

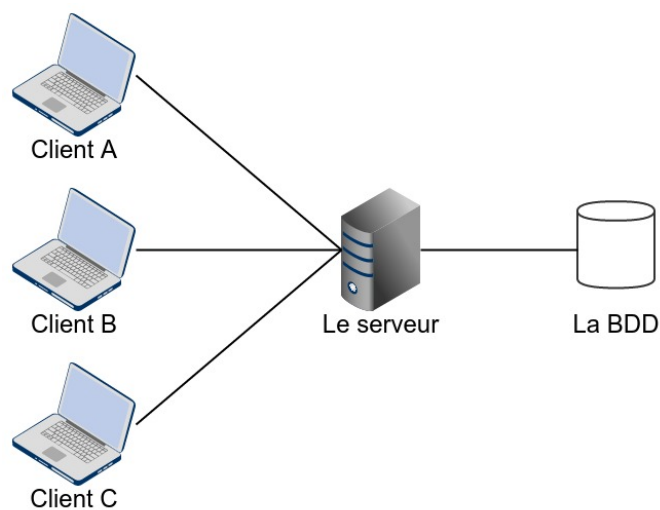
## PROJET DE CLOUD COMPUTING

### Objectifs

Ce TP a pour objectif la réalisation d'une architecture web basée sur le cloud computing. Pour ce faire, nous avons opté pour Amazon Aws qui est le leader dans le domaine.

A la fin de ce TP, chaque étudiant devra être capable de :

1. Réaliser l'architecture suivante sur Amazon Aws:



2. Créer une machine virtuelle et y accéder avec un client SSH
3. Créer une base de données et y accéder avec le client Workbench
4. Créer une application Laravel et déployer cette application sur un serveur d'application nginx

### Description

Amazon Web Services (AWS) est une plateforme évolutive complète de Cloud computing proposée par Amazon.com. Il est le fournisseur de cloud le plus sollicité dans le monde entier. Cette première partie du TP consistera à créer un compte sur amazon aws.

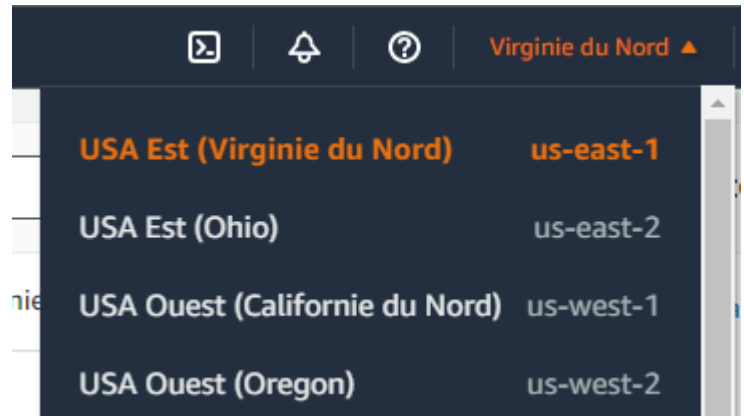
### Partie 1 : Création d'un compte sur Amazon AWS

1. Ouvrir le lien <https://aws.amazon.com> sur votre navigateur.

2. Cliquez sur **Créer un compte AWS gratuit**
3. Suivez ensuite les étapes de création de votre compte, en indiquant que vous voulez créer un compte gratuitement

