



PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology
(NBA Accredited)



Class / Branch: TE IT

Subject: Advanced Devops Lab (ADL)

Subject Lab Incharge: Prof. Manjusha Kashikar

Academic Year: 2023-24

Semester: V

EXPERIMENT NO. 01

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

Theory:

AWS Cloud9 is a cloud-based integrated development environment (IDE) that lets you write, run, and debug your code with just a browser. It includes a code editor, debugger, and terminal. Cloud9 comes prepackaged with essential tools for popular programming languages, including JavaScript, Python, PHP, and more, so you don't need to install files or configure your development machine to start new projects. Since your Cloud9 IDE is cloud-based, you can work on your projects from your office, home, or anywhere using an internet-connected machine. Cloud9 also provides a seamless experience for developing serverless applications enabling you to easily define resources, debug, and switch between local and remote execution of serverless applications. With Cloud9, you can quickly share your development environment with your team, enabling you to pair program and track each other's inputs in real time.

Benefits:

CODE WITH JUST A BROWSER

AWS Cloud9 gives you the flexibility to run your development environment on a managed Amazon EC2 instance or any existing Linux server that supports SSH. This means that you can write, run, and debug applications with just a browser, without needing to install or maintain a local IDE. The Cloud9 code editor and integrated debugger include helpful, time-saving features such as code hinting, code completion, and step-through debugging. The Cloud9 terminal provides a browser-based shell experience enabling you to install additional software, do a git push, or enter commands.

CODE TOGETHER IN REAL TIME

AWS Cloud9 makes collaborating on code easy. You can share your development environment with your team in just a few clicks and pair program together. While collaborating, your team members can see each other type in real time, and instantly chat with one another from within the IDE.



PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



BUILD SERVERLESS APPLICATIONS WITH EASE

AWS Cloud9 makes it easy to write, run, and debug serverless applications. It preconfigures the development environment with all the SDKs, libraries, and plug-ins needed for serverless development. Cloud9 also provides an environment for locally testing and debugging AWS Lambda functions. This allows you to iterate on your code directly, saving you time and improving the quality of your code.

DIRECT TERMINAL ACCESS TO AWS

AWS Cloud9 comes with a terminal that includes sudo privileges to the managed Amazon EC2 instance that is hosting your development environment and a preauthenticated AWS Command Line Interface. This makes it easy for you to quickly run commands and directly access AWS services

START NEW PROJECTS QUICKLY

AWS Cloud9 makes it easy for you to start new projects. Cloud9's development environment comes prepackaged with tooling for over 40 programming languages, including Node.js, JavaScript, Python, PHP, Ruby, Go, and C++. This enables you to start writing code for popular application stacks within minutes by eliminating the need to install or configure files, SDKs, and plug-ins for your development machine. Because Cloud9 is cloud-based, you can easily maintain multiple development environments to isolate your project's resources.

Steps:

1. Login with your AWS account.
2. Navigate to Cloud 9 service from Developer tools section as below:



The screenshot shows the AWS Management Console with the 'Services' dropdown open. Under 'Cloud9', the 'Cloud9' service is selected. Other services listed under 'Recently visited' include Route 53, Console Home, EC2, Certificate Manager, CloudWatch, Lambda, AWS Cost Explorer, Billing, IAM, CloudFormation, DynamoDB, AWS IoT, and VPC.

3. Click on Create Environment :

The screenshot shows the AWS Cloud9 service page. The main heading is 'AWS Cloud9' with the subtext 'A cloud IDE for writing, running, and debugging code'. Below this, a paragraph explains that AWS Cloud9 allows you to write, run, and debug your code with just a browser. It highlights immediate access to a rich code editor, integrated debugger, and built-in terminal with preconfigured AWS CLI. A call-to-action button 'Create environment' is prominently displayed.



PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY
Department of Information Technology
(NBA Accredited)



1. Provide name for the Environment (WebAppIDE) and click on next.

AWS Cloud9 > Environments > Create environment

Step 1
Name environment

Step 2
Configure settings

Step 3
Review

Name environment

Environment name and description

Name
The name needs to be unique per user. You can update it at any time in your environment settings.

Limit: 60 characters

Description - Optional
This will appear on your environment's card in your dashboard. You can update it at any time in your environment settings.

Limit: 200 characters

Cancel **Next step**



2. Keep all the Default settings as shown in below:

AWS Cloud9

Your environments Shared with you Account environments

How-to guide

AWS Cloud9 > Environments > Create environment

Step 1 Name environment Step 2 Configure settings Step 3 Review

Configure settings

Environment settings

Environment type Info Run your environment in a new EC2 instance or an existing server. With EC2 instances, you can connect directly through Secure Shell (SSH) or connect via AWS Systems Manager (without opening inbound ports).

Create a new EC2 instance for environment (direct access) Launch a new instance in this region that your environment can access directly via SSH.

Create a new no-ingress EC2 instance for environment (access via Systems Manager) Launch a new instance in this region that your environment can access through Systems Manager.

Create and run in remote server (SSH connection) Configure the secure connection to the remote server for your environment.

Instance type

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-sized web projects.

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

Other instance type Select an instance type: t3.nano

Platform

Amazon Linux 2 (recommended)

Amazon Linux AMI

Ubuntu Server 18.04 LTS

Cost-saving setting Choose a predetermined amount of time to auto-hibernate your environment and prevent unnecessary charges. We recommend a hibernation settings of half an hour of no activity to maximize savings.

After 30 minutes (default)

IAM role AWS Cloud9 creates a service-linked role for you. This allows AWS Cloud9 to call other AWS services on your behalf. You can delete the role from the AWS IAM console once you no longer have any AWS Cloud9 environments. [Learn more](#)

AWSServiceRoleForAWSCloud9

Network settings (advanced)

No tags associated with the resource.

Add new tag

You can add 50 more tags.

Cancel Previous step Next step



3. Review the Environment name and Settings and click on Create Environment:

The screenshot shows the 'Create environment' wizard in AWS Cloud9. On the left is a sidebar with links: 'Your environments', 'Shared with you', 'Account environments', and 'How-to guide'. The main area has a breadcrumb trail: 'AWS Cloud9 > Environments > Create environment'. It's divided into three tabs: 'Step 1 Name environment' (selected), 'Step 2 Configure settings' (disabled), and 'Step 3 Review' (disabled). The 'Review' tab displays the configuration details:

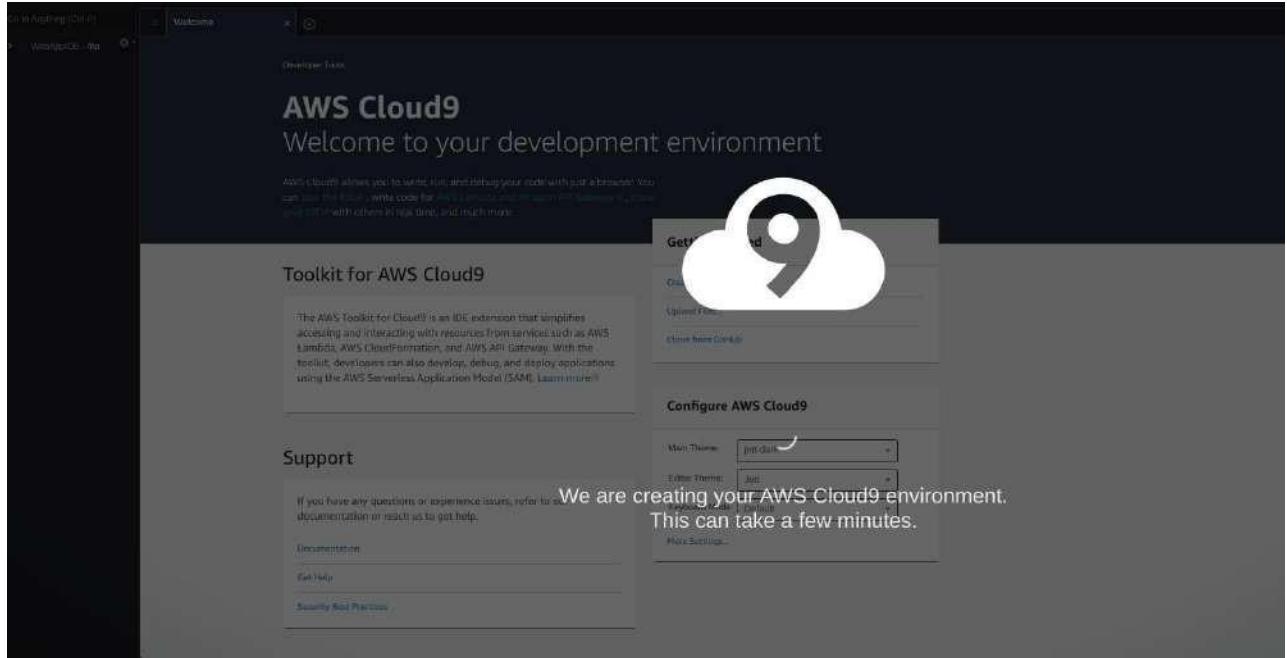
Environment name and settings	
Name	WebAppIDE
Description	No description provided
Environment type	EC2
Instance type	t2.micro
Subnet	
Platform	Amazon Linux 2 (recommended)
Cost-saving settings	After 30 minutes (default)
IAM role	AWSServiceRoleForAWSCloud9 (generated)

A callout box in the bottom right corner provides best practices:

We recommend the following best practices for using your AWS Cloud9 environment

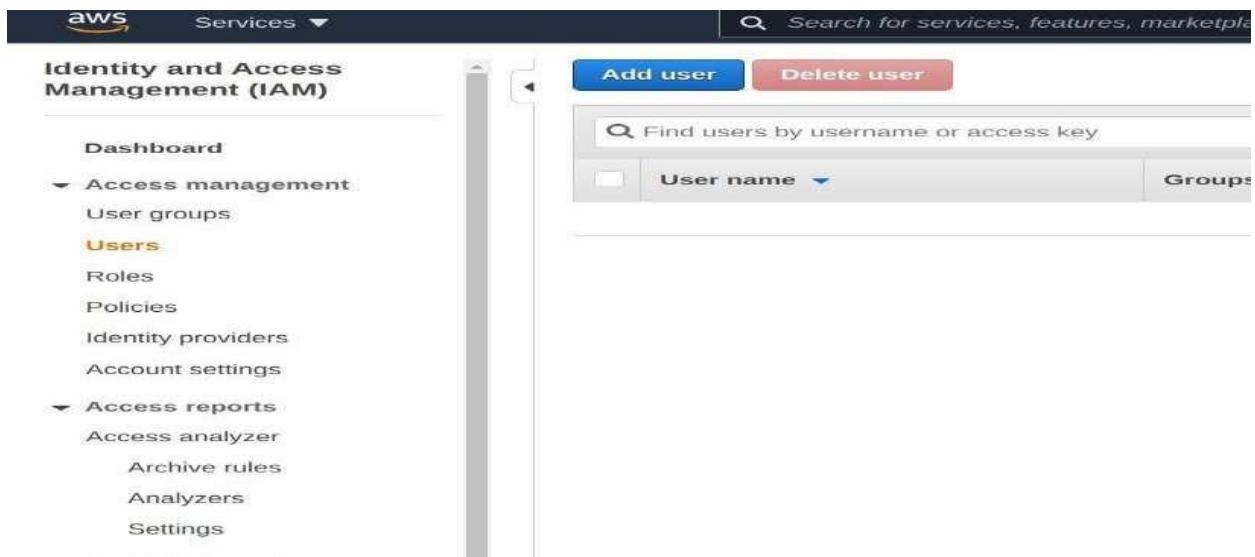
- Use source control and backup your environment frequently. AWS Cloud9 does not perform automatic backups.
- Perform regular updates of software on your environment. AWS Cloud9 does not perform automatic updates on your behalf.
- Turn on AWS CloudTrail in your AWS account to track activity in your environment. Learn more [\[link\]](#)
- Only share your environment with trusted users. Sharing your environment may put your AWS access credentials at risk. Learn more [\[link\]](#)

At the bottom are buttons for 'Cancel', 'Previous step', and a prominent orange 'Create environment' button.



It will take few minutes to create aws instance for your Cloud 9 Environment.

4. Till that time open IAM Identity and Access Management in order to Add user In other tab.





5. Add user provide manual password if you want and click on Next permission tab.

Add user

1 2 3 4 5

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name*

apsit

[+ Add another user](#)

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type*

Programmatic access

Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

AWS Management Console access

Enables a **password** that allows users to sign-in to the AWS Management Console.

Console password*

Autogenerated password

Custom password

Show password

Require password reset

User must create a new password at next sign-in

Users automatically get the **IAMUserChangePassword** policy to allow them to change their own password.

* Required

Cancel

Next: Permissions

6. Click on Create group

Add user

1

▼ Set permissions

[Add user to group](#)

[Copy permissions from existing user](#)

[Attach existing policies directly](#)



Get started with groups

You haven't created any groups yet. Using groups is a best-practice way to manage users' permissions by job access, or your custom permissions. Get started by creating a group. [Learn more](#)

[Create group](#)

► Set permissions boundary



7. Provide group name and click on create group.

Create group

Create a group and select the policies to be attached to the group. Using groups is a best-practice way to manage users' permissions by job functions, AWS service access, or your custom permissions. [Learn more](#)

Group name

[Create policy](#) [Refresh](#)

Filter policies [Search](#) Showing 669 results

	Policy name	Type	Used as	Description
<input type="checkbox"/>	AdministratorAccess	Job function	None	Provides full access to AWS services and resources.
<input type="checkbox"/>	AdministratorAccess-Amplify	AWS managed	None	Grants account administrative permissions while explicitly allowing direct access to resour...
<input type="checkbox"/>	AdministratorAccess-AWSElasticBeanst...	AWS managed	None	Grants account administrative permissions. Explicitly allows developers and administrators...
<input type="checkbox"/>	AlexaForBusinessDeviceSetup	AWS managed	None	Provide device setup access to AlexaForBusiness services
<input type="checkbox"/>	AlexaForBusinessFullAccess	AWS managed	None	Grants full access to AlexaForBusiness resources and access to related AWS Services
<input type="checkbox"/>	AlexaForBusinessGatewayExecution	AWS managed	None	Provide gateway execution access to AlexaForBusiness services
<input type="checkbox"/>	AlexaForBusinessLifesizeDelegatedAcc...	AWS managed	None	Provide access to Lifesize AVS devices

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

8. After that group is created click on next if u want to provide tag else click on Review for user settings and click on create user as shown in fig.

Add user

1 2 3 4 5

Review

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

User details

User name	apsit
AWS access type	AWS Management Console access - with a password
Console password type	Custom
Require password reset	No
Permissions boundary	Permissions boundary is not set

Permissions summary

The user shown above will be added to the following groups.

Type	Name
Group	WebAppsgroup

Tags

No tags were added.

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



9. Now close that window and Navigate to user Groups from left pane in IAM.

Group name	Users	Permissions	Creation time
WebAppapsitgroup	1	Not defined	4 minutes ago

10. click on your group name which you have created and navigate to permission tab as shown:

Policy Name	Type	Attached entities
No resources to display		



11. Now click on Add permission and select Attach Policy after that search for Cloud9 related policy and select Awscloud9EnviornmentMember policy and add it.

Other permission policies (Selected 1/669) [Info](#)
You can attach up to 10 managed policies to this user group. All of the users in this group inherit the attached permissions.

Filter policies by property or policy name and press enter 4 matches

"Cloud9" X Clear filters

Policy Name	Type	Attached entities
<input checked="" type="checkbox"/> AWSCloud9EnvironmentMember	AWS managed	0
<input type="checkbox"/> AWSCloud9Administrator	AWS managed	0
<input type="checkbox"/> AWSCloud9User	AWS managed	0
<input type="checkbox"/> AWSCloud9SSMInstanceProfile	AWS managed	0

Cancel Add permissions

12. now we move towards our cloud9 IDE Enviornment tab it shows as shown :



The screenshot shows the AWS Toolkit interface within a Cloud9 IDE. The AWS Explorer sidebar on the left lists services like API Gateway, CloudFormation stacks, ECR, Lambda functions, and S3. The main area displays a Lambda function named 'app.js' with its code:

```
13 // Return a JSON object with the message "Hello World"
14 const response = {
15   statusCode: 200,
16   body: 'Hello World',
17   headers: {
18     'Content-Type': 'application/json'
19   }
20 }
21 module.exports.handler = (event, context) => {
22   try {
23     // console.log('Received event');
24     // console.log(event);
25     // console.log(`Message : ${event.message}`);
26     // console.log(`Event : ${JSON.stringify(event)}`);
27     // console.log(`Context : ${JSON.stringify(context)}`);
28     // console.log(`Headers : ${JSON.stringify(headers)}`);
29     // console.log(`Body : ${JSON.stringify(body)}`);
30     // console.log(`Status Code : ${statusCode}`);
31     // console.log(`Headers : ${headers}`);
32     // console.log(`Body : ${body}`);
33     return response;
34   } catch (err) {
35     console.log(err);
36     return err;
37   }
38 }
39 module.exports = app;
```

Annotations highlight the AWS Explorer sidebar with a yellow arrow and the 'Current Credentials' button at the bottom right.

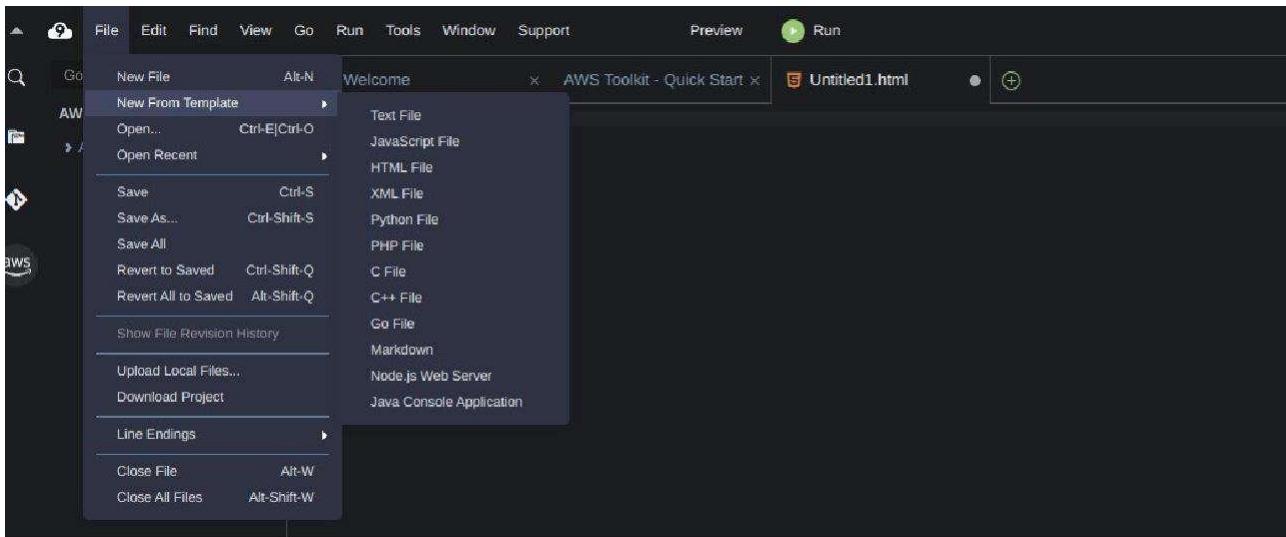
13. If you check at bottom side Cloud9 IDE also giving you and aws CLI for command operations: as we here checked git version, iam user details and so on...

The screenshot shows the AWS Toolkit interface within a Cloud9 IDE. The AWS Explorer sidebar on the left lists services like API Gateway, CloudFormation stacks, ECR, Lambda functions, and S3. The main area displays a terminal window with the following commands and output:

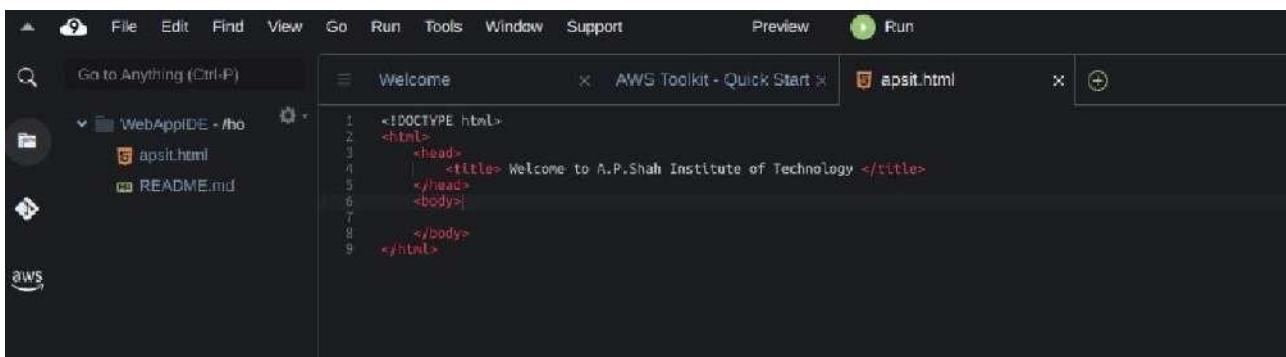
```
bash - "ip-172-31-10-50.a x" Immediate (Javascript (br x) +  
ec2-user:~/environment $ git --version  
ec2-user:~/environment $ git --version  
git version 2.23.4  
ec2-user:~/environment $ aws iam get-user  
{  
  "User": {  
    "PasswordLastUsed": "2021-07-07T05:34:24Z",  
    "CreateDate": "2021-06-03T18:03:54Z",  
    "UserId": "229296960472",  
    "Arn": "arn:aws:iam::229296960472:root"  
  }  
}
```



14. Now we will setup collaborative enviornment Click on File you can create new file or choose from template, here m opting html file to collaborate.

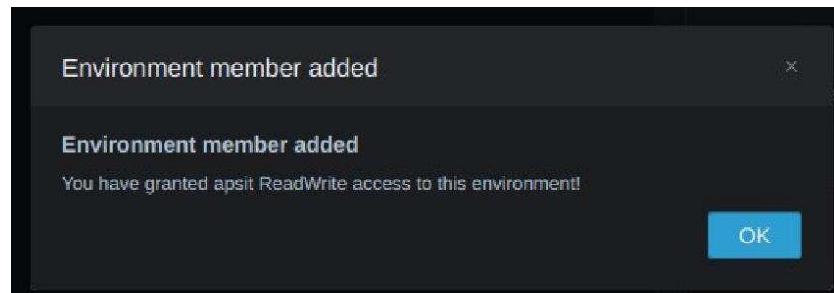
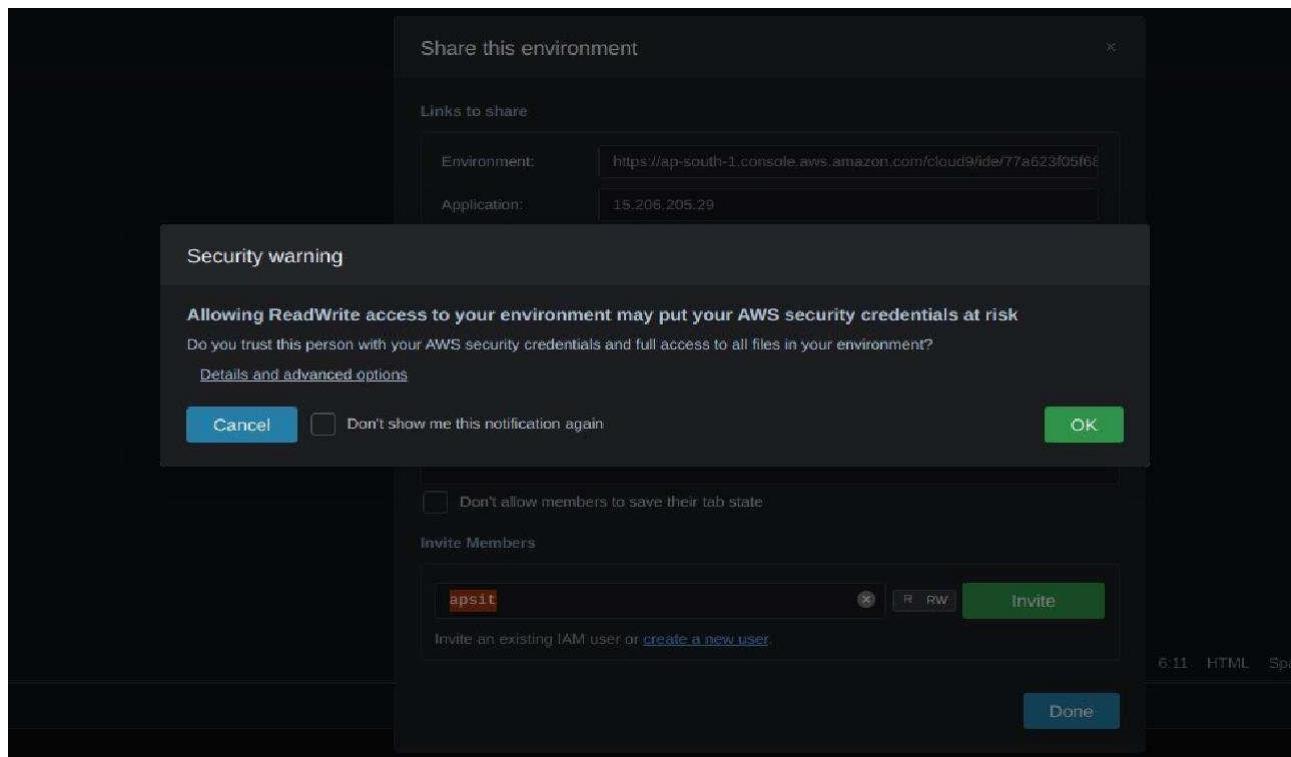


15. Edit html file and save it





16. now in order to share this file to collaborate with other members of your team click on Share option on Right Pane and username which you created in IAM before into Invite members and enable permission as RW (Read and Write) and click on Done. Click OK for Security warning.





