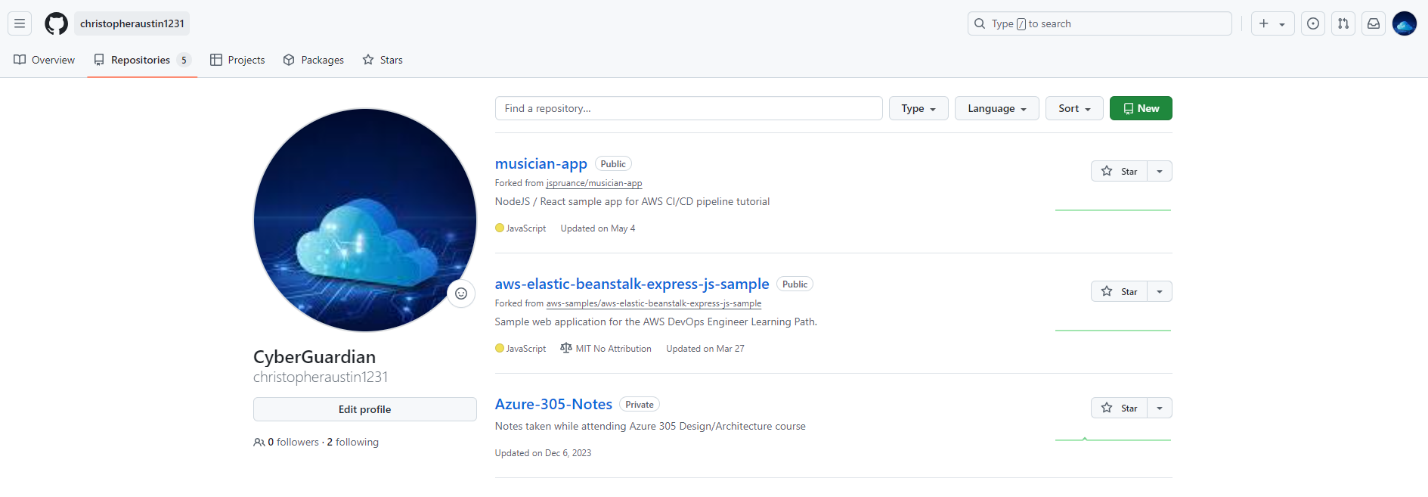
**Deploy application to AWS utilizing elastic beanstalk and AWS Code Pipeline**

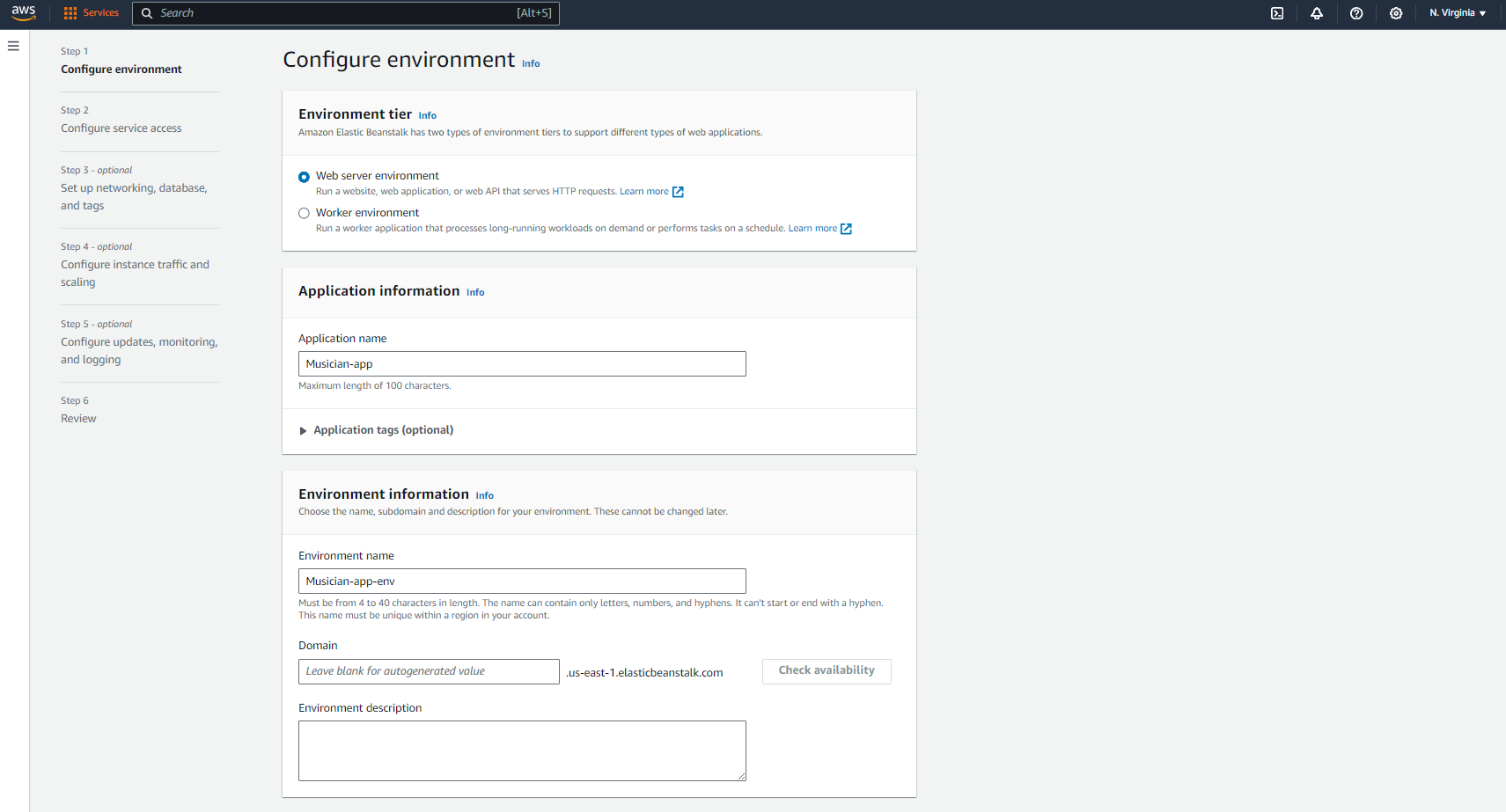
For this lab we will be using a sample node js application from GitHub. In order to do this, you will need to find an application that you wish to use and Fork the repo into your own Github account.

