

AWS Elastic Beanstalk

Allows you to quickly deploy and manage applications in the AWS Cloud without worrying about the infrastructure that runs those applications.

- Elastic Beanstalk automatically handles the details of capacity provisioning, load balancing, scaling, and application health monitoring for your applications.
- It is a Platform-as-a-Service
- Elastic Beanstalk supports the following languages:
 - Go
 - Java
 - .NET
 - Node.js
 - PHP
 - Python
 - Ruby
- Elastic Beanstalk supports the following web containers:
 - Tomcat
 - Passenger
 - Puma
- Elastic Beanstalk supports Docker containers.
- Elastic Beanstalk Workflow



- Your application's domain name is in the format:
subdomain.region.elasticbeanstalk.com

Environment Pages

- The **Configuration** page shows the resources provisioned for this environment. This page also lets you configure some of the provisioned resources.
- The **Health** page shows the status and detailed health information about the EC2 instances running your application.
- The **Monitoring** page shows the statistics for the environment, such as average latency and CPU utilization. You also use this page to create alarms for the metrics that you are monitoring.
- The **Events** page shows any informational or error messages from services that this environment is using.
- The **Tags** page shows tags — key-value pairs that are applied to resources in the environment. You use this page to manage your environment's tags.

AWS Elastic Beanstalk Concepts

- **Application** – a logical collection of Elastic Beanstalk components, including environments, versions, and environment configurations. It is conceptually similar to a folder.
- **Application Version** – refers to a specific, labeled iteration of deployable code for a web application. An application version points to an Amazon S3 object that contains the deployable code. Applications can have many versions and each application version is unique.
- **Environment** – a version that is deployed on to 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.
- **Environment Tier** – 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**. An environment that pulls tasks from an Amazon SQS queue runs in a **worker environment**.
- **Environment Configuration** – identifies a collection of parameters and settings that define how an environment and its associated resources behave.
- **Saved Configuration** – a starting point for creating unique environment configurations.
- **Platform** – a combination of OS, language runtime, web/application server and Elastic Beanstalk components.
- There is a limit to the number of application versions you can have. You can avoid hitting the limit by applying an *application version lifecycle policy* to your applications to tell Elastic Beanstalk to delete application versions that are old, or to delete application versions when the total number of versions for an application exceeds a specified number.

Environment Types

- Load-balancing, Autoscaling Environment – automatically starts additional instances to accommodate increasing load on your application.
- Single-Instance Environment – contains one Amazon EC2 instance with an Elastic IP address.

Environment Configurations

- Your environment contains:
 - Your **EC2 virtual machines** configured to run web apps on the platform that you choose.
 - An **Auto Scaling group** that ensures that there is always one instance running in a single-instance environment, and allows configuration of the group with a range of instances to run in a load-balanced environment.
 - When you enable load balancing, Elastic Beanstalk creates an **Elastic Load Balancing load balancer** to distributes traffic among your environment's instances.
 - Elastic Beanstalk provides integration with **Amazon RDS** to help you add a database instance to your Elastic Beanstalk environment : **MySQL, PostgreSQL, Oracle, or SQL Server**. When you add a database instance to your environment, Elastic Beanstalk provides connection information to your application by setting environment properties for the database hostname, port, user name, password, and database name.
 - You can use **environment properties** to pass secrets, endpoints, debug settings, and other information to your application. Environment properties help you run your application in multiple environments for different purposes, such as development, testing, staging, and production.
 - You can configure your environment to use **Amazon SNS** to notify you of important events that affect your application.
 - Your environment is available to users at a **subdomain of elasticbeanstalk.com**. When you create an environment, you can choose a unique subdomain that represents your application.
- You can use a shared Application Load Balancer to serve traffic for multiple applications running on multiple Elastic Beanstalk environments within the same VPC.
- Deployment policies:
 - All at once – deploys the new version to all instances simultaneously and will be out of service for a short time.
 - Rolling – deploys the new version in batches.

- Rolling with additional batch – deploys the new version in batches, but first launch a new batch of instances.
- Immutable – deploys the new version to a new set of instances.
- Traffic splitting – deploys the new version to a new set of instances and temporarily split incoming client traffic.
- The connections between your application's component environments can be specified as named references using environment links.
- You can rebuild terminated environments within six weeks of their termination with the same name, ID, and configuration.

AWS Elastic Beanstalk Monitoring

- Elastic Beanstalk Monitoring console displays your environment's status and application health at a glance.
- Elastic Beanstalk reports the health of a web server environment depending on how the application running in it responds to the health check.
- **Enhanced health reporting** is a feature that you can enable on your environment to allow AWS Elastic Beanstalk to gather additional information about resources in your environment. Elastic Beanstalk analyzes the information gathered to provide a better picture of overall environment health and aid in the identification of issues that can cause your application to become unavailable.
- You can create alarms for metrics to help you monitor changes to your environment so that you can easily identify and mitigate problems before they occur.
- EC2 instances in your Elastic Beanstalk environment generate logs that you can view to troubleshoot issues with your application or configuration files.

AWS Elastic Beanstalk Security

- When you create an environment, Elastic Beanstalk prompts you to provide two AWS IAM roles: a **service role** and an **instance profile**.
 - Service Roles – assumed by Elastic Beanstalk to use other AWS services on your behalf.
 - Instance Profiles – applied to the instances in your environment and allows them to retrieve application versions from S3, upload logs to S3, and perform other tasks that vary depending on the environment type and platform.
- User Policies – allow users to create and manage Elastic Beanstalk applications and environments.

AWS Elastic Beanstalk Pricing

- There is no additional charge for Elastic Beanstalk. You pay only for the underlying AWS resources that your application consumes.

[AWS Elastic Beanstalk Tutorials on YouTube:](#)

<https://www.youtube.com/user/AmazonWebServices/search?query=elastic+beanstalk>