

AdvDevops Case Study 1: Cloud Deployment with Automation

- **Concepts Used:** EC2, S3, CodeBuild, CodePipeline, CodeDeploy.
- **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

Case study overview:

This case study involves storing a web application in an S3 bucket, building it through AWS CodeBuild, automation using AWS CodePipeline to deploy the web app to an EC2 instance via CodeDeploy.

Key features and Applications:

Automated Deployment: AWS CodePipeline automates code deployment to S3 and EC2.

CI/CD Integration: Continuous integration and deployment with CodeBuild and CodeDeploy streamlines updates.

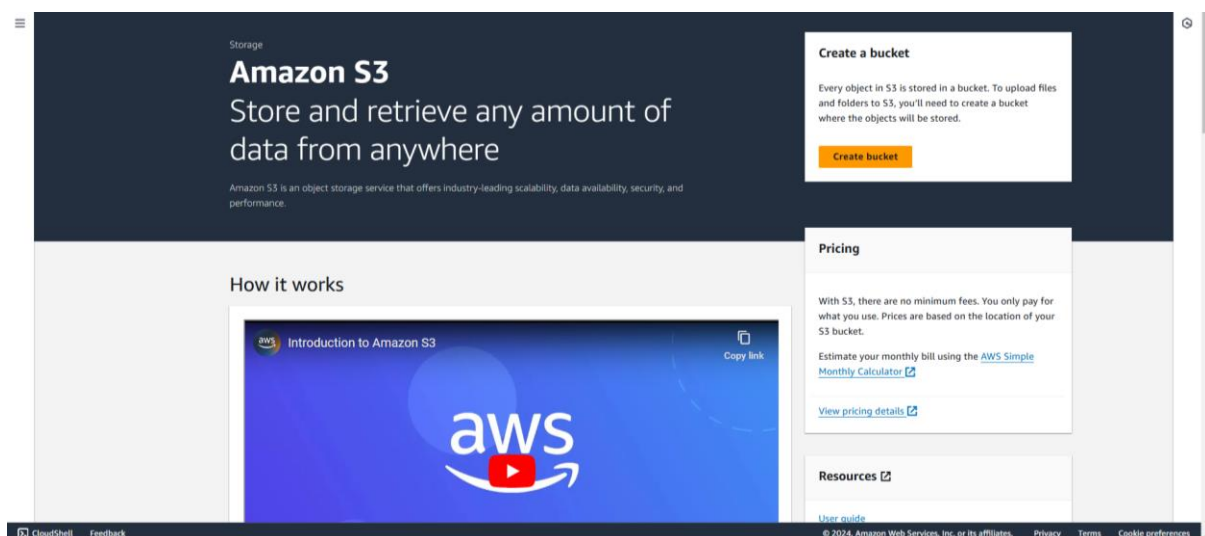
Scalability: EC2 provides scalable compute power and S3 offers scalable storage.

Version Control: CodeDeploy enables easy rollback to previous versions in case of failure

Web App Hosting: Ideal for hosting dynamic and static web apps.

Step 1: Create an S3 Bucket

In the AWS Management Console, navigate to **S3** by searching for it in the search bar at the top.



[Click Create bucket.](#)

Create a bucket

Every object in S3 is stored in a bucket. To upload files and folders to S3, you'll need to create a bucket where the objects will be stored.

Create bucket

Give your bucket a unique name (e.g., my-app-bucket). If you want the content of your web app (like the index.html file) to be publicly accessible, uncheck Block all public access.

For testing/deployment, you can allow public access

[Amazon S3](#) > [Buckets](#) > Create bucket

Create bucket [Info](#)

Buckets are containers for data stored in S3.

General configuration

AWS Region
Europe (Stockholm) eu-north-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](#)

abhinav-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

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.

Object Ownership
Bucket owner enforced

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.

Tags - optional (0)

You can use bucket tags to track storage costs and organize buckets. [Learn more](#)

No tags associated with this bucket.

Add tag

Default encryption [Info](#)

Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type [Info](#)

☒ Server-side encryption with Amazon S3 managed keys (SSE-S3)

☐ Server-side encryption with AWS Key Management Service keys (SSE-KMS)

☐ Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)

Secure your objects with two separate layers of encryption. For details on pricing, see [DSSE-KMS pricing](#) on the [Storage](#) tab of the [Amazon S3 pricing page](#).

Bucket Key

Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. [Learn more](#)

☐ Disable

☒ Enable

► Advanced settings

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

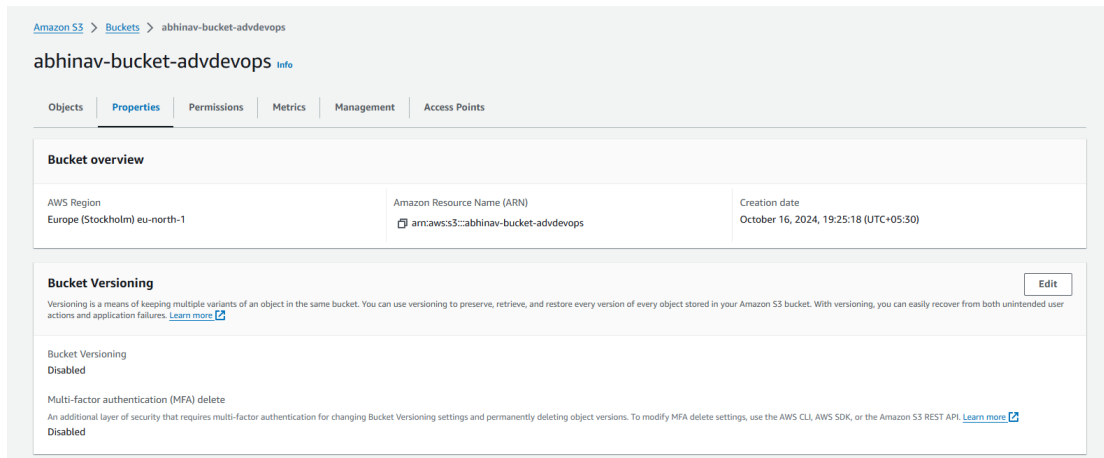
Cancel

Create bucket

To do this

Select Your Bucket: Click on your bucket named (abhinav-bucket-advdevops.)

Go to Properties: In the bucket settings, click on the "Properties" tab.

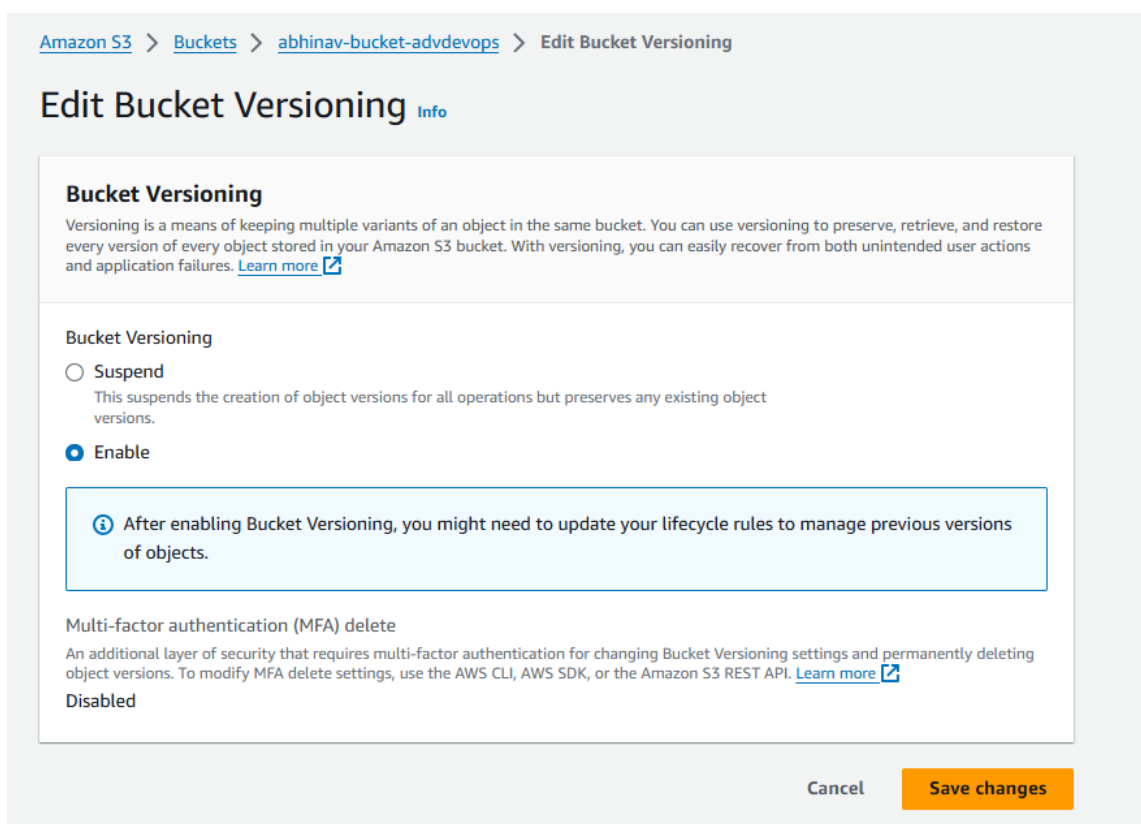


Scroll down to the "Bucket Versioning" section.

