

Quickstart: Create a Java app on Azure App Service

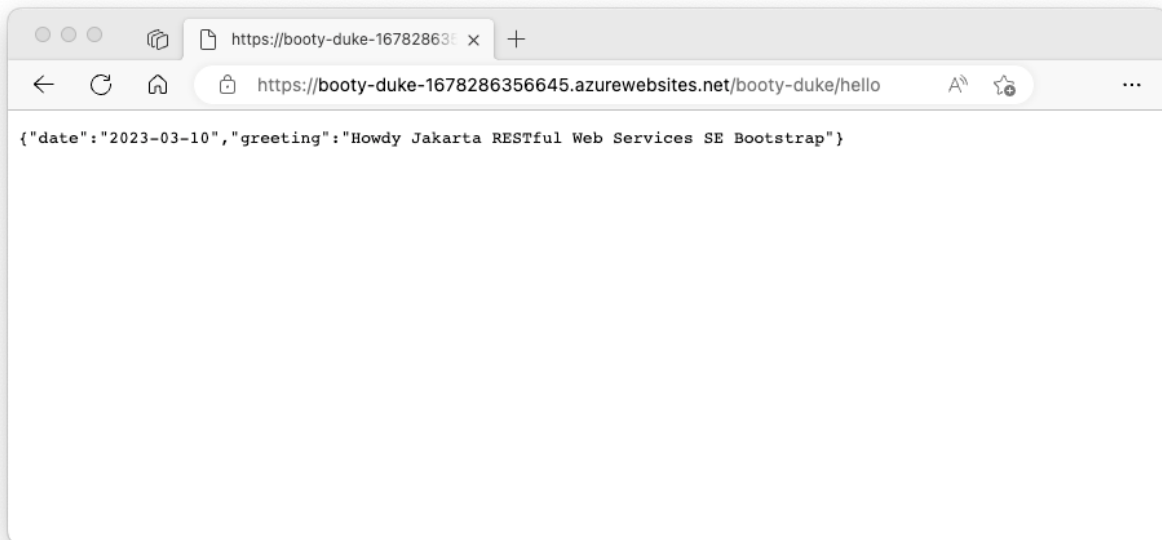
- Article
- 03/09/2023

In this article

1. [1 - Use Azure Cloud Shell](#)
2. [2 - Create a Java app](#)
3. [3 - Configure the Maven plugin](#)
4. [4 - Deploy the app](#)

[Azure App Service](#) provides a highly scalable, self-patching web app hosting service. This quickstart shows how to use the [Azure CLI](#) with the [Azure Web App Plugin for Maven](#) to deploy a .jar file or .war file. Use the tabs to switch between Java SE and Tomcat instructions.

- [Java SE](#)
- [Tomcat](#)
- [JBoss EAP](#)



If Maven isn't your preferred development tool, check out our similar tutorials for Java developers:

- [Gradle](#)
- [IntelliJ IDEA](#)
- [Eclipse](#)
- [Visual Studio Code](#)

If you don't have an [Azure subscription](#), create an [Azure free account](#) before you begin.

1 - Use Azure Cloud Shell

Azure hosts Azure Cloud Shell, an interactive shell environment that you can use through your browser. You can use either Bash or PowerShell with Cloud Shell to work with Azure services. You can use the Cloud Shell preinstalled commands to run the code in this article, without having to install anything on your local environment.

To start Azure Cloud Shell:

Option

Example/Link

Select **Try It** in the upper-right corner of a code or command block.



Selecting **Try It** doesn't automatically copy the code or command to Cloud Shell.

Go to <https://shell.azure.com>, or select the **Launch Cloud Shell** button to open Cloud Shell in your browser.



Select the **Cloud Shell** button on the menu bar at the upper right in the [Azure portal](#).



To use Azure Cloud Shell:

1. Start Cloud Shell.
2. Select the **Copy** button on a code block (or command block) to copy the code or command.

3. Paste the code or command into the Cloud Shell session by selecting **Ctrl+Shift+V** on Windows and Linux, or by selecting **Cmd+Shift+V** on macOS.
4. Select **Enter** to run the code or command.

2 - Create a Java app

- [Java SE](#)
- [Tomcat](#)
- [JBoss EAP](#)

Clone the [Spring Boot Getting Started](#) sample project.

Azure CLICopy

Open Cloudshell

```
git clone https://github.com/spring-guides/gs-spring-boot
```

Change directory to the completed project.

Azure CLICopy

Open Cloudshell

```
cd gs-spring-boot/complete
```

3 - Configure the Maven plugin

Tip

The Maven plugin supports **Java 17** and **Tomcat 10.0**. For more information about latest support, see [Java 17 and Tomcat 10.0 are available on Azure App Service](#).

The deployment process to Azure App Service will use your Azure credentials from the Azure CLI automatically. If the Azure CLI is not installed locally, then the Maven plugin will authenticate with OAuth or device login. For more information, see [authentication with Maven plugins](#).

Run the Maven command below to configure the deployment. This command will help you to set up the App Service operating system, Java version, and Tomcat version.

