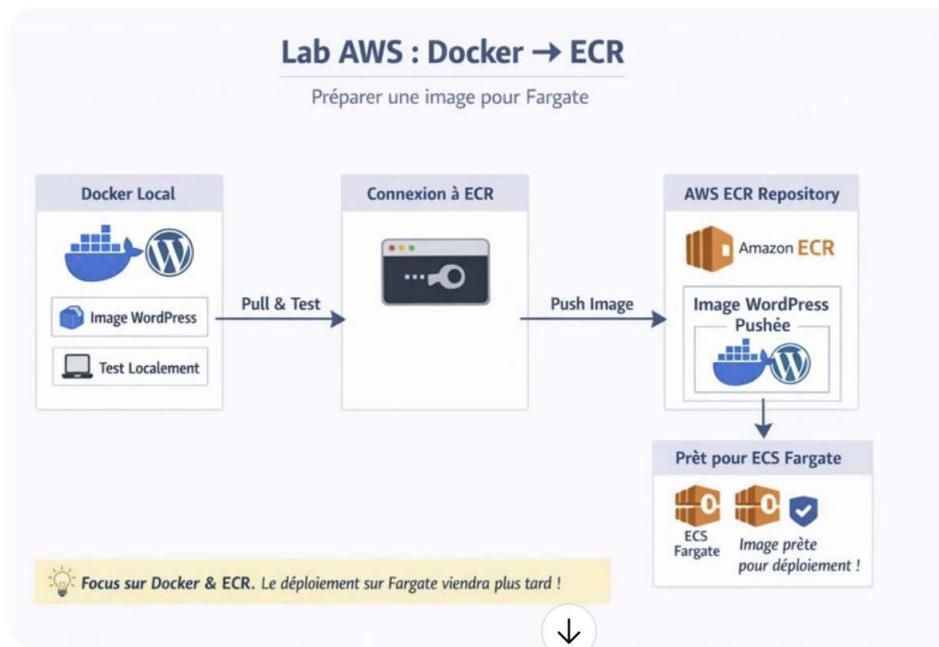


Lab AWS : Docker → ECR (Préparer une image pour Fargate)



Dans ce premier lab pratique, nous allons apprendre à **mettre une image Docker (WordPress) dans AWS ECR**. L'idée est de préparer cette image pour un futur déploiement sur **ECS Fargate** (qui fera l'objet d'un second lab).

- Créez un **repository ECR** dans AWS.
- Télécharger et tester une image Docker localement.
- Se connecter à ECR et **pousser l'image Docker**.
- Vérifier que tout est prêt pour le déploiement futur sur Fargate.

💡 Ce lab se concentre sur **Docker et ECR**. La partie déploiement sur Fargate sera traitée dans un second post pour garder chaque étape claire et facile à suivre.

Sommaire

Lab AWS : Docker → ECR (Préparer une image pour Fargate)	1
--	---

1 Prérequis	2
1.1 Docker.....	2
1.1.1 Pré-requis Windows	2
1.1.2 Télécharger Docker Desktop	3
1.1.3 Vérifier que Docker fonctionne (CLI).....	3
1.2 CLI.....	4
2 – Créer le repository ECR	4
3 – Récupérer l’URL du repository	5
4 – Télécharger WordPress depuis Docker Hub	5
5 - Se connecter à ECR.....	5
6 Taguer l’image pour ECR	6
7 – Pousser l’image vers ECR.....	6
8 – Vérification dans la console AWS	7
9 – Copier l’URL pour ECS	7

1 Prérequis

Système d’exploitation : **Windows**

Un **compte AWS actif**

AWS CLI installé et configuré

Docker installé sur le poste local

1.1 Docker

1.1.1 Pré-requis Windows

Vérifier Windows

Windows 10/11 64 bits

Virtualisation activée (BIOS)

Pour vérifier : systeminfo | findstr /i "Virtualization"

```
C:\Windows\System32>systeminfo | findstr /i "Virtualization"
Virtualization-based security: Status: Running
                                         Base Virtualization Support
```

```
C:\Windows\System32>
```

1.1.2 Télécharger Docker Desktop

Aller sur le site officiel Docker : <https://www.docker.com/products/docker-desktop/>

Télécharger Docker Desktop for Windows

Lancer l'installateur

Cocher :

