./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables:
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether 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 whether ln -s works... 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
checking for string.h... yes
checking for memory.h... yes
checking for strings.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for unistd.h... yes
checking arpa/inet.h usability... yes
checking arpa/inet.h presence... yes
checking for arpa/inet.h... yes
```

12. Compile the source code.

make all

```
[ec2-user@ip-172-31-40-254 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 nebmods.o nebmods.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: '%s' directive argument is null [-Wformat-overflows]
  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
gcc -Wall -I... -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o netutils.o netutils.c
netutils.c: In function 'my_tcp_connect':
netutils.c:50:47: warning: '%d' directive output may be truncated writing between 1 and 11 bytes into a region of size 6 [-Wformat-truncation=]
  50 |     sprintf(port_str, sizeof(port_str), "%d", port);
   |     ^
   |     ~~~
netutils.c:50:46: note: directive argument in the range [-2147483648, 65535]
  50 |     sprintf(port_str, sizeof(port_str), "%d", port);
   |     ^
netutils.c:50:9: note: 'sprintf' output between 2 and 12 bytes into a destination of size 6
  50 |     sprintf(port_str, sizeof(port_str), "%d", port);
   |     ^
   |     ~~~~~
gcc -Wall -I... -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o notifications.o notifications.c
```



The screenshot shows a terminal window with the AWS logo and services menu at the top. A search bar and a 'Close permanently' button are also visible. The terminal content includes support notes for Nagios, a warning about truncation, and a note about the 'sprintf' output. It ends with a copyright notice for Nagios Core 4.4.6 and a link to the official website.

```
*** 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-172-31-2-12 nagios-4.4.6]$
```

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

```
./sudo make install sudo make  
install-init sudo make install-  
config sudo make install-  
commandmode
```

```
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$ ./sudo make install  
sudo make install-init  
sudo make install-commandmode  
-bash: ./sudo: No such file or directory  
/usr/bin/install -c -m 755 -d -o root -g root /lib/systemd/system  
/usr/bin/install -c -m 755 -o root -g root startup/default-service /lib/systemd/system/nagios.service  
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc  
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc  
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/nagios.cfg /usr/local/nagios/etc/nagios.cfg  
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/cgi.cfg /usr/local/nagios/etc/cgi.cfg  
/usr/bin/install -c -b -m 660 -o nagios -g nagios sample-config/resource.cfg /usr/local/nagios/etc/resource.cfg  
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/templates.cfg /usr/local/nagios/etc/objects/templates.cfg  
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/commands.cfg /usr/local/nagios/etc/objects/commands.cfg  
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/contacts.cfg /usr/local/nagios/etc/objects/contacts.cfg  
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/timeperiods.cfg /usr/local/nagios/etc/objects/timeperiods.cfg  
/usr/bin/install -c -b -m 664 -o nagios sample-config/template-object/localhost.cfg /usr/local/nagios/etc/objects/localhost.cfg  
/usr/bin/install -c -b -m 664 -o nagios sample-config/template-object/windows.cfg /usr/local/nagios/etc/objects/windows.cfg  
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/printer.cfg /usr/local/nagios/etc/objects/printer.cfg  
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/switch.cfg /usr/local/nagios/etc/objects/switch.cfg  
  
*** Config files installed ***  
Remember, these are *SAMPLE* config files. You'll need to read  
the documentation for more information on how to actually define  
services, hosts, etc. to fit your particular needs.  
  
/usr/bin/install -c -m 775 -o nagios -g nagcmd -d /usr/local/nagios/var/rw  
chmod g+s /usr/local/nagios/var/rw  
  
*** External command directory configured ***
```

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

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

```
GNU nano 5.8                                         /usr/local/nagios/etc/objects/contacts.cfg                         Modifier  
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             bhagyeshpatil0702@gmail.com; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS *****  
  
    # CONTACT GROUPS  
    #  
    # We only have one contact in this simple configuration file, so there is  
    # no need to create more than one contact group.  
  
    define contactgroup {  
        ^G Help           ^C Write Out     ^W Where Is     ^K Cut           ^X Execute      ^C Location     M-U Undo      M-A Set Mark   M-J To Bracket M-Q Previous  
        ^X Exit          ^R Read File     ^V Replace      ^U Paste         ^J Justify      ^L Go To Line   M-E Redo      M-D Copy      M-Q Where Was  M-W Next  
    }  
}
```

15. Configure the web interface.

```
sudo make install-webconf
```

```
[ec2-user@ip-172-31-40-254 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-172-31-40-254 nagios-4.4.6]$
```

16. Create a nagiosadmin account for nagios login along with password. You'll have to specify the password twice. sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin

```
[ec2-user@ip-172-31-40-254 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-172-31-40-254 nagios-4.4.6]$
```

17. Restart Apache sudo systemctl restart httpd

```
Adding password for user nagiosadmin
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$ sudo systemctl restart httpd
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$
```

cd ~/downloads tar zxvf nagios-

plugins-2.3.3.tar.gz cd nagios-

plugins-2.3.3

```
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$ sudo systemctl restart httpd
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$ cd ~/downloads
tar zxvf 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-Catch-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
nagios-plugins-2.3.3/config.h.in
nagios-plugins-2.3.3/Changelog
nagios-plugins-2.3.3/LICENSES
nagios-plugins-2.3.3/lib/
nagios-plugins-2.3.3/lib/parse_ini.h
nagios-plugins-2.3.3/lib/extr_opts.c
nagios-plugins-2.3.3/lib/Makefile.in
```

18. Go back to the downloads folder and unzip the plugins zip file. ./configure --with-nagios-user=nagios --with-nagios-

```
group=nagios make sudo make install
```

```
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... /usr/bin/grep
```

19. Compile and install plugins 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...

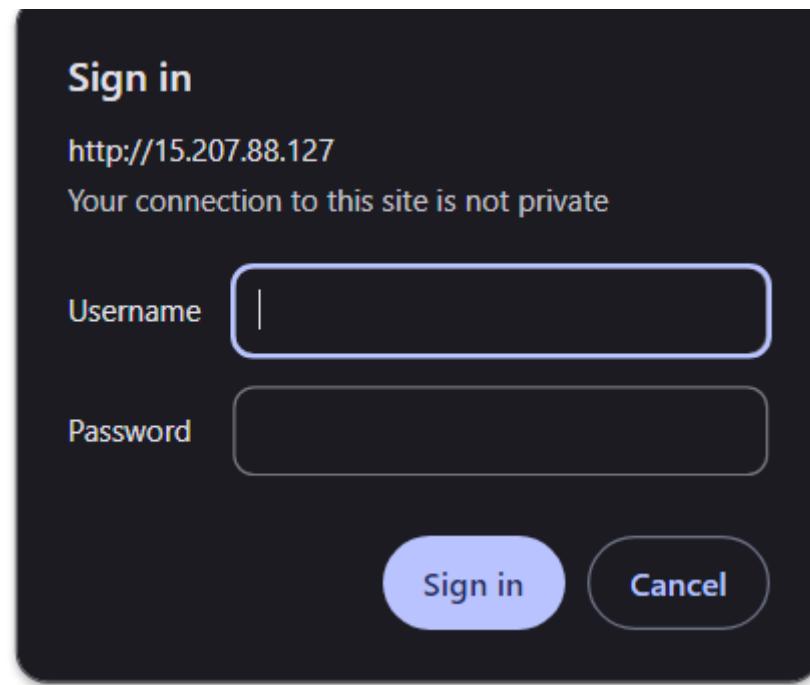
Checking objects...
  Checked 8 services.
  Checked 1 hosts.
  Checked 1 host groups.
```

