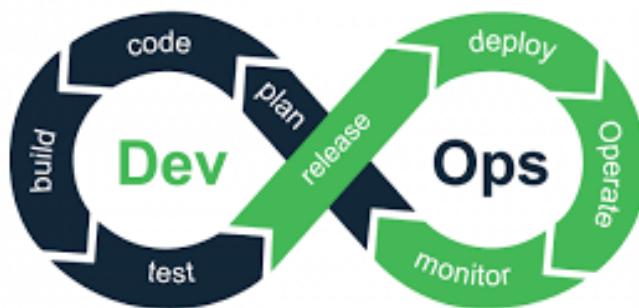


Delivery Management, DevOps et Pipeline



Kevin NAKKOUR

Samy SKIKER

Ryan SEBBANE

Yidir OUAHIOUNE

Laurent YU

Anatole PINCEMAIL

GROUPE 2 RS3

Table des matières

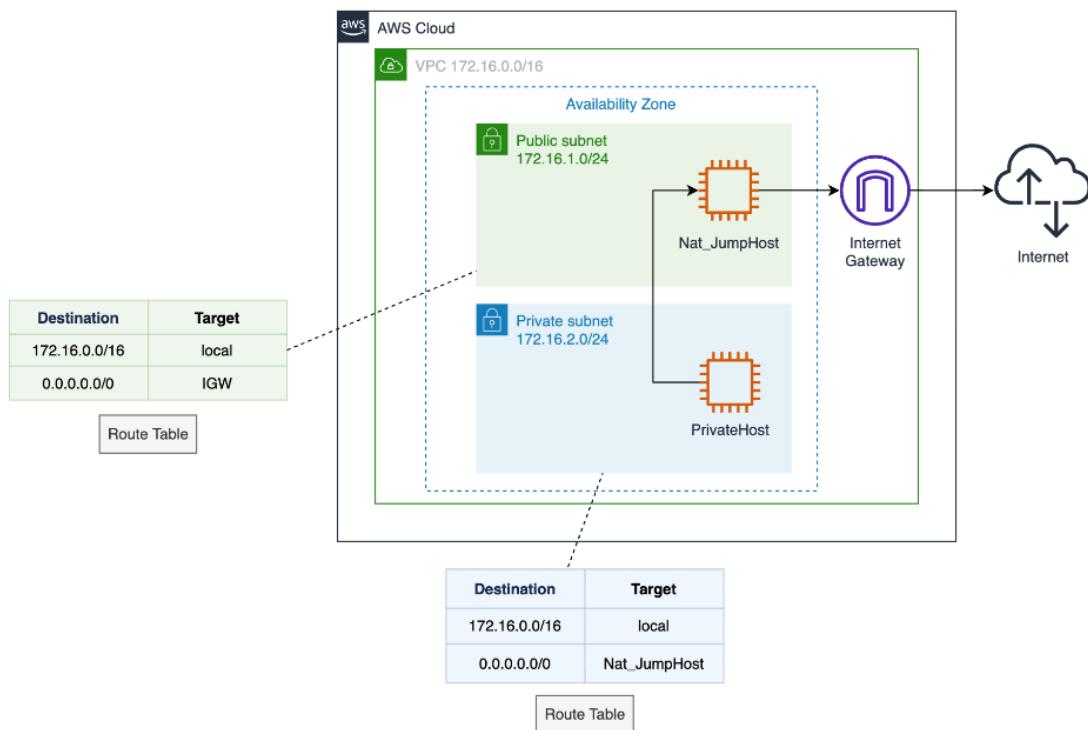
| | |
|--------------------------------------|-----------|
| TP1 Premier pas sur AWS | 3 |
| Introduction | 3 |
| Mise en place du réseau | 5 |
| Création des instances EC2..... | 9 |
| Tests | 10 |
| Instance PrivateHost..... | 11 |
| Questions | 12 |
| TP2 | 14 |
| TP3 | 33 |

TP1 Premier pas sur AWS

Introduction

Le schéma illustre un réseau cloud AWS configuré avec un VPC de plage d'adresses 172.16.0.0/16, divisé en sous-réseaux public et privé. Le sous-réseau public (172.16.1.0/24) comprend un Nat_JumpHost, permettant l'accès à Internet via la passerelle Internet, tandis que le sous-réseau privé (172.16.2.0/24) contient un PrivateHost qui utilise le Nat_JumpHost pour se connecter à Internet. Deux tables de routage sont définies pour diriger le trafic localement au sein du VPC ou vers l'Internet via le Nat_JumpHost ou la passerelle Internet, assurant ainsi une gestion efficace du trafic et de la sécurité au sein de l'infrastructure cloud.

Architecture :



Nous avons bien accès à notre plateforme AWS :

The screenshot shows the AWS Academy Learner Lab interface. On the left is a sidebar with navigation links: Account, Modules (selected), Discussions, Grades, Courses, Calendar, Inbox, History, and Help. The main area has a header "AWS.Used \$0 of \$100 03:37" and a toolbar with "Start Lab", "End Lab", "AWS Details", "Readme", and "Reset". Below the toolbar is a terminal window showing a command-line session. To the right of the terminal is a "Cloud Labs" section with session details: "Remaining session time: 03:59:46(240 minutes)", "Session started at: 2024-01-19T06:33:11-0800", "Session to end at: 2024-01-19T10:33:11-0800", and "Accumulated lab time: 00:00:00 (0 minutes)". Buttons for "SSH key", "Download PEM", "Download PPK", "AWS SSO", and "Download IPA" are visible.

The screenshot shows the AWS Console Home page. It features a "Recently visited" section with a placeholder message "No recently visited services" and links to EC2, S3, RDS, and Lambda. Below this is a "View all services" button. To the right is an "Applications" section with a "Create application" button, currently set to the "us-east-1 (Current Region)". It displays a message "No applications" and a "Create application" button. At the bottom of this section is a "Go to myApplications" link. Other sections include "Welcome to AWS" (Getting started with AWS), "AWS Health" (Open issues 0, Past 7 days), and "Cost and usage" (Current month costs \$0.00, Total costs per month No cost data available).

Lien de partage du LAB 1 (Kévin) :

Mise en place du réseau

- Création du VPC :

The screenshot shows the 'Create VPC' wizard in the AWS VPC console. The 'VPC settings' section is displayed, with 'VPC only' selected. A name tag 'TP_DevOps' is assigned. The IPv4 CIDR block is set to '172.16.0.0/16'. Under 'Tags', a key 'Name' is mapped to 'TP_DevOps'. The 'Create VPC' button is at the bottom right.

- Création des sous-réseaux :

The screenshot shows the 'Subnets' page in the AWS VPC console. It lists several subnets under the 'TP_DevOps' VPC. The subnets are: 'TP_DevOps_Public' (status: Available), 'subnet-0d583b0cc27beca26' (status: Available), 'subnet-03b4a99d1721e8c3d' (status: Available), 'vpc-07402f50859e9ee5e' (status: Available), 'vpc-0c0eb4e1b8adc7c96 | TP_D...' (status: Available), and 'vpc-0c0eb4e1b8adc7c96 | TP_D...' (status: Available). A 'Select a subnet' button is visible at the bottom left.

- Création de l'Internet Gateway :

Internet gateway ID: igw-0c37c35d29d1e0bd3

State: Detached

VPC ID: vpc-0c0eb4e1b8adc7c96 | TP_DevOps

Owner: 884149007211

Tags:

- Name: TP_DevOps

- Rattachement au VPC :

Internet gateway ID: igw-0c37c35d29d1e0bd3

State: Attached

VPC ID: vpc-0c0eb4e1b8adc7c96 | TP_DevOps

Owner: 884149007211

Tags:

- Name: TP_DevOps

- Modification du « Route table » par défaut

Route table ID: rtb-0a55291187045b10c | TP_DevOps_Public was created successfully.

Route table ID: rtb-0a55291187045b10c

VPC: vpc-0c0eb4e1b8adc7c96 | TP_DevOps

Main: No

Explicit subnet associations: -

Edge associations: -

Routes (1):

| Destination | Target | Status | Propagated |
|---------------|--------|--------|------------|
| 172.16.0.0/16 | local | Active | No |

- « Route table » publique

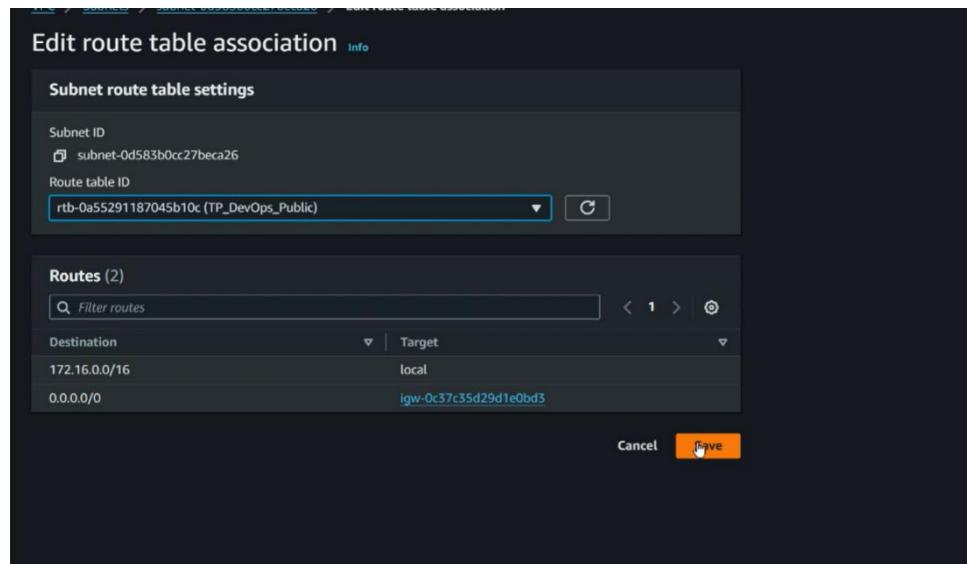
The screenshot shows the AWS VPC Route Tables console. A success message at the top states: "Route table rtb-0a55291187045b10c | TP_DevOps_Public was created successfully." Below this, the route table details are shown: Route table ID: rtb-0a55291187045b10c, Main: No, VPC: vpc-0c0eb4e1b8adc7c96 | TP_DevOps, Owner ID: 884149007211. The "Routes" tab is selected, showing one route: Destination 17.2.16.0/16, Target local, Status Active, Propagated No.

- Ajout de la route :

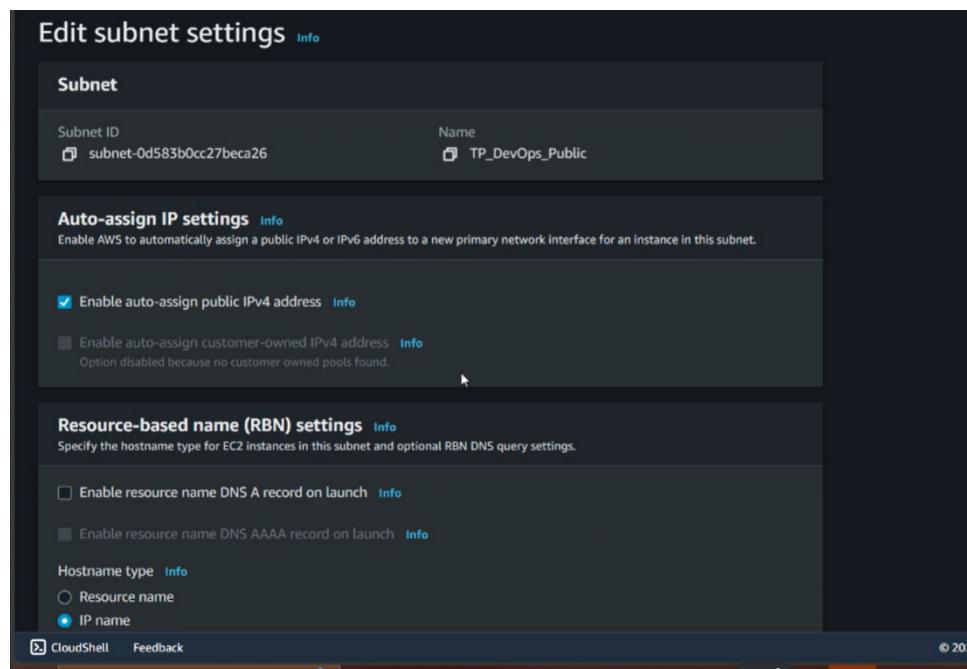
| Destination | Target |
|-------------|------------------------------|
| 0.0.0.0/0 | Internet Gateway : TP_DevOps |

The screenshot shows the AWS VPC Route Tables console with an update message: "Updated routes for rtb-0a55291187045b10c / TP_DevOps_Public successfully". The route table details remain the same as the previous screenshot. The "Routes" tab now shows two routes: Destination 0.0.0.0/0, Target igw-0c37c35d29d1e0bd3, Status Active, Propagated No; and Destination 17.2.16.0/16, Target local, Status Active, Propagated No.

- Attacher la route table « TP_DevOps_Public » au subnet « TP_DevOps_Public »

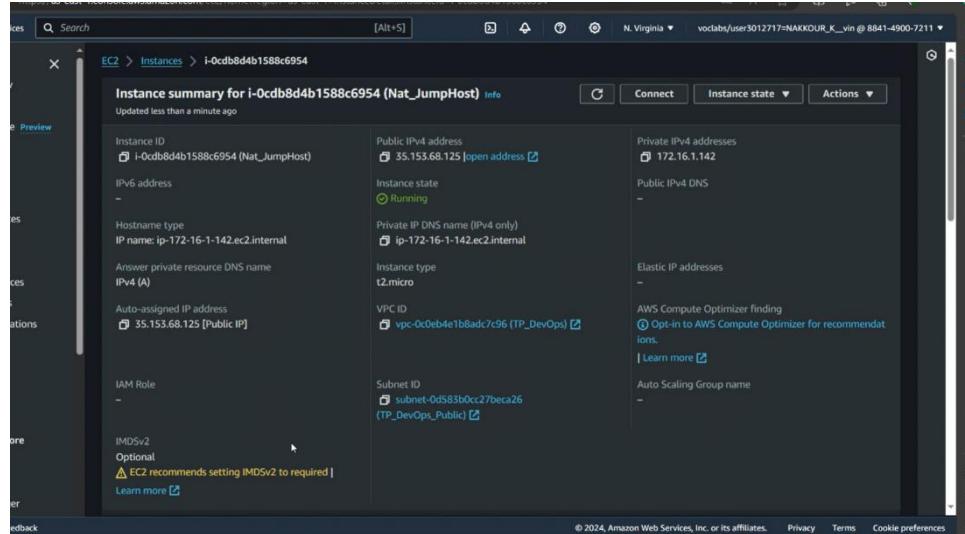


- Activer l'assignement automatique d'adresses IP publique dans le subnet public.

