Chapitre 3

Prérequis

Pour aborder ce chapitre dans de bonnes conditions, il est recommandé d'avoir :

☒Un compte AWS.

☒Des connaissances de base sur l'utilisation du système d'exploitation Linux, y compris les commandes essentielles pour la gestion des fichiers, des processus, et des permissions.

Objectifs

Ce chapitre explore les capacités de calcul offertes par Amazon EC2 (*Elastic Compute Cloud*), un service clé d'AWS pour exécuter des applications à haute disponibilité et performance. Vous apprendrez à configurer, déployer, et gérer des instances EC2, tout en comprenant leurs cas d'utilisation, caractéristiques, et options de tarification. Ce chapitre couvre également les stratégies d'optimisation des coûts, les fonctionnalités avancées, et les bonnes pratiques pour concevoir des architectures scalables et robustes adaptées à différents besoins métier.

L'objectif est de se familiariser avec les options de calcul sur AWS en déployant des instances, tout en explorant les différents types d'instances, modes de déploiement et options de tarification.

Positionnement dans la certification AWS

Ce chapitre est aligné avec les objectifs de la certification AWS Certified Solutions Architect - Associate (SAA-C03), en couvrant les domaines suivants :

- Domain 1 : Design Secure Architectures (30 %)
  - Concevoir des charges de travail et applications sécurisées :
    - Configurer des règles de réseau pour les groupes de sécurité afin de protéger les instances EC2.
    - Segmenter les réseaux à l'aide de sous-réseaux publics et privés pour un accès contrôlé aux ressources.
    - Configurer des adresses IP élastiques pour garantir une connectivité réseau persistante.
- Domain 3 : Design High-Performing Architectures (24 %)
  - Concevoir des solutions de calcul performantes :
    - Sélectionner les types d'instances EC2 optimisés pour différents cas d'utilisation, comme les charges de travail intensives en calcul, en mémoire, ou en stockage.
    - Configurer les instances pour répondre aux besoins spécifiques en termes de performance.
- Domain 4 : Design Cost-Optimized Architectures (20 %)
  - Concevoir des solutions de calcul rentables :
    - Comparer les modèles de tarification (à la demande, Instances Spot, Instances Réservées) pour réduire les coûts.

Identifier les tailles et types d'instances EC2 les plus adaptés aux charges de travail tout en optimisant les dépenses.

Résumé des contenus abordés dans ce chapitre

Nous analyserons et détaillerons les principaux concepts et fonctionnalités d'Amazon EC2, notamment :

- Introduction au calcul sur AWS : Présentation des avantages d'EC2 et des cas d'utilisation, tels que l'hébergement d'applications web et les environnements de test.
- **Composants essentiels d'EC2** :
  - Gestion des paires de clés pour sécuriser l'accès aux instances EC2.
  - Configuration des groupes de sécurité pour contrôler les flux réseau entrants et sortants.
  - Sélection des types et générations d'instances selon les besoins métiers (ex. : instances optimisées pour le calcul ou la mémoire).
  - Utilisation des images de machine Amazon (AMI) pour standardiser les déploiements d'applications.
  - Configuration des adresses IP élastiques pour garantir la connectivité des instances.
  - Exploitation des métadonnées et des données utilisateur pour automatiser la configuration des instances.
- **Cycle de vie des instances** : comprendre les étapes de lancement, arrêt, redémarrage et résiliation des instances EC2.
- **Modèles de tarification** :
  - Comparaison entre la tarification à la demande, les Instances Spot, et les Instances Réservées pour réduire les coûts.
  - Identification des cas d'utilisation appropriés pour chaque modèle.
- **Surveillance et maintenance** :
  - Surveillance des performances des instances via des outils comme Amazon CloudWatch.
- **Fonctionnalités avancées d'EC2** :
  - Présentation des Placement Groups pour optimiser les performances réseau ou la tolérance aux pannes.
- **Cas pratiques** :
  - Déploiement d'une infrastructure dynamique pour French Bakery.
  - Gestion des sauvegardes pour les instances EC2.
  - Résolution de problèmes courants, comme la perte de clé SSH.

A. Introduction au calcul sur AWS

1. Présentation d'Amazon EC2

Amazon EC2 (*Elastic Compute Cloud*) est un service phare d'Amazon Web Services (AWS), lancé en 2006. Il a apporté une révolution dans le cloud computing en offrant une solution de calcul virtuel à la fois flexible et économique. EC2 permet aux utilisateurs de louer des instances virtuelles, adaptant ainsi leurs ressources informatiques selon leurs besoins spécifiques.