20. Check the status of Nagios

```
things look okay - No serious problems were detected during the pre-flight check
ec2-user@ip-172-31-45-178 nagios-plugins-2.3.3]$ sudo systemctl status nagios
nagios.service - Nagios Core 4.4.6
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
     Active: active (running) since Wed 2024-10-02 05:37:36 UTC; 14s ago
       Docs: https://www.nagios.org/documentation
    Process: 67990 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Process: 67991 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 67992 (nagios)
   Tasks: 6 (limit: 1112)
  Memory: 2.0M
    CPU: 16ms
   CGroup: /system.slice/nagios.service
           └─ 67992 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
              ├─ 67993 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─ 67994 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─ 67995 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              ├─ 67996 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
              └─ 67997 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 02 05:37:36 ip-172-31-45-178.ec2.internal nagios[67992]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfully initialized
Oct 02 05:37:36 ip-172-31-45-178.ec2.internal nagios[67992]: qh: core query handler registered
```

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



Nagios® Core™

Process running with PID 90965

Nagios® Core™
Version 4.4.6
April 28, 2020
Check for updates

A new version of Nagios Core is available!
Visit nagios.org to download Nagios 4.5.5.

Get Started

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

Quick Links

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

Latest News

Don't Miss...

Copyright © 2010-2020 Nagios Core Development Team and Community Contributors. Copyright © 1999-2009 Ethan Galstad. See the THANKS file for more information on contributors.

Page Tour

Advance DevOps Exp 10

Name:- Soham Satpute

Roll :- 52

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

Procedure:-

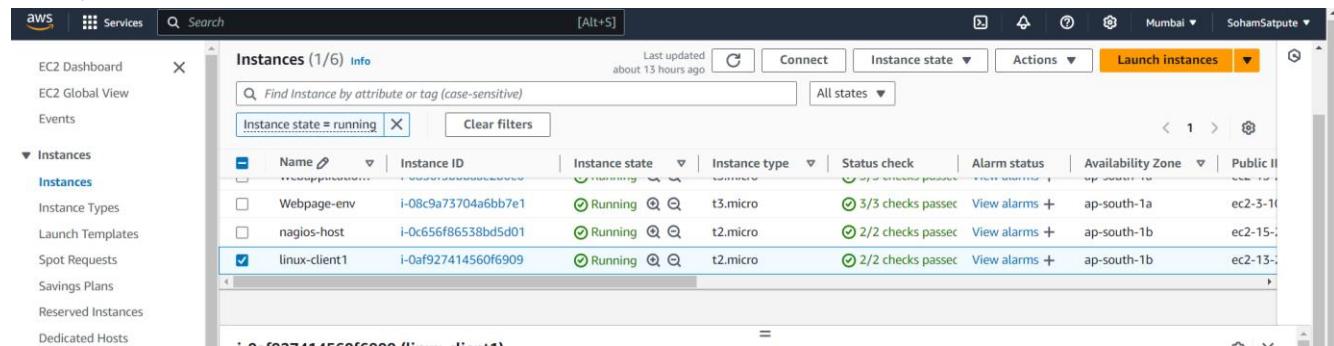
Check if the nagios service is running by executing following command

sudo systemctl status nagios

```
ubuntu@ip-172-31-89-161:~$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.4.6
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: enabled)
   Active: active (running) since Sat 2024-09-28 16:08:58 UTC; 1min 2s ago
     Docs: https://www.nagios.org/documentation
 Process: 15743 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Process: 15753 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
Main PID: 15764 (nagios)
   Tasks: 6 (limit: 1130)
  Memory: 2.4M (peak: 3.2M)
    CPU: 29ms
   CGroup: /system.slice/nagios.service
           ├─15764 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           ├─15765 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─15766 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─15767 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─15768 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─15769 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfully initialized
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: qh: core query handler registered
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: qh: echo service query handler registered
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: qh: help for the query handler registered
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: wproc: Successfully registered manager as @wproc with query handler
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: wproc: Registry request: name=Core Worker 15765;pid=15765
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: wproc: Registry request: name=Core Worker 15766;pid=15766
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: wproc: Registry request: name=Core Worker 15767;pid=15767
```

Now, create a new EC2 instance on AWS



Now perform the following commands on nagios-host EC2 instance. On the server, run this command

ps -ef | grep nagios

```
ubuntu@ip-172-31-89-161:~$ ps -ef | grep nagios
nagios 15764 1 0 16:08 ? 00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios 15765 15764 0 16:08 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
nagios 15766 15764 0 16:08 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
nagios 15767 15764 0 16:08 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
nagios 15768 15764 0 16:08 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
nagios 15769 15764 0 16:08 ? 00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
ubuntu 15957 1342 0 16:13 pts/0 00:00:00 grep --color=auto nagios
ubuntu@ip-172-31-89-161:~$
```

Sudo su

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

```
ubuntu@ip-172-31-89-161:~$ sudo su
mkdir /usr/local/nagios/etc/objects/monitorhosts
mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
root@ip-172-31-89-161:/home/ubuntu#
```

Copy localhost.cfg file to the mentioned location

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

```
root@ip-172-31-89-161:/usr/local/nagios/etc/objects# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
cp: cannot create regular file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts': No such file or directory
root@ip-172-31-89-161:/usr/local/nagios/etc/objects# sudo mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
root@ip-172-31-89-161:/usr/local/nagios/etc/objects# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
root@ip-172-31-89-161:/usr/local/nagios/etc/objects#
```

Open the nano editor for localhost.cfg file and make these changes. Add the Ip address of the linux-client for the address field.

