



# Elastic Beanstalk





# Table of Contents

- ▶ Introduction to Elastic Beanstalk
- ▶ Basic concepts of Elastic Beanstalk



1

# Introduction to Elastic Beanstalk



# Introduction to Elastic Beanstalk

## What is Elastic Beanstalk ?





- AWS Elastic Beanstalk is an easy-to-use service for **deploying and scaling web applications and services**.
- It is a kind of orchestration service offered by Amazon Web Services used to set up your application architecture.

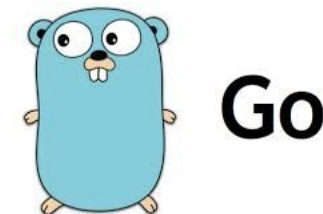
# Introduction to Elastic Beanstalk

## What is Elastic Beanstalk ?

On-site	IaaS	PaaS	SaaS
Applications	Applications	Applications	Applications
Data	Data	Data	Data
Runtime	Runtime	Runtime	Runtime
Middleware	Middleware	Middleware	Middleware
O/S	O/S	O/S	O/S
Virtualization	Virtualization	Virtualization	Virtualization
Servers	Servers	Servers	Servers
Storage	Storage	Storage	Storage
Networking	Networking	Networking	Networking

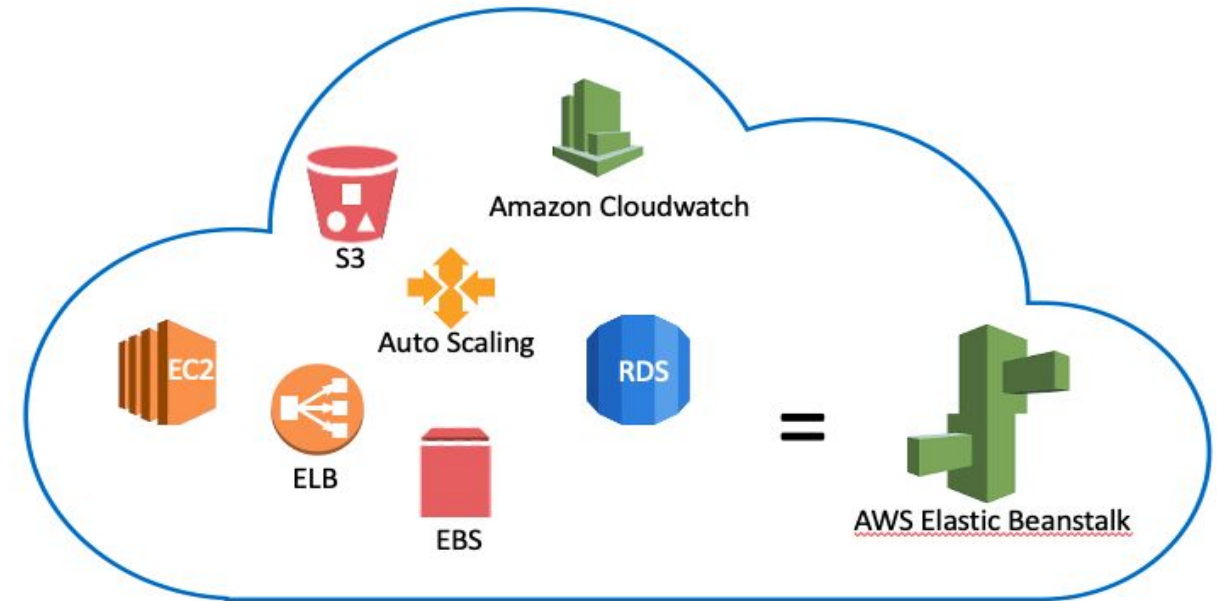
 You manage

 Service provider manages



# Introduction to Elastic Beanstalk

## Why AWS Elastic Beanstalk?



- Automates the details of capacity provisioning,
- Load balancing,
- Auto scaling,
- Application deployment,

# Introduction to Elastic Beanstalk

## Why AWS Elastic Beanstalk?



Monitoring

- Elastic Beanstalk automates management tasks—such as

- Monitoring,
- Application version deployment,
- A basic health check—and facilitates log file access.



Health Check

- By using Elastic Beanstalk, developers can focus on developing their application and are freed from deployment-oriented tasks, such as provisioning servers, setting up load balancing, or managing scaling.



