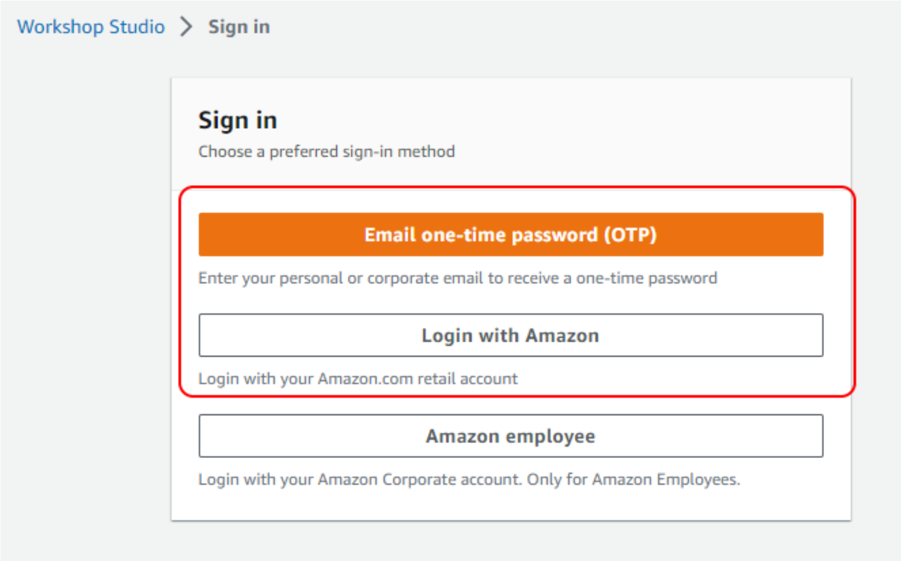
Developer Guide for Hackathon using Python Language - Sep 2023

Introduction:

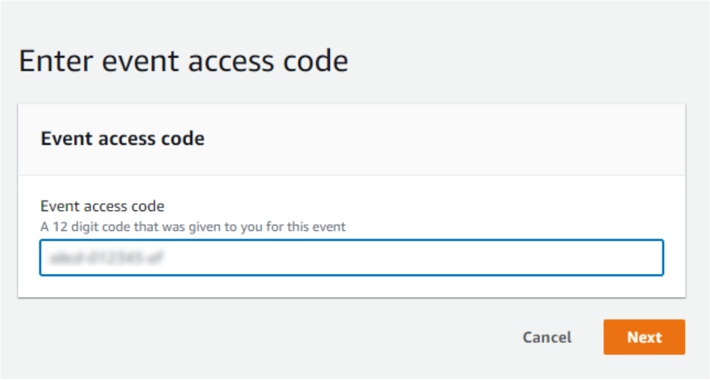
This document explains step by step how to use Apprunner and other AWS Services during the Hackathon for CITI developers, Also this showing a Sample code using **Python Language** running on Apprunner .

Accessing AWS environment for the Hackathon

Navigate to the link provided to you by the instructor. If you do not have a specific URL for this event, you can navigate to https://catalog.us-east-1.prod.workshops.aws/join and you will be prompted for an event code later.
You will need to login to the environment using one of the two indicated methods below.


Workshop Studio > Sign in
Sign in
Choose a preferred sign-in method
Email one-time password (OTP)
Enter your personal or corporate email to receive a one-time password
Login with Amazon
Login with your Amazon.com retail account
Amazon employee
Login with your Amazon Corporate account. Only for Amazon Employees.

If you are prompted for an event code, enter the value supplied by the lab instructor.



Enter event access code

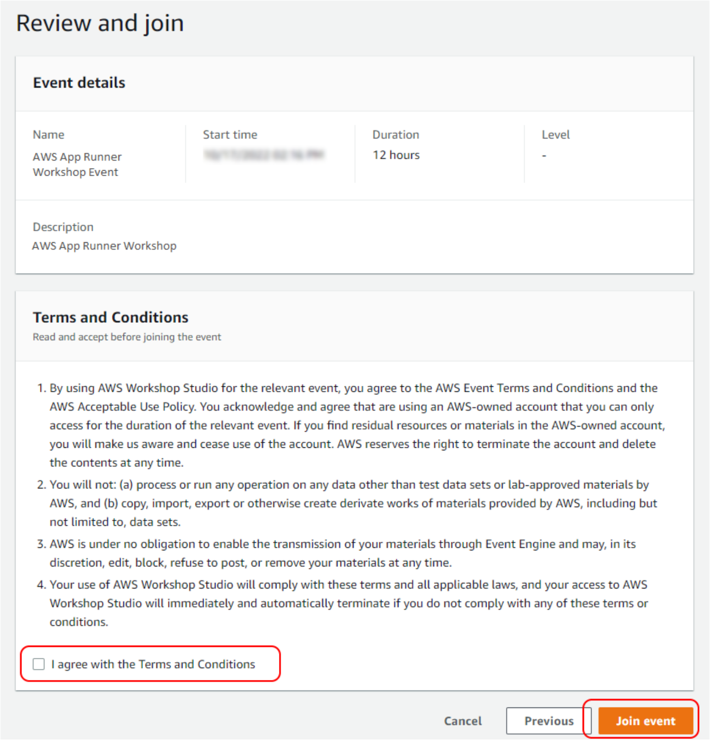
Event access code

Event access code
A 12 digit code that was given to you for this event

[redacted]

Finish the login process

You will be prompted to accept the terms and conditions of the event. Read the page, and then select the **I agree with the Terms and Conditions** checkbox. Choose **Join event**.



Review and join

Event details

Name	Start time	Duration	Level
AWS App Runner Workshop Event	[redacted]	12 hours	-

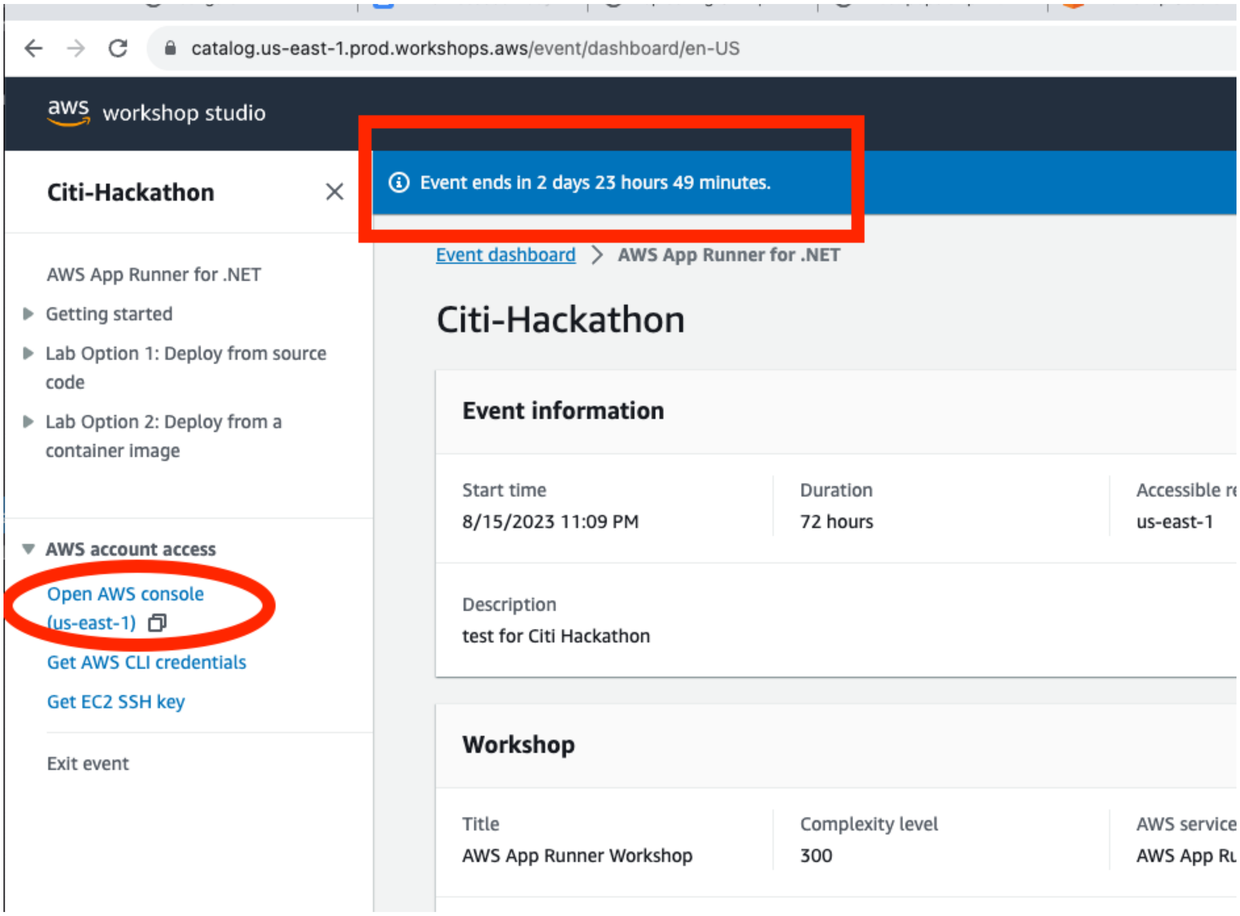
Description
AWS App Runner Workshop

Terms and Conditions
Read and accept before joining the event

1. By using AWS Workshop Studio for the relevant event, you agree to the AWS Event Terms and Conditions and the AWS Acceptable Use Policy. You acknowledge and agree that are using an AWS-owned account that you can only access for the duration of the relevant event. If you find residual resources or materials in the AWS-owned account, you will make us aware and cease use of the account. AWS reserves the right to terminate the account and delete the contents at any time.
2. You will not: (a) process or run any operation on any data other than test data sets or lab-approved materials by AWS, and (b) copy, import, export or otherwise create derivate works of materials provided by AWS, including but not limited to, data sets.
3. AWS is under no obligation to enable the transmission of your materials through Event Engine and may, in its discretion, edit, block, refuse to post, or remove your materials at any time.
4. Your use of AWS Workshop Studio will comply with these terms and all applicable laws, and your access to AWS Workshop Studio will immediately and automatically terminate if you do not comply with any of these terms or conditions.

☐ I agree with the Terms and Conditions

The workshop dashboard will be shown. At any time, to log into the AWS console, choose the **Open AWS Console** link in the **AWS account access menu**.



catalog.us-east-1.prod.workshops.aws/event/dashboard/en-US

Citi-Hackathon

AWS App Runner for .NET

- Getting started
- Lab Option 1: Deploy from source code
- Lab Option 2: Deploy from a container image

AWS account access

Open AWS console (us-east-1)

Get AWS CLI credentials

Get EC2 SSH key

Exit event

Event ends in 2 days 23 hours 49 minutes.

Event dashboard > AWS App Runner for .NET

Citi-Hackathon

Event information

Start time	Duration	Accessible re
8/15/2023 11:09 PM	72 hours	us-east-1

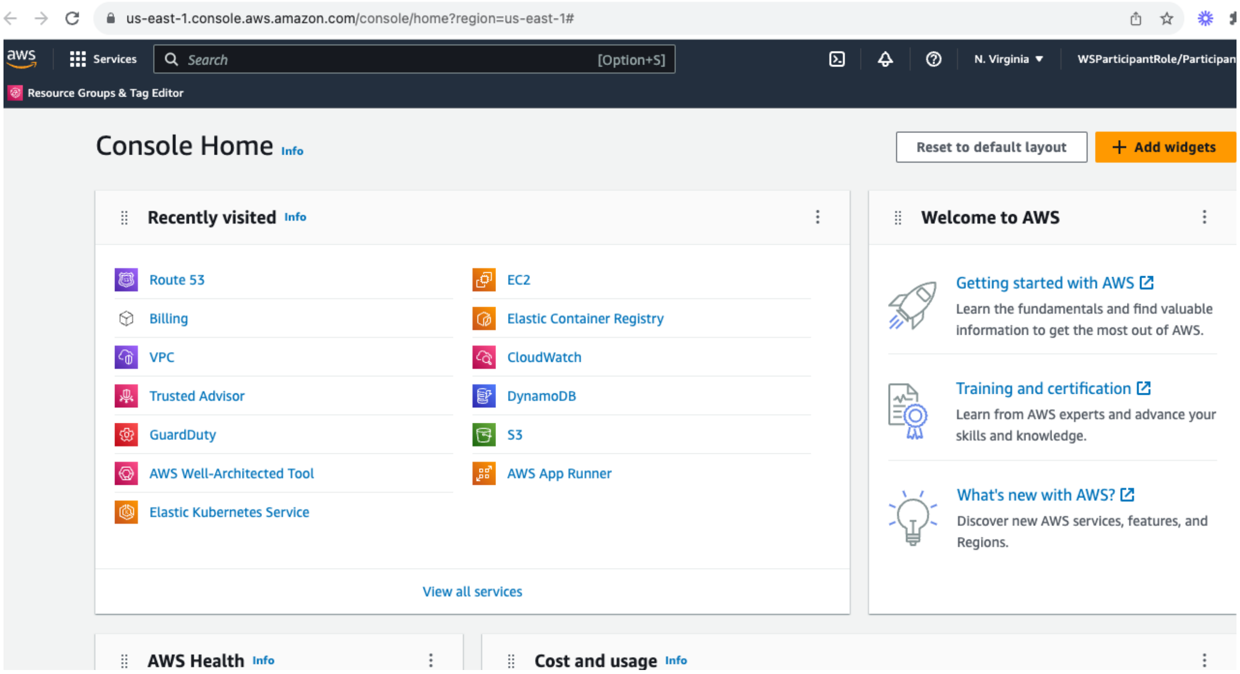
Description

test for Citi Hackathon

Workshop

Title	Complexity level	AWS service
AWS App Runner Workshop	300	AWS App Ru

Make sure you have access to AWS Services in the Console as shown below

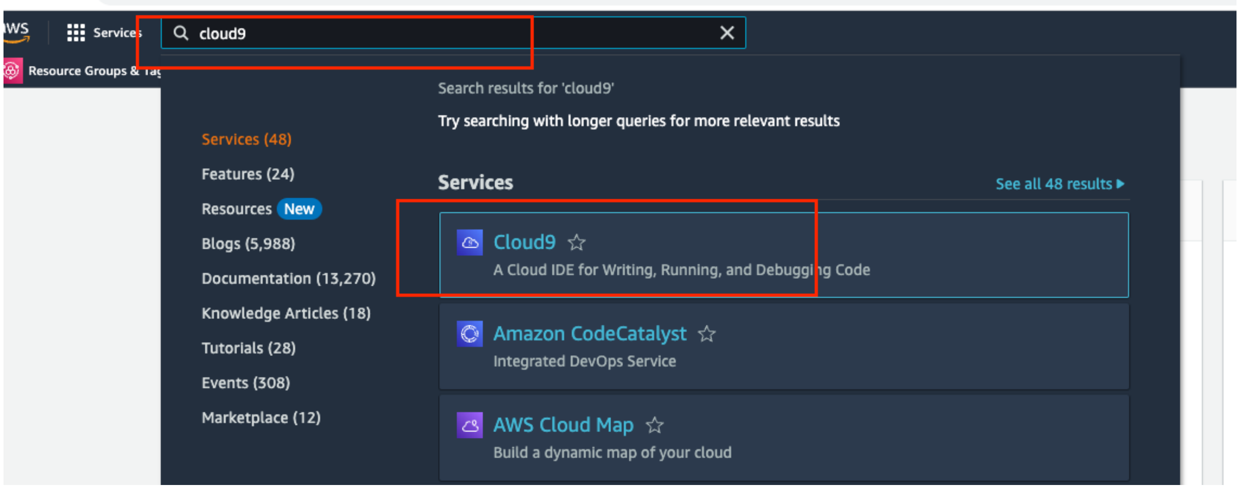


Learn the fundamentals and find valuable information to get the most out of AWS.

Learn from AWS experts and advance your skills and knowledge.

Discover new AWS services, features, and Regions.

