

# EXPERIMENT-3

**NAME:** SHAIKH MUBASHIRA TUFEL AHMED  
**ROLL NO:** 612055 **COURSE:** ADVANCE DEVOPS(ITL504)  
**BRANCH:** T.E. INFORMATION TECHNOLOGY (SEM 5)

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

➔ AWS Cloud9 is an integrated development environment, or IDE.

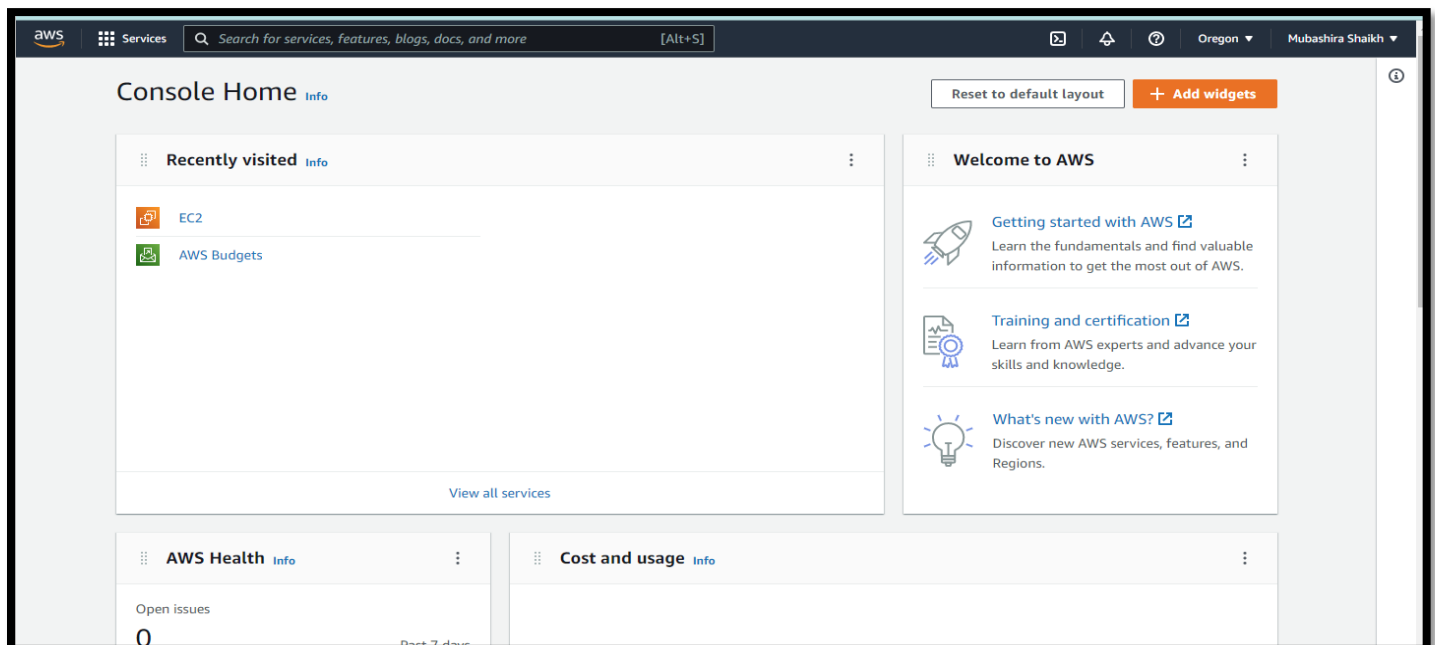
The AWS Cloud9 IDE offers a rich code-editing experience with support for several programming languages and runtime debuggers, and a built-in terminal. It contains a collection of tools that you use to code, build, run, test, and debug software, and helps you release software to the cloud.

You access the AWS Cloud9 IDE through a web browser. You can configure the IDE to your preferences. You can switch color themes, bind shortcut keys, enable programming language-specific syntax coloring and code formatting, and more.

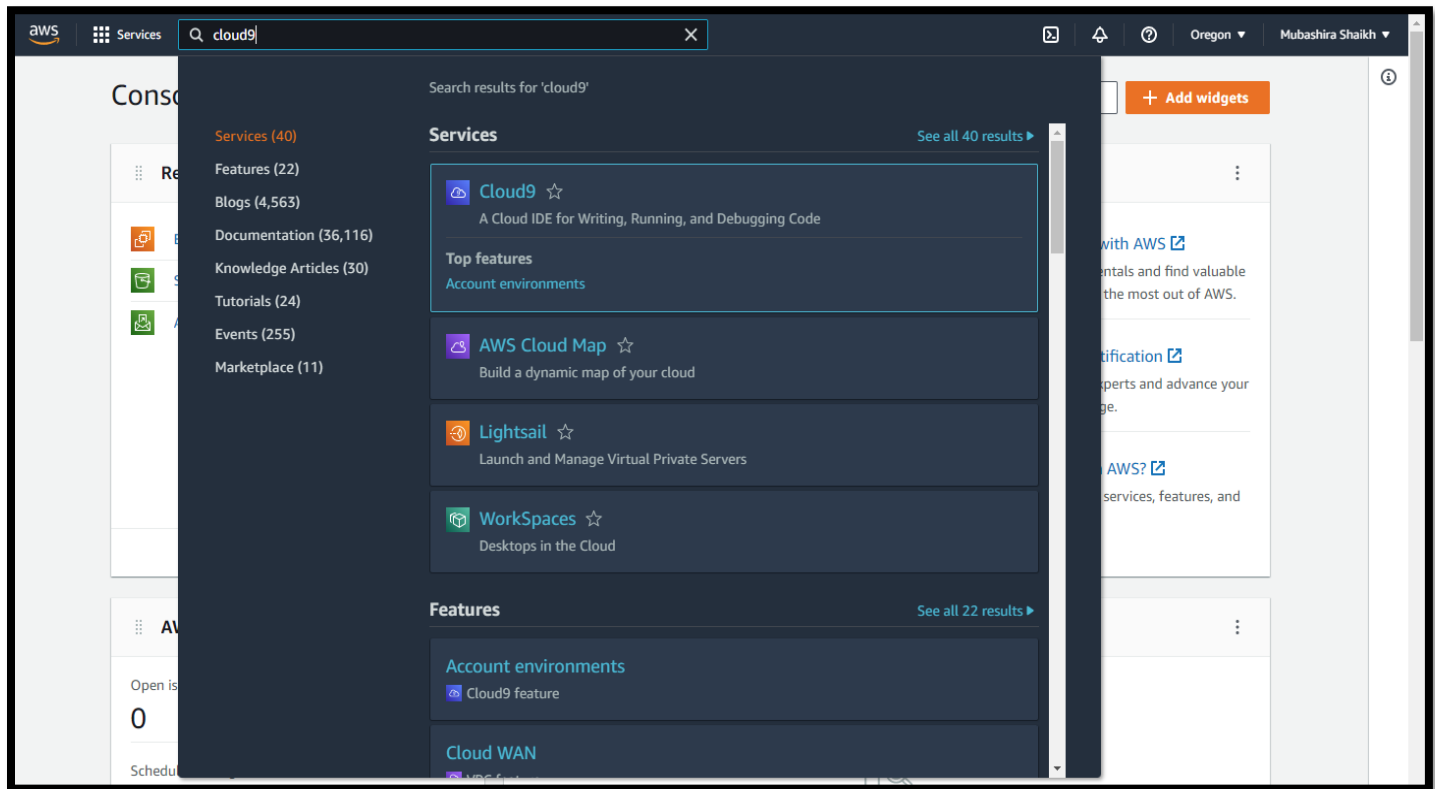
### ➔ Features of AWS Cloud9:

- ✓ Fully-featured Editor
- ✓ Broad Selection of Run Configurations
- ✓ Integrated Debugger
- ✓ Integrated Tools for Serverless Development
- ✓ Connectivity to Any Linux Server Platform
- ✓ Built-in Terminal
- ✓ Collaborative Editing and Chat
- ✓ Continuous Delivery Toolchain
- ✓ File Revision History
- ✓ Themes
- ✓ Keyboard Shortcuts
- ✓ Built-in Image Editor

