

Create a Docker container using Dockerfile and store the image in ECR

Lab Steps

Task 1: Sign in to AWS Management Console

1. Click on the **Open Console** button, and you will get redirected to AWS Console in a new browser tab.
2. On the AWS sign-in page,
 - a. Leave the Account ID as default. Never edit/remove the 12-digit Account ID present in the AWS Console. otherwise, you cannot proceed with the lab.
 - b. Now copy your **User Name** and **Password** in the Lab Console to the **IAM Username and Password** in AWS Console and click on the **Sign in** button.
3. Once Signed In to the AWS Management Console, Make the default AWS Region as **US East (N. Virginia) us-east-1**.

Task 2: Create a Key Pair for the EC2 instances, inside the ECS Cluster

1. In the left navigation pane (scroll down) within **Network & Security**, click on the **Key Pairs**
2. To create a new key pair, click on the **Create Key pair** button.

Key pairs [Info](#)

Search

Actions [Create key pair](#)

< 1 > ⚙

Name	Type	Created	Fingerprint	ID
No key pairs to display				

3. Fill in the details below:

- Name: Enter **WhizKeyPair**
- Key pair type: Select **RSA**
- Private key file format: Select **pem (Linux & Mac Users)** or **ppk (Windows users)**
- Leave other options as **default**.
- Click on the **Create Key pair**.

Key pair

A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name

WhizKeyPair

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type [Info](#)

☒ RSA

☐ ED25519

Private key file format

☐ .pem
For use with OpenSSH

☒ .ppk
For use with PuTTY

Tags (Optional)

No tags associated with the resource.

Add tag

You can add 50 more tags.

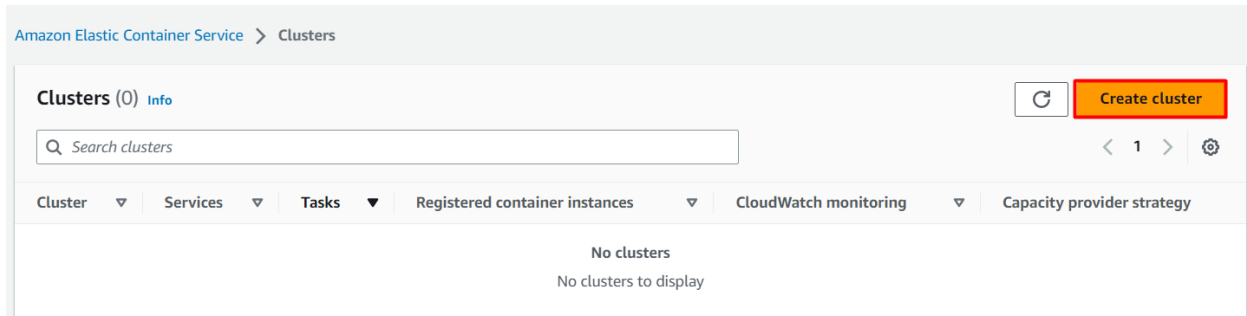
Cancel [Create key pair](#)

4. Key pair will be created.

✓ Successfully created key pair

Task 3: Launching an ECS Cluster

1. Make sure you are in the **N.Virginia** Region. Navigate to **Elastic Container Service** by clicking on the **Services** menu in the top, then click on **Elastic Container Service** in the **Containers** section.
2. On the left sidebar, click on the **Clusters** option present under the **Amazon ECS** section, then Click on the **Create cluster** button.



3. In Cluster configuration :

Cluster name : Enter **whiz**

Cluster configuration

Cluster name

whiz

There can be a maximum of 255 characters. The valid characters are letters (uppercase and lowercase), numbers, hyphens, and underscores.

4. In Infrastructure Section :

UnCheck AWS Fargate and **Check** Amazon EC2 Instances checkbox:

Auto Scaling group (ASG) : Select **Create new ASG**

Operating System/Architecture : Select **Amazon Linux 2**

EC2 instance type : Select **t2.micro**

Desired capacity :

Minimum : Enter **1**

Maximum : Enter **2**

SSH Key pair: Select **WhizKeyPair**

5. In Network settings:

Security group: Select **Create new security group**.

