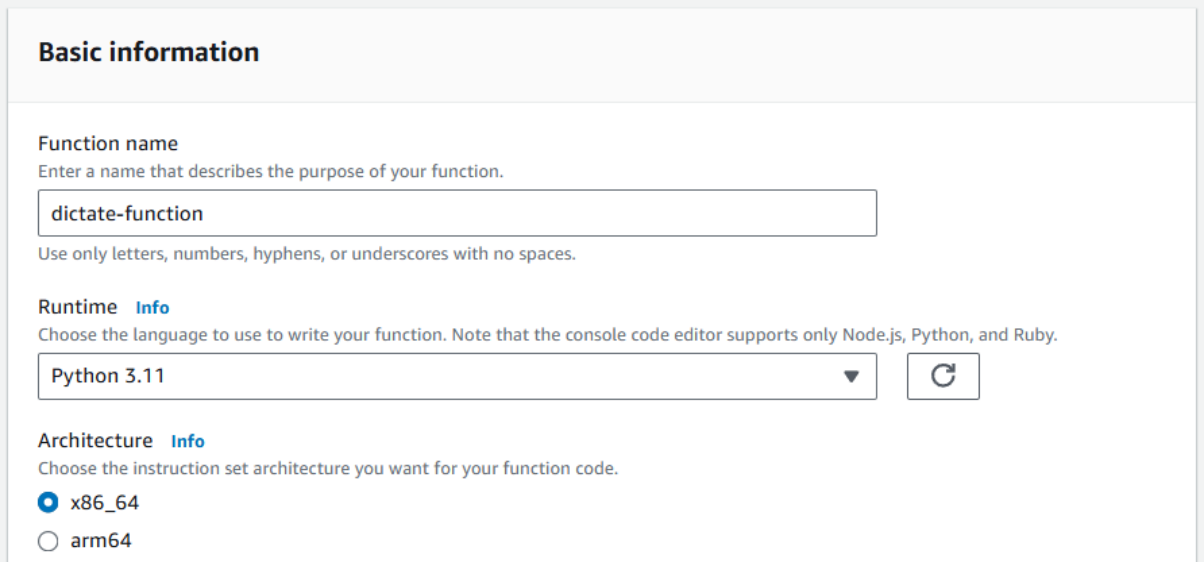
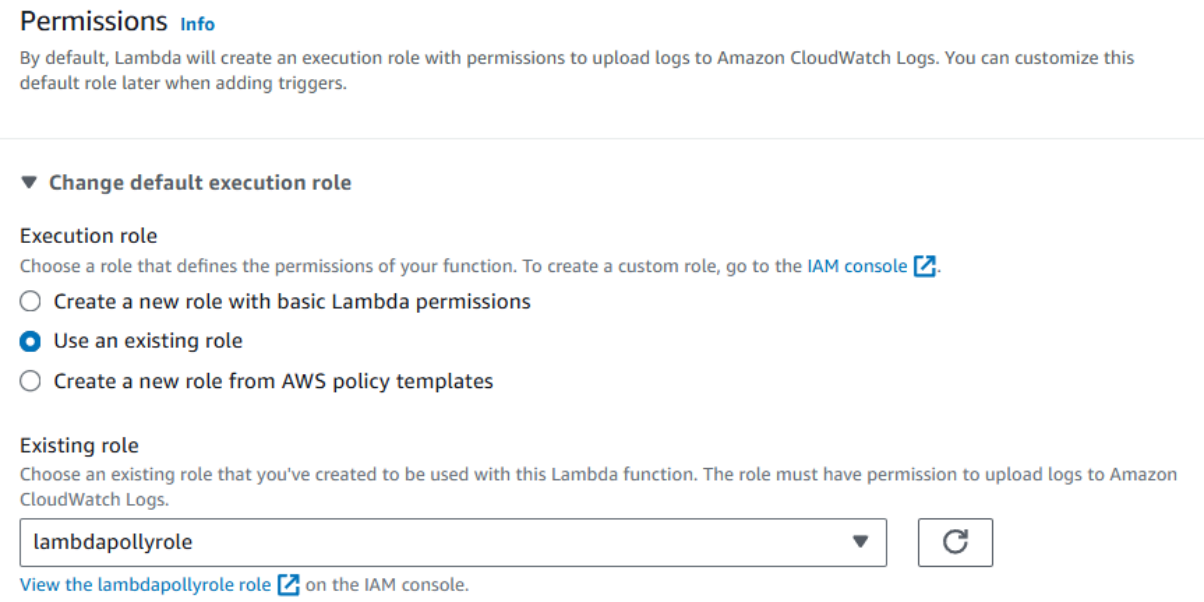
😊 Lab 4 (Python) - Develop Solutions Using AWS Lambda

