

Name: Manav Jawrani

Roll No.: 19

Subject: Advanced DevOps

Experiment No.: 1

Experiment No. 1

Aim: To understand the benefits of Cloud Infra structure and Setup AWS Cloud 9 IDE, Launch AWS Cloud 9 IDE and Perform Collaboration Demonstration.

Theory:

AWS Cloud 9 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. Cloud 9 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 Cloud 9 IDE is cloud-based, you can work on your projects from your office, home or anywhere using an internet-connected machine. Cloud 9 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 Cloud 9, 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:

1.

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 editors 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.

2.

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 programs together. While collaborating, your team members can see each other in real-time and instantly chat with one another from within the IDE.

3

Build serverless Applications with Ease

AWS Cloud9 makes it easy to write, run and debug serverless applications. It pre-configures the development environment with all SDKs, libraries

and plug-ins needed for serverless development. Cloud9 also provides an environment for locally testing and debugging AWS code directly saving you time and improving the quality of your code.

4. 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.

5. 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, JS, PHP, Ruby etc. This enables you to start writing code for popular applications 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. Search for Cloud9 under the list of services in the search menu.

The screenshot shows the AWS Management Console search results for the query 'cloud9'. The search bar at the top contains 'cloud9'. Below it, the 'Services' section is expanded, showing a list of services including Cloud9, AWS Cloud Map, Lightsail, WorkSpaces, and Account environments. To the right, there is a sidebar with a 'Recent' section containing links to Cloud9, EC2, CloudWatch Metrics, AWS Lambda, and IAM. A 'CloudWatch Metrics' section is also visible on the far right.

3. In the Cloud9 Console, click on Create Environment.

The screenshot shows the AWS Cloud9 home 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. On the right side, there is a callout box titled 'New AWS Cloud9 environment' with a 'Create environment' button. At the bottom left, there is a section titled 'How it works' with a detailed description of how to create a development environment. On the right, there is a 'Getting started' sidebar with links to 'Before you start', 'Create an environment', 'Working with environments', and 'Working with the IDE', each with a corresponding 'min read' time.

4. Proceed after entering a name for your environment.

The screenshot shows the 'Name environment' step of the AWS Cloud9 creation wizard. 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 'Environment name and description'. It contains a 'Name' field with 'WebAppID1' entered, a note about uniqueness, and a 'Description - Optional' field with placeholder text 'Write a short description for your environment'. Both fields have character limits of 60 and 200 respectively. At the bottom are 'Cancel' and 'Next step' buttons. The browser address bar shows 'us-east-1.console.aws.amazon.com/cloud9/home/create'. The status bar at the bottom right indicates 'ENG IN' and the date '26-07-2022'.

5. Proceed with default settings.

The screenshot shows the 'Configure settings' step of the AWS Cloud9 creation wizard. The sidebar now shows 'Step 2 Configure settings' as selected. The main area is titled 'Configure settings' and includes sections for 'Environment type', 'Instance type', 'Platform', 'Cost-saving setting', 'IAM role', and 'Network settings (advanced)'. Under 'Environment type', 'Create a new EC2 instance for environment (Direct access)' is selected. Under 'Instance type', 't2.micro (1 GB RAM + 1 vCPU)' is selected. Under 'Platform', 'Amazon Linux 2 (recommended)' is selected. Under 'Cost-saving setting', 'After 10 minutes (default)' is chosen. Under 'IAM role', it notes that AWS Cloud9 creates a service-linked role. Under 'Network settings (advanced)', there are tags associated with the resource. At the bottom are 'Cancel', 'Previous step', and 'Next step' buttons. The browser address bar shows 'us-east-1.console.aws.amazon.com/cloud9/home/create'. The status bar at the bottom right indicates 'ENG IN' and the date '26-07-2022'.

6. Review the settings and create the environment.

The screenshot shows the 'Create environment' review step in the AWS Cloud9 interface. The 'Name environment' section is filled with 'WebAppIDE'. Under 'Environment type', 'EC2' is selected with 't2.micro' as the instance type. The 'Platform' is set to 'Amazon Linux 2 (recommended)'. A note at the bottom provides best practices for using AWS Cloud9, such as using source control and backup, performing automatic backups, and turning on AWS CloudTrail. The 'Create environment' button is visible at the bottom right.

