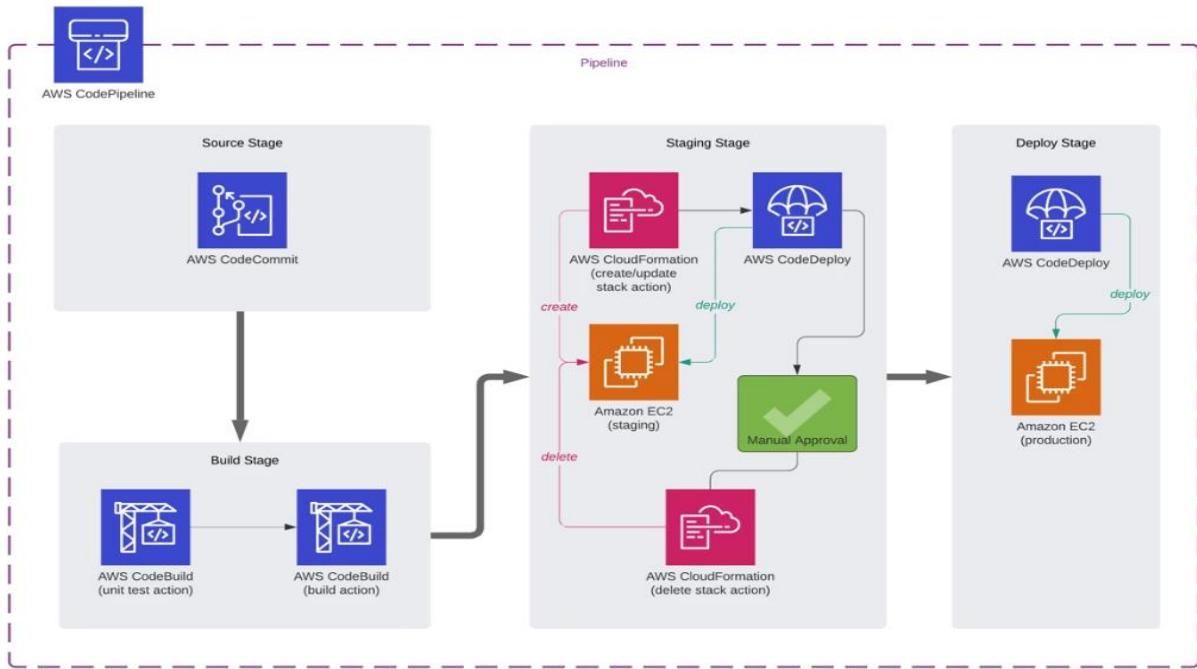


# cprime

## Code Deploy/Code Pipeline (CD/CP) on AWS



NAME : T MANOJ

EMPID : LYAKE2KHS

## **Introduction**

Continuous Delivery and Continuous Deployment (CD/CP) are crucial components of modern DevOps practices. They ensure that applications are delivered to users efficiently, reliably, and with minimal manual intervention. Leveraging AWS services such as EC2, S3, CodeDeploy, CodePipeline, CloudWatch, and SNS enables seamless automation, robust monitoring, and streamlined deployment processes.

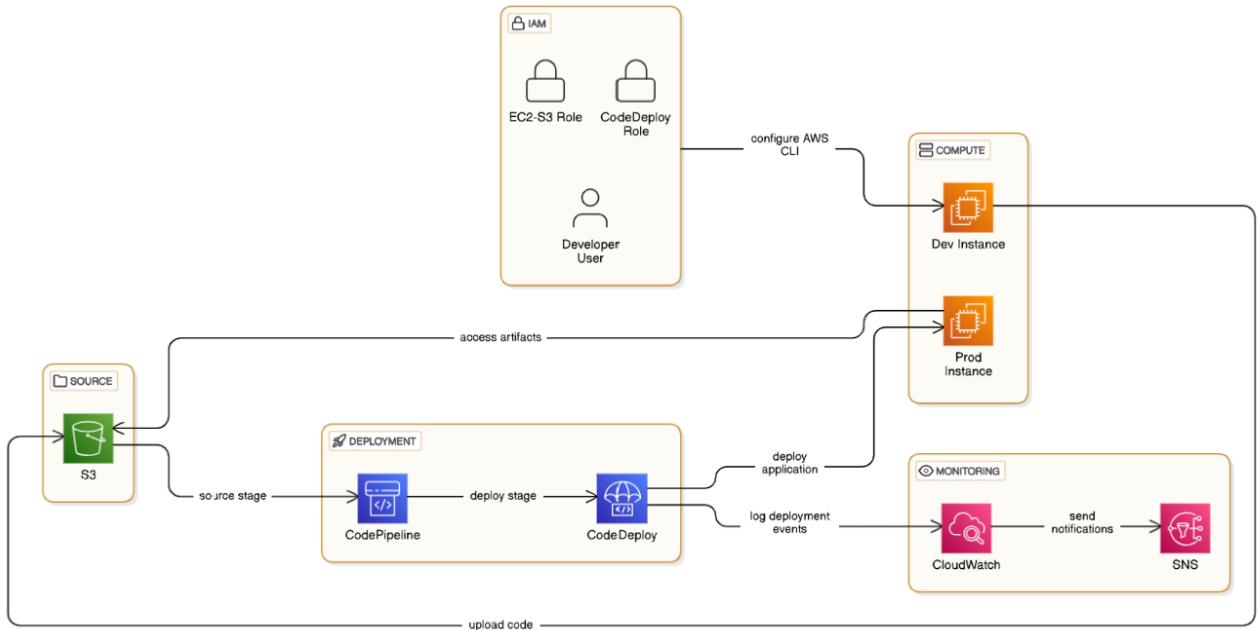
## **Overview**

This document outlines a step-by-step approach to implementing a CD/CP pipeline on AWS. The process involves creating development and production environments, automating deployments using CodePipeline and CodeDeploy, monitoring deployments with CloudWatch, and sending real-time notifications via SNS. The pipeline ensures rapid, consistent, and error-free deployments, reducing downtime and accelerating delivery cycles.

## **Prerequisites**

- **AWS Account:** Active AWS account with appropriate access.
- **IAM Roles and Users:** IAM roles for EC2 (Ec2-s3) and CodeDeploy (codedeploy), along with an IAM user for developers with CLI access.
- **EC2 Instances:** Developer and production instances configured appropriately.
- **AWS CLI:** Configured on the developer instance.
- **S3 Bucket:** Public bucket for storing application artifacts with versioning enabled.

**Goal: To automate the deployment of an application to an EC2 instance using AWS services like CodeDeploy and CodePipeline.**



## AWS Services Used:

- EC2 (Elastic Compute Cloud): Provides virtual servers for development and production.
- S3 (Simple Storage Service): Stores application code and deployment artifacts.
- CodeDeploy: Automates application deployments to EC2 instances.
- CodePipeline: Orchestrates the entire CD/CP pipeline.
- SNS (Simple Notification Service): Sends notifications for deployment events.
- CloudWatch: Monitors AWS resources and provides logging.
- IAM (Identity and Access Management): Manages permissions.

## Detailed WorkFlow:

### 1. Create IAM Roles:

- Role: Ec2-s3
  - Purpose: Grants EC2 instances permission to access S3.
  - Action:
    - Navigate to the IAM console.



- Create a new role.

This screenshot shows the 'Create role' wizard at Step 3: 'Name, review, and create'. The 'Use case' section is highlighted, showing options for 'AWS service', 'AWS account', 'Web identity', and 'SAML 2.0 federation'. The 'Service or use case' dropdown is set to 'EC2'.

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

- Select "EC2" as the trusted entity.
- Attach the "AmazonS3FullAccess" policy.

This screenshot shows the 'Role details' section of the 'Create role' wizard. It includes fields for 'Role name' (set to 'EC2-S3-ROLE') and 'Description' (set to 'Allows EC2 instances to call AWS services on your behalf').

**Role details**  
Role name  
Enter a meaningful name to identify this role.  
EC2-S3-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: \_+=,. @-/\[\]!#\$%^&\*();~`  
  
**Step 1: Select trusted entities**

- Name the role "Ec2-s3."

## Step 2: Add permissions

Edit

Permissions policy summary		
Policy name	Type	Attached as
<a href="#">AmazonS3FullAccess</a>	AWS managed	Permissions policy

- Role: codedeploy
  - Purpose: Grants CodeDeploy permission to deploy applications.
  - Action:
    - Create a new role.



- Attach the "AWSCodeDeployFullAccess" policy.

This screenshot shows the "Create role" wizard at Step 1: Select trusted entity. It lists three options: "Web identity", "SAML 2.0 federation", and "Custom trust policy". The "Custom trust policy" option is selected, with a sub-note: "Create a custom trust policy to enable others to perform actions in this account." Below this is a "Use case" section with a dropdown menu set to "CodeDeploy".

This screenshot shows the "Create role" wizard at Step 2: Add permissions. It shows the "Permissions policies" section with one policy listed: "AWSCodeDeployRole". The "Type" is "AWS managed". Below this is a section titled "Set permissions boundary - optional".

This screenshot continues from the previous one, showing the "Create role" wizard at Step 2: Add permissions. It shows the "Permissions policies" section with the "AWSCodeDeployRole" policy selected. The "Type" is "AWS managed". Below this is a section titled "Set permissions boundary - optional". At the bottom, there are "Cancel", "Previous", and "Next" buttons.

This screenshot shows the "Create role" wizard at Step 3: Name, review, and create. It displays the role details: Role name is "codedeploy", Type is "AWS Lambda", and Description is "Lambda role for AWS Lambda function". It also shows the "Permissions policies" section with the "AWSCodeDeployRole" policy selected. At the bottom, there are "Cancel", "Previous", and "Next" buttons.

- Name the role "codedeploy."

**Name, review, and create**

**Role details**

**Role name**  
Enter a meaningful name to identify this role.  
**CODEDEPLOY-ROLE**

**Description**  
Add a short explanation for this role.  
**Allows CodeDeploy to call AWS services such as Auto Scaling on your behalf.**

**Step 1: Select trusted entities**

## 2. Create IAM User (Developer):

- Purpose: Creates an IAM user for developers to interact with AWS via the CLI.
- Action:
  - Navigate to the IAM console.



- Create a new user.
- Provide a username.

