Elastic Beanstalk

Table of Contents

1 Introduction to Elastic Beanstalk

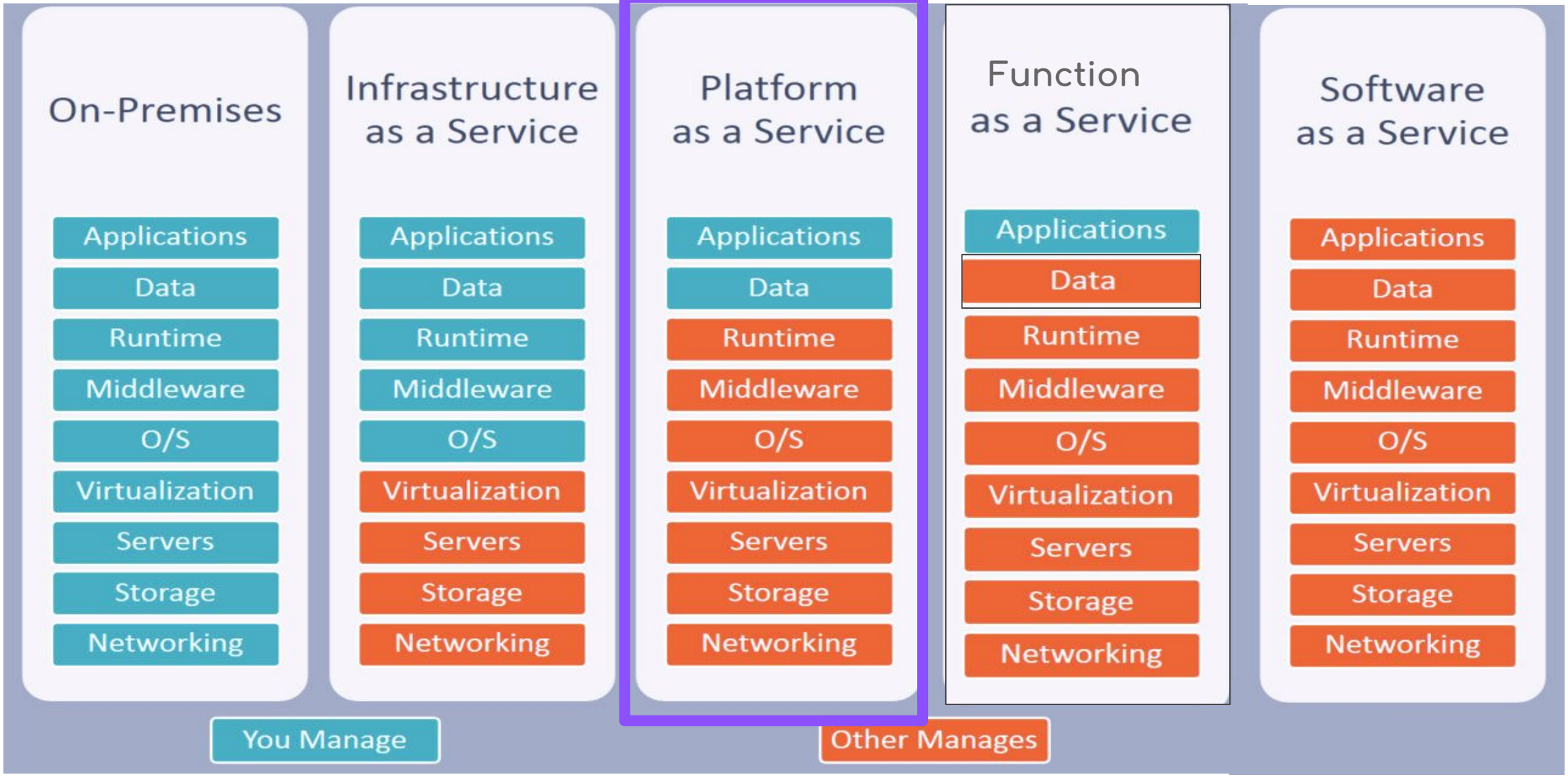
Introduction to Elastic Beanstalk

What is Elastic Beanstalk ?

- AWS Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services.
- In simple terms it is platform as a service (PaaS) which takes our application code and deploys it while provisioning the supporting architecture and compute resources required for our code to run. Elastic Beanstalk also fully manages the patching and security updates for those provisioned resources
- It is a kind of orchestration service offered by Amazon Web Services used to set up your application architecture.

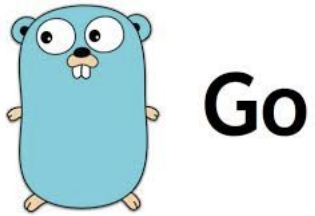
Introduction to Elastic Beanstalk

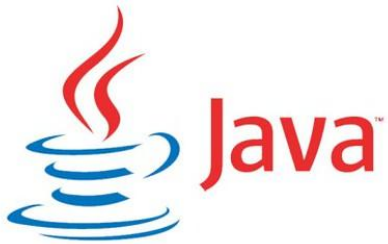
What is Elastic Beanstalk ?



Introduction to Elastic Beanstalk

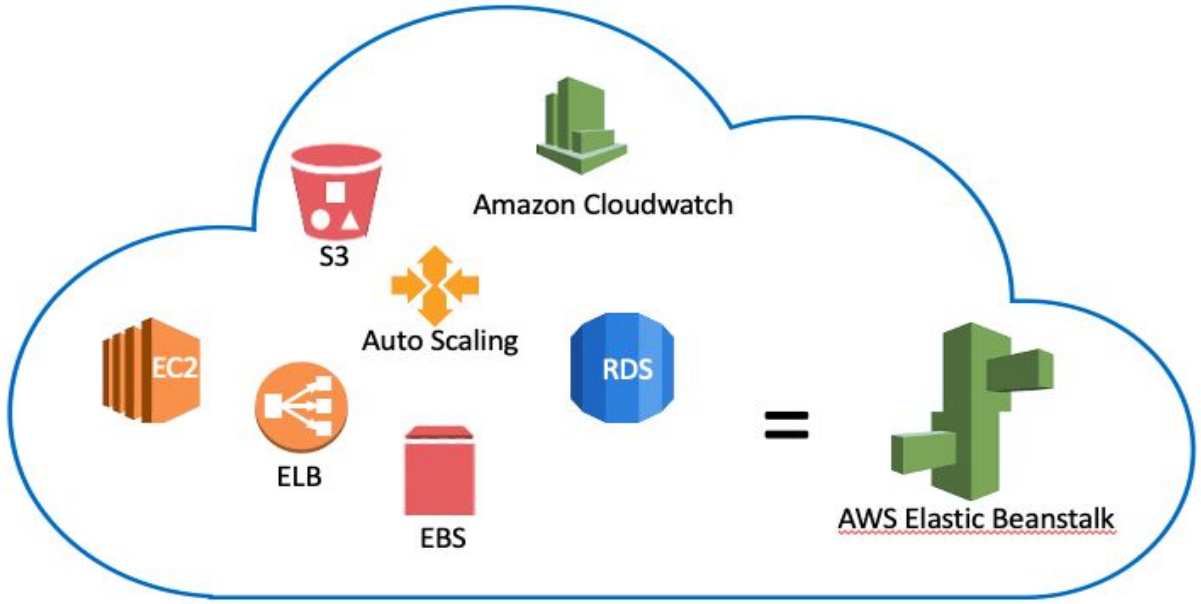
What is Elastic Beanstalk ?