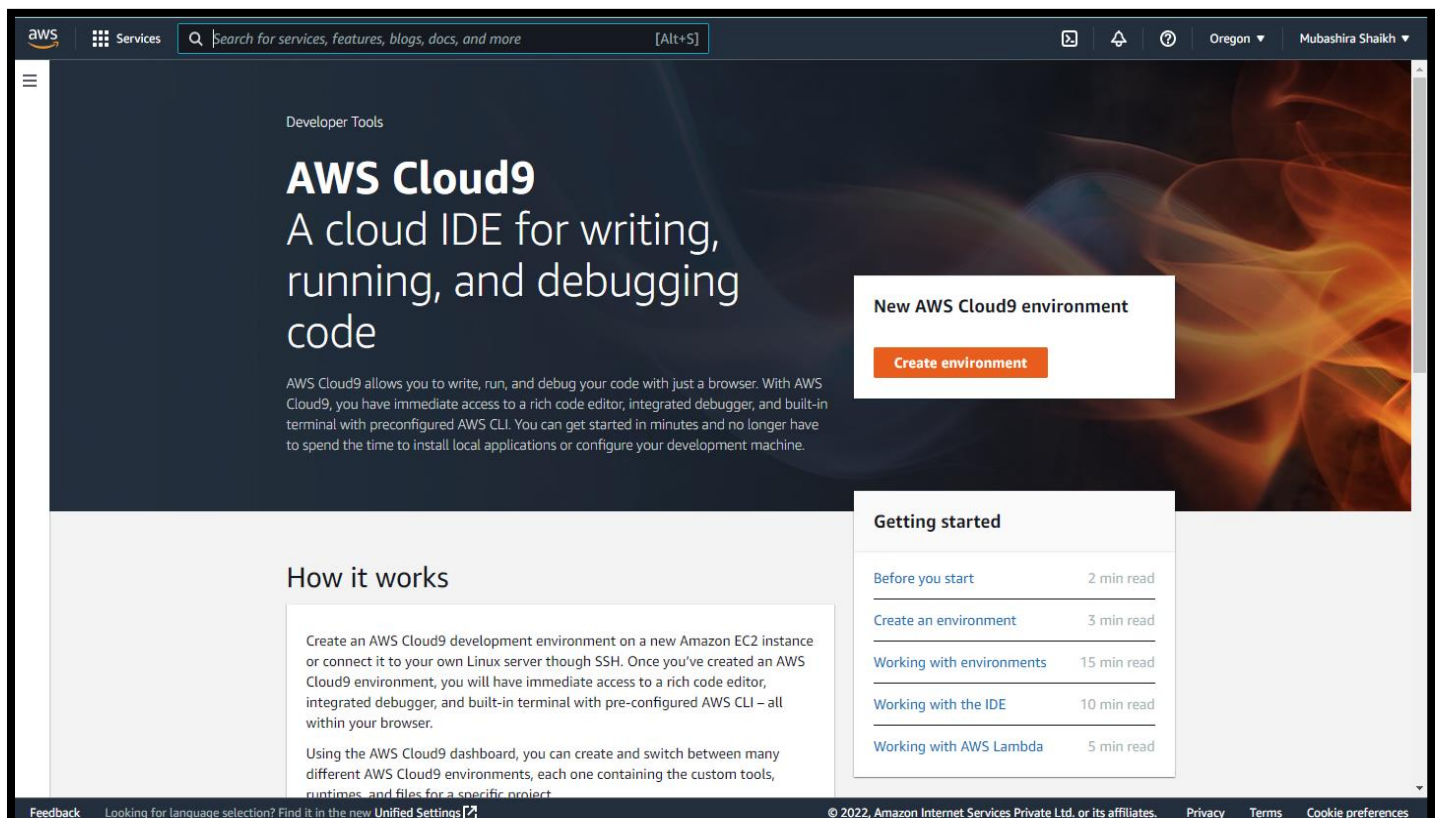
## Step 1: AWS Management Console Dashboard.



## Step 2: Search for Cloud9 and select it.



## Step 3: Click on 'Create environment'.



## Step 4: Name your environment and configure the settings.

This screenshot shows the 'Name environment' step of the AWS Cloud9 'Create environment' wizard. A notification at the top states: 'AWS root account login detected. We do not recommend using your AWS root account to create or work with environments. Use an IAM user instead. This is an AWS security best practice. For more information, see [Setting Up to Use AWS Cloud9](#).' The breadcrumb trail is 'AWS Cloud9 > Environments > Create environment'. The left sidebar shows 'Step 1 Name environment' as the active step, with 'Step 2 Configure settings' and 'Step 3 Review' listed below. The main content area is titled 'Environment name and description'. It contains a 'Name' field with the value 'Mubashira' and a 'Limit: 60 characters' note. Below this is a 'Description - Optional' section with a text area containing the placeholder 'Write a short description for your environment' and a 'Limit: 200 characters' note.

**Step 1**  
Name environment

**Step 2**  
Configure settings

**Step 3**  
Review

**Environment name and description**

**Name**  
The name needs to be unique per user. You can update it at any time in your environment settings.  
Mubashira  
Limit: 60 characters

**Description - Optional**  
This will appear on your environment's card in your dashboard. You can update it at any time in your environment settings.  
Write a short description for your environment  
Limit: 200 characters

This screenshot shows the 'Configure settings' step of the AWS Cloud9 'Create environment' wizard. The notification and breadcrumb trail are the same as in the previous step. The left sidebar shows 'Step 2 Configure settings' as the active step. The main content area is titled 'Environment settings'. It includes an 'Environment type' section with an 'Info' link and a description: 'Run your environment in a new EC2 instance or an existing server. With EC2 instances, you can connect directly through Secure Shell (SSH) or connect via AWS Systems Manager (without opening inbound ports)'. There are three radio button options: 'Create a new EC2 instance for environment (direct access)' (selected), 'Create a new no-ingress EC2 instance for environment (access via Systems Manager)', and 'Create and run in remote server (SSH connection)'. Below this is an 'Instance type' section with four radio button options: 't2.micro (1 GiB RAM + 1 vCPU)' (selected), 't3.small (2 GiB RAM + 2 vCPU)', 'm5.large (8 GiB RAM + 2 vCPU)', and 'Other instance type'. The 't2.micro' option has a note: 'Free-tier eligible. Ideal for educational users and exploration.' The 't3.small' option has a note: 'Recommended for small-sized web projects.' The 'm5.large' option has a note: 'Recommended for production and general-purpose development.'

**Environment settings**

**Environment type** [Info](#)  
Run your environment in a new EC2 instance or an existing server. With EC2 instances, you can connect directly through Secure Shell (SSH) or connect via AWS Systems Manager (without opening inbound ports).

- ☒ **Create a new EC2 instance for environment (direct access)**  
Launch a new instance in this region that your environment can access directly via SSH.
- ☐ **Create a new no-ingress EC2 instance for environment (access via Systems Manager)**  
Launch a new instance in this region that your environment can access through Systems Manager.
- ☐ **Create and run in remote server (SSH connection)**  
Configure the secure connection to the remote server for your environment.

**Instance type**

- ☒ **t2.micro (1 GiB RAM + 1 vCPU)**  
Free-tier eligible. Ideal for educational users and exploration.
- ☐ **t3.small (2 GiB RAM + 2 vCPU)**  
Recommended for small-sized web projects.
- ☐ **m5.large (8 GiB RAM + 2 vCPU)**  
Recommended for production and general-purpose development.
- ☐ **Other instance type**  
Select an instance type.

This screenshot shows the 'Review' step of the AWS Cloud9 'Create environment' wizard. The notification and breadcrumb trail are the same. The left sidebar shows 'Step 3 Review' as the active step. The main content area displays a summary of the configuration: 'No description provided', 'Environment type: EC2', 'Instance type: t2.micro', 'Subnet', 'Platform: Amazon Linux 2 (recommended)', 'Cost-saving settings: After 30 minutes (default)', and 'IAM role: AWSServiceRoleForAWSCloud9 (generated)'. Below the summary is a box with best practices: 'We recommend the following best practices for using your AWS Cloud9 environment'. The list includes: '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](#)', and 'Only share your environment with trusted users. Sharing your environment may put your AWS access credentials at risk. [Learn more](#)'. At the bottom, there are three buttons: 'Cancel', 'Previous step', and 'Create environment'.

