

EXPERIMENT:02

Aim: Deploy a Sample Application on Elastic Beanstalk using AWS CodePipeline and AWS CodeDeploy.

Theory:

AWS Elastic Beanstalk allows for the rapid deployment and management of applications in the AWS Cloud, simplifying the process by eliminating the need to understand the underlying infrastructure. AWS consists of over a hundred services, each offering specific functionalities. While the range of services provides extensive flexibility in managing your AWS infrastructure, determining which services to use and how to provision them can be complex. Elastic Beanstalk alleviates this complexity without sacrificing control or customization. By simply uploading your application, Elastic Beanstalk takes care of the details related to capacity provisioning, load balancing, scaling, and application health monitoring.

Elastic Beanstalk supports a variety of programming languages, including Go, Java, .NET, Node.js, PHP, Python, and Ruby. It also accommodates Docker containers, allowing you to define your programming language and application dependencies that might not be supported by other Elastic Beanstalk platforms. Upon deploying your application, Elastic Beanstalk automatically provisions the necessary AWS resources, such as Amazon EC2 instances, and configures the selected platform to run your application.

Interaction with Elastic Beanstalk can be done through multiple interfaces: the Elastic Beanstalk console, the AWS Command Line Interface (AWS CLI), or **eb**, a specialized high-level CLI tool designed specifically for Elastic Beanstalk.

Most deployment tasks, like scaling your fleet of Amazon EC2 instances or monitoring application performance, can be easily managed via the Elastic Beanstalk web interface (console). To get started with Elastic Beanstalk, you create an application, upload an application version (e.g., a Java **.war** file) as a source bundle, and provide necessary configuration details. Elastic Beanstalk will then automatically set up the environment and configure all required AWS resources to run your application. Once the environment is active, you can manage it and deploy new application versions as needed. The following workflow diagram outlines the process of using Elastic Beanstalk.

After your application is up and running, you can access information related to the application, such as metrics, events, and environment status, through the Elastic Beanstalk console, APIs, or the Command Line Interfaces, including the unified AWS CLI.

Implementation:

Deploying basic web page on Elastic Beanstalk

aws Services Search [Alt+S]

Configure updates, monitoring, and logging

Step 6
Review

Application name

MyWebApp

Maximum length of 100 characters.

► Application tags (optional)

Environment information Info

Choose the name, subdomain and description for your environment. These cannot be changed later.

Environment name

MyWebApp-dev

Must be from 4 to 40 characters in length. The name can contain only letters, numbers, and hyphens. It can't start or end with a hyphen. This name must be unique within a region in your account.

Domain

Leave blank for autogenerated value .eu-north-1.elasticbeanstalk.com Check availability

Environment description

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

aws Services Search [Alt+S]

