RUJUTA MEDHI **D15A-28**

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.

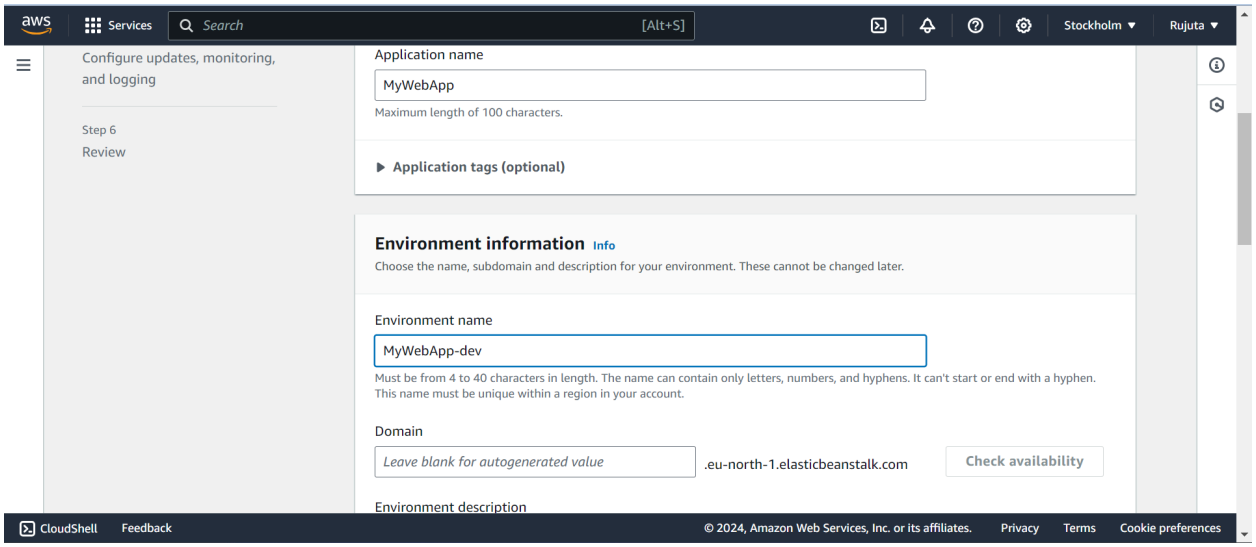
Interaction with Elastic Beanstalk can be done through multiple interfaces: the Elastic Beanstalk console, the AWS Command Line Interface (AWS CLI), or eb, a specialized high-level CLI tool designed specifically for Elastic Beanstalk.

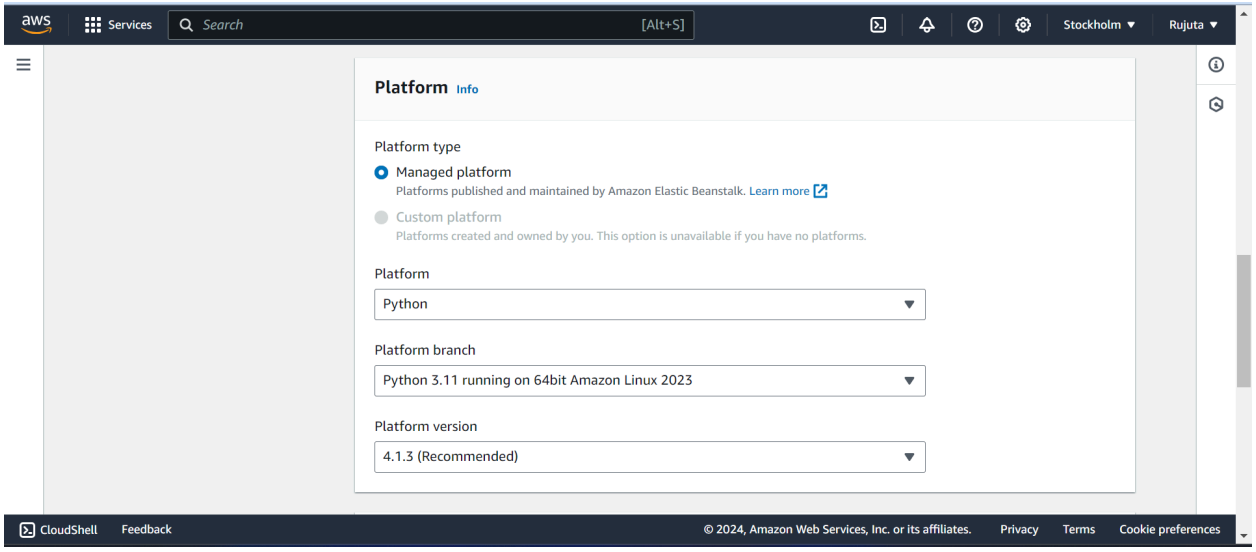
Most deployment tasks, like scaling your fleet of Amazon EC2 instances or monitoring application performance, can be easily managed via the Elastic Beanstalk web interface (console). To get started with Elastic Beanstalk, you create an application, upload an application version (e.g., a Java .war file) 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.

