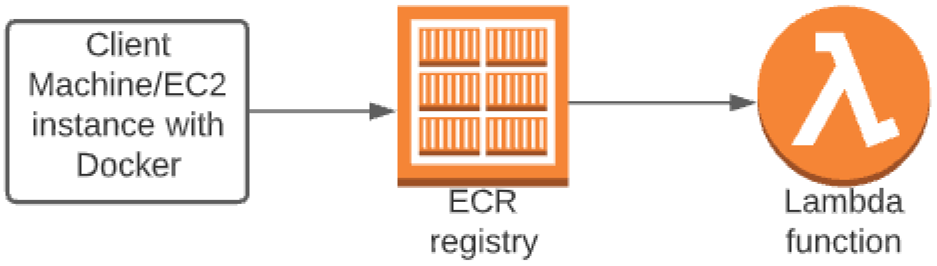
Declaration	
Questions in this exercise are intentionally complex and could be convoluted or confusing. This is by design and to simulate real life situations where customers seldom give crystal clear requirements and ask unambiguous questions.	
I have read the above statement and agree to these conditions	
I AGREE	Alka Sinha
	<Enter your name above this line to indicate that you are in agreement>

Instructions
Every screenshot requested in this workbook is compulsory and carries 1 point
Your AWS account ID must be clearly visible in every screenshot using the AWS console; missing id or using someone else's id is not permitted. Such cases will be considered as plagiarism and severe penalty will be imposed.
All screenshots must be in the order mentioned under "Expected Screenshots" for every step
DO NOT WAIT UNTIL THE LAST MINUTE. The program office will not extend the project submission deadline under any circumstances.
The file should be renamed in the format BATCH_FIRSTNAME_LASTNAME_PROJECT1. For example: PGPCCMAY18_VIJAY_DWIVEDI_PROJECT1.pdf

Resource Clean Up
Cloud is always pay per use model and all resources/services that we consume are chargeable. Cleaning up when you've completed your lab or project is always necessary. This is true whether you're doing a lab or implementing a project at your workplace.
After completing the lab, make sure to delete each resource created in reverse chronological order.

Scenario

The introduction of Lambda support for OCI container images provides customers with more choices when it comes to packaging formats. Developers can now choose to take advantage of the event-driven runtime model and cost-savings advantages of AWS Lambda, while taking advantage of the predictability and control offered by a container-based development and deployment cycle.
Architecture diagram



Architecture Implementation	
1	Download the Dockerfile and the app code folder provided with this workbook
2	Package the web application as a Docker image running on Alpine with Python
3	Create an ECR repository and login to it.
4	Build the image with the downloaded dockerfile and the support files
5	Tag the image appropriately and push it into the ECR repository.
6	Create a Lambda function with the image in ECR.

Step 1 : Docker Image creation

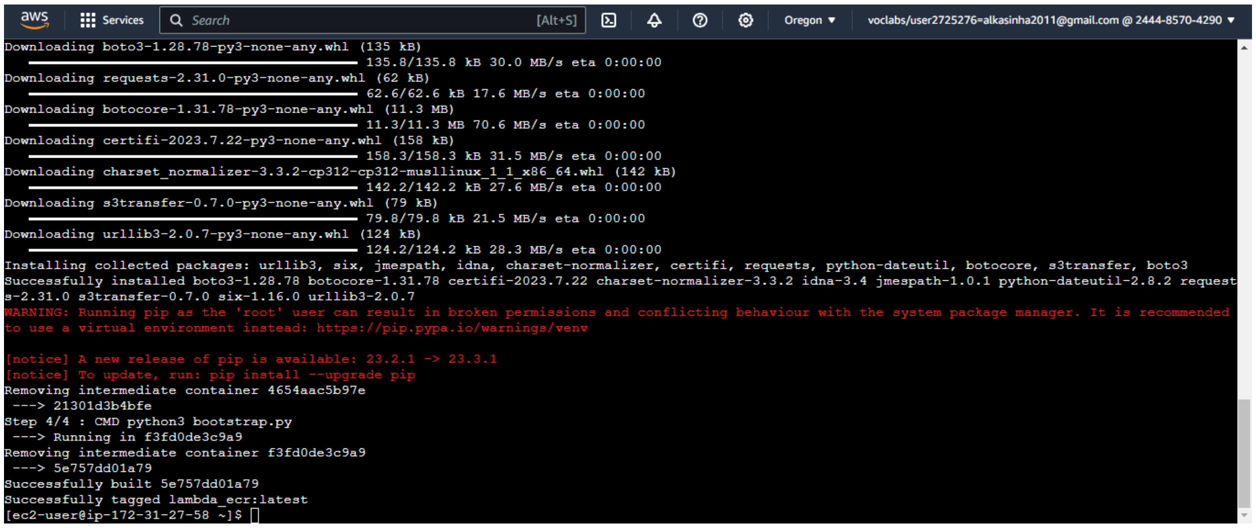
Step number	a
Step name	Creation of Docker image
Instructions	**1)** Create an EC2 instance using the Amazon Linux 2 AMI in the default VPC. 2) Attach the role “LabInstanceProfile” to the instance created above 3) Download the file OCI.zip provided with this workbook and copy it to the EC2 instance using the scp command scp -i <pem file name> ./OCI.zip ec2-user@<public IP of instance>:/home/ec2-user (Ensure that the file OCI.zip and the pem file are in the same folder before running this command)

4) Login to the instance using SSH and run the following commands to set up the environment
   sudo yum update
   sudo yum install unzip
   sudo unzip OCI.zip
   sudo amazon-linux-extras install docker
   sudo service docker start
   sudo usermod -a -G docker ec2-user
   (At this point, log out of the instance and log in again to ensure that the above command works. Then continue with the rest of the commands)
   sudo yum install awscli -y
   aws configure
   Skip the access key and secret access key fields by pressing the Enter key. Enter the region as
   us-east-1 and format as json
5) Run the below command to create the Docker image
   docker build -t lambda_ecr .

