SETTING UP BASTION / NAT INSTANCE IN AWS

Warning : when this documentation is released, the IP have already been changed !

Table des matières

[Aim 2](#_Toc515210064)

[Architecture summary 3](#_Toc515210065)

[How will we test ? 4](#_Toc515210066)

[Creating a key pair 6](#_Toc515210067)

[Converting the pair key from .pem to .ppk 7](#_Toc515210068)

[Generate two elastic IP 8](#_Toc515210069)

[Generate a VPC 9](#_Toc515210070)

[Shortcut creation 9](#_Toc515210071)

[Checks and enhancements of the VPC created by the wizard 10](#_Toc515210072)

[NAT instance 10](#_Toc515210073)

[VPC 11](#_Toc515210074)

[Subnets 12](#_Toc515210075)

[Bastion creation 15](#_Toc515210076)

[Test of the ping to google.com 18](#_Toc515210077)

[0 *manual* Check isolated instance has no internet IP 18](#_Toc515210078)

[1 *manual* Connect to bastion through SSH 18](#_Toc515210079)

[2 *manual* Connect to isolated instance through bastion 20](#_Toc515210080)

[3 *manual* ping google.com inside isolated instance 23](#_Toc515210081)

[4 *automatic* : isolated instance call NAT 23](#_Toc515210082)

[5 : *automatic*  NAT pings google.com, replacing the network IP with an internet IP 24](#_Toc515210083)

[6 : *automatic*  google.com answers to the call to NAT with the internet IP 24](#_Toc515210084)

[7 : *automatic*  NAT transfers google.com answers to isolated instance routing the IP 24](#_Toc515210085)

[8 : *manual* User can check the call has been returned : no time to leave ! 24](#_Toc515210086)

# Aim

In AWS, we want to create :

* VPC with private and public
* NAT instance / public
* Bastion / public
* Isolated server / private

The aim :

* Check that the isolated instance is able to ping google.com, despite its private location

# Architecture summary

AWS VPC

Private : 10.0.1/24

Public : 10.0.0/24

NAT

Network IP: 10.0.0.192

Internet IP : [52.214.232.124](https://eu-west-1.console.aws.amazon.com/ec2/v2/home?region=eu-west-1#Addresses:search=52.214.232.124;sort=publicIp)

Isolated instance

Network IP: 10.0.0.168

No access to internet

Aim : we want this isolated instance to ping « google.com »

How ???

Bastion

Network IP: 10.0.0.168

Internet IP : [52.208.163.155](https://eu-west-1.console.aws.amazon.com/ec2/v2/home?region=eu-west-1#Addresses:search=52.208.163.155;sort=publicIp)

Bastion accepts port 22

SSH protocol

NAT can call internet through ports 80 http and 433 https protocol

Internet

# How will we test ?

Private : 10.0.1/24

Public : 10.0.0/24

NAT

Network IP: 10.0.0.192

Internet IP : [52.214.232.124](https://eu-west-1.console.aws.amazon.com/ec2/v2/home?region=eu-west-1#Addresses:search=52.214.232.124;sort=publicIp)

3

6

Isolated instance

Network IP: 10.0.0.168

No access to internet

2

5

4

This isolated instance can ping « google.com »

Bastion

Network IP: 10.0.0.168

Internet IP : [52.208.163.155](https://eu-west-1.console.aws.amazon.com/ec2/v2/home?region=eu-west-1#Addresses:search=52.208.163.155;sort=publicIp)

google.com Internet

1

The steps :

* 0 *manual* Check isolated instance has no internet IP
* 1 *manual* Connect to bastion through SSH
* 2 *manual* Connect to isolated instance through bastion
* 3 *manual* ping google.com inside isolated instance
* 4 *automatic* : isolated instance call NAT
* 5 : *automatic*  NAT pings google.com, replacing the network IP with an internet IP
* 6 : *automatic*  google.com answers to the call to NAT with the internet IP
* 7 : *automatic*  NAT transfers google.com answers to isolated instance routing the IP
* 8 : *manual* User can check the call has been returned : no time to leave !

# Creating a key pair

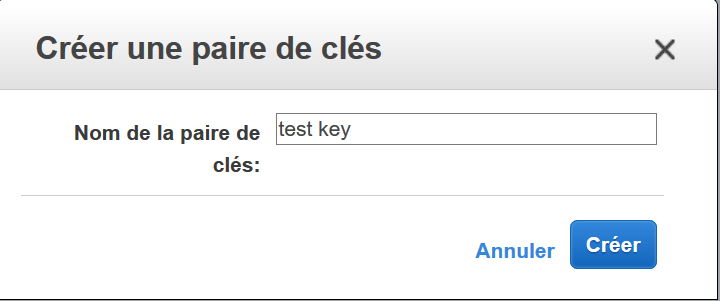
This is needed for accessing remotely to :

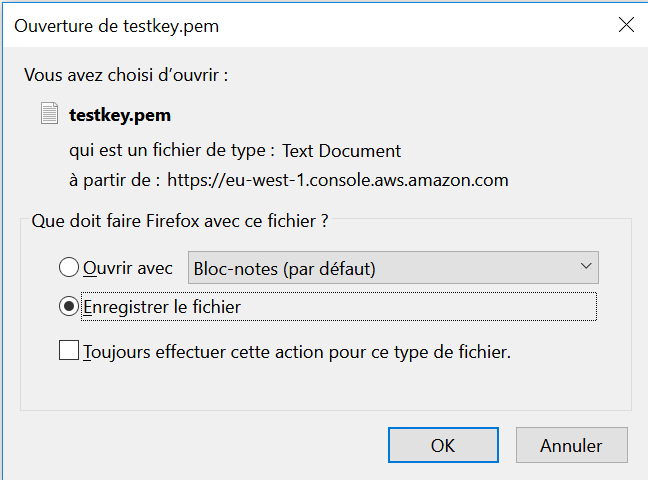
* Bastion with Putty
* Private instance with Putty

As a summary Desktop call bastion which call private instance

Configuration : EC2 > key pair







Save carefully this file it won’t never be generated again

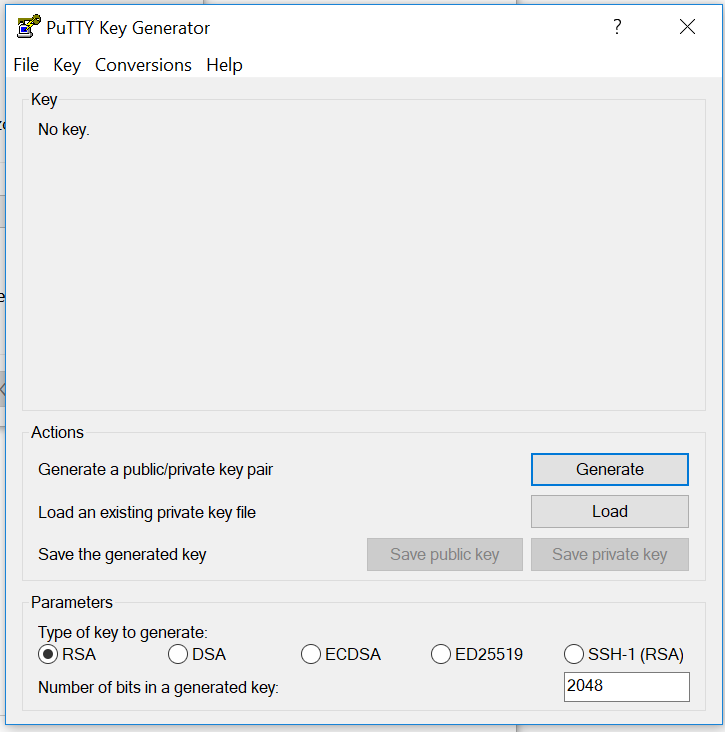
# Converting the pair key from .pem to .ppk

