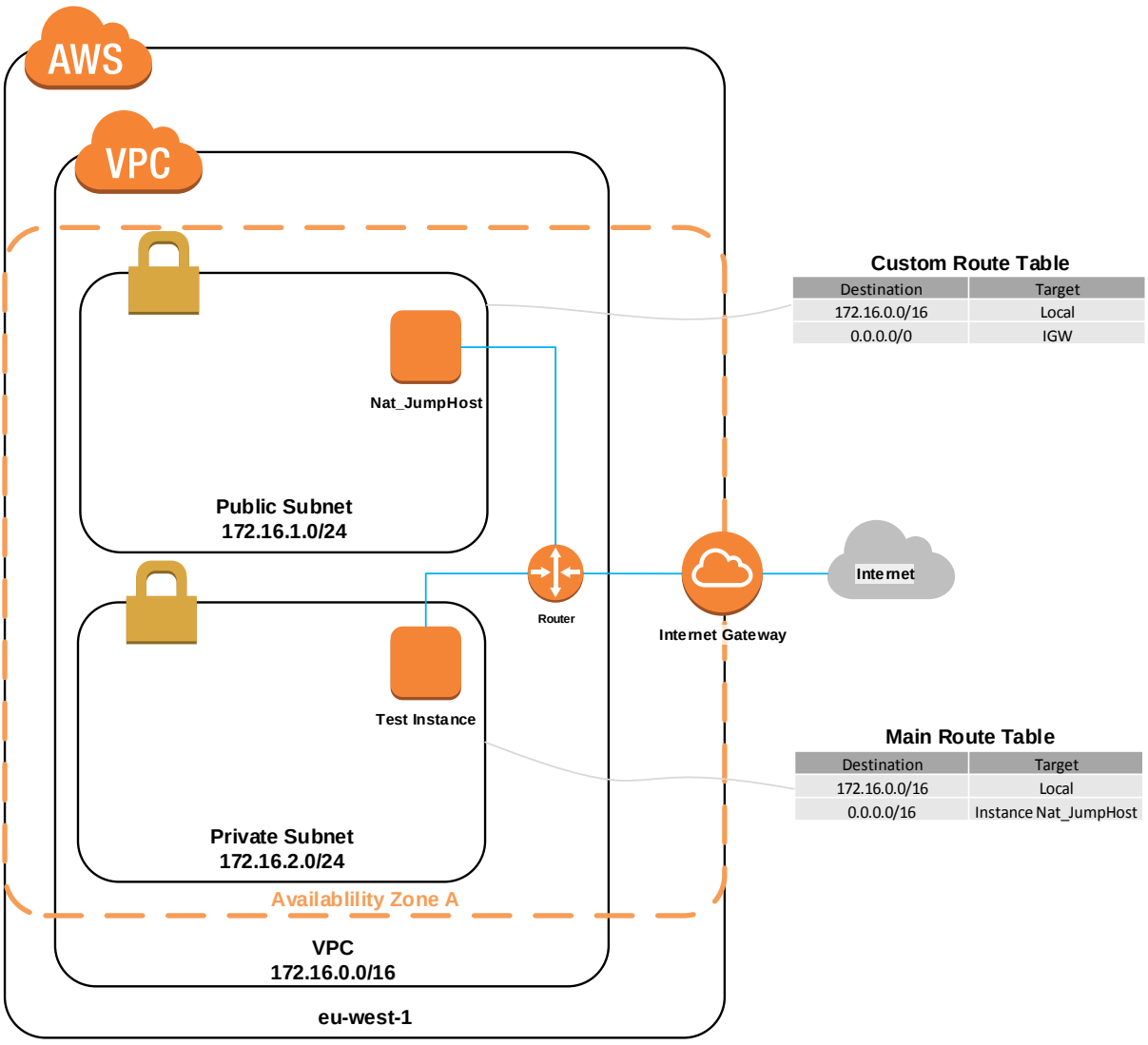
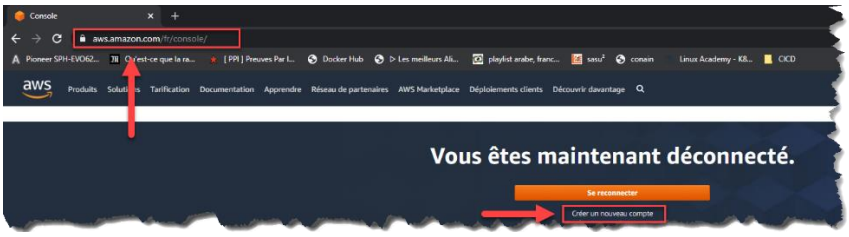
M1-APP-RI	Février 2019 **TP DevOps**

TP-1 Déploiement sous AWS

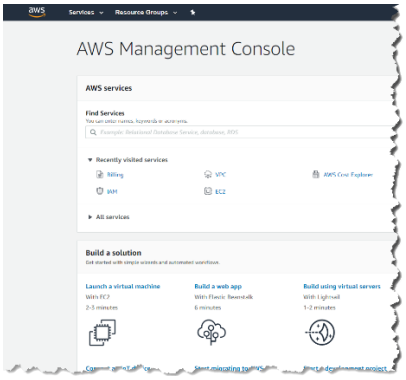


AWS Account

Créer un compte AWS Education avec un email EFREI pour obtenir un code promotionnel de 100$ de crédit AWS offert.



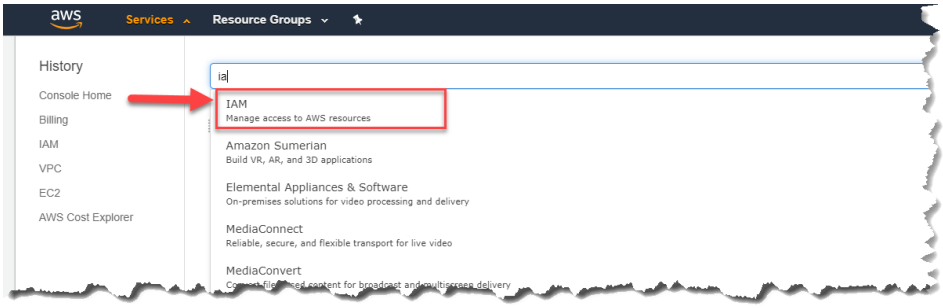
Se connecter ensuite avec les identifiants root du compte. (email + password).