6) Run the below command to verify the creation of the image
   docker images

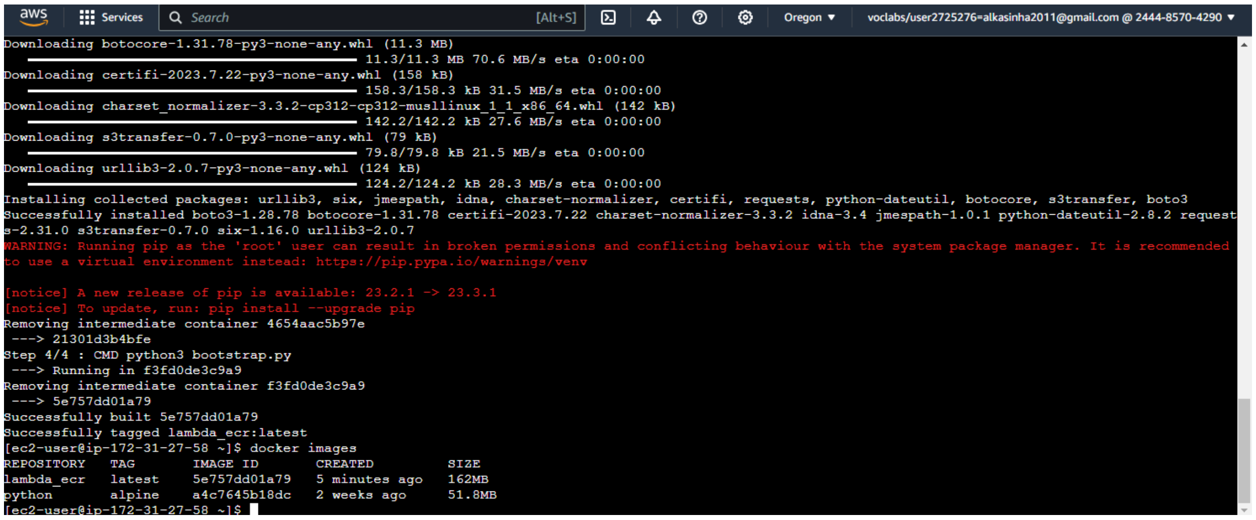
Expected screenshots 1)Building the Docker image 3) List of the created image

Building Docker Image


aws
Services
Search
[Alt+S]
Oregon ▾
voclabs/user2725276=alkasinha2011@gmail.com @ 2444-8570-4290 ▾

```
Downloading boto3-1.28.78-py3-none-any.whl (135 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 135.8/135.8 kB 30.0 MB/s eta 0:00:00
Downloading requests-2.31.0-py3-none-any.whl (62 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 62.6/62.6 kB 17.6 MB/s eta 0:00:00
Downloading botocore-1.31.78-py3-none-any.whl (11.3 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 11.3/11.3 MB 70.6 MB/s eta 0:00:00
Downloading certifi-2023.7.22-py3-none-any.whl (158 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 158.3/158.3 kB 31.5 MB/s eta 0:00:00
Downloading charset_normalizer-3.3.2-cp312-cp312-musllinux_1_1_x86_64.whl (142 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 142.2/142.2 kB 27.6 MB/s eta 0:00:00
Downloading s3transfer-0.7.0-py3-none-any.whl (79 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 79.8/79.8 kB 21.5 MB/s eta 0:00:00
Downloading urllib3-2.0.7-py3-none-any.whl (124 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 124.2/124.2 kB 28.3 MB/s eta 0:00:00
Installing collected packages: urllib3, six, jmespath, idna, charset-normalizer, certifi, requests, python-dateutil, botocore, s3transfer, boto3
Successfully installed boto3-1.28.78 botocore-1.31.78 certifi-2023.7.22 charset-normalizer-3.3.2 idna-3.4 jmespath-1.0.1 python-dateutil-2.8.2 request
s-2.31.0 s3transfer-0.7.0 six-1.16.0 urllib3-2.0.7
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended
to use a virtual environment instead: https://pip.pypa.io/warnings/venv

[notice] A new release of pip is available: 23.2.1 -> 23.3.1
[notice] To update, run: pip install --upgrade pip
Removing intermediate container 4654aac5b97e
 ---> 21301d3b4bfe
Step 4/4 : CMD python3 bootstrap.py
 ---> Running in f3fd0de3c9a9
Removing intermediate container f3fd0de3c9a9
 ---> 5e757dd01a79
Successfully built 5e757dd01a79
Successfully tagged lambda_ecr:latest
[ec2-user@ip-172-31-27-58 ~]$ 
```

