HONORIS
EDUCATIONAL NETWORK

ECOLE MAROCAINE DES SCIENCES DE L'INGENIEUR
EMSI
Membre de
HONORIS UNITED UNIVERSITIES

Rapport - Azure

Projet

Réalisé par :

CHEHBOUNI-EDDAOUDI Othmane

Supervisee par :

.....

Table des matières

1 Création et gestion d'une machine virtuelle Windows sur Azure

1.1 Introduction

Dans ce chapitre, je présente la démarche suivie pour la création et la gestion d'une machine virtuelle Windows sur Microsoft Azure. J'ai effectué plusieurs opérations fondamentales : création de la VM, connexion via RDP, configuration de stockage, ajustement des performances, capture d'image et surveillance des métriques. À travers ce travail pratique, j'ai pu explorer les fonctionnalités essentielles du service de machines virtuelles Azure et acquérir une expérience pratique de leur utilisation dans un environnement professionnel.

1.2 Création de la machine virtuelle Windows

Ma première tâche a consisté à créer une machine virtuelle Windows selon des spécifications précises.

1.2.1 Configuration initiale

J'ai commencé par accéder au portail Azure et naviguer vers la section "Machines virtuelles". Après avoir cliqué sur "Créer", j'ai choisi l'option "Machine virtuelle Azure" pour commencer la configuration.

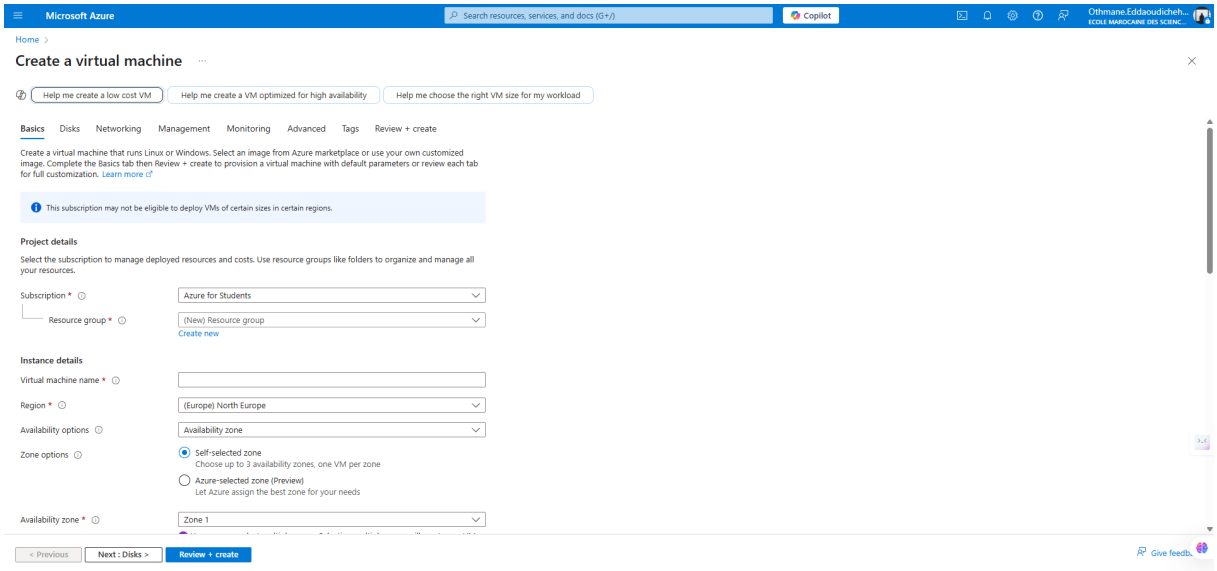

FIGURE 1 – Interface de création d'une machine virtuelle Azure

Les paramètres suivants ont été configurés conformément aux exigences :
- **Image :** Windows Server 2016 Datacenter–x64 G2
- **Nom :** VM1-Windows
- **Région :** France Central
- **Options de disponibilité :** Zones de disponibilité 1 et 2
- **Taille :** B1s (initialement)
- **OS Disk :** SSD standard (Stockage localement redondant)

1.2.2 Problème rencontré avec le nom de la VM

J'ai remarqué que le nom de la VM "VM1-Windows" respecte la convention de nommage Azure mais présente une particularité : le tiret "-" est un caractère spécial qui peut poser des problèmes dans certains contextes, notamment pour les noms DNS. Azure recommande généralement l'utilisation de caractères alphanumériques pour une compatibilité maximale. Néanmoins, dans ce contexte précis, le tiret est accepté pour les noms de machines virtuelles.

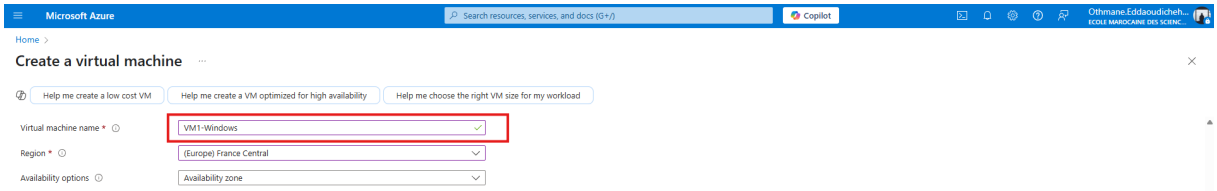

FIGURE 2 – Configuration du nom de la VM

1.2.3 Problème avec la taille B1s et les zones de disponibilité

Lors de la configuration de la taille B1s avec les zones de disponibilité 1 et 2, j'ai rencontré une erreur indiquant que cette taille n'était pas disponible dans les zones sélectionnées. Ce problème est courant car toutes les tailles de VM ne sont pas disponibles dans toutes les zones d'une région.

Pour résoudre ce problème, j'ai modifié la configuration des zones de disponibilité pour ne sélectionner que la Zone 1, où la taille B1s était disponible. Une alternative aurait été de changer la taille de la VM pour une série compatible avec les deux zones, comme D2s_v3, mais cela aurait augmenté le coût.

1.2.4 Configuration du disque OS

Pour le disque du système d'exploitation, j'ai choisi un SSD standard avec stockage localement redondant (LRS) comme spécifié. La taille standard utilisée pour Windows Server 2016 est de 127 GiB. Cette taille est prédéfinie par Microsoft Azure pour garantir des performances adéquates et suffisamment d'espace pour le système d'exploitation et les mises à jour futures.

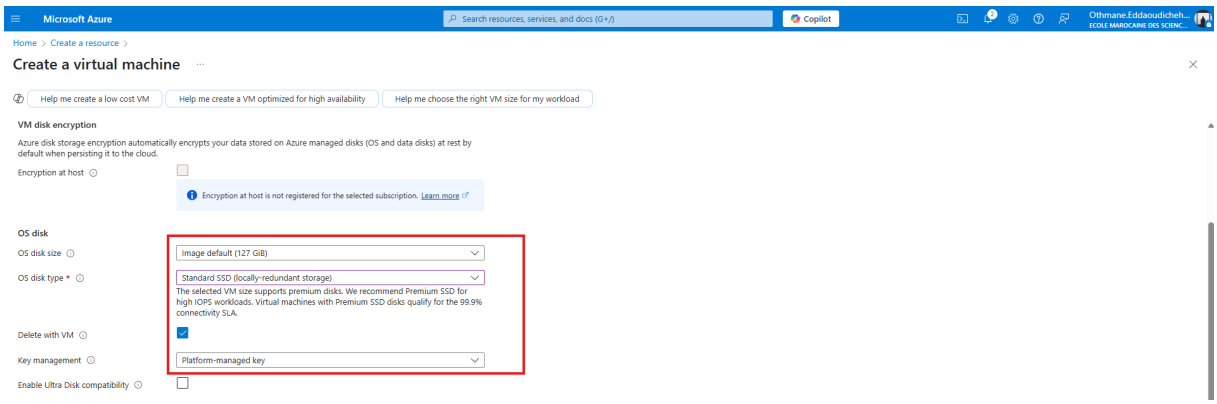

FIGURE 3 – Configuration du disque OS

1.3 Accès à la VM via RDP

Une fois la VM déployée, j'ai procédé à la connexion via le protocole RDP (Remote Desktop Protocol).

1.3.1 Obtention des informations de connexion

Depuis le portail Azure, j'ai sélectionné ma VM "VM1-Windows" puis cliqué sur "Connecter" pour obtenir les informations nécessaires à la connexion RDP.

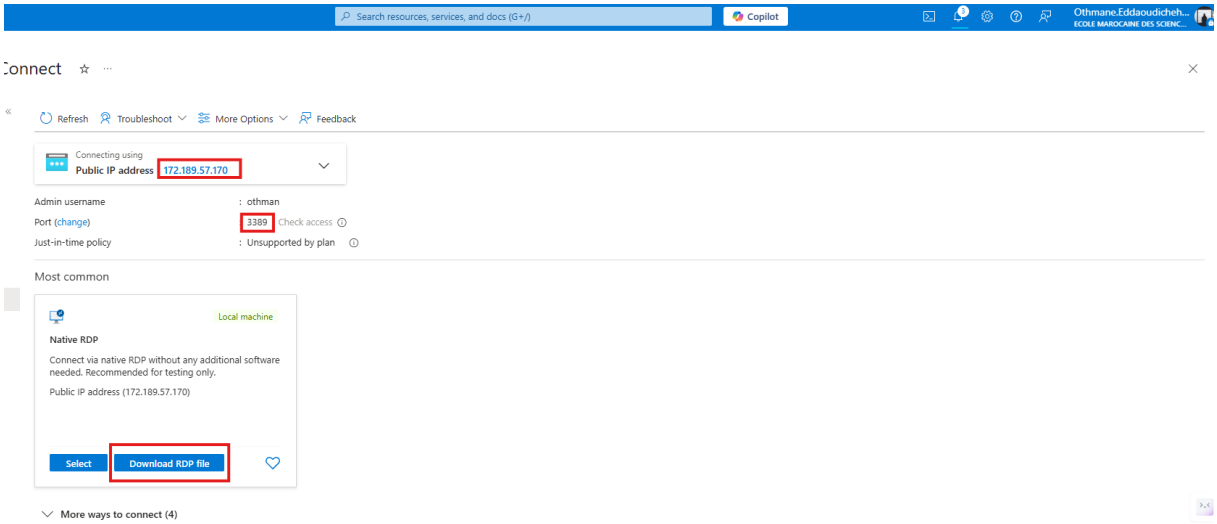

FIGURE 4 – Obtention des informations de connexion RDP

1.3.2 Problème de connexion RDP

Lors de ma tentative de connexion, j'ai rencontré une erreur "Impossible de se connecter à l'ordinateur distant". Après vérification, j'ai constaté que le problème venait des règles de sécurité réseau qui bloquaient le port 3389 (port RDP).

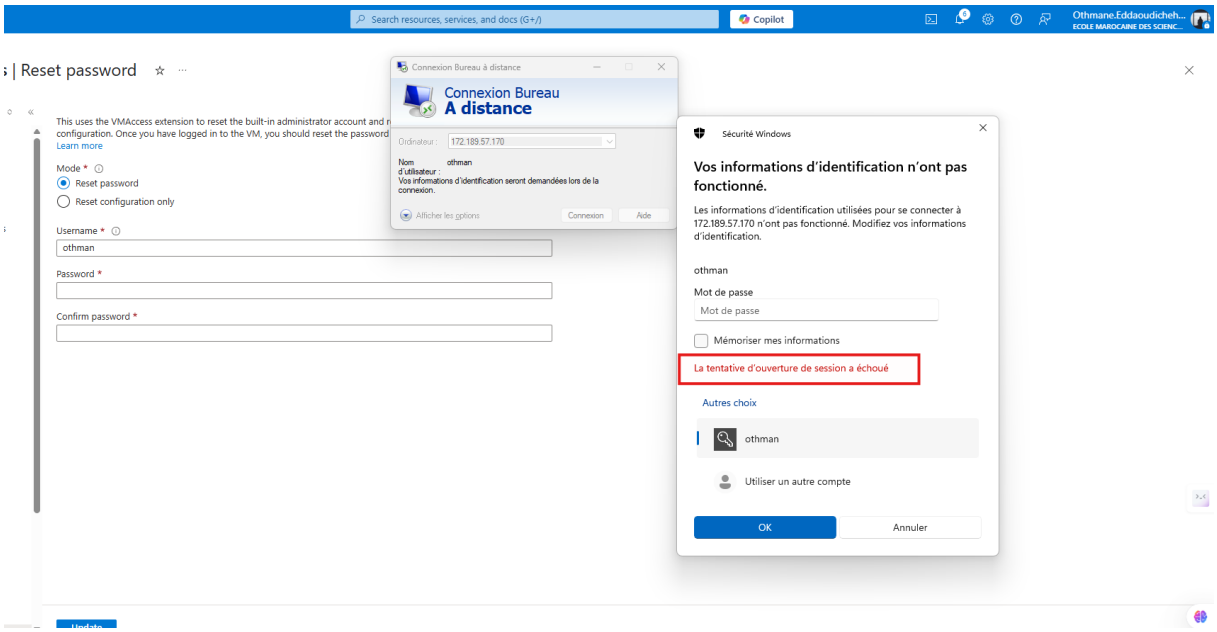

FIGURE 5 – Erreur de connexion RDP

Pour résoudre ce problème, j'ai dû :

1. Accéder à la section "Mise en réseau" de ma VM
2. Ajouter une règle entrante pour autoriser le trafic RDP (TCP/3389)
3. Attendre la propagation de la règle (environ 1 minute)
4. Réessayer la connexion

1.3.3 Exploration des disques disponibles

Après connexion réussie à la VM, j'ai ouvert l'Explorateur de fichiers puis "Ce PC" pour examiner les disques disponibles. J'ai constaté la présence d'un seul disque :

— Disque C : (127 GB) - le disque du système d'exploitation

1.4 Ajout d'un disque de données

Pour augmenter la capacité de stockage de ma VM, j'ai procédé à l'ajout d'un disque de données.

1.4.1 Procédure d'ajout du disque

Depuis le portail Azure, j'ai :

1. Navigué vers ma VM "VM1-Windows"
2. Sélectionné "Disques" dans le menu de gauche
3. Cliqué sur "Ajouter un disque de données"
4. Configuré un nouveau disque de 32 GiB avec un SSD Standard (LRS)
5. Enregistré les modifications

1.4.2 Limite du nombre de disques

Pour la taille B1s, le nombre maximal de disques de données pouvant être ajoutés est de 2. Cette limitation est due aux spécifications techniques de cette taille de VM, qui est conçue pour des charges de travail légères et économiques. Les VM de série B sont optimisées pour un coût minimal plutôt que pour les performances ou la capacité.

Si davantage de disques étaient nécessaires, il faudrait opter pour une taille de VM supérieure, comme DS2_v2 qui permet d'attacher jusqu'à 8 disques de données.

1.4.3 Initialisation du disque dans Windows

Après l'ajout du disque au niveau d'Azure, j'ai dû l'initialiser dans Windows avant de pouvoir l'utiliser. Cette étape n'est pas automatique et requiert une intervention manuelle.

J'ai procédé comme suit :

1. Ouvert "Gestion des disques" (diskmgmt.msc)
2. Initialisé le nouveau disque apparu comme "non initialisé"
3. Créé une nouvelle partition simple
4. Formaté le disque en NTFS
5. Assigné la lettre de lecteur E :

1.4.4 Problème d'initialisation du disque

Lors de l'initialisation, j'ai rencontré une erreur "Le périphérique n'est pas prêt". Ce problème est parfois observé lorsque le disque n'est pas correctement détecté par le système d'exploitation.

Pour résoudre ce problème, j'ai :

1. Vérifié dans le portail Azure que le disque était bien attaché
2. Redémarré la VM pour assurer la détection du nouveau matériel
3. Réouvert la Gestion des disques pour réessayer l'initialisation

Après ces étapes, j'ai pu initialiser et formater le disque avec succès.

1.5 Création d'un fichier sur le disque de données

Une fois le disque initialisé et formaté, j'ai créé un fichier texte pour vérifier son bon fonctionnement.

J'ai procédé comme suit :

1. Navigué vers le disque E : dans l'Explorateur de fichiers
2. Créé un nouveau document texte nommé "test-disque-donnees.txt"
3. Ajouté le contenu "Ce fichier confirme le bon fonctionnement du disque de données ajouté à ma VM Azure."
4. Enregistré le fichier

1.6 Modification de la taille de la VM

Après utilisation, j'ai constaté que les performances de la VM avec la taille B1s étaient insuffisantes pour mes besoins. Les opérations basiques présentaient une latence notable et l'interface utilisateur réagissait lentement.