1. Download the zip file for lab4, unzip it, and open it in VS Code.
2. Now go to AWS Console and open Lambda then click on create function.
3. Then give your function the same name and choose the runtime as python 3.11.



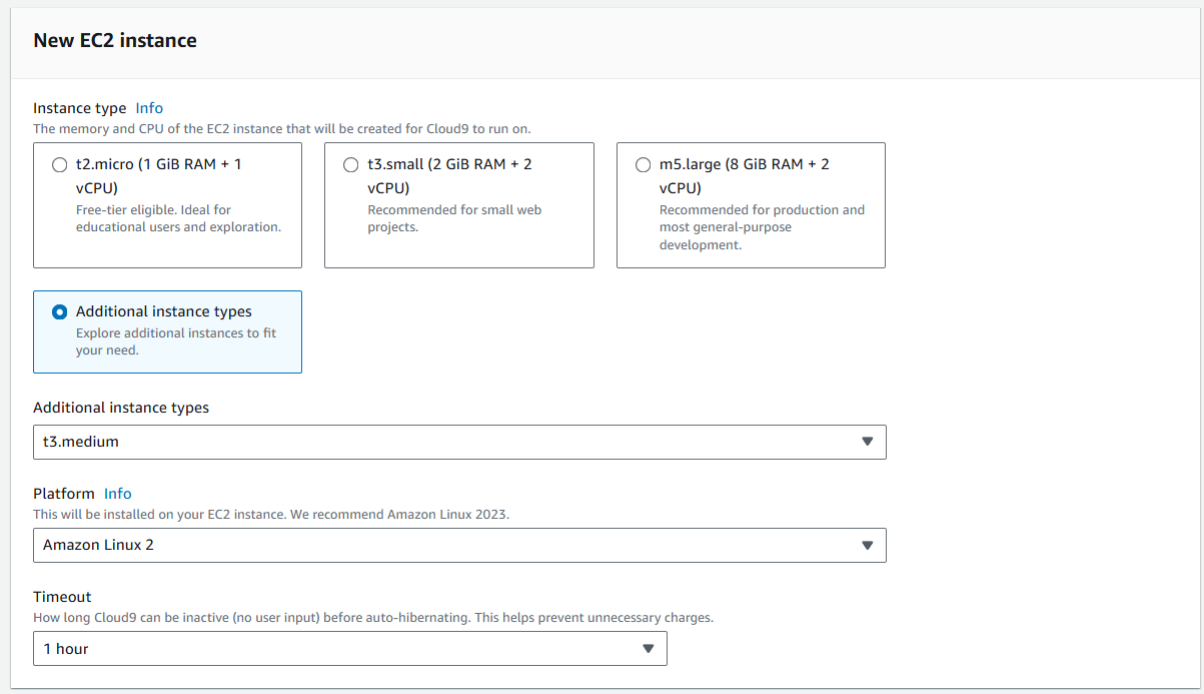
4. After that choose an existing role which is lambda polly role. In case if you didn't have this role created. Then you can copy the JSON code below and create this role.