List of created image

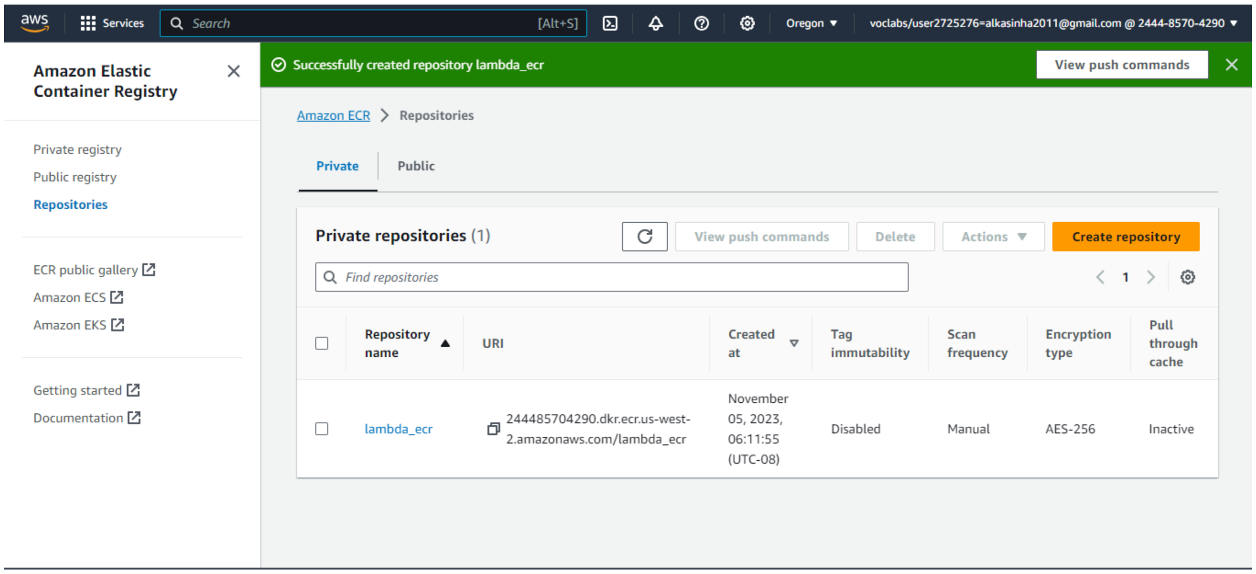

aws
Services
Search
[Alt+S]
Oregon ▼
voclabs/user2725276=alkasinha2011@gmail.com @ 2444-8570-4290 ▼

```
Downloading botocore-1.31.78-py3-none-any.whl (11.3 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 11.3/11.3 MB 70.6 MB/s eta 0:00:00
Downloading certifi-2023.7.22-py3-none-any.whl (158 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 158.3/158.3 kB 31.5 MB/s eta 0:00:00
Downloading charset_normalizer-3.3.2-cp312-cp312-musllinux_1_1_x86_64.whl (142 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 142.2/142.2 kB 27.6 MB/s eta 0:00:00
Downloading s3transfer-0.7.0-py3-none-any.whl (79 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 79.8/79.8 kB 21.5 MB/s eta 0:00:00
Downloading urllib3-2.0.7-py3-none-any.whl (124 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 124.2/124.2 kB 28.3 MB/s eta 0:00:00
Installing collected packages: urllib3, six, jmespath, idna, charset-normalizer, certifi, requests, python-dateutil, botocore, s3transfer, boto3
Successfully installed boto3-1.28.78 botocore-1.31.78 certifi-2023.7.22 charset-normalizer-3.3.2 idna-3.4 jmespath-1.0.1 python-dateutil-2.8.2 request
s-2.31.0 s3transfer-0.7.0 six-1.16.0 urllib3-2.0.7
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended
to use a virtual environment instead: https://pip.pypa.io/warnings/venv

[notice] A new release of pip is available: 23.2.1 -> 23.3.1
[notice] To update, run: pip install --upgrade pip
Removing intermediate container 4654aac5b97e
 ---> 21301d3b4bfe
Step 4/4 : CMD python3 bootstrap.py
 ---> Running in f3fd0de3c9a9
Removing intermediate container f3fd0de3c9a9
 ---> 5e757dd01a79
Successfully built 5e757dd01a79
Successfully tagged lambda_ecr:latest
[ec2-user@ip-172-31-27-58 ~]$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
lambda_ecr    latest    5e757dd01a79   5 minutes ago   162MB
python        alpine    a4c7645b18dc   2 weeks ago     51.8MB
[ec2-user@ip-172-31-27-58 ~]$
```

Step 2: Create ECR repository and upload image to ECR

Step number	a
Step name	Creating the ECR repository
Instructions	1) Go to the ECR service on the AWS console 2) Select the Repositories from the left pane 3) Create a new private repository named **lambda_ecr** with the default settings
Step number	b
Step name	Image upload to ECR
Instructions	1) Once the repository is created, select the repository and then click on "View push commands" on the top right 2) From the pop up screen which appears, run commands 1, 3 and 4 after logging into the EC2 instance created above. Note that command 2 was already executed in the previous step when the image was created. For reference, the commands will be in the format shown below: **aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin <xxxxxxx.dkr.ecr.us-east-1.amazonaws.com>** **docker tag lambda_ecr_image:latest <xxxxxxx.dkr.ecr.us-east-1.amazonaws.com/lambda_ecr>:latest** **docker push <xxxxxxx.dkr.ecr.us-east-1.amazonaws.com/lambda_ecr>:latest**
Expected screenshots	1) Creation of Repository 2) View push commands 3) Login Succeeded 4) Tagging of the image 5) Pushing of image to ECR 6) Image uploaded on the ECR repo

