



lab



lab title

**Creating a Python Development Environment with
AWS Cloud9**

V1.05



Course title

**BackSpace Academy
AWS Certified Associate**



Table of Contents

Contents

Table of Contents	1
About the Lab.....	2
Creating a Cloud9 Development Environment on EC2	3
Sending Commands to a Cloud9 EC2 Instance	8
Running Code on your Cloud9 EC2 Instance.....	11
Cloning a GitHub Repository to your Cloud9 EC2 Instance	14
Editing Files.....	15
Using the Cloud9 Debugger.....	16
Cleaning Up.....	18

About the Lab

Please note that not all AWS services are supported in all regions. Please use the US-East-1 (North Virginia) region for this lab.

These lab notes are to support the AWS Cloud9 lab in the Setting Up section of the AWS Certified Developer Associate Course.

Please note that AWS services change on a weekly basis and it is extremely important you check the version number on this document to ensure you have the latest version with any updates or corrections.

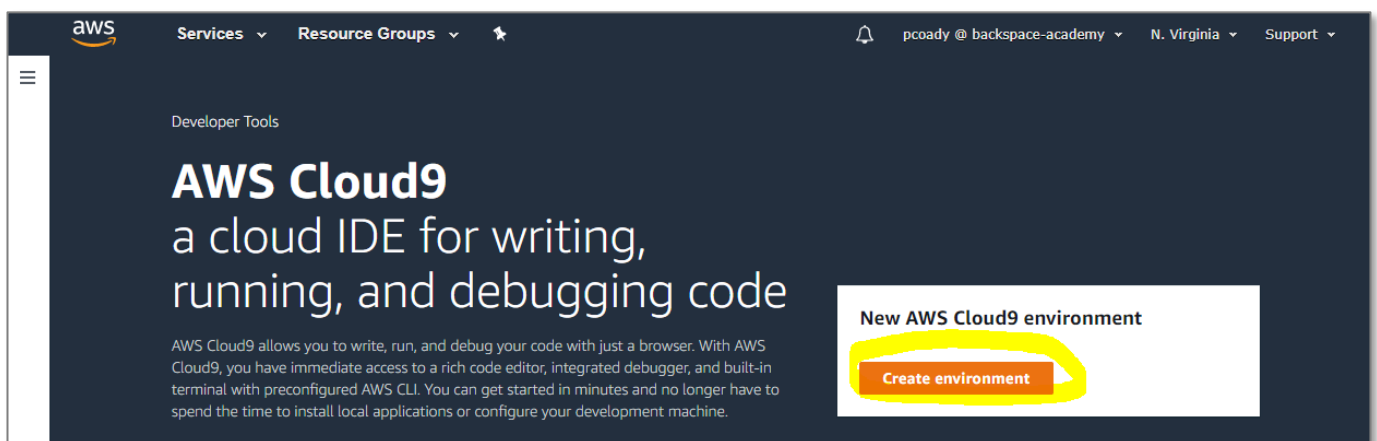
▶ Creating a Cloud9 Development Environment on EC2

In this section, we will use the Cloud9 service to create a development environment on an EC2 instance.

From the AWS console click 'Services'

Click 'Cloud9'

Click 'Create Environment'



Give your environment a unique name.

Click 'Next Step'

AWS Cloud9 X

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.

Backspace Labs - P Coady

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

Cancel **Next step**

Select EC2 environment.

Step 1 Name environment

Step 2 **Configure settings**

Step 3 Review

Environment settings

Environment type [Info](#)
Choose between creating a new EC2 instance for your new environment or connecting directly to your server over SSH.

☒ **Create a new instance for environment (EC2)**
Launch a new instance in this region to run your new environment.

☐ Connect and run in remote server (SSH)
Display instructions to connect remotely over SSH and run your new environment.

Select t2 micro to stay in the free tier

Instance type

☒ **t2.micro (1 GiB RAM + 1 vCPU)**
Free-tier eligible. Ideal for educational users and exploration.

☐ t2.small (2 GiB RAM + 1 vCPU)
Recommended for small-sized web projects.

☐ m4.large (8 GiB RAM + 2 vCPU)
Recommended for production and general-purpose development.

☐ Other instance type
Select an instance type.

t2.nano

Leave Platform as Amazon Linux. (Amazon Linux has the CLI, NodeJS and Python SDK pre-installed).

Leave hibernation setting at 30 mins

Platform

☒ Amazon Linux

☐ Ubuntu Server 18.04 LTS

Cost-saving setting
Choose a predetermined amount of time to auto-hibernate your environment and prevent unnecessary charges. We recommend a hibernation settings of half an hour of no activity to maximize savings.

