

NAME: MAYURI SHRIDATTA YERANDE

ROLL NO: 61

CLASS : D16B

ADVANCE DEVOPS

EXPERIMENT - ①

EXPERIMENT NO - 1

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 an integrated development environment or IDE.

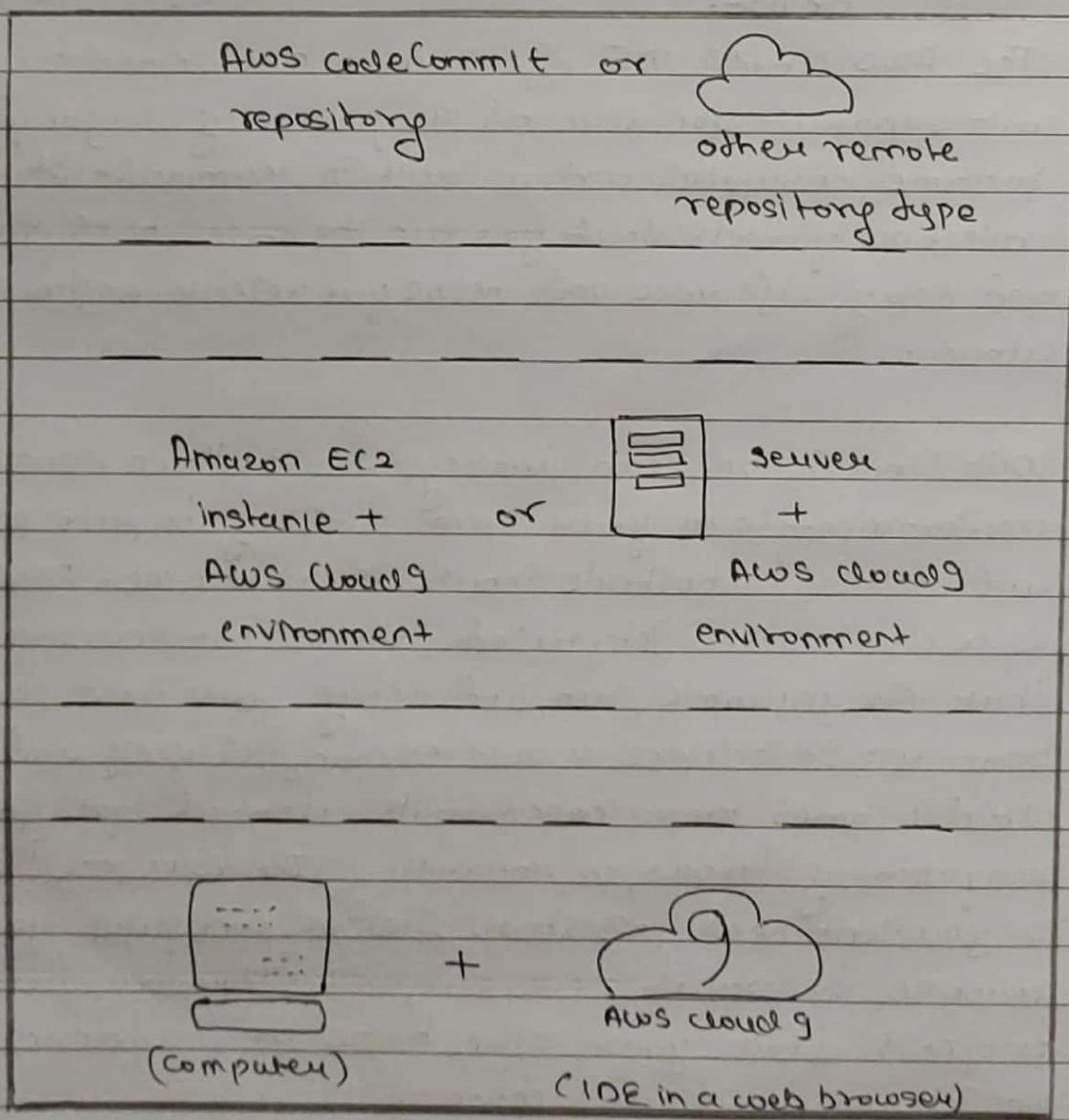
The AWS Cloud9 IDE offers a rich code-editing experience with support for several programming languages and runtime debuggers and a built-in terminal. It contains a collection of tools that you use to code, build, run, test and debug software and helps you release software to the cloud.

With Cloud9, one can quickly share their development environment with your team, enabling to pair programs and track each other's inputs in real time. 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 so you don't need to install files or configure your machine to start new projects. You can work on your projects from your office, home or anywhere using an Internet-connected machine.

Cloud9 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.

How does AWS Cloud9 work?

The following diagram shows a high-level overview of how AWS Cloud9 works.



From the diagram, you use the AWS Cloud9 IDE, running in a web browser on your local computer, to interact with your AWS Cloud9 environment. It connects to that environment. Finally, code is stored in an AWS CodeCommit repository or other type of remote repository.

With AWS Cloud9, you can code, build, run, test, debug and release software in many exciting scenarios and variations. These include:-

- Working with code in several programming languages and the AWS Cloud Development Kit (CDK).
- Working with code in a running Docker container.
- Using online code repositories.
- Collaborating with others in real time.
- Interacting with various database and website technologies.
- Targeting AWS Lambda, Amazon API Gateway and AWS Serverless applications.
- Taking advantage of other AWS products such as Amazon Lightsail, AWS CodeStar and AWS CodePipeline.

Benefits:-

- CODE WITH JUST A BROWSER

AWS Cloud 9 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 one can write, run and debug applications with just a browser, without needing to install or maintain a local IDE. The Cloud 9 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 Cloud 9 makes collaborating on code easy. You can share your development environment with your team in just a few clicks and pair programs together. Your team can instantly chat with one another within the IDE.

- BUILD SERVERLESS APPLICATIONS WITH EASE

AWS Cloud 9 makes it easy to write, run and debug serverless applications. It pre-configured the development environment with all the SDKs, libraries and plug-ins needed for serverless development.

- DIRECT TERMINAL ACCESS TO AWS

It makes it easy to quickly run commands and comes with a terminal that includes sudo privileges.

do the managed Amazon E22 instance that is hosting your development environment and a preauthenticated AWS command line interface.

- START NEW PROJECTS QUICKLY

Cloud9's development environment comes prepackaged with tooling for over 40 programming languages. This enables you to start coding code for popular application stacks within minutes by eliminating the need to install or configure files. You can easily maintain multiple development environments to isolate your project's resources.

IMPLEMENTATION

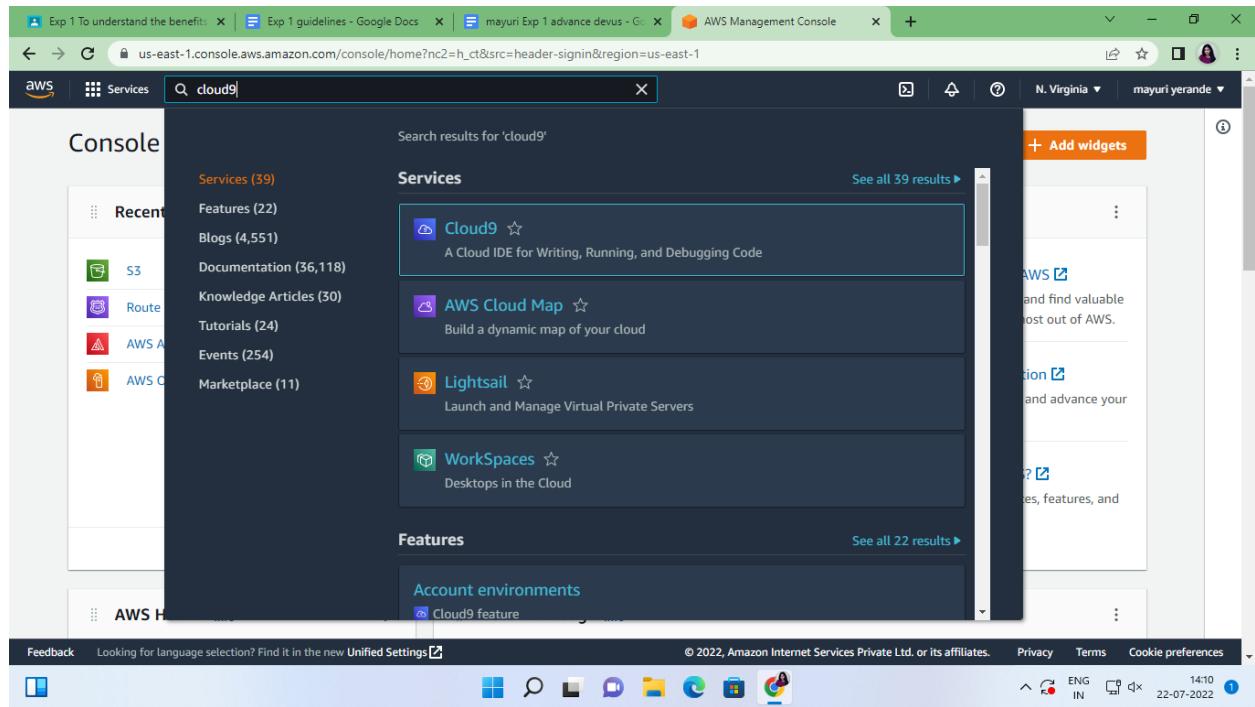
Steps:

1. Login with your AWS account.

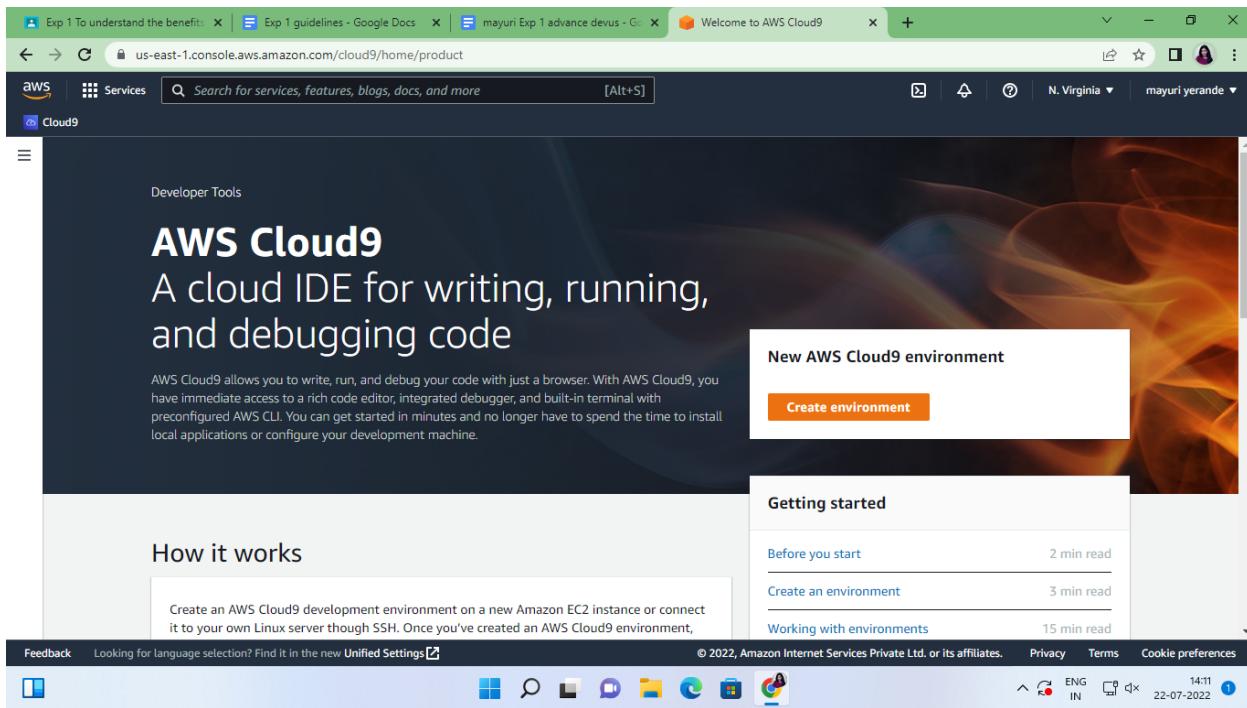
Create an AWS Account(Free)

<https://www.youtube.com/watch?v=oUoJBayrJT4>

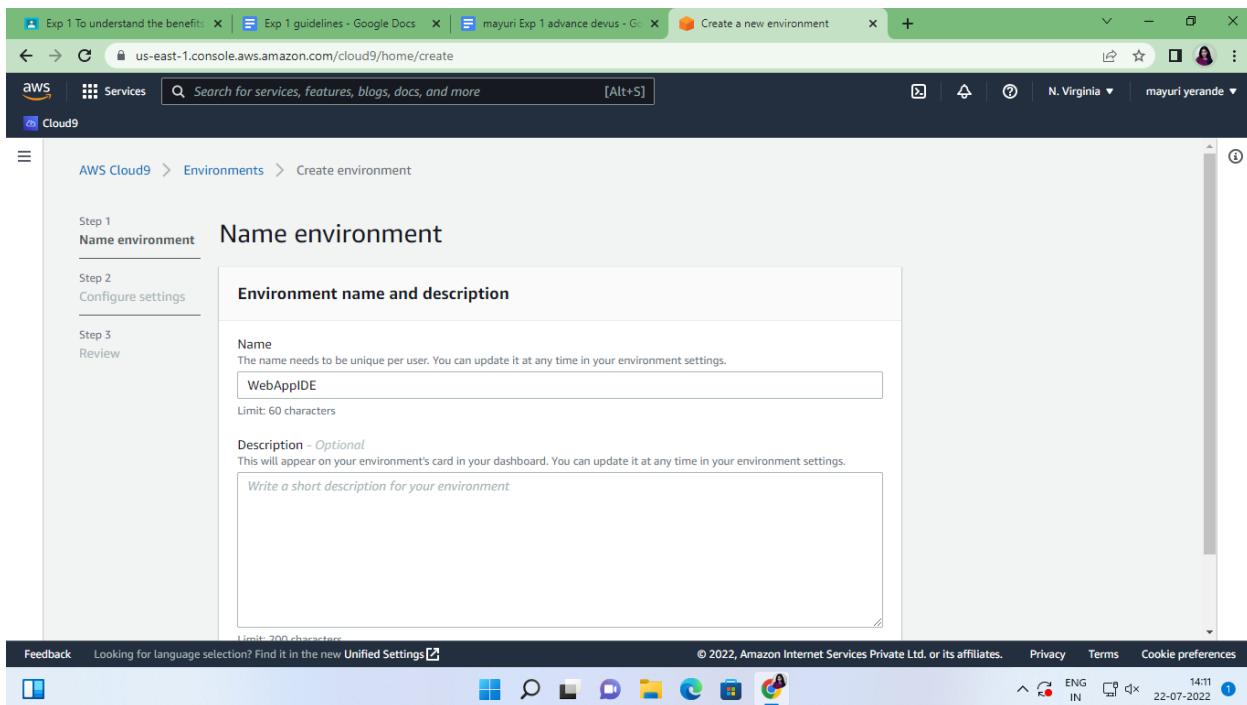
2. Navigate to Cloud 9 service from Developer tools section as below:



3. Click on Create Environment :



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



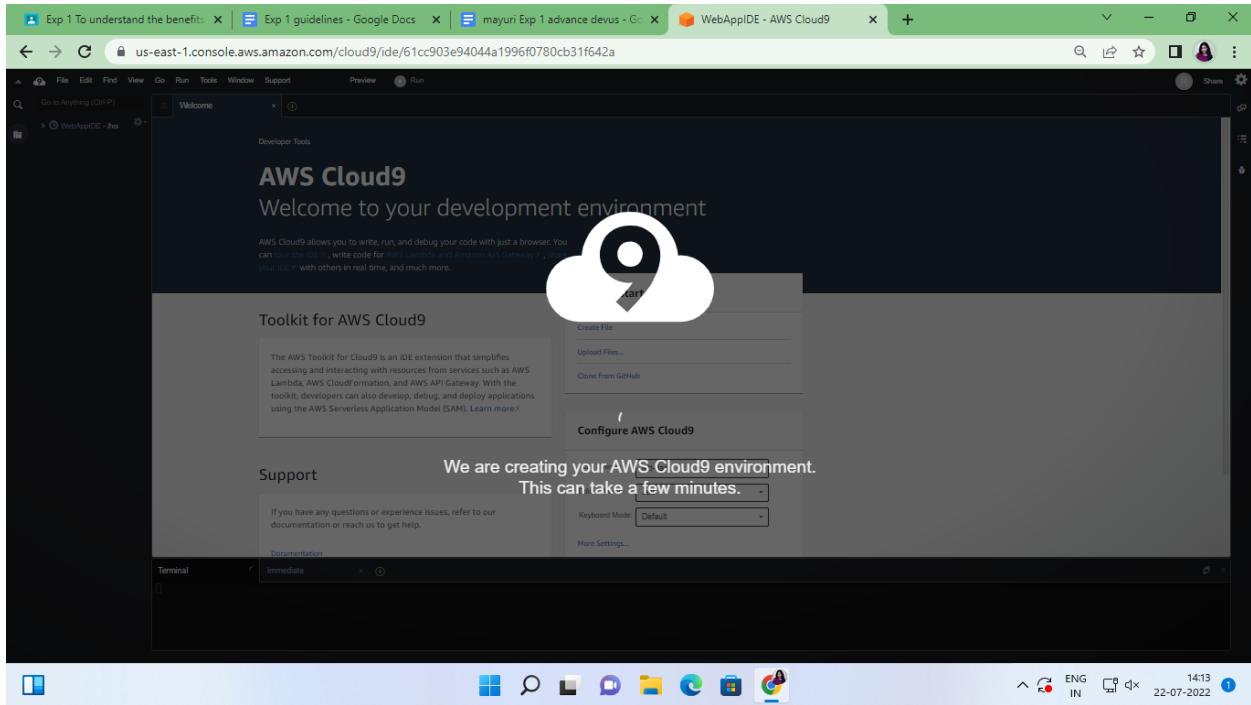
5. Keep all the Default settings as shown in below:

The screenshot shows the 'Configure settings' step of the AWS Cloud9 environment creation process. The 'Environment type' is set to 'EC2'. The 'Instance type' is selected as 't2.micro'. The 'Platform' is set to 'Amazon Linux 2 (recommended)'. The 'Cost-saving setting' is set to 'After 30 minutes (default)'. The 'IAM role' is 'AWSServiceRoleforAWSCloud9 (generated)'. A note at the bottom recommends best practices for using the AWS Cloud9 environment.