Creation of repository

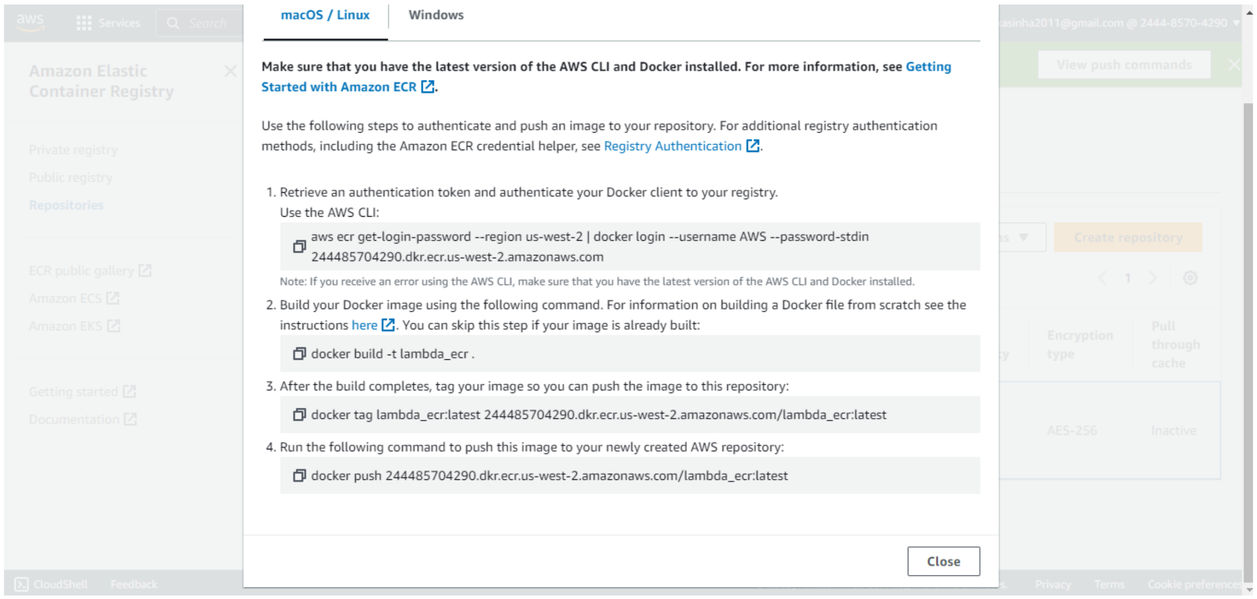


Successfully created repository lambda_ecr

Private repositories (1)

Repository name	URI	Created at	Tag immutability	Scan frequency	Encryption type	Pull through cache
lambda_ecr	244485704290.dkr.ecr.us-west-2.amazonaws.com/lambda_ecr	November 05, 2023, 06:11:55 (UTC-08)	Disabled	Manual	AES-256	Inactive

View push commands

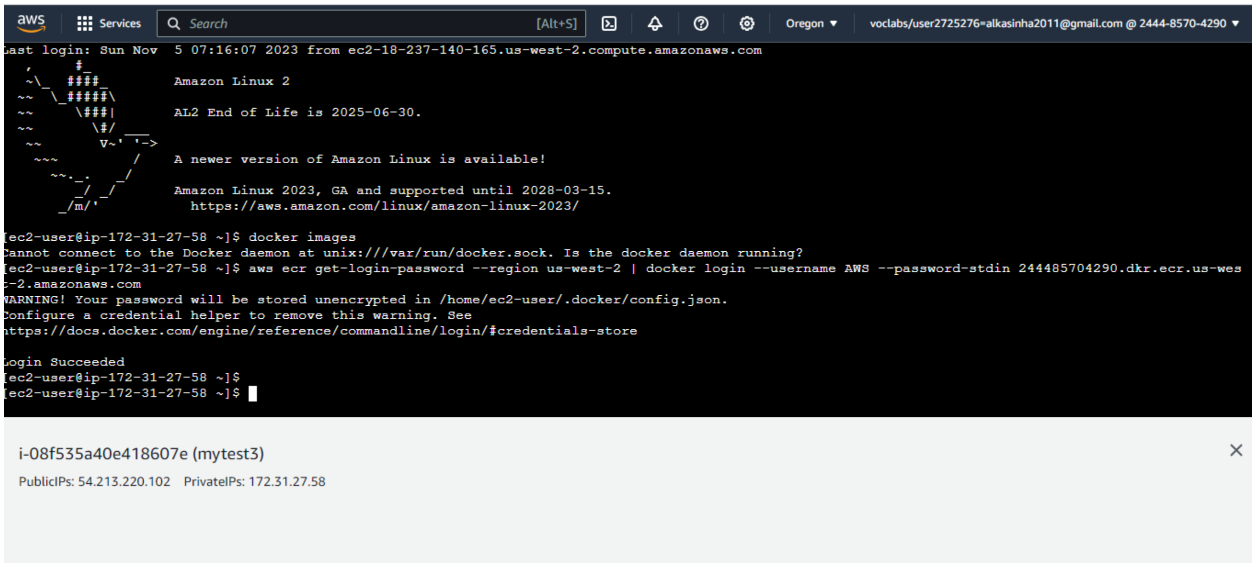


Make sure that you have the latest version of the AWS CLI and Docker installed. For more information, see Getting Started with Amazon ECR.

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 get-login-password --region us-west-2 | docker login --username AWS --password-stdin 244485704290.dkr.ecr.us-west-2.amazonaws.com
   ```
   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 lambda_ecr .
   ```
3. After the build completes, tag your image so you can push the image to this repository:
   ```
   docker tag lambda_ecr:latest 244485704290.dkr.ecr.us-west-2.amazonaws.com/lambda_ecr:latest
   ```
4. Run the following command to push this image to your newly created AWS repository:
   ```
   docker push 244485704290.dkr.ecr.us-west-2.amazonaws.com/lambda_ecr:latest
   ```

