lab

lab title

Creating a NodeJS Development Environment with AWS Cloud9

V1.06

Course title

BackSpace Academy
AWS Certified Associate

Table of Contents

Contents

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 lastest version with any updates or corrections.

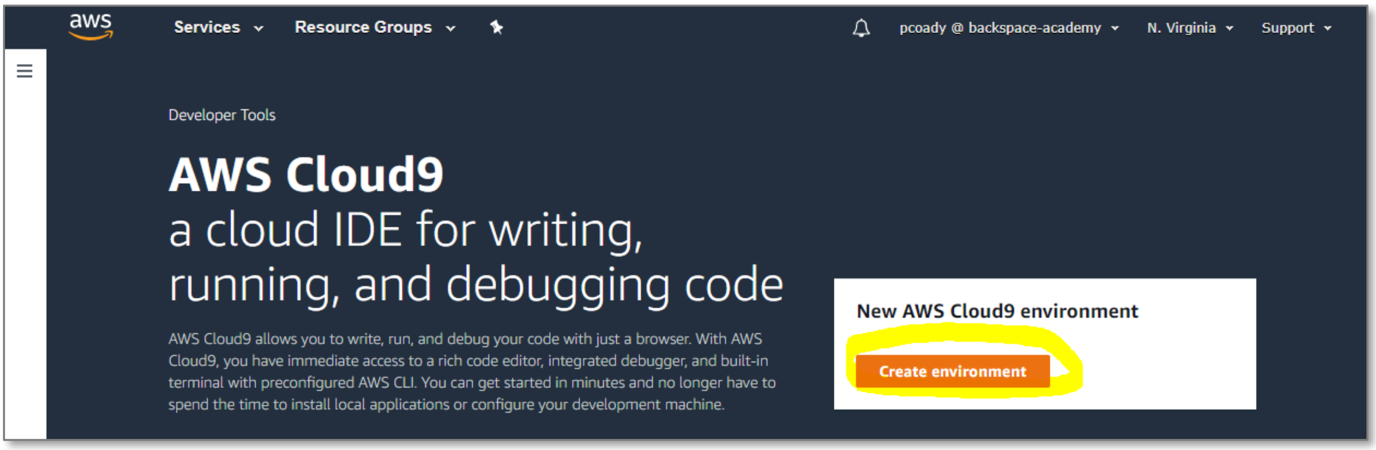
Creating a Cloud9 Development Environment on EC2

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"



Developer Tools

AWS Cloud9

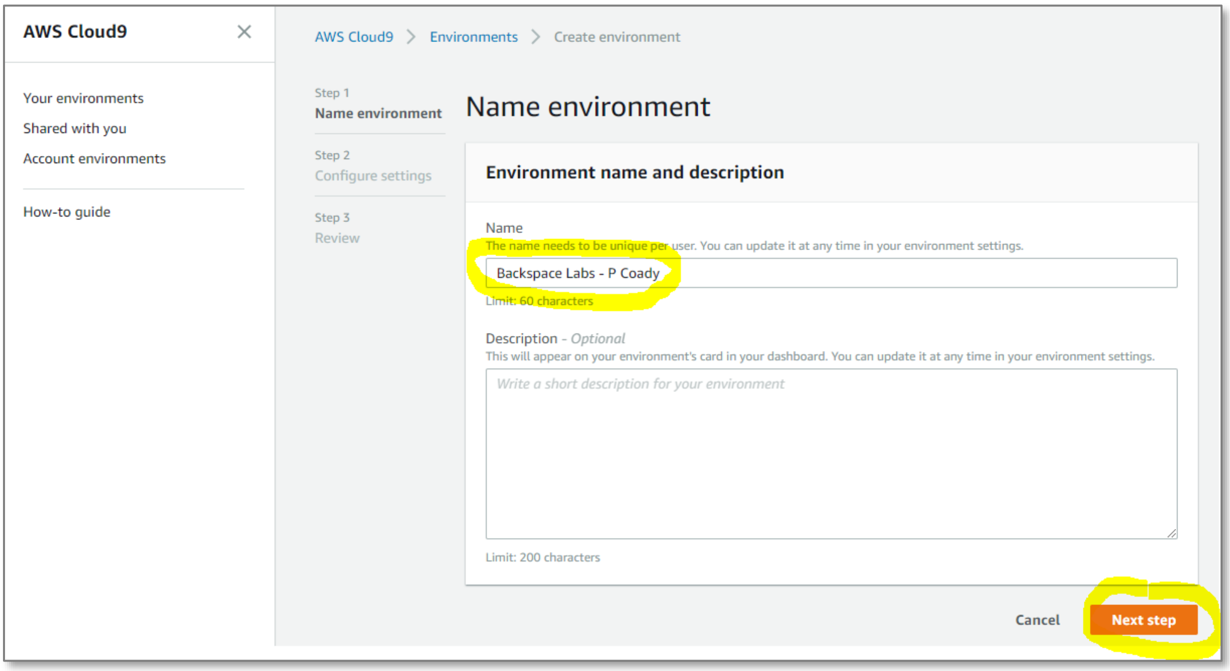
a cloud IDE for writing, running, and debugging code

AWS Cloud9 allows you to write, run, and debug your code with just a browser. With AWS Cloud9, you have immediate access to a rich code editor, integrated debugger, and built-in terminal with preconfigured AWS CLI. You can get started in minutes and no longer have to spend the time to install local applications or configure your development machine.

New AWS Cloud9 environment

Give your environment a unique name.

Click 'Next Step"



Name environment

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

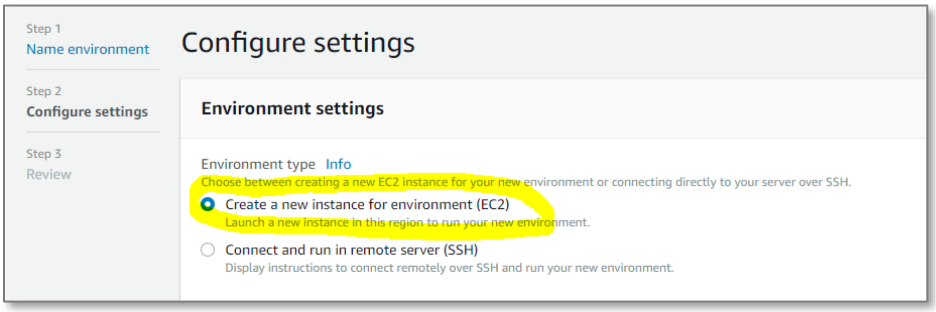
Limit: 60 characters

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

Select EC2 environment.



