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 devrons minimiser les coûts pour l'entreprise.

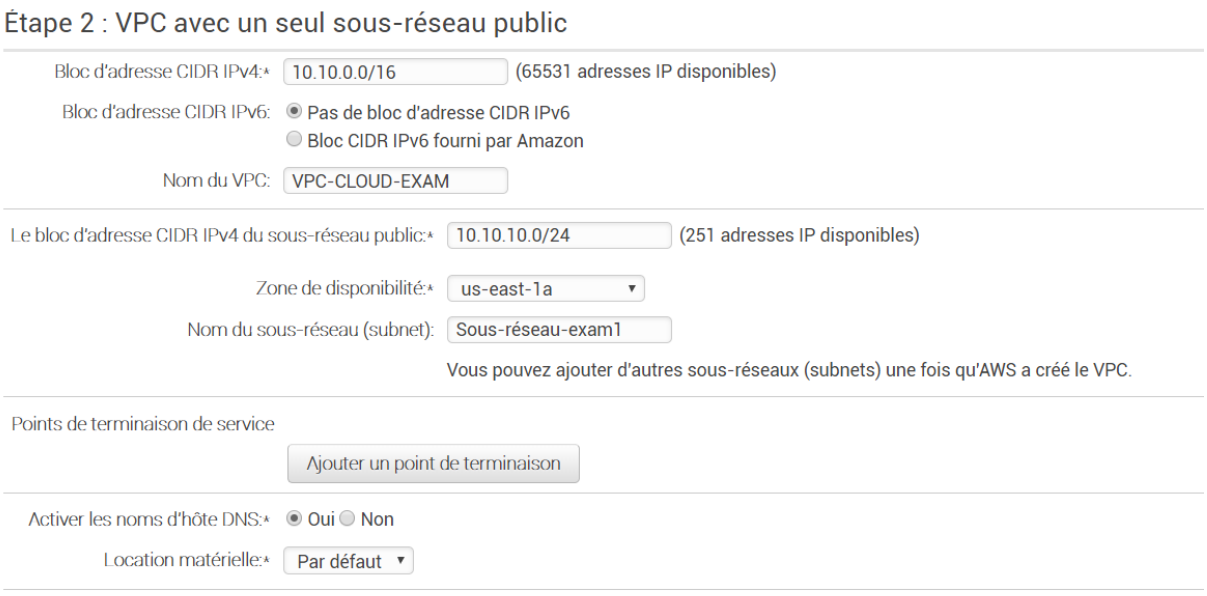
Pour répondre ainsi à ces demandes là nous utiliserons les solutions Cloud fournies par Amazon, à savoir AWS.

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

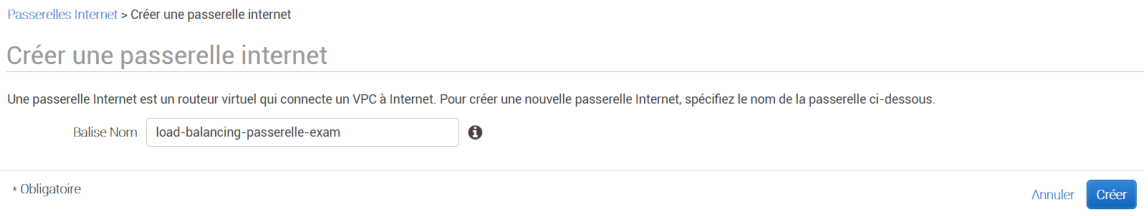


Nous lui attribuons aussi un sous réseau public avec l'adresse 10.10.10.0/24 dans la zone us-east-1a.

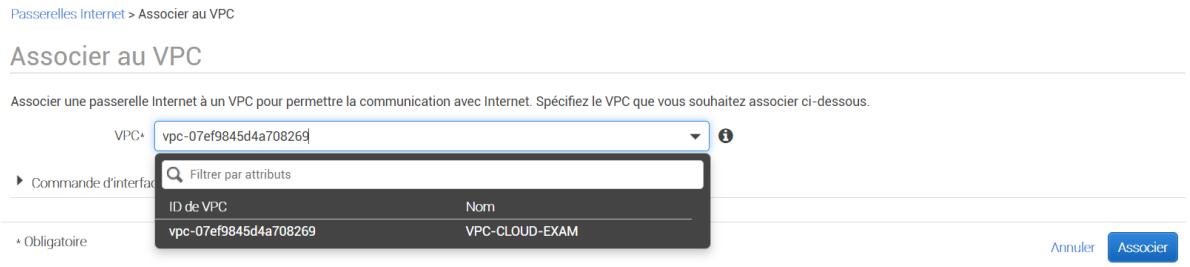
Après cela nous nous attaquons au deuxième sous-réseau qui sera créée dans une deuxième zone différente :