**Review**

No description provided

Environment type  
EC2

Instance type  
t2.micro

Subnet

Platform  
Amazon Linux 2 (recommended)

Cost-saving settings  
After 30 minutes (default)

IAM role  
AWSServiceRoleForAWSCloud9 (generated)

**We recommend the following best practices for using your AWS Cloud9 environment**

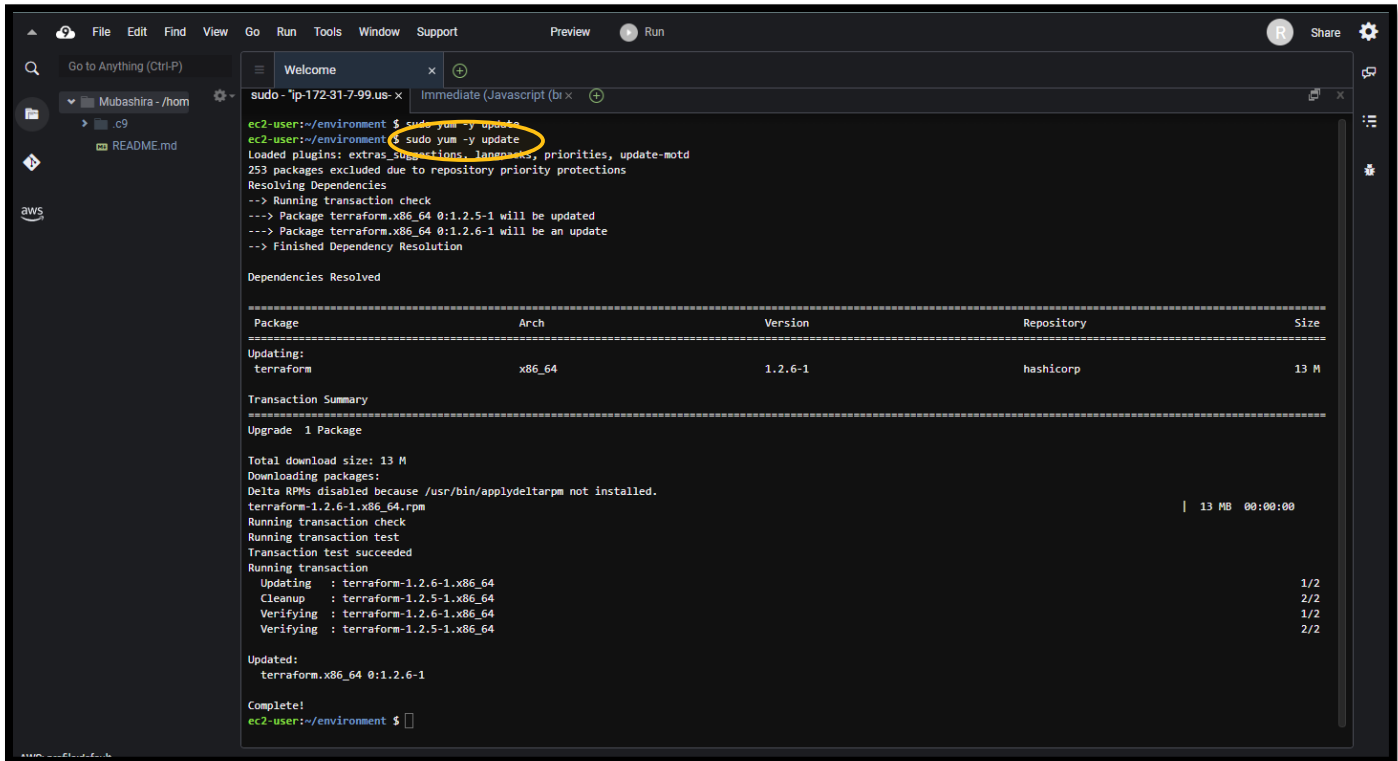
- Use source control and backup your environment frequently. AWS Cloud9 does not perform automatic backups.
- Perform regular updates of software on your environment. AWS Cloud9 does not perform automatic updates on your behalf.
- Turn on AWS CloudTrail in your AWS account to track activity in your environment. [Learn more](#)
- Only share your environment with trusted users. Sharing your environment may put your AWS access credentials at risk. [Learn more](#)

Cancel Previous step Create environment

## FOR PYTHON:

### Step 1: Install Python.

Run the **yum update** for Amazon Linux to help ensure the latest security updates and bug fixes are installed: `sudo yum -y update`. Install Python by running the install command. For Amazon Linux: `sudo yum -y install python3`



```
sudo - 'ip-172-31-7-99.us-east-1-ec2' Immediate (Javascript (br x
ec2-user:~/environment $ sudo yum -y update
ec2-user:~/environment $ sudo yum -y update
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
253 packages excluded due to repository priority protections
Resolving Dependencies
--> Running transaction check
--> Package terraform.x86_64 0:1.2.5-1 will be updated
--> Package terraform.x86_64 0:1.2.6-1 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

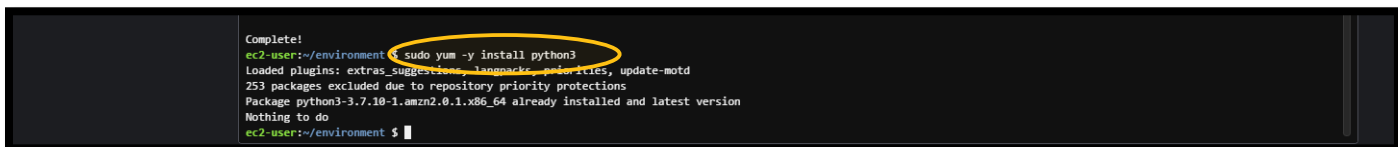
=====
Package                Arch          Version           Repository        Size
=====
Updating:
terraform              x86_64        1.2.6-1           hashicorp         13 M

Transaction Summary
=====
Upgrade 1 Package

Total download size: 13 M
Downloading packages:
Delta RPMs disabled because /usr/bin/applydeltarpm not installed.
terraform-1.2.6-1.x86_64.rpm                                | 13 MB 00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Updating : terraform-1.2.6-1.x86_64                        1/2
  Cleanup  : terraform-1.2.5-1.x86_64                       2/2
  Verifying : terraform-1.2.6-1.x86_64                      1/2
  Verifying : terraform-1.2.5-1.x86_64                      2/2

Updated:
  terraform.x86_64 0:1.2.6-1

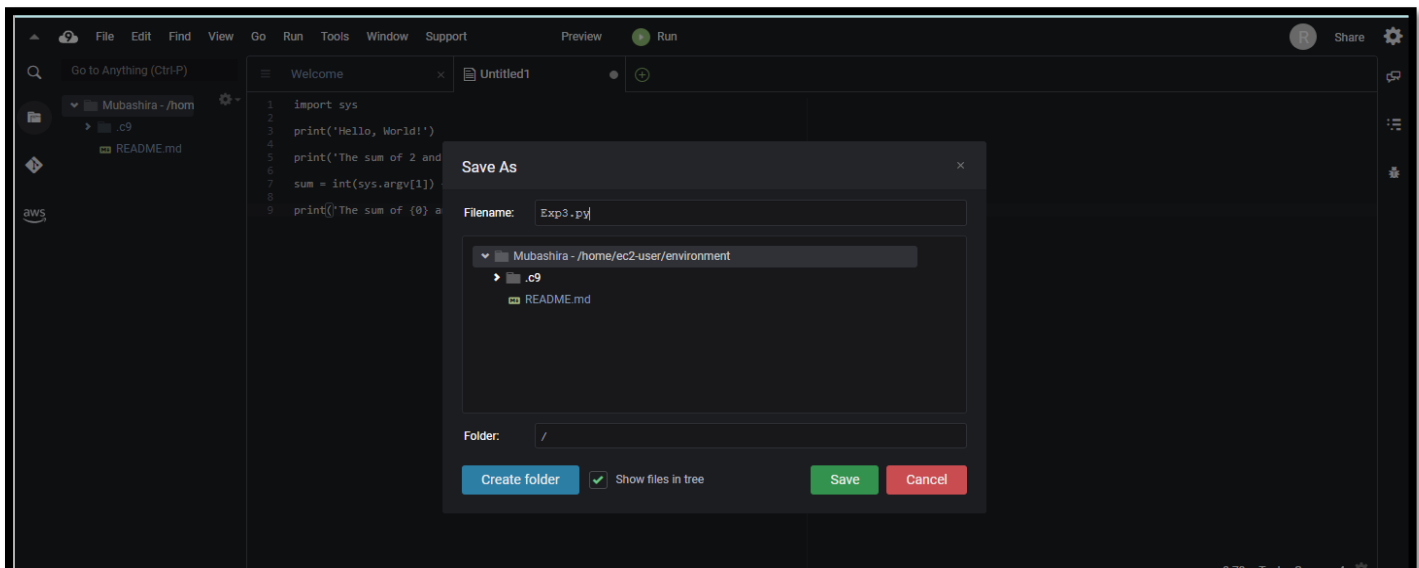
Complete!
ec2-user:~/environment $
```



```
Complete!
ec2-user:~/environment $ sudo yum -y install python3
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
253 packages excluded due to repository priority protections
Package python3-3.7.10-1.amzn2.0.1.x86_64 already installed and latest version
Nothing to do
ec2-user:~/environment $
```

