Department of Computer Engineering

Academic Term: JAN-MAY 2022

Class: ***BE COMPUTERS***

Subject Name: ***CLOUD COMPUTING LABORATORY***

Subject Code: CSL803

Practical No:	**10**
Title:	**AWS Beanstalk**
Date of Performance:	**09/02/22**
Date of Submission:	**11/04/2022**
Roll No:	**8626**
Name of the Student:	**Divita Phadakale**

Evaluation:

Sr. No	Rubric	Grade
1	**On time submission(2)**	
2	**Preparedness(2)**	
3	**Output(2)**	
4	**Post Lab Questions (4)**	
	TOTAL	

Signature of the Teacher:

AWS Elastic Beanstalk

Aim : Deployment of Java Web Application on Elastic Beanstalk.

Elastic Beanstalk Service :

Elastic Beanstalk is an easy-to-use service for deploying and scaling Java web applications. Elastic Beanstalk supports several platform configurations for Java applications, including multiple versions of Java with the Apache Tomcat application server and Java-only configurations for applications that do not use Tomcat.

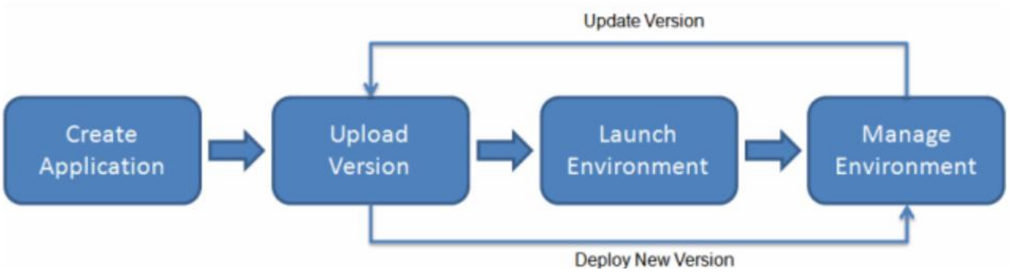
The Java-only option allows customers to include any required library JAR files in the source bundle for Java web applications that don't use a web container or use a different one, such as Jetty or GlassFish. Once deployed, Elastic Beanstalk automatically manages capacity provisioning, load balancing, and Auto Scaling. This approach is appropriate for companies deploying Java applications that include the following criteria:

- Require minimal OS changes. (Note that Elastic Beanstalk configuration files support advanced platform and OS configuration options. However, this requires additional Elastic Beanstalk packaging effort and expertise.)
- Either run in Apache Tomcat 7 or 8 or are packaged with their own web container

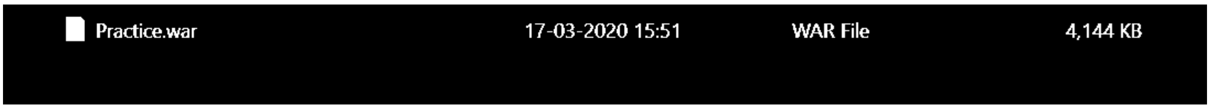
Elastic Beanstalk supports the following packaging and deployment mechanisms:

- Custom applications developed and deployed directly to Elastic Beanstalk using Eclipse and the AWS Toolkit for Eclipse.
- Applications packaged into a JAR, WAR, or ZIP file, then deployed with the Elastic Beanstalk console, EB CLI, or Elastic Beanstalk API calls. To deploy multiple applications to one Elastic Beanstalk environment, customers can bundle multiple WAR files into a single ZIP file.

Workflow in setting up Elastic Beanstalk :

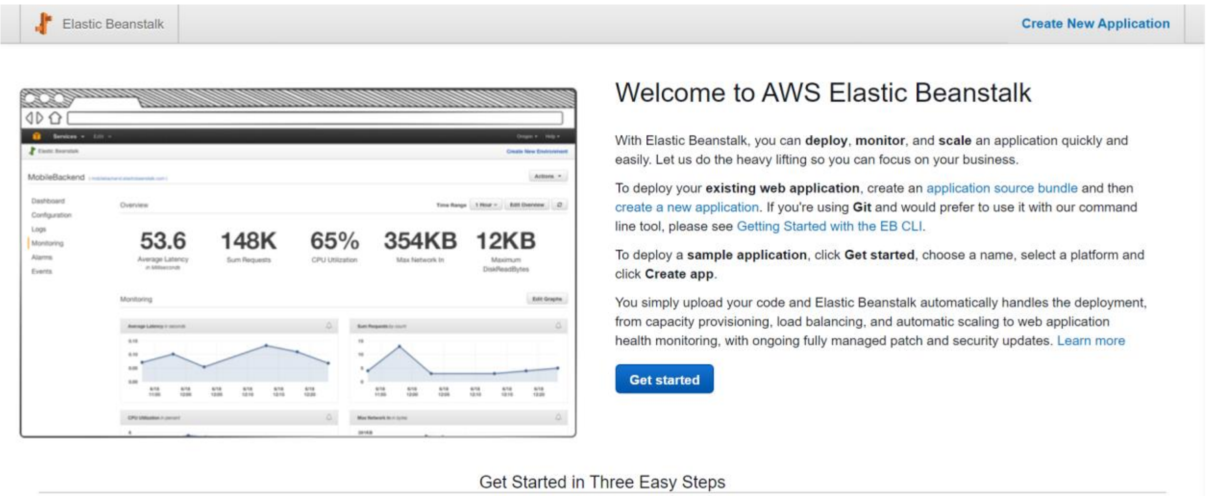


Screenshots :



Practice.war 17-03-2020 15:51 WAR File 4,144 KB

.War file of the project is built using Clean and Build option for the project.



