DOCUMENTATION TECHNIQUE

Installation et Configuration d'un Bastion SSH

Accès SSH sécurisé via Active Directory et menu interactif

Référence	BASTION-SSH-001
Version	1.1
Date	23/03/2026
Catégorie	Sécurité / Administration système
Niveau	BTS SIO / Admin système junior
Auteur	S. Dos Santos

1. Introduction

Un bastion SSH (également appelé jump host ou serveur de rebond) est un équipement intermédiaire qui centralise et sécurise tous les accès SSH à l'infrastructure. Plutôt que de permettre à chaque poste d'accéder directement aux serveurs et équipements réseau, seul le bastion est autorisé à établir ces connexions.

Cette architecture réduit considérablement la surface d'attaque et simplifie la traçabilité des accès : toute connexion SSH transite obligatoirement par le bastion, ce qui facilite la supervision et l'audit.

REMARQUE : Dans ce projet, le bastion SSH est accessible uniquement depuis le VLAN administrateur (VLAN 70). Tous les autres VLANs sont bloqués au niveau du routeur pour les connexions SSH vers les serveurs.

Fonctionnalités mises en œuvre

- Filtrage SSH par ACL Cisco : seul le réseau administrateur peut initier des connexions SSH vers les serveurs
- Menu interactif au lancement de session : affichage d'un menu de sélection à la connexion
- Authentification via Active Directory : connexion avec les comptes du domaine projet.com
- Création automatique du répertoire utilisateur à la première connexion

2. Prérequis

2.1 Prérequis matériels — Machine virtuelle Bastion

Paramètre	Valeur / Description
`Système d'exploitation`	Debian / Ubuntu Server (LTS recommandé)
`RAM`	1 Go minimum (2 Go recommandé)
`Stockage`	10 Go d'espace disque libre
`Processeur`	1 vCPU minimum
`Interface réseau`	Connectée au VLAN 70 (réseau administrateur)

2.2 Prérequis réseau

- VLAN 70 dédié au réseau administrateur, déclaré sur Proxmox et le routeur
- Sous-interface configurée sur le routeur pour le VLAN 70
- Routeur Cisco accessible pour la configuration des ACL
- Contrôleur de domaine Active Directory (domaine : projet.com) joignable depuis le bastion

2.3 Prérequis logiciels

- Proxmox VE installé et opérationnel
- Accès SSH ou console au routeur Cisco
- Accès root ou sudo sur la machine bastion

> ⚠ **ATTENTION :** Le bastion doit impérativement être joignable depuis le VLAN administrateur (VLAN 70) et doit lui-même pouvoir joindre tous les serveurs à administrer. Vérifier le plan d'adressage IP avant de démarrer.

3. Création du réseau administrateur

La première étape consiste à isoler le réseau d'administration dans un VLAN dédié. Ce VLAN (VLAN 70) sera le seul autorisé à initier des connexions SSH vers les serveurs et équipements réseau.

3.1 Configuration de la sous-interface sur le routeur Cisco

1. Se connecter au routeur Cisco en CLI (SSH ou console).
2. Créer la sous-interface correspondant au VLAN 70 et lui affecter l'adresse IP du réseau administrateur.

```
sw1(config)#vlan 70
sw1(config-vlan)#name Admin
sw1(config-vlan)#exit
```

Figure 1 — Création de la sous-interface pour le VLAN administrateur sur le routeur

```
R1(config)#int fastEthernet 0/1.70
```

Figure 2 — Configuration de l'encapsulation dot1Q et de l'adresse IP

```
R1(config-subif)#encapsulation dot1Q 70
R1(config-subif)#ip address 192.168.200.1 255.255.255.0
R1(config-subif)#ip nat
R1(config-subif)#ip nat in
R1(config-subif)#ip nat inside
```

Figure 3 — Vérification de la configuration de la sous-interface

REMARQUE : La sous-interface doit utiliser l'encapsulation 802.1Q (dot1Q) pour correspondre au VLAN tagué côté switch. Vérifier la cohérence entre le numéro de VLAN sur le routeur et celui déclaré sur Proxmox.

