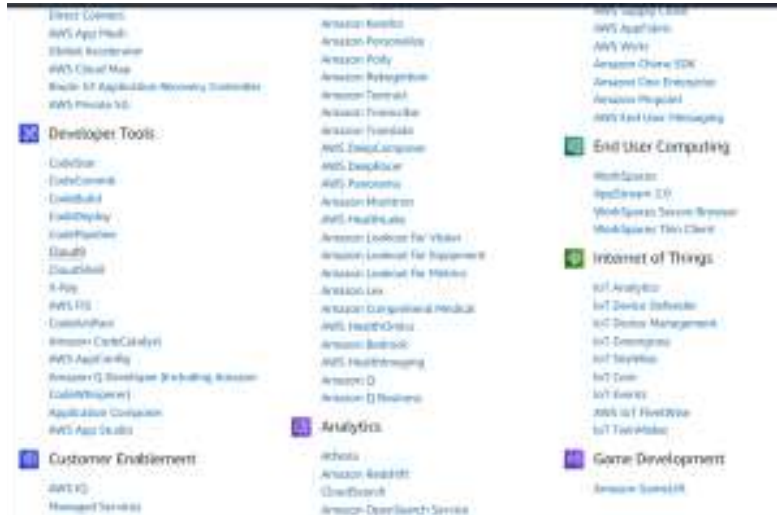


## Experiment No: 1(B)

## Step 1: Set up Cloud9 environment.

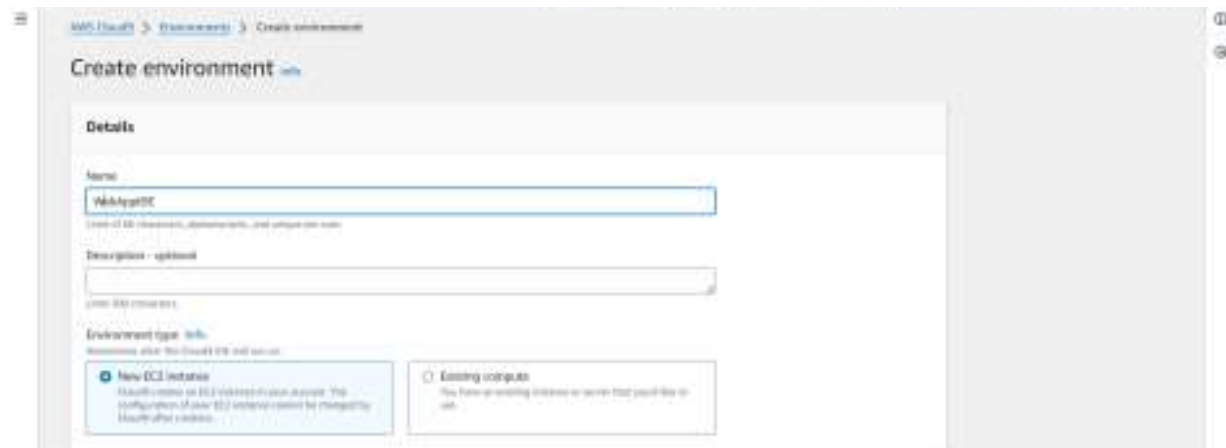
- 1) Go to Cloud9 services under developers tool in All services



- 2) Click on create environment



3) Give the name to your Environment ,keeping the other settings as default



The screenshot shows the 'Create environment' page in the AWS IAM console. The 'Name' field is filled with 'WebAppDEV'. The 'Description' field is empty. Under 'Environment type', 'New EC2 instance' is selected.

**Details**

Name:

Description:

Environment type: ☒ New EC2 instance ☐ Existing codegroup

4) Select the correct platform type as shown below and keep the others details as default



The screenshot shows the 'New EC2 instance' page in the AWS IAM console. The 'Instance type' section shows 't2.micro (1 GB RAM + 1 vCPU)' selected. The 'Platform' dropdown is set to 'Amazon Linux 2017'.

