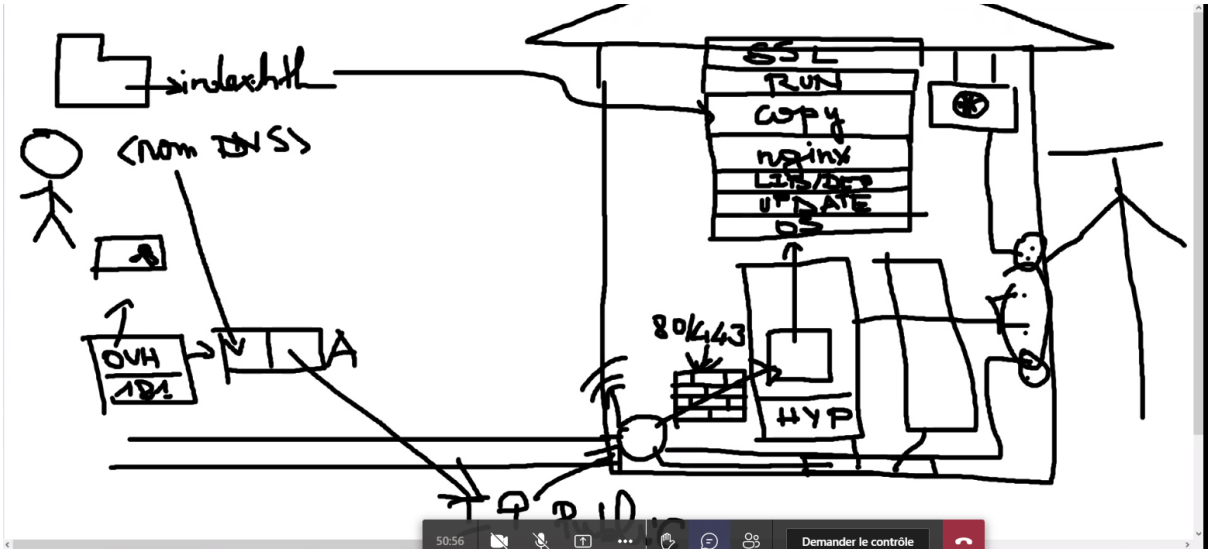
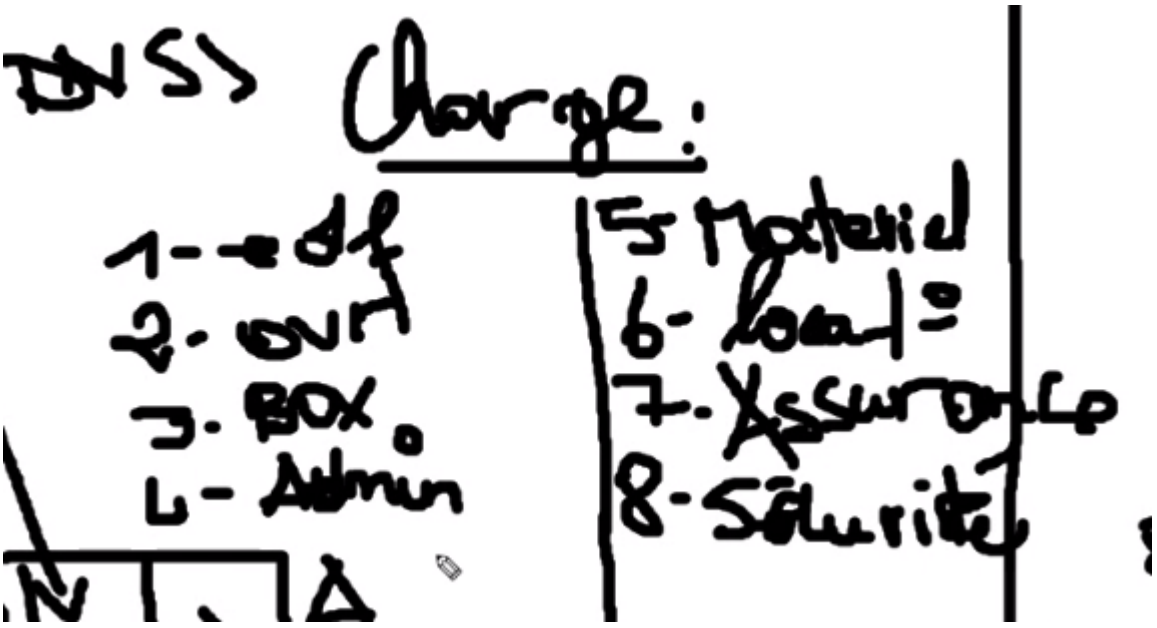
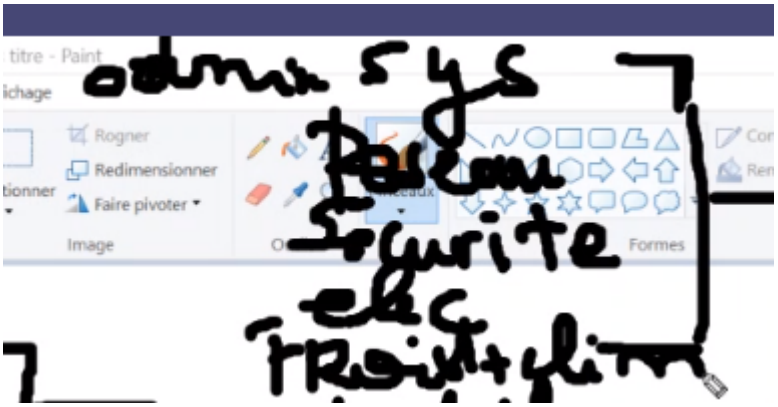
Exo Aws

Introduction






titre - Paint
Rogner
Redimensionner
Faire pivoter
Image
Formes

Connexion aws

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

EC2

Création d’instance

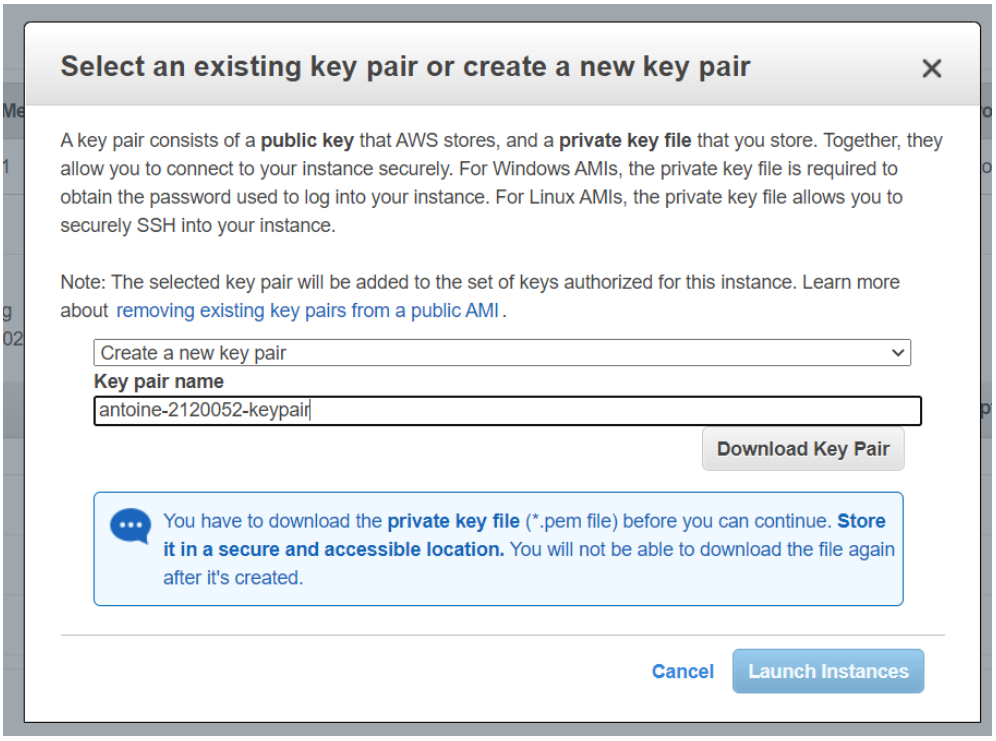
on a touché à rien jusqu’au tag on a gardé tout ce qui est gratuit

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

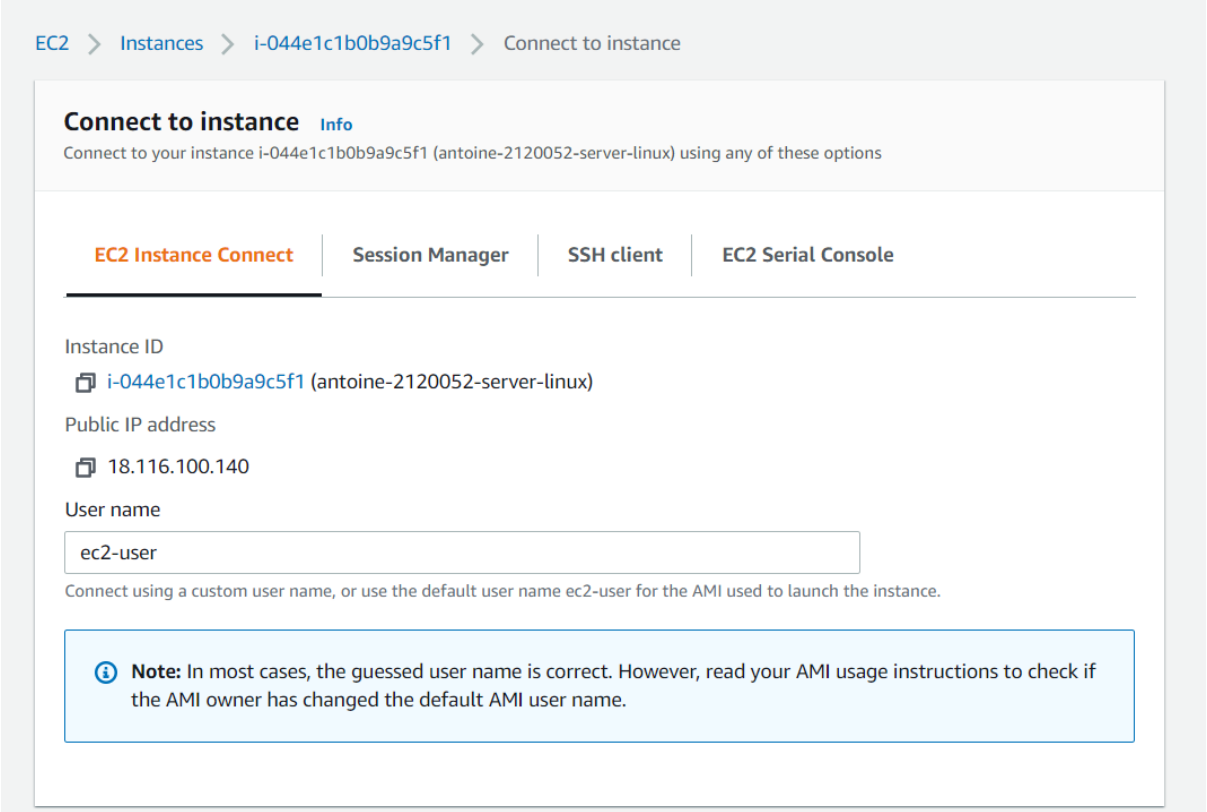
nom du security group : antoine-2120052-server-sg
source 0.0.0.0/0 → tout le monde a le droit de venir sur ma machine

