Rapport :

Hébergement de WordPress sur AWS

1.Introduction

1.1. Contexte Général

Le cloud computing, ou informatique en nuage, représente la fourniture à la demande de ressources informatiques telles que la puissance de calcul, le stockage sur base de données et les applications par le biais d'une plate-forme de services de cloud via Internet. Actuellement, divers fournisseurs de services cloud mettent en place des environnements cloud accessibles aux utilisateurs, qui, souvent, ne sont pas concernés par les détails techniques sous-jacents ou les défis auxquels sont confrontés ces fournisseurs.

Les utilisateurs aspirent à une capacité de calcul adaptable à la demande, sans se préoccuper du nombre de serveurs ou des autres ressources nécessaires pour répondre à leurs besoins informatiques. L'évolution du modèle de cloud computing est marquée par une augmentation significative du nombre d'applications exploitant cette approche, notamment en raison de la diminution des coûts de connectivité et de l'efficacité accrue du matériel informatique, particulièrement à grande échelle.

Les services cloud ne se limitent plus aux applications web, englobant également le stockage de données et l'accès à divers services spécialisés tels que l'Infrastructure en tant que Service (IaaS), la Plateforme en tant que Service (PaaS) et le Logiciel en tant que Service (SaaS). Cette évolution en fait l'environnement idéal pour des applications évolutives, offrant une allocation rapide des ressources en période de forte demande et leur désallocation en période de baisse de la demande.

L'évolutivité des applications peut revêtir différentes formes, mais en essence, l'application et son infrastructure sous-jacente doivent s'adapter dynamiquement à l'évolution des situations, promouvant ainsi la disponibilité et la fiabilité du service tout en minimisant les coûts. Cependant, les fournisseurs de services d'application sont confrontés à des défis liés à la demande imprévisible, notamment lorsque des événements externes entraînent des niveaux de trafic exceptionnels.

Deux approches ont été adoptées pour résoudre l'imprévisibilité du trafic et de la charge système. La première consiste à sur-approvisionner les ressources pour gérer les pics de trafic, bien que cela augmente la disponibilité, il est inefficace en termes d'utilisation des ressources et coûteux.

La seconde approche repose sur le dimensionnement du système pour une utilisation typique, avec la possibilité de provisionner dynamiquement des ressources supplémentaires uniquement lorsque nécessaire, suivie de leur désallocation. Cette approche reflète un véritable paradigme informatique utilitaire où les clients ne paient que pour la période d'utilisation des ressources.

1.2. Objectifs du Projet

Ce projet vise à fournir une architecture de référence pour l'hébergement de WordPress sur AWS.
En utilisant un ensemble de modèles YAML déployables via AWS CloudFormation, cette architecture exploite divers services AWS tels que Amazon VPC, Amazon EC2, Auto Scaling, Elastic Load Balancing (Application Load Balancer), Amazon RDS, Amazon ElastiCache, Amazon EFS.
L'objectif principal est de garantir la scalabilité, la haute disponibilité, la performance, la sécurité, l'efficacité des coûts, et la personnalisation du déploiement, offrant ainsi une solution complète pour héberger un site WordPress sur AWS.

2. Architecture Hautement Évolutive et Hautement Disponible

Dans ce chapitre, nous allons présenter l'architecture globale du projet. L'objectif est de fournir une vue d'ensemble détaillée de la structure et de l'organisation du projet, mettant en lumière les composants clés qui interagissent harmonieusement pour créer une plateforme robuste pour l'hébergement de WordPress sur AWS.
Cette section explorera les différents services AWS utilisés, les relations entre eux, et comment ils contribuent collectivement à l'infrastructure complète du système. En comprenant cette architecture globale, les utilisateurs pourront appréhender la manière dont chaque élément fonctionne en tandem pour assurer la fiabilité, les performances optimales et la flexibilité de l'environnement WordPress sur la plateforme AWS.

