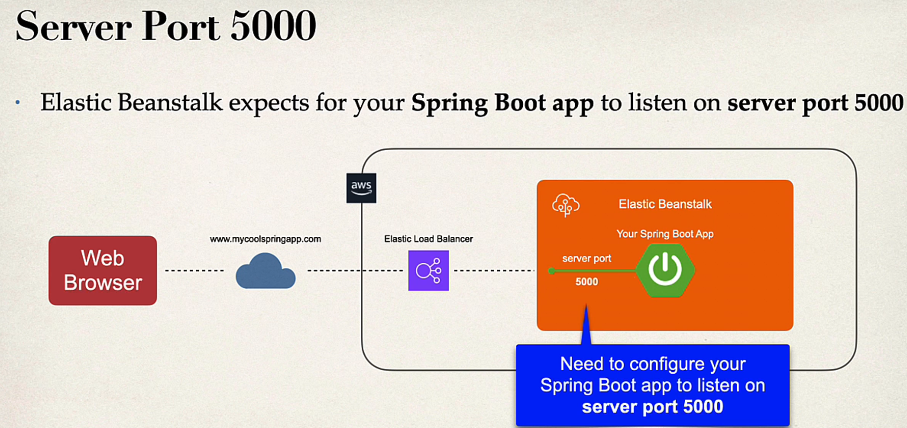
Download the app and import it into our IDE.   
And then we'll package the application using Maven.   
Next, we'll create a new application in Elastic Beanstalk,   
and then we'll upload our Spring Boot jar file to Elastic Beanstalk

Now in terms of spring boot rest API deployment.  
In this course, we'll deploy an MVC app, but you could also deploy a Rest API.  
For rest API, you simply follow the exact same deployment steps.



Now a couple of things regarding Elastic Beanstalk.  
Elastic Beanstalk expects for your Spring Boot app to listen on server port 5000.

So, in your web browser, when you go visit your application, it goes to the internet, and then it makes it over to the AWS location inside of AWS. They have an elastic load balancer that will route traffic to your application. And our application will be deployed in Elastic Beanstalk. So, we'll have our Spring Boot app. And we need to configure our application to listen on port 5000  
So from the public internet AWS will accept requests for your application.  
And then internally using the Elastic Load Balancer it'll route the traffic to port 5000. So, your application must listen on port 5000. Now by default Spring boot listens on port 8080.

So, we simply must configure our Spring Boot application to listen on server port 5000.

So we know that we need to configure our app to listen on port 5000.

We have two options.

Option one is setting an environment variable in Elastic Beanstalk.

So in Elastic Beanstalk you can configure an environment variable.

And we can set server port equals 5000.

So that'll have our spring boot application listening on server port 5000.

Another option is that we can make use of a property in Application.properties. So in Application.properties we'll set server port equals 5000. Now, this server port that's a well-known environment variable or a well-known property that Spring Boot can recognize.