

Partie 1 : Création d'un compte sur Amazon AWS

Nous avions déjà fait cette partie, lors de la dernière séance, mais sans avoir fait des captures d'écran.

Partie 2 : Service EC2

Création d'une machine virtuelle

1. Après la connexion on a l'image suivante :

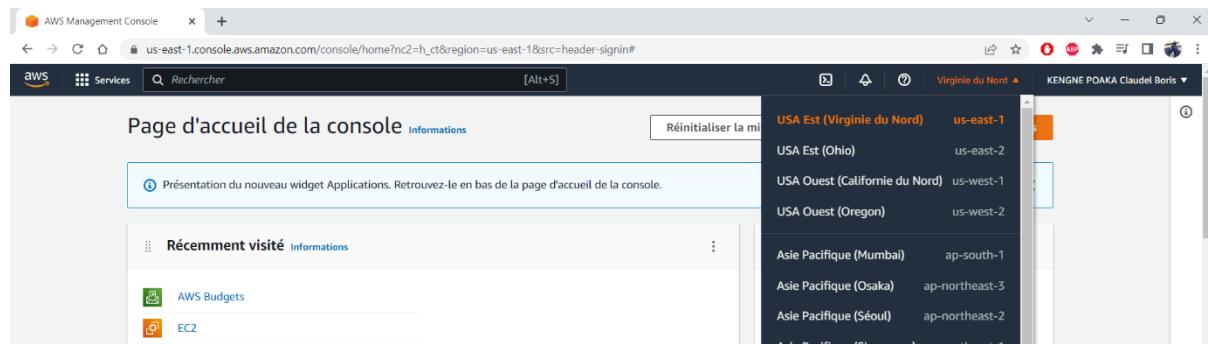


Figure 1 : Page d'accueil de la console dans mon compte AWS

2. Le service EC2

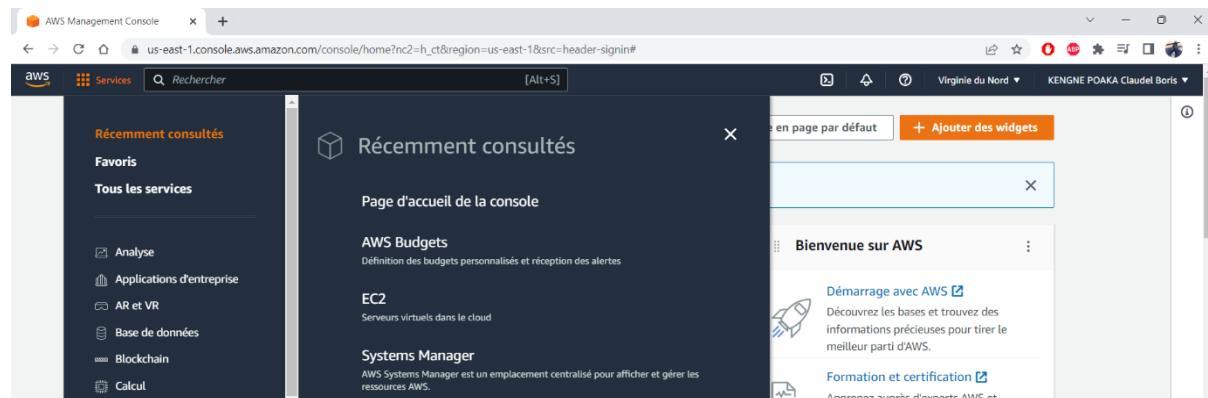


Figure 2 : Choix du service EC2

3. Configuration de la nouvelle machine virtuelle

Résumé

Nombre d'instances [Informations](#)
1

Image logicielle (AMI)
Canonical, Ubuntu, 22.04 LTS, ...en savoir plus
ami-08c40ec9ead489470

Type de serveur virtuel (type d'instance)
t2.micro

Pare-feu (groupe de sécurité)
Nouveau groupe de sécurité

Stockage (volumes)
1 volume(s) - 10 Gio

Offre gratuite : La première année inclut 750 heures d'utilisation mensuelle des instances t2.micro (ou t3.micro dans les régions où t2.micro n'est pas disponible) sur les AMI de

[Annuler](#) [Lancer l'instance](#)

Figure 3 : Etape 1

Démarrage rapide

Amazon Linux, macOS, Ubuntu, Windows, Red Hat, S, [Parcourir d'autres AMI](#)

Y compris des AMI provenant d'AWS, de Marketplace et de la communauté

Résumé

Nombre d'instances [Informations](#)
1

Image logicielle (AMI)
Canonical, Ubuntu, 22.04 LTS, ...en savoir plus
ami-08c40ec9ead489470

Type de serveur virtuel (type d'instance)
t2.micro

Pare-feu (groupe de sécurité)
Nouveau groupe de sécurité

Stockage (volumes)
1 volume(s) - 10 Gio

Offre gratuite : La première année inclut 750 heures d'utilisation mensuelle des instances t2.micro (ou t3.micro dans les régions où t2.micro n'est pas disponible) sur les AMI de

[Annuler](#) [Lancer l'instance](#)