**Specify user details**

**User details**

**User name**  
**DEV-USER**

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

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

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keypairs, you can generate them after you create this IAM user. [Learn more](#)

- Uncheck "AWS Management Console access."
- Attach "AmazonS3FullAccess" and "AWSCodeDeployFullAccess" policies.

The screenshot shows the final step of creating a new IAM user. The left sidebar lists three steps: Step 1 (Specify user details), Step 2 (Set permissions), and Step 3 (Review and create). The current step is highlighted in blue. The main area displays the user details and permissions summary.

**User details**

User name	DEV-USER	Console password type	None
		Require password reset	No

**Permissions summary**

Name	Type	Used as
AmazonS3FullAccess	AWS managed	Permissions policy
AWSCodeDeployFullAccess	AWS managed	Permissions policy

**Tags - optional**

- Create an access key and download the credentials.
- Choose "Amazon CLI" as the key type.

The screenshot shows the 'Users' page with one user listed. A success message is displayed in a green box.

**User created successfully**

You can view and download the user's password and email instructions for signing in to the AWS Management Console.

**Users (1) Info**

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

User name	Path	Groups	Last activity	MFA	Password age
DEV-USER	/	0	-	-	-

**Actions:** View user, Delete, Create user

### 3. Create EC2 Instances:

The screenshot shows the EC2 Instances page. At the top, there are buttons for Connect, Instance state, Actions, and Launch instances. The Launch instances button is highlighted in orange.

Last updated a minute ago

**Actions:** Connect, Instance state, Actions, Launch instances

- Developer Instance:
  - Purpose: Development environment.
  - Action:
    - Launch an EC2 instance.

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

Name  
DEV-SERVER [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.

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

Recents [Quick Start](#)

**Summary**

Number of instances [Info](#)  
1

Software Image (AMI)  
Amazon Linux 2023 AMI 2023.6.2...[read more](#)  
ami-05b10e08d247fb927

Virtual server type (instance type)  
t2.micro

Firewall (security group)  
New security group

Storage (volumes)

[Cancel](#) [Launch instance](#) [Preview code](#)

- Configure security group and storage (30GB).

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

Network [Info](#)  
vpc-0c33a120e93d325f9

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

Common security groups [Info](#)

Select security groups

default sg-0714a24d115a13f10 [X](#)  
VPC: vpc-0c33a120e93d325f9

Security groups that you add or remove here will be added to or removed from all your network interfaces.

**Summary**

Number of instances [Info](#)  
1

Software Image (AMI)  
Amazon Linux 2023 AMI 2023.6.2...[read more](#)  
ami-05b10e08d247fb927

Virtual server type (instance type)  
t2.micro

Firewall (security group)  
default

Storage (volumes)

[Cancel](#) [Launch instance](#) [Preview code](#)

- Production Instance:
  - Purpose: Hosts the deployed application.
  - Action:
    - Launch an EC2 instance.

The screenshot shows the 'Launch an instance' wizard. In the 'Name and tags' step, a tag named 'SampleApp' is selected. In the 'Application and OS Images (Amazon Machine Image)' step, an Amazon Linux 2023 AMI is chosen. The right panel displays summary information and a 'Launch instance' button.

- Attach the "Ec2-s3" IAM role in "Advanced details."

The screenshot shows the 'Advanced details' step where the IAM instance profile 'EC2-S3-ROLE' is selected. The right panel shows the 'Summary' and 'Launch instance' buttons. Below, the 'Instances' dashboard lists two running instances: 'SampleApp' and 'DEV-SERVER'.

Name	Instance ID	Instance state	Instance type	Status check
SampleApp	i-0df6ef564f0e192b2	Running	t2.micro	Initializing 2/2 checks passed
DEV-SERVER	i-0bb910e8b2784c51e	Running	t2.micro	View alarms + 2/2 checks passed

#### 4. Configure Developer Machine:

- Purpose: Configure AWS CLI on the developer instance.
- Action:
  - Connect to the developer instance via SSH (PuTTY).

# Access key 1

## Create access key

The image displays three sequential screenshots of the AWS IAM 'Create access key' wizard:

- Screenshot 1: Step 1 - Access key best practices & alternatives.** This step is optional and shows three use cases:
  - Command Line Interface (CLI)**: You plan to use this access key to enable the AWS CLI to access your AWS account.
  - Local code**: You plan to use this access key to enable application code in a local development environment to access your AWS account.
  - Application running on an AWS compute service**: You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.
- Screenshot 2: Step 2 - optional**. This step is optional and shows three steps:
  - Set description tag
  - Step 3
  - Retrieve access keys
- Screenshot 3: Step 3 - Retrieve access keys**. This step is optional and shows the results of the key creation:
  - Access key created**: A green banner states: "This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time."
  - Access key**: The access key ID is AKIAKEVXYYTCDIFAH6H, and the secret access key is displayed as a series of asterisks: \*\*\*\*\*. A 'Show' link is available to view the full secret key.
  - Access key best practices**: A list of best practices:
    - Never store your access key in plain text, in a code repository, or in code.
    - Disable or delete access key when no longer needed.
    - Enable least-privilege permissions.
    - Rotate access keys regularly.

- Switch to root user (sudo su -).

```
'~\_\#\#\#`      Amazon Linux 2023
~~\_\#\#\#\` 
~~\#\#\` 
~~\#\` 
~~V~'`-> https://aws.amazon.com/linux/amazon-linux-2023
~~` /` 
~~` /` 
~/` /` 
[ec2-user@ip-172-31-23-216 ~]$ aws configure
AWS Access Key ID [None]: AKIAJAXEVXYYTCPDIFAH6H
AWS Secret Access Key [None]: yTMscpfW5trY3xYWb5k5+G12ZNmWg+W8joVIAdzg
Default region name [None]: us-east-1
Default output format [None]:
[ec2-user@ip-172-31-23-216 ~]$
```

i-0bb910e8b2784c51e (DEV-SERVER)  
Public IPs: 54.242.71.101 Private IPs: 172.31.23.216

[CloudShell](#) [Feedback](#) © 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

- Run aws configure and provide IAM user credentials.

## 5. Install CodeDeploy Agent (Production Instance):

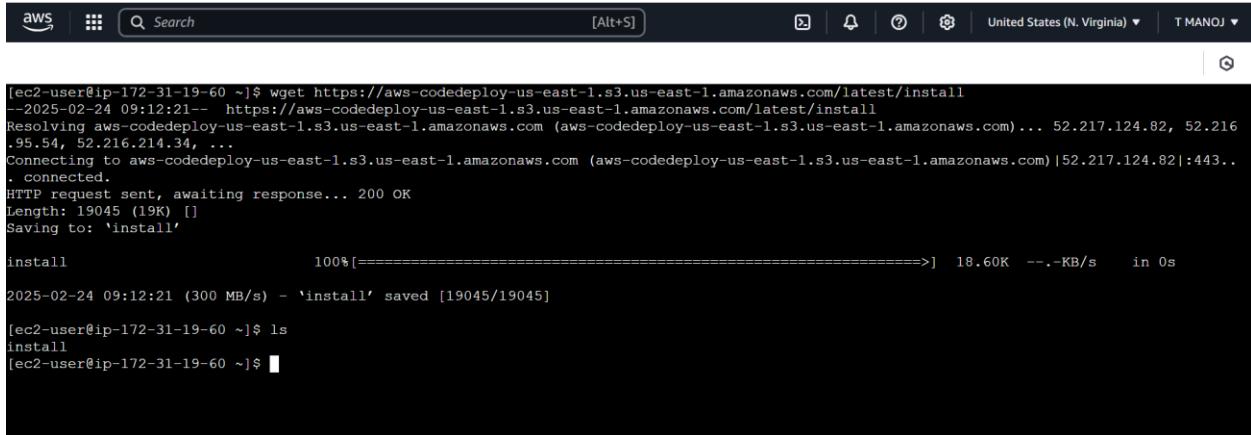
- Purpose: Install CodeDeploy agent on the production instance.
- Action:
  - Connect to the production instance via SSH.
  - Follow the AWS documentation:
    - yum update -y

```
'~\_\#\#\#`      Amazon Linux 2023
~~\_\#\#\#\` 
~~\#\#\#\` 
~~\#\#\` 
~~\#\` 
~~V~'`-> https://aws.amazon.com/linux/amazon-linux-2023
~~` /` 
~~` /` 
~/` /` 
[ec2-user@ip-172-31-19-60 ~]$ sudo yum update
Last metadata expiration check: 0:04:39 ago on Mon Feb 24 09:04:25 2025.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-19-60 ~]$
```

- yum install ruby
- yum install wget

```
[ec2-user@ip-172-31-19-60 ~]$ sudo yum install wget
Last metadata expiration check: 0:05:41 ago on Mon Feb 24 09:04:25 2025.
Package wget-1.21.3-1.amzn2023.0.4.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-19-60 ~]$
```

- wget <CodeDeploy agent download link>



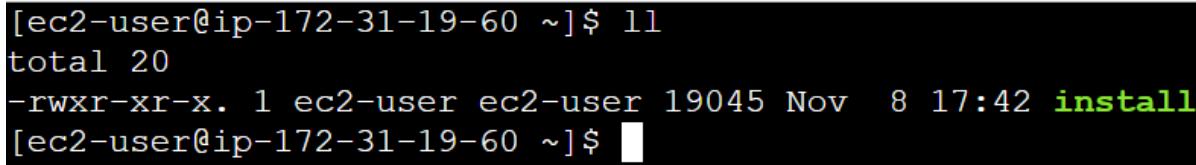
```
[ec2-user@ip-172-31-19-60 ~]$ wget https://aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com/latest/install
--2025-02-24 09:12:21-- https://aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com/latest/install
Resolving aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com (aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com) ... 52.217.124.82, 52.216
.95.54, 52.216.214.34, ...
Connecting to aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com (aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com)|52.217.124.82|:443...
HTTP request sent, awaiting response... 200 OK
Length: 19045 (19K) []
Saving to: 'install'

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

2025-02-24 09:12:21 (300 MB/s) - 'install' saved [19045/19045]

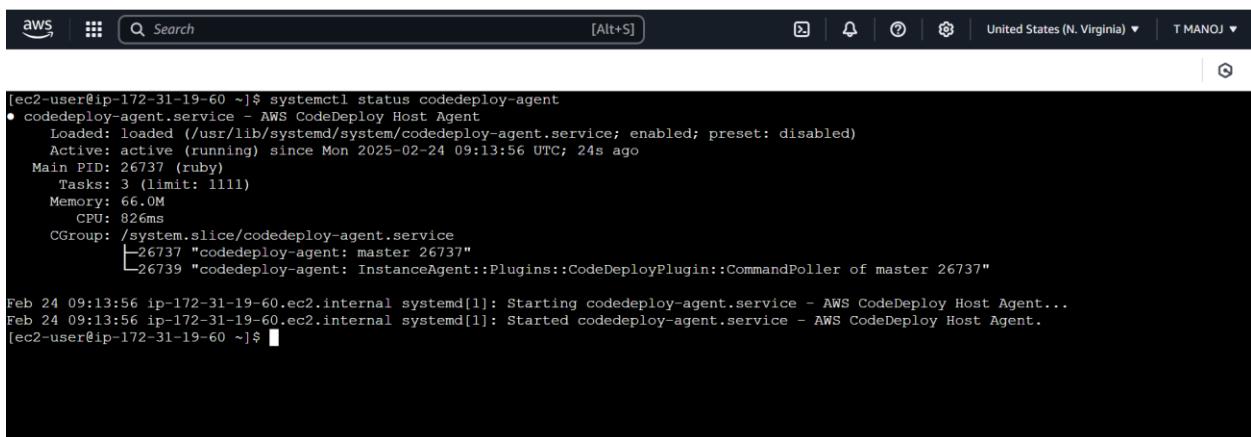
[ec2-user@ip-172-31-19-60 ~]$ ls
install
[ec2-user@ip-172-31-19-60 ~]$ 
```