Platform Info

Platform type

☒ 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

Python

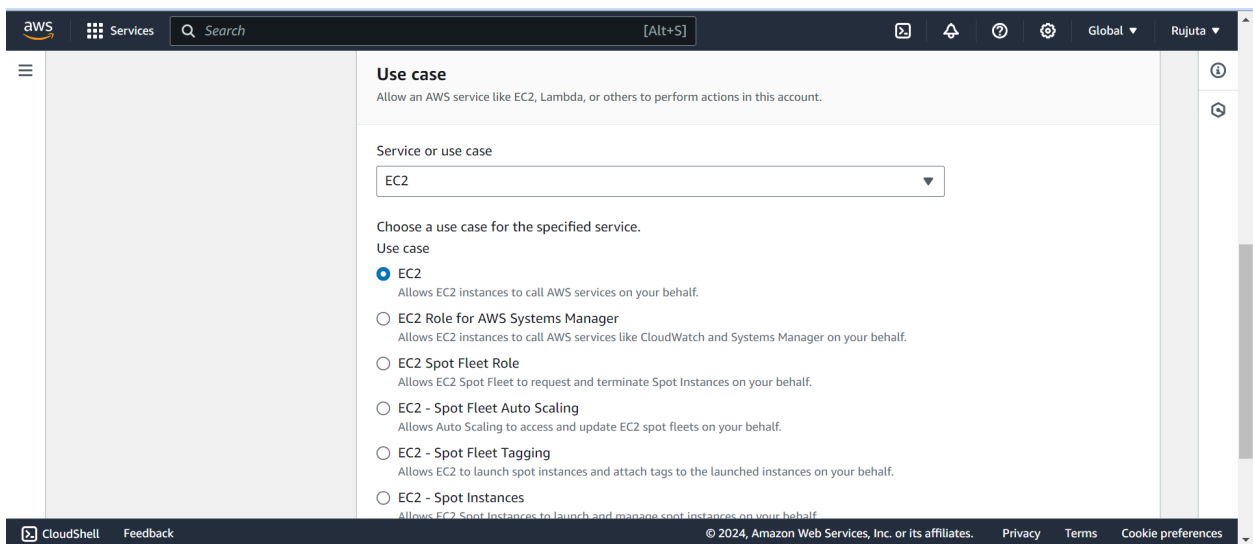
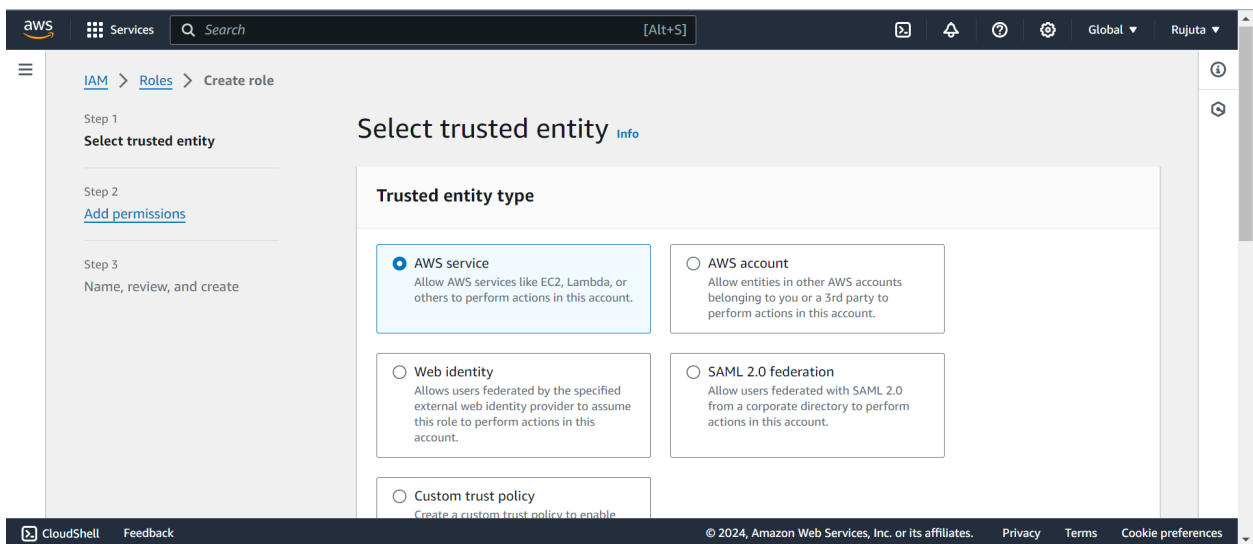
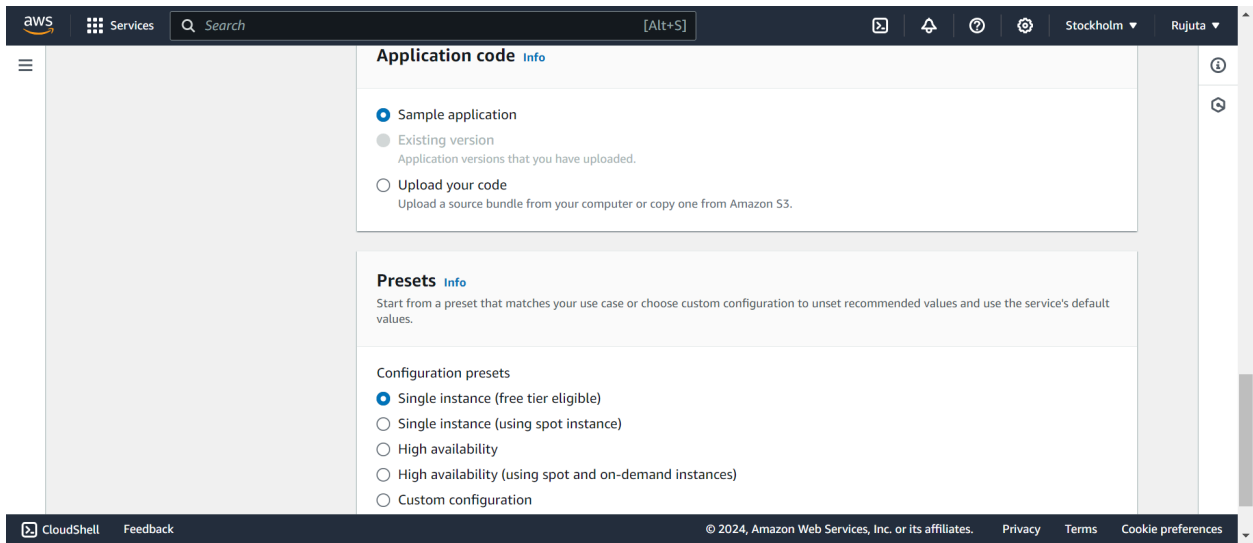
Platform branch

