

Advance Devops-2

Aim: To Build Your Application using AWS CodeBuild and Deploy on S3 / SEBS using AWS Code Pipeline, deploy Sample Application on EC2 instance using AWS Code Deploy.

Theory:

Continuous deployment allows you to deploy revisions to a production environment automatically without explicit approval from a developer, making the entire software release process automated. You will create the pipeline using AWS CodePipeline, a service that builds, tests, and deploys your code every time there is a code change. You will use your GitHub account, an Amazon Simple Storage Service (S3) bucket, or an AWS CodeCommit repository as the source location for the sample app's code. You will also use AWS Elastic Beanstalk as the deployment target for the sample app. Your completed pipeline will be able to detect changes made to the source repository containing the sample app and then automatically update your live sample app.

Step 1: CREATING ROLE

The screenshot shows the AWS IAM console interface. On the left, the 'Identity and Access Management (IAM)' sidebar is visible with a search bar and navigation links for Dashboard, Access management, User groups, Users, Roles, Policies, and Identity providers. The main content area displays the details for a user named 'arnav'. The 'Summary' section includes the following information:

ARN	Console access	Access key 1
arn:aws:iam::891376943264:user/arnav	Disabled	AKIA47CRUYCQMJB265 - Active
Created August 29, 2024, 00:48 (UTC+05:30)	Last console sign-in -	Access key 2 Create access key

The screenshot shows the 'Permissions policies' section of the AWS IAM console for the 'arnav' user. The 'Permissions policies (2)' section is active, showing a list of attached policies. The table below lists the policies:

Policy name	Type	Attached entities
AWSElasticBeanstalkEnhancedHealth	AWS managed	1
AWSElasticBeanstalkService	AWS managed	1

Below the table, there is a section for 'Permissions boundary (not set)' and a section for 'Generate policy based on CloudTrail events'.

Step 2: CREATING ENVIRONMENT

Environment update successfully completed.

Elastic Beanstalk > Environments

Arnav_beanStalk info

Environment overview

Health

Pending

Domain

KomalBeanstalk-env.eba-uys9suwn.ap-south-1.elasticbeanstalk.com

Environment ID

e-nzfjjupfar

Application name

Platform

Change version

Platform

PHP 8.3 running on 64bit Amazon Linux 2023/4.3.2

Running version

code-pipeline-1724847940332-2f9a2b526f7dd091d2c97d8679735dcee01cd2bc

Platform state

Supported

Events

Health

Logs

Monitoring

Alarms

Managed updates

Tags

Events (16) info

August 28, 2024 17:56:37 (UTC+5:30)	INFO	Environment update completed successfully.
August 28, 2024 17:56:37 (UTC+5:30)	INFO	New application version was deployed to running EC2 instances.
August 28, 2024 17:56:15 (UTC+5:30)	INFO	Instance deployment completed successfully.
August 28, 2024 17:56:08 (UTC+5:30)	INFO	Instance deployment: You didn't include a 'composer.json' file in your source bundle. The deployment didn't install Composer dependencies.
August 28, 2024 17:56:04 (UTC+5:30)	INFO	Deploying new version to instance(s).
August 28, 2024 17:55:42 (UTC+5:30)	INFO	Environment update is starting.
August 28, 2024 17:52:57 (UTC+5:30)	INFO	Successfully launched environment: KomalBeanstalk-env
August 28, 2024 17:52:05 (UTC+5:30)	INFO	Added instance [i-0759f922553b2bed7] to your environment.
August 28, 2024 17:51:52 (UTC+5:30)	INFO	Instance deployment completed successfully.
August 28, 2024 17:51:47 (UTC+5:30)	INFO	Instance deployment: You didn't include a 'composer.json' file in your source bundle. The deployment didn't install Composer dependencies.
August 28, 2024 17:51:23 (UTC+5:30)	INFO	Waiting for EC2 instances to launch. This may take a few minutes.
August 28, 2024 17:51:08 (UTC+5:30)	INFO	Created EIP: 65.0.39.79

Step 3: PIPELINE CREATION

Disable transition

Deploy

Succeeded

Pipeline execution ID: [cafd4498-c2af-4ede-910c-27647c9721b0](#)

Deploy

[AWS Elastic Beanstalk](#)

Succeeded - 1 minute ago

View details

[2f9a2b52](#) Source: Update index.html

Start rollback

✓

✓

Developer Tools > CodePipeline > Pipelines

ArnavPipeLine

Notify

Edit

Stop execution

Clone pipeline

Release change

Pipeline type: V2 Execution mode: QUEUED

Source

Succeeded

Pipeline execution ID: [cafd4498-c2af-4ede-910c-27647c9721b0](#)

