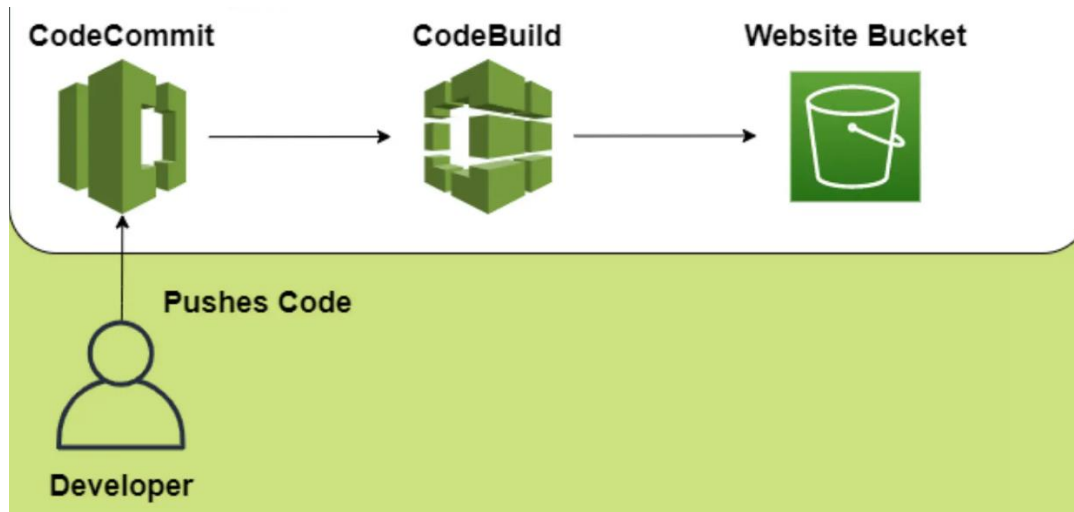


Step-by-Step Guide to DevOps Implementation on AWS – Jasvitha Buggana

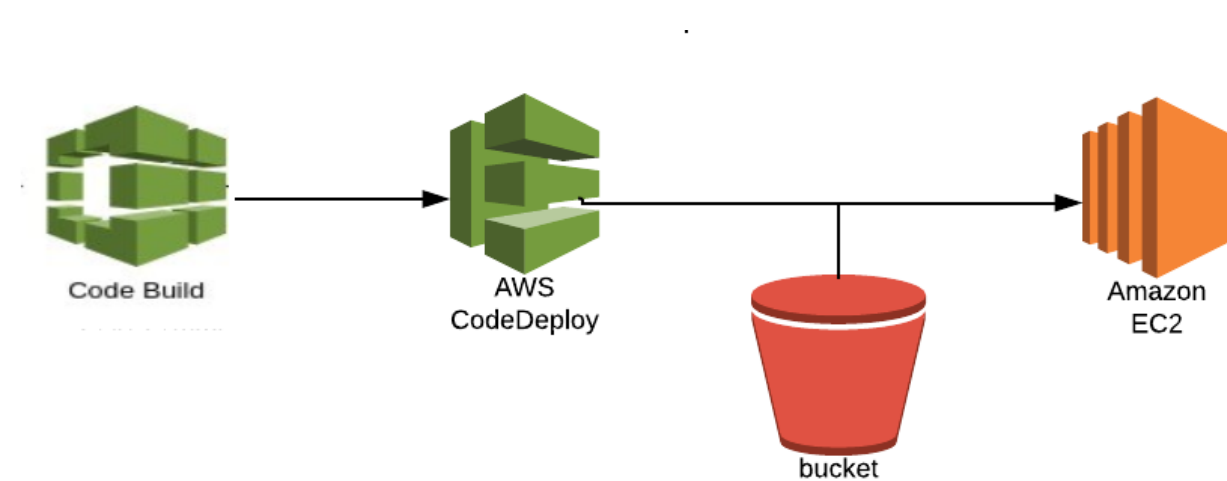
Contents

1.Setting up the AWS and Git environment:	3
i)Create a new repository on AWS:	3
ii)Clone the repository:	5
iii)Add files and commit changes:	6
2.Code Build:	7
ii)Define Build Specifications:	13
3.Start A Build:	15
i)Build Environment:	16
4.Creating EC2 Instance & Set-Up Agent:	16
i)Create an EC2 Instance:	16
ii)Setting Up CodeDeploy Agent in EC2:	19
5.CodeDeploy:	20
i) Prepare your application:	20
ii) Deployment Group:	21
iii) Service Role:	22
6.Create Deployment:	26
i)EC2-ROLE:	27
ii)Attach to EC2 instance:	28
iii)Adding App Specification File:	29
iv)Create Required Files:	29
v)Start the Build:	30
vi)Deploy your application:	31
7. Creating a Complete CI/CD Pipeline with CodePipeline:	32
i)Create a pipeline:	32
ii)Pipeline Execution:	35

Implementing DevOps Practices with AWS Infrastructure

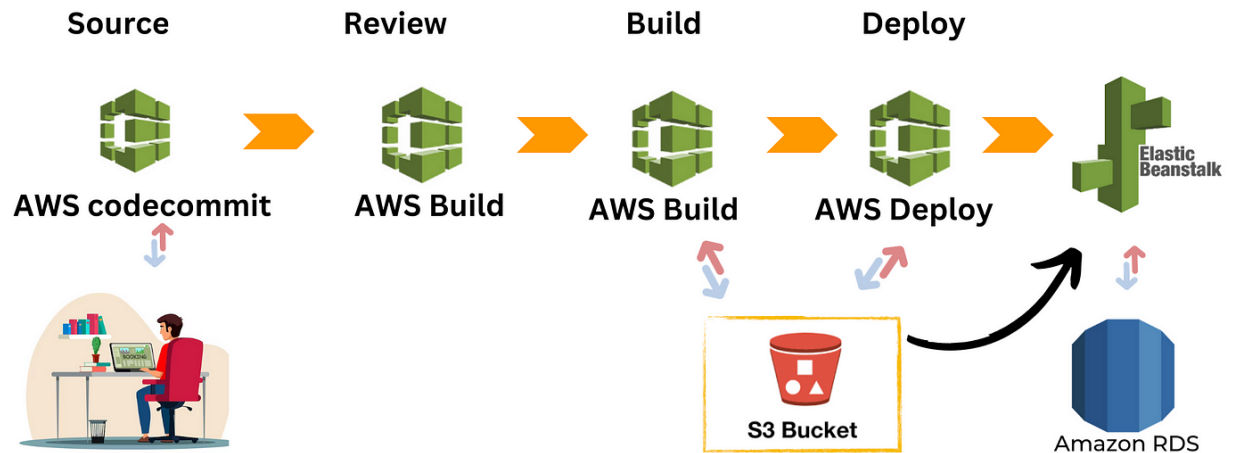


Project flow: A developer pushes code to a website stored in an **AWS CodeCommit** repository. Subsequently, **AWS CodeBuild** automatically builds the code and deploys it to an S3 bucket.



