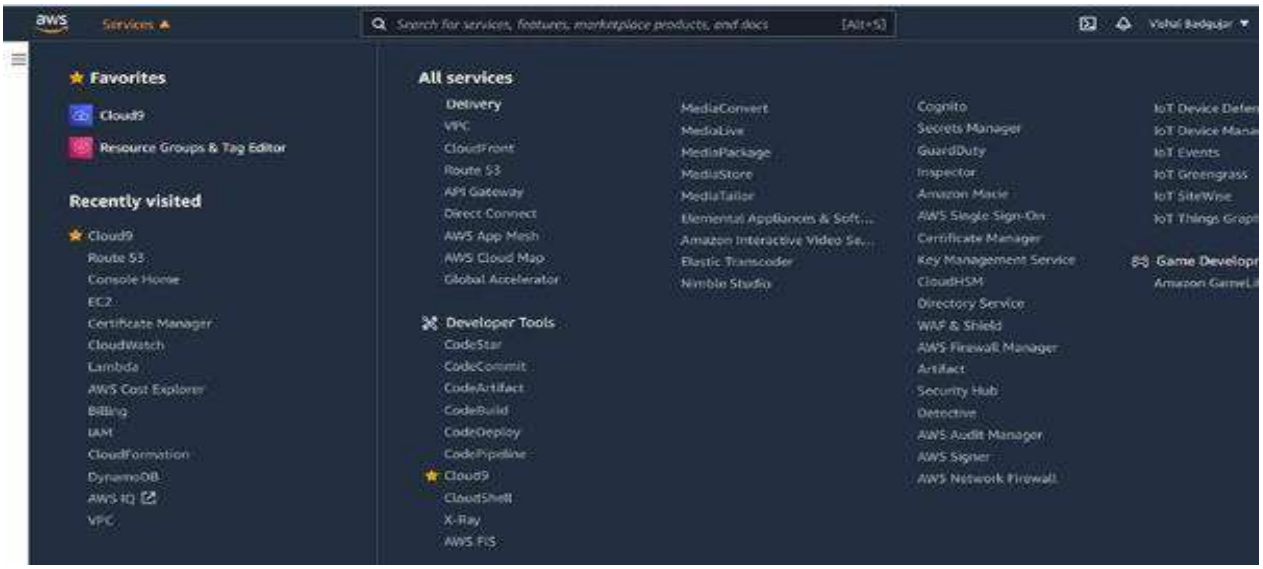
EXP1

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

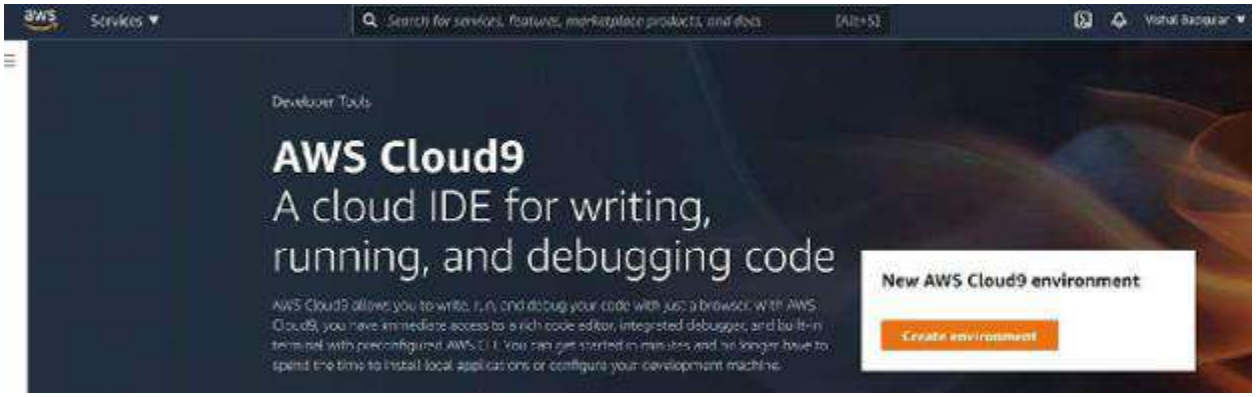
<u>Steps:</u>

1. Login with your AWS account.

2. Navigate to Cloud 9 service from Developer tools section as below:



3. Click on Create Environment :

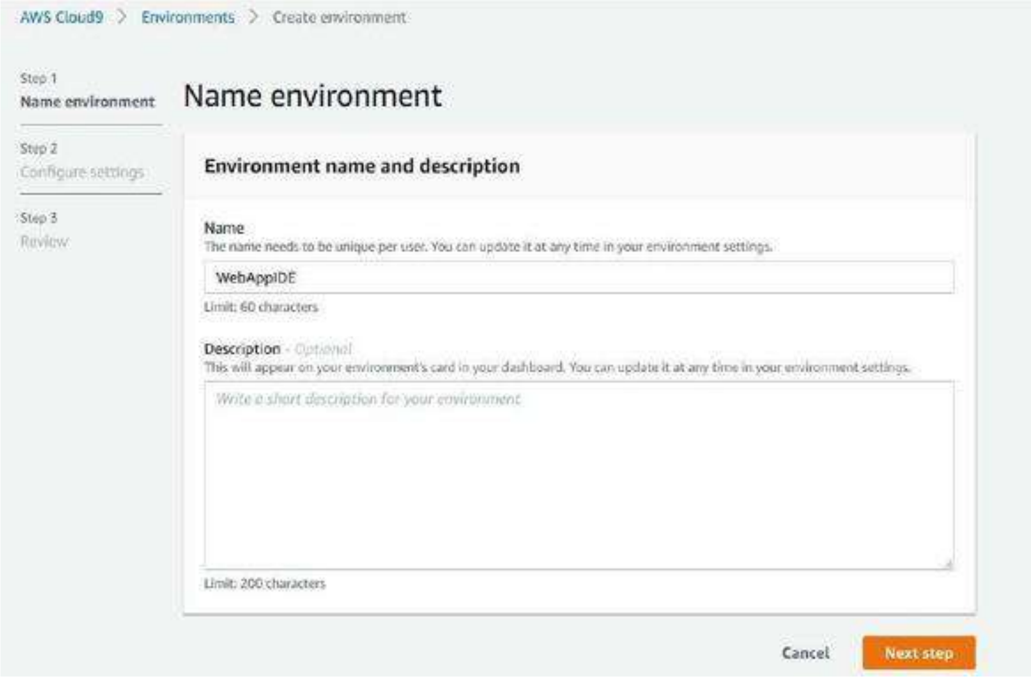

aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
Vishal Badgujar ▼
Developer Tools
New AWS Cloud9 environment
Create environment

AWS Cloud9

A cloud IDE for writing, running, and debugging code

AWS Cloud9 allows you to write, run, and debug your code with just a browser. With AWS Cloud9, you have immediate access to a rich code editor, integrated debugger, and built-in terminal with preconfigured AWS CLI. You can get started in minutes and no longer have to spend the time to install local applications or configure your development machine.

4. 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.

WebAppIDE

Limit: 60 characters

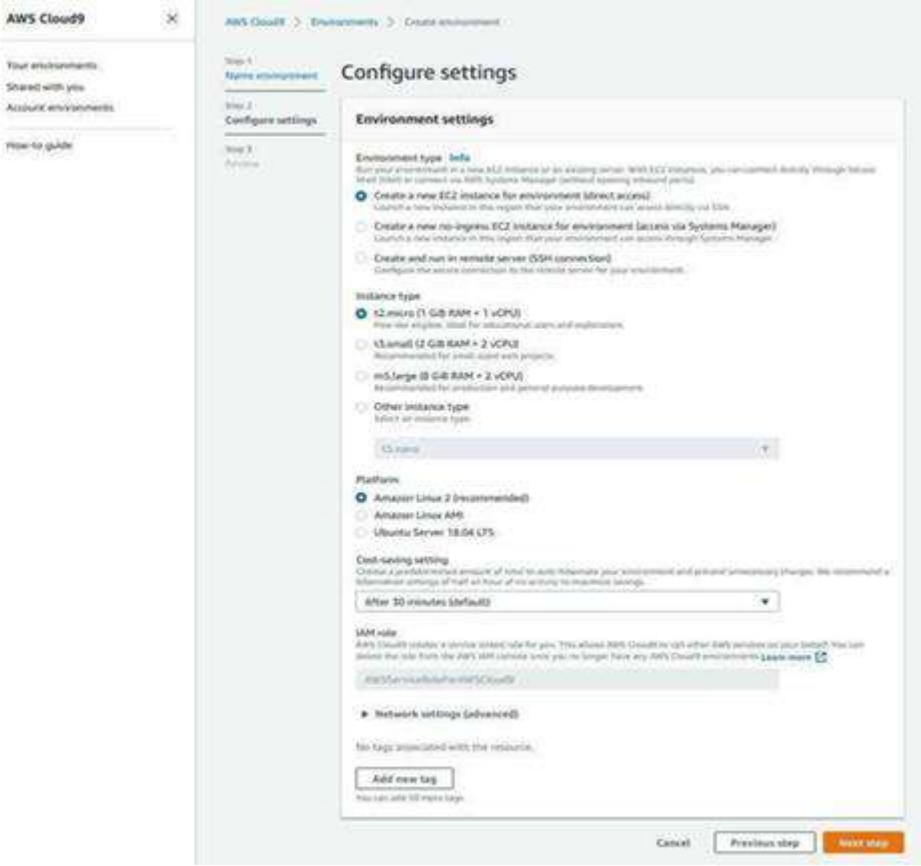
Description - Optional
This will appear on your environment's card in your dashboard. You can update it at any time in your environment settings.

Write a short description for your environment

Limit: 200 characters

Cancel | Next step

5. 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
[illegible]

● Create a new EC2 instance for environment (direct access)
[illegible]

○ Create a new no-ingress EC2 instance for environment (access via Systems Manager)
[illegible]

○ Create and run in remote server (SSH connection)
[illegible]

Instance type

● t2.micro (1 GiB RAM + 1 vCPU)
[illegible]

○ t3.small (2 GiB RAM + 2 vCPU)
[illegible]

○ m5.large (8 GiB RAM + 2 vCPU)
[illegible]

○ Other instance type
[illegible]

t3.nano

Platform

● Amazon Linux 2 (recommended)
○ Amazon Linux AMI
○ Ubuntu Server 18.04 LTS

Cost-saving setting
[illegible]

After 30 minutes (default)

IAM role
[illegible] Learn more

[illegible]

▶ Network settings (advanced)

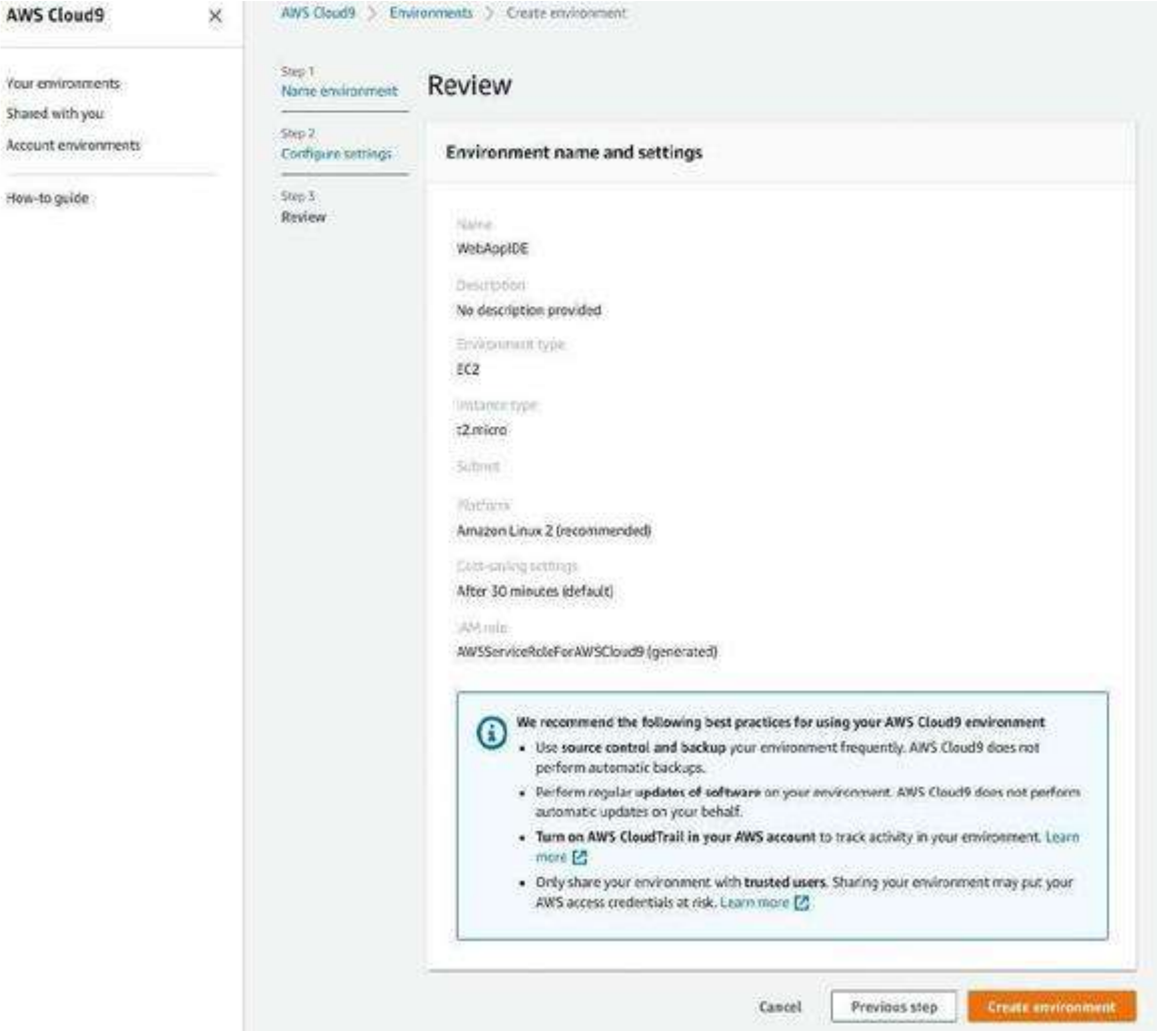
No tags associated with the resource.