2.1. Architecture de Référence

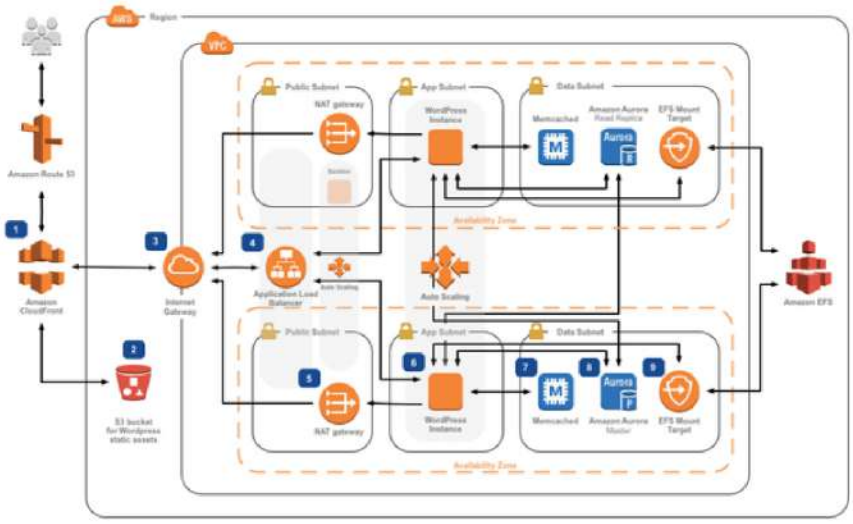

Figure 2.1 Architecture de référence

Pour la plateforme d'hébergement de WordPress sur AWS, nous avons élaboré une architecture évolutive et hautement disponible, déployée initialement sur deux zones de disponibilité, avec la possibilité d'extension en fonction des besoins. Cette architecture est étroitement alignée sur le modèle classique à plusieurs niveaux des applications web.

Au sein de notre conception architecturale, nous distinguons clairement la composante des données, la couche applicative, ainsi que la partie dédiée à l'équilibreur de charge et à la connectivité. Cette approche garantit une gestion efficace des ressources, une résilience accrue et une évolutivité optimale pour répondre aux exigences changeantes de l'environnement d'hébergement **WordPress**.

2.2. Loadbalancer

Dans ce premier segment de notre architecture, nous avons intégré un équilibreur de charge applicatif qui joue un rôle crucial dans la distribution du trafic applicatif entrant vers diverses cibles du deuxième segment. Cet équilibreur de charge surveille en permanence l'état des cibles enregistrées, ne dirige le trafic qu'en direction des cibles saines et transmet les informations sur l'état des cibles aux groupes responsables de l'élasticité. Cette approche garantit une gestion efficace du trafic, assurant une disponibilité optimale du système.

Nous avons également mis en place une passerelle de traduction d'adresses réseau (NAT) par zone de disponibilité, permettant aux

deuxième segment de se connecter à Internet ou à d'autres services AWS. Cependant, cette configuration empêche toute tentative de connexion initiée depuis Internet en direction de ces instances, renforçant ainsi la sécurité du système.

Pour renforcer davantage la sécurité de notre architecture, chaque sous-réseau public est équipé d'un hôte bastion Linux, doté d'une adresse statique (IP Elastic). Cette configuration autorise un accès sécurisé via Secure Shell (SSH) aux instances EC2 présentes dans les sous-réseaux publics et privés. Ainsi, nous assurons un contrôle d'accès rigoureux tout en facilitant la gestion et la maintenance des instances déployées dans notre infrastructure.

2.3.Partie Base de données

Au sein de notre projet, nous prévoyons d'intégrer une base de données MySQL essentielle pour le stockage et la gestion des données. Afin d'assurer une disponibilité optimale et une résilience accrue, cette base de données sera déployée dans deux zones de disponibilité distinctes, à savoir us-east-1a et us-east-1b. Positionnée dans un sous-réseau privé dédié baptisé "data subnet", la base de données sera ainsi accessible uniquement au sein de notre réseau privé, renforçant la sécurité des données contre tout accès non autorisé externe.

Pour permettre des connexions sortantes vers Internet, nécessaires par exemple pour les mises à jour logicielles, nous avons intégré une instance NAT (Network Address Translation) agissant comme passerelle pour ces connexions sortantes tout en garantissant une protection contre les connexions entrantes non sollicitées depuis Internet vers la base de données. Cette approche globale vise à garantir une haute disponibilité, une sécurité accrue, et un contrôle précis des connexions sortantes vers Internet, assurant ainsi un environnement de stockage et de gestion de données performant et fiable.