.pem is the aws key format

.ppk is the usual format

We need the .ppk format with Putty

Install on computer puttygen, it is found in Putty package :



 the .pem file provided by aws

 and get the .ppk converted thanks to Puttygen

# Generate two elastic IP

By default, aws set up is isolated from external world

If you want to connect to the web, generate elastic IP, one per instance requiring it in our project:

* Bastion which is receiving the putty SSH call from our desktop
* NAT instance which will call internet for us

Configuration : EC2 > elastic IP address

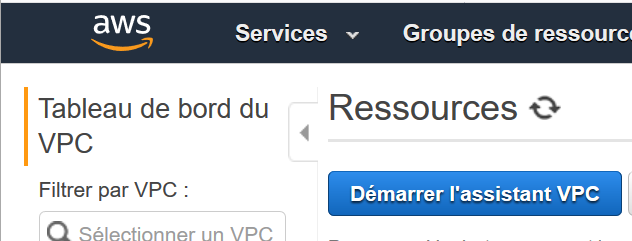


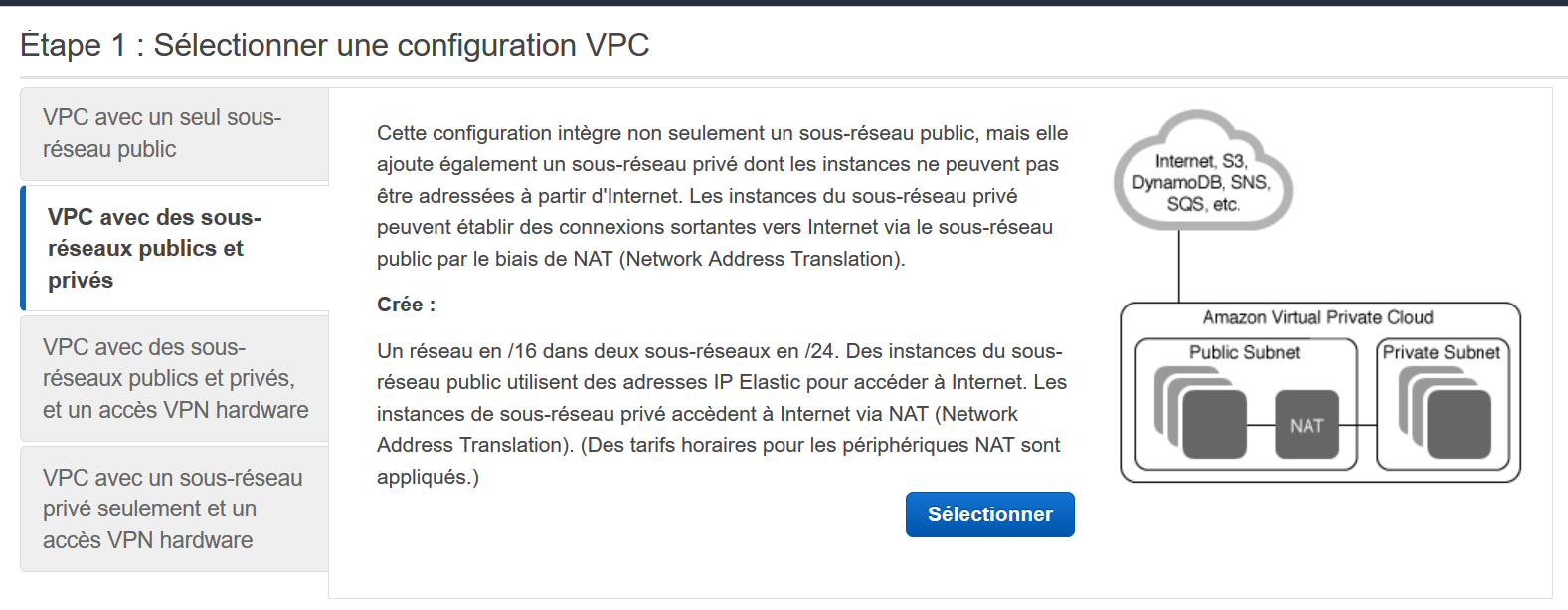


# Generate a VPC

## Shortcut creation

Configuration : VPC > dashboard > start the VPC wizard





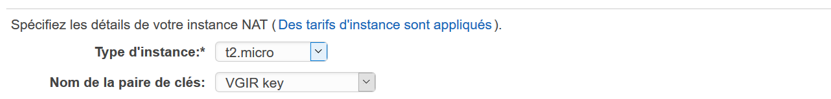


Change it for NAT instance !

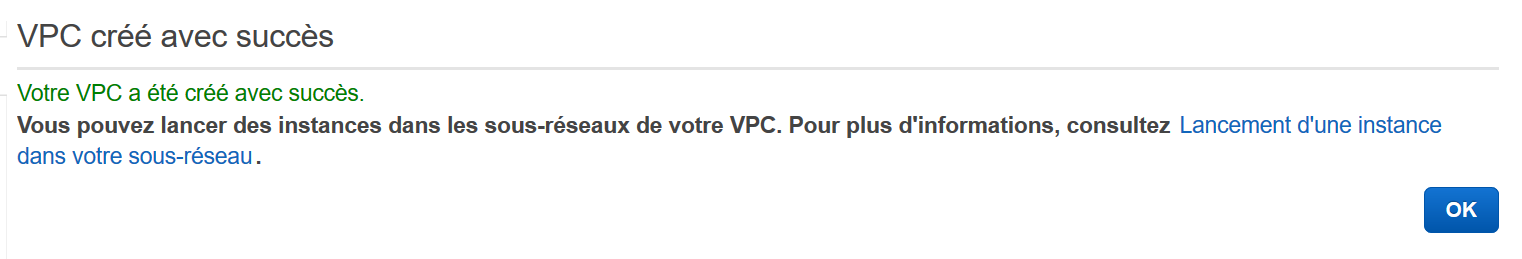
Public subnet has 10.0.0.xxx adresses

Private subnet has 10.0.1.xxx adresses

VPC NAME



The VPC has been successfully created



## Checks and enhancements of the VPC created by the wizard

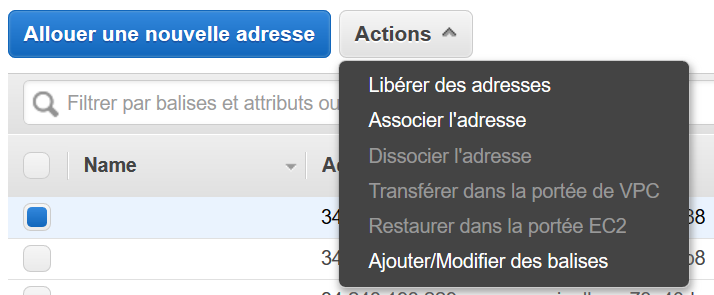
### NAT instance

A NAT instance has been created :



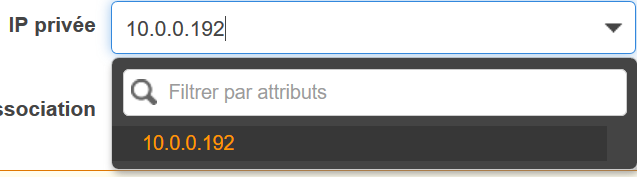
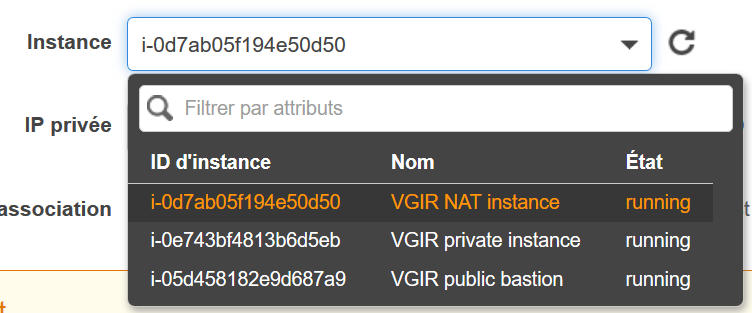
We add it the first elastic IP :

