

We will build this in two parts:

1. **Pipeline 1:** A simple pipeline to deploy a static website to an **S3 bucket**.
2. **Pipeline 2:** A more advanced pipeline to deploy an application to an **EC2 instance** using CodeDeploy.

---

## Prerequisites

Before you get started, ensure you have the following set up:

- An **AWS Account** with administrative access.
- A **GitHub Account**.
- **Git** installed and configured on your local machine.
- A text editor, such as **VS Code**.

---

## Part 1: GitHub Repository Setup

First, we'll create a simple web application and the necessary configuration files for our deployment pipelines.

### Step 1: Create a Sample Application

1. On your local machine, create a new MainFolder (e.g., `aws-pipeline-demo`).
2. Inside this folder, create a file named `index.html` with the following content:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>AWS Pipeline Demo</title>
  <style>
    body { font-family: sans-serif; text-align: center; margin-top: 100px; background-color:
#232F3E; color: #FFFFFF; }
    h1 { color: #FF9900; }
  </style>
</head>
<body>
  <h1>Welcome to My Application!</h1>
  <p>This page was deployed successfully using AWS CodePipeline!</p>
</body>
</html>

```

## Step 2: Create the buildspec.yml for CodeBuild

This file tells **AWS CodeBuild** what commands to run during the build phase. In the **same folder**, create a file named `buildspec.yml`

```

version: 0.2
phases:
  install:
    runtime-versions:
      nodejs: 18
  build:
    commands:
      - echo Build started on `date`
      - echo "No build steps needed for static HTML"
  post_build:
    commands:
      - echo Build completed on `date`

artifacts:
  files:
    - '**/*'

```

- **artifacts:** This section tells CodeBuild to bundle all files (\*\*/\*) into a single output artifact (.zip) for the deployment stage.

### Step 3: Create the appspec.yml for CodeDeploy

This file tells **AWS CodeDeploy** how to deploy the application onto an EC2 instance. Create a file named `appspec.yml` in the same folder.

```
version: 0.0
os: linux
files:
  - source: /
    destination: /var/www/html/
hooks:
  BeforeInstall:
    - location: scripts/install_dependencies.sh
      timeout: 300
      runas: root
  ApplicationStart:
    - location: scripts/start_server.sh
      timeout: 300
      runas: root
```

Now, create a new folder named `scripts` inside the MainFolder(`aws-pipeline-demo`).

1. `scripts/install_dependencies.sh`: This script installs the Apache web server.

```
#!/bin/bash
yum update -y
yum install -y httpd
```

2. `scripts/start_server.sh`: This script starts and enables the web server.

## Step 4: Push to GitHub

Initialize a Git repository, commit your files, and push them to a new repository on your GitHub account.

```
cd (path of your MainFolder)
```

```
# Initialize a new git repository
git init
```

```
# Add all files to the staging area
git add .
```

```
# Commit the files
git commit -m "Initial commit with sample app and config files"
```

```
# Add your GitHub repository as the remote origin and push
git remote add origin https://github.com/YOUR_USERNAME/YOUR_REPO_NAME.git
git branch -M main
git push -u origin main
```

Your final repository structure should look like this:

```
/
├── index.html
├── buildspec.yml
├── appspec.yml
├── scripts/
│   ├── install_dependencies.sh
│   └── start_server.sh
```

---

## Part 2: Pipeline for Deploying to Amazon S3


This pipeline will automatically deploy your `index.html` file to an S3 bucket configured for static website hosting.

### Step 1: Create and Configure the S3 Bucket

1. Navigate to the **S3** service in the AWS Console.
2. Click **Create bucket**.
3. Give it a **globally unique name** (e.g., `my-codepipeline-demo-site-unique-name`).

## Bucket name [Info](#)

himanshu-bucket12

Bucket names must be 3 to 63 characters and unique within the global namespace. Bucket names must also begin and end with a letter or number. Valid characters are a-z, 0-9, periods (.), and hyphens (-). [Learn More](#) 

## Copy settings from existing bucket - *optional*


Only the bucket settings in the following configuration are copied.

[Choose bucket](#)

Format: s3://bucket/prefix

4. **Uncheck "Block all public access". Acknowledge the warning.** This is necessary to make your website public.

## 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. If objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommender applying any of these settings, ensure that your applications will work correctly without public access. If you require some level 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 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

#### ☐ 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.

5. Click **create bucket**.
6. Go into your new bucket, click the **Properties** tab.

<input type="radio"/>	<a href="#">elasticbeanstalk-us-east-1-339932683315</a>	US East (N. Virginia) us-east-1	August 11, 2025, 09:32:58 (UTC+05:30)
<input type="radio"/>	<a href="#">himanshu-bucket12</a>	US East (N. Virginia) us-east-1	August 13, 2025, 19:34:17 (UTC+05:30)

7. Scroll down to **Static website hosting** and click **Edit**.
8. **Enable it**, set the **Index document** to `index.html`, and save changes.

### Index document

Specify the home or default page of the website.

index.html

9. Go to the **Permissions** tab, click **Edit** under **Bucket policy**, and paste the following policy. Replace `YOUR_BUCKET_NAME` with your bucket's name.

