AWS Elastic Beanstalk

End-toEnd Web application management

Cloud services ---> important services

IaaS ---> ex: EC2, S3, RDS, VPC

PaaS ---> Platform as a Service ex: Elastic Beanstalk

SaaS ---> Software as a Service (Gmail, Zoom, Teams)

Web Application ---> Deployment

We need the following:

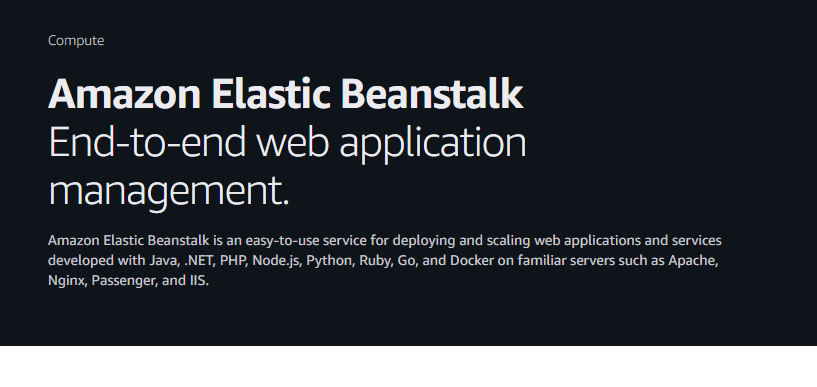
1. Create Network VPC
2. Create Security Groups
3. Enable Inbound Rules
4. Create EC2 instances
5. Install required Software/dependencies on EC2 to run our appn code --> Ex: Java, Tomcat, IIS
6. Create Load balancer
7. Setup Autoscaling groups for high availability
8. Deploy our application

If you use PaaS, you need not do the above steps manually, Elastic Beanstalk will take care of it

You only need to take care of Application code, other steps will be taken care by AWS itself

Elastic Beanstalk: it provides Platform as a Service ---> AWS will provide read-made platform to run our application

Note: Whenever we go with AWS Elastic Beanstalk, then the first 7 steps of given list will be taken care by Elastic Beanstalk and we will be taking care only Application deployment not everything. No specific or fixed charges/price for Elastic Beanstalk, it depends on user-case by use-case. Many resources are available like EC2 instances, Load balancers, AutoScaling Group etc

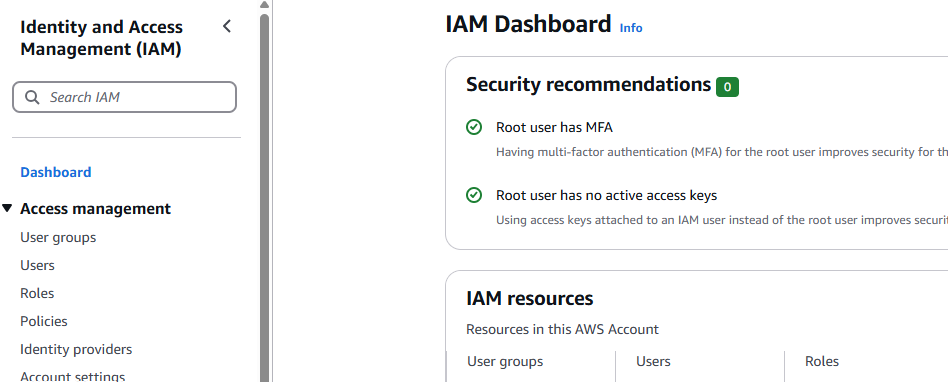


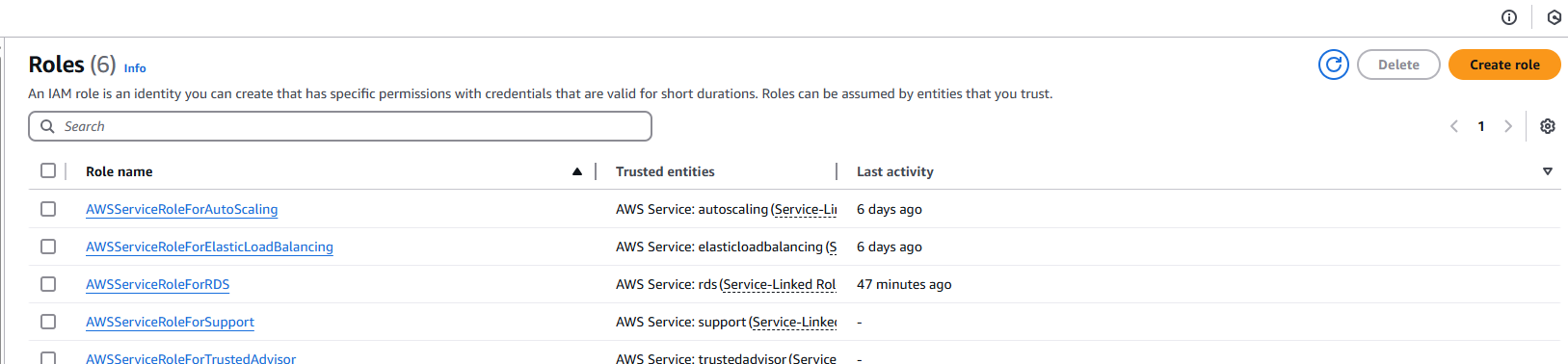
Pricing

There’s no additional charge for Elastic Beanstalk. You pay for Amazon Web Services resources that we create to store and run your web application, like Amazon S3 buckets and Amazon EC2 instances.

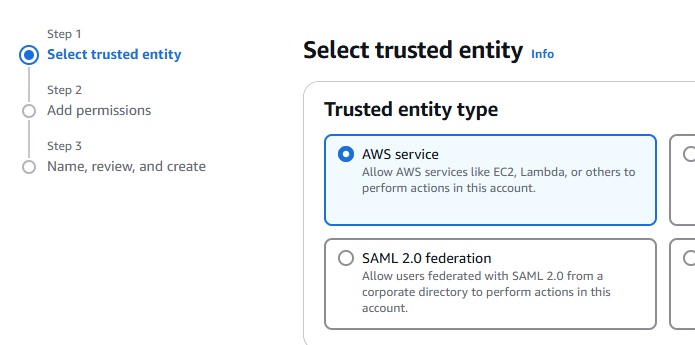
If you want to use EBS, we need to create an IAM role. To access one service from another service, we need IAM role