7. The process of creating your IDE has started. It will take some time and while that happens, we can create our IAM user for sharing our code.

The screenshot shows the AWS Cloud9 IDE creation progress. A large message in the center says 'We are creating your AWS Cloud9 environment. This can take a few minutes.' Below it, a smaller message says 'The environment is being created. You can view its status in the Environment tab of the Cloud9 console.' The 'Getting started' section is visible on the right.

8. Click on the AWS nav logo or open up the AWS console in a new tab and search for IAM. The IAM console opens up.

The screenshot shows the AWS IAM console with the 'Users' list page. The left sidebar has 'Identity and Access Management (IAM)' selected. The main area displays a table with one row, 'No resources to display'. A blue banner at the top says 'Introducing the new Users list experience'.

9. Create a new user.

The screenshot shows the 'Set user details' step of the IAM user creation wizard. It asks for a 'User name' (MJ) and 'Select AWS access type'. Under 'Select AWS credential type', 'Password - AWS Management Console access' is checked. The 'Console password' section shows 'Custom password' selected with a masked password field. At the bottom, there's a note about password reset requirements and a 'Next: Permissions' button.

10. Click on Create group

The screenshot shows the 'Set permissions' step of the 'Create group' wizard. At the top, there are three buttons: 'Add user to group' (selected), 'Copy permissions from existing user', and 'Attach existing policies directly'. Below these buttons is a section titled 'Get started with groups' which contains a message about creating groups and a 'Create group' button.

11. Name the group and some policies.

The screenshot shows the 'Create group' wizard with the 'Select policies' step open. It displays a list of AWS managed policies with checkboxes. The policies listed are:

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 n...
AdministratorAccess-AWSxElasticBeanstalk	AWS managed	None	Grants account administrative permissions. Explicitly allows developers and administrators to g...
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
AlexaForBusinessLifesizeDelegatedAccess...	AWS managed	None	Provide access to Lifesize AVS devices

12. Review the group information.

The screenshot shows the 'Review' step of creating a new IAM user named 'MJ'. The page displays the user's details, including their name, access type (AWS Management Console access - with a password), password type (Custom), and password reset requirements. It also shows the 'Permissions boundary' as 'Permissions boundary is not set'. Below this, the 'Permissions summary' section indicates that the user will be added to the 'Group1' group. There is a table showing the group assignment:

Type	Name
Group	Group1

The 'Tags' section notes that no tags have been added. At the bottom right, there are 'Cancel', 'Previous', and 'Create user' buttons. The browser address bar shows the URL: us-east-1.console.aws.amazon.com/iam/home#/users\$new?step=review&login&userNames=MJ&passwordType=manual&groups=Group1. The top navigation bar includes tabs for IAM Manager, IT-Sem5/Adv, and various Google Docs and Sheets.

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

The screenshot shows the AWS IAM User groups page. On the left, there's a navigation menu with 'User groups' selected under 'Access management'. The main area displays a table titled 'User groups (1)'. The table has columns for 'Group name', 'Users', 'Permissions', and 'Creation time'. One row is shown for 'Group1', which has 1 user, 'Not defined' permissions, and was created 11 minutes ago. There are buttons for 'Create group', 'Delete', and a search bar at the top of the table.

14. Under Permissions, add the AWSCloud9EnvironmentMember Permission.

The screenshot shows the AWS IAM User group details page for 'Group1'. The left sidebar shows 'User groups' selected. The main page has a 'Summary' section with details like the user group name (Group1), creation time (July 26, 2022, 11:34 UTC+05:30), and ARN (arn:aws:iam:698843610752:group/Group1). Below this is a 'Permissions' tab, which is currently active. It shows a table for 'Permissions policies (0)' with a 'Add permissions' button. The table has columns for 'Policy name', 'Type', and 'Description'. A note says 'You can attach up to 10 managed policies.' and there's a search bar at the top of the table.

The screenshot shows the AWS IAM console. On the left, there's a sidebar with 'Identity and Access Management (IAM)' selected. The main content area is titled 'Attach permission policies to Group1'. It shows a table of policies:

Policy name	Type	Description
AWSCloud9EnvironmentMember	AWS managed	Provides the ability to be invited to a Cloud9 environment.
AWSCloud9Administrator	AWS managed	Provides administrator access to a Cloud9 environment.
AWSCloud9User	AWS managed	Provides permission to create a Cloud9 environment.
AWSCloud9SSMInstanceProfile	AWS managed	This policy will be used to attach to an SSM instance.