```
{  
  "Version": "2012-10-17",
```

```

"Statement": [
  {
    "Sid": "PublicReadGetObject",
    "Effect": "Allow",
    "Principal": "*",
    "Action": "s3:GetObject",
    "Resource": "arn:aws:s3:::YOUR_BUCKET_NAME/*"
  }
]
}

```

### Policy

```

1 ▼ {
2     "Version": "2012-10-17",
3 ▼     "Statement": [
4 ▼         {
5             "Sid": "PublicReadGetObject",
6             "Effect": "Allow",
7             "Principal": "*",
8             "Action": "s3:GetObject",
9             "Resource": "arn:aws:s3:::himanshu-bucket12/*"
10        }
11    ]
12 }

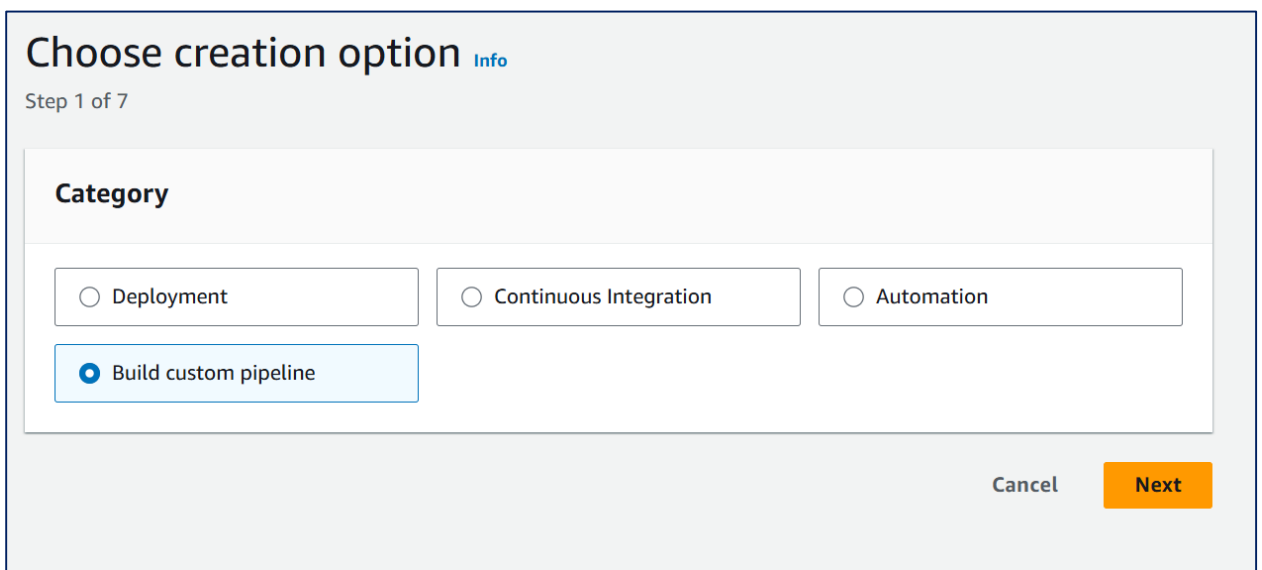
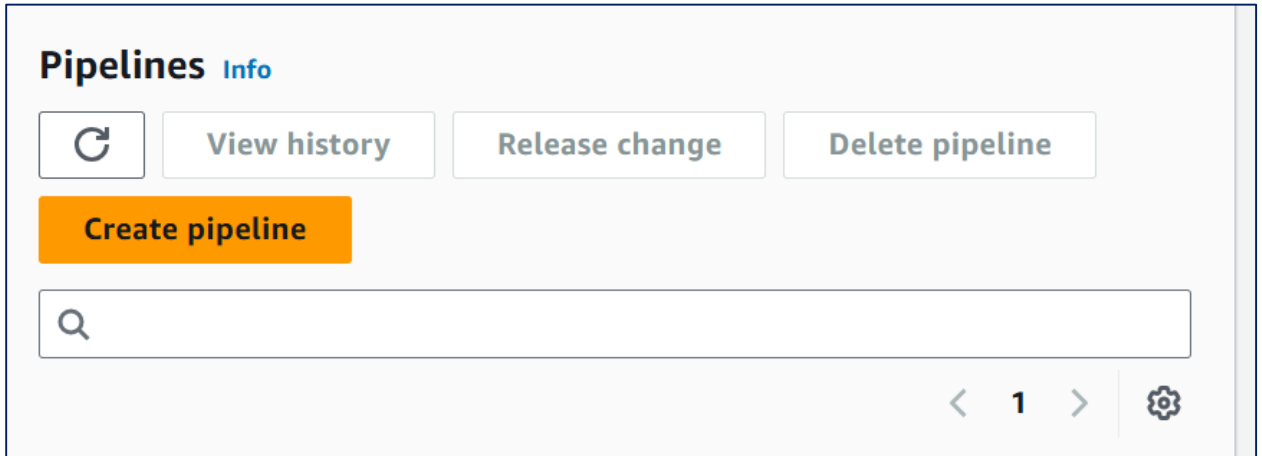
```

10. Save the policy.

## Step 2: Create the AWS CodePipeline



1. Navigate to the **CodePipeline** service in the AWS Console.
2. Click **Create pipeline**.



3. **Pipeline settings:**
  - **Pipeline name:** **My-S3-Website-Pipeline**.
  - Leave **Service role** as is; CodePipeline will create a new one for you.
  - Click **Next**.
4. **Source stage:**
  - **Source provider:** Select **GitHub (via GitHub App)**.
  - Click **Create a connection**. A new window will open. Name your

connection (e.g., `my-github-connection`) and connect to your GitHub account.

## Source

### Source provider

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

GitHub (via GitHub App) ▼

### Connection

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

Q

↺

or

Connect to GitHub

### Repository name

Choose a repository in your GitHub account.

Q

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

## Create a connection [Info](#)

### Create GitHub App connection [Info](#)

Connection name

h1connection

► **Tags - optional**

**Connect to GitHub**

## Source

### Source provider

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

GitHub (via GitHub App) ▼

### Connection

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

Q arn:aws:codeconnections:us-east-1:339932683315:connection/a8b9 X



or

**Connect to GitHub**

### Repository name

Choose a repository in your GitHub account.

Q himanshu5607/aws9 X

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

### Default branch

Default branch will be used only when pipeline execution starts from a different source or manually started.

Q master X

- Once connected, select your **Repository name** and the **Branch name** (main or master).
- Ensure **"Start your pipeline on push and pull request events."** is checked.

**Webhook events**

Webhook - *optional*

☒ Start your pipeline on push and pull request events.

► Webhook event filters - *optional*

Starts your pipeline on a specific event.

Remove filters

- Click **Next**.