Step 1 is IAM



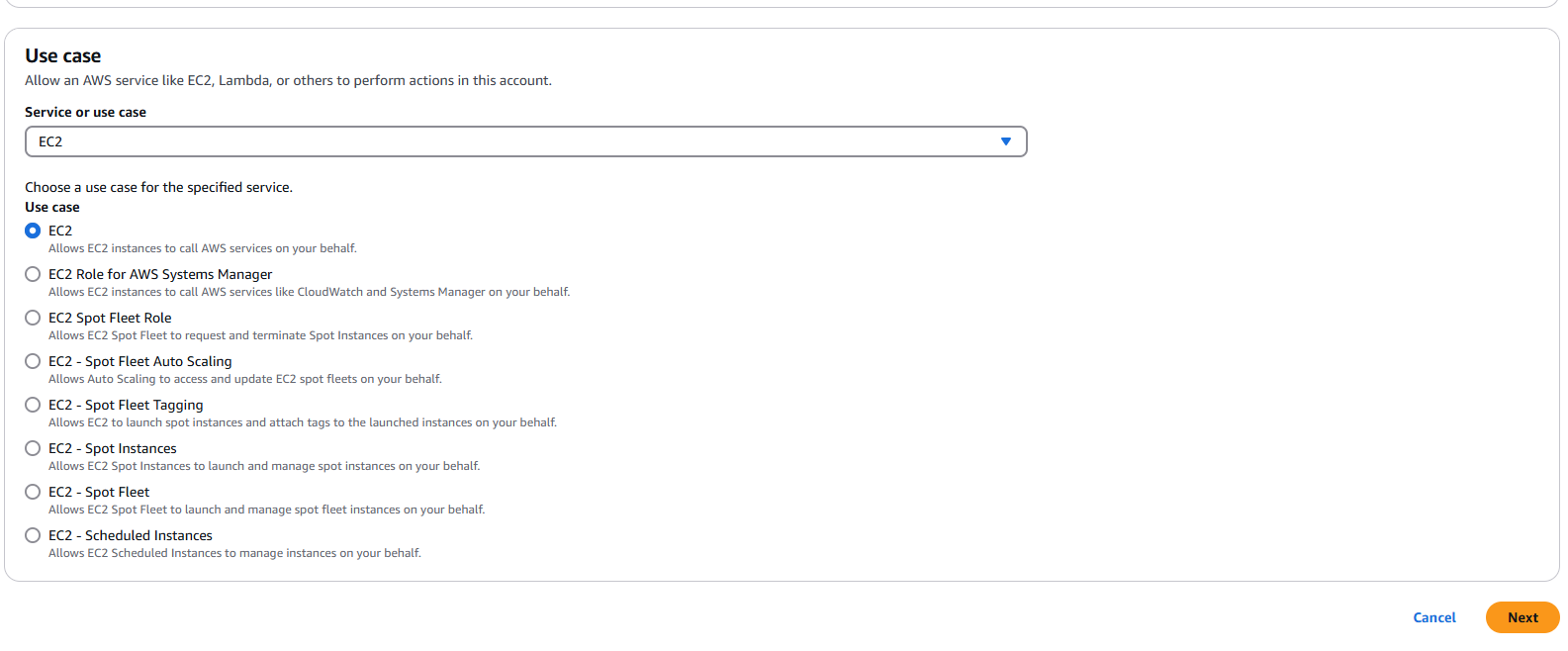


Create Role



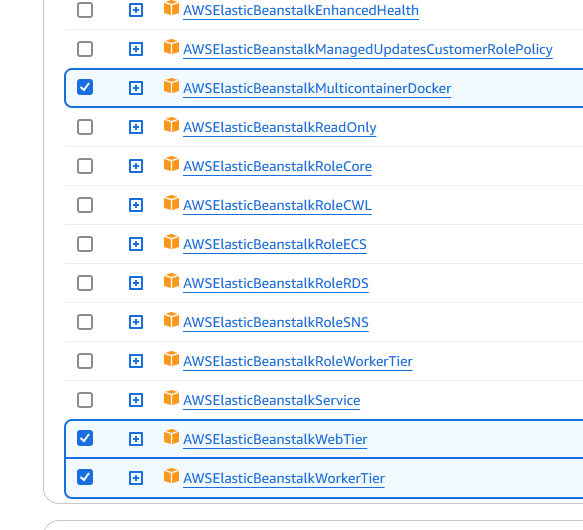
Select AWS service

EC2, Click Next

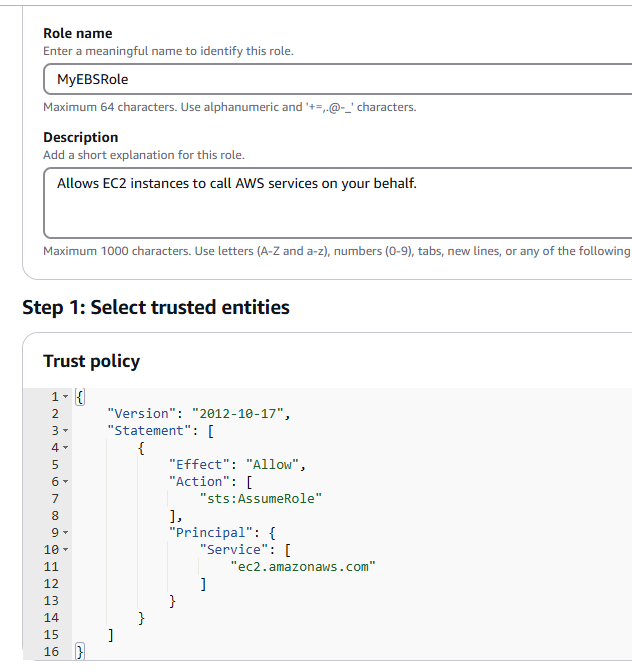


Note: this EC2 requires 3 minimum permissions

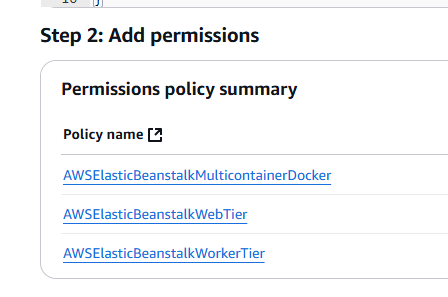
1. AWSElasticBeanstalkMulticontainerDocker
2. AWSElasticBeanstalkWebTier
3. AWSElasticBeanstalkWorkerTier

