

Brokers | Amazon MQ X S3 Management Cons X IAM > Roles X AWS Cloud9 X .NET Core sample for X Instances | EC2 Manager X IAM > Roles X + Incognito

docs.aws.amazon.com/cloud9/latest/user-guide/sample-dotnetcore.html

Gmail YouTube Maps Noticias Traducir News Translate Elastic Container Se... LattePanda 3 Delta... Turkey launches TO... ESP32 SIM800L GS... Basic-ESP32-Tutori... GPS Car Tracker Usi...

aws Buscar en esta guía Contacte con nosotros Español Volver a la consola Comentarios Preferencias

AWS Documentation AWS Cloud9 User Guide

AWS CDK sample LAMP sample WordPress sample Java sample C++ sample Python tutorial **.NET Core sample** Node.js sample PHP sample Ruby Go sample TypeScript sample Docker sample Advanced topics Security Troubleshooting Supported browsers Limits Document history

.NET Core sample for AWS Cloud9

PDF RSS

This sample enables you to run some .NET Core code in an AWS Cloud9 development environment.

Creating this sample might result in charges to your AWS account. These include possible charges for services such as Amazon EC2 and Amazon S3. For more information, see [Amazon EC2 Pricing](#) and [Amazon S3 Pricing](#).

Topics

- Prerequisites
- Step 1: Install required tools
- Step 2 (Optional): Install the .NET CLI extension for Lambda functions
- Step 3: Create a .NET Core console application project
- Step 4: Add code
- Step 5: Build and run the code
- Step 6: Create and set up a .NET Core console application project that uses the AWS SDK for .NET
- Step 7: Add AWS SDK code
- Step 8: Build and run the AWS SDK code
- Step 9: Clean up

En esta página

Prerequisites
Step 1: Install required tools
Step 2 (Optional): Install the .NET CLI extension for Lambda functions
Step 3: Create a .NET Core console application project
Step 4: Add code
Step 5: Build and run the code
Step 6: Create and set up a .NET Core console application project that uses the AWS SDK for .NET
Step 7: Add AWS SDK code
Step 8: Build and run the AWS SDK code
Step 9: Clean up

Type here to search Luxoft ENG US 10:48 AM 6/23/2023 72°F

<https://docs.aws.amazon.com/cloud9/latest/user-guide/sample-dotnetcore.html>

S Brokers | Amazon MQ X S3 Management Cons X IAM > Roles AWS Cloud9 .NET Core sample for Instances | EC2 Manat IAM > Roles Incognito

Gmail YouTube Maps Noticias Traducir News Translate Elastic Container Se... LattePanda 3 Delta... Turkey launches TO... ESP32 SIM800L GS... Basic-ESP32-Tutori... GPS Car Tracker Usi...

aws Services Search [Alt+S] N. Virginia cloudUserLuis @ 5501-4694-3653

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 pre-configured 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.

How it works

Create an AWS Cloud9 development environment on a new Amazon EC2 instance or connect it to your own Linux server through SSH. Once you've created an AWS Cloud9 environment, you will have immediate access to a rich code editor, integrated debugger and built-in terminal with pre-configured AWS CLI – all within your browser.

Using the AWS Cloud9 dashboard, you can create and switch between many different AWS Cloud9 environments, each one containing the customised tools, runtimes and files

New AWS Cloud9 environment

Create environment

Getting started

Before you start (2-min read)

Create an environment (2-min read)

Working with environments (15-min read)

Working with the IDE (10-min read)

Working with AWS Lambda (5-min read)

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Type here to search Luxoft ENG 10:47 AM 72°F 5/22/2023

S Brokers | Amazon X S3 Management X IAM > Roles X AWS Cloud9 X .NET Core sample X Instances | EC2 M X IAM > Roles X amazon mq forge X + Incognito

Gmail YouTube Maps Noticias Traducir News Translate Elastic Container Se... LattePanda 3 Delta... Turkey launches TO... ESP32 SIM800L GS... Basic-ESP32-Tutori... GPS Car Tracker Usi...

aws Services Search [Alt+S] N. Virginia cloudUserLuis @ 5501-4694-3653

AWS Cloud9 > Environments > Create environment

Create environment Info

Details

Name Limit of 60 characters, alphanumeric and unique per user.

Description – *optional* Limit 200 characters.

Environment type Info Determines what the Cloud9 IDE will run on.

New EC2 instance Cloud9 creates an EC2 instance in your account. The configuration of your EC2 instance cannot be changed by Cloud9 after creation.

Existing compute You have an existing instance or server that you'd like to use.

New EC2 instance

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Type here to search Luxoft ENG 10:55 AM US 6/23/2023

S Brokers | Amazon X S3 Management X IAM > Roles X AWS Cloud9 X .NET Core sample X Instances | EC2 M X IAM > Roles X amazon mq forge X + Incognito

Gmail YouTube Maps Noticias Traducir News Translate Elastic Container Se... LattePanda 3 Delta... Turkey launches TO... ESP32 SIM800L GS... Basic-ESP32-Tutori... GPS Car Tracker Usi...

aws Services Search [Alt+S] N. Virginia cloudUserLuis @ 5501-4694-3653

New EC2 instance

Instance type [Info](#)

The memory and CPU of the EC2 instance that will be created for Cloud9 to run on.

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 web projects.

m5.large (8 GiB RAM + 2 vCPU)
Recommended for production and most general-purpose development.

Additional instance types
Explore additional instances to fit your needs.

Platform [Info](#)

This will be installed on your EC2 instance. We recommend Amazon Linux 2.

Amazon Linux 2

Timeout

How long Cloud9 can be inactive (no user input) before auto-hibernating. This helps prevent unnecessary charges.

30 minutes

Network settings

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Type here to search Luxoft ENG 10:55 AM US 6/23/2023

Brokers | Amazon X | S3 Management X | IAM > Roles X | AWS Cloud9 X | .NET Core sample X | Instances | EC2 M X | IAM > Roles X | amazon mq forge X | + Incognito

Gmail YouTube Maps Noticias Traducir News Translate Elastic Container Se... LattePanda 3 Delta... Turkey launches TO... ESP32 SIM800L GS... Basic-ESP32-Tutori... GPS Car Tracker Usi...

aws Services Search [Alt+S] N. Virginia cloudUserLuis @ 5501-4694-3653

Network settings Info

Connection
How your environment is accessed.

AWS Systems Manager (SSM)
Accesses environment via SSM without opening inbound ports (no ingress).

Secure Shell (SSH)
Accesses environment directly via SSH, opens inbound ports.

▶ VPC settings Info

▶ Tags – optional Info
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

The following IAM resources will be created in your account

- AWSServiceRoleForAWSCloud9 – 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. [Find out more](#)
- AWSCloud9SSMAccessRole and AWSCloud9SSMInstanceProfile – A service role and an instance profile are automatically created if Cloud9 accesses its EC2 instance through AWS Systems Manager. If your environments no longer require EC2 instances that block incoming traffic, you can delete these roles using the AWS IAM console. [Find out more](#)

Cancel **Create**

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Type here to search Luxoft ENG 10:56 AM US 6/23/2023

S Create RabbitMQ X S3 Management X IAM > Roles X AWS Cloud9 .NET Core sample X Instances | EC2 M X IAM > Roles X amazon mq forge X + Incognito

Gmail YouTube Maps Notícias Traducir News Translate Elastic Container Se... LattePanda 3 Delta... Turkey launches TO... ESP32 SIM800L GS... Basic-ESP32-Tutori... GPS Car Tracker Usi...

aws Services Search [Alt+S] N. Virginia cloudUserLuis @ 5501-4694-3653

AWS Cloud9 Environments Documentation

Successfully created LuxoftTestEC2LinuxIDE. To get the most out of your environment, see Best practices for using AWS Cloud9 [?]

AWS Cloud9 > Environments

Environments (1)

Delete View details Open in Cloud9 Create environment

My environments

Name	Cloud9 IDE	Environment type	Connection	Permission	Owner ARN
LuxoftTestEC2LinuxIDE	Open	EC2 instance	AWS Systems Manager (SSM)	Owner	arn:aws:iam::550146943653:user/cloudUserLuis

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Type here to search luxsoft Type here to search ENG 10:58 AM US 6/23/2023

S Brokers | Amazon X S3 Management X IAM > Roles X AWS Cloud9 X .NET Core sample X Instances | EC2 M X IAM > Roles X amazon mq forge X + Incognito

Gmail YouTube Maps Noticias Traducir News Translate Elastic Container Se... LattePanda 3 Delta... Turkey launches TO... ESP32 SIM800L GS... Basic-ESP32-Tutori... GPS Car Tracker Usi...

aws Services Search [Alt+S] N. Virginia cloudUserLuis @ 5501-4694-3653 X

AWS Cloud9 Environments Documentation

Successfully created LuxoftTestEC2LinuxIDE. To get the most out of your environment, see Best practices for using AWS Cloud9 [?]

AWS Cloud9 > Environments > LuxoftTestEC2LinuxIDE

LuxoftTestEC2LinuxIDE

Delete Open in Cloud9

Details Edit

Name	Owner ARN	Status
LuxoftTestEC2LinuxIDE	arn:aws:iam::550146943653:user/cloudUserLuis	Ready
Description	Number of members	Lifecycle status
LuxoftTestEC2LinuxIDE	1	Created
Environment type		
EC2 instance		

EC2 instance Network settings Tags

EC2 instance Manage EC2 instance

ARN arn:aws:cloud9:us-east-1:550146943653:environment/LuxoftTestEC2LinuxIDE

Instance type t2.micro (1 GiB RAM, 1 vCPU)

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Type here to search luxsoft ENG 10:58 AM US 6/23/2023

S Brokers | Amazon X S3 Management X IAM > Roles X AWS Cloud9 X .NET Core sample X Instances | EC2 M X IAM > Roles X amazon mq forge X + Incognito

Gmail YouTube Maps Notícias Traducir News Translate Elastic Container Se... LattePanda 3 Delta... Turkey launches TO... ESP32 SIM800L GS... Basic-ESP32-Tutori... GPS Car Tracker Usi...

aws Services Search [Alt+S] N. Virginia cloudUserLuis @ 5501-4694-3653

AWS Cloud9

Details Edit

Name	Owner ARN	Status
LuxoftTestEC2LinuxIDE	arn:aws:iam::550146943653:user/cloudUserLuis	Ready
Description	Number of members	Lifecycle status
LuxoftTestEC2LinuxIDE	1	Created
Environment type		
EC2 instance		

EC2 instance Network settings Tags

EC2 instance

Manage EC2 instance

ARN	Instance type
arn:aws:cloud9:us-east-1:550146943653:environment:996bed50ab1e46d1b4ba102266923976	t2.micro (1 GiB RAM + 1 vCPU)
Platform	Storage
Amazon Linux 2	EBS only

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Type here to search Luxoft ENG 10:59 AM US 6/23/2023

S Brokers | Amazon X S3 Management X IAM > Roles X AWS Cloud9 X .NET Core sample X Instances | EC2 M X IAM > Roles X amazon mq forge X + Incognito

Gmail YouTube Maps Noticias Traducir News Translate Elastic Container Se... LattePanda 3 Delta... Turkey launches TO... ESP32 SIM800L GS... Basic-ESP32-Tutori... GPS Car Tracker Usi...

aws Services Search [Alt+S] AWS Cloud9 Environments LuxoftTestEC2LinuxIDE Delete Open in Cloud9 Edit

AWS Cloud9 Environments Documentation

LuxoftTestEC2LinuxIDE

Details

Name	Owner ARN	Status
LuxoftTestEC2LinuxIDE	arn:aws:iam::550146943653:user/cloudUserLuis	Ready
Description	Number of members	Lifecycle status
LuxoftTestEC2LinuxIDE	1	Created
Environment type		
EC2 instance	EC2 instance	

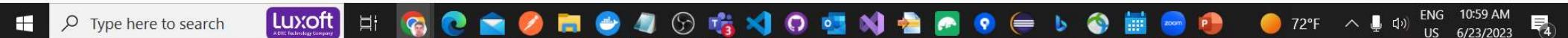
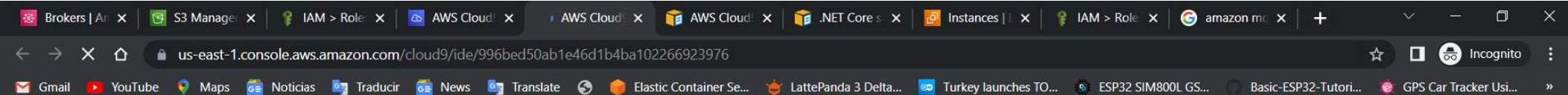
Network settings

Connection and VPC

Connection	Amazon Virtual Private Cloud (VPC)	Subnet
AWS Systems Manager (SSM)	vpc-0ca8fe43d428e0011	subnet-09d5051591a99e90b

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Type here to search luxsoft ENG 10:59 AM US 6/23/2023 4



The screenshot shows a browser window with the URL <https://us-east-1.console.aws.amazon.com/cloud9/ide/996bed50ab1e46d1b4ba102266923976>. The page displays the AWS Cloud9 interface, which includes a sidebar with developer tools, a central area with the title "AWS Cloud9" and subtitle "Welcome to your development environment", and a toolkit section. The browser's address bar and various tabs are visible at the top, and the Windows taskbar is at the bottom.

S3 Manager | IAM > Role | AWS Cloud9 | LuxoftTestEl | AWS Cloud9 | .NET Core S... | Instances | IAM > Role | amazon m... | + | Incognito

Gmail YouTube Maps Noticias Traducir News Translate Elastic Container Se... LattePanda 3 Delta... Turkey launches TO... ESP32 SIM800L GS... Basic-ESP32-Tutori... GPS Car Tracker Usi...

File Edit Find View Go Run Tools Window Support Preview Run

Go Anything (Ctrl-P)

Welcome

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.

