**Advance DevOps lab**

**Experiment 3**

**Name: BALOCH MUZAMMIL HAFEEZ ROLL NO: 612012**

**Semester: V**

**Branch: Information Technology**

1. **What is AWS Cloud9? Features of AWS Cloud9.**

Ans: AWS Cloud9 is an integrated development environment, sometimes known as an IDE.

The AWS Cloud9 IDE offers a powerful code-editing experience that includes support for numerous programming languages, runtime debuggers, and a built-in terminal. It comes with a suite of tools for coding, designing, executing, testing, and debugging applications, as well as assistance with cloud deployment.

A web browser is used to access the AWS Cloud9 IDE. You can customise the IDE to your specific requirements. You can change the colour themes, assign shortcut keys, enable programming language-specific syntax colouring and code formatting, and do other things.

Cloud9 provides the following functionalities:

1. Code with just a browser:

AWS Cloud9 enables you to build, execute, and debug programs using only a browser and without installing or maintaining a desktop IDE.

1. Start new projects quickly:

AWS Cloud9 EC2 environments have support for 40+ programming languages, allowing you to start creating code for popular application stacks in minutes.

1. Code together in real time:

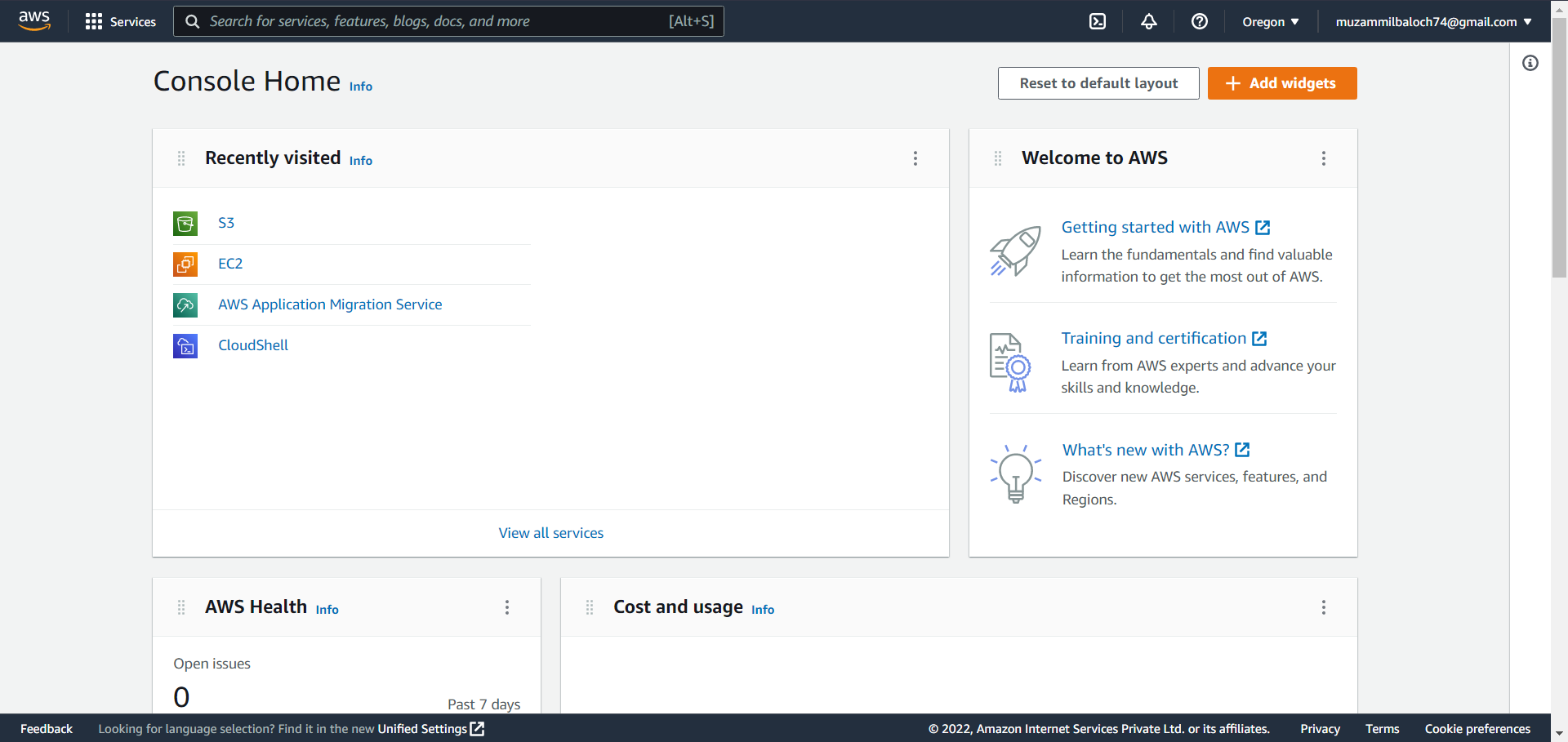
AWS Cloud9 makes it simple to collaborate on code. In just a few clicks, you can share your development environment with your team and pair program together.

1. Build serverless applications with ease:

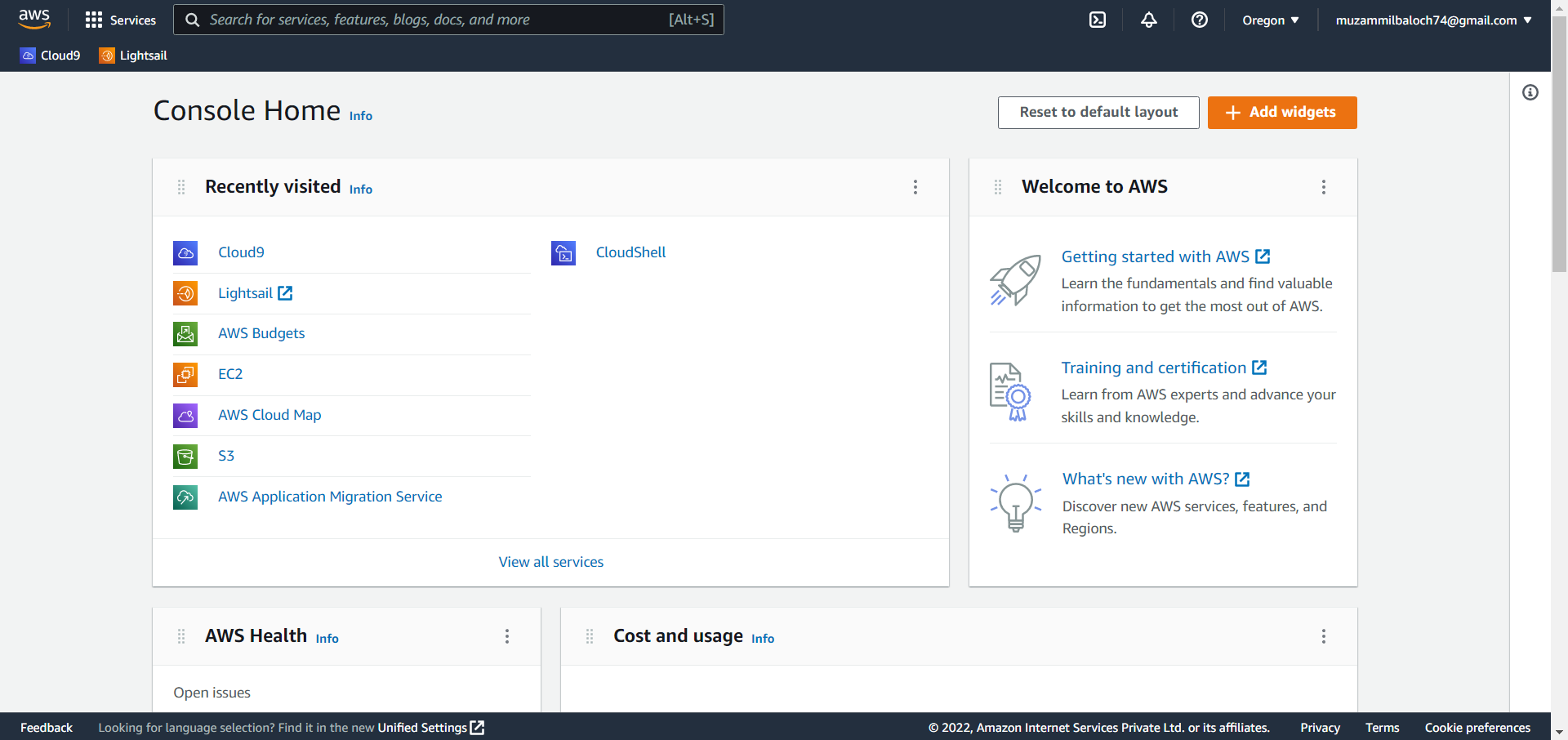
AWS Cloud9 offers a streamlined environment for developing serverless apps. It allows you to simply create resources, debug your code, and switch between local and remote execution.

