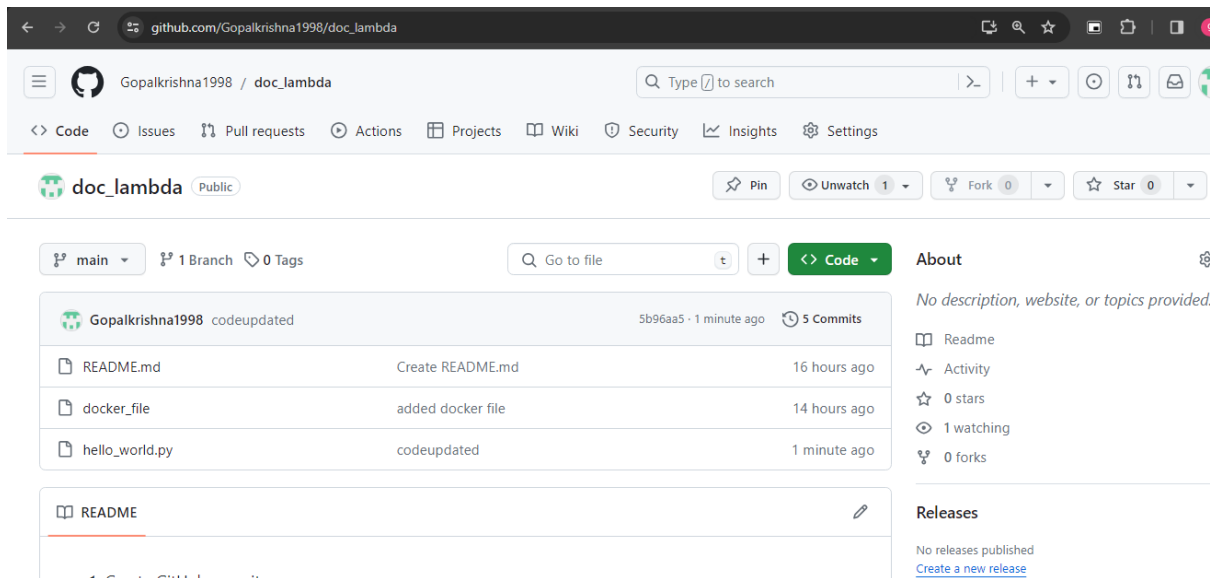


Name : - Gopalkrishna Umak

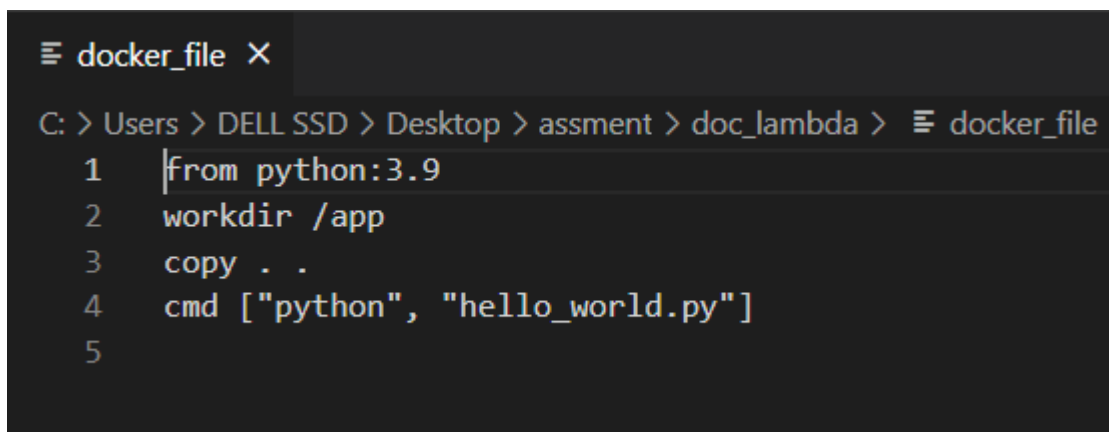
Email id :- [gopalumak4@gmail.com](mailto:gopalumak4@gmail.com) /9145246424

Given task:-

1. Create GitHub repository
2. Add docker file code to create image of sample hello world program and deploy this image to AWS ECR (Use python for hello world program and create docker image using that file )
3. Create lambda function using this image and test it.



