Deploy Spring Boot Applications for NGINX on Ubuntu 22.04

Published May 24, 2022 by Jeff Novotny

Select distribution: Ubuntu 22.04 LTS

Java is a powerful and widely used programming language. Unfortunately, it is also very complex and some of the common Java IDEs are difficult to master. Spring Boot and the Spring Framework simplify the Java development environment and make it easier to create applications. This guide explains how to use Spring Boot tools to create a simple Java application that runs on an Apache Tomcat server. It also describes how to register the application as a service and make it available remotely through an NGINX reverse proxy.

What is Spring Boot and the Spring Framework?

The Spring Framework is an application framework and inversion of control container for Java-based programs. It constructs the low-level infrastructure for a Java application, allowing developers to focus on the business logic.

instance. See our Getting Started with Linode and Creating a Compute Instance guides.

2. Follow our Setting Up and Securing a Compute Instance guide to update your system. You may also wish to set the timezone, configure your hostname, create a limited user account, and harden SSH access.

> **Note**
>
> This guide is written for a non-root user. Commands that require elevated privileges are prefixed with `sudo`. If you are not familiar with the `sudo` command, see the Users and Groups guide.

Installing Spring Boot and all Prerequisites on Ubuntu 22.04

A Spring Boot environment relies on several other components. These include the Java JDK, the NGINX web server, and the Gradle build tool. Both Spring Boot and Gradle can be downloaded using the SDKMAN! utility. The following guide is designed for Ubuntu 22.04 users, but the installation process is very similar in Ubuntu 20.04.

Installing Java JDK 17

To use Spring Boot, Java JDK must be installed. Spring Boot works with any release of the JDK between Java 8 and Java 18. However, a reasonably recent release is recommended.

Not all editions of Java are available for free. Starting with Java 11 and until recently, Oracle required a fee to use Java in production. However, beginning with Java 17, the Java software is now available for free again. The new Oracle Java license allows it to be used commercially at no cost.

To install Java JDK release 17, follow these instructions.

1. Ensure `software-properties-common` is installed. This package is often

4. Use `apt` to install Java JDK 17.

```
sudo apt install oracle-java17-installer --install-re
```

5. Confirm the system is running the correct release of Java.

```
java -version
```

```
java version "17.0.1" 2021-10-19 LTS
Java(TM) SE Runtime Environment (build 17.0.1+12-LTS-39)
Java HotSpot(TM) 64-Bit Server VM (build 17.0.1+12-LTS-39
```

Installing NGINX

Spring Boot works well with the NGINX web server, which is now available as part of the default Ubuntu software library. For more information on NGINX, consult the NGINX website. To install NGINX, follow these instructions.

1. Install the NGINX server.

```
sudo apt install nginx
```

2. Confirm NGINX is properly running using the `systemctl` utility.

```
sudo systemctl status nginx
```

```
nginx.service - A high performance web server and a rever
     Loaded: loaded (/lib/systemd/system/nginx.service; e
     Active: active (running) since Wed 2022-05-25 09:43:
```

The Basics

Kotlin Vs. Java: Key Differences

Get Started Building A Website With The Play Framework

Installing Java Development Kit 11 On CentOS 8

Installing Java Development Kit 11 On Debian 10

Installing Java Development Kit 11 On Ubuntu 20.04

```
sudo ufw status
```

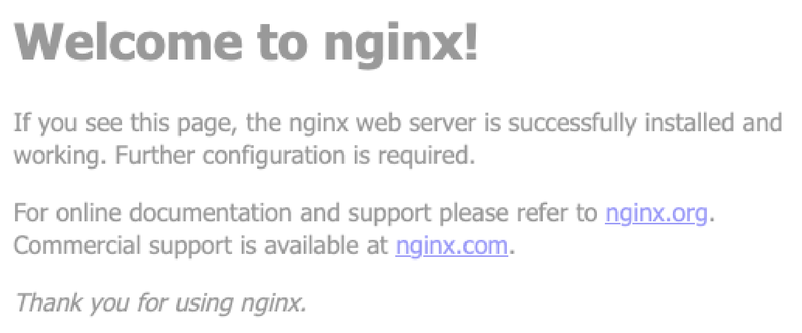
```
Status: active

To                           Action      From
--                           ------      ----
OpenSSH                      ALLOW       Anywhere
Nginx Full                   ALLOW       Anywhere
OpenSSH (v6)                 ALLOW       Anywhere (v6)
Nginx Full (v6)              ALLOW       Anywhere (v6)
```

6. Ensure the server is working properly. Visit the IP address of the Linode and confirm the default NGINX page appears.

> **Note**
>
> To determine the IP address of the Ubuntu system, use the Linode Dashboard.

```
http://server_IP_address/
```



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org. Commercial support is available at nginx.com.

Thank you for using nginx.

Installing the Spring Boot CLI

The Spring Boot CLI utility can be installed using several different methods.

