

Documentation de l'infrastructure cloud AWS – Projet 10

Procédure rédigée par : Vassenet-Guihot Romain

OPENCLASSROOMS

Table des matières

I -Introduction :	2
II -Construire le Réseau AWS :.....	3
III -Systèmes EC2, RDS, S3	5
IV -Load Balancer & Mise à l'échelle	12
V -VPN site à site & Intranet isolé :	16
VI -Script Cloud Formation :	20

I - Introduction :

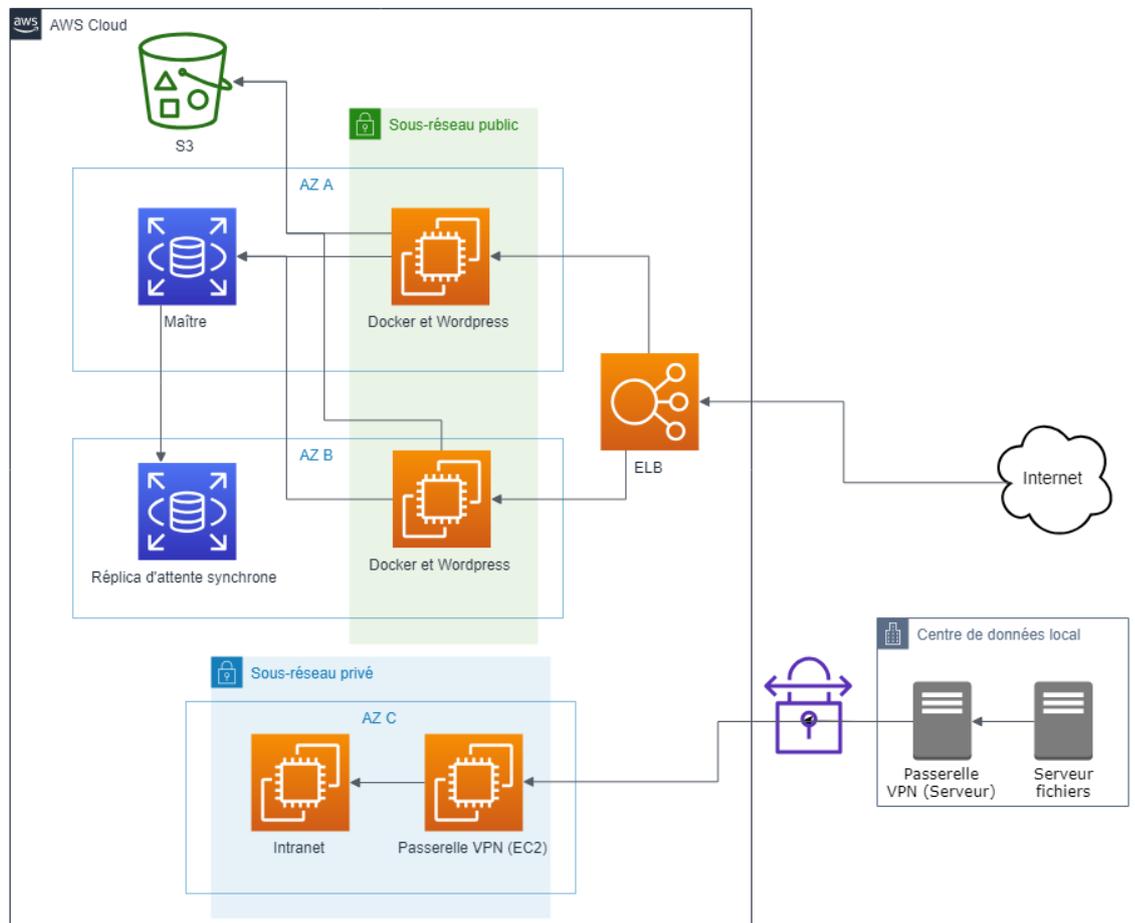
a) – Présentation du scénario

« Vous travaillez dans une petite entreprise qui se développe à l'international. L'entreprise ne dispose que de quelques machines pour la gestion de ses outils internes et pour héberger son site internet. Des incidents matériels sur le serveur de messagerie ont pénalisé l'entreprise le mois dernier. Le directeur veut éviter que la situation ne se reproduise mais réalise que fiabiliser les services en interne coûterait très cher.

L'entreprise décide donc de migrer l'ensemble de son SI vers le cloud. Les outils collaboratifs seront migrés vers Google Apps et le site de l'entreprise sera hébergé sur AWS.

L'entreprise garde simplement un serveur de fichiers dans ses locaux pour profiter de la vitesse du réseau local pour le transfert de fichiers volumineux. Ce serveur est également accessible par une liaison VPN depuis un site intranet privé hébergé sur AWS »

Schéma indicatif de l'infrastructure



b) – Instructions d'architecture

- Installez un serveur WordPress (le site de l'entreprise) sur AWS en utilisant :
 - RDS pour le stockage de la base de données
 - S3 pour le stockage des médias (via le plugin amazon-web-services)
 - EC2 et Docker pour le serveur web
 - ELB pour distribuer les requêtes sur les instances EC2
 - CloudFormation pour automatiser la création de l'infrastructure
- Tous les éléments de votre infrastructure publique devront être répartis sur plusieurs zones de disponibilité (multi-AZ). Vous utiliserez le service ELB pour la répartition des requêtes vers les différentes zones de disponibilité (AZ).
- Vous monterez une instance EC2 destinée à héberger l'application intranet sur un sous-réseau privé. Dans le cadre de ce cours, le contenu de l'intranet sera une simple page web HTML.
- En local sur votre machine, vous créerez deux machines virtuelles Linux : une pour le serveur de fichiers et une pour le serveur VPN.
- Vous établirez une liaison VPN entre votre serveur VPN local et le sous-réseau privé AWS via une instance EC2.
- Vous mettrez en place de l'auto-scaling sur les instances EC2 pour augmenter le nombre de machines dès que la charge CPU des serveurs atteint 80% en moyenne sur 5 minutes et vous veillerez à être informé par un mail à chaque fois que l'événement survient.
- Vous évalueriez les coûts de votre infrastructure AWS à partir de différentes hypothèses d'usage que vous formulerez.

II - Construire le Réseau AWS :

a) Choisir sa région :

Il est nécessaire de définir une région dans laquelle notre infrastructure devra être installée et déployée c'est pourquoi il est recommandé de répondre à différents critères comme la région des utilisateurs finaux (temps de réponse), les coûts horaires entre les régions, les lois du droit international (ex : respect des normes en vigueur en fonction du pays pour la prestation de service gratuite ou payante par application.) Dans ce projet nous choisirons la zone « Ohio » pour laquelle le coût horaire des machines est le plus intéressant.

t2.nano

1

Variable

0,5 Gio

EBS uniquement

0,0058 USD par heure

Pour en savoir plus sur les tarifications amazon :

<https://aws.amazon.com/fr/emr/pricing/>

<https://aws.amazon.com/fr/ec2/pricing/on-demand/>

b) VPC & Sous-Réseaux

Un VPC (Virtual Private Cloud) est un réseau privé virtuel. Il s'agit d'une section réseau isolée, dans laquelle nous pourrions mettre en place les machines. Un VPC réside dans une région AWS.

Un VPC couvre toutes les zones de disponibilité de la région. Au sein d'un VPC, nous pourrions déployer un ou plusieurs **sous-réseaux** dans chaque zone de disponibilité.

Un sous-réseau est une portion d'un VPC, et par conséquent correspond à un sous-ensemble du bloc CIDR du VPC. C'est pourquoi il est fortement recommandé d'adresser au VPC un masque de sous réseau plus large que des masques des sous réseaux.

Par exemple un VPC en /16 et ses sous réseaux en /24.