Welcome to AWS Elastic Beanstalk

With Elastic Beanstalk, you can **deploy**, **monitor**, and **scale** an application quickly and easily. Let us do the heavy lifting so you can focus on your business.

To deploy your **existing web application**, create an application source bundle and then create a new application. If you're using **Git** and would prefer to use it with our command line tool, please see Getting Started with the EB CLI.

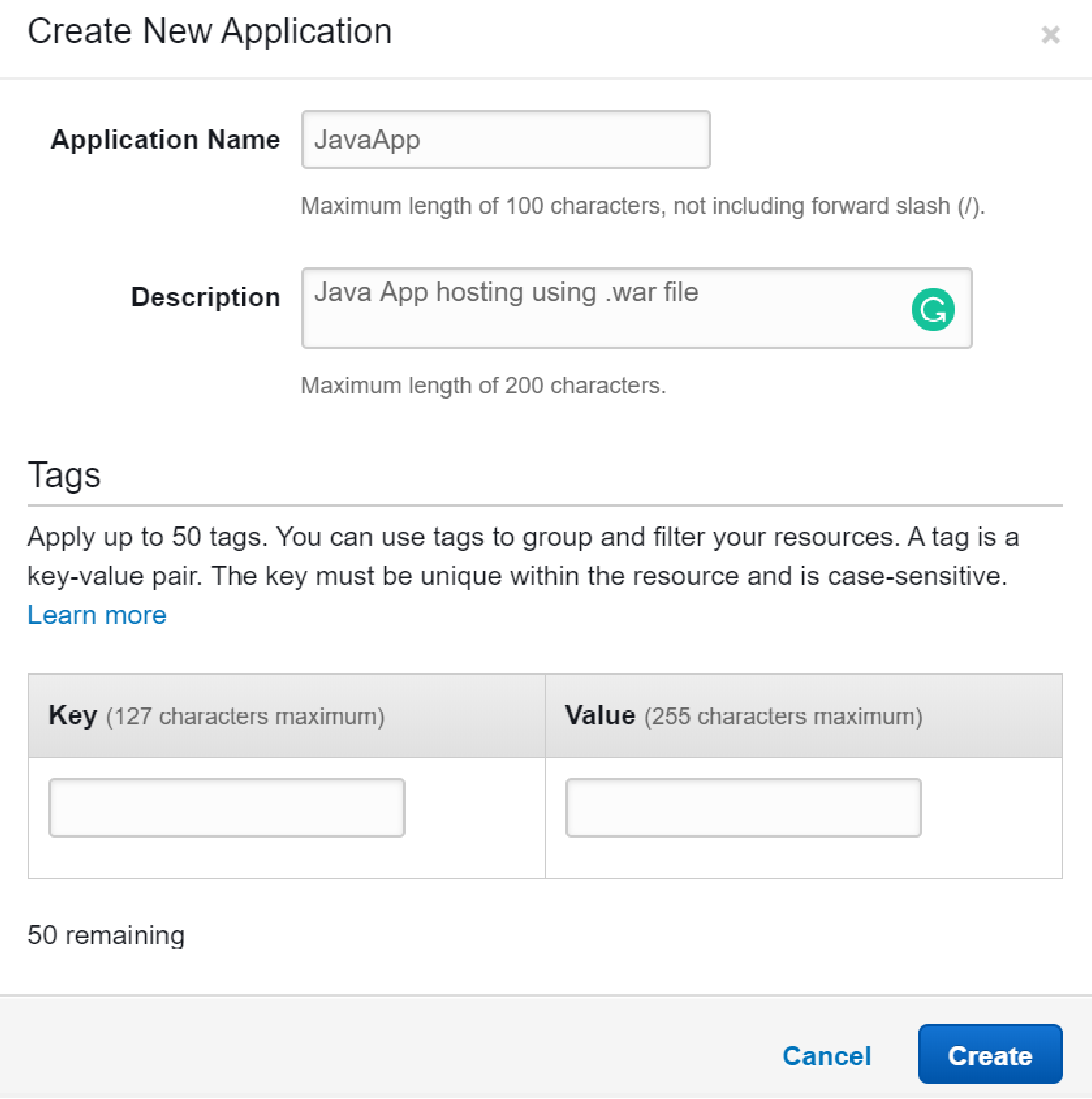
To deploy a **sample application**, click **Get started**, choose a name, select a platform and click **Create app**.