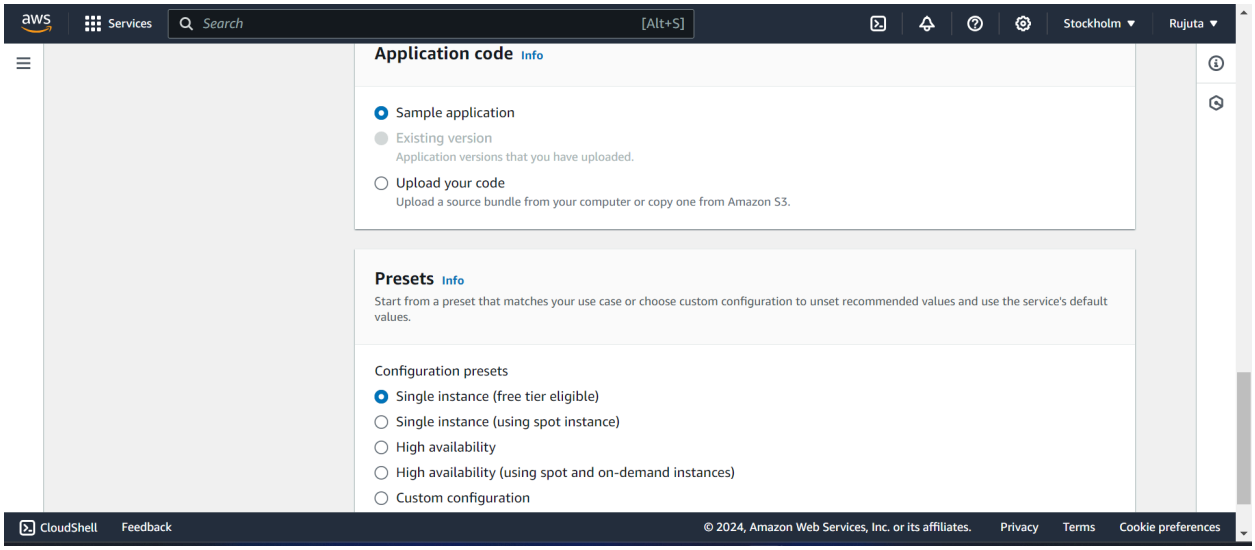
After your application is up and running, you can access information related to the application, such as metrics, events, and environment status, through the Elastic Beanstalk console, APIs, or the Command Line Interfaces, including the unified AWS CLI.

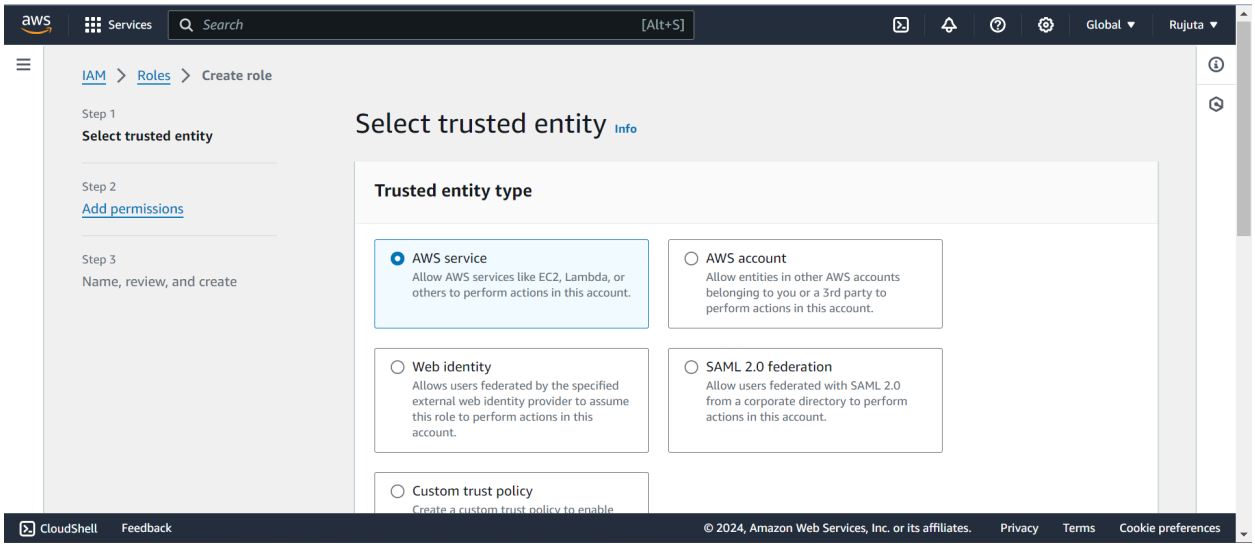
Implementation:

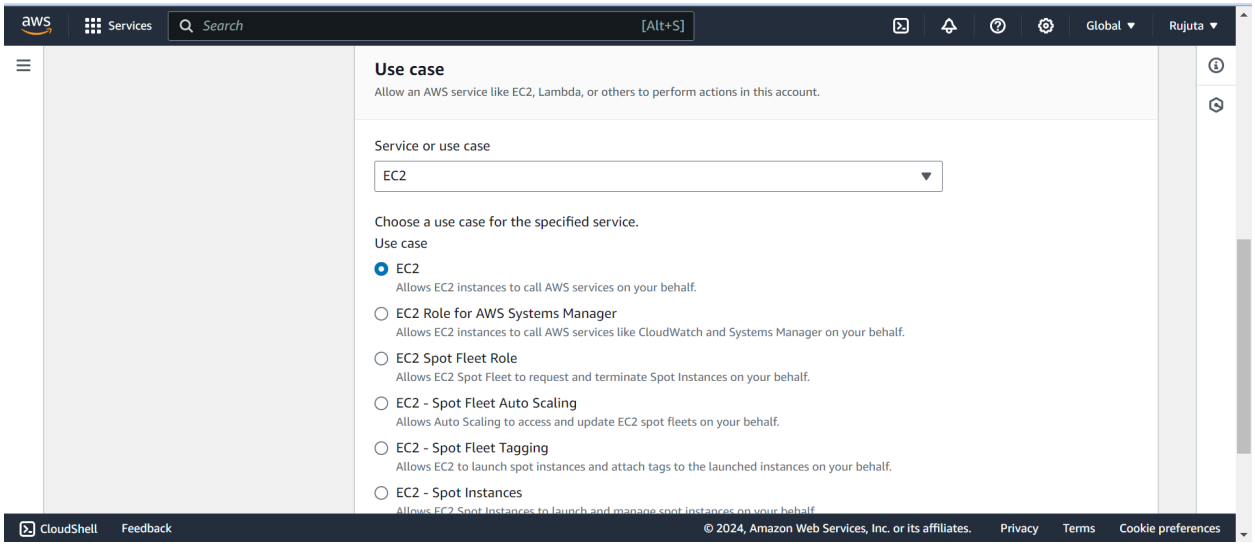
Deploying basic web page on Elastic Beanstalk

