Creating a CI/CD Pipeline

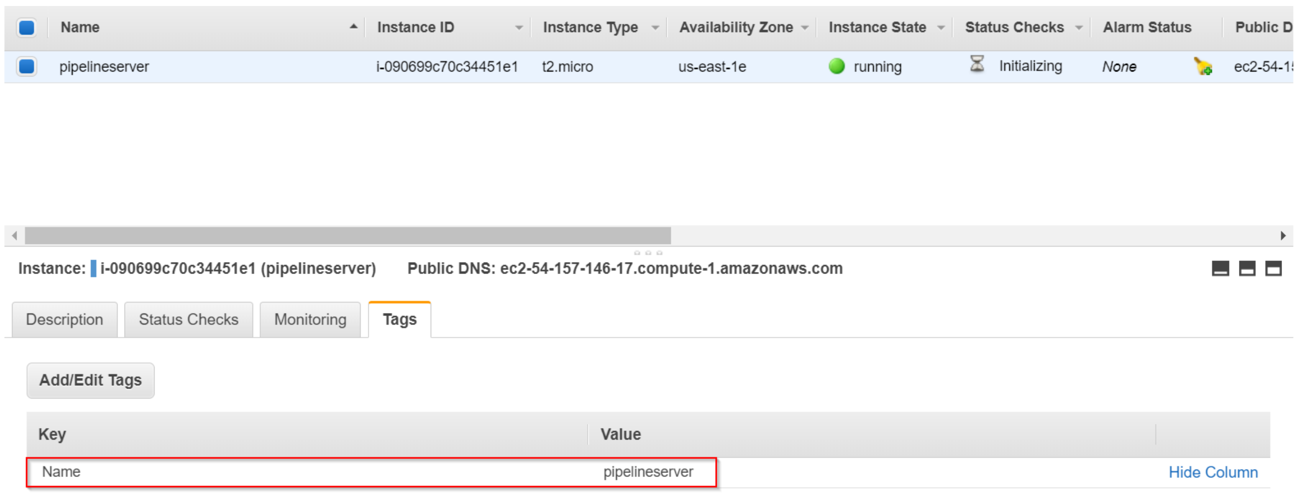
AWS DevOps Certification

support@intellipaat.com
+91-7022374614
US: 1-800-216-8930 (Toll Free)

Creating a CI/CD Pipeline

Step 1: Launch an Ubuntu EC2 instance. In Step 2 of the process, scroll down, and add the user data as provided below. Also, give it a tag, e.g., Name :: pipelineserver

```
#!/bin/bash
sudo apt-get update
sudo apt-get -y install ruby
sudo apt-get install wget
cd /home/ubuntu
wget https://aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com/latest/install
sudo chmod +x ./install
sudo ./install auto
sudo service codedeploy-agent start
```

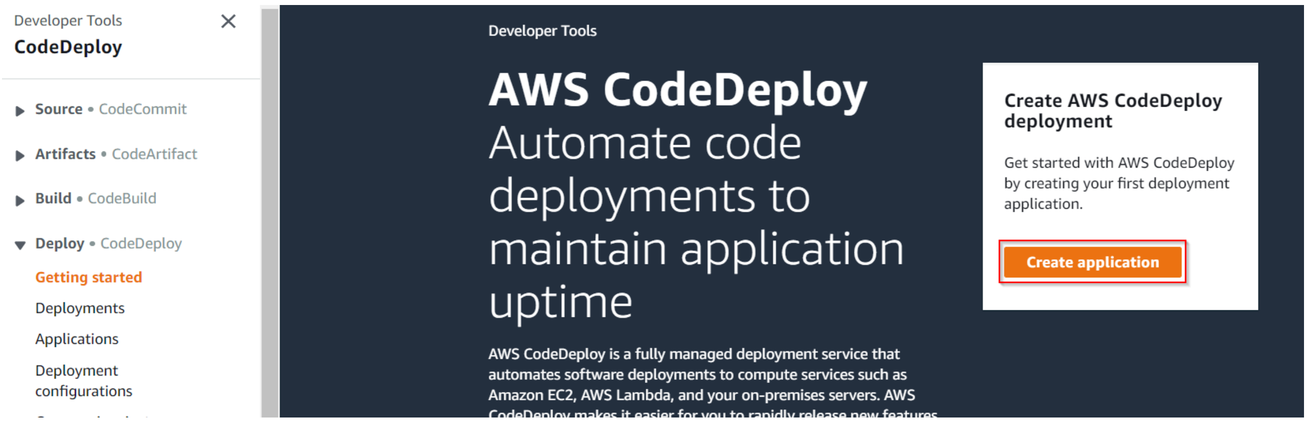


Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public D
pipelineserver	i-090699c70c34451e1	t2.micro	us-east-1e	running	Initializing	None	ec2-54-1

Instance: i-090699c70c34451e1 (pipelineserver)   Public DNS: ec2-54-157-146-17.compute-1.amazonaws.com

Key	Value
Name	pipelineserver

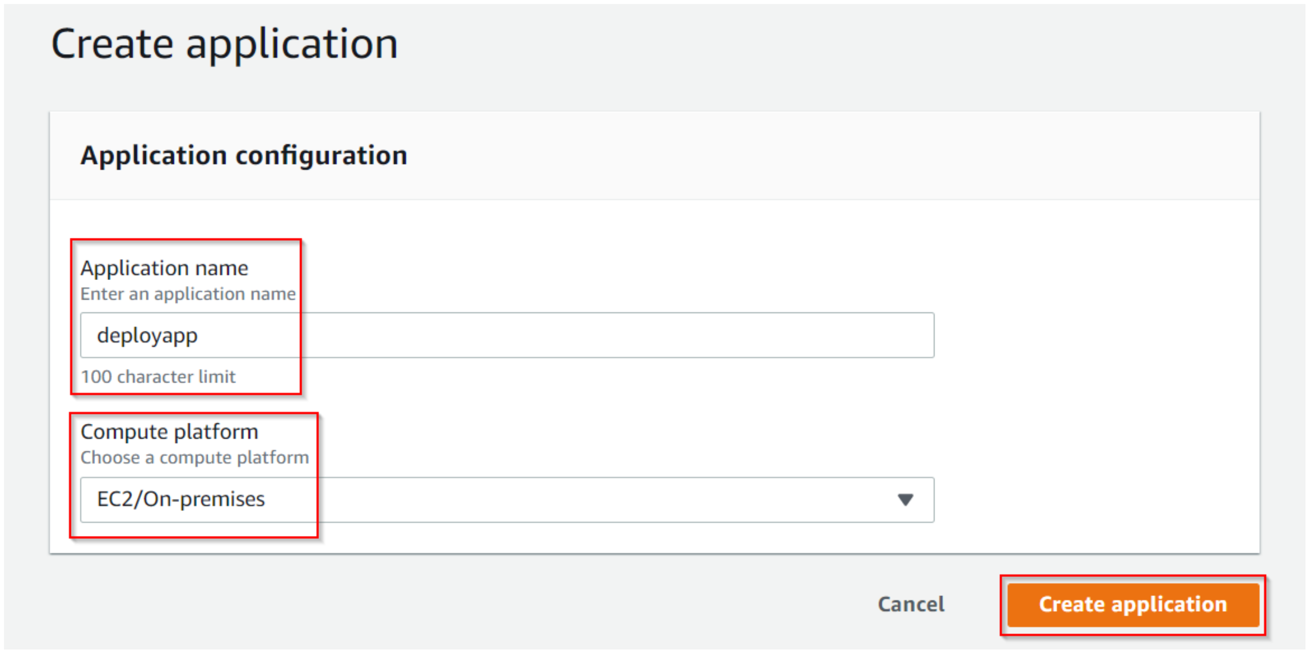
Step 2: Open **CodeDeploy** in Developer Tools to create an application



