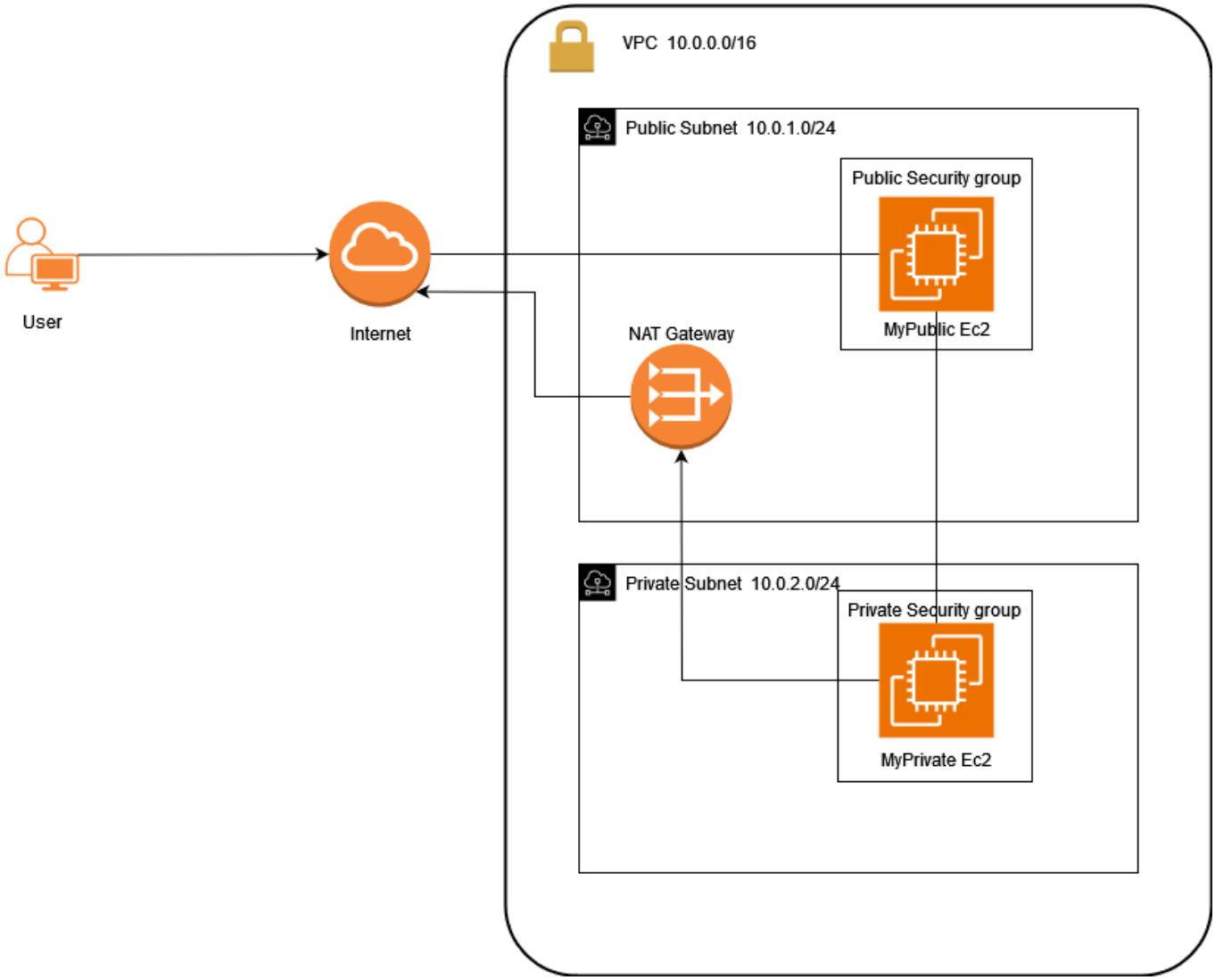
Nous allons voir comment réaliser l'infrastructure ci-dessous



