**Ecole d’Ingénieurs**

**Licence Professionnelle : « Informatique Appliquée »**

**2LIA**

**Compte rendu**

Titre du Projet : Mise en place de la supervision avec Zabbix.

**Réalisé par : Encadrant :** Prof. Azeddine KHIAT  
**MOUSSA DIAKITE**

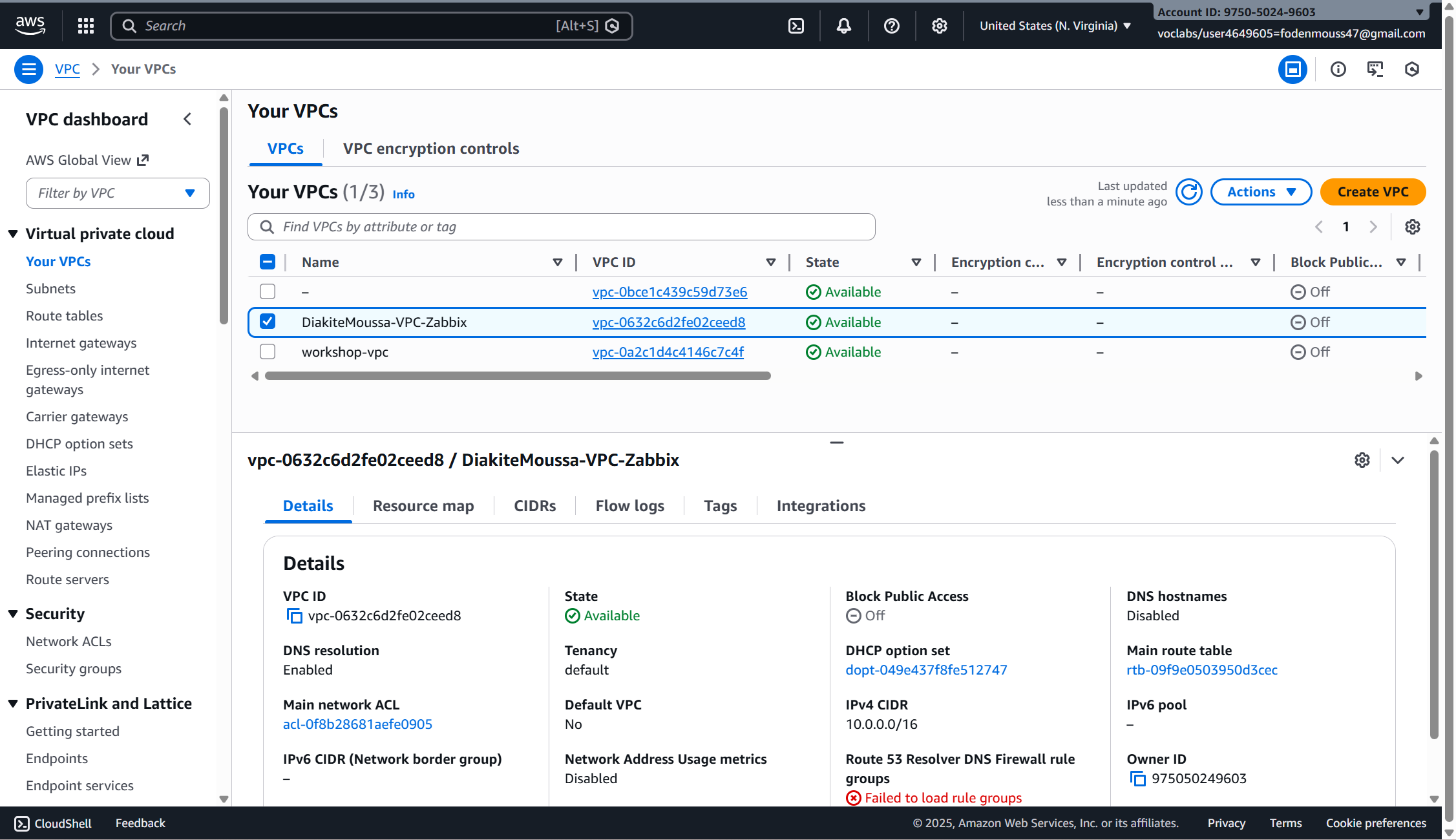
Année universitaire : 2025 / 2026

Introduction

Ce rapport décrit ensemble des étapes réalisées pour la mise en place d’une solution de supervision informatique à l’aide de l’outil Zabbix. L’objectif est de surveiller des machines Linux et Windows à distance.

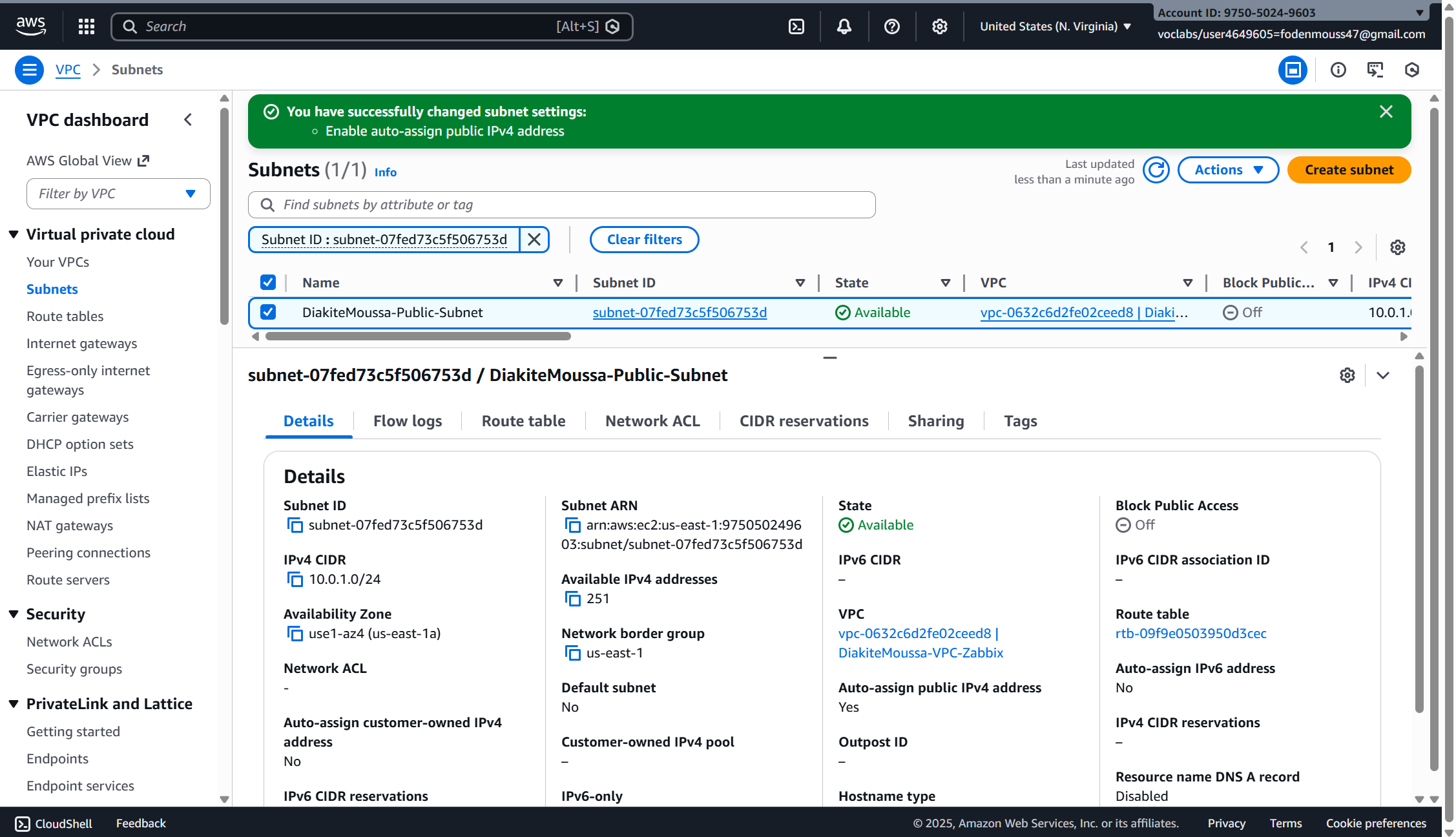
1. Creation du VPC (Reseau virtuel)

Création du VPC principal destiné à héberger l’infrastructure de supervision Zabbix.