2

# Basic Concepts of Elastic Beanstalk



# Basic Concepts of Elastic Beanstalk

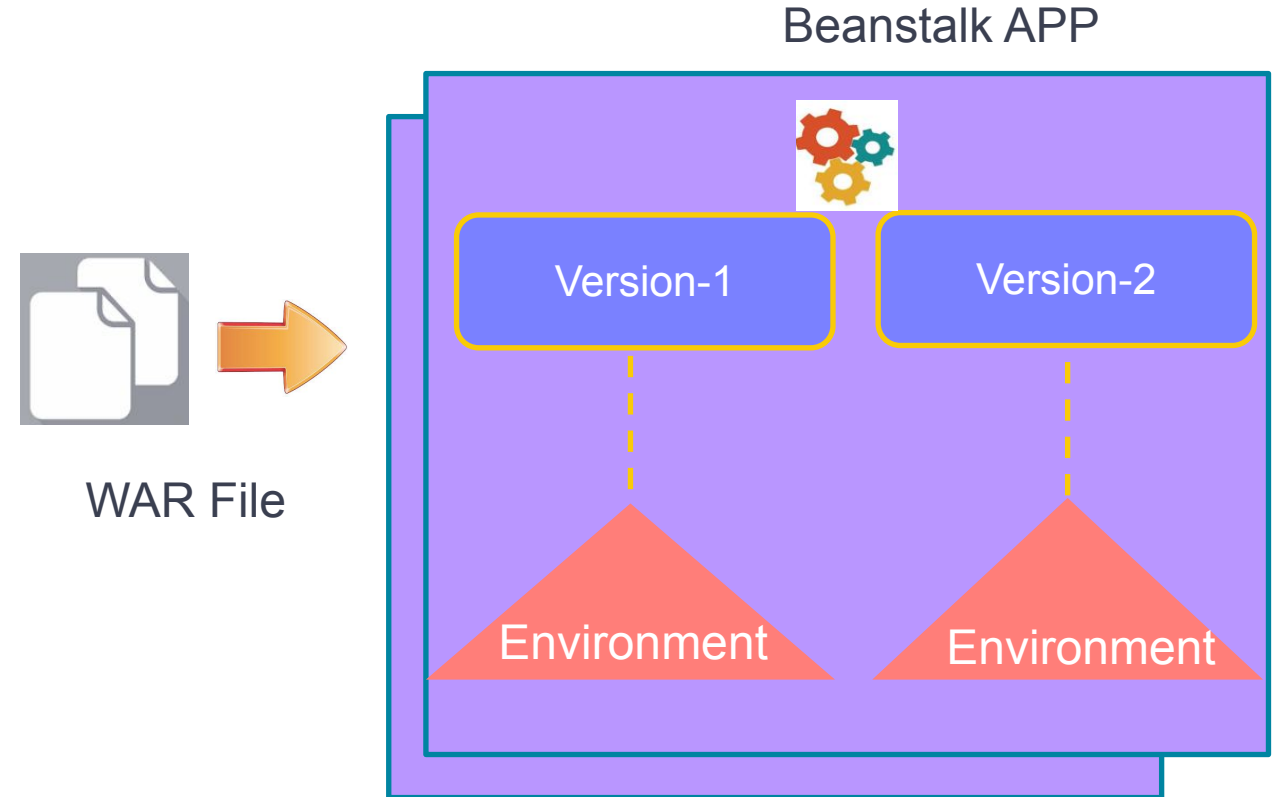


## Application

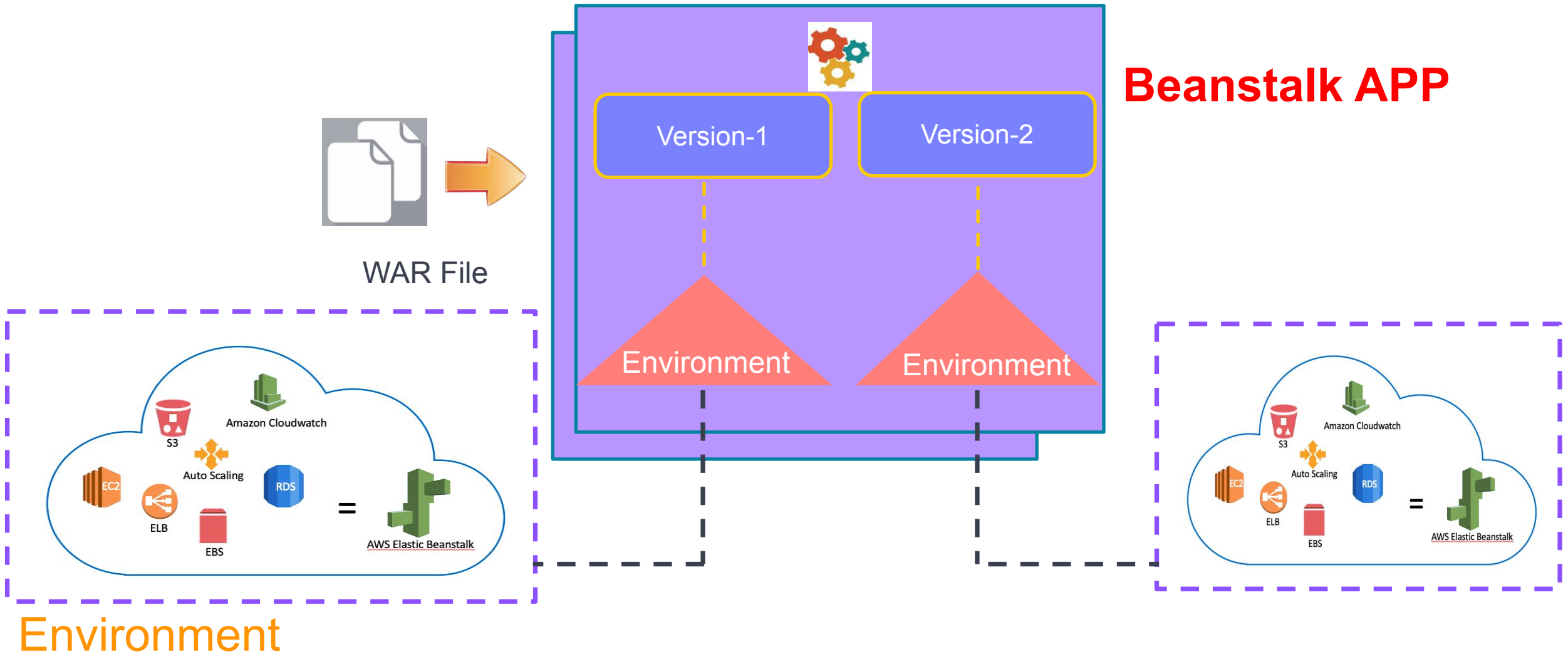
- An Elastic Beanstalk **application** is a **logical collection** of Elastic Beanstalk components, including **environments, versions, and environment configurations**. In Elastic Beanstalk an application is conceptually similar to a folder.

## Application version

- 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. Applications can have many versions and each application version is unique.