The screenshot displays the AWS Management Console interface for VPCs. At the top, it shows 'Vos VPC (2)' with a search filter and a 'Créer un VPC' button. Below this is a table listing two VPCs:

Name	ID du VPC	État	CIDR IPv4	CIDR IPv6	Groupe IPv6	Jeu d'options DHCP
docker-wp-vpc	vpc-fbb26590	Available	172.31.0.0/16	-	-	dopt-905cf5fb
intranet-vpn-vpc	vpc-046620c905a51ed30	Available	172.16.0.0/16	-	-	dopt-905cf5fb

The detailed view for 'vpc-fbb26590 / docker-wp-vpc' is shown below, with tabs for 'Détails', 'Blocs CIDR', 'Journaux de flux', and 'Balises'. The 'Détails' tab is active, showing the following information:

Propriété	Valeur	Propriété	Valeur
ID du VPC	vpc-fbb26590	Noms d'hôte DNS	Activé
Location	Default	Résolution DNS	Activé
VPC par défaut	Oui	Table de routage	rtb-3c73ef57
Owner ID	784822470424	ACL réseau	acl-d5aa18be
		Groupe IPv6	-
		Bloc CIDR IPv6	-
		Jeu d'options DHCP	dopt-905cf5fb
		Bloc CIDR IPv4	172.31.0.0/16

Créer le sous-réseau Actions

Filter par balises et attributs ou rechercher par mot clé

Name	ID de so	État	VPC	Bloc CIDR IPv	Adresses IPv4	Le b	Zone de disponi	ID de zone de dispo	Table de routage	ACL réseau
docker-wp-subnet-public-za	subnet-...	available	vpc-fbb26590 docker-w...	172.31.0.0/20	4089	-	us-east-2a	use2-az1	rtb-3c73ef57	acl-d5aa18be
docker-wp-subnet-public-zb	subnet-...	available	vpc-fbb26590 docker-w...	172.31.16.0/20	4088	-	us-east-2b	use2-az2	rtb-3c73ef57	acl-d5aa18be
intranet-vpn-subnet-privé-zc	subnet-...	available	vpc-046620c905a51ed30...	172.16.1.0/24	250	-	us-east-2c	use2-az3	rtb-0c17eacc082b8a8e4 intra...	acl-0285d4fceed12c
Intranet-vpn-subnet-public-zc-optionnel	subnet-...	available	vpc-046620c905a51ed30...	172.16.0.0/24	251	-	us-east-2c	use2-az3	rtb-0f0bc0bb23e1983f2	acl-0285d4fceed12c

Sous-réseau: subnet-06ad7e1e6a3508ca4

Description Journaux de flux Table de routage ACL réseau Balises Partage

ID de sous-réseau	subnet-06ad7e1e6a3508ca4	État	available
VPC	vpc-046620c905a51ed30 intranet-vpn-vpc	Bloc CIDR IPv4	172.16.1.0/24
Adresses IPv4 disponibles	250	Le bloc d'adresse CIDR IPv6	-
Zone de disponibilité	us-east-2c (use2-az3)	Table de routage	rtb-0c17eacc082b8a8e4 intranet-vpn-sub-route
ACL réseau	acl-0285d4fceed12cc20	Sous-réseau (subnet) par défaut	Non
Attribuer automatiquement une adresse IPv4 publique	Non	Auto-assign customer-owned IPv4 address	No
Customer-owned IPv4 pool	-	Attribuer automatiquement une adresse IPv6	Non
Outpost ID	-	Propriétaire	784822470424

Il est également possible de louer une adresse IP publique statique appelée « elastic ip » que nous pourrions si nécessaire attacher à une instance ou à une interface réseau.

Adresses IP Elastic (2)

Filter les adresses IP Elastic

Name	Adresse IPv4 allouée	Type	ID d'allocation	ID d'instance associée	Adresse IP privée	ID d'association
-	3.13.18.100	Adresse IP publique	eipalloc-000ad2ceb275165b	i-02ccda7fbd5488b25	172.31.15.5	eipassoc-0f7f95e90aa710
-	3.130.232.188	Adresse IP publique	eipalloc-0dcad052127918a5e	-	-	-

Afin de communiquer avec l'extérieur, AWS met en place des passerelles de sortie ; il en existe deux sortes :

- les **passerelles Internet**, permettant de router du trafic réseau dans et hors de votre VPC. Les passerelles Internet supportent l'IPv4 et l'IPv6 ;
- les **passerelles de sortie** uniquement, permettant de router du trafic réseau uniquement en sortie de votre VPC. Cela peut être une alternative à la création d'une instance NAT ou d'une passerelle NAT. Les passerelles de sortie uniquement ne supportent que l'IPv6.

Après avoir créé une passerelle internet, nous devons l'attacher à notre VPC.

VPC > Passerelles Internet > igw-b3bccddb

igw-b3bccddb / infra-gw-wp Actions

Détails Infos

ID de passerelle Internet	État	ID du VPC	Propriétaire
igw-b3bccddb	Attached	vpc-fbb26590 docker-wp-vpc	784822470424

III - Systèmes EC2, RDS, S3

a) Installation & Sécurité

Les machines ou instances de distribution (AMI) Linux ou Windows s'installent et se paramètrent grâce au service « EC2 »

Un ensemble d'options peuvent être configurés afin de répondre à différents besoins (Type d'instance : mémoire, cpu, stockage ; groupe de sécurité, zone de disponibilité)

Dans ce projet nous utiliserons l'offre gratuite disponible (t2.micro)

1. Choisir l'AMI 2. Choisir un type d'instance 3. Configurer l'instance 4. Ajouter le stockage 5. Ajouter des balises 6. Configurer le groupe de sécurité 7. Vérification

Étape 2 : Choisir un type d'instance

Amazon EC2 fournit un vaste éventail de types d'instances optimisés pour différents cas d'utilisation. Les instances sont des serveurs virtuels qui peuvent exécuter des applications. Les types d'instances se composent de différentes combinaisons de processeur, de mémoire, de stockage et de capacité réseau, et vous offrent une flexibilité dans le choix de l'association de ressources adaptées à vos applications. [En savoir plus](#) à propos des types d'instances et de la manière dont ils peuvent répondre à vos besoins informatiques.

Filtrer par: All instance families Génération actuelle Afficher / Masquer les colonnes

Actuellement sélectionné : t2.micro (1 ECU, 1 vCPU, 2.5 GHz, -, 1 Gio mémoire, EBS uniquement)

	Famille	Type	vCPU	Mémoire (Gio)	Stockage d'instance (Go)	Disponible en version optimisée pour EBS	Performances réseau	Prise en charge IPv6
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS uniquement	-	Faibles à modérées	Oui
<input checked="" type="checkbox"/>	t2	t2.micro <small>Offre à l'offre gratuite</small>	1	1	EBS uniquement	-	Faibles à modérées	Oui
<input type="checkbox"/>	t2	t2.small	1	2	EBS uniquement	-	Faibles à modérées	Oui

De la même façon nous paramétrons la base de données dans le service « RDS »

RDS > Bases de données

Bases de données

Ressources de groupe

Identifiant de base de données	Rôle	Moteur	Région et AZ	Taille	Statut	Processeur	Activité actuelle	Maintenance	VPC
<input checked="" type="radio"/> database-1	Principale	MariaDB	us-east-2a	db.t2.micro	Disponible	1.69%	1 Connexions		vpc-fbl
<input type="radio"/> replicadb	Réplica	MariaDB	us-east-2b	db.t2.micro	Disponible	1.17%	0 Connexions		vpc-fbl

Il est également important d'apporter un maximum de sécurité en filtrant le Trafic entrant/sortant.

C'est pourquoi nous devons également configurer les groupes de sécurité en fonction des besoins comme accepter les requêtes HTTP/HTTPS (80 / 443) pour les réseaux publics ou sera installé le docker wp en zone multi a-z ou bien autoriser l'écoute unique du port 3306 pour le GS base de donnée de manière à éviter la découverte du réseau par icmp par exemple et autoriser uniquement la connexion des machines EC2 Docker/Wordpress à la BDD.

