

Examen CLOUD B3 YNOV LYON



Introduction :

Un appel d'offre nous a été donné afin de pouvoir répondre à une problématique, la problématique étant de pouvoir parvenir à la modernisation d'une infrastructure.

Celle-ci étant possible facilement avec les nouvelles technologies présentes actuellement, il faudra aussi préciser par ailleurs que pour des raisons économiques nous devons minimiser les coûts pour l'entreprise.

Pour répondre ainsi à ces demandes là nous utiliserons les solutions Cloud fournies par Amazon, à savoir AWS.

Création des VPC :

Pour rappel un VPC va nous permettre de lancer des ressources AWS dans un réseau virtuel que nous aurons nous même défini. Ce même réseau virtuel va ressembler à un réseau de type « classique » que nous pouvons utiliser dans notre propre datacenter etc.

Dans notre cas nous allons donc créer un VPC dans la zone Virginia avec les adresses suivantes :

✓ 10.10.0.0/16

Étape 2 : VPC avec un seul sous-réseau public

Bloc d'adresse CIDR IPv4:* (65531 adresses IP disponibles)

Bloc d'adresse CIDR IPv6: ☒ Pas de bloc d'adresse CIDR IPv6
☐ Bloc CIDR IPv6 fourni par Amazon

Nom du VPC:

Le bloc d'adresse CIDR IPv4 du sous-réseau public:* (251 adresses IP disponibles)

Zone de disponibilité:*

Nom du sous-réseau (subnet):

Vous pouvez ajouter d'autres sous-réseaux (subnets) une fois qu'AWS a créé le VPC.

Points de terminaison de service

Activer les noms d'hôte DNS:* ☒ Oui ☐ Non

Location matérielle:*

Nous lui attribuons aussi un sous réseau public avec l'adresse 10.10.10.0/24 dans la zone us-east-1a.

Après cela nous nous attaquons au deuxième sous-réseau qui sera créée dans une deuxième zone différente :

Créer le sous-réseau

Spécifiez le bloc d'adresse IP de votre sous-réseau au format CIDR, par exemple, 10.0.0.0/24. Les tailles de bloc IPv4 doivent correspondre à votre VPC. Un bloc d'adresse CIDR IPv6 doit correspondre à un bloc d'adresse CIDR /64.

Balise Nom	<input type="text" value="Sous-reseau-exam2"/>					
VPC*	<input type="text" value="vpc-07ef9845d4a708269"/>					
CIDRS de VPC	<table> <thead> <tr> <th>CIDR</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td>10.10.0.0/16</td> <td>associated</td> </tr> </tbody> </table>	CIDR	Status	10.10.0.0/16	associated	
CIDR	Status					
10.10.0.0/16	associated					
Zone de disponibilité	<input type="text" value="us-east-1b"/>					
Bloc d'adresse CIDR IPv4*	<input type="text" value="10.10.11.0/24"/>					

* Obligatoire

Il sera donc lié à notre VPC Public « VPC-CLOUD-EXAM » et sera dotée de l'adresse 10.10.11.0/24 dans la seconde zone à savoir « us-east-1b ».

Avant de s'attaquer à la mise en place de l'équilibreur de charge nous allons créer la passerelle internet pour nos VPC et sous réseaux qui seront nécessaires :

[Passerelles Internet](#) > Créer une passerelle internet

Créer une passerelle internet

Une passerelle Internet est un routeur virtuel qui connecte un VPC à Internet. Pour créer une nouvelle passerelle Internet, spécifiez le nom de la passerelle ci-dessous.

Balise Nom

* Obligatoire

[Annuler](#) [Créer](#)

Nous l'associons ensuite au VPC « VPC-CLOUD-EXAM » :

[Passerelles Internet](#) > Associer au VPC

Associer au VPC

Associer une passerelle Internet à un VPC pour permettre la communication avec Internet. Spécifiez le VPC que vous souhaitez associer ci-dessous.

