

Get unlimited access to the best of Medium for less than \$1/week. [Become a member](#)



Mini-Project: Deploying Java Application with Tomcat



Husni B. · [Follow](#)

4 min read · Aug 4, 2020

Listen

Share

More

In this session, I'll show you how to deploy 'Hello World' Java application with tomcat. Before we heading to the main course, it's good to know the requirements.

Apache Tomcat

Apache Tomcat is an open-source implementation of the Java Servlet, JavaServer Pages, Java Expression Language, and WebSocket technologies. Tomcat provides a “pure Java” HTTP web server environment in which Java code can run.

Later, I’ll use Tomcat 9 that requires Java version installed on the system. We’ll install OpenJDK 11, the open-source implementation of the Java Platform.

Maven

Maven is a build automation tool used primarily for Java projects. Maven can also be used to build and manage projects written in C#, Ruby, Scala, and other languages. The Maven project is hosted by the Apache Software Foundation, where it was formerly part of the Jakarta Project.

Step 1 — Setting Up Environment

- Install Java 11

```
# apt install openjdk-11-jdk
```

- Check Java version

```
# java -version
openjdk version "11.0.8" 2020-07-14
```

- Download Tomcat from source

At the time of writing, the latest Tomcat version is 9.0.37. Before continuing with the next step, check the Tomcat 9 download page to see if a newer version is available.

```
# wget https://downloads.apache.org/tomcat/tomcat-
9/v9.0.37/bin/apache-tomcat-9.0.37.tar.gz
```

- Extract the file

```
# mkdir /etc/tomcat
```

Once the download is complete, make a directory named **tomcat** under **/etc** and then extract the tar file to the **/etc/tomcat** directory.

```
# tar -xf apache-tomcat-9.0.37.tar.gz -C /etc/tomcat/
```

- Make the script executable

```
# sh -c 'chmod +x /etc/tomcat/bin/*.sh'
```

- Make the service run as a service

```
# nano /etc/systemd/system/tomcat.service
```

Instead of using the shell scripts to start and stop the Tomcat server, we'll set it to run as a service. Paste the following script:

```
[Unit]
Description=Tomcat 9 servlet container
After=network.target

[Service]
Type=forking

User=root
Group=root

Environment="JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64"
Environment="JAVA_OPTS=-Djava.security.egd=file:///dev/urandom -Djava.awt.headless=true"

Environment="CATALINA_BASE=/etc/tomcat"
Environment="CATALINA_HOME=/etc/tomcat"
Environment="CATALINA_PID=/etc/tomcat/temp/tomcat.pid"
Environment="CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC"

ExecStart=/etc/tomcat/bin/startup.sh
ExecStop=/etc/tomcat/bin/shutdown.sh

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

- Restart the daemon

Notify systemd that a new unit file exists.

```
# systemctl daemon-reload
```

- Start the Tomcat

```
# systemctl start tomcat
```

If it's running properly, you can check on your web by submitting your VM's IP. By default, Tomcat runs on port 8080.

The screenshot shows the Apache Tomcat 9.0.37 homepage. The browser title is "Apache Tomcat/9.0.37". The address bar shows "Not secure | 35.198.231.31:8080". The page features a green banner at the top with the text "If you're seeing this, you've successfully installed Tomcat. Congratulations!". Below the banner is a cartoon cat logo. To the right of the banner are three buttons: "Server Status", "Manager App", and "Host Manager". The main content area has a light green background. It includes a "Recommended Reading" section with links to "Security Considerations How-To", "Manager Application How-To", and "Clustering/Session Replication How-To". Below this is a "Developer Quick Start" section with links to "Tomcat Setup", "Realms & AAA", "Examples", "Servlet Specifications", and "First Web Application", "JDBC DataSources", "Tomcat Versions". The bottom of the page is divided into three yellow boxes: "Managing Tomcat", "Documentation", and "Getting Help". The "Managing Tomcat" box contains links to "Release Notes", "Changelog", "Migration Guide", and "Security Notices". The "Documentation" box contains links to "Tomcat 9.0 Documentation", "Tomcat 9.0 Configuration", and "Tomcat Wiki". The "Getting Help" box contains links to "FAQ and Mailing Lists" and a list of available mailing lists: "tomcat-announce" (for announcements), "tomcat-users" (for user support), "taglibs-user" (for Taglibs support), and "tomcat-dev" (for development messages). At the very bottom, there are links for "Other Downloads", "Other Documentation", "Get Involved", "Miscellaneous", and "Apache Software Foundation".

Step 2 — Create a Java Application

- Install a Java build tool

As mentioned before, we will use Maven as the Java build tool. So, make sure Maven installed on your VM. If you don't have it already, you can easily install it using this line.