1. **a.** **Run Node.js scripts in an AWS Cloud9 development environment.**

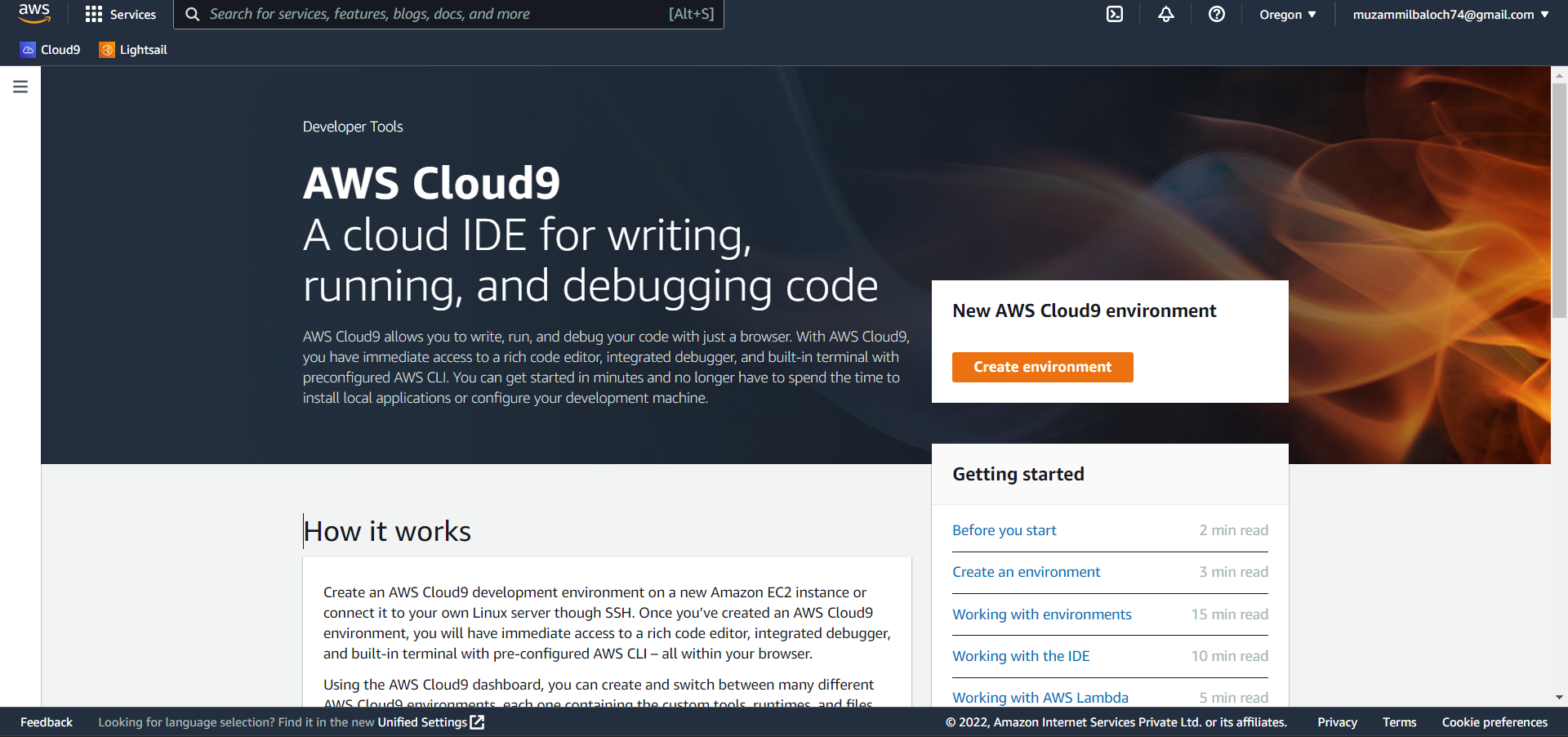
Step 1: Management Console Dashboard.



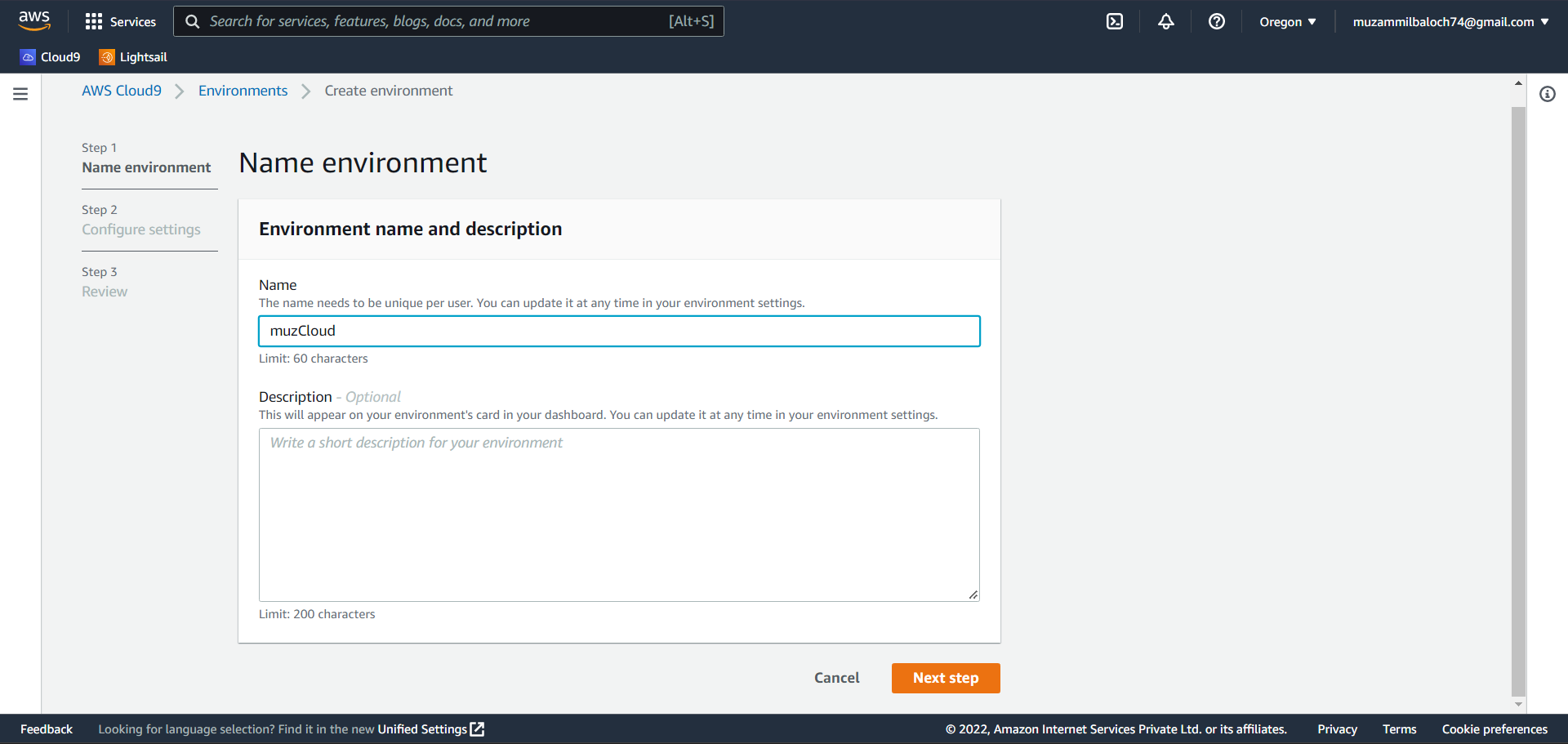
Step 2: Click on services and then click on Cloud9.