After 30 minutes (default)

IAM role
AWS Cloud9 creates a service-linked role for you. This allows AWS Cloud9 to call other AWS services on your behalf. You can delete the role from the AWS IAM console once you no longer have any AWS Cloud9 environments. [Learn more](#)

AWSServiceRoleForAWSCloud9

Leave Network settings as default

Click 'Next Step'

Network settings (advanced)

Network (VPC)
Launch your EC2 instance into an existing Amazon Private Cloud (VPC) or create a new one.

vpc-72d25a0b (default) [Create new VPC](#)

Subnet
Select a range of IP addresses in your VPC to isolate EC2 resources from each other.

No preference (default subnet in any Availability Zone) [Create new subnet](#)

Cancel Previous step **Next step**

Click 'Create Environment'

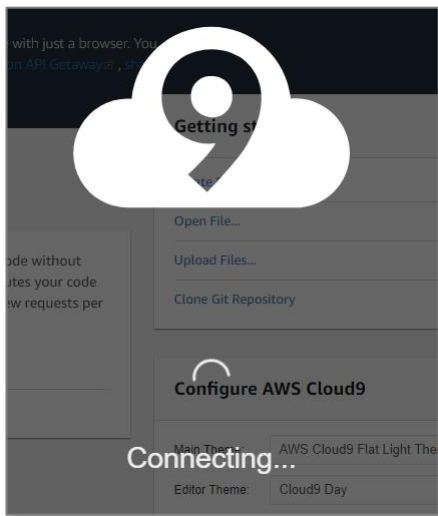
IAM role
AWSServiceRoleForAWSCloud9 (generated)

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

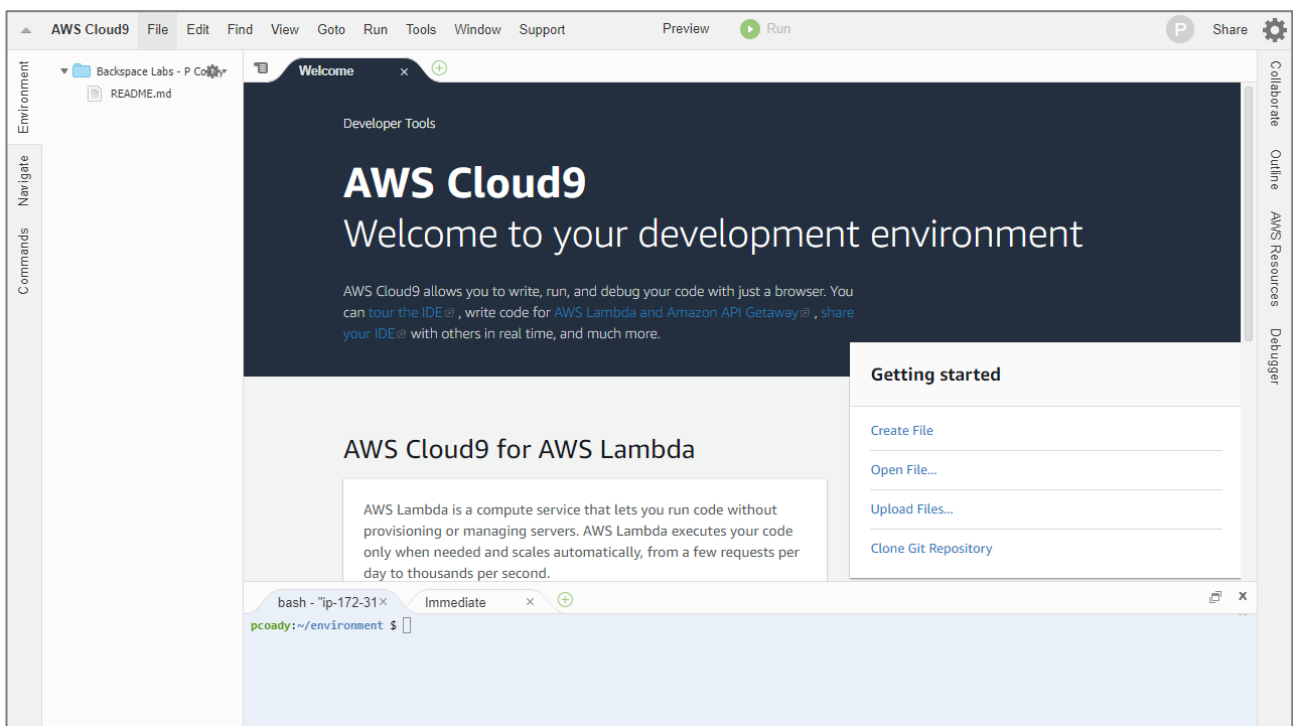
- Use **source control and backup** your environment frequently. AWS Cloud9 does not perform automatic backups.
- Perform regular **updates of software** on your environment. AWS Cloud9 does not perform automatic updates on your behalf.
- **Turn on AWS CloudTrail** in your AWS account to track activity 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**

The environment creation process will begin



After some time, your environment will be created.



