



# Installer une application flask sur AWS

|  |           |
|--|-----------|
| <b>I. CREATION D'UNE MACHINE VIRTUELLE LINUX VISIBLE SUR INTERNET .....</b>        | <b>3</b>  |
| 1. COMPTE AWS .....  | 3         |
| 2. VERIFIER LE COUT .....  | 4         |
| 3. CREATION D'UNE INSTANCE EC2 .....   | 5         |
| <b>II. ACCES A DISTANCE .....</b>  | <b>16</b> |
| 1. COMMANDE A EXECUTER .....   | 16        |
| 2. LA PLUS SIMPLE : DANS UN NAVIGATEUR .....                                       | 16        |
| 3. LE PLUS PERFORMANT : UTILISATION D'UN CLIENT SSH .....                          | 18        |
| a. <i>Un exemple de commande est donné :</i> .....                                 | 18        |
| b. <i>Modification des droits du fichier de clé</i> .....                          | 19        |
| c. <i>Accès à distance par ssh</i> .....   | 20        |
| d. <i>Copier/coller dans le terminal</i> .....                                     | 21        |
| <b>III. FIN DE LA CONFIGURATION LINUX .....</b>                                    | <b>22</b> |
| 1. PREPARATION ET MISE A JOUR DE LA DISTRIBUTION .....                             | 22        |
| 2. EXPLICATIONS IMPLEMENTATION CHOISIE .....                                       | 23        |
| a. <i>Serveur wsgi</i> .....   | 24        |
| b. <i>Pourquoi ne pas utiliser python/flask seul ?</i> .....                       | 24        |
| c. <i>Schéma de déploiement le plus souvent utilisé</i> .....                      | 24        |
| d. <i>Principe de fonctionnement</i> .....   | 24        |
| <b>IV. INSTALLATION DES APPLICATIONS SUR LE SERVEUR LINUX AWS .....</b>            | <b>25</b> |
| 1. FLASK ET APPLICATIONS .....   | 25        |
| a. <i>Environnement virtuel</i> .....  | 25        |
| b. <i>Fin installation des modules</i> .....                                       | 25        |
| 2. APPLICATION FLASK DE TEST .....   | 26        |
| 3. CONFIGURATION DE NGINX .....  | 27        |
| a. <i>Récupération de l'adresse publique si vous êtes dans le navigateur</i> ..... | 27        |
| b. <i>Si vous êtes en ssh</i> .....  | 28        |
| c. <i>Fichier de configuration nginx</i> .....                                     | 28        |
| d. <i>Ouverture du port 80</i> .....   | 29        |
| <b>V. TESTS DE FONCTIONNEMENT .....</b>  | <b>33</b> |
| 1. SUR LE PORT 80 .....  | 33        |
| 2. SUR UN AUTRE PORT : LE 8080 PAR EXEMPLE .....                                   | 33        |
| 3. ATTENTION .....   | 33        |
| <b>VI. FINALISATION .....</b>  | <b>33</b> |
| 1. CREATION DU SERVICE .....   | 34        |
| 2. RECONFIGURATION DE NGINX .....  | 35        |
| 3. LANCEMENT DU SERVICE AU DEMARRAGE DE LINUX .....                                | 35        |
| <b>VII. EXPLICATIONS DES FICHIERS .....</b>  | <b>37</b> |
| 1. LE FICHIER DE CREATION DE SERVICE GUNICORN .....                                | 37        |
| a. <i>sudo nano /etc/systemd/system/gunicorn.service</i> .....                     | 37        |
| b. <i>Explications de ligne 9</i> .....  | 37        |
| 2. MODIFICATION DU FICHIER DE CONFIGURATION GNINX PROPRE A GUNICORN .....          | 38        |



|              |   |           |
|--------------|---|-----------|
| <b>VIII.</b> | <b>OBTENIR UNE ADRESSE IP PUBLIQUE FIXE .....</b> | <b>38</b> |
| 1.           | GRATUITE .....                                    | 38        |
| 2.           | MISE EN PLACE D'UNE IP ELASTIC .....              | 38        |
| 3.           | TEST .....  | 42        |



Source : <https://www.youtube.com/watch?v=-Gc8CMjQZfc>

### **Précédent essai :**

Une autre piste de déploiement : [https://docs.aws.amazon.com/fr\\_fr/elasticbeanstalk/latest/dg/create-deploy-python-flask.html](https://docs.aws.amazon.com/fr_fr/elasticbeanstalk/latest/dg/create-deploy-python-flask.html) , <https://openclassrooms.com/fr/courses/4425111-perfectionnez-vous-en-python/4463278-travaillez-dans-un-environnement-virtuel>, et <https://programwithus.com/learn-to-code/Pip-and-virtualenv-on-Windows/>

## I. Création d'une machine virtuelle linux visible sur internet

### 1. Compte aws

Cela nécessite d'avoir un compte aws (<https://aws.amazon.com/fr/resources/create-account/>).

Malheureusement il est impossible d'en créer un sans carte bleue.

Heureusement, il existe des offres :

- Toujours gratuites
- Gratuit pendant 12 mois
- Période d'essais

 Pour plus d'informations <https://aws.amazon.com/fr/free/free-tier/>

#### Types d'offres

Découvrez plus de 60 produits et commencez à créer sur AWS grâce à notre offre gratuite. Trois types d'offres gratuites sont disponibles en fonction du produit que vous utilisez. Consultez les éléments ci-dessous pour plus d'informations sur chaque produit.



#### Toujours gratuit

Ces offres gratuites n'expirent pas et sont disponibles pour tous les clients AWS



#### 12 mois gratuits

Profitez de ces offres pendant 12 mois après votre date d'inscription de départ sur AWS



#### Essais

Les offres d'essai gratuit à court terme débutent à la date d'activation d'un service en particulier.

 Vous pouvez trier les offres :



aws Contactez l'équipe commerciale Support Français Mon compte [Connectez-vous à la console](#)

Produits Solutions Tarification Documentation Apprendre Réseau de partenaires AWS Marketplace Déploiements clients Découvrir davantage

Offre gratuite d'AWS Présentation Questions fréquentes (FAQ) Conditions générales

## Informations relatives à l'offre gratuite

### Filtrer par :

Effacer tous les filtres

#### Type d'offre

- Références présentées
- 12 mois gratuits
- Toujours gratuit
- Essais

#### Catégories de produits

- Analyse
- Intégration d'applications
- AR ET VR
- Productivité d'entreprise
- Calcul
- Engagement client

Rechercher des produits de l'offre gratuite

#### CALCUL

Offre gratuite 12 MOIS GRATUITS

Amazon EC2  
**750 heures**

par mois

Capacité de calcul redimensionnable dans le cloud.

750 heures par mois d'utilisation d'instances

#### STOCKAGE

Offre gratuite 12 MOIS GRATUITS

Amazon S3  
**5 Go**

de stockage standard

Infrastructure de stockage d'objets sécurisée, durable et scalable.

5 Go de stockage standard

#### BASE DE DONNÉES

Offre gratuite 12 MOIS GRATUITS

Amazon RDS  
**750 heures**

par mois d'utilisation de base de données db.t2.micro (moteurs de base de données)

Service de base de données relationnelle géré pour MySQL, PostgreSQL, MariaDB, Oracle BYOL ou

## 2. Vérifier le coût

Pour vérifier le coût :

- Mettez en place des alertes
- Vérifiez vos dépassements

Pour cela allez dans votre console et recherchez *billing*

aws Services Groupes de ressources rtk Oregon Support

# AWS Management Console

### Services AWS

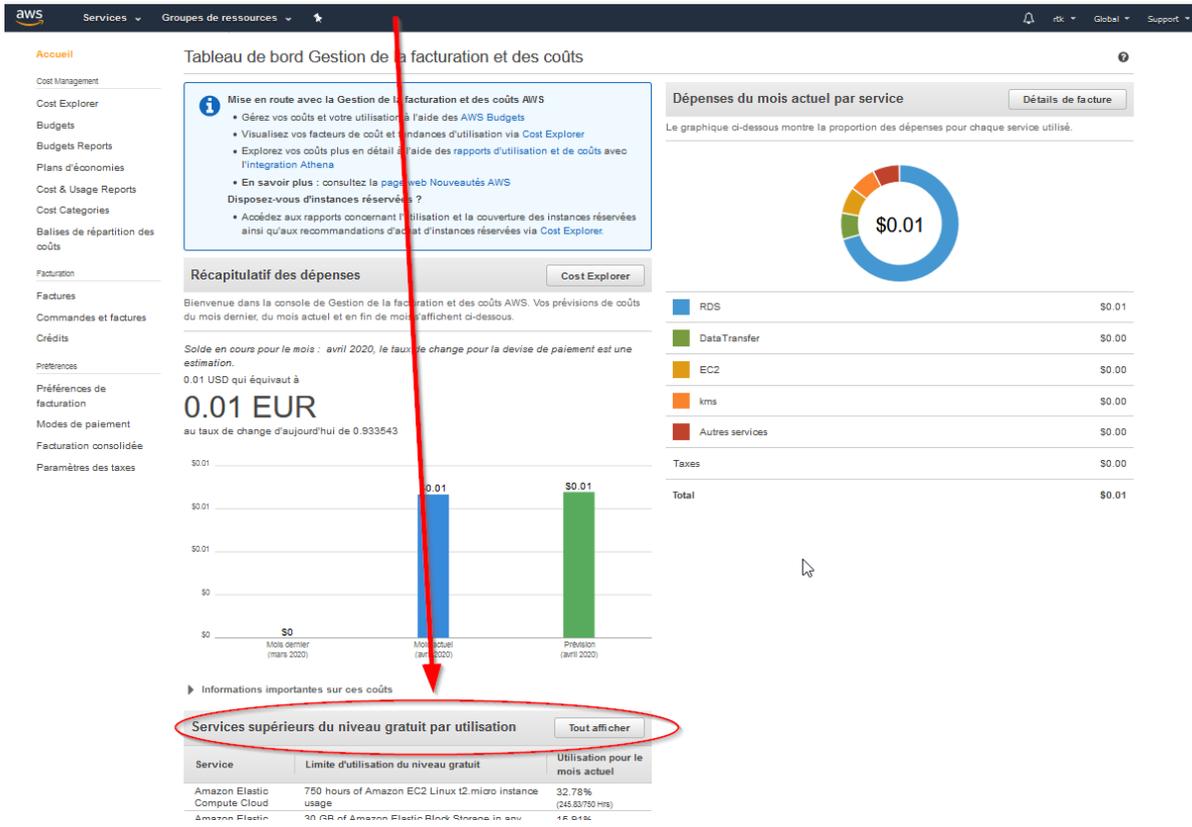
#### Trouvez des services

Vous pouvez entrer des noms, mots-clés ou acronymes.

### Restez connecté à vos ressources AWS pendant vos déplacements

Téléchargez l'application mobile de la console AWS sur votre appareil mobile iOS ou Android. [En savoir plus](#)

Puis descendez pour avoir un aperçu des services gratuits



### 3. Création d'une instance EC2

Dans cette partie, nous allons créer une machine virtuelle linux et plus précisément ubuntu-20. Pendant une année, vous pouvez utiliser EC2 gratuitement à raison d'un maximum de 750 heures.

**CALCUL**

---

Offre gratuite 12 MOIS GRATUITS

**Amazon EC2**

**750 heures**

par mois

Capacité de calcul redimensionnable dans le cloud.

750 heures par mois d'utilisation d'instances

Dans cette partie le service utilisé est *Amazon Elastic Compute Cloud* :

- Dans la limite de 750 heures de connexion par mois sur un linux de type micro, donc pas très puissant mais suffisant ici
- un disque dur/stockage de 30 Go au maximum



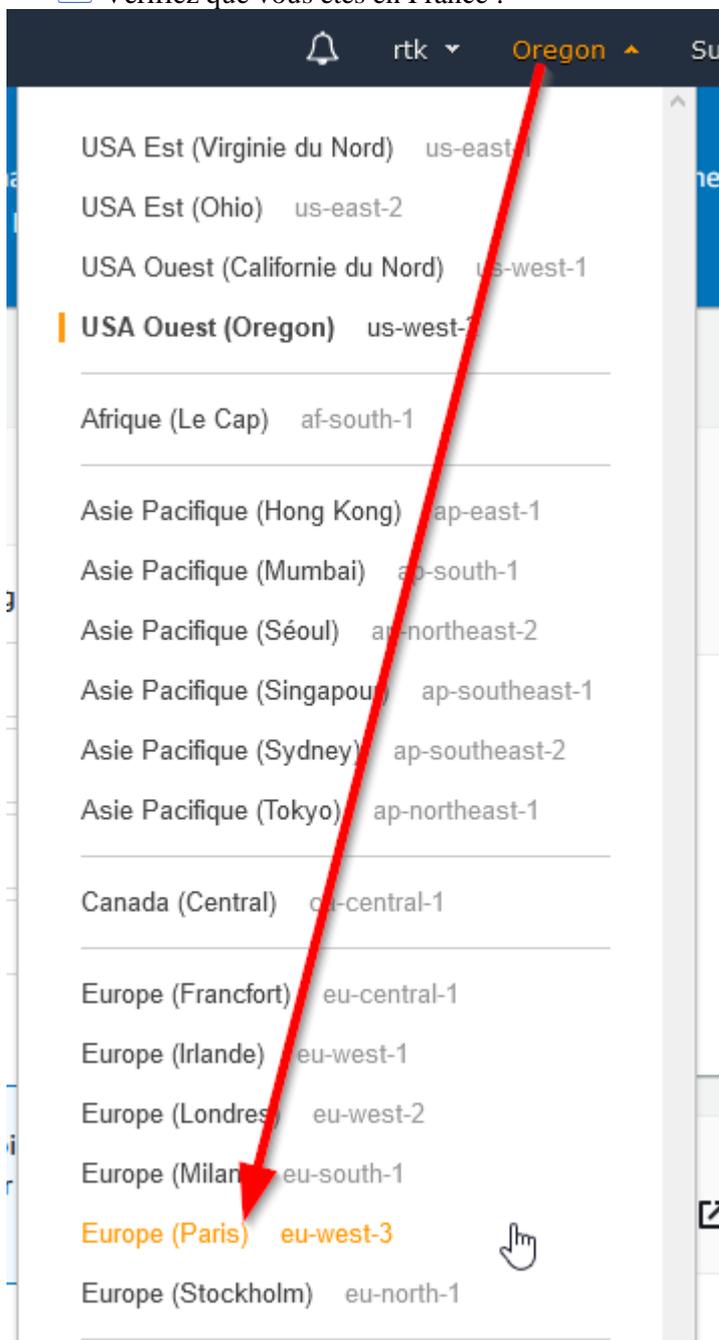
🔗 Connectez-vous sur la console d'AWS

## Connexion

### ● Utilisateur racine

Propriétaire du compte qui effectue des tâches requérant un accès illimité. [En savoir plus](#)

🔗 Vérifiez que vous êtes en France :





 Choisissez EC2 :

# AWS Management Console

## Services AWS

### Trouvez des services

Vous pouvez entrer des noms, mots-clés ou acronymes.

 Exemple : Service de base de données relationnelle, base de données, RDS

### ▼ Services récemment visités

 EC2

 RDS

 Billing

 Elastic Beanstalk

### ▼ Services AWS

 Calcul

EC2

Lightsail 

Lambda

 Machine Learning

Amazon SageMaker

Amazon CodeGuru

Amazon Comprehend

 Puis Lancer une instance



EC2

Ressources

Vous utilisez les ressources Amazon EC2 suivantes dans la région Europe (Paris) :

|                                |   |                        |   |
|--------------------------------|---|------------------------|---|
| Instances en cours d'exécution | 1 | Adresses IP Elastic    | 0 |
| Hôtes dédiés                   | 0 | Instantanés            | 0 |
| Volumes                        | 1 | Équilibreurs de charge | 0 |
| Paires de clés                 | 2 | Groupes de sécurité    | 4 |
| Groupes de placement           | 0 |                        |   |

Découvrez les nouveautés du calcul AWS d'AWS re:Invent en consultant les vidéos EC2.