17. Now Open your Browsers Incognito Window and login with IAM user which you configured before.

The screenshot shows the AWS Sign In page. At the top is the AWS logo. Below it is a 'Sign in' button. Two options are presented in boxes: 'Root user' (unchecked) and 'IAM user' (checked). The checked box has a blue outline. Below these boxes is a field labeled 'Account ID (12 digits) or account alias' containing the value '229296960472'. There is also a 'Remember this account' checkbox (unchecked). A large blue 'Next' button is at the bottom. Below the 'Next' button is a note about agreeing to the AWS Customer Agreement and Privacy Notice, followed by a 'New to AWS?' link and a 'Create a new AWS account' button.

aws

Sign in

Root user
Account owner that performs tasks requiring unrestricted access. [Learn more](#)

IAM user
User within an account that performs daily tasks. [Learn more](#)

Account ID (12 digits) or account alias
229296960472

Remember this account

Next

By continuing, you agree to the [AWS Customer Agreement](#) or other agreement for AWS services, and the [Privacy Notice](#). This site uses essential cookies. See our [Cookie Notice](#) for more information.

— New to AWS? —

Create a new AWS account



18. After Successful login with IAM user open Cloud9 service from dashboard services and click on shared with you enviornment to collaborate.

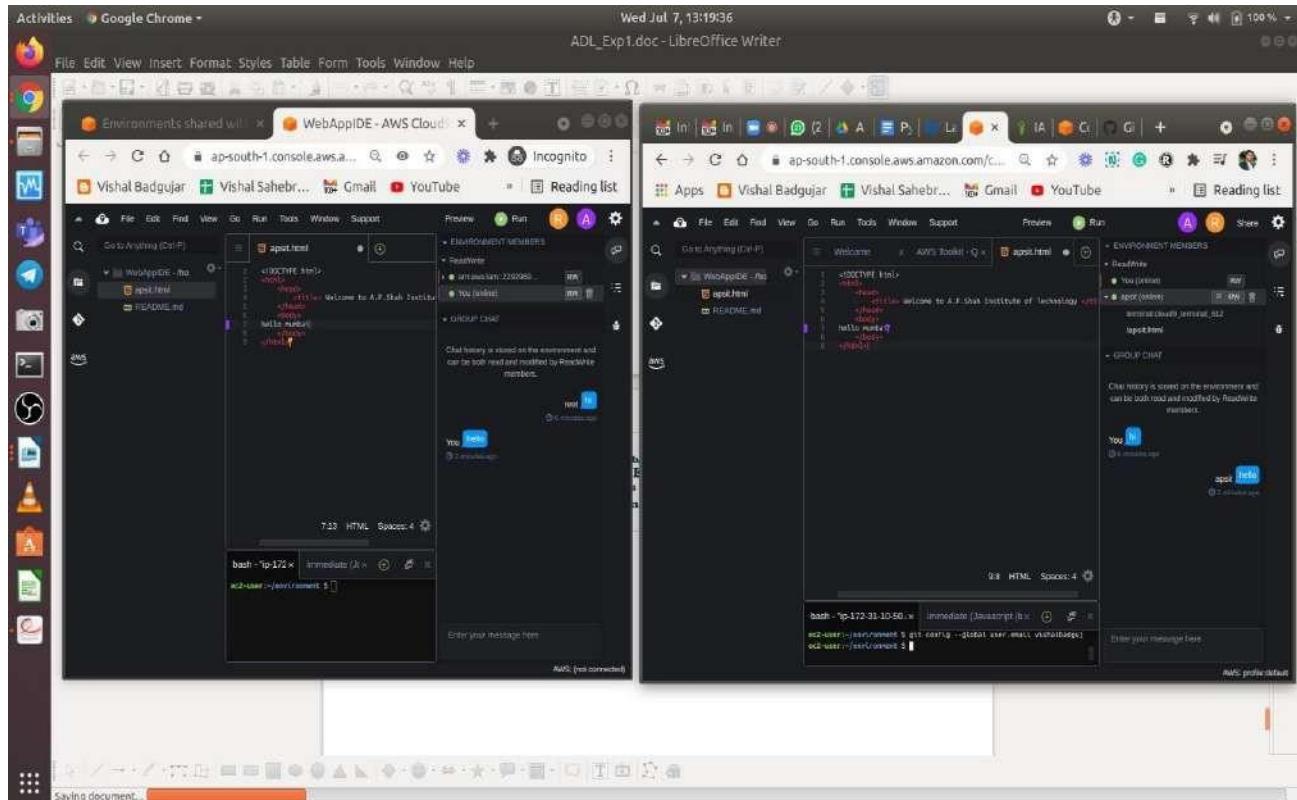
The screenshot shows the AWS Cloud9 interface. On the left, there's a sidebar with 'AWS Cloud9' at the top, followed by 'Your environments', 'Shared with you' (which is highlighted in orange), and 'Account environments'. Below that is a 'How-to guide'. The main area is titled 'AWS Cloud9 > Shared with you' and shows 'Shared with you (1)'. A card for 'WebAppIDE' is displayed, showing details: Type EC2, Permissions Read-write, Description No description available, and Owner Arn arn:aws:iam::229296960472:root. At the bottom of the card is a 'Open IDE' button.

19. Click on Open IDE you will same interface as your other member have to collaborate in real time, also you all within team can do group chats as shown below:

The screenshot shows the AWS Cloud9 IDE interface. The left side has a file tree with 'WebAppIDE' selected. The main area displays code in a terminal window, with the first few lines being: '#!/bin/bash -eu', 'echo "Welcome to A.P. Shah Institute of Technology"', and 'cd /var/www/html'. On the right, there are two panels: 'ENVIRONMENT MEMBERS' showing 'root' and 'You (root)' with a 'INVITE' button, and 'GROUP CHAT' which says 'Chat history is stored on the environment and can be both read and modified by free-form members'. The bottom status bar shows 'root ~ environment 1'.



PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY
Department of Information Technology
(NBA Accredited)



The screenshot displays a dual-terminal setup within the AWS Cloud9 IDE. Both terminals are running on the same host machine (ip-172-31-10-50). The left terminal shows the file structure of a directory containing 'apt.html', 'index.html', 'README.md', and 'hello-world'. The right terminal shows a similar directory structure. Both terminals have command-line interfaces with the user 'ec2-user' and the host 'ip-172-31-10-50'. The top of the screen shows the LibreOffice Writer application is open, and the status bar indicates the date and time as 'Wed Jul 7, 13:19:36'.



PARSHVANATH CHARITABLE TRUST'S

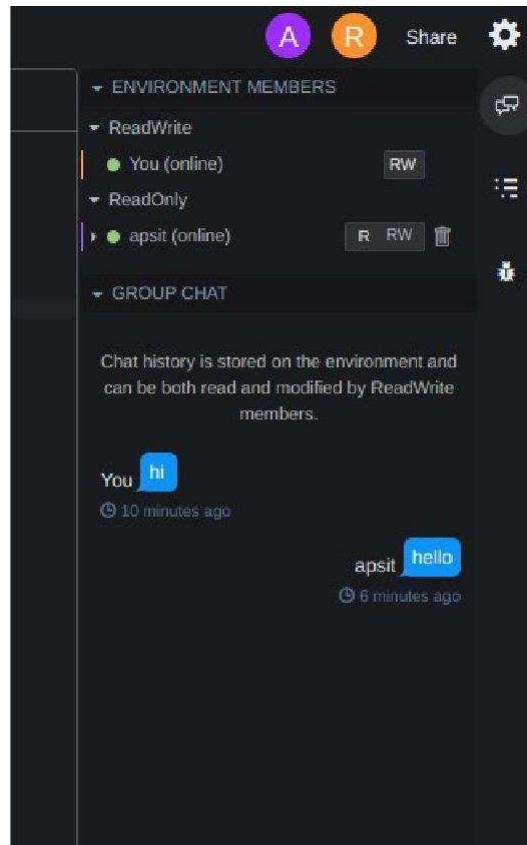
A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



24. you can also explore settings where you can update permissions of your temmates as from RW to R only or you can remove user too.



For more info related to AWS-Cloud 9 you all can refer following Docs.

<https://docs.aws.amazon.com/cloud9/latest/user-guide/aws-cloud9-ug.pdf>

Conclusion: Write your own findings.

**Academic Year: 2023-24****Semester: V****Class / Branch: TE IT****Subject: Advanced Devops Lab (ADL)****Subject Lab Incharge: Prof. Manjusha Kashikar/Prof.Sonal Jain/Prof.Yaminee Patil**

EXPERIMENT NO. 02

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.

Theory:

Continuous deployment allows you to deploy revisions to a production environment automatically without explicit approval from a developer, making the entire software release process automated.

You will create the pipeline using [AWS CodePipeline](#), a service that builds, tests, and deploys your code every time there is a code change. You will use your GitHub account, an Amazon Simple Storage Service (S3) bucket, or an AWS CodeCommit repository as the source location for the sample app's code. You will also use AWS Elastic Beanstalk as the deployment target for the sample app. Your completed pipeline will be able to detect changes made to the source repository containing the sample app and then automatically update your live sample app.

Step1: Create a deployment environment

Your continuous deployment pipeline will need a target environment containing virtual servers, or Amazon EC2 instances, where it will deploy sample code. You will prepare this environment before creating the pipeline.

1. To simplify the process of setting up and configuring EC2 instances for this tutorial, you will spin up a sample environment using AWS Elastic Beanstalk. Elastic Beanstalk lets you easily host web applications without needing to launch, configure, or operate virtual servers on your own. It automatically provisions and operates the infrastructure (e.g. virtual servers, load balancers, etc.) and provides the application stack (e.g. OS, language and framework, web and application



PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



server, etc.) for you.



Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY, THANE
(All Programs Accredited by NBA)
Department of Information Technology



2. Name your web app and choose PHP from the drop-down menu(or any other language you are interested in) and then click Create Application.

Compute

Amazon Elastic Beanstalk

End-to-end web application management.

Amazon Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.

Get started

Easily deploy your web application in minutes.

Create Application

Pricing



Elastic Beanstalk > Getting started

Create a web app

Create a new application and environment with a sample application or your own code. By creating an environment, you allow Amazon Elastic Beanstalk to manage Amazon Web Services resources and permissions on your behalf. [Learn more](#)

Application information

Application name

MyEBS

Up to 100 Unicode characters, not including forward slash (/).

Application tags

Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the resource and is case-sensitive. [Learn more](#)

Key

EBS

Value

CICD

[Remove tag](#)

[Add tag](#)

49 remaining





Platform

Platform: PHP

Platform branch: PHP 7.4 running on 64bit Amazon Linux 2

Platform version: 3.3.4 (Recommended)

Application code

Sample application
Get started right away with sample code.

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

[Cancel](#) [Configure more options](#) [Create application](#)

3. Elastic Beanstalk will begin creating a sample environment for you to deploy your application to. It will create an Amazon EC2 instance, a security group, an Auto Scaling group, an Amazon S3 bucket, Amazon CloudWatch alarms, and a domain name for your application.

Note: This will take several minutes to complete.

Step2: Get a copy of the sample code

In this step, you will retrieve a copy of the sample app's code and choose a source to host the code. The pipeline takes code from the source and then performs actions on it.

You can use one of three options as your source: a GitHub repository, an Amazon S3 bucket, or an AWS CodeCommit repository. Select your preference and follow the steps



below:

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1 Choose pipeline settings [Info](#)

Step 2 Add source stage

Step 3 Add build stage

Step 4 Add deploy stage

Step 5 Review

Pipeline settings

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

Service role

New service role Create a service role in your account

Existing service role Choose an existing service role from your account.

Role name
 Type your service role name

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

Advanced settings

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

- a. If you plan to use Amazon S3 as your source, you will retrieve the sample code from the AWS GitHub repository, save it to your computer, and upload it to an Amazon S3 bucket.

- Visit our GitHub repository containing the sample code at

<https://github.com/imoisharma/aws-codepipeline-s3-codedeploy-linux-2.0>

- Click the dist folder.



b. Save the source files to your computer:

- Click the file named aws-codepipeline-s3-aws-codedeploy_linux.zip
- Click View Raw.
- Save the sample file to your local computer.

c. open the Amazon S3 console and create your Amazon S3 bucket:

- Click Create Bucket
- Bucket Name: type a unique name for your bucket, such as awscodepipeline-demobucket- variables. All bucket names in Amazon S3 must be unique, so use one of your own, not one with the name shown in the example.
- Region: In the drop-down, select the region where you will create your pipeline, such as ap- South-1
- Click Create.

d. The console displays the newly created bucket, which is empty.



- Click Properties.
- Expand Versioning and select Enable Versioning. When versioning is enabled, Amazon S3 saves every version of every object in the bucket.

e. You will now upload the sample code to the Amazon S3 bucket:

- Click Upload.
- Follow the on-screen directions to upload the .zip file containing the sample code you downloaded from GitHub.





PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



Create bucket Info

Buckets are containers for data stored in S3. [Learn more](#)

General configuration

Bucket name

awscodepipeline-demobucket-variables1

Bucket name must be unique and must not contain spaces or uppercase letters. [See rules for bucket naming](#)

AWS Region

Asia Pacific (Mumbai) ap-south-1 ▾

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

[Choose bucket](#)

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



Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY, THANE
(All Programs Accredited by NBA)
Department of Information Technology





Amazon S3 > awscodepipeline-demobucket-variables11

awscodepipeline-demobucket-variables11 Info[Objects](#) [Properties](#) [Permissions](#) [Metrics](#) [Management](#) [Access Points](#)

Bucket overview

AWS Region
Asia Pacific (Mumbai) ap-south-1Amazon Resource Name (ARN)
`arn:aws:s3::awscodepipeline-demobucket-variables11`Creation date
August 2, 2021, 09:43:02 (UTC+05:30)

Bucket Versioning

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

Multi-factor authentication (MFA) delete

An additional layer of security that requires multi-factor authentication for changing Bucket Versioning settings and permanently deleting object versions. To modify MFA delete settings, use the AWS CLI, AWS SDK, or the Amazon S3 REST API. [Learn more](#)

you can upload directly zip file here from <https://github.com/imoisharma/aws-codepipeline- s3-codedeploy-linux-2.0>

Upload succeeded
View details below.

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

Summary	
Destination	s3://awscodepipeline-demobucket-variables11
Succeeded	<code>7 files, 12.2 KB (100.00%)</code>
Failed	<code>0 files, 0 B (0%)</code>

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

Files and folders (7 Total, 12.2 KB)

Name	Folder	Type	Size	Status
LICENSE	aws-codepipeline-s3-aws-codedeploy_linux/	-	10.6 KB	Successed
README.md	aws-codepipeline-s3-aws-codedeploy_linux/	text/markdown	249.0 B	Successed
appspec.yml	aws-codepipeline-s3-aws-codedeploy_linux/	application/x-yaml	359.0 B	Successed
index.html	aws-codepipeline-s3-aws-codedeploy_linux/	text/html	782.0 B	Successed
install_dependencies	aws-codepipeline-s3-aws-codedeploy_linux/scripts/	-	34.0 B	Successed
start_server	aws-codepipeline-s3-aws-codedeploy_linux/scripts/	-	33.0 B	Successed
stop_server	aws-codepipeline-s3-aws-codedeploy_linux/scripts/	-	105.0 B	Successed





Step3: Create your Pipeline

In this step, you will create and configure a simple pipeline with two actions: source and deploy. You will provide CodePipeline with the locations of your source repository and deployment environment.

A true continuous deployment pipeline requires a build stage, where code is compiled and unit tested. CodePipeline lets you plug your preferred build provider into your pipeline. However, in this we will skip the build stage.

Goto Pipeline again and create it

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1 Choose pipeline settings

Step 2 Add source stage

Step 3 Add build stage

Step 4 Add deploy stage

Step 5 Review

Add source stage Info

Source

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

Amazon S3

Bucket
codepipeline-ap-south-1-48704463255

S3 object key
s3://awscodepipeline-demobucket-variables11/aws-codepipeline-s3-aws-codedeploy.

Enter the object key. You can include a file path without the delimiter character (/) at the beginning. Include the file extension. Example: SampleApp.zip

Change detection options
Choose a detection mode to automatically start your pipeline when a change occurs in the source code.

Amazon CloudWatch Events (recommended)
Use Amazon CloudWatch Events to automatically start my pipeline when a change occurs

AWS CodePipeline
Use AWS CodePipeline to check periodically for changes

Cancel Previous Next

In above you can give zip file name in S3 object Key and choose bucket name which you created



In Step 4: Deploy Stage:

- Deployment provider: Click AWS Elastic Beanstalk.
- Application name: MYEBS.
- Environment name: Click Myebs-env.



Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1

[Choose pipeline settings](#)

Step 2

[Add source stage](#)

Step 3

[Add build stage](#)

Step 4

Add deploy stage

Step 5

Review

Add deploy stage Info



You cannot skip this stage

Pipelines must have at least two stages. Your second stage must be either a build or deployment stage. Choose a provider for either the build stage or deployment stage.

Deploy

Deploy provider

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

AWS Elastic Beanstalk

Region

Asia Pacific (Mumbai)

Application name

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

Q MyEBS X

Environment name

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

Q Myebs-env X

[Cancel](#)

[Previous](#)

[Next](#)



PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY
Department of Information Technology
(NBA Accredited)



The screenshot shows the AWS CodePipeline console. On the left, a sidebar lists pipeline stages: Source, Build, Deploy, Pipeline, and Settings. The main area displays two stages: 'Source' and 'Deploy'. The 'Source' stage is marked as 'Succeeded' with a green status icon. It shows a log message: 'Pipeline was saved successfully.' Below it, another message says: 'The most recent change will re-run through the pipeline. It might take a few moments for the status of the run to show in the pipeline view.' The 'Deploy' stage is also marked as 'Succeeded' with a green status icon. It shows a log message: 'Pipeline execution ID: 0a1f0e88-64e0-498e-ae02-72b865884a0b'. Below the stages, there is a button labeled 'Disable transition'.



Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY, THANE
(All Programs Accredited by NBA)
Department of Information Technology



Now go to your EBS environment and click on the URL to view the sample website you deployed.

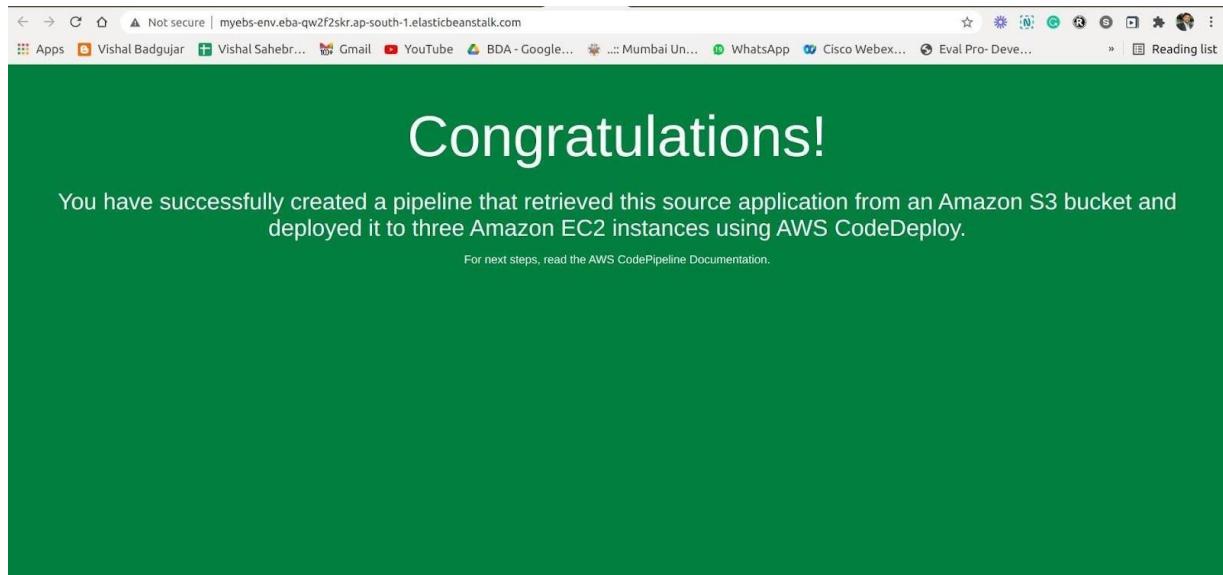
The screenshot shows the AWS Elastic Beanstalk console. On the left, a sidebar lists environments, applications, and configurations. The main area displays the 'MyEBS' environment details. The table shows one environment named 'Myebs-env' with the following details:

Environment name	Health	Date created	Last modified	URL	Running versions	Platform	Platform state	Tier name
Myebs-env	Ok	2021-08-02 09:30:03 UTC+0530	2021-08-02 10:11:20 UTC+0530	Myebs-env.eba-qw2fzskrap-south-1.elasticbeanstalk.com	code-pipeline-1627879215398-April0vY4ZworlP1vca1Tf2k5iTWFkxH	PHP 7.4 running on 64bit Amazon Linux 2	Supported	WebServer

You have successfully created an automated software release pipeline using AWS CodePipeline!



Using CodePipeline, you created a pipeline that uses GitHub, Amazon S3, or AWS CodeCommit as the source location for application code and then deploys the code to an Amazon EC2 instance managed by AWS Elastic Beanstalk.



Step 5: Commit a change and then update your app

In this step, you will revise the sample code and commit the change to your repository. CodePipeline will detect your updated sample code and then automatically initiate deploying it to your EC2 instance via Elastic Beanstalk.

Note that the sample web page you deployed refers to AWS CodeDeploy, a service that automates code deployments. In CodePipeline, CodeDeploy is an alternative to using Elastic Beanstalk for deployment actions. Let's update the sample code so that it correctly states that you deployed the sample using Elastic Beanstalk.



PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



- a. Visit your own copy of the repository that you forked in GitHub.
 - Open index.html
 - Select the Edit icon
- b. Update the webpage by copying and pasting the following text on line 30:
- c. Commit the change to your repository.
- d. Return to your pipeline in the CodePipeline console. In a few minutes, you should see the Source change to blue, indicating that the pipeline has detected the changes you made to your source repository. Once this occurs, it will automatically move the updated code to Elastic Beanstalk.
 - After the pipeline status displays Succeeded, in the status area for the Beta stage, click AWS Elastic Beanstalk.
- e. The AWS Elastic Beanstalk console opens with the details of the deployment. Select the environment you created earlier. And click the URL again from EBS environment again.



Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY, THANE
(All Programs Accredited by NBA)
Department of Information Technology





Congratulations!

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

For next steps, read the AWS CodePipeline Documentation.

Step 6: Clean up your resources

To avoid future charges, you will delete all the resources you launched throughout this tutorial, which includes the pipeline, the Elastic Beanstalk application, and the source you set up to host the code.

- a. First, you will delete your pipeline:

- In the pipeline view, click Edit.
- Click Delete.
- Type in the name of your pipeline and click Delete.

- b. Second, delete your Elastic Beanstalk application:



PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



- Visit the Elastic Beanstalk console.

- Click Actions.

- Then click Terminate Environment.



Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY, THANE
(All Programs Accredited by NBA)
Department of Information Technology



You have successfully created an automated software release pipeline using AWS CodePipeline! Using CodePipeline, you created a pipeline that uses GitHub, Amazon S3, or AWS CodeCommit as the source location for application code and then deploys the code to an Amazon EC2 instance managed by AWS Elastic Beanstalk. Your pipeline will automatically deploy your code every time there is a code change.

Conclusion: Write your own findings.



Academic Year: 2023-24

Semester: V

Class / Branch: TE IT

Subject: Advanced Devops Lab (ADL)

Subject Lab Incharge: Prof. Manjusha Kashikar/Prof.Sonal Jain/Prof.Yaminee Patil

EXPERIMENT NO. 3

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

Theory:

Prerequisites

Launch two or more Linux servers running Ubuntu 18.04 /20.04 on Virtual box OR Launch two or more EC2instances of Ubuntu 20.04 AMI free tier.

If using EC2 instances, connect to all instances using PUTTY (on Windows) or usingSSH (on Linux/Ubuntu).

Assign Unique Hostname for Each Server Node

```
$ sudo hostnamectl set-hostname master-node
```

Next, set a worker node hostname by entering the following on the worker server:

```
$ sudo hostnamectl set-hostname worker-node-01
```

Steps to Install Kubernetes on Ubuntu Set up Docker

Step 1 : Install Docker

Kubernetes requires an existing Docker installation. If you already have Docker installed, skip ahead to Step 2. If you do not have Kubernetes, install it by following these steps: Update the package list with the command:

```
$ sudo apt-get update
```

Next, install Docker with the command:

```
$ sudo apt-get install docker.io
```

Repeat the process on each server that will act as

a node. Check the installation (and version) by



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



entering the following:

```
$ docker --version
```

Step 2 : Start and Enable Docker

Set Docker to launch at boot by entering the following:

```
$ sudo systemctl enable docker
```

```
$ sudo systemctl status docker
```

```
$ sudo systemctl start docker
```

Install Kubernetes

Step 3 : Add Kubernetes Signing Key

(<https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/>)



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



(Please refer above link)

```
$ sudo apt-get update
```

```
$ sudo apt-get install -y apt-transport-https ca-certificates curl
```

```
$ sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg
https://packages.cloud.google.com/apt/doc/apt-key.gpg
```

```
$ echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/
kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list
```

```
$ sudo apt-get update
```

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

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

Allow the process to complete. Verify the installation with:

```
$ kubeadm version
```

Then repeat the previous command to install the signing keys. Repeat for each

server node.**Kubernetes Deployment**

Step 4 : Begin Kubernetes Deployment

```
$ sudo swapoff -a
```

Step 5 : Initialize Kubernetes on Master Node

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



If the kubeadm init command ran without error then ignore this part. If you receive this error "kubelet isn't running or healthy", then do the following.