Configure settings

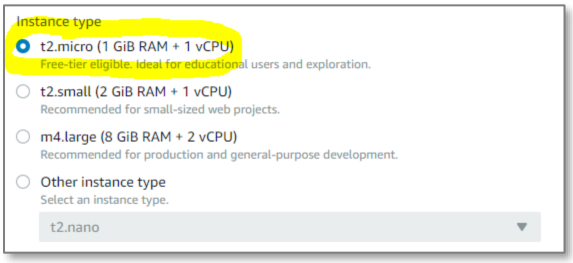
Environment settings

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.

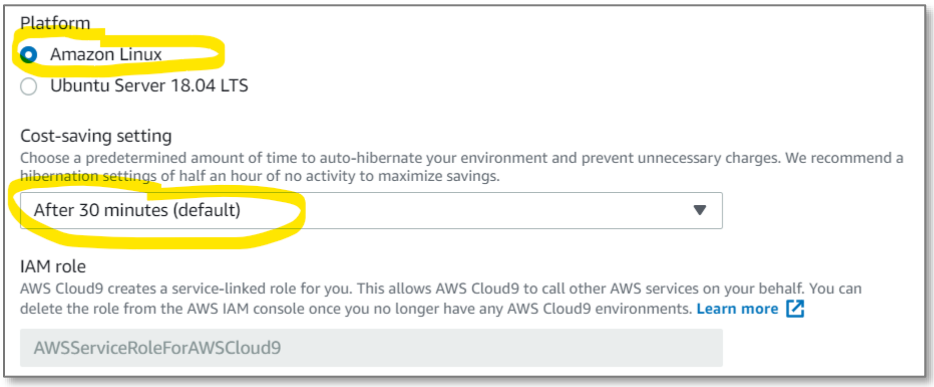
○ m4.large (8 GiB RAM + 2 vCPU)
Recommended for production and general-purpose development.

○ 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)

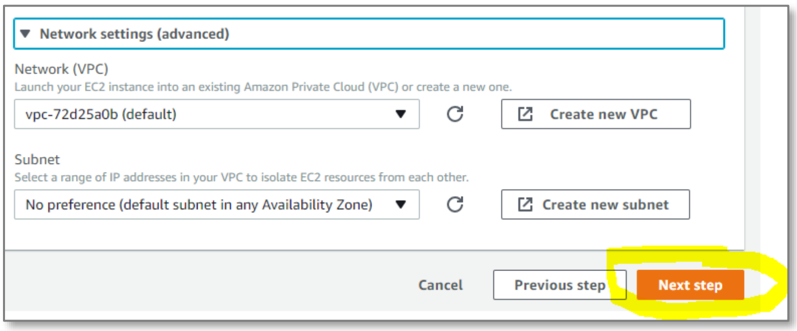
IAM role

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.

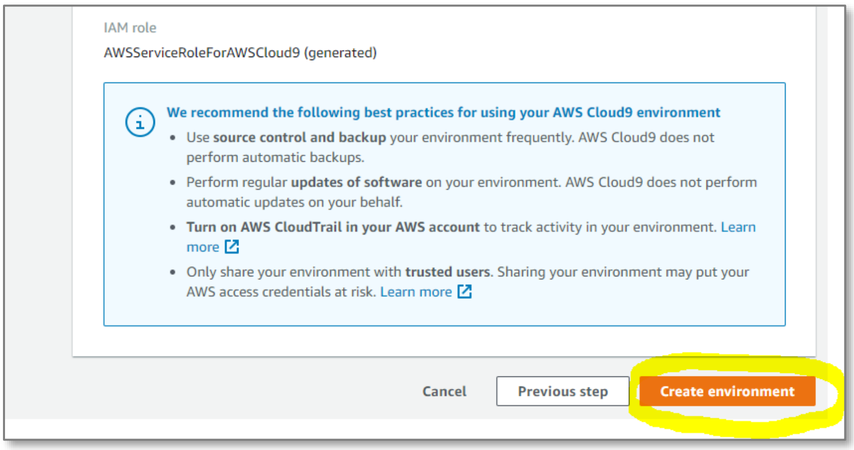
vpc-72d25a0b (default)

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)

Click 'Create Environment"



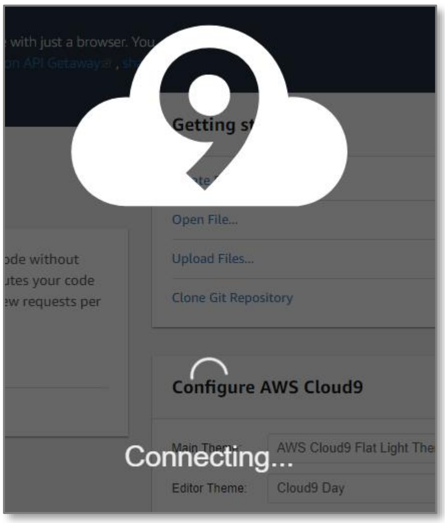
IAM role

AWSServiceRoleForAWSCloud9 (generated)