**Fork Repo in Github**



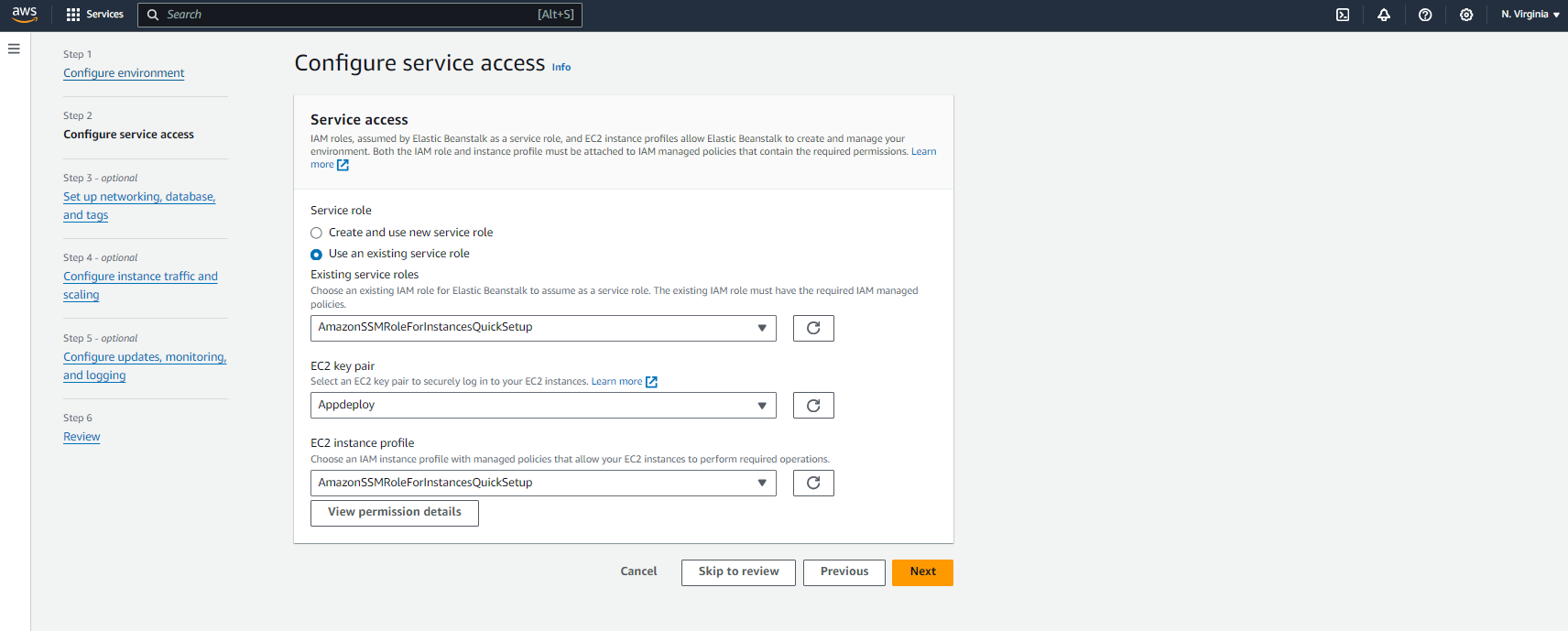
Once you have forked the repo of your choosing you will need to login to the AWS console and navigate to the Elastic Beanstalk service. For security purposes I would suggest not using the root account for this lab but creating an IAM user. There will also be IAM roles created and possibly EC2 key pairs if you choose to run the environment in EC2.

**Login in AWS and navigate to Elastic Beanstalk**

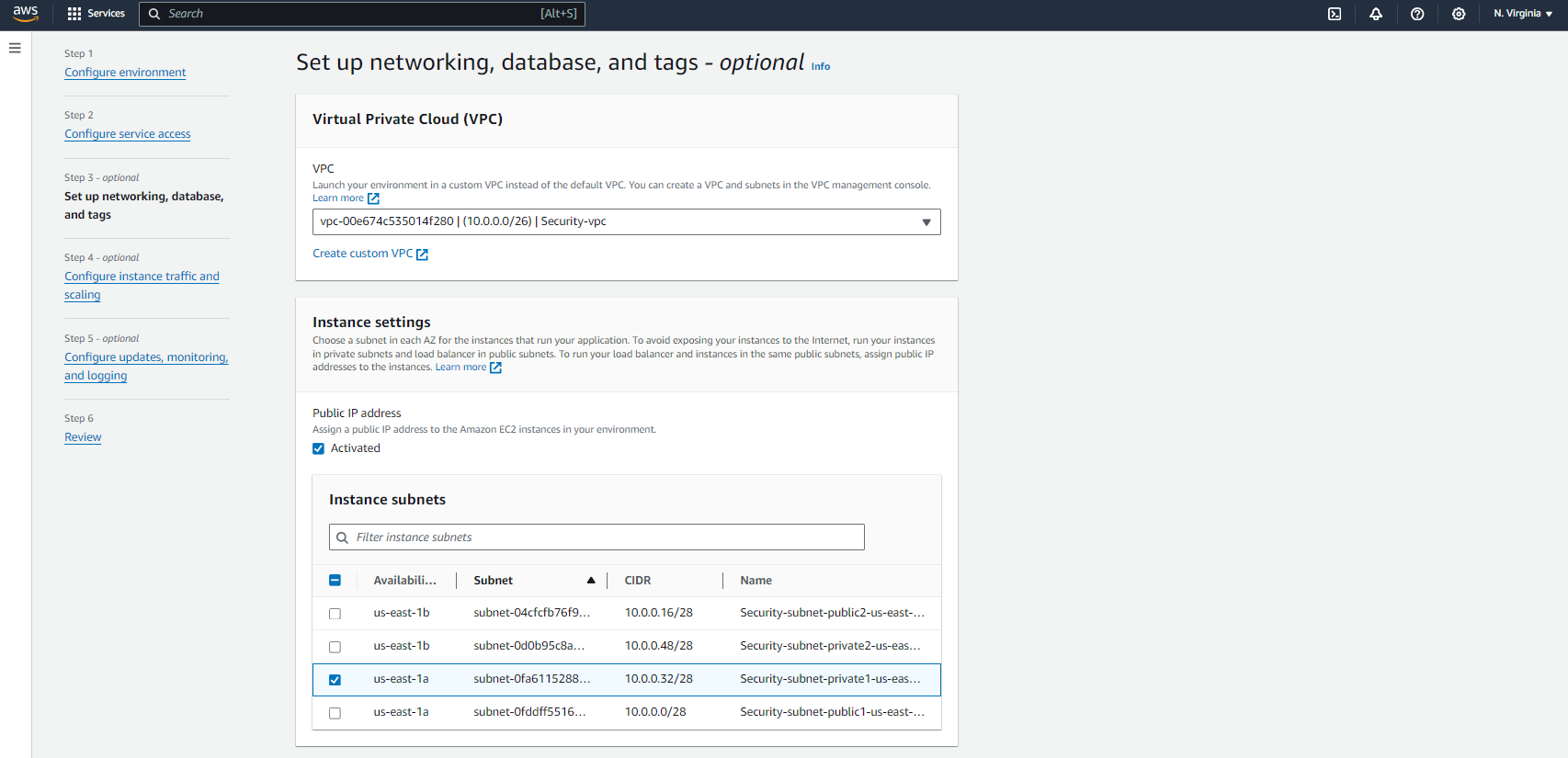


The firs thing you will want to do is to create the application and then create the environment your application will run in. All of this is customizable to how you wish to deploy the application. As mentioned above you can use existing service roles during this lab or you can create new roles.

