**Découvrez le cloud avec Amazon Web Services**

# Qu'est-ce que le cloud ?

Amazon Web Services (AWS) est justement une de ces solutions cloud. C'est un service qui vous permet d’utiliser les mécanismes du "cloud computing" et de bénéficier de leurs avantages pour héberger votre prochain site web.

Juillet 2002. [Amazon](https://www.amazon.fr/) est un site de vente en ligne qui cartonne. Simple site de vente de livres à l’origine, on peut aujourd’hui tout acheter : jeux vidéo, matériel hi-fi, vêtements, chaussures… Pour évoluer et gérer de plus en plus de clients, Amazon a dû construire lui-même une très grosse infrastructure technique. Eh oui, il faut des serveurs, beaucoup de serveurs.

Un site comme Amazon est donc hébergé sur d’innombrables serveurs, eux-mêmes regroupés dans des grands entrepôts appelés datacenters. Tous les très gros sites ont leurs propres datacenters, et Amazon ne fait évidemment pas exception à la règle.



Les serveurs sont des ordinateurs comme les autres. Ils ont des ports USB, des ports Ethernet, et évidemment un bouton Power.

En revanche, ils sont plats (pour les empiler) et ils ne sont quasiment jamais branchés à un écran, car on les administre à distance.

### La naissance du cloud

Tous les serveurs ne sont pas utilisés en même temps. Certains sont en attente, prêts pour faire face à la demande lors des pics de vente (comme Noël). Mais alors, que faire de ces serveurs qui dorment ?

Les ingénieurs d’Amazon ont alors eu l’idée de les louer à d’autres développeurs web. Mais pas comme n’importe quel hébergeur web qui louerait des serveurs dédiés, attention : Amazon s’est dit « on ne va pas louer la machine physique elle-même mais la puissance de nos machines »

Les ingénieurs d’Amazon ont eu l’idée de « cacher » le fonctionnement de leurs serveurs et de vendre uniquement de la puissance de calcul et du stockage. Bien sûr, au final, c’est toujours un vrai serveur qui répond à vos. Mais la différence est que vous ne savez pas quel est le serveur qui répond aux requêtes. Parfois, le serveur change dans la journée et vous n’êtes pas au courant, mais ça n’est pas grave : ce qui compte, c’est que votre site fonctionne toujours !

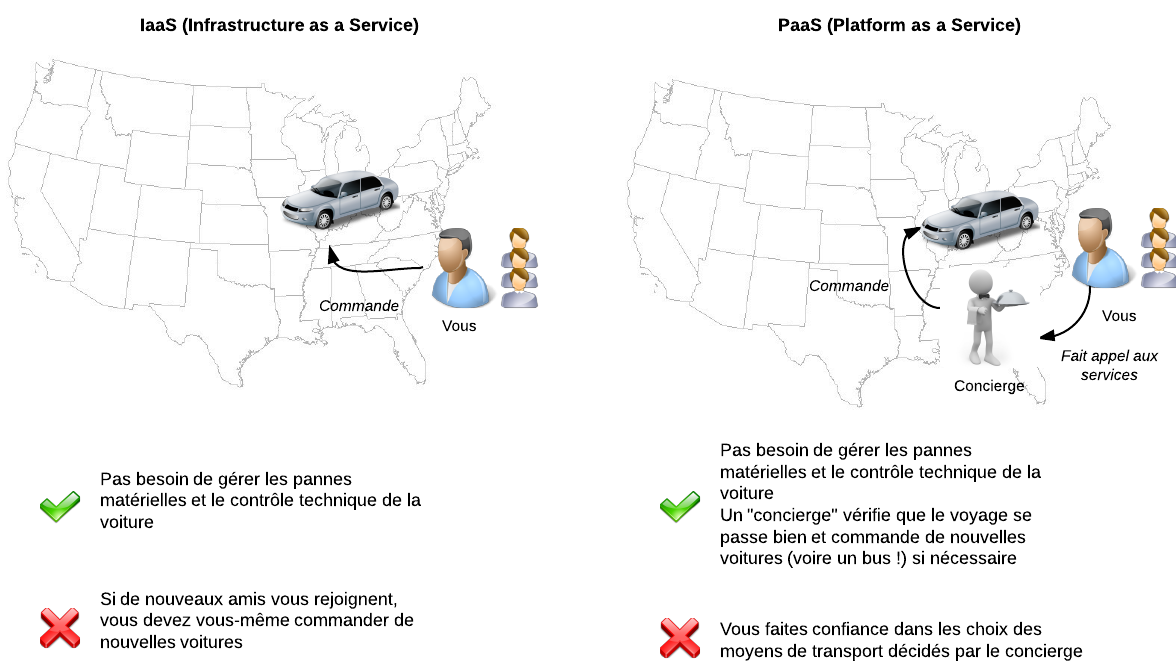
L’intérêt de ce fonctionnement, c’est que vous n’avez plus à vous préoccuper des problèmes physiques des machines. Si un disque dur tombe en panne, ce n'est pas à vous d'aller le remplacer. Vous avez juste besoin de demander à votre site de démarrer sur une autre machine.

## Choisir le bon type de cloud

**IaaS** (Infrastructure as a Service) : un prestataire vous fournit un accès à tout ou partie de son infrastructure technique, c’est-à-dire à ses serveurs.

**PaaS** (Platform as a Service) : on vous fournit non seulement un accès à l’infrastructure, mais on s’occupe aussi de gérer le nombre de machines nécessaires pour que votre application fonctionne bien quelle que soit la charge de trafic

**SaaS** (Software as a Service) : on vous fournit l’accès à un logiciel sous forme de service. Avant, vous deviez installer le logiciel sur votre machine (ex : Microsoft Office). Aujourd’hui, le logiciel se présente sous la forme d’une application web (qui n’est rien d’autre qu’une sorte de super site web !). Vous devez juste vous rendre à une adresse et vous pouvez l’utiliser (ex : Microsoft Office 365, Google Apps…)



### Les principaux fournisseurs de cloud

* Amazon
* Microsoft
* Google
* IBM

Amazon est celui qui possède la plus grande part de marché, et de loin. Il se trouve tout à droite du graphique en terme de part de marché mondiale.

Les autres fournisseurs, comme Google, Microsoft et IBM, ont une plus faible part de marché mais une croissance plus forte.

### Les services proposés par AWS

#### EC2 : Elastic Compute Cloud

Ce service permet de gérer des serveurs sous forme de machines virtuelles dans le cloud.

#### RDS : Relational Database Service

Permet de gérer des bases de données managées dans le cloud. Cela veut dire qu'on met à votre disposition un serveur de base de données préconfiguré.

#### S3 : Simple Storage Service

Amazon S3 (Simple Storage Service) est un service de stockage et de distribution de fichiers. C'est une sorte de gros FTP (même s'il n'est pas basé sur FTP). Utilisez-le pour faire télécharger des fichiers sur votre site ou pour y stocker des images.