Git repository created successfully



Creation of docker file

```
PS C:\Users\DELL SSD\Desktop\assment\doc_lambda> docker build . -t image1 -f docker_file
[+] Building 2.7s (8/8) FINISHED
=> [internal] load .dockerignore                                docker:default 0.1s
=> => transferring context: 2B                                  0.0s
=> [internal] load build definition from docker_file            0.1s
=> => transferring dockerfile: 113B                             0.0s
=> [internal] load metadata for docker.io/library/python:3.9   2.1s
=> [1/3] FROM docker.io/library/python:3.9@sha256:30678bb79d9eeaf98ec0ce83cdcd4d6f5301484ef86873a711e69df2ca77e8ac 0.0s
=> [internal] load build context                                0.1s
=> => transferring context: 3.26kB                               0.1s
=> CACHED [2/3] WORKDIR /app                                   0.0s
=> CACHED [3/3] COPY . .                                        0.0s
=> exporting to image                                           0.0s
=> => exporting layers                                           0.0s
=> => writing image sha256:7e7cbe4343f04f37692b360772e6164929a41cd27eba4fb75ebfe20a6451ad48 0.0s
=> => naming to docker.io/library/image1                        0.0s

View build details: docker-desktop://dashboard/build/default/default/62f5wnxtsnsf1mconn62vhkom

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview
PS C:\Users\DELL SSD\Desktop\assment\doc_lambda>
```

## Creation of docker image

<input type="checkbox"/>	Name	Tag	Status	Created	Size	Actions
<input type="checkbox"/>	image1 7e7cbe4343f0	latest	Unused	12 hours ago	996.79 MB	

## Successfully created docker image

Created public repository  
doc\_lam has been successfully created in public registry

[Amazon ECR](#) > [Public Registry](#) > Repositories

Public repositories

Repositories (1)

[View push commands](#)

[Delete](#)

[Actions](#)

Create repository

	Repository name	URI	Created at
<input type="radio"/>	doc_lam	public.ecr.aws/o0u1d2c0/doc_lam	December 22, 2023, 15:24:42 (UTC+05.5)

## ECR Repository created

arch

×

### Push commands for doc\_lam

Make sure that you have the latest version of the AWS CLI and Docker installed. For more information, see [Getting Started with Amazon ECR](#).

Use the following steps to authenticate and push an image to your repository. For additional registry authentication methods, including the Amazon ECR credential helper, see [Registry Authentication](#).

1. Retrieve an authentication token and authenticate your Docker client to your registry.  
Use the AWS CLI:  

```
aws ecr-public get-login-password --region us-east-1 | docker login --username AWS --password-stdin public.ecr.aws/o0u1d2c0
```

Note: If you receive an error using the AWS CLI, make sure that you have the latest version of the AWS CLI and Docker installed.
2. Build your Docker image using the following command. For information on building a Docker file from scratch see the instructions [here](#). You can skip this step if your image is already built:  

```
docker build -t doc_lam .
```
3. After the build completes, tag your image so you can push the image to this repository:  

```
docker tag doc_lam:latest public.ecr.aws/o0u1d2c0/doc_lam:latest
```
4. Run the following command to push this image to your newly created AWS repository:  

```
docker push public.ecr.aws/o0u1d2c0/doc_lam:latest
```

Close

Pushing docker image to ECR container

IAM > Users

Users (1) Info

Refresh

Delete

Create user

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

Search

<input type="checkbox"/>	User name	Path	Group:	Last activity	MFA	Password age
<input type="checkbox"/>	<a href="#">ecruser</a>	/	0	13 hours ago	-	-