At the bottom right of the main area are 'Cancel' and 'Add permissions' buttons.

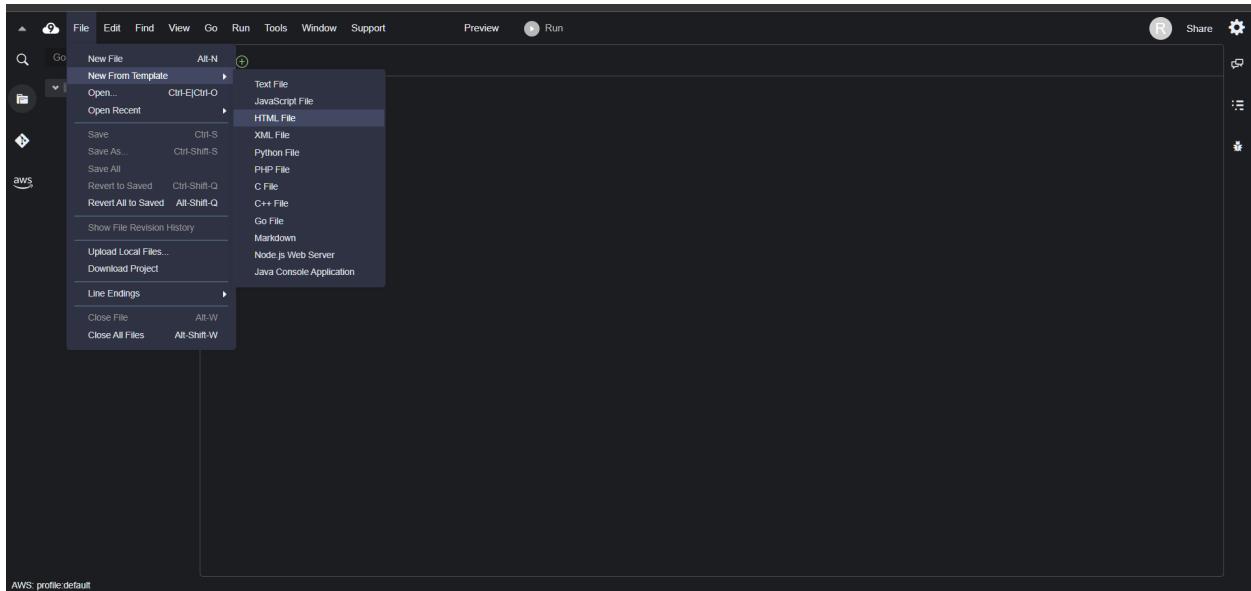
15. Now, once that's done, come back to the Cloud9IDE

16. You can use the command-line here, it also has the AWS CLI built in. You can use the aws get-user command and see your users.

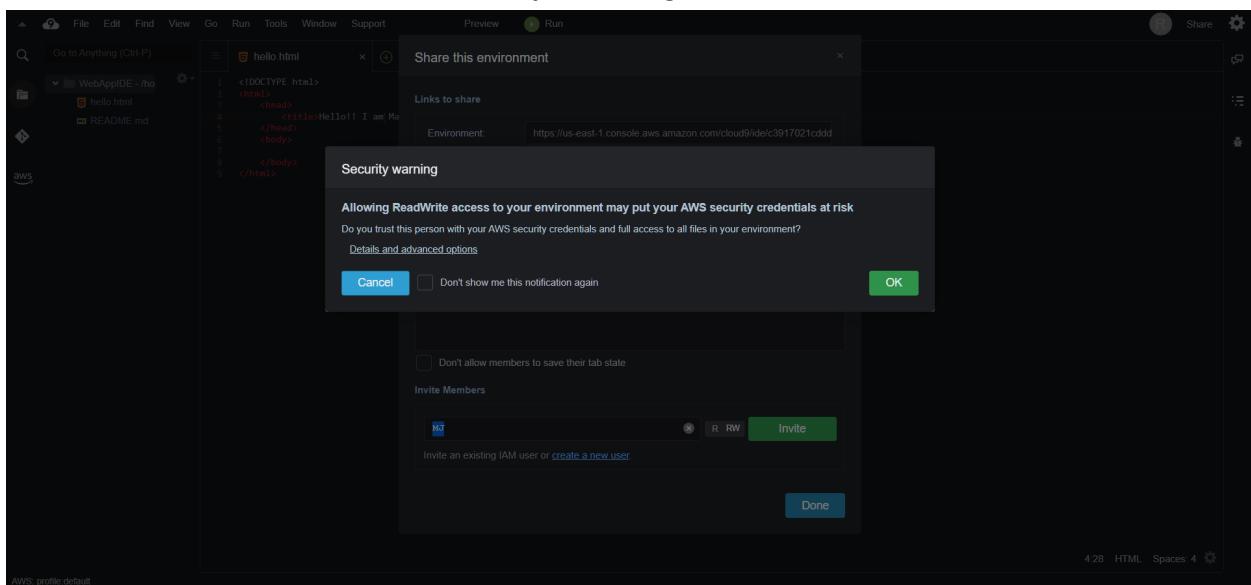
The screenshot shows the AWS Cloud9 IDE interface. At the top, there's a menu bar with File, Edit, Find, View, Go, Run, Tools, Window, Support, Preview, and Run. The main area has a 'Welcome' tab and a 'AWS Cloud9' section with the message 'Welcome to your development environment'. Below that is a 'Toolkit for AWS Cloud9' section. A terminal window is open at the bottom, showing a command-line session:

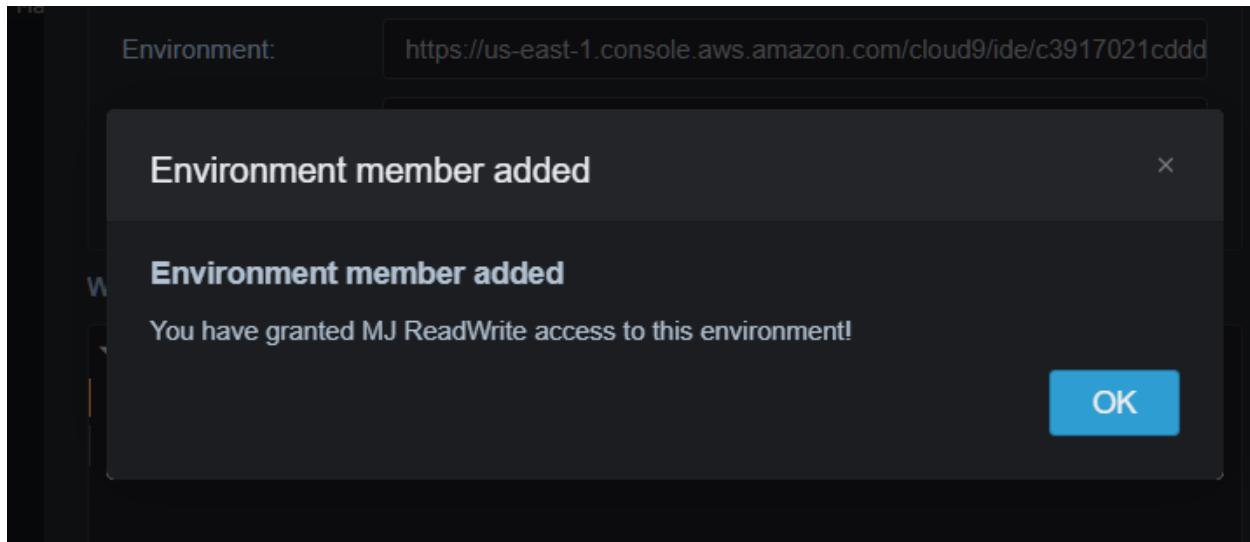
```
python2 -> ip-172-31-90-6 x
ec2-user:~/environment $ git --version
git version 2.37.1
ec2-user:~/environment $ aws iam get-user
{
    "User": {
        "PasswordLastUsed": "2022-07-27T16:51:42Z",
        "CreateDate": "2022-07-04T19:47:51Z",
        "UserId": "698843610752",
        "Arn": "arn:aws:iam::698843610752:root"
    }
}
ec2-user:~/environment $
```