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?

- Automates management tasks:

  - Monitoring,
  - Version deployment,
  - Health check
  - Log

Monitoring

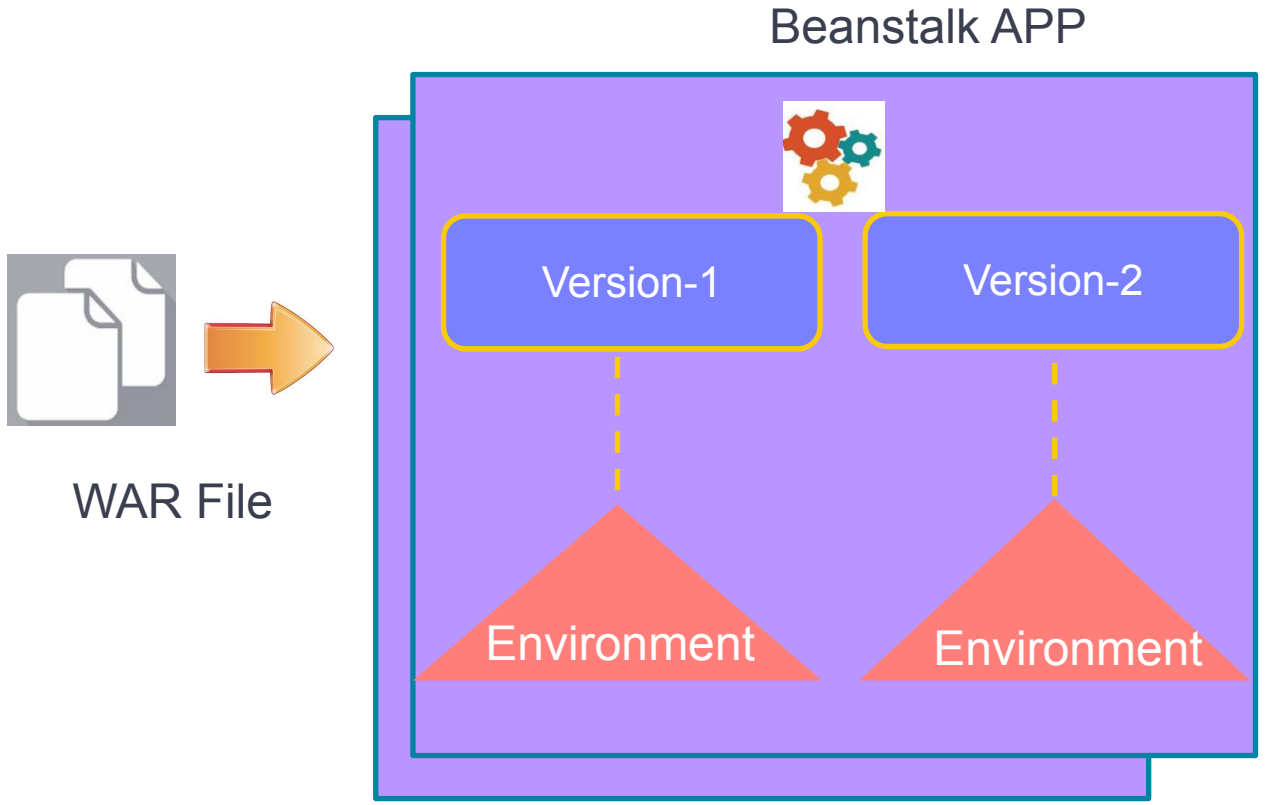
Health Check

2 Basic Concepts of Elastic Beanstalk

Basic Concepts of Elastic Beanstalk

Application

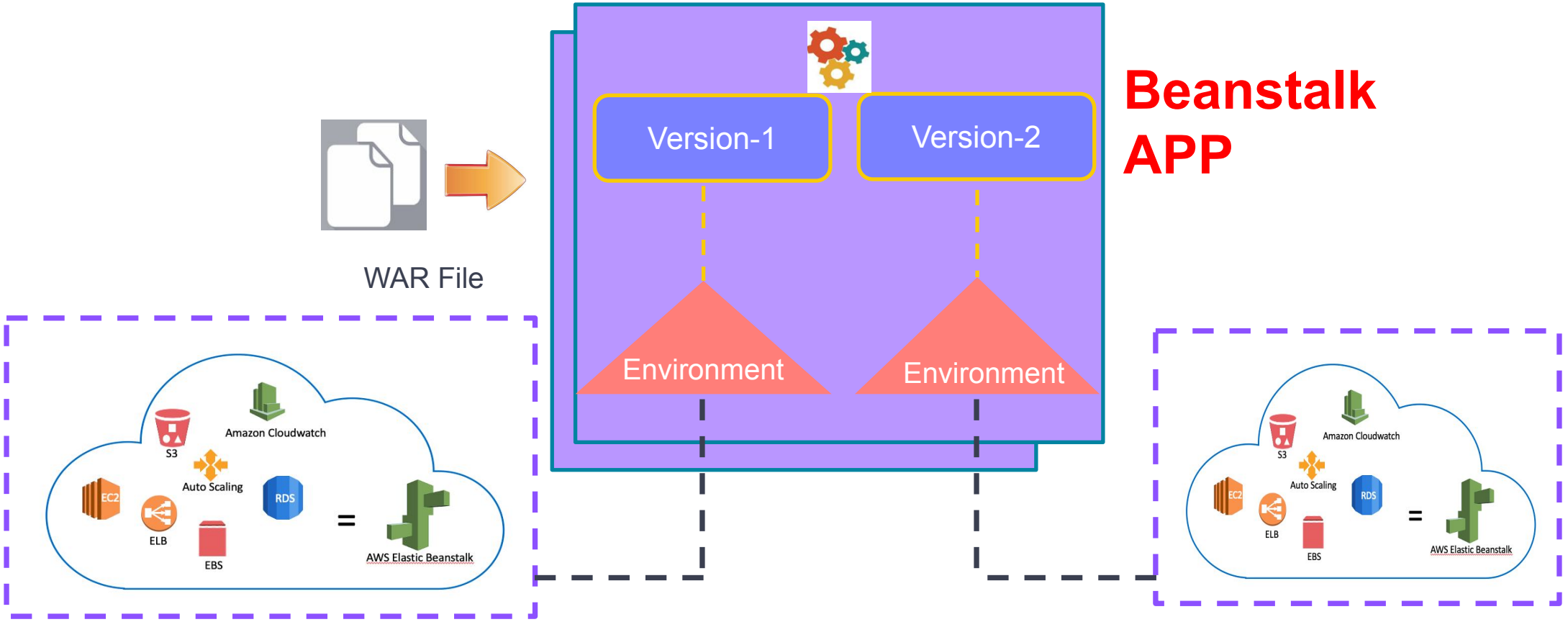
- An Elastic Beanstalk **application is a logical collection** of Elastic Beanstalk components, including **environments, versions, and environment configurations**. In Elastic Beanstalk an application is conceptually similar to a folder.