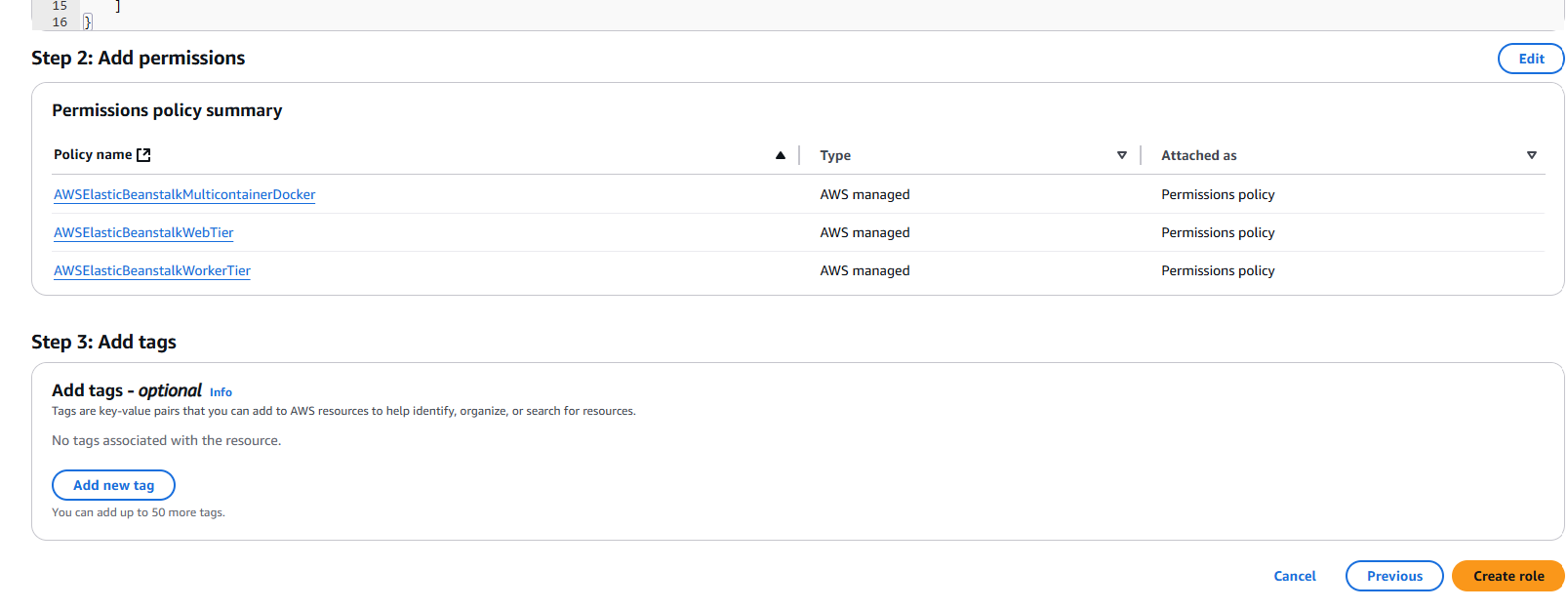


Click Next

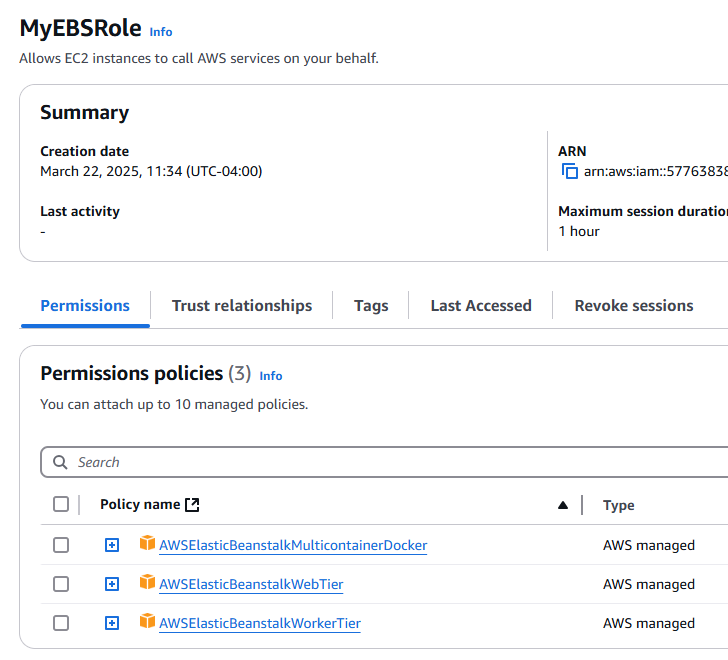


Double check the 3 policies discussed above





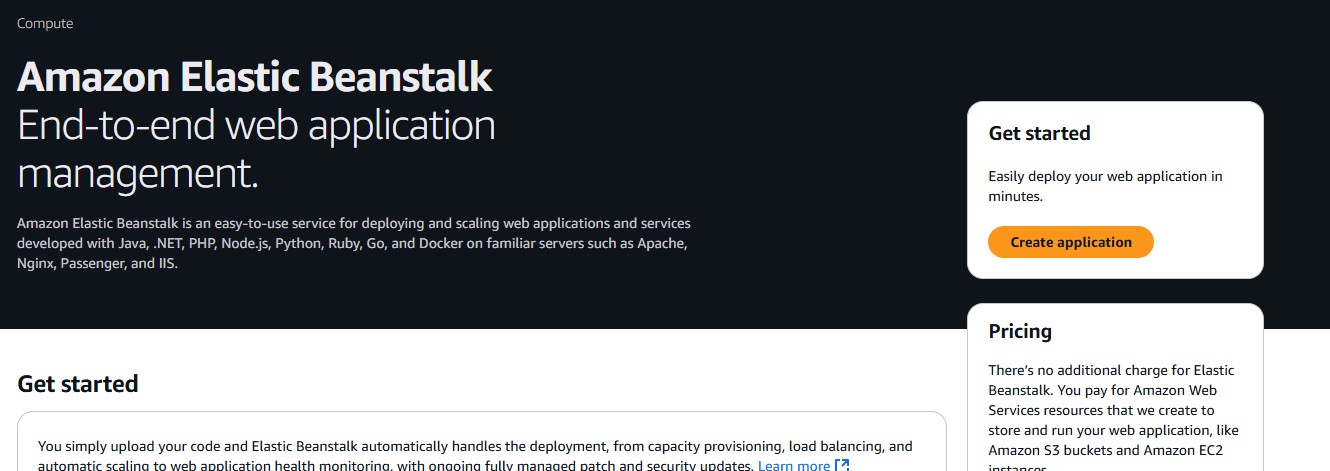
Click Create Role



Lab/Practical Task on Elastic Beanstalk

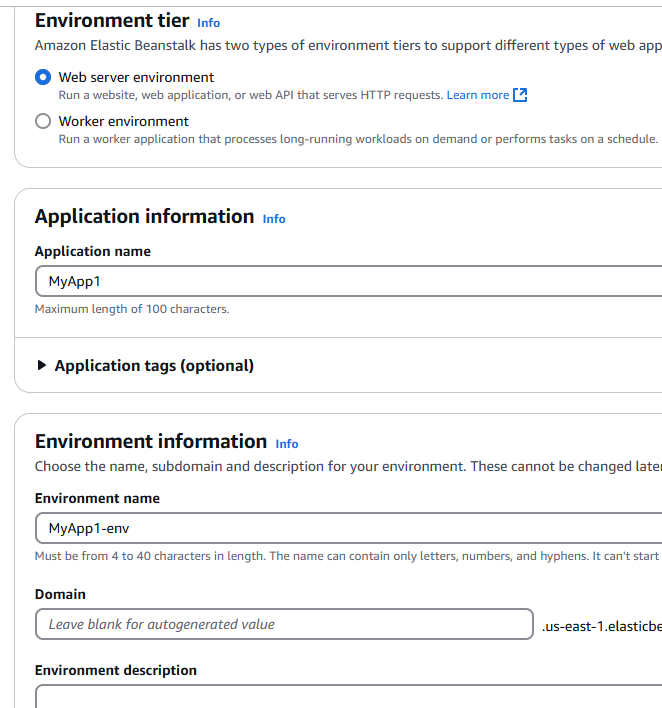
1. Create an IAM Role with following policies:
   1. AWSElasticBeanstalkMulticontainerDocker
   2. AWSElasticBeanstalkWebTier
   3. AWSElasticBeanstalkWorkerTier
2. Create Application using Elastic Beanstalk
3. Create Environment for the application by choosing required Runtime
4. After the environment is created, it will generate DNS to access application