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

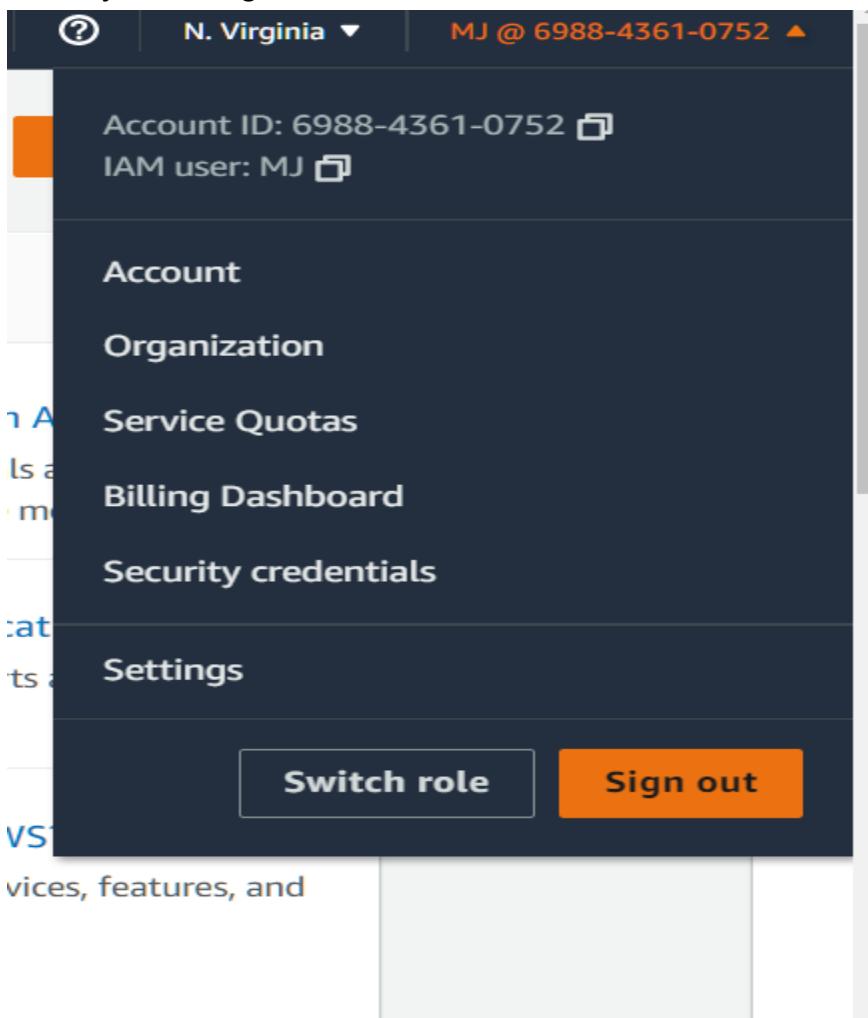


18. Now in order to share this file to collaborate with other members of your team click on Share option on Right Pane and username which you created in IAM before into Invite members and enable permission as RW (Read and Write) and click on Done. Click OK for Security warning.





19. Now Open your Browser's Incognito Window and login with the IAM user which you configured before.



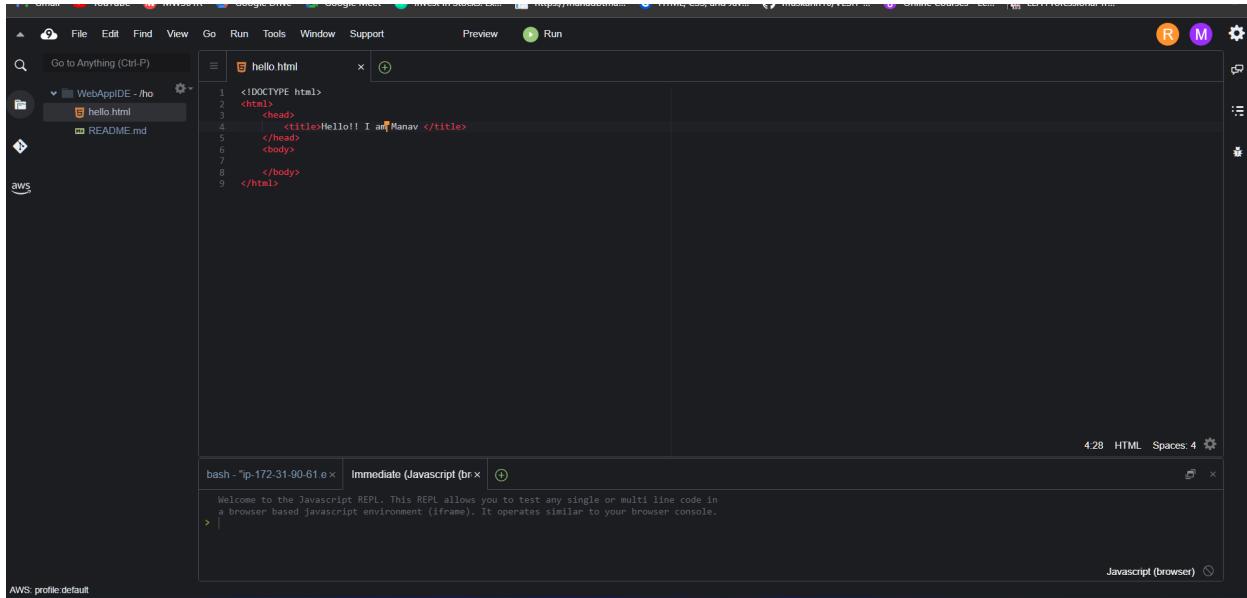
20. After Successful login with IAM user open Cloud9 service from dashboard services and click on shared with you environment to collaborate.

