

# **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.

## **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**

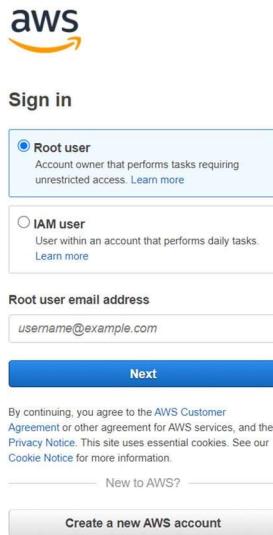
AWS Cloud9 comes with a terminal that includes sudo privileges to the managed Amazon EC2 instance that is hosting your development environment and a preauthenticated AWS Command Line Interface. This makes it easy for you to quickly run commands and directly access AWS services

## **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:**

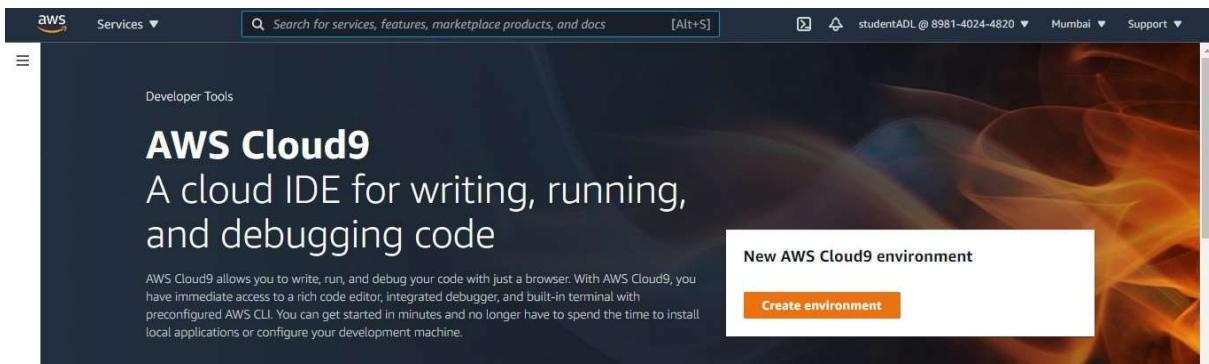
- 1. Login with your AWS account.**



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

The image shows the AWS home page with the "Developer Tools" section highlighted. On the left sidebar, there are sections for "Favorites" (empty), "Recently visited" (Billing, Console Home), and "Developer Tools" (CodeStar, CodeCommit, CodeArtifact, CodeBuild, CodeDeploy, CodePipeline, Cloud9, CloudShell, X-Ray, AWS FIS). The main content area is titled "All services" and lists various AWS services in a grid. The "Developer Tools" section is visible on the right side of the grid.

## 3. Click on Create Environment:



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

This screenshot shows the 'Name environment' step of a three-step process. On the left, a sidebar lists 'Step 1 Name environment', 'Step 2 Configure settings', and 'Step 3 Review'. The main area is titled 'Name environment' and contains a 'Environment name and description' section. Under 'Name', it says 'The name needs to be unique per user. You can update it at any time in your environment settings.' A text input field contains 'WebAppIDE'. Below it, a note says 'Limit: 60 characters'. Under 'Description - Optional', it says 'This will appear on your environment's card in your dashboard. You can update it at any time in your environment settings.' A text area placeholder is 'Write a short description for your environment'. Below it, a note says 'Limit: 200 characters'. At the bottom right are 'Cancel' and 'Next step' buttons.

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

The screenshot shows the AWS Cloud9 'Configure settings' step. On the left, a sidebar lists 'Step 1 Name environment', 'Step 2 Configure settings' (which is selected), and 'Step 3 Review'. The main content area is titled 'Configure settings' and contains a 'Environment settings' section. Under 'Environment type', 'Create a new EC2 instance for environment (direct access)' is selected. Under 'Instance type', 't2.micro (1 GiB RAM + 1 vCPU)' is selected. In the 'Platform' section, 'Amazon Linux 2 (recommended)' is selected. A 'Cost-saving setting' dropdown is set to 'After 30 minutes (default)'. An 'IAM role' section indicates 'AWS Cloud9 creates a service-linked role for you' with a link to 'Learn more'. Below that is a 'Network settings (advanced)' section with a plus sign icon. At the bottom, there's a note about adding tags, a 'Cancel' button, a 'Previous step' button, and an orange 'Next step' button.

