

# **Name-Ishan Kiran Joshi Div-D15C Roll No-21 A.Y.-2024-25**

## **Advance DevOps Practicals**

### Case Study 1

#### 1. Cloud Deployment with Automation

- Concepts Used: AWS CodePipeline, EC2, and S3.
  - Problem Statement: "Build a simple web application using AWS CodeBuild and deploy it to an S3 bucket. Then, automate the deployment process using AWS CodePipeline, ensuring the application is deployed on an EC2 instance. Use a sample index.html page for demonstration."
  - Tasks:
    - Set up AWS CodeBuild for the web app.
    - Create a pipeline that deploys to an S3 bucket.
    - Use AWS CodeDeploy to push updates to an EC2 instance.
- 

### 1-Introduction:

In today's fast-paced digital landscape, the need for efficient and reliable deployment processes has never been more critical. Businesses are increasingly migrating their applications to the cloud, seeking to enhance scalability, reduce downtime, and improve overall operational efficiency. This case study focuses on leveraging Amazon Web Services (AWS) to automate the deployment of a simple web application, showcasing key AWS tools and best practices that facilitate rapid and reliable application updates.

**Case Study Overview:** This case study aims to build and deploy a straightforward web application, consisting of a sample index.html page, utilizing a combination of AWS services, specifically AWS CodePipeline, CodeBuild, EC2, and S3. The primary goal is to illustrate how these services can work together to create a seamless and automated deployment workflow.

1. AWS CodePipeline will serve as the backbone of our automation process, orchestrating the various stages of the deployment lifecycle from source to production. This continuous integration and continuous delivery (CI/CD) service automates the workflow, allowing for rapid iterations and consistent deployment practices.
2. AWS CodeBuild will handle the compilation of the application. By automatically building and testing the application, CodeBuild ensures that only validated code is deployed, minimizing the risk of introducing bugs into production environments.
3. Amazon S3 (Simple Storage Service) will be used to host the static content of the web application. S3 offers a highly durable and scalable storage solution, allowing for fast access and distribution of files over the web. By hosting the web application on S3, we take advantage of its cost-effectiveness and ease of use.
4. AWS EC2 will serve as the compute resource where our application is deployed. EC2 provides scalable computing capacity in the cloud, allowing users to run applications without the need to invest in physical hardware.
5. AWS CodeDeploy will be responsible for managing the deployment process to the EC2 instance. By utilizing CodeDeploy, we can deploy application easily and efficiently, enabling automated deployments and rollbacks in case of failures.

#### Key Feature and Application:

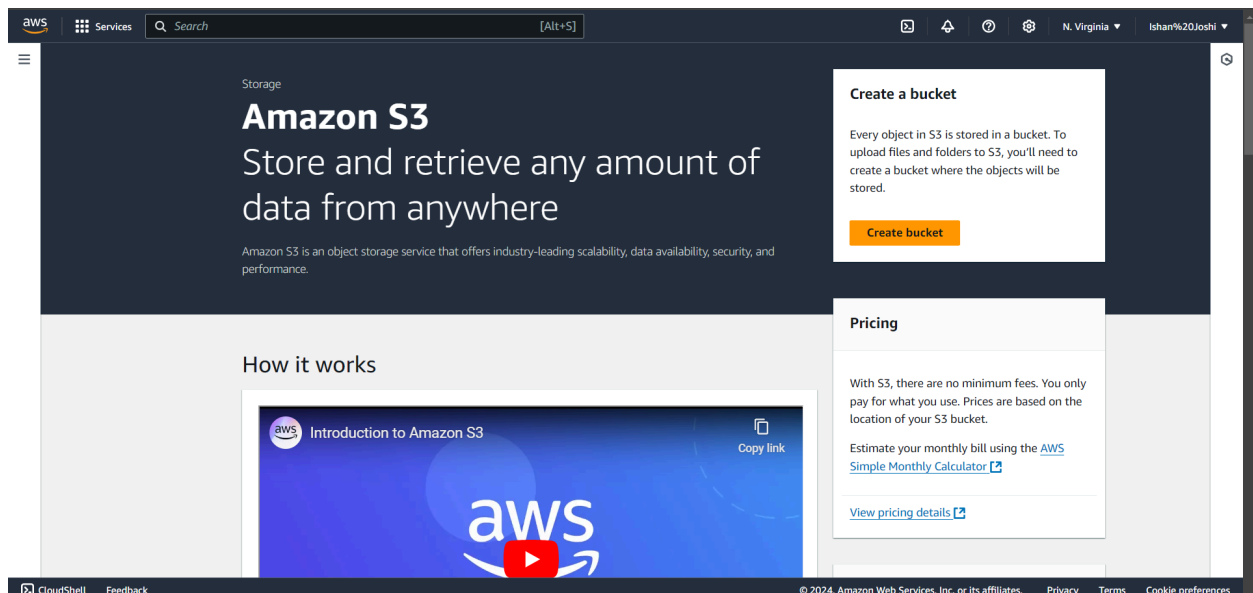
1. Automated Deployment Pipeline: AWS CodePipeline automates code deployment to S3 and EC2, reducing manual effort.
2. CI/CD Integration: Continuous integration and deployment with CodeBuild and CodeDeploy streamlines updates.
3. Scalability: EC2 provides scalable compute power, and S3 offers scalable storage.
4. Version Control: CodeDeploy enables easy rollback to previous versions in case of failure
5. Cost-Effective: S3 offers affordable static website hosting.
6. Web App Hosting: Ideal for hosting dynamic and static web apps.
7. Automated Updates: Simplifies the deployment of new features and updates

## 2. Step-by-Step Explanation

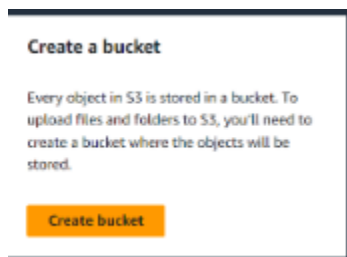
### Part 1: Set Up the S3 Bucket for Static Website Hosting

Create an S3 Bucket:

- Open the AWS S3 Console.



- Click Create Bucket.



- Enter a unique bucket name (e.g., ishan-bucket-advdevops).

## Create bucket [Info](#)

Buckets are containers for data stored in S3.

### General configuration

AWS Region

US East (N. Virginia) us-east-1

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**

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](#)

ishan-bucket-advdevops

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

- Uncheck "Block all public access" .

### 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.

- Acknowledge that the bucket will be public by checking the acknowledgment box.




**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.

- Click Create bucket.

#### ► Advanced settings

 After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

Cancel

Create bucket

Enable Static Website Hosting:


- In the S3 console, go to the newly created bucket.

○ [ishan-bucket-advdevops](#) US East (N. Virginia) us-east-1 [View analyzer for us-east-1](#) October 19, 2024, 21:24:58 (UTC+05:30)

- Select the Properties tab.

Objects **Properties** Permissions Metrics Management Access Points

#### Bucket overview

AWS Region US East (N. Virginia) us-east-1	Amazon Resource Name (ARN)  arn:aws:s3:::ishan-d15c-21-casestudy	Creation date October 19, 2024, 13:12:22 (UTC+05:30)
---	--	---

- Scroll down to Static website hosting and click Edit.