Add new tag

[illegible]

Cancel | Previous step | Next step

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



Review

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)

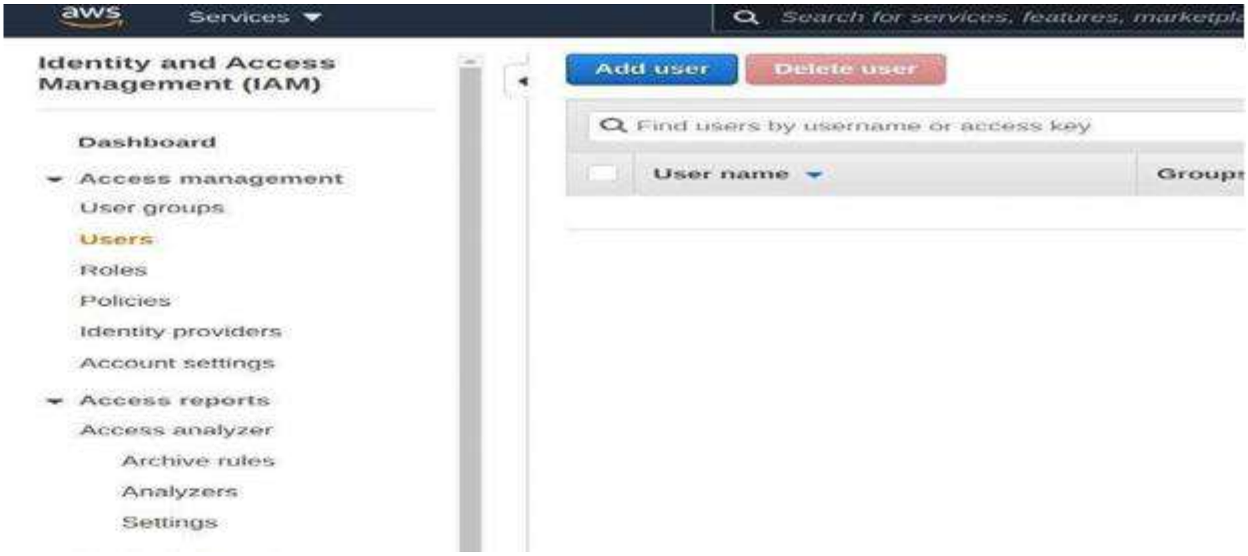
IAM role
AWSServiceRoleForAWSCloud9 (generated)

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

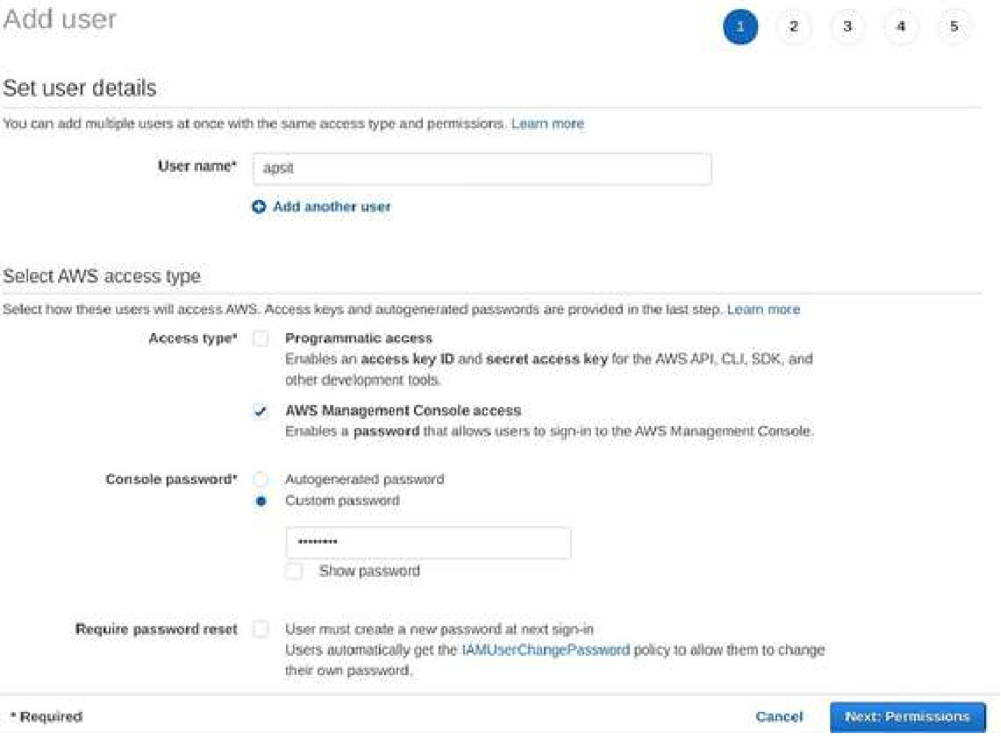
- Use **source control and backup** your environment frequently. AWS Cloud9 does not perform automatic backups.
- Perform regular **updates of software** on your environment. AWS Cloud9 does not perform automatic updates on your behalf.
- **Turn on AWS CloudTrail in your AWS account** to track activity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

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

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



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



Add user

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

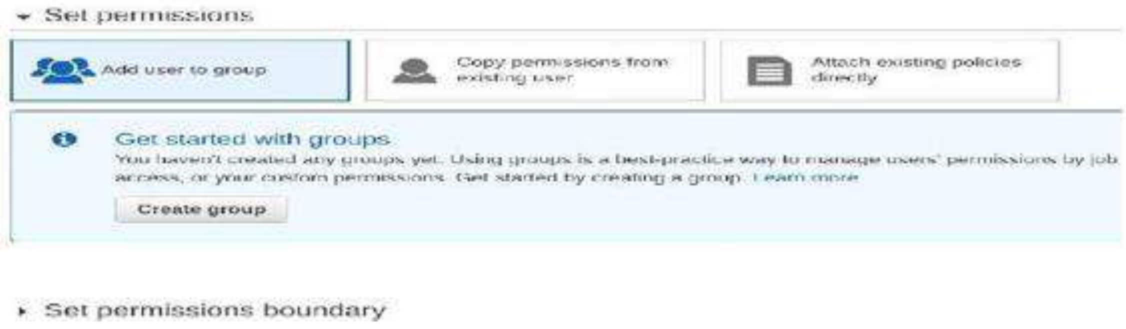
••••••••

☐ 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

9. Click on Create group



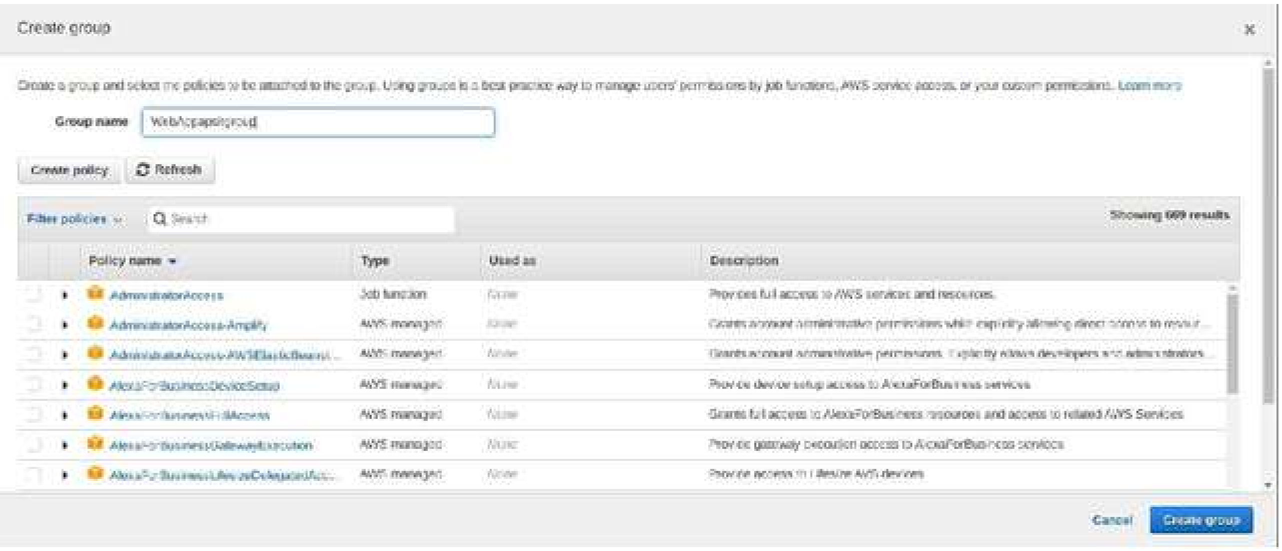
Set permissions
Add user to group
Copy permissions from existing user
Attach existing policies directly

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

10. Provide group name and click on create group.



Create group

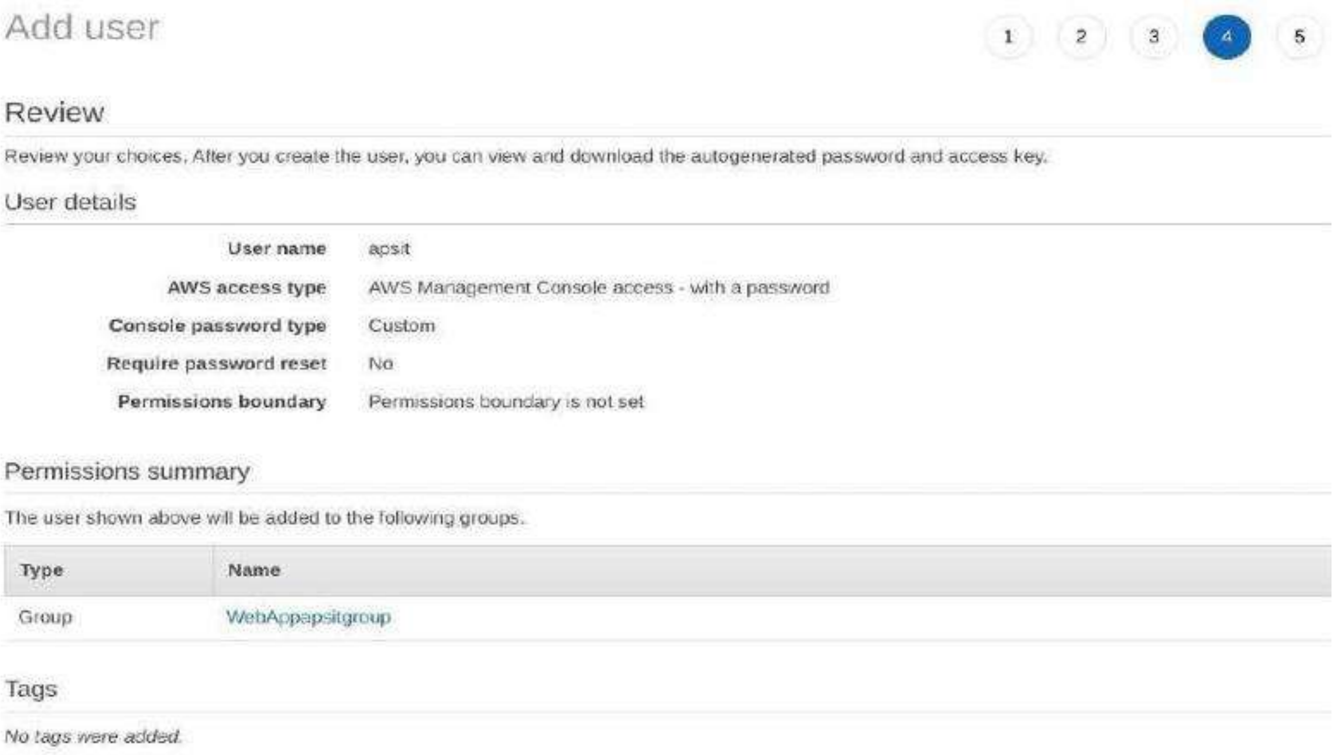
Create a group and select the policies to be attached to the group. Using groups is a best practice way to manage users' permissions by job functions, AWS service access, or your custom permissions. Learn more

Group name WebAppapsitgroup

Showing 669 results

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

11.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

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

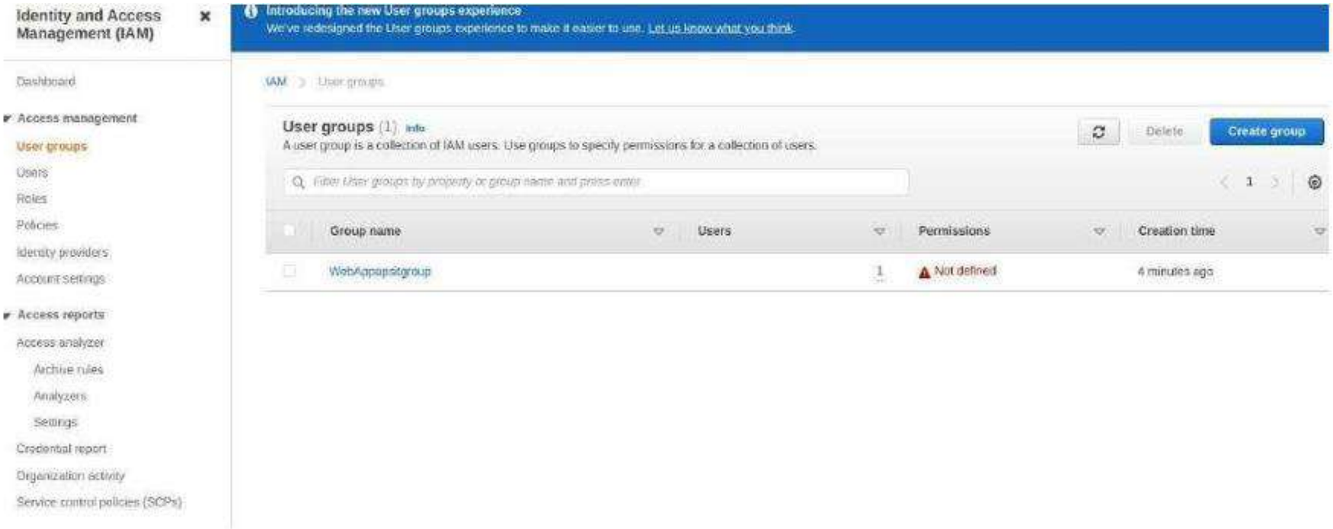
The user shown above will be added to the following groups.

Type	Name
Group	WebAppapsitgroup

Tags

No tags were added.

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



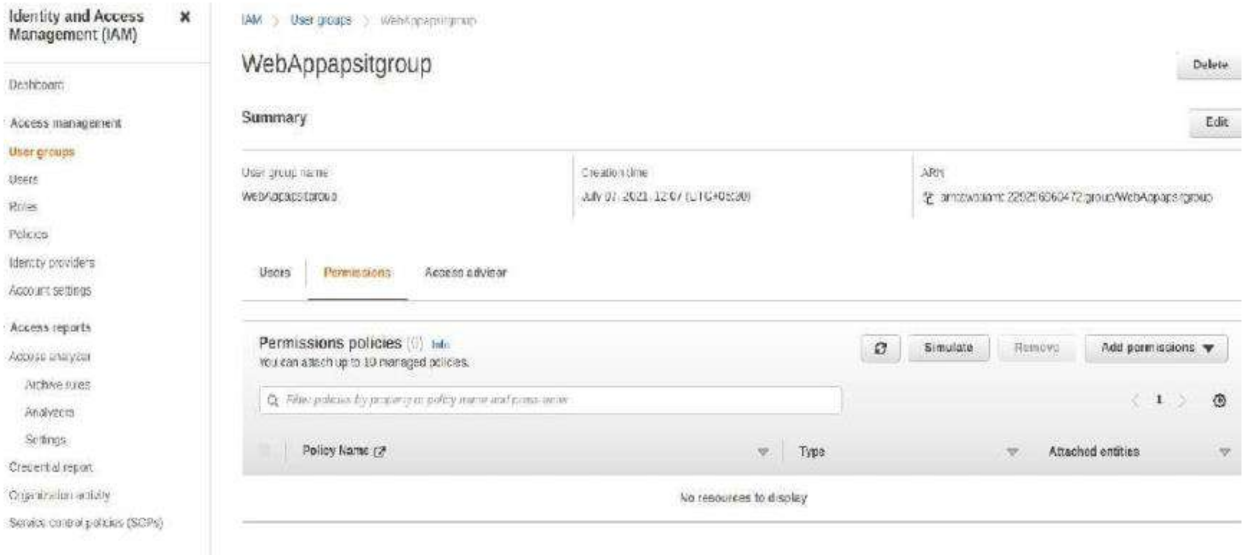
Introducing the new User groups experience
We've redesigned the User groups experience to make it easier to use. Let us know what you think.

User groups (1) Info
A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.

