


## Introduction à Amazon ECR

### Qu'est-ce que ECR ?

ECR signifie **Elastic Container Registry**.




C'est un service AWS qui permet de stocker, gérer et partager des **images Docker**.


- **Elastic** : signifie que le service est **hautement évolutif et disponible**, comme EC2, EBS, EKS.
- **Container** : car il s'agit de gérer des **conteneurs Docker**.
- **Registry** : comme **DockerHub**, **Quay.io**, ou **GCR (Google Container Registry)**, c'est un **registre d'images Docker**.

 **Un registre** est un espace de stockage pour les images Docker. Il facilite le partage et la distribution des conteneurs.

---

## Comparaison : ECR vs DockerHub

Fonctionnalité	DockerHub	Amazon ECR
Gratuité	Oui (public et privé)	Non (facturé selon le stockage/utilisation)
Sécurité	Basique	 Sécurité AWS (IAM, politiques, etc.)
Intégration AWS	 Non	 Intégration native avec IAM, ECS, EKS
Facilité d'usage	Très simple	Requiert quelques commandes AWS
Stockage privé	Oui (limité)	Oui (sécurisé et scalable)

 **ECR est idéal pour les entreprises** déjà dans l'écosystème AWS. Il offre **plus de contrôle, sécurité et intégration** que DockerHub.

---

## DÉMO – Accéder, Pousser et Tirer une image depuis Amazon ECR

### Prérequis :

- Un compte AWS
  - AWS CLI configuré (`aws configure`)
  - Docker installé sur ta machine
- 

### Étapes

## 1. Connexion à ECR via AWS CLI

```
bash
CopyEdit
aws ecr get-login-password --region us-east-1 | docker login --username AWS
--password-stdin <ACCOUNT_ID>.dkr.ecr.us-east-1.amazonaws.com
```

---

## 2. Créer un dépôt (repository) sur ECR

```
bash
CopyEdit
aws ecr create-repository --repository-name mon-premier-repo
```

---

## 3. Créer une image Docker

```
bash
CopyEdit
echo "FROM alpine \n CMD echo Hello depuis ECR" > Dockerfile
docker build -t mon-image-ecr .
```

---

## 4. Taguer l'image pour ECR

```
bash
CopyEdit
docker tag mon-image-ecr:latest <ACCOUNT_ID>.dkr.ecr.us-east-
1.amazonaws.com/mon-premier-repo
```

---

## 5. Pousser l'image dans ECR

```
bash
CopyEdit
docker push <ACCOUNT_ID>.dkr.ecr.us-east-1.amazonaws.com/mon-premier-repo
```

---

## 6. Tirer l'image depuis ECR

```
bash
CopyEdit
docker pull <ACCOUNT_ID>.dkr.ecr.us-east-1.amazonaws.com/mon-premier-repo
```

---

## Conclusion

ECR est une alternative professionnelle à DockerHub pour stocker vos conteneurs Docker, surtout si vous êtes déjà sur AWS.

Il est sécurisé, bien intégré avec IAM, ECS, et EKS, et scalable.

- Créer un serveur de build Docker avec plusieurs services (via Docker Compose)
  - Builder tes images Docker
  - Et les **pousser vers Amazon ECR**
- 

## ☐ PREREQUIS

- Ton compte AWS est prêt
  - Tu as accès à la **console AWS**
  - Tu as une **clé SSH (.pem)** pour te connecter à l'EC2
- 

## ☒ ÉTAPES COMPLÈTES

---

### Étape 1 : Créer l'instance EC2

1. Connecte-toi à ta [console AWS EC2](#)
  2. Clique sur **Launch Instance**
  3. Paramètres :
    - **Name:** docker-build-server
    - **AMI:** Amazon Linux 2 AMI (HVM)
    - **Instance type:** t2.medium ou mieux (t2.micro si budget serré)
    - **Key Pair:** Choisis ou crée une clé .pem
    - **Security Group:**
      - Port **22 (SSH)** ouvert à ton IP
      - (Optionnel) Ports 80/443 si tu veux servir du contenu
    - **Storage:** 20 GB ou plus
    - Laisse les autres options par défaut
  4. Lance l'instance
- 

### Étape 2 : Connecte-toi via SSH

```
chmod 400 your-key.pem
ssh -i your-key.pem ec2-user@<PUBLIC_IP>
```

---

### Étape 3 : Installe Docker et Docker Compose

```
# Docker
sudo yum update -y
sudo amazon-linux-extras install docker -y
```

```
sudo service docker start
sudo usermod -aG docker ec2-user

# Docker Compose (v2 - plugin officiel Docker)
mkdir -p ~/.docker/cli-plugins/
curl -SL
https://github.com/docker/compose/releases/download/v2.27.0/docker-compose-
linux-x86_64 -o ~/.docker/cli-plugins/docker-compose
chmod +x ~/.docker/cli-plugins/docker-compose

# Reconnecte-toi pour appliquer les droits
exit
ssh -i your-key.pem ec2-user@<PUBLIC_IP>
```

Vérifie que tout fonctionne :

```
docker --version
docker compose version
```

---

## Étape 4 : Installer AWS CLI (si pas déjà là)

```
sudo yum install -y aws-cli
```

Configure avec tes identifiants :

```
aws configure
# AWS Access Key ID
# AWS Secret Access Key
# Default region name: ca-central-1
# Default output format: json
```

---

## Étape 5 : Cloner ton projet ou copier tes fichiers

Tu peux :

- Utiliser git :
    - `sudo yum install git -y`
    - `git clone https://github.com/ton-projet.git`
    - `cd ton-projet`
  - Ou utiliser scp :
    - `scp -i your-key.pem -r ./my-docker-project ec2-user@<PUBLIC_IP>:~/`
- 

## Étape 6 : Authentification à ECR

```
aws ecr get-login-password --region ca-central-1 | docker login --username
AWS --password-stdin 769749267333.dkr.ecr.ca-central-1.amazonaws.com
```

---

## Étape 7 : Docker Compose Build & Push vers ECR

Exemple `docker-compose.yml`

```
version: '3.8'
services:
  web:
    build:
      context: .
      dockerfile: Dockerfile
    image: 769749267333.dkr.ecr.ca-central-1.amazonaws.com/mon-app-
web:latest
    ports:
      - "8000:8000"

  redis:
    image: redis:alpine
```

## Build des images

```
docker compose build
```

## Push uniquement les images personnalisées vers ECR

```
docker push 769749267333.dkr.ecr.ca-central-1.amazonaws.com/mon-app-
web:latest
```

---

## ☒ Résumé

Étape	Commande clé
Connecter l'EC2	<code>ssh -i key.pem ec2-user@IP</code>
Installer Docker	<code>amazon-linux-extras install docker</code>
Installer Compose v2	Voir le <code>curl</code> + <code>chmod</code>
Authentifier à ECR	<code>`aws ecr get-login-password ...</code>
Build images avec Compose	<code>docker compose build</code>
Pousser vers ECR	<code>docker push &lt;image&gt;</code>

---

💬 Tu veux que je t'écrive un script `setup.sh` automatique pour tout faire à la création de l'EC2 ?