Developer Tools
CodeDeploy
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Deploy • CodeDeploy
Getting started
Deployments
Applications
Deployment configurations
Developer Tools
AWS CodeDeploy
Automate code deployments to maintain application uptime
Create AWS CodeDeploy deployment
Get started with AWS CodeDeploy by creating your first deployment application.
Create application

AWS CodeDeploy is a fully managed deployment service that automates software deployments to compute services such as Amazon EC2, AWS Lambda, and your on-premises servers. AWS CodeDeploy makes it easier for you to rapidly release new features

Step 3: Give the application a name; choose **EC2/On-premises** for **Compute platform**, and then, click on **Create application**



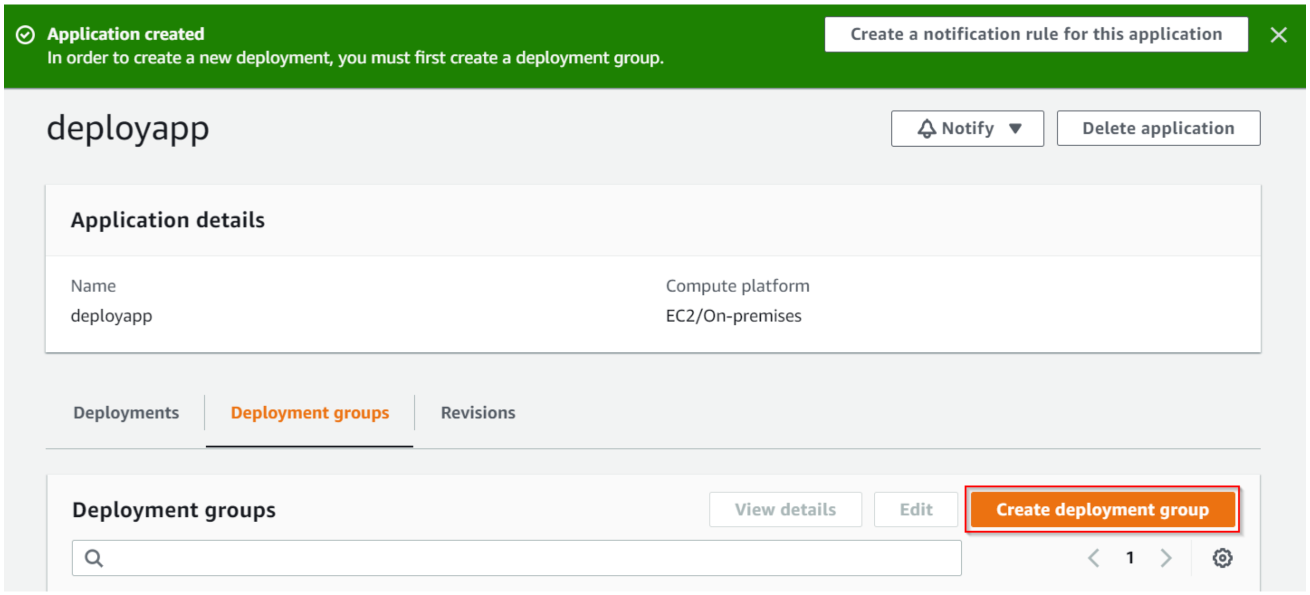
Create application

Application configuration

Application name
Enter an application name
deployapp
100 character limit

Compute platform
Choose a compute platform
EC2/On-premises

Step 4: Once the application is created, choose **Create deployment group**



Application created
In order to create a new deployment, you must first create a deployment group.

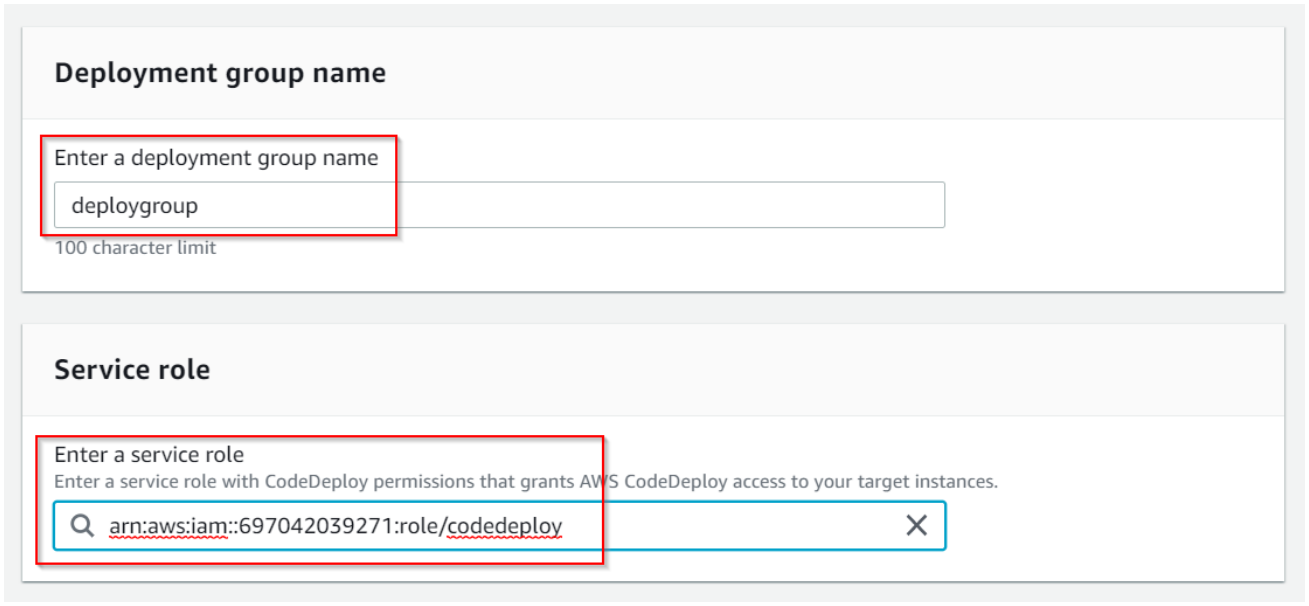
deployapp

Application details

Name
deployapp

Compute platform
EC2/On-premises

Step 5: Give a name to the deployment group, and choose the existing Service role



