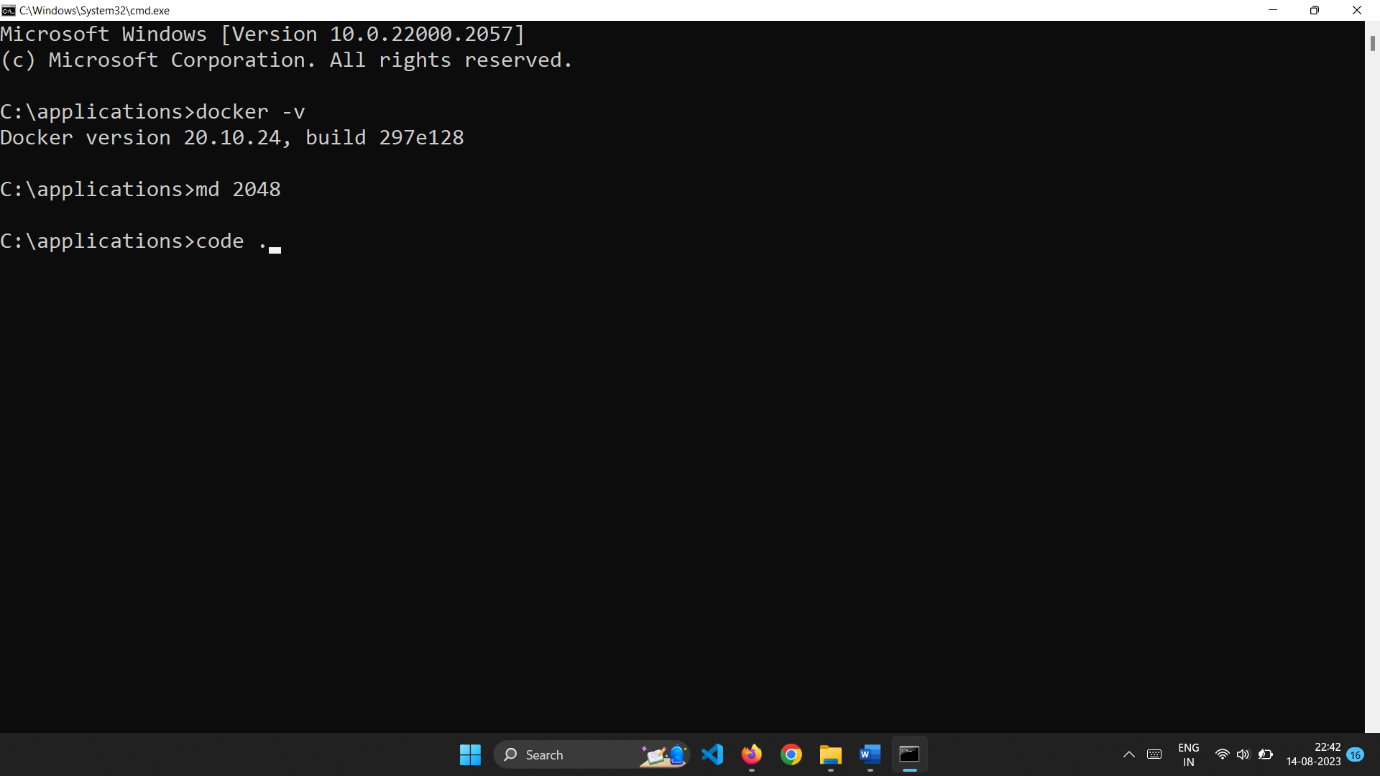
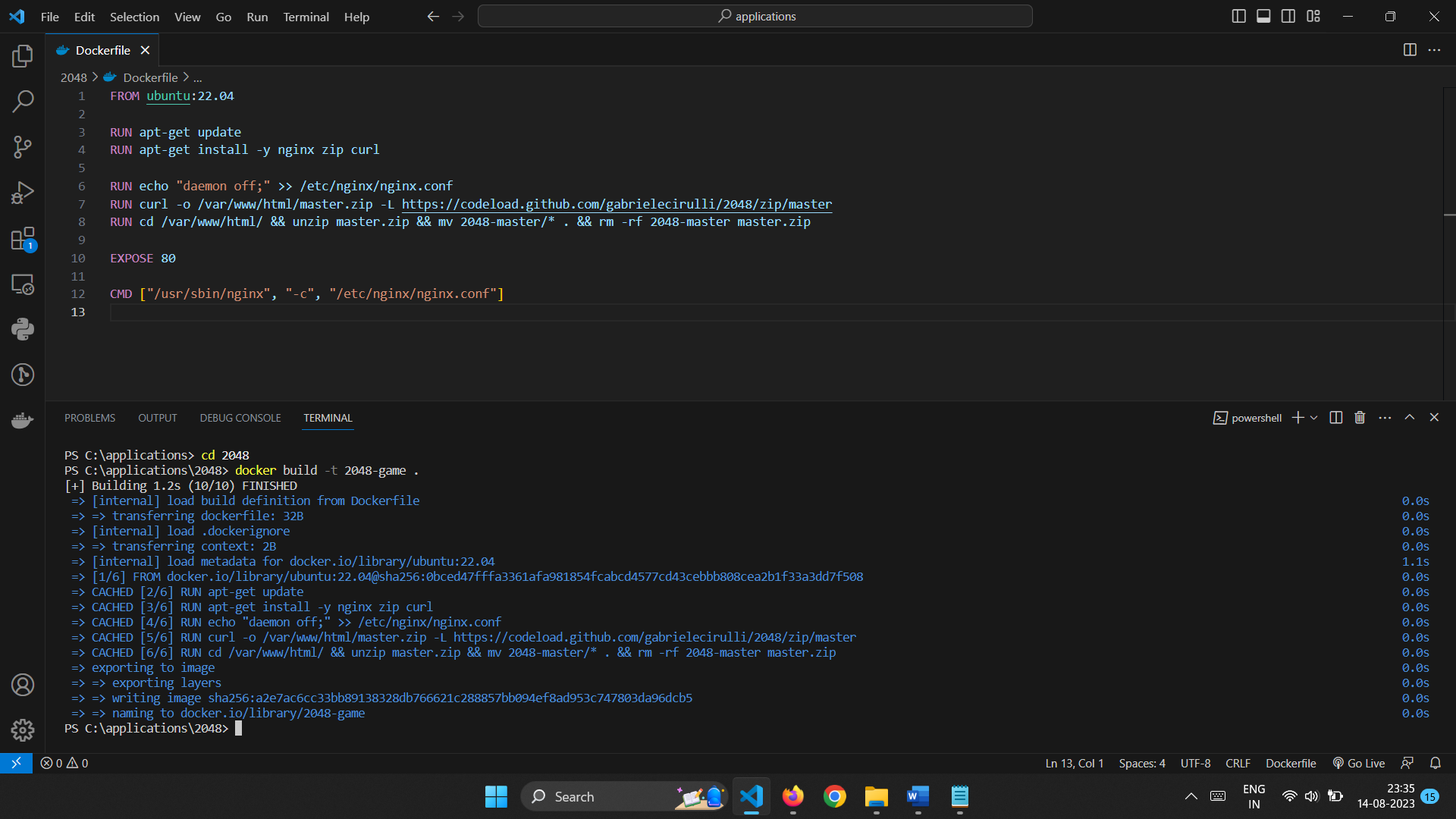
**Create a game using Docker and deploy in AWS Elasticbeanstalk**

Install docker in your local system and check if installed or not using docker -v .