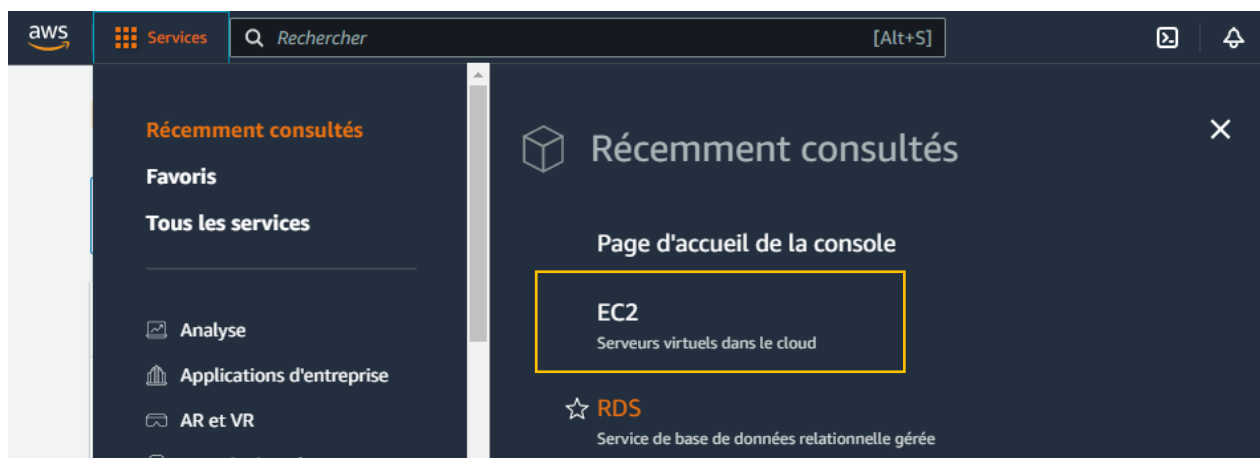
## Partie 2 : Service EC2


- **EC2 (Elastic Compute Cloud)** est un service de AWS qui vous permet de créer une machine virtuelle sur le Cloud d'Amazon.
- Un compte AWS fournit plusieurs Régions afin que vous puissiez lancer des instances Amazon EC2 dans des emplacements qui satisfont vos exigences. Par exemple, vous pouvez souhaiter lancer des instances en Europe afin d'être plus proche de vos clients européens ou pour satisfaire à des exigences légales.



### Création d'une machine virtuelle

1. Connectez-vous à votre compte créé précédemment pour accéder à votre console.
2. Cliquez sur services, puis sur EC2



- Sur le menu de gauche, vous avez la possibilité d'afficher la liste de vos machines virtuelles. Cliquez sur 
- Sur la nouvelle interface, choisissez les caractéristiques suivantes pour la création de votre machine virtuelle.
  - Le nom **TP1 Cloud Computing**
  - Une image 64 bits (x86) **Ubuntu Server 22.04 LTS (HVM), SSD Volume Type**
  - Une instance de type **t2.micro (1 vCPU, 1GiB RAM)**
  - Ne mettez aucune paire de clé. Le service EC2 se chargera de le faire. Une fois téléchargé, copiez-le en lieu sûr, car elle est générée une seule fois. Cette clé sera utilisée par le client SSH pour accéder à la machine virtuelle.
  - Laissez les paramètres réseau par défaut
  - Au niveau du stockage, augmentez à 10GiB pour un volume SSD à usage général (gp2)
  - Cliquez ensuite sur lancer l'image

#### Questions :

- Quel sont les différent état la machine virtuelle que vous avez créée ?
- Quelles sont les adresse IP publique et privée de votre machine virtuelle ?

### Partie 3 : Connexion à une VM

---

Sur la VM précédente, cliquez sur connexion. Dans la nouvelle interface, vous avez tous les méthodes de connexion à la machine virtuelle.

**Connectez-vous à l'instance** [Informations](#)

Connectez-vous à votre instance à i-02add163c76126929 (Resand Pay VPS) l'aide de l'une de ces options

EC2 Instance Connect

**Session Manager**

Client SSH

EC2 Serial Console

## Connexion par EC2 Instance Connect

- Cliquez sur l'onglet EC2 Instance Connect.
- Réinsérez ensuite le nom d'utilisateur associé utilisé lors de la création de la machine virtuelle. Par défaut, le nom qui s'affiche est le bon.
- Cliquez ensuite sur **Se connecter**. Vous allez alors voir l'interface suivante vous permettant de contrôler votre VM.

```
https://ubuntu.com/aws/pro

58 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

New release '22.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***

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.
```

## Connexion par un client SSH

- Cliquez sur l'onglet **client SSH**
- Vous avez alors la procédure pour vous connecter à votre VM, soit en utilisant un client spécialisé, soit en passant directement par la console (terminal) du système d'exploitation