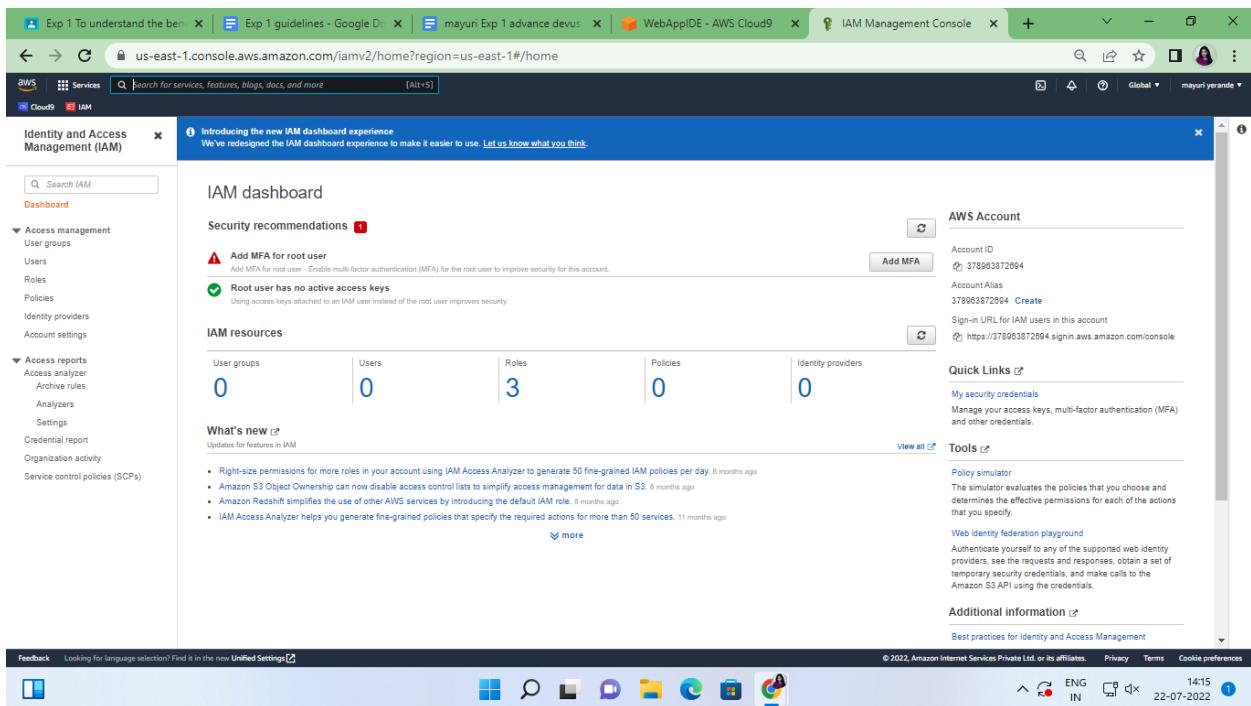
6. Review the Environment name and Settings and click on Create Environment:

The screenshot shows the 'Environment name and settings' step of the AWS Cloud9 environment creation process. The environment is named 'WebAppIDE'. The settings are identical to the previous screenshot. A note at the bottom reiterates the best practices for using the AWS Cloud9 environment. At the bottom right, there is a large orange 'Create environment' button.

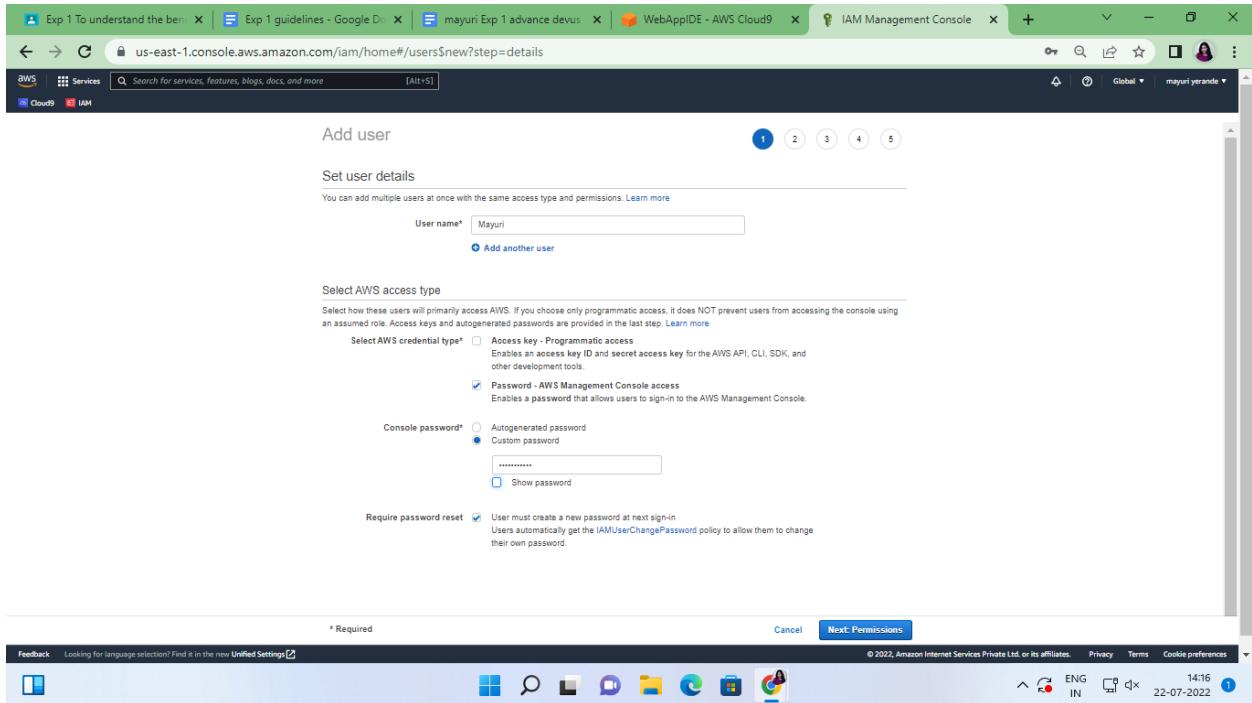
It will take a few minutes to create an aws instance for your Cloud 9 Environment.



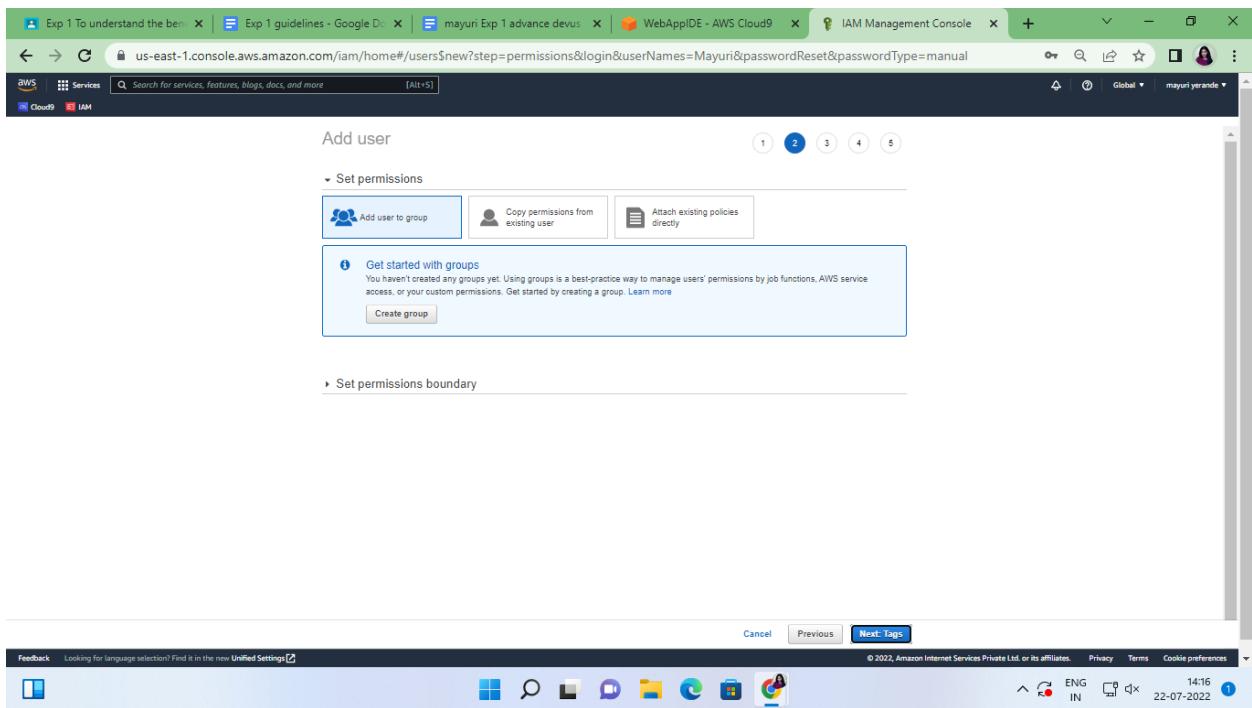
7. Till that time open IAM Identity and Access Management in order to Add user In another tab.