Toolkit for AWS Cloud9

The AWS Toolkit for Cloud9 is an IDE extension that simplifies accessing and interacting with resources from services such as AWS Lambda, AWS CloudFormation, and AWS API Gateway. With the toolkit, developers can also develop, debug, and deploy applications

Getting started

Create File

Upload Files...

Clone from GitHub

Terminal

Immediate

S Brokers | A S3 Manager X IAM > Role X AWS CloudWatch Metrics X LuxoftTestEc2Instances X AWS CloudWatch Metrics X .NET Core Services X Instances X IAM > Role X amazon m... X + Incognito

Gmail YouTube Maps Noticias Traducir News Translate Elastic Container Se... LattePanda 3 Delta... Turkey launches TO... ESP32 SIM800L GS... Basic-ESP32-Tutori... GPS Car Tracker Usi...

File Edit Find View Go Run Tools Window Support Preview Run

Go to Anything (Ctrl-P)

LuxoftTestEC2Instances .c9 README.md

Welcome Close Pane Ctrl-W Close All Tabs In All Panes Alt-Shift-W Close All But Current Tab Ctrl-Alt-W

Welcome Ctrl-1 bash - "ip-172-31-83-167.ec2.internal" Ctrl-2 Immediate Ctrl-3

Split Pane in Two Rows Split Pane in Two Columns

Cloud9 to your development environment

To write, run, and debug your code with just a browser. You can code for AWS Lambda and Amazon API Gateway, share your IDE with others in real time, and much more.

Getting started

Create File

Upload Files...

Clone from GitHub

Toolkit for AWS Cloud9

The AWS Toolkit for Cloud9 is an IDE extension that simplifies accessing and interacting with resources from services such as AWS Lambda, AWS CloudFormation, and AWS API Gateway. With the toolkit, developers can also develop, debug, and deploy applications

bash - "ip-172-31-83-167.ec2.internal" x Immediate x +

cloudUserLuis:~/environment \$

CodeWhisperer AWS: profile:default

Luxoft Type here to search

72°F ENG US 11:01 AM 6/23/2023

A screenshot of a web browser displaying a terminal session. The terminal window title is "bash - ip-172-31-83-167.e.x". The command being run is:

```
git clone cloudUserLuis:~/environment $ git clone https://github.com/luiscoco/JavaScript_date
```

The browser's address bar shows the URL: us-east-1.console.aws.amazon.com/cloud9/ide/996bed50ab1e46d1b4ba102266923976

The browser interface includes a tab bar with various AWS services like S3 Manager, IAM, AWS CloudWatch, and .NET Core. Below the tabs is a toolbar with links to Gmail, YouTube, Maps, and other resources.

The terminal window has a sidebar showing a file structure under "LuxoftTestEC2Lin":

- ..
- ↳ .c9
- ↳ README.md

At the bottom of the terminal window, there is another tab labeled "Immediate" with the prompt:

```
cloudUserLuis:~/environment $
```

The overall interface is designed for developers to work directly from their web browser using AWS Cloud9 services.

The screenshot shows a browser window with multiple tabs open, including various AWS services like S3 Manager, IAM, and AWS CloudWatch. The main content area is a terminal window titled "bash - ip-172-31-83-167.e.x". Inside the terminal, the user has run the command:

```
git clone cloudUserLuis:~/environment $ git clone https://github.com/luiscoco/JavaScript_date
Cloning into 'JavaScript_date'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 6 (delta 0), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.
cloudUserLuis:~/environment $
```

Below this terminal window is another one titled "Immediate", which shows the prompt:

```
cloudUserLuis:~/environment $
```

At the bottom of the screen is a taskbar with icons for various applications, including Luxoft, Microsoft Edge, and File Explorer. The system tray shows the date and time as 11:02 AM on 6/23/2023, along with other standard icons.

S Brokers | A S3 Manager X IAM > Role X AWS CloudWatch Metrics X LuxoftTestEc2L AWS CloudWatch Metrics .NET Core Services X Instances X IAM > Role X amazon m... X + Incognito

Gmail YouTube Maps Noticias Traducir News Translate Elastic Container Se... LattePanda 3 Delta... Turkey launches TO... ESP32 SIM800L GS... Basic-ESP32-Tutori... GPS Car Tracker Usi...

File Edit Find View Go Run Tools Window Support Preview Run

Go to Anything (Ctrl-P)

Welcome bash - ip-172-31-83-167.e x

```
git clone cloudUserLuis:~/environment $ git clone https://github.com/luiscoco/JavaScript_date
Cloning into 'JavaScript_date'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 6 (delta 0), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.
cloudUserLuis:~/environment $ ls
JavaScript_date README.md
cloudUserLuis:~/environment $
```

aws

LuxoftTestEC2Lin .c9 JavaScript_date .git JS date.js README.md README.md

bash - ip-172-31-83-167.e x Immediate

```
cloudUserLuis:~/environment $
```

main CodeWhisperer AWS:profile.default

Luxoft Type here to search

72°F ENG US 11:02 AM 6/23/2023

Brokers | Amazon S3 Manager | IAM > Roles | AWS CloudWatch Metrics | LuxoftTestEC2Linux | AWS CloudWatch Metrics | .NET Core Services | Instances | IAM > Roles | amazon music | + | us-east-1.console.aws.amazon.com/cloud9/ide/996bed50ab1e46d1b4ba102266923976 | Incognito

Gmail YouTube Maps Noticias Traducir News Translate Elastic Container Service LattePanda 3 Delta... Turkey launches TO... ESP32 SIM800L GS... Basic-ESP32-Tutorial... GPS Car Tracker Us...

File Edit Find View Go Run Tools Window Support Preview Run

Go to Anything (Ctrl+P)

Welcome bash - ip-172-31-83-167.e x

```
git clone cloudUserLuis:~/environment $ git clone https://github.com/luiscoco/JavaScript_date
Cloning into 'JavaScript_date'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 6 (delta 0), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.
cloudUserLuis:~/environment $ ls
JavaScript_date README.md
cloudUserLuis:~/environment $ cd JavaScript_date
```

aws

LuxoftTestEC2Linux (.c9) JavaScript_date (.git) JS date.js README.md README.md

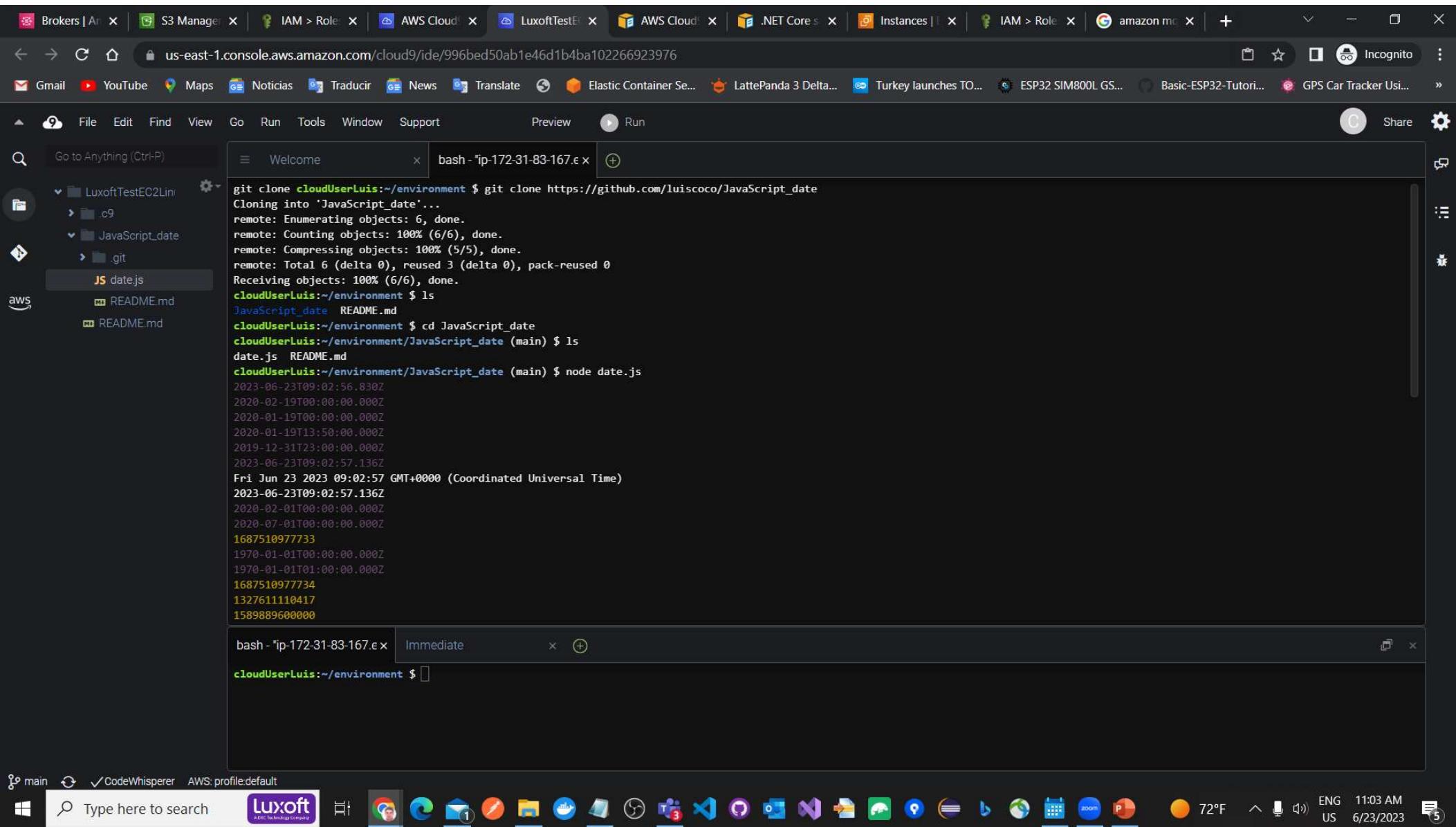
bash - ip-172-31-83-167.e x Immediate

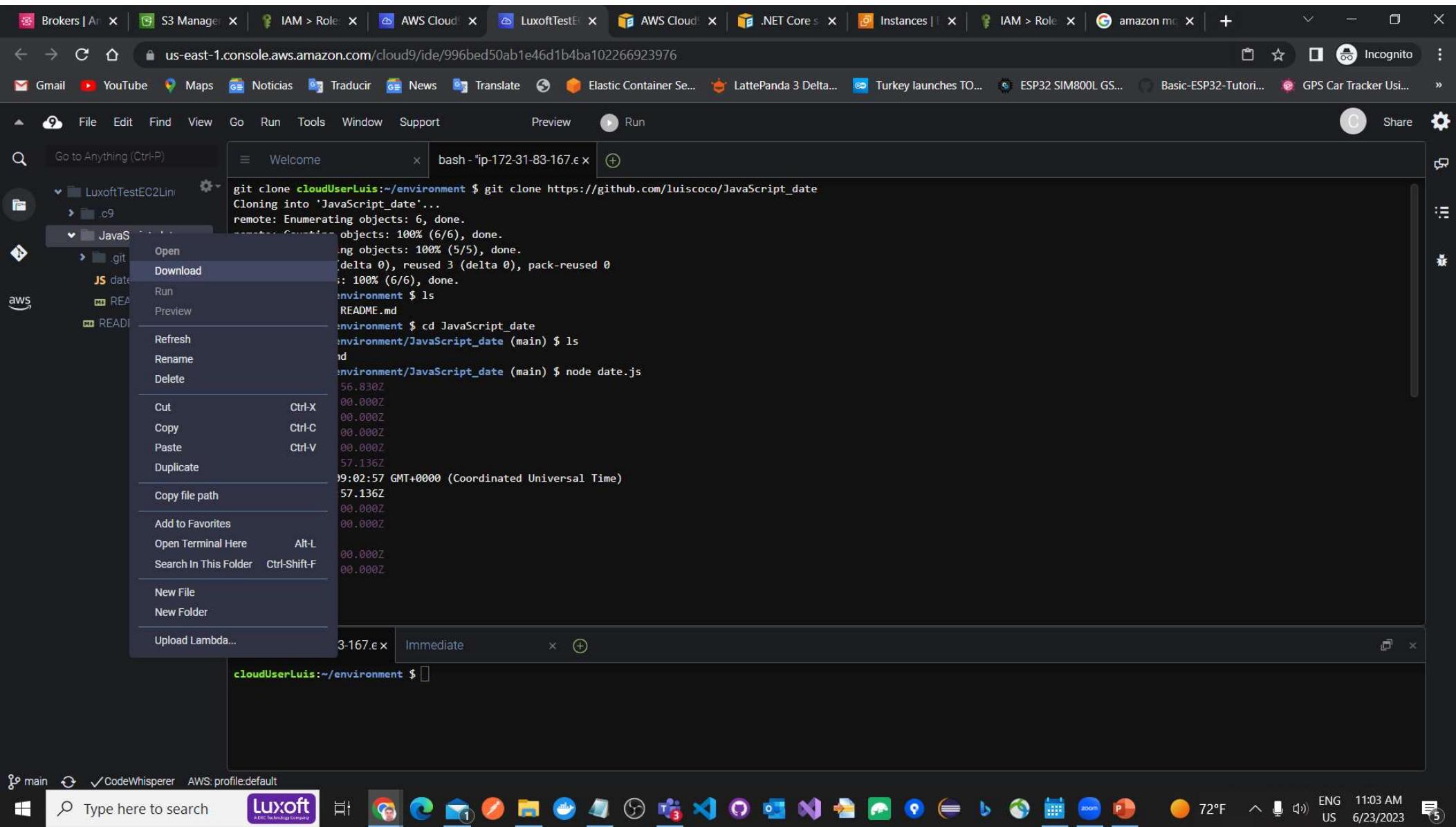
```
cloudUserLuis:~/environment $
```

main CodeWhisperer AWS:profile.default

Luxoft Type here to search

72°F ENG US 11:02 AM 6/23/2023





S3 Manager | IAM > Role | AWS CloudWatch Metrics | LuxoftTestEnv | AWS CloudWatch Metrics | .NET Core Samples | Instances | IAM > Role | amazon m... | + | Incognito

Gmail YouTube Maps Noticias Traducir News Translate Elastic Container Se... LattePanda 3 Delta... Turkey launches TO... ESP32 SIM800L GS... Basic-ESP32-Tutori... GPS Car Tracker Usi...

