

ECR

Elastic Container Registry (ECR)

Amazon Elastic Container Registry is a fully managed docker container registry hosted on the amazon AWS data centers. The ECR service is secured, reliable and scalable allowing you to grow your applications and services without worrying about capacity and security.

Amazon ECR comes with a few main components mentioned below:

S.No	Components	Description
1	Registry	The registry is the primary logical resource that holds all the images.
2	Authorization Token	The registry authentication mechanism secures the registry and allows access to authenticated users only.
3	Repository	The repository contains the docker images of an application
4	Repository Policy	Policies control access and life cycles.
5	Images	Container images are used with the docker push and pull commands.

- ECR has a limit of 1000 repositories per region and 1000 images for repository, which is very high and probably enough for 99.9% of AWS customers.
- The next limitation we should know about the number of pull and push requests we can run per second, per region and per account, which is 200 sustained requests and a burst of 400.
- Another limitation is maximum layers of image which AWS can sustain is 127 layers and 100 tags per image.

ECR Pricing:

The amazon ECR pricing structure is straightforward and based on usage; it does not have any upfront costs.

Specially, the ECR pricing is based on storage usage, meaning that you pay only for the amount of data that is stored in our repositories and the data transfer out of the internet.

- In free tier AWS account we get 500 MB of free storage for repositories
- And 1 GB data transfer over the internet

Creating an access key for AWS CLI:

- To push an image to ECR we need to have an authentication/access
- To pull an image from ECR we need to have an authentication/access
- We need to have a Programmatic IAM user with required permissions

Configuring Access to ECR:

- We can attach a role with required permissions to an EC2 instance is one method
- We can do "AWS configure" on EC2 instance or windows/mac machine
- Creating Repository:

We can create via two methods:

1. With Management Console
2. With AWS CLI

1. Management Console:

Create repository

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.

229978460853.dkr.ecr.us-east-1.amazonaws.com/

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

2. AWS Command Line:

```
PS C:\Users\kittu> aws ecr create-repository --repository-name my-cli-registry
{
  "repository": {
    "repositoryArn": "arn:aws:ecr:us-east-1:229978460853:repository/my-cli-registry",
    "registryId": "229978460853",
    "repositoryName": "my-cli-registry",
    "repositoryUri": "229978460853.dkr.ecr.us-east-1.amazonaws.com/my-cli-registry",
    "createdAt": "2022-06-28T15:16:21-04:00",
    "imageTagMutability": "MUTABLE",
    "imageScanningConfiguration": {
      "scanOnPush": false
    }
  }
}
```

Private

Public

Private repositories (1 of 2)

View push commands

Delete

Actions ▼

Create repository

Find repositories

1

Repository name ▲	URI	Created at ▼	Tag immutability	Scan frequency	Encryption type
aws-nginx	229978460853.dkr.ecr.us-east-1.amazonaws.com/aws-nginx	June 28, 2022, 14:44:54 (UTC-04)	Disabled	Manual	AES-256
my-cli-registry	229978460853.dkr.ecr.us-east-1.amazonaws.com/my-cli-registry	June 28, 2022, 15:16:21 (UTC-04)	Disabled	Manual	AES-256

Creating a "dockerfile":

```
FROM amazonlinux:2
RUN amazon-linux-extras install nginx1.12
EXPOSE 8080
CMD ["nginx", "-g", "daemon off;"]
```

Provide permissions to ECR from CLI console:

Amazon ECR > Repositories > ca-container-registry

ca-container-registry

View push commands

Edit

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
229978460853.dkr.ecr.us-east-1.amazonaws.com
```

Building the “dockerfile” with tag:

```
docker build -t <Repository URI>:<tag name> .
```

```
[root@ip-172-31-94-252 docker_project]# docker build -t 229978460853.dkr.ecr.us-east-1.amazonaws.com/aws-nginx:v4 .
Sending build context to Docker daemon 2.048kB
Step 1/4 : FROM amazonlinux:2
--> ef0e8aec8ddc
Step 2/4 : RUN amazon-linux-extras install nginx1.12
--> Using cache
--> fd96c305c19e
Step 3/4 : EXPOSE 8080
--> Using cache
--> 5bb5c1eb2a08
Step 4/4 : CMD ["nginx", "-g", "daemon off;"]
--> Using cache
--> cd0d0dc56d0a
Successfully built cd0d0dc56d0a
Successfully tagged 229978460853.dkr.ecr.us-east-1.amazonaws.com/aws-nginx:v4
[root@ip-172-31-94-252 docker_project]# docker images
```

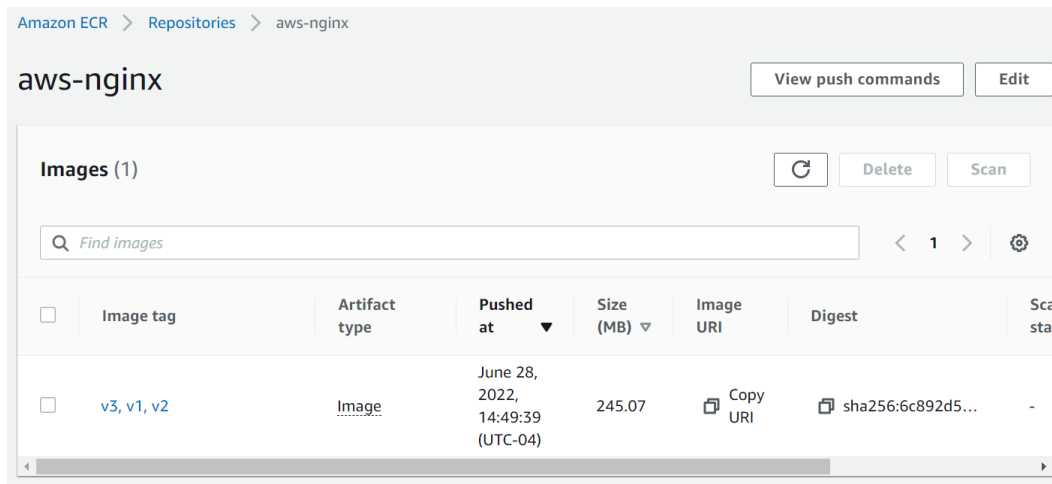
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
aws-nginx	v1	cd0d0dc56d0a	4 minutes ago	590MB
aws-nginx	v2	cd0d0dc56d0a	4 minutes ago	590MB
aws-nginx	v3	cd0d0dc56d0a	4 minutes ago	590MB
229978460853.dkr.ecr.us-east-1.amazonaws.com/aws-nginx	v1	cd0d0dc56d0a	4 minutes ago	590MB
229978460853.dkr.ecr.us-east-1.amazonaws.com/aws-nginx	v2	cd0d0dc56d0a	4 minutes ago	590MB
229978460853.dkr.ecr.us-east-1.amazonaws.com/aws-nginx	v3	cd0d0dc56d0a	4 minutes ago	590MB
229978460853.dkr.ecr.us-east-1.amazonaws.com/aws-nginx	v4	cd0d0dc56d0a	4 minutes ago	590MB
amazonlinux	2	ef0e8aec8ddc	6 days ago	164MB

```
[root@ip-172-31-94-252 docker_project]# docker push 229978460853.dkr.ecr.us-east-1.amazonaws.com/aws-nginx:v1
The push refers to repository [229978460853.dkr.ecr.us-east-1.amazonaws.com/aws-nginx]
26ef85c11d99: Pushed
209cb42bdfb7: Pushed
v1: digest: sha256:6c892d52c7e15437c52c8f1d3207687f2a28be02823c1e633c22a1e6f3336a4c size: 742
```

```
Docker push <Repository URI>:<tag name>
```

View/Listing Images:

To view the images pushed into the repository go to the below screen on Management console:



To view the images pushed into the repository go to the below screen on CLI:

```
[root@ip-172-31-94-252 docker_project]# aws ecr list-images --repository-name aws-nginx
{
  "imageIds": [
    {
      "imageTag": "v3",
      "imageDigest": "sha256:6c892d52c7e15437c52c8f1d3207687f2a28be02823c1e633c22a1e6f3336a4c"
    },
    {
      "imageTag": "v1",
      "imageDigest": "sha256:6c892d52c7e15437c52c8f1d3207687f2a28be02823c1e633c22a1e6f3336a4c"
    },
    {
      "imageTag": "v2",
      "imageDigest": "sha256:6c892d52c7e15437c52c8f1d3207687f2a28be02823c1e633c22a1e6f3336a4c"
    }
  ]
}
```

Pulling Images from ECR into Local:

`docker pull <Repository URI>:<tag name>`