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

```

GNU nano 7.2                               /usr/local/nagios/etc/nagios.cfg
#####
# HOST DEFINITION
#
#####

# Define a host for the local machine

define host {

    use          linux-server ; Name of host template
    ; This host definition is (or inherits) from the "linux-server" template

    host_name    linuxserver
    alias        linuxserver
    address      52.207.253.18
}

#####

# HOST GROUP DEFINITION

^G Help           ^O Write Out      ^W Where Is      ^K Cut            ^T Exit
^X Exit          ^R Read File       ^\ Replace       ^U Paste          ^J Ju

```

Note - Here replace hostname with linuxserver

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

Add the following line to the nagios.cfg file

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

```

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

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

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

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

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

```

After making the changes in nagios.cfg file now check validate the file by typing the following command in the terminal.

```
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
License: GPL

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

Running pre-flight check on configuration data...

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

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
root@ip-172-31-89-161:/usr/local/nagios/etc/objects/monitorhosts/linuxhosts#
```

Now restart the service by using this command

```
service nagios restart
```

```

root@ip-172-31-89-161:/usr/local/nagios/etc/objects/monitorhosts/linuxhosts# service nagios restart
root@ip-172-31-89-161:/usr/local/nagios/etc/objects/monitorhosts/linuxhosts# systemctl status nagios
● nagios.service - Nagios Core 4.4.6
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: enabled)
   Active: active (running) since Sat 2024-09-28 17:36:35 UTC; 19s ago
     Docs: https://www.nagios.org/documentation
 Process: 1870 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Process: 1872 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 1874 (nagios)
   Tasks: 8 (limit: 1130)
  Memory: 3.0M (peak: 3.2M)
    CPU: 24ms
   CGroup: /system.slice/nagios.service
           ├─1874 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           ├─1875 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─1876 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─1877 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─1878 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─1879 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/etc/nagios.cfg
           ├─1880 /usr/local/nagios/libexec/check_ping -H 52.207.253.18 -w 3000.0,80% -c 5000.0,100% -p 5
           └─1881 /usr/bin/ping -n -U -w 30 -c 5 52.207.253.18

Sep 28 17:36:35 ip-172-31-89-161 nagios[1874]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfully initialized
Sep 28 17:36:35 ip-172-31-89-161 nagios[1874]: qh: core query handler registered
Sep 28 17:36:35 ip-172-31-89-161 nagios[1874]: qh: echo service query handler registered
Sep 28 17:36:35 ip-172-31-89-161 nagios[1874]: qh: help for the query handler registered
Sep 28 17:36:35 ip-172-31-89-161 nagios[1874]: wproc: Successfully registered manager as @wproc with query handler
Sep 28 17:36:35 ip-172-31-89-161 nagios[1874]: wproc: Registry request: name=Core Worker 1875,pid=1875
lines 1-26

```

Now using this command update the apt repository of ubuntu (linux-client), install gcc, nagios-nrpe-server and nagios-plugin sudo apt update -y sudo apt install gcc -y

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

Now open nrpe.cfg file and add the ip address of the nagios host as shown. To open the nrpe.cfg file copy this command.

```

# Supported.
#
# Note: The daemon only does rudimentary checking
# address. I would highly recommend adding entries
# file to allow only the specified host to connect
# you are running this daemon on.
#
# NOTE: This option is ignored if NRPE is running
#       as a module in Apache.
allowed_hosts=127.0.0.1,54.167.169.0

#
# COMMAND ARGUMENT PROCESSING
# This option determines whether or not the NRPE
# to specify arguments to commands that are executed
# if the daemon was configured with the --enable-command
# option.

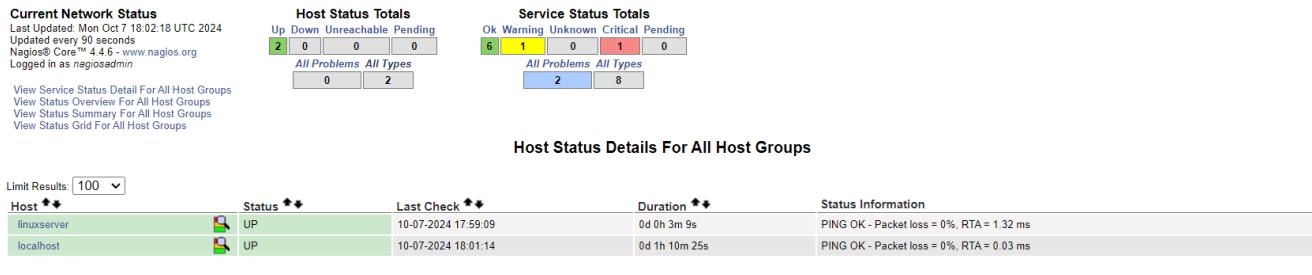
```

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

Now restart nrpe server by using this command

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

Now, check nagios dashboard, you should see linuxserver up and running, if not



Experiment 11

Soham Satpute D15A 52

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.

Packaging Functions

Lambda functions need to be packaged and sent to AWS. This is usually a process of compressing the function and all its dependencies and uploading it to an S3 bucket.

And letting AWS know that you want to use this package when a specific event takes place. To help us with this process we use the Serverless Stack Framework (SST). We’ll go over this in detail later on in this guide.

Execution Model

The container (and the resources used by it) that runs our function is managed completely by AWS. It is brought up when an event takes place and is turned off if it is not being used. If additional requests are made while the original event is being served, a new container is brought up to serve a request. This means that if we are undergoing a usage spike, the cloud provider simply creates multiple instances of the container with our function to serve those requests.

This has some interesting implications. Firstly, our functions are effectively stateless. Secondly, each request (or event) is served by a single instance of a Lambda function. This means that you are not going to be handling concurrent requests in your code.

AWS brings up a container whenever there is a new request. It does make some optimizations here. It will hang on to the container for a few minutes (5 - 15mins depending on the load) so it can respond to subsequent requests without a cold start.

Stateless Functions

The above execution model makes Lambda functions effectively stateless. This means that every time your Lambda function is triggered by an event it is invoked in a completely new environment. You don't have access to the execution context of the previous event.

However, due to the optimization noted above, the actual Lambda function is invoked only once per container instantiation. Recall that our functions are run inside containers. So when a function is first invoked, all the code in our handler function gets executed and the handler function gets invoked. If the container is still available for subsequent requests, your function will get invoked and not the code around it.

For example, the `createNewDbConnection` method below is called once per container instantiation and not every time the Lambda function is invoked. The `myHandler` function on the other hand is called on every invocation.

Common Use Cases for Lambda

Due to Lambda's architecture, it can deliver great benefits over traditional cloud computing setups for applications where:

1. Individual tasks run for a short time;
2. Each task is generally self-contained;

3. There is a large difference between the lowest and highest levels in the workload of the application.

Some of the most common use cases for AWS Lambda that fit these criteria are: Scalable APIs. When building APIs using AWS Lambda, one execution of a Lambda function can serve a single HTTP request. Different parts of the API can be routed to different Lambda functions via Amazon API Gateway. AWS Lambda automatically scales individual functions according to

the demand for them, so different parts of your API can scale differently according to current usage levels. This allows for cost-effective and flexible API setups.