### Step 2: Add code

In the AWS Cloud9 IDE, create a file with the python code and save the file with some name.



```
1 import sys
2
3 print('Hello, World!')
4
5 print('The sum of 2 and 3 is:')
6 sum = int(sys.argv[1]) + int(sys.argv[2])
7
8 print('The sum of {} and {} is: {}'.format(sys.argv[1], sys.argv[2], sum))
9
```

Save As

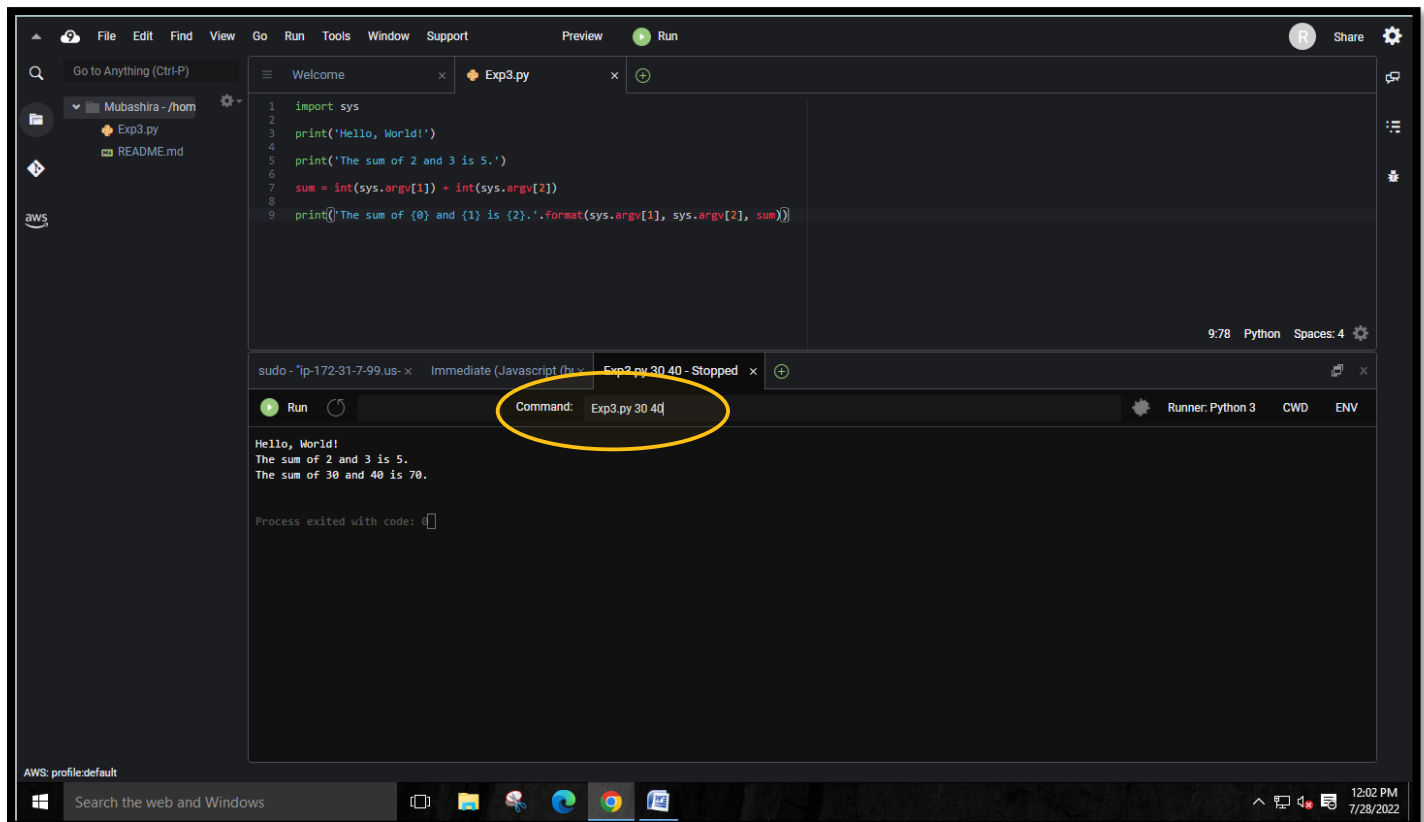
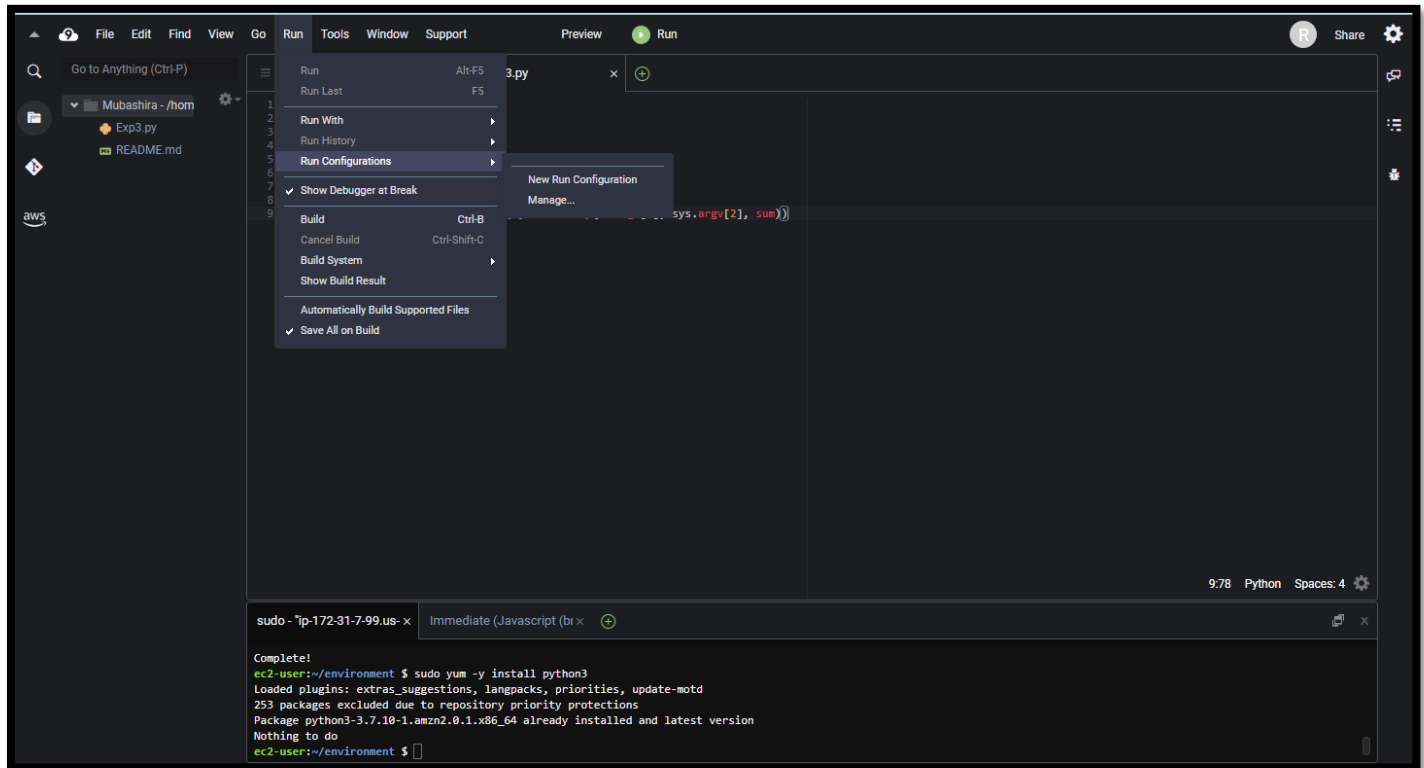
Filename: Exp3.py

Folder: /

Create folder Show files in tree Save Cancel

### Step 3: Run the code.

1. In AWS Cloud9 IDE, on the menu bar choose Run -> Run Configurations -> New Run Configuration.
2. On the [New] - Stopped tab, enter filename.py 3 5 for Command.
3. Choose Run.