Créer un vpc

Tout d'abord nous allons créer un vpc pour notre projet
aws ec2 create-vpc --cidr-block 10.0.0.0/16 --query Vpc.VpcId --output yaml
Nous pouvons ensuite le renommer avec en récupérant l'id vpc donner juste avant
aws ec2 create-tags --resources vpc-04f70745f7cbd2ec3 --tags Key=Name,Value=MyVpc

Créer un subnet

Nous allons créer un sous-réseau public pour accueillir l'application et accéder à notre second serveur plus tard en faisant bien attention de mettre notre vpc
aws ec2 create-subnet --vpc-id vpc-04f70745f7cbd2ec3 --cidr-block 10.0.1.0/24 --availability-zone us-east-1a --query Subnet.SubnetId --output yaml

Nous pouvons ensuite le renommer avec en récupérant l'id vpc donner juste avant
aws ec2 create-tags --resources subnet-02e129b73c7d08f49 --tags Key=Name,Value=MyPublicSubnet

Puis nous allons crée le sous-réseau privée avec en récupérant l'id vpc donner juste avant en faisant bien attention de mettre notre vpc
aws ec2 create-subnet --vpc-id vpc-04f70745f7cbd2ec3 --cidr-block 10.0.2.0/24 --availability-zone us-east-1a --query Subnet.SubnetId --output yaml
Nous pouvons ensuite le renommer
aws ec2 create-tags --resources subnet-063d198b0d190a305 --tags Key=Name,Value=MyPrivateSubnet

Attribuer l'ip publique

Nous allons entrée cette commande sur notre sous-réseau public pour lui allouer une ip
aws ec2 modify-subnet-attribute --subnet-id subnet-02e129b73c7d08f49 --map-public-ip-on-launch --region us-east-1

Création de la gateway

Maintenant nous allons crée la passerelle internet
aws ec2 create-internet-gateway --query InternetGateway.InternetGatewayId --output yaml
Puis la renommer
aws ec2 create-tags --resources igw-04f64ee49bea2bba1 --tags Key=Name,Value=MyGateway

Attacher une gateway à un vpc

Nous allons maintenant attacher la gateway au vpc
aws ec2 attach-internet-gateway --vpc-id vpc-04f70745f7cbd2ec3 --internet-gateway-id igw-04f64ee49bea2bba1

Créer une table de routage

Nous allons maintenant créer deux table de routage une privé et une public pour les instances en commençant la public
aws ec2 create-route-table --vpc-id vpc-04f70745f7cbd2ec3 --query RouteTable.RouteTableId --output yaml
Puis la renommer
aws ec2 create-tags --resources rtb-0c242c1ce63ca54ea --tags Key=Name,Value=MyPublicRoutable Ensuite nous allons crée le privé
aws ec2 create-route-table --vpc-id vpc-04f70745f7cbd2ec3 --query RouteTable.RouteTableId --output yaml
Puis la renommer
aws ec2 create-tags --resources rtb-0469d0f5d737d7801 --tags Key=Name,Value=MyPrivateRoutable

Ajouter une route internet pour la table de routage

Nous ajoutons la route internet précedemment créée sur la table de routage public
aws ec2 create-route --route-table-id rtb-0c242c1ce63ca54ea --destination-cidr-block 0.0.0.0/0 --gateway-id igw-04f64ee49bea2bba1

Associer la table de routage à un réseau

Nous allons ensuite associer chaque table de routage au sous-réseau correspondant
aws ec2 associate-route-table --route-table-id rtb-0c242c1ce63ca54ea --subnet-id subnet-02e129b73c7d08f49
aws ec2 associate-route-table --route-table-id rtb-0469d0f5d737d7801 --subnet-id subnet-063d198b0d190a305

Créer un groupe de sécurité

Puis nous allons créée un groupe de sécurité pour chaque instance
aws ec2 create-security-group --group-name sgPublicSubnet --description "Groupe de sécurité public" --vpc-id vpc-04f70745f7cbd2ec3
aws ec2 create-security-group --group-name sgPrivateSubnet --description "Groupe de sécurité privée" --vpc-id vpc-04f70745f7cbd2ec3