You simply upload your code and Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, and automatic scaling to web application health monitoring, with ongoing fully managed patch and security updates. Learn more

Get started

Get Started in Three Easy Steps

Create the New Application



Create New Application

Application Name: JavaApp

Maximum length of 100 characters, not including forward slash (/).

Description: Java App hosting using .war file

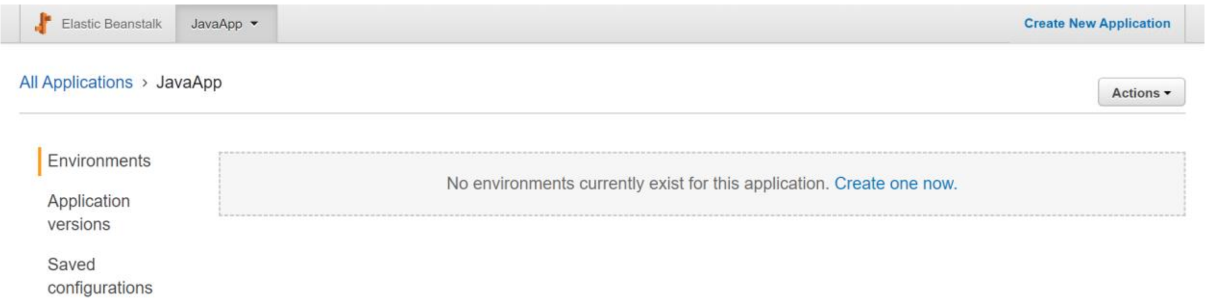
Maximum length of 200 characters.

Tags

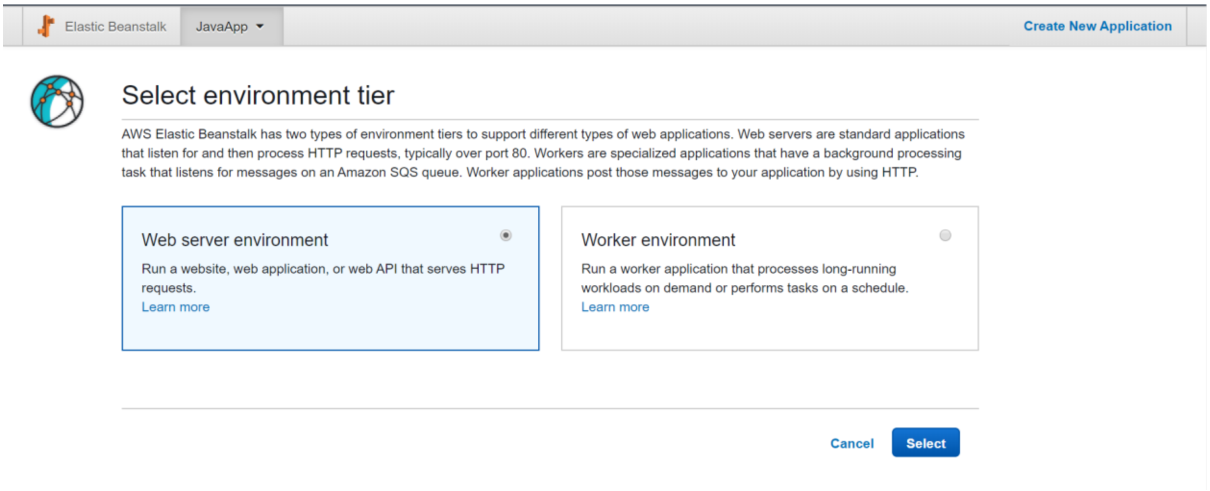
Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the resource and is case-sensitive. Learn more

Key (127 characters maximum)	**Value** (255 characters maximum)

50 remaining



No environments currently exist for this application. Create one now.



Select environment tier

AWS Elastic Beanstalk has two types of environment tiers to support different types of web applications. Web servers are standard applications that listen for and then process HTTP requests, typically over port 80. Workers are specialized applications that have a background processing task that listens for messages on an Amazon SQS queue. Worker applications post those messages to your application by using HTTP.