Ce service est conçu pour fournir une puissance de calcul adaptable pour tout, des applications d'entreprise aux traitements de données volumineuses. Avec EC2, les utilisateurs bénéficient d'une grande variété de configurations d'instances, ce qui leur permet de choisir la combinaison optimale de CPU, de mémoire et de stockage pour leurs applications. Les utilisateurs peuvent également choisir leur système d'exploitation préféré, y compris Linux, Windows, et macOS. De plus, Elastic Compute Cloud permet l'attachement de solutions de stockage telles qu'EBS, Instance Store ou EFS pour des systèmes compatibles, et FSx pour Windows, offrant une flexibilité accrue en matière de gestion des données.

La simplicité d'utilisation, couplée à un modèle de tarification flexible, rend EC2 particulièrement attrayant pour une large gamme d'entreprises, des start-up innovantes aux grandes multinationales. Ainsi, EC2 s'est imposé comme une solution incontournable dans le domaine du cloud computing, ouvrant de nouvelles perspectives en termes d'accessibilité et de puissance de calcul dans le cloud.

2. Avantages d'EC2

EC2 se distingue par plusieurs avantages clés qui le rendent particulièrement attractif pour une gamme variée d'entreprises :

- **Flexibilité** : EC2 offre une grande variété de configurations d'instances, permettant aux utilisateurs de choisir la meilleure combinaison de CPU, mémoire, stockage et capacités réseau pour leurs besoins spécifiques. Cette flexibilité est cruciale pour adapter les ressources informatiques aux exigences diverses des applications modernes.
- **Évolutivité** : la capacité d'EC2 à s'adapter rapidement à l'évolution des besoins de calcul est un atout majeur. Il est possible d'augmenter ou diminuer leurs ressources avec facilité, ce qui est particulièrement utile pour gérer les variations de charge de travail sans investir dans une infrastructure physique.
- **Modèle de Tarification Économique** : EC2 se distingue par son modèle de tarification qui permet aux entreprises de payer uniquement pour les ressources qu'elles consomment. Les options telles que les instances réservées et les instances spot offrent des moyens supplémentaires de contrôler les coûts, en fonction de la nature et de la durée de l'utilisation.
- **Fiabilité et Sécurité** : bénéficiant de la robustesse de l'infrastructure AWS, EC2 assure une haute disponibilité et des fonctionnalités de sécurité avancées. Les utilisateurs peuvent ainsi compter sur une plateforme sécurisée et fiable pour leurs applications critiques.
- **Intégration avec l'Écosystème AWS** : EC2 fonctionne de manière transparente avec d'autres services AWS, permettant aux entreprises de tirer parti d'un vaste éventail de services complémentaires, de la gestion de bases de données au stockage en passant par l'analyse de données.

B. Architecture et composants d'EC2

1. Paires de clés

Les paires de clés, ou « *Key Pairs* », sont un mécanisme de sécurité permettant l'accès aux instances Amazon EC2. Elles permettent l'authentification et l'assurance que seul le détenteur de la clé privée appropriée peut se connecter aux instances via SSH (*Secure Shell*).

Une paire de clés est constituée de deux clés complémentaires : une clé publique et une clé privée. Lors de la création d'une instance EC2, l'utilisateur a la possibilité de spécifier une clé publique qui sera installée dans l'instance. Cette clé publique est stockée dans le fichier .ssh/authorized_keys de l'instance conformément aux pratiques standard de SSH, permettant à quiconque possède la clé privée correspondante de s'y connecter sécuritairement.