Ajouter le code promotionnel dans le service « Billing » de votre compte AWS personnel.

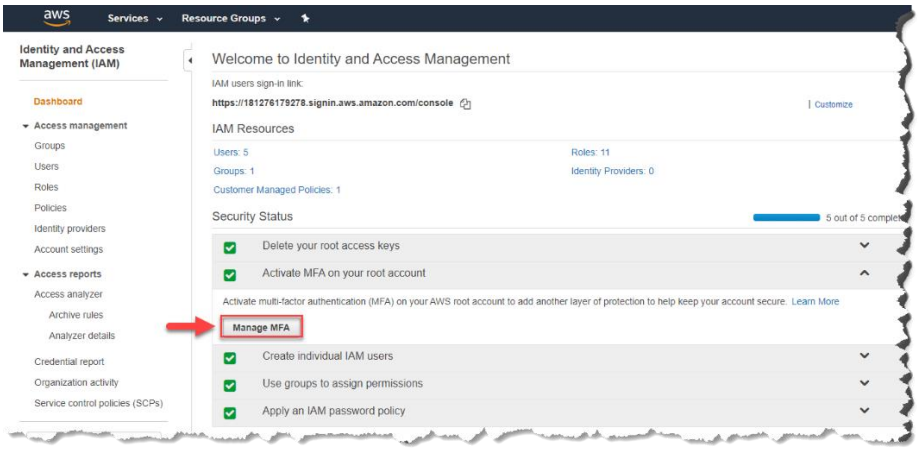
https://aws.amazon.com/premiumsupport/knowledge-center/add-aws-promotional-code/

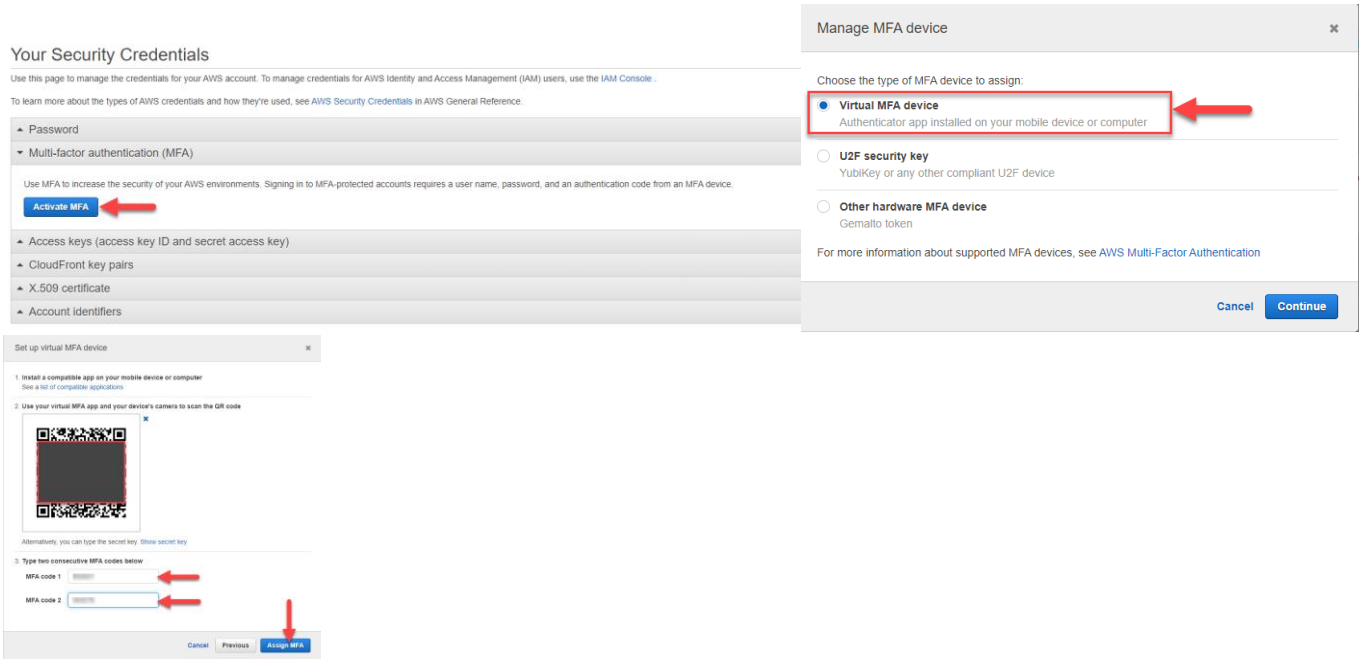
Root user



<u>Pour des raisons de sécurité, prendre l'habitude de toujours configurer un MFA pour le user « root » que pour les utilisateurs créés dans IAM.</u>

Télécharger l'application Authy depuis votre smartphone pour la gestion des MFA.

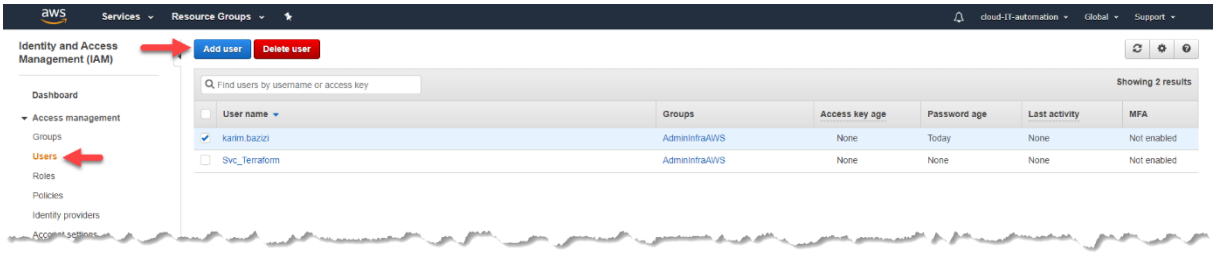




Users

Créer un utilisateur auquel nous donnons le droit Administrateur.

Se connecter ensuite avec cet utilisateur (Id du compte ou alias / User / Password).