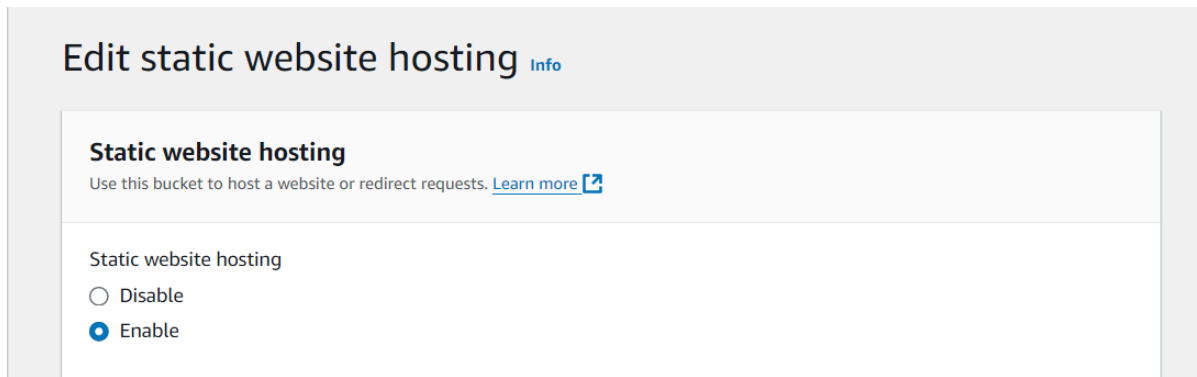
#### Static website hosting

Edit

Use this bucket to host a website or redirect requests. [Learn more](#)

S3 static website hosting  
Disabled

- Select Enable and choose Host a static website.



**Edit static website hosting** [Info](#)

**Static website hosting**  
Use this bucket to host a website or redirect requests. [Learn more](#) [link]

Static website hosting

☐ Disable

☒ Enable

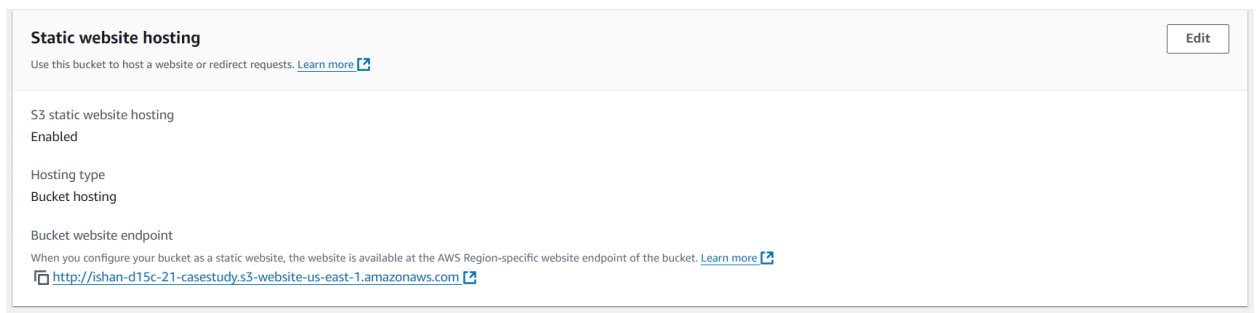
- Set the index document to index.html.

### Index document

Specify the home or default page of the website.

index.html

- Copy the Bucket website endpoint—this will be the URL for your site (you'll use this later for testing).



**Static website hosting** [Edit](#)

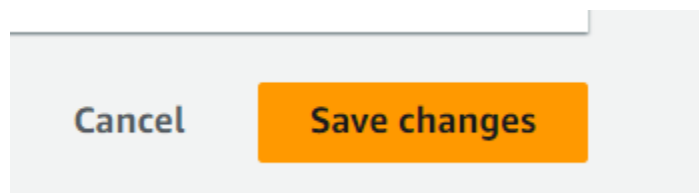
Use this bucket to host a website or redirect requests. [Learn more](#) [link]

S3 static website hosting  
Enabled

Hosting type  
Bucket hosting

Bucket website endpoint  
When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket. [Learn more](#) [link]  
<http://ishan-d15c-21-casestudy.s3-website-us-east-1.amazonaws.com> [link]

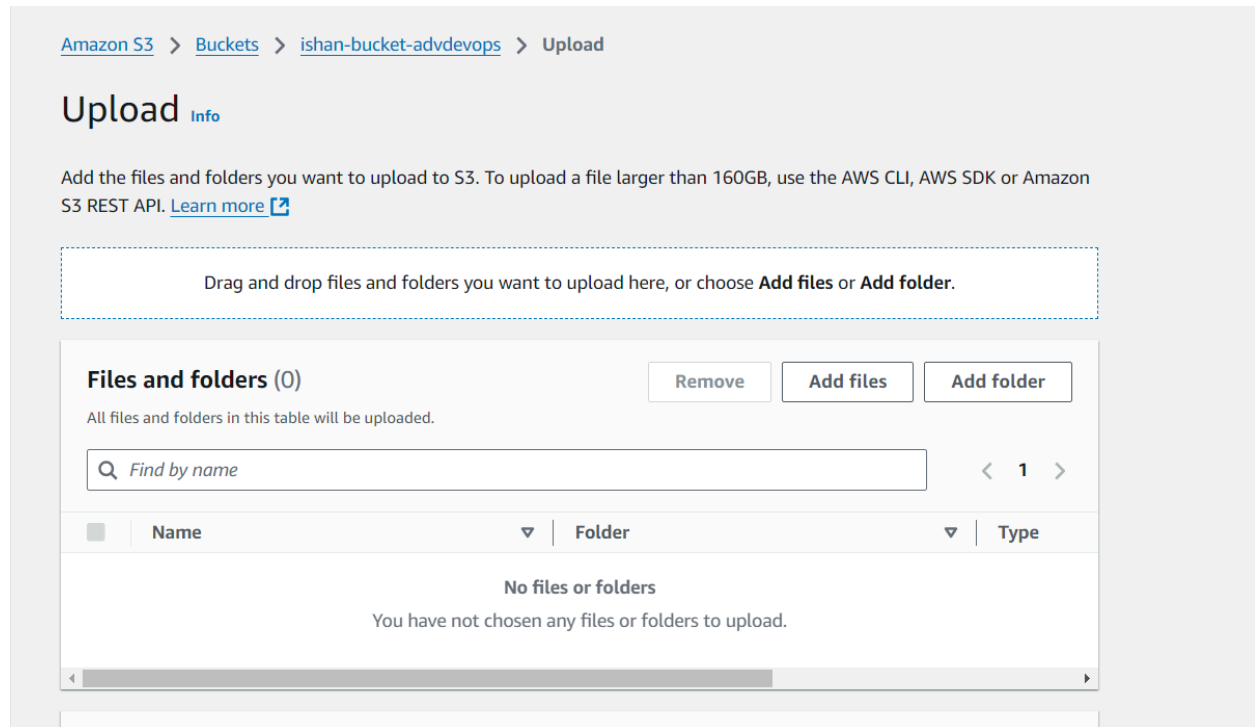
- Click Save changes.