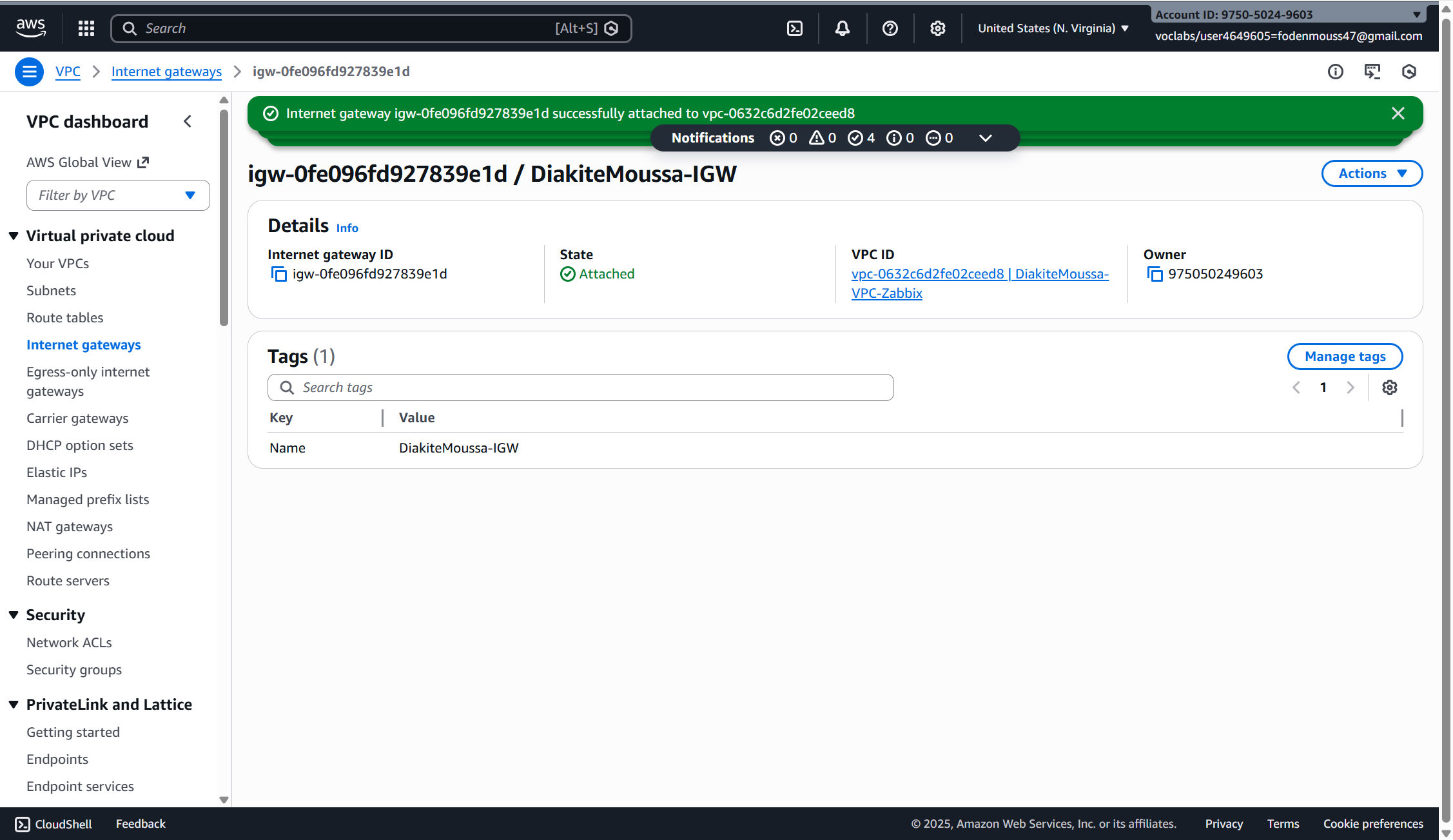
1. Creation du Subnet Public

Dans cette étape, un **subnet public** a été créé afin de permettre aux ressources (comme les instances EC2) d’avoir un accès direct à Internet.



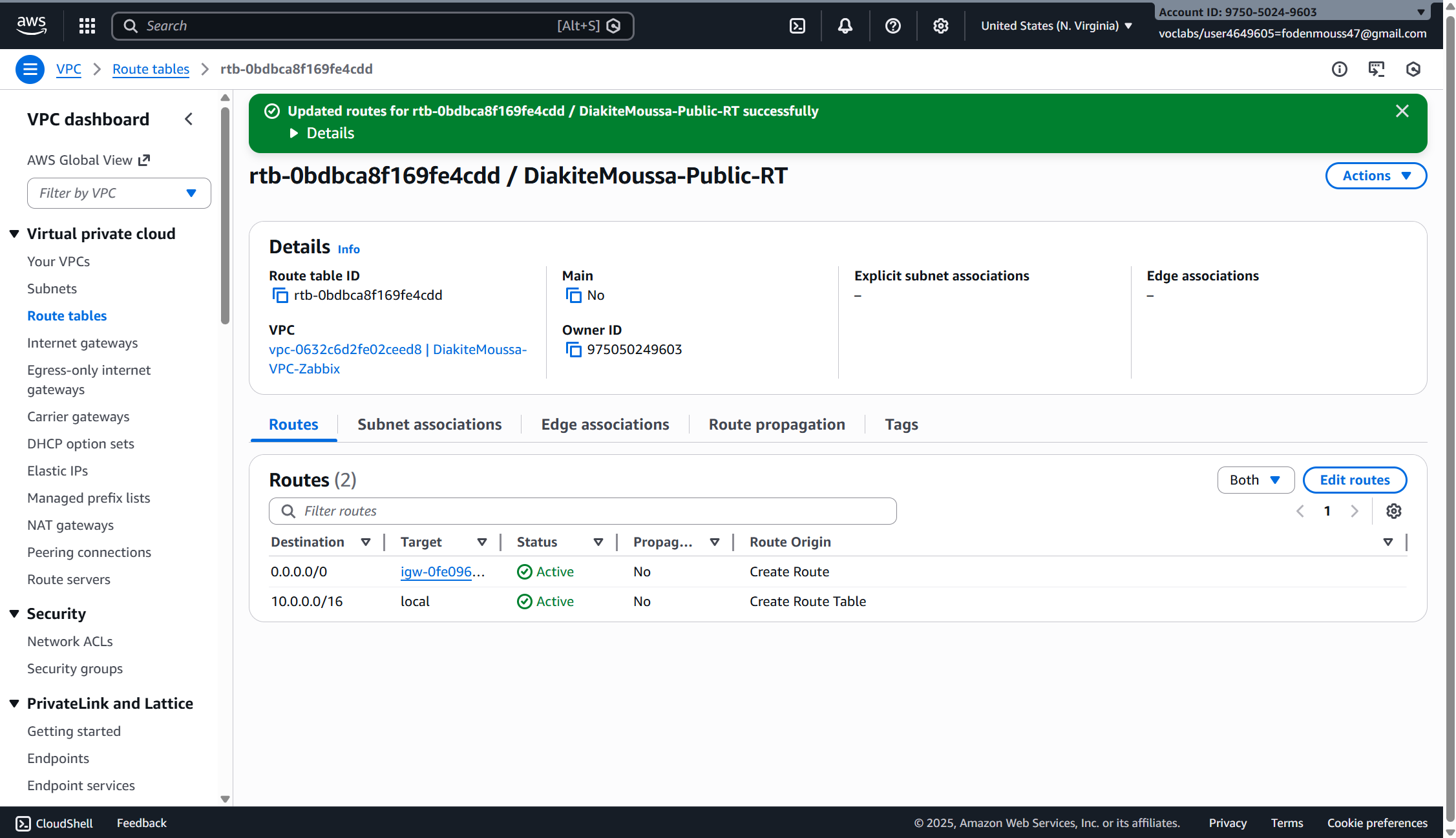
1. Internet Gateway

Dans cette étape, une **Internet Gateway (IGW)** a été créée et associée au VPC afin de permettre la communication entre le réseau privé AWS et Internet.



