



Vidyavardhini's College of Engineering & Technology

Department of Computer Science and Engineering (Data Science)

Course: SBL: Cloud Computing

Course code: CSL605

Year: TE SEM: VI

Experiment No.03

AIM:- To study and Implement Platform as a Service using AWS Elastic Beanstalk.

Name:

Roll Number:

Date of Performance:

Date of Submission:

Evaluation

Performance Indicator	Max. Marks	Marks Obtained
Performance	5	
Understanding	5	
Journal work and timely submission.	10	
Total	20	

Performance Indicator	Exceed Expectations (EE)	Meet Expectations (ME)	Below Expectations (BE)
Performance	5	3	2
Understanding	5	3	2
Journal work and timely submission.	10	8	4

Checked by

Name of Faculty : Ichhanshu Jaiswal

Signature :

Date :



Experiment No. 3

Aim: To study and Implement Platform as a Service using AWS Elastic Beanstalk.

Theory:

Platform as a Service (PaaS) is a complete cloud environment that includes everything developers need to build, run, and manage applications—from servers and operating systems to all the networking, storage, middleware, tools, and more.

Like [IaaS](#), PaaS includes infrastructure—servers, storage, and networking—but also middleware, development tools, business intelligence (BI) services, database management systems, and more. PaaS is designed to support the complete web application lifecycle: building, testing, deploying, managing, and updating.

PaaS allows you to avoid the expense and complexity of buying and managing software licenses, the underlying application infrastructure and middleware, container orchestrators such as [Kubernetes](#), or the development tools and other resources. You manage the applications and services you develop, and the cloud service provider typically manages everything else.

With Elastic Beanstalk, you can quickly deploy and manage applications in the AWS Cloud without having to learn about the infrastructure that runs those applications. Elastic Beanstalk reduces management complexity without restricting choice or control. You simply upload your application, and Elastic Beanstalk automatically handles the details of capacity provisioning, load balancing, scaling, and application health monitoring.

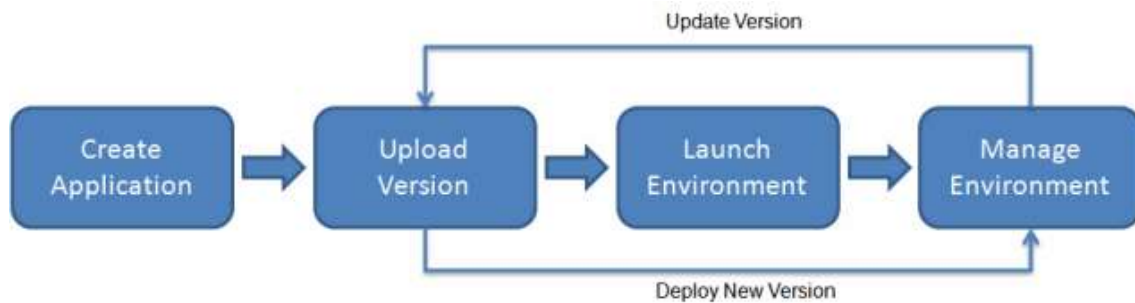
Elastic Beanstalk supports applications developed in Go, Java, .NET, Node.js, PHP, Python, and Ruby. When you deploy your application, Elastic Beanstalk builds the selected supported platform version and provisions one or more AWS resources, such as Amazon EC2 instances, to run your application.

You can interact with Elastic Beanstalk by using the Elastic Beanstalk console, the AWS Command Line Interface (AWS CLI), or **eb**, a high-level CLI designed specifically for Elastic Beanstalk.