Lancer une instance

Pour commencer, lancez une instance Amazon EC2 qui est un serveur virtuel dans le cloud.

Lancer une instance ▼

Remarque : vos instances seront lancées dans la région Europe (Paris).

Lancer une instance

Pour commencer, lancez une instance Amazon EC2 qui

Lancer une instance ▲

Lancer une instance

Lancer une instance à partir d'un modèle

région



AWS vous propose différentes images de systèmes d'exploitation.

Choisissez Ubuntu Server 20.xx en bas de la page

1. Choisir l'AMI 2. Choisir un type d'instance 3. Configurer l'instance 4. Ajouter le stockage 5. Ajouter des balises 6. Configurer le groupe de sécurité 7. Vérification

**Étape 1 : Sélection d'une Amazon Machine Image (AMI)** Annuler et quitter

Intel MKL-DNN, Docker & NVIDIA-Docker. For fully managed experience, check: <https://aws.amazon.com/sagemaker>

Type de périphérique racine: ebs Type de virtualisation: hvm ENA activée: Oui

- Deep Learning Base AMI (Ubuntu 18.04) Version 22.0** - ami-05b04007a78e69a1e Sélectionner  
Comes with foundational platform of NVidia CUDA, cuDNN, NCCL, GPU Drivers, Intel MKL-DNN and other system libraries to deploy your own custom deep learning environment. For a fully managed experience, check: <https://aws.amazon.com/sagemaker>  
Type de périphérique racine: ebs Type de virtualisation: hvm ENA activée: Oui 64 bits (x86)
- Ubuntu Server 16.04 LTS (HVM) with SQL Server 2017 Standard** - ami-0447a1f30868b20d Sélectionner  
Microsoft SQL Server 2017 Standard edition on Ubuntu Server 16.04 LTS. The AMI also comes pre-installed with .NET Core 2.0 and PowerShell 6.0.  
Type de périphérique racine: ebs Type de virtualisation: hvm ENA activée: Oui 64 bits (x86)
- Ubuntu Server 20.04 LTS (HVM), SSD Volume Type** - ami-0e60c6afa19d896ee Sélectionner  
Ubuntu Server 20.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).  
Éligible à l'offre 64 bits (x86)
- Deep Learning Base AMI (Ubuntu 16.04) Version 22.0** - ami-0129e7ad1b1a6329c Sélectionner  
Comes with foundational platform of NVidia CUDA, cuDNN, NCCL, GPU Drivers, Intel MKL-DNN and other system libraries to deploy your own custom deep learning environment. For a fully managed experience, check: <https://aws.amazon.com/sagemaker>  
Type de périphérique racine: ebs Type de virtualisation: hvm ENA activée: Oui 64 bits (x86)

Laissez le choix par défaut

Appuyez sur Suivant :

## Étape 2 : Choisir un type d'instance

Amazon EC2 fournit un vaste éventail de types d'instances optimisés pour différents cas d'utilisation. Les instances sont des serveurs virtuels qui peuvent exécuter des applications. Les types d'instances se composent de différentes combinaisons de processeur, de mémoire, de stockage et de capacité réseau, et vous offrent une flexibilité dans le choix de l'association de ressources adaptées à vos applications. En savoir plus à propos des types d'instances et de la manière dont ils peuvent répondre à vos besoins informatiques.

Filtrer par: Tous les types d'instances Génération actuelle Afficher / Masquer les colonnes

Actuellement sélectionné : t2.micro (Variable ECU, 1 vCPU, 2.5 GHz, Intel Xeon Family, 1 Gio mémoire, EBS uniquement)

|                                     | Famille       | Type   | vCPU | Mémoire (Gio) | Stockage d'instance (Go) | Disponible en version optimisée pour EBS | Performances réseau | Prise en charge IPv6 |
|-------------------------------------|---------------|--|------|---------------|--------------------------|--|---------------------|----------------------|
| <input type="checkbox"/>            | Usage général | t2.nano  | 1    | 0.5           | EBS uniquement           | -  | Faibles à modérées  | Oui                  |
| <input checked="" type="checkbox"/> | Usage général | t2.micro<br><small>Éligible à l'offre gratuite</small> | 1    | 1             | EBS uniquement           | -  | Faibles à modérées  | Oui                  |
| <input type="checkbox"/>            | Usage général | t2.small   | 1    | 2             | EBS uniquement           | -  | Faibles à modérées  | Oui                  |
| <input type="checkbox"/>            | Usage général | t2.medium  | 2    | 4             | EBS uniquement           | -  | Faibles à modérées  | Oui                  |
| <input type="checkbox"/>            | Usage général | t2.large   | 2    | 8             | EBS uniquement           | -  | Faibles à modérées  | Oui                  |
| <input type="checkbox"/>            | Usage général | t2.xlarge  | 4    | 16            | EBS uniquement           | -  | Modérées            | Oui                  |
| <input type="checkbox"/>            | Usage général | t2.2xlarge   | 8    | 32            | EBS uniquement           | -  | Modérées            | Oui                  |
| <input type="checkbox"/>            | Usage général | t3a.nano   | 2    | 0.5           | EBS uniquement           | Oui                                      | Jusqu'à 5 gigabits  | Oui                  |
| <input type="checkbox"/>            | Usage général | t3a.micro  | 2    | 1             | EBS uniquement           | Oui                                      | Jusqu'à 5 gigabits  | Oui                  |

Annuler Précédent Vérifier et lancer Suivant : Configurer les détails de l'instance



 Laissez par défaut cette autre page

### Étape 3 : Configurer les détails de l'instance

Configurez l'instance en fonction de vos besoins. Vous pouvez lancer plusieurs instances à partir de la même AMI, demander des instances Spot pour bénéficier un tarif inférieur, attribuer un rôle de gestion d'accès à l'instance et bien d'autres choses encore.

Nombre d'instances  [Lancer dans le groupe Auto Scaling](#)

Option d'achat  Demander des instances Spot

Réseau  [Créer un nouveau VPC](#)

Sous-réseau  [Créer un nouveau sous-réseau \(subnet\)](#)

Attribuer automatiquement l'adresse IP publique

Groupe de placement  Ajoutez une instance au groupe de placement.

Réserve de capacité  [Créer une nouvelle réserve de capacité](#)

Rôle IAM  [Créer un nouveau rôle IAM](#)

Comportement d'arrêt

Arrêt - Activer le comportement de veille prolongée  Activer la mise en veille prolongée comme comportement d'arrêt supplémentaire

Activer la protection de la résiliation  Protéger contre la résiliation accidentelle

[Annuler](#) [Précédent](#) [Vérifier et lancer](#) [Suivant : Ajouter le stockage](#)

 Mettez le maximum de taille du disque (c'est gratuit jusque 30 GB)

### Étape 4 : Ajouter le stockage

Votre instance sera lancée avec les paramètres suivants du périphérique de stockage. Vous pouvez attacher des volumes EBS supplémentaires et des volumes de stockage d'instance à votre instance ou modifier les réglages du volume racine. Vous pouvez également attacher des volumes EBS supplémentaires après le lancement d'une instance, mais pas des volumes de stockage d'instance. [En savoir plus](#) sur les options de stockage dans Amazon EC2.

| Type de volume | Dispositif | Instantané             | Taille (Gio)                    | Type de volume                   | IOPS       | Débit (Mbit/s) | Supprimer à la résiliation          | Chiffrement |
|----------------|------------|------------------------|---------------------------------|----------------------------------|------------|----------------|-------------------------------------|-------------|
| Racine         | /dev/sda1  | snap-08ee4dd4dc8cc5a1e | <input type="text" value="30"/> | Volume à usage général SSD (gp2) | 100 / 3000 | N/A            | <input checked="" type="checkbox"/> | Non chiffré |

[Ajouter un nouveau volume](#)

Les clients éligibles à l'offre gratuite peuvent obtenir jusqu'à 30 Go de stockage EBS à usage général (SSD) ou magnétique. [En savoir plus](#) à propos de l'éligibilité pour le niveau d'offre gratuite et des restrictions d'utilisation.

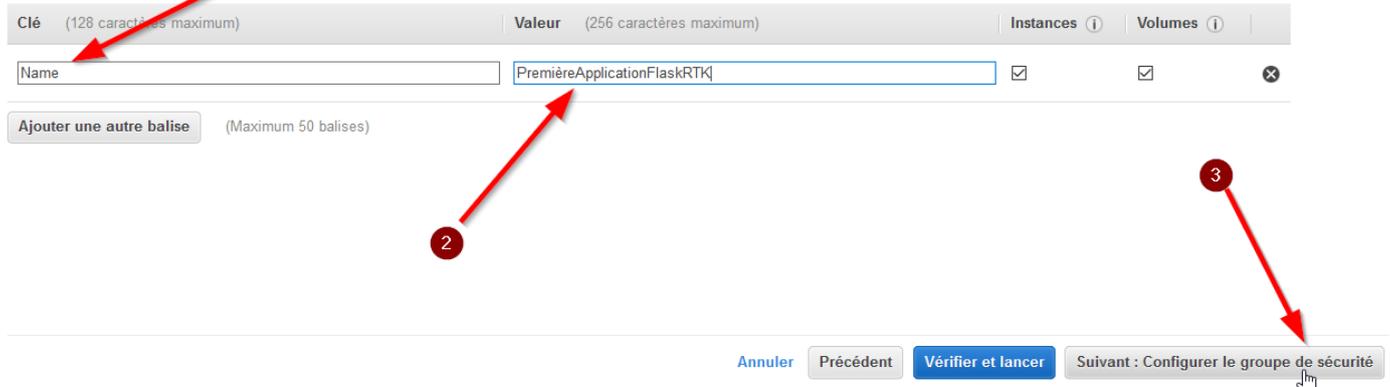
[Annuler](#) [Précédent](#) [Vérifier et lancer](#) [Suivant : Ajouter des balises](#)

 Ajoutez comme *Balise* (1) Name et comme valeur (2) PremièreApplicationFlaskRTK puis (3) Next..



## Étape 5 : Ajouter des balises

Une balise est composée d'une paire de clé-valeur. La clé est sensible à la casse. Par exemple, vous pourriez définir une balise avec clé = Nom et valeur = Serveur. Une copie de balise peut être appliquée aux volumes, aux instances ou aux deux. Les balises seront appliquées à toutes les instances et à tous les volumes. [En savoir plus](#) à propos du balisage de vos ressources Amazon EC2.

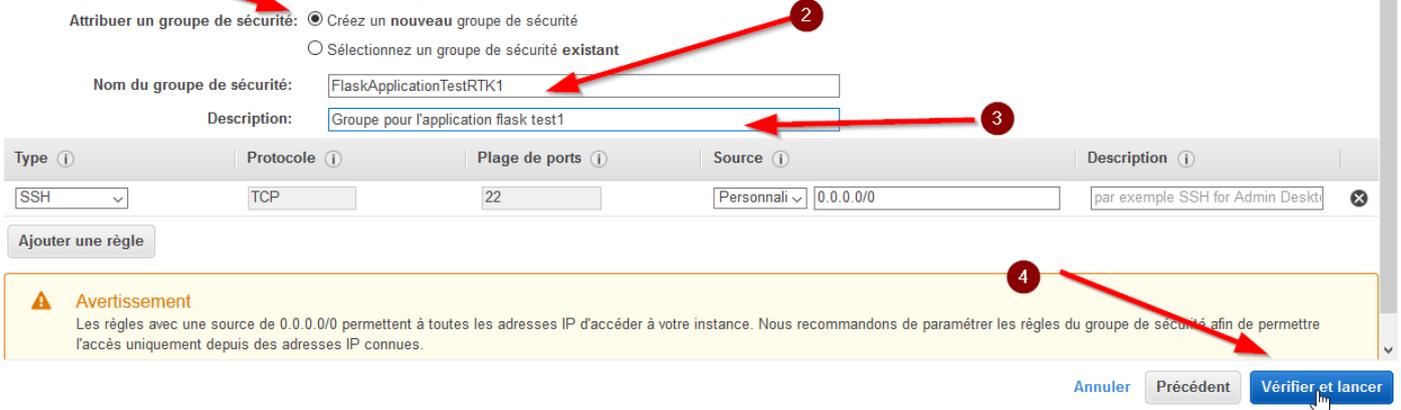


### Attention phase très importante :

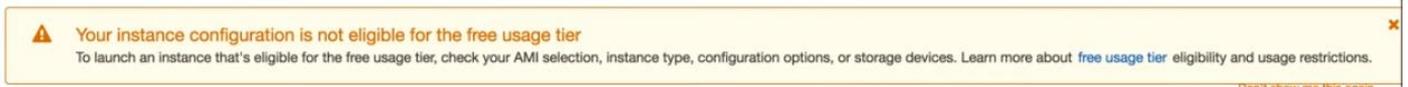
- (1) créez un nouveau groupe de sécurité
- (2) donnez lui un nom « *FlaskApplicationTestRTK1* »
- (3) une description « *Groupe pour application flask test1* » (pas 'comme dans l'exemple')
- (4) cliquez sur Vérifier...

## Étape 6 : Configurer le groupe de sécurité

Un groupe de sécurité est un ensemble de règles de pare-feu qui contrôlent le trafic de votre instance. Sur cette page, vous pouvez ajouter des règles pour permettre qu'un trafic spécifique atteigne votre instance. Par exemple, si vous voulez configurer un serveur Web et permettre au trafic Internet d'atteindre votre instance, ajoutez des règles qui autorisent un accès restreint aux ports HTTP et HTTPS. Vous pouvez créer un nouveau groupe de sécurité ou en sélectionner un parmi les groupes existants ci-dessous. [En savoir plus](#) à propos des groupes de sécurité Amazon EC2.



Vous avez un résumé de l'instance, vérifiez que tout est éligible aux services offerts par aws. Si vous avez ce message, retournez en arrière et modifiez votre instance :

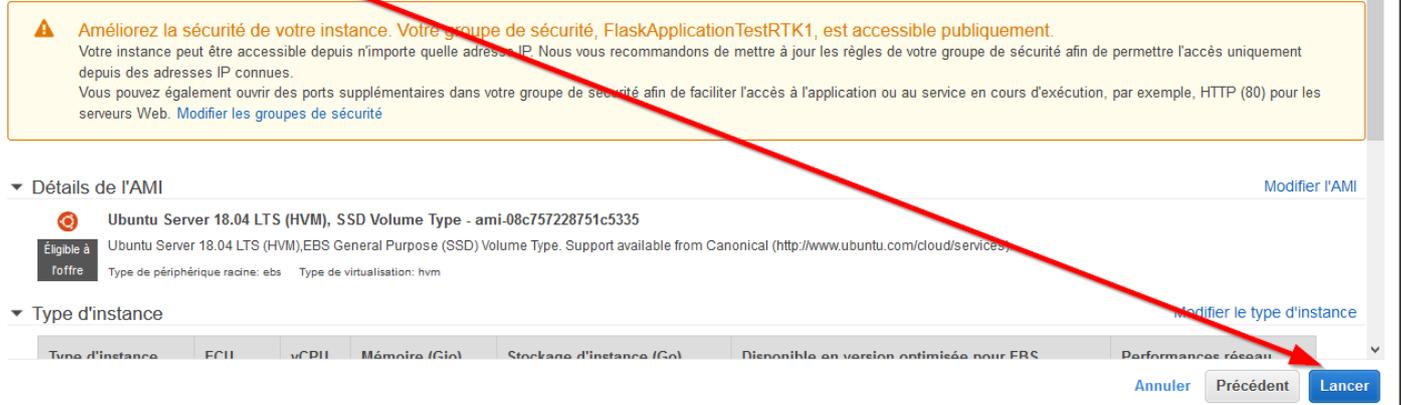


 Lancer



## Étape 7 : Examinez le lancement de l'instance

Veuillez vérifier les détails de votre lancement d'instance. Vous pouvez revenir en arrière pour modifier les changements pour chaque section. Cliquez sur **Lancer** pour affecter une paire de clés à votre instance et terminer la procédure de lancement.