Now create a Dockerfile.

It is used to install nginx server and get the 2048 game source code from github.

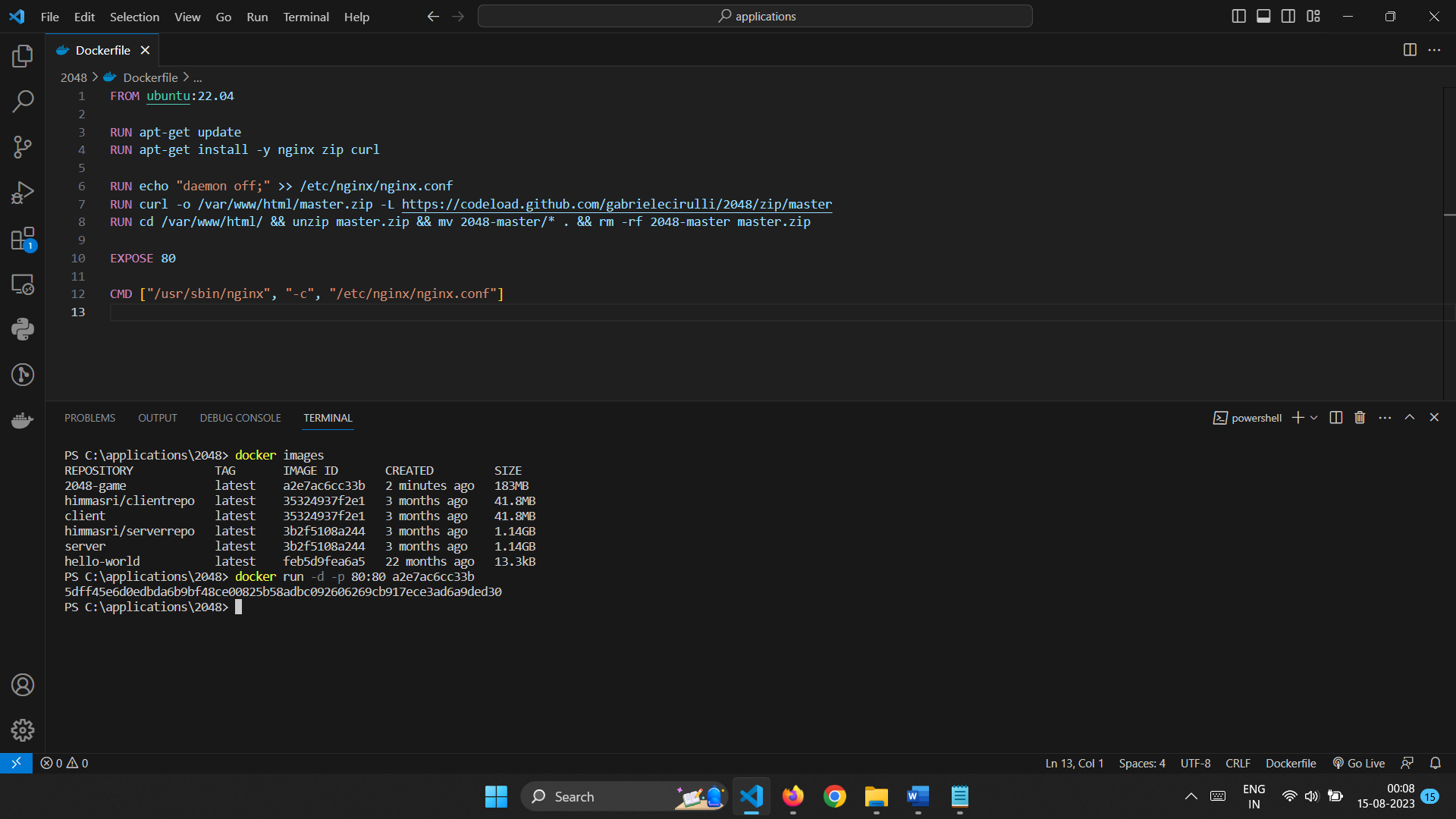


After creation of Dockerfile,now we can create an image from it.

It can be implemented using docker build command.

To view whether an image is created we can use docker images command.

This will list the images being created.

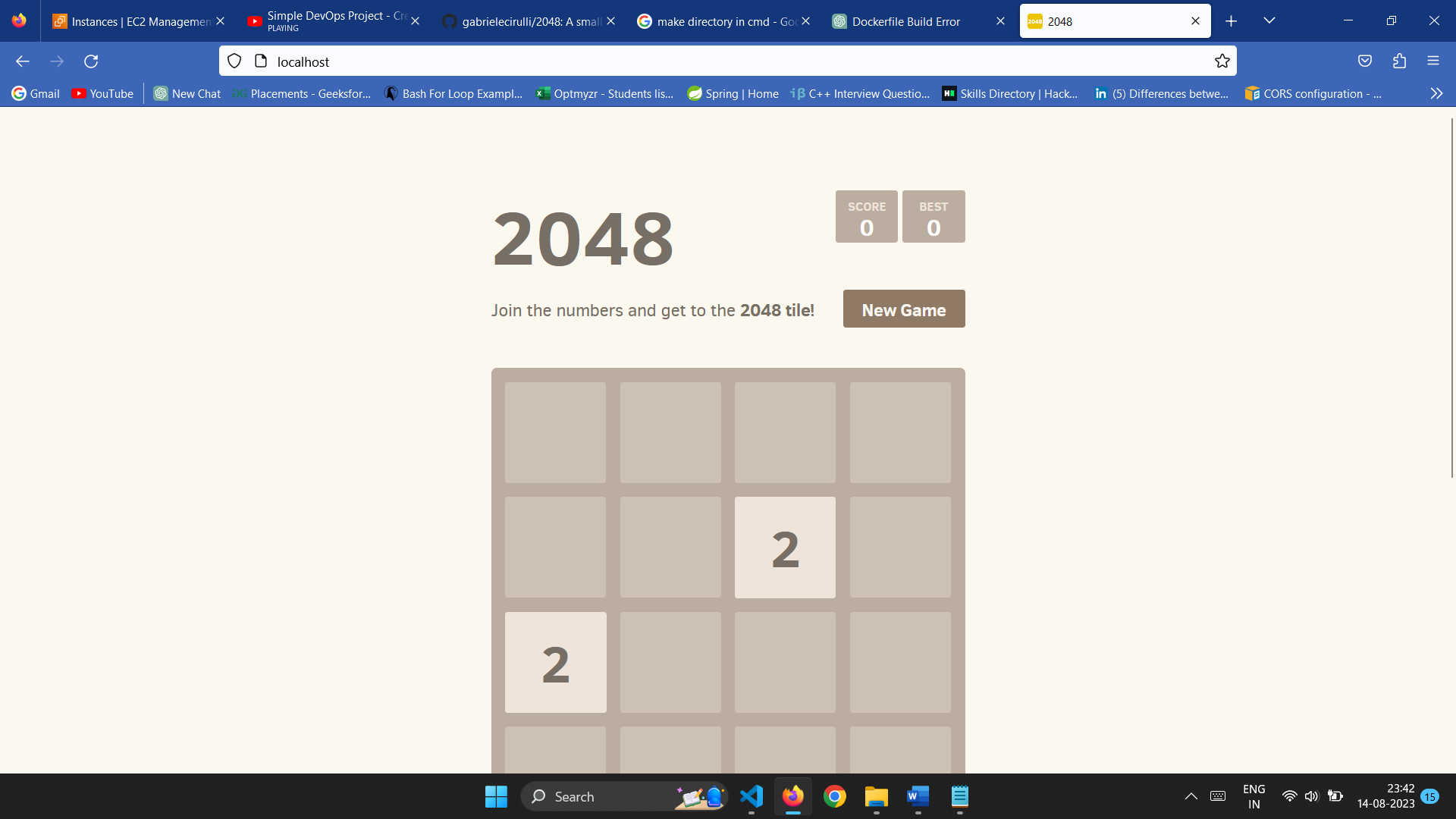


Now to run container from it we need to use docker run command.

