EC2 everything is Pay as you go

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

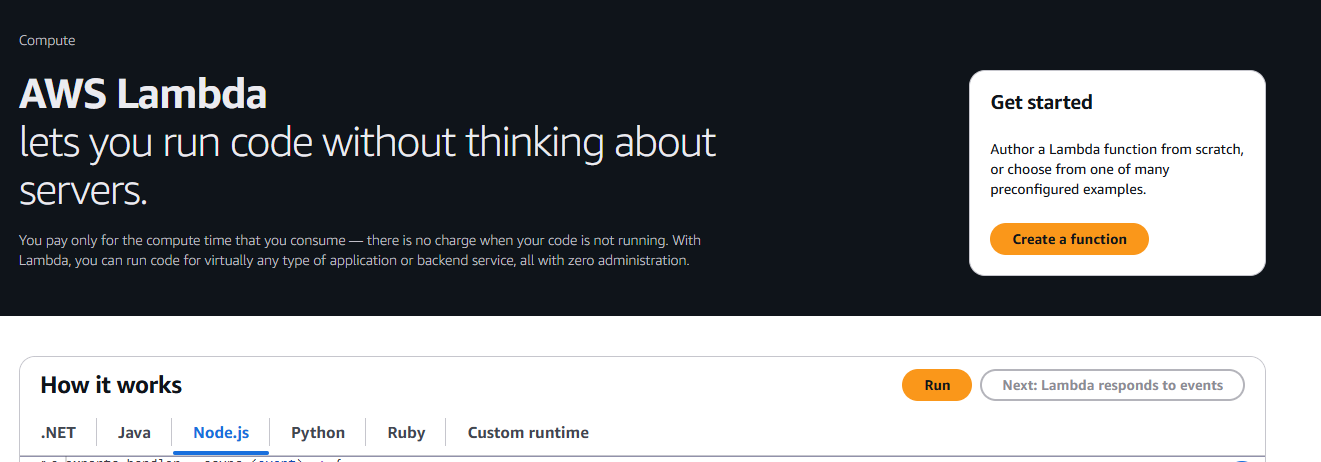
If code I executed for 10 min, then bill should be generated for only 10 minutes

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

Serverless means how much my application is getting used, only for that much I will be charged. For idle time, I wont be charged at all.

Run your application without thinking about servers. EC2 works like servers

AWS Lambdas



You pay only for the compute time that you consume — there is no charge when your code is not running.

AWS Lambdas are used to implement Serverless computing

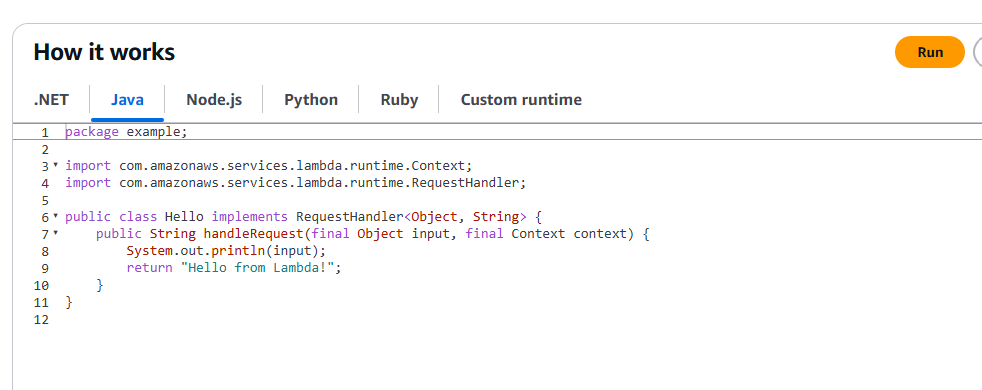
It is a way to run our application or code without creating / managing /paying for servers