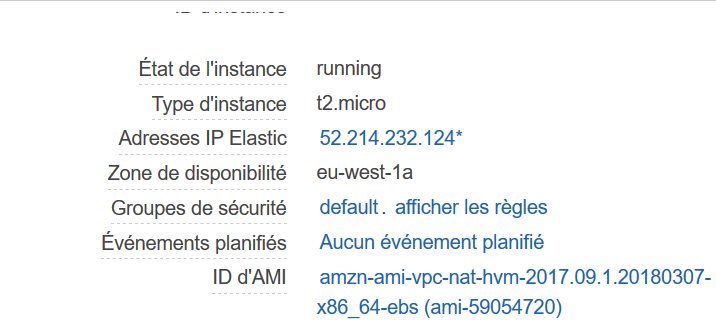
Configuration : EC2 > elastic IP > select IP > action > link the address



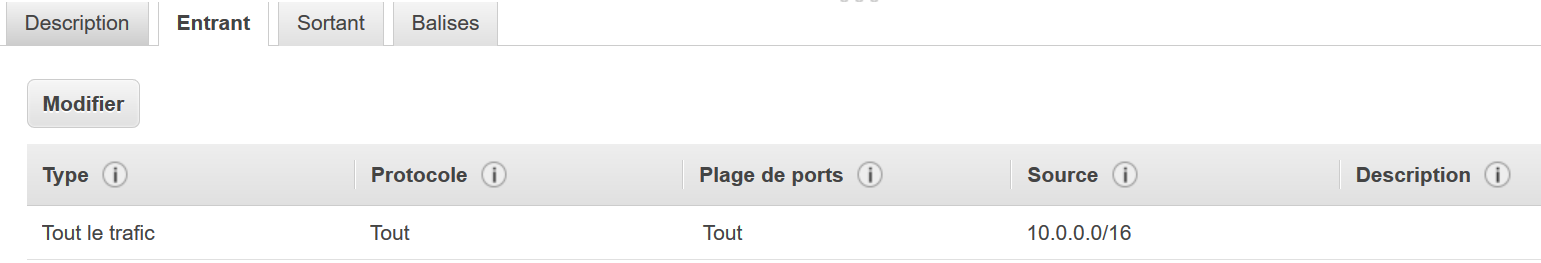
Its IP is [52.208.163.155](https://eu-west-1.console.aws.amazon.com/ec2/v2/home?region=eu-west-1#Addresses:search=52.208.163.155;sort=publicIp)

Choose both the instance ID and the private IP :

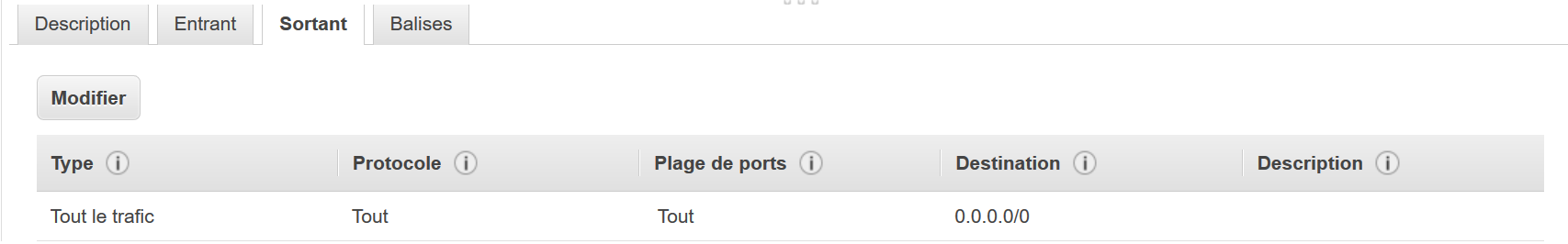




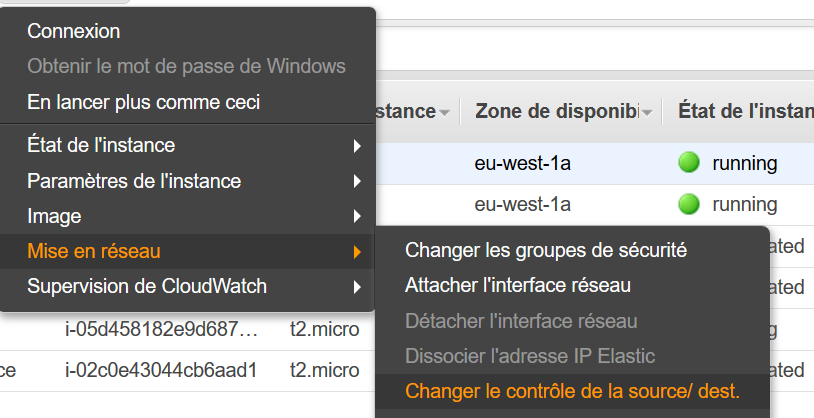
The security group of the NAT instance accepts all the inbound traffic from the aws network :

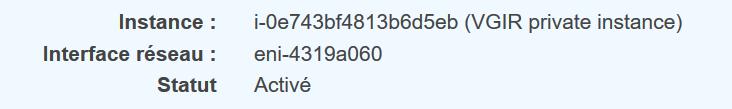


And it can communicate in outbound to everybody with all the ports :



The wizard has set up destination check, and it « activates » the purpose of the NAT :



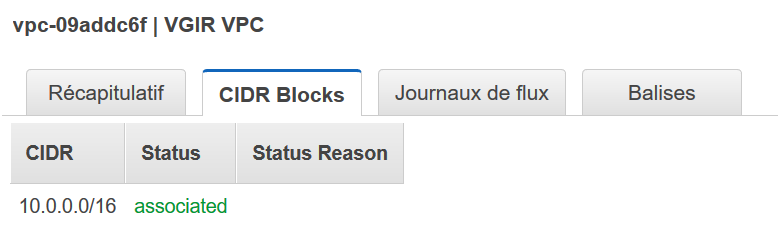


### VPC

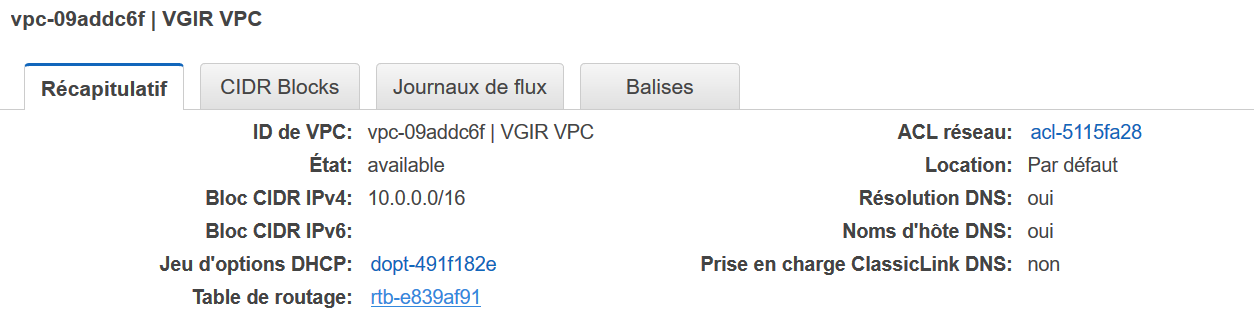
A VPC of course has been created :