Filter User groups by property or group name and press enter

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

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



WebAppapsitgroup

Summary

User group name: WebAppapsitgroup

Creation time: July 07, 2021, 12:07 (UTC+05:30)

ARN: arn:aws:iam::229256060472:group/WebAppapsitgroup

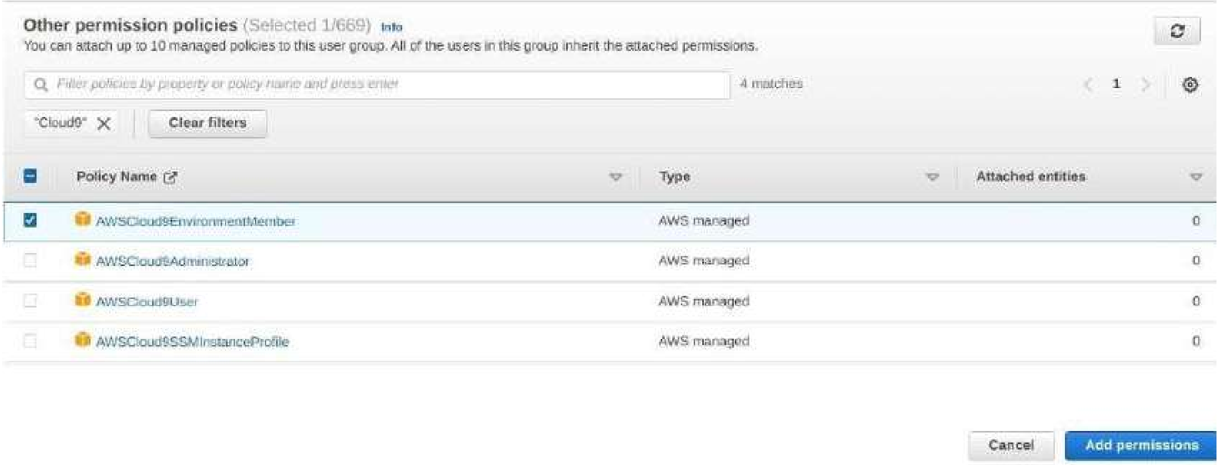
Permissions policies (0) Info
You can attach up to 10 managed policies.

Filter policies by property or policy name and press enter

Policy Name	Type	Attached entities

No resources to display

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



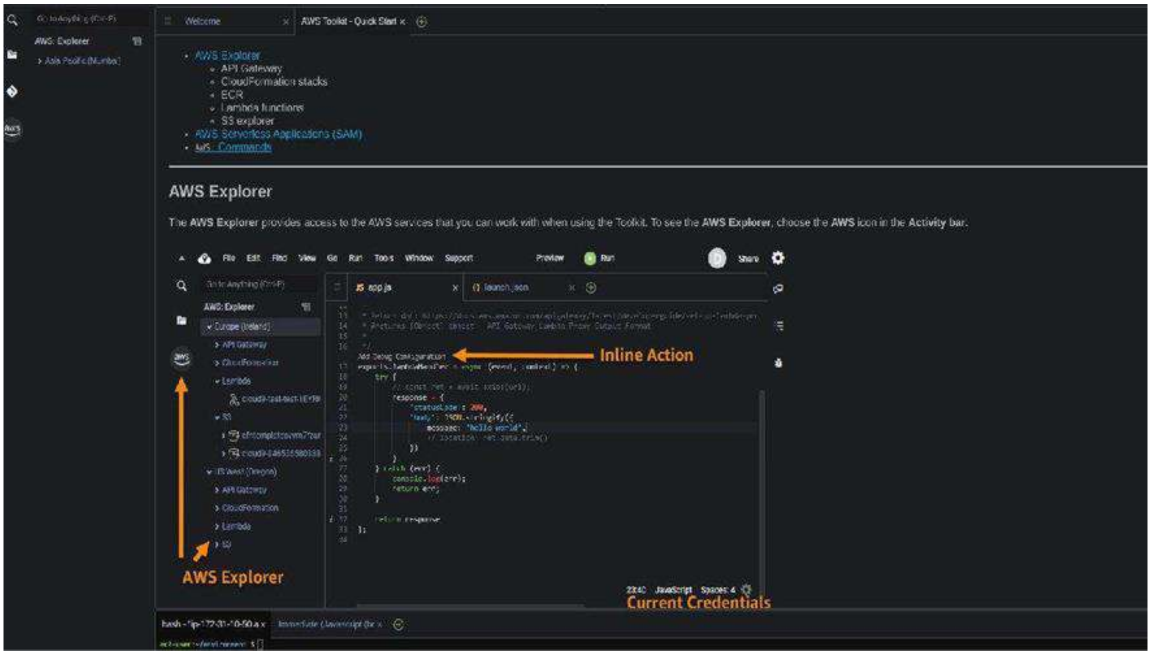
Other permission policies (Selected 1/669)
You can attach up to 10 managed policies to this user group. All of the users in this group inherit the attached permissions.

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

"Cloud9" ✕

☐	Policy Name	Type	Attached entities
☑	AWSCloud9EnvironmentMember	AWS managed	0
☐	AWSCloud9Administrator	AWS managed	0
☐	AWSCloud9User	AWS managed	0
☐	AWSCloud9SSMInstanceProfile	AWS managed	0

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



- AWS Explorer
  - API Gateway
  - CloudFormation stacks
  - ECR
  - Lambda functions
  - S3 explorer
- AWS Serverless Applications (SAM)
- AWS: Commands

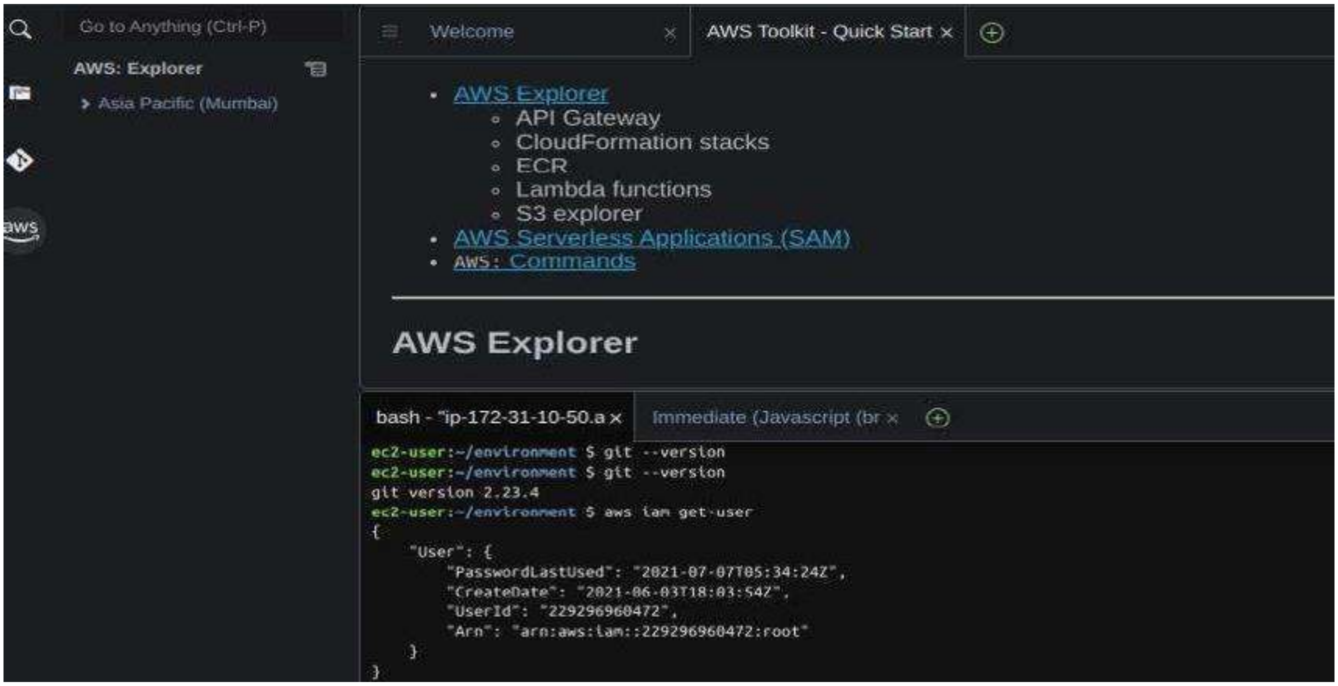
AWS Explorer

The **AWS Explorer** provides access to the AWS services that you can work with when using the Toolkit. To see the **AWS Explorer**, choose the **AWS** icon in the **Activity bar**.

16. 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...

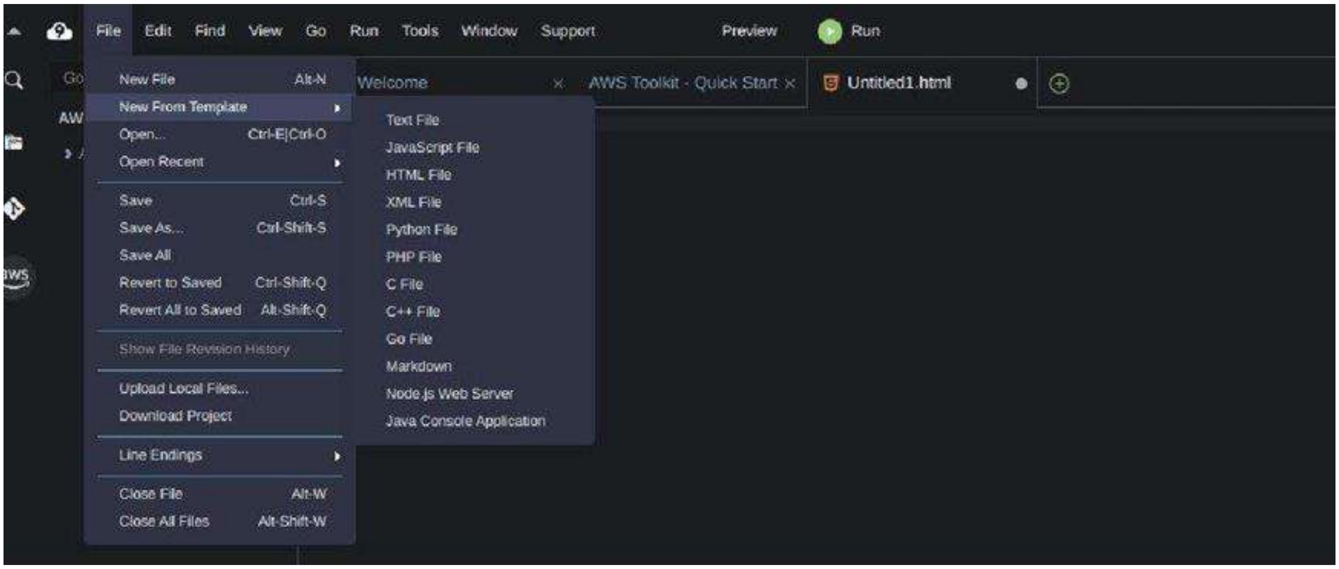
$git –version

$aws iam get-user

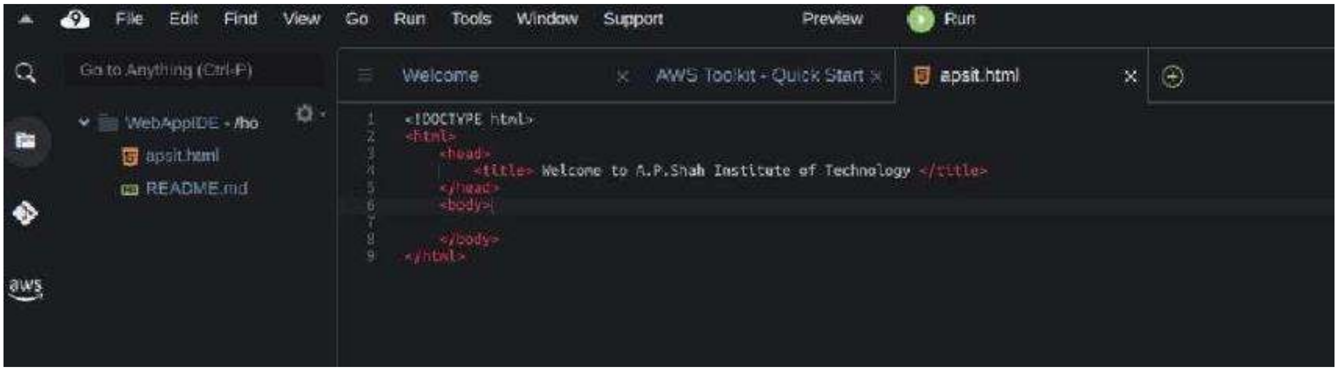


```
ec2-user:~/environment $ git --version
ec2-user:~/environment $ git --version
git version 2.23.4
ec2-user:~/environment $ aws iam get-user
{
    "User": {
        "PasswordLastUsed": "2021-07-07T05:34:24Z",
        "CreateDate": "2021-06-03T18:03:54Z",
        "UserId": "229296960472",
        "Arn": "arn:aws:iam::229296960472:root"
    }
}
```

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



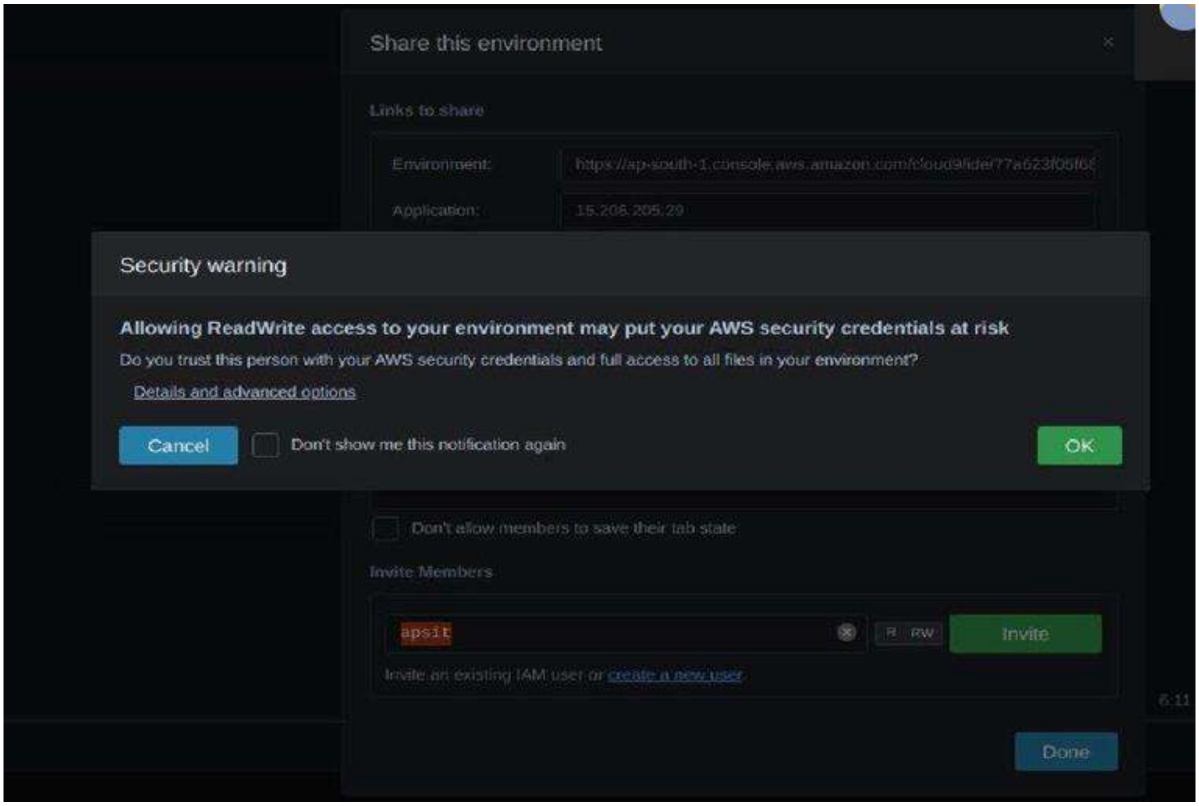
18. Edit html file and save it



```
<!DOCTYPE html>
<html>
    <head>
        <title> Welcome to A.P.Shah Institute of Technology </title>
    </head>
    <body>

    </body>
</html>
```

19. 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.


Share this environment

Links to share

Environment: https://ap-south-1.console.aws.amazon.com/cloud9/ide/77a623f05f6

Application: 15.206.205.29

Security warning

Allowing ReadWrite access to your environment may put your AWS security credentials at risk

Do you trust this person with your AWS security credentials and full access to all files in your environment?