AWS Documentation AWS Cloud9 User Guide

Step 1: Install required tools

In this step, you install the .NET Core SDK into your environment, which is required to run this sample.

1. Confirm whether the latest version of the .NET Core SDK is already installed in your environment. To do this, in a terminal session in the AWS Cloud9 IDE, run the .NET Core command line interface (CLI) with the `--version` option.

```
dotnet --version
```

If the .NET Command Line Tools version is displayed, and the version is 2.0 or greater, skip ahead to [Step 3: Create a .NET Core console application project](#). If the version is less than 2.0, or if an error such as `bash: dotnet: command not found` is displayed, continue on to install the .NET Core SDK.
2. For Amazon Linux, in a terminal session in the AWS Cloud9 IDE, run the following commands to help ensure the latest security updates and bug fixes are installed, and to install a `libunwind` package that the .NET Core SDK needs. (To start a new terminal session, on the menu bar, choose **Window**, **New Terminal**).

```
sudo yum -y update
sudo yum -y install libunwind
```

For Ubuntu Server, in a terminal session in the AWS Cloud9 IDE, run the following command to help ensure the latest security updates and bug fixes are installed. (To start a new terminal session, on the menu bar, choose **Window**, **New Terminal**.)

En esta página

- Install the .NET CLI extension for Lambda functions
- Step 3: Create a .NET Core console application project
- Step 4: Add code
- Step 5: Build and run the code
- Step 6: Create and set up a .NET Core console application project that uses the AWS SDK for .NET
- Step 7: Add AWS SDK code
- Step 8: Build and run the AWS SDK code
- Step 9: Clean up

Type here to search Luxoft ENG US 11:18 AM 6/23/2023 76°F

Download the .NET Core SDK installer script into your environment by running the following command.

```
 wget https://dot.net/v1/dotnet-install.sh
```

The screenshot shows a terminal window titled "bash - *ip-172-31-83-167.ex" with the command "wget https://dot.net/v1/dotnet-install.sh" being run. The output of the command is displayed, showing the progress of the download from Microsoft's servers. The terminal is part of a larger AWS Cloud9 IDE interface, which includes a sidebar with project files like ".NET5_StudentAdminPortal.API" and a bottom navigation bar with various application icons.

```
cloudUserLuis:~/environment $ wget https://dot.net/v1/dotnet-install.sh
--2023-06-23 09:18:49-- https://dot.net/v1/dotnet-install.sh
Resolving dot.net (dot.net)... 20.112.52.29, 20.53.203.50, 20.81.111.85, ...
Connecting to dot.net (dot.net)|20.112.52.29|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://dotnet.microsoft.com/download/dotnet/scripts/v1/dotnet-install.sh
--2023-06-23 09:18:49-- https://dotnet.microsoft.com/download/dotnet/scripts/v1/dotnet-install.sh
Resolving dotnet.microsoft.com (dotnet.microsoft.com)... 13.107.253.40, 13.107.226.40, 2620:1ec:46::40, ...
Connecting to dotnet.microsoft.com (dotnet.microsoft.com)|13.107.253.40|:443... connected.
HTTP request sent, awaiting response... 200 OK
Cookie coming from dotnet.microsoft.com attempted to set domain to dotnet.microsoft.com
Cookie coming from dotnet.microsoft.com attempted to set domain to dotnet.microsoft.com
Length: 58994 (58K) [application/x-sh]
Saving to: 'dotnet-install.sh'

100%[=====] 58,994 --.-K/s in 0.001s

2023-06-23 09:18:49 (82.0 MB/s) - 'dotnet-install.sh' saved [58994/58994]

cloudUserLuis:~/environment $
```

Make the installer script executable by the current user by running the following command.

```
sudo chmod u=rwx dotnet-install.sh
```

The screenshot shows a terminal window titled "bash - *ip-172-31-83-167.ex" with the following session history:

```
cloudUserLuis:~/environment $ wget https://dot.net/v1/dotnet-install.sh
--2023-06-23 09:18:49-- https://dot.net/v1/dotnet-install.sh
Resolving dot.net (dot.net)... 20.112.52.29, 20.53.203.50, 20.81.111.85, ...
Connecting to dot.net (dot.net)|20.112.52.29|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://dotnet.microsoft.com/download/dotnet/scripts/v1/dotnet-install.sh [following]
--2023-06-23 09:18:49-- https://dotnet.microsoft.com/download/dotnet/scripts/v1/dotnet-install.sh
Resolving dotnet.microsoft.com (dotnet.microsoft.com)... 13.107.253.40, 13.107.226.40, 2620:1ec:46::40, ...
Connecting to dotnet.microsoft.com (dotnet.microsoft.com)|13.107.253.40|:443... connected.
HTTP request sent, awaiting response... 200 OK
Cookie coming from dotnet.microsoft.com attempted to set domain to dotnet.microsoft.com
Cookie coming from dotnet.microsoft.com attempted to set domain to dotnet.microsoft.com
Length: 58994 (58K) [application/x-sh]
Saving to: 'dotnet-install.sh'

100%[=====] 58,994      --.-K/s   in 0.001s

2023-06-23 09:18:49 (82.0 MB/s) - 'dotnet-install.sh' saved [58994/58994]

cloudUserLuis:~/environment $ sudo chmod u=rwx dotnet-install.sh
cloudUserLuis:~/environment $
```

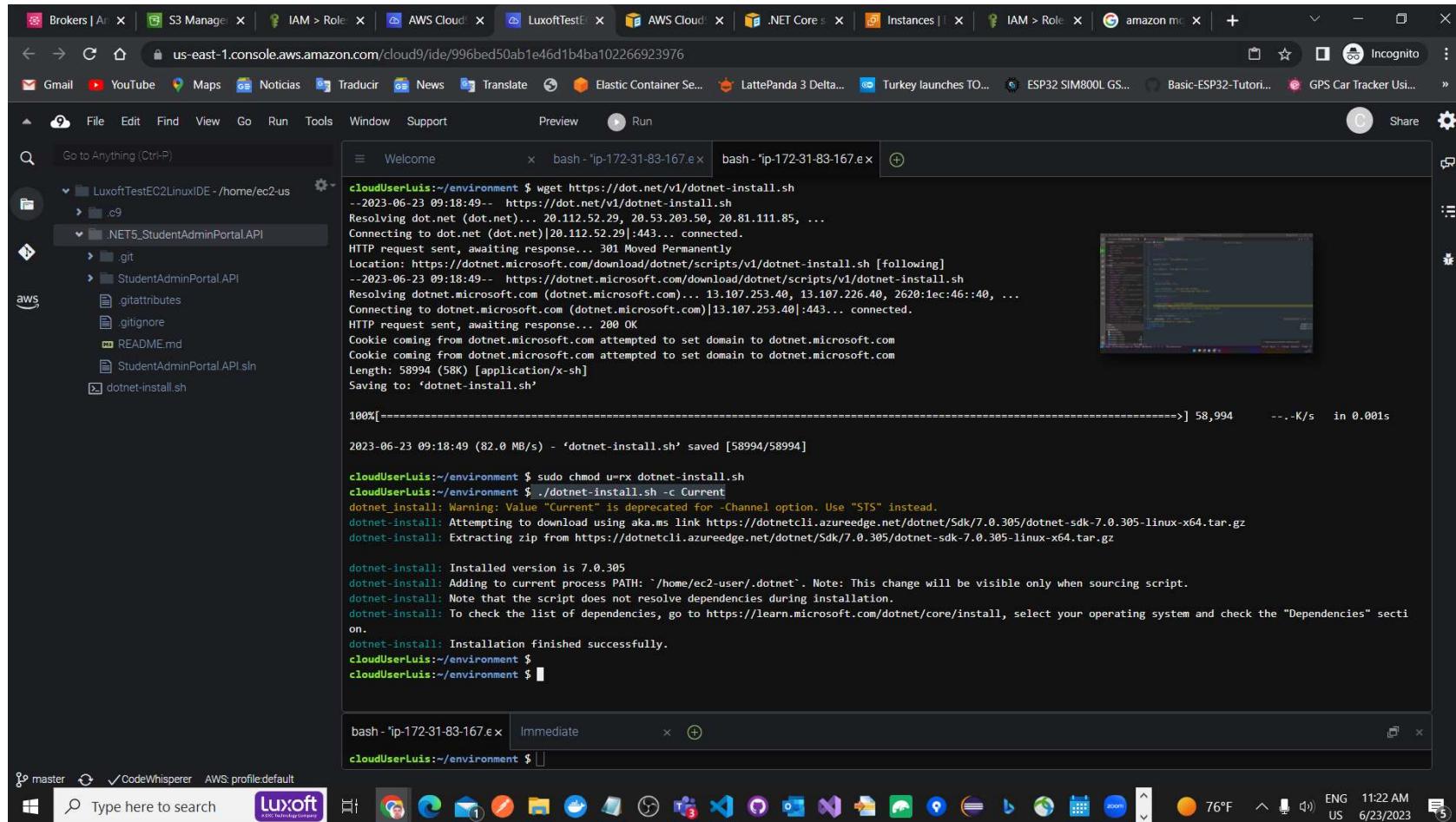
Below the main terminal window, there is a smaller "Immediate" tab window showing the prompt:

```
cloudUserLuis:~/environment $
```

The desktop environment includes a sidebar with file explorer, search, and other AWS services like S3 Manager, IAM, and AWS CloudWatch. The taskbar at the bottom shows various application icons and system status.

Run the installer script, which downloads and installs the .NET Core SDK, by running the following command.

```
./dotnet-install.sh -c Current
```



```
cloudUserLuis:~/environment $ wget https://dot.net/v1/dotnet-install.sh
--2023-06-23 09:18:49- https://dot.net/v1/dotnet-install.sh
Resolving dot.net (dot.net)... 20.112.52.29, 20.53.203.50, 20.81.111.85, ...
Connecting to dot.net (dot.net)|20.112.52.29|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://dotnet.microsoft.com/download/dotnet/scripts/v1/dotnet-install.sh
--2023-06-23 09:18:49- https://dotnet.microsoft.com/download/dotnet/scripts/v1/dotnet-install.sh
Resolving dotnet.microsoft.com (dotnet.microsoft.com)|13.107.253.40, 13.107.226.40, 2620:1ec:46::40, ...
Connecting to dotnet.microsoft.com (dotnet.microsoft.com)|13.107.253.40|:443... connected.
HTTP request sent, awaiting response... 200 OK
Cookie coming from dotnet.microsoft.com attempted to set domain to dotnet.microsoft.com
Cookie coming from dotnet.microsoft.com attempted to set domain to dotnet.microsoft.com
Length: 58994 (58K) [application/x-sh]
Saving to: 'dotnet-install.sh'

100%[=====] 58,994     --.K/s   in 0.001s

2023-06-23 09:18:49 (82.0 MB/s) - 'dotnet-install.sh' saved [58994/58994]

cloudUserLuis:~/environment $ sudo chmod u=rwx dotnet-install.sh
cloudUserLuis:~/environment $ ./dotnet-install.sh -c Current
dotnet_install: Warning: Value "Current" is deprecated for -Channel option. Use "STS" instead.
dotnet_install: Attempting to download using aka.ms link https://dotnetcli.azureedge.net/dotnet/Sdk/7.0.305/dotnet-sdk-7.0.305-linux-x64.tar.gz
dotnet-install: Extracting zip from https://dotnetcli.azureedge.net/dotnet/Sdk/7.0.305/dotnet-sdk-7.0.305-linux-x64.tar.gz

dotnet-install: Installed version is 7.0.305
dotnet-install: Adding to current process PATH: `/home/ec2-user/.dotnet`. Note: This change will be visible only when sourcing script.
dotnet-install: Note that the script does not resolve dependencies during installation.
dotnet-install: To check the list of dependencies, go to https://learn.microsoft.com/dotnet/core/install, select your operating system and check the "Dependencies" section.
dotnet-install: Installation finished successfully.
cloudUserLuis:~/environment $
cloudUserLuis:~/environment $
```

Add the .NET Core SDK to your `PATH`. To do this, in the shell profile for the environment (for example, the `.bashrc` file), add the `$HOME/.dotnet` subdirectory to the `PATH` variable for the environment, as follows.

- a. Open the `.bashrc` file for editing by using the `vi` command.

```
vi ~/.bashrc
```



- b. For Amazon Linux, using the down arrow or `j` key, move to the line that starts with `export PATH`.

For Ubuntu Server, move to the last line of the file by typing `G`.

- c. Using the right arrow or `$` key, move to the end of that line.
- d. Switch to insert mode by pressing the `i` key. (-- INSERT --- will appear at the end of the display.)
- e. For Amazon Linux, add the `$HOME/.dotnet` subdirectory to the `PATH` variable by typing `:$HOME/.dotnet`. Be sure to include the colon character (:). The line should now look similar to the following.

```
export PATH=$PATH:$HOME/.local/bin:$HOME/bin:$HOME/.dotnet
```



For Ubuntu Server, press the right arrow key and then press `Enter` twice, followed by typing the following line by itself at the end of the file.

