


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 UTM) peut y accéder.

Accès sécurisé & supervision centralisée

 **Objectif pédagogique** : 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.**

Travail demandé :

1. **Sur la VM locale (Ubuntu UTM) :**

Générer une paire de clés SSH pour se connecter au cloud.

Tester les connexions réseau depuis la VM vers AWS.

Documenter l’environnement local utilisé (nom VM, IP locale, version Ubuntu).

Préparer les scripts de supervision à exécuter depuis la bastion (ex. uptime, netstat, df -h...).

2. **Sur AWS :**

Créer un VPC avec :

Une instance **EC2 bastion** dans un **subnet public**

Deux instances **EC2 privées** dans un **subnet privé**

Configurer la bastion box pour qu’elle soit **accessible uniquement depuis l’IP de la VM locale** (filtrage IP dans les Security Groups).

Depuis la bastion, accéder aux deux EC2 privées par **SSH interne**.

Installer et lancer des scripts de supervision simple depuis la bastion vers les EC2 privées.

Activer **fail2ban** sur la bastion pour sécuriser les accès.

(Bonus) Installer **Netdata** sur les EC2 privées et les visualiser depuis la bastion.

3. **Automatisation / Sécurité :**

Documenter l’utilisation du **SSH Agent Forwarding** pour chaîner les connexions depuis la VM locale → bastion → EC2 privée.

Créer un script pour lancer automatiquement les vérifications de santé.

Technologies

- VirtualBox + Ubuntu (local)

- VPC, Subnets privés/publics
- EC2 x3
- SSH, Fail2ban, cron
- Scripts de supervision (bash)
- (Bonus) Netdata
- SSH agent forwarding

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 :

```
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]
adxgenuiscore@adxgenuiscore:~$ sudo /home/adxgenuiscore/.ssh/aws_bastion_rsa
sudo: /home/adxgenuiscore/.ssh/aws_bastion_rsa: command not found
adxgenuiscore@adxgenuiscore:~$ sudo ssh-keygen -t rsa -b 2048 -c
Enter file in which the key is (/root/.ssh/id_rsa): /home/adxgenuiscore/.ssh/aws_bastion_rsa
/home/adxgenuiscore/.ssh/aws_bastion_rsa: No such file or directory
adxgenuiscore@adxgenuiscore:~$ 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
adxgenuiscore@adxgenuiscore:~$ 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:jrcMvt0lP15TGw7VpzTX4PYIqkIFyfvT5KHxdvybVYM root@adxgenuiscore
The key's randomart image is:
+----[RSA 2048]-----+
| ..          . |
|  o.         o  |
| ..   o * +   |
| ..o oo +.B   |
| ...+SB000Eo.o|
| o+oo+. =+oo  o|
| .o.o.ooo.o.  |
| .o   .o. . .o|
| ..          o. |
+----[SHA256]-----+
adxgenuiscore@adxgenuiscore:~$
```

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`).

```

advgenuiscore@advgenuiscore:~$ cat /dev/urandom | tr -dc 'a-z0-9' | fold -n 32 | xargs -n 1 shuf -i 0-255 | xargs -n 1 printf '%02x' | fold -n 16 | xargs -n 16 tr -d '\n'
The key's randormart image is:
----[RSA 2048]-----
...
0. 0 ..
.. 0 * +
..0 00 +.B
...+SB000E0.0
0+00+.=+00 0|
..0.0.000.0. .
..0 .0. . .0
.. .. 0.
+----[SHA256]-----
advgenuiscore@advgenuiscore:~$ ping -c 4 aws.amazon.com
PING dr49lmg3n1n2s.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

--- dr49lmg3n1n2s.cloudfront.net ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 106.931/109.102/114.196/3.032 ms
advgenuiscore@advgenuiscore:~$

--- dr49lmg3n1n2s.cloudfront.net ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 106.931/109.102/114.196/3.032 ms
advgenuiscore@advgenuiscore:~$ 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: 512060B17E1B42070F63
cache-control: no-store, no-cache, must-revalidate
last-modified: Wed, 07 May 2025 12:02:20 GMT
vary: accept-encoding
set-cookie: aws_privacy=eyJ2IjoxLCJlIjdsIGRlcwUic9Q0IjBj; Version=1; Comments="Anonymous cookie for privacy regulations"; Domain=aws.amazon.com; Max-Age=31536000; Exp=Fri, 08 May 2026 01:13:01 GMT; Path=/; Secure
set-cookie: aws_lang=en; Domain=aws.amazon.com; Path=/
x-cache: Miss from cloudfront
via: 1.1 9993b6cb797df66ee02c875e3bce4148.cloudfront.net (CloudFront)
x-amz-cf-pop: CDG52-P6
x-amz-cf-id: d7N1QMBbFbhZ1HCNxm0Roa9W19YQIe7H_nIxc24H4vic4737sYvQW0u==

