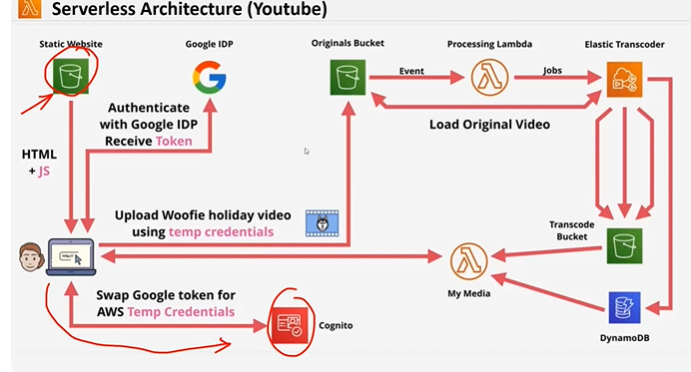


**Serverless architecture:**

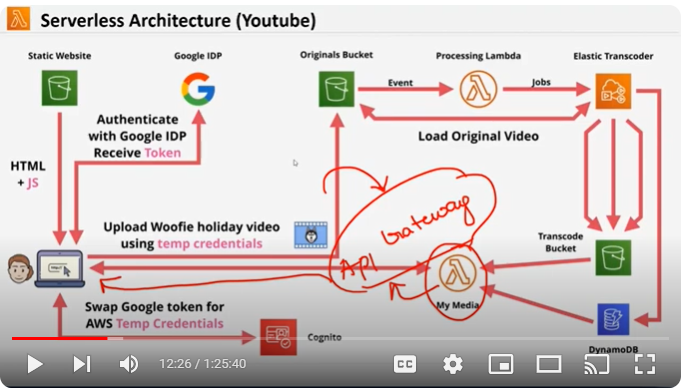
* **Here in s3 bucket we can only host static websites as it supports only static website that’s y we can use only html + js in backend we can use lambdas and for database we are using DynamoDB**
* **AWS cognito : its user pool for every application we need to create user and with that credentials only user can login to that particular application so aws cognito is aws managed service which can manage user sessions and user credentials**

1. **Usually user needs to manage these credentials in our own database but here aws is offering cognito as aws managed service**

****

1. **For example if you want to login to s3 static bucket you can only login if ur credentials are stored in cognito if not you need to create a new user**
2. **If you don’t have time to manage the user credentials you can use cognito**
3. **In cognito by default aws is offering authentications, forgot pwd,session maintainance,identity providers**
4. **So we can directly integrate cognito to the application by default**

* **Elastic transcoder is the service which coverts one video pixels to different formats for example 1080 pixels , 140 pixels , 360 pixels**
* **In the above pic processing lambda takes bucket name and object name from s3 and sends in the form of jobs to elastic transcoder**
* **Elastic transcoder processes the jobs and stores the results to another s3 bucket called transcode bucket**
* **Once the job is completed it updates in DynamoDB**
* **From front end to know the vedioes whether they are uploaded are not the mymedia lambda gets triggered after upload gets completed**
* **From dynamodb all the completed jobs will show in mymedia (lamda)**
* **Based on the url of jobs from dynamodb the objects from the s3 bucket will be returned in mymedia (lambda)**

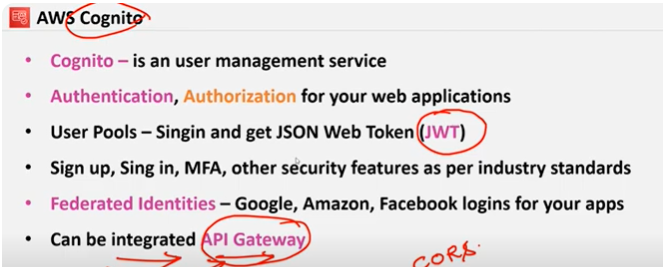
****

**From lambda(media) you will upload the url in api gateway and integrate api to frontend**

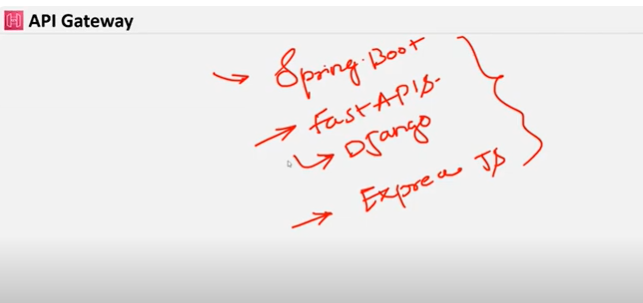
* **In youtube facebook etc what ever applications the sign in made by JWT tokens(json web token)**

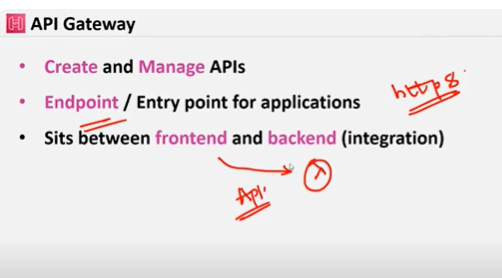
**Cognito :**

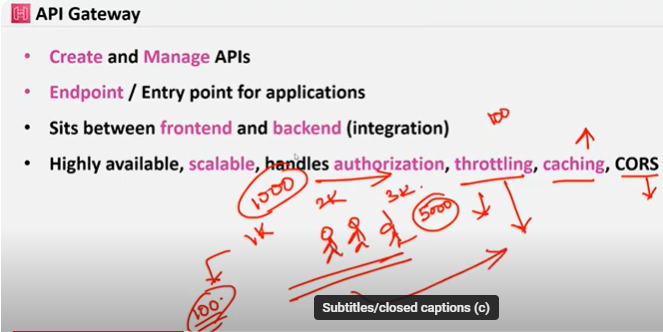
**By default u will get signup signin ,authentication**

