

## What is Devops ?

Development + Operations , the methodologies followed here is known as Continuous Integration and Continuous Deployment.

Continuous Integration means – Once the process is started it cannot be stopped, along with this we will Integrate the process with other stages of deployment.

The AWS Devops Services are as follows.

1. Codebuild
2. Codedeploy – this services is used for deployment, the codes are deployed in EC2 instance
3. Codecommit – this is similar to Github, here the developers commit the code and push it to the deployment stage
4. Codepipeline

The best way to use the AWS Devops tools is to use the codecommit to push the code to deployment stage ( EC2 instance) in a development environment, once its successful the same code can be deployed in the Production environment.

### **Codecommit –**

Code commit includes repositories; it means a place to store the code.

To work with Code commit we must be aware of the Git commands with purpose

You can download the Git Cli commands sheet from the below link

[Git cheat sheet | Atlassian Git Tutorial](#)

---

Create an EC2 instance in your AWS console, I am creating a Linux Instance with Public IP.

The first step is to create a VPC –

Check this youtube Link to understand how to Calculate VPC and Subnet CIDR - [\(1\) Simply Calculate AWS Subnets CIDR Blocks \[Create AWS subnets IPv4 range | Reserved 5 IP addresses\] - YouTube](#)

VPC > Your VPCs > vpc-0fca597223c819fe3 / AWSDevops VPC

**Details** [Info](#) [Actions](#)

VPC ID vpc-0fca597223c819fe3	State <span style="color: green;">Available</span>	DNS hostnames Disabled	DNS resolution Enabled
Tenancy Default	DHCP options set dopt-0948ae9919e617053	Main route table rtb-00ba5e58e7bc1b9f9	Main network ACL acl-0f35593b2e3dff700
Default VPC No	IPv4 CIDR 10.0.0.0/24	IPv6 pool -	IPv6 CIDR -
Route 53 Resolver DNS Firewall rule groups -	Owner ID 601007615373		

For the VPC CIDR 10.0.0.0/24 the subnet range will be as below

← → ⌂ <https://www.davidc.net/sites/default/subnets/subnets.html>

## Visual Subnet Calculator

Enter the network you wish to subnet:

Network Address 10.0.0.0	Mask bits /24	Update	Reset
-----------------------------	------------------	--------	-------

Show columns:  Subnet address  Netmask  Range of addresses  Useable IPs  Hosts  Divide  Join

Click below to split and join subnets.  
If you wish to save this subnetting for later, bookmark [this hyperlink](#).

Subnet address	Netmask	Range of addresses	Useable IPs	Hosts	Divide	Join
10.0.0.0/25	255.255.255.128	10.0.0.0 - 10.0.0.127	10.0.0.1 - 10.0.0.126	126	<a href="#">Divide</a>	/25
10.0.0.128/25	255.255.255.128	10.0.0.128 - 10.0.0.255	10.0.0.129 - 10.0.0.254	126	<a href="#">Divide</a>	/24

## Create an Internet Gateway and attach to the VPC you created

Internet gateways (1/2) <a href="#">Info</a>						<a href="#">Actions ▾</a>	<a href="#">Create internet gateway</a>	
<input type="text"/> Filter internet gateways						<a href="#">View details</a>	<a href="#">Attach to VPC</a>	
<input type="checkbox"/>	Name	Internet gateway ID	State	VPC ID		<a href="#">Detach from VPC</a>	<a href="#">Manage tags</a>	<a href="#">Delete internet gateway</a>
<input type="checkbox"/>	igw-0134984c6bd3b621a	igw-0134984c6bd3b621a	Detached	-		<a href="#">Detach from VPC</a>	<a href="#">Manage tags</a>	<a href="#">Delete internet gateway</a>
<input checked="" type="checkbox"/>	AWS-Devops-IGW	igw-0336fd99df0f71204	Detached	-		<a href="#">Detach from VPC</a>	<a href="#">Manage tags</a>	<a href="#">Delete internet gateway</a>

## Attach Internet Gateway to the Route table

Route tables (2) <a href="#">Info</a>									<a href="#">Actions ▾</a>	<a href="#">Create route table</a>
<input type="text"/> Filter route tables									<a href="#">View details</a>	<a href="#">Edit</a>
<input type="checkbox"/>	Name	Route table ID	Explicit...	Edge a...	Main	VPC	Ow...		<a href="#">View details</a>	<a href="#">Edit</a>
<input type="checkbox"/>	AWS-Devops-PUB-SUBNET-Routet...	rtb-00ba5e58e7bc1b9f9	-	-	Yes	vpc-0fcfa597223c819fe3   AWSDevops VPC	60100...		<a href="#">View details</a>	<a href="#">Edit</a>
<input type="checkbox"/>	-	rtb-012b884a70b3cf8b7	-	-	Yes	vpc-0e8cef2844fd37145	60100...		<a href="#">View details</a>	<a href="#">Edit</a>

Choose the route table and at the bottom choose route – Edit Routes – add routes 0.0.0.0/0 – choose the internet gateway you created.

https://ap-south-1.console.aws.amazon.com/vpc/home?region=ap-south-1#EditRoutes:RouteTableId=rtb-00ba5e58e7bc1b9f9

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/24	local	Active	No
0.0.0.0/0	igw-0336fd99df0f71204 (AWS-Devops-IGW)	-	No

Add route

Cancel Preview Save changes

The screenshot shows the AWS VPC Subnet Associations page. On the left, there's a sidebar with options like EC2 Global View, Filter by VPC, and various VPC components. The main area has tabs for Routes, Subnet associations (which is selected), Edge associations, Route propagation, and Tags. Under 'Explicit subnet associations (1)', it lists a single entry: subnet-0b126ca1f1fdb34e1 / AWSDevops-PublicSubnet-LinuxEC2-ap-south-1 with an IPv4 CIDR of 10.0.0.0/25. There's also a section for 'Subnets without explicit associations (0)' which is currently empty.

Launch an Ec2 instance .. choose a Linux AMI, instance type and configure an EC2 instance as below

The screenshot shows the AWS Launch Instance Wizard at Step 3: Configure Instance Details. The URL is https://ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#LaunchInstanceWizard:.