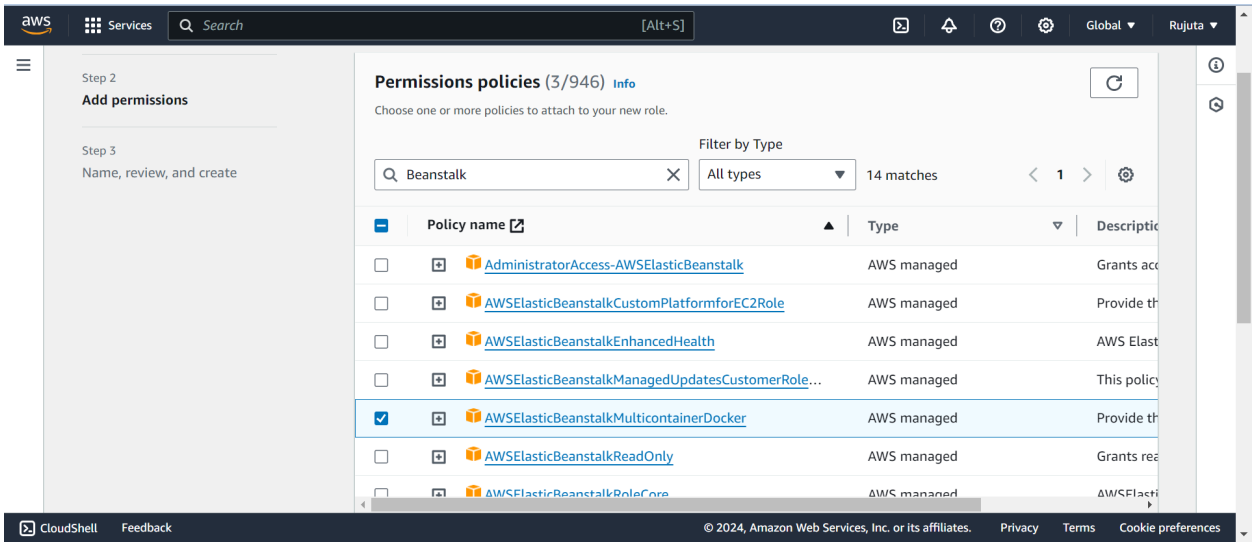


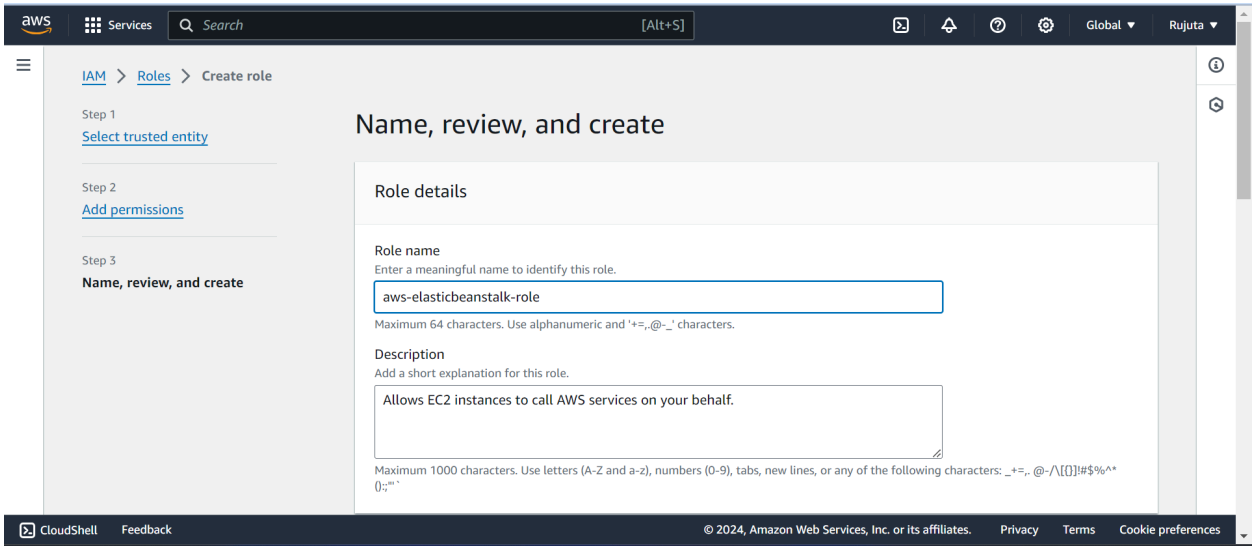


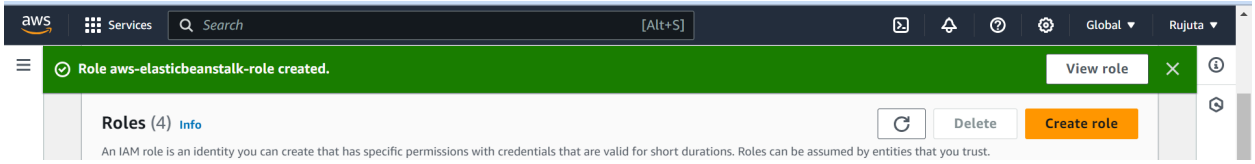

Services
Search
[Alt+S]
Stockholm
Rujuta
Application code Info
Sample application
Existing version
Application versions that you have uploaded.
Upload your code
Upload a source bundle from your computer or copy one from Amazon S3.
Presets Info
Start from a preset that matches your use case or choose custom configuration to unset recommended values and use the service's default values.
Configuration presets
Single instance (free tier eligible)
Single instance (using spot instance)
High availability
High availability (using spot and on-demand instances)
Custom configuration
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

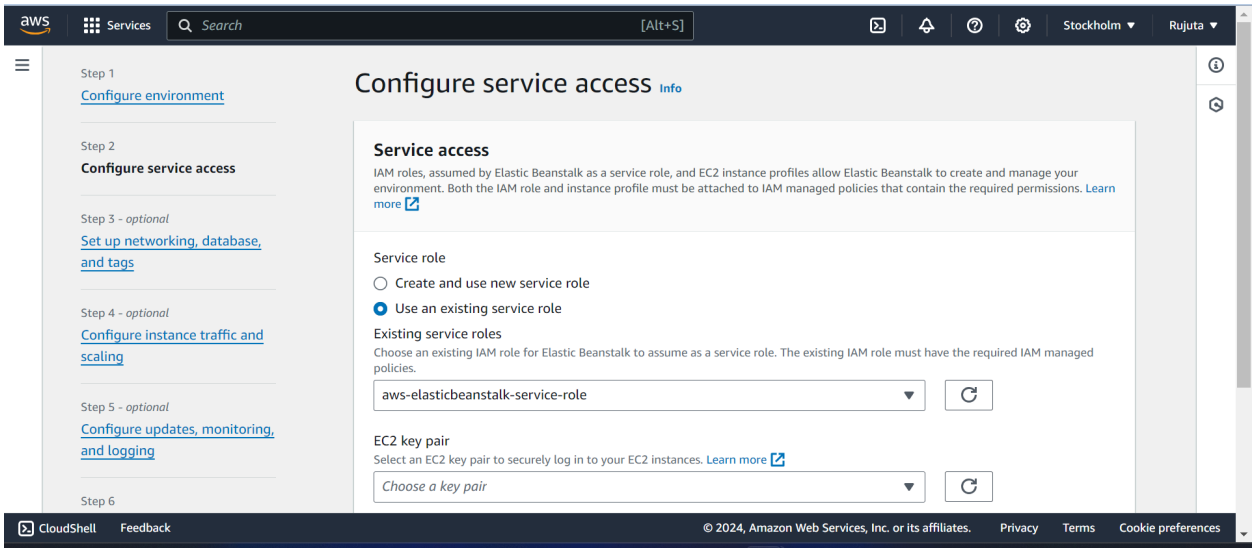

Services
Search
[Alt+S]
Global
Rujuta
IAM
Roles
Create role
Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create
Select trusted entity Info
Trusted entity type
AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.
AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
Web identity
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
SAML 2.0 federation
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
Custom trust policy
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

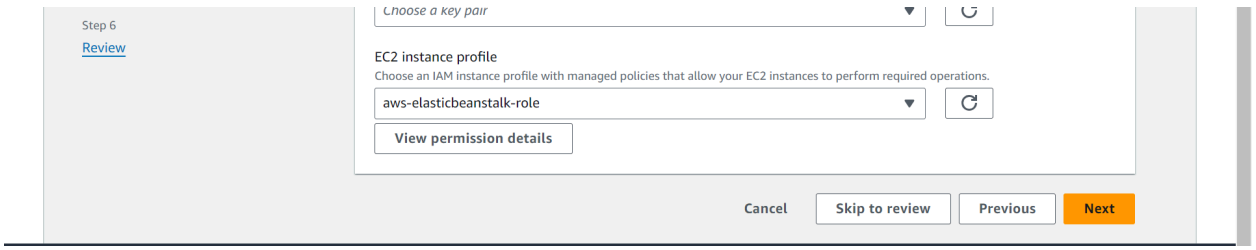

Services
Search
[Alt+S]
Global
Rujuta
Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.
Service or use case
EC2
Choose a use case for the specified service.
Use case
EC2
Allows EC2 instances to call AWS services on your behalf.
EC2 Role for AWS Systems Manager
Allows EC2 instances to call AWS services like CloudWatch and Systems Manager on your behalf.
EC2 Spot Fleet Role
Allows EC2 Spot Fleet to request and terminate Spot Instances on your behalf.
EC2 - Spot Fleet Auto Scaling
Allows Auto Scaling to access and update EC2 spot fleets on your behalf.
EC2 - Spot Fleet Tagging
Allows EC2 to launch spot instances and attach tags to the launched instances on your behalf.
EC2 - Spot Instances
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

