EXPERIMENT No.09: Orchestrating Serverless Functions with AWS Step Functions

9.1 Objectives: Orchestrating Serverless Functions with AWS Step Functions

9.2 Apparatus Required:

AWS Account

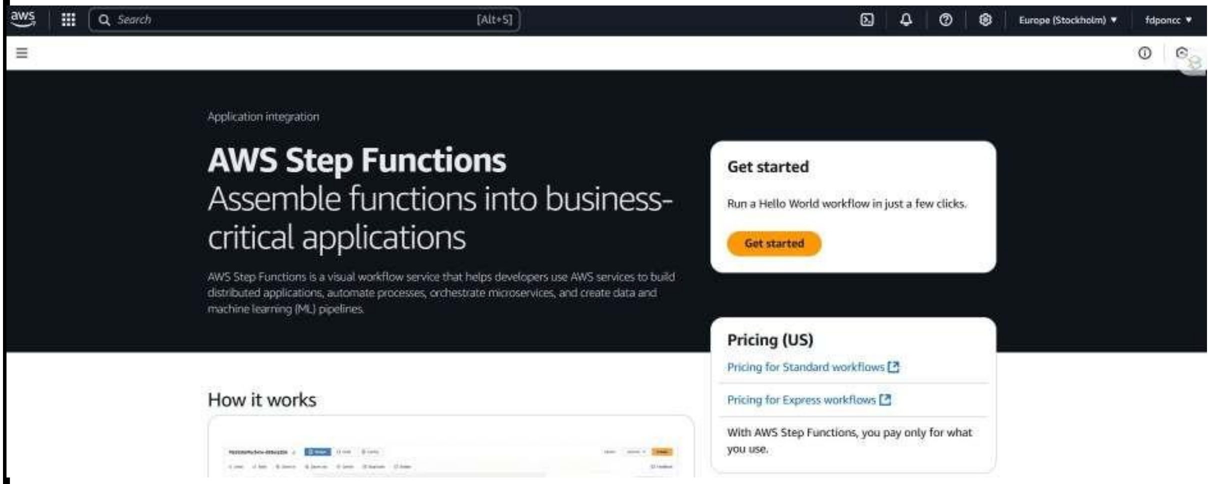
9.3 Pre-Requisite:

AWS basics, AWS step functions

9.4 Introduction:

Serverless functions let developers code in any language or framework with which they're comfortable. Simpler backend code. Serverless removes a lot of coding complexity for developers, allowing them to create simple, self-contained functions that independently perform one purpose.

GO TO STEP FUNCTION CONSOLE



Application integration

AWS Step Functions
Assemble functions into business-critical applications

AWS Step Functions is a visual workflow service that helps developers use AWS services to build distributed applications, automate processes, orchestrate microservices, and create data and machine learning (ML) pipelines.

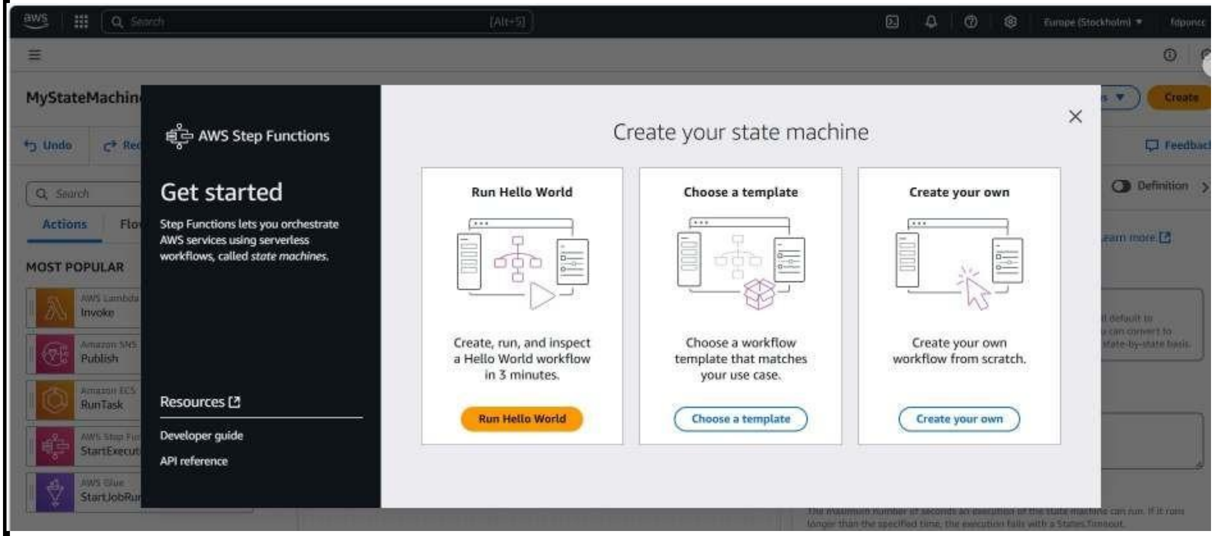
Get started

Run a Hello World workflow in just a few clicks.

Pricing (US)

With AWS Step Functions, you pay only for what you use.

How it works



AWS Step Functions

Get started

Step Functions lets you orchestrate AWS services using serverless workflows, called *state machines*.

Resources

Developer guide

API reference

Create your state machine

Run Hello World

Create, run, and inspect a Hello World workflow in 3 minutes.

Run Hello World

Choose a template

Choose a workflow template that matches your use case.

Choose a template

Create your own

Create your own workflow from scratch.

Create your own

CREATE YOUR OWN

########################

Lab 1. Step Function to Select/Reject a student based on Maths/Physics cut-off mark

########################

Task 1: Create SNS topic

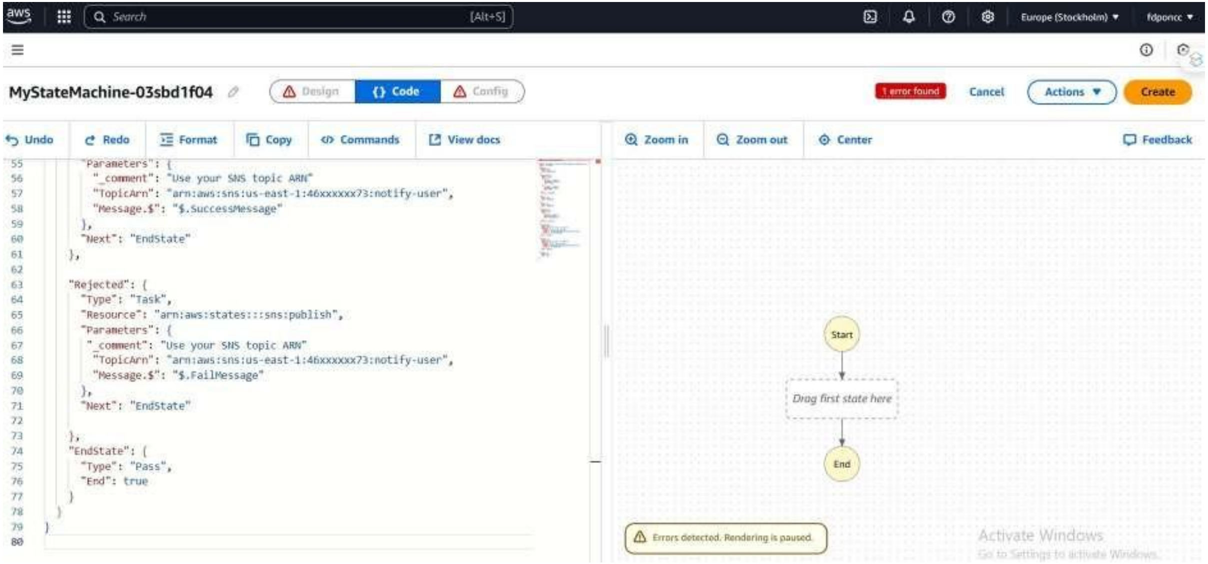
======================

1. Create an SNS topic. Subscribe your mail id to the topic.

Task 2: Definition of Step Function

==============

1. **Goto Step Functions console.**



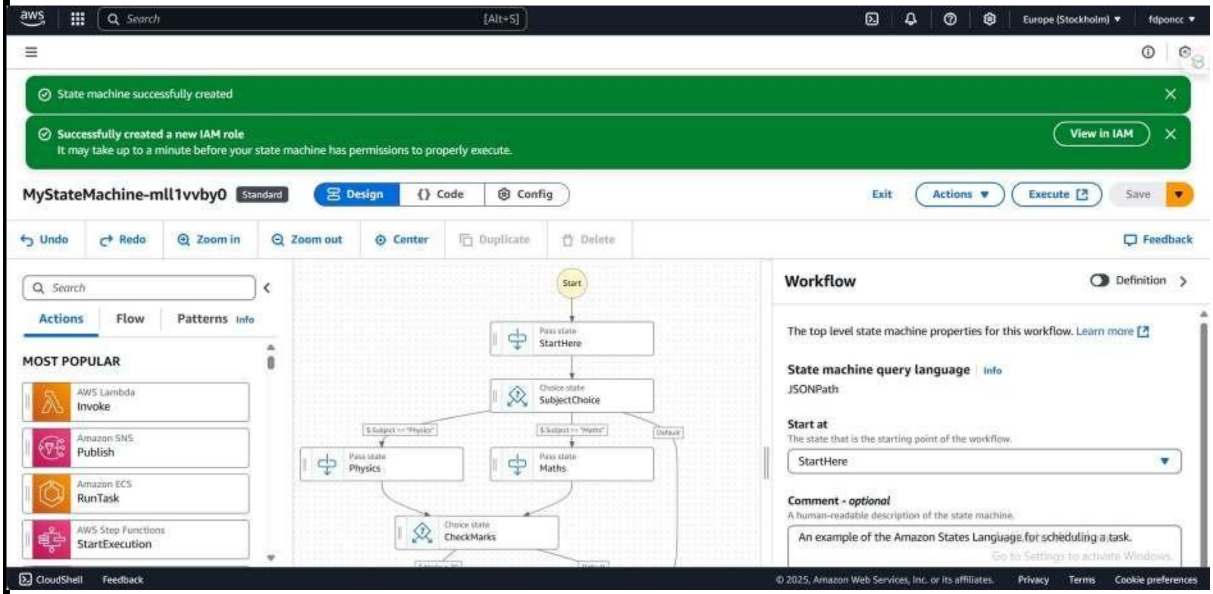
MyStateMachine-03sbd1f04

```
        "Parameters": {
          "_comment": "Use your SNS topic ARN"
          "TopicArn": "arn:aws:sns:us-east-1:46xxxxxx73:notify-user",
          "Message.$": "$.SuccessMessage"
        },
        "Next": "EndState"
      },

      "Rejected": {
        "Type": "Task",
        "Resource": "arn:aws:states:::sns:publish",
        "Parameters": {
         "_comment": "Use your SNS topic ARN"
          "TopicArn": "arn:aws:sns:us-east-1:46xxxxxx73:notify-user",
          "Message.$": "$.FailMessage"
        },
        "Next": "EndState"

      },
      "EndState": {
        "Type": "Pass",
        "End": true
      }
    }
  }
```

Create a State Machine. Use the below code.

Name: course_selection_state_machine

```
{
  "Comment": "An example of the Amazon States Language for scheduling a task.",
  "StartAt": "StartHere",
  "States": {
    "StartHere": {
      "Type": "Pass",
      "Next": "SubjectChoice"
    },
    "SubjectChoice": {
      "Type": "Choice",
      "Choices": [
        {
          "Variable": "$.Subject",
          "StringEquals": "Physics",
          "Next": "Physics"
        },
        {
          "Variable":  "$.Subject",
          "StringEquals":  "Maths",
          "Next": "Maths"
        }
      ],
      "Default": "EndState"
    },
    "Physics": {
      "Type": "Pass",
      "Next": "CheckMarks"
    },
    "Maths": {
      "Type": "Pass",
      "Next": "CheckMarks"
    },
    "CheckMarks": {
      "Type": "Choice",
      "Choices": [
        {
          "Variable": "$.Marks",
          "NumericGreaterThan": 70,
          "Next": "EndState"
        }
      ],
      "Default": "EndState"
    },
    "EndState": {
      "Type": "Pass",
      "End": true
      }
    }
}
```



State machine successfully created

Successfully created a new IAM role
It may take up to a minute before your state machine has permissions to properly execute.

MyStateMachine-mll1vvby0 Standard

Workflow

The top level state machine properties for this workflow. Learn more

State machine query language: JSONPath

Start at: StartHere
The state that is the starting point of the workflow.

Comment - *optional*: An example of the Amazon States Language for scheduling a task.
A human-readable description of the state machine.

2. Execute the state machine. Provide the below timer value as the input.

Execution input:

```
{
 "Subject": "Maths",
 "Marks": 76
}

{
 "Subject": "Physics",
 "Marks": 91
}
```

3. Observe the graph view. Observe each step turning green.

4. Cleanup - delete the step function

#####

End

#####

EXPERIMENT No. 10: Automating Application Deployment Using CI/CD Pipeline

9.1 Objectives: Automating Application Deployment Using CI/CD Pipeline

9.2 Apparatus Required:

AWS Account

9.3 Pre-Requisite:

AWS knowledge, basic CI/CD knowledge, IAM Roles, AWS Codedeploy

9.4 Introduction:

Serverless functions let developers code in any language or framework with which they're comfortable. Simpler backend code. Serverless removes a lot of coding complexity for developers, allowing them to create simple, self-contained functions that independently perform one purpose.

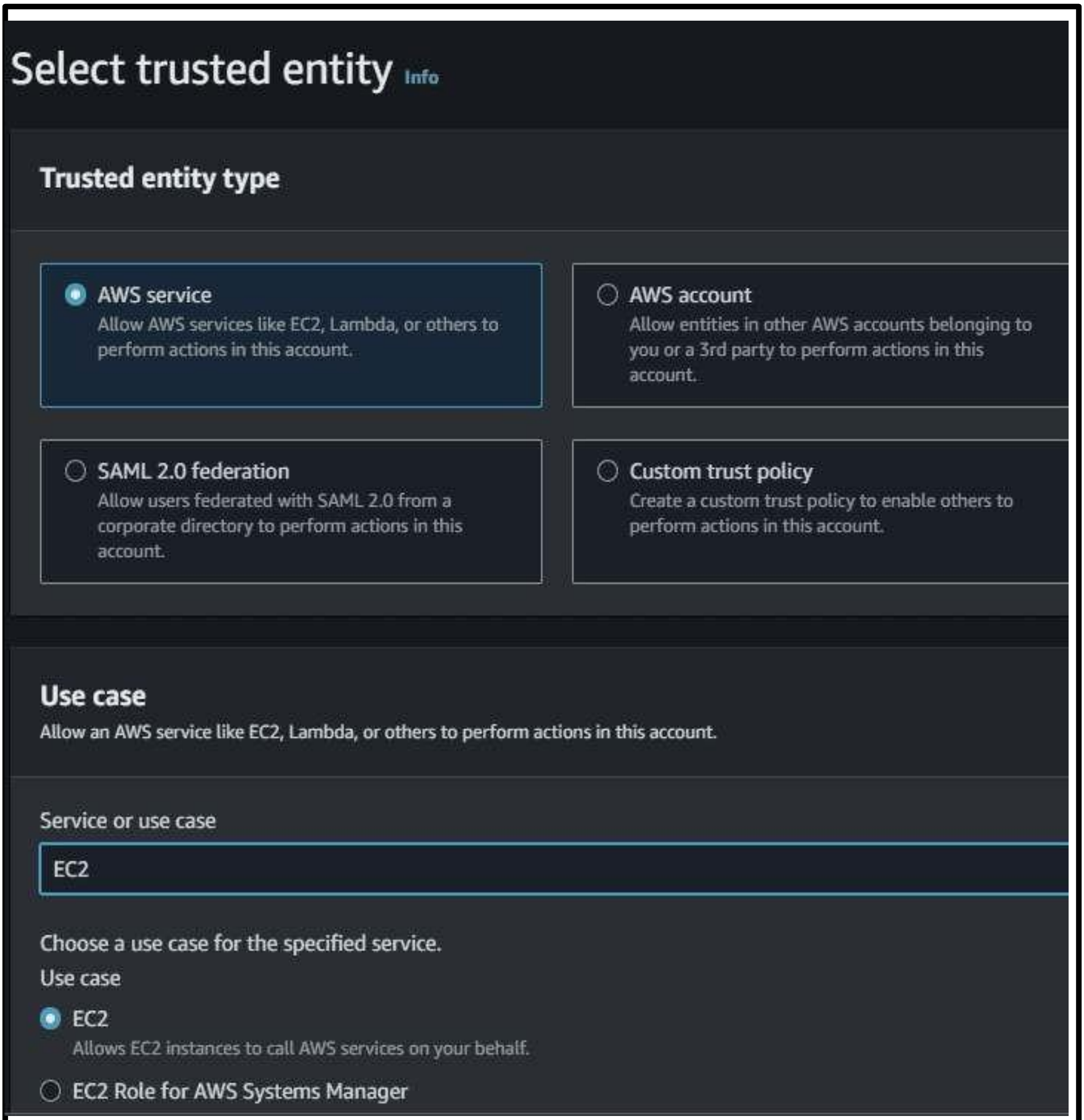
A continuous integration and continuous deployment (CI/CD) pipeline is a series of steps that must be performed in order to deliver a new version of software. CI/CD pipelines are a practice focused on improving software delivery throughout the software development life cycle via automation

AWS IAM - AWS Identity and Access Management (IAM) is a web service for securely controlling access to AWS services. With IAM, you can centrally manage users, security credentials such as access keys, and permissions that control which AWS resources users and applications can access.