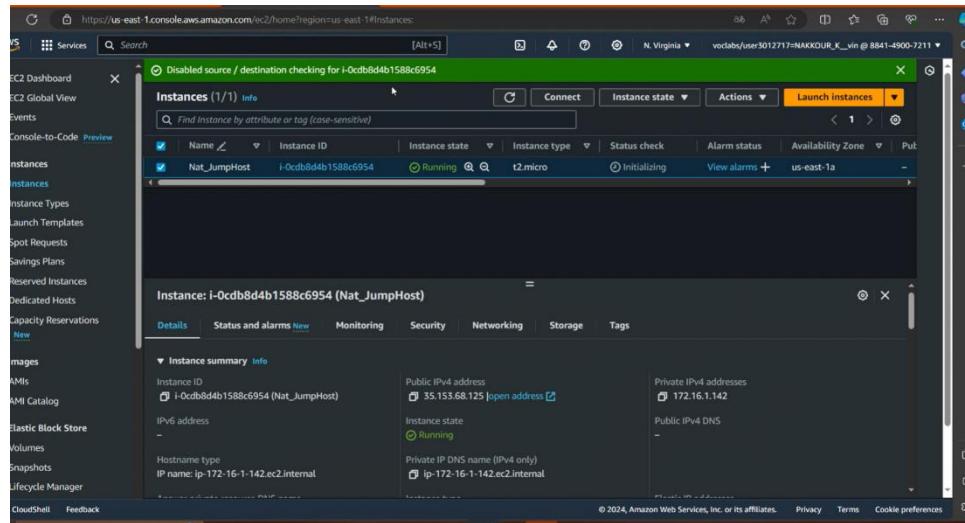


Création des instances EC2

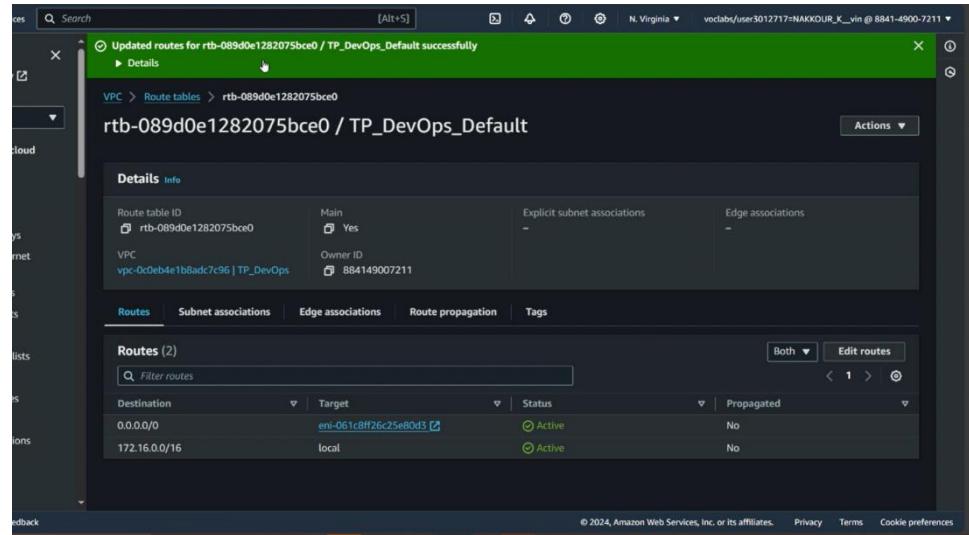
- Instance Nat_JumpHost



- Désactiver source/destination check de l'instance depuis la console. (En effet sans la désactivation de cette protection, il ne serait pas possible d'utiliser notre instance en tant que NAT)



- Maintenant, ajouter la route vers « Nat_JumpHost » dans la « main route table ». Cela va permettre aux instances qui seront déployées d'atteindre Internet par l'intermédiaire de cette « NAT Instance ».



Tests

```

[1] 10:00:27.026 0x152-10 25 /home/noboxterm/Desktop> ssh -i "vockey.pem" ubuntu@34.235.131.77
Warning: Permanently added '34.235.131.77' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1017-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support:   https://ubuntu.com/advantage

 System information as of Fri Jan 19 15:52:34 UTC 2024

 System load: 0.361328125  Processes:          101
 Usage of /: 20.6% of 7.57GB  Users logged in:    0
 Memory usage: 20%           IPv4 address for eth0: 172.16.1.142
 Swap usage: 0%

 Expanded Security Maintenance for Applications is not enabled.
 0 updates can be applied immediately.

 Enable ESM Apps to receive additional future security updates.
 See https://ubuntu.com/esm or run: sudo pro status

 The list of available updates is more than a week old.
 To check for new updates run: sudo apt update

 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.

 /usr/bin/xauth: file /home/ubuntu/.Xauthority does not exist
 To run a command as administrator (user "root"), use "sudo <command>".
 See "man sudo_root" for details.

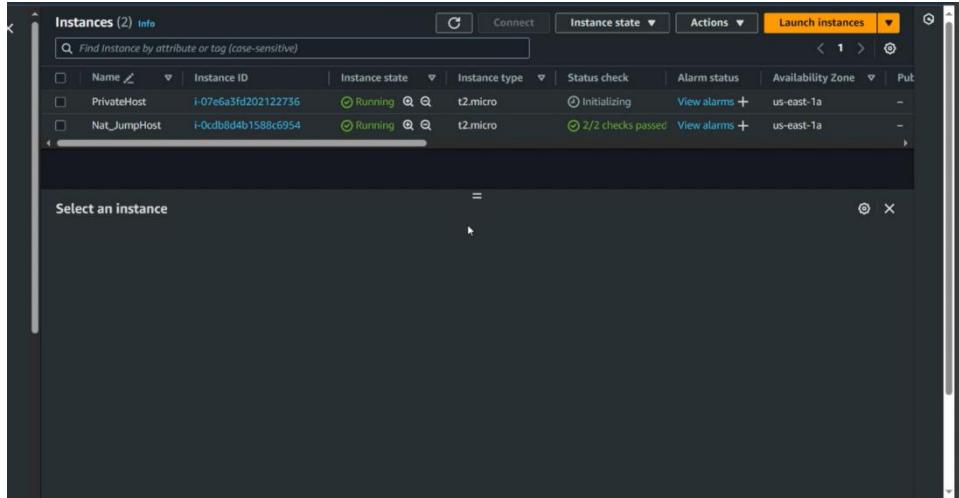
ubuntu@ip-172-16-1-142:~$ 

```

Cela fonctionne.

Instance PrivateHost

- Lancer une EC2 que nous nommerons « PrivateHost » dans le private subnet « TP_DevOps_Private »



Tests

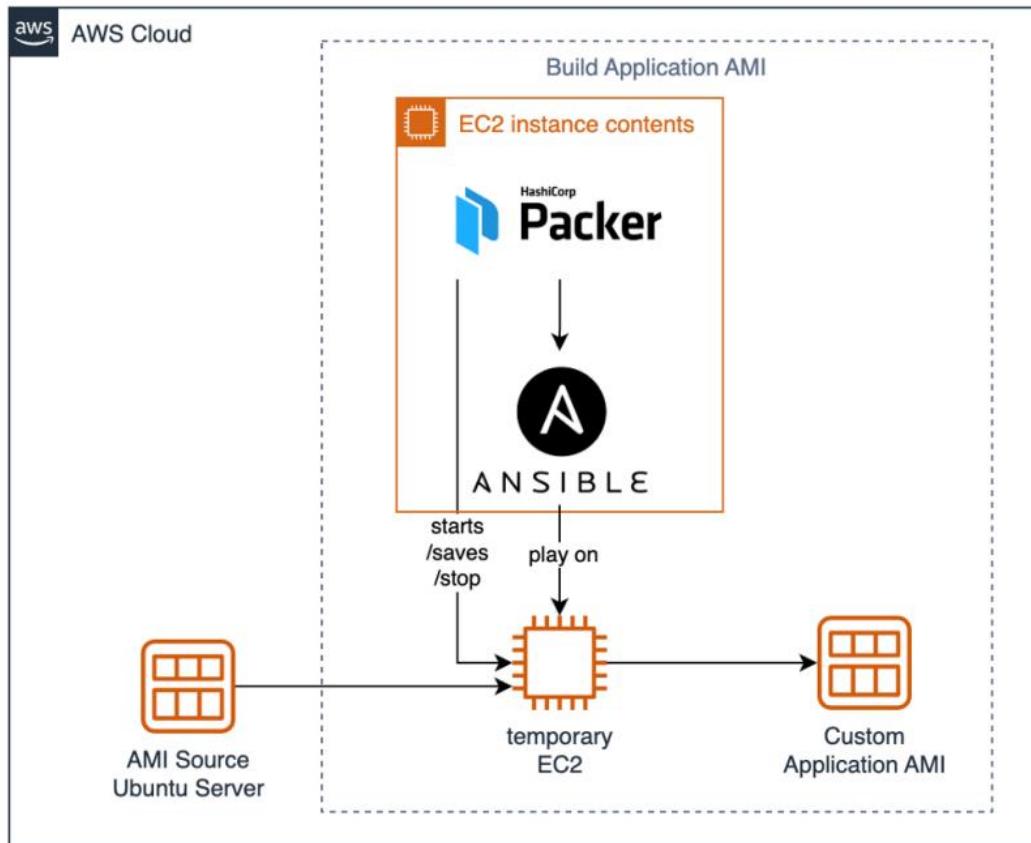
Malheureusement nous ne pouvons pas tester car il y'a un problème côté AWS.

Questions

- A quoi correspond une région AWS ?
 - Une région AWS désigne un lieu géographique précis où AWS met à disposition des infrastructures de cloud computing. Ces régions sont subdivisées en plusieurs zones de disponibilité, qui se réfèrent à des groupes de centres de données physiques séparés mais reliés au sein de la même région.
- Qu'est-ce qu'une « Availability Zone » ?
 - Il s'agit d'un ou plusieurs centres de données équipés de systèmes d'alimentation, de réseau et de refroidissement redondants, situés dans une région AWS. L'architecture des régions AWS inclut plusieurs de ces zones pour permettre le développement d'applications flexibles et disponibles en distribuant les charges à travers ces zones.
- Quel élément réseau est nécessaire pour que les instances d'un réseau privé puissent communiquer vers internet pour par exemple récupérer les packages et updates ?
 - Pour qu'une instance dans un réseau privé puisse accéder à Internet, par exemple pour télécharger des mises à jour, il est indispensable d'utiliser un Gateway de traduction d'adresse réseau (NAT Gateway) ou une instance NAT. Ce système permet aux instances de démarrer des connexions vers l'extérieur tout en bloquant les accès directs entrants depuis Internet.
- Sur AWS quelles sont les propriétés qui caractérisent un « Public Subnet » ?
 - Un sous-réseau public se distingue par la capacité de ses instances à être accessibles directement depuis Internet. Il doit contenir une route vers un Internet Gateway (IGW) dans sa table de routage, permettant la communication externe. Les instances au sein de ce sous-réseau nécessitent une adresse IP publique ou Elastic pour recevoir du trafic Internet entrant.
- Qu'est-ce qu'une « Default Route » Table ?
 - Cette table de routage est générée automatiquement par AWS pour chaque VPC (Virtual Private Cloud). Elle comprend des règles qui orientent par défaut le trafic réseau, définissant ainsi les chemins que les données suivent au sein du VPC.
- Les « Security Groups » sont-ils stateless ou statefull ?

- Les groupes de sécurité sur AWS sont conçus pour être stateful. Cela implique que l'autorisation de trafic entrant pour une règle spécifique permet également le passage automatique du trafic sortant associé, sans tenir compte des règles de sortie spécifiées.
- Quels sont les avantages qu'apporte SSM Session Manager par rapport à un Bastion SSH (comme Nat_JumpHost) ?
 - **Gestion unifiée** : Offre un contrôle et une visibilité centralisés sur l'infrastructure AWS et les ressources on-premise.
 - **Automatisation des processus** : Permet d'automatiser de nombreuses tâches de gestion, telles que le déploiement de correctifs, la configuration des machines et la gestion des licences.
 - **Sécurité renforcée et conformité** : Aide à sécuriser et à maintenir la conformité de l'infrastructure grâce à une gestion uniforme des ressources.
 - **Outils de diagnostic** : Propose des fonctionnalités pour le diagnostic et la résolution de problèmes sur les instances et les applications.
 - **Intégration et adaptabilité** : Se combine aisément avec d'autres services AWS pour fournir une solution de gestion globale et s'adapte à différents besoins d'utilisation

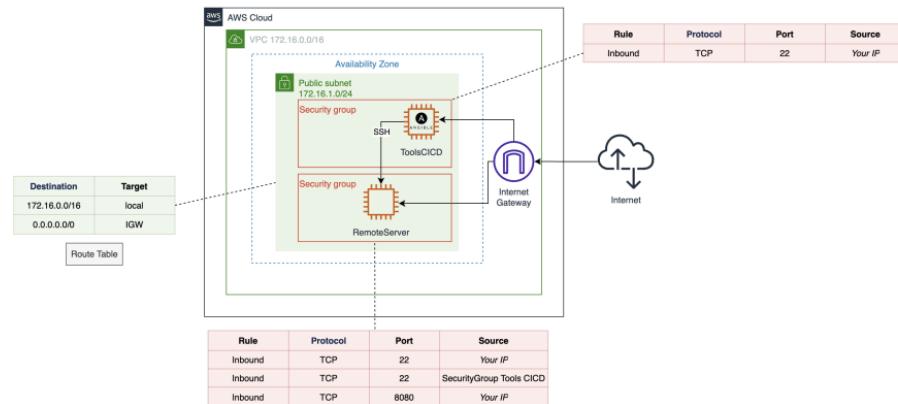
TP2



Automatisation avec Ansible

Présentation

<https://docs.ansible.com/ansible/latest/index.html>



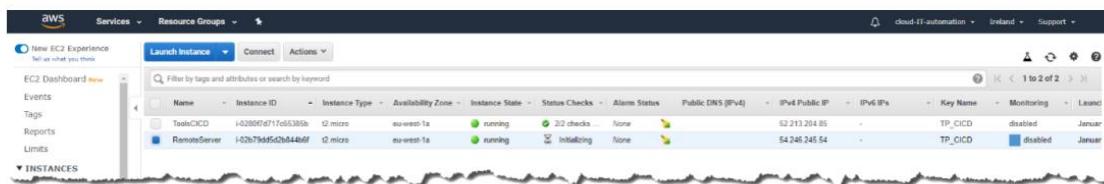
Nous allons dans un premier temps développer un Playbook Ansible qui déploie un serveur web et une application web.

Préparation de l'environnement

Les instances EC2 du TP1 ne sont plus nécessaires. Résiliez-les (Terminate) pour éviter d'être facturé.