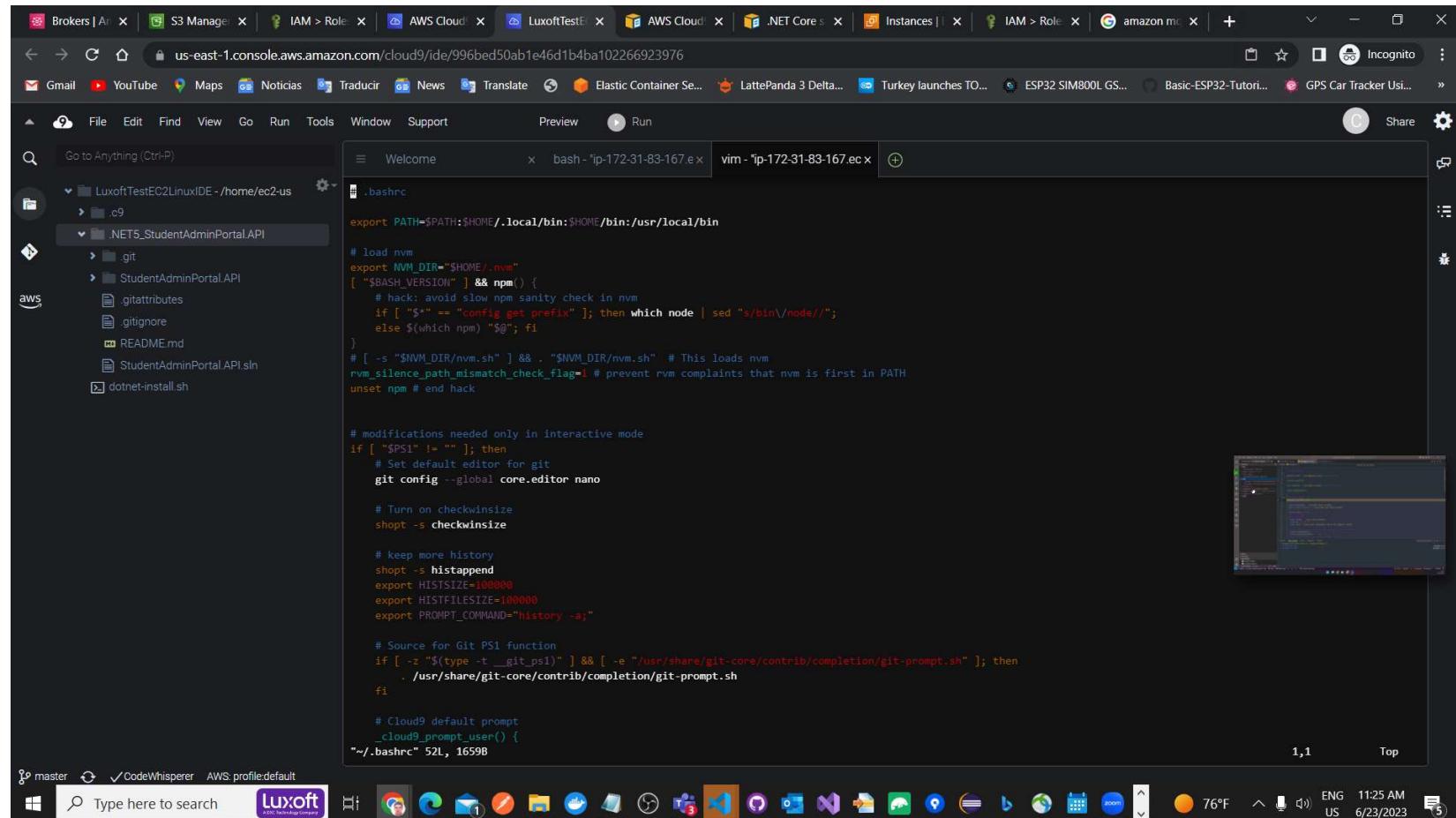
```
export PATH=$HOME/.dotnet:$PATH
```



- f. Save the file. To do this, press the `Esc` key (-- INSERT --- will disappear from the end of the display), type `:wq` (to write to and then quit the file), and then press `Enter`.

Open the .bashrc file for editing by using the vi command:

vi ~/.bashrc



The screenshot shows a Windows desktop environment. At the top, there is a taskbar with various pinned icons. Below the taskbar, a browser window is open to the URL <https://us-east-1.console.aws.amazon.com/cloud9/ide/996bed50ab1e46d1b4ba102266923976>. The browser title bar says "Welcome - bash - ip-172-31-83-167.ecx". The main content area of the browser shows a terminal session in vim editor mode, displaying the contents of the .bashrc file. The terminal window has tabs labeled ".bashrc" and "vim - ip-172-31-83-167.ecx". The vim status bar at the bottom right indicates "1,1 Top". The terminal code is as follows:

```
export PATH=$PATH:$HOME/.local/bin:$HOME/bin:/usr/local/bin

# load nvm
export NVM_DIR="$HOME/.nvm"
[ "$BASH_VERSION" ] && npm() {
    # hack: avoid slow npm sanity check in nvm
    if [ "$*" == "config get prefix" ]; then which node | sed "s/bin\/node//";
    else $(which npm) "$@"; fi
}
# [ -s "$NVM_DIR/nvm.sh" ] && . "$NVM_DIR/nvm.sh" # This loads nvm
rvm_silence_path_mismatch_check_flag=1 # prevent rvm complaints that nvm is first in PATH
unset npm # end hack

# modifications needed only in interactive mode
if [ "$PS1" != "" ]; then
    # Set default editor for git
    git config --global core.editor nano

    # Turn on checkwinsize
    shopt -s checkwinsize

    # keep more history
    shopt -s histappend
    export HISTSIZE=100000
    export HISTFILESIZE=100000
    export PROMPT_COMMAND="history -a"

    # Source for Git PS1 function
    if [ -z "$(type -t _git_ps1)" ] && [ -e "/usr/share/git-core/contrib/completion/git-prompt.sh" ]; then
        . /usr/share/git-core/contrib/completion/git-prompt.sh
    fi

    # Cloud9 default prompt
    _cloud9_prompt_user() {
        .bashrc" 52L, 1659B
```

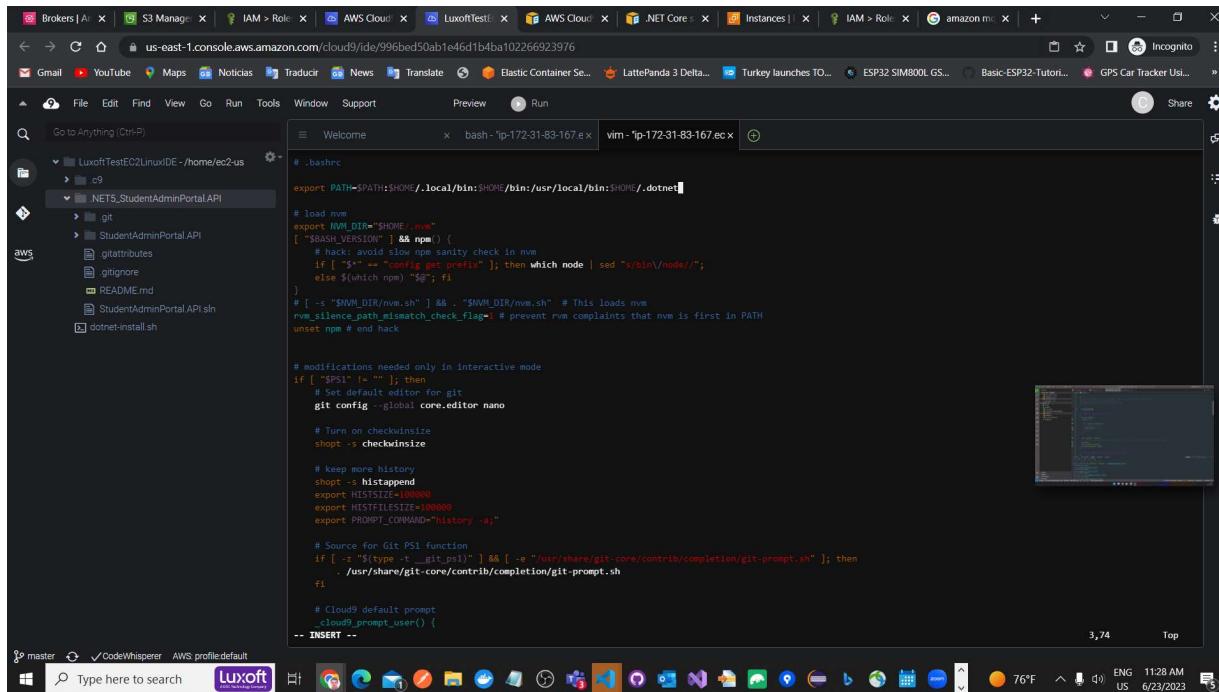
For Amazon Linux, using the **down arrow** or **j** key, move to the line that starts with `export PATH`.

Using the **right arrow** or **\$** key, move to the end of that line.

Switch to insert mode by pressing the **i** key. (**-- INSERT --** will appear at the end of the display.)

For Amazon Linux, add the `$HOME/.dotnet` subdirectory to the `PATH` variable by typing `:$HOME/.dotnet`. Be sure to include the colon character (:). The line should now look similar to the following.

```
export PATH=$PATH:$HOME/.local/bin:$HOME/bin:$HOME/.dotnet
```



```
# .bashrc

# Load nvm
export NVM_DIR="$HOME/.nvm"
[ "$BASH_VERSION" ] && npm () {
    # hack: avoid slow npm sanity check in nvm
    if [ "$_" == "config get prefix" ]; then which node | sed "s/bin/node/"
    else $which npm; fi
}
# [ -s "$NVM_DIR/nvm.sh" ] && . "$NVM_DIR/nvm.sh" # This loads nvm
#_silence_path_mismatch_check_flag=1 # prevent nvm complaints that nvm is first in PATH
unset nvm # end hack

# modifications needed only in interactive mode
if [ "$PS1" != "" ]; then
    # Set default editor for git
    git config --global core.editor nano

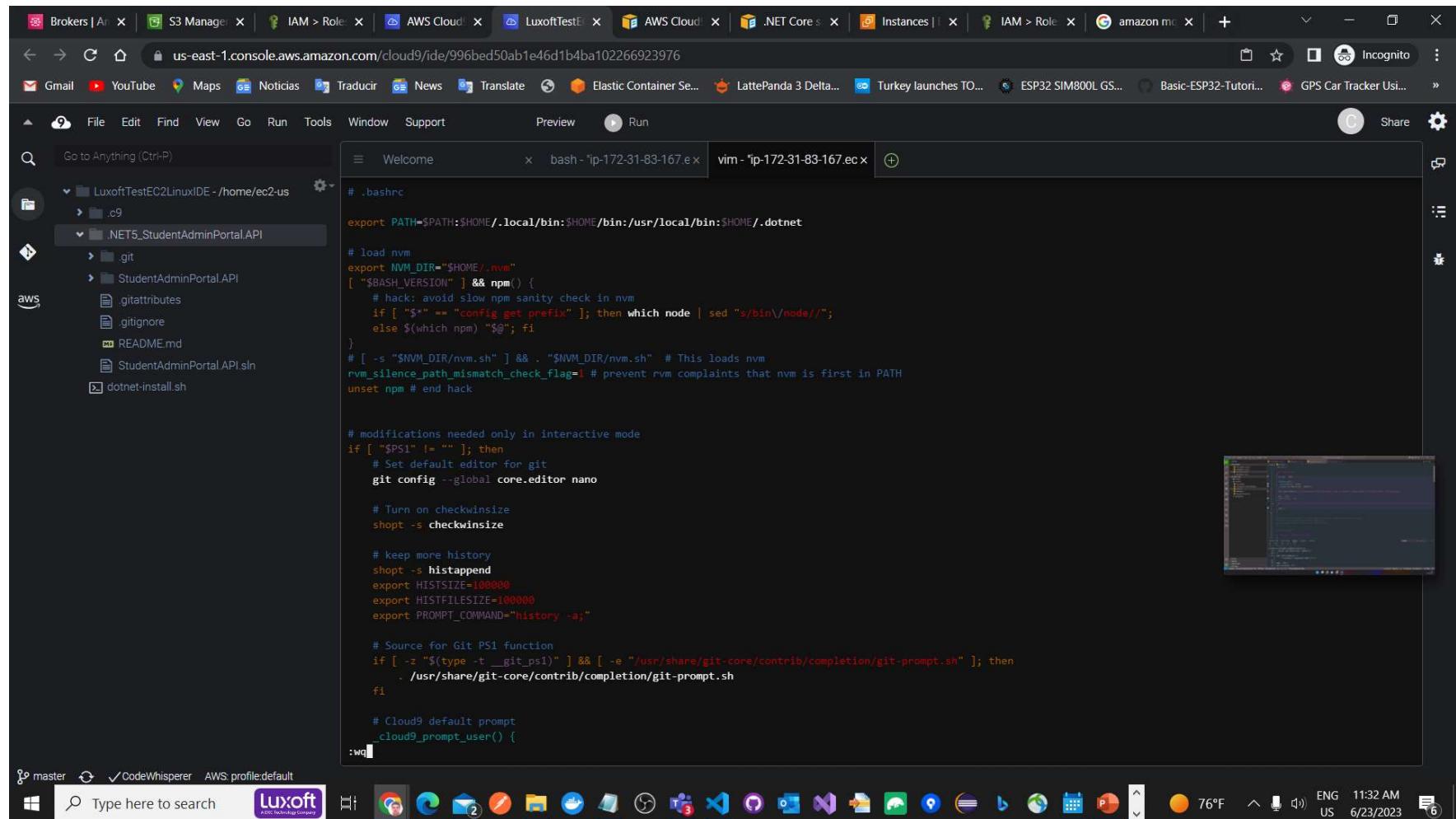
    # Turn on checkwintime
    shopt -s checkwintime

    # keep more history
    shopt -s histappend
    export HISTSIZE=100000
    export HISTFILESIZE=100000
    export PROMPT_COMMAND="history -a"

    # Source for Git PS1 function
    if [ -z "$(type -t __git_ps1)" ] && [ -e "/usr/share/git-core/contrib/completion/git-prompt.sh" ]; then
        /usr/share/git-core/contrib/completion/git-prompt.sh
    fi

    # Cloud9 default prompt
    cloud9_prompt_user()
-- INSERT --
```