**Step 3: Configure Instance Details**

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing,

**Number of instances**: 1

**Purchasing option**:  Request Spot instances

**Network**: vpc-0fca597223c819fe3 | AWSDevops VPC

**Subnet**: subnet-0b126ca1f1fdb34e1 | AWSDevops-PublicSub   
123 IP Addresses available

**Auto-assign Public IP**:

**Hostname type**: Use subnet setting (IP name)

**DNS Hostname**:

- Enable IP name IPv4 (A record) DNS requests
- Enable resource-based IPv4 (A record) DNS requests
- Enable resource-based IPv6 (AAAA record) DNS requests

**Placement group**:  Add instance to placement group

**Capacity Reservation**: Open

**Domain join directory**: No directory

1. Choose AMI    2. Choose Instance Type    3. Configure Instance    4. Add Storage    5. Add Tags    6. Configure Security Group    7. Review

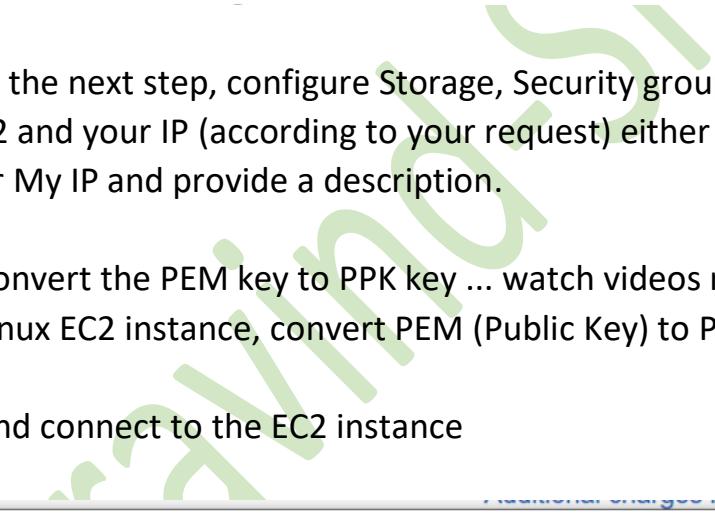
### Step 3: Configure Instance Details

Placement group	<input type="checkbox"/> Add instance to placement group
Capacity Reservation	Open
Domain join directory	No directory
IAM role	None
Shutdown behavior	Terminate
Stop - Hibernate behavior	<input type="checkbox"/> Enable hibernation as an additional stop behavior
Enable termination protection	<input checked="" type="checkbox"/> Protect against accidental termination
Monitoring	<input type="checkbox"/> Enable CloudWatch detailed monitoring Additional charges apply.
Tenancy	Shared - Run a shared hardware instance
Credit specification	<input type="checkbox"/> Unlimited Additional charges may apply
File systems	<input type="button" value="Add file system"/> <input type="button" value="Create new file system"/>

In the next step, configure Storage, Security group by choosing SSH, Port 22 and your IP (according to your request) either you can choose custom or My IP and provide a description.

Convert the PEM key to PPK key ... watch videos related to creating a Linux EC2 instance, convert PEM (Public Key) to PPK (Private key)

And connect to the EC2 instance



```
ec2-user@ip-10-0-0-53:~$ Using username "ec2-user".
ec2-user@ip-10-0-0-53:~$ Authenticating with public key "imported-openssh-key"
[ec2-user@ip-10-0-0-53 ~]$
```

Let's follow the **Codecommit** procedures now

1. Step -1 – Create a IAM user and provide him Codecommitpoweruser policy as permissions. PFB

## Add user

1 2 3 4 5

### Set permissions

Add user to group     Copy permissions from existing user     Attach existing policies directly

Create policy   

Policy name		Type	Used as
<input type="checkbox"/>	AWSCodeCommitFullAccess	AWS managed	None
<input checked="" type="checkbox"/>	AWSCodeCommitPowerUser	AWS managed	None
<input type="checkbox"/>	AWSCodeCommitReadOnly	AWS managed	None

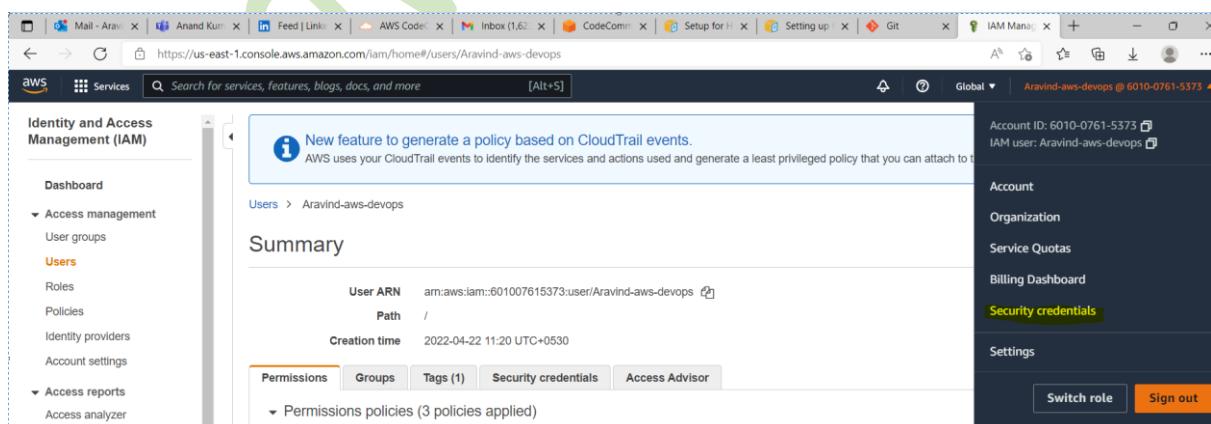
### Set permissions boundary

Go to ec2 instance and in terminal type aws configure , input the access key, secret key of the IAM user you created and your EC2 instance region.