3. Source the new SDKMAN! shell from the terminal window. Alternatively, open a new terminal to use SDKMAN!. In the following command, replace `userdir` with the name of the user directory.

```
source "/home/userdir/.sdkman/bin/sdkman-init.sh"
```

4. Verify SDKMAN! is properly installed. The `sdk help` command displays information about the release and usage information.

```
sdk help
```

5. Use `sdk` to install the Spring Boot CLI module.

```
sdk install springboot
```

```
Spring CLI v2.7.0
```

6. Install the most recent release of the Gradle build tool. This is currently `7.4.2`.

```
sdk install gradle 7.4.2
```

```
Installing: gradle 7.4.2
Done installing!
Setting gradle 7.4.2 as default.
```

How to Create a Spring Boot Application on Ubuntu 22.04

Note

The `spring init` command allows for many possible options. To see all the possible parameters, run the following command.

```
spring init --list
```

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

```
Using service at https://start.spring.io
Project extracted to '/home/userdir/hello-world'
```

2. The `init` command creates a `HelloApplication.java` file inside `~/hello-world/src/main/java/com/example/helloworld`. This file includes some essential `import` directives along with a public `HelloApplication` class. Modify this file as follows:

   - Add an `import` statement to import the `RestController` and `RequestMapping` functionality beneath the other import statements.
   - Add a new `Hello` class to display the text `Hello World`.
   - Precede the class with the `@RestController` annotation. This annotation simplifies the web service creation and indicates the class returns an object rather than a view.
   - Inside the class, add the Spring `@RequestMapping` annotation. This technique maps a URL to a Spring function. For the root directory `/`, the class invokes the `Hello` function. When a web user accesses the root directory, “Hello World” is printed out.

   Modify `HelloApplication.java` so it matches the following example.

   File: ~/hello-world/src/main/java/com/example/helloworld/HelloApplication.java

```
}

@RestController
class Hello {

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

3. From the root directory of the project, use Gradle to build the Java application. This command creates a new `build` directory inside the project.

```
cd hello-world
./gradlew build
```

```
BUILD SUCCESSFUL in 51s
7 actionable tasks: 7 executed
```

Running and Testing the Spring Boot Application

1. Run the application inside a Tomcat server. Apache Tomcat provides an HTTP web server environment that can run Java code. The following command runs a servlet at `localhost:8080`.

```
java -jar build/libs/hello-world-0.0.1-SNAPSHOT.jar
```

2. **(Optional)** Alternatively, it is possible to run the application in place without first building the jar file. This is a faster option for quick internal testing, especially if the application is not yet ready for a final build.

4. When testing is complete, stop the Tomcat server using **CTRL+C**.

For more detailed information on creating a Spring Boot application, see the Developing Your First Spring Boot Application guide.

Creating an Init Script for the Spring Boot Application

To access the new application externally across the internet, a few more steps are required. An init script for the Spring Boot application must be created inside the `systemd` server. This registers Spring Boot as a service and launches it at system start-up time.

1. Create a service script for `helloworld.service` in the `/etc/systemd/system` directory as follows. The `ExecStart` field must contain the full path to the application `.jar` file. This is the same file that ran inside Tomcat server earlier. For the path name, replace `userdir` with the name of the user directory.

   File: /etc/systemd/system/helloworld.service

   ```
   [Unit]
   Description=Spring Boot HelloWorld
   After=syslog.target
   After=network.target[Service]
   User=username
   Type=simple

   [Service]
   ExecStart=/usr/bin/java -jar /home/userdir/hello-wo
   Restart=always
   StandardOutput=syslog
   StandardError=syslog
   SyslogIdentifier=helloworld

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

2. Start the service.

Configuring a Reverse Proxy for the Spring Boot Application

To implement the application as a web service, a reverse proxy is required. A reverse proxy provides an external shell around a web server. The reverse proxy intercepts incoming requests to the server, and forwards or redirects them as necessary. This technique can improve server performance and enhance security.

In this case, the reverse proxy permits the Java application service to run on an unprivileged port. An unprivileged port does not have an official service associated with it. The proxy receives incoming HTTP requests for the root `/` directory and redirects them to port `8080` on the same host. The Tomcat server running on this port launches the Java application.

To create and test a reverse proxy for the application, follow these steps.

1. Create an NGINX configuration file for the service. The file should have the same name as the service and have the `.conf` extension. Place the file in the `sites-available` directory in the same manner as a regular site configuration file.

File: /etc/nginx/sites-available/helloworld.conf

```
server {
        listen 80;
        listen [::]:80;

        server_name example.com;

        location / {
            proxy_pass http://localhost:8080/;
            proxy_set_header X-Forwarded-For $proxy
            proxy_set_header X-Forwarded-Proto $sch
            proxy_set_header X-Forwarded-Port $serv
        }
}
```

```
nginx: the configuration file /etc/nginx/nginx.conf synta
nginx: configuration file /etc/nginx/nginx.conf test is s
```

5. Restart the NGINX server.

```
sudo systemctl restart nginx
```

6. Visit the IP address of the Linode to test the new service. The browser should display the "Hello World" message, which is the output of the Java application.

```
http://ip_address
```

Concluding Thoughts about Deploying Spring Boot Applications with NGINX on Ubuntu 22.04

Spring Boot is an extension of the Spring Framework that makes it easier for Ubuntu Java developers to create applications. Spring Boot is used with the NGINX web server to make standalone Spring applications available over the web. Spring requires a recent release of the Java JDK and can be downloaded and managed using the SDKMAN! package manager.

Spring Boot works with a large number of build tools, including Gradle. Developers can build upon a default Spring Boot configuration and use Spring Boot annotations to quickly add features. After building a `.jar` file using Gradle, developers can run the application locally using the Apache Tomcat server. To run the application over the web, add a service init file and create a reverse proxy to redirect traffic to the servlet. For more information about Spring Boot, consult the Spring Boot web page.

- Oracle Java license
- NGINX Website
- SDKMAN!
- Gradle
- Apache Maven
- Gradle vs Maven comparison guide
- Wikipedia page on Inversion of Control Containers

This page was originally published on May 24, 2022.

Your Feedback Is Important

Let us know if this guide was helpful to you.

Provide Feedback

Join the conversation.

Read other comments or post your own below. Comments must be respectful, constructive, and relevant to the topic of the guide. Do not post external links or advertisements. Before posting, consider if your comment would be better addressed by contacting our Support team or asking on our Community Site.

Site Map

Accessibility Commitment

Support

Legal Center

Partners

System Status