

# Exo Aws

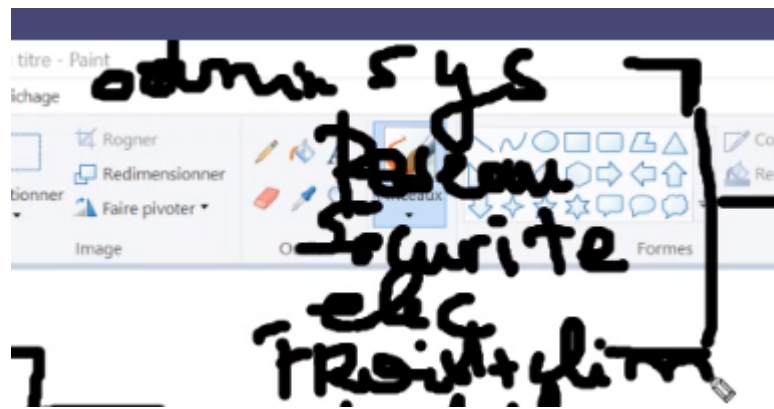
## Introduction

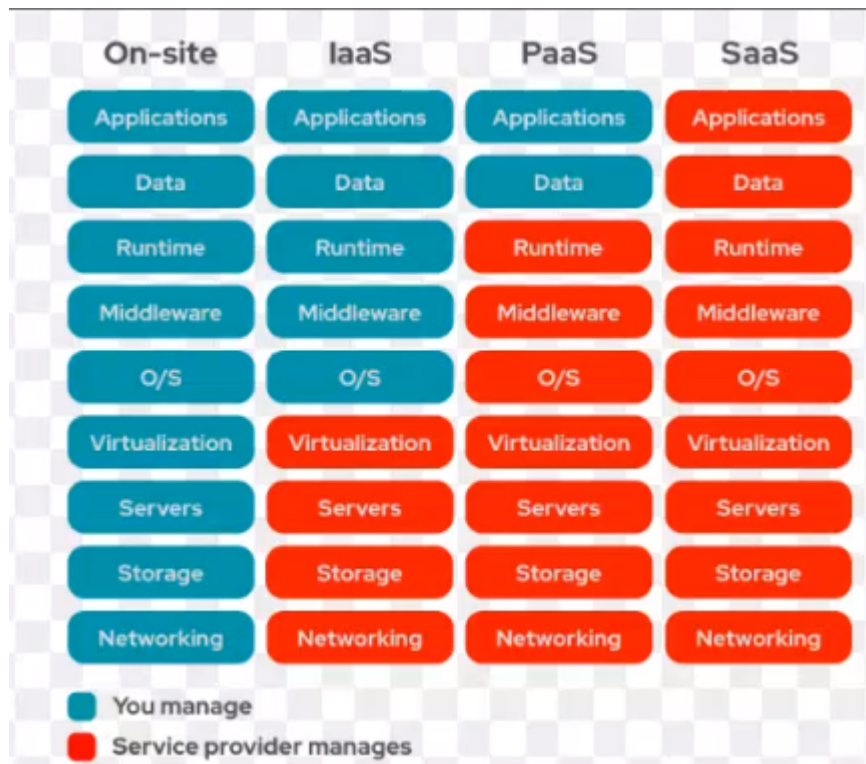


DNS Charge:

1- edf	5. Material
2. OVH	6. Load
3. BOX	7. Assurance
4. Admin	8. Sécurité

N.T.A





Connexion aws

ID de compte (12 chiffres) ou alias de compte

780076852474

Nom d'utilisateur :

antoine\_2120052

Mot de passe :

.....

creer son propre compte avec sa cb pour voir comment ca marche d'avoir un compte root

## EC2

### Création d'instance

on a touché à rien jusqu'au tag on a gardé tout ce qui est gratuit

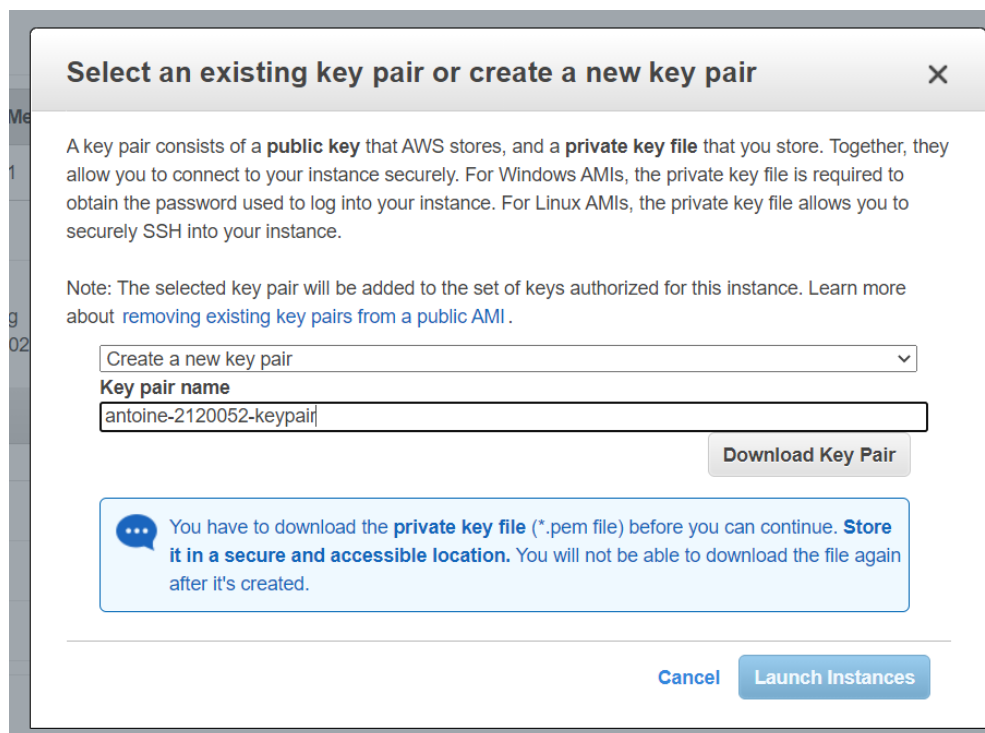
mettre tag **Name** forcément antoine-2120052-server-linux

nom du security group : antoine-2120052-server-sg

source 0.0.0.0/0 → tout le monde a le droit de venir sur ma machine

antoine-2120052-keypair-ohio clef ohio

antoine-2120052-keypair la clef virginie



The screenshot shows a modal dialog box titled "Select an existing key pair or create a new key pair" with a close button (X) in the top right corner. The dialog contains the following elements:

- Text:** "A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance."
- Note:** "Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#)."
- Form:** A dropdown menu with the text "Create a new key pair" and a downward arrow. Below it is a text input field labeled "Key pair name" containing the text "antoine-2120052-keypair".
- Buttons:** A "Download Key Pair" button is located to the right of the "Key pair name" field.
- Message Box:** A light blue box with a speech bubble icon containing the text: "You have to download the **private key file** (\*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created."
- Footer:** At the bottom right, there are two buttons: "Cancel" and "Launch Instances".

ma clef est telecharger dans les telechargement

EC2 > Instances > i-044e1c1b0b9a9c5f1 > Connect to instance

## Connect to instance [Info](#)

Connect to your instance i-044e1c1b0b9a9c5f1 (antoine-2120052-server-linux) using any of these options

**EC2 Instance Connect**

Session Manager

SSH client

EC2 Serial Console

Instance ID

 i-044e1c1b0b9a9c5f1 (antoine-2120052-server-linux)


Public IP address

 18.116.100.140

User name


ec2-user


Connect using a custom user name, or use the default user name ec2-user for the AMI used to launch the instance.