You can customize the look and feel of the IDE.

Welcome to your development environment

AWS Cloud9 allows you to write, run, and debug your code with just a browser. You can [tour the IDE](#), write code for [AWS Lambda](#) and [Amazon API Gateway](#), share your IDE with others in real time, and much more.

AWS Cloud9 for AWS Lambda

AWS Lambda is a compute service that lets you run code without provisioning or managing servers. AWS Lambda executes your code only when needed and scales automatically, from a few requests per day to thousands per second.

[Create Lambda Function...](#)

[Import Lambda Function...](#)

Support

If you have any questions or experience issues, refer to our [documentation](#) or reach us to get help.

Getting started

[Create File](#)

[Open File...](#)

[Upload Files...](#)

[Clone Git Repository](#)

Configure AWS Cloud9

Main Theme: [AWS Cloud9 Classic Dark Theme](#)

Editor Theme: [Cloud9 Night](#)

Keyboard Mode: [Default](#)

[More Settings...](#)

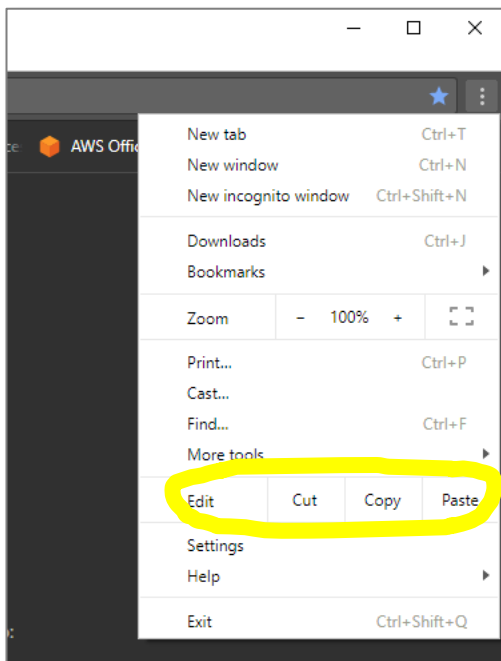
▶ Sending Commands to a Cloud9 EC2 Instance

In this section, we will use the Cloud9 service to send Linux commands to the Cloud9 EC2 instance.

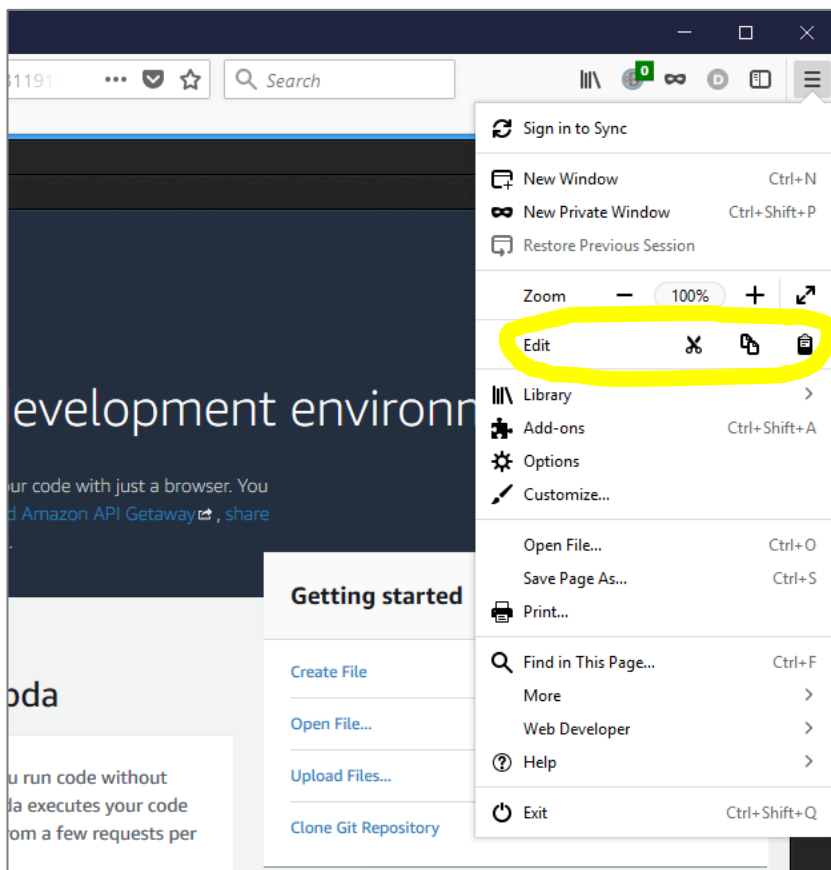
Please note:

Cut and Paste (right click or Cloud9 menu) may not work directly in Cloud9. If you cannot paste into Cloud9 then use the browser paste menu item or use ctrl-v (Windows) / cmd-v (MAC).

e.g. for Chrome:



e.g. for Firefox



At the bottom of the screen will be the Linux terminal console pane.