Data processing. Lambda functions are optimized for event-based data processing. It is easy to integrate AWS Lambda with data sources like Amazon DynamoDB and trigger a Lambda function for specific kinds of data events. For example, you could employ Lambda to do some work every time an item in DynamoDB is created or updated, thus making it a good fit for things like notifications, counters and analytics.

Steps to create an AWS Lambda function

Step 1: Create a Lambda Function

1. Choose a Function Creation Method: Select Author from scratch.

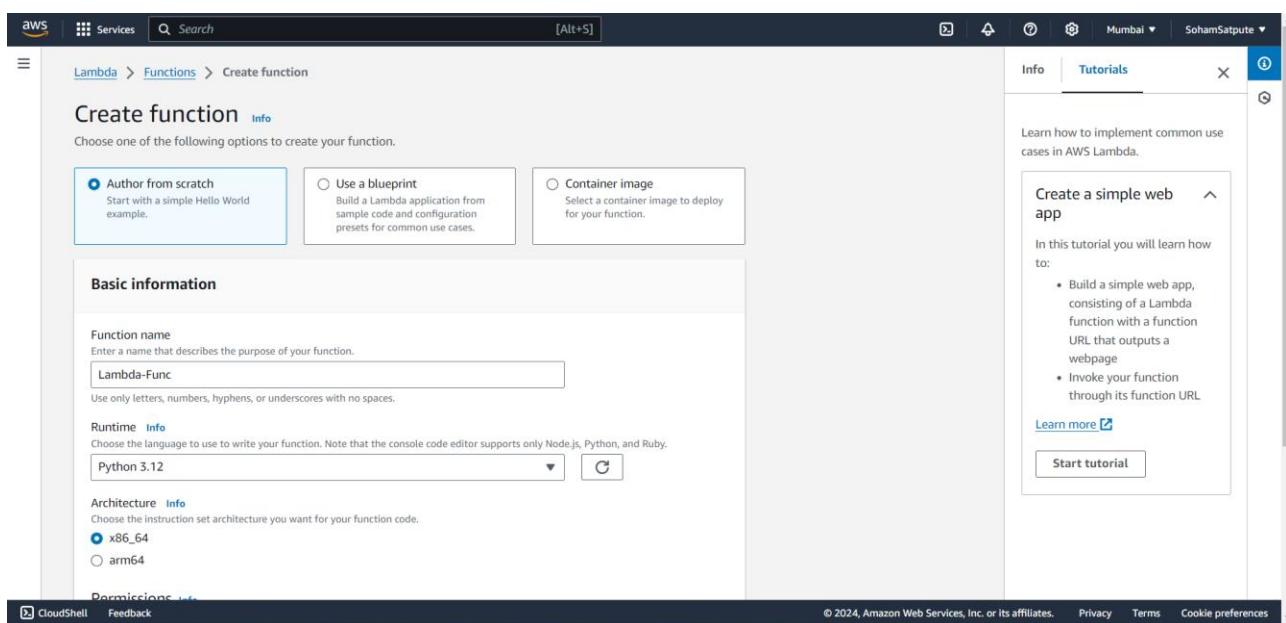
2. Configure the Function:

Function name: Enter a name for your function (e.g., MyFirstLambda).

Runtime: Choose Python 3.x (the latest available version).

Permissions: Choose Create a new role with basic Lambda permissions (this creates a role with the necessary permissions).

3. Click on Create function.



Step 2: Write Your Lambda Function Code

In the Function code section, you will see a code editor. Replace the default code with the following Python code:

```

python Copy code def lambda_handler(event, context): #
This function returns a greeting message name =
event.get('name', 'World')
return {
    'statusCode': 200, 'body': f'Hello, {name}!'
}

```

This function reads a name from the event and returns a greeting message. If no name is provided, it defaults to "World".

The screenshot shows the AWS Lambda console interface. On the left, there's a navigation bar with 'Services' and a search bar. Below it, the path 'Lambda > Functions > Lambda-Func' is shown. The main area is titled 'Lambda-Func' and contains a 'Function overview' section. It includes a 'Diagram' tab (selected), a 'Template' tab, a 'Layers' section (0 layers), and buttons for '+ Add trigger' and '+ Add destination'. To the right of the diagram is a 'Description' field with a minus sign, 'Last modified 2 minutes ago', and 'Function ARN' (arn:aws:lambda:ap-south-1:529088256210:function:Lambda-Func). Below that is a 'Function URL' field with a 'Info' link. On the far right, a sidebar titled 'Tutorials' is open, showing a section on 'Create a simple web app' with a list of steps: 'Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage' and 'Invoke your function through its function URL'. There are also 'Learn more' and 'Start tutorial' buttons.

lambda_function x Environment Vari x Execution results x

```

1 def lambda_handler(event, context):
2     # This function returns a greeting message
3     name = event.get('name', 'World')
4     return {
5         'statusCode': 200,
6         'body': f'Hello, {name}!'
7     }
8
9

```

Step3: 1. Configure a Test Event:

Click on the Test button.

In the Configure test event dialog, give your event a name (e.g., TestEvent). Replace the default JSON with the following:

```
{
  "name": "Lambda User"
}
```

```
}
```

2. Run the Test:

Click on the Test button again to execute your Lambda function.

You should see the execution results below the code editor, including the response: json

Copy code

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

Configure test event X

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

TestEvent

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

Event sharing settings

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

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

Template - optional

hello-world

Event JSON Format JSON

```
1 *  {
2   "name": "Lambda User"
3 }
4
```

Cancel

The screenshot shows the AWS Lambda function editor interface. On the left, there's a sidebar with 'Environment' selected. The main area has tabs for 'Code source' (which is currently active) and 'Info'. Below these are 'File', 'Edit', 'Find', 'View', 'Go', 'Tools', and 'Window' menus, along with 'Test' and 'Deploy' buttons. A search bar at the top right says 'Search [Alt+S]'. The main content area displays the code for 'lambda_function.py':

```
import json

def lambda_handler(event, context):
    # TODO implement
    return {
        'statusCode': 200,
        'body': json.dumps('Hello from Lambda!')
    }
```

Below the code, there are sections for 'Execution results', 'Function Logs', and 'Request ID'. The 'Execution results' section shows a successful execution with status 'Succeeded', max memory used: 32 MB, and time: 1.55 ms. The 'Function Logs' section shows log entries for the start, end, and report of the function execution. The 'Request ID' section shows the unique identifier for the request.

