Login to AWS console and search “Elastic beanstalk” let's go ahead and create an application and configure our environment. We’ll choose web server environment  
This allows us to run a web application.

A screenshot of a computer

AI-generated content may be incorrect.

So, our Spring boot application will be able to handle or serve HTTP requests. We’ll move down here, and we'll give the application name. Now you can give any name here. I'll simply call it hellodemo.

A screenshot of a computer

AI-generated content may be incorrect.

Then I'll move down to the environment Information. For the domain, we can choose any name, simply make it unique, you can also check the availability of the name that you want.

A blue and black text

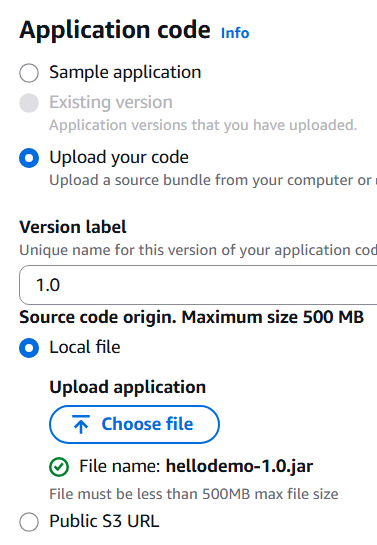
AI-generated content may be incorrect.

Now I can move down to the platform section. I'll use the managed platform. I'll choose the Java platform.

A close-up of a white background

AI-generated content may be incorrect.

I moved down to application code. Instead of using a sample application, I'll simply upload my code, so we chose the option here to upload your code. I'll give the version number of 1.0. And then I'll choose a local file. And I'll upload my file.



Select target folder and hellodemo1.0. jar

I'll move down for the presets. I'll choose the configuration presets of single instance to keep me in the free tier. click on next.

A close-up of a computer screen

AI-generated content may be incorrect.

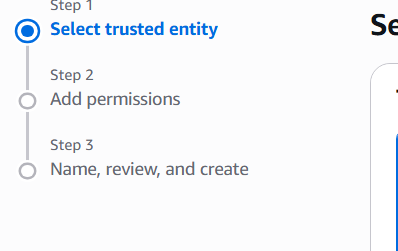
For the service role. I’ll go ahead and create and use a new service role. Now you may wonder what a service role is. A service role is used by a service to perform actions on our behalf. In our case, we're going to make use of the EC2 service Elastic Compute. We will allow this to deploy an application for us on Elastic Beanstalk. So, we'll set up the appropriate roles and permissions for that

A screenshot of a computer

AI-generated content may be incorrect.

Post created above 3 steps go ahead following   
A screenshot of a computer

AI-generated content may be incorrect.

So, I'll just move down here, and I need to choose an EC2 instance profile. Right now, it's empty. Create role complete below steps  


Refresh and select the created role  
A screenshot of a computer

AI-generated content may be incorrect.

step four configure instance traffic and scaling. And for the instances here we want to select the item here for the root volume boot device we want to select the volume type of general purpose three SSD. And this is the latest generation of the general-purpose SSD volumes. It's faster and cheaper than the previous versions. And then we can kind of scroll down here and keep all the other defaults. And then we'll click on next

Step five configuring updates, monitoring and logging. And what I'll do here is just kind of keep the defaults scroll down to the bottom.

And I want to set up some environment variables, because we know that we need to add an environment variable for the actual server port. So, we choose the option here to add environment variable.

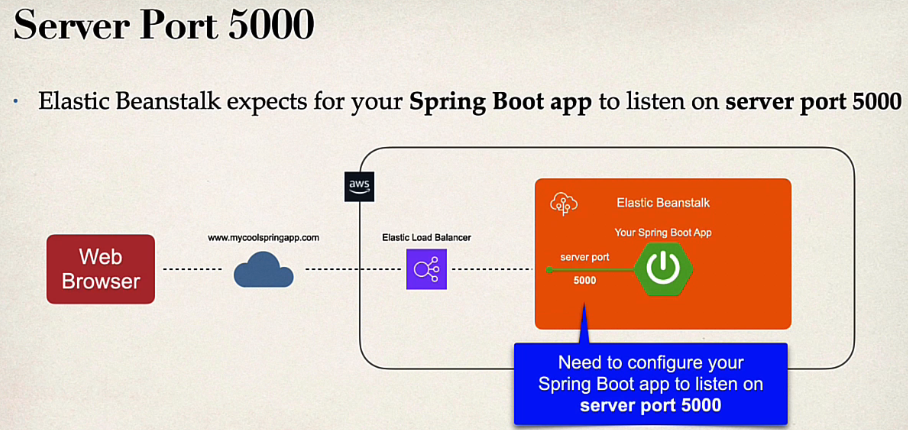
A screenshot of a computer

AI-generated content may be incorrect.

We give the name server\_port and be sure to type it in exactly like I have here on the screen the value of 5000.

And now just as a refresher, remember that Elastic Beanstalk expects for your Spring Boot app to listen on server port 5000.

So, the web request will come in from the browser through the internet to the Elastic Load Balancer in AWS. And it's expecting for your application to be listening on port 5000.



So, we must configure our Spring Boot app to listen on port 5000. And we can do that by setting up an environment variable for server port and giving it the value of 5000. And that's the main reason why we added this environment variable.

Let's go ahead and click on next. And now just kind of as a review as a checkpoint. I’ll kind of go through here and look at the information that I had set up already. That looks good. And then moving down confirming that we have the server port of 5000.

That looks good. Now we can go ahead and hit submit. And now it'll start launching our environment. Now this deployment process can take a while. It could take up to 5 to 10 minutes.

We can access our application by clicking on the link here for the domain and success.

It says Welcome to Spring Boot on AWS.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.