

EXPERIMENT 2

NAME: SHIVPRATIK HANDE

CLASS:D15C ROLL NO:14

Aim :To Build Your Application using AWS CodeBuild and Deploy on S3 / SEBS using AWS CodePipeline, deploy Sample Application on EC2 instance using AWS CodeDeploy.

Elastic Beanstalk

Step 1: create environment

Environment tier [Info](#)
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 [Info](#)

Application name
mohitbean
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
mohitbean-1

Step 2 : add your Ec2 key pair and instance profile

VPC
Launch your environment in a custom VPC instead of the default VPC. You can create a VPC and subnets in the VPC management console. [Learn more](#)

vpc-0b8f24d7c64f9775f | (172.31.0.0/16)

[Create custom VPC](#)

Instance settings
Choose a subnet in each AZ for the instances that run your application. To avoid exposing your instances to the Internet, run your instances in private subnets and load balancer in public subnets. To run your load balancer and instances in the same public subnets, assign public IP addresses to the instances. [Learn more](#)

Public IP address
Assign a public IP address to the Amazon EC2 instances in your environment.

☒ **Activated**

Instance subnets

Filter instance subnets

	Availability Zone	Subnet	CIDR	Name
<input checked="" type="checkbox"/>	ap-south-1a	subnet-05abae131...	172.31.32.0/20	
<input type="checkbox"/>	ap-south-1c	subnet-07c39e2a0...	172.31.16.0/20	
<input type="checkbox"/>	ap-south-1b	subnet-084bcd8f0...	172.31.0.0/20	

Step 3 : add security config and review all settings

This screenshot shows the 'Monitoring' configuration page in the AWS Elastic Beanstalk console. The left sidebar contains a list of steps: 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 titled 'Monitoring' and includes the following sections:

- Health reporting**: Enhanced health reporting provides free real-time application and operating system monitoring of the instances and other resources in your environment. The EnvironmentHealth custom metric is provided free with enhanced health reporting. Additional charges apply for each custom metric. For more information, see [Amazon CloudWatch Pricing](#).
- System**: Radio buttons for 'Basic' and 'Enhanced'. 'Enhanced' is selected.
- CloudWatch Custom Metrics - Instance**: A dropdown menu with 'Choose metrics' selected.
- CloudWatch Custom Metrics - Environment**: A dropdown menu with 'Choose metrics' selected.
- Health event streaming to CloudWatch Logs**: Configure Elastic Beanstalk to stream environment health events to CloudWatch Logs. You can set the retention up to a maximum of ten years and configure Elastic Beanstalk to delete the logs when you terminate your environment.
- Log streaming**: A checkbox for 'Activated (standard CloudWatch charges apply.)' is unchecked.
- Retention**: A dropdown menu with '7' selected.

This screenshot shows the 'Review' page in the AWS Elastic Beanstalk console. The left sidebar contains a list of 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 (Review). The main content area is titled 'Review' and includes the following sections:

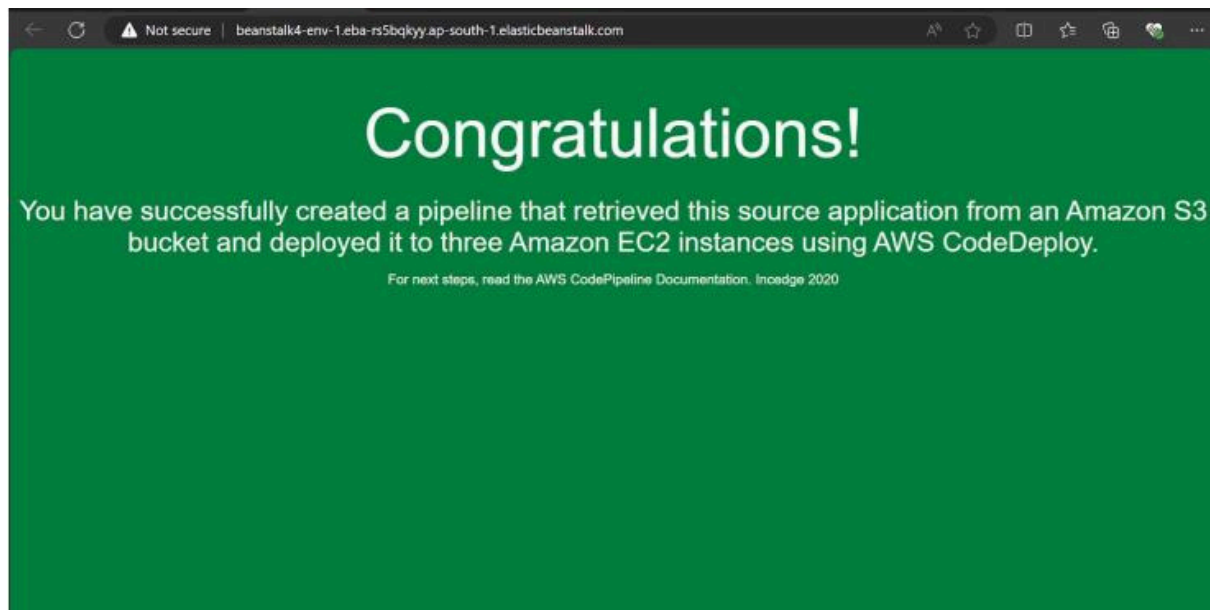
- Step 1: Configure environment**: Includes an 'Edit' button and a table for 'Environment information'.
- Environment information**:

Environment tier	Application name
Web server environment	mohitbean
Environment name	Application code
Mohitbean-env	Sample application
Platform	
amazonaws:elasticbeanstalk:ap-south-1:platform/PHP 8.3 running on 64bit Amazon Linux 2023/4.3.2	
- Step 2: Configure service access**: Includes an 'Edit' button and a section for 'Service access'.
- Service access**: 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.

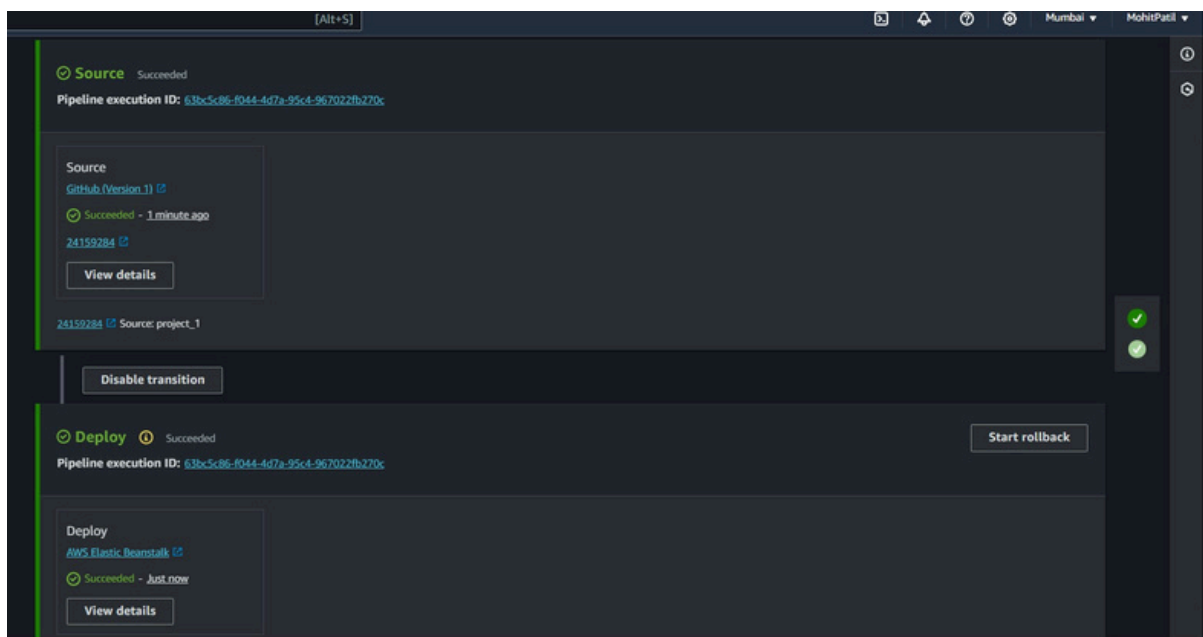
This screenshot shows the 'Mohitbean-env' overview page in the AWS Elastic Beanstalk console. The left sidebar contains a list of steps: Step 1 (Configure environment), Step 2 (Configure service access), Step 3 - optional (Set up networking, database, and tags), Step 4 - optional (Configure instance traffic and scaling), Step 5 - optional (Configure updates, monitoring, and logging), and Step 6 (Review). The main content area is titled 'Mohitbean-env' and includes the following sections:

- Environment overview**: Includes a 'Health' status of 'Ok' and an 'Environment ID' of 'e-8jcp97pasv'.
- Platform**: Includes a 'Platform' of 'PHP 8.3 running on 64bit Amazon Linux 2023/4.3.2' and a 'Platform state' of 'Supported'.
- Events**: A tab for 'Events (12)' with an 'Info' button.

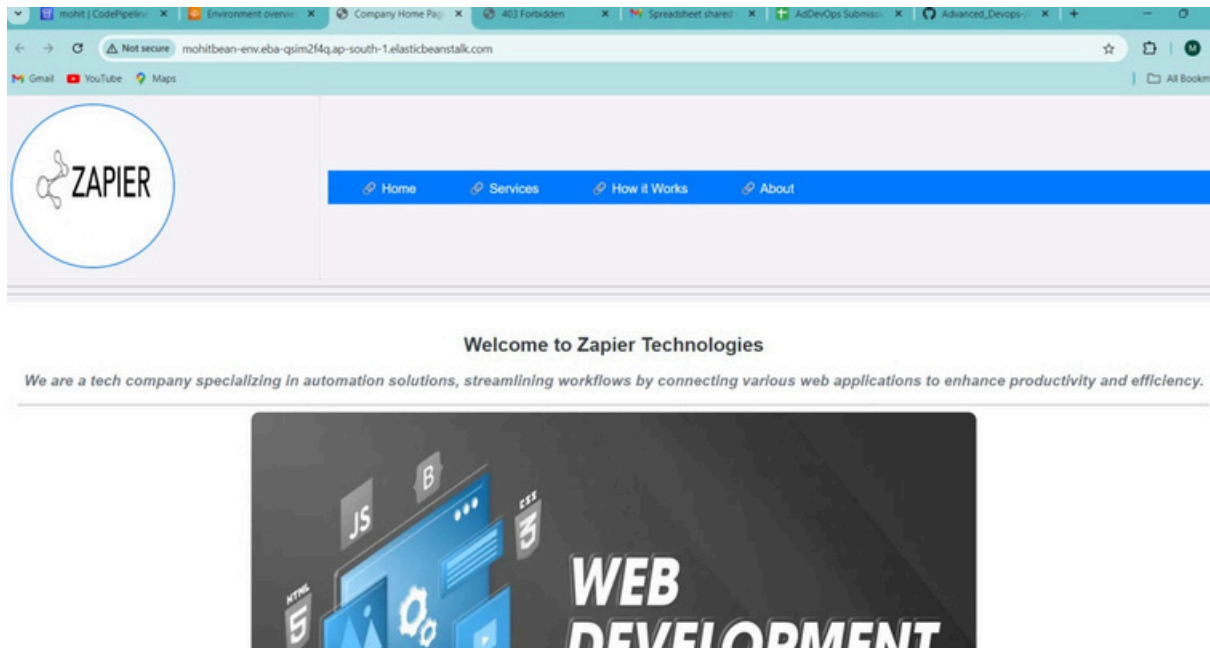
Step 4 : Beanstalk environment is created



view the pipeline build and deployment



Check the deployed website at beanstalk link



Conclusion:

The integration of AWS CodeBuild, CodePipeline, and CodeDeploy provides a powerful framework for continuous integration and continuous delivery (CI/CD) of applications. By leveraging these AWS services, developers can automate the entire lifecycle of application development—from building and testing to deployment—resulting in a more efficient and reliable workflow.