

Create an ECR repository. Connect it using AWS CLI and push sample docker images to it: -

1: - Created A IAM Role For ECR: -

The screenshot shows the AWS IAM console interface. On the left is a navigation menu with sections like 'Identity and Access Management (IAM)', 'Access management', and 'Access reports'. The main content area is titled 'Summary' and shows details for a role named 'allowecr'. The details include: Creation date (February 08, 2024, 09:07 (UTC+05:30)), ARN (arn:aws:iam::975050349146:role/MIHIR), Instance profile ARN (arn:aws:iam::975050349146:instance-profile/MIHIR), Last activity (21 minutes ago), and Maximum session duration (1 hour). Below the summary, there are tabs for 'Permissions', 'Trust relationships', 'Tags', 'Access Advisor', and 'Revoke sessions'. The 'Permissions' tab is active, showing 'Permissions policies (7)' and a table of policies. The table has columns for 'Policy name', 'Type', and 'Attached entities'. One policy named 'allowecr' is listed with a 'Customer inline' type and 0 attached entities.

- Created a IAM ROLE for the ECR
- Created custom inline policy for Role

Modify permissions in allowecr [Info](#)

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using

The screenshot shows the 'Policy editor' for the 'allowecr' role. It has two tabs: 'Visual' and 'JSON'. The 'JSON' tab is selected, showing a JSON policy document. The document is as follows:

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": "ecr:CreateRepository",
7       "Resource": "arn:aws:ecr:us-east-2:975050349146:repository/mihir"
8     }
9   ]
10 }
```

On the right side of the editor, there is an 'Edit statement' section and a 'Select an e' dropdown menu.

2:- Created a EC2 Instance and attach the IAM Role :-

The screenshot shows the AWS EC2 console 'Instances (1)' page. It includes a search bar, filters (Instance state = running), and a table of instances. The table has columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. One instance named 'MJ' is listed with Instance ID 'i-0b60b676dbb296985', state 'Running', type 't2.micro', and status '2/2 checks passed'.

- Modify the IAM role and attach the role to EC2 instance.


3:- Modify a IAM Role to EC2 Instance: -

EC2 > Instances > i-0b60b676dbb296985 > Modify IAM role

Modify IAM role [Info](#)


Attach an IAM role to your instance.



Instance ID

 i-0b60b676dbb296985 (MJ)

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.



 [Create new IAM role](#) 

[Cancel](#) [Update IAM role](#)

4:- SSH Into the EC2 instance :-

```
C:\Users\mihir.jain\Downloads>ssh -i "Key-pair.pem" ubuntu@ec2-18-216-77-212.us-east-2.compute.amazonaws.com
```

5:- Install the aws cli into the linux of Ec2 Instance :-

```
ubuntu@ip-172-31-38-31:~$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 57.2M  100 57.2M    0     0  109M      0  --:--:-- --:--:-- --:--:--  109M
ubuntu@ip-172-31-38-31:~$ sudo apt install unzip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  zip
The following NEW packages will be installed:
  unzip
```

```
ubuntu@ip-172-31-38-31:~$ unzip awscliv2.zip
Archive:  awscliv2.zip
```

6: - Created a repo with the aws cli:-

```
ubuntu@ip-172-31-38-31:~$ aws ecr create-repository --repository-name mihir
{
  "repository": {
    "repositoryArn": "arn:aws:ecr:us-east-2:975050349146:repository/mihir",
    "registryId": "975050349146",
    "repositoryName": "mihir",
    "repositoryUri": "975050349146.dkr.ecr.us-east-2.amazonaws.com/mihir",
    "createdAt": "2024-02-08T08:50:10.589000+00:00",
    "imageTagMutability": "MUTABLE",
    "imageScanningConfiguration": {
      "scanOnPush": false
    },
    "encryptionConfiguration": {
      "encryptionType": "AES256"
    }
  }
}
```

Amazon ECR > Private registry > Repositories

Private repositories

Repositories (1) View push commands Delete Actions Create repository

	Repository name ▲	URI	Created at ▼	Tag immutability	Scan frequency	Encryption type
<input type="radio"/>	mihir	975050349146.dkr.ecr.us-east-2.amazonaws.com/mihir	08 February 2024, 14:20:10 (UTC+05.5)	Disabled	Manual	AES-256

7: - I install the docker in the Linux machine: -

Command:- Sudo apt install docker.io