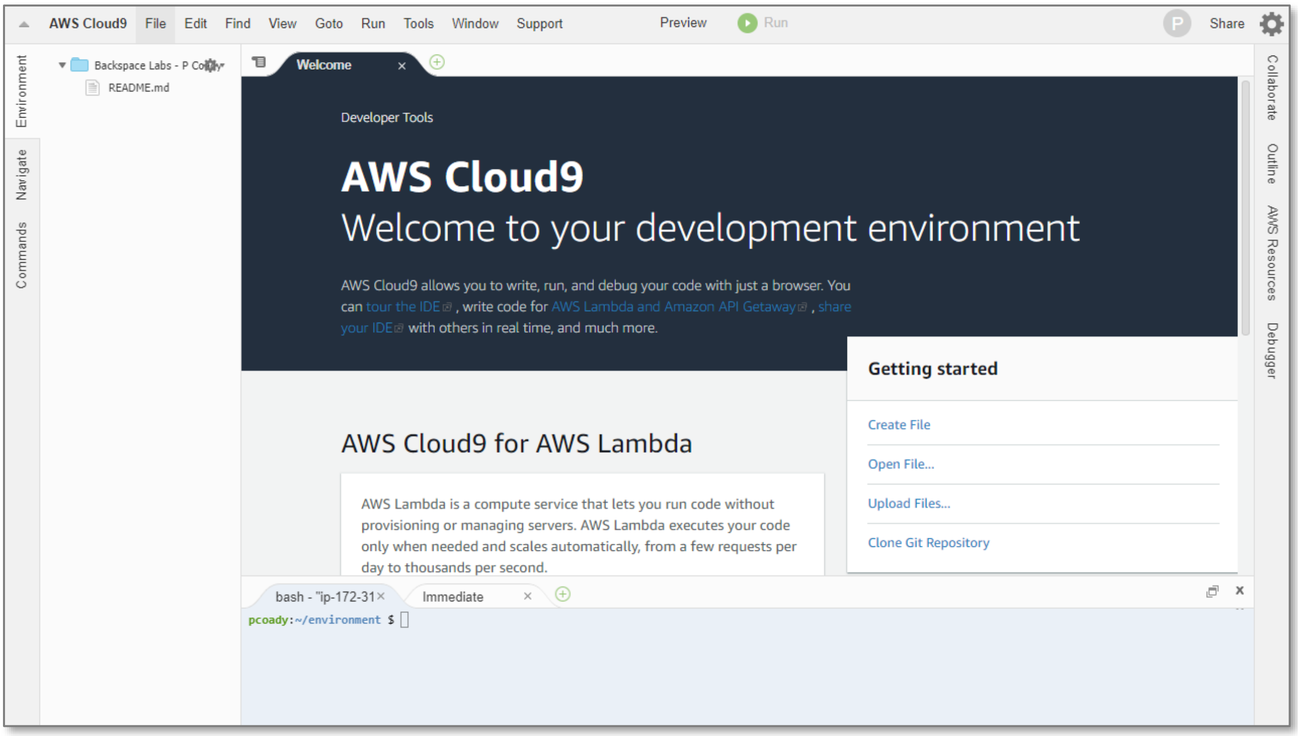
We recommend the following best practices for using your AWS Cloud9 environment

- 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

The environment creation process will begin



After some time, your environment will be created.



Developer Tools

AWS Cloud9

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.

Getting started

Create File

Open File...

Upload Files...

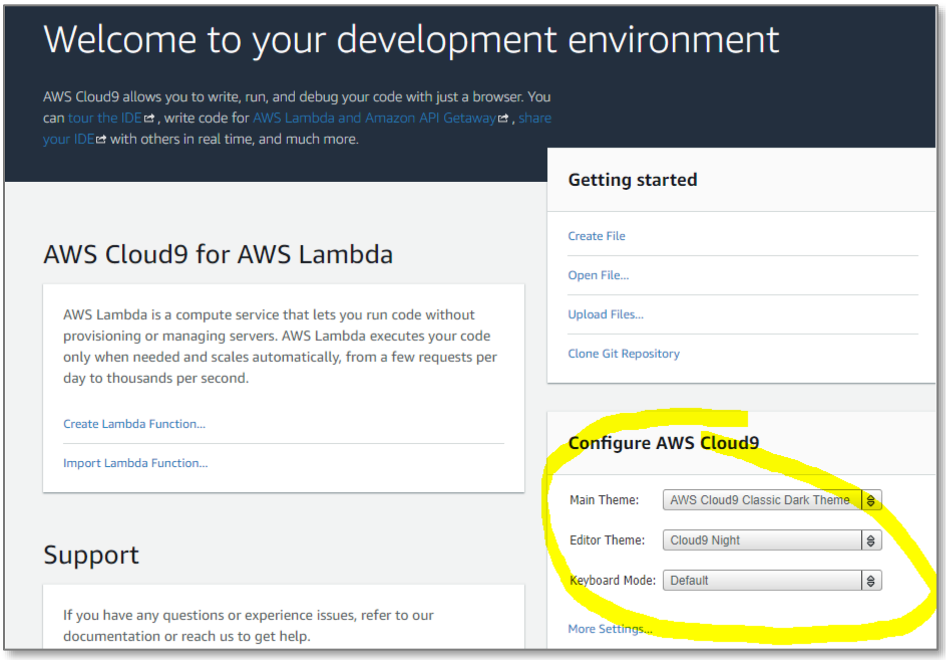
Clone Git Repository

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.

```
pcoady:~/environment $
```

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 Getaway, 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.

Support

If you have any questions or experience issues, refer to our documentation or reach us to get help.

Getting started

Configure AWS Cloud9

Main Theme: AWS Cloud9 Classic Dark Theme

Editor Theme: Cloud9 Night

Keyboard Mode: Default

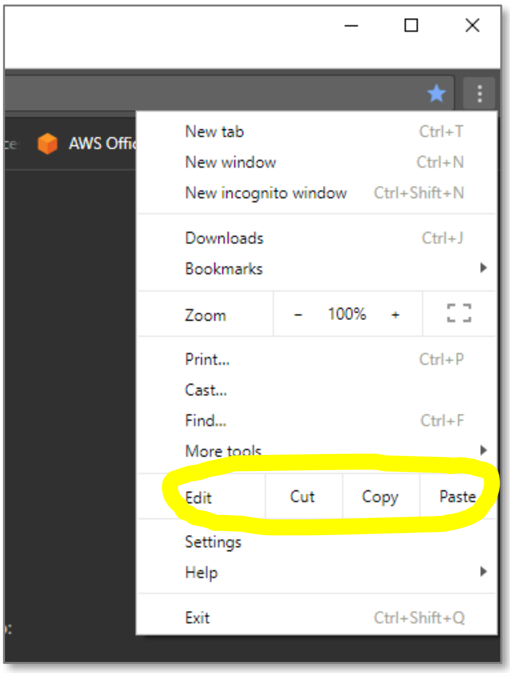
Sending Commands to a Cloud9 EC2 Instance

