

CONTACT SUPPORT SALES: 866-618-3282  
INTL: +1-408-335-0825 SCHEDULE A DEMO

BLOG TUTORIALS  
PARTNER WITH US SIGNUP  
LOGIN

**Try Atlantic.Net with a \$250 credit! You will receive a \$250 credit good for 60 days + Free Tier access for one year when you create a new account.**



[Contact Us](#)

# How to Deploy Spring Boot Application on Ubuntu 22.04



April 19, 2024 by [Hitesh Jethva \( 709 \)](#) under [Dedicated Server Hosting](#)  
[0 Comments](#)

## Table of Contents

Step 1 - Install Java JDK

Step 2 - Install Spring Boot CLI

Step 3 - Create a Spring Boot Application

Step 4 - Run Spring Boot Application

Step 5 - Create a Systemd File

Step 6 - Configure Nginx as a Reverse Proxy

Step 7 - Access Spring Boot Application

Conclusion

A Spring Boot application is a Java-based web application built using the Spring Boot framework. Spring Boot is an open-source framework designed to simplify the process of creating production-grade, stand-alone Spring-based applications. It provides various features and conventions that streamline configuration, dependency management, and application deployment.

In this tutorial, we will show you how to deploy spring boot application on Ubuntu 22.04.

## Step 1 – Install Java JDK

Spring Boot requires Java to be installed on your server. You can install it using the following command:

```
apt install openjdk-18-jdk-headless -y
```

After the successful installation, verify the Java version using the following command:

```
java -version
```

Output:

```
openjdk version "18.0.2-ea" 2022-07-19
OpenJDK Runtime Environment (build 18.0.2-ea+9-Ubuntu-22.04
OpenJDK 64-Bit Server VM (build 18.0.2-ea+9-Ubuntu-22.04, m
```



## Step 2 – Install Spring Boot CLI

Spring Boot CLI (Command Line Interface) is a command-line tool provided by the Spring Boot framework for quickly developing and testing Spring Boot applications without needing to set up a full-fledged project structure. It allows developers to write, run, and test Spring Boot applications with minimal configuration.

There are multiple ways to install Spring Boot CLI. In this section, we will install Spring Boot CLI using SDKMAN.

First, install the Zip and Unzip utility.

```
apt install unzip zip
```

Next, install SDKMAN using the following command:

```
curl -s https://get.sdkman.io | bash
```

Output:

```
All done!
```

```
You are subscribed to the STABLE channel.
```

```
Please open a new terminal, or run the following in the exis
```

```
source "/root/.sdkman/bin/sdkman-init.sh"
```

```
Then issue the following command:
```

```
sdk help
```

```
Enjoy!!!
```



Next, source the new SDKMAN! shell from the terminal window.

```
source "/root/.sdkman/bin/sdkman-init.sh"
```

Next, install the Spring Boot CLI using SDK.

```
sdk install springboot
```

## Step 3 – Create a Spring Boot Application

In this section, we will create a simple “Hello World” Spring Boot application using Maven.

First, install Gradle using the following command.

```
sdk install gradle 7.5.1
```

Next, generate a simple Spring Boot project named “hello” in the “hello-world” directory, with Gradle as the build tool.

```
spring init --build=gradle --dependencies=web --name=hello h
```

Next, edit the `HelloApplication.java` file.

```
nano ~/hello-world/src/main/java/com/example/helloworld/Hell
```

Modify is as shown below:

```
package com.example.helloworld;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.bind.annotation.RequestMapping;

@SpringBootApplication
public class HelloApplication {

    public static void main(String[] args) {
        SpringApplication.run(HelloApplication.class, args);
    }

}

@RestController
class Hello {

    @RequestMapping("/")
    String index() {
        return "Hello world";
    }

}
```

Save and close the file.

Then, navigate inside your application directory and create a new build with the following command:

```
cd hello-world
```



```
[Installl]  
WantedBy=multi-user.target
```

Save the file, then reload the systemd daemon.

```
systemctl daemon-reload
```

Now, start the Spring Boot service using the following command.

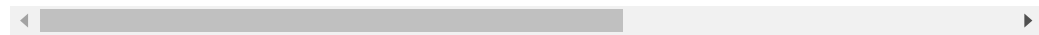
```
systemctl start helloworld
```

To check your application running status, run:

```
systemctl status helloworld
```

Output:

```
● helloworld.service - Spring Boot HelloWorld  
   Loaded: loaded (/etc/systemd/system/helloworld.service;  
   Active: active (running) since Sun 2024-03-24 08:26:48  
   Main PID: 169887 (java)  
     Tasks: 36 (limit: 4579)  
    Memory: 109.4M  
       CPU: 3.850s  
    CGroup: /system.slice/helloworld.service  
            └─169887 /usr/bin/java -jar /root/hello-world/b
```



## Step 6 – Configure Nginx as a Reverse Proxy

You will also need to configure Nginx reverse proxy to implement the application as a web service.