Click on "Edit".

Select "Enable" for versioning.

Click "Save changes".

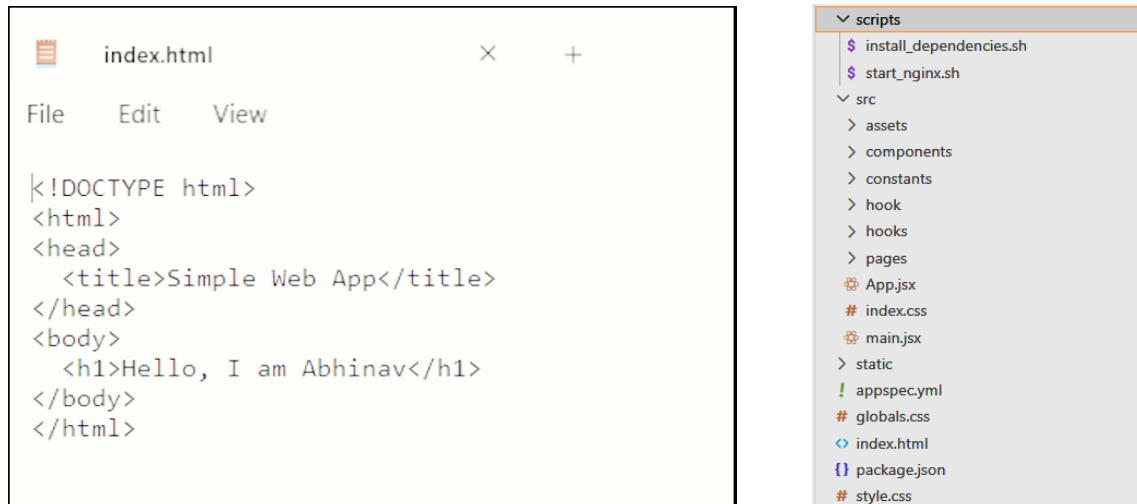


Step 2: Create a Simple Web Application (index.html)

On your local machine, create a new file called index.html.

Add the following simple HTML code into the file: index.html

This is necessary for a basic web app. I implemented my Project Integration (shop.co)



Create the following

appspec.yml (configuration file listing instructions to execute when the application is deployed)

```
version: 0.0
os: linux
files:
  - source: /
    destination: /usr/share/nginx/html
    overwrite: true
hooks:
  BeforeInstall:
    - location: scripts/install_dependencies.sh
      timeout: 300
      runas: ec2-user
  AfterInstall:
    - location: scripts/start_nginx.sh
      timeout: 300
      runas: ec2-user
```

scripts\install_dependencies.sh (update your system & install nginx)

```
#!/bin/bash
sudo yum update -y
sudo yum install -y nginx
```

scripts\start_nginx.sh (restart nginx service)

```
#!/bin/bash
sudo service nginx restart
```

Zip all into webapp.zip (project files for deployment).

scripts	21-10-2024 07:35 PM	File folder	
src	21-10-2024 08:40 PM	File folder	
static	21-10-2024 03:21 PM	File folder	
! appspec.yml	16-10-2024 11:46 PM	Yaml Source File	1 KB
# globals.css	21-10-2024 03:21 PM	CSS Source File	1 KB
index.html	21-10-2024 03:21 PM	Chrome HTML Do...	25 KB
package.json	21-10-2024 03:21 PM	JSON File	1 KB
# style.css	21-10-2024 03:21 PM	CSS Source File	28 KB

Step 3: Upload the Web App (webapp.zip) to S3

Go back to the S3 Console. Click on the bucket you created in Step 1

Successfully created bucket "abhinav-bucket-advdevops"
To upload files and folders, or to configure additional bucket settings, choose [View details](#).

Amazon S3 > Buckets

Account snapshot - updated every 24 hours
All AWS Regions
Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

View Storage Lens dashboard

General purpose buckets | Directory buckets

General purpose buckets (1) [Info](#) All AWS Regions
Buckets are containers for data stored in S3.

Find buckets by name

Name	AWS Region	IAM Access Analyzer	Creation date
abhinav-bucket-advdevops	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1	October 16, 2024, 19:25:18 (UTC+05:30)

In the bucket, click **Upload**.

Amazon S3 > Buckets > abhinav-bucket-advdevops

abhinav-bucket-advdevops [Info](#)

Objects | Properties | Permissions | Metrics | Management | Access Points

Objects (0) [Info](#)
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](#)

Find objects by prefix

Name	Type	Last modified	Size	Storage class
No objects You don't have any objects in this bucket.				

Upload

Click **Add files**, then select the webapp.zip file from your local machine. Leave all other options as default and click **Upload**.

Amazon S3 > Buckets > abhinav-bucket-advdevops > Upload

Upload Info

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#)

Drag and drop files and folders you want to upload here, or choose **Add files** or **Add folder**.

Files and folders (1 Total, 216.0 B)
All files and folders in this table will be uploaded.

< 1 >

<input type="checkbox"/>	Name	Folder
<input type="checkbox"/>	webapp.zip	-

Destination Info

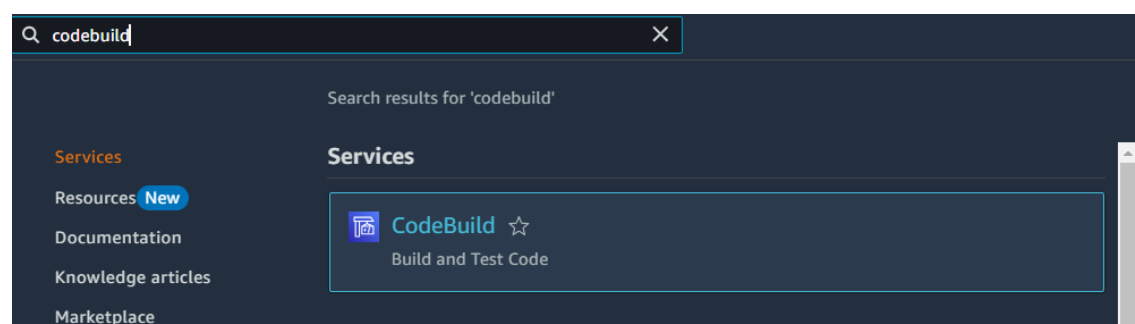
Destination
[s3://abhinav-bucket-advdevops](#)

► **Destination details**
Bucket settings that impact new objects stored in the specified destination.

Files and folders (1 Total, 216.0 B)						
<input type="text" value="Find by name"/>						
Name	Folder	Type	Size	Status	Error	
webapp.zip	-	application/x...	216.0 B	✓ Succeeded	-	

Step 4: Set Up AWS CodeBuild for the Web App

In the AWS Management Console, navigate to **CodeBuild** by searching for it in the top search bar.



Click Create build project.

Project Configuration:

- Project Name: Enter a name

[Developer Tools](#) > [CodeBuild](#) > [Build projects](#) > Create build project

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

- Source Provider: Select Amazon S3.
- Bucket: Select the bucket (your bucket name) you created earlier.
- Object Key: Select the webapp.zip file you uploaded.

Source Add source

Source 1 - Primary

Source provider

Bucket

S3 object key or S3 folder

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

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

- Operating System: Select Ubuntu.