antoine-2120052-keypair-ohio clef ohio

antoine-2120052-keypair la clef virginie

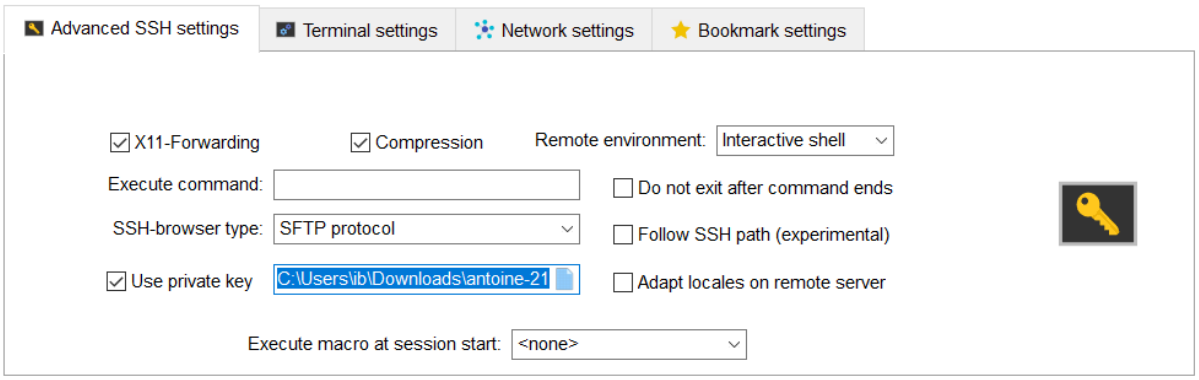


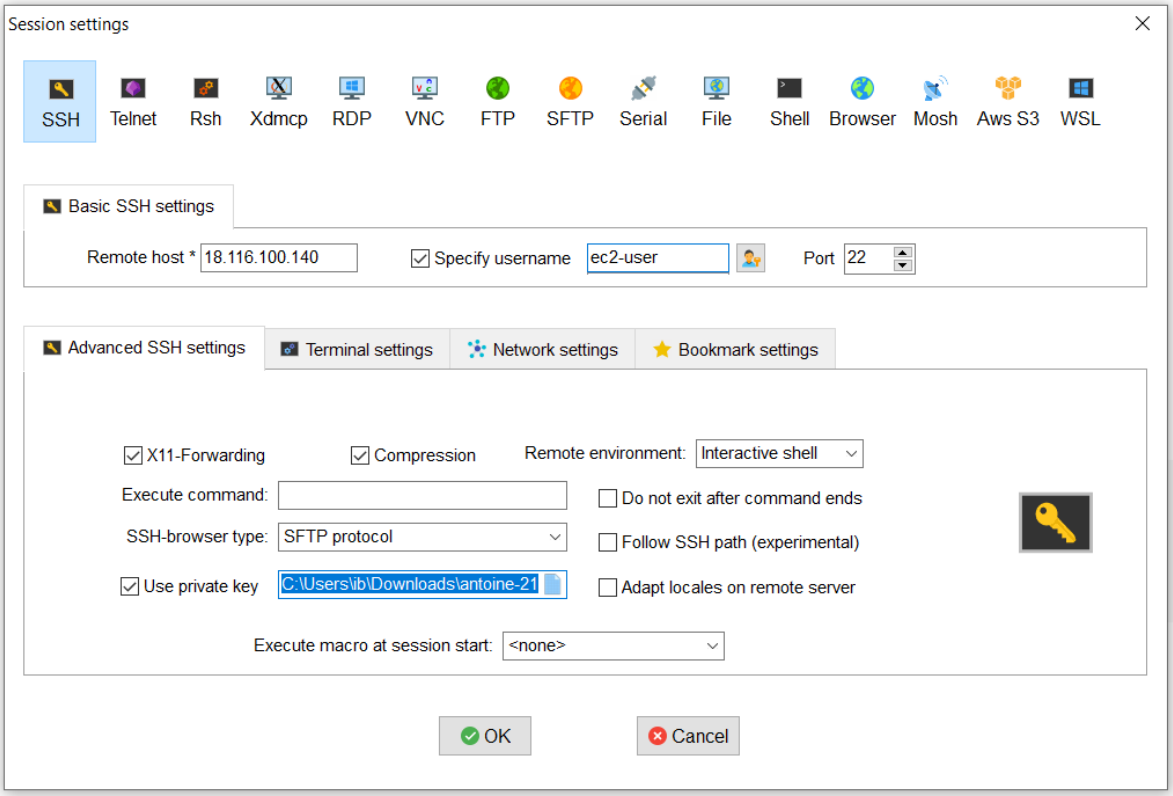
ma clef est telecharger dans les telechargement



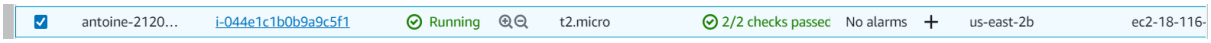
on va sur mobaxterm pour se connecter en ssh et ec2-user en username

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





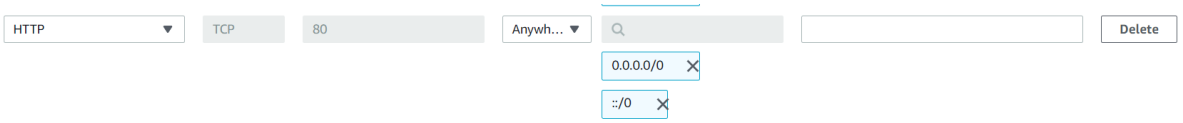
connexion via mobaxterm



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 un instance qui va respecter les normes mises en places et qui va utiliser la pair de clef creer vendredi idem pour 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 dis bonjour dirane site de Antoine

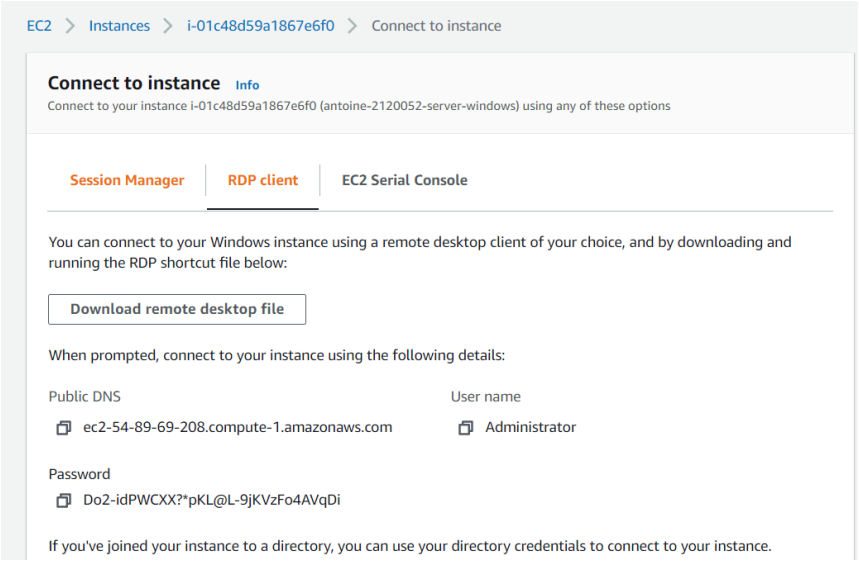
On refait comme au dessus
pour changer le contenu du site on va dans /var/www/html/index.html

Exercice Windows:

serveur w2k19 et plus amazon (Microsoft Windows Server 2019 Base)
mettre en place une nouvelle sg faites pour le windows
je dois depuis la maison m connecter a votre instance
protocole pour se connecter sur un windows a distance

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

rdp : remote desktop protocol
on clique sur notre instance puis sur connect et on telecharge ce qu'il nous demande après 4 min on peut faire get password et decrypter le password avec la clef privé
password pour cet exo : Do2-idPWCXX?*pKL@L-9jKVzFo4AVqDi



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 :

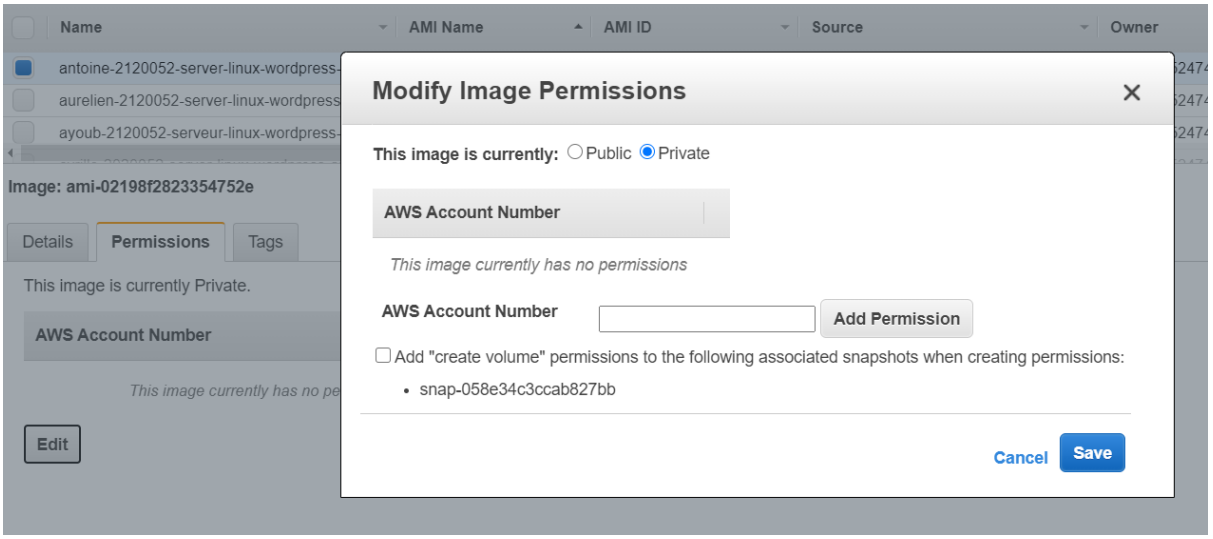
déployer autant d'instance que je veux avec wordpress à l'intérieur
la pour le moment on a le paramétrage initial de la machine il faudrait l'enlever
il faut permettre à l'user de fournir lui même ces infos
on commence par `docker-compose down -v`
on sélectionne notre instance
action > image and template > 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

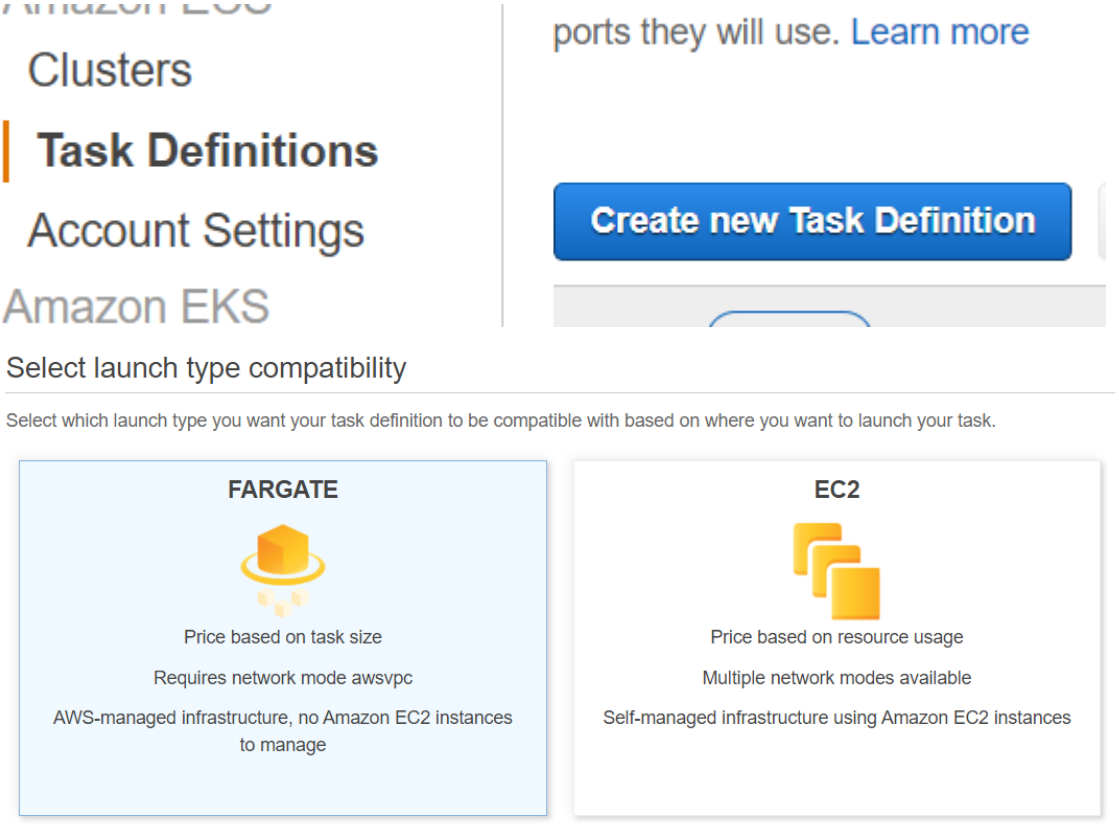
puis on delete le snapshot

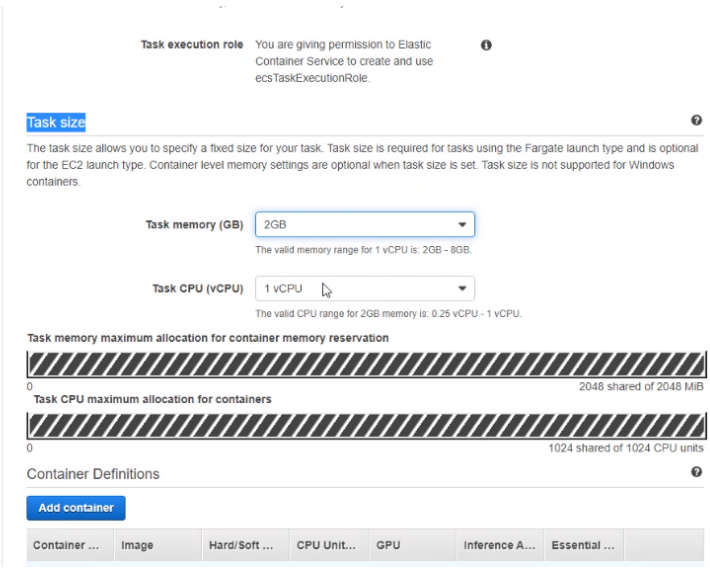
Exercice création task avec EBS :