1.6.1 Procédure de redimensionnement

Pour améliorer les performances, j'ai décidé de modifier la taille de la VM :

1. Depuis le portail Azure, j'ai accédé à ma VM "VM1-Windows"
2. Sélectionné "Taille" dans le menu de gauche
3. Choisi la taille B2s qui offre 2 vCPU et 4 GiB de RAM (contre 1 vCPU et 1 GiB pour B1s)
4. Confirmé le changement

1.6.2 Problème lors du redimensionnement

Lors de cette opération, j'ai rencontré un message d'avertissement indiquant que la VM allait être redémarrée, ce qui impliquait une interruption de service. Ce comportement est normal car le redimensionnement nécessite l'allocation de nouvelles ressources physiques.

Pour minimiser l'impact, j'ai planifié cette opération pendant une période de faible activité. Une fois le redimensionnement terminé, j'ai constaté une amélioration significative des performances.

1.7 Capture d'une image de la VM

La création d'une image permet de sauvegarder l'état actuel de la VM pour déployer ultérieurement des machines identiques.

1.7.1 Procédure de capture

Pour capturer une image sans perdre la VM existante, j'ai procédé comme suit :

1. Dans le portail Azure, j'ai accédé à ma VM "VM1-Windows"
2. Cliqué sur "Capturer" dans le menu supérieur
3. Configuré les paramètres suivants :
   — Nom de l'image : VM1-Windows-Image
   — Groupe de ressources : même groupe que la VM
   — Option "Capturer uniquement l'image" (pour conserver la VM)
4. Lancé la capture

1.7.2 Problème lors de la capture d'image

J'ai rencontré une erreur indiquant que la VM devait être déprovisionnée avant la capture. Cette erreur survient car Azure nécessite une préparation spécifique de la VM pour créer une image généralisée.

Pour résoudre ce problème, j'ai opté pour la création d'un instantané (snapshot) du disque OS plutôt qu'une image généralisée :

1. J'ai navigué vers "Disques" dans le portail Azure
2. Sélectionné le disque OS de ma VM
3. Cliqué sur "Créer un instantané"
4. Configuré un nom et un type de stockage
5. Créé l'instantané

Cette approche m'a permis de conserver une copie du disque sans avoir à déprovisionner la VM, ce qui aurait effacé mes configurations personnalisées.

1.8 Surveillance de la VM

Pour assurer le bon fonctionnement et optimiser les performances de ma VM, j'ai mis en place une surveillance des métriques clés.

1.8.1 Configuration de la surveillance

J'ai utilisé Azure Monitor pour surveiller ma VM avec les critères suivants :

1. **Utilisation du CPU** : Pour détecter les pics de charge et potentiellement ajuster la taille de la VM
2. **Utilisation de la mémoire** : Pour identifier d'éventuelles fuites mémoire ou besoins en RAM supplémentaire
3. **Opérations d'E/S disque** : Pour évaluer les performances du stockage
4. **Trafic réseau entrant/sortant** : Pour surveiller la bande passante utilisée

1.8.2 Création d'alertes

J'ai également configuré des alertes pour être notifié en cas de problèmes :

1. Alerte sur l'utilisation du CPU $> 90\%$ pendant plus de 5 minutes
2. Alerte sur l'espace disque disponible $< 10\%$
3. Alerte sur l'état de la VM (non disponible)

1.8.3 Problème avec l'agent de diagnostic

Lors de la configuration de la surveillance de la mémoire, j'ai constaté que les métriques n'étaient pas disponibles. Ce problème venait de l'absence de l'agent de diagnostic Azure.

Pour résoudre ce problème, j'ai :

1. Installé l'extension de diagnostic Azure sur ma VM
2. Configuré la collecte des compteurs de performance supplémentaires
3. Attendu environ 10 minutes pour que les données commencent à apparaître

1.9 Arrêt et suppression de la VM

Une fois toutes les opérations terminées, j'ai procédé à l'arrêt puis à la suppression de la VM.

1.9.1 Arrêt de la VM

Pour arrêter proprement la VM et éviter des coûts inutiles :

1. J'ai navigué vers ma VM "VM1-Windows" dans le portail Azure
2. Cliqué sur "Arrêter" dans le menu supérieur
3. Confirmé l'opération

1.9.2 Suppression de la VM et des ressources associées

Pour nettoyer complètement l'environnement, j'ai supprimé la VM et ses ressources associées :

1. J'ai sélectionné ma VM dans le portail Azure
2. Cliqué sur "Supprimer" dans le menu supérieur
3. Coché les options pour supprimer également :
   — Les disques (OS et données)
   — Les interfaces réseau
   — Les adresses IP publiques
4. Confirmé la suppression

1.9.3 Vérification de la suppression complète

Pour m'assurer qu'aucune ressource n'était laissée orpheline (ce qui pourrait générer des coûts), j'ai vérifié dans le groupe de ressources que toutes les ressources liées à ma VM avaient bien été supprimées.

1.10 Conclusion

Ce travail pratique m'a permis d'acquérir une expérience concrète dans la gestion des machines virtuelles Windows sur Azure. J'ai pu explorer les fonctionnalités essentielles : création, connexion, gestion du stockage, redimensionnement, capture d'image et surveillance.

Les problèmes rencontrés et leurs résolutions m'ont permis de développer une meilleure compréhension du comportement d'Azure et des bonnes pratiques à adopter. Par exemple, j'ai appris l'importance de vérifier la disponibilité des ressources dans les zones sélectionnées, ainsi que les limitations spécifiques à chaque taille de VM.

Cette expérience constitue une base solide pour des déploiements plus complexes, comme la mise en place d'environnements multi-VM avec équilibrage de charge ou la configuration de réseaux virtuels avancés.

2 Mise en place de la haute disponibilité et l'auto-scaling dans Azure

2.1 Introduction

Dans ce chapitre, nous allons explorer deux concepts fondamentaux pour garantir la fiabilité et l'évolutivité des applications cloud dans Microsoft Azure : les groupes à haute disponibilité et les ensembles de mise à l'échelle de machines virtuelles (VMSS). Ces technologies sont essentielles pour assurer que nos applications restent disponibles même en cas de défaillance matérielle et qu'elles puissent s'adapter automatiquement à la demande.

2.2 Exercice 2 : Création d'un groupe de haute disponibilité

2.2.1 Objectif

L'objectif de cet exercice est de créer un groupe à haute disponibilité dans Azure et d'y déployer trois machines virtuelles de différentes configurations, puis d'analyser les domaines d'erreur et de mise à jour attribués à chaque machine virtuelle.

2.2.2 Étapes de réalisation

Création du groupe à haute disponibilité

Pour commencer, nous allons créer un groupe à haute disponibilité dans la région France Central :

1. Dans le portail Azure, nous nous sommes rendus dans la section "Groupes à haute disponibilité"
2. Nous avons cliqué sur "Créer" pour lancer l'assistant de création
3. Nous avons configuré les paramètres suivants :
   — Abonnement : notre abonnement Azure
   — Groupe de ressources : un nouveau groupe appelé "TP-HA-RG"
   — Nom : "TP-AvailabilitySet"
   — Région : France Central
   — Domaines d'erreur : 2 (valeur par défaut)
   — Domaines de mise à jour : 5 (valeur par défaut)
4. Nous avons finalisé la création en cliquant sur "Vérifier + créer" puis "Créer"

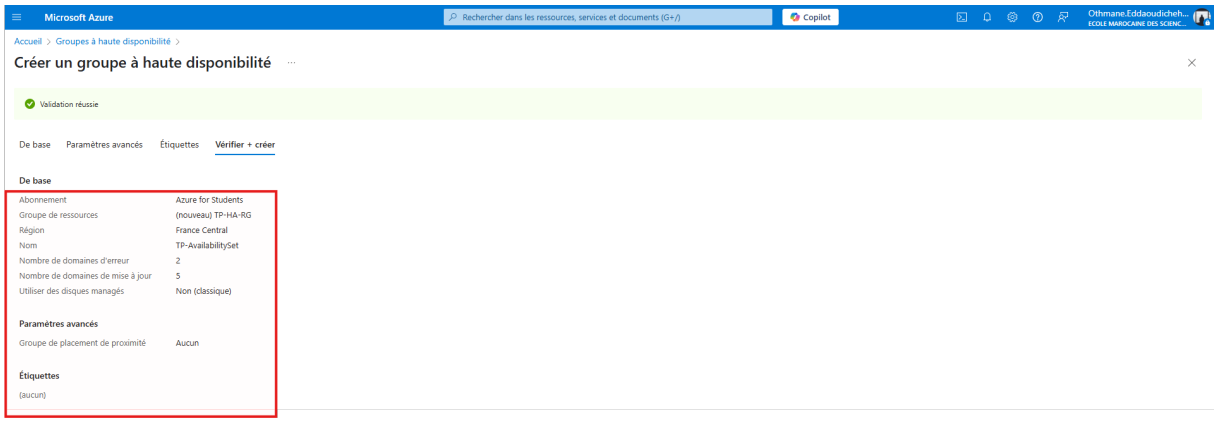

FIGURE 6 – Création du groupe à haute disponibilité dans le portail Azure

Création des machines virtuelles dans le groupe à haute disponibilité

Après la création du groupe à haute disponibilité, nous avons procédé à la création de trois machines virtuelles avec différentes configurations :

VM Windows basée sur l'image capturée

1. Dans le portail Azure, nous avons accédé à la section "Machines virtuelles"
2. Nous avons cliqué sur "Créer" puis sélectionné "Machine virtuelle"
3. Nous avons configuré les paramètres de base :
   — Abonnement : notre abonnement Azure

— Groupe de ressources : "TP-HA-RG" (le même que pour le groupe à haute disponibilité)
— Nom de la machine virtuelle : "win-vm-1"
— Région : France Central
— Options de disponibilité : Groupe à haute disponibilité
— Groupe à haute disponibilité : "TP-AvailabilitySet" (celui que nous venons de créer)
— Image : Nous avons sélectionné l'image Windows Server 2016 Datacenter capturée lors de l'exercice 1
— Taille : Standard D2s v3 (2 processeurs virtuels, 8 Go de RAM)

4. Nous avons configuré les informations d'identification et les ports d'entrée
5. Nous avons passé les étapes suivantes avec les valeurs par défaut
6. Nous avons validé la création en cliquant sur "Vérifier + créer" puis "Créer"

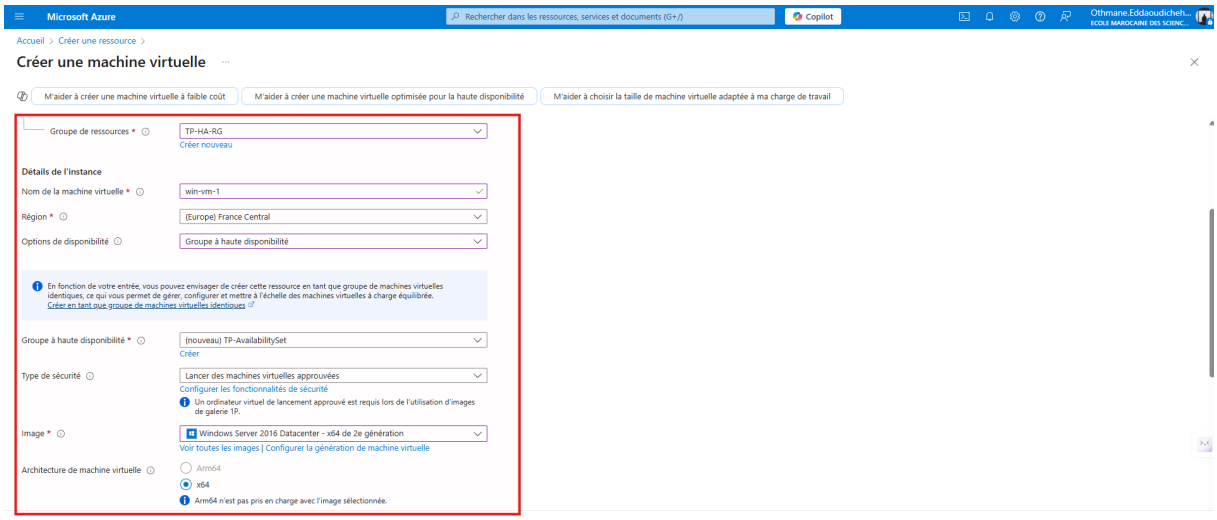

FIGURE 7 – Création de la VM Windows avec sélection du groupe à haute disponibilité

VM Linux 1

1. Nous avons répété le processus pour créer une VM Linux
2. Paramètres principaux :
   — Nom : "linux-vm-1"
   — Image : Ubuntu Server 20.04 LTS
   — Taille : Standard B2s (2 processeurs virtuels, 4 Go de RAM)
   — Groupe à haute disponibilité : "TP-AvailabilitySet"

VM Linux 2

1. Nous avons créé une deuxième VM Linux avec une configuration différente
2. Paramètres principaux :
   — Nom : "win-vm-2"
   — Image : image exercice 1
   — Taille : Standard B2s (2 processeurs virtuels, 4 Go de RAM)
   — Groupe à haute disponibilité : "TP-AvailabilitySet"

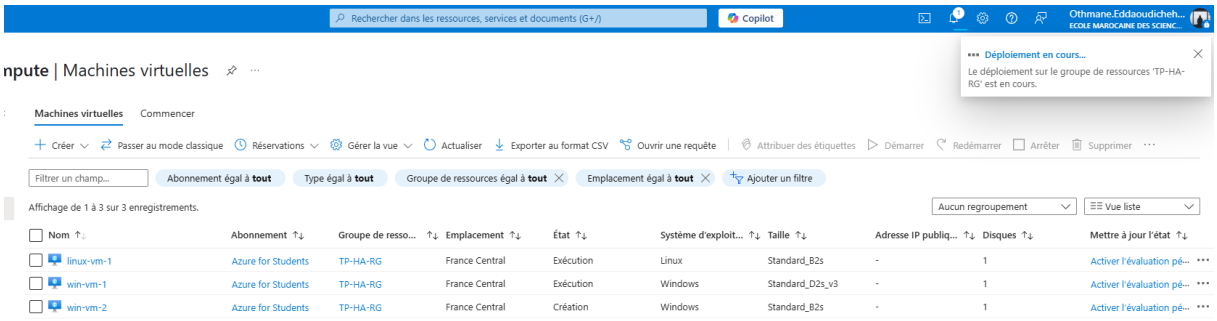

FIGURE 8 – Liste des machines virtuelles créées dans le groupe à haute disponibilité

Analyse des domaines d'erreur et de mise à jour

Après la création des trois machines virtuelles dans le groupe à haute disponibilité, nous avons examiné les domaines d'erreur et de mise à jour attribués à chacune d'elles. Pour cela, nous avons accédé à chaque machine virtuelle et consulté la section "Disponibilité + mise à l'échelle".

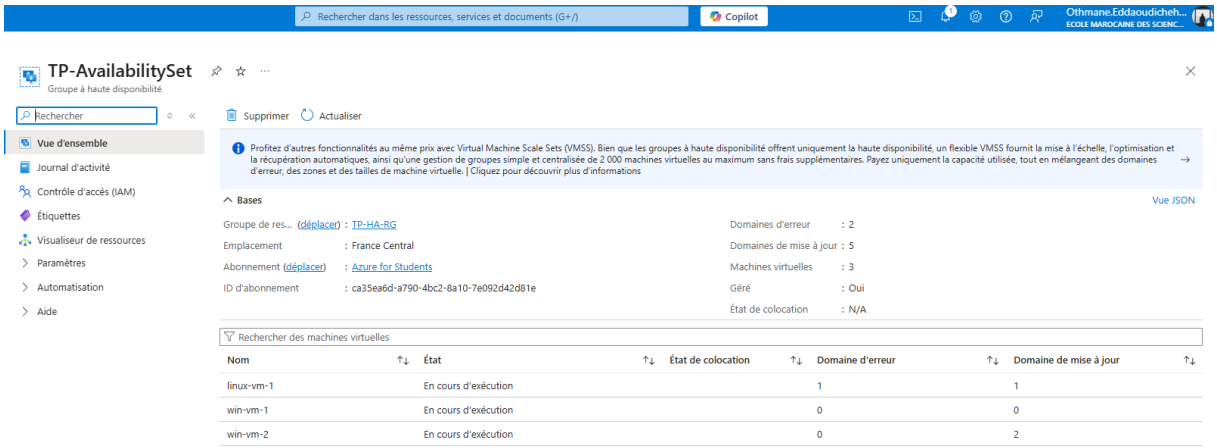

FIGURE 9 – Domaines d'erreur et de mise à jour pour la VM Windows

Machine virtuelle	Domaine d'erreur	Domaine de mise à jour
win-vm-1	0	0
linux-vm-1	1	1
win-vm-2	0	2

TABLE 1 – Récapitulatif des domaines d'erreur et de mise à jour attribués

Explication des domaines

— **Domaine d'erreur** : Représente un groupe isolé de ressources matérielles qui partagent une source d'alimentation et un commutateur réseau communs. Dans notre configuration, nous avons 2 domaines d'erreur, ce qui signifie que nos machines virtuelles sont réparties sur deux racks physiques différents. Nous observons que "win-vm-1" et "linux-vm-2" sont dans le domaine d'erreur 0, tandis que "linux-vm-1" est dans le domaine d'erreur 1. Cette répartition garantit qu'en cas de défaillance

matérielle d'un rack, toutes nos machines virtuelles ne seront pas affectées simultanément.