Go back to EBS page

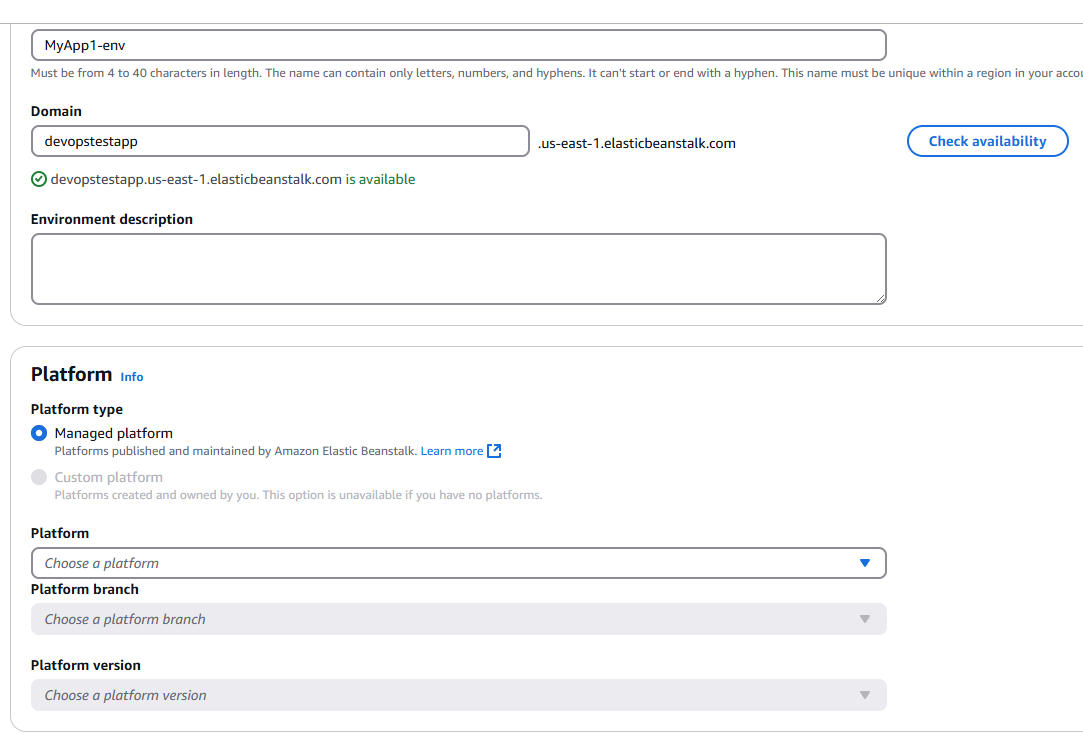


Create application

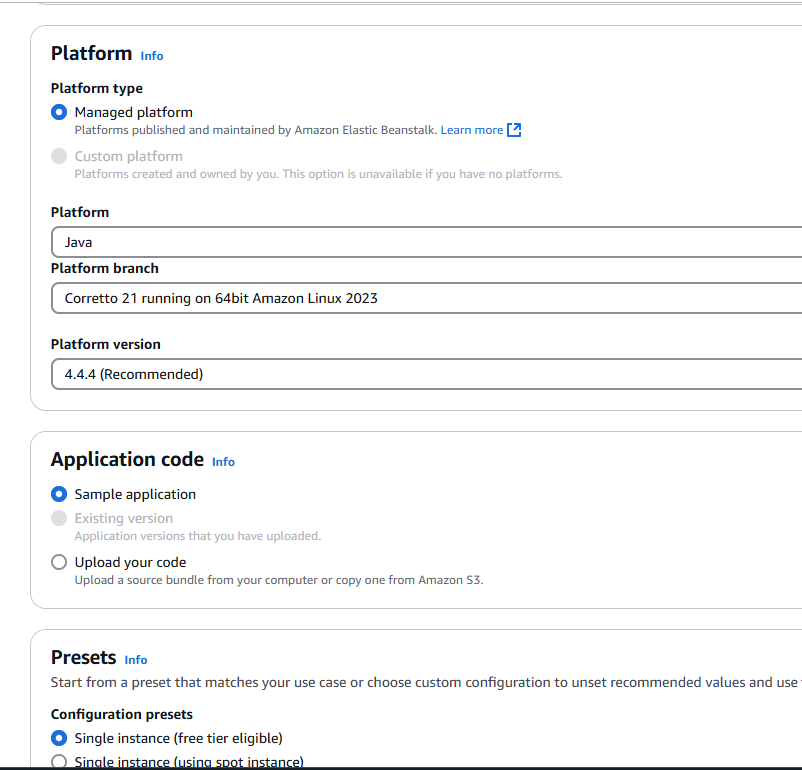
Click Web server environment

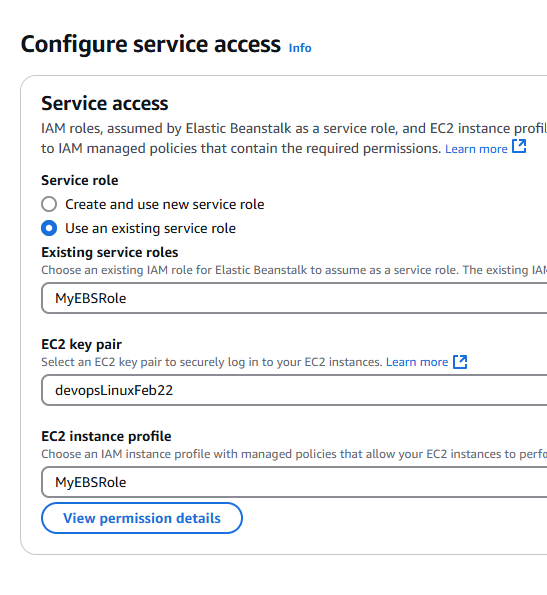


Enter Domain: devopstestapp

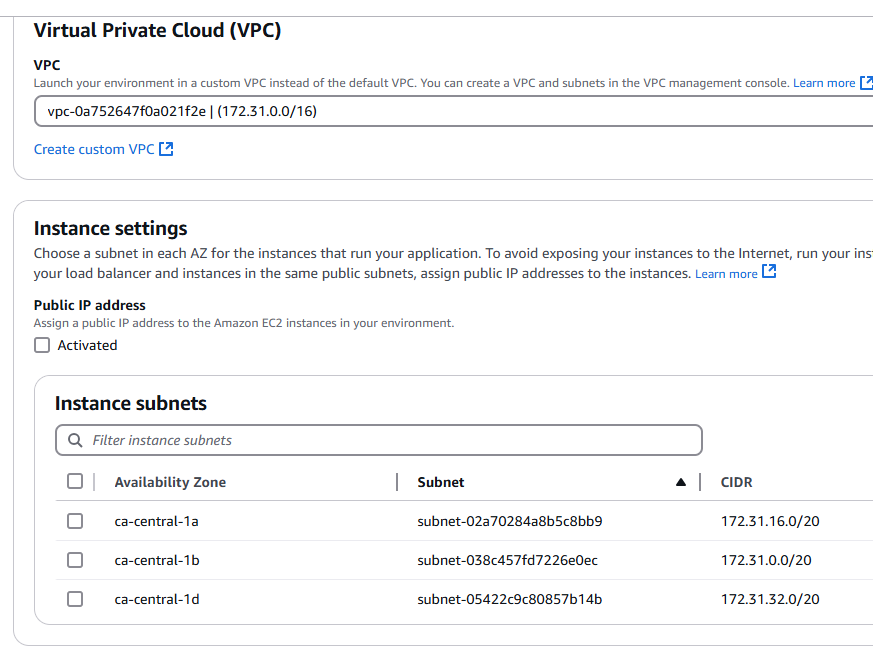