Environment

Environment image

☒ **Managed image**
Use an image managed by AWS CodeBuild

☐ **Custom image**
Specify a Docker image

Operating system

Ubuntu ▼

Runtime(s)

Standard ▼

Image

aws/codebuild/standard:7.0 ▼

Image version

Always use the latest image for this runtime version ▼

Service role

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

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

Role name

codebuild-abhinav-codebuild-advdevops-service-role

Type your service role name

► **Additional configuration**

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

Buildspec: Select "Insert build commands" and enter the following YAML script: (Switch to editor)

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 "Building project..."
6   artifacts:
7     files:
8       - index.html
9
```

- Uncheck uploading logs to CloudWatch as it unnecessary

Batch configuration

You can run a group of builds as a single execution. Batch configuration is also available in advanced option when starting build.

☐ Define batch configuration - *optional*
You can also define or override batch configuration when starting a build batch.

Artifacts

Add artifact

Artifact 1 - Primary

Type

No artifacts ▼

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

► Additional configuration
Cache, encryption key

Logs

CloudWatch

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

S3

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

Cancel

Create build project

Project created

You have successfully created the following project: abhinav-codebuild-advdevops

Create a notification rule for this project

Developer Tools > CodeBuild > Build projects > abhinav-codebuild-advdevops

abhinav-codebuild-advdevops

Actions ▼ Create trigger Edit Clone Debug build Start build with overrides Start build

Configuration

Source provider Amazon S3	Primary repository abhinav-bucket-advdevops/webapp.zip	Artifacts upload location -	Service role arn:aws:iam::010928207735:role/service-role/codebuild-abhinav-codebuild-advdevops-service-role
Public builds Disabled			

Build history

Batch history

Project details

Build triggers

Metrics

codepipeline

Search results for 'codepipeline'

Services

Resources **New**

Documentation

Knowledge articles

Marketplace

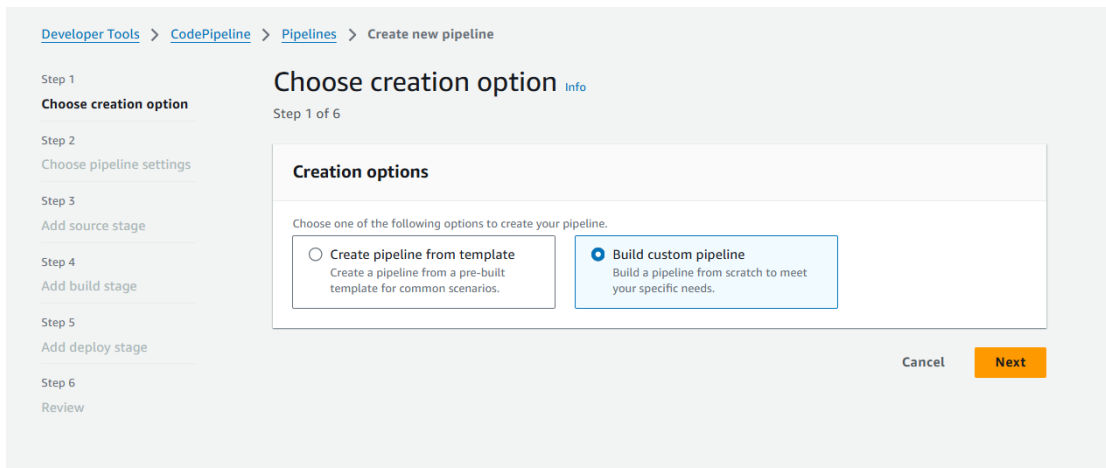
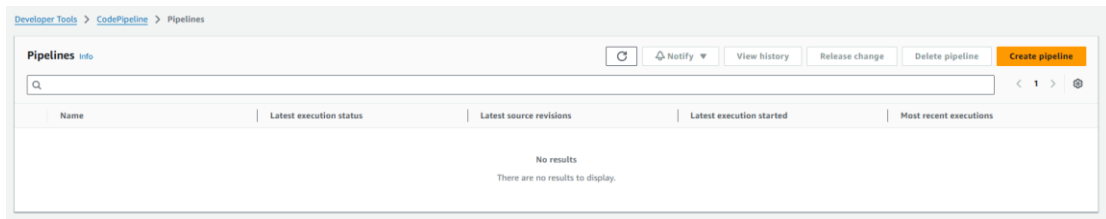
Services

CodePipeline

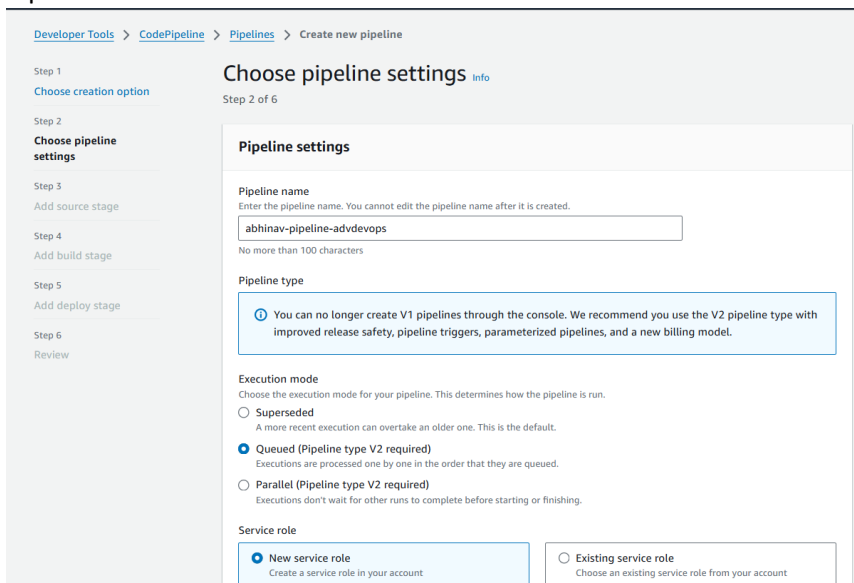
Release Software using Continuous Delivery

Click Create Pipeline.

10



- Pipeline Name: Enter a name



- Service Role: Select New service role to create a new role for this pipeline.

Role name

abhinav-pipeline-new-service-role

Type your service role name

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

Variables

You can add variables at the pipeline level. You can choose to assign the value when you start the pipeline. Choosing this option requires pipeline type V2. [Learn more](#)

No variables defined at the pipeline level in this pipeline.

[Add variable](#)

You can add up to 50 variables.

The first pipeline execution will fail if variables have no default values.

Advanced settings

Cancel Previous Next

Source Provider: Select Amazon S3.

Bucket: Select the same S3 bucket you used earlier.

S3 object key: Enter webapp.zip as the source file.

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1
Choose creation option

Step 2
Choose pipeline settings

Step 3
Add source stage

Step 4
Add build stage

Step 5
Add deploy stage

Step 6
Review

Add source stage [info](#)

Step 3 of 6

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
abhinav-bucket-advdevops

S3 object key
webapp.zip

Enter the object key. You can include a file path without the delimiter character (/) at the beginning. Include the file extension. Example: SampleApp.zip

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

Cancel Previous Next

Build Provider: Select Other Build Providers Then select AWS Codebuild

Project Name: Select the build project you created in Step 4

Add build stage Info

Step 4 of 6

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.

Q abhinav-codebuild-advdevops 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.

Region
Europe (Stockholm) ▼

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

SourceArtifact X
Defined by: Source

No more than 100 characters

☒ Enable automatic retry on stage failure

Cancel Previous Skip build stage **Next**

Deploy Provider: Select Amazon S3.

Bucket: Choose the same S3 bucket.

For Extract file before deploy, select Yes.

Add deploy stage [Info](#)

Step 5 of 6

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
Europe (Stockholm)

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

BuildArtifact X
Defined by: Build

No more than 100 characters

Bucket
abhinav-bucket-advdevops

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

Review your bucket and create the pipeline

Success
Congratulations! The pipeline abhinav-pipeline-advdevops has been created.

Create a notification rule for this pipeline

Developer Tools > CodePipeline > Pipelines > abhinav-pipeline-advdevops

abhinav-pipeline-advdevops

Notify Edit Stop execution Clone pipeline Release change

Pipeline type: V2 Execution mode: QUEUED

Source In progress
Pipeline execution ID: 561d8077-706a-4c06-844d-a0e0247a3a24

Source
Amazon S3
In progress - Just now
View details

Disable transition

Build Didn't Run
Start rollback

Build
AWS CodeBuild
Didn't Run
No executions yet

Disable transition

Deploy Didn't Run
Start rollback

Success
Stage Source successfully retried

Developer Tools > CodePipeline > Pipelines > abhinav-pipeline-advdevops

abhinav-pipeline-advdevops Notify Edit Stop execution Clone pipeline Release change

Pipeline type: V2 Execution mode: QUEUED

Source Succeeded Manual retry attempt View retry metadata
Pipeline execution ID: [5b1c8877-706a-4cd6-944d-a0a0247a3a24](#)

Source
[Amazon S3](#)
Succeeded - Just now
View details

Source: Amazon S3 version id: null

Disable transition

Build Succeeded Start rollback
Pipeline execution ID: [5b1c8877-706a-4cd6-944d-a0a0247a3a24](#)

Build
[AWS CodeBuild](#)
Succeeded - Just now
View details

Source: Amazon S3 version id: null

Disable transition

Deploy Succeeded Start rollback
Pipeline execution ID: [5b1c8877-706a-4cd6-944d-a0a0247a3a24](#)

Deploy
[Amazon S3](#)
Succeeded - Just now
View details

Source: Amazon S3 version id: null

Step 6: Create an EC2 Instance

Click on **Launch Instance**.

Select an instance type (e.g., t3.micro for free tier eligibility).

Instances Info Last updated less than a minute ago Connect Instance state Actions Launch instances

All states

Instance state: running Clear filters

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
No matching instances found									

EC2 > ... > Launch an instance

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

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.

Quick Start

