**AWS Elastic Beanstalk**

Amazon Web Services (AWS) comprises dozens of services, each of which exposes an area of functionality. While the variety of services offers flexibility for how you want to manage your AWS infrastructure, it can be challenging to figure out which services to use and how to provision them.

Ready to use Infrastructure for (Java,pthyon……) by creating new Ec2 instances and use tht.

Simple create and upload the application,it prepares the container with Monitoring.

With Elastic Beanstalk, you can quickly deploy and manage applications in the AWS Cloud without worrying about the infrastructure that runs those applications. AWS 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 Java, PHP, .NET, Node.js, Python, and Ruby, as well as different container types for each language.

A container defines the infrastructure and software stack to be used for a given environment. When you deploy your application, Elastic Beanstalk provisions one or more AWS resources, such as Amazon EC2 instances. The software stack that runs on your Amazon EC2 instances depends on the container type

# AWS Elastic Beanstalk Concepts

## Application

An Elastic Beanstalk application is a logical collection of Elastic Beanstalk components, including environments, versions, and environment configurations. In Elastic Beanstalk an application is conceptually similar to a folder.

## Application Version (Client Application)

In Elastic Beanstalk, an application version refers to a specific, labeled iteration of deployable code for a web application. An application version points to an Amazon Simple Storage Service (Amazon S3) object that contains the deployable code such as a Java WAR file. An application version is part of an application.

## Environment (for example tomcat)

An environment is a version that is deployed onto AWS resources. Each environment runs only a single application version at a time, however you can run the same version or different versions in many environments at the same time. When you create an environment, Elastic Beanstalk provisions the resources needed to run the application version you specified.

## Environment Tier

When you launch an Elastic Beanstalk environment, you first choose an environment tier. The environment tier that you choose determines whether Elastic Beanstalk provisions resources to support an application that handles HTTP requests or an application that pulls tasks from a queue. An application that serves HTTP requests runs in a web server environment.

## Environment Configuration

An environment configuration identifies a collection of parameters and settings that define how an environment and its associated resources behave. When you update an environment’s configuration settings, Elastic Beanstalk automatically applies the changes to existing resources or deletes and deploys new resources (depending on the type of change).

## Configuration Template

A configuration template is a starting point for creating unique environment configurations. Configuration templates can be created or modified by using the Elastic Beanstalk command line utilities or API. Predefined templates for Containers.

|  |
| --- |
| * **Single Container Docker** * **Multicontainer Docker** * **Preconfigured Docker (Glassfish)** * **Preconfigured Docker (Python 3)** * **Preconfigured Docker (Go)** * **Go** * **Java SE** * **Tomcat** * **.NET** * **Node.js** * **PHP** * **Ruby** |