In the second stage, the code built and tested in **CodeBuild** is uploaded to an Amazon S3 bucket, which can be configured to function as a website. This means that users can access the website by visiting the S3 bucket's URL.

AWS Code Pipeline CI/CD Project



After the code was built and tested in **CodeBuild**. This diagram shows how the built code is shipped to production. **AWS CodeDeploy** takes the tested code from CodeBuild and deploys it to an Elastic Beanstalk environment. Elastic Beanstalk is a service that lets you easily deploy and manage web applications in the cloud.

1. Setting up the AWS and Git environment:

i) Create a new repository on AWS:

- Log in to the AWS Management Console.
- Navigate to the **CodeCommit** service.
- Click on '**Create repository**' and follow the prompts to set up your repository.

Create repository

Create a secure repository to store and share your code. Begin by typing a repository name and a description for your repository. Repository names are included in the URLs for that repository.

Repository settings

Repository name
Devops-on-AWS
100 characters maximum. Other limits apply.

Description - optional
This repository serves as the platform for implementing DevOps practices within an AWS environment.
1,000 characters maximum

Tags
Add tag

Additional configuration
AWS KMS key

☐ Enable Amazon CodeGuru Reviewer for Java and Python - optional
Get recommendations to improve the quality of the Java and Python code for all pull requests in this repository.
A service-linked role will be created in IAM on your behalf if it does not exist.

Cancel Create

- Create an **IAM user** for this repo (Devops-on-AWS) because it can't be logged in with the root user.

Users (6) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

Search

Create user

Specify user details

User details

User name
code-commit-repo
The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

☒ Provide user access to the AWS Management Console - optional
If you're providing console access to a person, it's a **best practice** to manage their access in IAM Identity Center.

Are you providing console access to a person?

User type

☐ Specify a user in Identity Center - Recommended
We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.

☒ I want to create an IAM user
We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keyspaces, or a backup credential for emergency account access.

- Add the following permissions to this user.

Permissions policies (2)

Permissions are defined by policies attached to the user directly or through groups.



Remove



Add permissions ▼

Search

Filter by Type

All types ▼

< 1 > ⚙

<input type="checkbox"/>	Policy name ↗	Type	Attached via ↗
<input type="checkbox"/>	 AWSCodeCommitPowerUser	AWS managed	Directly
<input type="checkbox"/>	 IAMUserChangePassword	AWS managed	Directly

- Generate credentials for this user. Navigate to the section 'security credentials' > HTTPS Git credentials for **AWS CodeCommit**.

HTTPS Git credentials for AWS CodeCommit (0)

Generate a user name and password you can use to authenticate HTTPS connections to AWS CodeCommit repositories. You can have a maximum of 2 sets of credentials (active or inactive) at a time. [Learn more](#)

Actions ▼

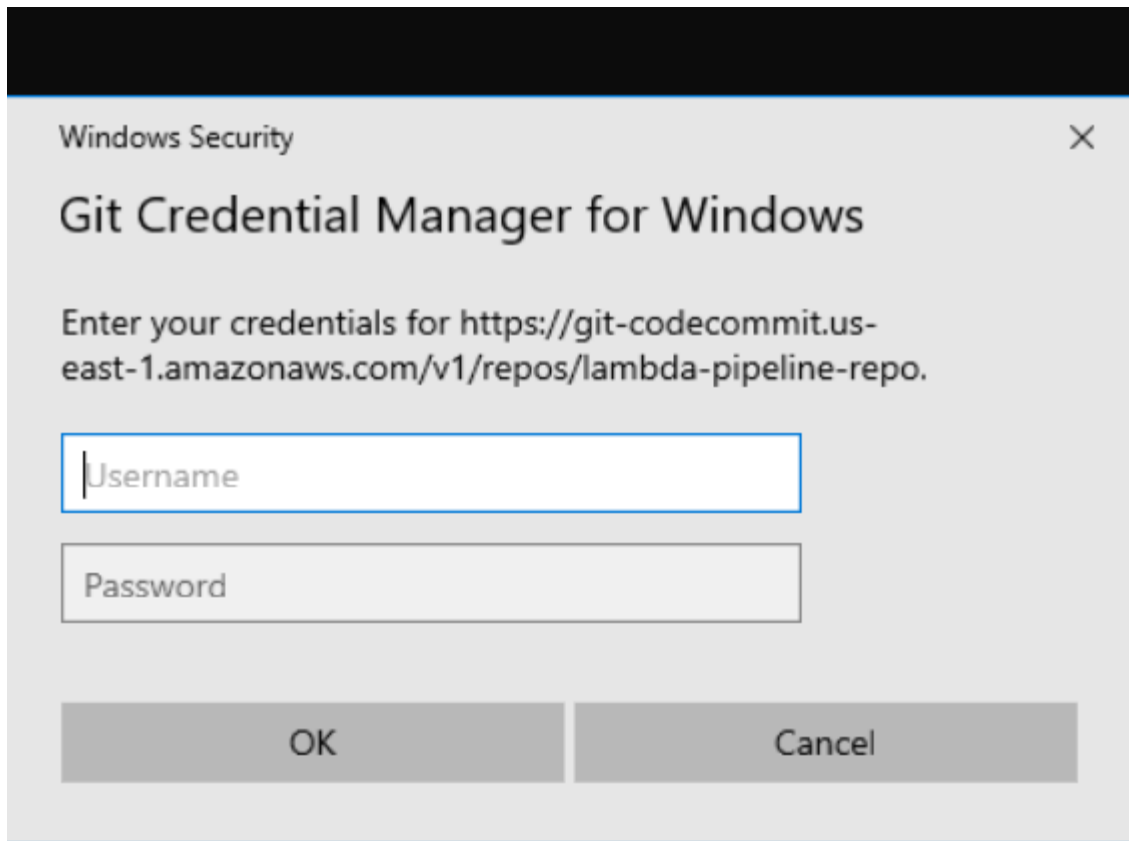
Generate credentials

User name	Created	Status
No credentials		
<div>Generate credentials</div>		

- Download the credentials after creating it from here.

ii) Clone the repository:

- Once the repository is created, clone it to your local machine using Git.
- Command: **git clone <repo-url>**
- Note: Git should be installed on your local machine.
- It will ask for credentials; copy the credentials we have downloaded for user **code-commit-repo**.