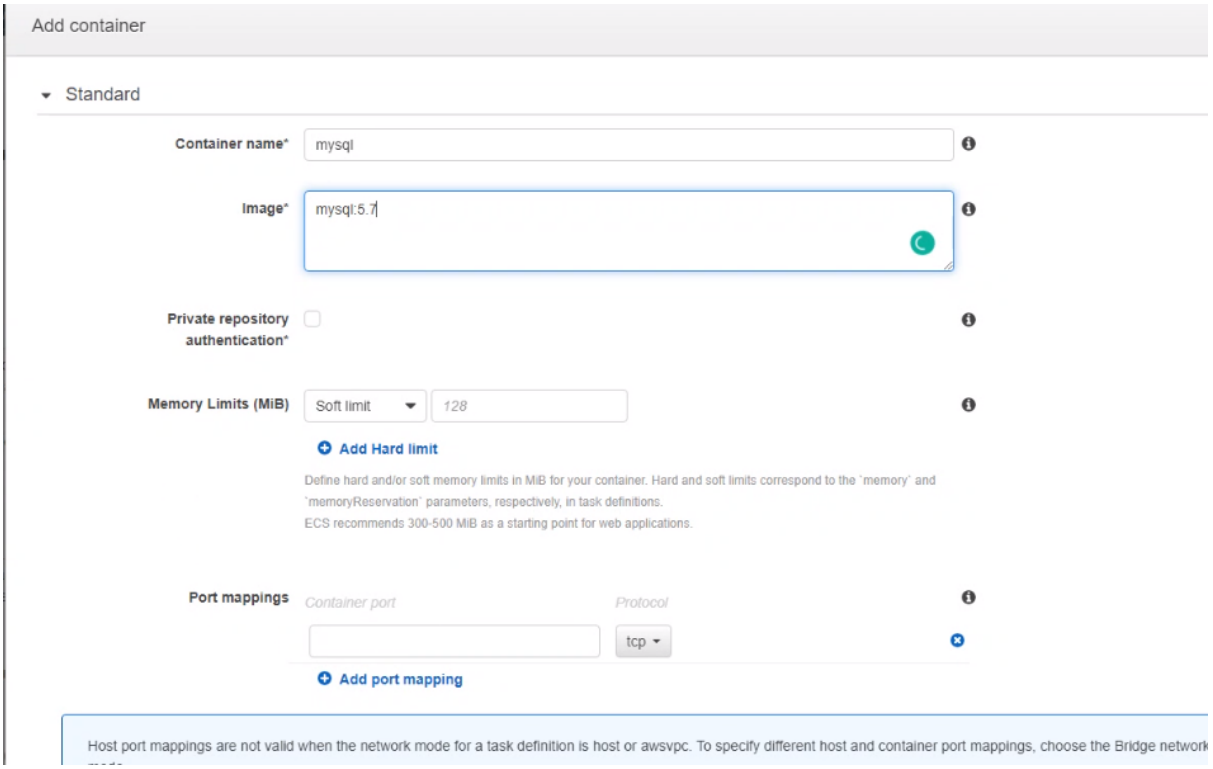
Elastic Container Service :
https://console.aws.amazon.com/ecs/home?region=us-east-1#/getStarted

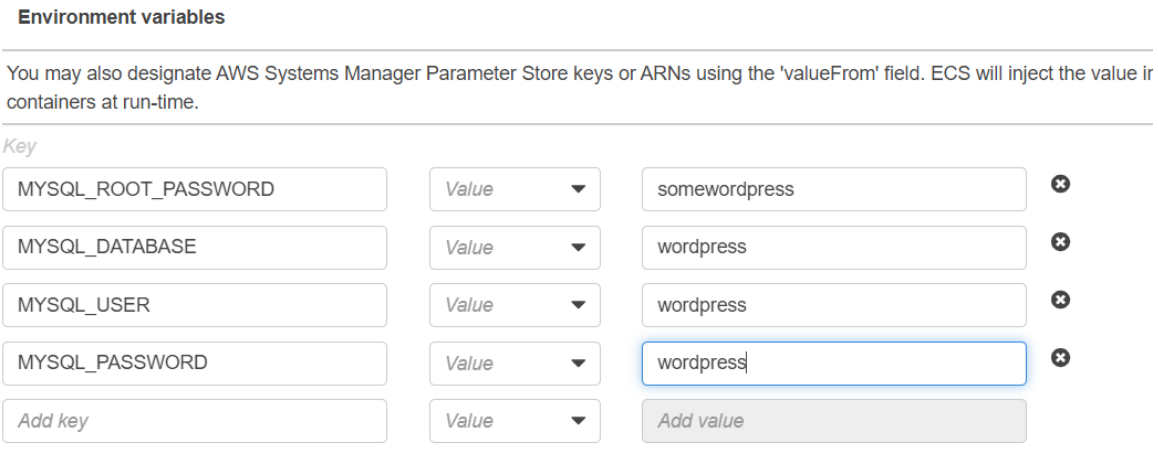
dirane fait un cluster et nous on va faire des task dessus

on clique sur **task definition** a gauche
puis sur **Create new Task Definition** et on choisit **fargate**

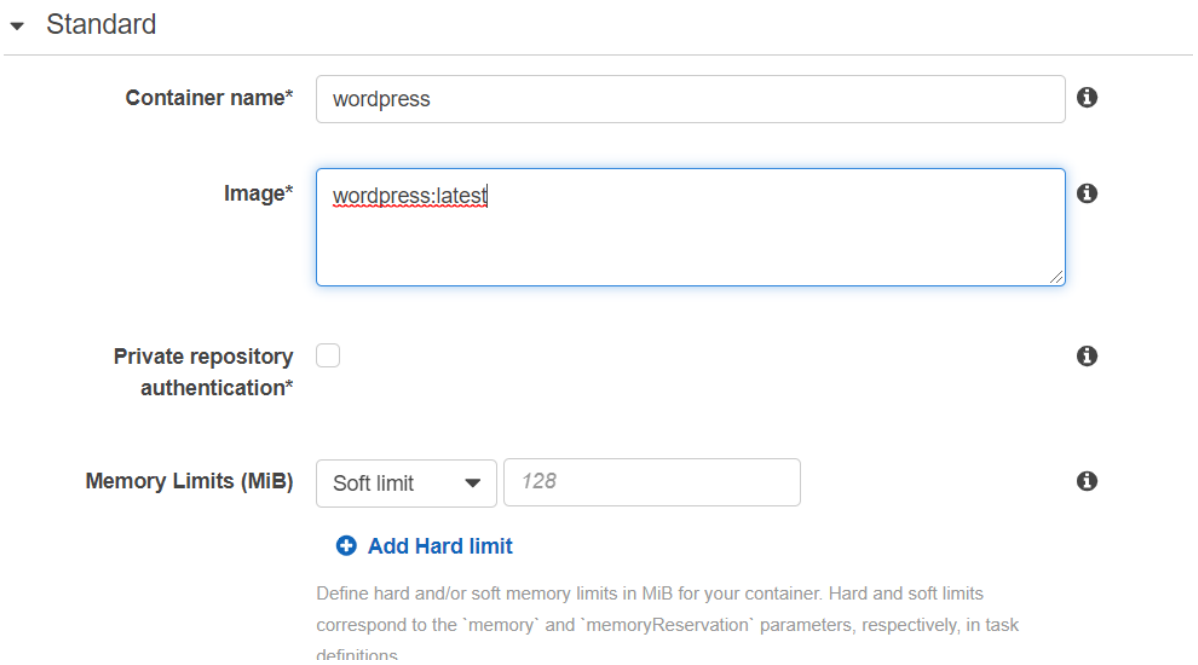



Task execution role
You are giving permission to Elastic Container Service to create and use ecsTaskExecutionRole.
Task size
The task size allows you to specify a fixed size for your task. Task size is required for tasks using the Fargate launch type and is optional for the EC2 launch type. Container 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 ...

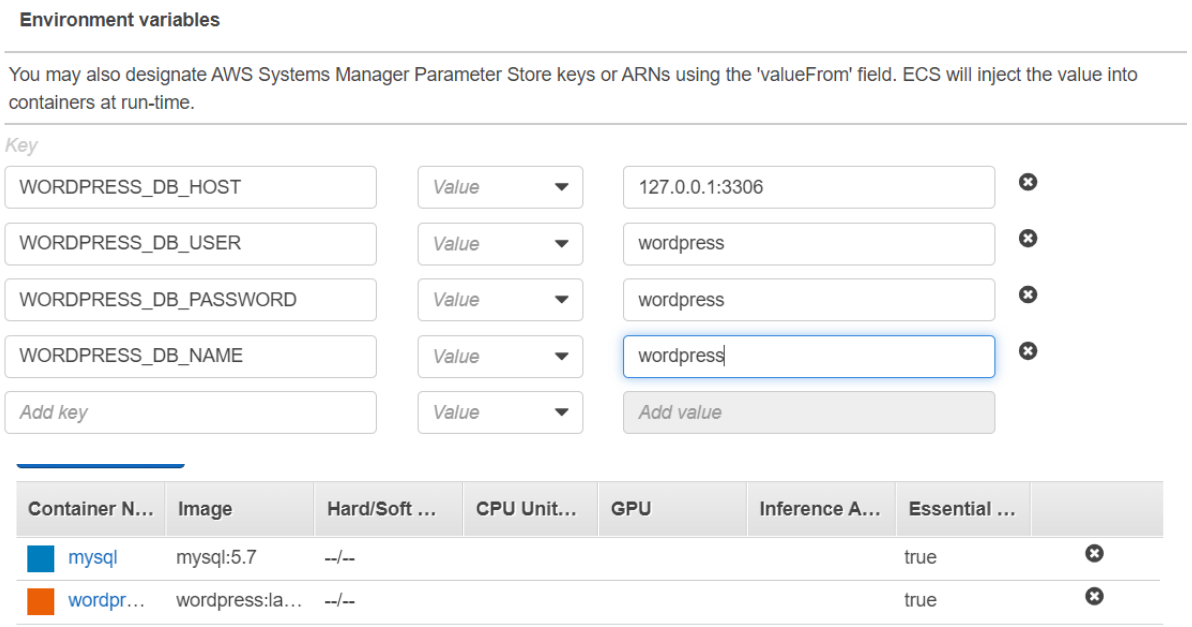

Add container
Standard
Container name*
mysql
Image*
mysql:5.7
Private repository authentication*
Memory Limits (MiB)
Soft limit
128
Add Hard limit
Define hard and/or soft memory limits in MiB for your container. Hard and soft limits correspond to the `memory` and `memoryReservation` parameters, respectively, 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 ir 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

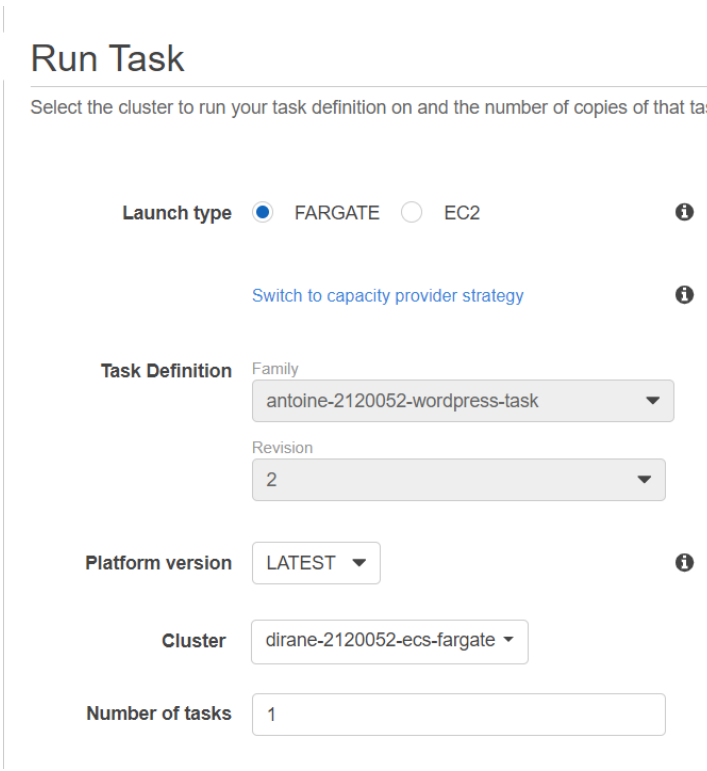


on met le port 80 pour wordpress

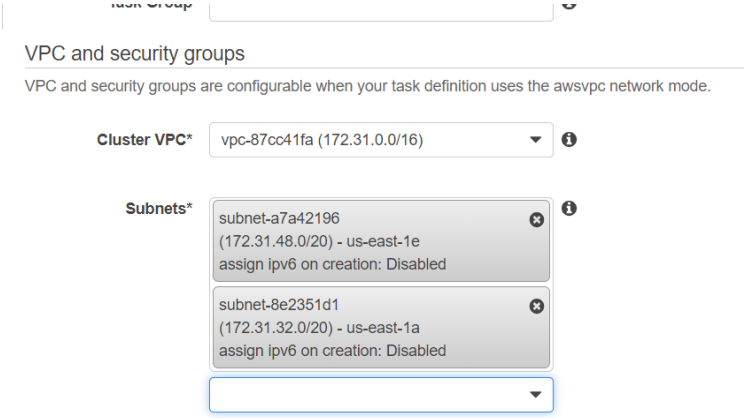


on peut créer une nouvelle révision (les task sont versionnées)

