

Rapport labo 1 CLD

CLD Lab 01 : Déploiement d'une application web sur Infrastructure as a Service

Auteurs : Amir Mouti et Harun Ouweis

Groupe : GrP

Date : 01.03.2024

Tâche 1 : Configuration initiale

Création d'une paire de clés publique/privée

Pour nous connecter à la machine virtuelle nouvellement créée, nous avons généré une paire de clés publique/privée nommée `GrP_Mouti/Harun`. Cette clé sera utilisée pour accéder de manière sécurisée à nos instances EC2 via SSH.

Création d'un groupe de sécurité par défaut

Nous avons créé un groupe de sécurité par défaut nommé `GrP_Mouti/Harun_default` pour notre serveur web. Ce groupe de sécurité est configuré pour autoriser le trafic SSH depuis n'importe quelle adresse (0.0.0.0/0) sur le port 22. Ce paramètre est destiné uniquement à l'utilisation en laboratoire et n'est pas recommandé dans un environnement de production.

Tâche 2 : Création d'une instance Amazon EC2

Délivrables

1. Le type d'instance le plus petit et le plus grand disponible :

La plus petite : 1 vCPU 0,5 GiB de mémoire La plus grande : 448 vCPU 24 TiB

2. Temps nécessaire pour que l'instance soit en état de fonctionnement : Uniquement quelques secondes

3. Nom DNS public de l'instance : ec2-52-90-155-30.compute-1.amazonaws.com

Exploration de l'instance

Après nous être connectés à l'instance via SSH, nous avons exécuté plusieurs commandes pour explorer l'instance.

Réponses aux questions :

- **Différence de temps entre la Suisse et l'heure réglée sur la machine** : Utilisé `date`, comparé à l'heure en Suisse. La machine utilise UTC. alors qu'en Suisse nous utilisons UTC+1.
- **Nom de l'hyperviseur** : Utilisation de `lscpu` a révélé que l'instance fonctionne sur un hyperviseur Xen.
- **Espace libre sur le disque** : Avec `df -h`, regardé l'espace disponible sur `/dev/xvda15` ainsi que `/dev/root`. on a trouvé 99 Mo ainsi que 5.7GB disponible donc 5.8Go disponible au total:

```
ubuntu@ip-172-31-90-86:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        7.6G  2.0G  5.7G  26% /
tmpfs            475M    0  475M   0% /dev/shm
tmpfs            190M  852K  190M   1% /run
tmpfs            5.0M    0   5.0M   0% /run/lock
/dev/xvda15      105M   6.1M   99M   6% /boot/efi
tmpfs            95M   4.0K   95M   1% /run/user/1000
```

- **Tentative de ping de l'instance depuis votre machine locale** : avant ajustement, nous avons un Time-Out car l'adresse n'est pas accepter . Après ajustement des règles du groupe de sécurité pour autoriser ICMP, utilisé `ping` 54.211.14.222. étant mon adresse IPV4, Noté des temps aller-retour moyens de 450 ms.

```
ubuntu@ip-172-31-90-86:~$ ping 54.211.14.222
PING 54.211.14.222 (54.211.14.222) 56(84) bytes of data.
64 bytes from 54.211.14.222: icmp_seq=1 ttl=63 time=0.446 ms
64 bytes from 54.211.14.222: icmp_seq=2 ttl=63 time=0.419 ms
64 bytes from 54.211.14.222: icmp_seq=3 ttl=63 time=0.495 ms
64 bytes from 54.211.14.222: icmp_seq=4 ttl=63 time=0.455 ms
64 bytes from 54.211.14.222: icmp_seq=5 ttl=63 time=0.437 ms
64 bytes from 54.211.14.222: icmp_seq=6 ttl=63 time=0.458 ms
64 bytes from 54.211.14.222: icmp_seq=7 ttl=63 time=0.468 ms
64 bytes from 54.211.14.222: icmp_seq=8 ttl=63 time=0.453 ms
64 bytes from 54.211.14.222: icmp_seq=9 ttl=63 time=0.514 ms
64 bytes from 54.211.14.222: icmp_seq=10 ttl=63 time=0.489 ms
64 bytes from 54.211.14.222: icmp_seq=11 ttl=63 time=0.484 ms
```

- **Adresse IP vue par le système d'exploitation dans l'instance EC2** : L'adresse IP identifiée par la commande `ifconfig` sur l'instance EC2, 172.31.90.86, est une adresse IP privée, typique des réseaux internes et non routable directement sur Internet. En contraste, l'adresse 54.211.14.222, vue lors de l'utilisation de la commande ping, est une adresse IP publique, accessible depuis n'importe quel point d'accès sur Internet.