#### ... et bien d'autres !

## Créer un compte sur AWS

Il faut le créer sur la [page d'accueil AWS](https://aws.amazon.com) :

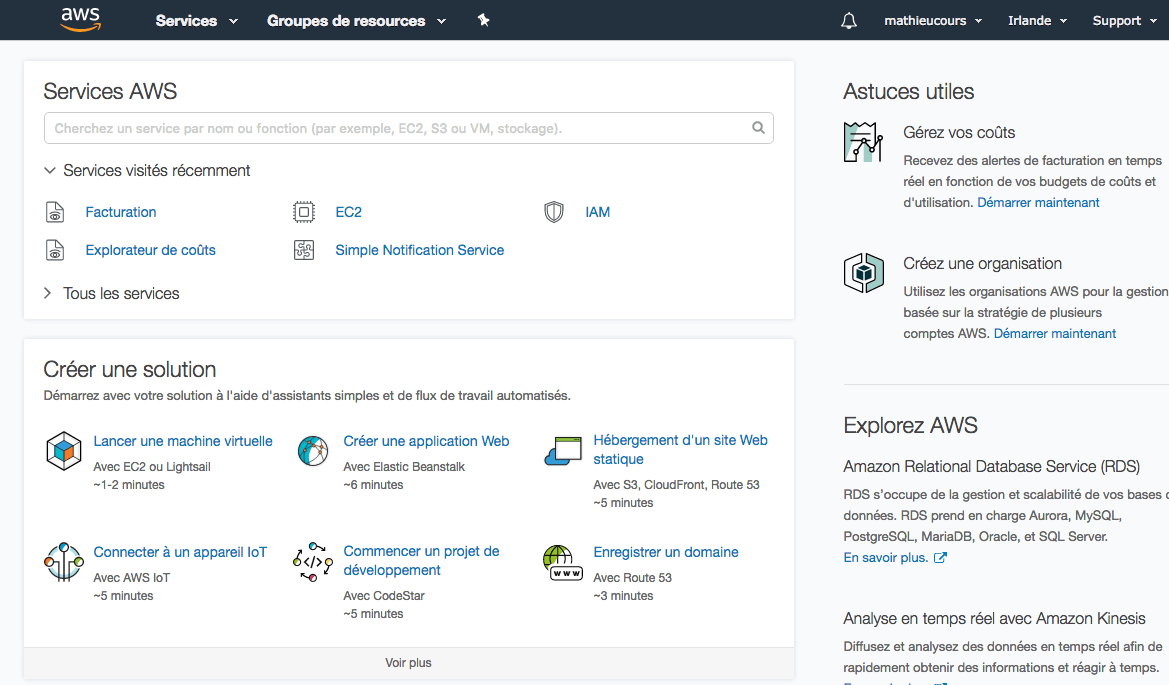
<https://aws.amazon.com>

Vous pouvez utiliser AWS gratuitement pendant la première année sur de nombreux services. Vous obtenez par exemple un serveur EC2 gratuit et un serveur RDS gratuit.

Même si vous n'envisagez pas de payer et que vous comptez rester sur l'offre gratuite, vous devez saisir votre carte bancaire.

## Premiers pas dans la console AWS

Une fois connecté, vous arrivez sur l'accueil de la console AWS :



La **section** "**Région**" (qui affiche "Irlande" sur ma capture d'écran) permet de choisir depuis quel datacenter nous lançons nos services. Par exemple, si je lance un serveur alors que "Irlande" est sélectionné, alors mon serveur démarrera en Irlande.

Quelle région choisir ?

De façon générale, il vaut mieux que vos serveurs soient lancés près de vous et de vos clients. Choisissez donc si possible **le lieu le plus proche**

Avec le cloud, on entend souvent un débat sur la propriété et la souveraineté des données. En théorie, un serveur lancé en France dépend de la juridiction française, un serveur lancé à Londres de la juridiction anglaise, etc.  
Si les considérations légales sont importantes pour vous, choisissez donc avec soin le pays depuis lequel vous lancez vos serveurs.

## Définir un budget

Vous pouvez définir un budget pour être averti dès que vous dépensez une certaine somme d'argent. Etant donné que nous allons utiliser uniquement des **éléments gratuits** d'AWS dans ce cours, cela a peu de chances d'arriver, mais on n'est jamais trop prudent !

Rendez-vous dans le "Tableau de bord Ma facturation" depuis le menu en haut à droite. L'accueil de la section facturation vous indique combien vous avez dépensé pour le mois en cours

Qu'est-ce que l'usage gratuit ?

Pendant la première année, vous pouvez lancer un serveur EC2, un serveur RDS et stocker un certain nombre de fichiers sur S3 entre autres. Vous pouvez retrouver la [liste de tous les services offerts en ligne](https://aws.amazon.com/fr/free/).

Ce n'est pas évident à lire, car la façon dont AWS calcule est un peu inhabituelle. On ne vous dit pas "Un serveur offert " mais "750h de serveur EC2 gratuit par mois". Ces 750h correspondent en fait à un petit serveur allumé en permanence pendant le mois (24h x 31j = 744h).

Comment être alerté si je dépasse l'usage gratuit ?

Rendez-vous sur "Budget" > "Créer un budget". Indiquez un nom à votre budget et un montant. Je propose 1$ :

## Créer un serveur facilement avec Elastic Beanstalk

Il s'agit d'une interface simplifiée qui vous aide à créer vos serveurs Amazon EC2, sans que vous ayez à vous préoccuper des détails.

### Création d'une application

Cliquez sur Démarrer, puis sur la page qui s'ouvre remplissez le formulaire :

Vous devez donner un nom à votre application. C'est le nom du projet que vous aimeriez héberger en ligne

Choisissez ensuite la plateforme de votre application. Vous avez le choix entre de nombreuses plateformes : PHP, Python, Ruby, Java, Go... Ça dépend en quoi votre site web est codé !  
En fonction de votre choix Elastic Beanstalk va démarrer un serveur préconfiguré pour vos besoins.

Cliquez ensuite sur "**Créer une application**". Une console s'affiche pour vous indiquer la progression du lancement du serveur :

Sous le capot, Elastic Beanstalk va démarrer un serveur virtuel EC2... mais ce n'est pas tout ! Il configure aussi plusieurs éléments, comme les groupes de sécurité et les IP dédiées. Cela nous évite d'avoir à réfléchir à ces choses-là pour l'instant et c'est très bien. Sachez que vous pourrez ensuite voir ce qu'a fait Elastic Beanstalk et le modifier sans aucun problème !

### L'interface d'Elastic Beanstalk

Voici à quoi ressemble l'accueil d'une application sur Elastic Beanstalk :



Vous pouvez y voir :

1. **L'état du serveur**. En cas de problème, vous pouvez cliquer sur "Causes" pour avoir les erreurs.
2. **Le nom de l'application** installée sur votre serveur. Nous avons mis une application de démo, mais nous pouvons envoyer notre projet sous format ZIP ici si nous le souhaitons.
3. **Le type de serveur**. Ici, nous avons sélectionné PHP, donc Elastic Beanstalk a lancé un serveur Linux avec PHP installé pour nous.
4. **Les actions** que vous pouvez faire. Vous pouvez par exemple redémarrer l'environnement (cela redémarrera le serveur), le cloner pour en faire une copie, ou bien le résilier pour le supprimer.
5. **L'adresse web de votre serveur** pour tester votre application. Cliquez dessus pour voir l'application de démo qui tourne sur le serveur !