**⚠ Améliorez la sécurité de votre instance. Votre groupe de sécurité, FlaskApplicationTestRTK1, est accessible publiquement.**  
Votre instance peut être accessible depuis n'importe quelle adresse IP. Nous vous recommandons de mettre à jour les règles de votre groupe de sécurité afin de permettre l'accès uniquement depuis des adresses IP connues.  
Vous pouvez également ouvrir des ports supplémentaires dans votre groupe de sécurité afin de faciliter l'accès à l'application ou au service en cours d'exécution, par exemple, HTTP (80) pour les serveurs Web. [Modifier les groupes de sécurité](#)

▼ Détails de l'AMI Modifier l'AMI

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-08c757228751c5335

Éligible à Ubuntu Server 18.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Offre Type de périphérique racine: ebs Type de virtualisation: hvm

▼ Type d'instance Modifier le type d'instance

| Type d'instance | ECU | vCPU | Mémoire (GiB) | Stockage d'instance (Go) | Disponible en version optimisée pour EBS | Performances réseau |
|-----------------|-----|------|---------------|--------------------------|--|---------------------|
|                 |     |      |               |                          |  |                     |

[Annuler](#) [Précédent](#) [Lancer](#)

Ensuite :

Une paire de clés se compose d'une clé publique stockée par AWS et d'un fichier de clé privée que vous stockez. Ensemble, ils vous permettent de vous connecter à votre instance en toute sécurité. Pour les AMI Windows, le fichier de clé privée est requis pour obtenir le mot de passe utilisé pour se connecter à votre instance. Pour les AMI Linux, le fichier de clé privée vous permet de SSH en toute sécurité dans votre instance

### Nous allons générer une nouvelle paire de clé

-  (1) modifiez pour avoir *Create a new pair*
-  (2) nommez-le : *FlaskRTK01*
-  (3) téléchargez le fichier de clef privée et sauvegardez-le sur votre disque (dans mon exemple il sera dans « *D:\Documents\Down* »)
-  (4) lancez l'instance



## Sélectionnez une paire de clés existante ou créez une nouvelle paire de clés

Une paire de clés se compose d'une **clé publique** conservée par AWS et d'un **fichier de clé privée** que vous conservez. Ensemble, elles vous permettent de vous connecter à votre instance en toute sécurité. Pour les AMI Windows, le fichier de clé privée est requis pour obtenir le mot de passe utilisé pour se connecter à votre instance. Pour les AMI Linux, le fichier de clé privée vous permet d'accéder en toute sécurité à votre instance via SSH.

Remarque : La paire de clés sélectionnée sera ajoutée à l'ensemble de clés autorisé pour cette instance. En savoir plus sur [la suppression de paires de clés existantes d'une AMI publique](#).

- 1 Créer une nouvelle paire de clés  
Nom de la paire de clés
- 2 FlaskRTK01
- 3 Télécharger une paire de clés
- 4 Lancer des instances

Vous devez télécharger le **fichier de clé privée** (fichier \*.pem) avant de pouvoir continuer. **Stockez-le dans un endroit sûr et accessible.** Vous ne pourrez pas le télécharger à nouveau après sa création.

### ATTENTION :

Gardez en lieu sûr ce fichier car il permet d'accéder à votre machine virtuelle, votre instance, sans mot de passe. Il est impossible de le télécharger à nouveau.

# Installer une application flask sur AWS



ation flask test1

## Sélectionnez une paire de clés

Une paire de clés se compose d'une clé privée que vous conservez. Ensemble, elle assure toute sécurité. Pour les AMI Windows, le fichier de clé privée est utilisé pour se connecter à votre instance et accéder en toute sécurité à votre instance.

Remarque : La paire de clés sélectionnée sera ajoutée à l'ensemble de clés autorisées pour cette instance. En savoir plus sur [la suppression de paires de clés existantes d'une AMI publique](#).

Créer une nouvelle paire de clés

Nom de la paire de clés  
FlaskRTK01

Télécharger une paire de clés

Vous devez télécharger le fichier de clé privée (fichier \*.pem) avant de pouvoir continuer. Stockez-le dans un endroit sûr et accessible. Vous ne pourrez pas le télécharger à nouveau après sa création.

Annuler Lancer des instances

Ouverture de FlaskRTK01.pem

Vous avez choisi d'ouvrir :

FlaskRTK01.pem  
qui est un fichier de type : Text Document  
à partir de : https://eu-west-3.console.aws.amazon.com

Que doit faire Firefox avec ce fichier ?

Ouvrir avec Bloc-notes (par défaut)

Enregistrer le fichier

Toujours effectuer cette action pour ce type de fichier.

OK Annuler

Il vous est proposé d'être informés des frais : n'hésitez pas à mettre des alertes

✓ **Votre instance est en cours de lancement**  
Le lancement d'instance suivant a été initié : i-08e888c60e552b3b4 [Afficher le journal de lancement](#)

i **Être informé des frais estimés**  
[Créer des alertes de facturation](#) pour obtenir une notification par e-mail lorsque les frais estimés imputés sur votre facture AWS dépassent un montant que vous définissez (par exemple, si vous dépassez le niveau d'offre gratuite).



## Préférences

### Préférences de facturation

**Facture PDF par e-mail**

Activez cette fonction pour recevoir une version PDF de votre facture par e-mail. Les factures sont généralement disponibles au cours des trois premiers jours du mois.

### Préférences de gestion des coûts

**Recevoir les alertes d'utilisation de l'offre gratuite**

Activez cette fonctionnalité pour recevoir des alertes par e-mail lorsque votre utilisation d'un service AWS approche ou a dépassé les limites d'utilisation de l'offre gratuite AWS. Pour recevoir ces alertes à une adresse e-mail qui n'est pas l'adresse e-mail principale associée à ce compte, spécifiez ci-dessous cette adresse e-mail.

Adresse e-mail :

**Recevoir les alertes de facturation**

Activez cette fonction pour surveiller automatiquement vos coûts d'utilisation des services AWS et frais récurrents afin de simplifier le suivi et la gestion de vos dépenses sur AWS. Vous pouvez configurer des alertes de facturation pour recevoir des notifications par e-mail lorsque vos coûts atteignent un seuil défini. Une fois activée, cette préférence ne peut plus être désactivée. [Gérer les alertes de facturation](#) ou [testez la nouvelle fonctionnalité de gestion de budgets!](#)

► [Rapports de facturation détaillée \[ancien\]](#)

[Enregistrer les préférences](#)

 Vérifiez l'état de l'instance :

### Statut de lancement

 **Votre instance est en cours de lancement**  
Le lancement d'instance suivant a été initié : `i-08e888c60e552b3b4` [Afficher le journal de lancement](#)

 **Être informé des frais estimés**  
[Créer des alertes de facturation](#) pour obtenir une notification par e-mail lorsque les frais estimés imputés sur votre facture AWS dépassent un montant que vous définissez (par exemple, si vous dépassez le niveau d'offre gratuite).

### Comment vous connecter à votre instance

Votre instance est en cours de lancement et quelques minutes pourraient être nécessaires avant qu'elle ne soit en état d'**exécution**, lorsqu'elle sera prête à être utilisée. Les heures d'utilisation de votre nouvelle instance commenceront immédiatement et continueront d'augmenter jusqu'à ce que vous arrêtez votre instance ou vous la mettiez hors service.

Cliquez sur **View instances** (Afficher les instances) pour contrôler le statut de votre instance. Une fois que votre instance est en **cours d'exécution**, vous pouvez vous y **connecter** depuis l'écran Instances. [Découvrez comment vous connecter à votre instance.](#)

### Voici quelques ressources utiles pour débuter

- [Comment vous connecter à votre instance Linux](#)
- [Amazon EC2 : Guide de l'utilisateur](#)
- [En savoir plus sur le niveau d'offre gratuite d'AWS](#)
- [Amazon EC2 : Forum de discussion](#)

Pendant le lancement de vos instances, vous pouvez également

- [Créer des alarmes de contrôle de statut](#) pour être informé des échecs des contrôles de statut de ces instances. (des frais supplémentaires peuvent être facturés)
- [Créer et attacher des volumes EBS supplémentaires](#) (des frais supplémentaires peuvent être facturés)
- [Gérer les groupes de sécurité](#)

[Afficher les instances](#)



Launch Instance ▼ Connect Actions ▼

Filter by tags and attributes or search by keyword

| Name           | Instance ID         | Instance Type | Availability Zone | Instance State | Status Checks  | Alarm Status | Public DNS   |
|----------------|---------------------|---------------|-------------------|----------------|----------------|--------------|--------------|
| PremièreApp... | i-0f868df1e6204f389 | t2.micro      | eu-west-3c        | running        | 2/2 checks ... | None         | ec2-52-47-17 |

## II. Accès à distance

Dans cette partie nous allons accéder à distance à notre machine virtuelle linux à l'aide de ssh et du fichier de clé.

### 1. Commande à exécuter

- ☞ Sélectionnez
- ☞ (1) votre instance
- ☞ (2) puis *Connect*

Launch Instance ▼ Connect Actions ▼

Filter by tags and attributes or search by keyword

| Name           | Instance ID         | Instance Type | Availability Zone | Instance State | Status Checks  | Alarm Status | Public DNS   |
|----------------|---------------------|---------------|-------------------|----------------|----------------|--------------|--------------|
| PremièreApp... | i-0f868df1e6204f389 | t2.micro      | eu-west-3c        | running        | 2/2 checks ... | None         | ec2-52-47-17 |

Il est proposé trois méthodes de connexion :

### Connectez-vous à votre instance

- Méthode de connexion
- Un client SSH autonome 1
  - Session Manager 2
  - Connexion d'instance EC2 (connexion SSH basée sur un navigateur) 3

1. Avec un client ssh
2. Avec une console EC2
3. Dans un navigateur

Les méthodes 1 et 3 sont décrites dans ce document : soit dans un navigateur soit avec un client ssh.

### 2. La plus simple : dans un navigateur

Bien plus simple à mettre en œuvre, malheureusement le copier/coller ne fonctionne pas très bien et il y a quelques « plantages » assez désagréable. A n'utiliser qu'en dépannage.

- ☞ Laissez le login par défaut



### Connectez-vous à votre instance

Méthode de connexion

Un client SSH autonome ⓘ

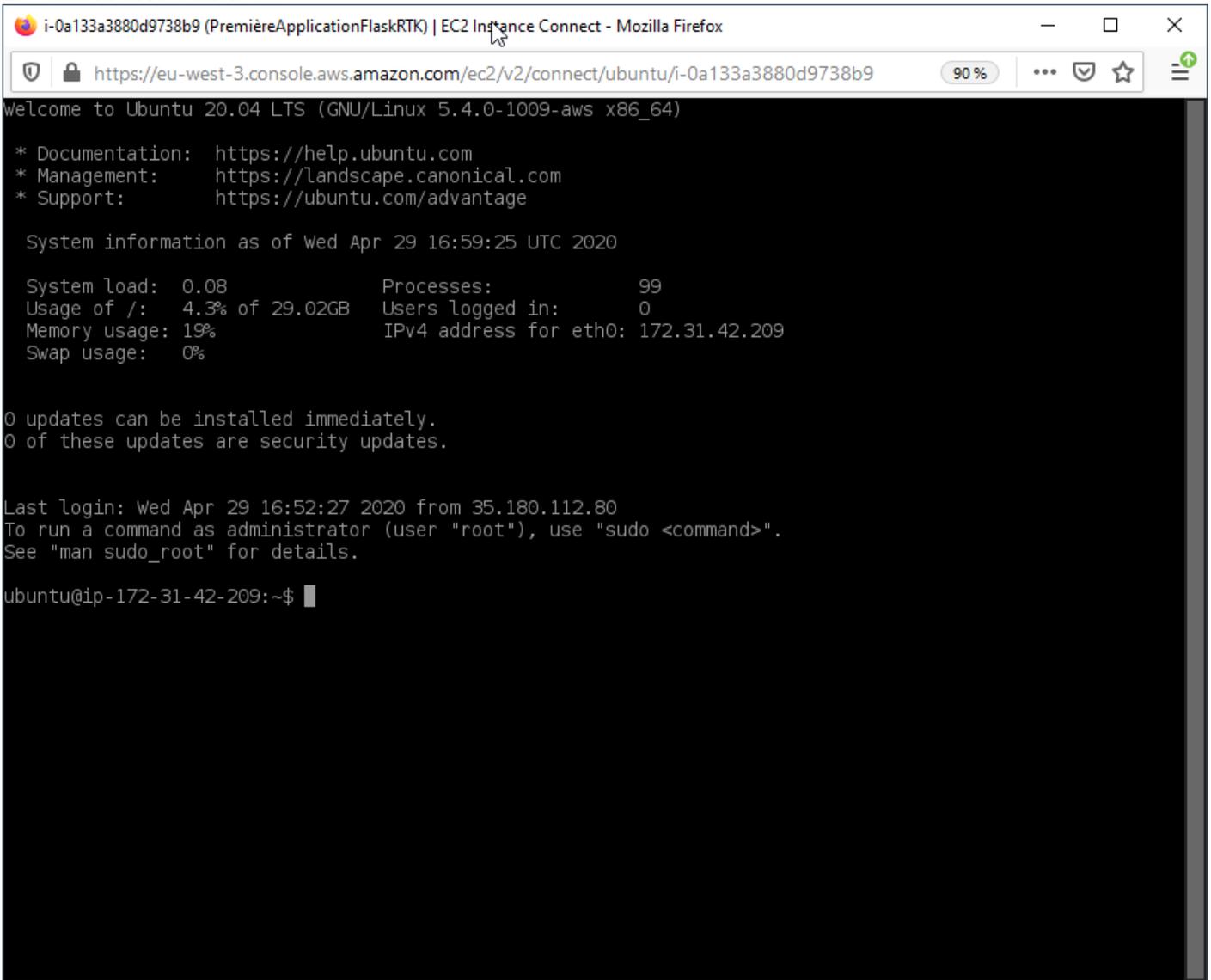
Session Manager ⓘ

Connexion d'instance EC2 (connexion SSH basée sur un navigateur) ⓘ

Connectez-vous à l'aide d'un nom d'utilisateur personnalisé ou indiquez le nom d'utilisateur par défaut pour l'AMI utilisée pour lancer l'instance. [En savoir plus](#)

Nom utilisateur  ⓘ

La connexion ressemble à :



```
i-0a133a3880d9738b9 (PremièreApplicationFlaskRTK) | EC2 Instance Connect - Mozilla Firefox
https://eu-west-3.console.aws.amazon.com/ec2/v2/connect/ubuntu/i-0a133a3880d9738b9
welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-1009-aws x86_64)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

System information as of Wed Apr 29 16:59:25 UTC 2020

System load: 0.08          Processes:                99
Usage of /:  4.3% of 29.02GB Users logged in:          0
Memory usage: 19%         IPv4 address for eth0: 172.31.42.209
Swap usage:  0%

0 updates can be installed immediately.
0 of these updates are security updates.

Last login: Wed Apr 29 16:52:27 2020 from 35.180.112.80
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-42-209:~$
```

i-0a133a3880d9738b9 (PremièreApplicationFlaskRTK)  
Public IPs: 15.188.119.241 Private IPs: 172.31.42.209



Et c'est tout...

