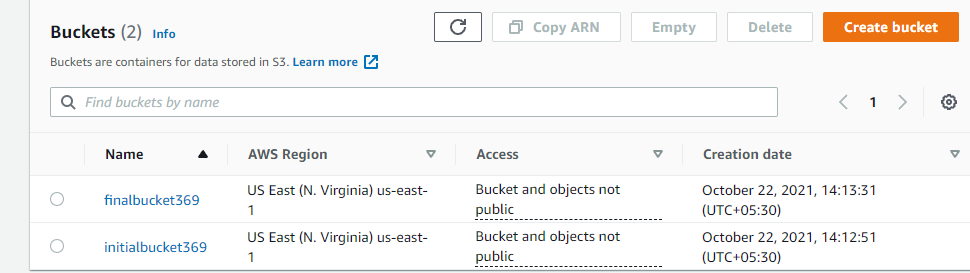
**AWS Lambda Triggering Step function**

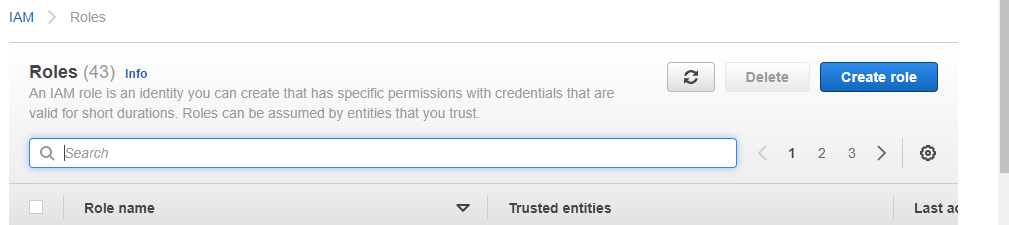
**Create two S3 buckets**

****

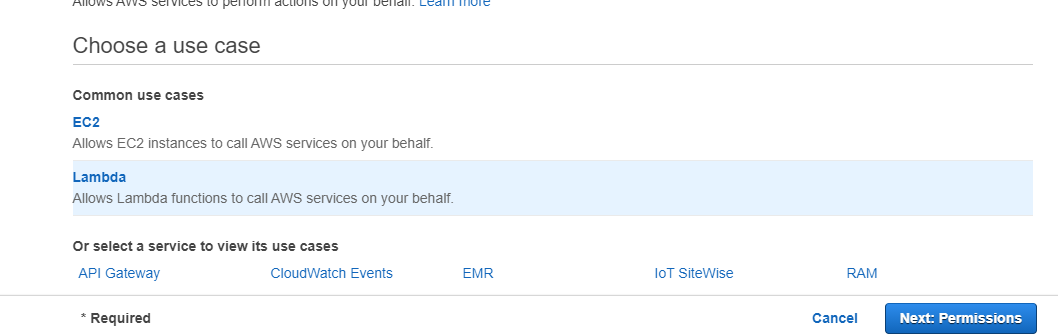
**Creating IAM role**

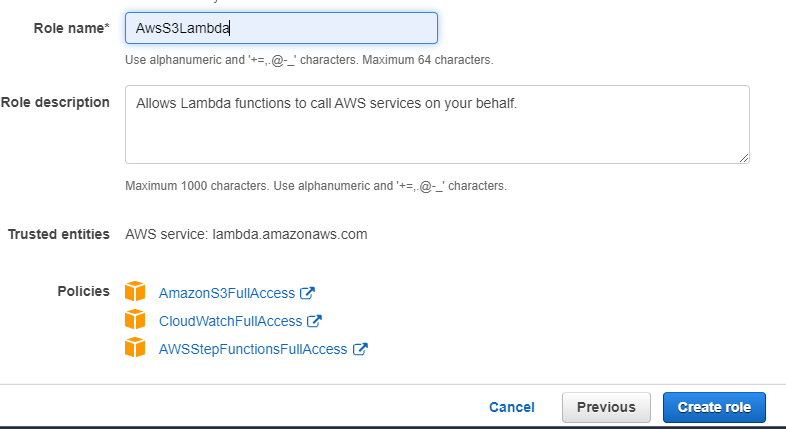
We need an IAM role that allows AWS S3 Full access, AWS Cloud watch full access, AWS step function full access

Go to IAM and Create role



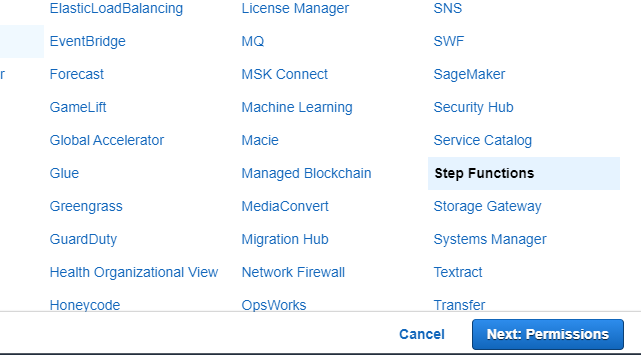
Choose Lambda

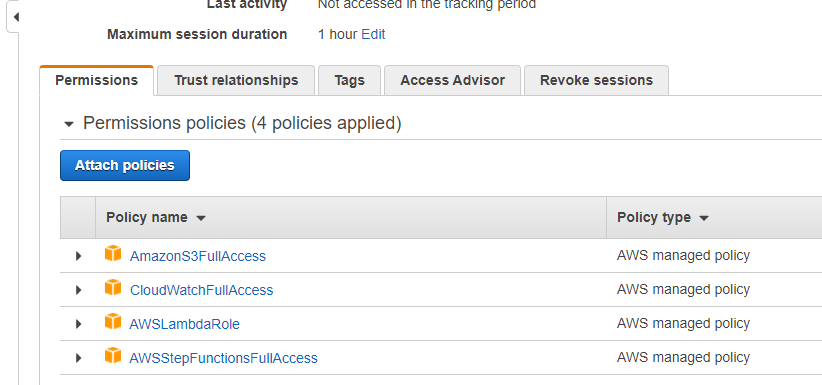
****

****

Now Create a Step function role that allows AWS S3 Full access, AWS Cloud watch full access, AWS step function full access