Create file `daemon.json` in `/etc/docker/` and add following lines in the file.

```
{  
  "exec-opts": ["native.cgroupdriver=systemd"]  
}
```

And run the following commands.

Do this on both master and worker nodes.

```
Master Node x Worker Node 1 x Worker Node 2 x | + v - □ ×  
ubuntu@master-node:~$ sudo touch "/etc/docker/daemon.json"  
ubuntu@master-node:~$ sudo vim "/etc/docker/daemon.json"  
ubuntu@master-node:~$ sudo cat "/etc/docker/daemon.json"  
{  
  "exec-opts": ["native.cgroupdriver=systemd"]  
}  
ubuntu@master-node:~$ sudo systemctl daemon-reload  
ubuntu@master-node:~$ sudo systemctl restart docker  
ubuntu@master-node:~$ sudo systemctl restart kubelet  
ubuntu@master-node:~$
```

After this run `sudo kubeadm reset` command and then the `init` or `join` command.

```
$ sudo touch "/etc/docker/daemon.json"  
$ sudo nano "/etc/docker/daemon.json"  
$ sudo cat "/etc/docker/daemon.json"  
{  
  "exec-opts": ["native.cgroupdriver=systemd"]  
}  
$ sudo systemctl daemon-reload  
$ sudo systemctl restart docker  
$ sudo systemctl restart kubelet  
$ sudo kubeadm reset
```

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

Once this command finishes, it will display a kubeadm join message at the end. Make a note of the whole entry. This will be used to join the worker nodes to the cluster.



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



```
Master Node      Worker Node 1      Worker Node 2
[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable kubelet client certificate and key
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy

Your Kubernetes control-plane has initialized successfully!

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

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

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

export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
https://kubernetes.io/docs/concepts/cluster-administration/addons/

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

kubeadm join 172.31.47.231:6443 --token 5paymh.65rulhijchj8n1pt \
--discovery-token-ca-cert-hash sha256:af86e8e8e7c2a528eeda91ddb34b98edd4674fef01629ccdc5de3c8942bc5
18
ubuntu@master-node:~$
```

Copy this command in
Notepad for further use

Next, enter the following to create a directory for the cluster:

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

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

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

Step 6 : Deploy Pod Network to Cluster

A Pod Network is a way to allow communication between different nodes in the cluster. This tutorial uses the flannel virtual network.

Enter the following:

```
$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
```

Allow the process to complete.

Verify that everything is running and communicating:

```
$ kubectl get pods --all-namespaces
```

```
Master Node      Worker Node 1      Worker Node 2
ubuntu@master-node:~$ kubectl get pods --all-namespaces
NAMESPACE     NAME                               READY   STATUS    RESTARTS   AGE
kube-system   coredns-78fcfd69978-bglqk        1/1    Running   0          37m
kube-system   coredns-78fcfd69978-lfrhf        1/1    Running   0          37m
kube-system   etcd-master-node                 1/1    Running   0          37m
kube-system   kube-apiserver-master-node       1/1    Running   0          37m
kube-system   kube-controller-manager-master-node 1/1    Running   1 (10m ago) 37m
kube-system   kube-flannel-ds-f6zj8            1/1    Running   0          10m
kube-system   kube-proxy-n8lkt                1/1    Running   0          37m
kube-system   kube-scheduler-master-node       1/1    Running   1 (10m ago) 37m
ubuntu@master-node:~$
```



Step 7 : Join Worker Node to Cluster

Switch to the worker-node-01 system and enter the command you noted from Step 5: \$ kubeadm join -- discovery-token abcdef.1234567890abcdef --discoverytoken-ca-cert-hash sha256:1234..cdef 1.2.3.4:6443 Replace the alphanumeric codes with those from your master server.

Repeat for each worker node on the cluster.

Wait a few minutes; then you can check the status of the nodes.

If you are trying to run this on EC2 you'll get an error message saying less CPU and memory to override the error run the above command with

--ignore-preflight-errors=all

Switch to the master server, and enter:

```
$ kubectl get nodes
```

The system should display the worker nodes that you joined to the cluster.

Conclusion: Write your own findings.



Academic Year: 2023-24

Class / Branch: TE IT

Subject: Advanced Devops Lab (ADL)

Subject Lab Incharge: Prof. Manjusha Kashikar

Semester: V

EXPERIMENT NO.4

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

Theory:

Prerequisites

Launch two or more Linux servers running Ubuntu 18.04 /20.04 on Virtual box OR Launch two or more EC2instances of Ubuntu 20.04 AMI free tier.

If using EC2 instances, connect to all instances using PUTTY (on Windows) or usingSSH (on Linux/Ubuntu).

Assign Unique Hostname for Each Server Node

```
$ sudo hostnamectl set-hostname master-node
```

Next, set a worker node hostname by entering the following on the worker server:

```
$ sudo hostnamectl set-hostname worker-node-01
```

Steps to Install Kubernetes on Ubuntu

Set up Docker

Step 1 : Install Docker

Kubernetes requires an existing Docker installation. If you already have Docker installed, skip ahead to Step 2. If you do not have Kubernetes, install it by following these steps: Update the package list with the command:

```
$ sudo apt-get update
```

Next, install Docker with the command:

```
$ sudo apt-get install docker.io
```

Repeat the process on each server that will act as a node. Check

the installation (and version) by entering the following:

```
$ docker --version
```

Step 2 : Start and Enable Docker

Set Docker to launch at boot by entering the following:



```
$ sudo systemctl enable docker
```

```
$ sudo systemctl status docker
```

```
$ sudo systemctl start docker
```

Install Kubernetes

Step 3 : Add Kubernetes Signing Key

(<https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/>)

```
$ sudo apt-get update
```

```
$ sudo apt-get install -y apt-transport-https ca-certificates curl
```

```
$ sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg
https://packages.cloud.google.com/apt/doc/apt-key.gpg
```

```
$ echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/
kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list
```

```
$ sudo apt-get update
```

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

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

Allow the process to complete. Verify the installation with:

```
$ kubeadm version
```



```
$ echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list
```

```
$ sudo apt-get update
```

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

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

Allow the process to complete. Verify the installation with:

```
$ kubeadm version
```

Then repeat the previous command to install the signing keys. Repeat for each server node.

Kubernetes Deployment

Step 4 : Begin Kubernetes Deployment

```
$ sudo swapoff -a
```

Step 5 : Initialize Kubernetes on Master Node

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



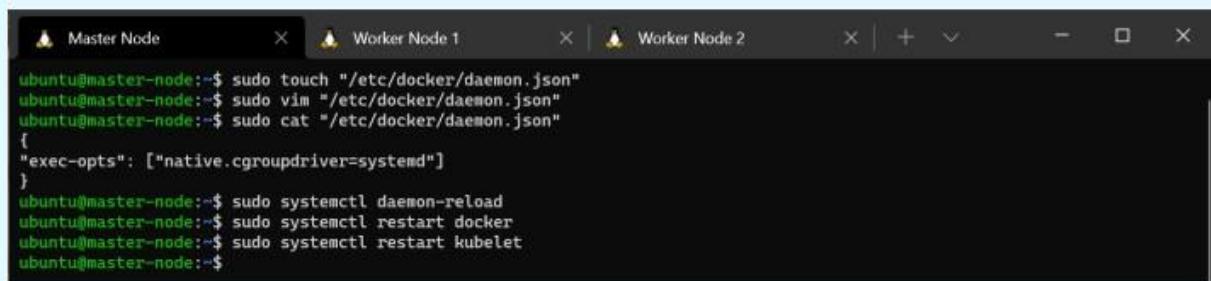
If the kubeadm init command ran without error then ignore this part. If you receive this error “kubelet isn't running or healthy”, then do the following.

Create file `daemon.json` in `/etc/docker/` and add following lines in the file.

```
{  
    "exec-opts": ["native.cgroupdriver=systemd"]  
}
```

And run the following commands.

Do this on both master and worker nodes.



```
Master Node      Worker Node 1      Worker Node 2      Worker Node 2  
ubuntu@master-node:~$ sudo touch "/etc/docker/daemon.json"  
ubuntu@master-node:~$ sudo vim "/etc/docker/daemon.json"  
ubuntu@master-node:~$ sudo cat "/etc/docker/daemon.json"  
{  
    "exec-opts": ["native.cgroupdriver=systemd"]  
}  
ubuntu@master-node:~$ sudo systemctl daemon-reload  
ubuntu@master-node:~$ sudo systemctl restart docker  
ubuntu@master-node:~$ sudo systemctl restart kubelet  
ubuntu@master-node:~$
```

After this run `sudo kubeadm reset` command and then the `init` or `join` command.

```
$ sudo touch "/etc/docker/daemon.json"  
$ sudo nano "/etc/docker/daemon.json"  
$ sudo cat "/etc/docker/daemon.json"  
{  
    "exec-opts": ["native.cgroupdriver=systemd"]  
}  
$ sudo systemctl daemon-reload  
$ sudo systemctl restart docker  
$ sudo systemctl restart kubelet  
$ sudo kubeadm reset
```

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

Once this command finishes, it will display a kubeadm join message at the end. Make a note of the whole entry. This will be used to join the worker nodes to the cluster.



```
Master Node X Worker Node 1 X | Worker Node 2 X + V - □ ×

[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable kubelet client certificate and key
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy

Your Kubernetes control-plane has initialized successfully!

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

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

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

export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
https://kubernetes.io/docs/concepts/cluster-administration/addons/

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

kubeadm join 172.31.47.231:6443 --token 5paymh.65rulhijchj8n1pt \
    --discovery-token-ca-cert-hash sha256:af86e8e8e7c2a528eeda91ddb34b98edd4674fef01629ccdc5de3c8942bc5
18
ubuntu@master-node:~$
```

Copy this command in
Notepad for further use

Next, enter the following to create a directory for the cluster:

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

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

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

Step 6 : Deploy Pod Network to Cluster

A Pod Network is a way to allow communication between different nodes in the cluster. This tutorial uses the flannel virtual network.

Enter the following:

```
$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-
```

flannel.yml Allow the process to complete.

Verify that everything is running and communicating:

```
$ kubectl get pods --all-namespaces
```

```
Master Node X Worker Node 1 X | Worker Node 2 X + V - □ ×

ubuntu@master-node:~$ kubectl get pods --all-namespaces
NAMESPACE     NAME           READY   STATUS    RESTARTS   AGE
kube-system   coredns-78fcfd69978-bglqk   1/1     Running   0          37m
kube-system   coredns-78fcfd69978-lfrhf   1/1     Running   0          37m
kube-system   etcd-master-node          1/1     Running   0          37m
kube-system   kube-apiserver-master-node  1/1     Running   0          37m
kube-system   kube-controller-manager-master-node  1/1     Running   1 (10m ago) 37m
kube-system   kube-flannel-ds-f6zj8      1/1     Running   0          10m
kube-system   kube-proxy-n8lkt          1/1     Running   0          37m
kube-system   kube-scheduler-master-node 1/1     Running   1 (10m ago) 37m
ubuntu@master-node:~$
```



Step 7 : Join Worker Node to Cluster

Switch to the worker-node-01 system and enter the command you noted from

Step 5: \$ kubeadm join -- discovery-token abcdef.1234567890abcdef --discoverytoken-ca-cert-hash

sha256:1234..cdef 1.2.3.4:6443Replace the alphanumeric codes with those from your master server.

Repeat for each worker node on the cluster.

Wait a few minutes; then you can check the status of the nodes.

If you are trying to run this on EC2 you'll get an error message saying less CPU and memory to override the error run the above command with

--ignore-preflight-errors=all

Switch to the master server, and enter:

```
$ kubectl get nodes
```

The system should display the worker nodes that you joined to the cluster.

Conclusion -Hence we have studied how to install Kubectl and execute Kubectl commands to manage the Kubernetes cluster and deploy Your First Kubernetes Application.



EXPERIMENT NO. 03

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

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

STEP 1:

Check security group, delete all SG only keep default

Name	Security group ID	Security group name	VPC ID	Description
-	sg-0a0885e28251a58dd	default	vpc-0c39a5e973216c8e0	default VPC security group

Create 2 instance



Create key pair

Summary

Create key pair

Key pair name
Key pairs allow you to connect to your instance securely.

Enter key pair name

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

Key pair type

RSA RSA encrypted private and public key pair

ED25519 ED25519 encrypted private and public key pair

Private key file format

.pem For use with OpenSSH

.ppk For use with PuTTY

⚠ When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)

Cancel **Create key pair**

Create security group and allow traffic

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

Network [Info](#)
vpc-0c39a5e973216c8e0

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-1**' with the following rules:

Allow SSH traffic from Anywhere
Helps you connect to your instance
0.0.0.0/0

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

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

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



Launch Instance

The screenshot shows the AWS EC2 Instances page. At the top, there is a breadcrumb navigation: EC2 > Instances > Launch an instance. Below this, a large blue progress bar indicates the status of the instance launch. The bar has the text "Launching instance" and "Launch initiation" at the top left, and "79%" at the bottom right. Below the progress bar, there is a "Details" link and a message: "Please wait while we launch your instance. Do not close your browser while this is loading."

Check security group of both instances

The screenshot shows the AWS EC2 Instances list. The sidebar on the left includes links for EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, and Network & Security. The main area displays a table titled "Instances (2) Info". The table has columns: Availability Zone, Public IPv4 DNS, Public IPv4 ..., Elastic IP, IPv6 IPs, Monitoring, Security group name, Key name, and Laun. Two instances are listed: one in us-east-1c with Public IPv4 DNS ec2-44-202-67-108.co... and another in us-east-1c with Public IPv4 DNS ec2-52-87-162-37.com.... Both instances are in the "running" state. Below the table, a modal window titled "Select an instance" is open.

Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs	Monitoring	Security group name	Key name	Laun
us-east-1c	ec2-44-202-67-108.co...	44.202.67.108	-	-	disabled	launch-wizard-1	exp3	2024
us-east-1c	ec2-52-87-162-37.com...	52.87.162.37	-	-	disabled	launch-wizard-1	exp3	2024



PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY
Department of Information Technology
(NBA Accredited)



Here security group is launch wizard-1

Now go to security group from left pane

Click on Security group id of Launch wizard-1

Name	Security group ID	Security group name	VPC ID	Description
-	sg-040c5ca4479b41563	launch-wizard-1	vpc-0c39a5e973216c8e0	launch-wizard-1 created 2024-07-22T08:10:53.297Z
-	sg-0a0885e28251a58dd	default	vpc-0c39a5e973216c8e0	default VPC security group

Details
Security group name: launch-wizard-1
Security group ID: sg-040c5ca4479b41563
Description: launch-wizard-1 created 2024-07-22T08:10:53.297Z
VPC ID: vpc-0c39a5e973216c8e0
Owner: 058264461049
Inbound rules count: 3 Permission entries
Outbound rules count: 1 Permission entry

Inbound rules (3)				
Name: sgr-00924faec7e1fce4	IP version: IPv4	Type: SSH	Protocol: TCP	Port range: 22
Name: sgr-0c575d4af8c53a204	IP version: IPv4	Type: HTTP	Protocol: TCP	Port range: 80



PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY
Department of Information Technology
(NBA Accredited)



Screenshot of the AWS Cloud9 interface showing the details of a security group named 'launch-wizard-1'. The 'Inbound rules' tab is selected, displaying three rules:

Name	Security group rule ID	IP version	Type	Protocol	Port range
-	sgr-00924faec7e1fce4	IPv4	SSH	TCP	22
-	sgr-0c575d4af8c53a204	IPv4	HTTP	TCP	80
-	sgr-0a9545571076a5a10	IPv4	HTTPS	TCP	443

Edit inbound rule

Screenshot of the AWS Cloud9 interface showing the 'Edit inbound rules' page for the security group 'sg-040c5ca4479b41563 - launch-wizard-1'. The page displays the three existing inbound rules and provides an 'Add rule' button.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-00924faec7e1fce4	SSH	TCP	22	Custom	0.0.0.0/0
sgr-0c575d4af8c53a204	HTTP	TCP	80	Custom	0.0.0.0/0
sgr-0a9545571076a5a10	HTTPS	TCP	443	Custom	0.0.0.0/0



Delete all rules

EC2 > Security Groups > sg-040c5ca4479b41563 - launch-wizard-1 > Edit inbound rules

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

This security group has no inbound rules.

[Add rule](#)

[Cancel](#)

[Preview changes](#)

Save rules

Add new rule

Select

ALL traffic

Anywhere IPV4

EC2 > Security Groups > sg-040c5ca4479b41563 - launch-wizard-1 > Edit inbound rules

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>
------------------------	--------------------------	------------------------------	--------------------------------	----------------------------	--

-

All traffic

All

All

Anywhe... ▾

Search

0.0.0.0/0 X

Delete

[Add rule](#)

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

X

[Cancel](#)

[Preview changes](#)

Save rules



PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY
Department of Information Technology
(NBA Accredited)



Now save rules

EC2 > Security Groups > sg-040c5ca4479b41563 - launch-wizard-1 > Edit inbound rules: Processing

Edit inbound rules: Processing

Modifying your security group

Revoke 0%

Details

Name the instances

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
<input type="checkbox"/> Master	i-095a903dc278f53de	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1c	ec2-44-202-67-108.
<input checked="" type="checkbox"/> Worker-node	i-046bea7b3a7423e7a	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1c	ec2-52-87-162-37.co...

Select Master and connect

EC2 Dashboard X Instances (1/2) Info

Find Instance by attribute or tag (case-sensitive)

All states

Instance state = running

Clear filters

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
<input checked="" type="checkbox"/> Master	i-095a903dc278f53de	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1c	ec2-44-202-67-108.co...
<input type="checkbox"/> Worker-node	i-046bea7b3a7423e7a	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1c	ec2-52-87-162-37.com...



Click on connect

[EC2](#) > [Instances](#) > [i-095a903dc278f53de](#) > Connect to instance

Connect to instance [Info](#)

Connect to your instance i-095a903dc278f53de (Master) using any of these options

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

Instance ID
 [i-095a903dc278f53de \(Master\)](#)

Connection Type

[Connect using EC2 Instance Connect](#)
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.

[Connect using EC2 Instance Connect Endpoint](#)
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address
 44.202.67.108

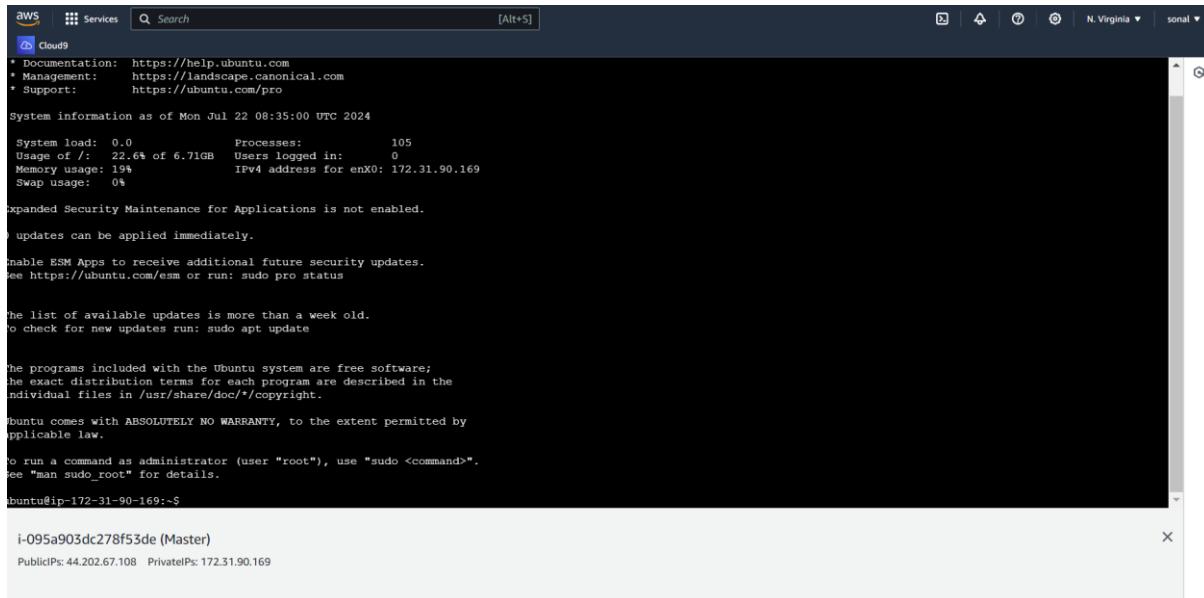
Username
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ubuntu.

ubuntu X

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



After Connecting



```
aws Services Search [Alt+S] Cloud9
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro

System information as of Mon Jul 22 08:35:00 UTC 2024
System load: 0.0      Processes:          105
Usage of /: 22.6% of 6.71GB  Users logged in:    0
Memory usage: 19%           IPv4 address for enX0: 172.31.90.169
Swap usage: 0%           Swap usage: 0% of 0B

expanded Security Maintenance for Applications is not enabled.
No updates can be applied immediately.

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

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

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

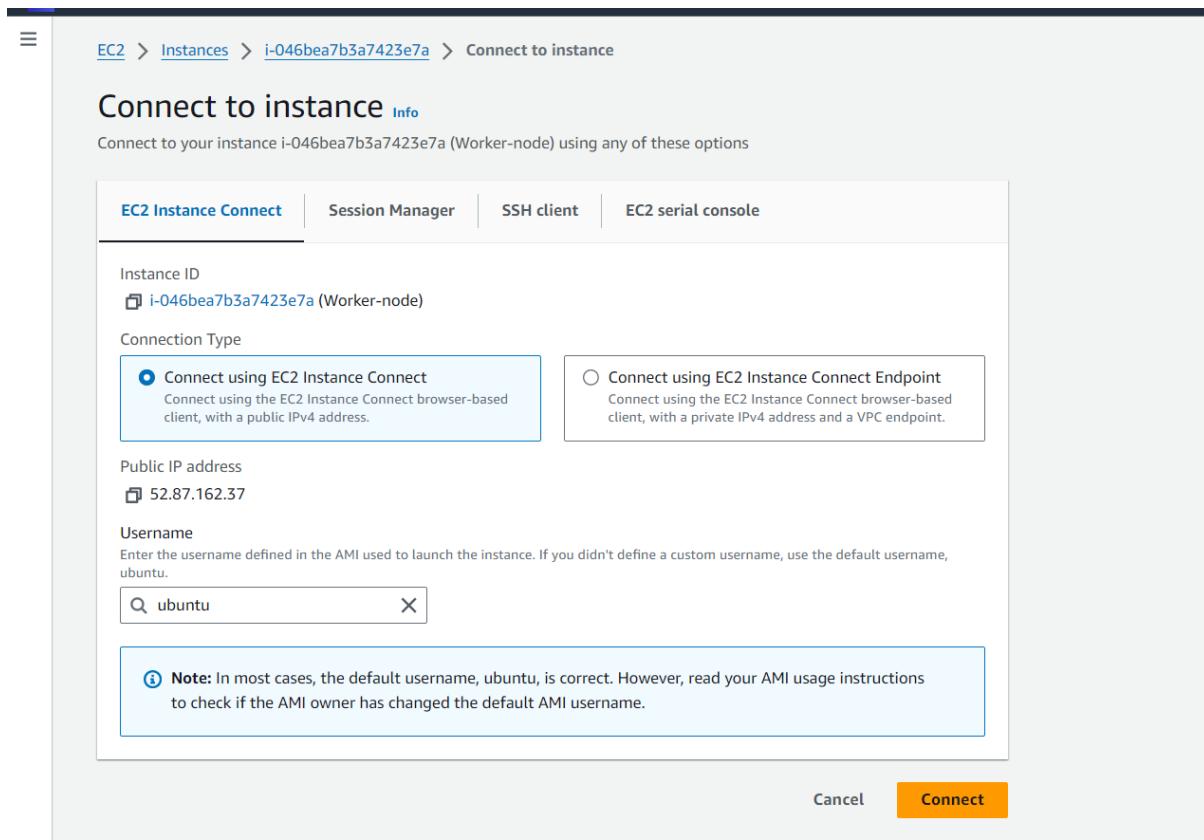
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

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

ubuntu@ip-172-31-90-169:~$
```

i-095a903dc278f53de (Master)
PublicIPs: 44.202.67.108 PrivateIPs: 172.31.90.169

Sameway connect to worker-node



EC2 Instances i-046bea7b3a7423e7a Connect to instance

Connect to instance Info

Connect to your instance i-046bea7b3a7423e7a (Worker-node) using any of these options

EC2 Instance Connect Session Manager SSH client EC2 serial console

Instance ID

Connection Type

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

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

Public IP address

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

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

Cancel



Step 2:

Assign Unique Hostname for Each Server Node

\$ sudo hostnamectl set-hostname master-node

Than **exit**

Refresh

```
System information as of Mon Jul 22 08:36:43 UTC 2024
System load:  0.08      Processes:          105
Usage of '/': 22.6% of 6.71GB  Users logged in:    0
Memory usage: 20%
Swap usage:   0%
Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.

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

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

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

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

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

ubuntu@ip-172-31-88-142:~$ sudo hostnamectl set-hostname master-node
ubuntu@ip-172-31-88-142:~$ exit
logout
[1]
```

Next, set a worker node hostname by entering the following on the worker server:

\$ sudo hostnamectl set-hostname worker1

STEP 3:

On both master and worker1

\$ sudo apt-get update

STEP 4:

On both master and worker1



Install docker

```
sudo apt-get install docker.io
```

STEP 5 : Start and Enable Docker

Set Docker to launch at boot by entering the following:

```
$ sudo systemctl enable docker
```

```
$ sudo systemctl status docker
```

```
ubuntu@worker1:~$ sudo systemctl enable docker
ubuntu@worker1:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
  Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
  Active: active (running) since Mon 2024-07-22 08:50:12 UTC; 1min 27s ago
TriggeredBy: • docker.socket
    Docs: https://docs.docker.com
   Main PID: 3121 (dockerd)
      Tasks: 8
     Memory: 32.8M (peak: 33.0M)
        CPU: 291ms
       CGroup: /system.slice/docker.service
               └─3121 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Jul 22 08:50:11 worker1 systemd[1]: Starting docker.service - Docker Application Container Engine...
Jul 22 08:50:11 worker1 dockerd[3121]: time="2024-07-22T08:50:11.603295601Z" level=info msg="Starting up"
Jul 22 08:50:11 worker1 dockerd[3121]: time="2024-07-22T08:50:11.605367608Z" level=info msg="detected 127.0.0.53 nameserver, assuming systemd-resolved, so using resolv.conf"
Jul 22 08:50:11 worker1 dockerd[3121]: time="2024-07-22T08:50:11.756581678Z" level=info msg="Loading containers: start."
Jul 22 08:50:12 worker1 dockerd[3121]: time="2024-07-22T08:50:12.266084496Z" level=info msg="Loading containers: done."
Jul 22 08:50:12 worker1 dockerd[3121]: time="2024-07-22T08:50:12.353942750Z" level=info msg="Docker daemon" commit="24.0.7-0ubuntu4" graphdriver="overlay2" version="24.0.7"
Jul 22 08:50:12 worker1 dockerd[3121]: time="2024-07-22T08:50:12.354065631Z" level=info msg="Daemon has completed initialization"
Jul 22 08:50:12 worker1 systemd[1]: Started docker.service - Docker Application Container Engine.
lines 1-21/21 (END)
```

Ctrl+c

Clear

```
sudo systemctl start docker
```

STEP 6 Install Kubernetes

<https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/>



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203).
ca-certificates set to manually installed.
curl is already the newest version (8.5.0-2ubuntu10.1).
curl 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
0 upgraded, 1 newly installed, 0 to remove and 22 not upgraded.
Need to get 3974 B of archives.
After this operation, 35.8 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 apt-transport-https all 2.7.14build2 [3974 B]
Fetched 3974 B in 0s (268 kB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 68104 files and directories currently installed.)
Preparing to unpack .../apt-transport-https 2.7.14build2_all.deb ...
Unpacking apt-transport-https (2.7.14build2) ...
Setting up apt-transport-https (2.7.14build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

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

Signining key

```
ubuntu@worker1:~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.30/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
ubuntu@worker1:~$ []
```

```
Cloud9
Preparing to unpack .../5-kubernetes-cni_1.4.0-1.1_amd64.deb ...
Unpacking kubernetes-cni (1.4.0-1.1) ...
Selecting previously unselected package socat.
Preparing to unpack .../6-socat 1.8.0.0-4build3_amd64.deb ...
Unpacking socat (1.8.0.0-4build3) ...
Selecting previously unselected package kubelet.
Preparing to unpack .../7-kubelet 1.30.3-1.1_amd64.deb ...
Unpacking kubelet (1.30.3-1.1) ...
Setting up cni-track (1:1.4.8-lubuntul) ...
Setting up kubectl (1.30.3-1.1) ...
Setting up ebttables (2.0.11-6build1) ...
Setting up socat (1.8.0.0-4build3) ...
Setting up cri-tools (1.30.0-1.1) ...
Setting up kubernetes-cni (1.4.0-1.1) ...
Setting up kubeadm (1.30.3-1.1) ...
Setting up kubelet (1.30.3-1.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@worker1:~$ sudo apt-mark hold kubelet kubeadm kubectl
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
```



PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology
(NBA Accredited)



Step 7 : Begin Kubernetes Deployment

\$ sudo swapoff -a

```
ubuntu@worker1:~$ sudo swapoff -a
ubuntu@worker1:~$ 
```

STEP 8:

Initialize Kubernetes on Master Node

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

If getting error

Run below code on both



PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



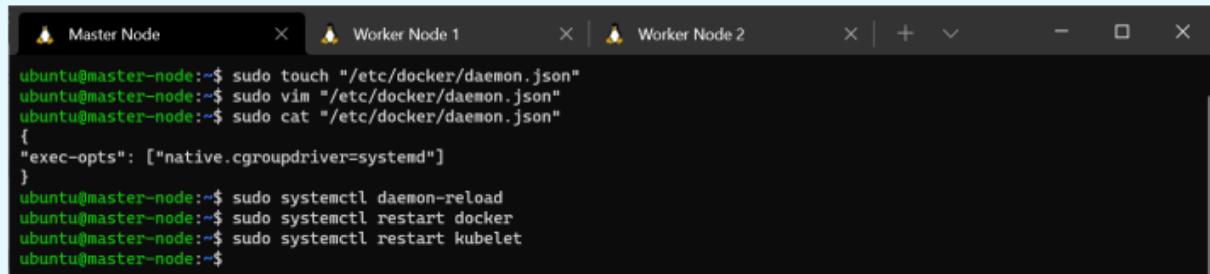
If the kubeadm init command ran without error then ignore this part. If you receive this error "kubelet isn't running or healthy", then do the following.

Create file `daemon.json` in `/etc/docker/` and add following lines in the file.

```
{  
  "exec-opts": ["native.cgroupdriver=systemd"]  
}
```

And run the following commands.

Do this on both master and worker nodes.



```
Master Node      Worker Node 1      Worker Node 2      + - ×  
ubuntu@master-node:~$ sudo touch "/etc/docker/daemon.json"  
ubuntu@master-node:~$ sudo vim "/etc/docker/daemon.json"  
ubuntu@master-node:~$ sudo cat "/etc/docker/daemon.json"  
{  
  "exec-opts": ["native.cgroupdriver=systemd"]  
}  
ubuntu@master-node:~$ sudo systemctl daemon-reload  
ubuntu@master-node:~$ sudo systemctl restart docker  
ubuntu@master-node:~$ sudo systemctl restart kubelet  
ubuntu@master-node:~$
```

After this run `sudo kubeadm reset` command and then the `init` or `join` command.

\$ sudo touch "/etc/docker/daemon.json"

\$ sudo nano "/etc/docker/daemon.json"

ADD code

CTRL+ O enter CTRL X

\$ sudo cat "/etc/docker/daemon.json"

{

"exec-opts": ["native.cgroupdriver=systemd"]

}

\$ sudo systemctl daemon-reload

\$ sudo systemctl restart docker

\$ sudo systemctl restart kubelet

\$ sudo kubeadm reset



PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



STEP 9 on master node

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

```
[ anders]
[mark-control-plane] Marking the node master as control-plane by adding the taints [node-role.kubernetes.io/control-plane:NoSchedule]
[bootstrap-token] Using token: sloata.5sllt69zvc8yj5tc
[bootstrap-token] Configuring bootstrap tokens, cluster-info ConfigMap, RBAC Roles
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to get nodes
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to post CSRs in order for nodes to get long term certificates
[bootstrap-token] Configured RBAC rules to allow the csrapprover controller automatically approve CSRs from a Node Bootstrap token
[bootstrap-token] Configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
[bootstrap-token] Creating the "cluster-info" ConfigMap in the "kube-public" namespace
[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable kubelet client certificate and key
addons] Applied essential addon: CoreDNS
addons] Applied essential addon: kube-proxy

Your Kubernetes control-plane has initialized successfully!

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

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

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

export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
https://kubernetes.io/docs/concepts/cluster-administration/addons/

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

kubeadm join 172.31.90.169:6443 --token sloata.5sllt69zvc8yj5tc \
    --discovery-token-ca-cert-hash sha256:80c23edc2552e4d0e671cb974fd0dc912a025d91f1c996f72c698709537d05e1
ubuntu@master:~$ []
```

Next, enter the following to create a directory for the cluster: (Master)

kubernetes-master \$ mkdir -p \$HOME/.kube

**kubernetes-master \$ sudo cp -i /etc/kubernetes/admin.conf
\$HOME/.kube/config**

kubernetes-master \$ sudo chown \$(id -u):\$(id -g) \$HOME/.kube/config



copy

```
kubeadm join 172.31.90.169:6443 --token s1oata.5sllt69zvc8yj5tc \
--discovery-token-ca-cert-hash
sha256:80c23edc2552e4d0e671cb974fd0dc912a025d91f1c996f72c698709537
d05e1
```

Make it as and copy in worker after flannel is created on master(after step 10)

```
kubeadm join 172.31.90.169:6443 --token s1oata.5sllt69zvc8yj5tc --
discovery-token-ca-cert-hash
sha256:80c23edc2552e4d0e671cb974fd0dc912a025d91f1c996f72c698709537
d05e1
```

it will give error

```
sudo kubeadm join 172.31.90.169:6443 --token s1oata.5sllt69zvc8yj5tc --
discovery-token-ca-cert-hash
sha256:80c23edc2552e4d0e671cb974fd0dc912a025d91f1c996f72c698709537
d05e1 --ignore-preflight-errors=all
```

STEP 10 Copy weblink from master node

<https://kubernetes.io/docs/concepts/cluster-administration/addons/>

goto flannel

copy this command and paste on master

For Kubernetes v1.17+

Deploying Flannel with kubectl

```
kubectl apply -f https://github.com/flannel-io/flannel/releases/latest/download/kube-flannel.yml
```

If you use custom `podCIDR` (not `10.244.0.0/16`) you first need to download the above manifest and modify the network to match your one.



Flannel created

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

```
kubeadm join 172.31.90.169:6443 --token sloata.5sllt69zvc8yj5tc \
    --discovery-token-ca-cert-hash sha256:80c23edc2552e4d0e671cb974fd0dc912a025d91f1c996f72c698709537d05e1
ubuntu@master:~$ mkdir -p $HOME/.kube
ubuntu@master:~$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
ubuntu@master:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@master:~$ ^C
ubuntu@master:~$ ^C
ubuntu@master:~$ kubectl apply -f https://github.com/flannel-io/flannel/releases/latest/download/kube-flannel.yml
namespace/kube-flannel created
serviceaccount/flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
ubuntu@master:~$ ^C
```

Step 11:

AFTER

```
r endpoint "unix:///var/run/containerd/containerd.sock": rpc error: code = Unavailable desc = connection error: desc = \"transport: Error while dialing: dial
/run/containerd/containerd.sock: connect: permission denied\""
, error: exit status 1
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
error: execution phase kubelet-start: couldn't save bootstrap-kubelet.conf to disk: open /etc/kubernetes/bootstrap-kubelet.conf: permission denied
To see the stack trace of this error execute with --v=5 or higher
ubuntu@worker1:~$ sudo ^[[200-kubeadm join 172.31.90.169:6443 --token sloata.5sllt69zvc8yj5tc --discovery-token-ca-cert-hash sha256:80c23edc2552e4d0e671cb974fd0dc912
96f72c698709537d05e1 --ignore-preflight-errors=all
sudo: kubeadm: command not found
ubuntu@worker1:~$ sudo kubeadm join 172.31.90.169:6443 --token sloata.5sllt69zvc8yj5tc --discovery-token-ca-cert-hash sha256:80c23edc2552e4d0e671cb974fd0dc912
96f72c698709537d05e1 --ignore-preflight-errors=all
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 1.004306297s
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap

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

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.
ubuntu@worker1:~$
```



Step 12:

Reboot both instances

```
Last login: Mon Jul 22 08:44:23 2024 from 18.206.107.29
ubuntu@master:~$ kubectl get pods --all-namespaces
NAMESPACE      NAME                READY   STATUS    RESTARTS   AGE
kube-flannel   kube-flannel-ds-qtfbw   1/1     Running   2 (66s ago)  16m
kube-flannel   kube-flannel-ds-t9v2g   1/1     Running   2 (65s ago)  6m29s
kube-system    coredns-7db6d8ff4d-2h5b4  1/1     Running   1 (119s ago) 30m
kube-system    coredns-7db6d8ff4d-sfzs8  1/1     Running   1 (119s ago) 30m
kube-system    etcd-master            1/1     Running   1 (119s ago) 31m
kube-system    kube-apiserver-master  1/1     Running   1 (119s ago) 31m
kube-system    kube-controller-manager-master  1/1     Running   1 (119s ago) 31m
kube-system    kube-proxy-8kd97       0/1     CrashLoopBackOff  5 (3s ago)  6m29s
kube-system    kube-proxy-9x78m       0/1     CrashLoopBackOff  10 (21s ago) 30m
kube-system    kube-scheduler-master  1/1     Running   1 (119s ago) 31m
ubuntu@master:~$ kubectl get pods --all-namespaces
NAMESPACE      NAME                READY   STATUS    RESTARTS   AGE
kube-flannel   kube-flannel-ds-qtfbw   1/1     Running   2 (118s ago) 17m
kube-flannel   kube-flannel-ds-t9v2g   1/1     Running   2 (117s ago) 7m21s
kube-system    coredns-7db6d8ff4d-2h5b4  1/1     Running   1 (2m51s ago) 31m
kube-system    coredns-7db6d8ff4d-sfzs8  1/1     Running   1 (2m51s ago) 31m
kube-system    etcd-master            1/1     Running   1 (2m51s ago) 31m
kube-system    kube-apiserver-master  1/1     Running   1 (2m51s ago) 31m
kube-system    kube-controller-manager-master  1/1     Running   1 (2m51s ago) 31m
kube-system    kube-proxy-8kd97       1/1     Running   6 (55s ago)  7m21s
kube-system    kube-proxy-9x78m       1/1     Running   11 (73s ago) 31m
kube-system    kube-scheduler-master  1/1     Running   1 (2m51s ago) 31m
ubuntu@master:~$
```

```
ubuntu@master:~$ kubectl get nodes
NAME      STATUS  ROLES      AGE   VERSION
master   Ready   control-plane  44m   v1.30.3
worker1  Ready   <none>        19m   v1.30.3
ubuntu@master:~$
```



Deploy service

On browser search for nginx deployment yaml

```
ubuntu@master:~$ sudo nano deploy.yaml
```

```
ubuntu@master:~$ sudo cat deploy.yaml
```

```
apiVersion: apps/v1
```

```
kind: Deployment
```

```
metadata:
```

```
  name: nginx-deployment
```

```
spec:
```

```
  selector:
```

```
    matchLabels:
```

```
      app: nginx
```

```
  replicas: 2 # tells deployment to run 2 pods matching the template
```

```
  template:
```

```
    metadata:
```

```
      labels:
```

```
        app: nginx
```

```
    spec:
```

```
      containers:
```

```
        - name: nginx
```

```
          image: nginx:1.14.2
```

```
          ports:
```

```
            - containerPort: 80
```



PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology
(NBA Accredited)



ubuntu@master:~\$ kubectl create -f deploy.yaml

```
- name: nginx
  image: nginx:1.14.2
  ports:
    - containerPort: 80
ubuntu@master:~$ kubectl create -f deploy.yaml
deployment.apps/nginx-deployment created
ubuntu@master:~$ ^C
ubuntu@master:~$ [ ]
```

ubuntu@master:~\$ kubectl get deploy

```
ubuntu@master:~$ kubectl get deploy
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   2/2     2           2          2m32s
ubuntu@master:~$ ^C
ubuntu@master:~$ ^C
ubuntu@master:~$ [ ]
```

kubectl expose deployment.apps/nginx-deployment --type="LoadBalancer"

ubuntu@master:~\$ kubectl get svc

```
- containerPort: 80
ubuntu@master:~$ kubectl create -f deploy.yaml
deployment.apps/nginx-deployment created
ubuntu@master:~$ ^C
ubuntu@master:~$ kubectl get deploy
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   2/2     2           2          2m32s
ubuntu@master:~$ ^C
ubuntu@master:~$ ^C
ubuntu@master:~$ kubectl expose deployment.apps/nginx-deployment --type="LoadBalance"
Error from server (NotFound): deployments.apps "nginx-deployment" not found
ubuntu@master:~$ ^C
ubuntu@master:~$ kubectl expose deployment.apps/nginx-deployment --type="LoadBalancer"
Error from server (NotFound): deployments.apps "nginx-deployment" not found
ubuntu@master:~$ kubectl expose deployment.apps/nginx-deployment --type="LoadBalancer"
service/nginx-deployment exposed
ubuntu@master:~$ kubectl get svc
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
kubernetes   ClusterIP   10.96.0.1   <none>       443/TCP      57m
nginx-deployment   LoadBalancer   10.96.174.79  <pending>     80:31825/TCP  15s
ubuntu@master:~$ ^C
ubuntu@master:~$ ^C
ubuntu@master:~$ [ ]
```



Go to instance, master select public ipv4

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with navigation links like EC2 Dashboard, EC2 Global View, Events, Console-to-Code, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, and Network & Security. The main area displays a table of instances. A green banner at the top says "Successfully initiated rebooting of i-046bea7b3a7423e7a,i-095a903dc278f53de". The table has columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 DNS. Two instances are listed: "Master" (i-095a903dc278f53de) and "Worker-node" (i-046bea7b3a7423e7a). The "Worker-node" instance is selected, and its details are shown in a modal window. The modal has tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. Under the Details tab, the "Instance summary" section shows the Instance ID (i-046bea7b3a7423e7a (Worker-node)), Instance state (Running), and Hostname type (IP name: ip-172-31-88-142.ec2.internal). It also lists Public IPv4 address (52.87.162.37 | open address), Private IP4 DNS name (IPv4 only) (ip-172-31-88-142.ec2.internal), and Instance type (t2.micro). To the right of the modal, there are sections for Private IPv4 addresses (172.31.88.142), Public IPv4 DNS (ec2-52-87-162-37.compute-1.amazonaws.com | open address), and Select ID addresses.

Go to brower

Ipv4:portnumber

The screenshot shows a web browser window. The address bar says "Not secure 52.87.162.37:31825". The page itself has a simple layout with a large "Welcome to nginx!" heading in bold black font. Below it, a smaller text says "If you see this page, the nginx web server is successfully installed and working. Further configuration is required." At the bottom, there's a link "For online documentation and support please refer to [nginx.org](#). Commercial support is available at [nginx.com](#)." and a "Thank you for using nginx." message.



Academic Year: 2023-24

Semester: V

Class / Branch: TE IT

Subject: Advanced Devops Lab (ADL)

Subject Lab Incharge: Prof. Manjusha Kashikar/Prof.Sonal Jain/Prof.Yaminee Patil

EXPERIMENT NO. 05

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

Theory:

Terraform

Terraform is an infrastructure as code (IaC) tool that allows you to build, change, and version infrastructure safely and efficiently. This includes low-level components such as compute instances, storage, and networking, as well as high-level components such as DNS entries, SaaS features, etc. Terraform can manage both existing service providers and custom in-house solutions.

Key Features

Infrastructure as Code:

You describe your infrastructure using Terraform's high-level configuration language in human-readable, declarative configuration files. This allows you to create a blueprint that you can version, share, and reuse.

Resource Graph

Terraform builds a resource graph and creates or modifies non-dependent resources in parallel. This allows Terraform to build resources as efficiently as possible and gives you greater insight into your infrastructure.

Change Automation

Terraform can apply complex changesets to your infrastructure with minimal human interaction. When you update configuration files, Terraform determines what changed and creates incremental execution plans that respect dependencies.

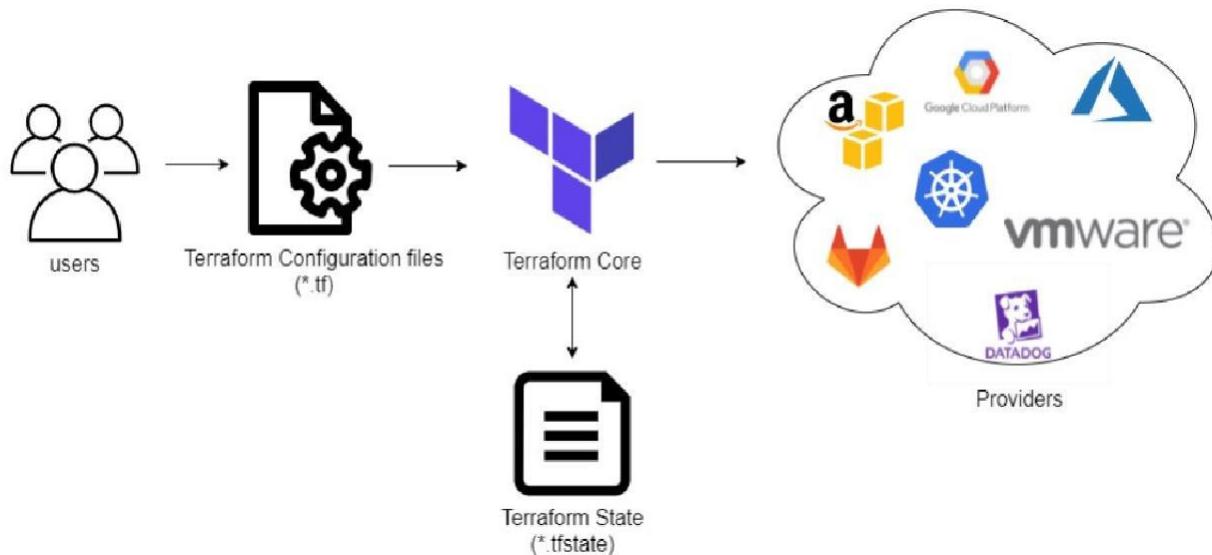


How does Terraform work?

Terraform actually works, there's sort of two major components:

one is the **terraform core**: it takes the terraform configuration which is being provided by the user and then takes the terraform state which is managed by terraform itself. As such, this gets fed into the core that is responsible for figuring out what is that graph of our different resources for example how these different pieces relate to each other or what needs to be created/updated/destroyed, it does all the essential lifecycle management.

On the backside, terraform supports many different **providers**, such as: cloud providers (AWS,GCP,AZURE) and they also could be on-premise infrastructure (VMware,OpenStack.) But this support is not restricted or limited only to Infrastructure As A Service , terraform can also manage higher level like Platform As A Service(Kubernetes,Lambdas..)or even Software As A Service (DataDog,GitHub..)



All of these are important pieces of the infrastructure, they are all part of the logical end-to-end delivery.

Terraform has over a hundred providers for different technologies, and each provider gives terraform users access to their resources. It also gives you the ability to create infrastructure at different levels.



Terraform Core Concepts:

Below are the core concepts/terminologies used in Terraform:

- **Variables:** Also used as input-variables, it is a key-value pair used by Terraform modules to allow customization.
- **Provider:** It is a plugin to interact with APIs of service and access its related resources.
- **Module:** It is a folder with Terraform templates where all the configurations are defined
- **State:** It consists of cached information about the infrastructure managed by Terraform and its related configurations.
- **Resources:** It refers to a block of one or more infrastructure objects (compute instances, virtual networks, etc.), which are used in configuring and managing the infrastructure.
- **Data Source:** It is implemented by providers to return information on external objects to terraform.
- **Output Values:** These are return values of a terraform module that can be used by other configurations.
- **Plan:** It is one of the stages where it determines what needs to be created, updated, or destroyed to move from the real/current state of the infrastructure to the desired state.
- **Apply:** It is one of the stages where it applies the changes in the real/current state of the infrastructure in order to move to the desired state.

Terraform Installation Steps on Ubuntu18.04

Step: 1 Terraform uses HashiCorp Configuration Language (HCL) to manage environments of Operators and Infrastructure teams. To download go to site
<https://www.terraform.io/downloads.html>

Select the appropriate package for your operating system and architecture.



terraform.io/downloads.html

Vishal Badgujar + Vishal Saheb... 📩 Gmail 🎵 YouTube 🚙 BDA - Google... 🌐 ...:: Mumbai Un... 📈 WhatsApp 🛡️ Cis

Navigation Development

- Terraform Registry Publishing
- Glossary

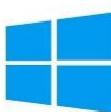
 **macOS**
64-bit | Arm64

 **FreeBSD**
32-bit | 64-bit | Arm

 **Linux**
32-bit | 64-bit | Arm | Arm64

 **OpenBSD**
32-bit | 64-bit

 **Solaris**
64-bit

 **Windows**
32-bit | 64-bit

Step:2 unzip the archive by using below command

```
vishal@master:~$ unzip terraform_1.0.3_linux_amd64.zip
```

The archive will extract a single binary called **terraform**.



Step 3: Change the directory to unzipped folder

```
vishal@master:~$ cd terraform_1.0.3_linux_amd64/
```

and Move the terraform binary to a directory included in your system's PATH in my case *usr/local/bin/*

```
vishal@master:~$ sudo mv terraform /usr/local/bin/
```

Step 4: To check whether Terraform is installed, run:

```
vishal@master:~$ terraform -v
Terraform v1.0.3
on linux_amd64
vishal@master:~$
```

Conclusion: Write your own findings.



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



Academic Year: 2022-23

Semester: V

Class / Branch: TE IT

Subject: Advanced Devops Lab (ADL)

Subject Lab Incharge: Prof. Manjusha K.