```
[root@ip-172-31-94-252 docker_project]# docker images
REPOSITORY                                TAG      IMAGE ID      CREATED      SIZE
229978460853.dkr.ecr.us-east-1.amazonaws.com/aws-nginx  v1       cd0d0dc56d0a  7 minutes ago  590MB
229978460853.dkr.ecr.us-east-1.amazonaws.com/aws-nginx  v3       cd0d0dc56d0a  7 minutes ago  590MB
229978460853.dkr.ecr.us-east-1.amazonaws.com/aws-nginx  v4       cd0d0dc56d0a  7 minutes ago  590MB
aws-nginx                                           v1       cd0d0dc56d0a  7 minutes ago  590MB
aws-nginx                                           v3       cd0d0dc56d0a  7 minutes ago  590MB
amazonlinux                                         2        ef0e8aec8ddc  6 days ago    164MB
[root@ip-172-31-94-252 docker_project]# docker pull 229978460853.dkr.ecr.us-east-1.amazonaws.com/aws-nginx:v2
v2: Pulling from aws-nginx
Digest: sha256:6c892d52c7e15437c52c8f1d3207687f2a28be02823c1e633c22a1e6f3336a4c
Status: Downloaded newer image for 229978460853.dkr.ecr.us-east-1.amazonaws.com/aws-nginx:v2
229978460853.dkr.ecr.us-east-1.amazonaws.com/aws-nginx:v2
[root@ip-172-31-94-252 docker_project]# docker images
REPOSITORY                                TAG      IMAGE ID      CREATED      SIZE
aws-nginx                                           v1       cd0d0dc56d0a  7 minutes ago  590MB
aws-nginx                                           v3       cd0d0dc56d0a  7 minutes ago  590MB
229978460853.dkr.ecr.us-east-1.amazonaws.com/aws-nginx  v1       cd0d0dc56d0a  7 minutes ago  590MB
229978460853.dkr.ecr.us-east-1.amazonaws.com/aws-nginx  v2       cd0d0dc56d0a  7 minutes ago  590MB
229978460853.dkr.ecr.us-east-1.amazonaws.com/aws-nginx  v3       cd0d0dc56d0a  7 minutes ago  590MB
229978460853.dkr.ecr.us-east-1.amazonaws.com/aws-nginx  v4       cd0d0dc56d0a  7 minutes ago  590MB
amazonlinux                                         2        ef0e8aec8ddc  6 days ago    164MB
```

Deleting images from Repository URI:

`docker rmi <Repository>:<Tag name>` → to delete from local repository

`aws ecr batch-delete-image --repository-name <Repository Name> --image-ids imageTag=<tag name>` → to delete from ECR repository uri

Creating LifeCycle Policies:

AWS ECR lifecycle policies allow developers and administrators to manage container images using specific rules that automatically helps with the following tasks:

- Delete old images based on criteria we define such as image age and size
- Manage the number of active images

Creating a LifeCycle policy:

- All the lifecycle policies created will be applied within 24 hours of creation

Create lifecycle rule

Lifecycle rule configuration

If you apply this rule as your lifecycle policy, it may be evaluated immediately. It is recommended that you dry run your rules first.

Rule priority
Specify a rule priority, which must be unique. Values do not need to be sequential across rules in a policy.

Rule description
Specify a description for your lifecycle policy.

Image status
Indicates whether the image is tagged or not.

☐ Tagged
☐ Untagged
☒ Any

Match criteria
Specify the count type to apply to the images. If "countType" is set to "imageCountMoreThan", you also specify "countNumber" to create a rule that sets a limit on the number of images that exist in your repository. If "countType" is set to "sinceImagePushed", you also specify "countUnit" and "countNumber" to specify a time limit on the images that exist in your repository.

Image count more than ▼

5

Rule action
The only supported value is expire. The image will be deleted from your repository. This action cannot be undone.

expire ▼