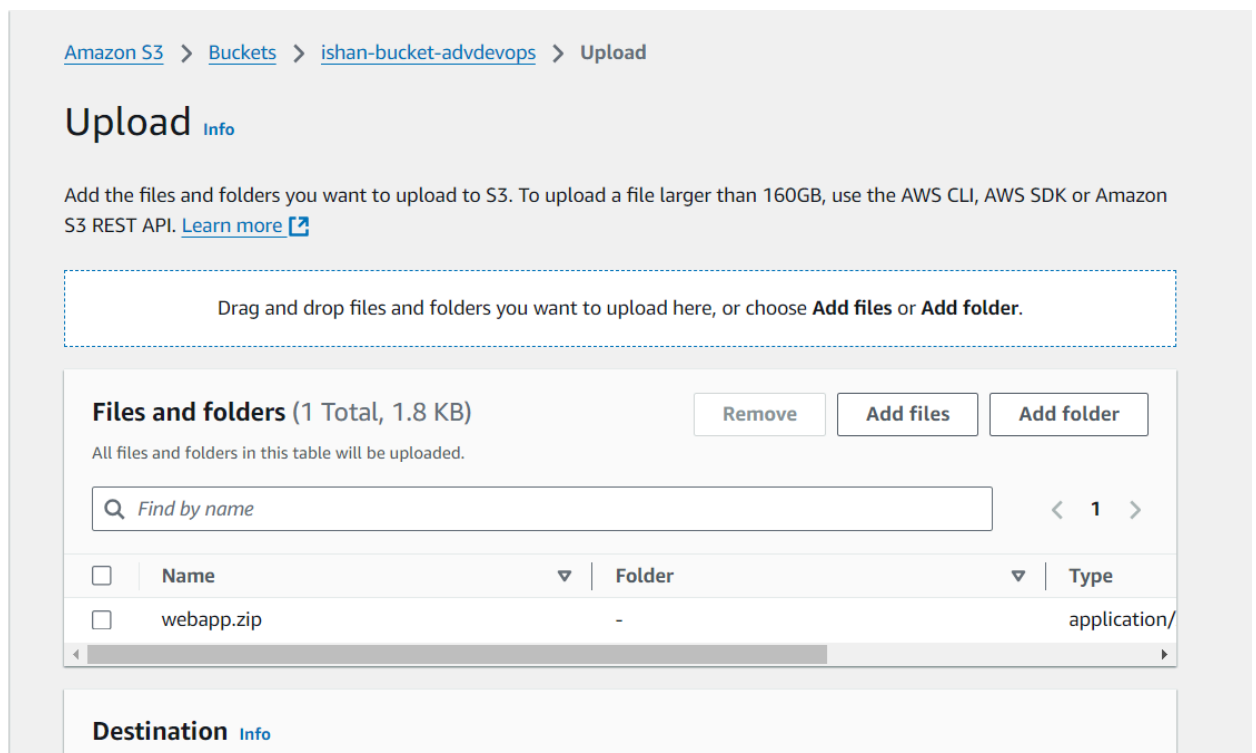
Cancel Save changes

Upload Sample index.html:

- Go to the Objects tab in your bucket.



- Click Upload, and add your webapp.zip



Set Bucket Policy to Allow Public Access:

- Go to the Permissions tab in the S3 bucket.

The screenshot shows the AWS S3 console's 'Permissions' tab for a bucket. At the top, there are tabs for 'Objects', 'Properties', 'Permissions' (which is selected), 'Metrics', 'Management', and 'Access Points'. Below the tabs is a 'Permissions overview' section with an 'Access finding' message. The main section is titled 'Block public access (bucket settings)' and includes an 'Edit' button. It explains that public access is granted through ACLs, bucket policies, and access point policies, and recommends turning on 'Block all public access'. Below this, there is a toggle switch for 'Block all public access' which is currently 'Off', and a link to view individual settings.

- Scroll to Bucket policy and click Edit.

The screenshot shows the 'Bucket policy' section in the AWS S3 console. It has an 'Edit' button and a 'Delete' button. Below the buttons, there is a message stating 'No policy to display.' and a 'Copy' button. The section explains that the bucket policy, written in JSON, provides access to the objects stored in the bucket and that bucket policies don't apply to objects owned by other accounts.

Add the following policy to allow public access to your web content:

json

Copy code

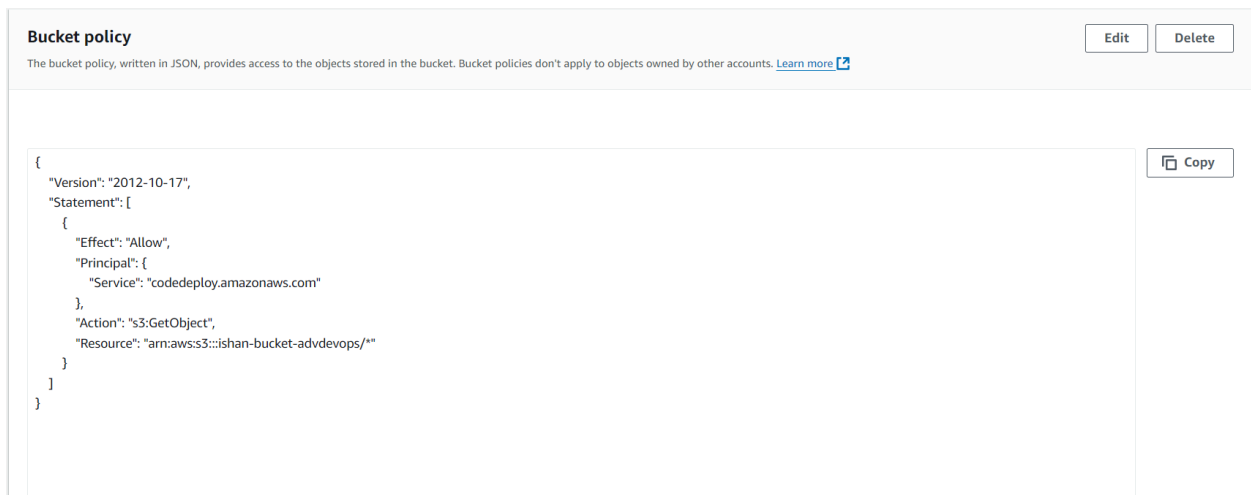
```
{  
  
  "Version": "2012-10-17",  
  
  "Statement": [  
  
    {  
  
      "Sid": "PublicReadGetObject",  
  
      "Effect": "Allow",  
  
      "Principal": "*",  
  
      "Action": "s3:GetObject",
```



```

    "Resource": "arn:aws:s3:::ishan-bucket-advdevops/*"
  }
]
}

```

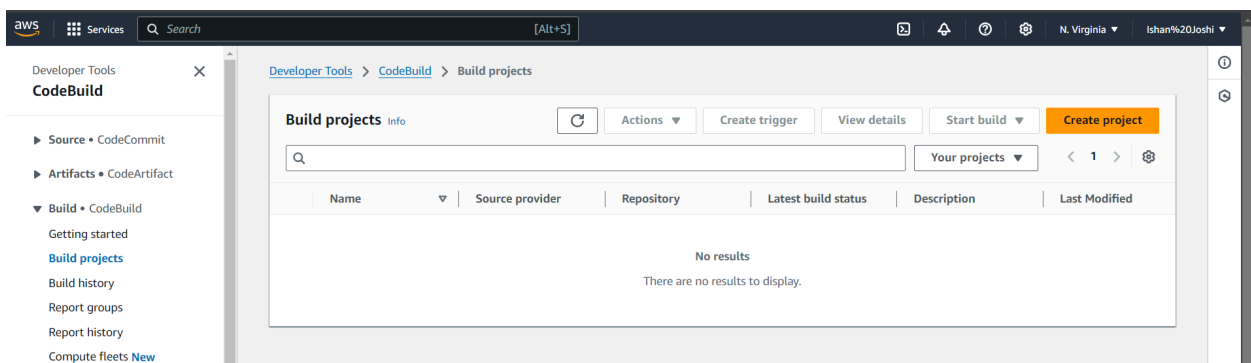


- Click Save changes.

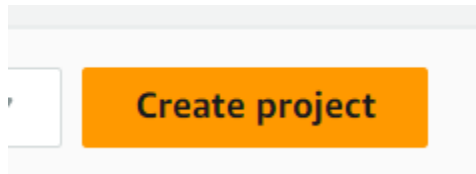


## Part 2: Set Up AWS CodeBuild

1. Create a Build Project:
  - Open the AWS CodeBuild Console.



- Click Create build project.



- Name your project (e.g., ishan-codebuild).