- chmod +x ./install
- sudo ./install auto



```
[ec2-user@ip-172-31-19-60 ~]$ ll
total 20
-rwxr-xr-x. 1 ec2-user ec2-user 19045 Nov  8 17:42 install
[ec2-user@ip-172-31-19-60 ~]$ 
```

- systemctl status codedeploy-agent
- systemctl start codedeploy-agent

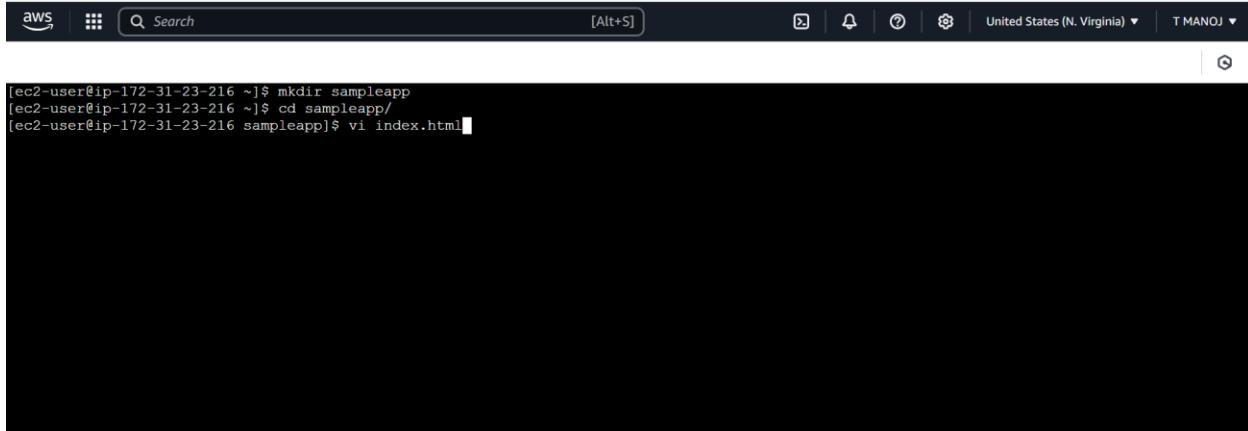


```
[ec2-user@ip-172-31-19-60 ~]$ systemctl status codedeploy-agent
● codedeploy-agent.service - AWS CodeDeploy Host Agent
    Loaded: loaded (/usr/lib/systemd/system/codedeploy-agent.service; enabled; preset: disabled)
      Active: active (running) since Mon 2025-02-24 09:13:56 UTC; 24s ago
        Main PID: 26737 (ruby)
           Tasks: 3 (limit: 1111)
         Memory: 66.0M
            CPU: 826ms
          CGroup: /system.slice/codedeploy-agent.service
                  └─26737 "codedeploy-agent: master 26737"
                      ├─26739 "codedeploy-agent: InstanceAgent::Plugins::CodeDeployPlugin::CommandPoller of master 26737"
                      └─26739 "codedeploy-agent: InstanceAgent::Plugins::CodeDeployPlugin::CommandPoller of master 26737"

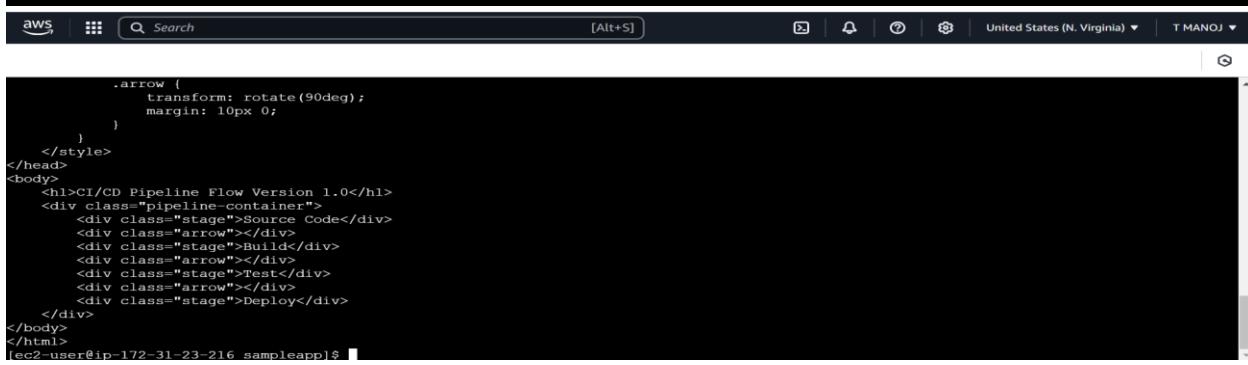
Feb 24 09:13:56 ip-172-31-19-60.ec2.internal systemd[1]: Starting codedeploy-agent.service - AWS CodeDeploy Host Agent...
Feb 24 09:13:56 ip-172-31-19-60.ec2.internal systemd[1]: Started codedeploy-agent.service - AWS CodeDeploy Host Agent.
[ec2-user@ip-172-31-19-60 ~]$ 
```

## 6. Create Sample Code:

- Purpose: Create application files and deployment scripts.
- Action:
  - On the developer instance:
    - Create index.html.



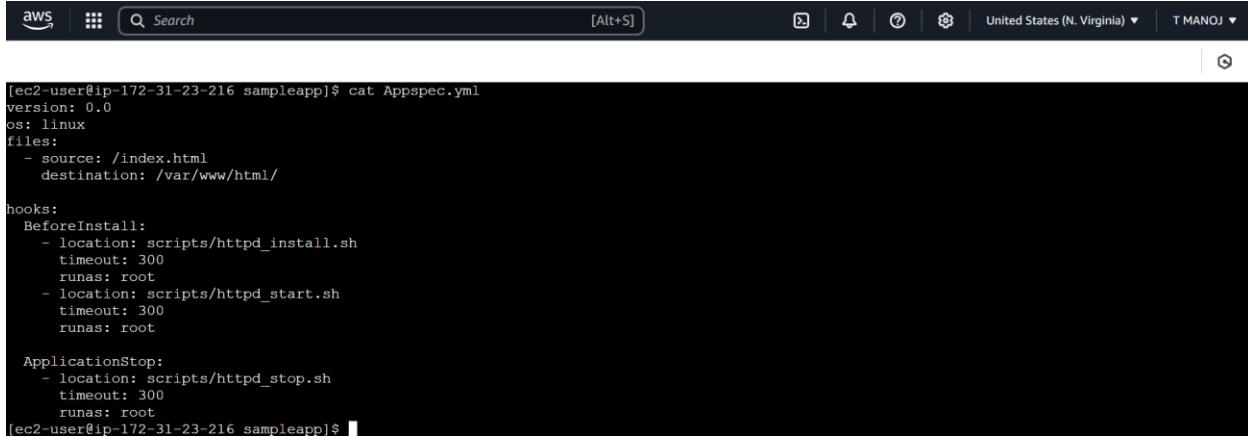
[ec2-user@ip-172-31-23-216 ~]\$ mkdir sampleapp  
[ec2-user@ip-172-31-23-216 ~]\$ cd sampleapp/  
[ec2-user@ip-172-31-23-216 sampleapp]\$ vi index.html



```
.arrow {
    transform: rotate(90deg);
    margin: 10px 0;
}
</style>
</head>
<body>
<h1>CI/CD Pipeline Flow Version 1.0</h1>
<div class="pipeline-container">
    <div class="stage">Source Code</div>
    <div class="arrow"></div>
    <div class="stage">Build</div>
    <div class="arrow"></div>
    <div class="stage">Test</div>
    <div class="arrow"></div>
    <div class="stage">Deploy</div>
</div>
</body>
</html>
```

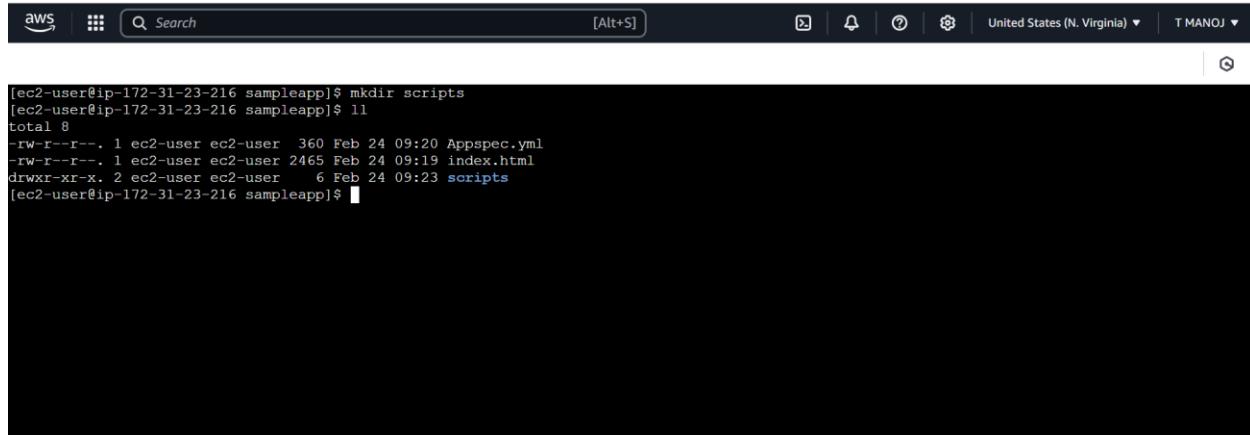
[ec2-user@ip-172-31-23-216 sampleapp]\$

- Create appspec.yml.