```
# apt install maven
```

- Check Maven version

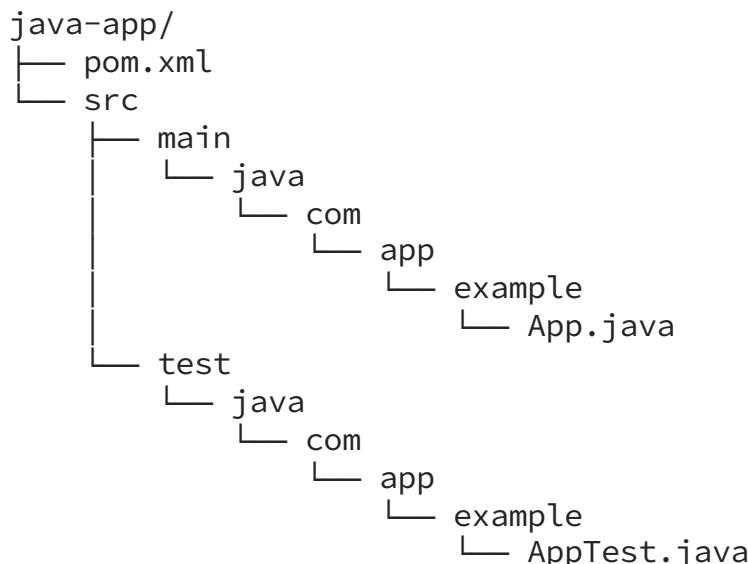
```
# mvn -version
Apache Maven 3.6.3
```

- Generate project

Maven has a feature that can generate an initial maven project with a folder structured. Just paste this following line:

```
# mvn archetype:generate -DgroupId=com.app.example -
DartifactId=java-app -DarchetypeArtifactId=maven-archetype-
quickstart -DinteractiveMode=false
```

The folder structure should be like this:



- Configure the application

```
# nano pom.xml
```

POM is the fundamental unit of work in Maven. It is an XML file that contains information about the project and configuration details used by Maven to build the project. It contains default values for most projects. So, we'll change the pom.xml with this buck of the script:

```
1  <?xml version = "1.0" encoding = "UTF-8"?>
2  <project xmlns = "http://maven.apache.org/POM/4.0.0"
3      xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
4
5      xsi:schemaLocation = "http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0
6      modelVersion>4.0.0</modelVersion>
7
8      <groupId>com.tutorialspoint</groupId>
9      <artifactId>hello-world</artifactId>
10     <version>1</version>
11     <packaging>war</packaging>
12
13     <parent>
14         <groupId>org.springframework.boot</groupId>
15         <artifactId>spring-boot-starter-parent</artifactId>
16         <version>2.3.0.RELEASE</version>
17         <relativePath/>
18     </parent>
19
20     <properties>
21         <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
22         <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
23         <java.version>1.8</java.version>
24         <tomcat.version>9.0.37</tomcat.version>
25     </properties>
26
27     <dependencies>
28         <dependency>
29             <groupId>org.springframework.boot</groupId>
30             <artifactId>spring-boot-starter-web</artifactId>
31         </dependency>
32         <dependency>
33             <groupId>org.springframework.boot</groupId>
34             <artifactId>spring-boot-starter-tomcat</artifactId>
35             <scope>provided</scope>
36         </dependency>
37         <dependency>
38             <groupId>org.springframework.boot</groupId>
39             <artifactId>spring-boot-starter-test</artifactId>
40             <scope>test</scope>
41         </dependency>
42     </dependencies>
43
44     <build>
45         <plugins>
46             <plugin>
47                 <groupId>org.springframework.boot</groupId>
48                 <artifactId>spring-boot-maven-plugin</artifactId>
```

```
49      </plugin>
50    </plugins>
51  </build>
52
53 </project>
```

- Create the application