Figure 4 : Etape 2

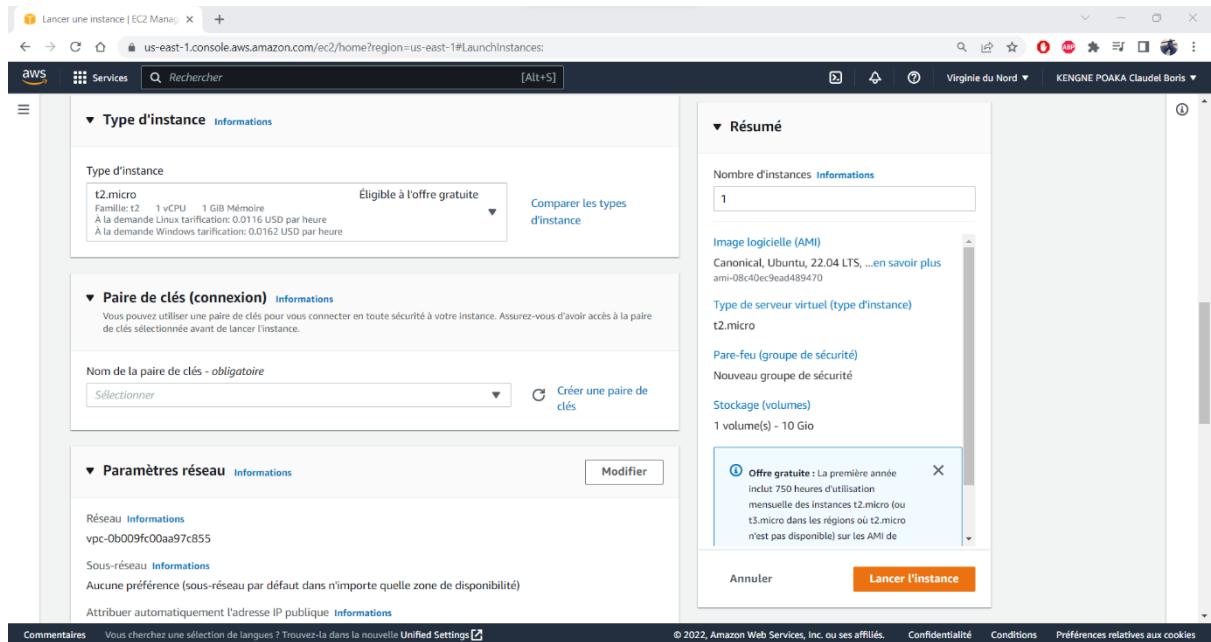


Figure 5 : Etape 3 Nous avons créé une paire de clés de type .pem

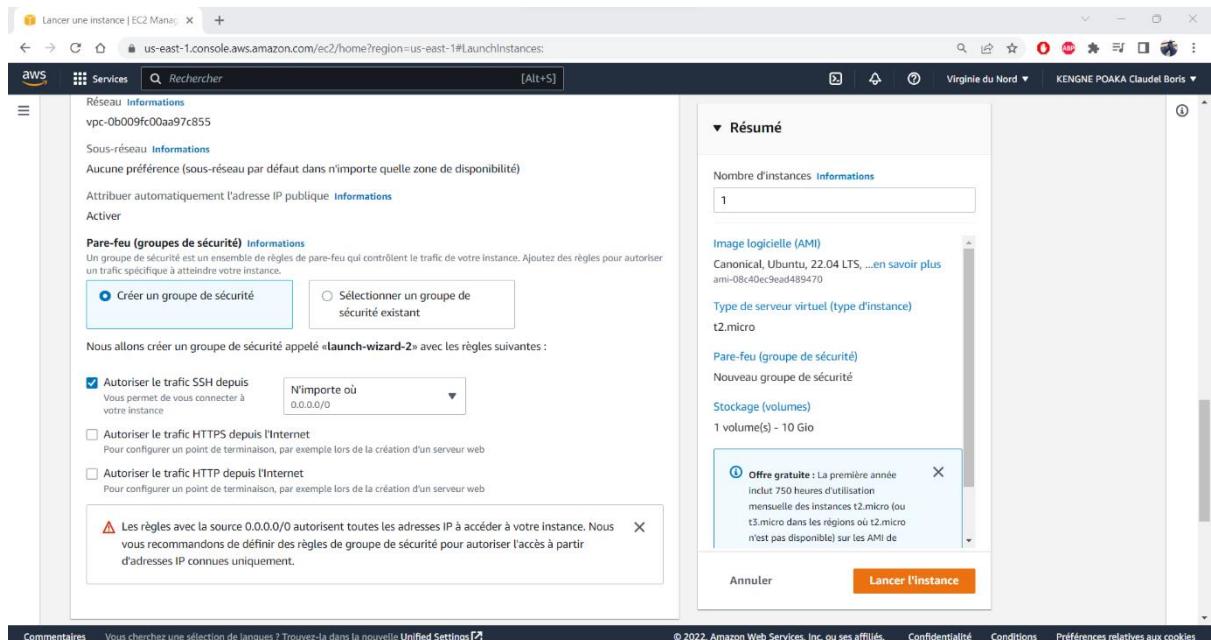


Figure 6 : Etape 4

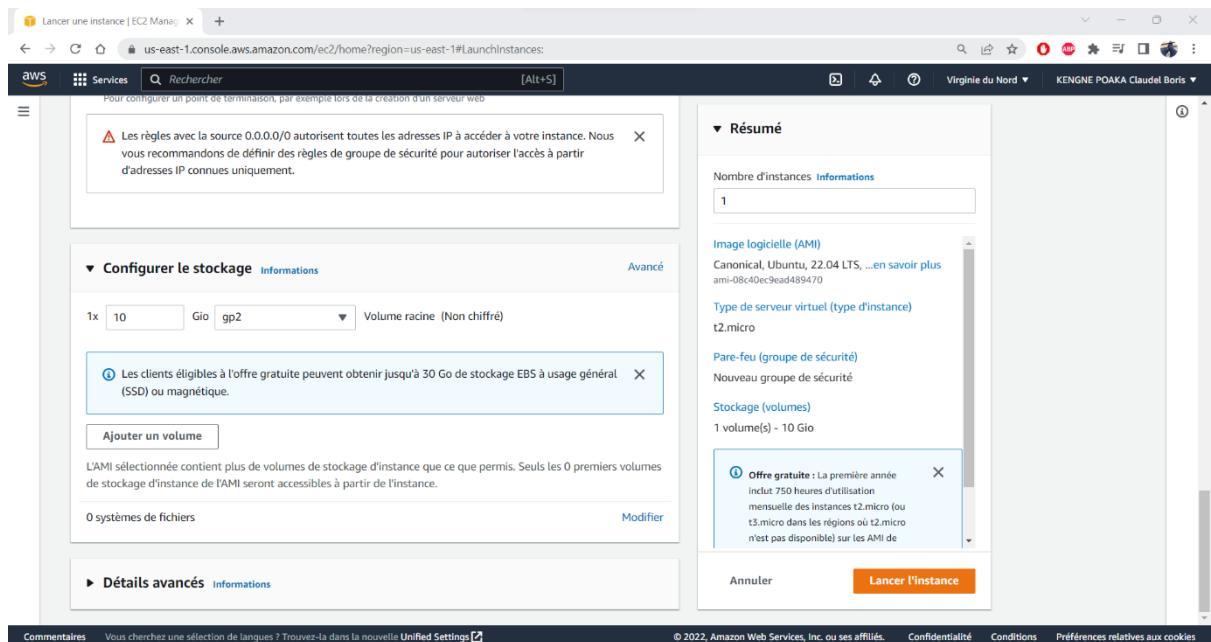


Figure 7 : Etape 5

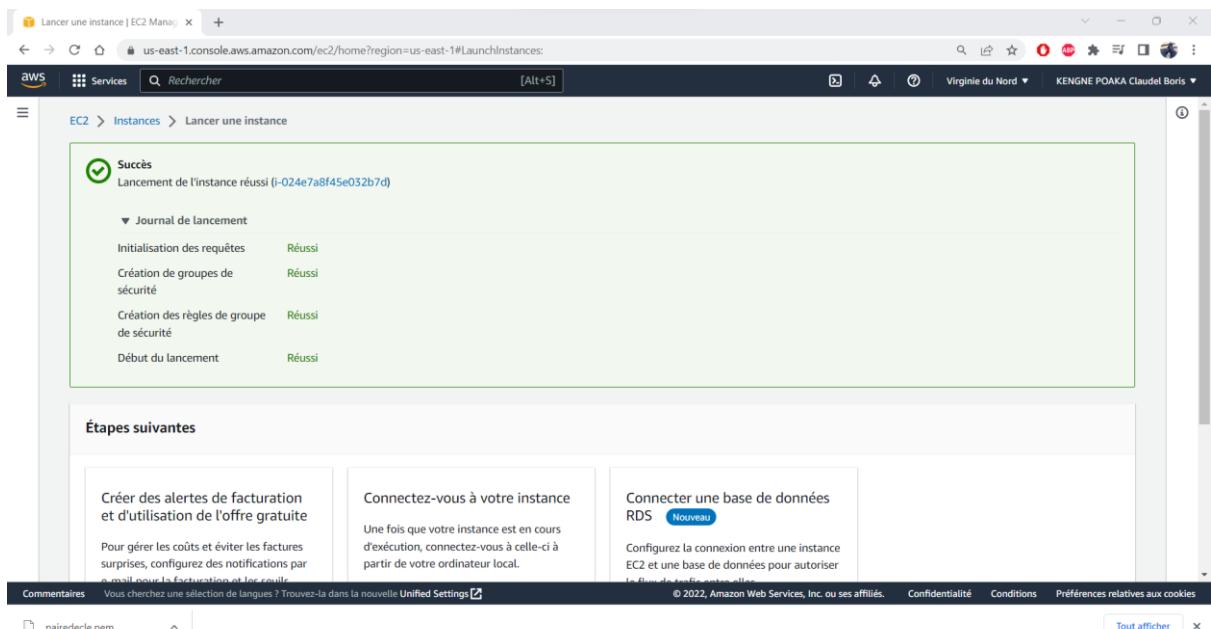


Figure 8 : Fin de la création de l'instance.