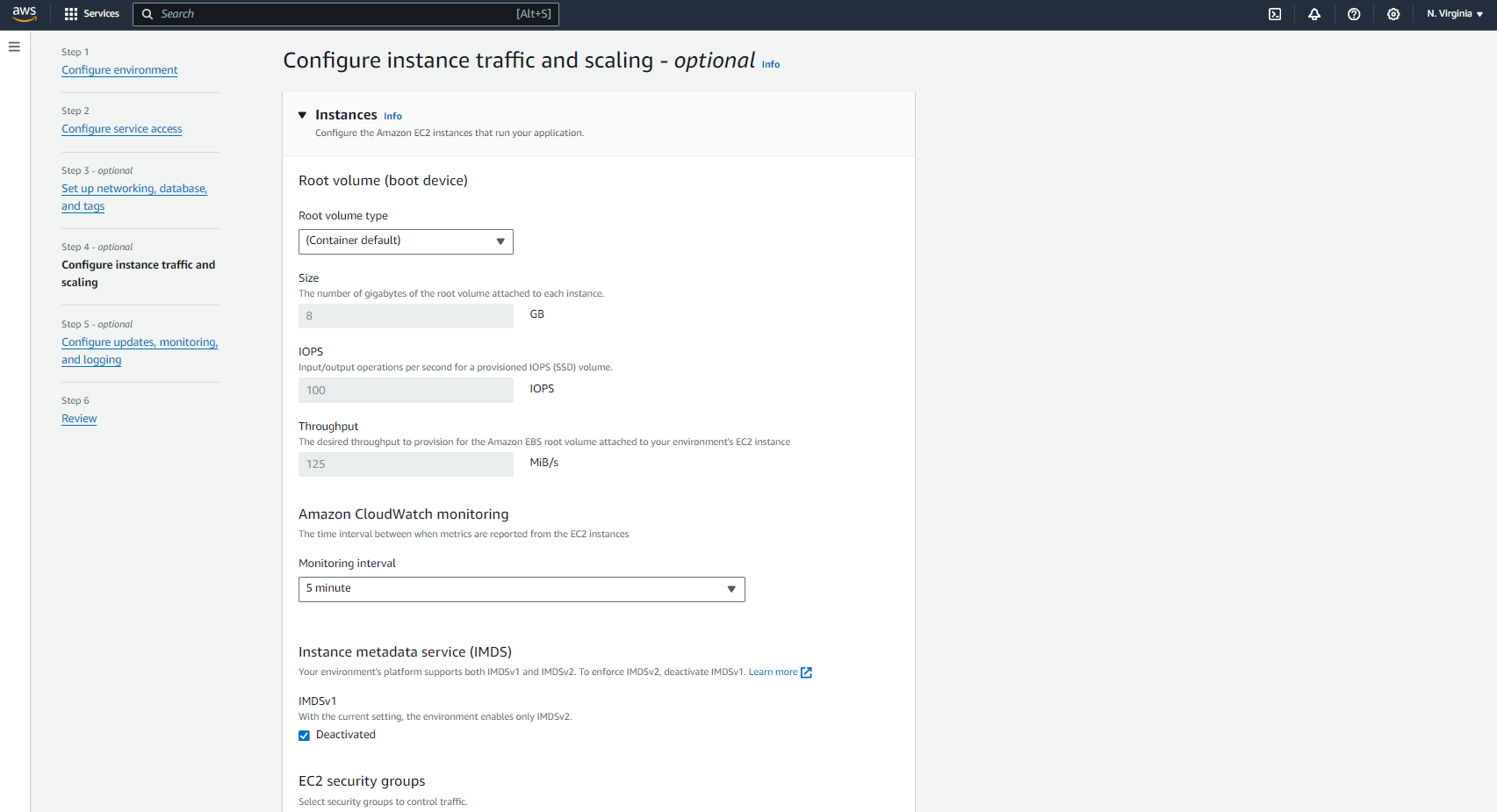
**Continue configuring service access.**