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

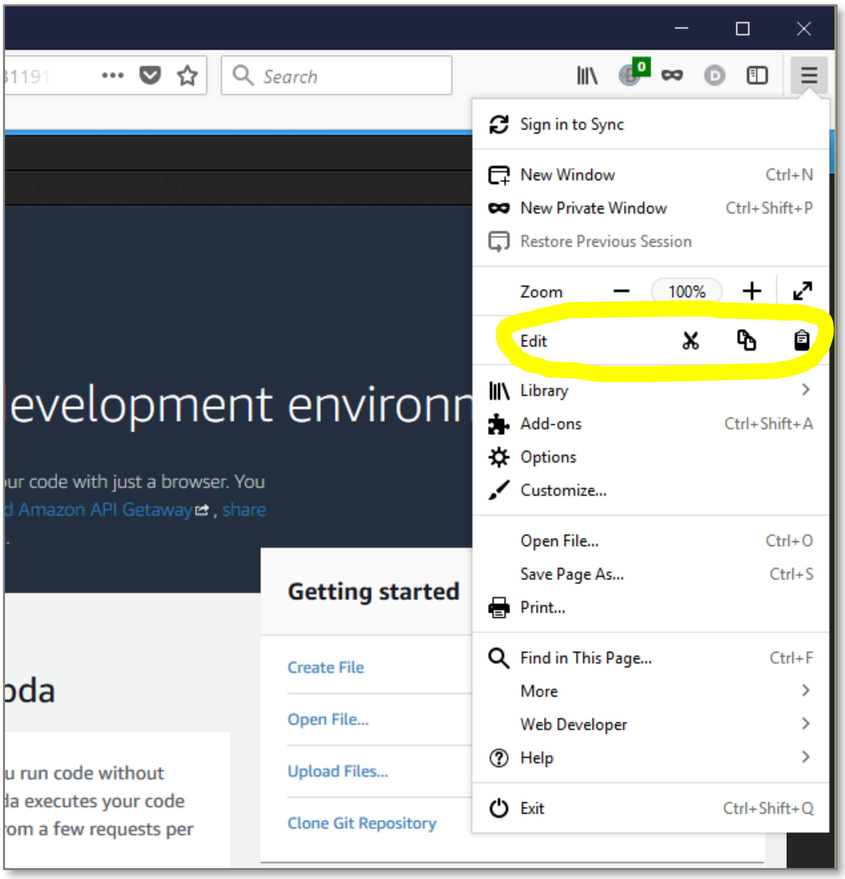
Please note:

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:



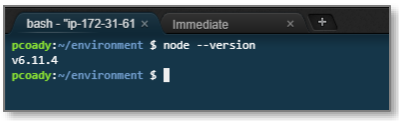
e.g. for Firefox



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

Check that NodeJS is already installed.

```
node --version
```



```
pcoady:~/environment $ node --version
v6.11.4
pcoady:~/environment $
```

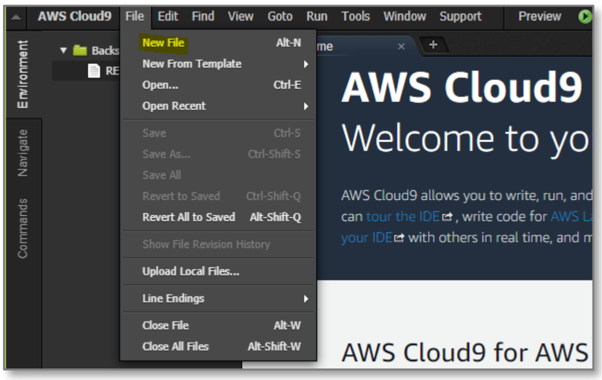
Update Amazon Linux (CentOS) operating system

```
sudo yum update -y
```

Running Code on your Cloud9 EC2 Instance

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

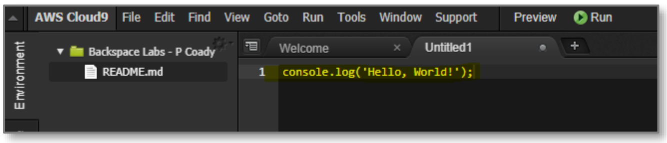
Select ‘File” – ‘New File”



AWS Cloud9 allows you to write, run, and
can tour the IDE, write code for AWS L
your IDE with others in real time, and m

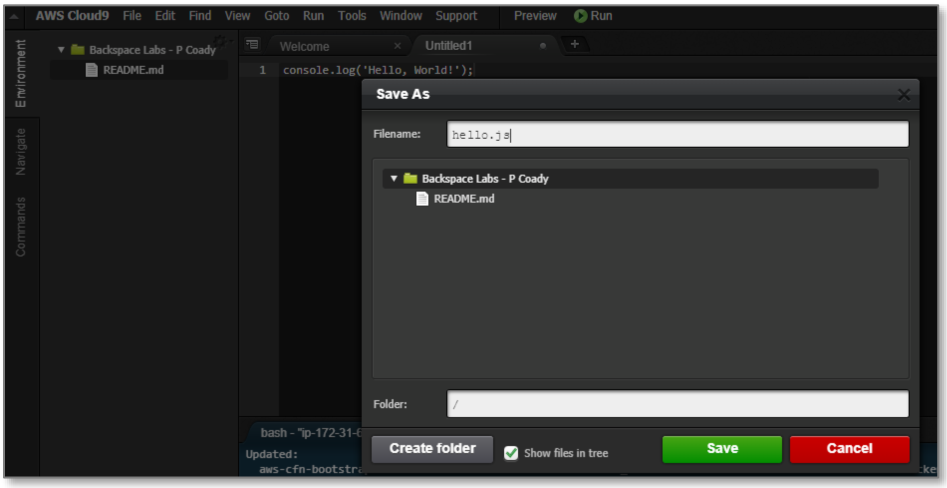
Create the classic ‘Hello World ’ application

```
console.log('Hello World!');
```



```
console.log('Hello, World!');
```

Save as ‘hello.js”