After including the new line press **ESC** key to exit from the INSERT mode
Type **:wq** to save the changes



The screenshot shows a Windows desktop environment with a terminal window open in a web browser. The browser URL is `us-east-1.console.aws.amazon.com/cloud9/ide/996bed50ab1e46d1b4ba102266923976`. The terminal window title is "bash - ip-172-31-83-167.ecx". The code being edited is the `.bashrc` file, which contains configuration for NVM and Git. The terminal window has a dark theme. The status bar at the bottom shows "master" and "CodeWhisperer AWS.profile:default". The taskbar at the bottom includes icons for various applications like File Explorer, Edge, and VS Code.

```
# .bashrc

export PATH=$PATH:$HOME/.local/bin:$HOME/bin:/usr/local/bin:$HOME/.dotnet

# load nvm
export NVM_DIR="$HOME/.nvm"
[ "$BASH_VERSION" ] && npm() {
    # hack: avoid slow npm sanity check in nvm
    if [ "$_" == "config get prefix" ]; then which node | sed "s/bin/node//";
    else $(which npm) "$@"; fi
}
# [ -s "$NVM_DIR/nvm.sh" ] && . "$NVM_DIR/nvm.sh" # This loads nvm
nvm_silence_path_mismatch_check_flag=1 # prevent nvm complaints that nvm is first in PATH
unset npm # end hack

# modifications needed only in interactive mode
if [ "$PS1" != "" ]; then
    # Set default editor for git
    git config --global core.editor nano

    # Turn on checkwinsize
    shopt -s checkwinsize

    # keep more history
    shopt -s histappend
    export HISTSIZE=100000
    export HISTFILESIZE=100000
    export PROMPT_COMMAND="history -a"

    # Source for Git PS1 function
    if [ -z "$(type -t __git_ps1)" ] && [ -e "/usr/share/git-core/contrib/completion/git-prompt.sh" ]; then
        . /usr/share/git-core/contrib/completion/git-prompt.sh
    fi

    # Cloud9 default prompt
    __cloud9_prompt_user() {
:wq
```

Load the .NET Core SDK by sourcing the .bashrc file.

```
. ~/bashrc
```

Confirm the .NET Core SDK is loaded by running .NET Core CLI with the --help option.

```
dotnet --help
```

If you no longer want to keep the .NET Core SDK installer script in your environment, you can delete it as follows.

```
rm dotnet-install.sh
```

Confirm whether the latest version of the .NET Core SDK is already installed in your environment. To do this, in a terminal session in the AWS Cloud9 IDE, run the .NET Core command line interface (CLI) with the --version option.

```
dotnet --version
```

Step 2 (Optional): Install the .NET CLI extension for Lambda functions

Although not required for this tutorial, you can deploy AWS Lambda functions and AWS Serverless Application Model applications using the dotnet CLI if you also install the `Amazon.Lambda.Tools` package.

1. To install the package, run the following command:

```
dotnet tool install -g Amazon.Lambda.Tools
```



2. Now set the `PATH` and `DOTNET_ROOT` environment variable to point to the installed Lambda tool. In the `.bashrc` file, find the `export PATH` section and edit it so that it appears similar to the following (see Step 1 for details on editing this file):

```
export PATH=$PATH:$HOME/.local/bin:$HOME/bin:$HOME/.dotnet:$HOME/.dotnet/tools  
export DOTNET_ROOT=$HOME/.dotnet
```



To install the package, run the following command: `dotnet tool install -g Amazon.Lambda.Tools`

Now set the `PATH` and `DOTNET_ROOT` environment variable to point to the installed Lambda tool. In the `.bashrc` file, find the `export PATH` section and edit it so that it appears similar to the following (see Step 1 for details on editing this file):

```
export PATH=$PATH:$HOME/.local/bin:$HOME/bin:$HOME/.dotnet:$HOME/.dotnet/tools  
export DOTNET_ROOT=$HOME/.dotnet
```

Brokers | Amazon S3 | IAM > Roles | AWS CloudWatch Metrics | LuxoftTestEC2LinuxIDE | AWS CloudWatch Metrics | .NET Core Tools | Instances | IAM > Roles | Amazon CloudWatch Metrics | +

us-east-1.console.aws.amazon.com/cloud9/ide/996bed50ab1e46d1b4ba102266923976

Gmail YouTube Maps Noticias Traducir News Translate Elastic Container Service LattePanda 3 Delta... Turkey launches TO... ESP32 SIM800L GS... Basic-ESP32-Tutor... GPS Car Tracker Usi...

File Edit Find Go Run Tools Window Support Preview Run

Go to Anything (Ctrl+P)

LuxoftTestEC2LinuxIDE - /home/ec2-user/.c9

.NET5_StudentAdminPortal.API

- ..git
- StudentAdminPortal.API
- .gitattributes
- .gitignore
- README.md
- StudentAdminPortal.API.sln

dotnet-install.sh

Welcome bash - "ip-172-31-83-167.ec2.us-west-2.amazonaws.com" dotnet - "ip-172-31-83-167.ec2.us-west-2.amazonaws.com"

```
cloudUserluis:~/environment $ dotnet tool install -g Amazon.Lambda.Tools
Tools directory '/home/ec2-user/.dotnet/tools' is not currently on the PATH environment variable.
If you are using bash, you can add it to your profile by running the following command:

cat << \EOF >> ~/.bash_profile
# Add .NET Core SDK tools
export PATH="$PATH:/home/ec2-user/.dotnet/tools"
EOF

You can add it to the current session by running the following command:

export PATH="$PATH:/home/ec2-user/.dotnet/tools"

You can invoke the tool using the following command: dotnet-lambda
Tool 'amazon.lambda.tools' (version '5.7.0') was successfully installed.
cloudUserluis:~/environment $
```

aws



master ↵ ✓ CodeWhisperer AWS:profile:default

Type here to search Luxoft

76°F ENG US 11:40 AM 6/23/2023

Amazon.Lambda.Tools adds commands to the dotnet cli to deploy AWS Lambda functions.

The screenshot shows the NuGet.org website displaying the package page for 'Amazon.Lambda.Tools' version 5.7.0. The page includes the package logo, name, version, download statistics (Total: 14.1M, Current version: 217.4K, Per day average: 5.9K), and a preview image of the software interface. Navigation links for README, Frameworks, Dependencies, and Versions are visible. A note at the bottom states: 'Amazon.Lambda.Tools adds commands to the dotnet cli to deploy AWS Lambda functions.' The browser taskbar at the bottom shows various open tabs and system status.

Downloads

Total 14.1M

Current version 217.4K

Per day average 5.9K

Full stats →

aws Amazon.Lambda.Tools 5.7.0

Prefix Reserved

.NET 6.0 .NET Core 3.1

.NET CLI (Global) .NET CLI (Local) Cake NUKE

> dotnet tool install --global Amazon.Lambda.Tools --version 5.7.0

This package contains a .NET tool you can call from the shell/command line.

About

Last updated 21 days ago

Project website

License Info

Download package (12.42 MB)

Open in NuGet Package Explorer

Open in FuGet Package Explorer

Report package

Owners Contact owners →

aws awsdotnet

Type here to search Luxoft A DDC Technology Company

76°F ENG US 11:42 AM 6/23/2023

Screenshot of a browser window showing the GitHub repository for "aws/ aws-extensions-for-dotnet-cli". The repository page displays the code structure, commit history, and various extension categories.

The repository has 12 branches and 0 tags. The commit history shows several updates, including support for .NET Native AOT Lambda functions and version bumps of Amazon.Lambda.Tools.

The "About" section highlights extensions for AWS Lambda, AWS ECS, and AWS Elastic Beanstalk, which are circled in red.

Key repository statistics:

- 474 commits
- 3 weeks ago
- 7 months ago
- 4 years ago
- 3 weeks ago
- 3 weeks ago
- 2 years ago
- 5 years ago
- 3 years ago
- 6 years ago
- 6 years ago

Repository details:

- Code: Code (highlighted)
- Issues: 32
- Pull requests: 3
- Discussions: 0
- Actions: 0
- Projects: 0
- Security: 0
- Insights: 0

Branches: master (selected), 12 branches, 0 tags

Commits:

Commit Message	Time Ago
chore: update ancient issue time	last month
Add support for building and deploying .NET Native AOT Lambda functi...	7 months ago
Updated layers docs	4 years ago
Version bump Amazon.Lambda.Tools to 5.7.0	3 weeks ago
Fix unit test to check for .NET 6	3 weeks ago
Update all Lambda test to use .NET 6 instead of .NET Core 3.1 (#280)	3 weeks ago
Update tools and tests to at least .NET Core 3.1	2 years ago
Adding standard files	5 years ago
Added notes on local tool development workflow to help new contribut...	3 years ago
Creating initial file from template	6 years ago
Creating initial file from template	6 years ago

Extensions (highlighted): aws-lambda, aws-ecs, aws-elastic-beanstalk

Readme, Apache-2.0 license, Code of conduct, Security policy, 341 stars, 44 watching, 85 forks, Report repository

Releases

Search bar: Type here to search

Bottom navigation: luxoft, 76°F, ENG US, 11:42 AM, 6/23/2023

<https://github.com/aws/aws-extensions-for-dotnet-cli>

Step 3: Create a .NET Core console application project

In this step, you use .NET Core to create a project named `hello`. This project contains all of the files that .NET Core needs to run a simple application from the terminal in the IDE. The application's code is written in C#.

Create a .NET Core console application project. To do this, run the .NET Core CLI with the `new` command, specifying the console application project template type and the programming language to use (in this sample, C#).

The `-n` option indicates that the project is outputted to a new directory, `hello`. We then navigate to that directory.

```
dotnet new console -lang C# -n hello  
cd hello
```



The preceding command adds a subdirectory named `obj` with several files, and some additional standalone files, to the `hello` directory. You should note the following two key files:

- The `hello/hello.csproj` file contains information about the console application project.
- The `hello/Program.cs` file contains the application's code to run.

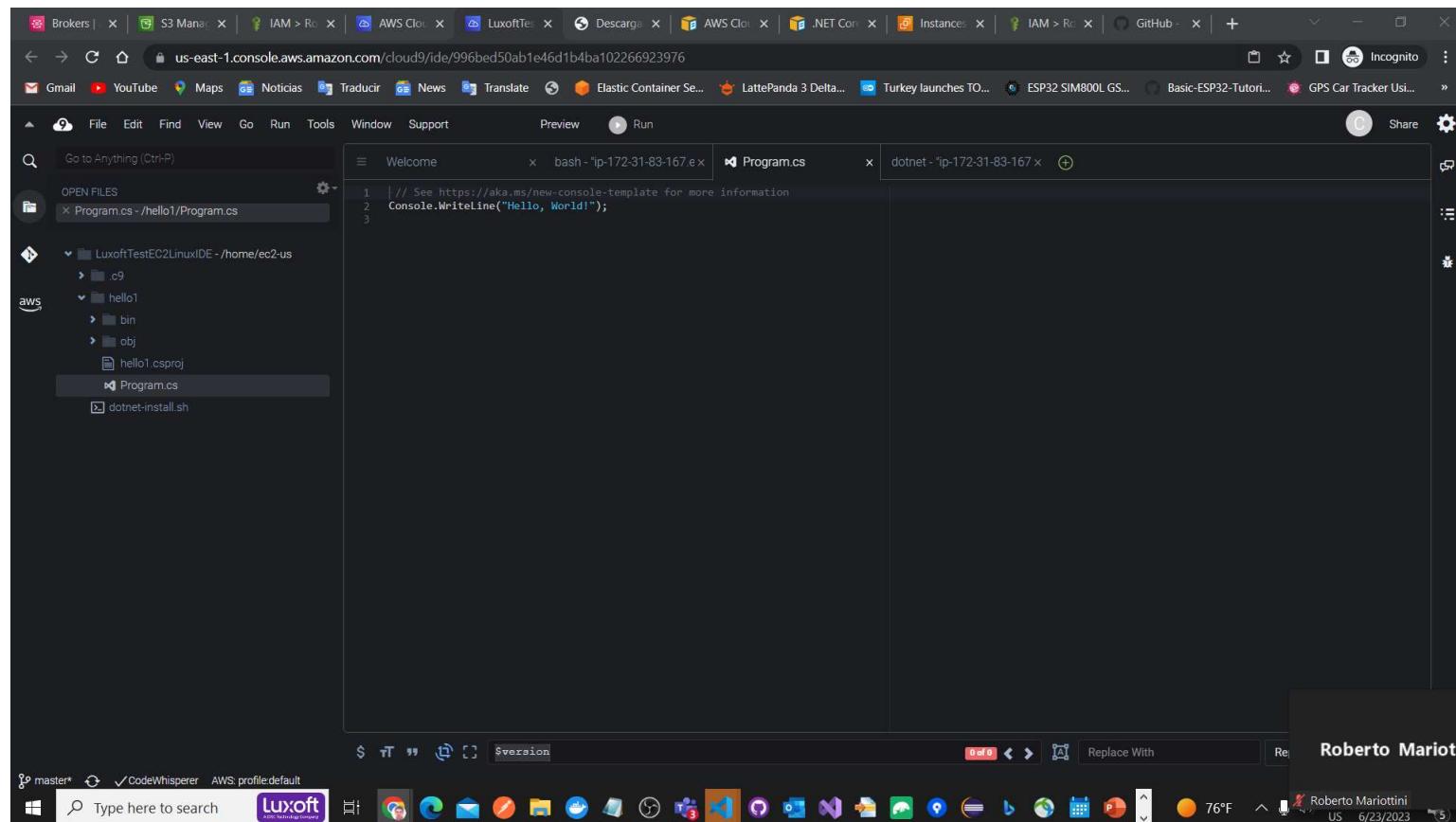
```
dotnet new console -lang C# -n hello  
cd hello
```

