

Deploying a Spring Boot application on Google Cloud Platform (GCP) using Compute Engine involves several steps. Below is a comprehensive guide to help you accomplish this:

---

## 1. Prerequisites

- **GCP Account:** Ensure you have a GCP account with billing enabled.
  - **Spring Boot App:** A built Spring Boot .jar file.
  - **gcloud CLI:** Installed and authenticated with your GCP account.
  - **Java Runtime:** Ensure the Compute Engine instance has a compatible JDK.
- 

## 2. Steps to Deploy

### Step 1: Build Your Spring Boot Application

1. Build your application using Maven or Gradle:
2. `mvn clean package`

This generates a .jar file in the target directory (e.g., app.jar).

---

### Step 2: Create a GCP Compute Engine Instance

1. **Navigate to the GCP Console:**
    - Open the [Google Cloud Console](#).
  2. **Create a Compute Engine Instance:**
    - Go to **Compute Engine > VM Instances** and click **Create Instance**.
    - Choose:
      - **Machine Type:** e.g., e2-micro for small applications.
      - **Operating System:** Ubuntu or Debian.
    - Allow HTTP and HTTPS traffic in the **Firewall** section.
    - Click **Create**.
- 

### Step 3: Connect to the Compute Engine Instance

1. SSH into the instance from the GCP Console or using the gcloud CLI:
  2. `gcloud compute ssh <INSTANCE_NAME>`
- 

### Step 4: Install Required Software on the VM

1. **Update the System:**
2. `sudo apt update && sudo apt upgrade -y`
3. **Install Java:**
4. `sudo apt install openjdk-17-jdk -y`

Verify installation:

`java -version`

---

### Step 5: Transfer the Spring Boot Application

1. **Upload the .jar File:** Use scp to transfer the .jar file to the instance:
2. `gcloud compute scp ./target/app.jar <INSTANCE_NAME>:~`

Alternatively, use the GCP Console to upload the file.

---

### Step 6: Run the Spring Boot Application

1. Run the application using the `java -jar` command:
2. `java -jar app.jar`

This runs the Spring Boot app on port 8080 by default.

---

### Step 7: Expose the Application

1. By default, Spring Boot apps run on port 8080. To expose this port:
    - Edit the firewall settings in GCP to allow traffic on port 8080.
    - Navigate to **VPC network > Firewall rules** in the GCP Console.
    - Click **Create Firewall Rule**:
      - Target: All instances in the network or specify the VM instance.
      - Protocol and Ports: `tcp:8080`.
      - Click **Create**.
- 

### Step 8: Access the Application

1. Use the external IP address of the VM to access the app:
2. `http://<EXTERNAL_IP>:8080`

Replace `<EXTERNAL_IP>` with the public IP of your Compute Engine instance.

---

## 9. Automate Startup with a Systemd Service (Optional)

To ensure the app runs automatically after reboot:

1. Create a systemd service file:
2. `sudo nano /etc/systemd/system/spring-boot-app.service`

Add the following content:

[Unit]

Description=Spring Boot Application

After=network.target

[Service]

User=<your-user>

ExecStart=/usr/bin/java -jar /home/<your-user>/app.jar

SuccessExitStatus=143

Restart=always

RestartSec=10

[Install]

WantedBy=multi-user.target

Replace <your-user> with the username of the instance.

3. Enable and start the service:
4. `sudo systemctl enable spring-boot-app`
5. `sudo systemctl start spring-boot-app`

---

## 10. Optional: Use a Custom Domain

1. Assign a static IP to your instance.
  2. Set up DNS records for your domain to point to the static IP.
-