VPC* ⓘ

► Commande d'interface

ID de VPC	Nom
vpc-07ef9845d4a708269	VPC-CLOUD-EXAM

* Obligatoire

[Annuler](#) [Associer](#)

Création de la table de routage

Ensuite nous créons notre table de routage « ROUTAGE-EXAM » et toujours lié à notre VPC créé précédemment :

Créer une table de routage

Une table de routage spécifie le nombre de paquets transférés entre les sous-réseaux au sein de votre VPC, sur Internet et votre

Balise Nom ⓘ

VPC* ⓘ [C](#) ⓘ

* Obligatoire

Nous y venons à nos deux subnets qui sont visibles sur la liste, nous les associons donc et nous enregistrons :

Modifier des associations de sous-réseau

Table de routage **rtb-08e1dc68a16a48b29 (ROUTAGE-EXAM)**

Sous-réseaux associés **subnet-09a34c30790fc95be** **subnet-007f9cb9db20a2cad**

< > 1 à 2 sur 2

<input type="checkbox"/>	ID de sous-réseau	Bloc CIDR IPv	Le bloc d'adresse CIDR I	Table de routage actuelle
<input checked="" type="checkbox"/>	subnet-007f9cb9db20a2cad Sous-réseau-exam1	10.10.10.0/24	-	rtb-0810ef8654aa45998
<input checked="" type="checkbox"/>	subnet-09a34c30790fc95be Sous-reseau-exam2	10.10.11.0/24	-	Principal

• Obligatoire

[Annuler](#) [Enregistrer](#)

Nous autorisons les communications sur la cible de notre VPC afin de pouvoir communiquer sereinement et pouvoir dialoguer :

Modifier des routes

Destination	Cible	Statut	Propagée
10.10.0.0/16	local	active	Non
0.0.0.0/0	igw-041a3b500e4b63e4c		Non

[Ajouter une route](#)

• Obligatoire

[Annuler](#) [Enregistrer des routes](#)

Création des Instances :

Nous allons désormais passer à la partie de la création des deux Instances qui utiliserons nginx et que nous allons automatiser par la suite.

Étape 1 : Sélection d'une Amazon Machine Image (AMI)

Une AMI est un template qui contient la configuration logicielle (par ex., un système d'exploitation, un serveur d'applications et des applications) nécessaire pour lancer votre instance. Vous pouvez sélectionner une AMI fournie par AWS, notre communauté d'utilisateurs ou AWS Marketplace ; vous pouvez également sélectionner une de vos propres AMI.

Recherchez une AMI en entrant un terme de recherche, par exemple, « Windows »

Quick Start

Mes AMI

AWS Marketplace

AMI de la communauté

Offre gratuite uniquement

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-00068cd7555f543d5 (64 bits x86) / ami-035240afa793cddb (64 bits Arm)

Amazon Linux 2 est accompagné de cinq ans de support. Ce service fournit un noyau Linux 4.14 pour des performances optimales sur Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1 et les derniers packages logiciels via des extras.

Type de périphérique racine: ebs Type de virtualisation: hvm ENA activée: Oui

[Sélectionner](#)

64 bits (x86)
64 bits (Arm)

Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-00eb20669e0990cb4

L'AMI Amazon Linux est une image basée sur EBS et prise en charge par AWS. L'image par défaut comprend des outils de ligne de commande AWS, Python, Ruby, Perl et Java. Les référentiels incluent Docker, PHP, MySQL, PostgreSQL, ainsi que d'autres packages.

Type de périphérique racine: ebs Type de virtualisation: hvm ENA activée: Oui

[Sélectionner](#)

64 bits (x86)

Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-0c322300a1dd5dc79 (64 bits x86) / ami-03587fa4048e9eb92 (64 bits Arm)

Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type

Type de périphérique racine: ebs Type de virtualisation: hvm ENA activée: Oui

[Sélectionner](#)

64 bits (x86)
64 bits (Arm)

On va tout d'abord sélectionner la première AMI.

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: **Tous les types d'instances** **Génération actuelle** [Afficher / Masquer les colonnes](#)

Actuellement sélectionné: t2.micro (Variable ECU, 1 vCPU, 2.5 GHz, Intel Xeon Family, 1 Gio mémoire, EBS uniquement)

	Famille	Type	vCPU	Mémoire (Go)	Stockage d'instance (Go)	Disponible en version optimisée pour EBS	Performances réseau	Prise en charge IPv6
<input type="checkbox"/>	Usage général	t2.nano	1	0.5	EBS uniquement	-	Faibles à modérées	Oui
<input checked="" type="checkbox"/>	Usage général	t2.micro <small>Éligible à l'offre gratuite</small>	1	1	EBS uniquement	-	Faibles à modérées	Oui
<input type="checkbox"/>	Usage général	t2.small	1	2	EBS uniquement	-	Faibles à modérées	Oui
<input type="checkbox"/>	Usage général	t2.medium	2	4	EBS uniquement	-	Faibles à modérées	Oui
<input type="checkbox"/>	Usage général	t2.large	2	8	EBS uniquement	-	Faibles à modérées	Oui
<input type="checkbox"/>	Usage général	t2.xlarge	4	16	EBS uniquement	-	Modérées	Oui

[Annuler](#) [Précédent](#) [Vérifier et lancer](#) [Suivant : Configurer les détails de l'instance](#)

Nous laissons le choix par défaut qui est éligible à l'offre gratuite et donc suffisante dans notre cas puis nous sélectionnons « Suivant ».

