

EXAMEN DE CLOUD COMPUTING

CLASS: 5 anne
FILIÈRE: IRST
MATRICULE: 950 2367

Mettre en place une architecture sécurisée type “bastion” dans le cloud AWS pour superviser des instances privées, en garantissant que seul un poste local (VM VirtualBox) peut y accéder.

1. Générer une paire de clés RSA (AWS-compatible)

AWS accepte les clés **RSA (2048 bits ou plus)** pour les connexions SSH EC2.

Dans le terminal de ta VM Ubuntu :

bash

CopyEdit

```
ssh-keygen -t rsa -b 2048
```

- `-t rsa` : spécifie RSA comme type de clé.
- `-b 2048` : définit la longueur à 2048 bits (minimum accepté par AWS).

Tu peux appuyer sur **Entrée** pour accepter le chemin par défaut, ou entrer un nom personnalisé comme :

```

file ...
ssh-keygen -Q [-l] -f krl_file [file ...]
ssh-keygen -Y find-principals -s signature_file -f allowed_signers_file
ssh-keygen -Y match-principals -I signer_identity -f allowed_signers_file
ssh-keygen -Y check-novalidate -n namespace -s signature_file
ssh-keygen -Y sign -f key_file -n namespace file [-O option] ...
ssh-keygen -Y verify -f allowed_signers_file -I signer_identity
-n namespace -s signature_file [-r krl_file] [-O option]
adngxenuiscore@adngxenuiscore:~$ sudo /home/adngxenuiscore/.ssh/aws_bastion_rsa
sudo: /home/adngxenuiscore/.ssh/aws_bastion_rsa: command not found
adngxenuiscore@adngxenuiscore:~$ sudo ssh-keygen -t rsa -b 2048 -c
Enter file in which the key is (/root/.ssh/id_rsa): /home/adngxenuiscore/.ssh/aws_bastion_rsa
/home/adngxenuiscore/.ssh/aws_bastion_rsa: No such file or directory
adngxenuiscore@adngxenuiscore:~$ sudo ssh-keygen -t rsa -b 2048 -c
Enter file in which the key is (/root/.ssh/id_rsa):
/root/.ssh/id_rsa: No such file or directory
adngxenuiscore@adngxenuiscore:~$ sudo ssh-keygen -t rsa -b 2048
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:jrcMvT0lP15TGw7Vp2TX4PYIqkIFyfvT5KHxdvbybVYM root@adngxenuiscore
The key's randomart image is:
----[RSA 2048]-----
...
o.      o ..
..      o * +
..o oo +.B
...+SB00oEO.o
o+ooo+.+=oo o
o.o.o.ooo.o. .
..o .o. . .o
..      ... o.
-----[SHA256]-----
adngxenuiscore@adngxenuiscore:~$ _

```

a) Test ping DNS

bash

CopyEdit

```
ping -c 4 aws.amazon.com
```

- Vérifie que la VM peut résoudre un nom de domaine et atteindre AWS.
- -c 4 envoie seulement 4 paquets.