Iam user created successfully

```
PS C:\Users\DELL SSD\Desktop\assment\doc_lambda> aws configure
AWS Access Key ID [*****YUAV]: AKIA3H272KYRTWPIYUAV
AWS Secret Access Key [*****wB6l]: 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
Default region name [us-east-1]: eu-north-1
Default output format [json]: json
PS C:\Users\DELL SSD\Desktop\assment\doc_lambda>
```

Cli connected with iam user

```
C:\Users\DELL SSD>aws ecr get-login-password --region eu-north-1 | docker login --username AWS --password-stdin 772758197795.dkr.ecr.eu-north-1.amazonaws.com
Login Succeeded

C:\Users\DELL SSD>docker tag doc_lam:latest 772758197795.dkr.ecr.eu-north-1.amazonaws.com/doc_lam:latest
Error response from daemon: No such image: doc_lam:latest

C:\Users\DELL SSD>docker tag image1:latest 772758197795.dkr.ecr.eu-north-1.amazonaws.com/doc_lam:latest

C:\Users\DELL SSD>docker push 772758197795.dkr.ecr.eu-north-1.amazonaws.com/doc_lam:latest
The push refers to repository [772758197795.dkr.ecr.eu-north-1.amazonaws.com/doc_lam]
6ba842d143bb: Layer already exists
f056594c46ae: Layer already exists
6b453b473b14: Layer already exists
90c3fd7d657b: Layer already exists
6fb1aacdf632: Layer already exists
a0814d1f5387: Layer already exists
ac7146fb6cf5: Layer already exists
209de9f22f2f: Layer already exists
777ac9f3cbb2: Layer already exists
ae134c61b154: Layer already exists
latest: digest: sha256:4a3fa437dc0ca197fbff0ad573d0210be98c3d6cb6dd2c7baec66bdc18cf3c44 size: 2422

C:\Users\DELL SSD>
```

Docker image pushed successfully on ecr user

[Lambda](#) > [Functions](#) > Create function

## Create function [Info](#)

Choose one of the following options to create your function.

☐ Author from scratch  
Start with a simple Hello World example.

☐ Use a blueprint  
Build a Lambda application from sample code and configuration presets for common use cases.

☒ Container image  
Select a container image to deploy for your function.

☐ Browse serverless app repository  
Deploy a sample Lambda application from the AWS Serverless Application Repository.

---

### Basic information

**Function name**  
Enter a name that describes the purpose of your function.

Use only letters, numbers, hyphens, or underscores with no spaces.

**Container image URI** [Info](#)  
The location of the container image to use for your function.

Requires a valid Amazon ECR image URI.

[Browse images](#)

Creation of function on lambda

Successfully created the function **lambda\_fun**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

[Lambda](#) > [Functions](#) > lambda\_fun

## lambda\_fun

[Throttle](#)
[Copy ARN](#)
[Actions](#)

[Function overview](#) [Info](#)
[Export to Application Composer](#)
[Download function](#)

[Diagram](#)
[Template](#)

lambda\_fun

[+ Add trigger](#)
[+ Add destination](#)

**Description**  
-

**Last modified**  
20 seconds ago

**Function ARN**  
arn:aws:lambda:eu-north-1:772758197795:function:lambda\_fun

**Function URL** [Info](#)  
-

Successfully created lambda function

Test event

Info

Save

Test

To invoke your function without saving an event, configure the JSON event, then choose Test.

Test event action

☒ Create new event

☐ Edit saved event

Event name

test1

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

Event sharing settings

☒ Private

☐ Shareable

This event is only available in the Lambda console and to the event creator. You can configure a total of 10. [Learn more](#)

This event is available to IAM users within the same account who have permissions to access and use shareable events. [Learn more](#)

Template - optional

hello-world

## Test event of lambda function

The section below shows the logging calls in your code. [Click here](#) to view the corresponding CloudWatch log group.

```
Hello World!  
INIT_REPORT Init Duration: 242.92 ms    Phase: init    Status: error    Error Type: Runtime.ExitError  
Hello World!  
INIT_REPORT Init Duration: 761.12 ms    Phase: invoke    Status: error    Error Type: Runtime.ExitError
```

Hello world printed using lambda function