Login succeeded

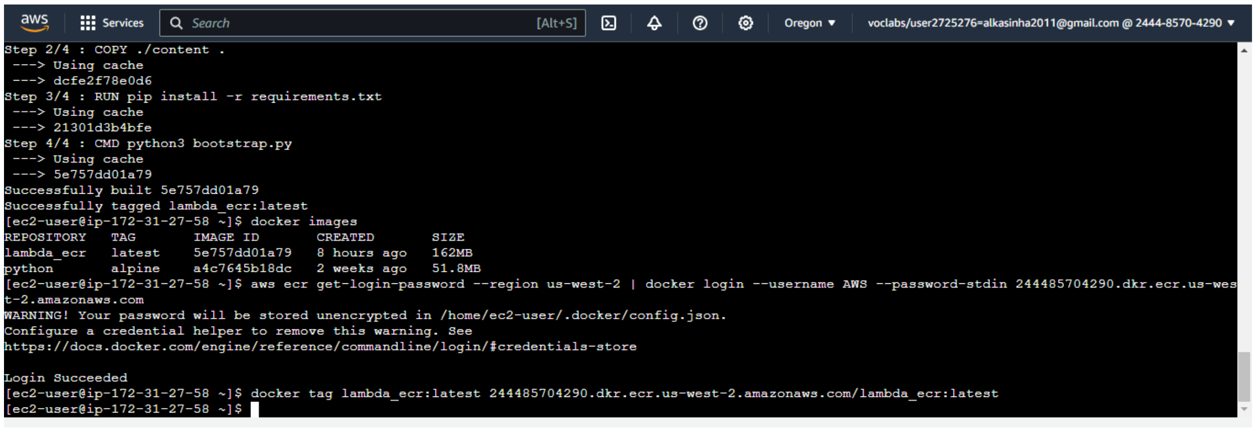

aws
Services
Search
[Alt+S]
Oregon ▼
voclabs/user2725276=alkasinha2011@gmail.com @ 2444-8570-4290 ▼

```
Last login: Sun Nov  5 07:16:07 2023 from ec2-18-237-140-165.us-west-2.compute.amazonaws.com
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2025-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-172-31-27-58 ~]$ docker images
Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?
[ec2-user@ip-172-31-27-58 ~]$ aws ecr get-login-password --region us-west-2 | docker login --username AWS --password-stdin 244485704290.dkr.ecr.us-wes
t-2.amazonaws.com
WARNING! Your password will be stored unencrypted in /home/ec2-user/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[ec2-user@ip-172-31-27-58 ~]$
[ec2-user@ip-172-31-27-58 ~]$
```

i-08f535a40e418607e (mytest3)

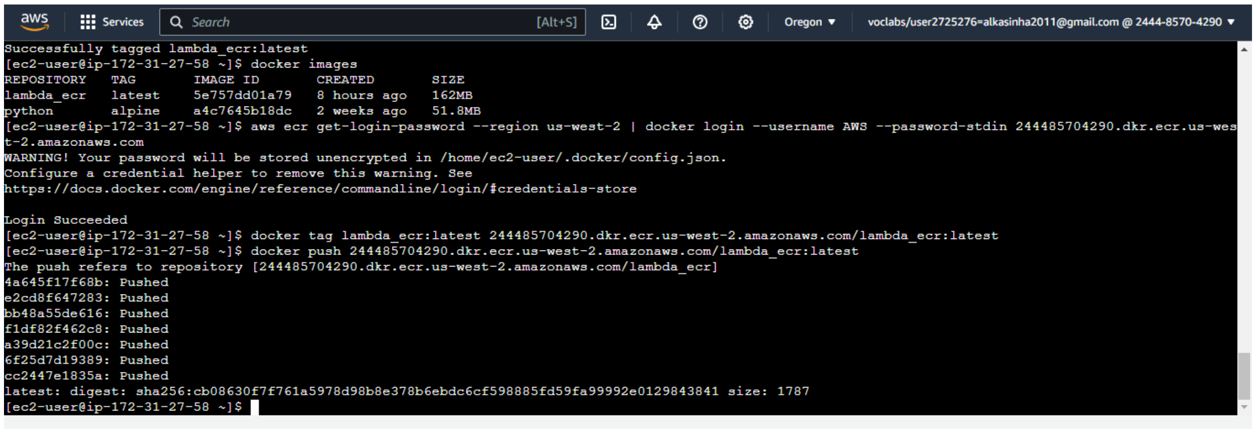
PublicIPs: 54.213.220.102 PrivateIPs: 172.31.27.58

Tagging of Image


aws
Services
Search
[Alt+S]
Oregon ▼
voclabs/user2725276=alkasinha2011@gmail.com @ 2444-8570-4290 ▼