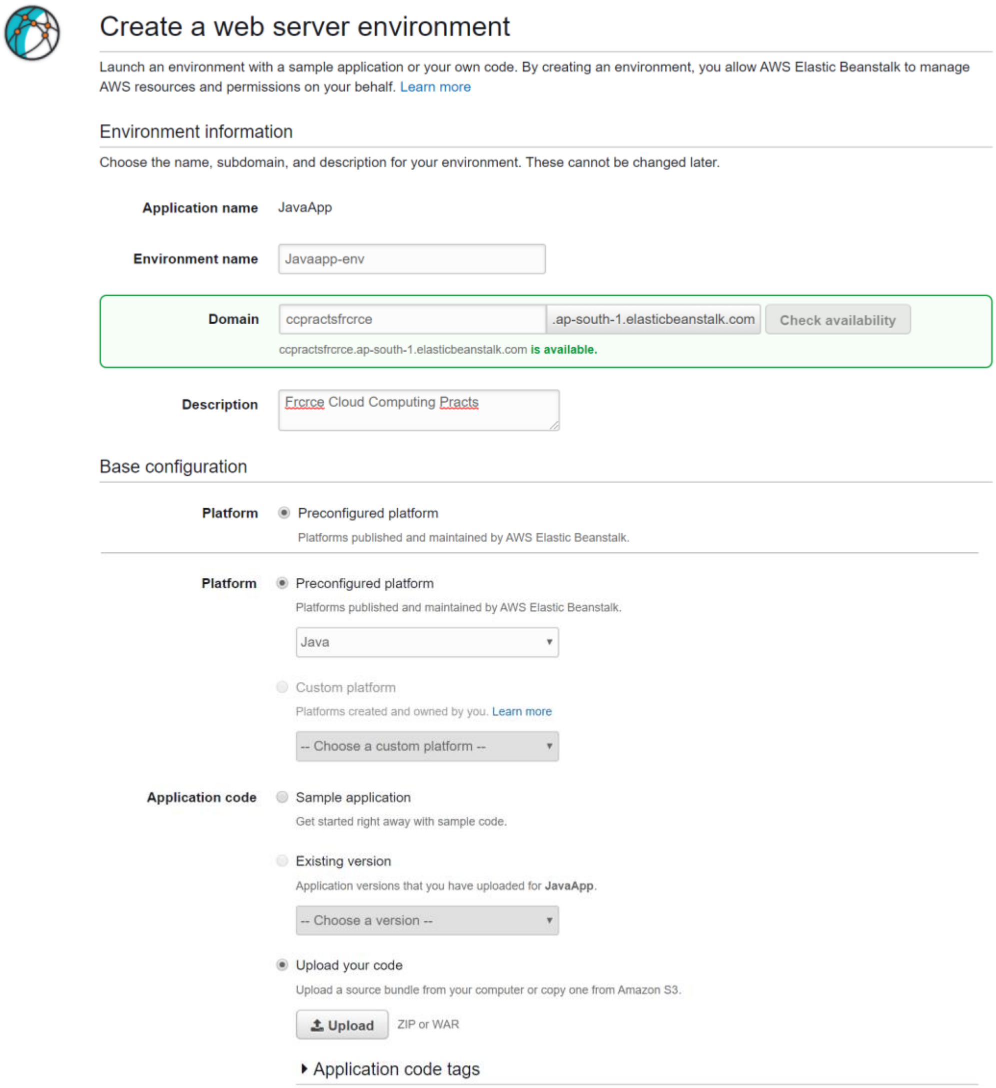
Web server environment ●

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

Setting up Web Server Environment



Create a web server environment

Launch an environment with a sample application or your own code. By creating an environment, you allow AWS Elastic Beanstalk to manage AWS resources and permissions on your behalf. Learn more

Environment information

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

Application name JavaApp

Environment name Javaapp-env

Domain ccpractsfrcrce .ap-south-1.elasticbeanstalk.com Check availability

ccpractsfrcrce.ap-south-1.elasticbeanstalk.com **is available.**

Description Frcrce Cloud Computing Practs

Base configuration

Platform ● Preconfigured platform

Platforms published and maintained by AWS Elastic Beanstalk.

Platform ● Preconfigured platform

Platforms published and maintained by AWS Elastic Beanstalk.

Java ▾

○ Custom platform

Platforms created and owned by you. Learn more

-- Choose a custom platform -- ▾

Application code ○ Sample application

Get started right away with sample code.

○ Existing version

Application versions that you have uploaded for **JavaApp**.