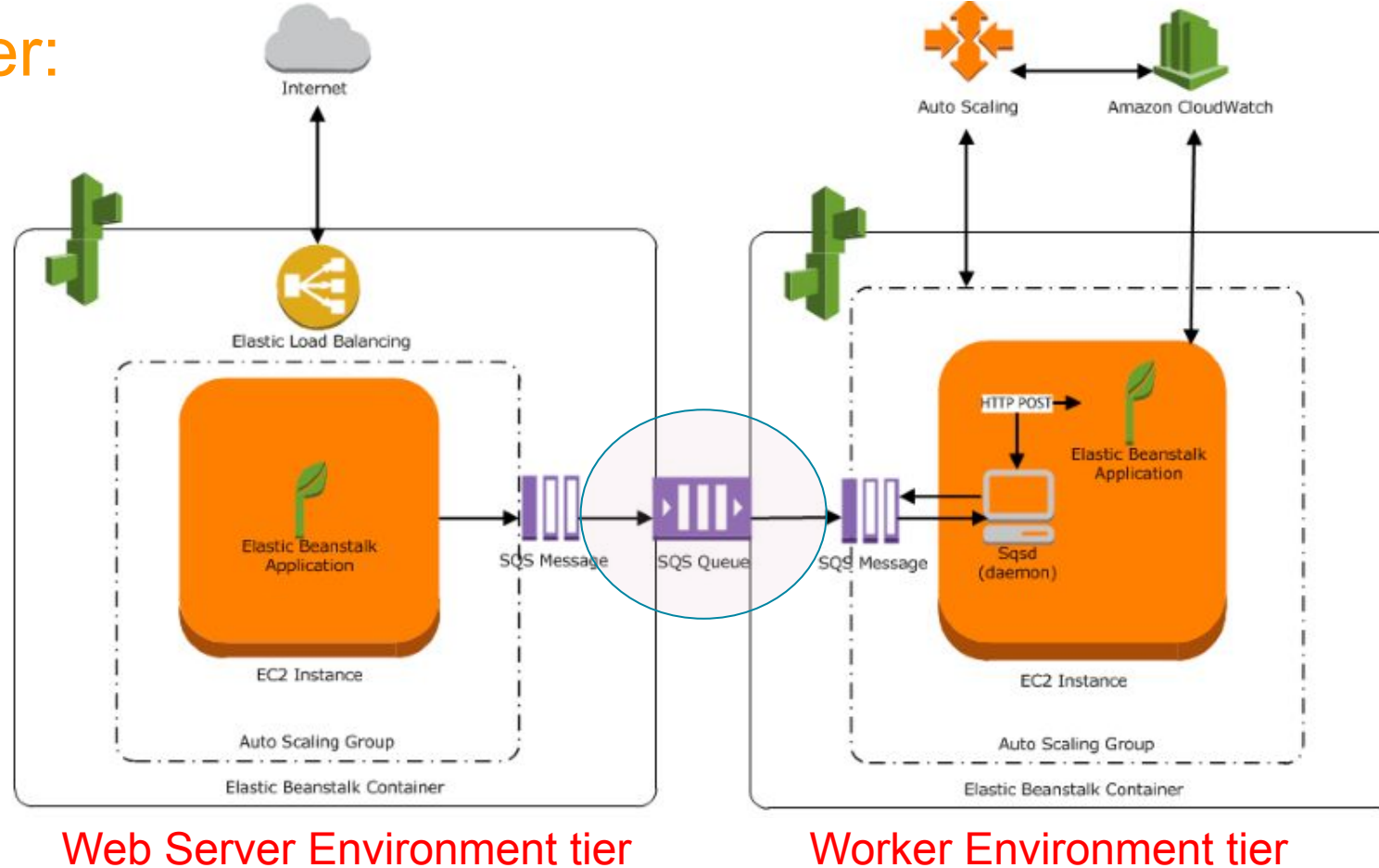
# Basic Concepts of Elastic Beanstalk



- An environment is **a collection of AWS resources** running an application version. Each environment runs only one application version at a time.

# Basic Concepts of Elastic Beanstalk

## Environment Tier:



Web Server Environment tier

Worker Environment tier

- The environment tier designates the type of application that the environment runs, and determines what resources Elastic Beanstalk provisions to support it. An application that **serves HTTP requests runs in a Web Server Environment tier**. An environment that pulls tasks from an **Amazon Simple Queue Service (Amazon SQS) queue runs in a Worker Environment tier**

# Basic Concepts of Elastic Beanstalk



## Platform:

Platform
Platform
Java ▼
Platform branch
Corretto 11 running on 64bit Amazon Linux 2 ▼
Platform version
3.0.3 (Recommended) ▼