EXPERIMENT NO. 06

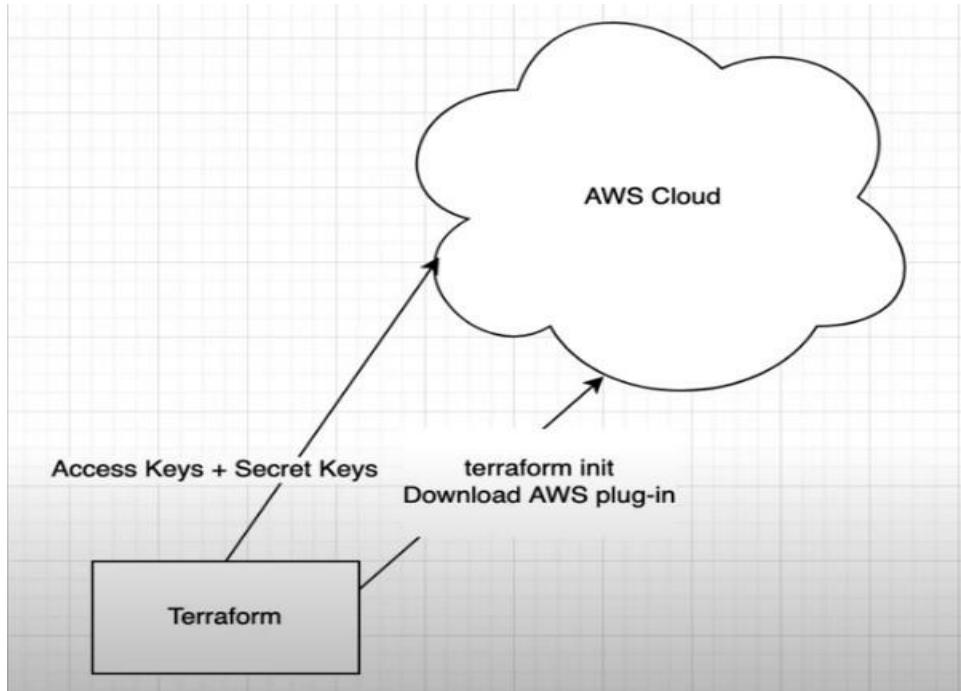
Aim: To Build, change, and destroy AWS infrastructure Using Terraform.

Theory:

Hashicorp's Terraform is an open-source tool for provisioning and managing cloud infrastructure. Terraform can provision resources on any cloud platform.

Terraform allows you to create infrastructure in configuration files(**tf files**) that describe the topology of cloud resources. These resources include virtual machines, storage accounts, and networking interfaces.

We will see how you can use Terraform to provision EC2 instance. Please do the below steps for provisioning EC2 instances on AWS:





PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



Pre-requisites:

1. Install the AWS CLI version 2 on Linux

Follow these steps from the command line to install the AWS CLI on Linux.

Install curl on linux

```
vishal@apsit:~$ sudo apt-get install curl
```

```
vishal@apsit:~$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
```

```
vishal@apsit:~$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
% Total    % Received % Xferd  Average Speed   Time     Time  Current
          Dload  Upload   Total Spent  Left  Speed
100 41.8M  100 41.8M    0      0  2529k      0  0:00:16  0:00:16  --:--:-- 2555k
```

```
vishal@apsit:~$ sudo apt install unzip
```

```
vishal@apsit:~$ sudo apt install unzip
```

```
vishal@apsit:~$ sudo unzip awscliv2.zip
```

```
vishal@apsit:~$ sudo unzip awscliv2.zip
```

```
vishal@apsit:~$ sudo ./aws/install
```

```
vishal@apsit:~$ sudo ./aws/install
You can now run: /usr/local/bin/aws --version
```

```
vishal@apsit:~$ aws --version
```

it should display the below output.

```
aws-cli/2.1.29 Python/3.8.8 Linux/5.4.0-1038-aws exe/x86_64.ubuntu.18 prompt/off
```



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

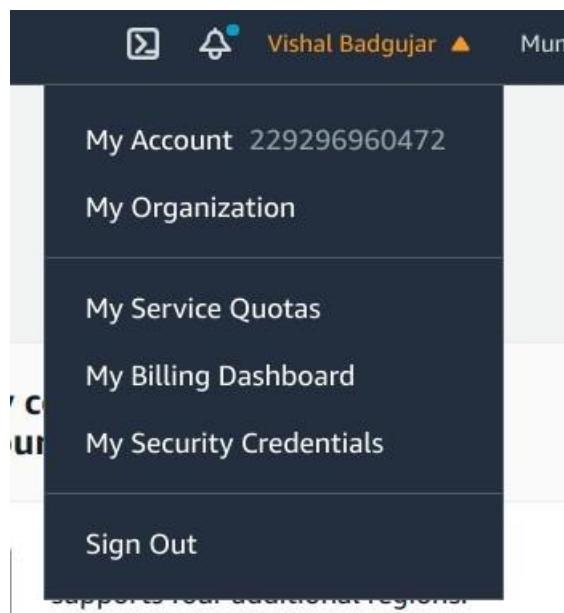
(NBA Accredited)



```
vishal@apsit:~$ aws --version
aws-cli/2.2.25 Python/3.8.8 Linux/5.4.0-80-generic exe/x86_64.ubuntu.18 prompt/off
```

2. Create a new access key if you don't have one. Make sure you download the keys in your local machine.

Login to AWS console, click on username and go to My security credentials.



Continue on security credentials, click on access keys

Your Security Credentials

Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity

To learn more about the types of AWS credentials and how they're used, see [AWS Security Credentials](#)

- ▲ Password
- ▲ Multi-factor authentication (MFA)
- ▼ Access keys (access key ID and secret access key)

Use access keys to make programmatic calls to AWS from the AWS CLI, Tools for PowerShell, AWS Lambda, and other services at any time.

For your protection, you should never share your secret keys with anyone. As a best practice, we recommend that you rotate your access keys regularly. If you lose or forget your secret key, you cannot retrieve it. Instead, create a new access key and delete the old one.

Created	Access Key ID	Last Used
---------	---------------	-----------



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



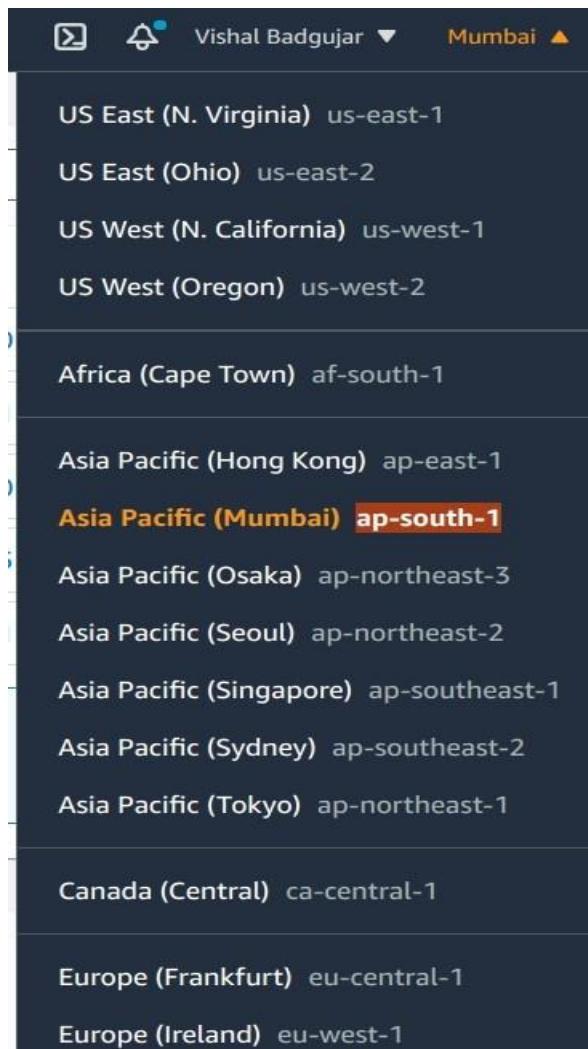
Perform below commands in Linux where you have installed Terraform

First setup your access keys, secret keys and region code locally.

```
vishal@apsit:~$aws configure
```

Created	Access Key ID	Last Used	Last Used Region	Last Used Service	Status
Jun 4th 2021	AKIATKYZJ6PMCN2VF436	2021-07-04 21:26 UTC+0530	us-east-1	sts	Active
Aug 1st 2021	AKIATKYZJ6PMFLTCGGPV	N/A	N/A	N/A	Active

You can check region as shown in below image :





PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



```
vishal@apsit:~$ aws configure
AWS Access Key ID [None]: AKIATKYZJ6PMFLTCGGPV
AWS Secret Access Key [None]: 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
Default region name [None]: ap-south-1
Default output format [None]:
vishal@apsit:~$ █
```

Create one Directory for Terraform project in which all files of terraform we can save

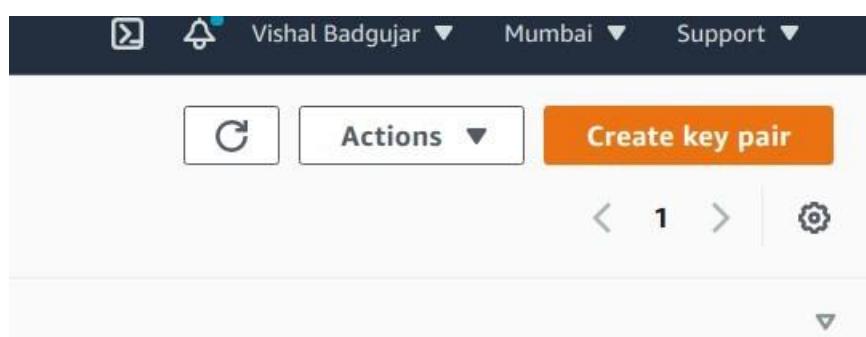
```
vishal@apsit:~$ cd ~
vishal@apsit:~$ mkdir project-terraform
vishal@apsit:~$ cd project-terraform
```

```
vishal@apsit:~$ mkdir project-terraform
vishal@apsit:~$ cd project-terraform/
vishal@apsit:~/project-terraform$ █
```

Create Terraform Files

[vishal@apsit:~\\$ sudo nano variables.tf](#)

In order to provide key name in variables first create key pair as shown:





PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



Give name to key pair file as **terraform**

Create key pair

Key pair

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

Name

terraform

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

Private key file format

- .pem
For use with OpenSSH
- .ppk
For use with PuTTY

Tags (Optional)

No tags associated with the resource.

Add tag

You can add 50 more tags.

Cancel

Create key pair

Key pair is generated

<input type="checkbox"/>	terraform	d4:aa:d4:24:a8:f5:a2:2a:28:59:e6:38:d...	key-080872ef28d76fe24



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



Use your Region and Key name in variable.tf as shown and provide instance type which you want to create.

```
File Edit View Search Terminal Help
GNU nano 2.9.3                               variables.tf                                Modified

variable "aws_region" {
  description = "The AWS region to create things in."
  default     = "ap-south-1"
}

variable "key_name" {
  description = " SSH keys to connect to ec2 instance"
  default     = "terraform"
}

variable "instance_type" {
  description = "instance type for ec2"
  default     = "t2.micro"
}

^G Get Help   ^O Write Out   ^W Where Is   ^K Cut Text   ^J Justify   ^C Cur Pos   M-U Undo
^X Exit      ^R Read File    ^H Replace    ^U Uncut Text  ^T To Spell   ^L Go To Line M-E Redo
```

After creating variable terraform file note down the AMI ID of instance which u want to create which we will use to configure our instance in main.tf file.



Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-04db49c0fb2215364 (64-bit x86) / ami

Amazon Linux

Free tier eligible

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on AWS Lambda, the latest version of the Amazon Linux 2 kernel, and the latest version of the AWS Lambda runtime environment. This AMI includes the latest version of the Amazon Linux 2 kernel, the latest version of the AWS Lambda runtime environment, and the latest version of the AWS Lambda configuration files. This AMI is the successor of the Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-04db49c0fb2215364 (64-bit x86) / ami



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



Now create main.tf file:

```
vishal@apsit:~/project-terraform$ sudo nano main.tf
```

```
provider "aws" {
```

```
    region = var.aws_region
```

```
}
```

```
#Create security group with firewall rules
```

```
resource "aws_security_group" "security_jenkins_port" {
```

```
    name      = "security_jenkins_port"
```

```
    description = "security group for jenkins"
```

```
    ingress {
```

```
        from_port = 8080
```

```
        to_port   = 8080
```

```
        protocol  = "tcp"
```

```
        cidr_blocks = ["0.0.0.0/0"]
```

```
}
```

```
    ingress {
```

```
        from_port = 22
```

```
        to_port   = 22
```

```
        protocol  = "tcp"
```

```
        cidr_blocks = ["0.0.0.0/0"]
```

```
}
```

```
# outbound from jenks server
```



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



```
egress {  
    from_port = 0  
    to_port   = 65535  
    protocol  = "tcp"  
    cidr_blocks = ["0.0.0.0/0"]  
}
```

```
tags= {  
    Name = "security_jenkins_port"  
}  
}
```

```
resource "aws_instance" "myFirstInstance" {  
    ami      = "ami-0b9064170e32bde34"  
    key_name = var.key_name  
    instance_type = var.instance_type  
    security_groups= [ "security_jenkins_port"]  
    tags= {  
        Name = "jenkins_instance"  
    }  
}
```

```
# Create Elastic IP address  
resource "aws_eip" "myFirstInstance" {  
    vpc      = true  
    instance = aws_instance.myFirstInstance.id
```



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



```
tags= {  
    Name = "jenkins_elstic_ip"  
}  
}  
  
}
```

Put AMI-ID in above highlighted space and Now execute the below command:

```
vishal@apsit:~/project-terraform$ terraform init
```

you should see like below screenshot.

```
vishal@apsit:~/project-terraform$ terraform init  
  
Initializing the backend...  
  
Initializing provider plugins...  
- Finding latest version of hashicorp/aws...  
- Installing hashicorp/aws v3.52.0...  
- Installed hashicorp/aws v3.52.0 (signed by HashiCorp)  
  
Terraform has created a lock file .terraform.lock.hcl to record the provider  
selections it made above. Include this file in your version control repository  
so that Terraform can guarantee to make the same selections by default when  
you run "terraform init" in the future.  
  
Terraform has been successfully initialized!  
  
You may now begin working with Terraform. Try running "terraform plan" to see  
any changes that are required for your infrastructure. All Terraform commands  
should now work.  
  
If you ever set or change modules or backend configuration for Terraform,  
rerun this command to reinitialize your working directory. If you forget, other  
commands will detect it and remind you to do so if necessary.
```



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



Execute the below command

```
vishal@apsit:~/project-terraform$ terraform plan
```

the above command will show how many resources will be added.

Plan: 3 to add, 0 to change, 0 to destroy.

```
vishal@apsit:~/project-terraform$ terraform plan

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

Terraform will perform the following actions:

# aws_eip.myFirstInstance will be created
+ resource "aws_eip" "myFirstInstance" {
    + allocation_id      = (known after apply)
    + association_id    = (known after apply)
    + carrier_ip         = (known after apply)
```

Plan: 3 to add, 0 to change, 0 to destroy.

Execute the below command

```
vishal@apsit:~/project-terraform$ terraform apply
```

Provide the value as Yes for applying terraform

Plan: 3 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



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



Plan: 3 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

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

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

```
aws_instance.myFirstInstance: Creating...
aws_instance.myFirstInstance: Still creating... [10s elapsed]
aws_instance.myFirstInstance: Still creating... [20s elapsed]
aws_instance.myFirstInstance: Still creating... [30s elapsed]
aws_instance.myFirstInstance: Creation complete after 32s [id=i-0a4a0fb7e55252d0f]
aws_eip.myFirstInstance: Creating...
aws_eip.myFirstInstance: Creation complete after 1s [id=eipalloc-0fd8f60524b10fc93]
```

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

Now login to EC2 console, to see the new instances up and running, you can see Jenkins_instance is up and running which we deploy from terraform.

Instances (2) Info							
<input type="checkbox"/> Filter instances		C	Connect	Instance state ▾	Actions ▾	Launch instances	▼
Instance state: running X		Clear filters					
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	BitnamiMoodleCfDb01Ec2Instance	i-07ca078b9bcb1598b	Running	t3a.medium	2/2 checks passed	No alarms	ap-south-1a
<input type="checkbox"/>	jenkins_instance	i-0a4a0fb7e55252d0f	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



You can also check the security group resource details which you created from terraform :

The screenshot shows the AWS EC2 Security Groups console. The top navigation bar includes 'EC2 > Security Groups > sg-0f04dc9c71cdcf3dd - security_jenkins_port'. Below the navigation is a 'Details' section with four columns: 'Security group name' (sg-0f04dc9c71cdcf3dd), 'Security group ID' (sg-0f04dc9c71cdcf3dd), 'Description' (security group for jenkins), and 'VPC ID' (vpc-18be7c73). Underneath this is another row with 'Owner' (229296960472) and 'Inbound rules count' (2 Permission entries). The 'Outbound rules count' and 'Permission entry' are listed below it. At the bottom of the 'Details' section are tabs for 'Inbound rules', 'Outbound rules', and 'Tags'. A note says 'You can now check network connectivity with Reachability Analyzer' with a 'Run Reachability Analyzer' button. Below this is a table titled 'Inbound rules (2)' with columns: Name, Security group rule..., IP version, Type, Protocol, Port range, Source, and D. The table lists two rules: one for SSH (TCP port 22) and one for Custom TCP (TCP port 8080).

Terraform destroy

you can also destroy or delete your instance by using terraform destroy command :

```
vishal@apsit:~/project-terraform$ terraform destroy
```

```
Plan: 0 to add, 0 to change, 3 to destroy.
```

```
Do you really want to destroy all resources?
```

Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

```
Enter a value: yes
```



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



```
Enter a value: yes
```

```
aws_eip.myFirstInstance: Destroying... [id=eipalloc-0fd8f60524b10fc93]
aws_security_group.security_jenkins_port: Destroying... [id=sg-0f04dc9c71cdcf3dd]
aws_eip.myFirstInstance: Destruction complete after 2s
aws_instance.myFirstInstance: Destroying... [id=i-0a4a0fb7e55252d0f]
aws_security_group.security_jenkins_port: Still destroying... [id=sg-0f04dc9c71cdcf3dd, 10s elapsed]
aws_instance.myFirstInstance: Still destroying... [id=i-0a4a0fb7e55252d0f, 10s elapsed]
aws_security_group.security_jenkins_port: Still destroying... [id=sg-0f04dc9c71cdcf3dd, 20s elapsed]
aws_instance.myFirstInstance: Still destroying... [id=i-0a4a0fb7e55252d0f, 20s elapsed]
aws_security_group.security_jenkins_port: Still destroying... [id=sg-0f04dc9c71cdcf3dd, 30s elapsed]
aws_instance.myFirstInstance: Still destroying... [id=i-0a4a0fb7e55252d0f, 30s elapsed]
aws_security_group.security_jenkins_port: Destruction complete after 38s
aws_instance.myFirstInstance: Still destroying... [id=i-0a4a0fb7e55252d0f, 40s elapsed]
aws_instance.myFirstInstance: Destruction complete after 40s
```

```
Destroy complete! Resources: 3 destroyed.
```

Now you can see instance which you created by using terraform is deleted successfully from aws console also you can check it will removed successfully:

Instances (1) Info							
		C	Connect	Instance state ▾	Actions ▾	Launch instances	▼
Instance state: running X		Clear filters					
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	BitnamiMoodleCfDb01Ec2Instance	i-07ca078b9bcb1598b	Running	t3a.medium	2/2 checks passed	No alarms	ap-south-1a

All the Resources including Security groups, EC2 instances using terraform will be deleted. In this way we can automate infrastructure set up using terraform in aws cloud.

Conclusion: Write your own findings.



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



Academic Year: 2022-23

Semester: V

Class / Branch: TE IT

Subject: Advanced Devops Lab (ADL)

Subject Lab Incharge: Prof. Manjusha Kashikar

EXPERIMENT NO. 07

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

Theory:

Static application security testing (SAST) is a way to perform automated testing and analysis of a program's source code without executing it to catch security vulnerabilities early on in the software development cycle. Also referred to as static code analysis, SAST is the process of parsing through the code looking at how it was written and checking for security vulnerabilities and safety concerns.

Because static application security testing tools don't need a running application to perform an analysis, they can be used early and often in the implementation phase of the software development life cycle (SDLC). As a developer is writing code, SAST can analyze it in real-time to inform the user of any rule violations, so you can immediately deal with issues and deliver higher quality applications out of the box while preventing issues at the end of the development process.

Additionally, as SAST helps you audit code and triage issues during implementation, test automation tools can also easily integrate into development ecosystems where continuous integration/continuous delivery (CI/CD) are part of the workflow that helps assure secure, safe, and reliable code during integration, and before it's delivered.

What's the Difference Between SAST and DAST?

While SAST analyses every line of code without running the application, dynamic application security testing (DAST) simulates malicious attacks and other external behaviors by searching for ways to exploit security vulnerabilities during runtime or black box testing.

DAST is particularly useful when catching unexpected vulnerabilities that development teams simply didn't think of. This additional level of insight that DAST brings offers a broad array of security testing to find flaws and prevent attacks like SQL injections, cross-site scripting (XSS), and



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



other exploits. Remember the 2014 Sony Pictures hack? That could have been prevented with DAST.

Comparing SAST against DAST, each is more effective than the other during different stages of the SDLC. SAST represents the developer's point of view to make sure that all coding procedures follow the appropriate safety standards to ensure the security of an application from the start. DAST, on the other hand, mimics the hacker approach to identify possible user behavior towards the end of development.

Steps:

- 1) Install and configure a Jenkins and SonarQube CICD environment using Docker containers.**
- 2) Configure Jenkins with the SonarQube Scanner plugin for automated static code analysis.**

1) Install and configure a Jenkins and SonarQube CICD environment using Docker containers.

Installation of Jenkins

The version of Jenkins included with the default Ubuntu packages is often behind the latest available version from the project itself. To take advantage of the latest fixes and features, you can use the project-maintained packages to install Jenkins.

```
manjusha@apsit:~$ wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key
| sudo apt-key add -
```

When the key is added, the system will return OK. Next, append the Debian package repository address to the server's sources.list:

```
manjusha@apsit:~$ sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ >
```



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



When both of these are in place, run `update` so that apt will use the new repository:

```
manjusha@apsit:~$ sudo apt update
```

Finally, install Jenkins and its dependencies:

```
manjusha@apsit:~$ sudo apt install jenkins
```

Let's start Jenkins using systemctl:

```
manjusha@apsit:~$ sudo systemctl start jenkins
```

Since systemctl doesn't display output, you can use its status command to verify that Jenkins started successfully:

```
manjusha@apsit:~$ sudo systemctl status jenkins
```

If everything went well, the beginning of the output should show that the service is active and configured to start at boot:

Now that Jenkins is running, let's adjust our firewall rules so that we can reach it from a web browser to complete the initial setup.

Opening the Firewall

By default, Jenkins runs on port 8080, so let's open that port using ufw:

```
manjusha@apsit:~$ sudo ufw allow
```

8080 Setting Up Jenkins

To set up your installation, visit Jenkins on its default port, 8080, using your server domain name or IP address: **http://your_server_ip_or_domain:8080**

You should see the Unlock Jenkins screen, which displays the location of the initial password:



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and [this file](#) on the server:

`/var/lib/jenkins/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

Continue

In the terminal window, use the cat command to display the password:

manjusha@apsit:~\$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword

Copy the 32-character alphanumeric password from the terminal and paste it into the Administrator password field, then click Continue.

The next screen presents the option of installing suggested plugins or selecting specific plugins:

Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY**Department of Information Technology**

(NBA Accredited)



We'll click the Install suggested plugins option, which will immediately begin the installation process:

Getting Started

Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding	** Pipeline: Milestone Step ** JavaScript GUI Lib: jQuery bundles (jQuery and jQuery UI) ** Jackson 2 API ** JavaScript GUI Lib: ACE Editor bundle ** Pipeline: SCM Step ** Pipeline: Groovy ** Pipeline: Input Step ** Pipeline: Stage Step ** Pipeline: Job ** Pipeline Graph Analysis ** Pipeline: REST API ** JavaScript GUI Lib: Handlebars bundle ** JavaScript GUI Lib: Moment.js bundle Pipeline: Stage View ** Pipeline: Build Step ** Pipeline: Model API ** Pipeline: Declarative Extension Points API ** Apache HttpComponents Client 4.x API ** JSch dependency
✓ Timestamper	✓ Workspace Cleanup	✓ Ant	✓ Gradle	
✗ Pipeline	✗ GitHub Branch Source	✗ Pipeline: GitHub Groovy Libraries	✓ Pipeline: Stage View	
✗ Git	✗ Subversion	✗ SSH Slaves	✗ Matrix Authorization Strategy	
✗ PAM Authentication	✗ LDAP	✗ Email Extension	✗ Mailer	



Getting Started

Create First Admin User

Username:	<input type="text" value="vishal"/>
Password:	<input type="password" value="....."/>
Confirm password:	<input type="password" value="....."/>
Full name:	<input type="text" value="Vishal Badgujar"/>
E-mail address:	<input type="text" value="vsbadgujar@apsit.edu.in"/>

Jenkins 2.289.2

[Skip and continue as admin](#)

[Save and Continue](#)

When the installation is complete, you will be prompted to set up the first administrative user. It's possible to skip this step and continue as admin using the initial password we used above, but we'll take a moment to create the user.

Instance Configuration

Jenkins URL:

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD_URL environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

After confirming the appropriate information, click Save and Finish. You will see a confirmation page confirming that "Jenkins is Ready!"



PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY
Department of Information Technology
(NBA Accredited)



Getting Started

Jenkins is ready!

Your Jenkins setup is complete.

[Start using Jenkins](#)

Click Start using Jenkins to visit the main Jenkins dashboard:

The screenshot shows the Jenkins management interface at the URL <http://127.0.0.1:8080/manage>. The left sidebar has a 'Manage Jenkins' section selected. The main content area is titled 'Manage Jenkins' and includes sections for 'System Configuration' (Configure System, Global Tool Configuration, Manage Plugins), 'Security' (Configure Global Security, Manage Credentials, Configure Credential Providers), and 'Status Information'. There are also collapsed sections for 'Build Queue' (No builds in the queue) and 'Build Executor Status' (2 idle). At the bottom, there are tabs for PDF files: RM_S.A.Rasal.pdf, RM_Lecture_....pdf, RM_Lecture_....pdf, and RM_Formula...pdf.