**Vous pouvez faire du copier/coller (CTRL V)**

 Allez directement à [Fin de la configuration Linux](#)

### 3. Le plus performant : utilisation d'un client ssh

Et la plus fiable sans discussion.

#### a. Un exemple de commande est donné :

### Connect to your instance ✕

**Connection method**

A standalone SSH client 

Session Manager 

EC2 Instance Connect (browser-based SSH connection) 

---

**To access your instance:**

1. Open an SSH client. (find out how to [connect using PuTTY](#))
2. Locate your private key file (FlaskRTK01.pem). The wizard automatically detects the key you used to launch the instance.
3. Your key must not be publicly viewable for SSH to work. Use this command if needed:  

```
chmod 400 FlaskRTK01.pem
```
4. Connect to your instance using its Public DNS:  

```
ec2-52-47-171-193.eu-west-3.compute.amazonaws.com
```

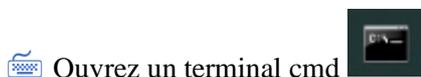
**Example:**

```
ssh -i "FlaskRTK01.pem" ubuntu@ec2-52-47-171-193.eu-west-3.compute.amazonaws.com
```

Please note that in most cases the username above will be correct, however please ensure that you read your AMI usage instructions to ensure that the AMI owner has not changed the default AMI username.

If you need any assistance connecting to your instance, please see our [connection documentation](#).

[Close](#)

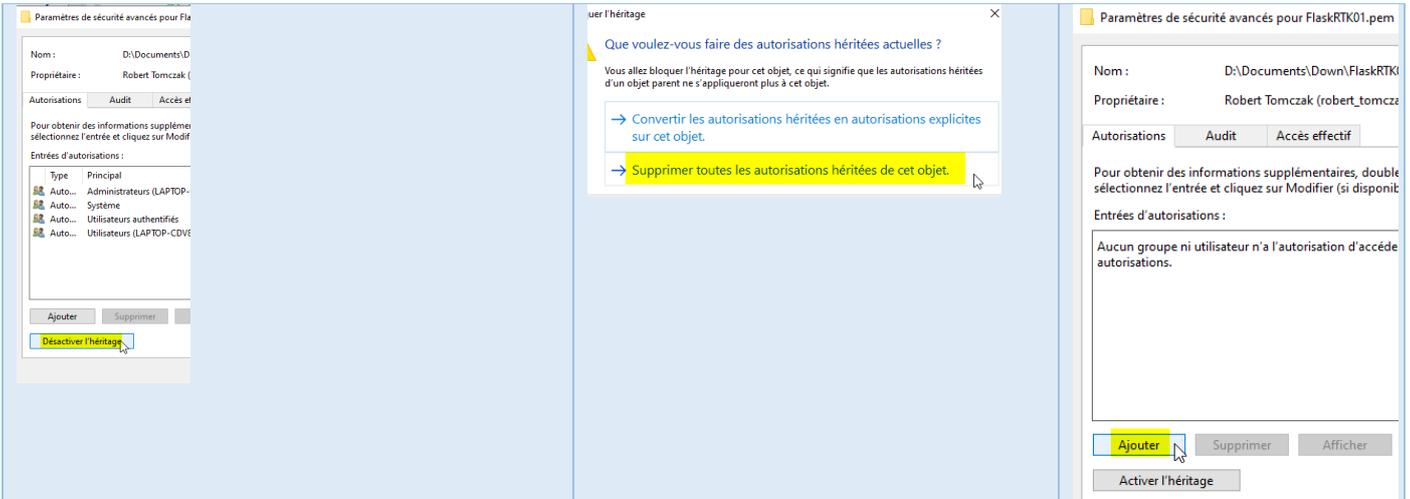


Ouvrez un terminal cmd (Clic sur " Démarrer " > Tous les programmes > Accessoires > Invite de commandes)

 Localisez votre fichier téléchargé précédemment (dans mon exemple D:\Documents\Down)

 Allez dans le répertoire dans lequel se trouve ce fichier par `cd /D D:\Documents\Down`

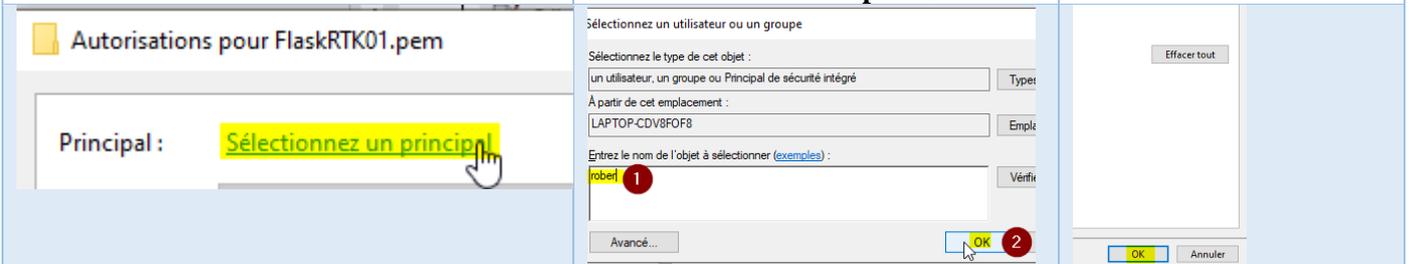


7 : Sélectionner un ...

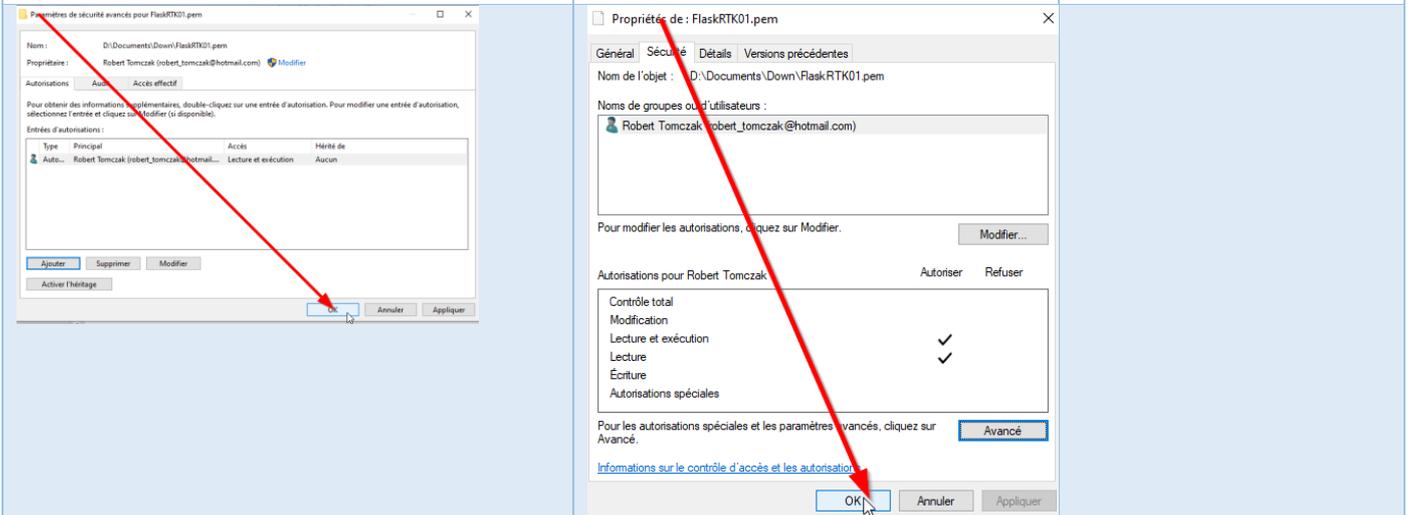
8 : mettez votre identifiant/login de connexion à windows puis Ok

9 : Ok



10 : Ok

11 : Ok



## c. Accès à distance par ssh

Après avoir vérifié que vous soyez dans le bon répertoire (`cd /D D:\Documents\Down` dans l'exemple)

Tapez `ssh -i "FlaskRTK01.pem" ubuntu@ec2-52-47-171-193.eu-west-3.compute.amazonaws.com`

### Rappels :

1. "FlaskRTK01.pem" est le fichier contenant la clé (c.f. [#FichierCle](#))
2. `ubuntu@ec2-52-47-171-193.eu-west-3.compute.amazonaws.com` est l'adresse de votre instance (c.f. [#Adresse\\_Instance](#))

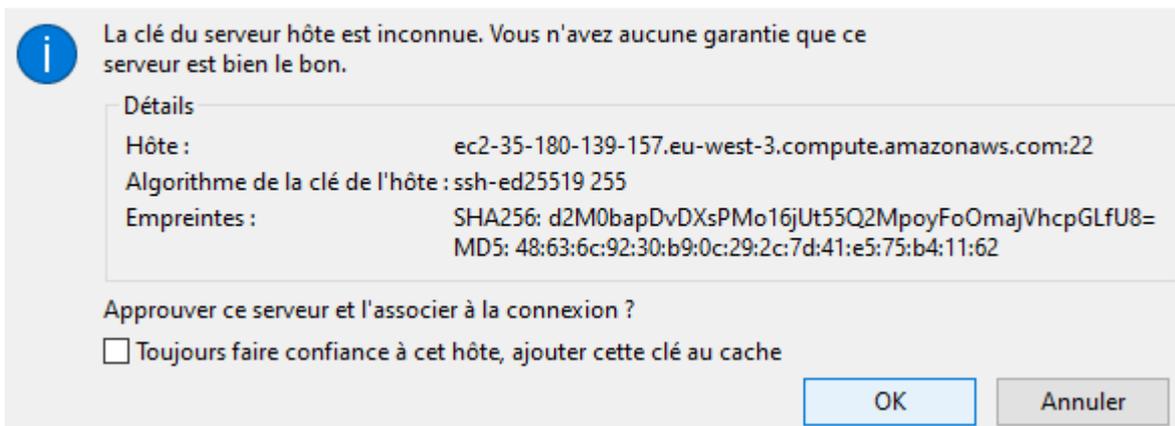
Puis vous êtes connecté à votre machine virtuelle :



```
ubuntu@ip-172-31-34-163: ~  
This private key will be ignored.  
Load key "FlaskRTK01.pem": bad permissions  
ubuntu@ec2-52-47-171-193.eu-west-3.compute.amazonaws.com: Permission denied (publickey).  
D:\Documents\Down>ssh -i "FlaskRTK01.pem" ubuntu@ec2-52-47-171-193.eu-west-3.compute.amazonaws.com  
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-1009-aws x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
  
System information as of Sun Apr 26 11:11:29 UTC 2020  
  
System load:  0.0          Processes:           99  
Usage of /:   4.8% of 29.02GB Users logged in:    0  
Memory usage: 19%        IPv4 address for eth0: 172.31.34.163  
Swap usage:   0%  
  
* Ubuntu 20.04 LTS is out, raising the bar on performance, security,  
and optimisation for Intel, AMD, Nvidia, ARM64 and Z15 as well as  
AWS, Azure and Google Cloud.  
  
https://ubuntu.com/blog/ubuntu-20-04-lts-arrives  
  
3 updates can be installed immediately.  
0 of these updates are security updates.  
To see these additional updates run: apt list --upgradable  
  
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings  
  
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-172-31-34-163:~$
```

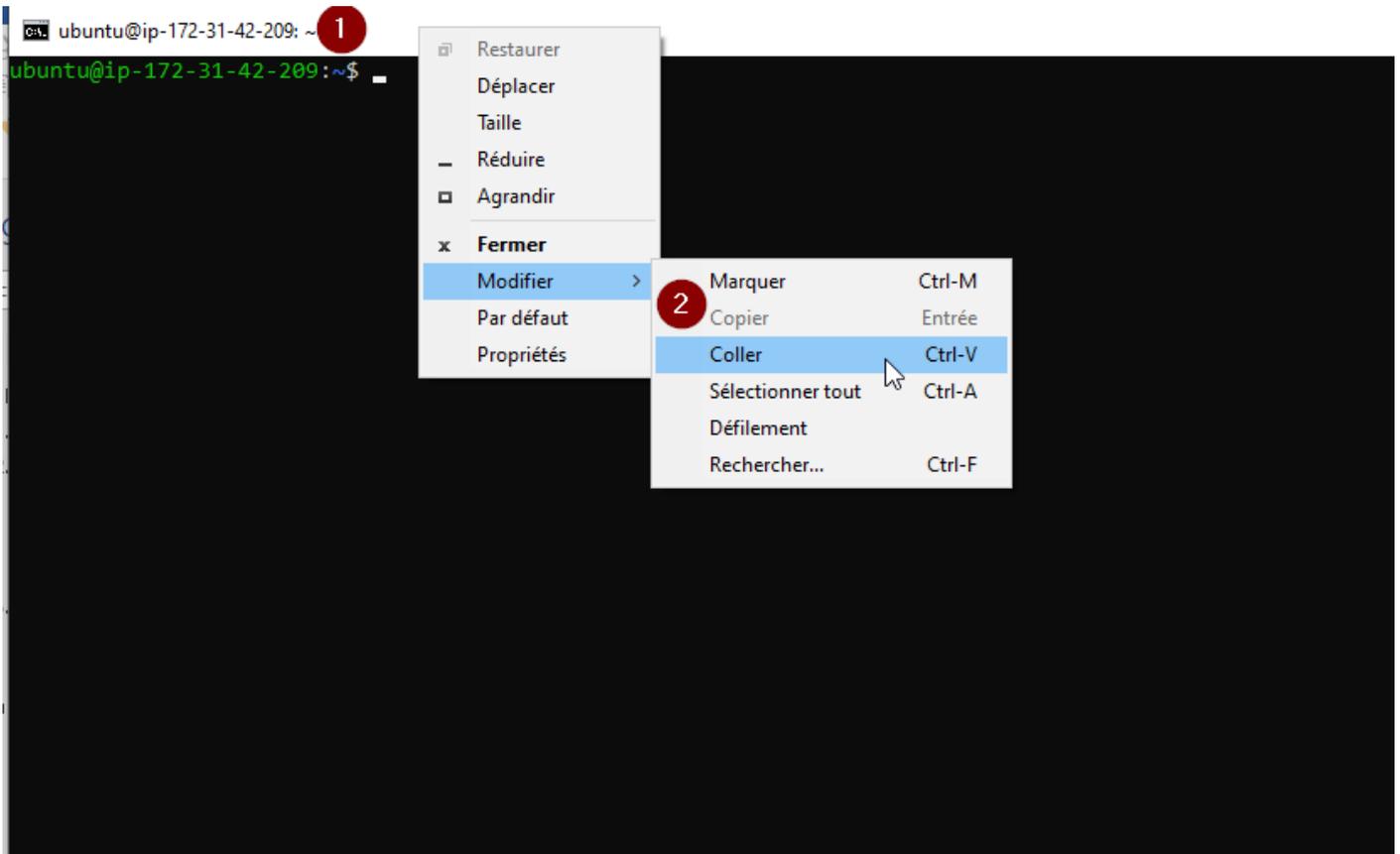
Remarque : vous pouvez avoir ce message, dans ce cas, acceptez (Ok)