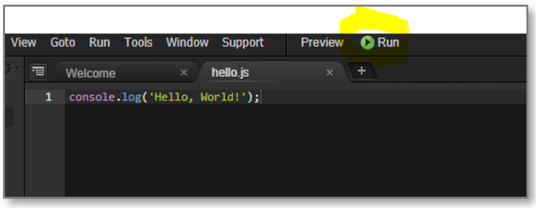


```
console.log('Hello, World!');
```

Filename: hello.js

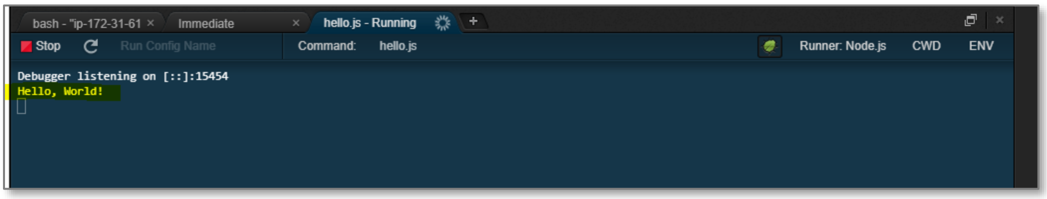
Folder: /

Click ‘Run”



```
console.log('Hello, World!');
```

You can see the console output from your application

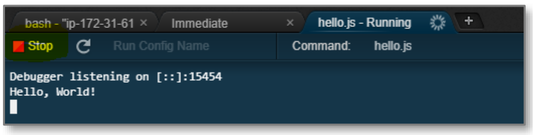


Command: hello.js

Runner: Node.js

```
Debugger listening on [::]:15454
Hello, World!
```

Click ‘Stop” to stop the application



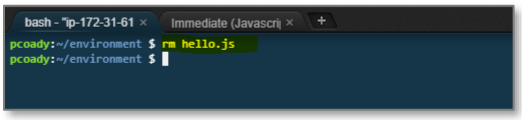
Command: hello.js

```
Debugger listening on [::]:15454
Hello, World!
```

Close the application tab to remove it from the IDE.

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

```
rm hello.js
```



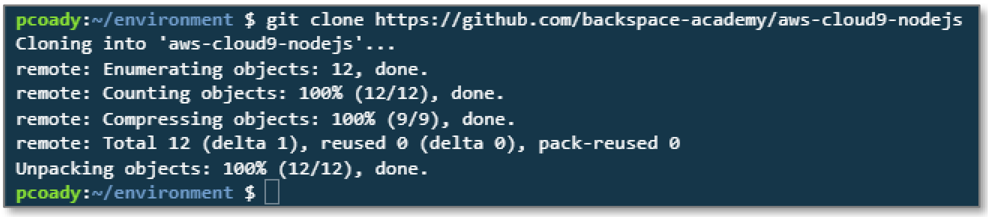
```
pcoady:~/environment $ rm hello.js
pcoady:~/environment $
```

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 NodeJS app repository.

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



```
pcoady:~/environment $ git clone https://github.com/backspace-academy/aws-cloud9-nodejs
Cloning into 'aws-cloud9-nodejs'...
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 12 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (12/12), done.
pcoady:~/environment $
```

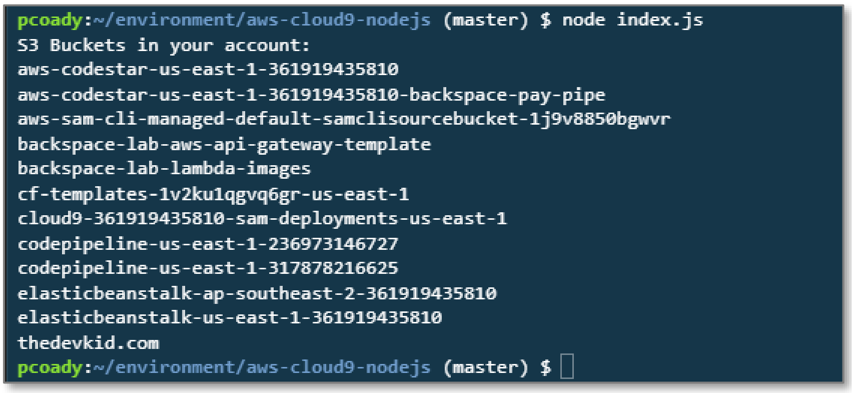
Install dependencies listed in package.json (the AWS SDK).

```
cd aws-cloud9-nodejs
npm install
```

Click *Run* to run index.js in NodeJS.

You should see a list of the S3 buckets in your account

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



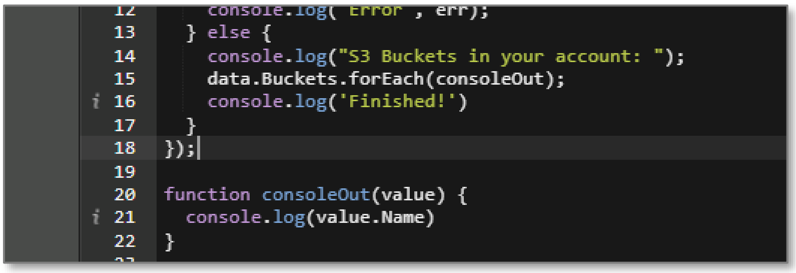
```
pcoady:~/environment/aws-cloud9-nodejs (master) $ node index.js
S3 Buckets in your account:
aws-codestar-us-east-1-361919435810
aws-codestar-us-east-1-361919435810-backspace-pay-pipe
aws-sam-cli-managed-default-samclisourcebucket-1j9v8850bgwvr
backspace-lab-aws-api-gateway-template
backspace-lab-lambda-images
cf-templates-1v2ku1qgvq6gr-us-east-1
cloud9-361919435810-sam-deployments-us-east-1
codepipeline-us-east-1-236973146727
codepipeline-us-east-1-317878216625
elasticbeanstalk-ap-southeast-2-361919435810
elasticbeanstalk-us-east-1-361919435810
thedevkid.com
pcoady:~/environment/aws-cloud9-nodejs (master) $
```

Editing Files

Double click ‘index.js” to open it for editing.