## 5. Build stage:

- **Build provider:** Select **AWS CodeBuild**.
- Click **Create project**. A new window will appear.

**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.

or

Create project 

- **Project name:** `my-s3-build-project`.
- **Environment image:** Managed image.
- **Operating system:** Amazon Linux 2.
- **Runtime(s):** Standard.
- **Image:** `aws/codebuild/amazonlinux2-x86_64-standard:5.0`.
- **Buildspec:** Check "Use a buildspec file". CodeBuild will automatically look for `buildspec.yml` in your repo's root.

**▼ 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

**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`

- Click **Continue to CodePipeline**.
- Click **Next**.

6. **Deploy stage:**

- **Deploy provider:** Select **Amazon S3**.
- **Region:** Your current region.


**Deploy - optional**

**Deploy provider**  
Choose how you want to deploy your application or content. Choose the provider, and then provide the configuration details for that provider.


Amazon S3 ▼

**Region**

United States (N. Virginia) ▼

**Input artifacts**  
Choose an input artifact for this action. [Learn more](#) 

▼

BuildArtifact 

Defined by: Build

No more than 100 characters

- **Bucket:** Select the S3 bucket you created earlier.
- Check the box for "**Extract file before deploy**". This unzips the build artifact and places the files directly in the bucket.

Bucket

himanshu-bucket12

Deployment path - *optional*

☒ Extract file before deploy  
The deployed artifact will be unzipped before deployment.

► Additional configuration

☒ Configure automatic rollback on stage failure

☐ Enable automatic retry on stage failure

Cancel Previous Skip deploy stage Next

- Click **Next**.
7. **Review** all the settings and click **Create pipeline**.

Success  
Congratulations! The pipeline My-S3-Website-Pipeline has been created.

Developer Tools > CodePipeline > Pipelines > My-S3-Website-Pipeline

My-S3-Website-Pipeline

Edit Stop execution Create trigger Clone pipeline Release change

Pipeline Executions Triggers Settings Tags Stage

Source Build Deploy

010d7bf5-4ca7-4f50-a9b5-aad82c407dfd  
All actions succeeded.

Source  
GitHub (via GitHub App)  
1 minute ago

5036fcd7 Source: aws9

010d7bf5-4ca7-4f50-a9b5-aad82c407dfd  
All actions succeeded.

Build  
AWS CodeBuild  
Just now

5036fcd7 Source: aws9

010d7bf5-4ca7-4f50-a9b5-aad82c407dfd  
All actions succeeded.

Deploy  
Amazon S3  
Just now

5036fcd7 Source: aws9

The pipeline will automatically start its first run. You can watch it progress through the stages. Once it succeeds, go to your S3 bucket's static website URL (from **Properties** -> **Static website hosting**) to see your live page!