Check that Python is already installed.

```
python --version
```

```
bash - "ip-172-31-36" x Immediate x +
pcoady:~/environment $ python --version
Python 3.6.10
pcoady:~/environment $
```

Check that AWS Python SDK (Boto3) is already installed.

```
pip show boto3
```

```
pcoady:~/environment $ pip show boto3
Name: boto3
Version: 1.12.14
Summary: The AWS SDK for Python
Home-page: https://github.com/boto/boto3
Author: Amazon Web Services
Author-email: UNKNOWN
License: Apache License 2.0
Location: /usr/local/lib/python3.6/site-packages
Requires: jmespath, botocore, s3transfer
You are using pip version 9.0.3, however version 20.1.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
pcoady:~/environment $
```

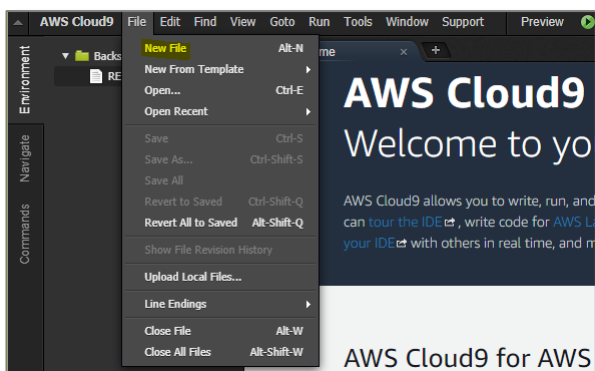
Update Amazon Linux (CentOS) operating system

```
sudo yum update -y
```

▶ Running Code on your Cloud9 EC2 Instance

In this section, we will use the Cloud9 service to run Python code on the Cloud9 EC2 instance.

Select 'File' – 'New File'