****

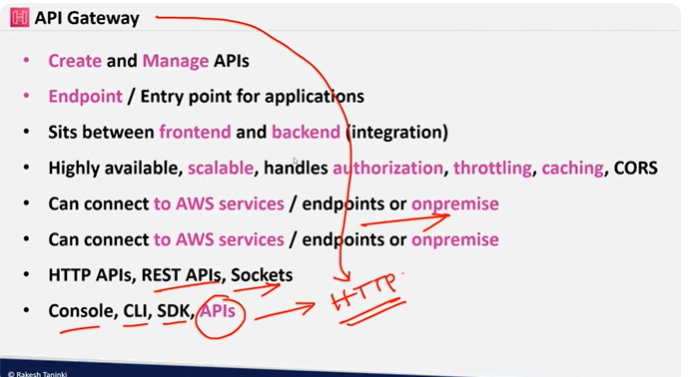
**API GATEWAY : used for rest API’S**

****

* **In java spring boot is an api library**
* **In python Fast Apis , Django is an api library.**
* **In nodejs express js is an api library for every language there will be rest api framework library**
* **If we want to use these libraries u need to create server in general to host these libraries**
* **To host these servers aws has provided one service called api gateway in this api gateway u don’t need to create any server u can directly host rest apis**
* ****
* **For example if you have website called example.com**
* **For api you might have api.example.com/login**
* **Even though domain will be same for each and everything like login ,signup,upload will have different apis and we can integrate each of them with lambda**
* ****

****

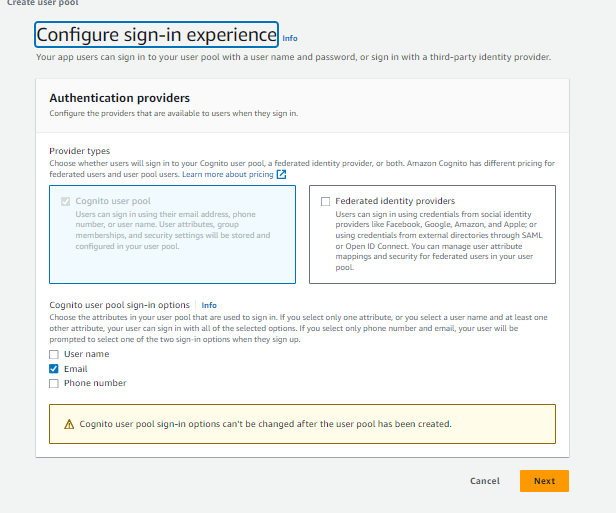
**In throttling we can set customer api calls limit per month based on pricing if it exceeds api wont accept the call request**

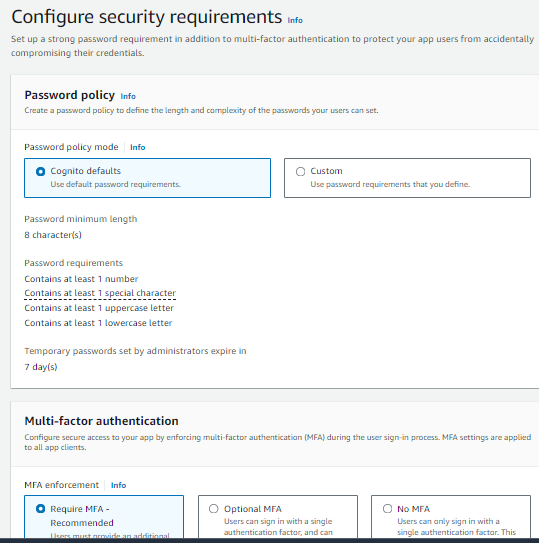
****

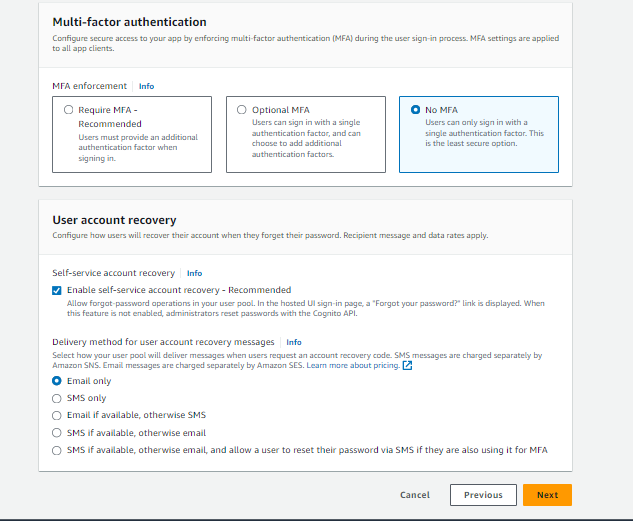
**One can connect to aws through console, cli,sdk,api’s**

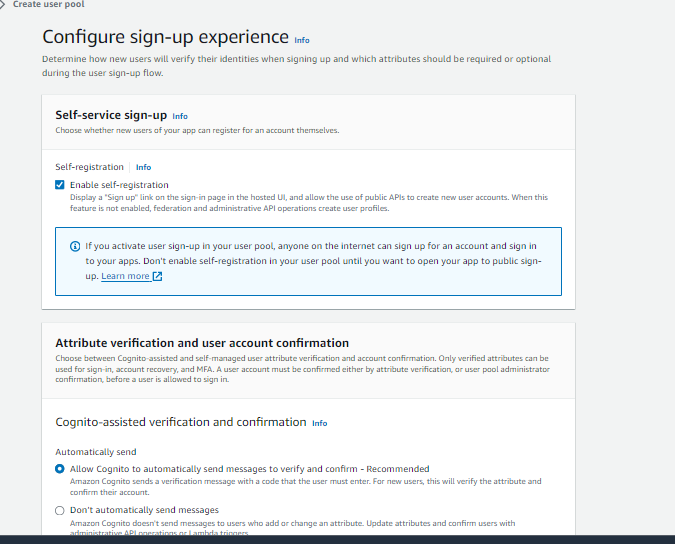
****

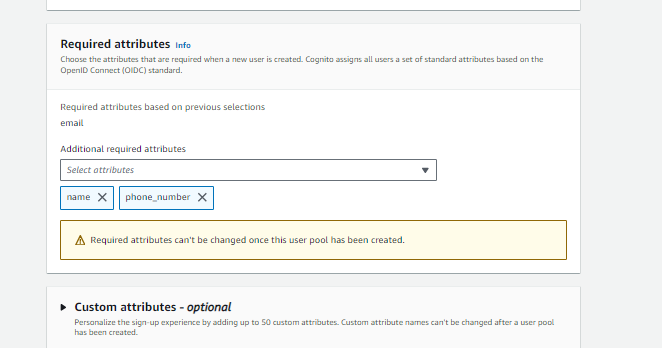
**Open aws console and open aws cognito**