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

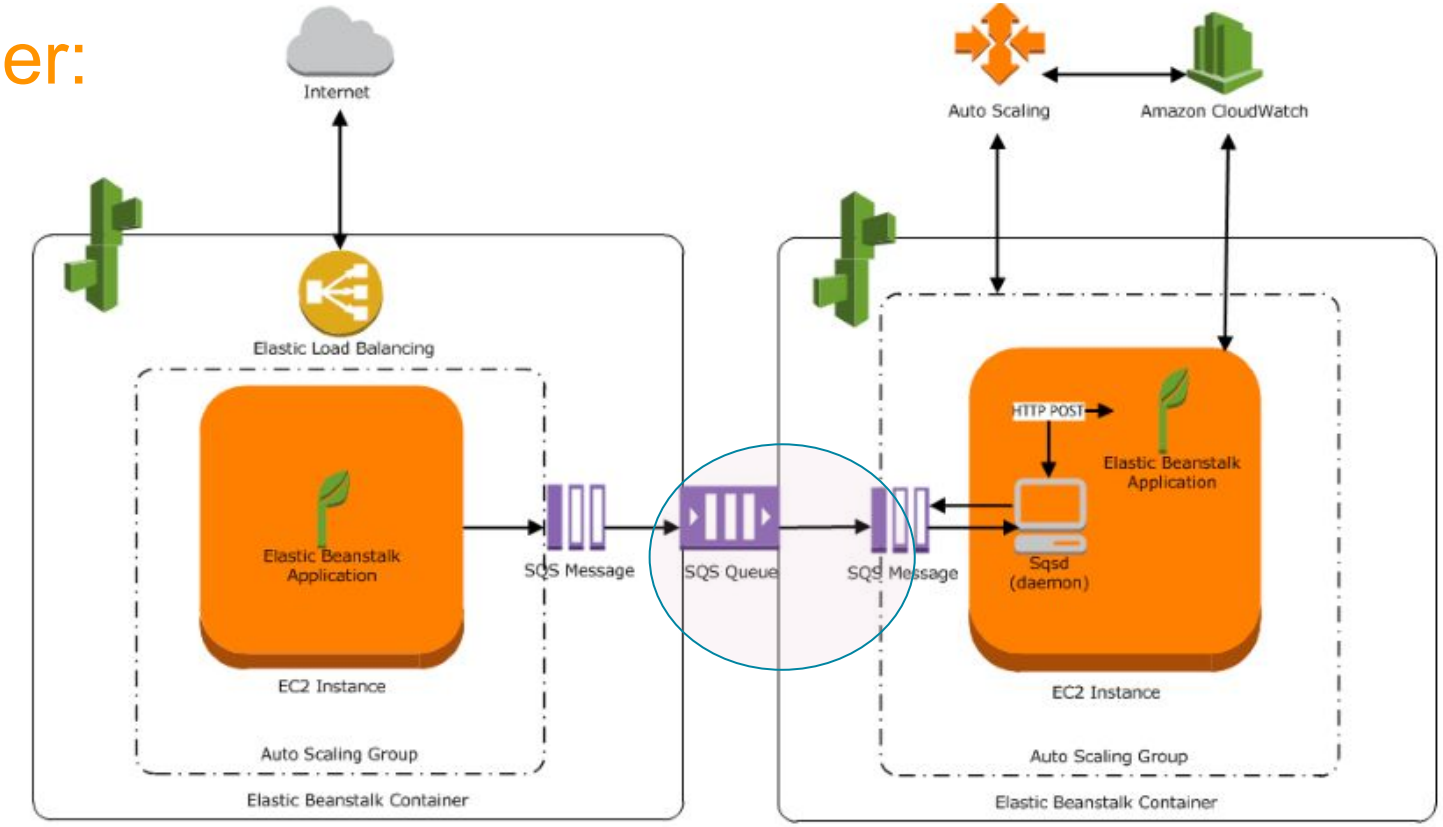


Environment

- 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.