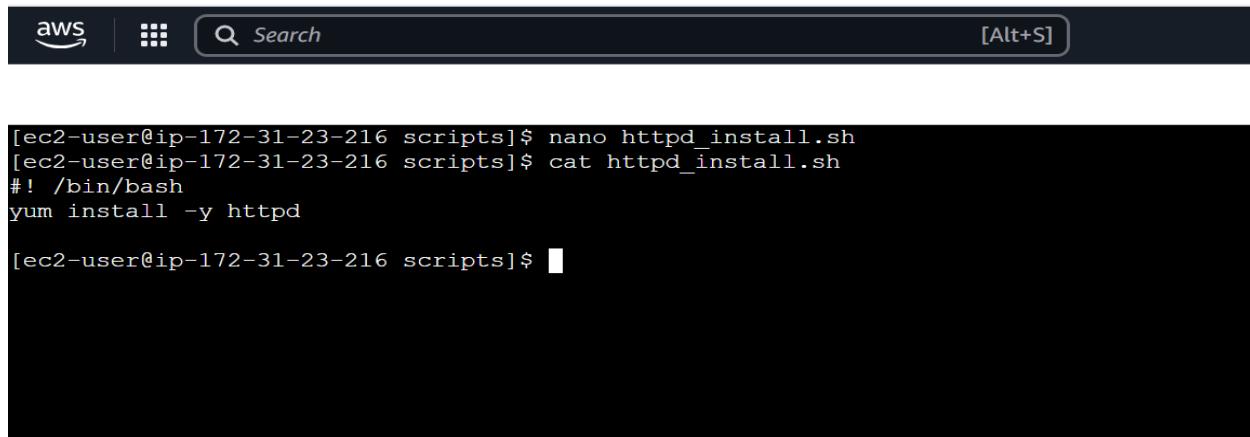


```
version: 0.0
os: linux
files:
- source: /index.html
  destination: /var/www/html/
hooks:
BeforeInstall:
- location: scripts/httpd_install.sh
  timeout: 300
  runas: root
- location: scripts/httpd_start.sh
  timeout: 300
  runas: root
ApplicationStop:
- location: scripts/httpd_stop.sh
  timeout: 300
  runas: root
```

- Create httpd\_start.sh and httpd\_stop.sh and httpd\_install.sh scripts.

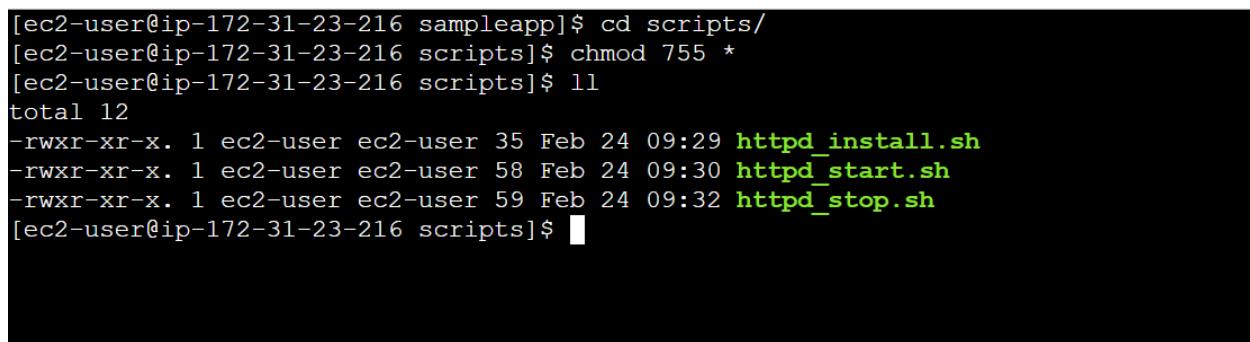


[ec2-user@ip-172-31-23-216 sampleapp]\$ mkdir scripts  
[ec2-user@ip-172-31-23-216 sampleapp]\$ ll  
total 8  
-rw-r--r--. 1 ec2-user ec2-user 360 Feb 24 09:20 Appspec.yml  
-rw-r--r--. 1 ec2-user ec2-user 2465 Feb 24 09:19 index.html  
drwxr-xr-x. 2 ec2-user ec2-user 6 Feb 24 09:23 scripts  
[ec2-user@ip-172-31-23-216 sampleapp]\$



[ec2-user@ip-172-31-23-216 scripts]\$ nano httpd\_install.sh  
[ec2-user@ip-172-31-23-216 scripts]\$ cat httpd\_install.sh  
#!/bin/bash  
yum install -y httpd  
  
[ec2-user@ip-172-31-23-216 scripts]\$

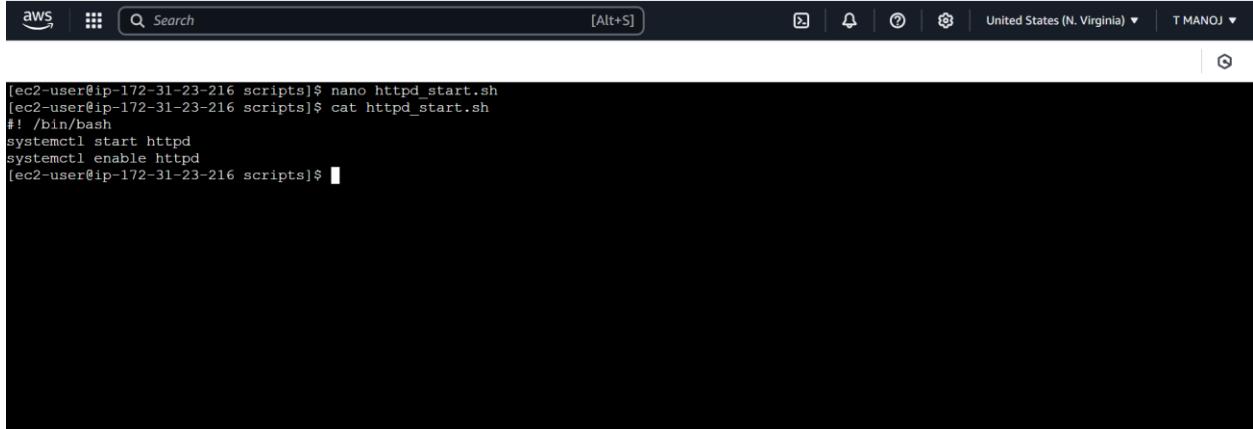
- Make scripts executable (chmod 775 \*).



[ec2-user@ip-172-31-23-216 sampleapp]\$ cd scripts/  
[ec2-user@ip-172-31-23-216 scripts]\$ chmod 755 \*  
[ec2-user@ip-172-31-23-216 scripts]\$ ll  
total 12  
-rwxr-xr-x. 1 ec2-user ec2-user 35 Feb 24 09:29 httpd\_install.sh  
-rwxr-xr-x. 1 ec2-user ec2-user 58 Feb 24 09:30 httpd\_start.sh  
-rwxr-xr-x. 1 ec2-user ec2-user 59 Feb 24 09:32 httpd\_stop.sh  
[ec2-user@ip-172-31-23-216 scripts]\$

```
[ec2-user@ip-172-31-23-216 scripts]$ cd ../
[ec2-user@ip-172-31-23-216 sampleapp]$ ll
total 8
-rw-r--r--. 1 ec2-user ec2-user 360 Feb 24 09:20 Appspec.yml
-rw-r--r--. 1 ec2-user ec2-user 2465 Feb 24 09:19 index.html
drwxr-xr-x. 2 ec2-user ec2-user 73 Feb 24 09:32 scripts
[ec2-user@ip-172-31-23-216 sampleapp]$ cd scripts/
[ec2-user@ip-172-31-23-216 scripts]$ ll
total 12
-rw-r--r--. 1 ec2-user ec2-user 35 Feb 24 09:29 httpd_install.sh
-rw-r--r--. 1 ec2-user ec2-user 58 Feb 24 09:30 httpd_start.sh
-rw-r--r--. 1 ec2-user ec2-user 59 Feb 24 09:32 httpd_stop.sh
[ec2-user@ip-172-31-23-216 scripts]$ █
```

- Inside of the http\_start.sh file include the commands to start the apache web server, and enable it.



The screenshot shows the AWS Cloud9 IDE interface. At the top, there's a navigation bar with icons for AWS Lambda, CloudWatch Metrics, CloudWatch Logs, CloudWatch Metrics Insights, CloudWatch Metrics Data API, CloudWatch Metrics Metrics Insights, and CloudWatch Metrics Metrics Data API. There's also a search bar, a 'Search' button, and a keyboard shortcut '[Alt+S]'. On the right side of the header, there are icons for notifications, a profile picture, and dropdown menus for 'United States (N. Virginia)' and 'T MANOJ'. Below the header, the main terminal window displays the following command-line session:

```
[ec2-user@ip-172-31-23-216 scripts]$ nano httpd_start.sh
[ec2-user@ip-172-31-23-216 scripts]$ cat httpd_start.sh
#!/bin/bash
systemctl start httpd
systemctl enable httpd
[ec2-user@ip-172-31-23-216 scripts]$ █
```

- Inside of the http\_stop.sh file include the commands to stop the apache web server, and disable it.

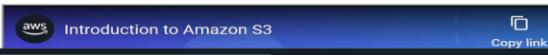
```
[ec2-user@ip-172-31-23-216 scripts]$ cat httpd_stop.sh
#!/bin/bash
systemctl stop httpd
systemctl disable httpd
[ec2-user@ip-172-31-23-216 scripts]$ █
```

## 7. Create S3 Bucket:

- Purpose: Store application artifacts.
- Action:
  - Create a public S3 bucket.