We can specify the port where the container should run.

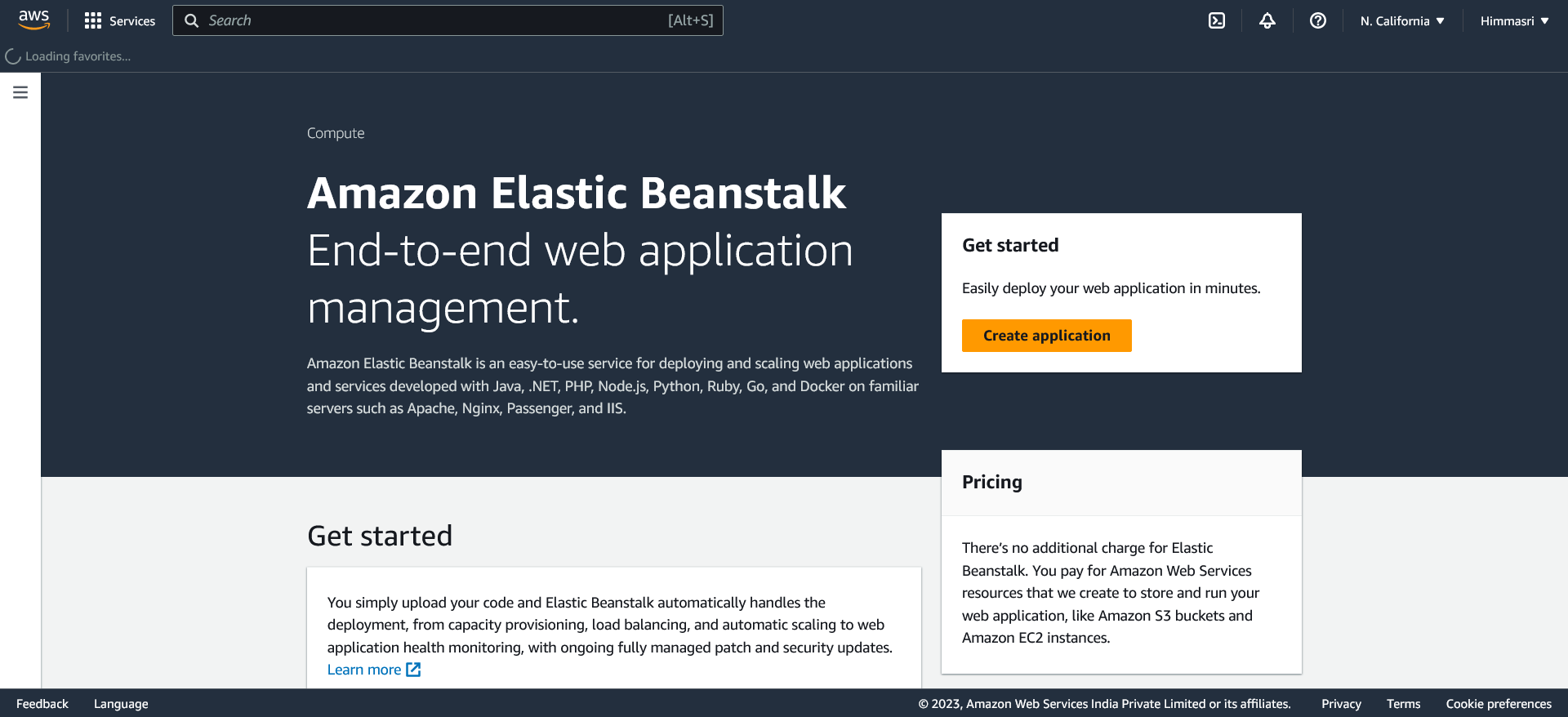
Here I’m using port 80.

If we go to the localhost we can see the application.



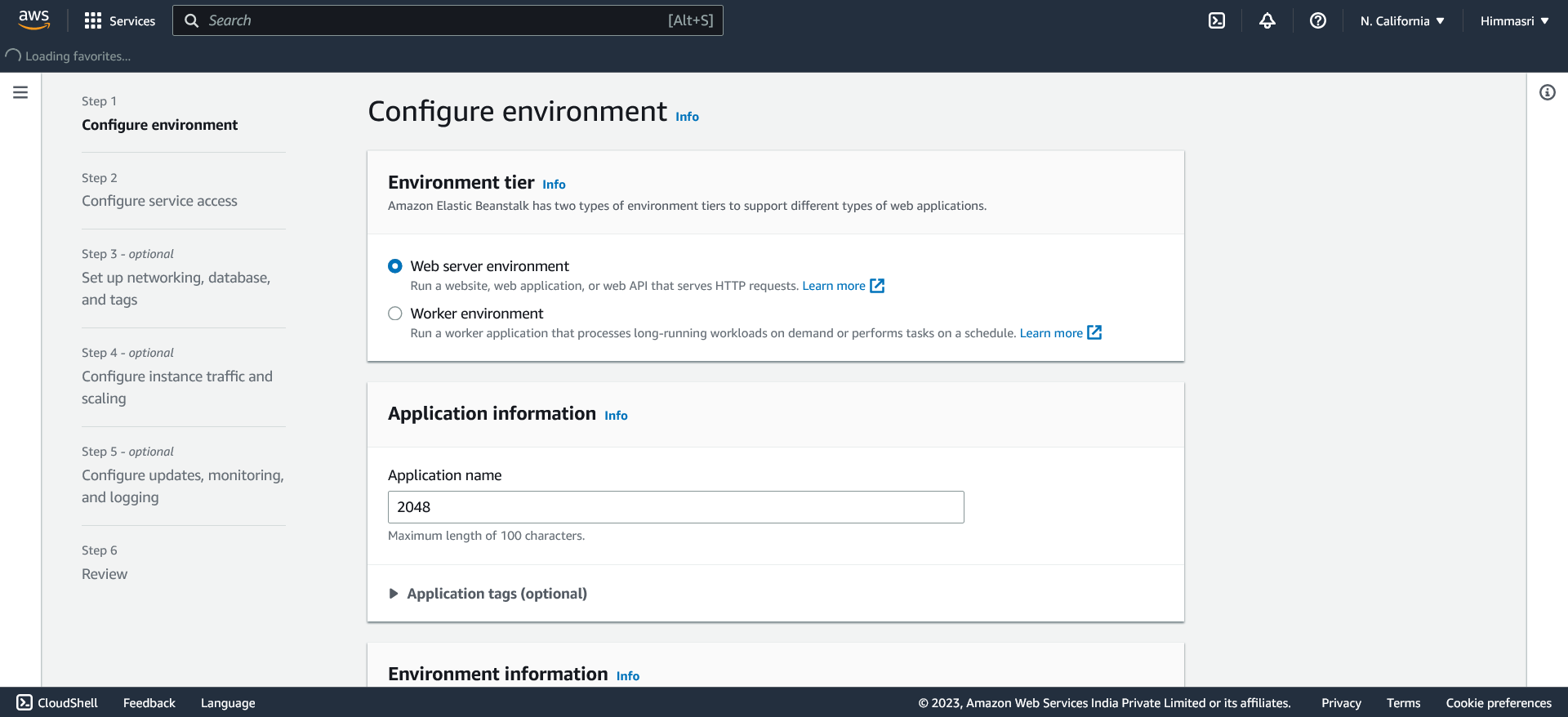
But it is running in my localhost to make it available to everyone I’m using Elastic beanstalk to deploy my application.