4. Création de la carte réseau et de la machine virtuelle

4.1 Déclaration du VLAN sur Proxmox

3. Se connecter à l'interface web de Proxmox.
4. Aller dans Datacenter > Nœud > Network.
5. Créer un nouveau bridge Linux (ou VLAN) associé au VLAN 70.

Figure 4 — Déclaration du VLAN 70 dans Proxmox

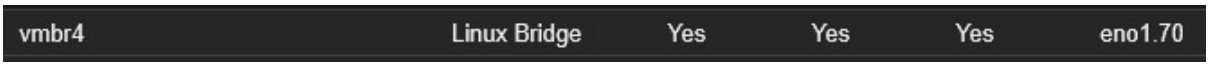

Figure 5 — Configuration du bridge réseau pour le VLAN 70

Figure 6 — Validation de la carte réseau VLAN 70 dans Proxmox

4.2 Création de la machine virtuelle Bastion

6. Dans Proxmox, cliquer sur Create VM.
7. Attribuer un nom à la VM (ex. : Bastion-SSH).
8. Sélectionner l'ISO Debian ou Ubuntu Server.
9. Allouer les ressources (RAM, disque, CPU) conformément aux prérequis.

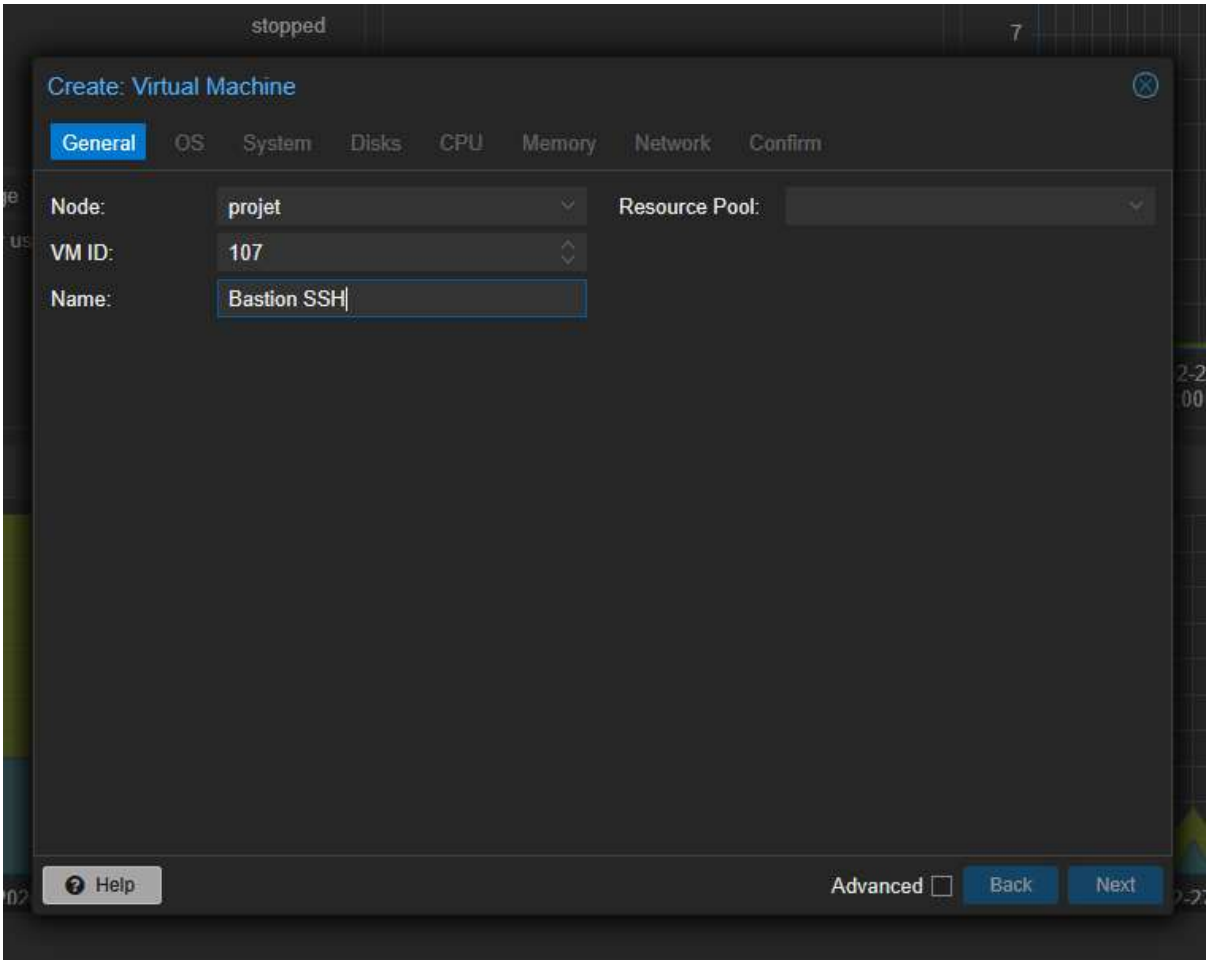

Figure 7 — Paramétrage de la VM Bastion dans Proxmox

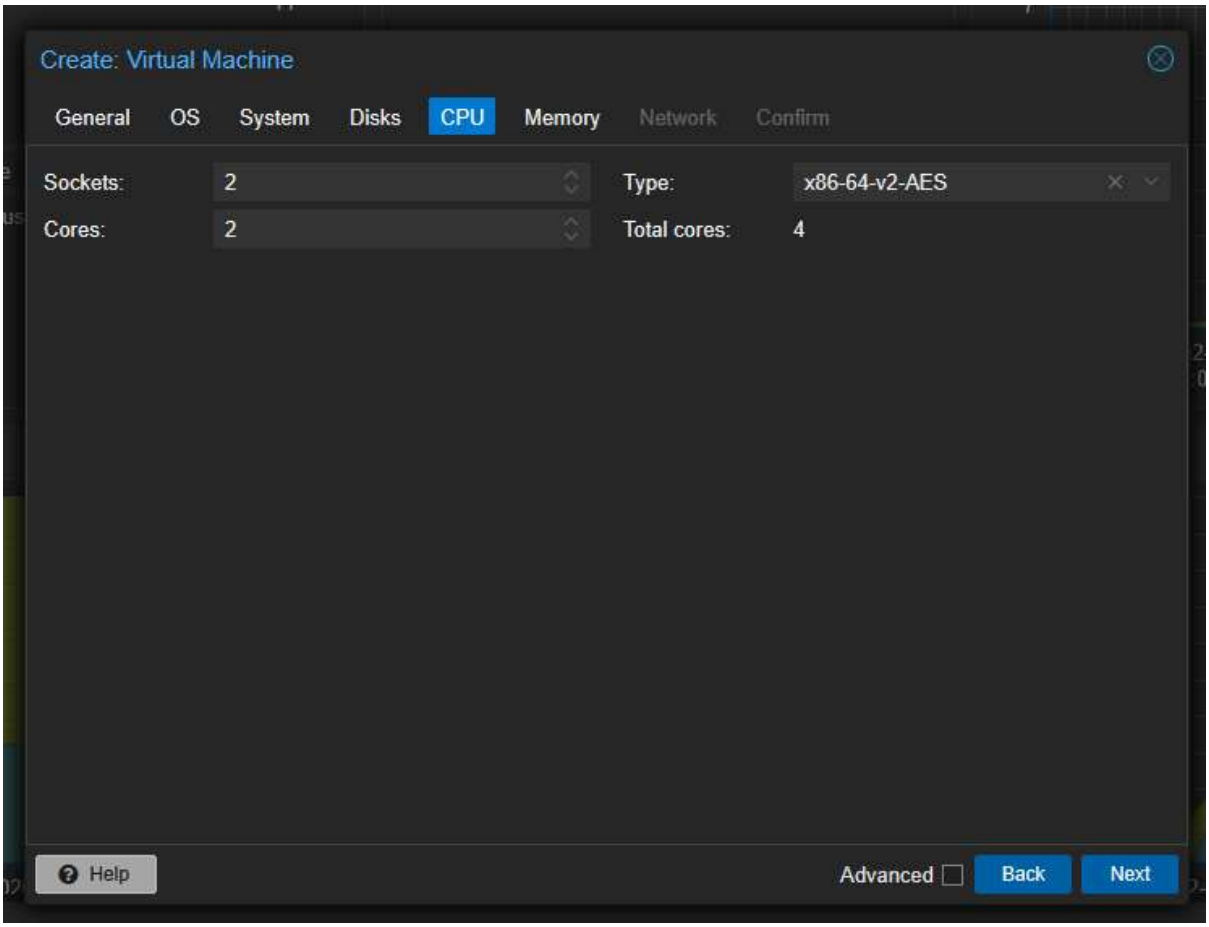