If Application has 100 functionalities ---> we create 100 Lambda functions

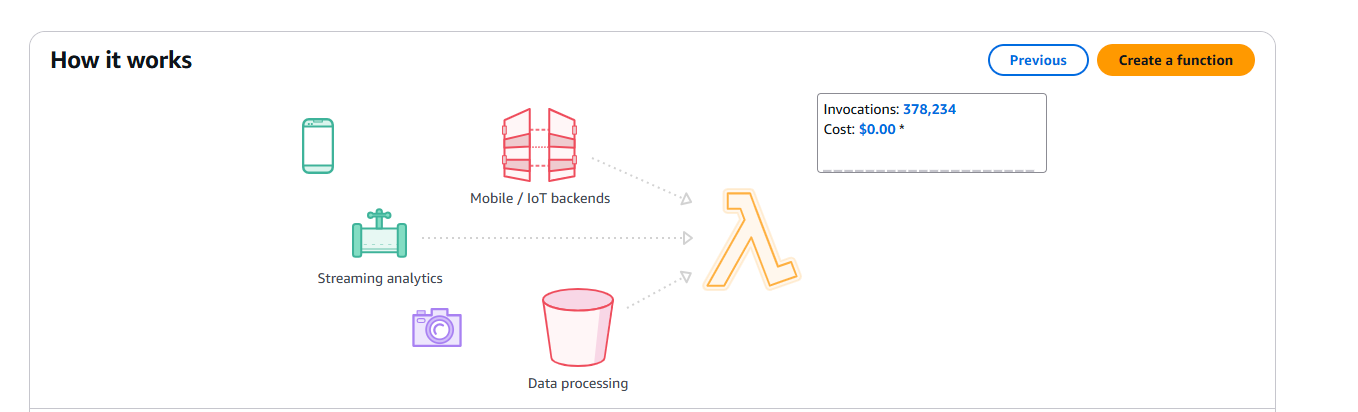
For bigger applications with hundreds of functionalities, would you go with AWS Lambda? The answer is No