Use WSL 2 instead of Hyper-V (recommandé)

Terminer l'installation

Redémarrer le PC

Puis vérifier que WSL est bien installé : wsl –list --verbose

```
C:\Windows\System32>wsl --list --verbose
NAME          STATE      VERSION
* docker-desktop Stopped    2
Ubuntu        Stopped    2
```

```
C:\Windows\System32>
```

1.1.3 Vérifier que Docker fonctionne (CLI)

Ouvrir PowerShell ou Terminal Windows :

```
docker --version
```

```
C:\Windows\System32>docker --version
Docker version 28.3.3, build 980b856
```

```
C:\Windows\System32>
```

Puis : lancer docker (menu windows)

Ensuite : vérifier que docker est lancé : wsl --list --verbose

```
C:\Windows\System32>wsl --list --verbose
NAME          STATE      VERSION
* Ubuntu        Running   2
docker-desktop Running   2
```

```
C:\Windows\System32>
```

On voit que docker est en cours d'exécution

```
docker run hello-world
```

Si on voit le message (voir figure ci-dessous), c'est OK : Hello from Docker!

```
C:\Windows\System32>docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

1.2 CLI

aws --version

Puis :

aws configure

2 – Créer le repository ECR

Aller dans AWS Console

Service → Elastic Container Registry (ECR)

Clique Create repository

Paramètres :

Repository name : wordpress

Tout le reste : par défaut

Clique Create repository

On vient de créer un dépôt Docker privé dans AWS.

The screenshot shows the AWS ECR console interface. On the left, there's a navigation sidebar with 'Amazon Elastic Container Service' at the top, followed by 'Private registry' and 'Public registry'. Under 'Private registry', 'Repositories' is selected. The main content area is titled 'Private repositories (1)'. It features a search bar labeled 'Search by repository substring'. Below the search bar is a table with columns: 'Repository name', 'URI', 'Created at', 'Tag immutability', and 'Encryption type'. A single row is visible, representing the 'wordpress' repository. The 'URI' column shows the full URL 'https://1234567890.dkr.ecr.ca-central-1.amazonaws.com/wordpress'. The 'Created at' column shows 'January 15, 2026, 17:36:21 (UTC-05)'. The 'Tag immutability' column shows 'Mutable'. The 'Encryption type' column shows 'AES-256'. At the top right of the table, there are buttons for 'View push commands', 'Delete', 'Actions', and a yellow 'Create' button.

3 – Récupérer l’URL du repository

Dans le repository ECR, on verra une URL du type :

(your-ID).dkr.ecr.ca-central-1.amazonaws.com/wordpress

Par exemple : 123456789.dkr.ecr.ca-central-1.amazonaws.com/wordpress

Avec your-ID : l’ID de votre compte AWS

Le copier

4 – Télécharger WordPress depuis Docker Hub

Sur le PC (terminal / PowerShell) :

```
docker pull wordpress
```

```
0839f2de0289: Pull complete
55b6173dd5b2: Pull complete
Digest: sha256:03e91f888c2beb1837d8aaca08ce2a7d043f20af7acc1dd680969a23ea671c4f
Status: Downloaded newer image for wordpress:latest
docker.io/library/wordpress:latest
C:\Windows\System32>
```

Cette commande télécharge l’image officielle wordpress:latest

Elle est maintenant stockée localement

Vérifier :