Déployer deux instances EC2 sur AWS

| Name Tag | Type | AMI | Subnet | Inbound Security Group Rules |
|--------------|----------|---------------|------------------|---|
| ToolsCICD | t2.micro | Ubuntu Server | TP_DevOps_Public | SSH – from "My IP" |
| RemoteServer | t2.micro | Ubuntu Server | TP_DevOps_Public | SSH – from "My IP" SSH – from ToolsCICD Security Group Custom - 8080 – from "My IP" |



On crée d'abord les security group puis les instances :

ToolsCICD Security Group :

Détails de base

Nom du groupe de sécurité **Informations**
ToolsCICD Security Group Custom
Le nom ne peut pas être modifié après sa création.

Description **Informations**
SSH -> from 'My IP'

VPC **Informations**
vpc-0f1ab18484a127b70 (ab01-vpc-devops)

Règles entrantes Informations

| Type | Informations | Protocole | Informations | Plage de ports | Informations | Source | Informations | Description - facultatif | Informations |
|------|--------------|-----------|--------------|----------------|--------------|--------|--------------|--------------------------|----------------|
| SSH | | TCP | | 22 | | Mon IP | | | 109.222.8.2/32 |

[Ajouter une règle](#)

Règles sortantes Informations

| Type | Informations | Protocole | Informations | Plage de ports | Informations | Destination | Informations | Description - facultatif | Informations |
|----------------|--------------|-----------|--------------|----------------|--------------|-----------------|--------------|--------------------------|--------------|
| Tout le trafic | | Tous | | Tous | | Personnalisé(e) | | | 0.0.0.0/0 |

[Ajouter une règle](#)

⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Balises facultatif

Une balise est une étiquette que vous attribuez à une ressource AWS. Chaque balise se compose d'un clé et d'une valeur facultative. Vous pouvez utiliser des balises pour rechercher et filtrer vos ressources ou suivre vos coûts AWS.

Aucune balise n'est associée à cette ressource.

[Ajouter une balise](#)

Vous pouvez ajouter jusqu'à 50 identifications supplémentaires.

[Annuler](#) [Créer un groupe de sécurité](#)

Remote Server Security Group :

[EC2](#) > [Groupes de sécurité](#) > [sg-0f1ab18484a127b70 - RemoteServerGroup](#) > Modifier les règles entrantes

Modifier les règles entrantes Informations

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

Règles entrantes Informations

| ID de règle de groupe de sécurité | Type | Informations | Protocole | Informations | Plage de ports | Informations | Source | Informations | Description - facultatif | Informations |
|-----------------------------------|------------------|--------------|-----------|--------------|----------------|--------------|-----------------|--------------|--------------------------|--------------|
| sgr-02225604a35efc2d | SSH | | TCP | | 22 | | Personnalisé(e) | | 109.222.8.2/32 | |
| sgr-0c2588002f0f046d3 | SSH | | TCP | | 22 | | Personnalisé(e) | | sg-0bbcf586c7552c126 | |
| - | TCP personnalisé | | TCP | | 8080 | | Mon IP | | 109.222.8.2/32 | |

[Ajouter une règle](#)

[Annuler](#) [Aperçu des modifications](#) [Enregistrer les règles](#)

Ensuite on crée les instances Tools CICD et Remote Security Group qui seront dans le sous réseau public de notre vpc que l'on a créé dans le TP1

Paire de clés (connexion) Informations

Vous pouvez utiliser une paire de clés pour vous connecter en toute sécurité à votre instance. Assurez-vous d'avoir accès à la paire de clés sélectionnée avant de lancer l'instance.

Nom de la paire de clés - **obligatoire**
vokey

Récapitulatif

Nombre d'instances Informations
1

Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ami-0c7217cdde317rfec

Virtual server type (instance type)
t2.micro

Firewall (security group)
ToolsCICDGROUP

Storage (volumes)
1 volume(s) - 8 GiB

Offre gratuite : La première année inclut 750 heures d'utilisation mensuelle des instances t2.micro (ou t3.micro dans les régions où t2.micro n'est pas disponible) sur les AMI de l'offre gratuite, 30 Gio de stockage EBS, 2 millions d'I/O, 1 Go d'instantanés et 100 Go de bande passante vers Internet

Lancer l'instance

Paramètres réseau

VPC - required Informations
vpc-081ab89f4a127b70 (lab01-vpc-devops)
172.16.0.0/16

Sous-réseau Informations
subnet-067fe1d2744e79b5e Propriétaire: 822892150932
VPC: vpc-081ab89f4a127b70 Zone de disponibilité: us-east-1a Adresses IP disponibles: 250 CIDR: 172.16.1.0/24

Attribuer automatiquement l'adresse IP publique Informations
Activer

Pare-feu (groupes de sécurité) Informations
Un groupe de sécurité est un ensemble de règles de pare-feu qui contrôlent le trafic de votre instance. Ajoutez des règles pour autoriser un trafic spécifique à atteindre votre instance.

Créer un groupe de sécurité
 Sélectionner un groupe de sécurité existant

Groupes de sécurité courants Informations
Selectionner les groupes de sécurité ToolsCICDGROUP sg-0b8cf586c7552c126 VPC: vpc-081ab89f4a127b70

Comparer les règles de groupe de sécurité

Les groupes de sécurité que vous ajoutez ou supprimez ici seront ajoutés ou supprimés de toutes vos interfaces réseau.

Configuration réseau avancée

Configurer le stockage

Advanced

1x 8 Gio gp2 Volume racine (Non chiffré)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

Ajouter un volume

L'AMI sélectionnée contient un nombre de volumes de stockage d'instances supérieur à ce qui est autorisé. Seuls les 0 premiers volumes de stockage d'instance de l'AMI seront accessibles à partir de l'instance.

Détails **Status and alarms** **New** **Surveillance** **Sécurité** **Mise en réseau** **Stockage** **Balises**

Résumé de l'instance i-0cc9931b5e154d596 (ToolsCICD) Informations

Mis à jour il y a less de une minute

ID d'instance i-0cc9931b5e154d596 (ToolsCICD)

Adresse IPv6 -

Type de nom d'hôte Nom de l'adresse IP: ip-172-16-1-122.ec2.internal

Réponse à un nom DNS de ressource privée -

Adresse IP attribuée automatiquement 3.88.139.94 (IP publique)

Rôle IAM -

IMDSv2 Required

Détails Status and alarms New Surveillance Sécurité Mise en réseau Stockage Balises

Détails de l'instance Informations

Plateforme Ubuntu (désactivé)

Informations sur la plateforme Linux/UNIX

Protection contre l'arrêt Désactivé

Récupération automatique de l'instance Par défaut

Index de lancement de l'AMI 0

Spécification des crédits standard

Opération d'utilisation RunInstances

Prise en charge d'Endclaves -

Autoriser les identifications dans les métadonnées d'instance Désactivé

Hôte et groupe de placement Informations

ID de l'hôte -

Affinité -

Adresse IPv4 publique 5.88.139.94 (adresse ouverte)

État de l'instance En cours d'exécution

Nom DNS de l'IP privée (IPv4 uniquement) ip-172-16-1-122.ec2.internal

Type d'instance t2.micro

ID de VPC vpc-081ab89f4a127b70 (lab01-vpc-devops)

ID de sous-réseau subnet-067fe1d2744e79b5e (tp-devops_public)

Adresses IPv4 privées 172.16.1.122

DNS IPv4 public -

Adresses IP élastiques -

Recherche d'AWS Compute Optimizer Inscrivez-vous à AWS Compute Optimizer pour obtenir des recommandations. | En savoir plus

Nom du groupe Auto Scaling -

Surveillance désactivé

Protection de la résilience Désactivé

Emplacement de l'AMI amazon/ubuntu/images/hvm-ssd/ubuntu-jammy-22.04-amd64-server-20231207

Comportement Arrêt - Mise en veille prolongée Désactivé

Motif de transition de l'état -

Message de transition de l'état -

Propriétaire 822892150932

Mode de démarrage de l'instance actuelle legacy-bios

Répondre à l'IPv4 de nom d'hôte DNS RBN Désactivé

Groupe de placement -

Remote Server

Nom et balises [Informations](#)

Nom Ajouter des balises supplémentaires

▼ Application and OS Images (Amazon Machine Image) [Informations](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Effectuer une recherche dans notre catalogue complet, qui comprend des milliers d'images d'applications et de systèmes

Recents | **Démarrage rapide**

Amazon Linux
macOS
Ubuntu
Windows
Red Hat
SUSE Linux

Amazon Machine Image (AMI)

| | |
|--|-----------------------------|
| AMI Amazon Linux 2023 ami-0a3c3a20c09d6f377 (64 bits (x86), uefi-preferred) / ami-0bb6ccb99aec63c04 (64 bits (Arm), uefi) Virtualisation: hvm ENA enabled: true Type de périphérique racine: ebs | Éligible à l'offre gratuite |
|--|-----------------------------|

Description
Amazon Linux 2023 AMI 2023.3.20240122.0 x86_64 HVM kernel-6.1

| | | | |
|-------------------------------|-------------------------------------|---------------------------------|----------------------------|
| Architecture 64 bits (x86) | Mode de démarrage uefi-preferred | ID AMI ami-0a3c3a20c09d6f377 | Fournisseur vérifié |
|-------------------------------|-------------------------------------|---------------------------------|----------------------------|

▼ Type d'instance [Informations](#) | [Get advice](#)

| | |
|---|-----------------------------|
| t2.micro Famille: t2 1 vCPU 1 Go Mémoire Génération actuelle: true À la demande Windows base tarification: 0.0162 USD par heure À la demande SUSE base tarification: 0.0116 USD par heure À la demande RHEL base tarification: 0.0716 USD par heure À la demande Linux base tarification: 0.0116 USD par heure | Éligible à l'offre gratuite |
|---|-----------------------------|

Des frais supplémentaires s'appliquent pour les AMI avec un logiciel préinstallé

[CloudWatch](#) [Commentaires](#)

Nombre d'instances [Informations](#)

1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.3.2...read more
ami-0a3c3a20c09d6f377

Virtual server type (instance type)
t2.micro

Firewall (security group)
Nouveau groupe de sécurité

Storage (volumes)
1 volume(s) - 8 GiB

Offre gratuite : La première année inclut 750 heures d'utilisation mensuelle des instances t2.micro (ou t3.micro dans les régions où t2.micro n'est pas disponible) sur les AMI de l'offre gratuite, 30 Go de stockage EBS, 2 millions d'I/O, 1 Go d'instantané et 100 Go de bande passante vers Internet

Annuler **Lancer l'Instance** [Examiner les commandes](#)

Page 19 sur 54

Paramètres réseau

- VPC - **requis** Informations
 - vpc-081ab09bf4a127b70 (lab01-vpc-devops) 172.16.0.0/16
- Sous-réseau Informations
 - subnet-06f7e1f2744a79b5e tp-devops_public VPC: vpc-081ab09bf4a127b70 Propriétaire: 82289215032 Zone de disponibilité: us-east-1a Adresses IP disponibles: 250 CIDR: 172.16.1.0/24
- Atribuer automatiquement l'adresse IP publique Informations
 - Activer
- Par-feu (groupes de sécurité) Informations
 - Un groupe de sécurité est un ensemble de règles de par-feu qui contrôlent le trafic de votre instance. Ajoutez des règles pour autoriser un trafic spécifique à atteindre votre instance.
 - Créer un groupe de sécurité
 - Sélectionner un groupe de sécurité existant
- Groupes de sécurité courants Informations
 - Selectionner les groupes de sécurité RemoteServerGroup sp-06ade1f2f75432cd3 [X] VPC: vpc-081ab09bf4a127b70
- Comparer les règles de groupe de sécurité
- Les groupes de sécurité que vous ajoutez ou supprimez ici seront ajoutés ou supprimés de toutes vos interfaces réseau.
- ▶ Configuration réseau avancée

Configurer le stockage

- Advanced
- 1x 8 GiB gp2 Volume racine (Non chiffré)
- Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage [X]
- Ajouter un volume
- L'AMI sélectionnée contient un nombre de volumes de stockage d'instances supérieur à ce qui est autorisé. Seuls les 0 premiers volumes de stockage d'instance de l'AMI seront accessibles à partir de l'instance.
- Cliquez sur Actualiser pour afficher les informations de sauvegarde The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.
- 0 systèmes de fichiers Modifier
- ▶ Détails avancés Informations

Récapitulatif

- Nombre d'instances Informations
 - 1
- Software Image (AMI)
 - Canonical_Ubuntu_22.04 LTS, ...read more ami-0c7217c7cb5d317fec
- Virtual server type (instance type)
 - t2.micro
- Firewall (security group)
 - RemoteServerGroup
- Storage (volumes)
 - 1 volume(s) - 8 GiB

Offre gratuite : La première année inclut 750 heures d'utilisation mensuelle des instances t2.micro (ou t3.micro dans les régions où t2.micro n'est pas disponible) sur les AMI de l'offre gratuite, 30 Go de stockage EBS, 2 millions d'I/O, 1 Go d'instantanés et 100 Go de bande passante vers Internet

Annuler Lancer l'instance Examiner les commandes

1- Connectez-vous en SSH ou SSM aux deux instances

2- Installer Ansible sur ToolsCICD uniquement

https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html

```
sudo apt update && \
sudo apt install software-properties-common --yes && \
sudo apt-add-repository --update ppa:ansible/ansible --yes && \
sudo apt install ansible --yes
```

On se connecte en SSH sur TOOLS CICD et on installe Ansible uniquement sur ToolsCICD

```
[+] abraham@172-9-152-1: ~ - - - - -  
Windows PowerShell  
Copyright (C) Microsoft Corporation. Tous droits réservés.  
Installez la dernière version de PowerShell pour de nouvelles fonctionnalités et améliorations : https://aka.ms/PSWindows  
  
PS C:\Users\abraham> ssh -i "C:\Users\abraham\cyberkey.ppk" ubuntu@91.8.114  
The authenticity of host '91.8.114 (91.8.114)' can't be established.  
ECDSA key fingerprint is 10:cd:65:bd:5a:99:9a:30:7f:9c:9e:0c:9c:0d:49:ad  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added '91.8.114' (ECDSA) to the list of known hosts.  
ubuntu@91.8.114:~$ whoami  
ubuntu@91.8.114:~$ id  
uid=1000(ubuntu) gid=1000(ubuntu) groups=1000(ubuntu)  
  
expanded Security Maintenance for Applications is not enabled.  
0 updates can be applied immediately.  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
conditional files in /usr/share/doc/*copyright*.  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
For a user command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@91.8.114:~$ sudo apt update &> /dev/null  
ubuntu@91.8.114:~$ sudo apt install software-properties-common -y &> /dev/null  
ubuntu@91.8.114:~$ sudo apt-add-repository --update ppa:ansible/ansible -y &> /dev/null  
  
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease  
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [139 kB]  
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-security InRelease [110 kB]  
Get:4 http://security.ubuntu.com/ubuntu jammy-security Translation-en [16.1 kB]  
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [102 kB]  
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe andro-c=none Metadata [286 kB]  
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe andro-c=Metadata [286 kB]  
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe andro-c=none Metadata [133 kB]  
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [112 kB]  
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse andro-c=none Metadata [295 kB]  
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse andro-c=Metadata [295 kB]  
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [257 kB]  
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [229 kB]  
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [202 kB]  
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe andro-c=Metadata [162 kB]  
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe andro-c=Metadata [162 kB]  
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe andro-c=Metadata [22.1 kB]  
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse andro-c=Metadata [42.1 kB]  
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse andro-c=Metadata [42.1 kB]  
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse andro-c=Metadata [472 kB]  
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [979 kB]  
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main andro-c=Metadata [979 kB]  
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main andro-c=Metadata [338 kB]
```