 **Note:** In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.


on va sur mobaxterm pour se connecter en ssh et ec2-user en username

on ajoute la clef privé (j'étais en ohio)

 Advanced SSH settings


 Terminal settings

 Network settings

 Bookmark settings


☒ X11-Forwarding

☒ Compression


Remote environment: Interactive shell 

Execute command:


☐ Do not exit after command ends

SSH-browser type: SFTP protocol 

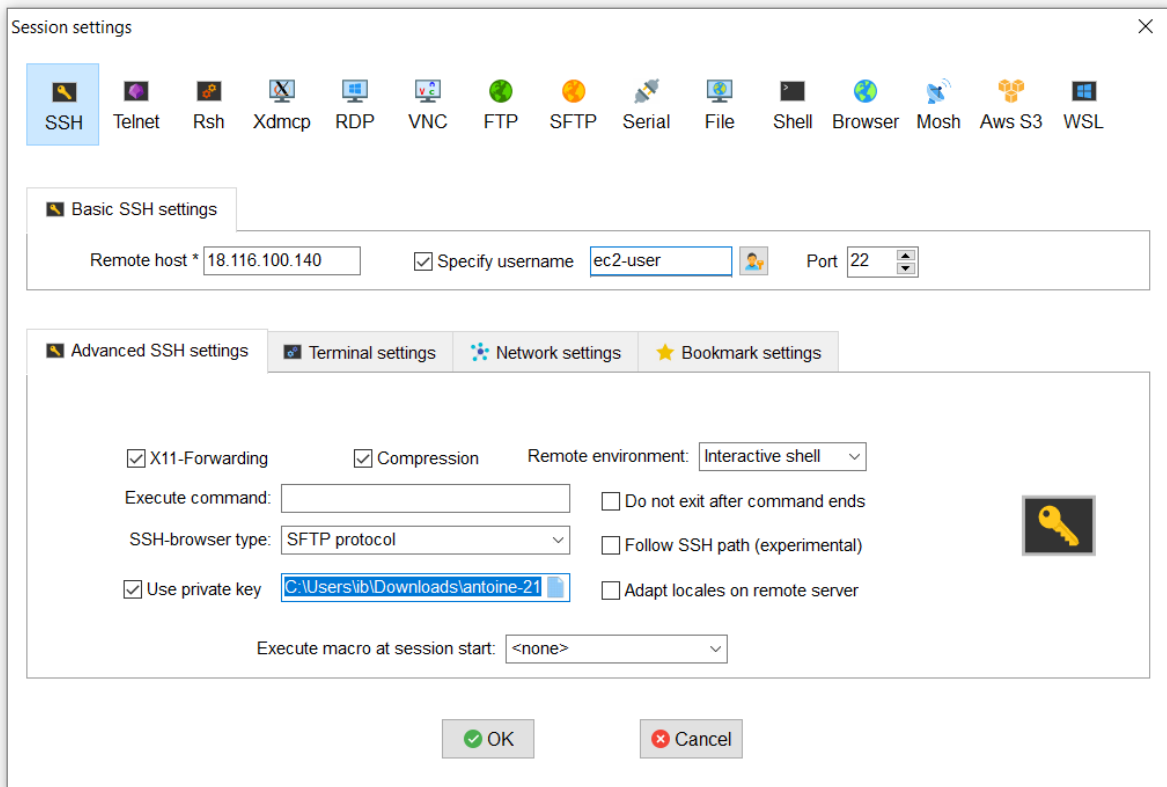
☐ Follow SSH path (experimental)

☒ Use private key C:\Users\lib\Downloads\antoine-21 

☐ Adapt locales on remote server

Execute macro at session start: <none> 





connexion via mobaxterm



on a des infos sur instance id

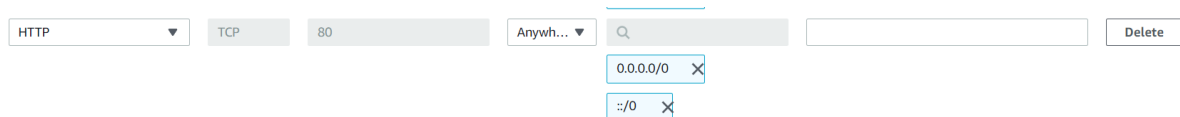
on va installer un serveur web maintenant  
sur amazon linux (viens de redhat)

```
sudo yum install -y httpd
sudo systemctl start httpd
curl 127.0.0.1
```

pour voir le site sur mobaxterm

pour voir le site sur le navigateur

aller sur le security groupe et ajouter une rule



premier site heberger en mode iaas pcq on ne consomme pas l'infra physique par contre on gère le reste

quand on utilise plus une ressource on les supprime

pour faire ca on sélectionne notre instance et on clique sur instance state et on choisit terminate instance

si on choisit stop instance elle n'est pas supprimer mais juste arreter donc on paie toujours  
bien penser que ca peut partir à tout moment et bien anticiper cela

## Exercice site perso:

Aws région virginie

créer une instance qui va respecter les normes mises en places et qui va utiliser la paire de clés créée vendredi idem pour le security group  
on va installer apache et modifier le fichier qui va bien pour que depuis chez moi je me connecte sur votre machine via son ip public j'ai un site internet qui me dit bonjour dirane site de Antoine