Go to IAM choose step function

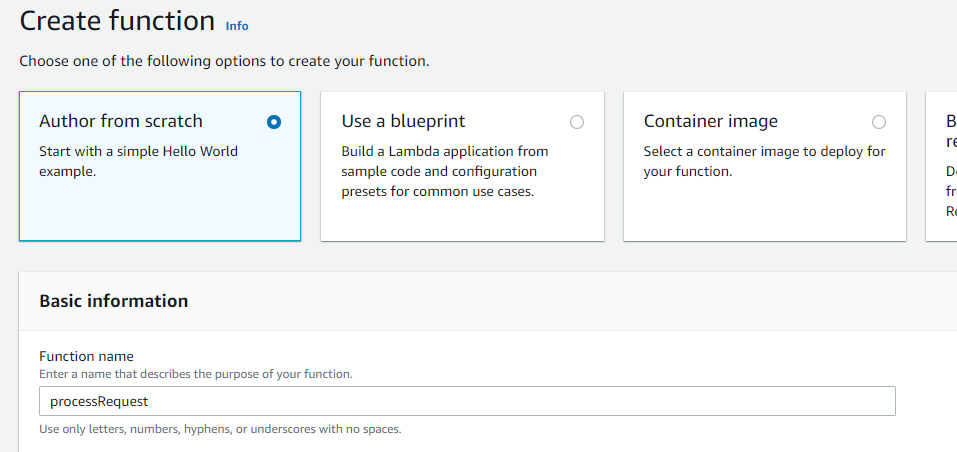


****

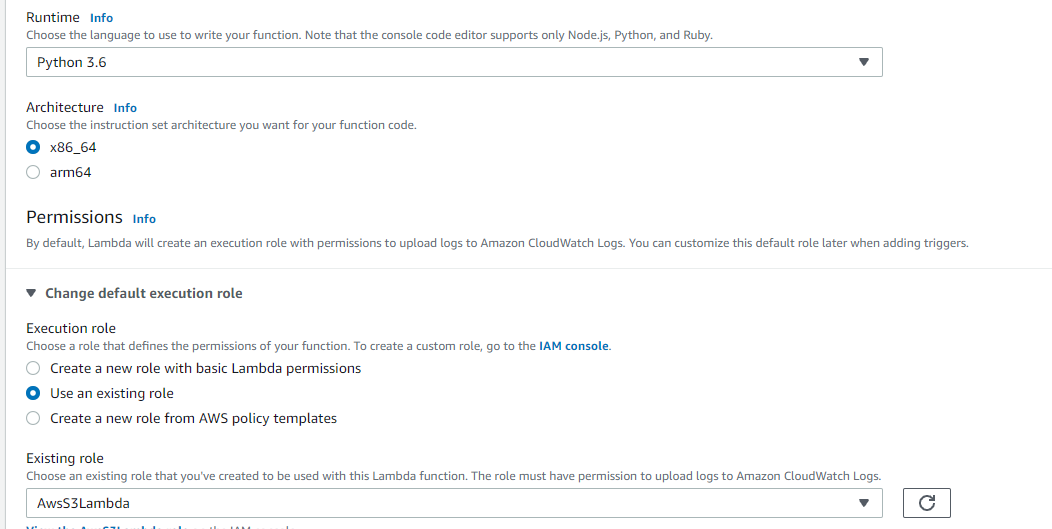
**Aws Step Funtion**

1. **Request**
2. **Response**
3. **Fail**

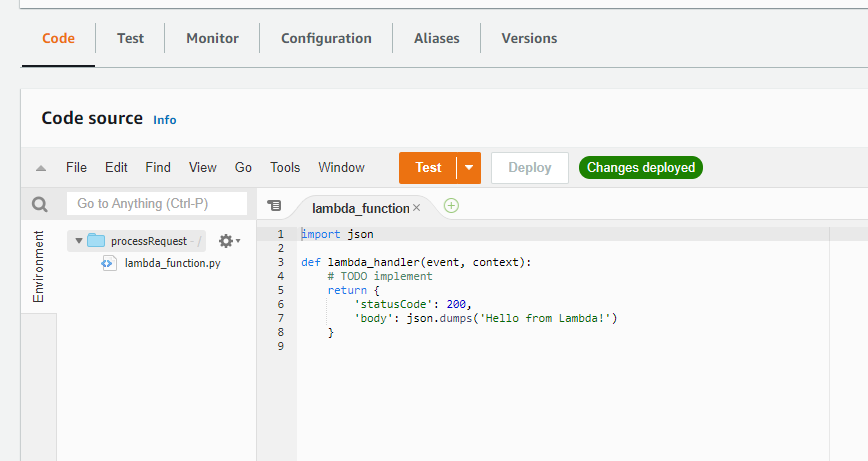
**Go to Aws Lambda and create three Lambda function for Request, Response, and Fail**

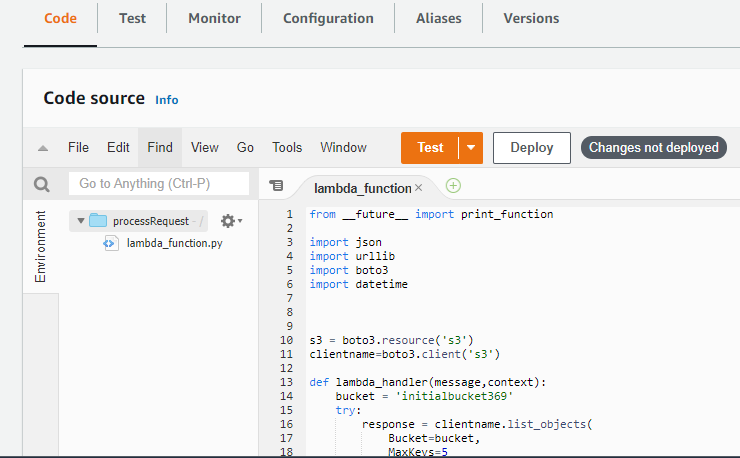
****

**Attach our previously created role**

****

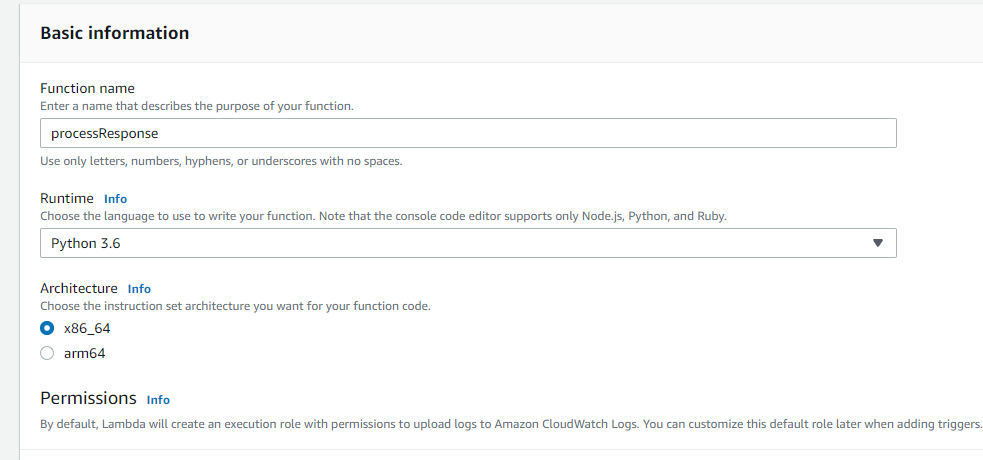
Now you can see a default python code , change that code with our code and deploy it