Clé de l'hôte inconnue



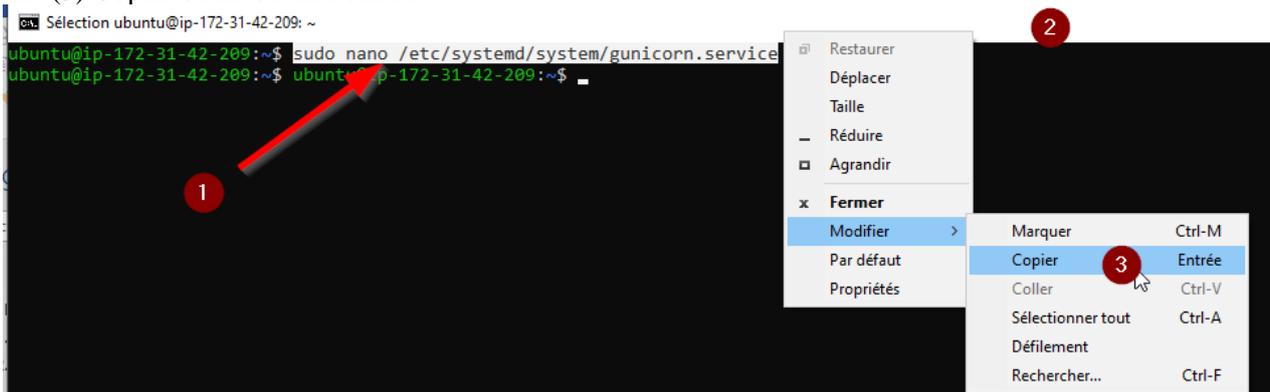
#### d. Copier/coller dans le terminal

Un clic droit sur la barre du haut de cmd (1) vous fait apparaître un menu permettant de faire du copier/coller :



Par exemple pour copier du terminal vers un document :

-  (1) Sélectionnez le texte avec la souris
-  (2) clic droit sur la barre
-  (3) Copier ou la touche entrée



Ou alors

-  Sélectionnez
-  Directement la touche entrée

## III. Fin de la configuration Linux

### 1. Préparation et mise à jour de la distribution

-  Mettez à jour les dépôts apt



sudo apt-get update

```
ubuntu@ip-172-31-34-163:~$ sudo apt-get update
Hit:1 http://eu-west-3.ec2.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://eu-west-3.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [97.0 kB]
Hit:3 http://eu-west-3.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [101 kB]
```

🔗 Vérifiez la version de python

python3 -V

```
ubuntu@ip-172-31-34-163:~$ python3 -V
Python 3.8.2
```

Il faut avoir au moins la version 3.6

🔗 Vérifiez l'installation de pip3 (gestionnaire de paquets)

pip3

```
ubuntu@ip-172-31-34-163:~$ pip3
Command 'pip3' not found, but can be installed with:
sudo apt install python3-pip
```

🔗 S'il n'est pas installé :

sudo apt install python3-pip

```
ubuntu@ip-172-31-34-163:~$ sudo apt install python3-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
 binutils binutils-common binutils-x86-64-linux-gnu build-essential cpp cpp-9 dpkg-dev fakeroot g++ g++-9 gcc gcc-9
 gcc-9-base libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan5 libatomic1 libbinutils
 libc-dev-bin libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libdpkg-perl libexpat1-dev libfakeroot
 libfile-fcntllock-perl libgcc-9-dev libgomp1 libisl22 libitm1 liblsan0 libmpc3 libpython3-dev libpython3.8-dev
 libquadmath0 libstdc++-9-dev libtsan0 libubsan1 linux-libc-dev make manpages-dev python-pip-whl python3-dev
 python3-wheel python3.8-dev zlib1g-dev
```

🔗 Appuyez sur la touche Y pour débiter l'installation

```
Setting up python3.8-dev (3.8.2-1ubuntu1) ...
Setting up g++ (4:9.3.0-1ubuntu2) ...
update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto mode
Setting up build-essential (12.8ubuntu1) ...
Setting up python3-dev (3.8.2-0ubuntu2) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9) ...
ubuntu@ip-172-31-34-163:~$
```

## 2. Explications implémentation choisie

Le but de cette partie est d'expliquer l'organisation choisie. Il n'y a aucune manipulation.

Sources :

<https://www.codeflow.site/fr/article/how-to-serve-flask-applications-with-gunicorn-and-nginx-on-ubuntu-16-04#>

<http://sametmax.com/quest-ce-que-wsgi-et-a-quoi-ca-sert/>

<https://qastack.fr/server/331256/why-do-i-need-nginx-and-something-like-gunicorn>



## a. Serveur wsgi

Le serveur http va recevoir une requête http et va renvoyer au client, le navigateur, le fichier demande au format HTML.

En utilisant python, le serveur doit pouvoir communiquer avec le programme écrit dans ce langage : le serveur « passe » la requête à l'application qui lui retourne la réponse.

Un serveur est dit "compatible WSGI" quand il est capable de transmettre une requête HTTP normale à votre application via le protocole WSGI, et qu'il est capable de récupérer une réponse HTTP depuis votre application

## b. Pourquoi ne pas utiliser python/flask seul ?

Il faut bien faire la différence entre teste et déploiement : en test et développement, il est possible d'utiliser le serveur wsgi intégré dans flask (Werzeug), par contre en déploiement il est fortement conseillé d'utiliser un serveur plus efficace.

## c. Schéma de déploiement le plus souvent utilisé

Un seul serveur  
HTTP

Serveur http = serveur http/extérieur + serveur d'application wsgi

Firefox

Ex : nginx

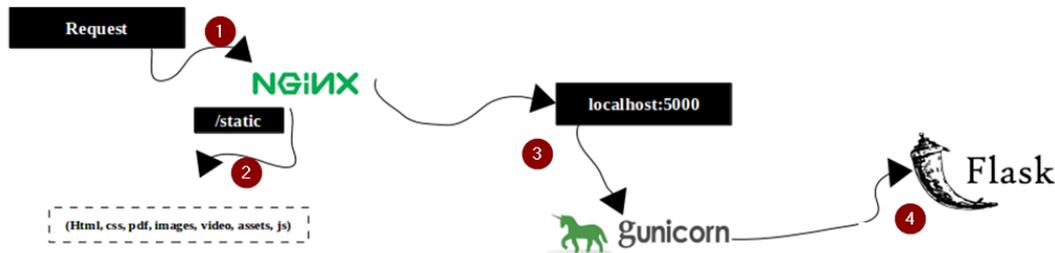
Ex : gunicorn

Ex : application flask

## d. Principe de fonctionnement

1. Un client (navigateur) effectue une requête sur le serveur web
2. Un serveur web (nginx dans notre cas) qui traitera de nombreuses demandes d'images et des ressources statiques et jouera le rôle de proxy
3. Les autres demandes (pages dynamique) sont transmises au serveur d'application wsgi (gunicorn)
4. Il les demande à l'application (application flask) pour être générées dynamiquement

Le résultat suit le chemin inverse.



## IV. Installation des applications sur le serveur linux aws

Sources :

<https://www.youtube.com/watch?v=fbljSY54u20>

[https://signin.aws.amazon.com/signin?redirect\\_uri=https%3A%2F%2Ffeu-west-3.console.aws.amazon.com%2Fec2%2Fv2%2Fhome%3Fregion%3Deu-west-3%26state%3DhashArgs%2523Instances%253A%253Ddesc%253A%253Atag%253AName%26isauthcode%3Dtrue&client\\_id=arn%3Aaws%3Aiam%3A%3A015428540659%3Auser%2Fec2&forceMobileApp=0&code\\_challenge=bC7-f8r-IJQ91ksUYCzlx9uhLdlb04RR5TOn9MjAkk&code\\_challenge\\_method=SHA-256](https://signin.aws.amazon.com/signin?redirect_uri=https%3A%2F%2Ffeu-west-3.console.aws.amazon.com%2Fec2%2Fv2%2Fhome%3Fregion%3Deu-west-3%26state%3DhashArgs%2523Instances%253A%253Ddesc%253A%253Atag%253AName%26isauthcode%3Dtrue&client_id=arn%3Aaws%3Aiam%3A%3A015428540659%3Auser%2Fec2&forceMobileApp=0&code_challenge=bC7-f8r-IJQ91ksUYCzlx9uhLdlb04RR5TOn9MjAkk&code_challenge_method=SHA-256)

<https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/RelatedResources.html>

<https://eu-west-3.console.aws.amazon.com/ec2/v2/home?region=eu-west-3#SecurityGroups:group-id=sg-00fb98e7aea71d077;sort=group-id>

### 1. Flask et applications

#### a. Environnement virtuel

Si vous pensez qu'il soit nécessaire pour votre projet, d'isoler l'environnement suivez ce guide :

<https://openclassrooms.com/fr/courses/4425111-perfectionnez-vous-en-python/4463278-travaillez-dans-un-environnement-virtuel>

Cela n'a pas été réalisé dans cette installation.

#### b. Fin installation des modules

Continuez avec flask

pip3 install flask

```
ubuntu@ip-172-31-34-163:~$ pip3 install flask
Collecting flask
  Downloading Flask-1.1.2-py2.py3-none-any.whl (94 kB)
    |#####| 94 kB 4.3 MB/s
Requirement already satisfied: Jinja2>=2.10.1 in /usr/lib/python3/dist-packages (from flask) (2.10.1)
Collecting itsdangerous>=0.24
  Downloading itsdangerous-1.1.0-py2.py3-none-any.whl (16 kB)
Collecting Werkzeug>=0.15
  Downloading Werkzeug-1.0.1-py2.py3-none-any.whl (298 kB)
    |#####| 298 kB 11.3 MB/s
Requirement already satisfied: click>=5.1 in /usr/lib/python3/dist-packages (from flask) (7.0)
Installing collected packages: itsdangerous, Werkzeug, flask
  WARNING: The script flask is installed in '/home/ubuntu/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed Werkzeug-1.0.1 flask-1.1.2 itsdangerous-1.1.0
```



 Puis avec nginx (serveur http)

sudo apt-get install nginx

 Confirmez par 'Y'

```
ubuntu@ip-172-31-34-163:~$ sudo apt-get install nginx
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  fontconfig-config fonts-dejavu-core libfontconfig1 libgd3 libjpeg-turbo8 libjpeg8
  libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream libtiff5 libwebp6
  libxpm4 nginx-common nginx-core
Suggested packages:
  libgd-tools fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  fontconfig-config fonts-dejavu-core libfontconfig1 libgd3 libjpeg-turbo8 libjpeg8
  libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream libtiff5 libwebp6
  libxpm4 nginx nginx-common nginx-core
0 upgraded, 17 newly installed, 0 to remove and 3 not upgraded.
Need to get 2431 kB of archives.
After this operation, 7891 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://eu-west-3.ec2.archive.ubuntu.com/ubuntu focal/main amd64 fonts-dejavu-core all 2.37-1 [1041 kB]
```

 Puis avec gunicorn (serveur wsgi) et n'oubliez pas dire Yes ou Y ou y

sudo apt-get install gunicorn

```
ubuntu@ip-172-31-34-163:~$ sudo apt-get install gunicorn
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  python3-gunicorn
Suggested packages:
  python3-pastedeploy python3-setproctitle python3-tornado
The following NEW packages will be installed:
  gunicorn python3-gunicorn
0 upgraded, 2 newly installed, 0 to remove and 3 not upgraded.
Need to get 68.5 kB of archives.
After this operation, 323 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://eu-west-3.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 python3-gunicorn all 20.0.4-3 [56.8 kB]
Get:2 http://eu-west-3.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 gunicorn all 20.0.4-3 [11.8 kB]
```

## 2. Application flask de test

 Commençons par créer un répertoire dans lequel on va mettre l'application de test

mkdir flaskapp

 Allez dans ce répertoire

 cd flaskapp/

```
ubuntu@ip-172-31-34-163:~$ cd flaskapp/
ubuntu@ip-172-31-34-163:~/flaskapp$
```

 Dans ce dernier créer les répertoires nécessaires pour flask : templates et static

mkdir templates

mkdir static

 Créez un fichier app.py avec nano

sudo nano app.py

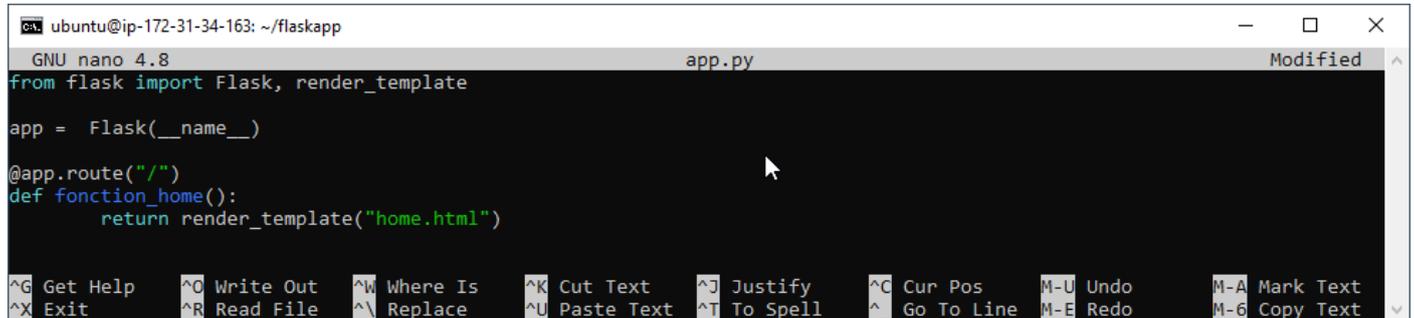
 Et mettez ce code :

```
from flask import Flask, render_template
```

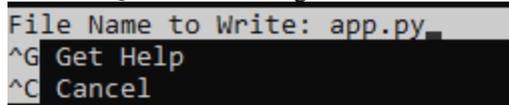


```
app = Flask(__name__)

@app.route("/")
def fonction_home():
    return render_template("home.html")
```



Quittez et sauvegardez avec CTRL X et y si nécessaire



De même allez dans le répertoire templates et créez un fichier home.html:

```
cd templates/
sudo nano home.py
```

Et le fichier home.html :

```
<body>
<h1>Bienvenue dans l'application flask de test </h1>
</body>
```

### 3. Configuration de nginx

Sources :

<https://medium.com/faun/deploy-flask-app-with-nginx-using-gunicorn-7fda4f50066a>

<https://www.digitalocean.com/community/tutorials/how-to-serve-flask-applications-with-gunicorn-and-nginx-on-ubuntu-18-04>

[https://www.codeflow.site/fr/article/how-to-serve-flask-applications-with-gunicorn-and-nginx-on-ubuntu-16-04#\\_configuration\\_de\\_nginx\\_en\\_requ%C3%AAtes\\_proxy](https://www.codeflow.site/fr/article/how-to-serve-flask-applications-with-gunicorn-and-nginx-on-ubuntu-16-04#_configuration_de_nginx_en_requ%C3%AAtes_proxy)

Nginx est un serveur web (http) et un proxy inverse : il peut servir des fichiers statiques (serveur web classique) et gère des requêtes dynamiques (proxy) en les acheminant vers les applicatifs dédiés.

Dans notre cas, il va acheminer les demandes dynamiques vers gunicorn sur le port 8000.