Cette configuration illustre l'utilisation du Network Address Translation (NAT) par AWS. Le NAT permet à l'adresse IP publique de l'instance EC2, accessible depuis l'extérieur, d'être mappée à son adresse IP privée interne. Ce mécanisme est crucial pour deux raisons principales :

1. **Sécurité** : Il masque l'adresse IP privée de l'instance, réduisant ainsi la surface d'attaque potentielle depuis l'extérieur du réseau AWS.

2. **Connectivité** : Il permet à l'instance de communiquer avec Internet et d'être accessible depuis l'extérieur, malgré l'utilisation d'une adresse IP privée au sein du réseau AWS.

Ainsi, même si l'instance opère en interne avec une adresse IP privée, la traduction d'adresse effectuée par le NAT permet aux utilisateurs externes de se connecter à l'instance en utilisant son adresse IP publique. Cette approche offre un équilibre entre la nécessité de connectivité externe et les impératifs de sécurité interne.

Tâche 3 : Installation d'une application web

Installation de la pile LAMP

Nous avons commencé par mettre à jour les paquets et installer la pile LAMP avec les commandes suivantes :

```
sudo apt update
sudo apt install lamp-server^
```

Configuration du pare-feu

Nous avons modifié la configuration du pare-feu pour autoriser les connexions HTTP entrantes en ajoutant une règle pour les connexions HTTP (port 80) depuis n'importe où.

Vérification de l'accès au serveur Apache

En saisissant le nom DNS public de l'instance dans un navigateur web, nous avons confirmé l'accès au serveur Apache, affichant la page de bienvenue par défaut.

Téléchargement et configuration de Wordpress

Nous avons téléchargé la dernière version de Wordpress, extrait les fichiers, et configuré la base de données MySQL pour Wordpress avec les commandes suivantes :

```
wget https://wordpress.org/latest.tar.gz
tar -xzf latest.tar.gz
sudo systemctl start mysql
sudo mysql -e "CREATE USER 'wordpress-user'@'localhost' IDENTIFIED BY 'wordpress-
password';"
sudo mysql -e "CREATE DATABASE wordpress-db;"
sudo mysql -e "GRANT ALL PRIVILEGES ON wordpress-db.* TO 'wordpress-
user'@'localhost';"
sudo mysql -e "FLUSH PRIVILEGES;"
```

Nous avons ensuite configuré Wordpress en ajustant les paramètres de connexion à la base de données dans le fichier `wp-config.php` et copié les fichiers dans le répertoire Apache pour le contenu du site.

Activation des permaliens Wordpress et redémarrage d'Apache

Nous avons modifié le fichier de configuration Apache pour permettre les permaliens Wordpress et redémarré le serveur Apache.