****

****

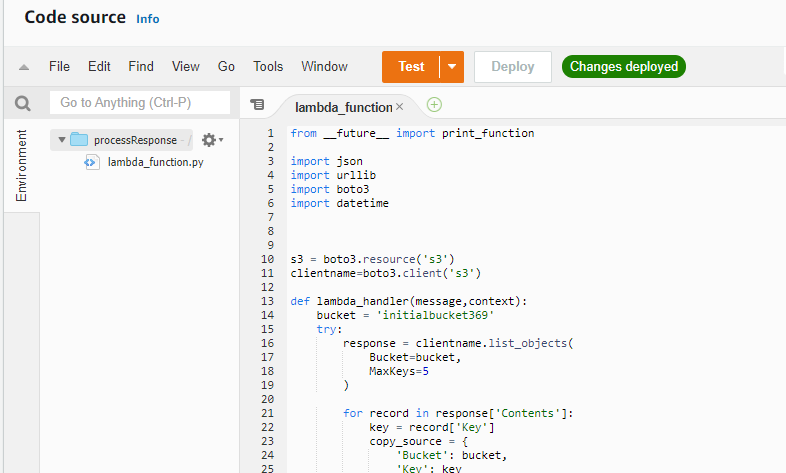
<https://github.com/vishnuvs369/Aws-Python/blob/main/Aws-stepfn-working-final/process_request.py>

Now similiarly for Response and Fail we need to create lambda function

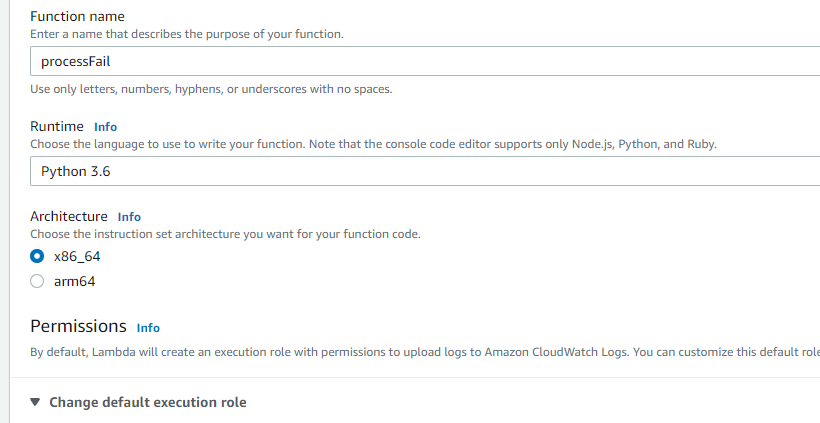


Now Add your code and deploy it

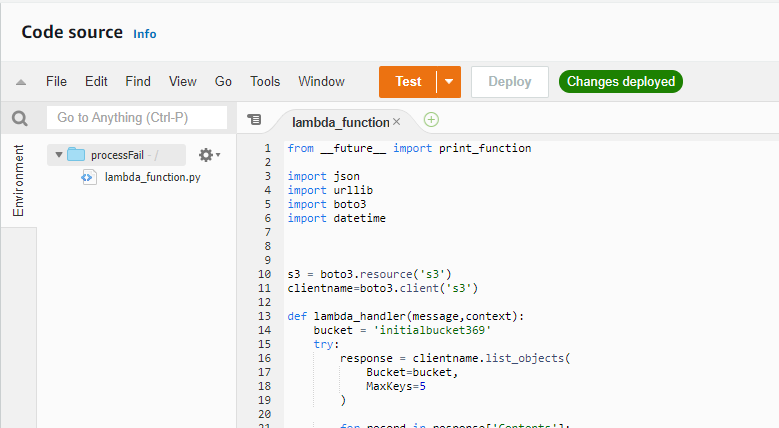
https://github.com/vishnuvs369/Aws-Python/blob/main/Aws-stepfn-working-final/process\_response.py



Now for Fail

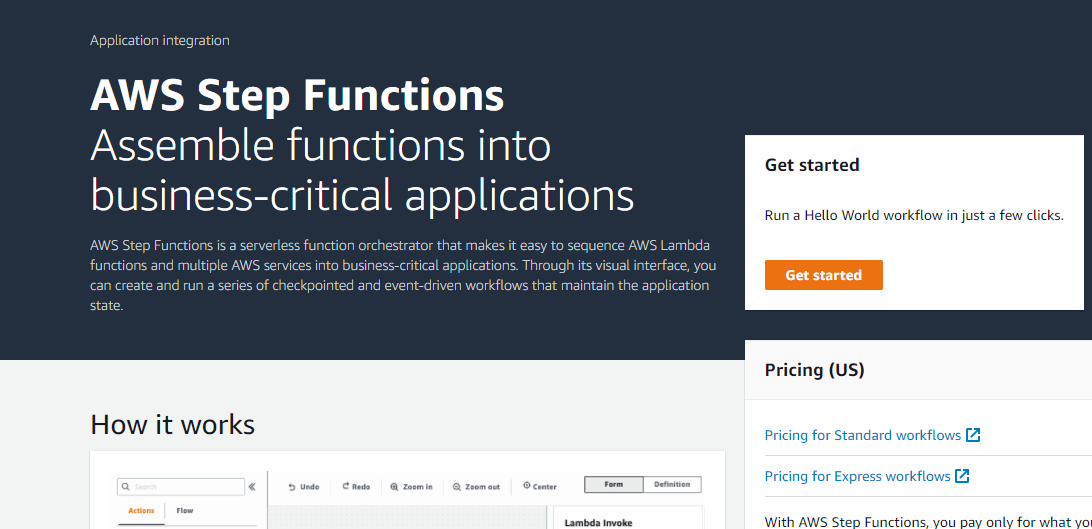
****

<https://github.com/vishnuvs369/Aws-Python/blob/main/Aws-stepfn-working-final/process_fail.py>