#### a. Récupération de l'adresse publique si vous êtes dans le navigateur

Si vous êtes en connexion dans un navigateur, l'adresse publique est indiquée



Terminal output: `ubuntu@ip-172-31-42-209: ~/flaskapp/templates$`

Instance ID: **i-0a133a3880d9738b9 (PremièreApplicationFlaskRTK)**  
Public IPs: **15.188.119.241** Private IPs: 172.31.42.209

## b. Si vous êtes en ssh

Retournez sur votre instance et récupérez son adresse IP publique (copie dans le presse-papier) :

| Name | Instance ID                  | Instance Type | Availability Zone | Instance State | Status Checks  | Alarm Status | Public DNS (IP)                                   |
|------|------------------------------|---------------|-------------------|----------------|----------------|--------------|---|
|      | <b>2</b> i-0bdd808b2dda27152 | t2.micro      | us-west-2a        | running        | 2/2 checks ... | None         | ec2-54-218-39-152.us-west-2.compute.amazonaws.com |

Instance: **1** i-0bdd808b2dda27152    Public DNS: ec2-54-218-39-152.us-west-2.compute.amazonaws.com

| Description       | Status Checks                   | Monitoring | Tags |
|-------------------|---------------------------------|------------|------|
| Instance ID       | i-0bdd808b2dda27152             |            |      |
| Instance state    | running                         |            |      |
| Instance type     | t2.micro                        |            |      |
| Public DNS (IPv4) | ec2-54-218-39-152.us-west-2.com |            |      |
| IPv4 Public IP    | <b>3</b> 54.218.39.152          |            |      |
| IPv6 IPs          | -                               |            |      |

## c. Fichier de configuration nginx

Allez dans le répertoire de configuration de nginx :

```
cd /etc/nginx/sites-enabled/
```

Créez un fichier propre à notre application flask

```
sudo nano flaskapp
```

Y mettre



```
server {  
    listen 80;  
    server_name 54.218.39.152;  
  
    location / {  
        proxy_pass http://127.0.0.1:8000;  
    }  
}
```

```
GNU nano 4.8  
server {  
    listen 80;  
    server_name 54.218.39.152;  
  
    location / {  
        proxy_pass http://127.0.0.1:8000;  
    }  
}
```

Ce fichier indique que notre proxy écoute sur l'adresse IP publique de votre instance, port 80, et ensuite toutes les demandes seront transmises sur l'adresse locale de la machine (localhost) sur le port 8000.

**Attention aux tabulations en début de lignes (deux tabulations avant proxy )**

 Testez l'application avec

```
sudo nginx -t
```

```
ubuntu@ip-172-31-34-163: /etc/nginx/sites-enabled  
ubuntu@ip-172-31-34-163:/etc/nginx/sites-enabled$ sudo nano flaskapp  
ubuntu@ip-172-31-34-163:/etc/nginx/sites-enabled$ sudo nginx -t  
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok  
nginx: configuration file /etc/nginx/nginx.conf test is successful  
ubuntu@ip-172-31-34-163:/etc/nginx/sites-enabled$
```

 Si tout va bien, redémarrez le service

```
sudo service nginx restart
```

```
ubuntu@ip-172-31-34-163: /etc/nginx/sites-enabled  
ubuntu@ip-172-31-34-163:/etc/nginx/sites-enabled$ sudo nano flaskapp  
ubuntu@ip-172-31-34-163:/etc/nginx/sites-enabled$ sudo nginx -t  
nginx: configuration file /etc/nginx/nginx.conf test is successful  
ubuntu@ip-172-31-34-163:/etc/nginx/sites-enabled$ sudo service nginx restart  
ubuntu@ip-172-31-34-163:/etc/nginx/sites-enabled$
```

### d. Ouverture du port 80

Telle quelle, l'application ne fonctionne pas : seul le port ssh est ouvert sur l'instance aws. Il faut donc ouvrir le port 80 en http.

 Allez sur instance puis cliquez sur le nom juste à côté de *Security groups* :



New EC2 Experience  
Tell us what you think

Lancer une Instance Se connecter Actions

Filtrer par balises et attributs ou rechercher par mot clé

1

| Name                        | ID d'instance      | Type d'instance | Zone de disponi | État de l'instan | Contrôles des statuts | Statut des alarm |
|-----------------------------|--------------------|-----------------|-----------------|------------------|-----------------------|------------------|
| PremièreApplicationFlaskRTK | i-0a133a3880d97... | t2.micro        | eu-west-3c      | running          | 2/2 contrôles réussis | Aucun(e)         |

Instance: i-0a133a3880d9738b9 (PremièreApplicationFlaskRTK) DNS public: ec2-15-188-119-241.eu-west-3.compute.amazonaws.com

2

| Description            | Contrôles des statuts   | Surveillance          | Balises  |
|------------------------|---|-----------------------|--|
| ID d'instance          | i-0a133a3880d9738b9   | DNS public (IPv4)     | ec2-15-188-119-241.eu-west-3.compute.amazonaws.com                       |
| État de l'instance     | running   | IP publique IPv4      | 15.188.119.241   |
| Type d'instance        | t2.micro  | Adresses IP IPv6      | -  |
| Résultat               | Acceptez AWS Compute Optimizer pour obtenir des recommandations. En savoir plus | Adresses IP Elastic   |  |
| Private DNS            | ip-172-31-42-209.eu-west-3.compute.internal                                     | Zone de disponibilité | eu-west-3c   |
| IP privées             | 172.31.42.209   | Groupes de sécurité   | FlaskRTK01. afficher les règles entrantes. afficher les règles sortantes |
| IP privées secondaires |   | Événements planifiés  | Aucun événement planifié   |

Puis sur le premier nom *FlaskRTK01*

EC2 > Groupes de sécurité

Groupes de sécurité (1/2) Infos

Créer un groupe de sécurité

ID du groupe de sécurité: sg-015e15d3bfe6b2915

| ID du groupe de sécu...     | Nom du groupe de sécurité | ID du VPC    | Description                |
|-----------------------------|---------------------------|--------------|----------------------------|
| <b>sg-015e15d3bfe6b2915</b> | FlaskRTK01                | vpc-29e2f140 | Groupe pour applicatio...  |
| sg-c6a0bfa9                 | default                   | vpc-29e2f140 | default VPC security gr... |

Ouvrez les règles de sécurité d'entrée



Supprimer le groupe de sécurité Copier vers un nouveau groupe de sécurité

### Informations

|   |   |   |                           |
|---|---|---|---------------------------|
| Nom du groupe de sécurité<br>FlaskRTK01 | ID du groupe de sécurité<br>sg-015e15d3bfe6b2915      | Description<br>Groupe pour application flask test1    | ID du VPC<br>vpc-29e2f140 |
| Propriétaire<br>732203693209            | Nombre de règles entrantes<br>1 Entrée d'autorisation | Nombre de règles sortantes<br>1 Entrée d'autorisation |                           |

Règles entrantes Règles sortantes Balises

### Règles entrantes

| Type | Protocole | Plage de ports | Source    | Description (facultative) |
|------|-----------|----------------|-----------|---------------------------|
| SSH  | TCP       | 22             | 0.0.0.0/0 | -                         |

[Modifier les règles entrantes](#)

 Ajoutez une règle

EC2 > Groupes de sécurité > sg-015e15d3bfe6b2915 - FlaskRTK01 > Modifier les règles entrantes

## Modifier les règles entrantes Infos

Les règles entrantes contrôlent le trafic entrant qui est autorisé à atteindre l'instance.

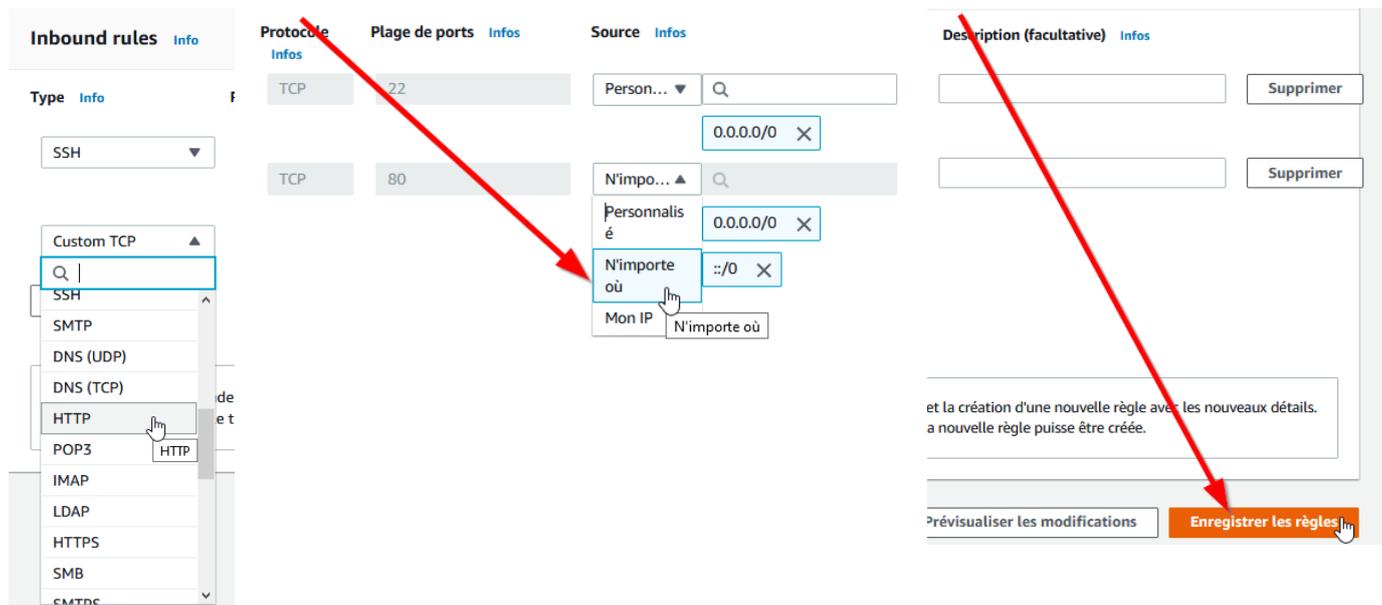
### Règles entrantes Infos

| Type <small>Infos</small> | Protocole <small>Infos</small> | Plage de ports <small>Infos</small> | Source <small>Infos</small> | Description (facultative) <small>Infos</small> |
|---------------------------|--------------------------------|-------------------------------------|-----------------------------|--|
| SSH                       | TCP                            | 22                                  | Person... Q<br>0.0.0.0/0    |  |

[Ajouter une règle](#) [Supprimer](#)

**REMARQUE :** toutes les modifications apportées aux règles existantes entraîneront la suppression de la règle modifiée et la création d'une nouvelle règle avec les nouveaux détails. Cela entraînera l'abandon du trafic qui dépend de cette règle pendant une très brève période de temps jusqu'à ce que la nouvelle règle puisse être créée.

 Ajoutez l'ouverture du port 80 en http :

The screenshot shows the AWS IAM console 'Inbound rules' configuration page. The 'Type' is set to 'SSH'. The 'Protocole' dropdown is set to 'TCP' for port 80. The 'Source' is set to 'N'importe où'. A red arrow points from the 'Protocole' dropdown to 'HTTP' in the list. Another red arrow points from the 'N'importe où' source to the 'Enregistrer les règles' button.

## a) gunicorn

L'application qui va recevoir les requêtes sur le port 8000 est gunicorn. Pour rappel, ce dernier va faire la passerelle entre le proxy http et le programme python.

Retournez dans le répertoire flaskapp

```
cd
cd flaskapp/
```

```
ubuntu@ip-172-31-34-163:/etc/nginx/sites-enabled$ ls
default flaskapp
ubuntu@ip-172-31-34-163:/etc/nginx/sites-enabled$ cd
ubuntu@ip-172-31-34-163:~$ cd flaskapp/
ubuntu@ip-172-31-34-163:~/flaskapp$
```

Pour que gunicorn fasse le lien avec le programme python il faut simplement lui indiquer avec le nom du module sans .py :

```
gunicorn app:app
```

```
ubuntu@ip-172-31-34-163: ~/flaskapp
default flaskapp
ubuntu@ip-172-31-34-163:/etc/nginx/sites-enabled$ cd
ubuntu@ip-172-31-34-163:~$ cd flaskapp/
ubuntu@ip-172-31-34-163:~/flaskapp$ ls
app.py static templates
ubuntu@ip-172-31-34-163:~/flaskapp$ sudo nano app.py
ubuntu@ip-172-31-34-163:~/flaskapp$ gunicorn app:app
[2020-04-29 14:09:35 +0000] [26628] [INFO] Starting gunicorn 20.0.4
[2020-04-29 14:09:35 +0000] [26628] [INFO] Listening at: http://127.0.0.1:8000 (26628)
[2020-04-29 14:09:35 +0000] [26628] [INFO] Using worker: sync
[2020-04-29 14:09:35 +0000] [26630] [INFO] Booting worker with pid: 26630
```

Notez que Gunicorn écoute sur le port 8000 en local :

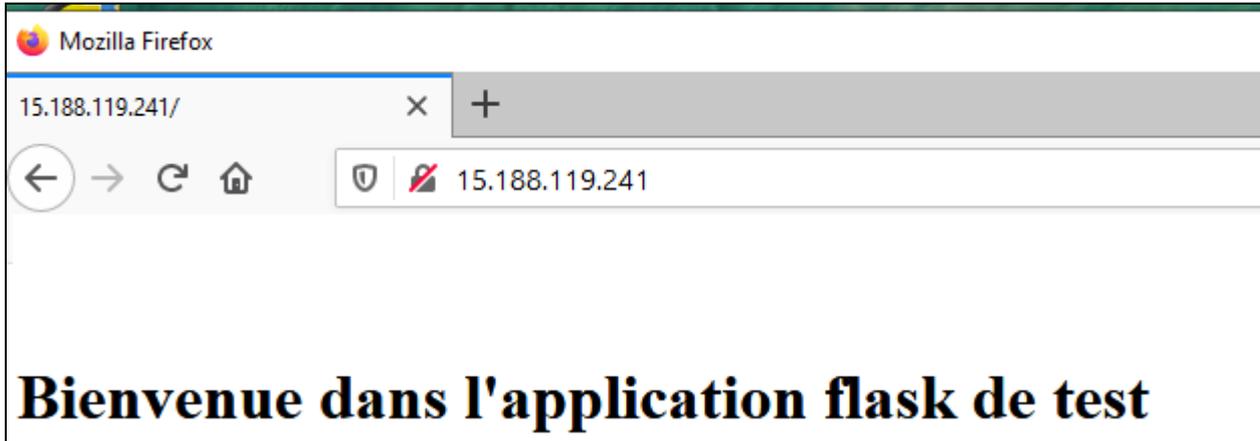
```
ubuntu@ip-172-31-34-163:~/flaskapp$ gunicorn app:app
[INFO] Starting gunicorn 20.0.4
[INFO] Listening at: http://127.0.0.1:8000 (26628)
[INFO] Using worker: sync
```



## V. Tests de fonctionnement

### 1. Sur le port 80

Allez dans un navigateur et entrez l'adresse de votre serveur : 54.218.39.152



### 2. Sur un autre port : le 8080 par exemple

Arrêtez gunicorn (CTRL C)

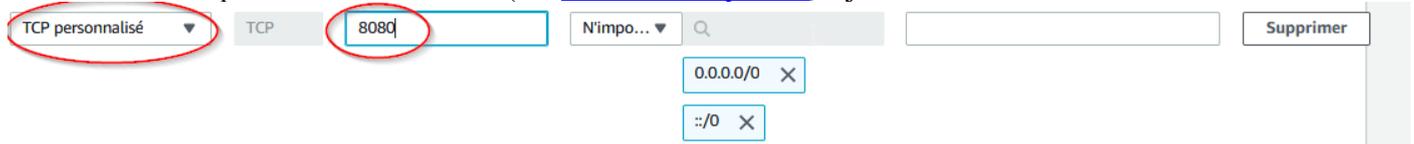
ouvrez le fichier de configuration nginx

```
sudo nano /etc/nginx/sites-enabled/flaskapp
```

modifiez-le comme ceci :

```
server {  
    listen 8080;  
    server_name 54.218.39.152;  
  
    location / {  
        proxy_pass http://127.0.0.1:8000;  
    }  
}
```

ouvrez le port 8080 sur l'instance (c.f. [Ouverture du port 80](#)), ajoutez à la fin :



### 3. Attention

L'adresse de l'instance n'est pas fixe donc elle peut changer au cours du temps. Vérifiez-la de temps en temps.