Groupes de sécurité (1/6) Informations

Filter les groupes de sécurité

Name	ID du groupe de sécu...	Nom du groupe de ...	ID de VPC	Description	Propriétaire	Nombre de règles e...	Nombr
<input type="checkbox"/> Security-Load-Balancer	sg-0573a1f3b3035693f	load-balancer-wizard-1	vpc-fbb26590	load-balancer-wizard-...	784822470424	6 Entrées d'autorisation	1 Entré
<input type="checkbox"/> Security-WP-AutoScalling	sg-05c75729845f8f769	launch-wizard-1	vpc-fbb26590	launch-wizard-1 create...	784822470424	5 Entrées d'autorisation	1 Entré
<input type="checkbox"/> Security-WP-za	sg-06e44c3dc9af8a98f	Docker-WP-za	vpc-fbb26590	launch-wizard-2 create...	784822470424	5 Entrées d'autorisation	1 Entré
<input type="checkbox"/> Security-Intranet	sg-0938df59c8c28eeac	default	vpc-046620c905a51ed30	default VPC security gr...	784822470424	8 Entrées d'autorisation	1 Entré
<input type="checkbox"/> Security-WP-zb	sg-0de53e964872108c1	Docker-WP-zb	vpc-fbb26590	launch-wizard-2 create...	784822470424	5 Entrées d'autorisation	1 Entré
<input checked="" type="checkbox"/> Security-BDD	sg-e448c79d	default	vpc-fbb26590	default VPC security gr...	784822470424	2 Entrées d'autorisation	1 Entré

sg-e448c79d - default

Détails Règles entrantes Règles sortantes Balises

Règles entrantes

Type	Protocole	Plage de ports	Source	Description - facultatif
MYSQL/Aurora	TCP	3306	sg-05c75729845f8f769 (launch-wizard-1)	-
MYSQL/Aurora	TCP	3306	sg-e448c79d (default)	-

b) Configuration Docker & Wordpress :

Voici la liste des commandes utilisées afin d'installer docker composer et la création d'une base de données sécurisé avec un utilisateur non root.

Installation de Docker :

```

sudo apt-get update
sudo apt install apt-transport-https ca-certificates curl software-properties-common
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/debian stretch
stable"
apt-cache policy docker-ce
sudo apt install docker-ce

sudo curl -L https://github.com/docker/compose/releases/download/1.25.5/docker-compose-
`uname -s`-`uname -m` -o /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose

```

Création d'une base de donnée dédiée à Wordpress avec utilisateurs spécifique

```

apt-get install mariadb-client

```

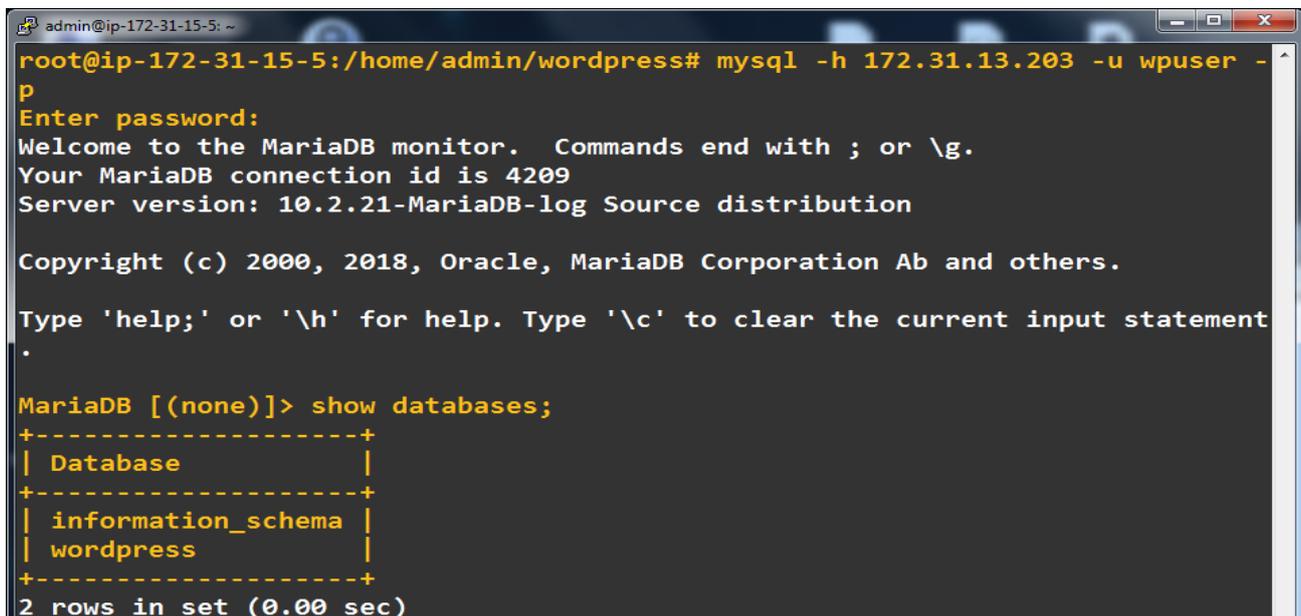
```
mysql -h database-1.czq2xzhlagls.us-east-2.rds.amazonaws.com -u admin -p (cnx à la bdd rds parent : mot de passe : awsawsocrocr)
```

```
CREATE DATABASE wordpress CHARACTER SET utf8 COLLATE utf8_general_ci; (creation de la bdd "wordpress" dans la bdd parent)
```

```
GRANT ALL PRIVILEGES ON wordpress.* TO 'wpuser'@'%' identified by 'wppassword';
```

```
mysql -h 172.31.13.203 -u wpuser -p ( connection a la BDD RDS )
```

```
( wppassword )
```



```
admin@ip-172-31-15-5: ~
root@ip-172-31-15-5:/home/admin/wordpress# mysql -h 172.31.13.203 -u wpuser -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 4209
Server version: 10.2.21-MariaDB-log Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement
.

MariaDB [(none)]> show databases;
+-----+
| Database                |
+-----+
| information_schema      |
| wordpress                |
+-----+
2 rows in set (0.00 sec)
```

```
nano docker-compose.yml ( creation de l'installation manuel wordpress docker )
```

Configuration de docker afin de connecter automatiquement le service cms wordpress sous conteneur à sa bdd via Docker-compose.yml

```
version: '2'

services:

  wordpress:

    image: wordpress:latest

    ports:

      - "80:80"

    restart: always
```

environment:

WORDPRESS_DB_HOST: database-1.czq2xzhlagls.us-east-2.rds.amazonaws.com:3306

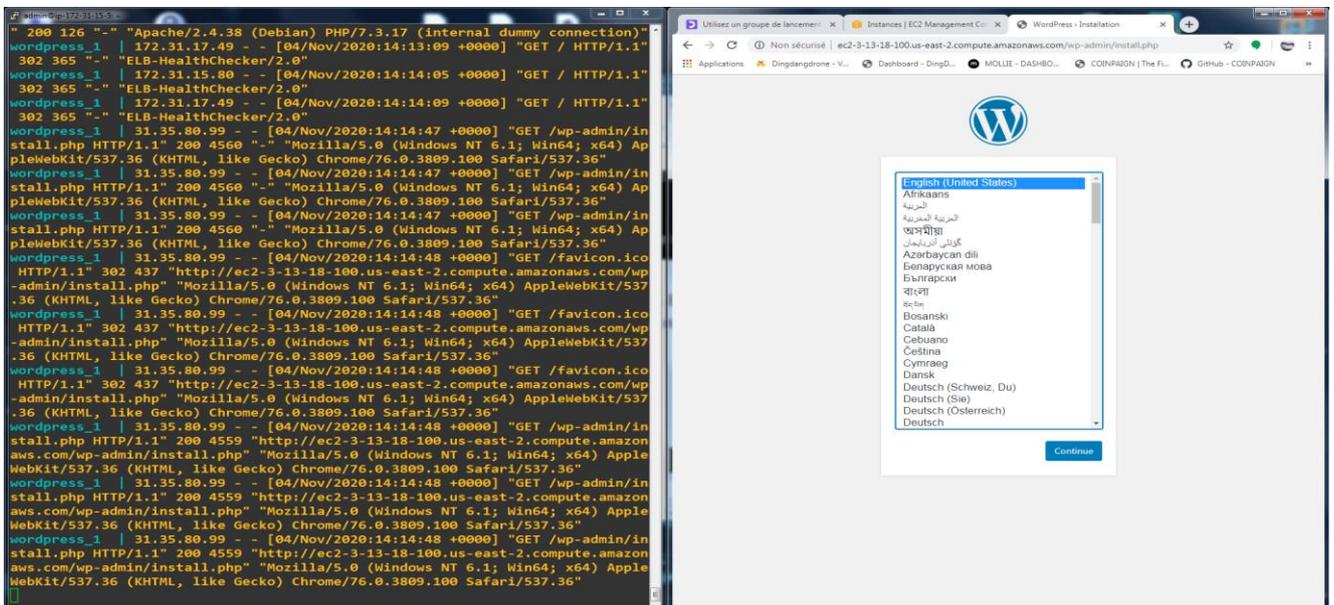
WORDPRESS_DB_USER: wpuser

WORDPRESS_DB_PASSWORD: wppassword

WORDPRESS_DB_DATABASE: wordpress

docker-compose up -d (lancement du docker compose (sudo avant si erreur daemon))