Figure 8 — Configuration des ressources de la VM

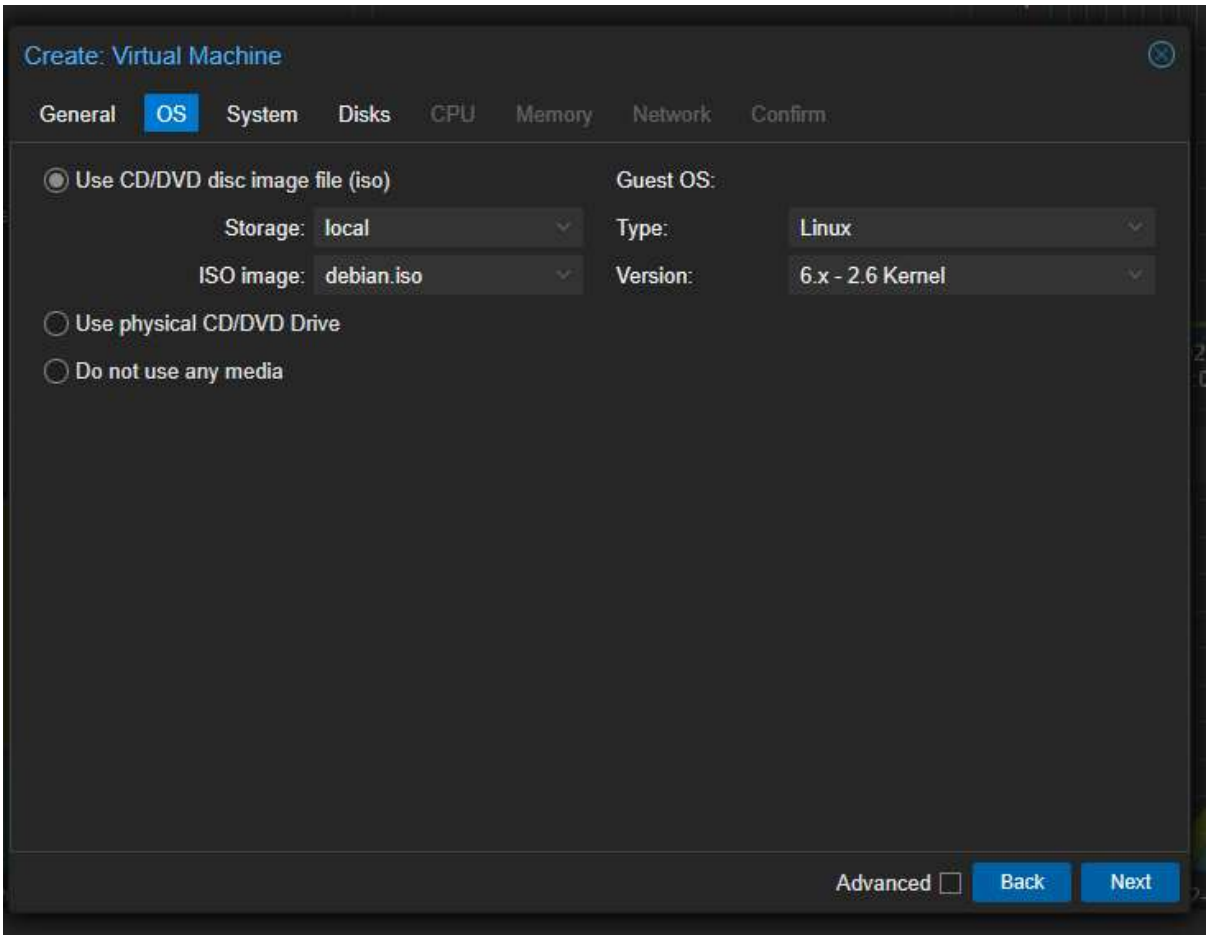

Figure 9 — Suite de la configuration de la VM

4.3 Ajout de la carte réseau VLAN 70

10. Dans les paramètres hardware de la VM, cliquer sur Add > Network Device.
11. Sélectionner le bridge correspondant au VLAN 70 créé précédemment.
12. Valider et démarrer la VM pour procéder à l'installation du système.