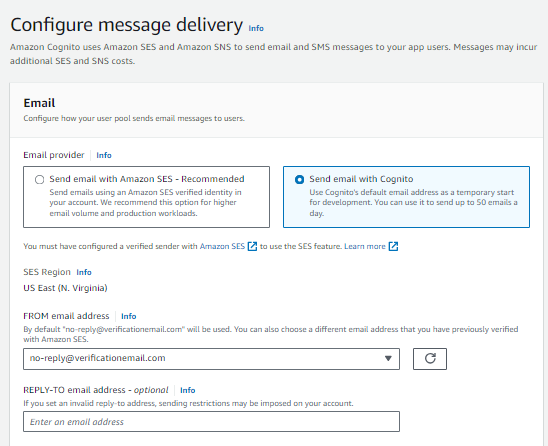
****

****

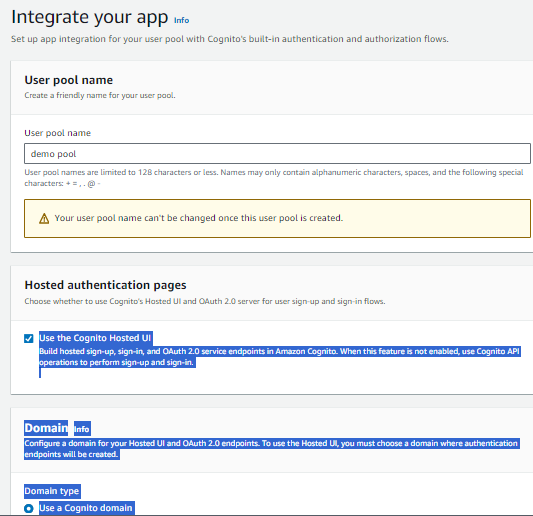
****

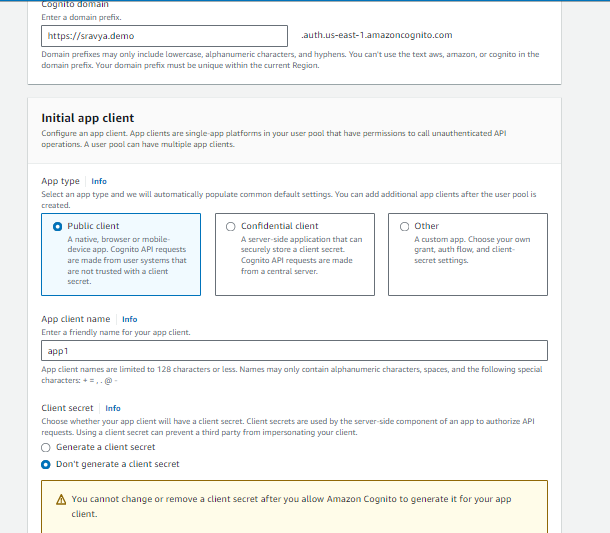
****

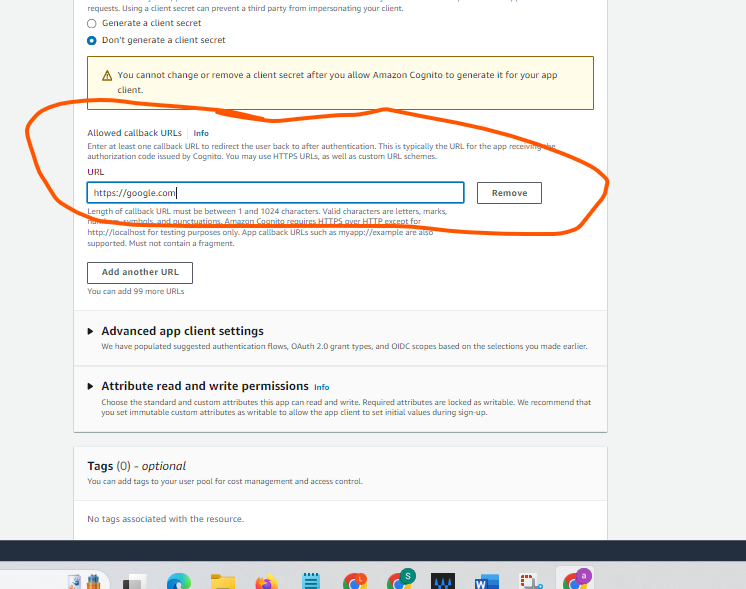
****

****

**In production level we can use SES for email service and with cognito you can only send 50 emails per day by default**

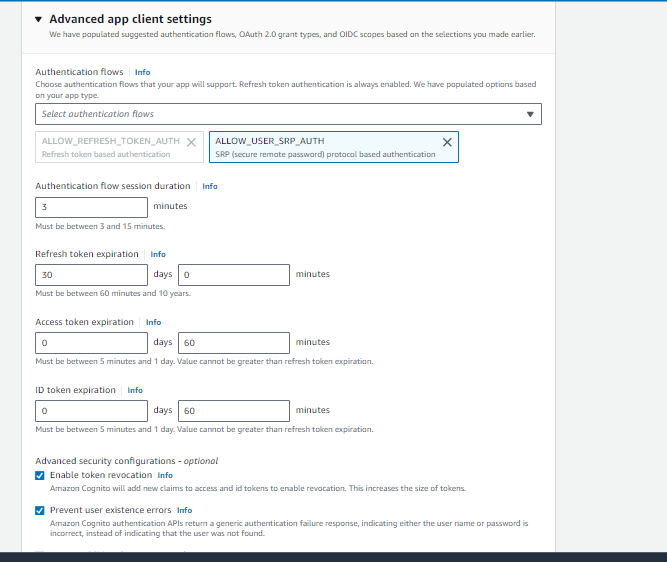
****

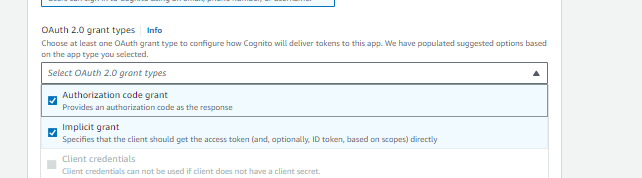
****

****

**After sign in which url it should redirect its asking for that url**

**Usuall in production level it redirects to the our application url**

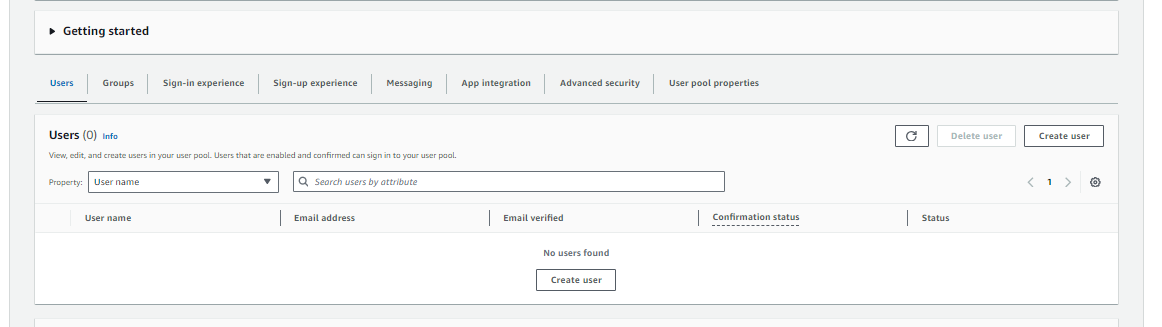