Installation de Wordpress

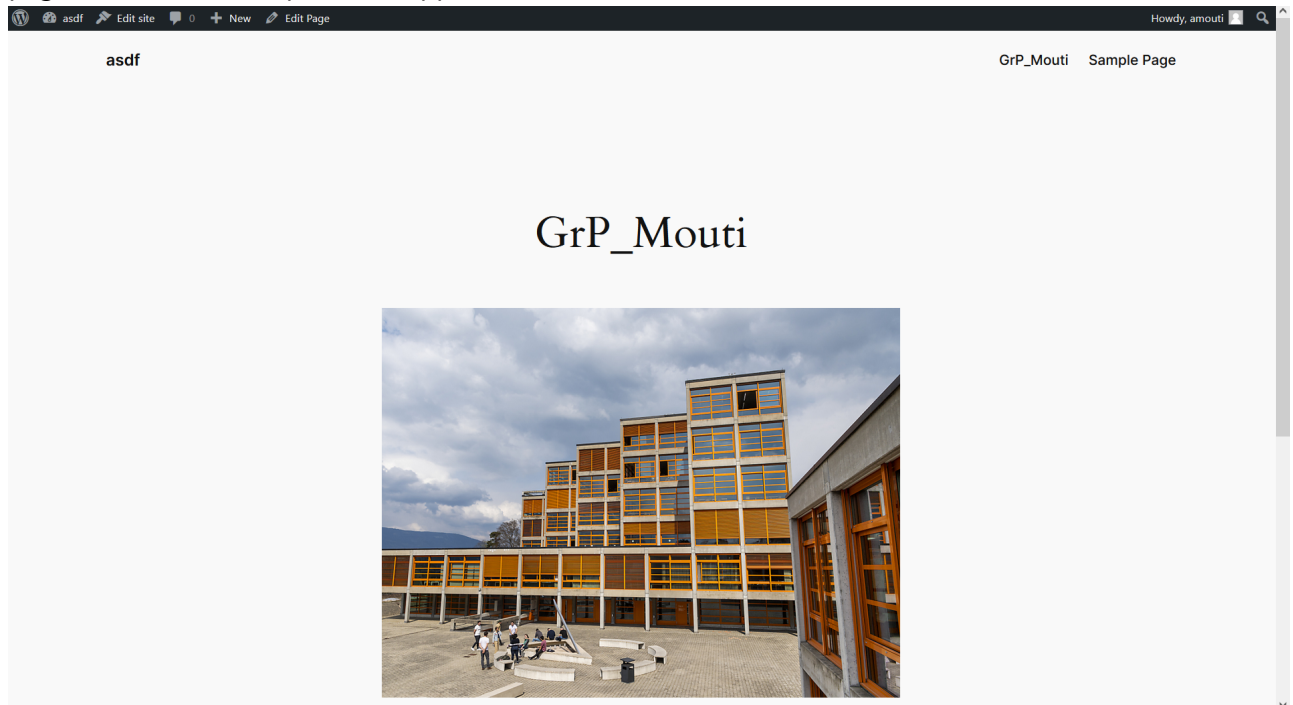
En naviguant vers la page d'administration de Wordpress via le navigateur, nous avons complété l'installation en configurant les paramètres du site.

Création et association d'une adresse IP élastique

Pour assurer la pérennité de l'accès au site, nous avons créé et associé une adresse IP élastique à l'instance, puis ajusté l'adresse IP dans la configuration de Wordpress en utilisant `wp-cli`.

Délivrables

- **Capture d'écran de la page créée dans Wordpress** : Nous avons ajouté une capture d'écran de la page créée dans Wordpress au rapport.



- **Adresse IP élastique créée** : Nous avons documenté l'adresse IP élastique attribuée à notre instance dans le rapport. IP élastique : `3.222.134.214`

Justification de l'utilisation d'une adresse IP élastique

L'utilisation d'une adresse IP élastique pour un site web (notre application web) est avantageuse car elle permet de maintenir une adresse IP fixe pour l'instance, facilitant la gestion DNS et l'accès constant au site. Contrairement au nom DNS public de l'instance qui peut changer lors de redémarrages, une adresse IP élastique reste associée à l'instance jusqu'à ce qu'elle soit explicitement dissociée ou remappée, offrant ainsi une stabilité et une fiabilité accrues pour les utilisateurs finaux.

Tâche 4 : Benchmark

- Fournissez les URL des résultats Geekbench pour l'instance EC2 et votre machine locale.

EC2 : <https://browser.geekbench.com/v3/cpu/9072780>
Locale : <https://browser.geekbench.com/v3/cpu/9072785>

- Fournissez des informations système sur l'instance EC2.