The screenshot shows the AWS Cloud9 interface. At the top, there is a navigation bar with the AWS logo, a 'Services' dropdown, a search bar containing 'Search for services, features, blogs, docs, and more', and a keyboard shortcut '[Alt+S]'. Below the navigation bar, the main title is 'AWS Cloud9' with a close button ('X'). On the left, a sidebar menu includes 'Your environments', 'Shared with you' (which is highlighted in orange), and 'Account environments'. Under 'How-to guide', there is a link to 'Shared environments'. The main content area shows a list titled 'Shared with you (1)'. A single item is listed: 'WebAppIDE'. The details for 'WebAppIDE' are as follows:

Type	Permissions
EC2	Read-write
Description	No description available
Owner Arn	arn:aws:iam::698843610752:root

At the bottom right of the 'WebAppIDE' card, there is a blue button labeled 'Open IDE' with a small icon.

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

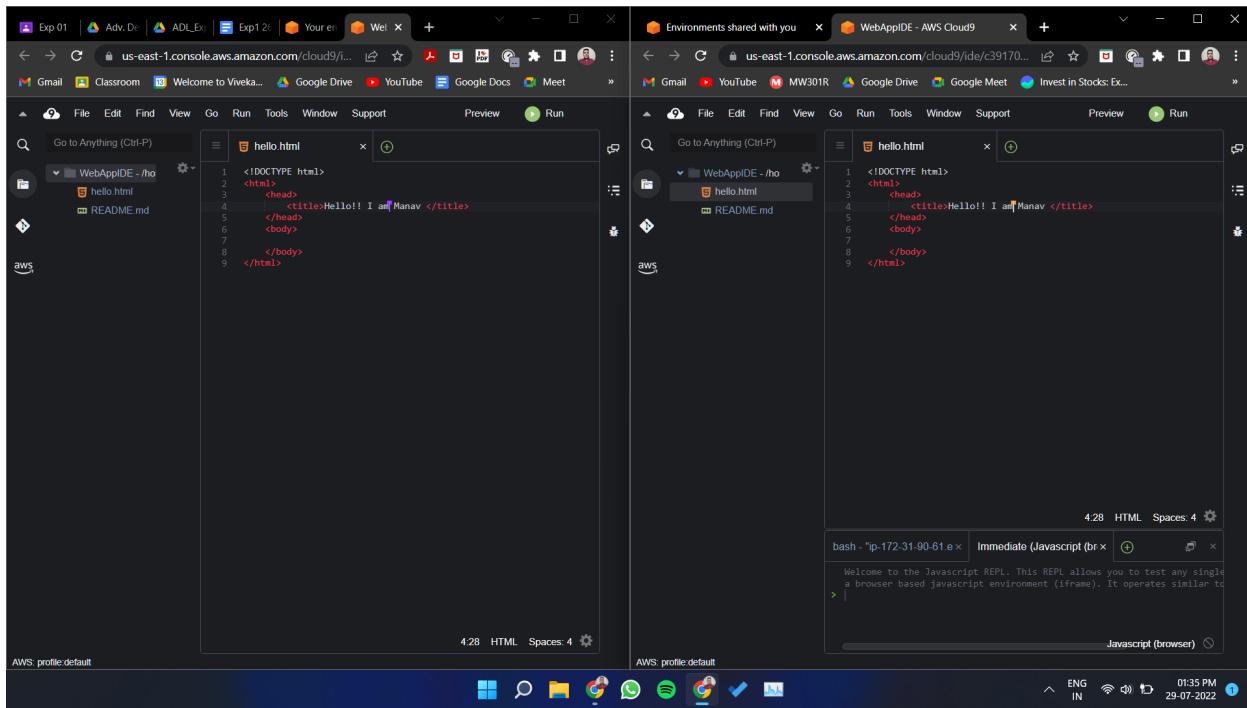


The screenshot shows a single instance of a web-based IDE. On the left, there's a sidebar with a file tree showing 'WebAppIDE - /ho' containing 'hello.html' and 'README.md'. The main editor area displays the content of 'hello.html':

```
<!DOCTYPE html>
<html>
<head>
<title>Hello!! I am Manav </title>
</head>
<body>
</body>
</html>
```

Below the editor is a terminal window titled 'bash - "ip-172-31-90-61.e...' with the command 'ls' entered. To the right of the terminal is an 'Immediate (Javascript)' REPL window with the message: 'Welcome to the Javascript REPL. This REPL allows you to test any single or multi line code in a browser based javascript environment (iframe). It operates similar to your browser console.' At the bottom right, there's a note 'Javascript (browser)'. The status bar at the bottom indicates 'AWS profile default'.