**New EC2 instance**

Instance type: ☒ t2.micro (1 GB RAM + 1 vCPU) ☐ t2.medium (2 GB RAM + 2 vCPU) ☐ m3.xlarge (24 GB RAM + 2 vCPU)

Platform:

**Connections**

View your AWS IAM connections

**AWS Systems Manager (SSM)**  
Automates environment for VMs without requiring inbound connections

**Secure Shell (SSH)**  
Automates environment for VMs via SSH, opens inbound ports

» **VM settings** [info](#)

**Tags - optional** [info](#)

A tag is a key-value pair you can assign to an AWS resource. AWS tags enable you to track and manage your resources. You can use tags to search and filter your resources in AWS, your AWS CLI, and your AWS SDKs.

**The following IAM resources will be created in your account:**

- `AWSCloudWatchRole`** - AWS CloudWatch creates a service-linked role for you. This allows AWS CloudWatch to call other AWS services on your behalf. You can delete this role from the AWS IAM console once you no longer have any AWS CloudWatch environments. [Learn more](#)
- `AWSCloudSSMAccessRole` and `AWSCloudSSMInstanceProfile`** - A service role and an instance profile are automatically created if CloudWatch accesses the EC2 instance through AWS Systems Manager. If your environments no longer require EC2 instances that back incoming traffic, you can delete these roles using the AWS IAM console. [Learn more](#)

Cancel

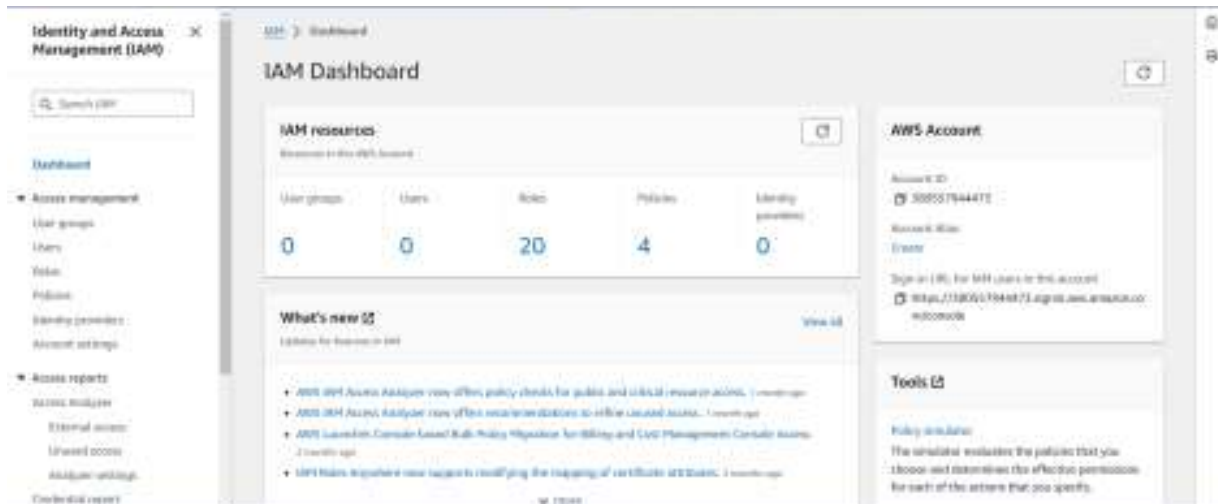
Create

The screenshot shows the AWS Cloud9 console interface. On the left is a sidebar with navigation options: 'My environments', 'Shared with me', 'All account environments', and 'Documentation'. The main area displays the 'Environments' section, which includes a search bar, a 'Create environment' button, and a table of existing environments. The table has columns for Name, Cloud9 IDE, Environment type, Connection, Framework, and Owner. One environment, 'WebApp01', is listed with the following details: Cloud9 IDE is 'User', Environment type is 'EC2 instance', Connection is 'Secure Shell (SSH)', Framework is 'None', and Owner is 'arn:aws:cloud9:us-east-1:123456789012:user:example'.