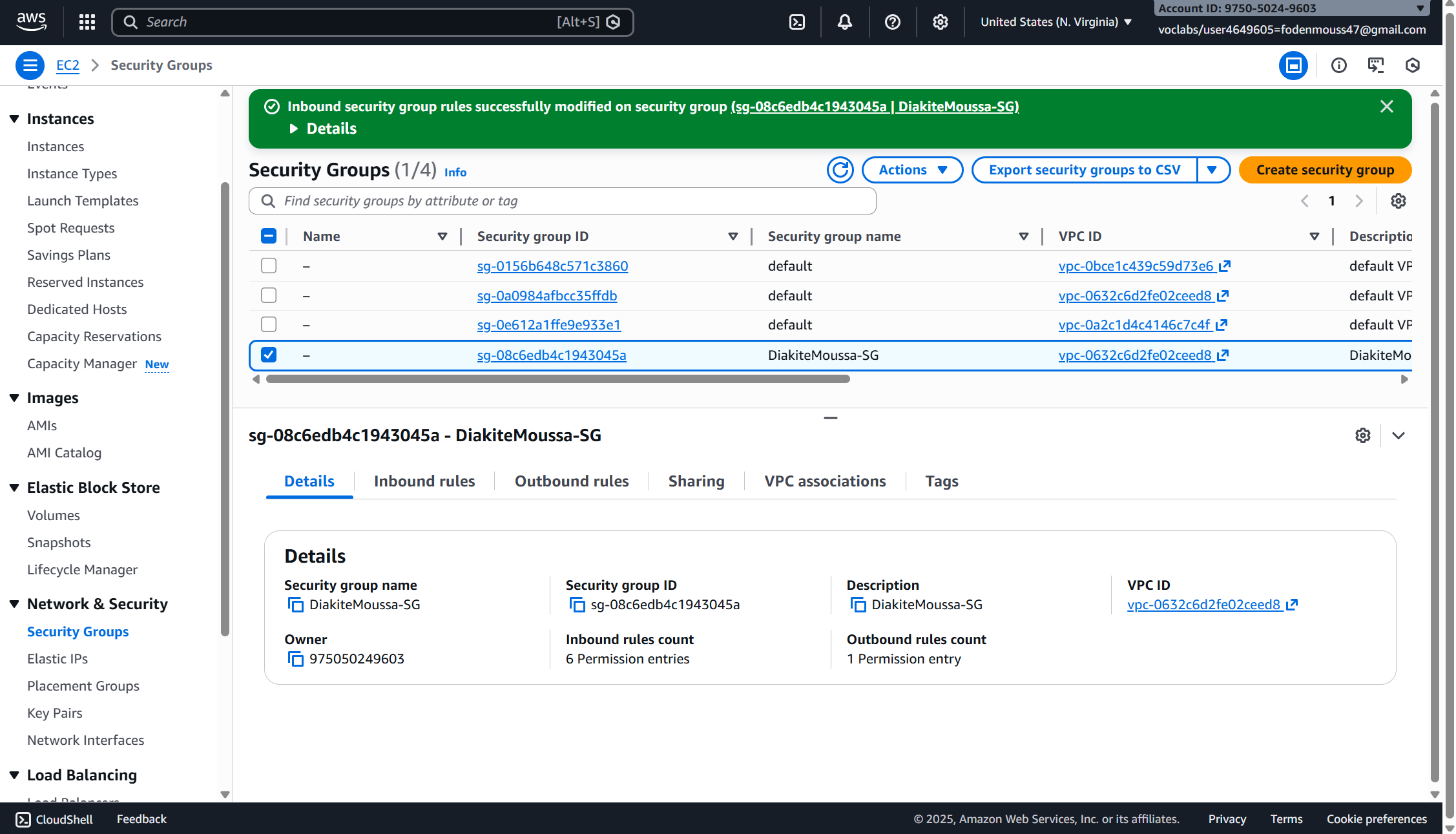
1. Table de Route

Cette étape a pour objectif de permettre au **subnet public** d’accéder à Internet en configurant une **table de routage** appropriée.