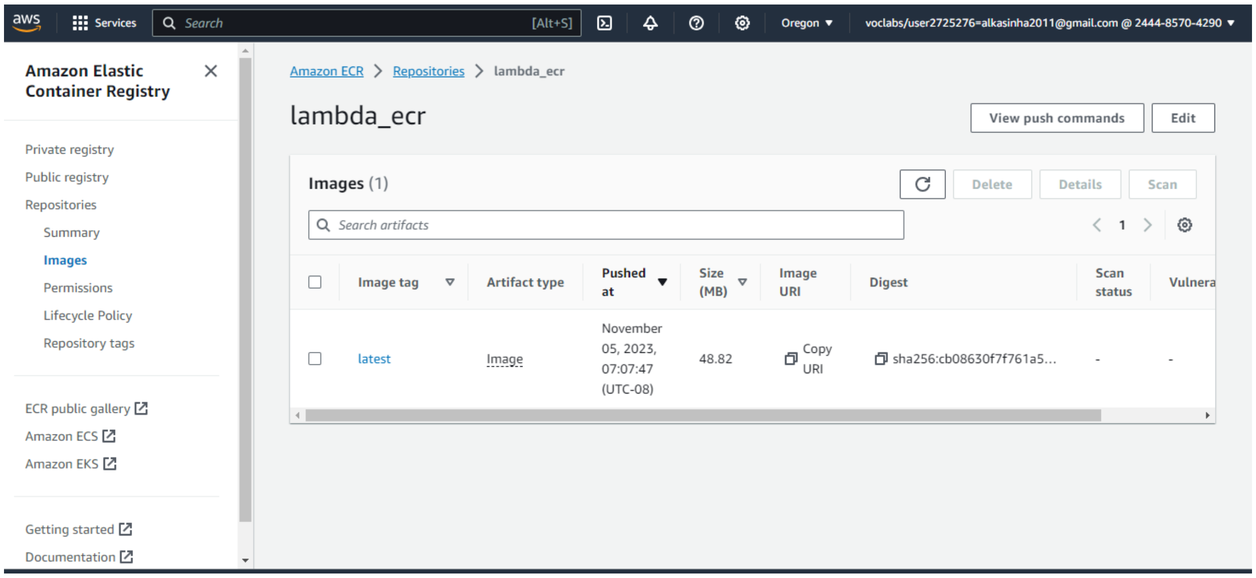
```
Step 2/4 : COPY ./content .
 ---> Using cache
 ---> dcfe2f78e0d6
Step 3/4 : RUN pip install -r requirements.txt
 ---> Using cache
 ---> 21301d3b4bfe
Step 4/4 : CMD python3 bootstrap.py
 ---> Using cache
 ---> 5e757dd01a79
Successfully built 5e757dd01a79
Successfully tagged lambda_ecr:latest
[ec2-user@ip-172-31-27-58 ~]$ docker images
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
lambda_ecr   latest    5e757dd01a79   8 hours ago   162MB
python       alpine    a4c7645b18dc   2 weeks ago   51.8MB
[ec2-user@ip-172-31-27-58 ~]$ aws ecr get-login-password --region us-west-2 | docker login --username AWS --password-stdin 244485704290.dkr.ecr.us-wes
t-2.amazonaws.com
WARNING! Your password will be stored unencrypted in /home/ec2-user/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[ec2-user@ip-172-31-27-58 ~]$ docker tag lambda_ecr:latest 244485704290.dkr.ecr.us-west-2.amazonaws.com/lambda_ecr:latest
[ec2-user@ip-172-31-27-58 ~]$
```

Pushing of images and uploaded to ecr repo


aws
Services
Search
[Alt+S]
Oregon ▼
voclabs/user2725276=alkasinha2011@gmail.com @ 2444-8570-4290 ▼

```
Successfully tagged lambda_ecr:latest
[ec2-user@ip-172-31-27-58 ~]$ docker images
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
lambda_ecr   latest    5e757dd01a79   8 hours ago   162MB
python       alpine    a4c7645b18dc   2 weeks ago   51.8MB
[ec2-user@ip-172-31-27-58 ~]$ aws ecr get-login-password --region us-west-2 | docker login --username AWS --password-stdin 244485704290.dkr.ecr.us-wes
t-2.amazonaws.com
WARNING! Your password will be stored unencrypted in /home/ec2-user/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[ec2-user@ip-172-31-27-58 ~]$ docker tag lambda_ecr:latest 244485704290.dkr.ecr.us-west-2.amazonaws.com/lambda_ecr:latest
[ec2-user@ip-172-31-27-58 ~]$ docker push 244485704290.dkr.ecr.us-west-2.amazonaws.com/lambda_ecr:latest
The push refers to repository [244485704290.dkr.ecr.us-west-2.amazonaws.com/lambda_ecr]
4a645f17f68b: Pushed
e2cd8f647283: Pushed
bb48a55de616: Pushed
f1df82f462c8: Pushed
a39d21c2f00c: Pushed
6f25d7d19389: Pushed
cc2447e1835a: Pushed
latest: digest: sha256:cb08630f7f761a5978d98b8e378b6ebdc6cf598885fd59fa99992e0129843841 size: 1787
[ec2-user@ip-172-31-27-58 ~]$
```

Image uploaded to ECR



lambda_ecr

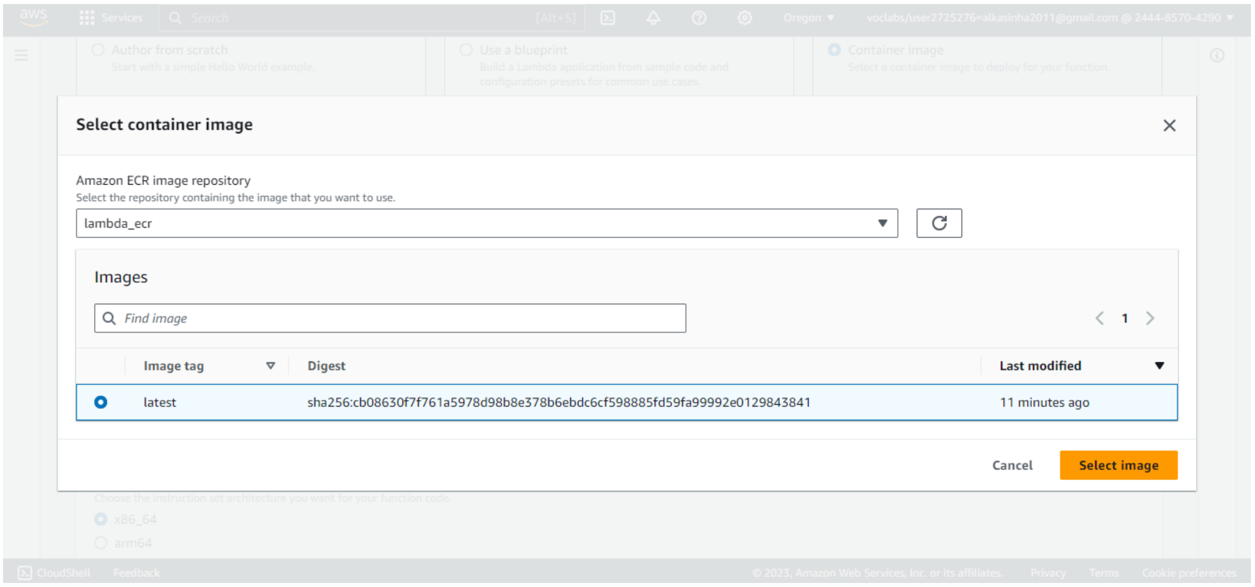
Images (1)

Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest	Scan status	Vulnera
latest	Image	November 05, 2023, 07:07:47 (UTC-08)	48.82	Copy URI	sha256:cb08630f7f761a5...	-	-

Step 3: Creation of Lambda function to test the image

Step number	a
Step name	Create the Lambda function and test the image
Instructions	1) Navigate to the AWS Lambda service using the AWS Console 2) Click on **Create Function** 3) Under Create Function page select the 'Container image' option and enter a function name of your choice 4) For 'Container image URI' Click on "Browse Images" and select the repository and the image 5) Use the existing IAM role – LabRole. 6) Click on Create 7) Wait a few minutes for the function to be created 8) Test the function with the default "Hello World" test to see the result.
Expected screenshots	1) Container image selection 2) Execution role selection 3) Created function 4)Test result of function

Image selection

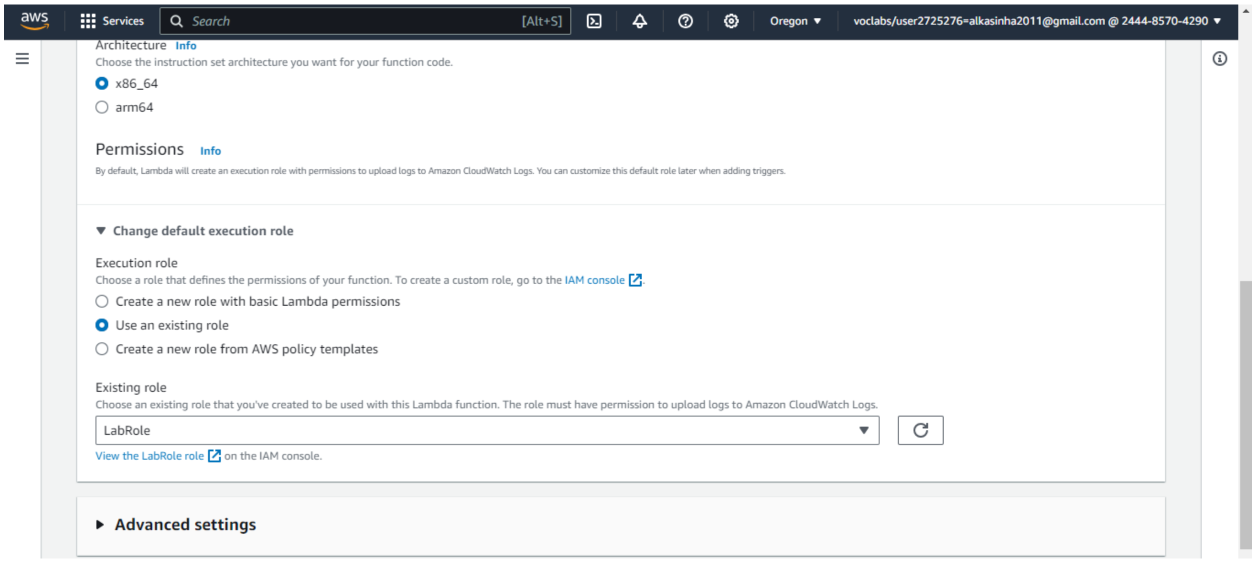

Select container image

Amazon ECR image repository
Select the repository containing the image that you want to use.
lambda_ecr

Images

	Image tag	Digest	Last modified
●	latest	sha256:cb08630f7f761a5978d98b8e378b6ebdc6cf598885fd59fa99992e0129843841	11 minutes ago

Lab role


Architecture Info

Choose the instruction set architecture you want for your function code.

- [✓ x86_64]
- arm64

Permissions Info

By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ Change default execution role

Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.

- Create a new role with basic Lambda permissions
- [✓ Use an existing role]
- Create a new role from AWS policy templates

Existing role

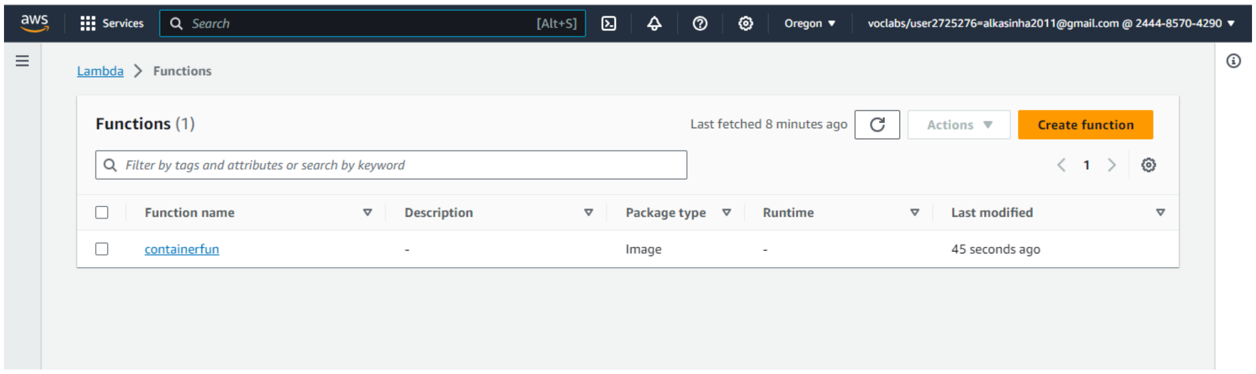
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

LabRole

View the LabRole role on the IAM console.