Step -2 – Install GIT in terminal – follow this link for commands - [Git \(git-scm.com\)](https://git-scm.com)

In my case the command is sudo yum install -y

Step – 3 – Sign out of your aws console and login as the IAM user that you created and then click on the top right of your IAM user from the console and choose security credentials .. PFB



The screenshot shows the AWS IAM User Security Credentials page. The URL is https://us-east-1.console.aws.amazon.com/iam/home#/users/Aravind-aws-devops. The left sidebar shows 'Identity and Access Management (IAM)' with 'Users' selected. The main content area shows the 'Summary' for the user 'Aravind-aws-devops'. It includes details like User ARN: arn:aws:iam::601007615373:user/Aravind-aws-devops, Path: /, and Creation time: 2022-04-22 11:20 UTC+0530. Below this, there are tabs for 'Permissions', 'Groups', 'Tags (1)', 'Security credentials', and 'Access Advisor'. The 'Security credentials' tab is highlighted with a green bar. On the right side, there is a sidebar with links for 'Account', 'Organization', 'Service Quotas', 'Billing Dashboard', and 'Security credentials' (which is also highlighted). At the bottom right of the page are 'Switch role' and 'Sign out' buttons.

Choose AWS code commit credentials and click on generate HTTPS git credentials and download the credentials

## My security credentials

### Account details

User name	Aravind-aws-devops (created on 2022-04-22 11:20 UTC+0530)
User ARN	arn:aws:iam::601007615373:user/Aravind-aws-devops
AWS account ID	601007615373
Account canonical user ID	4a24616b598e0621c12fdca506065b94d0e2f4bc1b46b4e9e14ccfc69912215b

AWS IAM credentials    AWS CodeCommit credentials    Amazon MCS credentials

SSH keys

Use SSH public keys to authenticate access to AWS CodeCommit repositories. [Learn more](#)

[Upload SSH public key](#)

No SSH keys have been uploaded.

HTTPS Git credentials for AWS CodeCommit

Generate a user name and password you can use to authenticate HTTPS connections to AWS CodeCommit repositories. You can generate a new password at any time.

[Generate credentials](#)

Open the terminal and we need to clone the repository we created. PFB..

The screenshot shows the AWS CodeCommit console. On the left, there's a sidebar with 'Developer Tools' and 'CodeCommit' selected. Under 'CodeCommit', 'Repositories' is highlighted. The main area shows a repository named 'Awsdevops-sample-repo'. The URL 'https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/Awsdevops-sample-repo' is displayed, with a 'Copy' button next to it. Below the URL, there are three tabs: 'HTTPS' (which is selected), 'SSH', and 'HTTPS (GRC)'. The page is divided into sections: 'Step 1: Prerequisites', 'Step 2: Git credentials', and 'Step 3: Clone the repository'. The 'Clone URL' section contains the HTTPS URL of the repository.

```
[ec2-user@ip-10-0-0-53 ~]$ git clone https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/Awsdevops-sample-repo  
Cloning into 'Awsdevops-sample-repo'...  
Username for 'https://git-codecommit.ap-south-1.amazonaws.com': Aravind-aws-devo  
ps-at-601007615373  
Password for 'https://Aravind-aws-devops-at-601007615373@git-codecommit.ap-south  
-1.amazonaws.com':  
warning: You appear to have cloned an empty repository.  
[ec2-user@ip-10-0-0-53 ~]$
```

The username and password for the git clone is the credentials of code commit you downloaded in the above step.

Step 4 – Lets download some data to our repository

Consider the below sample URL from github

- <https://github.com/cameronmcnz/rock-paper-scissors.git>

Follow the steps in below snip in your terminal

```
[ec2-user@ip-10-0-0-53 ~]$ git clone https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/Awsdevops-sample-repo
Cloning into 'Awsdevops-sample-repo'...
Username for 'https://git-codecommit.ap-south-1.amazonaws.com': Aravind-aws-devops-at-601007615373
Password for 'https://Aravind-aws-devops-at-601007615373@git-codecommit.ap-south-1.amazonaws.com':
warning: You appear to have cloned an empty repository.
[ec2-user@ip-10-0-0-53 ~]$ ls
Awsdevops-sample-repo
[ec2-user@ip-10-0-0-53 ~]$ cd Awsdevops-sample-repo/
[ec2-user@ip-10-0-0-53 Awsdevops-sample-repo]$ ls
[ec2-user@ip-10-0-0-53 Awsdevops-sample-repo]$ wget https://github.com/cameronmcnz/rock-paper-scissors.git
--2022-04-22 06:43:40-- https://github.com/cameronmcnz/rock-paper-scissors.git
Resolving github.com (github.com) ... 13.234.210.38
Connecting to github.com (github.com) |13.234.210.38|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://github.com/cameronmcnz/rock-paper-scissors [following]
--2022-04-22 06:43:40-- https://github.com/cameronmcnz/rock-paper-scissors
Reusing existing connection to github.com:443.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'rock-paper-scissors.git'

[ <=>
2022-04-22 06:43:41 (44.1 MB/s) - 'rock-paper-scissors.git' saved [166724]

[ec2-user@ip-10-0-0-53 Awsdevops-sample-repo]$ ls
rock-paper-scissors.git
[ec2-user@ip-10-0-0-53 Awsdevops-sample-repo]$ ls -ltr
total 164
-rw-rw-r-- 1 ec2-user ec2-user 166724 Apr 22 06:43 rock-paper-scissors.git
[ec2-user@ip-10-0-0-53 Awsdevops-sample-repo]$ git Add -A
git: 'Add' is not a git command. See 'git --help'.

The most similar command is
    add
[ec2-user@ip-10-0-0-53 Awsdevops-sample-repo]$ git add -A
```

[ec2-user@ip-10-0-0-53 Awsdevops-sample-repo]\$ touch shiva.txt we are creating a file inside the directory here

[ec2-user@ip-10-0-0-53 Awsdevops-sample-repo]\$ ls  
rock-paper-scissors.git shiva.txt

[ec2-user@ip-10-0-0-53 Awsdevops-sample-repo]\$ git add shiva.txt

We are adding the file to the directory here

[ec2-user@ip-10-0-0-53 Awsdevops-sample-repo]\$ git commit -m " adding files"

We are committing the changes here

[master 02c49c5] adding files

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 shiva.txt

```
create mode 100644 shiva.txt
[ec2-user@ip-10-0-0-53 Awsdevops-sample-repo]$ git push
Username for 'https://git-codecommit.ap-south-1.amazonaws.com': Aravind-aws-devops-at-601007615373
Password for 'https://Aravind-aws-devops-at-601007615373@git-codecommit.ap-south-1.amazonaws.com':
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 29.19 KiB | 5.84 MiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0
To https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/Awsdevops-sample-repo
 * [new branch]      master -> master
```

Git push – this command helps us to push the files to the repository

In the above steps the files are pushed to a new branch known as “Master”

If you navigate to the aws console and check the empty repository we created, you will find its added with data. PFB..