Vérification de l'installation

```
ansible --version
```

```
[x] ubuntu@ip-172-16-1-122:~ x + - Setting up python3-nacl (1.5.0-2) ... Setting up python3-requests-ntlm (1.1.0-1.1) ... Setting up ansible-core (2.15.9-1ppa-jammy) ... Setting up python3-winrm (0.3.0-2) ... Setting up ansible (8.7.0-1ppa-jammy) ... Setting up python3-paramiko (2.9.3-0ubuntu1.2) ... Processing triggers for man-db (2.10.2-1) ... Scanning processes... Scanning linux images... Running kernel seems to be up-to-date. No services need to be restarted. No containers need to be restarted. No user sessions are running outdated binaries. No VM guests are running outdated hypervisor (qemu) binaries on this host. ubuntu@ip-172-16-1-122:~$ ansible --version ansible [core 2.15.9] config file = /etc/ansible/ansible.cfg configured module search path = ['/home/ubuntu/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules'] ansible python module location = /usr/lib/python3/dist-packages/ansible[|||] ansible collection location = /home/ubuntu/.ansible/collections:/usr/share/ansible/collections executable location = /usr/bin/ansible python version = 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0] (/usr/bin/python3) jinja version = 3.0.3 libyaml = True ubuntu@ip-172-16-1-122:~$ |
```

On se connecte en SSH sur Remote Sever

```

[ctrl] ubuntu@ip-172-16-1-118: ~      X   +   v

System information as of Thu Feb  1 13:10:46 UTC 2024

System load:  0.0          Processes:         97
Usage of /:   20.6% of 7.57GB  Users logged in:     0
Memory usage: 20%           IPv4 address for eth0: 172.16.1.118
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

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-172-16-1-118:~$ |

```

3- Copier la private key (fichier pem) sur ToolsCICD uniquement

Sur ToolsCICD Copier votre clé privée dans le répertoire « .ssh » de votre profile, soit via un outil (MobaXterm), la commande scp ou manuellement (en insérant le contenu du fichier pem) :

```

cat > ~/.ssh/labsuser.pem << EOF
-----BEGIN RSA PRIVATE KEY-----
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
-----END RSA PRIVATE KEY-----
EOF

```

Modifier les permissions sur la clé privée (privé en lecture seule)

```
chmod 400 ~/.ssh/labsuser.pem
```

4- Créer un répertoire de travail dans la home de votre user

```
mkdir -p ~/tp2 && cd ~/tp2
```

On a copié la clé privée vockey sur l'instance ToolsCICD et on modifie les permissions :

```
Vos disques durs sont accessibles au travers du dossier /drives
Le DISPLAY est positionné à 192.168.1.29:0.0
Lors d'une connexion SSH, le DISPLAY est automatiquement exporté
Le statut de chaque commande est indiqué par un symbole (> ou x)

* Important:
Vous utilisez la version personnelle de MobaXterm.
En achetant la version professionnelle de MobaXterm, vous pourrez
personnaliser le logiciel à votre convenance, en pré-définissant
vos options, votre propre message de bienvenue, votre logo ainsi
que plusieurs autres paramètres.
En plus de vous assurer un support professionnel, nous pouvons
modifier le programme MobaXterm ou développer de nouveaux plugins
selon vos besoins. Pour plus d'informations, utilisez Ctrl + Clic
sur le lien suivant : https://mobaxterm.mobatek.net/download.html

[ 01/02/2024 09:42:45 ] ssh -i "C:\Users\Yk\Downloads\vockey.pem" ubuntu@3.91.8.114
Warning: Permanently added '3.91.8.114' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1017-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

 System information as of Thu Feb 1 13:25:48 UTC 2024

 System load: 0.0          Processes:           102
 Usage of /: 30.1% of 7.57GB   Users logged in:      1
 Memory usage: 29%          IPv4 address for eth0: 172.16.1.122
 Swap usage:  0%

 Expanded Security Maintenance for Applications is not enabled.

 61 updates can be applied immediately.
 36 of these updates are standard security updates.
 To see these additional updates run: apt list --upgradable

 1 additional security update can be applied with ESM Apps.
 Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Last login: Thu Feb 1 13:12:02 2024 from 109.222.2.2.2
/usr/bin/xauth: file /home/ubuntu/.Xauthority does not exist
ubuntu@ip-172-16-1-122:~$ chmod 400 vockey.pem
ubuntu@ip-172-16-1-122:~$ [ ]
```

On crée le répertoire tp2 et on s'y rend

```
Last login: Thu Feb  1 13:12:02 2024 from 109.222.8.2
/usr/bin/xauth:  file /home/ubuntu/.Xauthority does not exist
ubuntu@ip-172-16-1-122:~$ chmod 400 vockey.pem
ubuntu@ip-172-16-1-122:~$ mkdir -p ~/tp2 && cd ~/tp2
ubuntu@ip-172-16-1-122:~/tp2$ █
```

Ping de l'instance RemoteServer avec Ansible

Remplacez la valeur de `RemoteServerPrivateIp` par l'adresse IP privée de votre instance `RemoteServer`.

```
RemoteServerPrivateIp="172.16.1.57"  
ansible all -m ping -i ubuntu@${RemoteServerPrivateIp} --private-key ~/.ssh/labuser.pem
```

```
ansible all -m ping -i ubuntu@$RemoteServerIp, --private-key ~/.ssh/labsuser.pem  
ubuntu@172.16.1.57 | SUCCESS => {  
    "ansible_facts": {  
        "discovered_interpreter_python": "/usr/bin/python3"  
    },  
    "changed": false,  
    "ping": "pong"  
}
```

```
ubuntu@ip-172-16-1-122:~/tp2$ RemoteServerPrivateIp="172.16.1.118"
ansible all -m ping -i ubuntu@$RemoteServerPrivateIp, --private-key ../vokey.pem
vokey.pem
ubuntu@ip-172-16-1-122:~/tp2$ RemoteServerPrivateIp="172.16.1.118"
ansible all -m ping -i ubuntu@$RemoteServerPrivateIp, --private-key ../vokey.pem
ubuntu@172.16.1.118 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-16-1-122:~/tp2$ █
```

Playbook

https://docs.ansible.com/ansible/latest/user_guide/playbooks.html

Créer un fichier `ping.yml` dans le dossier `tp2` avec ce contenu :

```
cat > ping.yml <<EOF
---
- hosts: all
  tasks:
    - name: test connection
      ping:
EOF
```

Source : <https://gitlab.com/efrei-devops/tps/-/blob/main/tp2/ping.yml>

Exécuter le playbook

```
ansible-playbook -i ubuntu@$RemoteServerPrivateIp, --private-key ~/.ssh/labsuser.pem ping.yml
```

```
ubuntu@ip-172-16-1-40:~/tp2$ ansible-playbook -i ubuntu@$RemoteServerPrivateIp, --private-key ~/.ssh/labsuser.pem ping.yml
PLAY [all] ****
TASK [Gathering Facts] ****
ok: [ubuntu@ip-172.16.1.57]
TASK [test connection] ****
ok: [ubuntu@ip-172.16.1.57]
PLAY RECAP ****
ubuntu@ip-172.16.1.57 : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

```
* here
ubuntu@ip-172-16-1-122:~/tp2$ cat > ping.yml <<EOF
---
- hosts: all
  tasks:
    - name: test connection
      ping:
EOF
ubuntu@ip-172-16-1-122:~/tp2$ ansible-playbook -i ubuntu@172.16.1.118, --private-key ./vockey.pem ping.yml
PLAY [all] ****
TASK [Gathering Facts] ****
ok: [ubuntu@ip-172.16.1.118]
TASK [test connection] ****
ok: [ubuntu@ip-172.16.1.118]
PLAY RECAP ****
ubuntu@ip-172.16.1.118 : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu@ip-172-16-1-122:~/tp2$
```

Déployer un serveur Web avec Ansible

Créez un second fichier playbook (`play.yml`) dans le dossier `tp2`. Il intègrera toutes les installations/configurations nécessaires à l'exécution d'un site web.

Nous l'utiliserons ensuite avec Packer pour créer une AMI.

Ce playbook doit :

- Installer Git
- Installer Apache dans sa dernière version
- Changer le port d'écoute d'Apache et du Virtualhost sur le port 8080
- Supprimer le default website d'Apache (`/var/www/html`)
- Déployer un website <https://github.com/cloudacademy/static-website-example>
- Redémarrer le service Apache

Exécutez ce playbook dans le dossier `tp2` :

```
ansible-playbook -i ubuntu@$RemoteServerPrivateIp, --private-key ~/.ssh/labsuser.pem play.yml
```

```

...
- name: Déploiement d'un serveur web avec Apache
hosts: all
become: yes
tasks:
  - name: Installer Git
    apt:
      name: git
      state: latest

  - name: Installer Apache
    apt:
      name: apache2
      state: latest

  - name: Changer le port d'écoute d'Apache
    lineinfile:
      path: /etc/apache2/ports.conf
      regexp: '^Listen 80'
      line: Listen 8080
    notify: redémarrer apache

  - name: Changer le port dans le VirtualHost
    lineinfile:
      path: /etc/apache2/sites-available/000-default.conf
      regexp: '<VirtualHost *:80>'
      line: '<VirtualHost *:8080>'
    notify: redémarrer apache

  - name: Supprimer le site web par défaut
    file:
      path: /var/www/html
      state: absent

  - name: Déployer le site web statique depuis GitHub
    git:
      repo: 'https://github.com/cloudacademy/static-website-example'
      dest: /var/www/html

  - name: Assurer que le dossier /var/www/html existe
    file:
      path: /var/www/html
      state: directory

  - name: Redémarrer Apache
    service:
      name: apache2
      state: restarted
EOF

```

Exemple de résultat d'exécution du playbook :

```
ubuntu@ip-172-16-1-40:~/tp2$ ansible-playbook -i ubuntu@$RemoteServerIp, --private-key ~/ssh/labsuser.pem play.yml
PLAY [all] ****
TASK [Gathering Facts] ****
ok: [ubuntu@ip-172-16-1-57]
TASK [Install Git package] ****
ok: [ubuntu@ip-172-16-1-57]
TASK [ensure apache is at the latest version] ****
changed: [ubuntu@ip-172-16-1-57]
TASK [enabled mod_rewrite] ****
changed: [ubuntu@ip-172-16-1-57]
TASK [apache2 listen on port 8080] ****
changed: [ubuntu@ip-172-16-1-57]
TASK [apache2 virtualhost on port 8080] ****
changed: [ubuntu@ip-172-16-1-57]
TASK [remove default website directory] ****
changed: [ubuntu@ip-172-16-1-57]
TASK [Git checkout website] ****
changed: [ubuntu@ip-172-16-1-57]
RUNNING HANDLER [restart apache2] ****
changed: [ubuntu@ip-172-16-1-57]
PLAY RECAP ****
ubuntu@ip-172-16-1-57 : ok=9    changed=7    unreachable=0   failed=0    skipped=0   rescued=0   ignored=0
```

Testez l'accès au site web dans votre navigateur sur l'**IP Publique de RemoteServer**:

<http://RemoteServerPublicIp:8080>