It allows to host our web application with ease.All we need to do is give our code.

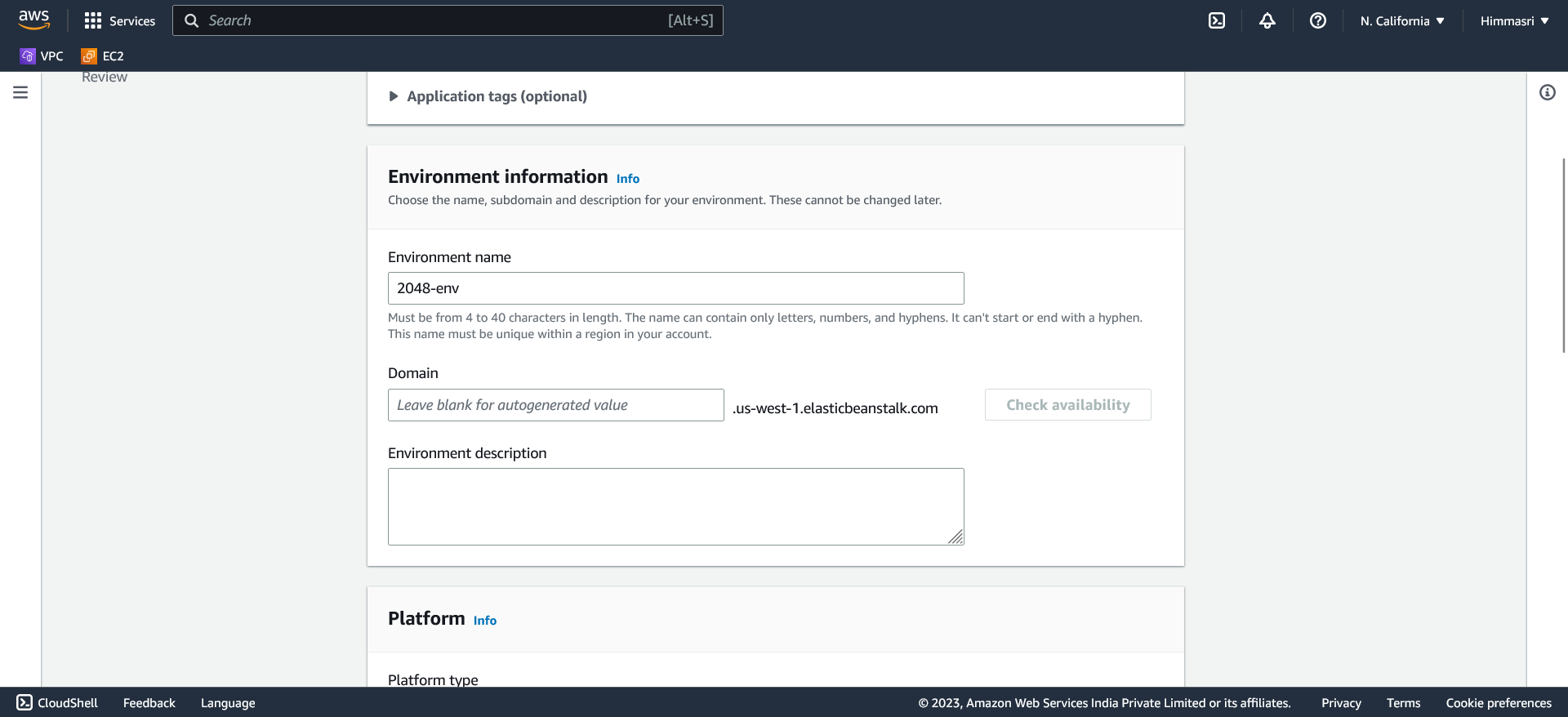


We need to select the type of enviornment.

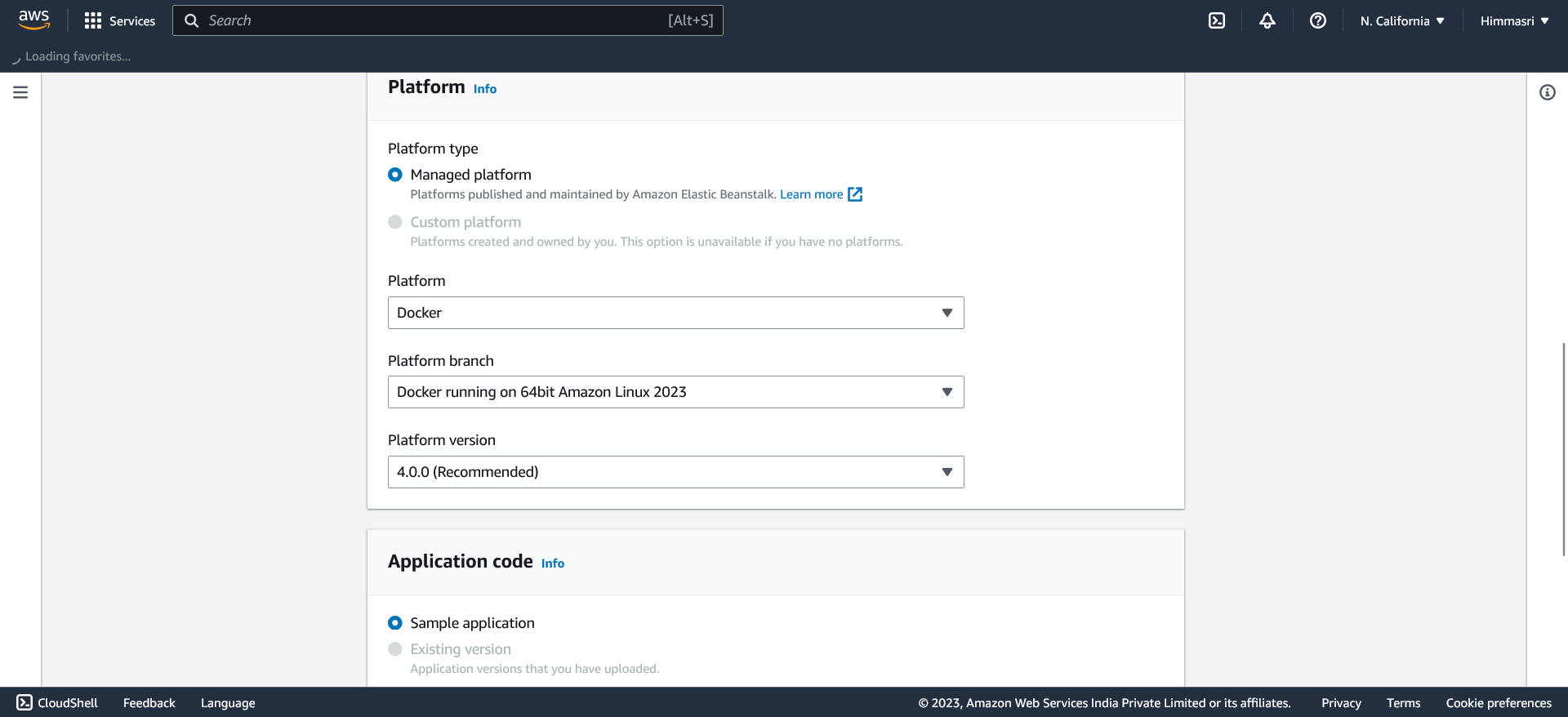
Since 2048 is an web application web server environment is choosen.