Create a Cloud 9 Environment which will help on executing the AWS CLI commands and running big jobs (if needed) without disturbance

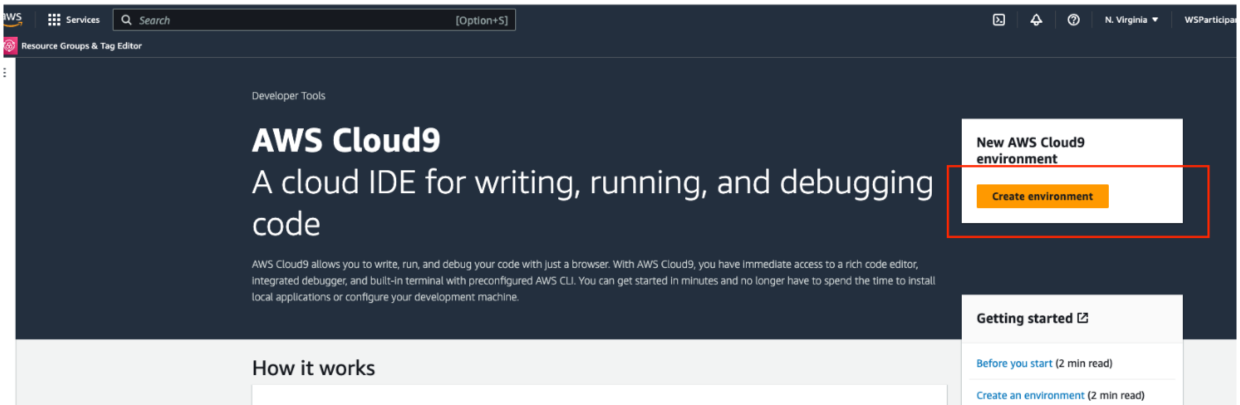


Search results for 'cloud9'

Try searching with longer queries for more relevant results

A Cloud IDE for Writing, Running, and Debugging Code

Build a dynamic map of your cloud



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

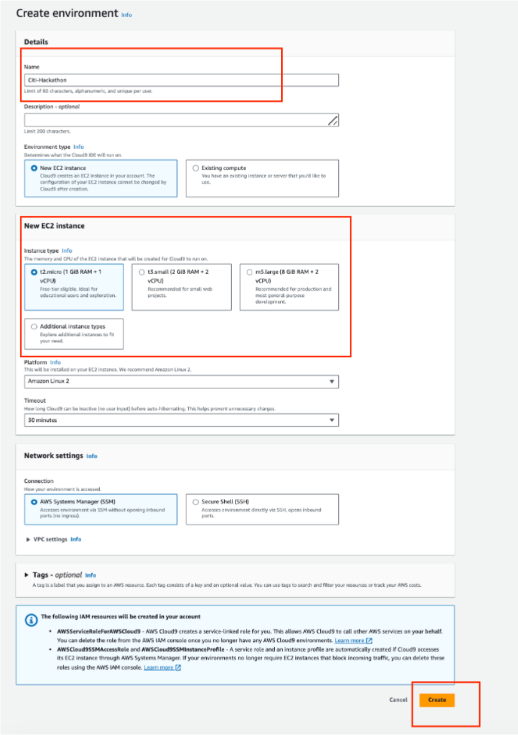
Getting started

Before you start (2 min read)

Create an environment (2 min read)

How it works

Give it a name and change the EC2 Instance size to be t3.small - Then Create the environment without changing any other configurations.



Create environment Info

Details

Name

Citi-Hackathon

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

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 need.

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 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

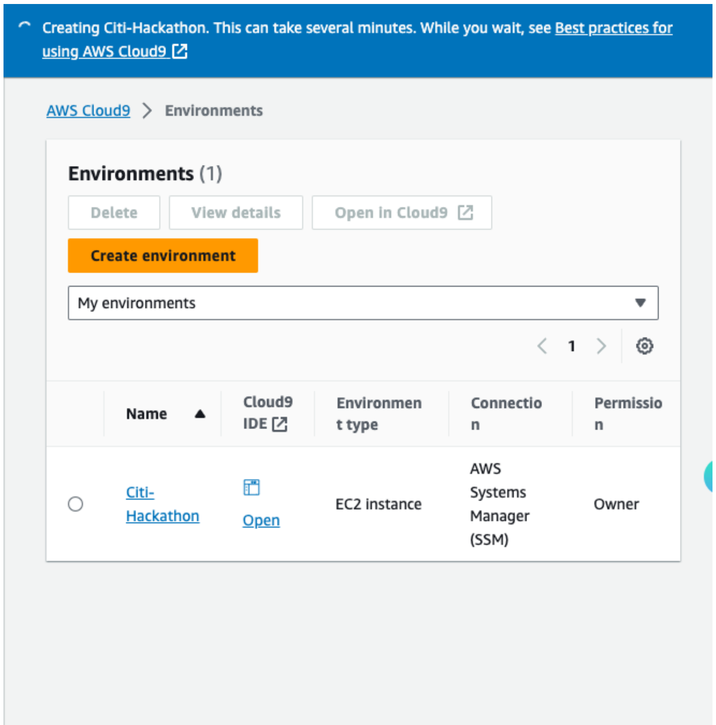
▸ **Tags** - *optional* Info

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. Learn 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. Learn more

This can take couple of minutes to create your EC2 Instance for Cloud9



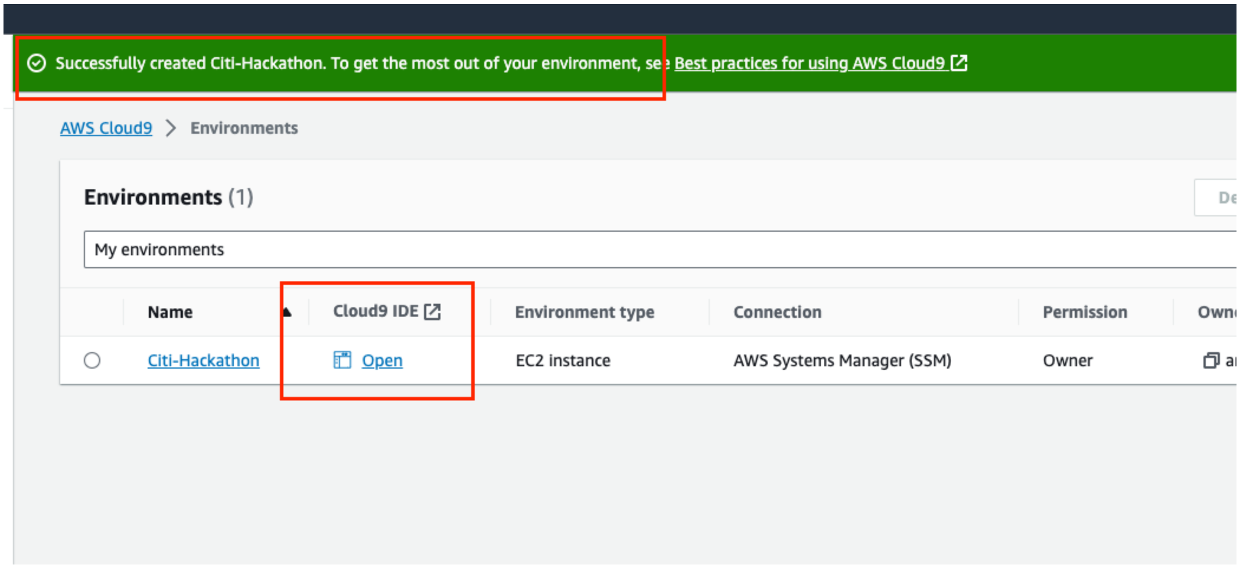
Creating Citi-Hackathon. This can take several minutes. While you wait, see Best practices for using AWS Cloud9

AWS Cloud9 > Environments

Environments (1)

	Name ▲	Cloud9 IDE	Environment type	Connection	Permission
○	Citi-Hackathon	Open	EC2 instance	AWS Systems Manager (SSM)	Owner

Once creation completed Successfully, Click Open :



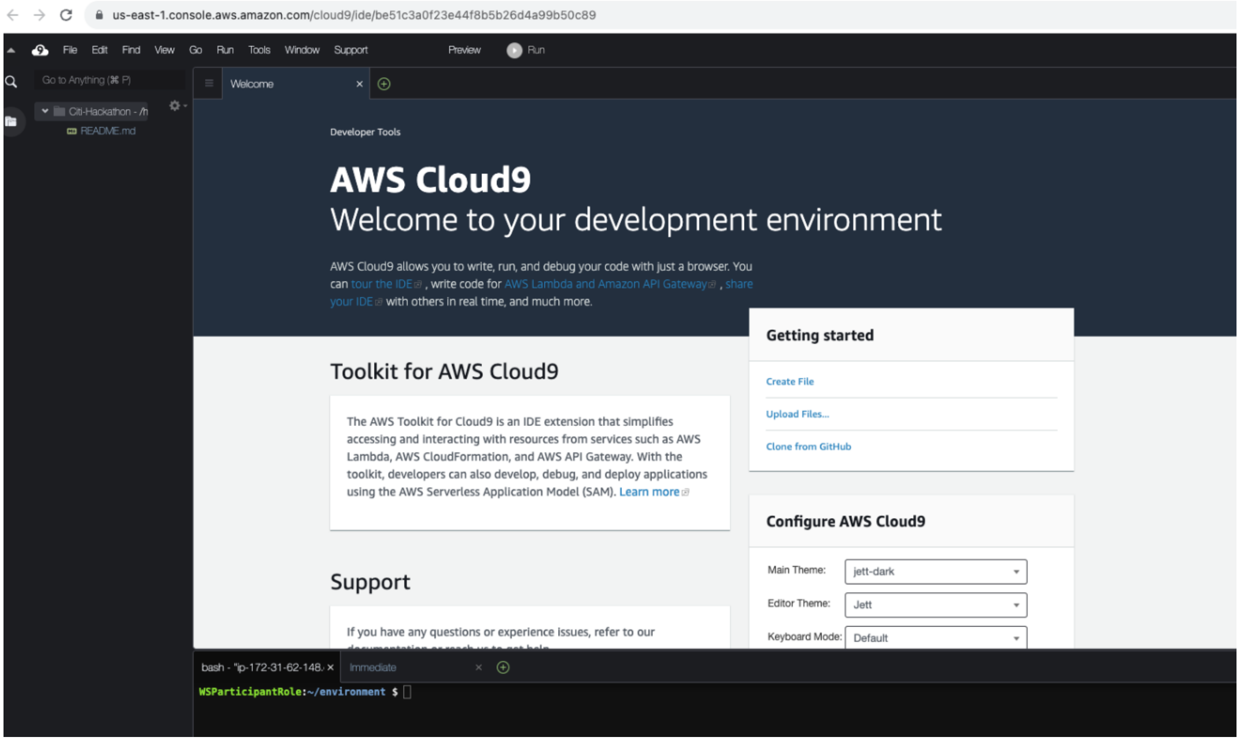
Successfully created Citi-Hackathon. To get the most out of your environment, see Best practices for using AWS Cloud9

AWS Cloud9 > Environments

Environments (1)

	Name ▲	Cloud9 IDE	Environment type	Connection	Permission	Own
○	Citi-Hackathon	Open	EC2 instance	AWS Systems Manager (SSM)	Owner	ar

This is the Terminal of your Cloud9 Environment (Ec2 Instance)


us-east-1.console.aws.amazon.com/cloud9/ide/be51c3a0f23e44f8b5b26d4a99b50c89
File Edit Find View Go Run Tools Window Support Preview Run
Go to Anything (⌘ P)
Citi-Hackathon - /h
README.md
Welcome
bash - "ip-172-31-62-148.
Immediate

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 using the AWS Serverless Application Model (SAM). Learn more

Support

If you have any questions or experience issues, refer to our

Getting started

Create File

Upload Files...

Clone from GitHub

Configure AWS Cloud9

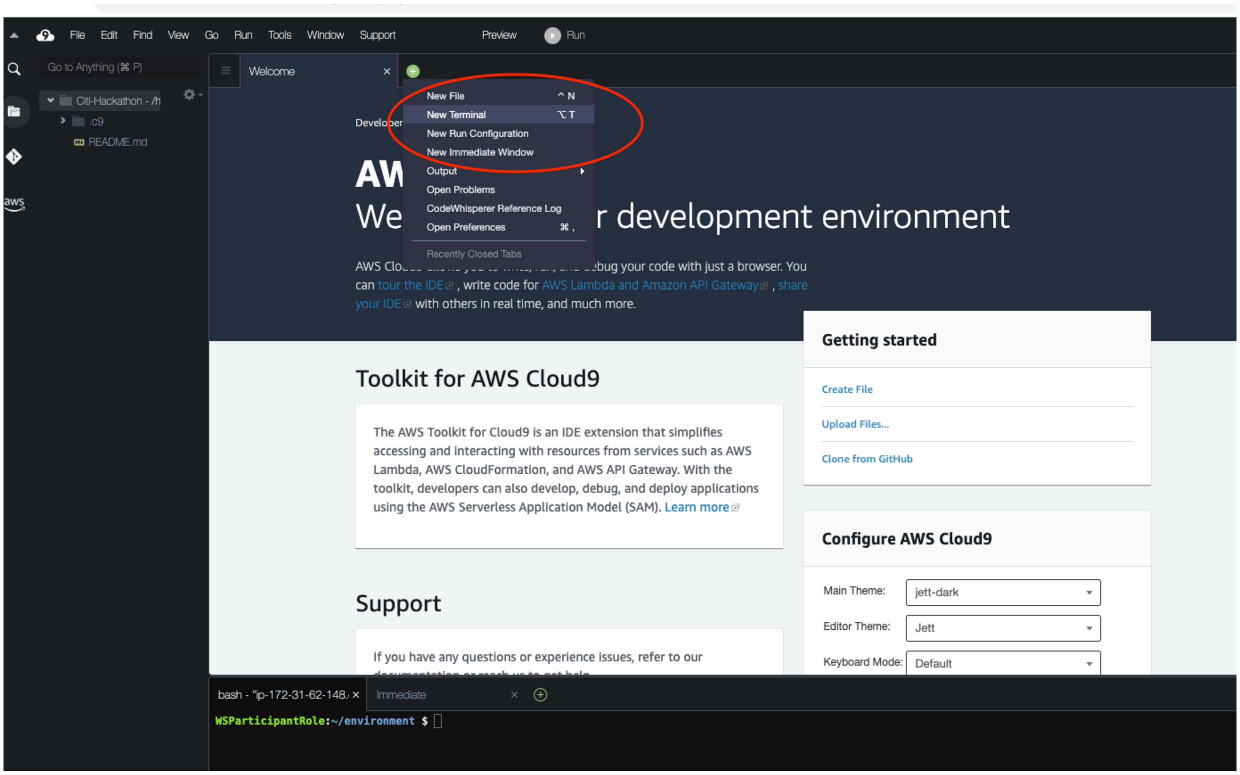
Main Theme: jett-dark

Editor Theme: Jett

Keyboard Mode: Default

```
WSParticipantRole:~/environment $ 
```

You can open a new terminal using the green “+” sign as shown below:



Develo

AW

We r development environment

AWS Clo bug 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 using the AWS Serverless Application Model (SAM). Learn more

Support

If you have any questions or experience issues, refer to our

Getting started

Create File

Upload Files...

Clone from GitHub

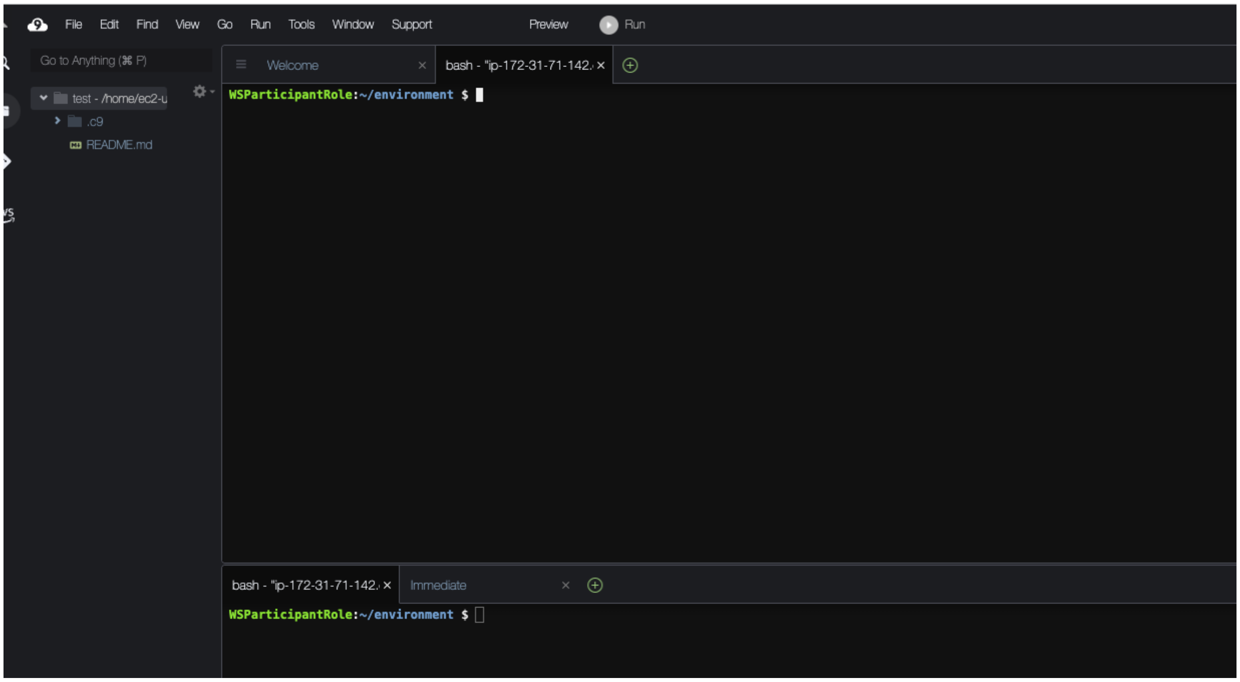
Configure AWS Cloud9

Main Theme: jett-dark

Editor Theme: Jett

Keyboard Mode: Default

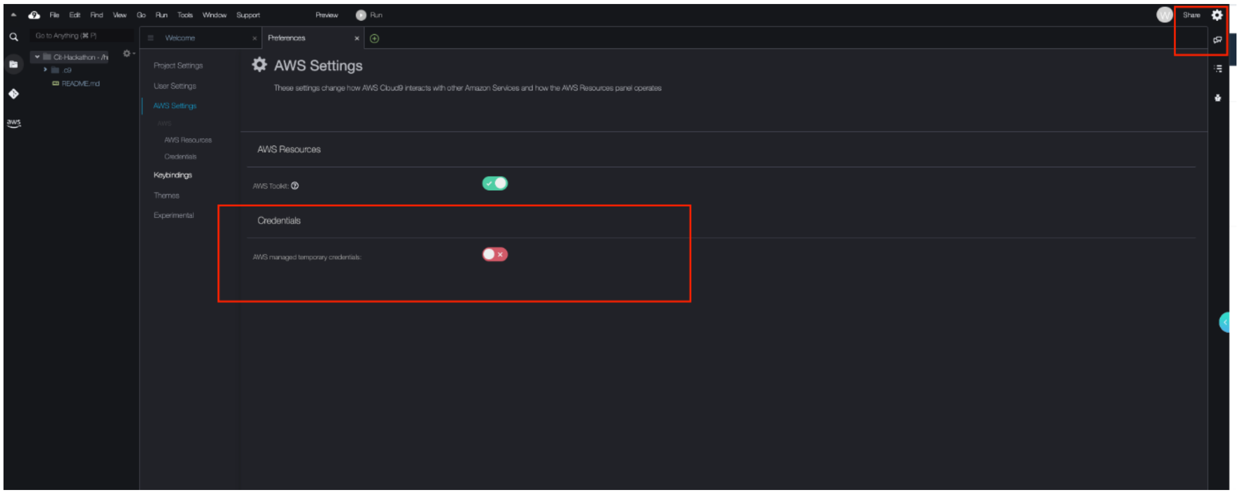
```
WSParticipantRole:~/environment $ 
```



```
WSParticipantRole:~/environment $
```

```
WSParticipantRole:~/environment $
```

you need to disable the AWS Temporary credential from the Cloud9 machine from the machine Setting , you can find setting button on the up right corner then make sure AWS managed temporary credential is disabled as shown below:



AWS Settings

These settings change how AWS Cloud9 interacts with other Amazon Services and how the AWS Resources panel operates

AWS Resources

AWS Toolkit: ✓

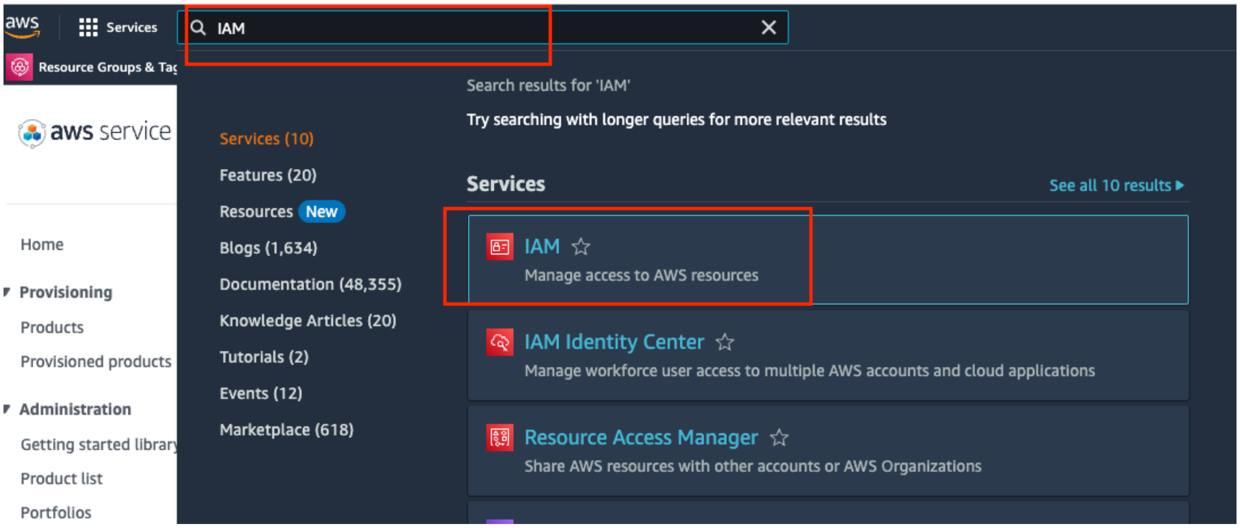
Credentials

AWS managed temporary credentials: ✗

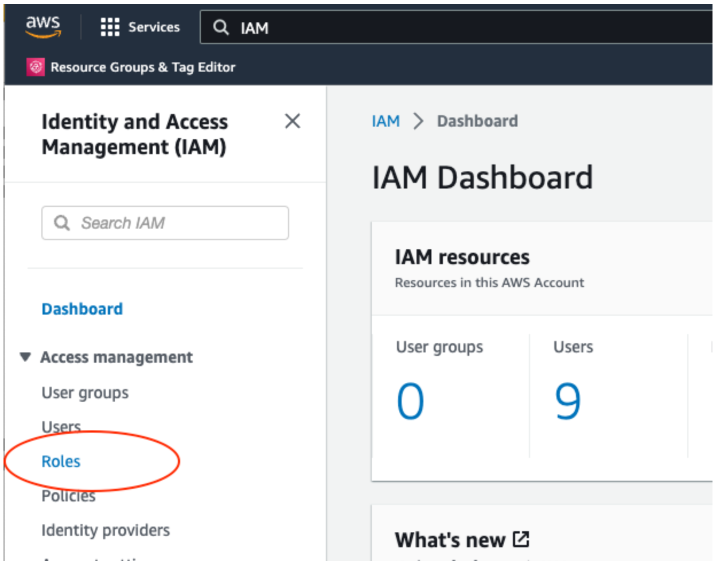
Now, We need to assign a proper IAM Permission to Cloud9 EC2 instance to be able to execute AWS CLI commands on other AWS Services like DynamoDB and S3

First you need to create Admin Role to attach it to Cloud9 :
Please Note: for the Demo purposes we choose the Full Access permissions for Cloud9 Ec2 instance but best practice here is to always give the least privilege permissions .

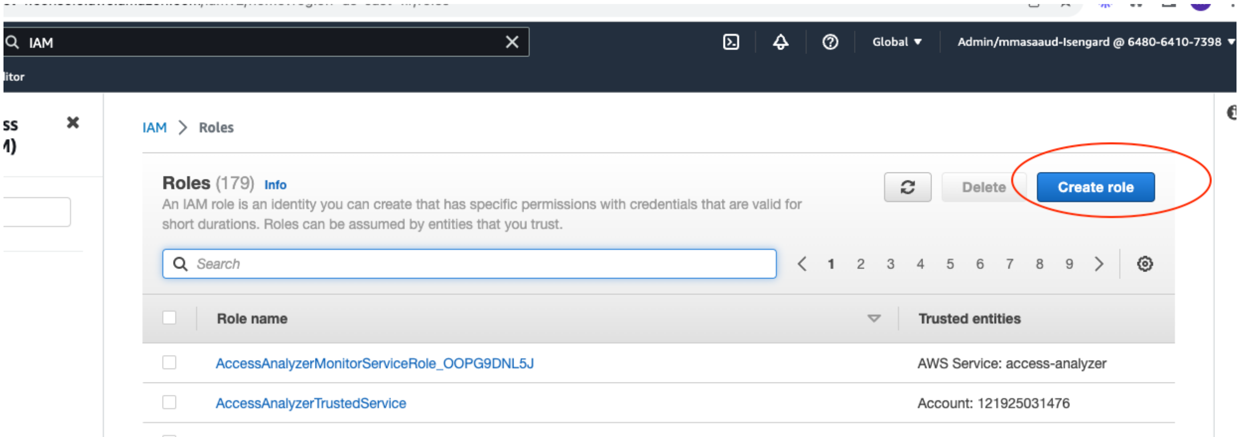
Go back to AWS console and Search for IAM on the Service search bar as below:



Go to Roles on the left side:



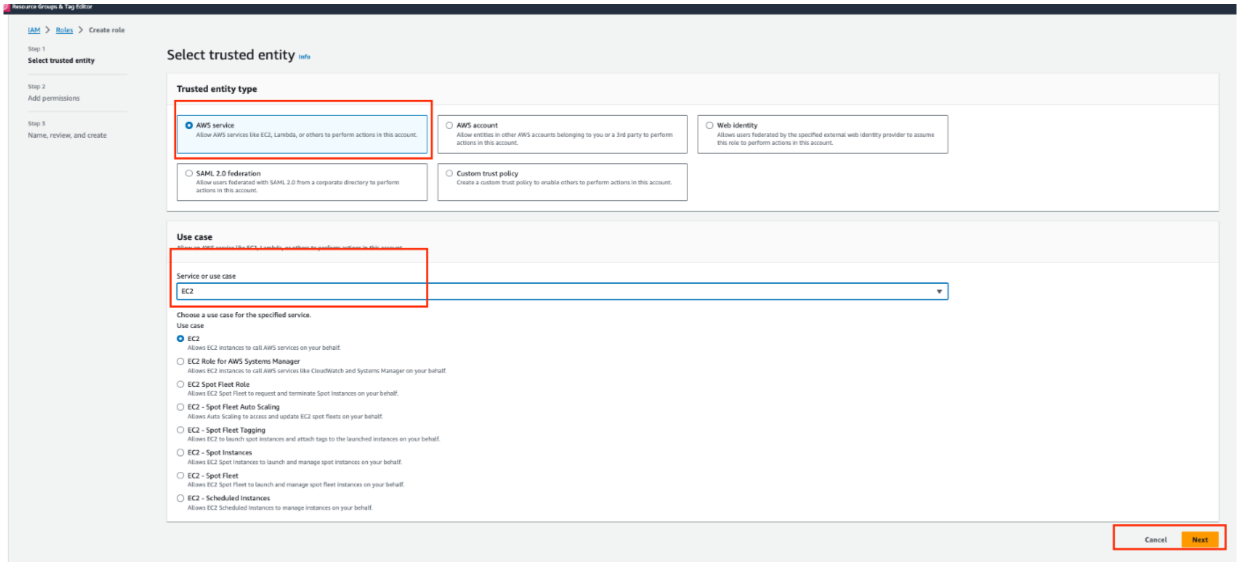
Then create a new role which will be attached to EC2 Instance:



Roles (179) 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.

□	Role name	Trusted entities
□	AccessAnalyzerMonitorServiceRole_OOPG9DNL5J	AWS Service: access-analyzer
□	AccessAnalyzerTrustedService	Account: 121925031476



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.

Service or use case: EC2

Choose a use case for the specified service.

Use case

● EC2
Allows EC2 instances to call AWS services on your behalf.

○ EC2 Role for AWS Systems Manager
Allows EC2 instances to call AWS services like CloudWatch and Systems Manager on your behalf.

○ EC2 Spot Fleet Role
Allows EC2 Spot Fleet to request and terminate Spot Instances on your behalf.

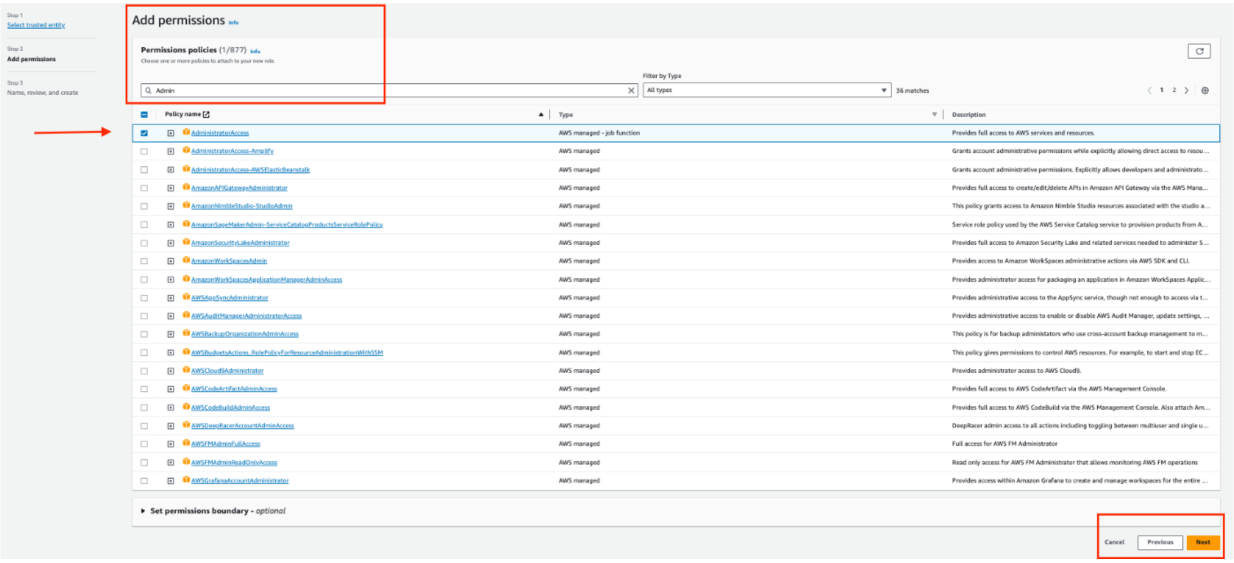
○ EC2 - Spot Fleet Auto Scaling
Allows Auto Scaling to access and update EC2 spot fleets on your behalf.

○ EC2 - Spot Fleet Tagging
Allows EC2 to launch spot instances and attach tags to the launched instances on your behalf.

○ EC2 - Spot Instances
Allows EC2 Spot Instances to launch and manage spot instances on your behalf.

○ EC2 - Spot Fleet
Allows EC2 Spot Fleet to launch and manage spot fleet instances on your behalf.

○ EC2 - Scheduled Instances
Allows EC2 Scheduled Instances to manage instances on your behalf.



Add permissions Info

Permissions policies (1/877) Info

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

Admin

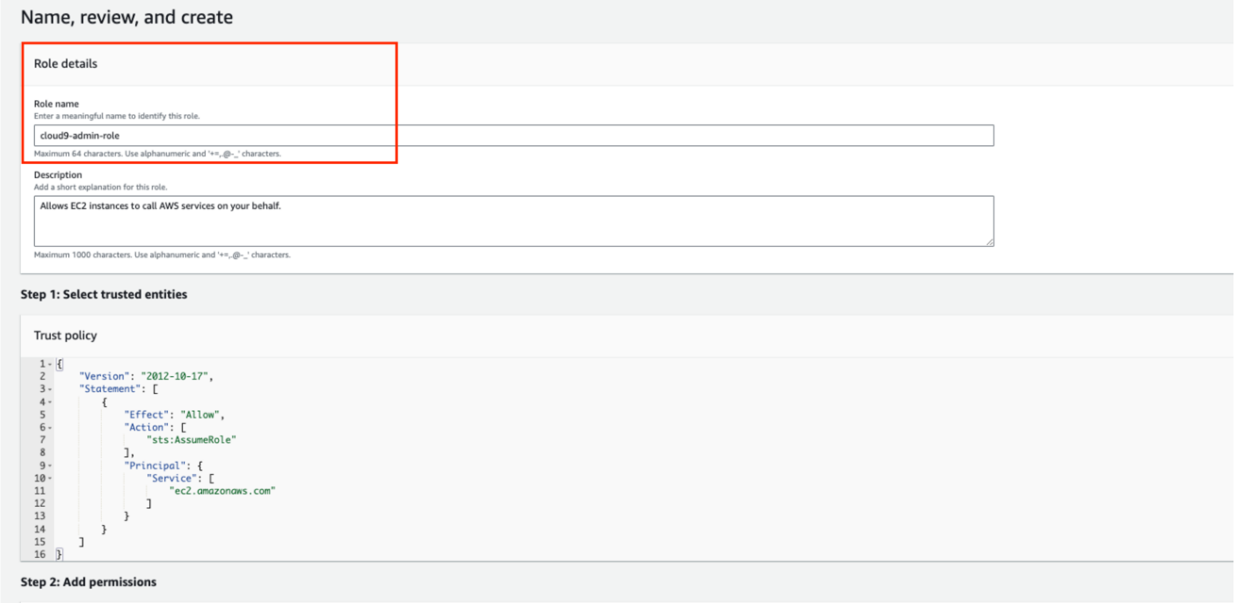
Filter by Type: All types

36 matches

	Policy name	Type	Description
☑	AdministratorAccess	AWS managed - job function	Provides full access to AWS services and resources.
□	AdministratorAccess-Amplify	AWS managed	Grants account administrative permissions while explicitly allowing direct access to resou...
□	AdministratorAccess-AWSElasticBeanstalk	AWS managed	Grants account administrative permissions. Explicitly allows developers and administrato...
□	AmazonAPIGatewayAdministrator	AWS managed	Provides full access to create/edit/delete APIs in Amazon API Gateway via the AWS Mana...
□	AmazonNimbleStudio-StudioAdmin	AWS managed	This policy grants access to Amazon Nimble Studio resources associated with the studio a...
□	AmazonSageMakerAdmin-ServiceCatalogProductsServiceRolePolicy	AWS managed	Service role policy used by the AWS Service Catalog service to provision products from A...
□	AmazonSecurityLakeAdministrator	AWS managed	Provides full access to Amazon Security Lake and related services needed to administer S...
□	AmazonWorkSpacesAdmin	AWS managed	Provides access to Amazon WorkSpaces administrative actions via AWS SDK and CLI.
□	AmazonWorkSpacesApplicationManagerAdminAccess	AWS managed	Provides administrator access for packaging an application in Amazon WorkSpaces Applic...
□	AWSAppSyncAdministrator	AWS managed	Provides administrative access to the AppSync service, though not enough to access via t...
□	AWSAuditManagerAdministratorAccess	AWS managed	Provides administrative access to enable or disable AWS Audit Manager, update settings, ...
□	AWSBackupOrganizationAdminAccess	AWS managed	This policy is for backup administators who use cross-account backup management to m...
□	AWSBudgetsActions_RolePolicyForResourceAdministrationWithSSM	AWS managed	This policy gives permissions to control AWS resources. For example, to start and stop EC...
□	AWSCloud9Administrator	AWS managed	Provides administrator access to AWS Cloud9.
□	AWSCodeArtifactAdminAccess	AWS managed	Provides full access to AWS CodeArtifact via the AWS Management Console.
□	AWSCodeBuildAdminAccess	AWS managed	Provides full access to AWS CodeBuild via the AWS Management Console. Also attach Am...
□	AWSDeepRacerAccountAdminAccess	AWS managed	DeepRacer admin access to all actions including toggling between multiuser and single u...
□	AWSFMAdminFullAccess	AWS managed	Full access for AWS FM Administrator
□	AWSFMAdminReadOnlyAccess	AWS managed	Read only access for AWS FM Administrator that allows monitoring AWS FM operations
□	AWSGrafanaAccountAdministrator	AWS managed	Provides access within Amazon Grafana to create and manage workspaces for the entire ...