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



SonarQube Setup

Before proceeding with the integration, we will setup SonarQube Instance. we are using SonarQube Docker Container.

[manjusha@apsit:~\\$ docker run -d -p 9000:9000 sonarqube](#)

```
vishal@apsit:~$ sudo docker run -d -p 9000:9000 sonarqube
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
5843afab3874: Pull complete
a131164fad71: Pull complete
d77763c1bc70: Pull complete
572e2a545fb3: Pull complete
f32e9b0d93df: Pull complete
Digest: sha256:d1f18c804d8bdcea0a90d13d93f6ec9af9012d48747fcb63dff
b7c8f06b5666f
Status: Downloaded newer image for sonarqube:latest
cf5325b4e2e80064d0d8faf76c8600ddd13cc26a5892074cac09674185f72fdc
vishal@apsit:~$ █
```

In the above command, we are forwarding port 9000 of the container to the port 9000 of the host machine as SonarQube is will run on port 9000. Then, from the browser, enter <http://localhost:9000>. After That, you will see the SonarQube is running. Then, login using default credentials (admin:admin).

Log In to SonarQube



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



Generate User Token

Now, we need to get the SonarQube user token to make connection between Jenkins and SonarQube. For the same, go to **Administration > User > My Account > Security** and then, from the bottom of the page you can create new tokens by clicking the Generate Button. Copy the Token and keep it safe.

C96798e9bd081e117189b516c868ddb7d87ee785 SonarQube

Tokens of Administrator

Generate Tokens

Enter Token Name Generate

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

Copy `c96798e9bd081e117189b516c868ddb7d87ee785`

Name	Last use	Created
Jenkin	Never	July 28, 2021

[Revoke](#)

[Done](#)



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



2) Configure Jenkins with the SonarQube Scanner plugin for automated static code analysis.

Jenkins Setup for SonarQube

Before all, we need to install the SonarQube Scanner plugin in Jenkins. For the same, go to **Manage Jenkins > Plugin Manager > Available**. From here, type SonarQube Scanner then select and install.

The screenshot shows the Jenkins Plugin Manager interface. The search bar at the top contains 'sonarqube scanner'. Below it, there are tabs for 'Updates', 'Available' (which is selected), 'Installed', and 'Advanced'. A table lists the 'SonarQube Scanner' plugin. The table columns are 'Name', 'Version', and 'Re'. The plugin details show 'SonarQube Scanner' version 2.13.1, with 'External Site/Tool Integrations' and 'Build Reports' listed under it. Below the table are buttons for 'Install without restart' (disabled), 'Download now and install after restart', and 'Check now'. A note says 'Update information obtained: 29 min ago'.

Tool Configuration SonarQube Scanner

Now, we need to configure the Jenkins plugin for SonarQube Scanner to make a connection with the SonarQube Instance. For that, got to **Manage Jenkins > Configure System > SonarQube Server**. Then, Add SonarQube. In this, give the Installation Name, Server URL then Add the Authentication token in the Jenkins Credential Manager and select the same in the configuration.



Jenkins

Dashboard > Credentials > System > Global credentials (unrestricted) > sonarqube

Back to Global credentials (unrestricted)

Update Delete Move

Scope: Global (Jenkins, nodes, items, all child items, etc.)

Secret: Concealed

ID: sonarqube

Description: sonarqube

Save

SonarQube servers

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

SonarQube installations

Name

SonarQube

Server URL

http://localhost:9000

Default is http://localhost:9000

Server authentication token

sonarqube

Add

SonarQube authentication token. Mandatory when anonymous access is disabled.

Then, we need to set-up the SonarQube Scanner to scan the source code in the various stage. For the same, go to **Manage Jenkins > Global Tool Configuration > SonarQube Scanner**. Then, Click **Add SonarQube Scanner Button**. From there, give some name of the scanner type and **Add Installer** of your choice. In this case, I have selected SonarQube Scanner from Maven Central.



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



SonarQube Scanner

SonarQube Scanner installations

[Add SonarQube Scanner](#)

 SonarQube Scanner

Name

SonarQube

Install automatically

 [Install from Maven Central](#)

Version

SonarQube Scanner 4.6.2.2472 ▾

[Add Installer ▾](#)



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



SonarQube Scanner in Jenkins Pipeline

Now, It's time to integrate the SonarQube Scanner in the Jenkins Pipeline. For the same, we are going to add one more stage in the Jenkinsfile called SonarQube and inside that, I am adding the following settings and code.

Enter an item name

SonarQube.
» Required field

Freestyle project
 This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

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.

Bitbucket Team/Project
 Scans a Bitbucket Cloud Team (or Bitbucket Server Project) for all repositories matching some defined markers.

Folder
 Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



General Build Triggers Advanced Project Options Pipeline

Description

Hello Pipeline job

[Plain text] [Preview](#)

Discard old builds [?](#)

Do not allow concurrent builds [?](#)

Do not allow the pipeline to resume if the controller restarts [?](#)

GitHub project [?](#)

Project url

<https://github.com/vishal003/jenkins-sonarqube/>

Github Configuration in Jenkins Pipeline

Pipeline

Definition

Pipeline script

Script

```
1 node
2 {
3   stage('clonning from GIT'){
4     git branch: 'main', credentialsId: 'GIT_REPO', url: 'https://github.com/vishal003/jenkins-sonarqube.git'
5   }
6 }
7
```

Git Clonning into Jenkins



PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY
Department of Information Technology
(NBA Accredited)



github.com/vishal003/jenkins-sonarqube

Apps Vishal Badgujar Vishal Saheb... Gmail YouTube BDA - Google... ...:: Mumbai Un... WhatsApp Cisco

Search or jump to... Pull requests Issues Marketplace Explore

vishal003 / **jenkins-sonarqube**
forked from devopshint/jenkins-sonarqube

Code Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags Go to file Add file Code

This branch is 6 commits ahead of devopshint:main. Contribute Fetch upstream

vishal003 Update README.md	80c34f4 41 minutes ago	16 commits
project	Update sonar-analysis	42 minutes ago
README.md	Update README.md	41 minutes ago

README.md

jenkins-Github-sonarqube CICD Pipeline

Github Repository Contents



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



Dashboard > sonarqube > #2

[Back to Project](#)

[Status](#)

[Changes](#)

[Console Output](#)

[View as plain text](#)

[Edit Build Information](#)

[Delete build #2'](#)

[Git Build Data](#)

[Open Blue Ocean](#)

[Replay](#)

[Pipeline Steps](#)

[Workspaces](#)

[Previous Build](#)

[Next Build](#)

Console Output

```
Started by user unknown or anonymous
Running in Durability level: MAX_SURVIVABILITY
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/sonarqube
[Pipeline]
[Pipeline] Stage
[Pipeline] { (clonning from GIT)
[Pipeline] git
The recommended git tool is: NONE
Warning: CredentialId 'GIT REPO' could not be found.
Cloning the remote Git repository
Cloning repository https://github.com/vishal003/jenkins-sonarqube.git
> git init /var/lib/jenkins/workspace/sonarqube # timeout=10
Fetching upstream changes from https://github.com/vishal003/jenkins-sonarqube.git
> git --version # timeout=10
> git --version # 'git version 2.17.1'
> git fetch --tags --progress -- https://github.com/vishal003/jenkins-sonarqube.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://github.com/vishal003/jenkins-sonarqube.git # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision ea3f635e2ceeb7b1e2d8b1fedf3394270961ea38 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f ea3f635e2ceeb7b1e2d8b1fedf3394270961ea38 # timeout=10
> git branch a v no abbrev # timeout=10
> git checkout -b main ea3f635e2ceeb7b1e2d8b1fedf3394270961ea38 # timeout=10
Commit message: "Update sonar-analysis"
First time build. Skipping changelog.
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Successfully Build Github Repository in Jenkins

Pre-requisite required for Integration settings of Jenkins SAST with SonarQube we have done here successfully, now in order to Integrate of Jenkins CICD with SonarQube with the help of sample JAVA program we will implement in next experiment.

Conclusion: Write your own findings.



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

(All Branches NBA Accredited)



Academic Year: 2022-23

Semester: V

Class / Branch: TE IT

Subject: Advanced Devops Lab (ADL)

Subject Lab Incharge: Prof. Manjusha K.

EXPERIMENT NO. 08

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 Java application.

Theory:

Integrating Jenkins with SonarQube provides you with an automated platform for performing continuous inspection of code for quality and security assurance.

Everyday enhancement simplifies developers' tasks. Let's assume a situation where Developers have committed their codes to the repository, and then they want to know the project source code quality, code smells, any bugs, vulnerabilities, code analysis, etc. So it is extremely challenging for them to know all this information. So what if they want all these source codes and information beforehand? For such cases, Jenkins is the best fit. If a Software developer starts to build any new project, then the source code is automatically or manually saved while using Jenkins and their daily commit operation is not needed every time.

For this purpose, we can go for CI/CD i.e. Continuous Integration &Continuous Deployment of the code using SonarQube-Jenkins Integration.

SonarQube :

SonarQube is an open-source platform, which is used for continuous analysis of source code quality by performing analysis on your code to detect duplications, bugs, security vulnerabilities and code smells on programming languages.

Jenkins :

Jenkins an open-source automation tool is created using Java programming language. For the initial setup, it facilitates users with CI/CD(continuous integration (CI) or continuous delivery) technique that simplifies the use and management of processes. It is fundamentally focused on continuously



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

(All Branches NBA Accredited)



building and testing software projects for developers and to implement changes in real-time. In addition, it also allows users to plan a new build whenever the need arises.

Steps:

- 1) Install and configure a Jenkins and SonarQube CICD environment using Docker containers.
- 2) Configure Jenkins with the SonarQube Scanner plugin for automated static code analysis.
- 3) Create and set up a Jenkins build pipeline using a Jenkinsfile stored within a GitHub repo.
- 4) Use the SonarQube web application to examine and review the generated static analysis report.
- 5) Use the Blue Ocean Plugin to review Pipeline Steps.

Note: From Step 1 and 2 we have already done in Expt. 7 as a Pre-requisite required for Integration settings of Jenkins SAST with SonarQube so in this Experiment we will continue from 3rd Step.

Check the contents of jenkins-sonarqube repository which we are using for Pipeline Project.

vishal003 / jenkins-sonarqube
forked from devopshint/jenkins-sonarqube

Code Pull requests Actions Projects Wiki Security Insights Settings

main · jenkins-sonarqube / project /

This branch is 6 commits ahead of devopshint/main.

Go to file Add file ...

Contribute Fetch upstream

vishal003 Update sonar-analysis · e7555eb · 1 hour ago · History

src · Updater HelloWorldTest.java · 1 hour ago

.gitignore · new · 2 months ago

pom.xml · Update pom.xml · 2 hours ago

sonar-analysis · Update sonar-analysis · 1 hour ago



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

(All Branches NBA Accredited)



Check path for the Source and Test Java Programs from repository

The screenshot shows a Jenkins pipeline interface. At the top, there's a dropdown menu set to 'main' and a blue link 'jenkins-sonarqube / project / src /'. Below this, a message says 'This branch is 6 commits ahead of devopshint:main.' The main area displays a commit from 'vishal003' with the message 'Update HelloWorldTest.java'. The commit details show two changes: 'main/java' was updated with 'Update HelloWorld.java' and 'test/java' was updated with 'Update HelloWorldTest.java'.

Provide sonar host as <http://127.0.0.1:9000> in POM.xml which is available in Project on Github.

```
<activation>
    <activeByDefault>true</activeByDefault>
</activation>
<properties>
    <!-- Optional URL to server. Default value is http://localhost:9000 -->
    <sonar.host.url>
        http://127.0.0.1:9000/
    </sonar.host.url>
</properties>
</profile>
<profile>
    <id>coverage</id>
    <activation>
```

To integrate the SonarQube Scanner in the Jenkins Pipeline. For the same, we are going to add one more stage in the Jenkinsfile called SonarQube and inside that, I am adding the following settings and code.



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

(All Branches NBA Accredited)



Enter an item name

SonarQube.

» Required field



Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.



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.



Bitbucket Team/Project

Scans a Bitbucket Cloud Team (or Bitbucket Server Project) for all repositories matching some defined markers.



Folder

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

Github Repository Configuration in Jenkins Pipeline Project

General Build Triggers Advanced Project Options Pipeline

Description

Hello Pipeline job

[Plain text] [Preview](#)

Discard old builds [?](#)

Do not allow concurrent builds [?](#)

Do not allow the pipeline to resume if the controller restarts [?](#)

GitHub project [?](#)

Project url



A. P. SHAH INSTITUTE OF TECHNOLOGY

(All Branches NBA Accredited)



Pipeline Script where stages are written along with scanner tool, repository path for source and test Java sample program, SonarQube Credential for integration, Application name on sonarqube, code language,etc.

Pipeline

Definition

Pipeline script

```
1 node
2 ▼ {
3     stage('clonning from GIT'){
4         git branch: 'main', credentialsId: 'GIT_REPO', url: 'https://github.com/vishal003/jenkins-sonarqube.git'
5     }
6
7 ▼ stage('SonarQube Analysis') {
8     def scannerHome = tool 'SonarQube'
9     withSonarQubeEnv('SonarQube') {
10         sh """/var/lib/jenkins/tools/hudson.plugins.sonar.SonarRunnerInstallation/SonarQube/bin/sonar-scanner \
11             -D sonar.projectVersion=1.0-SNAPSHOT \
12             -D sonar.login=admin \
13             -D sonar.password=India@11 \
14             -D sonar.projectBaseDir=/var/lib/jenkins/workspace/sonarqube \
15             -D sonar.projectKey=my-app1 \
16             -D sonar.sourceEncoding=UTF-8 \
17             -D sonar.language=java \
18             -D sonar.sources=project/src/main/java \
19             -D sonar.tests=project/src/test/java \
20             -D sonar.host.url=http://127.0.0.1:9000"""
21     }
22 }
```

After creating a Pipeline Script Build it in Jenkins , Click on save and then Click on Build Now



PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY
(All Branches NBA Accredited)



Jenkins

Dashboard → sonarqube → #7

[Back to Project](#)

[Status](#)

[Changes](#)

[Console Output](#)

[View as plain text](#)

[Edit Build Information](#)

[Delete build '#7'](#)

[Git Build Data](#)

[Open Blue Ocean](#)

[Replay](#)

[Pipeline Steps](#)

[Workspaces](#)

[Previous Build](#)

Console Output

```
Started by user unknown or anonymous
Replayed #6
Running in Durability level: MAX_SURVIVABILITY
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/sonarqube
[Pipeline] {
[Pipeline] stage
[Pipeline] { (clonning from GIT)
[Pipeline] git
The recommended git tool is: NONE
Warning: CredentialId "GIT_REPO" could not be found.
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/sonarqube/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/vishal003/jenkins-sonarqube.git # timeout=10
Fetching upstream changes from https://github.com/vishal003/jenkins-sonarqube.git
> git --version # timeout=10
> git --version # 'git version 2.17.1'
> git fetch --tags --progress -- https://github.com/vishal003/jenkins-sonarqube.git +refs/heads/*:refs/
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 80c34f4818e25f7733e50784c2f7639d9884ed90 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 80c34f4818e25f7733e50784c2f7639d9884ed90 # timeout=10
> git branch -a -v --no-abbrev # timeout=10
> git branch -D main # timeout=10
> git checkout -b main 80c34f4818e25f7733e50784c2f7639d9884ed90 # timeout=10
Commit message: "Update README.md"
> git rev-list --no-walk ea3f635e2cee7b1e2d8b1fedf33942709611ea38 # timeout=10
[Pipeline]
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (SonarQube Analysis)
```

Click on Console Output to check output whether build is successful or not.



Click on Pipeline Steps to check Sequence of events during building of pipeline.

Dashboard > sonarqube > #7 > Pipeline Steps

Step	Arguments	Status
Start of Pipeline - (1 sec in block)		✓
Allocate node : Start - (1 sec in block)		✓
Allocate node : Body : Start - (1 sec in block)		✓
Stage : Start - (1.1 sec in block)	clonning from GIT	✓
clonning from GIT - (1 sec in block)		✓
Git - (1 sec in self)		✓
Stage : Start - (10 sec in block)	SonarQube Analysis	✓
SonarQube Analysis - (10 sec in block)		✓
Use a tool from a predefined Tool Installation - (0.18 sec in self)	SonarQube	✓
Prepare SonarQube Scanner environment : Start - (10 sec in block)	SonarQube	✓
General Build Wrapper : Body : Start - (9.9 sec in block)		✓
Shell Script - (9.8 sec in self)	/var/lib/jenkins/tools/hudson.plugins.sonar.SonarRunnerInstallation/SonarQube/bin/sonar-scanner -D sonar.projectVersion=1.0-SNAPSHOT -D sonar.login=admin -D sonar.password=India@11 -D sonar.projectBaseDir=/var/lib/jenkins/workspace/sonarqube -D sonar.projectKey=my-app1 -D sonar.sourceEncoding=UTF-8 -D sonar.language=java -D sonar.sources=project/src/main/java -D sonar.tests=project/src/test/java -D sonar.host.url=http://127.0.0.1:9000/	✓

Also you can use Blue Ocean to check Pipeline execution stage by stage and log of the pipeline too.

127.0.0.1:8080/blue/organizations/jenkins/sonarqube/detail/sonarqube/7/pipeline/

Branch: - 12s No changes
Commit: - 32 minutes ago Replied #6

Pipeline Changes Tests Artifacts

SonarQube Analysis - 10s

- > SonarQube - Use a tool from a predefined Tool Installation <1s
- > /var/lib/jenkins/tools/hudson.plugins.sonar.SonarRunnerInstallation/SonarQube/bin/sonar-scanner -D sonar.projectVersion=1.0-SNAPSHOT -D sonar.login=admin -D sonar.password=India@11 ... - Shell Script 10s



If you login to the SonarQube and visit the Dashboard, you will see the Analysis of the project there.

The screenshot shows the SonarQube dashboard with the following details:

- Project:** my-app1 (Passed)
- Last analysis:** 1 hour ago
- Metrics:**
 - Bugs: 0
 - Vulnerabilities: 0
 - Hotspots Reviewed: 0
 - Code Smells: 3
 - Coverage: 0.0%
 - Duplications: 0.0%
 - Lines: 8 (Java)
- Filters:**
 - Quality Gate:** Passed (1), Failed (0)
 - Reliability (Bug):** A (1), B (0), C (0), D (0), E (0)
 - Security (Vulnerability):** A (1), B (0), C (0), D (0), E (0)
 - Security Review (Security Hotspots):** A (1), B (0), C (0), D (0), E (0)

For Detailed Report for code analysis you can go to application overview and check for all Bugs, Vulnerabilities, code smells and all parameters as shown in below image.

The screenshot shows the SonarQube application overview for 'my-app1' with the following details:

- Quality Gate Status:** Passed (All conditions passed)
- Measures:**
 - New Code: Since July 29, 2021, Started 1 hour ago
 - Overall Code
 - 0 Bugs
 - 0 Vulnerabilities
 - 0 Security Hotspots
 - 3 Code Smells
 - 30min Debt
 - 0.0% Coverage on 4 Lines to cover
 - 0 Unit Tests
 - 0.0% Duplications on 8 Lines
 - 0 Duplicated Blocks
- Project Settings & Information:** Last analysis had 1 warning, July 29, 2021, 11:42 PM, Version 1.0-SNAPSHOT, Project Settings, Project Information



Since we have both Jenkins and SonarQube in the Enterprise standard, we have a lot of features including the alert system. Where we can configure the Email, or Instance message Notification system for the findings in the SonarQube or Jenkins. In the best case, we can auto convert certain bugs or findings as ticket and assign to the respective developer as a one option.

Conclusion: Write your own findings.



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



Academic Year: 2021-22 Semester: V

Class / Branch: TE IT

Subject: Advanced Devops Lab (ADL)

Subject Lab Incharge: Prof. Manjusha Kashikar

EXPERIMENT NO. 09

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

Theory:

Continuous monitoring is a process to detect, report, respond all the attacks which occur in its infrastructure. Once the application is deployed into the server, the role of continuous monitoring comes in to play. The entire process is all about taking care of the company's infrastructure and respond appropriately.

Why We Need Nagios tool?

Here, are the important reasons to use Nagios monitoring tool:

- Detects all types of network or server issues
 - Helps you to find the root cause of the problem which allows you to get the permanent solution to the problem
 - Active monitoring of your entire infrastructure and business processes
 - Allows you to monitors and troubleshoot server performance issues
 - Helps you to plan for infrastructure upgrades before outdated systems create failures
 - You can maintain the security and availability of the service
 - Automatically fix problems in a panic situation
-
- Nagios is the most popular, open source, powerful monitoring system for any kind of infrastructure. It enables organizations to identify and resolve IT infrastructure problems before they affect critical business processes. Nagios has the capability of monitoring application, services, entire IT infrastructure.
-
- NRPE is known as **Nagios Remote Plugin Executor**. The NRPE add-on is designed to execute plugins on remote Nix systems. In this setup, NRPE daemon is installed on the remote system to which services need to monitor through Nagios server.
 - NRPE runs as a daemon on remote systems and waits for Nagios requests. When Nagios server needs to check the status of any resources or applications to that remote host, sends



PARSHVANATH CHARITABLE TRUST'S

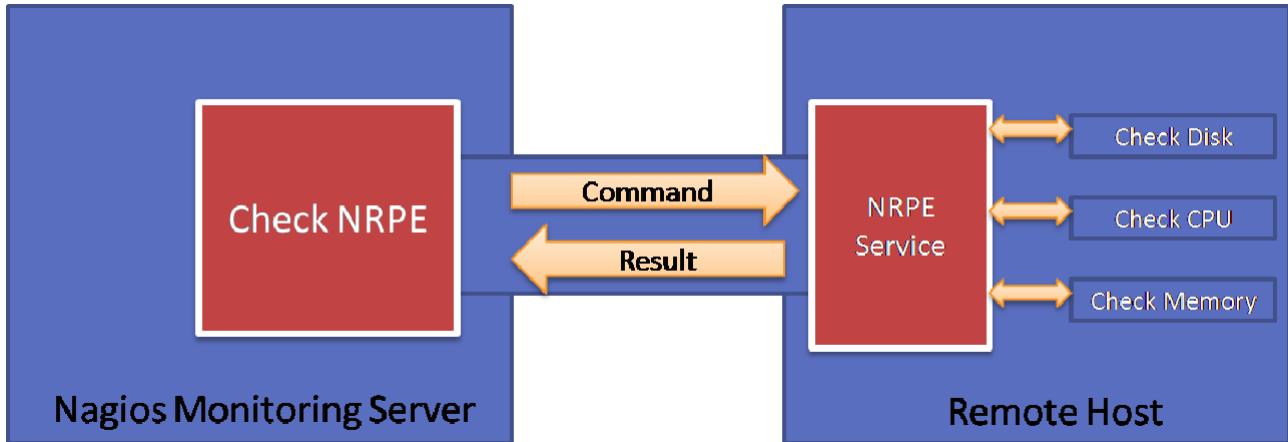
A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



and commands signal, which command definition is stored on NRPE service. NRPE takes Nagios server request and execute the command on the local system and sends the result back to Nagios.



1 - Pre-requisite

First requirement is to install Apache and PHP first. Use the following commands to complete it. And use commands to install required packages for Nagios.

```
manjusha@apsit:~$ sudo apt-get update  
manjusha@apsit:~$ sudo apt-get install wget build-essential unzip openssl libssl-dev  
manjusha@apsit:~$ sudo apt-get install apache2 php libapache2-mod-php php-gd libgd-dev
```

2 – Create Nagios User

Create a new user account for Nagios in your system and assign a password.
manjusha@apsit:~\$ sudo adduser nagios

Now create a group for Nagios setup “nagcmd” and add nagios user to this group. Also, add nagios user in the Apache group.

```
manjusha@apsit:~$ sudo groupadd nagcmd  
manjusha@apsit:~$ sudo usermod -a -G nagcmd nagios  
manjusha@apsit:~$ sudo usermod -a -G nagcmd www-data
```



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



Step 3 – Install Nagios Core Service

After installing required dependencies and adding user accounts and Nagios core installation. Download latest Nagios core service from the official site.

```
manjusha@apsit:~$cd /opt/
```

```
manjusha@apsit:~$sudo wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.3.tar.gz
```

```
manjusha@apsit:~$sudo tar xzf nagios-4.4.3.tar.gz
```

After extracting navigate to nagios source directory and install using make command.

```
manjusha@apsit:~$cd nagios-4.4.3
```

```
manjusha@apsit:~$sudo ./configure --with-command-group=nagcmd  
manjusha@apsit:~$sudo make all  
manjusha@apsit:~$sudo make install  
manjusha@apsit:~$sudo make install-init  
manjusha@apsit:~$sudo make install-daemoninit
```

```
manjusha@apsit:~$sudo make install-config  
manjusha@apsit:~$sudo make install-commandmode  
manjusha@apsit:~$sudo make install-exfoliation
```

Now copy event handlers scripts under libexec directory. These binaries provides multiple events triggers for your Nagios web interface.

```
manjusha@apsit:~$sudo cp -R contrib/eventhandlers/ /usr/local/nagios/libexec/
```

```
manjusha@apsit:~$sudo chown -R nagios:nagios /usr/local/nagios/libexec/eventhandlers
```

Step 4 – Setup Apache with Authentication

Now create an Apache configuration file for your Nagios server as below:

```
manjusha@apsit:~$sudo nano /etc/apache2/conf-available/nagios.conf
```

Add below lines to nagios.conf file.