Étape 3 : Configurer les détails de l'instance

Configurez l'instance en fonction de vos besoins. Vous pouvez lancer plusieurs instances à partir de la même AMI, demander des instances Spot pour bénéficier un tarif inférieur, attribuer un rôle de gestion d'accès à l'ins d'autres choses encore.

Nombre d'instances: [Lancer dans le groupe Auto Scaling](#)

Option d'achat: ☒ Demander des instances Spot

Réseau: [Créer un nouveau VPC](#)

Sous-réseau: [Créer un nouveau sous-réseau](#)
250 adresses IP disponibles

Attribuer automatiquement l'adresse IP publique:

Groupe de placement: ☒ Ajoutez une instance au groupe de placement.

Réserve de capacité: [Créer une nouvelle réserve de capacité](#)

Rôle IAM: [Créer un nouveau rôle IAM](#)

Comportement d'arrêt:

[Annuler](#) [Précédent](#) [Vérifier et lancer](#) [Suivant](#)

Dans la configuration des détails de l'instance il y a plusieurs choses que nous allons paramétrer, la première chose est le réseau où nous sélectionnerons notre VPC « VPC-CLOUD-EXAM ».

Puis pour automatiser le processus d'installation de nginx et récupérer nos données sources, nous passerons un script à la création de l'instance, ça nous permettra ainsi de gagner du temps de d'avoir une certaine automatisation présente :

▼ Détails avancés

Données utilisateur [?](#) ☒ Sous forme de texte ☐ Sous forme de fichier ☐ L'entrée est déjà codée en base64

```
#!/bin/bash
sudo yum update -y
sudo amazon-linux-extras install nginx1
sudo systemctl start nginx
sudo yum install unzip -y
cd /usr/share/nginx/html/
sudo rm -f index.html
sudo wget https://github.com/diranetafen/static-website-example/archive/master.zip
sudo unzip master.zip
cd static-website-example-master
sudo mv * /usr/share/nginx/html/
```

[3](#)

Détail :

Pour les commandes nous commençons par l'initialisation du langage que nous allons utiliser qui est donc le bash, ensuite nous mettons à jour l'instance, nous installons les paquets nginx et nous lançons le service installé, puis nous installons unzip qui est un utilitaire qui va nous permettre de dézipper l'archive que nous allons récupérer ensuite, une fois les installations terminées, nous nous plaçons dans le bon répertoire à savoir /usr/share/nginx/html, nous supprimons l'ancien index.html car il ne nous intéresse pas et nous déployerons le notre, et dans ce répertoire nous récupérerons le code source de notre projet nous le décompressons, et nous irons ensuite dans notre dossier static-website-example-master pour effectuer un moove de celui-ci (une sorte de couper/coller).

Une fois cela fait notre nginx est normalement fonctionnel et notre projet est déjà prêt à être accessible.

Ensuite le sous-réseau « exam1 » de la première zone us-east-1a.

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 6 : Configurer le groupe de sécurité

Un groupe de sécurité est un ensemble de règles de pare-feu qui contrôlent le trafic de votre instance. Sur cette page, vous pouvez ajouter des règles pour permettre qu'un trafic spécifique atteigne votre instance. Par exemple, si vous voulez configurer un serveur Web et permettre au trafic Internet d'atteindre votre instance, ajoutez des règles qui autorisent un accès restreint aux ports HTTP et HTTPS. Vous pouvez créer un nouveau groupe de sécurité ou en sélectionner un parmi les groupes existants ci-dessous. [En savoir plus](#) à propos des groupes de sécurité Amazon EC2.

Attribuer un groupe de sécurité : ☒ Créez un nouveau groupe de sécurité ☐ Sélectionnez un groupe de sécurité existant

Nom du groupe de sécurité :

Description :

Type	Protocole	Plage de ports	Source	Description
SSH	TCP	22	Personnali: 0.0.0.0/0	par exemple SSH for Admin Desktop

Ajouter une règle

Avertissement

Les règles avec une source de 0.0.0.0/0 permettent à toutes les adresses IP d'accéder à votre instance. Nous recommandons de paramétrer les règles du groupe de sécurité afin de permettre l'accès uniquement depuis des adresses IP connues.