## Create build project

### Project configuration

Project name

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**

Description, Build badge, Concurrent build limit, tags

- For Source provider, select Amazon S3 . Select your S3 bucket or repository where the web app code is stored.

### Source

Add source

#### Source 1 - Primary

Source provider

Amazon S3 ▼

Bucket

ishan-bucket-advdevops ✕

S3 object key or S3 folder

webapp.zip

Source version - *optional* [Info](#)

Enter the version ID of the object that represents the build input ZIP file.

- Under Environment, select Managed image and choose:
  - Operating system: Amazon Linux 2.
  - Runtime: Standard.
  - Image: aws/codebuild/standard:5.0 (or whichever version supports your language/runtime).
  - For Service Role, allow CodeBuild to create a new role or use an existing role if you have one.

Amazon Linux ▼

Runtime(s)

Standard ▼

Image

aws/codebuild/amazonlinux2-x86\_64-standard:5.0 ▼

Image version

Always use the latest image for this runtime version ▼

☐ Use GPU-enhanced compute

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-ishan-codebuild-service-role

Type your service role name

► Additional configuration

BuildSpec:

Buildspec

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

Build commands Info

```

1 version: 0.2
2 phases:
3   build:
4     commands:
5       - echo "Hello World"
6   artifacts:
7     files:
8       - index.html

```

1:1 YAML Spaces: 2

Test the Build:

- In the CodeBuild Console, manually trigger a build by clicking Start Build.
- Confirm that the build completes successfully.

Developer Tools > CodeBuild > Build projects > ishan-codebuild > ishan-codebuild:f9f6ae35-993c-4afa-a4d0-e8427aeaccfd

ishan-codebuild:f9f6ae35-993c-4afa-a4d0-e8427aeaccfd

Stop build Retry build

Build status

Status	Initiator	Build ARN	Resolved source version
Succeeded	codepipeline/ishan-pipeline	arn:aws:codebuild:us-east-1:850995542805:build/ishan-codebuild:f9f6ae35-993c-4afa-a4d0-e8427aeaccfd	pKXvtKvmx0Slb0Vq.WEsgg_ZSyMq5N.6
Start time	End time	Build number	
Oct 19, 2024 9:36 PM (UTC+5:30)	Oct 19, 2024 9:37 PM (UTC+5:30)	1	

Build logs

Phase details

Reports

Environment variables

Build details

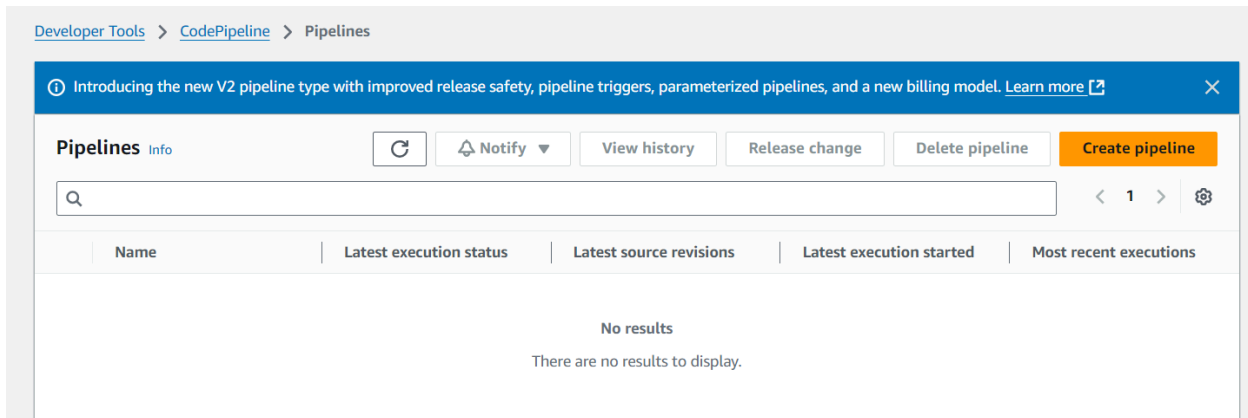
Resource utilization

Logs

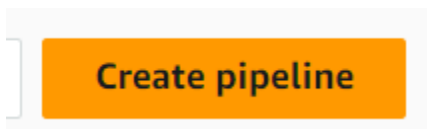
CloudWatch logs	CloudWatch group name	CloudWatch stream name
DISABLED	/aws/codebuild/ishan-codebuild	-
S3 logs	S3 location	Encryption disabled

### Part 3: Set Up AWS CodePipeline

1. Create a New Pipeline:
  - Open the AWS CodePipeline Console.



- Click Create Pipeline.



- Enter a Pipeline Name (e.g., ishanWebAppPipeline1).

**Pipeline settings**

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

ishan-pipeline

No more than 100 characters

**Pipeline type**

- For Service role, choose the default role created by CodePipeline, or create a new role with the necessary permissions.

**Service role**

☒ **New service role**  
Create a service role in your account

☐ **Existing service role**  
Choose an existing service role from your account

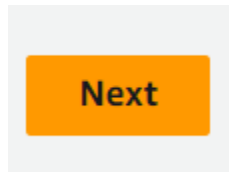
**Role name**

AWSCodePipelineServiceRole-us-east-1-ishan-pipeline

Type your service role name

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

- Click Next.



Source Stage:

- In the Add source stage, select Amazon S3.

### Source

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

Amazon S3 ▼

**Bucket**

- Select the S3 bucket or repository where your code is stored.

Bucket

ishan-bucket-advdevops X

S3 object key or S3 folder

webapp.zip

- For Change detection options, select Amazon CloudWatch Events to automatically detect changes in your source.

**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

☒ Enable automatic retry on stage failure

- Click Next.

Next

Build Stage:

- For Build provider, select AWS CodeBuild.

### Build - *optional*

#### Build provider

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

☐ Commands

☒ Other build providers

AWS CodeBuild ▼

- Select the CodeBuild project you created earlier (e.g., ishan-codebuild).

### Build - *optional*

#### Build provider

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

☐ Commands

☒ Other build providers

AWS CodeBuild ▼

#### 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.

🔍 ishan-codebuild



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

- Click Next.

**Next**

Deploy Stage (S3):

- For Deploy provider, select Amazon S3.

### Deploy - *optional*

#### Deploy provider

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

Amazon S3

#### Region

- Choose the S3 bucket where you want to deploy your website (this will be the same bucket used for static website hosting).

#### Bucket

ishan-bucket-advdevops

#### S3 object key or S3 folder

webapp.zip

- Click Next.

**Next**

### 1. Review and Create Pipeline:

- Review all the configurations and click Create Pipeline.