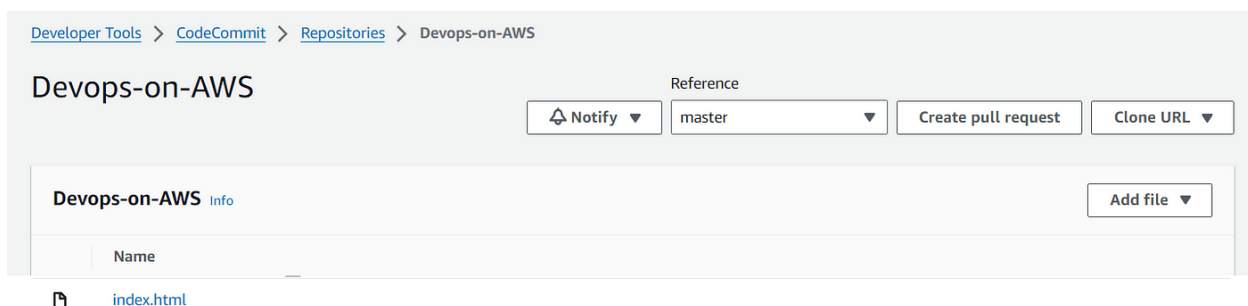
iii) Add files and commit changes:

- Add the first project files and commit the changes. We have an **index.html** file here.

```
cd Devops-on-AWS/  
vim index.html
```

```
git add .  
git commit -m "index.html added"  
git push origin master
```

- You can observe the file pushed to the AWS repository.



2. Code Build: CodeBuild streamlines your development process by compiling your source code, executing unit tests, and generating deployable artifacts. It removes the necessity of setting up, maintaining, and scaling your build servers.

i) Create a build project:

- Navigate to the **CodeBuild** service in the AWS Management Console.
- Click on ‘Create build project’.
- Configure the build settings including source provider (CodeCommit), environment, and buildspec file.

Project configuration

Project name

code-build-for-devops

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 _.

Public build access - *optional*

Public build access allows you to make the build results, including logs and artifacts, for this project available for the general public.

☐ Enable public build access

► Additional configuration

tags

Source

Add source

Source 1 - Primary

Source provider

AWS CodeCommit

Repository

Devops-on-AWS

Reference type

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

☒ Branch

☐ Git tag

☐ Commit ID

Branch

Choose a branch that contains the code to build.

master

Commit ID - optional

Choose a commit ID. This can shorten the duration of your build.

Source version [Info](#)

refs/heads/master

83d634d5 index.html updated

Additional configuration

Git clone depth, Git submodules

- Select the **Provisioning model** as shown.

Environment

Provisioning model [Info](#)

☒ On-demand

Automatically provision build infrastructure in response to new builds.

☐ Reserved capacity

Use a dedicated fleet of instances for builds. A fleet's compute and environment type will be used for the project.

Environment image

☒ Managed image

Use an image managed by AWS CodeBuild

☐ Custom image

Specify a Docker image

Compute

☒ EC2

Optimized for flexibility during action runs

☐ Lambda

Optimized for speed and minimizes the start up time of workflow actions

Operating system

Ubuntu

Runtime(s)

Standard

Image

aws/codebuild/standard:7.0

Image version

Always use the latest image for this runtime version ▼

☐ Use GPU-enhanced compute

► Additional configuration

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

Buildspec

Current buildspec

Using the buildspec.yml in the source code root directory

Build specifications

☐ Insert build commands
Store build commands as build project configuration

☒ Use a buildspec file
Store build commands in a YAML-formatted buildspec file

Buildspec name - *optional*

By default, CodeBuild looks for a file named buildspec.yml in the source code root directory. If your buildspec file uses a different name or location, enter its path from the source root here (for example, buildspec-two.yml or configuration/buildspec.yml).

buildspec.yml

- Now for the next section, which is artifacts, you must create an S3 **bucket**.
- Go to the Amazon S3 dashboard and click on ‘**Create Bucket**’.

Create bucket [Info](#)

Buckets are containers for data stored in S3.

General configuration

AWS Region

US West (Oregon) us-west-2

Bucket type [Info](#)

☒ General purpose

Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

☐ Directory - *New*

Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name [Info](#)

devops-on-aws-artifact

Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

[Choose bucket](#)

Format: s3://bucket/prefix

- Click on **ACL enabled** and unclick **Block all public access** to avoid any kind of errors.

Object Ownership [Info](#)

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

☐ ACLs disabled (recommended)

All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

☒ ACLs enabled

Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

⚠ We recommend disabling ACLs, unless you need to control access for each object individually or to have the object writer own the data they upload. Using a bucket policy instead of ACLs to share data with users outside of your account simplifies permissions management and auditing.

Object Ownership

☒ Bucket owner preferred

If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.

☐ Object writer

The object writer remains the object owner.

i If you want to enforce object ownership for new objects only, your bucket policy must specify that the bucket-owner-full-control canned ACL is required for object uploads. [Learn more](#)

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

☐ Block all public access

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- ☐ **Block public access to buckets and objects granted through *new* access control lists (ACLs)**
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- ☐ **Block public access to buckets and objects granted through *any* access control lists (ACLs)**
S3 will ignore all ACLs that grant public access to buckets and objects.
- ☐ **Block public access to buckets and objects granted through *new* public bucket or access point policies**
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- ☐ **Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.