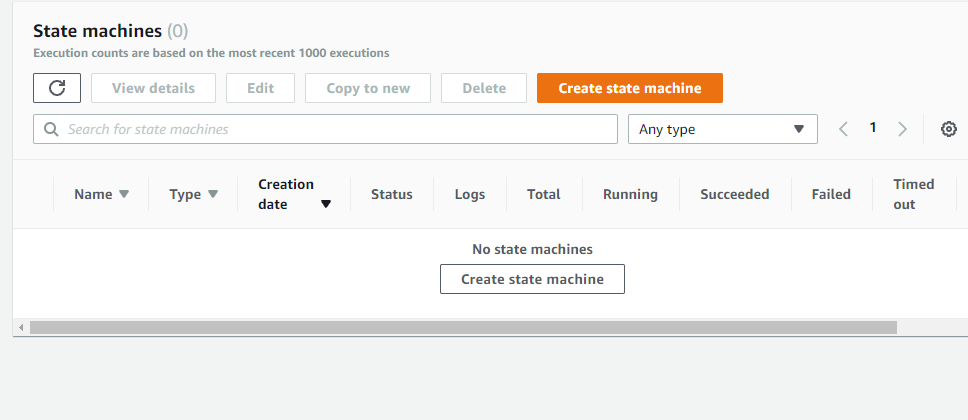


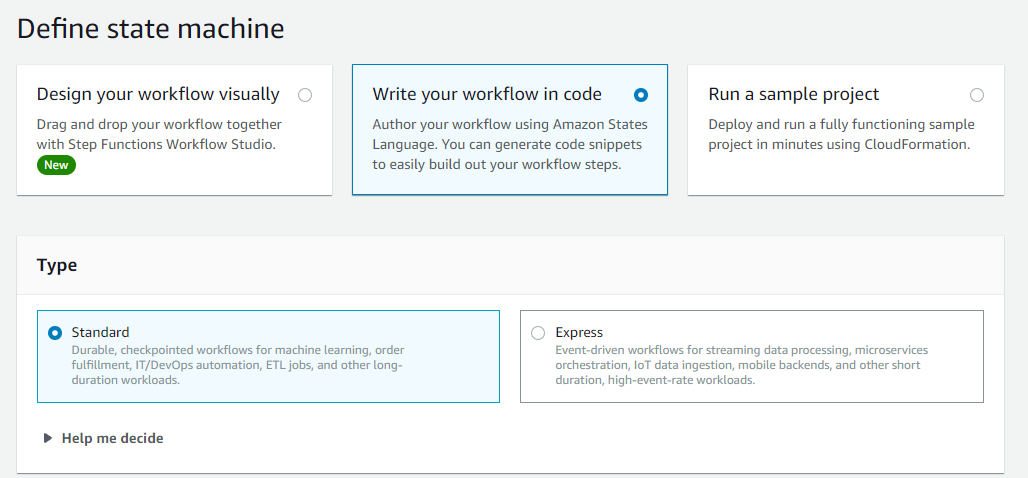
Now We can create our Step Function

Go to Step Function Console

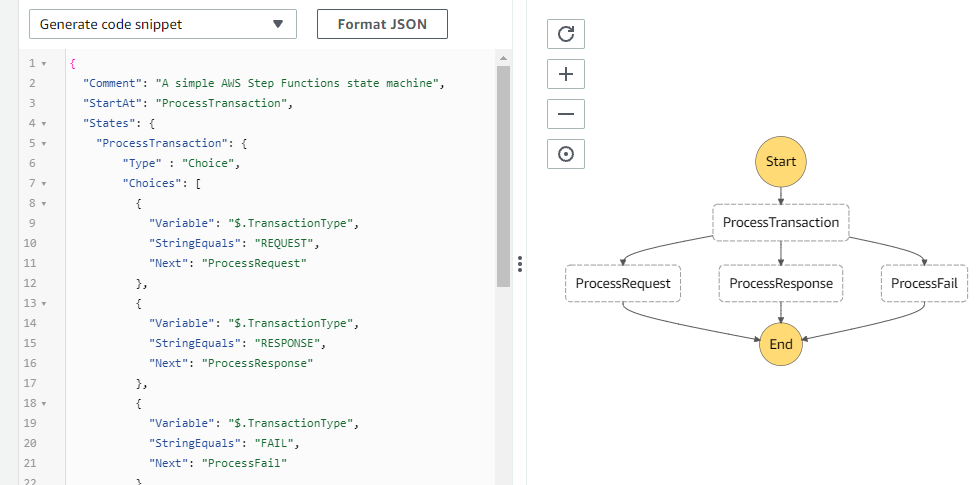


Create a state machine





Now add your workflow code there

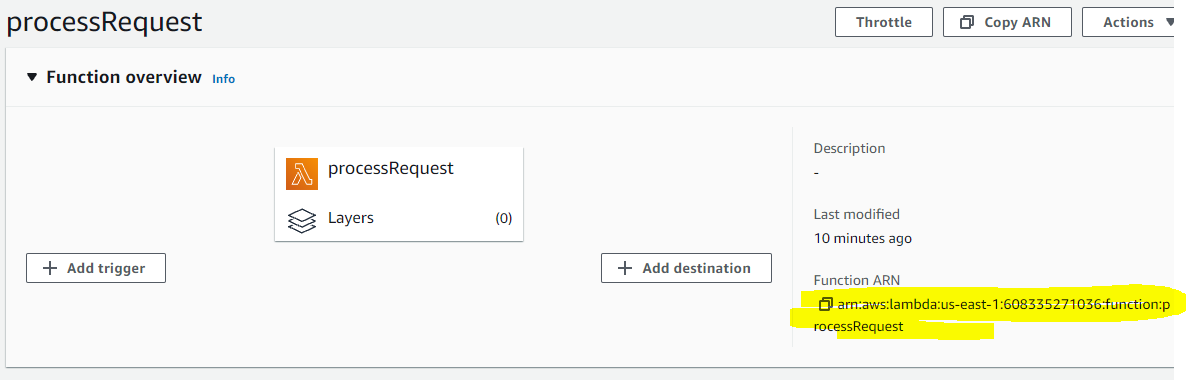


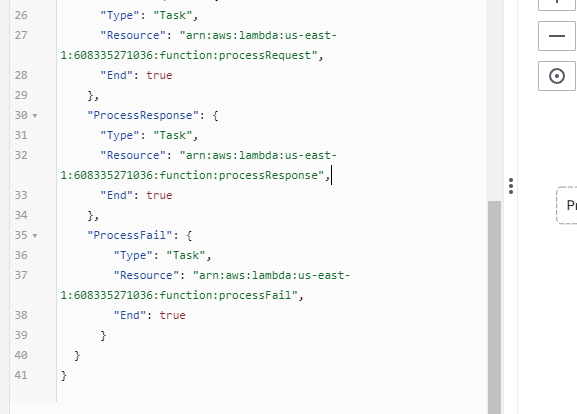
<https://github.com/vishnuvs369/Aws-Python/blob/main/Aws-stepfn-working-final/MyStateMachine>



Now you can see error in our workflow code, so there we need to add the ARN of previously created lambda functions.