Depends on what features or functionalities we have, we go with Lambdas

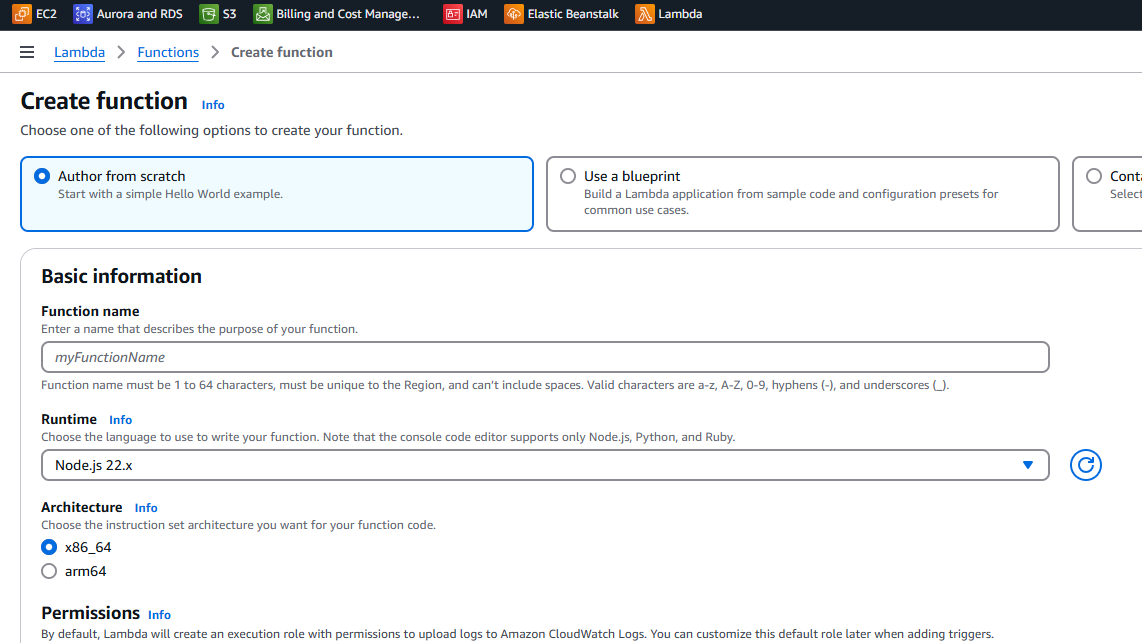
Click Run

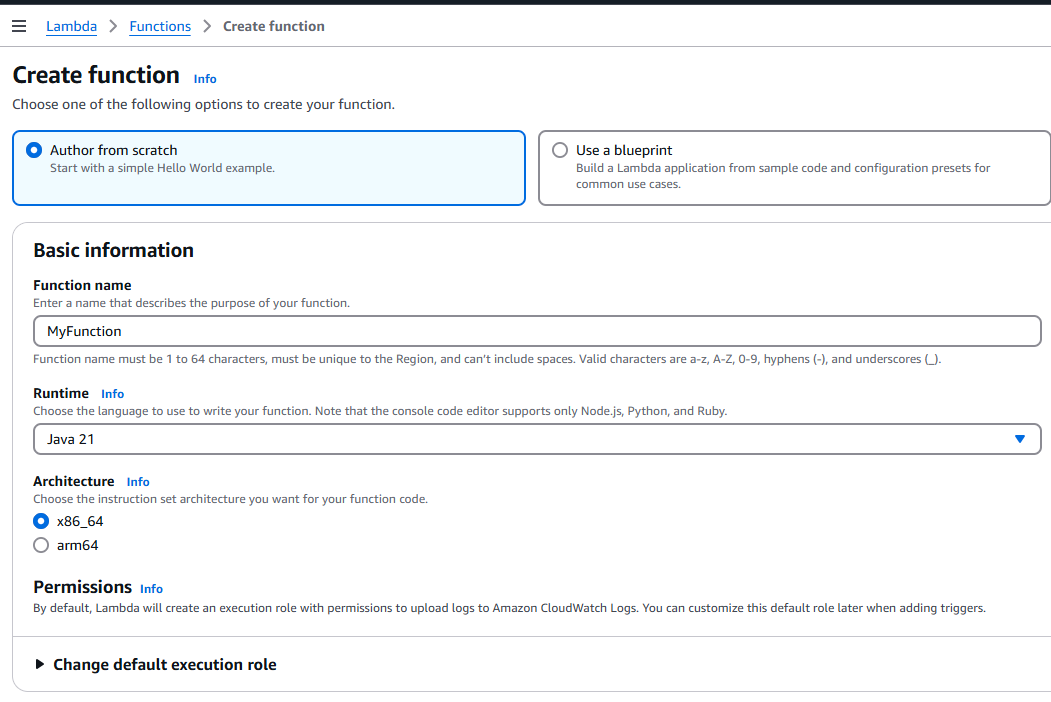


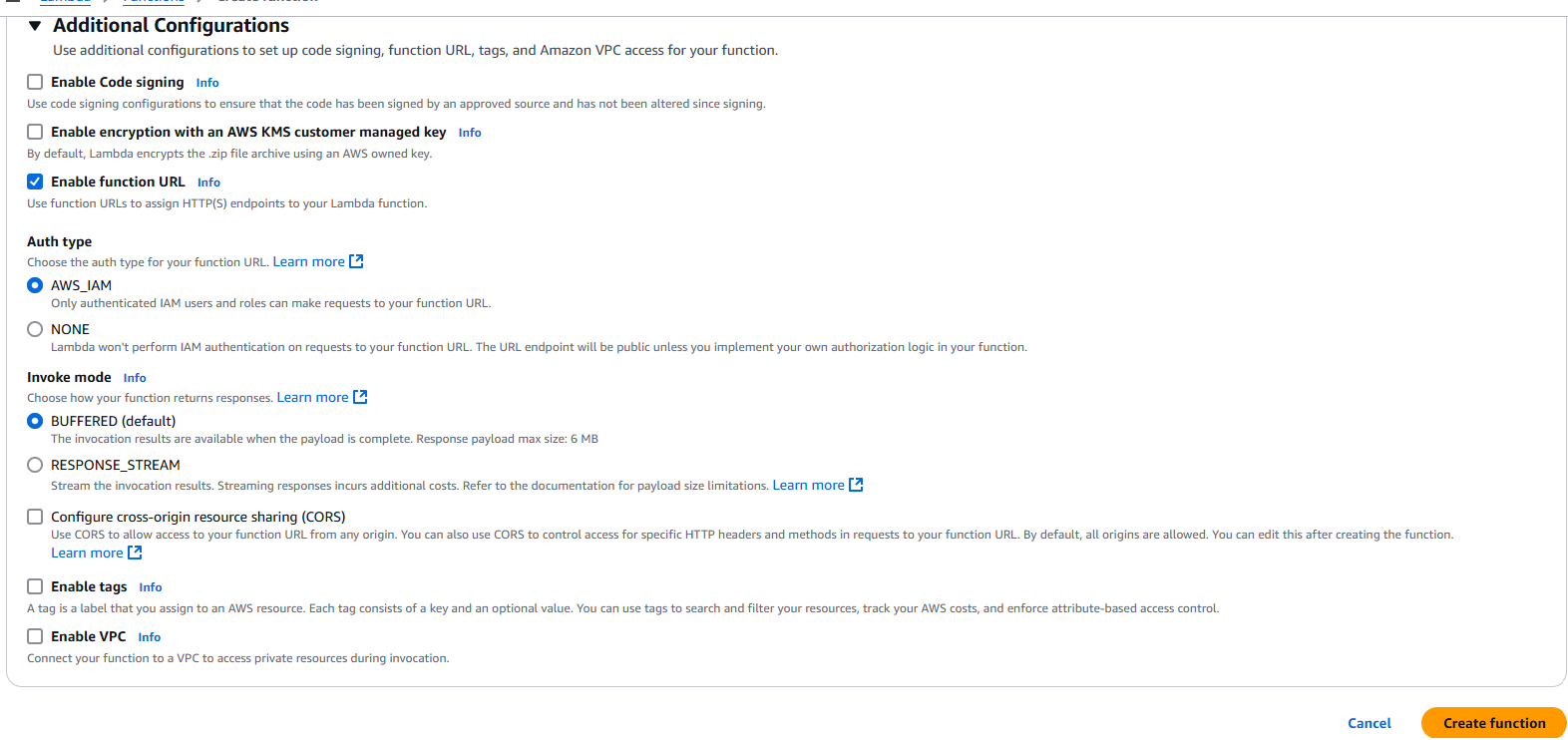
As we get requests, it charges accordingly