Les connexions HTTP sont désactivé dans le VPC par default. Il faut donc les autoriser à partir du groupe de sécurité et en modifiant les règles de connexions entrantes en ajoutant le protocole http



c) Snapshots & AMI personnalisée(s)

De manière à reproduire facilement notre machine ec2 de la zone A sur la zone B sans avoir à refaire l'ensemble des configurations précédentes (système uniquement. Certains paramètres devront être ajouter à cette nouvelle instance comme le groupe de sécurité ou la modification des instances à intégrer dans l'auto-scaling groupe) il faut créer une sauvegarde à un instant t du notre disque dur aussi appelé snapshot.

Créer un instantané Actions

M'appartenant Filtre par balises et attributs ou rechercher par mot clé

Name	ID d'instantané	Taille	Statut	Démarré(e)	Progression	Chiffrement	ID de clé KMS	Alias de clé KM-
snap-057443fc12ec...	8 Gio	completed	28 avril 2020 02:37:26 UTC+2	disponible (100 %)	Non chiffré			

Instantané: snap-057443fc12ec3a683

Description Autorisations Balises

ID d'instantané	snap-057443fc12ec3a683	Progression	100%
Statut	completed	Capacité	8 Gio
Volume	vol-04ea51370d64f7028	Chiffrement	Non chiffré
Démarré(e)	28 avril 2020 02:37:26 UTC+2	ID de clé KMS	
Propriétaire	784822470424	Alias de clé KMS	
Codes produit	-	ARN de clé KMS	
Description	Website Zone B	Restauration rapide d'instantanés	-

Après quoi il nous restera à créer une image système basé sur notre snapshot. Celle-ci sera utilisé pour choisir notre AMI personnalisé lorsque nous devons déployer une nouvelle instance dans la zone B ou une mise à l'échelle automatisé en cas de Trafic élevé.

Lancer EC2 Image Builder Actions

M'appartenant Filtre par balises et attributs ou rechercher par mot clé

Name	Nom d'AMI	ID d'AMI	Source	Propriétaire	Visibilité	Statut	Date de création	Plateforme	Type de périp-	Virtualisation
DockerWP E...	Website - B	ami-0c565851243d3f811	784822470424/...	784822470424	Privé	available	28 avril 2020 02:43:03 UTC+2	Other Linux	eba	hvm

Image: ami-0c565851243d3f811

Détails Autorisations Balises

ID d'AMI	ami-0c565851243d3f811	Nom d'AMI	Website - B
Propriétaire	784822470424	Source	784822470424/Website - B
Statut	available	Raison de l'état	-
Date de création	28 avril 2020 02:43:03 UTC+2	Platform details	Linux/UNIX
Architecture	x86_64	Usage operation	RunInstances
Type d'image	machine	Type de virtualisation	hvm
Description	Website Zone B	Nom du périphérique racine	/dev/sda1
Type de périphérique racine	eba	ID de disque RAM	-
ID du noyau	-	Codes produit	-
Périphériques de stockage en mode bloc	/dev/sda1=nap-057443fc12ec3a683:8:true:gp2		

Modifier

d) Bucket S3 & Role IAM

Pour connecter ses compartiments S3 depuis nos instances EC2, il nous faut :
Créer et joindre un rôle de profil AWS Identity and Access Management (IAM) aux instances qui autorise l'accès à Amazon S3.

Vérifier que la stratégie de compartiment S3 ne possède pas de stratégie refusant l'accès.
Confirmer la connectivité réseau entre l'instance EC2 et Amazon S3.

Services

Identity and Access Management (IAM)

Rôles > s3

Récapitulatif

Supprimer le rôle

ARN de rôle	arn:aws:iam::784822470424:role/s3
Description du rôle	Allows EC2 instances to call AWS services on your behalf. Modifier
ARN des profils d'instance	arn:aws:iam::784822470424:instance-profile/s3
Chemin	/
Heure de création	2020-10-25 17:47 UTC+0100
Dernière activité	2020-10-27 13:52 UTC+0100 (8 jours il y a)
Durée maximale de la session	1 heure Modifier

Autorisations Relations d'approbation Balises Access Advisor Révoquer les sessions

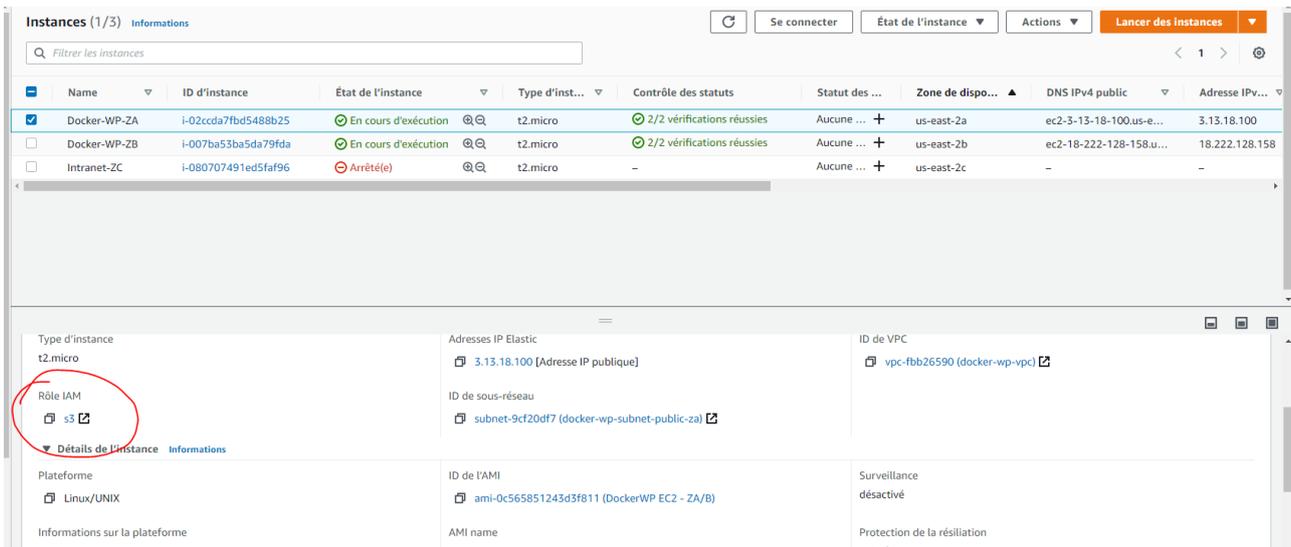
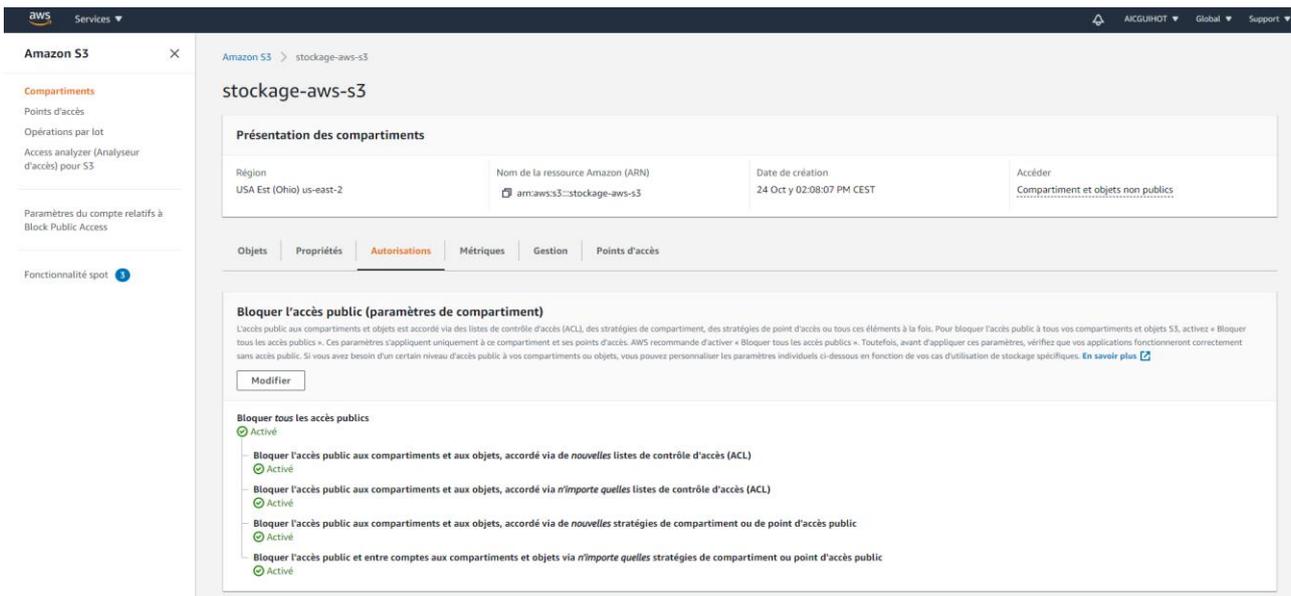
Permissions policiées (1 stratégie appliquée)