The screenshot shows the AWS CodeCommit interface. On the left, there's a sidebar with 'Developer Tools' at the top, followed by 'CodeCommit'. Under 'Source > CodeCommit', several options are listed: 'Getting started', 'Repositories', 'Code' (which is highlighted in orange), 'Pull requests', 'Commits', 'Branches', 'Git tags', and 'Settings'. The main content area is titled 'Awsdevops-sample-repo'. Below it, there's a section titled 'Awsdevops-sample-repo' with a 'Info' link. A table lists two items: 'Name' with 'rock-paper-scissors.git' and 'shiva.txt'.

## AWS Codebuild

– Using this we can build codes without a necessity for a code build server

AWS CodeBuild is a fully managed continuous integration service that compiles source code, runs tests, and produces software packages that are ready to deploy. With CodeBuild, you don't need to provision, manage, and scale your own build servers.

Step – 1 – create a YAML file named buildspec.yml inside the Awsdevops-sample-repo directory.

[ec2-user@ip-10-0-0-53 Awsdevops-sample-repo]\$ sudo nano buildspec.yml **and type the below YAML sample code**

```
- hosts: ec2 #server host or group name
  sudo: yes
  tasks:
    - name: install apache2
      apt: name=apache2 update_cache=yes state=latest

    - name: enabled mod_rewrite
      apache2_module: name=rewrite state=present
      notify:
        - restart apache2

  handlers:
    - name: restart apache2
      service: name=apache2 state=restarted
```

---

[ec2-user@ip-10-0-0-53 Awsdevops-sample-repo]\$ git add buildspec.yml

[ec2-user@ip-10-0-0-53 Awsdevops-sample-repo]\$ git commit -m "adding buildspec.yml"

[ec2-user@ip-10-0-0-53 Awsdevops-sample-repo]\$ git push

[ec2-user@ip-10-0-0-53 Awsdevops-sample-repo]\$ ls

buildspec.yml rock-paper-scissors.git shiva.txt

Step : 2 – Navigate to your aws console and follow the below

Developer Tools > CodeBuild > Build projects > Create build project

## Create build project

### Project configuration

**Project name:** devops-codebuild  
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 \_.



### Source

Add source

Source 1 - Primary

Source provider: AWS CodeCommit

Repository: Awsdevops-sample-repo

Reference type: Branch  
Choose the source version reference type that contains your source code.  
 Branch  
 Git tag  
 Commit ID

Branch: master  
Choose a branch that contains the code to build.

Commit ID - optional  
Choose a commit ID. This can shorten the duration of your build.  
Search bar:

Source version [Info](#)  
refs/heads/master

#### Environment image

Managed image

Use an image managed by AWS CodeBuild

Custom image

Specify a Docker image

#### Operating system

Amazon Linux 2

 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/amazonlinux2-x86\_64-standard:3.0

#### Image version

Always use the latest image for this runtime version

#### 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

codebuild-devops-codebuild-service-role

Type your service role name

#### ► Additional configuration

Timeout, certificate, VPC, compute type, environment variables, file systems

## Buildspec

#### Build specifications

Use a buildspec file

Store build commands in a YAML-formatted buildspec file

Insert build commands

Store build commands as build project configuration

**Artifacts**

Add artifact

### Artifact 1 - Primary

Type

Amazon S3

You might choose no artifacts if you are running tests or pushing a Docker image to Amazon ECR.

Bucket name

aravindshivabucket

Name

The name of the folder or compressed file in the bucket that will contain your output artifacts. Use Artifacts packaging under Additional configuration to choose whether to use a folder or compressed file. If the name is not provided, defaults to project name.

If you have a bucket choose it in the above .. in my case the bucket name is aravindshivabucket... you don't have one ? well create one and choose it here

### Step – 3 – Click on create Build project and click on start build

Developer Tools > CodeBuild > Build projects > devops-codebuild

devops-codebuild

Notify ▾ Share Edit ▾ Delete build project Start build with overrides **Start build**

Configuration			
Source provider AWS CodeCommit	Primary repository Awsdevops-sample-repo	Artifacts upload location aravindshivabucket	Build badge Disabled
Public builds Disabled			

### Code Deploy

This feature can be used to deploy codes in EC2 instances, On-premise Instances and Lambda functions.

Step – 1 – Hope you are still Logged in as an IAM user, in aws console navigate to IAM role and create role

IAM > Roles

**Roles (11) 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.

**Create role**

Role name	Trusted entities	Last activity
...	...	...
...	...	...
...	...	...
...	...	...
...	...	...
...	...	...
...	...	...
...	...	...
...	...	...
...	...	...
...	...	...

## Step – 2 – Choose the AWS service as EC2

Step 1  
**Select trusted entity**

Step 2  
Add permissions

Step 3  
Name, review, and create

### Select trusted entity

#### Trusted entity type

- AWS service**  
Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account**  
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- Web identity**  
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

- SAML 2.0 federation**  
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- Custom trust policy**  
Create a custom trust policy to enable others to perform actions in this account.

#### Use case

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

#### Common use cases

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

Use cases for other AWS services:

Choose a service to view use case ▾

## Step – 3 – choose the permission as AWSCodeDeployFullAccess

IAM > Roles > Create role

Step 1  
**Select trusted entity**

Step 2  
**Add permissions**

Step 3  
Name, review, and create

### Add permissions

#### Permissions policies (Selected 1/752)

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

Filter policies by property or policy name and press enter

11 matches

"codedeploy" X Clear filters

Policy name	Type	Description
AmazonEC2RoleforAWSCodeDeploy	AWS m...	Provides EC2 access to S3 bucket to download revision. This role...
AWSCodeDeployRoleForECS	AWS m...	Provides CodeDeploy service wide access to perform an ECS blu...
AWSCodeDeployReadOnlyAccess	AWS m...	Provides read only access to CodeDeploy resources.
<input checked="" type="checkbox"/> AWSCodeDeployFullAccess	AWS m...	Provides full access to CodeDeploy resources.