```

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 supervise.sh
#!/bin/bash

echo "==== Supervision de $(hostname) - $(date) ====="
echo

echo "=== Uptime ==="
uptime
echo

echo "=== Utilisation CPU (1 sec) ==="
top -bn1 | grep "Cpu(s)"
echo

echo "=== Utilisation memoire ==="
free -h
echo

echo "=== Espace disque monte ==="
df -h --total
echo

echo "=== Connexions reseau actives (ports ecoutes) ==="
ss -tuln
echo

echo "=== Nombre de connexions etablies ==="
ss _s | grep estab
echo

echo "=== Utilisateurs connectes ==="
who
echo

echo "=== Services actifs ==="
systemctl list-units --type=services --state=running | head -15
echo

echo "=== Derniers journaux systeme ( 5 ligne) ==="
journalctl -n 5 --no-pager
echo

echo "=== Fin de supervision ==="
```

```
Mem:          total        used        free        shared  buff/cache   available
Swap:         3.8Gi        696Mi        2.7Gi        5.3Mi        578Mi        3.1Gi

=== Espace disque monte ===
Filesystem      Size      Used Avail Use% Mounted on
tmpfs            391M    1.4M   389M   1% /run
efivarfs         256K    48K   209K  19% /sys/firmware/efi/efivars
/dev/mapper/ubuntu--vg-ubuntu--lv  30G    7.9G   21G  26% /
tmpfs            2.0G     0     2.0G   0% /dev/shm
tmpfs            5.0M     0     5.0M   0% /run/lock
/dev/vda2        2.0G   191M   1.6G  11% /boot
/dev/vda1        1.1G    6.4M   1.1G   1% /boot/efi
tmpfs            391M    12K   391M   1% /run/user/1000
total            36G    8.0G   26G  24% -

=== Connexions reseau actives (ports ecoutes) ===
Netid      State      Recv-Q      Send-Q      Local Address:Port      Peer Address:Port      Process
udp        UNCONN     0            0            127.0.0.54:53            0.0.0.0:*
udp        UNCONN     0            0            127.0.0.53:10:53        0.0.0.0:*
udp        UNCONN     0            0            192.168.64.4:enp0s1:60  0.0.0.0:*
udp        UNCONN     0            0            [fe80::8476:41ff:fe6f:8f0]:enp0s1:546  [::]:*
udp        UNCONN     0            0            [fe80::d2:21ff:fe76:335a]:enp0s2:546  [::]:*
tcp        LISTEN     0            4096         127.0.0.54:53            0.0.0.0:*
tcp        LISTEN     0            4096         127.0.0.53:10:53        0.0.0.0:*
tcp        LISTEN     0            151         0.0.0.0:3306             0.0.0.0:*
tcp        LISTEN     0            70          127.0.0.1:33060          0.0.0.0:*
tcp        LISTEN     0            4096         *:22                     *:22
tcp        LISTEN     0            511         *:80                     *:80

=== Nombre de connexions etablies ===
Error: an inet prefix is expected rather than "_s".
Cannot parse dst/src address.

=== Utilisateurs connectes ===
adxgeniuscore tty1          2025-05-08 22:58

=== Services actifs ===
Unknown unit type or load state 'services'.
Use -t help to see a list of allowed values.

=== Derniers journaux systeme ( 5 ligne) ===
May 09 10:14:56 adxgeniuscore sudo[2373]: pam_unix(sudo:session): session opened for user root(uid=0) by adxgeniuscore(uid=1000)
May 09 10:15:01 adxgeniuscore CRON[2376]: pam_unix(cron:session): session opened for user root(uid=0) by root(uid=0)
May 09 10:15:01 adxgeniuscore CRON[2377]: (root) CMD (command -v debian-sa1 > /dev/null && debian-sa1 1 1)
May 09 10:15:01 adxgeniuscore CRON[2376]: pam_unix(cron:session): session closed for user root
May 09 10:15:47 adxgeniuscore sudo[2373]: pam_unix(sudo:session): session closed for user root

=== Fin de supervision ===
adxgeniuscore@adxgeniuscore:~$ ./supervise.sh
```

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**