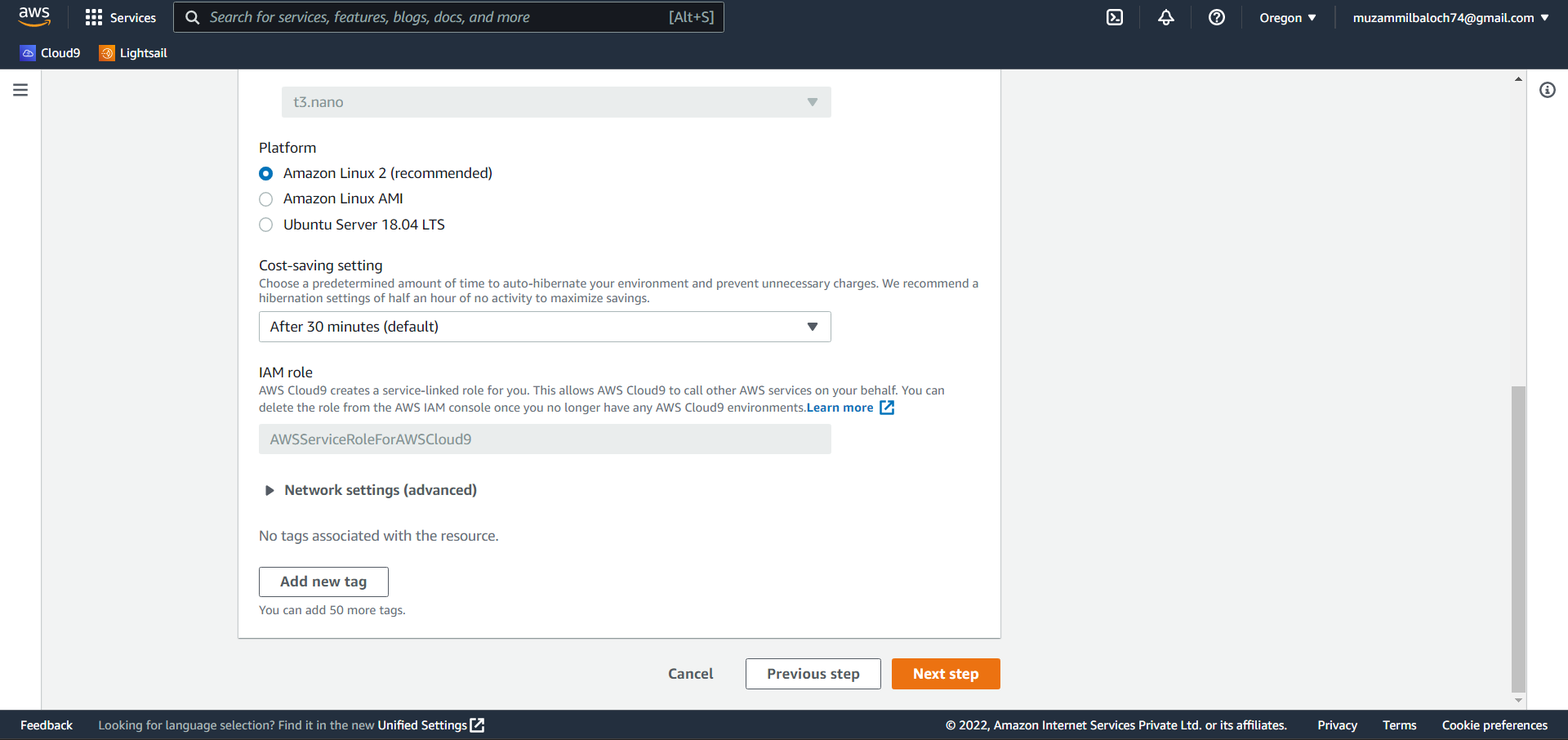
Step 3: Create an environment.



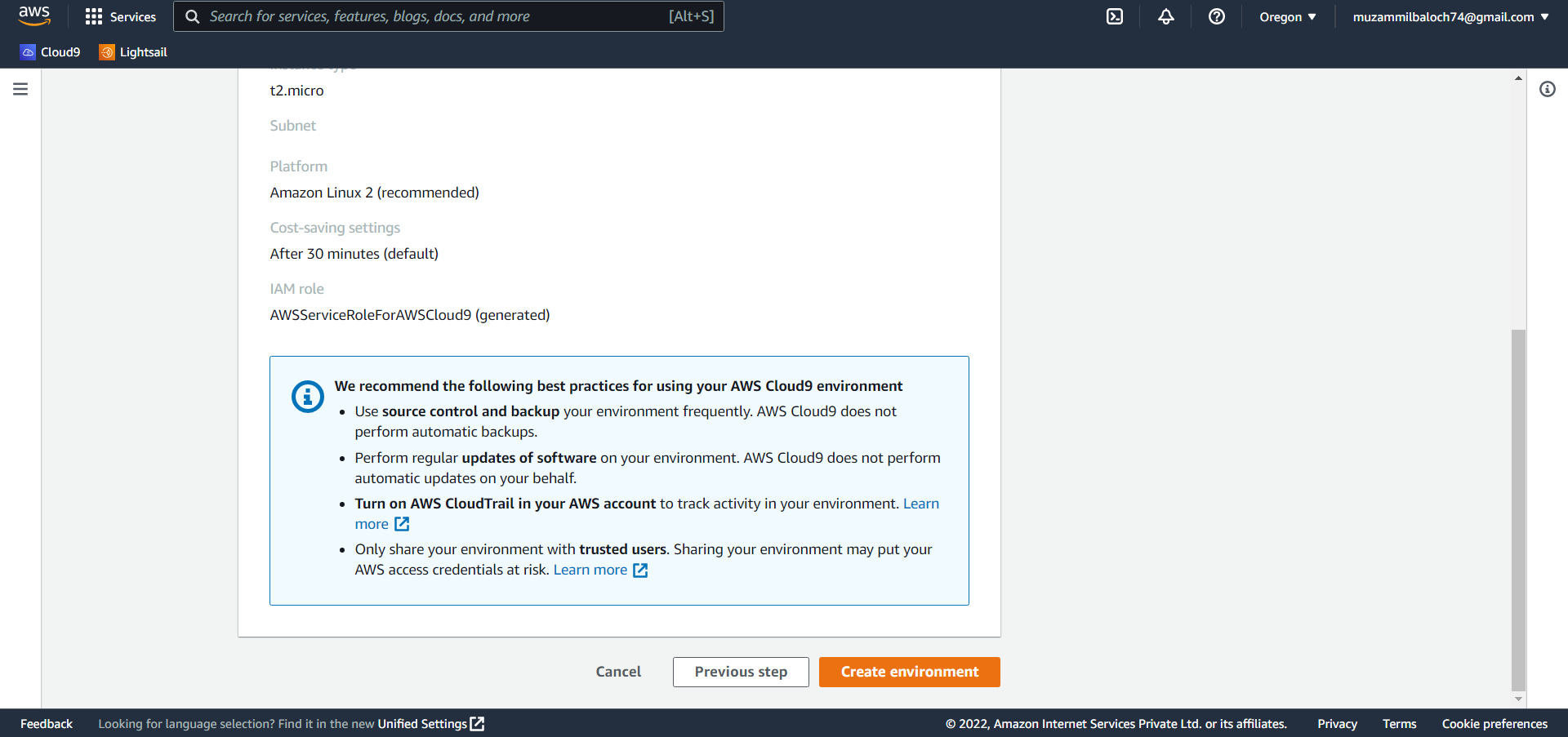
Step 4: Name the environment.



Step 5: Configure the environment settings according to your needs.

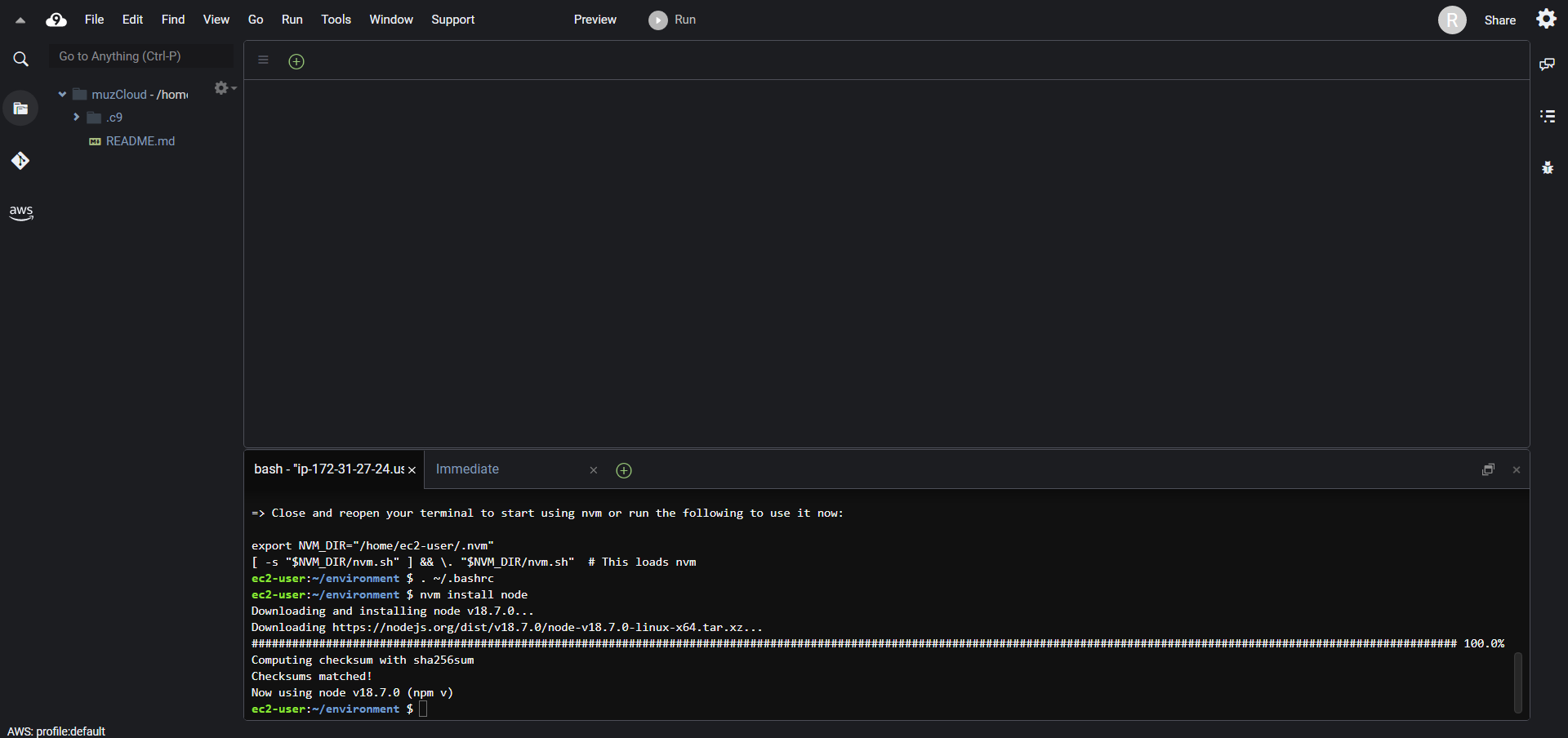


Step 6: Review the settings and create the environment.