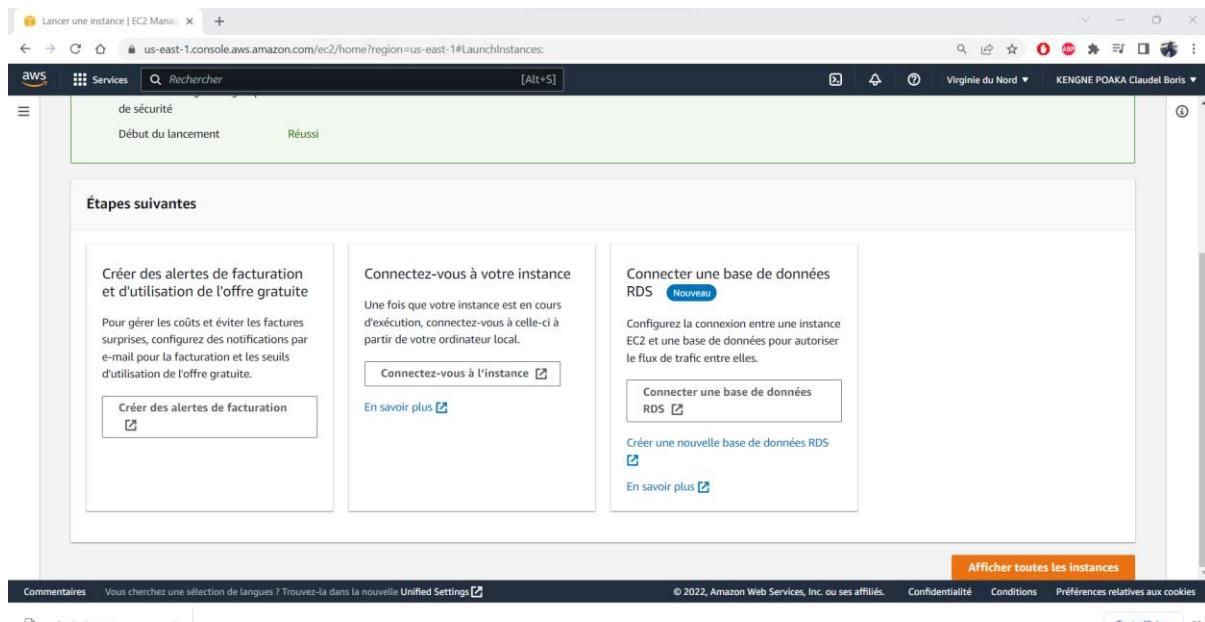


Figure 9 : Fin de la création de l'instance.

Réponses aux questions :

- Les états de la machine virtuelle que j'ai créée sont : **En cours d'exécution** ou **Arrêté**
- L'adresse IP publique de ma machine virtuelle est : **3.83.118.10** et l'adresse IP privée est : **172.31.92.45**

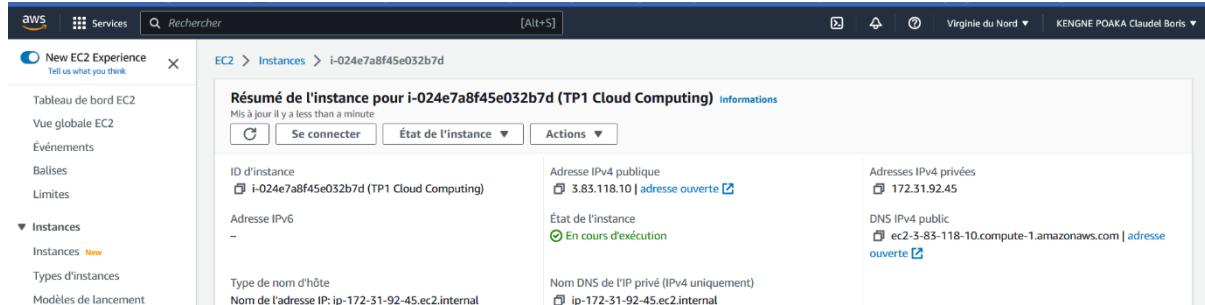


Figure 10 : Adresses IP publique et privée de ma machine virtuelle.

Partie 3 : Connexion à une VM

Ici j'ai choisi me connecter par **EC2 Instance Connect**

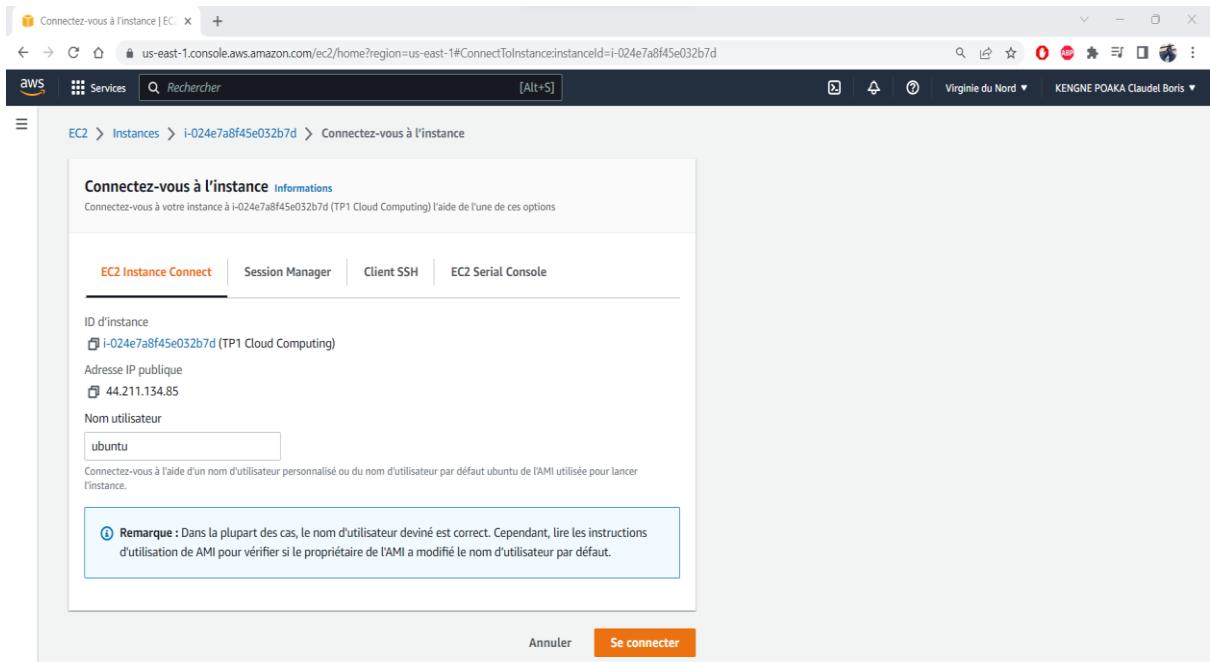


Figure 11 : Interface de connexion à distance par EC2 Instance Connect (nom d'utilisateur par défaut **ubuntu**)