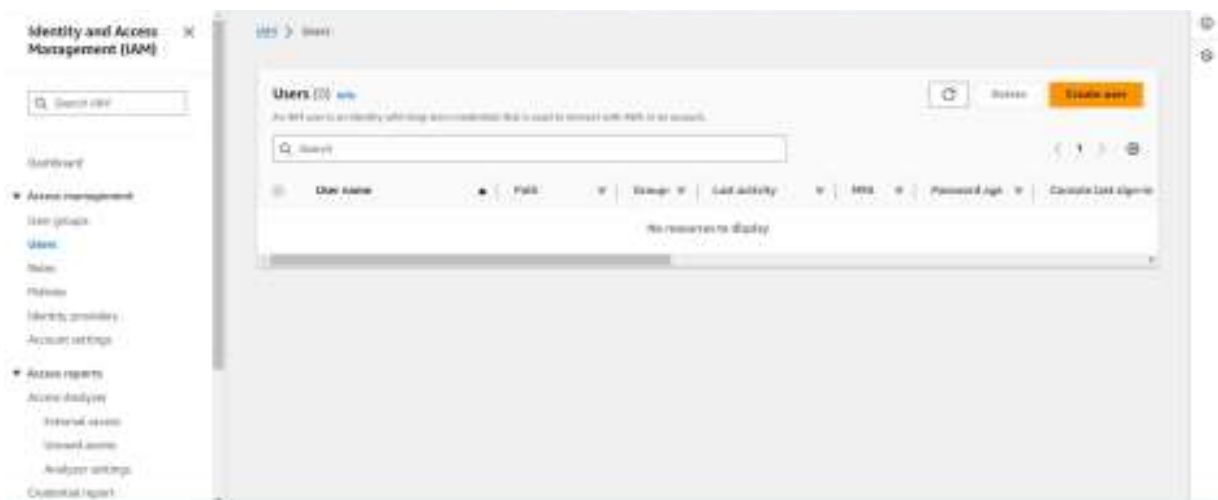
Name	Cloud9 IDE	Environment type	Connection	Framework	Owner
WebApp01	User	EC2 instance	Secure Shell (SSH)	None	arn:aws:cloud9:us-east-1:123456789012:user:example

## Step 2: Creating IAM user.

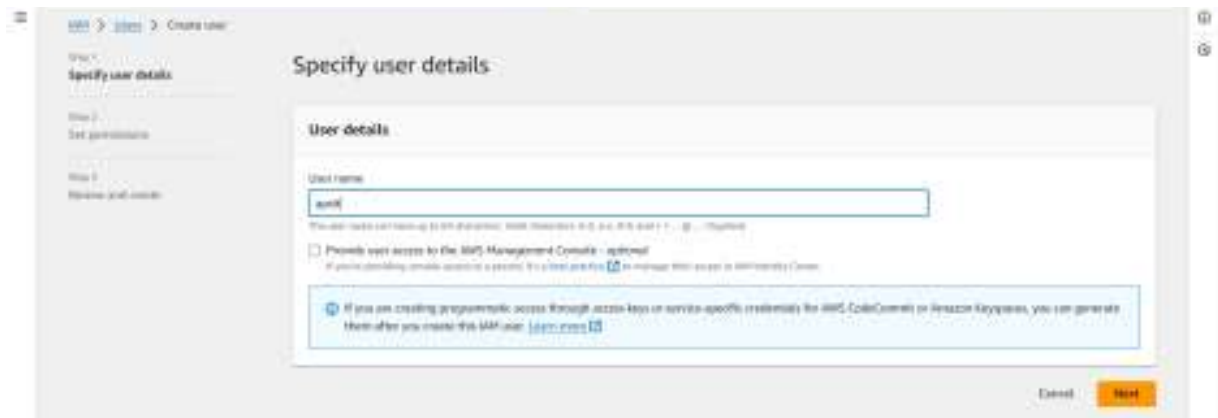
1) Search IAM on the services search bar and open it. Click on Create User