b) Test HTTP avec curl

bash

CopyEdit

```
curl -I https://aws.amazon.com
```

- Vérifie que HTTPS fonctionne depuis la VM.
- L'option **-I** affiche uniquement les en-têtes (ex. : **HTTP/2 200**).

```

The key's randomart image is:
+---[RSA 2048]-----+
|      .      |
|      o      |
|      o * +   |
|      .. oo +.B |
|      ...+SBoooEo.o |
|      o+oo+..+oo o |
|      .o.o.ooo.o.  |
|      .o  .o.  . .o |
|      ..   ..   o.  |
+---[SHA256]-----+
adxgenuiscore@adxgenuiscore:~$ ping -c 4 aws.amazon.com
PING dr49lmg3nin2s.cloudfront.net (3.162.38.78) 56(84) bytes of data:
64 bytes from server-3-162-38-78.cdg52.r.cloudfront.net (3.162.38.78): icmp_seq=1 ttl=242 time=107 ms
64 bytes from server-3-162-38-78.cdg52.r.cloudfront.net (3.162.38.78): icmp_seq=2 ttl=242 time=114 ms
64 bytes from server-3-162-38-78.cdg52.r.cloudfront.net (3.162.38.78): icmp_seq=3 ttl=242 time=108 ms
64 bytes from server-3-162-38-78.cdg52.r.cloudfront.net (3.162.38.78): icmp_seq=4 ttl=242 time=106 ms

--- dr49lmg3nin2s.cloudfront.net ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 106.331/109.102/114.196/3.032 ms
adxgenuiscore@adxgenuiscore:~$

--- dr49lmg3nin2s.cloudfront.net ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 106.331/109.102/114.196/3.032 ms
adxgenuiscore@adxgenuiscore:~$ curl -I https://aws.amazon.com
HTTP/2 200
content-type: text/html; charset=UTF-8
date: Thu, 08 May 2025 01:13:01 GMT
x-content-type-options: nosniff
server: Server
x-frame-options: SAMEORIGIN
x-xss-protection: 1; mode=block
strict-transport-security: max-age=47304000; includeSubDomains
x-amz-id-1: 512068817E1B4207BF63
cache-control: no-store, no-cache, must-revalidate
last-modified: Wed, 07 May 2025 12:02:20 GMT
vary: accept-encoding
set-cookie: aws-priv=eyJ2IjoxLjIldSI6MCwic3Q1OjB9; Version=1; Comment="Anonymous cookie for privacy regulations"; Domain=aws.amazon.com; Max-Age=31536000; Exp.
res=Fri, 08 May 2026 01:13:01 GMT; Path=/; Secure
set-cookie: aws_lang=en; Domain=amazon.com; Path=/
x-cache: Miss from cloudfront
via: 1.1 9993b6cb797df66e002c875e3bce4148.cloudfront.net (CloudFront)
x-amz-cf-pop: CDG52-P6
x-amz-cf-id: d7NI0M8BFbH2IHCNxmDRpa9W19YQIe7H_nLxc24Wvic4737sYgQ40u==

```

c) Test DNS directement

bash

CopyEdit

nslookup ec2.amazonaws.com

- Vérifie que le nom de domaine des services EC2 est bien résolu par un serveur DNS.

Si un test échoue :

- Vérifie dans VirtualBox que la VM est en mode **NAT** ou **Bridged**

```

adxgenuiscore@adxgenuiscore:~$ sudo nslookup ec2.amazonaws.com
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   ec2.amazonaws.com
Address: 209.54.180.217

adxgenuiscore@adxgenuiscore:~$

```

- Teste aussi **ip a** pour voir si l'interface a une IP correcte.

4. Préparons les scripts de supervision à exécuter depuis la bastion

But : préparer un script Bash que tu exécutes depuis la bastion pour superviser les EC2 privées.

Crée le fichier :

nano [supervise.sh](#)

```

GNU nano 7.2
#!/bin/bash

# Liste des IPs privées des instances EC2
HOSTS=("172.31.0.0" "172.31.255.255")

#Boucle de supervision
for host in "${HOSTS[@]}"; do
    echo "==== Supervision de $host ====="
    ssh -tt ubuntu@$host << EOF
        echo "[Uptime]"
        uptime
        echo

        echo "[Utilisations du disque]"
        df -h
        echo

        echo "[Ports ouverts]"
        netstat -tulnp | grep LISTEN
        echo "=====
EOF
done

```

2. Mise en place sur AWS

Étape 1 : Créer un VPC personnalisé

1. Va dans **VPC > Your VPCs > Create VPC**
2. Choisis **VPC only**
3. Paramètres :

- Nom : **bastion-vpc**
 - IPv4 CIDR : **10.0.0.0/16**
 - Laisse IPv6 désactivé
 - DNS hostnames : **Activé**
4. Clique sur **Create VPC**