AWS CodeDeploy is a service that automates code deployments to any instance, including Amazon EC2 instances and instances running on-premises. AWS CodeDeploy makes it easier for you to rapidly release new features, helps you avoid downtime during deployment, and handles the complexity of updating your applications.

Step 1: Create IAM Role for EC2 and AWS CodeDeploy

- **Navigate to IAM service.**
- **Then go to roles and create a new role.**
- **Select trusted entity type as AWS Service and use case as EC2**



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.

○ 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

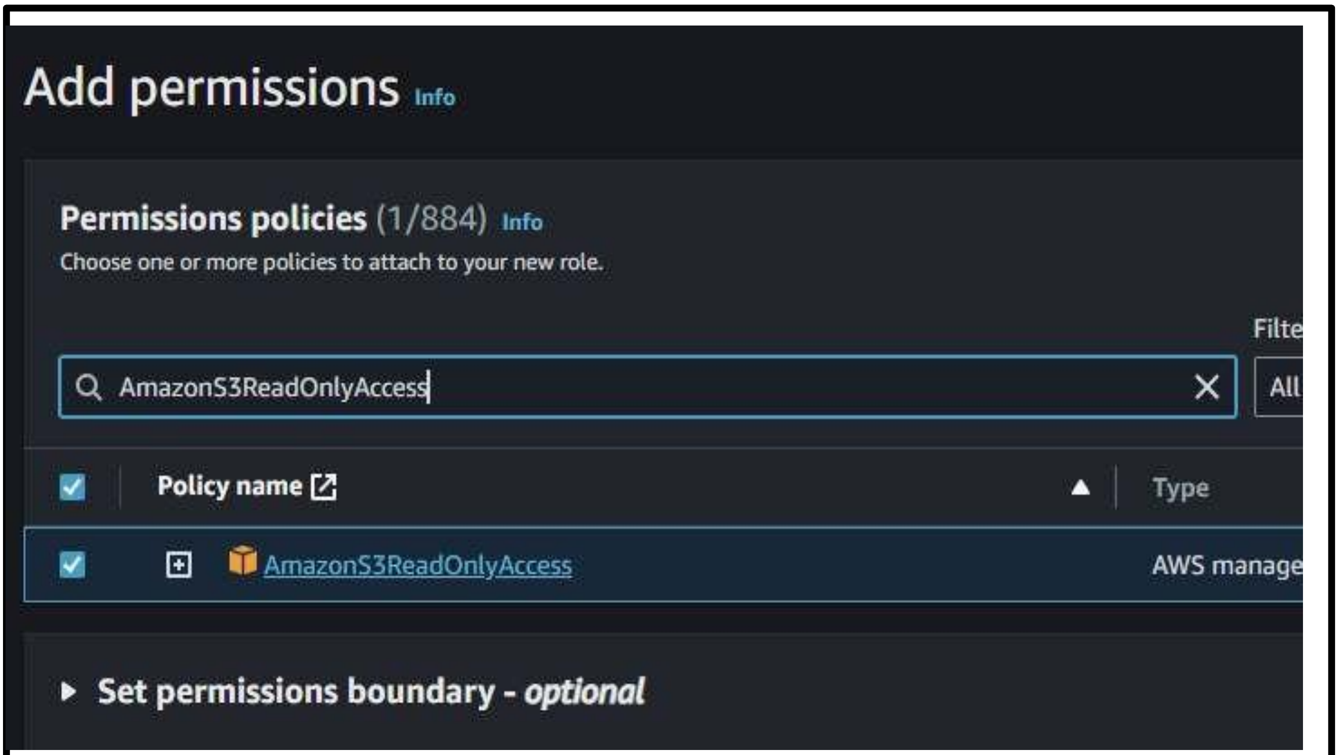
Choose a use case for the specified service.
Use case

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

○ EC2 Role for AWS Systems Manager

Step 2: Add permissions To IAM Role

- **Select AmazonS3ReadOnlyAccess permission. It will allow our EC2 instance to access stored artifacts from the Amazon S3 bucket.**



Add permissions Info

Permissions policies (1/884) Info

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

Filte

AmazonS3ReadOnlyAccess

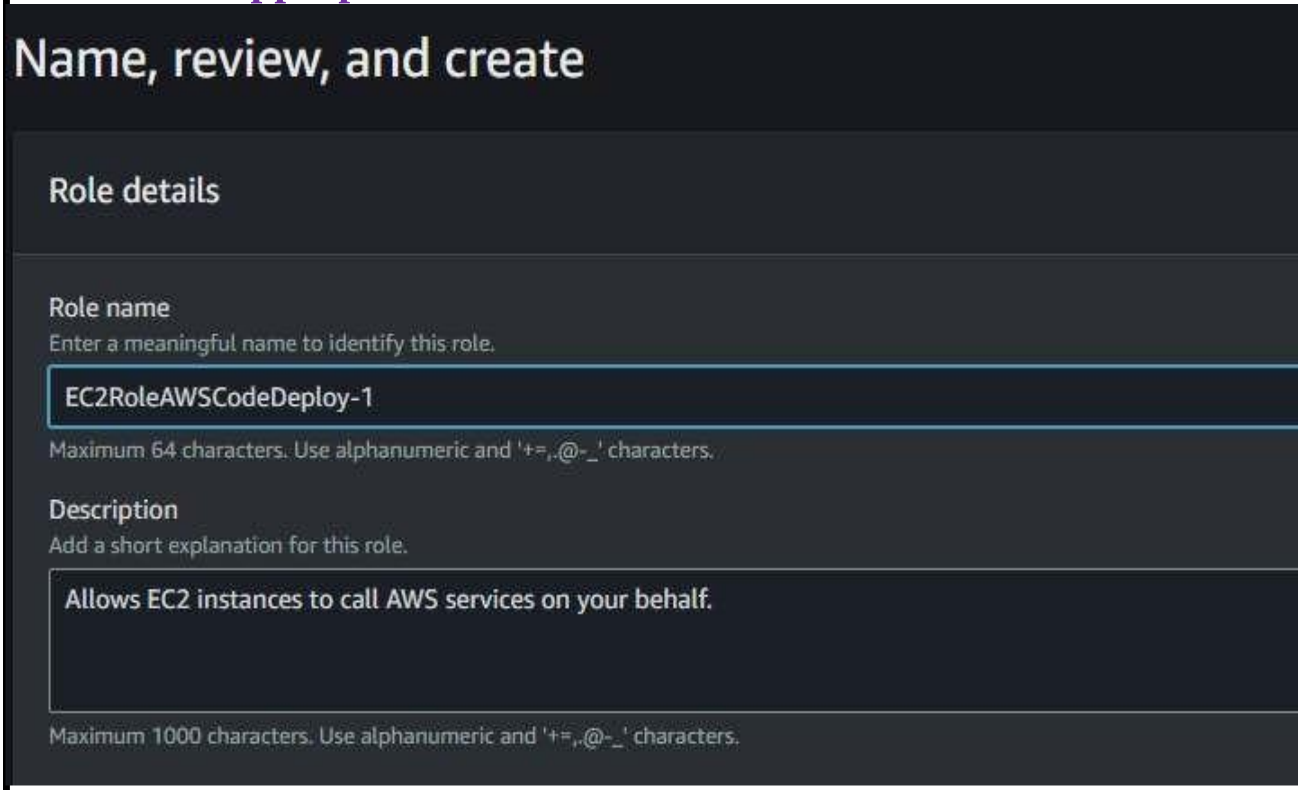
All

☑	Policy name	Type
☑	AmazonS3ReadOnlyAccess	AWS manage

▸ Set permissions boundary - *optional*

Step 3: Creating The Role For AWS CodeDeploy

- **Provide the Name, review and Click on Create for creating the Role.**
- **Select an appropriate role name and click on create role.**



Name, review, and create

Role details

Role name

Enter a meaningful name to identify this role.

EC2RoleAWSCodeDeploy-1

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

Description

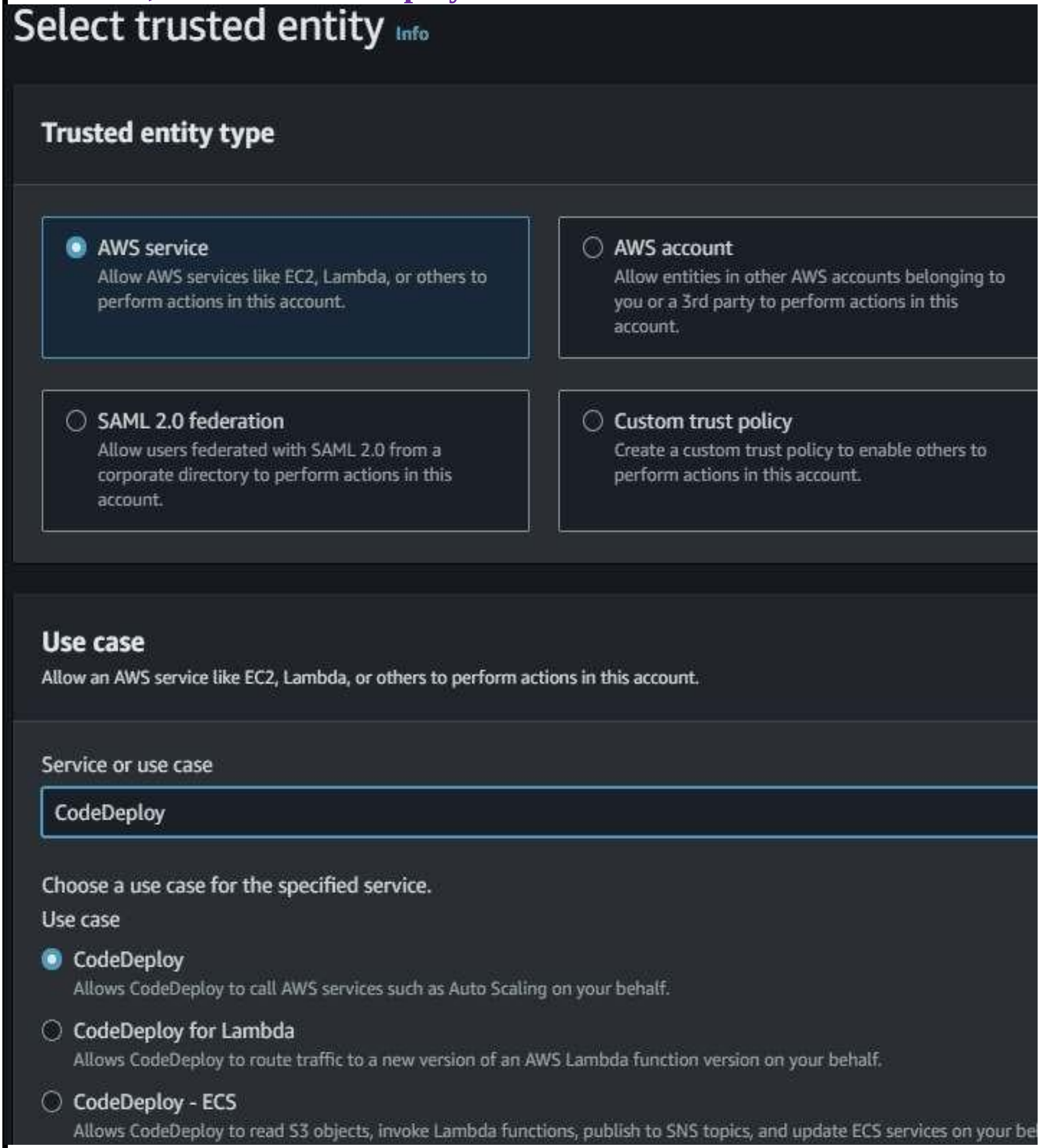
Add a short explanation for this role.

Allows EC2 instances to call AWS services on your behalf.

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

Step 4: Creating New Service Role For CodeDeploy

- **Create a new service role for CodeDeploy and attach AWSCodeDeployRole policy which will provide the permissions for our service role to read tags of our EC2 instance, publish information to Amazon SNS topics and much more task.**
- **Repeat the Above 3 steps again with trusted entity type AWS Service, use case CodeDeploy.**



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.

○ 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
CodeDeploy

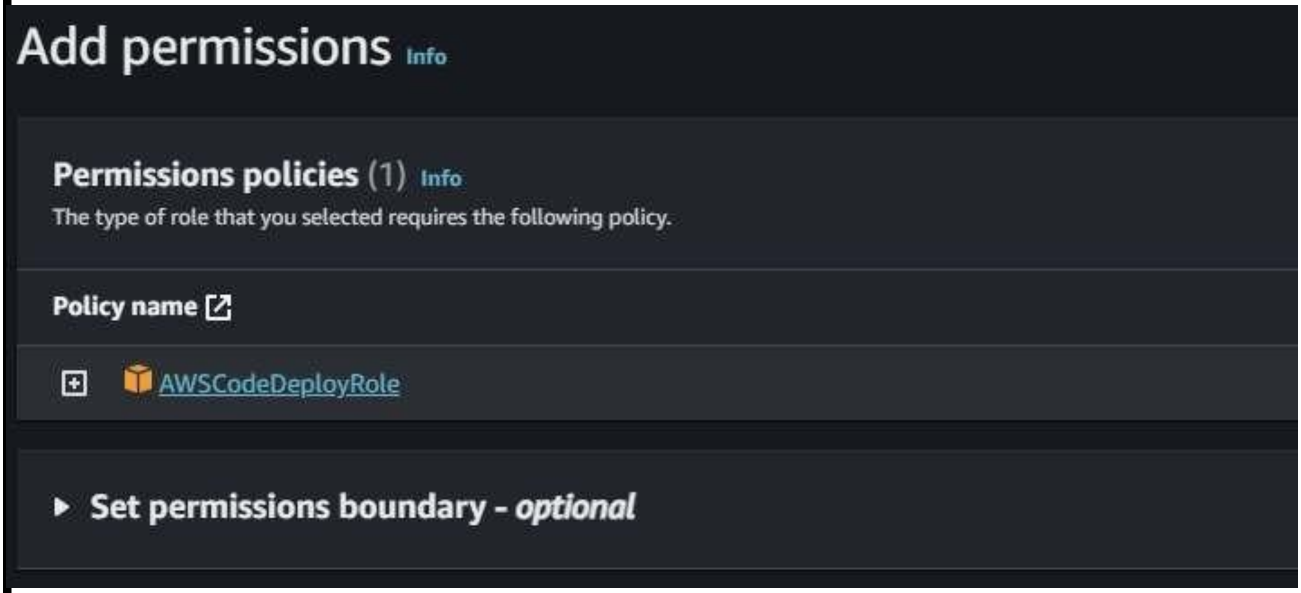
Choose a use case for the specified service.
Use case

● CodeDeploy
Allows CodeDeploy to call AWS services such as Auto Scaling on your behalf.

○ CodeDeploy for Lambda
Allows CodeDeploy to route traffic to a new version of an AWS Lambda function version on your behalf.