Give application a name.



Select the platform and it’s version.

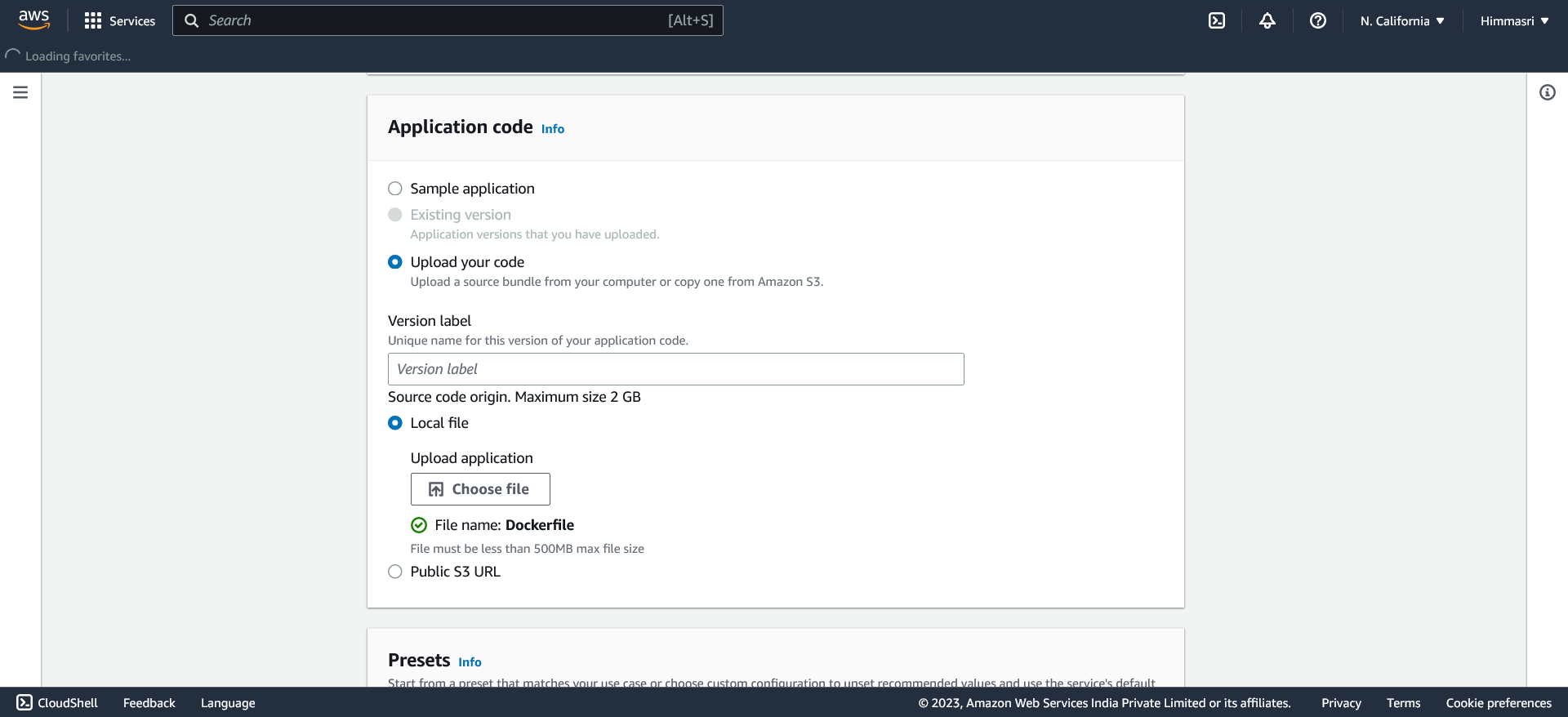


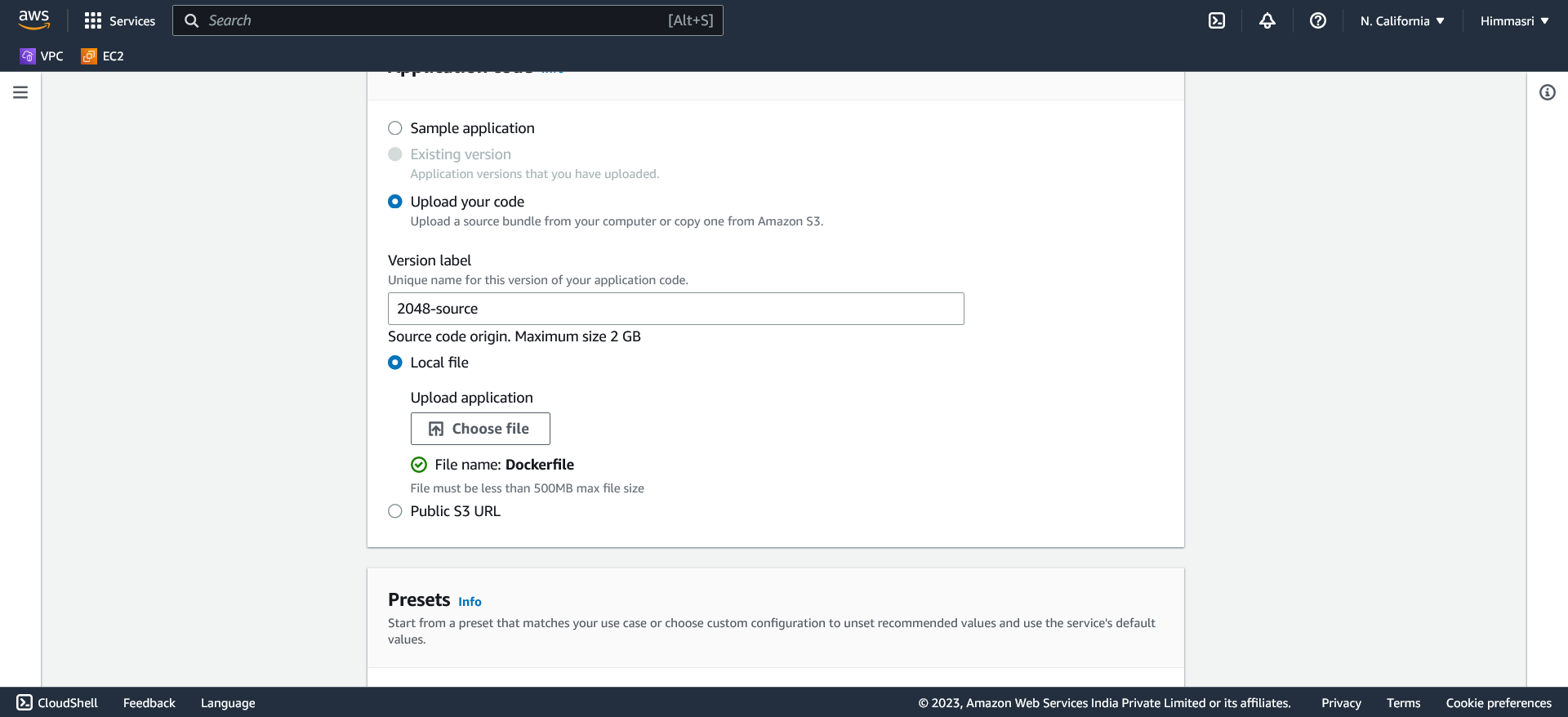
We can upload our code.