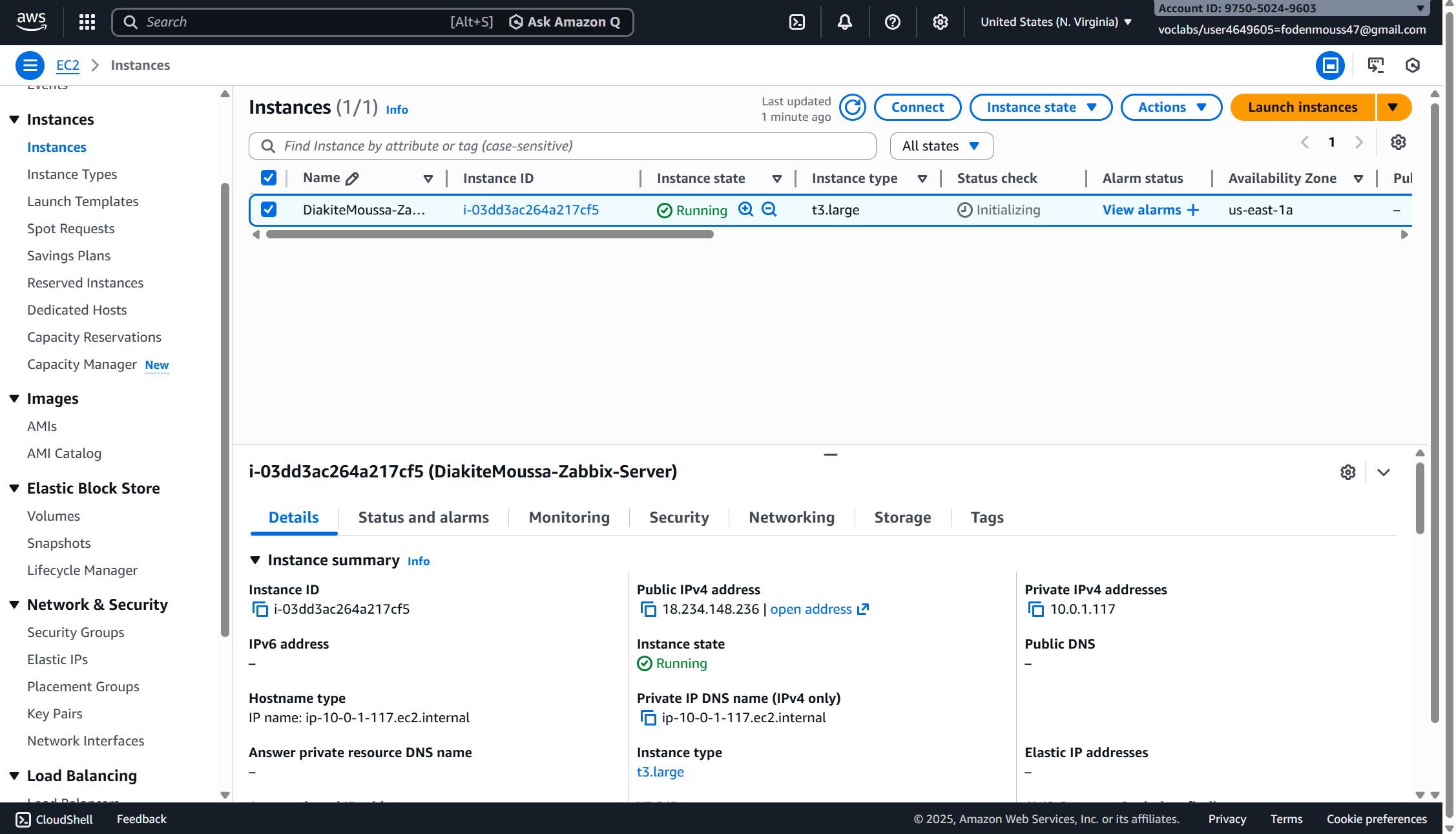
1. Security Group

Dans cette étape, un **Security Group** a été créé afin de contrôler le trafic entrant vers les instances qui hébergeront le serveur Zabbix.



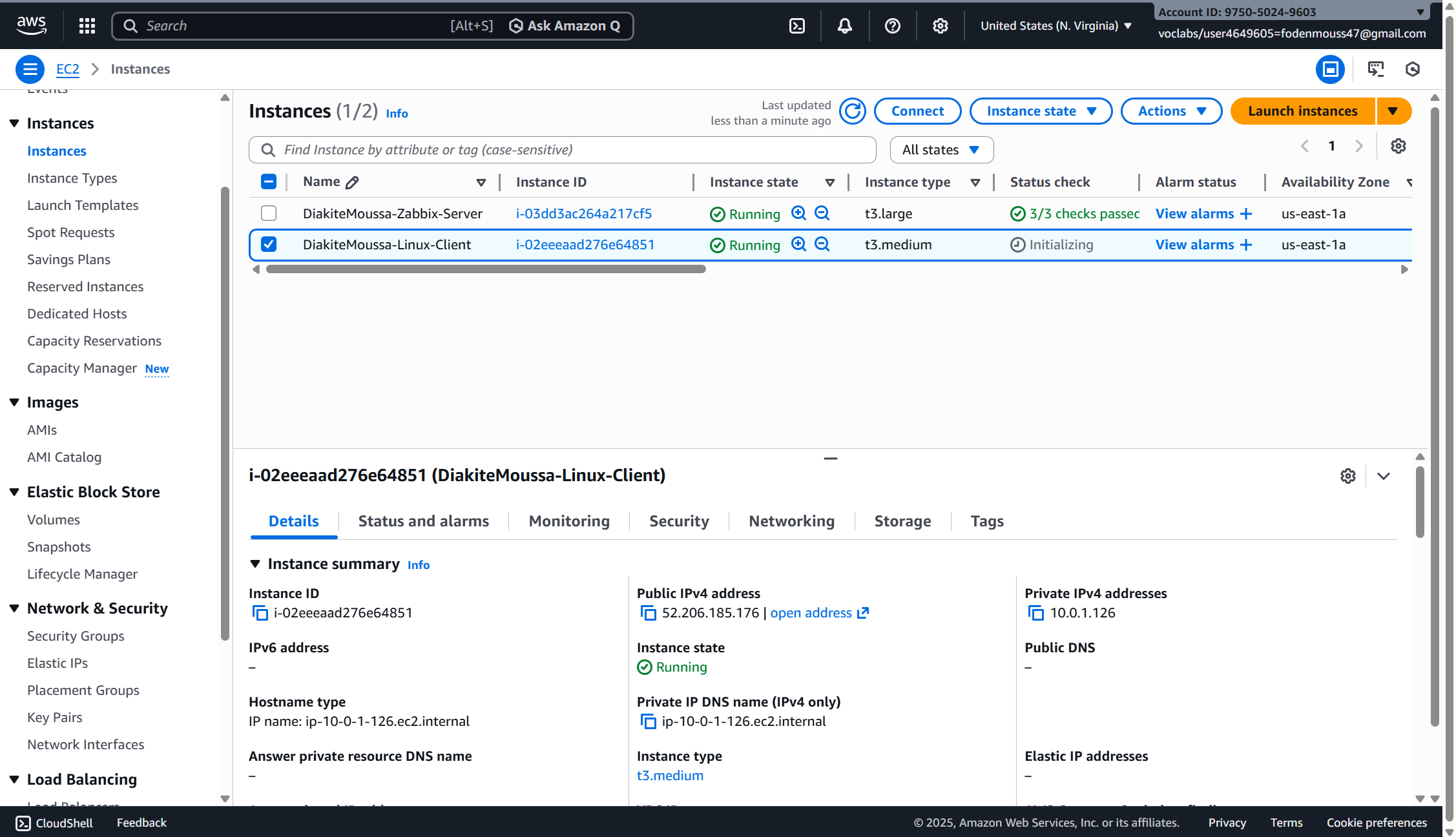