Attacher des stratégies Ajouter une stratégie en ligne

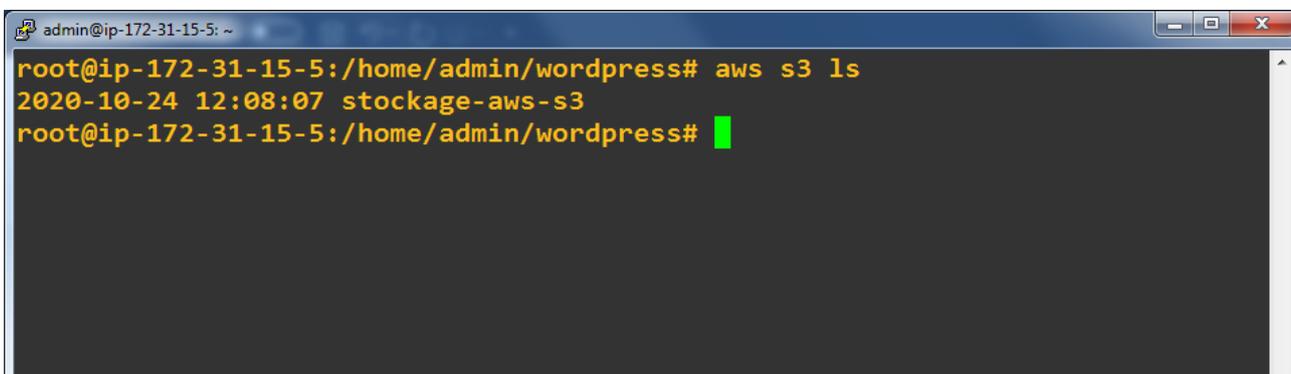
Nom de la stratégie	Type de stratégie
AmazonS3FullAccess	Stratégie gérée par AWS

Permissions boundary (not set)

Dans notre scénario il est bien sur évident que le bucket ne doit pas être accessible depuis l'extérieur mais uniquement depuis nos instance EC2 comme serveur de média.



Puis nous pouvons vérifier l'accès du compartiment S3 (créé depuis la console via le service s3) depuis nos instances EC2 de la zone A/B



IV - Load Balancer & Mise à l'échelle

AWS fournit un **service d'équilibrage de charge** ou **Load Balancer**, c'est-à-dire un composant qui va distribuer les requêtes réseau vers différentes machines, afin de **répartir** la charge sur plusieurs éléments de l'infrastructure, permettant de paralléliser les traitements.



a) Configuration du Load Balancing

Les options indispensables dans notre cas sont d'ajouter la disponibilité multi a-z grâce à leurs sous-réseaux respectif de manière à équilibrer la charge entre les machines EC2 des deux zones.

Créer un équilibreur de charge Actions

Filter par balises et attributs ou rechercher par mot clé

Nom	Nom du DNS	État	ID de VPC	Zones de disponibilité	Type	Créé le	Surveillance
ELB	ELB-275071419.us-east-2.elb.amazonaws.com	active	vpc-fbb26590	us-east-2b, us-east-2a	application	26 octobre 2020 15:25:46 U...	

Équilibreur de charge: ELB

Description Écouteurs Surveillance Services intégrés Balises

Configuration de base

Nom: ELB
ARN: arn:aws:elasticloadbalancing:us-east-2:784822470424:loadbalancer/app/ELB39693293922bfe6c
Nom du DNS: ELB-275071419.us-east-2.elb.amazonaws.com (Enregistrement A)
État: active
Type: application
Méthode: internet-facing
Type d'adresse IP: ipv4 (Modifier le type d'adresse IP)
VPC: vpc-fbb26590
Zones de disponibilité: subnet-7de9dc07 - us-east-2b (Adresse IPv4: Attribuées par AWS), subnet-9cf20df7 - us-east-2a (Adresse IPv4: Attribuées par AWS) (Modifier les sous-réseaux (subnets))
Zone hébergée: Z3AADJGX6KTTL2
Heure de création: 26 octobre 2020 15:25:46 UTC+1

EC2 > Groupes cibles > site Internet

site Internet

arn: aws: elasticloadbalancing: us-east-2: 784822470424: groupe cible / site web / 37ed6b66416f8e9f

[Supprimer](#)

Configuration de base

Type de cible instance d'instance	Protocole: Port HTTP : 80 Version du protocole HTTP1	VPC vpc-fbb26590 🔗	Équilibreur de charge ELB 🔗
--------------------------------------	---	---------------------------------------	--

Attributs

[Modificateur](#)

Permanence Activée Durée de la permanence 4 heures Algorithme d'équilibrage de charge Tourniquet (round robin)	Retard d'annulation d'enregistrement 300 secondes Durée de démarrage lent 0 seconde
---	--

Détails du groupe | **Cibles** | Surveillance | Balises

Cibles enregistrées (2)

[🔄](#) [Annuler l'enregistrement](#) [Enregistrer les cibles](#)

<input type="checkbox"/>	ID d'instance	Nom	Port	Zone	Détails du statut
<input type="checkbox"/>	i-02ccda7fbd5488b25	Docker-WP-ZA	80	us-east-2a	Target is in the stopped state
<input type="checkbox"/>	i-007ba53ba5da79fda	Docker-WP-ZB	80	us-east-2b	Target is in the stopped state

Vérification du fonctionnement ELB avec algorithme du tourniquet & permanence de 4H/Session

← → Non sécurisé | elb-27507419.us-east-2.elb.amazonaws.com/wp-admin/install.php

Applications Dingdangrone - V... Dashboard - DingD... MOLLIE - DASHBO... COINPAIGN | The Fl... GitHub - COINPAIGN Quest Nuisibles - A... Google Adwords Google Analytics My Account Panel ~... bank.to | Tableau d... OpenClassRooms

de mise à l'échelle selon des alarmes basées sur des métriques d'instance, la capacité souhaitée, minimal & maximal de notre déploiement automatisé.