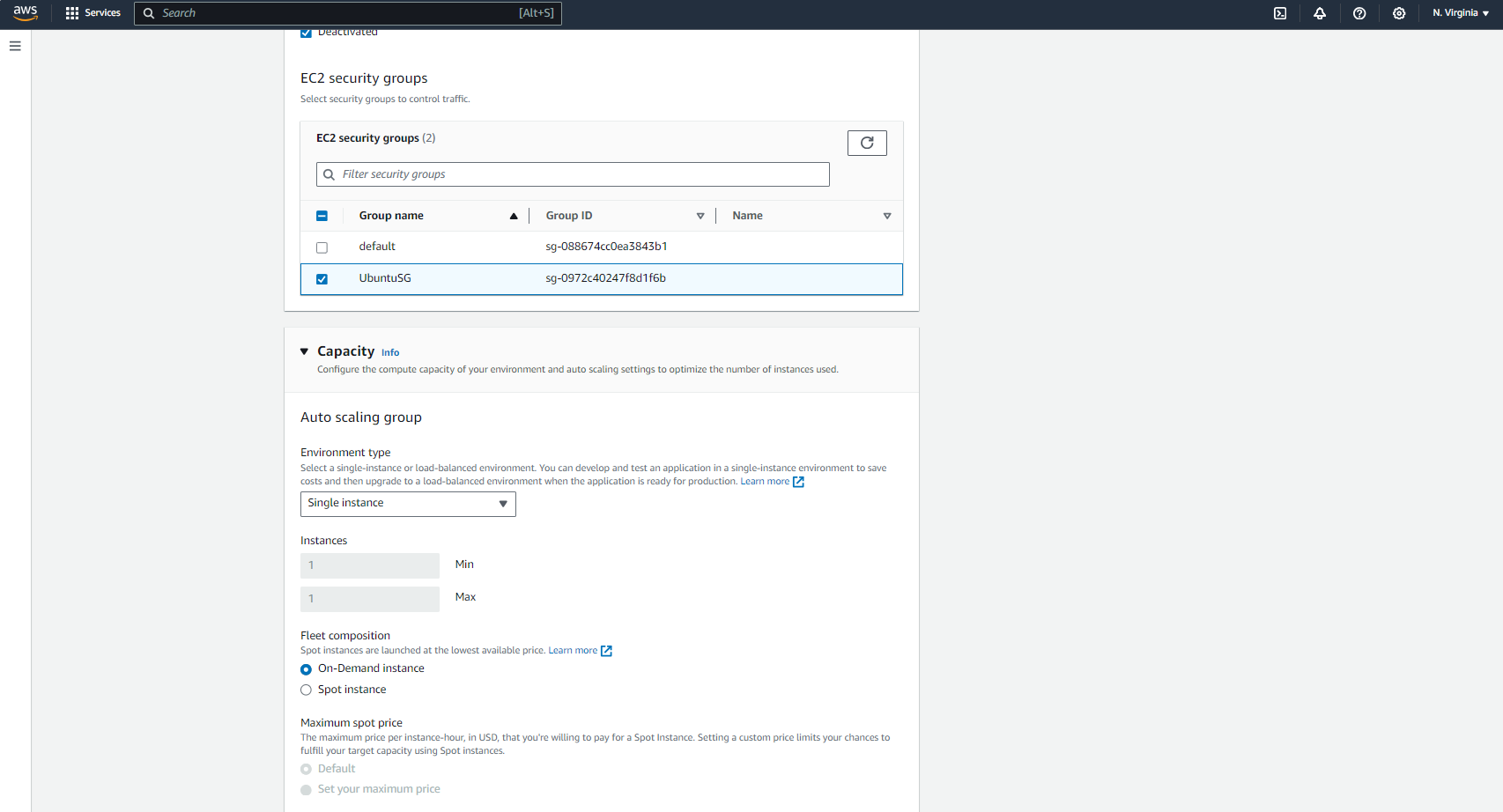


**Set up the networking by selecting a VPC and subnets. In the case of this example, I am using a /26 VPC with public and private subnets. (For lab and testing purposes you can simply use the default VPC, however it is not recommended to use the default VPC in production environments)**

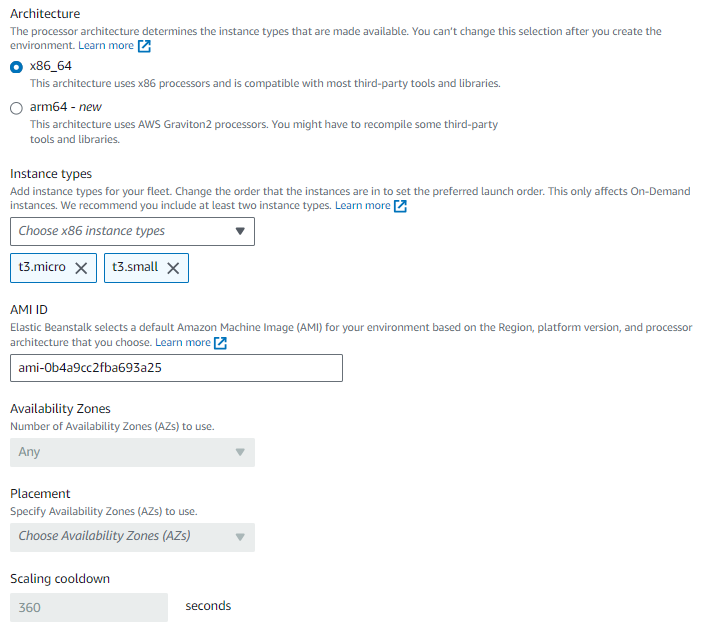


Configure the instance details that will be hosting the application. Here is where you can choose between hosting your application on an EC2 or running it as a container.

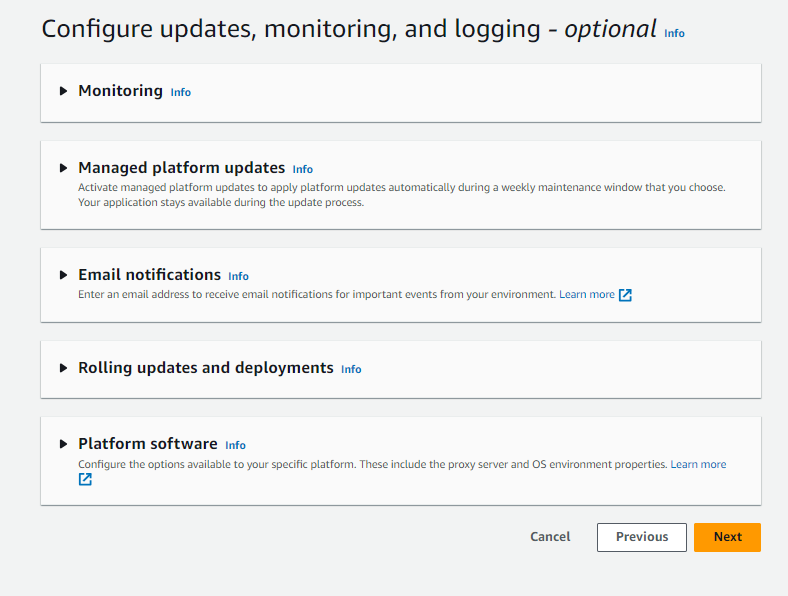




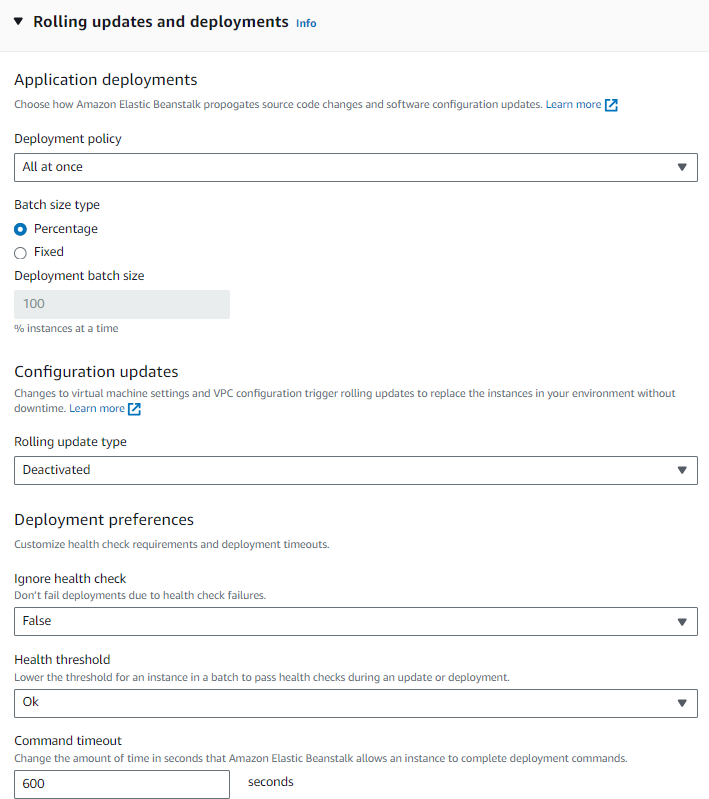
**Follow the prompts. As you can see below I am choosing smaller instances within the free tier.**