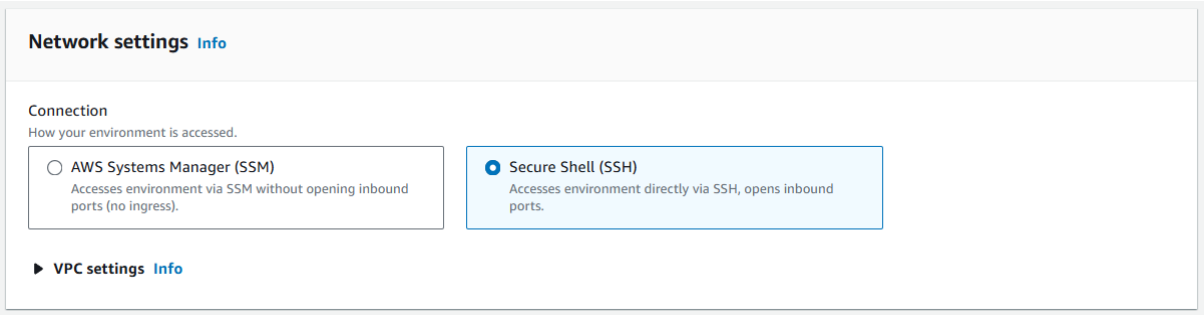

```
{
"Version": "2012-10-17",
"Statement": [
```

```
        {
            "Action": [
                "dynamodb:DeleteItem",
                "dynamodb:PutItem",
                "dynamodb:GetItem",
                "dynamodb:Query",
                "dynamodb:Scan",
                "dynamodb:UpdateItem",
                "dynamodb:DescribeTable",
                "polly:SynthesizeSpeech",
                "s3:PutObject",
                "s3:GetObject",
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "lambda:TagResource"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

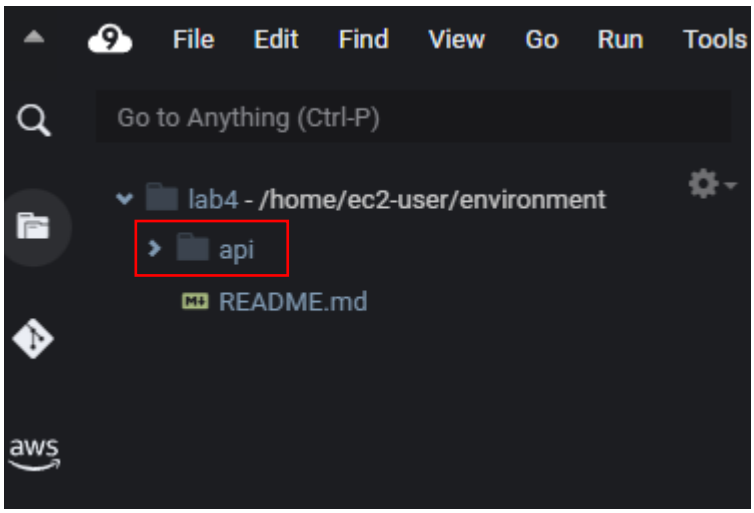
5. After creating the function in your Amazon console search for Cloud9 and create an environment.
6. For EC2 instance choose t3.medium as instance type and the platform should be Amazon Linux 2.



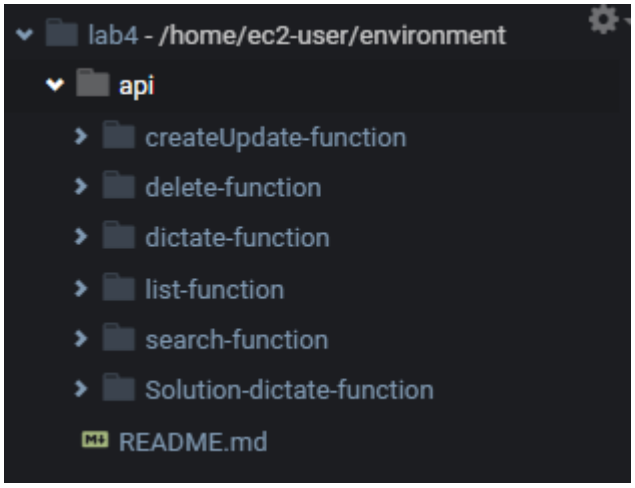
7. In the network settings choose SSH and create your environment.



8. Once your environment is ready open the IDE and there you need to create a folder with the name api.
9. Then in this api folder you need to drag and drop all the files from lab4 folder.



10. It would look like this with all files pasted inside the api folder.



11. Now you are going to run some commands in the cloud9 IDE terminal below you can see the commands run one by one. Also, you will find a text file for commands.