## Step 4: Install and configure the AWS SDK for Python(Boto3).

1. Install pip: `sudo python3.7 get-pip.py`
2. Install the AWS SDK for Python (Boto3) - After you install pip, install the AWS SDK for Python
3. (Boto3) by running the pip install command.

```
us-west-2.console.aws.amazon.com/cloud9/ide/f75d8f320b214262a6ecfc9409152c1
File Edit Find View Go Run Tools Window Support Preview Run
Go to Anything (Ctrl-P)
Mubashira - /home
  Exp3.py
  README.md
aws
bash - "ip-172-31-7-99.us-ec2-user:~/environment $ sudo yum -y install python3
Complete!
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
253 packages excluded due to repository priority protections
Package python3-3.7.10-1.amzn2.0.1.x86_64 already installed and latest version
Nothing to do
ec2-user:~/environment $ sudo python36 -m pip install boto3
sudo: python36: command not found
ec2-user:~/environment $ sudo python37 -m pip install boto3
sudo: python37: command not found
ec2-user:~/environment $ sudo python3.7 -m pip install boto3
WARNING: Running pip install with root privileges is generally not a good idea. Try 'python3.7 -m pip install --user' instead.
Collecting boto3
  Downloading boto3-1.24.39-py3-none-any.whl (132 kB)
    | 132 kB 10.2 MB/s
Requirement already satisfied: jmespath<2.0.0,>=0.7.1 in /usr/local/lib/python3.7/site-packages (from boto3) (1.0.1)
Collecting s3transfer<0.7.0,>=0.6.0
  Downloading s3transfer-0.6.0-py3-none-any.whl (79 kB)
    | 79 kB 8.7 MB/s
Collecting botocore<1.28.0,>=1.27.39
  Downloading botocore-1.27.39-py3-none-any.whl (9.0 MB)
    | 9.0 MB 22 kB/s
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /usr/local/lib/python3.7/site-packages (from botocore<1.28.0,>=1.27.39->boto3) (2.8.2)
Requirement already satisfied: urllib3<1.27,>=1.25.4 in /usr/local/lib/python3.7/site-packages (from botocore<1.28.0,>=1.27.39->boto3) (1.26.11)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/site-packages (from python-dateutil<3.0.0,>=2.1->botocore<1.28.0,>=1.27.39->boto3) (1.16.0)
Installing collected packages: botocore, s3transfer, boto3
Attempting uninstall: botocore
  Found existing installation: botocore 1.27.37
  Uninstalling botocore-1.27.37:
    Successfully uninstalled botocore-1.27.37
Successfully installed boto3-1.24.39 botocore-1.27.39 s3transfer-0.6.0
ec2-user:~/environment $
```

## Step 5: Add AWS SDK code.

Add code that uses Amazon S3 to create a bucket, list your available buckets, and optionally delete the bucket you just created.

In the AWS Cloud9 IDE, create a file with the code content and save the file with some name.

```
File Edit Find View Go Run Tools Window Support Preview Run
Go to Anything (Ctrl-P)
Mubashira - /home
  boto3.py
  Exp3.py
  README.md
aws
23 print('Content of bucket or buckets for the demo, here is my')
24 f'{e.response['Error']['Message']}'")
25 raise
26
27 bucket.wait_until_
28 list_my_buckets(s3
29
30 if not keep_bucket
31 print('\nDelet
32 bucket.delete(
33
34 bucket.wait_un
35 list_my_bucket
36 else:
37 print('\nKeepi
38
39 def main():
40 import argparse
41
42 parser = argparse.
43 parser.add_argumen
44 parser.add_argumen
45 parser.add_argumen
46
47
48
49
50 args = parser.pars
51 s3_resource = (
52 boto3.resource
53 else boto3.res
54
55 try:
56 create_and_delete_my_bucket(s3_resource, args.bucket_name, args.keep_bucket)
57 except ClientError:
58 print('Exiting the demo.')
59
60 if __name__ == '__main__':
61 main()
```

Save As

Filename: s3.py

Mubashira - /home/ec2-user/environment

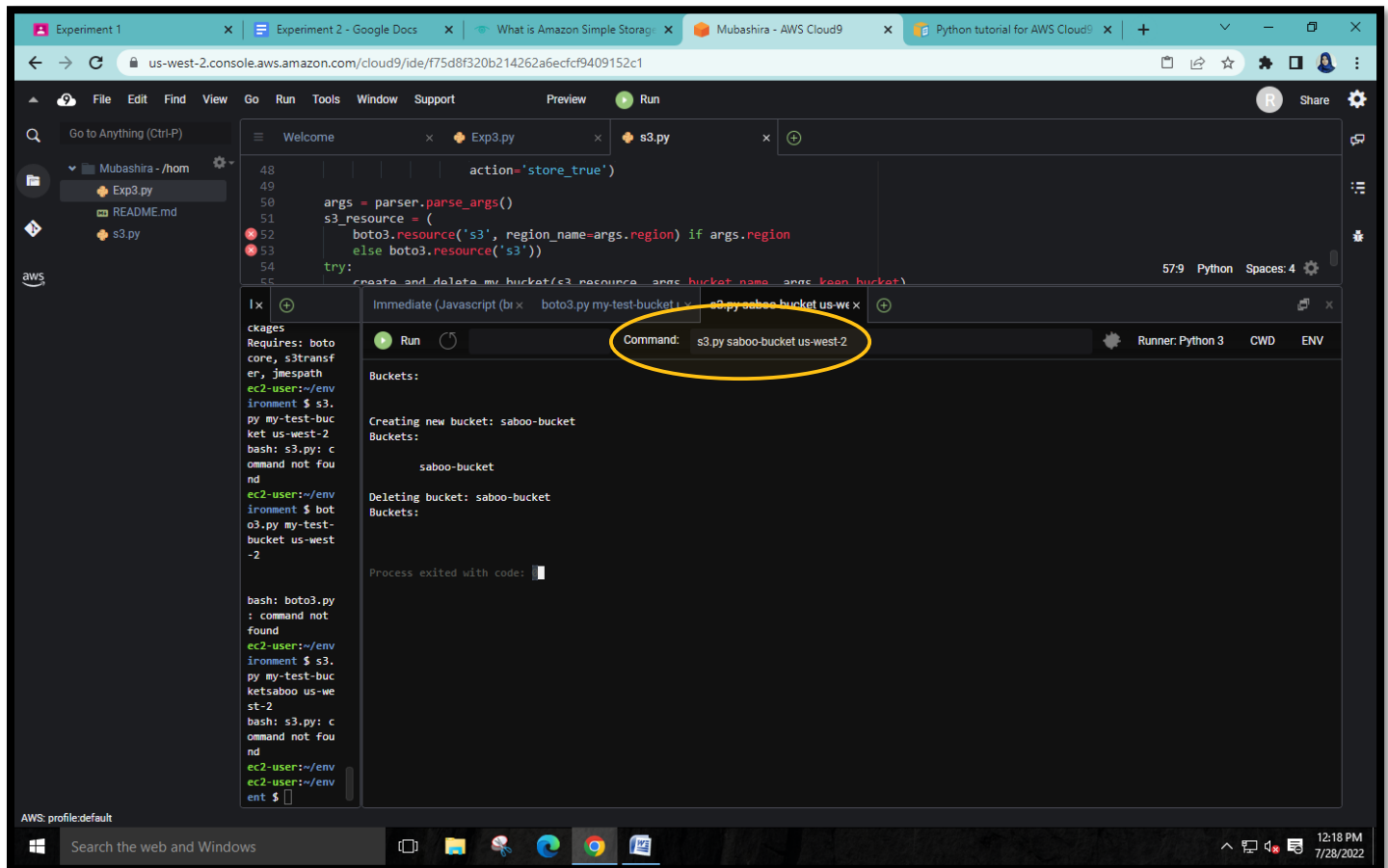
- boto3.py
- Exp3.py
- README.md

Folder: /

Create folder Show files in tree Save Cancel

### Step 6: Run the AWS SDK code.

1. On the menu bar choose Run -> Run Configurations -> New Run Configuration.
2. For Command, enter filename.py 'name of bucket' us-west-2, where us-west-2 is the ID of the AWS Region where your bucket is created. By default, your bucket is deleted before the script exits.
3. Choose Run.