**Configure updates and monitoring for your environment as shown below.**

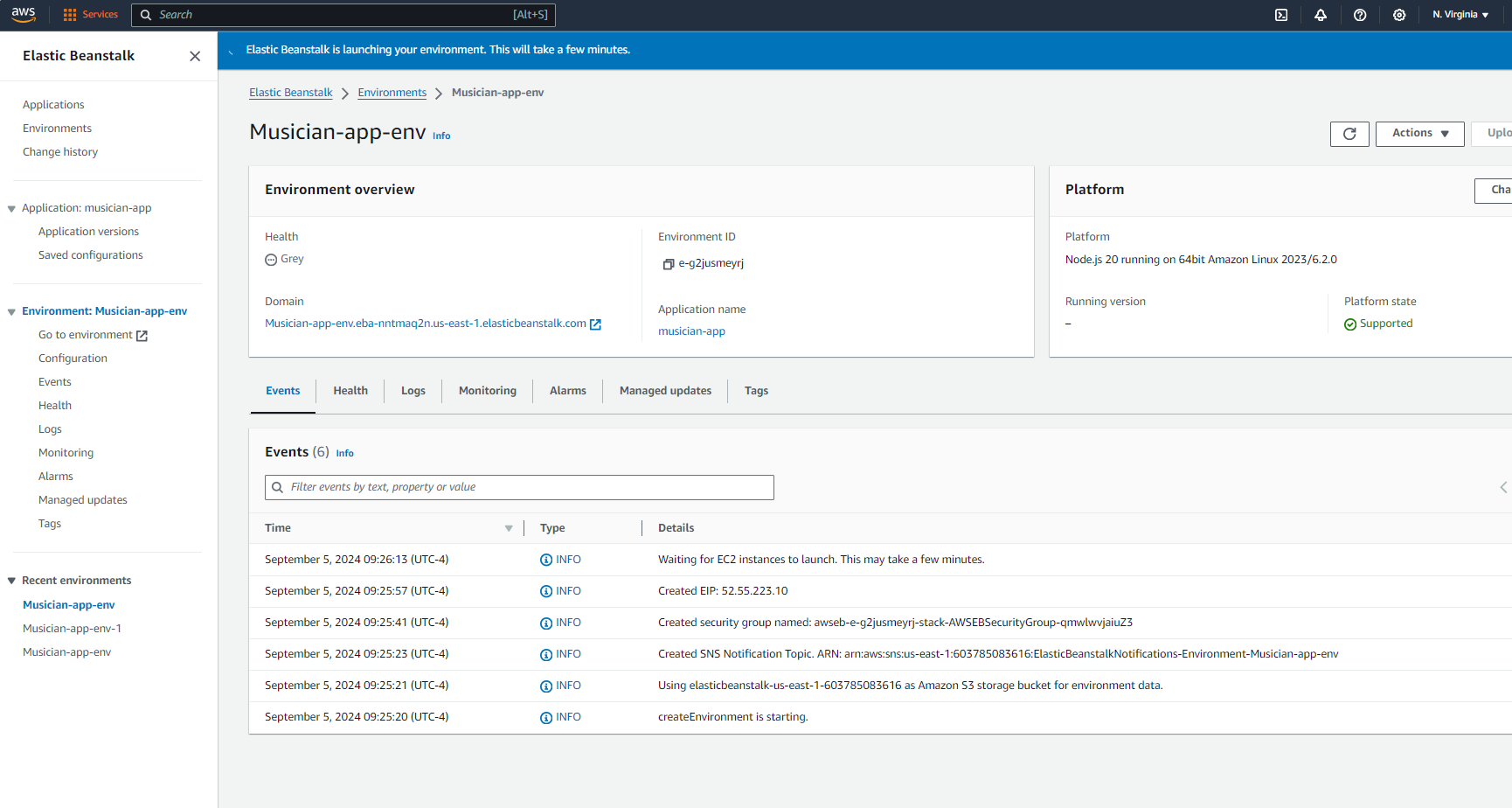


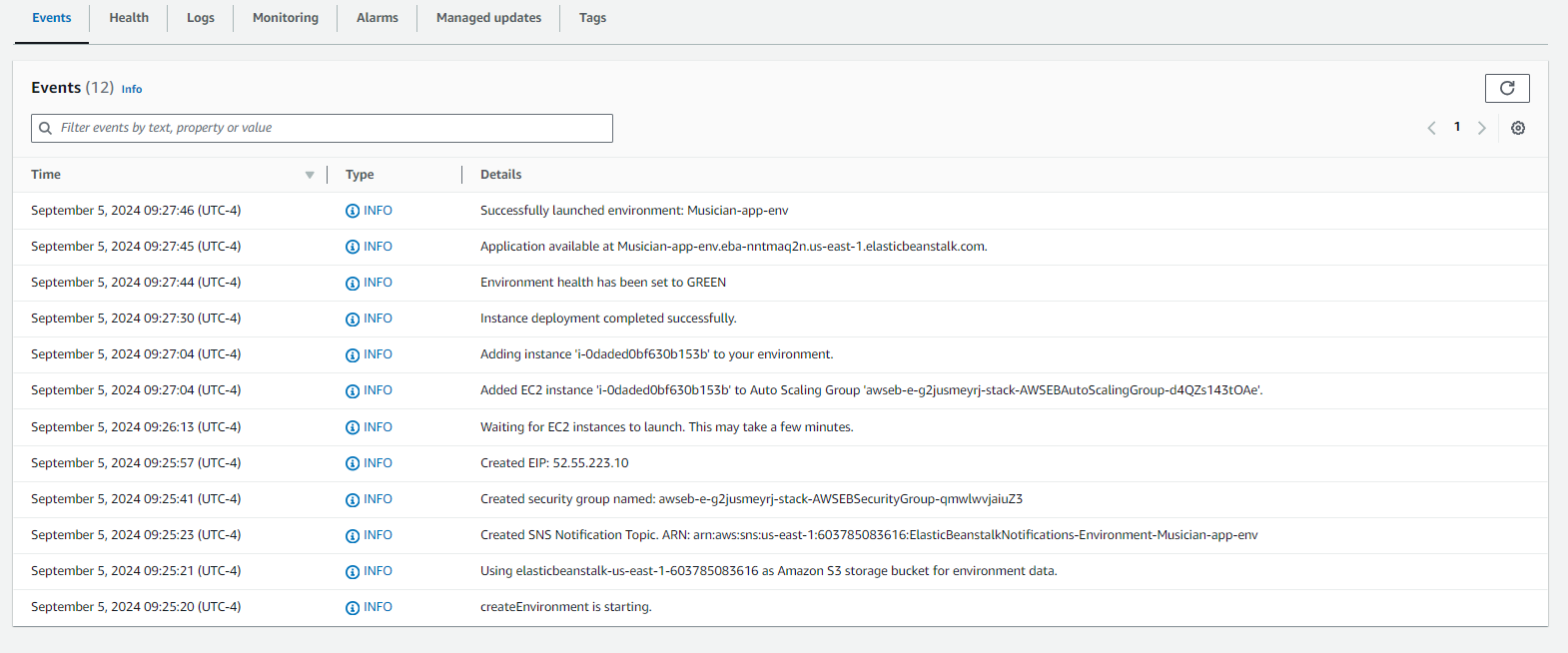
**Configure rolling updates and platform software**