▸ **Set permissions boundary** - *optional*

give a Role Name (Please keep a record of that Role name as we need while assigning it to the Ec2 Instance for Cloud9 :



Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.
cloud9-admin-role
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.

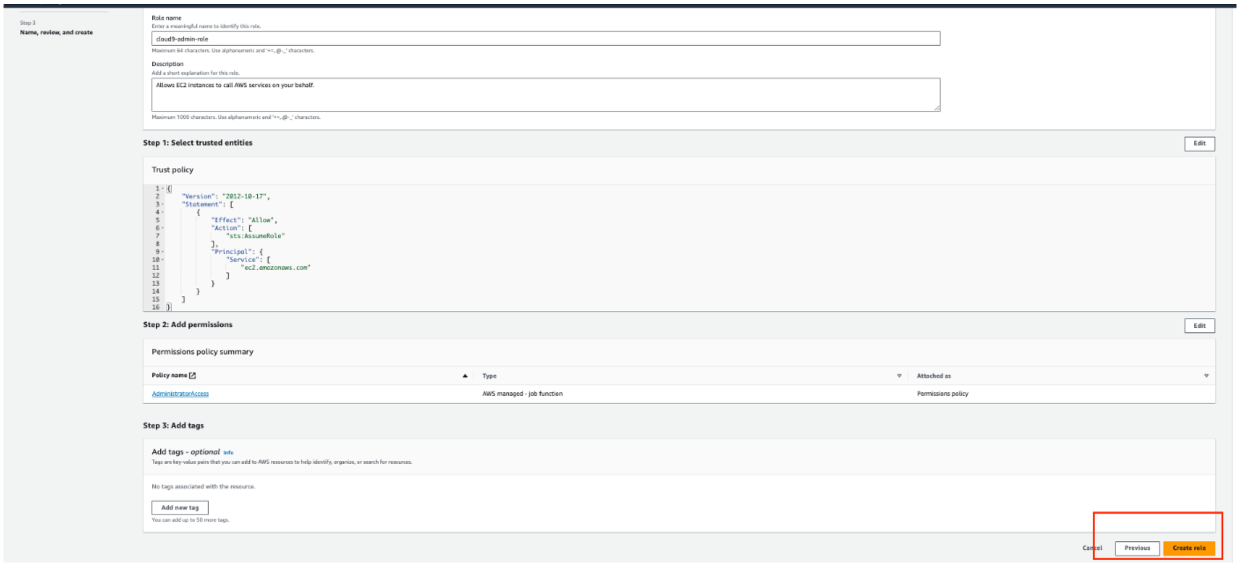
Description
Add a short explanation for this role.
Allows EC2 instances to call AWS services on your behalf.
Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

Step 1: Select trusted entities

Trust policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sts:AssumeRole"
            ],
            "Principal": {
                "Service": [
                    "ec2.amazonaws.com"
                ]
            }
        }
    ]
}
```

Step 2: Add permissions



Step 3

Name, review, and create

Role name

Enter a meaningful name to identify this role.

cloud9-admin-role

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

Description

Add a short explanation for this role.

Allows EC2 instances to call AWS services on your behalf.

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

Step 1: Select trusted entities

Trust policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sts:AssumeRole"
            ],
            "Principal": {
                "Service": [
                    "ec2.amazonaws.com"
                ]
            }
        }
    ]
}
```

Step 2: Add permissions

Permissions policy summary

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

Step 3: Add tags

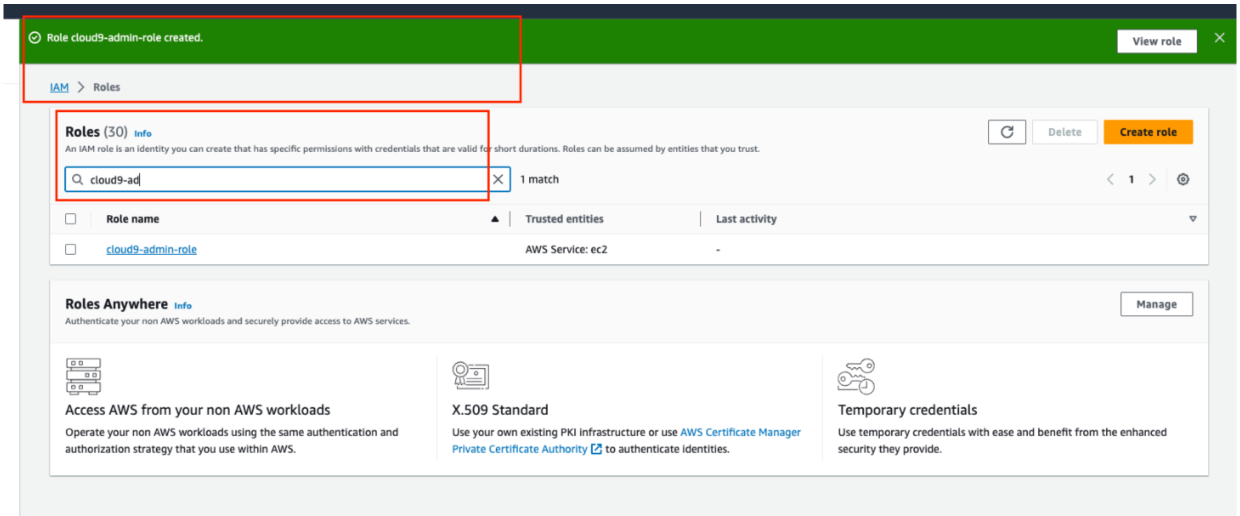
Add tags - *optional* Info

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.

You can add up to 50 more tags.

To validate role creation please type in role name created on Search and make sure it got created Successfully



Role cloud9-admin-role 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.

cloud9-ad

1 match

☐	Role name	Trusted entities	Last activity
☐	cloud9-admin-role	AWS Service: ec2	-

Roles Anywhere Info

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

Access AWS from your non AWS workloads

Operate your non AWS workloads using the same authentication and authorization strategy that you use within AWS.

X.509 Standard

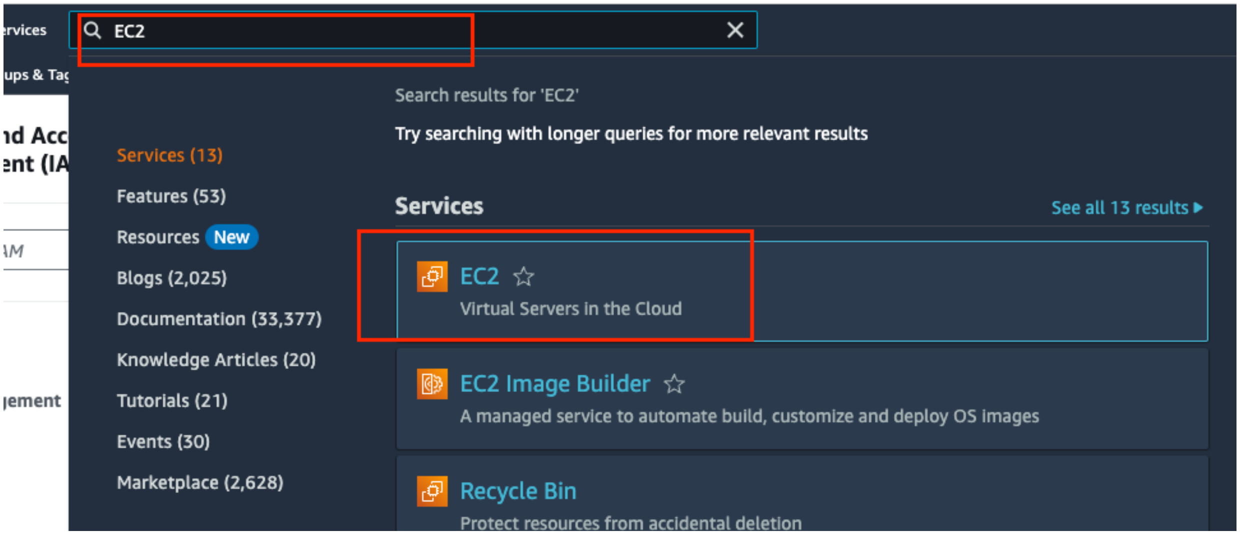
Use your own existing PKI infrastructure or use AWS Certificate Manager Private Certificate Authority to authenticate identities.

Temporary credentials

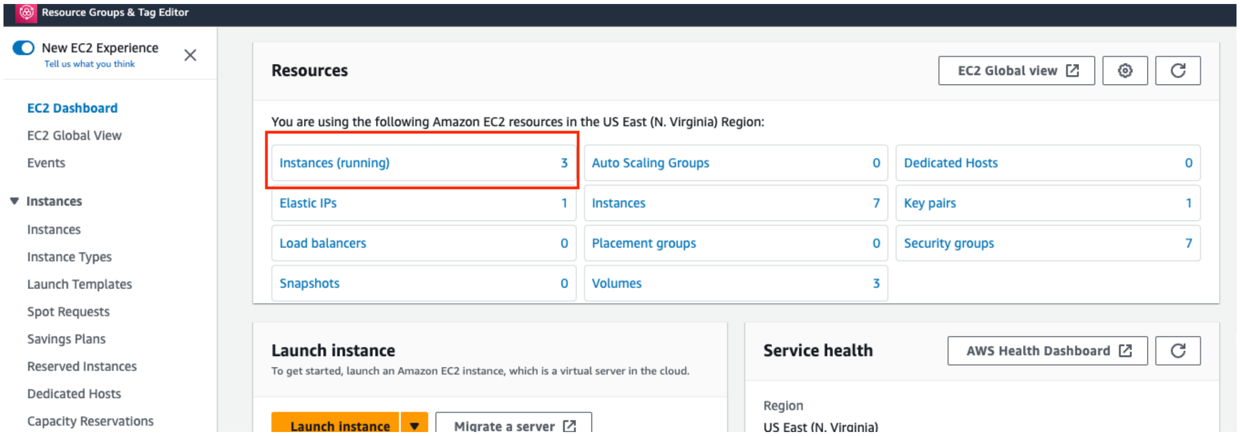
Use temporary credentials with ease and benefit from the enhanced security they provide.

After creating the IAM Role you will need to attach it to the EC2 instance to enable Cloud9 to execute any AWS command needed based on the permissions you gave on that Role:

back to AWS Console and Look for EC2 as shown below :



Click on the “ Instance Running”

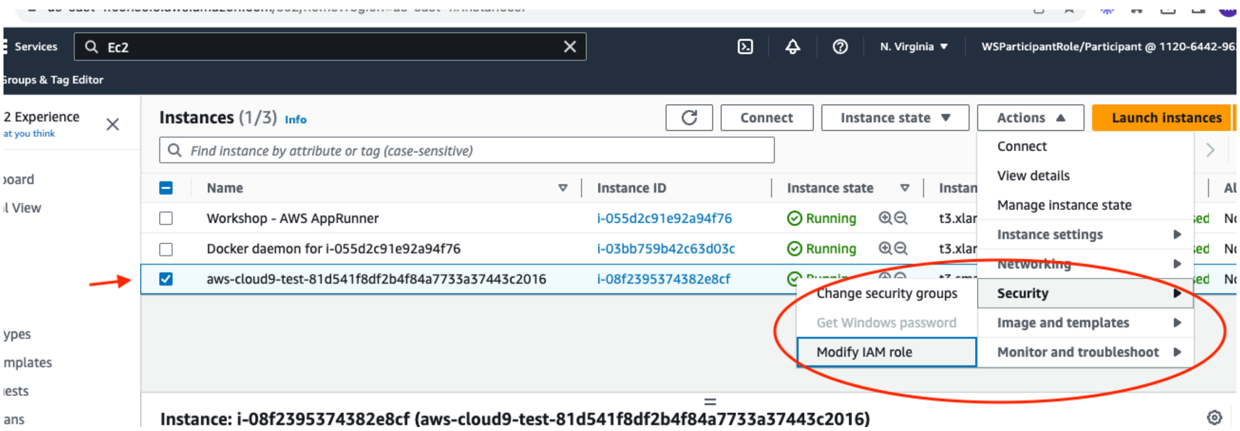


You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Instances (running)	3	Auto Scaling Groups	0	Dedicated Hosts	0
Elastic IPs	1	Instances	7	Key pairs	1
Load balancers	0	Placement groups	0	Security groups	7
Snapshots	0	Volumes	3		

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

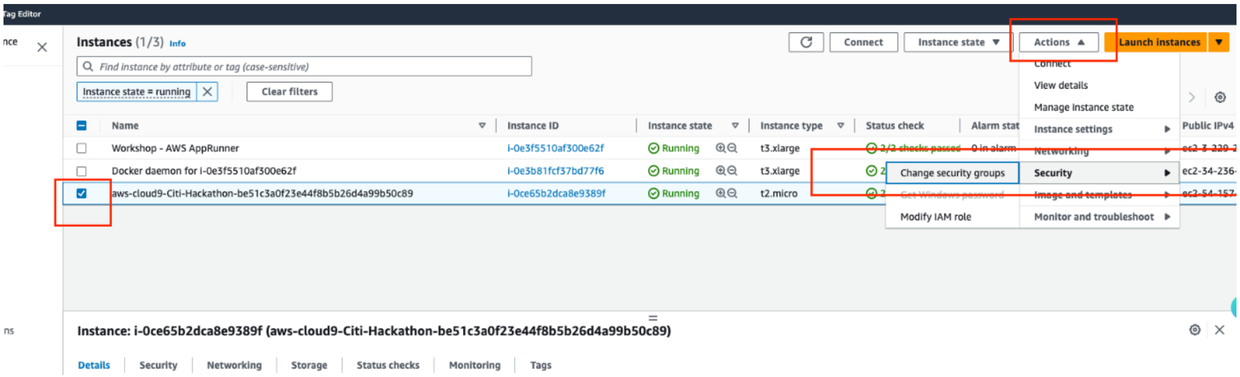
Region
US East (N. Virginia)



	Name	Instance ID	Instance state	Instan
□	Workshop - AWS AppRunner	i-055d2c91e92a94f76	Running	t3.xlar
□	Docker daemon for i-055d2c91e92a94f76	i-03bb759b42c63d03c	Running	t3.xlar
☑	aws-cloud9-test-81d541f8df2b4f84a7733a37443c2016	i-08f2395374382e8cf		

Instance: i-08f2395374382e8cf (aws-cloud9-test-81d541f8df2b4f84a7733a37443c2016)