Details and advanced options

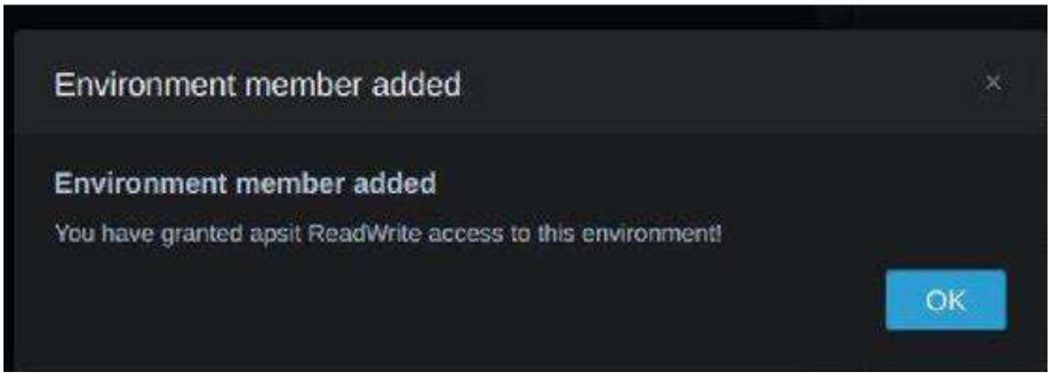
□ Don't show me this notification again

□ Don't allow members to save their tab state

Invite Members

apsit

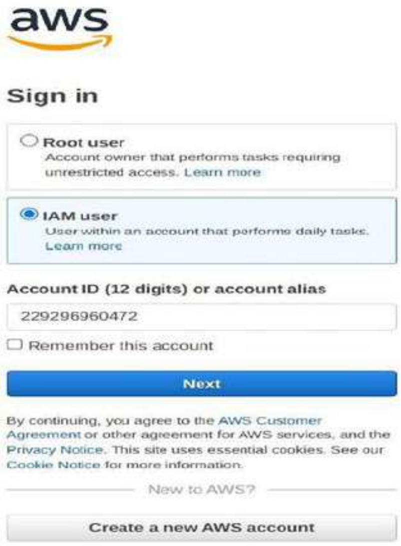
Invite an existing IAM user or create a new user


Environment member added

Environment member added

You have granted apsit ReadWrite access to this environment!

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



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

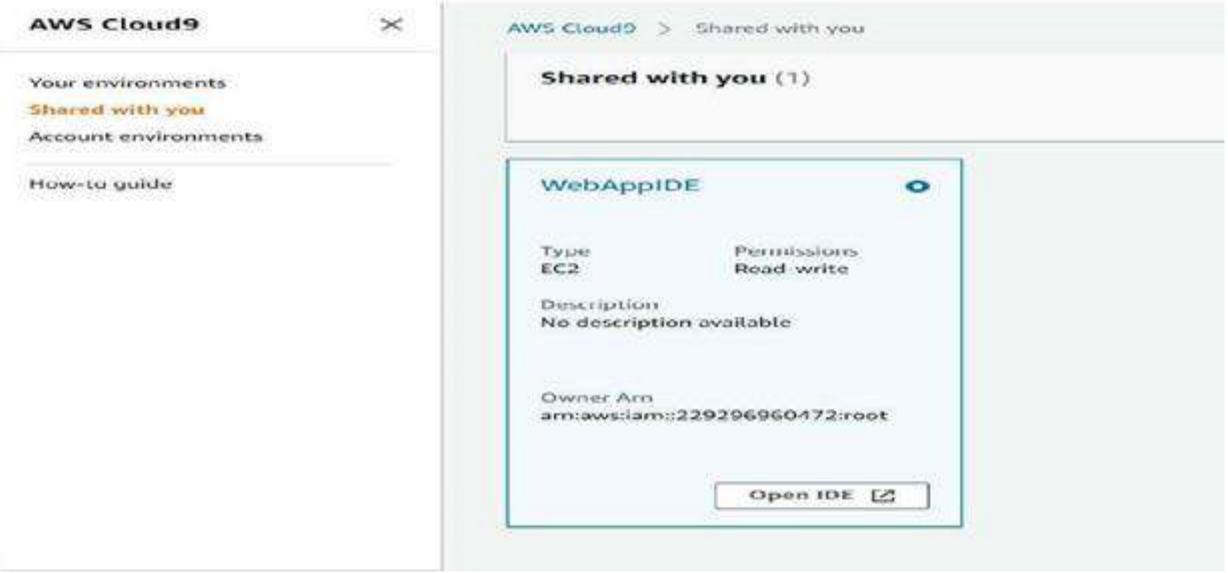
Next

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

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



AWS Cloud9

Your environments
Shared with you
Account environments

How-to guide

AWS Cloud9 > Shared with you

Shared with you (1)

WebAppIDE

Type: EC2

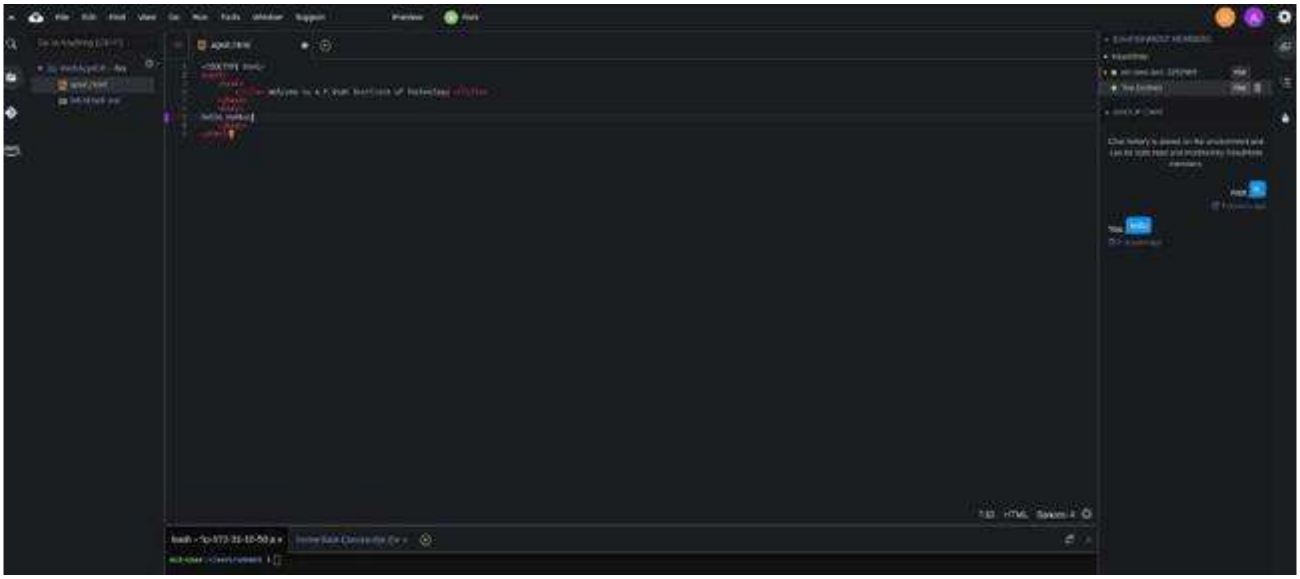
Permissions: Read-write

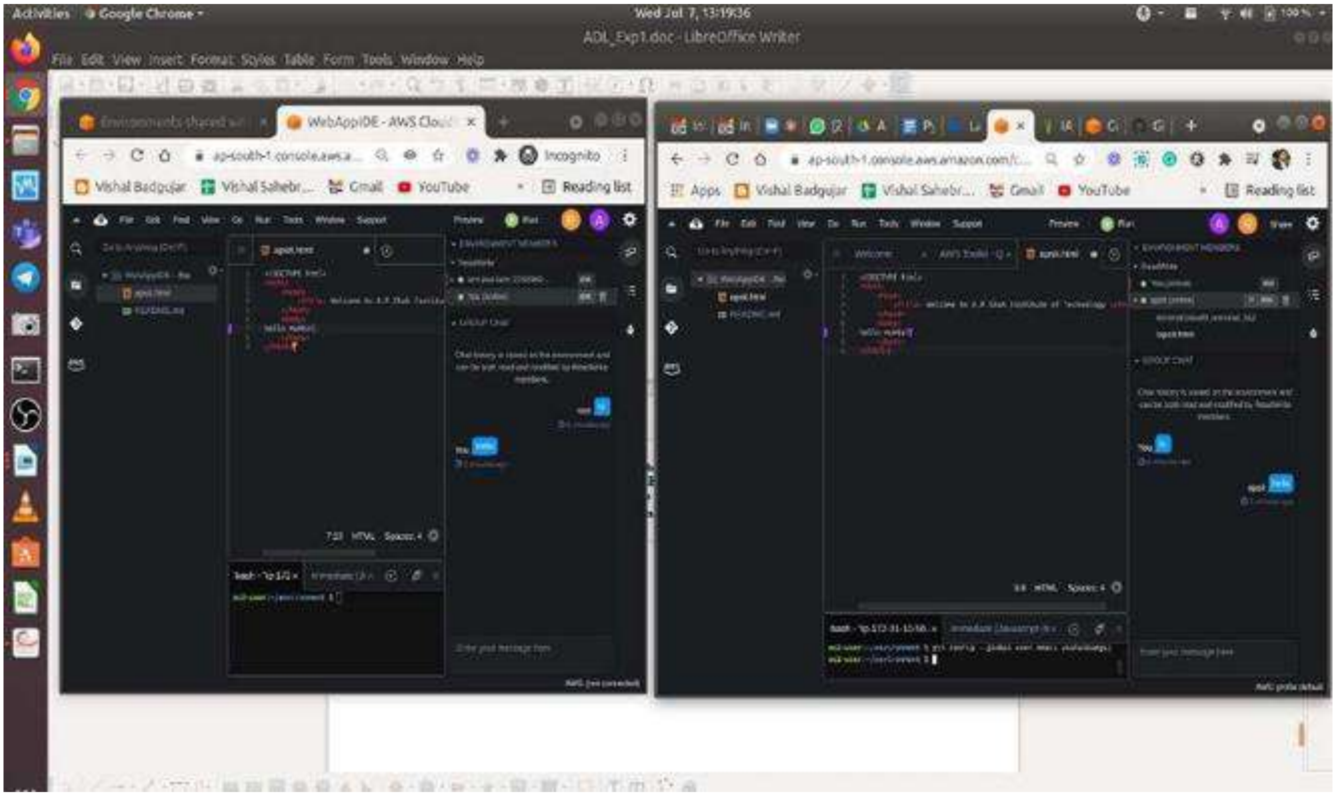
Description: No description available

Owner Arn: arn:aws:iam::229296960472:root

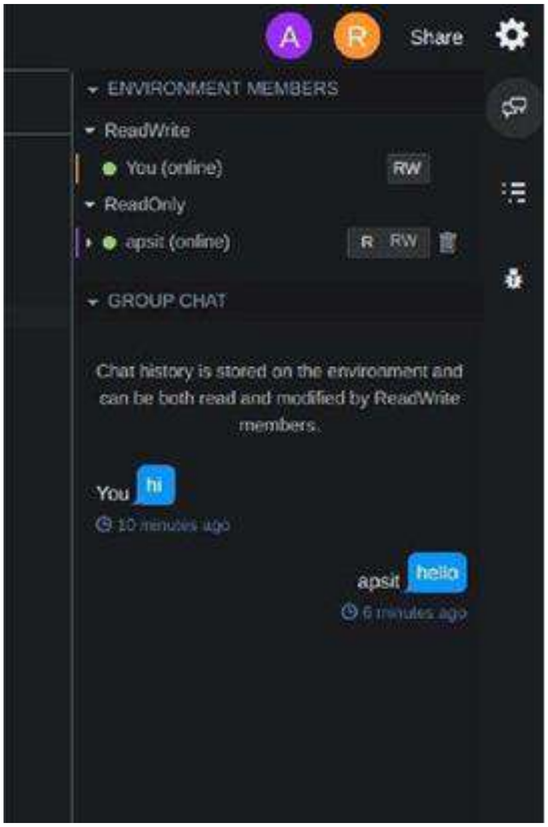
Open IDE

22. **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:**





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



ENVIRONMENT MEMBERS

ReadWrite

● You (online) RW

ReadOnly

● apsit (online) R RW

GROUP CHAT

Chat history is stored on the environment and can be both read and modified by ReadWrite members.

You hi

10 minutes ago

apsit hello

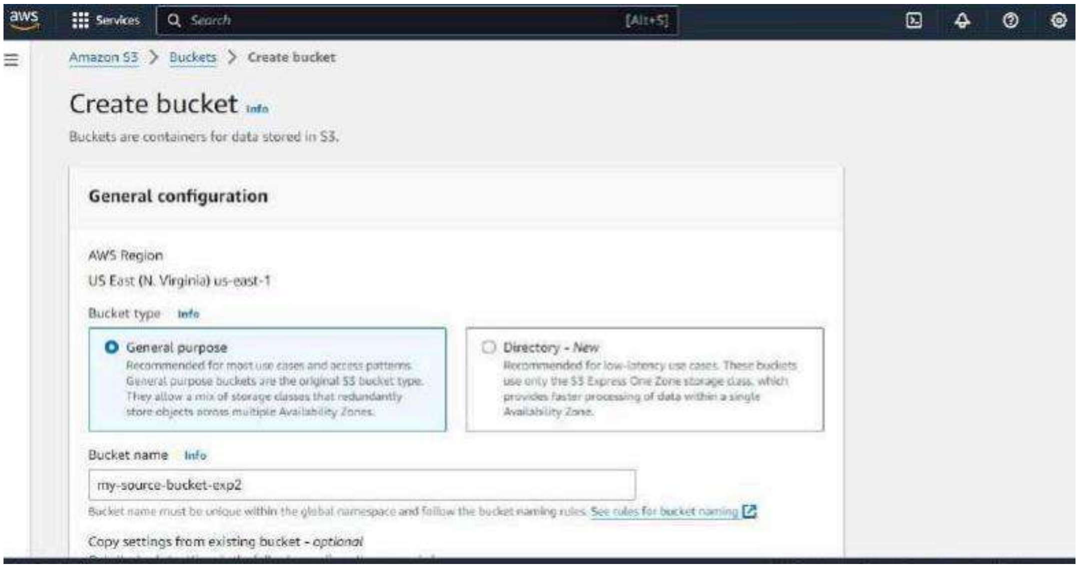
6 minutes ago

EXP2

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.

go to amazon s3 > buckets > create bucket

create a bucket



Amazon S3 > Buckets > Create bucket

Create bucket Info

Buckets are containers for data stored in S3.