Source

[GitHub \(Version 2\)](#)

Succeeded - 2 minutes ago

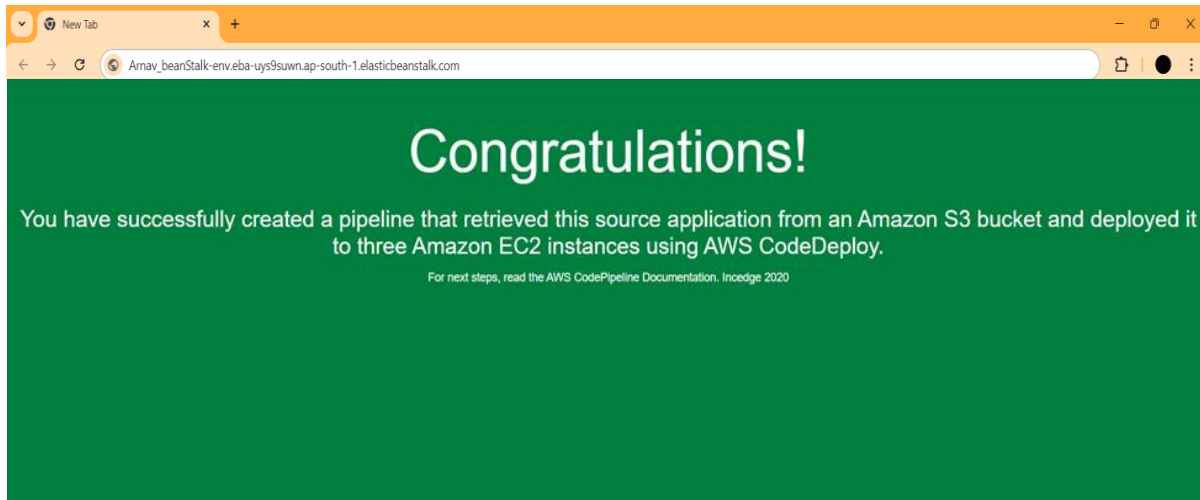
[2f9a2b52](#)

View details

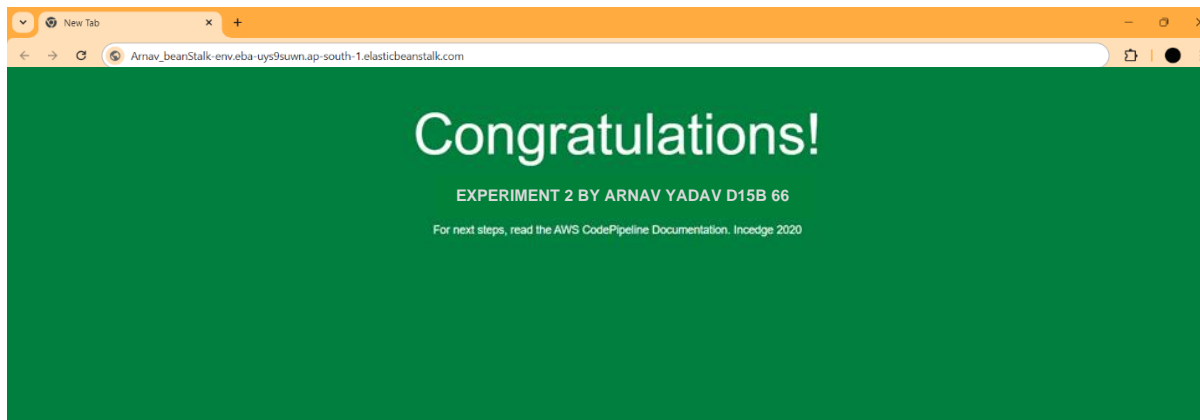
[2f9a2b52](#) Source: Update index.html

Disable transition

Step 4 :BEFORE UPDATING



Step 4 :After UPDATING



Conclusion:

Building and deploying an application using AWS CodeBuild, Code Pipeline, and Code Deploy demonstrates the power of automated CI/CD in the cloud. AWS CodeBuild compiles code, runs tests, and prepares software packages, while Code Pipeline automates the release process, ensuring faster and consistent deployments. Deploying to S3 or SEBS enables scalable hosting of static and serverless applications, and Code Deploy manages the deployment to EC2 instances, ensuring minimal downtime and easy rollback. This streamlined approach enhances development efficiency, reduces errors, and accelerates application delivery, showcasing the benefits of cloud-based automation and infrastructure management.