# Production-Level Blue-Green Deployment CI/CD Pipeline

## Scenario

Upgrade an application with new features while ensuring zero downtime by redirecting traffic from the blue to the green deployment. In case of issues, revert changes from Green back to Blue.

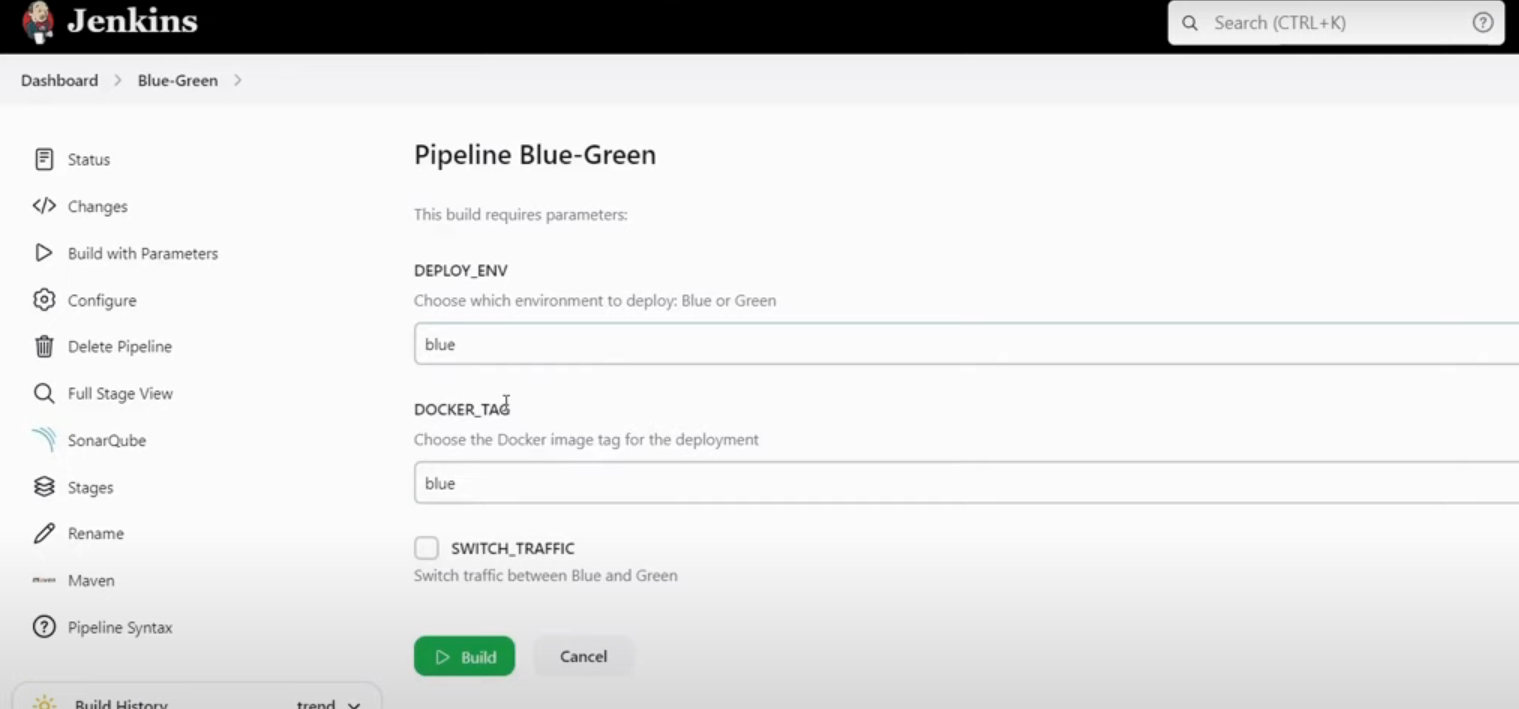
## Tools Used

* **EKS Cluster Deployment using Terraform**
* **SonarQube** (Code Quality and Security Analysis)
* **Nexus** (Artifact Repository)
* **Jenkins** (CI/CD Pipeline)