Deployment group name

Enter a deployment group name

deploygroup

100 character limit

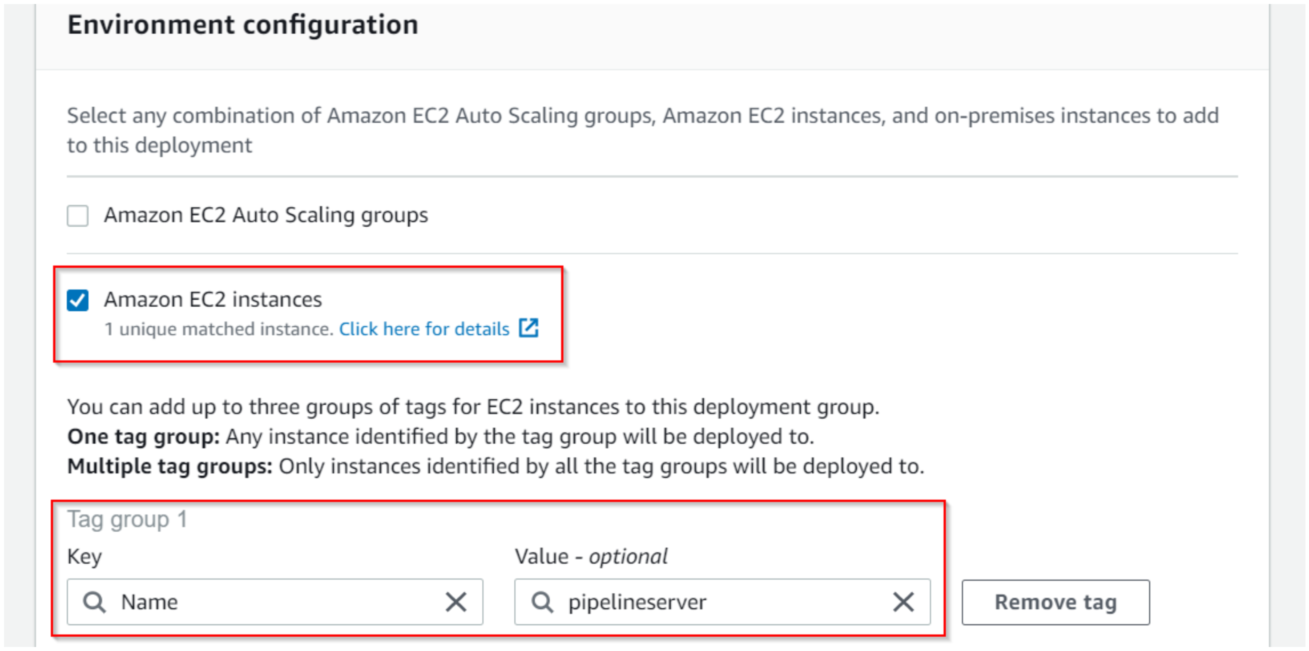
Service role

Enter a service role

Enter a service role with CodeDeploy permissions that grants AWS CodeDeploy access to your target instances.

arn:aws:iam::697042039271:role/codedeploy

- Select **Amazon EC2 instances** as you are using just one instance here, and enter the tag to mention the instance



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

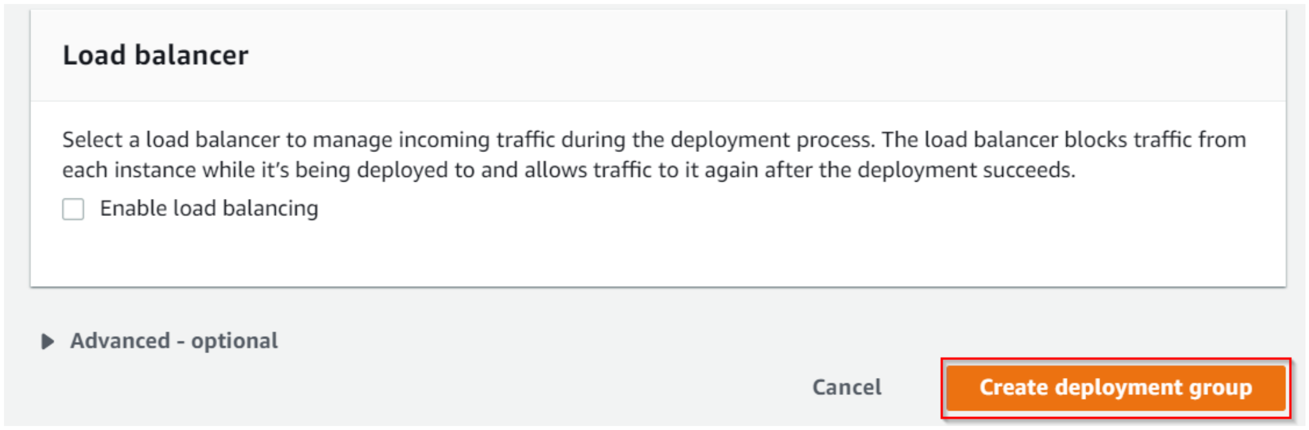
1 unique matched instance. Click here for details

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	Value - *optional*
Name	pipelineserver

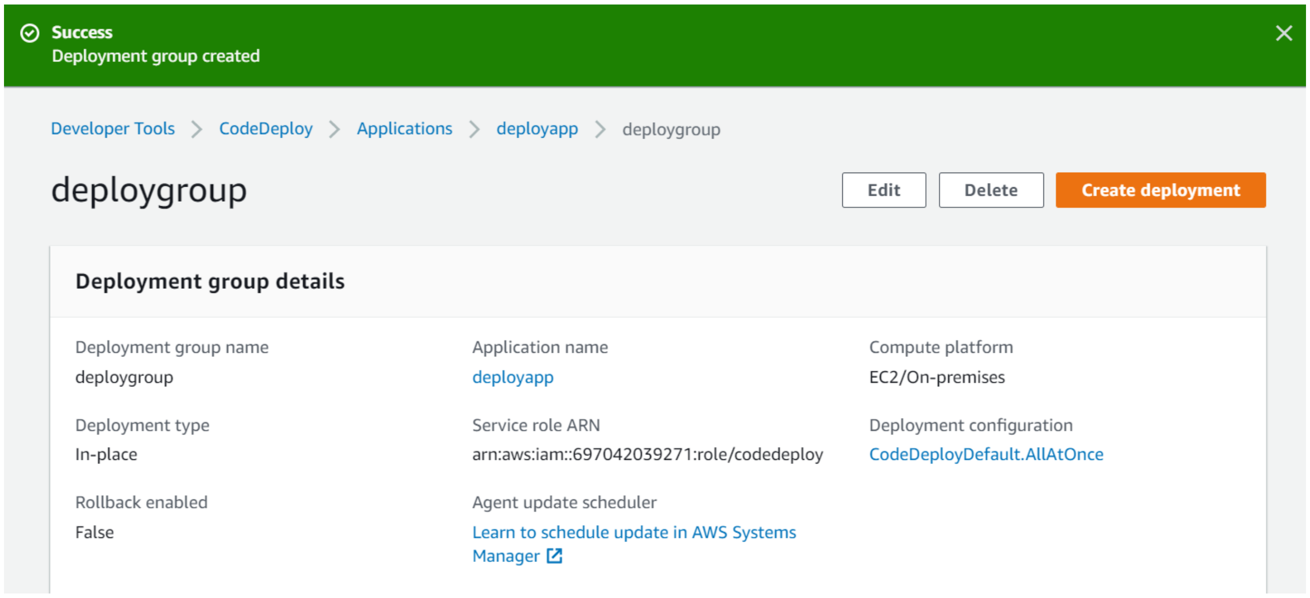
- Once the given information is provided, click on **Create deployment group**