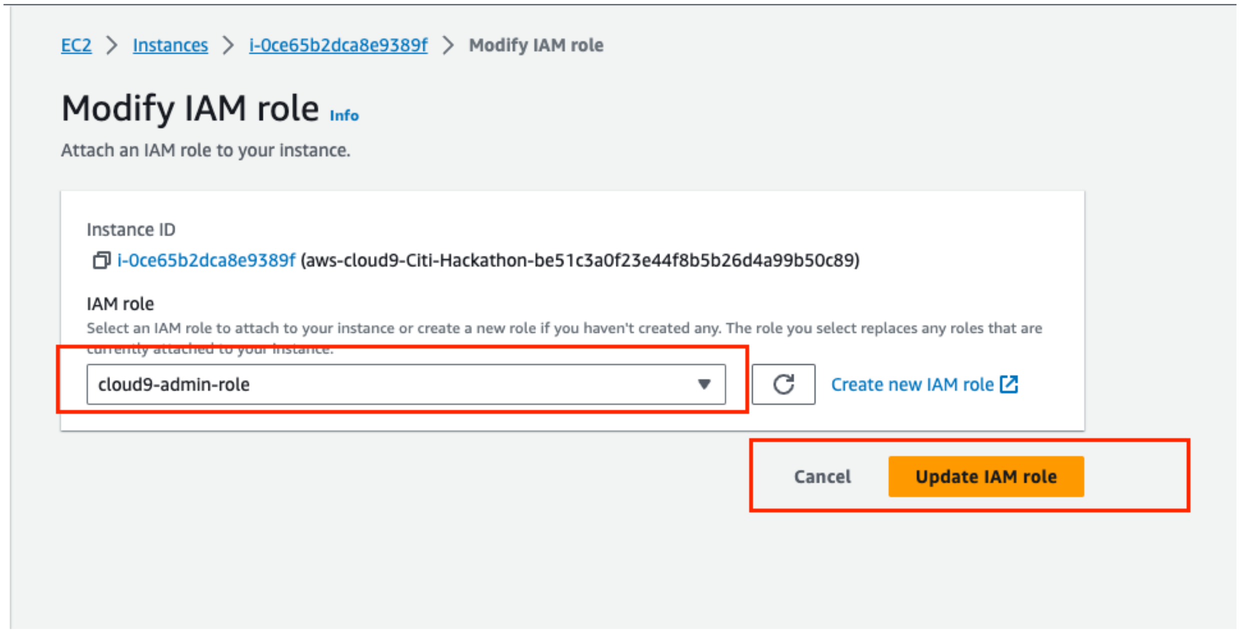
check the box for the EC2 instance related to Cloud9 then go to Security > Modify IAM Policy :



	Name	Instance ID	Instance state	Instance type	Status check	Alarm stat	Public IPv4
☐	Workshop - AWS AppRunner	i-0e3f5510af300e62f	Running	t3.xlarge	2/2 checks passed	0 in alarm	ec2-3-229-2
☐	Docker daemon for i-0e3f5510af300e62f	i-0e3b81fcf37bd77f6	Running	t3.xlarge	2		ec2-34-236-
☑	aws-cloud9-Citi-Hackathon-be51c3a0f23e44f8b5b26d4a99b50c89	i-0ce65b2dca8e9389f	Running	t2.micro	2		ec2-54-157-

Instance: i-0ce65b2dca8e9389f (aws-cloud9-Citi-Hackathon-be51c3a0f23e44f8b5b26d4a99b50c89)

Select the newly created admin role and click Update IAM Role them Click on Update IAM Role :



Modify IAM role

Attach an IAM role to your instance.

Instance ID

i-0ce65b2dca8e9389f (aws-cloud9-Citi-Hackathon-be51c3a0f23e44f8b5b26d4a99b50c89)

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.

cloud9-admin-role

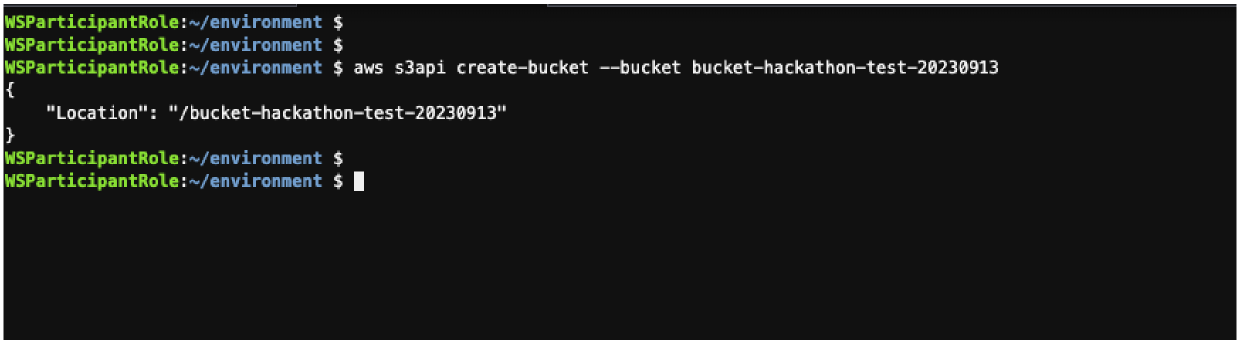
After assigning the Proper Permissions to the Cloud9 machine, you can now you can execute AWS commands from Cloud9 environment as shown below :

1- Create S3 Bucket

give example from random number to actual number , open up word or notepad and past that there to copy it on AWS CLI on Cloud9

S3 bucket name must be Globally unique, so please make sure to add random number or date to make it unique:

```
aws s3api create-bucket –bucket bucket-hackathon-test-<random number>
```



```
WSParticipantRole:~/environment $
WSParticipantRole:~/environment $
WSParticipantRole:~/environment $ aws s3api create-bucket --bucket bucket-hackathon-test-20230913
{
    "Location": "/bucket-hackathon-test-20230913"
}
WSParticipantRole:~/environment $
WSParticipantRole:~/environment $
```

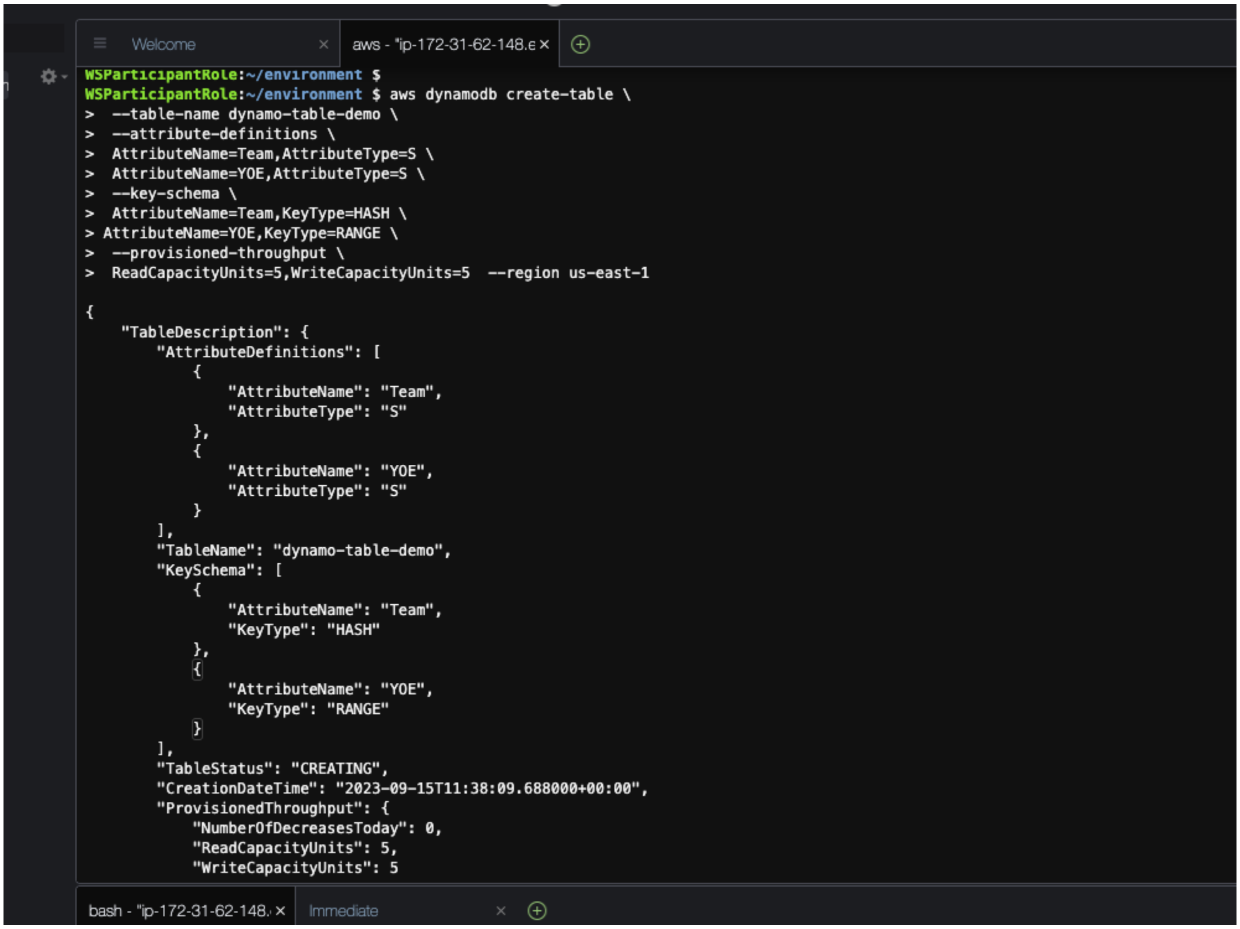
2- Create dynamoDB table

For my Sample Code I’m using DynamoDB for my no-SQL Database, here is the command to create the table :

Note:

Please make sure to use the right region name while excusing that command below, for example I’m using us-east-1 so the command should be like below :

```
aws dynamodb create-table \
 --table-name dynamo-table-demo \
 --attribute-definitions \
AttributeName=Team,AttributeType=S \
AttributeName=YOE,AttributeType=S \
 --key-schema \
AttributeName=Team,KeyType=HASH \
AttributeName=YOE,KeyType=RANGE \
 --provisioned-throughput \
 ReadCapacityUnits=5,WriteCapacityUnits=5  --region us-east-1
```



```
WSParticipantRole:~/environment $
WSParticipantRole:~/environment $ aws dynamodb create-table \
>   --table-name dynamo-table-demo \
>   --attribute-definitions \
>   AttributeName=Team,AttributeType=S \
>   AttributeName=YOE,AttributeType=S \
>   --key-schema \
>   AttributeName=Team,KeyType=HASH \
> AttributeName=YOE,KeyType=RANGE \
>   --provisioned-throughput \
>   ReadCapacityUnits=5,WriteCapacityUnits=5  --region us-east-1

{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "Team",
                "AttributeType": "S"
            },
            {
                "AttributeName": "YOE",
                "AttributeType": "S"
            }
        ],
        "TableName": "dynamo-table-demo",
        "KeySchema": [
            {
                "AttributeName": "Team",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "YOE",
                "KeyType": "RANGE"
            }
        ],
        "TableStatus": "CREATING",
        "CreationDateTime": "2023-09-15T11:38:09.688000+00:00",
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 5,
            "WriteCapacityUnits": 5
```

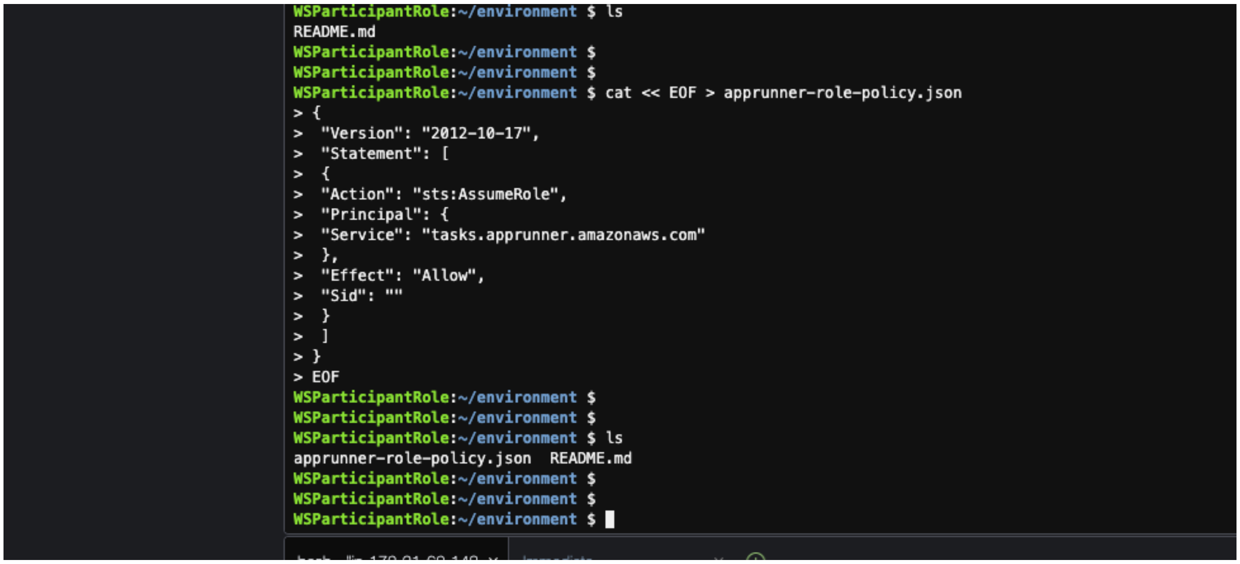
3- Create IAM Role for app Runner:

Please Note: for the Demo purposes we choose the Full Access permissions to DynamoDB , best practices is to give the least privilege permissions for AppRunner which enable it to execute the code on the targeted services.

Finally, We need to assign IAM Role for Apprunner Service to enable the service communicate with S3 and DynamoDB while running the code, please follow the steps below so Apprunner can communicate with your Github account and with other AWS Service :

```
cat << EOF > apprunner-role-policy.json
{
"Version": "2012-10-17",
"Statement": [
{
"Action": "sts:AssumeRole",
"Principal": {
"Service": "tasks.apprunner.amazonaws.com"
},
"Effect": "Allow",
"Sid": ""
}
]
}
```

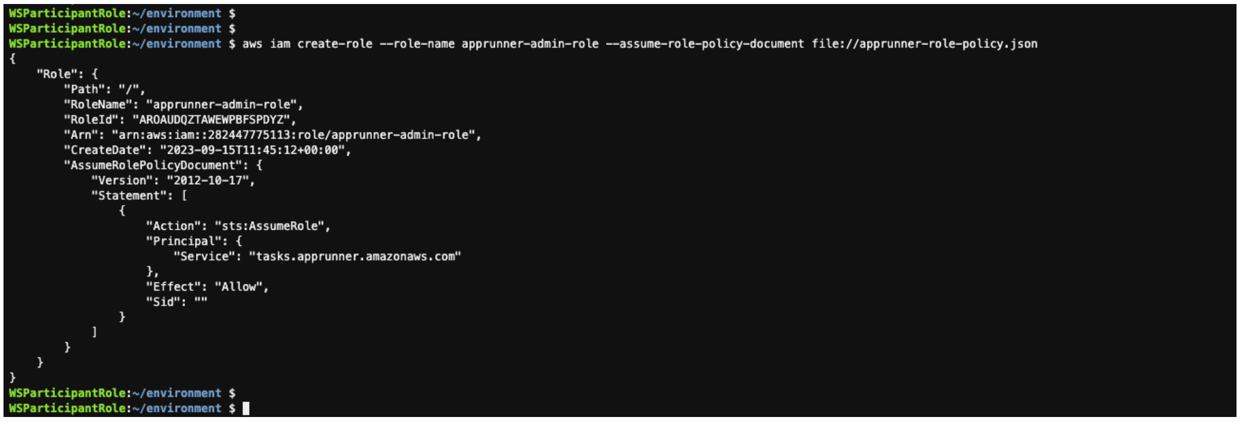
```
EOF
```



```
WSParticipantRole:~/environment $ ls
README.md
WSParticipantRole:~/environment $
WSParticipantRole:~/environment $
WSParticipantRole:~/environment $ cat << EOF > apprunner-role-policy.json
> {
>  "Version": "2012-10-17",
>  "Statement": [
>  {
>  "Action": "sts:AssumeRole",
>  "Principal": {
>  "Service": "tasks.apprunner.amazonaws.com"
>  },
>  "Effect": "Allow",
>  "Sid": ""
>  }
>  ]
> }
> EOF
WSParticipantRole:~/environment $
WSParticipantRole:~/environment $
WSParticipantRole:~/environment $ ls
apprunner-role-policy.json  README.md
WSParticipantRole:~/environment $
WSParticipantRole:~/environment $
WSParticipantRole:~/environment $
```

Create IAM Assume Role for apprunner to enable it communicate with Github and other API as needed :