On refait comme au dessus

pour changer le contenu du site on va dans /var/www/html/index.html

## Exercice Windows:

serveur w2k19 et plus amazon (Microsoft Windows Server 2019 Base)

mettre en place une nouvelle sg faite pour le windows

je dois depuis la maison me connecter à votre instance

protocole pour se connecter sur un windows à distance

on lance l'instance de Microsoft Windows Server 2019 Base (la première dans la liste)  
tout pareil sauf au niveau du sg où on a un rdp à la place du ssh

rdp : remote desktop protocol

on clique sur notre instance puis sur connect et on télécharge ce qu'il nous demande après 4 min on peut faire get password et decrypter le password avec la clé privée

password pour cet exo : Do2-idPWCXX?\*pKL@L-9jKVzFo4AVqDi

The screenshot shows the AWS Management Console interface for connecting to an EC2 instance. The breadcrumb trail at the top reads: EC2 > Instances > i-01c48d59a1867e6f0 > Connect to instance. The main heading is 'Connect to instance' with an 'Info' link. Below it, a sub-header states: 'Connect to your instance i-01c48d59a1867e6f0 (antoine-2120052-server-windows) using any of these options'. There are three tabs: 'Session Manager' (highlighted in orange), 'RDP client' (selected with a blue underline), and 'EC2 Serial Console'. The 'RDP client' tab contains the following text: 'You can connect to your Windows instance using a remote desktop client of your choice, and by downloading and running the RDP shortcut file below:'. Below this is a button labeled 'Download remote desktop file'. Further down, it says 'When prompted, connect to your instance using the following details:'. This is followed by a table with two columns: 'Public DNS' and 'User name'. The 'Public DNS' row shows 'ec2-54-89-69-208.compute-1.amazonaws.com'. The 'User name' row shows 'Administrator'. Below this is a 'Password' row showing 'Do2-idPWCXX?\*pKL@L-9jKVzFo4AVqDi'. At the bottom, a note states: 'If you've joined your instance to a directory, you can use your directory credentials to connect to your instance.'

on clique sur le fichier qu'on a dl et on accepte de se connecter et après on rentre le mdp et tadaa

## Exercice GAAS (gaming as a service) (Mario) :

déployer une instance qui va utiliser du amazon linux et dessus on va déployer docker  
ensuite on va deployer le conteneur mario  
pour l'installation de docker on va regarder l'installation de docker sur la doc aws  
ne pas créer de key pair ni de sg

installation de docker sur aws :

<https://docs.aws.amazon.com/AmazonECS/latest/developerguide/docker-basics.html>

se deco reco pour que le changement de group soit effectif

déployer mario

```
docker run -d -p 80:8080 pengbai/docker-supermario
```

## Exercice waas :

heberger le site wordpress

deployer une amazon linux (pas de création de clef ni de sg)

puis docker

```
sudo systemctl enable docker
```

pour rendre docker enable au demarage

puis docker compose

<https://docs.docker.com/compose/install/>

dossier wordpress

creer docker-compose.yaml (j'ai pris celui de la doc ci dessous)

<https://docs.docker.com/samples/wordpress/>

et le lancer

```
docker-compose up -d
```

## Exercice création ami :

déployer autant d'instance que je veux avec wordpress à l'intérieur

la pour le moment on a le paramétrage initial de la machine il faudrait l'enlever

il faut permettre à l'user de fournir lui même ces infos

on commence par `docker-compose down -v`

on sélectionne notre instance

action > image and template > create image

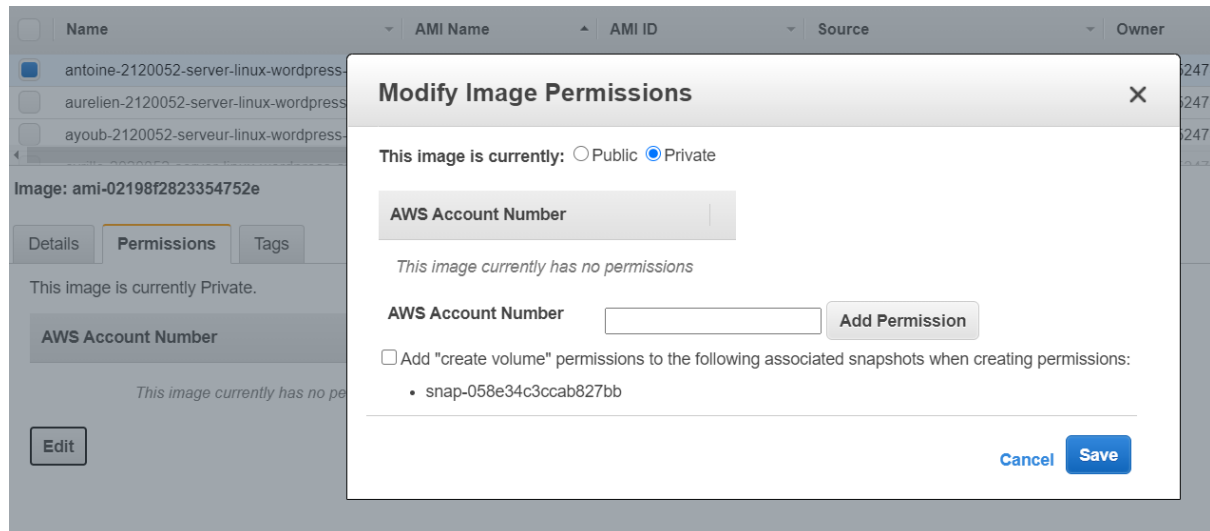


image name : antoine-2120052-server-linux-wordpress-ami  
ajouter un tag Name avec le nom de l'instance

(pour actualiser sur aws le bouton est en haut à droite)

maintenant pour créer une instance on aller dans myAmi lors de la création de l'instance

pour la partager avec d'autre gens on peut aller dans my image permissions



On termine l'instance wordpress qui etait en cours lorsque le build de notre image est ok.

on crée notre instance en la prenant dans myami  
on s'arrête sur la page

## Step 3: Configure Instance Details

en bas on modifie les user data et on fait du scripting

```
#!/bin/bash
```

```
docker-compose -f /home/ec2-user/wordpress/docker-compose.yml up -d
```

ca marche pas donc on peut aller voir les logs(clic droit > monitoring and troubleshoot > get system logs)

on peut se connecter à notre machine  
et aller dans le dossier cat /var/log/cloud-init/

pour notre problème on a ajouter

```
systemctl start docker
```

a user data lors de la création de l'instance

Nettoyage de ce qu'on a fait :  
d abord on termine l'instance  
puis deregister de l'AMI

puis on delete le snapshot

## Exercice création task avec EBS :

Elastic Container Service :

<https://console.aws.amazon.com/ecs/home?region=us-east-1#/getStarted>

dirane fait un cluster et nous on va faire des task dessus

on clique sur **task definition** a gauche

puis sur **Create new Task Definition** et on choisit **fargate**


ports they will use. [Learn more](#)

**Create new Task Definition**

Select launch type compatibility

Select which launch type you want your task definition to be compatible with based on where you want to launch your task.