Turning off block all public access might result in this bucket and the objects within becoming public

AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

- ☒ I acknowledge that the current settings might result in this bucket and the objects within becoming public.

- After creating the bucket, create one folder inside it where **codebuild** will store the build files.

[Amazon S3](#) > [Buckets](#) > devops-on-aws-artifact

devops-on-aws-artifact [Info](#)

[Objects](#) | [Properties](#) | [Permissions](#) | [Metrics](#) | [Management](#) | [Access Points](#)

Objects (3) [Info](#)

Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Folder


Folder name

/

Folder names can't contain "/" . [See rules for naming](#) 

Server-side encryption [Info](#)

Server-side encryption protects data at rest.

 The following encryption settings apply only to the folder object and not to sub-folder objects.


Server-side encryption

☒ Do not specify an encryption key

The bucket settings for default encryption are used to encrypt the folder object when storing it in Amazon S3.

☐ Specify an encryption key

The specified encryption key is used to encrypt the folder object before storing it in Amazon S3.

 If your bucket policy requires objects to be encrypted with a specific encryption key, you must specify the same encryption key when you create a folder. Otherwise, folder creation will fail.

Cancel

Create folder

- Now add this bucket information into the **Artifacts** section in the code build configuration.

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

devops-on-aws-artifact

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.

build-code

☐ Enable semantic versioning

Use the artifact name specified in the buildspec file

Path - *optional*

The path to the build output ZIP file or folder.

build-code/artifact.zip

Example: MyPath/MyArtifact.zip.

Namespace type - *optional*

None

Choose Build ID to insert the build ID into the path to the build output ZIP file or folder, e.g. MyPath/MyBuildID/MyArtifact.zip. Otherwise, choose None.

Artifacts packaging

☐ None

The artifact files will be uploaded to the bucket.

☒ Zip

AWS CodeBuild will upload artifacts into a compressed file that is put into the specified bucket.

☐ Disable artifact encryption

Disable encryption if using the artifact to publish a static website or sharing content with others

► Additional configuration

Cache, encryption key

ii) Define Build Specifications:

A buildspec serves as the blueprint for CodeBuild to execute your build process effectively. It's crafted in YAML format and comprises essential build commands and configurations. Devoid of a build spec, CodeBuild lacks the instructions necessary to transform your input into output and identify the resultant artifact in the build environment for uploading to your designated output bucket.

- Create a buildspec.yml file in the root of your CodeCommit repository.

- Define the build phases such as install, pre-build, build, and post-build commands in the buildspec.yml.

```
MINGW64 ~/Documents/Devops-AWS/Devops-on-AWS (master)
```

- Command: `$ vim buildspec.yml` (In Git Bash, the ``$vim`` command launches the Vim text editor. Vim is a powerful and highly configurable text editor that operates within the terminal. When you run ``$vim`` in Git Bash, it opens the Vim editor, allowing you to create, edit, and manipulate text files directly from the command line.)

In git bash:

```
version: 0.2

phases:
  install:
    commands:
      - echo Installing NGINX
      - sudo apt-get update
      - sudo apt-get install nginx -y

  build:
    commands:
      - echo Build started on `date`
      - cp index.html /var/www/html

  post_build:
    commands:
      - echo Configuring NGINX

artifacts:
  files:
    - '**/*'
```

- After creating this, add the file, commit the changes, and push it into your code **commit** repo.

```
git add .
git commit -m "adding buildspec file"
git push
vim appsec.yml
```

Devops-on-AWS

Reference



master

Create pull request

Clone URL

Devops-on-AWS [Info](#)

Add file

Name



buildspec.yml



index.html

3. Start A Build:

- After uploading your **buildspec.yml** file, start your build.

code-build-for-devops

Actions

Create trigger

Edit

Debug build

Start build with overrides

Start build

Configuration

Source provider
AWS CodeCommit

Primary repository
Devops-on-AWS

Artifacts upload location
devops-on-aws-artifact

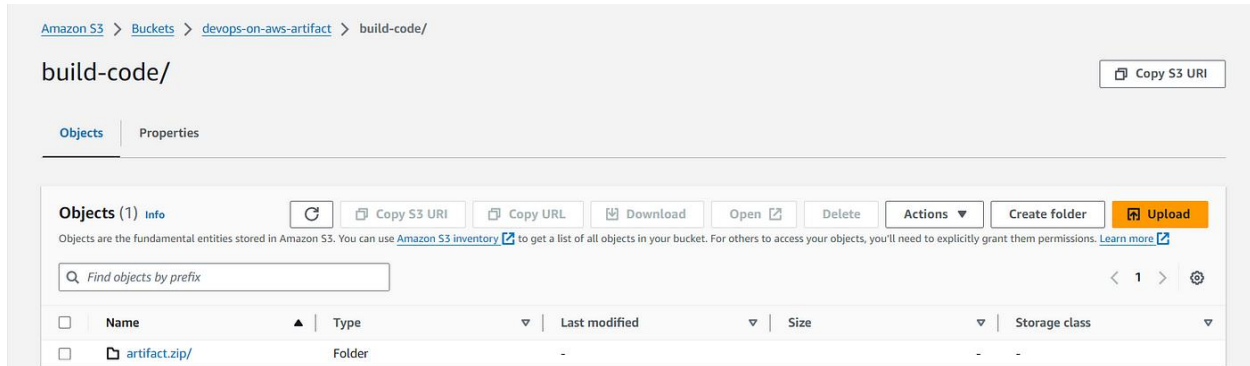
Service role
arn:aws:iam::084298470880:role/service-role/codebuild-code-build-for-devops-service-role

Public builds
Disabled

- You can trace your build in the **‘Phase details’** section of your project.