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 :

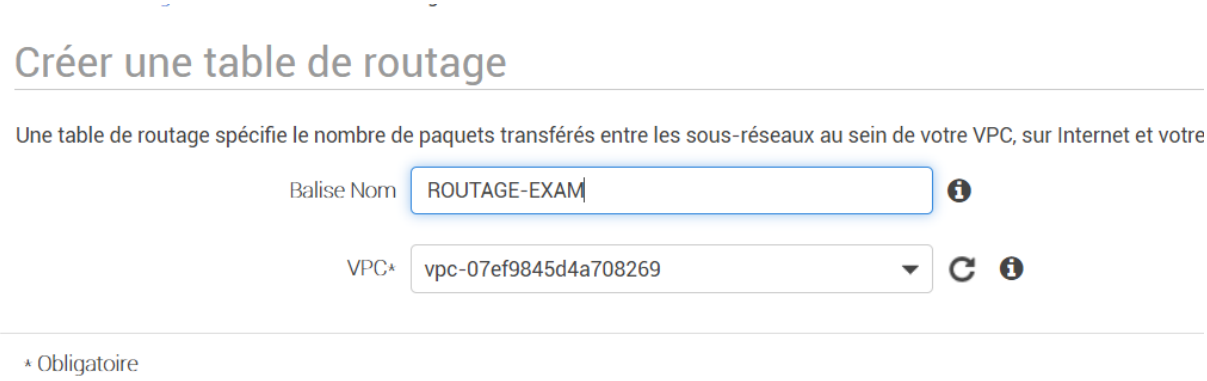


Nous l'associons ensuite au VPC « VPC-CLOUD-EXAM » :