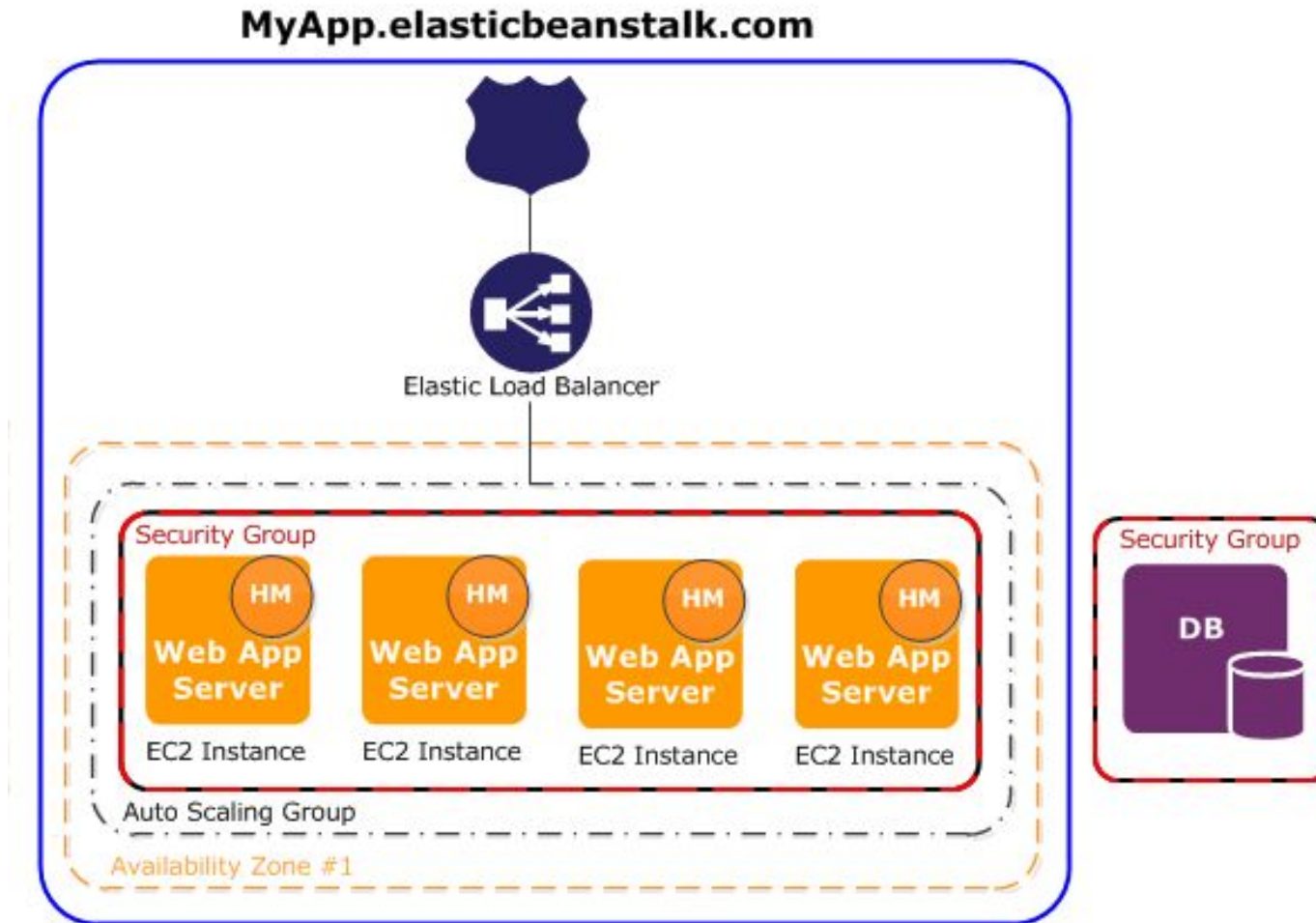
### Supported platform versions

- Docker
- Go
- Java SE
- Tomcat
- .NET Core on Linux
- .NET on Windows Server
- Node.js
- PHP
- Python
- Ruby

- A platform is a **combination of an operating system, programming language runtime, web server, application server, and Elastic Beanstalk components**. You design and target your web application to a platform. Elastic Beanstalk provides a variety of platforms on which you can build your applications.