The screenshot shows the Amazon S3 homepage. At the top right, there is a 'Create a bucket' button. Below it, a section titled 'Pricing' states: 'With S3, there are no minimum fees. You only pay for what you use. Prices are based on the location of your S3 bucket.' A link to the 'AWS Simple Monthly Calculator' is provided.

### How it works



- Enable ACLs.

The screenshot shows the 'Create bucket' configuration page. Under 'Bucket type', there are two options: 'General purpose' (selected) and 'Directory'. The 'General purpose' option is described as recommended for most use cases and access patterns, allowing a mix of storage classes across multiple Availability Zones. The 'Directory' option is described as recommended for low-latency use cases, using the S3 Express One Zone storage class within a single Availability Zone.

### General configuration

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

Bucket type | [Info](#)

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

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

Bucket name | [Info](#)

sampleapp-24

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

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

The screenshot shows the 'Create bucket' configuration page. Under 'Object Ownership', there are two options: 'Bucket owner preferred' (selected) and 'Object writer'. The 'Bucket owner preferred' option is described as new objects written to this bucket specifying the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer. The 'Object writer' option is described as the object writer remaining the object owner. A note at the bottom states: 'If you want to enforce object ownership for new objects only, your bucket policy must specify that the bucket-owner-full-control canned ACL is required for object uploads. [Learn more](#)'

#### Block Public Access settings for this bucket

- Disable encryption.

**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.

General purpose buckets (1)			
<a href="#">Info</a> <a href="#">All AWS Regions</a>			
	<a href="#">Copy ARN</a>	<a href="#">Empty</a>	<a href="#">Delete</a>
<a href="#">Create bucket</a> Buckets are containers for data stored in S3.			
Find buckets by name <input type="text"/>			
Name	AWS Region	IAM Access Analyzer	Creation date
<a href="#">sampleapp-24</a>	US East (N. Virginia) us-east-1	<a href="#">View analyzer for us-east-1</a>	February 24, 2025, 15:05:35 (UTC+05:30)

## 8. Create CodeDeploy Application:

- Purpose: Manage application deployments.
- Action:
  - On the developer machine:
    - Run aws deploy create-application --application-name sampleapp.

```
[ec2-user@ip-172-31-23-216 sampleapp]$ aws deploy create-application --application-name sampleapp
{
    "applicationId": "435cbe28-c0c3-4879-b232-701f1d1231ff"
}
[ec2-user@ip-172-31-23-216 sampleapp]$
```

Developer Tools > CodeDeploy > Applications

Application name	Compute platform	Created
sampleapp	EC2/on-premises	Just now

- Zip the application files (zip -r sampleapp.zip \*).
- Upload to S3: aws deploy push --application-name sampleapp --s3-location s3://gir-sampleapp/sampleapp.zip.

```
[ec2-user@ip-172-31-23-216 sampleapp]$ aws deploy push --application-name sampleapp --s3-location s3://sampleapp-24/sampleapp.zip
To deploy with this revision, run:
aws deploy create-deployment --application-name sampleapp --s3-location bucket=sampleapp-24,key=sampleapp.zip,bundleType=zip,eTag=6b81b3c630
de44129cf97b621974fc0 --deployment-group-name <deployment-group-name> --deployment-config-name <deployment-config-name> --description <desc
ription>
[ec2-user@ip-172-31-23-216 sampleapp]$
```

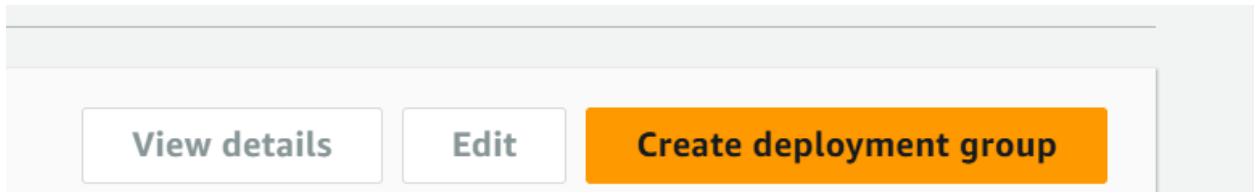
**sampleapp-24** [Info](#)

Objects	Metadata	Properties	Permissions	Metrics	Management	Access Points										
<b>Objects (1)</b>																
<input type="button" value="Create folder"/> <input type="button" value="Upload"/>	<input type="button" value="Copy S3 URI"/> <input type="button" value="Copy URL"/> <input type="button" value="Download"/> <input type="button" value="Open"/> <input type="button" value="Delete"/> <input type="button" value="Actions"/>															
<p>Objects are the fundamental entities stored in Amazon S3. You can use <a href="#">Amazon S3 inventory</a> to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. <a href="#">Learn more</a></p>																
<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Last modified</th> <th>Size</th> <th>Storage class</th> </tr> </thead> <tbody> <tr> <td><a href="#">sampleapp.zip</a></td> <td>zip</td> <td>February 24, 2025, 15:13:58 (UTC+05:30)</td> <td>1.6 KB</td> <td>Standard</td> </tr> </tbody> </table>							Name	Type	Last modified	Size	Storage class	<a href="#">sampleapp.zip</a>	zip	February 24, 2025, 15:13:58 (UTC+05:30)	1.6 KB	Standard
Name	Type	Last modified	Size	Storage class												
<a href="#">sampleapp.zip</a>	zip	February 24, 2025, 15:13:58 (UTC+05:30)	1.6 KB	Standard												

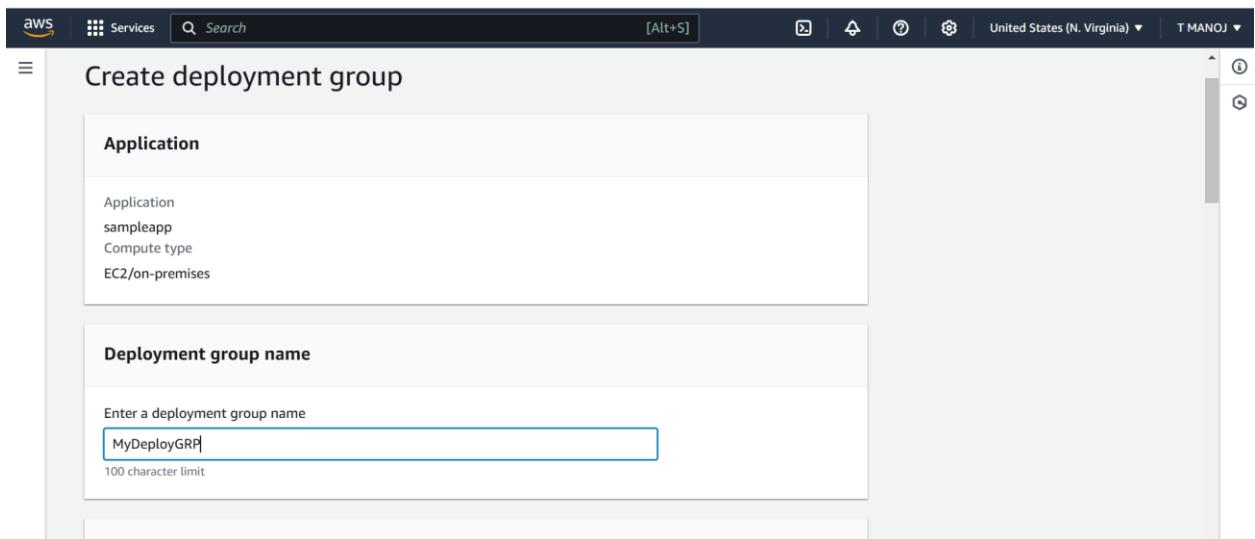
## 9. Create Deployment Group:

- Purpose: Define deployment settings.
- Action:

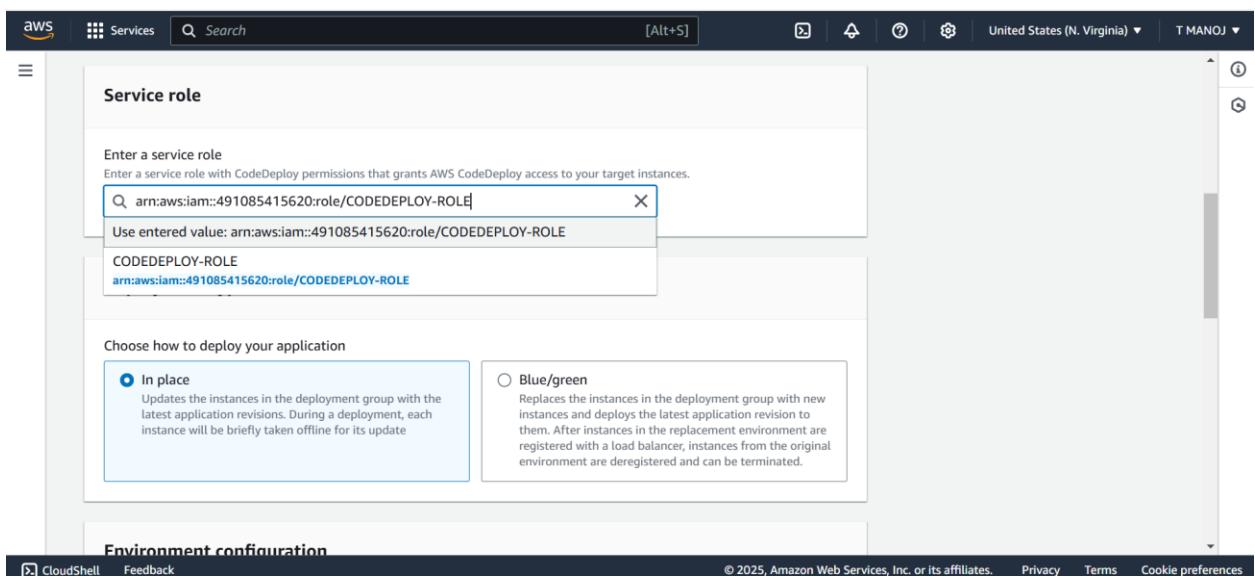
- In the CodeDeploy console:
  - Create a deployment group.



- Provide a name.



- Attach the "codedeploy" service role.



- Select "Amazon EC2 instances."

- Choose "In-place" deployment.
- Select the production instance.

