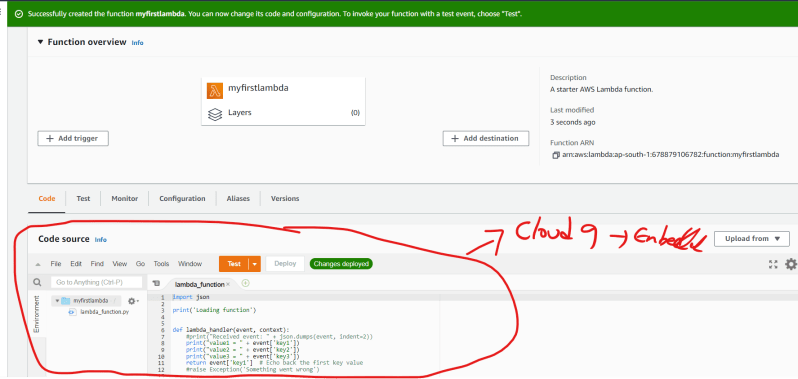
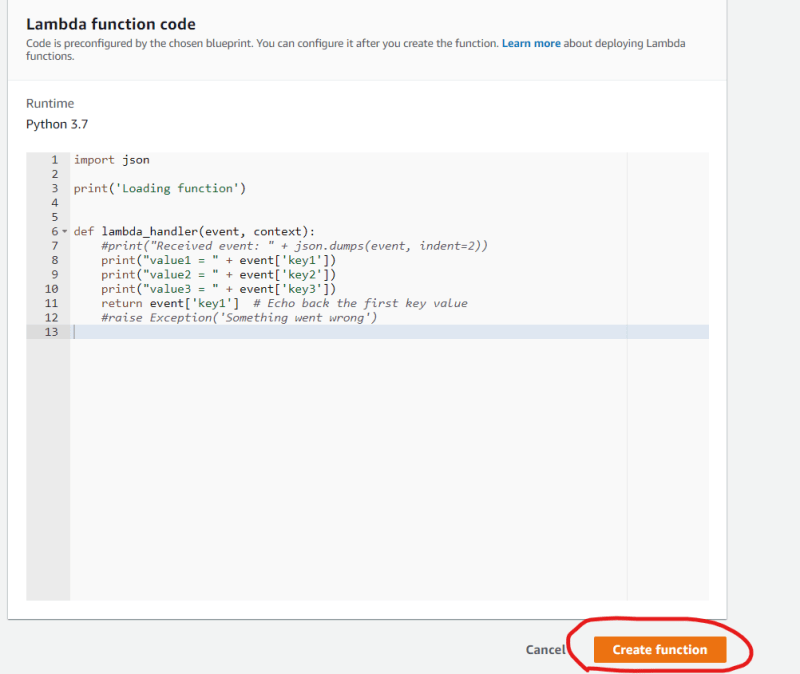
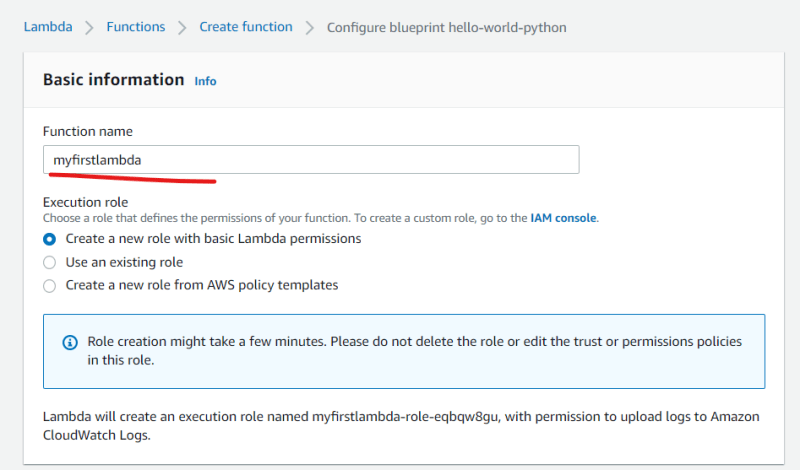
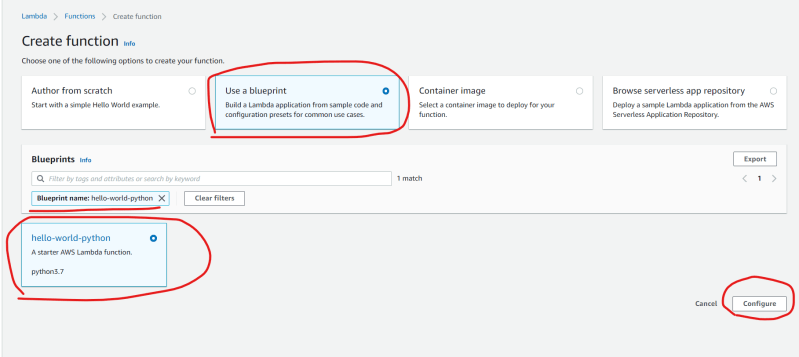
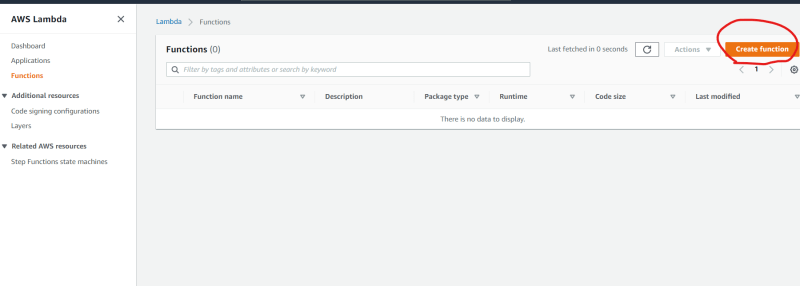
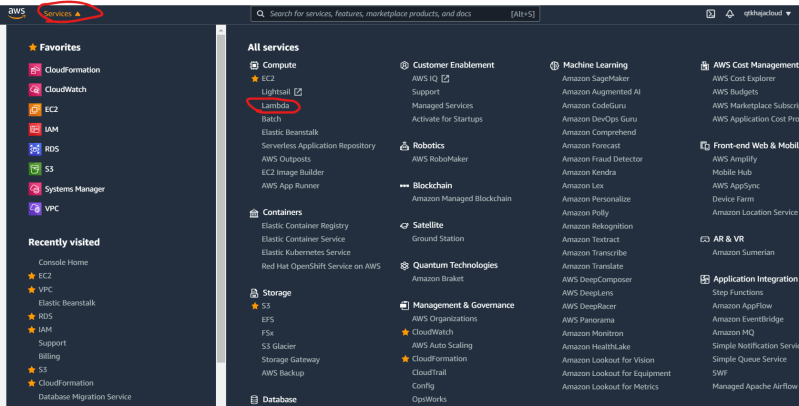
**AWS Lambda**