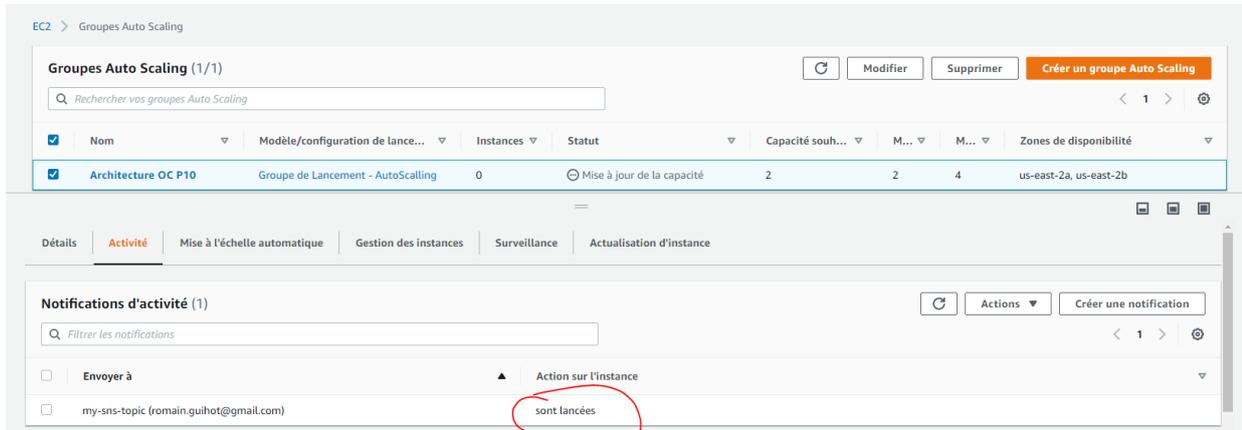
The screenshot displays the 'Groupes Auto Scaling' page in the AWS console. At the top, there's a search bar and buttons for 'Modifier', 'Supprimer', and 'Créer un groupe Auto Scaling'. Below is a table with columns for 'Nom', 'Modèle/configuration de lance...', 'Instances', 'Statut', 'Capacité souh...', 'M...', 'M...', and 'Zones de disponibilité'. The first row shows 'Architecture OC P10' with 0 instances and a status of 'Mise à jour de la capacité'. Below the table are tabs for 'Détails', 'Activité', 'Mise à l'échelle automatique', 'Gestion des instances', 'Surveillance', and 'Actualisation d'instance'. The 'Détails du groupe' section shows fields for 'Capacité souhaitée' (2), 'Capacité minimale' (2), 'Capacité maximale' (4), 'Nom du groupe Auto Scaling' (Architecture OC P10), 'Date de création' (Mon May 11 2020 13:44:02 GMT+0200), and 'Amazon Resource Name (ARN)'. The 'Configuration de lancement' section shows 'ID d'AMI' (ami-0c565851243d3f811), 'Groupes de sécurité' (sg-06e44c3dc9af8a98f), and 'Type d'instance'.

The screenshot shows the 'Stratégies de mise à l'échelle' page for the 'Architecture OC P10' group. It features a search bar, 'Actions' dropdown, and 'Ajouter une stratégie' button. Two strategy cards are visible: 'AutoScaling_WP_ZA' and 'AutoScaling_WP_ZB'. Both are 'Mise à l'échelle simple' and 'Désactivée'. The 'AutoScaling_WP_ZA' strategy is triggered when 'CPU>80%_WPZA' alarm is active, adding 1 unit of capacity. The 'AutoScaling_WP_ZB' strategy is triggered when 'CPU>80%_WPZB' alarm is active, also adding 1 unit of capacity. Both strategies include a 300-second delay before scaling.

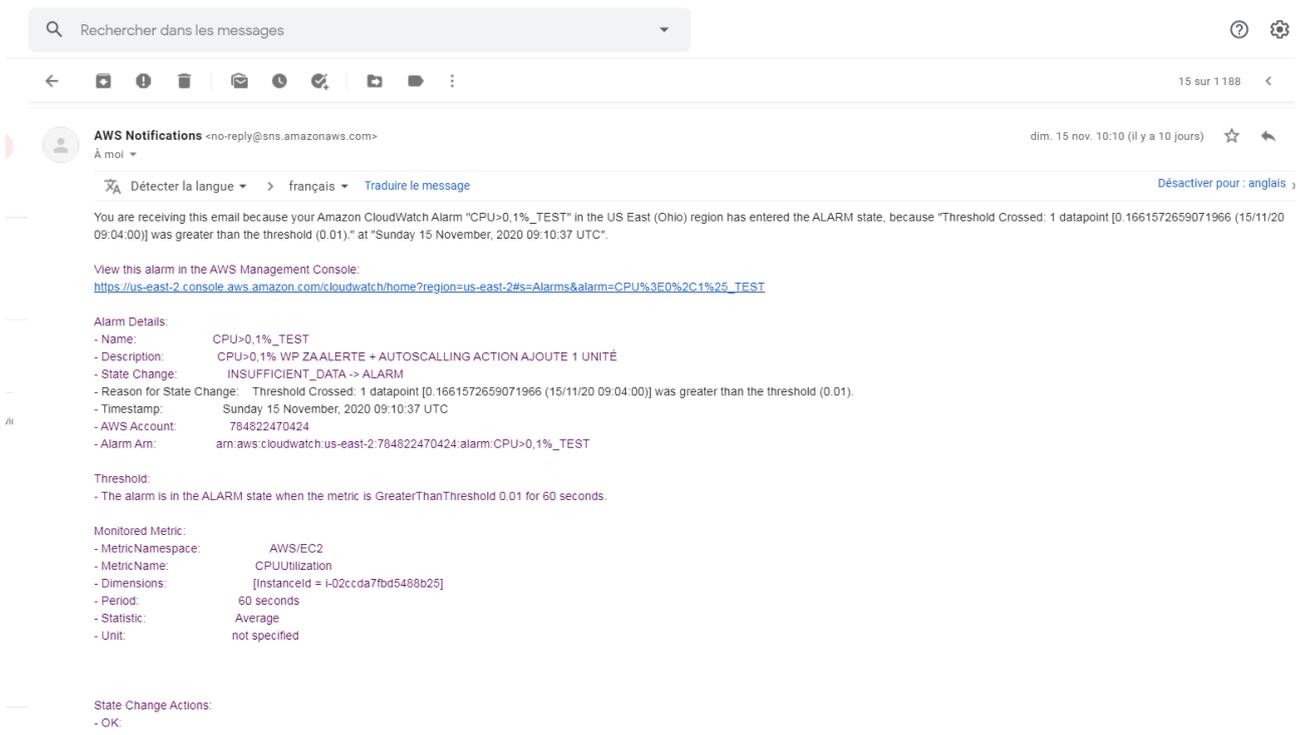
Enfin nous veillerons à être averti par notification email dès qu'une machine est lancée par effet de mise à l'échelle selon notre stratégie de CPU>80%.

Ces stratégies nécessitent la création d'alarmes « cloudwatch » avec action. Cette prise de mesure ou action est activé dès lors que le seuil de l'alarme est dépassé afin d'ajouter une unité de capacité supplémentaire à notre groupe de mise à l'échelle.

C'est à dire dimensionner de façon automatisée et donc « scaler » notre infrastructure (élastique) pour répondre aux besoins croissants d'une application qui fonctionne et qui obtient toujours plus de trafic, ou simplement des « pics » à des moments clés.

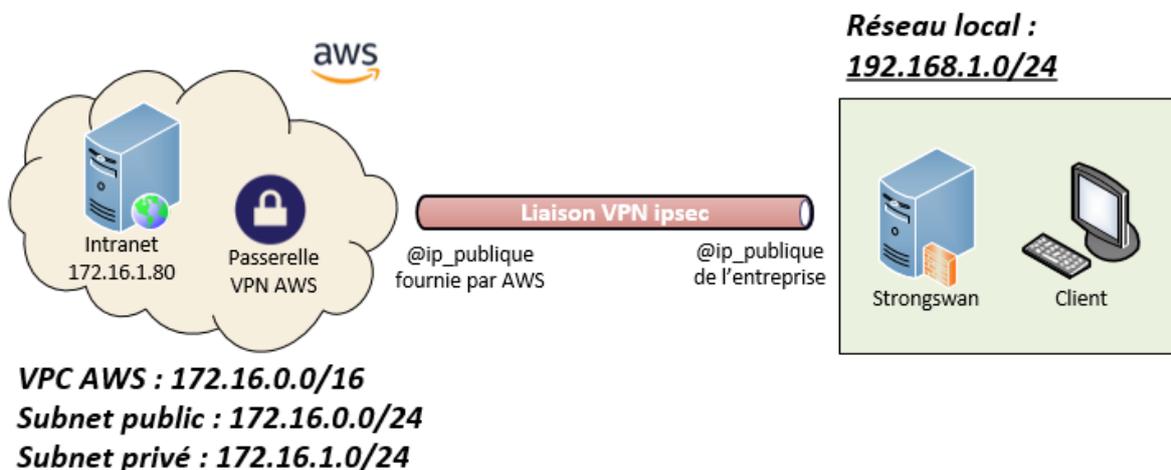


Un test a été réalisé avec la création d'une stratégie pour CPU>0,1% de manière à vérifier la scalabilité, et les notifications 😊



V - VPN site à site & Intranet isolé :

Le but étant de faire communiquer notre réseau d'entreprise locale (simulé par des VM de distribution debian sur mon réseau privé) à l'intranet hébergé chez aws dans un sous réseau privé isolé et inaccessible autrement que par le tunnel VPN.