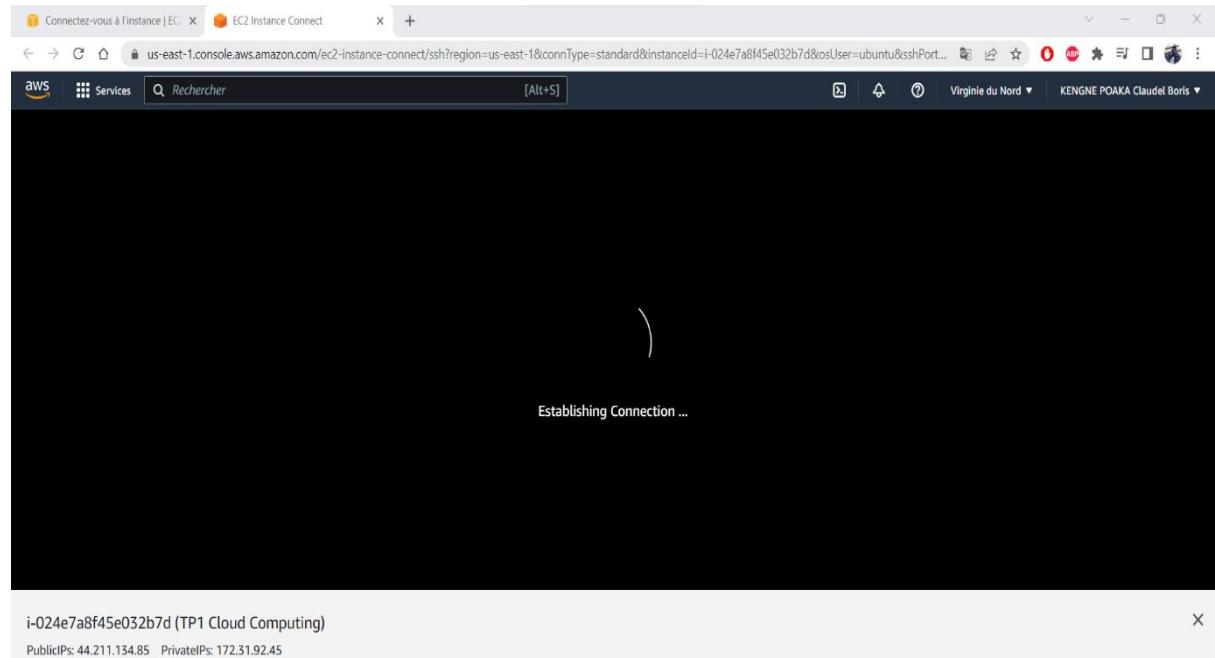


Figure 12 : Etablissement de la connexion à distance

Détails de l'instance | EC2 Manager EC2 Instance Connect

us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-024e7a8f45e032b7d&osUser=ubuntu®ion=us-east-1&sshPort...

aws Services Rechercher [Alt+S]

* Management: <https://landscape.canonical.com>
* Support: <https://ubuntu.com/advantage>

System information as of Mon Nov 21 11:15:35 UTC 2022

System load: 0.16015625 Processes: 97
Usage of /: 15.6% of 9.51GB Users logged in: 0
Memory usage: 21% IPv4 address for eth0: 172.31.92.45
Swap usage: 0%

0 updates can be applied immediately.

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-92-45:~\$

Figure 13 : Une fois la connexion établie on a le terminal de ci

Partie 4 : Notre première application

➤ Installation du serveur d'application nginx

```
$ sudo apt update
```

```
ubuntu@ip-172-31-92-45:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [725 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [163 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [10.9 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [445 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [68.3 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [584 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [755 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [125 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [4404 B]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [7216 B]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [2360 B]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [420 B]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [3008 B]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [1432 B]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [272 B]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 B]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [6740 B]
```

Figure 14 : Mise à jour des paquets de ma machine virtuelle.

```
$ sudo apt install nginx
```

Figure 15 : Téléchargement des paquets pour installation de nginx

➤ Installation de notre première application web

```
$ cd /usr/share/nginx/html
```

```
nginx: /usr/sbin/nginx /usr/lib/nginx /etc/nginx /usr/share/nginx /usr/share/man/man8/nginx.8.gz
ubuntu@ip-172-31-92-45:~$ cd /usr/share/nginx
ubuntu@ip-172-31-92-45:/usr/share/nginx$ ls
html modules modules-available
ubuntu@ip-172-31-92-45:/usr/share/nginx$ cd /usr/share/nginx/html
ubuntu@ip-172-31-92-45:/usr/share/nginx/html$ ls
index.html
```

Figure 16 : Contenu du dossier html de notre serveur web nginx

```
$ sudo apt install php-mbstring php-xml php-bcmath
```

```
ubuntu@ip-172-31-92-45:/usr/share/nginx/html$ sudo apt install php-mbstring php-xml php-bcmath  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  libbonig5 php-common php8.1-bcmath php8.1-common php8.1-mbstring php8.1-xml  
The following NEW packages will be installed:  
  libbonig5 php-bcmath php-common php-mbstring php-xml php8.1-bcmath php8.1-common php8.1-mbstring php8.1-xml  
0 upgraded, 9 newly installed, 0 to remove and 85 not upgraded.  
Need to get 1934 kB of archives.  
After this operation, 11.5 MB of additional disk space will be used.  
Do you want to continue? [Y/n] Y  
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libbonig5 amd64 6.9.7.1-2build1 [172 kB]  
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 php-common all 2:9.2ubuntu1 [12.4 kB]  
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-common amd64 8.1.2-2ubuntu2.8 [1126 kB]  
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 php8.1-bcmath amd64 8.1.2-1ubuntu2.8 [16.5 kB]  
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 php-bcmath all 2:8.1+92ubuntu1 [1834 B]  
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-mbstring amd64 8.1.2-2ubuntu2.8 [404 kB]  
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 php-mbstring all 2:8.1+92ubuntu1 [1844 B]  
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-xml amd64 8.1.2-1ubuntu2.8 [118 kB]
```

Figure 17 : Installation des dépendances de php

```
$ php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
```

```
ubuntu@ip-172-31-92-45:/usr/share/nginx/html$ sudo php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
ubuntu@ip-172-31-92-45:/usr/share/nginx/html$ ls
composer-setup.php  index.html
ubuntu@ip-172-31-92-45:/usr/share/nginx/html$
```

Figure 18 : Installation de composer

```
ubuntu@ip-172-31-92-45:~$ sudo php composer-setup.php --install-dir=/usr/local/bin --filename=composer
Could not open input file: composer-setup.php
ubuntu@ip-172-31-92-45:~$ cd /usr/share/nginx/html
ubuntu@ip-172-31-92-45:/usr/share/nginx/html$ ls
composer-setup.php  index.html
ubuntu@ip-172-31-92-45:/usr/share/nginx/html$ sudo php composer-setup.php --install-dir=/usr/local/bin --filename=composer
All settings correct for using Composer
Downloading...
Composer (version 2.4.4) successfully installed to: /usr/local/bin/composer
Use it: php /usr/local/bin/composer
ubuntu@ip-172-31-92-45:/usr/share/nginx/html$
```