Ne pas oublier d'assigner un MFA à vos utilisateurs.

La bonne pratique est d'appliquer une « policy » aux utilisateurs qui interdit toutes actions si l'utilisateur n'a pas de MFA configuré. Pour plus de détails suivre le lien ci-dessous :

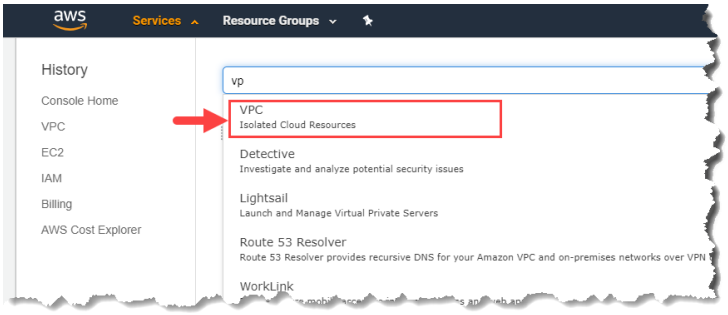
https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_examples_aws_my-sec-creds-self-manage.html

VPC (Virtual Private Cloud)

https://docs.aws.amazon.com/fr_fr/vpc/latest/userguide/VPC_Subnets.html

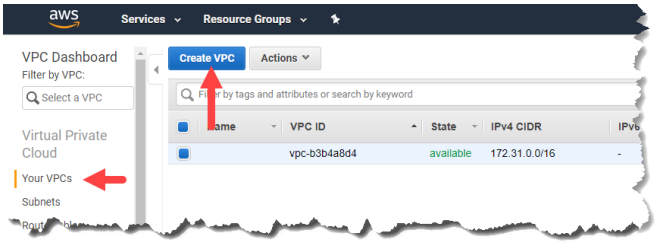
https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Scenario2.html

Se rendre dans le service VPC.

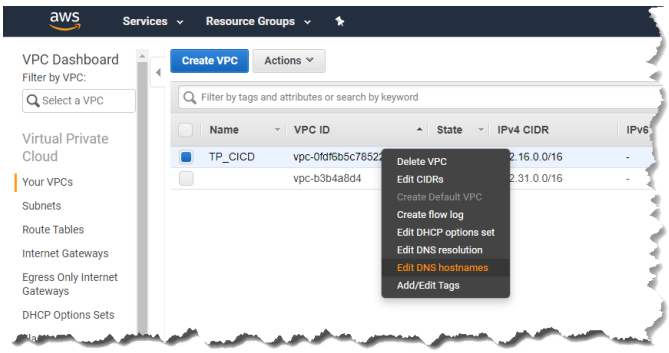


Créer un VPC avec le CIDR suivant : 172.16.0.0/16

Name Tag	CIDR
TP_CICD	172.16.0.0/16



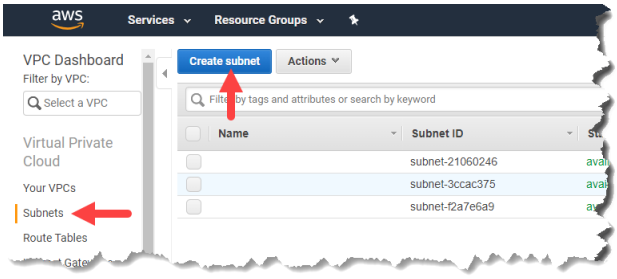
Une fois le VPC créé, activer la fonctionnalité **DNS hostnames.**