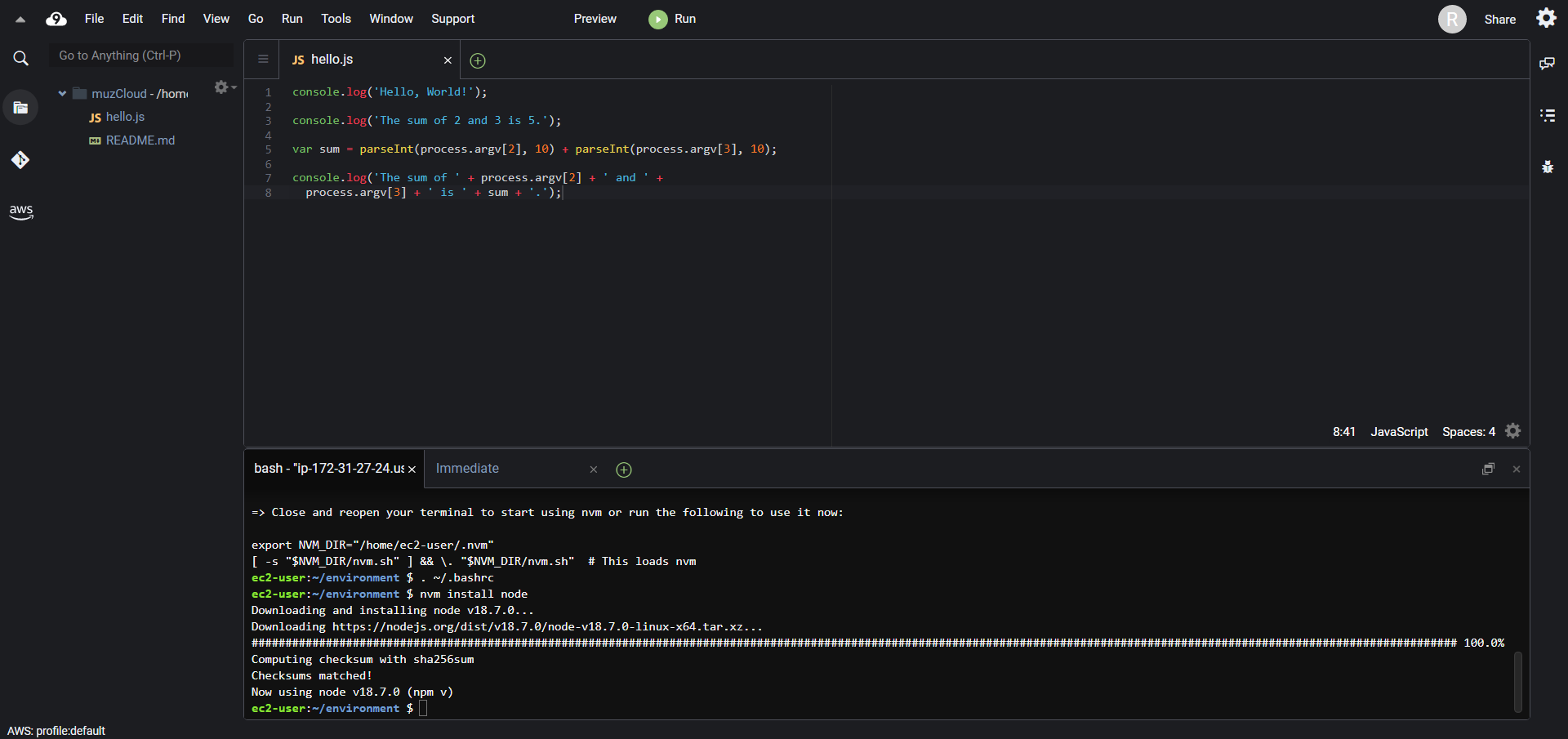


Step 7: Install the necessary tools in the ec2 instance.

1. sudo yum -y update – To ensure that the most recent security and bug patches are installed.
2. curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.33.0/install.sh | bash - To begin installing Node.js, use this command to download Node Version Manager (nvm).
3. . ~/.bashrc - To begin utilising nvm, either close the terminal session and restart it, or source the /.bashrc file that contains the nvm loading commands.
4. nvm install node - Run this command to install the most recent Node.js version.

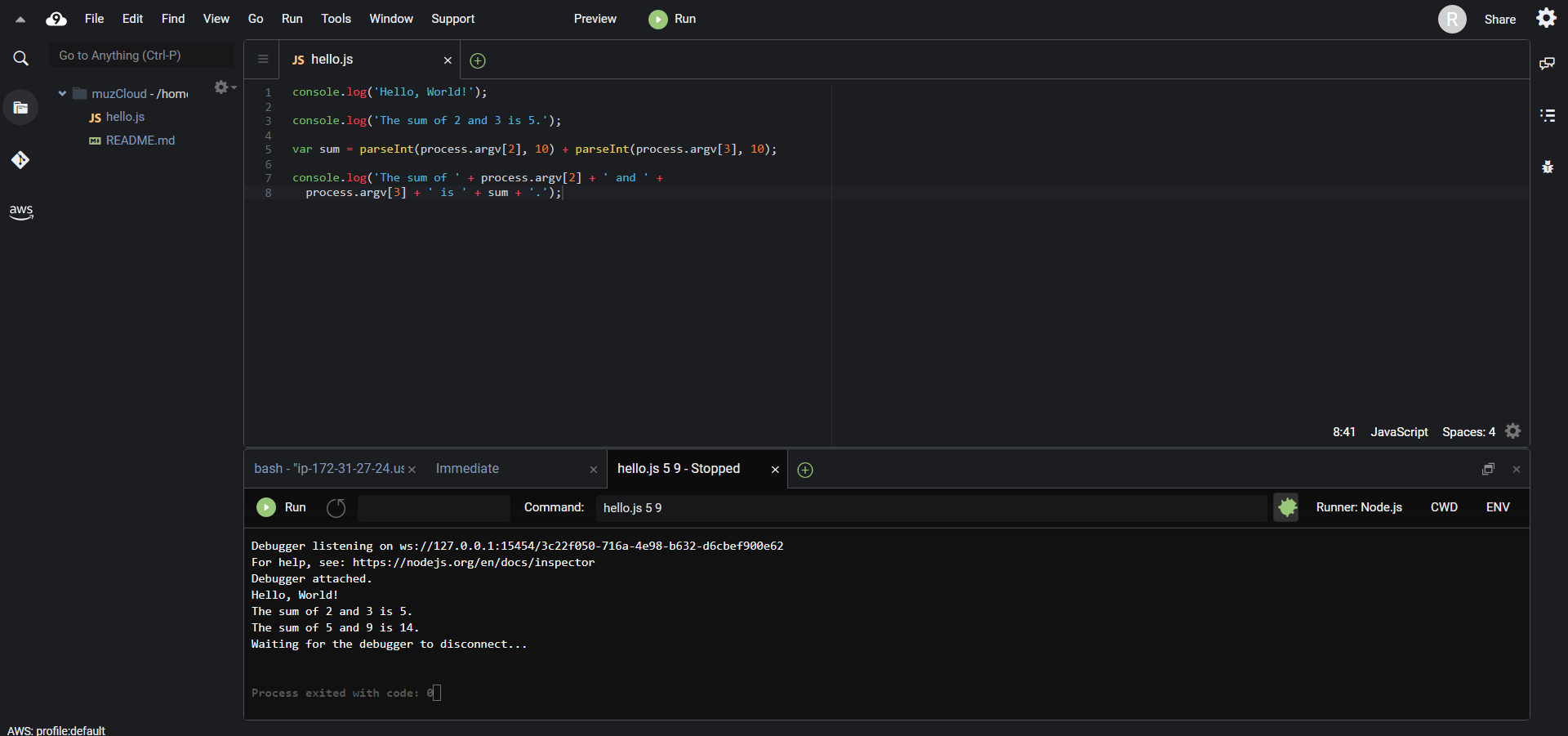


Step 8: Create a new file in the AWS Cloud9 IDE and save it as hello.js. (On the menu bar, select File, New File; to save the file, select File, Save.).



Step 9: Run the code.