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



Success
Deployment group created

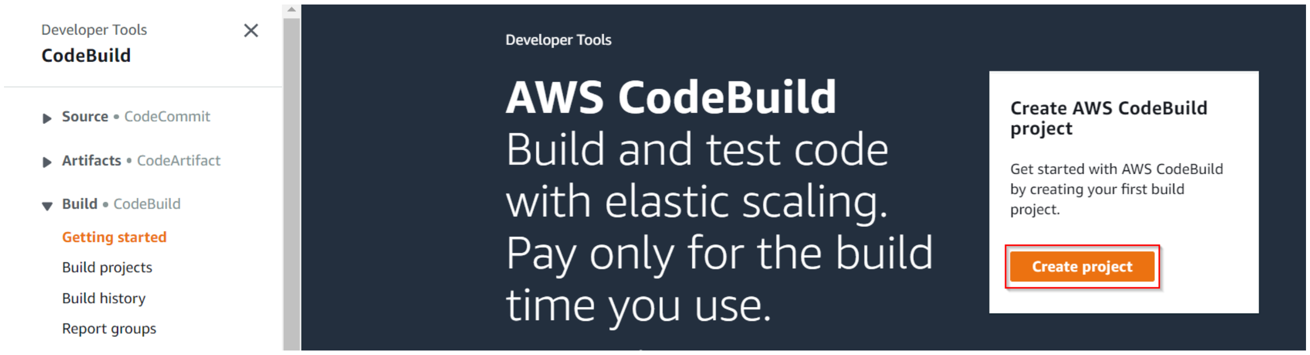
Developer Tools > CodeDeploy > Applications > deployapp > deploygroup

deploygroup

Deployment group details

Deployment group name	Application name	Compute platform
deploygroup	deployapp	EC2/On-premises
Deployment type	Service role ARN	Deployment configuration
In-place	arn:aws:iam::697042039271:role/codedeploy	CodeDeployDefault.AllAtOnce
Rollback enabled	Agent update scheduler	
False	Learn to schedule update in AWS Systems Manager	

Step 6: Next, create a CodeBuild build project by clicking on **Create project**

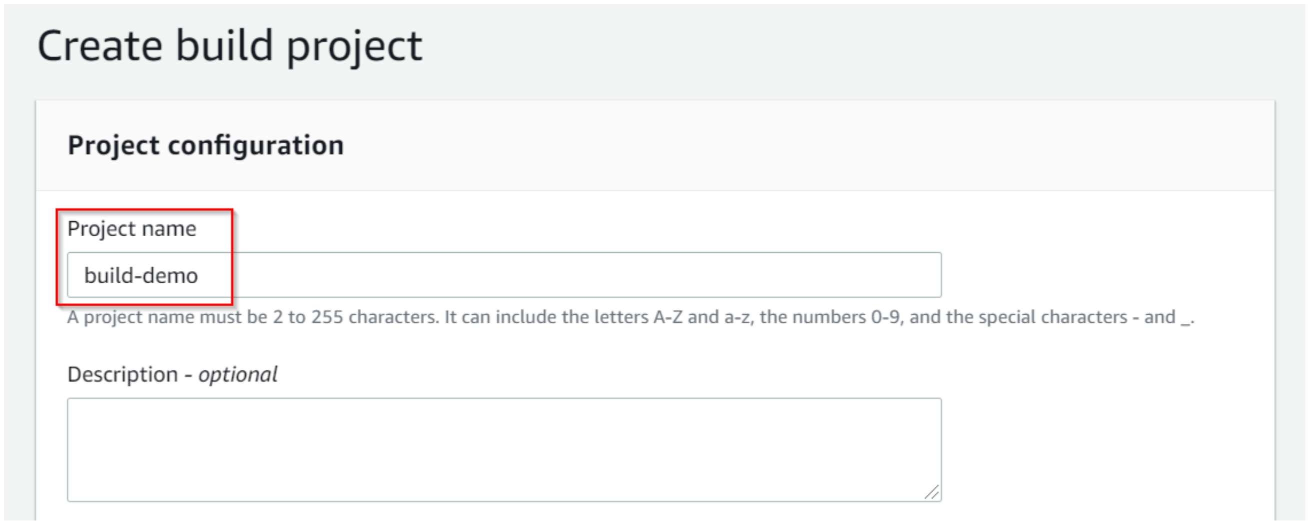


AWS CodeBuild
Build and test code with elastic scaling. Pay only for the build time you use.

Create AWS CodeBuild project

Get started with AWS CodeBuild by creating your first build project.

