

Experiment 2

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

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.

Output:

The screenshot shows the AWS Elastic Beanstalk 'Configure environment' console page. The left sidebar contains a navigation menu with steps: Step 1 (Configure environment), Step 2 (Configure service access), Step 3 - optional (Set up networking, database, and tags), Step 4 - optional (Configure instance traffic and scaling), Step 5 - optional (Configure updates, monitoring, and logging), and Step 6 (Review). The main content area is titled 'Configure environment' and includes an 'Info' link. It features two sections: 'Environment tier' and 'Application information'. The 'Environment tier' section has two radio buttons: 'Web server environment' (selected) and 'Worker environment'. The 'Application information' section has a text input field for 'Application name' with the value 'Exp2' and a note 'Maximum length of 100 characters.' Below this is a section for 'Application tags (optional)' with a plus icon. The footer of the console shows 'CloudShell', 'Feedback', and copyright information for Amazon Web Services, Inc. or its affiliates, along with links for 'Privacy', 'Terms', and 'Cookie preferences'.

Step 1
Configure environment

Step 2
Configure service access

Step 3 - optional
Set up networking, database, and tags

Step 4 - optional
Configure instance traffic and scaling

Step 5 - optional
Configure updates, monitoring, and logging

Step 6
Review

Configure environment [Info](#)

Environment tier [Info](#)
Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.

☒ **Web server environment**
Run a website, web application, or web API that serves HTTP requests. [Learn more](#)

☐ **Worker environment**
Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. [Learn more](#)

Application information [Info](#)

Application name
Exp2
Maximum length of 100 characters.

► Application tags (optional)

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

aws

Services

Search

[Alt+S]

N. Virginia

voclabs/user3398173=2022.girish.chougule@ves.ac.in @ 6573-114...

Managed platform

Platforms published and maintained by Amazon Elastic Beanstalk. [Learn more](#)

Custom platform

Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform

PHP

Platform branch

PHP 8.3 running on 64bit Amazon Linux 2023

Platform version

4.3.1 (Recommended)

Application code

Info

Sample application

Existing version

Application versions that you have uploaded.

Upload your code

Upload a source bundle from your computer or copy one from Amazon S3.

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Permissions

Trust relationships

Tags

Access Advisor

Revoke sessions

Permissions policies (2)

Info

You can attach up to 10 managed policies.

Filter by Type

All types

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

Permissions boundary (not set)

Generate policy based on CloudTrail events

You can generate a new policy based on the access activity for this role, then customize, create, and attach it to this role. AWS uses your CloudTrail events to identify the

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

Congratulations!

You have successfully created a pipeline that retrieved this source application from an Amazon S3 bucket and deployed it to three Amazon EC2 instances using AWS CodeDeploy.

For next steps, read the [AWS CodePipeline Documentation](#). Incoedge 2020

Conclusion:

Building and deploying an application using AWS CodeBuild, CodePipeline, and CodeDeploy demonstrates the power of automated CI/CD in the cloud. AWS CodeBuild compiles code, runs tests, and prepares software packages, while CodePipeline automates the release process, ensuring faster and consistent deployments. Deploying to S3 or SEBS enables scalable hosting of static and serverless applications, and CodeDeploy 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.