    $apiBucket='YOURBUCKETNAME'

    $notesTable='Notes'

    aws lambda update-function-configuration --function-name dictate-function --environment Variables="{MP3_BUCKET_NAME=$apiBucket, TABLE_NAME=$notesTable}"

12. Now run the below commands to create a zip file of dictate function code so that we can publish this to our lambda function.

    cd ~/environment/api/dictate-function

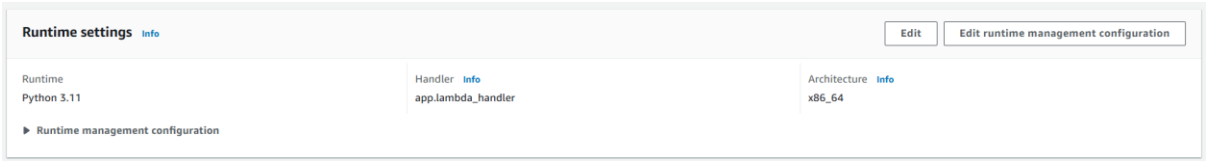
    zip dictate-function.zip app.py

13. Then run the command below to upload the code to your lambda function. If you go to the lambda function you will see that the code has been uploaded.

    **aws lambda update-function-code **
    **--function-name dictate-function **
    --zip-file fileb://dictate-function.zip

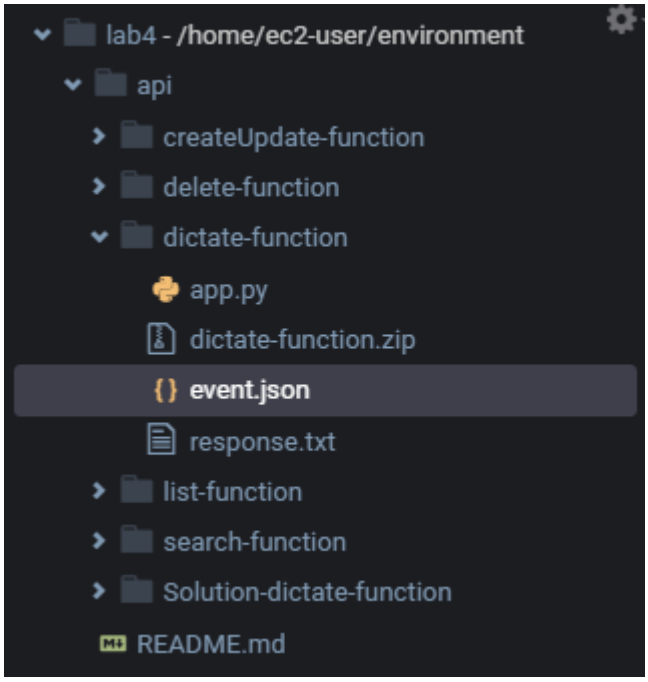
14. Run this command to update the handler of your lambda function.

    **aws lambda update-function-configuration **
    **--function-name dictate-function **
    --handler app.lambda_handler



15. Now, in cloud9, right-click on the dictate function and choose to create a new file with the name event.json, paste the script below, and then save it.

    {
    "UserId": "newbie",
    "NoteId": "2",
    "VoiceId": "Joey"
    }