General configuration

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

Bucket type Info

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

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

Bucket name Info
my-source-bucket-exp2
Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming

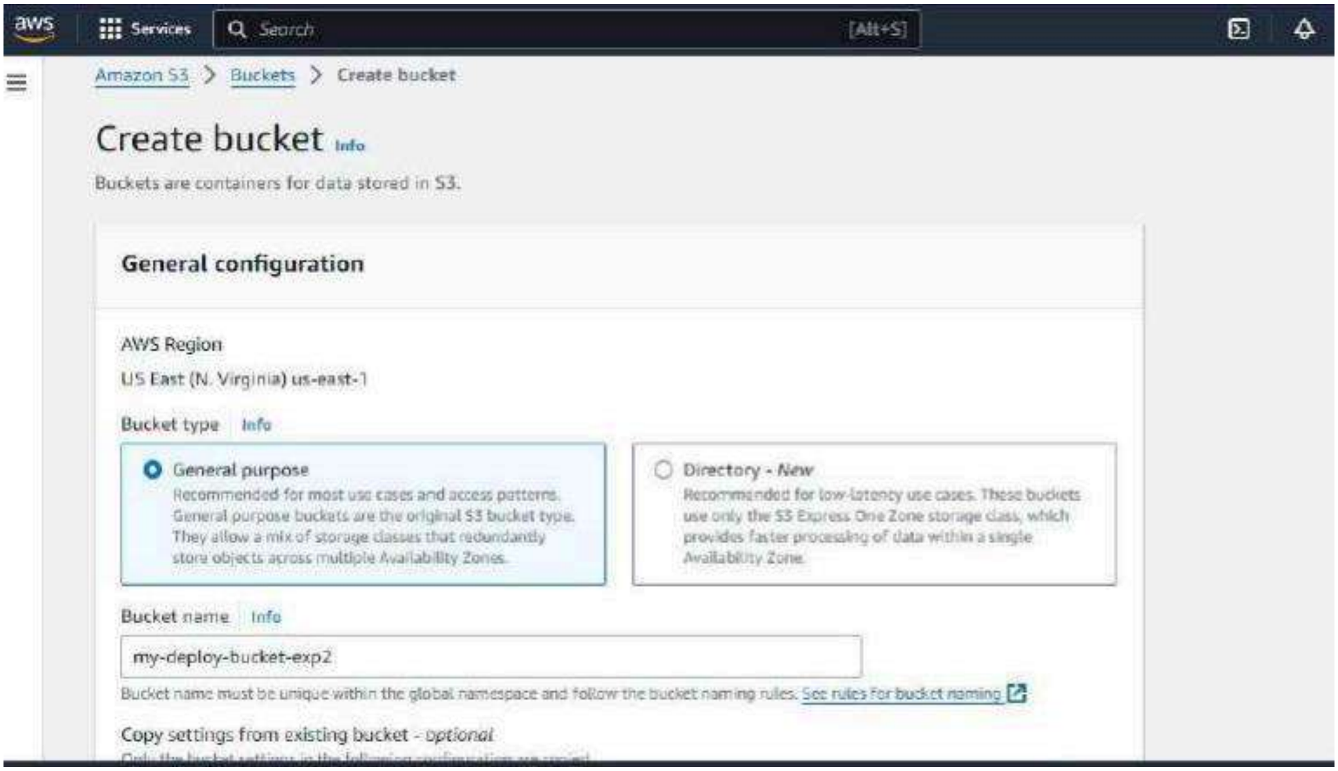
Copy settings from existing bucket - *optional*

bucket name: my-source-bucket

then click create

create another bucket

bucket name: my-deploy-bucket



Amazon S3 > Buckets > Create bucket

Create bucket Info

Buckets are containers for data stored in S3.

General configuration

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

Bucket type Info

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

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

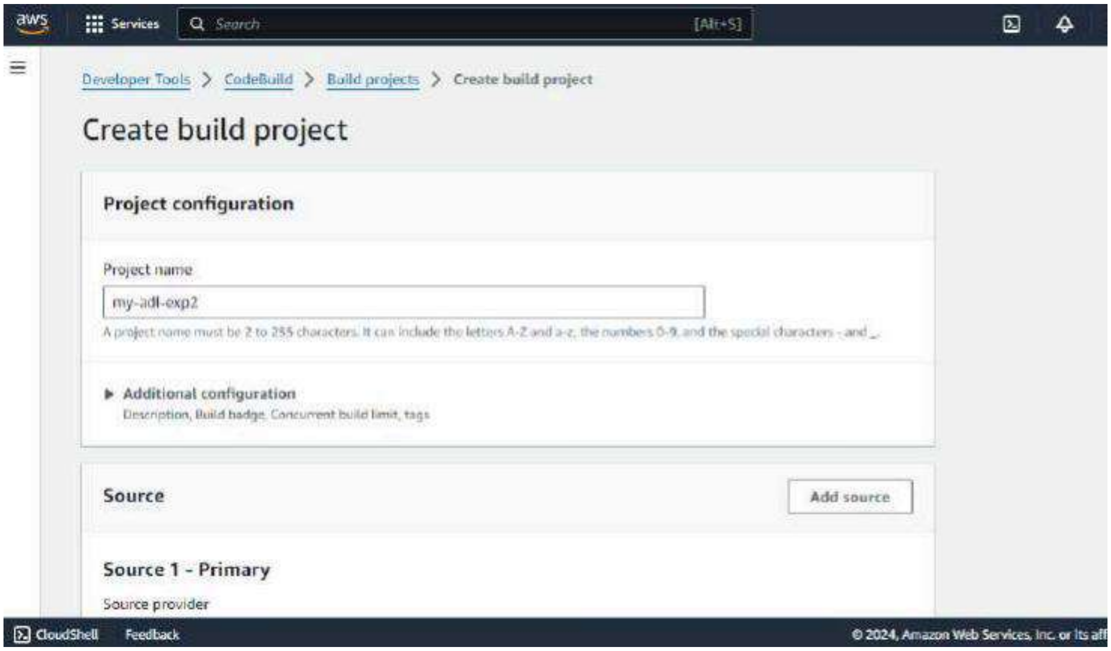
Bucket name Info
my-deploy-bucket-exp2
Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming

Copy settings from existing bucket - *optional*

open new tab ,

go to codebuild > build project >create build project

project name: my-exp2



Developer Tools > CodeBuild > Build projects > Create build project

Create build project

Project configuration

Project name: my-adl-exp2

A project name must be 2 to 255 characters. It can include the letters A-Z and a-z, the numbers 0-9, and the special characters - and _.

▶ Additional configuration
Description, Build badge, Concurrent build limit, tags

Source

Source 1 - Primary

Source provider

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

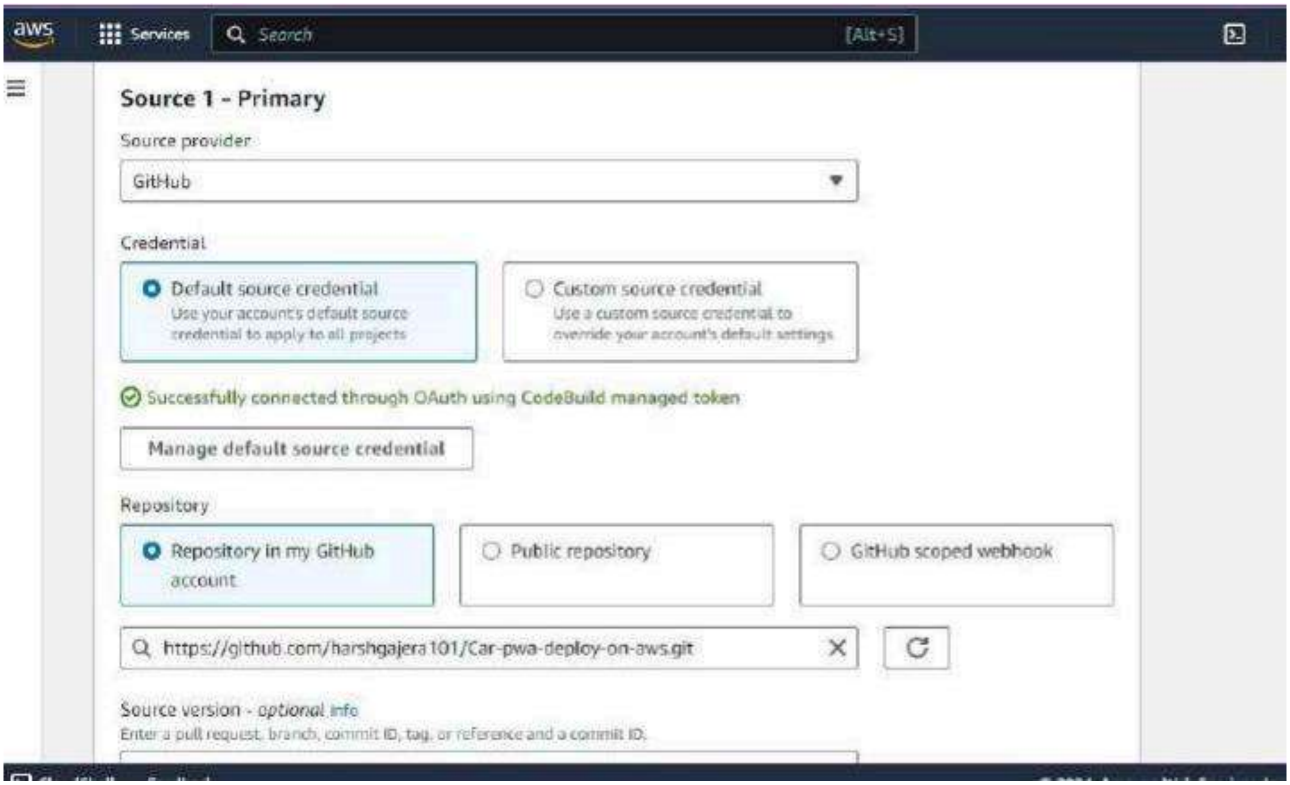
source1 - primary

source provider: github

credential : default source credential

repository: repository in my github

https://github.com/harshgajera101/Car-pwa-deploy-on-aws.git



Source 1 - Primary

Source provider: GitHub

Credential

● Default source credential
Use your account's default source credential to apply to all projects

○ Custom source credential
Use a custom source credential to override your account's default settings

✓ Successfully connected through OAuth using CodeBuild managed token

Repository

● Repository in my GitHub account

○ Public repository

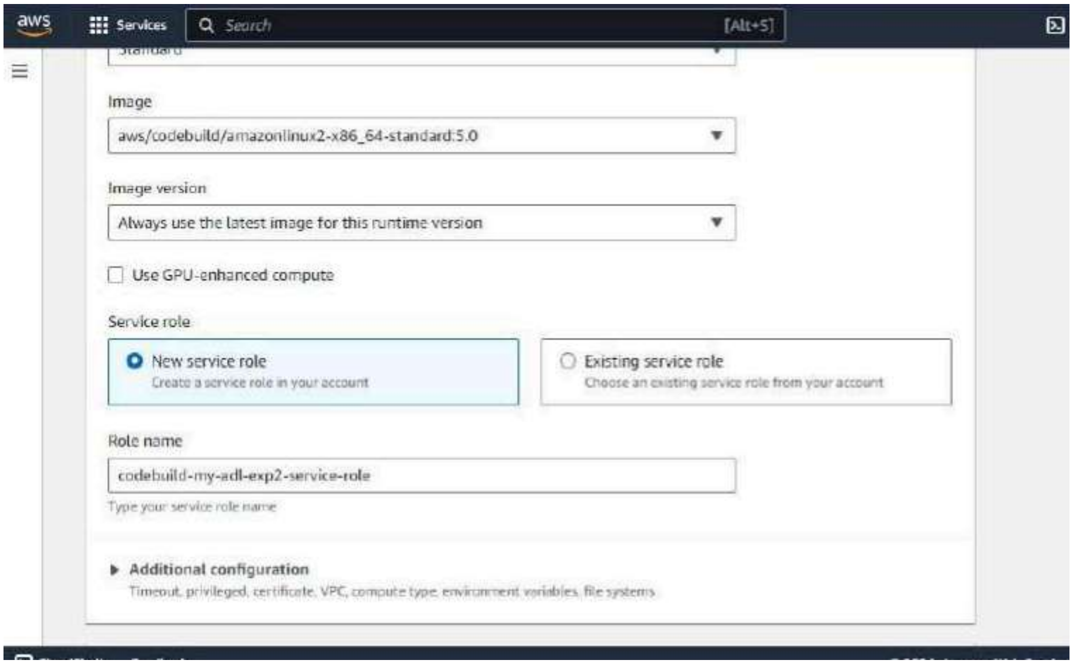
○ GitHub scoped webhook

https://github.com/harshgajera101/Car-pwa-deploy-on-aws.git

Source version - *optional* Info
Enter a pull request, branch, commit ID, tag, or reference and a commit ID.

service role: new service role

role name: codebuild-my-exp2-service-role



Standard

Image

aws/codebuild/amazonlinux2-x86_64-standard:5.0 ▼

Image version

Always use the latest image for this runtime version ▼

☐ Use GPU-enhanced compute

Service role

● New service role
Create a service role in your account

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

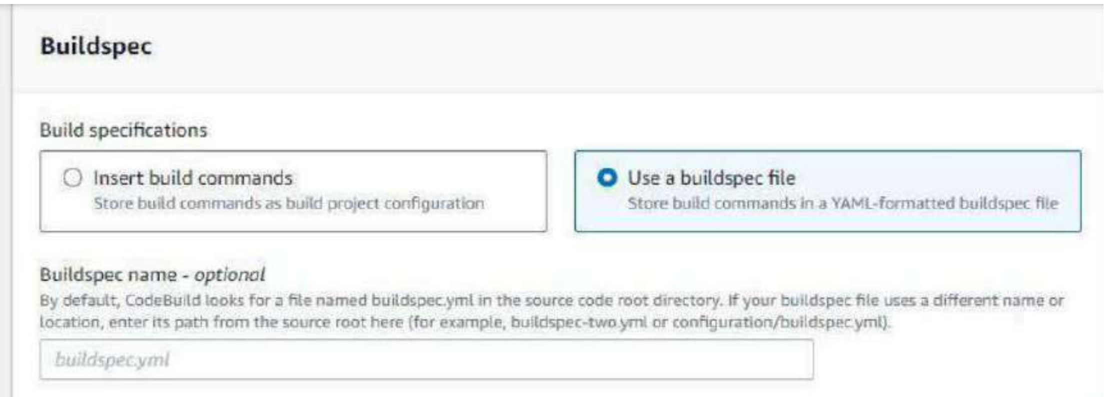
Role name

codebuild-my-adl-exp2-service-role

Type your service role name

▶ **Additional configuration**
Timeout, privileged, certificate, VPC, compute type, environment variables, file systems

buildspecs: use a buildspecs files



Buildspec

Build specifications

○ Insert build commands
Store build commands as build project configuration

● Use a buildspec file
Store build commands in a YAML-formatted buildspec file

Buildspec name - *optional*

By default, CodeBuild looks for a file named buildspec.yml in the source code root directory. If your buildspec file uses a different name or location, enter its path from the source root here (for example, buildspec-two.yml or configuration/buildspec.yml).

buildspec.yml

CREATE BUILD PROJECT

now go to

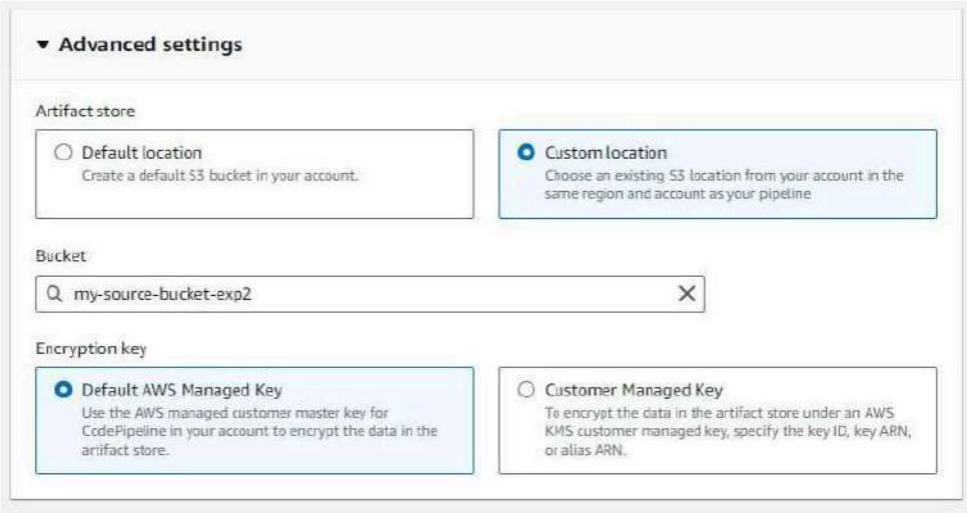
codepipline > pipelines > create a new pipeline

step 1 of 5
pipeline name: my-adl-pipline

advances settings-

artifact store : custom location

bucket: my-souce-bucket



▾ Advanced settings

Artifact store

○ Default location
Create a default S3 bucket in your account.