```
ansible-playbook -i ubuntu@172.16.1.118, --private-key ./rockey.pem play.yml
PLAY [Déploiement d'un serveur web avec Apache] ****
TASK [Gathering Facts] ****
ok: [ubuntu@172.16.1.118]
TASK [Installer Git] ****
ok: [ubuntu@172.16.1.118]
TASK [Installer Apache] ****
changed: [ubuntu@172.16.1.118]
TASK [Changer le port d'écoute d'Apache] ****
changed: [ubuntu@172.16.1.118]
TASK [Changer le port dans le VirtualHost] ****
changed: [ubuntu@172.16.1.118]
TASK [Supprimer le site web par défaut] ****
changed: [ubuntu@172.16.1.118]
TASK [Déployer le site web statique depuis GitHub] ****
changed: [ubuntu@172.16.1.118]
TASK [Assurer que le dossier /var/www/html existe] ****
ok: [ubuntu@172.16.1.118]
TASK [Redémarrer Apache] ****
changed: [ubuntu@172.16.1.118]
RUNNING HANDLER [redémarrer apache] ****
changed: [ubuntu@172.16.1.118]
PLAY RECAP ****
ubuntu@172.16.1.118 : ok=10    changed=7    unreachable=0   failed=0    skipped=0   rescued=0   ignored=0
```

Ubuntu 172.16.1.118 is a free item by RoboTeam. Get the professional edition here: <https://roboteam.mirabek.net>

On voit bien que l'on a le même site que ici



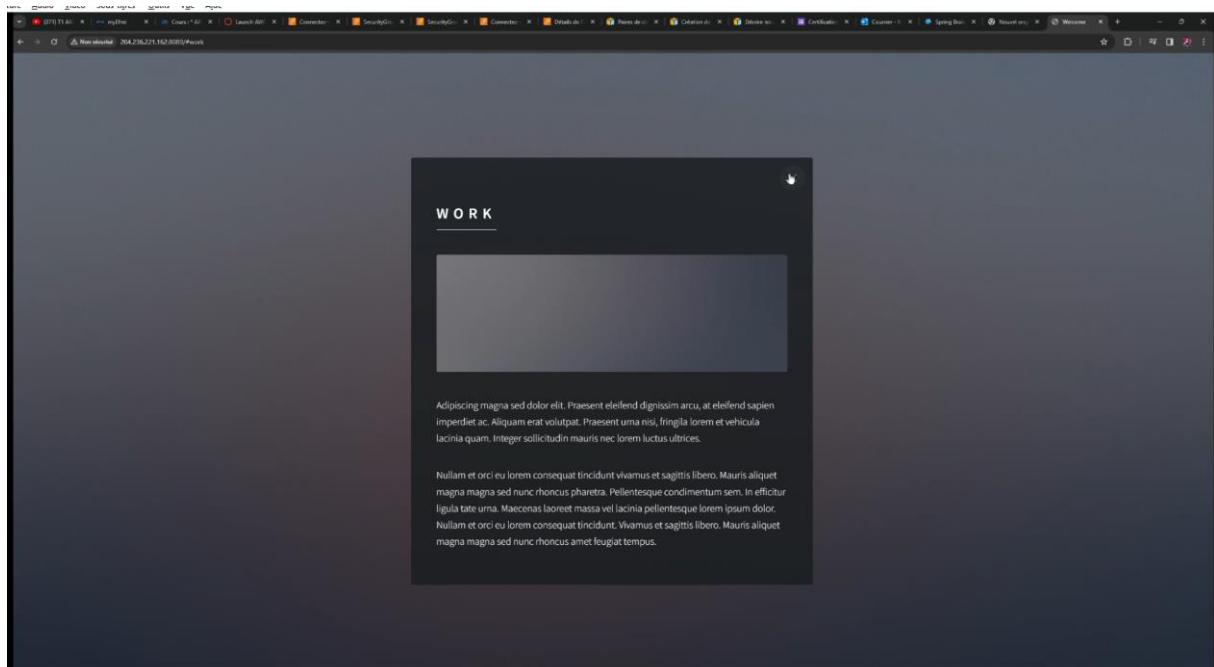
! Pour pouvoir accéder au site Web :

- Vérifiez bien que le « Security Group » de l'instance RemoteServer autorise bien l'accès (Inbound Rule) sur le port 8080 depuis vers votre IP publique

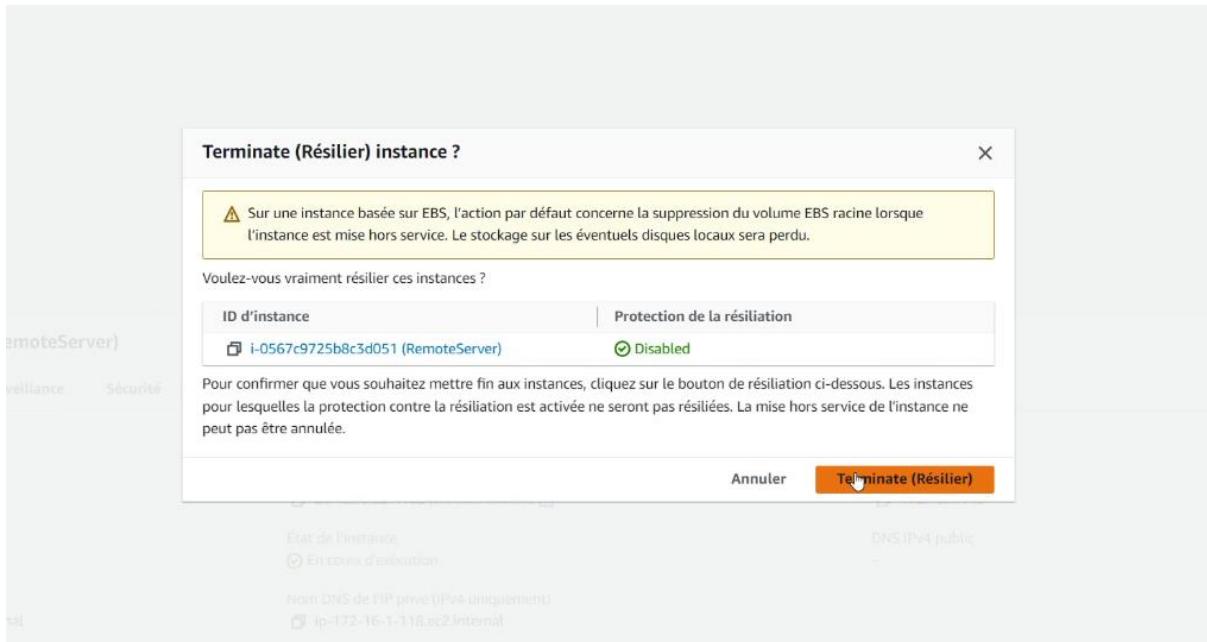
! Une fois que votre playbook fonctionne :

- Résiliez (*Terminate*) l'instance RemoteServer car nous ne l'utiliserons plus.

Documentations



On n'oublie pas de réinitialiser RemoteServer



Maintenant on va créer une AMI avec Packer

Installation de Packer

<https://www.packer.io/intro/getting-started/install.html>

Connectez-vous sur l'instance ToolsCICD puis exécutez :

```
curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo apt-key add - && \
sudo apt-add-repository -y "deb [arch=amd64] https://apt.releases.hashicorp.com $(lsb_release -cs) \
main" && \
sudo apt-get update && sudo apt-get install -y packer
```

Vérifiez que **Packer** est bien installé :

```
packer -v
```

<https://www.packer.io/docs/builders/amazon.html>

```
Last login: Thu Feb  1 13:25:49 2024 from 109.222.8.2
ubuntu@ip-172-16-1-122:~$ curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo apt-key add - && \
sudo apt-add-repository -y "deb [arch=amd64] https://apt.releases.hashicorp.com $(lsb_release -cs) \
main" && \
sudo apt-get update && sudo apt-get install -y packer
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
Repository: 'deb [arch=amd64] https://apt.releases.hashicorp.com jammy main'
Description:
Archive for codename: jammy components: main
More info: https://apt.releases.hashicorp.com
Adding repository.
Adding deb entry to /etc/apt/sources.list.d/archive_uri-https_apt_releases_hashicorp_com-jammy.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/archive_uri-https_apt_releases_hashicorp_com-jammy.list
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 https://apt.releases.hashicorp.com jammy InRelease [12.9 kB]
Hit:5 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy InRelease
Get:6 https://apt.releases.hashicorp.com jammy/main amd64 Packages [116 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
0% [Working]
```

```
No containers need to be restarted.

No user sessions are running outdated binaries.

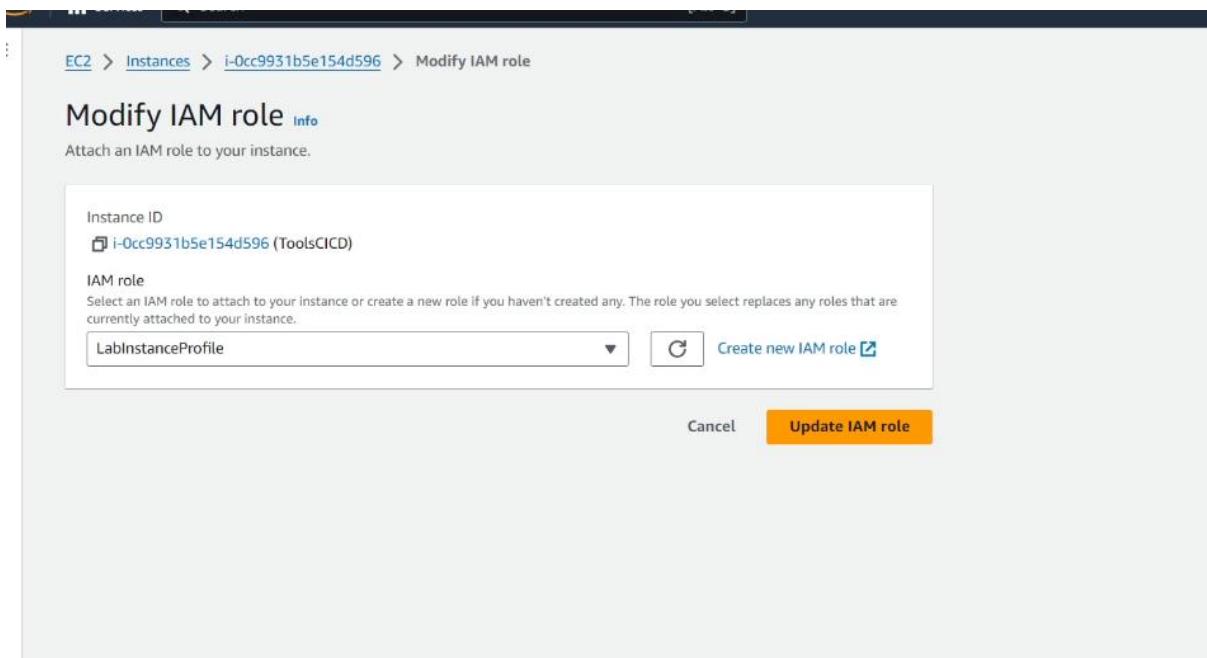
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-16-1-122:~$ packer -v
Packer v1.10.1
ubuntu@ip-172-16-1-122:~$
```

Configuration des permissions IAM

Par l'intermédiaire du **Builder amazon-ebs**, **Packer** va interagir avec les **API AWS**.

Il lui faudra donc les droits nécessaires pour s'y authentifier. Nous allons associer un rôle IAM pour octroyer les permissions nécessaires à l'instance par l'intermédiaire des « MetaData » de l'instance.

Attacher le rôle **LabInstanceProfile** à l'instance **ToolsCICD**



Questions :

- Configuration d'Ansible : Décrivez les étapes suivies pour configurer Ansible sur votre instance EC2. Quels obstacles avez-vous rencontrés et comment les avez-vous surmontés ?
 - **Initialisation avec Ansible** : Accès à mon serveur EC2 suivi par l'installation d'Ansible conformément aux instructions officielles.
 - **Préparation de l'environnement de travail** : Après avoir installé Ansible, transfert de la clé privée (.pem) vers le serveur, création d'un répertoire dédié pour l'organisation des projets et mise en place des configurations de base nécessaires.
 - **Création d'un playbook Ansible** : Conception d'un playbook pour l'implémentation d'un serveur web et le déploiement d'une application web,

incluant l'installation de Git, la configuration d'Apache sur le port 8080, la création d'un hôte virtuel, et le déploiement de l'application depuis un dépôt spécifique.

- Développement du playbook : Expliquez le processus de création de votre playbook Ansible. Comment avez-vous vérifié qu'il répondait à toutes les exigences pour le déploiement d'un serveur web ?
 - **Configuration initiale avec Ansible** : Installation d'Ansible sur le serveur EC2 en suivant les instructions de la documentation.
 - **Organisation de l'espace de travail** : Configuration de l'environnement post-installation d'Ansible, incluant la copie de la clé privée, la création d'un dossier pour les projets et l'ajustement des paramètres nécessaires.
 - **Conception d'un playbook pour le déploiement** : Réalisation d'un playbook pour le déploiement d'un serveur web et d'une application web, comportant l'installation de Git, la configuration d'Apache et la mise en place d'un hôte virtuel.
 - **Assurance de la conformité du playbook** : Exécution et tests du playbook pour garantir qu'il remplit toutes les conditions pour le déploiement efficace d'un serveur web.
- Intégration de Packer : Discutez de la manière dont vous avez intégré Ansible avec Packer dans votre TP. Quel rôle chaque outil a-t-il joué dans le processus de création de l'AMI ?
 - **Installation de Packer** : Procédure d'installation de Packer sur le serveur EC2.
 - **Configuration des droits IAM** : Ajustement des droits IAM pour autoriser Packer à interagir avec les services AWS, attribution d'un rôle IAM avec les permissions nécessaires via les métadonnées.
 - **Création du script Packer** : Élaboration d'un script Packer définissant les étapes pour démarrer une instance EC2 temporaire, l'utilisation d'Ansible pour le provisionnement et la finalisation de l'AMI.
 - **Exécution de Packer** : Démarrage du processus de création de l'AMI avec Packer, impliquant l'instanciation d'une EC2 temporaire et l'exécution du playbook Ansible pour le provisionnement.
 - **Rôles de Packer et Ansible** : Importance de Packer dans la préparation de l'instance EC2 et d'Ansible comme outil de provisionnement pour préparer l'image avec les applications nécessaires.
- Résolution de problèmes : Partagez un problème spécifique rencontré durant le TP et comment vous l'avez résolu. Quels outils ou ressources vous ont été les plus utiles ?
 - **Problème avec Apache** : Difficulté avec le serveur Apache n'acceptant pas les requêtes.

- **Solution adoptée :** Ajout d'une règle personnalisée pour autoriser le trafic sur le port TCP 8080.
- **Ressources utiles :** Utilisation de recherches sur internet pour trouver des solutions.
- Réflexion sur l'automatisation et les pratiques DevOps : En quoi ces tâches ont-elles amélioré votre compréhension de l'automatisation dans le DevOps ? Discutez de l'importance de l'automatisation dans la gestion de l'infrastructure cloud.
 - La manière dont l'expérience a souligné l'importance de l'automatisation pour une gestion efficace de l'infrastructure, la réduction des erreurs et l'amélioration de la cohérence des environnements.
- Nettoyage des ressources : Pourquoi est-il important de désenregistrer les AMI et de supprimer les instantanés dans AWS ? Quelles pourraient être les conséquences de ne pas le faire ?
 - L'importance de désenregistrer les AMI et de supprimer les snapshots pour la gestion des coûts et le maintien d'un environnement cloud organisé, ainsi que les conséquences potentielles de négliger cette étape.
- Améliorations futures : Si vous deviez refaire ce projet, quels aspects amélioreriez-vous ou ferez-vous différemment ? Discutez de toute amélioration potentielle ou optimisation.
 - Réflexions sur les améliorations potentielles pour le projet, incluant l'optimisation des playbooks, l'intégration des processus CI/CD, et l'amélioration des pratiques de sécurité.

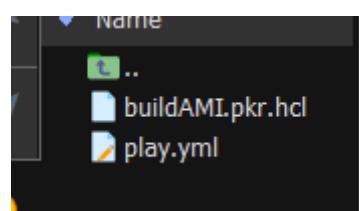
TP3

Nous disposons de l'instance ToolsCICD :

```

3. ubuntu@ip-172-16-1-135: ~
ubuntu@ip-172-16-1-135:~$ packer -v
Packer v1.10.1
ubuntu@ip-172-16-1-135:~$ ansible --version
ansible [core 2.15.9]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['~/home/ubuntu/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = ~/home/ubuntu/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0] (/usr/bin/python3)
  jinja version = 3.0.3
  libyaml = True
ubuntu@ip-172-16-1-135:~$ 

```



```

ubuntu@ip-172-16-1-135:~/TP3$ packer init buildAMI.pkr.hcl
Installed plugin github.com/hashicorp/amazon v1.3.0 in "/home/ubuntu/.config/packer/
thub.com/hashicorp/amazon/packer-plugin-amazon_v1.3.0_x5.0_linux_amd64"
Installed plugin github.com/hashicorp/ansible v1.1.1 in "/home/ubuntu/.config/packer
ithub.com/hashicorp/ansible/packer-plugin-ansible_v1.1.1_x5.0_linux_amd64"

```

```

ubuntu@ip-172-16-1-135:~/TP3$ packer init buildAMI.pkr.hcl
ubuntu@ip-172-16-1-135:~/TP3$ packer build buildAMI.pkr.hcl
amazon-ebs.static-web-ami: output will be in this color.