-- Choose a version -- ▾

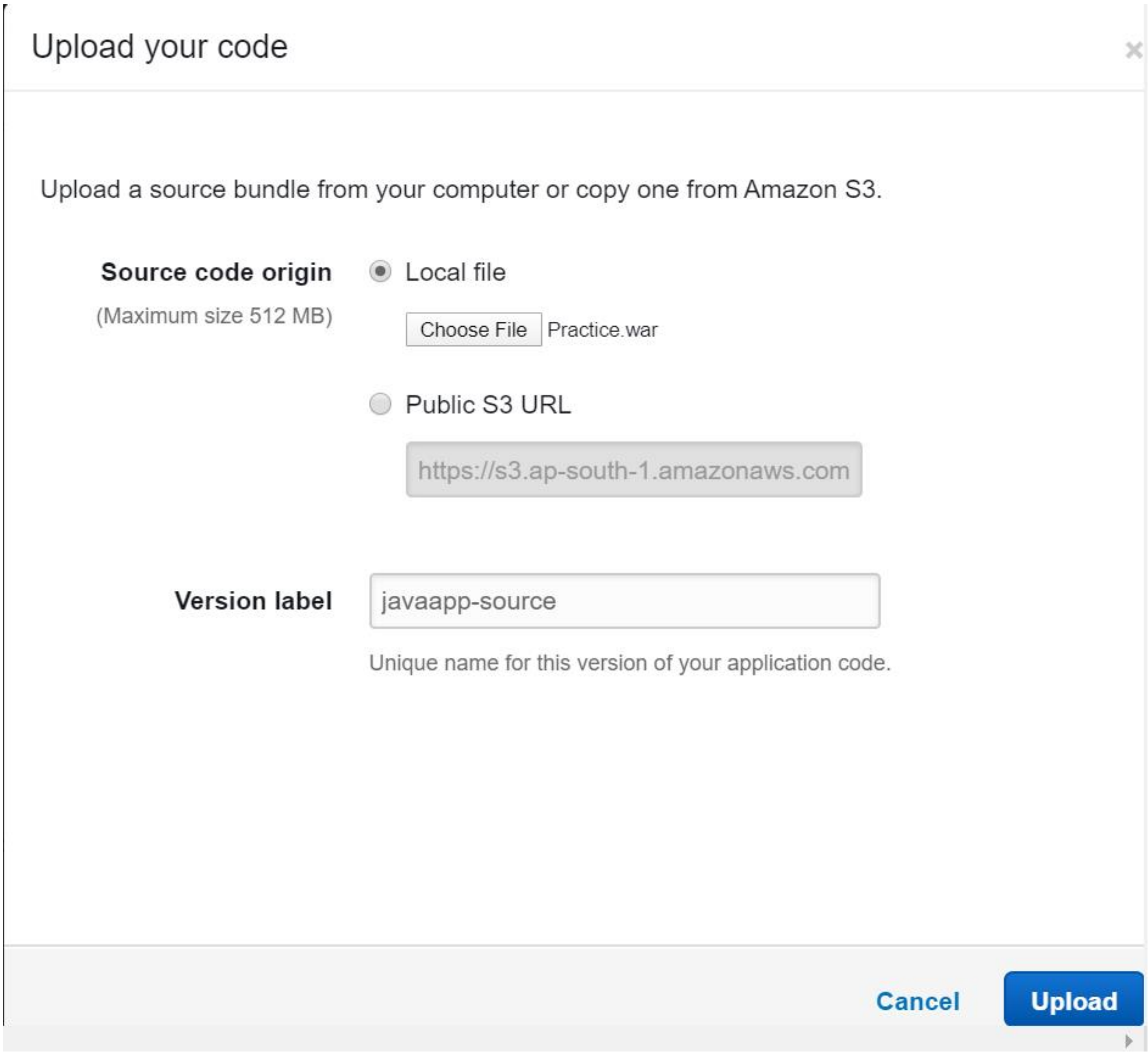
● Upload your code

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

Upload ZIP or WAR

▸ Application code tags

Uploading .WAR FILE



Upload your code

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

Source code origin ● Local file

(Maximum size 512 MB)

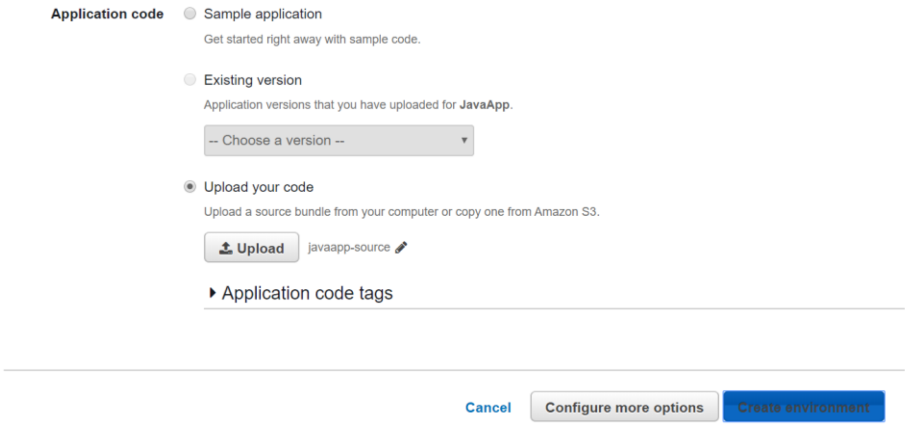
Choose File Practice.war

○ Public S3 URL

https://s3.ap-south-1.amazonaws.com

Version label javaapp-source

Unique name for this version of your application code.



Application code ○ Sample application

Get started right away with sample code.

○ Existing version

Application versions that you have uploaded for **JavaApp**.