It’s managing the 10.0.xxx.xxx adresses :



It has a main route table :



### Subnets

They have been created as requested

Configuration : VPC > subnet

Public :





Private :

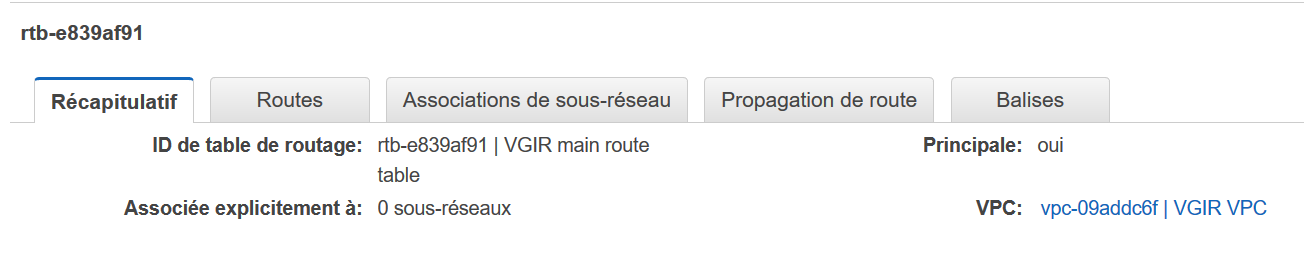


The route tables are explained in the following paragraph

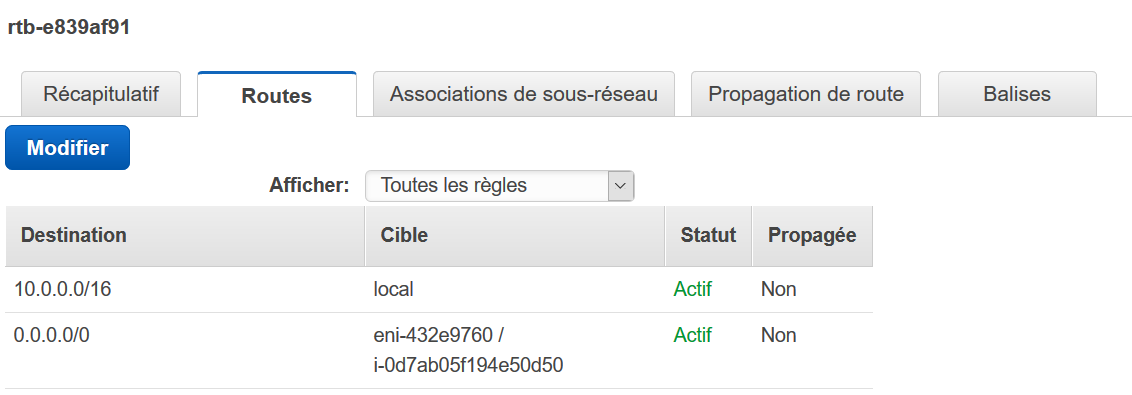
Route table

Configuration : VPC > route table



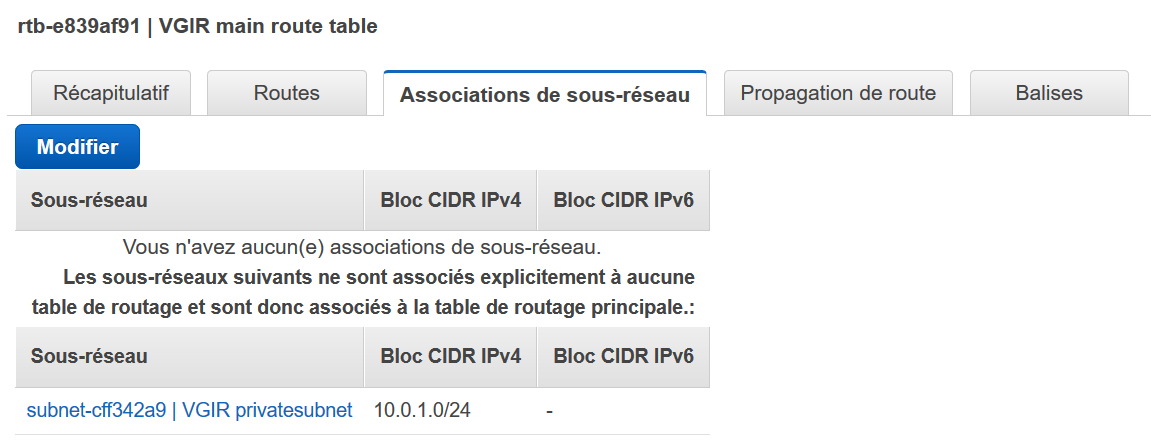


Thanks to the wizard, the NAT instance has been already attached to it : remark, aws displays only its id, check in EC2 if you have doubt

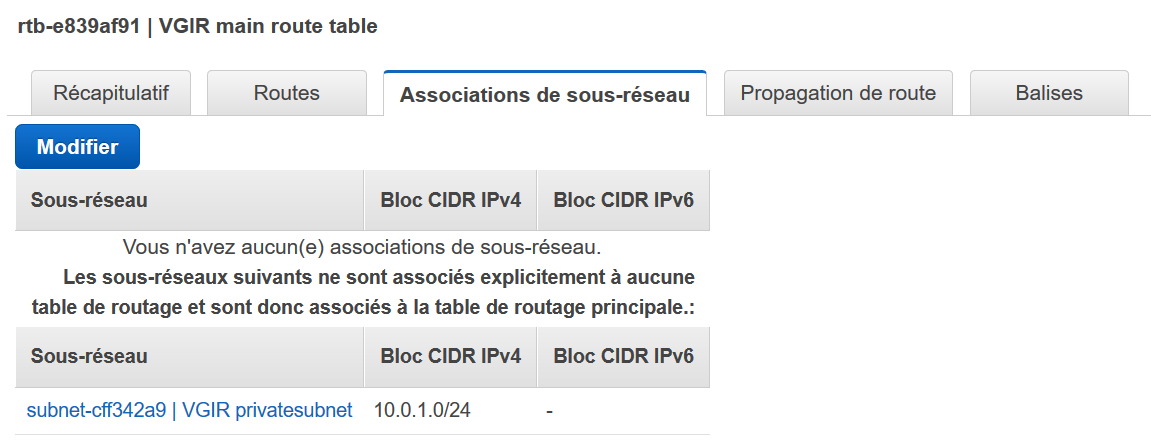


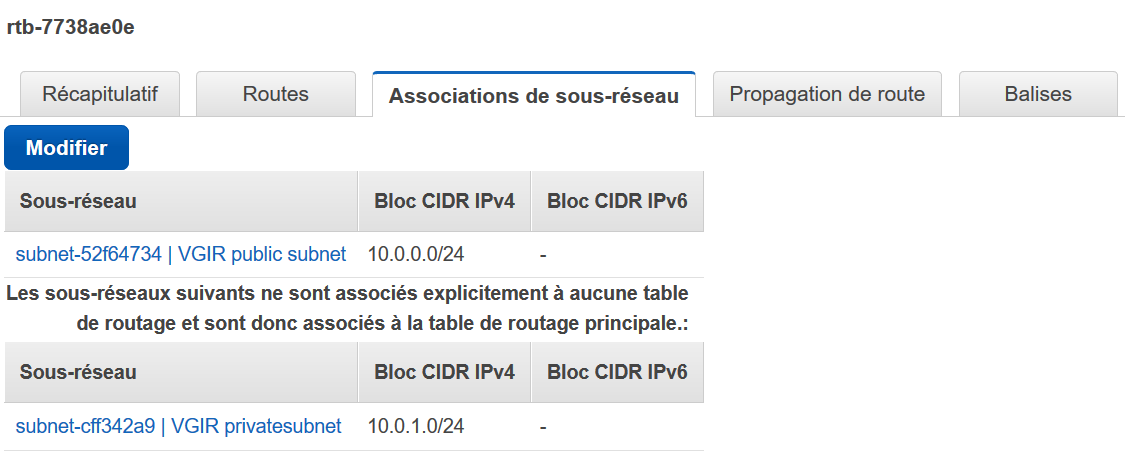


As the private subnet has been placed by the wizard inside the main route table, it is already benefiting from the NAT instance :