**FARGATE**




Price based on task size

Requires network mode awsvpc

AWS-managed infrastructure, no Amazon EC2 instances to manage

**EC2**



Price based on resource usage

Multiple network modes available

Self-managed infrastructure using Amazon EC2 instances

**Task execution role** You are giving permission to Elastic Container Service to create and use ecsTaskExecutionRole. ⓘ

**Task size** ⓘ

The task size allows you to specify a fixed size for your task. Task size is required for tasks using the Fargate launch type and is optional for the EC2 launch type. Container level memory settings are optional when task size is set. Task size is not supported for Windows containers.

**Task memory (GB)** 2GB ⓘ  
The valid memory range for 1 vCPU is: 2GB - 8GB.

**Task CPU (vCPU)** 1 vCPU ⓘ  
The valid CPU range for 2GB memory is: 0.25 vCPU - 1 vCPU.

**Task memory maximum allocation for container memory reservation**  
0 2048 shared of 2048 MiB

**Task CPU maximum allocation for containers**  
0 1024 shared of 1024 CPU units

**Container Definitions** ⓘ

**Add container**

Container ...	Image	Hard/Soft ...	CPU Unit...	GPU	Inference A...	Essential ...
---------------	-------	---------------	-------------	-----	----------------	---------------

Add container

▼ Standard

**Container name\*** mysql ⓘ

**Image\*** mysql:5.7 ⓘ

**Private repository authentication\*** ☐ ⓘ

**Memory Limits (MiB)** Soft limit 128 ⓘ  
[+ Add Hard limit](#)  
Define hard and/or soft memory limits in MiB for your container. Hard and soft limits correspond to the 'memory' and 'memoryReservation' parameters, respectively, in task definitions. ECS recommends 300-500 MiB as a starting point for web applications.

**Port mappings** ⓘ

Container port	Protocol
	tcp

[+ Add port mapping](#)

Host port mappings are not valid when the network mode for a task definition is host or awsvpc. To specify different host and container port mappings, choose the Bridge network

## Environment variables

You may also designate AWS Systems Manager Parameter Store keys or ARNs using the 'valueFrom' field. ECS will inject the value in containers at run-time.

Key

MYSQL_ROOT_PASSWORD	Value	somewordpress	×
MYSQL_DATABASE	Value	wordpress	×
MYSQL_USER	Value	wordpress	×
MYSQL_PASSWORD	Value	wordpress	×
Add key	Value	Add value	

ensuite on ajoute wordpress

▼ Standard

**Container name\***  ⓘ

**Image\***  ⓘ

**Private repository authentication\*** ☐ ⓘ

**Memory Limits (MiB)** Soft limit ▼  ⓘ

[+ Add Hard limit](#)

Define hard and/or soft memory limits in MiB for your container. Hard and soft limits correspond to the `memory` and `memoryReservation` parameters, respectively, in task definitions

on met le port 80 pour wordpress

Environment variables

You may also designate AWS Systems Manager Parameter Store keys or ARNs using the 'valueFrom' field. ECS will inject the value into containers at run-time.

Key

<input type="text" value="WORDPRESS_DB_HOST"/>	<span>Value ▼</span>	<input type="text" value="127.0.0.1:3306"/>	✖
<input type="text" value="WORDPRESS_DB_USER"/>	<span>Value ▼</span>	<input type="text" value="wordpress"/>	✖
<input type="text" value="WORDPRESS_DB_PASSWORD"/>	<span>Value ▼</span>	<input type="text" value="wordpress"/>	✖
<input type="text" value="WORDPRESS_DB_NAME"/>	<span>Value ▼</span>	<input type="text" value="wordpress"/>	✖
<input type="text" value="Add key"/>	<span>Value ▼</span>	<input type="text" value="Add value"/>	

Container N...	Image	Hard/Soft ...	CPU Unit...	GPU	Inference A...	Essential ...	
 mysql	mysql:5.7	--/--				true	✖
 wordpr...	wordpress:la...	--/--				true	✖

on peut créer une nouvelle révision (les task sont versionnées)

aller sur notre task cliquez sur action>run task  
on choisit fargate et on s'assure que le cluster est le bon

## Run Task

Select the cluster to run your task definition on and the number of copies of that task to run. To apply c

**Launch type** ☒ FARGATE ☐ EC2 ?

[Switch to capacity provider strategy](#) ?

**Task Definition** Family  
antoine-2120052-wordpress-task ▼

Revision  
2 ▼

**Platform version** LATEST ▼ ?

**Cluster** dirane-2120052-ecs-fargate ▼

**Number of tasks** 1

on ajoute un vpc et deux sous réseaux :

### VPC and security groups

VPC and security groups are configurable when your task definition uses the awsvpc network mode.

**Cluster VPC\*** vpc-87cc41fa (172.31.0.0/16) ?

**Subnets\*** ?

- subnet-a7a42196  
(172.31.48.0/20) - us-east-1e  
assign ipv6 on creation: Disabled
- subnet-8e2351d1  
(172.31.32.0/20) - us-east-1a  
assign ipv6 on creation: Disabled

▼

pour récupérer l'adresse ip on va dans cluster on clique sur le cluster de dirane puis sur task et on clique sur notre task pour avoir les infos

**Desired status** RUNNING

**Created at** 2021-04-26 14:42:14 +0200

**Started at** 2021-04-26 14:43:02 +0200

## Network

**Network mode** awsvpc

**ENI Id** eni-0a210cc92a95fc385

**Subnet Id** subnet-a7a42196

**Private IP** 172.31.63.9

**Public IP** 54.160.82.139

**Mac address** 06:2d:40:87:3b:89

# Storage

## Exercice persistance des données :

le client ne veut pas que la bdd soit stocké dans un volume de type volumes mais dans un volumes de type bind mount qui se trouverait dans /data/

la ligne `db_data:/var/lib/mysql`

par `/data:/var/lib/mysql` et supprimer le volume en bas

créer une ec2 qui va utiliser amazon linux avec docker docker compose et un fichier wordpress avec le docker compose qu'on vient d'écrire

on vire tout ce qu'il y a dans /data

```
sudo rm -rf /data/*
```

et après on va faire un ebs qui se trouve à gauche dans le menu ec2

on clique sur volume

condition pour attacher un volume à une machine est qu'il soit dans la même zone de disponibilité

la je suis dans us-east-1c

création du volume

The screenshot shows the 'Create Volume' page in the AWS Management Console. The 'Volume Type' is set to 'General Purpose SSD (gp2)'. The 'Size (GiB)' is set to 1. The 'IOPS' are set to 100 / 3000. The 'Throughput (MB/s)' is set to 'Not applicable'. The 'Availability Zone' is set to 'us-east-1c'. The 'Snapshot ID' is set to 'Select a snapshot'. The 'Encryption' checkbox is unchecked. At the bottom, there is a table with two columns: 'Key' and 'Value'. The 'Key' is 'Name' and the 'Value' is 'antoine-2120052-server-linux-wordpress-data-ebs'.