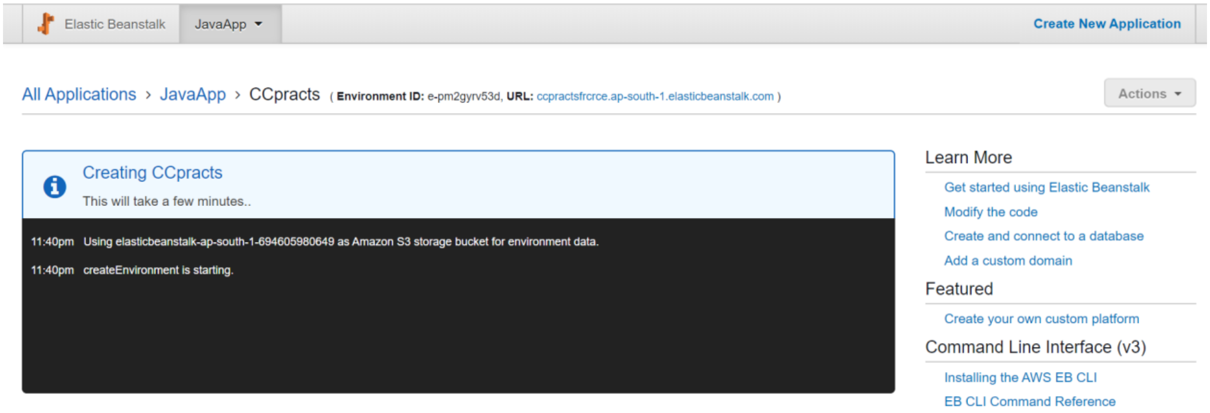
-- Choose a version --

● Upload your code

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

Upload javaapp-source

CREATION OF APPLICATION



All Applications › JavaApp › CCpracts (**Environment ID:** e-pm2gyrv53d, **URL:** ccpractsfrcrce.ap-south-1.elasticbeanstalk.com)

Creating CCpracts

This will take a few minutes..

```
11:40pm  Using elasticbeanstalk-ap-south-1-694605980649 as Amazon S3 storage bucket for environment data.
11:40pm  createEnvironment is starting.
```

Learn More

- Get started using Elastic Beanstalk
- Modify the code
- Create and connect to a database
- Add a custom domain

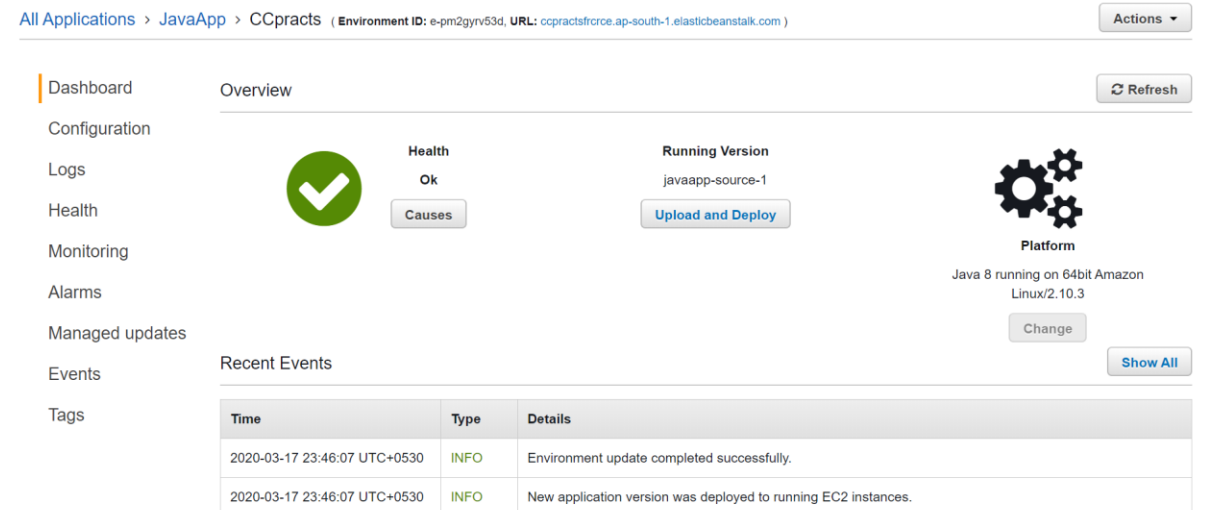
Featured

- Create your own custom platform

Command Line Interface (v3)

- Installing the AWS EB CLI
- EB CLI Command Reference

APPLICATION RUNNING AT THE ENDPOINT URL



All Applications › JavaApp › CCpracts (**Environment ID:** e-pm2gyrv53d, **URL:** ccpractsfrcrce.ap-south-1.elasticbeanstalk.com)