```
ssh -i "adxgenuiscore.pem"
ubuntu@ec2-18-199-84-28.eu-central-1.compute.amazonaws.com
```

The screenshot shows the AWS Management Console interface for a VPC. At the top, there's a header 'Vos VPC (1/1) Infos' with a search bar and a table of VPCs. The table has columns: Name, ID de VPC, État, Bloquer l'accès public, CIDR IPv4, and CIDR IPv6. One VPC is listed: 'bastion-vpc' with ID 'vpc-0f647fed4c26ad41e', state 'Available', and CIDR '172.31.0.0/16'. Below the table, there's a section for 'vpc-0f647fed4c26ad41e / bastion-vpc' with tabs for 'Détails', 'Mappage des ressources', 'CIDR', 'Journaux de flux', 'Balises', and 'Intégrations'. The 'Détails' tab is active, showing a grid of VPC configuration details.

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-0537b0a075662c0e8	VPC par défaut Oui	CIDR IPv4 172.31.0.0/16	Groupe IPv6

Étape 2 : Créer 2 sous-réseaux

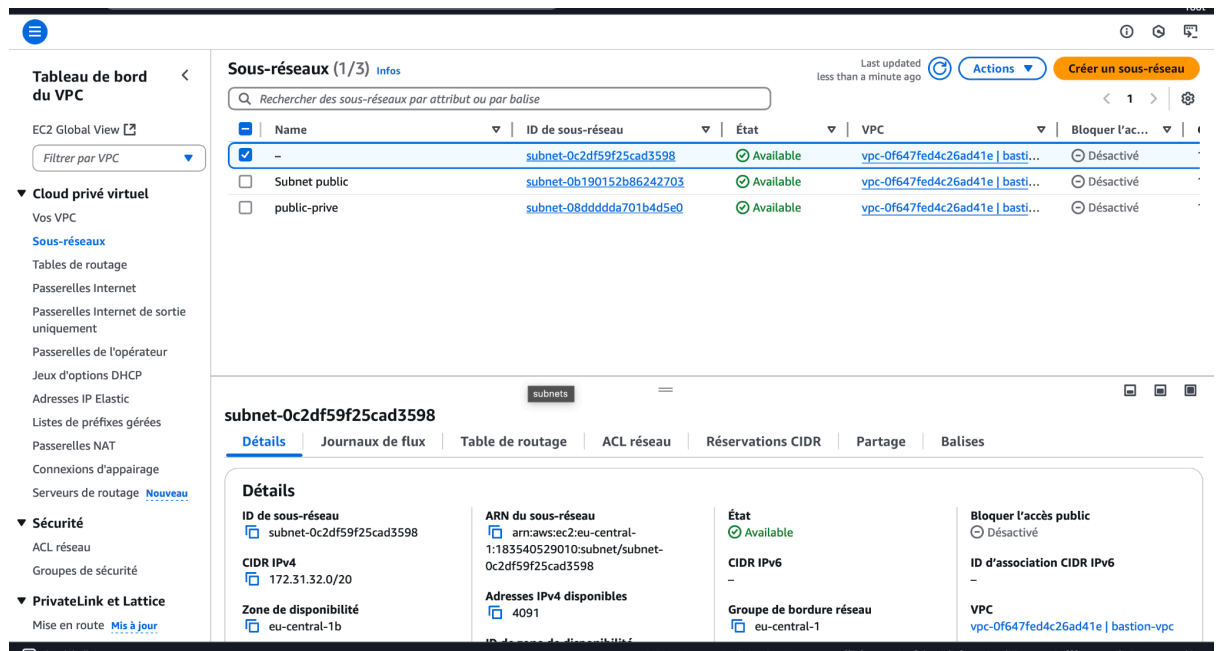
a) Subnet public (pour la bastion)

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



Sous-réseaux (1/3) Infos

Rechercher des sous-réseaux par attribut ou par balise

	Name	ID de sous-réseau	État	VPC	Bloquer l'acc...
<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

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 : l'**Internet Gateway**
5. Clique sur **Subnet associations > Associer à public-subnet**

Le subnet privé n'a pas de route vers Internet

The screenshot shows the AWS Management Console interface. On the left, the 'Tableau de bord du VPC' (VPC Dashboard) is visible, with a sidebar containing links to 'Cloud privé virtuel' (Virtual Private Cloud) and 'Sécurité' (Security). The main content area is titled 'Tables de routage (1/1)' (Route Tables). It features a search bar and a table with columns: Name, ID de la table de routage, Associations de sous-réseau, Associations de périphérie, Principale, and VPC. The table lists one route table, 'rt-public', with ID 'rtb-060642c48c094cd9b'. Below the table, the details for 'rtb-060642c48c094cd9b / rt-public' are displayed, including its ID, VPC, and principal status.

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
```