**Create pipeline**



[Developer Tools](#) > [CodePipeline](#) > [Pipelines](#) > ishan-pipeline

ishan-pipeline

Notify

Edit

Stop execution

Clone pipeline

Release change

Pipeline type: V2 Execution mode: QUEUED

Source

Succeeded

Pipeline execution ID: 78b9fbee-db4e-4f7a-bc12-4bd42ad28e66

Source

[Amazon S3](#)

Succeeded - 14 hours ago

View details

Source: Amazon S3 version id: pKXvtkVmx0Sib0Vq.WEsgg\_ZSyMq5N.6

Disable transition

✓

✓

✓

Build

Succeeded

Pipeline execution ID: 78b9fbee-db4e-4f7a-bc12-4bd42ad28e66

Build

[AWS CodeBuild](#)

Succeeded - 14 hours ago

View details

Source: Amazon S3 version id: pKXvtkVmx0Sib0Vq.WEsgg\_ZSyMq5N.6

Disable transition

Start rollback

✓

✓

✓

Deploy

Succeeded

Pipeline execution ID: 78b9fbee-db4e-4f7a-bc12-4bd42ad28e66

Deploy

[Amazon S3](#)

Succeeded - 14 hours ago

View details

Source: Amazon S3 version id: pKXvtkVmx0Sib0Vq.WEsgg\_ZSyMq5N.6

Start rollback

✓

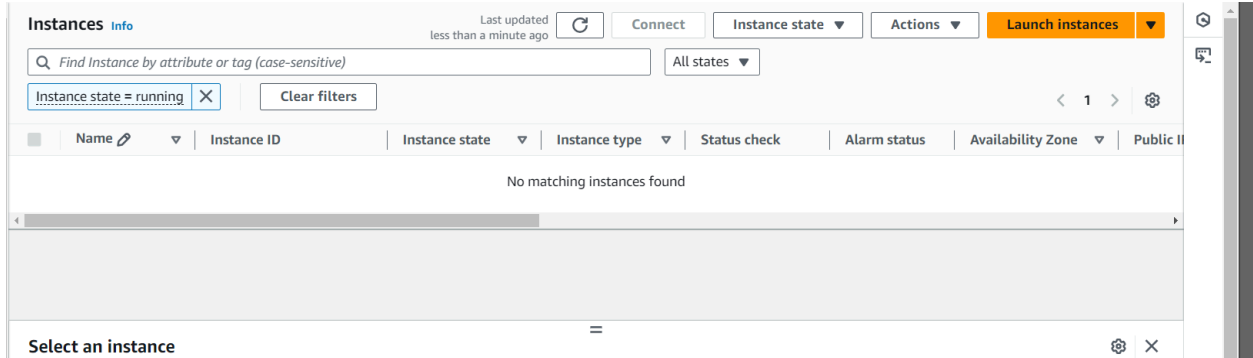
✓

© 2024 Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy | Terms | Feedback

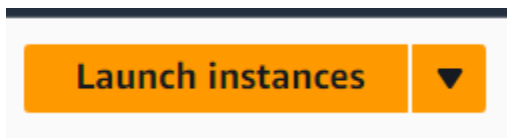
## Part 4: Set Up EC2 and CodeDeploy

### 1. Create an EC2 Instance:

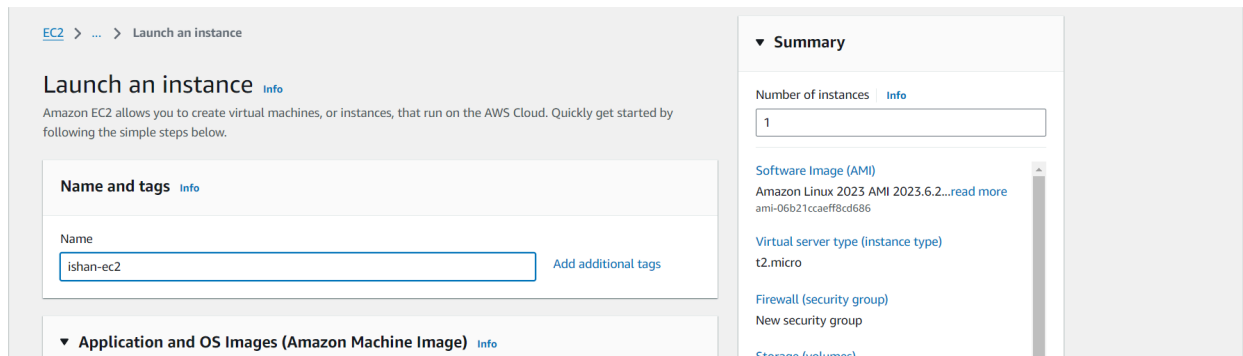
- Go to the AWS EC2 Console.



- Click Launch Instance.



- Select the Amazon Linux 2 AMI (free tier).



- Choose an instance type (e.g., t2.micro).

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

Instance type

t2.micro Free tier eligible

Family: t2  
1 vCPU 1 GiB Memory Current generation: true  
On-Demand Windows base pricing: 0.0162 USD per Hour  
On-Demand SUSE base pricing: 0.0116 USD per Hour  
On-Demand RHEL base pricing: 0.026 USD per Hour  
On-Demand Linux base pricing: 0.0116 USD per Hour

Additional costs apply for AMLs 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*

new

Create new key pair

▼ **Network settings** [Info](#) [Edit](#)

Number of instances [Info](#)

1

New security group

Storage (volumes)

1 volume(s) - 8 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMLs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel **Launch instance**

[Preview code](#)

- Open the Security Group settings and add a rule to allow HTTP traffic (port 80).

**Edit inbound rules** [Info](#)

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

**Inbound rules** [Info](#)

Security group rule ID	Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Source <a href="#">Info</a>	Description - optional <a href="#">Info</a>	
sgr-09b28b46854418499	HTTPS	TCP	443	Anyw...		Delete
sgr-095e2bc74e0bcae55	SSH	TCP	22	Anyw...		Delete
sgr-0dc9a9e1d2ad0d9af	HTTP	TCP	80	Anyw...		Delete

Add rule

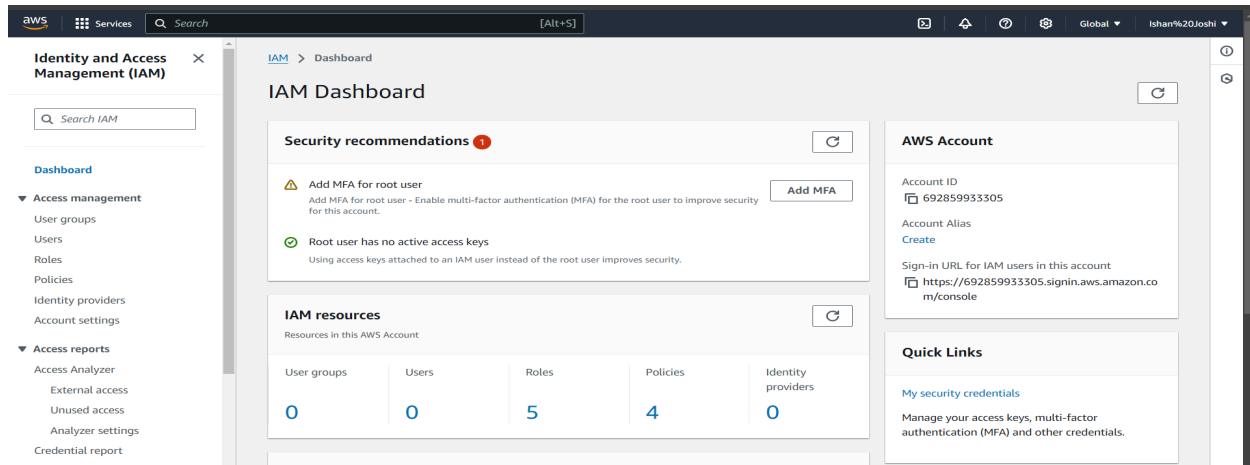
- Launch the instance.