### Console de l'OS :

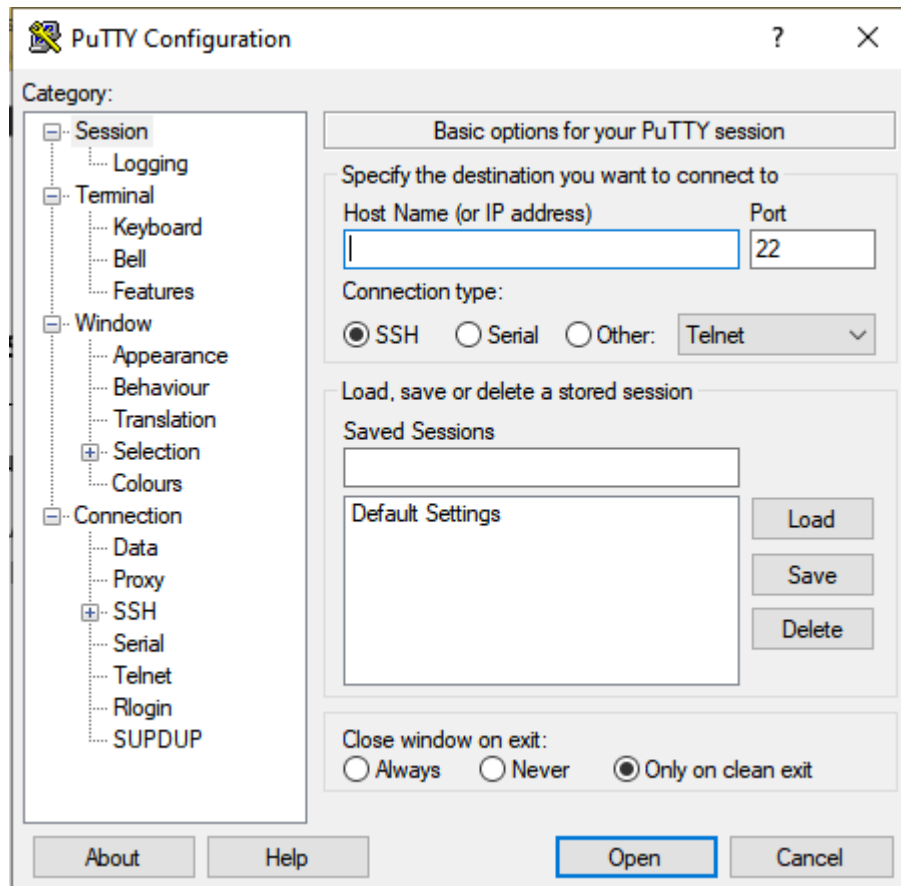
Entrer la commande suivante dans le terminal : `ssh -i "Clé Instance.pem" nom-utilisateur@adresse-instance`.

- o « **Clé Instance.pem** » est la clé que vous avez téléchargé lors de la création. Vous devez renseigner le chemin complet.
- o **nom-utilisateur** est le nom d'utilisateur renseigné toujours lors du processus de création de la VM
- o **adresse-instance** est l'adresse ip publique ou le nom de domaine de votre VM.

## Client SSH spécialisé : PuTTY

PuTTY est un client SSH Open source, développé à l'origine par Simon Tatham pour la plateforme Windows. Nous allons l'utiliser dans le cadre de ce TP

- Allez sur le site <https://www.putty.org> et téléchargez PuTTY
- Installez le logiciel et lancez-le



- Au niveau de nom d'hôte, renseignez l'adresse IP publique ou le nom de domaine de votre machine virtuelle.
- Vérifiez que vous êtes sur le port 22
- Cliquez ensuite sur SSH, puis sur Auth et téléchargez la clé que vous avez téléchargée lors de la création de votre VM. Cette clé sera utilisée par le client SSH pour vous connecter
- Cliquez ensuite sur Open et renseignez les informations comme nom d'utilisateur et mot de passe en cas de besoin.

## Partie 4 : Notre première application

---



Laravel est un framework qui nous permettra de développer nos applications web rapidement avec un très grand confort. Il est possible d'y ajouter plusieurs packagent qui augmenter ses fonctionnalités. Il est basé sur le Framework Symphonie.

### Installation du serveur d'application nginx

Nginx (prononcé "Engine-X") est un serveur web open source souvent utilisé comme reverse proxy ou cache HTTP. Il est disponible gratuitement pour Linux.

- Connectez vous à votre VM par l'une des méthodes évoquées plus haut
- Taper ensuite les commandes suivantes :
  - o **\$ sudo apt update**
  - o **\$ sudo apt install nginx**
- Ouvrez ensuite votre navigateur et tapez l'adresse ip publique ou le nom de domaine de votre VM. Que constatez-vous ?

### Installation de notre première application Web

Nous allons maintenant installer et configurer notre application laravel.

- Ouvre l'invite de commande et tapez les commandes suivantes :
  - o **\$ cd /usr/share/nginx/html**
- Nous allons maintenant installer les dépendances de php par les commandes suivantes
  - o **\$ sudo apt install php-mbstring php-xml php-bcmath**
- Nous allons maintenant installer composer qui va nous aider à créer un projet laravel
  - o **\$ php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"**
  - o **\$ COMPOSER=CHAINE COMPOSER** (Chaîne de caractère affiché par la commande précédente)
  - o **\$ sudo apt-get install curl php-cli php-mbstring git unzip**