Security name : Enter **mysg**

Security group description : Enter **Security group for instance creation**

Inbound rules for security groups:

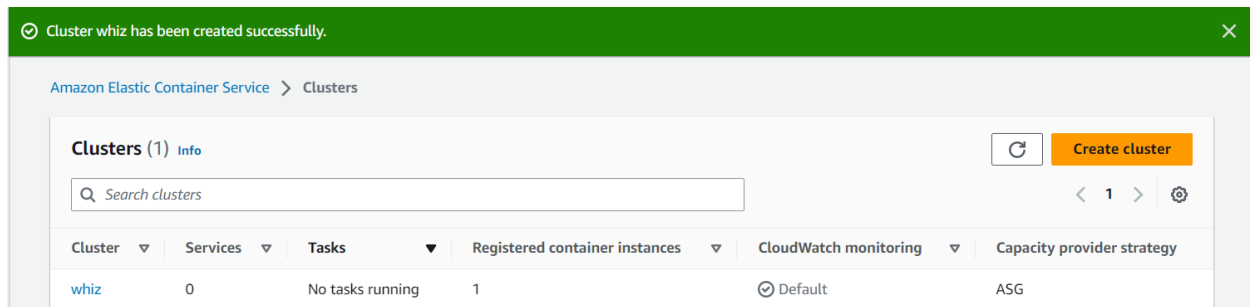
Type : Select **SSH**

Source : Select **Anywhere**

6. Keep rest things as default and click on **Create** button

7. **whiz** ECS Cluster will be created

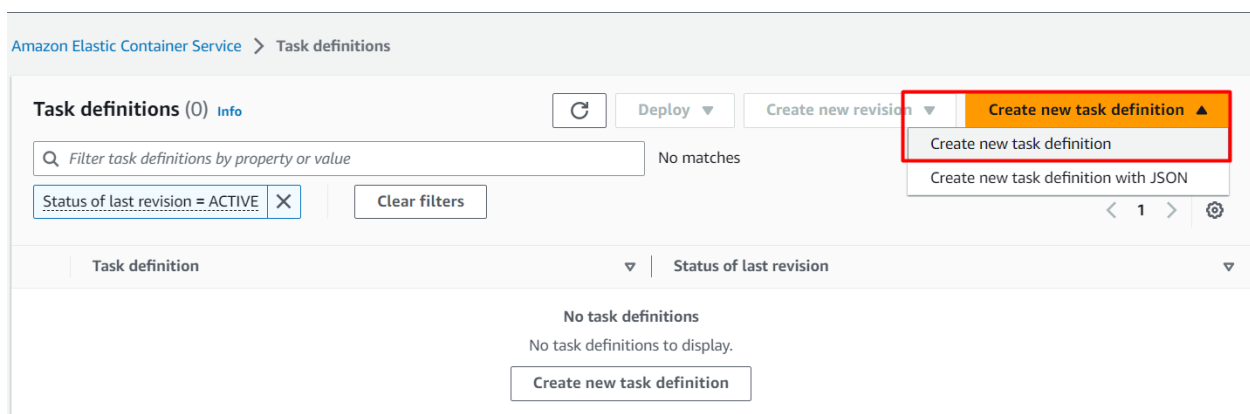
8. It will take few minutes to provision the ECS Instance.



Task 4: Create Task Definitions

In this task, we are going to create a task definition for the cluster. A task definition provides a blueprint for creating tasks, which are the basic unit of work in ECS. Each task definition can define one or more containers that are run together on the same underlying EC2 instances or Fargate tasks.

1. On the left sidebar, click on the **Task Definitions** option present under the **Amazon ECS** section.
2. Click on the **Create new task definition** button.



3. In Task definition configuration:

Task definition family: Enter **ecs-demo**

Task definition configuration

Task definition family [Info](#)

Specify a unique task definition family name.

ecs-demo

Up to 255 letters (uppercase and lowercase), numbers, hyphens, and underscores are allowed.