```
# nano src/main/java/com/app/example/App.java
```

Change the contents of the default java application with this code:

```
1 package com.app.example;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5 import org.springframework.boot.builder.SpringApplicationBuilder;
6 import org.springframework.boot.web.servlet.support.SpringBootServletInitializer;
7 import org.springframework.web.bind.annotation.RequestMapping;
8 import org.springframework.web.bind.annotation.RestController;
9
10 @SpringBootApplication
11 @RestController
12 public class App extends SpringBootServletInitializer {
13     @Override
14     protected SpringApplicationBuilder configure(SpringApplicationBuilder application) {
15         return application.sources(App.class);
16     }
17     public static void main(String[] args) {
18         SpringApplication.run(App.class, args);
19     }
20 }
```

Open in app ↗



Search



25 ↴

App.java hosted with ❤️ by GitHub

view raw

- Build the application

Build the application using this command that will download application dependencies, compile, test, and package it as a distributed application.

```
# mvn package
```

Step 2 — Deploy the Application

- Copy application to Tomcat directory

The result of the previous step is a distributed application with .war extension stored in the `target` folder under `java-app`. That .war application will later be deployed on the Tomcat service.

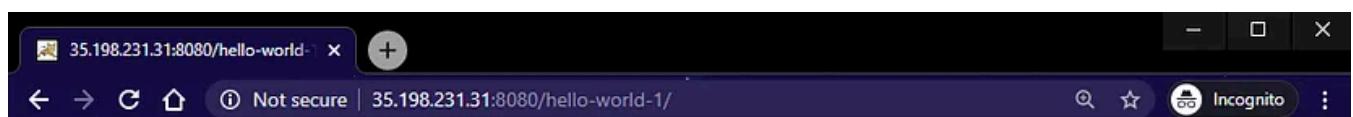
```
# cp java-app/target/hello-world-1.war /etc/tomcat/webapps/
```

Once we've done with copying the app, restart Tomcat service.