[Amazon Linux](#) [macOS](#) [Ubuntu](#) [Windows](#) [Red Hat](#) [SUSE Linux](#) [Browse more AMIs](#)

Summary

Number of instances Info

Software Image (AMI)
Amazon Linux 2023 AMI 2023.6.2...read more
ami-02db68a01488594c5

Virtual server type (instance type)
t3.micro

Firewall (security group)
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 AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel Launch instance Preview code

Create a new key-pair of RSA type. (Will be used to SSH into our instance later)

Instance type

t3.micro

Free tier eligible

Family: t3 2 vCPU 1 GiB Memory Current generation: true
On-Demand RHEL base pricing: 0.0396 USD per Hour
On-Demand SUSE base pricing: 0.0108 USD per Hour
On-Demand Linux base pricing: 0.0108 USD per Hour
On-Demand Windows base pricing: 0.02 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*

abhinav-key ▼

Create new key pair

Allow HTTP/HTTPS traffic from anywhere

▼ Network settings Info

Edit

Network Info

vpc-0ee3aadd1d4ec704a

Subnet Info

No preference (Default subnet in any availability zone)

Auto-assign public IP Info

Enable

Additional charges apply when outside of free tier allowance

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-10' 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.

×

Instances (1/1) Info

Find Instance by attribute or tag (case-sensitive) All states

Instance state: running Clear filters

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
abhinav-ec2-advdevops	i-0007f32909cb1c955	Running	t3.micro	3/3 checks passed	View alarms +	eu-north-1b	ec2-13-60-85-90.eu-no...	13.60.85.90	--

i-0007f32909cb1c955 (abhinav-ec2-advdevops)

Details | Status and alarms | Monitoring | Security | Networking | Storage | Tags

Instance summary Info

Instance ID i-0007f32909cb1c955 (abhinav-ec2-advdevops)	Public IPv4 address 13.60.85.90 open address	Private IPv4 addresses 172.31.39.8
IPv6 address --	Instance state Running	Public IPv4 DNS ec2-13-60-85-90.eu-north-1.compute.amazonaws.com open address
Hostname type IP name: ip-172-31-39-8.eu-north-1.compute.internal	Private IP DNS name (IPv4 only) ip-172-31-39-8.eu-north-1.compute.internal	Elastic IP addresses --
Answer private resource DNS name IPv4 (A)	Instance type t3.micro	Amazon CloudWatch Logs
Auto-assign IAM ID address	UDF ID	

Edit inbound rules in security groups

Inbound rules (8) Manage tags Edit inbound rules

Search

Name	Security group rule...	IP version	Type	Protocol	Port range	Source
--	sgr-036a0549b96ddc4...	IPv4	HTTPS	TCP	443	0.0.0.0/0
--	sgr-09631799bcf329d25	IPv4	Custom TCP	TCP	8080	0.0.0.0/0
--	sgr-0fef37f116da859a6	IPv4	HTTP	TCP	80	0.0.0.0/0
--	sgr-087684ae158630a...	IPv4	All ICMP - IPv4	ICMP	All	0.0.0.0/0
--	sgr-0218aae10453425ca	IPv4	All traffic	All	All	0.0.0.0/0
--	sgr-0fb1658d98fdc170b	IPv4	All ICMP - IPv6	IPv6 ICMP	All	0.0.0.0/0
--	sgr-002172b6435afe427	IPv4	SSH	TCP	22	0.0.0.0/0
--	sgr-059edd8933bd94...	IPv4	All TCP	TCP	0 - 65535	0.0.0.0/0

Take note of tag of instance

i-0007f32909cb1c955 (abhinav-ec2-advdevops)

Details | Status and alarms | Monitoring | Security | Networking | Storage | **Tags**

Tags

Search

Key	Value
Name	abhinav-ec2-advdevops

Step 7: Create an IAM Role for EC2 with CodeDeploy Permissions

In the AWS Management Console, go to **Services** and select **IAM**.

Search results for 'iam'

Services [Show more](#)

IAM Manage access to AWS resources

Click on **Roles** in the left sidebar and then click **Create role**.

The screenshot shows the AWS IAM console interface. On the left is a navigation menu with options like Dashboard, Access management, Users, Roles, Policies, and Identity providers. The main content area is divided into two sections. The top section, titled 'Roles (8)', shows a list of roles with columns for Role name, Trusted entities, and Last activity. The bottom section, titled 'IAM Dashboard', provides an overview of the account's security and resources. It includes a 'Security recommendations' card with an 'Add MFA' button, an 'IAM resources' card showing counts for User groups (0), Users (0), Roles (8), Policies (8), and Identity providers (0), and a 'What's new' section with updates from the AWS IAM Access Analyzer.

Choose **AWS service** and then select **EC2**.

This screenshot shows the 'Select trusted entity' step in the AWS IAM console. It presents several options for the trusted entity type: AWS service, AWS account, Web identity, SAML 2.0 federation, and Custom trust policy. The 'AWS service' option is selected. Below this, the 'Use case' section is shown, with 'EC2' selected from a dropdown menu. A list of specific use cases for EC2 is provided, including 'EC2 - Spot Fleet Auto Scaling', 'EC2 - Spot Fleet Tagging', 'EC2 - Spot Instances', 'EC2 - Spot Fleet', and 'EC2 - Scheduled Instances'. The 'EC2 - Spot Fleet Auto Scaling' option is selected.

Search for and attach the **AWSCodeDeployRole**, **AmazonEC2RoleforCodeDeploy** policy to grant the necessary permissions.

This screenshot shows the 'Add permissions' step in the AWS IAM console. It displays a list of permissions policies that match the search term 'awscodedeployrole'. The policies are filtered by type, and the 'AWSCodeDeployRole' policy is selected. The list includes policies like 'AWSCodeDeployRoleForCloudFormation', 'AWSCodeDeployRoleForECS', 'AWSCodeDeployRoleForECSLimited', 'AWSCodeDeployRoleForLambda', and 'AWSCodeDeployRoleForLambdaLimited'. The 'AWSCodeDeployRole' policy is highlighted, and the 'Set permissions boundary - optional' section is visible at the bottom.

[IAM](#) > [Roles](#) > [abhinav-created-codedeploy-role](#) > Add permissions

Attach policy to abhinav-created-codedeploy-role

► Current permissions policies (1)

Other permissions policies (961)

Filter by Type

amazonec2role All types 5 matches

Policy name	Type	Description
AmazonEC2RoleforAWSCodeDeploy	AWS managed	Provides EC2 access to S3 bucket to dow...

Name your role and click **Create role**.

Name, review, and create

Role details

Role name

Enter a meaningful name to identify this role.

abhinav-created-codedploy-role

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

Description

Add a short explanation for this role.

Allows EC2 instances to call AWS services on your behalf.

Maximum 1000 characters. Use letters [A-Z and a-z], numbers [0-9], tabs, new lines, or any of the following characters: _ + ^ ! @ # \$ % & * ~ - = . , : ; ' " { } [] \ | ` ~ . , : ; ' " { } [] \ | `

Role abhinav-created-codedeploy-role created.

View role

✕

IAM > Roles

Roles (9)
Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Delete
Create role

<
1
>
⚙️

<input type="checkbox"/>	Role name	Trusted entities	Last activity
<input type="checkbox"/>	abhinav-created-codedeploy-role	AWS Service: ec2	-

Goto trusted relationships and edit trust policy with code

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "codedeploy.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

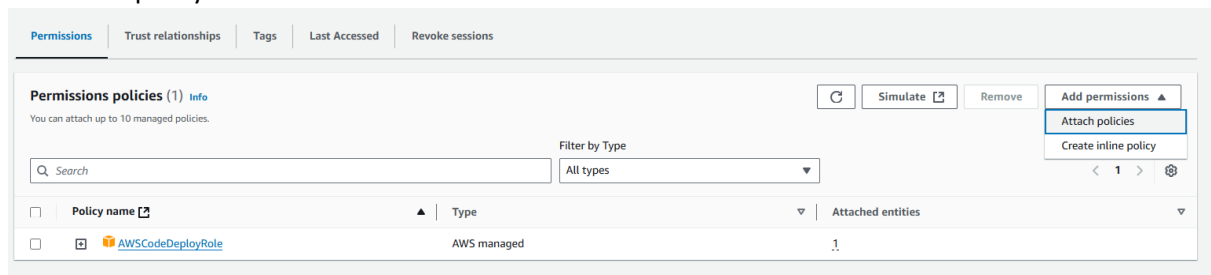
Edit trust policy