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

The screenshot shows the AWS Cloud9 'Create environment' review step. On the left, a sidebar lists 'Step 1 Name environment', 'Step 2 Configure settings' (which is selected), and 'Step 3 Review'. The main area is titled 'Review' and contains a section titled 'Environment name and settings'. It displays the following configuration details:

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

A callout box provides best practices for using the 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, there are 'Cancel', 'Previous step', and 'Create environment' buttons.



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.

**Users (4) Info**  
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

User name	Path	Groups	Last activity	MFA	Password a...

## 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\*** studentADL [+ 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](#)

## 9. Click on Create group

Set permissions

Add user to group

Create group Refresh

Group	Attached policies
<input type="checkbox"/> TEIT	Billing and 1 more

Showing 1 result

Cancel Previous Next: Tags

## 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		Q Search	Showing 668 results	
	Policy name	Type	Used as	Description
<input type="checkbox"/>	AdministratorAccess	Job function	Permissions policy (2)	Provides full access to AWS services and resources.
<input type="checkbox"/>	AdministratorAccess-A...	AWS managed	None	Grants account administrative permissions while explicitly a...
<input type="checkbox"/>	AdministratorAccess-A...	AWS managed	None	Grants account administrative permissions. Explicitly allow...
<input type="checkbox"/>	AlexaForBusinessDevice...	AWS managed	None	Provide device setup access to AlexaForBusiness services
<input type="checkbox"/>	AlexaForBusinessFullAccess...	AWS managed	None	Grants full access to AlexaForBusiness resources and acc...
<input type="checkbox"/>	AlexaForBusinessGateway...	AWS managed	None	Provide gateway execution access to AlexaForBusiness se...

Cancel Create group

## 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.

AWS Services ▾ Search for services, features, marketplace products, and docs [Alt+S] Student\_demo ▾ Global ▾ Support ▾

### 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	studentADL
AWS access type	AWS Management Console access - with a password
Console password type	Custom
Require password reset	Yes
Permissions boundary	Permissions boundary is not set

#### Permissions summary

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

Type	Name
Group	WebAppADLGroup

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

AWS Services ▾ Search for services, features, marketplace products, and docs [Alt+S] Student\_demo ▾ Global ▾ Support ▾

### Add user to group

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		Showing 2 results
Group ▾	Attached policies	
<input checked="" type="checkbox"/> WebAppADLGroup	None	
<input type="checkbox"/> TEIT	Billing and 1 more	

[Set permissions boundary](#)

[Cancel](#) [Previous](#) [Next: Tags](#)

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

The screenshot shows the AWS IAM User groups page. On the left, there's a navigation sidebar with sections for Dashboard, Access management (User groups, Users, Roles, Policies, Identity providers, Account settings), and Access reports (Access analyzer, Archive rules, Analyzers, Settings, Credential report, Organization activity, Service control policies (SCPs)). The main content area has a title 'User groups (3) Info' with a description: 'A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.' It includes a search bar, a toolbar with 'Create group' (blue button), and a table header with columns: Group name, Users, Permissions, Creation time. A single row is shown: 'WebAppADLGroup' (with a checkbox icon), 'Defined' (green checkmark), and '15 hours ago'.

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

The screenshot shows the 'WebAppADLGroup' details page under the IAM service. The top navigation bar includes 'Services', a search bar, and account information ('Student\_demo', 'Global', 'Support'). The left sidebar is identical to the one in the previous screenshot. The main content has a 'Summary' section with fields: User group name (WebAppADLGroup), Creation time (July 22, 2021, 19:24 (UTC+05:30)), and ARN (arn:aws:iam:898140244820:group/WebAppADLGroup). Below this is a tab navigation with 'Users' (grayed out), 'Permissions' (orange, selected), and 'Access advisor'. The 'Permissions' tab shows a section for 'Permissions policies (0) Info' with a note: 'You can attach up to 10 managed policies.' It includes a search bar, a toolbar with 'Simulate' and 'Remove' buttons, and a table header with columns: Policy Name, Type, Attached entities. The table is currently empty.