➔ A noter que nous ajouterons aussi une règle de type http pour notre cas après la règle SSH

launch-wizard-18 created 2019-12-10T10:46:52.177+01:00

Type	Protocole	Plage de ports	Source	Description
SSH	TCP	22	Personnali: 0.0.0.0/0	par exemple SSH for Admin Desktop
HTTP	TCP	80	Personnali: 0.0.0.0/0, ::/0	par exemple SSH for Admin Desktop

Ajouter une règle

Après avoir passé les étapes précédentes, celle du groupe de sécurité arrive, nous en créons un nouveau qui ira avec notre nouvelle instance donc « launch-wizard-16 » et la règle SSH est bien mise en place.

Nous répétons cette opération deux fois pour nos deux instances, et nous obtiendrons ce résultat :

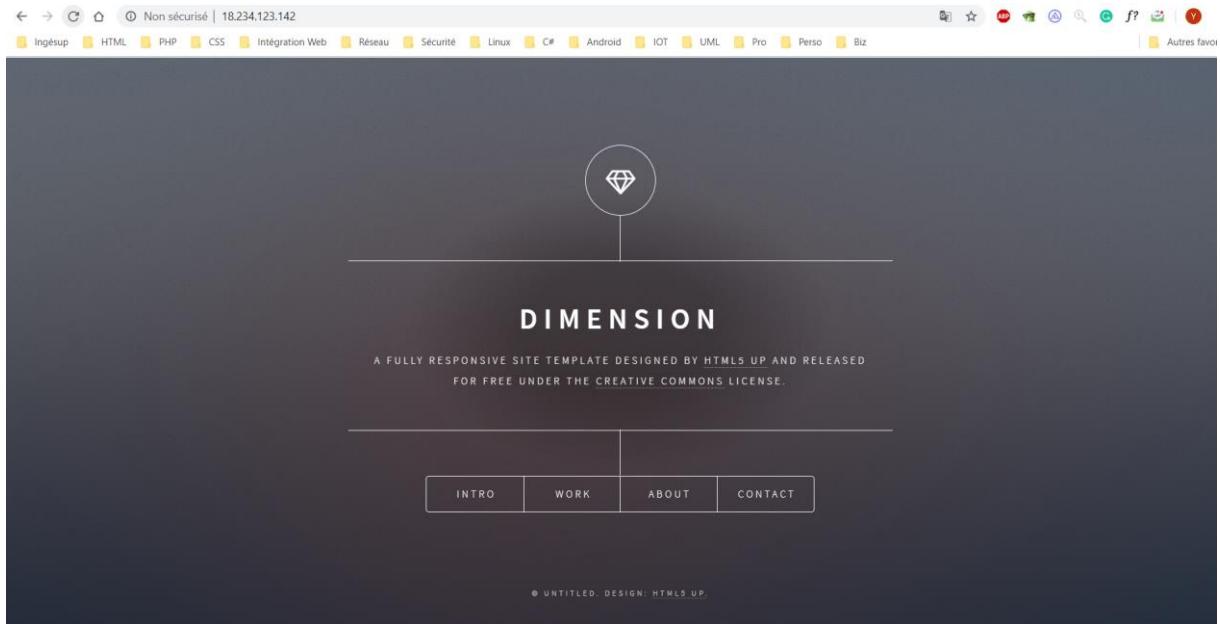
Lancer une instance Se connecter Actions

Filter par balises et attributs ou rechercher par mot clé

Name	ID d'instance	Type d'instance	Zone de disponibilité	État de l'instance	Contrôles des statuts	Statut des alarmes	DNS public (IPv4)	IP put
	i-0aa5b08ac0a6f3ef5	t2.micro	us-east-1b	running	2/2 contrôles réussis	Aucun(e)	ec2-18-234-123-142 co...	18.234...
	i-0fcd3fe0d7ed0fa3d	t2.micro	us-east-1a	running	2/2 contrôles réussis	Aucun(e)	ec2-3-95-37-197 comp...	3.95.3...