— **Domaine de mise à jour** : Représente un groupe de machines virtuelles qui peuvent être redémarrées simultanément lors des opérations de maintenance planifiée. Dans notre configuration, nous avons 5 domaines de mise à jour, et nos machines virtuelles sont réparties sur trois d'entre eux (0, 1 et 2). Cette répartition assure que lors des mises à jour de la plateforme Azure, toutes nos machines virtuelles ne seront pas redémarrées en même temps, maintenant ainsi une disponibilité continue de notre service.

La répartition automatique des machines virtuelles dans différents domaines d'erreur et de mise à jour illustre parfaitement la façon dont Azure implémente la haute disponibilité. Cette configuration nous protège contre deux types de défaillances :

1. **Défaillances matérielles non planifiées** (via les domaines d'erreur)
2. **Interruptions planifiées pour maintenance** (via les domaines de mise à jour)

Suppression des ressources

Une fois notre analyse terminée, nous avons procédé à la suppression des ressources pour éviter des coûts inutiles :

1. Nous avons sélectionné le groupe de ressources "TP-HA-RG"
2. Nous avons cliqué sur "Supprimer le groupe de ressources"
3. Nous avons confirmé la suppression en saisissant le nom du groupe de ressources
4. Nous avons attendu la confirmation de la suppression de toutes les ressources associées

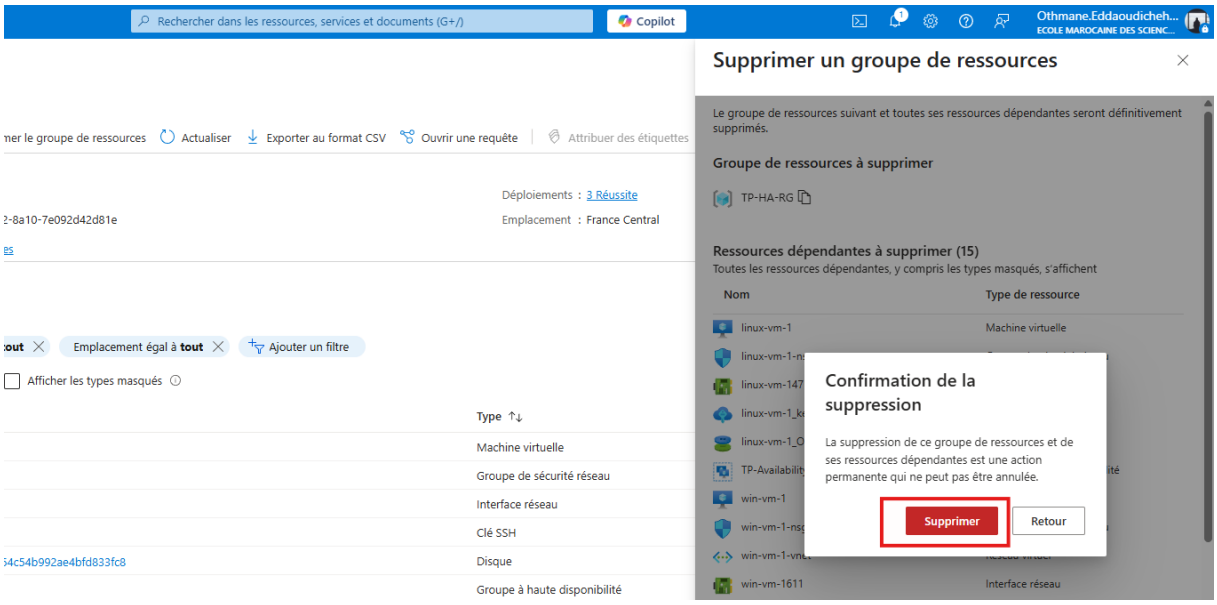

FIGURE 10 – Suppression du groupe de ressources et de toutes les ressources associées

2.3 Exercice 3 : Création d'un ensemble de mise à l'échelle de machines virtuelles (VMSS)

2.3.1 Objectif

L'objectif de cet exercice est de créer un ensemble de mise à l'échelle de machines virtuelles (VMSS) comprenant initialement 3 VM, et de configurer l'auto-scaling pour adapter automatiquement la capacité en fonction de la charge.

2.3.2 Étapes de réalisation

Création du VMSS

1. Dans le portail Azure, nous nous sommes rendus dans la section "Ensembles de mise à l'échelle de machines virtuelles"
2. Nous avons cliqué sur "Créer" pour lancer l'assistant de création
3. Nous avons configuré les paramètres de base :
   — Abonnement : notre abonnement Azure
   — Groupe de ressources : "TP-VMSS-RG" (nouveau groupe de ressources)
   — Nom du groupe identique de machines virtuelles : "tp-vmss"
   — Région : France Central
   — Zone de disponibilité : Zone 1 (pour répartir les instances dans des centres de données physiquement séparés)
   — Image : Ubuntu Server 20.04 LTS
   — Taille : Standard DS1_v2
   — Type d'authentification : mot de passe
   — Nom d'utilisateur et mot de passe : configurés selon les besoins du TP
4. Dans la section "Mise à l'échelle", nous avons configuré :
   — Nombre d'instances initial : 3
   — Stratégie de mise à l'échelle : Flexible (pour permettre l'auto-scaling)
   — Nombre minimal d'instances : 2
   — Nombre maximal d'instances : 5
   — Nombre souhaité d'instances : 3

Configuration de l'auto-scaling
Une fois le VMSS créé, nous avons configuré les règles d'auto-scaling pour ajuster automatiquement le nombre d'instances en fonction de la charge CPU :

1. Dans le VMSS, nous avons accédé à la section "Mise à l'échelle"
2. Nous avons sélectionné "Mise à l'échelle automatique personnalisée"
3. Nous avons créé deux règles de mise à l'échelle :

Règle d'augmentation de capacité
— Nom : "Scale-Out-Rule"
— Métrique : pourcentage CPU moyen
— Opérateur : supérieur à
— Seuil : 70%
— Durée : 10 minutes (période d'agrégation)
— Action : augmenter le nombre d'instances de 1
— Délai d'attente : 5 minutes

Règle de diminution de capacité
— Nom : "Scale-In-Rule"
— Métrique : pourcentage CPU moyen
— Opérateur : inférieur à
— Seuil : 30%
— Durée : 10 minutes (période d'agrégation)
— Action : diminuer le nombre d'instances de 1
— Délai d'attente : 5 minutes

Vérification du fonctionnement
Après la création du VMSS avec auto-scaling, nous avons vérifié que les trois instances initiales étaient bien créées et opérationnelles :

1. Nous avons accédé à la section "Instances" du VMSS
2. Nous avons confirmé que 3 instances étaient en cours d'exécution
3. Nous avons vérifié l'état de chaque instance et leur répartition dans la zone de disponibilité

Simulation de charge pour tester l'auto-scaling
Pour vérifier le fonctionnement de l'auto-scaling, nous avons simulé une charge élevée sur les instances du VMSS :

1. Nous nous sommes connectés à l'une des instances via SSH
2. Nous avons exécuté un script pour générer une charge CPU élevée :

```
while true; do openssl speed; done
```

3. Nous avons exécuté ce script sur les trois instances pour générer une charge supérieure à 70% du CPU
4. Nous avons observé dans le portail Azure que le VMSS a détecté l'augmentation de la charge CPU
5. Après environ 10 minutes (période d'agrégation), nous avons constaté que le VMSS avait ajouté une instance supplémentaire, passant de 3 à 4 instances

Suppression des ressources
Une fois notre test d'auto-scaling terminé, nous avons procédé à la suppression des ressources :

1. Nous avons sélectionné le groupe de ressources "TP-VMSS-RG"
2. Nous avons cliqué sur "Supprimer le groupe de ressources"
3. Nous avons confirmé la suppression en saisissant le nom du groupe de ressources
4. Nous avons attendu la confirmation de la suppression de toutes les ressources associées

2.4 Conclusion

Groupe à haute disponibilité
Le groupe à haute disponibilité nous a permis de déployer nos machines virtuelles avec une protection contre deux types de défaillances :

— Les **défaillances matérielles** grâce à la répartition sur différents domaines d'erreur
— Les **interruptions de maintenance planifiée** grâce à la répartition sur différents domaines de mise à jour

Cette configuration offre une disponibilité accrue pour nos applications sans nécessiter de modifications au niveau de l'application elle-même. Toutefois, cette approche présente certaines limitations :

— Elle ne permet pas d'adapter automatiquement la capacité en fonction de la charge
— Elle nécessite de gérer individuellement chaque machine virtuelle
— Le nombre maximal de machines virtuelles est limité à 200 par groupe à haute disponibilité

Ensemble de mise à l'échelle de machines virtuelles (VMSS)

Le VMSS nous a offert les avantages suivants :

— **Élasticité automatique** : ajustement du nombre d'instances en fonction de la charge CPU
— **Haute disponibilité** : répartition des instances sur différentes zones de disponibilité
— **Gestion simplifiée** : création et configuration d'un groupe d'instances identiques plutôt que de VM individuelles
— **Économies potentielles** : réduction automatique du nombre d'instances pendant les périodes de faible charge

Notre test a démontré que le VMSS répond efficacement aux variations de charge en ajoutant automatiquement des instances lorsque la charge CPU dépasse 70% pendant plus de 10 minutes.

Comparaison et cas d'utilisation

Caractéristique	**Groupe à haute disponibilité**	**VMSS**
Objectif principal	Protection contre les défaillances	Élasticité
Configuration des VM	Peut être hétérogène	Homogène
Auto-scaling	Non	Oui
Gestion	Individuelle	Groupe
Cas d'utilisation	Applications stables avec charge prévisible	Applications avec charge variable et imprévisible

Table 2 – Comparaison entre groupe à haute disponibilité et VMSS

En conclusion, le choix entre un groupe à haute disponibilité et un VMSS dépend des besoins spécifiques de l'application :

— Utilisez un **groupe à haute disponibilité** lorsque vous avez besoin de déployer des machines virtuelles avec des configurations différentes et que la charge est relativement stable et prévisible.
— Utilisez un **VMSS** lorsque vous avez besoin de déployer de nombreuses machines virtuelles identiques et que vous souhaitez adapter automatiquement la capacité en fonction de la charge.

Ces deux exercices nous ont permis de comprendre et de mettre en pratique les concepts fondamentaux de la haute disponibilité et de l'élasticité dans Azure, deux piliers essentiels pour la conception d'applications cloud résilientes et performantes.

3 Réseaux Virtuels Azure

3.1 Introduction

Ce chapitre présente les travaux pratiques réalisés sur la plateforme Microsoft Azure concernant la création et la configuration des réseaux virtuels (VNets). Ces exercices pratiques permettent de mettre en application les concepts théoriques abordés dans le cours et de développer des compétences essentielles pour l'administration cloud.

Les objectifs principaux de ces travaux pratiques sont :

— Créer et configurer des réseaux virtuels avec des sous-réseaux
— Mettre en place des groupes de sécurité réseau (NSG)
— Configurer la communication entre machines virtuelles au sein d'un même réseau
— Établir une communication entre machines virtuelles de différents réseaux
— Configurer une connexion VPN point-à-site entre une machine locale et un réseau virtuel Azure

3.2 Exercice 1 : Communication entre VMs du même réseau virtuel

Cet exercice vise à établir une communication entre des machines virtuelles situées dans le même réseau virtuel mais dans des sous-réseaux différents.

3.2.1 Création du réseau virtuel et des sous-réseaux

La première étape consiste à créer un réseau virtuel avec deux sous-réseaux distincts :

1. J'ai créé un réseau virtuel nommé `BahaSamia-vNet1` dans le groupe de ressources `vnet-RG` et dans la région `North Europe`.
2. J'ai défini l'espace d'adressage principal comme `192.168.0.0/16` après avoir supprimé l'espace par défaut.
3. J'ai créé deux sous-réseaux :
   — `Direction` : `192.168.1.0/29` permettant d'avoir 6 adresses IP disponibles (8 adresses au total, moins les 2 adresses réservées par Azure)
   — `RH` : `192.168.2.0/28` permettant d'avoir 14 adresses IP disponibles (16 adresses au total, moins les 2 adresses réservées par Azure)

L'interface de création du réseau virtuel est présentée dans la figure 11.

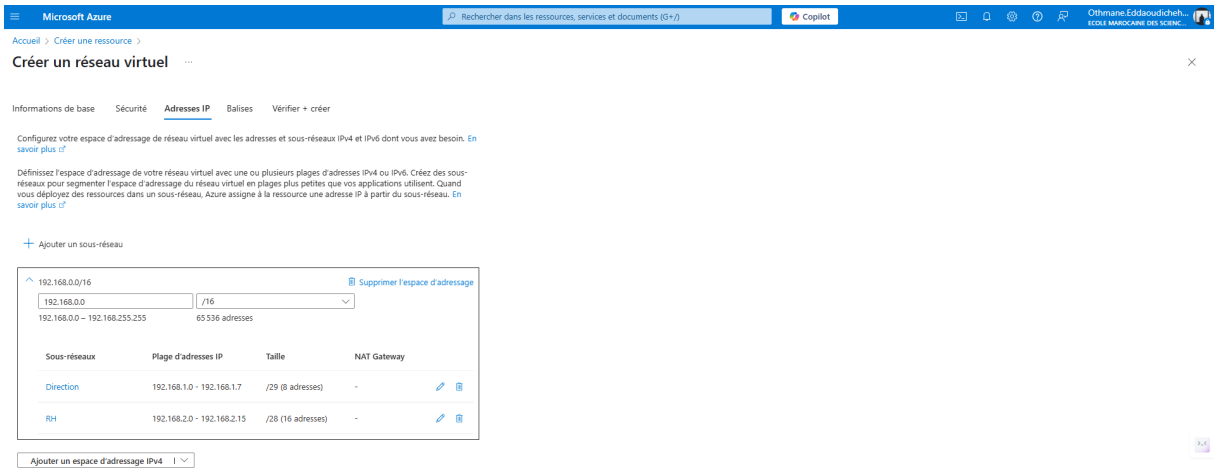

FIGURE 11 – Création du réseau virtuel BahaSamia-vNet1 avec deux sous-réseaux

3.2.2 Création et configuration du groupe de sécurité réseau

Pour sécuriser notre réseau, j'ai créé et configuré un groupe de sécurité réseau (NSG) :

1. J'ai créé un NSG nommé `BahaSamia-NSG1` dans le groupe de ressources `vnet-RG` et dans la région `North Europe`.
2. J'ai ajouté une règle de sécurité de trafic entrant autorisant le port 22 (SSH) pour n'importe quelle source et destination.
3. J'ai associé ce NSG aux deux sous-réseaux `Direction` et `RH`.

La figure 12 montre la configuration des règles du NSG.