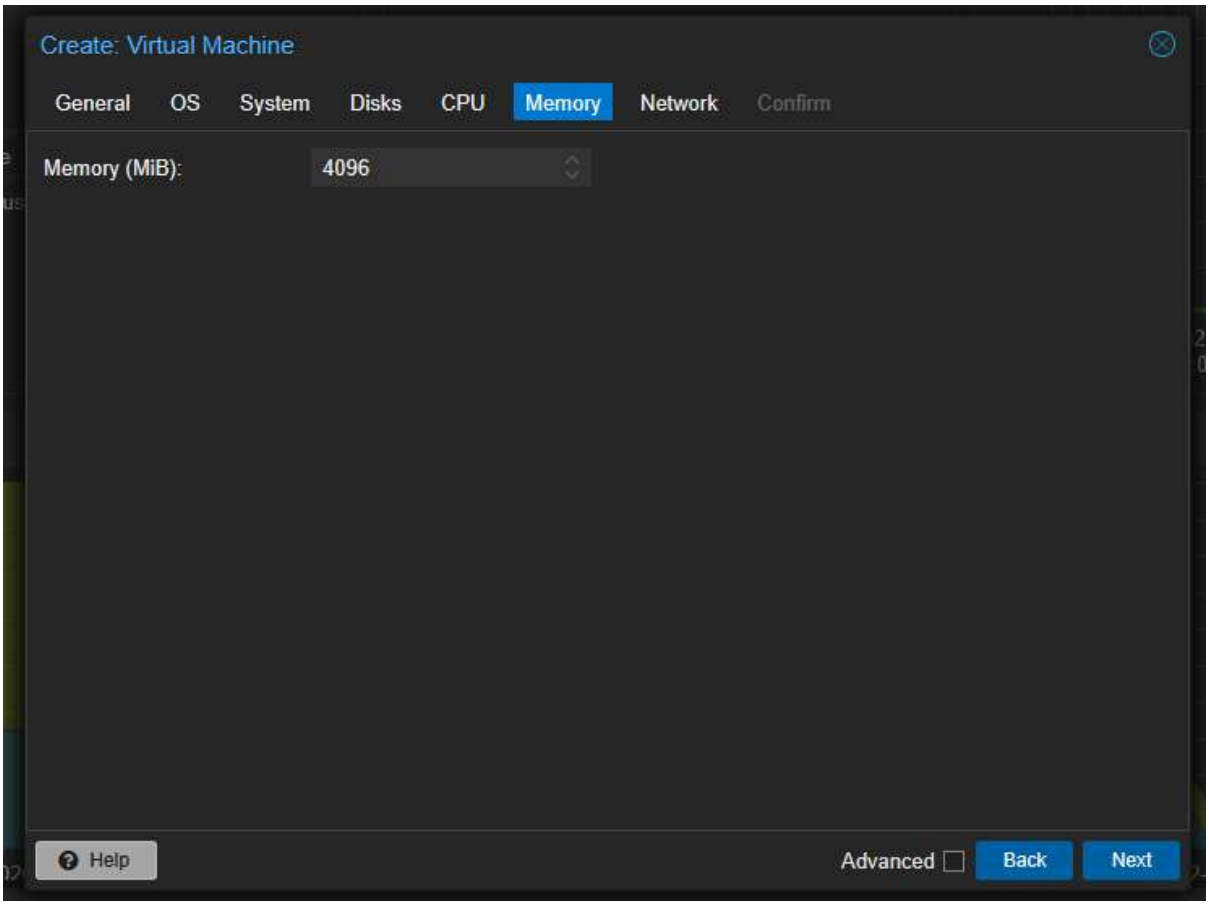

Figure 10 — Ajout de la carte réseau VLAN 70 à la machine virtuelle Bastion

BONNE PRATIQUE : Attribuer une adresse IP statique au bastion dans la plage du VLAN 70. Une adresse dynamique pourrait changer et rendre le bastion inaccessible.

5. Configuration des règles d'accès SSH (ACL Cisco)

Afin de restreindre les connexions SSH aux seuls administrateurs, des listes de contrôle d'accès (ACL) étendues sont configurées sur le routeur Cisco. Ces règles bloquent tout flux SSH entre les VLANs clients, les VLANs serveurs, et entre serveurs, tout en autorisant le reste du trafic.

5.1 Principe de filtrage

- Les VLANs clients (VLAN 10 à VLAN 60) ne peuvent pas initier de connexion SSH vers les serveurs.
- Les serveurs ne peuvent pas initier de connexion SSH entre eux.
- Seul le VLAN 70 (réseau administrateur / bastion) est autorisé à se connecter en SSH.

SÉCURITÉ : Le principe de moindre privilège est appliqué : tout ce qui n'est pas explicitement autorisé est bloqué pour SSH. La règle finale « permit ip any any » maintient le reste du trafic opérationnel.

5.2 ACL configurées sur le routeur