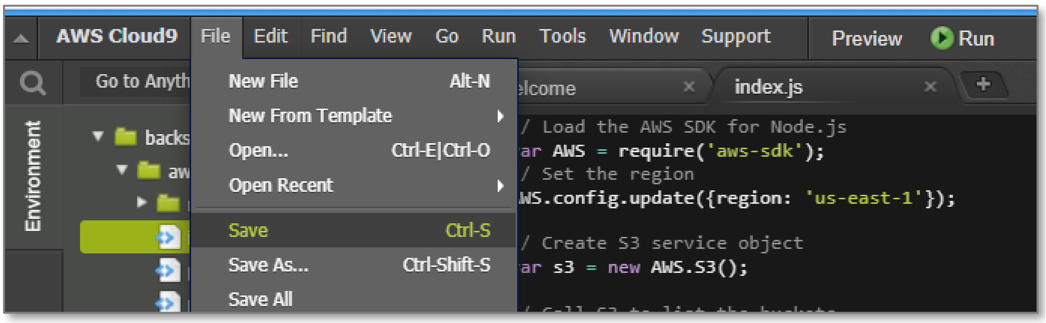
Add another console.log message

console.log('Finished!')



```
      console.log("Error", err);
  } else {
      console.log("S3 Buckets in your account: ");
      data.Buckets.forEach(consoleOut);
      console.log('Finished!')
  }
});

function consoleOut(value) {
  console.log(value.Name)
}
```

File > Save

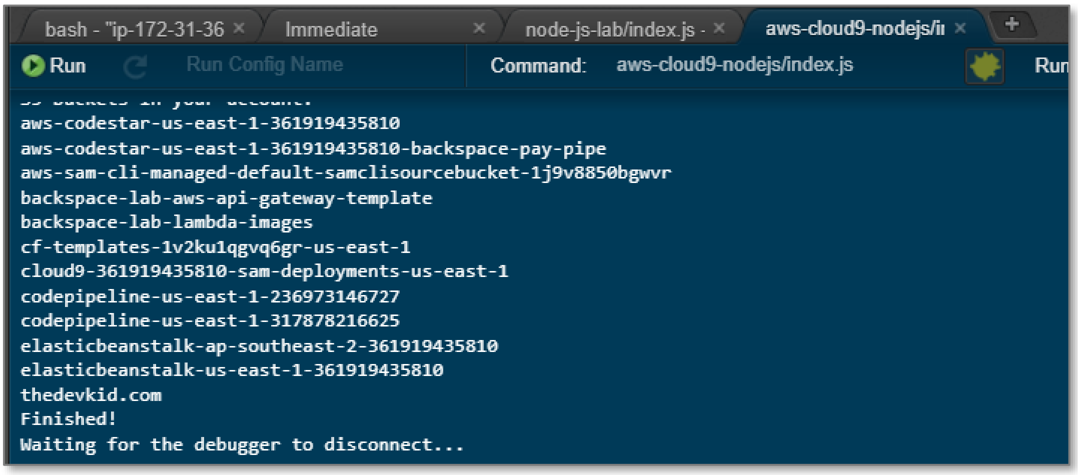


```
/ Load the AWS SDK for Node.js
ar AWS = require('aws-sdk');
/ Set the region
WS.config.update({region: 'us-east-1'});

/ Create S3 service object
ar s3 = new AWS.S3();
```

Click “Run” to run index.js in NodeJS.

The Linux terminal will show the app running again

You will see your new message added to the end of the list



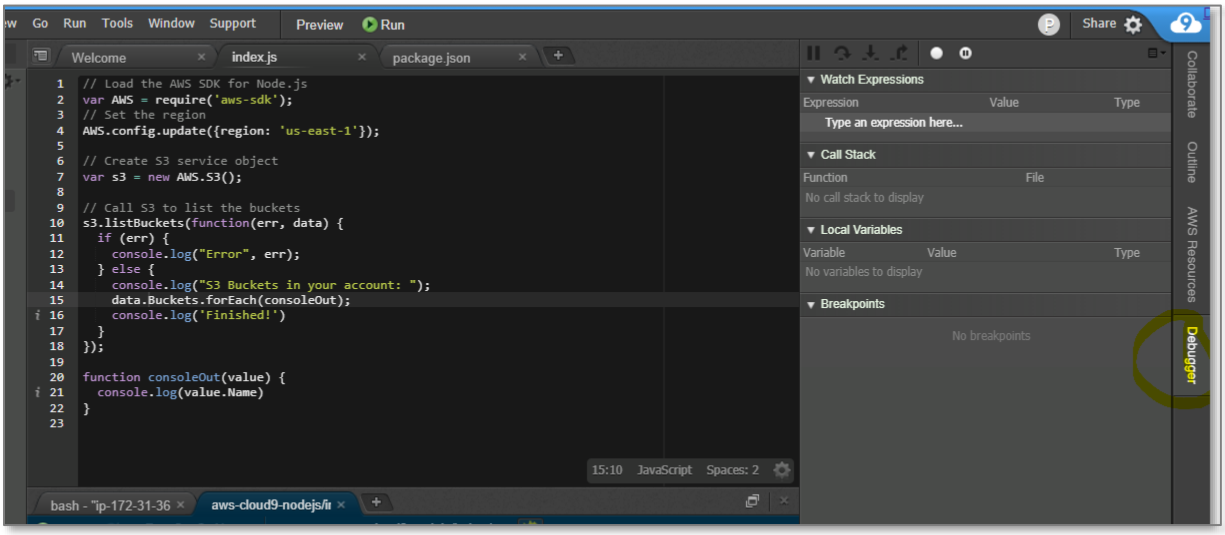
Command: aws-cloud9-nodejs/index.js

```
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-ap-southeast-2-361919435810
elasticbeanstalk-us-east-1-361919435810
thedevkid.com
Finished!
Waiting for the debugger to disconnect...
```