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

This screenshot shows the AWS Lambda function editor with the 'Code' tab selected. It has tabs for 'Code', 'Test', 'Monitor', 'Configuration', 'Aliases', and 'Versions'. The main area is identical to the previous screenshot, displaying the 'lambda_function.py' code. To the right, there's a sidebar with the function name 'function:Lambda-Func' and a 'Function URL' link. The 'Configuration' tab is visible above the sidebar.

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.

Adv. DevOps Exp. 12

Soham Satpute

D15A - 52

Step 1: Open the IAM (user)

The screenshot shows the AWS IAM Roles page. On the left, there's a sidebar with navigation links like Dashboard, Access management, and Access reports. The main area displays a table of roles with columns for Role name, Trusted entities, and Last activity. The roles listed are: `any-elasticbeanstalk-service-role-2`, `AWSServiceRoleForAutoScaling`, `AWSServiceRoleForSupport`, `AWSServiceRoleForTrustedAdvisor`, `myPythonLambdaFunction-role-a2x7el65`, and `test-2-role`. The last activity for most roles is 40 days ago.

Step 2: Under Attach Policies, add S3-ReadOnly and CloudWatchFull permissions to this role.

The screenshot shows the detailed view of the role `myPythonLambdaFunction-role-a2x7el65`. It includes a Summary section with creation date (October 07, 2023), ARN, and last activity. Below it is a Permissions tab where you can manage policies. A modal window is open over the page, specifically the "Add permissions" section under "Attach policies". It lists two policies: `arn:aws:iam::447953971928:role/service-role/myPythonLambdaFunction-role-a2x7el65` and `arn:aws:s3:::myPythonLambdaFunction-a2x7el65`. There are buttons for "Simulate" and "Remove". At the bottom of the modal, there are buttons for "Add permissions" (with an upward arrow icon), "Attach policies", and "Create inline policy".

S3-ReadOnly

The screenshot shows the 'Add permissions' dialog in the AWS IAM console. The URL is [IAM > Roles > myPythonLambdaFunction-role-a2x7el65 > Add permissions](#). The title is 'Attach policy to myPythonLambdaFunction-role-a2x7el65'. The 'Current permissions policies (1)' section is collapsed. The 'Other permissions policies (882)' section is expanded, showing a search bar with 'S3read' and a filter 'All types'. One result, 'AmazonS3ReadOnlyAccess', is listed as 'AWS managed' with a description: 'Provides read only access to all bucket...'. At the bottom are 'Cancel' and 'Add permissions' buttons.

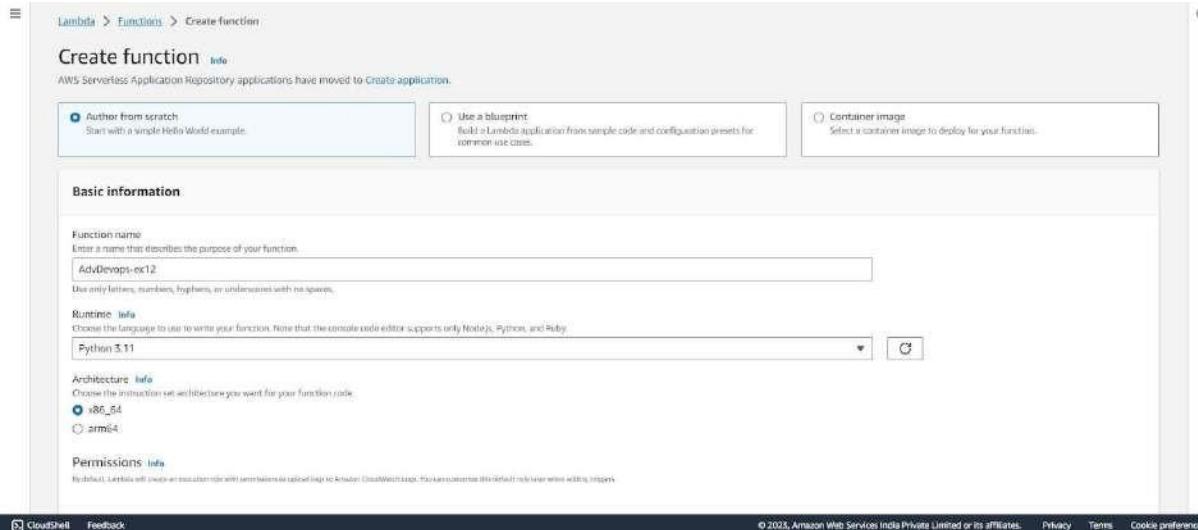
CloudWatchFull

The screenshot shows the 'Add permissions' dialog in the AWS IAM console. The URL is [IAM > Roles > myPythonLambdaFunction-role-a2x7el65 > Add permissions](#). The title is 'Attach policy to myPythonLambdaFunction-role-a2x7el65'. The 'Current permissions policies (2)' section is collapsed. The 'Other permissions policies (881)' section is expanded, showing a search bar with 'cloudwatchfull' and a filter 'All types'. Two results, 'CloudWatchFullAccess' and 'CloudWatchFullAccessV2', are listed as 'AWS managed' with descriptions: 'Provides full access to CloudWatch.' and 'Provides full access to CloudWatch.' respectively. At the bottom are 'Cancel' and 'Add permissions' buttons.

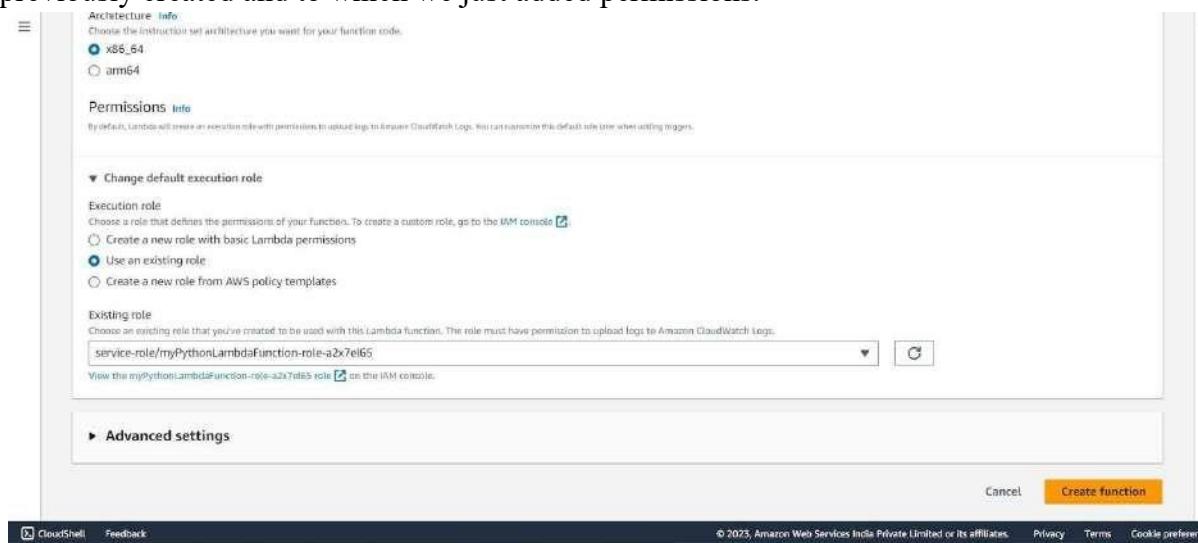
After successful attachment of policy you will see something like this you will be able to see the updated policies.