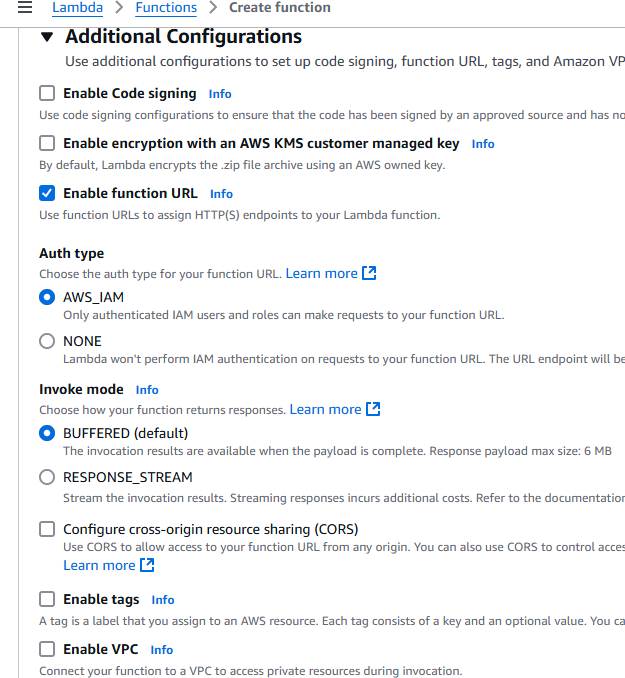
Create function

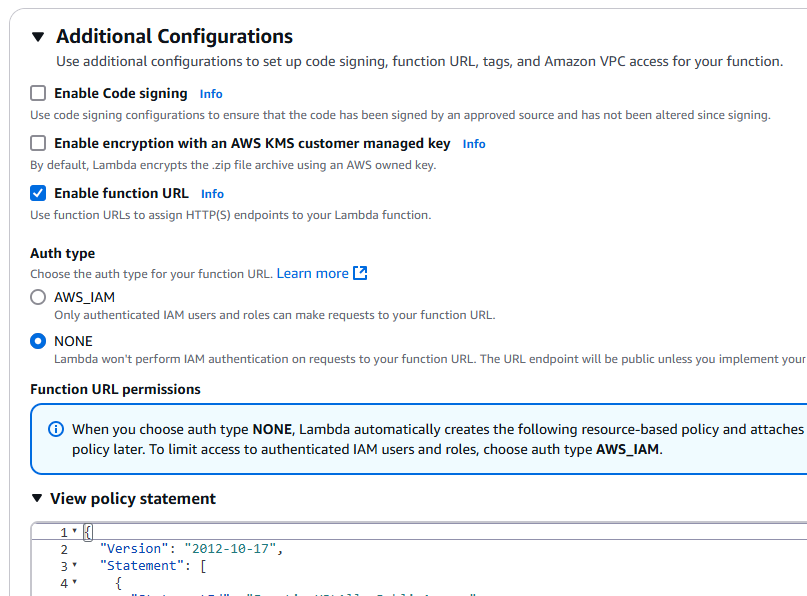






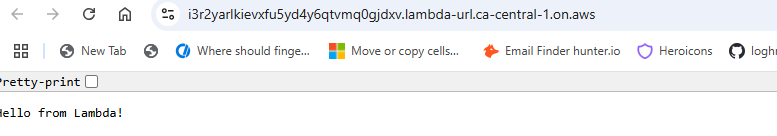
Click Enable function URL, AuthType is NONE

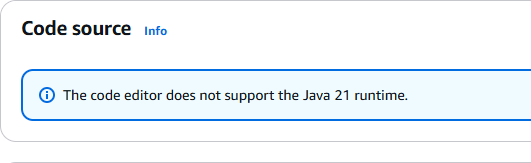




Create Function

Click on the URL





But we can upload our own functions

We can zip application and upload also

AWS Lambda is a way to run code without creating, managing or paying for servers

We pay for it for the time AWS runs and nothing more

AWS will scale our code depending on number of requests we receive. It is promoting event-driven architecture