

## **TP N°2**

### **SUJET :**

**INFRASTRUCTURE AS A SERVICE**

**UE 3.6  
INFRASTRUCTURE SYSTEMES ET RESEAUX**

**BRICE AUGUSTIN**

**Durée : 4 heures**

## Sommaire

1. Introduction.....	3
2. Configuration du VPC.....	4
3. Utilisation de la CLI .....	6
4. Infrastructure as code .....	7
5. Mémos .....	8
5.1. Divers.....	8
5.2. AWS .....	10
5.3. Packer .....	13

## 1. Introduction

Vous êtes chargés d'externaliser une partie de l'infrastructure IT d'une petite entreprise :

- Son site Web
- Un serveur Git
- Un serveur de fichiers

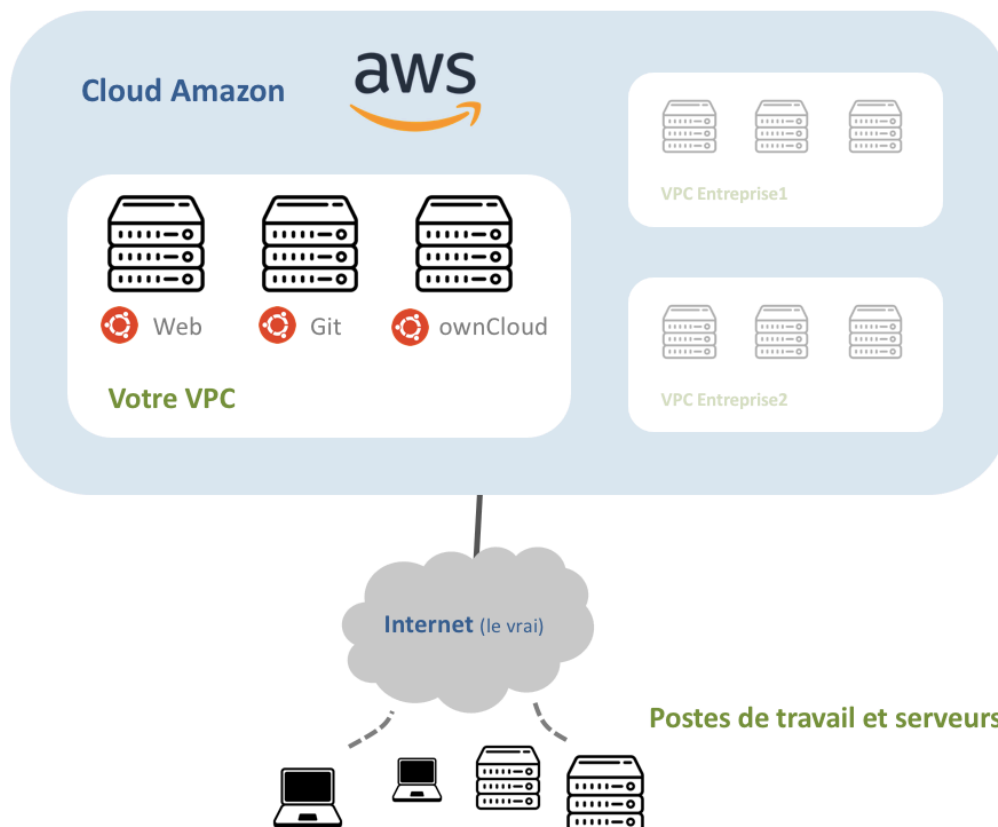


Fig. 1 Maquette

Vous décidez d'utiliser la plate-forme AWS (*Amazon Web services*) et en particulier son offre IaaS nommée EC2 (*elastic compute cloud*).

## 2. Configuration du VPC

Votre VPC doit respecter le cahier des charges suivant :

- Tous les serveurs tournent sous `Ubuntu 18.04 LTS`
- Le serveur Web héberge le site Web de l'entreprise. Il contient une seule page Web (*avec le contenu de votre choix*)
- Le serveur Web est accessible via son adresse IP publique (*il ne dispose pas de nom de domaine*)
- Chaque employé dispose d'un dépôt sur le serveur Git
- Le serveur de fichiers est hébergé par `ownCloud`

**Préparation** : Avant de commencer à configurer votre maquette, répondez aux questions suivantes :

1. Dans une connexion SSH, quel est le rôle du fichier `~/.ssh/id_rsa` ?
2. Qu'est-ce que le NAT statique ?
3. Quel est le port TCP par défaut du protocole SSH ?