Azure CLICopy

Open Cloudshell

mvn com.microsoft.azure:azure-webapp-maven-plugin:2.11.0:config

- [Java SE](#)
- [Tomcat](#)
- [JBoss EAP](#)

1. If prompted with **Subscription** option, select the proper subscription by entering the number printed at the line start.
2. When prompted with **Web App** option, select the default option, <create>, by pressing enter.
3. When prompted with **OS** option, select **Windows**.
4. When prompted with **javaVersion** option, select **Java 11**.
5. When prompted with **Pricing Tier** option, select **P1v2**.
6. Finally, press enter on the last prompt to confirm your selections.

Your summary output will look similar to the snippet shown below.

Copy

```
Please confirm webapp properties
Subscription Id : *****_****_****_****_*****
AppName : spring-boot-1599007390755
ResourceGroup : spring-boot-1599007390755-rg
Region : centralus
PricingTier : P1v2
OS : Windows
Java : Java 11
Web server stack : Java SE
Deploy to slot : false
Confirm (Y/N)? : Y
[INFO] Saving configuration to pom.
[INFO] -----
-----
[INFO] BUILD SUCCESS
[INFO] -----
-----
[INFO] Total time: 41.118 s
[INFO] Finished at: 2020-09-01T17:43:45-07:00
[INFO] -----
-----
```

You can modify the configurations for App Service directly in your `pom.xml`. Some common configurations are listed below:

Property	Required	Description	Version
<schemaVersion>	false	Specify the version of the configuration schema. Supported values are: v1, v2.	1.5.2

Property	Required	Description	Version
<subscriptionId>	false	Specify the subscription ID.	0.1.0+
<resourceGroup>	true	Azure Resource Group for your Web App.	0.1.0+
<appName>	true	The name of your Web App.	0.1.0+
<region>	false	Specifies the region where your Web App will be hosted; the default value is centralus . All valid regions at Supported Regions section.	0.1.0+
<pricingTier>	false	The pricing tier for your Web App. The default value is P1v2 for production workload, while B2 is the recommended minimum for Java dev/test. For more information, see App Service Pricing	0.1.0+
<runtime>	false	The runtime environment configuration. For more information, see Configuration Details .	0.1.0+
<deployment>	false	The deployment configuration. For more information, see Configuration Details .	0.1.0+

Be careful about the values of <appName> and <resourceGroup> (helloworld-1590394316693 and helloworld-1590394316693-rg accordingly in the demo), they'll be used later.

I ran into an issue

4 - Deploy the app

With all the configuration ready in your pom file, you can deploy your Java app to Azure with one single command.

- [Java SE](#)
- [Tomcat](#)
- [JBoss EAP](#)

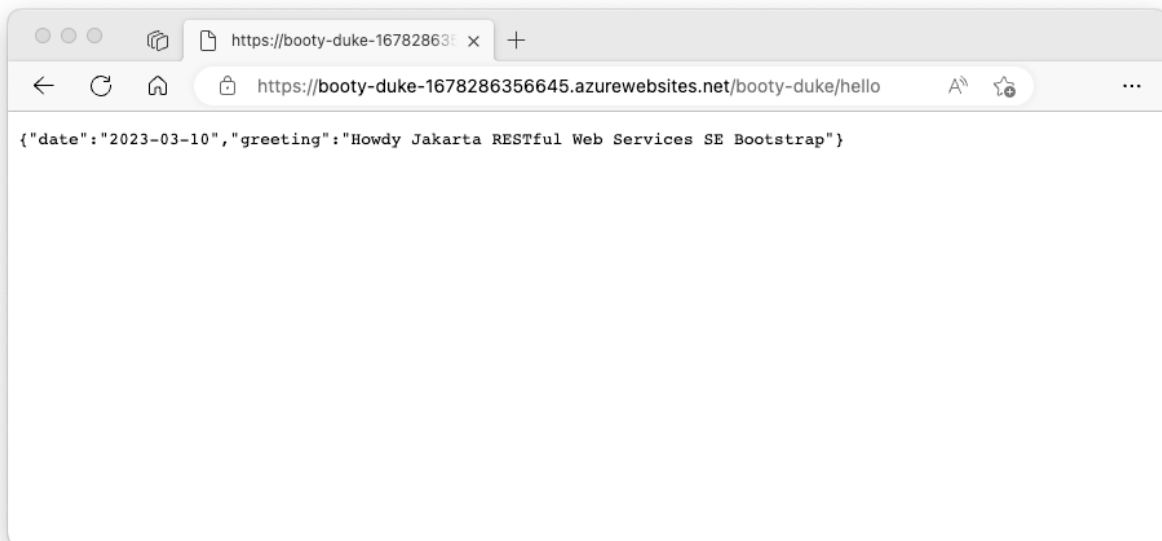
Azure CLI Copy

Open Cloudshell

```
mvn package azure-webapp:deploy
```

Once deployment is completed, your application will be ready at `http://<appName>.azurewebsites.net/` (`http://helloworld-1590394316693.azurewebsites.net` in the demo). Open the url with your local web browser, you should see

- [Java SE](#)
- [Tomcat](#)
- [JBoss EAP](#)



Congratulations! You've deployed your first Java app to App Service.

[I ran into an issue](#)

5 - Clean up resources

In the preceding steps, you created Azure resources in a resource group. If you don't need the resources in the future, delete the resource group from portal, or by running the following command in the Cloud Shell:

Azure CLICopy

Open Cloudshell

```
az group delete --name <your resource group name; for example: helloworld-1558400876966-rg> --yes
```

This command may take a minute to run.