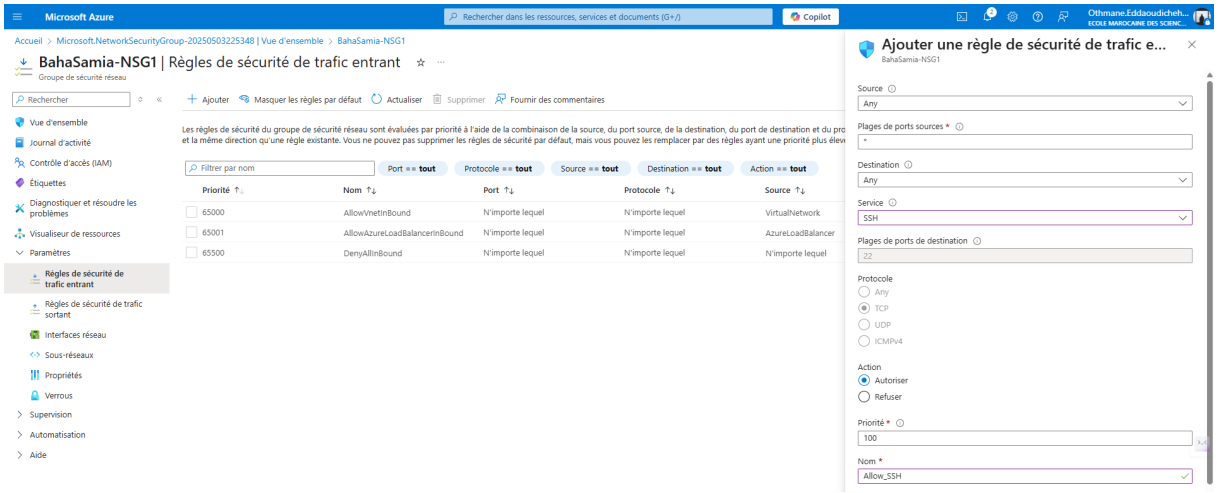

FIGURE 12 – Configuration des règles de sécurité pour BahaSamia-NSG1

La figure 13 illustre l'association du NSG aux sous-réseaux.

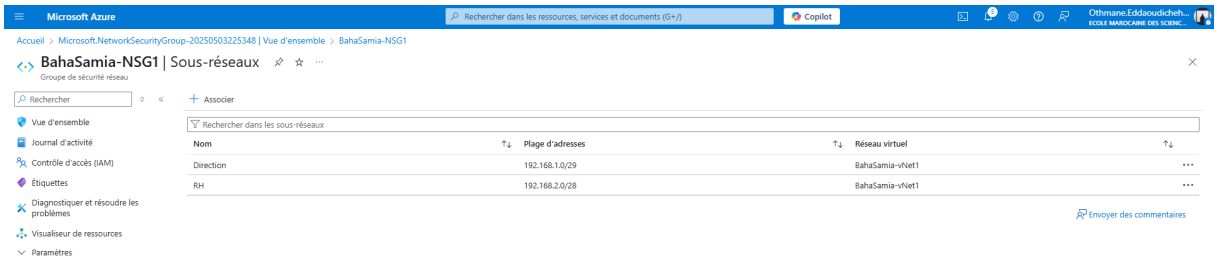

FIGURE 13 – Association du NSG aux sous-réseaux Direction et RH

3.2.3 Création des machines virtuelles

J'ai ensuite créé deux machines virtuelles Linux, chacune dans un sous-réseau différent :

1. **VM1** dans le sous-réseau `Direction` :
   — Région : `North Europe` (même région que le réseau virtuel)
   — Adresse IP : `192.168.1.4` (première adresse disponible après les adresses réservées)
   — Port SSH non autorisé directement dans la configuration de la VM
2. **VM2** dans le sous-réseau `RH` :
   — Région : `North Europe` (même région que le réseau virtuel)
   — Adresse IP : `192.168.2.4` (première adresse disponible après les adresses réservées)
   — Port SSH non autorisé directement dans la configuration de la VM

La figure 14 montre la création d'une des machines virtuelles avec la configuration réseau.

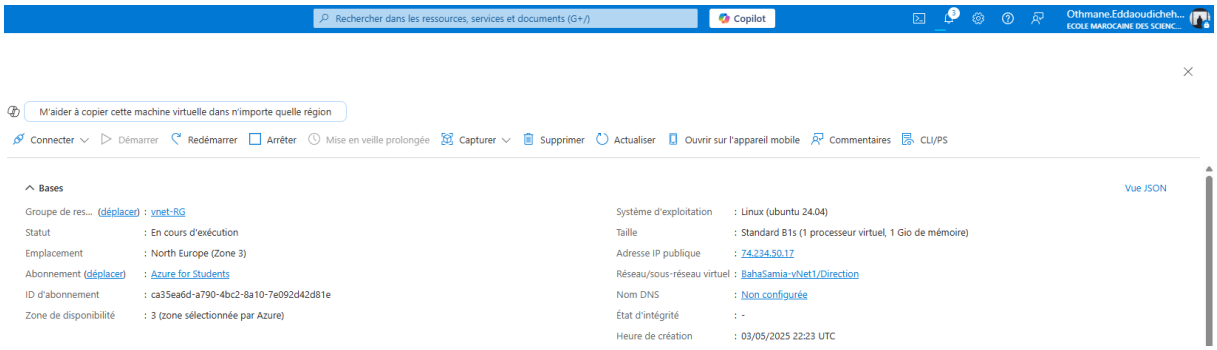

FIGURE 14 – Création de VM1 dans le sous-réseau Direction

3.2.4 Test de connectivité

Après avoir créé les ressources, j'ai procédé aux tests de connectivité :

1. **Connexion SSH** : J'ai pu me connecter aux deux machines virtuelles via SSH malgré le fait que le port SSH n'ait pas été autorisé directement dans la configuration des VMs. Cela est dû au fait que la règle du NSG autorise le trafic SSH pour tout le réseau.

2. **Test de ping** : J'ai testé la communication entre VM1 et VM2 en utilisant la commande ping depuis VM1 :

```
ping 192.168.2.5
```

Le test a réussi, confirmant que les machines du même réseau virtuel peuvent communiquer entre elles, même si elles se trouvent dans des sous-réseaux différents.

La figure 15 montre le résultat du test de ping entre les deux VMs.

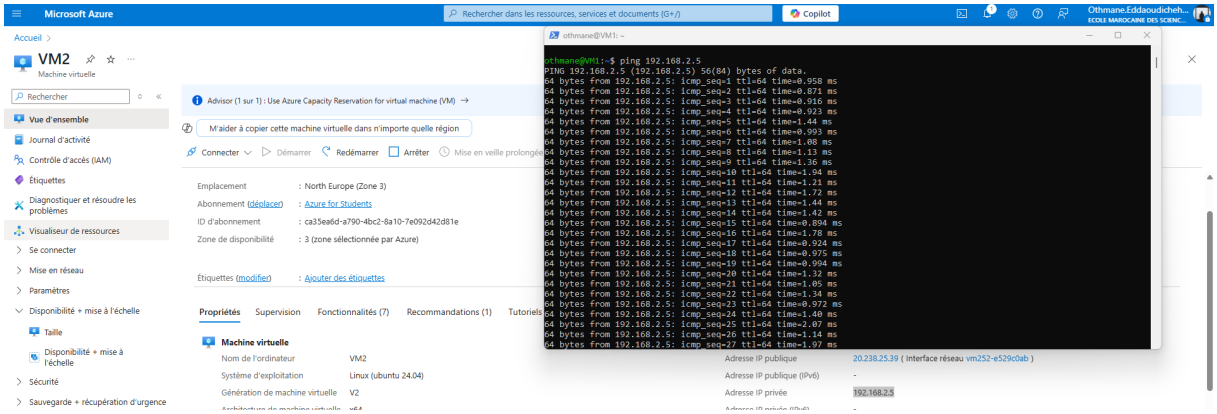

FIGURE 15 – Test de ping entre VM1 et VM2 du même réseau virtuel

Le diagramme du réseau obtenu est présenté dans la figure 16.

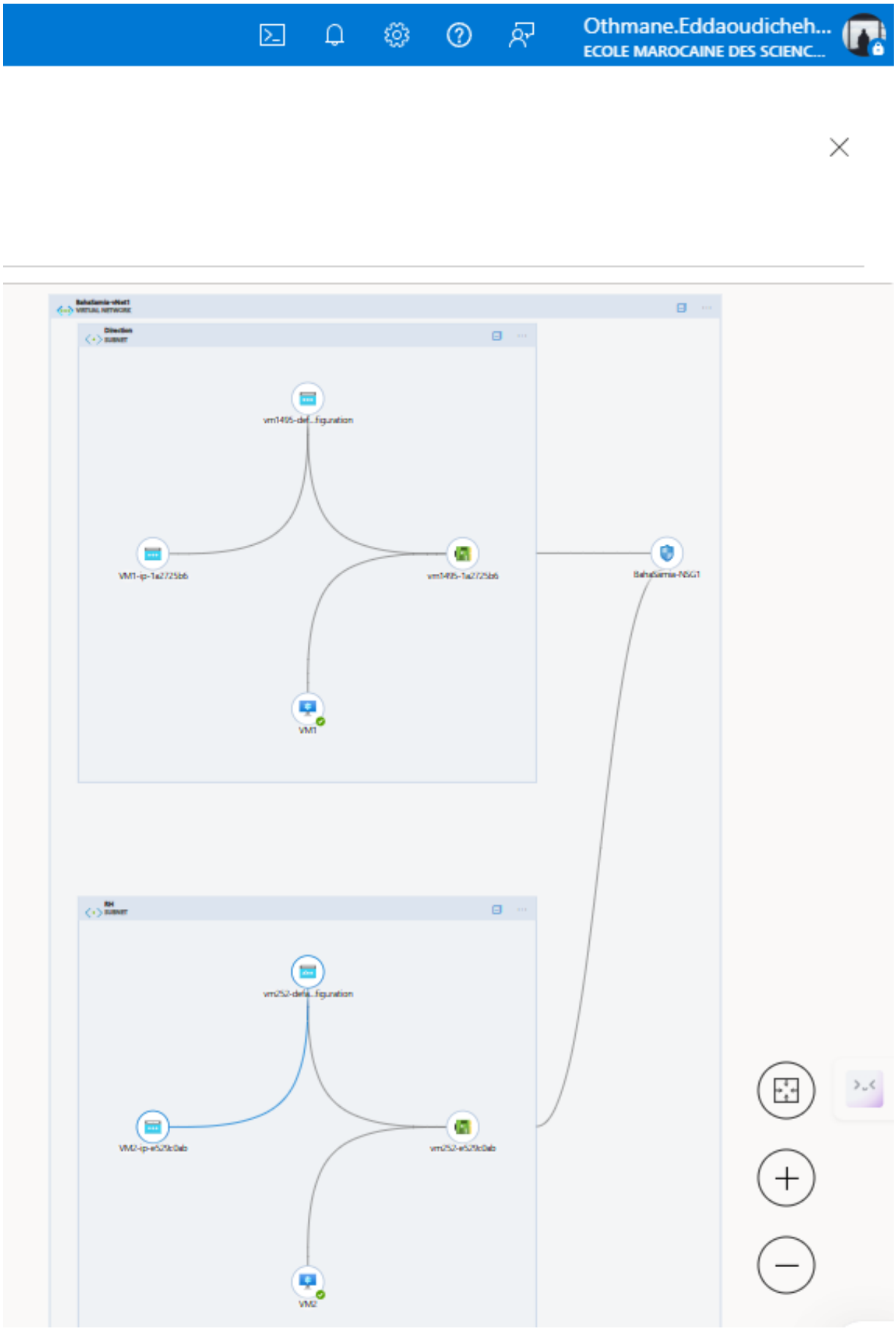

FIGURE 16 – Diagramme du réseau pour l'exercice 1

3.3 Exercice 2 : Communication entre VMs de différents réseaux virtuels

Cet exercice vise à établir une communication entre des machines virtuelles situées dans des réseaux virtuels différents en utilisant le peering de réseaux virtuels.

3.3.1 Création du second réseau virtuel

J'ai créé un second réseau virtuel avec un sous-réseau :

1. J'ai créé un réseau virtuel nommé `BahaSamia-vNet2` dans le groupe de ressources `vnet-RG` et dans la région `France Central`.
2. J'ai défini l'espace d'adressage principal comme `172.16.0.0/16` après avoir supprimé l'espace par défaut.
3. J'ai créé un sous-réseau `Personnel` avec l'espace d'adressage `172.16.1.0/24`, permettant d'avoir 254 adresses IP disponibles (256 adresses au total, moins les 2 adresses réservées par Azure).

La figure 17 montre la création du second réseau virtuel.

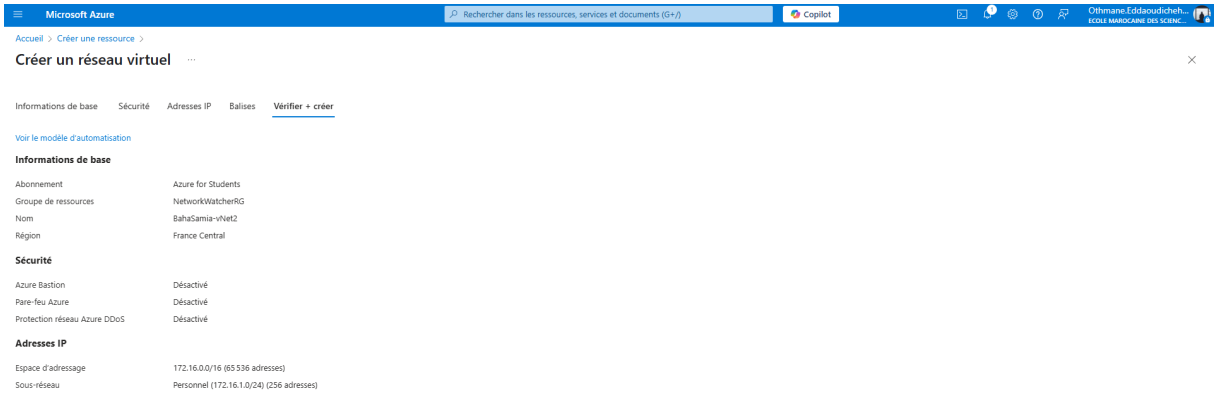

FIGURE 17 – Création du réseau virtuel BahaSamia-vNet2 avec un sous-réseau

3.3.2 Création de la troisième machine virtuelle

J'ai créé une troisième machine virtuelle dans le nouveau réseau :

1. **VM3** dans le sous-réseau `Personnel` :
   — Région : `France Central` (même région que le réseau virtuel)
   — Adresse IP : `172.16.1.4` (première adresse disponible après les adresses réservées)
   — Port SSH autorisé directement dans la configuration de la VM

La figure 18 montre la création de VM3.

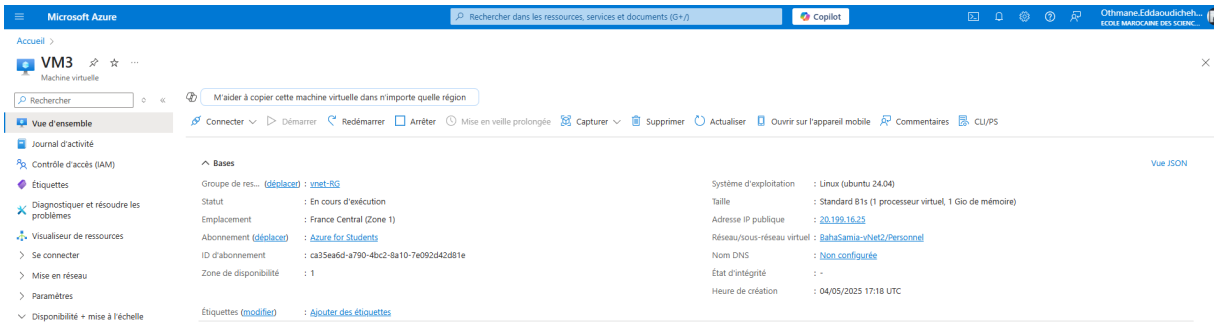

FIGURE 18 – Création de VM3 dans le sous-réseau Personnel

3.3.3 Test de connectivité initial

J'ai testé la communication entre VM1 (dans vNet1) et VM3 (dans vNet2) en utilisant ping :

```
ping 172.16.1.4
```

Ce test a échoué car, par défaut, les réseaux virtuels dans Azure sont isolés les uns des autres. La figure 19 montre l'échec du test de ping.