Key	Value
Name	antoine-2120052-server-linux-wordpress-data-ebs

sélectionner son volume puis cliquer sur action et choisir **attach volume**

Attach Volume

Volume

vol-0191918bd87752b14 (antoine-2120052-server-linux-wordpress-data-ebs) in us-east-1c

Instance

i-0ebb020b482620c97 in us-east-1c

Device

/dev/sdf

Linux Devices: /dev/sdf through /dev/sdp

Note: Newer Linux kernels may rename your devices to /dev/xvdf through /dev/xvdp internally, even when the device name entered here (and shown in the details) is /dev/sdf through /dev/sdp.

Cancel

Attach

commande pour voir le disque reconnu par la machine :

```
lsblk
```

commande permettant de formater le disque

```
sudo mkfs -t xfs /dev/xvdf
```

commande permettant de monter le disque

```
sudo mount /dev/xvdf /data
```

et pour vérifier

```
mount
```

on refait le up

on va simuler un accident

on va terminer notre instance

on recréer une machine et on lui attache le ebs

mettre la nouvelle machine dans le même sous réseau

Subnet subnet-25c9b504 | Default in us-east-1c [Create new subnet](#)

on attache notre volume comme tout à l'heure mais on fait pas le formatage on recréer /data

et on réinstalle tout

bon après il nous renvoie sur l'ancienne ip

Bonus :

on va dans /tmp

on exporte la bdd dans un fichier

```
docker exec wordpress_db_1 /usr/bin/mysqldump -u root
--password=somewordpress wordpress > backup.sql
```

on affiche backup et on voit en bas qu'il y a l'adresse public de l'ancienne machine

on fait un grep pour trouver les endroits où est mon ancienne adresse ip dans le fichier

on utilise sed pour remplacer des chaînes dans un fichier avec l'option -i

ancienne ip : 54.152.186.225

nouvelle ip : 3.88.133.78

```
docker exec -it wordpress_db_1 /bin/bash
```

en prod on travaille avec des noms de domaines comme ça si on change de machine pas de problèmes

ensuite on recree la base de donnee

on se connecte à notre bdd

```
docker exec -it wordpress_db_1 /bin/bash
```

```
mysql -uroot -psomewordpress
```

d'abord on la supprime pour éviter les conflits

```
SHOW DATABASES;  
DROP DATABASE wordpress;  
SHOW DATABASES;
```

et on en recréer une

```
CREATE DATABASE wordpress;
```

puis ctrl D puis exit

```
cd /tmp
```

on rerepli la bdd

```
cat backup.sql | docker exec -i wordpress_db_1 /usr/bin/mysql -u root  
--password=somewordpress wordpress
```

## Exercice S3 :

Partage de moments avec la famille

création d'un bucket

nom : antoine-2120052-holidays-media

a l'intérieur 2 dossier un images et un videos

et apres donner l'url

on creer le bucket

on creer nos fichier on ajoute ce qu'on veut

on choisit notre fichier et on clique sur actions > make public

maintenant creer une vm amazon linux



on va installer apache  
créer un index avec des balise pointant sur notre image et notre photo

exercice s3 site internet :  
héberger un site internet sur aws sur un s3

il faut que ca garde l'arborescence  
dl le code du client

<https://github.com/diranetafen/static-website-example/archive/refs/heads/master.zip>

on crée notre bucket  
on met public  
on va dans properties et en bas on edit static website hosting  
et pour tout rendre public on selectionne tout et on clique sur actions > Make public

## VPC

### Exercice VPC

VPC (10.0.0.0/16)

sous reseaux public 10.0.0.0/24 (lies au IGW (internet gateway)  
et ensuite on va y déployer une vm avec un apache dessus

on fait la création assistée

on choisit la config qui se rapproche le plus de ce qu'on veut (ici la premiere)

Step 2: VPC with a Single Public Subnet

The screenshot shows the 'Create VPC' wizard in the AWS Management Console, specifically Step 2: VPC with a Single Public Subnet. The form is divided into several sections:

- IPv4 CIDR block:** A text input field containing '10.0.0.0/16' with a note '(65531 IP addresses available)'.
- IPv6 CIDR block:** Three radio button options: 'No IPv6 CIDR Block' (selected), 'Amazon provided IPv6 CIDR block', and 'IPv6 CIDR block owned by me'.
- VPC name:** A text input field containing 'antoine-2120052-vpc'.
- Public subnet's IPv4 CIDR:** A text input field containing '10.0.0.0/24' with a note '(251 IP addresses available)'.
- Availability Zone:** A dropdown menu set to 'No Preference'.
- Subnet name:** A text input field containing 'antoine-2120052-vpc-publiq'.
- Service endpoints:** A section with an 'Add Endpoint' button.
- Enable DNS hostnames:** Two radio button options: 'Yes' (selected) and 'No'.
- Hardware tenancy:** A dropdown menu set to 'Default'.

Below the 'Subnet name' field, there is a note: 'You can add more subnets after AWS creates the VPC.'

on verifie la plage et on indique les noms  
il le provisionne de maniere automatique

ensuite on deploie notre machine et on installe apache dessus

faire un deuxième sous réseau dans une autre zone de dispo que le précédent

### Subnet settings

Specify the CIDR blocks and Availability Zone for the subnet.

**Subnet 1 of 1**

Subnet name  
Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Availability Zone [Info](#)  
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

IPv4 CIDR block [Info](#)

▼ Tags - optional

Key

Value - optional

Remove

Add new tag

You can add 49 more tags.

Remove

on clique ensuite sur notre sous réseau et on clique sur actions > Modify auto-assign IP settings et on clique sur enable (on le fait pour nos deux sous réseaux)

mais notre nouveau sous réseaux n'est pas lié à IGW, (on peut le voir dans la table route de notre premier sous réseaux)

on prend l'id de la routing table de notre premier reseaux et on l'utilise pour modifier la table de routage

sinon on peut cliquer sur routes a gauche choisir l'id de la table de routage a modifier et ajouter la nouvelle route

Maintenant, il faut mettre en place un load balancer

on va dans la partie load balancer de ec2 ensuite on cree un load balancer de type http https nom : antoine-2120052-vpc-elb

## Step 1: Configure Load Balancer

### Basic Configuration

To configure your load balancer, provide a name, select a scheme, specify one or more listeners, and select a network. Th traffic on port 80.

Name ⓘ

Scheme ⓘ

☒ internet-facing
☐ internal

IP address type ⓘ

### Listeners

## Availability Zones

Specify the Availability Zones to enable for your load balancer. The load balancer routes traffic to the targets in the Availability Zones to increase the availability of your load balancer.