System Information	
Xen HVM domU	
Operating System	Ubuntu 22.04.3 LTS 6.2.0-1017-aws x86_64
Model	Xen HVM domU
Processor	Intel Xeon E5-2686 v4 @ 2.30 GHz 1 Processor, 1 Core
Processor ID	GenuineIntel Family 6 Model 79 Stepping 1
L1 Instruction Cache	32 KB
L1 Data Cache	32 KB
L2 Cache	256 KB
L3 Cache	46080 KB
Motherboard	N/A
BIOS	Xen 4.11.amazon
Memory	949 MB

- Fournissez les scores de performance pour un seul cœur et pour plusieurs cœurs pour les performances globales, entières, en virgule flottante et de mémoire de l'instance EC2.

Geekbench 3 Score

2447

Single-Core Score

2455

Multi-Core Score

Geekbench 3.3.0 Tryout for Linux x86 (32-bit)

Integer Performance

Single-core	2748
Multi-core	2731
AES Single-core	3829 3.28 GB/sec
AES Multi-core	3861 3.30 GB/sec
Twofish Single-core	2584 145.0 MB/sec
Twofish Multi-core	2596 145.7 MB/sec
SHA1 Single-core	3103 336.8 MB/sec
SHA1 Multi-core	3127 339.4 MB/sec
SHA2 Single-core	3149 136.2 MB/sec
SHA2 Multi-core	3186 137.9 MB/sec
BZip2 Compress Single-core	2213 9.00 MB/sec
BZip2 Compress Multi-core	2195 8.93 MB/sec
BZip2 Decompress	2576

Single-core	14.0 MB/sec
BZip2 Decompress	2573
Multi-core	13.9 MB/sec
JPEG Compress	2360
Single-core	32.9 Mpixels/sec
JPEG Compress	2360
Multi-core	32.9 Mpixels/sec
JPEG Decompress	2769
Single-core	68.5 Mpixels/sec
JPEG Decompress	2706
Multi-core	66.9 Mpixels/sec
PNG Compress	2282
Single-core	1.82 Mpixels/sec
PNG Compress	2280
Multi-core	1.82 Mpixels/sec
PNG Decompress	2484
Single-core	28.7 Mpixels/sec
PNG Decompress	2468
Multi-core	28.5 Mpixels/sec
Sobel	2784
Single-core	101.3 Mpixels/sec
Sobel	2789
Multi-core	101.5 Mpixels/sec
Lua	3484
Single-core	3.13 MB/sec
Lua	3409
Multi-core	3.06 MB/sec
Dijkstra	2583
Single-core	9.27 Mpairs/sec
Dijkstra	2447
Multi-core	8.78 Mpairs/sec

Floating Point Performance	
Single-core	2559
Multi-core	2455
BlackScholes Single-core	2295 10.2 Mnodes/sec
BlackScholes Multi-core	2330 10.4 Mnodes/sec
Mandelbrot Single-core	2734 2.80 Gflops
Mandelbrot Multi-core	2732 2.80 Gflops
Sharpen Filter Single-core	2521 1.87 Gflops
Sharpen Filter Multi-core	2544 1.89 Gflops
Blur Filter Single-core	2381 2.27 Gflops
Blur Filter Multi-core	2382 2.27 Gflops
SGEMM Single-core	2453 6.87 Gflops
SGEMM Multi-core	2438 6.83 Gflops
DGEMM Single-core	2377 3.49 Gflops
DGEMM Multi-core	2343 3.44 Gflops

SFFT Single-core	2649 2.79 Gflops
SFFT Multi-core	2643 2.79 Gflops
DFFT Single-core	2064 1.88 Gflops
DFFT Multi-core	1191 1.09 Gflops
N-Body Single-core	3550 1.32 Mpairs/sec
N-Body Multi-core	3991 1.48 Mpairs/sec
Ray Trace Single-core	2827 3.33 Mpixels/sec
Ray Trace Multi-core	2875 3.39 Mpixels/sec

Memory Performance	
Single-core	1622
Multi-core	1904
Stream Copy Single-core	1076 4.30 GB/sec
Stream Copy Multi-core	3456 13.8 GB/sec
Stream Scale Single-core	1921 7.67 GB/sec
Stream Scale Multi-core	1621 6.47 GB/sec
Stream Add Single-core	1758 7.95 GB/sec
Stream Add Multi-core	1654 7.48 GB/sec
Stream Triad Single-core	1909 8.39 GB/sec
Stream Triad Multi-core	1421 6.25 GB/sec

- Fournissez des informations système sur votre machine locale.