1. Instance Zabbix Server (Ubuntu)

Dans cette étape, une **instance EC2** a été créée afin d’héberger le serveur **Zabbix** avec Docker.



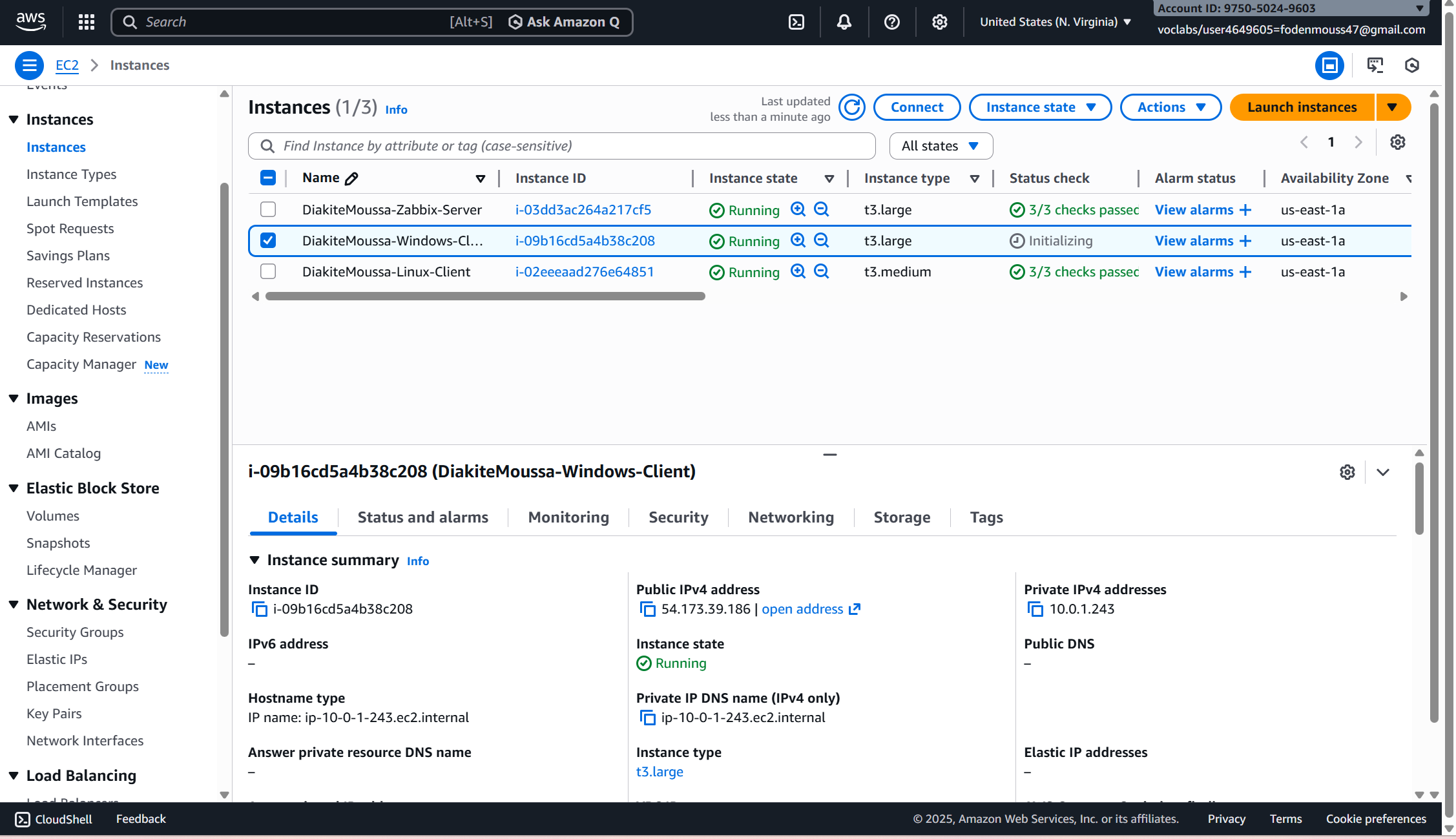
1. Instance Client Linux (Ubuntu)