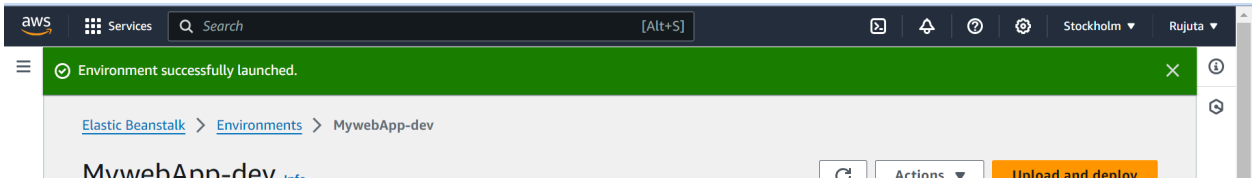

Services
Search
[Alt+S]
Global
Rujuta
Step 2
Add permissions
Step 3
Name, review, and create
Permissions policies (3/946) Info
Choose one or more policies to attach to your new role.
Filter by Type
Beanstalk
All types
14 matches
1
Policy name
Type
AdministratorAccess-AWSElasticBeanstalk
AWS managed
AWSElasticBeanstalkCustomPlatformforEC2Role
AWS managed
AWSElasticBeanstalkEnhancedHealth
AWS managed
AWSElasticBeanstalkManagedUpdatesCustomerRole...
AWS managed
AWSElasticBeanstalkMulticontainerDocker
AWS managed
AWSElasticBeanstalkReadOnly
AWS managed
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


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


Services
Search
[Alt+S]
Global
Rujuta
Role aws-elasticbeanstalk-role created.
View role
Roles (4) 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.







Congratulations

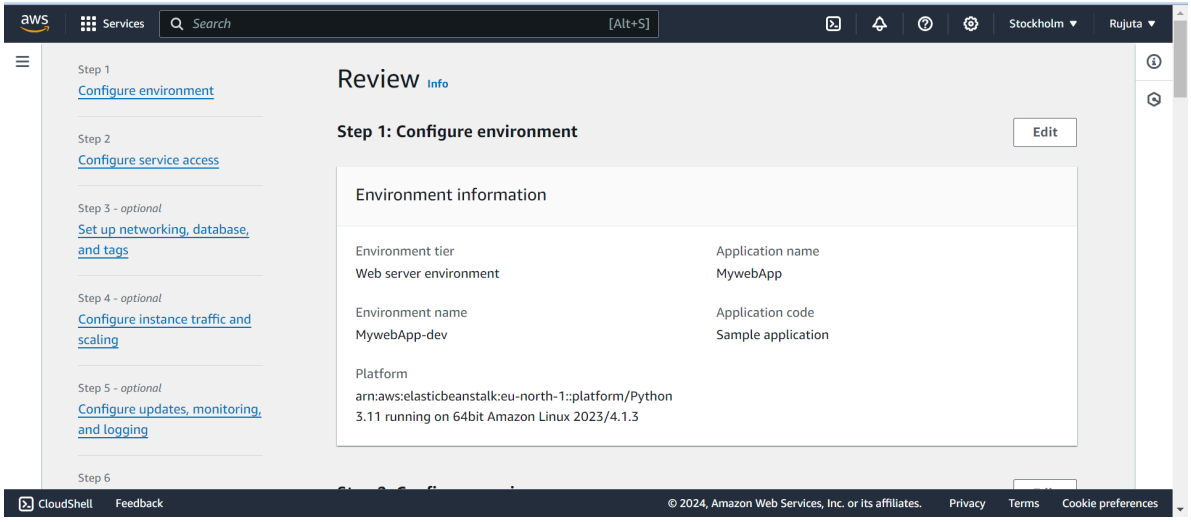
Your first AWS Elastic Beanstalk Python Application is now running on your own dedicated environment in the AWS Cloud

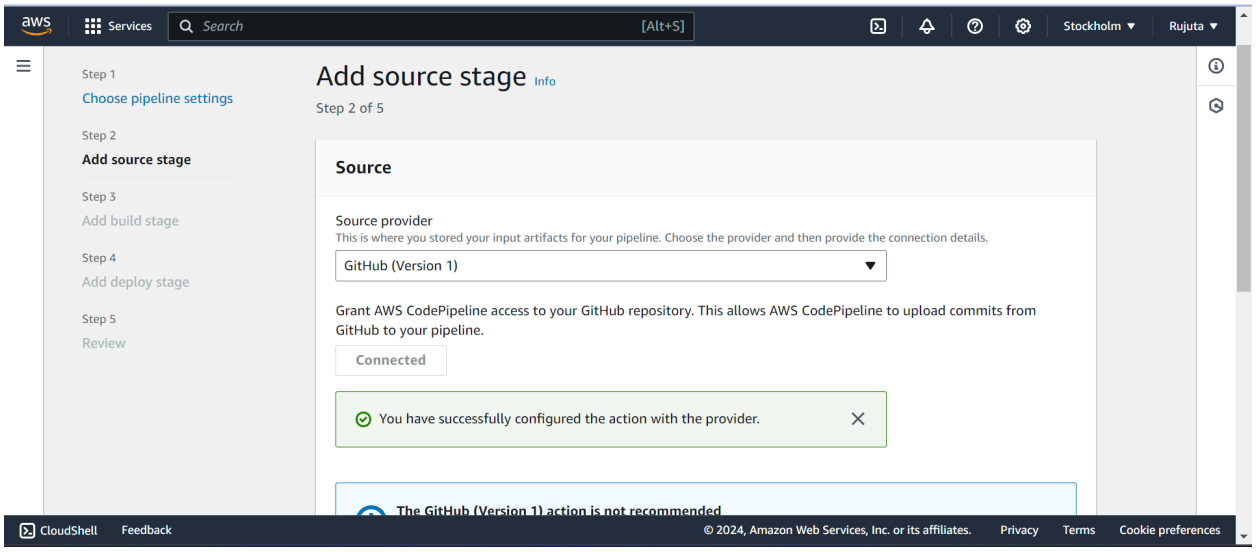
This environment is launched with Elastic Beanstalk Python Platform