● Custom location
Choose an existing S3 location from your account in the same region and account as your pipeline

Bucket

my-source-bucket-exp2

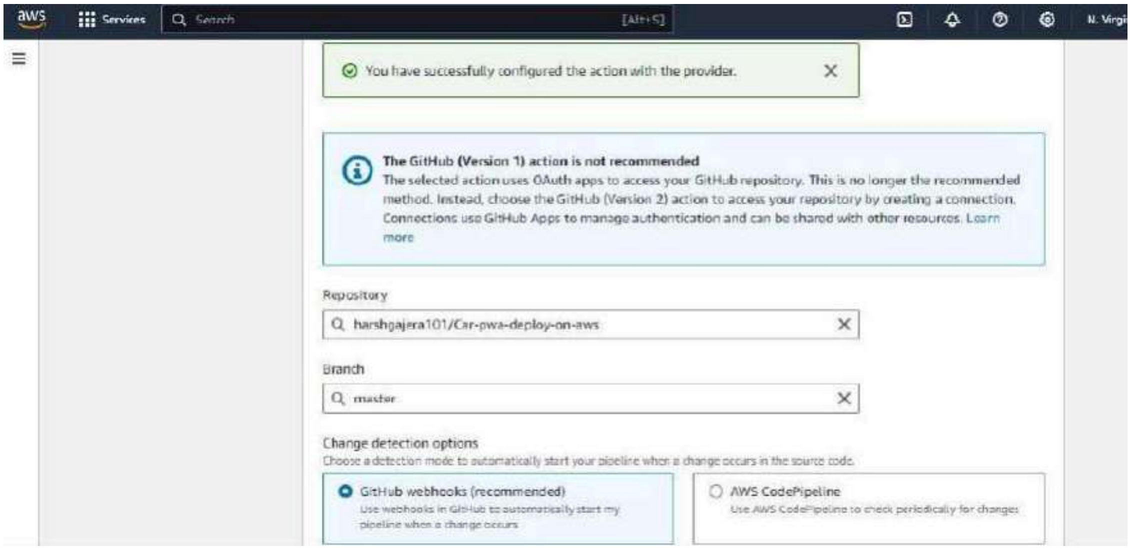
Encryption key

● Default AWS Managed Key
Use the AWS managed customer master key for CodePipeline in your account to encrypt the data in the artifact store.

○ Customer Managed Key
To encrypt the data in the artifact store under an AWS KMS customer managed key, specify the key ID, key ARN, or alias ARN.

step 2 of 5
repository :

branch: master



You have successfully configured the action with the provider.

The GitHub (Version 1) action is not recommended
The selected action uses OAuth apps to access your GitHub repository. This is no longer the recommended method. Instead, choose the GitHub (Version 2) action to access your repository by creating a connection. Connections use GitHub Apps to manage authentication and can be shared with other resources. Learn more

Repository

harshgajera101/Car-pwa-deploy-on-aws

Branch

master

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

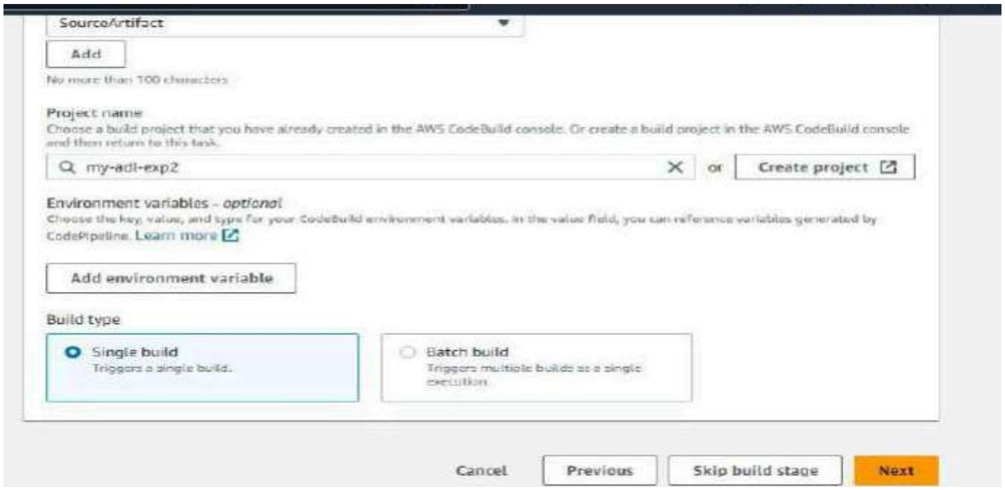
● GitHub webhooks (recommended)
Use webhooks in GitHub to automatically start my pipeline when a change occurs

○ AWS CodePipeline
Use AWS CodePipeline to check periodically for changes

step 3 of 5

project name: my-exp2

build type : single build



SourceArtifact

Add

No more than 100 characters

Project name
Choose a build project that you have already created in the AWS CodeBuild console. Or create a build project in the AWS CodeBuild console and then return to this task.

my-adl-exp2

Environment variables - *optional*
Choose the key, value, and type for your CodeBuild environment variables. In the value field, you can reference variables generated by CodePipeline. Learn more

Build type

● Single build
Triggers a single build.

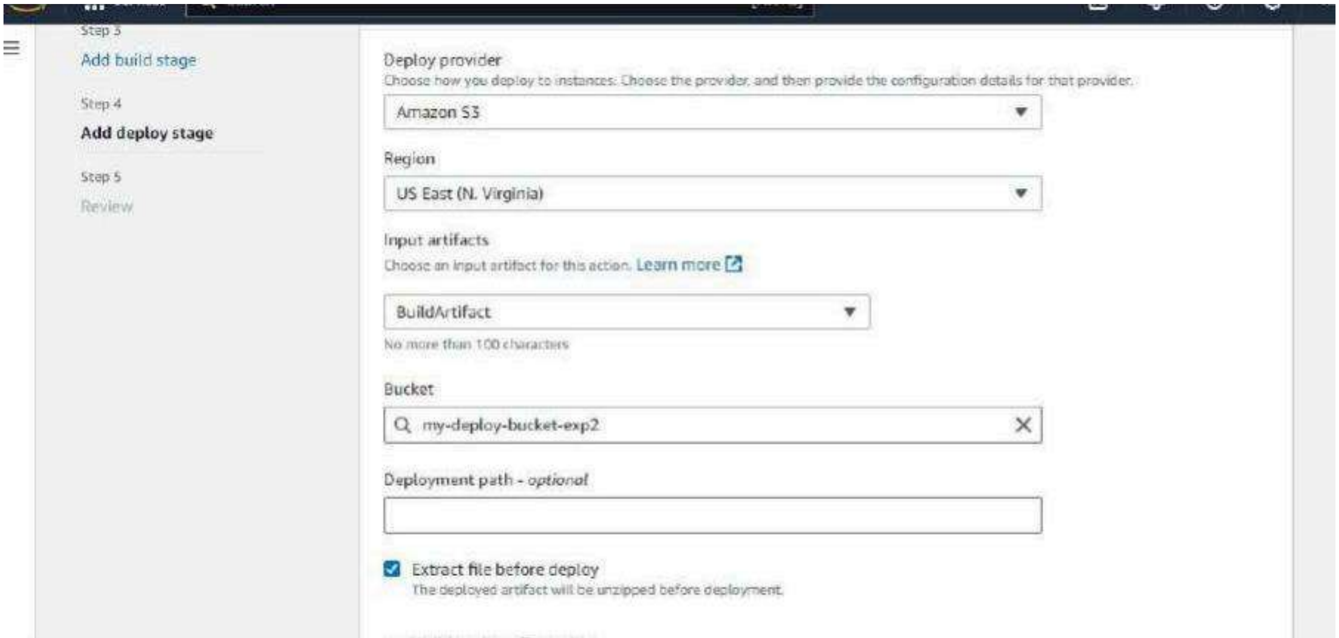
○ Batch build
Triggers multiple builds as a single execution.

step 4 of 5

deploy provider: amazon s3

region ; us east virginia

bucket: my-deploy-bucket



Step 3
Add build stage
Step 4
Add deploy stage
Step 5
Review

Deploy provider
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.
Amazon S3

Region
US East (N. Virginia)

Input artifacts
Choose an input artifact for this action. Learn more
BuildArtifact
No more than 100 characters

Bucket
my-deploy-bucket-exp2

Deployment path - *optional*

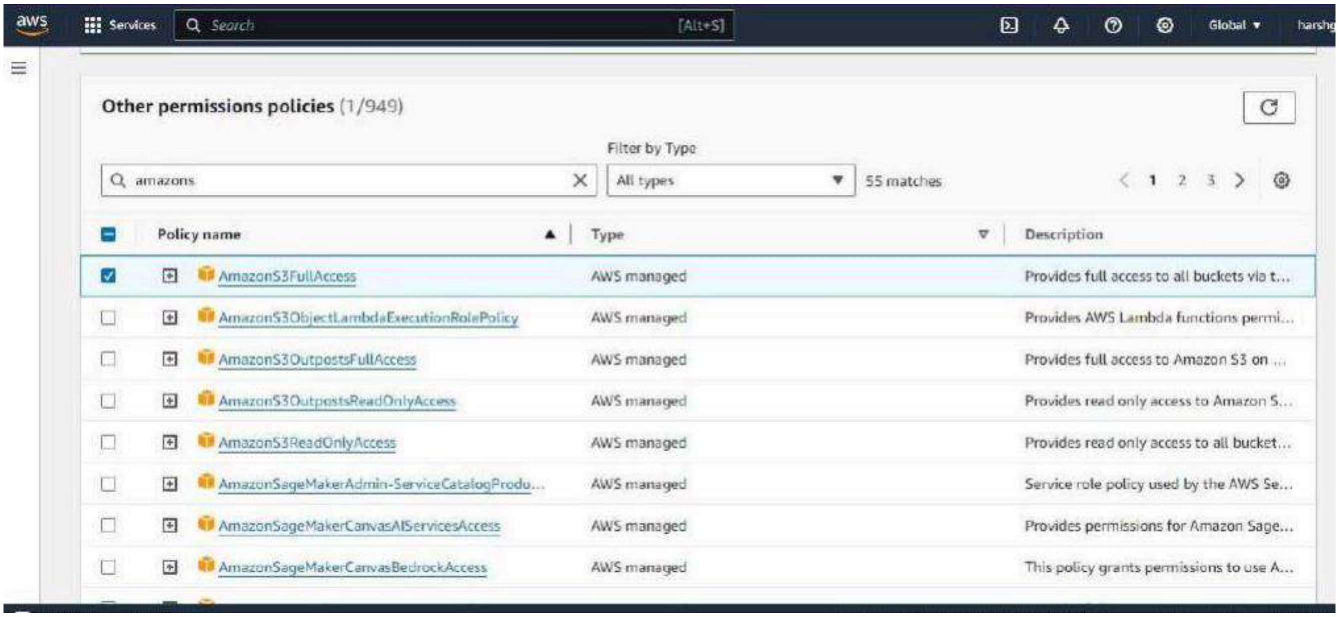
☑ Extract file before deploy
The deployed artifact will be unzipped before deployment.

CREATE PIPLINE

go to IAM , access management, roles

permission > add permissions policy

add : AmazonS3FullAccess



Other permissions policies (1/949)

Search: amazons

Filter by Type: All types

55 matches

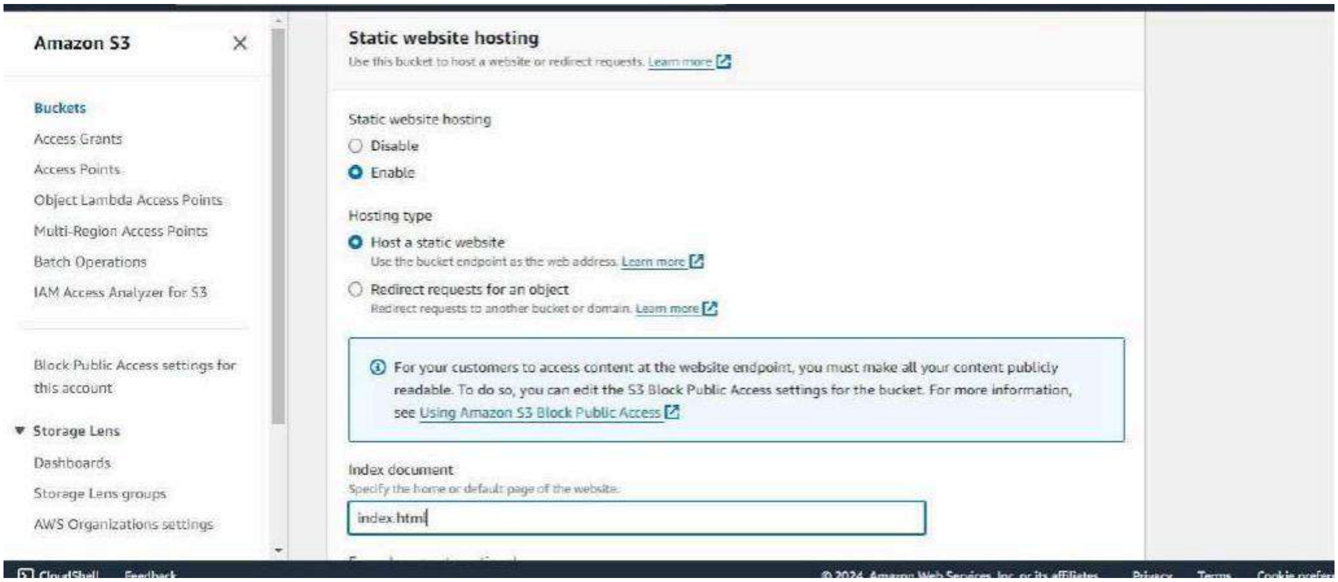
	Policy name	Type	Description
☑	AmazonS3FullAccess	AWS managed	Provides full access to all buckets via t...
□	AmazonS3ObjectLambdaExecutionRolePolicy	AWS managed	Provides AWS Lambda functions permi...
□	AmazonS3OutpostsFullAccess	AWS managed	Provides full access to Amazon S3 on ...
□	AmazonS3OutpostsReadOnlyAccess	AWS managed	Provides read only access to Amazon S...
□	AmazonS3ReadOnlyAccess	AWS managed	Provides read only access to all bucket...
□	AmazonSageMakerAdmin-ServiceCatalogProdu...	AWS managed	Service role policy used by the AWS Se...
□	AmazonSageMakerCanvasAIServicesAccess	AWS managed	Provides permissions for Amazon Sage...
□	AmazonSageMakerCanvasBedrockAccess	AWS managed	This policy grants permissions to use A...