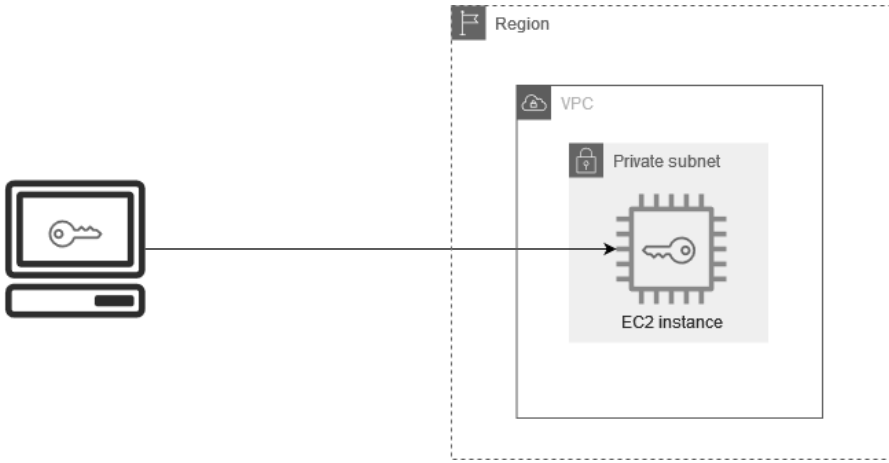
Pour rappel la différence entre clé publique et privée :

- **Clé publique** : installée dans l'instance EC2, elle peut être vue et utilisée par n'importe qui, mais elle ne peut pas être utilisée seule pour accéder à l'instance.
- **Clé privée** : doit être conservée en sécurité par l'utilisateur. Elle est utilisée pour initier une session SSH vers l'instance. La sécurité de l'instance dépend directement de la confidentialité de cette clé privée.

Il est crucial de stocker la clé privée dans un endroit sécurisé. La perte de la clé privée signifie perdre l'accès à l'instance, et si la clé est compromise, cela peut permettre à des utilisateurs non autorisés d'accéder à vos instances. De plus, il ne faut **jamais** partager votre clé privée à des personnes externes. La sécurité de vos instances EC2 dépend de la confidentialité de vos clés privées.

☞*En réalité, il est en fait possible de récupérer l'accès à une instance EC2 dont la clé privée a été perdue. Il suffit de démonter le disque root de l'instance concernée et de le monter sur une autre instance EC2. Une fois monté, vous pouvez accéder au système de fichiers et modifier le fichier .ssh/authorized_keys en ajoutant une nouvelle clé publique. Après cette modification, remontez le disque sur l'instance originale. Cette opération vous permettra de vous reconnecter à l'instance initiale en utilisant la nouvelle clé publique correspondante à la clé privée que vous possédez, restaurant ainsi l'accès sécurisé à l'instance.*

Les paires de clés peuvent être générées par les utilisateurs ou bien par AWS lors de la création d'une nouvelle instance ou lors de la première configuration du compte AWS.



Ce schéma illustre un utilisateur qui utilise une clé privée pour se connecter à une instance EC2. La clé privée, stockée localement sur l'ordinateur de l'utilisateur, permet une authentification sécurisée lors de l'accès à l'instance. Cette méthode garantit que seuls les utilisateurs disposant de la clé privée correspondante peuvent établir une connexion à l'instance.

2. Types et génération d'instances

Les instances EC2 d'Amazon Web Services sont catégorisées selon leurs capacités et leurs optimisations spécifiques, adaptées à diverses charges de travail dans le cloud. Le nom des instances EC2 fournit des informations détaillées sur la configuration de l'instance, incluant sa classe, sa génération, sa taille, et parfois des spécifications supplémentaires :

Prenons par exemple le type d'instance **m5.2xlarge**. Le « m » correspond à la classe de l'instance (Usage général). Le chiffre « 5 » correspond à la cinquième génération de la classe d'instance « m ». Chaque nouvelle génération d'instances EC2 offre généralement des améliorations par rapport à la précédente, comme de meilleures performances, une efficacité énergétique accrue, un meilleur rapport qualité-prix, ou l'accès à de nouvelles technologies sous-jacentes (processeurs, réseaux, etc.). Les améliorations peuvent inclure des processeurs plus récents, plus de mémoire, ou des capacités de stockage améliorées, ce qui permet de soutenir de manière plus efficace des charges de travail plus diverses et plus exigeantes. Enfin, « 2xlarge » correspond à la taille de l'instance indiquant des ressources spécifiques. Plus le nombre est élevé, plus il y aura du CPU ou de RAM ou bien de meilleure spécificité sur l'instance EC2.

Parfois, des lettres supplémentaires après le chiffre de la génération indiquent une spécification particulière :

- **c5d** : « d » signifie que l'instance est équipée de disques SSD locaux pour un stockage à très haute performance.
- **r6g** : « g » indique une instance équipée de processeurs ARM, optimisant ainsi le coût et l'efficacité énergétique.