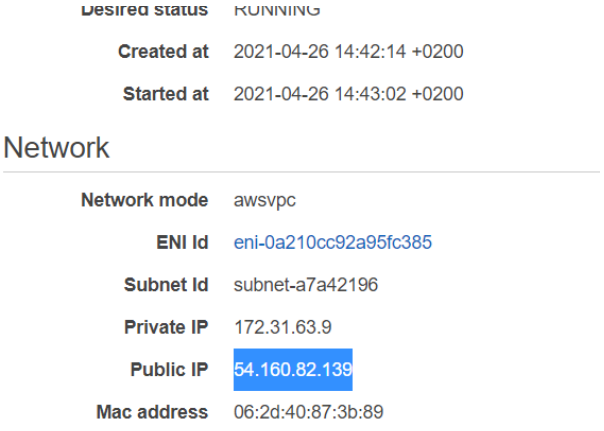
aller sur notre task cliquez sur action>run task
on choisit fargate et on s’assure que le cluster est le bon



on ajoute un vpc et deux sous réseaux :



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



Storage

Exercice persistance des données :

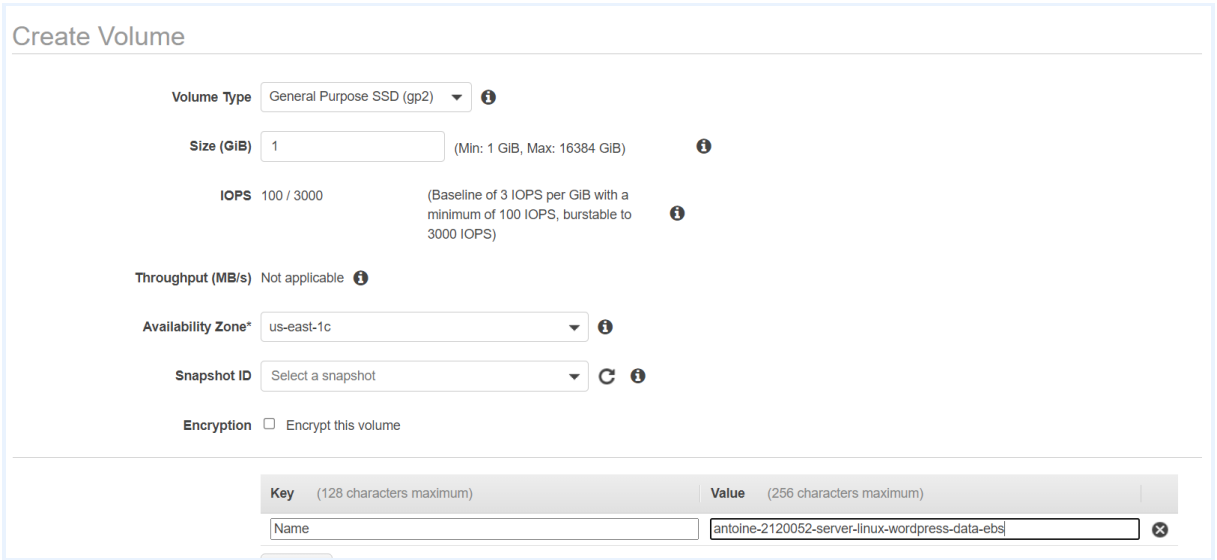
le client ne veut pas que la bdd soit stocké dans un volume de type volumes mais dans un volumes de type bind mount qui se trouverait dans /data/
la ligne `db_data:/var/lib/mysql`
par `/data:/var/lib/mysql` et supprimer le volume en bas

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

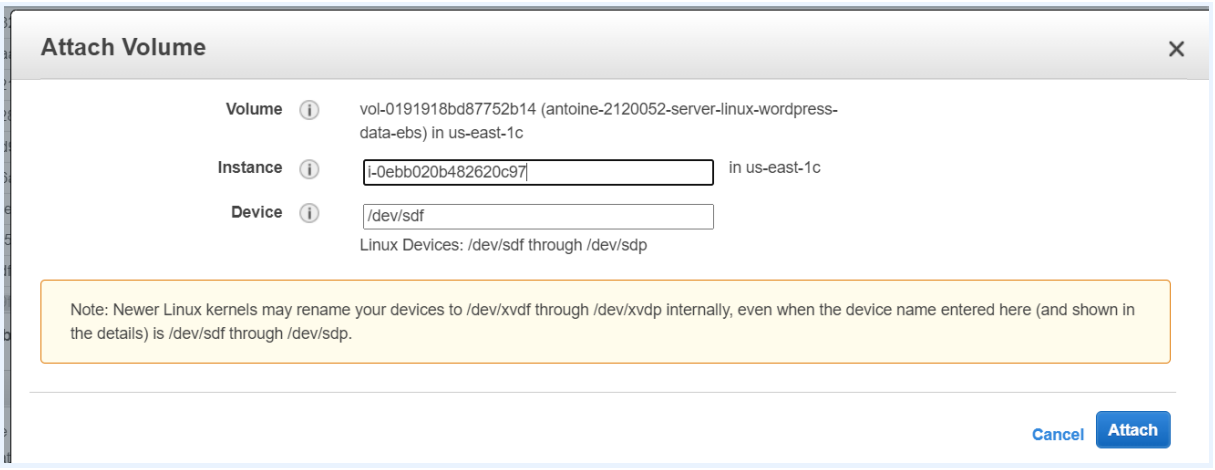
on vire tout ce qu'il y a dans /data

```
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



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



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

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 recreer 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 rerempli 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

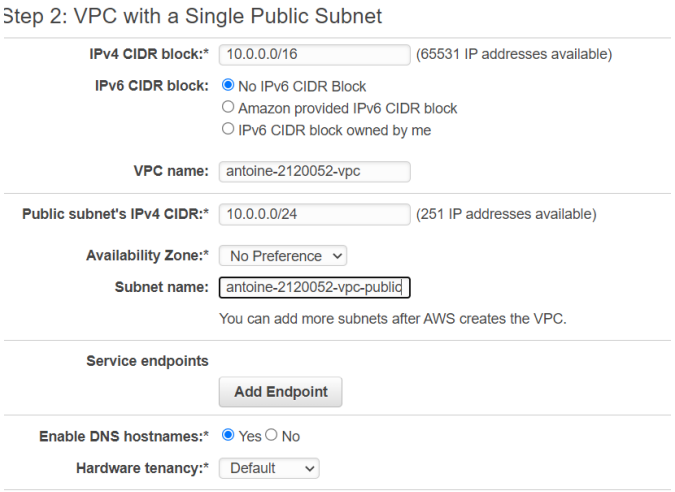
il faut que ca garde l'arborescence
dl le code du client
https://github.com/diranetafen/static-website-example/archive/refs/heads/master.zip

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

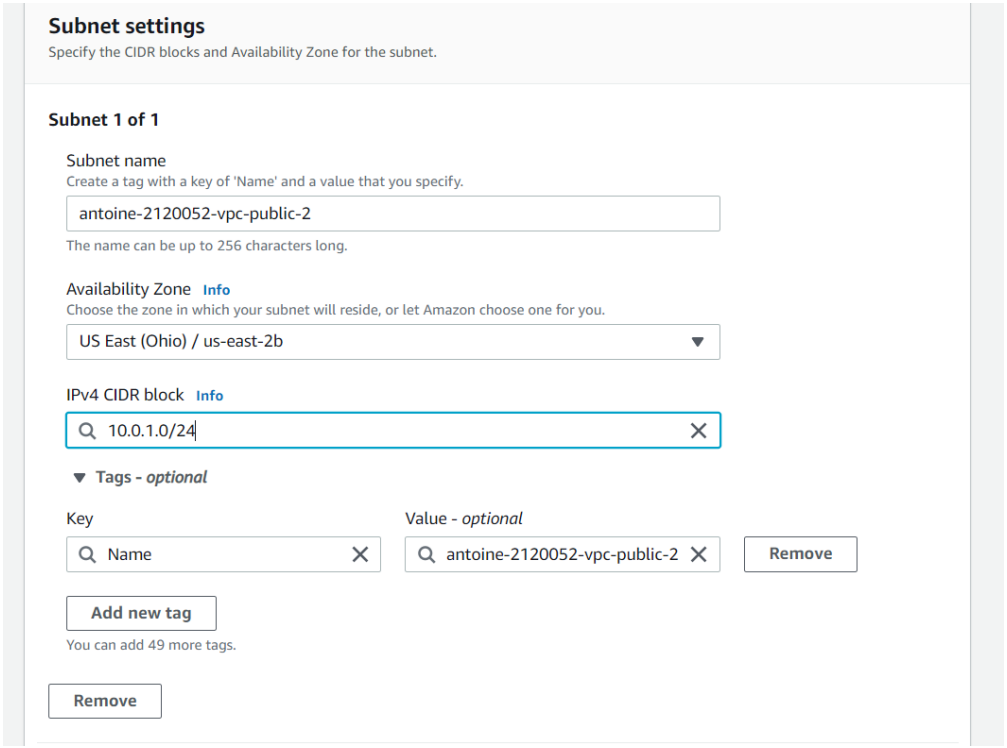
on fait la création assistée
on choisit la config qui se rapproche le plus de ce qu'on veut (ici la premiere)



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

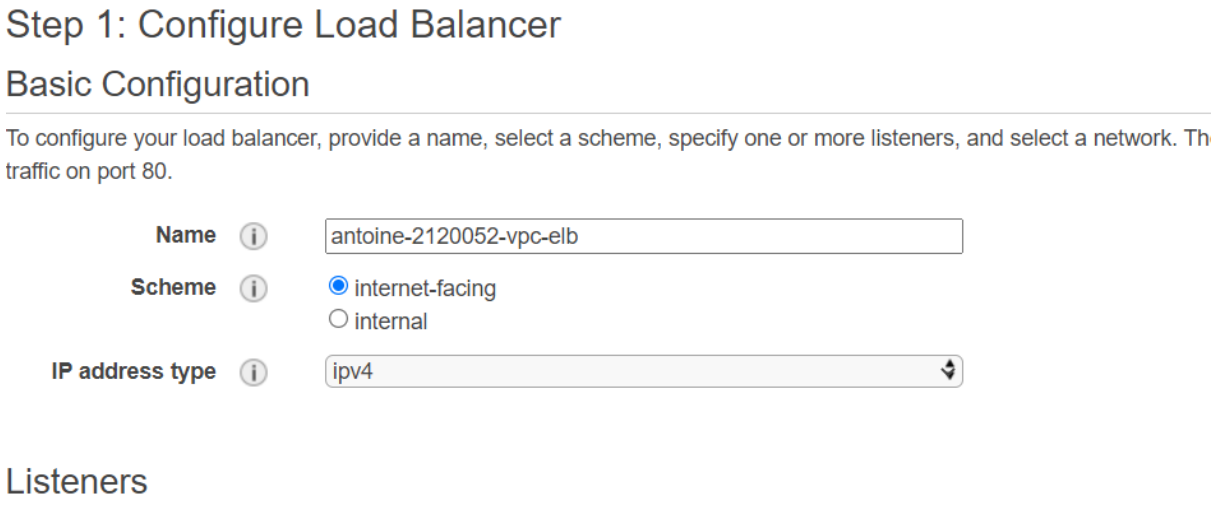


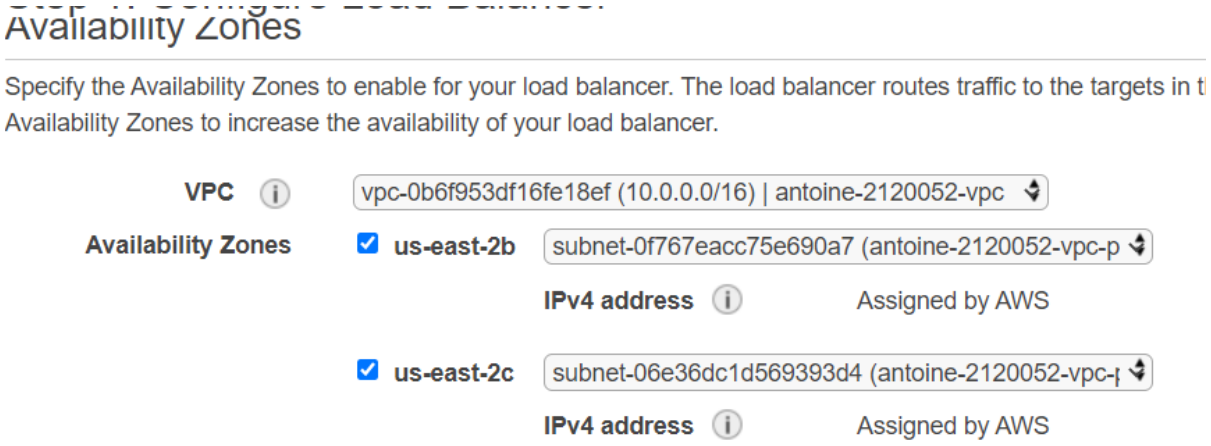
on clique ensuite sur notre sous réseau et on clique sur actions > Modify auto-assign IP settings et on clique sur enable (on le fait pour nos deux sous réseaux)

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

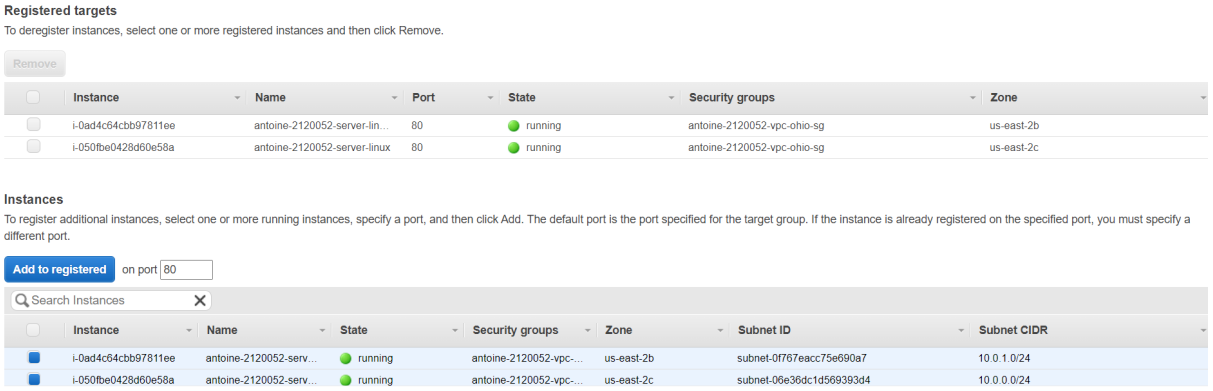
on prend l'id de la routing table de notre premier reseaux et on l'utilise pour modifier la table de routage
sinon on peut cliquer sur routes a gauche choisir l'id de la table de routage a odifier et ajouter la nouvelle route

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



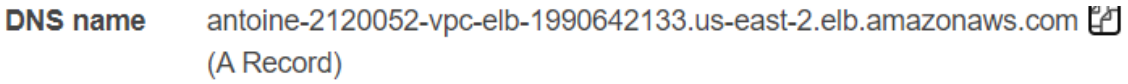


antoine-2120052-vpc-tg
on cree notre target group (la ou le lb va dispatcher les gens)



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



Devops:

CodeCommit

sur aws ecs amazon linux
git cloner le code du client
dans un dossier webapp
dockerfile qui va permetre de buld l'app du client

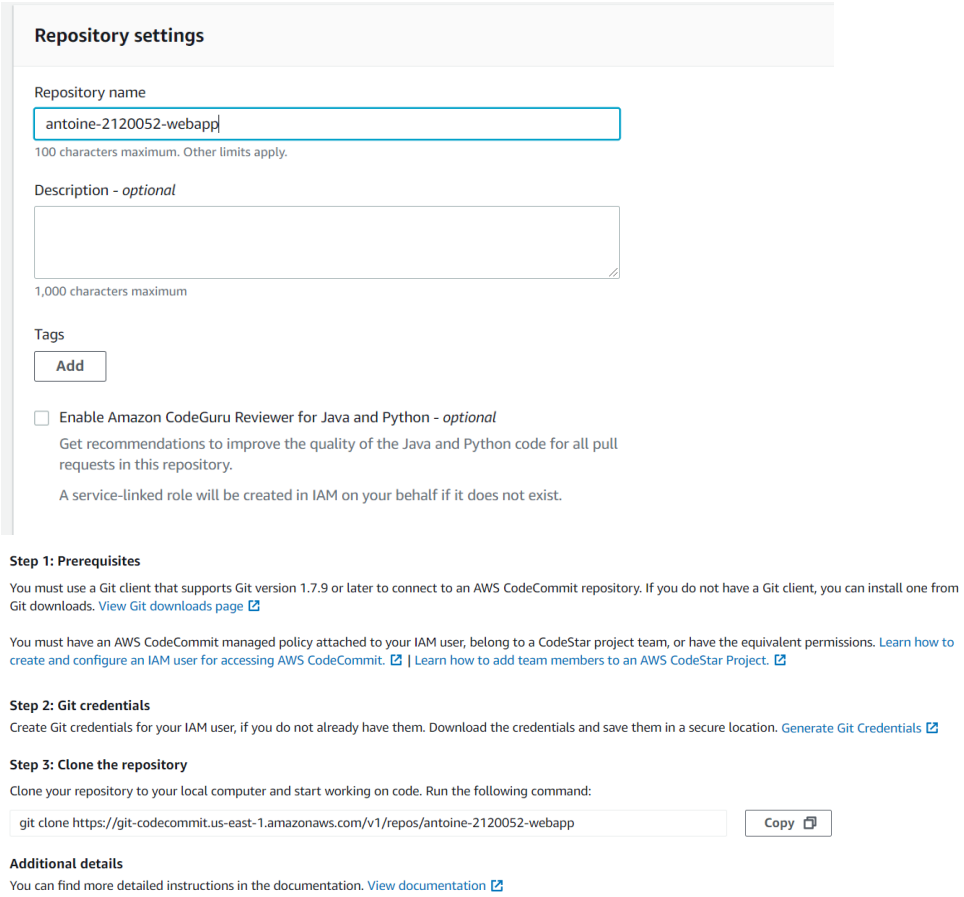
et delpoyer l’app pour verifier que ca fonctionne
webapp:latest

une fois déployer 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



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
```