Name	Status	Context	Duration	Start time	End time
SUBMITTED	✓ Succeeded	-	<1 sec	Apr 1, 2024 1:57 AM (UTC+5:30)	Apr 1, 2024 1:57 AM (UTC+5:30)
QUEUED	✓ Succeeded	-	<1 sec	Apr 1, 2024 1:57 AM (UTC+5:30)	Apr 1, 2024 1:57 AM (UTC+5:30)
PROVISIONING	✓ Succeeded	-	4 secs	Apr 1, 2024 1:57 AM (UTC+5:30)	Apr 1, 2024 1:57 AM (UTC+5:30)
DOWNLOAD_SOURCE	✓ Succeeded	-	1 sec	Apr 1, 2024 1:57 AM (UTC+5:30)	Apr 1, 2024 1:57 AM (UTC+5:30)
INSTALL	✓ Succeeded	-	29 secs	Apr 1, 2024 1:57 AM (UTC+5:30)	Apr 1, 2024 1:57 AM (UTC+5:30)
PRE_BUILD	✓ Succeeded	-	<1 sec	Apr 1, 2024 1:57 AM (UTC+5:30)	Apr 1, 2024 1:57 AM (UTC+5:30)
BUILD	✓ Succeeded	-	<1 sec	Apr 1, 2024 1:57 AM (UTC+5:30)	Apr 1, 2024 1:57 AM (UTC+5:30)
POST_BUILD	✓ Succeeded	-	<1 sec	Apr 1, 2024 1:57 AM (UTC+5:30)	Apr 1, 2024 1:57 AM (UTC+5:30)
UPLOAD_ARTIFACTS	✓ Succeeded	-	<1 sec	Apr 1, 2024 1:57 AM (UTC+5:30)	Apr 1, 2024 1:57 AM (UTC+5:30)
FINALIZING	✓ Succeeded	-	<1 sec	Apr 1, 2024 1:57 AM (UTC+5:30)	Apr 1, 2024 1:57 AM (UTC+5:30)
COMPLETED	✓ Succeeded	-	-	Apr 1, 2024 1:57 AM (UTC+5:30)	-

- Verify by going to your artifacts manually, that is S3 bucket whether your build is uploading there or not.



The build is finally completed.

i) Build Environment: CodeBuild offers pre-configured environments like Ubuntu, Amazon Linux, and Windows, so pick the one that fits your project. You can also create custom environments using Docker images for more specialized requirements.

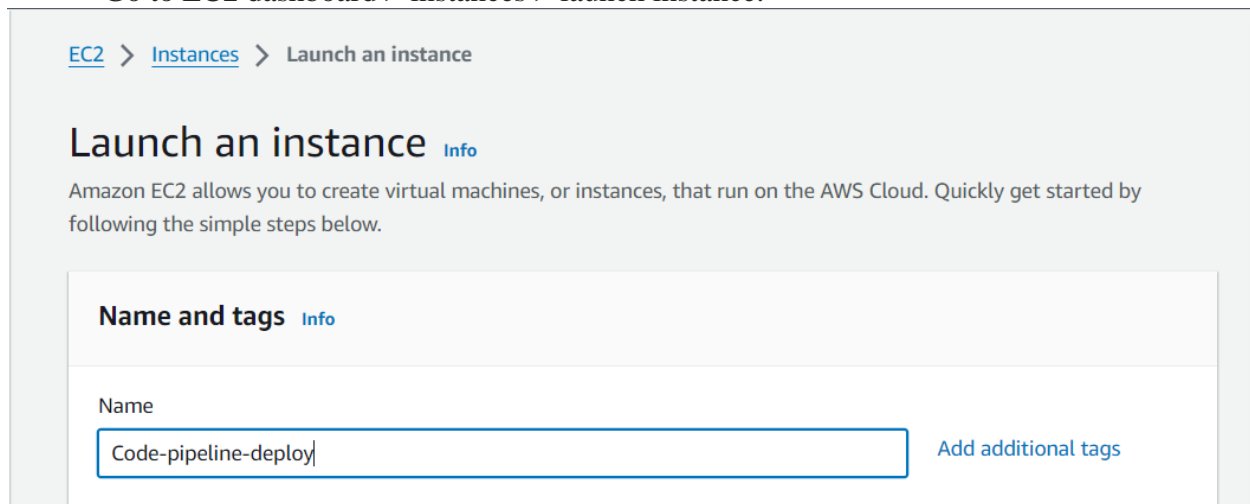
ii) Build Caching: Improve build performance by enabling build caching. CodeBuild caches dependencies and intermediate build artifacts, reducing build times for subsequent runs.

4. Creating EC2 Instance & Set-Up Agent:

- Before proceeding with AWS CodeDeploy to automate software deployments, it's essential to set up an EC2 instance where your application will be deployed. This instance serves as the target environment for your deployments. Therefore, the initial step involves creating an EC2 instance within your AWS environment.

i) Create an EC2 Instance:


- Go to EC2 dashboard > instances > launch instance.



- Choose image ‘Ubuntu 22.04’.




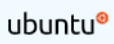
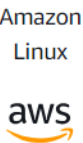
▼ Application and OS Images (Amazon Machine Image) [Info](#)


An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

 Search our full catalog including 1000s of application and OS images

Recents

Quick Start




[Browse more AMIs](#)
Including AMIs from
AWS, Marketplace and
the Community

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type

Free tier eligible ▼

ami-08116b9957a259459 (64-bit (x86)) / ami-012bf399e76fe4368 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2024-03-01

Architecture

AMI ID

64-bit (x86) ▼

ami-08116b9957a259459

Verified provider

- Choose instance type ‘t2.micro’, it is enough for this project.

▼ Instance type [Info](#) | [Get advice](#)

Instance type

t2.micro Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true

On-Demand Linux base pricing: 0.0116 USD per Hour

On-Demand SUSE base pricing: 0.0116 USD per Hour

On-Demand Windows base pricing: 0.0162 USD per Hour

On-Demand RHEL base pricing: 0.0716 USD per Hour

☒ All generations

[Compare instance types](#)

[Additional costs apply for AMIs with pre-installed software](#)

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

- Allow all **‘SSH Traffic’** and **‘HTTPS Traffic’** in the firewall.

▼ Network settings [Info](#)

Edit

Network [Info](#)

vpc-0c8b7481e2980ed0d

Subnet [Info](#)

No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)

Enable

Firewall (security groups) [Info](#)


A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group

☐ Select existing security group

We'll create a new security group called **'launch-wizard-3'** with the following rules:

- ☒ Allow SSH traffic from Anywhere
0.0.0.0/0
Helps you connect to your instance
- ☒ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server
- ☒ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

 Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

×

ii) Setting Up 'CodeDeploy' Agent in EC2:

- The AWS CodeDeploy agent plays a crucial role in facilitating deployments by acting as a bridge between your EC2 instances and the CodeDeploy service. Once installed and configured on an instance, this software package enables seamless integration with CodeDeploy, allowing the instance to participate in deployment processes.
- Connect to your instance.