****

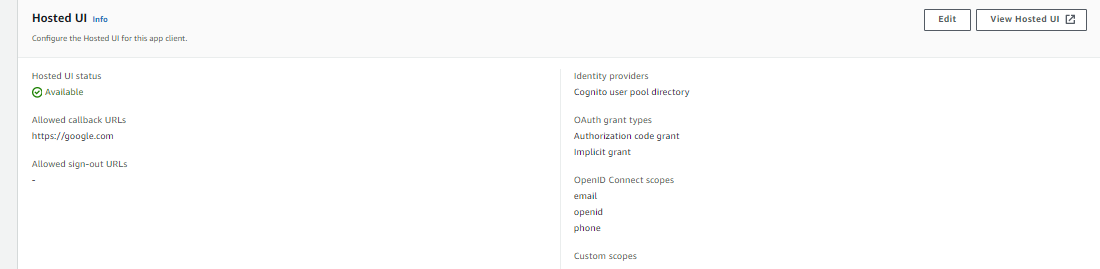
****

**After completion click on next and click on user pool creation**

**After creation u can view the cognito pool in that u can create users for application**

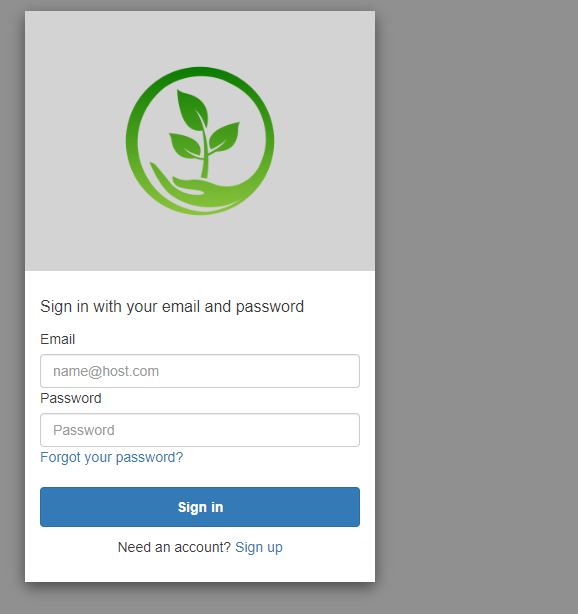
**U Can import existing users**

****

****

**You can add logo to here**

**After clicking on view hosted ui of app1 you will get this screen**

****

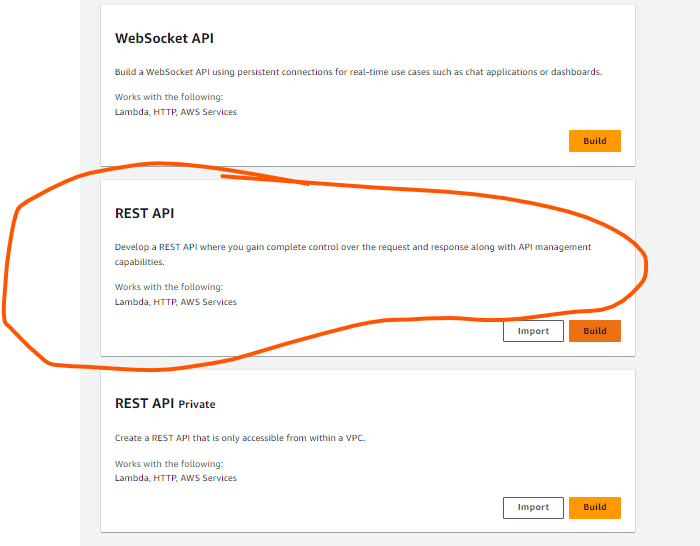
**You can signup**

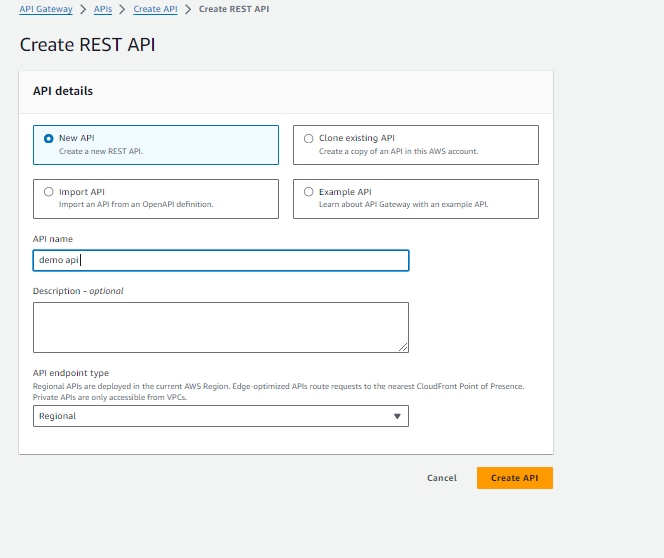
**After signing up u will get an verification code**