**VPC** ⓘ  ⌵

**Availability Zones** ☒ **us-east-2b**  ⌵

**IPv4 address** ⓘ Assigned by AWS

☒ **us-east-2c**  ⌵

**IPv4 address** ⓘ Assigned by AWS

antoine-2120052-vpc-tg

on cree notre target group (la ou le lb va dispatcher les gens)

### Registered targets

To deregister instances, select one or more registered instances and then click Remove.

<input type="checkbox"/>	Instance	Name	Port	State	Security groups	Zone
<input type="checkbox"/>	i-0ad4c64cb97811ee	antoine-2120052-server-lin...	80	running	antoine-2120052-vpc-ohio-sg	us-east-2b
<input type="checkbox"/>	i-050f8e0428d60e58a	antoine-2120052-server-linux	80	running	antoine-2120052-vpc-ohio-sg	us-east-2c

### Instances

To register additional instances, select one or more running instances, specify a port, and then click Add. The default port is the port specified for the target group. If the instance is already registered on the specified port, you must specify a different port.

**Add to registered** on port

<input type="checkbox"/>	Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR
<input checked="" type="checkbox"/>	i-0ad4c64cb97811ee	antoine-2120052-serv...	running	antoine-2120052-vpc-...	us-east-2b	subnet-0f767eacc75e690a7	10.0.1.0/24
<input checked="" type="checkbox"/>	i-050f8e0428d60e58a	antoine-2120052-serv...	running	antoine-2120052-vpc-...	us-east-2c	subnet-06e36dc1d569393d4	10.0.0.0/24

et on register nos instances dans le target group

une fois que le lb est prêt on a l'adresse ip permettant d'y accéder

**DNS name** antoine-2120052-vpc-elb-1990642133.us-east-2.elb.amazonaws.com ⓘ  
(A Record)

## Devops:

## CodeCommit

sur aws ecs amazon linux

git cloner le code du client

dans un dossier webapp

dockerfile qui va permettre de build l'app du client

et deployer l'app pour verifier que ca fonctionne  
webapp:latest

une fois deployer on met tout sur docker hub

```
docker tag a997839e16e2 antoinebouquet1010/webapp
docker push antoinebouquet1010/webapp
```

on supprime le fichier .git  
rm -Rf static-website-example/.git

on va sur codecommit  
on créer un repository

### Repository settings

Repository name

antoine-2120052-webapp

100 characters maximum. Other limits apply.

Description - *optional*

1,000 characters maximum

Tags

Add

☐ Enable Amazon CodeGuru Reviewer for Java and Python - *optional*

Get recommendations to improve the quality of the Java and Python code for all pull requests in this repository.

A service-linked role will be created in IAM on your behalf if it does not exist.

#### Step 1: Prerequisites

You must use a Git client that supports Git version 1.7.9 or later to connect to an AWS CodeCommit repository. If you do not have a Git client, you can install one from [Git downloads](#). [View Git downloads page](#)

You must have an AWS CodeCommit managed policy attached to your IAM user, belong to a CodeStar project team, or have the equivalent permissions. [Learn how to create and configure an IAM user for accessing AWS CodeCommit.](#) | [Learn how to add team members to an AWS CodeStar Project.](#)

#### Step 2: Git credentials

Create Git credentials for your IAM user, if you do not already have them. Download the credentials and save them in a secure location. [Generate Git Credentials](#)

#### Step 3: Clone the repository

Clone your repository to your local computer and start working on code. Run the following command:

git clone https://git-codecommit.us-east-1.amazonaws.com/v1/repos/antoine-2120052-webapp

Copy

#### Additional details

You can find more detailed instructions in the documentation. [View documentation](#)

les credentials

au niveau d'un user :

security credentials > https git credentials for aws codecommit

antoine\_2120052-at-780076852474  
LbaEfVco9sXQ2ihB1LRDUT+vjfN6ays7n/DOqtMh94M=

et on push ce qu'on a sur code commit (meme maniere que sur git)

# CodeBuild

mettre en place la partie intégration continue

```
#login
docker login -u $DOCKERHUB_LOGIN -p $DOCKERHUB_PASSWORD
#build
docker build -t $DOCKERHUB_LOGIN/$IMAGE_REPO_NAME:$IMAGE_TAG .
#push vers le registry
docker push $DOCKERHUB_LOGIN/$IMAGE_REPO_NAME:$IMAGE_TAG
```

```
export DOCKERHUB_LOGIN=login
```

pour externaliser les variables

commande de base pour login build push

produire un fichier buildspec.yml qui aura dans la section prebuild le login

la partie build le build et la partie post build le push

```
version: 0.2

phases:
  pre_build:
    commands:
      - echo Log in DockerHub...
      - docker login -u $DOCKERHUB_LOGIN -p $DOCKERHUB_PASSWORD
  build:
    commands:
      - echo Build started on `date`
      - echo Building the Docker image...
      - docker build -t $DOCKERHUB_LOGIN/$IMAGE_REPO_NAME:$IMAGE_TAG .
  post_build:
    commands:
      - echo Build completed on `date`
      - echo Pushing the Docker image...
      - docker push $DOCKERHUB_LOGIN/$IMAGE_REPO_NAME:$IMAGE_TAG
```

ensuite on repush sur codecommit

ensuite codebuild > build project > create project build

## Project configuration

Project name

antoine-2120052-build

A project name must be 2 to 255 characters. It can include the letters A-Z and a-z, the numbers 0-9, and the special characters - and \_.

Description - *optional*

Build badge - *optional*

☒ Enable build badge

## Source

Add source

Source 1 - Primary

Source provider

AWS CodeCommit

Repository

Q antoine-2120052-webapp

Reference type

Choose the source version reference type that contains your source code.

- ☒ Branch
- ☐ Git tag
- ☐ Commit ID

Branch

Choose a branch that contains the code to build.

master

Commit ID - *optional*

Choose a commit ID. This can shorten the duration of your build.

Q

Source version [Info](#)

refs/heads/master

eb9c67b4 merge

Environnement

#### Environment image



##### Managed image

Use an image managed by AWS CodeBuild



##### Custom image

Specify a Docker image

#### Operating system

Ubuntu



The programming language runtimes are now included in the standard image of Ubuntu 18.04, which is recommended for new CodeBuild projects created in the console. See [Docker Images Provided by CodeBuild](#) for details [\[2\]](#).