## Static website hosting

Edit

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

**i We recommend using AWS Amplify Hosting for static website hosting**  
Deploy a fast, secure, and reliable website quickly with AWS Amplify Hosting. Learn more about [Amplify Hosting](#) or [View your existing Amplify apps](#)


Create Amplify app

**S3 static website hosting**  
Enabled

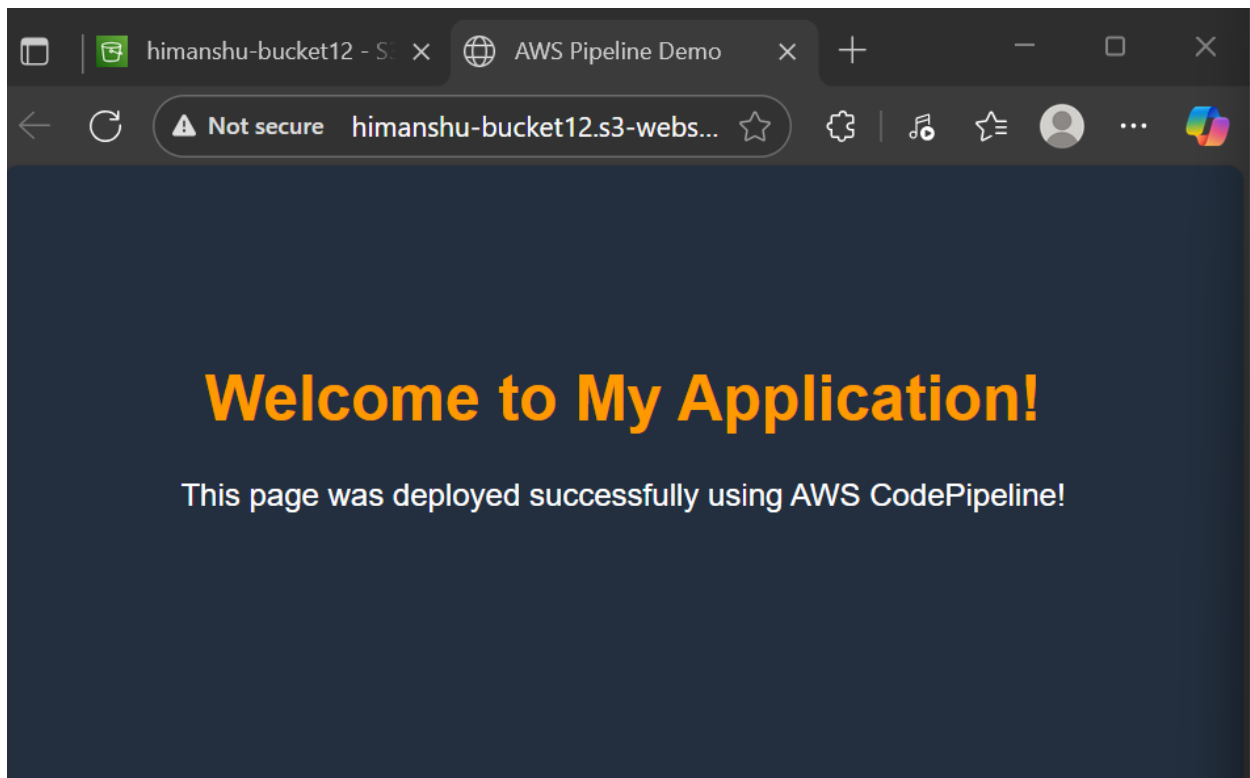
**Hosting type**  
Bucket hosting

**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](#)

<http://himanshu-bucket12.s3-website-us-east-1.amazonaws.com>







---

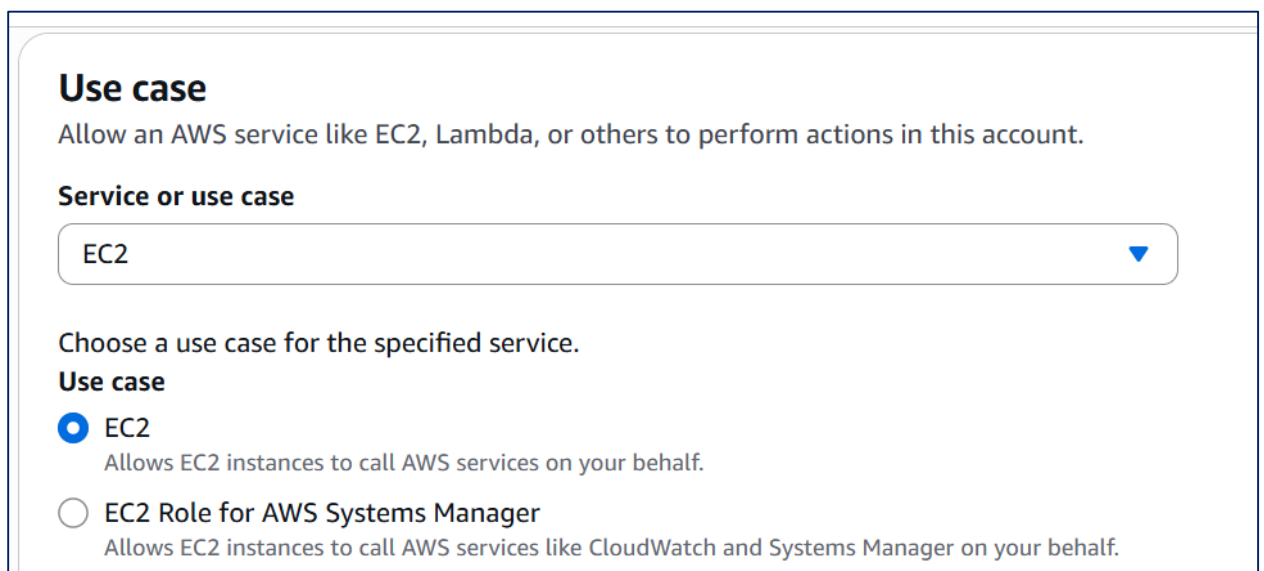
## Part 3: Pipeline for Deploying to an EC2 Instance

This pipeline uses the same source and build steps but adds a CodeDeploy stage to deploy the files to a running EC2 instance.

### Step 1: Create an IAM Role (i)

The EC2 instance needs permission to communicate with the CodeDeploy service.

1. Navigate to the **IAM** service.
2. Go to **Roles** and click **Create role**.
3. **Trusted entity type**: Select **AWS service**.
4. **Use case**: Select **EC2**. Click **Next**.
5. Search for and add the permission policy:  
[AmazonEC2RoleforAWSCodeDeploy](#). Click **Next**.



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

**Service or use case**

EC2 ▼

Choose a use case for the specified service.

**Use case**

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

☐ **EC2 Role for AWS Systems Manager**  
Allows EC2 instances to call AWS services like CloudWatch and Systems Manager on your behalf.

## Add permissions Info

### Permissions policies (2/1074) Info

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

×

Filter by Type

All types

2 matches

< 1 >
⚙️

<input checked="" type="checkbox"/>	Policy name <small>🔗</small>	Type	Description
<input checked="" type="checkbox"/>	AmazonEC2Rolefo...	AWS managed	Provides EC2 access to S3 bucket to download revision. This rol...
<input checked="" type="checkbox"/>	AmazonEC2Rolefo...	AWS managed	Provides EC2 limited access to S3 bucket to download revision. ...

▶ Set permissions boundary - optional

Cancel

Previous

Next

## 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.

## Step 2: Create an IAM Role (ii)

The EC2 instance needs permission to communicate with the CodeDeploy service.

- Navigate to the **IAM** service.
- Go to **Roles** and click **Create role**.
- Trusted entity type**: Select **AWS service**.
- Use case**: Select CodeDeploy. Click **Next**.

## 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.



##### CodeDeploy for Lambda

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