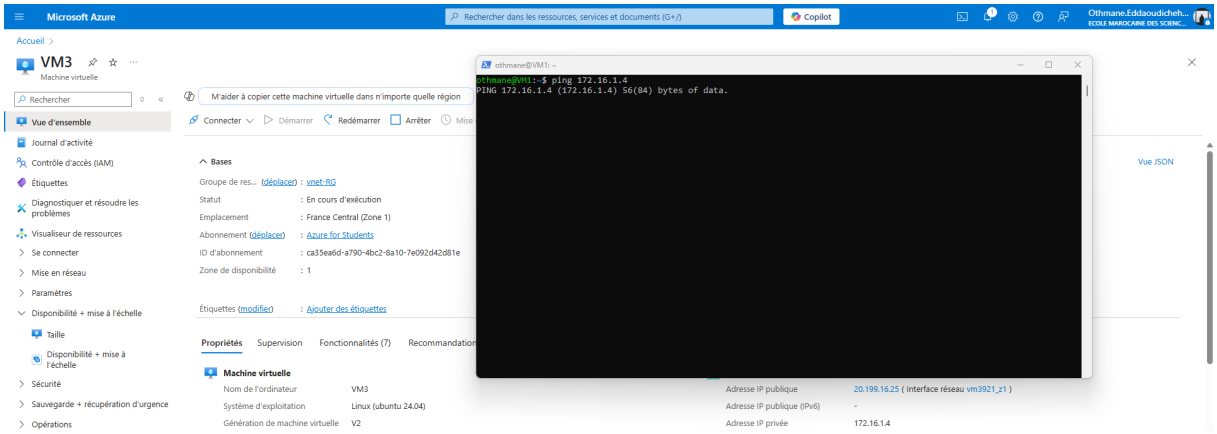

FIGURE 19 – Échec du test de ping entre VM1 et VM3 avant le peering

3.3.4 Configuration du peering de réseaux virtuels

Pour permettre la communication entre les deux réseaux virtuels, j'ai configuré un peering :

1. J'ai accédé au réseau virtuel `BahaSamia-vNet1`.
2. Dans les paramètres, j'ai sélectionné `Peerings` puis `Ajouter`.
3. J'ai configuré le peering dans les deux sens :
   — Nom du lien de peering de `vNet1` vers `vNet2` : `vNet1-to-vNet2`
   — Nom du lien de peering de `vNet2` vers `vNet1` : `vNet2-to-vNet1`
   — J'ai sélectionné `BahaSamia-vNet2` comme réseau virtuel distant.

La figure 20 illustre la configuration du peering.

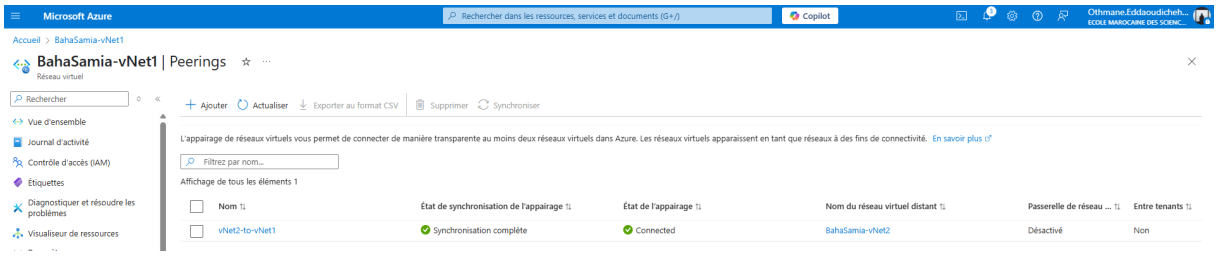

FIGURE 20 – Configuration du peering entre les deux réseaux virtuels

3.3.5 Test de connectivité après peering

Après avoir configuré le peering, j'ai testé à nouveau la communication entre VM1 et VM3 :

```
ping 172.16.1.4
```

Cette fois, le test a réussi, confirmant que le peering permet aux machines virtuelles de différents réseaux virtuels de communiquer entre elles. La figure 21 montre le succès du test de ping.

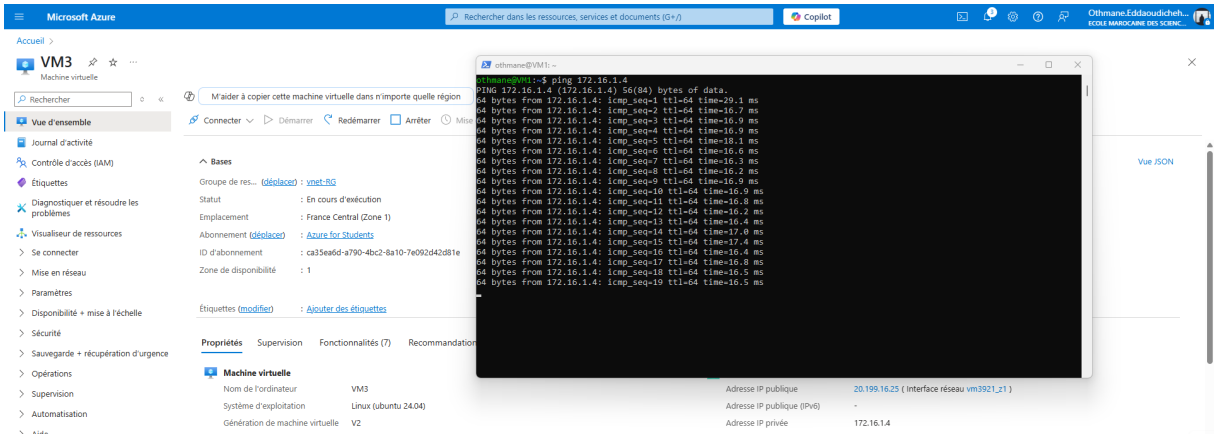

FIGURE 21 – Test de ping réussi entre VM1 et VM3 après la configuration du peering

Le diagramme du réseau obtenu pour l'exercice 2 est présenté dans la figure 22.

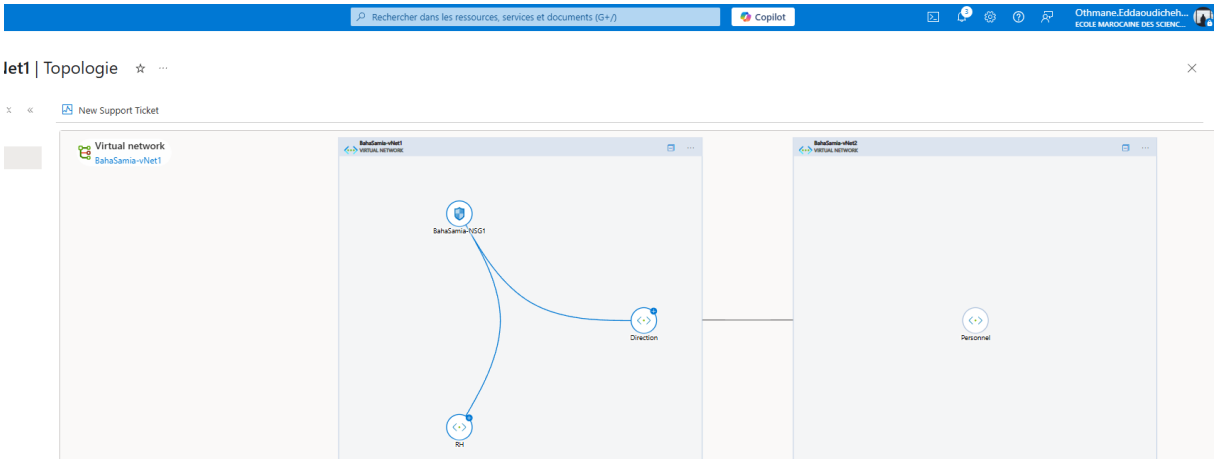

FIGURE 22 – Diagramme du réseau pour l'exercice 2

3.3.6 Nettoyage des ressources

Après avoir terminé les exercices 1 et 2, j'ai supprimé toutes les ressources associées pour éviter des frais inutiles :

- Les machines virtuelles VM1, VM2 et VM3
- Les réseaux virtuels BahaSamia-vNet1 et BahaSamia-vNet2
- Le groupe de sécurité réseau BahaSamia-NSG1
- Toutes les ressources associées (interfaces réseau, disques, etc.)

3.4 Exercice 3 : Communication entre VM Azure et machine locale

Cet exercice vise à établir une connexion VPN point-à-site entre un réseau virtuel Azure et une machine locale.

3.4.1 Création du réseau virtuel et du sous-réseau

J'ai créé un nouveau réseau virtuel avec un sous-réseau :

1. J'ai créé un réseau virtuel nommé `BahaSamia-vNet1` dans le groupe de ressources `vnet-RG` et dans la région `France Centrale`.
2. J'ai défini l'espace d'adressage principal comme `172.16.0.0/16` après avoir supprimé l'espace par défaut.
3. J'ai créé un sous-réseau `Personnel` avec l'espace d'adressage `172.16.1.0/24`.

La figure 23 montre la création du réseau virtuel pour l'exercice 3.

3.4.2 Création de la passerelle de réseau virtuel

J'ai créé une passerelle de réseau virtuel pour permettre la connexion VPN :

1. J'ai créé une passerelle nommée `BahaSamia-VPN-gateway1` dans le groupe de ressources `vnet-RG` et dans la région `France Centrale`.

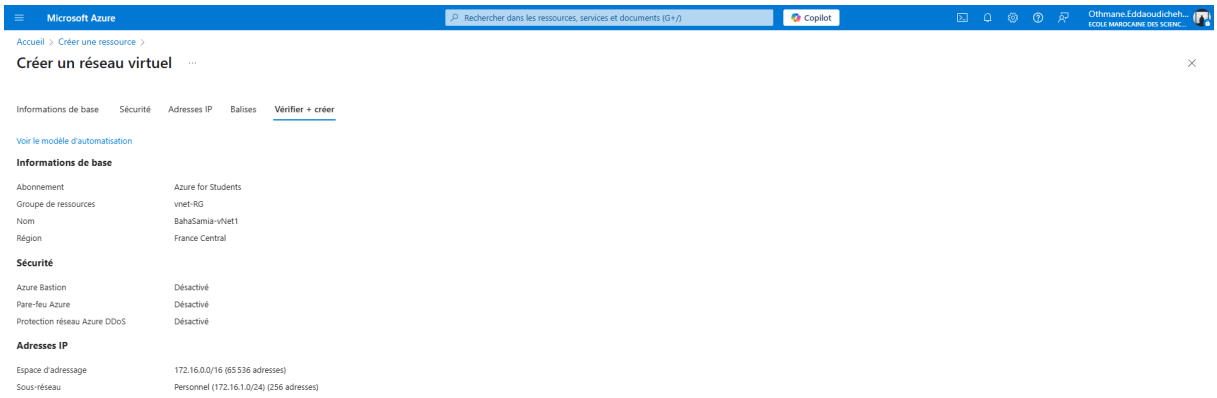

FIGURE 23 – Création du réseau virtuel pour l'exercice 3

2. J'ai configuré les paramètres comme suit :
   - Type de passerelle : `VPN`
   - Type de VPN : `Basé sur itinéraires`
   - SKU : `De base`
   - Génération : `Generation1`
   - Réseau virtuel : `BahaSamia-vNet1`
   - Plage d'adresses de sous-réseau de la passerelle : `172.16.2.0/24`
   - Mode actif/passif : `Désactivé`
   - Adresse IP publique : `VPN-ip1` (nouvelle adresse créée)

La figure 24 illustre la création de la passerelle VPN.

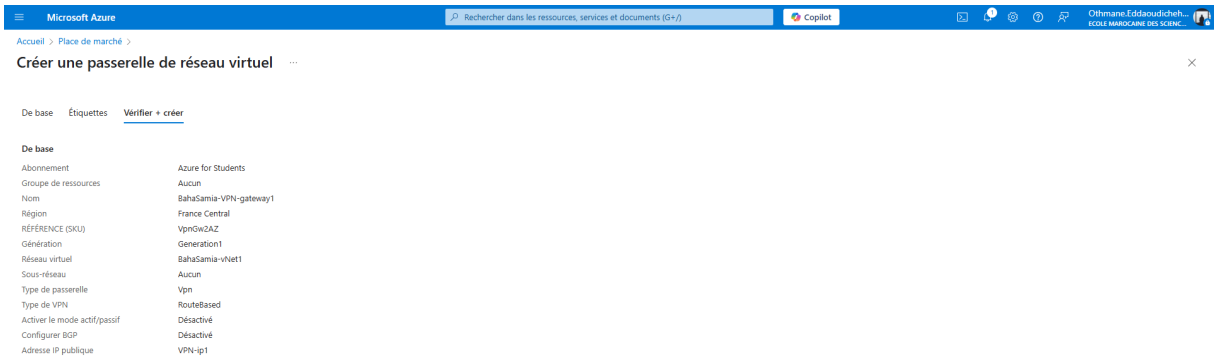

FIGURE 24 – Création de la passerelle de réseau virtuel

3.4.3 Génération et exportation des certificats

Pour configurer la connexion VPN point-à-site, j'ai généré et exporté les certificats nécessaires :

1. J'ai généré le certificat racine auto-signé (self-signed root certificate) à l'aide de PowerShell :

```
$cert = New-SelfSignedCertificate -Type Custom -KeySpec Signature ‘
-Subject "CN=P2SRootCert" -KeyExportPolicy Exportable ‘
-HashAlgorithm sha256 -KeyLength 2048 ‘
-CertStoreLocation "Cert:\CurrentUser\My" -KeyUsageProperty Sign -
KeyUsage CertSign
```

2. J'ai généré le certificat client à l'aide de PowerShell :

```
New-SelfSignedCertificate -Type Custom -DnsName P2SChildCert -KeySpec Signat
-Subject "CN=P2SChildCert" -KeyExportPolicy Exportable ‘
-HashAlgorithm sha256 -KeyLength 2048 ‘
-CertStoreLocation "Cert:\CurrentUser\My" ‘
-Signer $cert -TextExtension @("2.5.29.37={text}1.3.6.1.5.5.7.3.2")
```

3. J'ai exporté le certificat racine sans la clé privée au format base64 (`.cer`).
4. J'ai exporté le certificat client avec la clé privée au format PFX (`.pfx`) en définissant un mot de passe.

La figure 25 montre l'exportation des certificats.

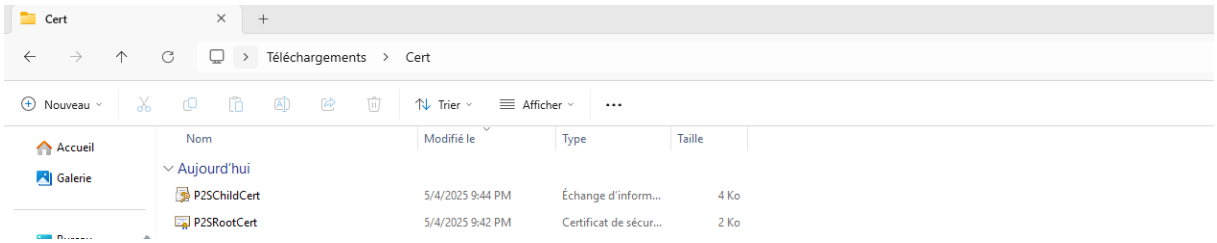

FIGURE 25 – Exportation des certificats pour la connexion VPN

3.4.4 Configuration point-à-site de la passerelle VPN

J'ai configuré les paramètres point-à-site de la passerelle VPN :

1. Dans la passerelle créée, j'ai accédé à la section `Configuration point-à-site`.
2. J'ai défini le pool d'adresses pour les clients VPN : `192.168.1.0/24`.
3. J'ai ajouté le certificat racine :
   — Nom : `BahaSamia-Root`
   — Données du certificat public : contenu du certificat racine au format base64

La figure 26 illustre la configuration point-à-site.