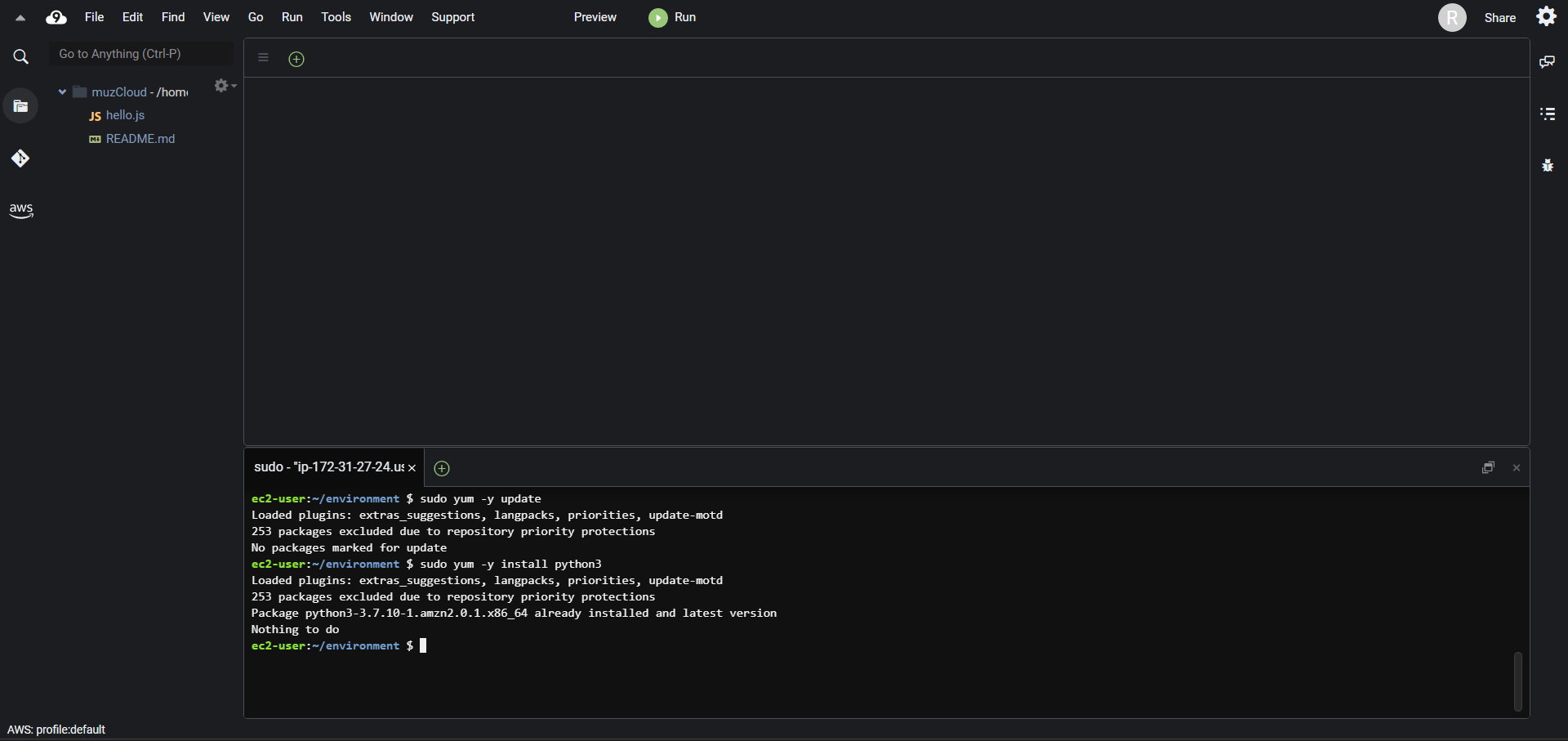
1. Select Run, Run Configurations, New Run Configuration from the menu bar in the AWS Cloud9 IDE.
2. Choose Runner: Auto, then Node.js from the [New] - Idle tab.
3. Type hello.js 5 9 for Command. The numbers 5 and 9 in the code reflect process.argv[2] and process.argv[3], respectively. (The name of the runtime (node) is represented by process.argv[0], and the name of the file (hello.js) is represented by process.argv[1].)
4. Select the Run button and compare the results.



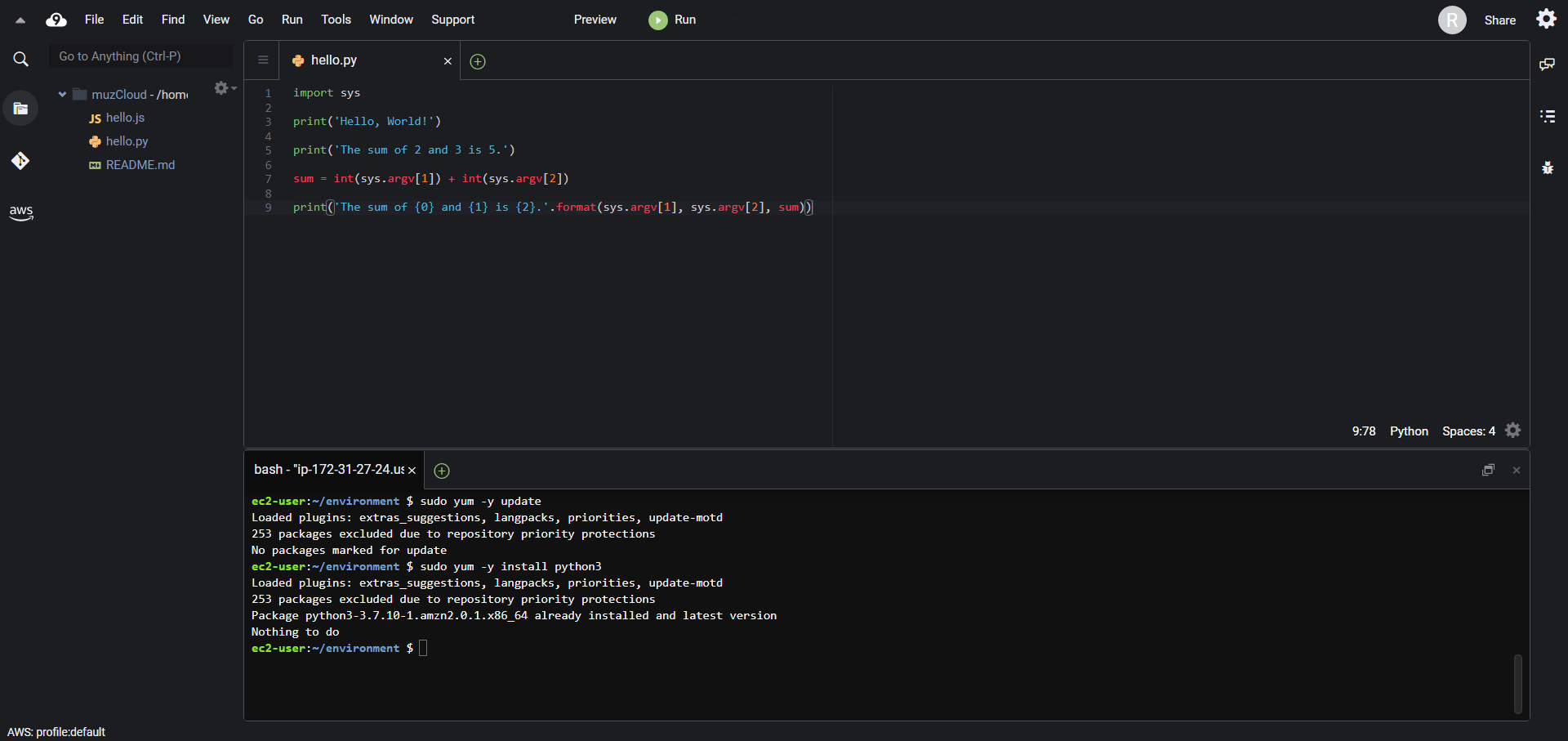
**b. Run Python code in an AWS Cloud9 development environment.**

Step 1: Install python in the ec2 instance.

1. sudo yum -y update – To ensure that the most recent security and bug patches are installed.
2. sudo yum -y install python3- Run install command to install Python.