- Provide a name, and scroll down to the **Source** section



Create build project

Project configuration

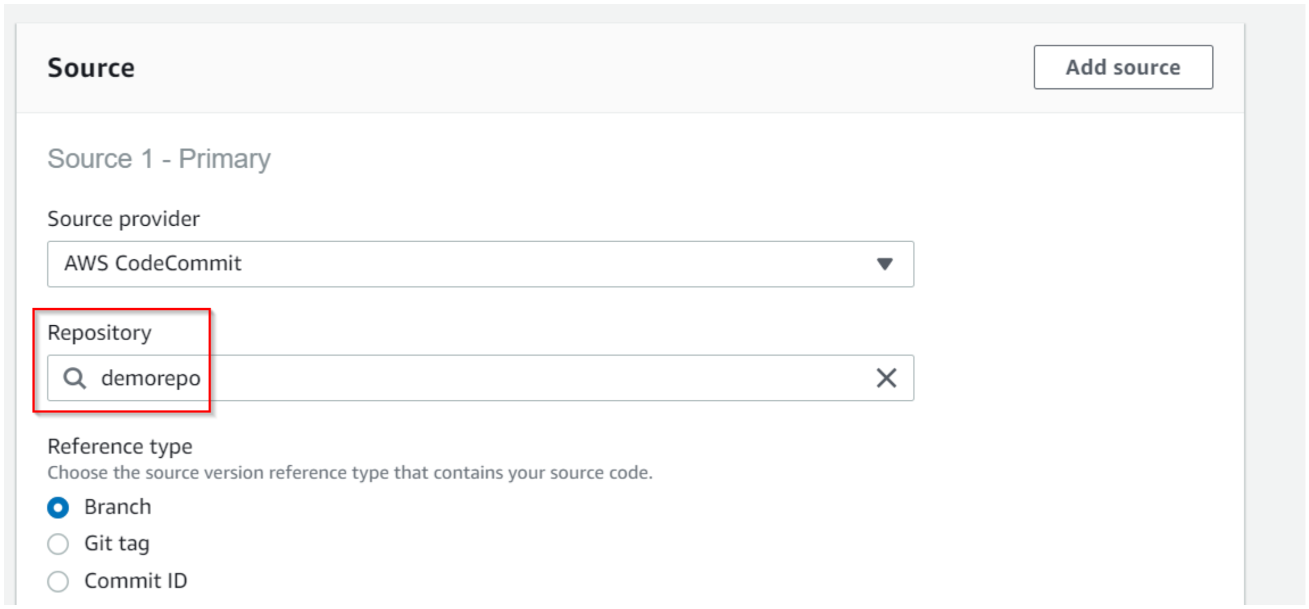
Project name

build-demo

A project name must be 2 to 255 characters. It can include the letters A-Z and a-z, the numbers 0-9, and the special characters - and _.

Description - *optional*

- Select the Source provider (here, it is CodeCommit, but if your code is in GitHub, then select that), and then choose the repository in which the code is available



Source

Source 1 - Primary

Source provider

AWS CodeCommit

Repository

demorepo

Reference type

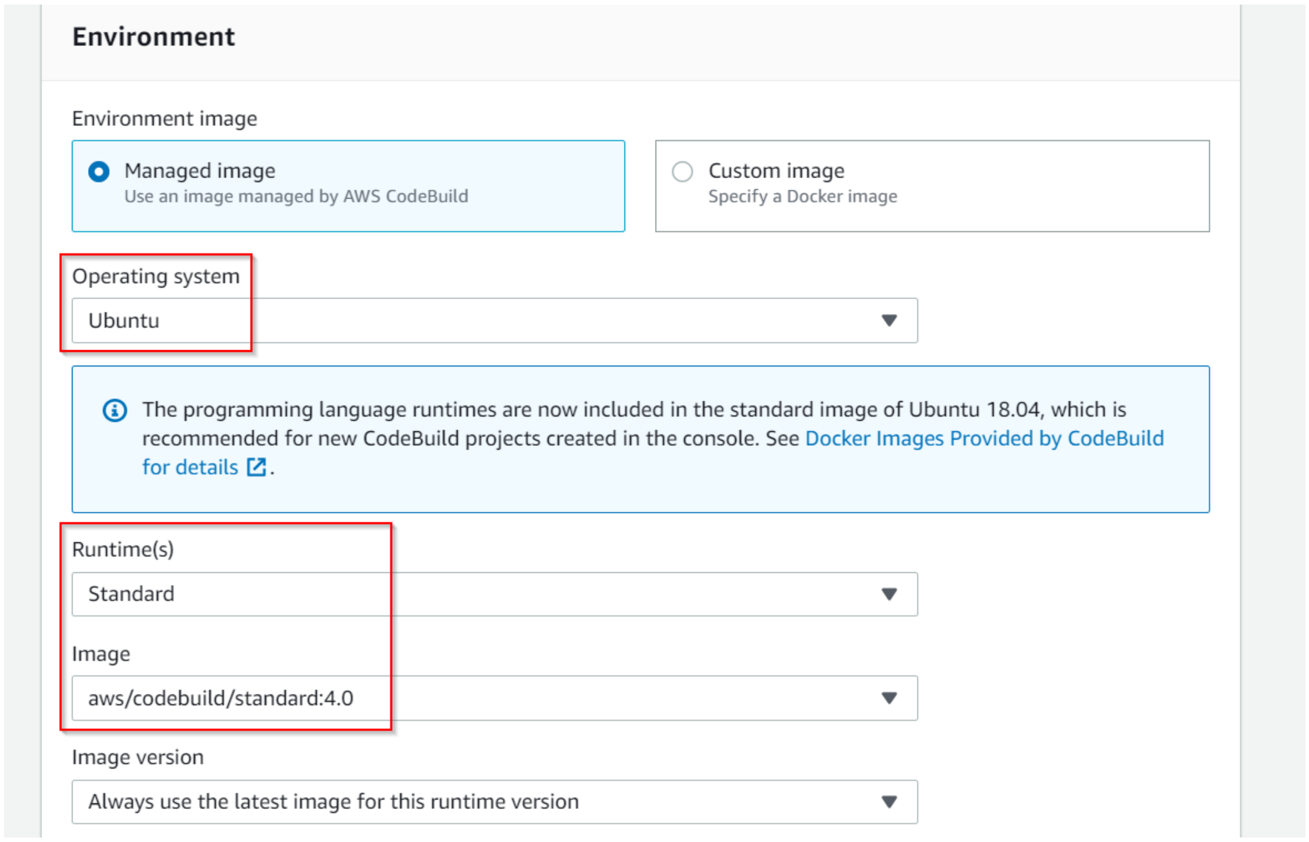
Choose the source version reference type that contains your source code.

● Branch

○ Git tag

○ Commit ID

- Let the image be a **Managed image**; choose Ubuntu as the OS; and let the Runtime and Image be the same as given in the screenshot below:



Environment

Environment image

● Managed image
Use an image managed by AWS CodeBuild

○ Custom image
Specify a Docker image

Operating system: Ubuntu

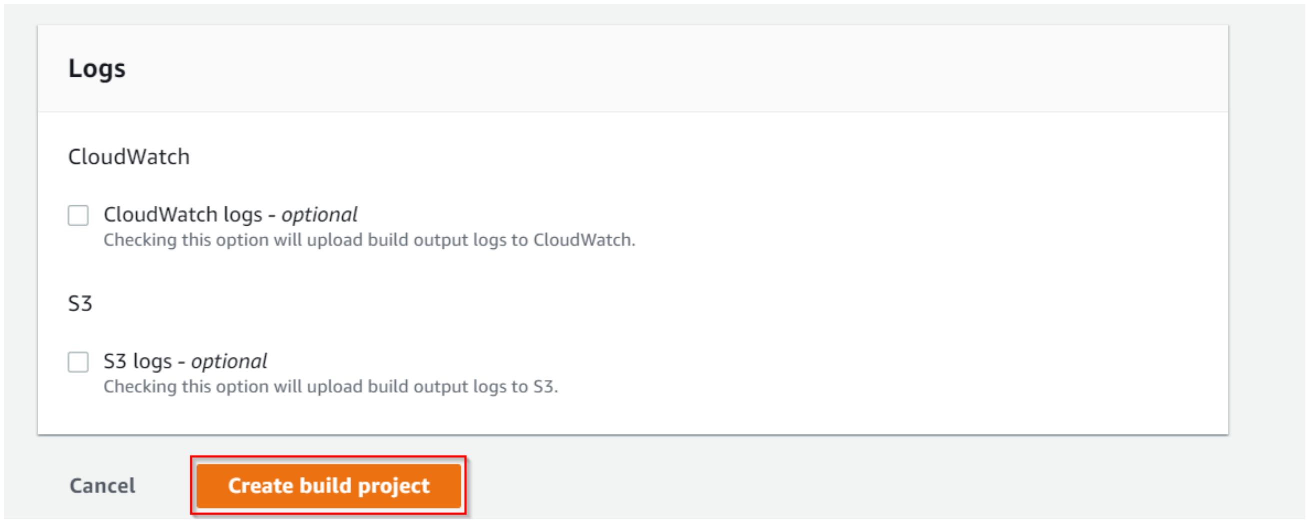
The programming language runtimes are now included in the standard image of Ubuntu 18.04, which is recommended for new CodeBuild projects created in the console. See Docker Images Provided by CodeBuild for details.