2) Click on the create user

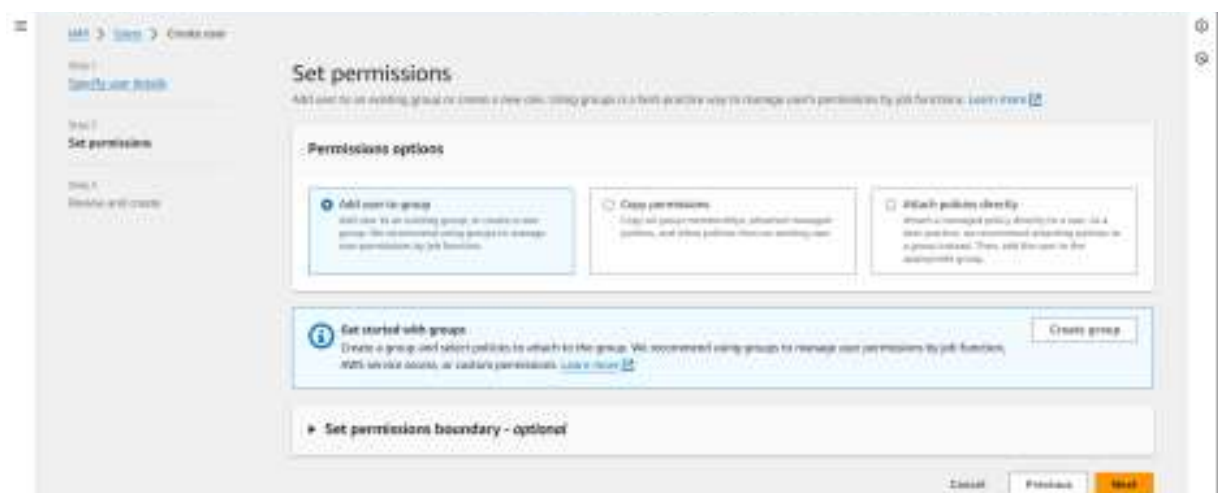


3) Write the name of the user you want to add and click on next



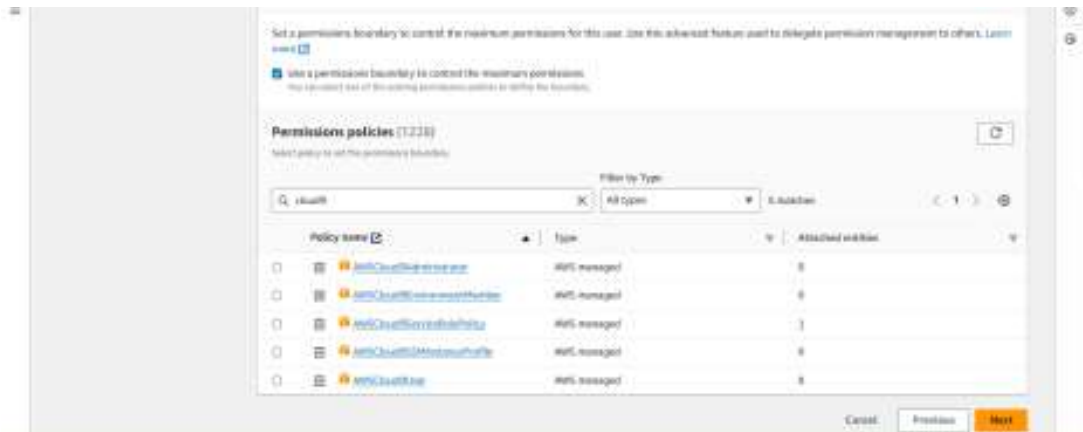
The screenshot shows the 'Specify user details' step in the AWS IAM console. The 'User name' field contains 'sumit'. Below it, there is a checkbox for 'Provide user access to the IAM Management Console - optional'. A blue information box at the bottom states: 'If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user.' The 'Next' button is highlighted in orange.

4) Click on the drop down menu of the set permissions boundary



The screenshot shows the 'Set permissions' step in the AWS IAM console. Under 'Permissions options', there are three radio buttons: 'Add user to group' (selected), 'Copy permissions', and 'Attach policies directly'. Below these, a blue information box says: 'Get started with groups: Create a group and select policies to attach to the group. We recommend using groups to manage user permissions by job function, AWS service access, or custom permissions.' A 'Create group' button is next to it. At the bottom, there is a section for 'Set permissions boundary - optional' with a dropdown menu. The 'Next' button is highlighted in orange.