- `$ sudo php composer-setup.php --install-dir=/usr/local/bin --filename=composer`
- `$ composer --version` (Pour vérifier que tout s'est bien passé)
- Nous pouvons à présent installer laravel. Nous allons l'appeler simplement tp1
  - `$ composer create-project --prefer-dist laravel/laravel tp1`
  - `$ cd tp1`
  - `$ sudo chown -R www-data:www-data storage`
  - `$ sudo chown -R www-data:www-data bootstrap/cache`
- Configuration de Nginx pour lancer notre application  
 Nous allons maintenant configurer un hôte virtuel pour pouvoir accéder à notre application à partir du navigateur. Cette application doit être enregistrée dans le serveur d'application Nginx. Pour ce faire, suivez les étapes suivantes :
  - `$ sudo nano /etc/nginx/sites-available/tp1.conf`

## Configuration de laravel dans nginx

Nous allons maintenant configurer une hôte virtuel dans le serveur nginx. Cet hôte nous permettra à nginx de rediriger les requêtes dans notre application laravel. Suivez les étapes suivantes :

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

Cette commande va créer un nouveau fichier de configuration **tp1.com** à partir du fichier de configuration de nginx par défaut.

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

Cette commande nous permettra de modifier notre fichier tp1.com précédemment créé. Nous allons modifier les paramètres suivants :

- ❖ Enlever la désignation `default_server`,
- ❖ Modifier la racine du site web en changeant la directive `root`,
- ❖ Modifier le nom du serveur en changeant la directive `server_name` qui pointera vers un nom de domaine particulier. (Pour le test, nous allons mettre unique l'adresse IP)
- ❖ Mettre à jour URI des requêtes via la directive `try_files`.

```
server {
    listen 80;
    listen [::]:80;

    . . .

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

    server_name // example.com www.example.com;

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

    . . .
}
```

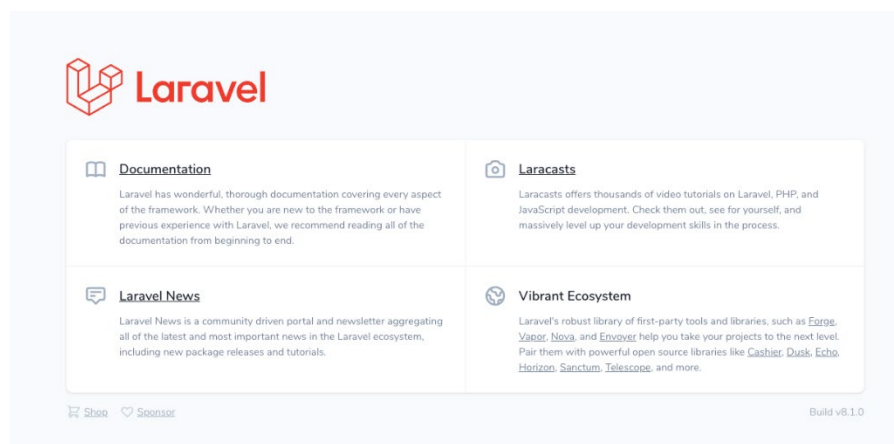
Nous allons maintenant activer la nouvelle configuration en créant un lien symbolique du fichier vers le dossier des sites web activés **sites-enabled** . Cela se fait par la commande suivante :

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

Nous allons maintenant configurer relancer le serveur nginx.

```
$ sudo systemctl reload nginx
```

Ouvrez le navigateur et taper l'adresse ip publique de notre instance. Vous devriez avoir un page semblable à celle-ci en fonction de la version que vous avez installé.





## Partie 5 : Connexion à une base de données du cloud

Nous avons déjà créé notre application web sur le cloud. Nous allons maintenant connecter cette application web à une base de données du cloud, en particulier le service RDS.

- Au niveau de votre tableau de bord, vous allez cliquer sur **services**, juste en haut et à gauche de la page.
- Vous allez ensuite rechercher le services ADS et cliquer dessus.
- Vous devez faire attention à la région dans laquelle la nouvelle instance de base données va être créée. Elle peut être différente de celle dans laquelle vous avez créé la machine virtuelle.