Figure 19 : Installation de composer (suite)

```
$ sudo apt-get install curl php-cli php-mbstring git unzip
```

```
ubuntu@ip-172-31-92-45:/usr/share/nginx/html$ sudo apt-get install curl php-cli php-mbstring git unzip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
curl 7.40.0-1ubuntu0.6
php-cli 7.0.1+92ubuntu1.6
php-mbstring 7.0.1+92ubuntu1.6
git 2.34.1-1ubuntu1.5
unzip 6.0-2ubuntu3.1
0 upgraded, 4 newly installed, 0 to remove and 82 not upgraded.
Need to get 3790 kB in 0s (35.0 MB/s).
After this operation, 305 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 curl amd64 7.01.0-1ubuntu1.6 [194 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libcurl4 amd64 7.01.0-1ubuntu1.6 [290 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 git amd64 1:2.34.1-1ubuntu1.5 [3132 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 unzip amd64 6.0-2ubuntu3.1 [174 kB]
Fetched 3790 kB in 0s (35.0 MB/s)
Preparing to unpack .../curl_7.01.0-1ubuntu1.6_amd64.deb ...

```

Figure 20 : Installation d'un composant de php

```
$ sudo php composer-setup.php --install-dir=/usr/local/bin --filename=composer
```

```
ubuntu@ip-172-31-92-45:/usr/share/nginx/html$ sudo apt install php-cli
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
bzip2 mailcap mime-support php8.1-cli php8.1-opcache php8.1-readline
Suggested packages:
bzip2-doc php-pear
The following NEW packages will be installed:
bzip2 mailcap mime-support php8.1-cli php8.1-opcache php8.1-readline
0 upgraded, 7 newly installed, 0 to remove and 85 not upgraded.
Need to get 2277 kB of archives.
After this operation, 6945 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 bzip2 amd64 1.0.8-5build1 [34.8 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 mailcap all 3.70+nmulubuntu1 [23.8 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 mime-support all 3.66 [3696 B]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-opcache amd64 8.1.2-1ubuntu2.8 [365 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-readline amd64 8.1.2-1ubuntu2.8 [13.5 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-cli amd64 8.1.2-1ubuntu2.8 [1933 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 php8.1-readline amd64 8.1.2-1ubuntu2.8 [2234 B]
Fetched 2277 kB in 0s (20.3 MB/s)
Selecting previously unselected package bzip2.
(Reading database ... 64032 files and directories currently installed.)
```

Figure 21 : Installation de l'interface en ligne de commande de php

```
$ composer -version
```

```
ubuntu@ip-172-31-92-45:/usr/share/nginx/html$ composer --version
Composer version 2.4.4 2022-10-27 14:39:29
ubuntu@ip-172-31-92-45:/usr/share/nginx/html$
```

Figure 22 : Vérification de la version installée

➤ Installation de Laravel

```
$ composer create-project --prefer-dist laravel/laravel tp1
```

(On a eu besoin de l'exécuter en mode super utilisateur *sudo*)

```
ubuntu@ip-172-31-92-45:/usr/share/nginx/html$ sudo composer create-project --prefer-dist laravel/laravel tp1
Do not run Composer as root/super user! See https://getcomposer.org/root for details
Continue as root/super user [yes]? yes
Creating a "laravel/laravel" project at "./tp1"
Info from https://repo.packagist.org: #StandWithUkraine
Installing laravel/laravel (v9.3.11)
  - Installing laravel/laravel (v9.3.11): Extracting archive
Created project in /usr/share/nginx/html/tp1
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 107 installs, 0 updates, 0 removals
- Locking brick/math (0.10.2)
- Locking dflydev/dot-access-data (v3.0.2)
- Locking doctrine/inflector (2.0.6)
- Locking doctrine/instantiator (1.4.1)
- Locking doctrine/lexer (1.2.3)
- Locking dragonmantank/cron-expression (v3.3.2)
- Locking egulias/email-validator (3.2.1)
- Locking fakerphp/faker (v1.20.0)
- Locking filp/whoops (2.14.6)
- Locking fruitcake/php-cors (v1.2.0)
- Locking graham-campbell/result-type (v1.1.0)
- Locking guzzlehttp/guzzle (7.5.0)
- Locking guzzlehttp/promises (1.5.2)
- Locking guzzlehttp/psr7 (2.4.3)
- Locking hamcrest/hamcrest-php (v2.0.1)
```

Figure 23 : Création du projet tp1 dans Laravel

```
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
INFO Discovering packages.

laravel/sail ..... DONE
laravel/sanctum ..... DONE
laravel/tinker ..... DONE
nesbot/carbon ..... DONE
nunomaduro/collision ..... DONE
nunomaduro/termwind ..... DONE
spatie/laravel-ignition ..... DONE

81 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
> @php artisan vendor:publish --tag=laravel-assets --ansi --force
INFO No publishable resources for tag [laravel-assets].
No security vulnerability advisories found
> @php artisan key:generate --ansi
INFO Application key set successfully.

ubuntu@ip-172-31-92-45:/usr/share/nginx/html$ ls
composer-setup.php index.html tp1
ubuntu@ip-172-31-92-45:/usr/share/nginx/html/tp1$ ls
README.md artisan composer.json config lang phpunit.xml resources storage vendor
app bootstrap composer.lock database package.json public routes tests vite.config.js
```

Figure 24 : Création du projet tp1 dans Laravel (fin)

```
$ cd tp1
```

```
ubuntu@ip-172-31-92-45:/usr/share/nginx/html$ ls
composer-setup.php index.html tp1
ubuntu@ip-172-31-92-45:/usr/share/nginx/html$ cd tp1
ubuntu@ip-172-31-92-45:/usr/share/nginx/html/tp1$ ls
README.md artisan composer.json config lang phpunit.xml resources storage vendor
app bootstrap composer.lock database package.json public routes tests vite.config.js
```

Figure 25 : Contenue du dossier de notre projet

```
$ sudo chown -R www-data.www-data storage
```

```
$ sudo chown -R www-data.www-data bootstrap/cache
```

```
ubuntu@ip-172-31-92-45:/usr/share/nginx/html/tp1$ sudo chown -R www-data.www-data bootstrap/cache
ubuntu@ip-172-31-92-45:/usr/share/nginx/html/tp1$ ls
README.md artisan composer.json config lang phpunit.xml resources storage vendor
app bootstrap composer.lock database package.json public routes tests vite.config.js
ubuntu@ip-172-31-92-45:/usr/share/nginx/html/tp1$ []
```

Figure 26 : Attribution des droits de lecture au dossiers storage et cache

➤ Configuration de laravel dans nginx

```
$ sudo cp /etc/nginx/sites-available/default /etc/nginx/sites-available/tp1.com
```

```
ubuntu@ip-172-31-92-45:~$ cd /etc/nginx/sites-available
ubuntu@ip-172-31-92-45:/etc/nginx/sites-available$ ls
default tp1.com
```