After creating that go back to amazon s3 > buckets . my-deploy-bucket , click on it

go to objects / properties

then on static web hosting section

index document: index.html


Amazon S3
Buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms

Static website hosting

Use this bucket to host a website or redirect requests. Learn more

Static website hosting

○ Disable

● Enable

Hosting type

● Host a static website

Use the bucket endpoint as the web address. Learn more

○ Redirect requests for an object

Redirect requests to another bucket or domain. Learn more

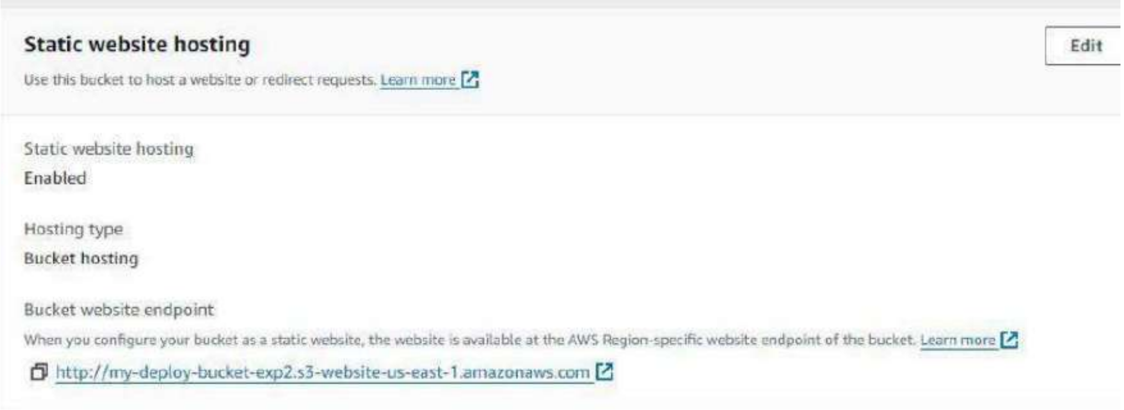
For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see Using Amazon S3 Block Public Access

Index document

Specify the home or default page of the website.

index.html

save



Static website hosting

Use this bucket to host a website or redirect requests. Learn more

Static website hosting

Enabled

Hosting type

Bucket hosting

Bucket website endpoint

When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket. Learn more

http://my-deploy-bucket-exp2.s3-website-us-east-1.amazonaws.com

go to amazon s3 bucket and change the bucket policy to

http://moodle.apsit.org.in/moodle/mod/resource/view.php?id=188320

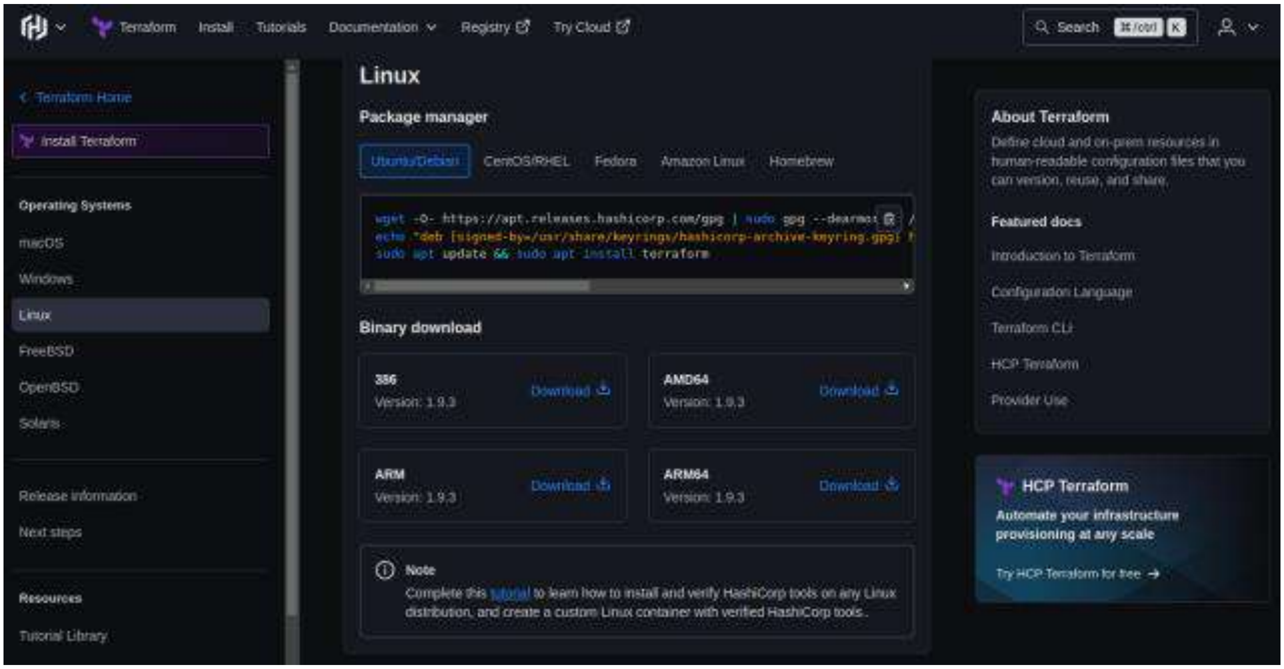
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::Bucket-Name/*"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

visit the link

EXP5

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

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



Linux

Package manager

```
wget -O- https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmo
echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg]
sudo apt update && sudo apt install terraform
```

Binary download

386 Version: 1.9.3

AMD64 Version: 1.9.3

ARM Version: 1.9.3

ARM64 Version: 1.9.3

Note: Complete this tutorial to learn how to install and verify HashiCorp tools on any Linux distribution, and create a custom Linux container with verified HashiCorp tools.

About Terraform

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

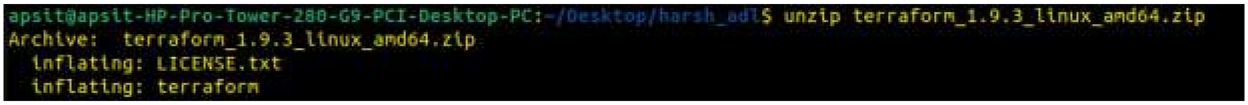
Featured docs

HCP Terraform

Automate your infrastructure provisioning at any scale

Step:2 unzip the archive by using below command

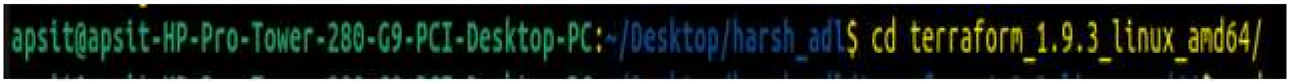
$unzip terraform_1.9.3_linux_amd64.zip



```
apsit@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:~/Desktop/harsh_adl$ unzip terraform_1.9.3_linux_amd64.zip
Archive:  terraform_1.9.3_linux_amd64.zip
  inflating: LICENSE.txt
  inflating: terraform
```

Step 3: Change the directory to unzipped folder

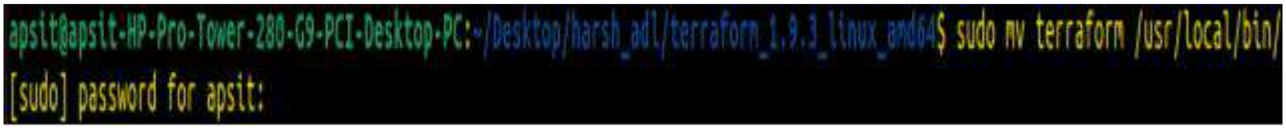
$cd terraform_1.9.3_linux_amd64/



```
apsit@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:~/Desktop/harsh_adl$ cd terraform_1.9.3_linux_amd64/
```

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

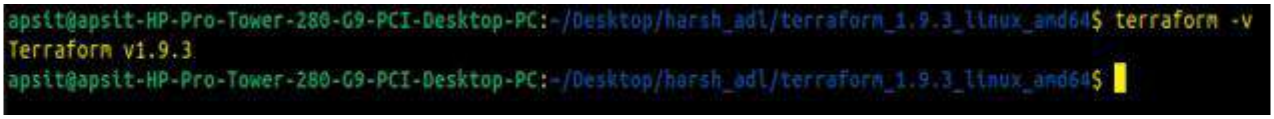
$sudo mv terraform /usr/local/bin/



```
apsit@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:~/Desktop/harsh_adl/terraform_1.9.3_linux_amd64$ sudo mv terraform /usr/local/bin/
[sudo] password for apsit:
```

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

$terraform -v



```
apsit@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:~/Desktop/harsh_adl/terraform_1.9.3_linux_amd64$ terraform -v
Terraform v1.9.3
apsit@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:~/Desktop/harsh_adl/terraform_1.9.3_linux_amd64$
```

EXP 5:To Build, change, and destroy AWS infrastructure Using Terraform.

Cmd:

```
$ sudo apt-get install curl
$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o"awscliv2.zip"
$ sudo apt install unzip
$ sudo unzip awscliv2.zip
$ sudo ./aws/install
$ aws --version
```

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

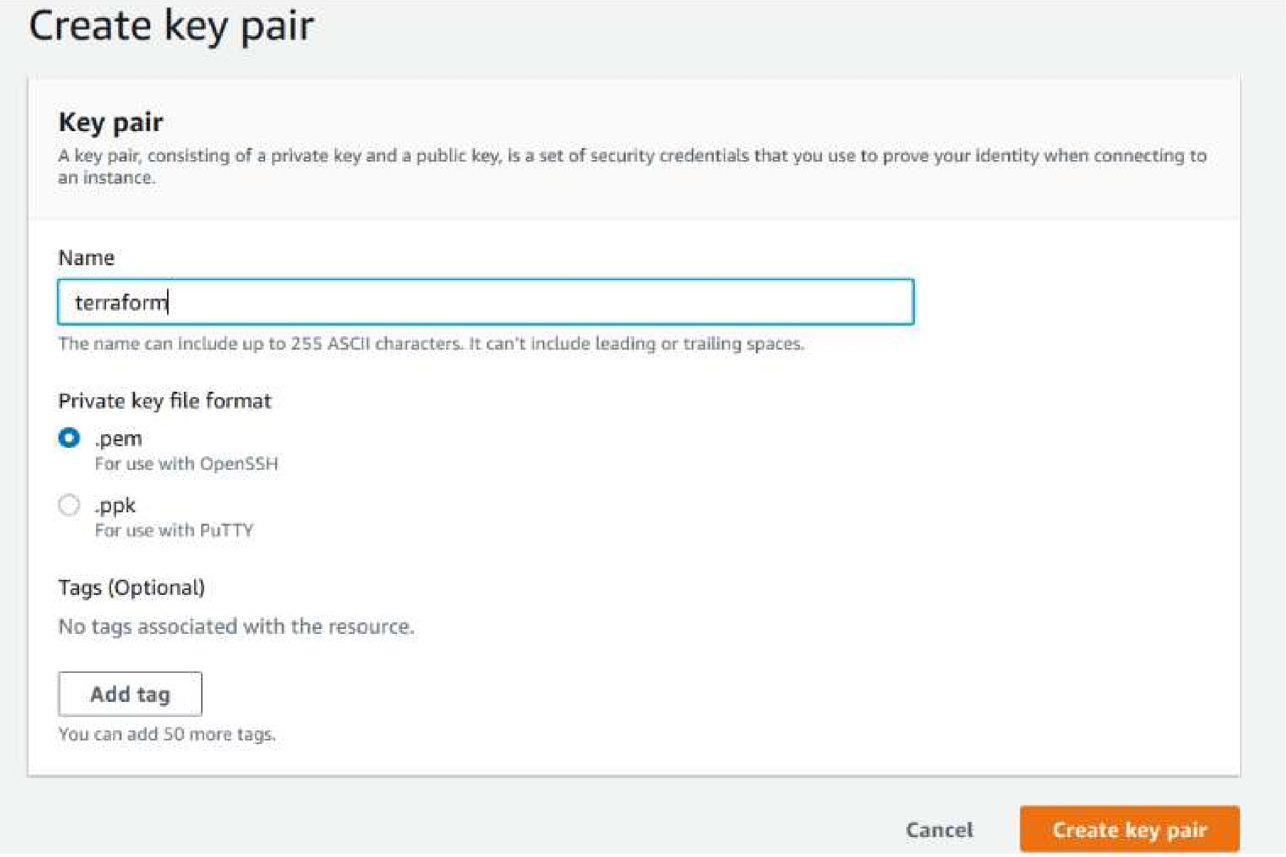
Login to AWS console, click on username and go to My security credentials.
Continue on security credentials, click on access keys
create an access key and copy both(Access kay and Secret access key)

```
$ aws configure
```

paste it here(): Access kay and Secret access key value **and** region as us-east-1

```
$ cd ~
$ mkdir project-terraform
$ cd project-terraform
```

Create key pair[AWS]: name teraform



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.

You can add 50 more tags.

```
$ sudo nano variables.tf
```

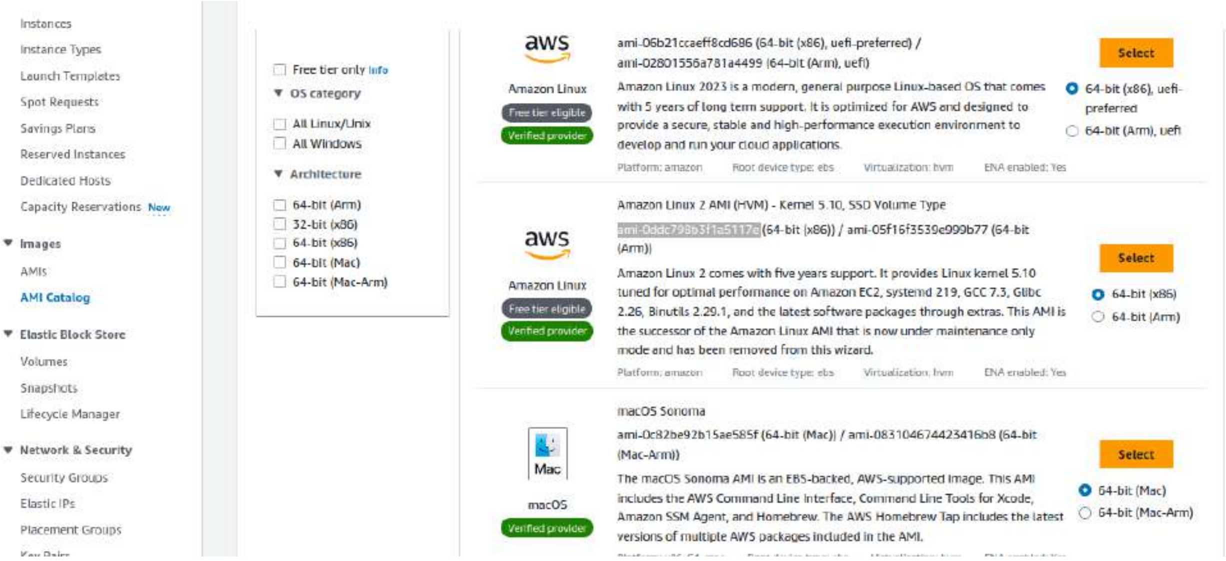
code:

```
variable "aws_region" {
  description = "AWS region"
  default     = "us-east-1"
}

variable "key_name" {
  description = "AWS key"
  default     = "terraform"
}

variable "instance_type" {
  description = "Instance type"
  default     = "t2.micro"
}
```

AWS:search AMI catalog and copy the ami of second AMI Given below:



ami-06b21ccaeff8cd686 (64-bit (x86), uefi-preferred) / ami-02801556a781a4499 (64-bit (Arm), uefi)

Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Platform: amazon Root device type: ebs Virtualization: hvm ENA enabled: Yes

● 64-bit (x86), uefi-preferred
○ 64-bit (Arm), uefi

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type

ami-0ddc798b3f1a5117e (64-bit (x86)) / ami-05f16f3539e999b77 (64-bit (Arm))

Amazon Linux 2 comes with five years support. It provides Linux kernel 5.10 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is now under maintenance only mode and has been removed from this wizard.

Platform: amazon Root device type: ebs Virtualization: hvm ENA enabled: Yes

● 64-bit (x86)
○ 64-bit (Arm)

macOS Sonoma

ami-0c82be92b15ae585f (64-bit (Mac)) / ami-0831046744234l6b8 (64-bit (Mac-Arm))

The macOS Sonoma AMI is an EBS-backed, AWS-supported image. This AMI includes the AWS Command Line Interface, Command Line Tools for Xcode, Amazon SSM Agent, and Homebrew. The AWS Homebrew Tap includes the latest versions of multiple AWS packages included in the AMI.