Vos VPC (1/1) Infos

Rechercher des VPC par attribut ou par balise

<input checked="" type="checkbox"/>	Name	ID de VPC	État	Bloquer l'accès public	CIDR IPv4	CIDR IPv6
<input checked="" type="checkbox"/>	bastion-vpc	vpc-0f647fed4c26ad41e	Available	Désactivé	172.31.0.0/16	-

vpc-0f647fed4c26ad41e / bastion-vpc

Détails | Mappage des ressources | CIDR | Journaux de flux | Balises | Intégrations

Détails

ID de VPC vpc-0f647fed4c26ad41e	État Available	Bloquer l'accès public Désactivé	Noms d'hôte DNS Activé
Résolution DNS Activé	Location default	Jeu d'options DHCP dopt-0f717fb47b5277cec	Table de routage principale rtb-060642c48c094cd9b
ACL réseau principal acl-0537b0a07552c09a8	VPC par défaut Oui	CIDR IPv4 172.31.0.0/16	Groupe IPv6

Étape 2 : Créer 2 sous-réseaux

a) Subnet public (pour la bastion)

1. Va dans **Subnets > Create subnet**
2. Sélectionne le VPC **bastion-vpc**
3. Nom : **public-subnet**
4. AZ : ex **eu-west-3a**
5. CIDR : **10.0.1.0/24**
6. Crée le subnet

b) Subnet privé (pour les 2 EC2 privées)

1. Nom : **private-subnet**
2. AZ : même ou autre que le public
3. CIDR : **10.0.2.0/24**
4. Crée le subnet

Tableau de bord du VPC

EC2 Global View

Filtrer par VPC

▼ Cloud privé virtuel

- Vos VPC
- Sous-réseaux
- Tables de routage
- Passerelles Internet
- Passerelles Internet de sortie uniquement
- Passerelles de l'opérateur
- Jeux d'options DHCP
- Adresses IP Elastic
- Listes de préfixes gérées
- Passerelles NAT
- Connexions d'appairage
- Serveurs de routage [Nouveau](#)

▼ Sécurité

- ACL réseau
- Groupes de sécurité

▼ PrivateLink et Lattice

- Mise en route [Mis à jour](#)

Sous-réseaux (1/3) Infos

Rechercher des sous-réseaux par attribut ou par balise

<input checked="" type="checkbox"/>	Name	ID de sous-réseau	État	VPC	Bloquer l'accès public
<input checked="" type="checkbox"/>	-	subnet-0c2df59f25cad3598	Available	vpc-0f647fed4c26ad41e basti...	Désactivé
<input type="checkbox"/>	Subnet public	subnet-0b190152b86242703	Available	vpc-0f647fed4c26ad41e basti...	Désactivé
<input type="checkbox"/>	public-private	subnet-08ddddd701b4d5e0	Available	vpc-0f647fed4c26ad41e basti...	Désactivé

subnets

subnet-0c2df59f25cad3598

Détails | Journaux de flux | Table de routage | ACL réseau | Réservations CIDR | Partage | Balises