=> amazon-ebs.static-web-ami: Prevalidating any provided VPC information
=> amazon-ebs.static-web-ami: Prevalidating AMI Name: WebApp-01_02_2024-15_07
  amazon-ebs.static-web-ami: Found Image ID: ami-0c7217cdde317cfec
=> amazon-ebs.static-web-ami: Setting public IP address to true on instance without a subnet ID
=> amazon-ebs.static-web-ami: No VPC ID provided, Packer will use the default VPC
=> amazon-ebs.static-web-ami: Inferring subnet from the selected VPC "vpc-074b2f56858e9ee3e"
=> amazon-ebs.static-web-ami: Set subnet as "subnet-0b8dfdf4791a6c4f3"
=> amazon-ebs.static-web-ami: Creating temporary keypair: packer_65bbb3a1-19f5-42ac-f13c-1220b76c2597
=> amazon-ebs.static-web-ami: Creating temporary security group for this instance: packer_65bbb3a3-42ce-3bb1-6766-130d64e22f8f
=> amazon-ebs.static-web-ami: Authorizing access to port 22 from [0.0.0.0/0] in the temporary security groups...
=> amazon-ebs.static-web-ami: Launching a source AWS instance...
=> amazon-ebs.static-web-ami: changing public IP address config to true for instance on subnet "subnet-0b8dfdf4791a6c4f3"
  amazon-ebs.static-web-ami: Instance ID: i-08b242fea9bb7c6bb
=> amazon-ebs.static-web-ami: Waiting for instance (i-08b242fea9bb7c6bb) to become ready...
=> amazon-ebs.static-web-ami: Using SSH communicator to connect: 100.26.171.162
=> amazon-ebs.static-web-ami: Waiting for SSH to become available...
=> amazon-ebs.static-web-ami: Connected to SSH!
=> amazon-ebs.static-web-ami: Provisioning with Ansible...
  amazon-ebs.static-web-ami: Not using Proxy adapter for Ansible run:
    amazon-ebs.static-web-ami: Using ssh keys from Packer communicator

```

```

ubuntu@ip-172-16-1-135:~/TP3$ sudo apt-get install terraform
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
terraform is already the newest version (1.7.2-1).
0 upgraded, 0 newly installed, 0 to remove and 57 not upgraded.
ubuntu@ip-172-16-1-135:~/TP3$ 

```

```

3. ubuntu@ip-172-16-1-135:~/TP3$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.34.0...
- Installed hashicorp/aws v5.34.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
ubuntu@ip-172-16-1-135:~/TP3$ terraform plan
data.aws_ami.web_ami: Reading...
data.aws_ami.web_ami: Read complete after 0s [id=ami-0d0bcd668a5a11702]

Changes to Outputs:
+ web_ami = {
    + id   = "ami-0d0bcd668a5a11702"
    + name = "WebApp-01_02_2024-15_07"
  }

You can apply this plan to save these new output values to the Terraform state, without changing any real infrastructure.

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply".
ubuntu@ip-172-16-1-135:~/TP3$ 
```

```

3. ubuntu@ip-172-16-1-135:~/TP3$ alias tf="terraform"
ubuntu@ip-172-16-1-135:~/TP3$ alias tfi="tf init"
ubuntu@ip-172-16-1-135:~/TP3$ alias tfp="tf plan"
ubuntu@ip-172-16-1-135:~/TP3$ alias tfa="tf apply"
ubuntu@ip-172-16-1-135:~/TP3$ alias tfd="tf destroy"
ubuntu@ip-172-16-1-135:~/TP3$ 
```

LE TP 1 AS A CODE :

```

2. ubuntu@ip-172-16-1-135:~/TP3/tp1_asacode$ terraform init
Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/http...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/http v3.4.1...
- Installed hashicorp/http v3.4.1 (signed by HashiCorp)
- Installing hashicorp/aws v5.34.0...
- Installed hashicorp/aws v5.34.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
ubuntu@ip-172-16-1-135:~/TP3/tp1_asacode$ terraform plan
var.personal_ip_address
  Enter a value: 91.168.221.173

```

Cela me recrée les ressources du TP 1.

```

Apply complete! Resources: 12 added, 0 changed, 0 destroyed.

Outputs:

nat_private_ip = "172.16.1.145"
nat_public_ip = "54.145.87.19"
private_host_ip = "172.16.2.128"
ubuntu@ip-172-16-1-135:~/TP3/tp1_asacode$ █

```

J'arrive à me connecter à NAT_JUMPHOST :

```
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '54.145.87.19' (ED25519) to the list of known hosts.
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1052-aws x86_64)
```

- * Documentation: <https://help.ubuntu.com>
- * Management: <https://landscape.canonical.com>
- * Support: <https://ubuntu.com/pro>

System information as of Thu Feb 1 15:54:58 UTC 2024

| | |
|-----------------------------|-------------------------------------|
| System load: 0.0 | Processes: 100 |
| Usage of /: 23.6% of 7.57GB | Users logged in: 0 |
| Memory usage: 24% | IPv4 address for eth0: 172.16.1.145 |
| Swap usage: 0% | |

Expanded Security Maintenance for Applications is not enabled.

2 updates can be applied immediately.

2 of these updates are standard security updates.

To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.

See <https://ubuntu.com/esm> or run: sudo pro status

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-172-16-1-145:~$ ls
ubuntu@ip-172-16-1-145:~$ ls
ubuntu@ip-172-16-1-145:~$ exit
logout
Connection to 54.145.87.19 closed.
ubuntu@ip-172-16-1-135:~/TP3/tp1_asacode$ █
```

Tout fonctionne correctement nous sommes connectés en SSH sur NAT_jumphost depuis ToolsCICD et nous sommes connecté à PRIVATHOST depuis NAT_jumphost.

Nous avons réussi à réaliser le ping et le curl :

```

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-172-16-2-128:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=15 time=1.69 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=15 time=1.83 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=15 time=1.84 ms

--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 1.691/1.789/1.844/0.069 ms
ubuntu@ip-172-16-2-128:~$ curl https://www.google.fr
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="en"><head><meta content="Search the world's information, ideas and more. Google has many special features to help you find exactly what you're looking for." name="description"><meta content="text/html; charset=UTF-8" http-equiv="Content-Type"><meta content="/logos/doodles/2024/celebrating-james-baldwin-6753651837110181-2x.png" data-content="Celebrating James Baldwin" property="twitter:title"><meta content="Celebrating James Baldwin! #GoogleDoodle" property="tent="Celebrating James Baldwin! #GoogleDoodle" property="og:description"><meta content="summary_large_image" property="twitter:card"> property="twitter:site"><meta content="https://www.google.com/logos/doodles/2024/celebrating-james-baldwin-6753651837110181-2x.png" content="https://www.google.com/logos/doodles/2024/celebrating-james-baldwin-6753651837110181-2x.png" property="og:image"><meta content="400" property="og:image:height"><title>Google</title><script nonce="RS6qzbVxFND-Yz5y15SFcw!">(function(){var _g= ,_KEPI:'0,1365468,206,4804,1132070,1963,668573,327179,380776,44798,23792,12319,2815,14765,4998,17075,38444,2872,2891,3926,213,7615,60 6916,2652,4,57402,2215,2986,24067,6627,7596,1,11943,30211,2,16395,342,23024,6699,31122,4568,6256,24673,33064,2,2,1,10956,15676,8155,2 99,36746,3801,364,2048,3019,3030,11151,4665,1804,7734,6626,1,11471,21250,1632,8842,4653,42867,5210139,2,365,1104,625,5992871,2806666 3,1603,3,262,3,234,3,2121276,2585,22636438,392913,8163,4636,8489,8027,8638,13022,4427,1225,9352,5878,6478,8128,2849,7650,2,5885,4570, 4410,4226,845,50,884,285,2167,4,2250,2527,452,4454,2752,4,6,2816,288,7744,2,242,2,521,2491,2158,245,288,5,2851,4195,4,5

```

Voilà la preuve lors de la déconnexion :

```

ubuntu@ip-172-16-2-128:~$ exit
logout
Connection to 172.16.2.128 closed.
ubuntu@ip-172-16-1-145:~/key$ exit
logout
Connection to 54.145.87.19 closed.
ubuntu@ip-172-16-1-135:~/TP3/tp1_asacode$ 

```

Effectivement les ressources créées ont été détruit :

```

Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

aws_instance.tp_devops_private_instance: Destroying... [id=i-0c82d06514c02e462]
aws_route_table_association.public: Destroying... [id=rtbassoc-0b51c6c67aabf2d9c]
aws_default_route_table.private_route_table: Destroying... [id=rtb-0da0580781a8d]
aws_instance.tp_devops_nat_instance: Destroying... [id=i-0cd11338af9302528]
aws_default_route_table.private_route_table: Destruction complete after 0s
aws_route_table_association.public: Destruction complete after 1s
aws_route_table.public_route_table: Destroying... [id=rtb-04b0efeee0c1c3ab8]
aws_route_table.public_route_table: Destruction complete after 0s
aws_internet_gateway.tp_devops_igw: Destroying... [id=igw-02bd24434ba2cb5ec]

```

Nous avons demandé à chatgpt de répondre à notre question voici sa réponse :

Dans le fichier variable.tf :

```
variable "vpc_cidr" {
  description = "CIDR block for the VPC"
  default    = "172.16.0.0/16"
}

variable "public_subnet_cidr" {
  description = "CIDR block for the public subnet"
  default    = "172.16.1.0/24"
}

variable "private_subnet_cidr" {
  description = "CIDR block for the private subnet"
  default    = "172.16.2.0/24"
}
```

Dans provider.tf on aura :

```
provider "aws" {
  region = "us-west-2" # Change to your desired AWS region
}
```

Dans main.tf nous aurons :

```
resource "aws_vpc" "TP_DevOps" {
  cidr_block = var.vpc_cidr
  tags = {
    Name = "TP_DevOps"
  }
}

resource "aws_subnet" "TP_DevOps_Public" {
```

```

vpc_id    = aws_vpc.TP_DevOps.id

cidr_block = var.public_subnet_cidr

map_public_ip_on_launch = true # Important for public subnet

tags = {

  Name = "TP_DevOps_Public"

}

}

resource "aws_subnet" "TP_DevOps_Private" {

vpc_id    = aws_vpc.TP_DevOps.id

cidr_block = var.private_subnet_cidr

tags = {

  Name = "TP_DevOps_Private"

}

}

resource "aws_instance" "Nat_JumpHost" {

ami      = "ami-123456" # Replace with the latest Ubuntu Server AMI in your region
instance_type = "t2.micro" # Adjust as needed
subnet_id   = aws_subnet.TP_DevOps_Public.id
iam_instance_profile = "LabInstanceProfile"

tags = {

  Name = "Nat_JumpHost"

}

}

resource "aws_instance" "PrivateHost" {

ami      = "ami-123456" # Replace with the latest Ubuntu Server AMI in your region
instance_type = "t2.micro" # Adjust as needed
subnet_id   = aws_subnet.TP_DevOps_Private.id

```

```
iam_instance_profile = "LabInstanceProfile"
```

```
tags = {  
    Name = "PrivateHost"  
}  
}
```

Voici les explications :

Identifiants AMI : Remplacez "ami-123456" par les identifiants AMI réels pour Ubuntu Server dans votre région AWS. Vous pouvez trouver ces identifiants dans la console de gestion AWS ou via l'AWS CLI.

Types d'Instances : Cet exemple utilise t2.micro pour les instances NAT et privées. Ajustez cela en fonction de vos besoins et des ressources disponibles chez AWS.

Configuration NAT : Cet exemple crée une instance NAT manuellement. Alternativement, vous pourriez utiliser une Gateway NAT, qui est un service géré et généralement recommandé pour les environnements de production.

Groupes de Sécurité et Tables de Routage : Cet exemple n'inclut pas les configurations de groupes de sécurité et de tables de routage. Vous devrez les configurer pour contrôler le trafic vers/depuis l'instance NAT et entre les sous-réseaux.

Profil IAM : Le script suppose que le profil IAM LabInstanceProfile existe déjà et possède les autorisations nécessaires.

Région : Modifiez la région dans le fichier provider.tf selon vos besoins.

Après avoir configuré ces fichiers, exécutez terraform init pour initialiser le projet Terraform, terraform plan pour voir les changements d'infrastructure prévus, et terraform apply pour appliquer les changements à votre compte AWS.

Assurez-vous toujours de revoir le plan de Terraform avant d'appliquer, pour éviter des changements non intentionnés dans votre infrastructure.

Utilisation de l'AMI du TP2 :

```

aws_vpc_tp_devops_vpc: Creating...
aws_vpc_tp_devops_vpc: Creation complete after 1s [id=vpc-0d135ae938203e795]
aws_internet_gateway_tp_devops_igw: Creating...
aws_subnet_tp_devops_public_subnet: Creating...
aws_security_group.public: Creating...
aws_internet_gateway_tp_devops_igw: Creation complete after 1s [id=igw-01a4dd90b3c5f13c8]
aws_route_table.public_route_table: Creating...
aws_route_table.public_route_table: Creation complete after 1s [id=rtb-0ccfa22ec74f8c6ad]
aws_security_group.public: Creation complete after 2s [id=sg-086f1fbea2eec1972]
aws_subnet_tp_devops_public_subnet: Still creating... [10s elapsed]
aws_subnet_tp_devops_public_subnet: Creation complete after 11s [id=subnet-0d5dbbd9fdc6d968c]
aws_network_interface.nat: Creating...
aws_route_table_association.public: Creating...
aws_route_table_association.public: Creation complete after 0s [id=rtbassoc-014c3e2100dfc346d]
aws_network_interface.nat: Creation complete after 1s [id=eni-04f39799bf75d2a49]
aws_instance_tp_devops_nat_instance: Creating...
aws_instance_tp_devops_nat_instance: Still creating... [10s elapsed]
aws_instance_tp_devops_nat_instance: Still creating... [20s elapsed]
aws_instance_tp_devops_nat_instance: Still creating... [30s elapsed]
aws_instance_tp_devops_nat_instance: Creation complete after 33s [id=i-0ce1bcc2e0a8c83f]

Apply complete! Resources: 8 added, 0 changed, 0 destroyed.
  