3. Description Détaillée des Composants de l'Architecture Proposée

3.3.Réseau virtuel privé (VPC)

Amazon VPC (Virtual Private Cloud) joue un rôle central en offrant une solution de cloud computing qui nous permet de provisionner une section isolée et sécurisée du cloud privé AWS. Cette section, logiquement distincte des réseaux traditionnels et des espaces réseau des autres clients d'AWS, nous permet de déployer nos ressources AWS au sein d'un réseau virtuel dédié. Amazon VPC nous accorde une totale maîtrise sur

notre environnement réseau virtuel, nous autorisant à choisir notre propre plage d'adresses IP, à créer des sous-réseaux spécifiques, ainsi qu'à configurer des tables de routage et des passerelles réseau selon nos besoins spécifiques. Cette approche nous confère une flexibilité et un contrôle inégalés sur notre infrastructure cloud, garantissant un déploiement personnalisé et sécurisé de nos ressources.

3.4.Passerelle Internet (Internet Gateway)

La passerelle Internet, intégrée à notre VPC, constitue un composant redondant, hautement disponible et dimensionné horizontalement. Elle facilite la communication sans contraintes de bande passante entre les instances dans notre VPC et Internet, éliminant ainsi tout risque de perturbation de disponibilité.

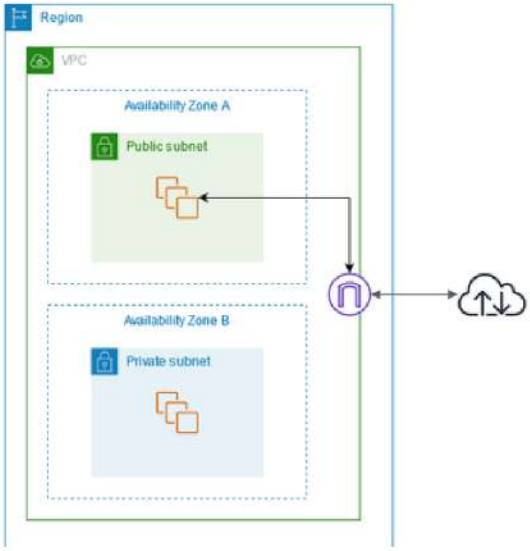

Figure 3.4 Architecture du VPC et de la passerelle Internet

3.5. Passerelle NAT (NAT Instance)

Une instance NAT (Network Address Translation) sera déployée dans le sous-réseau public des deux zones de disponibilité. Cette instance jouera un rôle crucial en facilitant la traduction des adresses IP privées de nos instances, notamment celles de WordPress et de la base de données MySQL, en une adresse IP publique et vice versa.

Elle agira comme une passerelle pour permettre à ces instances, situées dans des sous-réseaux privés, de communiquer avec Internet, assurant ainsi le routage sécurisé du trafic sortant et empêchant les connexions entrantes non sollicitées.

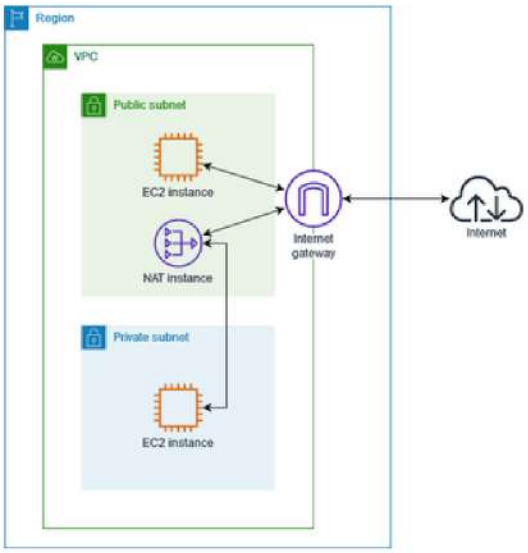

Figure 3.5 Architecture du VPC et la passerelle NAT

3.5. Le cas dans notre projet

Dans notre projet, nous allons intégrer une instance NAT (Network Address Translation). Cette instance, bien qu'apparemment ordinaire, jouera un rôle clé en agissant comme une passerelle NAT.
Elle permettra au trafic Internet de transiter par elle, et avec l'assistance des groupes de sécurité, assurera un cheminement optimal et sécurisé.