Runtime(s): Standard

Image: aws/codebuild/standard:4.0

Image version: Always use the latest image for this runtime version

- Finally, click on **Create build project**



Logs

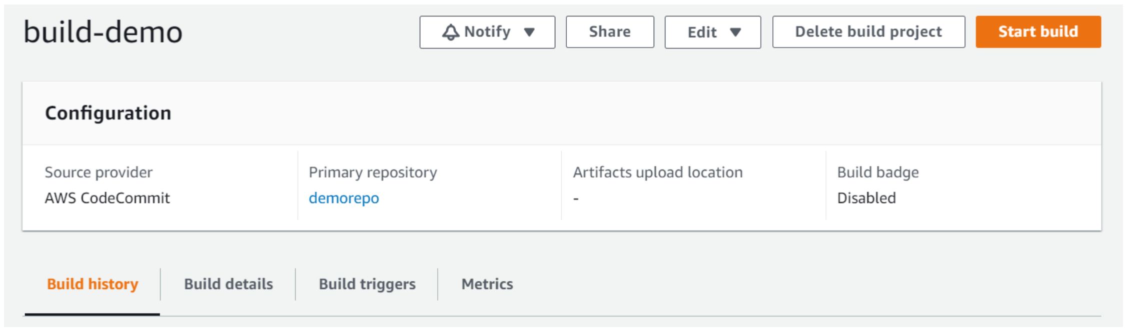
CloudWatch

☐ CloudWatch logs - *optional*
Checking this option will upload build output logs to CloudWatch.

S3

☐ S3 logs - *optional*
Checking this option will upload build output logs to S3.

- It has been created successfully

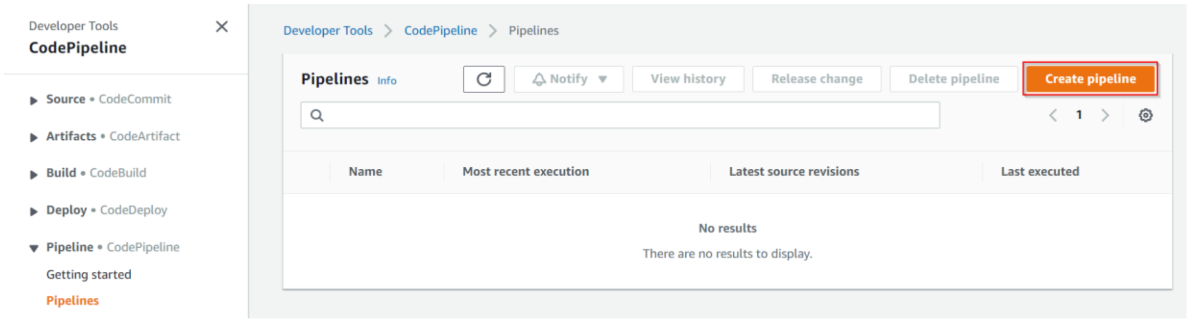


build-demo

Configuration

Source provider	Primary repository	Artifacts upload location	Build badge
AWS CodeCommit	demorepo	-	Disabled

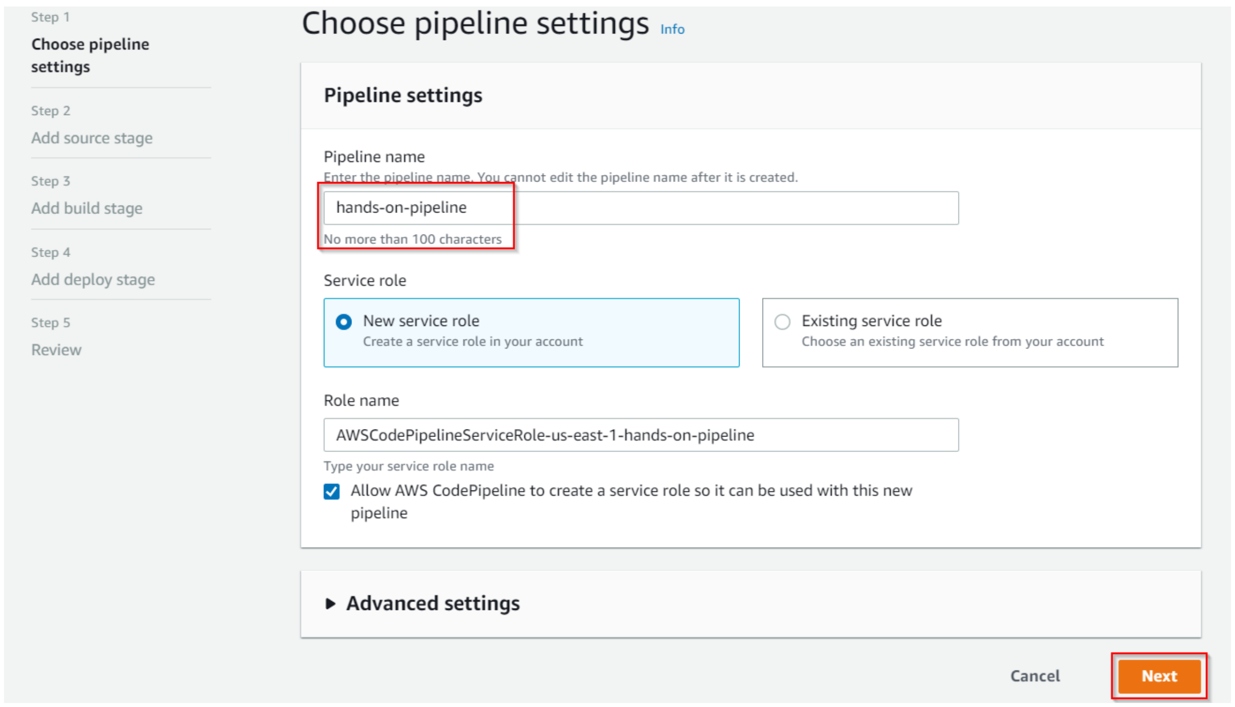
Step 7: Now, go to **CodePipeline** to start building the pipeline



Name	Most recent execution	Latest source revisions	Last executed

No results
There are no results to display.