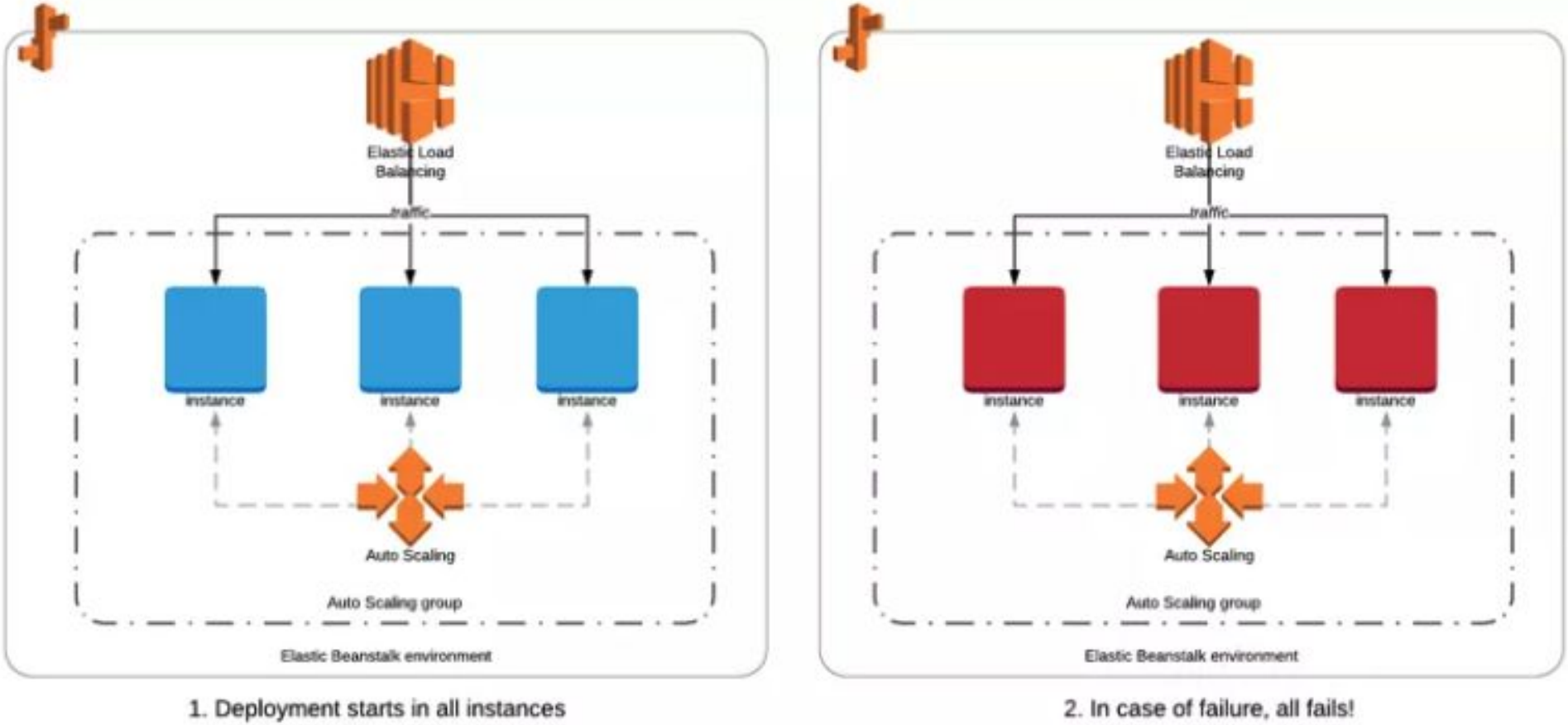
Basic Concepts of Elastic Beanstalk

Platform:

Supported platform versions

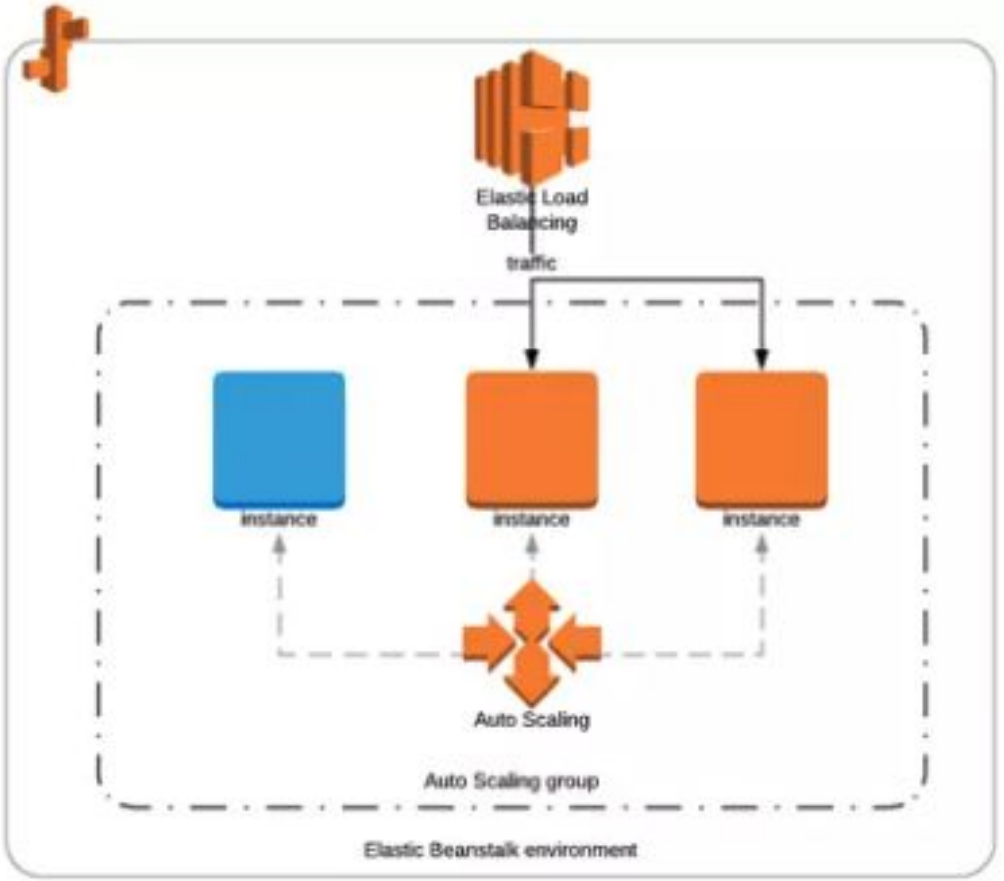
- Docker
- Multicontainer Docker
- Preconfigured Docker
- Go
- Java SE
- Tomcat
- .NET Core on Linux
- .NET on Windows Server
- Node.js
- PHP
- Python
- Ruby