```
docker images
```

```
C:\Windows\System32>docker images
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
wordpress       latest        03e91f888c2b  2 days ago    1.05GB
nginx-demo     latest        455ff7cdc03f  4 months ago   279MB
hello-world     latest        a0dfb02aac21  5 months ago   20.3kB
C:\Windows\System32>
```

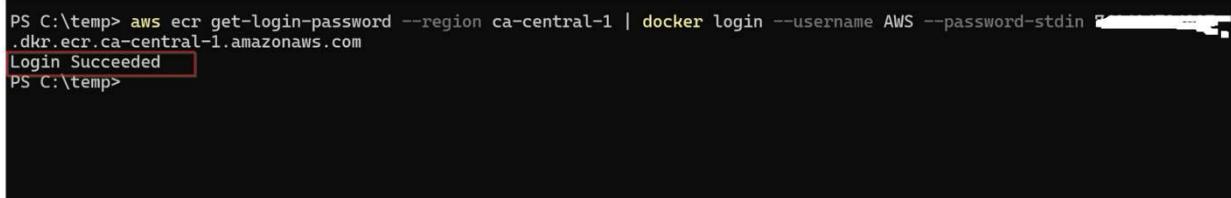
5- Se connecter à ECR

Commande CORRECTE (attention aux espaces et au |) :

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

AWS génère un mot de passe temporaire
Docker l'utilise pour s'authentifier sur ECR
Cette connexion est obligatoire avant un docker push

Si on voit :Login Succeeded
→ correct



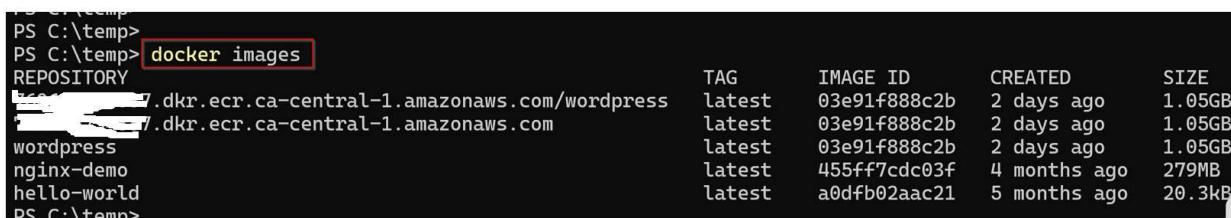
```
PS C:\temp> aws ecr get-login-password --region ca-central-1 | docker login --username AWS --password-stdin [REDACTED].dkr.ecr.ca-central-1.amazonaws.com  
Login Succeeded  
PS C:\temp>
```

6 - Taguer l'image pour ECR

Docker ne peut pas pousser une image sans un tag correspondant au repository.

```
docker tag wordpress:latest (your-ID).dkr.ecr.ca-central-1.amazonaws.com/wordpress:latest
```

Ce que ça fait : L'image est la même, mais elle a maintenant un nom compréhensible par ECR
On voit dans le screen ci-dessous, que l'image est tagée



REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
[REDACTED].dkr.ecr.ca-central-1.amazonaws.com/wordpress	latest	03e91f888c2b	2 days ago	1.05GB
[REDACTED].dkr.ecr.ca-central-1.amazonaws.com	latest	03e91f888c2b	2 days ago	1.05GB
wordpress	latest	03e91f888c2b	2 days ago	1.05GB
nginx-demo	latest	455ff7cdc03f	4 months ago	279MB
hello-world	latest	a0dfb02aac21	5 months ago	20.3kB

7 – Pousser l'image vers ECR

```
docker push (your-ID).dkr.ecr.ca-central-1.amazonaws.com/wordpress:latest
```

Ça peut prendre 1–3 minutes.
Si tout est OK :
Plusieurs lignes Pushed

Pas d'erreur IAM

On obtient (le bas de l'image) figure ci-dessous :

```
d9c4e45f6083: Pushed
6584c9695393: Pushed
353783efef2d: Pushed
cffc893f05bf: Pushed
7625a55b22d4: Pushed
latest: digest: sha256:70ff8c2e3875f76246631c7ce3fd300c877e80a87c7c76aa47af97d5fe26b834 size: 5536
Info → Not all multiplatform-content is present and only the available single-platform image was pushed
sha256:03e91f888c2beb1837d8aaca08ce2a7d043f20af7acc1dd680969a23ea671c4f -> sha256:70ff8c2e3875f76246631c7ce3fd300c877e80a87c7c76aa47af97d5fe26b834
PS C:\temp>
```

8 – Vérification dans la console AWS

Retourner dans ECR

Cliquer sur le repo wordpress

Onglet Images

On doit voir :

Image

Tag : latest

Taille : 263.63

L'image est bien dans AWS ECR.

Image tags	Type	Created at	Image size	Image digest
latest	Image	January 15, 2026, 18:04:18 (UTC-05)	263.63	sha256:70ff8c2e3875f76246631c7ce3fd300c877e80a87c7c76aa47af97d5fe26b834

9 – Copier l'URL pour ECS

Cliquer sur Copy URI du repository

On en aura besoin pour : Task Definition ECS et Fargate

L'URL est :

(your-ID).dkr.ecr.ca-central-1.amazonaws.com/wordpress