Coté AWS, il est indispensable de créer un sous réseau privée et un sous réseau public qui sera utilisé uniquement afin de sortir sur internet provisoirement pour installer les packets d'un frontal web (apache, php : service httpd)

On pensera également à créer un groupe de sécurité pour n'autoriser que notre réseau privée d'entreprise à communiquer et définir les routes associées.

Name	ID de so	État	VPC	Bloc CIDR IPv	Adresses IPv4	Le b	Zone de disponi	ID de zone de dispo	Table de routage	ACL réseau
docker-wp-subnet-public-za	subnet...	available	vpc-fbb26590 docker-w...	172.31.0.0/20	4088	-	us-east-2a	use2-az1	rtb-3c73ef57	acl-d5aa18be
docker-wp-subnet-public-zb	subnet...	available	vpc-fbb26590 docker-w...	172.31.16.0/20	4088	-	us-east-2b	use2-az2	rtb-3c73ef57	acl-d5aa18be
intranet-vm-subnet-privé-zc	subnet...	available	vpc-046620c905a51ed30...	172.16.1.0/24	250	-	us-east-2c	use2-az3	rtb-0c17eacc082b8a8e4 Intra...	acl-0285d4feed12x
intranet-vm-subnet-public-zc-optionnel	subnet...	available	vpc-046620c905a51ed30...	172.16.0.0/24	251	-	us-east-2c	use2-az3	rtb-0fdb0bb23e1983f2	acl-0285d4feed12x

Sous-réseau: subnet-06ad7e1e6a3508ca4

Table de routage: rtb-0c17eacc082b8a8e4 | intranet-vm-priv-sub-route

Destination	Cible
192.168.1.0/24	vgw-01ad971f696ec61fa
172.16.0.0/16	local

Comme pour AWS, notre serveur VPN local de distribution debian devra être en mode routeur.

En editant le fichier /etc/sysctl.conf

```
net.ipv4.ip_forward = 1
```

Explication de l'installation de Strongswan sur notre serveur VPN local avec la commande suivante :

```
apt-get install strongswan -y
```

Ensuite on va préparer la configuration de la future liaison VPN. En éditant le fichier /etc/ipsec.conf

```
conn aws

    type=tunnel

    authby=secret

    left=192.168.1.10 # Adresse IP Du serveur VPN Local

    leftid=31.35.80.99 # Adresse Public Local Entreprise ( Fournit par mon FAI )

    right=3.15.75.58 # Adresse Public AWS

    leftauth=psk

    rightauth=psk

    keyexchange=ikev1

    ike=aes128-sha1-modp1024

    ikelifetime=8h

    esp=aes128-sha1-modp1024

    lifetime=1h

    keyingtries=%forever

    leftsubnet=192.168.1.0/24 # Réseau Privé Local Entreprise

    rightsubnet=172.16.1.0/24 # Réseau Privé AWS Zone C

    dpddelay=10s

    dpdtimeout=30s

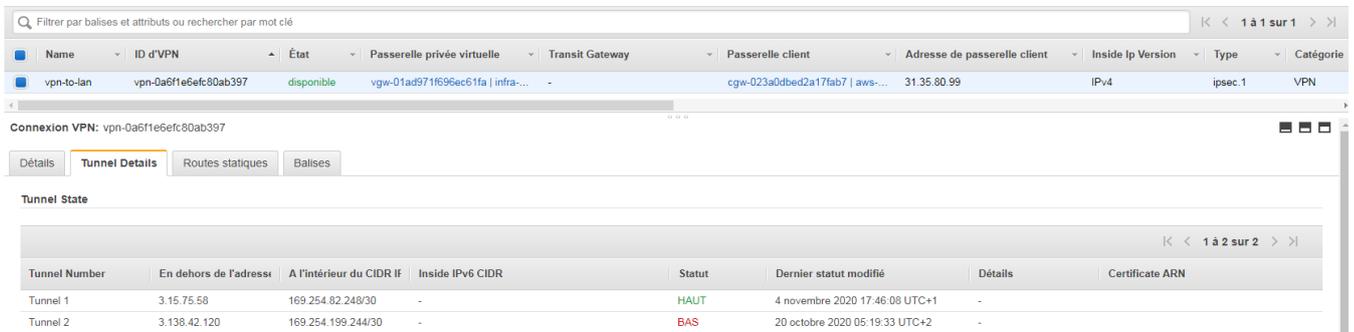
    dpdaction=restart

    auto=start
```

On edite également le fichier /etc/ipsec.secrets afin de procéder à l'échange de clé entre nos deux points de terminaison.

```
31.35.80.99 3.15.75.58 : PSK "NBr3REZUHoC8pgjooi6C0VNoHuToX_Oq"
```

Après un redémarrage du vpn client/serveur notre tunnel vpn est montée et nous pouvons le vérifier des deux côtés de notre infrastructure (nous constatons également que nos routes & groupes de sécurité fonctionne correctement puisque depuis le routeur vpn local nous accédons aux travers du tunnel à notre intranet en 172.16.1.80 :



Connexion VPN: vpn-0a6f1e6efc80ab397

Détails Tunnel Details Routes statiques Balises

Tunnel State

Tunnel Number	En dehors de l'adresse	A l'intérieur du CIDR IF	Inside IPv6 CIDR	Statut	Dernier statut modifié	Détails	Certificate ARN
Tunnel 1	3.15.75.58	169.254.82.248/30	-	HAUT	4 novembre 2020 17:46:08 UTC+1	-	-
Tunnel 2	3.138.42.120	169.254.199.244/30	-	BAS	20 octobre 2020 05:19:33 UTC+2	-	-

```
ocr@Client-VPN-VM: ~  
root@Client-VPN-VM:/home/ocr# ipsec up aws  
generating QUICK_MODE request 2989813632 [ HASH SA No KE ID ID ]  
sending packet: from 192.168.1.10[4500] to 3.15.75.58[4500] (316 bytes)  
received packet: from 3.15.75.58[4500] to 192.168.1.10[4500] (316 bytes)  
parsed QUICK_MODE response 2989813632 [ HASH SA No KE ID ID ]  
connection 'aws' established successfully  
root@Client-VPN-VM:/home/ocr# ping 172.16.1.80  
PING 172.16.1.80 (172.16.1.80) 56(84) bytes of data.  
64 bytes from 172.16.1.80: icmp_seq=1 ttl=254 time=120 ms  
64 bytes from 172.16.1.80: icmp_seq=2 ttl=254 time=126 ms  
64 bytes from 172.16.1.80: icmp_seq=3 ttl=254 time=120 ms  
64 bytes from 172.16.1.80: icmp_seq=4 ttl=254 time=120 ms  
^C  
--- 172.16.1.80 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3007ms  
rtt min/avg/max/mdev = 120.413/122.116/126.497/2.572 ms  
root@Client-VPN-VM:/home/ocr#
```

Il nous faudra bien sur définir à notre serveur de fichier local (192.168.1.20) comme passerelle par défaut l'adresse ip de notre serveur vpn local (192.168.1.10)