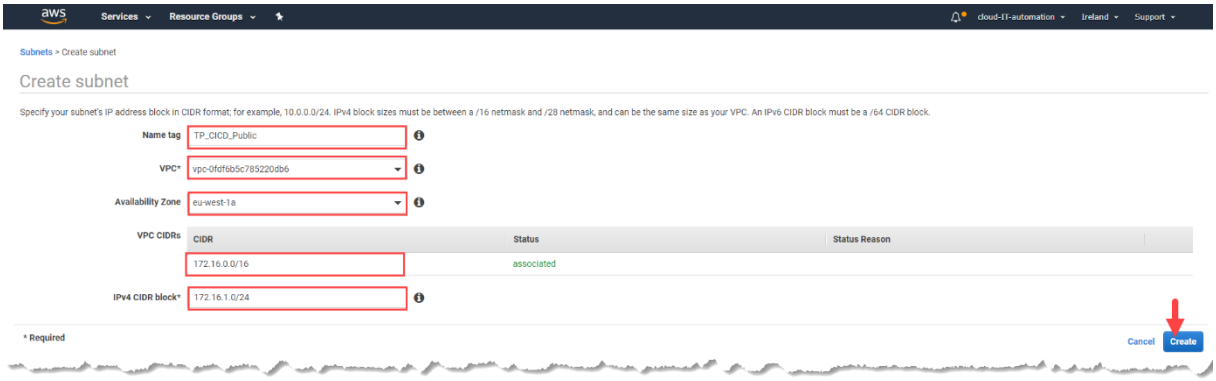


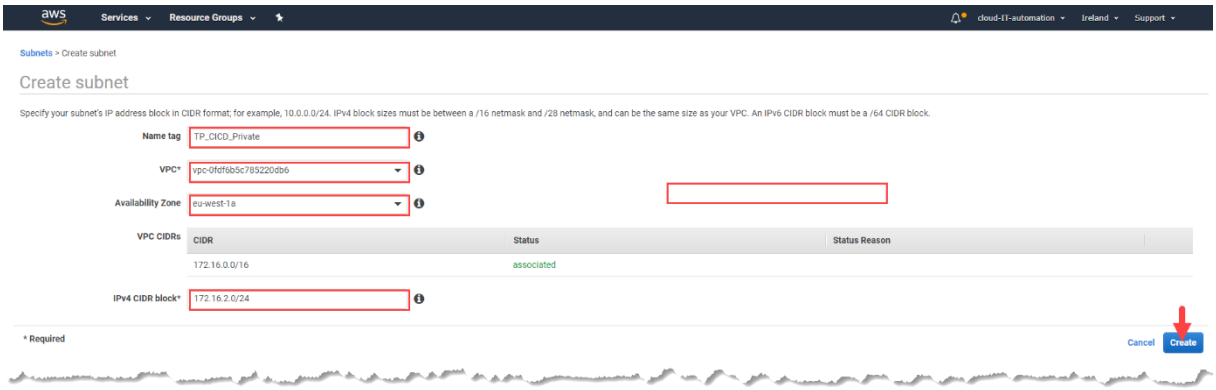
Subnets

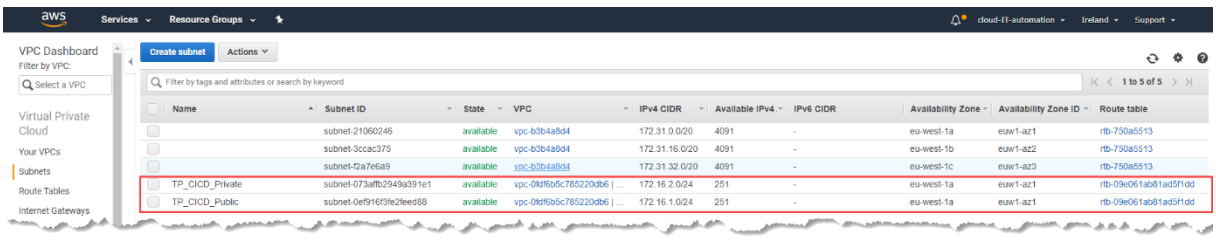
Créer les 2 subnets suivants.

Name Tag	VPC	Availability Zone	CIDR
TP_CICD_Public	172.16.1.0/24	eu-west-1a	172.16.1.0/24
TP_CICD_Private	172.16.2.0/24	eu-west-1a	172.16.2.0/24

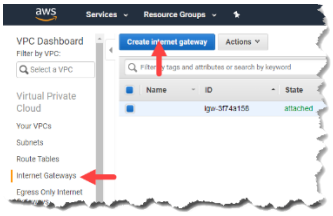






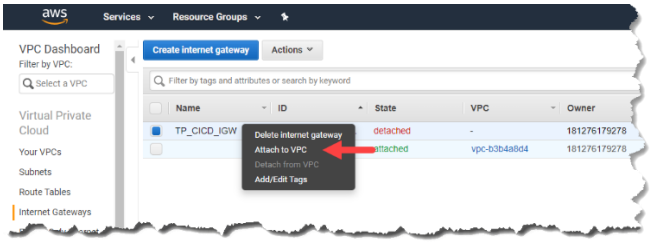


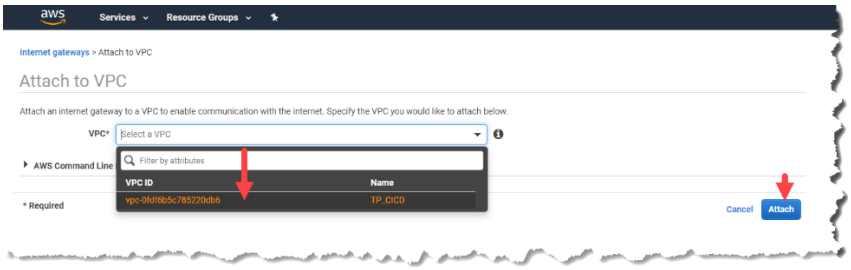
Internet Gateway



Créer une Internet Gateway, la rattacher au VPC TP_CICD et la nommer TP_CICD_IGW

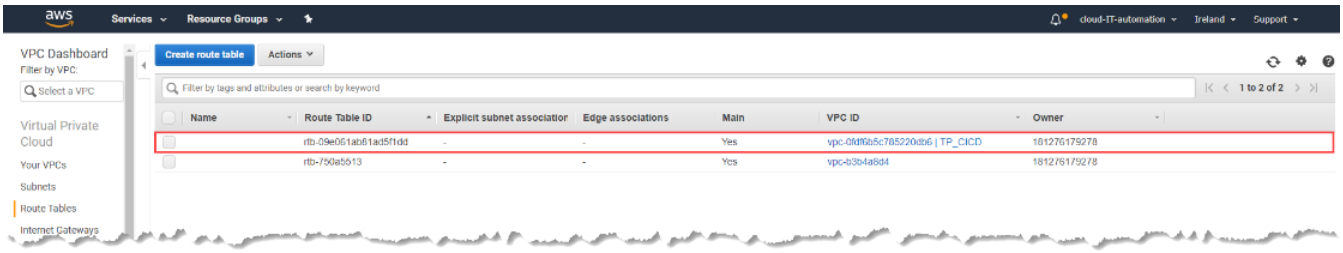
Rattacher l'internet Gateway au VPC.





Route Tables

« Route table » par défaut



Renommer le « Name Tag » de la « route table » créée par default avec le VPC par « TP_CICD_Default »

La route par défaut de la route table « TP_CICD_Default », pourra être déclarée lorsque nous aurons créé l'instance NAT.