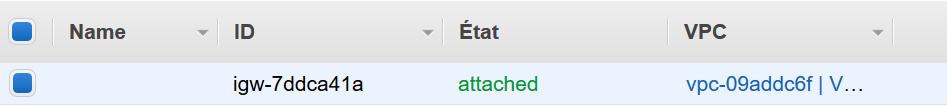
The wizard has set up the private subnet in the secondary route table, and through an internet gateway : this could be changed , and use the NAT if we wish so :

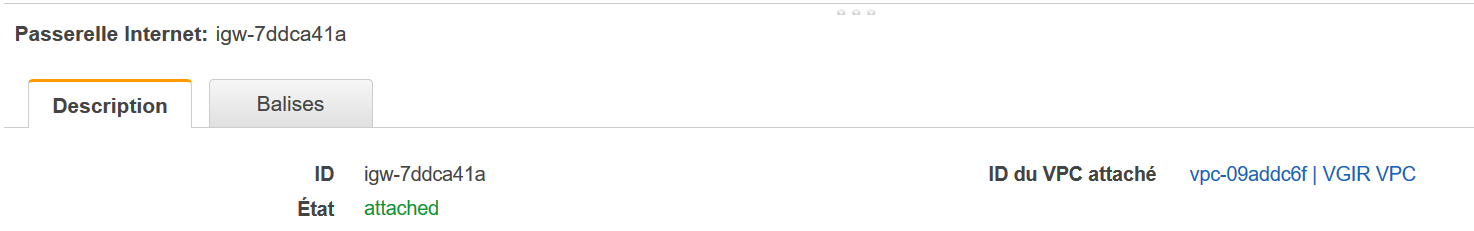




Internet Gateway :

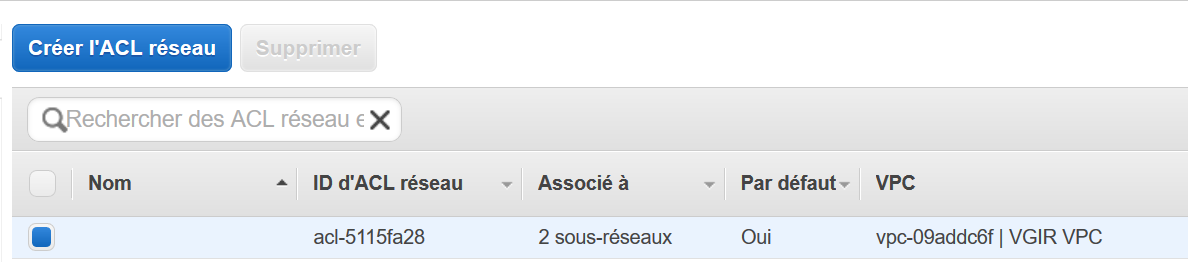
A mentioned previously, even if we asked a NAT instance to the wizard, he created an internet gateway for the public subnet



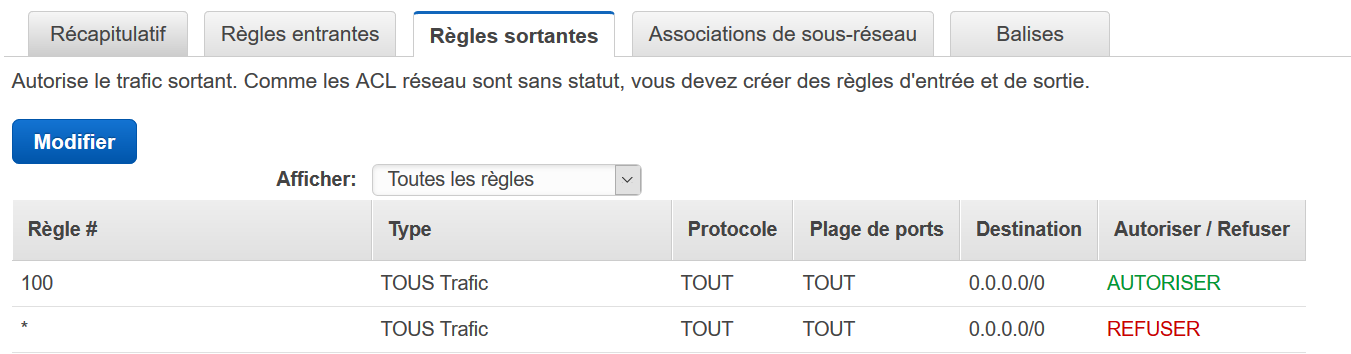


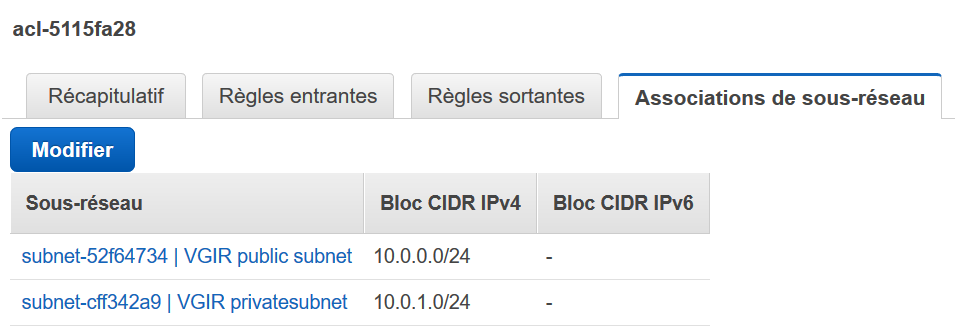
ACL network

It has been created by the wizard :