Select Platform Java for now

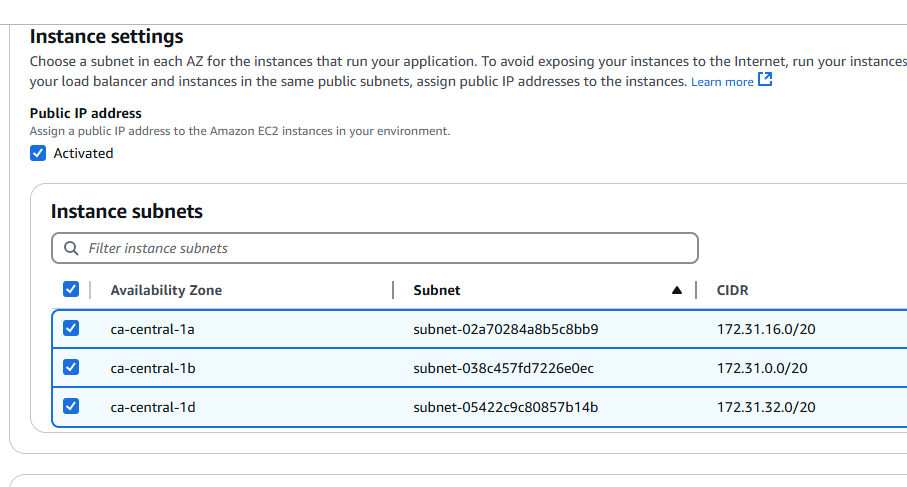




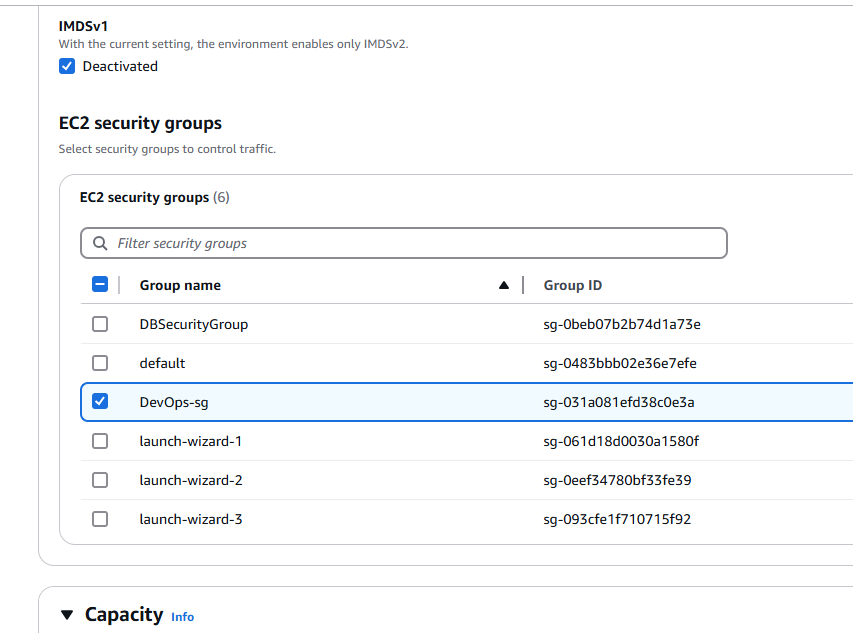
Select default options only



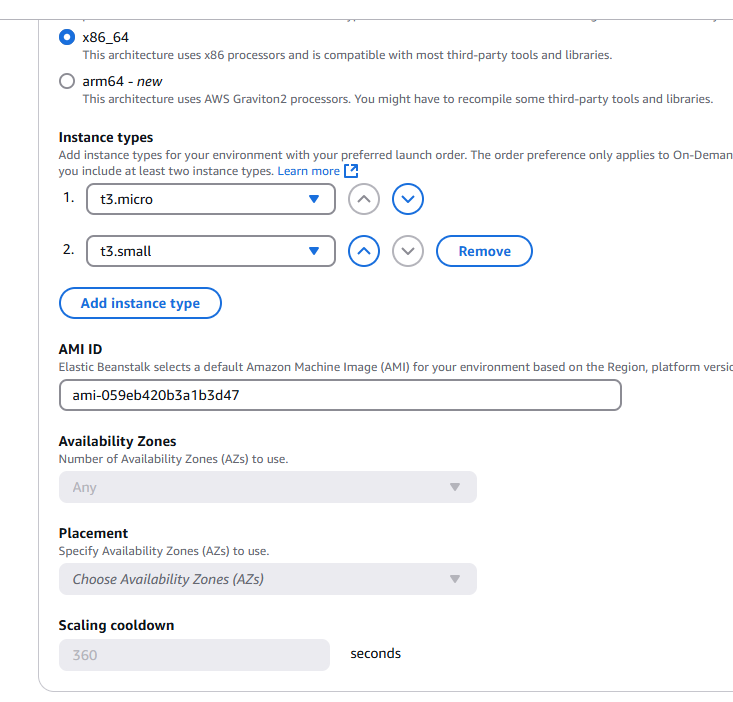
Select Activated



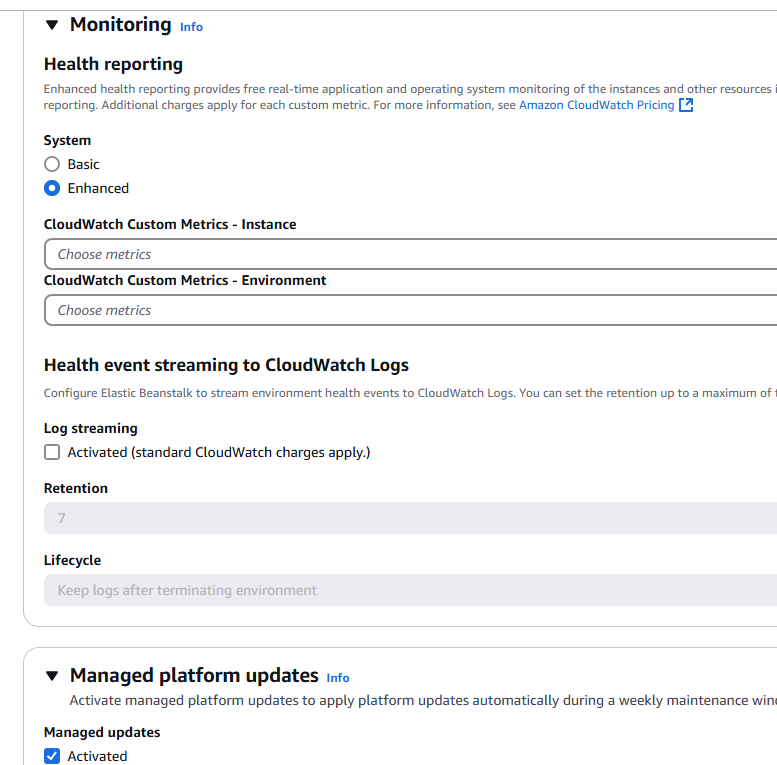
Select Security group we already created