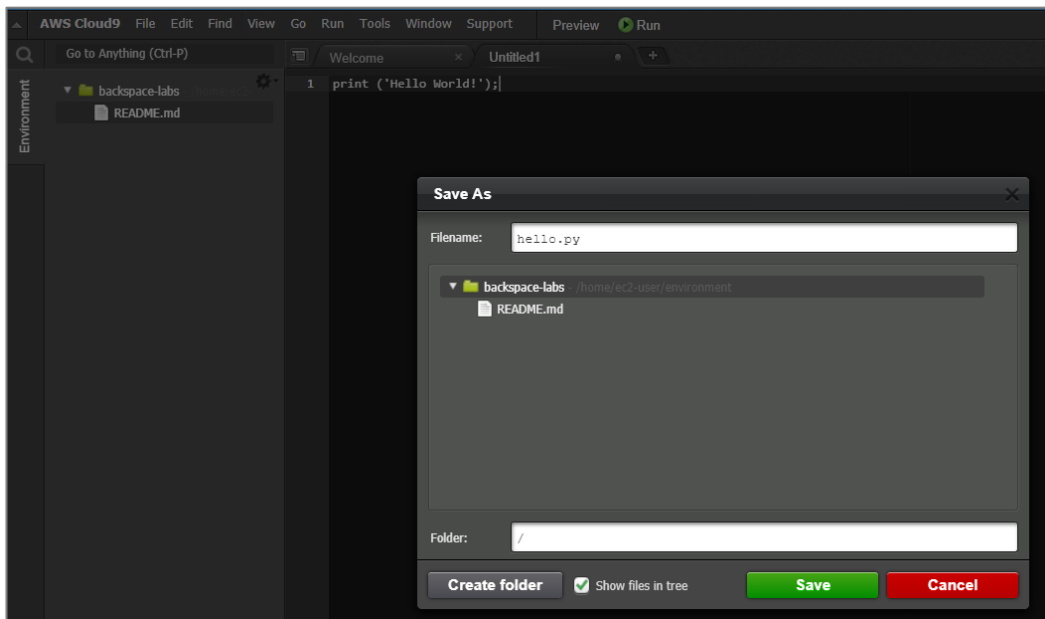


Create the classic 'Hello World ' application

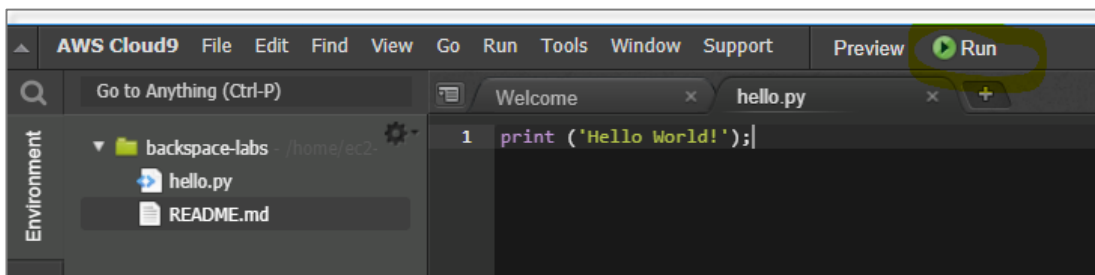
```
print ('Hello World!');
```



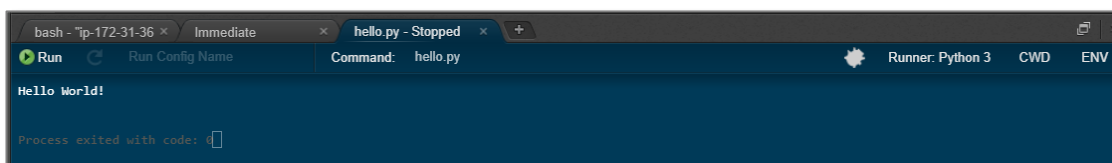
Save as 'hello.py'



Click *Run*



You can see the console output from your application



Close the application tab to remove it from the IDE.

Go back to the Linux console and remove the hello.js file

```
rm hello.py
```



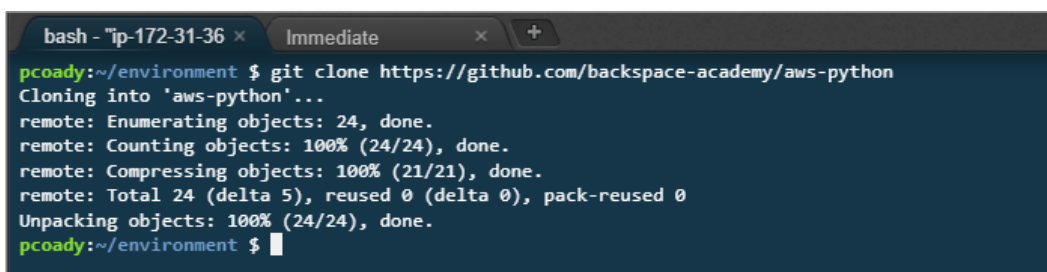
```
bash - "ip-172-31-36" x Immediate x hello.py - Stopped x +
pcoady:~/environment $
pcoady:~/environment $
pcoady:~/environment $
pcoady:~/environment $ python --version
Python 3.6.10
pcoady:~/environment $ rm hello.py
pcoady:~/environment $
```

Cloning a GitHub Repository to your Cloud9 EC2 Instance

In this section, we will use the Cloud9 service to clone a GitHub repository to the Cloud9 EC2 instance. We will then run the code in the repository.

Go back to the Linux console and clone the following sample Python app repository.

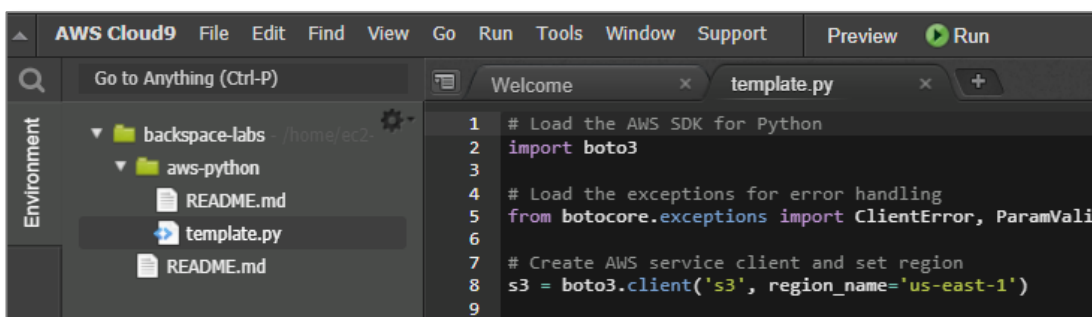
```
git clone https://github.com/backspace-academy/aws-cloud9-python
```



```
bash - "ip-172-31-36" x Immediate x +
pcoady:~/environment $ git clone https://github.com/backspace-academy/aws-python
Cloning into 'aws-python'...
remote: Enumerating objects: 24, done.
remote: Counting objects: 100% (24/24), done.
remote: Compressing objects: 100% (21/21), done.
remote: Total 24 (delta 5), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (24/24), done.
pcoady:~/environment $
```