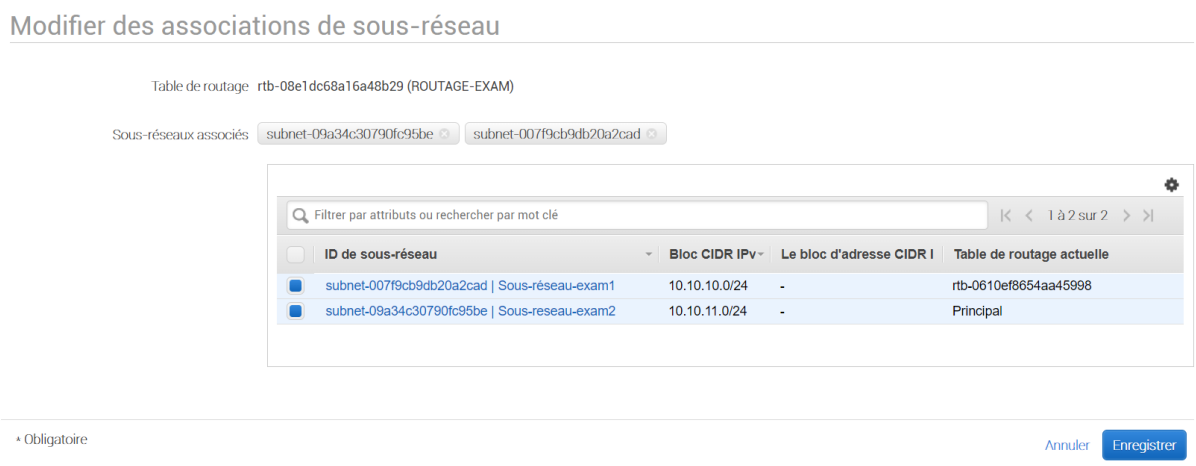


Création de la table de routage

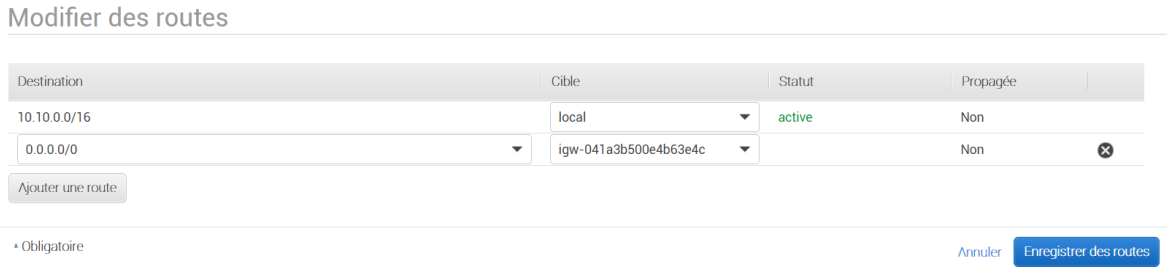
Ensuite nous créons notre table de routage « ROUTAGE-EXAM » et toujours lié à notre VPC créé précédemment :



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

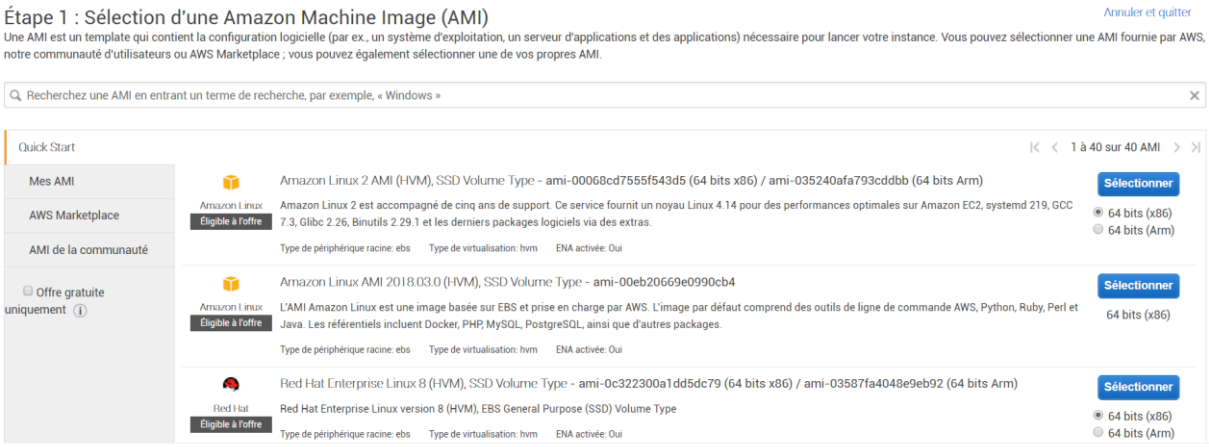


Nous autorisons les communications sur la cible de notre VPC afin de pouvoir communiquer sereinement et pouvoir dialoguer :

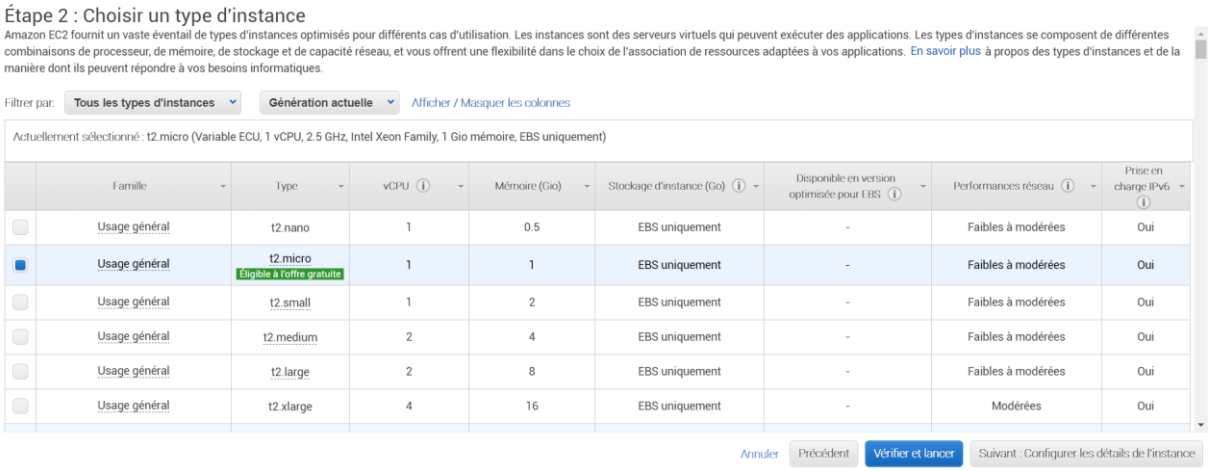


Création des Instances :