Click next, give a name for the role and click ok/next.

Step 4 – We need to create another role , this time choose the trusted entity as AWS Service and Use case as CodeDeploy

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

Common use cases

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

Use cases for other AWS services:

CodeDeploy

- 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 behalf.

Click on next and choose the permission as [AWSCodeDeployRole](#)

We have created two roles as show in below snip

The screenshot shows the AWS IAM Roles page. At the top, there is a search bar and a filter section with "Role name" and "Trusted entities". Below this, a table lists four roles:

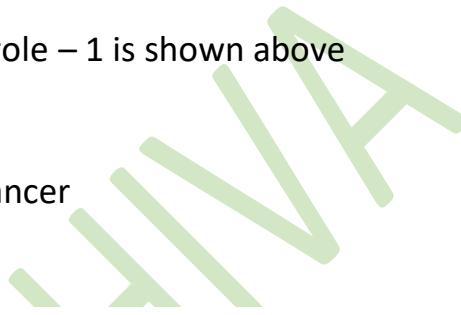
Role name	Trusted entities
AmazonAppStreamServiceAccess	AWS Service: appstream
ApplicationAutoScalingForAmazonAppStreamAccess	AWS Service: application-autoscaling
AWSDevops-Codedeploy-role2	AWS Service: codedeploy
AWSDevopscodedeployrole-1	AWS Service: ec2

In the AWSDevopscodedeployrole – 1 we created we need to add one more permission ie s3fullaccess.

Permissions		Trust relationships	Tags	Access Advisor	Revoke sessions				
		<span>Simulate</span> <span>Remove</span> <span>Add permissions ▾</span>							
<b>Permissions policies (2)</b> You can attach up to 10 managed policies.									
<input type="text"/> Filter policies by property or policy name and press enter									
Policy name ▾	Type	Description							
<input type="checkbox"/> AmazonS3FullAccess	AWS managed	Provides full access to all buckets via t...							
<input type="checkbox"/> AWSCodeDeployFullAccess	AWS managed	Provides full access to CodeDeploy res...							

The permission for AWSDevopscodedeployrole – 1 is shown above

## Step 5 – Lets create an Application Loadbalancer



Load balancer name  
Name must be unique within your AWS account and cannot be changed after the load balancer is created.  
 LoadbalancerforCodedeploytoacg

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme [Info](#)  
Scheme cannot be changed after the load balancer is created.  
 **Internet-facing**  
An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#)

Internal  
An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type [Info](#)  
Select the type of IP addresses that your subnets use.  
 **IPv4**  
Recommended for internal load balancers.  
 Dualstack  
Includes IPv4 and IPv6 addresses.

**Network mapping** [Info](#)  
The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC [Info](#)  
Select the virtual private cloud (VPC) for your targets. Only VPCs with an internet gateway are enabled for selection. The selected VPC cannot be changed after the load balancer is created. To confirm the VPC for your targets, view your [target groups](#).  
 AWSDevops VPC  
vpc-0fcfa59723c819fe3  
IPv4: 10.0.0.0/24

To create a Loadbalancer we need total two subnets, we created one.

I am going to create one more.

The screenshot shows the AWS VPC console interface. At the top, a table lists subnets: 'Codedeploy-PUB-Subnet-1' (selected, yellow background) and 'Codedeploy-PUB-Subnet-2'. Below the table, a message says 'You can now check network connectivity with Reachability Analyzer' with a 'Run Reachability Analyzer' button. A green checkmark icon is present on the right.

**Route table:** rtb-00ba5e58e7bc1b9f9 / AWS-Devops-PUB-SUBNET-Routetable

**Routes (2)**

Destination	Target
10.0.0.0/24	local
0.0.0.0/0	igw-0336fd99df0f71204

**VPC Info**  
Select the virtual private cloud (VPC) for your targets. Only VPCs with an Internet gateway are enabled for selection. The selected VPC cannot be changed after the load balancer is created. To confirm the VPC for your targets, view your target groups.

**AWSDevops VPC**  
vpc-0fc597223c819fe3  
IPv4: 10.0.0.0/24

**Mappings Info**  
Select at least one Availability Zone and one subnet for each zone. We recommend selecting at least two Availability Zones. The load balancer will route traffic only to targets in the selected Availability Zones. Zones that are not supported by the load balancer or VPC cannot be selected. Subnets can be added, but not removed, once a load balancer is created.

**ap-south-1a**  
Subnet: subnet-0b126ca1f1fdb34e1

**IPv4 settings**  
Assigned by AWS

**ap-south-1b**  
Subnet: subnet-06610fdf4e65a28c3

**Basic details**

Security group name: **Info**  
SecuritygroupforLoadBalancer

Description: **Info**  
Allows SSH access to developers

VPC: **Info**  
vpc-0e8cef2844fd37145

**Inbound rules** **Info**

Type <b>Info</b>	Protocol <b>Info</b>	Port range <b>Info</b>	Source <b>Info</b>	Description - optional <b>Info</b>
Custom TCP	TCP	80	My IP	Aravind-IP
183.83.187.154/32 X				

## Security groups Info

A security group is a set of firewall rules that control the traffic to your load balancer.

### Security groups

Select security groups

Create new security group

default sg-039ed8c94304d5de6 X  
VPC: vpc-0fca597223c819fe3

AWS Devops SG -1 sg-0d6aaf053b67fc39d X  
VPC: vpc-0fca597223c819fe3



## Listeners and routing Info

A listener is a process that checks for connection requests, using the protocol and port you configure. Traffic received by the listener is then routed per your specification. You can specify multiple rules and multiple certificates per listener after the load balancer is created.

### ▼ Listener HTTP:80

Remove

Protocol

HTTP

Port

80

Default action Info

Forward to Select a target group

Create target group



We need to create a target group

EC2 > Target groups > Create target group

Step 1  
**Specify group details**

## Specify group details

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

Step 2  
Register targets

### Basic configuration

Settings in this section cannot be changed after the target group is created.

#### Choose a target type

Instances

- Supports load balancing to instances within a specific VPC.
- Facilitates the use of [Amazon EC2 Auto Scaling](#) to manage and scale your EC2 capacity.