**Launch instance**

<input type="checkbox"/>	ishan-ec2	i-06dddbf880648ceab	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	us-east-1a	ec2-54-
--------------------------	-----------	---------------------	---------	----------	-------------------	-----------------------------	------------	---------

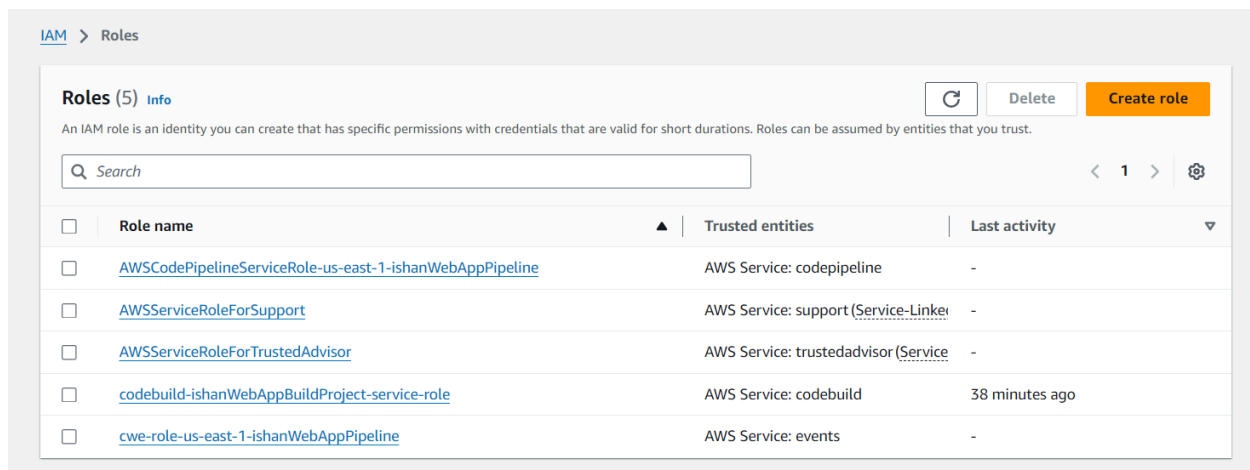
## Part 5: Create the IAM Role for EC2

1. Open the IAM Management Console:
  - Go to the AWS IAM Console at [AWS IAM Console](#).

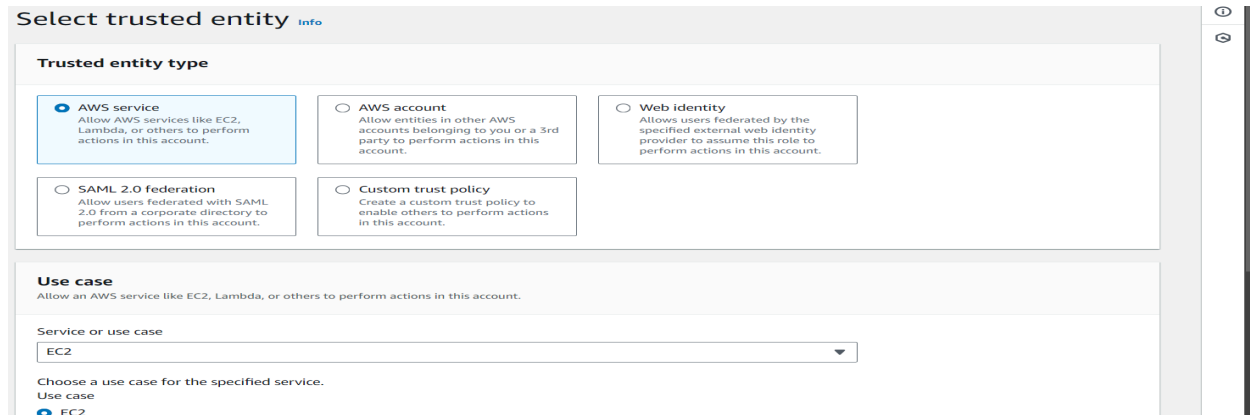


## 2. Create a New Role:

- On the left sidebar, click Roles.
- Click the Create role button.



## 3. Select EC2 as Trusted Entity:



#### 4. Attach the Policies:

- On the Permissions page, search for and select the following policies:
  - AmazonS3ReadOnlyAccess: Grants read-only access to S3.
  - AWSCodeDeployRole: Grants CodeDeploy permissions to deploy applications to EC2.
- Click Next.



#### 5. Set Role Name:

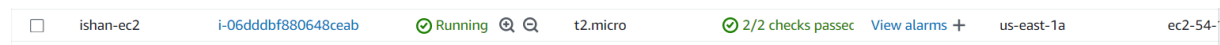
- Give the role a name (e.g., ishan-iam-role).
- Optionally, add a description.
- Click Create role.

A screenshot of the AWS IAM console 'Name, review, and create' page for a new role. The page has a sidebar with a search icon and a list icon. The main content area is titled 'Name, review, and create' and contains a 'Role details' section. This section has two input fields: 'Role name' with the value 'ishan-iam-role' and 'Description' with the value 'Allows EC2 instances to call AWS services on your behalf.' Below the description field is a 'Step 1: Select trusted entities' section with an 'Edit' button.

#### Attach IAM Role to the EC2 Instance

##### 1. Open the EC2 Management Console:

- Go to the AWS EC2 Console at [AWS EC2 Console](#).




##### 2. Attach IAM Role:

- With your instance selected, click the Actions button, go to Security, and then click Modify IAM role.


EC2 > Instances > i-081ec58d070a138aa > Modify IAM role


## Modify IAM role [Info](#)

Attach an IAM role to your instance.

Instance ID  
 i-081ec58d070a138aa (ishand15c21casestudy)

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](#)

 If you choose **No IAM Role**, any IAM role that is currently attached to the instance will be removed. Are you sure you want to remove from the selected instance?


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

- In the pop-up, select the IAM role you created (ishan-iam-role).


EC2 > Instances > i-06dddbf880648ceab > Modify IAM role

## Modify IAM role [Info](#)

Attach an IAM role to your instance.

Instance ID  
 i-06dddbf880648ceab (ishan-ec2)

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](#)

- Click Update IAM role.

[Update IAM role](#)

Install CodeDeploy Agent on EC2

1. SSH into Your EC2 Instance:

Use SSH to connect to your instance:

bash

Copy code

ssh -i "your-key.pem" ec2-user@<public-ip-address>

```
C:\Windows\System32>ssh -i "new.pem" ec2-user@ec2-34-238-164-37.compute-1.amazonaws.com
Warning: Identity file new.pem not accessible: No such file or directory.
The authenticity of host 'ec2-34-238-164-37.compute-1.amazonaws.com (34.238.164.37)' can't be established.
ED25519 key fingerprint is SHA256:5RgbfFDLK9gGwp/7Lh00WAKMrOE7qwTff62C2jEVNNE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-34-238-164-37.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
ec2-user@ec2-34-238-164-37.compute-1.amazonaws.com: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
```

2. Update the Instance:

Update the package lists:

sudo yum update -y

```
[ec2-user@ip-172-31-45-252 ~]$ sudo yum update -y
Last metadata expiration check: 0:12:12 ago on Sat Oct 19 09:20:12 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-45-252 ~]$
```