L'adresse du site web générée pour vous n'est évidemment pas pratique à retenir (qui a envie de dire "Eh retrouve mon site sur http://test-env.umpm3qms3k.us-east-2.elasticbeanstalk.com" ?.

Si on veut, on peut rediriger un nom de domaine (ex : monsite.com) vers ce serveur, pour que ça soit plus simple d'accès. Cela se fait depuis AWS Route 53, ou depuis le registrar chez qui vous avez acheté votre nom de domaine (ex : OVH, Gandi...)

Enfin, sur le côté gauche de l'interface d'administration d'Elastic Beanstalk, vous avez quelques menus que je vous laisse parcourir. Il est par exemple possible de configurer l'environnement pour changer la puissance du serveur et même d'y ajouter un serveur de base de données (ça utilise RDS, le service AWS qui fournit les bases de données).

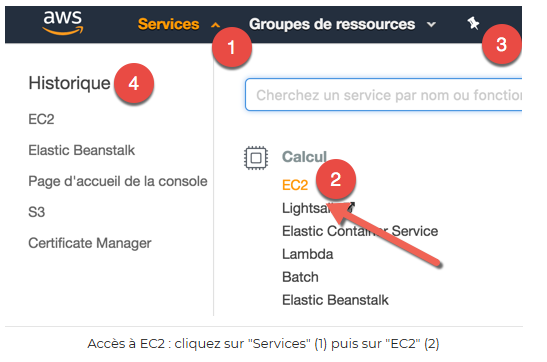
# Introduction aux services d'Amazon EC2

**EC2 est sans aucun doute LE service par excellence d'AWS**. C'est le premier service qui a été lancé historiquement avec AWS. C'est encore aujourd'hui le service le plus important, même si de nombreux autres ont été ajoutés depuis.

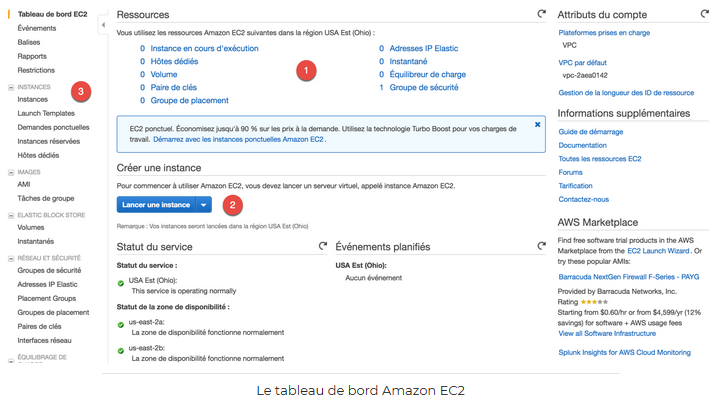
Pourquoi est-ce toujours le service le plus important ?

Parce que c'est le plus... basique. Tout le monde a besoin d'un serveur, que ce soit pour héberger des sites web, pour faire des calculs afin de résoudre des problèmes, pour permettre le fonctionnement d'applications mobiles, etc.

### Tableau de bord EC2



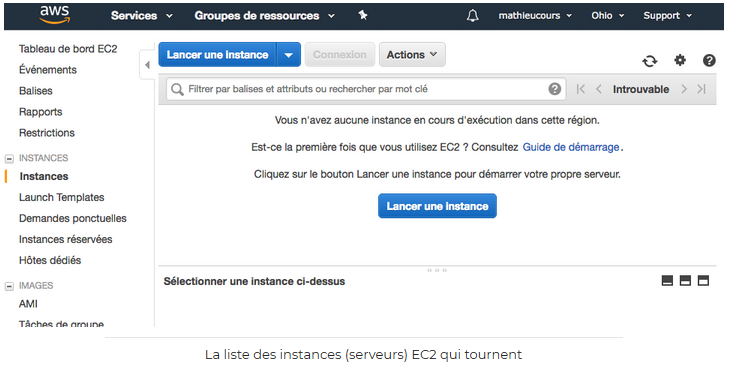
Une fois que vous arrivez sur EC2, respirez à nouveau un grand coup pour ne pas tomber en syncope... cette fois devant le grand nombre de services proposés par EC2 lui-même :



### Les services EC2

#### Instances

La page **Instances** affiche la liste des serveurs EC2 que vous faites tourner. S'il y a un écran à connaître, c'est celui-là.

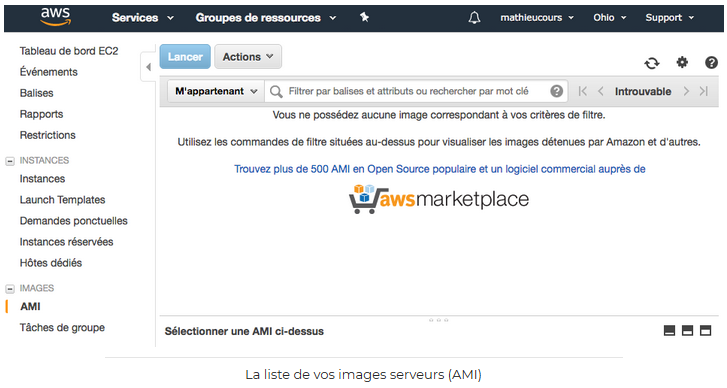


#### Images

Le lien **AMI** affiche les images de vos instances EC2 (AMI signifie : **A**mazon **M**achine **I**mage).

On se sert souvent d'un AMI au départ pour lancer un serveur EC2 préconfiguré . On récupèrera pour notre part un AMI proposé par Amazon ou par sa communauté sur le "Marketplace" (on y trouve des AMI gratuits et payants).

Il est aussi possible de créer vos propres AMI à partir de vos serveurs.



### Elastic Block Store

La section **Volumes** affiche tous les disques utilisés par vos serveurs. Avec AWS, on distingue en effet l'instance (juste la puissance de calcul du processeur en gros) et les disques (Elastic Block Store, ou EBS pour les intimes). Les volumes EBS sont en quelque sorte des disques durs.

Quant aux **Instantanés**, ce sont les sauvegardes des disques EBS de vos serveurs.