The screenshot shows the 'Permissions' tab in the AWS IAM console. The URL is [Identity and Access Management \(IAM\)](#). A green banner at the top says 'Policy was successfully attached to role.' Below it is a message 'Last modified: 1 hour ago'. The 'Permissions' tab is selected. The 'Permissions policies (3)' section shows three policies: 'AmazonS3ReadOnlyAccess' (AWS managed), 'AWSLambdaBasicExecutionRole' (Customer managed), and 'CloudWatchFullAccess' (AWS managed). Each policy has a checkbox, a type indicator, and an 'Attached entities' column showing '1'. At the bottom is a 'Permissions boundary (not set)' section.

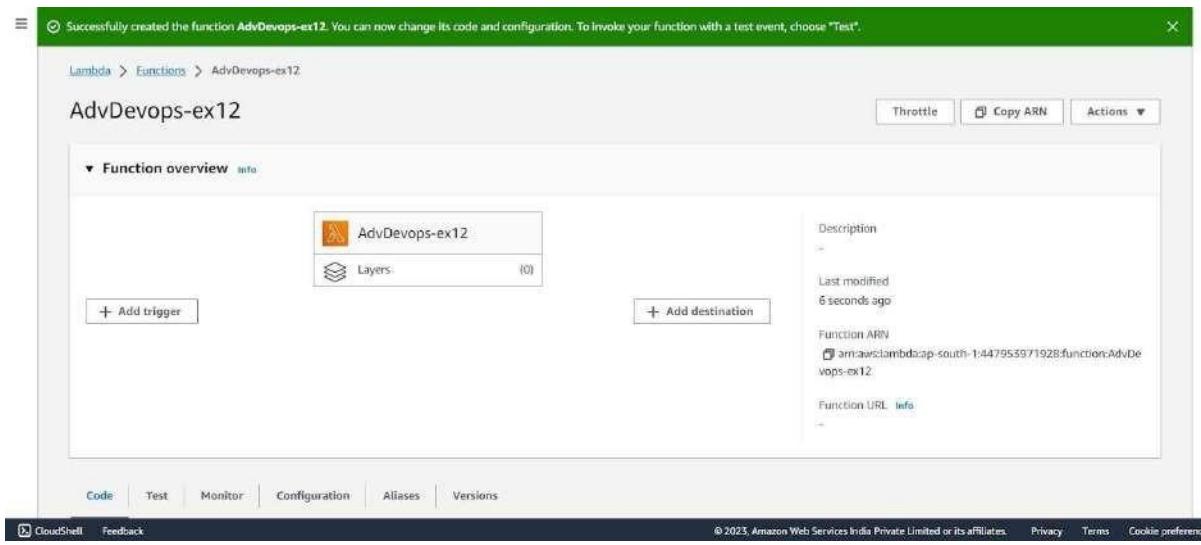
Step 3: Open up AWS Lambda and create a new Python function.



Under Execution Role, choose the existing role, then select the one which was previously created and to which we just added permissions.



Step 4: The function is up and running.



Step 5: Make the following changes to the function and click on the deploy button. This code basically logs a message and logs the contents of a JSON file which is uploaded to an S3 Bucket and then deploy the code.

```
1 import json
2 import boto3
3 import urllib
4
5 def lambda_handler(event, context):
6
7     s3_client = boto3.client('s3')
8     bucket_name = event['Records'][0]['s3']['bucket']['name']
9     key = event['Records'][0]['s3']['object']['key']
10    key_urlib.parse.unquote_plus(key, encoding='utf-8')
11    message = 'An file has been added with key ' + key + ' to the bucket ' + bucket_name
12    print(message)
13    response = s3_client.get_object(Bucket=bucket_name, Key=key)
14    contents = response['Body'].read().decode()
15    contents = json.loads(contents)
16
17    print("These are the Contents of the File: \n", contents)
18
19
```

CloudShell Feedback

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Step 6: Click on Test and choose the 'S3 Put' Template.

The screenshot shows the AWS Lambda console interface. At the top, a green banner indicates that a function has been successfully created. Below the banner, the navigation bar includes tabs for Code, Test, Monitor, Configuration, Aliases, and Versions. The 'Code' tab is selected, showing the 'Code source' section with a file browser. A file named 'lambda_function.py' is open, displaying Python code for a Lambda function:

```
1 import json
2 import boto3
3 import urllib
4
5 def lambda_handler(event, context):
```

Below the code editor is a modal window titled 'Configure test event'. The modal provides instructions for creating a test event and includes fields for 'Test event action' (radio buttons for 'Create new event' and 'Edit saved event', with 'Create new event' selected), 'Event name' (text input field containing 'test'), and 'Event sharing settings' (radio buttons for 'Private' and 'Shareable', with 'Private' selected). The 'Event JSON' section contains a dropdown menu set to 's3-put' and a 'Format JSON' button. At the bottom of the modal are 'Cancel', 'Invoke', and 'Save' buttons.

And Save it.

Step 7: Open up the S3 Console and create a new bucket.

The screenshot shows the Amazon S3 buckets list page. At the top, there's an 'Account snapshot' section with a link to 'View Storage Lens dashboard'. Below it, a table lists three buckets:

Name	AWS Region	Access	Creation date
elasticbeanstalk-ap-south-1-447953971928	Asia Pacific (Mumbai) ap-south-1	Objects can be public	August 7, 2023, 14:24:02 (UTC+05:30)
www.hellorachana.com	Asia Pacific (Mumbai) ap-south-1	Public	July 30, 2023, 15:05:54 (UTC+05:30)
www.htmlwebside.com	Asia Pacific (Mumbai) ap-south-1	Public	July 30, 2023, 15:49:06 (UTC+05:30)

At the bottom of the page, there are links for 'CloudShell', 'Feedback', and copyright information: '© 2023, Amazon Web Services India Private Limited or its affiliates.' followed by 'Privacy', 'Terms', and 'Cookie preferences'.

Step 8: With all general settings, create the bucket in the same region as the function.