Python 3.11 running on 64bit Amazon Linux 2023

Platform version

4.1.3 (Recommended)

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



aws

Services

Search

[Alt+S]

Global

Rujuta

Step 2

Add permissions

Step 3

Name, review, and create

Permissions policies (3/946)

info

Choose one or more policies to attach to your new role.

Filter by Type

Beanstalk

All types

14 matches

< 1 >

Policy name

Type

Description

☐ AdministratorAccess-AWSElasticBeanstalk

AWS managed

Grants acc

☐ AWSElasticBeanstalkCustomPlatformforEC2Role

AWS managed

Provide th

☐ AWSElasticBeanstalkEnhancedHealth

AWS managed

AWS Elast

☐ AWSElasticBeanstalkManagedUpdatesCustomerRole...

AWS managed

This policy

☒ AWSElasticBeanstalkMulticontainerDocker

AWS managed

Provide th

☐ AWSElasticBeanstalkReadOnly

AWS managed

Grants rea

☐ AWSFleatirBeanstalkRoleCore

AWS managed

AWSFlecti

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

aws

Services

Search

[Alt+S]

Global

Rujuta

IAM > Roles > Create role

Step 1

Select trusted entity

Step 2

Add permissions

Step 3

Name, review, and create

Name, review, and create

Role details

Role name

Enter a meaningful name to identify this role.

aws-elasticbeanstalk-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: _+=, @-/_[]:~#%&*()-"

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

aws

Services

Search

[Alt+S]

Global

Rujuta

Role aws-elasticbeanstalk-role created.

View role

Roles (4)

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.

Create role

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

aws

Services

Search

[Alt+S]

Stockholm

Rujuta

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

Configure service access

Service access

IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. [Learn more](#)

Service role

☐ Create and use new service role

☒ Use an existing service role

Existing service roles

Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.

aws-elasticbeanstalk-service-role

EC2 key pair

Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

Choose a key pair

Step 6
Review

Choose a key pair

EC2 instance profile

Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

aws-elasticbeanstalk-role

View permission details

Cancel

Skip to review

Previous

Next

aws

Services

Search

[Alt+S]

Stockholm

Rujuta

Environment successfully launched.

Elastic Beanstalk > Environments > MywebApp-dev

MywebApp-dev

Actions

Upload and deploy

Congratulations

Your first AWS Elastic Beanstalk Python Application is now running on your own dedicated environment in the AWS Cloud

This environment is launched with Elastic Beanstalk Python Platform

What's Next?

- [AWS Elastic Beanstalk overview](#)
- [AWS Elastic Beanstalk concepts](#)
- [Deploy a Django Application to AWS Elastic Beanstalk](#)
- [Deploy a Flask Application to AWS Elastic Beanstalk](#)
- [Customizing and Configuring a Python Container](#)
- [Working with Logs](#)

Code Deployment using Codepipeline:

The screenshot shows the AWS CodePipeline console. On the left, a sidebar lists 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. The main area is titled 'Review' and 'Step 1: Configure environment'. It displays 'Environment information' in a table:

Environment tier	Application name
Web server environment	MywebApp
Environment name	Application code
MywebApp-dev	Sample application
Platform	
arn:aws:elasticbeanstalk:eu-north-1::platform/Python 3.11 running on 64bit Amazon Linux 2023/4.1.3	

At the bottom, there is a footer with 'CloudShell', 'Feedback', and copyright information for Amazon Web Services, Inc. or its affiliates.