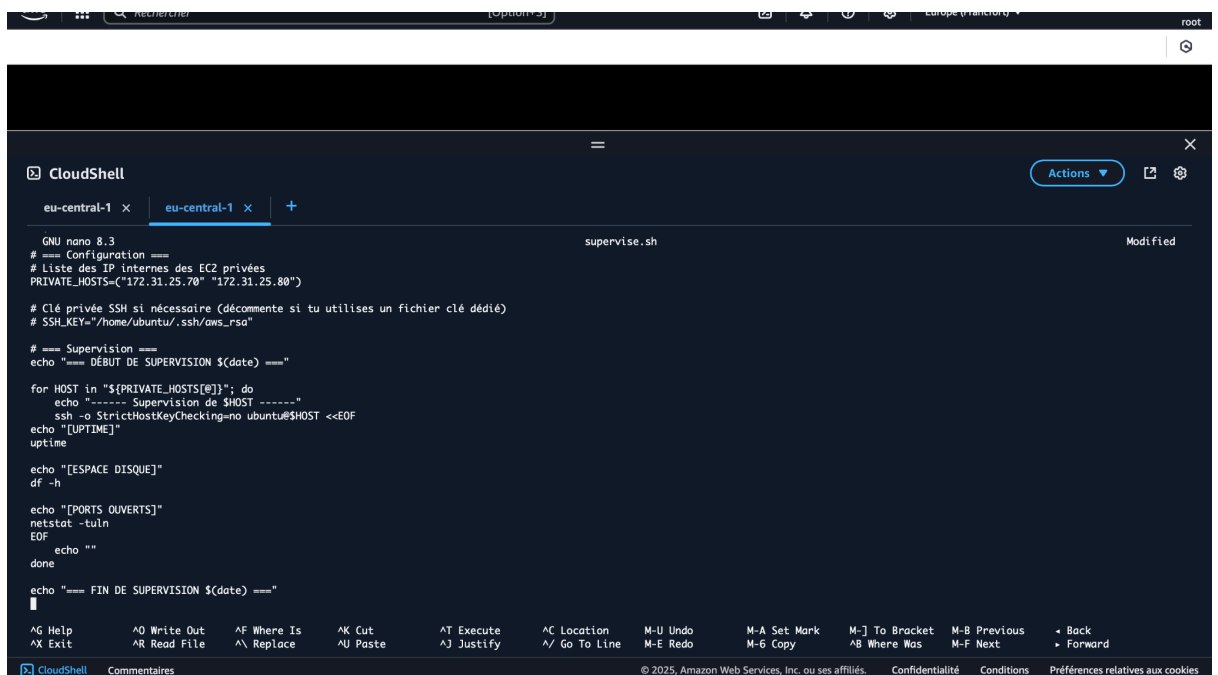
Colle le script (identique à celui que tu as préparé en local), et rends-le exécutable :

```
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 in the nano editor. The script is a shell script that configures supervision for EC2 instances. It includes comments in French and a loop that iterates over a list of private IP addresses (PRIVATE_HOSTS) and performs various checks and actions for each host. The script ends with a "done" statement and a final echo statement. The bottom of the terminal shows a status bar with copyright information for Amazon Web Services, Inc. and links to Confidentialité, Conditions, and Préférences relatives aux cookies.

```
GNU nano 8.3 supervise.sh
# == Configuration ==
# Liste des IP internes des EC2 privées
PRIVATE_HOSTS=("172.31.25.70" "172.31.25.80")

# Clé privée SSH si nécessaire (décommente si tu utilises un fichier clé dédié)
# SSH_KEY="/home/ubuntu/.ssh/aws_rsa"

# == Supervision ==
echo "=== DÉBUT DE SUPERVISION $(date) ==="

for HOST in "${PRIVATE_HOSTS[@]}; do
    echo "----- Supervision de $HOST -----"
    ssh -o StrictHostKeyChecking=no ubuntu@$HOST <<EOF
    echo "[UPTIME]"
    uptime
    echo "[ESPACE DISQUE]"
    df -h
    echo "[PORTS OUVERTS]"
    netstat -tln
    EOF
    echo ""
done

echo "=== FIN DE SUPERVISION $(date) ==="
```

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 ]  
    |  
    SSH (clé RSA)  
    |  
[ Bastion EC2 Publique ] ----> [ EC2 privée 1 ]  
    |                               [ EC2 privée 2 ]  
    |  
    Scripts de supervision + fail2ban
```

2. Sur AWS

✓ Créer le VPC

- CIDR VPC : 10.0.0.0/16
- Subnet public : 10.0.1.0/24
- Subnet privé : 10.0.2.0/24

✓ Lancer les instances EC2

- Bastion (Ubuntu) dans le **subnet public**
- Deux EC2 privées (Ubuntu) dans le **subnet privé**

✓ Sécuriser la Bastion Box (Security Group)

- Port 22 autorisé **uniquement** depuis l'IP publique de la VM locale
 - À trouver avec : `curl ifconfig.me`
- Bloquer tout autre accès non autorisé.