**Target group name**

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol	Port
HTTP	: 80

**VPC**

Select the VPC with the instances that you want to include in the target group.

AWSDevops VPC vpc-0fca597223c819fe3 IPv4: 10.0.0.0/24	▼
-------------------------------------------------------------	---

**Protocol version**

**HTTP1**  
Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.

**HTTP2**  
Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.

**gRPC**  
Send requests to targets using gRPC. Supported when the request protocol is gRPC.

---

**Advanced health check settings**

**Port**  
The port the load balancer uses when performing health checks on targets. The default is the port on which each target receives traffic from the load balancer, but you can specify a different port.

**Traffic port**

**Override**

**Healthy threshold**  
The number of consecutive health checks successes required before considering an unhealthy target healthy.

2-10

**Unhealthy threshold**  
The number of consecutive health check failures required before considering a target unhealthy.

2-10

**Timeout**  
The amount of time, in seconds, during which no response means a failed health check.

 seconds
 

2-120

**Interval**  
The approximate amount of time between health checks of an individual target

 seconds
 

5-300

**Success codes**  
The HTTP codes to use when checking for a successful response from a target. You can specify multiple values (for example, "200,202") or a range of values (for example, "200-299").

Click on create ..

Step -7- We need an autoscalinggroup to see how Codedeploy service deploys an application prior to this we need to install Code-Deploy agent in our ec2-instatnce

So execute the following commands in the EC2 instance we created earlier

Follow the below steps

. Click on EC2- and then Launch an EC2 instance, choose an AMI , in my case it's an Amazon Linux AMI, then select the instance type as t2-micro and then configure the instance. Choose the number of instance of 5 and choose the other fields and Click on Launch into Auto scaling group

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot Instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances  Launch into Auto Scaling Group

You may want to consider launching these instances into an Auto Scaling Group to help you maintain application availability and for easy scaling in the future. [Learn how Auto Scaling can help your application stay healthy and cost effective.](#)

Purchasing option  Request Spot Instances

Network  Create new VPC

Subnet  Create new subnet  
121 IP Addresses available

Auto-assign Public IP

Hostname type

DNS Hostname  Enable IP name IPv4 (A record) DNS requests  
 Enable resource-based IPv4 (A record) DNS requests  
 Enable resource-based IPv6 (AAAA record) DNS requests

Placement group  Add instance to placement group

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Storage](#)

Go back to EC2 instance

In the terminal type –

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

```
sudo yum install ruby -y
```

```
cd /home/ec2-user
```

```
aws s3 cp s3://aws-codedeploy-ap-south-1/latest/install . --region ap-south-1
```

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

```
sudo ./install auto
```

```
[ec2-user@ip-10-0-0-53 ~]$ sudo service codedeploy-agent status
```

The AWS CodeDeploy agent is running as PID 2053

## Create an AMI for the EC2 Instance



The screenshot shows the AWS CloudFormation console with a stack named "ASGCodeDeploy\_Linux". The "Outputs" tab is selected, displaying the output "ASGCodeDeploy\_Linux" which points to the ARN: arn:aws:cloudformation:ap-south-1:123456789012:stack/ASGCodeDeploy\_Linux/1234567890123456.

**Instances (1/1) Info**

Name	Instance ID	Instance state	Instance type	Status check	Alarm
AWSDevopsEC2	i-0d0b2b2d4008c829f	Running	t2.micro	2/2 checks passed	No alarm

**Actions ▾**

- Connect
- View details
- Manage instance state
- Instance settings
- Networking
- Security
- Image and templates
- Monitor and troubleshoot
- Launch more like this

**Instance: i-0d0b2b2d4008c829f (AWSDevopsEC2)**

**Instance ID:** i-0d0b2b2d4008c829f (AWSDevopsEC2)

**Image name:** ASGCodeDeploy\_Linux

**Image description - optional:** ASGCodeDeploy\_Linux - April22-2022

**No reboot:**

**Instance volumes:**

Volume type	Device	Snapshot	Size	Volume type	IOPS	Throughput	Delete on termination	Encrypted
EBS	/dev/x...	Create new snapshot fr...	8	EBS General Purpose S5...	100		<input checked="" type="checkbox"/> Enable	<input type="checkbox"/> Enable

**Add volume**

Name	Source	Status	Creation date
a... 601007615373/ASGCodeDeploy_Linux	6...	P... Available	2022/04/22 15:34 GMT+5:30

Navigation to the Launch configuration and choose the AMI that you created under My AMI

EC2 > Launch configurations > Create launch configuration

### Create launch configuration

**Launch configuration name**

Name  
ACGforCodeDeployALB

**Amazon machine image (AMI)**

AMI  
ASGCodeDeploy\_Linux

**Instance type**

Instance type  
Choose instance type

**Instance type**

t2.micro (1 vCPUs, 1 GiB, EBS Only) [Choose instance type](#)

**Additional configuration - optional**

Purchasing option [Info](#)  
 Request Spot Instances

**IAM instance profile** [Info](#)

AWSDevopscodedeployrole-1

Create a new security group  
 Select an existing security group

**Security groups**

Security group ID	Name	VPC ID	Description
sg-039ed8c94304d5de6	default	vpc-0fca597223c819fe3	default VPC security group
sg-0b7426aa77ca08cea	SecuritygroupforLoadBalancer	vpc-0e8cef2844fd37145	Allowaccesstocodedeploy
sg-0d6aaf053b67fc39d	AWS Devops SG -1	vpc-0fca597223c819fe3	AWS Devops SG -1
sg-0e9748fdf7e6ca9e0	default	vpc-0e8cef2844fd37145	default VPC security group

**Key pair (login)** [Info](#)

Key pair options

Choose an existing key pair

Existing key pair

AWSDevopsapril22-2022

I acknowledge that I have access to the selected private key file (AWSDevopsapril22-2022.pem), and that without this file I will not be able to log in to my instances.

Create an autoscaling group

**configuration**

Step 2  
**Choose instance launch options**

Step 3 (optional)  
Configure advanced options

Step 4 (optional)  
Configure group size and scaling policies

Step 5 (optional)  
Add notifications

Step 6 (optional)  
Add tags

Step 7  
Review

Choose the VPC network environment that your instances are launched into, and customize the instance types and purchase options.