○ CodeDeploy - ECS
Allows CodeDeploy to read S3 objects, invoke Lambda functions, publish to SNS topics, and update ECS services on your be

- **Add AWSCodeDeployRole permissions to this creating Role**



Add permissions Info

Permissions policies (1) Info

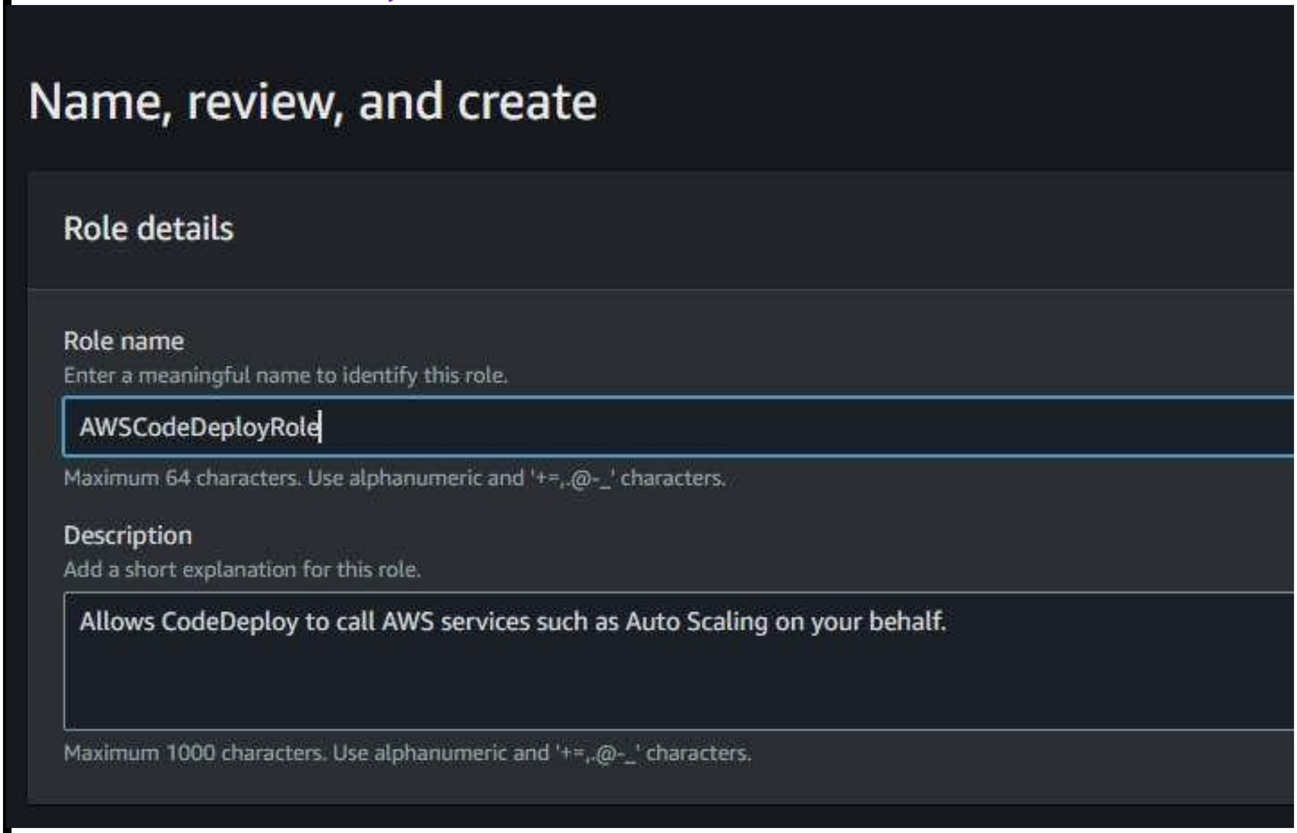
The type of role that you selected requires the following policy.

Policy name

⊞ AWSCodeDeployRole

▸ Set permissions boundary - *optional*

- **Provide the Name, review and create the role.**



Name, review, and create

Role details

Role name

Enter a meaningful name to identify this role.

AWSCodeDeployRole

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

Description

Add a short explanation for this role.

Allows CodeDeploy to call AWS services such as Auto Scaling on your behalf.

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

Step 5: Launch An Linux EC2 instance

- **Select the instance with AMI such as "Amazon Linux" and connect to CLI Console.**
- **Switch to root user from ec2-user to gain admin access power by using following command "sudo su" in Linux.**

```
sudo su
```

Step 6: Update The Packages

- The command "sudo yum update" is used in Amazon Linux, CentOS, and Red Hat Linux distributions to update installed packages on your system to their latest available versions.

```
sudo yum update
```

Step 7: Install The Ruby And Wget Software

- The command 'sudo yum install ruby' is used to install the Ruby programming software using the YUM package manager.

```
sudo yum install ruby
```

- The command sudo yum install wget is used to install the "wget" package on a system running Amazon Linux, CentOS, or other Red Hat-based Linux distributions that use the YUM package manager.

```
sudo yum install wget
```

Step 8: Download CodeDeploy Agent Script

- Downloading the AWS CodeDeploy agent installation script from the AWS S3 bucket is an essential step in setting up AWS CodeDeploy for your infrastructure.
- The CodeDeploy agent is a lightweight, scalable software component that enables AWS CodeDeploy to deploy and manage applications on your EC2 instances or on-premises servers.

```
wget https://aws-codedeploy-us-east-1.s3.amazonaws.com/latest/install
```

Step 9: Run Installation Script

- The command chmod +x ./install is used to make a file executable in a Unix-like operating system, including Linux.

```
chmod +x ./install
```

The command 'sudo ./install auto' is likely used to run an installation script with superuser (administrator) privileges and pass the "auto" argument to the script.

```
sudo ./install auto
```

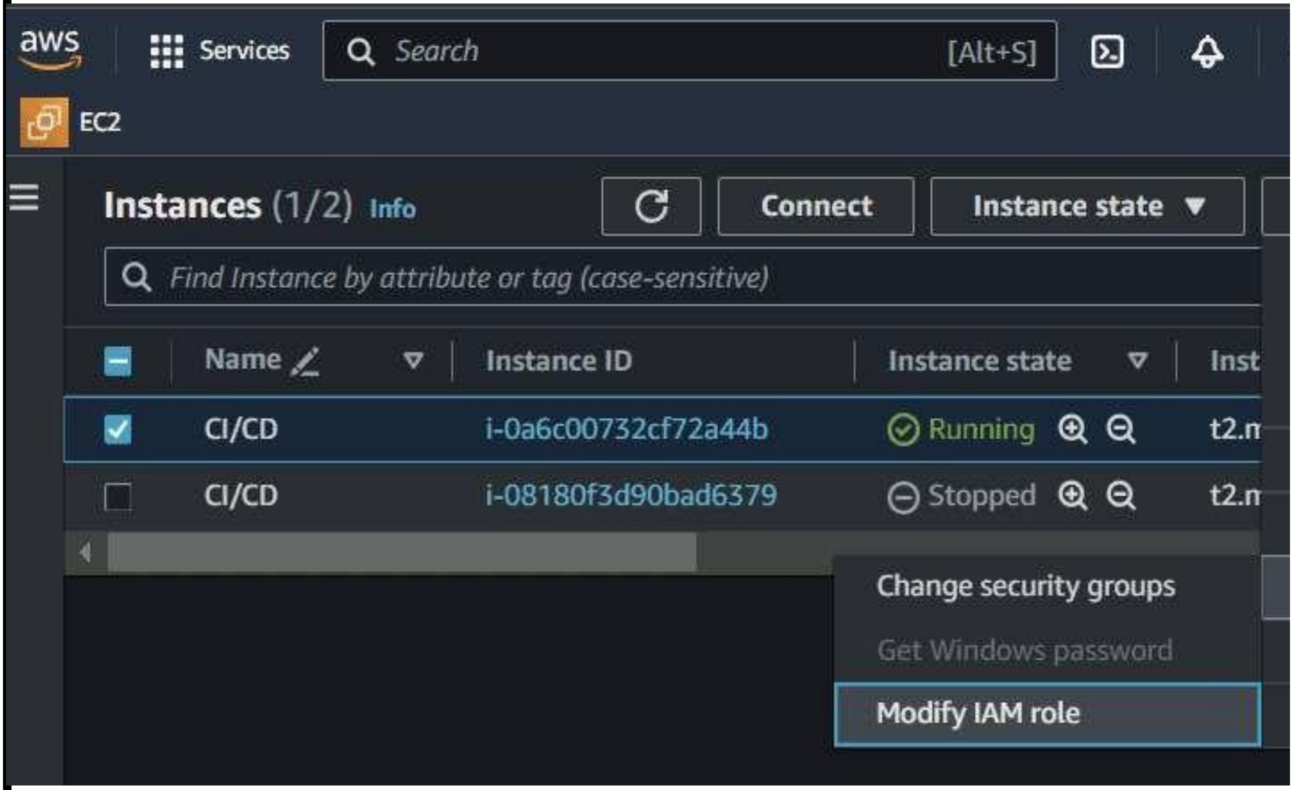
Step 10: Check CodeDeploy Agent Status

- The command sudo service codedeploy-agent status is used to check the status of the AWS CodeDeploy agent running on your system.

```
sudo service codedeploy-agent status
```

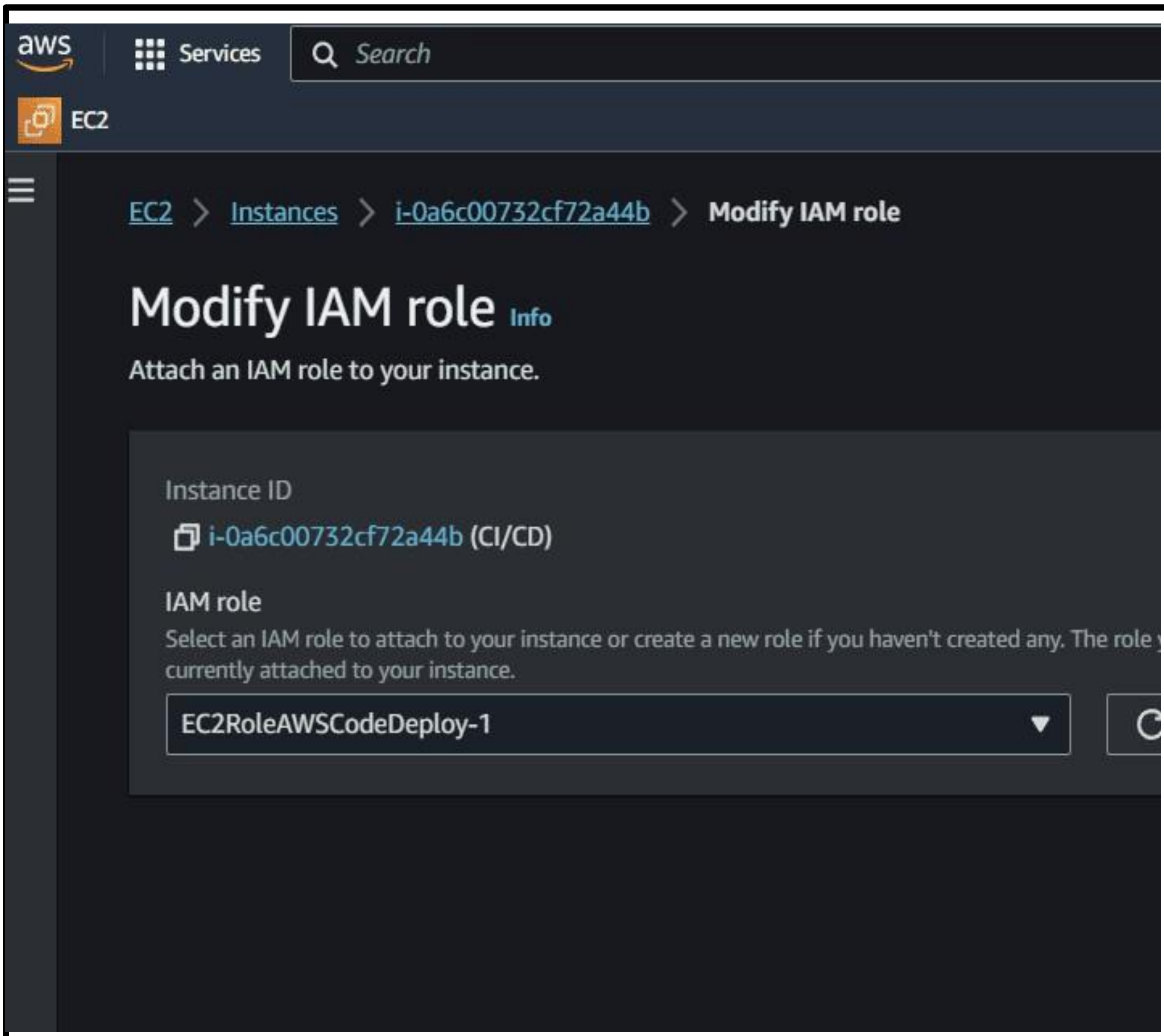
Step 11: Modifying IAM Role

- After running the following commands, select the instance and click on "Actions", then click on "Security" and click on "Modify IAM Role". Then choose the above created IAM Role and click on "Update IAM Role".
- After this step, your EC2 instance gets attached with your above created IAM Role.



	Name	Instance ID	Instance state	Inst
☑	CI/CD	i-0a6c00732cf72a44b	Running	t2.n
□	CI/CD	i-08180f3d90bad6379	Stopped	t2.n

- Modify the IAM role by clicking on the button Update IAM role as shown in the figure.



Modify IAM role

Attach an IAM role to your instance.

Instance ID

i-0a6c00732cf72a44b (CI/CD)

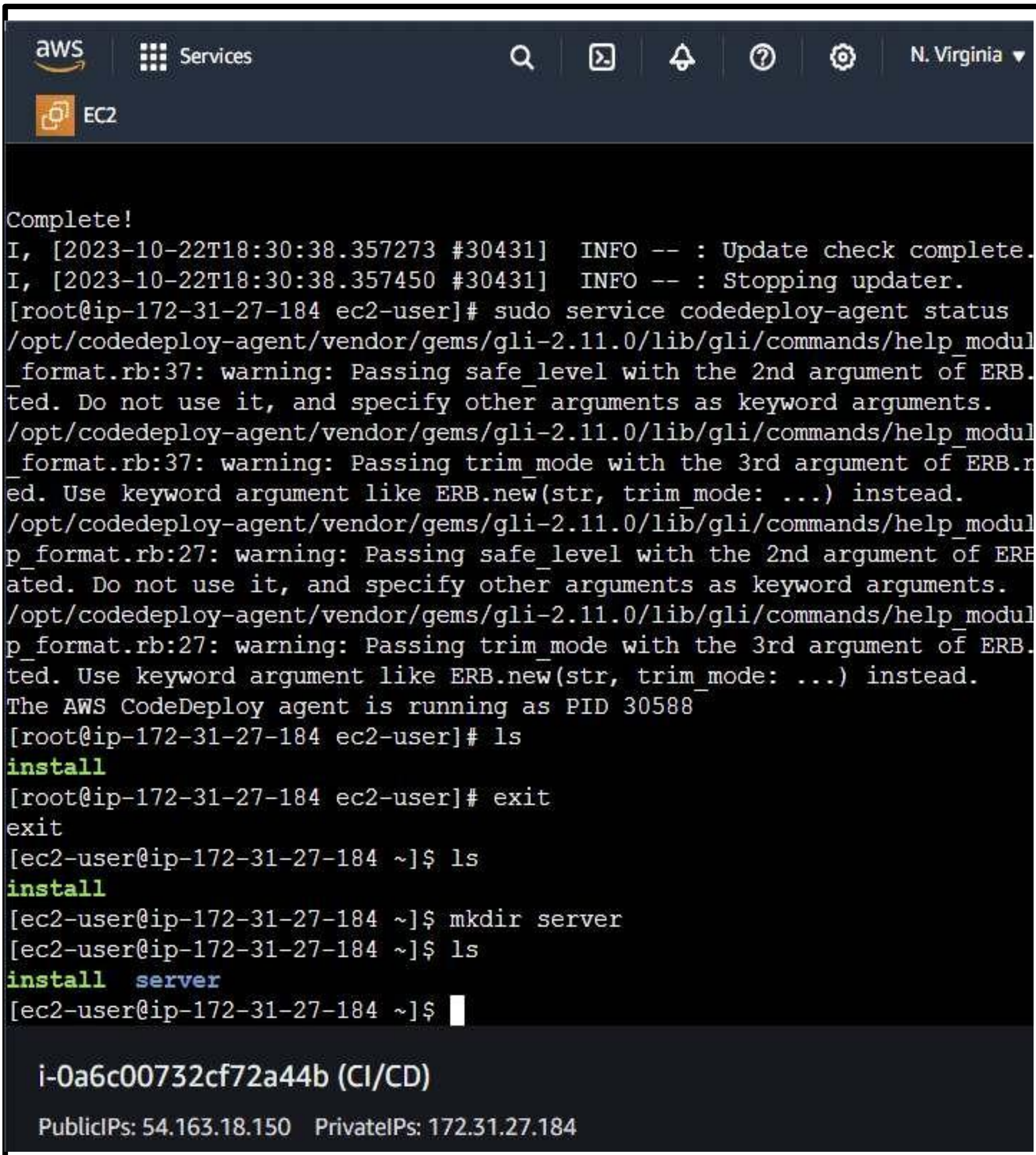
IAM role

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

EC2RoleAWSCodeDeploy-1

Step 12: Finalizing The Configuration

After this process, go to the console where your instance is connected and run the command "exit" to exit from the root folder and go back to the EC2 folder. Make a directory on the EC2 folder named "server", this is the directory where my source code will be deployed.