Activer ssh pour les deux groupes

Puias acctier le ssh pour les deux groupes de sécurité
aws ec2 authorize-security-group-ingress --group-id sg-0e815646cf7f72950 --protocol tcp --port 22 --cidr 0.0.0.0/0
aws ec2 authorize-security-group-ingress --group-id sg-0634a729b3d167dae --protocol tcp --port 22 --cidr 0.0.0.0/0

Activer le port pour l’application pour le groupe public

Et nous allons seulement ouvrir le port pour l'application sur le groupe de sécurité public
aws ec2 authorize-security-group-ingress --group-id sg-0e815646cf7f72950 --protocol tcp --port 5085 --cidr 0.0.0.0/0

Modifier une acl

On commence par afficher les acls pour avoir l'id

aws ec2 describe-network-acls
Puis après avoir récupérer l'id de l'acl nous pouvons modifier les règles d'entrée et de sortie
aws ec2 create-network-acl-entry --network-acl-id acl-028dbedab6a9af87c --rule-number 110 --ingress --protocol ssh --rule-action allow --cidr-block 0.0.0.0/0 --port-range From=22,To=22
aws ec2 create-network-acl-entry --network-acl-id acl-028dbedab6a9af87c --rule-number 120 --ingress --protocol tcp --rule-action allow --cidr-block 0.0.0.0/0 --port-range From=5085,To=5085
aws ec2 create-network-acl-entry --network-acl-id acl-028dbedab6a9af87c --rule-number 130 --ingress --protocol tcp --rule-action allow --cidr-block 0.0.0.0/0 --port-range From=1024,To=65535

Créer une ip elastic

Puis nous allons crée une adress ip elastic pour la nat gateway
aws ec2 allocate-address --domain vpc --query 'AllocationId' --output yaml --region us-east-1 --tag-specifications 'ResourceType=elastic-ip,Tags=[{Key=Name,Value=}]'

Créer la nat gateway

Maitenant nous pouvons crée notre nat gateway
aws ec2 create-nat-gateway --subnet-id subnet-02e129b73c7d08f49 --allocation-id eipalloc-065616131ef9d7444
Lier la nat au groupe privée pour accéder à internet par la suite
aws ec2 create-route --route-table-id rtb-0469d0f5d737d7801 --destination-cidr-block 0.0.0.0/0 --nat-gateway-id nat-010d83e04a97124a1

Créer une instance EC2 pour l’application

Et maintenant nous allons voir comment crée deux instance debian pour notre infrastructure en commencant par la public
aws ec2 run-instances --image-id ami-058bd2d568351da34 --count 1 --instance-type t2.micro --key-name mysshkey --security-group-ids sg-0e815646cf7f72950 --subnet-id subnet-02e129b73c7d08f49 --associate-public-ip-address --tag-specifications --region us-east-1 --tag-specifications 'ResourceType=instance,Tags=[{Key=DebianServer,Value=Beta}]'
On peut ajouter un tag avec cette commande
aws ec2 create-tags --resources i-00adaa6c294898d29--tags Key=Name,Value=MyPublicEc2

Afficher la vm avec son ip publique

aws ec2 describe-instances --filters Name=instance-state-name,Values=running

Elle s'affichera dans "PublicIp: "

Créer une instance EC2 privée pour l’application
aws ec2 run-instances --image-id ami-058bd2d568351da34 --count 1 --instance-type t2.micro --key-name mysshkey --security-group-ids sg-0634a729b3d167dae --subnet-id subnet-063d198b0d190a305 --tag-specifications --region us-east-1 --tag-specifications 'ResourceType=instance,Tags=[{Key=DebianServer,Value=Beta}]'

On peut ajouter un tag avec cette commande
aws ec2 create-tags --resources i-0352cdd8d1f62c8ab--tags Key=Name,Value=MyPrivateEc2

Afficher la vm avec son @ip publique

aws ec2 describe-instances --filters Name=instance-state-name,Values=running

Elle s'affichera dans "PublicIp: "