- Provide the Pipeline name; let it create a **New service role** for this pipeline, and hit **Next**



Choose pipeline settings Info

Pipeline settings

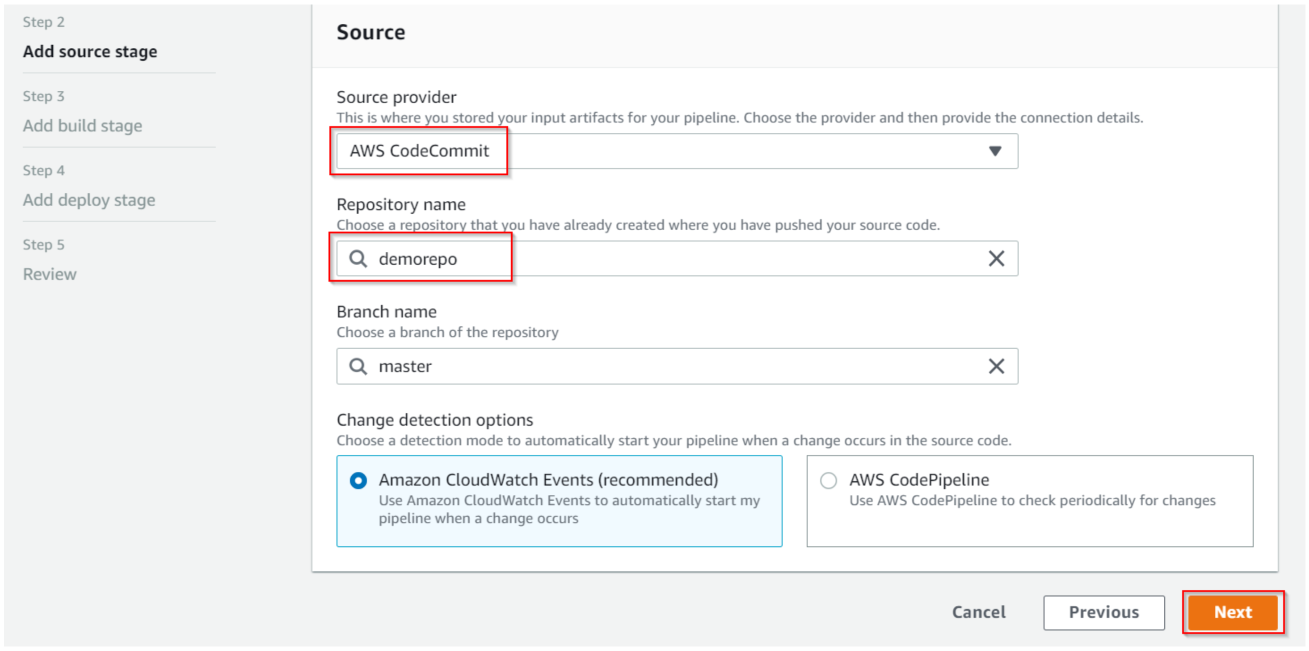
Pipeline name
Enter the pipeline name. You cannot edit the pipeline name after it is created.
hands-on-pipeline
No more than 100 characters

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-us-east-1-hands-on-pipeline
Type your service role name

✓ Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

Step 8: In the Source stage, select the Source provider and the Repository name according to where your application's code is residing, and click on **Next**



Step 2
Add source stage
Step 3
Add build stage
Step 4
Add deploy stage
Step 5
Review

Source

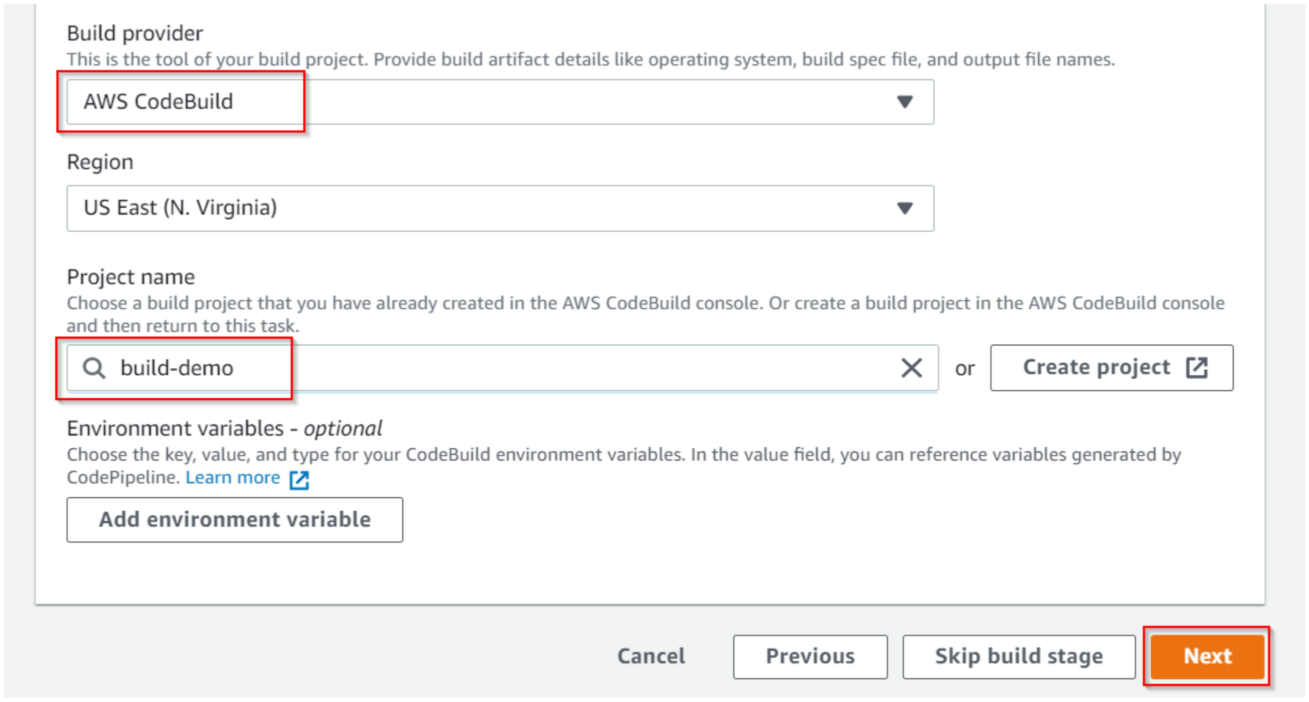
Source provider
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.
AWS CodeCommit

Repository name
Choose a repository that you have already created where you have pushed your source code.
demorepo

Branch name
Choose a branch of the repository
master

Change detection options
Choose a detection mode to automatically start your pipeline when a change occurs in the source code.
[✓ Amazon CloudWatch Events (recommended)] Use Amazon CloudWatch Events to automatically start my pipeline when a change occurs
AWS CodePipeline — Use AWS CodePipeline to check periodically for changes

Step 9: In the Build stage, select the Build Provider and then the created Build project, and click on **Next**