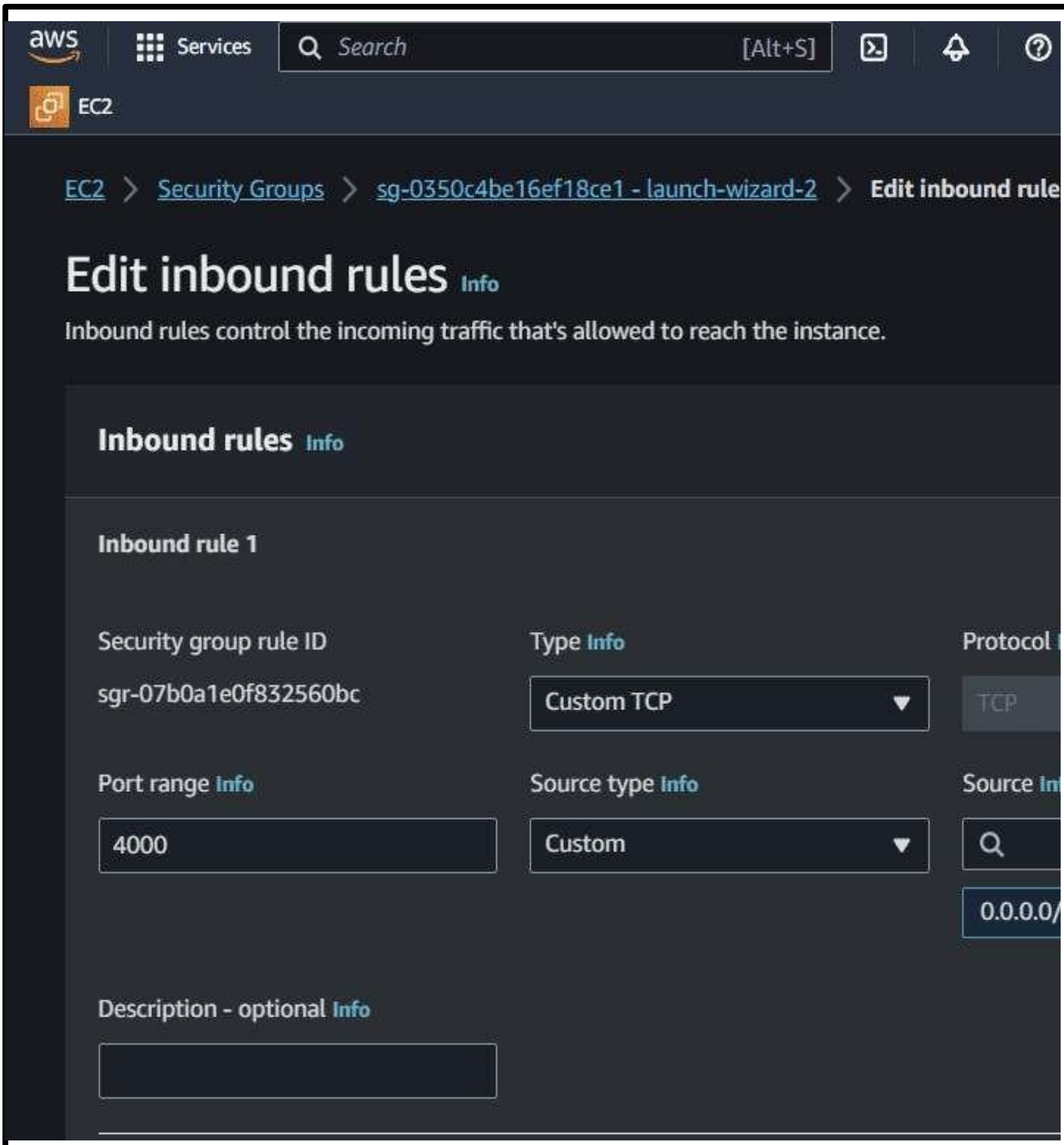
```
Complete!
I, [2023-10-22T18:30:38.357273 #30431]  INFO -- : Update check complete.
I, [2023-10-22T18:30:38.357450 #30431]  INFO -- : Stopping updater.
[root@ip-172-31-27-184 ec2-user]# sudo service codedeploy-agent status
/opt/codedeploy-agent/vendor/gems/gli-2.11.0/lib/gli/commands/help_modul
_format.rb:37: warning: Passing safe_level with the 2nd argument of ERB.
ted. Do not use it, and specify other arguments as keyword arguments.
/opt/codedeploy-agent/vendor/gems/gli-2.11.0/lib/gli/commands/help_modul
_format.rb:37: warning: Passing trim_mode with the 3rd argument of ERB.n
ed. Use keyword argument like ERB.new(str, trim_mode: ...) instead.
/opt/codedeploy-agent/vendor/gems/gli-2.11.0/lib/gli/commands/help_modul
p_format.rb:27: warning: Passing safe_level with the 2nd argument of ERB
ated. Do not use it, and specify other arguments as keyword arguments.
/opt/codedeploy-agent/vendor/gems/gli-2.11.0/lib/gli/commands/help_modul
p_format.rb:27: warning: Passing trim_mode with the 3rd argument of ERB.
ted. Use keyword argument like ERB.new(str, trim_mode: ...) instead.
The AWS CodeDeploy agent is running as PID 30588
[root@ip-172-31-27-184 ec2-user]# ls
install
[root@ip-172-31-27-184 ec2-user]# exit
exit
[ec2-user@ip-172-31-27-184 ~]$ ls
install
[ec2-user@ip-172-31-27-184 ~]$ mkdir server
[ec2-user@ip-172-31-27-184 ~]$ ls
install  server
[ec2-user@ip-172-31-27-184 ~]$ 
```

i-0a6c00732cf72a44b (CI/CD)

PublicIPs: 54.163.18.150   PrivateIPs: 172.31.27.184

- **Then after doing the above process, come back to the running instances list.**
- **Select your currently created running instance and go to the "Security" section present at the end of the page.**
- **Click on the link present under the "Security Groups". After redirecting to the required page, click on "Edit Inbound rules" under the section of "Inbound rules" present at the end of the page.**

- Then add a rule, select a port range of your choice and select the source as "Anywhere-IPv4" from the dropdown menu and then click on "Save rules".
- Basically, let me give you a overview what we are actually doing here. In brief, when you add an inbound rule to a security group for an instance with port range (in my case, it was 4000) and set the source to "Anywhere-IPv4," you are allowing any computer or device on the internet to connect to your instance through port 4000.
- This is like opening a door (port 4000) on your server and letting anyone from anywhere access the service or application running on that port.



EC2 > Security Groups > sg-0350c4be16ef18ce1 - launch-wizard-2 > Edit inbound rule

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Inbound rule 1

Security group rule ID: sgr-07b0a1e0f832560bc

Type Info: Custom TCP

Protocol: TCP

Port range Info: 4000

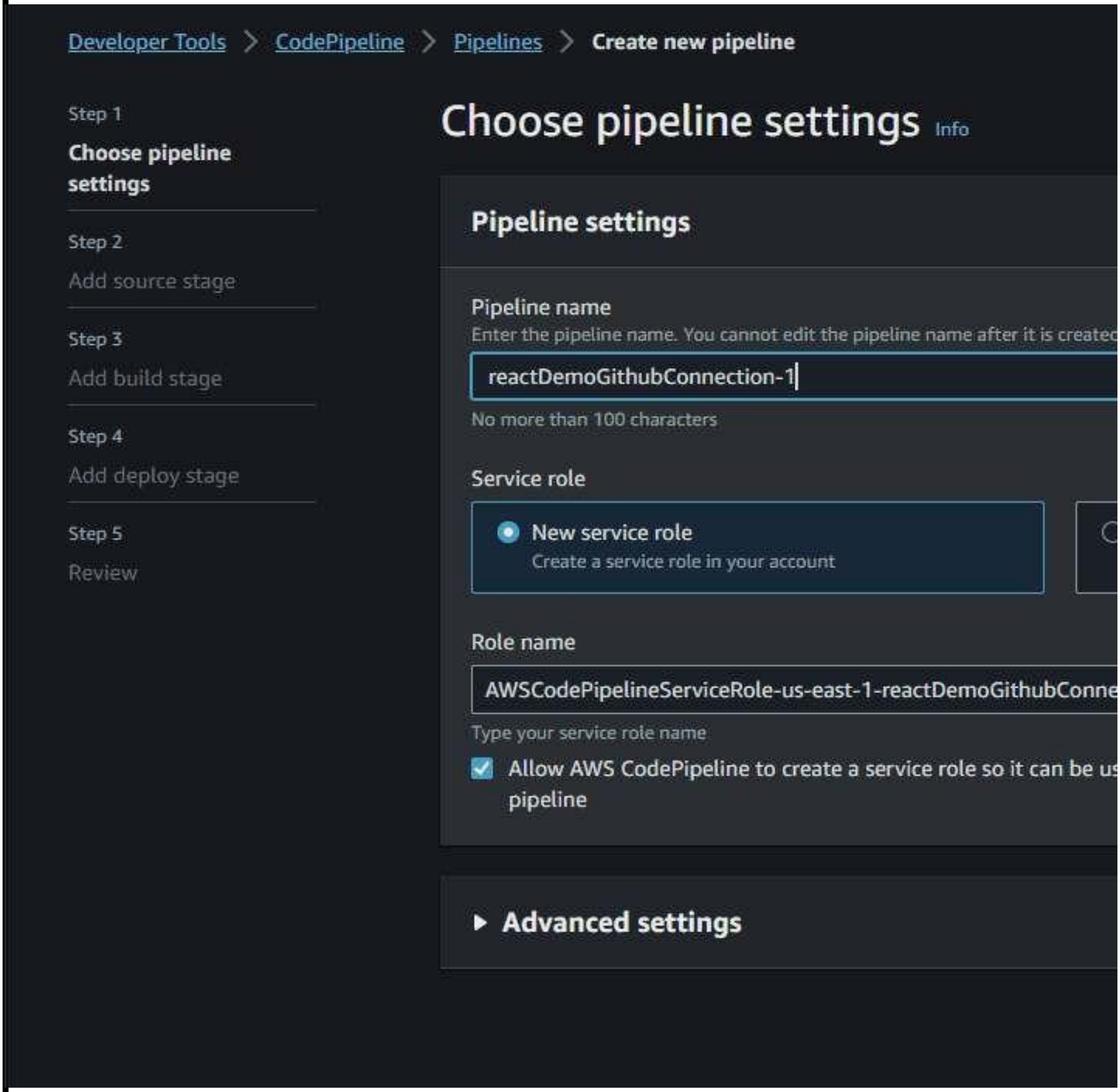
Source type Info: Custom

Source: 0.0.0.0/

Description - optional Info

Step 13: Create A New Pipeline

- Create a CodePipeline using Github, CodeBuild and CodeDeploy
- Firstly Create CodePipeline navigate to CodePipeline via AWS Management Console and click on Create pipeline.



Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1
Choose pipeline settings

Step 2
Add source stage

Step 3
Add build stage

Step 4
Add deploy stage

Step 5
Review

Choose pipeline settings Info

Pipeline settings

Pipeline name
Enter the pipeline name. You cannot edit the pipeline name after it is createc

reactDemoGithubConnection-1

No more than 100 characters

Service role

● New service role
Create a service role in your account

Role name

AWSCodePipelineServiceRole-us-east-1-reactDemoGithubConne

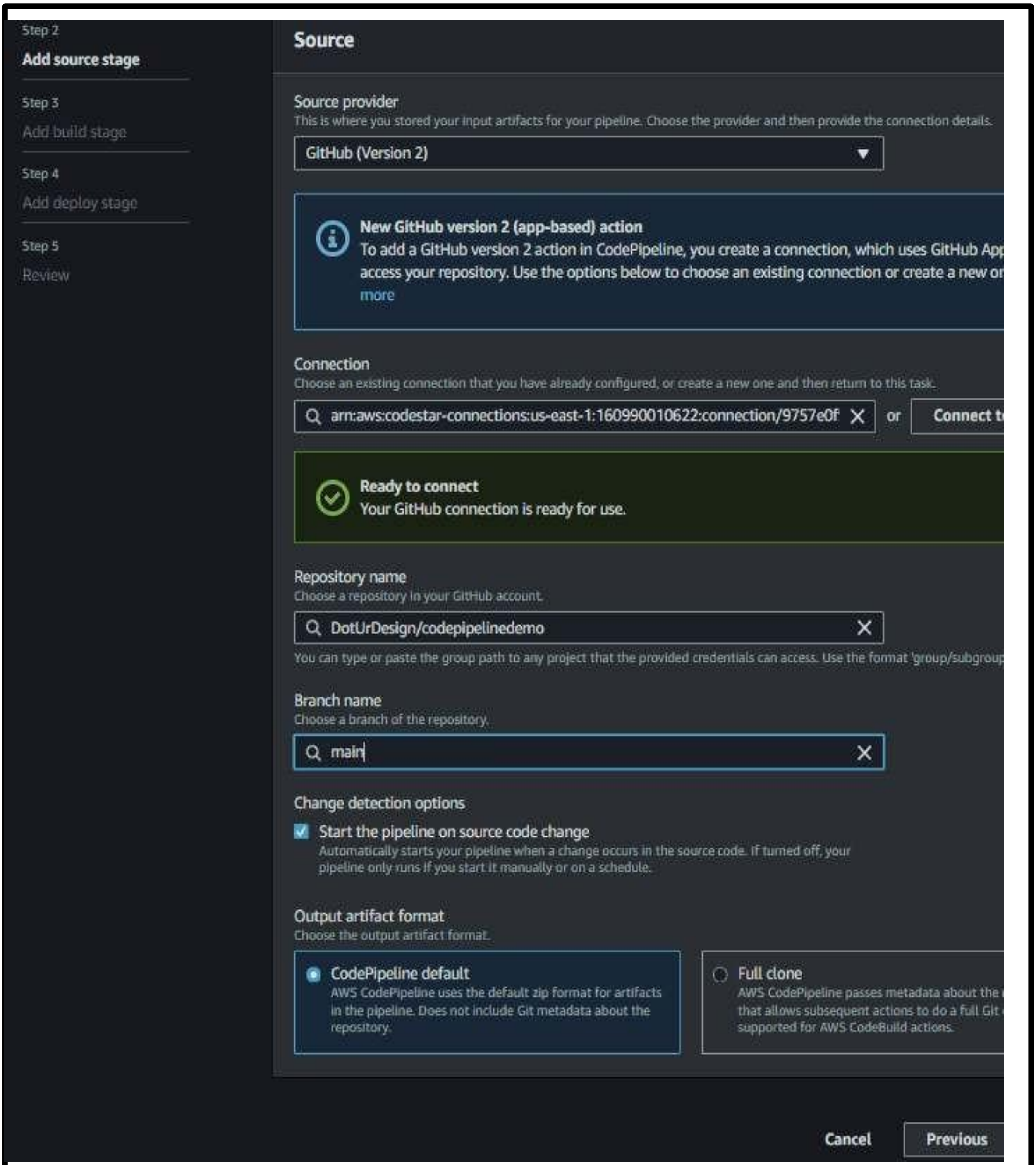
Type your service role name

☑ Allow AWS CodePipeline to create a service role so it can be us pipeline

▸ Advanced settings

Step 14: Choose Github In Code Source

- After selecting GitHub as the source provider, click on the Connect to GitHub button. You’ll then be prompt to enter your GitHub login credentials.
- Once you grant AWS CodePipeline access to your GitHub repository, you can select a repository and branch for CodePipeline to upload commits to this repository to your pipeline.



Step 2
Add source stage

Step 3
Add build stage

Step 4
Add deploy stage

Step 5
Review

Source

Source provider
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.
GitHub (Version 2) ▼

New GitHub version 2 (app-based) action
To add a GitHub version 2 action in CodePipeline, you create a connection, which uses GitHub Ap
access your repository. Use the options below to choose an existing connection or create a new o
more

Connection
Choose an existing connection that you have already configured, or create a new one and then return to this task.
arn:aws:codestar-connections:us-east-1:160990010622:connection/9757e0f ✕ or Connect t

Ready to connect
Your GitHub connection is ready for use.