```
ubuntu@ip-172-31-38-31:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-doc rinse zfs-fuse
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 59 not upgraded.
Need to get 69.8 MB of archives.
```

Setup The docker :-

- To enable the docker:-

Command:- sudo systemctl enable docker

- To check the status of the docker: -

Command:- sudo systemctl status docker

- To start the docker :-

Command :- sudo systemctl start docker

I create a folder for docker and files:-

```
ubuntu@ip-172-31-38-31:~$ mkdir myapp
ubuntu@ip-172-31-38-31:~$ cd myapp
cb: command not found
ubuntu@ip-172-31-38-31:~/myapp$ echo "hello,world!" > index.html
ubuntu@ip-172-31-38-31:~/myapp$ ls
index.html
ubuntu@ip-172-31-38-31:~/myapp$ cat index.html
hello,world!
ubuntu@ip-172-31-38-31:~/myapp$ touch DockerFile
ubuntu@ip-172-31-38-31:~/myapp$ ls -l
total 4
-rw-rw-r-- 1 ubuntu ubuntu 0 Feb  8 12:04 DockerFile
-rw-rw-r-- 1 ubuntu ubuntu 13 Feb  8 12:03 index.html
ubuntu@ip-172-31-38-31:~/myapp$ vi DockerFile
ubuntu@ip-172-31-38-31:~/myapp$ 3L, 50B written
ubuntu@ip-172-31-38-31:~/myapp$ cat DockerFile
FROM nginx
COPY index.html /usr/share/nginx/html
```

- For checking the docker info

```
ubuntu@ip-172-31-38-31:~/myapp$ sudo docker info
Client:
```

- We build docker file

```
ubuntu@ip-172-31-38-31:~/myapp$ sudo docker build -f DockerFile -t myapp .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
Install the buildx component to build images with BuildKit:
https://docs.docker.com/go/buildx/

Sending build context to Docker daemon 3.072kB
Step 1/2 : FROM nginx
latest: Pulling from library/nginx
c57ee500d61: Pull complete
9b0163235c08: Pull complete
f24a6f652778: Pull complete
9f3589a5fc50: Pull complete
f0bd99a47d4a: Pull complete
398157bc5c51: Pull complete
1ef1c1a36ec2: Pull complete
Digest: sha256:84c52dfd55c467e12ef85cad6a252c0990564f03c4850799bf41dd738738691f
Status: Downloaded newer image for nginx:latest
--> b690f5f0a2d5
Step 2/2 : COPY index.html /usr/share/nginx/html
--> 36fa428f7482
Successfully built 36fa428f7482
Successfully tagged myapp:latest
```

- To find the images of the docker

```
ubuntu@ip-172-31-38-31:~/myapp$ sudo docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
myapp	latest	36fa428f7482	2 minutes ago	187MB
nginx	latest	b690f5f0a2d5	3 months ago	187MB
hello-world	latest	d2c94e258dc	9 months ago	13.3kB

```
ubuntu@ip-172-31-38-31:~/myapp$ sudo docker run -d -p 8080:80 myapp
bec4ab09b161bfde1ffbd290aebde61545a23a31da7282715fae373e0927d30a
```

- We build that image and push the docker image into the ECR.

```
ubuntu@ip-172-31-38-31:~/myapp$ sudo docker build -f DockerFile -t mihir .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
Install the buildx component to build images with BuildKit:
https://docs.docker.com/go/buildx/

Sending build context to Docker daemon 3.072kB
Step 1/2 : FROM nginx
--> b690f5f0a2d5
Step 2/2 : COPY index.html /usr/share/nginx/html
--> Using cache
--> 36fa428f7482
Successfully built 36fa428f7482
Successfully tagged mihir:latest
ubuntu@ip-172-31-38-31:~/myapp$ docker tag mihir:latest 975050349146.dkr.ecr.us-east-2.amazonaws.com/mihir:latest
ubuntu@ip-172-31-38-31:~/myapp$ docker push 975050349146.dkr.ecr.us-east-2.amazonaws.com/mihir:latest
The push refers to repository [975050349146.dkr.ecr.us-east-2.amazonaws.com/mihir]
e47f59e22a22: Pushed
f205d290cd76: Pushed
2b28485849ea: Pushed
9f21a390e3f6: Pushed
06536efc503a: Pushed
84e0c9ef07d7: Pushed
83bdf27d9eaa: Pushed
fb1bd2fc5282: Pushed
latest: digest: sha256:0722d6c21d0caeff6578741a9e44c727ffccacdf13bf08fdcfec34ad84573068e size: 1985
ubuntu@ip-172-31-38-31:~/myapp$
```

- We pushed the docker image in the ECR :-