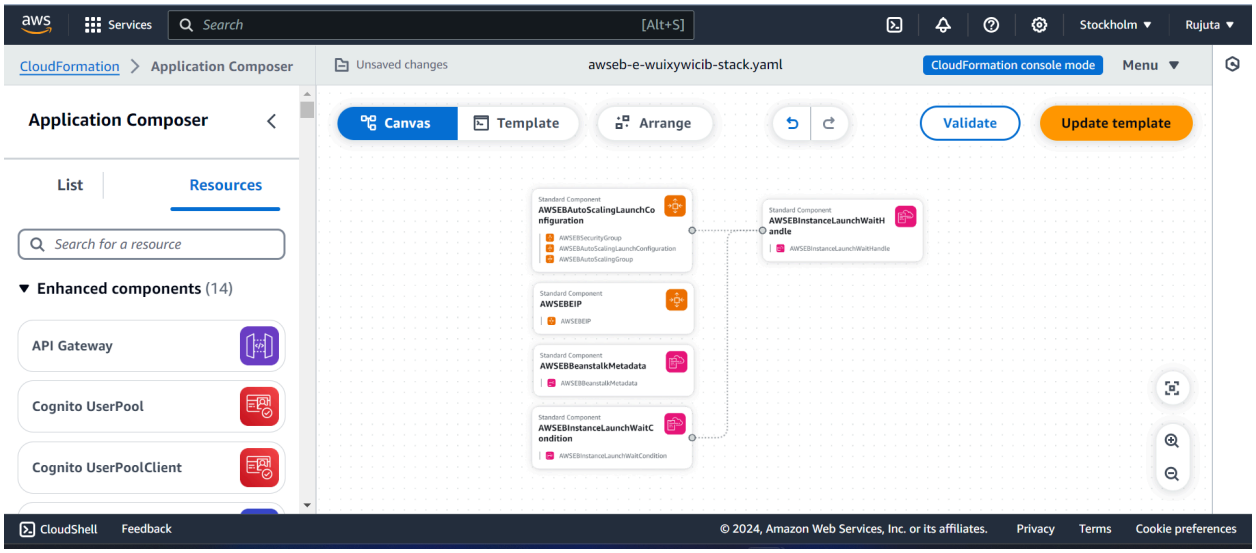
What's Next?

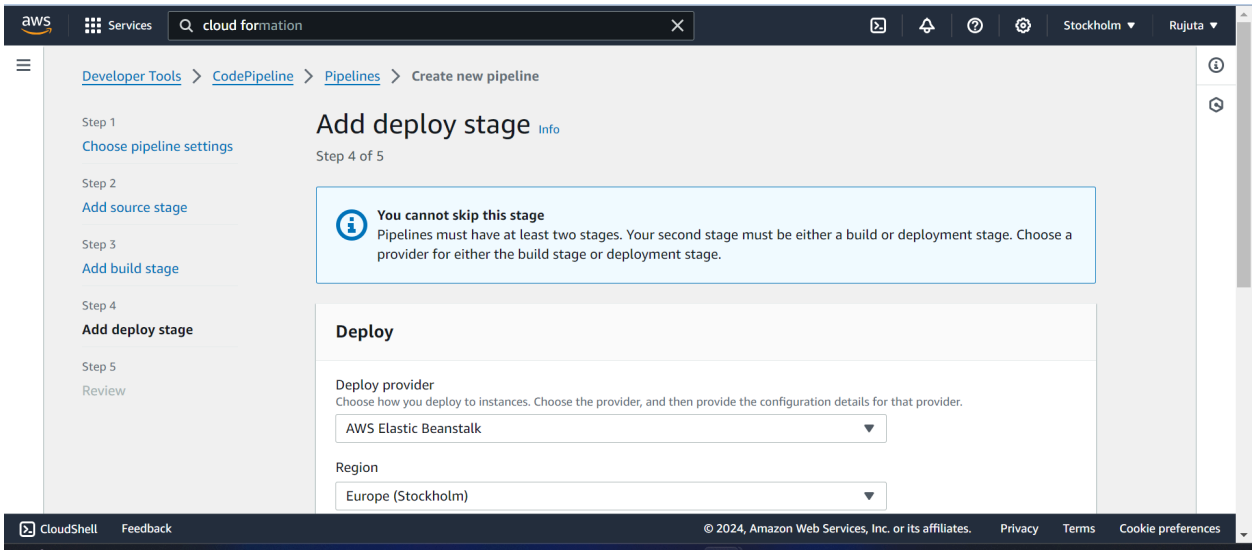
- AWS Elastic Beanstalk overview
- AWS Elastic Beanstalk concepts
- Deploy a Django Application to AWS Elastic Beanstalk
- Deploy a Flask Application to AWS Elastic Beanstalk
- Customizing and Configuring a Python Container
- Working with Logs