```
# systemctl restart tomcat
```

- Check the service

You should now be able to go to your VM's IP at port 8080 with location `/hello-world-1` in your web browser and see your app running.



Hello World Aahhh Mantaapp

References

Linuxize

How to Install Tomcat 9 on Ubuntu 20.04

This tutorial describes how to install and configure Tomcat 9 on Ubuntu 20.04. Apache Tomcat is an open-source web...

linuxize.com

[Java](#)[Programming](#)[Deployment](#)[Web Development](#)[Tomcat](#)[Follow](#)

Written by Husni B.

18 Followers

Bad at writing stuff

More from Husni B.



Husni B. in The Startup

Mini-Project: Deploying Python Application With Nginx

Deploying a clean and simple “Hello World” Python application with Nginx.

5 min read · Aug 3, 2020



107



...



Maya Sayvanova

Want to Freelance for Google?

Google employs 120,000 contractors/freelancers worldwide.

◆ · 4 min read · Apr 21, 2024

👏 2.4K

💬 27



...



👤 Desiree Peralta in The Startup

This Is the Smartest Move You Can Make With Crypto and Bitcoin Right Now

How to take advantage of the most you can this harvest season without losing everything in the process.

◆ · 9 min read · Apr 12, 2024

👏 1.3K

💬 25



...



 Husni B. in FAUN—Developer Community 

Mini-Project: Simple Automation Using Ansible

Implementing configuration management technology using an Ansible.

6 min read · Nov 17, 2019



...

See all from Husni B.

Recommended from Medium

 Rina Reyes

Execute bash script on a remote host using Jenkins Pipeline and Publish Over SSH plugin

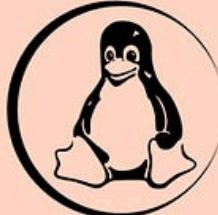
I decided to install Jenkins on another server because I don't want it to mess up with my containers and I will execute different remote...

4 min read · Feb 5, 2024

 39

...

Tips & Tricks to Work with Linux

 Sagar #OpenToWork in FAUN—Developer Community 🐾

Essential Linux Terminal Hacks for Efficiency.

Tips and tricks in Linux terminal for better productivity and ease of use.

4 min read · 3 days ago

25

1



...

Lists



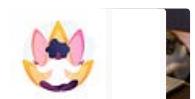
General Coding Knowledge

20 stories · 1192 saves



Coding & Development

11 stories · 596 saves



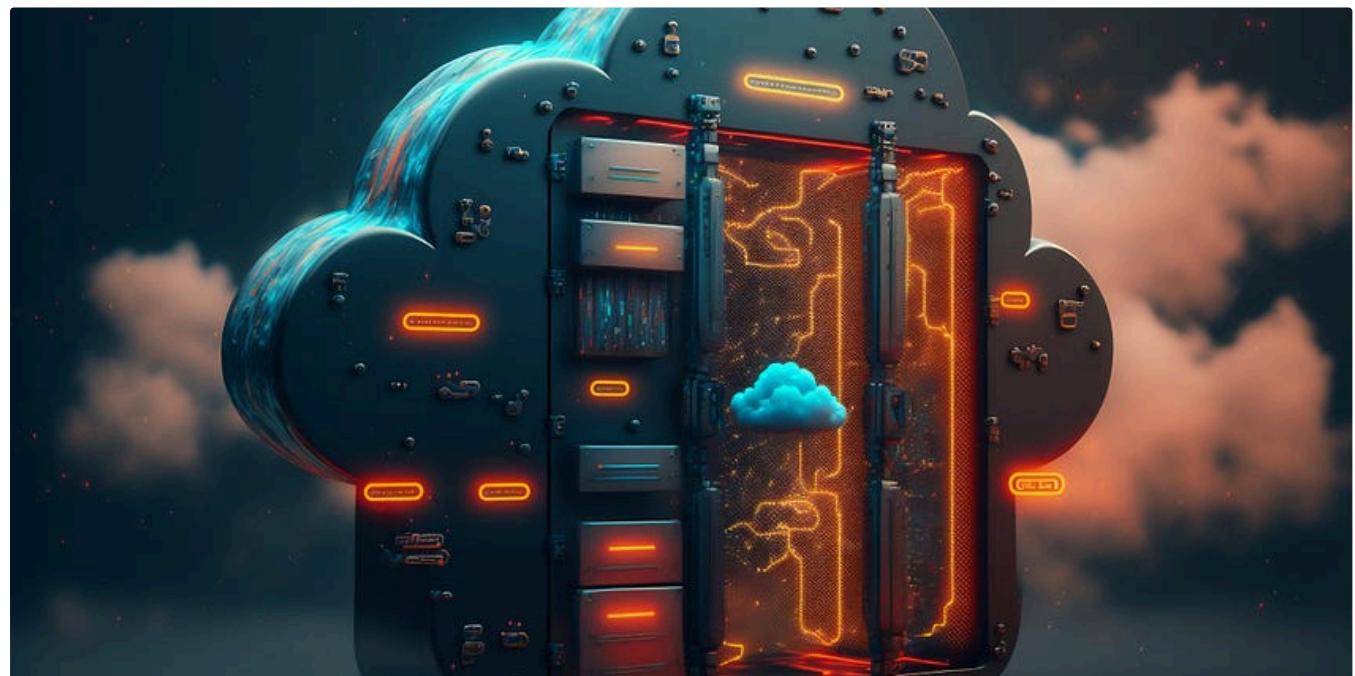
Stories to Help You Grow as a Software Developer

19 stories · 1035 saves