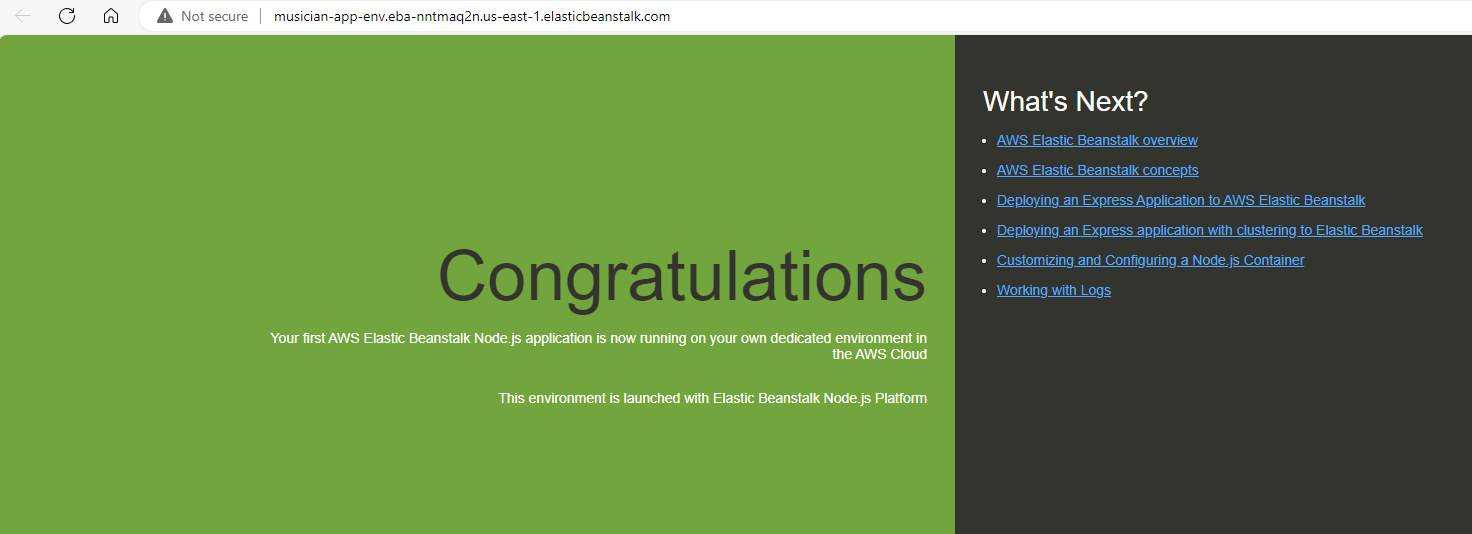


**Review and submit to create.**

**Elastic Beanstalk will launch and create the environment for the application.**



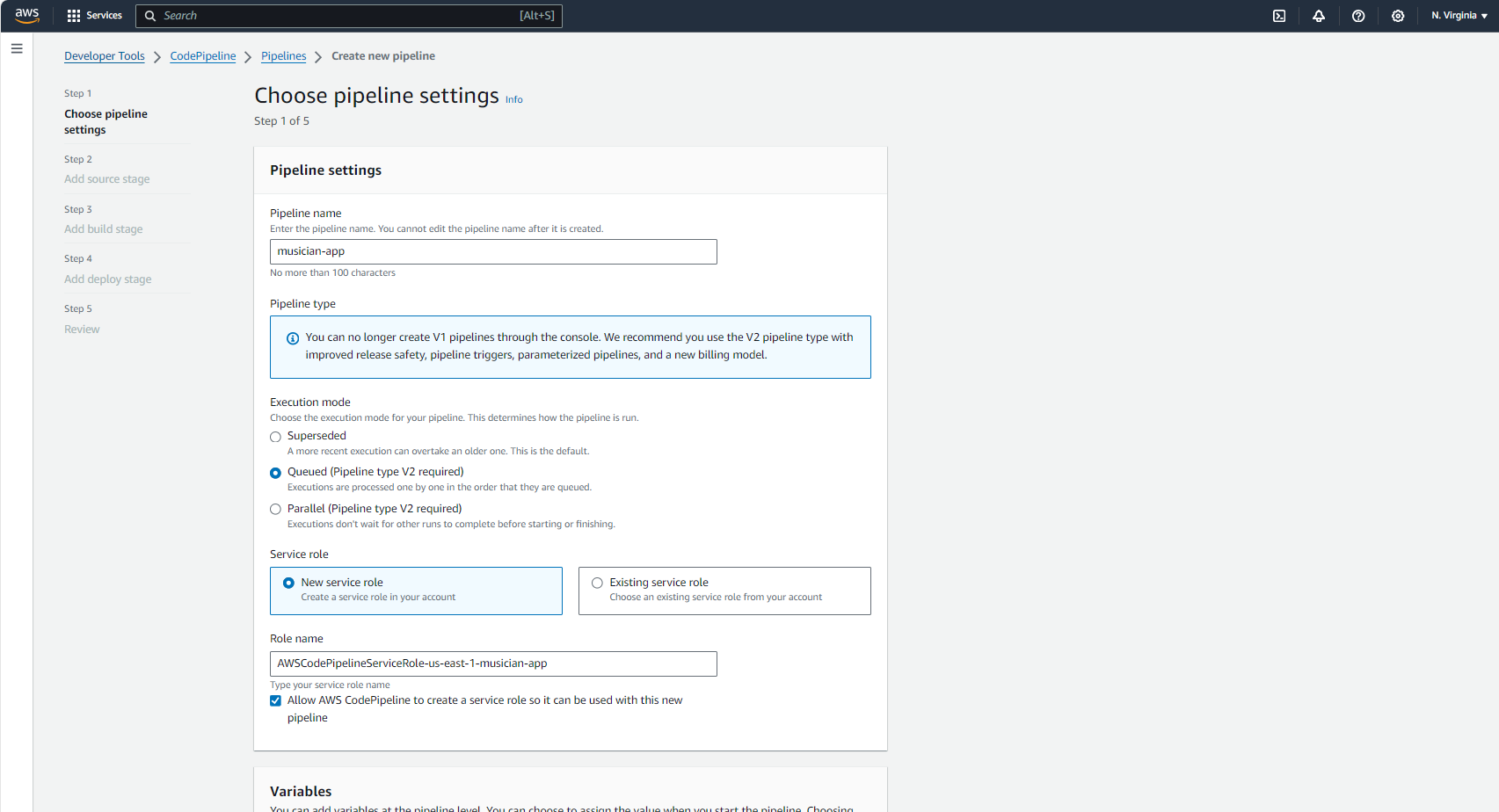




Navigate to Code Pipeline and create a new Pipeline.