Les ACL suivantes sont appliquées sur chaque sous-interface VLAN du routeur en direction entrante (in). Chaque liste bloque le trafic TCP sur le port 22 (SSH) entre le VLAN source et l'ensemble des sous-réseaux serveurs.

ACL VLAN10_CLIENTS (192.168.230.0/24)

```
Extended IP access list VLAN10_CLIENTS
  10 deny tcp 192.168.230.0 0.0.0.255 192.168.200.0 0.0.0.255 eq 22
  20 deny tcp 192.168.230.0 0.0.0.255 192.168.231.0 0.0.0.255 eq 22
  30 deny tcp 192.168.230.0 0.0.0.255 192.168.232.0 0.0.0.255 eq 22
  40 deny tcp 192.168.230.0 0.0.0.255 192.168.233.0 0.0.0.255 eq 22
  50 deny tcp 192.168.230.0 0.0.0.255 192.168.234.0 0.0.0.255 eq 22
  60 deny tcp 192.168.230.0 0.0.0.255 192.168.235.0 0.0.0.255 eq 22
  70 permit ip any any
```

ACL VLAN20_SSH (192.168.231.0/24)

```
Extended IP access list VLAN20_SSH
  10 deny tcp 192.168.231.0 0.0.0.255 192.168.200.0 0.0.0.255 eq 22
  20 deny tcp 192.168.231.0 0.0.0.255 192.168.230.0 0.0.0.255 eq 22
  30 deny tcp 192.168.231.0 0.0.0.255 192.168.232.0 0.0.0.255 eq 22
  40 deny tcp 192.168.231.0 0.0.0.255 192.168.233.0 0.0.0.255 eq 22
  50 deny tcp 192.168.231.0 0.0.0.255 192.168.234.0 0.0.0.255 eq 22
  60 deny tcp 192.168.231.0 0.0.0.255 192.168.235.0 0.0.0.255 eq 22
  70 permit ip any any
```