The screenshot shows the AWS CodeDeploy console interface. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, search bar, and account information ('United States (N. Virginia) ▾ T MANOJ ▾'). Below the navigation is a search bar with placeholder text 'to this deployment'. Underneath, there are two checkboxes: 'Amazon EC2 Auto Scaling groups' (unchecked) and 'Amazon EC2 instances' (checked). A note below says '1 unique matched instance. Click here for details [i]'. Further down, instructions for tag groups are provided: 'You can add up to three groups of tags for EC2 instances to this deployment group.' and 'One tag group: Any instance identified by the tag group will be deployed to.' followed by 'Multiple tag groups: Only instances identified by all the tag groups will be deployed to.' A 'Tag group 1' section is shown with a key 'APPNAME' and value 'SampleApp', with buttons for 'Add tag' and '+ Add tag group'. Below this is an unchecked checkbox for 'On-premises instances'. A section titled 'Matching instances' shows '1 unique matched instance. Click here for details [i]'. At the bottom of the page are links for 'CloudShell', 'Feedback', and copyright information ('© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences').

- Disable load balancing.

The screenshot shows the AWS CodeDeploy console after creating a new deployment group. A green success message at the top says 'Success Deployment group created'. Below it, the breadcrumb navigation shows 'Developer Tools > CodeDeploy > Applications > sampleapp > MyDeployGRP'. The main title is 'MyDeployGRP' with buttons for 'Edit', 'Delete', and 'Create deployment'. Under the title, a section titled 'Deployment group details' lists the following configuration:

Deployment group name	Application name	Compute platform
MyDeployGRP	sampleapp	EC2/on-premises
Deployment type	Service role ARN	Deployment configuration
In place	<a href="#">arn:aws:iam::491085415620:role/CODE DEPLOY-ROLE</a>	<a href="#">CodeDeployDefault.AllAtOnce</a>
Rollback enabled	Agent update scheduler	
False	<a href="#">Learn to schedule update in AWS Systems Manager [i]</a>	

At the bottom of the page are links for 'CloudShell', 'Feedback', and copyright information ('© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences').

## 10. Create Deployment:

- Purpose: Deploy the application.
- Action:
  - In the CodeDeploy console:
    - Create a deployment.

- Select the S3 bucket and sampleapp.zip.

The image consists of three vertically stacked screenshots of the AWS console, specifically the CodeDeploy service.

**Screenshot 1: Amazon S3 Bucket Objects**

This screenshot shows the Amazon S3 console with the path `Buckets > sampleapp-24`. The left sidebar shows various bucket types like General purpose buckets, Directory buckets, Table buckets, etc. The main area displays 10 objects, all named `sampleapp.zip` and are zip files. The last modified date for the first object is February 24, 2025, 17:01:00 (UTC+05:30).

Name	Type	Version ID	Last modified	Size	Storage class
<code>sampleapp.zip</code>	zip	SpzmRLYBRxgcl03Uedg4SMd26kRD8ZcE	February 24, 2025, 17:01:00 (UTC+05:30)	2.3 KB	Standard
<code>sampleapp.zip</code>	zip	f85GxQ_ADkeAD5rse2aKNOq1t3Gogro	February 24, 2025, 16:57:57 (UTC+05:30)	2.3 KB	Standard
<code>sampleapp.zip</code>	zip	zLLcz9N8.GB6TNxfNsptLEq9FJW	February 24, 2025, 16:44:20	2.3 KB	Standard

**Screenshot 2: Developer Tools > CodeDeploy > Applications > sampleapp > Create deployment**

This screenshot shows the "Create deployment" page for the `sampleapp` application. Under "Deployment settings", the "Deployment group" field contains `MyDeployGRP`. Other settings include "Compute platform: EC2/on-premises" and "Deployment type: In place".

**Screenshot 3: Services > Application sampleapp > Create deployment**

This screenshot shows the same "Create deployment" page, but the "Deployment group" field is empty. The other settings are identical to the second screenshot.

Success  
Deployment created

Developer Tools > CodeDeploy > Deployments > d-RI3X4N8CB

## d-RI3X4N8CB

Copy deployment    Retry deployment

**Deployment status**

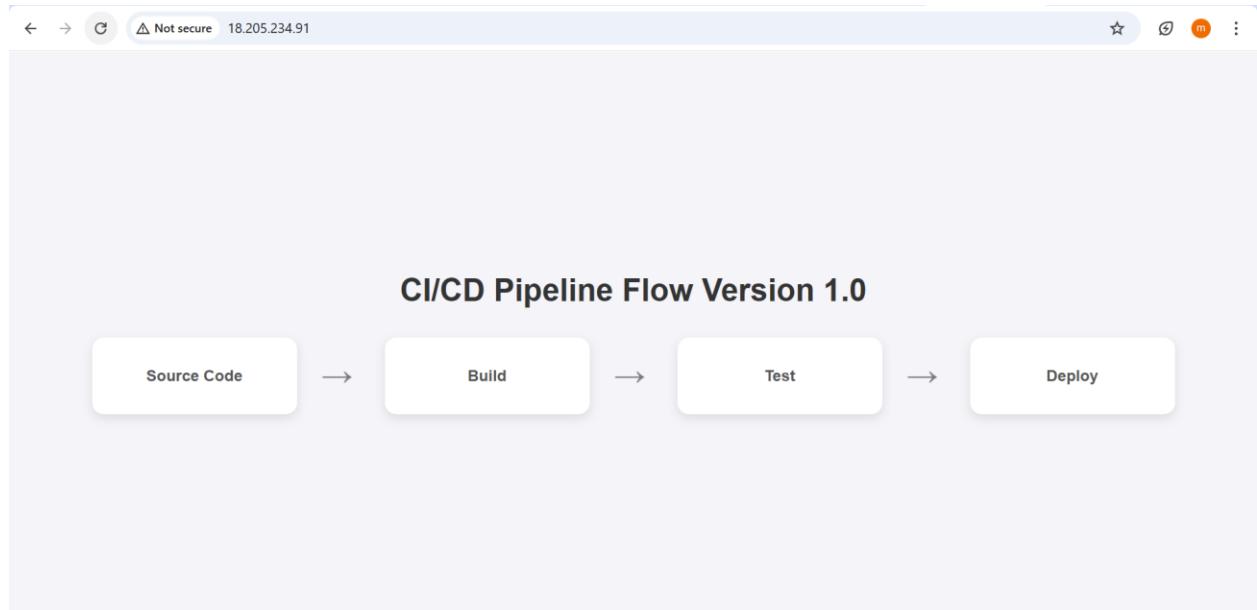
Installing application on your instances  
100%  
1 of 1 instances updated Succeeded

**Deployment details**

Application	Deployment ID	Status
sampleapp	d-RI3X4N8CB	<span style="color: green;">Succeeded</span>

## 11. Test the Output:

- Purpose: Verify the deployed application.
- Action:
  - Get the production instance's IP address.
  - Open a browser and enter the IP address.



## 12. Create CodePipeline:

- Purpose: Automate the deployment pipeline.
- Action:
  - In the CodePipeline console:
    - Create a pipeline.

Developer Tools > CodePipeline > Pipelines

Pipelines	Info		View history	Release change	Delete pipeline	<b>Create pipeline</b>
<input type="text"/> <span style="margin-left: 10px;">🔍</span> <span style="margin-left: 10px;">1</span> <span style="margin-left: 10px;">⚙️</span>						
Name	Latest execution status	Latest source revisions	Latest execution started	Most recent executions		
<b>No results</b> There are no results to display.						

aws Services Search [Alt+S] United States (N. Virginia) T MANOJ

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Choose creation option Info  
Step 1 of 7

Category

Deployment    Continuous Integration    Automation

**Build custom pipeline**

Cancel **Next**

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

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

AWS Services Search [Alt+S] United States (N. Virginia) T MANOJ

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

## Choose pipeline settings Info

Step 2 of 7

### Pipeline settings

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

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

- Source stage: Select S3, enable bucket versioning.

Amazon S3 > Buckets > sampleapp-24 > sampleapp.zip

Amazon S3 General purpose buckets

Directory buckets Table buckets Access Grants Access Points Object Lambda Access Points Multi-Region Access Points Batch Operations IAM Access Analyzer for S3

Block Public Access settings for this account

Storage Lens Dashboards

CloudShell Feedback

Successfully edited Bucket Versioning for "sampleapp-24"  
To transition, archive, or delete older object versions, [configure lifecycle rules](#) for this bucket.

sampleapp.zip Info

Properties Permissions Versions

Versions (0)

Version ID	Type	Last modified	Size	Storage class
No versions This object has no versions to display because Bucket Versioning has not been enabled for this bucket.				

CloudShell Feedback

Amazon S3 > Buckets > sampleapp-24 > sampleapp.zip

Amazon S3 General purpose buckets

Directory buckets Table buckets Access Grants Access Points Object Lambda Access Points Multi-Region Access Points Batch Operations IAM Access Analyzer for S3

Block Public Access settings for this account

Storage Lens Dashboards

CloudShell Feedback

Successfully edited Bucket Versioning for "sampleapp-24"  
To transition, archive, or delete older object versions, [configure lifecycle rules](#) for this bucket.

sampleapp.zip Info

Properties Permissions Versions

Versions (0)

Version ID	Type	Last modified	Size	Storage class
No versions This object has no versions to display because Bucket Versioning has not been enabled for this bucket.				

Screenshot of the AWS CodePipeline 'Create new pipeline' wizard Step 3: Add source stage.

**Source**

**Source provider**: Amazon S3

**Bucket**: sampleapp-24

**S3 object key**: sampleapp.zip

Enable automatic retry on stage failure

Cancel Previous Next

- Skip build and test stages.
- Deploy stage: Select CodeDeploy, choose the application and deployment group.

Screenshot of the AWS CodePipeline 'Create new pipeline' wizard Step 2: Choose pipeline settings.