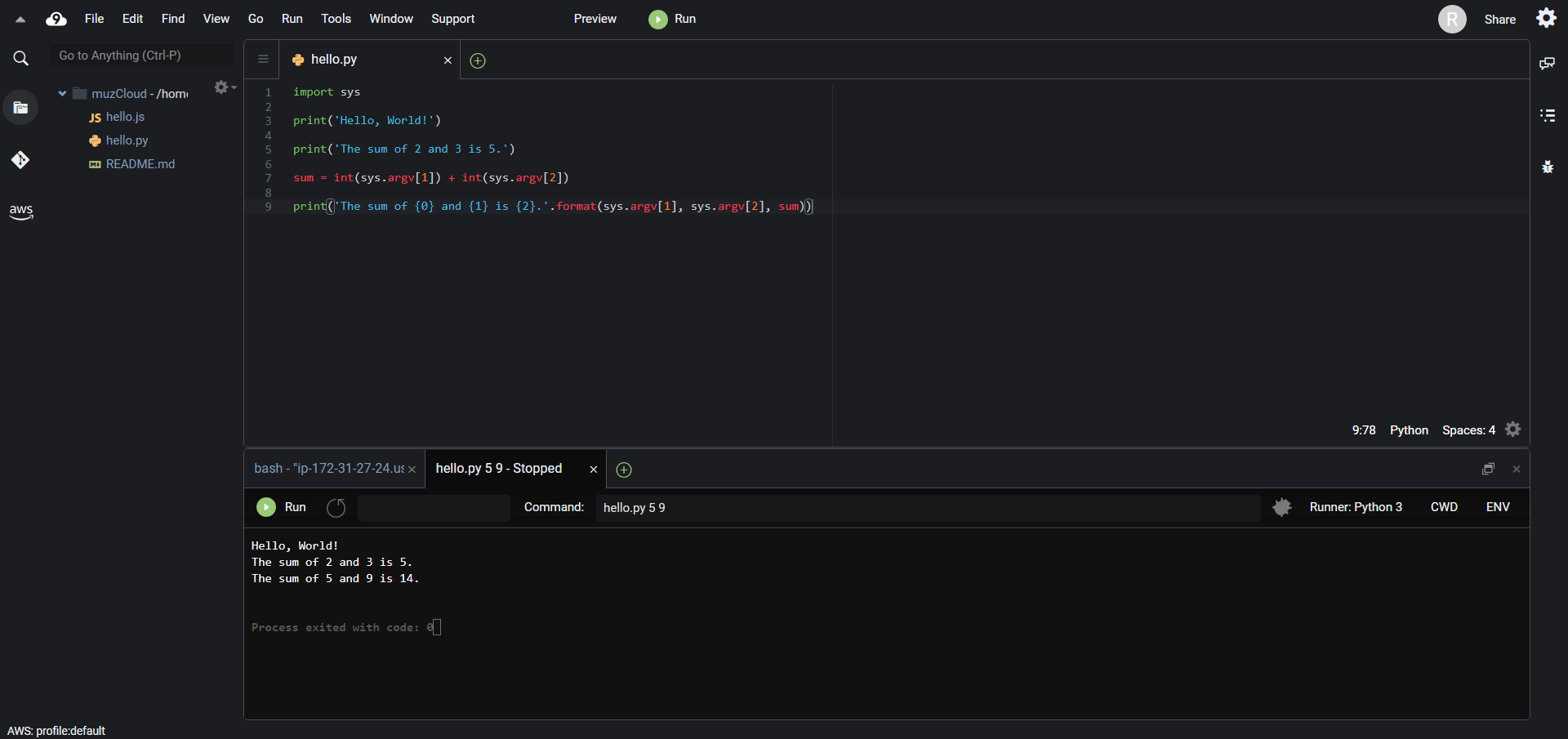


Step 2: Create a new file in the AWS Cloud9 IDE and save it as hello.py. (On the menu bar, select File, New File; to save the file, select File, Save.).



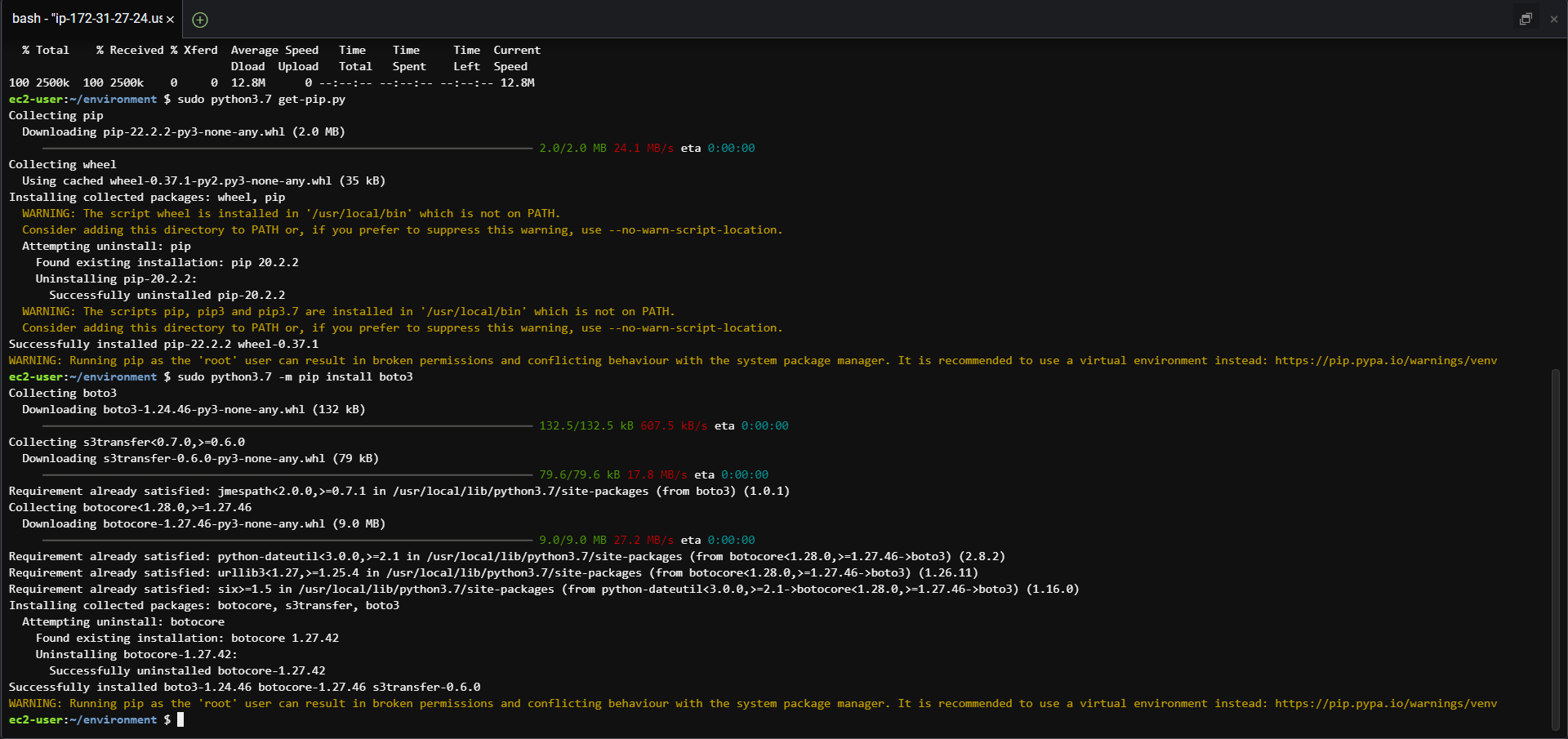
Step 3: Execute the code.

1. Choose Run, Run Configurations, New Run Configuration from the menu bar in the AWS Cloud9 IDE.
2. Enter hello.py 5 9 for Command on the [New] - Stopped tab. The numbers 5 and 9 in the code reflect sys.argv[1] and sys.argv[2], respectively.
3. Select Run and compare your results.



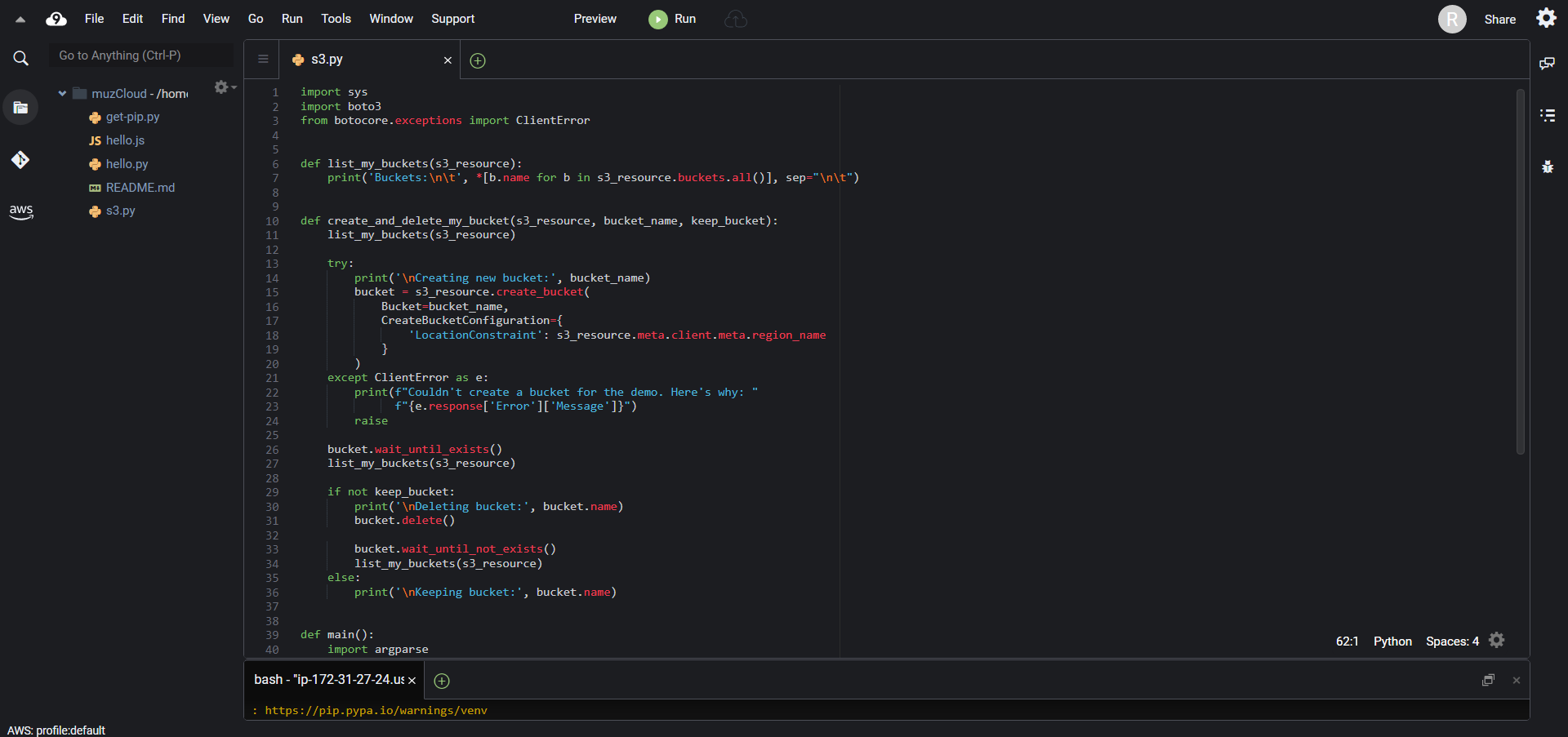
**c. Install AWS SDK boto3 for python and node.js and create S3 bucket and delete it using boto3 API.**

Step 1: Install and configure the AWS SDK for Python (Boto3).