Open *template.py*

Click *Run*



```
AWS Cloud9 File Edit Find View Go Run Tools Window Support Preview Run
Go to Anything (Ctrl-P) Welcome x template.py x +
Environment
  backspace-labs /home/ec2-
    aws-python
      README.md
      template.py
      README.md
1 # Load the AWS SDK for Python
2 import boto3
3
4 # Load the exceptions for error handling
5 from botocore.exceptions import ClientError, ParamVali
6
7 # Create AWS service client and set region
8 s3 = boto3.client('s3', region_name='us-east-1')
9
```

Note: If you have an error stating your credentials have expired, close the run console, refresh the browser, and try again.

You should see a list of buckets in your account in the run console

```

bash - "ip-172-31-36" x Immediate x aws-python/template x +
Run Run Config Name Command: aws-python/template.py Runner: Python 3 CW
Code loaded successfully
S3 Buckets in your account:
aws-codestar-us-east-1-361919435810
aws-codestar-us-east-1-361919435810-backspace-pay-pipe
aws-sam-cli-managed-default-samclisourcebucket-1j9v8850bgwvr
backspace-lab-aws-api-gateway-template
backspace-lab-lambda-images
cf-templates-1v2ku1qgvq6gr-us-east-1
cloud9-361919435810-sam-deployments-us-east-1
codepipeline-us-east-1-236973146727
codepipeline-us-east-1-317878216625
elasticbeanstalk-an-southeast-2-361919435810

```

Editing Files

Go back to the *template.py* edit window

Change the print message to 'Application loaded successfully'

Select *File > Save*

Click *Run*

The Linux terminal will show the app running again with the different message

```

20
21 # Main program
22 def main():
23     print('Application loaded successfully')
24     response = s3_list_buckets()
25     print('S3 Buckets in your account:')
26     for bucket in response['Buckets']:
27         print(f' {bucket["Name"]}')

```

```

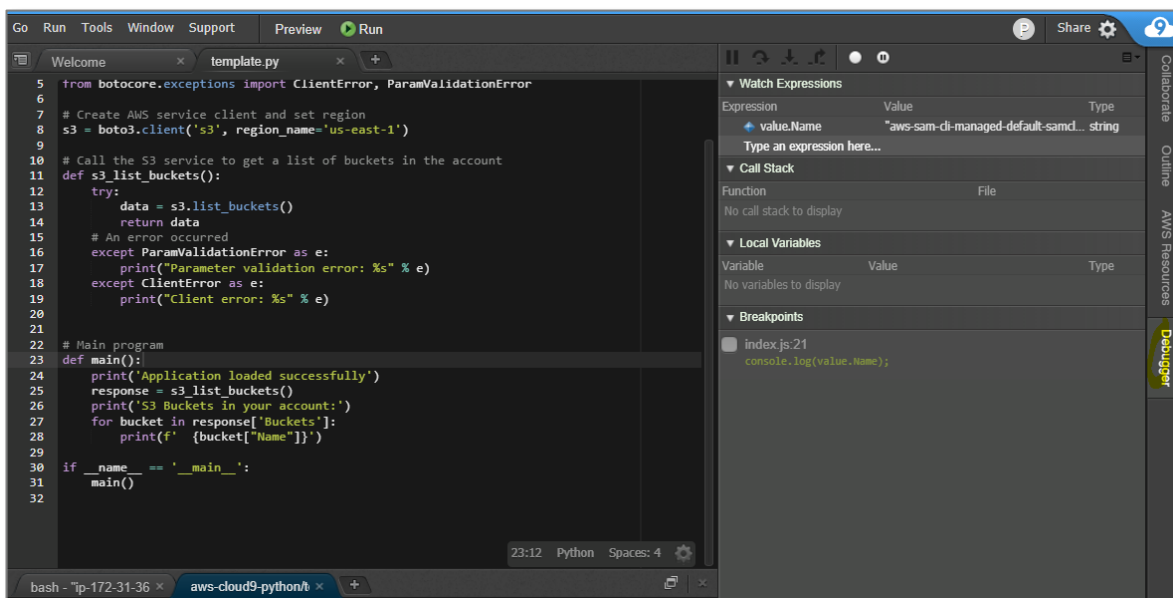
bash - "ip-172-31-36" x Immediate x aws-python/template x +
Run Run Config Name Command: aws-python/template.py
Application loaded successfully
S3 Buckets in your account:
aws-codestar-us-east-1-361919435810
aws-codestar-us-east-1-361919435810-backspace-pay-pipe
aws-sam-cli-managed-default-samclisourcebucket-1j9v8850bgwvr
backspace-lab-aws-api-gateway-template
backspace-lab-lambda-images
cf-templates-1v2ku1qgvq6gr-us-east-1
cloud9-361919435810-sam-deployments-us-east-1