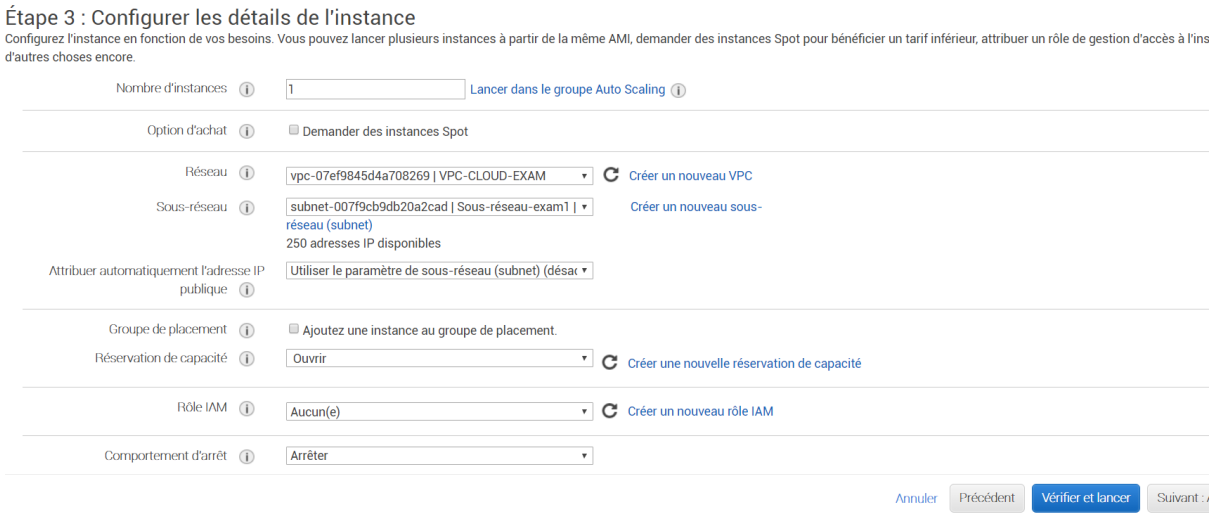
Nous allons désormais passer à la partie de la création des deux Instances qui utiliserons nginx et que nous allons automatiser par la suite.



On va tout d'abord sélectionner la première AMI.

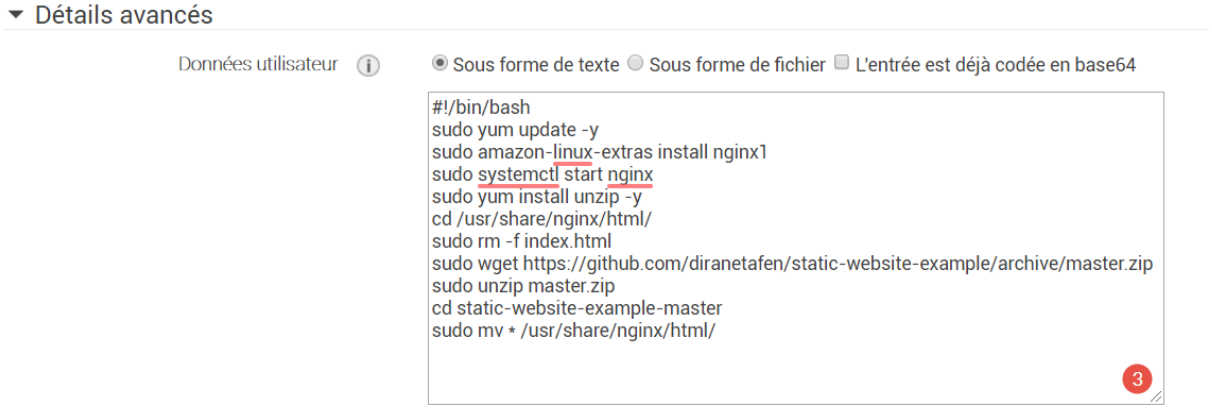


Nous laissons le choix par défaut qui est éligible à l'offre gratuite et donc suffisante dans notre cas puis nous sélectionnons « Suivant ».