* AWS Lambda is a fully managed service from AWS that offers Functions as a Service (FaaS).
* Function is a piece of code that we want to execute to achieve functionality which is often triggered by an event.
* Function can be written in many different languages
  + Node js
  + Python
  + Ruby
  + Java
  + Go
  + .NET Core
* Developers are responsible for writing, testing and deploying the code to the AWS lambda
* When deploying aws lambda function we get options on how much memory should be allocated to the function, CPU share.
* The minimum amount of memory you can allocate is 128 MB, this goes all the way up 3008 MB
* Once we deploy the function, functions are invoked by something called as events
* Lets try to create our first AWS lambda function. 
* We can create lambda functions and then to trigger them we need to configure triggers

**Pricing model**

* With AWS Lambda we pay only for the amount of time the code is executing.
* Each AWS Lambda function is charged per request based on the combination of following
  + How long function executes (rounded to the nearest 100 ms)
  + The allocated memory that function was configured with.
* AWS calls this gigabyte-seconds or GB-sec.
* <https://aws.amazon.com/lambda/pricing/> for the offical pricing and examples from AWS
* With free tier account we can do 1 million invocations per account per mont (or the equivelent of 400000 GB-Secs) for free

**Lambda Foundational concepts**

* Functions: This is a resource to run our code in AWS Lambda
* Triggers: A trigger is a resource or configuration that invokes AWS Lambda function. Triggers can include AWS services that you can configure to invoke a function.
* Event: An Even is a JSON Formatted document that contains data for Lambda function to process.
* Execution Environment: An execution environment provides isolated runtime environment for lambda function.
* Deployment package: We can deploy lambda function function code using a deployment package. Lambda supports two types of deployment packages
  + A .zip file that contains the function code & its dependencies
  + Container image
* Runtime: The runtime provides language specific environment that runs in an execution environment

**Ephemeral state**

* What happens when the code finishes executing in Lambda?
  + The function runtime is terminated
  + This means access to the memory state and any temporary files that were created on the filesystem etc or in short everything is gon
* So if we need to maintain or keep tarck of state, then it should be saved external to the function such as AWS DynamoDB
* Functions have finited execution, Any function has a hardlimit of 15 minutes i.e when Lambda function execution takes more than 15 minutes it will be terminated

**Service Limits**

* For concurrency, there is a soft limit of 1000 executions at a time (This can be raised by creating support request)
* Size of environmental variables can’t be greater than 4 KB
* Size of invocation payload can be no greater that 6 MB for synchronous requests and 256 KB for asynchronous requests.

**Use Cases**

* AWS Lambda enables event-driven architectures
* Some of the popular use cases of lambda are
  + Backend-compute:
  + Data Processing
  + Chatbots
  + Alexa Skills
  + Operations and automation

**Execution Policies**

* Lambda functions executing on AWS might require to access other AWS services like ec2, s3 or others. For this to work we need to give permissions
* IN AWS giving a permission to an AWS service to access other AWS service is called as IAM Role.
* To create an execution role follow the steps as mentioned below
  + Navigate to IAM select roles and click on create role 