## VI. Finalisation

Pour l'instant notre installation n'est pas autonome : il faut lancer « à la main » gunicorn.

Dans cette partie nous allons en faire un service qui va être exécuté automatiquement à chaque démarrage.



## 1. Création du service

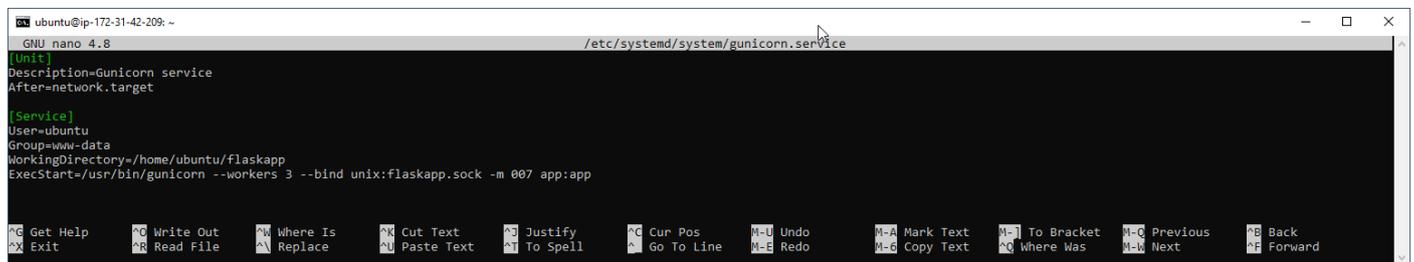
Nous allons créer un fichier de service :

```
sudo nano /etc/systemd/system/gunicorn.service
```

Entrez ces lignes

```
[Unit]
Description=Gunicorn service
After=network.target

[Service]
User=ubuntu
Group=www-data
WorkingDirectory=/home/ubuntu/flaskapp
ExecStart=/usr/bin/gunicorn --workers 3 --bind unix:flaskapp.sock -m 007 app:app
```

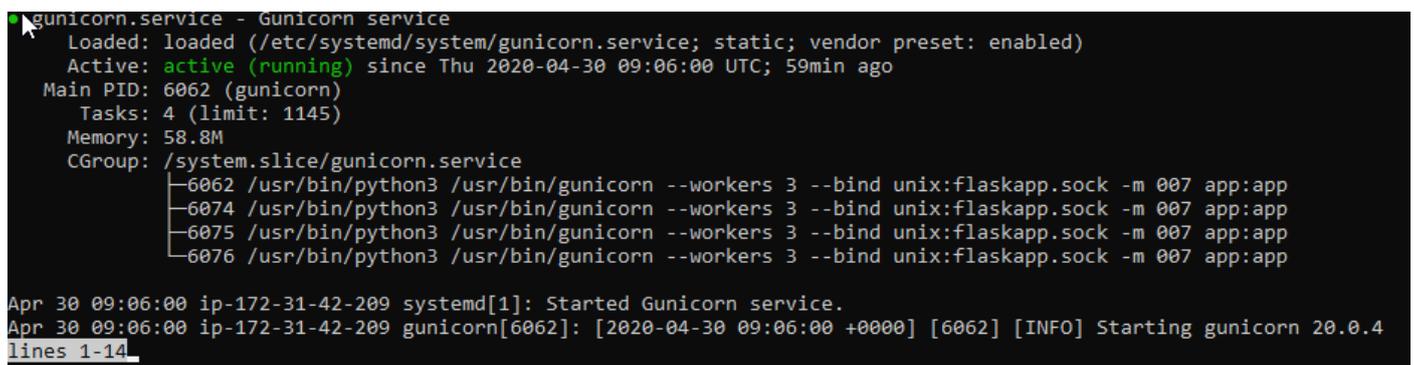


Lancez ce service

```
sudo service gunicorn start
```

Vérifiez son état

```
sudo service gunicorn status
```



```
gunicorn.service - Gunicorn service
  Loaded: loaded (/etc/systemd/system/gunicorn.service; static; vendor preset: enabled)
  Active: active (running) since Thu 2020-04-30 09:06:00 UTC; 59min ago
  Main PID: 6062 (gunicorn)
  Tasks: 4 (limit: 1145)
  Memory: 58.8M
  CGroup: /system.slice/gunicorn.service
          └─6062 /usr/bin/python3 /usr/bin/gunicorn --workers 3 --bind unix:flaskapp.sock -m 007 app:app
            └─6074 /usr/bin/python3 /usr/bin/gunicorn --workers 3 --bind unix:flaskapp.sock -m 007 app:app
              └─6075 /usr/bin/python3 /usr/bin/gunicorn --workers 3 --bind unix:flaskapp.sock -m 007 app:app
                └─6076 /usr/bin/python3 /usr/bin/gunicorn --workers 3 --bind unix:flaskapp.sock -m 007 app:app

Apr 30 09:06:00 ip-172-31-42-209 systemd[1]: Started Gunicorn service.
Apr 30 09:06:00 ip-172-31-42-209 gunicorn[6062]: [2020-04-30 09:06:00 +0000] [6062] [INFO] Starting gunicorn 20.0.4
lines 1-14
```

Vous pouvez faire défiler en appuyant sur entrée et terminer avec la touche q

Si vous n'avez pas d'erreur continuez sinon vérifiez scrupuleusement le fichier.

Vérifiez qu'un fichier flaskapp.sock a bien été créé dans le répertoire flask

```
ls flaskapp/
```



```
ubuntu@ip-172-31-42-209:~$ ls flaskapp/
__pycache__  app.py  flaskapp.sock  history.txt  static  templates
ubuntu@ip-172-31-42-209:~$
```



## 2. Reconfiguration de nginx

Pour l'instant nginx transmettait les requêtes dynamiques à gunicorn sur le port 8000. Maintenant il doit le faire mais avec le service.

 Editez le fichier de configuration nginx propre à gunicorn

```
sudo nano /etc/nginx/sites-enabled/flaskapp
```

 Modifiez-le de cette façon :

```
server{
  listen 80;
  server_name 15.188.119.241;

  location / {
    proxy_pass http://unix:/home/ubuntu/flaskapp/flaskapp.sock;
    #proxy_pass http://127.0.0.1:8000;
  }
}
```

 Relancez les deux services

```
sudo service nginx restart
sudo service gunicorn restart
```



## 3. Lancement du service au démarrage de linux

 Modifiez le fichier de configuration de gunicorn ainsi

```
sudo nano /etc/nginx/sites-enabled/flaskapp
```

Le fichier modifié est :

```
server{
  listen 80;
  server_name 15.188.119.241;

  location / {
    proxy_pass http://unix:/home/ubuntu/flaskapp/flaskapp.sock;
    #proxy_pass http://127.0.0.1:8000;
```



}

🔧 Maintenant créez un fichier de configuration de service gunicorn

```
sudo nano /etc/systemd/system/gunicorn.service
```

Et le fichier :

```
[Unit]
Description=Gunicorn service
After=network.target

[Service]
User=ubuntu
Group=www-data
WorkingDirectory=/home/ubuntu/flaskapp
ExecStart=/usr/bin/gunicorn --workers 3 --bind unix:flaskapp.sock -m 007 app:app

[Install]
WantedBy=multi-user.target
```

🔧 Redémarrez les deux services

```
sudo service nginx restart
sudo service gunicorn restart
```

🔧 Redémarrez l'instance

```
sudo reboot
```

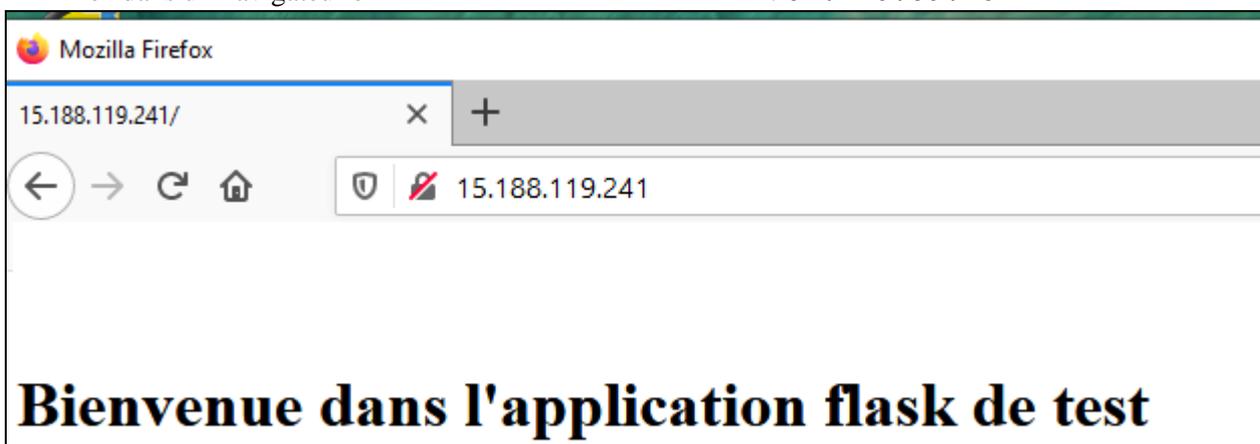
🔧 Attendez un peu et reconnectez-vous

🔧 Autorisez le service au démarrage du linux

```
sudo systemctl enable gunicorn.service
```

```
ubuntu@ip-172-31-42-209:~$ ubuntsystemctl enable gunicornemd/system/gunicorn.service
Created symlink /etc/systemd/system/multi-user.target.wants/gunicorn.service → /etc/systemd/system/gunicorn.service.
```

🔧 Allez dans un navigateur et entrez l'adresse de votre serveur : 54.218.39.152





## VII. Explications des fichiers

### 1. Le fichier de création de service gunicorn

#### a. sudo nano /etc/systemd/system/gunicorn.service

Les explications sont mises après les flèches « -> »

1. [Unit] -> informations sur le service
2. Description=Gunicorn service -> description
3. After=network.target -> démarre après le service
- 4.
5. [Service]
6. User=ubuntu -> l'utilisateur qui va le lancer
7. Group=www-data -> groupe
8. WorkingDirectory=/home/ubuntu/flaskapp -> répertoire de travail
9. ExecStart=/usr/bin/gunicorn --workers 3 --bind unix:flaskapp.sock -m 007 app:app
- 10.
11. [Install]
12. WantedBy=multi-user.target -> indique que le service est lancé lorsque le démarrage du système arrive au niveau runlevel2

#### b. Explications de ligne 9

13. ExecStart=/usr/bin/gunicorn --workers 3 --bind unix:flaskapp.sock -m 007 app:app

/usr/bin/gunicorn :

->Est la localisation du l'exécutable de gunicorn. Vous pouvez le savoir avec la commande which

which gunicorn

```
ubuntu@ip-172-31-42-209:~$ which gunicorn
/usr/bin/gunicorn
```

--workers 3

-> nombre de « worker process », cela correspond au nombre de processus linux qui charge l'application python. La recommandation est (2\*CPU)+1. Dans notre cas un seul CPU.

--bind unix:flaskapp.sock

Crée et « bind » relie un fichier socket (≈ le port) de type unix à l'intérieur du répertoire de travail avec le masque 007 (accès au propriétaire et à son groupe)

Vous pouvez le voir après le lancement du service :

```
ubuntu@ip-172-31-42-209:~$ ls flaskapp/
__pycache__  app.py  flaskapp.sock  history.txt  static  templates
ubuntu@ip-172-31-42-209:~$
```

app:app

Point d'entrée au sens wsgi



### 2. Modification du fichier de configuration nginx propre à gunicorn

```
sudo nano /etc/nginx/sites-enabled/flaskapp
```

Le code est :

```
server{  
    listen 80;  
    server_name 15.188.119.241;  
  
    location / {  
        proxy_pass http://unix:/home/ubuntu/flaskapp/flaskapp.sock;  
        #proxy_pass http://127.0.0.1:8000;  
    }  
}
```

```
#proxy_pass http://127.0.0.1:8000;
```

➔ Mise en commentaire de toute la ligne

```
proxy_pass http://unix:/home/ubuntu/flaskapp/flaskapp.sock;
```

➔ La requête est envoyé vers le fichier qui correspond à la socket sur laquelle attend gunicorn associé à notre application flask.

## VIII. Obtenir une adresse ip publique fixe

L'adresse attribuée à votre instance est certes publique, mais elle peut changer au cours du temps. Pour palier cela Amazon en propose une fixe qui dans aws s'appelle IP Elastic.

### 1. Gratuité

Une adresse IP Elastic n'est pas facturée tant que les conditions suivantes sont réunies :

- L'adresse IP Elastic est associée à une instance EC2.
- L'instance associée à l'adresse IP Elastic est en cours d'exécution.
- Une seule adresse IP Elastic est associée à l'instance

**Dans les autres cas elle est facturée à l'heure !!!!**

Si vous n'avez pas besoin d'une adresse IP Elastic, vous pouvez arrêter l'imputation des frais en libérant l'adresse IP.

Alors faites attention et surveillez vos consommations et/ou mettez des alertes.

Malgré tout cela peut être très intéressant d'avoir une adresse fixe.

### 2. Mise en place d'une IP elastic

 Par défaut vous n'avez qu'une adresse publique qui n'est pas fixe



Instance: **i-0a133a3880d9738b9 (PremièreApplicationFlaskRTK)** DNS public: ec2-15-188-119-241.eu-west-3.compute.amazonaws.com

| Description         | Contrôles des statuts  | Surveillance | Balises |
|---------------------|--|--------------|---------|
| ID d'instance       | i-0a133a3880d9738b9  |              |         |
| État de l'instance  | running  |              |         |
| Type d'instance     | t2.micro   |              |         |
| Résultat            | Acceptez AWS Compute Optimizer pour obtenir des recommandations.<br>En savoir plus |              |         |
| DNS public (IPv4)   | ec2-15-188-119-241.eu-west-3.compute.amazonaws.com                                 |              |         |
| IP publique IPv4    | 15.188.119.241   |              |         |
| Adresses IPv6       | -  |              |         |
| Adresses IP Elastic |  |              |         |

- Allez dans
- (1) Actions
- (2) Mise en réseau
- (3) Gérer les adresse IP

Instance: **i-0a133a3880d9738b9 (PremièreApplicationFlaskRTK)**

État de l'instance: running

Type d'instance: t2.micro

Résultat: Acceptez AWS Compute Optimizer pour obtenir des recommandations.

Vous avez la confirmation, cliquez sur *allouez une adresse IP élastique*



## Gérer les adresses IP

Vous pouvez attribuer et annuler l'attribution des adresses IP IPv4 et IPv6 sur chaque interface réseau. Laissez le champ de l'adresse IP vierge pour qu'une adresse disponible soit attribuée, ou saisissez l'adresse IP à attribuer.

Pour ajouter ou modifier une adresse IP publique IPv4, **allouez une adresse IP élastique** à cette instance ou interface réseau.