Overview

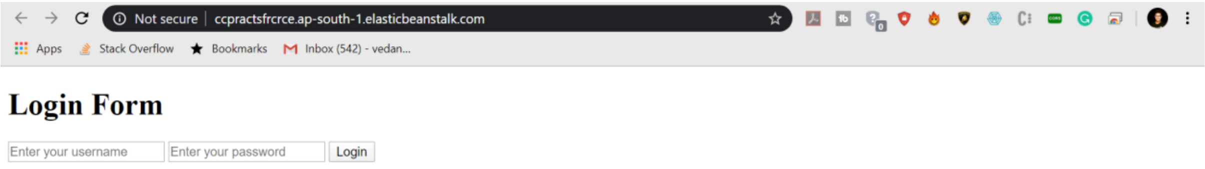
Health: Ok

Running Version: javaapp-source-1

Platform: Java 8 running on 64bit Amazon Linux/2.10.3

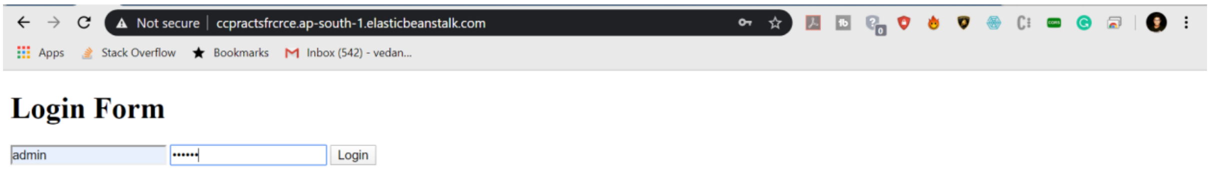
Recent Events

Time	Type	Details
2020-03-17 23:46:07 UTC+0530	INFO	Environment update completed successfully.
2020-03-17 23:46:07 UTC+0530	INFO	New application version was deployed to running EC2 instances.



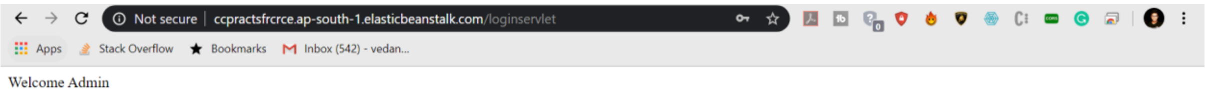
Login Form

Enter your username | Enter your password | Login



Login Form

admin ••••••


Not secure | ccpractsfrcrce.ap-south-1.elasticbeanstalk.com/loginservlet
Apps
Stack Overflow
Bookmarks
Inbox (542) - vedan...

Welcome Admin

Conclusion :

Java web application was deployed on the AWS cloud based service Elastic Beanstalk which supported Apache Tomcat server deployment and scales the application as and when scaling is required. Elastic Beanstalk automatically manages capacity provisioning, load balancing, and Auto Scaling. This approach is appropriate for companies deploying Java applications.

Post Lab Questions:

Q1. How does works Elastic Beanstalk?

Elastic beanstalk is a pre-configured EC2 server that can directly take up your application code and environment configurations and use it to automatically provision and deploy the required resources within AWS to run the web application. Unlike EC-2 which is Infrastructure as a service, Elastic beanstalk is a Platform As A service (PAAS) as it allows users to directly us a pre-configured server for their application. You can deploy applications without ever having to use elastic beanstalk but that would mean having to chose the appropriate service from the vast array of services offered by AWS, manually provisioning these AWS resources and stitching them up together to form a complete web-application. Elastic beanstalk abstracts the underlying configuration work and allows you as a user to focus on more pressing matters.

Q2. What are Benefits and features of Elastic Beanstalk?

Benefits & Features :-

1. Application: Elastic Beanstalk directly takes in out project code. So Elastic Beanstalk application is named the same as your project home directory.

2. Application Environments: Users may want their application to run on different environments like DEV, UAT and PROD. You can create and configure different environments to run application on different stages.

3. Environment Health: One of the most lucrative features about running application on AWS or most of the other cloud platforms is the automated health checks. AWS runs automatic health checks on all EC-2 deployments (Elastic Beanstalk is a managed EC-2 service) which can be monitored from AWS console. For example, in case of web applications AWS will regularly, as scheduled by the developers, ping the application to check if the response is status code 200 and the application is running as expected. Health check responses:

Red: Application failed all health tests.

Yellow: Application failed some of the health tests.

Grey: Application is updating.

Green: Application passed health check successfully.

4. Isolated: All environments within a single application are isolated from each other (independent of each others' running states). Needless to say two different applications are also isolated.

5. Scalability: Using Auto-Scaling within Elastic beanstalk makes the application dynamically scalable.

6. Elastic Load Balancing: All the web requests to the application are not directly relayed to application instances. They first hit the Elastic Load Balancer (ELB), which, as the name suggests, balances the load across all the application instances.

7. Language support: Elastic Beanstalk supports the applications developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.

8. Pricing: There is no extra charge for using Elastic Beanstalk. Users ar only required to pay for the services and resources provisioned by Elastic Beanstalk Service.

9. Automatic Provisioning: Elastic Beanstalk takes away the burden of choosing the right services and configuring their security groups to work together.

10. Impossible to Outgrow: AWS claims that since Elastic Beanstalk uses Auto Scaling feature it can, in theory, handle any amount of internet traffic.

Q3. Differentiate between AWS CloudFormation and AWS Elastic Beanstalk?