```
1 {  
2     "Version": "2012-10-17",  
3     "Statement": [  
4         {  
5             "Effect": "Allow",  
6             "Principal": {  
7                 "Service": "codedeploy.amazonaws.com"  
8             },  
9             "Action": "sts:AssumeRole"  
10        }  
11    ]  
12 }
```

Copy the arn as it will be needed later

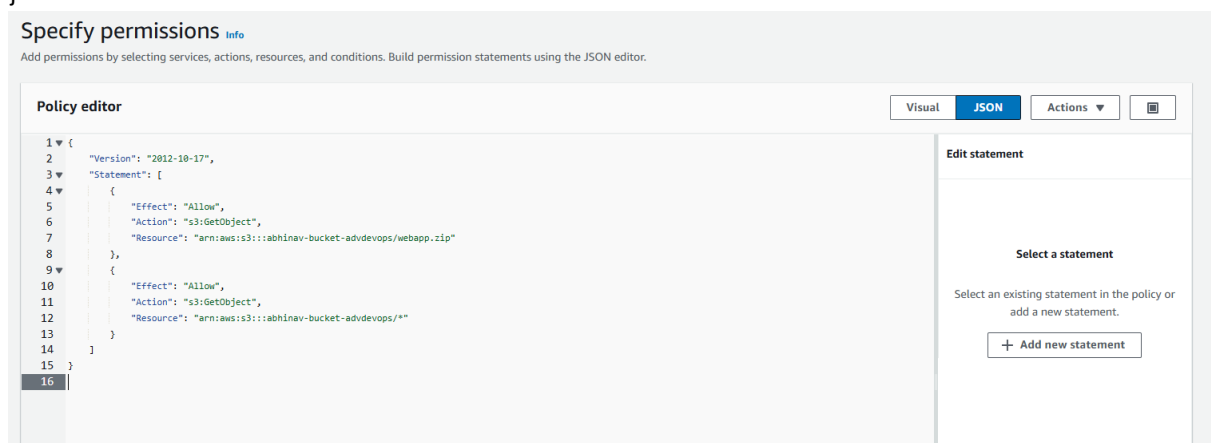


Add inline policy



In Json editor

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::abhinav-bucket-advdevops/webapp.zip"
    },
    {
      "Effect": "Allow",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::abhinav-bucket-advdevops/*"
    }
  ]
}
```



Review and create

Review and create [info](#)

Review the permissions, specify details, and tags.

Policy details

Policy name
Enter a meaningful name to identify this policy.

Maximum 128 characters. Use alphanumeric and '+', '=', '@', '-', '_' characters.

Permissions defined in this policy [info](#) Edit

Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, group, or role), attach a policy to it

Allow (1 of 423 services) Show remaining 422 services

Service	Access level	Resource	Request condition
S3	Limited: Read	Multiple	None

Cancel Previous Create policy

Step 8: Install codedeploy-agent and nginx on ec2

Refer to the documentation to install

<https://docs.aws.amazon.com/codedeploy/latest/userguide/codedeploy-agent-operations-install-linux.html>

sudo yum install -y ruby

```

[ec2-user@ip-172-31-39-8 ~]$ sudo yum install -y ruby
created by dnf config-manager from https://aws-codedeploy-us-east-1.s3.us-east-1.amazo 419 B/s | 305 B 00:00
Errors during downloading metadata for repository 'aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com_latest_install':
 - Status code: 404 for https://aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com/latest/install/repodata/repomd.xml
(IP: 52.217.87.0)
Error: Failed to download metadata for repo 'aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com_latest_install': Cannot
download repomd.xml: Cannot download repodata/repomd.xml: All mirrors were tried
Ignoring repositories: aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com_latest_install
Last metadata expiration check: 1:01:45 ago on Wed Oct 16 15:02:49 2024.
Dependencies resolved.
=====
Package                                Architecture  Version                                Repository    Size
=====
Installing:
ruby3.2                                x86_64        3.2.2-180.amzn2023.0.3                amazonlinux    43 k
Installing dependencies:
ruby3.2-default-gems                   noarch        3.2.2-180.amzn2023.0.3                amazonlinux    34 k
ruby3.2-libs                           x86_64        3.2.2-180.amzn2023.0.3                amazonlinux    4.0 M
ruby3.2-rubygem-io-console             x86_64        0.6.0-180.amzn2023.0.3                amazonlinux    25 k
ruby3.2-rubygem-json                   x86_64        2.6.3-180.amzn2023.0.3                amazonlinux    53 k
ruby3.2-rubygem-psych                  x86_64        5.0.1-180.amzn2023.0.3                amazonlinux    52 k
Installing weak dependencies:
ruby3.2-rubygem-bigdecimal             x86_64        3.1.3-180.amzn2023.0.3                amazonlinux    69 k
ruby3.2-rubygem-bundler                noarch        2.4.10-180.amzn2023.0.3                amazonlinux    384 k
ruby3.2-rubygem-rdoc                   noarch        6.5.0-180.amzn2023.0.3                amazonlinux    461 k
ruby3.2-rubygems                       noarch        3.4.10-180.amzn2023.0.3                amazonlinux    257 k
=====
Transaction Summary
=====

```

cd /tmp

(wget https://**bucket-name**.s3.**region-identifier**.amazonaws.com/latest/install)

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

```

ec2-user@ip-172-31-39-8:~$ cd /tmp
[ec2-user@ip-172-31-39-8 tmp]$ cd /tmp
wget https://aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com/latest/install
--2024-10-16 16:05:31-- https://aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com/latest/install
Resolving aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com (aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com)...
16.182.64.50, 52.217.85.240, 52.217.233.146, ...
Connecting to aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com (aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com)
[16.182.64.50]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 19045 (19K) []
Saving to: 'install'

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

2024-10-16 16:05:31 (39.9 MB/s) - 'install' saved [19045/19045]

[ec2-user@ip-172-31-39-8 tmp]$ chmod +x ./install
[ec2-user@ip-172-31-39-8 tmp]$ sudo ./install auto
I, [2024-10-16T16:05:52.481558 #27254] INFO -- : Starting Ruby version check.
W, [2024-10-16T16:05:52.481635 #27254] WARN -- : The Ruby version in /usr/bin/ruby3.2 is 3.2.2, . Attempting to instal
l anyway.
I, [2024-10-16T16:05:52.481668 #27254] INFO -- : Starting update check.
I, [2024-10-16T16:05:52.481687 #27254] INFO -- : Attempting to automatically detect supported package manager type for
system...
I, [2024-10-16T16:05:52.497174 #27254] INFO -- : Checking AWS_REGION environment variable for region information...
I, [2024-10-16T16:05:52.497220 #27254] INFO -- : Checking EC2 metadata service for region information...
I, [2024-10-16T16:05:52.506852 #27254] INFO -- : Checking AWS_DOMAIN environment variable for domain information...
I, [2024-10-16T16:05:52.506892 #27254] INFO -- : Checking EC2 metadata service for domain information...
I, [2024-10-16T16:05:52.509392 #27254] INFO -- : Downloading version file from bucket aws-codedeploy-eu-north-1 and ke
y latest/LATEST_VERSION...