3.6.Equilibreur de charge (ELB)/(ALB)

Dans notre cas, nous allons déployer un équilibreur de charge élastique (Elastic Load Balancer) qui surveillera à la fois le trafic HTTP et HTTPS.

Son rôle sera de distribuer équitablement ce trafic entre différentes cibles dans notre VPC, telles que des instances Amazon EC2, des conteneurs, et des adresses IP. Cette approche garantira une répartition optimale de la charge, assurant ainsi que notre application reste toujours réactive et rapide, même face à des variations de trafic.

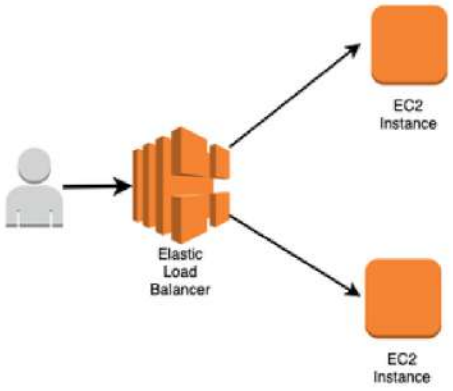

Figure 3.6 Fonctionnement de ELB

3.7.Bastion host SSH

Les deux instances sont privées et ont uniquement la possibilité de se connecter à Internet via l'instance NAT. Par conséquent, un accès direct à ces instances n'est pas possible. Pour remédier à cela, nous introduisons un hôte bastion, une instance à laquelle nous pouvons accéder à partir des instances privées.
Nous avons déployé deux hôtes bastion, chacun situé dans une zone de disponibilité différente, assurant ainsi une redondance et une disponibilité élevée pour la gestion sécurisée des connexions SSH vers nos instances privées.

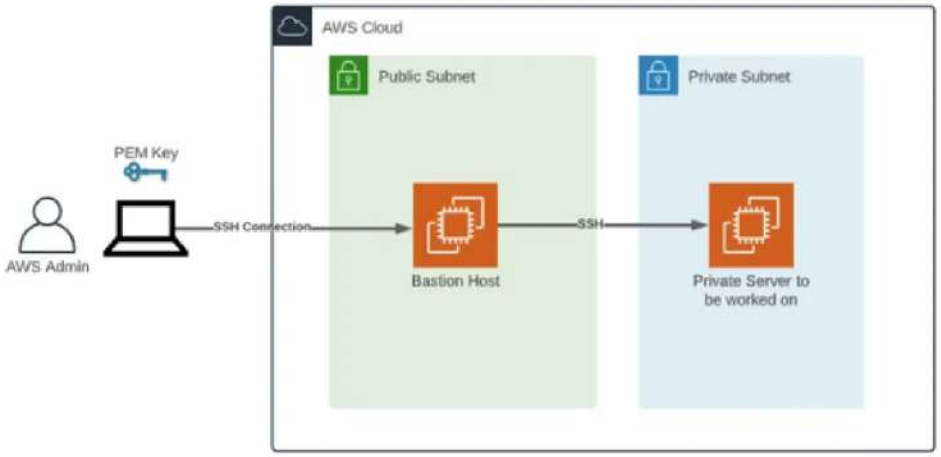

Figure 3.7 Bastion host SSH

3.8.Groupe d'autoscaling (AG)

Dans notre projet, les groupes d'autoscaling jouent un rôle crucial dans la gestion des applications, en ajustant automatiquement la capacité pour maintenir des performances constantes et prévisibles de manière optimale sur le plan des coûts. Ces groupes assurent une élasticité dynamique en fonction des besoins de charge de travail, garantissant ainsi

une expérience utilisateur fluide tout en optimisant les ressources et les coûts associés.