### Réseau et sécurité

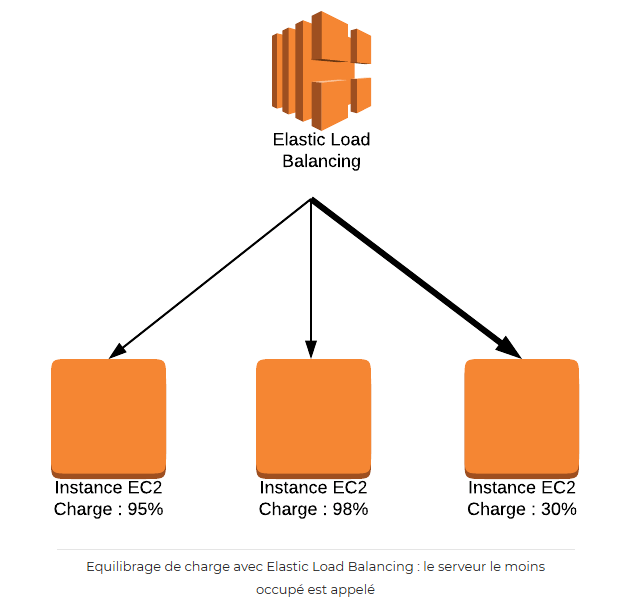
Les **Groupes de sécurité** constituent vos règles de sécurité pour gérer le trafic entrant et sortant de vos serveurs. C'est comme ça que vous pouvez mettre en place un *firewall* (pare-feu)  pour protéger vos serveurs.

Les **Adresses IP Elastic** vous permettent d'affecter une IP statique à vos serveurs. C'est utile car, par défaut, un serveur que vous redémarrez va en général changer d'adresse IP. Grâce aux IP Elastic, vous pouvez faire en sorte que votre serveur garde la même IP (ce qui est indispensable si vous voulez héberger correctement un site web !).

Les **Paires de clés** affichent vos paires de clés créées pour accéder aux serveurs en ligne de commande

### Equilibrage de charge

Si vous avez beaucoup de trafic, l'**Equilibrage de charge** (Elastic Load Balancing) vous permet d'envoyer le trafic vers plusieurs serveurs en choisissant automatiquement le serveur qui est actuellement le moins occupé.



### Auto Scaling

Si vous avez un trafic qui augmente ou qui diminue, l'**Auto Scaling** vous permet d'ajouter ou de retirer des serveurs en fonction du trafic.

Imaginez : si vous avez besoin uniquement de 3 serveurs en temps normal, l'auto scaling fait tourner 3 serveurs. Puis, si vous passez à la télévision et que tout le monde veut voir votre site d'un coup, l'auto scaling va s'activer et lancer plus de serveurs (jusqu'à une limite que vous avez définie). Ainsi, votre site peut passer de 3 serveurs à 10 serveurs au besoin en quelques minutes pour gérer le trafic !

L'auto scaling peut être un bon complément à l'équilibrage de charge lorsque vous avez un gros trafic.

## Choisir le bon type d’instance

### Les types d’instance

* **Usage général** : ce sont des serveurs "équilibrés" qui ne privilégient ni le processeur, ni la mémoire vive, ni le disque dur. Ce sont ceux qu'on utilisera en général pour héberger des sites web. Oui mais voilà, EC2 ne sert pas qu'à héberger des sites web, c'est pour ça qu'on a aussi d'autres types.
* **Calcul optimisé** : ces serveurs sont des monstres de calcul. Les processeurs sont capables d'avaler de nombreuses instructions par seconde. On peut les utiliser pour faire de la modélisation scientifique, de l'analyse distribuée... ou pour des serveurs web qui ont besoin de très hautes performances.
* **Mémoire optimisée** : ces serveurs ont beaucoup de mémoire vive. Beaucoup beaucoup. Vous pouvez en réalité monter jusqu'à 3 900 Go de mémoire vive (soit près de 4 To !). Il faut avoir une application très gourmande en mémoire, comme une base de données stockée en mémoire vive, pour avoir besoin de ça.
* **Calcul accéléré** : ces serveurs sont aussi des monstres de calcul, qui utilisent cette fois des cartes graphiques (GPU). On peut les utiliser pour faire de l'apprentissage machine, gérer des véhicules autonomes, faire de la reconnaissance vocale...
* **Stockage optimisé** : ces serveurs proposent des disques durs très performants. Ce n'est pas tant le stockage qui est grand que la vitesse d'accès aux disques qui est rapide. On peut s'en servir pour faire du calcul distribué MapReduce ou Hadoop, ou encore gérer des systèmes de fichiers distribués.

Pour *chacune* de ces catégories, on vous propose des dizaines de types de serveurs différents plus ou moins puissants.

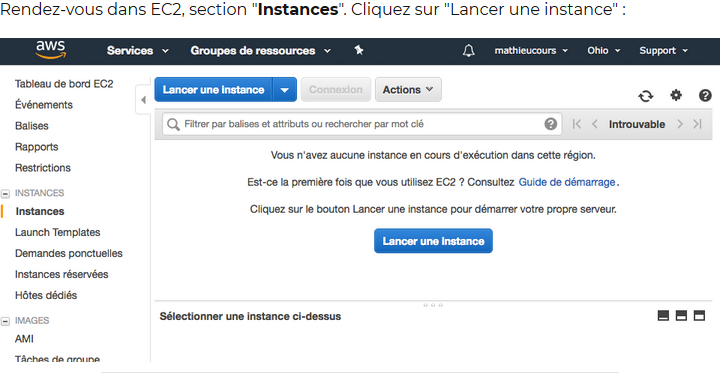
### Les différents types de tarification

On distingue :

* **Tarification à la demande** : c'est ce que nous utiliserons, le plus simple. Il n'y a aucun engagement. Vous avez besoin d'un serveur ? Vous en demandez un. Vous n'en avez plus besoin ? Vous le rendez.
* **Instances spot** : vous êtes radins ? Si vous êtes prêts à attendre les heures les moins chères, les instances spot sont pour vous ! C'est une enchère inversée qui vous permet de payer des serveurs uniquement lorsqu'ils sont peu utilisés, et donc pas chers. Vous ne pouvez pas vraiment vous en servir pour héberger un site web en revanche, car les instances spot démarrent uniquement aux moments les moins chers (par exemple entre 3h et 5h du matin).  
  Avantage : vous pouvez faire jusqu'à 90% d'économie.
* **Instances réservées** : vous savez que vous aurez besoin d'un serveur à coup sûr pendant 1 an ? Si vous êtes prêts à vous engager, les instances réservées vous permettent de payer moins cher (jusqu'à 75% moins cher). Elles sont ensuite à vous pendant la durée de votre engagement.
* **Hôtes dédiés** : plutôt que de gérer des machines virtuelles, vous pouvez demander à avoir une machine physique entière. Vous pouvez en avoir besoin si vous avez des licences logicielles liées à des serveurs physiques (comme Windows Server) ou si vous êtes paranoïaque et ne voulez pas partager votre machine avec d'autres personnes.