System Information	
LENOVO 21BSS09F00	
Operating System	Microsoft Windows 10 Professionnel (64-bit)
Model	LENOVO 21BSS09F00
Processor	Intel Core i7-1260P @ 2.10 GHz 1 Processor, 12 Cores, 16 Threads
Processor ID	GenuineIntel Family 6 Model 154 Stepping 3
Processor Codename	
Processor Package	
L1 Instruction Cache	32 KB x 8
L1 Data Cache	48 KB x 8
L2 Cache	1280 KB x 2
L3 Cache	18432 KB
Motherboard	LENOVO 21BSS09F00
Northbridge	Intel ID4621 02
Southbridge	Intel ID5182 01
BIOS	LENOVO N3CET59W (1.40)
Memory	15.71 GB -1MHz

- Fournissez les scores de performance pour un seul cœur et pour plusieurs cœurs pour les performances globales, entières, en virgule flottante et de mémoire de votre machine locale.

Geekbench 3 Score

Valid

4824

Single-Core Score

23931

Multi-Core Score

Geekbench 3.4.4 Tryout for Windows x86 (32-bit)

Integer Performance	
Single-core	5134
Multi-core	32568
AES Single-core	8932 7.64 GB/sec
AES Multi-core	29994 25.7 GB/sec
Twofish Single-core	4764 267.4 MB/sec
Twofish Multi-core	36370 1.99 GB/sec
SHA1 Single-core	19655 2.08 GB/sec
SHA1 Multi-core	162518 17.2 GB/sec
SHA2 Single-core	7488 324.0 MB/sec
SHA2 Multi-core	39222 1.66 GB/sec
BZip2 Compress Single-core	3901 15.9 MB/sec
BZip2 Compress Multi-core	29919 121.6 MB/sec
BZip2 Decompress	3700

Single-core	20.1 MB/sec
BZip2 Decompress	26973
Multi-core	146.2 MB/sec
JPEG Compress	3704
Single-core	51.6 Mpixels/sec
JPEG Compress	25199
Multi-core	351.1 Mpixels/sec
JPEG Decompress	5744
Single-core	142.0 Mpixels/sec
JPEG Decompress	37111
Multi-core	917.4 Mpixels/sec
PNG Compress	3184
Single-core	2.54 Mpixels/sec
PNG Compress	19494
Multi-core	15.6 Mpixels/sec
PNG Decompress	3589
Single-core	41.4 Mpixels/sec
PNG Decompress	30241
Multi-core	348.7 Mpixels/sec
Sobel	5103
Single-core	185.7 Mpixels/sec
Sobel	28547
Multi-core	1.04 Gpixels/sec
Lua	4708
Single-core	4.23 MB/sec
Lua	27333
Multi-core	24.6 MB/sec
Dijkstra	3265
Single-core	11.7 Mpairs/sec
Dijkstra	19218
Multi-core	69.0 Mpairs/sec

Floating Point Performance	
Single-core	4546
Multi-core	24193
BlackScholes Single-core	4725 21.0 Mnodes/sec
BlackScholes Multi-core	43077 191.7 Mnodes/sec
Mandelbrot Single-core	3846 3.94 Gflops
Mandelbrot Multi-core	28730 29.4 Gflops
Sharpen Filter Single-core	3091 2.29 Gflops
Sharpen Filter Multi-core	16790 12.4 Gflops
Blur Filter Single-core	3676 3.50 Gflops
Blur Filter Multi-core	18217 17.4 Gflops
SGEMM Single-core	5224 14.6 Gflops
SGEMM Multi-core	23597 66.1 Gflops
DGEMM Single-core	6146 9.03 Gflops
DGEMM Multi-core	17786 26.1 Gflops