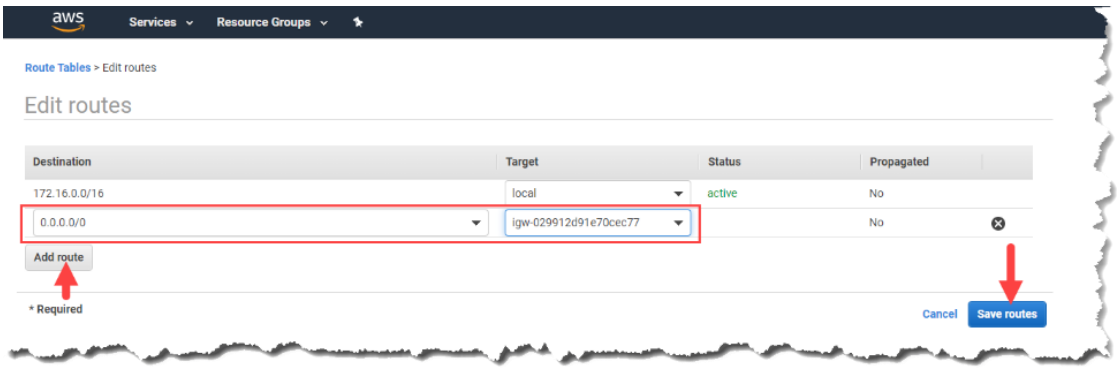
« Route table » publique

Créer la « route table » TP_CICD_Public

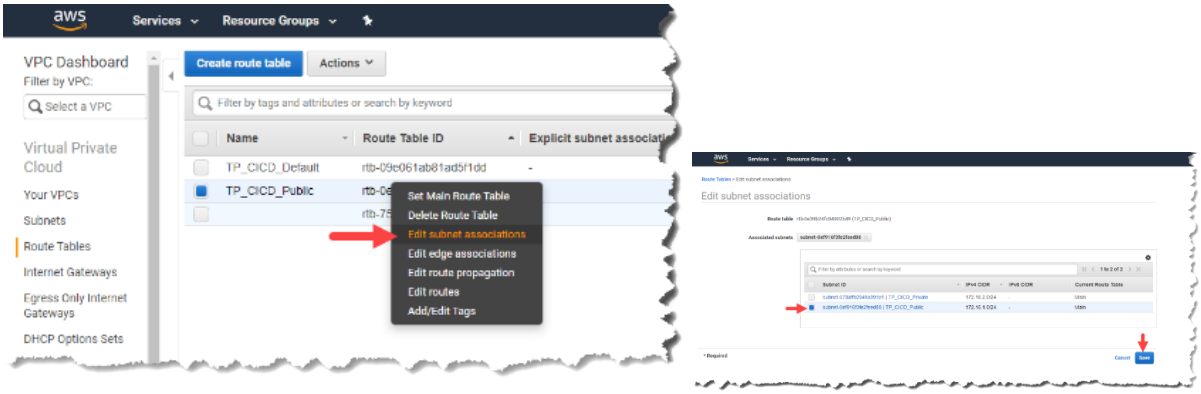
Ajouter la route suivante dans la table de routage.

Destination	Target
0.0.0.0/0	TP_CICD_IGW

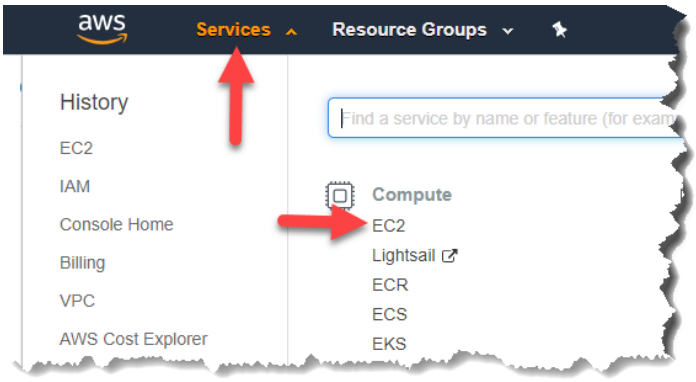
Le fait d'avoir cette route rends mon « subnet » public lorsque je l'attache à ma route table.



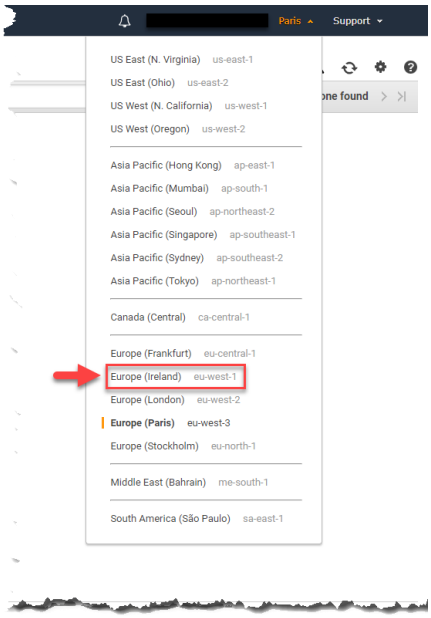
Attacher la route table « TP_CICD_Public » au subnet « TP_CICD_Public »



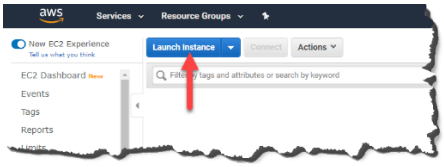
Jump Host/NAT instance



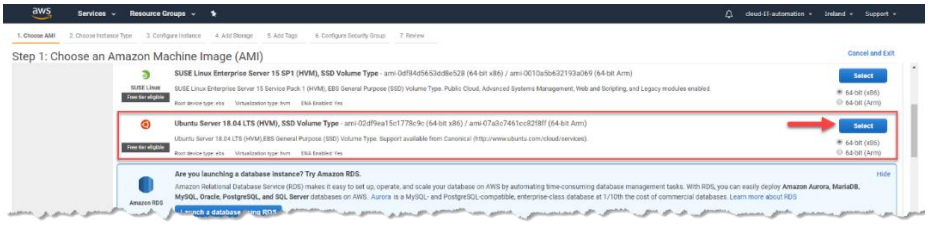
Sélectionner la région eu-west-1



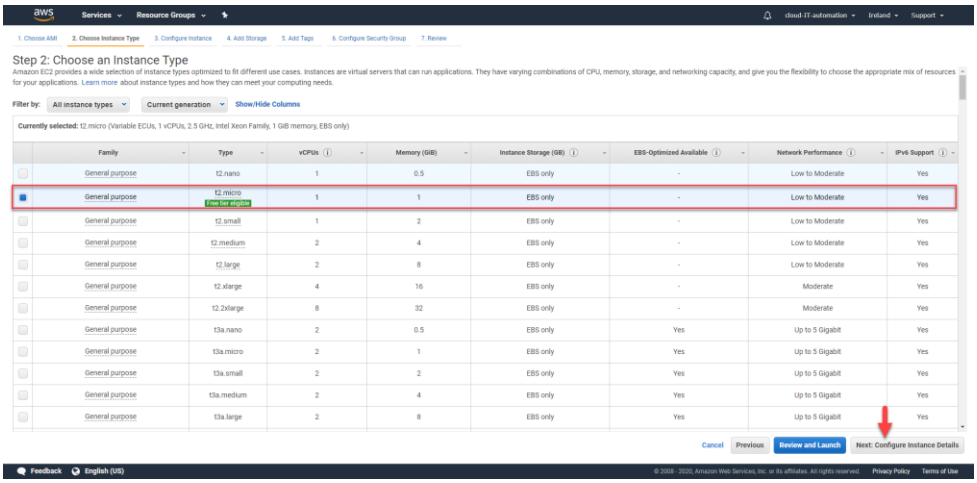
Démarrer une EC2.