Dans cette étape, une **instance EC2 client (Client Linux)** a été créée pour tester et communiquer avec le serveur Zabbix.

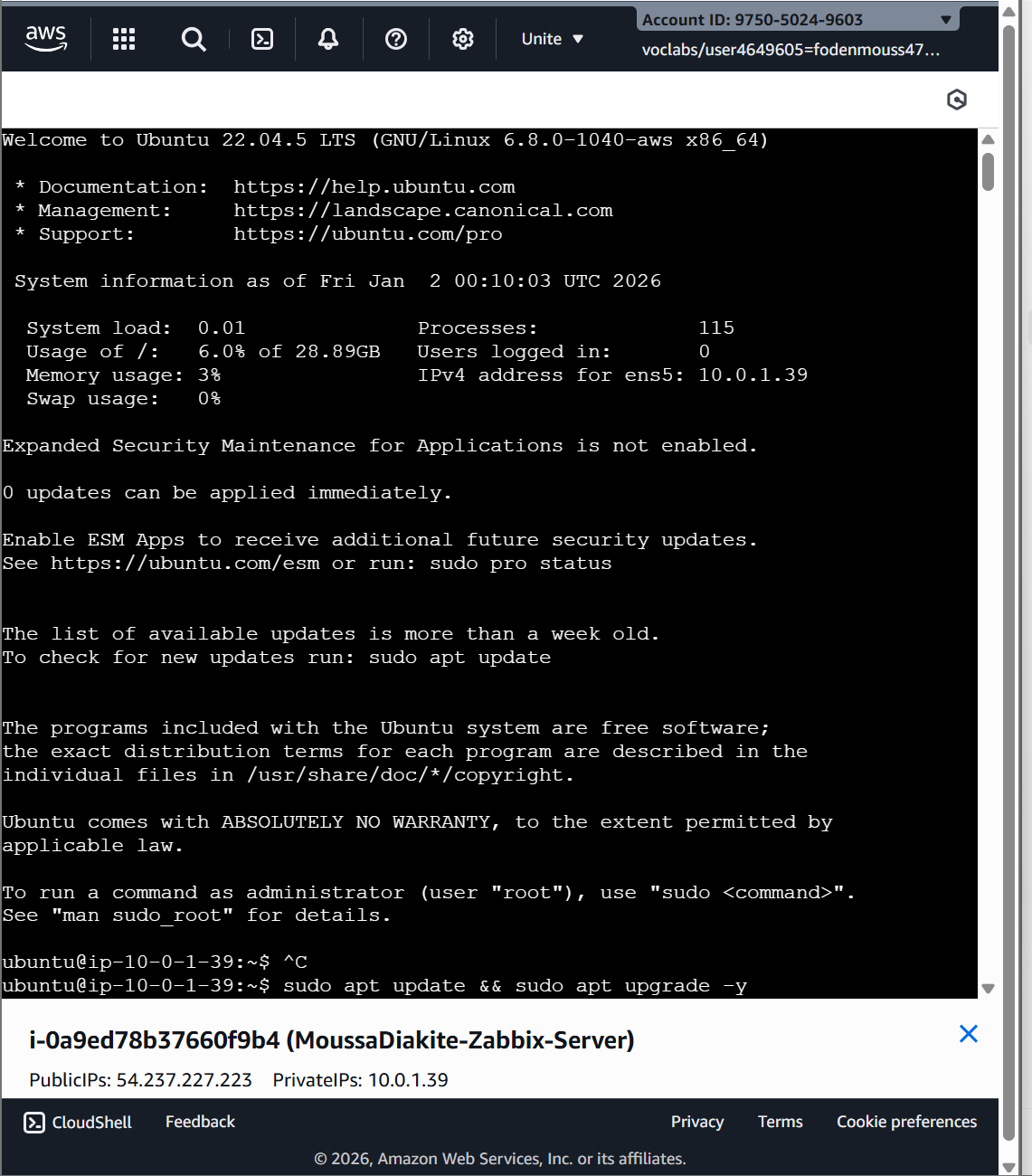


1. Instance Client Windows Server

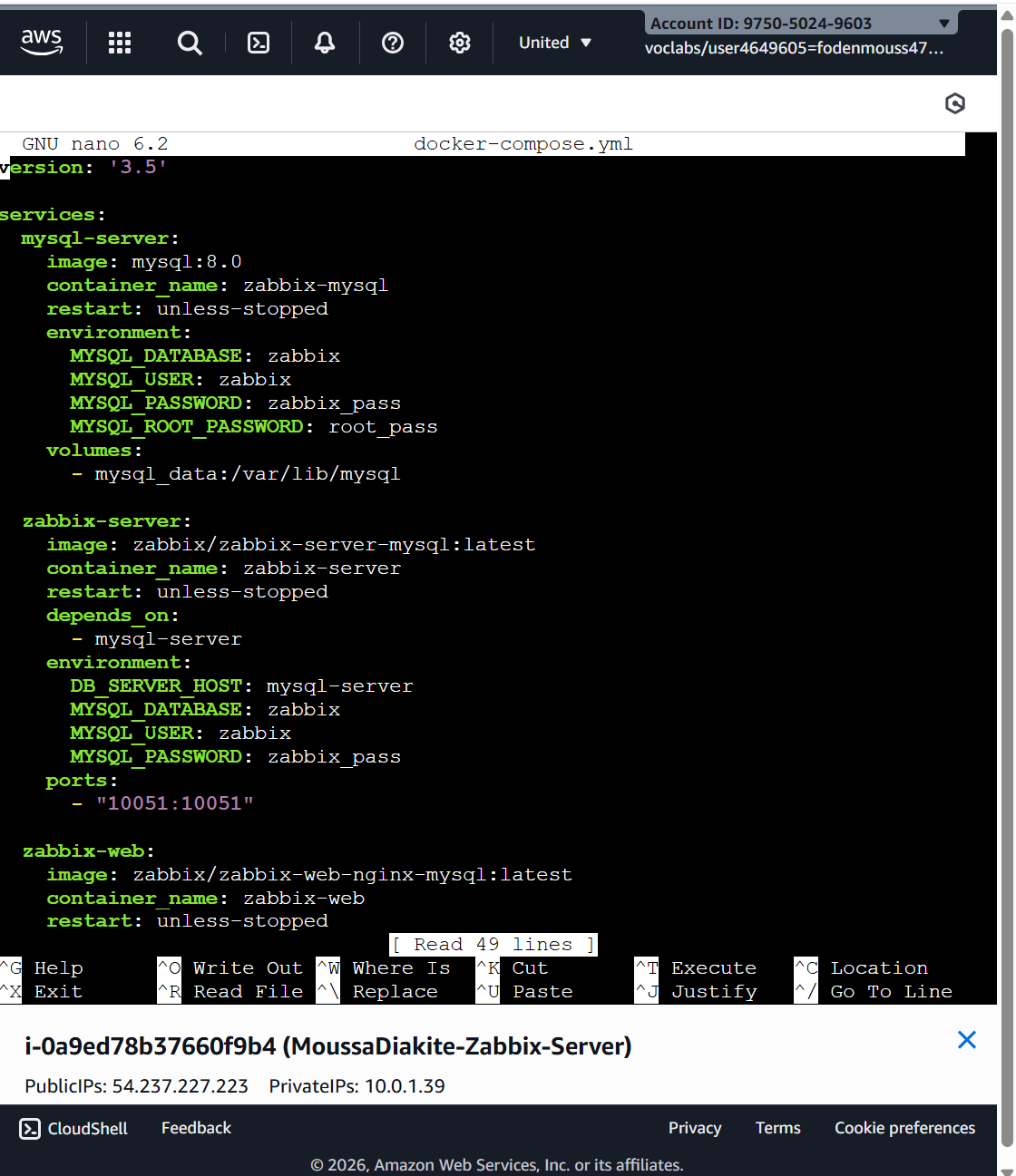
Dans cette étape, une **instance EC2 Windows Server** a été déployée afin de servir de machine cliente pour la supervision avec **Zabbix**.