▼ eth0: eni-04f88b978726ed2df - 172.31.32.0/20

### Adresses IPv4

| IP privée     | IP publique    |
|---------------|----------------|
| 172.31.42.209 | 15.188.119.241 |

[Attribuer une nouvelle adresse IP](#)

Autoriser la réattribution ⓘ

[Annuler](#) [Oui, mettre à jour](#)

Puis sur *allouer*

EC2 > Adresses IP Elastic > Allouer l'adresse IP Elastic

## Allouer l'adresse IP Elastic

Pour allouer une adresse IP Elastic, sélectionnez le pool d'adresses IPv4 publiques à partir duquel l'adresse IP publique est allouée. Les adresses IP Elastic entraînent des frais si elles ne sont pas associées à une instance en cours d'exécution ou à une interface réseau attachée à une instance en cours d'exécution. [En savoir plus](#)

### Configurations d'adresse IP Elastic

Pool d'adresses IPv4 publiques  
Indique si l'adresse IPv4 publique est allouée à partir du pool d'adresses IPv4 Amazon, ou à partir du pool d'adresses IP que vous avez ajouté à votre compte AWS.

- Pool d'adresses IPv4 Amazon
- Mon pool d'adresses IPv4 publiques(option désactivée, car aucun pool n'a été trouvé) [En savoir plus](#)
- Customer owned pool of IPv4 addresses(option disabled because no customer owned pools found) [En savoir plus](#)

[Annuler](#) [Allouer](#)

Et vous avez une adresse fixe :

# Installer une application flask sur AWS



EC2 > Adresses IP Elastic

Adresses IP Elastic (1/1) Actions Allouer l'adresse IP Elastic

Filter Elastic IP addresses

Adresse IPv4 publique: 35.181.124.223 Clear filters

| <input checked="" type="checkbox"/> | Name | Adresse IPv4 publi... | ID d'allocation            | Instance associée | Adresse IP privée | ID d'association |
|-------------------------------------|------|-----------------------|----------------------------|-------------------|-------------------|------------------|
| <input checked="" type="checkbox"/> |      | 35.181.124.223        | eipalloc-0d79dc795ff9ee62b | -                 | -                 | -                |

🔗 Pour l'allouer à notre instance : Associer ...

EC2 > Adresses IP Elastic

Adresses IP Elastic (1/1) Actions Allouer l'adresse IP Elastic

Filter Elastic IP addresses

Adresse IPv4 publique: 35.181.124.223 Clear filters

| <input checked="" type="checkbox"/> | Name | Adresse IPv4 publi... | ID d'allocation            | Instance associée | Adresse | ID d'association |
|-------------------------------------|------|-----------------------|----------------------------|-------------------|---------|------------------|
| <input checked="" type="checkbox"/> |      | 35.181.124.223        | eipalloc-0d79dc795ff9ee62b | -                 | -       | -                |

Actions menu: Afficher les détails, Libérer les adresses IP Elastic, Associer l'adresse IP Elastic, Dissocier l'adresse IP Elastic

🔗 Cliquez sur le champ instance et choisissez celle qui contient l'application flask(1) puis Associer(2)



EC2 > Adresses IP Elastic > Associer l'adresse IP Elastic

## Associer l'adresse IP Elastic

Choisissez l'instance ou l'interface réseau à associer à cette adresse IP Elastic (35.181.124.223)

Adresse IP Elastic: 35.181.124.223

### Type de ressource

Choisissez le type de ressource auquel l'adresse IP Elastic doit être associée.

- instance  
 Interface réseau

**⚠** Si vous associez une adresse IP Elastic à une instance à laquelle une adresse IP Elastic est déjà associée, l'adresse IP Elastic associée précédemment sera dissociée, mais toujours allouée à votre compte. [En savoir plus.](#)

### instance

🔍 Choisir un type d'instance

i-0a133a3880d9738b9 (PremièreApplicationFlaskRTK) - running

### Adresse IP privée

L'adresse IP privée à laquelle associer l'adresse IP Elastic.

🔍 Sélectionnez une adresse IP privée

### Réassociation

Spécifiez si l'adresse IP Elastic peut être réassociée à une autre ressource si elle est déjà associée à une ressource.

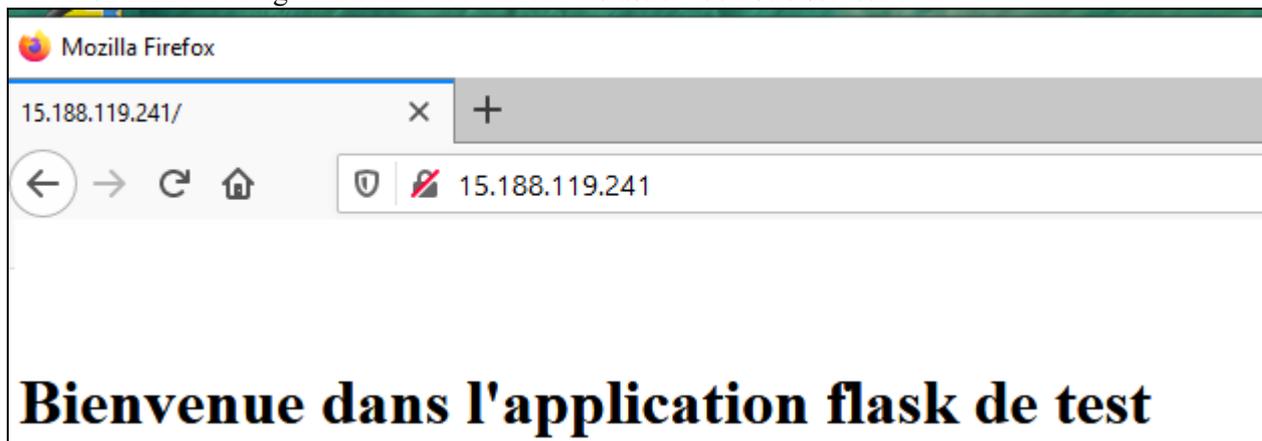
- Autoriser la réassociation de l'adresse IP Elastic

Annuler

Associer

### 3. Test

Allez dans un navigateur et entrez l'adresse « elastic » de votre serveur :





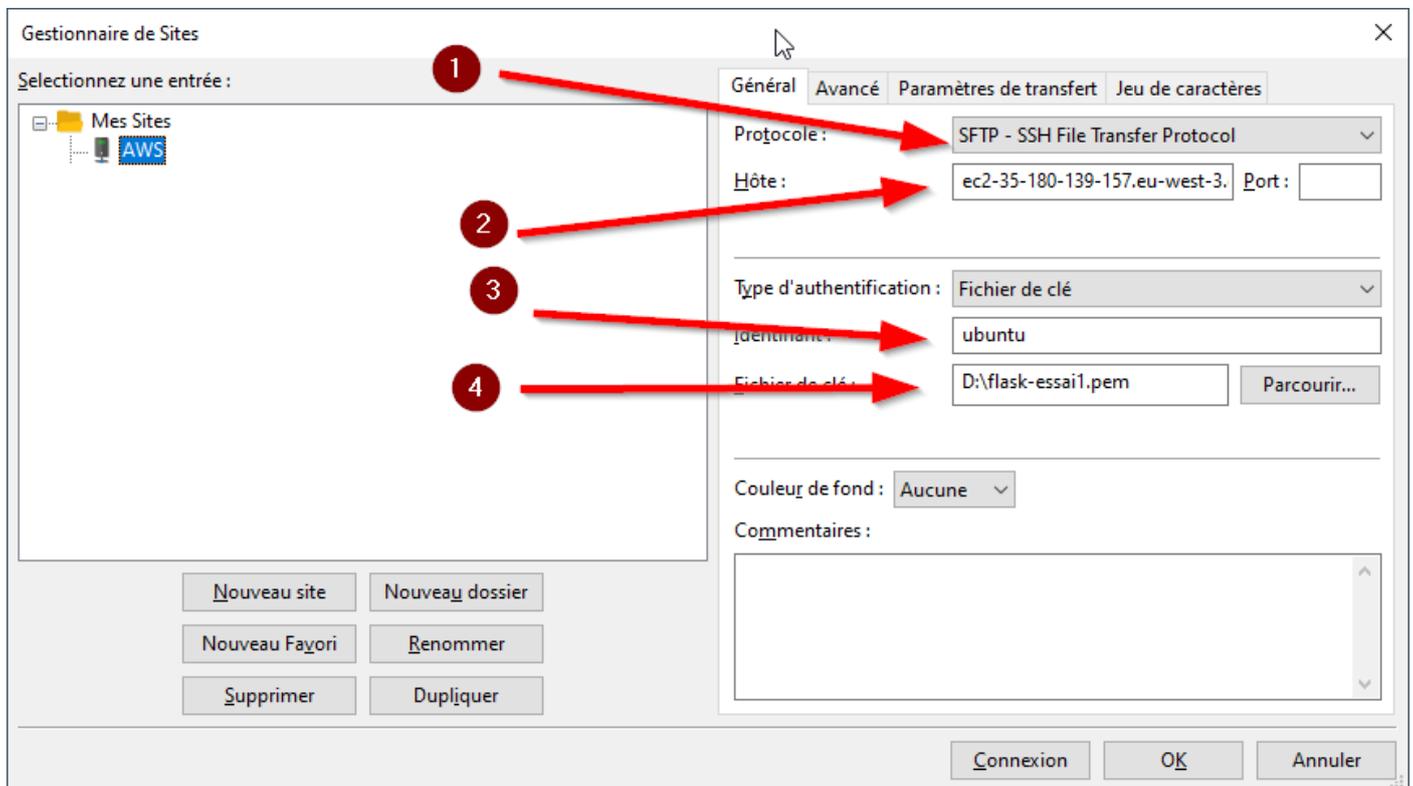
## IX. Téléchargement – Ajout de fichiers

Dans cette partie, nous allons utiliser le logiciel *filezilla* dans le but de déposer des fichiers sur l'instance créée.

### 1. Configuration de filezilla

Dans le gestionnaire de sites :

1. Mettez le protocole *sftp*
2. Là c'est l'adresse Ip ou l'adresse de l'hôte
3. *Ubuntu* c'est le *login*, l'utilisateur au sens linux, qui va se connecter
4. C'est le fichier qui contient la clé de connexion



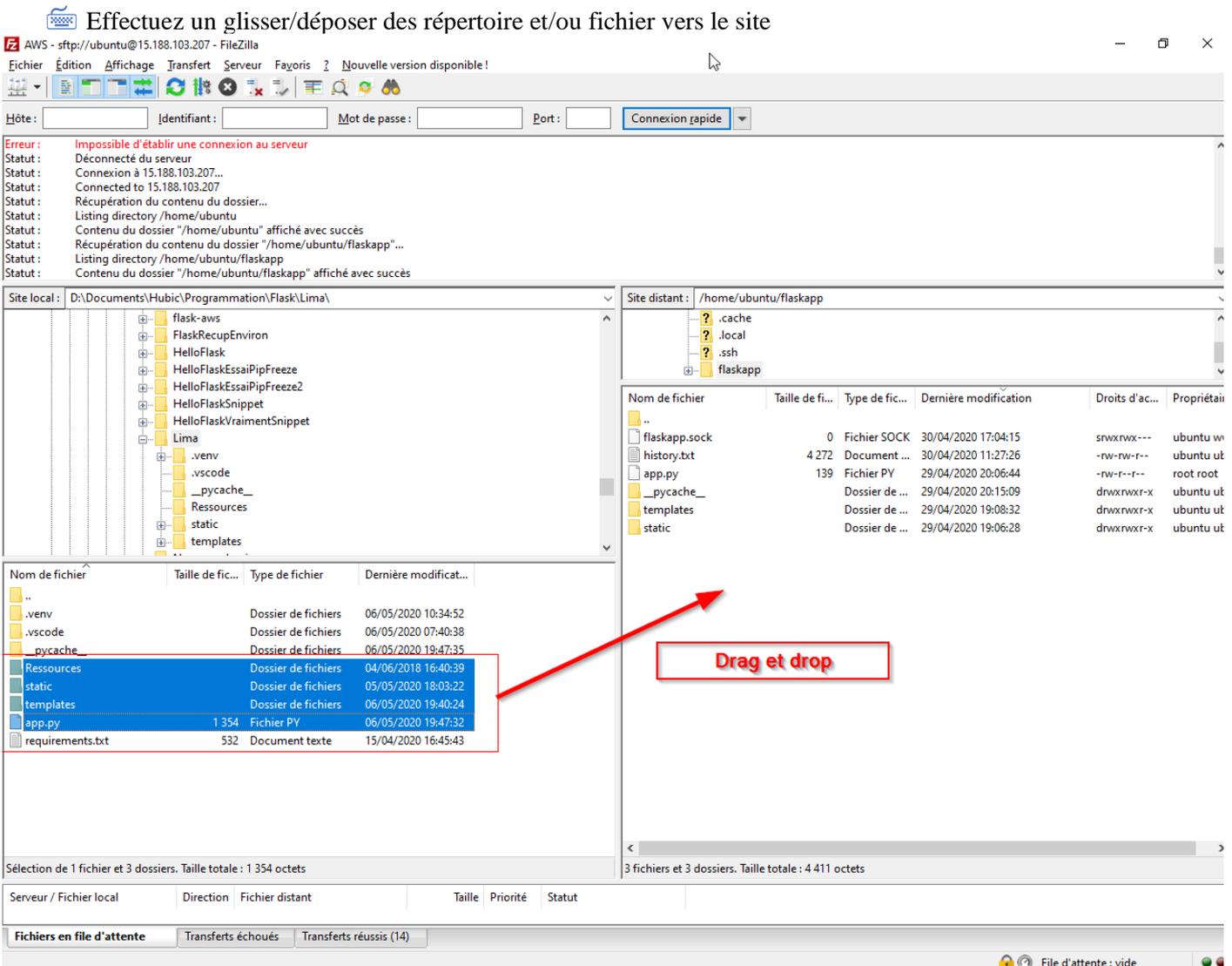
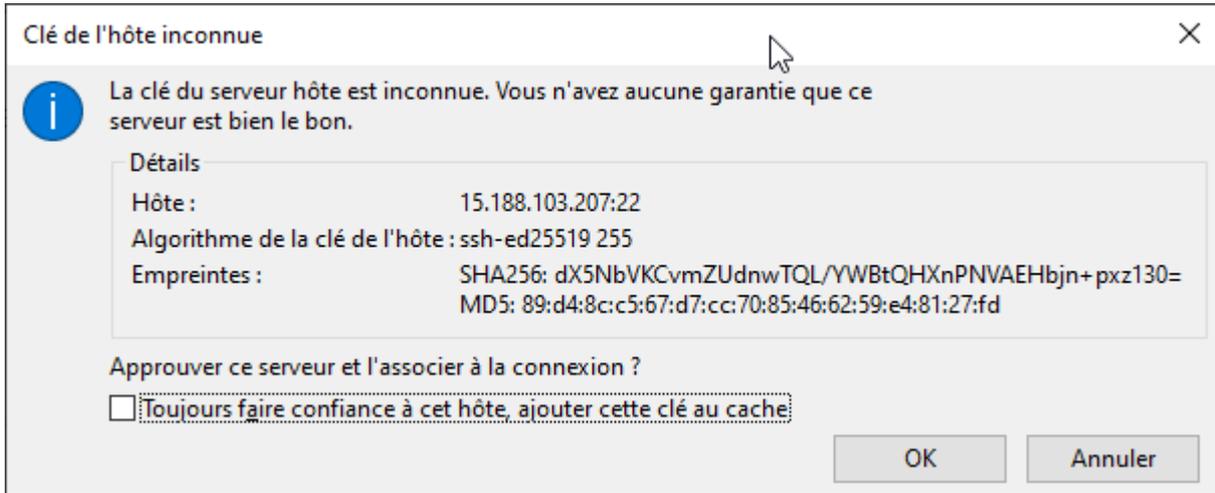
### 2. connexion

La touche *Ok* permet de mémoriser la configuration.

*Connexion* se connecte directement

Sauvegardez votre configuration et connectez-vous à l'instance flask

A la première connexion vous avez ce message



Confirmez le remplacement pour chaque fichier

1. Remplacer
2. Toujours ...