# Basic Concepts of Elastic Beanstalk

## Environment Tier:

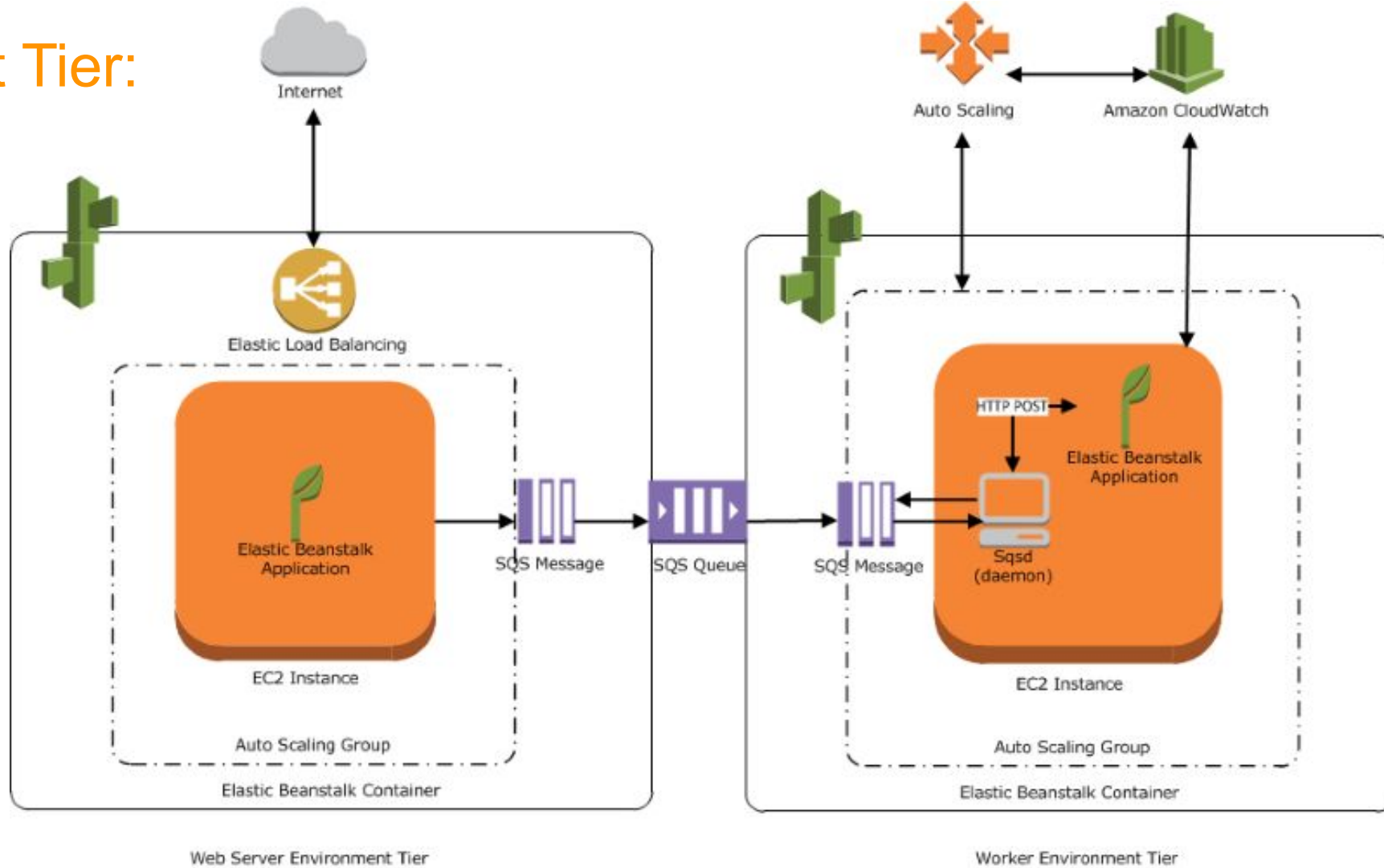


Web Server Environment tier



# Basic Concepts of Elastic Beanstalk

## Environment Tier:



## Worker Environment tier

# Basic Concepts of Elastic Beanstalk



## EB Deployment Strategies:

- **All at once** – Deploy the new version to all instances simultaneously. All instances in your environment are out of service for a short time while the deployment occurs.
- **Rolling** – Deploy the new version in batches. Each batch is taken out of service during the deployment phase, reducing your environment's capacity by the number of instances in a batch.
- **Rolling with additional batch** – Deploy the new version in batches, but first launch a new batch of instances to ensure full capacity during the deployment process.
- **Immutable** – Deploy the new version to a fresh group of instances by performing an immutable update.
- **Traffic splitting** – Deploy the new version to a fresh group of instances and temporarily split incoming client traffic between the existing application version and the new one.



# Basic Concepts of Elastic Beanstalk



## EB CLI:

```
guile@LAPTOP-O4O3C63R: ~  
guile@LAPTOP-O4O3C63R:~$ eb --version  
EB CLI 3.19.0 (Python 3.8.1)  
guile@LAPTOP-O4O3C63R:~$ eb  
usage: eb (sub-commands ...) [options ...] {arguments ...}  
  
Welcome to the Elastic Beanstalk Command Line Interface (EB CLI).  
For more information on a specific command, type "eb {cmd} --help".  
  
commands:  
  abort          Cancels an environment update or deployment.  
  appversion     Listing and managing application versions  
  clone          Clones an environment.  
  codesource     Configures the code source for the EB CLI to use by default.  
  config         Modify an environment's configuration. Use subcommands to manage saved configurations.  
  console        Opens the environment in the AWS Elastic Beanstalk Management Console.  
  create         Creates a new environment.  
  deploy         Deploys your source code to the environment.  
  events         Gets recent events.  
  health         Shows detailed environment health.  
  init           Initializes your directory with the EB CLI. Creates the application.  
  labs           Extra experimental commands.  
  list           Lists all environments.
```





# Elastic Beanstalk

Let's get our hands dirty!

- Creating Application



# THANKS!

## Any questions