```

Outputs:

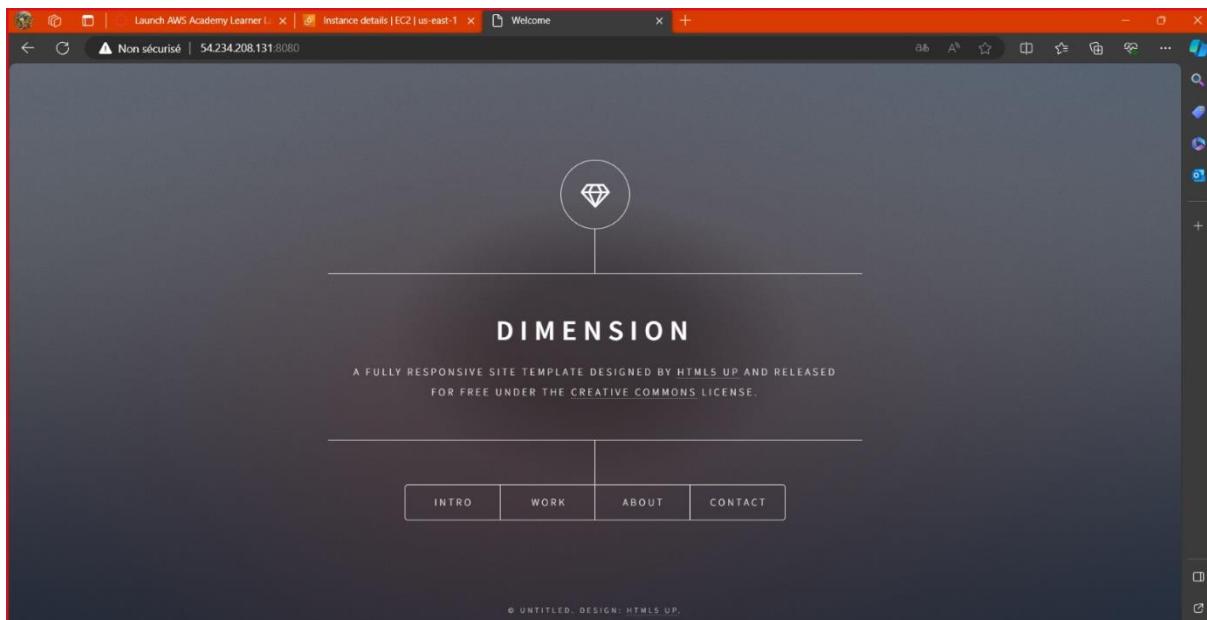
```

nat_private_ip = "172.16.1.138"
nat_public_ip = "54.234.208.131"
ubuntu@ip-172-16-1-135:~/TP3/Tp2 ascode$ 
  
```

Nous avons supprimé les ressources réseau private et on a créé l'instance :

| Parameter | Value |
|---------------------------------|--|
| Instance ID | i-0ce1bcc2e0a8c83f (WebServer) |
| Public IPv4 address | 54.234.208.131 |
| Private IPv4 address | 172.16.1.138 |
| Instance state | Running |
| Private IP DNS name (IPv4 only) | ip-172-16-1-138.ec2.internal |
| Instance type | t2.micro |
| Elastic IP addresses | - |
| VPC ID | vpc-0d135ae938203e795 (TP_DevOps) |
| AWS Compute Optimizer finding | Opt-in to AWS Compute Optimizer for recommendations. |
| Subnet ID | subnet-0d5dbbd9fdc6d968c (TP_DevOps_Public) |
| Auto Scaling Group name | - |
| IAM Role | LabRole |
| IMDSv2 | Optional |
| Platform | Linux/UNIX (Inferred) |
| AMI ID | ami-0d0bcd668a5a11702 (WebApp) |
| AMI name | WebApp-01_02_2024-15_07 |
| Monitoring | disabled |
| Termination protection | Disabled |

Nous pouvons voir que l'instance a été correctement créée et que l'image AMI est la bonne.

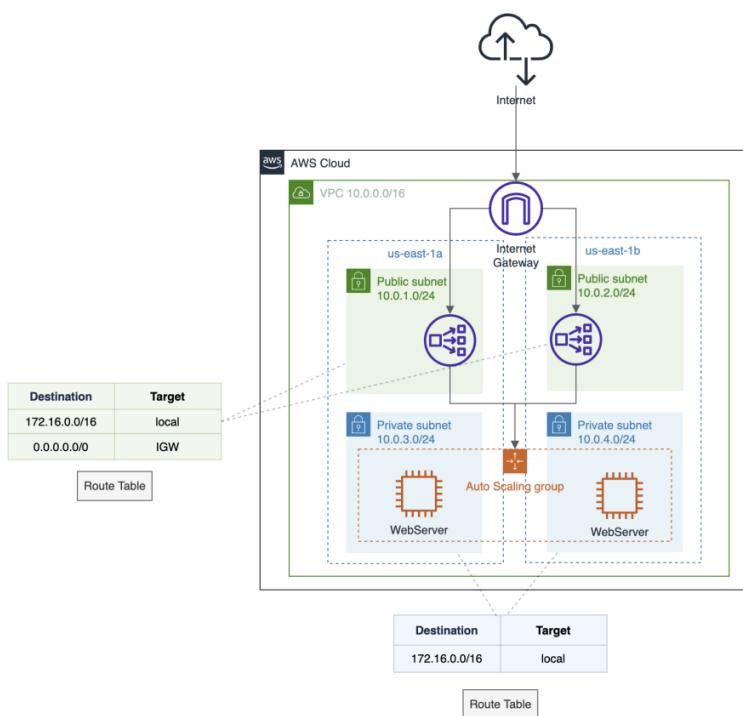


Tout est bon et fonctionne correctement.

Partie 3 Terraform

Application Web en Haute Disponibilité et réseau privé

A partir des codes terraform précédents, vous allez déployer une infrastructure hautement disponible déployée sur 2 zones de disponibilité (AZ).



Cette infrastructure comporte :

- 2 subnets publics et 2 subnets privés
 - les routes tables n'ont pas besoin d'être dupliquées
- un répartiteur de charge (Load Balancer) présent sur les 2 zones
- un Autoscaling group capable de démarrer des instances EC2 **WebServer** utilisant votre AMI dans les 2 zones
 - un Autoscaling group nécessite une launch configuration. C'est à dire les propriétés des instances EC2 à lancer.

On crée le fichier terraform qui nous permet d'avoir l'architecture demandée et on fait un terraform init plan et apply

```

aws_lb_listener.front_end: Creating...
aws_lb_listener.front_end: Creation complete after 1s [id=arn:aws:elasticloadbalancing:us-east-1:822892150932

Apply complete! Resources: 23 added, 0 changed, 0 destroyed.

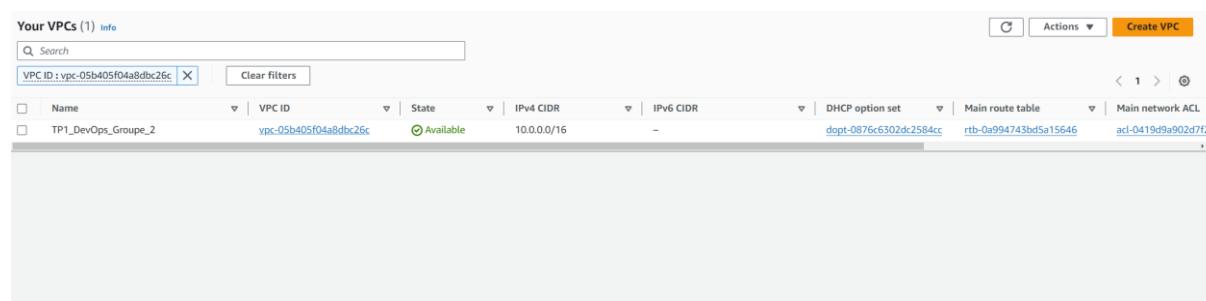
Outputs:

name_of_dns_of_load_balancer = "webloadbalancer-285627085.us-east-1.elb.amazonaws.com"
web_ami = {
  "id" = "ami-048a05937b3301985"
  "name" = "WebApp-11_02_2024-15_05"
}
ubuntu@ip-172-16-1-122:~/test$ █

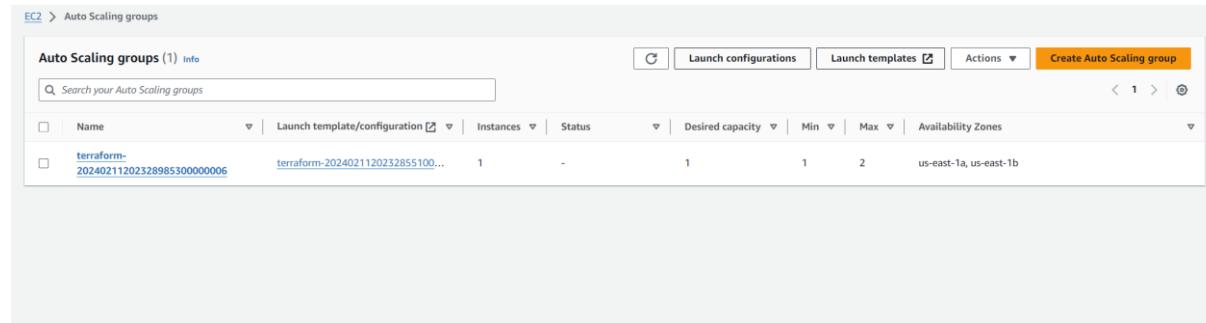
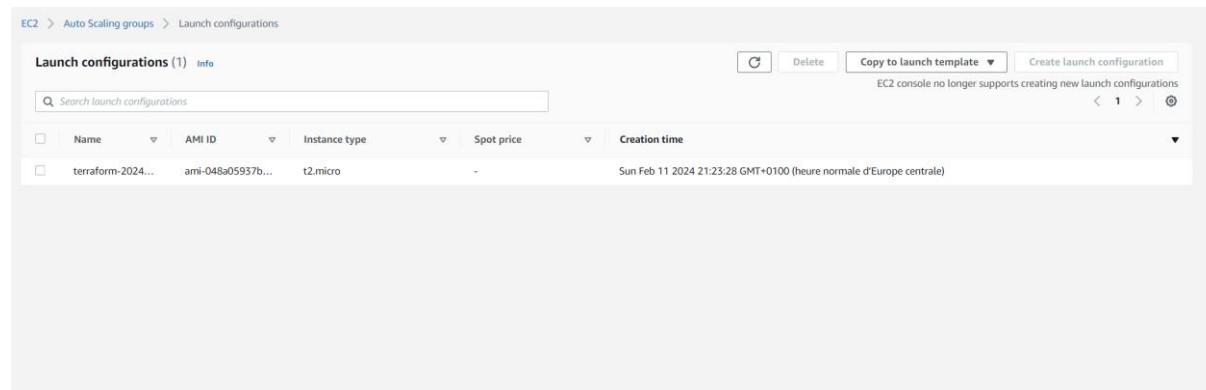
```

MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

On a l'instance Web Server qui s'est créée



| | TP3-Subnet-web_private-0 | subnet-0a87c668fadcc4016d | Available | vpc-05b405f04a8dbc26c TP1... | 10.0.3.0/24 | - | 251 | us-ei |
|---|--------------------------|---------------------------|-----------|--------------------------------|-------------|---|-----|-------|
| □ | TP3-Subnet-web_private-1 | subnet-0a69b5c0bf46de688 | Available | vpc-05b405f04a8dbc26c TP1... | 10.0.4.0/24 | - | 250 | us-ei |
| □ | TP3-Subnet-WebPublic-0 | subnet-0043eb4eb5aa1e8a7 | Available | vpc-05b405f04a8dbc26c TP1... | 10.0.1.0/24 | - | 250 | us-ei |
| □ | TP3-Subnet-WebPublic-1 | subnet-0d76a72f8fc0281c | Available | vpc-05b405f04a8dbc26c TP1... | 10.0.2.0/24 | - | 250 | us-ei |



Load balancers (1)

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

| Name | DNS name | State | VPC ID | Availability Zones | Type | Date created |
|-----------------|---|--------|-----------------------|----------------------|-------------|--------------------------------------|
| webleadbalancer | webleadbalancer-285627085.us-east-1.elb.amazonaws.com | Active | vpc-05b405f04a8dbc26c | 2 Availability Zones | application | February 11, 2024, 21:23 (UTC+01:00) |

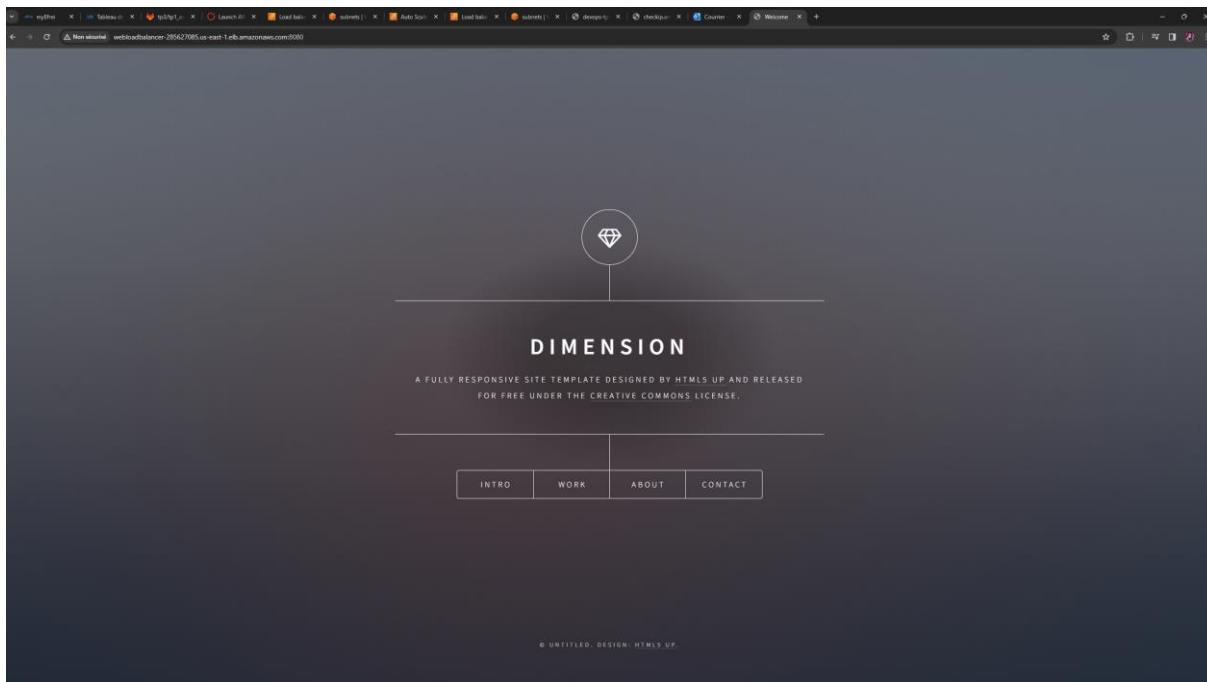
Les alarmes cloud watch

CloudWatch > Alarms

Alarms (2)

| Name | State | Last state update | Conditions | Actions |
|-----------------|-------------------|---------------------|--|-----------------|
| scale-out-alarm | OK | 2024-02-11 20:26:09 | CPUUtilization >= 70 for 2 datapoints within 10 minutes | Actions enabled |
| scale-in-alarm | Insufficient data | 2024-02-11 20:24:14 | CPU_Utilization <= 30 for 2 datapoints within 10 minutes | Actions enabled |

Pour accéder au site web on fait <http://webleadbalancer-285627085.us-east-1.elb.amazonaws.com/8080> et on a :



Questions Terraform

Terraform

- Conceptuelle :
 - **Automatisation** : Réduit le risque d'erreurs humaines comparé aux méthodes manuelles.
 - **Idempotence** : Permet des déploiements reproductibles sans effet cumulatif.
 - **Gestion de l'état** : Suit l'état de l'infrastructure, facilitant la gestion des changements.
 - **Interopérabilité** : Supporte de nombreux fournisseurs de services cloud et technologies.
 - **Infrastructure as Code** : Permet de versionner et de réviser l'infrastructure comme du code.
- Application
 - Accéder à l'URL de l'application pour vérifier la réponse.
 - Utiliser des outils de surveillance et d'alerte comme CloudWatch pour vérifier les métriques de performance.
 - Effectuer des tests d'intégration et de charge si nécessaire.
- Analyse :
 - Vérifier les logs d'erreur de Terraform pour identifier le problème spécifique.
 - Corriger les configurations erronées identifiées dans les messages d'erreur.
 - Utiliser terraform plan à nouveau pour valider les changements avant de réappliquer.
- Critique :
 - Complexité : Peut être complexe pour les infrastructures très grandes ou dynamiques.
 - Dépendance au fournisseur : Certaines fonctionnalités spécifiques au fournisseur peuvent ne pas être prises en charge.
 - Gestion de l'état : La gestion de l'état peut devenir un défi dans les équipes importantes ou pour les projets complexes.
- Extension :
 - Utiliser des outils CI/CD comme Jenkins, GitLab CI, ou GitHub Actions pour exécuter Terraform lors des déploiements.
 - Stocker l'état de Terraform de manière sécurisée, par exemple dans un backend S3 avec verrouillage d'état.
- Sécurité :
 - Utiliser des modules Terraform éprouvés et vérifiés.
 - Gérer les secrets en dehors des fichiers de configuration, en utilisant AWS Secrets Manager ou Vault.
 - Limiter les permissions IAM au strict nécessaire.
 - Utiliser des backends sécurisés pour l'état Terraform.
- Optimisation :
 - Opter pour des instances réservées ou spot afin de réduire les coûts.
 - Automatiser l'arrêt des ressources inutilisées.