Instances (1/4) [Info](#) [Refresh](#) [Connect](#) [Instance state](#) [Actions](#) [Launch instances](#)

[All states](#) [< 1 >](#) [Settings](#)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
<input checked="" type="checkbox"/>	Code-pipeline-...	i-0d17e0117b57298b6	Running	t2.micro	2/2 checks passed	View alarms	us-west-2a	ec2-35-...

[EC2](#) > [Instances](#) > [i-0d17e0117b57298b6](#) > [Connect to instance](#)

Connect to instance [Info](#)

Connect to your instance i-0d17e0117b57298b6 (Code-pipeline-deploy) using any of these options

EC2 Instance Connect

Session Manager

SSH client

EC2 serial console

Instance ID

[i-0d17e0117b57298b6](#) (Code-pipeline-deploy)

Connection Type

☒ **Connect using EC2 Instance Connect**
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.

☐ **Connect using EC2 Instance Connect Endpoint**
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address

[35.88.137.13](#)

Username

Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ubuntu.

Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

[Cancel](#) [Connect](#)

- Now create a script on the instance, '**agent-install.sh**', and copy the below content in that file.

```
#!/bin/bash
# This installs the CodeDeploy agent and its prerequisites on Ubuntu 22.04.
sudo apt-get update
sudo apt-get install ruby-full ruby-webrick wget -y
cd /tmp
```

```
wget https://aws-codedeploy-us-west-2.s3.us-west-2.amazonaws.com/releases/codedeploy-agent_1.3.2-1902_all.deb
mkdir /usr/local/bin
dpkg-deb -R codedeploy-agent_1.3.2-1902_all.deb codedeploy-agent_1.3.2-1902_ubuntu22
sed 's/Depends:./Depends:ruby3.0/' -i /usr/local/bin/codedeploy-agent_1.3.2-1902_ubuntu22/DEBIAN/control
dpkg-deb -b /usr/local/bin/codedeploy-agent_1.3.2-1902_ubuntu22/
sudo dpkg -i /usr/local/bin/codedeploy-agent_1.3.2-1902_ubuntu22.deb
systemctl list-units --type=service | grep codedeploy
sudo service codedeploy-agent status
```

- If your region is different than **Oregon**, then replace us-west-2 in the above script with your region code.
- Now run the script with the following command.

```
bash agent-install.sh
```

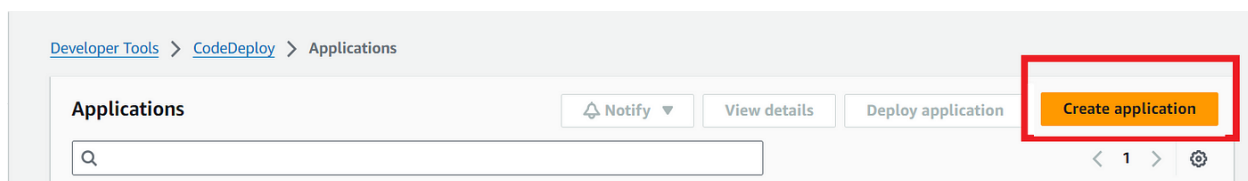
- This command will install an agent inside the instance.

5. CodeDeploy:

- CodeDeploy automates application deployments across Amazon EC2 instances, on-premises servers, Lambda functions, and ECS services, streamlining the deployment process.

i) Prepare your application:

- Prepare your application for deployment by packaging it into a .zip file containing all necessary code and dependencies.



Create application

Application configuration

Application name

Enter an application name

100 character limit

Compute platform

Choose a compute platform

Tags

Cancel

Create application

- After creating the Application, now create the deployment group.

ii) Deployment Group: In an EC2/On-Premises deployment, a deployment group serves as a collection of specific instances designated for a deployment. This group comprises individually tagged instances, Amazon EC2 instances, or those within Amazon EC2 Auto Scaling groups. It provides a means to organize and manage the deployment process, ensuring that updates are applied precisely to the intended instances.

- Navigate to the ‘**CodeDeploy**’ service in the AWS Management Console.
- Click ‘**Create deployment group**’ and specify details such as deployment configuration, EC2 instances, and deployment type.

Devops-on-AWS

Notify ▼

Delete application

Application details

Name	Compute platform
Devops-on-AWS	EC2/On-premises

Deployments

Deployment groups

Revisions

Deployment groups

View details

Edit

Create deployment group

Q

< 1 > ⚙

Application

Application
Devops-on-AWS
Compute type
EC2/On-premises

Deployment group name

Enter a deployment group name

code-pipeline-deployment-group

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.

Q arn:aws:iam::084298470880:role/code-deploy-service-role

×

iii) Service Role:

- Now to enter this service role we must create it first.
- Go to **IAM** > Roles > Create Role.

Step 1

Select trusted entity

Step 2

Add permissions

Step 3

Name, review, and create

Select trusted entity [Info](#)

Trusted entity type

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

☐ **AWS account**
 Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

☐ **Web identity**
 Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

☐ **SAML 2.0 federation**
 Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

☐ **Custom trust policy**
 Create a custom trust policy to enable others to perform actions in this account.

Use case

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

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.

- Add the following permission to this Role.

Permissions policies (6) [Info](#)

[Refresh](#)
[Simulate](#)
[Remove](#)
[Add permissions](#)

You can attach up to 10 managed policies.

Filter by Type

All types

[<](#)
[1](#)
[>](#)
[Settings](#)

<input type="checkbox"/>	Policy name Info	Type	Attached entities
<input type="checkbox"/>	AmazonEC2FullAccess	AWS managed	2
<input type="checkbox"/>	AmazonEC2RoleforAWSCodeDeploy	AWS managed	1
<input type="checkbox"/>	AmazonEC2RoleforAWSCodeDeployLi...	AWS managed	1
<input type="checkbox"/>	AmazonS3FullAccess	AWS managed	2
<input type="checkbox"/>	AWSCodeDeployFullAccess	AWS managed	2
<input type="checkbox"/>	AWSCodeDeployRole	AWS managed	1

- Give a name and create it.