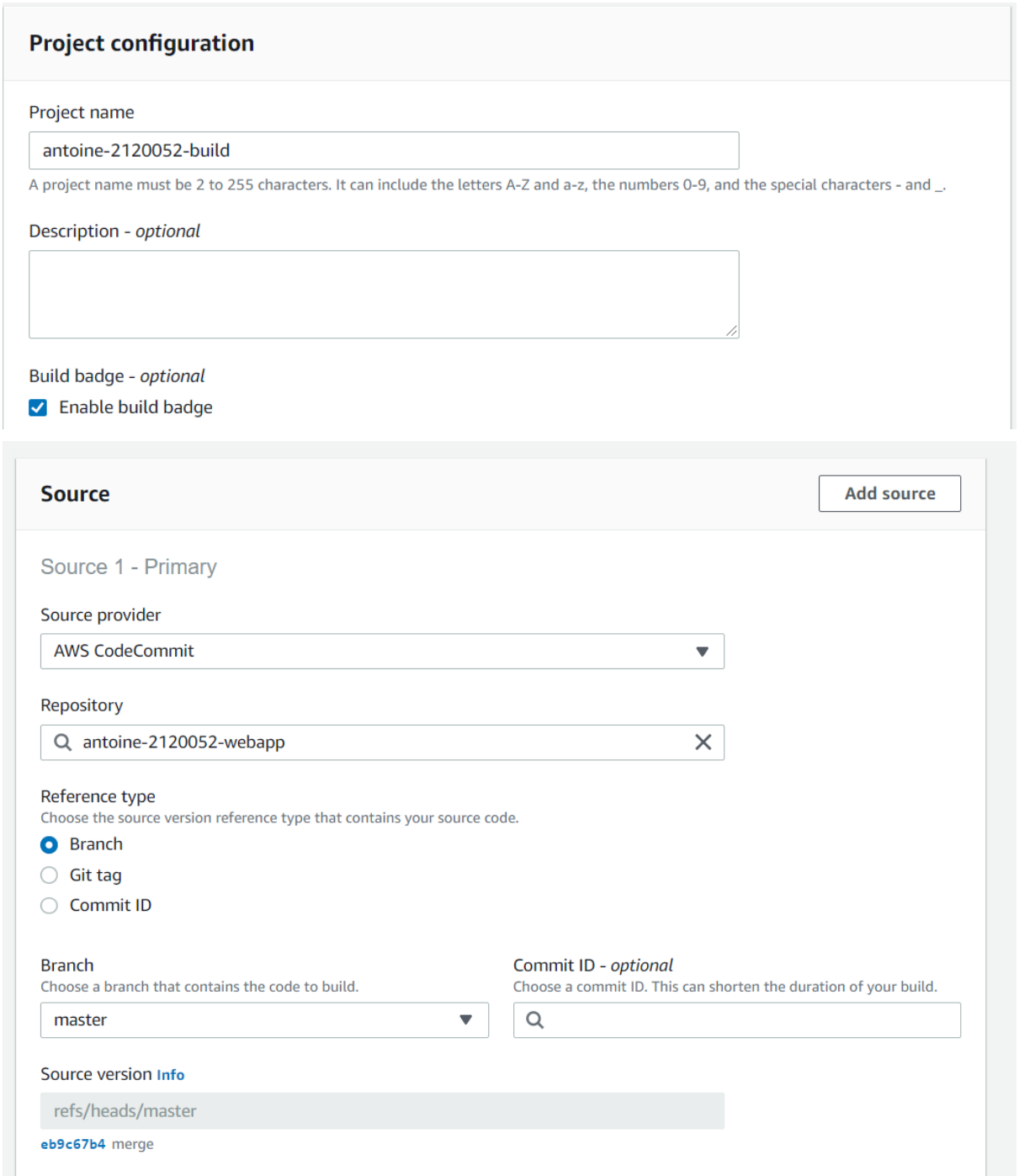
pour externaliser les variables
commande de base pour login build push