```

chmod +x ./install

sudo ./install auto

```

ec2-user@ip-172-31-39-8:~$ cd /tmp
[ec2-user@ip-172-31-39-8 tmp]$ cd /tmp
[ec2-user@ip-172-31-39-8 tmp]$ chmod +x ./install
[ec2-user@ip-172-31-39-8 tmp]$ sudo ./install auto
I, [2024-10-16T16:05:52.481558 #27254] INFO -- : Starting Ruby version check.
W, [2024-10-16T16:05:52.481635 #27254] WARN -- : The Ruby version in /usr/bin/ruby3.2 is 3.2.2, . Attempting to instal
l anyway.
I, [2024-10-16T16:05:52.481668 #27254] INFO -- : Starting update check.
I, [2024-10-16T16:05:52.481687 #27254] INFO -- : Attempting to automatically detect supported package manager type for
system...
I, [2024-10-16T16:05:52.497174 #27254] INFO -- : Checking AWS_REGION environment variable for region information...
I, [2024-10-16T16:05:52.497220 #27254] INFO -- : Checking EC2 metadata service for region information...
I, [2024-10-16T16:05:52.506852 #27254] INFO -- : Checking AWS_DOMAIN environment variable for domain information...
I, [2024-10-16T16:05:52.506892 #27254] INFO -- : Checking EC2 metadata service for domain information...
I, [2024-10-16T16:05:52.509392 #27254] INFO -- : Downloading version file from bucket aws-codedeploy-eu-north-1 and ke
y latest/LATEST_VERSION...
I, [2024-10-16T16:05:52.510750 #27254] INFO -- : Endpoint: https://aws-codedeploy-eu-north-1.s3.eu-north-1.amazonaws.c
om/latest/LATEST_VERSION
I, [2024-10-16T16:05:52.600903 #27254] INFO -- : Downloading package from bucket aws-codedeploy-eu-north-1 and key rel
eases/codedeploy-agent-1.7.0-92.noarch.rpm...
I, [2024-10-16T16:05:52.601127 #27254] INFO -- : Endpoint: https://aws-codedeploy-eu-north-1.s3.eu-north-1.amazonaws.c
om/releases/codedeploy-agent-1.7.0-92.noarch.rpm
I, [2024-10-16T16:05:52.713860 #27254] INFO -- : Executing '/usr/bin/yum -y localinstall /tmp/codedeploy-agent-1.7.0-9
2.noarch.rpm-20241016-27254-ot4m6z.rpm'...

```

sudo service codedeploy-agent start

sudo service codedeploy-agent status

```

[ec2-user@ip-172-31-39-8 tmp]$ sudo service codedeploy-agent start
Starting codedeploy-agent:[ec2-user@ip-172-31-39-8 tmp]$ sudo service codedeploy-agent status
The AWS CodeDeploy agent is running as PID 27399
[ec2-user@ip-172-31-39-8 tmp]$ |

```

sudo yum install -y nginx

```

[ec2-user@ip-172-31-39-8 ~]$ sudo yum install -y nginx
created by dnf config-manager from https://aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com/latest/install
Errors during downloading metadata for repository 'aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com_latest_install':
  Status code: 404 for https://aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com/latest/install/repo/data/repo.xml (IP: 54.231.137.66)
Error: Failed to download metadata for repo 'aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com_latest_install': Cannot download repomd.xml: Cannot download repodata/repomd.xml: All mirrors were tried
Ignoring repositories: aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com_latest_install
Last metadata expiration check: 2:21:52 ago on Wed Oct 16 15:02:49 2024.
Dependencies resolved.

```

Package	Architecture	Version	Repository	Size
Installing:				
nginx	x86_64	1:1.24.0-1.amzn2023.0.4	amazonlinux	33 k
Installing dependencies:				
generic-logos-httpd	noarch	18.0.0-12.amzn2023.0.3	amazonlinux	19 k
openssl-libs	x86_64	2.9.1-1.amzn2023.0.3	amazonlinux	308 k
libuv	x86_64	1.4.0-5.amzn2023.0.2	amazonlinux	66 k
nginx-core	x86_64	1:1.24.0-1.amzn2023.0.4	amazonlinux	596 k
nginx-filesystem	noarch	1:1.24.0-1.amzn2023.0.4	amazonlinux	9.8 k
nginx-mime-types	noarch	2.1.49-3.amzn2023.0.3	amazonlinux	21 k

Transaction Summary

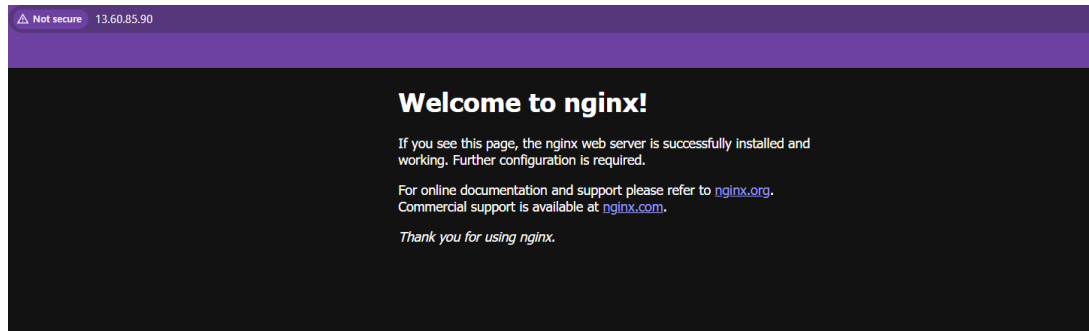
sudo service nginx start

sudo service nginx status

```
[ec2-user@ip-172-31-39-8 ~]$ sudo service nginx status
Redirecting to /bin/systemctl status nginx.service
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; preset: disabled)
   Active: active (running) since Wed 2024-10-16 17:25:47 UTC; 16s ago
     Process: 33739 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
     Process: 33740 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
     Process: 33741 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
   Main PID: 33742 (nginx)
    Tasks: 3 (limit: 1059)
   Memory: 2.9M
      CPU: 54ms
   CGroup: /system.slice/nginx.service
           └─33742 "nginx: master process /usr/sbin/nginx"
             └─33743 "nginx: worker process"
               └─33744 "nginx: worker process"

Oct 16 17:25:47 ip-172-31-39-8.eu-north-1.compute.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server...
Oct 16 17:25:47 ip-172-31-39-8.eu-north-1.compute.internal nginx[33740]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Oct 16 17:25:47 ip-172-31-39-8.eu-north-1.compute.internal nginx[33740]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Oct 16 17:25:47 ip-172-31-39-8.eu-north-1.compute.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.
[ec2-user@ip-172-31-39-8 ~]$
```

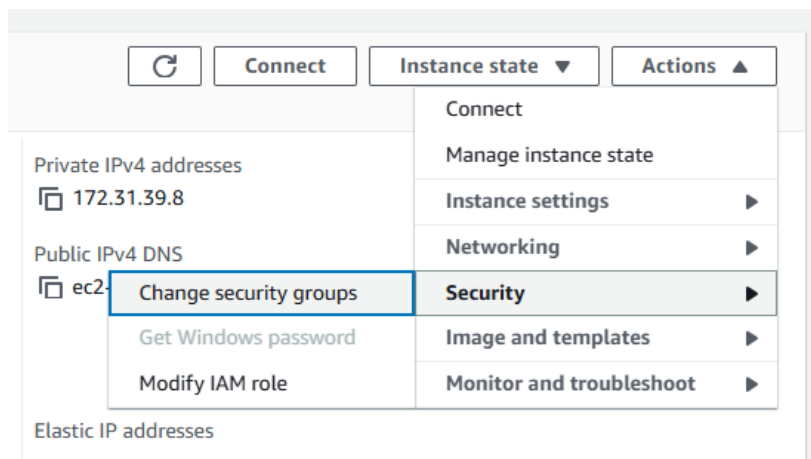
Check if nginx installed by opening the instance's public ipv4 ip



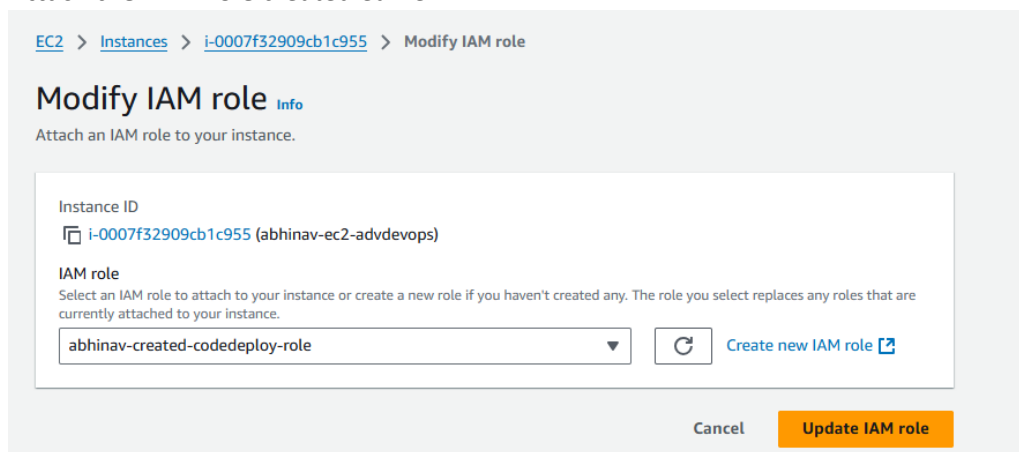
Step 9: Attach the IAM Role to the EC2 Instance

Go back to the EC2 Dashboard.