```
ScriptAlias /nagios/cgi-bin "/usr/local/nagios/sbin"  
<Directory "/usr/local/nagios/sbin"> Options ExecCGI  
AllowOverride None
```



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



Order allow,deny Allow from all

AuthName "Restricted Area" AuthType Basic

AuthUserFile /usr/local/nagios/etc/htpasswd.users Require valid-user

</Directory>

Alias /nagios "/usr/local/nagios/share"

<Directory "/usr/local/nagios/share"> Options None

AllowOverride None Order allow,deny Allow from all

AuthName "Restricted Area" AuthType Basic

AuthUserFile /usr/local/nagios/etc/htpasswd.users Require valid-user

</Directory>

To setup apache authentication for user **nagiosadmin**

```
manjusha@apsit:~$sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

Enable Apache configuration and restart Apache service to make the new settings take effect.

```
manjusha@apsit:~$sudo a2enconf nagios manjusha@apsit:~$sudo a2enmod cgi rewrite  
manjusha@apsit:~$sudo service apache2 restart
```

Step 5 – Installing Nagios Plugins

After installing and configuring Nagios core service, Download latest nagios-plugins source and install using following commands.

```
manjusha@apsit:~$cd /opt
```

```
manjusha@apsit:~$sudo wget http://www.nagios-plugins.org/download/nagios-plugins- 2.2.1.tar.gz
```

```
manjusha@apsit:~$sudo tar xzf nagios-plugins-2.2.1.tar.gz manjusha@apsit:~$cd nagios-plugins-  
2.2.1
```

Now compile and install Nagios plugins

```
manjusha@apsit:~$sudo ./configure --with-nagios-user=nagios --with-nagios-group=nagios  
--with-openssl
```

```
manjusha@apsit:~$sudo make
```

```
manjusha@apsit:~$sudo make install
```



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



Step 6 – Verify Settings

Use the Nagios commands to verify the Nagios installation and configuration file. After successfully verify start the Nagios core service.

```
manjusha@apsit:~$ /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg  
manjusha@apsit:~$ sudo service nagios start
```

Also configure Nagios to auto start on system boot.

Step 7 – Access Nagios Web Interface

Access your nagios setup by access nagios server using hostname or ip address followed by /nagios.

<http://127.0.0.1/nagios/>

Prompting for Apache Authentication Password –

username: nagiosadmin

Password : 123456 (which you enter while configuration)

Nagios After login screen –

The screenshot shows the Nagios Core web interface at <http://127.0.0.1/nagios/>. The top navigation bar includes links for Home, Documentation, and Current Status. The Current Status section displays a green checkmark indicating "Daemon running with PID 18593". The main content area features three large cards: "Nagios XI" (Easy Configuration Advanced Reporting), "Nagios Log Server" (Monitor and analyze logs from anywhere), and "Nagios Network Analyzer" (Real-time netflow and bandwidth analysis). Below these cards are sections for "Get Started", "Quick Links", "Latest News", and "Don't Miss...". The "Get Started" section lists items like "Start monitoring your infrastructure", "Change the look and feel of Nagios", and "Extend Nagios with hundreds of addons". The "Quick Links" section provides links to Nagios Library, Labs, Exchange, Support, and other resources. The "Latest News" and "Don't Miss..." sections contain links to specific news articles.



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



We have successfully installed and configured Nagios Monitoring Server core service in our system now we need to install NRPE on all remote Linux systems to monitor with Nagios.

Conclusion: Write your own findings.



Academic Year: 2022-23

Semester: V

Class / Branch: TE IT

Subject: Advanced Devops Lab (ADL)

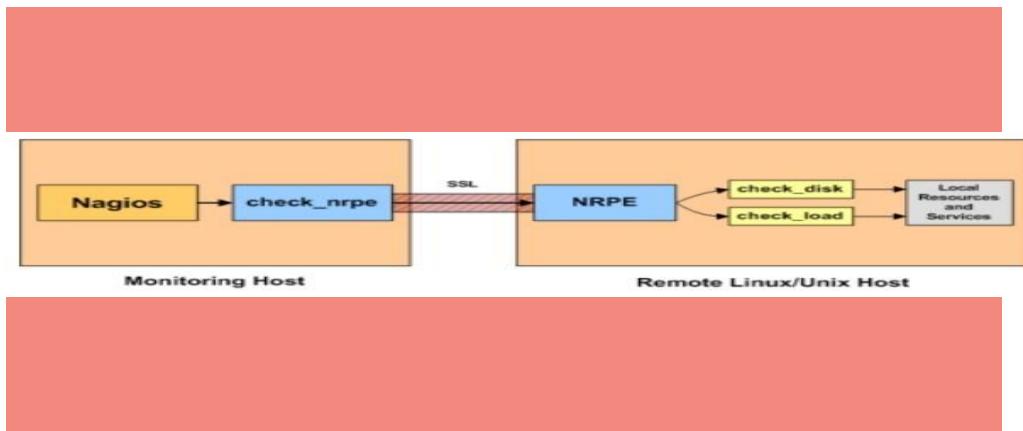
Subject Lab Incharge: Prof.Manjusha K.

EXPERIMENT NO. 10

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

Theory:

Monitoring remote Linux/Unix hosts is to use the NRPE addon. NRPE allows you to execute plugins on remote Linux/Unix hosts. This is useful if you need to monitor local resources/attributes like disk usage, CPU load, memory usage, etc. on a remote host.



Note: To perform this experiment Experiment 9 is pre-requisite where we have configured Nagios on Linux System. Here In this Experiment we will Add a Linux Host to Nagios for Monitoring purpose.

Step 1 – Configure NRPE on Linux Host

Follow the below steps to install and configure NRPE on client machine and check connectivity with Nagios server.

Step 1.1 – Install NRPE

```
vishal@apsit:~$ sudo apt-get install nagios-nrpe-server nagios-plugins
```

Step 1.2 – Configure NRPE

After successfully installing NRPE service, Edit nrpe configuration file /etc/nagios/nrpe.cfg in your favorite editor and add your nagios service ip in allowed hosts.



```
vishal@apsit:~$ sudo nano /etc/nagios/nrpe.cfg
```

```
allowed_hosts=127.0.0.1, 192.168.64.3, 192.168.1.100
```

Where **192.168.1.100** is your Nagios server ip address.

After making above changes in nrpe configuration file, Lets restart NRPE service as per your system

```
vishal@apsit:~$ sudo /etc/init.d/nagios-nrpe-server restart
```

Step 1.3 – Verify Connectivity from Nagios

Now run the below command from Nagios server to make sure your nagios is able to connect nrpe client on remote Linux system. Here **192.168.64.3** is your remote Linux system ip.

```
vishal@apsit:~$ /usr/local/nagios/libexec/check_nrpe -H 192.168.64.3
NRPE v2.15
```

Step 2 – Add Linux Host in Nagios

First create a configuration file using below values. for example you Linux hosts ip is . We also need to define a service with host. So add a ping check service, which will continuously check that host is up or not.

```
vishal@apsit:~$ sudo nano /usr/local/nagios/etc/servers/MyLinuxHost001.cfg
```

```
define host {
    use                      linux-server
    host_name                Linux_Host_001
    alias                     Linux Host 001
    address                  192.168.64.3
    register                  1
}
define service{
    host_name                Linux_Host_001
    service_description        PING
    check_command              check_ping!100.0,20%!500.0,60%
    max_check_attempts          2
    check_interval             2
}
```



```
retry_interval          2
check_period            24x7
check_freshness         1
contact_groups          admins
notification_interval   2
notification_period     24x7
notifications_enabled   1
register                1
}
```

Now verify configuration files using following command. If there are no errors found in configuration, restart nagios service.

```
vishal@apsit:~$ sudo nagios -v /usr/local/nagios/etc/nagios.cfg
vishal@apsit:~$ sudo service nagios restart
```

Step 3 – Check Host in Nagios Web Interface

Open your Nagios web interface and check for new Linux hosts added in Nagios core service.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
APSSIT	CURRENT LOAD	OK	11-08-2013 04:17:03	24d 18h 3m 36s	1/1	OK - host average: 0.00, 0.00, 0.00
APSSIT	CURRENT USERS	OK	11-08-2013 04:17:37	24d 18h 3m 43s	1/1	USERS OK - 1 user currently logged in.
HTTP	HTTP	WARNING	11-08-2013 04:18:19	24d 18h 3m 2s	4/4	HTTP WARNING: HTTP/1.1 403 Forbidden - 8267 bytes in 0.001 second response time.
PNR2	PNR2	OK	11-08-2013 04:18:54	24d 18h 3m 45s	1/1	PNR2 OK - Packet loss = 0%, RTT = 0.03 ms
Root Partition	Root Partition	OK	11-08-2013 04:19:32	24d 18h 3m 15s	1/1	DISK OK - free space: 1442 MB (98% used<94%).
SSH	SSH	OK	11-08-2013 04:19:19	24d 18h 3m 37s	1/1	SSH OK - OpenSSH_5_3 (protocol 2.0)
Swap Usage	Swap Usage	OK	11-08-2013 04:19:05	24d 18h 3m 6s	1/1	SWAP OK - 100% free (128 MB out of 256 MB)
Total Processes	Total Processes	OK	11-08-2013 04:19:54	24d 18h 3m 23s	1/1	PROCS-OK - 95 processes with STATE = R/SZDT
APSSIT	CPU Load	OK	11-08-2013 04:19:01	0d 0h 1m 48s	1/1	OK - host average: 0.00, 0.00, 0.00
APSSIT	Current Users	OK	11-08-2013 04:19:01	0d 0h 1m 48s	1/1	USERS OK - 1 user currently logged in.
SSH Monitoring	SSH Monitoring	OK	11-08-2013 04:19:51	0d 0h 1m 18s	1/1	SSH OK - OpenSSH_5_3 (protocol 2.0)
Total Processes	Total Processes	OK	11-08-2013 04:19:58	0d 0h 1m 12s	1/1	PROCS-OK - 95 processes with STATE = R/SZDT

Conclusion: Write your own findings.



PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY
Department of Information Technology
(NBA Accredited)





PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



Academic Year: 2023-24

Semester: V

Class / Branch: TE IT

Subject: Advanced Devops Lab (ADL)

Subject Lab Incharge: Prof. Manjusha Kashikar

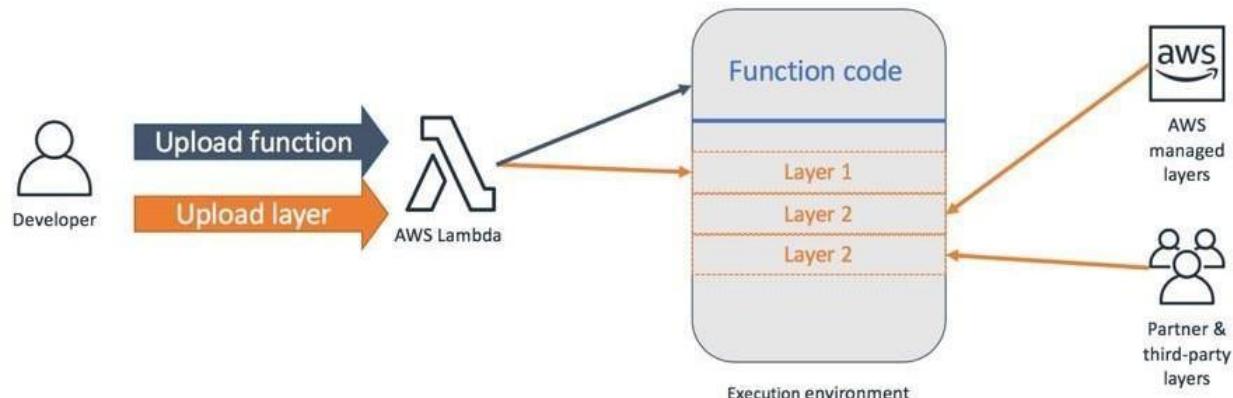
EXPERIMENT NO. 11

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

Theory:

Lambda is a compute service that lets you run code without provisioning or managing servers. Lambda runs your code on a high-availability compute infrastructure and performs all of the administration of the compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, code monitoring and logging. With Lambda, you can run code for virtually any type of application or backend service. All you need to do is supply your code in one of the languages that Lambda supports.

You organize your code into Lambda functions. Lambda runs your function only when needed and scales automatically, from a few requests per day to thousands per second. You pay only for the compute time that you consume—there is no charge when your code is not running.





PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



You can invoke your Lambda functions using the Lambda API, or Lambda can run your functions in response to events from other AWS services. For example, you can use Lambda to:

- Build data-processing triggers for AWS services such as Amazon Simple Storage Service (Amazon S3) and Amazon DynamoDB.
- Process streaming data stored in Amazon Kinesis.
- Create your own backend that operates at AWS scale, performance, and security.

What is a Lambda function?

The code you run on AWS Lambda is called a “Lambda function.” After you create your Lambda function, it is always ready to run as soon as it is triggered, similar to a formula in a spreadsheet. Each function includes your code as well as some associated configuration information, including the function name and resource requirements. Lambda functions are “stateless”, with no affinity to the underlying infrastructure, so that Lambda can rapidly launch as many copies of the function as needed to scale to the rate of incoming events.

After you upload your code to AWS Lambda, you can associate your function with specific AWS resources, such as a particular Amazon S3 bucket, Amazon DynamoDB table, Amazon Kinesis stream, or Amazon SNS notification. Then, when the resource changes, Lambda will execute your function and manage the compute resources as needed to keep up with incoming requests.

AWS Lambda is a serverless compute service that lets you run code without provisioning or managing servers, creating workload-aware cluster scaling logic, maintaining event integrations, or managing runtimes. With Lambda, you can run code for virtually any type of application or backend service - all with zero administration. Just upload your code as a ZIP file or container image, and Lambda automatically and precisely allocates compute execution power and runs your code based on the incoming request or event, for any scale of traffic. You can set up your code to automatically trigger from over 200 AWS services and SaaS applications or call it directly from any web or mobile app. You can write Lambda functions in your favorite language (Node.js, Python,



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



Go, Java, and more) and use both serverless and container tools, such as AWS SAM or Docker CLI, to build, test, and deploy your functions.

Benefits

1. No servers to manage

AWS Lambda automatically runs your code without requiring you to provision or manage infrastructure. Just write the code and upload it to Lambda either as a ZIP file or container image.

2. Continuous scaling

AWS Lambda automatically scales your application by running code in response to each event. Your code runs in parallel and processes each trigger individually, scaling precisely with the size of the workload, from a few requests per day, to hundreds of thousands per second.

3. Cost optimized with millisecond metering

With AWS Lambda, you only pay for the compute time you consume, so you're never paying for over-provisioned infrastructure. You are charged for every millisecond your code executes and the number of times your code is triggered. With Compute Savings Plan, you can additionally save up to 17%.

4. Consistent performance at any scale

With AWS Lambda, you can optimize your code execution time by choosing the right memory size for your function. You can also keep your functions initialized and hyper-ready to respond within double digit milliseconds by enabling Provisioned Concurrency.



Steps: First Lambda functions using Python

1. Open Aws Console and search for Lambda Service and open home screen of Lambda.

The screenshot shows the AWS Lambda service page. At the top, there's a search bar with 'lambda'. Below the search bar, a sidebar lists 'Services (5)', 'Features (2)', 'Documentation (47,194)', 'Knowledge Articles (30)', and 'Marketplace (175)'. The main area is titled 'Search results for "lambda"' and shows a list of services: 'Lambda' (selected, highlighted in blue) and 'CodeBuild'. To the right, there's a message about being connected to AWS resources and a note about the mobile app.

2. Choose region in which you need to create Lambda function as it is region specific.

The screenshot shows the 'Region' dropdown menu in the AWS Lambda service. The menu lists various regions: US East (N. Virginia) us-east-1, US East (Ohio) us-east-2, US West (N. California) us-west-1, US West (Oregon) us-west-2, Africa (Cape Town) af-south-1, Asia Pacific (Hong Kong) ap-east-1, **Asia Pacific (Mumbai) ap-south-1** (highlighted in yellow), Asia Pacific (Osaka) ap-northeast-3, and Asia Pacific (Seoul) ap-northeast-2. The dropdown is currently set to Mumbai.



3. Create sum as a Lambda Function in Python Language so select latest version of Python and choose role with basic Lambda Permission to allow cloudwatch for monitoring.

Lambda > Functions > Create function

Create function Info

Choose one of the following options to create your function.

- Author from scratch** Start with a simple Hello World example.
- Use a blueprint** Build a Lambda application from sample code and configuration presets for common use cases.
- Container image** Select a container image to deploy for your function.
- Browse serverless app repository** Deploy a sample Lambda application from the AWS Serverless Application Repository.

Basic information

Function name Info
Enter a name that describes the purpose of your function.
sum
Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Python 3.8

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

▼ Change default execution role

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

- Create a new role with basic Lambda permissions
- Use an existing role

4. Lambda sum function is created successfully

aws Services ▾ Search for services, features, marketplace products, and docs [Alt+S] Vishal Badgjar Mumbai Sup

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

Lambda > Functions > sum

sum

▼ Function overview Info

Code source Info

File Edit Find View Go Tools Window Test Deploy Changes deployed

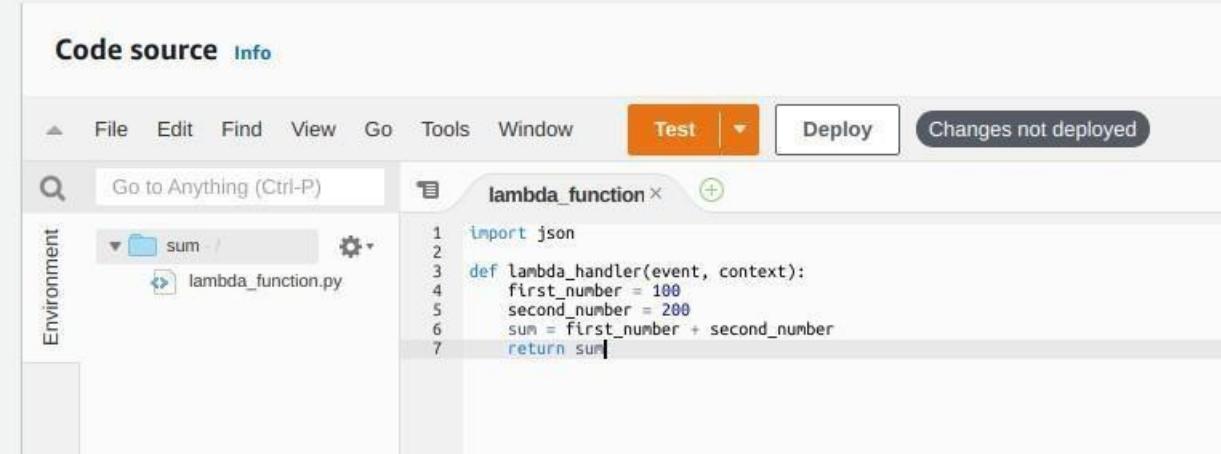
Environment: sum

lambda_function.py

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

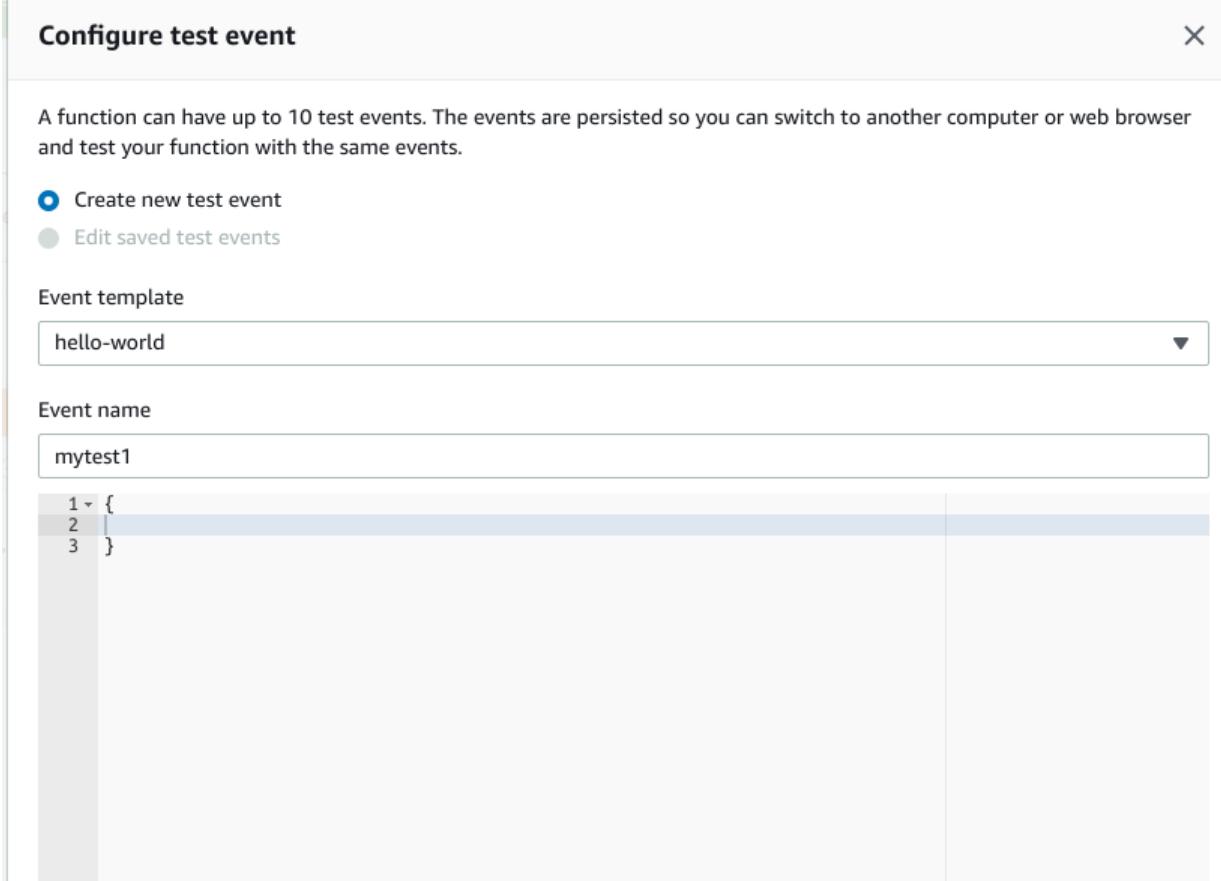


5. Write a sample python code for sum of two numbers:



```
Code source Info
File Edit Find View Go Tools Window Test Deploy Changes not deployed
Go to Anything (Ctrl-P)
Environment
sum/
lambda_function.py
lambda_function.x
import json
def lambda_handler(event, context):
    first_number = 100
    second_number = 200
    sum = first_number + second_number
    return sum
```

6. Configure Test Event in Json Format



Configure test event

A function can have up to 10 test events. The events are persisted so you can switch to another computer or web browser and test your function with the same events.

Create new test event
 Edit saved test events

Event template

hello-world

Event name

mytest1

```
1 {  
2 }  
3 }
```



PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY
Department of Information Technology
(NBA Accredited)



Code source [Info](#)

File Edit Find View Go Tools Window **Test** Deploy Changes deployed

Go to Anything (Ctrl-P) sum1.py Execution result

Environment sum / sum1.py

Execution results
Test Event Name sum
Response 300

Function Logs
START RequestId: bc4cb31a-e6ff-42e3-8804-0e3d78afa7d6 Version: \$LATEST
END RequestId: bc4cb31a-e6ff-42e3-8804-0e3d78afa7d6
REPORT RequestId: bc4cb31a-e6ff-42e3-8804-0e3d78afa7d6 Duration: 1.07 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 51 MB Init Duration: 125.07 ms

Request ID
bc4cb31a-e6ff-42e3-8804-0e3d78afa7d6

Write a sample Second sample python Code:

Code source [Info](#)

File Edit Find View Go Tools Window **Test** Deploy Changes deployed

Go to Anything (Ctrl-P) lambda_function Execution results

Environment sum / lambda_function.py

```
def lambda_handler(event, context):
    if event["name"] == "vishal":
        return "apsit"
```



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



Configure Test Event

Configure test event



A function can have up to 10 test events. The events are persisted so you can switch to another computer or web browser and test your function with the same events.

- Create new test event
- Edit saved test events

Saved Test Events

vishal1



```
1 {  
2   "name": "vishal"  
3 }
```

If condition met returns a value as apsit

▼ Execution results

Status: Succeeded !

Test Event Name vishal1	Response "apsit"
Function Logs	
START RequestId: cddd9b6e-59b3-4457-be2b-04e4784e5c3e Version: \$LATEST	
END RequestId: cddd9b6e-59b3-4457-be2b-04e4784e5c3e	
REPORT RequestId: cddd9b6e-59b3-4457-be2b-04e4784e5c3e Duration: 1.12 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 50 MB Init Duration: 124.74 ms	
Request ID cddd9b6e-59b3-4457-be2b-04e4784e5c3e	

Conclusion: Write your own findings.



Academic Year: 2022-23

Semester: V

Class / Branch: TE IT

Subject: Advanced Devops Lab (ADL)

Subject Lab Incharge: Prof. Manjusha K.

EXPERIMENT NO. 12

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

Theory:

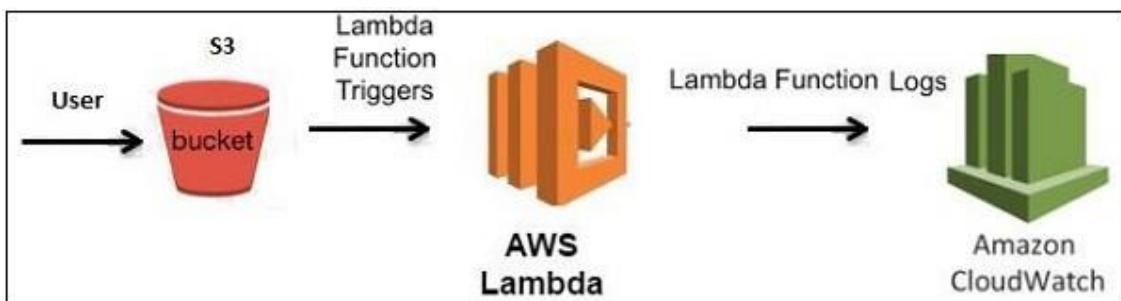
Using Lambda Function with Amazon S3

Amazon S3 service is used for file storage, where you can upload or remove files. We can trigger AWS Lambda on S3 when there are any file uploads in S3 buckets. AWS Lambda has a handler function which acts as a start point for AWS Lambda function. The handler has the details of the events. In this chapter, let us see how to use AWS S3 to trigger AWS Lambda function when we upload files in S3 bucket.