##### CodeDeploy - ECS

Allows CodeDeploy to read S3 objects, invoke Lambda functions, publish to SNS topics, and update ECS services on your behalf.



[Cancel](#)

[Next](#)

### Add permissions [Info](#)

#### Permissions policies (1) [Info](#)

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

Policy name <a href="#">?</a>	Type
  <a href="#">AWSCodeDeployRole</a>	AWS managed

► Set permissions boundary - *optional*

[Cancel](#) [Previous](#) [Next](#)

### 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.

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

## Step 3: Launch and Prepare the EC2 Instance

### Step 2: Launch and Prepare the EC2 Instance

1. Navigate to the **EC2** service and click **Launch instances**.
2. **Name:** `My-CodeDeploy-Server`.

3. **AMI:** Select **Amazon Linux 2 AMI (HVM)** - Free tier eligible.
4. **Instance type:** **t2.micro** (Free tier eligible).
5. **Key pair:** Create or select an existing key pair to be able to SSH into the instance if needed.

### Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

#### Name and tags [Info](#)

**Name**

[Add additional tags](#)

### (Creating Key Pair)

**Key pair name**  
Key pairs allow you to connect to your instance securely.

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

**Key pair type**


☒ **RSA**  
RSA encrypted private and public key pair


☐ **ED25519**  
ED25519 encrypted private and public key pair

**Private key file format**

☒ **.pem**  
For use with OpenSSH

☐ **.ppk**  
For use with PuTTY

 When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.**

[Learn more](#) 

[Cancel](#) [Create key pair](#)

6. **Network settings:**
  - In the **Security group**, ensure "Allow HTTP traffic from the internet" is

checked. This opens port 80 so you can view the website.

**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  
Helps you connect to your instance  
Anywhere  
0.0.0.0/0

☐ 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. 

## 7. Advanced details:

- Expand this section and for **IAM instance profile**, select the **EC2-CodeDeploy-Role** you just created.

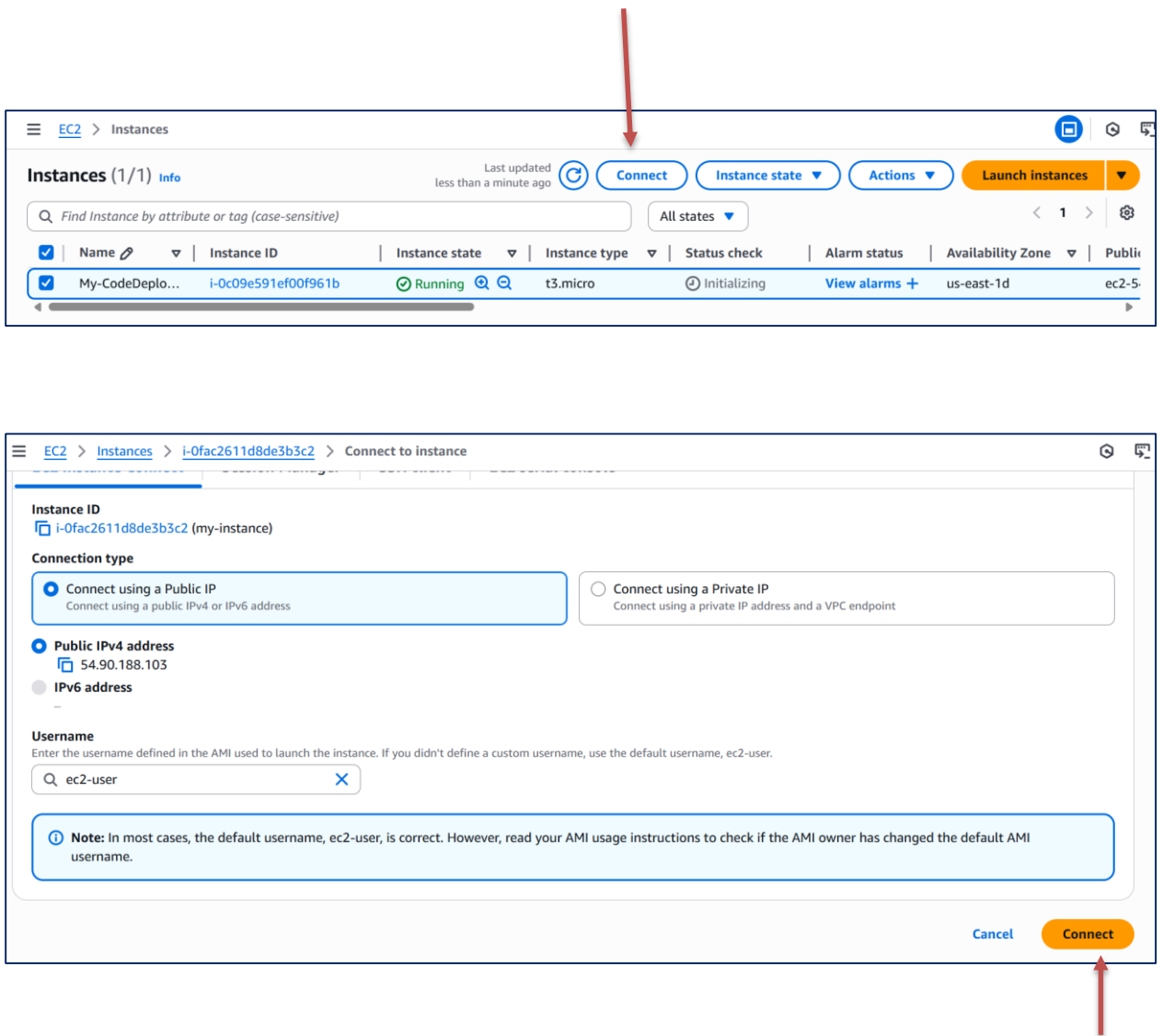
**▼ Advanced details** | [Info](#)

**Domain join directory** | [Info](#)  
Select   [Create new directory](#) 

**IAM instance profile** | [Info](#)  
h3-1e-role  
arn:aws:iam::339932683315:instance-profile/h3-1e-role   [Create new IAM profile](#) 

8. Click **Launch instance**.

9. Once the instance is running, select it and click **Connect**. Use the **EC2 Instance Connect** or your SSH client to connect to it.



10. In the instance's terminal, install the CodeDeploy agent:

```
#!/bin/bash
sudo yum update -y
sudo yum install ruby -y
sudo yum install wget -y
cd /home/ec2-user
wget https://aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com/latest/install
chmod +x ./install
sudo ./install auto
sudo service codedeploy-agent status
```



(Make sure the `codedeploy-agent` is running). The URL is for the `us-east-1` region; adjust it if you are in a different region.

```
I, [2025-08-13T16:08:26.041396 #4471] INFO -- : Stopping updater.  
The AWS CodeDeploy agent is running as PID 9090  
[ec2-user@ip-172-31-44-221 ~]$
```

## Step 4: Create the CodeDeploy Application

### Step 3: Create the CodeDeploy Application

1. Navigate to the **CodeDeploy** service.
2. On the left menu, select **Applications** and click **Create application**.
3. **Application name:** `My-Demo-Application`.
4. **Compute platform:** Select **EC2/On-premises**.
5. Click **Create application**.

### Application configuration

Application name

Enter an application name

My-Demo-Application1

100 character limit

Compute platform

Choose a compute platform

EC2/On-premises ▼

Tags

Add tag

Cancel

Create application

## Step 5: Create a Deployment Group

### Step 4: Create a Deployment Group

1. Inside your new application, click **Create deployment group**.

Deployment groups

View details

Edit

Create deployment group

< 1 > ⚙

Name	Status	Last atte...	Last succe...	Trigger count
<div>No deployment groups</div> <div>Before you can deploy your application using CodeDeploy, you must create a deployment group.</div> <div>Create deployment group</div>				

2. **Deployment group name:** `My-EC2-Deployment-Group`.
3. **Service Role:** Click **Create a new service role** or select an existing one with the `AWSCodeDeployRole` policy. This role gives CodeDeploy permission to interact with your EC2 instances.

### Deployment group name

Enter a deployment group name

100 character limit

### Service role

Enter a service role

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

4. **Deployment type:** In-place.
5. **Environment configuration:**
  - Check **Amazon EC2 instances**.
  - In the **Key** field, select **Name**. In the **Value** field, enter the name you gave your EC2 instance: **My-CodeDeploy-Server**. This tells CodeDeploy which instances to target.

☐ 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="My-CodeDeploy-Server"/>	<input type="button" value="Remove tag"/>


6. **Deployment settings:** Choose **CodeDeployDefault.AllAtOnce**.
7. **Load Balancer:** Uncheck "Enable load balancing".

### 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

8. Click **Create deployment group**.

 **Success**  
Deployment group created

[Developer Tools](#) > [CodeDeploy](#) > [Applications](#) > [My-Demo-Application1](#) > My-demo-Group

## My-demo-Group

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

### Deployment group details

Deployment group name	Application name	Compute platform
My-demo-Group	<a href="#">My-Demo-Application1</a>	EC2/On-premises
Deployment type	Service role ARN	Deployment configuration
In-place	<a href="#">arn:aws:iam::339932683315:role/h1role</a>	<a href="#">CodeDeployDefault.AllAtOnce</a>
Rollback enabled	Agent update scheduler	
False	<a href="#">Learn to schedule update in AWS Systems Manager</a>	

## Step 6: Create the Full Pipeline for EC2

Now we'll create a new pipeline that includes the CodeDeploy stage.

1. Go back to **CodePipeline** and click **Create pipeline**.

# Choose creation option [Info](#)

Step 1 of 7

## Category

☐ Deployment

☐ Continuous Integration

☐ Automation

☒ Build custom pipeline

Cancel

Next

2. Pipeline name: **My-EC2-App-Pipeline**.