Name, review, and create

Role details

Role name

Enter a meaningful name to identify this role.

code-deploy-service-role

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.

- After creating a role, return to your codeDeploy configuration and add this role to the service role section.

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::084298470880:role/code-deploy-service-role



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.


- Now we must enter our EC2-instance in **‘Environment configuration’**.

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*


Remove tag

Add tag

+ Add tag group

☐ On-premises instances

Matching instances

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

- Save the changes.

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

Save changes

6. Create Deployment:

- For creating a deployment, go to **Application > deployment group > create deployment**.

[Developer Tools](#) > [CodeDeploy](#) > [Applications](#) > [Devops-on-AWS](#) > [Create deployment](#)

Create deployment

Deployment settings

Application

Devops-on-AWS

Deployment group

Q code-pipeline-deployment-group X

Compute platform

EC2/On-premises

Deployment type

In-place

Managed hook execution role

The IAM role used by the CodeDeploy Managed Hook function to perform actions. [Edit Managed Hook execution role](#).

-

- Paste the link of your S3 artifact where your build files are stored by code build, in the below column.

Revision type

☒ My application is stored in Amazon S3

☐ My application is stored in GitHub

Revision location

Copy and paste the Amazon S3 bucket where your revision is stored

s3://bucket-name/folder/object.[zip|tar|tgz]

Revision file type

- And create deployment.

i)EC2-ROLE:

- After creating a deployment, it's essential to set up an additional role for the EC2 instance. This role facilitates seamless communication between the EC2 instance and both CodeDeploy and Amazon S3. This setup ensures that the instance can securely access and interact with the necessary services during the deployment process.
- Create Role. Go to **IAM > Roles > Create role**.

Trusted entity type

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

☐ **AWS account**
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

☐ **Web identity**
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

☐ **SAML 2.0 federation**
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

☐ **Custom trust policy**
Create a custom trust policy to enable others to perform actions in this account.

Use case

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

Service or use case

Choose a use case for the specified service.

Use case

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

- Add the below permissions to this role.

Permissions policies (3) [Info](#)

You can attach up to 10 managed policies.

Filter by Type

Search All types < 1 > [Settings](#)

<input type="checkbox"/>	Policy name Info	Type	Attached entities
<input type="checkbox"/>	AmazonEC2FullAccess	AWS managed	2
<input type="checkbox"/>	AmazonS3FullAccess	AWS managed	2
<input type="checkbox"/>	AWSCodeDeployFullAccess	AWS managed	2

- Give a name and create.

Role details

Role name
Enter a meaningful name to identify this role.

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

Description
Add a short explanation for this role.

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

ii) Attach to EC2 instance:

- After creating this role attach it to your EC2-instance.

[EC2](#) > [Instances](#) > i-0d17e0117b57298b6

Instance summary for i-0d17e0117b57298b6 (Code-pipeline-deploy) [Info](#)

Updated less than a minute ago

<p>Instance ID</p> <p> i-0d17e0117b57298b6 (Code-pipeline-deploy)</p> <p>IPv6 address</p> <p>–</p> <p>Hostname type</p> <p>IP name: ip-172-31-21-178.us-west-2.compute.internal</p> <p>Answer private resource DNS name</p> <p>IPv4 (A)</p> <p>Auto-assigned IP address</p> <p> 35.88.137.13 [Public IP]</p> <p>IAM Role</p> <p> ec2-code-deploy</p> <p>IMDSv2</p> <p>Required</p>	<p>Public IPv4 address</p> <p> 35.88.137.13 Open address</p> <p>Instance state</p> <p> Running</p> <p>Private IP DNS name (IPv4 only)</p> <p> ip-172-31-21-178.us-west-2.compute.internal</p> <p>Instance type</p> <p>t2.micro</p> <p>VPC ID</p> <p> vpc-0c8b7481e2980ed0d</p> <p>Subnet ID</p> <p> subnet-01df4d256579466d9</p>	<p>Private IPv4 address</p> <p> 172.31.21.178</p> <p>Public IPv4 DNS</p> <p>Elastic IP addresses</p> <p>–</p> <p>AWS Compute Optimizer finding</p> <p> Opt-in to AWS Compute Optimizer for recommendations.</p> <p>Learn more</p> <p>Auto Scaling Group name</p> <p>–</p>	<p>Connect</p> <p>Manage instance state</p> <p>Instance settings ▶</p> <p>Networking ▶</p> <p>Security ▶</p> <p>Image and templates ▶</p> <p>Monitor and troubleshoot ▶</p>
--	---	---	---

Change security groups

Get Windows password

Modify IAM role


- Add the service role here to give required permissions to EC2-instance.

[EC2](#) > [Instances](#) > [i-0d17e0117b57298b6](#) > [Modify IAM role](#)

Modify IAM role [Info](#)



Attach an IAM role to your instance.

Instance ID

 **i-0d17e0117b57298b6** (Code-pipeline-deploy)

IAM role

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

 [Create new IAM role](#) 

[Cancel](#)
[Update IAM role](#)

After this, restart the code deploy agent service in your instance.

Command: **sudo service codedeploy-agent restart**

```
ubuntu@ip-172-31-21-178:~$ sudo service codedeploy-agent restart
```

iii) Adding App Specification File:

The Application Specification file (AppSpec file) is a YAML or JSON formatted configuration file utilized by CodeDeploy to orchestrate deployments. It contains instructions for CodeDeploy to manage the deployment process, including how to handle various lifecycle events, such as application installation, code deployment, and cleanup tasks.

iv) Create Required Files:

- Add the '**appspec.yml**' file with other required files to our code commit repository.
- \$vim appspec.yml

```

version: 0.0
os: linux
files:
  - source: /
    destination: /var/www/html
hooks:
  AfterInstall:
    - location: scripts/install_nginx.sh
      timeout: 300
      runas: root
  ApplicationStart:
    - location: scripts/start_nginx.sh
      timeout: 300
      runas: root
~
~
~
~
appspec.yml [unix] (00:26 01/04/2024)

```

- Also add some scripts in the script folder that will perform the required task on the instance.
- Commands:
 - \$ mkdir scripts/ (Under Devops-on-AWS folder)
 - \$ cd scripts/
 - \$ cat install_nginx.sh
 - \$ cat start_nginx.sh
- Add and commit all the changes:

```

$ ls
appspec.yml  buildspec.yml  index.html  scripts/

```

```

git add .
git commit -m "add appspec.yml"
git push

```

v)Start the Build:

Build logs	Phase details	Reports
Name	Status	
SUBMITTED	✓ Succeeded	
QUEUED	✓ Succeeded	
PROVISIONING	✓ Succeeded	
DOWNLOAD_SOURCE	✓ Succeeded	
INSTALL	✓ Succeeded	
PRE_BUILD	✓ Succeeded	
BUILD	✓ Succeeded	
POST_BUILD	✓ Succeeded	
UPLOAD_ARTIFACTS	✓ Succeeded	
FINALIZING	✓ Succeeded	
COMPLETED	✓ Succeeded	

- Now your artifact is uploaded to the targeted bucket and path.

vi)Deploy your application:

- Start the deployment.