Deployment Models - All at once

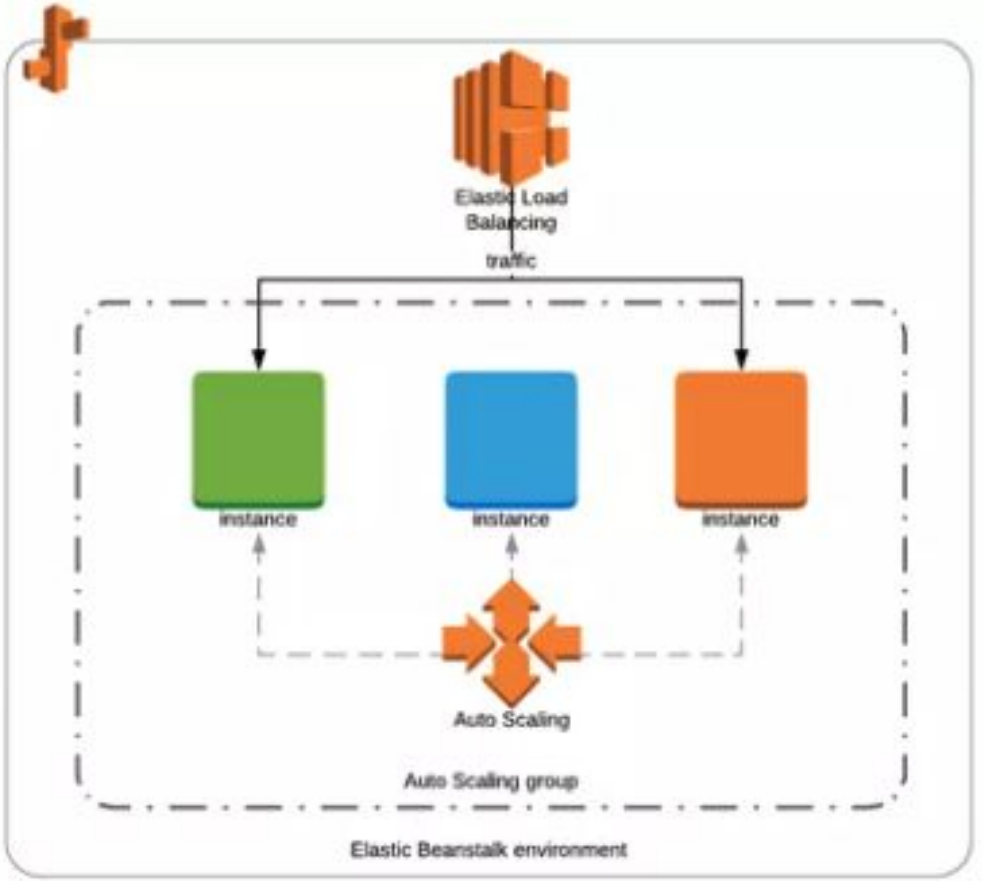


All at once – Deploy the new version to all instances simultaneously. All instances in your environment are out of service for a short time while the deployment occurs.

Deployment Models - Rolling



1. Deployment starts in the first batch.

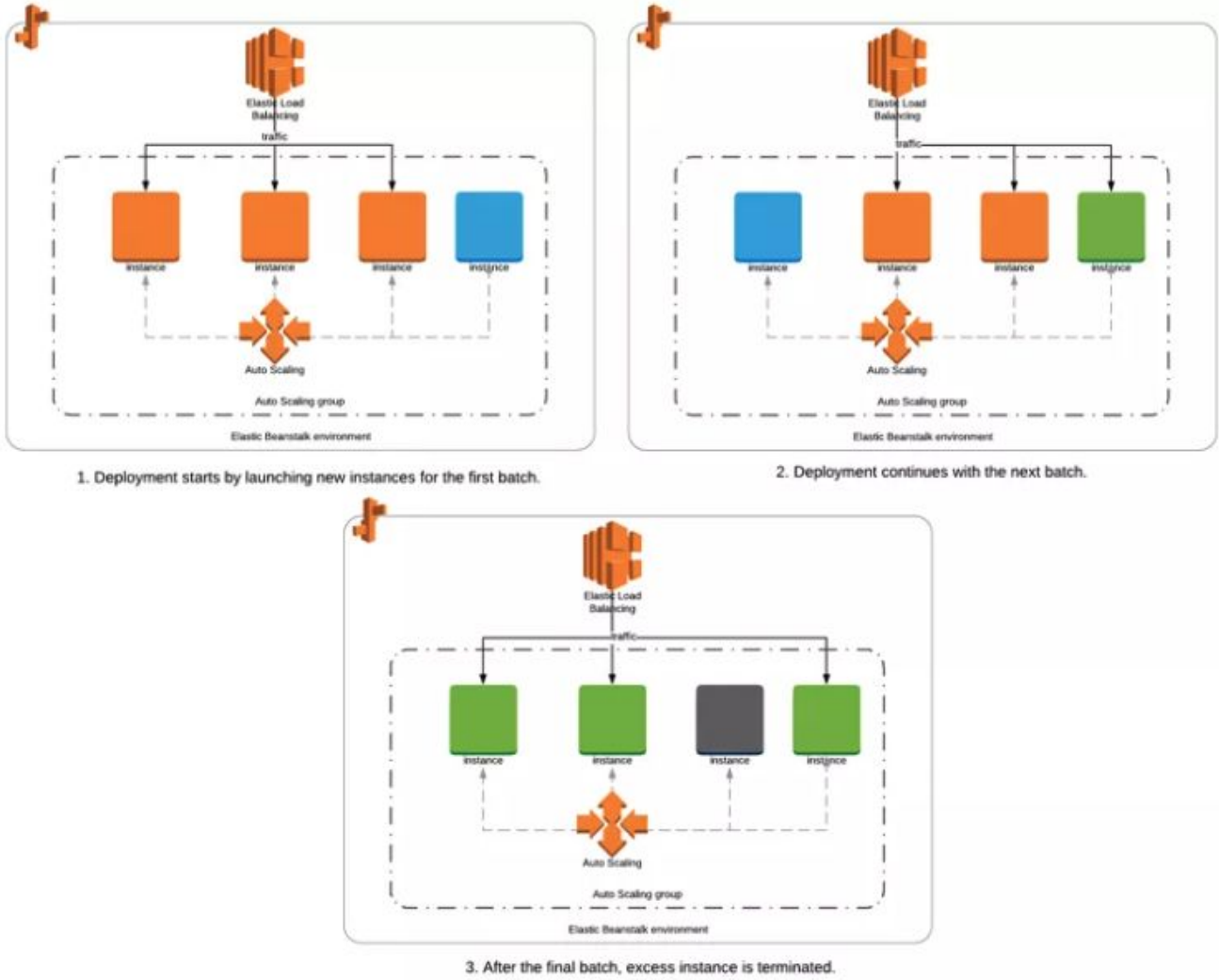


2. Deployment continues with the next batch.

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.

Deployment Models - Rolling with additional 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.