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



9. Click on Create group



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

Create policy Refresh

Filter policies Showing 753 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 resources needed by Amplify applications.
AdministratorAccess-AWSElasticBeanstalk	AWS managed	None	Grants account administrative permissions. Explicitly allows developers and administrators to gain direct access to resources they need.
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
AlexaForBusinessLifesizeDelegatedAccessPolicy	AWS managed	None	Provide access to Lifesize AVS devices
AlexaForBusinessPolyDelegatedAccessPolicy	AWS managed	None	Provide access to Poly AVS devices
AlexaForBusinessReadOnlyAccess	AWS managed	None	Provide read only access to AlexaForBusiness services
AmazonCloudWatchLogsFullAccess	AWS managed	None	Provides full access to CloudWatch Logs API and CloudWatch Metrics API. Also provides CloudWatch Metrics Insights access.

Create group Cancel

Add user

1 2 3 4 5

Set permissions

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

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

Add user to group

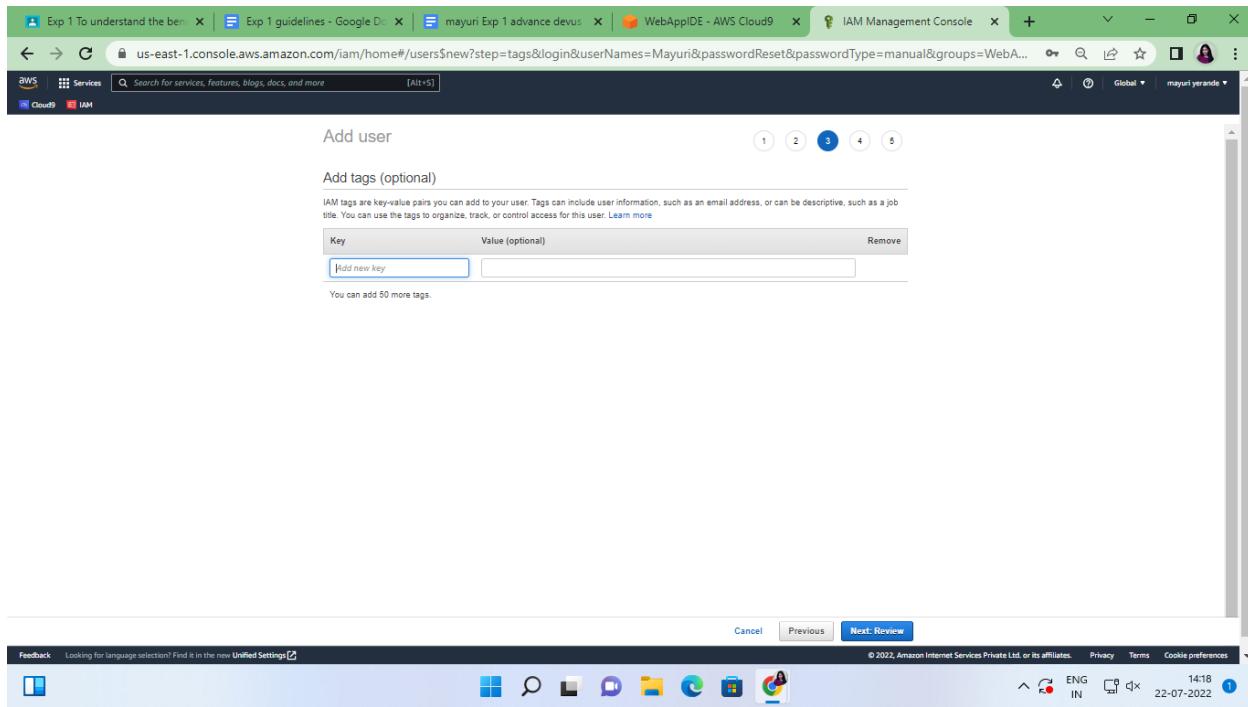
Create group Refresh

Search Group Attached policies Showing 1 result

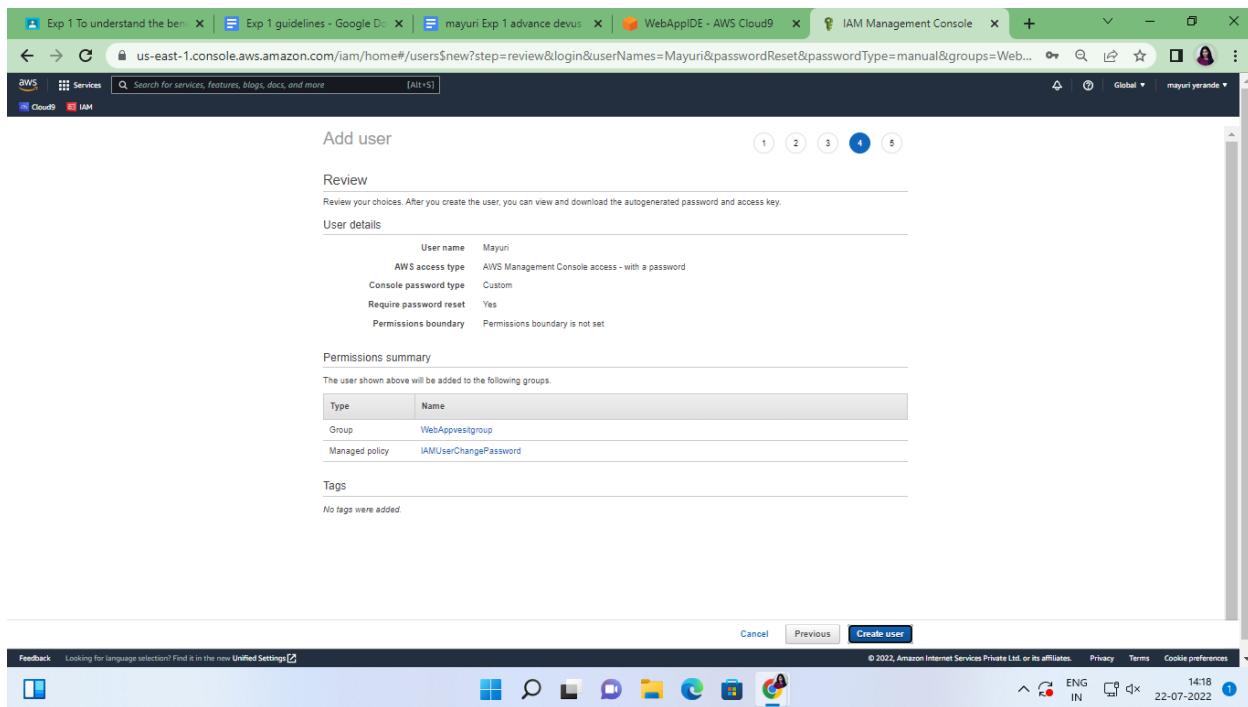
WebAppvisitgroup None

Set permissions boundary

Cancel Previous Next: Tags



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

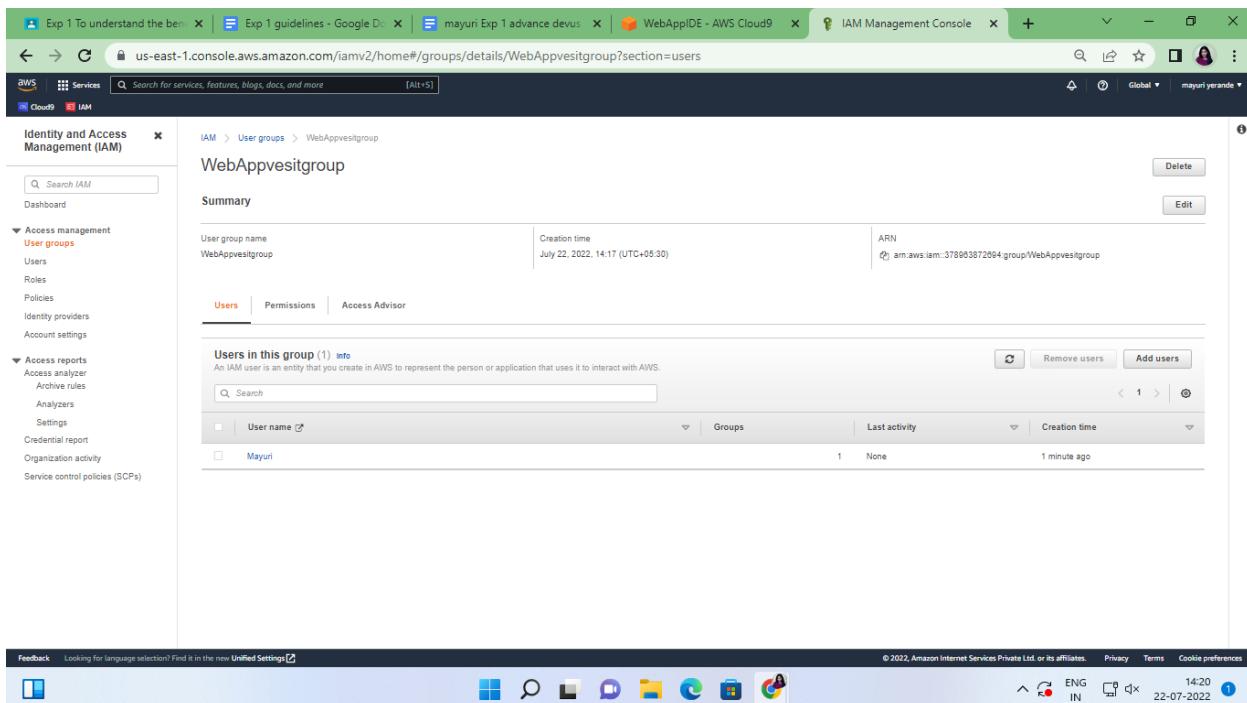