Step 4: Add code

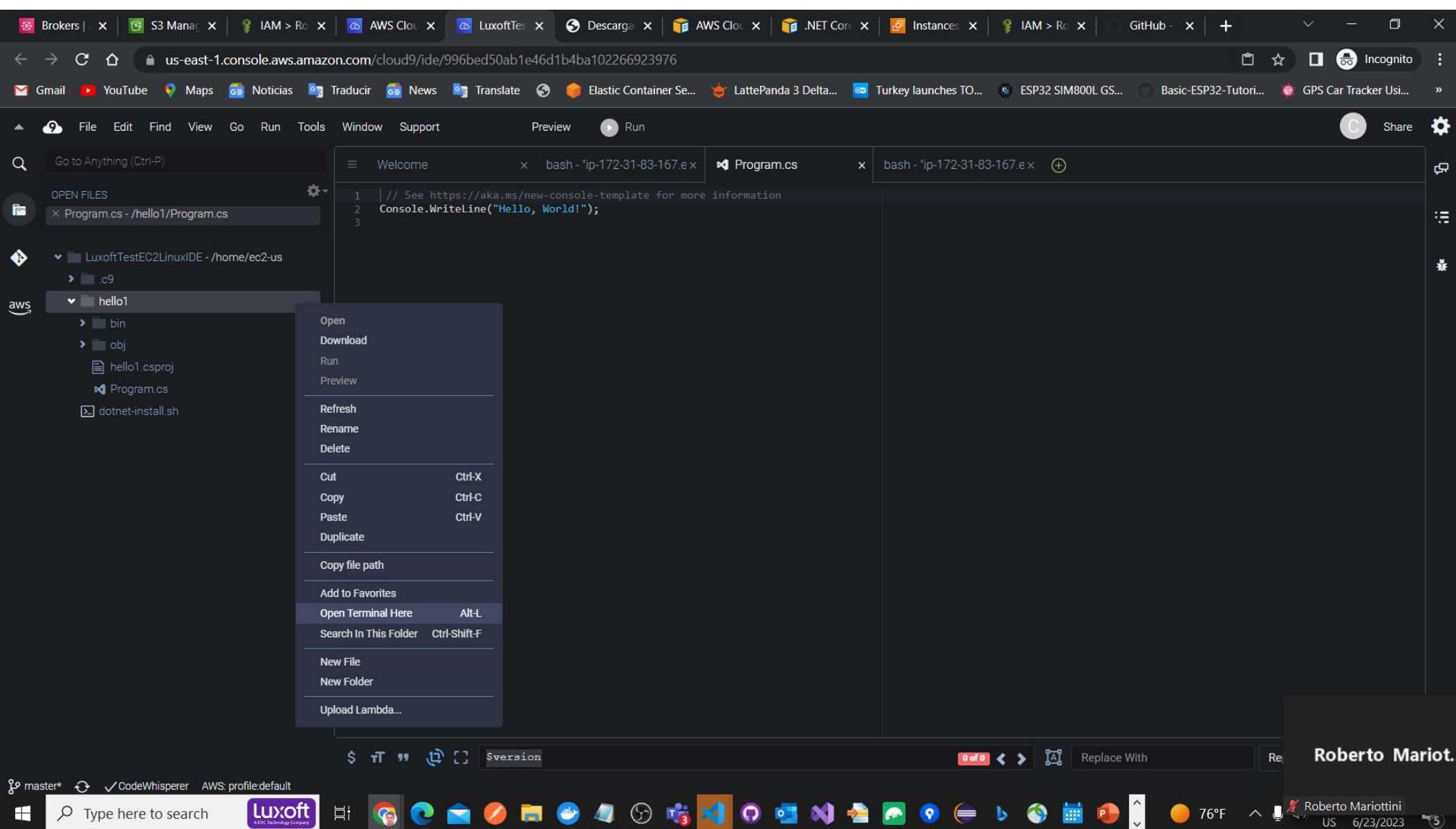
In this step, you add some code to the application.

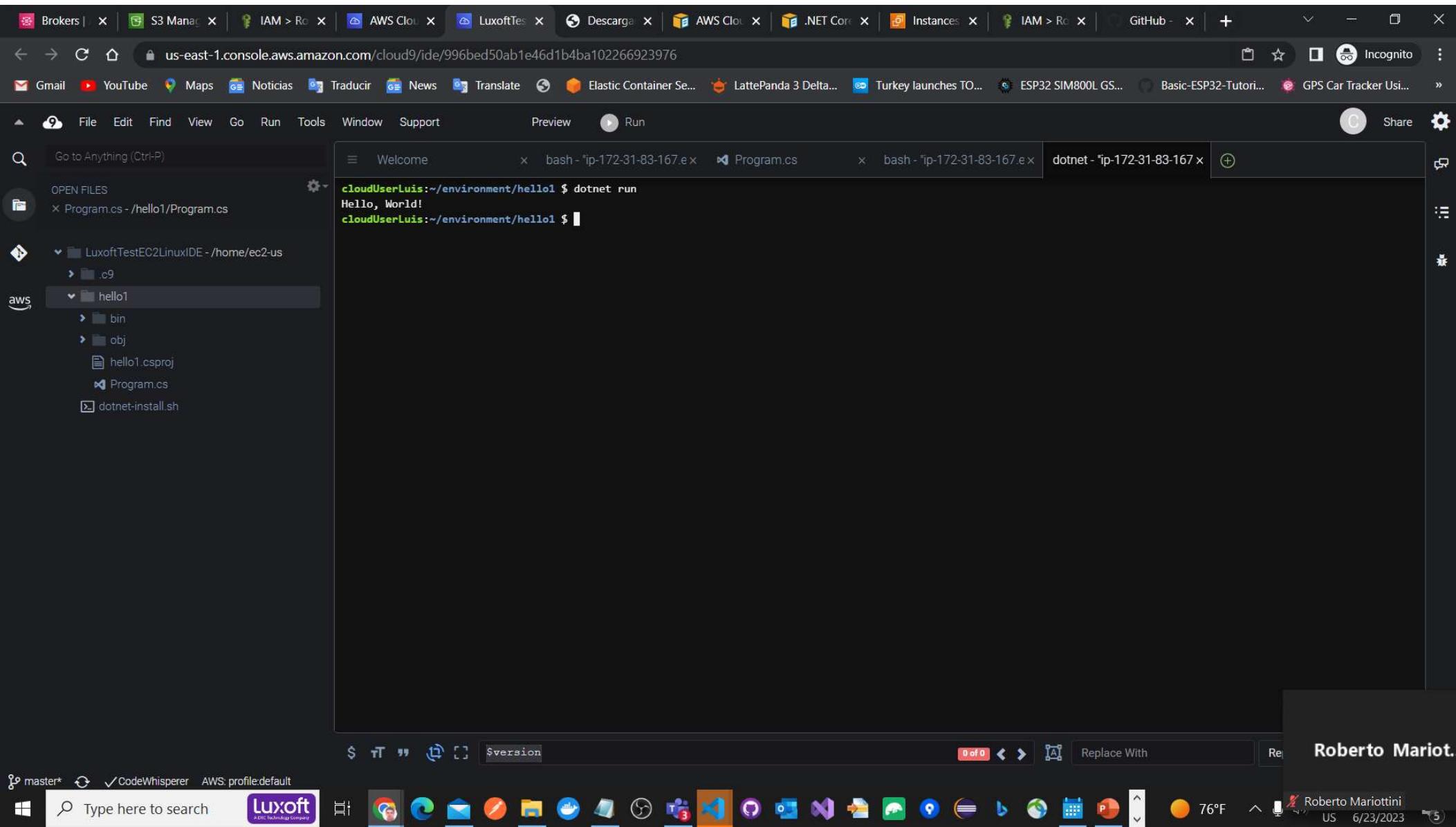
From the Environment window in the **AWS Cloud9 IDE**, open the **hello/Program.cs** file.

In the editor, replace the file's current contents with the following code, and then save the **Program.cs** file.



```
// See https://aka.ms/new-console-template for more information
Console.WriteLine("Hello, World!");
```





S Brokers | X S3 Manag X Instances X AWS Cloud X LuxoftTes X Descarga X AWS Clot X .NET Core X Instances X IAM > Re X GitHub - X +

Gmail YouTube Maps Noticias Traducir News Translate Elastic Container Se...

docs.aws.amazon.com/cloud9/latest/user-guide/sample-dotnetcore.html

Incognito

aws Buscar en esta guía Contacte con nosotros Español Volver a la consola Comentarios Preferencias

AWS Documentation AWS Cloud9 User Guide

▶ AWS Toolkit

▼ Tutorials and samples

- AWS CLI and AWS CloudShell sample
- AWS CodeCommit sample
- Amazon DynamoDB sample
- AWS CDK sample
- LAMP sample
- WordPress sample
- Java sample
- C++ sample
- Python tutorial

.NET Core sample

- Node.js sample
- PHP sample
- Ruby
- Go sample
- TypeScript sample
- Docker sample

▶ Advanced topics

▶ Security

Troubleshooting Supported browsers Limits Document history

Step 5: Build and run the code

In this step, you build the project and its dependencies into a set of binary files, including a runnable application file. Then you run the application.

1. In the IDE, create a builder for .NET Core as follows.
 - a. On the menu bar, choose **Run, Build System, New Build System**.
 - b. On the **My Builder.build** tab, replace the tab's contents with the following code.

```
{  
    "cmd" : ["dotnet", "build"],  
    "info" : "Building..."  
}
```

 - c. Choose **File, Save As**.
 - d. For **Filename**, type **.NET Core.build**.
 - e. For **Folder**, type **./c9/builders**.
 - f. Choose **Save**.- 2. With the contents of the **Program.cs** file displayed in the editor, choose **Run, Build System, .NET Core**. Then choose **Run, Build**. This builder adds a subdirectory named **bin** and adds a subdirectory named **Debug** to the **hello/obj** subdirectory. Note the following three key files.
 - The **hello/bin/Debug/netcoreapp3.1/hello.dll** file is the runnable application file.
 - The **hello/bin/Debug/netcoreapp3.1/hello.deps.json** file lists the application's dependencies.
 - The **hello/bin/Debug/netcoreapp3.1/hello.runtimeconfig.json** file specifies the shared runtime and its version for the application.

Note
The folder name, **netcoreapp3.1**, reflects the version of the .NET Core SDK used in this example. You may see a different number in the folder name depending on the version you've installed.

3. Create a runner for .NET Core as follows.

Like Comment

80°F ENG US 12:21 PM 6/23/2023

luxoft

S Brokers | X S3 Manag X Instances X AWS CloudWatch Metrics AWS CloudWatch Metrics Descarga X AWS CloudWatch Metrics .NET Core X Instances X IAM > Roles X GitHub - X +

docs.aws.amazon.com/cloud9/latest/user-guide/sample-dotnetcore.html

Gmail YouTube Maps Noticias Traducir News Translate Elastic Container Se...

aws Buscar en esta guía Contacte con nosotros Español Volver a la consola Comentarios Preferencias

AWS Documentation AWS Cloud9 User Guide

AWS Toolkit Tutorials and samples AWS CLI and AWS CloudShell sample AWS CodeCommit sample Amazon DynamoDB sample AWS CDK sample LAMP sample WordPress sample Java sample C++ sample Python tutorial .NET Core sample Node.js sample PHP sample Ruby Go sample TypeScript sample Docker sample Advanced topics Security Troubleshooting Supported browsers Limits Document history

3. Create a runner for .NET Core as follows.

- a. On the menu bar, choose Run, Run With, New Runner.
- b. On the My Runner.run tab, replace the tab's contents with the following code.

```
{  
    "cmd" : ["dotnet", "run", "$args"],  
    "working_dir": "$file_path",  
    "info" : "Running..."  
}
```

- c. Choose File, Save As.
- d. For Filename, type .NET Core.run.
- e. For Folder, type /.c9/runners.
- f. Choose Save.

4. Run the application with two integers to add (for example, 5 and 9) as follows.

- a. With the contents of the Program.cs file displayed in the editor, choose Run, Run Configurations, New Run Configuration.
- b. In the [New] - Idle tab, choose Runner: Auto, and then choose .NET Core.
- c. In the Command box, type hello 5 9.
- d. Choose Run.

By default, this runner instructs .NET Core to run the hello.dll file in the hello/bin/Debug/netcoreapp3.1 directory. Compare your output to the following.

```
Hello, World!  
The sum of 2 and 3 is 5.  
The sum of 5 and 9 is 14.
```

En esta página Prerequisites Step 1: Install required tools Step 2 (Optional): Install the .NET CLI extension for Lambda functions Step 3: Create a .NET Core console application project Step 4: Add code Step 5: Build and run the code Step 6: Create and set up a .NET Core console application project that uses the AWS SDK for .NET Step 7: Add AWS SDK code Step 8: Build and run the AWS SDK code Step 9: Clean up

80°F ENG US 12:21 PM 6/23/2023

S Brokers | X S3 Manag X IAM > Ro X AWS Cloud X LuxoftTes X Descarga X AWS Clou X .NET Core X Instances X IAM > Ro X GitHub - X +

Gmail YouTube Maps Noticias Traducir News Translate Elastic Container Se... LattePanda 3 Delta... Turkey launches TO... ESP32 SIM800L GS... Basic-ESP32-Tutori... GPS Car Tracker Usi...

docs.aws.amazon.com/cloud9/latest/user-guide/sample-dotnetcore.html

Incognito

aws

Buscar en esta guía Contacte con nosotros Español Volver a la consola

AWS Documentation AWS Cloud9 User Guide Comentarios Preferencias

▶ AWS Toolkit

▼ Tutorials and samples

- AWS CLI and AWS CloudShell sample
- AWS CodeCommit sample
- Amazon DynamoDB sample
- AWS CDK sample
- LAMP sample
- WordPress sample
- Java sample
- C++ sample
- Python tutorial

.NET Core sample

- Node.js sample
- PHP sample
- Ruby
- Go sample
- TypeScript sample
- Docker sample

▶ Advanced topics

▶ Security

Troubleshooting Supported browsers Limits Document history

Step 6: Create and set up a .NET Core console application project that uses the AWS SDK for .NET

You can enhance this sample to use the AWS SDK for .NET to create an Amazon S3 bucket, list your available buckets, and then delete the bucket you just created.