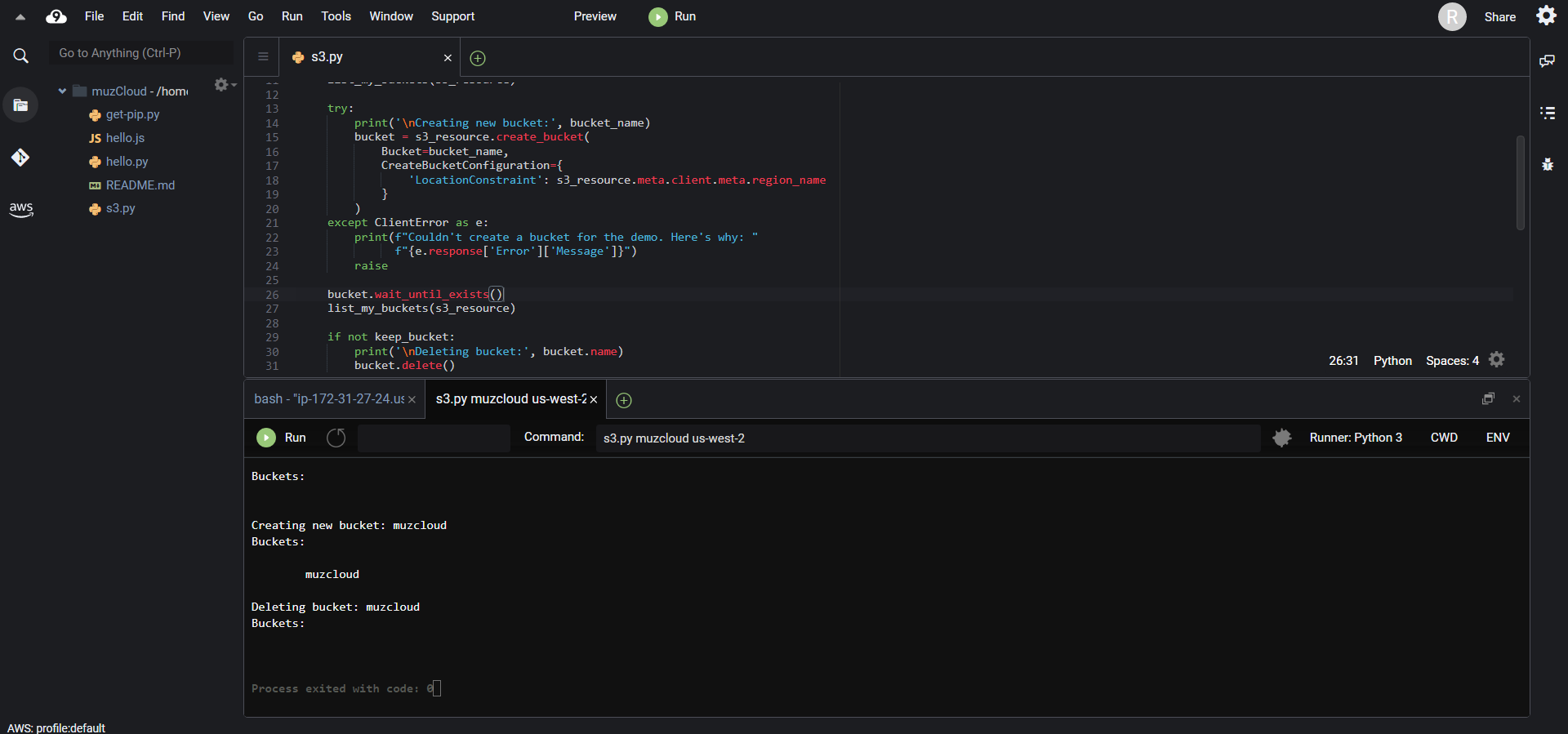
Step 2: Add AWS SDK code.

In the AWS Cloud9 IDE, create a file and save the file with the name s3.py.

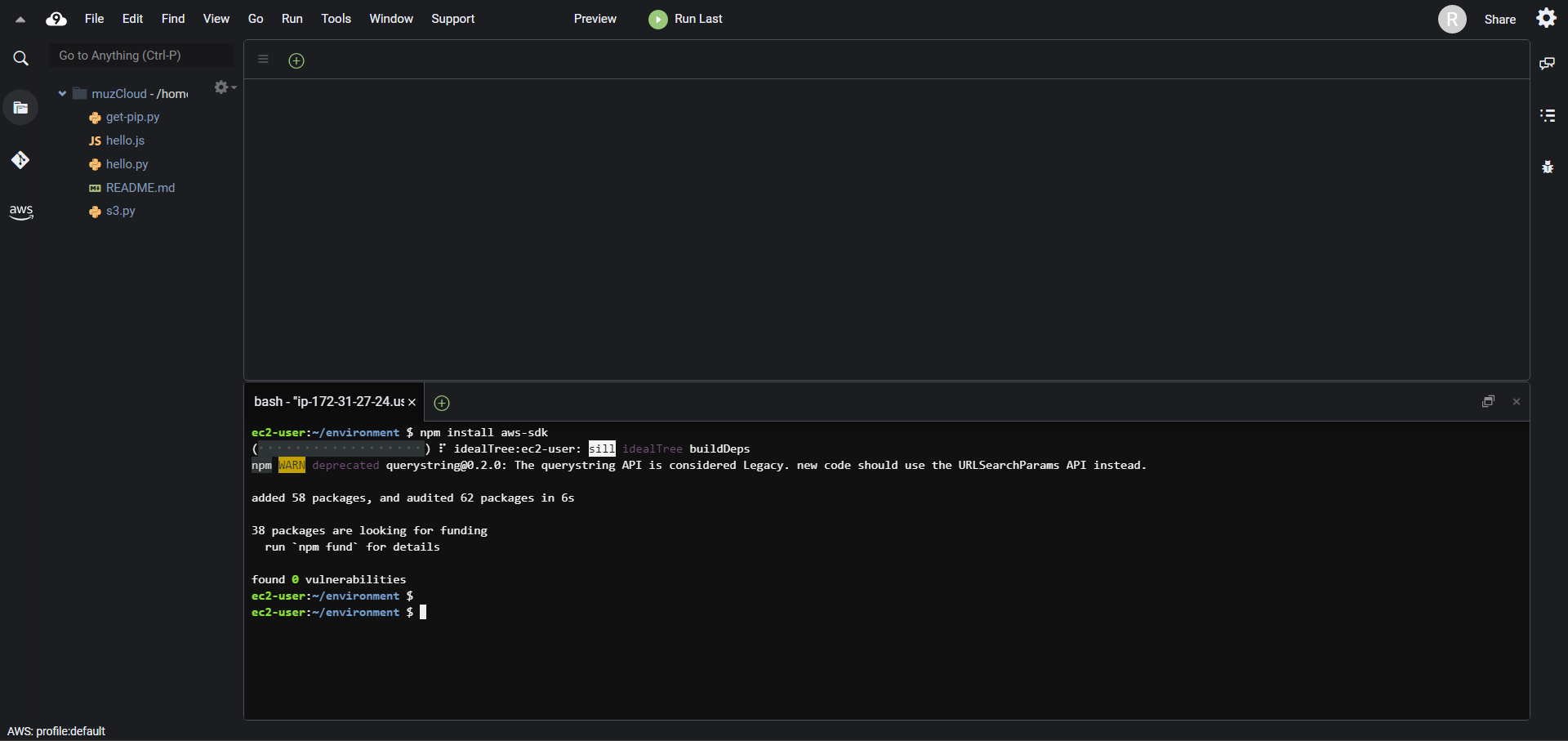


Step 3: Run the AWS SDK code.

1. Select Run, Run Configurations, and New Run Configuration from the menu bar.
2. Enter s3.py my-test-bucket us-west-2 for Command, where my-test-bucket is the name of the bucket to create and us-west-2 is the AWS Region ID where your bucket is formed.