The screenshot shows the AWS CodePipeline console. On the left, a sidebar lists steps: Step 1 (Choose pipeline settings), Step 2 (Add source stage), Step 3 (Add build stage), Step 4 (Add deploy stage), and Step 5 (Review). The main area is titled 'Add source stage' and 'Step 2 of 5'. It displays 'Source' information:

Source provider: GitHub (Version 1)

Grant AWS CodePipeline access to your GitHub repository. This allows AWS CodePipeline to upload commits from GitHub to your pipeline.

Connected

You have successfully configured the action with the provider.

The GitHub (Version 1) action is not recommended

At the bottom, there is a footer with 'CloudShell', 'Feedback', and copyright information for Amazon Web Services, Inc. or its affiliates.

The screenshot shows the AWS CloudFormation console. On the left, a sidebar lists 'List' and 'Resources'. The main area is titled 'Application Composer' and shows a 'Canvas' view of a stack named 'aws-e-wuixywicib-stack.yaml'. The canvas displays a diagram of resources and their dependencies:

- Standard Component: **AWSEBAutoScalingLaunchConfiguration** (AWSEBLaunchConfiguration, AWSEBAutoScalingGroup, AWSEBAutoScalingGroup)
- Standard Component: **AWSEBInstanceLaunchWaitHandle** (AWSEBInstanceLaunchWaitHandle)
- Standard Component: **AWSEBInstanceLaunchWaitCondition** (AWSEBInstanceLaunchWaitCondition)
- Standard Component: **AWSEBInstanceLaunchWaitCondition** (AWSEBInstanceLaunchWaitCondition)

At the bottom, there is a footer with 'CloudShell', 'Feedback', and copyright information for Amazon Web Services, Inc. or its affiliates.

aws

Services

cloud formation

Stockholm

Rujuta

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1
Choose pipeline settings

Step 2
Add source stage

Step 3
Add build stage

Step 4
Add deploy stage

Step 5
Review

Add deploy stage

Step 4 of 5

You cannot skip this stage

Pipelines must have at least two stages. Your second stage must be either a build or deployment stage. Choose a provider for either the build stage or deployment stage.

Deploy

Deploy provider

Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.

AWS Elastic Beanstalk

Region

Europe (Stockholm)

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

aws

Services

cloud formation

Stockholm

Rujuta

Input artifacts

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

No more than 100 characters

Application name

Choose an application that you have already created in the AWS Elastic Beanstalk console. Or create an application in the AWS Elastic Beanstalk console and then return to this task.

MyWebApp

Environment name

environment in the AWS Elastic

MywebApp-dev

MywebApp-dev

☐ Configure automatic rollback on stage failure

Cancel

Previous

Next

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

aws

Services

cloud formation

Stockholm

Rujuta

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1
Choose pipeline settings

Step 2
Add source stage

Step 3
Add build stage

Step 4
Add deploy stage

Step 5
Review

Review

Step 5 of 5

Step 1: Choose pipeline settings

Pipeline settings

Pipeline name

rujuta_pipeline

Pipeline type

V2

Execution mode

QUEUED

Artifact location

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

aws

Services

cloud formation

Stockholm

Rujuta

Success

Congratulations! The pipeline rujuta_pipeline has been created.

Create a notification rule for this pipeline

Developer Tools

CodePipeline

Pipelines

rujuta_pipeline

rujuta pipeline

Notify

Edit

Stop execution

Clone pipeline

Release change

aws

Services

cloud formation

Stockholm

Rujuta

Succeeded - 1 minute ago

6463aacc Add files via upload

View details

Deploy Succeeded

Start rollback

Deploy

AWS Elastic Beanstalk

Succeeded - Just now

View details

CloudShell

Feedback

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

Source Succeeded

Source

GitHub (Version 1)

Succeeded - 3 minutes ago

6463aacc Add files via upload

View details

Infosys

Company Details Services Media Contact Us

Welcome to Infosys

Since 1992nd

"~Innovation distinguishes between a leader and a follower."