Pour tester son fonctionnement nous pouvons essayer d'accéder à la page uploadée en commande avec l'adresse IP public de notre instance à savoir 18.234.123.142 ou encore son DNS public ec2-18-234-123-142.compute-1.amazonaws.com

Lorsque nous accédons donc à cette adresse sur notre navigateur, nous pouvons voir le site web qui s’affiche bien correctement :



Création de l’ELB :

Pour rappel l’ELB alias Elastic Load Balancing est un outil qui va se charger de répartir automatiquement le trafic entrant d’application sur plusieurs cibles, comme des EC2, conteneurs, IP etc...

Nous utiliserons cet outil dans le cadre de la modernisation de l’infrastructure pour offrir des bonnes performances et garantir une gestion « basique » de la montée en charge.

Sélectionner un type d’équilibreur de charge

Elastic Load Balancing prend en charge trois types d’équilibreurs de charge : les équilibreurs de charge d’applications, ceux du réseau (nouveau) et les classiques. Choisissez le type d’équilibreur de charge correspondant à vos besoins. [En savoir plus sur l’équilibreur de charge adapté à vos besoins](#)

Équilibreur de charge d'application	Équilibreur de charge réseau	Équilibreur de charge classique
Créer	Créer	Créer
<p>Choisissez un équilibreur de charge d'application quand vous avez besoin d'un ensemble de fonctions flexible pour vos applications web avec un trafic HTTP et HTTPS. En opérant au niveau des demandes, les équilibreurs de charge d'application fournissent des fonctions avancées de routage et de visibilité ciblant les architectures d'application, y compris les microservices et les conteneurs.</p> <p>En savoir plus ></p>	<p>Choisissez un équilibreur de charge réseau quand vous avez besoin de performances très élevées, de déchargement TLS à grande échelle, d'un déploiement de certificat centralisé, de la prise en charge d'UDP et d'adresses IP statiques pour votre application. En opérant au niveau de la connexion, les équilibreurs de charge réseau sont capables de traiter des millions de demandes par seconde en toute sécurité tout en assurant des latences ultra-faibles.</p> <p>En savoir plus ></p>	<p>Choisissez un équilibreur de charge classique quand vous disposez d'une application existante en cours d'exécution dans le réseau EC2-Classic.</p> <p>En savoir plus ></p>

On va donc mettre en place l’équilibreur de charge d’application, surligné ici ci-dessus sur cette capture d’écran.

Étape 1: Configurer l'équilibreur de charge

HTTP	80
<button>Ajouter un écouteur</button>	

Zones de disponibilité

Spécifiez les zones de disponibilité à activer pour votre équilibreur de charge. L'équilibreur de charge achemine le trafic jusqu'aux cibles de disponibilité. Vous devez spécifier les sous-réseaux d'au moins deux zones de disponibilité afin d'accroître la disponibilité de l'équilibreur de charge.

VPC	vpc-07ef9845d4a708269 (10.10.0.0/16) VPC-CLOUD-EXAM
Zones de disponibilité	<input checked="" type="checkbox"/> us-east-1a subnet-007f9cb9db20a2cad (Sous-réseau-exam1)
	Adresse IPv4 Attribuées par AWS
	<input checked="" type="checkbox"/> us-east-1b subnet-09a34c30790fc95be (Sous-reseau-exam2)
	Adresse IPv4 Attribuées par AWS

Nous configurons celui-ci de manière à assurer une certaine disponibilité entre nos deux sous réseaux, c'est la tout l'intérêt de la création de notre ELB, nous aurons une assurance de service sur chaque région différente à savoir « us-east-1a » et « us-east-1b ».

Étape 3: Configurer les groupes de sécurité

Un groupe de sécurité est un ensemble de règles de pare-feu qui contrôlent le trafic vers l'équilibreur de charge. Sur cette page, vous pouvez ajouter des règles pour permettre qu'un trafic spécifique atteigne l'équilibreur de charge. Vous devez d'abord décider si vous préférez créer un groupe de sécurité ou en sélectionner un qui existe déjà.

Attribuer un groupe de sécurité:

- ☐ Créez un nouveau groupe de sécurité
- ☒ Sélectionnez un groupe de sécurité existant

ID de groupe de sécurité	Nom	Description
sg-0ee4e63df851dc5e9	default	default VPC security group

Nous utilisons le groupe de sécurité par défaut déjà existant.

Instances

Pour enregistrer des instances supplémentaires, sélectionnez une ou plusieurs instances en cours d'exécution, spécifiez un port, puis cliquez sur Ajouter. Le port par défaut est le port spécifié pour le groupe cible. Si l'instance a déjà été enregistrée sur le port spécifié, vous devez indiquer un port différent.