Select your Instance and click on Actions > Security > Modify IAM Role.

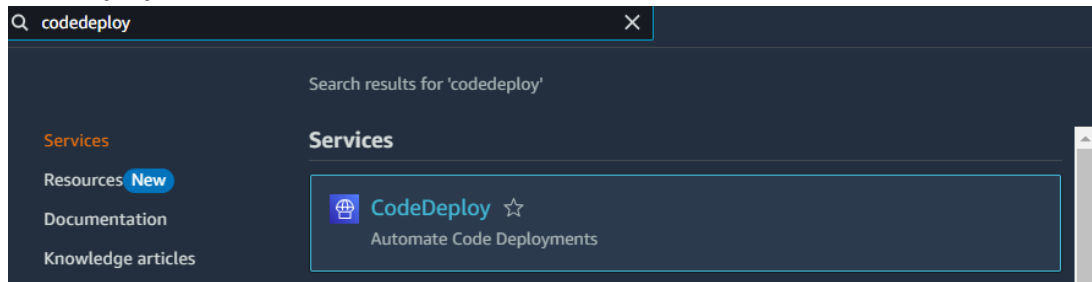


Attach the IAM Role created earlier.

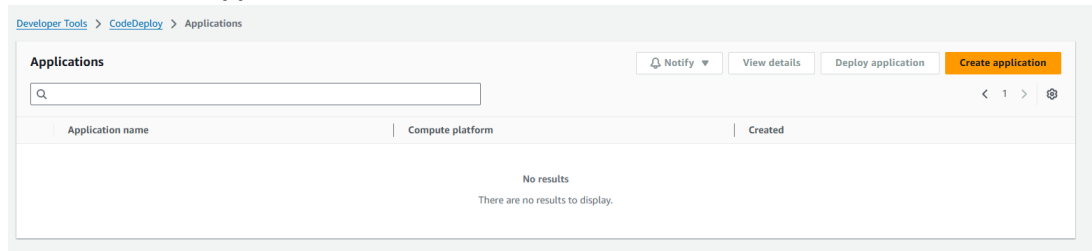


Step 10: Set Up AWS CodeDeploy

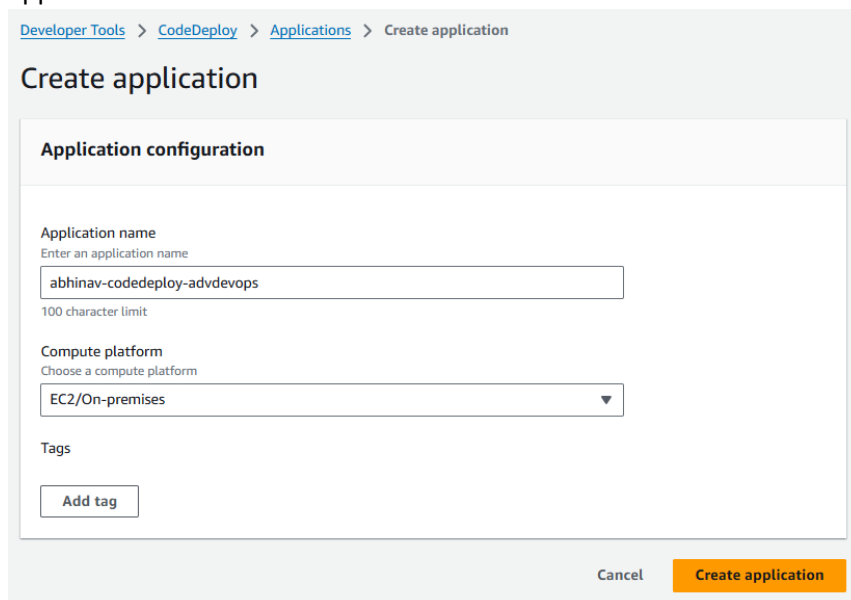
Open the CodeDeploy Console: In the AWS Management Console, navigate to the CodeDeploy service.



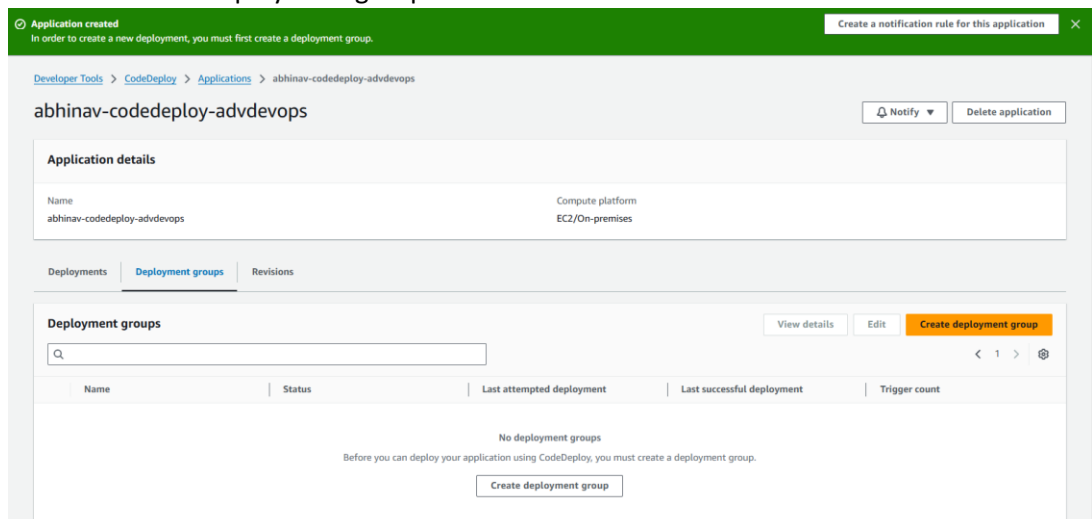
Click on "Create application".



Enter an application name , For Compute platform, select EC2/On-premises. Click "Create application"



After creating the application, you will be prompted to create a deployment group. Click on "Create deployment group".



[Developer Tools](#) > [CodeDeploy](#) > [Applications](#) > [abhinav-codedeploy-advdevops](#) > Create deployment group

Create deployment group

Application

Application
abhinav-codedeploy-advdevops

Compute type
EC2/On-premises

Deployment group name

Enter a deployment group name

abhinav-deployment-group

100 character limit

Enter the **ARN** of the service role you created for CodeDeploy (the one with permissions for CodeDeploy).

arn:aws:iam::010928207735:role/abhinav-created-codedeploy-role

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::010928207735:role/abhinav-created-codedeploy-role

X

Use entered value: arn:aws:iam::010928207735:role/abhinav-created-codedeploy-role

For **Environment configuration**, choose **EC2 instances**. Enter your key value pair which was noted from under tags

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	
<div>Q Name X</div>	<div>Q abhinav-ec2-advdevops X</div>	Remove tag
<div>Add tag</div>		
<div>+ Add tag group</div>		

☐ On-premises instances

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

Disable load balancing.

Agent configuration with AWS Systems Manager [Info](#)

**We recommend configuring your CodeDeploy Agent install and updates with AWS Systems Manager.** AWS Systems Manager provides more control over CodeDeploy Agent version updates and rollbacks than installing using other methods. [Learn more](#)

Install AWS CodeDeploy Agent

☒ Never

☐ Only once

☐ Now and schedule updates

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

Create deployment group

Create deployment group

Success
Deployment group created

[Developer Tools](#) > [CodeDeploy](#) > [Applications](#) > [abhinav-codedeploy-advdevops](#) > [abhinav-deployment-group](#)

abhinav-deployment-group

Edit Delete Create deployment

Deployment group details

Deployment group name	Application name	Compute platform
abhinav-deployment-group	abhinav-codedeploy-advdevops	EC2/On-premises
Deployment type	Service role ARN	Deployment configuration
In-place	arn:aws:iam::010928207735:role/abhinav-created-codedeploy-role	CodeDeployDefault.AllAtOnce
Rollback enabled	Agent update scheduler	
False	Learn to schedule update in AWS Systems Manager	

Step 11: Give access of S3 bucket to codedeploy

Navigate to the **S3 service** in the AWS Management Console.

Find and select your bucket (e.g., abhinav-bucket-advdevops).

Go to the **Permissions** tab and check the **Bucket Policy**.

The screenshot shows the AWS S3 console interface for the bucket 'abhinav-bucket-advdevops'. The 'Permissions' tab is selected, displaying the 'Permissions overview' section. Below this, there is a 'Block public access (bucket settings)' section with a toggle switch set to 'OFF'. At the bottom, the 'Bucket policy' section is visible, showing the JSON policy for the bucket.