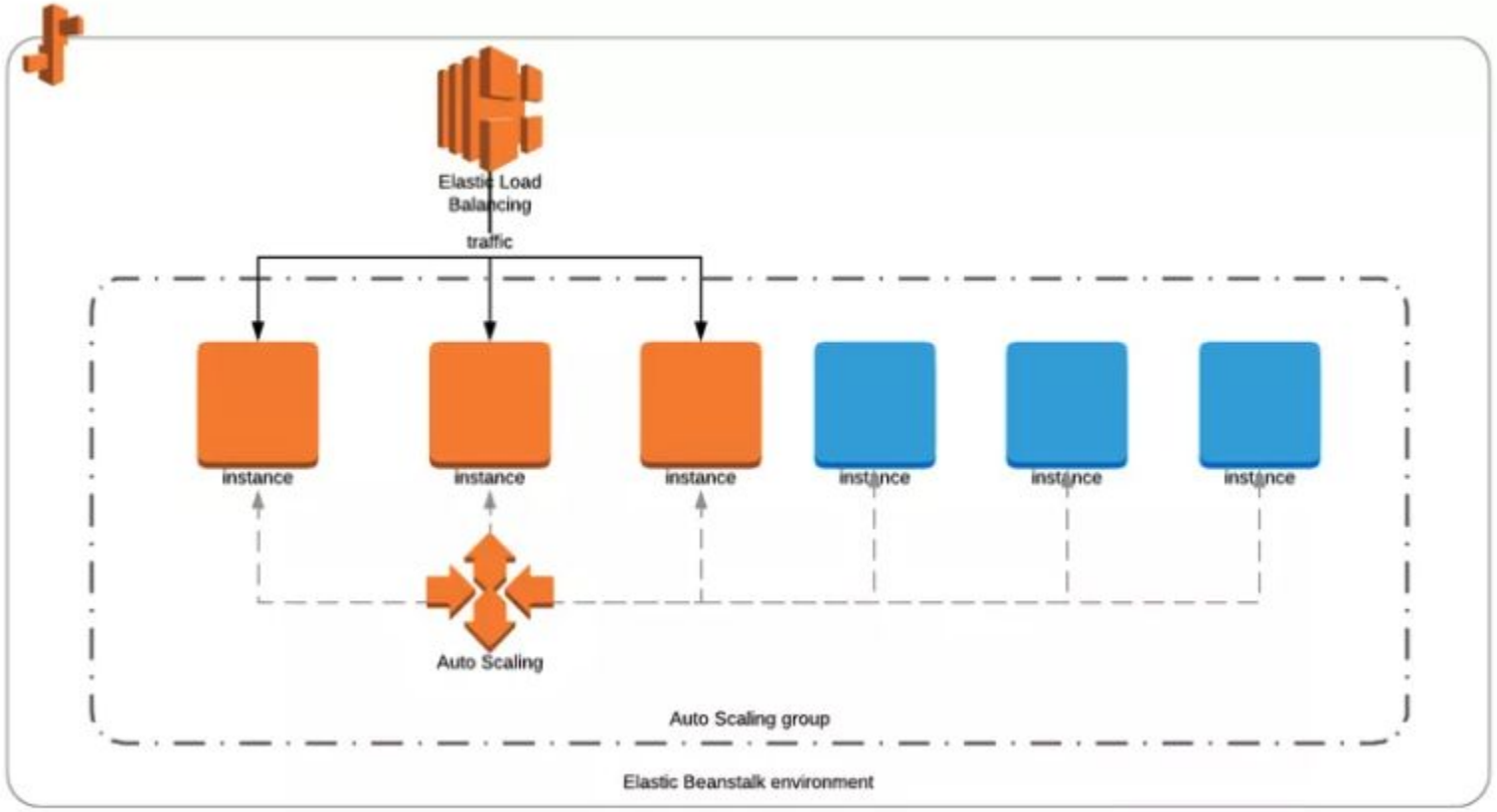


1. Deployment starts by launching new instances for the first batch.

2. Deployment continues with the next batch.

3. After the final batch, excess instance is terminated.

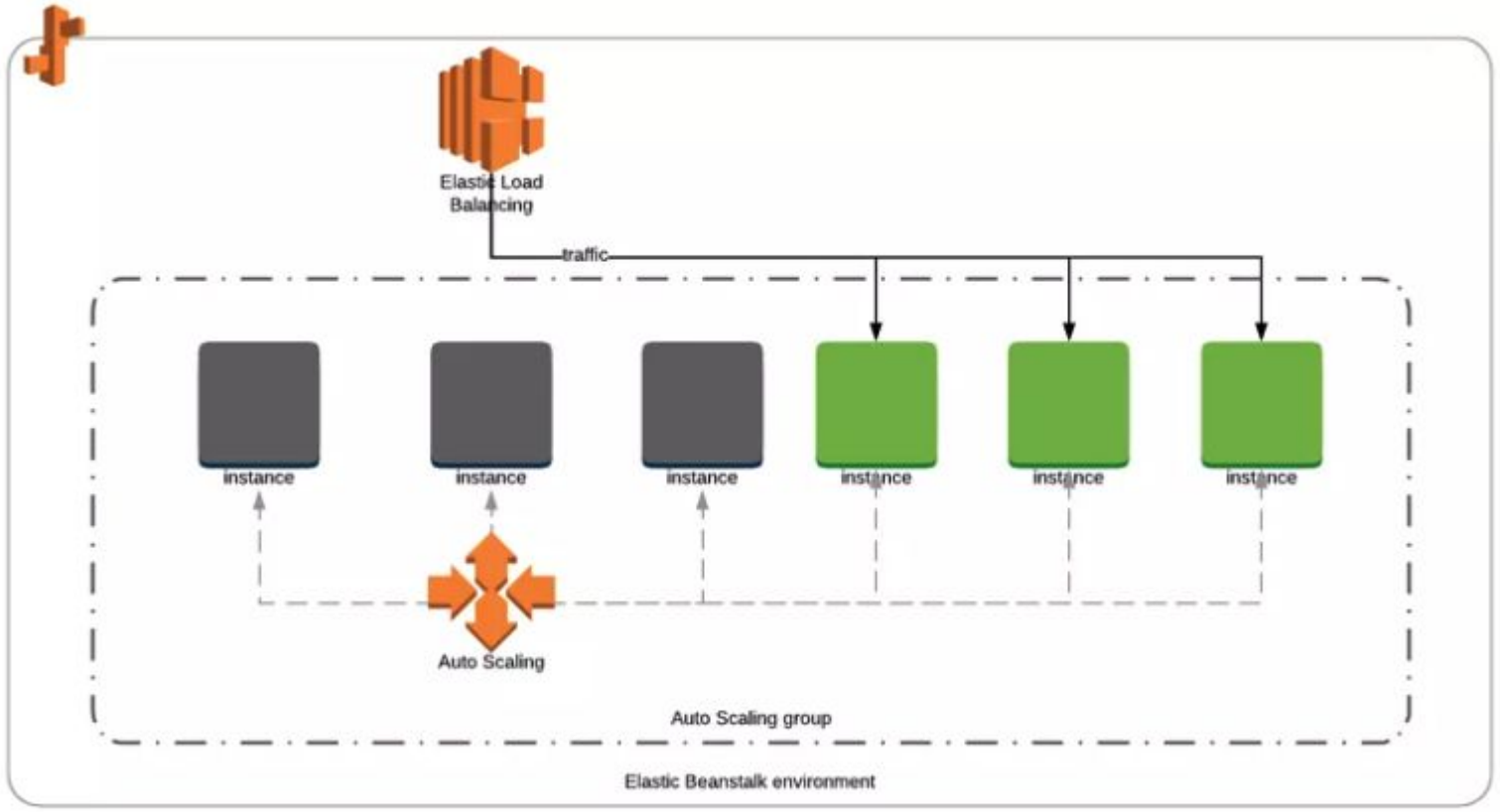
Deployment Models - Immutable