Repository name
Choose a repository in your GitHub account.
DotUrDesign/codepipelinedemo ✕
You can type or paste the group path to any project that the provided credentials can access. Use the format 'group/subgrou

Branch name
Choose a branch of the repository.
main ✕

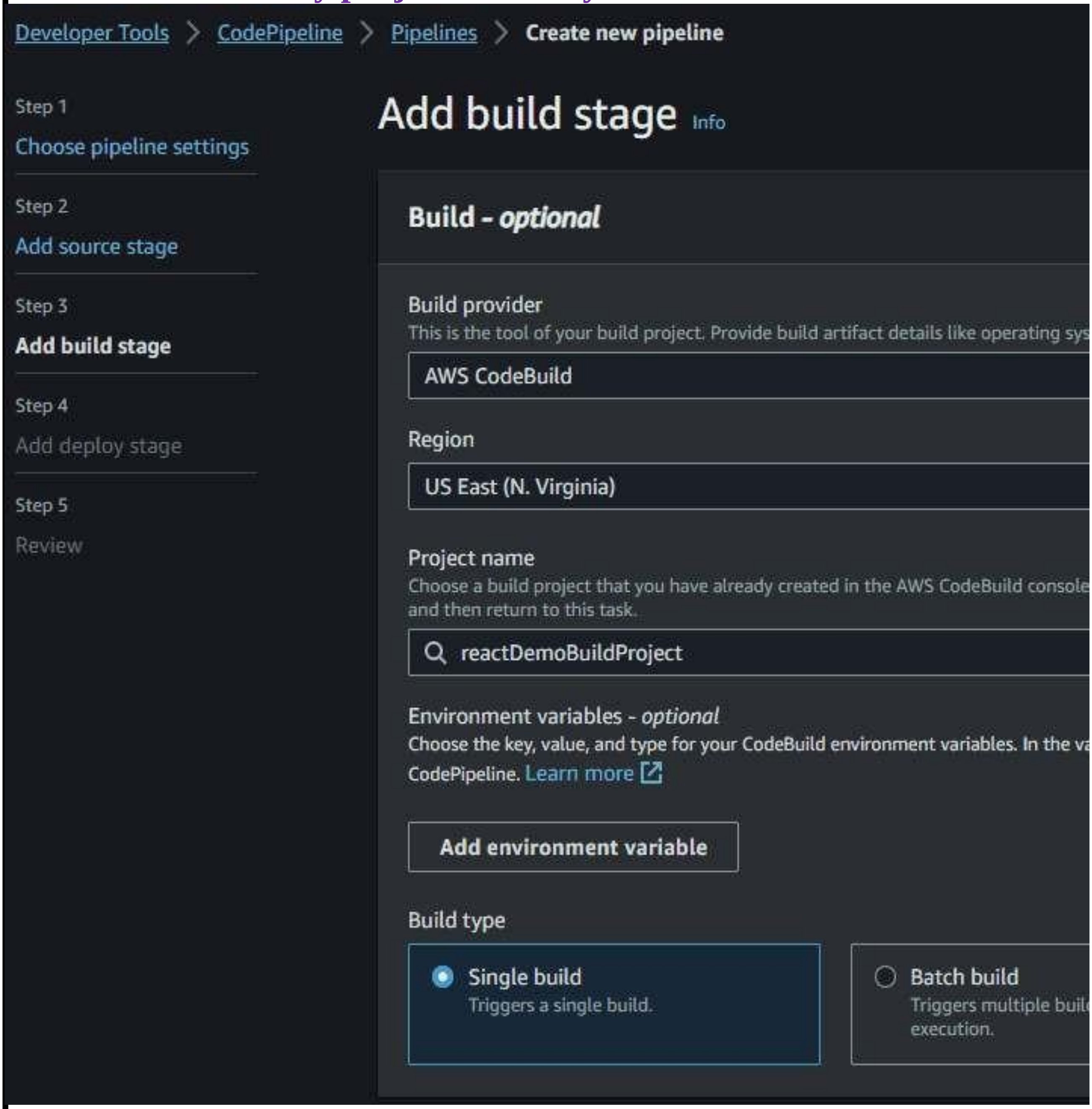
Change detection options
☑ Start the pipeline on source code change
Automatically starts your pipeline when a change occurs in the source code. If turned off, your pipeline only runs if you start it manually or on a schedule.

Output artifact format
Choose the output artifact format.
● CodePipeline default
AWS CodePipeline uses the default zip format for artifacts in the pipeline. Does not include Git metadata about the repository.
○ Full clone
AWS CodePipeline passes metadata about the
that allows subsequent actions to do a full Git
supported for AWS CodeBuild actions.

Step 15: Configure CodeBuild (Optional)

- **If you haven’t created a project prior to creating your pipeline, then you can create a project directly from here by clicking Create project button.**
- **Note: Buildspec file is a collection of build commands and related settings, in YAML format, that CodeBuild uses to run a**

build. For my project, I created a buildspec.yaml file and added it in the root of my project directory.



Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1
Choose pipeline settings

Step 2
Add source stage

Step 3
Add build stage

Step 4
Add deploy stage

Step 5
Review

Add build stage Info

Build - *optional*

Build provider
This is the tool of your build project. Provide build artifact details like operating sys

AWS CodeBuild

Region

US East (N. Virginia)

Project name
Choose a build project that you have already created in the AWS CodeBuild console and then return to this task.

reactDemoBuildProject

Environment variables - *optional*
Choose the key, value, and type for your CodeBuild environment variables. In the va CodePipeline. Learn more

Build type

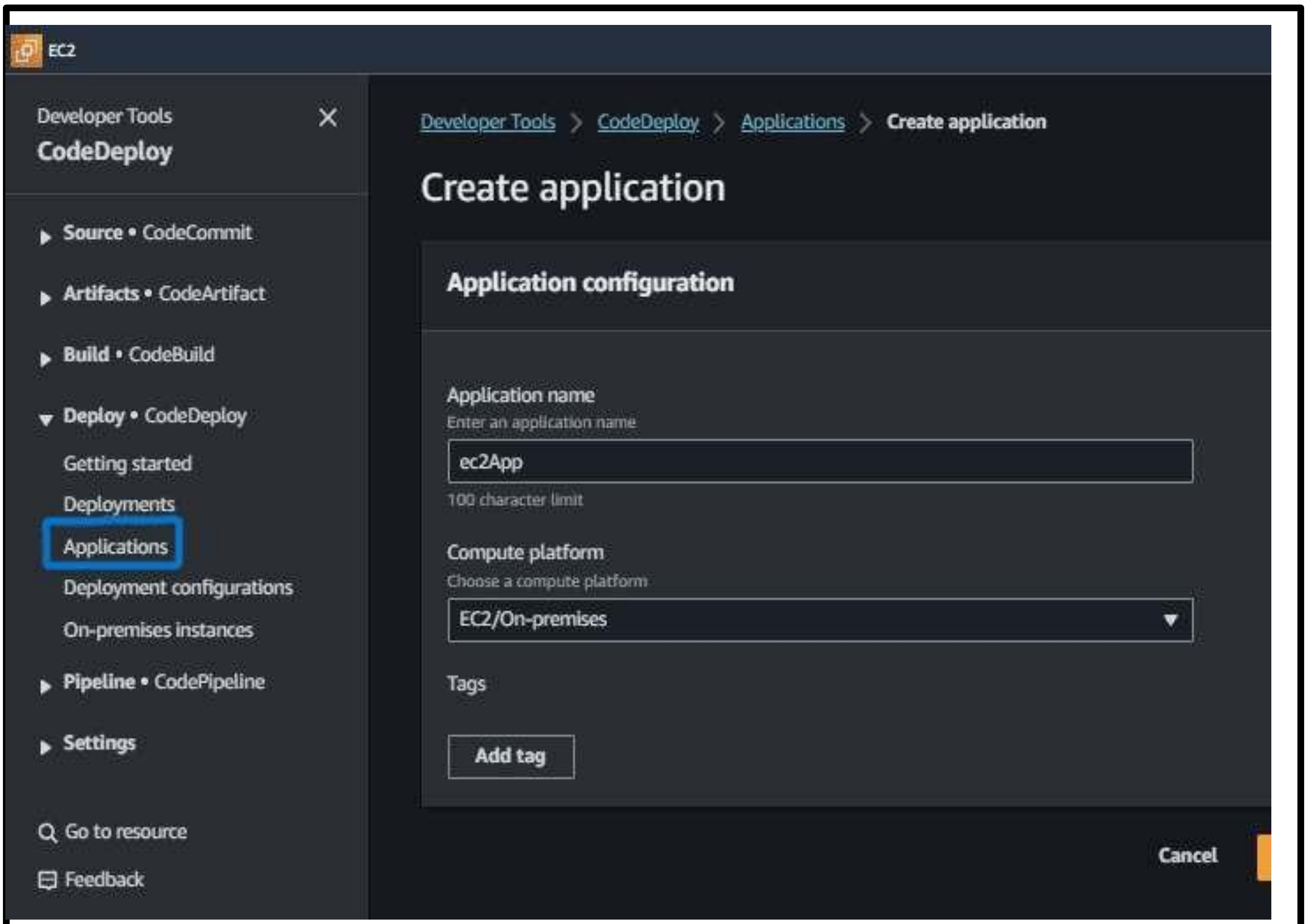
● Single build
Triggers a single build.

○ Batch build
Triggers multiple buil execution.

Step 16: Add Deploy Stage

Note : Before going to configure Add Deploy Stage, Let's make duplicate tab of current tab.

- **Go to code deploy in the navigation, Select Application, then add create a deployment group.**



Create application

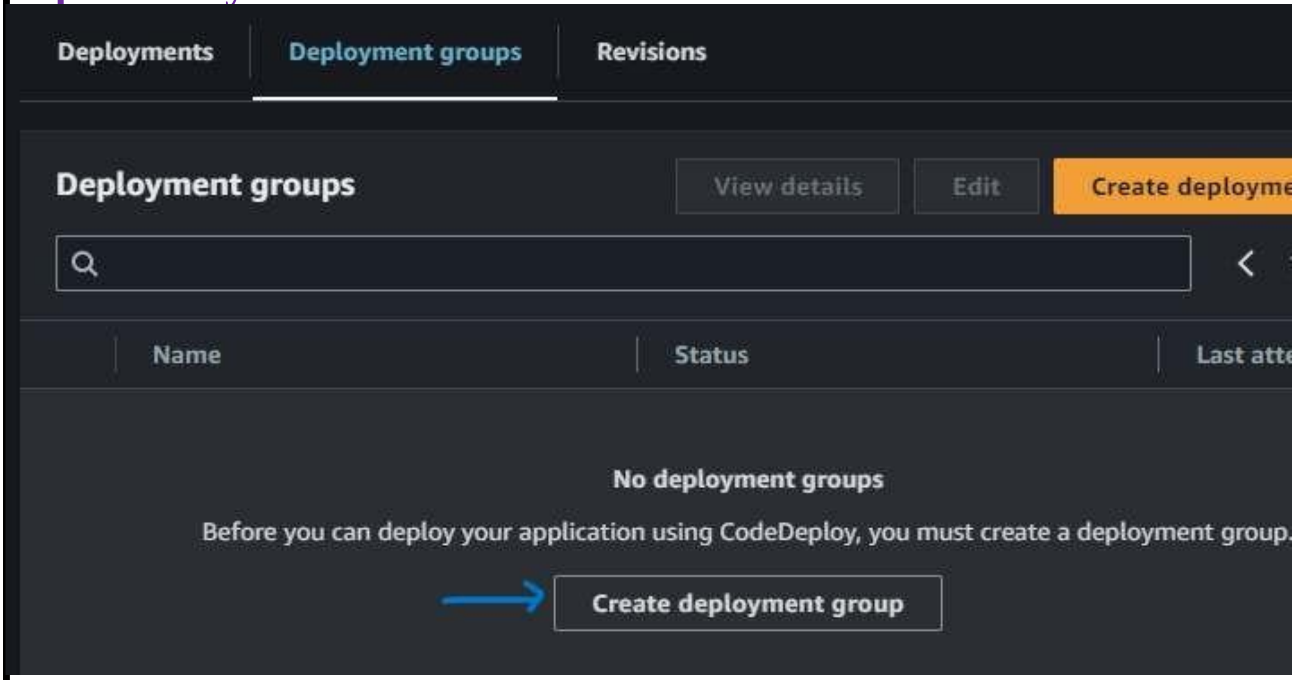
Application configuration

Application name
Enter an application name
ec2App
100 character limit

Compute platform
Choose a compute platform
EC2/On-premises

Tags

- **Create a deployment Group by clicking on the button "Create deployment group", the following screenshot illustrates with practically.**



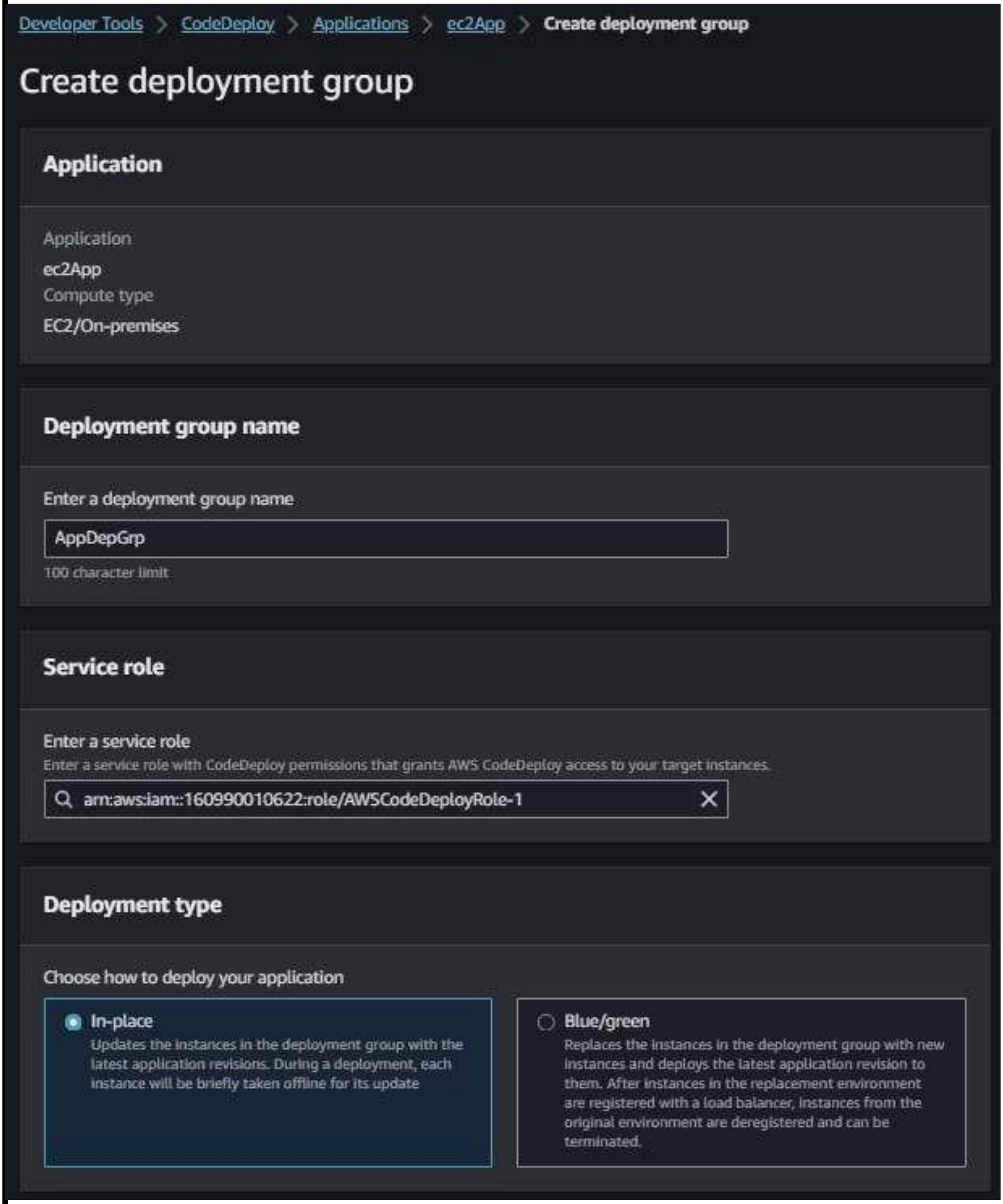
Deployment groups

Name	Status	Last atte

No deployment groups

Before you can deploy your application using CodeDeploy, you must create a deployment group.