**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 1/668) [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

< 1 2 3 4 5 6 7 ... 34 > | [?](#)

Policy Name	Type	Attached entities
AWSQuicksightAthenaAccess	AWS managed	0
AWSCloudMapRegisterInstanceAccess	AWS managed	0
AWSMarketplaceImageBuildFullAccess	AWS managed	0
AWSCodeCommitPowerUser	AWS managed	0
AWSCodeCommitFullAccess	AWS managed	0
IAMSelfManageServiceSpecificCredentials	AWS managed	0
<b>AWSCloud9EnvironmentMember</b>	AWS managed	0

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

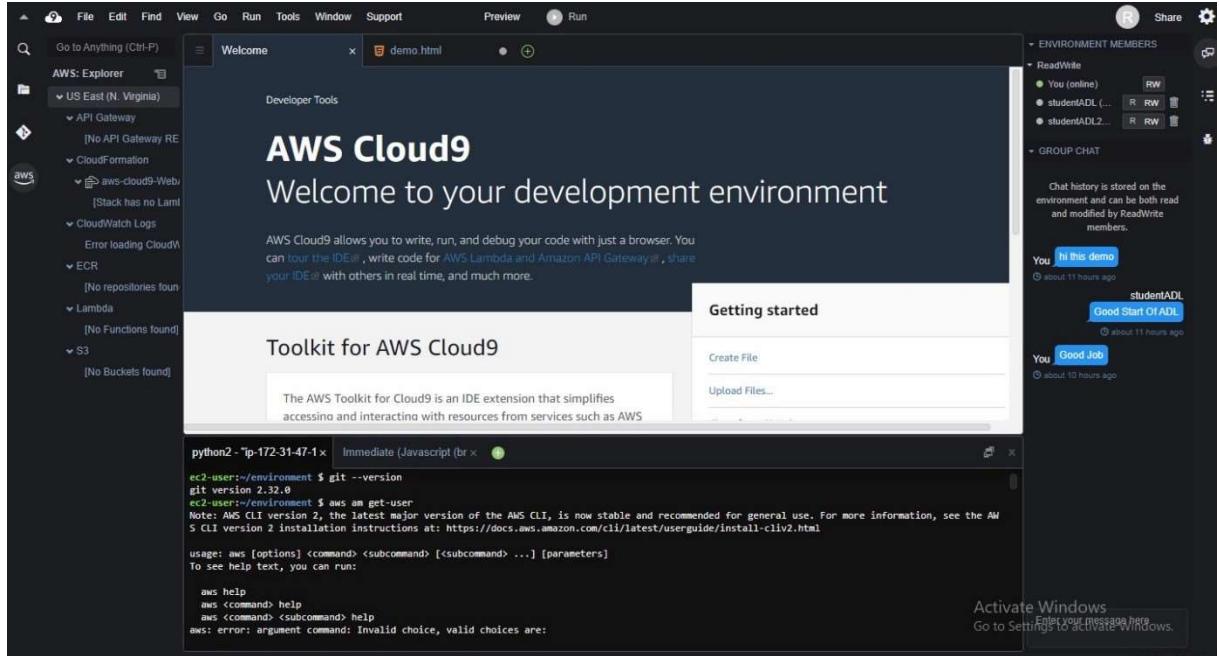
The AWS Explorer provides access to the AWS services that you can work with when using the Toolkit. To see the AWS Explorer, choose the AWS icon in the Activity bar.

```

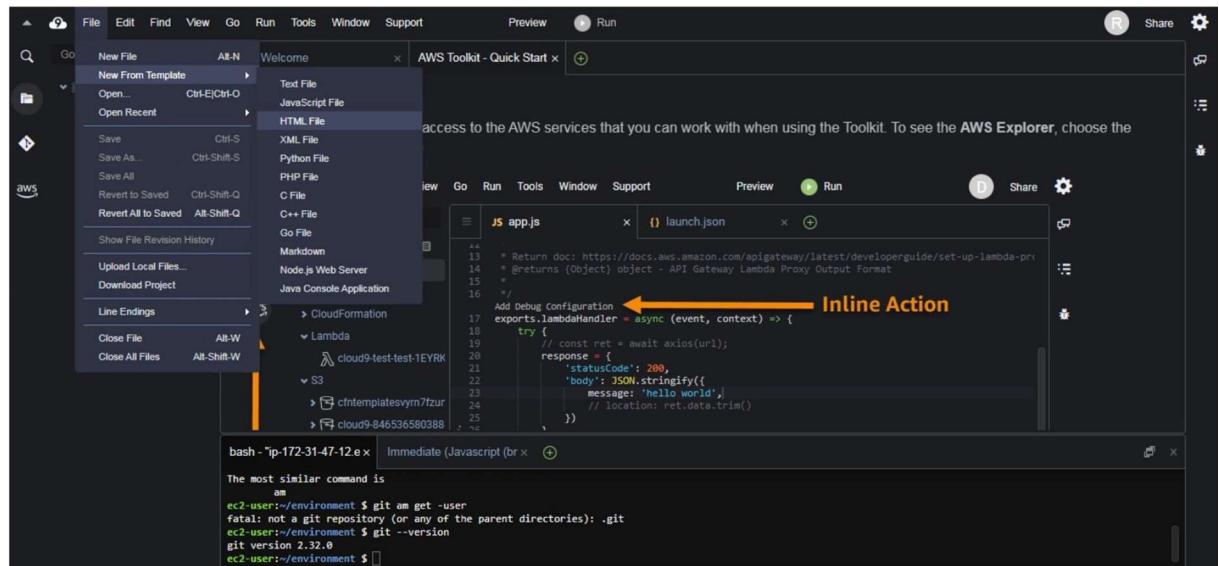
    1 // Return doc: https://docs.aws.amazon.com/apigateway/latest/developerguide/set-up-lambda-proxy-integrations.html#api-gateway-lambda-proxy-output-format
    2
    3 // Returns a JSON object
    4
    5
    6
    7 exports.handler = async (event, context) => {
    8     try {
    9         // const result = await axios(url);
   10         response = {
   11             statusCode: 200,
   12             body: JSON.stringify({
   13                 message: 'Hello world',
   14                 // location: ret.data.trim()
   15             })
   16         }
   17     } catch (err) {
   18         console.log(err);
   19         return err;
   20     }
   21     return response;
   22 }
   23
   24
   25
   26
   27
   28
   29
   30
   31
   32
   33
   34
  