1. Connection en SSH avec Zabbix

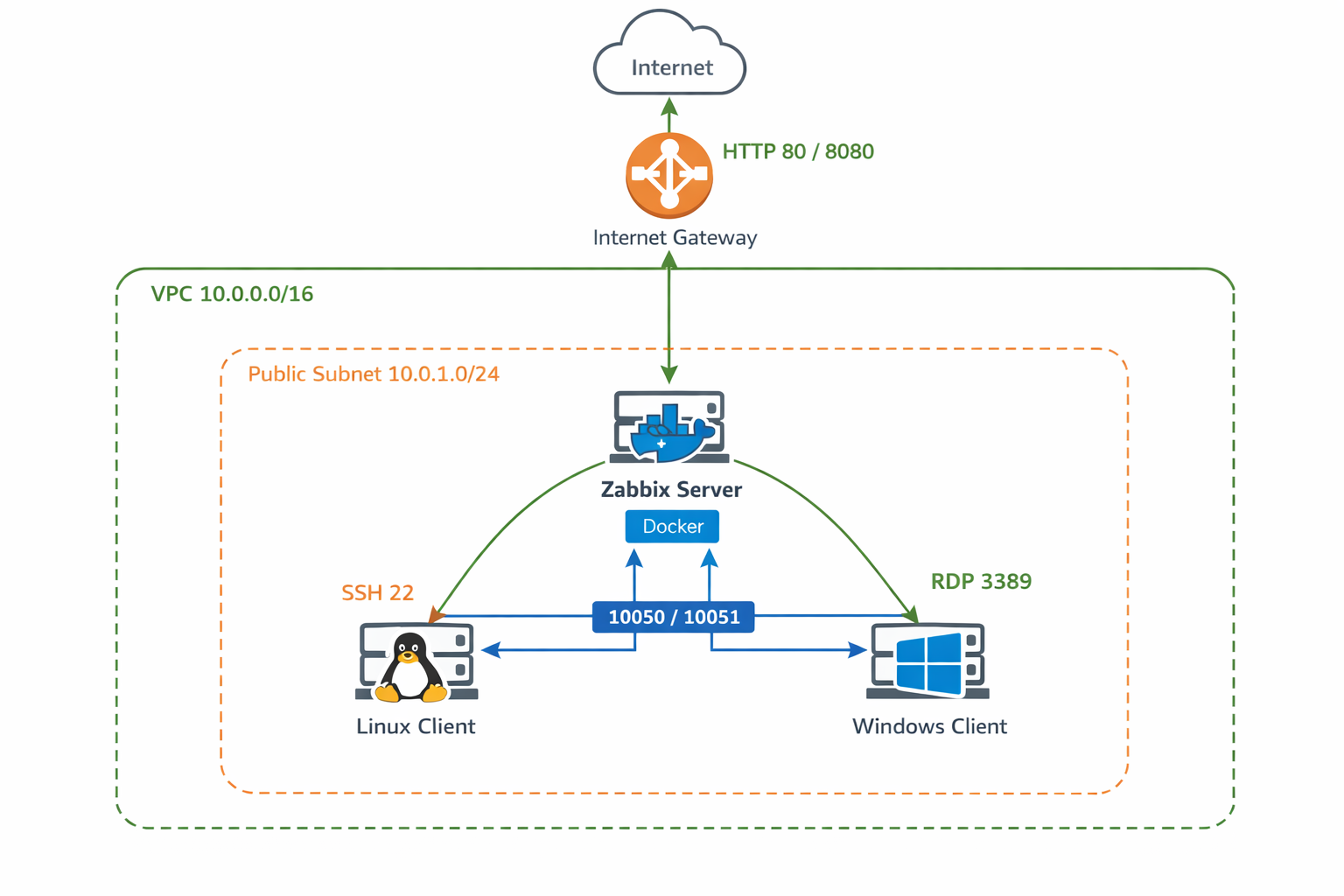


1. Install Docker et Docker Compose



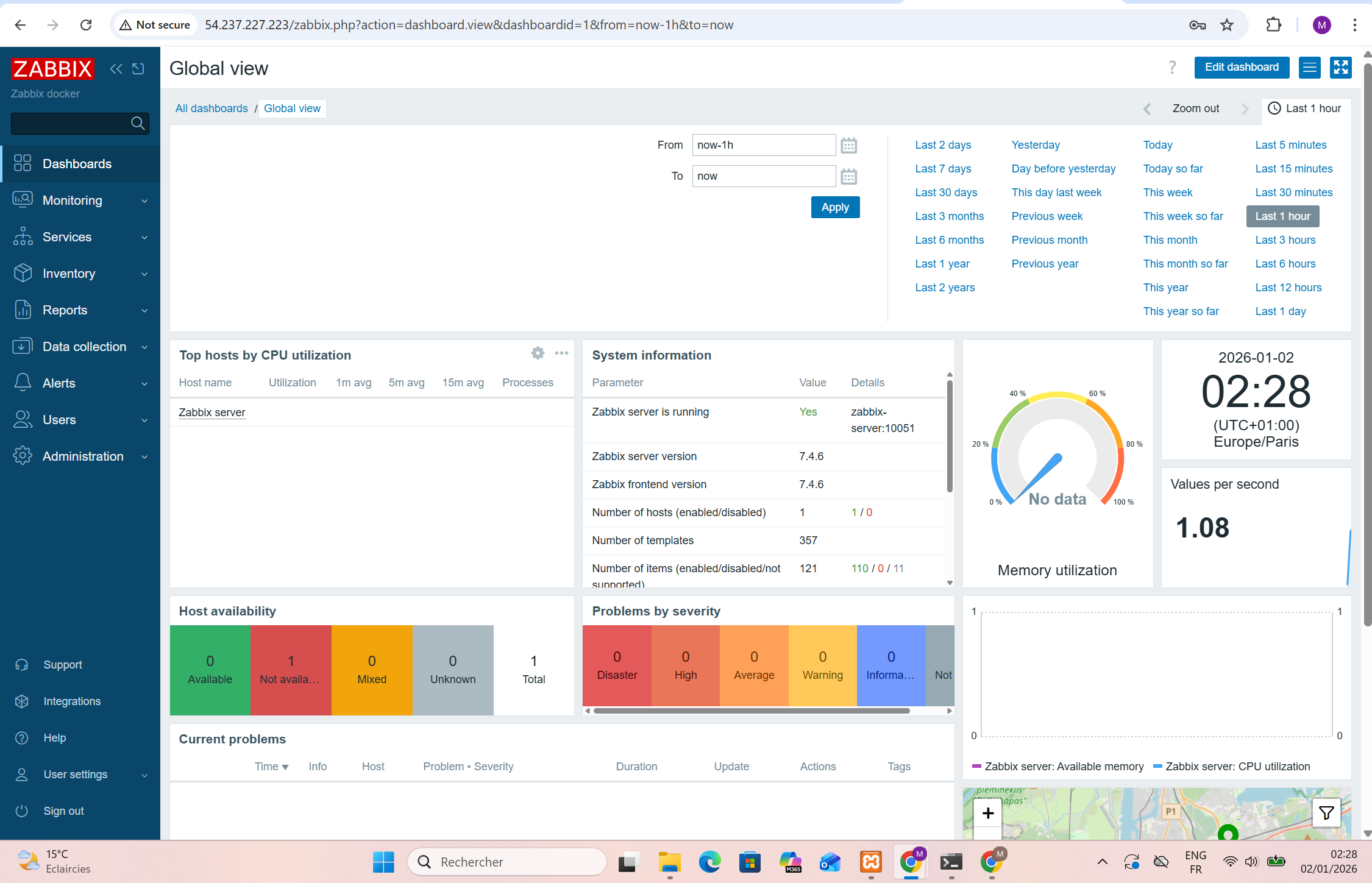
1. Architecture du Projet

Architecture est composée d’un serveur Zabbix et de deux machines clientes : une machine Linux et une machine Windows. Le serveur Zabbix centralise toutes les données collectées.

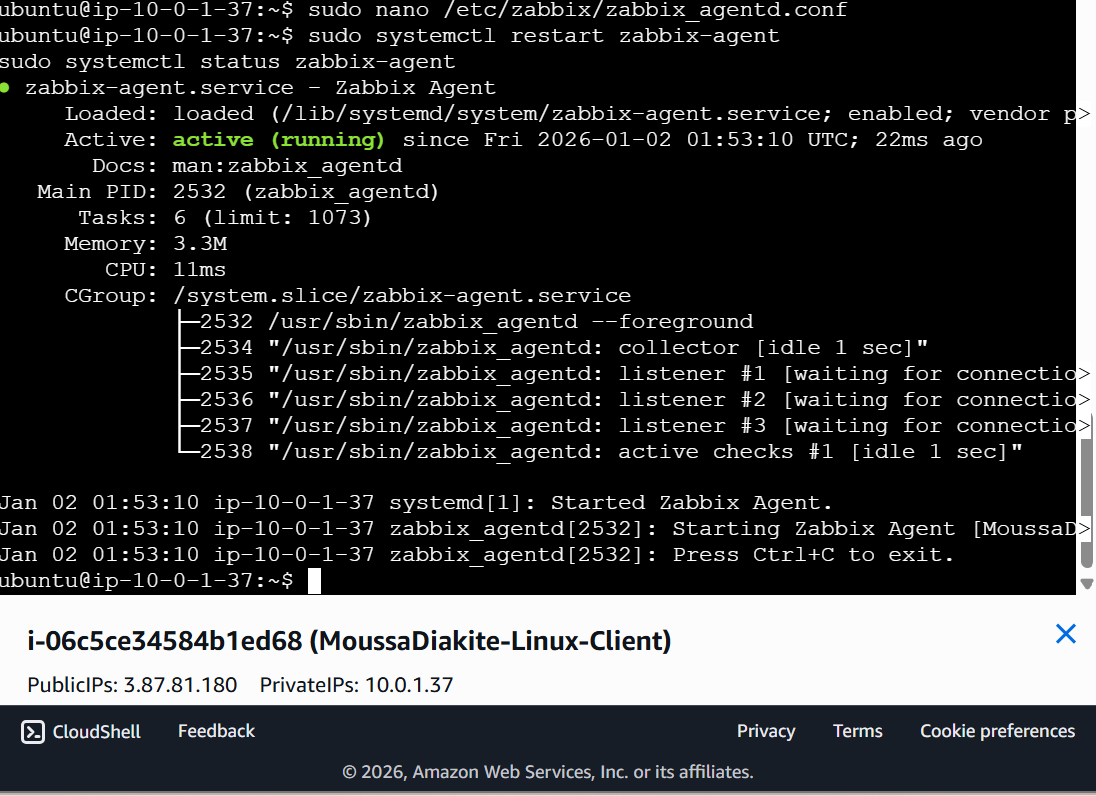


1. Installation du serveur Zabbix

Le serveur Zabbix a été installé et configuré avec succès. Les services Zabbix Server, Frontend Web et Base de données fonctionnent correctement.