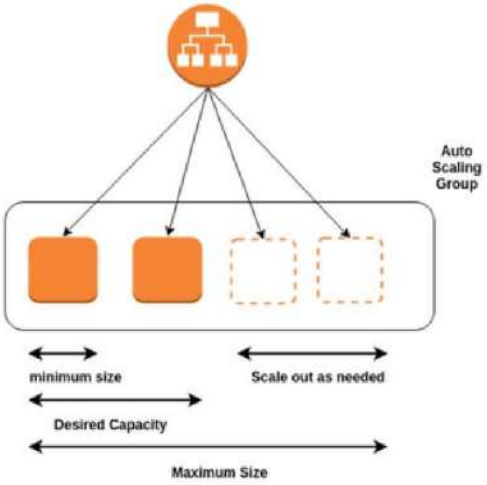

Figure 3.8 Le fonctionnement du AG

3.8.1.Le cas dans notre projet

Dans notre projet, le groupe d'autoscaling fonctionne de manière proactive pour garantir la disponibilité continue de nos deux instances dans les deux zones de disponibilité. À intervalles réguliers, il vérifie l'état de santé de ces instances.

Si l'une d'entre elles, voire les deux, est identifiée comme hors service, le groupe d'autoscaling prend automatiquement des mesures correctives. Il crée alors une nouvelle instance similaire en utilisant une image Amazon Machine Image (**AMI**) préalablement configurée.

Cette approche proactive garantit que notre application reste robuste et toujours opérationnelle, offrant ainsi une expérience utilisateur fiable et sans interruption.

3.9.Capacité de calcul évolutive (EC2)

Amazon EC2 constitue le service fondamental qui nous offre une capacité de calcul sécurisée et adaptable dans le cloud. Grâce à Amazon EC2, nous sommes en mesure de provisionner et de lancer rapidement des instances de serveurs, puis d'ajuster leur capacité en fonction des fluctuations de nos besoins en calcul.
Cette flexibilité nous permet de maintenir une infrastructure

informatique dynamique, garantissant ainsi une réponse agile à nos exigences opérationnelles changeantes.

3.9.1.Le cas dans notre projet

les instances EC2 joueront un rôle central en hébergeant divers composants de notre application. Elles seront responsables de l'exécution de l'instance WordPress, du bastion host, de la base de données (MySQL) et de l'instance NAT.

3.10. Service de Base de données Relationnelle (RDS)

C'un service géré de bases de données relationnelles d'AWS. Cela simplifie la configuration, l'exploitation et la mise à l'échelle des bases de données relationnelles.
Amazon RDS automatise les tâches administratives courantes, permettant ainsi aux développeurs de se concentrer davantage sur le développement d'applications, tout en réduisant la charge liée à la gestion de l'infrastructure de base de données.

3.10.1.Le cas dans notre projet

Dans notre projet, nous utiliserons Amazon RDS pour établir une liaison avec l'instance WordPress, permettant une configuration réussie de la base de données. Cela simplifiera la gestion de la base de données relationnelle et assurera une intégration fluide avec l'application WordPress.

3.11.Service de stockage de fichiers (S3)

un compartiment Amazon **S3** est une ressource de stockage cloud accessible dans la plateforme Amazon Simple Storage Service (S3) d'Amazon Web Services (AWS). Il offre un stockage basé sur des objets, où les données sont stockées à l'intérieur de compartiments S3 sous forme d'unités distinctes appelées objets plutôt que de simples fichiers.

3.11.1.Le cas dans notre projet

Nous allons utiliser Amazon S3 pour stocker les médias, en faisant de ce service un moyen de sauvegarde au cas où nous perdions nos

médias de site.

4. Implémentation de l'Architecture

4.1.1.Création du VPC

Article	Gamme CIDR	IP utilisables	Description
VPC	10.0.0.0/16	65,536	Toute la gamme utilisée pour le VPC et tous les sous-réseaux
Web Subnet	10.0.0.0/22	1022	Sous-réseau privé dans la première zone de disponibilité(us-east-1a)
Web Subnet	10.0.4.0/22	1022	Sous-réseau privé dans la première zone de disponibilité(us-east-1b)
Data Subnet	10.0.100.0/22	254	Sous-réseau privé dans la première zone de disponibilité(us-east-1a)
Data Subnet	10.0.101.0/22	254	Sous-réseau privé dans la première zone de disponibilité(us-east-1b)

Public Subnet	**10.0.201.0/24**	**254**	**Sous-réseau public dans la première zone de disponibilité(us-east-1a)**
Public Subnet	**10.0.202.0/24**	**254**	**Sous-réseau public dans la première zone de disponibilité(us-east-1b)**

4.1.2.Création de groupes de sécurité

Dans notre configuration, nous avons défini plusieurs groupes de sécurité pour garantir un contrôle précis du trafic entre les composants de notre infrastructure.