```

23:40 JavaScript Spaces: 4 Current Credentials

## 16. If you check at bottom side Cloud9 IDE also giving you and AWS CLI for command operations: as we here checked git version, I am 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.**

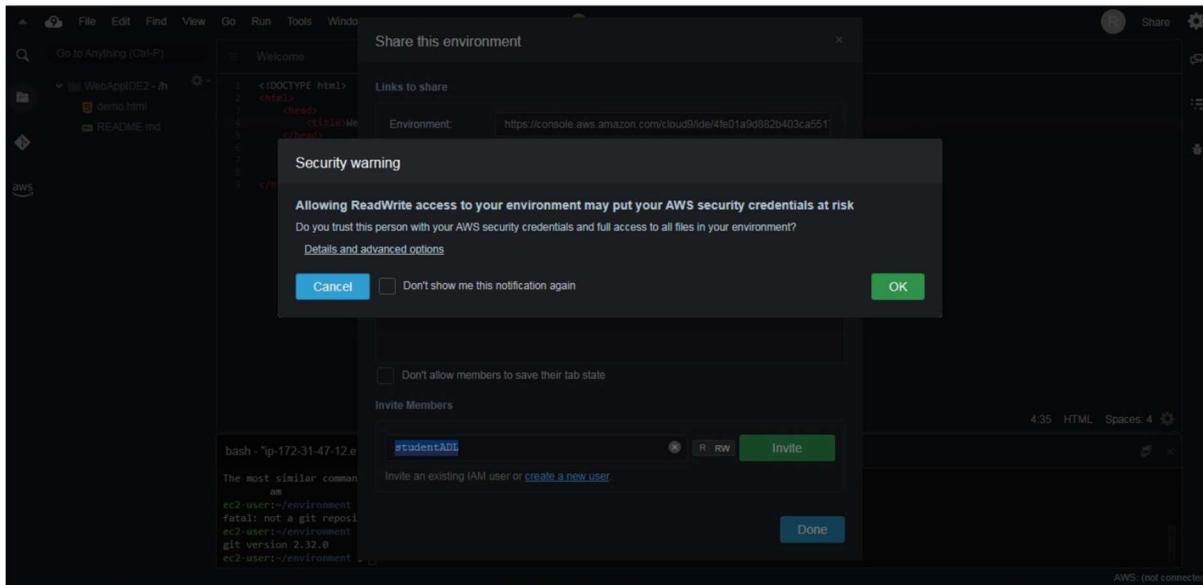


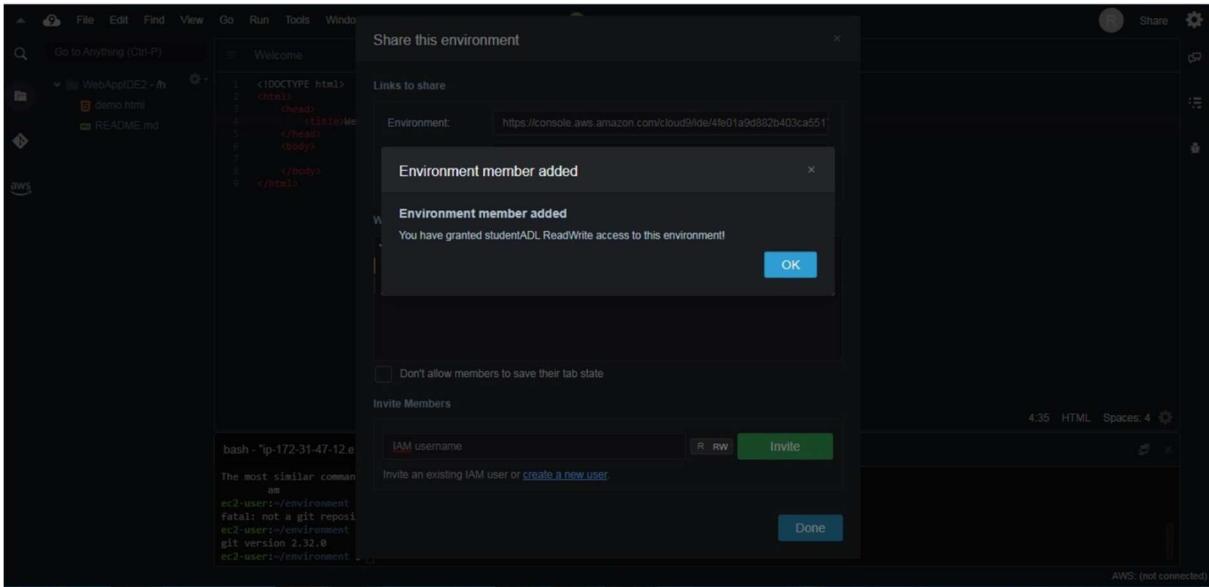
**18. Edit html file and save it**

```
<!DOCTYPE html>
<html>
  <head>
    <title>Welcome to AWS demo</title>
  </head>
  <body>
  </body>
</html>
```

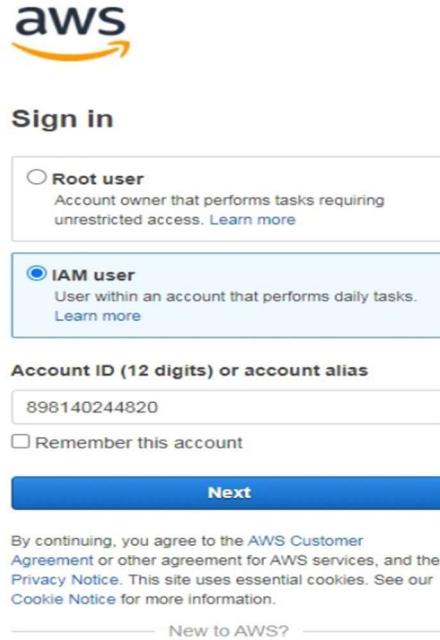
```
bash - ip-172-31-47-12.e x Immediate (Javascript (br x +)
The most similar command is
am
ec2-user:/environment $ git am get -user
fatal: not a git repository (or any of the parent directories): .git
ec2-user:/environment $ git --version
git version 2.32.0
ec2-user:/environment $
```

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.