The screenshot shows the AWS IAM Management Console with a success message: "You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time." Below the message, there is a table with one row for "Mayuri". The table has columns for "User" and "Email login instructions". A "Send email" button is located next to the "Email login instructions" column.

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

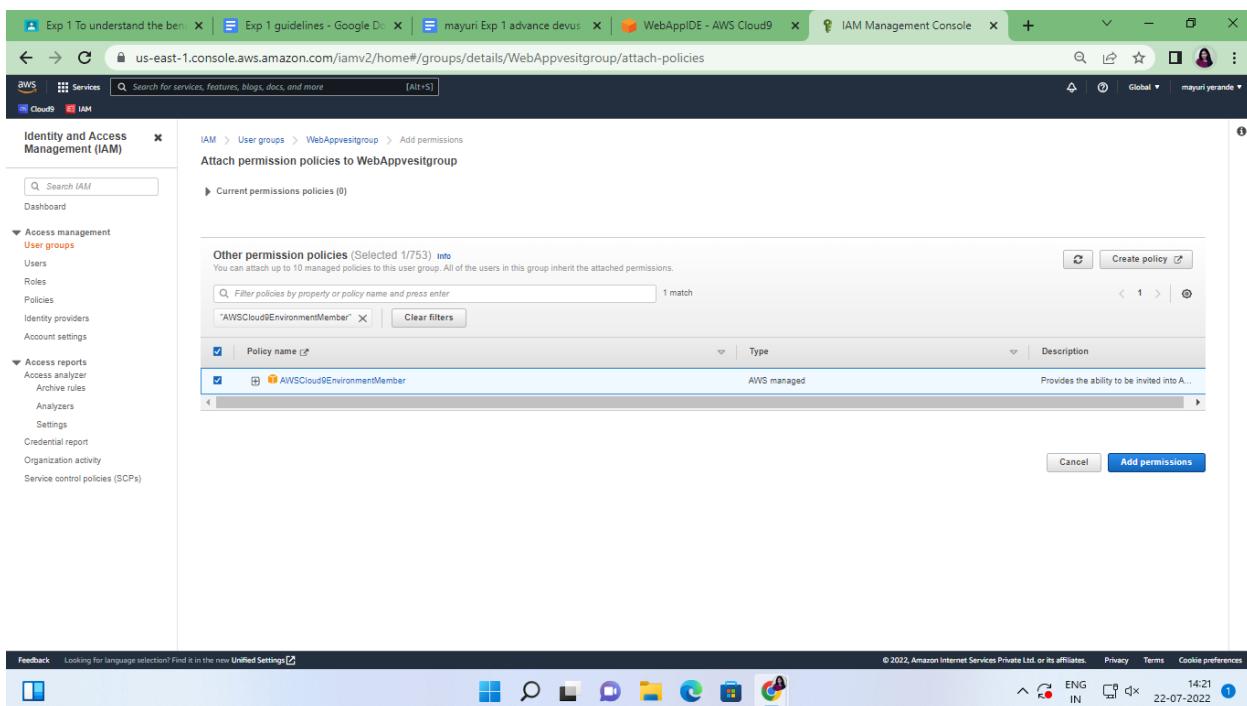
The screenshot shows the AWS IAM Management Console with the "User groups" page selected. The left sidebar shows the navigation menu for IAM. The main content area displays a table titled "User groups (1) info". The table has one row for "WebAppsgroup". The table includes columns for "Group name", "Users", "Permissions", and "Creation time". A "Create group" button is located at the top right of the table area.

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



The screenshot shows the AWS IAM Management Console. In the left sidebar, under 'Access management', 'User groups' is selected. The main content area shows a user group named 'WebAppvesitgroup'. The 'Users' tab is active, displaying a single user 'Mayuri'. The ARN of the group is listed as 'arn:aws:iam:378963872594:group/WebAppvesitgroup'.

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



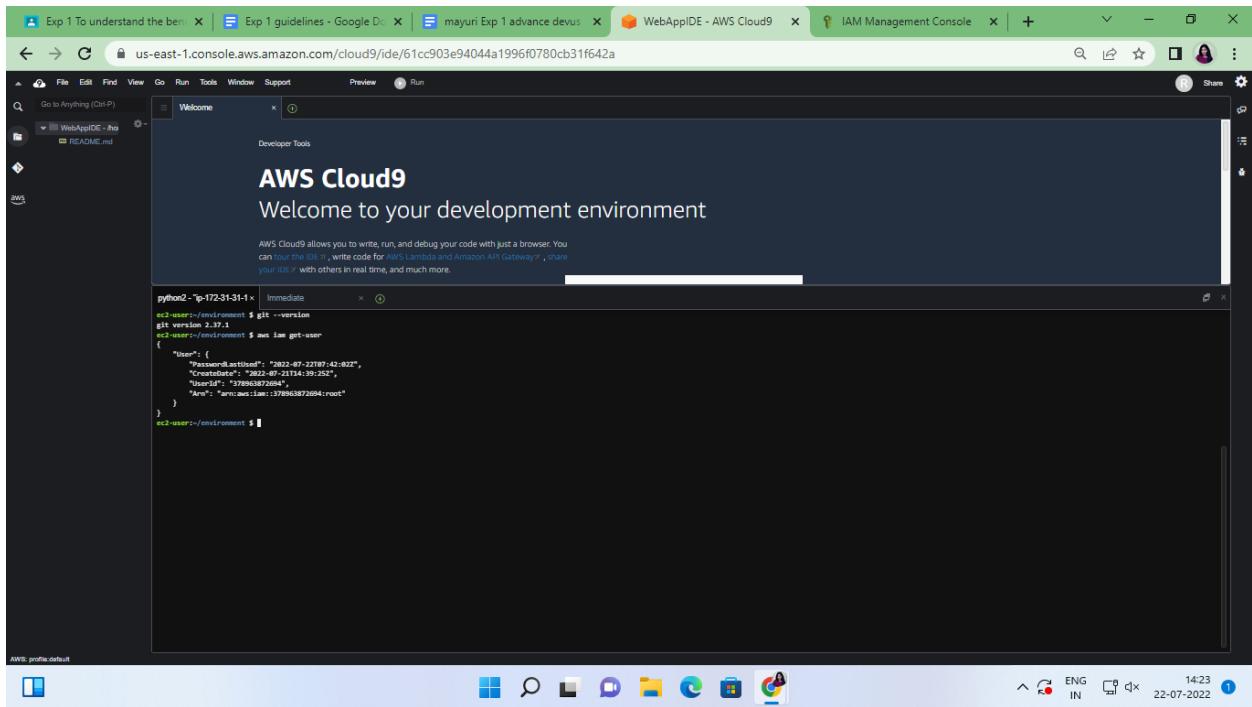
The screenshot shows the 'Attach permission policies to WebAppvesitgroup' page. Under 'Other permission policies', a policy named 'AWScloud9EnvironmentMember' is selected and highlighted. The 'Add permissions' button is visible at the bottom right.