## Implementation Walkthrough

### Select Deployment Environment

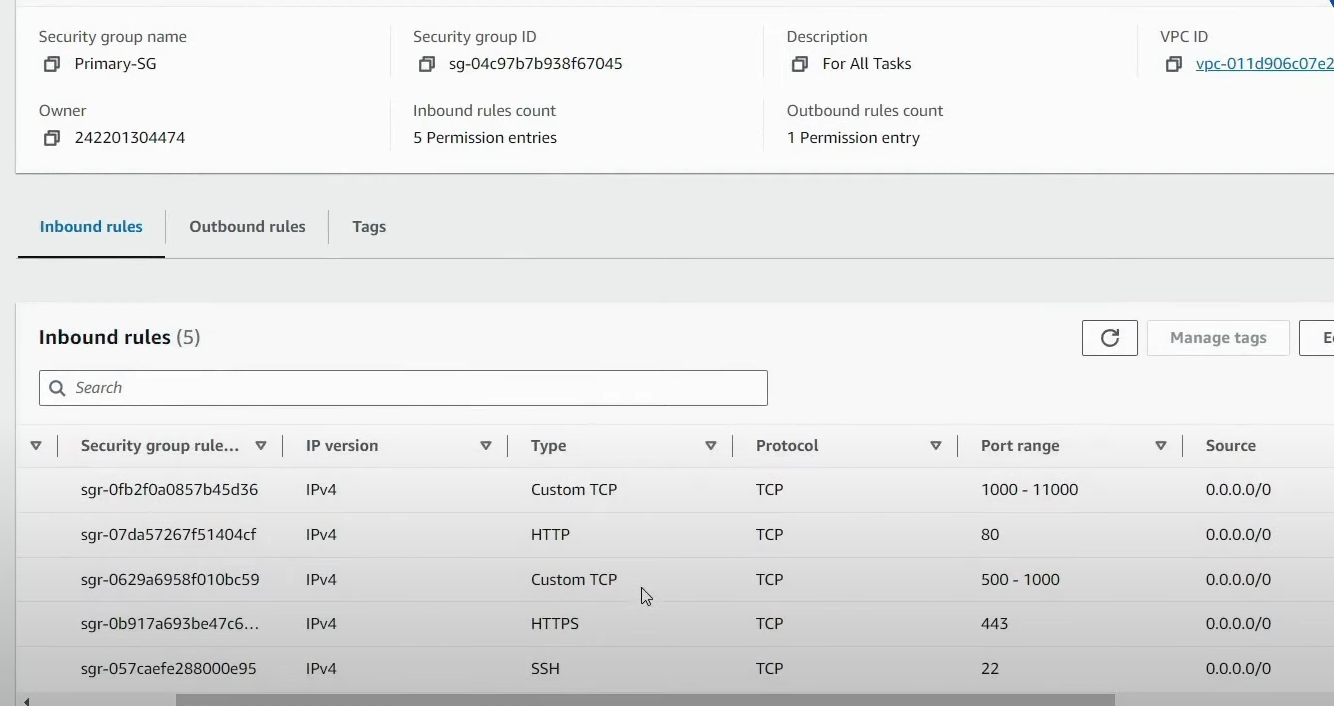
* Ensure traffic switch capabilities are configured.
* Define security group port settings.



Select Environment in which we want to deploy and make sure to check switch traffic.



Security Group Port :



## Step 1: Create Master Server (Ubuntu) for EKS Cluster

1. **Update System**

sudo apt update

1. **Establish AWS CLI Connection**
2. **Install Terraform**

sudo snap install terraform --classic

1. **Clone Repository**

git clone https://github.com/jaiswaladi246/Blue-Green-Deployment.git

1. **Initialize and Apply Terraform**
2. cd Blue-Green-Deployment/cluster
3. terraform init

terraform apply -auto-approve

1. **Install kubectl**
2. **Configure EKS Cluster Connection**

aws eks --region ap-south-1 update-kubeconfig --name <cluster-name>

1. **Set Up RBAC for Cluster Access**
2. **Add Jenkins Secret Token for Authentication**
   * Go to **Manage Jenkins** → **Credentials** → **Secret Text**

## Step 2: Create Jenkins Server

1. **Update System**

sudo apt update

1. **Install Java**

sudo apt install openjdk-17-jre-headless -y

1. **Install Jenkins** from the official site
2. **Enable Jenkins Service**
3. **Install Required Tools on Jenkins**
   * **Docker**
   * sudo apt install docker.io –y
   * sudo usermod -aG docker jenkins
   * **Trivy** (Security Scanning)
   * **kubectl**
4. **Install Jenkins Plugins**
   * SonarQube Scanner, Maven, Docker Pipeline, Kubernetes Plugins
5. **Configure Jenkins Pipeline** – Blue-green

**Pipeline Setup**:

* + - 1. **Discard old builds (max 2)**
      2. **Configure Maven & SonarQube**
    1. Credentials setup
       1. Token for SonarQube
          1. Need to add SonarQube server inside jenking