1. Deployment starts by duplicating the instances and deploying the app in new instances

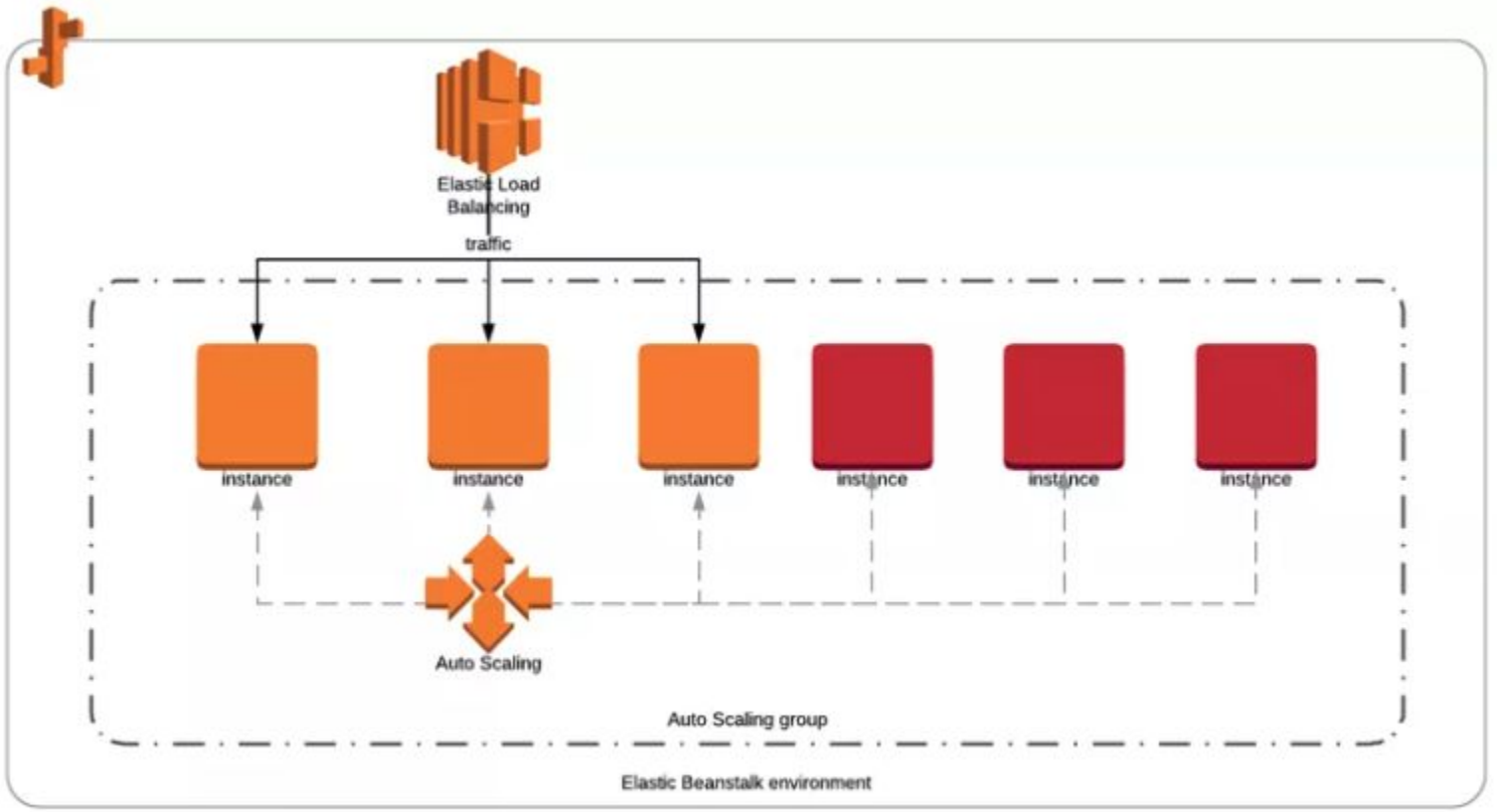
Immutable – Deploy the new version to a fresh group of instances by performing an immutable update.

Deployment Models - Immutable



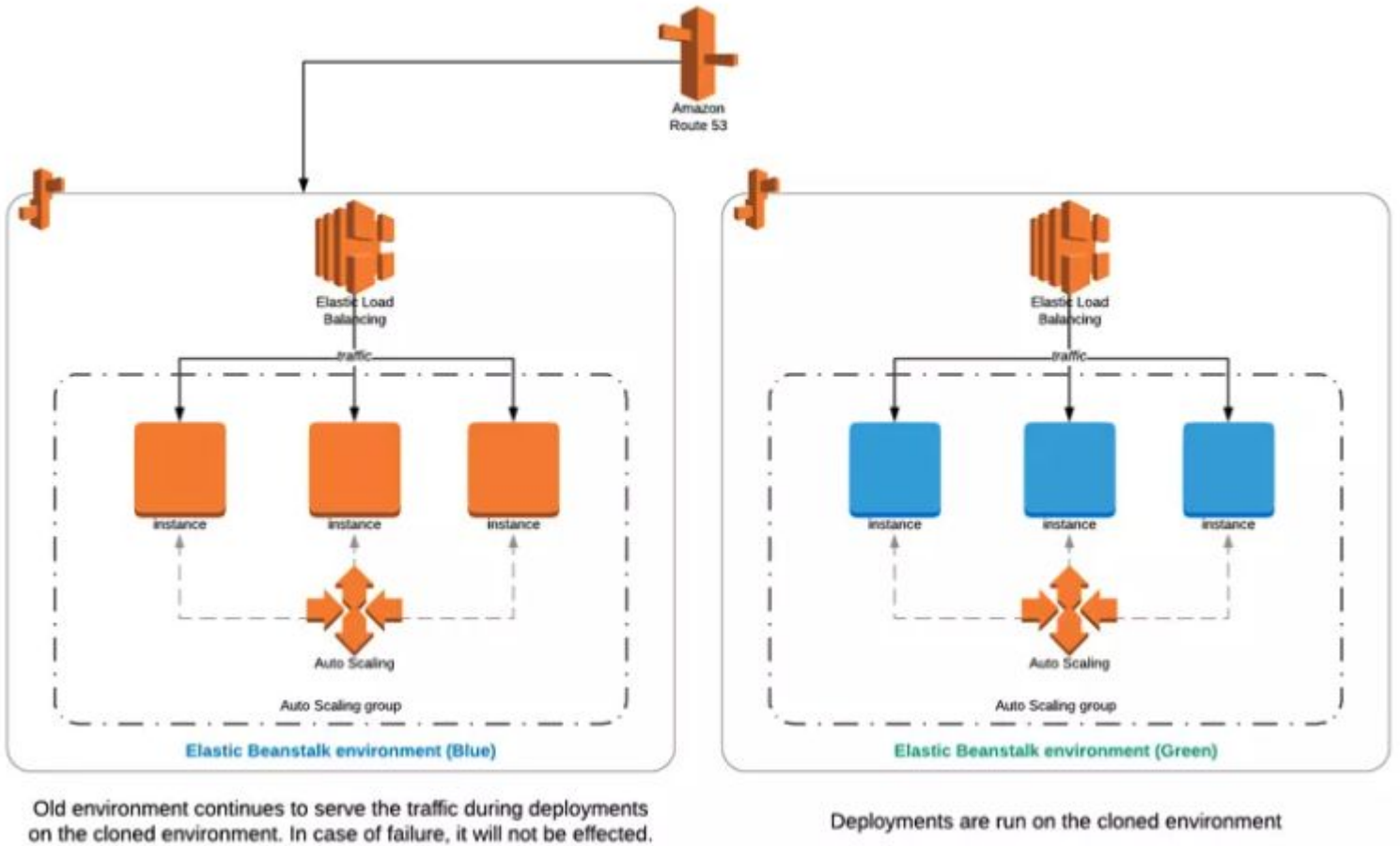
2. After deployment succeeds, older instances are terminated.