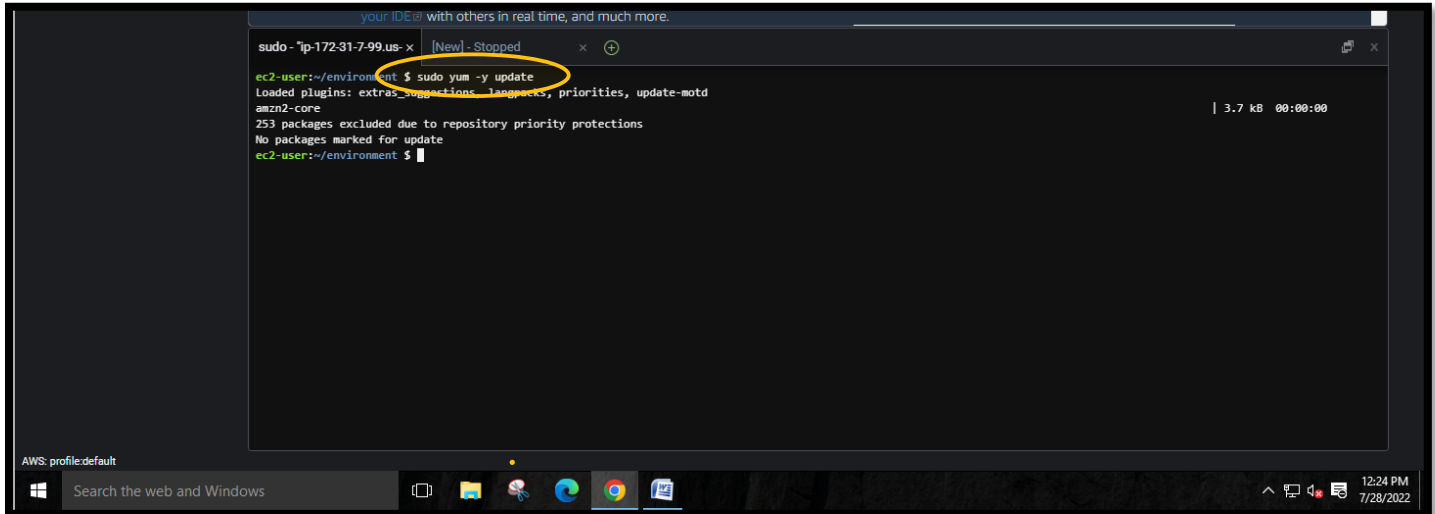
Either delete the current Cloud9 environment and create a new one to run the Node.js scripts or close all the python files and continue in the same environment.

## FOR NODE.JS:

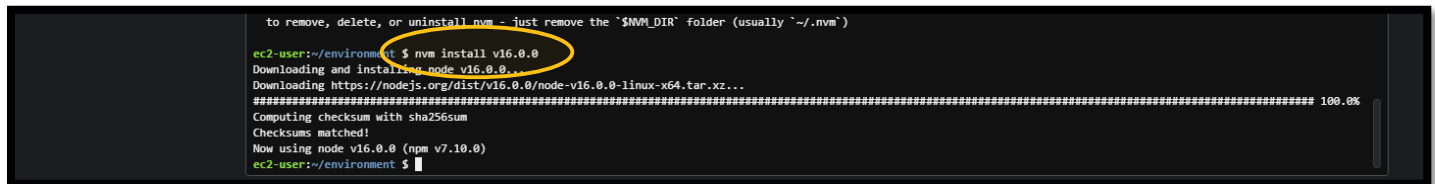
### Step 1: Install required tools.

Run the **yum update** for Amazon Linux to help ensure the latest security updates and bug fixes are installed: `sudo yum -y update`

Run this command to install Node.js – `nvm install v16.0.0`



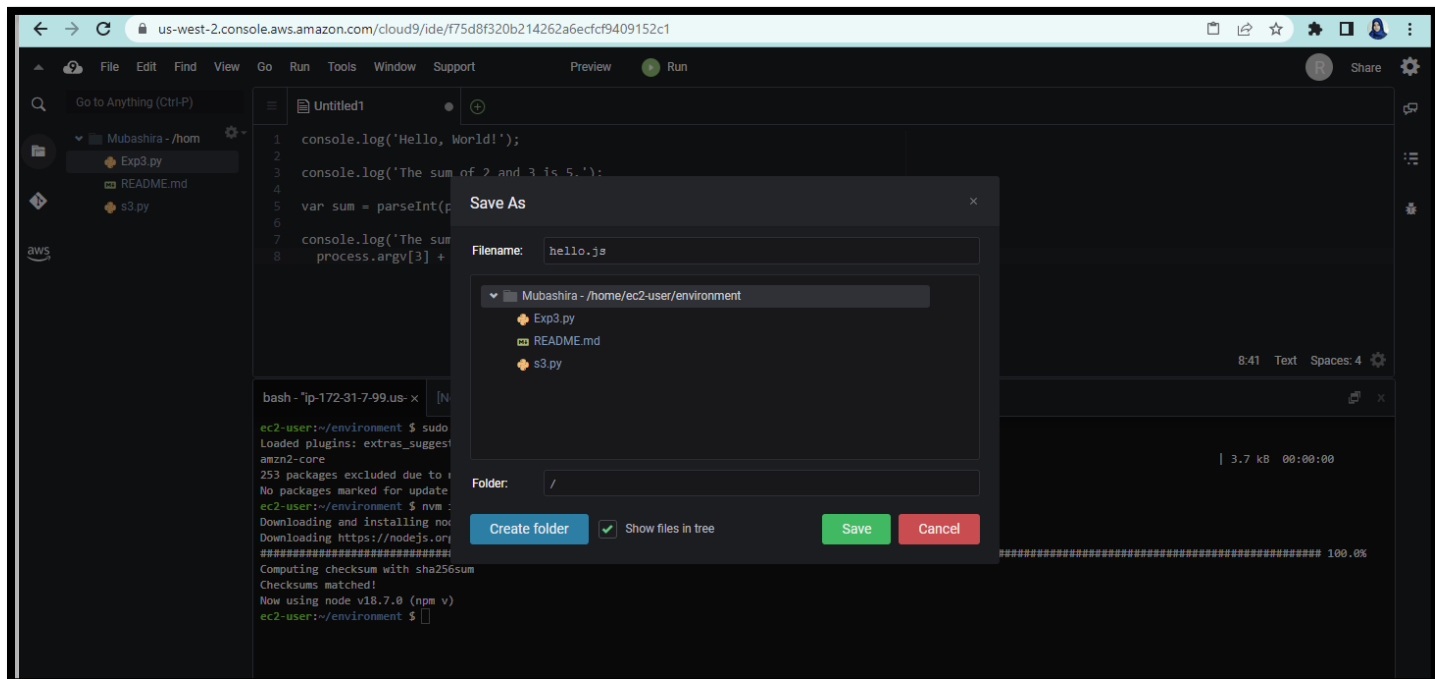
```
your IDE with others in real time, and much more.
[New] - Stopped x +
sudo - 'ip-172-31-7-99.us- x'
ec2-user:~/environment $ sudo yum -y update
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core | 3.7 kB 00:00:00
253 packages excluded due to repository priority protections
No packages marked for update
ec2-user:~/environment $
```



```
to remove, delete, or uninstall nvm - just remove the '$NVM_DIR' folder (usually '~/.nvm')
ec2-user:~/environment $ nvm install v16.0.0
Downloading and installing node v16.0.0...
Downloading https://nodejs.org/dist/v16.0.0/node-v16.0.0-linux-x64.tar.xz...
##### 100.0%
Computing checksum with sha256sum
Checksums matched!
Now using node v16.0.0 (npm v7.10.0)
ec2-user:~/environment $
```

### Step 2: Add code

In the AWS Cloud9 IDE, create a file with the *node.js* code and save the file with some name.