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

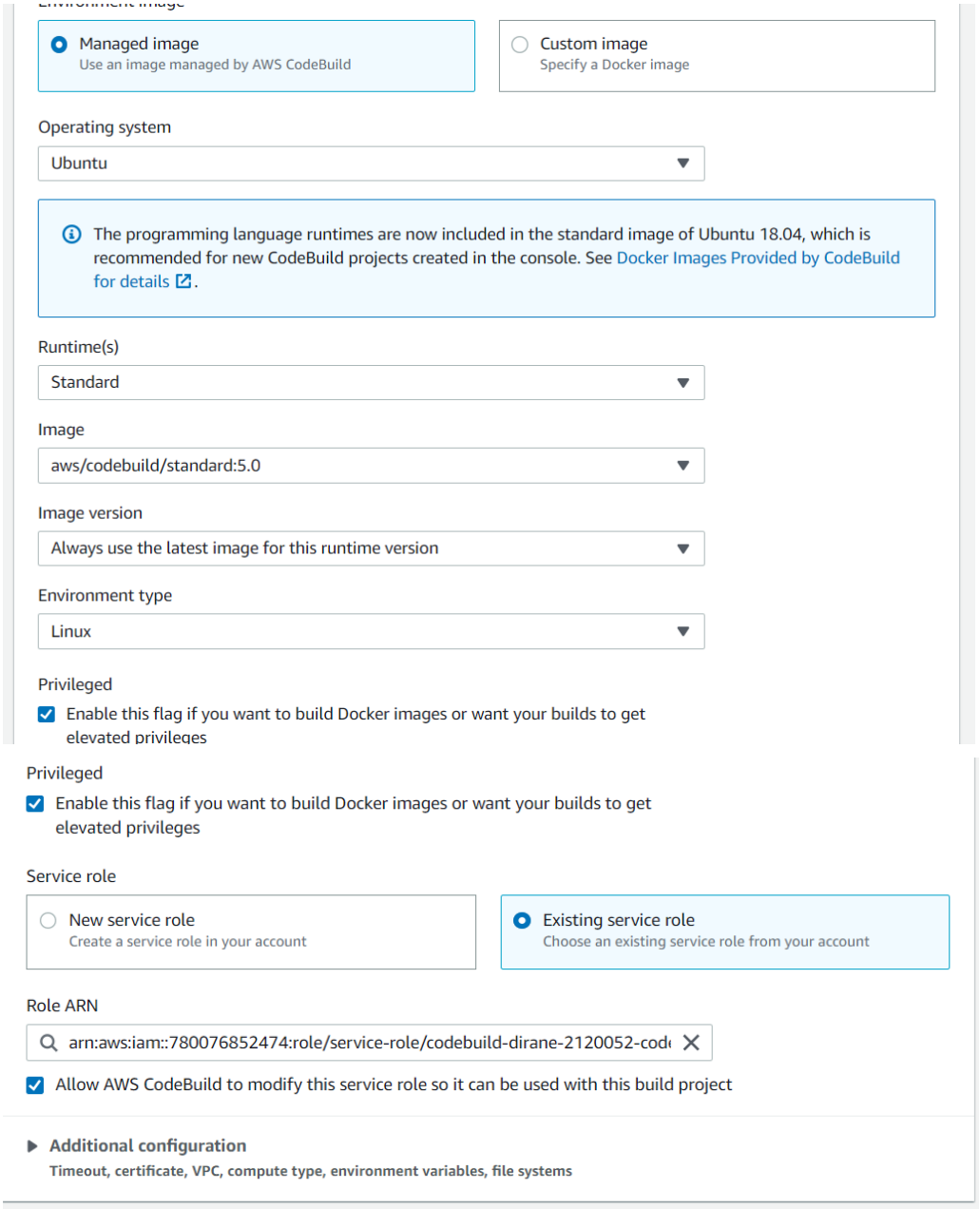
```
version: 0.2

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



Environnement



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

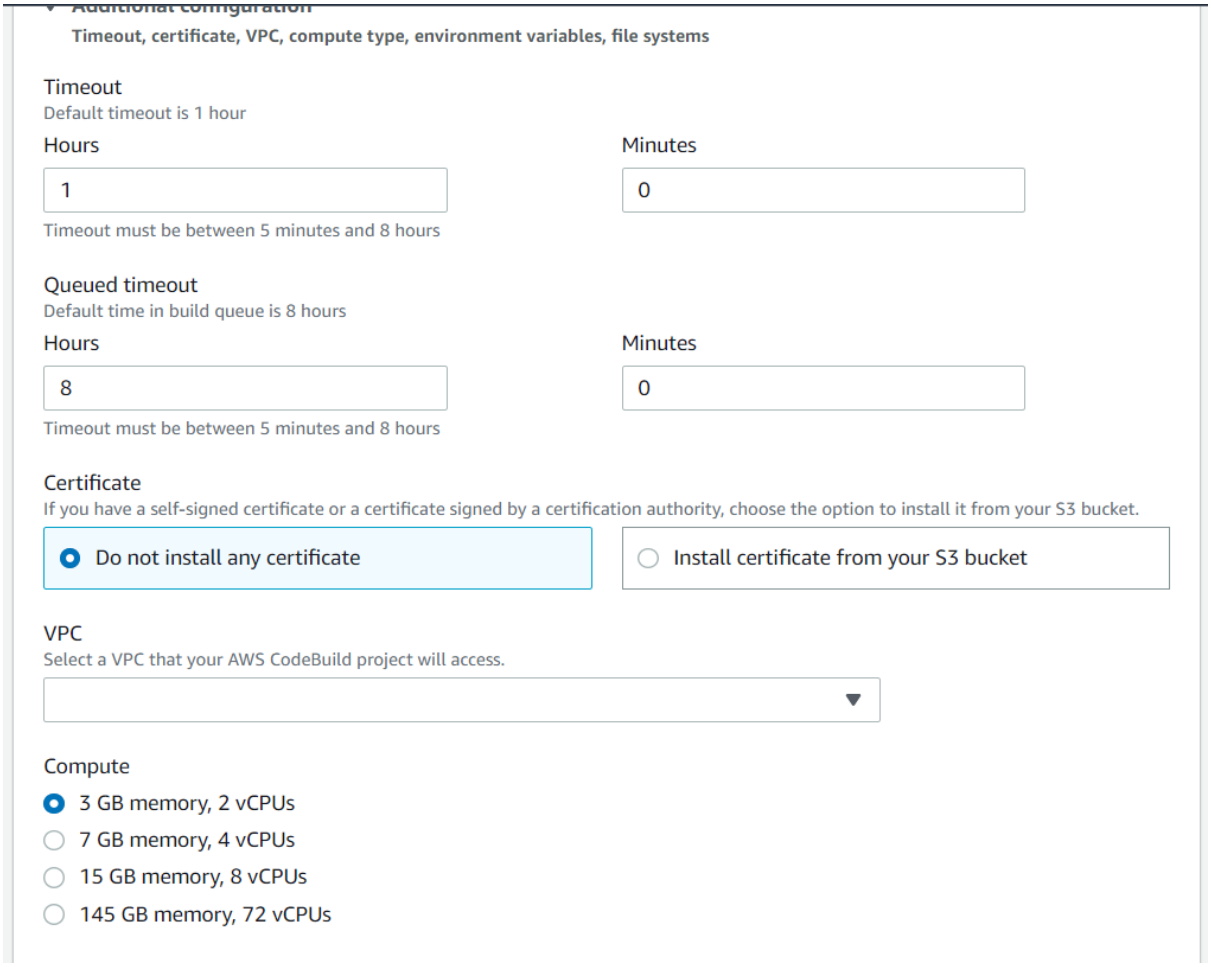
Existing service role
Choose an existing service role from your account

Role ARN

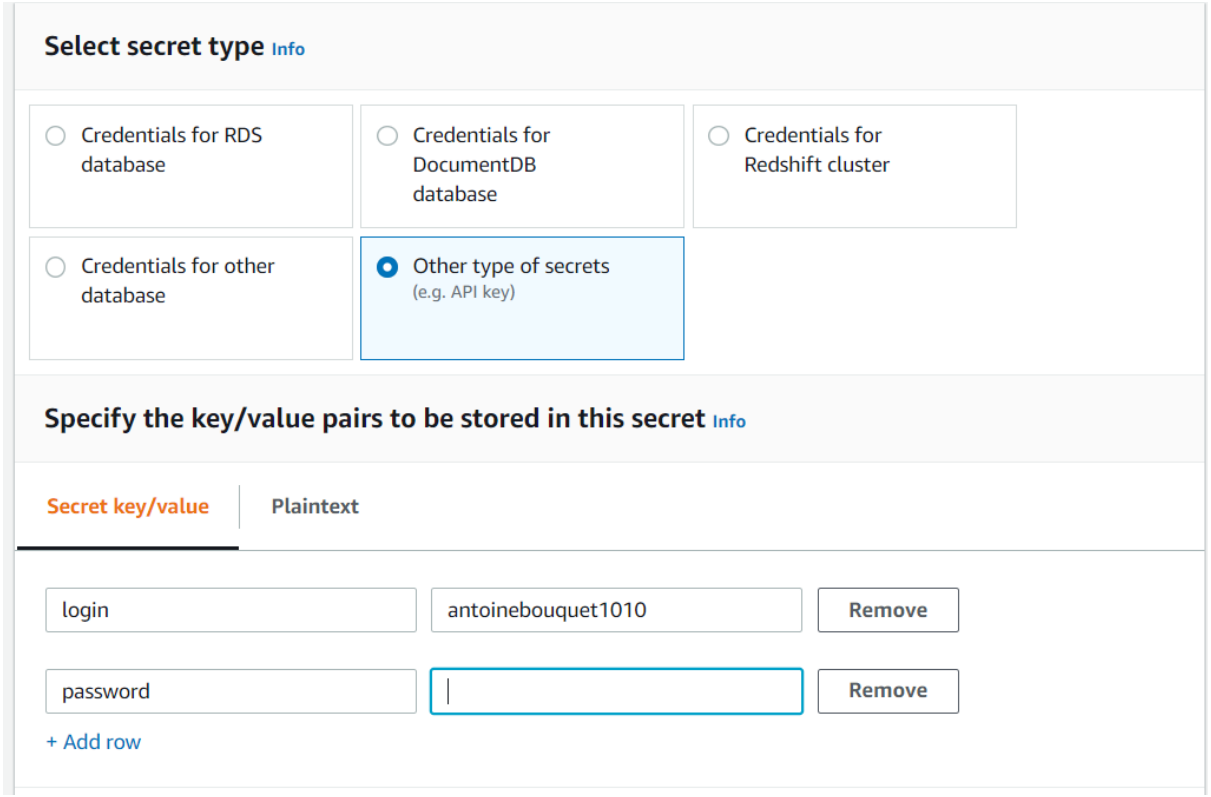
arn:aws:iam::780076852474:role/service-role/codebuild-dirane-2120052-cod

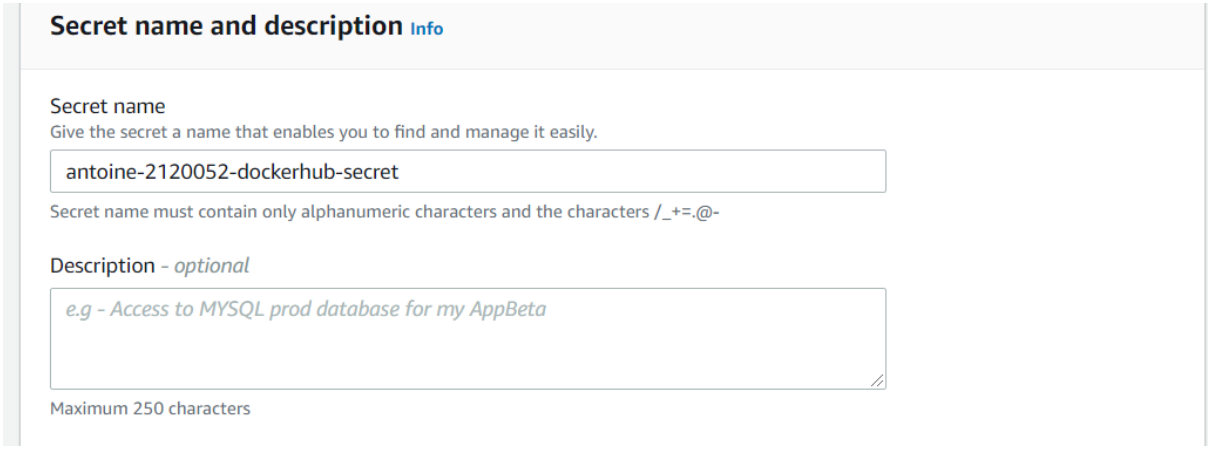
Allow AWS CodeBuild to modify this service role so it can be used with this build project

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

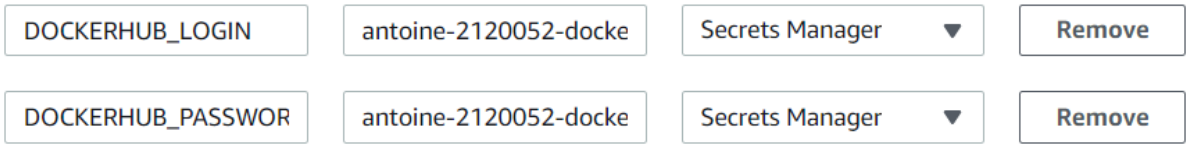


on va dans secret manager pour stocker nos login et mdp :

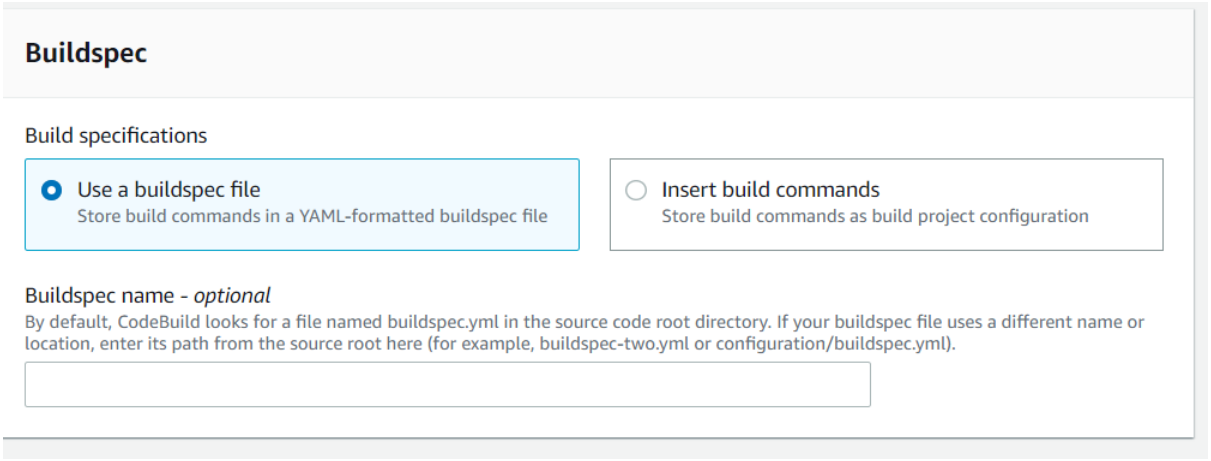




puis suivant>suivant>store



ajout de nos variable secret
nomdusecret:variable
antoine-2120052-dockerhub-secret:login
antoine-2120052-dockerhub-secret:password



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 crée notre task maintenant il faut la lancer (appuyer sur start buiild)
il a permit 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