In this new project, you add a reference to the AWS SDK for .NET. The AWS SDK for .NET provides a convenient way to interact with AWS services such as Amazon S3, from your .NET code. You then set up AWS credentials management in your environment. The AWS SDK for .NET needs these credentials to interact with AWS services.

To create the project

1. Create a .NET Core console application project. To do this, run the .NET Core CLI with the `new` command, specifying the console application project template type and the programming language to use.
The `-n` option indicates that the project is outputted to a new directory, `s3`. We then navigate to that directory.

```
dotnet new console -lang C# -n s3
cd s3
```
2. Add a project reference to the Amazon S3 package in the AWS SDK for .NET. To do this, run the .NET Core CLI with the `add package` command, specifying the name of the Amazon S3 package in NuGet. (NuGet defines how packages for .NET are created, hosted, and consumed, and provides the tools for each of those roles.)

```
dotnet add package AWSSDK.S3
```

When you add a project reference to the Amazon S3 package, NuGet also adds a project reference to the rest of the AWS SDK for .NET.

Note
For the names and versions of other AWS related packages in NuGet, see [NuGet packages tagged with aws-sdk](#) on the NuGet website.

En esta página

Prerequisites Step 1: Install required tools Step 2 (Optional): Install the .NET CLI extension for Lambda functions Step 3: Create a .NET Core console application project Step 4: Add code Step 5: Build and run the code **Step 6: Create and set up a .NET Core console application project that uses the AWS SDK for .NET** Step 7: Add AWS SDK code Step 8: Build and run the AWS SDK code Step 9: Clean up

Roberto Mariottini 76°F US 6/23/2023

Step 6: Create and set up a .NET Core console application project that uses the AWS SDK for .NET

Create a .NET Core console application project. To do this, run the .NET Core CLI with the new command, specifying the console application project template type and the programming language to use.

The -n option indicates that the project is outputted to a new directory, s3. We then navigate to that directory.

```
dotnet new console -lang C# -n s3  
cd s3
```

Add a project reference to the Amazon S3 package in the AWS SDK for .NET. To do this, run the .NET Core CLI with the add package command, specifying the name of the Amazon S3 package in NuGet. (NuGet defines how packages for .NET are created, hosted, and consumed, and provides the tools for each of those roles.)

```
dotnet add package AWSSDK.S3
```

The screenshot shows a terminal window titled "Welcome" with the command "dotnet new console -lang C# -n s3" being run. The output indicates that the template "Console App" was created successfully, followed by processing post-creation actions, restoring the project, and a successful restore. The terminal is running on a Linux EC2 instance, as evidenced by the "aws" icon in the dock.

```
cloudUserLus:~/environment $ dotnet new console -lang C# -n s3  
The template "Console App" was created successfully.  
Processing post-creation actions...  
Restoring /home/ec2-user/environment/s3/s3.csproj:  
  Determining projects to restore...  
    Restored /home/ec2-user/environment/s3/s3.csproj (in 111 ms).  
Restore succeeded.
```

To set up AWS credentials management

Each time you use the AWS SDK for .NET to call an AWS service, you must provide a set of AWS credentials with the call. These credentials determine whether the AWS SDK for .NET has the appropriate permissions to make that call. If the credentials don't cover the appropriate permissions, the call will fail.

To store your credentials within the environment, follow the instructions in [Calling AWS services from an environment in AWS Cloud9](#), and then return to this topic.

For additional information, see [Configuring AWS Credentials](#) in the *AWS SDK for .NET Developer Guide*.

<https://docs.aws.amazon.com/cloud9/latest/user-guide/credentials.html>

Brokers | X S3 Manager | X IAM > Roles | X AWS CloudWatch Metrics | X Luxoft Test Automation | X Descarga | X AWS CloudWatch Metrics | X Calling API | X Instances | X IAM > Roles | X GitHub - X +

docs.aws.amazon.com/cloud9/latest/user-guide/credentials.html

Gmail YouTube Maps Noticias Traducir News Translate Elastic Container Se... LattePanda 3 Delta... Turkey launches TO... ESP32 SIM800L GS... Basic-ESP32-Tutori... GPS Car Tracker Usi...

aws Buscar en esta guía Contacte con nosotros Español Volver a la consola

AWS Documentation AWS Cloud9 User Guide Comentarios Preferencias

Create an instance profile with the IAM console

Note
If you already have an IAM role that contains an instance profile, skip ahead to [Attach an Instance Profile to an Instance with the Amazon EC2 Console](#).

1. Sign in to the IAM console, at <https://console.aws.amazon.com/iam>.
For this step, we recommend you sign in using administrator-level credentials in your AWS account. If you can't do this, check with your AWS account administrator.
2. In the navigation bar, choose **Roles**.

Note
You cannot use the IAM console to create an instance profile by itself. You must create an IAM role, which contains an instance profile.

3. Choose **Create role**.
4. On the **Select type of trusted entity** page, with **AWS service** already chosen, for **Choose the service that will use this role**, choose **EC2**.
5. For **Select your use case**, choose **EC2**.
6. Choose **Next: Permissions**.
7. On the **Attach permissions policies** page, in the list of policies, select the box next to **AdministratorAccess**, and then choose **Next: Review**.

Note
The **AdministratorAccess** policy allows unrestricted access to all AWS actions and resources across your AWS account. Use it only for experimentation purposes. For more information, see [IAM Policies in the IAM User Guide](#).

8. On the **Review** page, for **Role Name**, enter a name for the role (for example, `my-demo-cloud9-instance-profile`).
9. Choose **Create Role**.

Skip ahead to [Attach an Instance Profile to an Instance with the Amazon EC2 Console](#).

En esta página Create and use an instance profile to manage temporary credentials Create and store permanent access credentials in an Environment

Roberto Mariot..

Type here to search Luxoft ENG US 12:04 PM 6/23/2023 5

Create a new Role to attach to the EC2 virtual machine where is hosted the AWS Cloud9 IDE

The screenshot shows the AWS Identity and Access Management (IAM) service interface. The left sidebar is titled "Identity and Access Management (IAM)" and includes sections for "Dashboard", "Access management" (with "Roles" selected), "Access reports", and "Organization activity". The main content area is titled "Roles (29)" and contains a table listing 29 IAM roles. The columns in the table are "Role name", "Trusted entities", and "Last activity". The table lists various roles such as "AnotherFunction-role-dgk8bsb0", "aws-ec2-spot-fleet-tagging-role", "AWSCloud9SSMAccessRole", etc. A "Create role" button is located at the top right of the table. The browser address bar shows the URL "us-east-1.console.aws.amazon.com/iamv2/home?region=us-east-1#/roles". The bottom of the screen shows the Windows taskbar with various pinned icons and system status information.

Role name	Trusted entities	Last activity
AnotherFunction-role-dgk8bsb0	AWS Service: lambda	221 days ago
aws-ec2-spot-fleet-tagging-role	AWS Service: spotfleet	-
AWSCloud9SSMAccessRole	AWS Service: cloud9, and 1 more.	1 hour ago
AwsMicroservicesStack-productApiCloudWatchRole3F00-1DNCVGYU7NNYD	AWS Service: apigateway	217 days ago
AWSServiceRoleForAmazonMQ	AWS Service: mq (Service-Linked Role)	2 hours ago
AWSServiceRoleForAPIGateway	AWS Service: ops.apigateway (Service-Linked Role)	-
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable	AWS Service: dynamodb.application-autoscaling (Service-L...	218 days ago
AWSServiceRoleForAutoScaling	AWS Service: autoscaling (Service-Linked Role)	7 days ago
AWSServiceRoleForAWSCloud9	AWS Service: cloud9 (Service-Linked Role)	1 hour ago
AWSServiceRoleForECS	AWS Service: ecs (Service-Linked Role)	189 days ago

S Brokers X | RabbitM X | S3 Man X | IAM > F X | AWS Cl X | Luxoft X | Descarg X | AWS Cl X | Calling X | Instances X | IAM > F X | GitHub X | + | Incognito

Gmail YouTube Maps Noticias Traducir News Translate Elastic Container Se... LattePanda 3 Delta... Turkey launches TO... ESP32 SIM800L GS... Basic-ESP32-Tutori... GPS Car Tracker Usi...

aws Services Search [Alt+S]

IAM > Roles > Create role

Step 1 Select trusted entity

Step 2 Add permissions

Step 3 Name, review, and create

Select trusted entity Info

Trusted entity type

- AWS service Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- Web identity Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- SAML 2.0 federation Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- Custom trust policy Create a custom trust policy to enable others to perform actions in this account.

Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Common use cases

- EC2 Allows EC2 instances to call AWS services on your behalf.
- Lambda Allows Lambda functions to call AWS services on your behalf.

Use cases for other AWS services:

Choose a service to view use case

CloudShell Feedback Language

Roberto Mariot..

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy

Type here to search Luxoft A DIC Technology Company

76°F ENG US 12:08 PM 6/23/2023

Brokers X | RabbitMQ X | S3 Man X | IAM > F X | AWS CLI X | Luxoft X | Descard X | AWS CLI X | Calling X | Instances X | IAM > I X | GitHub X | + | Incognito

Gmail YouTube Maps Noticias Traducir News Translate Elastic Container Se... LattePanda 3 Delta... Turkey launches TO... ESP32 SIM800L GS... Basic-ESP32-Tutori... GPS Car Tracker Usi...

aws Services Search [Alt+S]

Step 1 Select trusted entity

Step 2 Add permissions

Step 3 Name, review, and create

Add permissions Info

Permissions policies (Selected 1/868) Info

Choose one or more policies to attach to your new role.

Filter policies by property or policy name and press enter. 4 matches

"AdministratorAccess"

Policy name <small>Info</small>	Type	Description
<input checked="" type="checkbox"/> AdministratorAccess <small>AWS managed policy</small>	AWS managed policy	Provides full access to AWS services and resources.
<input type="checkbox"/> AdministratorAccess-Amplify <small>AWS managed policy</small>	AWS managed policy	Grants account administrative permissions while explicitly allowing direct ...
<input type="checkbox"/> AWSAuditManagerAdministratorAccess <small>AWS managed policy</small>	AWS managed policy	Provides administrative access to enable or disable AWS Audit Manager, u...
<input type="checkbox"/> AdministratorAccess-AWSElasticBeanstalk <small>AWS managed policy</small>	AWS managed policy	Grants account administrative permissions. Explicitly allows developers an...

Set permissions boundary - optional Info

Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.

Cancel Previous Next

Step 1 Select trusted entity

Step 2 Add permissions

Step 3 Name, review, and create

Name, review, and create

Role details

Role name

Enter a meaningful name to identify this role.

Maximum 64 characters. Use alphanumeric and '+=_,@_-' characters.

Description

Add a short explanation for this role.

Maximum 1000 characters. Use alphanumeric and '+=_,@_-' characters.

Step 1: Select trusted entities

Edit

```
1 [{}  
2     "Version": "2012-10-17",  
3     "Statement": [  
4         {  
5             "Effect": "Allow",  
6             "Action": [  
7                 "sts:AssumeRole"  
8             ],  
9             "Principal": {  
10                 "Service": [
```

S Brokers X | RabbitMQ X | S3 Man X | IAM > X | AWS CLI X | Luxoft X | Descarg X | AWS CLI X | Calling X | Instances X | IAM > X | GitHub X | + | Incognito

Gmail YouTube Maps Noticias Traducir News Translate Elastic Container Se... LattePanda 3 Delta... Turkey launches TO... ESP32 SIM800L GS... Basic-ESP32-Tutori... GPS Car Tracker Usi...

aws Services Search [Alt+S]

Step 2: Add permissions

Permissions policy summary

Policy name	Type	Attached as
AdministratorAccess	AWS managed - job function	Permissions policy

Tags

Add tags - optional Info

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

Add tag

You can add up to 50 more tags.

Cancel Previous Create role

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Type here to search Luxoft ENG 76°F 12:09 PM US 6/23/2023

S Brokers X | RabbitM X | S3 Man X | IAM > X | AWS Cl X | LuxoftT X | Descarg X | AWS Cl X | Calling X | Instances X | IAM > X | GitHub X | + | Incognito

Gmail YouTube Maps Noticias Traducir News Translate Elastic Container Se... LattePanda 3 Delta... Turkey launches TO... ESP32 SIM800L GS... Basic-ESP32-Tutori... GPS Car Tracker Usi...

aws Services Search [Alt+S] View role

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

User groups

Users

Roles

Policies

Identity providers

Account settings

Access reports

Access analyzer

Archive rules

Analyzers

Settings

Credential report

Organization activity

CloudShell Feedback Language

luxoft

us-east-1.console.aws.amazon.com/iamv2/home?region=us-east-1#roles

Role my-demo-cloud9-instance-profile created.

IAM > Roles

Roles (30) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Search: my 4 matches 1 / 1

Role name	Trusted entities	Last activity
my-demo-cloud9-instance-profile	AWS Service: ec2	1 hour ago
my-function-role-fbomtqg	AWS Service: lambda	-
my-function-role-gftv1utw	AWS Service: lambda	218 days ago
my-function-role-qq406q7r	AWS Service: lambda	220 days ago

Roles Anywhere Info

Authenticate your non AWS workloads and securely provide access to AWS services.