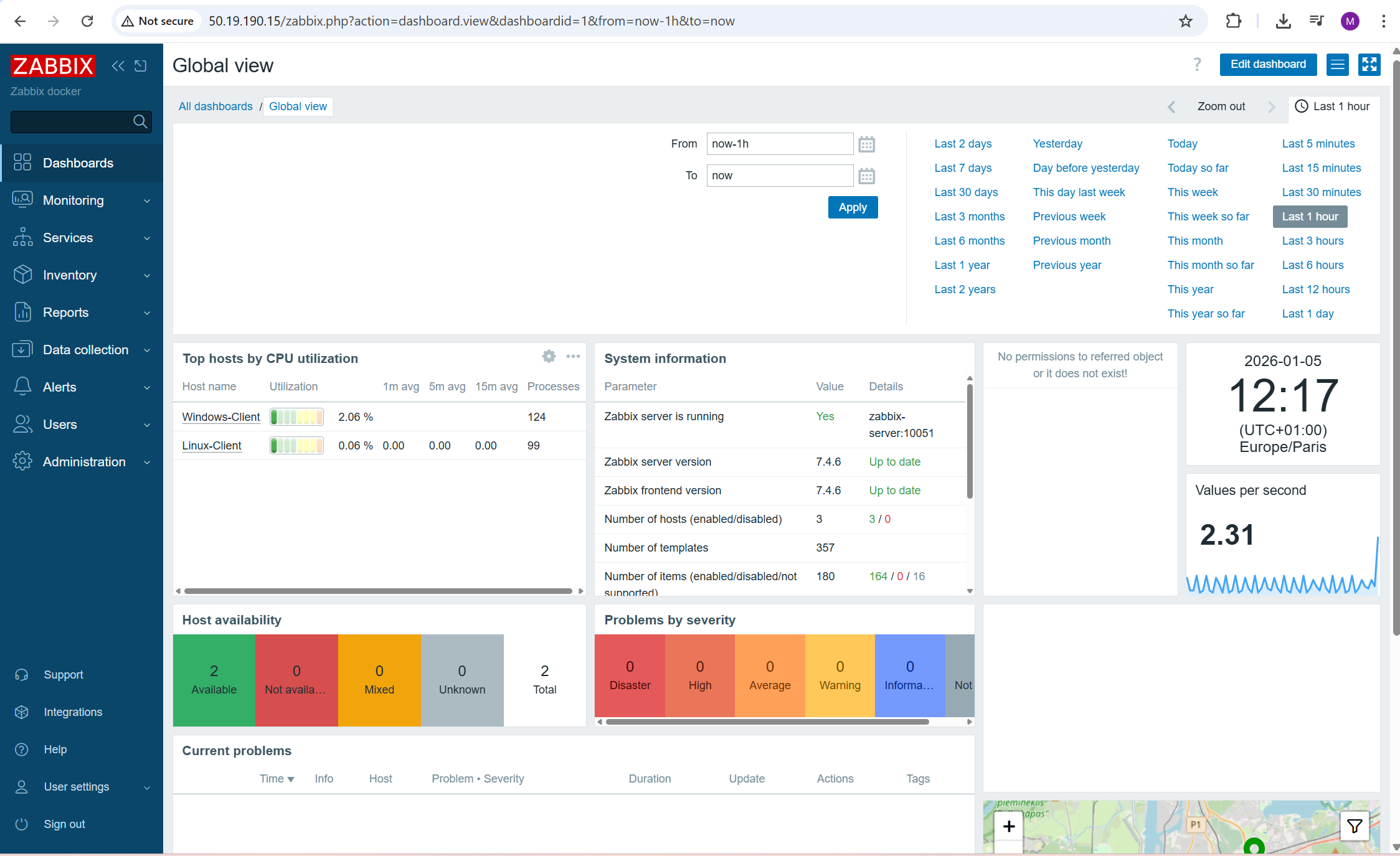


1. Demarage du l’Agent Linux



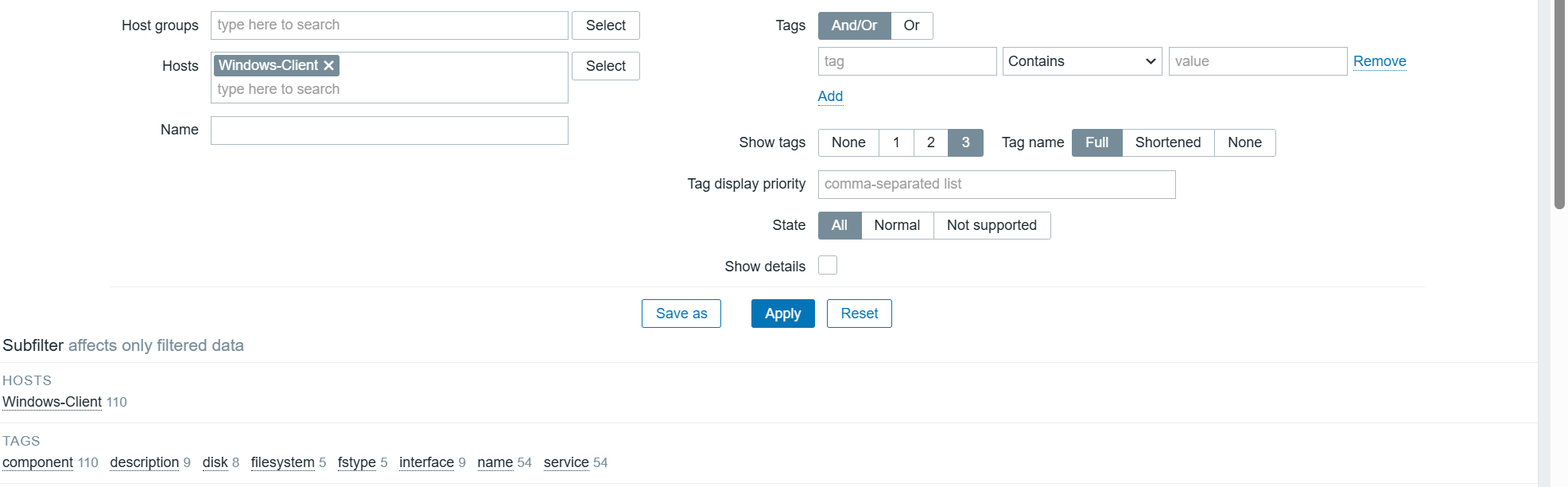
1. Les 2 hosts en vert (ZBX)

Après avoir déployé le serveur Zabbix et les instances clients, nous avons ajouté les machines **Linux** et **Windows** dans l’interface web de Zabbix afin de superviser leur état et leurs métriques.



1. Voir les Donnees

Une fois les agents installés et les clients ajoutés dans Zabbix, il est possible de **consulter en temps réel les métriques de chaque machine** depuis l’interface web.



Conclusion Générale et Difficultés Racontées

Ce projet de mise en place d’une solution de supervision avec **Zabbix** a permis d’acquérir des compétences pratiques solides en administration systèmes, en réseaux et en supervision d’infrastructures informatiques. Il a consisté à déployer un **serveur Zabbix sous Docker**, puis à superviser deux machines distantes : une instance **Linux** et une instance **Windows** hébergées sur **AWS EC2**.

Tout au long de la réalisation du projet, plusieurs **difficultés techniques** ont été rencontrées.

La première difficulté concernait la **connectivité réseau** entre les différentes instances EC2. Des erreurs telles que *“Failed to connect to your instance”* ou *“Missing active route to internet gateway”* ont été rencontrées. Ces problèmes étaient liés à une mauvaise configuration des **tables de routage**, des **Security Groups** et de l’accès Internet via l’Internet Gateway. Cette étape a permis de mieux comprendre l’importance de l’architecture réseau dans un environnement cloud.

La deuxième difficulté majeure concernait la **connexion SSH et RDP** aux instances Linux et Windows. L’accès distant a parfois été bloqué par le pare-feu ou par des règles de sécurité incorrectes. La récupération du mot de passe administrateur Windows et la configuration du **Bureau à distance (RDP)** ont nécessité plusieurs ajustements avant d’obtenir un accès stable.

Une autre difficulté importante a été rencontrée lors de l’**installation et la configuration des agents Zabbix**.  
 Sur Linux, l’agent était installé mais refusait les connexions à cause d’une mauvaise configuration du fichier zabbix\_agentd.conf, notamment les paramètres Server, ListenIP et StartAgents.  
 Sur Windows, le service **Zabbix Agent** ne démarrait pas correctement et générait des erreurs telles que *“System error 109”* ou *“Connection refused”*. Ces problèmes étaient causés par des conflits de configuration, des ports bloqués par le pare-feu Windows et des erreurs de service. L’analyse des fichiers de logs a permis d’identifier précisément les causes et de corriger la configuration.

Une autre difficulté concernait l’absence de données visibles dans la section **Monitoring → Latest data** du tableau de bord Zabbix. Bien que les hôtes apparaissaient comme *Enabled*, aucune donnée n’était affichée au départ. Ce problème était dû à des filtres actifs, à un délai de collecte des métriques et à l’état de disponibilité des agents. Après vérification des filtres et du statut des hôtes, les données ont commencé à s’afficher correctement.

Malgré ces difficultés, chaque problème rencontré a été **analysé, corrigé et documenté**, ce qui a permis d’obtenir une solution de supervision pleinement fonctionnelle. À la fin du projet, le serveur Zabbix supervise correctement les deux hôtes Linux et Windows, avec des métriques détaillées concernant le processeur, la mémoire, le disque, le réseau et les services système.

En conclusion, ce projet a été très enrichissant sur le plan technique et pédagogique. Il a permis de comprendre le fonctionnement réel d’une solution de supervision professionnelle, de renforcer les compétences en administration systèmes et réseaux, et de développer une méthode de résolution de problèmes efficace. Les difficultés rencontrées ont constitué une réelle valeur ajoutée à l’apprentissage et ont contribué à la réussite globale du projet.

Le code source complet du projet est accessible sur GitHub :  
 **Lien du dépôt :** [**https://github.com/diakitemoussa/Projet-Zabbix-AWS**](https://github.com/diakitemoussa/Projet-Zabbix-AWS)