Steps for Using AWS Lambda Function with Amazon S3

To start using AWS Lambda with Amazon S3, we need the following –

- Create S3 Bucket
- Create role which has permission to work with s3 and lambda
- Create lambda function and add s3 as the trigger.





Let us see these steps with the help of an example which shows the basic interaction between Amazon S3 and AWS Lambda

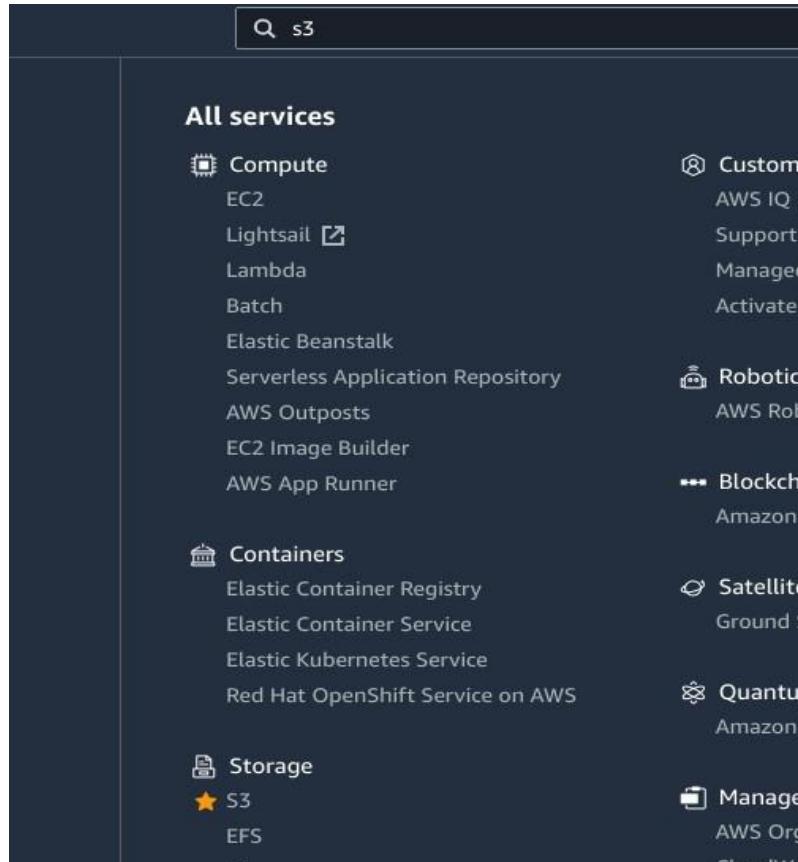
- User will upload a file in Amazon S3 bucket
- Once the file is uploaded, it will trigger AWS Lambda function in the background which will display an output in the form of a console message that the file is uploaded.
- The user will be able to see the message in Cloudwatch logs once the file is uploaded.

Creating S3 Bucket

Let us start first by creating a s3 bucket in AWS console using the steps given below –

Step 1

Go to Amazon services and click **S3** in storage section as highlighted in the image given below –





Step 2

Click S3 storage and **Create bucket** which will store the files uploaded.

The screenshot shows the Amazon S3 console. At the top, there is a 'Account snapshot' section with a link to 'View Storage Lens dashboard'. Below it, the 'Buckets' list is shown with three items. To the right of the list are several buttons: a refresh icon, 'Copy ARN', 'Empty', 'Delete', and a prominent orange 'Create bucket' button. Below the list, there is a note: 'Buckets are containers for data stored in S3. Learn more'.

Step 3

Once you click Create bucket button, you can see a screen as follows –

The screenshot shows the 'Create bucket' wizard. The title is 'Create bucket' with an 'Info' link. Below it, a note says 'Buckets are containers for data stored in S3. Learn more'. The main section is titled 'General configuration'. It contains fields for 'Bucket name' (with 'myawsbucket' entered), 'AWS Region' (set to 'Asia Pacific (Mumbai) ap-south-1'), and a section for 'Copy settings from existing bucket - optional' (with a 'Choose bucket' button). There is also a note: 'Bucket name must be unique and must not contain spaces or uppercase letters. See rules for bucket naming'.



Step 4

Enter the details Bucket name, Select the Region and click Create button at the bottom left side.
Thus, we have created bucket with name :

<input type="radio"/> lambdawiths3	Asia Pacific (Mumbai) ap-south-1	Bucket and objects not public	August 3, 2021, 11:22:23 (UTC+05:30)
------------------------------------	----------------------------------	-------------------------------	--------------------------------------

Step 5

Now, click the bucket name and it will ask you to upload files as shown below –

The screenshot shows the AWS S3 console interface for the 'lambdawiths3' bucket. The 'Objects' tab is active, displaying a message stating 'Objects (0)'. Below this, there are buttons for actions like Copy S3 URI, Copy URL, Download, Open, Delete, Actions, and Create folder. An orange 'Upload' button is prominently displayed. A search bar labeled 'Find objects by prefix' is present. At the bottom, a table header for object listing includes columns for Name, Type, Last modified, Size, and Storage class. A message 'No objects' is centered at the bottom of the list area.

Thus, we are done with bucket creation in S3.



Create Role that Works with S3 and Lambda

To create role that works with S3 and Lambda, please follow the Steps given below –

Step 1

Go to AWS services and select IAM as shown below –

The screenshot shows the AWS search interface with the search bar containing 'iam'. Below the search bar, it says 'Search results for 'iam''. Under the 'Services' section, there is a card for 'IAM' with the subtext 'Manage access to AWS resources'. There are also other items listed: '(16)' and '(27)'. In the 'Features' section, there is a link 'See all 11 results ▶'.

Step 2

Now, click **IAM -> Roles** as shown below –

The screenshot shows the 'Roles' page under the IAM service. It displays 18 IAM roles. At the top, there is a header with 'Roles (18) Info' and a note about IAM roles. On the right side, there are buttons for 'Create role', 'Delete', and a refresh icon. Below the header, there is a search bar and a navigation bar with page numbers (1, 2, 3). The main area lists the roles, though the names are not clearly legible.



Step 3

Now, click **Create role** and choose the services that will use this role. Select Lambda and click **Permission** button.

Create role

1 2 3 4

Select type of trusted entity

AWS service EC2, Lambda and others	Another AWS account Belonging to you or 3rd party	Web identity Cognito or any OpenID provider	SAML 2.0 federation Your corporate directory
---------------------------------------	--	--	---

Allows AWS services to perform actions on your behalf. [Learn more](#)

Choose a use case

Common use cases

EC2

Allows EC2 instances to call AWS services on your behalf.

Lambda

Allows Lambda functions to call AWS services on your behalf.

Or select a service to view its use cases

API Gateway	CodeBuild	EMR Containers	IoT SiteWise	RDS
AWS Backup	CodeDeploy	ElasticCache	IoT Things Graph	Redshift
AWS Chatbot	CodeGuru	Elastic Beanstalk	KMS	Rekognition
AWS Marketplace	CodeStar Notifications	Elastic Container Registry	Kinesis	RoboMaker
AWS Support	Comprehend	Elastic Container Service	Lake Formation	S3
Amplify	Config	Elastic Transcoder	Lambda	SMS
AppStream 2.0	Connect	ElasticLoadBalancing	Lex	SNS
AppSync	DMS	EventBridge	License Manager	SWF
Application Auto Scaling	Data Lifecycle Manager	Forecast	MQ	SageMaker
Application Discovery	Data Pipeline	GameLift	Machine Learning	Security Hub

* Required

Cancel

Next: Permissions



Step 4

Add the permission from below and click Review.

AmazonS3FullAccess, AWSLambdaFullAccess and CloudWatchFullAccess.

Step 5

Observe that we have chosen the following permissions –

Create role

1 2 3

Review

Provide the required information below and review this role before you create it.

Role name*

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

Role description

Allows Lambda functions to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '+=.,@-' characters.

Trusted entities AWS service: lambda.amazonaws.com

Policies

AmazonS3FullAccess

AWSLambda_FullAccess

CloudWatchFullAccess

Permissions boundary Permissions boundary is not set

No tags were added.

Observe that the Policies that we have selected are **AmazonS3FullAccess, AWSLambdaFullAccess and CloudWatchFullAccess**.



Step 6

Now, enter the Role name, Role description and click Create Role button at the bottom.

<input type="checkbox"/> lambdawiths3service	AWS Service: lambda
--	---------------------

Thus, our role named lambdawiths3service is created.

Create Lambda function and Add S3 Trigger

In this section, let us see how to create a Lambda function and add a S3 trigger to it. For this purpose, you will have to follow the Steps given below –

Step 1

Go to AWS Services and select Lambda as shown below –

The screenshot shows the AWS Lambda service search results. The search bar at the top contains the text 'Lambda'. Below the search bar, the heading 'Search results for 'lambda'' is displayed. Under the heading 'Services', there is a card for 'Lambda' with the subtext 'Run Code without Thinking about Servers'. To the right of the Lambda card, there is a link 'See all 5 results ▶'.



Step 2

Click **Lambda** and follow the process for adding **Name**. Choose the **Runtime**, **Role** etc. and createthe function. The Lambda function that we have created is shown in the screenshot below –

The screenshot shows the 'Create function' wizard in the AWS Lambda console. The top navigation bar shows 'Lambda > Functions > Create function'. The main heading is 'Create function' with an 'Info' link. Below it, a sub-instruction says 'Choose one of the following options to create your function.' Two options are available:

- Author from scratch** (radio button selected): 'Start with a simple Hello World example.'
- Use a blueprint** (radio button unselected): 'Build a Lambda application from sample code and configuration presets for common use cases.'

Below this, the 'Basic information' section is expanded. It includes fields for 'Function name' (containing 'lambdawiths3bucket'), 'Runtime' (set to 'Node.js 14.x'), and 'Permissions' (with a note about default execution role creation). The 'Change default execution role' section is collapsed, showing options for creating a new role or using an existing one ('Use an existing role' is selected). An 'Existing role' field contains 'lambdawiths3service', with a note about viewing the role on the IAM console.

Step 3

Now let us add the S3 trigger.



Lambda > Functions > lambdawiths3bucket

lambdawiths3bucket

Throttle Copy ARN Actions ▾

▶ Function overview Info

Code Test Monitor Configuration Aliases Versions

General configuration Triggers Permissions Destinations Environment variables Tags VPC

Triggers (0)

Find triggers Enable Disable Fix errors Delete Add trigger < 1 >

No triggers

No triggers are configured.

Add trigger

Step 4

Choose the trigger from above and add the details as shown below –

Add trigger

Trigger configuration

S3 aws storage

Bucket
Please select the S3 bucket that serves as the event source. The bucket must be in the same region as the function.
lambdawiths3

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

Prefix - optional
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters.
e.g. images/

Suffix - optional
Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters.
.jpg

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

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

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

Cancel Add



You can add Prefix and File pattern which are used to filter the files added. For Example, to trigger lambda only for .jpg images. as we need to trigger Lambda for all jpg image files uploaded. Click Add button to add the trigger.

Step 5

You can find the trigger display for the Lambda function as shown below –

The screenshot shows the AWS Lambda Functions overview for a function named 'lambdawiths3bucket'. The 'Configuration' tab is selected. On the left, a sidebar lists 'General configuration', 'Triggers' (which is currently selected), 'Permissions', 'Destinations', 'Environment variables', and 'Tags'. The main content area displays a 'Triggers (1)' section with a table. The table has one row, showing a green S3 icon next to the text 'S3: lambdawiths3 arn:aws:s3:::lambdawiths3'. Below this row is a 'Details' link. At the top of the triggers section, there are buttons for 'Throttle', 'Copy ARN', and 'Actions' (with dropdown options). A success message at the top of the page states: 'The trigger lambdawiths3 was successfully added to function lambdawiths3bucket. The function is now receiving events from the trigger.'

Step 6

Let's add the details for the aws lambda function. Here, we will use the online editor to add our code and use nodejs as the runtime environment.

To trigger S3 with AWS Lambda, we will have to use S3 event in the code as shown below –



Lambda > Functions > lambdawiths3bucket

lambdawiths3bucket

The trigger lambdawiths3 was successfully added to function lambdawiths3bucket. The function is now receiving events from the trigger.

Function overview [Info](#)

Code Test Monitor Configuration Aliases Versions

Code source [Info](#)

File Edit Find View Go Tools Window Test Deploy Changes not deployed

index.js

```
1 exports.handler = function(event, context, callback) {
2     console.log("Incoming Event: ", event);
3     const bucket = event.Records[0].s3.bucket.name;
4     const filename = decodeURIComponent(event.Records[0].s3.object.key.replace(/\+/g, ' '));
5     const message = `An Image has been added - ${bucket} -> ${filename}`;
6     console.log(message);
7     callback(null, message);
8 };
```

Upload from

8:3 JavaScript Spaces: 4

Code properties

Package size: 304.0 byte	SHA256 hash: uTJfxT0sQYdBf6CtxoZoBcLT6Hd0A48LnMm4gpXgDw=	Last modified: August 3, 2021, 11:36 AM GMT+5:30
-----------------------------	---	---

Runtime settings [Info](#)

Runtime: Node.js 14.x	Handler: Info index.handler
--------------------------	--

Step 7:

let us save the changes and test the lambda function with S3upload.



Step 8:

Now, save the Lambda function. Open S3 from Amazon services and open the bucket we created earlier namely lambdawiths3.

Upload the image in it as shown below –

Click **Add files** to add files. You can also drag and drop the files. Now, click **Upload** button.

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 folders**.

Files and folders (1 Total, 44.0 KB)					Remove	Add files	Add folder
All files and folders in this table will be uploaded.							
<input type="text"/> Find by name < 1 >							
<input type="checkbox"/>	Name	▲	Folder	▼	Type	▼	Size
<input type="checkbox"/>	apsit_logo.jpg	-			image/jpeg		44.0 KB

Thus, we have uploaded one image in our S3 bucket.

Step 9

To see the trigger details, go to AWS service and select CloudWatch. Open the logs for the Lambda AWS Lambda function gets triggered when file is uploaded in S3 bucket and the details are logged in Cloudwatch as shown below –



PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY
Department of Information Technology
(NBA Accredited)



CloudWatch > Log groups > /aws/lambda/lambdawiths3bucket > 2021/08/03/[LATEST]0f36a60d46ca40078172fc11de9d735f

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

[View as text](#)

Filter events Clear 1m

▶	Timestamp	Message
		No older events at this moment. Retry
▶	2021-08-03T12:01:00.069+05:30	START RequestId: ae43508a-8eb7-4b08-8fa1-841814d597c1 Version: \$LATEST
▶	2021-08-03T12:01:00.098+05:30	2021-08-03T06:31:00.097Z ae43508a-8eb7-4b08-8fa1-841814d597c1 INFO Incoming Event: { Records: [{ eventVersion: '2.1', eventSource: 'aws:s3', awsRegion: 'ap-south-1',
▼	2021-08-03T12:01:00.098+05:30	2021-08-03T06:31:00.098Z ae43508a-8eb7-4b08-8fa1-841814d597c1 INFO An Image has been added - lambdawiths3 -> apsit_logo.jpg 2021-08-03T06:31:00.098Z ae43508a-8eb7-4b08-8fa1-841814d597c1 INFO An Image has been added - lambdawiths3 -> apsit_logo.jpg
▶	2021-08-03T12:01:00.119+05:30	END RequestId: ae43508a-8eb7-4b08-8fa1-841814d597c1
▶	2021-08-03T12:01:00.119+05:30	REPORT RequestId: ae43508a-8eb7-4b08-8fa1-841814d597c1 Duration: 49.40 ms Billed Duration: 50 ms Memory Size: 128 MB Max Memory Used: 65 MB Init Duration: 155.37 ms
		No newer events at this moment. Auto retry paused. Resume

An image has been Added -> **apsit_logo.jpg** you can see in cloudwatch logs.

Conclusion: Write your own findings.

```
export const handler = async (event, context) => {  
    console.log("Incoming Event:", event);  
  
    if (!event.Records || event.Records.length === 0) {  
        const errorMessage = "No records found in the event.";  
        console.log(errorMessage);  
        return errorMessage;  
    }  
  
    const bucket = event.Records[0].s3.bucket.name;  
    const filename = decodeURIComponent(event.Records[0].s3.object.key.replace(/\+/g, " "));  
    const message = An Image has been added - ${bucket} -> ${filename};  
    console.log(message);  
  
    return message;  
};
```



Semester: V

Academic Year: 2023-24

Year :TE IT

Subject: Advanced Devops Lab (ADL)

Name of Instructor: Prof. Manjusha K.

EXPERIMENT NO. 13

Aim: To demonstrate working of GAE launcher to launch the web applications.

Theory:

Prerequisites:

Installed Google Cloud SDK: Ensure that you have the Google Cloud SDK installed on your machine. If not, follow the installation guide provided by Google Cloud.

Lab Steps:

1. Create a Google Cloud Project:

- Open the Google Cloud Console.
- Create a new project or select an existing one.

2. Enable App Engine API:

- Navigate to the "APIs & Services" > "Dashboard" section.
- Click on the "+ ENABLE APIS AND SERVICES" button.
- Search for "App Engine Admin API" and enable it.

3. Install GAE Launcher:

- Download and install the Google App Engine Launcher for your operating system.

4. Configure GAE Launcher:

- Open GAE Launcher and sign in with your Google Cloud account.
- Click on "Create a new application" and enter a unique application ID.

5. Develop a Simple Web Application:

- Create a new directory for your web application.
- Inside the directory, create an 'app.yaml' file with the following content:
yaml runtime: python39 handlers: - url: /* script: main.app
- Create a 'main.py' file with a simple Python web application.

6. Run the Application Locally:

- In GAE Launcher, click on the "Run" button next to your application.
- Open a web browser and navigate to <http://localhost:8080> to view your locally running application.



7. Deploy the Application to GAE:

- Click on the "Deploy" button in GAE Launcher.
- Follow the prompts to deploy your application to the Google Cloud Platform.

8. Access the Deployed Application:

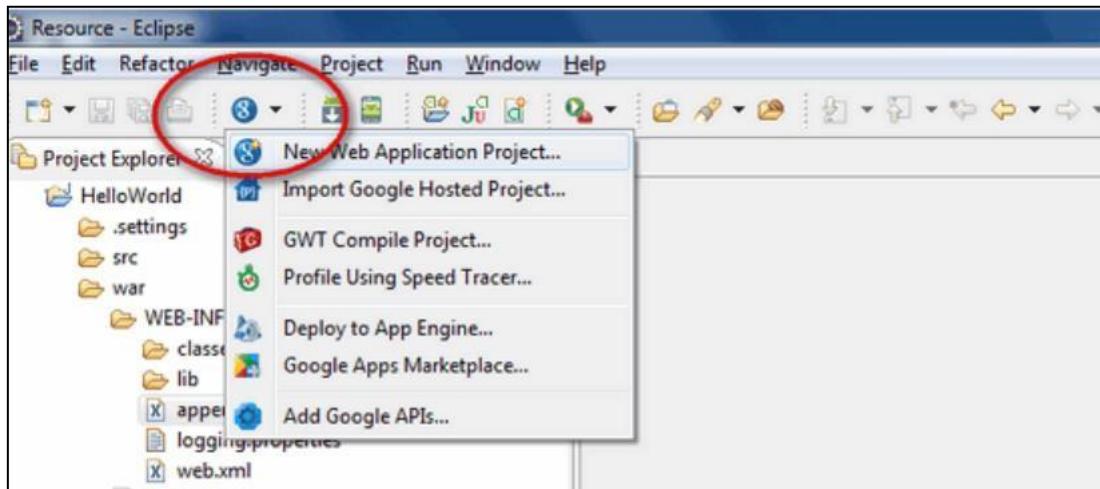
- Once deployed, open a web browser and navigate to [https://\[YOUR_PROJECT_ID\].appspot.com](https://[YOUR_PROJECT_ID].appspot.com).

How to install Plugins -

1. Install Google Plugin for Eclipse

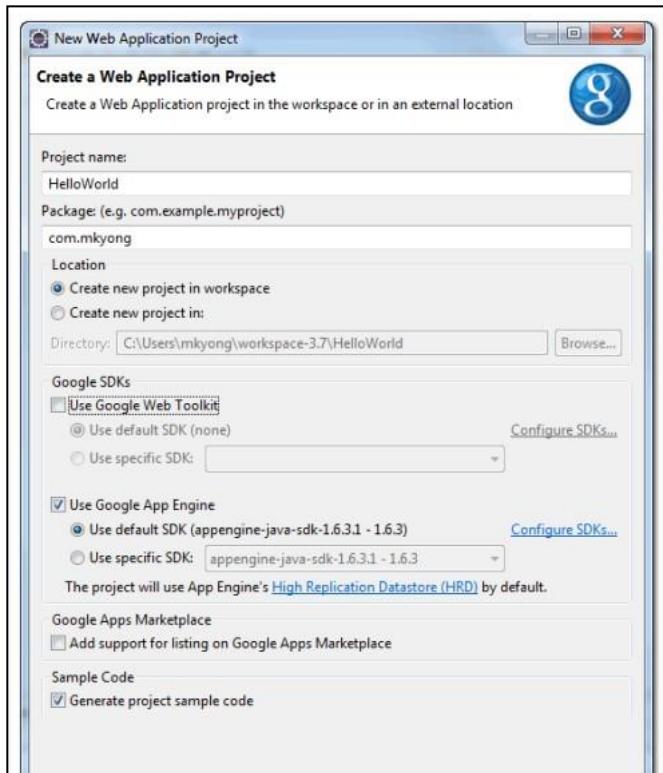
2. Create New Web Application Project

In Eclipse toolbar, click on the Google icon, and select “New Web Application Project...”





PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY
Department of Information Technology
(NBA Accredited)



Click finished, Google Plugin for Eclipse will generate a sample project automatically.

3. Hello World

Review the generated project directory

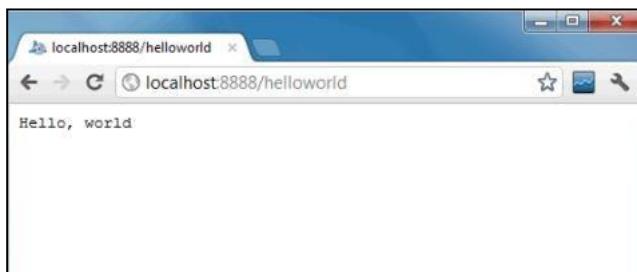




PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY
Department of Information Technology
(NBA Accredited)



4. Run it local Right click on the project and run as "Web Application". Eclipse console : //... INFO: The server is running at http://localhost:8888/ 30 Mac 2012 11:13:01 PM com.google.appengine.tools.development.DevAppServerImpl start INFO: The admin console is running at http://localhost:8888/_ah/admin Copy Access URL http://localhost:8888/, see output

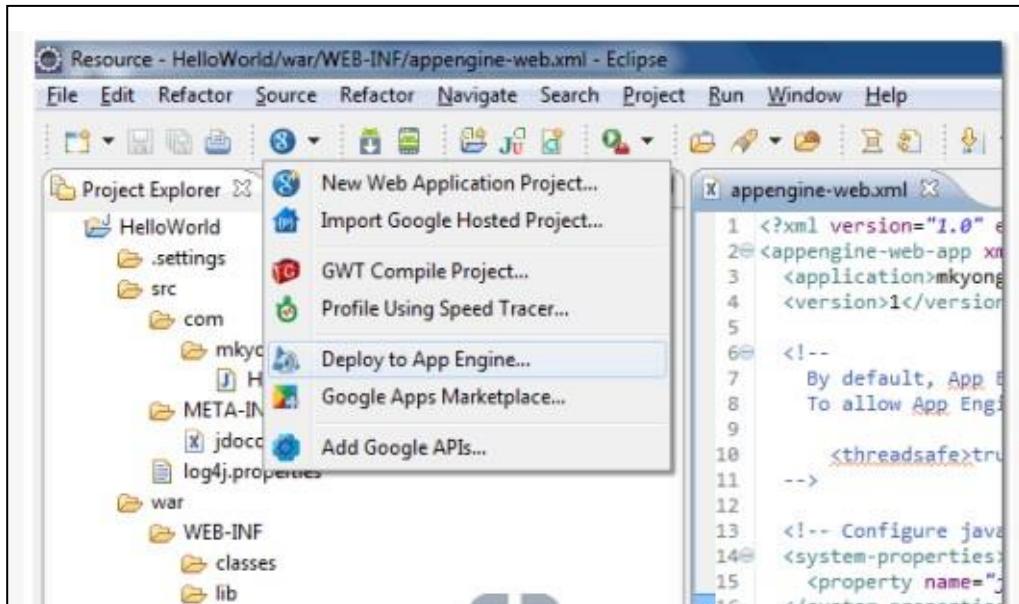


5. Deploy to Google App Engine

Register an account on <https://appengine.google.com/>, and create an application ID for your web application.

In this demonstration, I created an application ID, named "mkyong123", and put it in appenginemeweb.xml.

File : appengine-web.xml





Sign in with your Google account and click on the Deploy button. If everything is fine, the hello world web application will be deployed to this URL – <http://mkyong123.appspot.com/>



Conclusion: Thus we have learned how to use the Google App Engine Launcher to deploy a web application. This practical provides a foundation for deploying more complex applications on the Google App Engine platform.