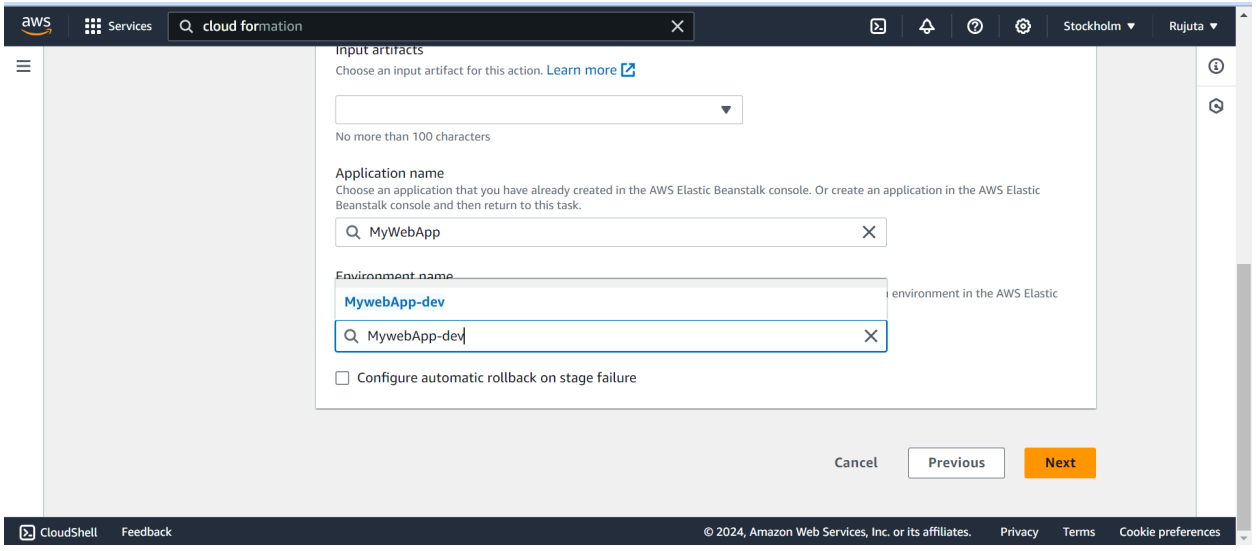
Code Deployment using Codepipeline:

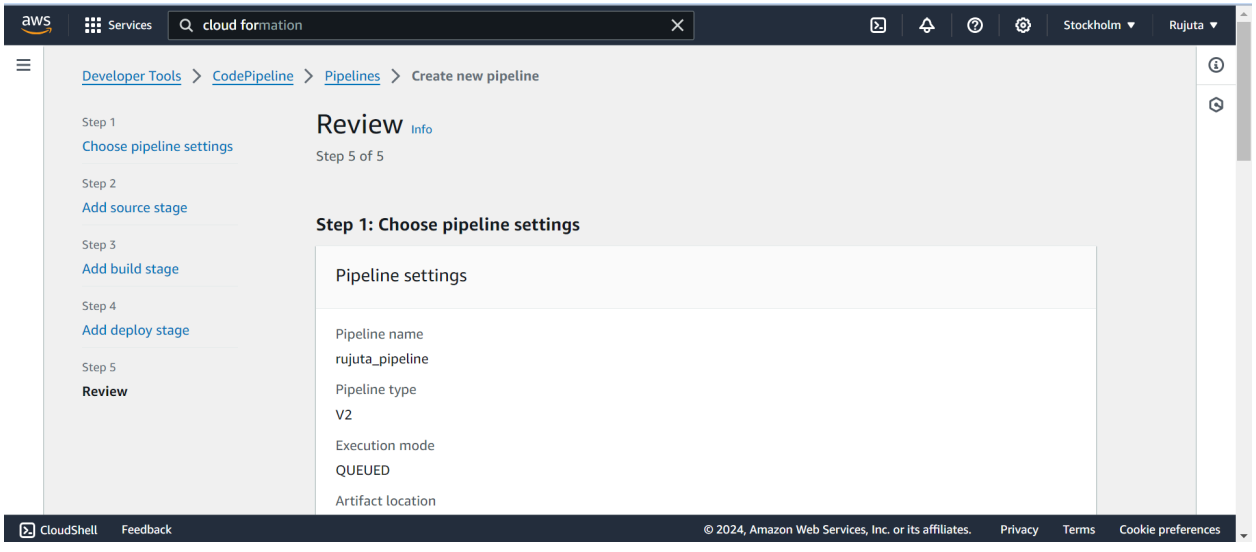


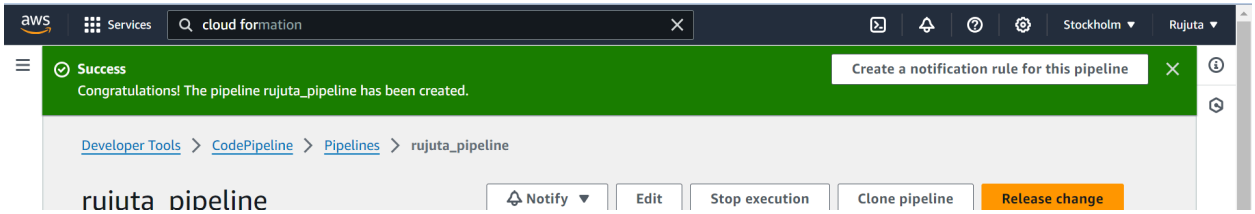


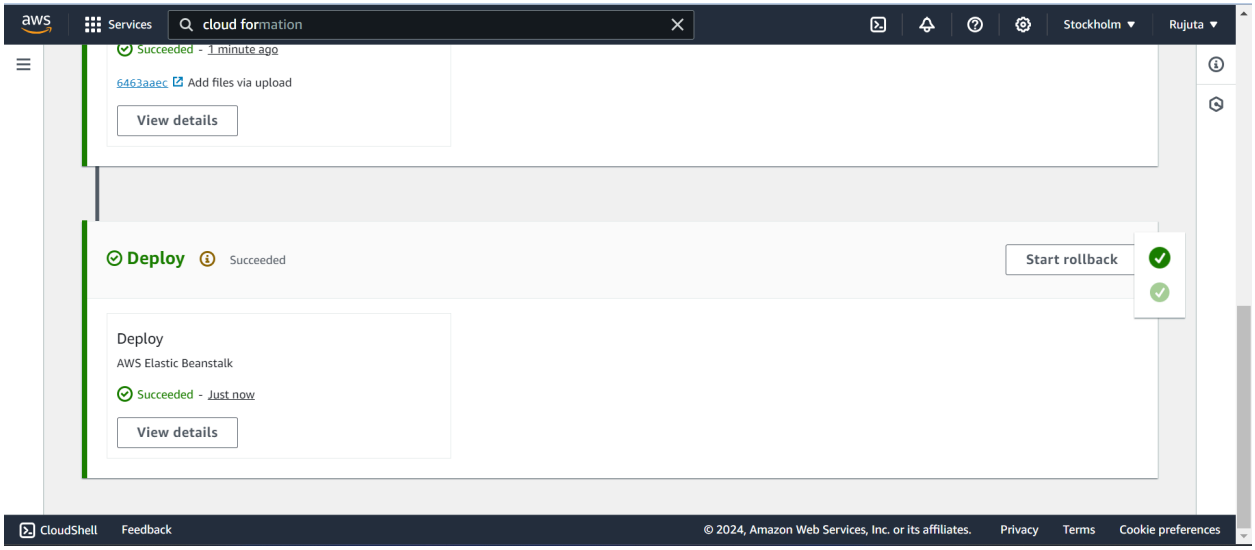


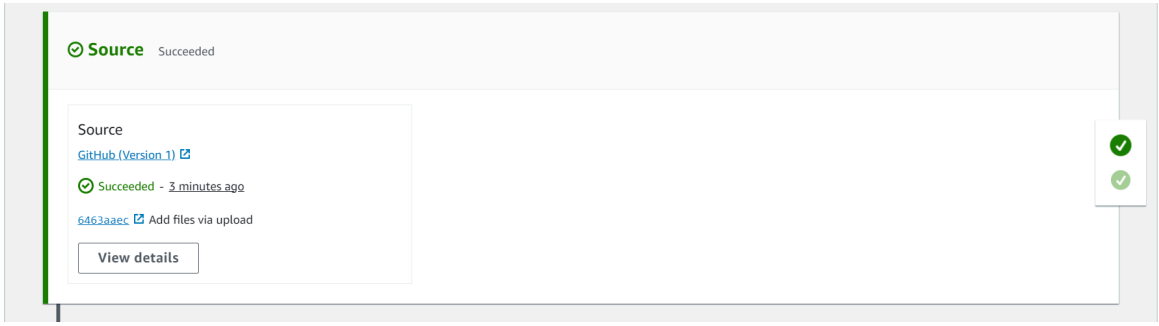

aws
Services
cloud formation
Stockholm
Rujuta
Developer Tools > CodePipeline > Pipelines > Create new pipeline
Step 1
Choose pipeline settings
Step 2
Add source stage
Step 3
Add build stage
Step 4
Add deploy stage
Step 5
Review
Add deploy stage Info
Step 4 of 5
You cannot skip this stage
Pipelines must 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

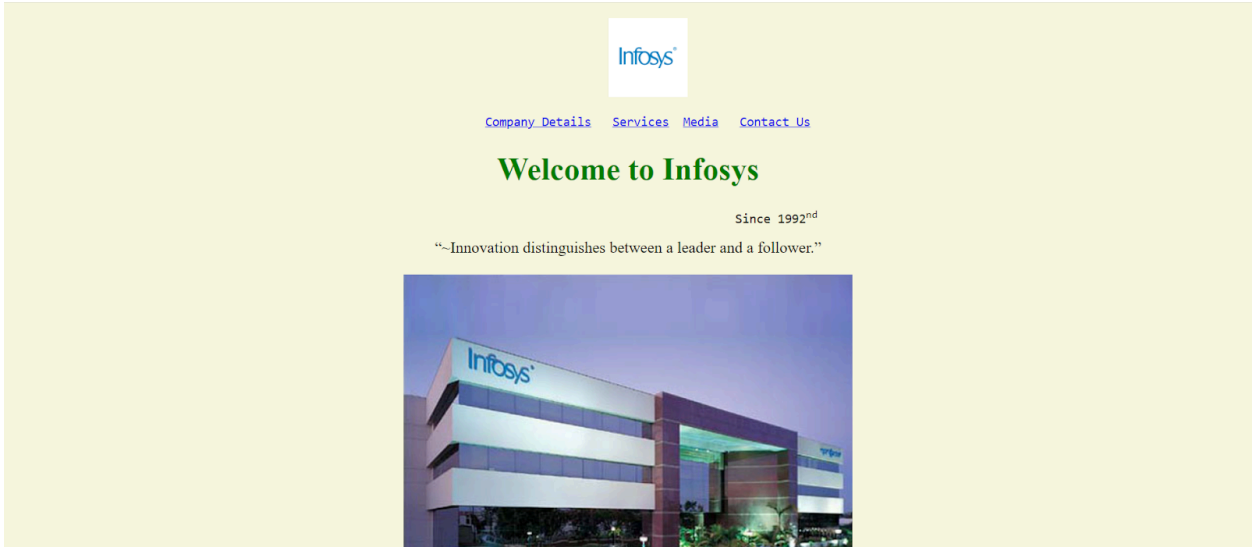

aws
Services
cloud formation
Stockholm
Rujuta
Input artifacts
Choose an input artifact for this action. Learn more
No more than 100 characters
Application name
Choose an application that you have already created in the AWS Elastic Beanstalk console. Or create an application in the AWS Elastic Beanstalk console and then return to this task.
MyWebApp
Environment name
MywebApp-dev
MywebApp-dev
environment in the AWS Elastic
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 Info
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
6463aaec 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
6463aaec 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."
Infosys