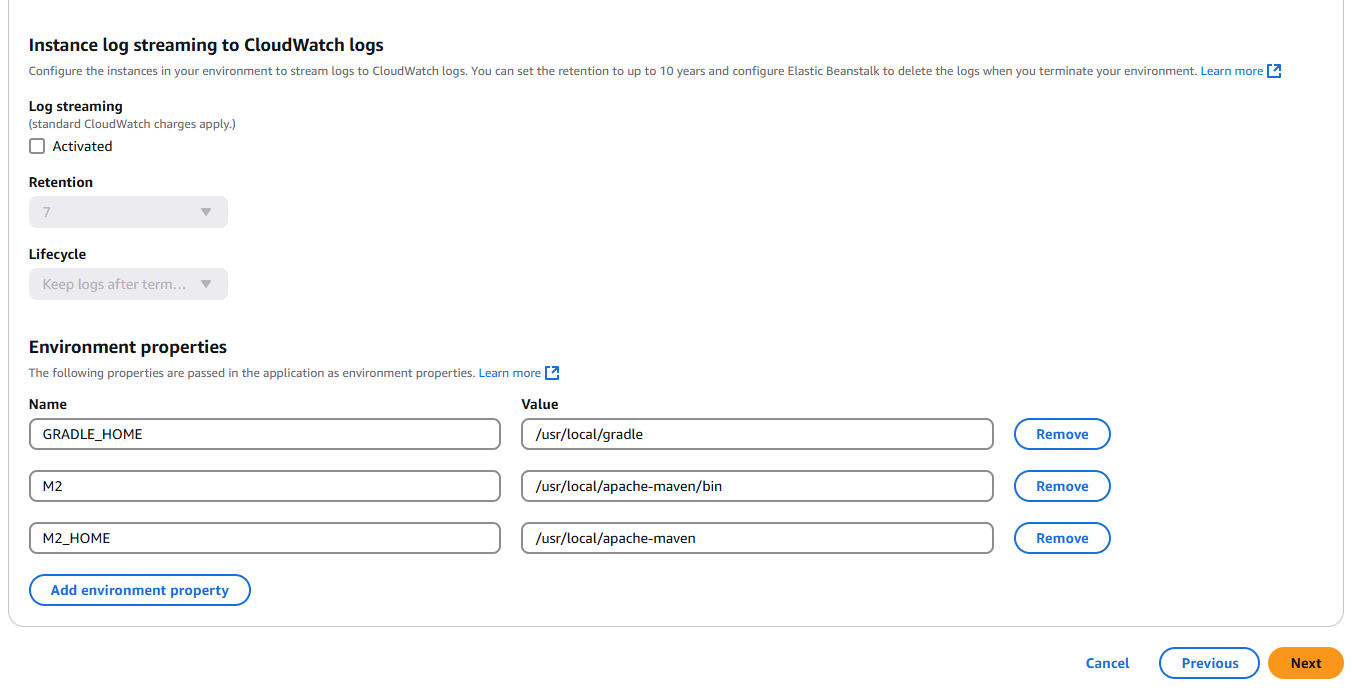


Keep these things default

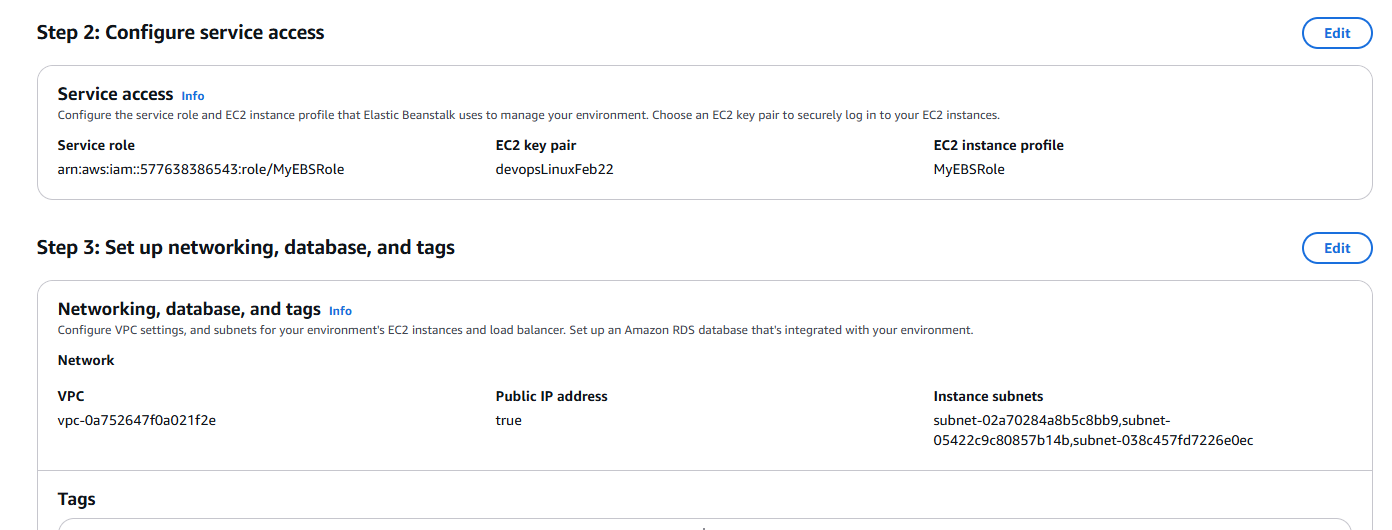


All default -> click Next

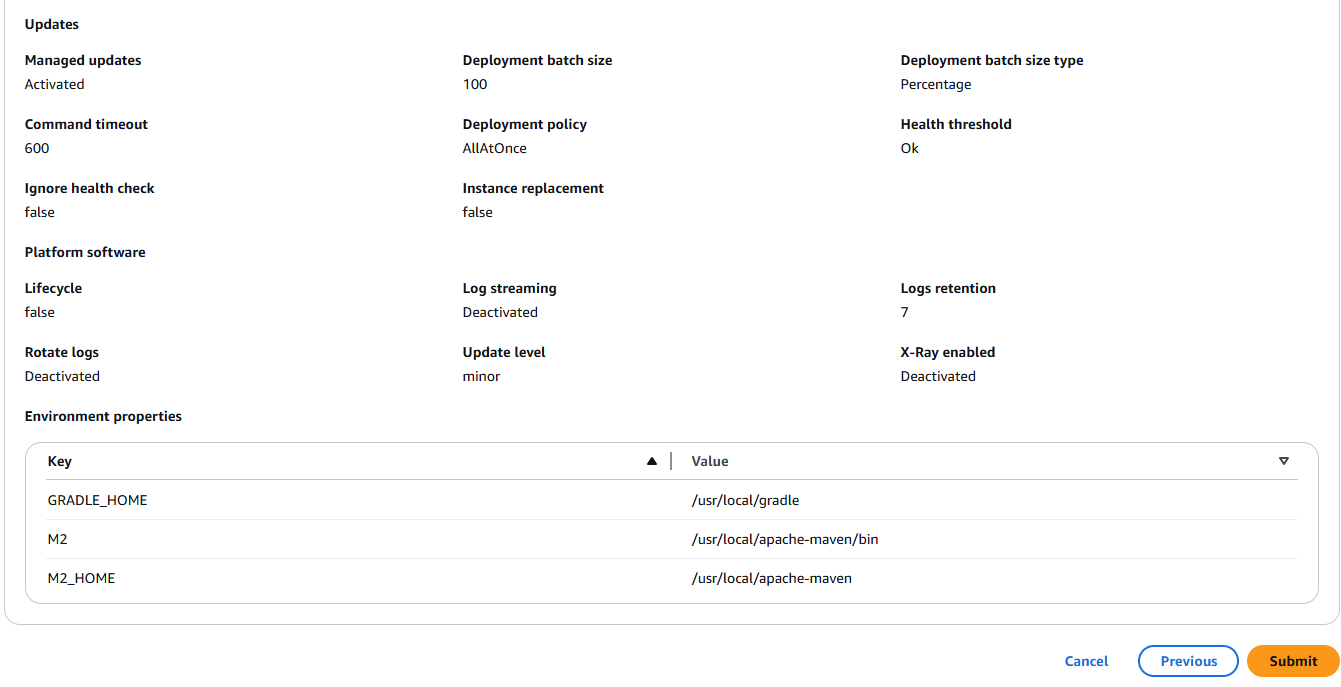




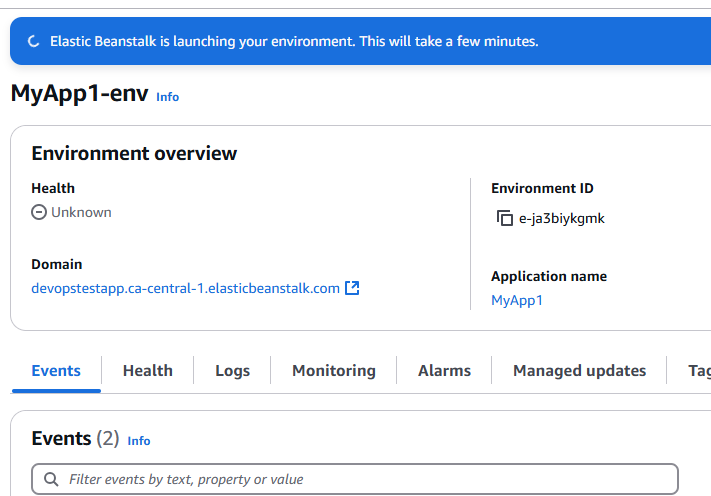
Review our configuration



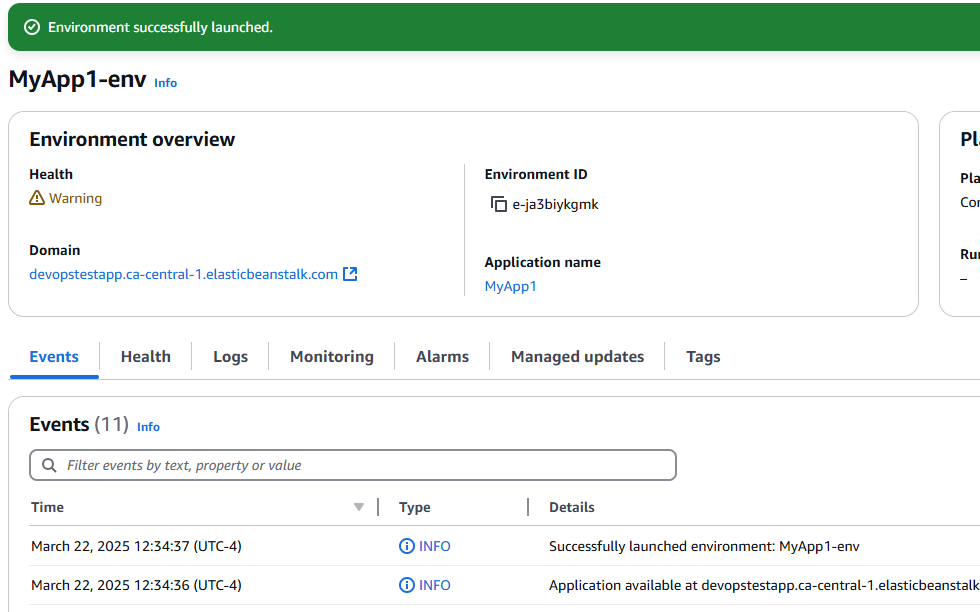
Click Submit



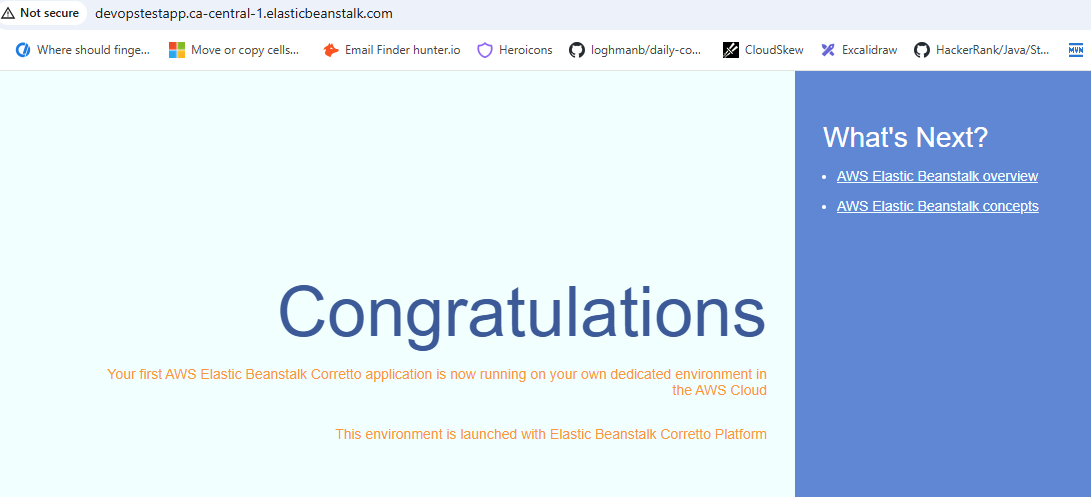
My application will be available on this domain



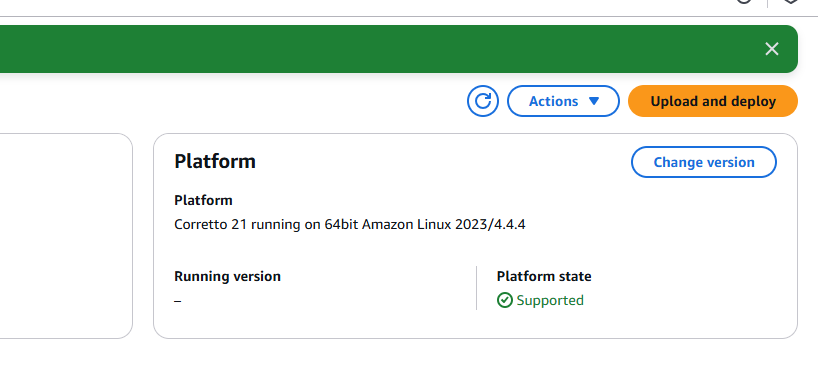
Successfully launched

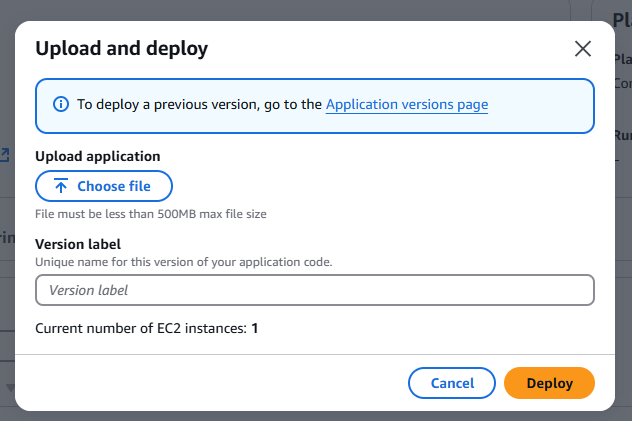