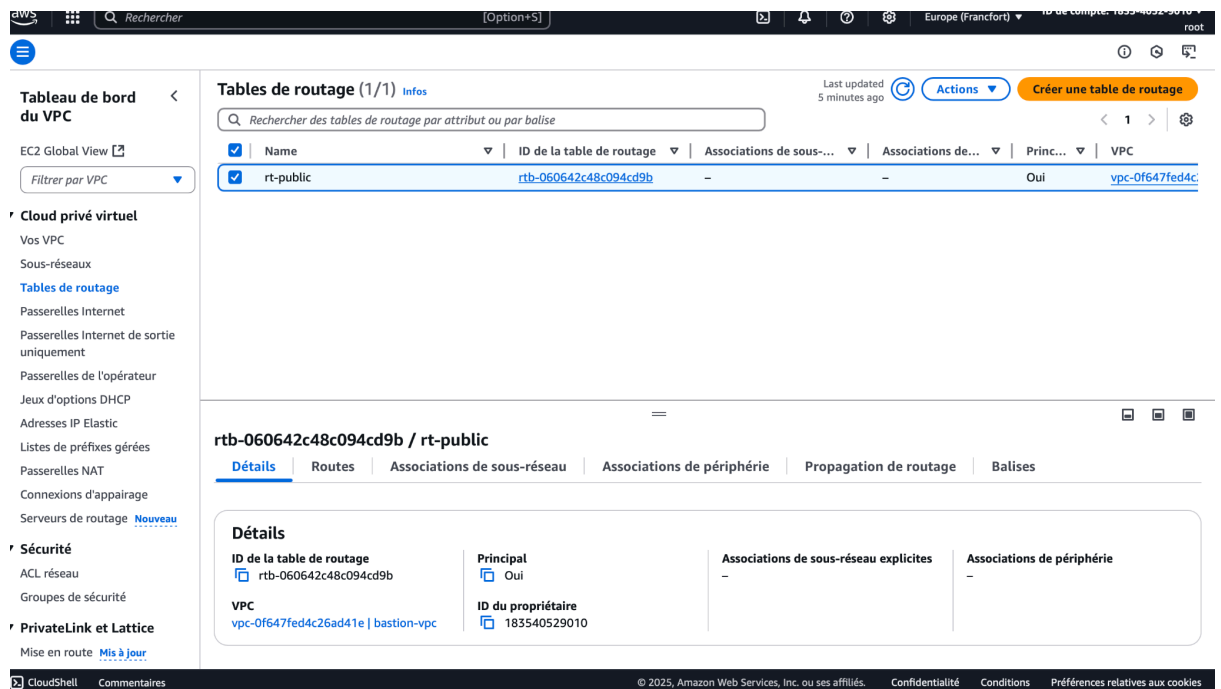
Détails

ID de sous-réseau subnet-0c2df59f25cad3598	ARN du sous-réseau arn:aws:ec2:eu-central-1:183540529010:subnet/subnet-0c2df59f25cad3598	État Available	Bloquer l'accès public Désactivé
CIDR IPv4 172.31.32.0/20	Adresses IPv4 disponibles 4091	CIDR IPv6 -	ID d'association CIDR IPv6 -
Zone de disponibilité eu-central-1b		Groupe de bordure réseau eu-central-1	VPC vpc-0f647fed4c26ad41e bastion-vpc

Étape 4 : Table de routage pour le public

1. Va dans **Route Tables > Create**
2. Nom : **rt-public**
3. Associe-la à **bastion-vpc**
4. Édite les routes :
 - Destination : **0.0.0.0/0**
 - Target : **Internet Gateway**
5. Clique sur **Subnet associations > Associer à public-subnet**

Le subnet privé n'a pas de route vers Internet



Étape 7 : Connexion Bastion → EC2 privées

1. Depuis ta VM locale :

bash

CopyEdit

```
ssh -i ~/.ssh/aws_bastion_rsa ubuntu@<IP-PUBLIQUE_BASTION>
```

2. Sur la bastion, connecte-toi aux instances privées :

bash

CopyEdit

```
ssh ubuntu@10.0.2.10
```

```
ssh ubuntu@10.0.2.11
```

Tu dois copier la **même clé publique** sur `/home/ubuntu/.ssh/authorized_keys` dans les EC2 privées si ce n'est pas fait.