## Pipeline settings

### Pipeline name

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

My-EC2-App-Pipeline9

No more than 100 characters

### Execution mode [Info](#)

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

☐ Superseded

☒ Queued

☐ Parallel

### Service role

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

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

3. **Source Stage:** Configure it exactly as before, connecting to your GitHub repository.
4. **Build Stage:** Configure it exactly as before, selecting the same **AWS CodeBuild** project (`my-s3-build-project`). The build artifact works for both S3 and CodeDeploy.

## Add source stage [Info](#)

Step 3 of 7

### Source

#### Source provider

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

GitHub (via GitHub App) ▼

#### Connection

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



or

**Connect to GitHub**

#### Repository name

Choose a repository in your GitHub account.



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

### Source

#### Source provider

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

GitHub (via GitHub App) ▼

#### Connection

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



arn:aws:codeconnections:us-east-1:123456789012:codeconnections:my-connection X



or

**Connect to GitHub**

#### Repository name

Choose a repository in your GitHub account.



himanshu5607/aws9 X

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

#### Default branch

Default branch will be used only when pipeline execution starts from a different source or manually started.



master X

## Add build stage [Info](#)

Step 4 of 7

### Build - *optional*

#### Build provider

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

☐ Commands

☒ Other build providers

AWS CodeBuild ▼

#### 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.



or

Create project [↗](#)

☐ Define buildspec override - *optional*

Buildspec file or definition that overrides the latest one defined in the build project, for this build only.

#### 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](#) [↗](#)

## Create build project

### Project configuration

#### Project name

h1-project

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 \_.

#### Project type

Select what type of project you would like to create. [Info](#) [↗](#)

☒ Default project

Create a custom CodeBuild project.

☐ Runner project

Create a CodeBuild managed runner for workflows in GitHub Actions, GitHub Enterprise Actions, GitLab, or Buildkite.

#### ► Additional configuration

Description, public build access, build badge, concurrent build limit, tags



► **Additional configuration**  
Description, public build access, build badge, concurrent build limit, tags

► **Environment**

▼ **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

**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

**CloudWatch**

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

**Group name - optional**

aws/codebuild/h1-project

The group name of the logs in CloudWatch Logs. The log group name will be /aws/codebuild/<project-name> by default.

**Stream name prefix - optional**

The prefix of the stream name of the CloudWatch Logs.

**S3**

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

Cancel

Continue to CodePipeline

Step 1

Choose creation option

Step 2

Choose pipeline settings

Step 3

Add source stage

Step 4

Add build stage

Step 5

Add test stage

Step 6

Add deploy stage

Step 7

Review

## Add build stage [Info](#)

Step 4 of 7

### Build - optional

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

☐ Commands

☒ Other build providers

AWS CodeBuild

**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.

or