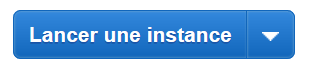






## Bastion creation

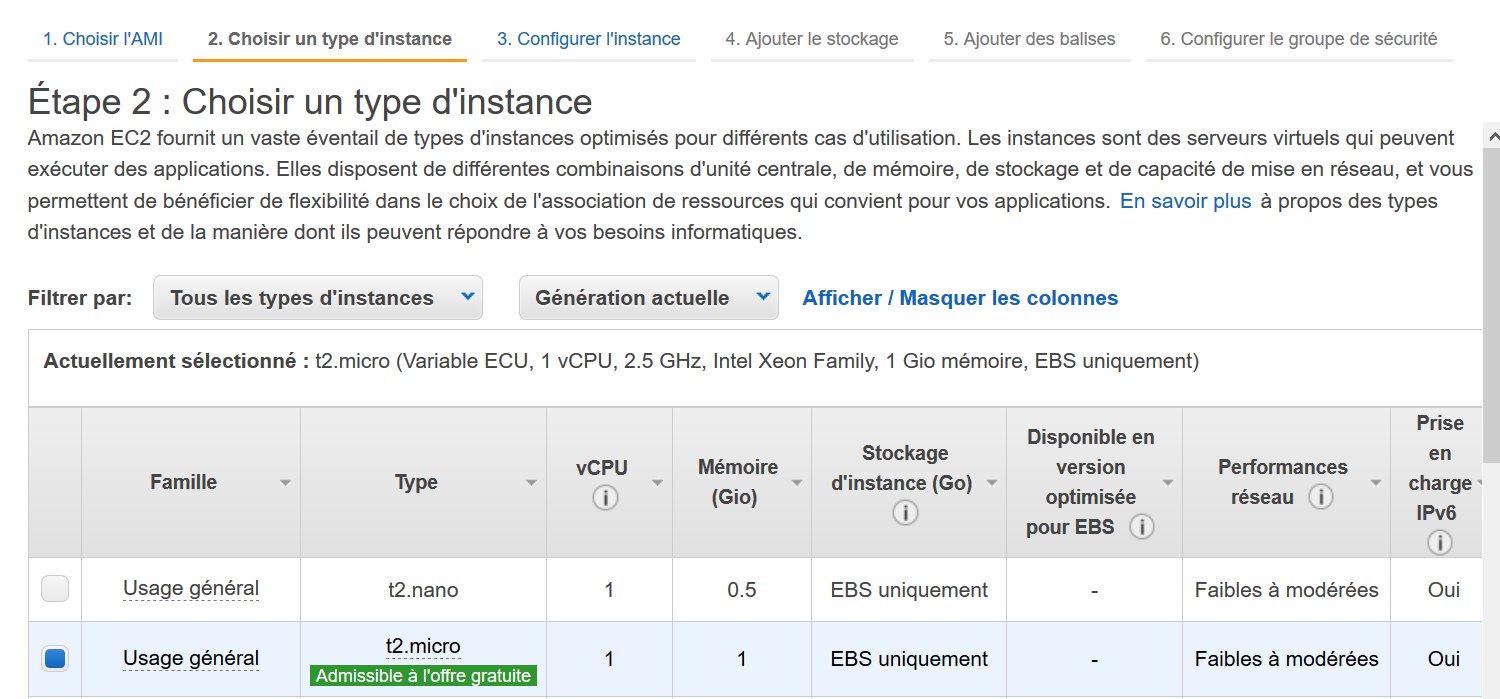
Configuration : EC2 > Dashboard > launch an instance



Choose a linux AMI :



Choose your instance type :



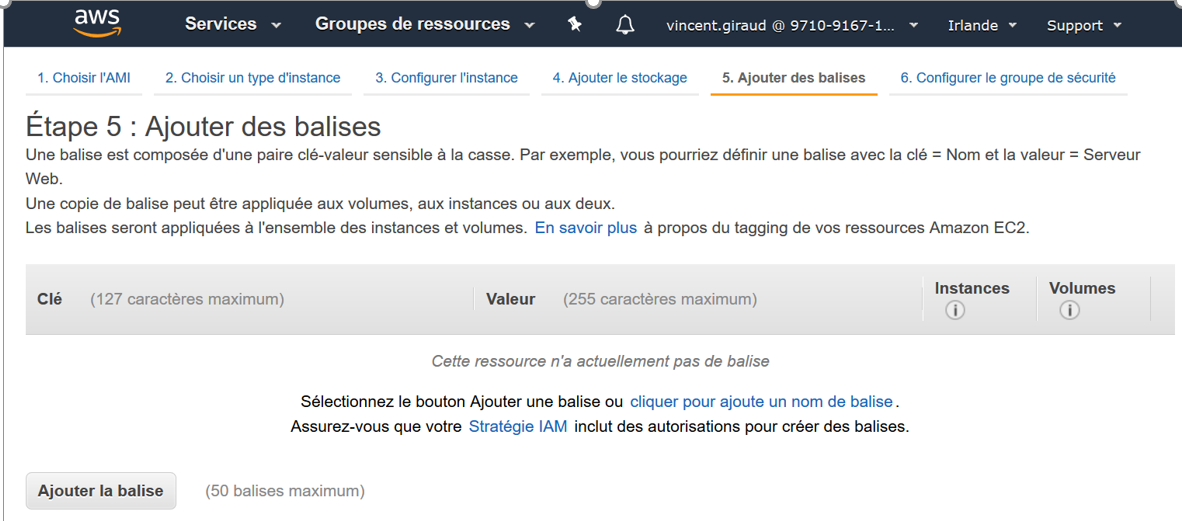
Keep storage as is :



For better visibility of the instance, select :

Key : Name

Name : Your own description



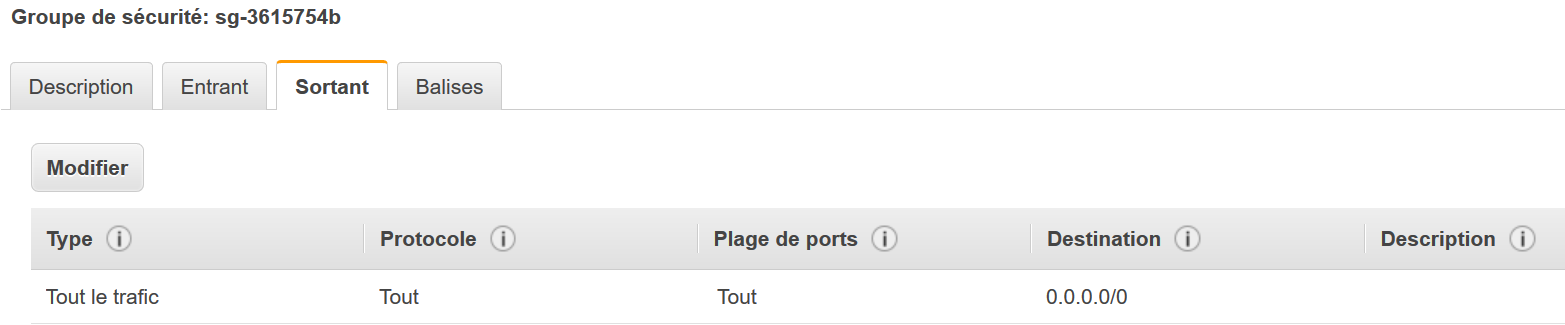
Rename the security group and its description :



As inbound we authorize only SSH, because we will  use the bastion for entering into private instance  with putty:



We are maybe too confident, and trust him (and by ricochet all the aws network) :



We attach the second and last IP to the bastion :

Configuration : EC2 > elastic IP > action > allocate

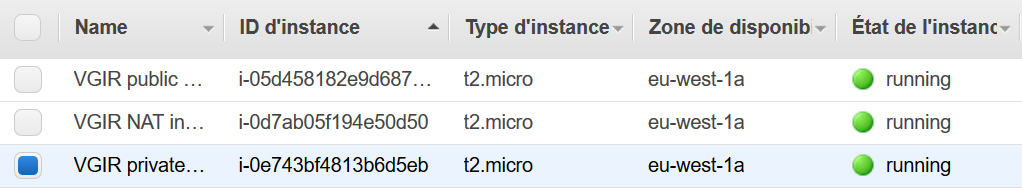
It’s IP is [52.208.163.155](https://eu-west-1.console.aws.amazon.com/ec2/v2/home?region=eu-west-1#Addresses:search=52.208.163.155;sort=publicIp)



# Test of the ping to google.com

## 0 *manual* Check isolated instance has no internet IP

All instances are running :