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 :

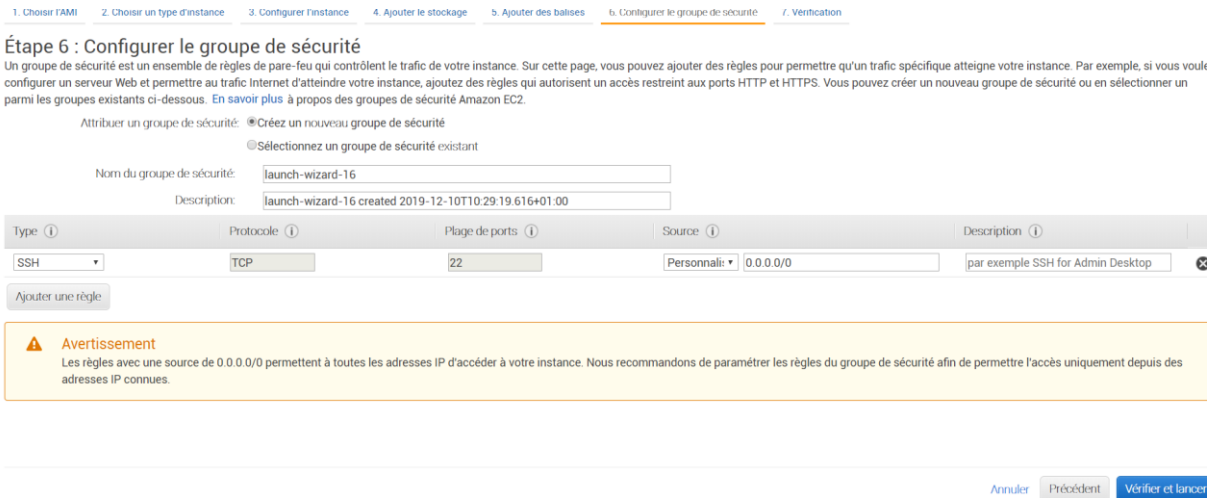


Détail :

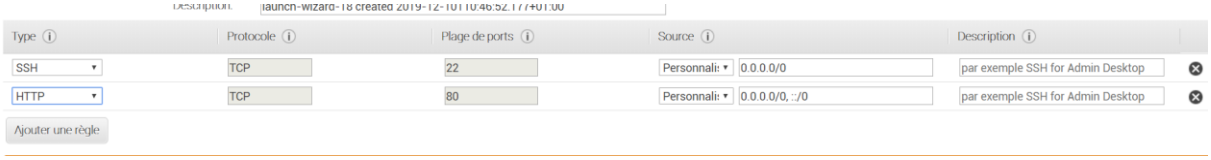
Pour les commandes nous commençons par l'initialisation du langage que nous allons utiliser qui est donc le bash, ensuite nous mettons à jour l'instance, nous installons les paquets nginx et nous lançons le service installé, puis nous installons unzip qui est un utilitaire qui va nous permettre de dézipper 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 déploierons le notre, et dans ce répertoire nous récupérons 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 fois 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.



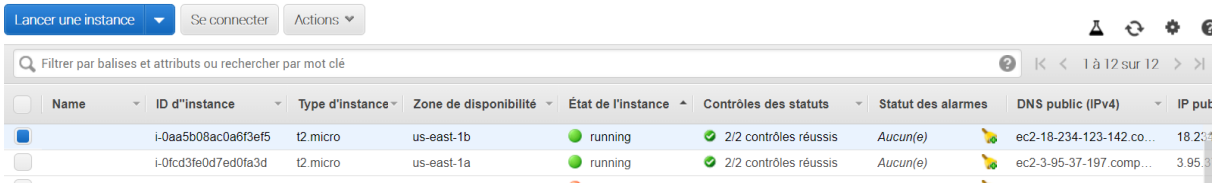
➔ A noter que nous ajouterons aussi une règle de type http pour notre cas après la règle SSH