- Utiliser des outils tels que AWS Cost Explorer pour surveiller et optimiser les coûts.
- Evolution :
 - Utiliser Terraform pour mettre à jour ou modifier les ressources en décrivant l'état désiré.
 - Terraform calcule automatiquement les modifications nécessaires.
- Cas d'usage :
 - Gestion d'environnements multi-cloud ou hybrides, où Terraform peut unifier la gestion de diverses infrastructures cloud.
- Réflexion :
 - Désynchronisation : Entre l'état réel et l'état géré par Terraform, menant à des erreurs lors des déploiements futurs.
 - Gestion des dépendances : Peut causer des erreurs ou des comportements inattendus dans les ressources dépendantes.

Autoscalling

- Compréhension :
 - L'autoscaling AWS ajuste automatiquement le nombre d'instances EC2 selon la demande, utilisant des politiques basées sur des indicateurs comme l'utilisation CPU. Cela permet de lancer des instances supplémentaires lors des pics de demande et d'en réduire le nombre en périodes creuses, optimisant ainsi les coûts tout en maintenant les performances et la disponibilité des applications.
- Configuration :
 - **Créer un groupe d'Auto Scaling** : Utilisez la ressource aws_autoscaling_group pour créer un groupe. Vous devrez spécifier des détails tels que le nombre désiré, minimum et maximum d'instances, ainsi que l'AMI et le type d'instance
 - **Définir les politiques d'autoscaling** : Utilisez des ressources telles que aws_autoscaling_policy pour définir des politiques basées sur des métriques spécifiques (comme l'utilisation CPU) qui déclencheront l'ajustement de la capacité.
 - **Appliquer la configuration** : Exécutez terraform init pour initialiser Terraform, puis terraform apply pour déployer votre configuration sur AWS.
- Scénarios :
 - **Applications web à trafic variable** : Pour des sites e-commerce lors du Black Friday ou des lancements de produits, où la charge peut varier considérablement.
 - **Applications SaaS** : Pour ajuster les ressources en fonction de l'augmentation ou de la diminution du nombre d'utilisateurs.
 - **Microservices et architectures orientées services** : Pour maintenir les performances des services critiques en ajustant dynamiquement les ressources.
 - **Traitements batch et calcul haute performance (HPC)** : Pour gérer les jobs nécessitant une grande quantité de ressources pour une durée limitée.
 -
- Dépannage :
 - **Vérifier les logs d'événements** : Consultez les logs d'événements du groupe d'Auto Scaling dans la console AWS pour identifier les erreurs.
 - **Examiner la configuration Terraform** : Assurez-vous que les paramètres de l'AMI, du type d'instance et des politiques d'autoscaling sont correctement configurés.
 - **Vérifier les limites de service** : Assurez-vous que vous n'avez pas atteint les limites de service pour le nombre d'instances EC2 dans la région.
 - **Contrôler les politiques IAM** : Vérifiez que les politiques IAM associées au groupe d'Auto Scaling ont les permissions nécessaires
- Optimisation :
 - **Utiliser des instances Spot** : Intégrez des instances Spot dans votre groupe d'Auto Scaling pour réduire les coûts.
 - **Diversifier les types d'instances** : Utilisez une combinaison de types d'instances pour équilibrer coût et performance.

- **Ajuster les politiques d'autoscaling :** Affinez les seuils de déclenchement pour éviter des ajustements trop fréquents ou inutiles.
- **Surveillance et ajustement :** Utilisez AWS CloudWatch pour surveiller l'efficacité et ajuster les politiques d'autoscaling en conséquence
- Sécurité :
 - **Gestion des identités et des accès :** Assurez-vous que les rôles IAM et les politiques associées au groupe d'Auto Scaling sont strictement nécessaires.
 - **Sécurité des instances :** Utilisez des groupes de sécurité pour contrôler le trafic vers et depuis les instances.
 - **Gestion des clés :** Utilisez AWS Key Management Service (KMS) pour gérer les clés de chiffrement utilisées par les instances.
- Évaluation :
 - **Performance :** Analysez les métriques de performance comme l'utilisation CPU, la latence et le taux d'erreur pour évaluer si les instances répondent bien aux demandes.
 - **Coût :** Examinez les coûts associés au groupe d'Auto Scaling pour s'assurer qu'ils restent dans le budget tout en répondant aux besoins.
 - **Disponibilité :** Vérifiez la disponibilité et la fiabilité des applications pour s'assurer que les objectifs de niveau de service (SLA) sont atteints.
 - **Adaptabilité :** Évaluez la capacité du système à s'adapter aux changements de charge sans intervention manuelle excessive.
- Mise à jour :
 - **Utilisation de lancements pilotés :** Mettez à jour l'AMI dans une configuration de lancement et utilisez le remplacement progressif pour tester les nouvelles instances avant de les déployer à grande échelle.
 - **Déploiement bleu/vert :** Créez un nouveau groupe d'Auto Scaling avec les nouvelles AMI et basculez progressivement le trafic du groupe ancien au nouveau.
 - **Auto Healing :** Utilisez la fonctionnalité d'auto healing d'AWS pour remplacer automatiquement les instances défectueuses avec des instances utilisant la nouvelle AMI.

AMI

- Optimisation :
 - Choisir un système d'exploitation spécialement conçu pour l'efficacité, comme Amazon Linux, favorisant une meilleure gestion des ressources.
 - Adapter la configuration du noyau pour maximiser l'utilisation des ressources disponibles.
 - Ajuster les configurations réseau et déployer un suivi continu avec des outils comme CloudWatch pour une surveillance efficace.
 - Personnaliser les ressources de l'instance pour qu'elles correspondent aux exigences précises de l'application.
 - Procéder à l'installation des logiciels nécessaires, des dépendances et à la création de scripts de démarrage pour préparer l'environnement d'exécution.
 - Effectuer des tests de performance pour identifier et corriger les points de friction selon les différents cas d'usage.
 - Assurer une mise à jour régulière pour inclure les derniers correctifs de sécurité et maintenir l'intégrité du système.
 - Consigner méticuleusement le processus d'optimisation pour simplifier les interventions futures.
- Sécurité :
 - Implémenter une gestion rigoureuse des accès et des clés, en restreignant l'accès aux instances à l'aide d'IAM et de clés SSH cryptées.
 - Mettre en place une surveillance continue pour identifier toute activité anormale et prévenir les intrusions.
 - Assurer le chiffrement des données, tant au repos qu'en transit, pour protéger les informations sensibles.
 - Configurer précisément les groupes de sécurité pour limiter l'accès au réseau conformément aux besoins opérationnels.
 - Définir et appliquer des politiques d'accès minutieuses à travers la gestion des identités et des autorisations, en veillant à ne conférer que les droits nécessaires.
 - Employer des outils automatisés pour l'application des standards de sécurité et leur maintien à travers le temps.
 - Conduire des tests d'intrusion et des audits de sécurité de manière régulière pour s'assurer de la robustesse de l'AMI et de sa conformité avec les dernières recommandations de sécurité.
- Dépannage :
 - Inspecter l'AMI sélectionnée pour garantir son bon fonctionnement.
 - Examiner les logs d'instance pour détecter d'éventuelles anomalies durant l'initialisation à partir de l'AMI.
 - Vérifier les droits associés aux rôles IAM pour les instances déployées via l'AMI, afin de s'assurer de leur adéquation.
 - Surveiller les activités du groupe d'Auto Scaling pour identifier les éventuelles difficultés liées au déploiement des instances.

- Contrôler les configurations essentielles telles que l'AMI, le type d'instance, et les groupes de sécurité, pour confirmer leur correcte mise en place.

Cloud Watch

- Fonctionnement :
 - Les alarmes de surveillance CloudWatch interviennent en coordination avec les groupes d'Auto Scaling en analysant des indicateurs spécifiques tels que la consommation CPU ou le volume de trafic réseau. Si ces indicateurs franchissent les seuils établis, les alarmes CloudWatch activent des procédures d'auto-ajustement, incluant le déploiement ou la suppression d'instances EC2. Cette interaction assure une allocation des ressources ajustée en temps réel, optimisant ainsi la gestion de la capacité et la réduction des dépenses.
- Configuration :
 - **Établissement d'une politique d'auto-ajustement** : Employez la fonctionnalité `aws_autoscaling_policy` pour élaborer une stratégie d'auto-ajustement, en définissant clairement les opérations à réaliser (par exemple, augmenter ou diminuer le nombre d'instances).
 - **Mise en place d'une alarme CloudWatch** : Servez-vous de la fonction `aws_cloudwatch_metric_alarm` pour mettre en œuvre une alerte basée sur un indicateur spécifique, tel que la charge CPU, en précisant les seuils qui activeront cette alarme.
 - **Association de l'alarme à la politique d'auto-ajustement** : Connectez l'alarme CloudWatch à la politique d'auto-ajustement préalablement créée pour permettre à l'alarme de lancer les actions d'auto-ajustement.
- Sensibilité :
 - La fixation des seuils pour les alarmes CloudWatch nécessite une considération de divers éléments, comme les performances souhaitées pour l'application, les modèles de trafic, et les objectifs financiers. Il est recommandé d'examiner les données historiques pour saisir les niveaux usuels d'activité. Modifiez ensuite les seuils pour atteindre un compromis entre une réponse adéquate et la prévention des actions en cascade provoquées par des variations ordinaires
- Analyse :
 - **Charge CPU** : Révèle l'intensité de l'activité des instances.
 - **Volume de Trafic Réseau** : Entrée et sortie, pour évaluer la demande sur les ressources.
 - **Nombre de Requêtes (pour les répartiteurs de charge)** : Fournit une estimation de la demande des utilisateurs.
 - **Latence de l'Application** : Permet de contrôler la performance de l'application et d'ajuster les ressources de façon appropriée.
 - **Défaillance des Instances** : Identifie et remplace automatiquement les instances non fonctionnelles.
- Optimisation :
- L'emploi des alarmes CloudWatch facilite un ajustement proactif des ressources selon l'utilisation actuelle, améliorant l'efficience de l'auto-ajustement. Par le biais de seuils bien définis, il est possible de prévenir le surdimensionnement (et par conséquent diminuer les coûts) tout en assurant que les performances restent optimales. Les

alarmes contribuent également à une détection rapide de problèmes de performance ou de disponibilité, permettant une réaction immédiate pour préserver la qualité de service. En affinant les politiques d'auto-ajustement à partir d'une analyse approfondie des indicateurs, des économies significatives et une meilleure répartition des ressources peuvent être réalisées.

Load Balancer (Autoscaling)

- Classic Load Balancer (CLB) : Représentant la première génération de la famille, ce répartiteur est adapté aux applications de base. Il permet une distribution des requêtes tant sur le plan de la couche applicative (niveau 7) que de la couche de transport (niveau 4), bien que ses capacités soient restreintes comparées à celles des versions ultérieures.
- Application Load Balancer (ALB) : Crée pour répondre aux exigences des applications web contemporaines, cet outil fonctionne principalement au niveau 7 (HTTP/HTTPS). Il propose des options avancées, notamment le routage ciblé basé sur le contenu (comme l'URL ou le domaine hôte), une compatibilité avec les systèmes de conteneurisation et une meilleure intégration avec divers services AWS, dont ECS.
- Network Load Balancer (NLB) : Opérant au niveau 4 (TCP/UDP), ce modèle se distingue par sa haute performance et sa capacité d'évolutivité. Il convient parfaitement aux applications demandant une faible latence ou capable de gérer des volumes élevés de requêtes par seconde. Le NLB peut également attribuer des adresses IP fixes.
- Rôle Fondamental
 - Le rôle principal d'un load balancer dans un environnement avec autoscaling sur AWS est d'optimiser les performances et d'assurer une haute disponibilité en répartissant le trafic entre les instances en fonction de la charge. Il s'ajuste automatiquement lors du lancement ou de l'arrêt d'instances en raison de l'autoscaling, garantissant ainsi une répartition équilibrée du trafic. Cela optimise l'utilisation des ressources et maintient la stabilité et la réactivité de l'application, même en cas de variations importantes de la charge de travail.
- Distribution de Trafic
 - En réorientant de manière dynamique le trafic vers les instances selon leur charge et leur état de fonctionnement, le répartiteur de charge joue un rôle central dans la gestion du trafic au sein d'un groupe d'auto-échelonnement. Il permet une distribution équilibrée du trafic et une utilisation rationnelle des ressources en dirigeant les requêtes vers les instances disponibles et en interrompant le routage vers celles qui sont hors service.
- Scalabilité
 - Le répartiteur de charge facilite l'évolutivité en redistribuant automatiquement le trafic entre les instances ajoutées ou retirées. Il ajuste de manière dynamique la distribution du trafic en réponse à l'ajout ou à la suppression d'instances dues à l'auto-échelonnement, assurant ainsi une gestion efficace des ressources et une distribution homogène du trafic.
- Haute Disponibilité
 - Dans un environnement d'auto-échelonnement, le répartiteur de charge est vital pour maintenir une haute disponibilité. Il s'ajuste dynamiquement aux changements de capacité, orientant le trafic vers les nouvelles instances ajoutées et cessant de router les requêtes vers celles retirées. Cette gestion

assure une distribution équilibrée du trafic et une utilisation optimale des ressources, contribuant à la haute disponibilité de l'ensemble du système.