3. Install Required Dependencies:

Install Ruby (required by CodeDeploy):

sudo yum install ruby -y

```
[ec2-user@ip-172-31-45-252 ~]$ sudo yum install ruby -y
Last metadata expiration check: 0:12:29 ago on Sat Oct 19 09:20:12 2024.
Dependencies resolved.
=====
Package                                Architecture      Version            Repository          Size
-----
Installing:
ruby3.2                                x86_64            3.2.2-180.amzn2023.0.3  amazonlinux         43
Installing dependencies:
ruby3.2-default-gems                   noarch            3.2.2-180.amzn2023.0.3  amazonlinux         34
ruby3.2-libs                           x86_64            3.2.2-180.amzn2023.0.3  amazonlinux         4.0
ruby3.2-rubygem-io-console             x86_64            0.6.0-180.amzn2023.0.3  amazonlinux         25
ruby3.2-rubygem-json                   x86_64            2.6.3-180.amzn2023.0.3  amazonlinux         53
ruby3.2-rubygem-psych                  x86_64            5.0.1-180.amzn2023.0.3  amazonlinux         52
Installing weak dependencies:
ruby3.2-rubygem-bigdecimal             x86_64            3.1.3-180.amzn2023.0.3  amazonlinux         69
ruby3.2-rubygem-bundler                noarch            2.4.10-180.amzn2023.0.3  amazonlinux        384
ruby3.2-rubygem-rdoc                   noarch            6.5.0-180.amzn2023.0.3  amazonlinux        461
ruby3.2-rubygems                       noarch            3.4.10-180.amzn2023.0.3  amazonlinux        257
=====

Installed:
ruby3.2-3.2.2-180.amzn2023.0.3.x86_64
ruby3.2-libs-3.2.2-180.amzn2023.0.3.x86_64
ruby3.2-rubygem-bundler-2.4.10-180.amzn2023.0.3.noarch
ruby3.2-rubygem-json-2.6.3-180.amzn2023.0.3.x86_64
ruby3.2-rubygem-rdoc-6.5.0-180.amzn2023.0.3.noarch
ruby3.2-rubygem-3.4.10-180.amzn2023.0.3.noarch
ruby3.2-default-gems-3.2.2-180.amzn2023.0.3.noarch
ruby3.2-rubygem-bigdecimal-3.1.3-180.amzn2023.0.3.x86_64
ruby3.2-rubygem-io-console-0.6.0-180.amzn2023.0.3.x86_64
ruby3.2-rubygem-psych-5.0.1-180.amzn2023.0.3.x86_64
ruby3.2-rubygems-3.4.10-180.amzn2023.0.3.noarch

Complete!
[ec2-user@ip-172-31-45-252 ~]$
```

Install wget to download CodeDeploy agent:

sudo yum install wget -y

```
[ec2-user@ip-172-31-45-252 ~]$ sudo yum install wget -y
Last metadata expiration check: 0:13:29 ago on Sat Oct 19 09:20:12 2024.
Package wget-1.21.3-1.amzn2023.0.4.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-45-252 ~]$
```

#### 4. Download and Install CodeDeploy Agent:

Download the CodeDeploy installer:

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

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

```
[ec2-user@ip-172-31-45-252 ~]$ cd /home/ec2-user
wget https://aws-codedeploy-us-east-1.s3.amazonaws.com/latest/install
--2024-10-19 09:34:04-- https://aws-codedeploy-us-east-1.s3.amazonaws.com/latest/install
Resolving aws-codedeploy-us-east-1.s3.amazonaws.com (aws-codedeploy-us-east-1.s3.amazonaws.com)... 54.231.137.129, 16.182.103.97, 52.217.169.17, ...
Connecting to aws-codedeploy-us-east-1.s3.amazonaws.com (aws-codedeploy-us-east-1.s3.amazonaws.com)|54.231.137.129|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 19045 (19K) []
Saving to: 'install'

install                               100%[=====>] 18.60K  --.-KB/s  in 0s

2024-10-19 09:34:04 (270 MB/s) - 'install' saved [19045/19045]

[ec2-user@ip-172-31-45-252 ~]$
```

Make the install file executable:

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

```
[ec2-user@ip-172-31-45-252 ~]$ chmod +x ./install
```

Run the installation:

```
sudo ./install auto
```

```
[ec2-user@ip-172-31-45-252 ~]$ wget https://aws-codedeploy-us-east-1.s3.amazonaws.com/latest/install
--2024-10-19 09:34:58-- https://aws-codedeploy-us-east-1.s3.amazonaws.com/latest/install
Resolving aws-codedeploy-us-east-1.s3.amazonaws.com (aws-codedeploy-us-east-1.s3.amazonaws.com)... 3.5.29.54, 3.5.29.219, 16.15.216.235, ...
Connecting to aws-codedeploy-us-east-1.s3.amazonaws.com (aws-codedeploy-us-east-1.s3.amazonaws.com)|3.5.29.54|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 19045 (19K) []
Saving to: 'install.1'

install.1                               100%[=====>] 18.60K  --.-KB/s  in 0s

2024-10-19 09:34:58 (102 MB/s) - 'install.1' saved [19045/19045]

[ec2-user@ip-172-31-45-252 ~]$ chmod +x ./install
[ec2-user@ip-172-31-45-252 ~]$ sudo ./install auto
I, [2024-10-19T09:35:32.625339 #26587] INFO -- : Starting Ruby version check.
W, [2024-10-19T09:35:32.625542 #26587] WARN -- : The Ruby version in /usr/bin/ruby3.2 is 3.2.2, . Attempting to install anyway.
I, [2024-10-19T09:35:32.625616 #26587] INFO -- : Starting update check.
I, [2024-10-19T09:35:32.625672 #26587] INFO -- : Attempting to automatically detect supported package manager type for system...
I, [2024-10-19T09:35:32.639121 #26587] INFO -- : Checking AWS_REGION environment variable for region information...
I, [2024-10-19T09:35:32.639252 #26587] INFO -- : Checking EC2 metadata service for region information...
I, [2024-10-19T09:35:32.648780 #26587] INFO -- : Checking AWS_DOMAIN environment variable for domain information...
I, [2024-10-19T09:35:32.648928 #26587] INFO -- : Checking EC2 metadata service for domain information...
I, [2024-10-19T09:35:32.653372 #26587] INFO -- : Downloading version file from bucket aws-codedeploy-us-east-1 and key latest/LATEST_VERSION...
I, [2024-10-19T09:35:32.653577 #26587] INFO -- : Endpoint: https://aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com/latest/LATEST_VERSION
```



```
Installed:
  codedeploy-agent-1.7.0-92.noarch
```

```
Complete!
```

```
I, [2024-10-19T09:35:35.867442 #26587] INFO -- : Update check complete.
I, [2024-10-19T09:35:35.867604 #26587] INFO -- : Stopping updater.
[ec2-user@ip-172-31-45-252 ~]$
```

---

○

## 5. Start CodeDeploy Agent:

- Start the CodeDeploy agent service:

```
sudo service codedeploy-agent start
```

```
[ec2-user@ip-172-31-45-252 ~]$ sudo service codedeploy-agent start
Starting codedeploy-agent:[ec2-user@ip-172-31-45-252 ~]$
```

Verify CodeDeploy Agent is Running

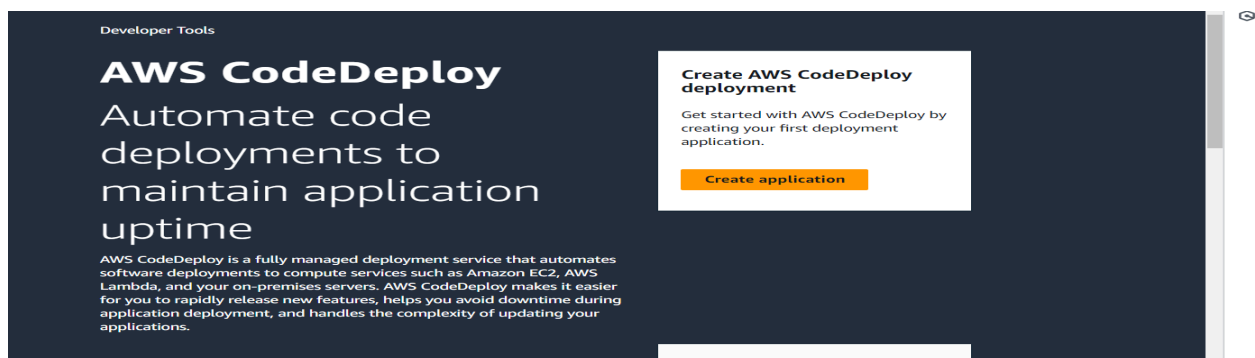
- Check the status of the CodeDeploy agent to ensure it's running:

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

```
[ec2-user@ip-172-31-45-252 ~]$ sudo service codedeploy-agent start
Starting codedeploy-agent:[ec2-user@ip-172-31-45-252 ~]$ sudo service codedeploy-agent status
The AWS CodeDeploy agent is running as PID 26670
[ec2-user@ip-172-31-45-252 ~]$
```

Part 6 : Create a CodeDeploy Application:

- Open the AWS CodeDeploy Console.