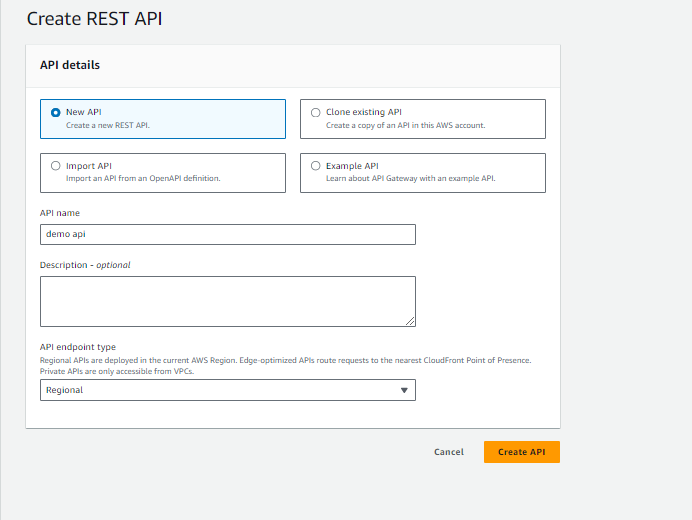
**It gets redirected to google .com**

**Now create aws api gateway**

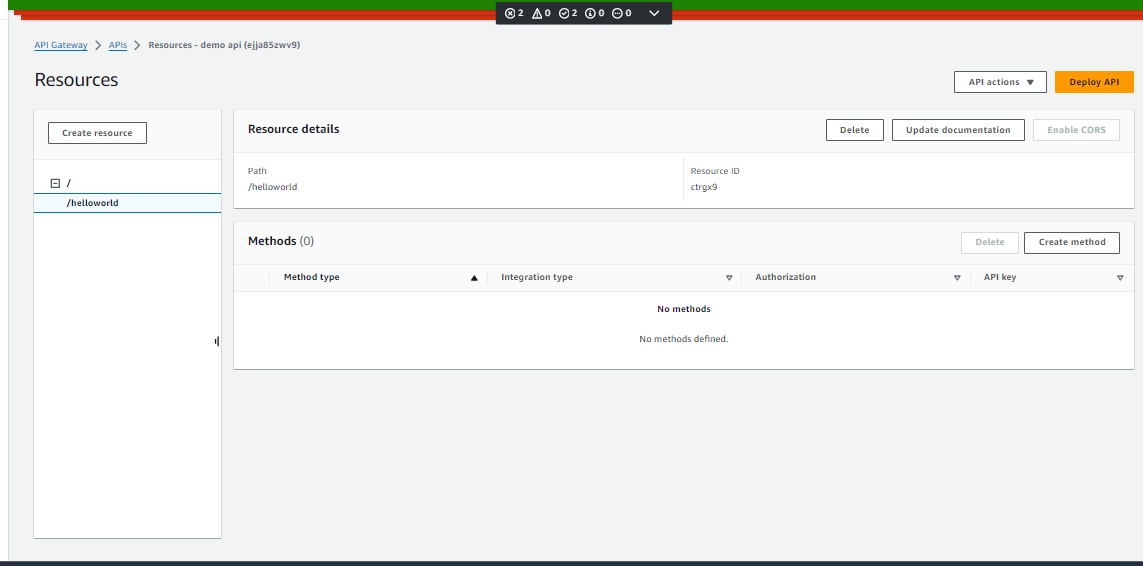
**Lets cfeate rest api public**

****

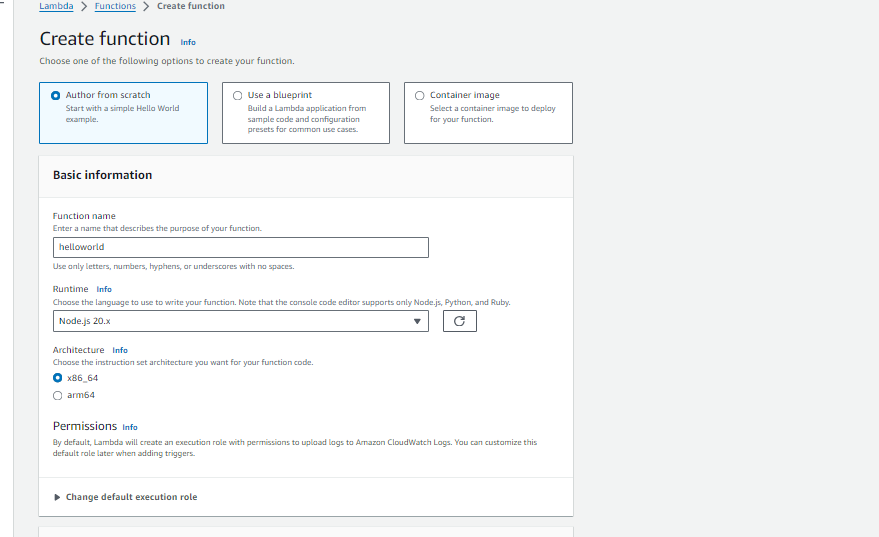
****

****

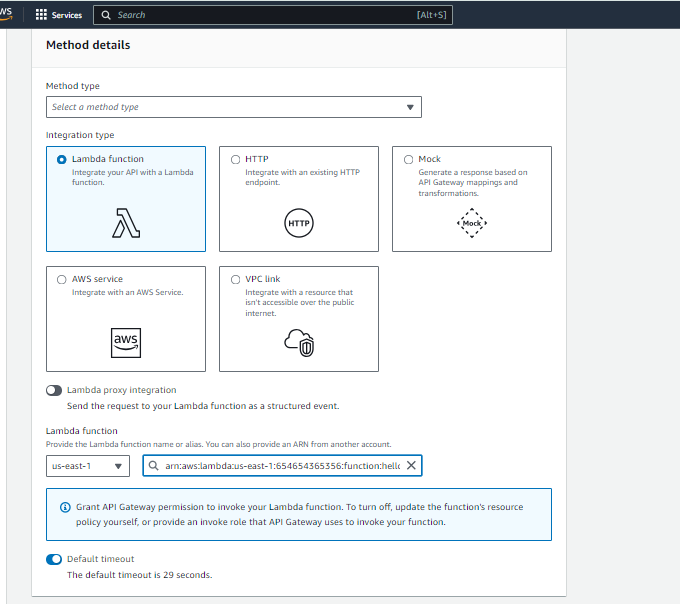
**Api gateway has been created**

****

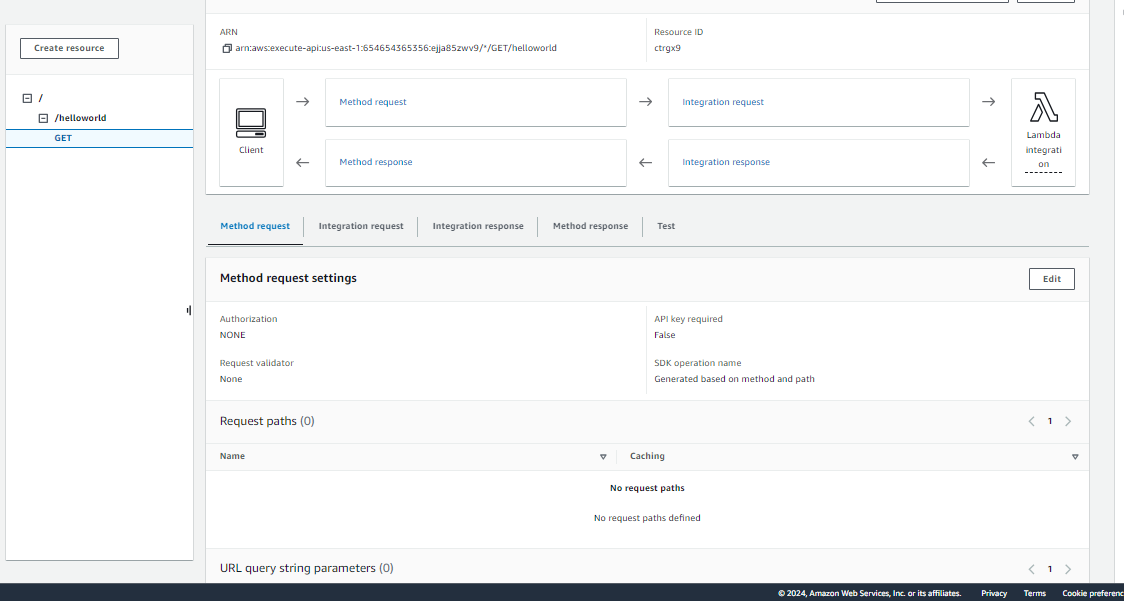
**After creating api create hello world**

****

**Create a lamda function