### Conseils :

Une instance de type `t2.micro` sera suffisante pour chaque serveur.

Sur EC2, les instances `Ubuntu` sont accessibles en SSH avec le compte `ubuntu`. Connectez-vous en utilisant l'[authentification par clé publique](#)\*. Votre clé publique est automatiquement copiée sur l'instance lors de sa création.

Les instances EC2 se trouvent derrière un routeur NAT. Chaque instance possède :

- Une adresse IP privée, dans le plan d'adressage de votre VPC
- Une adresse IP publique, routable sur internet

Les instances EC2 sont protégées par un pare-feu à état (il s'agit d'une abstraction nommée `security groups`). Par défaut, aucune communication depuis internet n'est autorisée.

Pour [configurer ownCloud](#)\*, utilisez l'excellent tutorial de Digital Ocean.

Pour [configurer le dépôt Git](#)\*, utilisez un tutoriel sur internet.

Attention à la sécurité lorsque vous créez des comptes sur vos instances. Leurs adresses IP étant publiques, elles sont accessibles par tous ... *Choisissez des mots de passe forts !*

**Synthèse 1 :** Expliquez en 4-6 lignes les grandes étapes de réalisation de votre maquette.

Appelez votre chargé de TP et montrez-lui votre VPC.

### 3. Utilisation de la CLI

Sur un de vos postes de travail, [installez la CLI AWS\\*](#).

[Configurez la connexion\\*](#) avec votre compte AWS.

Prenez quelques minutes pour vous familiariser avec la [manipulation des instances EC2 via la CLI\\*](#) (création, liste, destruction), puis utilisez-la pour créer un second serveur Web identique au premier.

#### Conseils :

Les instances sont créées à partir d'images de base nommées AMI. Chaque AMI est identifiée par un identifiant unique (`ami-*`). A vous de déterminer celui de l'AMI Ubuntu 18.04 LTS.

Pour créer une instance, vous devez obligatoirement fournir les renseignements suivants :

- Identifiant de l'AMI
- Nombre d'instance à créer
- Type d'instance (`t2.micro`, etc.)
- La paire de clés à utiliser

Vous devez également [configurer le firewall\\*](#) AWS avec la CLI.

**Synthèse 2** : Expliquez en 4-6 lignes les grandes étapes de création de votre script.

Appelez votre chargé de TP et montrez-lui que vous pouvez créer une instance EC2 'sans les mains'.

## 4. Infrastructure as code

Automatisez la création d'une AMI personnalisée. Celle-ci doit avoir les mêmes caractéristiques que le serveur Web précédent.

### Conseils :

Packer possède un *builder* pour Amazon EC2\*. Inspirez-vous du fichier de configuration fourni par votre prof. Attention à la *gestion des identifiants\** d'accès à votre compte AWS.

**Synthèse 3** : Expliquez en 4-6 lignes les grandes étapes de création de votre fichier de configuration Packer.

Appelez votre chargé de TP et montrez-lui que vous pouvez créer une AMI personnalisée 'sans les mains'.

## 5. Mémos

### 5.1. Divers

Se connecter en SSH avec une authentification par clé publique :

Le client utilise la clé privée présente dans le fichier `~/.ssh/id_rsa`. Pour utiliser une clé différente :

```
# Le fichier mykey contient la clé privée à utiliser
ssh -i mykey otabenga@203.0.0.113.2
# On peut également écraser ~/.ssh/id_rsa ...
```

Configurer ownCloud sur Ubuntu 18.04 LTS :



Utiliser l'excellent tutorial de Digital Ocean :

<https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-owncloud-on-ubuntu-18-04>

**Préparation** : Voici la liste de *toutes* les étapes suggérées par le tutorial précédent.

Certains prérequis du tutorial sont déjà satisfaits. D'autres ne sont pas nécessaires pour réaliser une maquette de base. Par exemple, il n'est pas nécessaire de configurer le firewall, de renforcer la sécurité avec une suite cryptographique forte, ni de forcer la redirection du trafic non chiffré vers HTTPS.