- **In deployment group Select EC2 instances and select Tag and Value**



Developer Tools > CodeDeploy > Applications > ec2App > Create deployment group

Create deployment group

Application

Application
ec2App
Compute type
EC2/On-premises

Deployment group name

Enter a deployment group name
AppDepGrp
100 character limit

Service role

Enter a service role
Enter a service role with CodeDeploy permissions that grants AWS CodeDeploy access to your target instances.
arn:aws:iam::160990010622:role/AWSCodeDeployRole-1

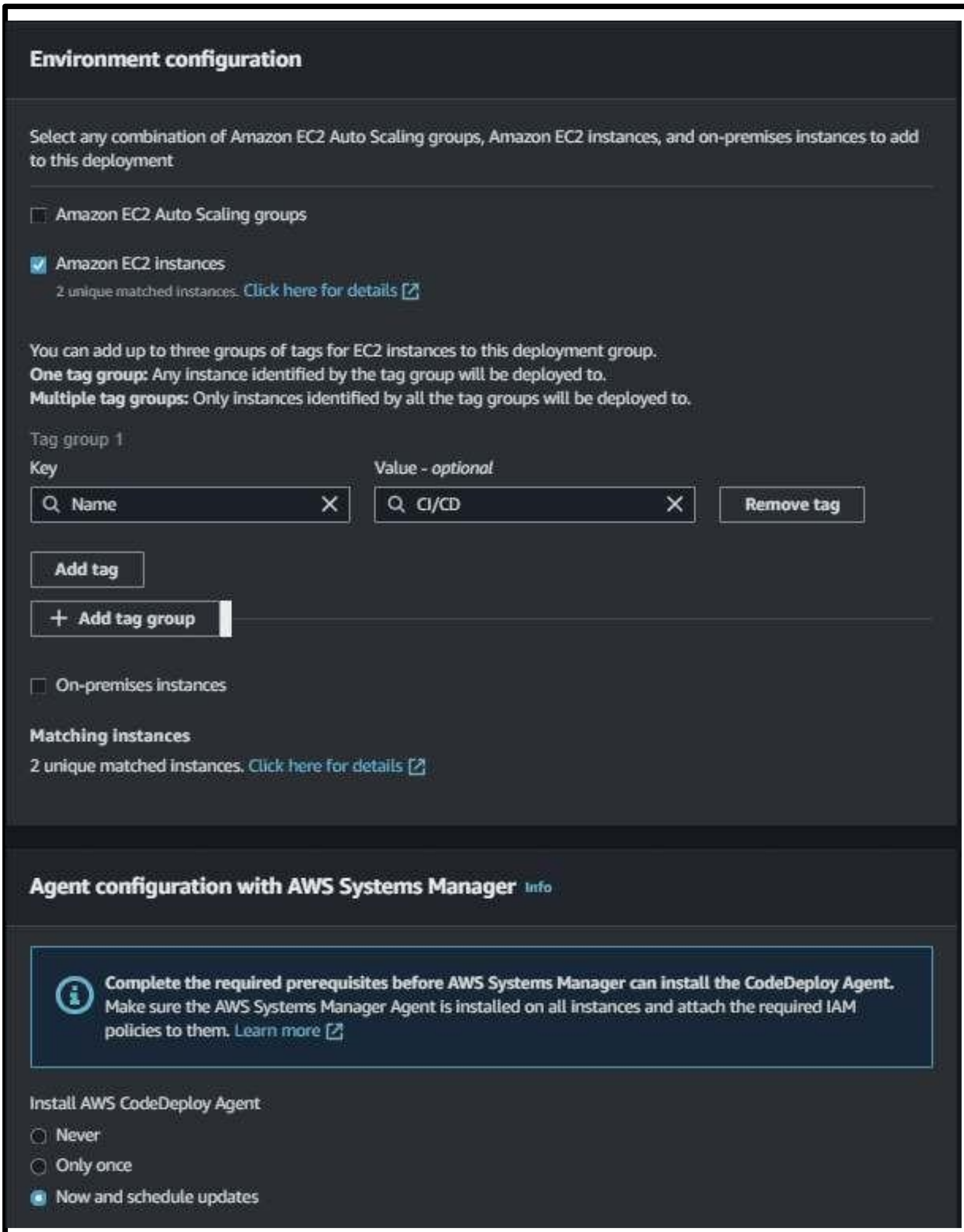
Deployment type

Choose how to deploy your application

● In-place
Updates the instances in the deployment group with the latest application revisions. During a deployment, each instance will be briefly taken offline for its update

○ Blue/green
Replaces the instances in the deployment group with new instances and deploys the latest application revision to them. After instances in the replacement environment are registered with a load balancer, instances from the original environment are deregistered and can be terminated.

- **Provide the Environment configurations such as select the Amazon EC2 Instances and provide the key and values to it.**



Environment configuration

Select any combination of Amazon EC2 Auto Scaling groups, Amazon EC2 instances, and on-premises instances to add to this deployment

□ Amazon EC2 Auto Scaling groups

☑ Amazon EC2 instances
2 unique matched instances. Click here for details

You can add up to three groups of tags for EC2 instances to this deployment group.
One tag group: Any instance identified by the tag group will be deployed to.
Multiple tag groups: Only instances identified by all the tag groups will be deployed to.

Tag group 1

Key: Name
Value - *optional*: CI/CD

□ On-premises instances

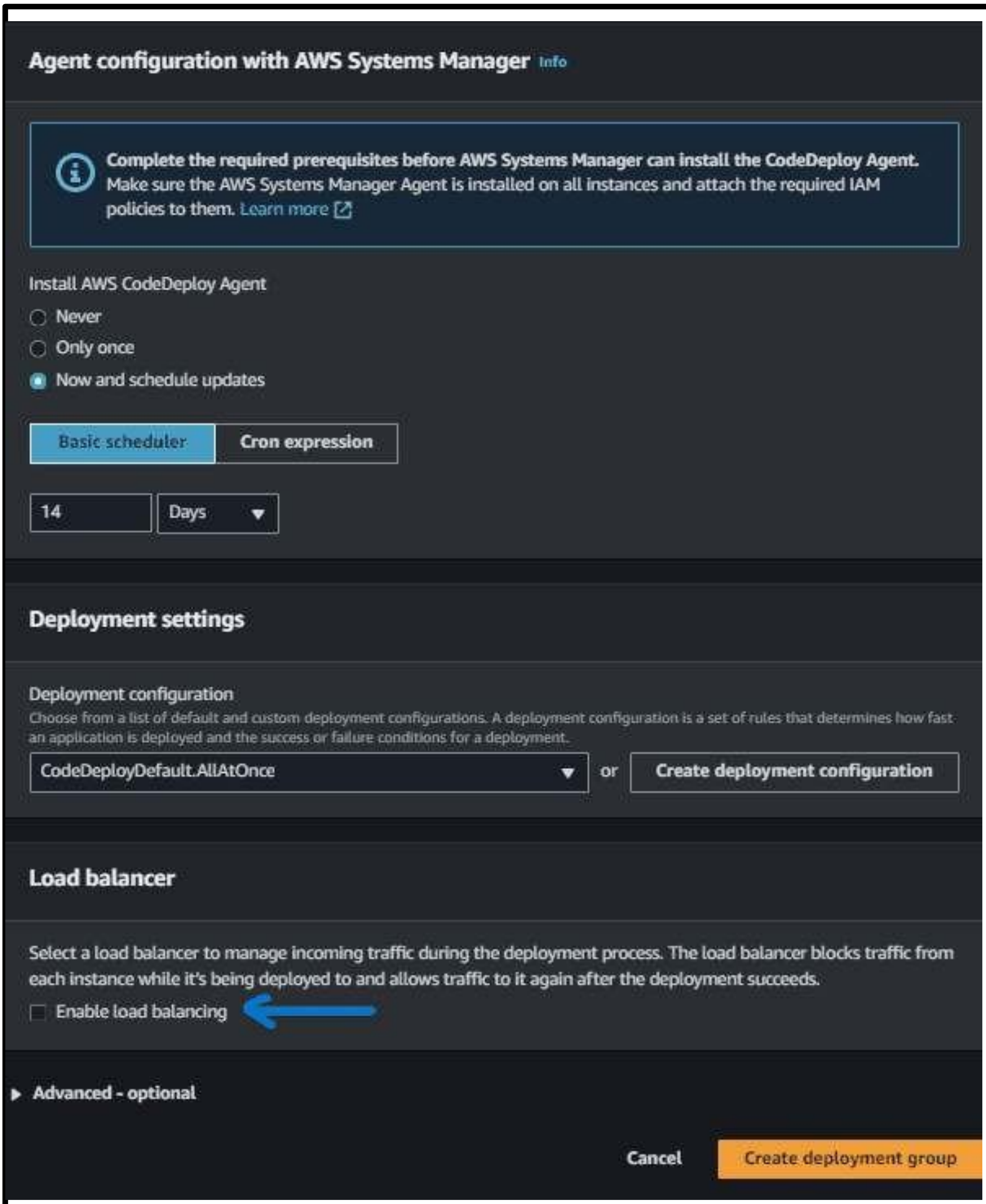
Matching instances
2 unique matched instances. Click here for details

Agent configuration with AWS Systems Manager

Complete the required prerequisites before AWS Systems Manager can install the CodeDeploy Agent. Make sure the AWS Systems Manager Agent is installed on all instances and attach the required IAM policies to them. Learn more

Install AWS CodeDeploy Agent
○ Never
○ Only once
● Now and schedule updates

- **Uncheck Load Balancer Option**



Agent configuration with AWS Systems Manager Info

Complete the required prerequisites before AWS Systems Manager can install the CodeDeploy Agent. Make sure the AWS Systems Manager Agent is installed on all instances and attach the required IAM policies to them. Learn more

Install AWS CodeDeploy Agent

○ Never

○ Only once

● Now and schedule updates

Basic scheduler | Cron expression

14 Days

Deployment settings

Deployment configuration

Choose from a list of default and custom deployment configurations. A deployment configuration is a set of rules that determines how fast an application is deployed and the success or failure conditions for a deployment.

CodeDeployDefault.AllAtOnce or Create deployment configuration

Load balancer

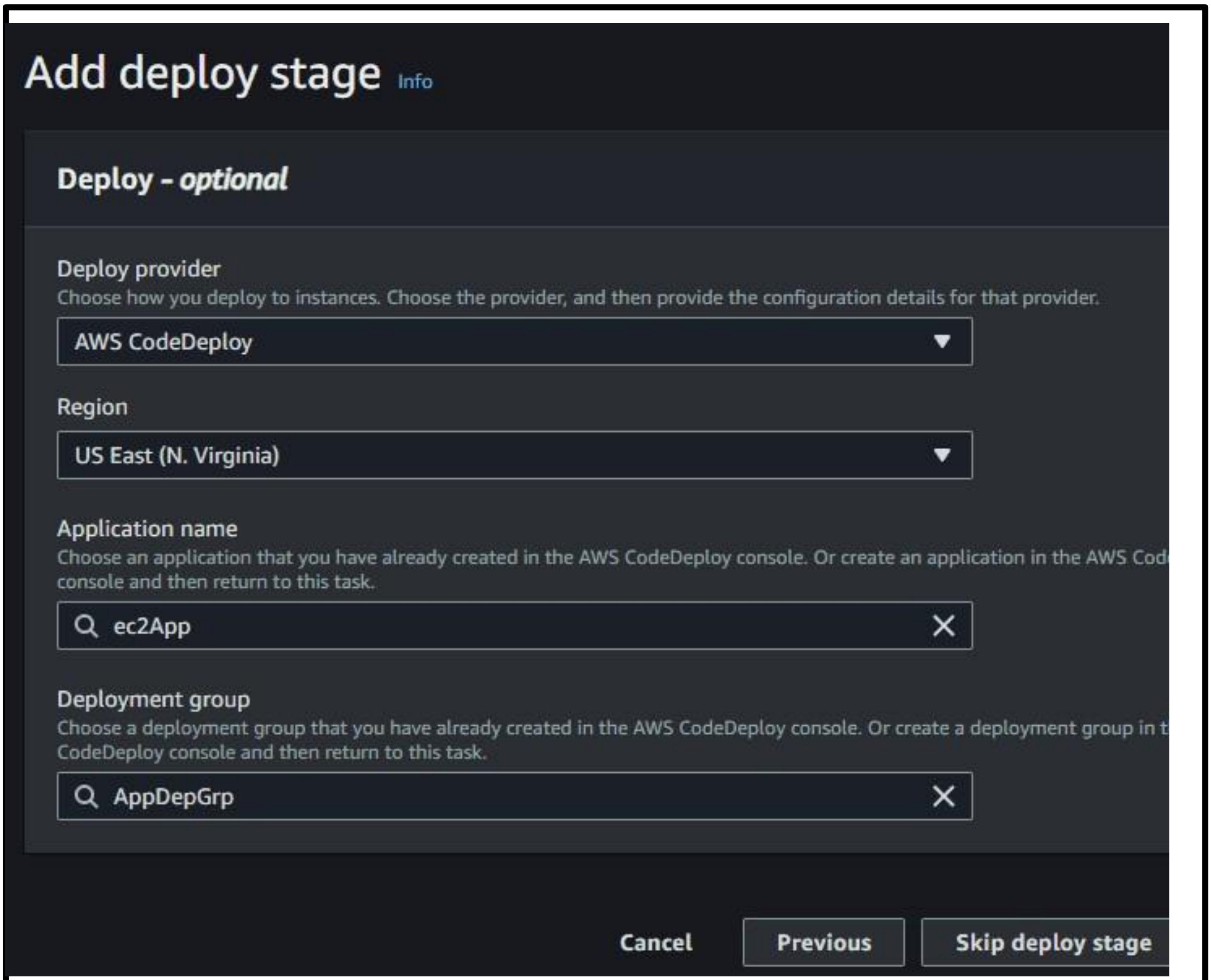
Select a load balancer to manage incoming traffic during the deployment process. The load balancer blocks traffic from each instance while it's being deployed to and allows traffic to it again after the deployment succeeds.

□ Enable load balancing

▸ Advanced - optional

Cancel Create deployment group

- **Finally Come on Add Deploy Stage and select that created Application name & Deployment group**



Add deploy stage Info

Deploy - *optional*

Deploy provider
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.
AWS CodeDeploy ▼

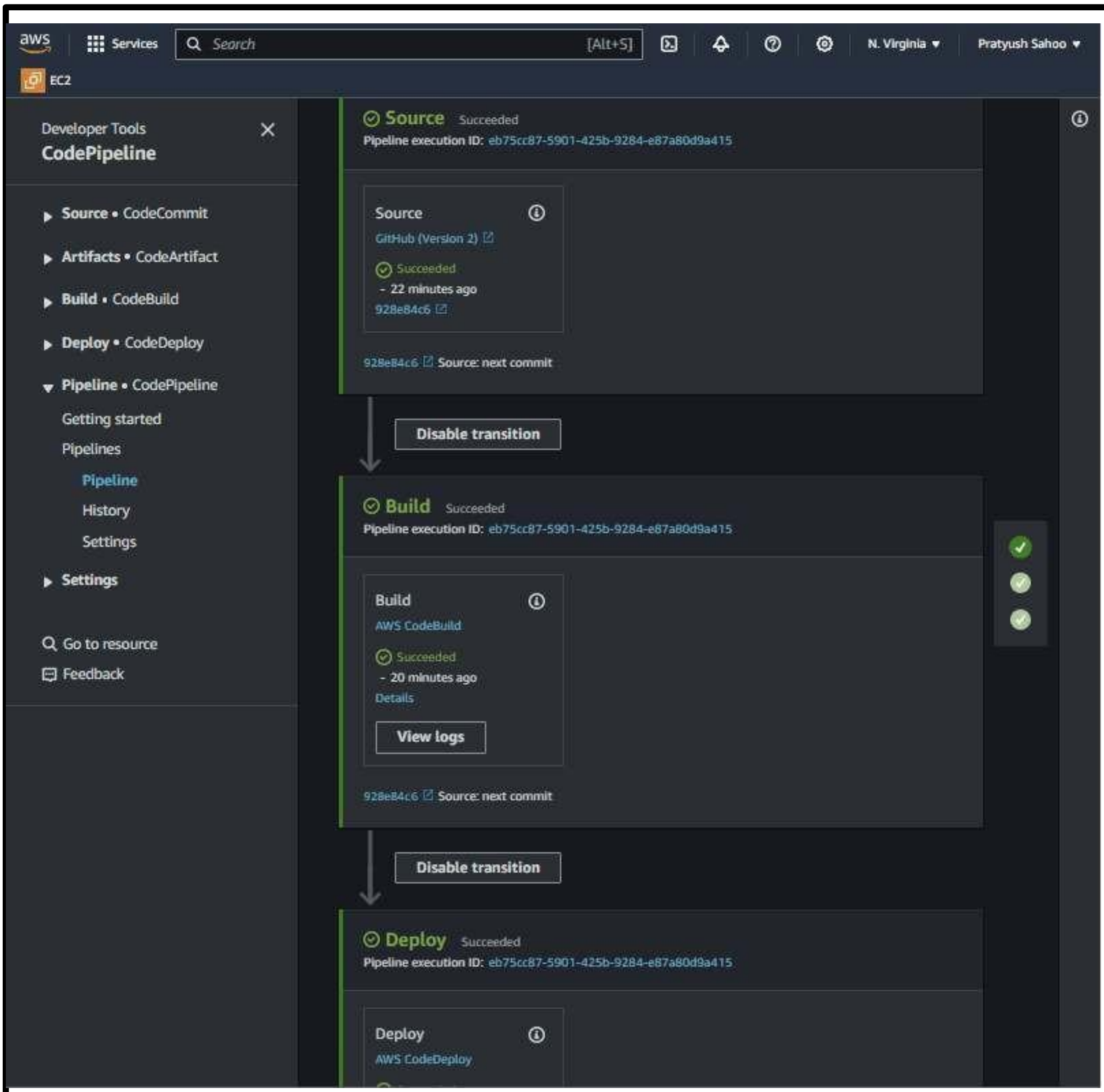
Region
US East (N. Virginia) ▼

Application name
Choose an application that you have already created in the AWS CodeDeploy console. Or create an application in the AWS Cod
console and then return to this task.
ec2App ✕

Deployment group
Choose a deployment group that you have already created in the AWS CodeDeploy console. Or create a deployment group in t
CodeDeploy console and then return to this task.
AppDepGrp ✕

Step 17: Review And Create

- **As a final step review and create it. By creating this we have successful the created a CI/CD pipeline in AWS.**