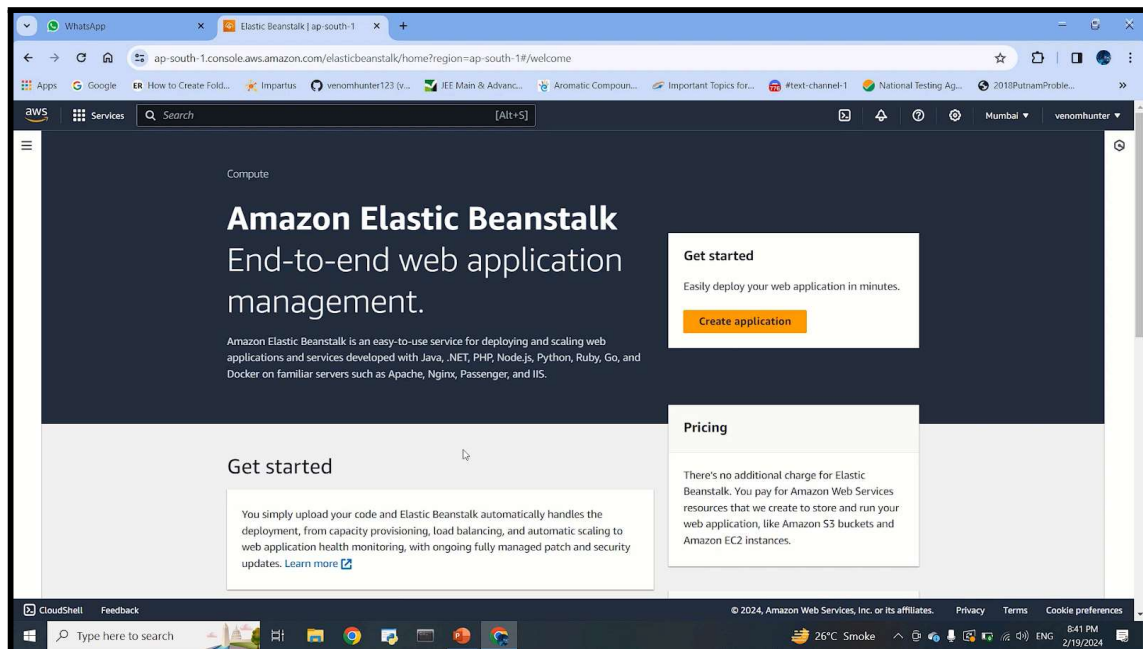


Vidyavardhini's College of Engineering & Technology

Department of Computer Science and Engineering (Data Science)



Snapshots of implementation:





Vidyavardhini's College of Engineering & Technology

Department of Computer Science and Engineering (Data Science)

Roles (3) 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.

Role name	Trusted entities	Last activity
aws-elasticbeanstalk-service-role	AWS Service: elasticbeanstalk	-
AWSServiceRoleForSupport	AWS Service: support (Service-Linker)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service)	-

Roles Anywhere Info

Authenticate your non AWS workloads and securely provide access to AWS services.

Access AWS from your non AWS workloads

Operate your non AWS workloads using the same authentication and authorization strategy that you use within AWS.

X.509 Standard

Use your own existing PKI infrastructure or use AWS Certificate Manager Private Certificate Authority to authenticate identities.

Temporary credentials

Use temporary credentials with ease and benefit from the enhanced security they provide.

Role SAIRAJDEMOIAM created.

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.

Role name	Trusted entities	Last activity
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service)	-
SAIRAJDEMOIAM	AWS Service: ec2	-

Roles Anywhere Info

Authenticate your non AWS workloads and securely provide access to AWS services.

Access AWS from your non AWS workloads

Operate your non AWS workloads using the same authentication and authorization strategy that you use within AWS.

X.509 Standard

Use your own existing PKI infrastructure or use AWS Certificate Manager Private Certificate Authority to authenticate identities.

Temporary credentials

Use temporary credentials with ease and benefit from the enhanced security they provide.



Vidyavardhini's College of Engineering & Technology

Department of Computer Science and Engineering (Data Science)