## Lancer une instance

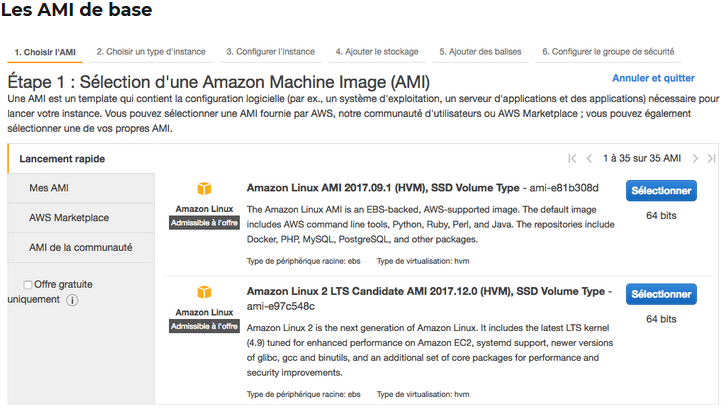
### Page de lancement d’une instance



Lorsque vous cliquez sur "Lancer une instance", un assistant en plusieurs étapes s'ouvre. La première consiste à choisir l'AMI.

### Sélection d’une image de départ(AMI)

Les AMI sont des images du disque. Elles représentent un serveur dans un état donné et servent de point de départ



La première étape sera pour vous l'occasion de choisir votre système d'exploitation :

* **Linux AMI** : une distribution Linux spécialement optimisée pour Amazon EC2. C'est un peu le choix par défaut.
* **Ubuntu Server** : la distribution Linux Ubuntu en version serveur, que vous pouvez choisir si vous l'aimez bien.
* **SUSE Linux** : si votre truc c'est plutôt SUSE.
* **Red Hat** : si votre truc c'est plutôt Red Hat.
* **Windows Server** :

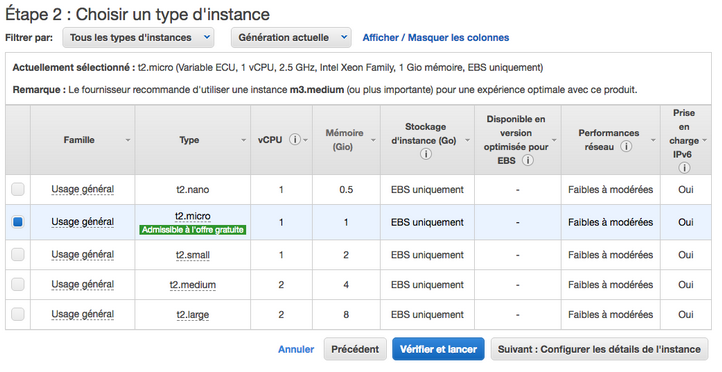
Les AMI préconfigurés du Marketplace

En plus de ces images, il existe de nombreux autres AMI "moins vierges". Ces serveurs préconfigurés peuvent vous faire gagner beaucoup de temps, car les logiciels de base dont vous avez besoin seront déjà installés.

Par exemple, si vous avez besoin de monter un serveur web exécutant un site en PHP, vous aurez besoin d'Apache, PHP, et peut-être que vous voudrez aussi des outils comme phpMyAdmin, Memcache, etc. A moins que vous aimiez passer des heures à tout réinstaller à la main (chacun son truc !), il vaut mieux un AMI préconfiguré.

Allez dans AWS Marketplace (pas de panique, il y a des AMI gratuits). Cherchez-y "LAMP bitnami" et sélectionnez "**LAMP with PHP 7.1 Certified by Bitnami**"

### Choisir un type d’instance



Puis on configure l’instance

### Ajouter le stockage

EC2 sépare les choses :

* Il y a d'un côté le **serveur** (la puissance de calcul)...
* ... et de l'autre les **disques**. Il peut s'agir de disques durs ou de SSD (mémoire flash) selon le type d'instance que vous avez sélectionné.

Les disques sont gérés par un sous-service d'EC2 qu'on appelle EBS (Elastic Block Store). Par défaut, on nous a configuré ici un volume EBS de 10 Go :



Par défaut, votre disque est un SSD. C'est ce qu'on vous recommande si vous ne savez pas ce que vous voulez.  
Sachez que si vous changez le "Type de volume", vous pouvez aussi avoir un SSD encore plus rapide, ou bien un disque magnétique (c'est-à-dire un bon vieux disque dur).

### Ajouter des balises

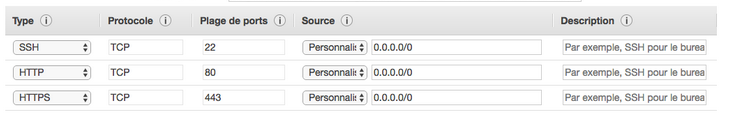
Cette section permet de "tagger" le serveur pour pouvoir le retrouver plus rapidement ensuite

Cela fonctionne par un système de clé / valeur. Par exemple, si vous voulez, vous pouvez ajouter une balise  site = openclassrooms  . La clé est "site", la valeur est "openclassrooms". Ainsi, si vous faites une recherche par la suite, vous pourrez facilement retrouver tous les serveurs qui sont reliés au site OpenClassrooms.

Ces balises sont utiles principalement quand vous devez gérer beaucoup de serveurs.

Configurer le groupe de sécurité

C'est ici que vous configurez votre firewall (pare-feu) :



Ici, mon serveur sera lancé avec un nouveau groupe de sécurité, qui autorise uniquement un certain type de trafic *entrant*. Par exemple, on voit que mon serveur autorise l'accès SSH, HTTP et HTTPS depuis toutes les IP (d'où le 0.0.0.0/0).

Pour une sécurité optimale, vous pouvez par exemple autoriser l'accès SSH depuis uniquement votre IP en cliquant sur le menu déroulant. Attention, il ne faut pas que votre IP change.

Les groupes de sécurité peuvent être partagés entre plusieurs serveurs. Donnez-leur un nom facile à retrouver (exemple : "Web server firewall config").

Il est important de bien configurer votre groupe de sécurité. Si vous oubliez SSH, vous ne pourrez pas vous connecter à votre serveur ! Si vous oubliez HTTP, personne ne pourra charger un site web sur votre serveur !

Lancement du serveur

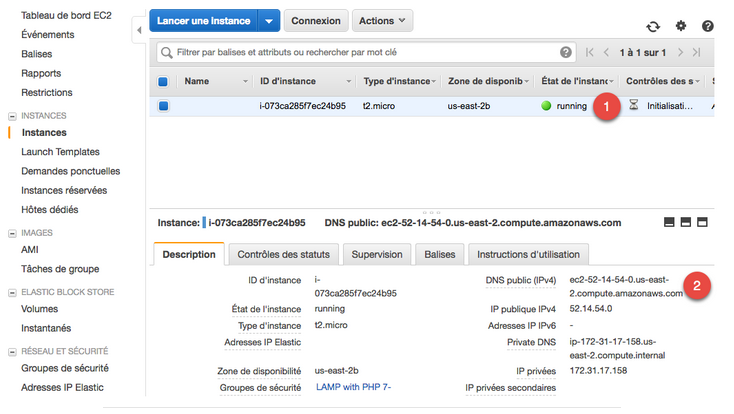
Cliquez ensuite sur "Vérifier et lancer". On vous proposera de vérifier à nouveau votre configuration.

Ensuite, on vous demande avec quelle clé SSH vous souhaitez vous connecter au serveur. En effet, pour vous connecter en ligne de commande, vous aurez besoin d'une paire de clés (publique/privée).