**Network** Info

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC  
Choose the VPC that defines the virtual network for your Auto Scaling group.

ap-south-1a | subnet-0b126ca1f1fdb34e1  
(AWSDevops-PublicSubnet-LinuxEC2-ap-south-1)  
10.0.0.0/25

ap-south-1b | subnet-06610fdf4e65a28c3  
(Codedeploy-PUB-Subnet-2)  
10.0.0.128/25

Select Availability Zones and subnets ▾

can use in the chosen VPC.

ap-south-1a | subnet-0b126ca1f1fdb34e1 X  
(AWSDevops-PublicSubnet-LinuxEC2-ap-south-1)  
10.0.0.0/25

ap-south-1b | subnet-06610fdf4e65a28c3 X  
(Codedeploy-PUB-Subnet-2)  
10.0.0.128/25

Create a subnet

**Cancel** **Previous** **Skip to review** **Next**

**configuration**

Step 2  
Choose instance launch options

Step 3 (optional)  
Configure advanced options

Step 4 (optional)  
Configure group size and scaling policies

Step 5 (optional)  
Add notifications

Step 6 (optional)  
Add tags

Step 7  
Review

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group. If you currently use launch configurations, you might consider migrating to launch templates.

**Name**

Auto Scaling group name  
Enter a name to identify the group.  
**ASG-CodeDeploy**

Must be unique to this account in the current Region and no more than 255 characters.

**Launch configuration** Info **Switch to launch template**

Launch configuration  
Choose a launch configuration that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

ACGforCodeDeployALB

**Create a launch configuration** Launch configuration ACGforCodeDeployALB AMI ID ami-081d602a338ca362a Date created Fri Apr 22 2022 16:08:49 GMT+0530 (India Standard Time)

Security groups sg-0d6aa0f053b67fc39d Instance type t2.micro Key pair name AWSDevopsapril22-2022

**Cancel** **Next**

Servic... Search for services, features, blogs, docs, and more [Alt+S]

Choose launch template or configuration

Step 2 Choose instance launch options

Step 3 (optional) Configure advanced options

Step 4 (optional) Configure group size and scaling policies

Step 5 (optional) Add notifications

Step 6 (optional) Add tags

Step 7 Review

**Load balancing - optional** Info

Choose a load balancer to distribute incoming traffic for your application across instances to make it more reliable and easily scalable. You can also set options that give you more control over health check replacements and monitoring.

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

No load balancer Traffic to your Auto Scaling group will not be fronted by a load balancer.

Attach to an existing load balancer Choose from your existing load balancers.

Attach to a new load balancer Quickly create a basic load balancer to attach to your Auto Scaling group.

**Attach to an existing load balancer**

Select the load balancers that you want to attach to your Auto Scaling group.

Choose from your load balancer target groups This option allows you to attach Application, Network, or Gateway Load Balancers.

Choose from Classic Load Balancers

Existing load balancer target groups Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

Select target groups

TGforALB | HTTP Application Load Balancer: LoadbalancerforCodedeploytoacg

Click on next until you reach the below screen

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1 Choose launch template or configuration

Step 2 Choose instance launch options

Step 3 (optional) Configure advanced options

Step 4 (optional) Configure group size and scaling policies

Step 5 (optional) Add notifications

Step 6 (optional) Add tags

Step 7 Review

**Add tags** Info

Add tags to help you search, filter, and track your Auto Scaling group across AWS. You can also choose to automatically add these tags to instances when they are launched.

**Tags (1)**

Key Value - optional Tag new instances

site 80  Remove

Add tag 49 remaining

Cancel Previous Next

Click on next and create the auto scaling group

## Heading back to the Code Deploy

The screenshot shows the 'Create application' page in the AWS CodeDeploy console. The application name is set to 'CodeDeployLinuxApplication' and the compute platform is set to 'EC2/On-premises'. The 'Create application' button is highlighted with a yellow box.

Next step is to create a deployment group

The screenshot shows the 'CodeDeployLinuxApplication' page in the AWS CodeDeploy console. A green banner at the top indicates that the application has been created. The 'Deployment groups' section is displayed, showing a table with columns for Name, Status, Last attempted deployment, and Last successful deployment. The 'Create deployment group' button is highlighted with a yellow box.

Enter a deployment group name  
 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.

**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.

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.

**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**  
1 unique matched instance. [Click here for details](#)

You can select up to 10 Amazon EC2 Auto Scaling groups to deploy your application revision to.

**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

Application Load Balancer or Network Load Balancer

Classic Load Balancer

Choose a target group

TGforALB ▾

▶ Advanced - optional

Cancel [Create deployment group](#)

Let's create a deployment now

Developer Tools [X](#)

**CodeDeploy**

- Source [CodeCommit](#)
- Artifacts [CodeArtifact](#)
- Build [CodeBuild](#)
- Deploy [CodeDeploy](#)
  - Getting started
  - Deployments
  - Applications
    - Application**
    - Settings
  - Deployment configurations
- On-premises instances

**CodDeploymentgroup**

Success Deployment group created

Developer Tools > CodeDeploy > Applications > CodeDeployLinuxApplication > CodDeploymentgroup

[Edit](#) [Delete](#) [Create deployment](#)

**Deployment group details**

Deployment group name	CodeDeploymentgroup	Application name	CodeDeployLinuxApplication	Compute platform	EC2/On-premises
Deployment type	In-place	Service role ARN	arn:aws:iam::601007615373:role/AWSDevops-Codedeploy-role2	Deployment configuration	CodeDeployDefault.AllAtOnce
Rollback enabled		Agent update scheduler			

Prior to this in the S3 bucket which we chose earlier, upload an image zip file

The screenshot shows the Amazon S3 console interface. On the left, there's a sidebar with options like Buckets, Storage Lens, and Feature spotlight. The main area shows a bucket named 'aravindshivabucket'. Under the 'Objects' tab, there is one object listed: 'Shivaapril22-2022'. The object details show it was uploaded on April 22, 2022, at 16:36:10 (UTC+05:30) and has a size of 97.7 KB.