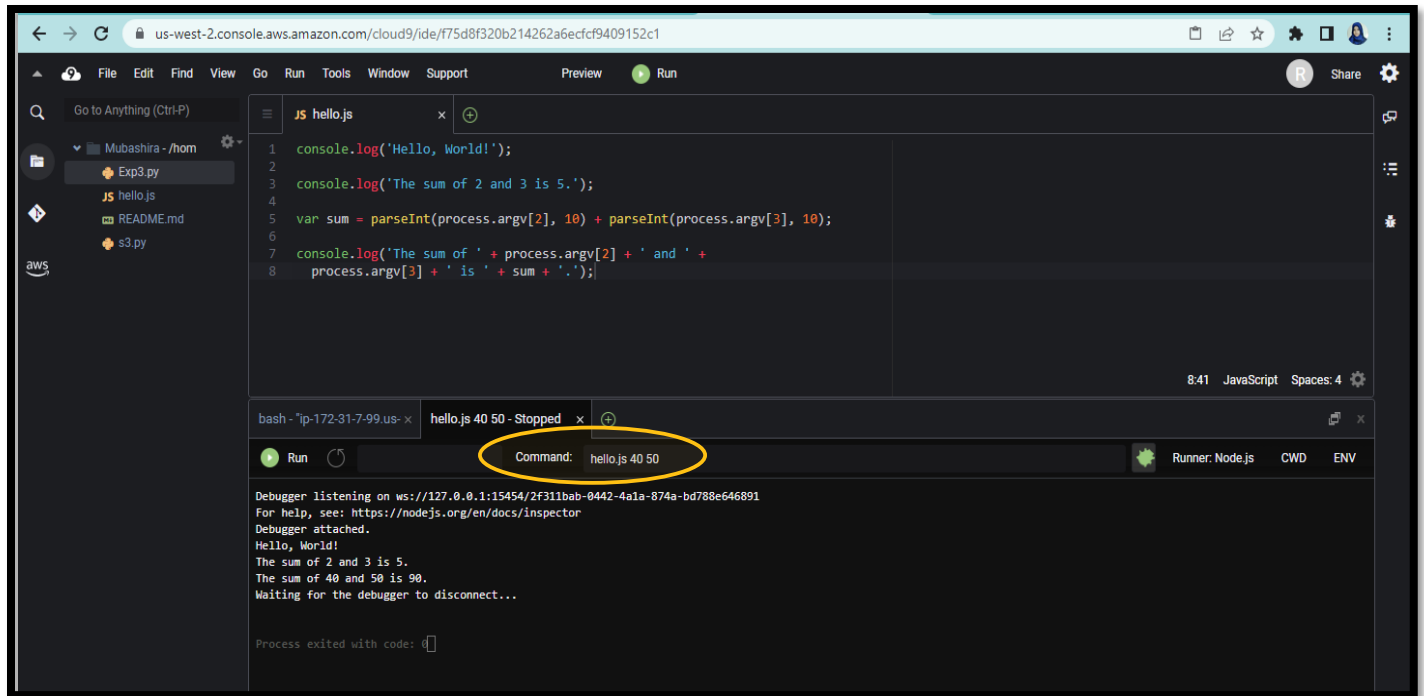


```
us-west-2.console.aws.amazon.com/cloud9/ide/f75d8f320b214262a6ecfc9409152c1
File Edit Find View Go Run Tools Window Support Preview Run Share
Go to Anything (Ctrl-P)
Untitled1
1 console.log('Hello, World!');
2
3 console.log('The sum of 2 and 3 is 5.'):
4
5 var sum = parseInt(p
6
7 console.log('The sum
8 process.argv[3] +
Save As
Filename: hello.js
Mubashira - /home/ec2-user/environment
Exp3.py
README.md
s3.py
8:41 Text Spaces: 4
3.7 kB 00:00:00
bash - 'ip-172-31-7-99.us- x' [N
ec2-user:~/environment $ sudo
Loaded plugins: extras_suggest
amzn2-core
253 packages excluded due to
No packages marked for update
ec2-user:~/environment $ nvm
Downloading and installing no
Downloading https://nodejs.org
##### 100.0%
Computing checksum with sha256sum
Checksums matched!
Now using node v18.7.0 (npm v
ec2-user:~/environment $
```



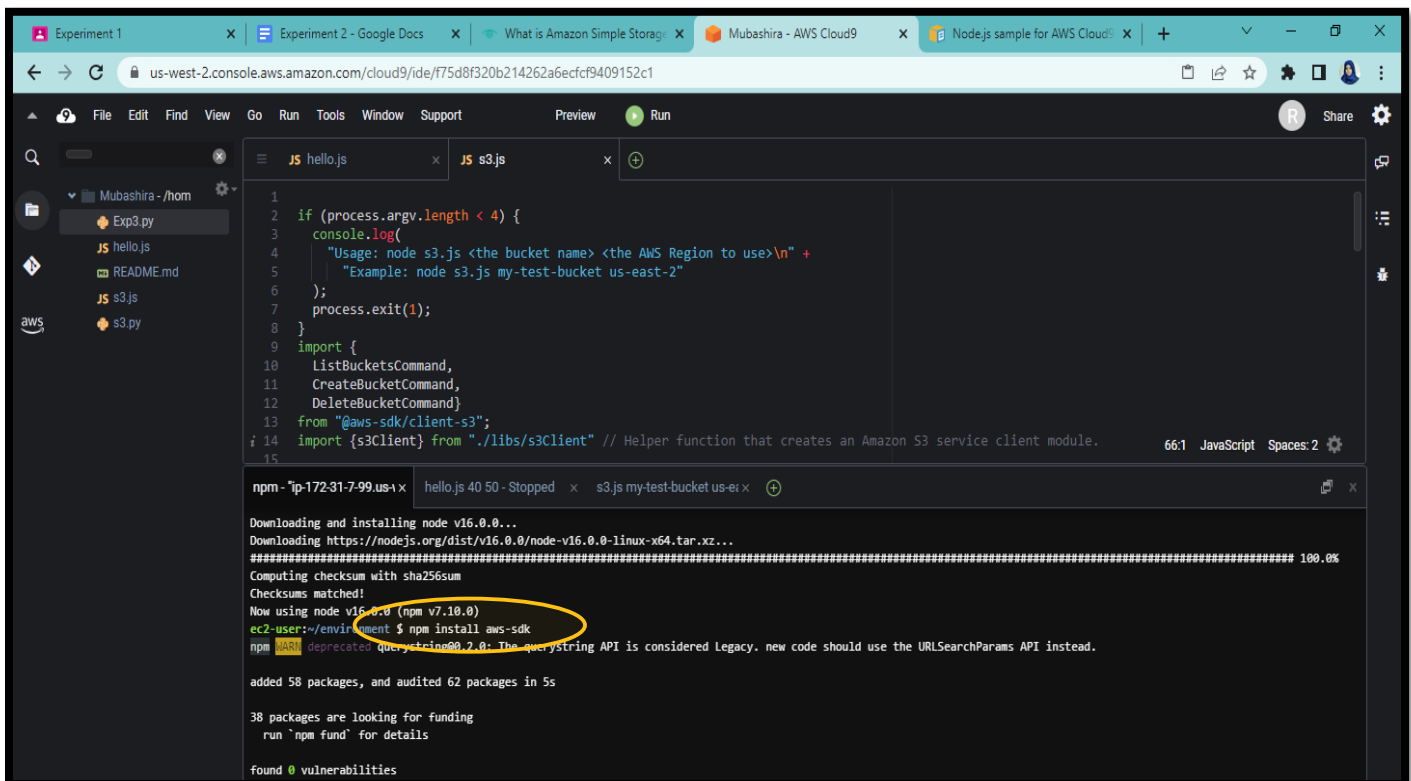
### Step 3: Run the code.

1. In AWS Cloud9 IDE, on the menu bar choose Run -> Run Configurations -> New Run Configuration.
2. On the [New] - Idle tab, enter filename.js 3 5 for Command.
3. Choose Run.