Go to the Permissions tab and check the Bucket Policy.
Ensure that the policy allows CodeDeploy to access the bucket

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "codedeploy.amazonaws.com"
      },
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::{your_bucket_name}/*"
    }
  ]
}
```

The screenshot shows the 'Edit bucket policy' page in the AWS S3 console. The 'Bucket policy' section is expanded, displaying the JSON policy for the bucket. The policy is as follows:

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Principal": {
7         "Service": "codedeploy.amazonaws.com"
8       },
9       "Action": "s3:GetObject",
10      "Resource": "arn:aws:s3:::abhinav-bucket-advdevops/*"
11    }
12  ]
13 }
```

Copy the S3 bucket's URI

Objects (2) info

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](#)

Find objects by prefix: Show versions: ☐

Name	Type	Last modified	Size	Storage class
<input type="checkbox"/> index.html	html	October 16, 2024, 23:48:02 (UTC+05:30)	133.0 B	Standard
<input checked="" type="checkbox"/> webapp.zip	zip	October 21, 2024, 20:55:43 (UTC+05:30)	8.8 MB	Standard

Amazon S3 > Buckets > abhinav-bucket-advdevops

abhinav-bucket-advdevops info

Objects (2) info

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](#)

Find objects by prefix: Show versions: ☐

Name	Type	Last modified	Size	Storage class
<input type="checkbox"/> index.html	html	October 16, 2024, 20:14:20 (UTC+05:30)	133.0 B	Standard
<input checked="" type="checkbox"/> webapp.zip	zip	October 16, 2024, 19:34:35 (UTC+05:30)	216.0 B	Standard

Step 12: Automate CodeDeploy using CodePipeline

Under your CodePipeline > Deploy > Edit > Add an Action > Set Action Provider as “**AWS CodeDeploy**” > configure our created application and deployment groups > **Save**

Edit action

Action name:

Action provider:

Region:

Input artifacts:

Application name:

Deployment group:

Variable namespace - optional:

Cancel Done

Step 13: Create a Deployment for Your Application

Click on the Deployments tab.

Click on Create deployment.

Developer Tools > CodeDeploy > Applications > abhinav-codedeploy-advdevops

abhinav-codedeploy-advdevops Notify Delete application

Application details

Name: abhinav-codedeploy-advdevops Compute platform: EC2/On-premises

Deployments Deployment groups Revisions

Application deployment history View details Actions Copy deployment Retry deployment Create deployment

Deployment Id	Status	Deployment type	Deployment group	Revision location	Initiating event	Start time	End time
No results There are no results to display.							

Select your deployment group and paste the URI you copied

[Developer Tools](#) > [CodeDeploy](#) > [Applications](#) > [abhinav-codedeploy-advdevops](#) > [Create deployment](#)

Create deployment

Deployment settings

Application
abhinav-codedeploy-advdevops

Deployment group

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

[Developer Tools](#) > [CodeDeploy](#) > [Deployments](#) > [d-EUG05DOW6](#)

d-EUG05DOW6

[Copy deployment](#) [Retry deployment](#)

Deployment status

Installing application on your instances

100%

1 of 1 instances updated ✓ Succeeded

Deployment details

Application abhinav-codedeploy-advdevops	Deployment ID d-EUG05DOW6	Status ✓ Succeeded
Deployment configuration CodeDeployDefault.AllAtOnce	Deployment group abhinav-deployment-group	Initiated by User action
Deployment description -		

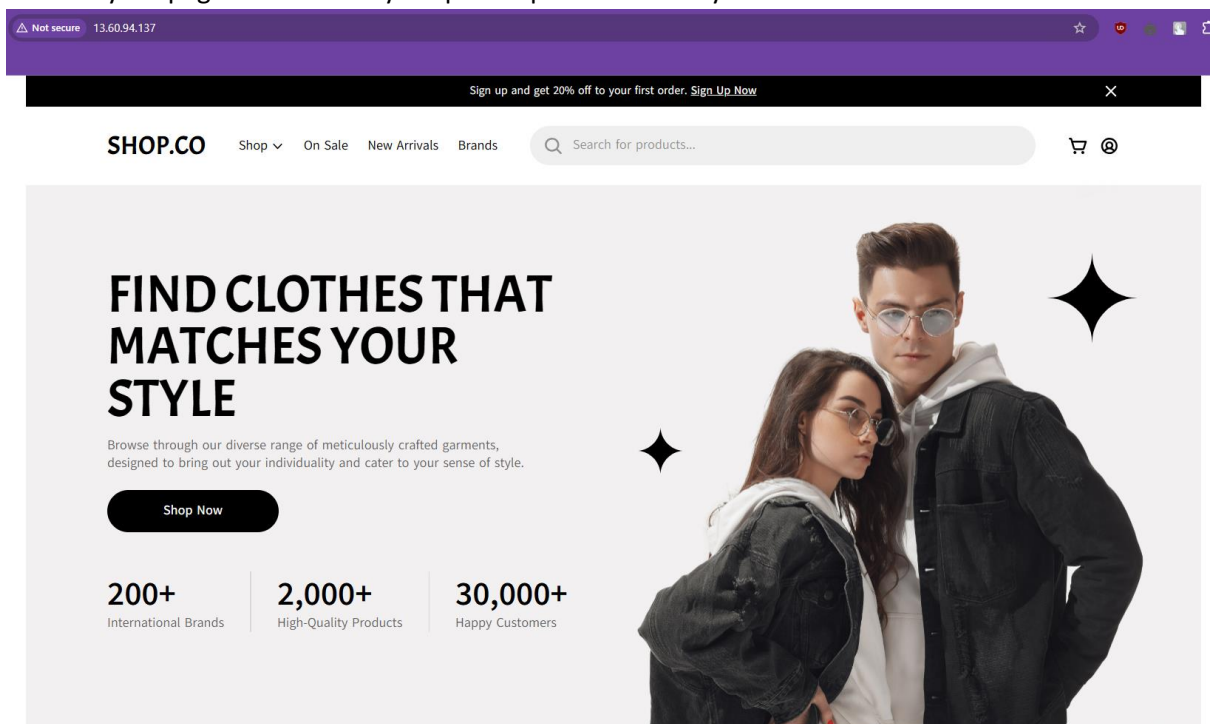
arn:aws:ec2:eu-north-1:010928207735:instance/i-0007f32909cb1c955

Deployment details		
Application abhinav-codeddeploy-advdevops	Deployment ID d-EUG6SDOW6	Status Succeeded
Deployment configuration CodeDeployDefault:ALIA1Once	Deployment group abhinav-deployment-group	Initiated by User action
Deployment description -		

Revision details		
Revision location s3://abhinav-bucket-advdevops/webapp.zip	Revision created 1 hour ago	Revision description Application revision registered by Deployment ID: d-STBX4MHW6

Event	Duration	Status	Error code	Start time	End time
ApplicationStop	less than one second	Succeeded	-	Oct 16, 2024 11:48 PM (UTC+5:30)	Oct 16, 2024 11:48 PM (UTC+5:30)
DownloadBundle	less than one second	Succeeded	-	Oct 16, 2024 11:48 PM (UTC+5:30)	Oct 16, 2024 11:48 PM (UTC+5:30)
BeforeInstall	2 seconds	Succeeded	-	Oct 16, 2024 11:48 PM (UTC+5:30)	Oct 16, 2024 11:48 PM (UTC+5:30)
Install	less than one second	Succeeded	-	Oct 16, 2024 11:48 PM (UTC+5:30)	Oct 16, 2024 11:48 PM (UTC+5:30)
AfterInstall	less than one second	Succeeded	-	Oct 16, 2024 11:48 PM (UTC+5:30)	Oct 16, 2024 11:48 PM (UTC+5:30)
ApplicationStart	less than one second	Succeeded	-	Oct 16, 2024 11:48 PM (UTC+5:30)	Oct 16, 2024 11:48 PM (UTC+5:30)
ValidateService	less than one second	Succeeded	-	Oct 16, 2024 11:48 PM (UTC+5:30)	Oct 16, 2024 11:48 PM (UTC+5:30)

Check if your page is hosted on your public ipv4 address of your EC2 bucket



Conclusion

In this project, we successfully automated the deployment of a web application. The application was hosted on an EC2 instance and managed through S3 for storage. Codebuild created the Build Artifact of the project. Code Pipeline stored the source & build artifacts on an S3 bucket. It then handled automation of the project linking with AWS CodeDeploy which deployed the project to our EC2 instance. The instance had nginx setup as a web server which enabled rendering of the web application over the public IPV4 address.