The screenshot shows the AWS IAM Management Console interface. On the left, there's a navigation sidebar with options like 'User groups', 'Access reports', and 'Service control policies (SCPs)'. The main area is titled 'WebAppvesitgroup' under 'User groups'. It shows a summary with the user group name 'WebAppvesitgroup', creation time 'July 22, 2022, 14:17 (UTC+05:30)', and ARN 'arn:aws:iam::378953872694:group/WebAppvesitgroup'. Below this, there are tabs for 'Users', 'Permissions' (which is selected), and 'Access Advisor'. Under 'Permissions', it says 'Permissions policies (1) [info](#)' and lists a single policy: 'AWSCloud9EnvironmentMember' (AWS managed). A tooltip for this policy states: 'Provides the ability to be invited into AWS Cloud9 shared environments'. At the bottom of the page, there's a feedback link, a toolbar with various icons, and a footer with copyright information and a timestamp '22-07-2022 14:21'.

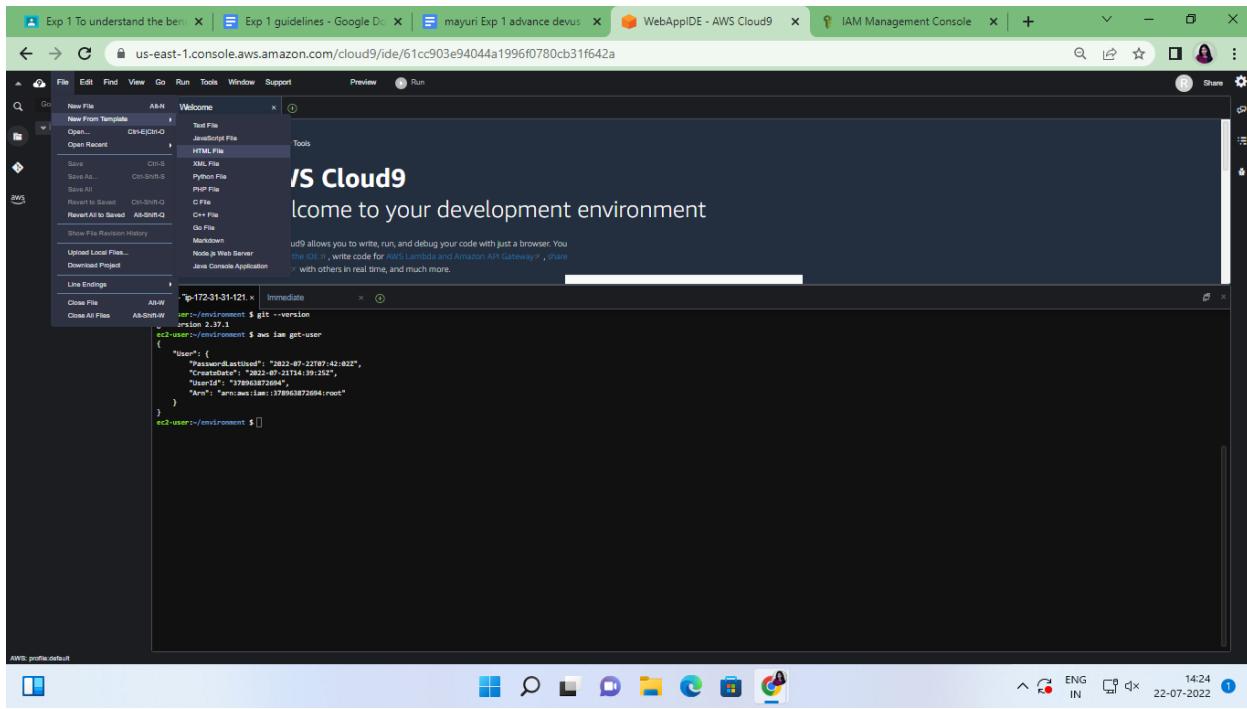
15. now we move towards our cloud9 IDE Environment tab it shows as shown :

The screenshot shows the AWS Cloud9 IDE Environment tab. The interface has a dark theme. On the left, there's a file browser showing files like 'WebAppIDE - hml', '.c9', and 'README.md'. Below the file browser is a 'Developer Tools' section. The main area features a large 'AWS Cloud9' header with the subtext 'Welcome to your development environment'. Below this, there's a 'Toolkit for AWS Cloud9' section with a description of its purpose and a 'Getting started' sidebar with options like 'Create File', 'Upload Files...', and 'Clone from GitHub'. There's also a 'Configure AWS Cloud9' sidebar with dropdowns for 'Main Theme' (set to 'jet-dark'), 'Editor Theme' (set to 'Jet'), and 'Keyboard Mode' (set to 'Default'). At the bottom, there's a terminal window titled 'bash - [ip-172-31-31-121]' showing a command prompt. The bottom of the screen has a toolbar with various icons and a footer with a timestamp '22-07-2022 14:21'.

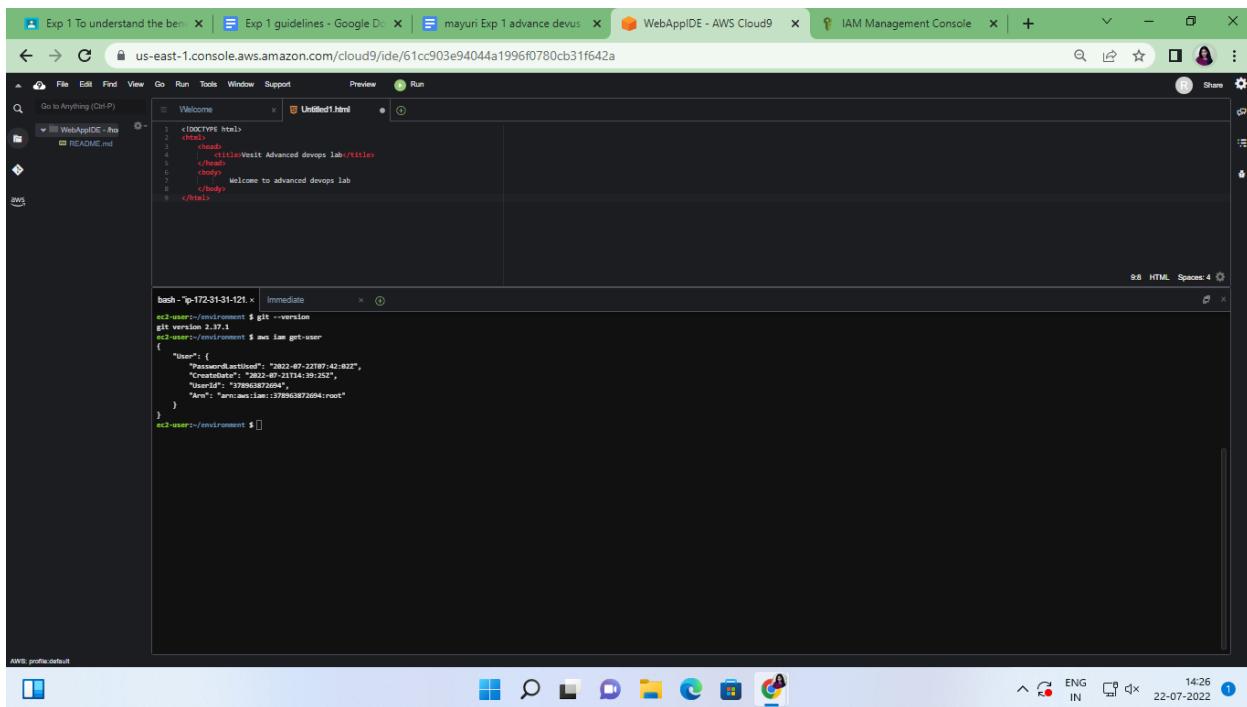
16. If you check at bottom side Cloud9 IDE also giving you and aws CLI for command operations: as we here checked git version, iam user details and so on