Heading back to Code Deploy to create a Deployment

The screenshot shows the 'Create deployment' wizard in the AWS CodeDeploy console. It's on the 'Deployment settings' step. The configuration includes:

- Application: CodeDeployLinuxApplication
- Deployment group: CodDeploymentgroup
- Compute platform: EC2/On-premises
- Deployment type: In-place
- Revision type: My application is stored in Amazon S3 (selected)
- Revision location: s3://aravindshivabucket/Shivaapril22-2022
- Revision file type: .zip

Click on next and create deployment

The screenshot shows the 'Success' status page for a deployment named 'd-DP8076W0G'. The deployment status is shown as 'In progress' with 0% completion. There are buttons for 'Stop deployment' and 'Stop and roll back deployment'.

Thus the CodeDeploy deploys the code in all the autoscaling instances, We saw how the CodeDeploy works with Autoscaling instances along with an Application LoadBalancer.

## CodePipeline

Navigate to CodePipeline in AWS Console and click on create Pipeline

Step 3  
Add build stage

Step 4  
Add deploy stage

Step 5  
Review

Enter the pipeline name. You cannot edit the pipeline name after it is created.  
Codepipeline-April22-2022  
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-ap-south-1-Codepipeline-April22-2022

Type your service role name

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

▼ Advanced settings

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.

Click on Next

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 system, build spec file, and output file names.

Cancel Previous **Skip build stage** Next

## Choose the things which we created earlier for the below phase

Choose pipeline settings

Add deploy stage info

You cannot skip this stage  
Pipelines must have at least two stages. Your second stage must be either a build or deployment stage. Choose a provider for either the build stage or deployment stage.

**Deploy**

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

AWS CodeDeploy

Region  
Asia Pacific (Mumbai)

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.

CodeDeployLinuxApplication

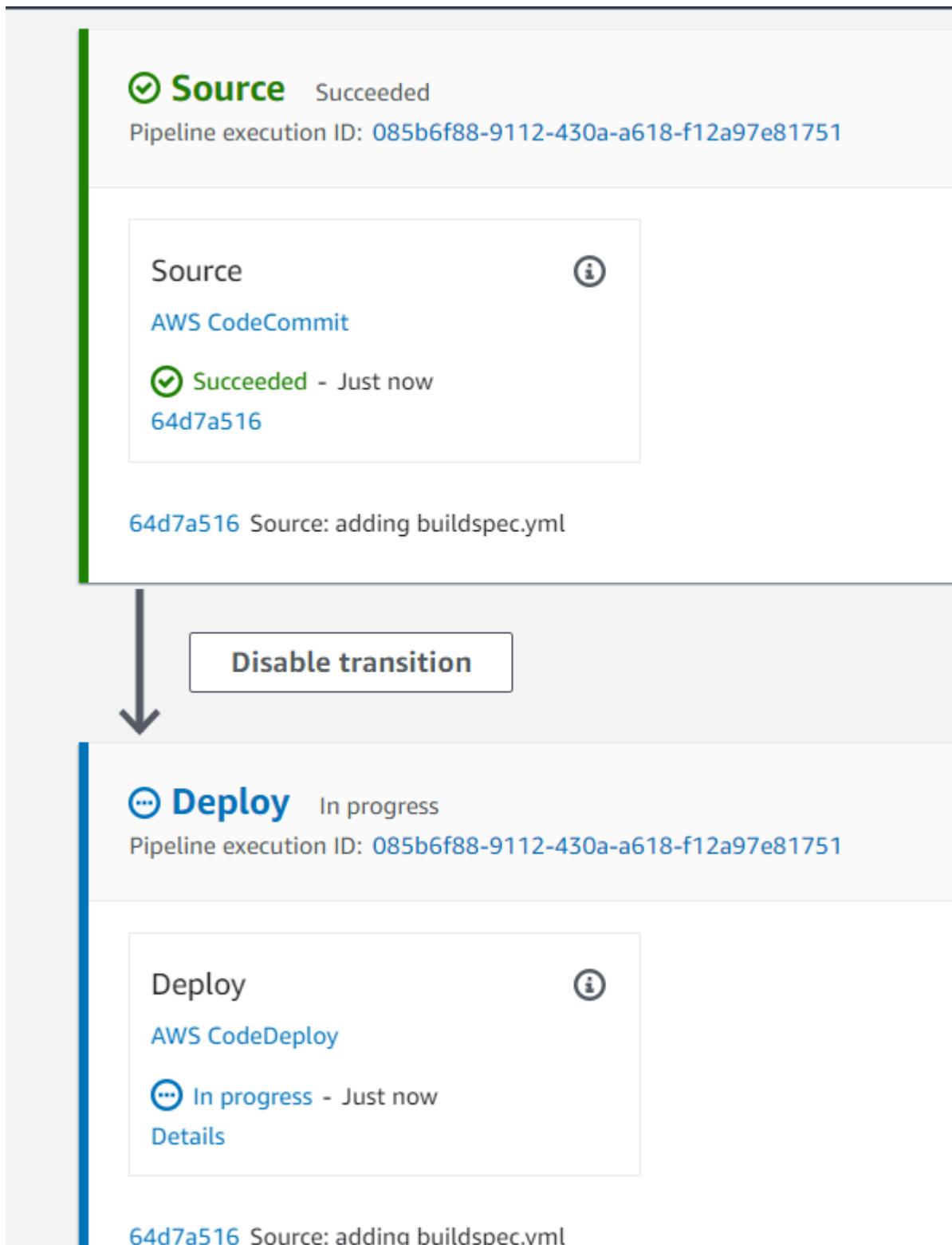
Deployment group  
Choose a deployment group that you have already created in the AWS CodeDeploy console. Or create a deployment group in the AWS CodeDeploy console and then return to this task.

CodDeploymentgroup

Cancel Previous Next

Click on Next and click on create Pipeline

Aravind.SR



**Once the above step is completed, you can check the pipeline working by navigating to the EC2 terminal...create a new file and do a git add "filename that you added"**

**Git commit “with some information for your understanding”**

**Git push**

**Once this is done the CodeDeploy ..Deploys the code automatically in EC2 instances... Whatever changes you make using Git commit then without manual intervention the loop of CodePipeline fuctions**

**CodeCommit-CodeDeploy**

Aravind.SHIVA