#### Runtime(s)

Standard



#### Image

aws/codebuild/standard:5.0



#### Image version

Always use the latest image for this runtime version



#### Environment type

Linux



#### Privileged

☒ Enable this flag if you want to build Docker images or want your builds to get elevated privileges

#### Privileged

☒ Enable this flag if you want to build Docker images or want your builds to get elevated privileges

#### Service role



##### New service role

Create a service role in your account



##### Existing service role

Choose an existing service role from your account

#### Role ARN

arn:aws:iam::780076852474:role/service-role/codebuild-dirane-2120052-codi X

☒ Allow AWS CodeBuild to modify this service role so it can be used with this build project

#### ► Additional configuration

Timeout, certificate, VPC, compute type, environment variables, file systems

**Additional configuration**  
Timeout, certificate, VPC, compute type, environment variables, file systems

**Timeout**  
Default timeout is 1 hour

Hours:  Minutes:   
Timeout must be between 5 minutes and 8 hours

**Queued timeout**  
Default time in build queue is 8 hours

Hours:  Minutes:   
Timeout must be between 5 minutes and 8 hours

**Certificate**  
If you have a self-signed certificate or a certificate signed by a certification authority, choose the option to install it from your S3 bucket.

☒ Do not install any certificate ☐ Install certificate from your S3 bucket

**VPC**  
Select a VPC that your AWS CodeBuild project will access.

**Compute**

☒ 3 GB memory, 2 vCPUs  
☐ 7 GB memory, 4 vCPUs  
☐ 15 GB memory, 8 vCPUs  
☐ 145 GB memory, 72 vCPUs

on va dans secret manager pour stocker nos login et mdp :

**Select secret type** [Info](#)

☐ Credentials for RDS database ☐ Credentials for DocumentDB database ☐ Credentials for Redshift cluster

☐ Credentials for other database ☒ Other type of secrets (e.g. API key)

**Specify the key/value pairs to be stored in this secret** [Info](#)

**Secret key/value** | Plaintext

login	antoinebouquet1010	Remove
password		Remove

[+ Add row](#)



## Secret name and description [Info](#)

### Secret name

Give the secret a name that enables you to find and manage it easily.

antoine-2120052-dockerhub-secret

Secret name must contain only alphanumeric characters and the characters /\_+@-

### Description - optional

*e.g - Access to MYSQL prod database for my AppBeta*

Maximum 250 characters

puis suivant>suivant>store

DOCKERHUB\_LOGIN

antoine-2120052-docke

Secrets Manager ▼

Remove

DOCKERHUB\_PASSWOR

antoine-2120052-docke

Secrets Manager ▼

Remove

ajout de nos variable secret

nomdusecret:variable

antoine-2120052-dockerhub-secret:login

antoine-2120052-dockerhub-secret:password

## Buildspec

### Build specifications



Use a buildspec file

Store build commands in a YAML-formatted buildspec file



Insert build commands

Store build commands as build project configuration

### Buildspec name - optional

By default, CodeBuild looks for a file named buildspec.yml in the source code root directory. If your buildspec file uses a different name or location, enter its path from the source root here (for example, buildspec-two.yml or configuration/buildspec.yml).

le buildspec est a la racine de notre repo donc on laisse ca  
apres on créé

si ca marche pas a cause du role

on peut creer un role de nom :

codebuild-antoine-2120052-codebuild-service-role

On a créé notre task maintenant il faut la lancer (appuyer sur start build)

il a permis a codebuild de lire les secrets

dans phase details on a les détails de ce qu'il se passe

moi j'avais pas mis les bons noms de phases  
après je recommit et je restart le build  
et la ca fonctionne

ensuite on va déployer son app avec ecs pour déployer son image  
nom webapp  
image antoinebouquet1010/webapp:latest

Maintenant on veut mettre en place la partie déploiement depuis ecs  
Mise en place d'un load balancer et de replica

on va transformer notre task en service et on va créer un elb

d'abord l'elb

on prend notre sg et on décoche celui coché par défaut

#### Step 4: Configure Routing

Target group ⓘ New target group

Name ⓘ antoine-2120052-webapp

Target type

- ☐ Instance
- ☒ IP
- ☐ Lambda function

Protocol ⓘ HTTP


Port ⓘ 80

Protocol version ⓘ

- ☒ HTTP1  
Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.
- ☐ HTTP2  
Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.
- ☐ gRPC  
Send requests to targets using gRPC. Supported when the request protocol is gRPC.

cette fois on a des target de types ip  
et apres on ne register pas de cible

maintenant sur ecs on va créer un service qui va créer les tasks

Launch type ☒ FARGATE ☐ EC2 

[Switch to capacity provider strategy](#) 


**Task Definition** Family


antoine-2120052-webapp-task ▼

Enter a value


Revision


1 (latest) ▼

**Platform version** LATEST ▼ 

**Cluster** dirane-2120052-cluster-webapp ▼ 

**Service name** antoine-2120052-webapp-service 


**Service type\*** REPLICAS 


**Number of tasks** 1 


deploiement sur blue green


**Cluster VPC\*** vpc-87cc41fa (172.31.0.0/16) 

**Subnets\***

subnet-837273ce  
(172.31.16.0/20) - us-east-1d  
assign ipv6 on creation: Disabled 

subnet-75c9bf13  
(172.31.0.0/20) - us-east-1b  
assign ipv6 on creation: Disabled 





**Security groups\*** antoin-4211  

**Auto-assign public IP** ENABLED 


#### Load balancing

An Elastic Load Balancing load balancer distributes incoming traffic across the tasks running in your service. Choose an existing load balancer, or create a new one in the [Amazon EC2 console](#).

**Load balancer type\***

- ☒ **Application Load Balancer**  
Allows containers to use dynamic host port mapping (multiple tasks allowed per container instance). Multiple services can use the same listener port on a single load balancer with rule-based routing and paths.
- ☐ **Network Load Balancer**  
A Network Load Balancer functions at the fourth layer of the Open Systems Interconnection (OSI) model. After the load balancer receives a request, it selects a target from the target group for the default rule using a flow hash routing algorithm.

**Service IAM role** Task definitions that use the awsvpc network mode use the AWSServiceRoleForECS service-linked role, which is created for you automatically. [Learn more](#).

**Load balancer name** antoine-2120052-webapp-elb 

la on a cliqué sur le bouton add to load balancer

### ▼ Additional configuration

To facilitate blue/green deployments with AWS CodeDeploy, you need two target groups. Each target group binds to a separate task set in the deployment. [Learn more](#)

Target group 1 name\*   ⓘ

Target group 1 protocol\*

Target type\*  ⓘ

Path pattern\*  Evaluation order

**Path pattern:** The first path pattern for a listener is the default path (/), which accepts all traffic that does not match another rule. You can later add additional patterns and priority values to this listener for other services.

**Evaluation order:** Rules are evaluated in priority order, from the lowest value to the highest value.

Once a path pattern rule is matched, all other rules are ignored. You can route traffic from this listener to multiple services by creating a path for each service.

Path pattern\*  Evaluation order

**Path pattern:** The first path pattern for a listener is the default path (/), which accepts all traffic that does not match another rule. You can later add additional patterns and priority values to this listener for other services.