Voici les différents types d'instances disponibles sur AWS :

- **Usage général (General Purpose, m, t)** : les instances T, comme les t2.micro, sont des instances AWS à usage général particulièrement adaptées aux charges de travail légères et imprévisibles. Elles fonctionnent selon un concept de burst, permettant à l'instance de fournir des performances de base garanties tout en ayant la capacité de gérer des pics de performance CPU lorsque la demande augmente temporairement. Ces pics sont possibles grâce à un système de crédits CPU : l'instance accumule des crédits lorsqu'elle fonctionne en dessous de son niveau de base et les utilise lors des périodes de forte activité.

  Par exemple, une instance t2.micro est idéale pour les serveurs web à faible trafic, les environnements de développement ou encore les petites bases de données. En outre, elle fait partie de l'offre Free Tier d'AWS, offrant jusqu'à 750 heures gratuites par mois, ce qui la rend idéale pour des travaux pratiques ou pour se familiariser avec AWS. Cependant, il est important de noter qu'une utilisation prolongée au-delà des crédits CPU disponibles entraîne une réduction des performances, ce qui rend ces instances moins adaptées aux charges de travail soutenues.

☞*Le fonctionnement des instances T peut être comparé à un athlète : elles accumulent des crédits CPU en période de repos et les utilisent pour des pics de performance, comme un coureur qui alterne entre sprint et récupération. Si les crédits sont épuisés, les performances diminuent, ce qui illustre l'importance des phases de repos pour maintenir un équilibre.*

  En parallèle, les instances de la série m, comme les m6i.large, représentent une autre catégorie d'instances à usage général. Contrairement aux instances T, elles ne reposent pas sur le concept de crédits CPU et offrent des performances constantes. Ces instances sont idéales pour des charges de travail variées telles que les applications d'entreprise, les bases de données relationnelles, les serveurs web de trafic modéré et les environnements de développement ou de test. Grâce à leur équilibre en termes de CPU, mémoire, réseau et stockage, les instances M conviennent aux applications qui nécessitent des performances stables et prévisibles.

  En résumé, les instances T sont économiques et adaptées aux charges de travail irrégulières avec des pics occasionnels, tandis que les instances M offrent une solution plus robuste et polyvalente pour des charges de travail générales nécessitant des performances constantes. Le choix entre ces deux types dépendra des exigences spécifiques de votre application, notamment en termes de stabilité des performances et de coûts.

- **Optimisée pour le calcul (Compute Optimized, c)** : ces instances sont idéales pour des tâches exigeant des processeurs à haute performance, telles que le traitement par lots, le transcodage de médias, les serveurs web à fort trafic, les applications de calcul haute performance (HPC), la modélisation scientifique, l'apprentissage automatique et les serveurs de jeux.
- **Optimisée pour la mémoire (Memory Optimized, r, x)** : les instances r, telles que r5, sont conçues pour des applications de bases de données en mémoire, des systèmes de gestion de bases de données relationnelles et des caches distribués.

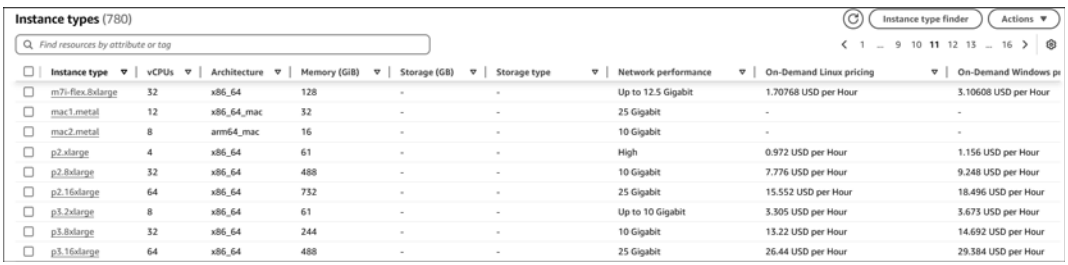
  Les instances x, comme x1, sont parfaites pour des charges de travail encore plus gourmandes en mémoire, incluant le traitement de grandes bases de données, le big data et les applications ERP à grande échelle.
- **Calcul accéléré (Accelerated Computing, p, g, f)** : conçues pour les applications tirant parti de cartes graphiques dédiées, d'unités de traitement tensoriel ou de circuits programmables (FPGA), comme le rendu graphique intensif, l'apprentissage automatique et les simulations scientifiques.