```


Using the Cloud9 Debugger

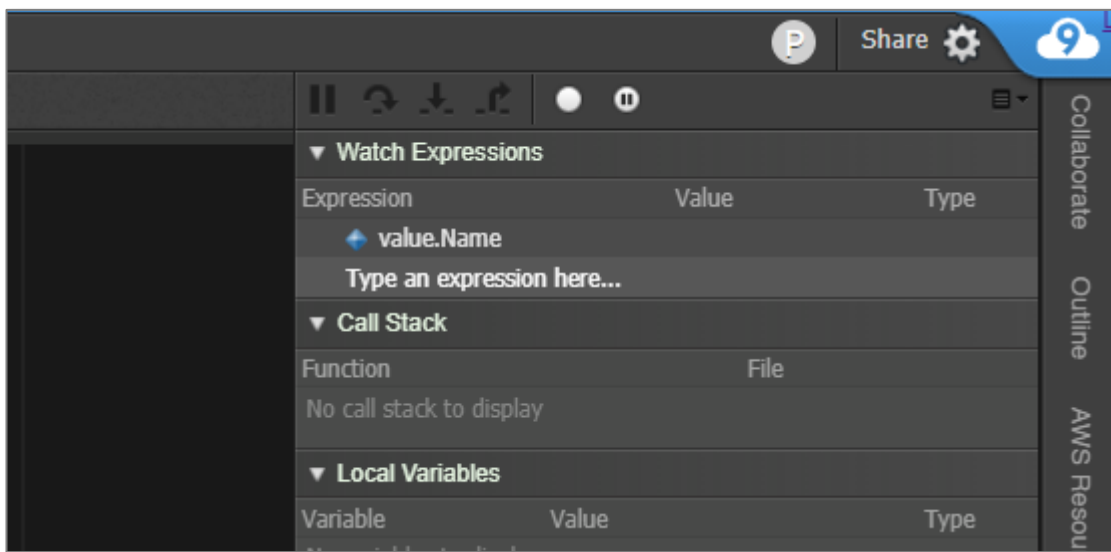
In this section, we will use the Cloud9 debugger to create breakpoints in our code and create a watch expression.

Open the debugger tab on the right hand side



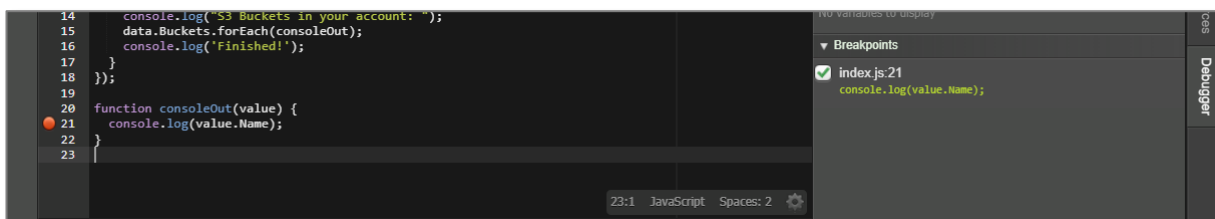
Double click on the *Watch Expressions* input field

Enter *value.Name*

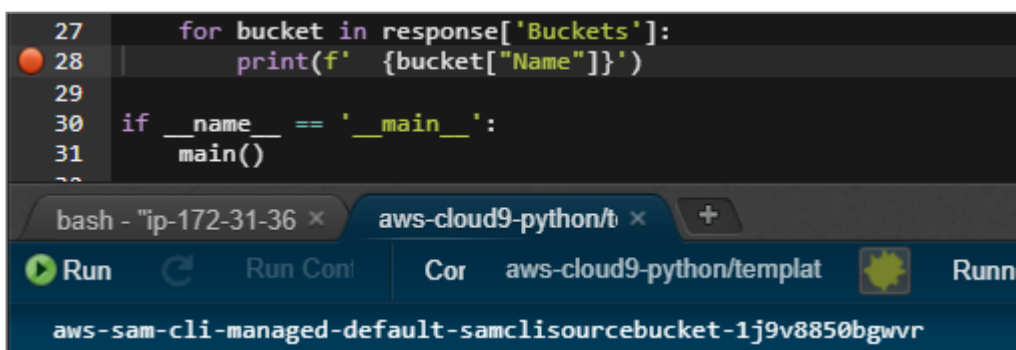


Click next to the line number for the `console.log(value.Name)`. This will create a breakpoint and pause code execution every time a bucket name is output to the console.

