Maven Project

Step 1: Installing Tomcat

1. Update packages

```
apt-get update && apt-get upgrade
```

2. Install tomcat

```
apt-get install tomcat8
```

3. Install docs, admin and examples

```
apt-get install tomcat8-docs tomcat8-examples tomcat8-admin
```

4. If you need to start, stop or restart Tomcat you can use the following commands:

```
systemctl start tomcat8
systemctl stop tomcat8
systemctl restart tomcat8
```

Step 2: Configure Tomcat

1. Edit `/var/lib/tomcat8/conf/tomcat-users.xml.`
   And add the following to access tomcat manager:

```
<role rolename="manager-gui"/>
<role rolename="admin-gui"/>
<user username="username" password="password" roles="manager-gui,admin-gui"/>
```

2. Edit `/usr/share/tomcat7-admin/manager/WEB-INF/web.xml.` And add the following to adjust max size of war file that can be deployed on tomcat. (Here 250MB)

```
<multipart-config>
  <max-file-size>262144000</max-file-size>
  <max-request-size>262144000</max-request-size>
  <file-size-threshold>0</file-size-threshold>
</multipart-config>
```

3. Restart tomcat

```
systemctl restart tomcat8
```

Step 3: Configure your application

Maven

1. Add the following dependency to pom.xml:

```
<dependency>
   <groupId>org.springframework.boot</groupId>
     <artifactId>spring-boot-starter-tomcat</artifactId>
     <scope>provided</scope>
</dependency>
```

2. Change the package type to WAR

```
<packaging>war</packaging>
```

3. Extend SpringBootServletInitializer in main Application class

```
@Configuration
@EnableAutoConfiguration
@SpringBootApplication
public class MyApplication extends SpringBootServletInitializer {

  @Override
  protected SpringApplicationBuilder configure(SpringApplicationBuilder application) {
    return application.sources(MyApplication.class);
  }

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

4. Exclude slf4j logging in pom.xml file.

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter</artifactId>
  <exclusions>
    <exclusion>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-logging</artifactId>
    </exclusion>
  </exclusions>
</dependency>


<dependency>
  <groupId>org.springframework.boot</groupId>
```

```
  <artifactId>spring-boot-starter-web</artifactId>
  <exclusions>
    <exclusion>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-logging</artifactId>
    </exclusion>
  </exclusions>
</dependency>
```

5. Build the WAR file

```
mvn clean package
```

Gradle

1. Add the following dependency to build.gradle:

```
providedRuntime 'org.springframework.boot:spring-boot-starter-tomcat'
```

2. Change the package type to WAR

```
apply plugin: "war"
```

3. Extend SpringBootServletInitializer in main Application class

```
@Configuration
@EnableAutoConfiguration
@SpringBootApplication
public class Application extends SpringBootServletInitializer {

  @Override
  protected SpringApplicationBuilder configure(SpringApplicationBuilder application) {
    return application.sources(Application.class);
  }

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

4. Exclude slf4j logging in build.gradle file

```
configurations {
    all*.exclude module : 'spring-boot-starter-logging'
    all*.exclude module: "log4j-over-slf4j"
}
```

5. Build the WAR file

```
gradle clean
gradle build

Or

gradle clean
./gradlew build -x test
```

Step 4: Deploy .war file

1. Open localhost:8080
2. Click on mana