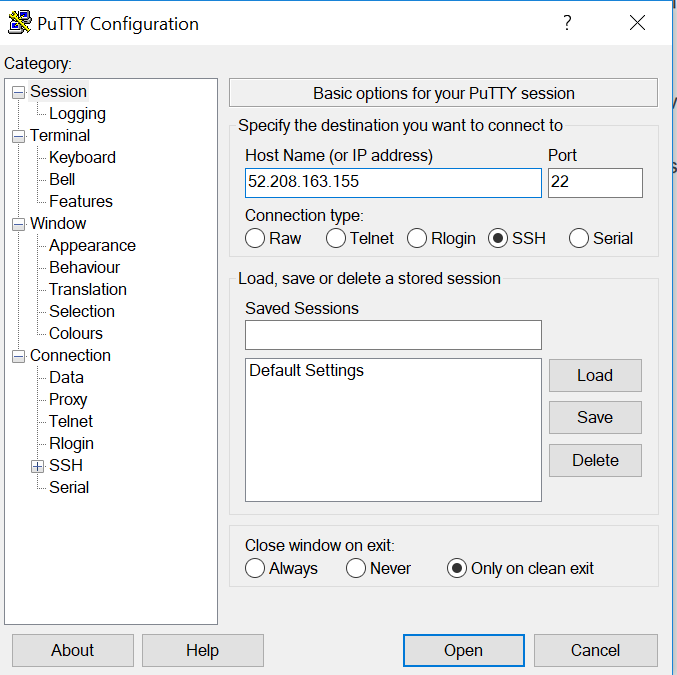
Isolated instance has no internet IP :



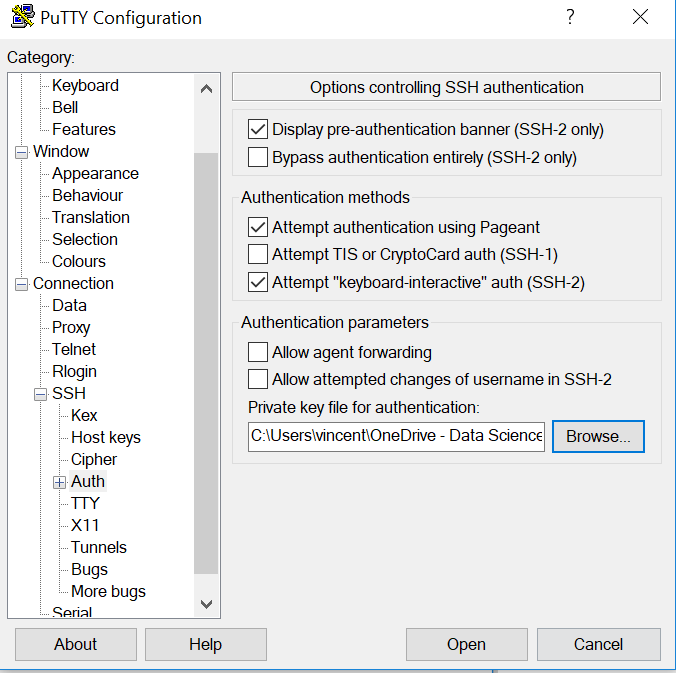
## 1 *manual* Connect to bastion through SSH

Run Putty

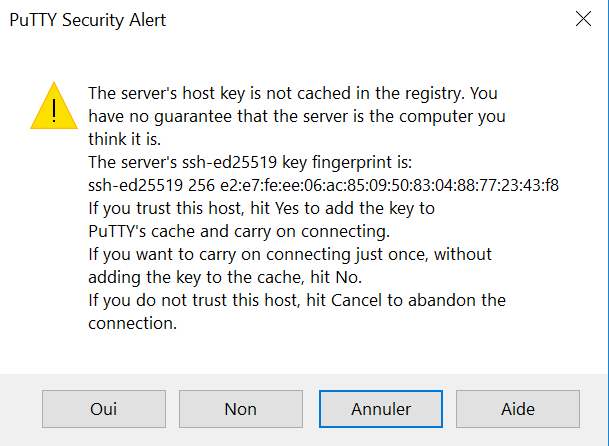
IP address is bastion’s :



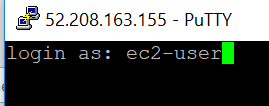
For security , add the pair keys :



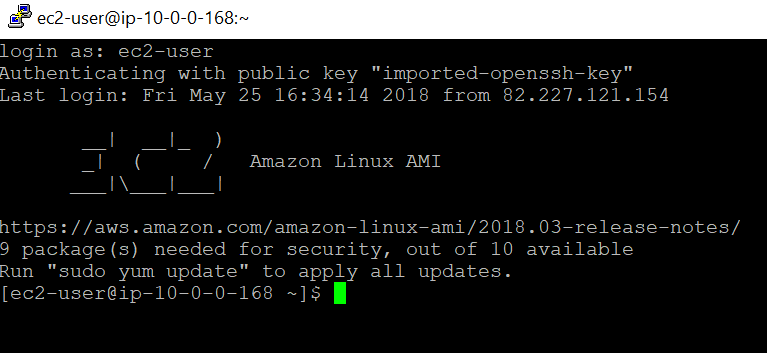
Message is ok :



For this AMI linux, the user is ec2-user :



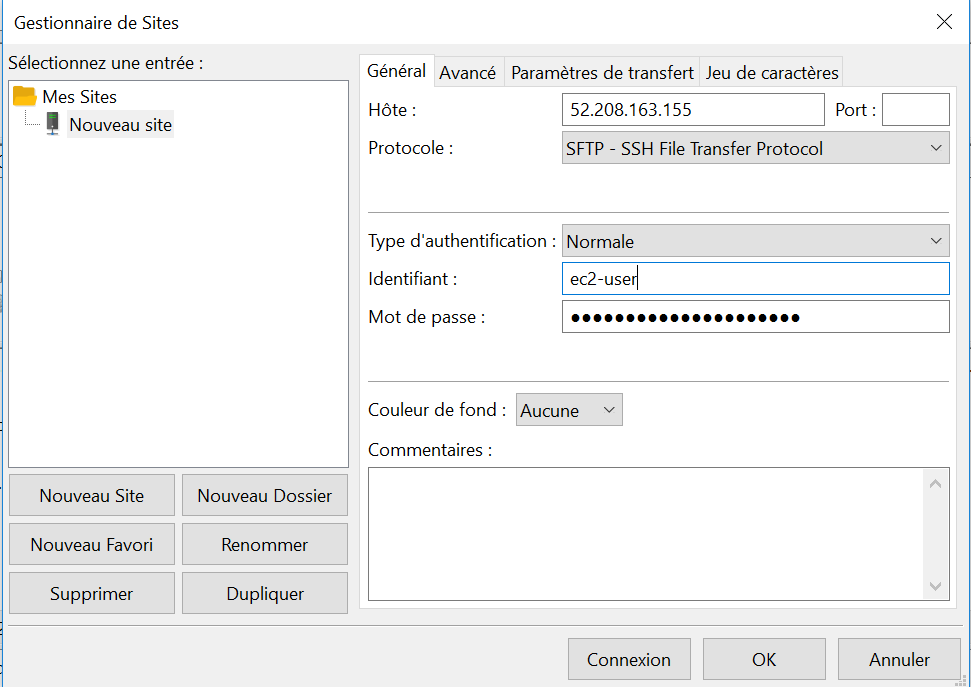
You’re in bastion:



## 2 *manual* Connect to isolated instance through bastion

At the very first time, add with filezille with SFTP protocol the pair key in order to grant  access :

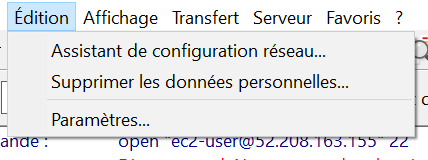
Here we have choose same key between all instances, it simplifies when creating a connection with : 

