

Experiment No: 5

Aim: To study and implement platform as a service using AWS Elastic Beanstalk service.

Theory:

• Amazon AWS Elastic Beanstalk (EBS)

⇒ EBS is a cloud computing service provided by AWS that simplifies the management, scaling of web applications as services.

• It abstracts away the complexities of the infrastructure management, allowing developers to focus on writing code while AWS handles provisioning, load balancing, auto-scaling and other operational tasks.

• Key Features include:

1) Easy development

⇒ Developers can simply upload their application code as EBS automatically handles the deployment process, including provisioning the necessary resources.

2) Auto-scaling

⇒ EBS can automatically scale the number of EC2 instances based on application load, ensuring that the application

3) Monitoring & Logging
 ⇒ EBS provides built-in monitoring and logging capabilities, allowing developers to monitor application health and troubleshoot issues easily.

Languages / Frameworks supported by EBS

1) AWS Elastic Beanstalk supports a variety of programming languages and frameworks providing flexibility for developers to deploy their application using their technology stack.

2) EBS provides support for Docker containers allowing developers to package their applications and dependencies into Docker containers allowing developers to package their applications.

3) Ruby applications - allows developers to develop web applications built with Ruby on Rails Framework.

4) Go applications - are supported by EBS, enabling developers to deploy web applications written in the Go applications written web application by Go Language.

• Elastic Load Balancing (ELB)

⇒ ELB is provided by AWS that automatically distribute incoming application traffic across multiple targets, such as EC2 instances or Lambda functions, to ensure optimum performance, availability, and fault tolerance of the applications.

- Features include:

1) Distribution of incoming traffic

⇒ ELB automatically distribute incoming application traffic across multiple targets, ensuring that each target receives a balanced load of requests.

2) Scalability

⇒ ELB can automatically scale its request handling capacity in response to changes in incoming traffic.

3) High Availability

⇒ ELB enhances the availability of the applications by automatically detecting unhealthy targets and rerouting traffic to health targets.

- Comparison between EC2 and EBS

⇒

EC2 (Elastic Compute Cloud) vs EBS (Elastic Block Store)

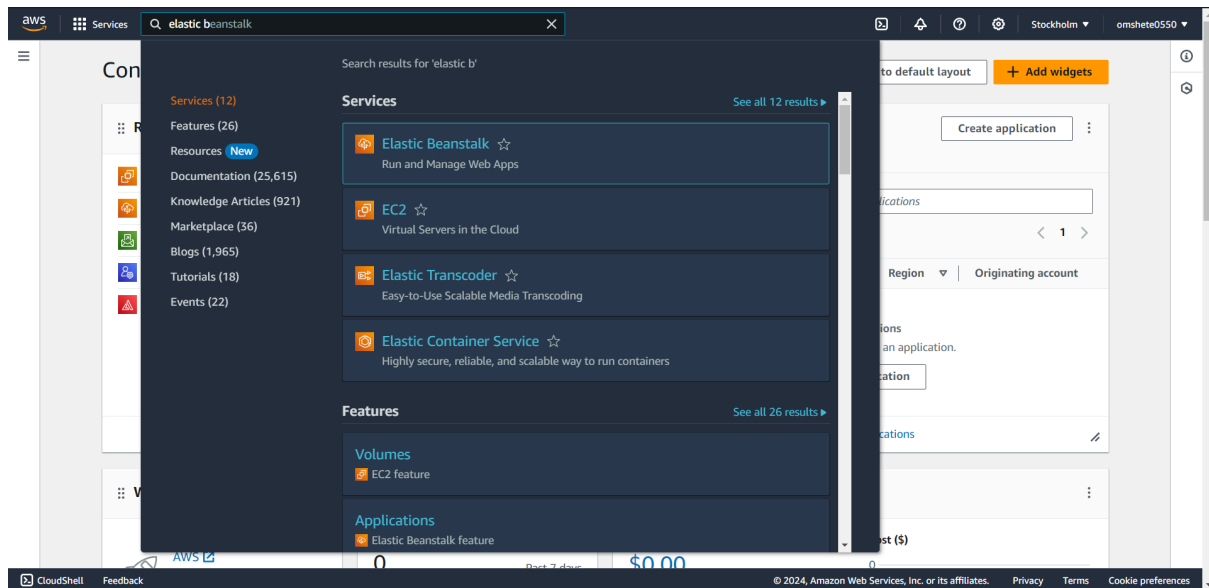
<p>1) Provides full control over the virtual computers, allowing users to configure their OS and applications according to their requirements.</p> <p>2) Requires manual configuration and management of infrastructure components.</p> <p>3) Users are responsible for manually scaling EC2 instances application load.</p> <p>4) Users pay for the EC2 instances, other resources provisioned manually with pricing based on the instance types.</p>	<p>1) Abstracts away the infrastructure management, automatically provisioning, load balancing, auto scaling and application health monitoring.</p> <p>2) Simplifies deployment and management tasks by automatically provisioning & configuration.</p> <p>3) Automatically handles scaling based on the application load.</p> <p>4) Users pay for the underlying AWS resources provisioned by EBS along with any additional services.</p>
--	--