Figure 27 : Résultat de la requête, configuration tp1.com à partir de default

```
$ sudo nano /etc/nginx/sites-available/tp1.com
```

```
# Virtual Host configuration for example.com
#
# You can move that to a different file under sites-available/ and symlink that
# to sites-enabled/ to enable it.
#
server {
    listen 80;
    listen [::]:80;

    server_name 54.205.119.187;

    root /var/www/html/tp1/public;
    index index.php index.html index.htm;

    location / {
        try_files $uri $uri/ /index.php?$query_string;
    }
}

File Name to Write: /etc/nginx/sites-available/tp1.com
^G Help          M-D DOS Format      M-A Append      M-B Backup File
^C Cancel        M-M Mac Format      M-P Prepend    ^T Browse
```

Figure 28 : Modification des paramètres de configuration et enregistrement

```
$ sudo ln -s /etc/nginx/sites-available/tp1.com /etc/nginx/sites-enabled/
```

```
ubuntu@ip-172-31-92-45:~$ sudo ln -s /etc/nginx/sites-available/tp1.com /etc/nginx/sites-enabled/
ubuntu@ip-172-31-92-45:~$ []
```

Figure 29 :

```
$ sudo systemctl reload nginx
```

```
ubuntu@ip-172-31-92-45:/etc/nginx$ sudo systemctl reload nginx
nginx.service is not active, cannot reload.
ubuntu@ip-172-31-92-45:/etc/nginx$ sudo systemctl start nginx
ubuntu@ip-172-31-92-45:/etc/nginx$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
  Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
  Active: active (running) since Mon 2022-11-28 14:31:46 UTC; 10s ago
    Docs: man:nginx(8)
   Process: 6914 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited)
   Process: 6915 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited)
    Main PID: 6916 (nginx)
      Tasks: 2 (limit: 1143)
     Memory: 3.6M
        CPU: 27ms
      CGroup: /system.slice/nginx.service
              ├─6916 "nginx: master process /usr/sbin/nginx -g daemon on; master_process o>
              └─6917 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" "" "" "" "" >
Nov 28 14:31:46 ip-172-31-92-45 systemd[1]: Starting A high performance web server and a r>
Nov 28 14:31:46 ip-172-31-92-45 systemd[1]: Started A high performance web server and a r>
lines 1-16/16 (END) []
```

Figure 30 : Etat de notre serveur de fichier nginx

En tapant l'adresse ip publique de notre instance, notre navigateur télécharge un fichier .php

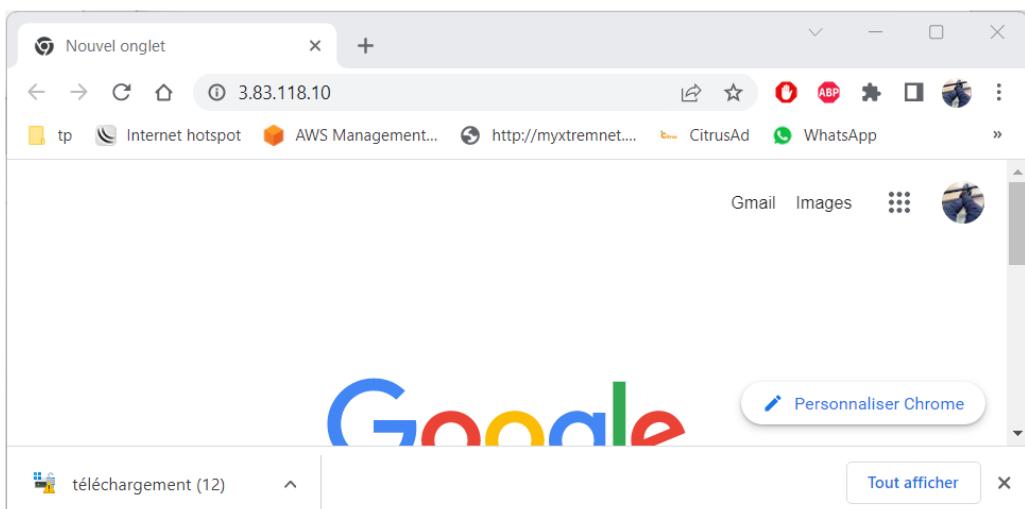


Figure 31 :

Partie 5 : Connexion à une base de données du cloud

Dans cette partie, nous allons créer une base de données et la connecter à notre serveur de fichiers.

Commençons par choisir le service de AWS qui offre la possibilité de créer les bases de données

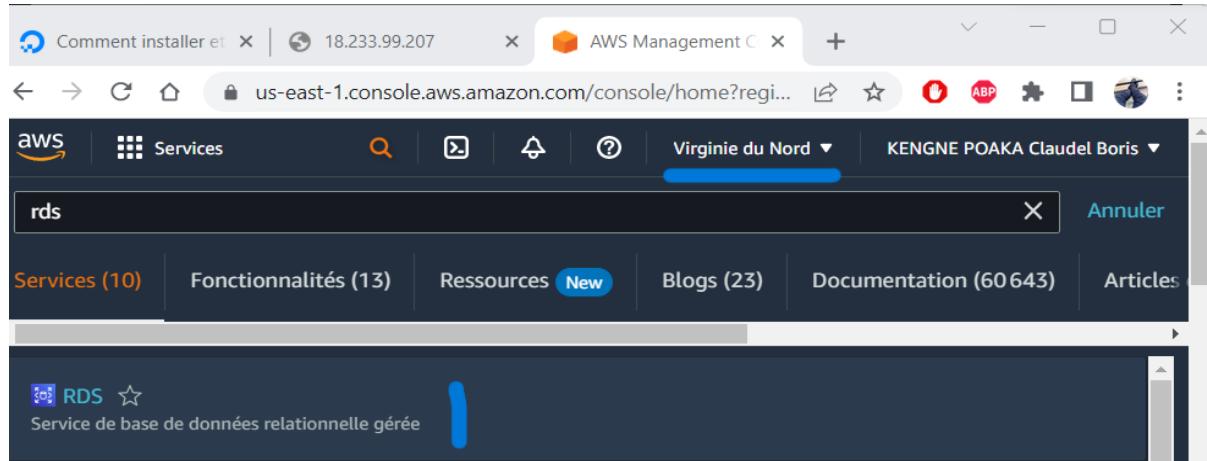


Figure 32 : Choix du service RDS

Etapes de création de notre base de données :

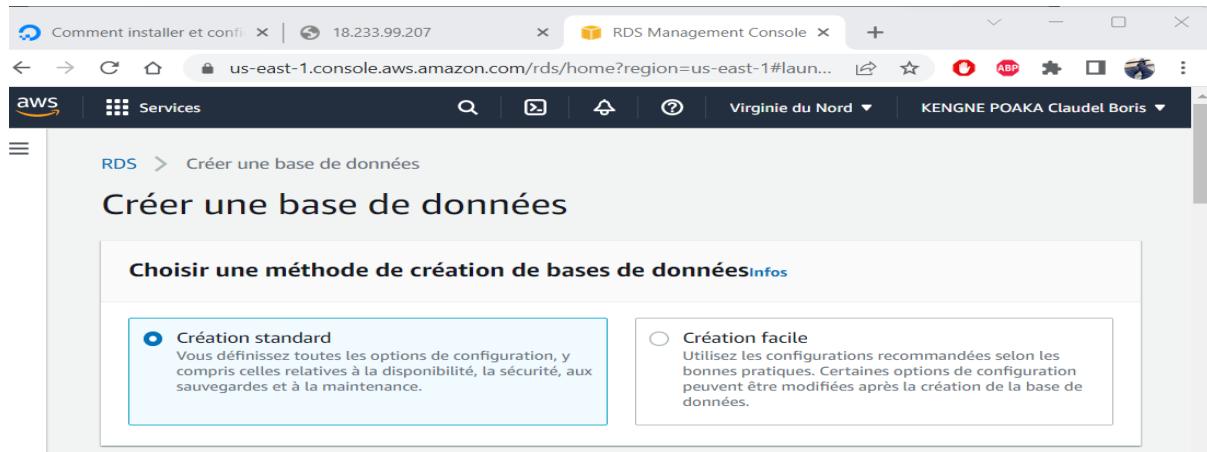


Figure 33 : Choix de création standard.