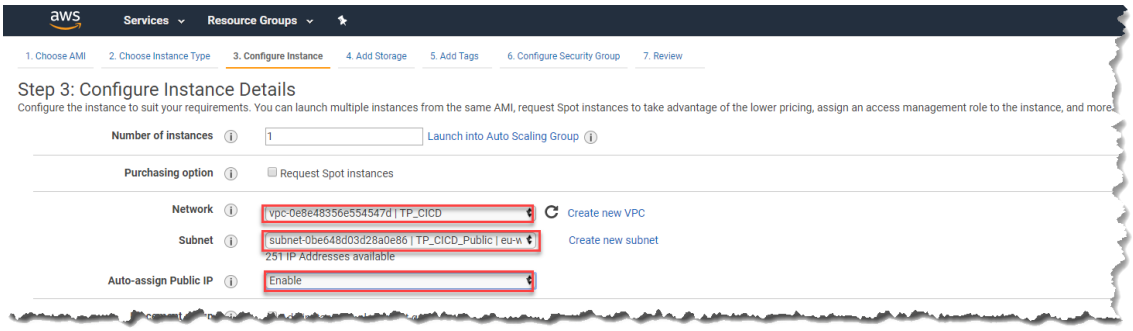
Utiliser une AMI Publique Ubuntu.



Utiliser une instance de type T2 Micro.

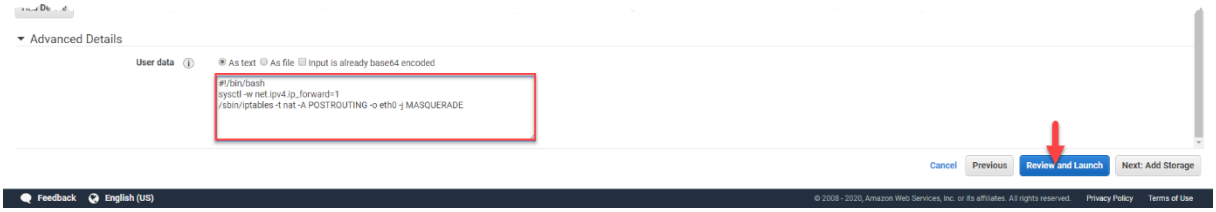


Sélectionner votre VPC et le subnet « Public » et activer l'affectation d'une IP publique.

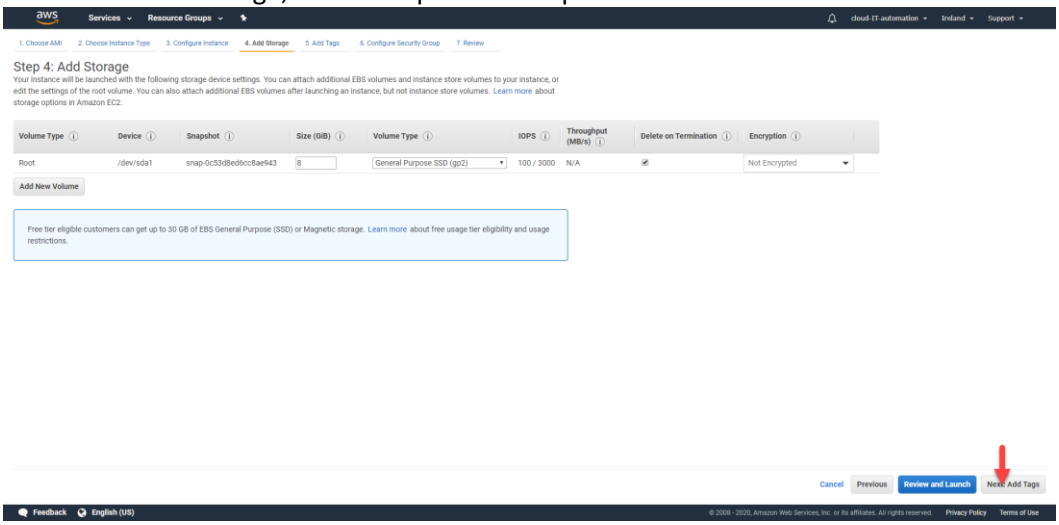


Insérer les « user data » pour activer l'IP Forwarding sur l'instance :

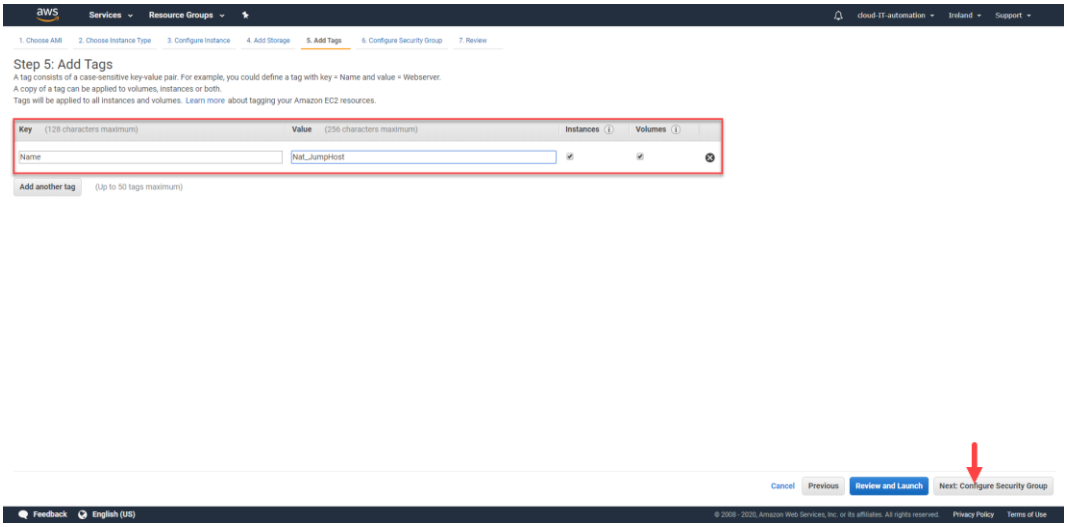
#!/bin/bash
sysctl -w net.ipv4.ip_forward=1
/sbin/iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE



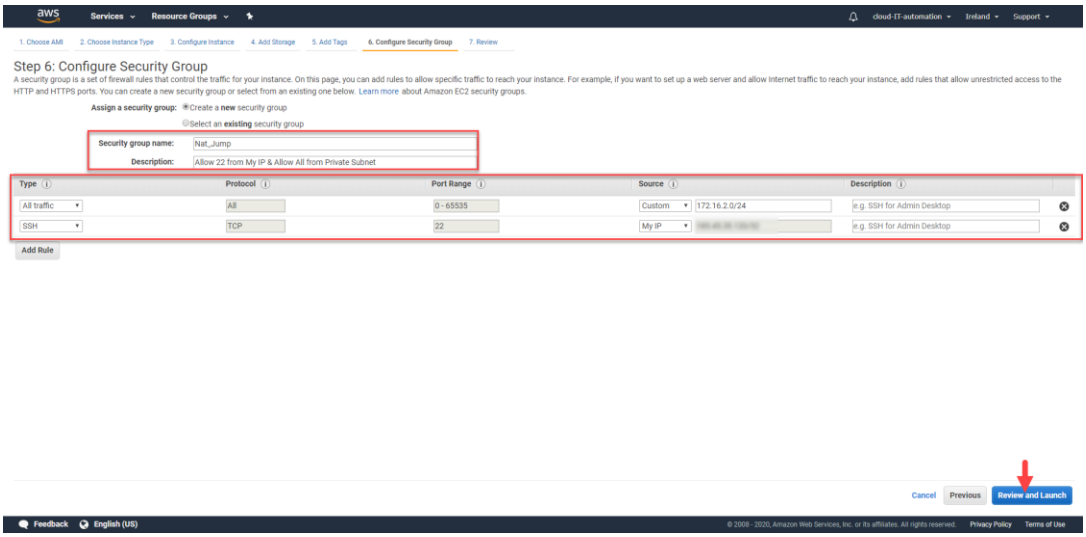
Concernant le stockage, laisser les paramètres par défaut.

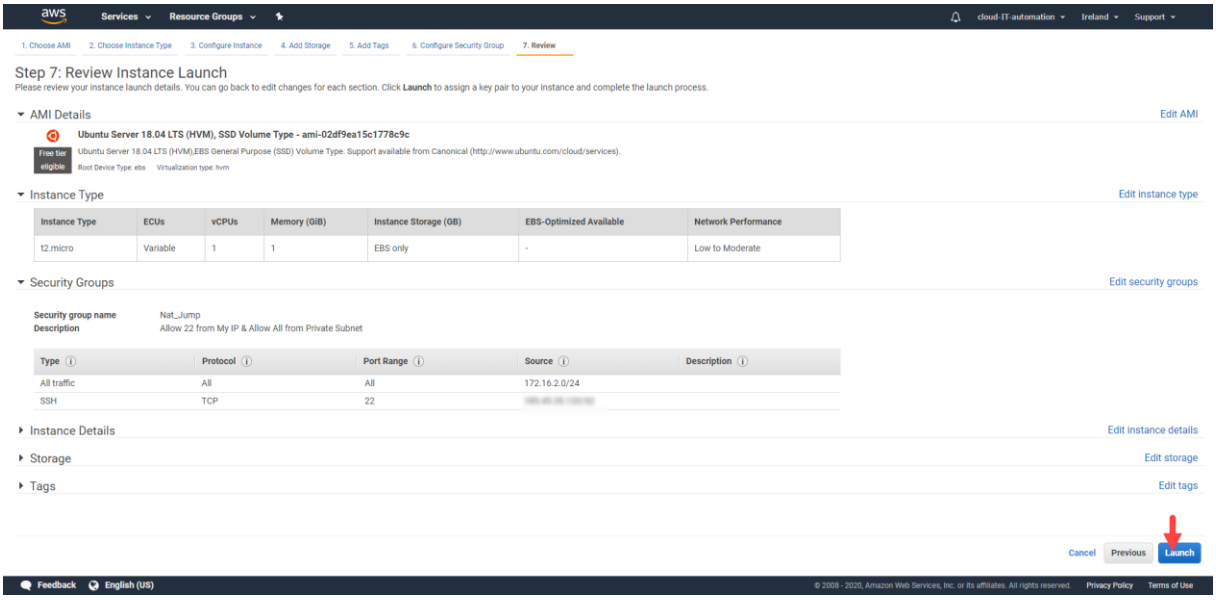


Définir un « Name Tag » « Nat_JumpHost »

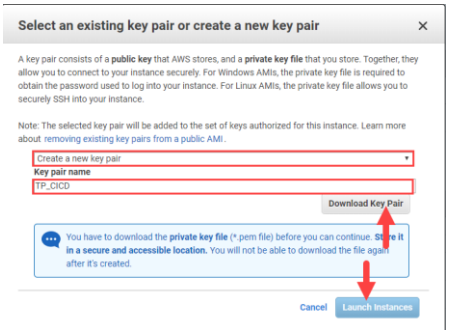


Autoriser les connexions entrantes sur le port 22 depuis sa propre IP publique (EFREI).
Autoriser toutes les connexions entrantes depuis le sous-réseau privé pour autoriser les flux provenant des futures instances déployées dans le réseau privé vers l'instance de NAT.