(A)

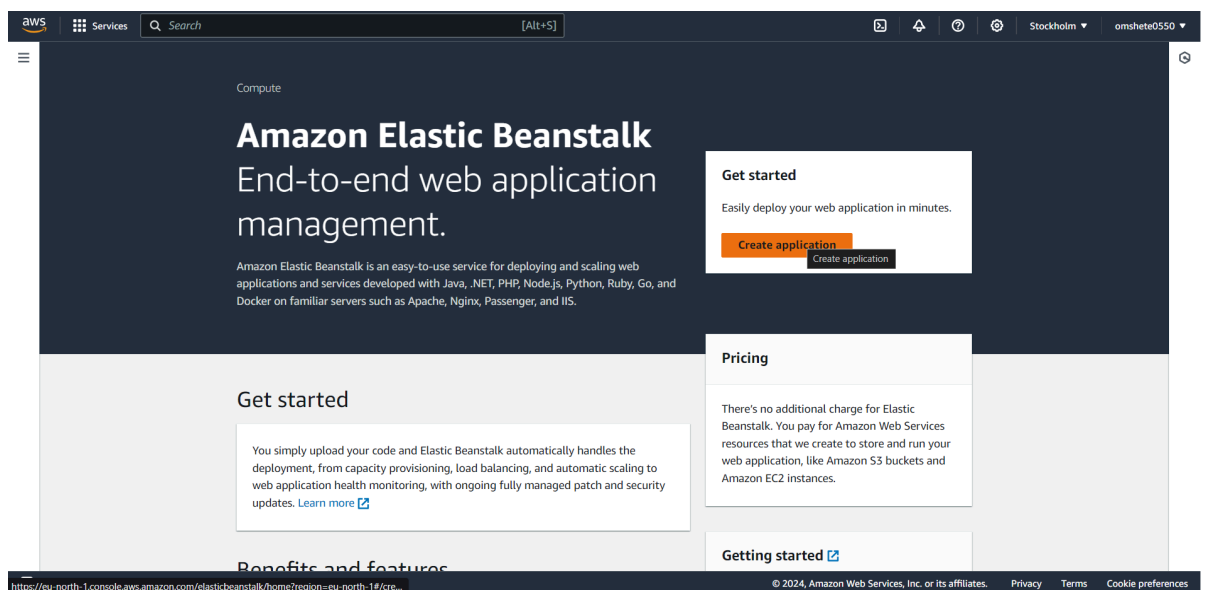
15/3/24

Output:

1. Login to the AWS console and go to Elastic Beanstalk



2. Click on Create Application



3. Write Application information: Name, Tag, Platform etc.

aws

Services

Search

[Alt+S]

Stockholm

omshete0550

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

Environment tier

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

Application name

MyApp

Maximum length of 100 characters.