● 64-bit (Mac)
○ 64-bit (Mac-Arm)

ami-0ddc798b3f1a5117e(Paste this in the highlighted text)

```
$ 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 Jenkins server
  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-0ddc798b3f1a5117e"
  key_name      = var.key_name
  instance_type = var.instance_type
  security_groups = [aws_security_group.security_jenkins_port.name]

  tags = {
    Name = "jenkins_instance"
  }
}

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

  tags = {
    Name = "jenkins_elastic_ip"
  }
}
```

```
$ terraform init
$ terraform plan
$ terraform apply
```

Take Screenshots of Instances and Security group

```
$ terraform destroy
```

EXP7:To understand Static Analysis SAST process and learn to integrate Jenkins SAST to SonarQube/GitLab.

```
$ docker run -d -p 9000:9000 sonarqube
localhost:9000
```

if doesn’t work

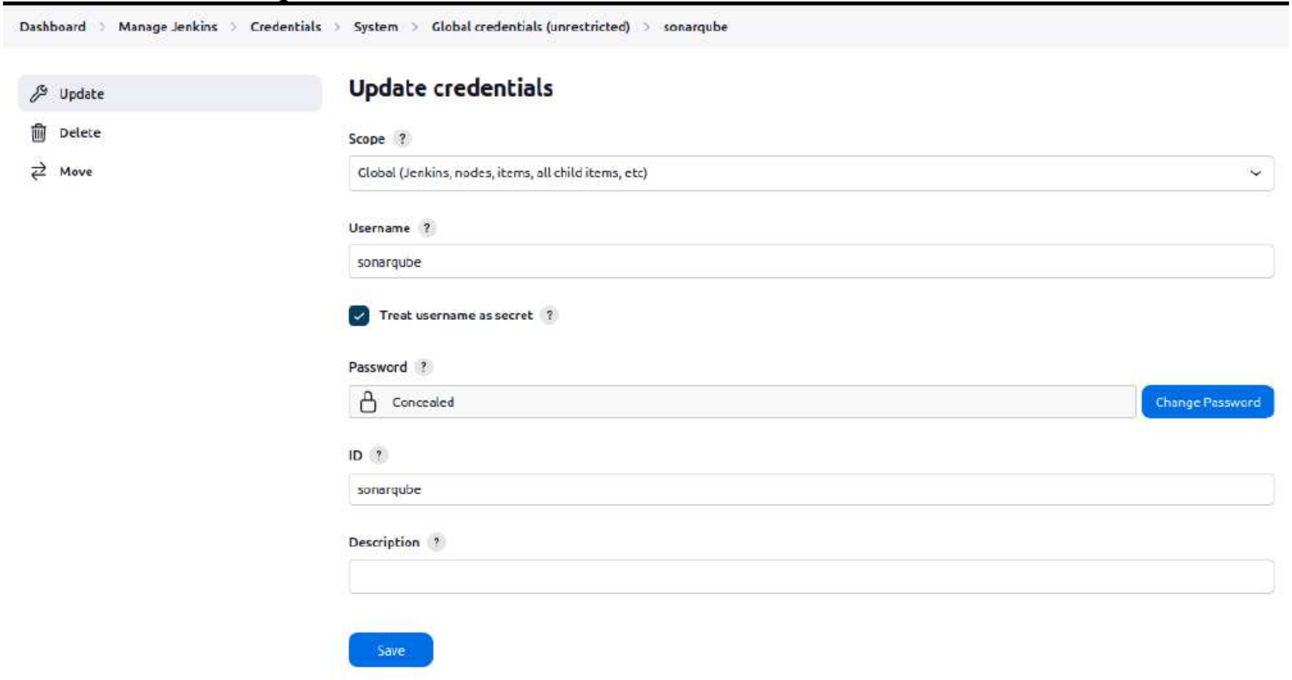
```
sudo docker ps
sudo docker logs <container id>
```

login& pass : admin

Now goto (Right upper corner)Administrator > My Account > Security

Create token name **jenkin** and copy the code
squ_7076fc9c35736f13af2d467ee81e611a66721562

Jenkin create sonarqube


Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) > sonarqube
Update
Delete
Move
Change Password
Save

Update credentials

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

Username: sonarqube

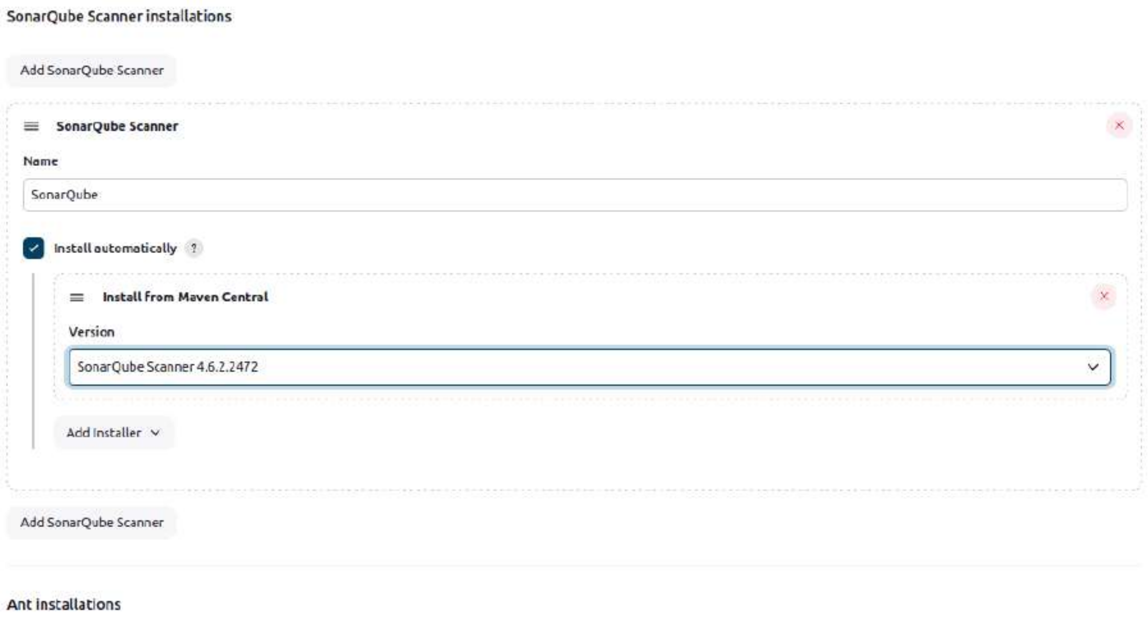
☑ Treat username as secret

Password: Concealed

ID: sonarqube

Description:

Manage Jenkins >Tools > SonarQube Scanner.



SonarQube Scanner installations

SonarQube Scanner

Name: SonarQube

☑ Install automatically

Install from Maven Central

Version: SonarQube Scanner 4.6.2.2472

Ant installations

Save it

Manage > New Item > SonarQube(select Pipeline) >save

Description
Hello pipeline

Github project: https://github.com/vishal003/jenkins-sonarqube/

pipeline script:

```
node {
    stage('cloning from GIT') {
        git branch: 'main', credentialsId: 'GIT_REPO', url: 'http://github.com/vishal003/jenkins-sonarqube/'
    }
}
```

Jenkin
Click build now and click #1 and take screenshot.

Exp11: To understand AWS Lambda, its workflow, various functions and create your first Lambda functions using Python / Java / Nodejs.

AWS Console:

Lambda>Create function

name: Sum
runtime:Python 3.12
create it

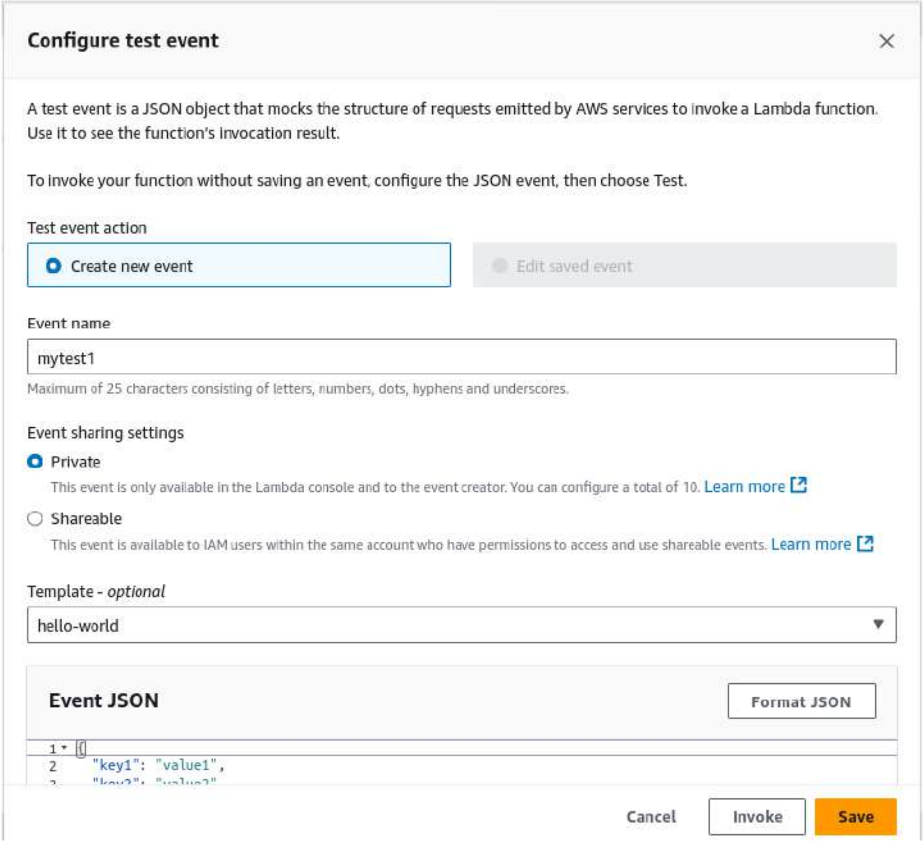
Source code: > lambda_function :code

```
import json

def lambda_handler(event,context):
        a = 10
        b = 20
        c = a + b
        return c
```

Test

name:mytest1
save it



Configure test event

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

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

Test event action

● Create new event ○ Edit saved event

Event name: mytest1

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

Event sharing settings

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

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

Template - *optional*: hello-world

Event JSON

```
{
  "key1": "value1",
  "key2": "value2"
```

Test it

Second sample python Code:

```
def lambda_handler(event,context):
    for i in range(3):
        print("Hello")
```

Test

name: mytest2
save it

Test it

Exp12: To create a Lambda function which will log “An Image has been added” once you add an object to a specific bucket in S3

AWS Console:

S3 buckets > Create bucket > my-lambda-bucket-1

IAM > Roles > Create role

Select Service or use case – Lambda

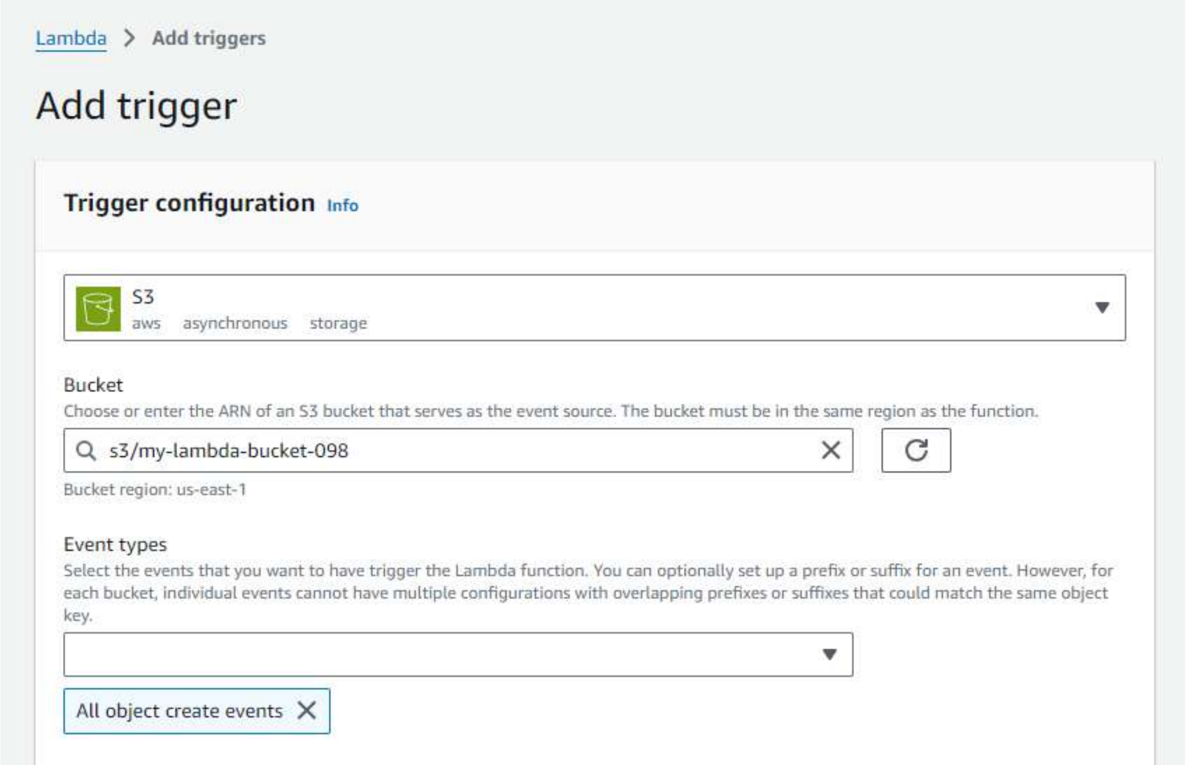
Add permissions:
AmazonS3FullAccess, AWSLambda_FullAccess and CloudWatchFullAccess

Give Role name

see the permissions and create

AWS Console:
Lambda >Create a Function> name:lambdawiths3
in this:
Change default execution role
select : **Use an existing role**
Existing role: <Your created role>
Create

now scroll down:
Configuration > Triggers >Add Triggers



Add trigger

Trigger configuration Info

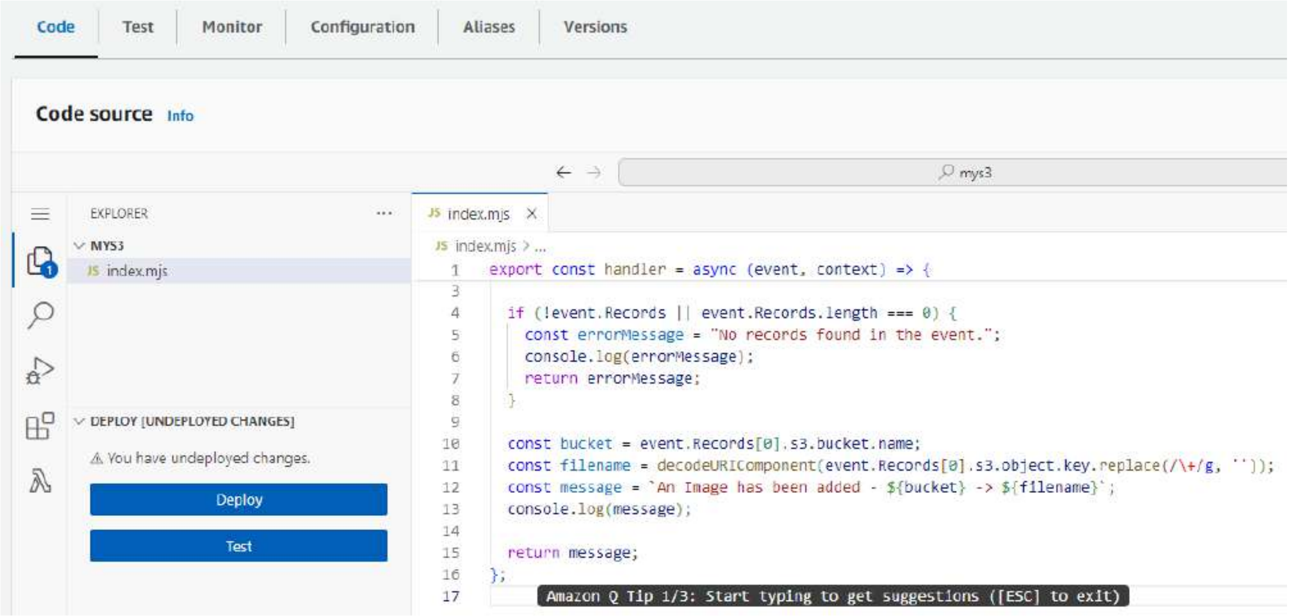
S3
aws asynchronous storage

Bucket
Choose or enter the ARN of an S3 bucket that serves as the event source. The bucket must be in the same region as the function.
s3/my-lambda-bucket-098
Bucket region: us-east-1

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

All object create events

Add it



⚠ You have undeployed changes.

```
export const handler = async (event, context) => {

  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;
};
```

Amazon Q Tip 1/3: Start typing to get suggestions ([ESC] to exit)

Add this code:

```
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;
};
```

Save and test it

AWS console
Search Buckets > click on [my-lambda-bucket-1] > upload
Upload any one image.jpg

And then
Search Logs > Logs groups > select your created bucket > select the displayed logs stream
Take a screenshot of it