3.4.5 Installation et configuration du client VPN

J'ai téléchargé et installé le client VPN sur ma machine locale :

1. J'ai téléchargé le client VPN directement depuis le portail Azure, dans la section `Télécharger le client VPN` de la passerelle.
2. J'ai installé le client VPN sur ma machine locale.
3. Avant de démarrer la connexion VPN, j'ai vérifié l'adresse IP privée de ma machine locale avec la commande `ipconfig` :

```
Adresse IPv4 . . . . . . . . . . . . . .: 192.168.0.15
```

4. J'ai démarré le client VPN et établi la connexion avec le réseau virtuel `BahaSamia-vNet1`.

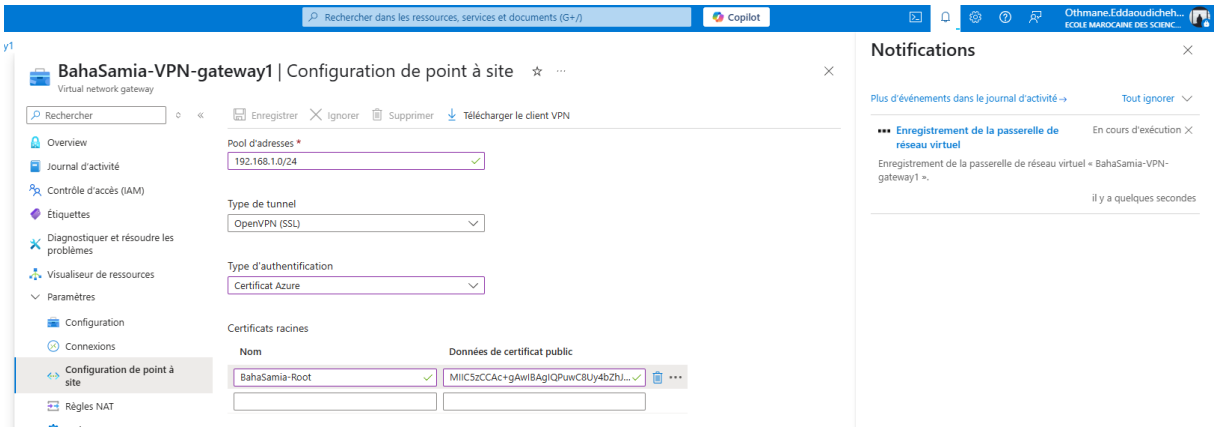

FIGURE 26 – Configuration point-à-site de la passerelle VPN

5. Après la connexion, j'ai vérifié à nouveau l'adresse IP avec `ipconfig` et constaté qu'une nouvelle interface réseau avait été créée avec une adresse IP issue du pool configuré :

```
Adresse IPv4 . . . . . . . . . . . . . .: 192.168.1.4
```

La figure **??** montre le client VPN en fonctionnement.

3.4.6 Test de connectivité

J'ai testé la connectivité entre ma machine locale et la machine virtuelle dans le réseau Azure :

1. J'ai créé une VM dans le sous-réseau `Personnel` avec l'adresse IP `172.16.1.4`.
2. Depuis ma machine locale connectée au VPN, j'ai exécuté un ping vers cette VM :

```
ping 172.16.1.4
```

3. Le test a réussi, confirmant que la connexion VPN point-à-site fonctionne correctement.

La figure **??** montre le test de ping réussi depuis la machine locale.
Le diagramme du réseau obtenu pour l'exercice 3 est présenté dans la figure **??**.

3.4.7 Nettoyage des ressources

Après avoir terminé l'exercice 3, j'ai effectué les opérations de nettoyage suivantes :

1. J'ai déconnecté le client VPN.
2. J'ai désinstallé le client VPN de ma machine locale.
3. J'ai supprimé toutes les ressources Azure associées à l'exercice 3 :
   — La machine virtuelle
   — La passerelle de réseau virtuel
   — Le réseau virtuel et son sous-réseau
   — L'adresse IP publique

3.5 Conclusion

Ces travaux pratiques m'ont permis d'acquérir une expérience précieuse dans la configuration des réseaux virtuels Azure et dans l'établissement de différents types de connectivité réseau. J'ai appris à :

- Créer et configurer des réseaux virtuels avec des sous-réseaux adaptés aux besoins
- Mettre en place des groupes de sécurité réseau pour contrôler le trafic
- Établir une communication entre machines virtuelles dans le même réseau
- Configurer un peering pour permettre la communication entre différents réseaux virtuels
- Mettre en place une connexion VPN point-à-site entre Azure et une machine locale

Ces compétences sont essentielles pour concevoir et déployer des infrastructures cloud sécurisées et évolutives sur Microsoft Azure. La maîtrise des réseaux virtuels est fondamentale pour tout administrateur cloud, car elle permet d'isoler les ressources, de contrôler les flux de communication et d'intégrer les environnements cloud avec les infrastructures existantes.

4 Création et gestion du stockage Azure

4.1 Introduction

Ce TP a pour objectif de mettre en pratique les concepts du stockage Azure vus dans le cours. Nous allons créer et configurer différents types de stockage Azure (Blob, File, Queue et Table), et explorer leurs fonctionnalités. Ce rapport présente les étapes suivies et les résultats obtenus lors de la réalisation des exercices demandés.

4.2 Exercice 1 : Création d'un compte de stockage

4.2.1 Création du compte

Pour commencer, j'ai créé un compte de stockage avec les paramètres suivants :

- **Groupe de ressources** : storage-RG
- **Nom** : J'ai d'abord essayé d'utiliser "TEST" comme nom, mais une erreur est apparue indiquant que le nom du compte de stockage doit être unique au niveau mondial dans Azure. J'ai donc utilisé "othmanechehbouni" (mon nom et prénom).
- **Région** : West Europe
- **Performances** : Standard. Les autres types de performances disponibles sont Premium pour les objets blob de blocs, Premium pour les objets blob de pages et Premium pour les partages de fichiers.
- **Redondance** : Stockage localement redondant (LRS). Le SLA pour LRS est de 99,9% de disponibilité. Les autres types de redondance disponibles sont :
  - ZRS (Zone-Redundant Storage) : données répliquées dans trois zones de disponibilité
  - GRS (Geo-Redundant Storage) : données répliquées dans une région secondaire
  - GZRS (Geo-Zone-Redundant Storage) : combinaison de ZRS et GRS

  Lorsqu'on choisit GRS ou GZRS, un message apparaît pour autoriser l'accès en lecture à la région secondaire (RA-GRS ou RA-GZRS). Cette option permet d'accéder aux données dans la région secondaire en cas d'indisponibilité de la région primaire.

— **Data Lake Storage Gen2** : J'ai décoché cette option. Elle sert à activer des fonctionnalités hiérarchiques pour l'analytique de Big Data sur le stockage Blob.
— **Niveau d'accès stockage blob** : Chaud (Hot)

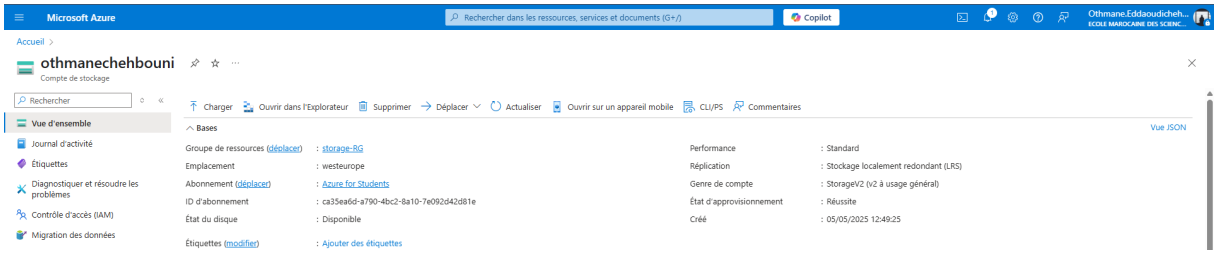

FIGURE 27 – Création du compte de stockage

4.2.2 Ajout de Blob Storage

Pour ajouter du Blob Storage, j'ai dû choisir l'option "Containers" car c'est là que les objets blob sont stockés.

— J'ai ajouté un container nommé "cours" avec un niveau d'accès "Privé".
— J'ai chargé les 3 premiers chapitres Azure en format PDF dans ce container.
— L'URL pour accéder à chapitres est : `https://othmanechehbouni.blob.core.windows.net/cours/`
— Je ne peux pas y accéder directement car le niveau d'accès est défini comme "Privé", ce qui signifie que l'accès n'est possible qu'avec une autorisation explicite (comme une signature d'accès partagé).
— Le type du blob "chapitre1" est "Block blob", ce qui est approprié pour les fichiers PDF.
— J'ai supprimé le chapitre 3, puis vérifié s'il était possible de le restaurer. Effectivement, la fonctionnalité "soft delete" était activée avec une période de rétention de 7 jours par défaut.
— J'ai restauré le chapitre 3 en utilisant l'option de récupération.

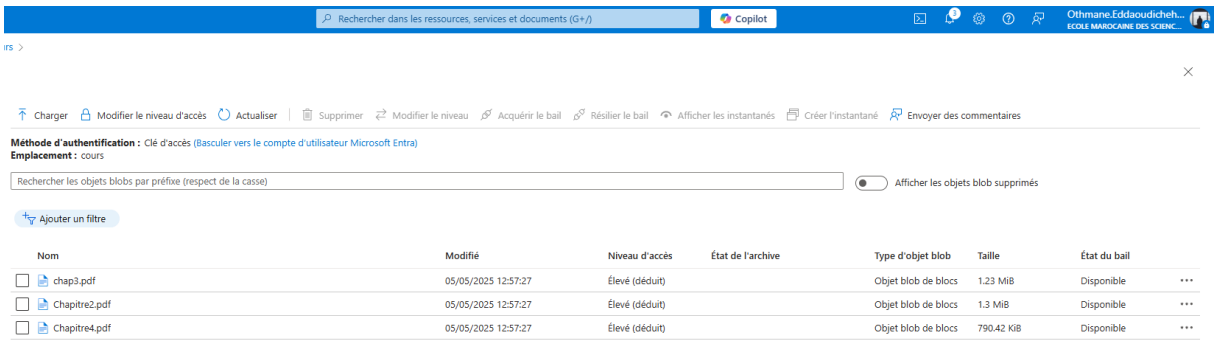

FIGURE 28 – Configuration du container Blob et upload des fichiers

J'ai également créé un deuxième container nommé "text-files" avec un niveau d'accès "Container" :

— J'ai chargé un fichier texte et l'ai modifié directement à partir du portail Azure.
— L'URL pour accéder au fichier est : `https://othmanechehbouni.blob.core.windows.net/text-files/notes.txt`

— Je peux y accéder directement car le niveau d'accès est défini comme "Container", ce qui permet un accès en lecture anonyme aux blobs dans le container.
— J'ai constaté qu'il est possible de créer une structure de dossiers dans ce type de stockage, mais ces "dossiers" sont en réalité virtuels et font partie du nom de l'objet blob.

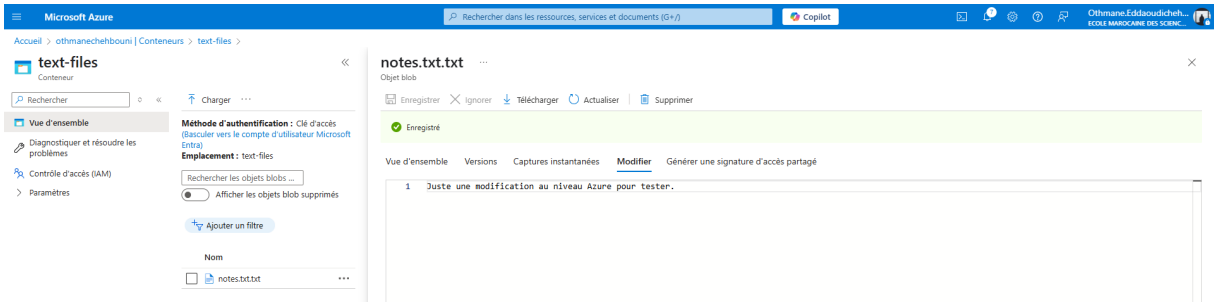

FIGURE 29 – Container avec niveau d'accès Container et modification de fichier texte

4.2.3 Ajout de File Storage

Pour ajouter du File Storage, j'ai dû choisir l'option "File Shares" :
— J'ai créé un partage de fichiers nommé "fileshare1" avec un niveau "Chaud".
— La capacité maximale (Quota) par défaut était de 5120 GiB (5 TiB).
— J'ai modifié le quota de "fileshare1" à 5 GiB.
— À partir du portail Azure, j'ai connecté le file share avec le protocole SMB à ma machine locale.
— Dans le disque monté, j'ai déplacé les 3 premiers chapitres Azure.
— J'ai vérifié que les données avaient été correctement transférées sur Azure en actualisant l'affichage dans le portail.
— J'ai créé quelques dossiers dans le partage de fichiers.
— L'URL pour accéder à chapitre1 est : `https://othmanechehbouni.file.core.windows.net/fileshare1/chapitre1.pdf`
— Finalement, j'ai déconnecté le file share de ma machine locale.

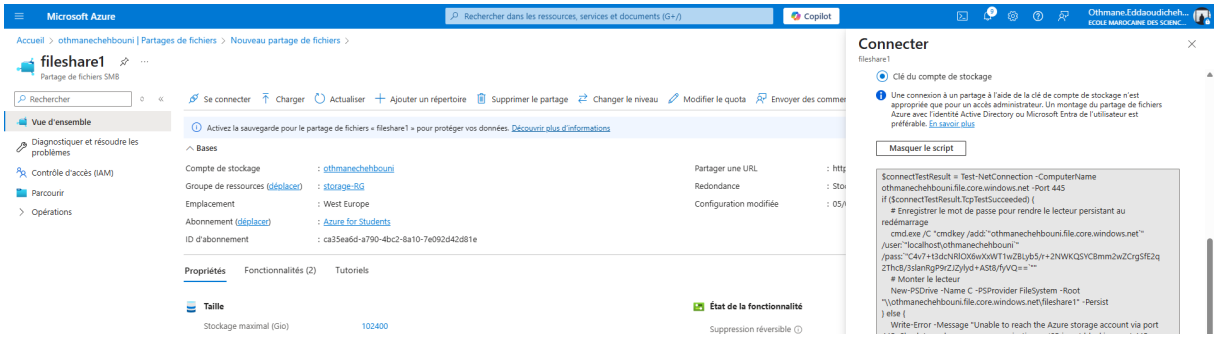

FIGURE 30 – Création et connexion du File Share

4.2.4 Ajout de Table Storage

Pour ajouter du Table Storage (pour les données semi-structurées), j'ai choisi l'option "Tables" :

— J'ai créé une table nommée "table1".
— À partir du Navigateur de stockage, j'ai ajouté les entités suivantes :
  — Une entité : Ahmed Ahmedi
  — Une entité : Amine Amini avec la propriété age = 24
  — Une entité : Saida Saidi avec la propriété age = 22
— J'ai effectué une requête avec un filtre pour sélectionner l'entité avec age=24 OU PartitionKey='Ahmed'. Le résultat a bien affiché ces deux entités.

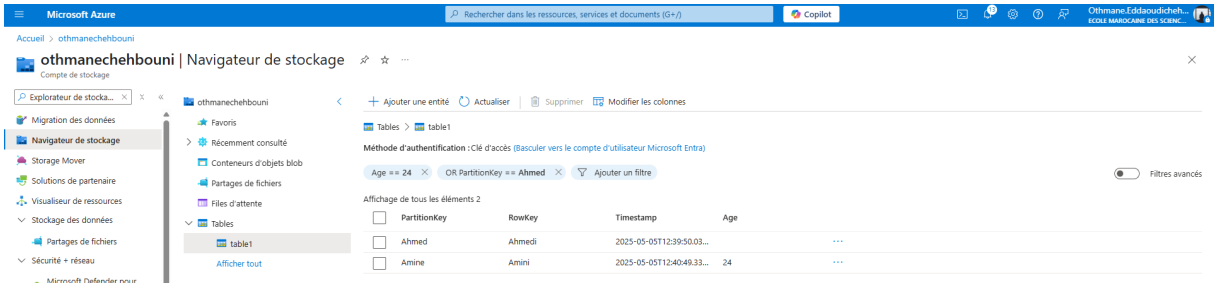

FIGURE 31 – Création et gestion de Table Storage

4.2.5 Ajout de Queue Storage

Pour le Queue Storage, j'ai procédé comme suit :
— J'ai créé une file d'attente nommée "Queue1".
— J'ai ajouté 2 messages à cette file d'attente.