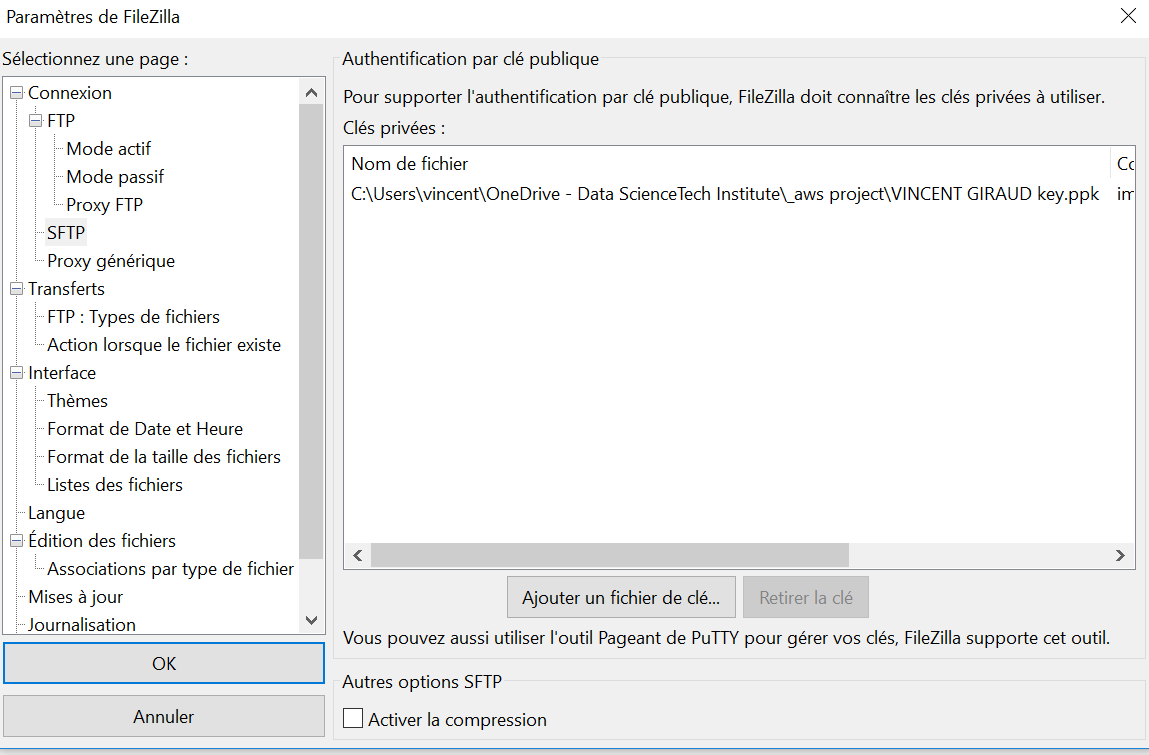


For classic unix

Bastion IP

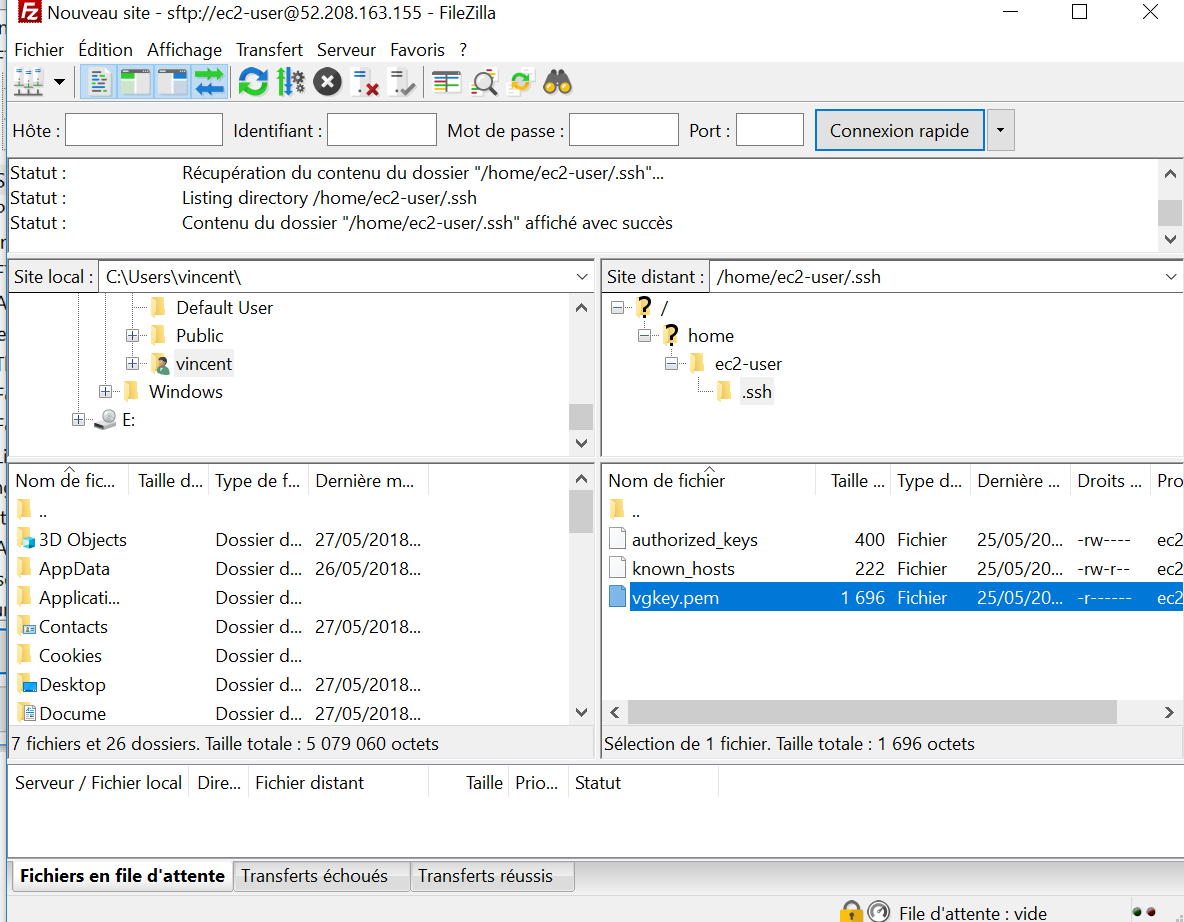
We need to place the key pair too :





The .ppk key

Drop the key inside the bastion, the one in .pem format, as this is in AWS area : here we have chosen /home/ec2-user/.ssh :



In console, remember we are still in bastion, activate the key and SSH-connect to the isolated instance with the 3 command lines :

*Go to the right directory*

cd /home/ec2-user/.ssh

*With authorization management, activate the right key*

chmod 400 vgkey.pem

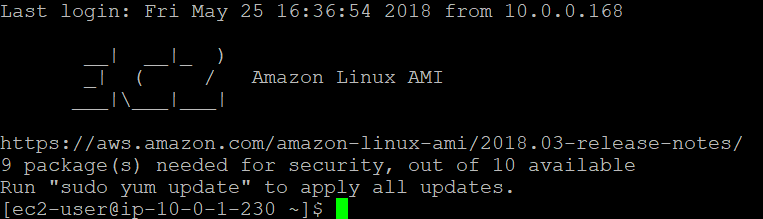
file name of key

*SSH-coonnect*

ssh -i vgkey.pem ec2-user@ 10.0.1.230

file name of key / linux user of isolated instance / network IP of isolated instance

And now you’re in the isolated instance :



## 3 *manual* ping google.com inside isolated instance



## 4 *automatic* : isolated instance call NAT

Remember it’s automatic

## 5 : *automatic*  NAT pings google.com, replacing the network IP with an internet IP

Remember it’s automatic

## 6 : *automatic*  google.com answers to the call to NAT with the internet IP

Remember it’s automatic

## 7 : *automatic*  NAT transfers google.com answers to isolated instance routing the IP

Remember it’s automatic

## 8 : *manual* User can check the call has been returned : no time to leave !

Google.com is answering to you !