<button>Ajouter au membre</button>	sur le port 80					
<input type="text" value="Rechercher des instances"/>						
Instance	Nom	État	Groupes de sécurité	Zone	ID de sous-réseau (subnet)	CIDR du sous-réseau (subnet)
<input checked="" type="checkbox"/>	i-0fcd3fe0d7ed0fa3d	running	default	us-east-1a	subnet-007f9cb9db20a2cad	10.10.10.0/24
<input checked="" type="checkbox"/>	i-0aa5b08ac0af63ef5	running	default	us-east-1b	subnet-09a34c30790fc95be	10.10.11.0/24

Puis on enregistre nos instances en tant que cible pour les associer à notre ELB.

État de création de l'équilibreur de charge

✓ Équilibreur de charge créé avec succès

L'équilibreur de charge [ELB-EXAM](#) a été créé avec succès

. Remarque : il est possible que vous deviez attendre quelques minutes avant que l'équilibreur de charge soit totalement opérationnel et prêt à acheminer le trafic, q
vérifications initiales de la santé soient réussies pour l'enregistrement.

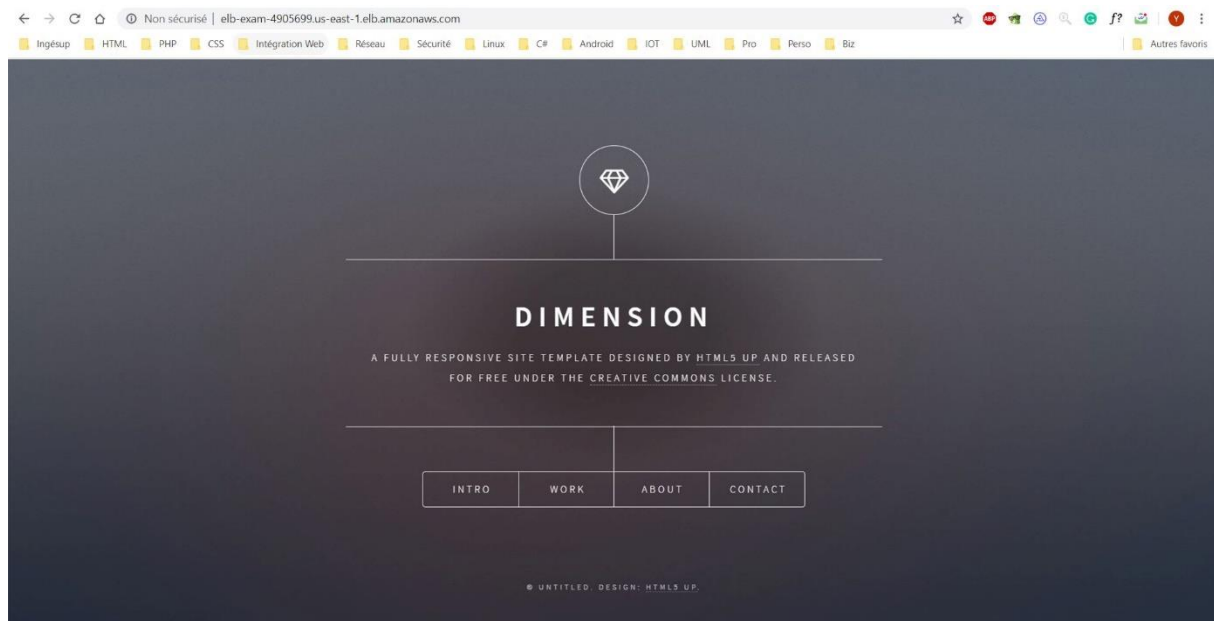
Étapes suivantes suggérées

- Découvrez d'autres services que vous pouvez intégrer à votre équilibreur de charge. Consultez l'onglet Integrated services (Services intégrés) dans [ELB-EXAM](#)
- Envisagez d'utiliser AWS Global Accelerator afin d'améliorer la disponibilité de votre application. [Console AWS Global Accelerator](#)

Ensuite un message nous indique que notre ELB au nom de « ELB-EXAM » viens d'être créé avec succès.

Et si nous essayons d'accéder à l'interface depuis notre ELB créé grâce à son URL : <http://elb-exam-4905699.us-east-1.elb.amazonaws.com/>

Nous voyons donc que notre ELB est bien fonctionnel :



L'architecture pour POZOS est donc prête et fonctionnelle.