ACL VLAN30_SSH (192.168.232.0/24)

```
Extended IP access list VLAN30_SSH
  10 deny tcp 192.168.232.0 0.0.0.255 192.168.200.0 0.0.0.255 eq 22
  20 deny tcp 192.168.232.0 0.0.0.255 192.168.230.0 0.0.0.255 eq 22
  30 deny tcp 192.168.232.0 0.0.0.255 192.168.231.0 0.0.0.255 eq 22
  40 deny tcp 192.168.232.0 0.0.0.255 192.168.233.0 0.0.0.255 eq 22
  50 deny tcp 192.168.232.0 0.0.0.255 192.168.234.0 0.0.0.255 eq 22
  60 deny tcp 192.168.232.0 0.0.0.255 192.168.235.0 0.0.0.255 eq 22
  70 permit ip any any
```

ACL VLAN40_SSH (192.168.233.0/24)

```
Extended IP access list VLAN40_SSH
  10 deny tcp 192.168.233.0 0.0.0.255 192.168.200.0 0.0.0.255 eq 22
  20 deny tcp 192.168.233.0 0.0.0.255 192.168.230.0 0.0.0.255 eq 22
  30 deny tcp 192.168.233.0 0.0.0.255 192.168.231.0 0.0.0.255 eq 22
  40 deny tcp 192.168.233.0 0.0.0.255 192.168.232.0 0.0.0.255 eq 22
  50 deny tcp 192.168.233.0 0.0.0.255 192.168.234.0 0.0.0.255 eq 22
  60 deny tcp 192.168.233.0 0.0.0.255 192.168.235.0 0.0.0.255 eq 22
  70 permit ip any any
```

ACL VLAN50_SSH (192.168.234.0/24)

```
Extended IP access list VLAN50_SSH
```

```
  10 deny tcp 192.168.234.0 0.0.0.255 192.168.200.0 0.0.0.255 eq 22
  20 deny tcp 192.168.234.0 0.0.0.255 192.168.230.0 0.0.0.255 eq 22
  30 deny tcp 192.168.234.0 0.0.0.255 192.168.231.0 0.0.0.255 eq 22
  40 deny tcp 192.168.234.0 0.0.0.255 192.168.232.0 0.0.0.255 eq 22
  50 deny tcp 192.168.234.0 0.0.0.255 192.168.233.0 0.0.0.255 eq 22
  60 deny tcp 192.168.234.0 0.0.0.255 192.168.235.0 0.0.0.255 eq 22
  70 permit ip any any
```

ACL VLAN60_SSH (192.168.235.0/24)

```
Extended IP access list VLAN60_SSH
  10 deny tcp 192.168.235.0 0.0.0.255 192.168.200.0 0.0.0.255 eq 22
  20 deny tcp 192.168.235.0 0.0.0.255 192.168.230.0 0.0.0.255 eq 22
  30 deny tcp 192.168.235.0 0.0.0.255 192.168.231.0 0.0.0.255 eq 22
  40 deny tcp 192.168.235.0 0.0.0.255 192.168.231.0 0.0.0.255 eq 22
  50 deny tcp 192.168.235.0 0.0.0.255 192.168.232.0 0.0.0.255 eq 22
  60 deny tcp 192.168.235.0 0.0.0.255 192.168.233.0 0.0.0.255 eq 22
  70 permit ip any any
```

5.3 Application des ACL aux interfaces

Chaque ACL doit être appliquée en entrée (in) sur la sous-interface correspondant à son VLAN :

Paramètre	Valeur / Description
`interface GigX.10`	ip access-group VLAN10_CLIENTS in
`interface GigX.20`	ip access-group VLAN20_SSH in
`interface GigX.30`	ip access-group VLAN30_SSH in
`interface GigX.40`	ip access-group VLAN40_SSH in
`interface GigX.50`	ip access-group VLAN50_SSH in
`interface GigX.60`	ip access-group VLAN60_SSH in

REMARQUE : GigX correspond à l'interface physique de votre routeur (ex. : GigabitEthernet0/0). Adapter le numéro en fonction de votre équipement. Le VLAN 70 (réseau administrateur) ne porte pas d'ACL SSH restrictive : c'est lui qui autorise l'accès au bastion.