Select create a pipeline and configure the pipeline.

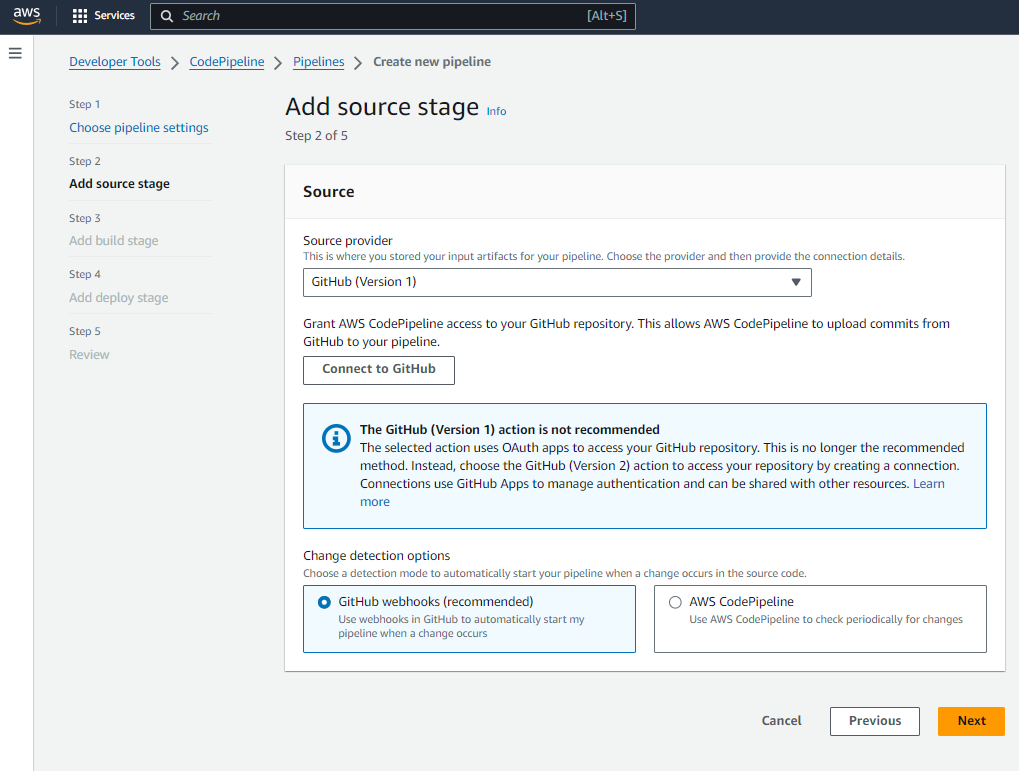
This will Create a new IAM role.



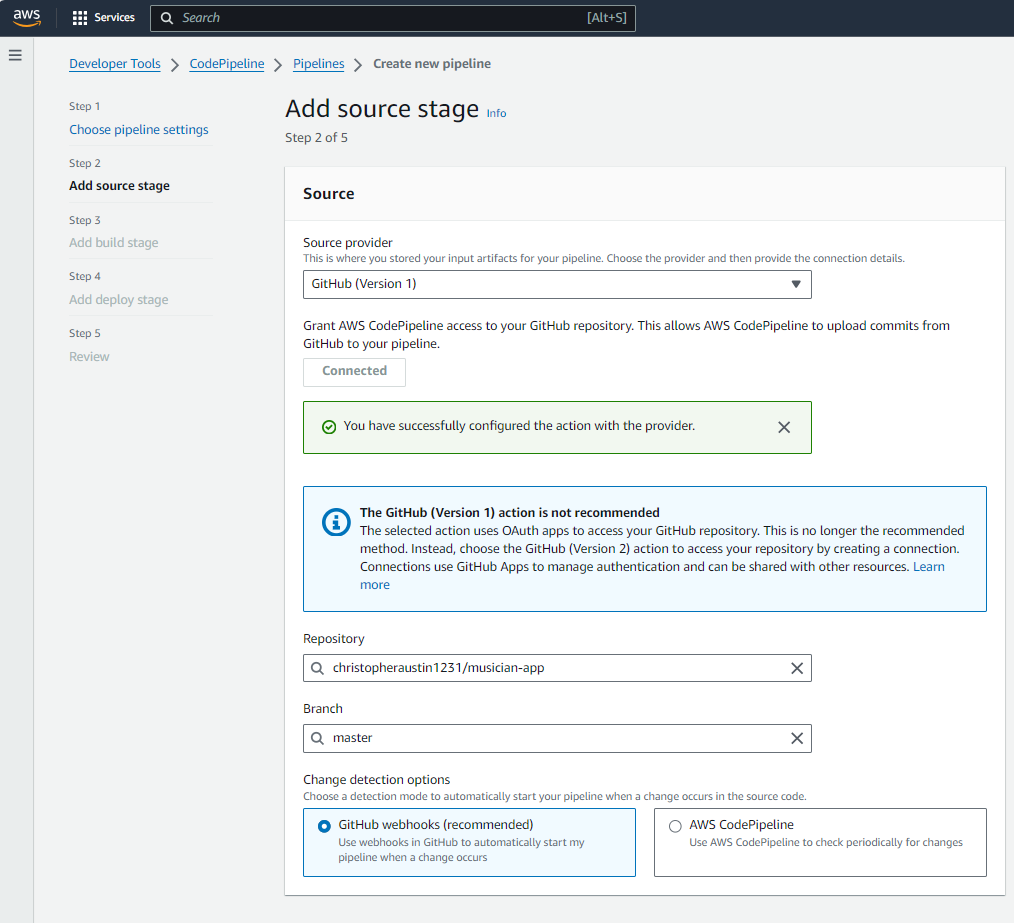
In the advanced settings you can leave the default location and use an AWS managed Key for this project.

Connect your Source Stage to your pipeline (Github or Gitlab) ensure to use Github Webhooks as the detection option.