naming hello world**

****

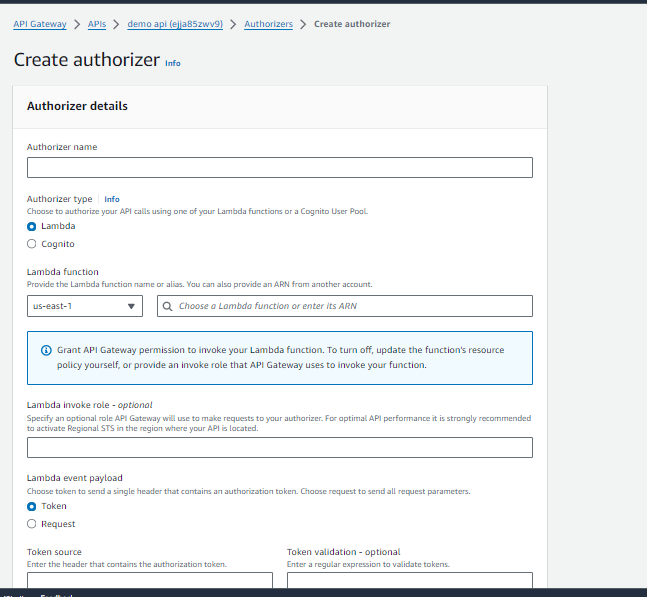
**Attach that lamda to api methods**

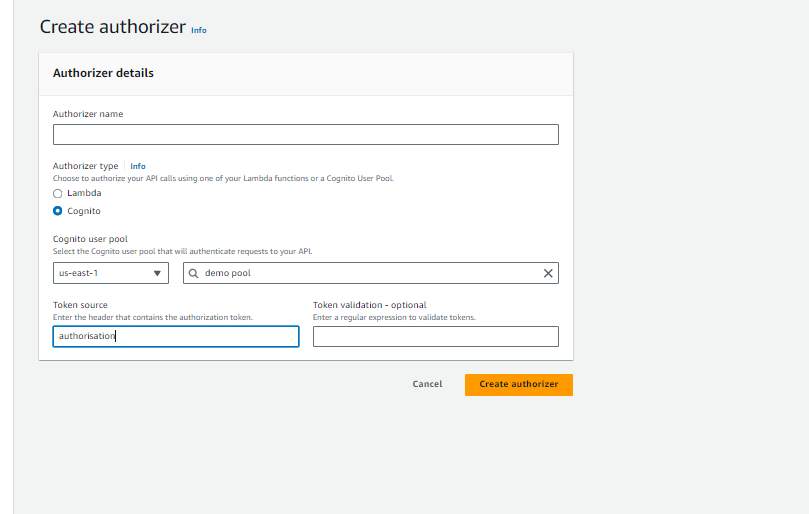
****

**You have to deploy api for api url**

**After deploying** [**https://ejja85zwv9.execute-api.us-east-1.amazonaws.com/v1**](https://ejja85zwv9.execute-api.us-east-1.amazonaws.com/v1)**/hellowold you will get api gateway url if you access that url it trigger to lambda and gives response**