A red breakpoint dot will appear next to the line number and the breakpoint will be added to the breakpoints list on the left of the window.



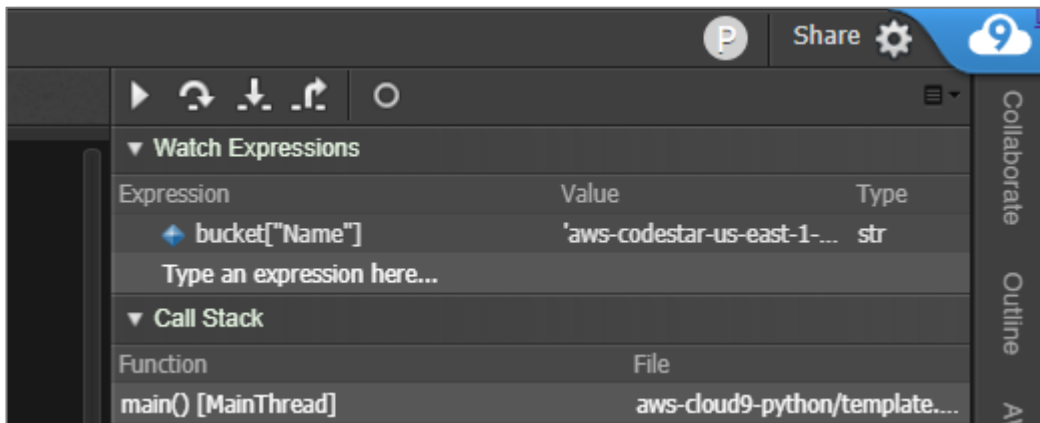
In the terminal window make sure *Run in Debug Mode*  is selected.



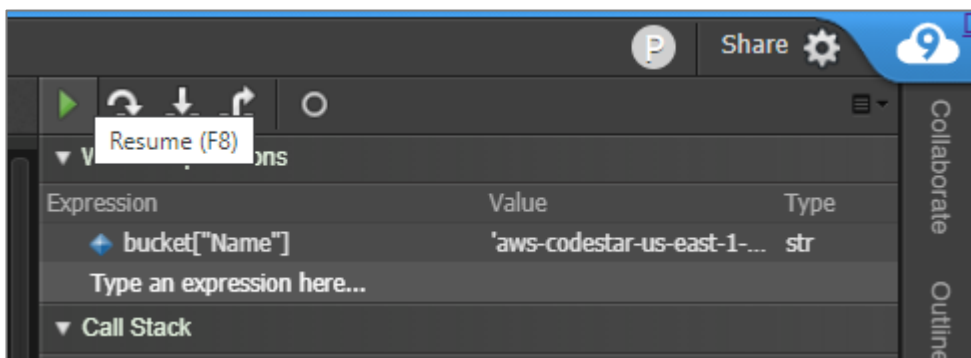
Click *Run*

Note: If you have an error stating your credentials have expired or invalid, refresh the browser to renew the security token, and try again.

The code will run and pause at the breakpoint. The watch expression will have the value of the first basket in the list.

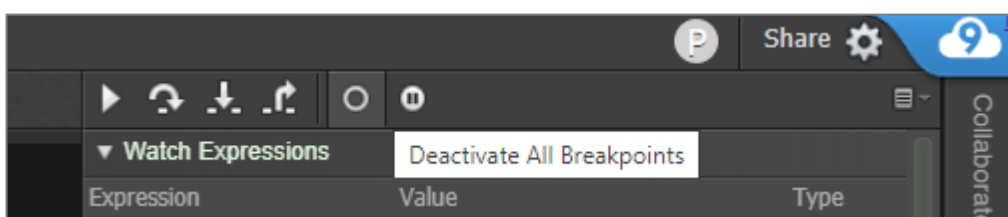


Click the resume button to keep running.



The watch expression value will change.

Click *Deactivate All Breakpoints* to remove the breakpoint.



Click *Resume* again to finish running the code.

Cleaning Up

Your Cloud 9 EC2 development environment will automatically go into hibernation after 30 mins of inactivity. You will not need to terminate it to save costs.

You should clean up the code in environment ready for the next lab by deleting the watch expression, breakpoints and the cloned repository.