If you click on the link



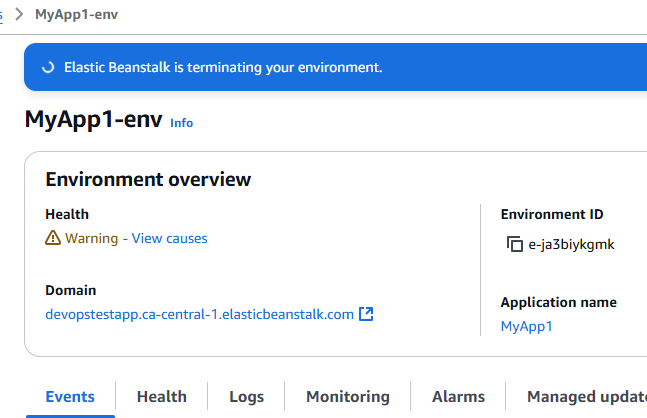
This is the Sample application, we can deploy our own application also ---> Click on Upload and deploy



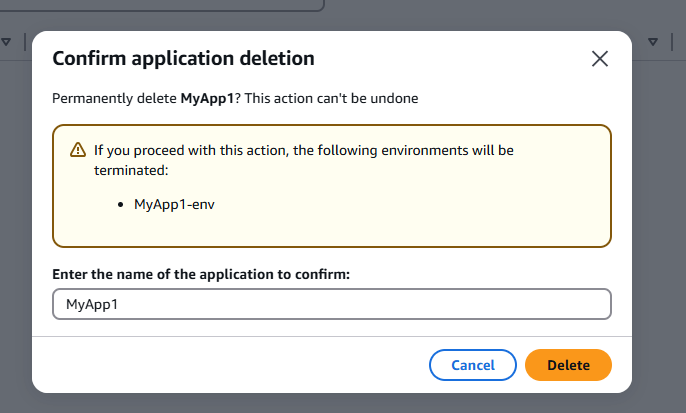


You can choose JAR or WAR file and upload to deploy your own application

Then Terminate the environment



Go back to Application



Elastic Beanstalk

PaaS ---> inside platform it will manage other resources and it is Pay-as-you-Go model

Inside platform, it will manage

1. EC2 instances
2. S3 buckets
3. LBR
4. Elastic IPs

EC2 VM ---> Hourly billing

EC2

App deployed

There is another concept ---> Pay as you use (you have deployed the application, but noone is making a request) . say if application is not getting executed, it will not be charged

If code is executed only then bill should be generated, till then bill will not be generated

If code I executed for 10 min, then bill should be generated for only 10 minutes

If you want this then we need ====> **Serverless computing comes into picture**

AWS Lambdas