DevOps

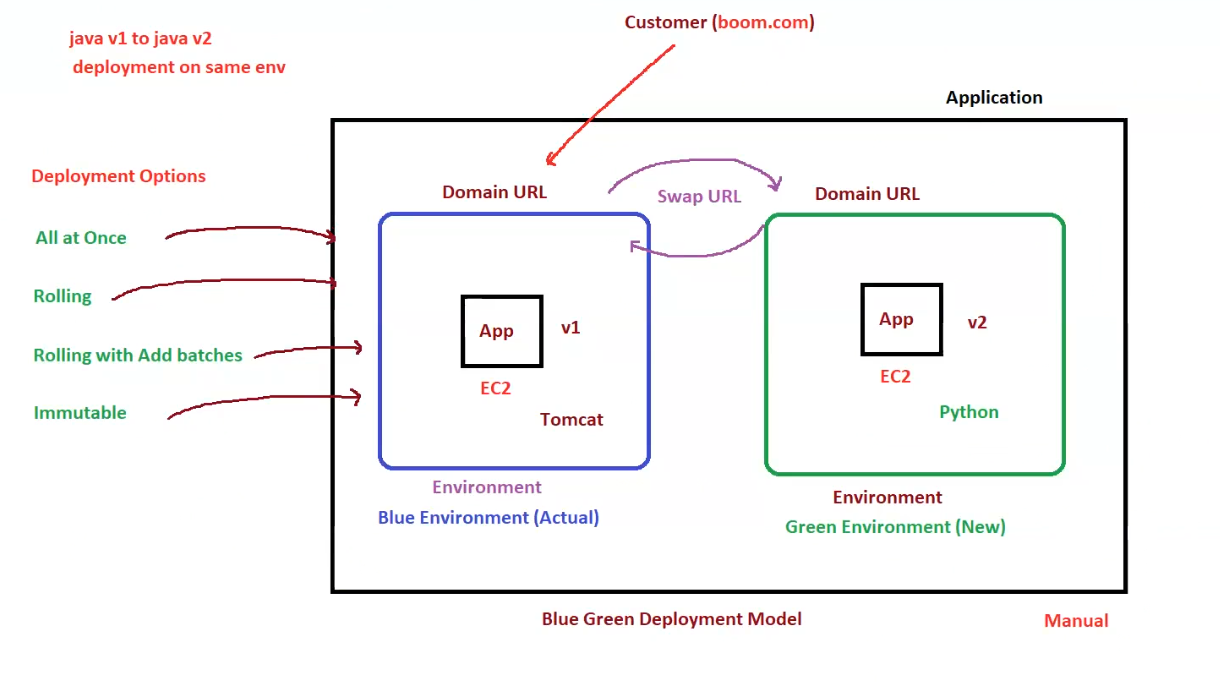
[Blue-Green Deployment Model 2](#_Toc193571589)

[Using Elastic Bean Stalk 2](#_Toc193571590)

[Using Ec2 4](#_Toc193571591)

# Blue-Green Deployment Model

## Using Elastic Bean Stalk



**1. Overview**

* **Blue-Green Deployment** is a strategy used for **zero-downtime** deployments.
* Involves two environments:
  + **Blue Environment (Current/Actual)**: Running **version 1 (v1)** of the application.
  + **Green Environment (New)**: Running **version 2 (v2)** of the application.
* **Domain URL Swap** enables seamless transition from the old to the new version.

**2. Deployment Flow**

1. **Current Setup (Blue Environment)**
   * Runs **Java v1** application on **Tomcat server**.
   * Associated with a **Domain URL** accessed by customers (e.g., boom.com).
2. **New Deployment (Green Environment)**
   * Deploys **Java v2** (or another version) on **Python-based EC2 instance**.
   * Tested separately before switching traffic.
3. **Swap URL Process**
   * Once **v2 is verified**, the **Domain URL is swapped** from Blue to Green.
   * The **Green environment becomes active**, and Blue can be retired or used for rollback.

