

ECR

Creating Repositories	1
Local Access	2
Pushing Docker Images	3
Movie Service	3
Customer Service	4

Creating Repositories

We will create 2 repositories.

- movie-service
- customer-service

The screenshot shows the 'Visibility settings' section of the AWS ECR repository creation wizard. It includes an 'Info' link, a note about choosing visibility, and two options: 'Private' (selected) and 'Public'. Below is the 'Repository name' section with a note about providing a concise name, followed by a text input field containing '941077029185.dkr.ecr.us-east-1.amazonaws.com/' and 'movie-service'.

Individual services will provide commands on how to push images.

Push commands for movie-service

macOS / Linux

Windows

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-east-1 | docker login --username AWS --password-stdin  
941077029185.dkr.ecr.us-east-1.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 movie-service .
```

3. After the build completes, tag your image so you can push the image to this repository:

```
docker tag movie-service:latest 941077029185.dkr.ecr.us-east-1.amazonaws.com/movie-service:latest
```

4. Run the following command to push this image to your newly created AWS repository:

```
docker push 941077029185.dkr.ecr.us-east-1.amazonaws.com/movie-service:latest
```

Create these 2 repositories

Repositories (2)

Filter status

Repository name

URI

customer-service

941077029185.dkr.ecr.us-east-1.amazonaws.com/customer-service

movie-service

941077029185.dkr.ecr.us-east-1.amazonaws.com/movie-service

Local Access

- Ensure that your dev user has access to push docker images into the ECR.

- I added the below permission to the “vins-dev” user.

The screenshot shows the AWS IAM Permissions Policies page. At the top, there are three tabs: 'Users (1)', 'Permissions', and 'Access Advisor'. The 'Permissions' tab is selected. Below the tabs, the heading 'Permissions policies (6) Info' is displayed, followed by the sub-instruction 'You can attach up to 10 managed policies.' A search bar labeled 'Search' and a filter button labeled 'Filter b' are visible. The main list contains one item: 'AmazonEC2ContainerRegistryPowerUser' with a blue link. The list has columns for checkboxes, policy name, and actions.

	Policy name	Actions
<input type="checkbox"/>	AmazonEC2ContainerRegistryPowerUser	All

Pushing Docker Images

At this point, we can push docker images. **Ensure that you have configured your aws credentials** in your local.

You should have credentials under

```
~/.aws/credentials
```

You should have config directory

```
~/.aws/config
```

Movie Service

- Create the app jar

```
mvn clean package
```

- Build the docker image - **DO NOT FORGET** the --platform

```
docker build -t movie-service . --platform=linux/amd64
```

- We can follow the rest of the instructions from the ECR push-commands for movie-service

The screenshot shows the AWS ECR console interface. At the top, there is a breadcrumb navigation: Amazon ECR > Private registry > Repositories > movie-service. Below this, the repository name "movie-service" is displayed. Under the heading "Images (1)", there is a search bar labeled "Search artifacts". A table lists one image entry:

	Image tag	Artifact type
<input type="checkbox"/>	latest	Image

Customer Service

Repeat the steps for customer-service as well.

The screenshot shows the AWS ECR console interface. At the top, there is a breadcrumb navigation: Amazon ECR > Private registry > Repositories > customer-service. Below this, the repository name "customer-service" is displayed. Under the heading "Images (1)", there is a search bar labeled "Search artifacts". A table lists one image entry:

	Image tag	Artifact type
<input type="checkbox"/>	latest	Image