➔

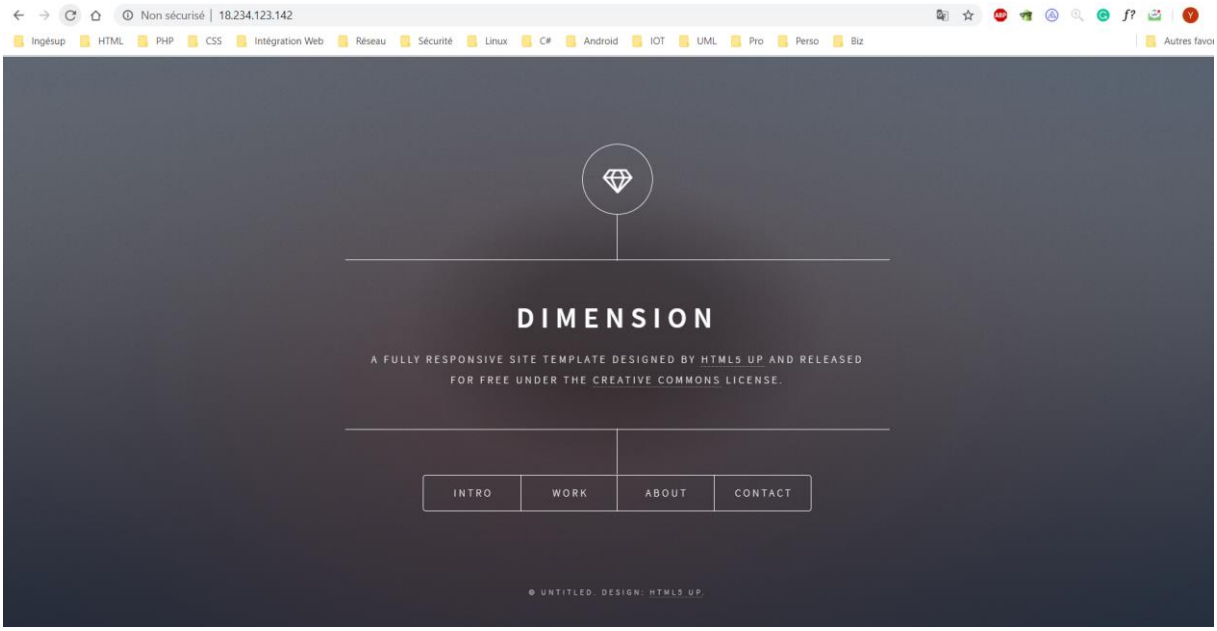
Après avoir passé les étapes précédentes, celle du groupe de sécurité arrive, nous en créons un nouveau qui ira avec notre nouvelle instance donc « launch-wizard-16 » et la règle SSH est bien mise en place.

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



Pour tester son fonctionnement nous pouvons essayer d'accéder a 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 :

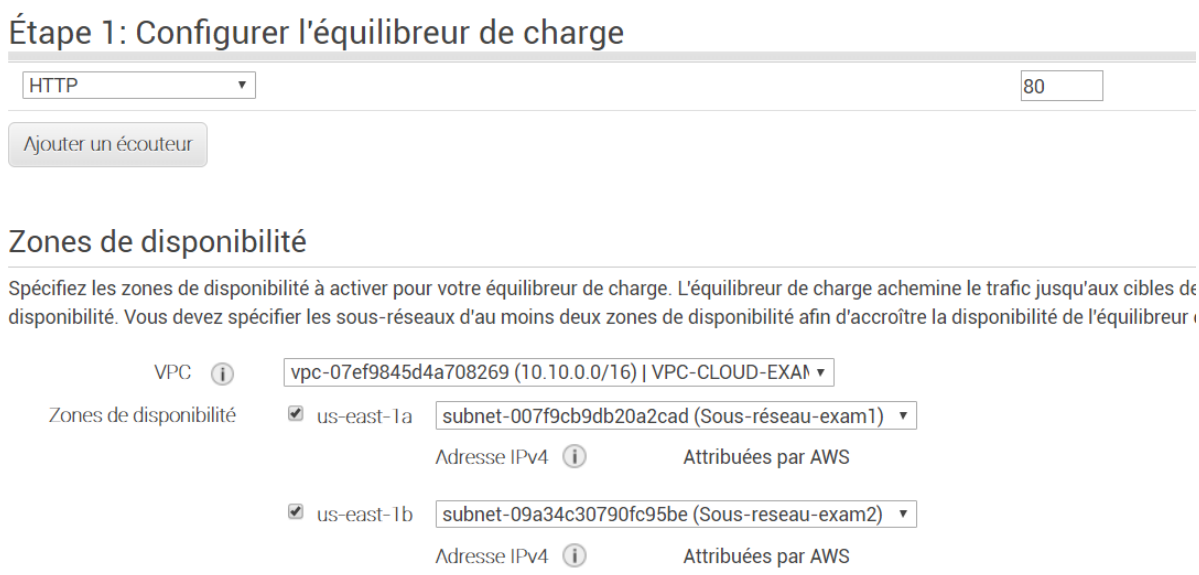


Création de l'ELB :