**3. Deployment Options**

Elastic Beanstalk provides multiple deployment strategies to minimize downtime:

| **Deployment Option** | **Downtime** | **Description** |
| --- | --- | --- |
| **All at Once** | ❌ **Downtime** | Deploys the new version to all instances at once |
| **Rolling** | ✅ **No Downtime** | Updates instances in batches while keeping others live |
| **Rolling with Additional Batches** | ✅ **No Downtime** | Temporarily increases capacity to deploy without downtime |
| **Immutable** | ✅ **No Downtime** | Deploys to new instances before redirecting traffic |

**4. Key Benefits of Blue-Green Deployment**

✅ **Zero Downtime Deployment** – Users always have access to a working version.  
✅ **Rollback Capability** – If the new version fails, switching back to the previous version is easy.  
✅ **Testing Before Production** – The Green environment can be tested before swapping URLs.  
✅ **Improved Stability & Reliability** – Reduces deployment risks by keeping both versions active temporarily.

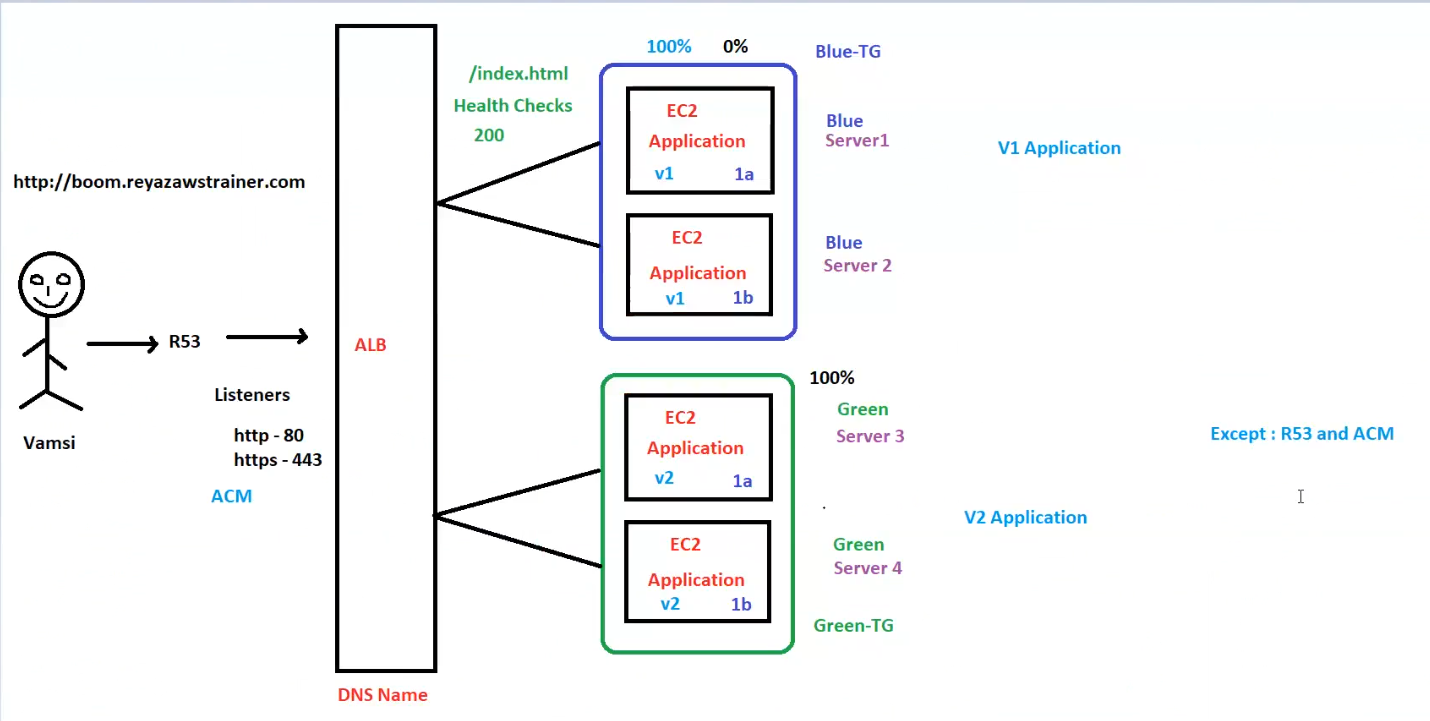
**5. Summary**

| **Environment** | **Purpose** |
| --- | --- |
| **Blue Environment** | Current active environment running v1 (Tomcat) |
| **Green Environment** | New environment with v2 (Python) |
| **Domain URL Swap** | Redirects traffic to Green after successful validation |

**Deployment Model: Blue-Green (Manual)**

* The swap between environments is performed **manually**.
* Ensures controlled deployment with rollback capability.