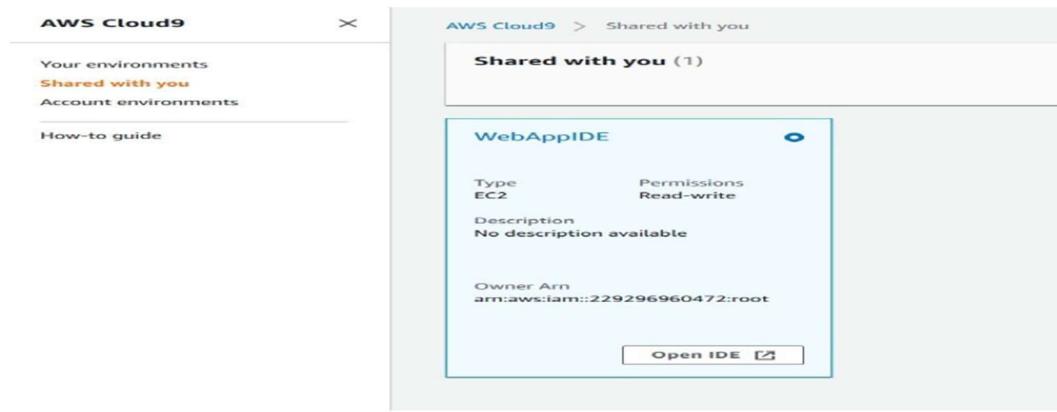




**20. Now Open your Browsers Incognito Window and login with IAM user which you configured before.**



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



22. Click on Open IDE you will same interface as your other member have to collaborate in real time.

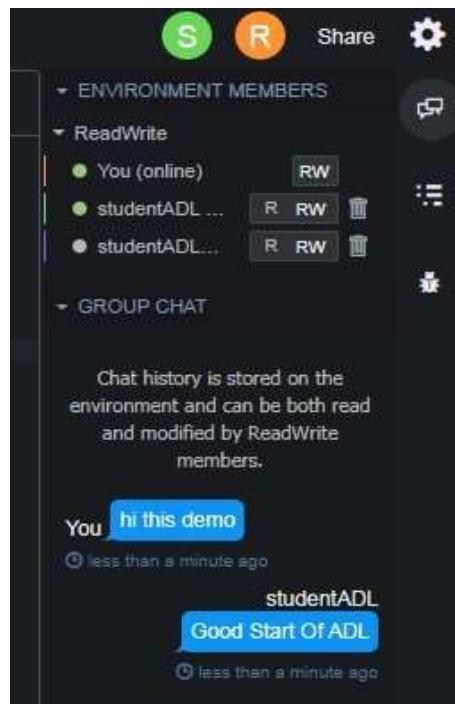
The screenshot shows the AWS Cloud9 IDE. At the top, there's a menu bar with File, Edit, Find, View, Go, Run, Tools, Window, Support, Preview, and Run. On the right, there are icons for R (red), S (green), and settings. The status bar says 'You (studentADL)'. The left sidebar shows a file tree with 'demo.html' selected. The main area has a code editor with the following HTML code:

```
<!DOCTYPE html>
<html>
<head>
    <title>Welcome to BVCOE College, kharghar </title>
</head>
<body>
</body>
</html>
```

Below the code editor is a terminal window titled 'bash - ip-172-31-47-12.e' with the command 'ec2-user:~/environment \$'. The status bar at the bottom of the terminal says '9:0 HTML Spaces: 4'.

23. Also, you all within team can do group chats as shown below:

**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:** After Successful setup of AWS account, Cloud9IDE environment service from the AWS installed successfully. The AWS Console web Application provided by Amazon Web Services helped me in viewing and managing a select set of resources to support incident response while on-the-go. I can use the Console Mobile Application to monitor resources through a dedicated dashboard and view configuration details, metrics, and alarms for select AWS services. AWS Cloud9 which is a cloud-based

integrated development environment (IDE) that helped me to write, run, and debug my code with just a browser. I do not have to wait for an environment or a device to be code-ready. By providing a code-ready climate, AWS Cloud9 brings collaboration to Dev Ops, improves productivity, and helps with better time management.