Using the Cloud9 Debugger

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

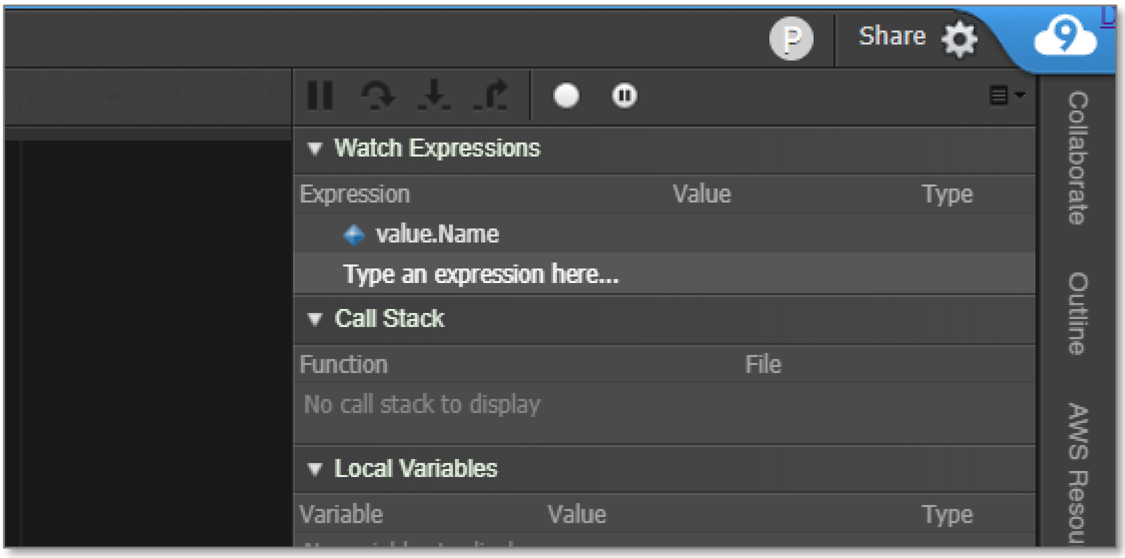
Open the debugger tab on the right hand side



```
// Load the AWS SDK for Node.js
var AWS = require('aws-sdk');
// Set the region
AWS.config.update({region: 'us-east-1'});

// Create S3 service object
var s3 = new AWS.S3();

// Call S3 to list the buckets
s3.listBuckets(function(err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("S3 Buckets in your account: ");
    data.Buckets.forEach(consoleOut);
    console.log('Finished!')
  }
});

function consoleOut(value) {
  console.log(value.Name)
}
```

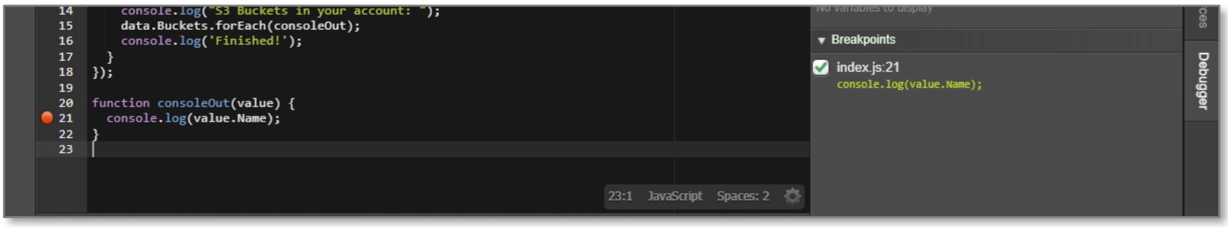
Double click on the *Watch Expressions* input field

Enter *value.Name*



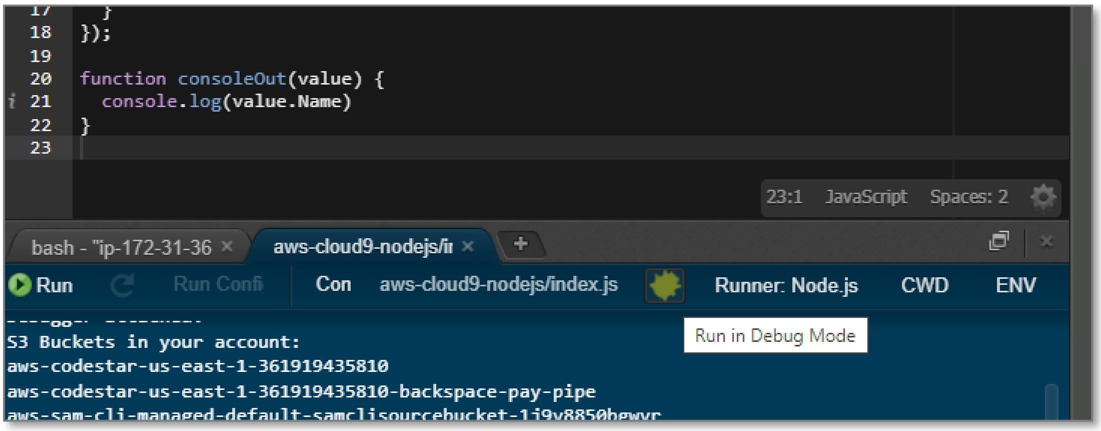
Click next to the line number for the *console.log(value.Name).* This will create a breakpoint and pause code execution every time a bucket name is output to the console.

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.



```
    console.log("S3 Buckets in your account: ");
    data.Buckets.forEach(consoleOut);
    console.log('Finished!');
  }
});

function consoleOut(value) {
  console.log(value.Name);
}

```

Make sure *Run in Debug Mode* is green



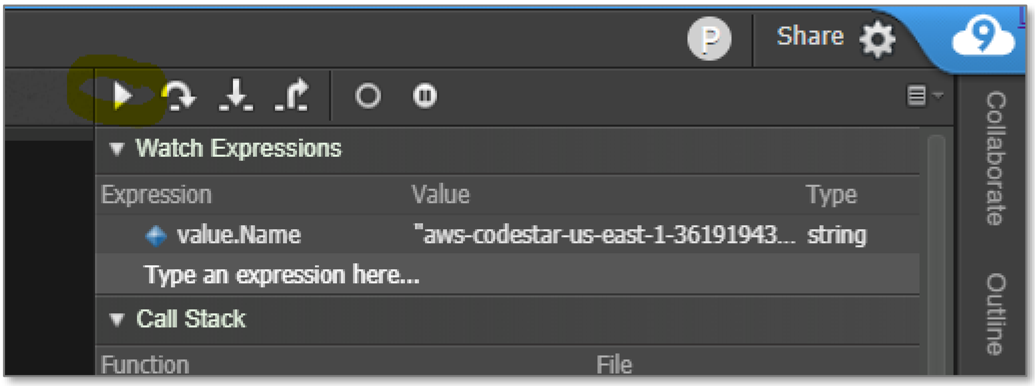
```
  }
});

function consoleOut(value) {
  console.log(value.Name)
}

```

```
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
```

Click *Run* to run index.js in NodeJS.