**Pipeline settings**

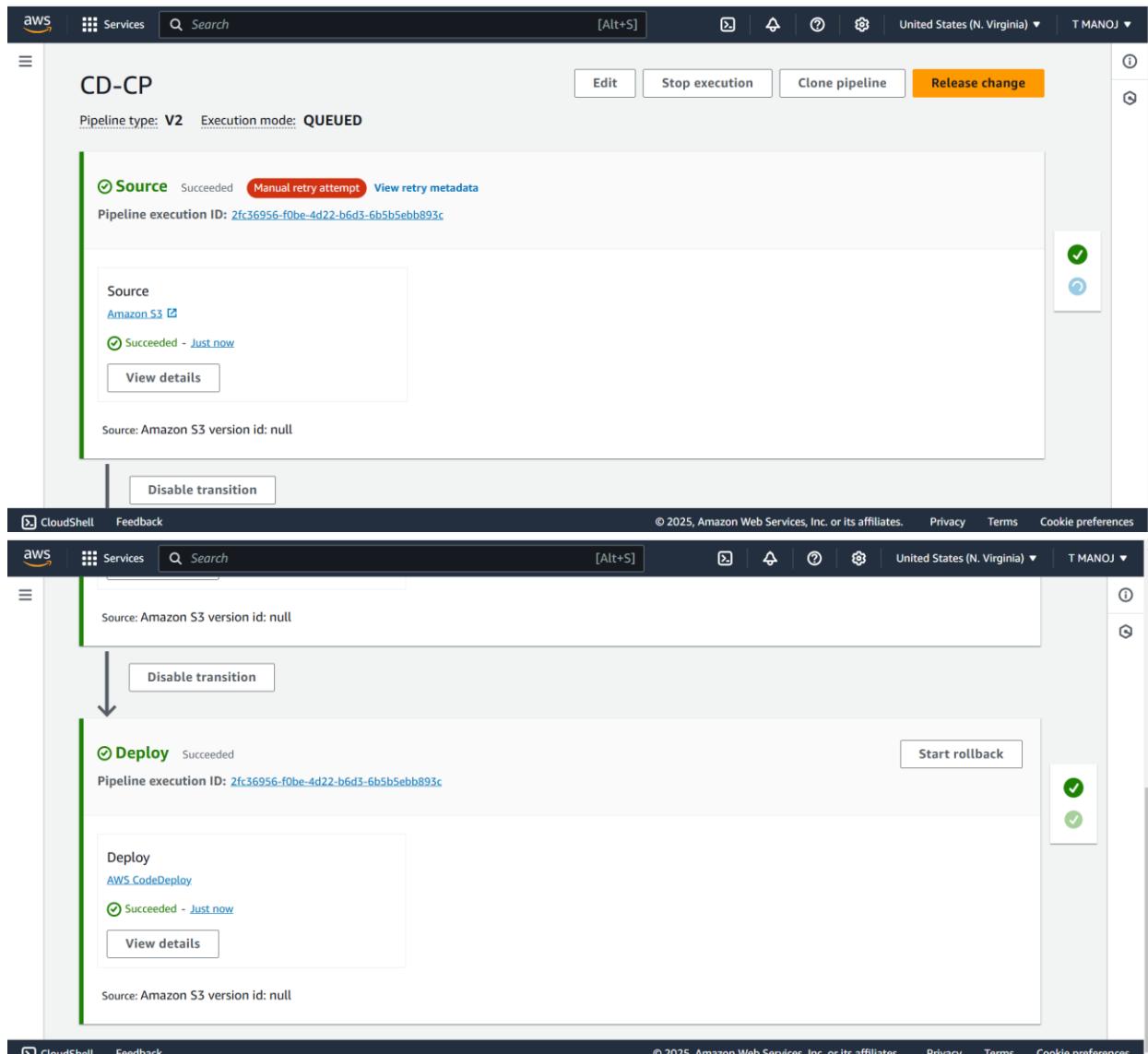
**Pipeline name**: CD-CP

**Pipeline type**: V2

**Execution mode**: QUEUED

**Artifact location**: A new Amazon S3 bucket will be created as the default artifact store for your pipeline

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



### 13. Testing:

- Purpose: Test the CodePipeline.
- Action:
  - On the developer instance:
    - Change the source code.
    - Zip the application.
    - Upload the zip to S3.
    - The CodePipeline should trigger a deployment.

Enhancements:

- Manual Approval: Add an approval stage in CodePipeline.

The screenshot shows the AWS CodePipeline interface for editing a manual approval stage. The top section is titled "Edit: Manual-Approval" with buttons for "Add stage", "Cancel", "Delete", and "Done". Below this are buttons for "Add entry condition", "Add success condition", and "Add failure condition", along with a "+ Add action group" button. A dropdown for "Automated stage configuration" is set to "None". The middle section contains a detailed view of the stage configuration, including a "Conditions" table with a single row for "Manual\_approval" (Status: Not configured). An "Edit stage" button is available here. The bottom section is titled "Edit: Deploy" and includes a "Edit stage" button. The footer of the page includes copyright information (© 2025, Amazon Web Services, Inc. or its affiliates.) and links for Privacy, Terms, and Cookie preferences.

+ Add stage

Edit: Manual-Approval

Add entry condition ▾ Add success condition ▾ Add failure condition

+ Add action group

Automated stage configuration: None ▾

Conditions Entry: Not configured Success: Not configured Failure: Not configured View details

Manual\_approval ⓘ

Manual approval

Automated stage configuration: None ▾

+ Add stage

Edit: Deploy

Edit stage

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

The screenshot shows the 'Create topic' page in the AWS SNS console. The 'Type' dropdown is set to 'FIFO (first-in, first-out)', which is highlighted with a blue border. The 'Name' field contains 'CDCP'. The 'High throughput' section is collapsed. At the bottom, there are links for CloudShell, Feedback, and a copyright notice.

The screenshot shows a success message from the AWS SNS service. It reads: 'Subscription confirmed! You have successfully subscribed. Your subscription's id is: arn:aws:sns:us-east-1:491085415620:CDP:b7d8f6ca-9103-4cb8-ba98-c9cde9e027be. If it was not your intention to subscribe, [click here to unsubscribe](#).'

- CloudWatch Monitoring and Notifications: Set up CloudWatch alarms and SNS notifications for CodeDeploy and CodePipeline.

The screenshot shows the 'Trails' page in the AWS CloudTrail console. A trail named 'codepipeline-source-trail' is selected. The 'General details' section shows the trail is logging to the 'codepipeline-cloudtrail-placeholder-bucket-us-east-1/cloud-trail-491085415620-3106bac4-ac2-4f2a-aa97-293b1e2e0315/AWSLogs/491085415620' bucket. The 'CloudWatch Logs' section is visible at the bottom.

The screenshot shows the 'Amazon EventBridge > Rules' section. A single rule is listed:

- Rule details** (Info)
 

<b>Rule name</b>	<b>Status</b>	<b>Event bus name</b>	<b>Type</b>
codepipeline-41382830-sampleapp-rule	Enabled	default	Standard
- Description**: Amazon CloudWatch Events rule to automatically start your pipeline when a change occurs in the Amazon S3 object key or S3 folder. Deleting this may prevent changes from being detected in that pipeline. Read more: <http://docs.aws.amazon.com/codepipeline/latest/userguide/pipelines-about-starting.html>
- Rule ARN**: arn:aws:events:us-east-1:491085415620:rule/codepipeline-41382830-sampleapp-rule
- Event bus ARN**: arn:aws:events:us-east-1:491085415620:event-bus/default

- Notifications: Configure SNS notifications for deployment success and failure.

The screenshot shows the 'Create notification rule' page in the 'Developer Tools > CodeDeploy > Applications > sampleapp > Notification rules' section. The 'Notification rule settings' form is filled out:

<b>Notification name</b>	CDCPNOTIFICATION	
<b>Detail type</b>	<input checked="" type="radio"/> <b>Full</b> Includes any supplemental information about events provided by the resource or the notifications feature.	<input type="radio"/> <b>Basic</b> Includes only information provided in resource events.

AWS Services Search [Alt+S] United States (N. Virginia) T MANOJ

Events that trigger notifications

Select none Select all

Deployment

Succeeded  
 Failed  
 Started

Targets

Create a target to use specifically for this notification rule. SNS topics created as targets have no subscribers but have all policies applied to act as a target for notifications. If you choose AWS Chatbot, you will be redirected to create a client in the AWS Chatbot console. [Learn more](#)

Configured targets

Choose target type Choose target

SNS topic arn:aws:sns:us-east-1:49108541... Remove row

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

aws Services Search [Alt+S] United States (N. Virginia) T MANOJ

Notification rule created

Developer Tools > CodeDeploy > Applications > sampleapp > Notification rules > View notification rule

## CDCPNOTIFICATION notification rule

View all notification rules Edit Delete

**Notification rule settings**

Notification name	Notification ARN
CDCPNOTIFICATION	arn:aws:codestar-notifications:us-east-1:491085415620:notificationrule/5475f3935e39cd52439cc6ddd4d57e320be8145b
Notification status	Resource ARN
<input checked="" type="radio"/>	arn:aws:codedeploy:us-east-1:491085415620:application:sampleapp

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## Notification rule targets [Learn more about creating rules.](#)

Delete

Protocol	Address or endpoint	Notification target status
<input checked="" type="radio"/> SNS	<a href="#">arn:aws:sns:us-east-1:491085...</a>	<input checked="" type="radio"/> Active

Screenshot of the AWS CodePipeline console showing the pipeline named "CD-CP".

The pipeline type is V2 and the execution mode is QUEUED. The current step is the Source stage, which has succeeded. The pipeline execution ID is 9b87acea-8c00-40a4-8dae-def4f9e56317.

The source is an Amazon S3 bucket, and it was successful just now. A "View details" button is available. The source information is: Source: Amazon S3 version id: PWI6TkmSMRTaFTch7U050m.96sFW30i.

At the bottom, there are buttons for "Edit", "Stop execution", "Clone pipeline", and "Release change". On the right side, there are three green checkmark icons.

Screenshot of a Gmail inbox showing an email from AWS Notifications.

