**Advanced DevOps Experiment-2**

**Aim:** Using AWS CodePipeline, deploy Sample Application on Elastic BeanStalk using AWS CodeDeploy.

# Theory:-

Elastic Beanstalk simplifies the process of deploying and managing applications on the AWS cloud, allowing developers to focus on building applications without needing to worry about the underlying infrastructure. AWS offers over 100 services, which can sometimes make it challenging to know exactly how to configure and provision the right resources. Elastic Beanstalk removes this complexity by automatically managing the necessary AWS resources— such as Amazon EC2 instances, load balancing, and scaling—while still giving you control and flexibility over your setup.

Elastic Beanstalk supports a variety of programming languages including Go, Java, .NET, Node.js, PHP, Python, Ruby, and Docker platforms. This means that if your app is built in one of these languages or platforms, Elastic Beanstalk can handle its deployment. For Docker containers, you can even use your own custom configurations and dependencies, expanding the potential use cases beyond those natively supported languages.

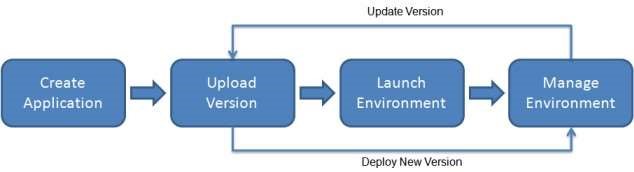
After your environment is set up, you can continue to manage and monitor it through the web interface or the CLI tools. You can scale your application by adjusting the number of EC2 instances or deploying new versions of your app with ease. The end result is a fully automated workflow that takes care of the technical heavy lifting, allowing you to deliver updates and maintain your app without diving deep into infrastructure management.

Here's a simplified breakdown of how Elastic Beanstalk works:

Create the application: Upload your code and define an application version.Elastic Beanstalk provisions resources: It automatically sets up the required AWS services and resources (EC2, load balancers, etc.).

Deploy and manage: You can monitor the application’s performance, scale resources, and deploy new versions as needed.

Elastic Beanstalk makes AWS cloud management simple, while keeping flexibility intact. It’s like having a smart assistant for infrastructure that automates the complexities of cloud provisioning but still allows you to customize when needed.



After you create and deploy your application, information about the application—including metrics, events, and environment status—is available through the Elastic Beanstalk console, APIs, or Command Line Interfaces, including the unified AWS CLI.

**Implementation:-**

**Deploying basic web page on Elastic Beanstalk**