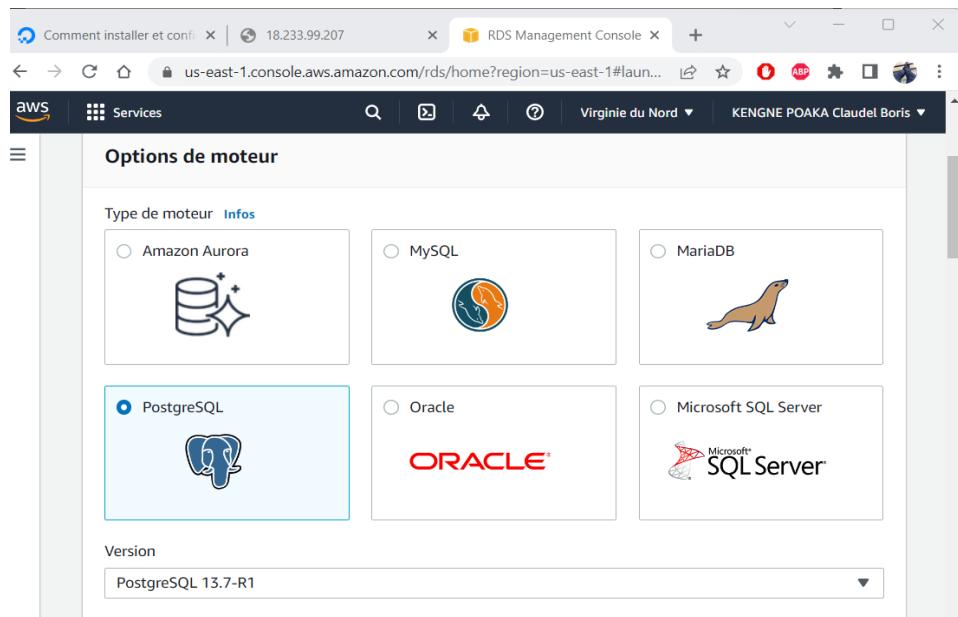


Figure 34 : Choix du type de moteur (moteur PostgreSQL dans notre cas et version 13.7-R1).

Par la suite nous avons choisi le modèle d'**offre gratuite**.

Identifiant d'instance de base de données	Infos
Saisissez un nom pour votre instance de base de données. Le nom doit être unique parmi toutes les instances de base de données appartenant à votre compte AWS dans la région AWS actuelle.	
<input type="text" value="database-1"/>	
L'identifiant de l'instance de base de données n'est pas sensible à la casse, mais stocké intégralement en minuscules (comme dans « mydbinstance »). Contraintes : doit contenir entre 1 et 60 caractères alphanumériques ou traits d'union. Le premier caractère doit être une lettre. Ne peut pas contenir deux traits d'union consécutifs. Ne peut pas se terminer par un trait d'union.	
▼ Configuration des informations d'identification	
Identifiant principal	Infos
Saisissez un ID de connexion pour l'utilisateur principal de votre instance de base de données.	
<input type="text" value="root"/>	
Entre 1 et 16 caractères alphanumériques. Le premier caractère doit être une lettre	
<input type="checkbox"/> Générer automatiquement un mot de passe Amazon RDS peut générer un mot de passe pour vous. Vous pouvez aussi spécifier votre propre mot de passe.	
Mot de passe principal	Infos
<input type="password" value="*****"/>	
Contraintes : au moins 8 caractères ASCII imprimables. Ne peut pas contenir les caractères suivants : / (barre oblique), ' (apostrophe), " (guillemets) et @ (arobase).	
Confirmer le mot de passe	Infos
<input type="password" value="*****"/>	

Figure 35 : Paramètres de notre base de données.

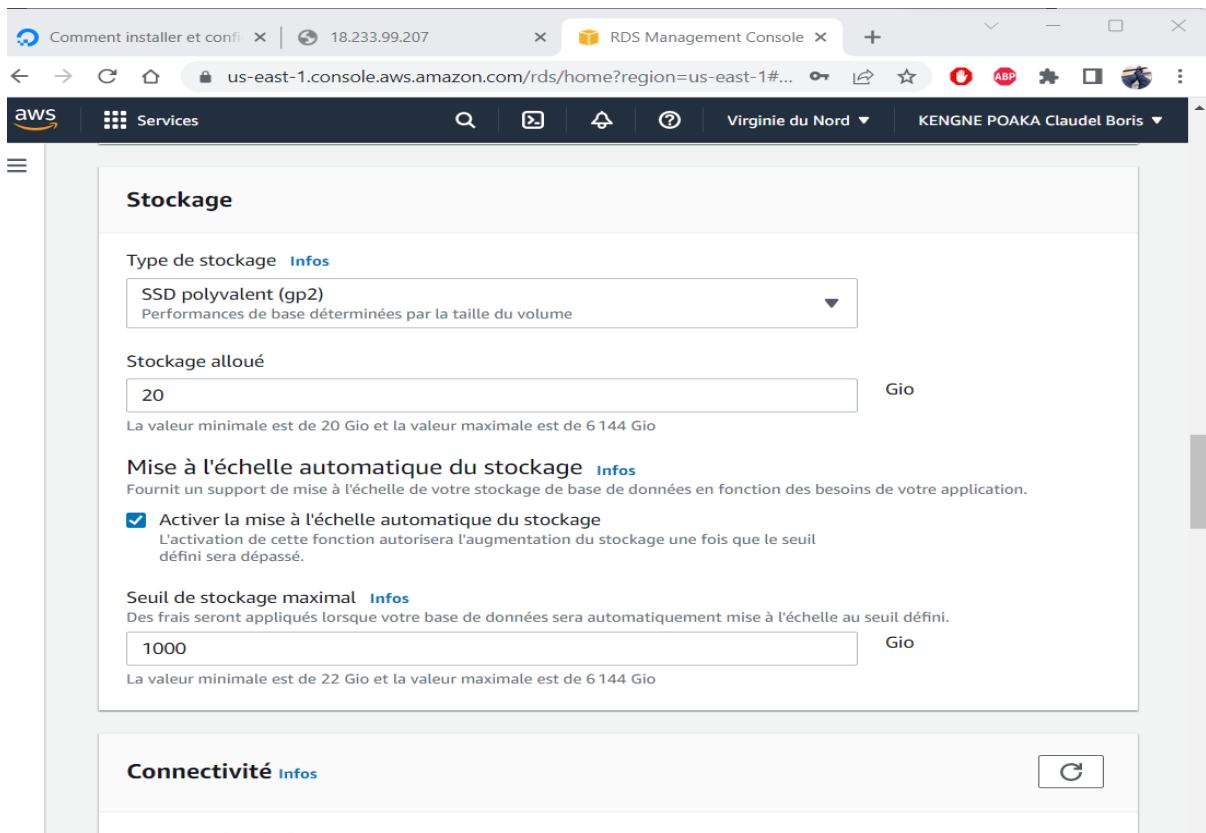


Figure 36 : Options de stockage de notre base de données.