16. Then you need to run this command to invoke your lambda function.

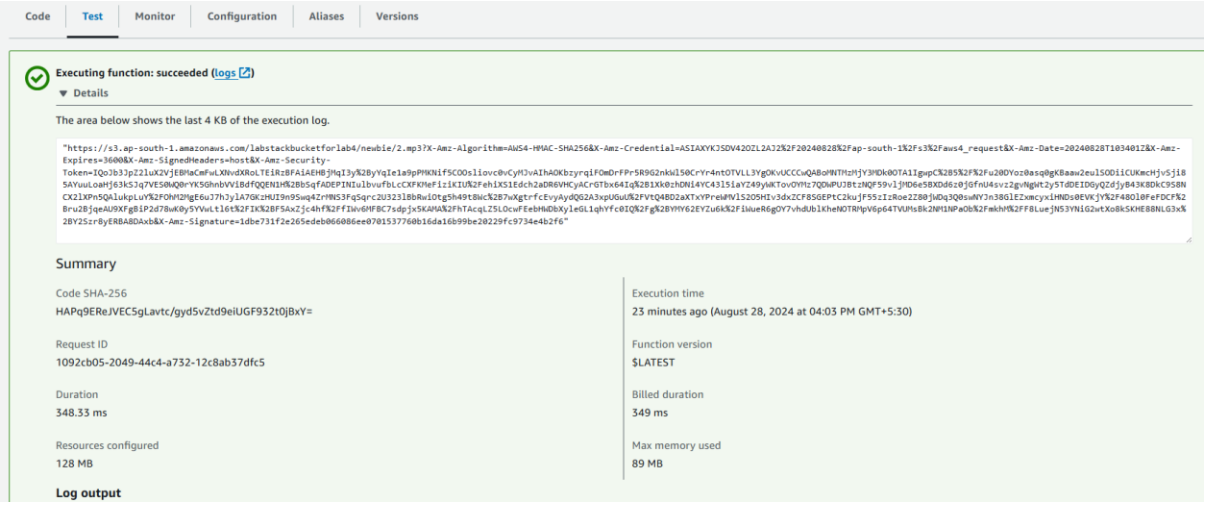
    **aws lambda invoke **
    **--function-name dictate-function **
    --payload fileb://event.json response.txt

```
ec2-user:~/environment/api/dictate-function (main) $ cd ~/environment/api/dictate-function
ec2-user:~/environment/api/dictate-function (main) $ aws lambda invoke \
> --function-name dictate-function \
> --payload fileb://event.json response.txt

{
    "StatusCode": 200,
    "ExecutedVersion": "$LATEST"
}
ec2-user:~/environment/api/dictate-function (main) $
ec2-user:~/environment/api/dictate-function (main) $
```

17. Now come to your lambda function, go to test and create a new test and paste the below statements in it. Then click on the test.
18. If you get the execution result like you are seeing below then it means that your have completed the lab. Just copy this link and paste it into a new tab. A file will be downloaded.

    {
    "UserId": "newbie",
    "NoteId": "2",
    "VoiceId": "Joey"
    }



19. Now there are some optional tasks to create other remaining functions.
    createUpdate-function
    search-function
    delete-function
    list-function

20. So, I'll be creating one function and similarly you can create other functions.
21. There are some commands which you need to run in your cloud9 IDE if you want to run create those function.
22. Below you can see the commands to create a lambda function. Now you need to execute them one by one. Just remember to change the role name with your given name for your role.
23. Also, when you will be creating other function you just need to change the folder name command rest of all the commands are the same.

    roleArn=$(aws iam list-roles --output text --query 'Roles[?contains(RoleName, `lambdapollyrole`)== `true`].Arn')

    folderName=createUpdate-function /////// folderName=delete-function /////// folderName=search-function /////// folderName=list-function /////// folderName=delete-function

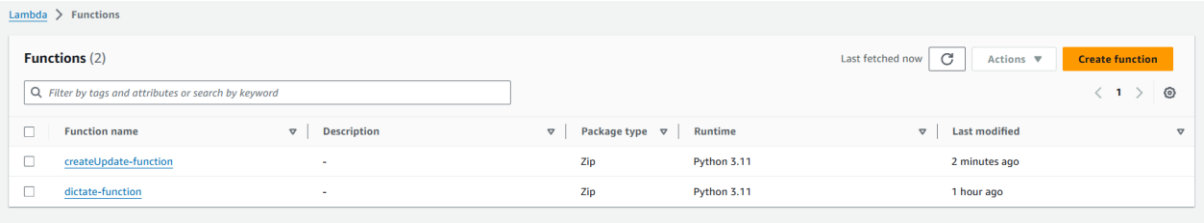
cd ~/environment/api/$folderName