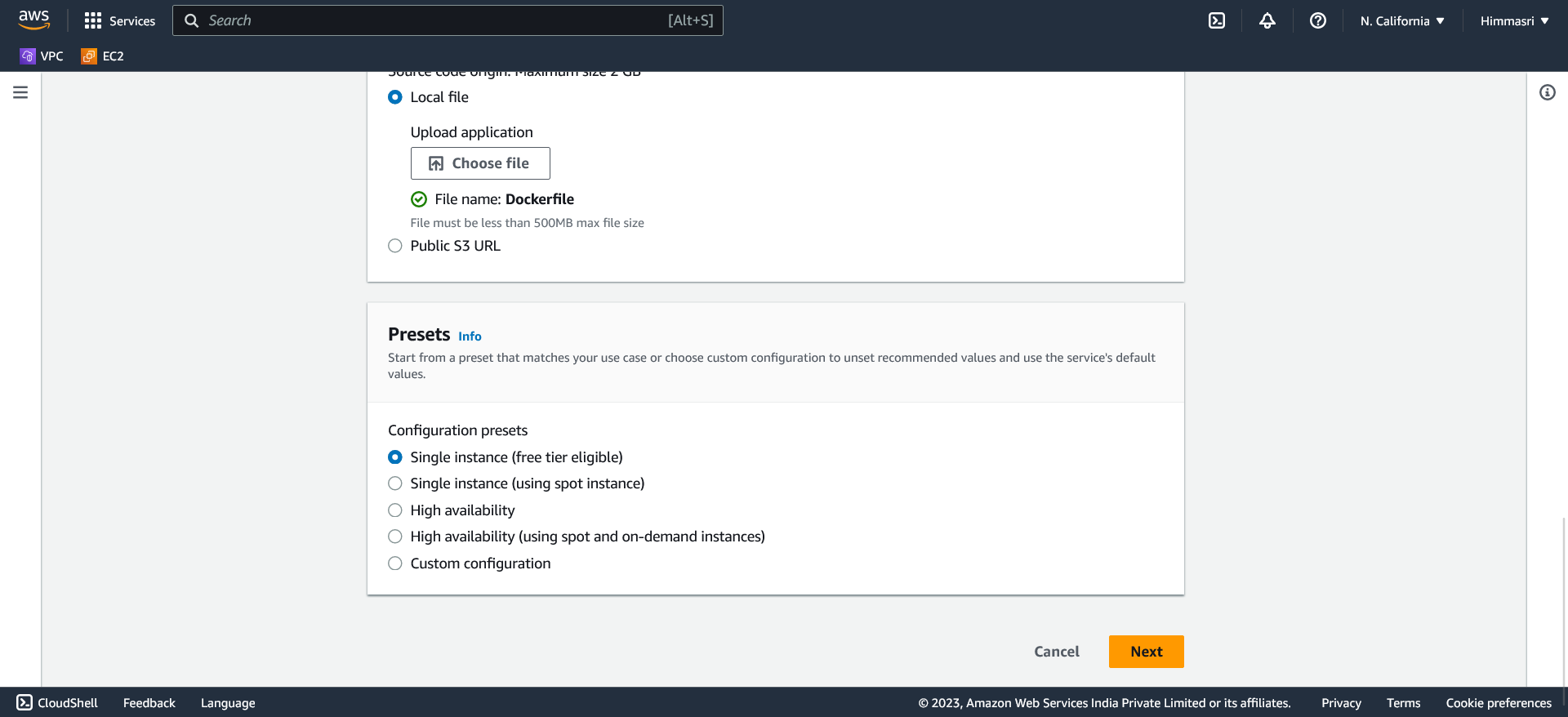
Here I’m uploading my Dockerfile.

I have selected the single instance because it’s uder free tier eligible.

We can choose depending on our application requirements.