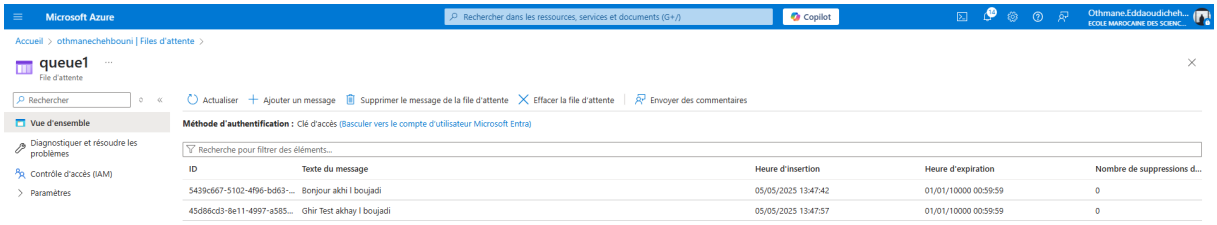

FIGURE 32 – Création et ajout de messages dans Queue Storage

4.3 Vue d'ensemble avec Storage Explorer

J'ai utilisé Storage Explorer pour avoir une vue d'ensemble du contenu de mon compte de stockage :

4.4 Question sur la haute disponibilité

Pour créer un blob storage hautement disponible, le type de redondance de stockage à choisir serait ZRS (Zone-Redundant Storage) ou GZRS (Geo-Zone-Redundant Storage). Je ne peux pas modifier le type de redondance dans mon compte de stockage existant "abdelazizbenali" car le type de redondance est défini lors de la création du compte et ne peut être modifié ultérieurement pour certaines transitions (par exemple, de LRS vers ZRS). Il faudrait créer un nouveau compte de stockage avec le type de redondance souhaité.

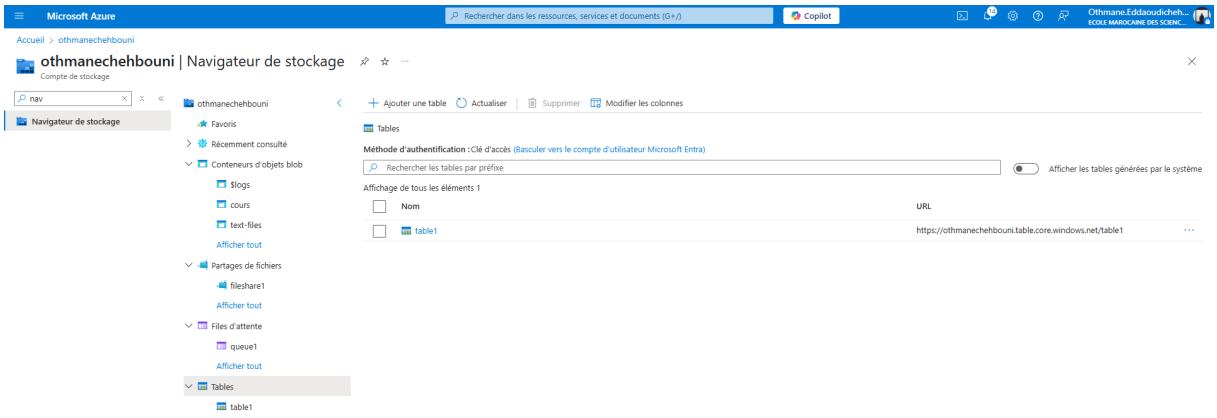

FIGURE 33 – Vue d'ensemble du compte de stockage avec Storage Explorer

4.5 Libération des ressources

Pour finir, j'ai libéré les ressources en supprimant le groupe de ressources "storage-RG", ce qui a automatiquement supprimé le compte de stockage et tous ses services associés.

4.6 Conclusion

Ce TP m'a permis de mettre en pratique les différents concepts de stockage Azure vus en cours. J'ai pu créer et configurer les quatre types de stockage (Blob, File, Queue et Table) et explorer leurs fonctionnalités spécifiques. J'ai également compris les différentes options de redondance disponibles et leur impact sur la disponibilité des données. Le stockage Azure offre une grande flexibilité avec ses différents services, chacun adapté à des besoins spécifiques : le Blob Storage pour les données non structurées, le File Storage pour les partages de fichiers compatibles SMB, le Queue Storage pour la communication asynchrone entre applications, et le Table Storage pour les données semi-structurées.

5 Azure App Service

5.1 Introduction

Ce chapitre présente les travaux pratiques réalisés pour la compréhension et la maîtrise d'Azure App Service. À travers plusieurs exercices, nous explorons la création d'applications web, leur déploiement, la mise à l'échelle et la supervision des ressources Azure.

5.2 Exercice 1 : Création et gestion des applications web Azure

5.2.1 Création d'une première application web

Pour débuter nos travaux pratiques, nous avons créé une première application web dans Azure App Service avec les caractéristiques suivantes :

- **Groupe de ressources** : webApp-RG
- **Nom** : BoualiKarim (nom unique pour identifier l'application)
- **Région** : East US
- **Pile d'exécution** : Java 8
- **Serveur** : Apache Tomcat 9.0

— **Système d'exploitation** : Linux
— **Plan tarifaire** : B1 (nommé plan-B1)

Pour créer cette application, nous avons suivi les étapes suivantes dans le portail Azure :

1. Dans le menu du portail Azure, nous avons sélectionné *Créer une ressource*
2. Dans la barre de recherche, nous avons tapé "Web App" et sélectionné *Web App*
3. Dans l'onglet *Informations de base*, nous avons saisi les informations requises comme indiqué ci-dessus
4. Nous avons créé un nouveau plan de service App nommé *plan-B1* avec le niveau tarifaire B1

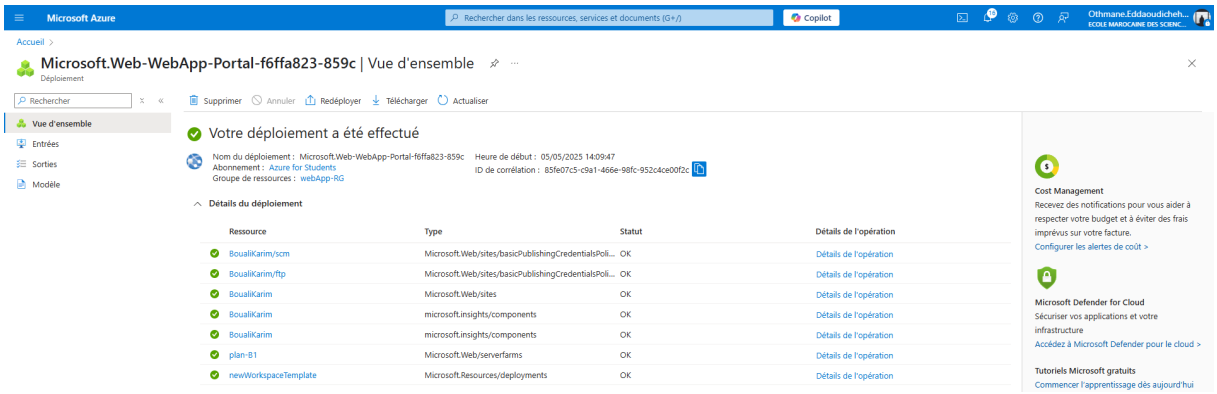

FIGURE 34 – Création de la première application web avec le plan B1

Concernant les caractéristiques du plan B1 et son SLA (Service Level Agreement), nous avons observé :

— **Configuration matérielle** : 1 cur CPU, 1,75 Go de RAM, 10 Go de stockage
— **Fonctionnalités** : Domaines personnalisés, certificats SSL, mise à l'échelle manuelle
— **SLA** : 99,95% de disponibilité (les niveaux Basic, Standard et Premium bénéficient d'un SLA, contrairement aux niveaux Free et Shared)

Il est important de noter que le choix du système d'exploitation affecte effectivement les plans et leurs caractéristiques. Certaines fonctionnalités peuvent varier entre les systèmes Windows et Linux, notamment la prise en charge de certaines piles d'exécution spécifiques.

5.2.2 Création d'une deuxième application web

Nous avons ensuite créé une deuxième application web en utilisant le même plan App Service :

— **Groupe de ressources** : webApp-RG
— **Nom** : BoualiKarim1
— **Région** : East US
— **Pile d'exécution** : Java 8
— **Serveur** : Apache Tomcat 9.0
— **Système d'exploitation** : Linux
— **Plan** : plan-B1 (réutilisation du plan existant)

Cette étape démontre la capacité d'Azure App Service à héberger plusieurs applications web dans un même plan, optimisant ainsi les coûts.

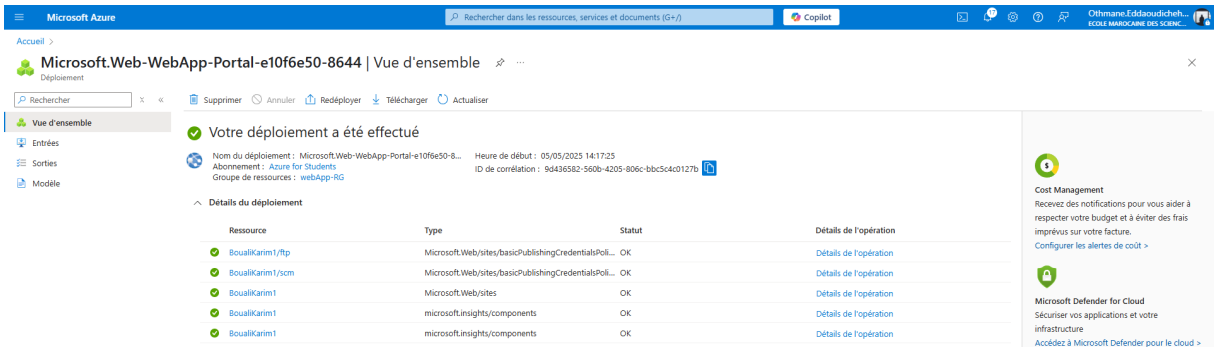

FIGURE 35 – Création de la deuxième application web avec le même plan B1

5.2.3 Création d'une troisième application web et analyse des contraintes

Pour la troisième application web, nous avons testé différentes configurations pour comprendre les limitations :

— **Groupe de ressources** : webApp-RG

— **Nom** : BoualiKarim2

Test de région différente Nous avons d'abord tenté de créer l'application en région *France Central* avec le plan-B1. Cette tentative a échoué car un plan App Service est spécifique à une région. Il n'est pas possible d'héberger des applications dans des régions différentes sur le même plan. Nous avons donc dû revenir à la région *East US*.

Test de pile d'exécution différente Nous avons ensuite essayé d'utiliser Python 3.11 comme pile d'exécution. Bien que le plan B1 prenne en charge différentes piles d'exécution, certaines combinaisons de piles et de systèmes d'exploitation peuvent ne pas être compatibles. Dans notre cas, nous avons pu constater que Python 3.11 est compatible avec notre plan B1 sous Linux.

Test de système d'exploitation différent Enfin, nous avons testé avec Windows comme système d'exploitation. Cette configuration n'a pas fonctionné avec notre plan-B1 existant car les plans App Service sont spécifiques à un système d'exploitation. Un plan créé pour Linux ne peut pas héberger des applications Windows et vice versa. Nous avons donc dû revenir à Linux comme système d'exploitation.

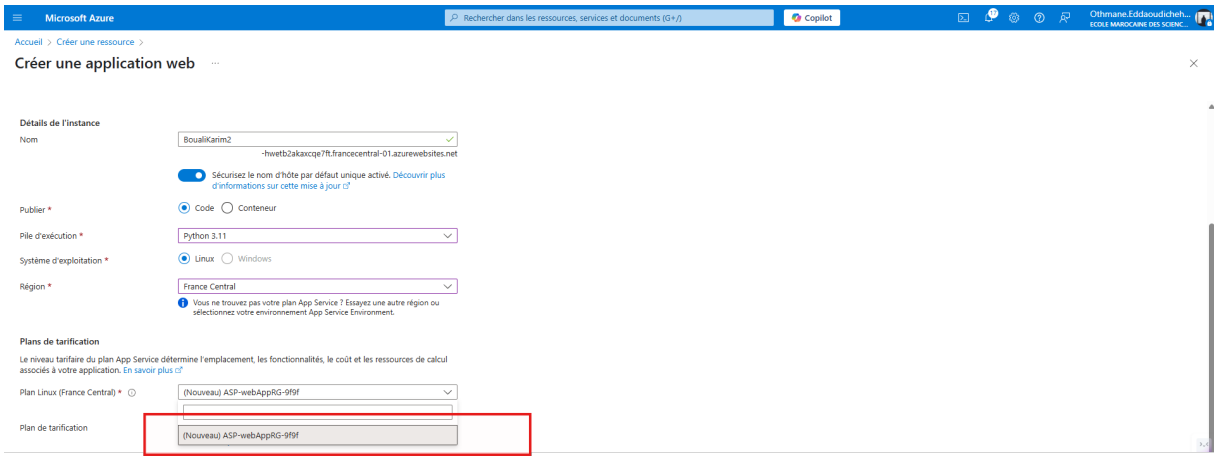

FIGURE 36 – Tentative de création d'une application web avec des paramètres incompatibles (Impossible - manque de plan)

La configuration finale pour notre troisième application était :
— **Région** : East US
— **Pile d'exécution** : Java 8
— **Serveur** : Apache Tomcat 9.0
— **Système d'exploitation** : Linux
— **Plan** : plan-B1

5.3 Gestion des ressources et mise à l'échelle

5.3.1 Scale-out manuel du plan App Service

Après avoir créé nos trois applications web, nous les avons toutes hébergées sur le même plan App Service (plan-B1). Nous avons procédé à l'analyse de la scalabilité :
— **Applications hébergées** : 3 applications web (BoualiKarim, BoualiKarim1, BoualiKarim2)
— **Instances en cours d'exécution** : 1 instance par défaut
Pour effectuer un scale-out manuel, nous avons suivi ces étapes :

1. Accès au *Groupe de ressources webApp-RG*
2. Sélection du *plan-B1* dans la liste des ressources
3. Navigation vers *Mise à l'échelle > Scale-out (nombre d'instances)*
4. Modification du nombre d'instances de 1 à 2
5. Enregistrement des modifications

Le plan B1 permet d'effectuer un scale-out manuel jusqu'à 3 instances maximum. Cette limitation est due au niveau tarifaire choisi. Pour augmenter davantage le nombre d'instances, il faudrait passer à un niveau tarifaire supérieur.

Concernant le scale-out automatique, il n'est pas disponible avec le plan B1. Cette fonctionnalité n'est disponible qu'à partir des niveaux Standard (S1, S2, S3) et supérieurs.

5.3.2 Scale-up du plan App Service

Pour répondre au besoin d'un scale-out automatique pouvant atteindre 10 instances et offrant une redondance interzone, nous avons décidé de faire un scale-up du plan B1.