- CodeDeploy will automatically deploy your application to the specified EC2 instances, ensuring minimal downtime and rollback capabilities.

[Developer Tools](#) > [CodeDeploy](#) > [Deployments](#) > d-44QYY2WJ4

d-44QYY2WJ4 Copy deployment Retry deployment

Deployment status

Installing application on your instances

1 of 1 instances updated ✓ Succeeded 100%

Deployment details

Application Devops-on-AWS	Deployment ID d-44QYY2WJ4	Status ✓ Succeeded
Deployment configuration CodeDeployDefault.AllAtOnce	Deployment group code-pipeline-deployment-group	Initiated by User action
Deployment description		

RESULT:

- Access your EC2 instance by copying its IP address and pasting it into your web browser. This action allows you to view the content of your **'index.html'** file hosted on the instance.

Best Practices:

- Use blue-green deployments to minimize downtime and risk during deployments.
- Leverage deployment hooks to run custom scripts before and after deployment.
- Monitor deployment health using **CloudWatch** metrics and alarms.

7. Creating a Complete CI/CD Pipeline with CodePipeline:

CodePipeline: AWS CodePipeline is a comprehensive continuous integration and continuous delivery service that automates the build, test, and deployment stages of your software release process. Setting up a CI/CD pipeline involves defining the workflow for your application's lifecycle, including source code management, build automation, testing, and deployment, all orchestrated seamlessly within CodePipeline.

i) Create a pipeline:

- Navigate to the **'CodePipeline'** service in the AWS Management Console.
- Click on **'Create pipeline'** and provide a name for your pipeline.

Pipeline settings

Pipeline name

Enter the pipeline name. You cannot edit the pipeline name after it is created.

No more than 100 characters

Pipeline type

The pipeline type determines the pipeline structure and availability of parameters such as triggers. Pipeline type selection will impact features and pricing. [Which pipeline is right for me?](#)

☐ V1☒ V2

Execution mode

Choose the execution mode for your pipeline. This determines how the pipeline is run.

☐ Superseded

A more recent execution can overtake an older one. This is the default.

☒ Queued (Pipeline type V2 required)

Executions are processed one by one in the order that they are queued.

☐ Parallel (Pipeline type V2 required)

Executions don't wait for other runs to complete before starting or finishing.

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

Type your service role name

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

- **Define the source stage:** Choose **'CodeCommit'** as the source provider and select your repository.

Source

Source provider

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

Repository name

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

Branch name

Choose a branch of the repository

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

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.

- **Define the build stage:** Choose ‘CodeBuild’ as the build provider and select your build project.

Add build stage [Info](#)

Step 3 of 5

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.

AWS CodeBuild ▼

Region

US West (Oregon) ▼

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.

code-build-for-devops X 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.

- **Define the deploy stage:** Choose ‘CodeDeploy’ as the deployment provider and specify your deployment group.

[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 deploy stage [Info](#)

Step 4 of 5

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 West (Oregon) ▼

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.

Devops-on-AWS X

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.

code-pipeline-deployment-group X

- **Review and create it:** Review the pipeline configuration and click **‘Create pipeline’** to initiate the pipeline creation process.

i)Pipeline Execution:

Once the pipeline is created, it will automatically trigger when you commit to your code commit repo.

Pipeline execution flow works:

1. Source stage:

- **CodePipeline** retrieves the source code from the specified **CodeCommit** repository.

2. Build stage:

- **CodePipeline** triggers the **CodeBuild** project, which compiles the source code, runs tests, and produces build artifacts.

3. Deploy stage:

- **CodePipeline** deploys the build artifacts to the specified **CodeDeploy** deployment group, automating the deployment process to EC2 instances.

4. Monitor and manage:

- Monitor pipeline execution status and view detailed execution logs in the CodePipeline console.
- Manage pipeline stages, actions, and settings as per your project requirements.

VERIFICATION:

- Committing and pushing changes to the **‘index.html’** file in the **CodeCommit** repository will trigger the CI/CD pipeline. This event initiates the automated workflow defined in the pipeline, encompassing processes like build, test, and deployment, ensuring that the latest changes are seamlessly integrated and deployed to the target environment.

Developer Tools > CodePipeline > Pipelines > Devops-On-AWS

Devops-On-AWS

Pipeline type: **V2** Execution mode: **QUEUED**

Source Succeeded

Pipeline execution ID: [c7a52095-bf36-4f39-8ab4-ec5209b7660c](#)

Source

[AWS CodeCommit](#)

Succeeded - 14 hours ago

[83d634d5](#)

View details

[83d634d5](#) Source: index.html updated

↓ Disable transition

Build Succeeded

Pipeline execution ID: [c7a52095-bf36-4f39-8ab4-ec5209b7660c](#)

Build

[AWS CodeBuild](#)

Succeeded - 14 hours ago

View details

[83d634d5](#) Source: index.html updated

↓ Disable transition

Deploy Succeeded

Pipeline execution ID: [c7a52095-bf36-4f39-8ab4-ec5209b7660c](#)

Deploy

[AWS CodeDeploy](#)

Succeeded - 14 hours ago

View details

[83d634d5](#) Source: index.html updated

You've successfully created a complete CI/CD pipeline using AWS CodePipeline. By automating the build, test, and deployment phases of your release process, you can accelerate software delivery and improve overall efficiency.