SFFT Single-core	5023 5.30 Gflops
SFFT Multi-core	26107 27.5 Gflops
DFFT Single-core	4106 3.74 Gflops
DFFT Multi-core	19050 17.3 Gflops
N-Body Single-core	6549 2.43 Mpairs/sec
N-Body Multi-core	28407 10.5 Mpairs/sec
Ray Trace Single-core	4219 4.98 Mpixels/sec
Ray Trace Multi-core	30616 36.1 Mpixels/sec

Memory Performance	
Single-core	4761
Multi-core	6134
Stream Copy Single-core	4705 18.8 GB/sec
Stream Copy Multi-core	6437 25.7 GB/sec
Stream Scale Single-core	4928 19.7 GB/sec
Stream Scale Multi-core	6198 24.7 GB/sec
Stream Add Single-core	4689 21.2 GB/sec
Stream Add Multi-core	6070 27.5 GB/sec
Stream Triad Single-core	4726 20.8 GB/sec
Stream Triad Multi-core	5849 25.7 GB/sec

- Comparez les scores globaux des deux machines.

La performance globale pour un seul cœur est meilleure sur la machine locale que sur l'instance EC2, ce qui est logique puisque c'est une instance bon marché. La plus grande différence est avec la performance multi-cœurs, où la performance de l'EC2 reste inchangée car elle n'a qu'un seul cœur, tandis que la machine locale a un score beaucoup plus élevé en raison de la présence de plusieurs cœurs.

Tâche 5 : Consommation de ressources et tarification

Délivrables recalculés :

1. Coût de l'instance par heure et du disque par heure :

- Coût de l'instance `t2.micro` par heure : 0,0116 \$/heure
- Coût du volume SSD EBS par heure pour 8 Go : 0,08 \$ par Go-mois. Pour 8 Go : $(0,08 \$ * 8 \text{ Go}) / (30 \text{ jours} * 24 \text{ heures}) = 0,0008889 \$/\text{heure}$

2. Coût pour ce laboratoire (estimation approximative) pour une seule personne :

Si le laboratoire dure 6 heures :

- Coût total de l'instance pour le laboratoire : $0,0116 \$ * 6 = 0,0696 \$$
- Coût total du disque pour le laboratoire : $0,0008889 \$ * 6 = 0,0053334 \$$
- Coût total estimé pour le laboratoire : $0,0696 \$ + 0,0053334 \$ = 0,0749334 \$$

3. Calcul du coût total de la configuration utilisée dans le laboratoire pour un mois de fonctionnement continu en production :

- Coût mensuel de l'instance **t2.micro** : $0,0116 \$ * 24 \text{ heures} * 30 \text{ jours} = 8,352 \$$
- Coût mensuel du volume SSD EBS de 8 Go : $0,08 \$ * 8 \text{ Go} = 0,64 \$$

Transfert de données : 100'000 visiteurs téléchargeant chacun 85 MB.

- Coût du transfert de données : 0,09 \$ par Go pour les premiers 10 TB / mois : $100'000 * 85 \text{ MB} = 8\,500 \text{ GB} = 8,5 \text{ TB}$. Coût : $8,5 \text{ TB} * 0,09 \$ = 765 \$$

Adresse IPv4 publique :

- **Coût mensuel** : 0,005 \$ par heure, tant pour une adresse active qu'une adresse inactive.
- Calcul pour une adresse active pour un mois : $0,005 \$ * 24 \text{ heures} * 30 \text{ jours} = 3,60 \$$

Coût total pour un mois :

- Coût de l'instance + Coût du disque + Coût du transfert de données + Coût de l'adresse IPv4 = $8,352 \$ + 0,64 \$ + 765 \$ + 3,60 \$ = 777,592 \$$

Détails de votre calcul :

Cette révision intègre les coûts pour une adresse IPv4 publique, facturés à 0,005 \$ par heure, que l'adresse soit utilisée ou non. Le coût total pour l'utilisation continue des ressources cloud pendant un mois inclut désormais le coût de l'instance, le stockage SSD EBS, le transfert de données, et l'adresse IPv4 publique, menant à une estimation ajustée de 780 \$ pour un mois de fonctionnement.

Tâche 6

Nous avons interrompu les instances mais laissé disponible pour une utilisation antérieure.