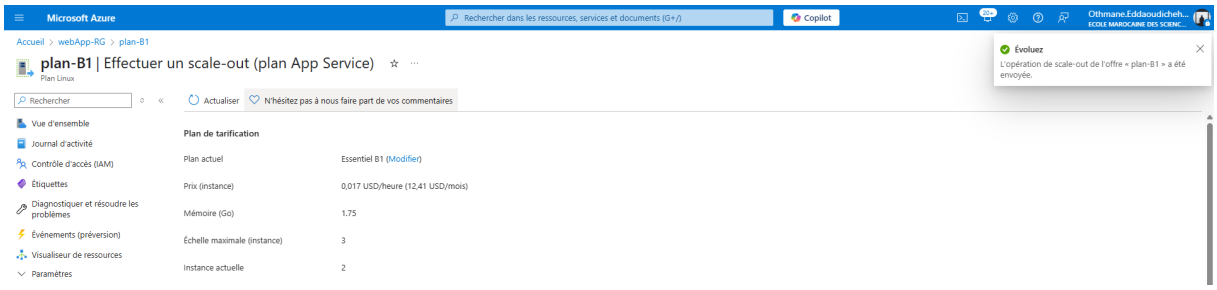

FIGURE 37 – Scale-out manuel du plan B1 de 1 à 2 instances

Après analyse des différentes options, nous avons choisi le plan Standard S1 comme étant le plus approprié en termes de coût-efficacité :

- **Nouveau plan** : Standard S1
- **Configuration matérielle** : 1 cur CPU, 1,75 Go de RAM, 50 Go de stockage
- **Fonctionnalités supplémentaires** : mise à l'échelle automatique jusqu'à 10 instances, environnements de test
- **Redondance interzone** : Disponible avec ce niveau
- **Coût** : Plus élevé que B1 mais offrant un bien meilleur rapport qualité-prix pour nos besoins

Pour effectuer le scale-up, nous avons suivi ces étapes :

1. Dans le plan-B1, navigation vers *Mise à l'échelle > Scale-up (niveau tarifaire)*
2. Sélection du niveau *Standard (S1)*
3. Examen des fonctionnalités additionnelles obtenues
4. Validation du changement

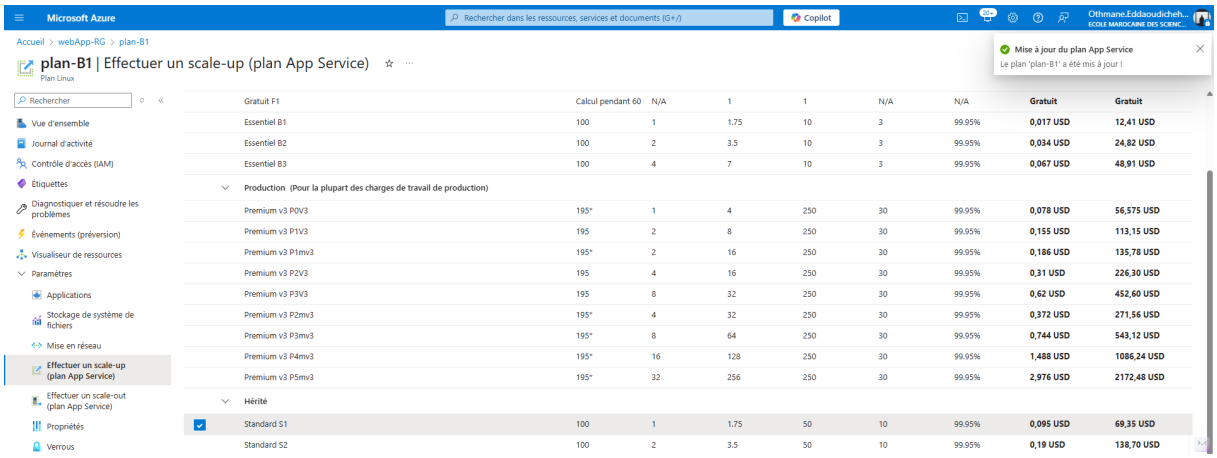

FIGURE 38 – Scale-up du plan B1 vers le plan Standard S1

5.3.3 Surveillance du plan App Service

Pour la surveillance du plan, nous avons configuré deux métriques principales avec différents types de visualisation :

1. **Pourcentage d'utilisation CPU** : Représenté par un graphique en courbe

2. **Pourcentage d'utilisation de la mémoire** : Représenté par un graphique en aire

Procédure suivie :

1. Navigation vers le plan App Service
2. Sélection de *Métriques* dans le menu de gauche
3. Ajout d'un graphique pour le pourcentage de CPU
4. Ajout d'un second graphique pour l'utilisation de la mémoire
5. Configuration du type de graphique approprié pour chacun

FIGURE 39 – Configuration des métriques de surveillance pour le plan App Service

Concernant la possibilité de supprimer le plan App Service, il n'est pas possible de le supprimer directement tant qu'il héberge des applications web. Il faudrait d'abord supprimer toutes les applications web associées ou les déplacer vers un autre plan avant de pouvoir supprimer le plan lui-même.

5.3.4 Scale-out automatique du plan App Service

Après avoir effectué le scale-up vers le plan Standard S1, nous avons configuré le scale-out automatique :

1. Navigation vers *Mise à l'échelle > Scale-out (nombre d'instances)*
2. Sélection du mode *Basé sur une règle*
3. Configuration des paramètres par défaut :
   — Instance minimale : 1
   — Instance maximale : 10
   — Instance par défaut : 2
4. Ajout des règles de déclenchement :
   — **Règle 1 (Scale-out)** : Si le pourcentage CPU > 70% pendant 5 minutes, augmenter le nombre d'instances de 1
   — **Règle 2 (Scale-in)** : Si le pourcentage CPU < 20% pendant 5 minutes, diminuer le nombre d'instances de 1

5. Configuration de la durée de refroidissement (cool down) de 5 minutes entre les opérations de mise à l'échelle

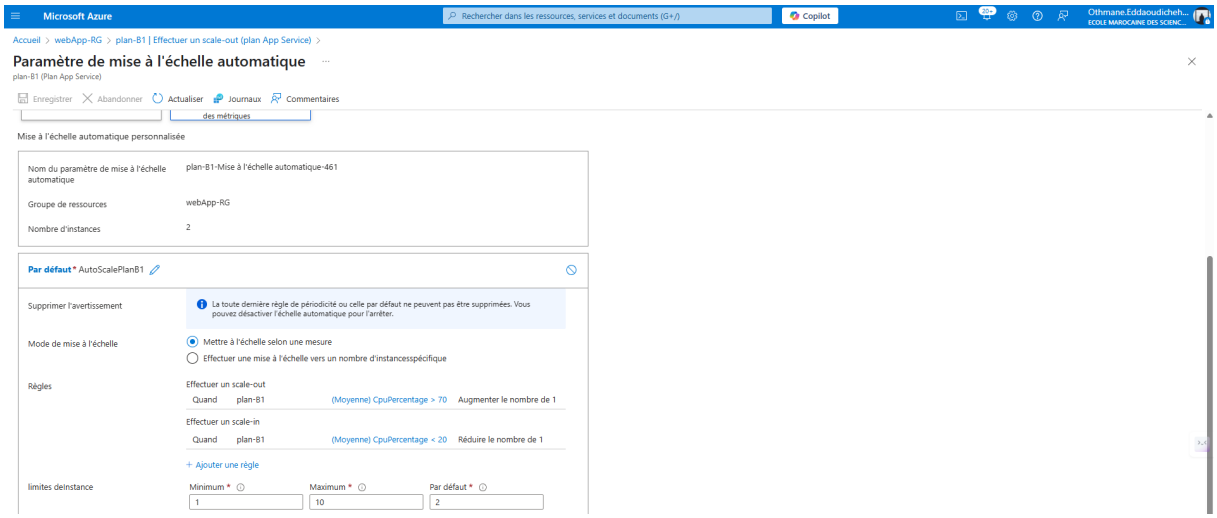

FIGURE 40 – Configuration du scale-out automatique basé sur des règles

Le nombre maximal d'instances que nous ne pouvons pas dépasser est de 10, conformément à notre configuration. Cette limite est également dictée par les capacités du plan Standard S1 qui permet jusqu'à 10 instances.

5.4 Déploiement d'applications

5.4.1 Scale-down vers Standard S1

Après avoir testé différentes configurations, nous avons effectué un scale-down du plan vers Standard S1, qui offre un bon équilibre entre performances et coût pour nos besoins.

5.4.2 Déploiement de code Java

Pour le déploiement d'une application Java dans notre première application web, nous avons utilisé l'environnement de développement Intellij avec Azure SDK :

1. Installation du SDK Azure dans Eclipse via $>$ *Intellij Marketplace*
2. Développement d'une application Java simple
3. Configuration des informations de déploiement Azure
4. Déploiement de l'application vers notre première application web (BoualiKarim)

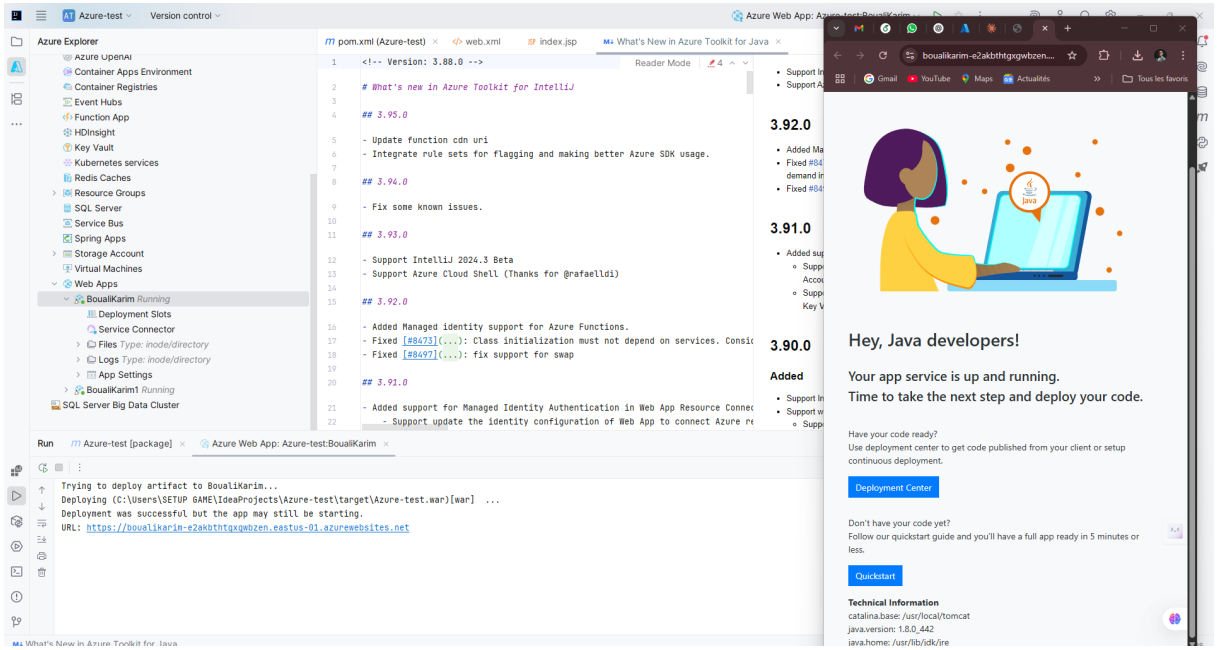

FIGURE 41 – Déploiement de l'application Java depuis Intellij vers Azure App Service

L'URL pour accéder à notre application web est : https://boualikarim.azurewebsites.net

5.4.3 Gestion des emplacements de déploiement

Pour faciliter les tests et assurer une meilleure gestion des versions, nous avons créé un emplacement de déploiement (slot) :

1. Navigation vers notre application web dans le portail Azure
2. Sélection de *Emplacements de déploiement* dans le menu de gauche
3. Création d'un nouvel emplacement nommé *test*
4. Déploiement du code vers cet emplacement de test
5. Modification légère du code pour différencier les versions

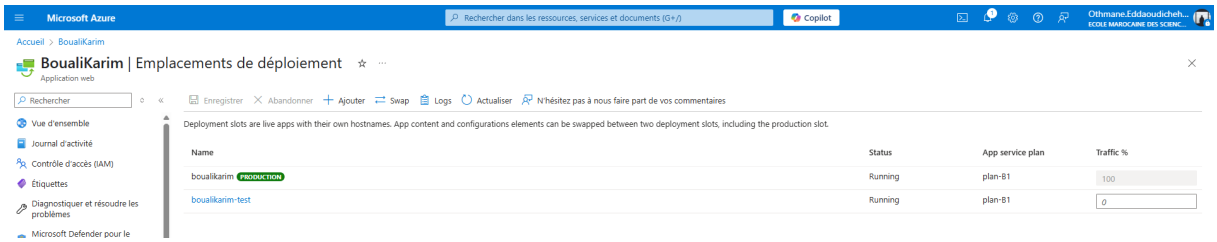

FIGURE 42 – Création et configuration d'un emplacement de déploiement pour les tests

Une fois les modifications et les tests validés dans l'emplacement de test, nous avons procédé à l'échange (swap) avec l'emplacement de production :

1. Sélection de l'emplacement *test*
2. Clic sur *Échanger*
3. Sélection de l'emplacement source (*test*) et cible (*production*)
4. Confirmation de l'échange

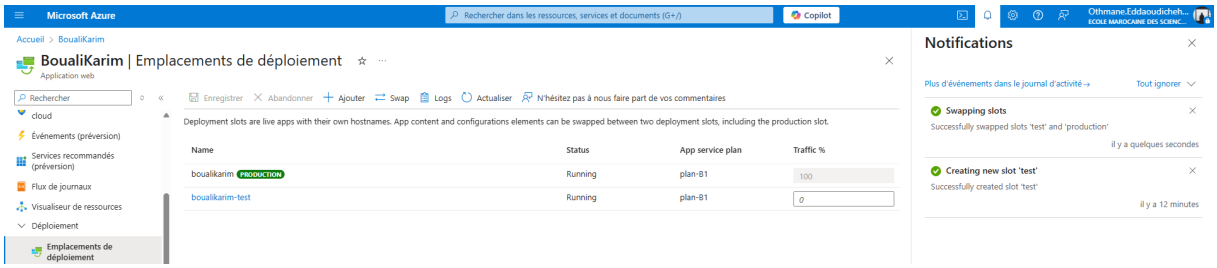

FIGURE 43 – Échange entre l'emplacement de test et l'emplacement de production

5.4.4 Personnalisation du nom de domaine

Avec notre plan Standard S1, la personnalisation du nom de domaine (DNS) est possible. Azure App Service offre deux méthodes principales pour personnaliser le nom de domaine :

1. **Achat d'un domaine directement via Azure** :
   - Navigation vers *Noms de domaine personnalisés*
   - Sélection de *Acheter un domaine*
   - Choix du nom de domaine souhaité
   - Achat et configuration automatique
2. **Utilisation d'un domaine existant** :
   - Navigation vers *Noms de domaine personnalisés*
   - Sélection de *Ajouter un nom d'hôte personnalisé*
   - Configuration des enregistrements DNS (CNAME ou A) chez le fournisseur de domaine
   - Validation de la propriété du domaine
   - Ajout du domaine à l'application web

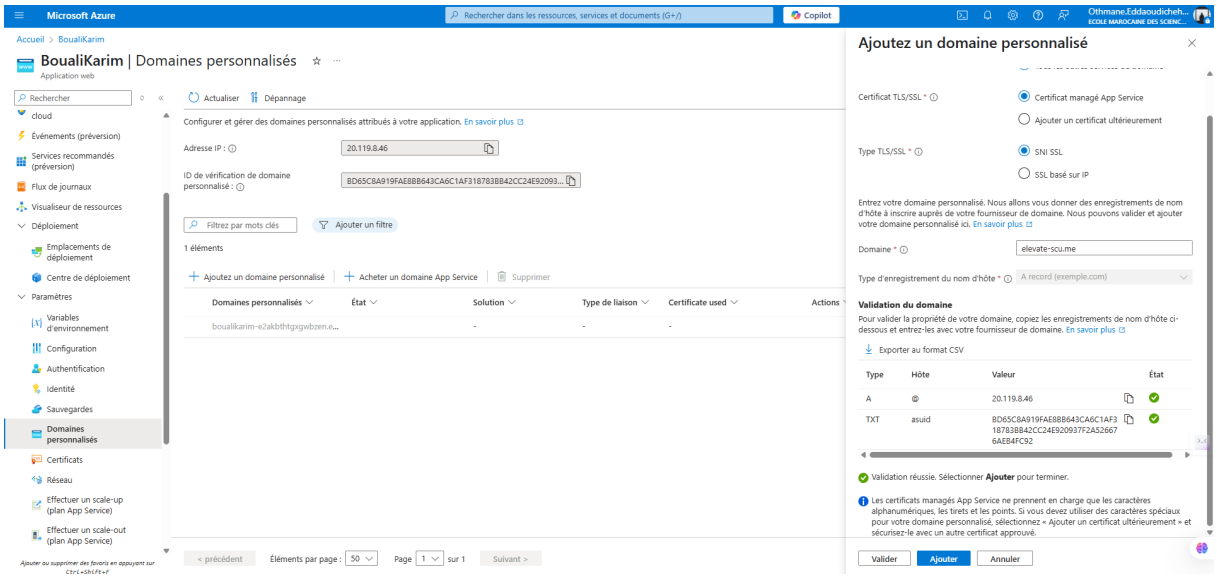

FIGURE 44 – Configuration d'un nom de domaine personnalisé pour l'application web

5.5 Conclusion

Ces travaux pratiques nous ont permis d'explorer en profondeur les fonctionnalités d'Azure App Service, de la création d'applications web à leur déploiement et leur mise à l'échelle. Nous avons appris à :

- Créer et configurer des applications web dans différents contextes
- Comprendre les contraintes liées aux régions, aux piles d'exécution et aux systèmes d'exploitation
- Effectuer la mise à l'échelle manuelle et automatique des applications
- Surveiller les performances des applications
- Déployer du code et gérer les versions avec les emplacements de déploiement
- Personnaliser les noms de domaine

Ces compétences sont essentielles pour optimiser l'utilisation d'Azure App Service dans un environnement de production, en équilibrant performances et coûts selon les besoins spécifiques de chaque application.