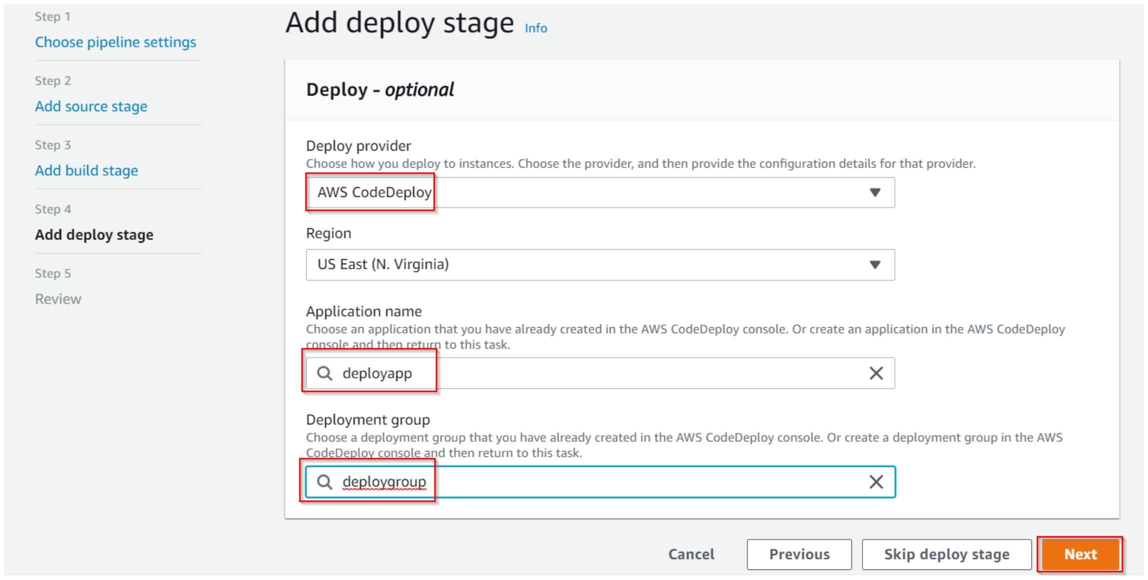
Build provider
This is the tool of your build project. Provide build artifact details like operating system, build spec file, and output file names.
AWS CodeBuild

Region
US East (N. Virginia)

Project name
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.
build-demo

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

Step 10: Select the deployment provider; choose the application; select the corresponding deployment group, and press **Next**


Step 1
Choose pipeline settings
Step 2
Add source stage
Step 3
Add build stage
Step 4
Add deploy stage
Step 5
Review

Add deploy stage Info

Deploy - *optional*

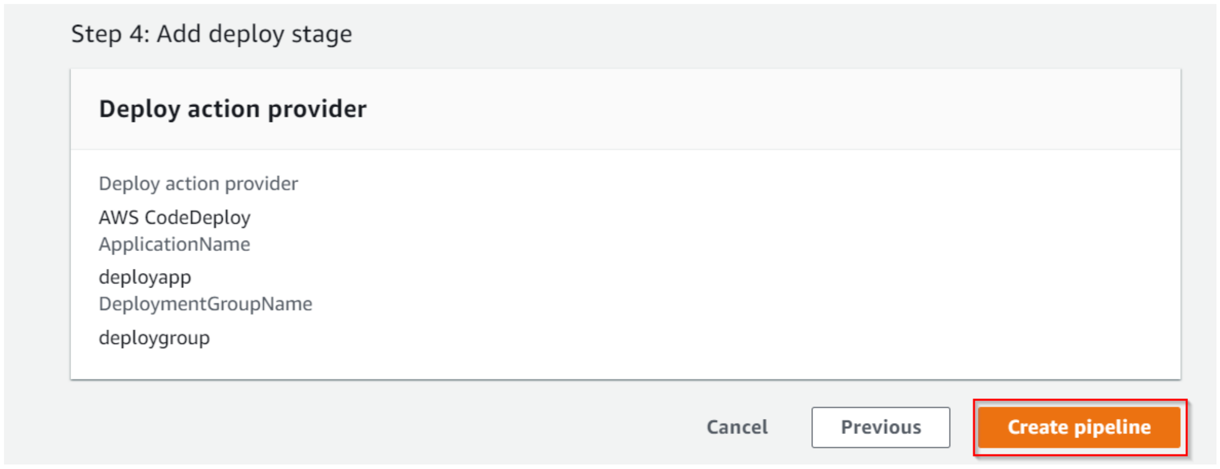
Deploy provider
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.
AWS CodeDeploy

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 CodeDeploy console and then return to this task.
deployapp

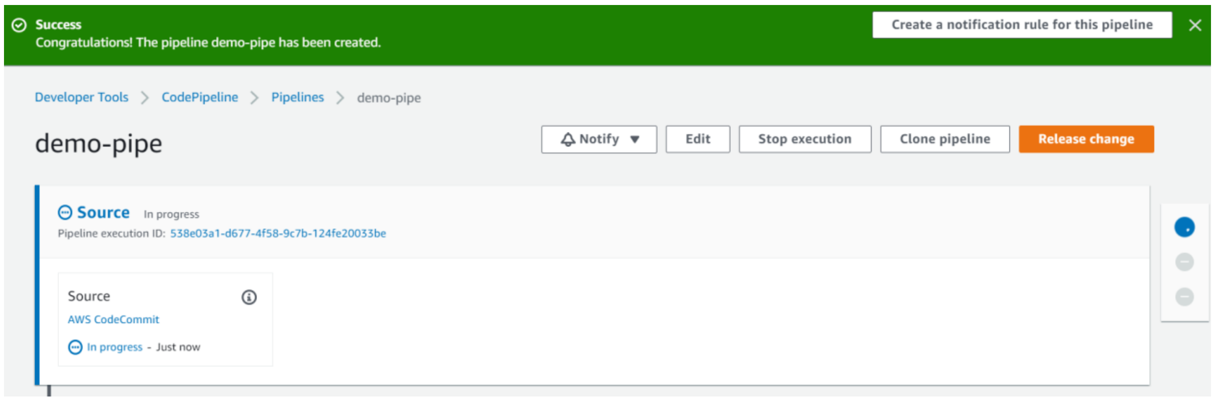
Deployment group
Choose a deployment group that you have already created in the AWS CodeDeploy console. Or create a deployment group in the AWS CodeDeploy console and then return to this task.
deploygroup

Step 11: Final step is to review and click on **Create pipeline**. Once it is done, wait until it succeeds to check out your application


Step 4: Add deploy stage

Deploy action provider

Deploy action provider
AWS CodeDeploy
ApplicationName
deployapp
DeploymentGroupName
deploygroup



Success
Congratulations! The pipeline demo-pipe has been created.

Create a notification rule for this pipeline
Developer Tools > CodePipeline > Pipelines > demo-pipe
Notify
Edit
Stop execution
Clone pipeline
Release change

demo-pipe

Source In progress
Pipeline execution ID: 538e03a1-d677-4f58-9c7b-124fe20033be

Source
AWS CodeCommit
In progress - Just now