- Click Create Application.

## Create application

- Name your application (e.g., ishan-codedeploy).

### Create application

#### Application configuration

Application name

Enter an application name

100 character limit

Compute platform

- For the Compute Platform, choose EC2/On-premises.

Compute platform

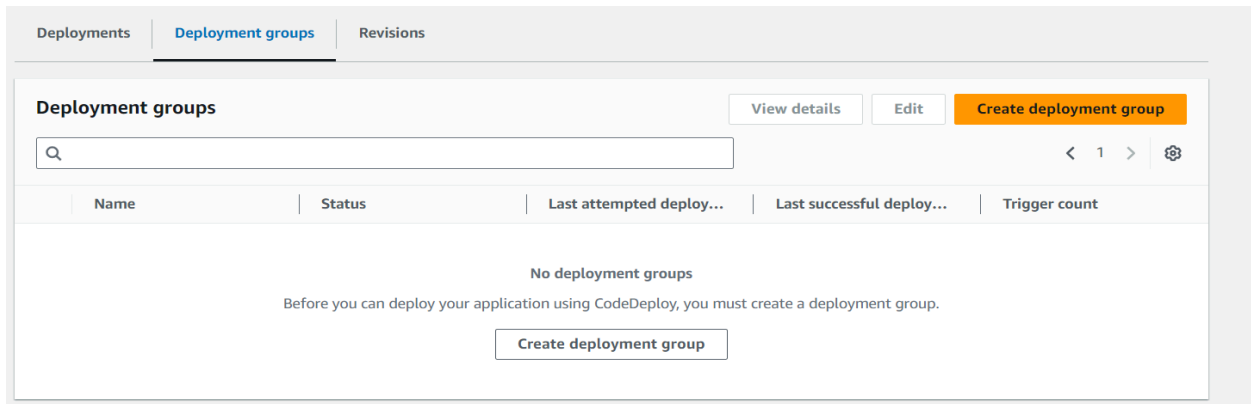
Choose a compute platform

- Click Create application.

## Create application

Create a Deployment Group:

- In your CodeDeploy application, click Create deployment group.



- Name the group (e.g., ishan-deployment-group).

**Deployment group name**

Enter a deployment group name

ishan-deployment-group

100 character limit

- Choose the EC2 instance as the environment.

**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	
<input type="text" value="Name"/>	<input type="text" value="ishan-ec2"/>	<input type="button" value="Remove tag"/>

☐ On-premises instances

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

- Select the Service Role that has the appropriate permissions (e.g., AWSCodeDeployRole).

## 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::850995542805:role/ishan-iam-role



- Click Create deployment group.

Create deployment group

Success

Deployment group created

[Developer Tools](#) > [CodeDeploy](#) > [Applications](#) > [ishan261104-codedepoly](#) > ishan-deployment-group

ishan-deployment-group

EditDeleteCreate deployment

Deployment group details

Deployment group name	Application name	Compute platform
ishan-deployment-group	<a href="#">ishan261104-codedepoly</a>	EC2/On-premises
Deployment type	Service role ARN	Deployment configuration
In-place	<a href="#">arn:aws:iam::692859933305:role/ishan2611</a>	<a href="#">CodeDeployDefault.AllAtOnce</a>
Rollback enabled	Agent update scheduler	
False	<a href="#">Learn to schedule update in AWS Systems Manager</a>	

## Create a Deployment for Your Application

Click on the Deployments tab.

Click on Create deployment.

[Developer Tools](#) > [CodeDeploy](#) > [Applications](#) > [ishan261104-codedepoly](#) > ishan-deployment-group

ishan-deployment-group

EditDeleteCreate deployment

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.](#)  
-

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

## Deployment

[Developer Tools](#) > [CodeDeploy](#) > [Deployments](#) > [d-7A66MB909](#) > arn:aws:ec2:us-east-1:850995542805:instance/i-06dddbf880648ceab

arn:aws:ec2:us-east-1:850995542805:instance/i-06dddbf880648ceab

### Deployment details

Application  
[ishan-codedeploy](#)

Deployment ID  
[d-7A66MB909](#)

Status  
✔ Succeeded

Deployment configuration  
[CodeDeployDefault.AllAtOnce](#)

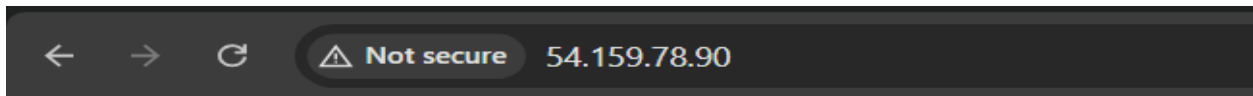
Deployment group  
[ishan-deployment-group](#)

Initiated by  
User action

Deployment description  
-

Event	Duration	Status	Error code	Start time	End time
ApplicationStop	less than one second	✔ Succeeded	-	Oct 19, 2024 10:06 PM (UTC+5:30)	Oct 19, 2024 10:06 PM (UTC+5:30)
DownloadBundle	less than one second	✔ Succeeded	-	Oct 19, 2024 10:06 PM (UTC+5:30)	Oct 19, 2024 10:06 PM (UTC+5:30)
BeforeInstall	1 second	✔ Succeeded	-	Oct 19, 2024 10:06 PM (UTC+5:30)	Oct 19, 2024 10:06 PM (UTC+5:30)
Install	less than one second	✔ Succeeded	-	Oct 19, 2024 10:06 PM (UTC+5:30)	Oct 19, 2024 10:06 PM (UTC+5:30)
AfterInstall	less than one second	✔ Succeeded	-	Oct 19, 2024 10:06 PM (UTC+5:30)	Oct 19, 2024 10:06 PM (UTC+5:30)
ApplicationStart	less than one second	✔ Succeeded	-	Oct 19, 2024 10:06 PM (UTC+5:30)	Oct 19, 2024 10:06 PM (UTC+5:30)
ValidateService	less than one second	✔ Succeeded	-	Oct 19, 2024 10:06 PM (UTC+5:30)	Oct 19, 2024 10:06 PM (UTC+5:30)

## Deployed Web Page



# Hello, I am Ishan

**-Conclusion:** In this case study, we focused on automating cloud deployment using several AWS services, mainly AWS CodePipeline, EC2, S3, and CodeDeploy. Our goal was to build a simple web application with a sample index.html page and show how these services can work together for an easy deployment process. We started by creating an S3 bucket for hosting our static website. After setting it up, we uploaded our HTML file and made sure the permissions allowed public access so users could visit the website. Next, we used AWS CodeBuild to create a build project that builds our application code. This step is must because it automates the build process, making sure that only our code is deployed. Then, we set up AWS CodePipeline, which helps us manage the whole deployment flow. We connected the source stage to our S3 bucket and the build stage to our CodeBuild project, making the process easier. We then launched an EC2 instance to host our application. We created an IAM role to give the instance the right permissions to access S3 and CodeDeploy services. After installing the CodeDeploy agent on the EC2 instance, we were able to manage deployments easily. We created a CodeDeploy application and a deployment group linked to our EC2 instance. Finally, we started a deployment to push our application to the EC2 instance and checked that everything was working properly. Overall, this case study showed how effective AWS tools can be for automating deployment.