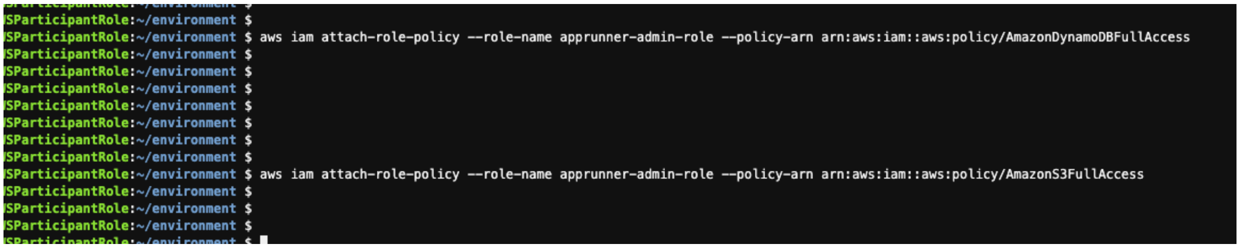
```
aws iam create-role --role-name apprunner-role --assume-role-policy-document file://ap
```



```
WSParticipantRole:~/environment $
WSParticipantRole:~/environment $
WSParticipantRole:~/environment $ aws iam create-role --role-name apprunner-admin-role --assume-role-policy-document file://apprunner-role-policy.json
{
    "Role": {
        "Path": "/",
        "RoleName": "apprunner-admin-role",
        "RoleId": "AROAUDQZTAWEWPBFSPDYZ",
        "Arn": "arn:aws:iam::282447775113:role/apprunner-admin-role",
        "CreateDate": "2023-09-15T11:45:12+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Action": "sts:AssumeRole",
                    "Principal": {
                        "Service": "tasks.apprunner.amazonaws.com"
                    },
                    "Effect": "Allow",
                    "Sid": ""
                }
            ]
        }
    }
}
WSParticipantRole:~/environment $
WSParticipantRole:~/environment $
```

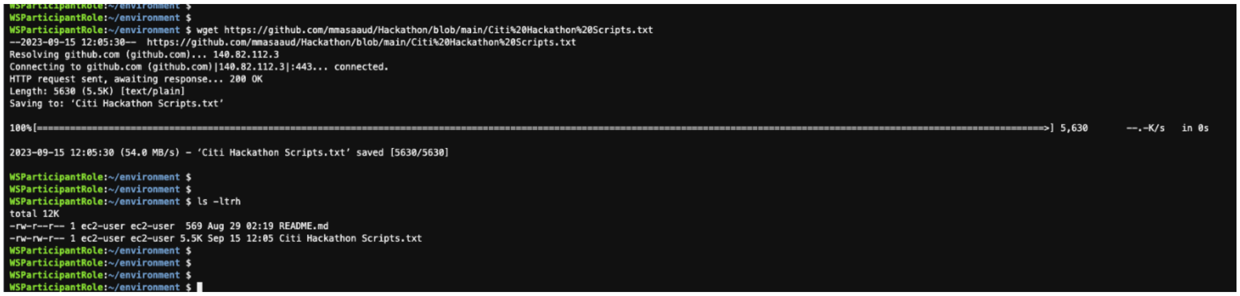
Give also permissions to APprunner to communicate with DynamoDB and S3 as We need it to run the code there

```
aws iam attach-role-policy --role-name apprunner-role --policy-arn arn:aws:iam::aws:po

aws iam attach-role-policy --role-name apprunner-role --policy-arn arn:aws:iam::aws:po
```



```
SParticipantRole:~/environment $
SParticipantRole:~/environment $ aws iam attach-role-policy --role-name apprunner-admin-role --policy-arn arn:aws:iam::aws:policy/AmazonDynamoDBFullAccess
SParticipantRole:~/environment $
SParticipantRole:~/environment $
SParticipantRole:~/environment $
SParticipantRole:~/environment $
SParticipantRole:~/environment $
SParticipantRole:~/environment $
SParticipantRole:~/environment $
SParticipantRole:~/environment $ aws iam attach-role-policy --role-name apprunner-admin-role --policy-arn arn:aws:iam::aws:policy/AmazonS3FullAccess
SParticipantRole:~/environment $
SParticipantRole:~/environment $
SParticipantRole:~/environment $
SParticipantRole:~/environment $
```

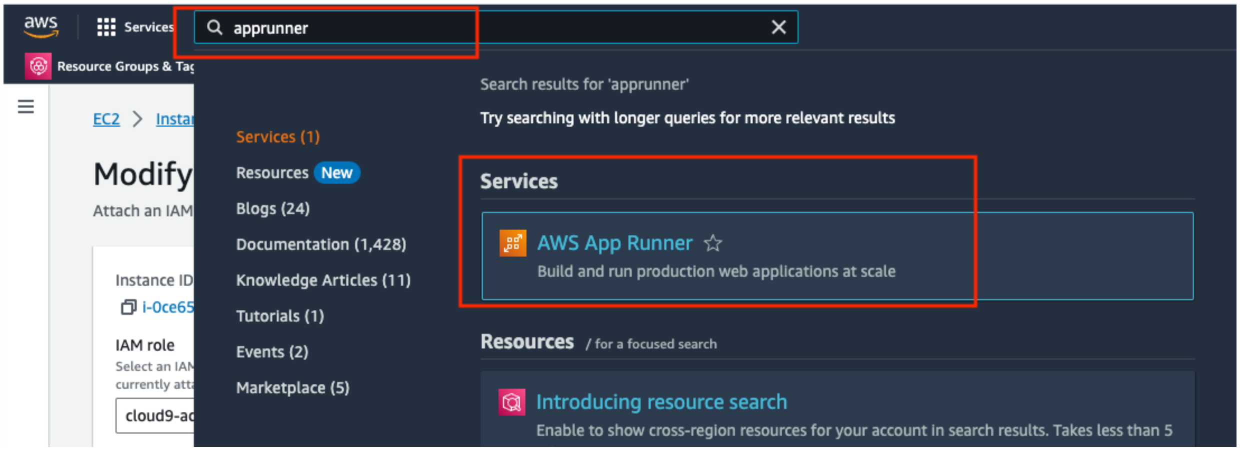
Optional Step You can also download all those commands executed previously from Github using the command below that might easier for copy/paste commands from there directly:



```
WSParticipantRole:~/environment $
WSParticipantRole:~/environment $ wget https://github.com/mmasaaud/Hackathon/blob/main/Citi%20Hackathon%20Scripts.txt
--2023-09-15 12:05:30--  https://github.com/mmasaaud/Hackathon/blob/main/Citi%20Hackathon%20Scripts.txt
Resolving github.com (github.com)... 140.82.112.3
Connecting to github.com (github.com)|140.82.112.3|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5630 (5.5K) [text/plain]
Saving to: 'Citi Hackathon Scripts.txt'

100%[=====================================================================================================================================================>] 5,630      --.-K/s   in 0s

2023-09-15 12:05:30 (54.0 MB/s) - 'Citi Hackathon Scripts.txt' saved [5630/5630]

WSParticipantRole:~/environment $
WSParticipantRole:~/environment $
WSParticipantRole:~/environment $ ls -ltrh
total 12K
-rw-r--r-- 1 ec2-user ec2-user  569 Aug 29 02:19 README.md
-rw-rw-r-- 1 ec2-user ec2-user 5.5K Sep 15 12:05 Citi Hackathon Scripts.txt
WSParticipantRole:~/environment $
WSParticipantRole:~/environment $
WSParticipantRole:~/environment $
WSParticipantRole:~/environment $
```

AppRunner Code execution :

Back to AWS Console and look for Apprunner Service on the search bar :



Search results for 'apprunner'

Try searching with longer queries for more relevant results

Services

AWS App Runner

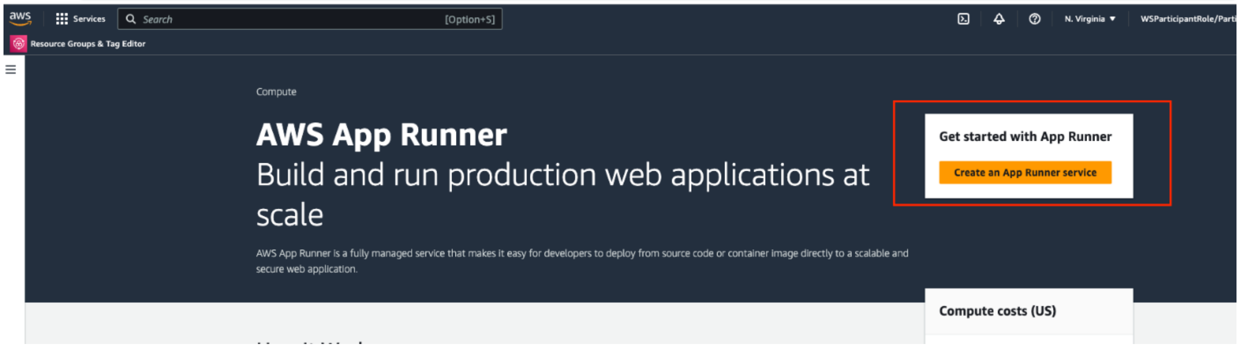
Build and run production web applications at scale

Resources / for a focused search

Introducing resource search

Enable to show cross-region resources for your account in search results. Takes less than 5

Click on create a new Service :



Compute

AWS App Runner

Build and run production web applications at scale

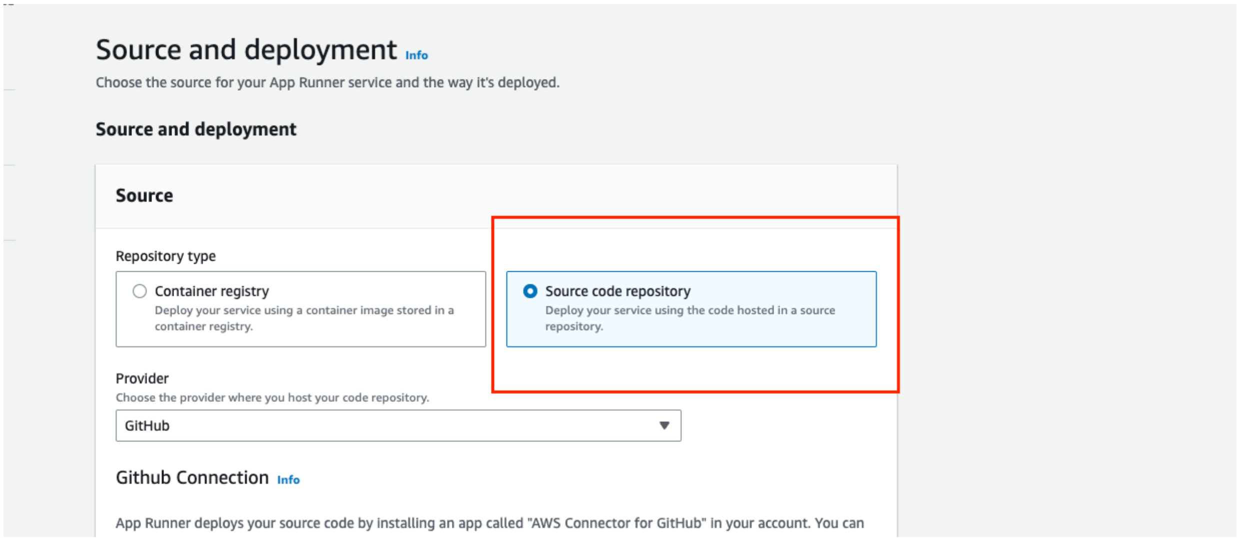
AWS App Runner is a fully managed service that makes it easy for developers to deploy from source code or container image directly to a scalable and secure web application.

Get started with App Runner

Create an App Runner service

Compute costs (US)

Choose your repository type either you can use the source code directly from your Github account or you can use it from a container image which you uploaded already to Amazon ECR, for the sample code I'm using Source Code repository directly from Github as shown below


Source and deployment Info

Choose the source for your App Runner service and the way it's deployed.

Source and deployment

Source

Repository type

○ Container registry
Deploy your service using a container image stored in a container registry.

● Source code repository
Deploy your service using the code hosted in a source repository.

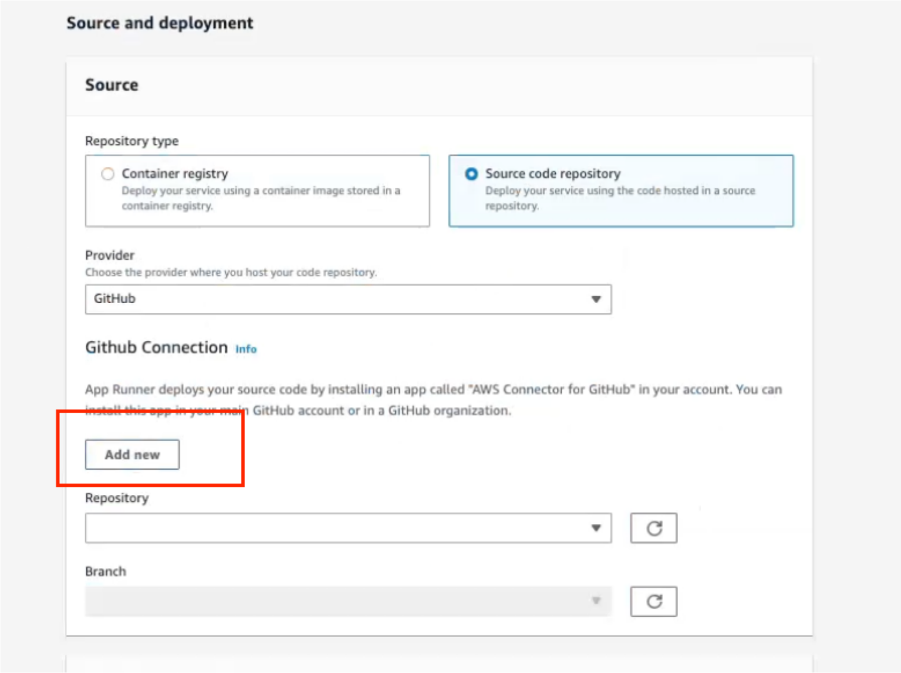
Provider
Choose the provider where you host your code repository.

GitHub

Github Connection Info

App Runner deploys your source code by installing an app called "AWS Connector for GitHub" in your account. You can

Then you need to give your Github credentials to Apprunner so it can access your repositories from Github as shown below:



Source and deployment

Source

Repository type

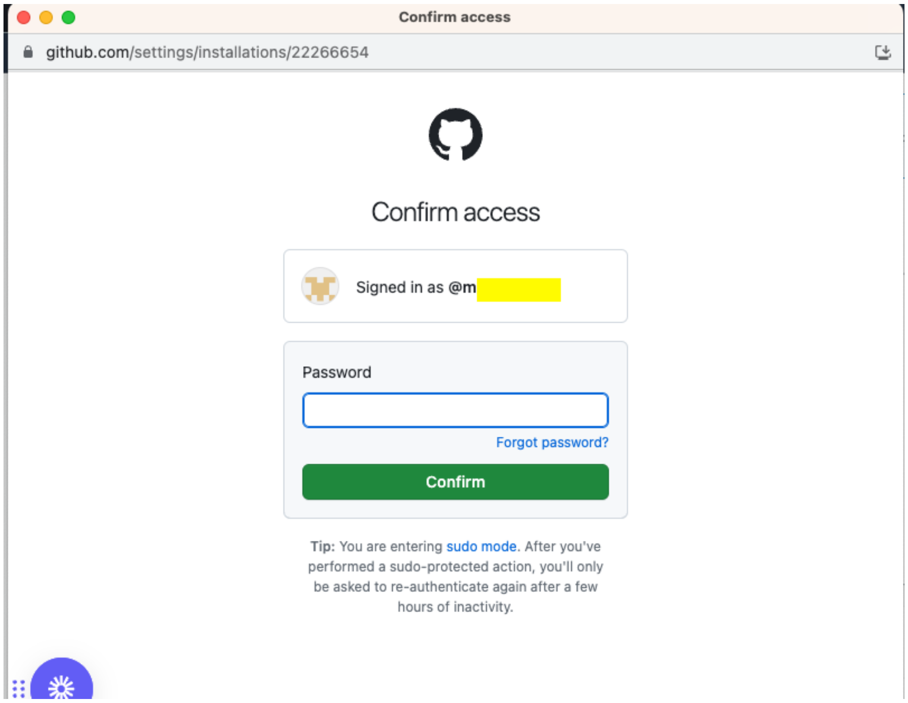
○ Container registry
Deploy your service using a container image stored in a container registry.

● Source code repository
Deploy your service using the code hosted in a source repository.

Provider
Choose the provider where you host your code repository.
GitHub

Github Connection Info

App Runner deploys your source code by installing an app called "AWS Connector for GitHub" in your account. You can install this app in your main GitHub account or in a GitHub organization.



Confirm access

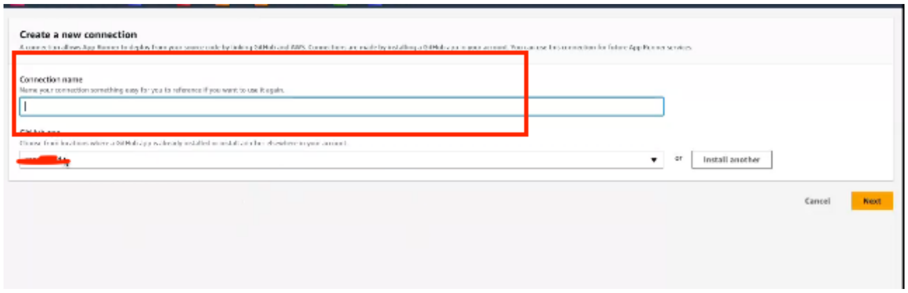
github.com/settings/installations/22266654

Confirm access

Signed in as @m[redacted]

Password

Tip: You are entering sudo mode. After you've performed a sudo-protected action, you'll only be asked to re-authenticate again after a few hours of inactivity.