En temps normal, on se connecte à un serveur avec un login et un mot de passe (par exemple : "root" / "superpassword"). AWS considère que cette méthode n'est pas optimale et vous invite plutôt à vous connecter avec une paire de clés. C'est généralement plus sûr et surtout plus pratique, puisque vous n'aurez pas à saisir votre mot de passe à chaque fois.

Dans la fenêtre qui s'affiche, vous pouvez soit sélectionner une paire de clés existante (si vous avez déjà créé un serveur sur AWS), soit créer une nouvelle paire de clés si c'est votre première fois. On va dire que c'est votre première **Cette étape est très importante !** Cliquez sur "Télécharger une paire de clés". Vous allez télécharger un fichier .pem qui vous permettra ensuite d'accéder à votre serveur. Mettez-la dans un lieu sûr (ne la publiez pas sur Internet ou c'est foutu !).

Si vous perdez votre clé, vous ne pourrez plus accéder à votre serveur



Vous pouvez voir que l'instance de type t2.micro est en statut vert "running" **(1)**.

Il y a de nombreuses informations dans la section en bas de l'écran. La plus importante est peut-être "DNS public (IPv4)" **(2)** qui affiche l'adresse de votre serveur. Si vous la copiez-collez dans votre navigateur, vous pourrez voir ce que votre serveur web affiche par défaut.

Mon serveur affiche une page web par défaut parce que j'ai choisi un AMI préconfiguré. Si j'avais pris un AMI "vierge", il n'y aurait pas eu de serveur web et donc rien ne se serait affiché à cette URL.

Vous pouvez faire un clic droit sur l'instance pour l'arrêter, la redémarrer ou la résilier. Si vous la résiliez, elle sera supprimée (elle restera juste affichée quelques temps avant de disparaître complètement).

Se connecter à une instance

TP : Héberger une page web

Configurer une IP élastique

Sauvegarder et restaurer une instance

## Qu'est-ce que Simple Storage Service (S3) ?

Amazon Simple Storage Service (abrégé S3) est un service de stockage de données. En fait, il s'agit tout bêtement d'un moyen de stocker des fichiers sur Internet, qui est devenu très populaire. Si populaire qu'on pourrait même dire qu'il s'agit du service le plus célèbre d'AWS avec EC2.

C’est un peu comme un hébergeur mais pas un hébergeur de sites web. Il faut plutôt le voir comme un gros hébergeur de fichiers. (en fait, on peut héberger des pages HTML basiques, mais pas de scripts PHP, Ruby...).

C’est un FTP ?

Non. S3 n'est pas accessible en FTP, mais vous verrez que ça y ressemble par bien des aspects. Vous pouvez y déposer des fichiers et les lire.

**Tarif ?**

Il y a un usage gratuit qui vous suffira probablement au début (5 Go de stockage).

En temps normal, vous payez pour l'espace nécessaire, mais aussi pour le nombre d'envois et de téléchargements des fichiers, ainsi que la bande passante utilisée.

**Utilité ?**

Pour stocker toutes sortes de fichiers sur Internet. Par exemple :

* Netflix s'en sert pour stocker les fichiers vidéo de ses films et séries.
* Airbnb s'en sert pour stocker les images de ses utilisateurs.
* OpenClassrooms s'en sert pour stocker les images de ses cours, mais aussi les exemples et les vidéos à télécharger, ainsi que les travaux réalisés par les apprenants dans les activités et projets.
* Beaucoup s'en servent aussi pour stocker des backups de leurs bases de données.

**Les avantages ?**

En résumé, S3 est une sorte de gros hébergeur FTP... mais qui n'utilise pas FTP. "Okay", vous dites-vous. Mais qu'est-ce qu'on y gagne ? On pourrait très bien héberger les fichiers sur notre serveur web EC2 après tout !

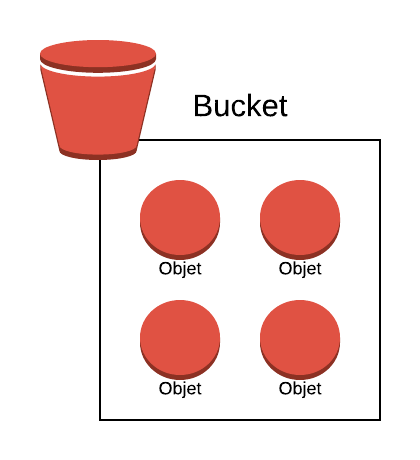
Voici une petite liste d'avantages que S3 vous apporte :

* Vous pouvez configurer facilement les droits d'accès pour chaque fichier. Qui peut lire, modifier et supprimer chaque fichier.
* Vous pouvez chiffrer toute une partie du contenu sur S3 si vous le souhaitez pour des raisons de sécurité.
* Vos fichiers peuvent être versionnés : vous pouvez revenir à une version précédente à tout moment !
* Vos fichiers peuvent avoir une date d'expiration et être supprimés automatiquement au bout d'un moment.
* Vos fichiers peuvent être répliqués automatiquement sur plusieurs datacenters AWS. Ainsi, vous diminuez le risque de perdre des données importantes.
* Il n'y a pas de limite de place. Vous ne risquez pas d'être bloqués par un disque dur rempli

**Structure de S3**

Amazon S3 propose de stocker des données dans des *buckets* (littéralement des... seaux). Ce sont des sortes de gros conteneurs qui peuvent stocker autant de fichiers que l'on veut (répartis dans des dossiers à l'intérieur s'il le faut).

A l'intérieur de chaque bucket, vous pouvez déposer des fichiers (on parle d'*objets*) et y associer des métadonnées. Vous pouvez indiquer ce que vous voulez dans ces métadonnées (l'auteur du fichier par exemple).



Il faut aller sur la console AWS puis sur S3 (Simple Storage Service) et choisir créer un bucket (compartiment)

Il vous faudra donner un nom unique à votre bucket. Ce nom ne doit jamais avoir été utilisé par quelqu'un d'autre. Ca fonctionne un peu comme les noms de domaine !

Ajouter et modifier des objets dans le bucket

Une fois le fichier rendu public, vous pouvez le rendre privé à nouveau en allant dans les autorisations, puis en décochant le droit "Lire l'objet" pour "Everyone".

Doit-on vraiment passer par l'interface d'AWS pour ajouter et supprimer des fichiers ?

Non ! L'interface d'AWS est pratique, mais en fait on passe généralement par une API. Les API d'AWS sont très bien faites. Celle de S3 peut être utilisée pour charger des fichiers depuis votre site web.

**Définir les droits d’accès à S3**

Les droits d'accès à un bucket et ses fichiers sont déterminés par un processus assez complexe de *policies* :

Il y a 2 types de règles :

* User policy : définit ce qu'un utilisateur a le droit de faire.
* Resource-based policy : définit ce qu'on a le droit de faire sur un bucket ou un fichier.

Exemples : User policy : "Jennifer a le droit de lire et modifier tous les fichiers dans le bucket A, ainsi que d'ajouter des fichiers PNG dans le bucket B".