**This url can be accessed by anyone so please create authorizers for this particular url**

****

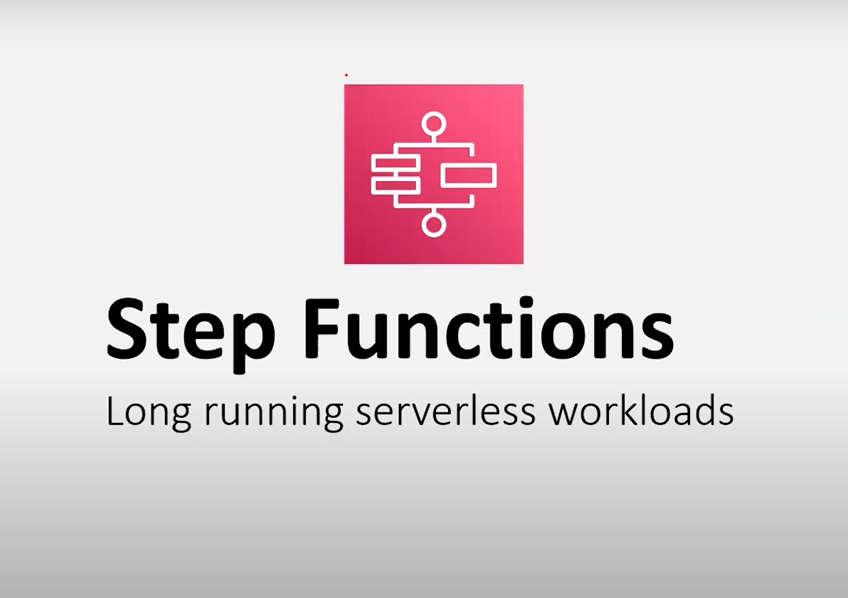
****

**When u create cognito u will get one token and when u create api u will get one url**

**And when u keep both url and token in post man by using get method u will get the response from lambda**

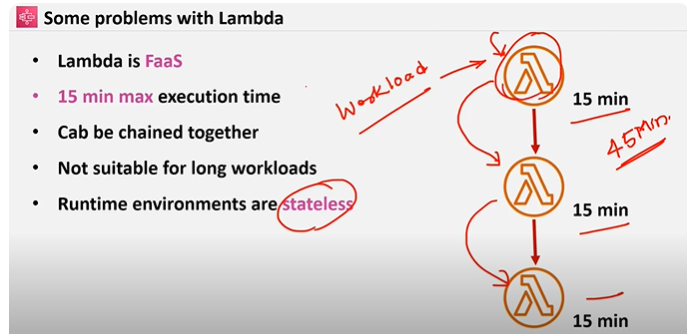
**The whole process of the demo is u need to create cognito and u need to create the user for sign in to application in that cognito u just give backend url which it need to be routed after user gets signed in later create api and give cognito name to api in the resources section give names and add methods like get post etc and integrate lambda function to it once you deploy the api u will get url of api so when u call the url of api the lambda gets triggered and gives the response**

**If there is security in api there is authorisation for security it means in cognito created will get some token for authorisation by keepig that token and api url in post man site and selecting particular methods you will get response from lambda**

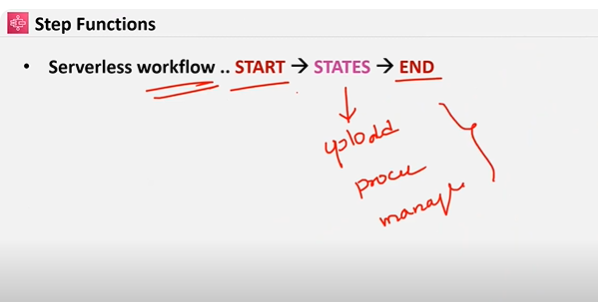
****

**In lambda u cant run more than 15 mins**

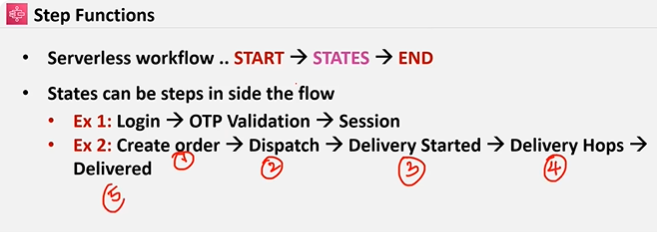
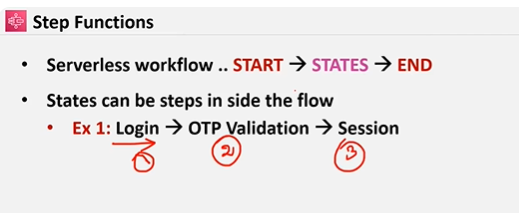
**So for longer run u need step functions and its also serverless**

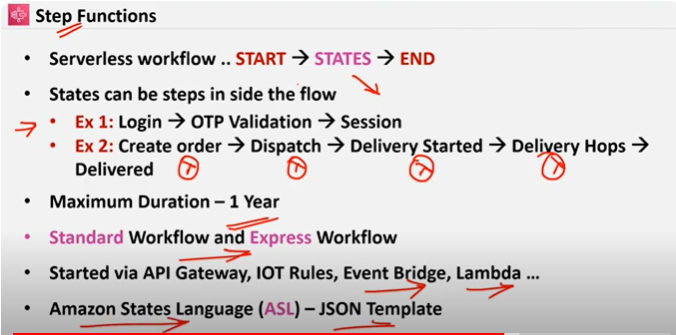
****

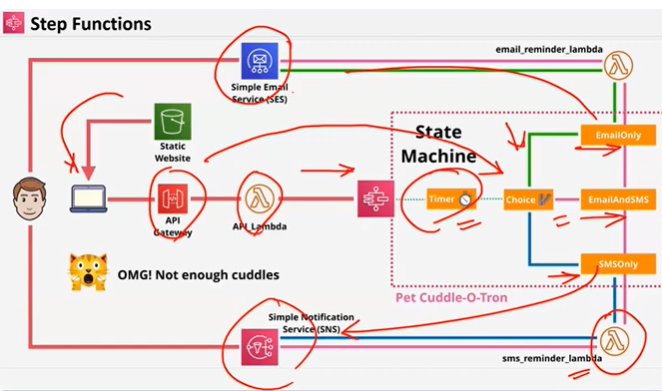
**Some problems with lambda are if there is a long you need to create multiple lambdas for multiple piece of codes**