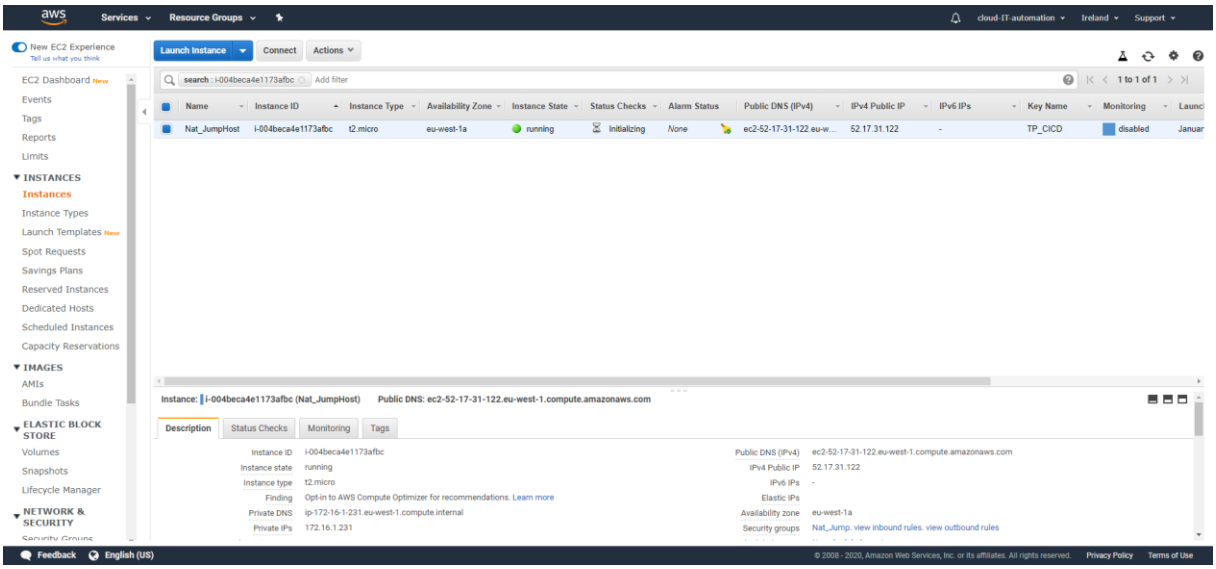




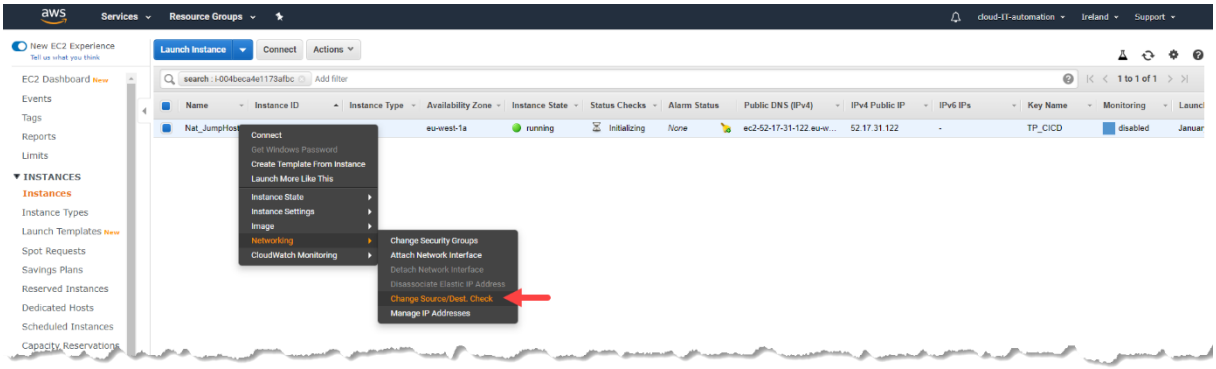
Générer et télécharger une « key pair »



Retrouver l'instance en cours de démarrage.



Désactiver source/destination check de l'instance depuis la console



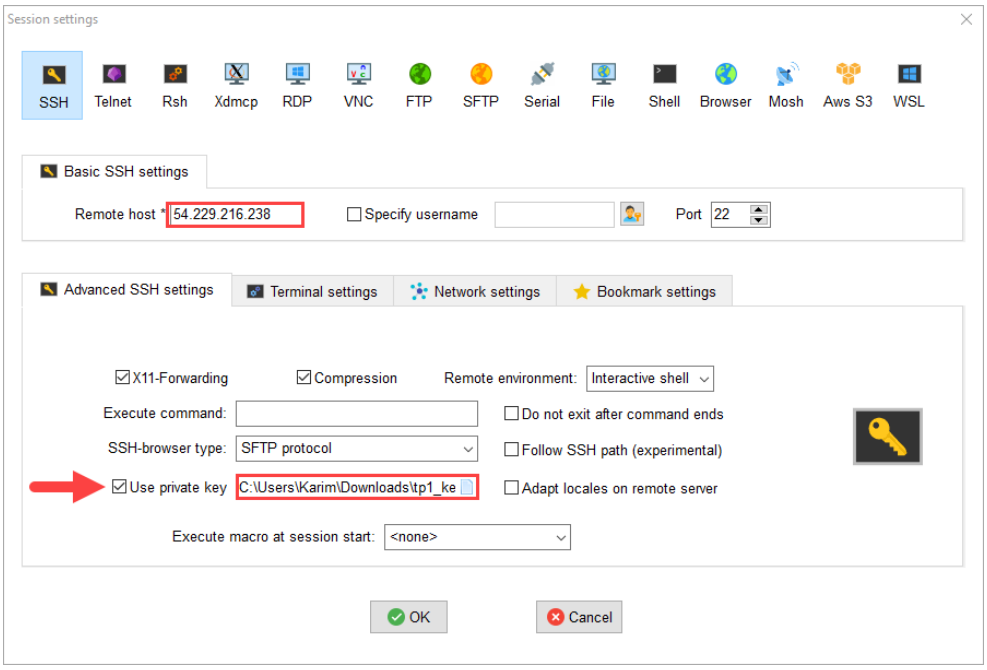
Maintenant, ajouter la route vers la « NAT Instance » dans la « main route table ».

Destination	Target
0.0.0.0/0	NAT Instance

-

Tests

Se connecter au bastion à l'aide de mobaXterm ou de putty.



Lancer une EC2 « T2.micro »vdans le private subnet, depuis la console.

Créer un « Sécurity Group » et autoriser le port 22 entrant depuis l'IP privée du Nat_JumpHost.

Se connecter en SSH à votre machine à l'aide de la « private key » précédemment copiée sur le JumpHost.

ssh -i <private_key> ubuntu@<ip_PrivateHost>

Exécuter un ping vers 8.8.8.8

Eteindre ou détruire vos instances à la fin de votre travail pour éviter de gaspiller du crédit AWS inutilement !!!