## Using Ec2



**AWS Blue-Green Deployment Architecture with ALB**

**1. Overview**

* This diagram illustrates **Blue-Green Deployment** using **Application Load Balancer (ALB)**.
* The deployment is for the domain: [**http://boom.reyazawstrainer.com**](http://boom.reyazawstrainer.com/).
* It includes:
  + **Blue Environment (v1 application)**
  + **Green Environment (v2 application)**
  + **Route 53 (R53) for DNS resolution**
  + **Application Load Balancer (ALB) for traffic distribution**
  + **Health Checks to monitor application status**

**2. Architecture Components**

**a) User Request Flow**

1. **User (Vamsi)** accesses http://boom.reyazawstrainer.com.
2. The request is resolved via **Route 53 (R53)**.
3. Route 53 directs traffic to the **Application Load Balancer (ALB)**.
4. ALB has **listeners** on:
   * **HTTP (Port 80)**
   * **HTTPS (Port 443)**
5. ALB distributes traffic to either:
   * **Blue Target Group (TG) → v1 Application**
   * **Green Target Group (TG) → v2 Application**

**b) Load Balancing & Target Groups**

* **ALB distributes traffic** based on target group weights:
  + **Blue-TG (v1 Application)**
    - Includes **Blue Server 1 (1a) and Blue Server 2 (1b)**.
    - Initially receives **100% traffic**.
  + **Green-TG (v2 Application)**
    - Includes **Green Server 3 (1a) and Green Server 4 (1b)**.
    - Becomes active when switching traffic to v2.
* **Traffic Shift Strategy**:
  + Initially, Blue receives **100% traffic**.
  + After deployment, Green can be gradually tested and switched to **100% traffic**.
  + ALB **health checks (/index.html)** ensure the new version is running properly.

**3. Deployment & Version Management**

* **V1 Application (Blue Environment)**
  + Runs on **EC2 instances** in the **Blue Target Group (TG)**.
  + Traffic is initially routed here.
* **V2 Application (Green Environment)**
  + Runs on **EC2 instances** in the **Green Target Group (TG)**.
  + Traffic is shifted here after successful testing.
* **Traffic Shift Process**
  + **Blue Environment starts with 100% traffic.**
  + Once **Green (v2) is tested and stable**, **traffic shifts to 100% on Green**.
  + Blue can be **terminated or retained** for rollback.

**4. Supporting AWS Services**

| **AWS Service** | **Role in Deployment** |
| --- | --- |
| **Route 53 (R53)** | Resolves domain name (boom.reyazawstrainer.com) to ALB |
| **Application Load Balancer (ALB)** | Distributes traffic to Blue/Green environments |
| **Listeners (80, 443)** | Handles HTTP/HTTPS traffic |
| **Health Checks (/index.html)** | Ensures instances are operational |
| **ACM (AWS Certificate Manager)** | Manages SSL/TLS certificates for secure HTTPS |

**5. Key Benefits of Blue-Green Deployment**

✅ **Zero Downtime** – Ensures smooth traffic cutover between versions.  
✅ **Rollback Capability** – Easily revert to previous version if issues occur.  
✅ **Improved Testing** – Green environment is validated before going live.  
✅ **Load Balancer Optimization** – ALB dynamically shifts traffic based on health checks.

**6. Summary**

| **Component** | **Details** |
| --- | --- |
| **User** | Accesses boom.reyazawstrainer.com |
| **Route 53 (R53)** | Resolves DNS to ALB |
| **ALB Listeners** | HTTP (80) / HTTPS (443) |
| **Blue Environment** | v1 Application (Initial Active) |
| **Green Environment** | v2 Application (New Release) |
| **Traffic Shift** | Moves from Blue-TG → Green-TG |