Go to our Request, Response, Fail Lambda function and copy the ARN and paste it here.

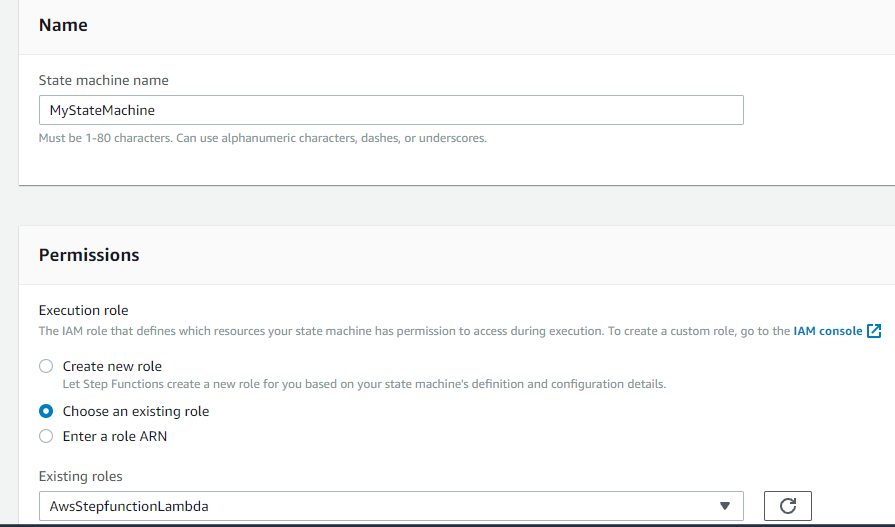




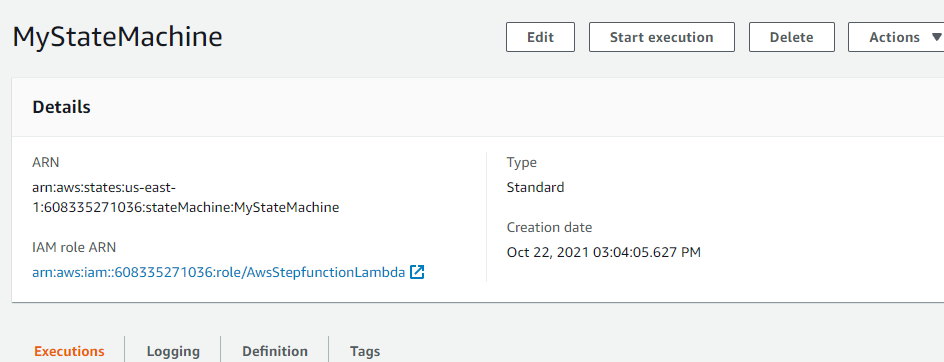
Now the error gone..

Now click on next and create state machine

Choose a Name and Attach our previously created Stepfn role



So the State machine is created, copy the ARN of State machine



**Create Our Main Lambda Function and Deploying packages with dependencies**

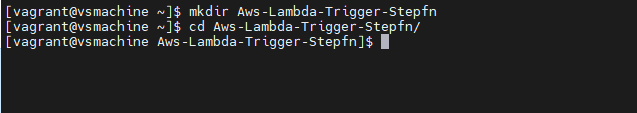
To create the deployment package

1. Open a command prompt and create a my Aws-Lambda-Trigger-Stepfn project directory.

For example

mkdir Aws-Lambda-Trigger-Stepfn

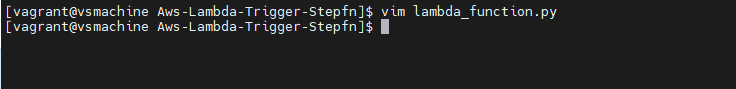
1. Then Navigate to the  project directory.



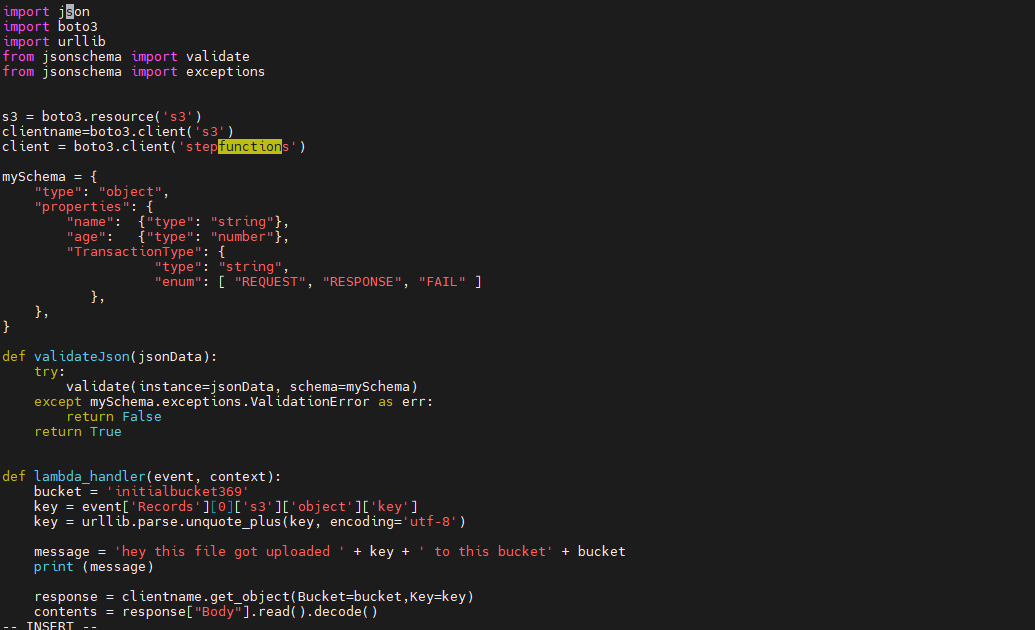
1. Copy the contents of the following sample Python code and save it in a new file

named lambda\_function.py:

$ vim lambda\_function.py

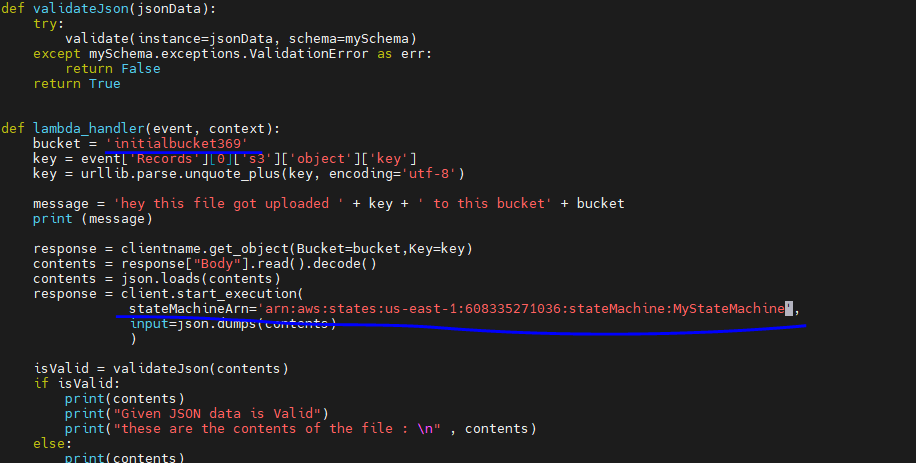


Add our code there



<https://github.com/vishnuvs369/Aws-Python/blob/main/Aws-stepfn-working-final/lambda_function.py>

Now edit the code and Add your source bucket name and your previously created State machine ARN there and save it.

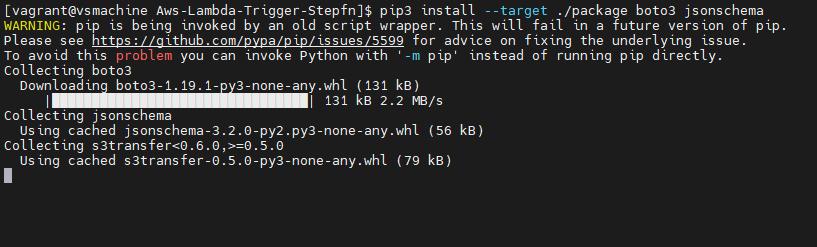


Your directory structure look like this:

Aws-Lambda-Trigger-Stepfn$

| lambda\_function.py

4.Install the requests library to a new package directory.



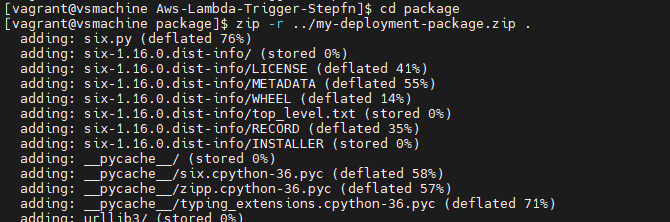
$ pip3 install --target ./package boto3 jsonschema

5.Create a deployment package with the installed library at the root.

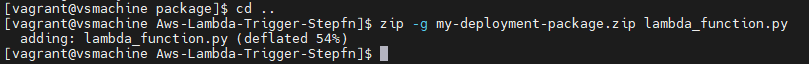
cd package

zip -r ../my-deployment-package.zip .

This generates a my-deployment-package.zip file in your project directory



6. Add the lambda\_function.py file to the root of the zip file.

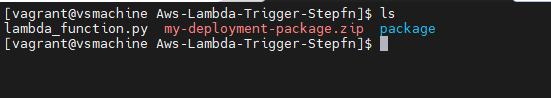


cd ..

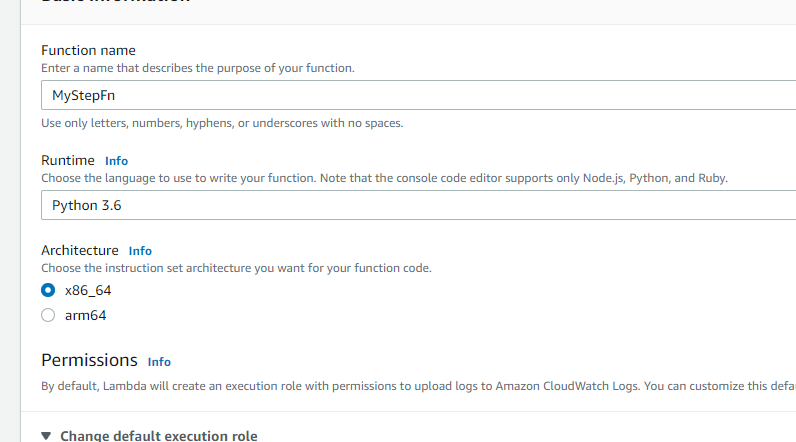
zip -g my-deployment-package.zip lambda\_function.py

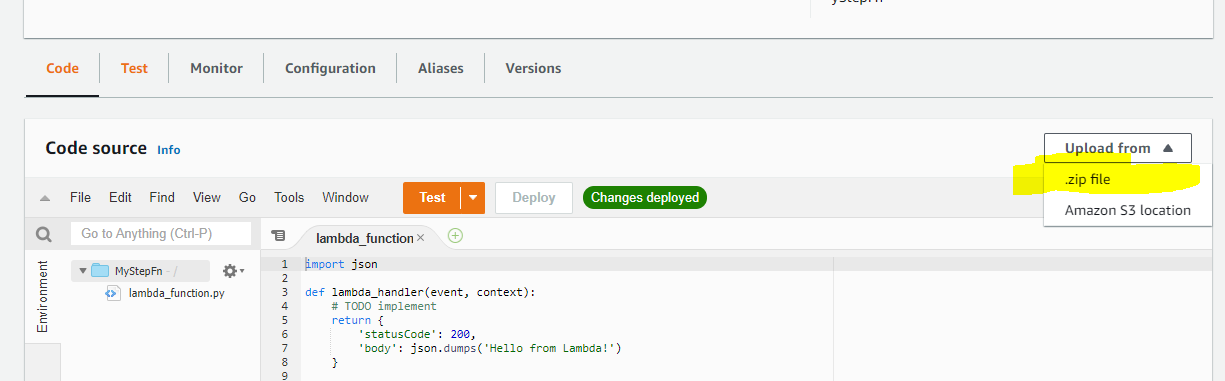
7. Deploy your .zip file to the function

To deploy the new code to your function, you upload the new .zip file deployment package. You can use the Lambda console to upload a .zip file to the function.