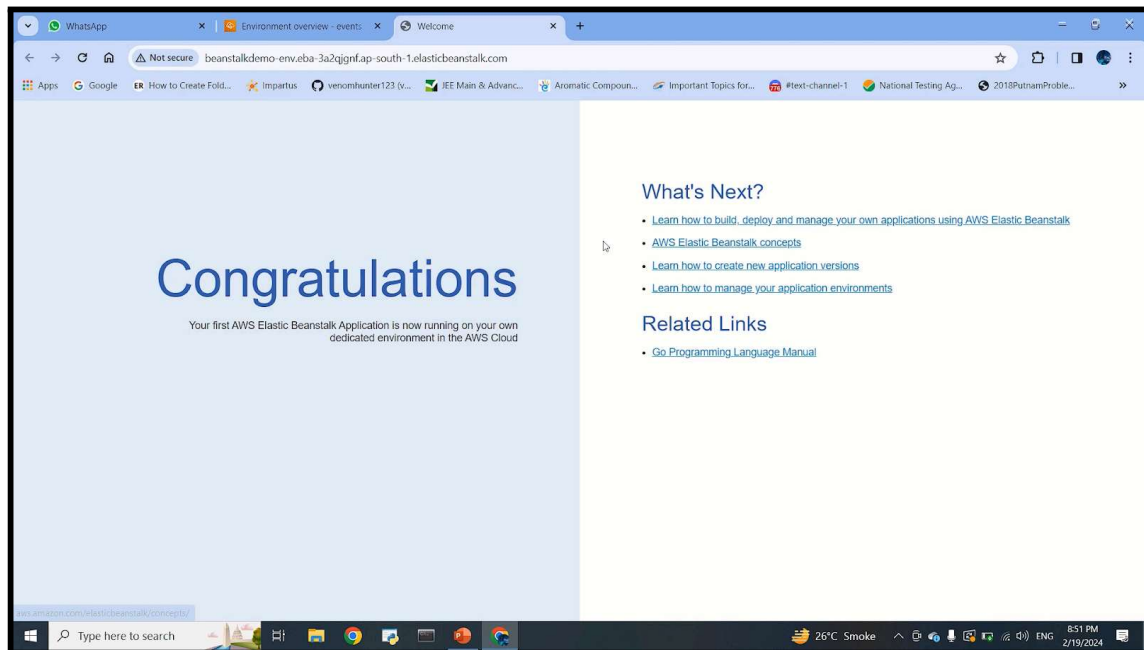
The screenshot shows the 'Review' step of the AWS Elastic Beanstalk console. The left sidebar lists the steps: Step 1: Configure environment (selected), Step 2: Configure service access, Step 3 - optional: Set up networking, database, and tags, Step 4 - optional: Configure instance traffic and scaling, Step 5 - optional: Configure updates, monitoring, and logging, and Step 6: Review. The main content area is divided into two sections: 'Step 1: Configure environment' and 'Step 2: Configure service access'. The 'Step 1' section shows 'Environment information' with fields for Environment tier (Web server environment), Environment name (BEANSTALKDEMO-env), Platform (arn:aws:elasticbeanstalk:ap-south-1:platform/Go 1 running on 64bit Amazon Linux 2023/4.0.3), Application name (BEANSTALKDEMO), and Application code (Sample application). The 'Step 2' section shows 'Service access' with a description: 'Configure the service role and EC2 instance profile that Elastic Beanstalk uses to manage your environment. Choose an EC2 key pair to securely log in to your EC2 instances.'

The screenshot shows the 'Environment overview' page for the 'BEANSTALKDEMO-env' environment. The left sidebar shows the 'Elastic Beanstalk' navigation menu with options: Applications, Environments, Change history, Application: BEANSTALKDEMO (Application versions, Saved configurations), Environment: BEANSTALKDEMO-env (Go to environment, Configuration, Events, Health, Logs, Monitoring, Alarms, Managed updates). The main content area shows the 'Environment overview' with fields for Health (Unknown), Environment ID (e-jpe5hwkh3h), Domain (---), and Application name (BEANSTALKDEMO). The 'Platform' section shows 'Go 1 running on 64bit Amazon Linux 2023/4.0.3', 'Running version' (---), and 'Platform state' (Supported). The 'Events' section shows 'Events (2) Info'. The top of the page has a blue banner stating 'Elastic Beanstalk is launching your environment. This will take a few minutes.'



Vidyavardhini's College of Engineering & Technology

Department of Computer Science and Engineering (Data Science)



Video Link: [Dhanashree_Thakur_56_Exp3.mp4](#)



Vidyavardhini's College of Engineering & Technology

Department of Computer Science and Engineering (Data Science)

Conclusion:

Comment on the features provided by Elastic Beanstalk

Ans: Elastic Beanstalk is a Platform as a Service (PaaS) offering from AWS that simplifies the deployment, management, and scaling of web applications and services. Here are some of the key features provided by Elastic Beanstalk:

Easy Deployment: Elastic Beanstalk automates the process of deploying applications by handling the provisioning of underlying infrastructure components, such as EC2 instances, load balancers, databases, and networking resources. Developers can deploy their applications with just a few clicks or through the command-line interface (CLI), without needing to manage the underlying infrastructure.

Multiple Language and Framework Support: Elastic Beanstalk supports a wide range of programming languages and frameworks, including Java, .NET, Python, Node.js, Ruby, Go, and Docker containers. This flexibility allows developers to choose the language and framework that best suits their application requirements.

Automatic Scaling: Elastic Beanstalk provides built-in support for automatic scaling, allowing applications to dynamically adjust their compute capacity based on traffic demands. Users can configure auto-scaling policies to scale up or down the number of EC2 instances or containers in response to changes in application load, ensuring optimal performance and cost efficiency.

Managed Updates: Elastic Beanstalk handles the deployment of application updates and patches, including operating system updates and security fixes. It automatically manages the rolling updates of instances to minimize downtime and ensure high availability of the application.