1. BastionSecurityGroup :
- Description : Groupe de sécurité pour les instances bastion.
- Règles d'entrée : Autorise le trafic TCP sur le port 22 (SSH) depuis la plage CIDR spécifiée pour l'accès aux instances bastion.

2. DatabaseSecurityGroup :
- Description : Groupe de sécurité pour le cluster Amazon RDS.
- Règles d'entrée : Autorise le trafic TCP sur le port 3306 (MySQL) depuis le groupe de sécurité associé à WebSecurityGroup.

3. PublicAlbSecurityGroup :
- Description : Groupe de sécurité pour le répartiteur de charge d'applications (ALB).
- Règles d'entrée : Autorise le trafic TCP sur le port 443 (HTTPS) et le port 80 (HTTP) depuis n'importe quelle adresse IP (0.0.0.0/0).

4. WebSecurityGroup :
- Description : Groupe de sécurité pour les instances web.
- Règles d'entrée : Autorise le trafic TCP sur le port 80 (HTTP) et le port 443 (HTTPS) depuis le groupe de sécurité associé à PublicAlbSecurityGroup. Autorise également le trafic TCP sur le port 22 (SSH) depuis le groupe de sécurité associé à BastionSecurityGroup.

4.1.3.Création de la base de données

Nous créons une base de données en utilisant Amazon RDS, en optant pour le moteur MySQL. Cette base de données sera placée dans le sous-réseau de données (data subnet).

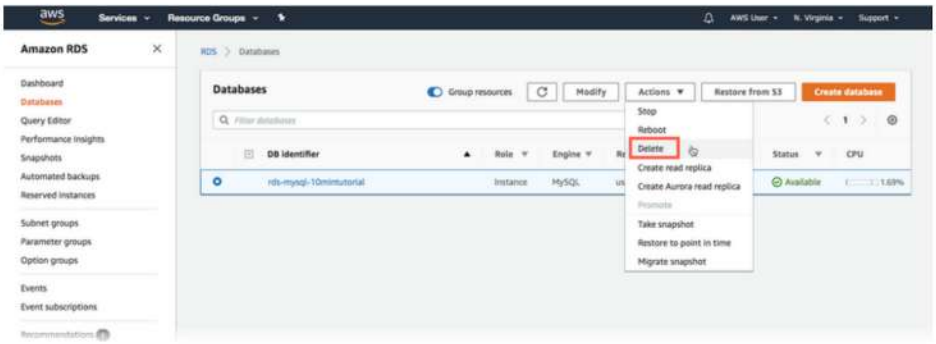

Figure 4.1.3 Création de la base de données

4.1.4.Création de l'instance master

Nous avons créé une instance Amazon avec les paramètres par défaut, installée dans le sous-réseau public de la première zone de disponibilité.
Ensuite, nous avons accédé à la console de cette instance et tenté d'installer une instance Wordpress.

