

Name : Nidhi Rijhwani

Subject : Adv Devops Exp No 02

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.

Steps:

1. **Create a role in an IAM.**
2. **Add EC2 for a service or use case.**
3. **Give name to the role.**
4. **Required policies (permissions) to be added while creating IAM users.**
5. **IAM Role Created.**
6. **Go to the Elastic beanstalk and create an application. Give the appropriate name for the application.**
7. **Select the platform as PHP.**
8. **In an ec2 instance profile, select the created IAM role.**
9. **Environment is launched successfully.**
10. **Go to the CodePipeline and select the source as GitHub (version 1).**
11. **After skipping the build stage, AWS Elastic beanstalk is to be selected in the Deploy Provider. Select your recently created application name and environment name.**
12. **Go to the elastic beanstalk environment and click on domain.**

Outputs:

aws

Services

Search

[Alt+S]

Global

voclabs/user3402666=d2022.niddhi.rijhwani@ves.ac.in @ 7369-08...

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

User groups

Users

Roles

Policies

Identity providers

Account settings

IAM > Policies > AWSElasticBeanstalkMulticontainerDocker

AWSElasticBeanstalkMulticontainerDocker

Info

Provide the instances in your multicontainer Docker environment access to use the Amazon EC2 Container Service to manage container deployment tasks.

Policy details

Type	Creation time	Edited time	ARN
AWS managed	February 09, 2016, 04:45 (UTC+05:30)	March 24, 2023, 03:34 (UTC+05:30)	arn:aws:iam:aws:policy/AWSElasticBeanstalkMulticontainerDocker

Permissions

Entities attached

Policy versions (5)

Last Accessed

[Alt+S]

Global

voclabs/user3402666=d2022.niddhi.rijhwani@ves.ac.in @ 7369-08...

10

Step 2: Add permissions

Edit

Permissions policy summary

Policy name	Type	Attached as
AWSElasticBeanstalkMulticontainerDocker	AWS managed	Permissions policy
AWSElasticBeanstalkRoleWorkerTier	AWS managed	Permissions policy
AWSElasticBeanstalkWebTier	AWS managed	Permissions policy

Step 3: Add tags

Add tags - optional

Info

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel

Previous

Create role

aws

Services

Search

[Alt+S]

Global

SR

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

User groups

Users

Roles

Role Nidhi_Iam created.

View role

Roles (8)

Info

Delete

Create role

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.

Search

< 1 >

Role name	Trusted entities
<input type="checkbox"/> codebuild-SAACHI-service-role	AWS Service: codebuild
<input type="checkbox"/> mypythonFunction-role-bdvj8n4	AWS Service: lambda
<input type="checkbox"/> Nidhi_Iam	AWS Service: ec2

aws

Services

Search

[Alt+S]

Stockholm

Compute

Amazon Elastic Beanstalk

End-to-end web application management.

Amazon Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar

Get started

Easily deploy your web application in minutes.

Create application

Configure environment

Configure environment

Info

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,

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

aws

Services

Search

[Alt+S]

Stockholm

Elastic Beanstalk

Applications

Environments

Change history

Application: Saachi_Web

Application versions

Saved configurations

Environment: SaachiWeb-env

Go to environment

Configuration

Environment successfully launched.

Elastic Beanstalk > Environments > SaachiWeb-env

Nidhi Web-env

Info

Actions

Upload and deploy

Environment overview

Health

Warning

Domain

SaachiWeb-env.eba-megrwftp.eu-north-1.elasticbeanstalk.com

Environment ID

e-atmqxcaprm

Application name

Saachi_Web

aws

Services

Search

[Alt+S]

Stockholm

SR

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1

Choose pipeline settings

Step 1 of 5

Pipeline settings

Pipeline name

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

Nidhi-pipeline

No more than 100 characters

Pipeline type

You can no longer create V1 pipelines through the console. We recommend you use the V2 pipeline type with improved release safety, pipeline triggers, parameterized pipelines, and a new billing model.

Execution mode

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.