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

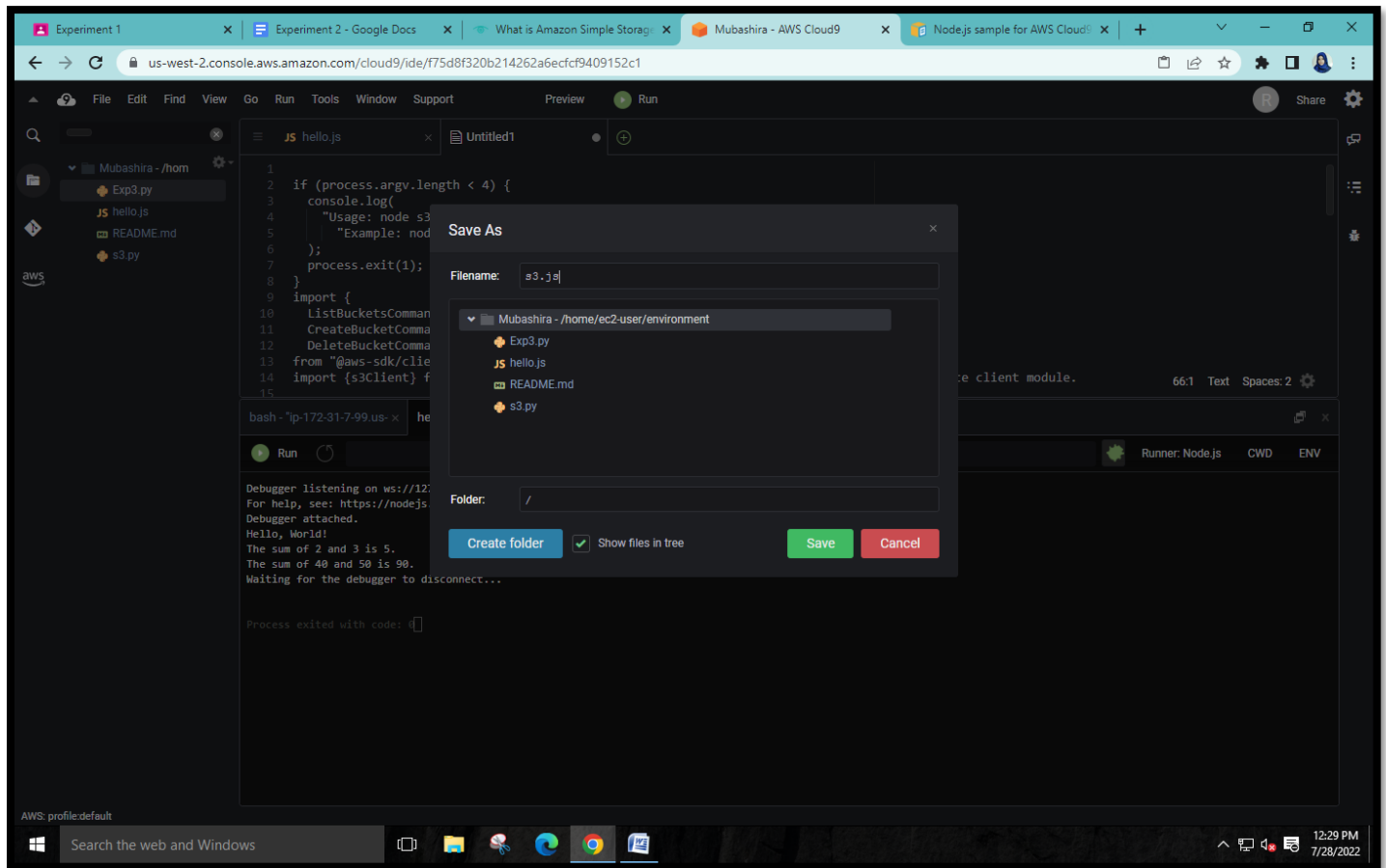
To install the AWS SDK for JavaScript(V2) in Node.js Use npm to run the install command.



### Step 5: Add AWS SDK code.

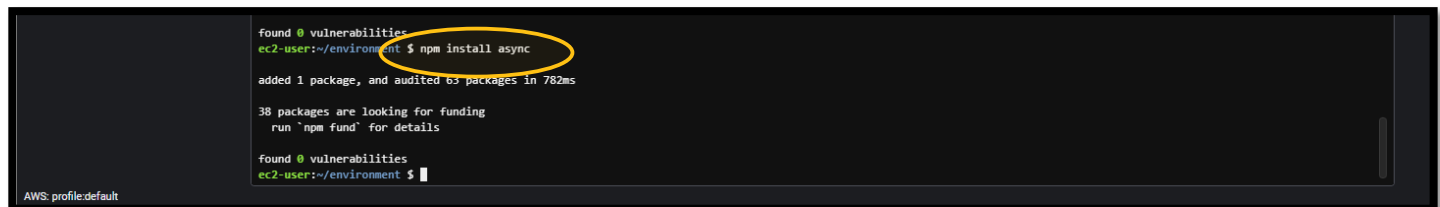
In this step, you add some more code, this time to interact with Amazon S3 to create a bucket, list your available buckets, and then delete the bucket you just created.

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

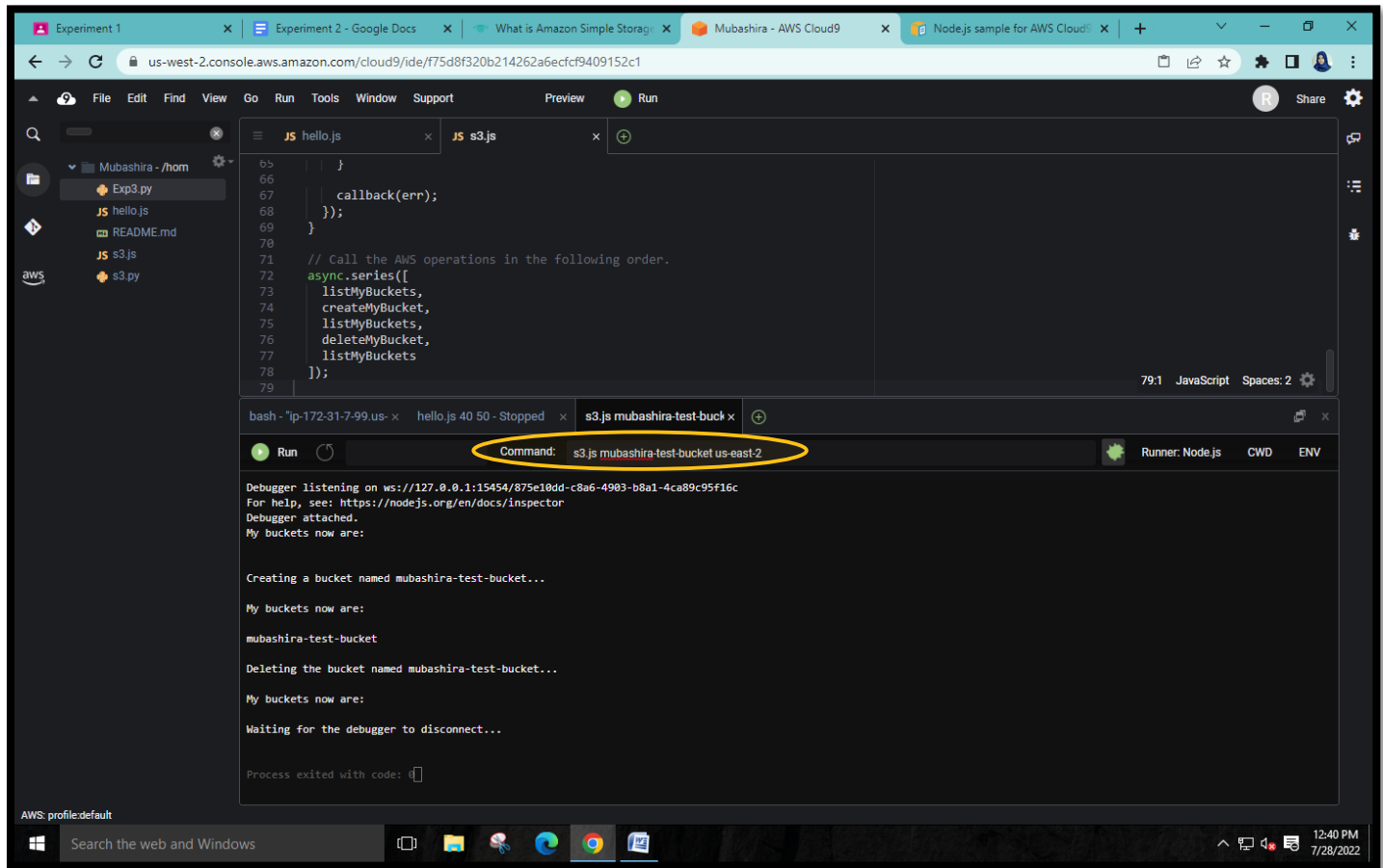


### Step 6: Run the AWS SDK code.

1. Enable the code to call Amazon S3 operations asynchronously by using npm to run the install command → `npm install async`



2. On the menu bar choose Run -> Run Configurations -> New Run Configuration.
3. For Command, type filename.js 'name of bucket' us-east-2, where useast-2 is the ID of the AWS Region you want to create the bucket in.
4. Choose Run.



Finally, close all terminals and delete the Cloud9 environment.