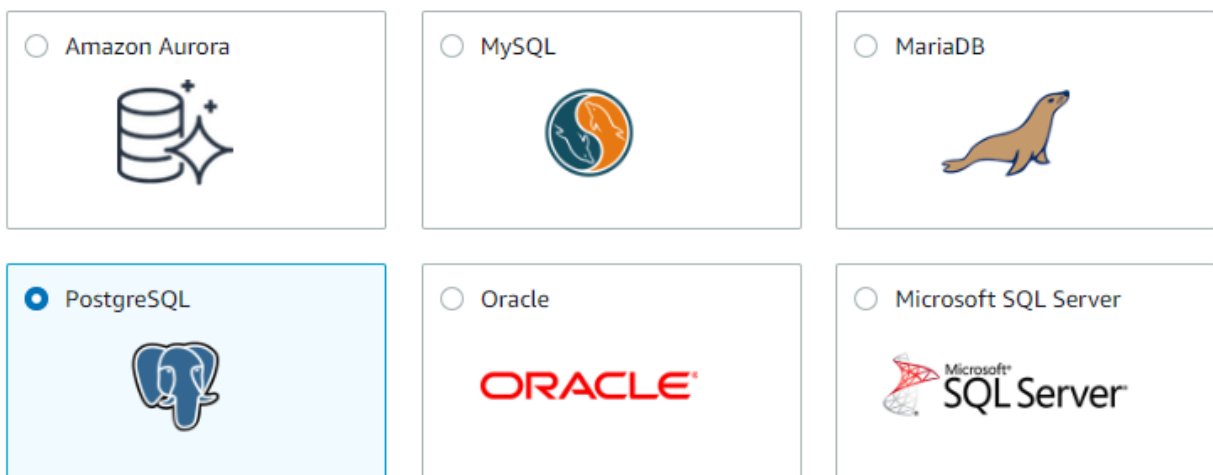


- Dans la nouvelle interface, cliquez sur

**Créer une base de données**

Amazon nous propose plusieurs Systèmes de Gestion de Base de Données relationnelle.

Type de moteur [Infos](#)



- Pour notre TP, nous allons prendre la base de données MySQL.
- Choisissez alors la version de la base de données
- Nous allons prendre une version gratuite de la base de données

<input type="radio"/> <b>Production</b> Utilisez les valeurs par défaut pour la haute disponibilité et pour des performance rapides, uniformes.	<input type="radio"/> <b>Dev/Test</b> Cette instance est destinée au développement en dehors d'un environnement de production.	<input checked="" type="radio"/> <b>Offre gratuite</b> Utilisez l'offre gratuite RDS afin de développer de nouvelles applications, de tester des applications existantes ou de vous familiariser avec Amazon RDS. <a href="#">Infos</a>
--	---	--

- Nous allons maintenant configurer notre base donnée. Je vous laisse le choix des paramètres de votre base de données. Vous allez néanmoins noter les paramètres suivants quelque part (ça nous servira par la suite) : identifiant principal, mot de passe. Vous pouvez aussi générer un mot de passe aléatoirement.

appartenant à votre compte AWS dans la région AWS actuelle.

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.

admin

Entre 1 et 16 caractères alphanumériques. Le premier caractère doit être une lettre

☐ 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](#)

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](#)

- Nous allons ensuite rendre notre base de données accessible par d'autre application d'interrogation de base de données comme Workbench.

#### 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.

- Cliquez ensuite sur

**Créer une base de données**

Vous allez être redirigé dans la liste des instances suivante :

Identifiant de base de données	Rôle	Moteur	Région et AZ	Taille	Statut
database-1	Instance	MySQL Community	-	db.t3.micro	Stopped

- Cliquez sur le nom de votre instance et copier le paramètre **Point de terminaison** : Il nous servira d'heure.

## Partie 5 : Connexion de notre application à notre base de données.

- Vous allez tout d'abord installer le service de base d'authentification sur Laravel qui n'est pas actif par défaut.

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

- Création des pages et services d'authentification

```
$ php artisan make:auth
```

- Vous avez alors de nouvelles pages qui sont créées et accessibles via /login, /register, etc.
- Nous allons maintenant ouvrir le fichier de configuration de Laravel nommé .env

```
sudo nano .env
```

Modifier les paramètres suivants :

```
DB_CONNECTION=mysql
```

```
DB_HOST=// Mettre le nom d'hôte que vous avez noté sur Amazon ADS
DB_PORT=3306
DB_DATABASE= // Mettre le nom de la base de données ADS
DB_USERNAME= // Nom d'utilisateur ADS ou admin par défaut
DB_PASSWORD= // Mettre le mot de passe ADS
```

Nous allons maintenant initialiser la base de données. Si vous avez été un peu curieux, vous auriez vu dans le dossier **database/migrations**, la liste des migrations qui permet générer le schéma relationnel de notre base de données sans avoir besoin d'écrire un code SQL. Ce dossier contient les migrations d'authentification de laravel. Nous allons maintenant lancer les migrations.

```
php artisan migrate
```

- Essayer maintenant de créer un compte par le lien /register
- Essayer de vous connecter /login
- Enjoy

Nous avons réalisé une architecture 3 tiers sur le cloud. Compléter le rapport et renvoyez-le-moi. Nous allons corriger ensemble.