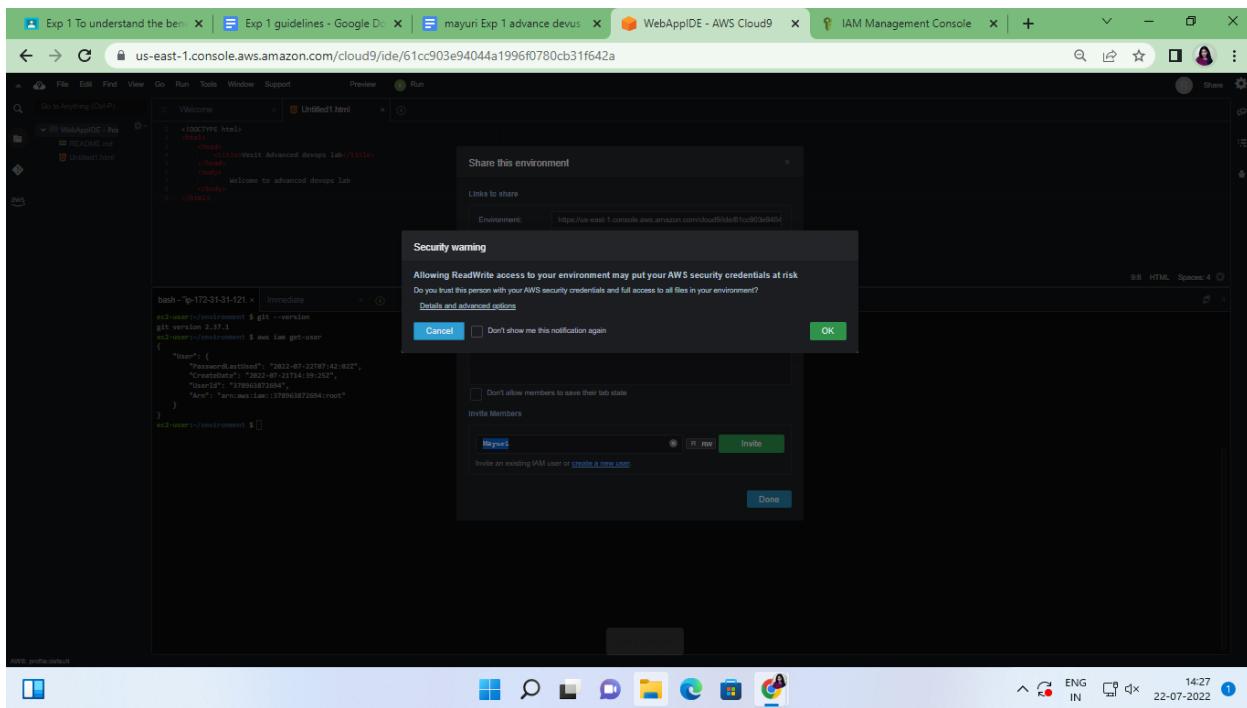
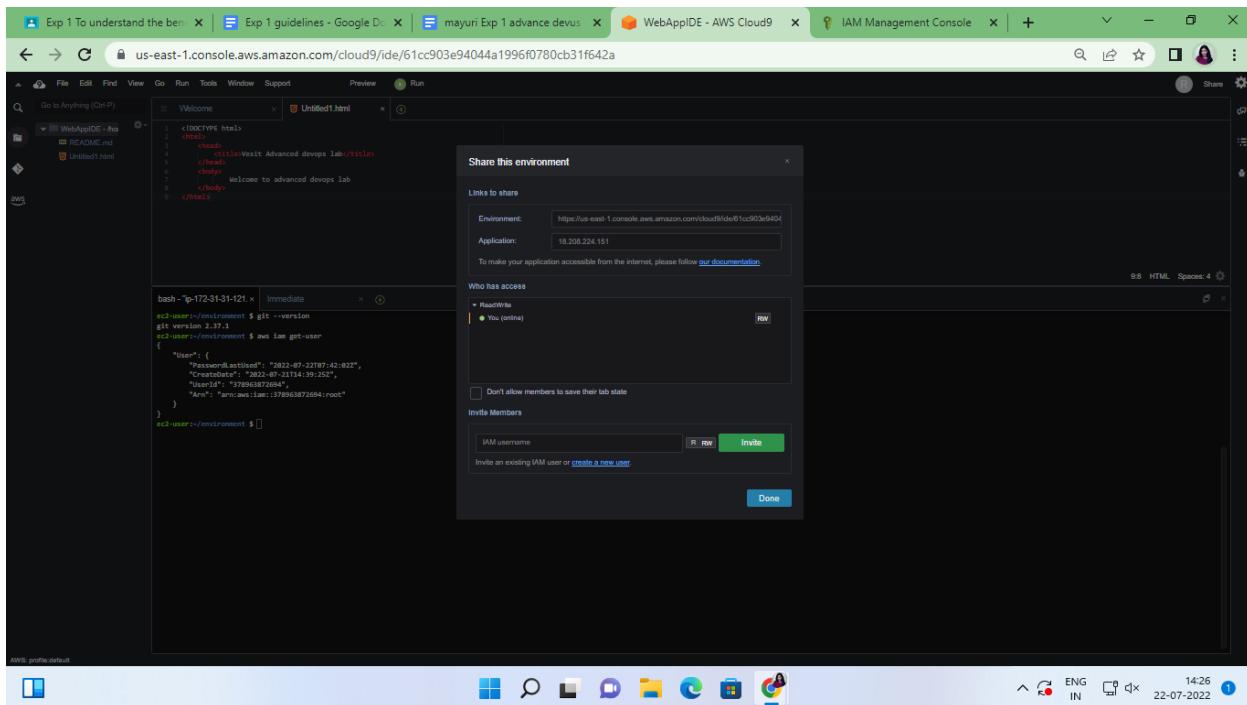
17. Now we will setup collaborative environment Click on File you can create new file or choose from template, here m opting html file to collaborate