First, install the Nginx package.

```
apt install nginx -y
```

Next, create an Nginx virtual host:

```
nano /etc/nginx/conf.d/helloworld.conf
```

Add the following configuration:

```
server {  
    listen 80;  
  
    server_name helloworld.example.com;  
  
    location / {  
        proxy_pass http://localhost:8080/;  
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
        proxy_set_header X-Forwarded-Proto $scheme;  
        proxy_set_header X-Forwarded-Port $server_port;  
    }  
}
```

Save the file and then edit the Nginx main configuration file:

```
nano /etc/nginx/nginx.conf
```

Add the following line after the line `http{`:

```
server_names_hash_bucket_size 64;
```

Save and close the file, then verify Nginx for any syntax errors.

```
nginx -t
```

Output:

```
nginx: the configuration file /etc/nginx/nginx.conf syntax is correct  
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

Finally, restart the Nginx service to apply the changes.

```
systemctl restart nginx
```

## Step 7 – Access Spring Boot Application

At this point, your Spring Boot application is created with Nginx as a reverse proxy. You can now access it using the URL

**`http://helloworld.example.com`.**

## Conclusion

The Spring Boot application empowers developers to confidently build and deploy robust, production-grade applications while optimizing resource utilization and enhancing the development workflow. You can easily create and deploy your Spring Boot application on [dedicated server hosting](#) from Atlantic.Net!

**Try Atlantic.Net with a \$250 credit!**

You will receive a \$250 credit good for 60 days + Free Tier access for one year when you create a new account.

**Get Started!**

**Subscribe to our newsletter!**



Email Address \*

Subscribe

< Previous post

Next post >

ALSO ON ATLANTIC.NET

**How to Install MongoDB on ...**

3 years ago · 2 comments

In this How-To we will walk you through the basic install and configuration of ...

**How to Install Fail2ban with Firewalld on ...**

2 years ago · 1 comment

Fail2ban is a free and open-source firewall tool. In this guide, we will show you ...

**How to Ins on Ubuntu**

2 months ago

In this tutori through the process of i

0 Comments

1 Login ▼

G

Start the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name



Share

Best Newest Oldest

Be the first to comment.

# Get a \$250 Credit and Access to Our Free Tier!

Free Tier includes:

- G3.2GB Cloud VPS a Free to Use for One Year
- 50 GB of Block Storage Free to Use for One Year
- 50 GB of Snapshots Free to Use for One Year

## Get a Free-to-Use Cloud VPS

CLOUD  
PLATFORM

Cloud VPS  
Hosting

VPS Pricing

Windows VPS  
Hosting

DEDICATED  
HOSTING

GPU Server  
Hosting

Dedicated  
Cloud Hosts

Dedicated  
Servers

COMPLIANCE  
HOSTING

HIPAA  
Compliant  
Hosting

HIPAA  
WordPress  
Hosting

ABOUT US

Why  
Atlantic.Net

Our  
Certifications

Our Data  
Centers

Linux VPS Hosting	Bare Metal Servers	Pharma & BioTech Solutions	Media Press Room
WordPress VPS Hosting	Managed Private Cloud	HIPAA Disaster Recovery	Partners Program
Backup & Replication	Virtual Private Cloud	Hosting with BAA	Career Opportunities
Secure Block Storage	Colocation Hosting	PCI Compliant Hosting	Contact Us
	Managed Services	Digital Advertising	

SUPPORT

100% Uptime SLA

Network Status

API Documentation

Speed Test

Support

Service Policies

Privacy Policy

SALES:

866-  
618-  
3282

INTL:

+1-408-  
335-  
0825

RESOURCES

How to Make a Website HIPAA Compliant

How to Make WordPress HIPAA Compliant

How to Make a Database HIPAA Compliant

How to Become HIPAA Compliant

How to Make Storage HIPAA Compliant

How to Make File Sharing HIPAA Compliant

How to Protect PHI in the Cloud

What Is HIPAA Compliance?

What Is a HIPAA BAA?

What Is the HIPAA Security Rule?

What is HIPAA Compliant Hosting?

What Is HIPAA Cloud Hosting?

What Is Healthcare Hosting?

What Is Protected Health Information?

What Is IAAS?

What Is PAAS?

What Is VPS Hosting?

What Is Cloud Hosting?

What Is Colocation Hosting?

What Is Private Hosting?

What Is VMWare?

What Is Server Virtualization?

What Is an Apache Server?

What Is an IIS Server?

What Is MSSQL?

What Is Database Hosting?

What Is Block Storage?

What Is MySQL?

What Is PCI Compliance?

What Is a SAN?

What Is a Cyber Attack?

© 2024 Atlantic.Net, All Rights Reserved.