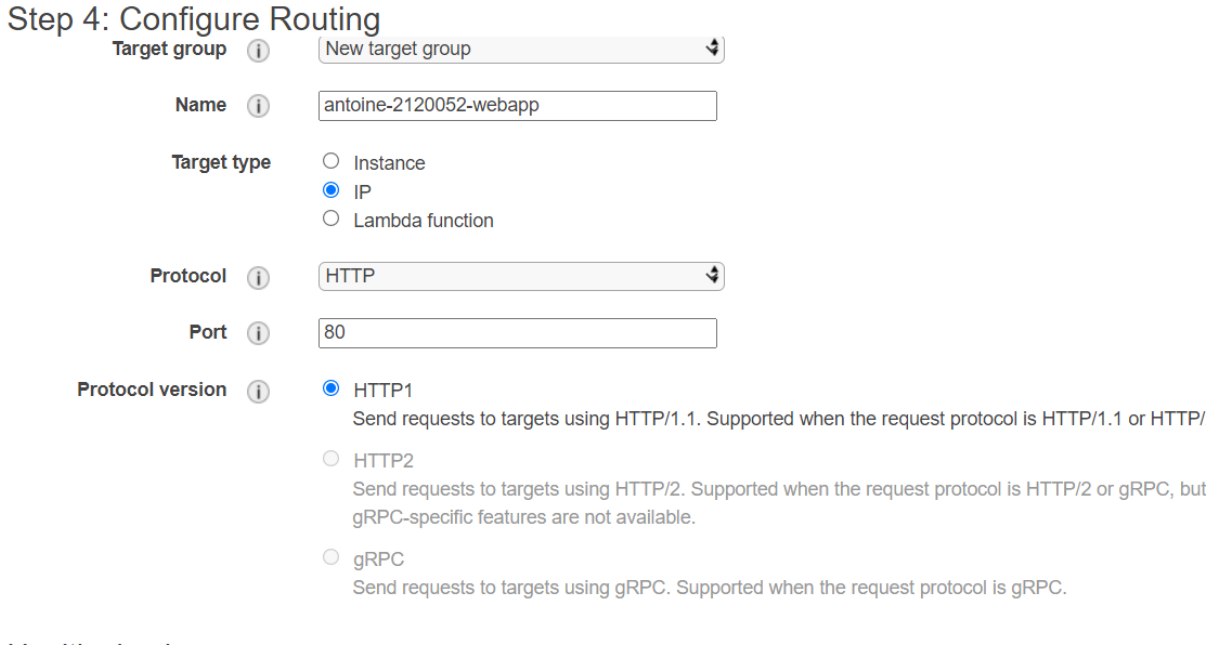
ensuite on va déployer son app avec ecs pour déployer son image
nom webapp
image antoinebouquet1010/webapp:latest

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

on va transformer notre task en sevice et on va creer un elb

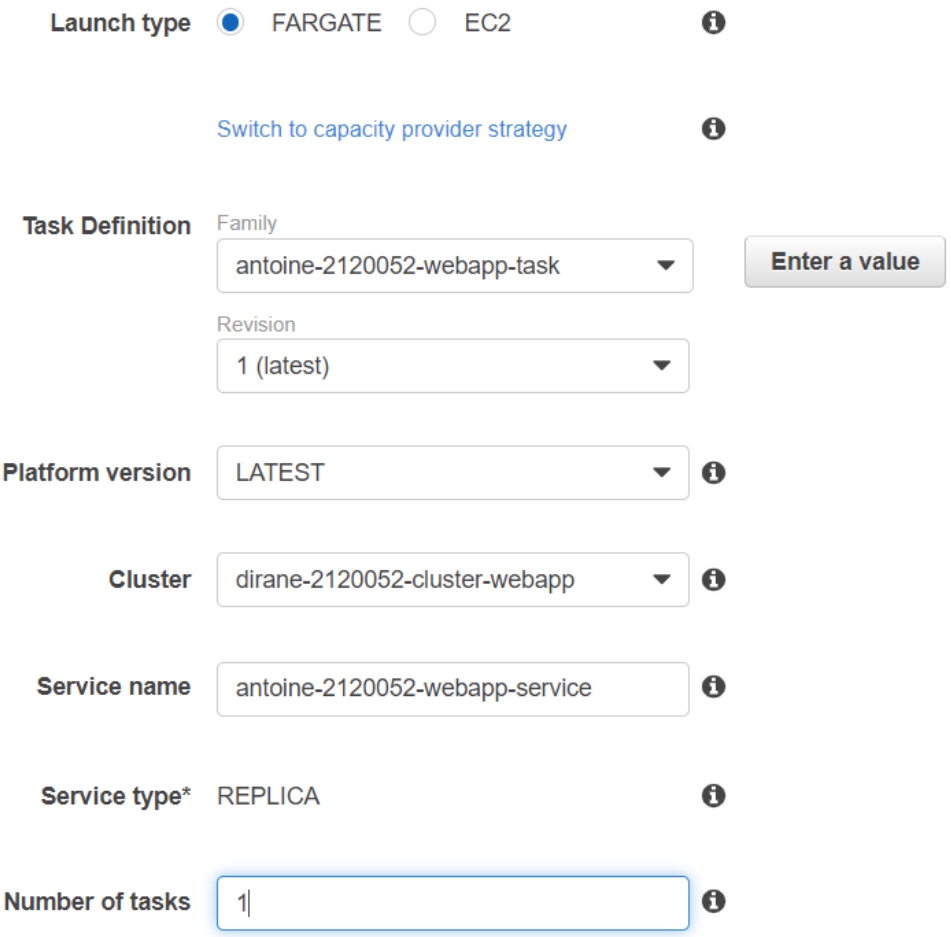
d’abord l’elb

on prend notre sg et on decoche celui cochez par defaut

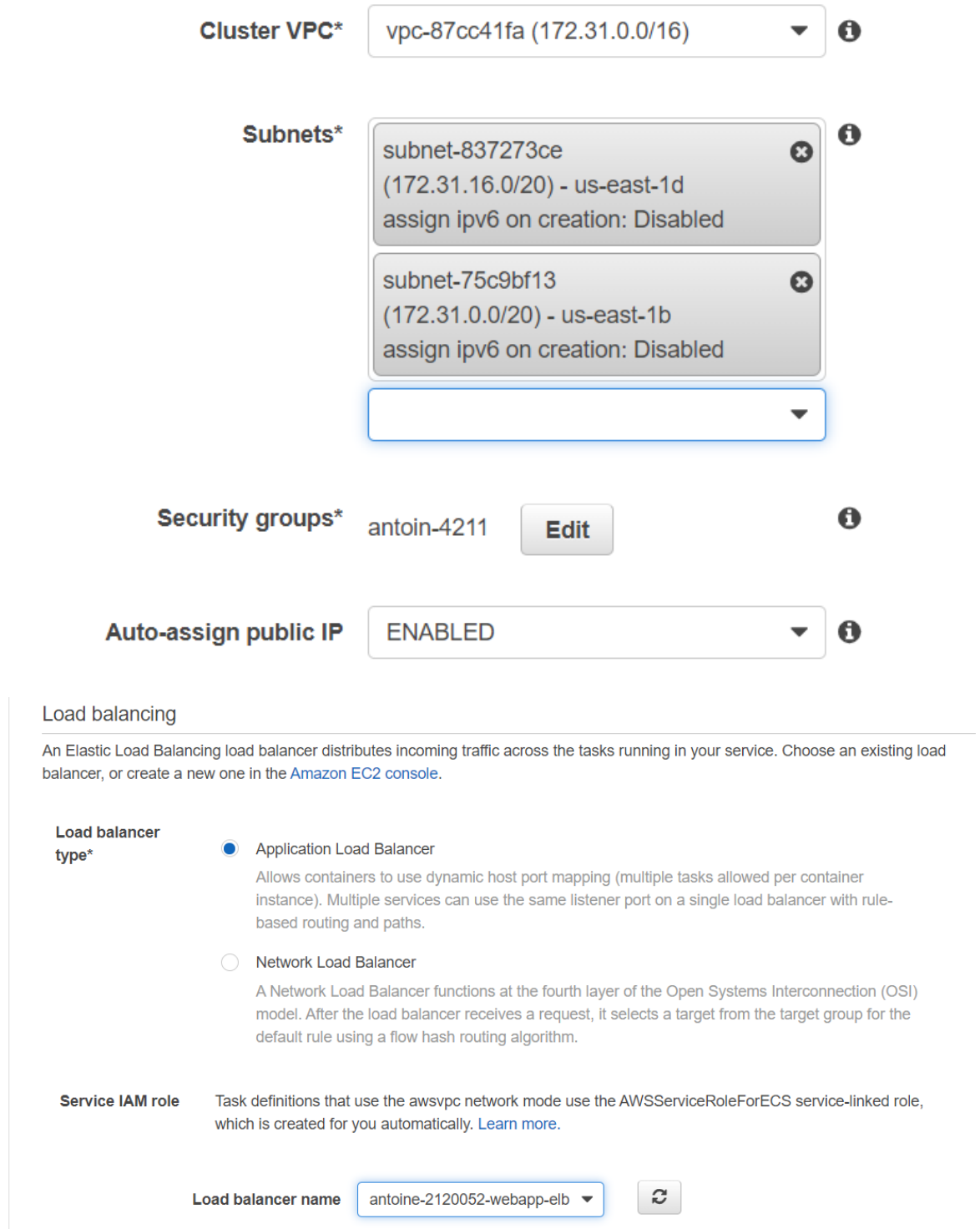


cette fois on a des target de types ip
et apres on ne register pas de cible

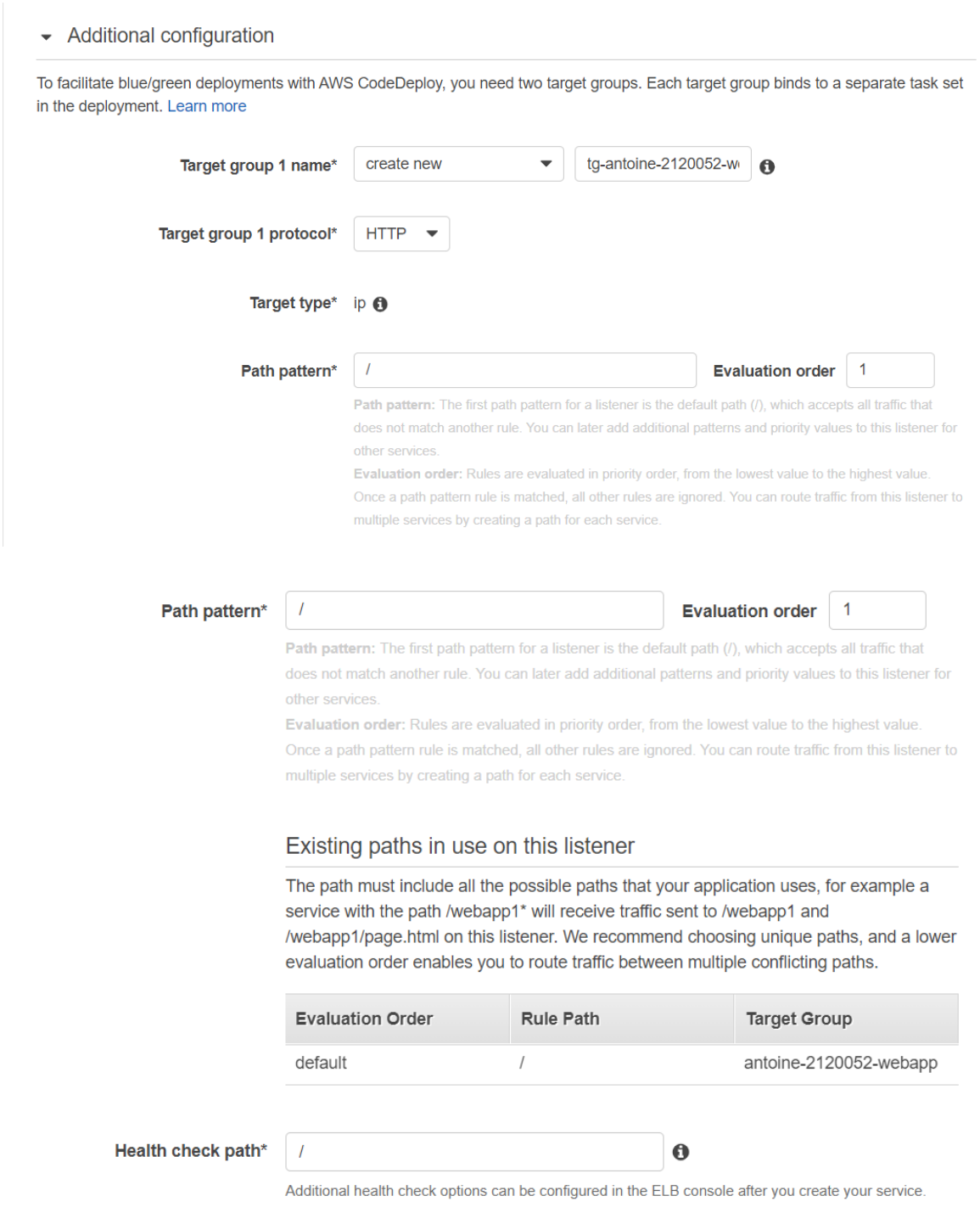
maintenant sur ecs on va créer un service qui va créer les tasks



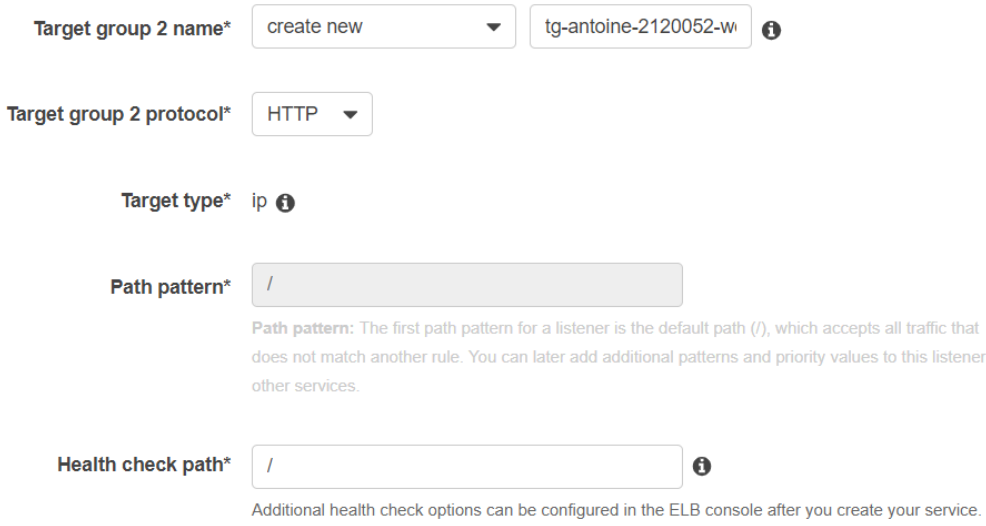
deploiement sur blue green



la on a cliqué sur le bouton add to load balancer



c’est pas / mais /* qu’il faut mettre

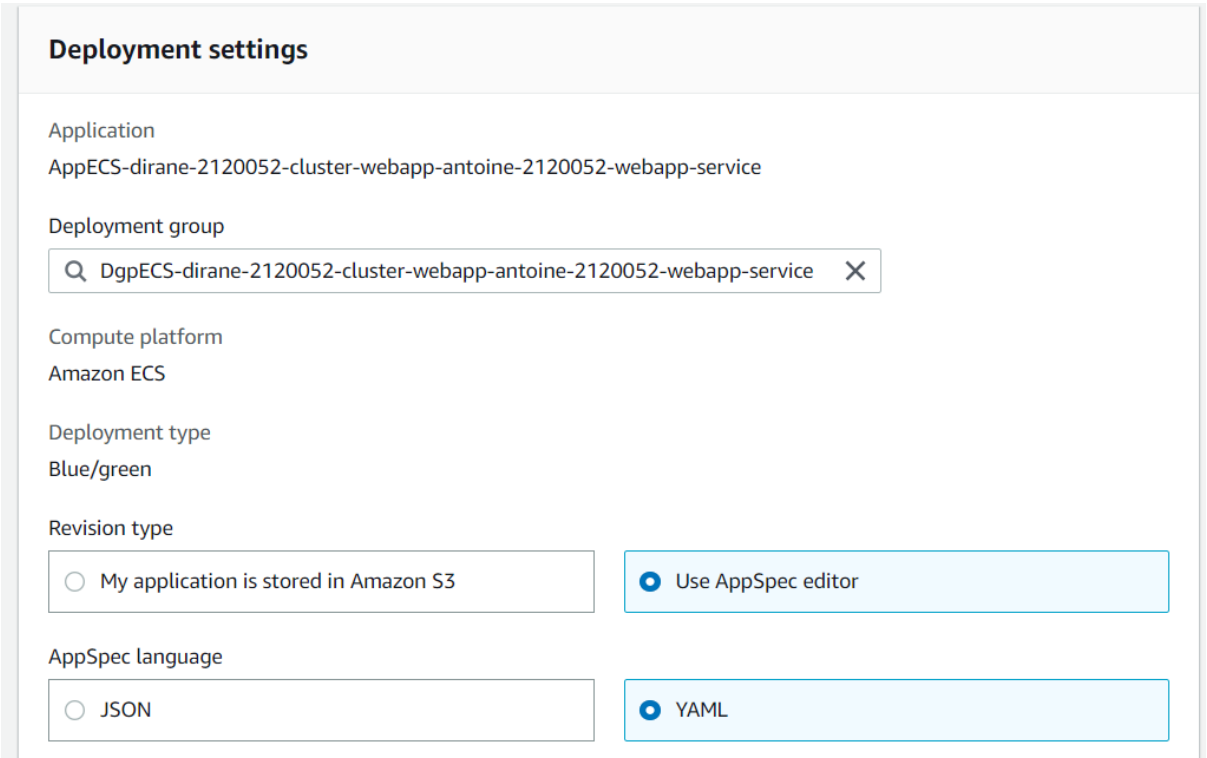


le site es pas beau mais il faudrait faire des paramétrages de loadbalancer

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 deploiement



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-weba
pp-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 mecanique 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.