Source Succeeded
Pipeline execution ID: eb75cc87-5901-425b-9284-e87a80d9a415

Source
GitHub (Version 2)
Succeeded
- 22 minutes ago
928e84c6

928e84c6 Source: next commit

Build Succeeded
Pipeline execution ID: eb75cc87-5901-425b-9284-e87a80d9a415

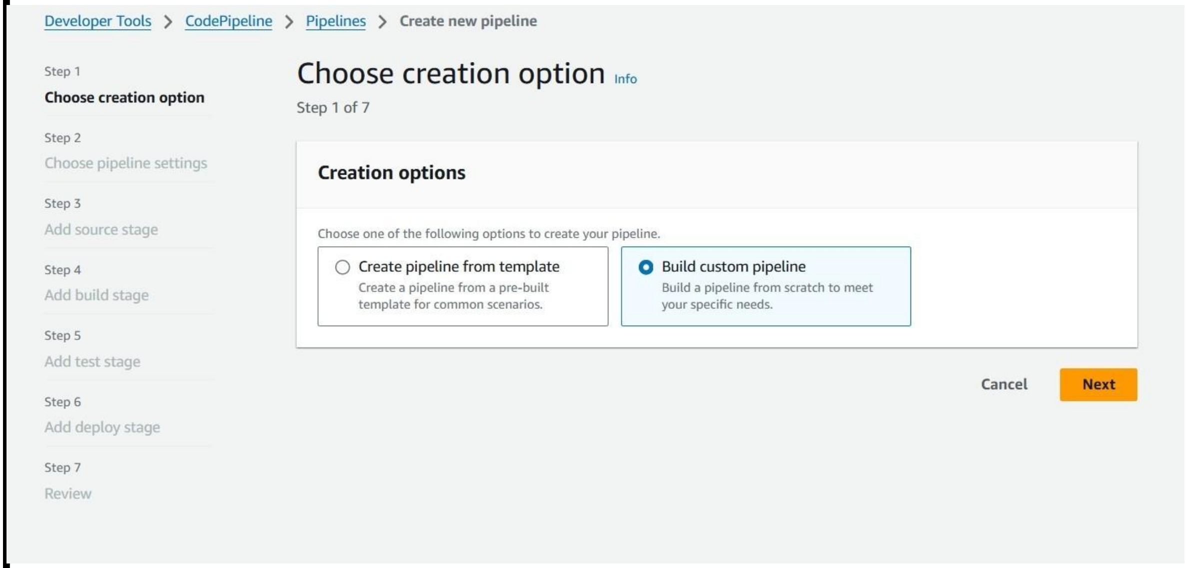
Build
AWS CodeBuild
Succeeded
- 20 minutes ago
Details

928e84c6 Source: next commit

Deploy Succeeded
Pipeline execution ID: eb75cc87-5901-425b-9284-e87a80d9a415

Deploy
AWS CodeDeploy

PIPELINE CREATED STAGE ALTERNATE METHOD



Choose creation option Info

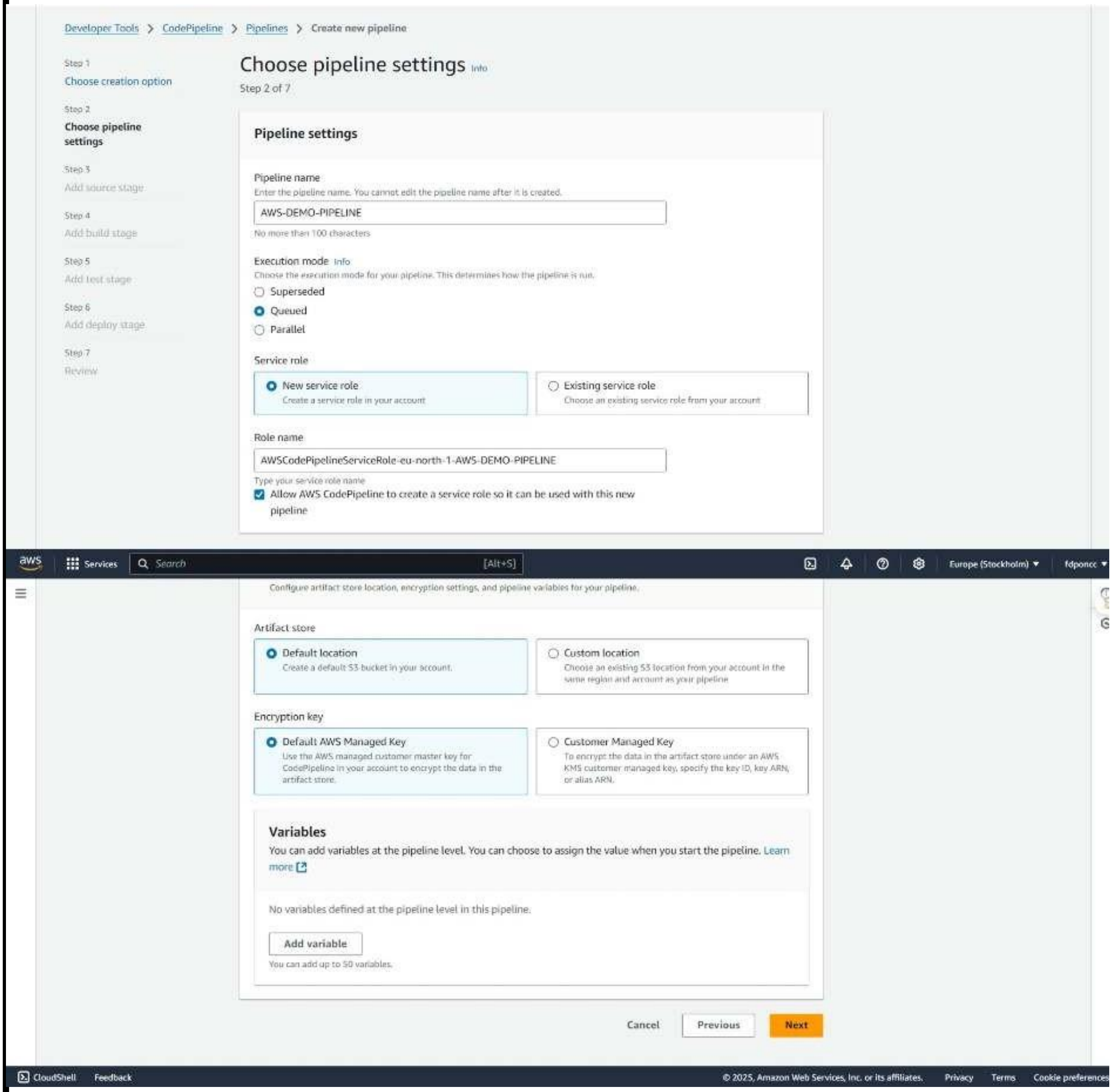
Step 1 of 7

Creation options

Choose one of the following options to create your pipeline.

○ Create pipeline from template
Create a pipeline from a pre-built template for common scenarios.

● Build custom pipeline
Build a pipeline from scratch to meet your specific needs.



Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1
Choose creation option

Step 2
Choose pipeline settings

Step 3
Add source stage

Step 4
Add build stage

Step 5
Add test stage

Step 6
Add deploy stage

Step 7
Review

Choose pipeline settings Info

Step 2 of 7

Pipeline settings

Pipeline name
Enter the pipeline name. You cannot edit the pipeline name after it is created.
AWS-DEMO-PIPELINE
No more than 100 characters

Execution mode Info
Choose the execution mode for your pipeline. This determines how the pipeline is run.
○ Superseded
● Queued
○ Parallel

Service role
● New service role
Create a service role in your account
○ Existing service role
Choose an existing service role from your account

Role name
AWSCodePipelineServiceRole-eu-north-1-AWS-DEMO-PIPELINE
Type your service role name
☑ Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

Configure artifact store location, encryption settings, and pipeline variables for your pipeline.

Artifact store
● Default location
Create a default S3 bucket in your account.
○ Custom location
Choose an existing S3 location from your account in the same region and account as your pipeline

Encryption key
● Default AWS Managed Key
Use the AWS managed customer master key for CodePipeline in your account to encrypt the data in the artifact store.
○ Customer Managed Key
To encrypt the data in the artifact store under an AWS KMS customer managed key, specify the key ID, key ARN, or alias ARN.

Variables

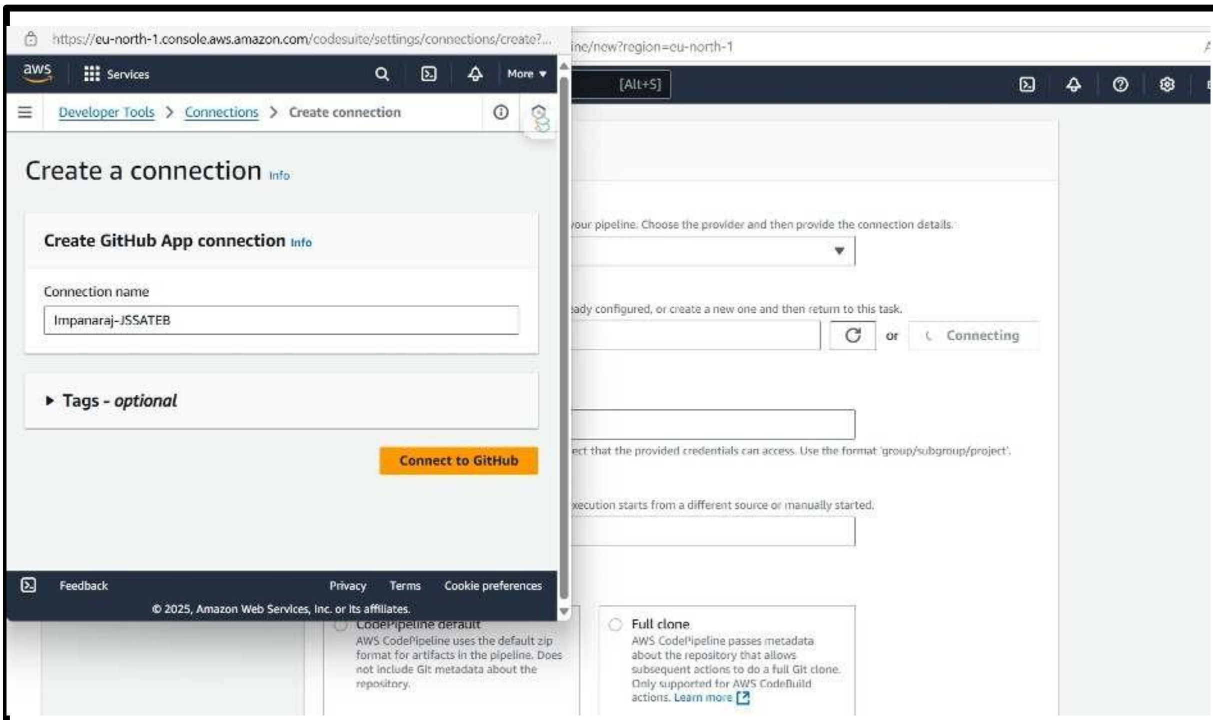
You can add variables at the pipeline level. You can choose to assign the value when you start the pipeline. Learn more

No variables defined at the pipeline level in this pipeline.

Add variable
You can add up to 50 variables.

NEXT

CONNECT TO GITHUB

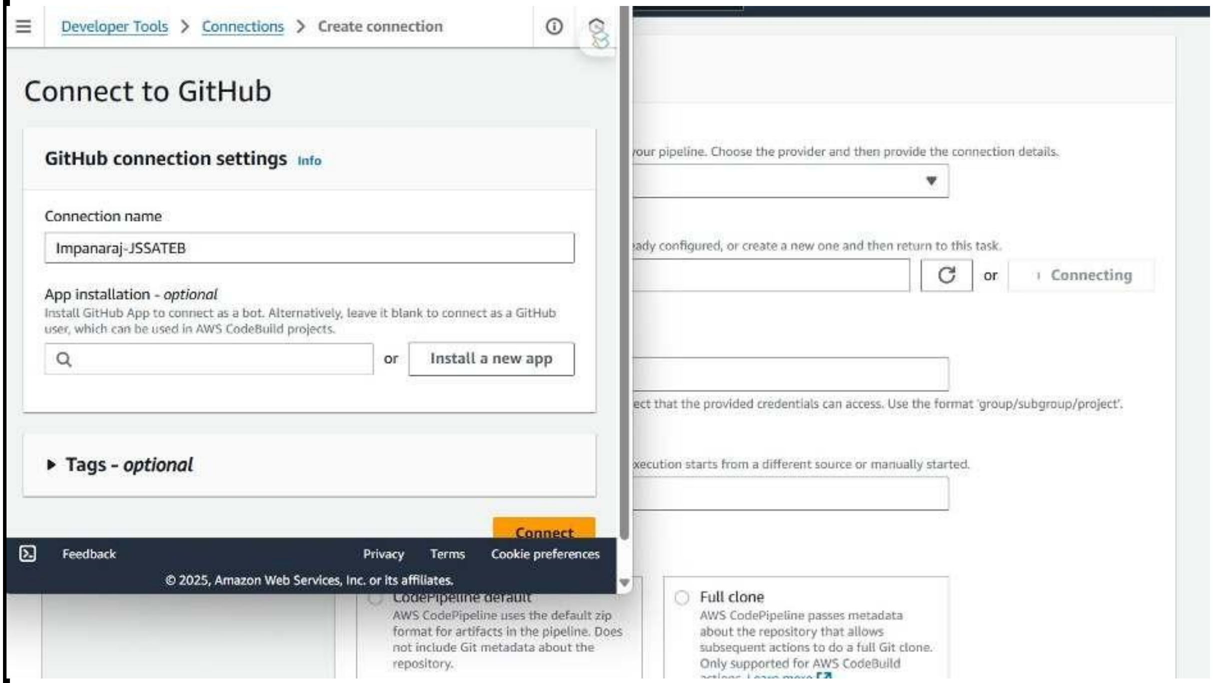


Create a connection

Create GitHub App connection

Connection name: Impanaraj-JSSATEB

CONNECT TO GITHUB



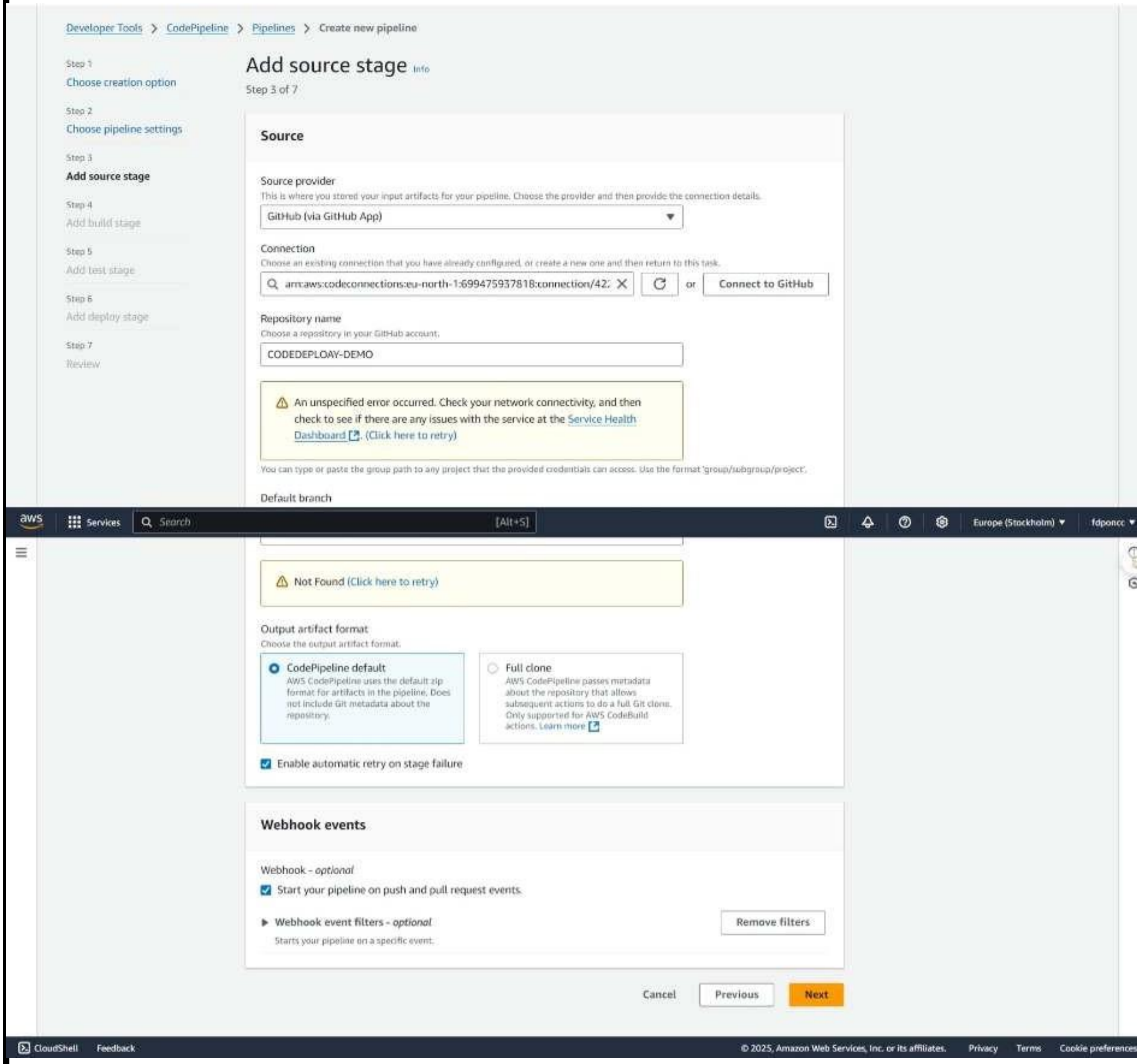
Connect to GitHub

GitHub connection settings

Connection name: Impanaraj-JSSATEB

App installation - optional
Install GitHub App to connect as a bot. Alternatively, leave it blank to connect as a GitHub user, which can be used in AWS CodeBuild projects.