- **Optimisée pour le stockage (Storage Optimized, d, h, i)** : idéales pour les charges de travail qui nécessitent un accès fréquent à des volumes importants de données sur le disque local, telles que les bases de données transactionnelles et les entrepôts de données.
- **Optimisée pour le Calcul Haute Performance (High Performance Computing HPC Optimized, h)** : conçues pour des applications de calcul intensif, ces instances sont utilisées dans des domaines comme la bio-informatique, la simulation de fluides et l'analyse financière.

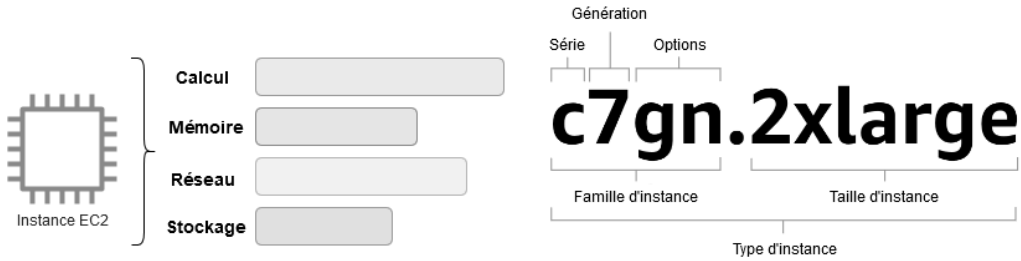
La diversité des instances EC2 permet aux utilisateurs de choisir précisément les ressources adaptées à leurs besoins, optimisant ainsi les performances et les coûts pour leurs applications spécifiques. Voici quelques exemples de différents types d'instances EC2 :

Taille d'instance	vCPU	Mémoire	Stockage d'instance	Bande passante du réseau (Gbit/s)	Bande passante EBS
t2.micro	1	2	EBS uniquement	Faible à modéré	
m5.4xlarge	16	64	EBS uniquement	Jusqu'à 10	4750
m5.8xlarge	32	128	EBS uniquement	10	6800
r5.8xlarge	32	256	EBS uniquement	10	6800
c5ad.large	2	4	1 disque SSD NVMe de 75 Go	Jusqu'à 10	Jusqu'à 3 170

La liste complète de tous les types d'instances se trouve à cette adresse : https://aws.amazon.com/fr/ec2/instance-types/ ou dans la console AWS dans EC2-> Instances -> Instance types. On pourra notamment constater que certains types d'instances ont certaines particularités comme les instances P5 avec du GPU.



Cette capture d'écran présente une liste des types d'instances Amazon EC2 disponibles, avec des informations clés telles que le nombre de vCPU, l'architecture (x86_64, arm64), la mémoire (en GiB), les performances réseau, et les tarifs pour les systèmes d'exploitation Linux et Windows. On peut voir des instances adaptées à différents besoins, comme les familles m7-flex et mac pour des charges générales ou spécifiques, ou encore les séries p2 et p3 optimisées pour des charges intensives en calcul, comme l'apprentissage machine. Les tarifs horaires permettent de comparer rapidement les coûts selon le type d'instance et l'environnement requis.



Ce schéma illustre la nomenclature des types d'instances EC2 d'AWS et leurs composants. Une instance EC2 combine plusieurs ressources comme la capacité de calcul, la mémoire, le réseau et le stockage pour répondre aux besoins d'une application. La classification des types d'instances est détaillée dans leur nom, ici représenté par c7gn.2xlarge. La première lettre (c) indique la famille d'instances, ici optimisée pour le calcul. Le chiffre suivant (7) représente la génération de l'instance. Les options supplémentaires, comme gn, précisent des caractéristiques spécifiques telles que l'optimisation réseau ou GPU. Enfin, le suffixe (2xlarge) désigne la taille de l'instance, qui détermine les ressources attribuées, comme le nombre de vCPU et la mémoire disponible. Cette structure permet une identification rapide et claire des capacités et usages de chaque type d'instance.

3. Lancement d'une instance EC2

Voici les étapes afin de créer une instance EC2 depuis la console AWS :

➢ Rendez-vous dans la section EC2 à cette adresse : https://eu-west-1.console.aws.amazon.com/ec2/home ou depuis la console en recherchant le service EC2.