Pour la configuration de notre base de données, on a les paramètres suivants :

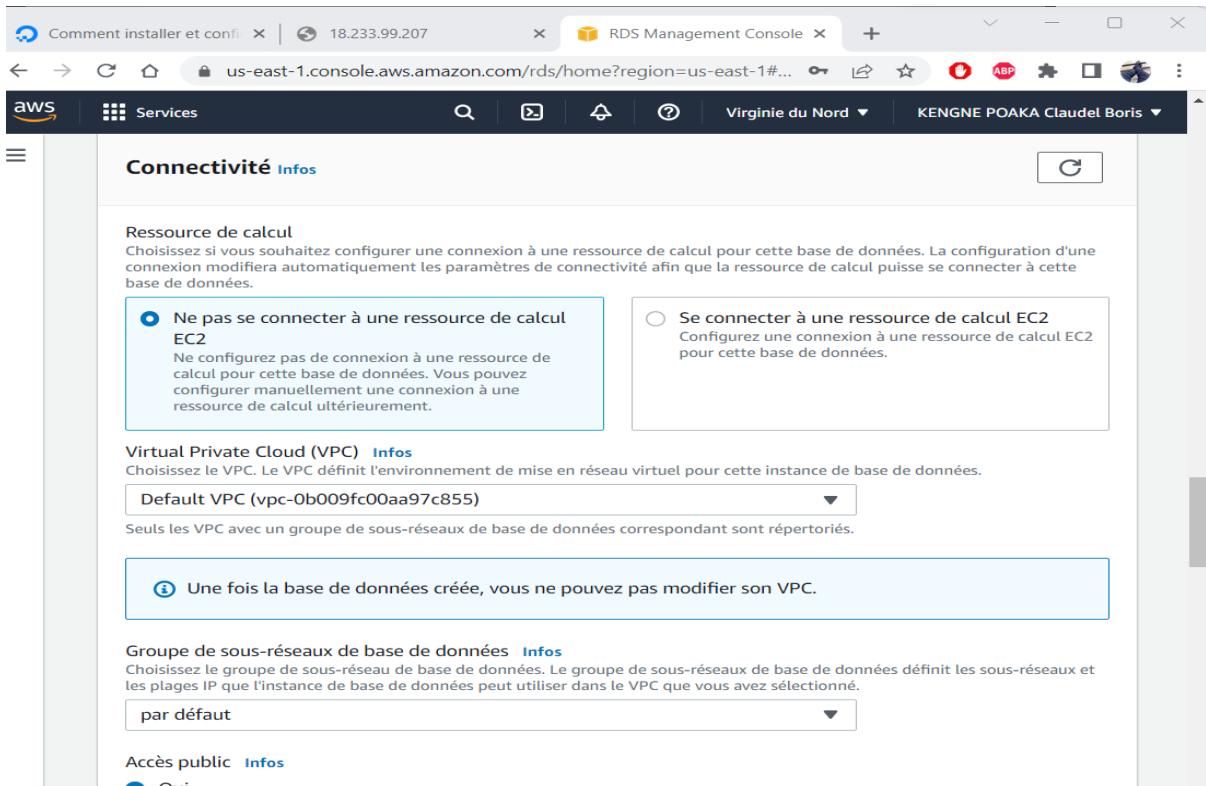


Figure 37 : Options de connectivité

Accès public [Infos](#)

Oui
RDS attribue une adresse IP publique à la base de données. Les instances Amazon EC2 et les autres ressources en dehors du VPC peuvent se connecter à votre base de données. Les ressources à l'intérieur du VPC peuvent également se connecter à la base de données. Choisissez un ou plusieurs groupes de sécurité VPC qui spécifient quelles ressources peuvent se connecter à la base de données.

Non
RDS n'attribue pas d'adresse IP publique à la base de données. Seules les instances Amazon EC2 et les autres ressources à l'intérieur du VPC peuvent se connecter à votre base de données. Choisissez un ou plusieurs groupes de sécurité VPC qui spécifient quelles ressources peuvent se connecter à la base de données.

Groupe de sécurité VPC (pare-feu) [Infos](#)
Choisissez un ou plusieurs groupes de sécurité VPC pour autoriser l'accès à votre base de données. Assurez-vous que les règles du groupe de sécurité autorisent le trafic entrant approprié.

Choisir existants
Choisir les groupes de sécurité VPC existants

Créer
Créer un groupe de sécurité VPC

Groupes de sécurité VPC existants
Choose one or more options

default [X](#)

Zone de disponibilité [Infos](#)
Aucune préférence

Proxy RDS
Le proxy RDS est un proxy de base de données hautement disponible et entièrement géré qui améliore la capacité de mise à l'échelle, la résilience et la sécurité des applications.

Créer un proxy RDS [Infos](#)

Figure 38 : Options d'accès public.

Nous avons choisi l'authentification de la base de données **par mot de passe** et après avoir cliqué sur « **Créer une base de données** » on est conduit sur l'interface ci :

Création de la base de données database-1
Votre base de données peut prendre quelques minutes pour se lancer.

[Afficher les détails des informations d'identification](#)

Bases de données

Identifiant de base de données	Rôle	Moteur	Région et AZ	Taille	Statut	Processeur
database-1	Instance	PostgreSQL	-	db.t3.micro	Création en cours	-

Figure 39 : Finalisation et lancement de notre base de données.

Partie 6 : Connexion de notre application à notre base de données

\$ cd /usr/share/nginx/html/tp1

```
ubuntu@ip-172-31-92-45:/usr/share/nginx/html$ cd tp1
ubuntu@ip-172-31-92-45:/usr/share/nginx/html/tp1$ ls
README.md  bootstrap    config    package.json  resources   tests
app        composer.json database  phpunit.xml  routes     vendor
artisan    composer.lock lang      public      storage   vite.config.js
```

Figure 40 : Contenu de notre dossier tp1

\$ sudo nano .env

```
GNU nano 6.2
.env *
APP_NAME=Laravel[]
APP_ENV=local
APP_KEY=base64:3ESs/xzY1Ged2zJEADXI5+cuEGrMV6d5v/WOG/4hLBw=
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug

DB_CONNECTION=mysql
DB_HOST=database-1.citjkkcqni3d.us-east-1.rds.amazonaws.com
DB_PORT=3306
DB_DATABASE=database-1
DB_USERNAME=root
DB_PASSWORD=admin123

BROADCAST_DRIVER=log
CACHE_DRIVER=file
FILESYSTEM_DISK=local
QUEUE_CONNECTION=sync
SESSION_DRIVER=file
```

Figure 41 : Paramètres de notre fichier de configuration Laravel.

\$ php artisan make:auth

```
ubuntu@ip-172-31-92-45:/usr/share/nginx/html/tp1$ sudo php artisan make:auth
ERROR | Command "make:auth" is not defined. Did you mean one of these?

| make:cast
| make:channel
| make:command
| make:component
| make:controller
| make:event
| make:exception
| make:factory
| make:job
| make:listener
| make:mail
| make:middleware
| make:migration
| make:model
```

Figure 42 : On rencontre une erreur, notre terminal ne reconnaît pas la commande.