Creating a Repository

Before you can push your Docker images to Amazon ECR, you need to create a repository to store them in. You can create Amazon ECR repositories with the AWS Management Console, or with the AWS CLI and AWS SDKs.

To create a repository

1. Open the Amazon ECS console at https://console.aws.amazon.com/ecs/.
2. From the navigation bar, choose the region to create your repository in.

   Note

   Amazon ECR is available in the following regions:

Region Name	Region
US East (N. Virginia)	us-east-1
US West (Oregon)	us-west-2
EU (Ireland)	eu-west-1
US West (N. California)	us-west-1
EU (Frankfurt)	eu-central-1
Asia Pacific (Tokyo)	ap-northeast-1
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2

3. In the navigation pane, choose **Repositories**.
4. On the **Repositories** page, choose **Create repository**.
5. For **Repository name**, enter a unique name for your repository and choose **Next step**.
6. (Optional) On the **Build, tag, and push Docker image** page, complete the following steps to push an image to your new repository. If you do not want to push an image at this time, you can choose **Done** to finish.

a. Retrieve the **docker login** command that you can use to authenticate your Docker client to your registry by pasting the **aws ecr get-login** command from the console into a terminal window.

   Note

   The **get-login** command is available in the AWS CLI starting with version 1.9.15. You can check your AWS CLI version with the **aws --version** command.

b. Run the **docker login** command that was returned in the previous step. This command provides an authorization token that is valid for 12 hours.

   Important

   When you execute this **docker login** command, the command string can be visible by other users on your system in a process list (**ps -e**) display. Because the **docker login** command contains authentication credentials, there is a risk that other users on your system could view them this way and use them to gain push and pull access to your repositories. If you are not on a secure system, you should consider this risk and log in interactively by omitting the `-p`*`password`* option, and then entering the password when prompted.

c. (Optional) If you have a Dockerfile for the image you want to push, build the image and tag it for your new repository by pasting the **docker build** command from the console into a terminal window (make sure you are in the same directory as your Dockerfile).
d. Tag the image for your Amazon ECR registry and your new repository by pasting the **docker tag** command from the console into a terminal window. The console command assumes that your image was built from a Dockerfile in the previous step; if you did not build your image from a Dockerfile, replace the first instance of *`repository`*`:latest` with the image ID or image name of your local image that you want to push.
e. Push the newly tagged image to your Amazon ECR repository by pasting the **docker push** command into a terminal window.
f. Choose **Done** to finish.

Pushing an Image

If you have a Docker image available in your development environment, you can push it to an Amazon ECR repository with the **docker push** command.

Important

Amazon ECR users require permissions to call `ecr:GetAuthorizationToken` before they can authenticate to a registry and push or pull any images from any Amazon ECR repository. Amazon ECR provides several managed policies to control user access at varying levels; for more information, seeAmazon ECR Managed Policies.

To push a Docker image to an Amazon ECR repository

1. Authenticate your Docker client to the Amazon ECR registry you intend to push your image to. Authentication tokens must be obtained for each registry used, and the tokens are valid for 12 hours. For more information, see Registry Authentication.
2. If your image repository does not exist in the registry to intend to push to yet, create it. For more information, see Creating a Repository.
3. Identify the image to push. Run the **docker images** command to list the images on your system.

   ```
   $ docker images
   ```

   You can identify an image with the *repository:tag* or the image ID in the resulting command output.

4. Tag your image with the Amazon ECR registry, repository, and optional image tag name combination to use. The registry format is *aws_account_id*.dkr.ecr.*region*.amazonaws.com. The repository name should match the repository that you created for your image. If you omit the image tag, we assume the tag is `latest`.

   The following example tags an image with the ID *e9ae3c220b23* as*aws_account_id*.dkr.ecr.*region*.amazonaws.com/my-web-app.

   ```
   $ docker tag e9ae3c220b23
   aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app
   ```

5. Push the image using the **docker push** command.

```
$ docker push aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app
The push refers to a repository
[aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app] (len: 1)
e9ae3c220b23: Pushed
a6785352b25c: Pushed
 0998bf8fb9e9: Pushed
 0a85502c06c9: Pushed
```

```
latest: digest:
sha256:01f58d96d1fa90e3eb0dd0ac3d893bcaf00d736f2bc82539d3531170e707648c
size: 6778
```

Pulling an Image

If you have a Docker image available in Amazon ECR, you can pull it to your local environment with the **docker pull**command.

Important

Amazon ECR users require permissions to call `ecr:GetAuthorizationToken` before they can authenticate to a registry and push or pull any images from any Amazon ECR repository. Amazon ECR provides several managed policies to control user access at varying levels; for more information, seeAmazon ECR Managed Policies.

To pull a Docker image from an Amazon ECR repository

1. Authenticate your Docker client to the Amazon ECR registry you intend to pull your image from. Authentication tokens must be obtained for each registry used, and the tokens are valid for 12 hours. For more information, see Registry Authentication.
2. (Optional) Identify the image to pull.
    - You can list the repositories in a registry with the **aws ecr describe-repositories** command.

```
$ aws ecr describe-repositories
{
    "repositories": [
        {
            "registryId": "aws_account_id",
            "repositoryName": "my-web-app",
            "repositoryArn": "arn:aws:ecr:us-east-
1:aws_account_id:repository/my-web-app"
        }
```

```
    ]
}
```

The example registry above has a repository called `my-web-app`.

- You can list the images within a repository with the **aws ecr list-images** command.

```
$ aws ecr list-images --repository-name my-web-app
{
    "imageIds": [
        {
            "imageTag": "latest",
            "imageDigest":
"sha256:01f58d96d1fa90e3eb0dd0ac3d893bcaf00d736f2bc82539d3531170e
707648c"
        }
    ]
}
```

The example repository above has an image tagged as `latest`.

3. Pull the image using the **docker pull** command. The image name format should be *registry*/*repository*[:*tag*].

```
$$ docker pull aws_account_id.dkr.ecr.us-east-1.amazonaws.com/my-web-
app:latest
latest: Pulling from my-web-app

0a85502c06c9: Pull complete
0998bf8fb9e9: Pull complete
a6785352b25c: Pull complete
 e9ae3c220b23: Pull complete
 Digest:
sha256:01f58d96d1fa90e3eb0dd0ac3d893bcaf00d736f2bc82539d3531170e707648c

Status: Downloaded newer image for aws_account_id.dkr.ecr.us-east-
1.amazonaws.com/my-web-app:latest
```