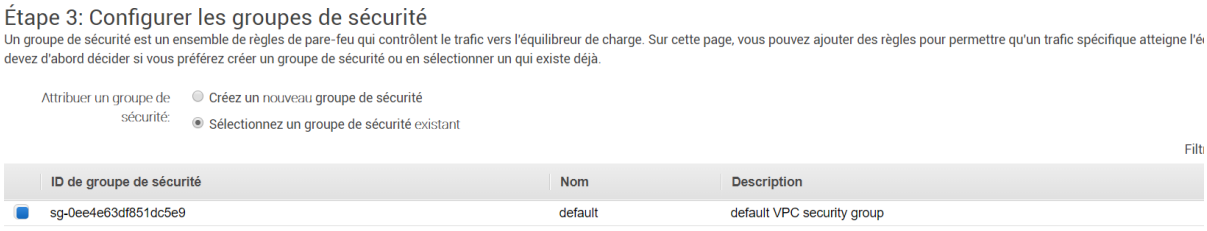
Pour rappel l'ELB alias Elastic Load Balancing est un outil qui va se charger de répartir automatiquement le trafic entrant d'application sur plusieurs cibles, comme des EC2, conteneurs, IP etc…

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.

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

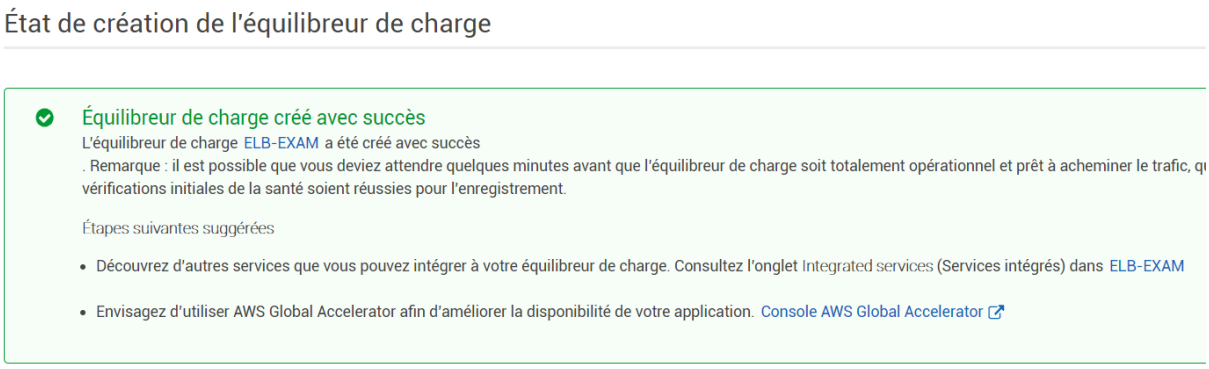


Nous configurons celui-ci de manière a assurer une certaine disponibilité entre nos deux sous réseaux, c'est la tout l'intérêt de la création de notre ELB, nous aurons une assurance de service sur chaque région différente à savoir « us-east-1a » et « us-east-1b ».



Nous utilisons le groupe de sécurité par défaut déjà existant.