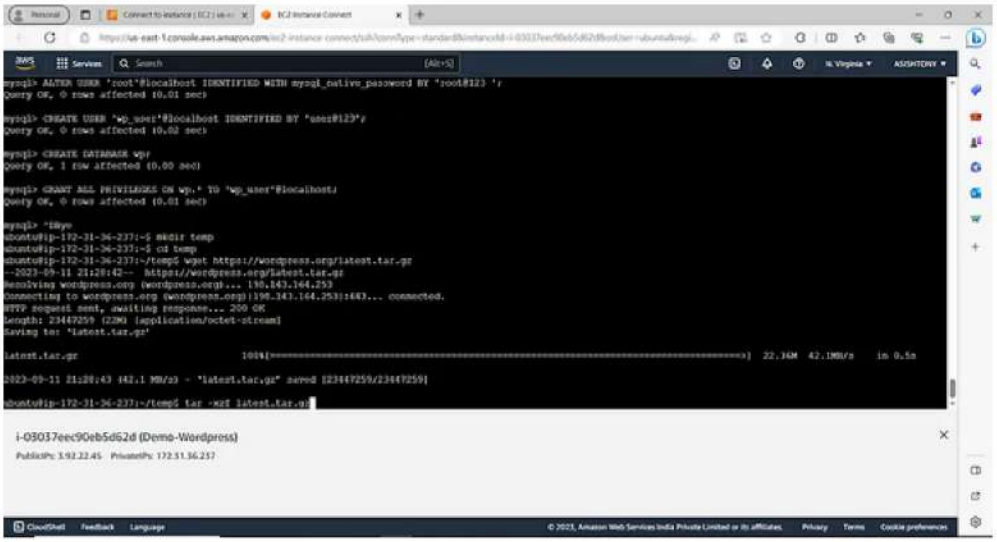

Figure 4.1.4 Installation de wordpress

Après cela, nous avons installé le serveur Apache2 et configuré le fichier de configuration de Wordpress avec les informations de notre base de données.

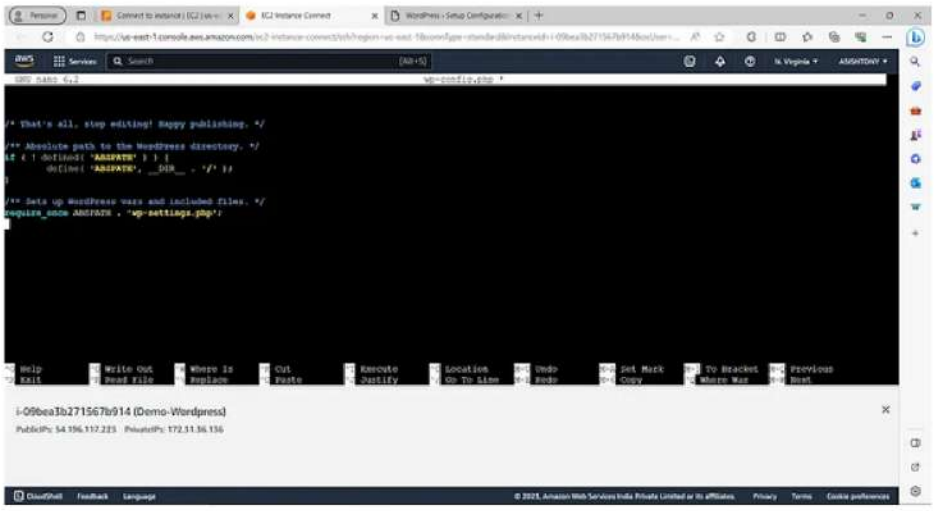

Figure 4.1.4.1 Configuration de wordpress

Enfin, nous avons activé Amazon **EFS** (Elastic File System) et l'avons monté directement sur le chemin d'accès où les fichiers multimédias sont stockés.