The screenshot shows the 'Create bucket' wizard. The first step, 'General configuration', is displayed. It includes fields for 'Bucket name' (set to 'AdvDevopsexp12') and 'AWS Region' (set to 'Asia Pacific (Mumbai) ap-south-1'). There's also a 'Copy settings from existing bucket - optional' section with a 'Choose bucket' button. At the bottom, there's an 'Object Ownership' section with a note about controlling object ownership and access control lists (ACLs). The footer contains links for 'CloudShell', 'Feedback', and copyright information: '© 2023, Amazon Web Services India Private Limited or its affiliates.' followed by 'Privacy', 'Terms', and 'Cookie preferences'.

Step 9: Click on the created bucket and under properties, look for events.

Event notifications (0)
Send a notification when specific events occur in your bucket. [Learn more](#)

Name	Event types	Filters	Destination type	Destination
		No event notifications		

Choose [Create event notification](#) to be notified when a specific event occurs.

[Create event notification](#)

Amazon EventBridge
For additional capabilities, use Amazon EventBridge to build event-driven applications at scale using S3 event notifications. [Learn more](#) or see [EventBridge policies](#).

Transfer acceleration
Use an accelerated endpoint for faster data transfers. [Learn more](#)

Click on Create Event Notification.

Step 10: Mention an event name and check Put under event types.

General configuration

Event name
S3putrequest
Event name can contain up to 255 characters.

Prefix - optional
Limit the notifications to objects with key starting with specified characters.
images/

Suffix - optional
Limit the notifications to objects with key ending with specified characters.
.jpg

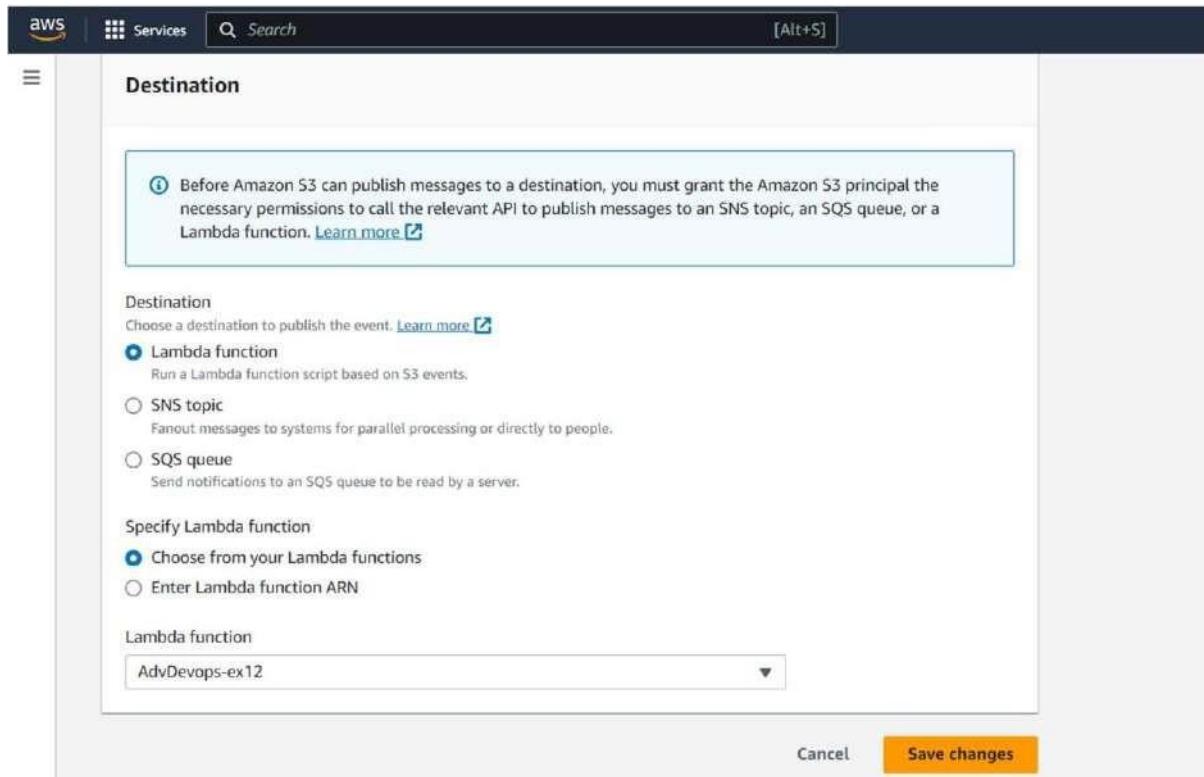
Event types
Specify at least one event for which you want to receive notifications. For each group, you can choose an event type for all events, or you can choose one or more individual events.

Object creation

All object create events
s3:ObjectCreated:
 Put
s3:ObjectCreated:Put

Post
s3:ObjectCreated:Post

Choose Lambda function as destination and choose your lambda function and save the changes.



CloudShell Feedback © 2023, Amazon Web Services

Step 11: Refresh the Lambda function console and you should be able to see an S3 Trigger in the overview.

The screenshot shows the 'Lambda-Func' function overview. A green message box at the top states: 'The trigger elasticbeanstalk-ap-south-1-529088256210 was successfully added to function Lambda-Func. The function is now receiving events from the trigger.' Below this, the 'Function overview' section shows the function name 'Lambda-Func' and its ARN: 'arn:aws:lambda:ap-south-1:529088256210:function:Lambda-Func'. It also shows the last modified time as '3 days ago' and the function URL. On the left, there are tabs for 'Diagram' (selected) and 'Template'. On the right, there is a sidebar titled 'Create a simple web app' with a 'Start tutorial' button.

Step 12: Now, create a dummy JSON file locally.

Step 13: Go back to your S3 Bucket and click on Add Files to upload a new file.

Step 14: Select the dummy data file from your computer and click Upload.

The screenshot shows the AWS S3 'Upload' interface. At the top, the navigation bar includes 'Services' and a search bar. Below the navigation, the path 'Amazon S3 > Buckets > advopssexp12 > Upload' is displayed. The main area is titled 'Upload' with a 'Info' link. A note at the top says: 'Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more.' Below this is a large dashed box with the placeholder text 'Drag and drop files and folders you want to upload here, or choose Add files or Add folder.' Underneath is a table titled 'Files and folders (1 Total, 89.0 B)' containing one item: 'dummy.json' (application/json, 89.0 B). There are 'Remove', 'Add files', and 'Add folder' buttons above the table. A search bar labeled 'Find by name' is present. The 'Destination' section shows 's3://advopssexp12'. At the bottom, there are 'CloudShell' and 'Feedback' links, and a copyright notice: '© 2023, Amazon Web Services India Private Limited or its affiliates.'

Step 15: After this make the necessary changes in the Test configuration file which we created it previously by replacing the Bucket Name and the ARN of Bucket.