Étape 8 : Installer scripts de supervision sur la bastion

Depuis la bastion :

```
bash
```

```
CopyEdit
```

```
nano supervise.sh
```

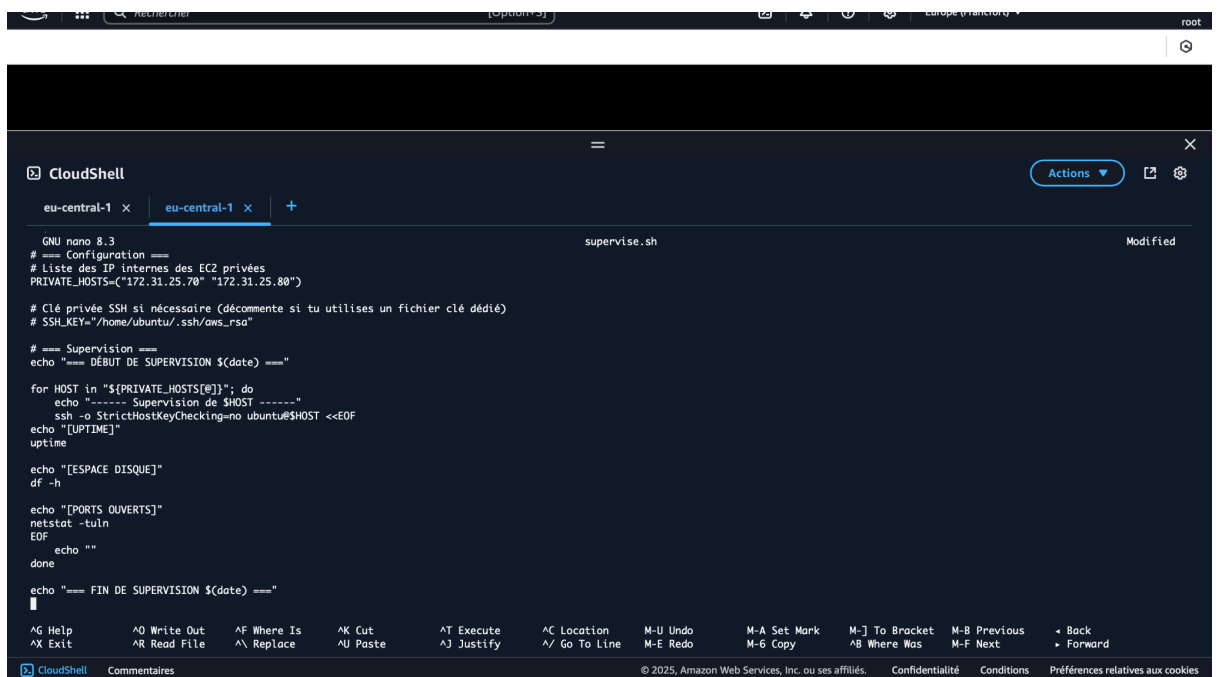
Colle le script (identique à celui que tu as préparé en local), et rends-le exécutable :

```
bash
```

```
CopyEdit
```

```
chmod +x supervise.sh
```

```
./supervise.sh
```



The screenshot shows a CloudShell terminal window with a dark theme. The terminal title is "CloudShell" and it shows two tabs for "eu-central-1". The active tab is displaying the contents of the "supervise.sh" script being edited in nano 8.3. The script includes configuration for private IP addresses, SSH key, and a supervision loop that runs commands like "df -h", "netstat -tuln", and "uptime" on a list of hosts. The bottom of the terminal shows a standard nano editor status bar with various keyboard shortcuts and a footer with "CloudShell Commentaires" and copyright information for Amazon Web Services.

Étape 9 : Activer Fail2ban sur la bastion

```
bash
```

```
CopyEdit
```

```
sudo apt update && sudo apt install -y fail2ban  
  
sudo systemctl enable --now fail2ban
```

Optionnel : éditer `/etc/fail2ban/jail.local` pour ajuster les règles (ex. pour sshd).

Installer Netdata sur les EC2 privées

Depuis la bastion, SSH dans chaque EC2 privée :

bash

CopyEdit

```
bash <(curl -Ss https://my-netdata.io/kickstart.sh)
```

Netdata écoute sur le port 19999 (en local), donc tu peux y accéder en **SSH port forwarding** si nécessaire.

✓ Résumé de l'architecture

text

CopyEdit

[VM Ubuntu Locale]

1

SSH (clé RSA)

1

```
[ Bastion EC2 Publique ] ----> [ EC2 privée 1 ]
```

1

[EC2 privée 2]

1

Scripts de supervision + fail2ban