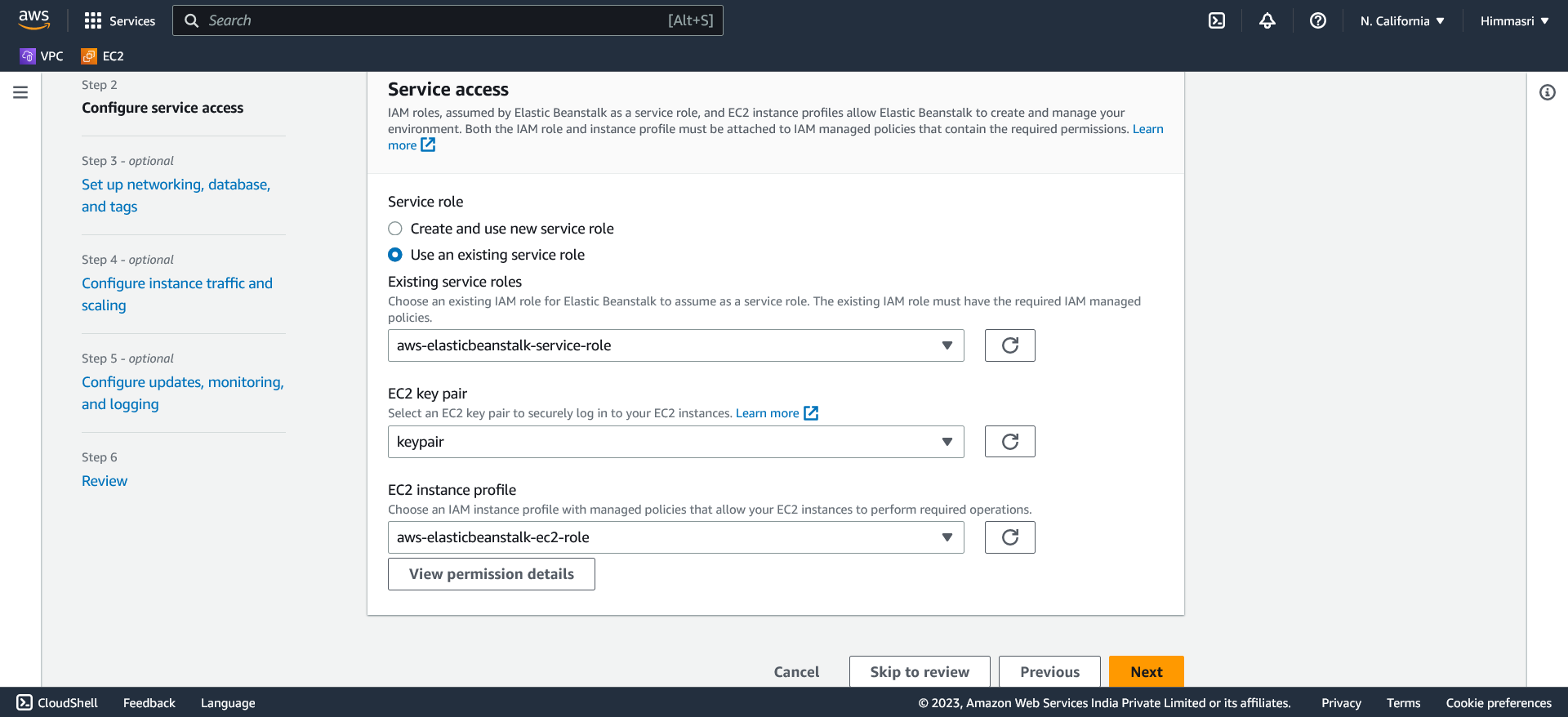




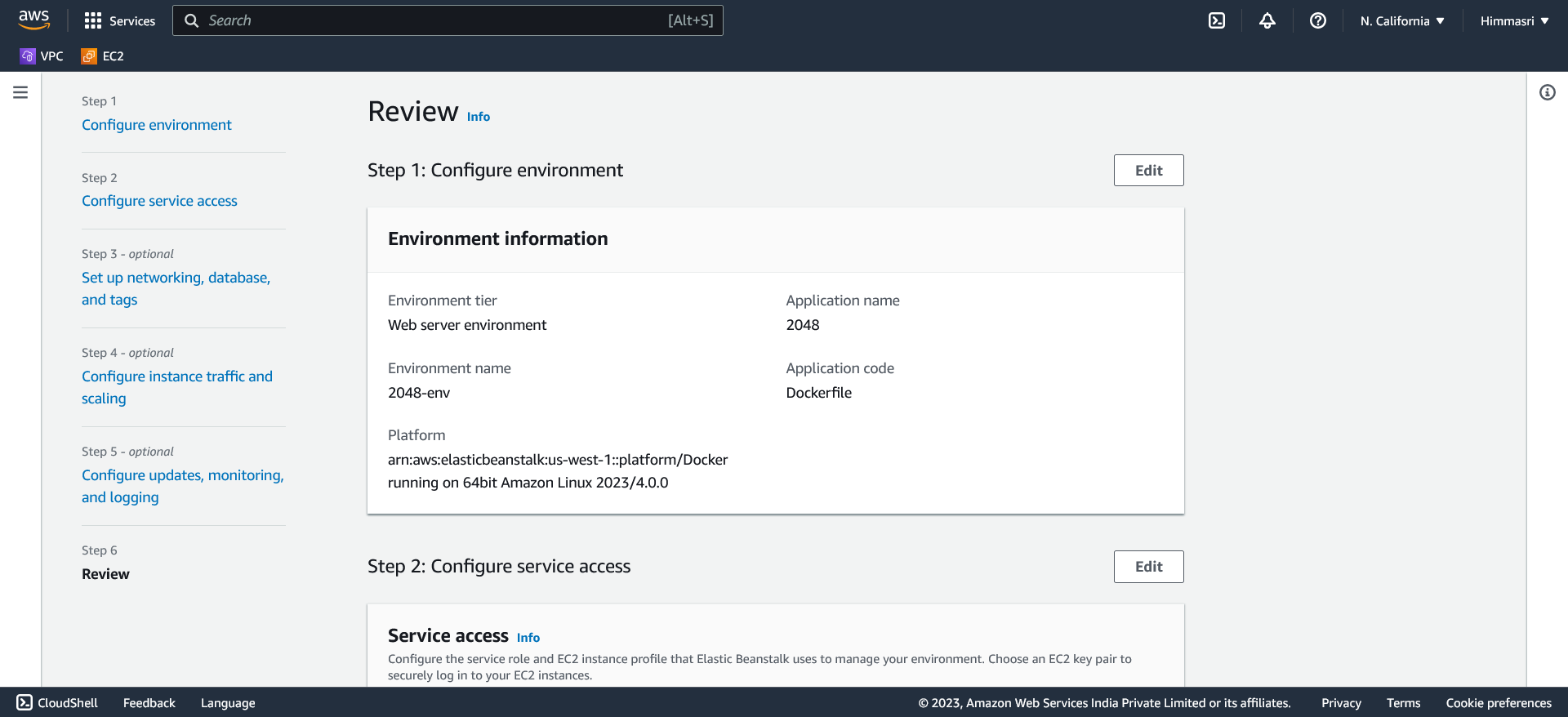
Select the service role as aws-elasticbeanstalk.

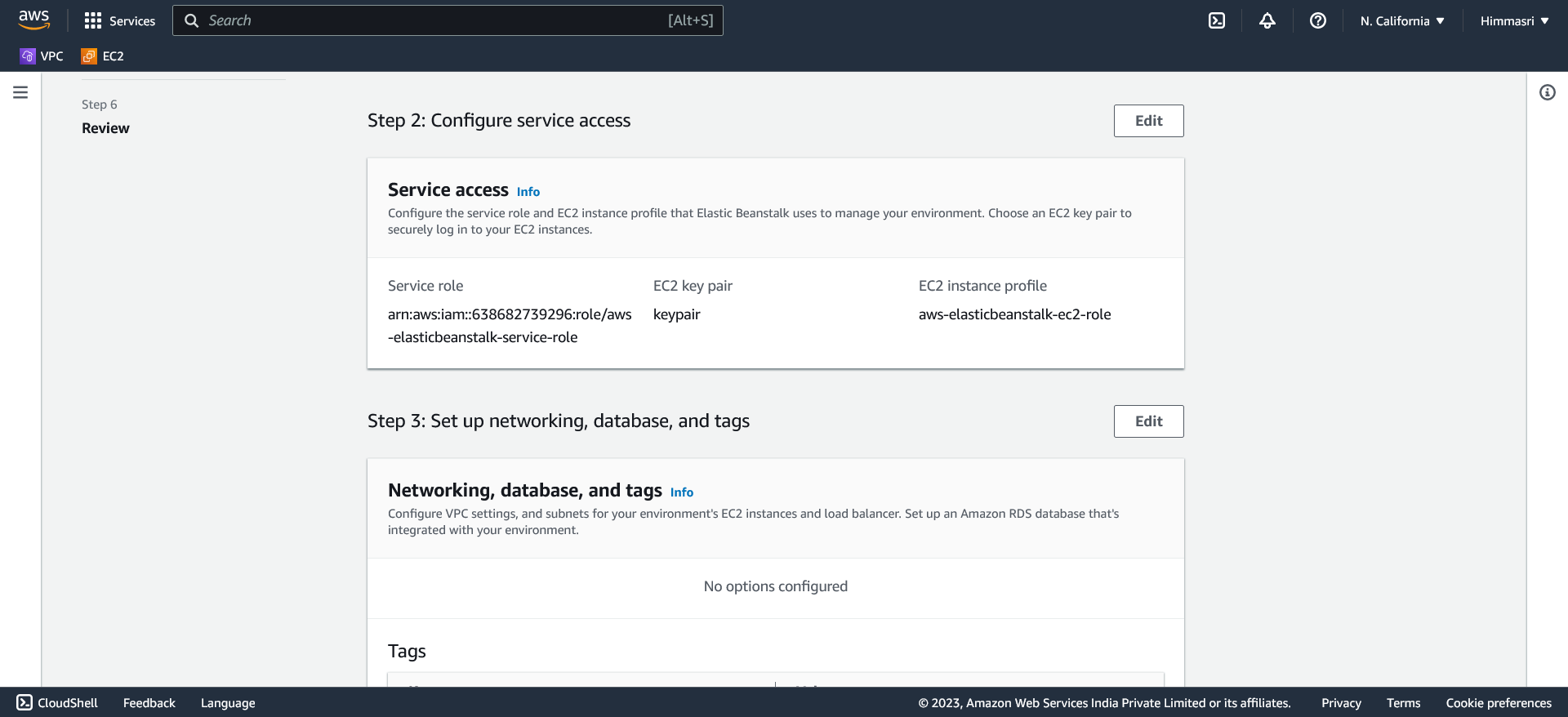
If it doesn’t exist we can create a new service role.