✓ Connexion SSH locale → Bastion

bash

CopyEdit

```
ssh -i ~/.ssh/id_rsa ubuntu@<bastion-public-ip>
```

```
adngxenuiscore.pem home supervise.sh
adngxenuiscore@adngxenuiscore:~$ sudo ssh ubuntu@18.199.84.28
ubuntu@18.199.84.28: Permission denied (publickey).
adngxenuiscore@adngxenuiscore:~$ sudo ssh -i "adngxenuiscore.pem"
Warning: Identity file adngxenuiscore.pem not accessible: No such file or directory.
usage: ssh [-6haCfGgkMNnqsTtVvXxYy] [-B bind_interface] [-b bind_address]
           [-c cipher_spec] [-D [bind_address]:port] [-E log_file]
           [-e escape_char] [-F configfile] [-I pkcs11] [-i identity_file]
           [-J destination] [-L address] [-l login_name] [-m mac_spec]
           [-O ctl_cmd] [-o option] [-P tag] [-p port] [-R address]
           [-S ctl_path] [-W host:port] [-w local_tun[:remote_tun]]
           destination [command [argument ...]]
           ssh [-Q query_option]
adngxenuiscore@adngxenuiscore:~$ sudo ssh -i "adngxenuiscore.pem" ubuntu@18.199.84.28
Warning: Identity file adngxenuiscore.pem not accessible: No such file or directory.
ubuntu@18.199.84.28: Permission denied (publickey).
adngxenuiscore@adngxenuiscore:~$ sudo ssh -i "adngxenuiscore.pem" ubuntu@18.199.84.28
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 5.8.0-1024-aws aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Fri May  9 20:57:19 UTC 2025

System load:  0.0           Temperature:   -273.1 C
Usage of /:   31.9% of 6.71GB Processes:      140
Memory usage: 50%          Users logged in: 0
Swap usage:   0%           IPv4 address for ens5: 172.31.25.79

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.
   https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

87 updates can be applied immediately.
44 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Fri May  9 20:53:13 2025 from 41.73.105.236
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-25-79:~$ _
```

✓ SSH interne Bastion → EC2 privées

- Sur Bastion : placer la **même clé privée** (`id_rsa`) ou transférer une autre autorisée sur les EC2 privées.
- Exemple :

bash

CopyEdit

```
ssh -i ~/.ssh/id_rsa ubuntu@ip
```

✓ Scripts de supervision depuis Bastion vers EC2 privées

Exécuter le script avec :

bash

CopyEdit

```
ssh ubuntu@10.0.2.10 'bash -s' < ./scripts/supervision.sh
```

✓ Activer Fail2Ban sur la Bastion

bash

CopyEdit

```
sudo apt update && sudo apt install fail2ban -y
```

```
sudo systemctl enable fail2ban
```

```
sudo systemctl start fail2ban
```

Netdata sur EC2 privées

Installation :

bash

CopyEdit

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

Visualiser depuis Bastion :

- Se connecter à l'IP privée : <http://10.0.2.10:19999>
- Utiliser un tunnel SSH pour l'accès :

bash

CopyEdit

```
ssh -L 19999:localhost:19999 ubuntu@ip
```

3. Automatisation / Sécurité

✓ SSH Agent Forwarding

Sur ta VM locale :

bash

CopyEdit

```
eval "$(ssh-agent -s)"
```

```
ssh-add ~/.ssh/id_rsa
```

```
ssh -A ubuntu@<bastion-public-ip>
```

```
# Puis depuis Bastion :
```

```
ssh ubuntu@ip
```

✓ Script de supervision automatique (depuis Bastion)

Créer `check_all.sh` :

bash

CopyEdit

```
#!/bin/bash
```

```
for ip in 10.0.2.10 10.0.2.11; do  
    echo "Checking $ip..."  
    ssh ubuntu@$ip 'bash -s' < ./scripts/supervision.sh  
done
```