4. For Environment:

App Environment : **Remove AWS Fargate** And Select **Amazon EC2 Instances**

Task Size :

CPU : Enter **.25 VCPU**

Memory : Enter **.5 GB**

▼ Infrastructure requirements

Specify the infrastructure requirements for the task definition.

Launch type [Info](#)

Selection of the launch type will change task definition parameters.

☐ AWS Fargate

Serverless compute for containers.

☒ Amazon EC2 instances

Self-managed infrastructure using Amazon EC2 instances.

OS, Architecture, Network mode

Network mode is used for tasks and is dependent on the compute type selected.

Operating system/Architecture [Info](#)

Linux/X86_64 ▼

Network mode [Info](#)

awsvpc ▼

Task size [Info](#)

Specify the amount of CPU and memory to reserve for your task.

CPU

.25 vCPU

Memory

.5 GB

For Container Details :

Enter **httpd**

Image URL: Enter **httpd:2.4**

For Port mappings:

Container Port: Enter **80**

Protocol: Select **TCP**

App protocol: Select **HTTP**

Container - 1

Info

Essential container

Remove

Container details

Specify a name, container image, and whether the container should be marked as essential. Each task definition must have at least one essential container.

Name

Image URI

Essential container

httpd

httpd:2.4

Yes

Private registry

Info

Store credentials in Secrets Manager, and then use the credentials to reference images in private registries.

☐ Private registry authentication

Port mappings

Info

Add port mappings to allow the container to access ports on the host to send or receive traffic. Any changes to port mappings configuration impacts the associated service connect settings.

Container port

Protocol

Port name

App protocol

80

TCP

httpd-80-tcp

HTTP

Remove

Add more port mappings

5. **Uncheck** use log collection box.

Log collection

[Info](#)

Configure your task to send container logs to a log

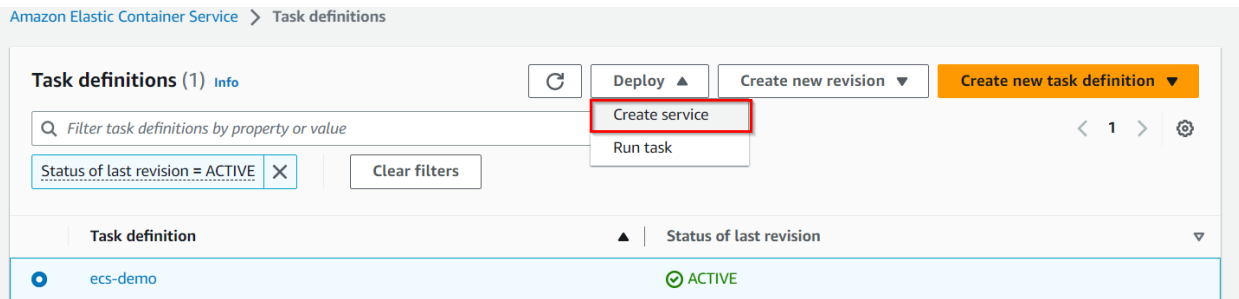
☐ Use log collection

6. Keep rest things as default and click on the **Create** button.
7. Task Definition **ecs-demo** is now created.

Task 5: Create a service and start HTTPD container in ECS

In this task, we are going to create a service and start the HTTPD container. In AWS ECS (Elastic Container Service), a service is a long-running task that ensures that a specified number of instances of a task definition are running and maintained in an ECS cluster. Services allow you to define the desired state of your tasks and automatically handle task placement, scaling, and recovery.

1. On the left sidebar, click on the **Task definitions** option present under the **Amazon ECS** section.
2. Select **ecs-demo** and click on **Deploy** and click **Create service** button.