Deployment Models - Immutable



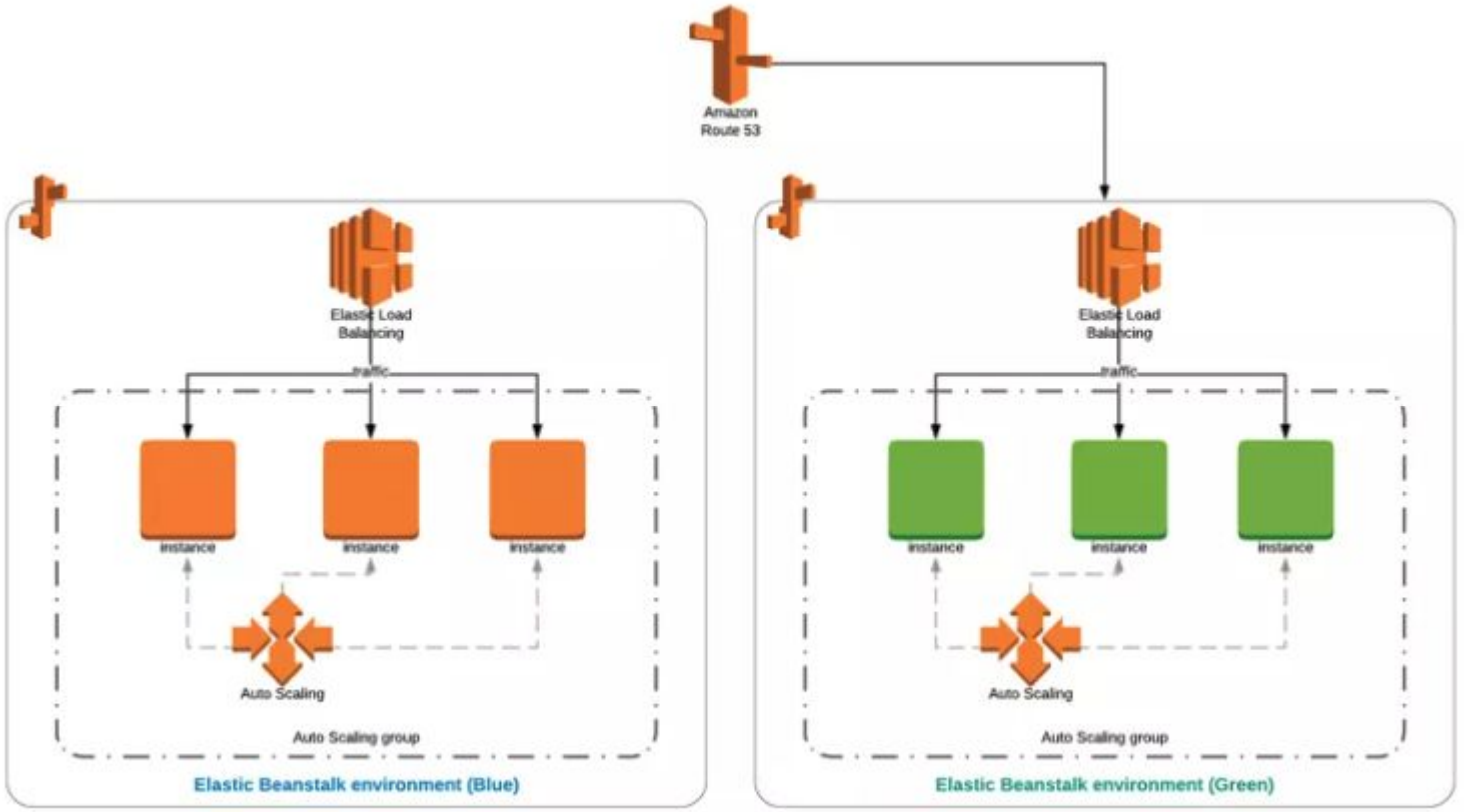
In case of failure, the new instances is terminated and traffic is not effected.

Deployment Models - Blue/Green



Blue/Green or Traffic splitting –
Blue/Green deployments replicate your current environment (blue), deploy the new application to your new, cloned environment (green), and redirect the traffic to the green one after deployment.
If the deployment fails, you terminate the green environment, and nothing will be affected.
If something goes wrong after deployment, for example, your users experience a problem in the new version, you can simply redirect the traffic back to the old version. Hence, it would be wise to keep the old environment running until you verify that the deployment is successful and DNS propagation is completed.

Deployment Models - Blue/Green



After your new environment is ready, you deploy the new version on this environment and verify that the deployment is successful.

Once you are sure that everything is fine, you swap the URLs of the two environments using AWS Management Console, AWS CLI, or EB CLI. Again, Elastic Beanstalk provides a specific action for this. Then, the traffic will start to flow to your new environment after the DNS propagation completes.

Summary of Terms / Concepts

Concept	What it Means
Application	Logical collection of Elastic Beanstalk components required for a working deployment
Application Version	A labelled version of an application (e.g. 1.0, 1.1, 2.0, etc…)
Environment	A set of AWS resources running a specific application version (e.g. DEV, TEST, PROD)
Environment Tier	The type of application that an environment runs (either Web or Worker)
Platform	Combination of OS, programming language, web server - i.e. the “technology stack”

Basic Concepts of Elastic Beanstalk

Elastic Beanstalk Command Line Interface (EB CLI)


```
user@clarusway-MacBook~ % eb --version
EB CLI 3.19.4 (Python 3.9.4)
```

Elastic Beanstalk

Let's get our hands dirty!

- Creating Application

THANKS!

Any questions?

You can find me at:

- @sumod
- sumod@clarusway.com