antoine-2120052-pipeline

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

arn:aws:iam::780076852474:role/service-role/AWSCodePipelineServiceRole-e

▸ Advanced settings

Source provider

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

AWS CodeCommit

Repository name

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

antoine-2120052-webapp

Branch name

Choose a branch of the repository

master

Change detection options

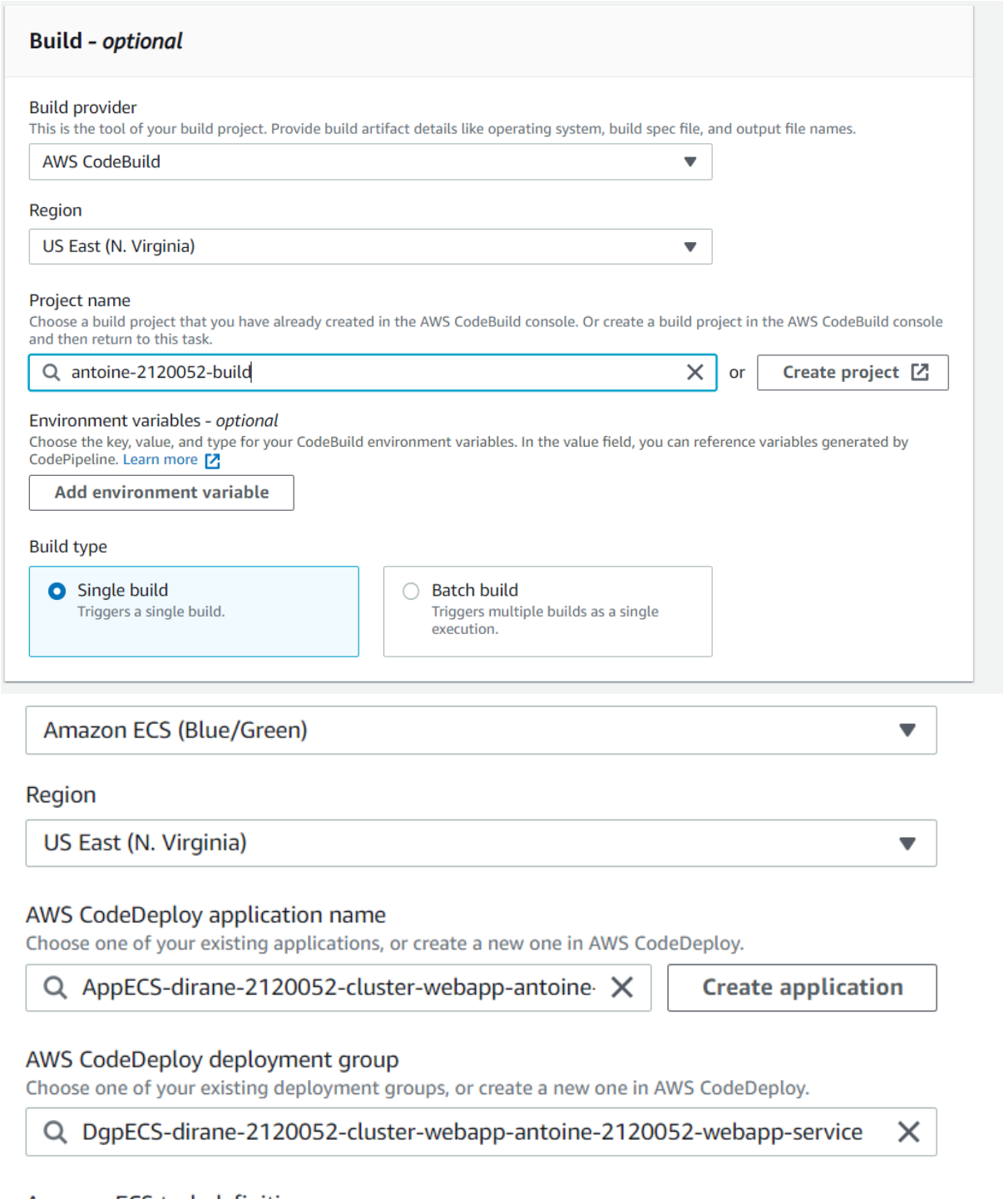
Choose a detection mode to automatically start your pipeline when a change occurs in the source code.

- (•) 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.



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.

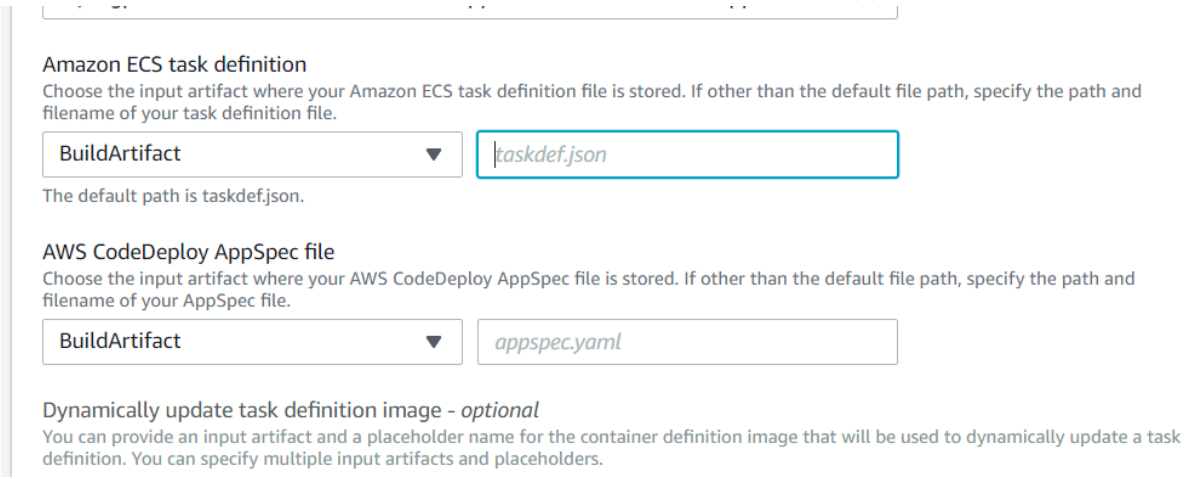
AppECS-dirane-2120052-cluster-webapp-antoine·

Create application

AWS CodeDeploy deployment group

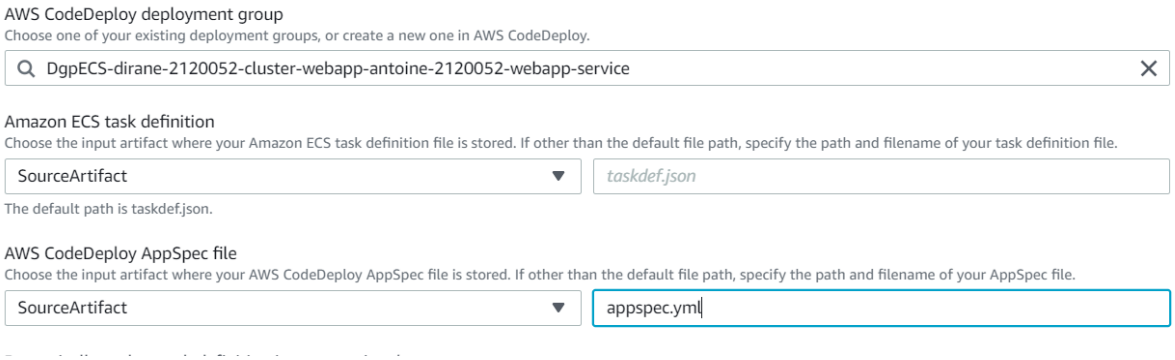
Choose one of your existing deployment groups, or create a new one in AWS CodeDeploy.

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



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

et ca :



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