Manage Jenkins – System – Soanr Qube installation

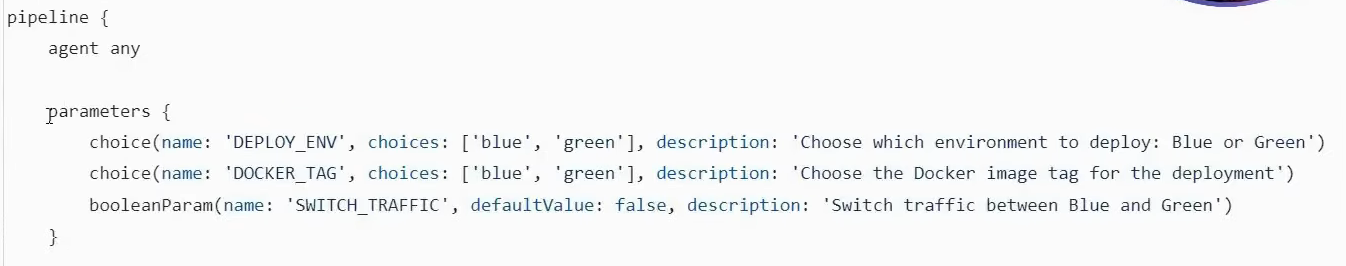
* + 1. Manage Jenkins – managed files – Global Maven settings – add credentials for nexus server (maven release and maven snapshot)
    2. Update maven release and maven snapshot url in pom.xml
    3. Stages :
       1. Add tools maven inside tools sections
       2. Stage 1- Git checkout source code
       3. Stage 2 – compile ( mvn compile)
       4. Stage 3 test 🡪 mvn test –DskipTests=true
       5. Stage – trivy file system scan –
       6. Stage SonarQube scan
          1. We need two things sonarqube server to publish report
          2. And scanner is the tool to perform analysis and generate report on server
          3. To define sonarqube server – need to define it under environment section
       7. Perform Code quality check – name – Quality gate check
          1. Copy Jenkins URL – go to SonarQube – administration – configuration – webhooks
          2. Pipeline syntax for timeout – 1 hour

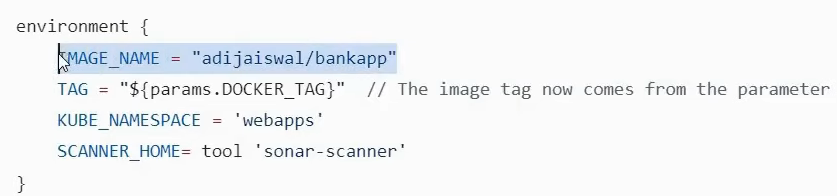


* + - 1. **Build Stage – mvn package –DskipTests=true**
      2. **Publish artifact to nexus**

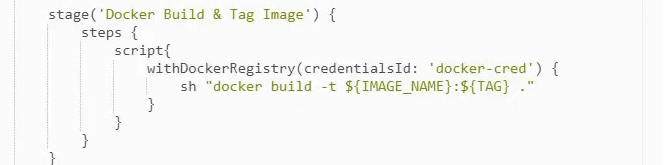


* + - 1. **We need to create docker image and we need two docker image for blue and green environment**

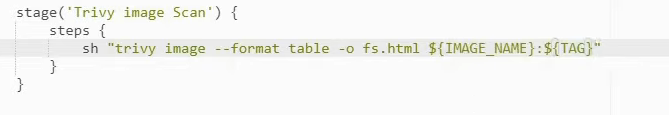




* + - 1. **Stage – Docker Build & Tag image**



* + - 1. **Docker image scan stage**



* + - 1. **Push image to Docker hub repo**



Important change – SQL server and LB are not going to change

* + - 1. **Deploy MySQL Deployment and Service**
         1. Deploy MySQL as a backend database in Kubernetes.



* + - 1. **Deploy Application Load Balancer Service**
         1. Setup Load Balancer to manage traffic between Blue and Green environments.
      2. Blue-**Green** Traffic Switching
         1. Update the Kubernetes service to switch traffic from Blue to Green.
         2. In case of issues, revert back to the Blue deployment.

Complete Jenkins file in the code for reference

1. Create Nexus Server
   1. sudo apt update
   2. setup nexus using Docker container
      1. Install Docker 🡪 sudo apt install docker.io –y
      2. Add permission for other user to run Docker -🡪 sudo usermod –aG docker ubuntu
      3. Restart machine or run newgrp docker
      4. Docker run –d –p 8081:8081 sonatype/nexus3
      5. Username – admin and get the password from inside container
         1. Docker exec –it container ID /bin/bash
         2. Cd sonatype-work/nexcus3 and cat password
2. Create SonarQube Server
   1. sudo apt update
   2. setup SonarQube using Docker container
      1. Install Docker 🡪 sudo apt install docker.io –y
      2. Add permission for other user to run Docker -🡪 sudo usermod –aG docker ubuntu
      3. Restart machine or run newgrp docker
      4. Docker run –d –p 9000:9000 sonarqube:lts/community
   3. Default user name and password - admin