23. Both users now should be able to see the same thing.

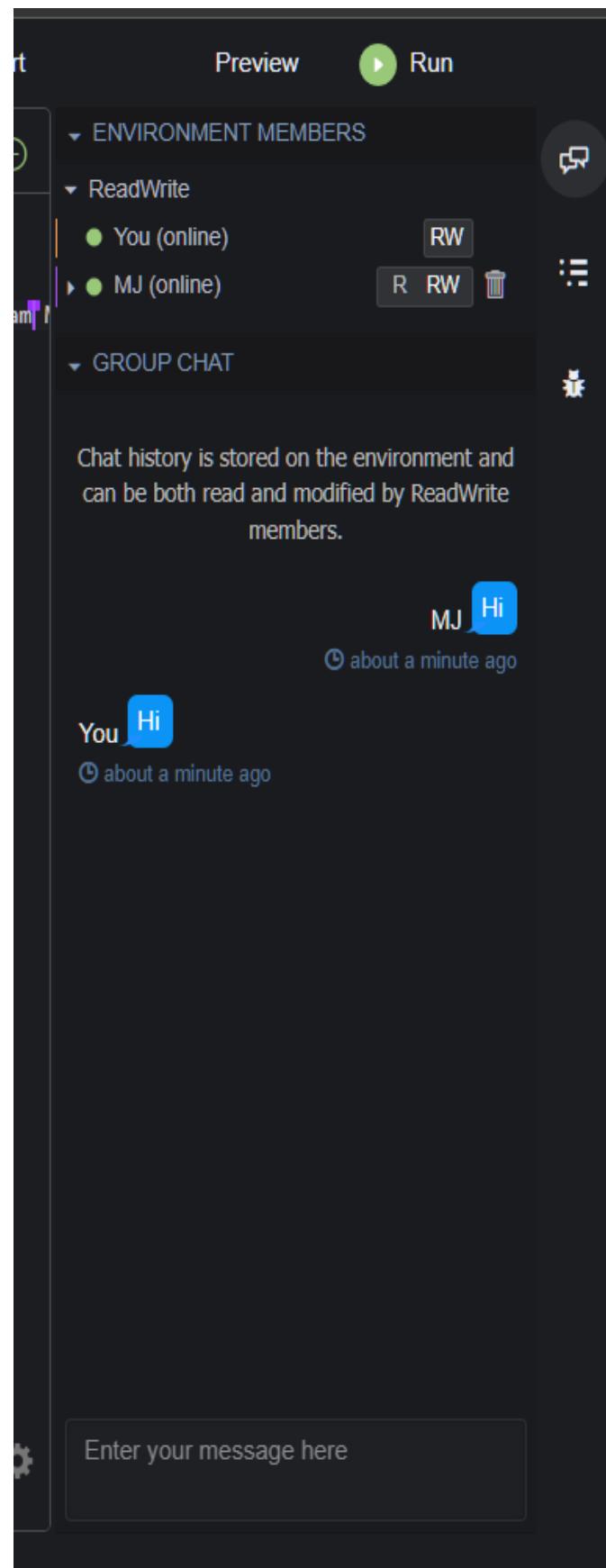
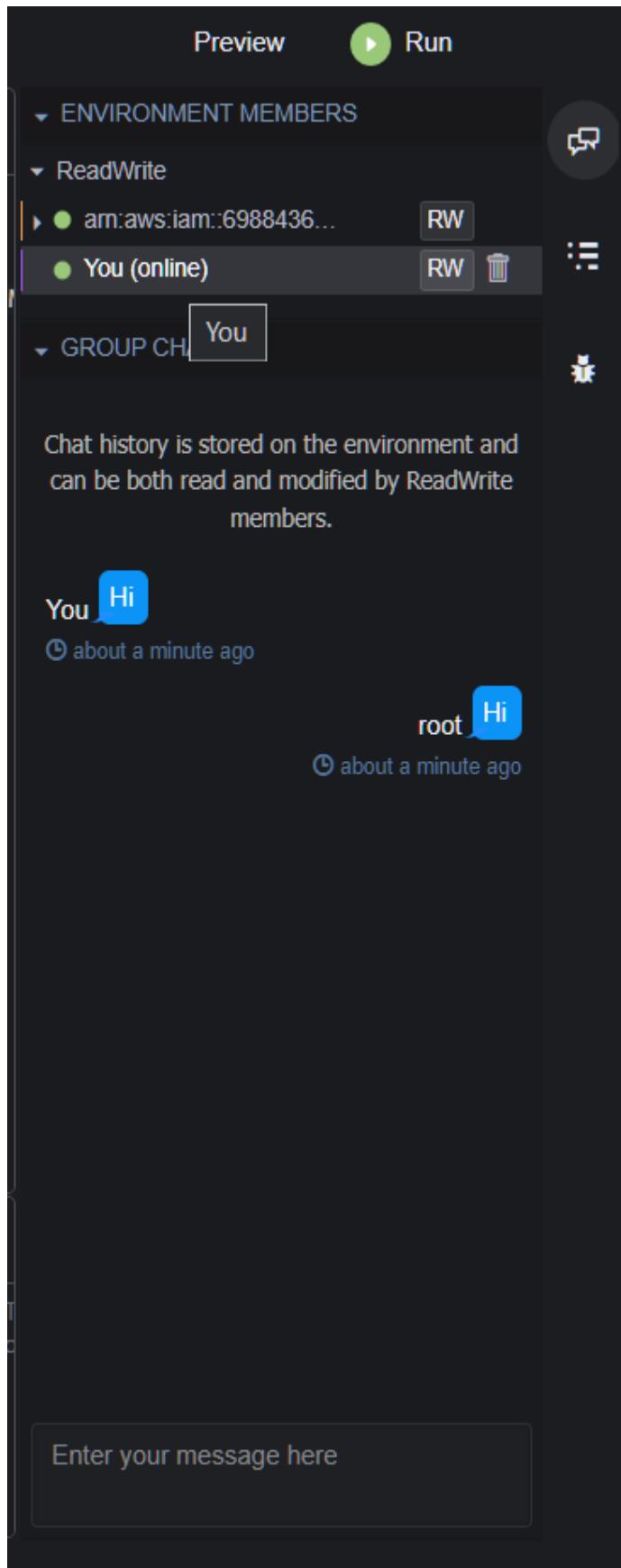


This screenshot shows two separate instances of the web-based IDE running side-by-side. Both instances have identical interfaces, including the sidebar with 'WebAppIDE - /ho' and the 'hello.html' file content:

```
<!DOCTYPE html>
<html>
<head>
<title>Hello!! I am Manav </title>
</head>
<body>
</body>
</html>
```

The terminal windows in both instances show the command 'ls' and the same Javascript REPL message. The status bars at the bottom of both windows indicate 'AWS profile default'.

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



Conclusion:

From this experiment, we got to explore the AWS and all its services. We came across the most used service Cloud9. Cloud9 is used to create an IDE and code in a collaborative environment, creating and managing IAM users, creating user groups, setting permissions. Also got to know the benefits of cloud computing and scope of growth in cloud.