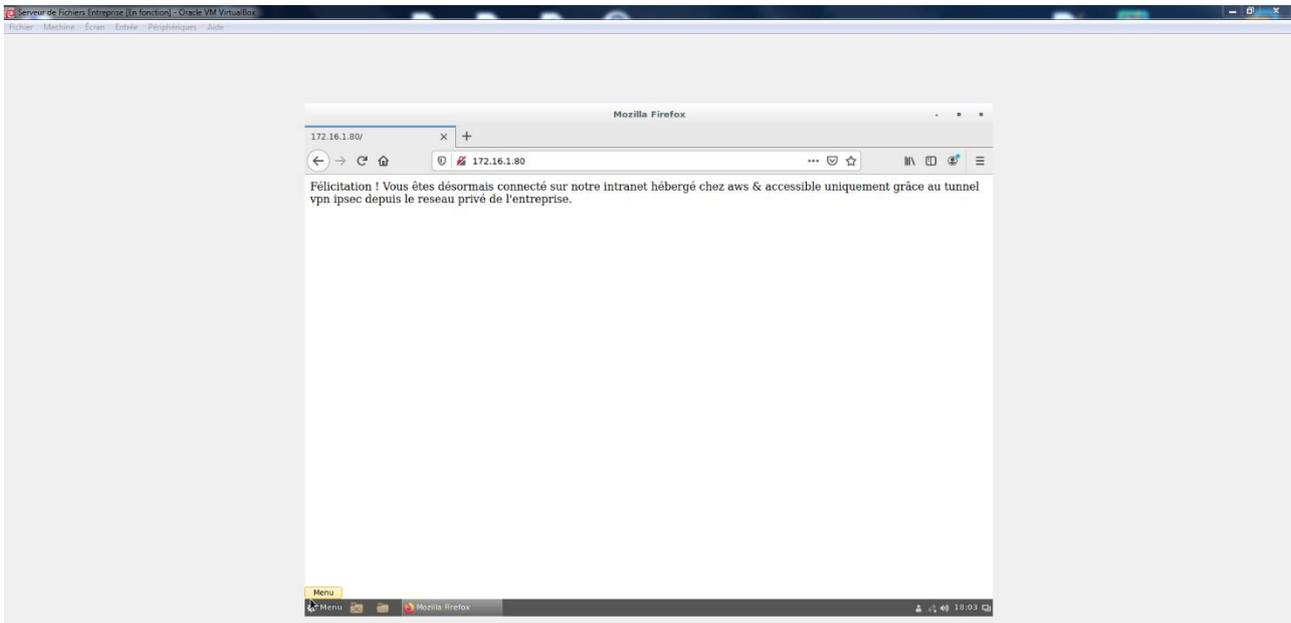
De ce fait nos postes clients ou tout autre serveur d'entreprises pourront traverser le tunnel VPN à condition d'avoir toujours comme route par défaut le serveur VPN local en mode routeur.

```

PING 172.16.1.80 (172.16.1.80) 56(84) bytes of data.
From 192.168.1.10: icmp_seq=1 Redirect Host(New nexthop: 192.168.1.254)
64 bytes from 172.16.1.80: icmp_seq=1 ttl=253 time=120 ms
From 192.168.1.10: icmp_seq=2 Redirect Host(New nexthop: 192.168.1.254)
64 bytes from 172.16.1.80: icmp_seq=2 ttl=253 time=120 ms
From 192.168.1.10: icmp_seq=3 Redirect Host(New nexthop: 192.168.1.254)
64 bytes from 172.16.1.80: icmp_seq=3 ttl=253 time=120 ms
From 192.168.1.10: icmp_seq=4 Redirect Host(New nexthop: 192.168.1.254)
64 bytes from 172.16.1.80: icmp_seq=4 ttl=253 time=121 ms
From 192.168.1.10: icmp_seq=5 Redirect Host(New nexthop: 192.168.1.254)
64 bytes from 172.16.1.80: icmp_seq=5 ttl=253 time=121 ms
From 192.168.1.10: icmp_seq=6 Redirect Host(New nexthop: 192.168.1.254)
64 bytes from 172.16.1.80: icmp_seq=6 ttl=253 time=121 ms
^C
--- 172.16.1.80 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5007ms
rtt min/avg/max/mdev = 120.524/120.993/121.355/0.306 ms
root@Client-VPN-VM:/home/ocr# route -n
Table de routage IP du noyau
Destination      Passerelle      Genmask          Indic Metric Ref       Use Iface
0.0.0.0          192.168.1.10    0.0.0.0          UG    0       0         0 enp0s3
192.168.1.0      0.0.0.0         255.255.255.0   U     0       0         0 enp0s3

```

Il nous reste à vérifier que l'on accède via le service web de notre serveur de fichier à la page intranet de notre serveur sur le réseau privé aws au travers du tunnel ipsec 😊



```
ec2-user@ip-172-16-1-80:~  
root@Client-VPN-VM:/home/ocr# ssh -i "awsocr.pem" ec2-user@172.16.1.80  
Last login: Wed Nov  4 17:07:35 2020 from 192.168.1.10  
  
  _ | _ | _ )  
  _ | (   /   Amazon Linux 2 AMI  
  _ |\_ | _ |  
  
https://aws.amazon.com/amazon-linux-2/  
[ec2-user@ip-172-16-1-80 ~]$ ping 192.168.1.20  
PING 192.168.1.20 (192.168.1.20) 56(84) bytes of data.  
64 bytes from 192.168.1.20: icmp_seq=1 ttl=63 time=120 ms  
64 bytes from 192.168.1.20: icmp_seq=2 ttl=63 time=121 ms  
64 bytes from 192.168.1.20: icmp_seq=3 ttl=63 time=121 ms  
64 bytes from 192.168.1.20: icmp_seq=4 ttl=63 time=121 ms  
64 bytes from 192.168.1.20: icmp_seq=5 ttl=63 time=121 ms  
^C  
--- 192.168.1.20 ping statistics ---  
5 packets transmitted, 5 received, 0% packet loss, time 4005ms  
rtt min/avg/max/mdev = 120.931/121.159/121.289/0.400 ms  
[ec2-user@ip-172-16-1-80 ~]$
```

VI - Script Cloud Formation :

La force de ce projet est également la capacité à concevoir entièrement son architecture grâce à la programmation avec la création de modèle aussi appelé « pile ».

Ces piles au format json ou yaml regroupes l'ensembles des ressources et dépendances nécessaires à la création complète d'une infrastructure répondant à divers besoins.

Coinbase utilise par exemple la puissance de L'IaC AWS pour centraliser, maitriser, tester, concevoir et déployer en production toutes les ressources de l'infrastructure hébergeant l'application de portefeuilles numériques basé sur les technologies de la blockchain.

- Veuillez consulter le fichier .yaml afin de prendre connaissance du modèle complet pour la création totale de l'infrastructure du projet.

```

Description: "Architecture Projet 10 OCR"
Resources:
  EC2VPC:
    Type: "AWS::EC2::VPC"
    Properties:
      CidrBlock: "172.31.0.0/16"
      EnableDnsSupport: true
      EnableDnsHostnames: true
      InstanceTenancy: "default"
      Tags:
        -
          Key: "Name"
          Value: "docker-wp-vpc"
  EC2VPC2:
    Type: "AWS::EC2::VPC"
    Properties:
      CidrBlock: "172.16.0.0/16"
      EnableDnsSupport: true
      EnableDnsHostnames: false
      InstanceTenancy: "default"
      Tags:
        -
          Key: "Name"
          Value: "intranet-vpn-vpc"
  EC2Subnet:
    Type: "AWS::EC2::Subnet"
    Properties:
      AvailabilityZone: !GetAtt EC2Instance3.AvailabilityZone
      CidrBlock: "172.31.0.0/20"
      VpcId: !Ref EC2VPC
      MapPublicIpOnLaunch: true
      Tags:
        -
          Key: "Name"
          Value: "docker-wp-subnet-public-za"
  EC2Subnet2:
    Type: "AWS::EC2::Subnet"
    Properties:
      AvailabilityZone: !GetAtt EC2Instance.AvailabilityZone
      CidrBlock: "172.31.16.0/20"
      VpcId: !Ref EC2VPC
      MapPublicIpOnLaunch: true
      Tags:
        -
          Key: "Name"
          Value: "docker-wp-subnet-public-zb"

```