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

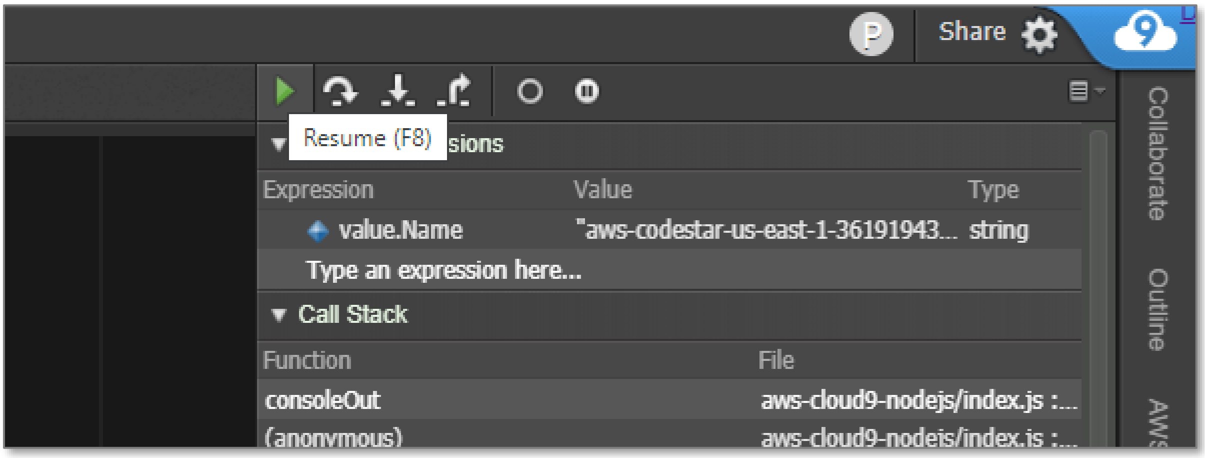
The code will run and pause at the breakpoint. The watch expression will have the value of the first basket in the list.



Expression	Value	Type
value.Name	"aws-codestar-us-east-1-36191943...	string
Type an expression here...		

Function	File

Click the resume button to keep running.



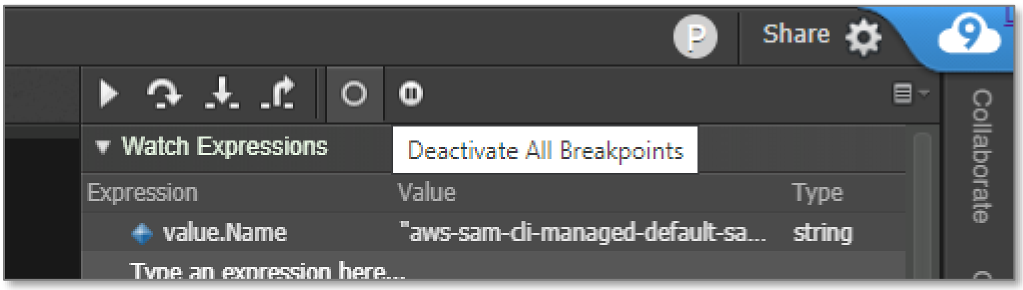
Expression	Value	Type
value.Name	"aws-codestar-us-east-1-36191943...	string
Type an expression here...		

Call Stack

Function	File
consoleOut	aws-cloud9-nodejs/index.js :...
(anonymous)	aws-cloud9-nodejs/index.js :...

The watch expression value will change.

Click *Deactivate All Breakpoints* to remove the breakpoint.



Watch Expressions

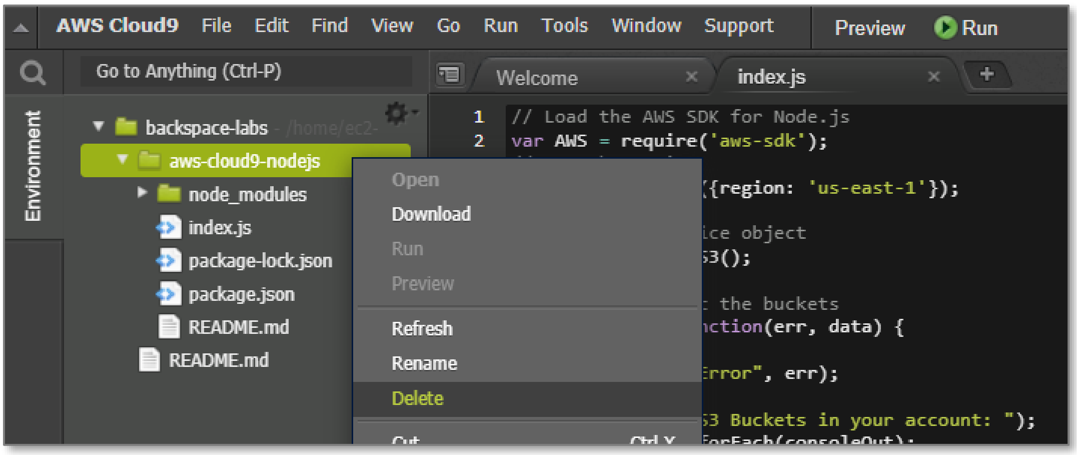
Expression	Value	Type
value.Name	"aws-sam-cli-managed-default-sa...	string
Type an expression here...		

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 breakpoints, watch expressions and the cloned repository.



```
// Load the AWS SDK for Node.js
var AWS = require('aws-sdk');
{region: 'us-east-1'});
ice object
3();
 the buckets
ction(err, data) {
rror", err);
3 Buckets in your account: ");
orEach(consoleOut);
```