3. Existing cluster : Select **whiz** cluster
4. Compute options : Select **Launch Type**

Launch type: Select **EC2**

Environment

Amazon EC2

Existing cluster

Select an existing cluster. To create a new cluster, go to [Clusters](#).

whiz

▼ Compute configuration (advanced)

Compute options

[Info](#)

To ensure task distribution across your compute types, use appropriate compute options.

☐ Capacity provider strategy

Specify a launch strategy to distribute your tasks across one or more capacity providers.

☒ Launch type

Launch tasks directly without the use of a capacity provider strategy.

Launch type

[Info](#)

Select either managed capacity (Fargate), or custom capacity (EC2 or user-managed, External instances). External instances are registered to your cluster using the ECS Anywhere capability.

EC2

5. In Deployment configurations :

Service name: Enter **httpd**

Service type: Select **REPLICA**

Desired tasks : Enter **1**

6. Keep other options as default, and click on the **Create** button.

Task 6: Test the HTTPD container in ECS Cluster

1. On the left sidebar, click on the **Clusters** option present under the **Amazon ECS** section.
2. whiz ECS Cluster will be listed here, click on the **whiz**.
3. To view the ECS Instance, switch to the **Infrastructure** tab.
4. Scroll down to the Container Instances

Container instances (1) [Info](#)

Filter container instances by property or value

<input type="checkbox"/>	Container instance ▾	Status ▾	Type ▾	Instance ID 🔗 ▾	Capacit... ▾	Availability zo... ▾	Running tasks... ▾	CF
<input type="checkbox"/>	651321a8d35a4cd...	Active	EC2	i-04ac8bbc136e...	Infra-ECS-...	us-east-1d	1	76

5. **Click** on the EC2 Instance ID, and you will be redirected to see the running EC2 instance.

Instances (1) [Info](#)

Find instance by attribute or tag (case-sensitive)

[Clear filters](#)

<input type="checkbox"/>	Name ▾	Instance ID	Instance state ▾	Instance type ▾	Status check	Alarm status	Availability Zone ▾	Public IPv4 DNS
<input type="checkbox"/>	-	i-06ed759c4c1160881	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-3-95-162-15

6. Select the EC2 Instance and Go to **Actions**. Select **Modify IAM role** under **Security**.

Instances (1/1) [Info](#)

Find Instance by attribute or tag (case-sensitive)

<input checked="" type="checkbox"/>	Name ✎ ▾	Instance ID	Instance state ▾	Instance type ▾	Status check
<input checked="" type="checkbox"/>	ECS Instance - whiz	i-0676daa98d112a536	Running	t2.micro	2/2 checks passed

Actions ▴

- Connect
- View details
- Manage instance state
- Instance settings
- Networking
- Security**
- Image and templates
- Monitor and troubleshoot

Change security groups

Get Windows password

Modify IAM role

7. Select **task_199_<RANDOM_NUMBER>** from the drop-down and click on **Update IAM role** button.

Modify IAM role [Info](#)

Attach an IAM role to your instance.

Instance ID

 [i-0676daa98d112a536](#) (ECS Instance - whiz)

IAM role

Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.

[task199_profile_50811.92679541](#)



[Create new IAM role](#) 

Cancel

Update IAM role

Task 6: SSH into the underlying EC2 instance

1. Please follow the steps in [SSH into EC2 Instance](#).

Task 7: Create a Dockerfile and build a docker image

1. Get the root access using the following command:

```
sudo su
```

2. Now run the updates using the following command:

```
yum -y update
```

3. Check the Docker version by running the following command:

```
docker version
```

4. Check all the docker processes running in the ECS Cluster by running the following command:

```
docker ps
```

5. Install AWS CLI by running the following command:

```
yum install awscli -y
```

6. Create a Dockerfile by running the following command:

```
touch Dockerfile
```

7. Put the instructions in the Dockerfile by running the following command:

```
vi Dockerfile
```

8. Paste the below code in Dockerfile after pressing **i** or insert mode

```
#This is a sample Image
FROM ubuntu
MAINTAINER someone@example.com
RUN apt-get update
RUN apt-get install nginx -y
CMD ["echo","Image created"]
```

