EMP No: 2577108
Name: Pramod

Algorithm- Deploy Application on Cloud

1. **Create an AWS Account and Set Up AWS CLI:** If you don't already have an AWS account, sign up for one. Install the AWS Command Line Interface (CLI) on your local machine and configure it with your AWS credentials.
2. **Package Your Spring Boot Application:** Build and package your Spring Boot application into an executable JAR or WAR file. You can use Maven or Gradle to do this.
3. **Prepare AWS Resources:** Determine the necessary AWS resources for your application, such as an Amazon EC2 instance and an Amazon RDS instance (if your application requires a database). Plan the appropriate configurations like instance type, database engine, storage, etc.
4. **Create an Amazon EC2 Instance:** Once you've planned your instance configuration, launch an Amazon EC2 instance with the desired specifications. Configure the security group to allow access to the necessary ports, for example, allowing incoming traffic on HTTP port 80.
5. **Transfer Application Files:** Use SCP (Secure Copy Protocol) or any other secure method to transfer your packaged Spring Boot application to the AWS EC2 instance.
6. **Configure Security:** Set up appropriate security settings, including security groups and IAM (Identity and Access Management) roles, to ensure secure access to your AWS resources. Restrict access to only the necessary services and ports.
7. **Install Dependencies on EC2 Instance:** Connect to your EC2 instance securely and install Java and any other necessary dependencies required to run your Spring Boot application.
8. **Set Up Environment Variables:** If your application requires environment variables, configure them on the EC2 instance to ensure proper functioning.
9. **Deploy and Run Your Spring Boot Application:** Deploy your Spring Boot application on the EC2 instance using the appropriate command you used during development (e.g., java -jar your-app.jar).
10. **Test the Deployment:** Perform thorough testing to ensure your Spring Boot application is running correctly on the AWS environment. Test various functionalities and use cases to validate its performance and stability.