Heroku

1.

>heroku login

(enter username and password)

To logout use

> heroku auth:logout

## 2. Prepare the app

>git clone https://github.com/heroku/java-getting-started.git

>cd java-getting-started

## 3. Deploy the app

>heroku create {app\_name}

>git push heroku master

Ensure atleast 1 instance is running

>heroku ps:scale web=1

Open app

> heroku open

## 3. View Logs

> heroku logs –tail

4.Use external db

heroku config:add **DATABASE\_URL**=mysql://user:pass@server:port/database\_name

**spring boot:**

1.create **Procfile**

web: java $JAVA\_OPTS -Dserver.port=$PORT -jar target/\*.jar

**2.optionally create system.properties**

java.runtime.version=1.7

$ git init .

$ git add .

$ git commit -m 'initial add'

$ heroku create

$ git push heroku master

$ heroku open

You can use the following to examine system variables on Heroku with:

heroku config

or to get a specific variable.

heroku config:get JAVA\_OPTS

**Deploy WAR**

## [Configure Maven to download Webapp Runner](https://devcenter.heroku.com/articles/java-webapp-runner#configure-maven-to-download-webapp-runner)

<build>

...

<plugins>

...

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-dependency-plugin</artifactId>

<version>2.3</version>

<executions>

<execution>

<phase>package</phase>

<goals><goal>copy</goal></goals>

<configuration>

<artifactItems>

<artifactItem>

<groupId>com.github.jsimone</groupId>

<artifactId>webapp-runner</artifactId>

<version>8.0.24.0</version>

<destFileName>webapp-runner.jar</destFileName>

</artifactItem>

</artifactItems>

</configuration>

</execution>

</executions>

</plugin>

</plugins>

</build>

## [Run your application](https://devcenter.heroku.com/articles/java-webapp-runner#run-your-application)

To build your application simply run:

$ mvn package

And then run your app using the java command:

$ java -jar target/dependency/webapp-runner.jar target/\*.war

### [**Create a Procfile**](https://devcenter.heroku.com/articles/java-webapp-runner#create-a-procfile)

You declare how you want your application executed in Procfile in the project root. Create this file with a single line:

web: java $JAVA\_OPTS -jar target/dependency/webapp-runner.jar --port $PORT target/\*.war

## [Visit your application](https://devcenter.heroku.com/articles/getting-started-with-spring-mvc-hibernate#visit-your-application)

You’ve deployed your code to Heroku, and specified the process types in a Procfile. You can now instruct Heroku to execute a process type. Heroku does this by running the associated command in a [dyno](https://devcenter.heroku.com/articles/dynos) - a lightweight container which is the basic unit of composition on Heroku.

Let’s ensure we have one dyno running the web process type:

$ heroku ps:scale web=1

You can check the state of the app’s dynos. The heroku ps command lists the running dynos of your application:

$ heroku ps

=== web: `java $JAVA\_OPTS -jar target/dependency/jetty-runner.jar --port $PORT target/\*.war`

web.1: up for 6m

Here, one dyno is running.

We can now visit the app in our browser with heroku open.

$ heroku open

Opening pure-window-800... done