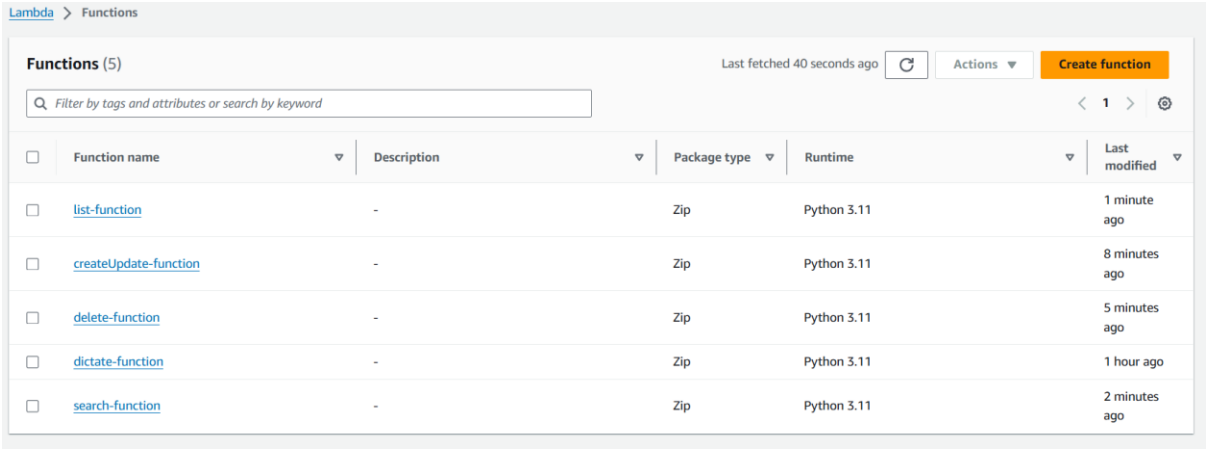
zip $folderName.zip app.py

**aws lambda create-function **
**--function-name $folderName **
**--handler app.lambda_handler **
**--runtime python3.11 **
**--role $roleArn **
**--environment Variables={TABLE_NAME=$notesTable} **
--zip-file fileb://$folderName.zip

```
bash - "ip-172-31-26-176.a ×   Immediate
ec2-user:~/environment $ roleArn=$(aws iam list-roles --output text --query 'Roles[?contains(RoleName, `lambdapollyrole`) == `true`].Arn')
ec2-user:~/environment $ folderName=createUpdate-function
ec2-user:~/environment $ cd ~/environment/api/$folderName
ec2-user:~/environment/api/createUpdate-function (main) $ zip $folderName.zip app.py
updating: app.py (deflated 54%)
ec2-user:~/environment/api/createUpdate-function (main) $ aws lambda create-function \
> --function-name $folderName  \
> --handler app.lambda_handler \
> --runtime python3.11 \
> --role $roleArn \
> --environment Variables={TABLE_NAME=$notesTable} \
> --zip-file fileb://$folderName.zip
{
    "FunctionName": "createUpdate-function",
    "FunctionArn": "arn:aws:lambda:ap-south-1:533267094905:function:createUpdate-function",
    "Runtime": "python3.11",
    "Role": "arn:aws:iam::533267094905:role/lambdapollyrole",
    "Handler": "app.lambda_handler",
    "CodeSize": 731,
    "Description": "",
    "Timeout": 3,
    "MemorySize": 128,
    "LastModified": "2024-08-28T11:16:20.449+0000",
    "CodeSha256": "TgGbOzYdhebiO2FwwJs69xrYREnKirLoU8tIK2St47A=",
    "Version": "$LATEST",
    "Environment": {
        "Variables": {
            "TABLE_NAME": "Notes"
```

24. Likewise, you can create the other lambda functions. Just repeat the steps as you can see above.




Lambda
Functions
Functions (5)
Last fetched 40 seconds ago
Actions
Create function
Filter by tags and attributes or search by keyword
1
Function name
Description
Package type
Runtime
Last modified
list-function
-
Zip
Python 3.11
1 minute ago
createUpdate-function
-
Zip
Python 3.11
8 minutes ago
delete-function
-
Zip
Python 3.11
5 minutes ago
dictate-function
-
Zip
Python 3.11
1 hour ago
search-function
-
Zip
Python 3.11
2 minutes ago