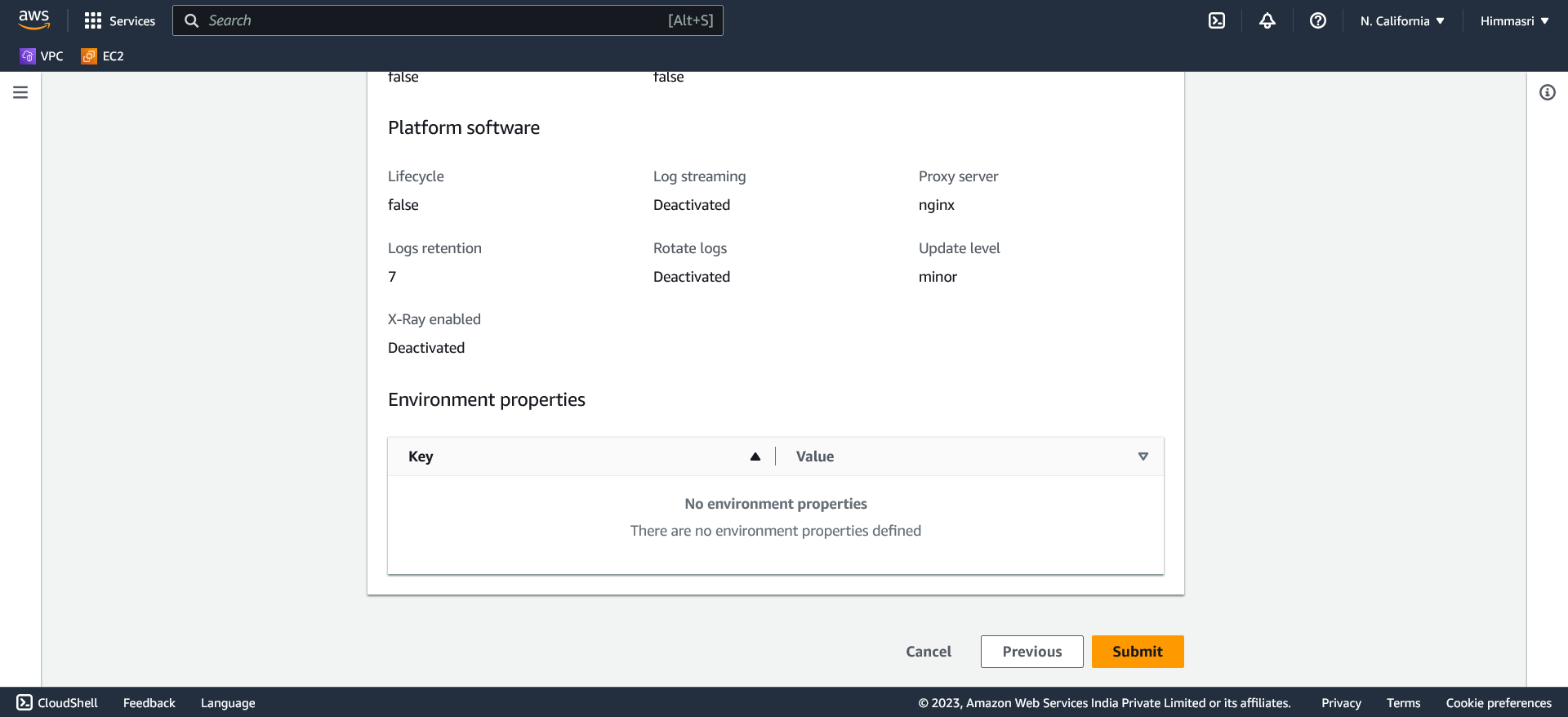
We can choose the keypair of EC2.



Review the overall configuration of the environment and create.



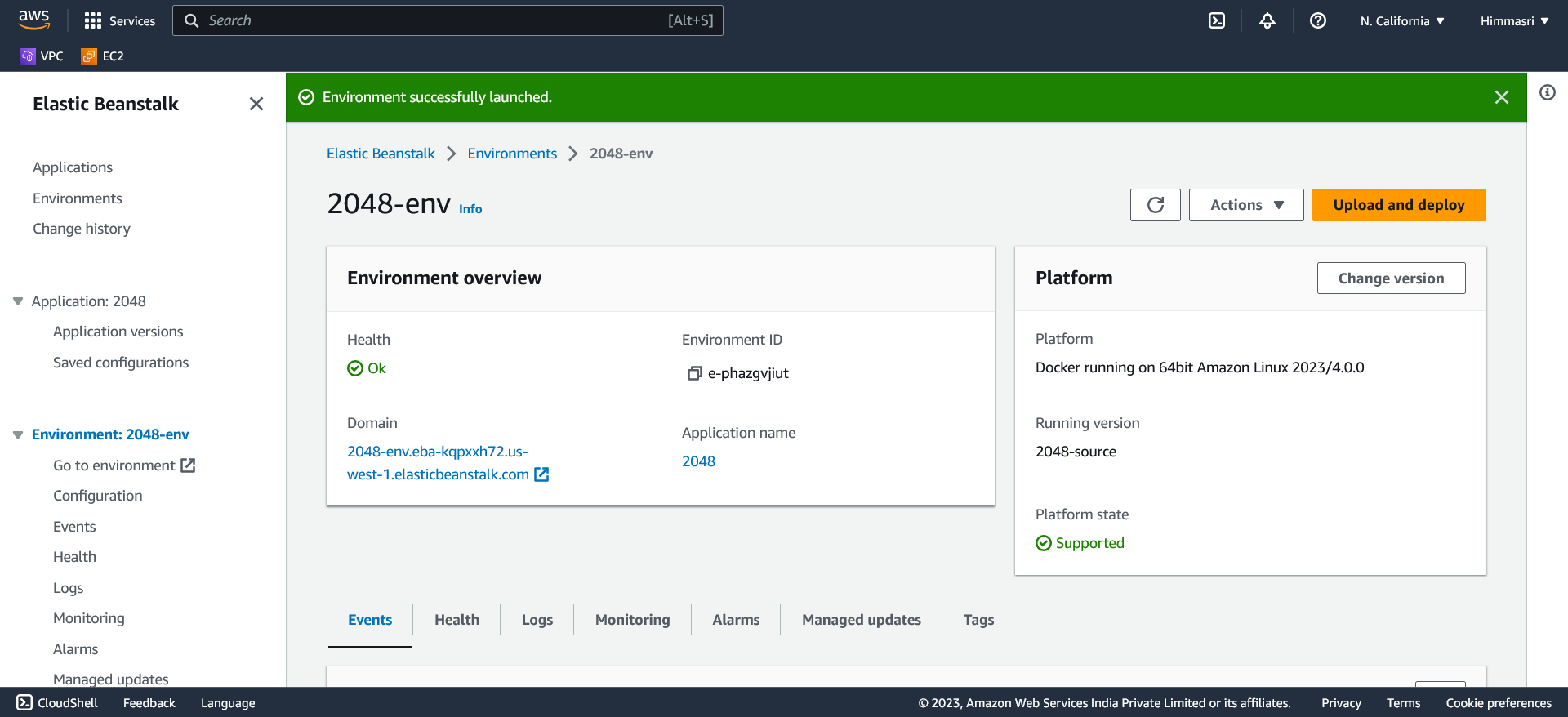




After environment successfully created we get a domain.

This is the link to your application.

We can copy and paste it in the browser to access the application.



We have successfully launced our web application using Docker and AWS Elastic beanstalk globally.