Application tags (optional)

Apply up to 50 tags. You can use tags to group and filter resources. Tags must be unique within the resource and is case-sensitive. [Learn more](#)

Key

Q Name X

Use "My Server"

My Server

Q My Server X

Remove

Add new tag

You can add 49 more tags.

CloudShell

Feedback

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

Privacy

Terms

Cookie preferences

aws

Services

Search

[Alt+S]

Stockholm

omshete0550

Platform

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

Node.js

Platform branch

Node.js 20 running on 64bit Amazon Linux 2023

Platform version

6.1.2 (Recommended)

Application code

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

aws

Services

Search

[Alt+S]

Stockholm

omshete0550

4.2.2 (Recommended)

Application code

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

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

Cancel

Next

CloudShell

Feedback

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

Privacy

Terms

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 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](#)

mykeypair

EC2 instance profile

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

View permission details

Cancel

Skip to review

Previous

Next

CloudShell

Feedback

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

4. Enter submit to create the application

Activated100Percentage

Command timeout600Deployment policyAllAtOnceHealth thresholdOk

Ignore health checkfalseInstance replacementfalse

Platform software

LifecyclefalseLog streamingDeactivatedLogs retention7

Rotate logsDeactivatedUpdate levelminorX-Ray enabledDeactivated

Environment properties

Key	Value
GRADLE_HOME	/usr/local/gradle
M2	/usr/local/apache-maven/bin
M2_HOME	/usr/local/apache-maven

Cancel

Previous

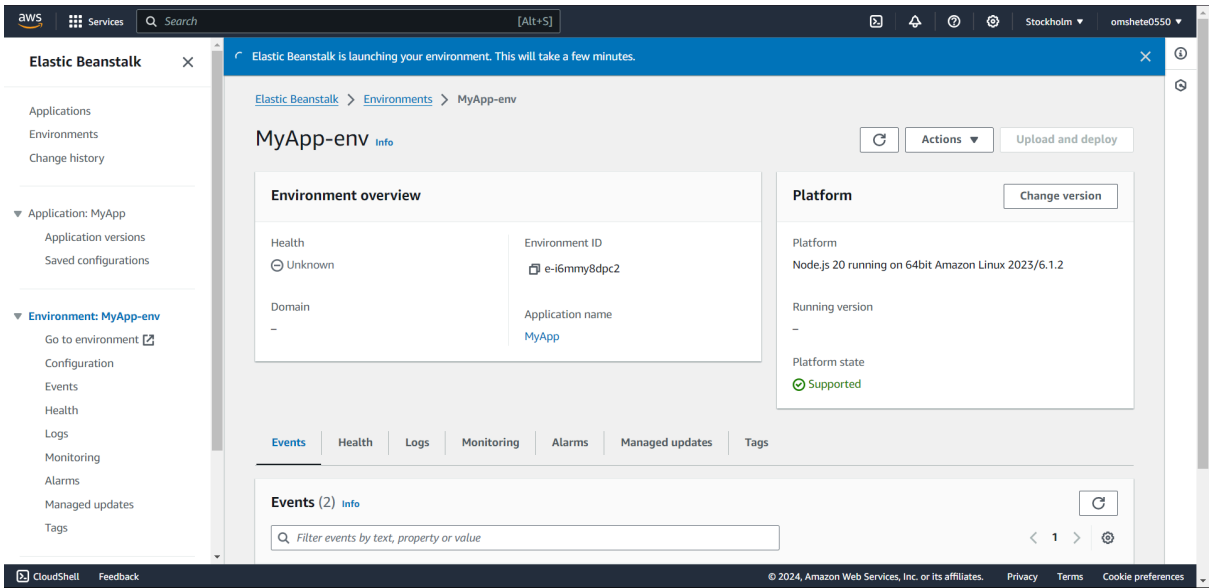
Submit

CloudShell

Feedback

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

This will take a few minutes.



5. Click on Environments -> Check Environment's health until it becomes 'OK'