The email subject is "AWS Notifications <no-reply@sns.amazonaws.com> to me". It was sent at 15:56 (0 minutes ago). The message content is:

Hello,

The following Approval action is waiting for your response:

--Pipeline Details--

Pipeline name: CD-CP  
Stage name: Manual-Approval  
Action name: Manual\_approval  
Region: us-east-1

--Approval Details--

Approve or reject: [https://console.aws.amazon.com/codesuite/codepipeline/pipelines/CD-CP/view?region=us-east-1#/Manual-Approval/Manual\\_approval/approve/a2f60e7a-246a-4a04-bc4e-79e2cdb821bc](https://console.aws.amazon.com/codesuite/codepipeline/pipelines/CD-CP/view?region=us-east-1#/Manual-Approval/Manual_approval/approve/a2f60e7a-246a-4a04-bc4e-79e2cdb821bc)  
Deadline: This review request will expire on 2025-03-03T10:26Z

Sincerely,  
Amazon Web Services

**Review**

Action name: Manual\_approval Status: Waiting for approval

---

**Details**    **Revisions**

**Trigger**  
**CloudWatchEvent** - [codepipeline-41382830-sampleapp-rule](#)

**Comments about this action**  
 -

**URL for review**

**Decision**

**Approve**  
 Approving will resume the pipeline execution.

**Reject**  
 Rejecting will stop the pipeline execution with a failed status.

**Comments - optional**  
 Preview markdown  
[Learn more](#)

---

**Cancel**    **Submit**

**aws**   **Services**   **Search**   [Alt+S]     United States (N. Virginia) ▾   T MANOJ ▾

Source: Amazon S3 version id: PWI6TkmSMRTaFTch7U050m.96sFW30i

**Disable transition**

**Deploy**   Succeeded  
 Pipeline execution ID: [9b87acea-8c00-40a4-8dae-def4f9e56317](#)   **Start rollback**

**Deploy**  
[AWS CodeDeploy](#)  
 **Succeeded** - Just now  
**View details**

Source: Amazon S3 version id: PWI6TkmSMRTaFTch7U050m.96sFW30i

**CloudShell**   **Feedback**   © 2025, Amazon Web Services, Inc. or its affiliates.   **Privacy**   **Terms**   **Cookie preferences**

Gmail

Compose

Inbox 504

Starred

Snoozed

Sent

Drafts 2

More

Labels +

Search mail

AWS Notification Message Inbox

AWS Notifications <no-reply@sns.amazonaws.com> to me ▾ 16:58 (0 minutes ago)

{"account": "491085415620", "detailType": "CodeDeploy Deployment State-change Notification", "region": "us-east-1", "source": "aws.codedeploy", "time": "2025-02-24T11:28:41Z", "notificationRuleArn": "arn:aws:codestar-notifications:us-east-1:491085415620:notificationrule/30ec9fe232a276fb0644f1391056a930ee20e27", "detail": {"application": "sampleapp", "deploymentId": "d-7CHKCABC", "deploymentGroup": "MyDeployGRP", "instanceGroupId": "2214ff5e-da5e-4327-bdff-2111629e34fb", "state": "START", "region": "us-east-1"}, "resources": ["arn:aws:codedeploy:us-east-1:491085415620:application:sampleapp", "arn:aws:codedeploy:us-east-1:491085415620:deploymentgroup:sampleapp/MyDeployGRP"], "additionalAttributes": {}}

If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:  
<https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-1:491085415620:notification:fe53efc9-9497-4675-a825-f5b52a8b834&Endpoint=dreamers2k22@gmail.com>

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at  
<https://aws.amazon.com/support>

AWS Notifications <no-reply@sns.amazonaws.com> 16:58 (0 minutes ago)

mail.google.com/mail/u/0/#inbox/FFfcgzQZTMTtMFPzsLCGqXgvIPmrpSfm

Gmail

Compose

Inbox 504

Starred

Snoozed

Sent

Drafts 2

More

Labels +

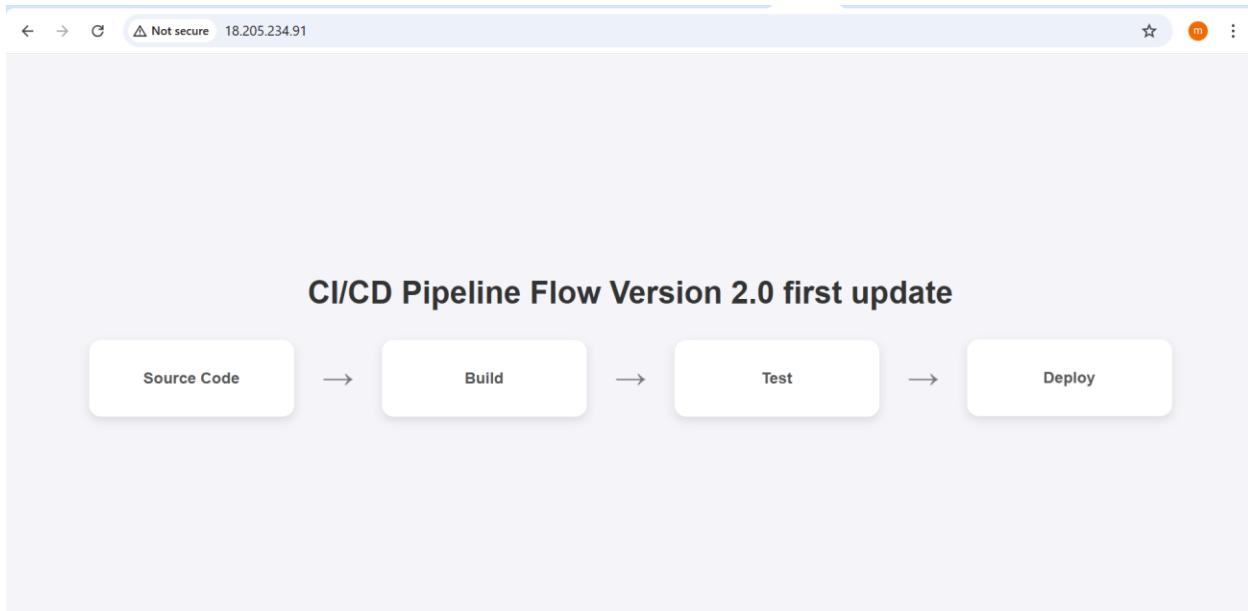
Search mail

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at  
<https://aws.amazon.com/support>

AWS Notifications <no-reply@sns.amazonaws.com> to me ▾ 16:58 (1 minute ago)

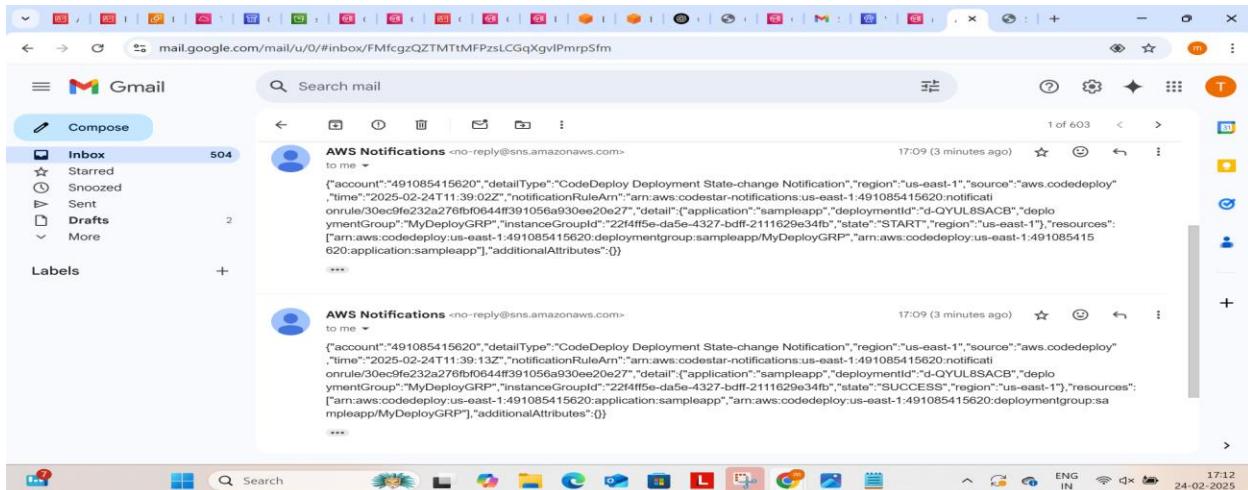
{"account": "491085415620", "detailType": "CodeDeploy Deployment State-change Notification", "region": "us-east-1", "source": "aws.codedeploy", "time": "2025-02-24T11:28:52Z", "notificationRuleArn": "arn:aws:codestar-notifications:us-east-1:491085415620:notificationrule/30ec9fe232a276fb0644f1391056a930ee20e27", "detail": {"application": "sampleapp", "deploymentId": "d-7CHKCABC", "deploymentGroup": "MyDeployGRP", "instanceGroupId": "2214ff5e-da5e-4327-bdff-2111629e34fb", "state": "SUCCESS", "region": "us-east-1"}, "resources": ["arn:aws:codedeploy:us-east-1:491085415620:application:sampleapp", "arn:aws:codedeploy:us-east-1:491085415620:deploymentgroup:sampleapp/MyDeployGRP"], "additionalAttributes": {}}

Reply Forward



## Rejecting the Manual Approval :

This screenshot shows the AWS CloudWatch Pipelines console. The top navigation bar includes 'aws', 'Services', a search bar, and account information for 'United States (N. Virginia)' and 'T MANOJ'. The main area displays a pipeline execution step named 'Manual\_approval'. The status is 'Failed' with a red circular icon. A message indicates 'Pipeline execution ID: d5dc15be-25f4-45f0-b0bd-4996819bd44e'. Below the status are three buttons: 'Start rollback', 'Retry stage', and 'Retry failed actions'. To the right of the status are three colored circular icons: green, red, and green. A red vertical line highlights the 'Rejected' status, which is also shown in a tooltip above the step. The step details show 'Manual approval' and 'Rejected - Just now'. A 'View details' button is present. At the bottom of the step, there is another 'Disable transition' button. The footer of the page includes links for 'CloudShell', 'Feedback', and copyright information: '© 2025, Amazon Web Services, Inc. or its affiliates.', 'Privacy', 'Terms', and 'Cookie preferences'.



## Advantages

- Automation: Reduces manual tasks through automated pipelines.
- Faster Deployments: Enables rapid and frequent releases.
- Reliability: Minimizes human errors and ensures consistent deployments.
- Scalability: Easily adaptable for scaling applications across multiple environments.
- Monitoring & Alerts: Integrated CloudWatch and SNS for real-time monitoring and notifications.
- Cost-Effective: Pay-as-you-go AWS services optimize resource usage and costs.
- Improved Collaboration: Facilitates seamless integration between development and operations teams.

## Conclusion

Implementing a CD/CP pipeline on AWS optimizes the deployment process by automating critical stages, ensuring faster time-to-market, and enhancing application reliability. The integration of AWS services like CodeDeploy, CodePipeline, and CloudWatch provides robust monitoring and efficient management. With proper configuration, this pipeline becomes a vital asset in any DevOps workflow, promoting agility, consistency, and scalability in application deployment cycles.