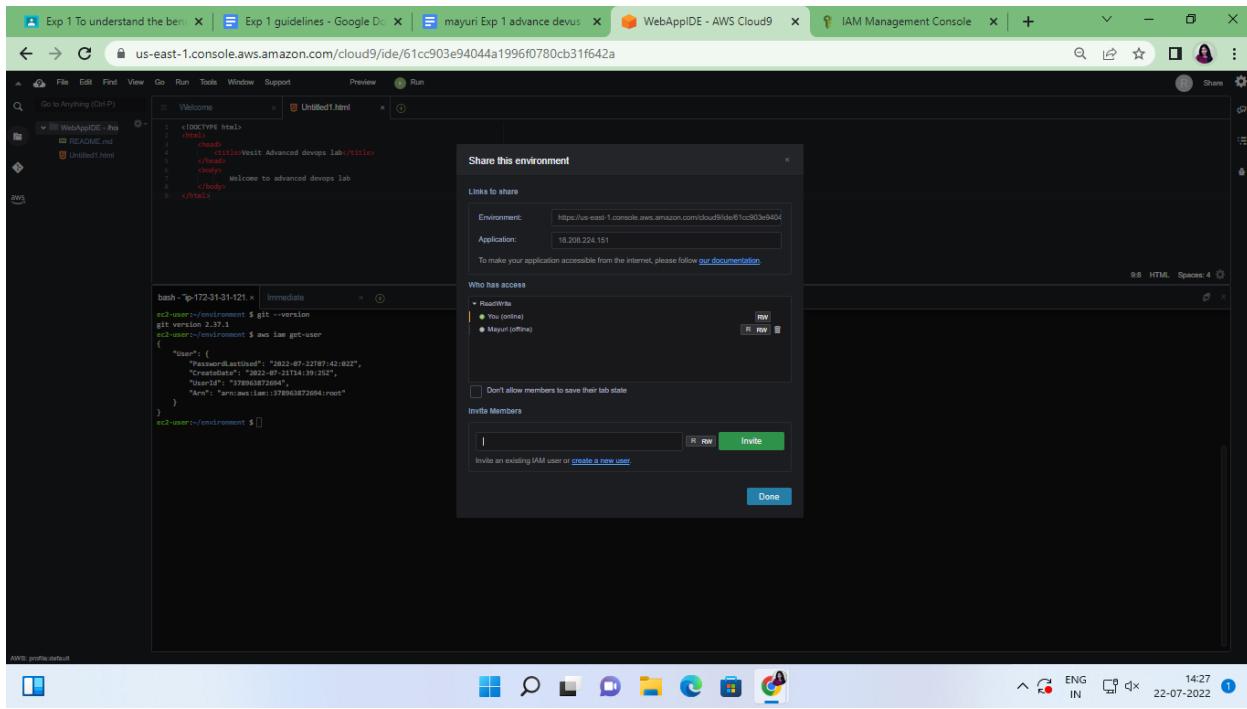


18. Edit html file and save it

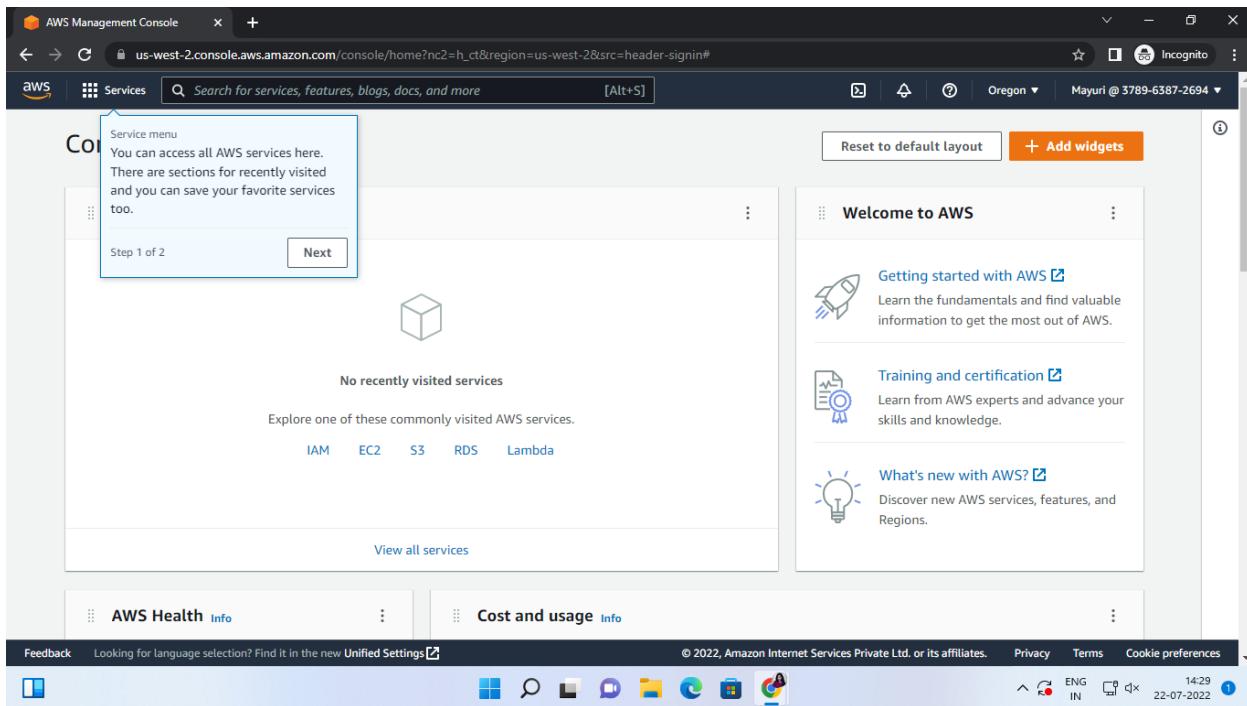


19. Now in order to share this file to collaborate with other members of your team click on Share option on Right Pane and username which you created in IAM before into Invite members and enable permission as RW (Read and Write) and click on Done. Click OK for Security warning.





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



Select region mumbai

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

The screenshot shows the AWS Cloud9 interface with the sidebar expanded. Under 'Shared with you', there is one environment named 'WebAppIDE'. It is listed as an EC2 type environment with read-write permissions, no description, and owned by 'am:awsiam:378965872694:root'. There is a 'Create environment' button at the top right of the list.

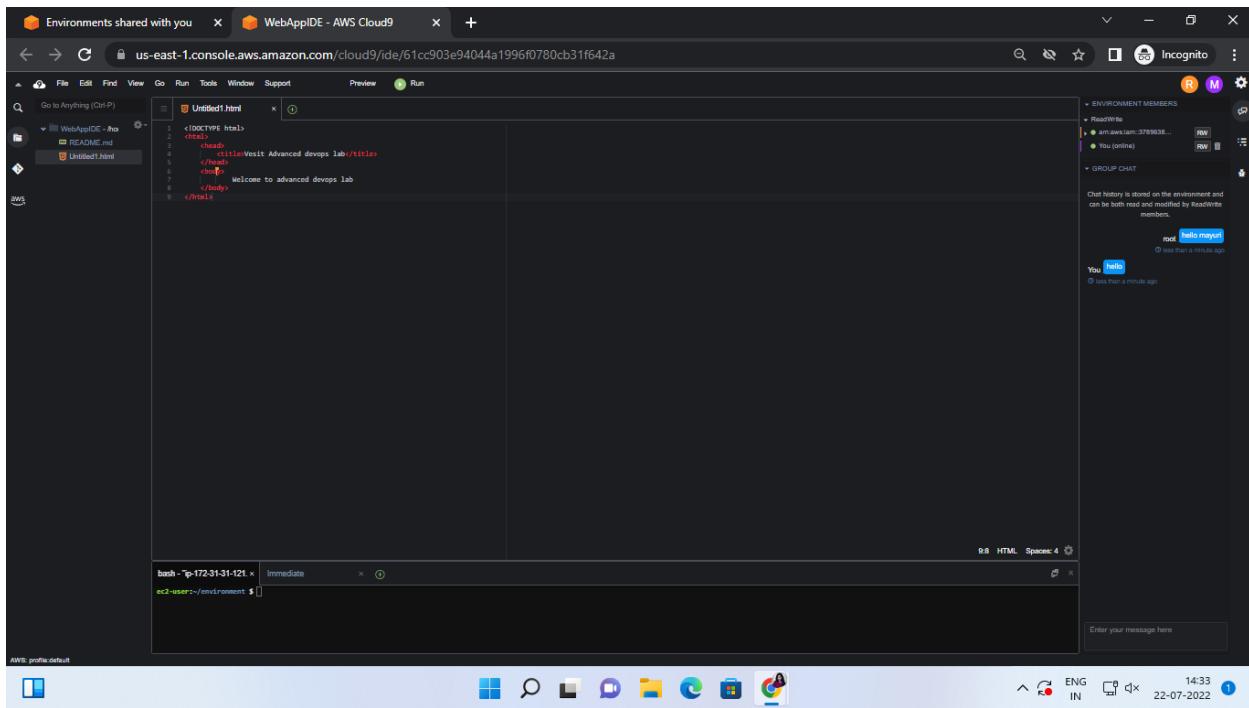
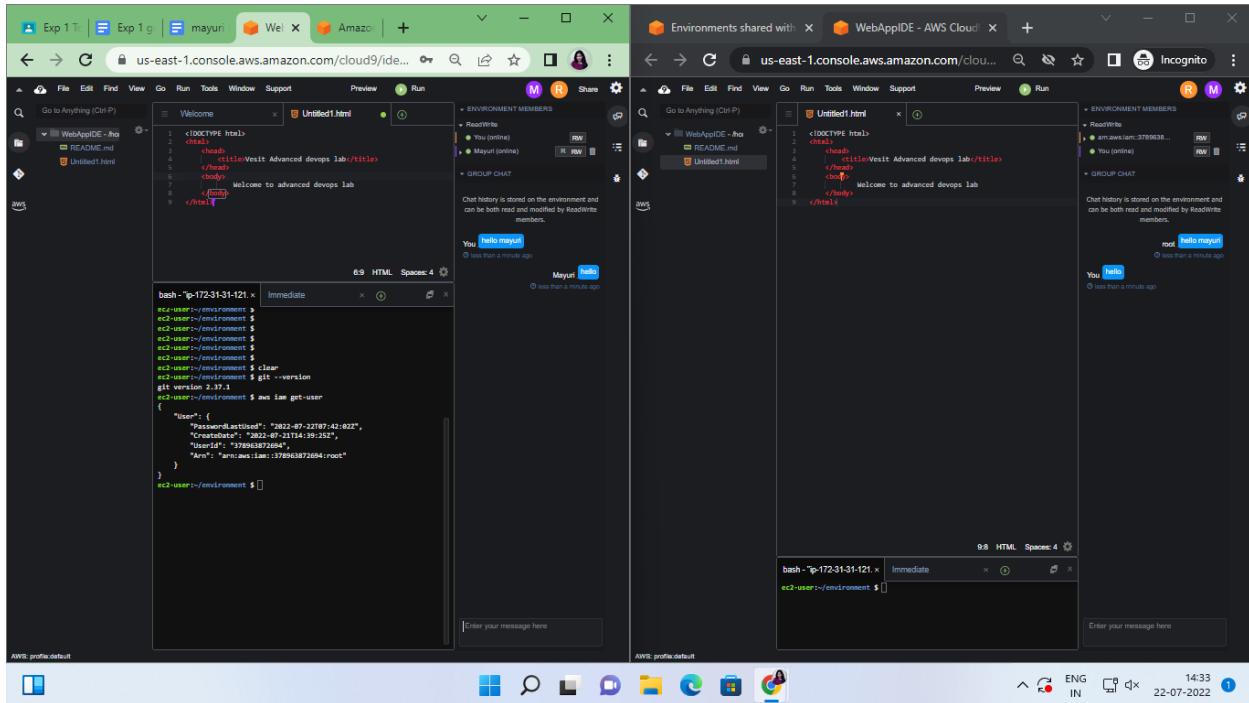
This screenshot displays two separate Cloud9 IDE windows. Both windows show the same code editor with the file 'Untitled1.html' containing the following HTML:

```
<!DOCTYPE html>
<html>
<head>
<title>Visit Advanced devops lab</title>
</head>
<body>
    welcome to advanced devops lab
</body>
</html>
```

Below the code editor, both windows show a terminal window with the following command-line session:

```
bash -> ip-172-31-121 ~
git version 2.37.1
aws iam get-user
{
    "User": {
        "PasswordLastUsed": "2022-07-22T07:42:02Z",
        "AccessKeyId": "AKIAJ1A14X95Z3L",
        "UserId": "378965872694",
        "Arn": "arn:aws:iam::378965872694:root"
    }
}
aws iam
```

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



Conclusion: We have successfully performed a collaboration demonstration on cloud9.