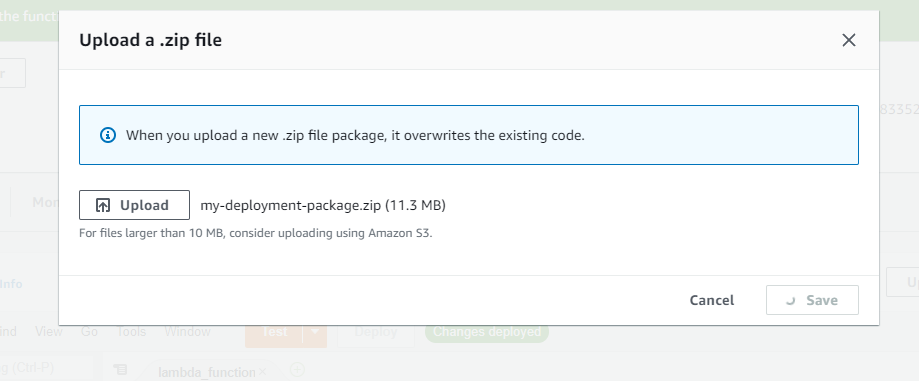


Go to Lambda console and create the function



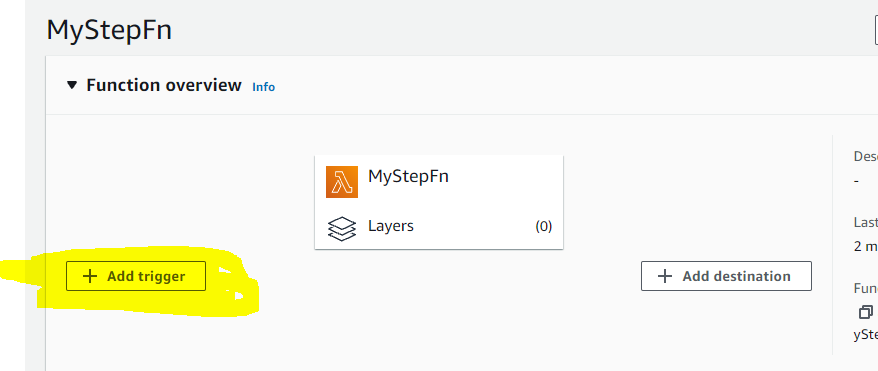


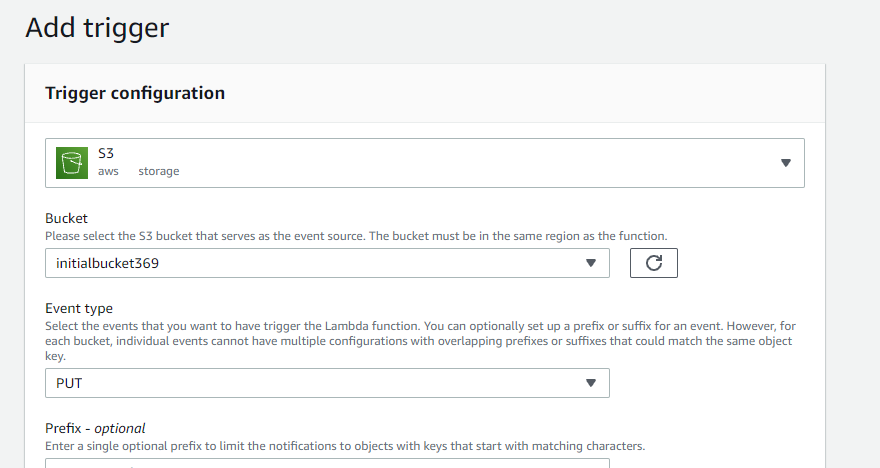
Now upload our zip code



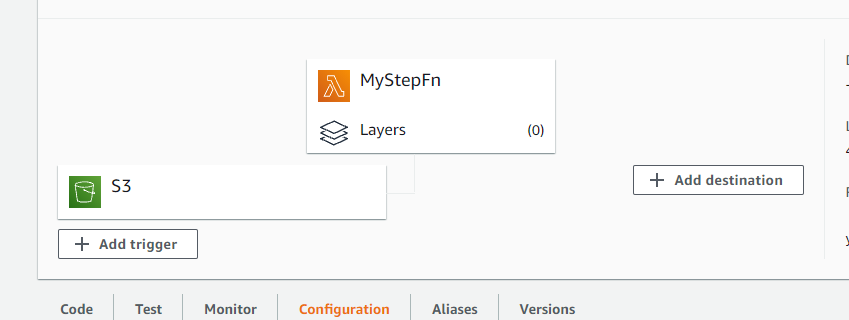
Now Add your S3 trigger

Go to Add triggers and select S3, Choose your bucket



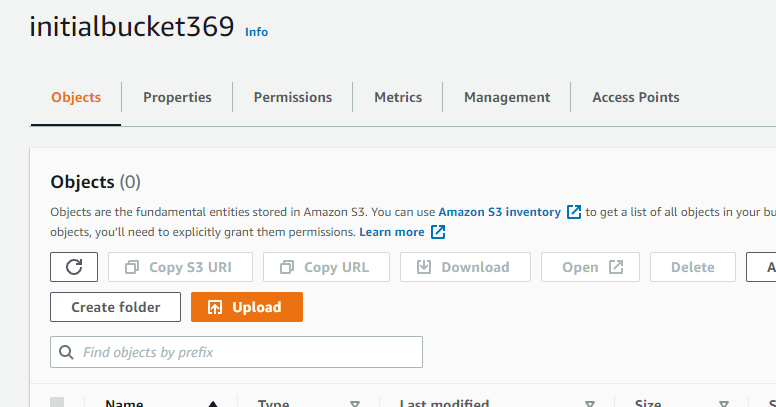


Now S3 trigger is Added



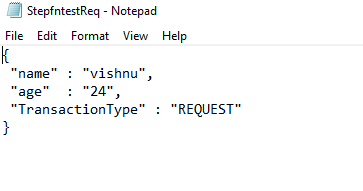
So the process is complete, Now we can test our function

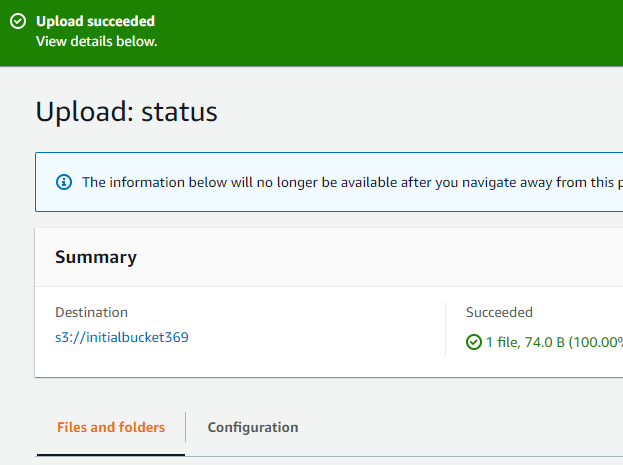
Just go to our Source bucket and Upload a json file