Lisez le tutorial en diagonale, puis faites appel à votre expérience et votre bon sens pour déterminer les étapes qui sont *strictement indispensables* :

1. Créer un utilisateur avec les privilèges sudo
2. Activer le firewall
3. Installer Apache
4. Autoriser le trafic Web dans le firewall
5. Installer un serveur de base de données MySQL
6. Sécuriser le serveur MySQL
7. Installer PHP
8. Redémarrer le service Apache
9. Installer des modules optionnels pour PHP
10. Vérifier que PHP fonctionne bien sur le serveur
11. Créer un certificat SSL associé à un nom de domaine
12. Créer un certificat SSL sans nom de domaine
13. Renforcer la sécurité d'Apache avec une suite cryptographique forte
14. Modifier le VirtualHost pour activer SSL
15. Rediriger le trafic non chiffré vers HTTPS
16. Autoriser le trafic HTTPS dans le firewall
17. Activer les changements dans Apache

18. Vérifier que le chiffrement est bien activé
19. Forcer la redirection du trafic non chiffré vers HTTPS
20. Installer ownCloud
21. Faire pointer le DocumentRoot sur le dossier d'ownCloud
22. Configurer la base de données MySQL
23. Configurer ownCloud

Faites valider votre analyse par votre prof.

*Si vous êtes en retard, utilisez le script d'installation fourni par votre prof !*

Configurer un dépôt `Git` :

Le tutorial suivant est clair. Il décrit également la méthode pour partager un dépôt entre plusieurs utilisateurs (*pas nécessaire ici*) :

<http://www.ganuq.com/2017/10/10/creer-serveur-git/>

## 5.2. AWS

Installer la CLI AWS :

Installer le paquetage `python`, puis la bibliothèque `awscli` pour Python :

```
pip install awscli
```

Récupérer ses identifiants de connexion :

Sur la page principale du compte AWS Educate, cliquer sur `Account Details > (AWS CLI) Show`

### Your Classroom Account Status




	<b>Active</b> full access ()
	<b>\$48.86</b> remaining credits (estimated)
	<b>2:60</b> session time
<a href="#">Account Details</a> <a href="#">AWS Console</a>	

Fig. 2 Statut du compte AWS Educate

Configurer la connexion à son compte AWS :

Récupérer ses **identifiants de connexion\*** et les copier dans le fichier `~/.aws/credentials`.

Il faut aussi indiquer le *datacenter* Amazon où se situe votre VPC. Ajouter dans le fichier `~/.aws/config` :

```
# Datacenter de Virginie
region=us-east-1
```

Manipuler des instances EC2 avec la CLI :

Consulter la documentation d'Amazon :

[https://docs.aws.amazon.com/fr\\_fr/cli/latest/userguide/cli-services-ec2-instances.html](https://docs.aws.amazon.com/fr_fr/cli/latest/userguide/cli-services-ec2-instances.html)

Commandes `aws ec2` usuelles :

- `run-instances`
- `terminate-instances`
- `describe-instances`

Ajouter une règle entrante dans le firewall AWS :

```
# Renvoie l'identifiant du security group (sg-*)
# associé à l'instance (dont l'identifiant
# est $INSTANCE_ID)
aws ec2 describe-instance-attribute \
    --instance-id $INSTANCE_ID \
    --attribute groupSet

# Ajoute une règle dans le security group $SG_ID
# pour autoriser les connexions TCP sur le port 4242
# de l'instance, depuis n'importe où.
aws ec2 authorize-security-group-ingress \
    --group-id $SG_ID \
    --protocol tcp --port 4242 --cidr 0.0.0.0/0
```

### 5.3. Packer

Syntaxe du *builder* pour Amazon EC2 :

<https://www.packer.io/docs/builders/amazon-ec2.html>

Gestion des identifiants AWS :

Plusieurs méthodes au choix :

- Dans le fichier `~/.aws/credentials` (**recommandé**)
- En dur dans le fichier de configuration
- Variables d'environnement

<https://www.packer.io/docs/builders/amazon.html#specifying-amazon-credentials>