☒ **Define buildspec override - optional**  
Buildspec file or definition that overrides the latest one defined in the build project, for this build only.

**Buildspec override**

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

☐ Insert build commands  
Store build commands as build project configuration

### Buildspec override

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

☐ Insert build commands  
Store build commands as build project configuration

**Buildspec name**  
Enter the path of your buildspec file from the source root (for example, configuration/buildspec.yml).



Build type

☒ Single build

Triggers a single build.

☐ Batch build

Triggers multiple builds as a single execution.

Region

United States (N. Virginia) ▼

Input artifacts

Choose an input artifact for this action. [Learn more](#)

SourceArtifact

Defined by: Source

☒ Enable automatic retry on stage failure

Cancel

Previous

Skip build stage

Next

5. **Deploy Stage:** This is where it changes.

- ☐ **Deploy provider:** Select **AWS CodeDeploy**.

Step 2

Choose pipeline settings

Step 3

Add source stage

Step 4

Add build stage

Step 5

Add test stage

Step 6

**Add deploy stage**

Step 7

Review

Deploy - optional

Deploy provider

Choose how you want to deploy your application or content. Choose the provider, and then provide the configuration details for that provider.

AWS CodeDeploy ▼

Region

United States (N. Virginia) ▼

Input artifacts

Choose an input artifact for this action. [Learn more](#)

BuildArtifact

Defined by: Build

No more than 100 characters

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.

Q my-1 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

**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.

Q my-1 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.

Q my-g X

my-g

☐ Enable automatic retry on stage failure

Cancel Previous Skip deployment stage Next

- **Application name:** Select **My-Demo-Application** from the dropdown.
- **Deployment group:** Select **My-EC2-Deployment-Group** from the dropdown.

**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.

Q My-Demo-Application1 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.

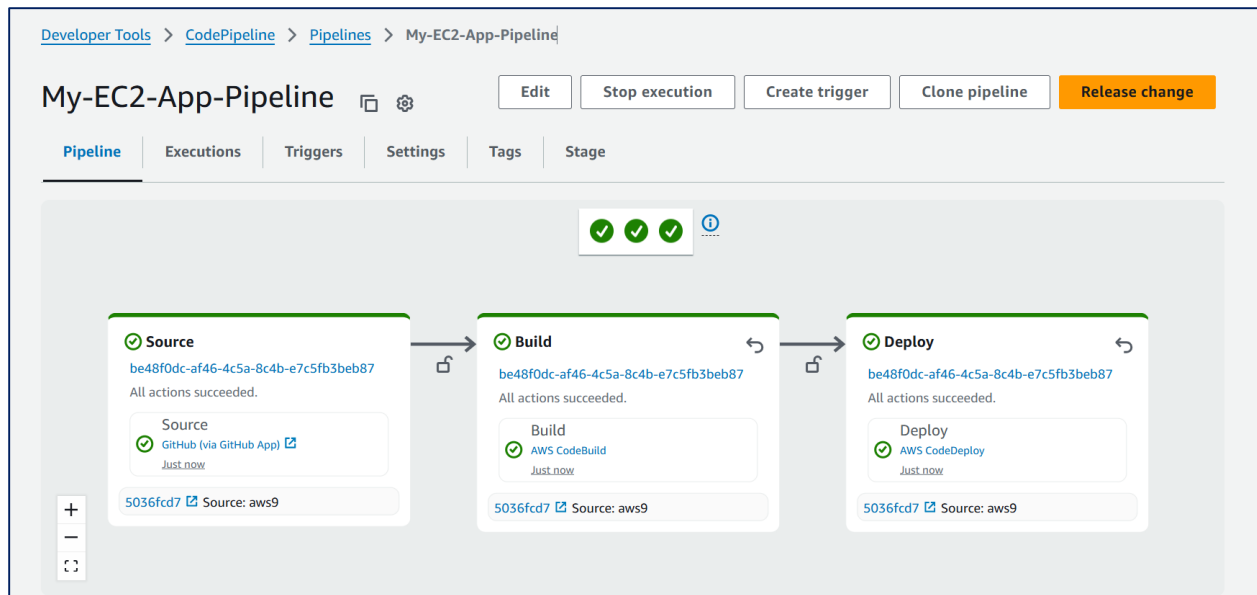
Q My-demo-Group X

☒ Configure automatic rollback on stage failure

☐ Enable automatic retry on stage failure

- Click **Next**.

6. **Review** and click **Create pipeline**.



The pipeline will run. The **Deploy** stage will now use CodeDeploy to copy the files to `/var/www/html/` on your EC2 instance and run the scripts in your `appspec.yml` file to start the Apache server.

To verify, get the **Public IPv4 address** from your EC2 instance's details page and paste it into your browser. You should see your sample web page!

The screenshot shows the AWS Management Console for an EC2 instance named 'kk-1e-server' with ID 'i-039fe4a24a64b27dc'. The instance is in a 'Running' state. A red arrow points to the 'Public IPv4 address' field, which shows '3.88.86.3'. The console also displays the instance's name, ID, type (t3.micro), status check (3/3 checks passed), alarm status, and availability zone (us-east-1d).

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
kk-1e-server	i-039fe4a24a64b27dc	Running	t3.micro	3/3 checks passed	View alarms	us-east-1d

**i-039fe4a24a64b27dc (kk-1e-server)**

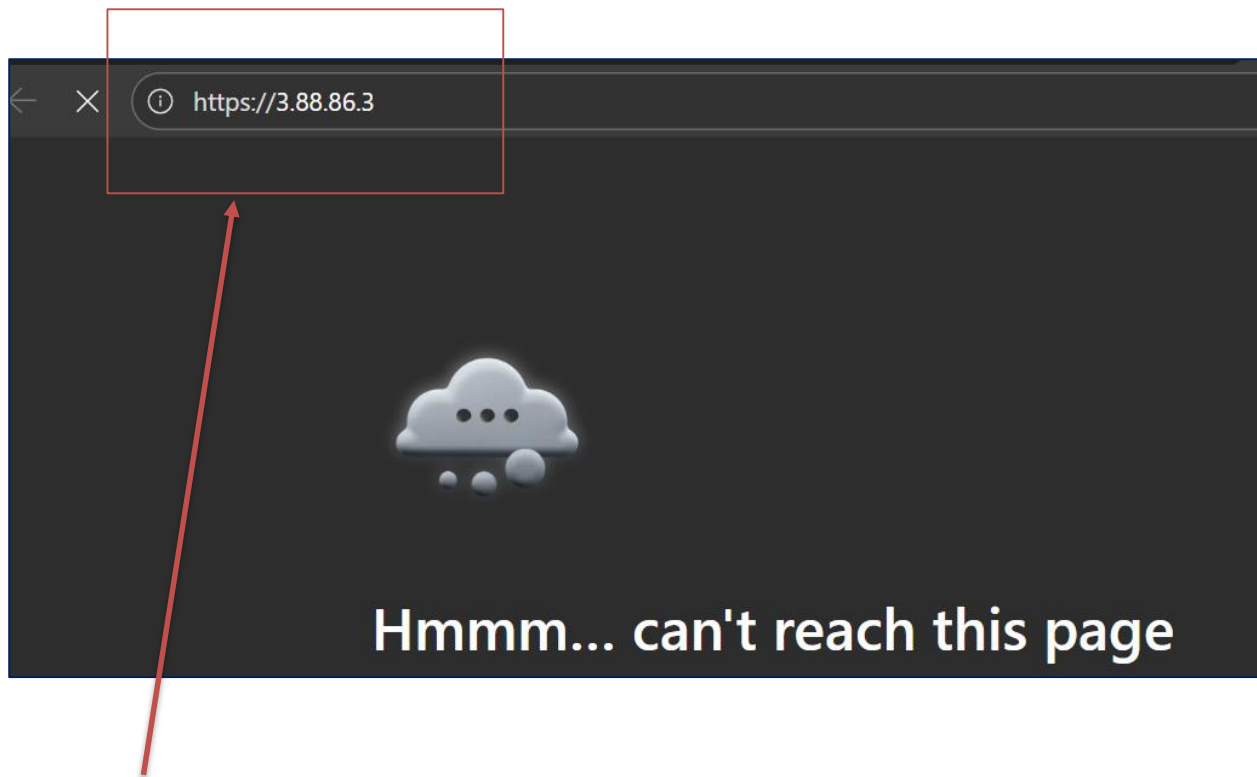
▼ Instance summary Info

Instance ID	Public IPv4 address	Private IPv4 addresses
i-039fe4a24a64b27dc	3.88.86.3   open address	172.31.43.80

IPv6 address: -

Instance state: Running

Public DNS: ec2-3-88-86-3.compute-1.amazonaws.com | open address



- **Change to HTTP** (<http://3.88.86.3>)