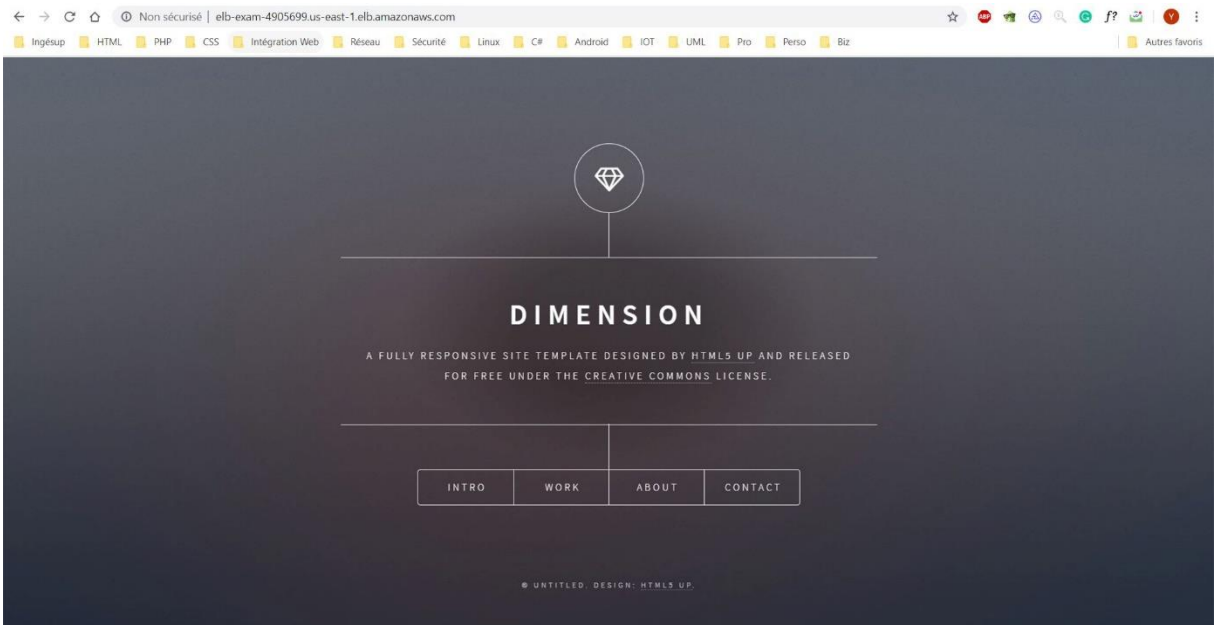
Puis on enregistre nos instances en tant que cible pour les associer a notre ELB.



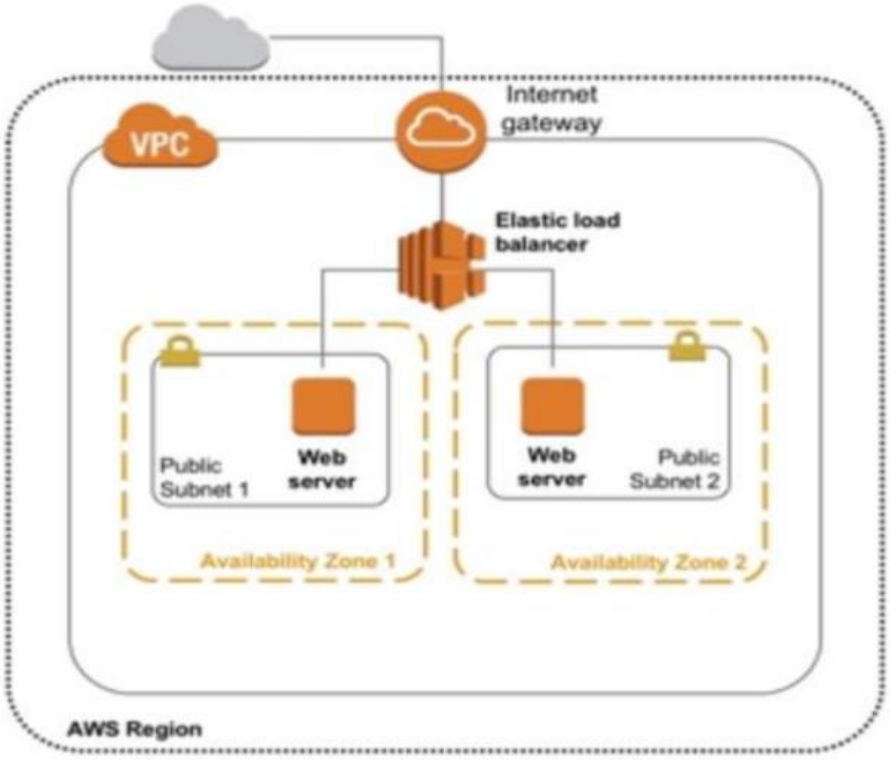
Ensuite un message nous indique que notre ELB au nom de « ELB-EXAM » viens d'être créé avec succès.

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.



POZOS IT