Create a new connection

A connection allows App Runner to deploy from your source code by linking GitHub and AWS. Connections are made by installing a GitHub app in your account. You can use this connection for future App Runner services.

Connection name

Name your connection something easy for you to reference if you want to use it again.

GitHub app

Choose from locations where a GitHub app is already installed or install a new one elsewhere in your account.

[redacted] ▼ or Install another

Cancel Next

For this Sample application repository, please use the Github link on the reference section and fork the repository into your personal account so you can create the Apprunner Service using that forked Repo.

After Choosing the required repository and the right branch, you would need to choose the Deployment Setting as shown below, you have two option:

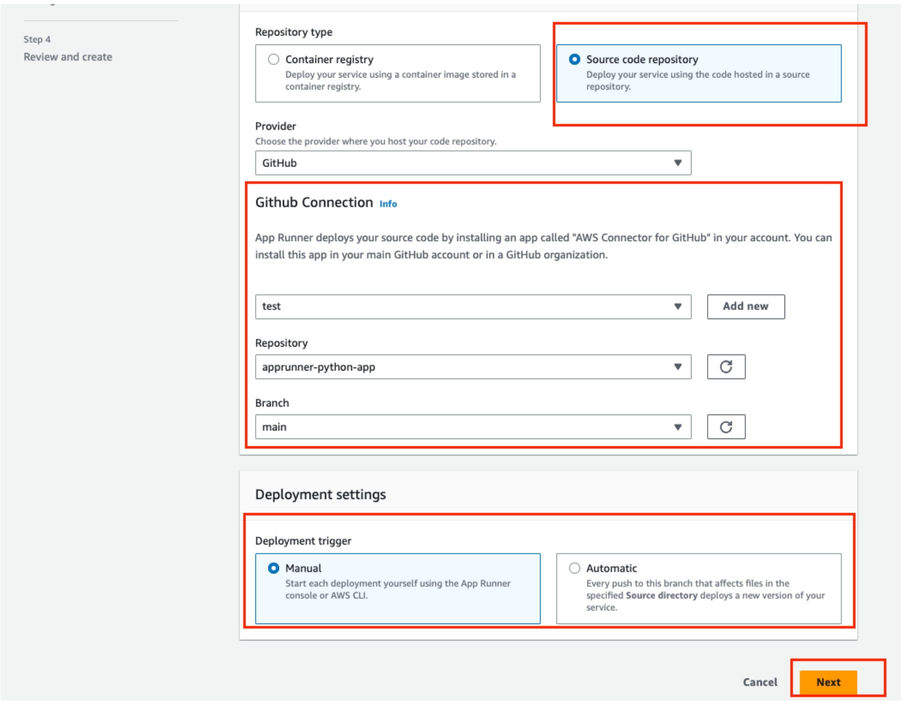
1- Manual:

Which will not reflect any changes happening on your repository unless you explicitly restart your service after you commit your code changes on Github.

2- Automatic:

This will instantly restart your running sevice once any changes committed on Github repository without any user interaction.

For more information please check out Deployment methods details. here:
 https://docs.aws.amazon.com/apprunner/latest/dg/manage-deploy.html#:~:text=Deployment-,methods,-App%20Runner%20provides



Step 4

Review and create

Repository type

○ Container registry
Deploy your service using a container image stored in a container registry.

● Source code repository
Deploy your service using the code hosted in a source repository.

Provider
Choose the provider where you host your code repository.

GitHub ▼

Github Connection Info

App Runner deploys your source code by installing an app called "AWS Connector for GitHub" in your account. You can install this app in your main GitHub account or in a GitHub organization.

test ▼ Add new

Repository

apprunner-python-app ▼

Branch

main ▼

Deployment settings

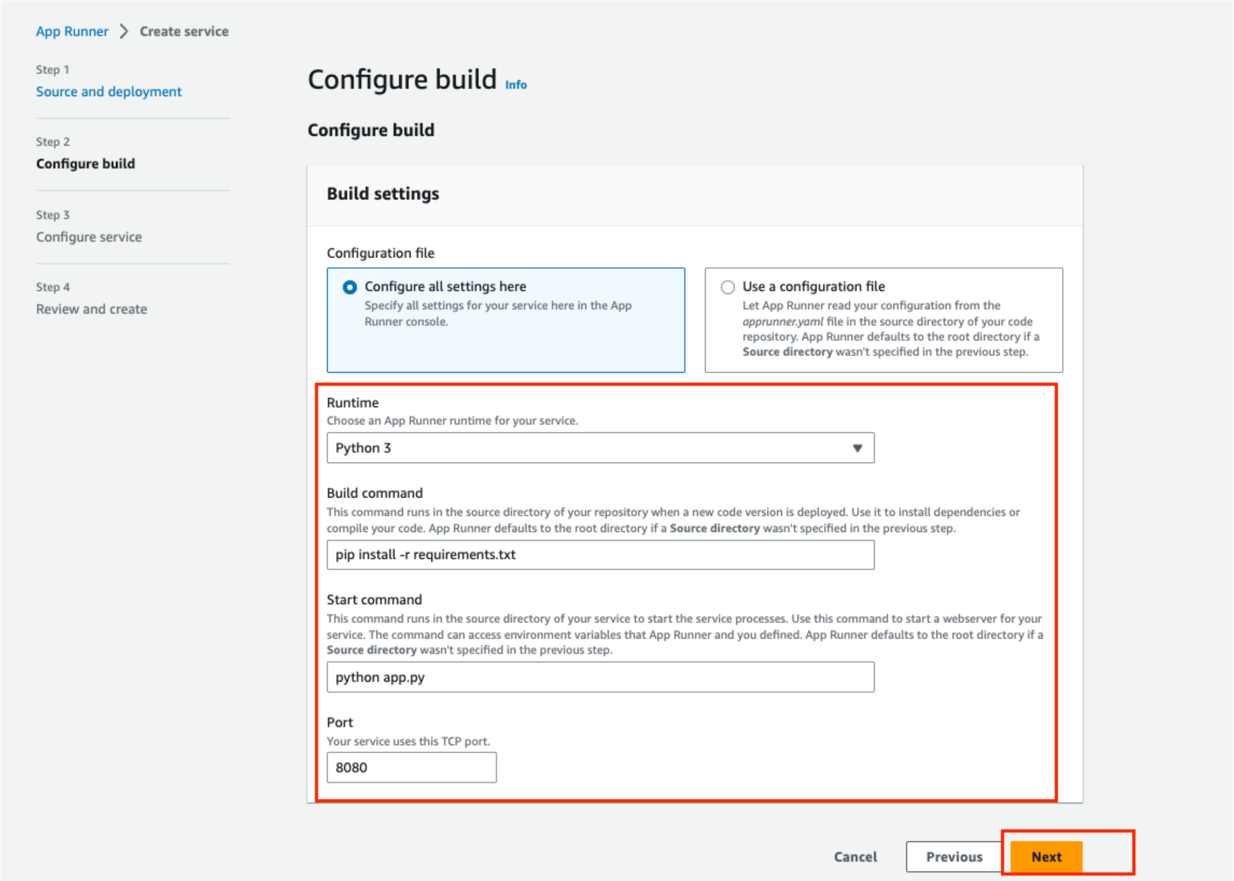
Deployment trigger

● Manual
Start each deployment yourself using the App Runner console or AWS CLI.

○ Automatic
Every push to this branch that affects files in the specified **Source directory** deploys a new version of your service.

Cancel Next

Next step is to choose your Runtime configuration, Build Command, Start Command and port number based on the language used in your code, I’m using Java for that Sample code so my configurations like below:



Configure build Info

Configure build

Build settings

Configuration file

● Configure all settings here
Specify all settings for your service here in the App Runner console.

○ Use a configuration file
Let App Runner read your configuration from the *apprunner.yaml* file in the source directory of your code repository. App Runner defaults to the root directory if a **Source directory** wasn't specified in the previous step.

Runtime
Choose an App Runner runtime for your service.
Python 3

Build command
This command runs in the source directory of your repository when a new code version is deployed. Use it to install dependencies or compile your code. App Runner defaults to the root directory if a **Source directory** wasn't specified in the previous step.
pip install -r requirements.txt

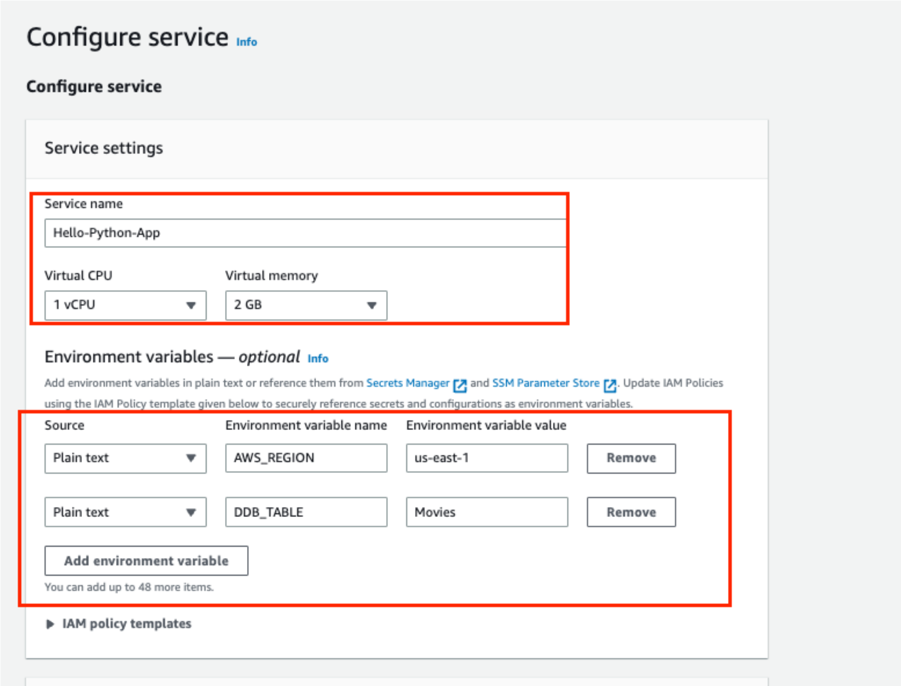
Start command
This command runs in the source directory of your service to start the service processes. Use this command to start a webserver for your service. The command can access environment variables that App Runner and you defined. App Runner defaults to the root directory if a **Source directory** wasn't specified in the previous step.
python app.py

Port
Your service uses this TCP port.
8080

Give a service name and make sure it would have enough CPU and Memory to start your application, one of the main features of using Apprunner that it takes care of compute resources scalability without any manual interaction from the user so even if you have spikes happening apprunner will scale the resources accordingly based on the number of HTTP/HTTPS request received

For my Sample Code I need to set couple of Environment variables so click on Add environment variable and add the following two environment variables:

	Key	Value
1	AWS_REGION	AWS Region ID (Eg: us-east-1)
2	DDB_TABLE	Movies



Configure service Info

Configure service

Service settings

Service name

Hello-Python-App

Virtual CPU: 1 vCPU

Virtual memory: 2 GB

Environment variables — *optional* Info

Add environment variables in plain text or reference them from Secrets Manager and SSM Parameter Store. Update IAM Policies using the IAM Policy template given below to securely reference secrets and configurations as environment variables.

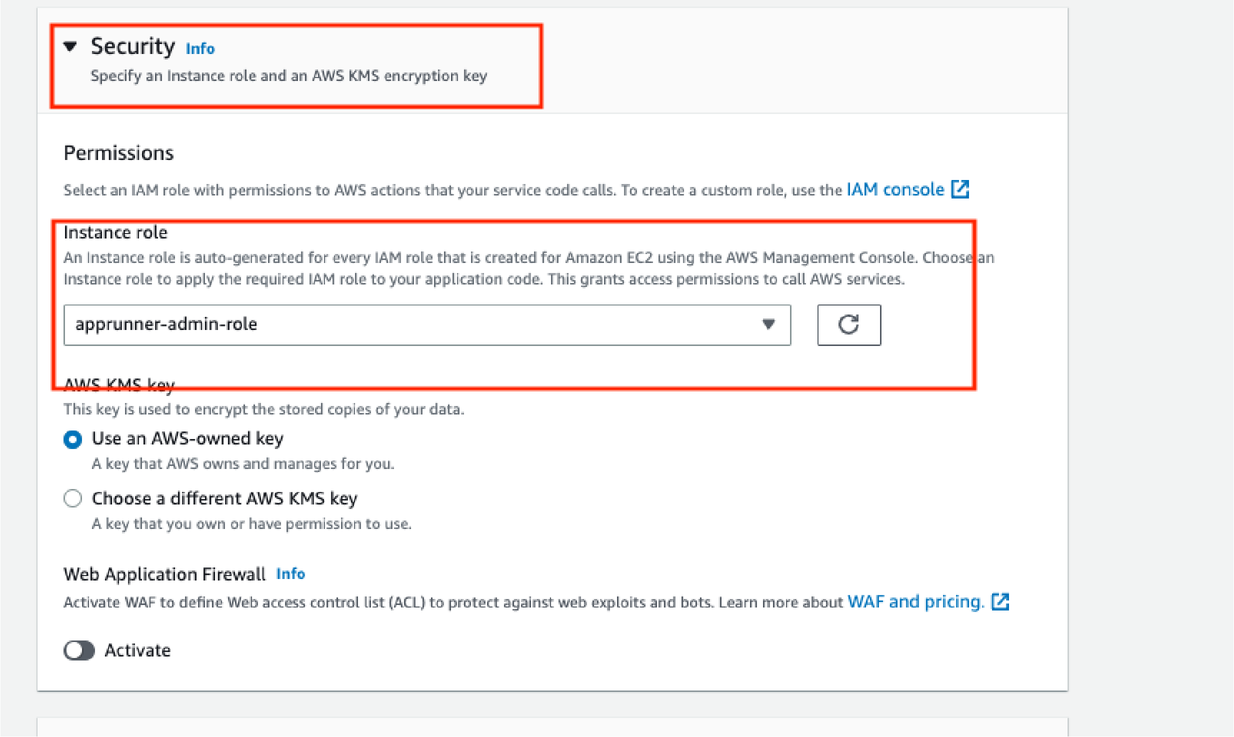
Source	Environment variable name	Environment variable value	
Plain text	AWS_REGION	us-east-1	Remove
Plain text	DDB_TABLE	Movies	Remove

Add environment variable

You can add up to 48 more items.

▶ IAM policy templates

Scroll down on the same page to the Security Section and make sure to choose the IAM Security Role we created in Cloud9 command line for Apprunner to give it access to S3 and DynamoDB



▼ Security Info

Specify an Instance role and an AWS KMS encryption key

Permissions

Select an IAM role with permissions to AWS actions that your service code calls. To create a custom role, use the IAM console

Instance role

An Instance role is auto-generated for every IAM role that is created for Amazon EC2 using the AWS Management Console. Choose an Instance role to apply the required IAM role to your application code. This grants access permissions to call AWS services.

apprunner-admin-role

AWS KMS key

This key is used to encrypt the stored copies of your data.

● Use an AWS-owned key

A key that AWS owns and manages for you.

○ Choose a different AWS KMS key

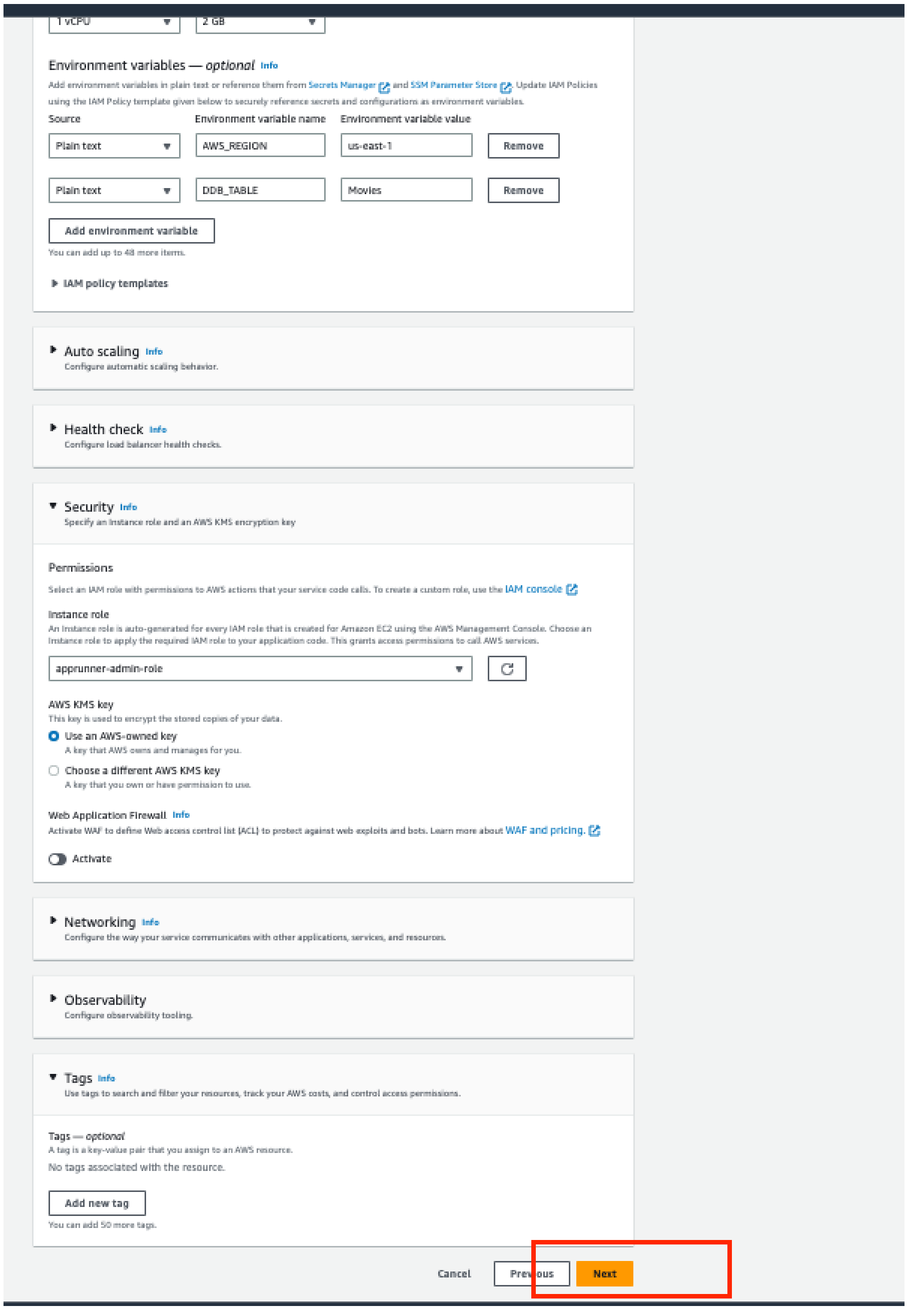
A key that you own or have permission to use.