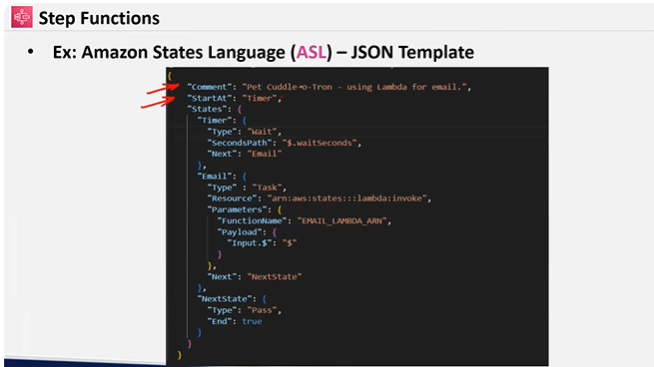
****

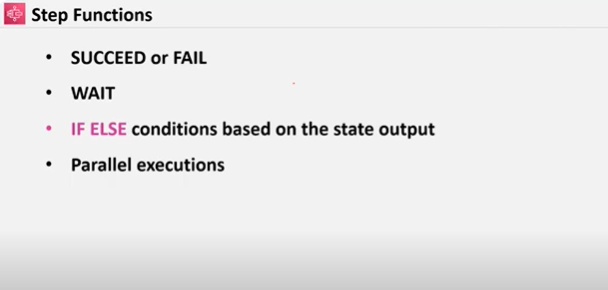
**In step functions there are serverless workflow strt and end and in middle there are states that means let us take example as youtube in youtube upload will be different state manage will will be different state process will be different state**

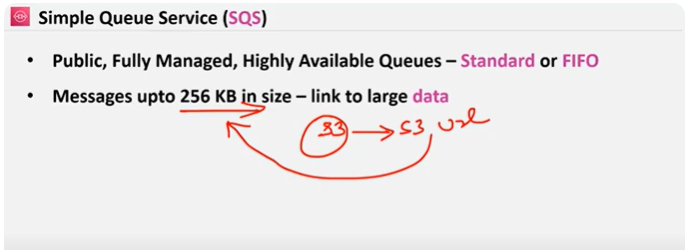
****

****

****

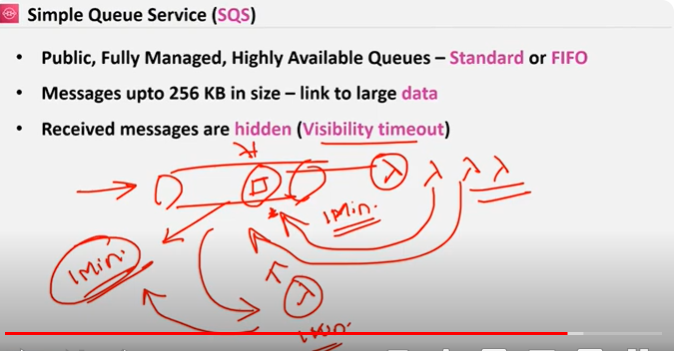
****

****

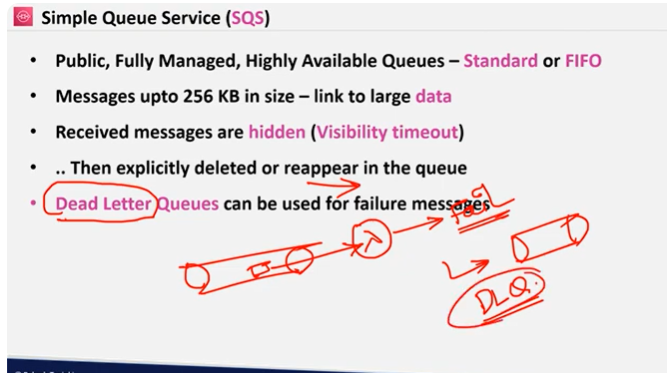
****

**SQS is highly available queues which is standard or FIFO in standard aws doesnot guarantee that 1st element will comeout first but in Fifo it is possible but aws charges extra for it**

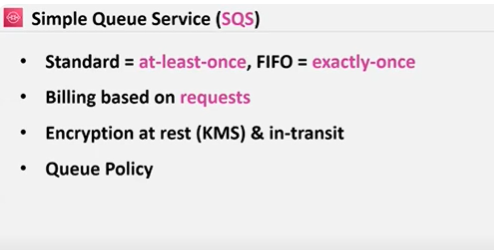
* **Messages upto 256 kb in size if u want large data to upload you will upload that file in s3 and will link that url to 256kb and when lambda triggers it downloads that file from s3**

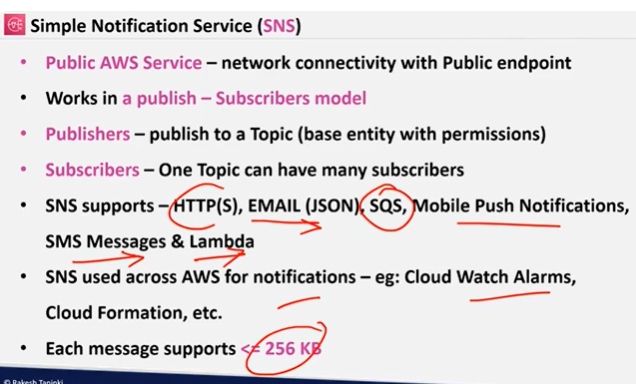
****

**Visibility time out is that if in queue there are certain messages and there are so many lambdas so if all the lambdas picks that messages its an unnecessary timewaste so when 1 lamda picks that message it should be kept under visibility timeout for that particular time so other lamdas cant see that message**

****

**In dead letter queues all the failed queues gets stored to know y they are failed**

****

****