Resource-based policy : "Dans le bucket A, Jennifer a le droit de lire et modifier tous les fichiers, tandis que Patrick a le droit uniquement d'ajouter des fichiers.".

Les utilisateurs dans AWS sont créés via le service IAM (allez y faire un tour pour voir !). L'utilisateur principal dont vous vous servez actuellement est appelé "root". Il est recommandé de créer des utilisateurs IAM, plutôt que de passer par le compte root.

**Structure d’une resource-based policy**

Voici une policy très simple. Elles sont écrites au format JSON :

{

"Version":"2012-10-17",

"Statement": [

{

"Effect":"Allow",

"Principal": "\*",

"Action":["s3:GetObject"],

"Resource":["arn:aws:s3:::examplebucket/\*"]

}

]

}

* **Version** : c'est le numéro de version de la policy. Ne mettez pas la date du jour, copiez la même date que moi (ça revient en gros à dire "j'utilise la version 3 du système de policy").
* **Statement** : ce sont les règles de la policy. Ici, il y en a une seule. Voyons ce que ça dit à l'intérieur :
  + **Effect** : Allow (autoriser) ou Deny (refuser). Par défaut, tout est refusé par sécurité. Il faut donc explicitement autoriser des choses.
  + **Principal** : le nom de l'utilisateur à qui on donne le droit. Ici, l'étoile \* signifie "tout le monde, y compris le grand public sur internet sans compte AWS".
  + **Action** : c'est l'action qu'on veut autoriser (il peut y en avoir plusieurs). Ici,  s3:GetObject  permet de télécharger un objet (un fichier). La [liste des actions](https://docs.aws.amazon.com/fr_fr/AmazonS3/latest/dev/using-with-s3-actions.html) peut être retrouvée dans la doc. Oui, il y en a beaucoup ! Parmi les plus importantes, citons :
    - s3:DeleteObject : autorise la suppression des fichiers
    - s3:GetObject  : autorise la lecture des fichiers et leur téléchargement
    - s3:PutObject   : autorise l'ajout de fichiers
    - s3:ListBucket  : autorise la récupération du nom de tous les fichiers dans le bucket
    - s3:ListAllMyBuckets  : autorise l'affichage de la liste de tous les buckets.
  + **Resource** : le nom de la ressource qui est autorisée. Il y a un format un peu spécial. Ici,  arn:aws:s3:::examplebucket/\*  indique qu'on effectue l'autorisation sur tous les fichiers (  \*  ) du bucket nommé  examplebucket  .

**Un autre exemple :**

{

"Version": "2012-10-17",

"Id": "ExamplePolicy01",

"Statement": [

{

"Sid": "ExampleStatement01",

"Effect": "Allow",

"Principal": {

"AWS": "arn:aws:iam::Account-ID:user/Dave"

},

"Action": [

"s3:GetObject",

"s3:GetBucketLocation",

"s3:ListBucket"

],

"Resource": [

"arn:aws:s3:::examplebucket/\*",

"arn:aws:s3:::examplebucket"

]

}

]

}

Cette policy autorise l'utilisateur Dave (créé sur AWS IAM) à lire un objet, connaître la localisation d'un bucket et à lister le contenu du bucket. Les ressources qui sont affectées sont examplebucket et tous ses fichiers à l'intérieur.

Exemple 3

{

"Version":"2012-10-17",

"Id":"ExamplePolicy02",

"Statement":[

{

"Effect":"Allow",

"Principal":{

"AWS":"\*"

},

"Action":"s3:GetObject",

"Resource":"arn:aws:s3:::my-brand-new-bucket/public/\*"

},

{

"Effect":"Allow",

"Principal":{

"AWS":"arn:aws:iam::Account-ID:user/Dave"

},

"Action":[

"s3:PutObject",

"s3:DeleteObject"

],

"Resource":"arn:aws:s3:::my-brand-new-bucket/\*"

}

]

}

Cette policy autorise tout le monde à lire les objets dans le dossier "public" de "my-brand-new-bucket". Cela veut dire que n'importe quel internaute pourra télécharger les fichiers du dossier "public" de ce bucket, s'il connaît son URL. Si vous hébergez les images de votre site sur S3, vous aurez sûrement besoin de donner ce genre de droit.

Cette policy donne aussi d'autres autorisations (vous pouvez voir qu'il y a 2 *statements*). Elle dit que Dave peut ajouter et supprimer des fichiers partout dans ce bucket.

En résumé : tout le monde peut lire les fichiers dans "public", mais seul Dave peut en ajouter ou en supprimer.

**Activer une policy**

Si vous voulez ajouter une policy au format JSON pour un bucket, comme on vient de le voir, il faut aller dans le bucket, onglet "Autorisations" / "Stratégies de compartiment"

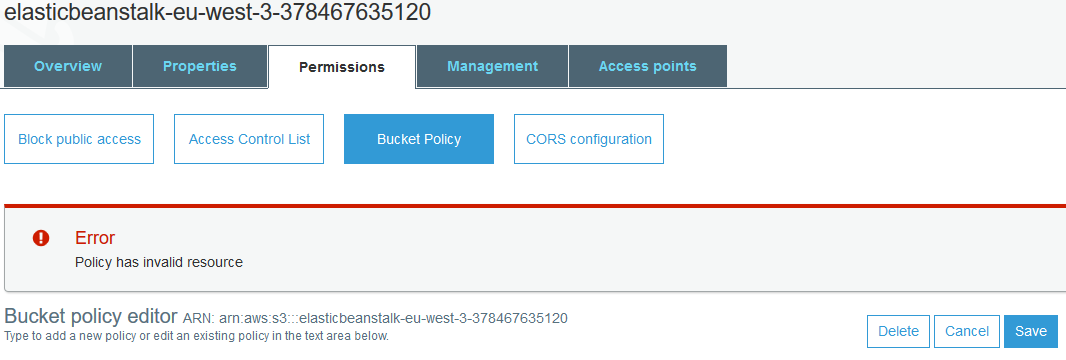
Il existe un générateur de policies en ligne pour aider à les écrire :

https://awspolicygen.s3.amazonaws.com/policygen.html

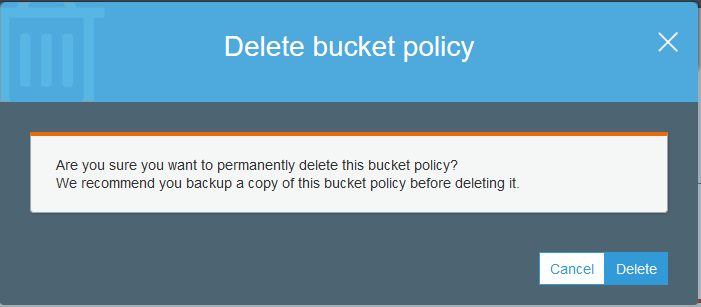
# Learn AWS by doing

**Supprimer un bucket**

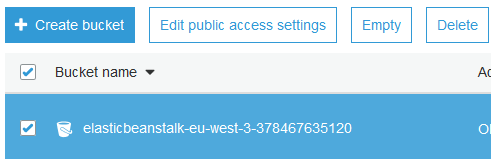
Avant de pouvoir supprimer un bucket il faut d’abord supprimer les règles ( policy) du bucket qui interdisent de le supprimer.