5) Click on the checkbox and search for cloud9 under permissions policies, click on next



6) Scroll down and click on create user

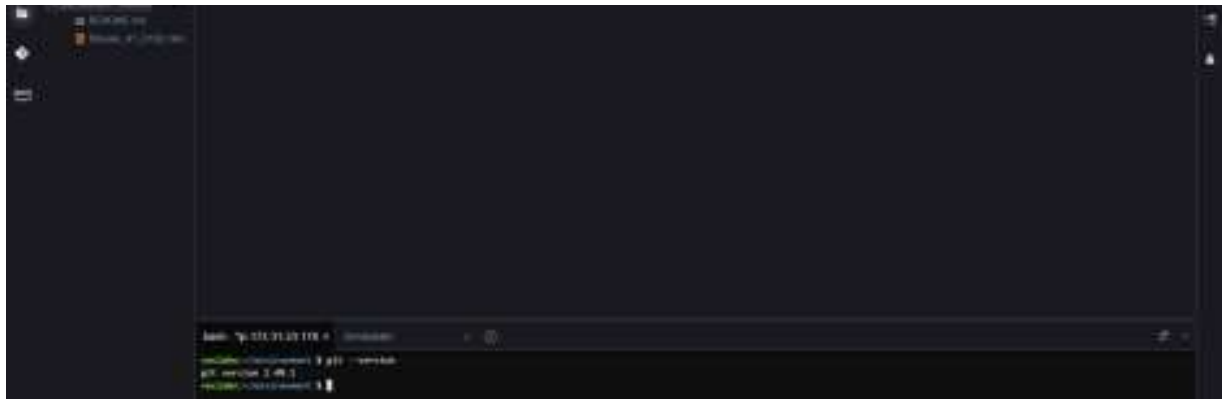


### Step 3: Working on Cloud9 IDE

1) Go to Cloud9 services. Click on Open under Cloud9 IDE



2) This is the Cloud9 IDE interface. The major part of the screen is the coding IDE. There is a command console just below it. For example, the command `git --version` is run.



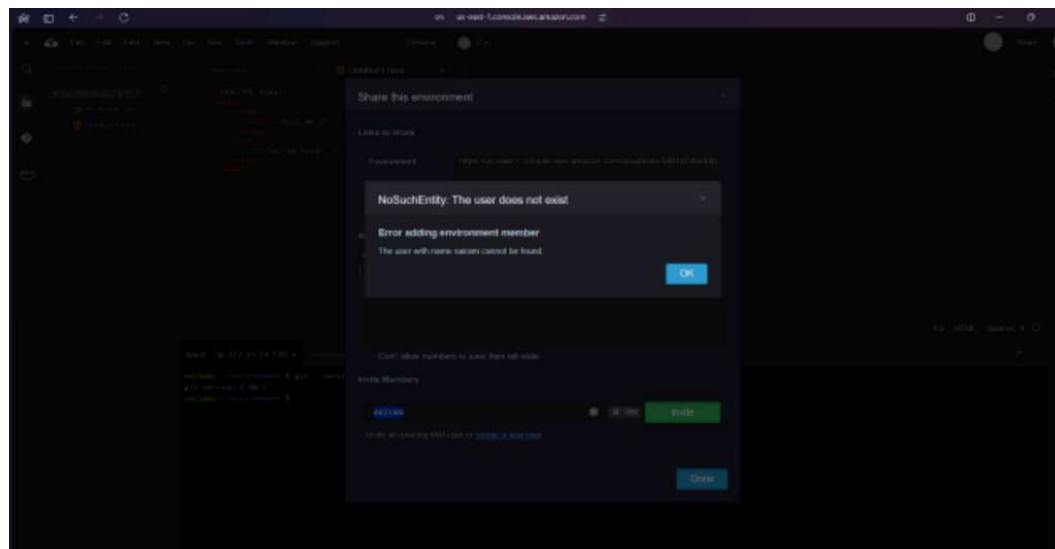
3) To add a file, click on file. For this experiment, we are to add an HTML file. So go to File → New From Template → HTML file. This gives a basic HTML template on the coding IDE



4) Make a basic website on the HTML template and save it.



After saving, on the toolbar towards the far right, click on Share.  
Then put the username that you had put during creating IAM user.



Here, it gives an error as Cloud9 was created on the academy account where creating an IAM group is not available, meanwhile on the personal account, the services of Cloud9 have been deprecated. So currently, it is not possible to integrate the cloud9 and IAM parts of the experiment.