**Evaluation order:** Rules are evaluated in priority order, from the lowest value to the highest value.

Once a path pattern rule is matched, all other rules are ignored. You can route traffic from this listener to multiple services by creating a path for each service.

### Existing paths in use on this listener

The path must include all the possible paths that your application uses, for example a service with the path /webapp1\* will receive traffic sent to /webapp1 and /webapp1/page.html on this listener. We recommend choosing unique paths, and a lower evaluation order enables you to route traffic between multiple conflicting paths.

Evaluation Order	Rule Path	Target Group
default	/	antoine-2120052-webapp

Health check path\*  ⓘ

Additional health check options can be configured in the ELB console after you create your service.

c'est pas / mais /\* qu'il faut mettre

Target group 2 name\*   ⓘ

Target group 2 protocol\*

Target type\*  ⓘ

Path pattern\*

Path pattern: The first path pattern for a listener is the default path (/), which accepts all traffic that does not match another rule. You can later add additional patterns and priority values to this listener for other services.

Health check path\*  ⓘ

Additional health check options can be configured in the ELB console after you create your service.

le site es pas beau mais il faudrait faire des paramétrages de loadbalancer

Mise en place du déploiement continue en utilisant le 2eme target group

## CodeDeploy

on va sur codedeploy

et on clique sur applications puis sur l'application que l'on a créé (ici notre service ECS)

on crée un déploiement

### Deployment settings

Application  
AppECS-dirane-2120052-cluster-webapp-antoine-2120052-webapp-service

Deployment group

Compute platform  
Amazon ECS

Deployment type  
Blue/green

Revision type  
☐ My application is stored in Amazon S3 ☒ Use AppSpec editor

AppSpec language  
☐ JSON ☒ YAML

Code expliquant comment on va deployer

```
version: 0.0
Resources:
  - TargetService:
      Type: AWS::ECS::Service
      Properties:
        TaskDefinition:
          "arn:aws:ecs:us-east-1:780076852474:task-definition/antoine-2120052-webapp-task:1"
          LoadBalancerInfo:
            ContainerName: "webapp"
            ContainerPort: 80
            PlatformVersion: "LATEST"
```

on a fait des modif sur le nom de la task definition et le container port

## CodePipeline:

on va dans codepipeline > pipeline > create pipeline

sur notre machine on crée un fichier appspec.yml (indication sur la mécanique de deployment)

avec le code d'au dessus dedans

et un fichier taskdef.json (permet de donner des infos sur la taskdef a deployer)

on va dans notre task definitions on clique sur json et on copie tout le json

on push

puis on revient sur codepipeline

## Pipeline settings

### Pipeline name

Enter the pipeline name. You cannot edit the pipeline name after it is created.

No more than 100 characters

### Service role

☐ New service role  
Create a service role in your account

☒ Existing service role  
Choose an existing service role from your account

### Role ARN

## ► Advanced settings

### Source provider

This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.

### Repository name

Choose a repository that you have already created where you have pushed your source code.

### Branch name

Choose a branch of the repository

### Change detection options

Choose a detection mode to automatically start your pipeline when a change occurs in the source code.

☒ Amazon CloudWatch Events (recommended)  
Use Amazon CloudWatch Events to automatically start my pipeline when a change occurs

☐ AWS CodePipeline  
Use AWS CodePipeline to check periodically for changes

### Output artifact format

Choose the output artifact format.

☒ CodePipeline default  
AWS CodePipeline uses the default zip format for artifacts in the pipeline. Does not include git metadata about the repository.

☐ Full clone  
AWS CodePipeline passes metadata about the repository that allows subsequent actions to do a full git clone. Only supported for AWS CodeBuild actions.



## Build - *optional*

### Build provider

This is the tool of your build project. Provide build artifact details like operating system, build spec file, and output file names.

AWS CodeBuild

### Region

US East (N. Virginia)

### Project name

Choose a build project that you have already created in the AWS CodeBuild console. Or create a build project in the AWS CodeBuild console and then return to this task.

Q antoine-2120052-build



or

Create project

### Environment variables - *optional*

Choose the key, value, and type for your CodeBuild environment variables. In the value field, you can reference variables generated by CodePipeline. [Learn more](#)

Add environment variable

### Build type



Single build

Triggers a single build.



Batch build

Triggers multiple builds as a single execution.

Amazon ECS (Blue/Green)



### Region

US East (N. Virginia)



### AWS CodeDeploy application name

Choose one of your existing applications, or create a new one in AWS CodeDeploy.

Q AppECS-dirane-2120052-cluster-webapp-antoine



Create application

### AWS CodeDeploy deployment group

Choose one of your existing deployment groups, or create a new one in AWS CodeDeploy.

Q DgpECS-dirane-2120052-cluster-webapp-antoine-2120052-webapp-service



#### Amazon ECS task definition

Choose the input artifact where your Amazon ECS task definition file is stored. If other than the default file path, specify the path and filename of your task definition file.

BuildArtifact ▼

taskdef.json

The default path is taskdef.json.

#### AWS CodeDeploy AppSpec file

Choose the input artifact where your AWS CodeDeploy AppSpec file is stored. If other than the default file path, specify the path and filename of your AppSpec file.

BuildArtifact ▼

appspec.yaml

#### Dynamically update task definition image - optional

You can provide an input artifact and a placeholder name for the container definition image that will be used to dynamically update a task definition. You can specify multiple input artifacts and placeholders.

on stop direct le pipeline

on fait un edit (en haut)

et on va dans la partie deploy et on fait modifier

on modifie ca

#### Input artifacts

Choose an input artifact for this action. [Learn more](#)

SourceArtifact ▼

Add

No more than 100 characters

et ca :

#### AWS CodeDeploy deployment group

Choose one of your existing deployment groups, or create a new one in AWS CodeDeploy.

Q DgpECS-dirane-2120052-cluster-webapp-antoine-2120052-webapp-service X

#### Amazon ECS task definition

Choose the input artifact where your Amazon ECS task definition file is stored. If other than the default file path, specify the path and filename of your task definition file.

SourceArtifact ▼

taskdef.json

The default path is taskdef.json.

#### AWS CodeDeploy AppSpec file

Choose the input artifact where your AWS CodeDeploy AppSpec file is stored. If other than the default file path, specify the path and filename of your AppSpec file.

SourceArtifact ▼

appspec.yml

ensuite on met done partout puis on save

maintenant on va tenter de déclencher notre pipeline en faisant un pipeline

et ca marche il a dit comment voir ce qu'il est en train de faire mais j'ai pas vu

(encore une erreur sur deploy) il faut qu'on termine nos déploiement de tout à l'heure avec code deploy

on retourne sur notre pipeline et on clique sur release change