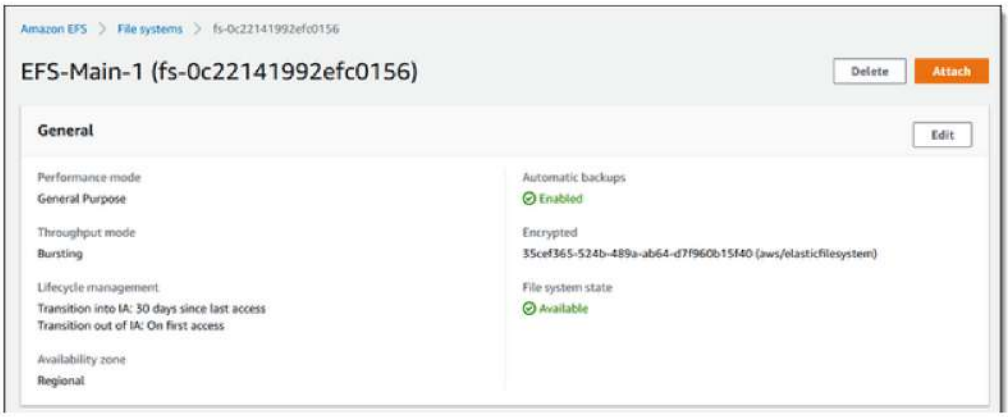

Figure 4.1.4.2 Création de EFS

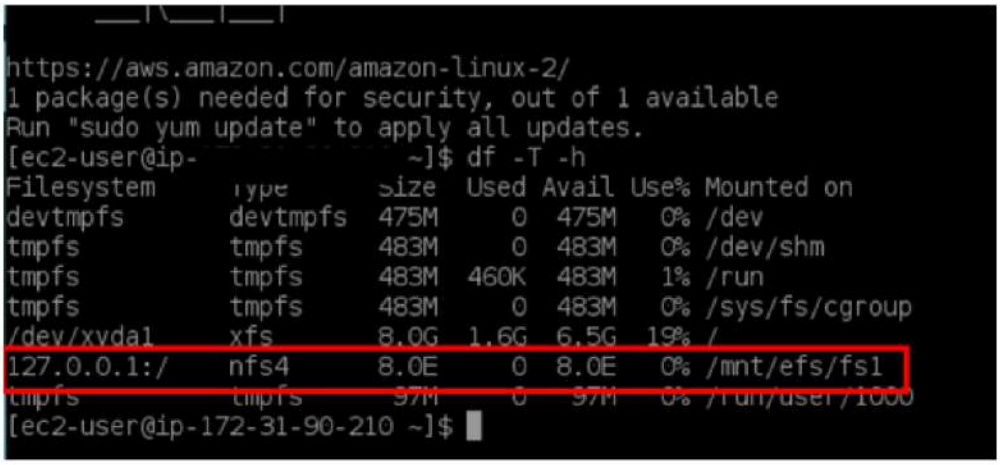

Figure 4.1.4.3 Monter la EFS sur l'instance master

4.1.5.Création de l'AMI d'apartir l'instance master

Après la configuration de WordPress sur l'instance principale, nous allons créer une image à partir de cette instance.

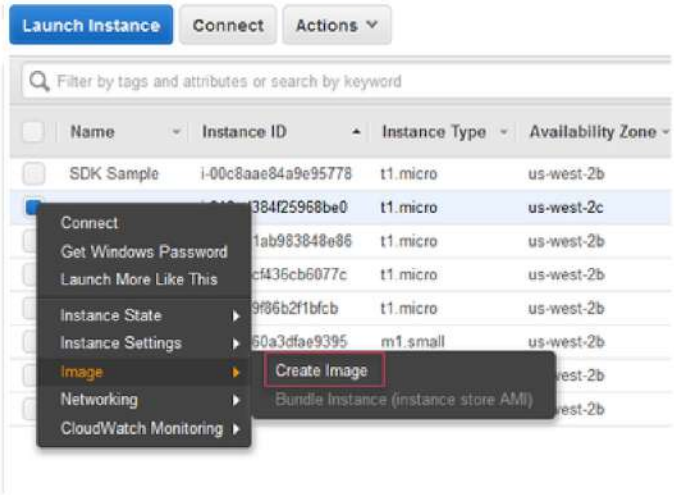

Figure 4.1.5 Initialisation de l'AMI

4.1.5.Configuration et lancement du groupe autoscaling et le Bastion

Nous allons créer un groupe d'auto-scaling avec une configuration de lancement qui utilisera l'AMI que nous avons créé précédemment, et le placer dans le sous-réseau d'application avec un minimum de deux instances et un maximum de deux également.
Ainsi, nous aurons toujours deux instances opérationnelles en permanence.

5. Conclusion

En conclusion, la conception de notre infrastructure cloud pour le projet présente plusieurs avantages et caractéristiques clés.
En utilisant Amazon VPC, nous avons établi un environnement isolé et contrôlable logiquement, permettant le lancement de ressources AWS dans un réseau virtuel personnalisé.

L'utilisation de passerelles Internet et de passerelles NAT dans des sous-réseaux publics et privés assure une connectivité sécurisée avec Internet tout en protégeant nos instances.
Les groupes d'autoscaling contribuent à maintenir des performances constantes et rentables en ajustant automatiquement la capacité en fonction des besoins de calcul. L'usage d'une base de données Amazon RDS simplifie la gestion des bases de données relationnelles, et l'intégration avec WordPress facilite la configuration de notre application. Enfin, l'utilisation d'Amazon S3 pour le stockage des médias assure la disponibilité et la sauvegarde des données.

Cependant, il est crucial de surveiller attentivement les coûts, la sécurité et les performances pour optimiser l'efficacité de notre infrastructure cloud.