On clique directement sur Delete puis on Save



Ensuite on peut le supprimer :



**Hébergement d’un site Web statique avec S3 sur AWS**

Ce que vous accomplirez :

Déployer un site Web statique en utilisant [Amazon S3](https://aws.amazon.com/fr/s3/). Amazon S3 fournira la base pour votre site Web ainsi que du stockage pour votre contenu statique.

Associer un nom de domaine à votre site Web à l'aide d'[Amazon Route 53](https://aws.amazon.com/fr/route53/). Amazon Route 53 indiquera le Domain Name System (DNS) où trouver votre site Web.

Permettre à votre site Web de charger rapidement en utilisant [Amazon CloudFront](https://aws.amazon.com/fr/cloudfront/). Amazon CloudFront créera un réseau de diffusion de contenu (CDN) qui hébergera le contenu de votre site Web à proximité de vos utilisateurs.

**Amazon Route 53 –** You use Route 53 to register domains and to define where you want to route internet traffic for your domain. We explain how to create Route 53 alias records that route traffic for your domain (example.com) and subdomain (www.example.com) to an Amazon S3 bucket that contains an HTML file.

**Amazon S3 –** You use Amazon S3 to create buckets, upload a sample website page, configure permissions so that everyone can see the content, and then configure the buckets for website hosting.

## Step 1: Register a Domain

## Step 2: Create and Configure Buckets and Upload Data

To support requests from both the root domain such as example.com and subdomain such as www.example.com, you create two buckets. One bucket contains the content. You configure the other bucket to redirect requests.

### Step 2.1: Create Two Buckets

The bucket names must match the names of the website that you are hosting. For example, to host your example.comwebsite on Amazon S3, you would create a bucket named example.com. To host a website under www.example.com, you would name the bucket www.example.com. In this example, your website supports requests from both example.com and www.example.com.

In this step, you sign in to the Amazon S3 console with your AWS account credentials and create the following two buckets.

* *example.com*
* www.*example.com*

# Ensuite il faut aller dans Autorisations puis Paramètres d’accès public pour autoriser l’accès en public au compartiment.

# Comment charger des fichiers ou dossiers vers un compartiment S3 ?

Avant de pouvoir charger des fichiers ou des dossiers vers un compartiment Amazon S3, vous devez disposer d'autorisations en écriture pour le compartiment.

Lorsque vous chargez un fichier vers Amazon S3, il est stocké en tant qu'objet S3. Les objets se composent des données du fichier et des métadonnées décrivant l'objet. Vous pouvez disposer d'un nombre illimité d'objets dans un compartiment.

Vous pouvez charger n'importe quel type de fichier (images, sauvegardes, données, films, etc.) dans un compartiment S3. La console Amazon S3 vous permet de charger des fichiers de 78 Go maximum.

Vous pouvez charger des fichiers en les faisant glisser et en les déposant ou en pointant vers ceux-ci et en cliquant dessus. Pour charger des dossiers, vous devez les faire glisser et les déposer. La fonctionnalité de glisser-déposer est prise en charge uniquement pour les navigateurs Chrome et Firefox.

Lorsque vous chargez un dossier, Amazon S3 charge tous les fichiers et sous-dossiers du dossier spécifié dans le compartiment. Ensuite, il attribue un nom de clé d'objet qui combine le nom du fichier chargé et le nom du dossier. Par exemple, si vous chargez un dossier appelé /images qui contient deux fichiers, sample1.jpg et sample2.jpg, Amazon S3 charge les fichiers et attribue ensuite les noms de clé correspondants, images/sample1.jpg et images/sample2.jpg. Les noms de clé incluent le nom de dossier comme préfixe. La console Amazon S3 affiche uniquement la partie du nom de clé qui suit le dernier signe « / ». Par exemple, dans un dossier images, les objets images/sample1.jpg et images/sample2.jpg s'affichent sous la forme sample1.jpg et sample2.jpg

Pour héberger un site Web, le compartiment doit avoir un accès en lecture public. Il est souhaité que chacun dans le monde dispose d'un accès en lecture à ce compartiment. Pour accorder un accès public en lecture, attachez la stratégie de compartiment suivante au compartiment *example.com* en utilisant le nom de votre compartiment au lieu de *example.com*.

{

"Version":"2012-10-17",

"Statement":[{

"Sid":"PublicReadGetObject",

"Effect":"Allow",

"Principal": "\*",

"Action":["s3:GetObject"],

"Resource":["arn:aws:s3:::example.com/\*"]

}]

}

**Pour configurer vos compartiments pour l'hébergement de site Web**

1. Connectez-vous à AWS Management Console et ouvrez la console Amazon S3 à l'adresse<https://console.aws.amazon.com/s3/>.
2. Dans la liste **Nom du compartiment**, choisissez le nom du compartiment pour lequel vous souhaitez activer l'hébergement de site web statique.
3. Choisissez Propriétés.
4. Choisissez **Hébergement de site Web statique**.
5. Configurez le compartiment *example.com* pour l'hébergement de site Web. Dans la zone **Document d'index**, tapez le nom que vous avez attribué à votre page d'index.

**Pour rediriger les demandes de www.*example.com* vers *example.com***

1. Dans la liste **Compartiments** de la console Amazon S3, choisissez votre compartiment (www.*example.com*, dans cet exemple).
2. Choisissez Propriétés.
3. Choisissez **Hébergement de site Web statique**.
4. Choisissez **Rediriger les demandes**. Dans la zone **Domaine ou compartiment cible**, tapez *example.com*.
5. Choisissez **Save**.

## Étape 3 : ajoutez des enregistrements d'alias pour example.com et www.example.com

Dans cette étape, vous créez les enregistrements d'alias que vous ajoutez à la zone hébergée pour le domaine mappent *example.com* et *www.example.com* vers les compartiments S3 correspondants. Au lieu d'utiliser des adresses IP, les enregistrements d'alias utilisent les points de terminaison de site web Amazon S3. Amazon Route 53 maintient un mappage entre les enregistrements d'alias et les adresses IP où résident les compartiments Amazon S3.

**Pour acheminer le trafic vers votre site web**

**Pour acheminer le trafic vers votre site web**

1. Ouvrez la console Route 53 à l'adresse <https://console.aws.amazon.com/route53/>.
2. Dans le volet de navigation, choisissez **Hosted zones**.

**Note**

Lorsque vous avez enregistré votre domaine, une zone hébergée portant le même nom a été créée automatiquement par Amazon Route 53. Une zone hébergée contient des informations sur la façon dont vous souhaitez qu'Route 53 achemine le trafic pour le domaine.

1. Dans la liste des zones hébergées, choisissez le nom de votre domaine.

d