▶ Advanced settings

Created function

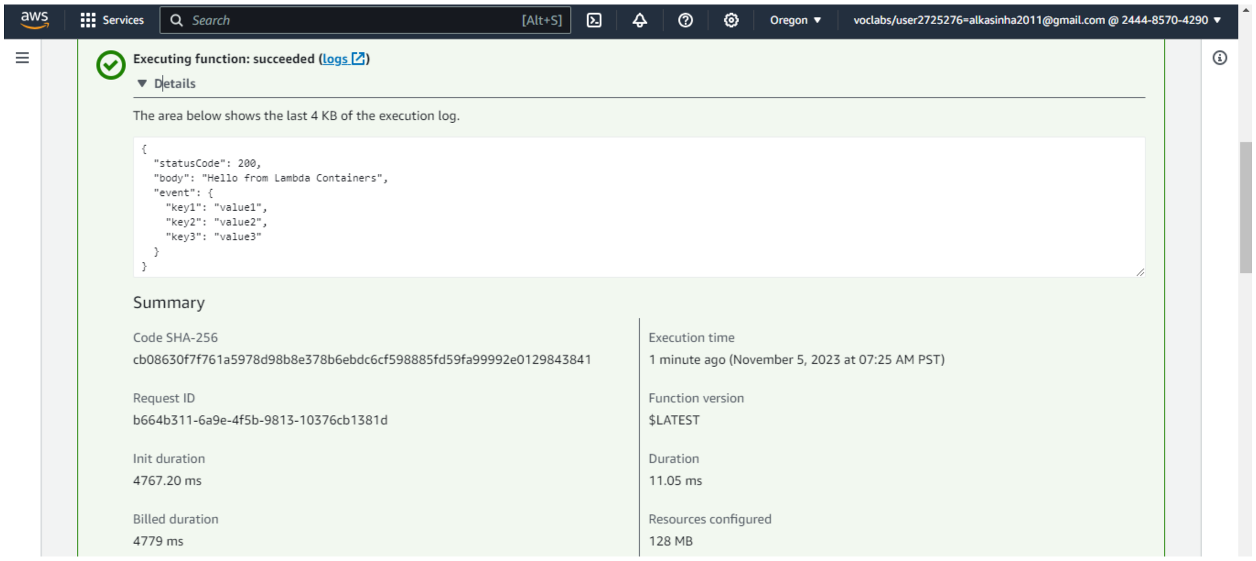

Lambda > Functions

Functions (1)

Last fetched 8 minutes ago

Function name	Description	Package type	Runtime	Last modified
containerfun	-	Image	-	45 seconds ago

Tested function


aws
Services
Search [Alt+S]
Oregon
voclabs/user2725276=alkasinha2011@gmail.com @ 2444-8570-4290

Executing function: succeeded (logs)

Details

The area below shows the last 4 KB of the execution log.

```
{
  "statusCode": 200,
  "body": "Hello from Lambda Containers",
  "event": {
    "key1": "value1",
    "key2": "value2",
    "key3": "value3"
  }
}
```

Summary

Code SHA-256 cb08630f7f761a5978d98b8e378b6ebdc6cf598885fd59fa99992e0129843841	Execution time 1 minute ago (November 5, 2023 at 07:25 AM PST)
Request ID b664b311-6a9e-4f5b-9813-10376cb1381d	Function version $LATEST
Init duration 4767.20 ms	Duration 11.05 ms
Billed duration 4779 ms	Resources configured 128 MB

Answer the following questions

Q1 How long does a container stay in the running state if it is not manually halted?

a) As long as the container's PID 1 is running

b) Has a set timeout after which it pauses

c) Until its container is expunged

d) Docker daemon process scheduler decides on load

Enter your answer here a

Q2 Which of the following best illustrates the relationship between an image and a container?

a) Executable and its hard link

b) Executable and process

c) Parent and child process

d) Many to one

Enter your answer here b

Q3 What is the maximum amount of RAM a container can consume if the memory flag is not used?

a) 8GiB

b) 32GiB

c) None of these

d) As much as the host instance has free

Enter your answer here b

Q4 Which of the following will happen in the same Docker image is pushed to Docker Hub multiple times with different tags

a) Dockerhub will refuse to upload the image

b) The layers in the first image (if unchanged) will be reused in subsequent pushes

c) Dockerhub will merge the images

d) The same image cannot have multiple tags

Enter your answer here c

Q5 Which of the following will run a Docker container in interactive mode?

a) -v

b) -it

c) -b

d) -u

Enter your answer here: b

Q6 How would data persistence be handled in a container environment set up for autoscaling?

Use a shared volume that is mounted on all of the containers in the environment. This shared volume can be used to store data that needs to be persistent across all of the containers in the environment.

Q7 Why is this statement false? "Docker is the only popular choice for microservices deployment".

The claim is untrue because there are numerous ways to install microservices. In a microservices architecture, Docker usage is not required.

Grades distribution	
MCQs	5 (1 point each)
Subjective questions	11 points (5+6)
Implementation screenshots	24 points (2

	point each)
Total	40 points