ChatGPT

21 stories · 620 saves



Amit Kumar Upadhyay

Deploying Spring Boot Web App on AWS Using EC2 and S3

image by svstudioart on freepik.com

4 min read · Nov 16, 2023



•••

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: jenkins Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Search our full catalog including 1000s of application and OS images

Recent OS Images

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Linux Quick Start

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOPS, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Chnarsimha

Jenkins CI/CD Pipeline for Building and Pushing Docker Images of a Simple Hello World API...

In this tutorial, we will guide you through the process of setting up a Jenkins CI/CD pipeline to automate the building and pushing of...

10 min read · Feb 8, 2024



•••

Stan's Robot Shop

Login / Register
Cart
Empty
Categories
• Artificial Intelligence
• Robot

Welcome to Stan's Robot Shop
Here you will find all of Stan's friends. Have a browse around and see who is here.

This is a simple example microservices ecommerce application. It has been built using various technologies:

- AngularJS (1.x)
- Nginx
- NodeJS
- Java
- Python
- Golang
- PHP (Apache)
- MongoDB
- Redis
- MySQL

When deployed into an environment monitored by Instana, these technology stacks will be automatically detected and monitored, all with minimum configuration. Every request will be traced end to end. Stan will keep an eye on all those metrics, events and traces and let you know what needs your attention.

To find out more visit the [Instana](#) site.

All the code is available on [Github](#).

anonymous-1

By Sayantan Samanta

Sayantan Samanta

Deploying 3-Tier Architecture with 8 Services and 2 Databases on Amazon EKS | E-commerce Robot Shop

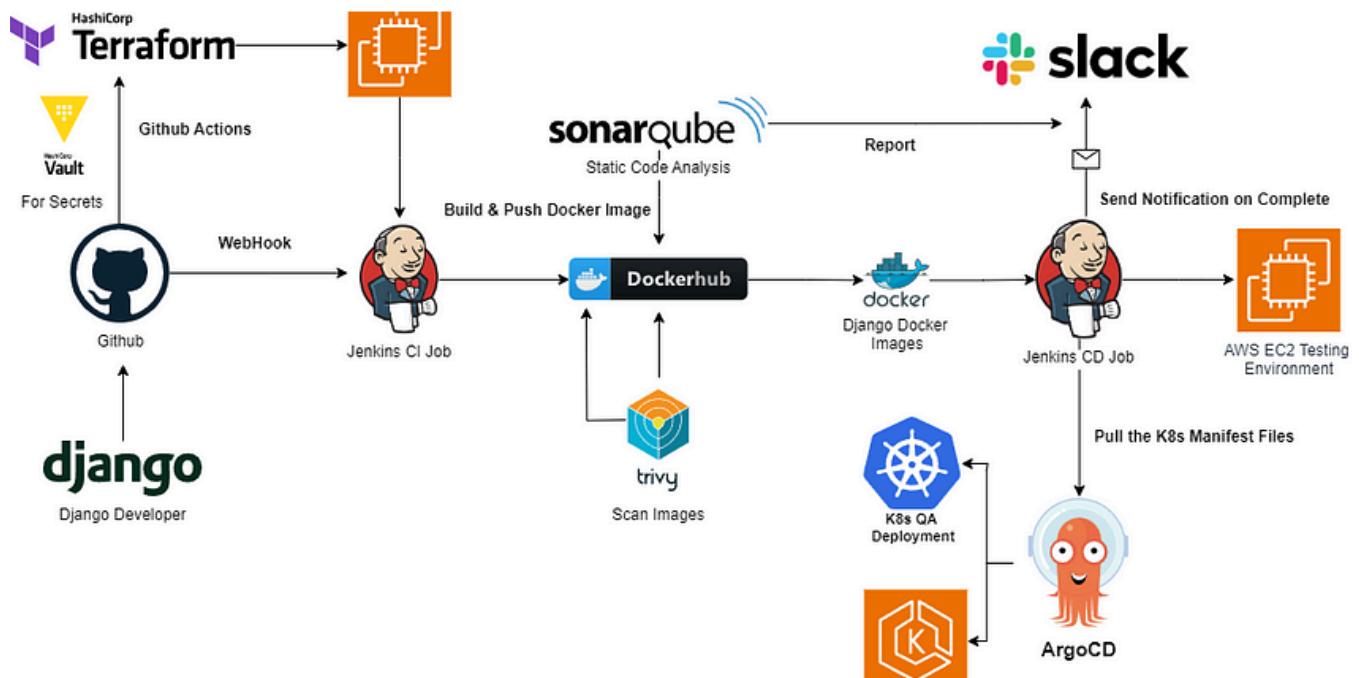
A Comprehensive Guide to deploy a 3-tier app on Kubernetes

13 min read · Jan 3, 2024

👏 20



...



Joel Wembo in Django Unleashed

Technical Guide: End-to-End CI/CD DevOps with Jenkins, Docker, Kubernetes, ArgoCD, Github Actions ...

Building an end-to-end CI/CD pipeline for Django applications using Jenkins, Docker, Kubernetes, ArgoCD, AWS EKS, AWS EC2

48 min read · Apr 12, 2024

👏 338

💬 9



...

See more recommendations