The screenshot shows the AWS Lambda 'Event JSON' editor. The JSON code is as follows:

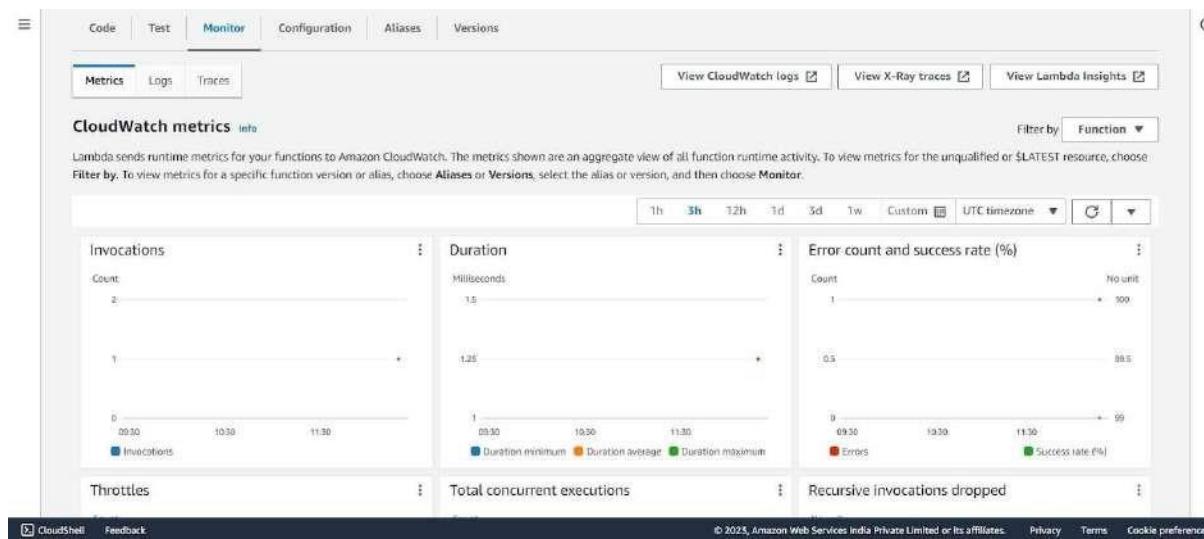
```

10     "principalId": "EXAMPLE"
11   },
12   "requestParameters": {
13     "sourceIPAddress": "127.0.0.1"
14   },
15   "responseElements": {
16     "x-amz-request-id": "EXAMPLE123456789",
17     "x-amz-id-2": "EXAMPLE123/5678abcdefghijklmnaqrstuvwxyzABCDEFGHIJKLMN"
18   },
19   "s3": {
20     "s3SchemaVersion": "1.0",
21     "configurationId": "testConfigRule",
22     "bucket": {
23       "name": "advopssexp12",
24       "ownerIdentity": {
25         "principalId": "EXAMPLE"
26       },
27       "arn": "arn:aws:s3:::advopssexp12"
28     },
29     "object": {
30       "key": "test%2Fkey",
31       "size": 1024,
32       "eTag": "0123456789abcdef0123456789abcdef",
33       "sequencer": "0A1B2C3D4E5F678901"
34     }
35   }
36 }
37 ]
38 }

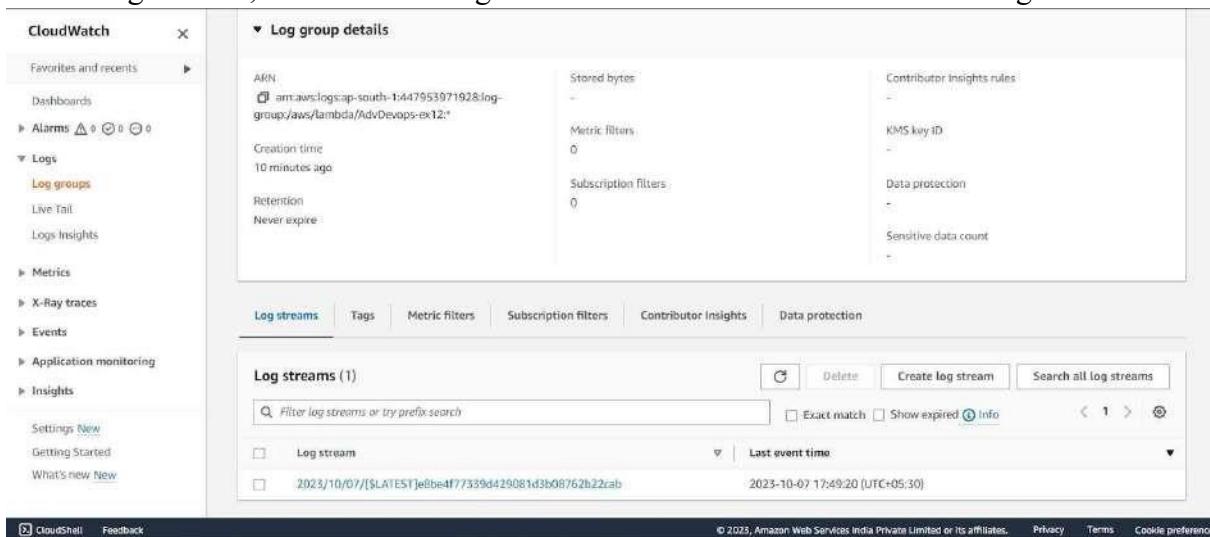
```

At the top right of the editor is a 'Format JSON' button.

Step 16: Go back to your Lambda function , Refresh it and check the Monitor tab.



Under Log streams, click on View logs in Cloudwatch to check the Function logs.



Step 17: Click on this log Stream that was created to view what was logged by your function.

The screenshot shows the AWS CloudWatch Logs interface. On the left, there's a navigation sidebar with options like Favorites and recent, Dashboards, Alarms, Logs (selected), Log groups, Log Anomalies, Live Tail, Logs Insights, Contributor Insights, Metrics, X-Ray traces, Events, Application Signals (New), Network monitoring, Insights, Settings, and Getting Started. At the bottom of the sidebar are CloudShell and Feedback links. The main area displays a log group path: CloudWatch > Log groups > /aws/lambda/Lambda-Func > 2024/10/11/[...LATEST]aff54e3f606143548ec12e1f2f25a4df. Below this, a section titled "Log events" shows a table with columns for "Timestamp" and "Message". The table contains several log entries, each starting with a timestamp (e.g., 2024-10-11T05:23:33.473Z) followed by a log message. The messages include INIT_START, START, END, and REPORT requests for a Lambda function. The interface includes a search bar, filter buttons (Clear, 1m, 30m, 1h, 12h, Custom, UTC timezone), and display settings (Display, Show more). At the bottom right, there are links for © 2024, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

Conclusion: Thus, we have created a Lambda function which logs “An Image has been added” once you add an object to a specific bucket in S3.