9. While pasting the above script in Dockerfile's vi editor, you may get issue with formatting. Please follow the below instructions:

```
#This is a sample Image

FROM ubuntu

MAINTAINER someone@example.com

RUN apt-get update

RUN apt-get install nginx ?~@~Sy

CMD [?~@~\echo?~@~],?~@~]Image created?~@~]]
~
~
~
~
```

Replace by -y

Remove entire line and rewrite as
CMD ["echo","Image created"]

10. Save the file by pressing **esc** and enter: **wq** to close the vi editor.
11. Run the Build command:

```
docker build -t whiz .
```

Note: demo will be the name of our newly created docker image. Don't forget to add a **(.)** **dot** after the demo (image name).

12. Check the availability of a new image, by running the following command:

```
docker images
```

Task 8: Create a repository in ECR and run the push commands

1. Navigate to **Elastic Container Registry** by clicking on the **Services** menu at the top, then click on **Elastic Container Registry** in the **Containers** section.
2. Click on **Get Started** under Create a Repository.
3. Fill in the below details to create an ECR repository.
 - a. Visibility settings: Select **Private**
 - b. Repository name: Enter **whiz**
 - c. Keep other options as default.

General settings

Visibility settings [Info](#)

Choose the visibility setting for the repository.

☒ **Private**
Access is managed by IAM and repository policy permissions.

☐ **Public**
Publicly visible and accessible for image pulls.

Repository name

Provide a concise name. A developer should be able to identify the repository contents by the name.

679770537133.dkr.ecr.us-east-1.amazonaws.com/ **whiz**

4 out of 256 characters maximum (2 minimum). The name must start with a letter and can only contain lowercase letters, numbers, hyphens, underscores, and forward slashes.

Tag immutability [Info](#)

Enable tag immutability to prevent image tags from being overwritten by subsequent image pushes using the same tag. Disable tag immutability to allow image tags to be overwritten.

☐ **Disabled**

[i](#) Once a repository is created, the visibility setting of the repository can't be changed.

4. Finally, click on the **Create Repository** to create an ECR Repository.
5. The repository is now created, select the repository and click on the **View push commands** button to upload the created custom image in the ECR Repository.

✓ Successfully created repository whiz View push commands ✕

ECR > Repositories

Private Public

Private repositories (1 of 1)

↻ View push commands Delete Edit Create repository

Repository name	URI	Created at	Tag immutability	Scan on push	Encryption type
<input checked="" type="radio"/> whiz	730885193804.dkr.ecr.us-east-1.amazonaws.com/whiz	09 Feb 2021 14:36:54	Disabled	Disabled	AES-256

6. **Copy the commands one by one** and **paste the same on the terminal** to execute.
7. Since we have already built the Dockerfile to the docker image. We can **skip copying and pasting the 2nd command**.

Push commands for whiz×

[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-east-1 | docker login --username AWS --password-
```

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:

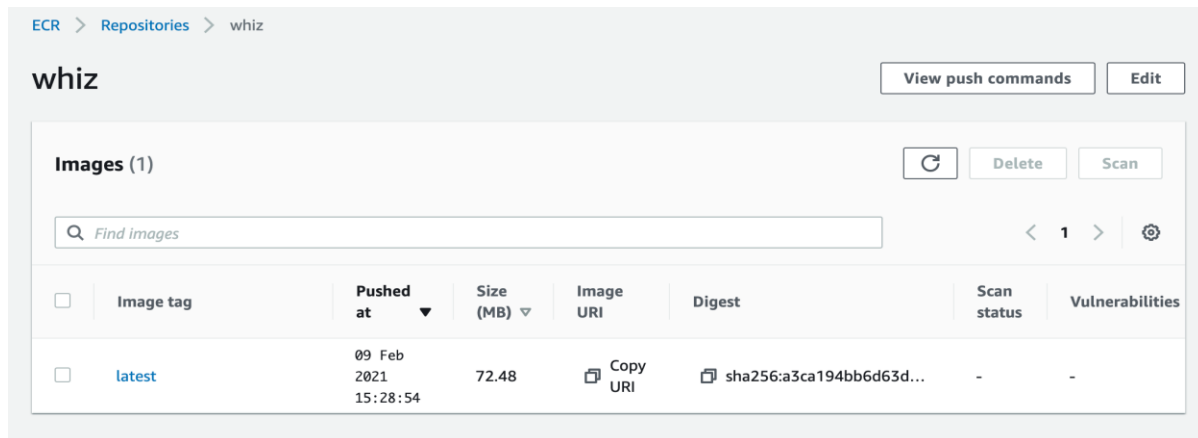
```
docker build -t whiz .
```
3. After the build completes, tag your image so you can push the image to this repository:

```
docker tag whiz:latest 730885193804.dkr.ecr.us-east-1.amazonaws.com/whiz:latest
```
4. Run the following command to push this image to your newly created AWS repository:

```
docker push 730885193804.dkr.ecr.us-east-1.amazonaws.com/whiz:latest
```

Close

8. After running all the 3 commands, can click on the repository name to check the created image.



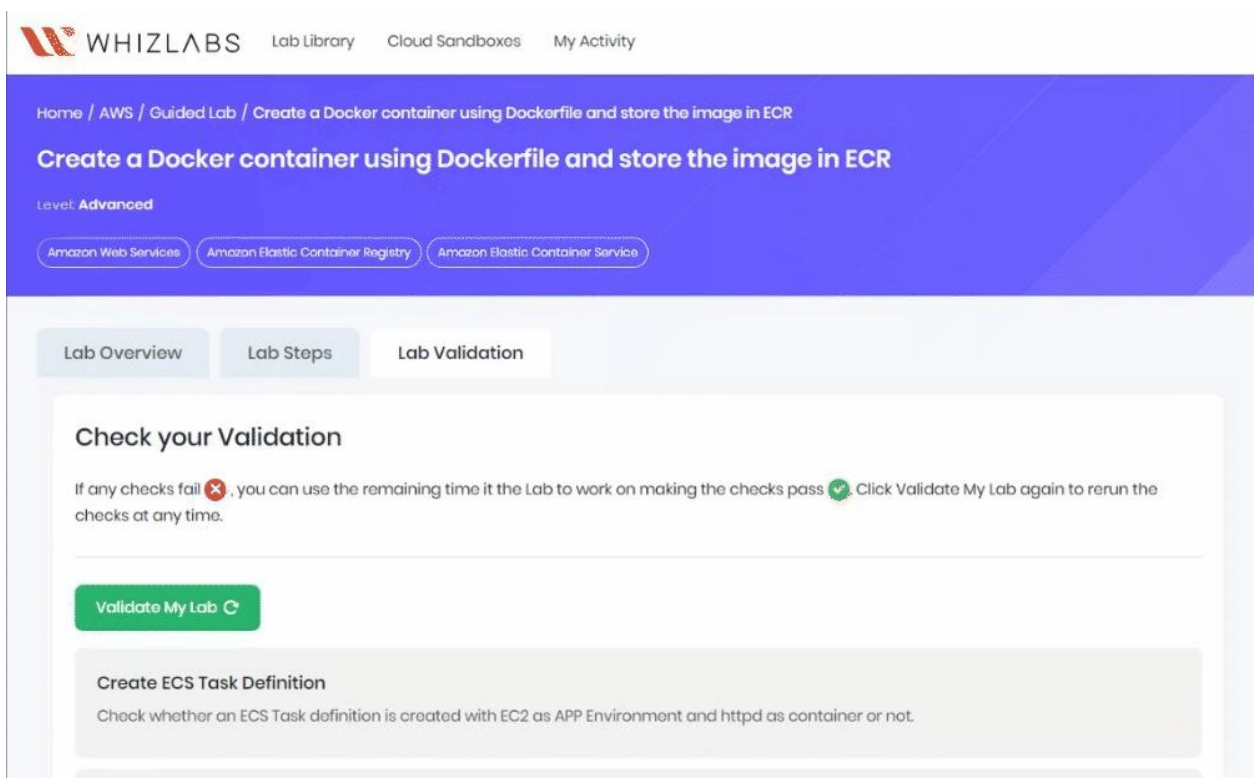
9. Image is now pushed to ECR repository, it can now be used in task definition to create services and tasks.

Do you know ?

AWS ECS is often used in conjunction with other AWS services, such as Amazon ECR (Elastic Container Registry) to store and manage container images, AWS CloudFormation for infrastructure as code, AWS IAM for access management, and AWS CloudWatch for monitoring and logging.

Task 9: Validation Test

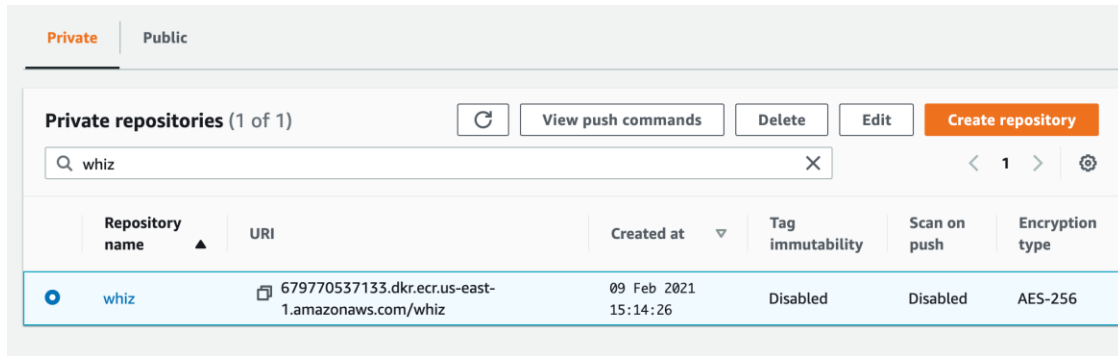
1. Once the lab steps are completed, please click on the **Validation** button on the Right side panel.
2. This will validate the resources in the AWS account and displays whether you have completed this lab successfully or not.
3. Sample output :



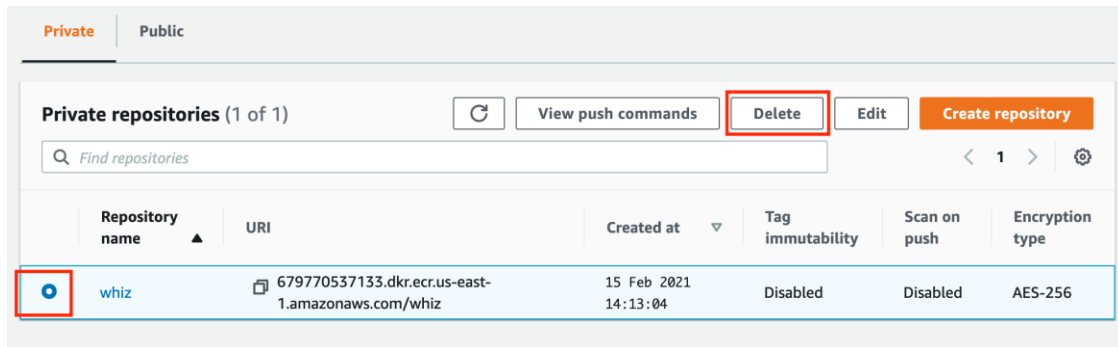
Task 10: Delete AWS Resources

Deleting ECR repository

1. Navigate to **Elastic Container Registry** by clicking on the **Services** menu in the top, then click on **Elastic Container Registry** in the **Containers** section.