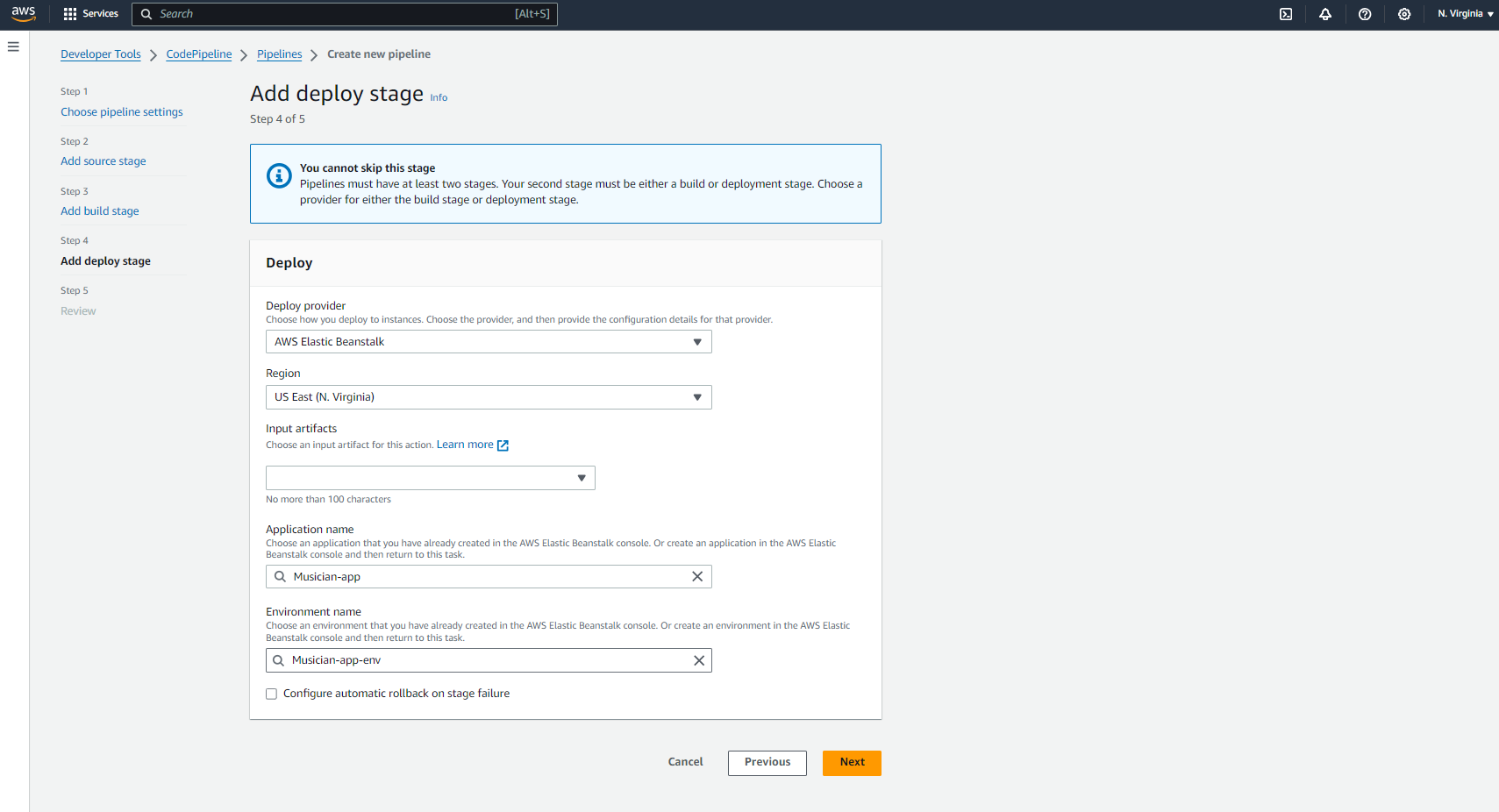
Utilize Github webhooks for the detection mechanism to kick off your pipeline. This is more efficient than having AWS Code pipeline check for changes.



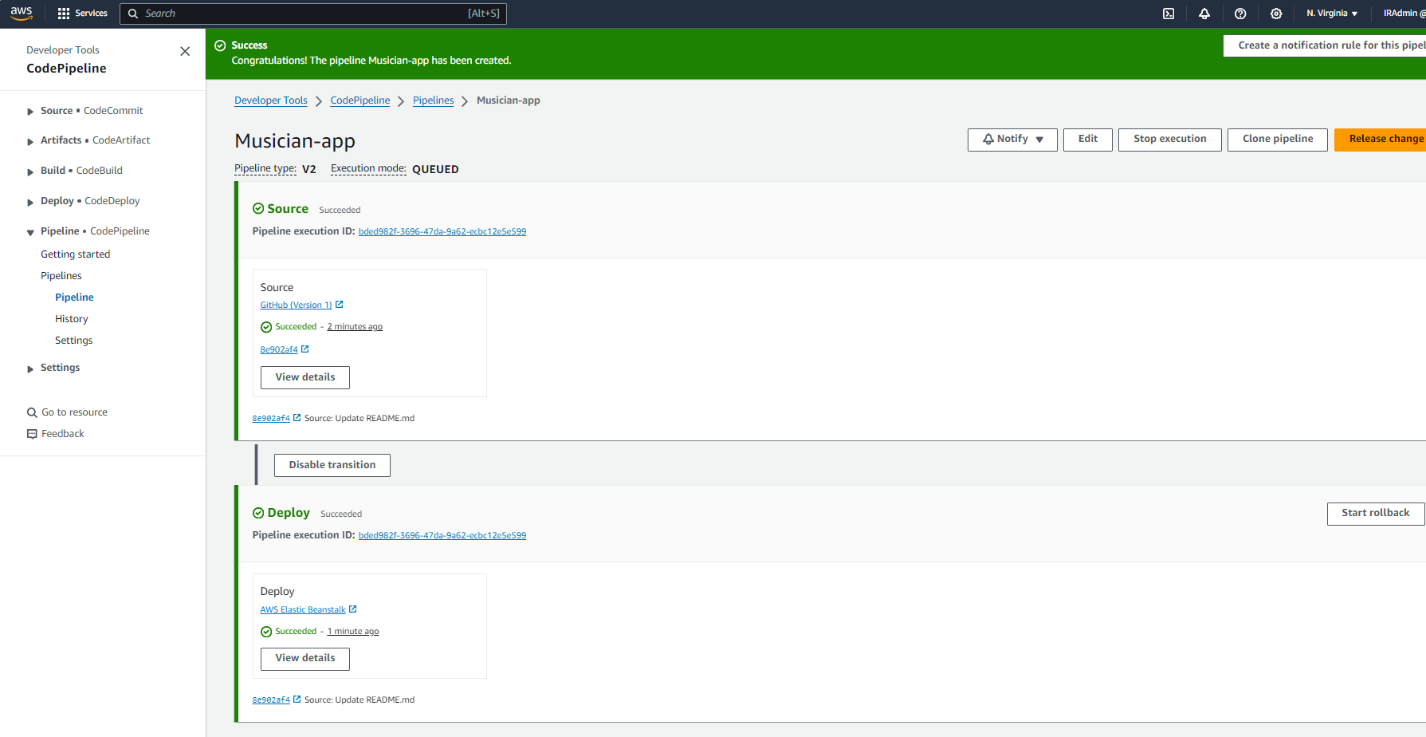
Continue out the build stages in Code pipeline.



Add a deploy stage and choose how you wish to deploy.



With the environment configured and the pipeline successfully built out you will see the pipeline success.



You have now successfully utilized the elastic beanstalk service to deploy out your own Node JS application and integrated with a CI/CD pipeline in AWS Code Pipeline.