AWS CloudFormation: Create and manage a collection of related AWS resources. You can use AWS CloudFormation's sample templates or create your own templates to describe the AWS resources, and any associated dependencies or runtime parameters, required to run your application. You don't need to figure out the order in which AWS services need to be provisioned or the subtleties of how to make those dependencies work; AWS Elastic Beanstalk: Quickly deploy and manage applications in the AWS cloud. Once you upload your application, Elastic Beanstalk automatically handles the deployment details of capacity provisioning, load balancing, auto-scaling, and application health monitoring.

AWS CloudFormation belongs to "Infrastructure Build Tools" category of the tech stack, while AWS Elastic Beanstalk can be primarily classified under "Platform as a Service".

Some of the features offered by AWS CloudFormation are:

AWS CloudFormation comes with the following ready-to-run sample templates: WordPress (blog),Tracks (project tracking), Gollum (wiki used by GitHub), Drupal (content management), Joomla (content management), Insoshi (social apps), Redmine (project mgmt)

No Need to Reinvent the Wheel – A template can be used repeatedly to create identical copies of the same stack (or to use as a foundation to start a new stack)

Transparent and Open – Templates are simple JSON formatted text files that can be placed under your normal source control mechanisms, stored in private or public locations such as Amazon S3 and exchanged via email.

On the other hand, AWS Elastic Beanstalk provides the following key features:

Elastic Beanstalk is built using familiar software stacks such as the Apache HTTP Server for Node.js, PHP and Python, Passenger for Ruby, IIS 7.5 for .NET, and Apache Tomcat for Java

There is no additional charge for Elastic Beanstalk - you pay only for the AWS resources needed to store and run your applications.

Easy to begin – Elastic Beanstalk is a quick and simple way to deploy your application to AWS. You simply use the AWS Management Console, Git deployment, or an integrated development environment (IDE) such as Eclipse or Visual Studio to upload your application

Q4. Who should use AWS Elastic Beanstalk?

Those who want to deploy and manage their applications within minutes in the AWS Cloud. You don't need experience with cloud computing to get started. AWS Elastic Beanstalk supports Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker web applications.

Q5. What Language Supported by Elastic Beanstalk

AWS Elastic Beanstalk is designed so that it can be extended to support multiple development stacks and programming languages in the future. AWS is working with solution providers on the APIs and capabilities needed to create additional Elastic Beanstalk offerings. However, it supports the following languages and development stacks:

Apache Tomcat for Java applications

Apache HTTP Server for PHP applications

Apache HTTP Server for Python applications

Nginx or Apache HTTP Server for Node.js applications

Passenger or Puma for Ruby applications

Microsoft IIS 7.5, 8.0, and 8.5 for .NET applications

Java SE

Docker

Go

Q6. Can we use Elastic beanstalk to deploy the web application which is made in Node OR PHP?

yes

Q7. What kinds of applications are supported by AWS Elastic

Beanstalk?

With AWS Elastic Beanstalk, you can:

Select the operating system that matches your application requirements (e.g., Amazon Linux or Windows Server 2016)

Choose from several Amazon EC2 instances including On-Demand, Reserved instances, and Spot instances

Choose from several available database and storage options

Enable login access to Amazon EC2 instances for immediate and direct troubleshooting

Quickly improve application reliability by running in more than one Availability Zone

Enhance application security by enabling HTTPS protocol on the load balancer

Access built-in Amazon CloudWatch monitoring and getting notifications on application health and other important events

Adjust application server settings (e.g., JVM settings) and pass environment variables

Run other application components, such as a memory caching service, side-by-side in Amazon EC2

Access log files without logging in to the application servers

MCQS:

Q1] How does Elastic Beanstalk apply updates?

A. By having a duplicate ready with updates before swapping.
B. By updating on the instance while it is running
C. By taking the instance down in the maintenance window
D. Updates should be installed manually

Answer A.

Explanation: Elastic Beanstalk prepares a duplicate copy of the instance, before updating the original instance, and routes your traffic to the duplicate instance, so that, incase your updated application fails, it will switch back to the original instance, and there will be no downtime experienced by the users who are using your application.

Q2] Elastic Beanstalk is used for?

A. Database
B. Deployment
C. Storage
D. Volume Based Storage

Answer B.

Explanation : AWS Elastic Beanstalk makes it even easier for developers to quickly deploy and manage applications in the AWS Cloud. Developers simply upload their application, and Elastic Beanstalk automatically handles the deployment details of capacity provisioning, load balancing, auto-scaling, and application health monitoring.

Q3] Which OS does Elastic Beanstalk use?

A. **Linux AMI**
B. **Windows AMI**
C. **Both A and B**
D. **None of the above**

Answer C.

Explanation : AWS Elastic Beanstalk runs on the Amazon Linux AMI and the Windows Server AMI. Both AMIs are supported and maintained by Amazon Web Services and are designed to provide a stable, secure, and high-performance execution environment for Amazon EC2 Cloud computing.