2. On the left sidebar, click on the **Repositories** option present under the **Amazon ECR** section.
3. whiz ECS repository will be listed here.



4. To delete the service, do the following task:
 - a. Select the present repository,
 - b. Click on the **Delete** button.



5. Confirm the deletion by typing the **delete** phrase in the required field, and click on the **Delete** button.

Delete whiz

Are you sure you want to delete whiz? Your images will also be deleted.

To confirm deletion, type *delete* in the field.

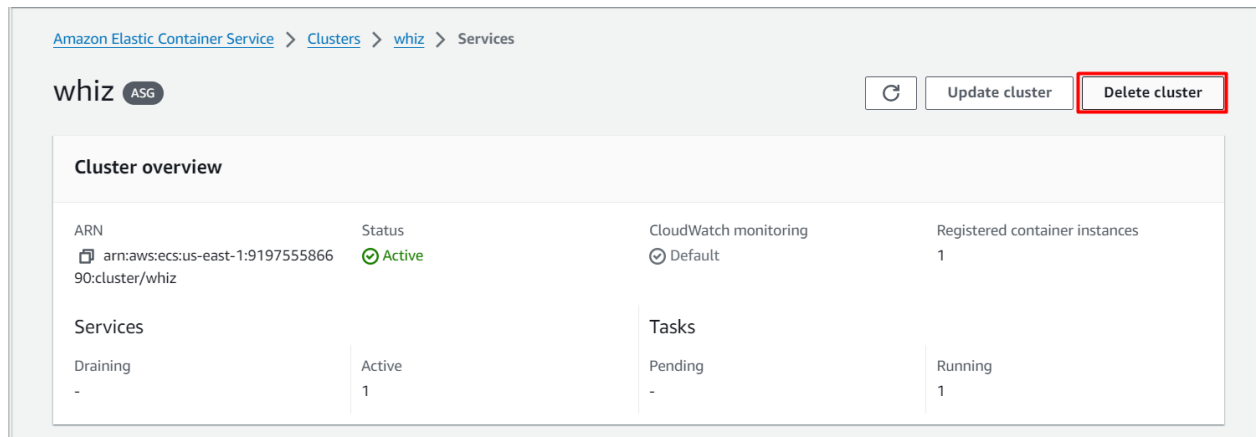
Cancel Delete

6. It will be deleted automatically.

✔ Successfully deleted repository whiz

Deleting ECS Cluster

1. Navigate to **Clusters** from the left navigation panel. Select the cluster and Click on the **Delete Cluster** button.



The screenshot shows the Amazon Elastic Container Service (ECS) console interface. The breadcrumb navigation at the top reads: [Amazon Elastic Container Service](#) > [Clusters](#) > [whiz](#) > [Services](#). Below the navigation, the cluster name 'whiz' is displayed with an 'ASG' tag. To the right of the cluster name are three buttons: a refresh icon, 'Update cluster', and 'Delete cluster' (which is highlighted with a red rectangular border). Below the buttons is the 'Cluster overview' section, which contains a table with the following data:

Cluster overview			
ARN	Status	CloudWatch monitoring	Registered container instances
arn:aws:ecs:us-east-1:9197555866:cluster/whiz	✔ Active	✔ Default	1
Services		Tasks	
Draining	Active	Pending	Running
-	1	-	1

2. Confirm the deletion by entering the phrase **delete whiz** in the pop-up window.

Delete whiz

×

By confirming, all the container instances currently registered to this cluster will be forcefully deregistered. All running services will also be deleted.

ⓘ

Deleting the cluster also deletes the CloudFormation stack [Infra-ECS-Cluster-whiz-820fcb6a](#) [↗](#)

After deletion, the cluster transitions to the INACTIVE state. Clusters with an INACTIVE status might remain discoverable in your account for a period of time.

Enter the phrase "delete whiz" to confirm deletion.

delete whiz

Cancel

Delete

3. Deletion will take up to 3-5 minutes.

Completion and Conclusion

1. You have successfully created and launched Amazon ECS Cluster.
2. You have successfully created an ECR repository.
3. You have successfully pushed the custom image to the ECR repository.

End Lab

1. Sign out of AWS Account.

2. You have successfully completed the lab.
3. Once you have completed the steps, click on **End Lab** from your Whizlabs lab console and wait till the process gets completed.