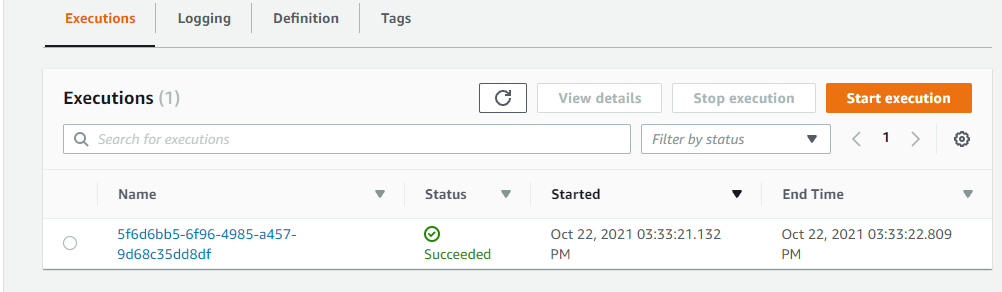
This is demo json which contains Type as Request

If it works, when we upload this file to our source bucket it will go to respective step function and transfer the file to our destination bucket

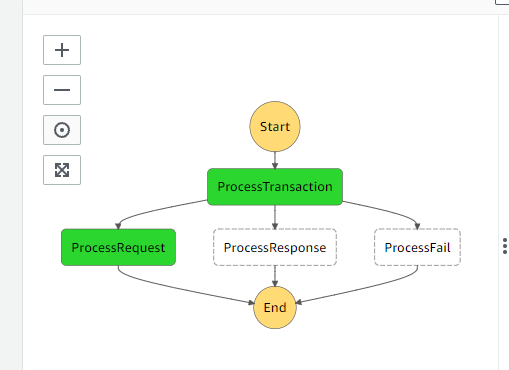




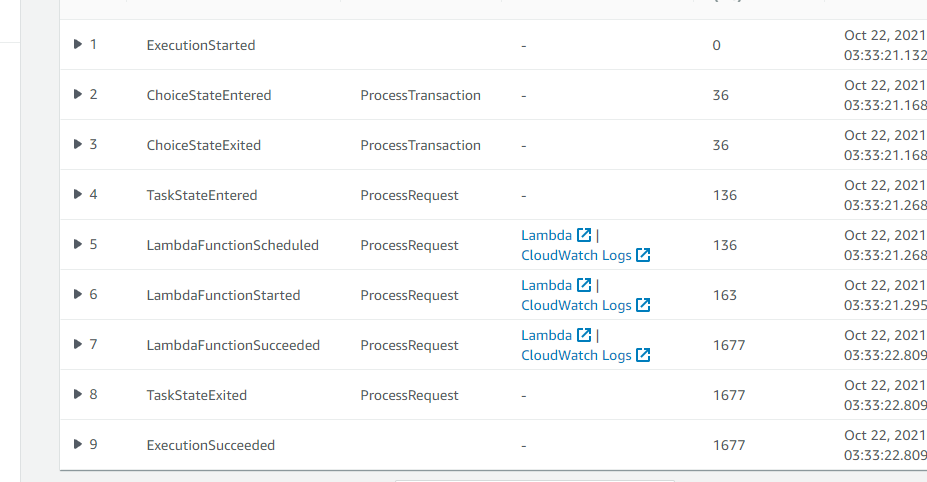
Now go to our step function where you can see the logs

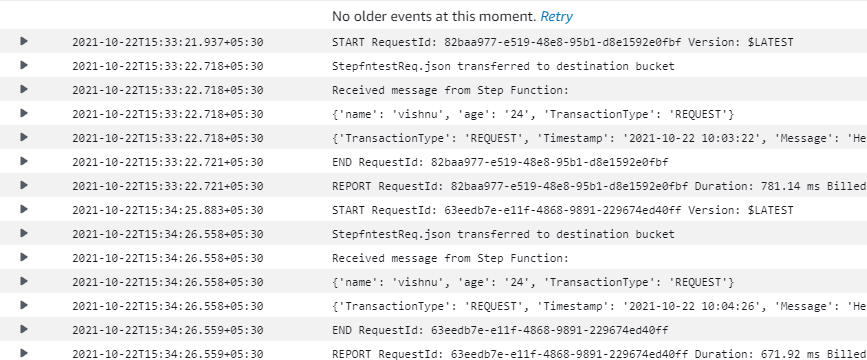


It is succeeded open it, you can see it automatically triggered to our step function



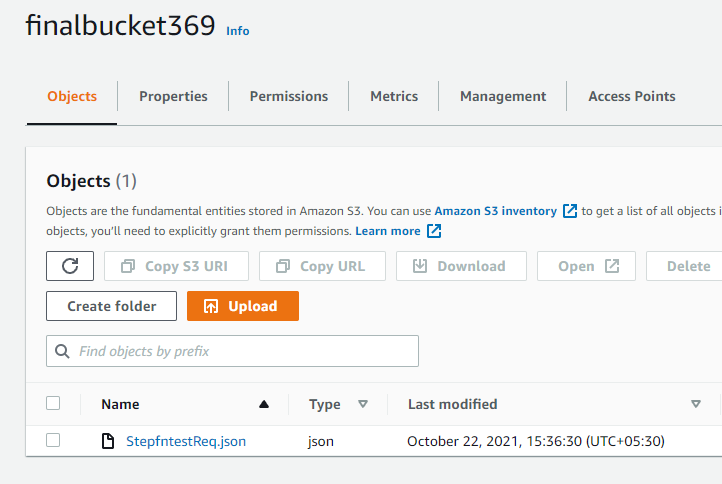
Now open the cloud watch to see more logs





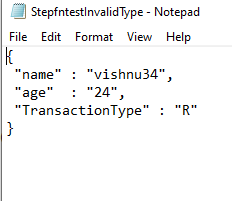
You can see the logs there.

Now check the destination bucket

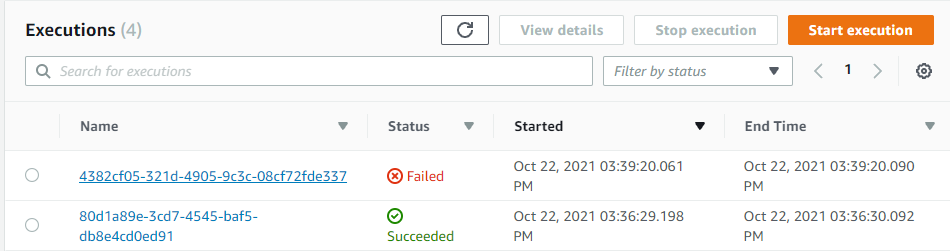


The uploaded file is transferred to our Destination bucket

If we upload a Invalid json the the execution become fails



This is an invalid json now im going to upload this



Now you can see the execution fails, Check the logs