Web Application Firewall Info

Activate WAF to define Web access control list (ACL) to protect against web exploits and bots. Learn more about WAF and pricing.

Activate

Finally, we will have to review the create the service if everything looks good as explained previously :



1 vCPU ▾ 2 GB ▾

Environment variables — *optional* Info

Add environment variables in plain text or reference them from Secrets Manager and SSM Parameter Store. Update IAM Policies using the IAM Policy template given below to securely reference secrets and configurations as environment variables.

Source	Environment variable name	Environment variable value	
Plain text ▾	AWS_REGION	us-east-1	Remove
Plain text ▾	DDB_TABLE	Movies	Remove

Add environment variable

You can add up to 48 more items.

▸ IAM policy templates

▸ Auto scaling Info

Configure automatic scaling behavior.

▸ Health check Info

Configure load balancer health checks.

▾ Security Info

Specify an instance role and an AWS KMS encryption key

Permissions

Select an IAM role with permissions to AWS actions that your service code calls. To create a custom role, use the IAM console

Instance role

An instance role is auto-generated for every IAM role that is created for Amazon EC2 using the AWS Management Console. Choose an instance role to apply the required IAM role to your application code. This grants access permissions to call AWS services.

apprunner-admin-role ▾

AWS KMS key

This key is used to encrypt the stored copies of your data.

● Use an AWS-owned key
A key that AWS owns and manages for you.

○ Choose a different AWS KMS key
A key that you own or have permission to use.

Web Application Firewall Info

Activate WAF to define Web access control list (ACL) to protect against web exploits and bots. Learn more about WAF and pricing.

Activate

▸ Networking Info

Configure the way your service communicates with other applications, services, and resources.

▸ Observability

Configure observability tooling.

▾ Tags Info

Use tags to search and filter your resources, track your AWS costs, and control access permissions.

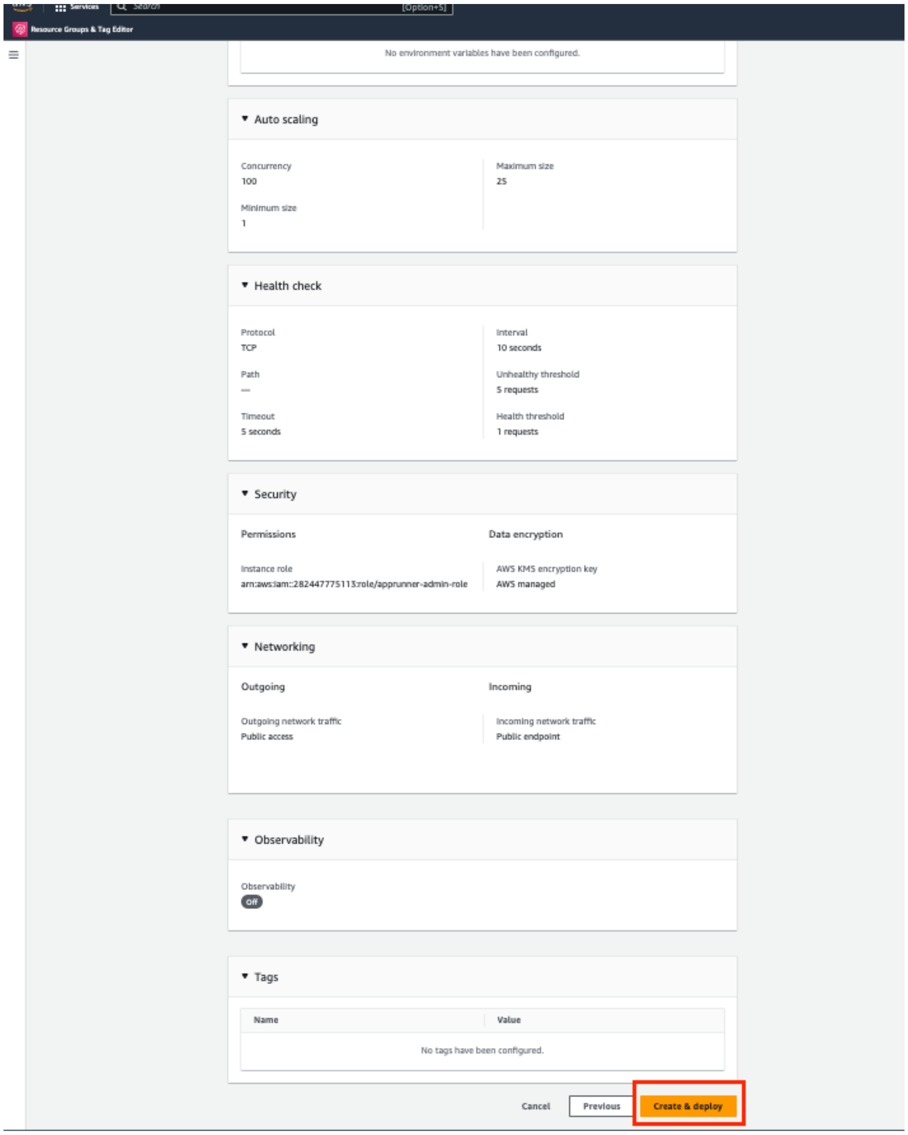
Tags — *optional*

A tag is a key-value pair that you assign to an AWS resource.

No tags associated with the resource.

Add new tag

You can add 50 more tags.



No environment variables have been configured.

Auto scaling

Concurrency: 100
Maximum size: 25
Minimum size: 1

Health check

Protocol: TCP
Interval: 10 seconds
Path: —
Unhealthy threshold: 5 requests
Timeout: 5 seconds
Health threshold: 1 requests

Security

Permissions
Instance role: arn:aws:iam::282447775113:role/apprunner-admin-role

Data encryption
AWS KMS encryption key: AWS managed

Networking

Outgoing
Outgoing network traffic: Public access

Incoming
Incoming network traffic: Public endpoint

Observability

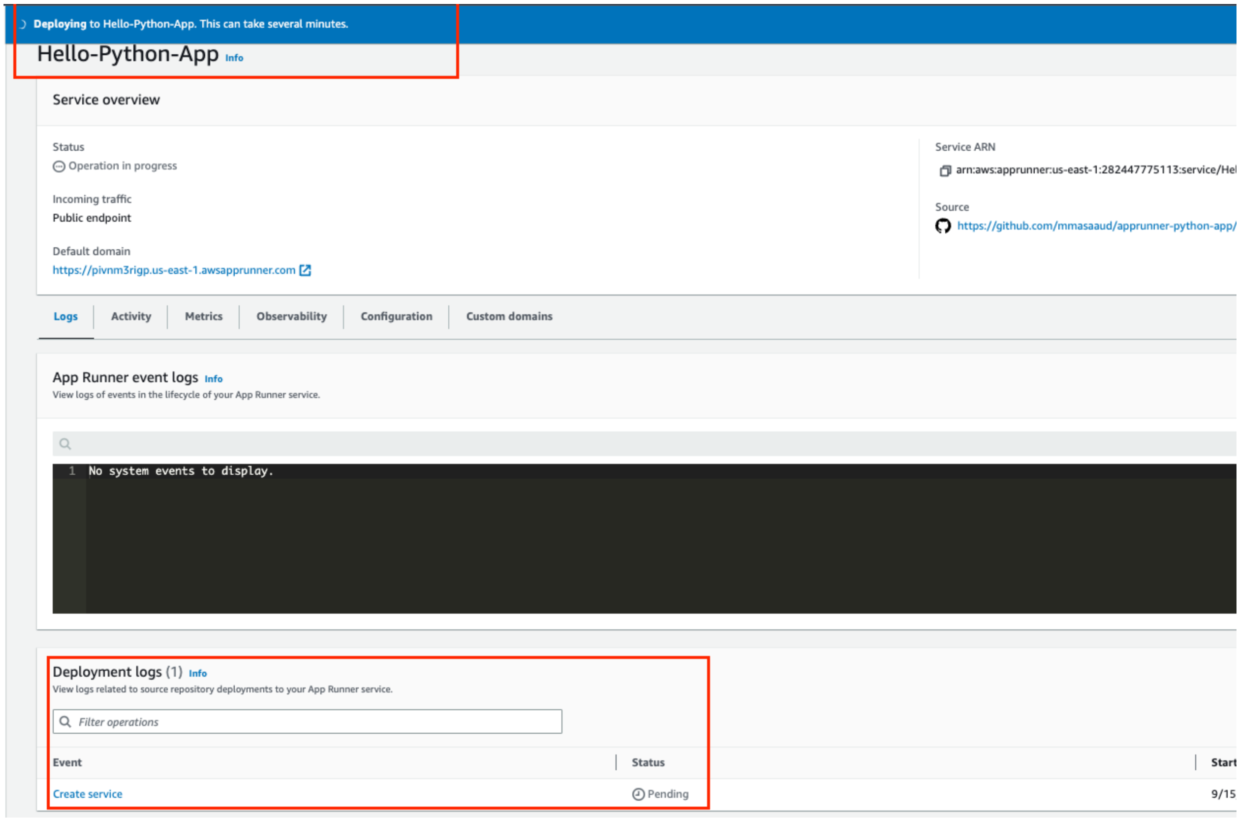
Observability: Off

Tags

Name	Value
No tags have been configured.	

This will take between 5-7 minute to create the service, you can also check the deployment log while deploying is taking place :

After few minutes, service created Successfully and you can access if from the default domain link, which will be created automatically as a part of service creation :



Deploying to Hello-Python-App. This can take several minutes.

Hello-Python-App

Service overview

Status
Operation in progress

Incoming traffic
Public endpoint

Default domain
https://pivnm3rigp.us-east-1.awsapprunner.com

Service ARN
arn:aws:apprunner:us-east-1:282447775113:service/Hel

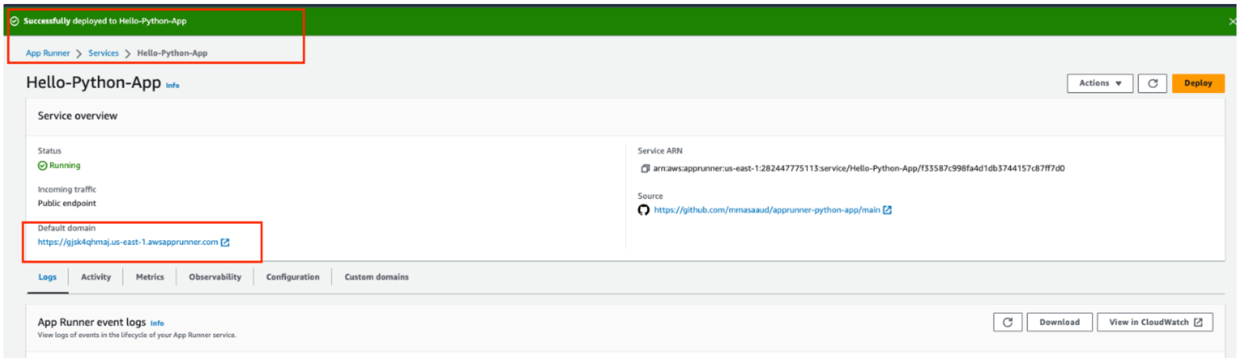
Source
https://github.com/mmasaaud/apprunner-python-app/

App Runner event logs
View logs of events in the lifecycle of your App Runner service.

```
1  No system events to display.
```

Deployment logs (1)
View logs related to source repository deployments to your App Runner service.

Event	Status	Start
Create service	Pending	9/15



Successfully deployed to Hello-Python-App

Hello-Python-App

Service overview

Status
Running

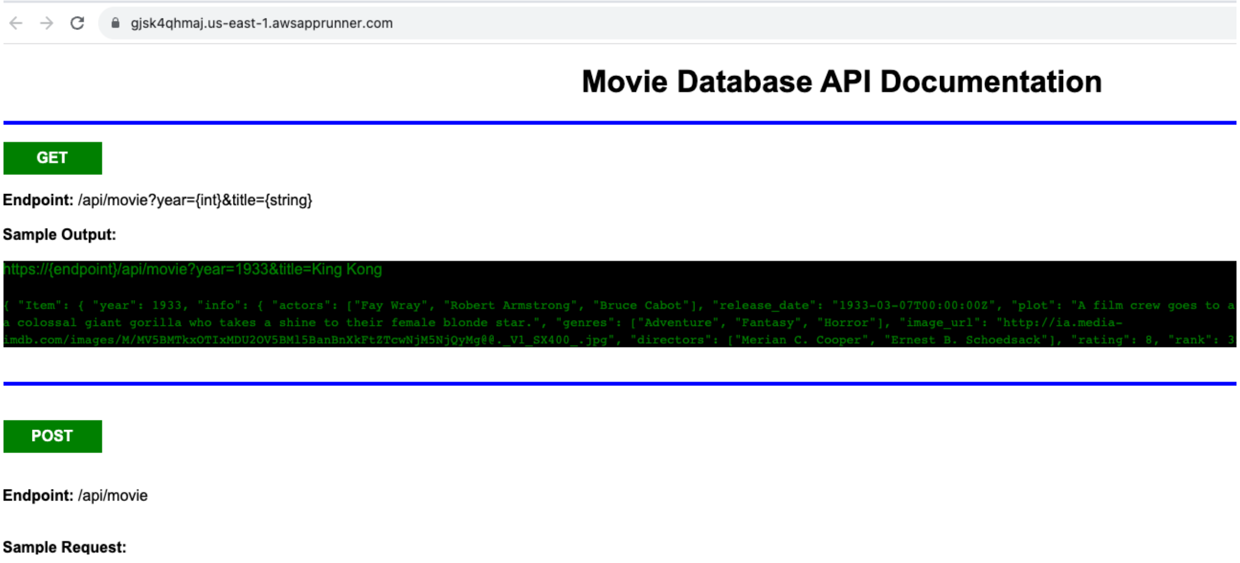
Incoming traffic
Public endpoint

Default domain
https://gjsk4qhmaj.us-east-1.awsapprunner.com

Service ARN
arn:aws:apprunner:us-east-1:282447775113:service/Hello-Python-App/f33587c998fa4d1db3744157c87ff7d0

Source
https://github.com/mmasaaud/apprunner-python-app/main

App Runner event logs
View logs of events in the lifecycle of your App Runner service.



Movie Database API Documentation

GET

Endpoint: /api/movie?year={int}&title={string}

Sample Output:

```
https://{endpoint}/api/movie?year=1933&title=King Kong

{ "Item": { "year": 1933, "info": { "actors": ["Fay Wray", "Robert Armstrong", "Bruce Cabot"], "release_date": "1933-03-07T00:00:00Z", "plot": "A film crew goes to a
a colossal giant gorilla who takes a shine to their female blonde star.", "genres": ["Adventure", "Fantasy", "Horror"], "image_url": "http://ia.media-
imdb.com/images/M/MV5BMTkxOTIxMDU2OV5BMl5BanBnXkFtZTcwNjM5NjQyMg@@._V1_SX400_.jpg", "directors": ["Merian C. Cooper", "Ernest B. Schoedsack"], "rating": 8, "rank": 3
```

POST

Endpoint: /api/movie

Sample Request:

Importnat Links :

1- AWS SDK for Python :
https://aws.amazon.com/developer/language/python/

2- Github link for Sample Python app :
https://github.com/mmasaaud/apprunner-python-app

3- DynamoDB document :
https://us-east-2.console.aws.amazon.com/dynamodbv2/home?region=us-east-2#service

4- S3 document :
https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html