Manage

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Type here to search 76°F ENG US 12:10 PM 6/23/2023

Attach the new Role to the EC2 virtual machine where is hosted the AWS Cloud9 IDE

The screenshot shows the AWS IAM (Identity and Access Management) console. The URL in the address bar is <https://us-east-1.console.aws.amazon.com/iamv2/home?region=us-east-1#/roles>. The search bar at the top contains the text 'ec2'. On the left, the navigation menu is expanded to show 'Identity and Access Management (IAM)' and its sub-sections: 'Dashboard', 'Access management' (with 'User groups', 'Users', and 'Roles' selected), 'Access reports', and 'Organization activity'. The main content area displays search results for 'ec2' under 'Services'. The first result is 'EC2' (Virtual Servers in the Cloud), which is highlighted. Below it are 'EC2 Image Builder' (A managed service to automate build, customize and deploy OS images), 'Recycle Bin' (Protect resources from accidental deletion), and 'Amazon Inspector' (Continual vulnerability management at scale). To the right of the search results, a modal window titled 'Create role' is open. It contains fields for 'Role name' (with 'Cloud9 IDE' typed in), 'AWS Identity and Access Management (IAM)' (selected), and 'AWS Lambda' (selected). Under 'Permissions', there is a single policy named 'AdministratorAccess'. At the bottom of the modal are buttons for 'Cancel', 'Delete', and 'Create role'. The status bar at the bottom of the browser window shows the URL <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1>, the date and time '12:11 PM 6/23/2023', and the temperature '76°F'.

Brokers X | RabbitMQ X | S3 Man X | Instances X | AWS CLI X | Luxoft X | Descarg X | AWS CLI X | Calling X | Instances X | IAM X | GitHub X | + | Incognito

Gmail YouTube Maps Noticias Traducir News Translate Elastic Container Se... LattePanda 3 Delta... Turkey launches TO... ESP32 SIM800L GS... Basic-ESP32-Tutori... GPS Car Tracker Usi...

aws Services Search [Alt+S]

Instances (1/1) Info Find instance by attribute or tag (case-sensitive)

New EC2 Experience Tell us what you think

EC2 Dashboard EC2 Global View Events Limits Instances Instances Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts Scheduled Instances Capacity Reservations Images AMIs AMI Catalog Elastic Block Store Volumes

aws-cloud9-LuxoftTestEC2LinuxIDE-996bed50ab1e46d1b4ba102266923976

Details Security Networking Storage Status checks Monitoring Tags

Instance summary Info

Instance ID i-0cf66948d11a78016 (aws-cloud9-LuxoftTestEC2LinuxIDE-996bed50ab1e46d1b4ba102266923976)	Public IPv4 address 54.173.204.99 open address ↗	Private IPv4 addresses 172.31.83.167
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-54-173-204-99.compute-1.amazonaws.com open address ↗
Hostname type	Private IP DNS name (IPv4 only)	

Type here to search Luxoft 76°F ENG US 12:11 PM 6/23/2023

Instances (1/1) Info Find instance by attribute or tag (case-sensitive)

Name Instance ID Instance state Instance type Status check

aws-cloud9-LuxoftTestEC2LinuxIDE-996bed50ab1e46d1b4ba102266923976 i-0cf66948d11a78016 Running t2.micro 2/2 checks passed

Actions Connect Instance state Actions Launch instances

Connect View details Manage instance state Instance settings Networking Security Image and templates Modify IAM role Monitor and troubleshoot

Change security groups Get Windows password

Instance: i-0cf66948d11a78016 (aws-cloud9-LuxoftTestEC2LinuxIDE-996bed50ab1e46d1b4ba102266923976)

Details Security Networking Storage Status checks Monitoring Tags

Instance summary Info

Instance ID i-0cf66948d11a78016 (aws-cloud9-LuxoftTestEC2LinuxIDE-996bed50ab1e46d1b4ba102266923976)	Public IPv4 address 54.173.204.99 open address ↗	Private IPv4 addresses 172.31.83.167
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-54-173-204-99.compute-1.amazonaws.com open address ↗
Hostname type	Private IP DNS name (IPv4 only)	

Type here to search Luxoft 76°F ENG US 12:11 PM 6/23/2023

S Brokers X | RabbitM X | S3 Man X | AWS Cl X | LuxoftT X | Descarg X | AWS Cl X | Calling X | Instances X | IAM > X | GitHub X | + | Incognito

Gmail YouTube Maps Noticias Traducir News Translate Elastic Container Se... LattePanda 3 Delta... Turkey launches TO... ESP32 SIM800L GS... Basic-ESP32-Tutori... GPS Car Tracker Usi...

aws Services Search [Alt+S]

EC2 > Instances > i-0cf66948d11a78016 > Modify IAM role

Modify IAM role Info

Attach an IAM role to your instance.

Instance ID
i-0cf66948d11a78016 (aws-cloud9-LuxoftTestEC2LinuxIDE-996bed50ab1e46d1b4ba102266923976)

IAM role
Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.

AWSCloud9SSMInstanceProfile

No IAM Role
Choose this option to detach an IAM role

AWSCloud9SSMInstanceProfile arn:aws:iam::550146943653:instance-profile/cloud9/AWSCloud9SSMInstanceProfile

my-demo-cloud9-instance-profile arn:aws:iam::550146943653:instance-profile/my-demo-cloud9-instance-profile my-demo-cloud9-instance-profile

Cancel Update IAM role

Type here to search Luxoft 76°F ENG US 12:11 PM 6/23/2023

S Brokers X | RabbitM X | S3 Man X | AWS Cl X | Luxoft X | Descarg X | AWS Cl X | Calling X | Instances X | IAM > X | GitHub X | + | Incognito

Gmail YouTube Maps Noticias Traducir News Translate Elastic Container Se... LattePanda 3 Delta... Turkey launches TO... ESP32 SIM800L GS... Basic-ESP32-Tutori... GPS Car Tracker Usi...

aws Services Search [Alt+S]

EC2 > Instances > i-0cf66948d11a78016 > Modify IAM role

Modify IAM role Info

Attach an IAM role to your instance.

Instance ID
i-0cf66948d11a78016 (aws-cloud9-LuxoftTestEC2LinuxIDE-996bed50ab1e46d1b4ba102266923976)

IAM role
Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.

my-demo-cloud9-instance-profile Create new IAM role

Cancel

Type here to search Luxoft 76°F ENG US 12:11 PM 6/23/2023

Step 7: Add AWS SDK code

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

From the Environment window in the AWS Cloud9 IDE, open the `s3/Program.cs` file. In the editor, replace the file's current contents with the following code, and then save the `Program.cs` file.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using Amazon.S3.Util;
using System;
using System.Threading.Tasks;

namespace s3
{
    class Program
    {
        async static Task Main(string[] args)
        {
            if (args.Length < 2) {
                Console.WriteLine("Usage: <the bucket name> <the AWS Region to use>");
                Console.WriteLine("Example: my-test-bucket us-east-2");
                return;
            }

            if (args[1] != "us-east-2") {
                Console.WriteLine("Cannot continue. The only supported AWS Region ID is " +
                    "'us-east-2'.");
                return;
            }

            var bucketRegion = RegionEndpoint.USEast2;
            // Note: You could add more valid AWS Regions above as needed.

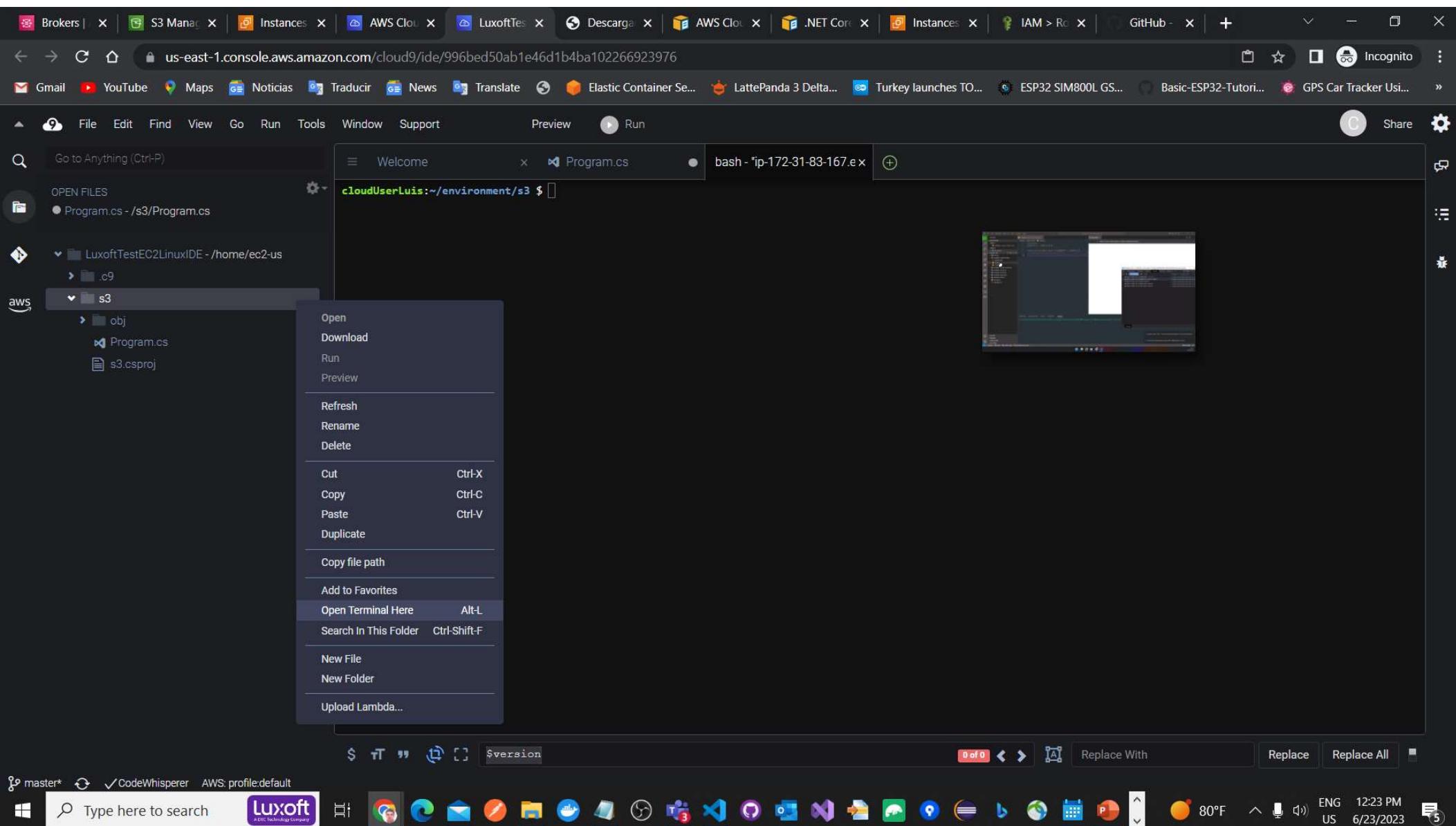
            using (var s3Client = new AmazonS3Client(bucketRegion)) {
                var bucketName = args[0];

                // Create the bucket.
                try
                {
```

The screenshot shows a browser window with multiple tabs open, including 'Brokers', 'S3 Manager', 'Instances', 'AWS CloudWatch Metrics', 'Luxoft TestEC2LinuxIDE', 'Descarga', 'AWS CloudWatch Metrics', '.NET Core', 'Instances', 'IAM > Roles', 'GitHub', and 'Incognito'. The main content area displays a code editor for a C# file named 'Program.cs'.

```
1  using Amazon;
2  using Amazon.S3;
3  using Amazon.S3.Model;
4  using Amazon.S3.Util;
5  using System;
6  using System.Threading.Tasks;
7
8  namespace s3
9  {
10    class Program
11    {
12      [B]async static Task Main(string[] args)[E]
13      {
14        if (args.Length < 2) {
15          Console.WriteLine("Usage: <the bucket name> <the AWS Region to use>");
16          Console.WriteLine("Example: my-test-bucket us-east-2");
17          return;
18        }
19
20        if (args[1] != "us-east-2") {
21          Console.WriteLine("Cannot continue. The only supported AWS Region ID is " +
22            "'us-east-2'.");
23          return;
24        }
25
26        var bucketRegion = RegionEndpoint.USEast2;
27        // Note: You could add more valid AWS Regions above as needed.
28
29        using (var s3Client = new AmazonS3Client(bucketRegion)) {
30          var bucketName = args[0];
31
32          // Create the bucket.
33          try
34          {
35            if (await AmazonS3Util.DoesS3BucketExistV2Async(s3Client, bucketName))
36            {
37              Console.WriteLine("Cannot continue. Cannot create bucket. \n" +
38                "A bucket named '{0}' already exists.", bucketName);
39            } else {
40              Console.WriteLine("\nCreating the bucket named '{0}'...", bucketName);
41            }
42          }
43        }
44      }
45    }
46  }
```

The code is a C# program that creates an S3 bucket. It uses the Amazon.S3 library. The program expects two command-line arguments: the bucket name and the AWS Region ID. It checks if the Region ID is 'us-east-2'. If it's not, it prints an error message and exits. Otherwise, it creates a new bucket with the specified name in the 'us-east-2' region.



For running the application run the command:

dotnet run my-bucket-luiscocoenriquez us-east-2

The screenshot shows a terminal window with the following session history:

```
cloudUserLuis:~/environment/s3 $ dotnet run my-bucket-luiscoco us-east-2
Creating the bucket named 'my-bucket-luiscoco'...
Created the bucket named 'my-bucket-luiscoco'.

Deleting the bucket named 'my-bucket-luiscoco'...
Deleted the bucket named 'my-bucket-luiscoco'.

My buckets now are:
cdk-hnb659fds-assets-550146943653-eu-west-3
cloudUserLuis:~/environment/s3 $
```

The terminal is part of a larger interface, likely a cloud-based development environment. The sidebar on the left shows project files like `Program.cs`, `s3.csproj`, and AWS-related items such as `LuxoftTestEC2LinuxIDE`. The bottom of the screen shows a Windows taskbar with various icons and system status.