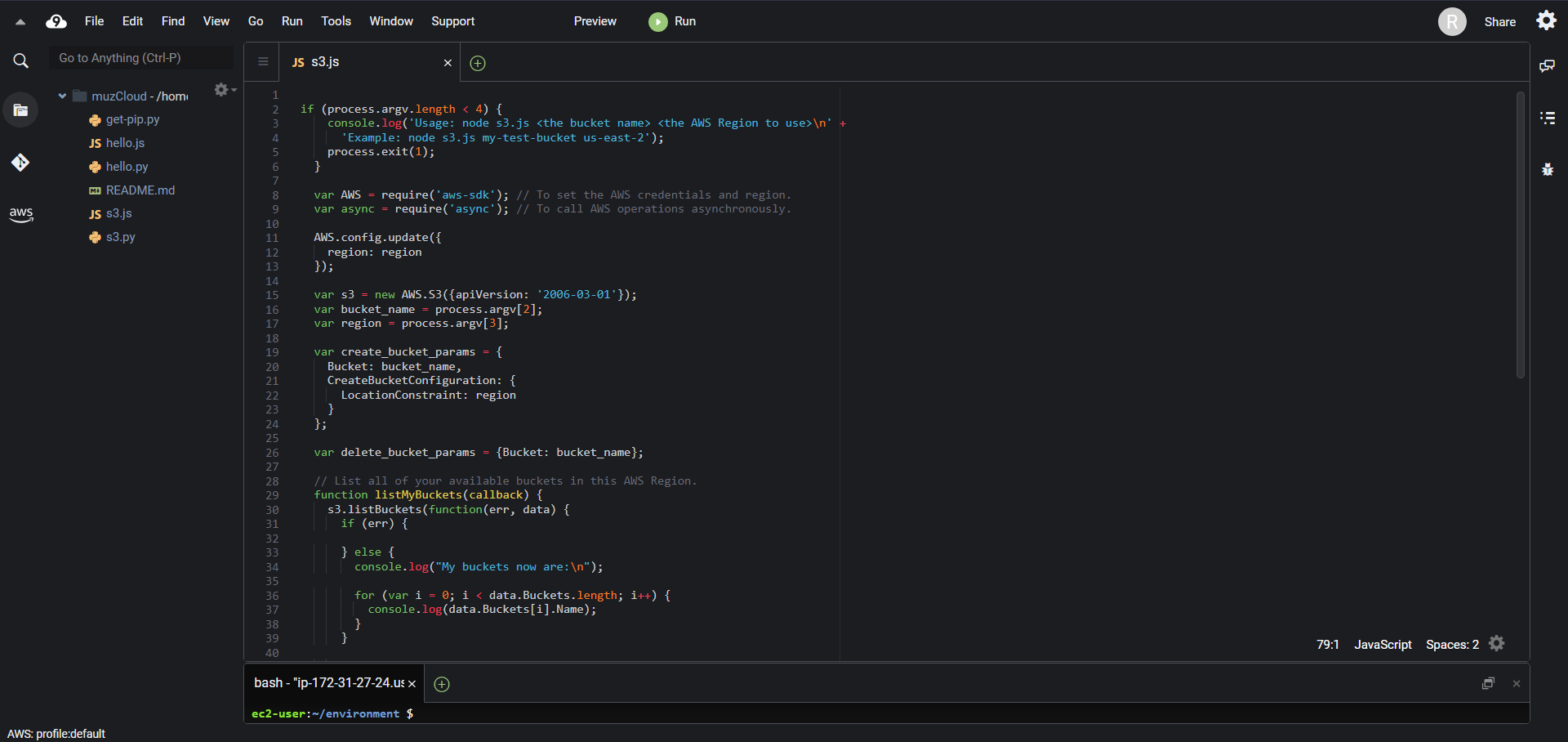
****

Step 4: Install and configure the AWS SDK for JavaScript in Node.js).

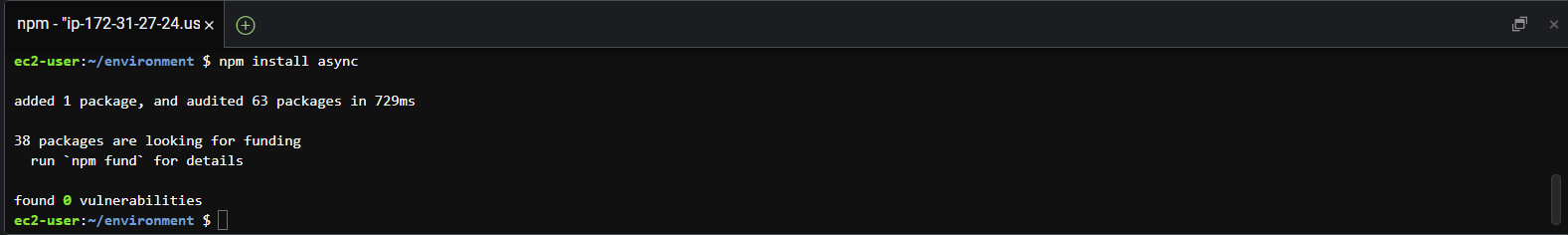
****

Step 5 : Add AWS SDK code.

In the AWS Cloud9 IDE, create a file with this content, and save the file with the name s3.js.

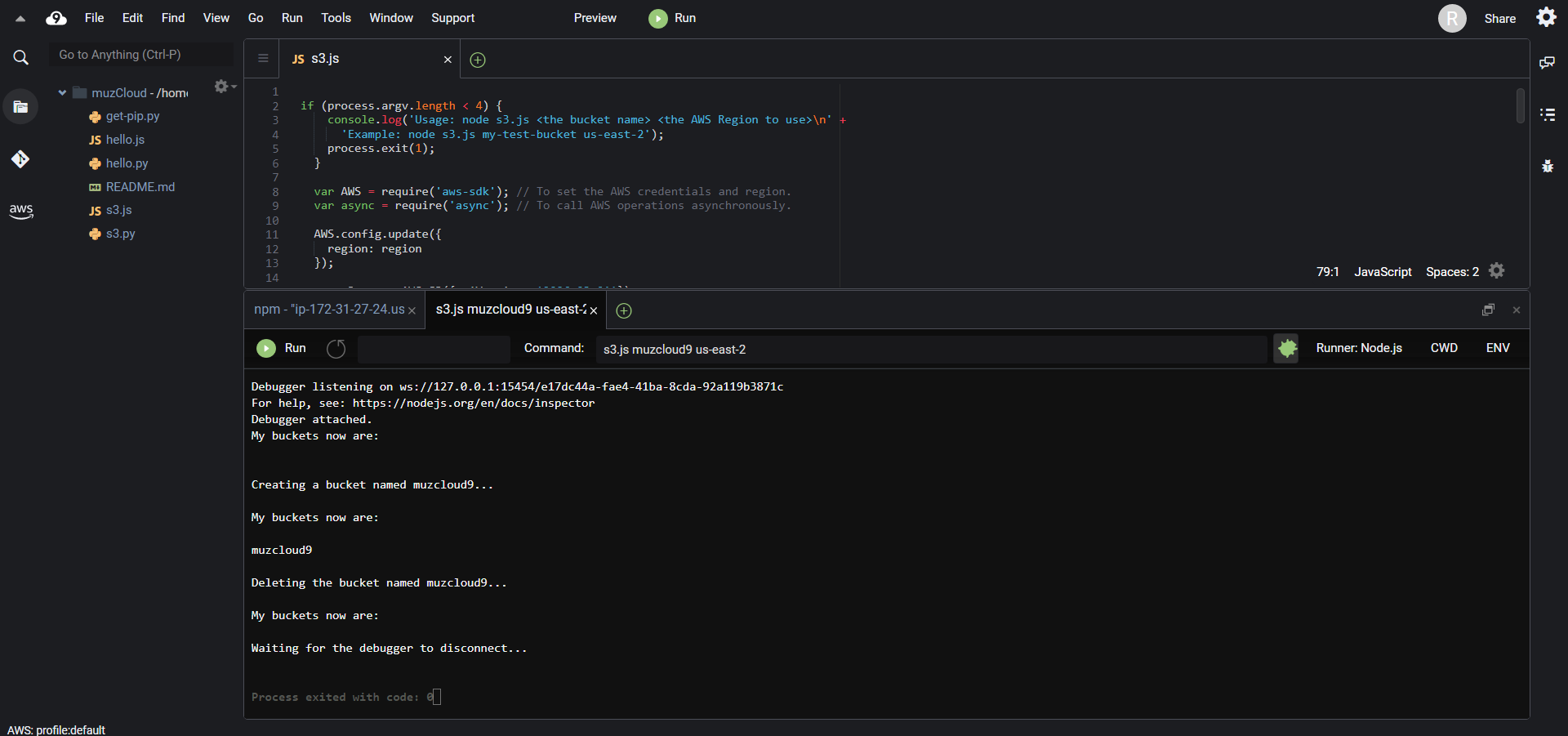


Step 6: Use npm to perform the install command to enable the code to call Amazon S3 operations asynchronously.



Step 7: Run the AWS SDK code.

1. Select Run, Run Configurations, and New Run Configuration from the menu bar.
2. Enter s3.js my-test-bucket us-west-2 for Command, where my-test-bucket is the name of the bucket to create and us-west-2 is the AWS Region ID where your bucket is formed.



Step 8: Delete the environment