CONNECT



Add source stage

Step 3 of 7

Source

Source provider

This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.

GitHub (via GitHub App)

Connection

Choose an existing connection that you have already configured, or create a new one and then return to this task.

arn:aws:codeconnections:eu-north-1:699475937818:connection/42:

Repository name

Choose a repository in your GitHub account.

CODEDEPLOAY-DEMO

⚠ An unspecified error occurred. Check your network connectivity, and then check to see if there are any issues with the service at the Service Health Dashboard. (Click here to retry)

You can type or paste the group path to any project that the provided credentials can access. Use the format 'group/subgroup/project'.

Default branch

⚠ Not Found (Click here to retry)

Output artifact format

Choose the output artifact format.

● CodePipeline default
AWS CodePipeline uses the default zip format for artifacts in the pipeline. Does not include Git metadata about the repository.

○ Full clone
AWS CodePipeline passes metadata about the repository that allows subsequent actions to do a full Git clone. Only supported for AWS CodeBuild actions.

☑ Enable automatic retry on stage failure

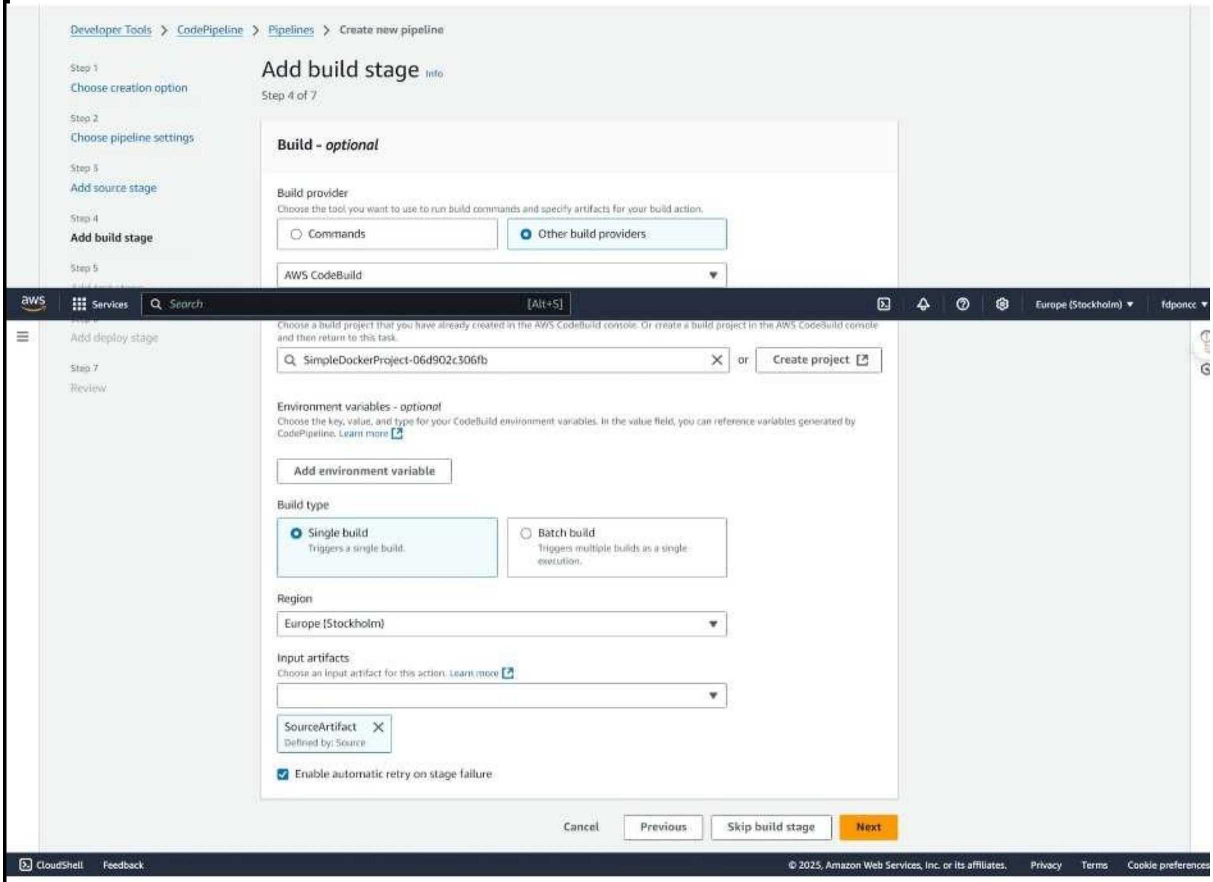
Webhook events

Webhook - *optional*

☑ Start your pipeline on push and pull request events.

▸ Webhook event filters - *optional*

Starts your pipeline on a specific event.



Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1 Choose creation option
Step 2 Choose pipeline settings
Step 3 Add source stage
Step 4 **Add build stage**
Step 5
Step 6 Add deploy stage
Step 7 Review

Add build stage Info

Step 4 of 7

Build - *optional*

Build provider
Choose the tool you want to use to run build commands and specify artifacts for your build action.

○ Commands ● Other build providers

AWS CodeBuild

Choose a build project that you have already created in the AWS CodeBuild console. Or create a build project in the AWS CodeBuild console and then return to this task.

SimpleDockerProject-06d902c306fb or Create project

Environment variables - *optional*
Choose the key, value, and type for your CodeBuild environment variables. In the value field, you can reference variables generated by CodePipeline. Learn more

Add environment variable

Build type
● Single build — Triggers a single build.
○ Batch build — Triggers multiple builds as a single execution.

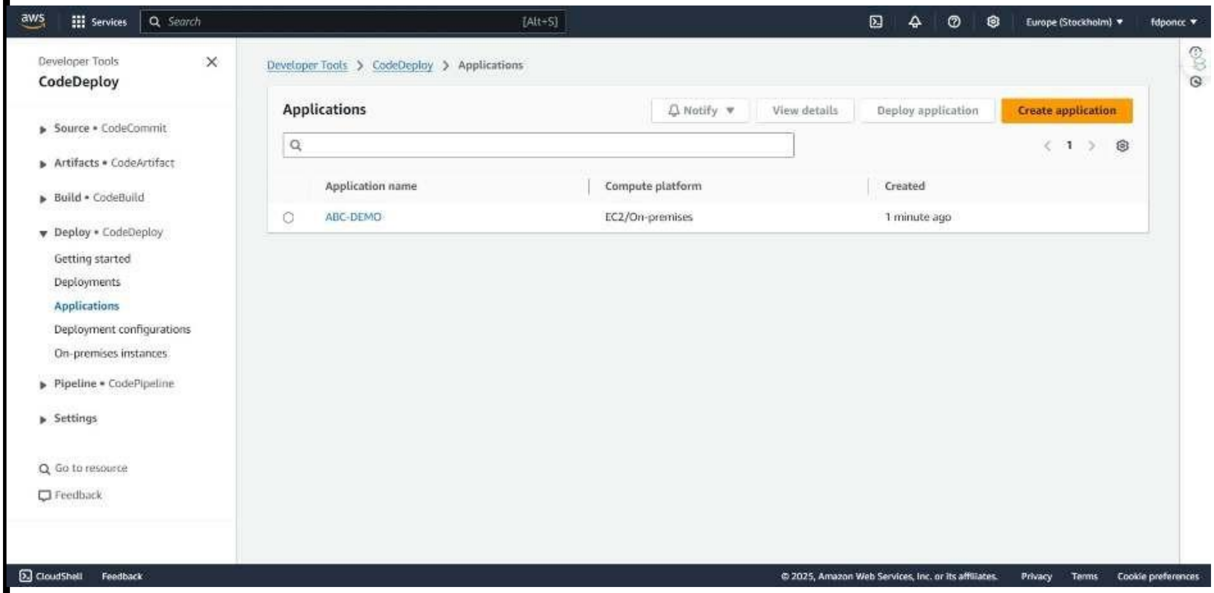
Region
Europe (Stockholm)

Input artifacts
Choose an input artifact for this action. Learn more

SourceArtifact — Defined by: Source

☑ Enable automatic retry on stage failure

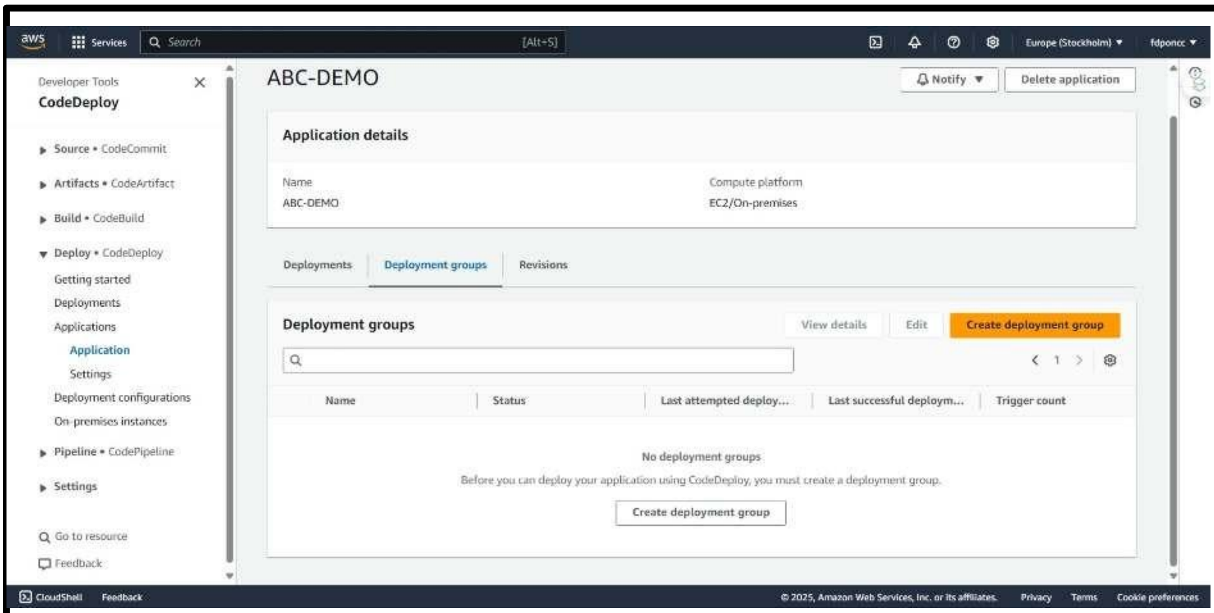
NEW CONSOLE CREATE APPLICATION AND DEPLOYEMENT GROUP



Developer Tools > CodeDeploy > Applications

Applications

	Application name	Compute platform	Created
○	ABC-DEMO	EC2/On-premises	1 minute ago



ABC-DEMO

Application details

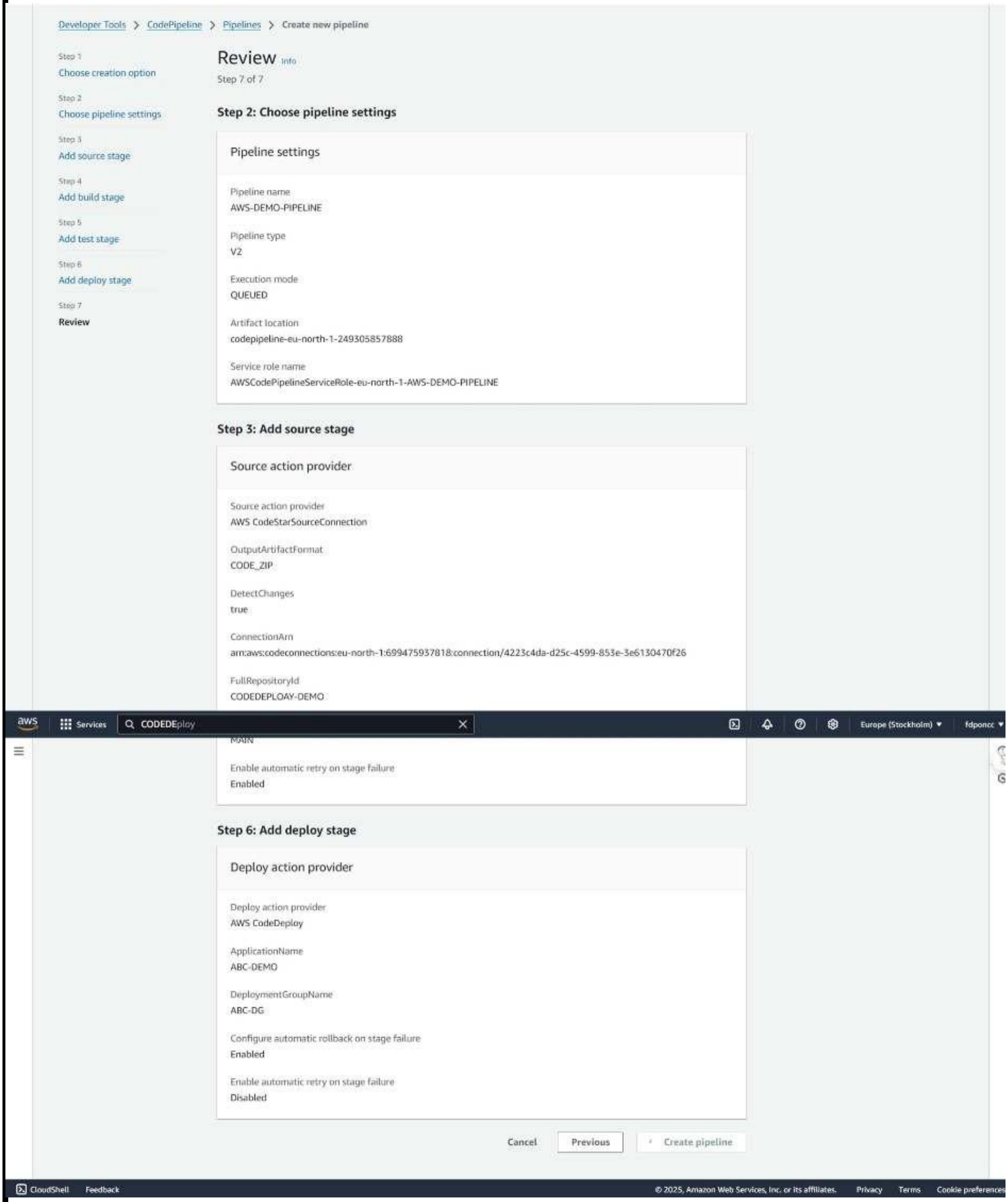
Name	Compute platform
ABC-DEMO	EC2/On-premises

Deployment groups

Name	Status	Last attempted deploy...	Last successful deploym...	Trigger count

No deployment groups

Before you can deploy your application using CodeDeploy, you must create a deployment group.



Review Info

Step 7 of 7

Step 2: Choose pipeline settings

Pipeline settings

Pipeline name
AWS-DEMO-PIPELINE

Pipeline type
V2

Execution mode
QUEUED

Artifact location
codepipeline-eu-north-1-249305857888

Service role name
AWSCodePipelineServiceRole-eu-north-1-AWS-DEMO-PIPELINE

Step 3: Add source stage

Source action provider

Source action provider
AWS CodeStarSourceConnection

OutputArtifactFormat
CODE_ZIP

DetectChanges
true

ConnectionArn
arn:aws:codeconnections:eu-north-1:699475937818:connection/4223c4da-d25c-4599-853e-3e6130470f26

FullRepositoryId
CODEDEPLOAY-DEMO

MAIN

Enable automatic retry on stage failure
Enabled

Step 6: Add deploy stage

Deploy action provider

Deploy action provider
AWS CodeDeploy

ApplicationName
ABC-DEMO

DeploymentGroupName
ABC-DG

Configure automatic rollback on stage failure
Enabled

Enable automatic retry on stage failure
Disabled