6. Mise en place du menu interactif à la connexion

Afin de simplifier la navigation pour les administrateurs, un script Bash est exécuté automatiquement à chaque connexion SSH sur le bastion. Ce script affiche un menu interactif permettant de sélectionner l'équipement ou le serveur auquel se connecter, sans avoir à mémoriser les commandes SSH.

6.1 Présentation du script bastion-menu.sh

Le script bastion-menu.sh est déposé dans /usr/local/bin/. Il gère :

- L'affichage d'un en-tête ASCII art coloré (Bastion)
- Un menu principal proposant : Routeur, Switch, Proxmox, Serveurs Linux, CLI locale, Déconnexion
- Un sous-menu dédié aux serveurs Linux (Zabbix, Asterix, GLPI, Cloudflare)
- La journalisation de chaque action via logger (visible dans /var/log/syslog)

6.2 Configuration des cibles SSH dans le script

Les hôtes de destination sont déclarés en variables en tête du script. Il suffit de les adapter à l'environnement :

Paramètre	Valeur / Description
`ROUTER_HOST`	R1 — Nom d'hôte SSH du routeur
`SWITCH_HOST`	SW1 — Nom d'hôte SSH du switch
`PROXMOX_HOST`	root@192.168.230.10 — Accès Proxmox en root
`ZABBIX_HOST`	ZABBIX — Nom d'hôte SSH du serveur Zabbix
`ASTERIX_HOST`	Asterix — Nom d'hôte SSH du serveur Asterix
`GLPI_HOST`	GLPI — Nom d'hôte SSH du serveur GLPI
`CLOUFLARE_HOST`	CLOUDFLARE — Nom d'hôte SSH du serveur Cloudflare

REMARQUE : Pour que les connexions fonctionnent avec les noms d'hôtes (ex. : ZABBIX, R1), ces entrées doivent être renseignées dans /etc/hosts sur le bastion ou résolues via DNS.

6.3 Contenu complet du script

```
#!/bin/bash

# ==========================
# CONFIGURATION COULEURS
# ==========================
RED='\033[0;31m'
GREEN='\033[0;32m'
YELLOW='\033[1;33m'
BLUE='\033[1;34m'
CYAN='\033[1;36m'
MAGENTA='\033[1;35m'
WHITE='\033[1;37m'
BOLD='\033[1m'
DIM='\033[2m'
NC='\033[0m'

# ==========================
# CONFIGURATION CIBLES
# ==========================
ROUTER_HOST="R1"
SWITCH_HOST="SW1"
```

```
PROXMOX_HOST="root@192.168.230.10"
ZABBIX_HOST="ZABBIX"
ASTERIX_HOST="Asterix"
GLPI_HOST="GLPI"
CLOUFLARE_HOST="CLOUDFLARE"

# ==========================
# FONCTIONS UTILITAIRES
# ==========================
draw_line()    { echo -e
"${BLUE}================================================================${NC}"; }
pause_return() { echo; read -p "Appuie sur Entrée pour revenir au menu..." _; }

log_action() { logger "[BASTION-MENU] user=$(whoami) action=$1"; }

connect_host() {
  local label="$1" host="$2"
  clear; draw_line
  echo -e "${GREEN}${BOLD}Connexion à ${label}${NC}"
  draw_line
  log_action "connexion_${label}"
  ssh "$host"
  echo -e "${YELLOW}Retour au menu principal...${NC}"; sleep 1
}

# ==========================
# SOUS-MENU SERVEURS LINUX
# ==========================
linux_menu() {
  while true; do
    show_header; draw_line
    echo -e "${WHITE}${BOLD}SERVEURS LINUX${NC}"; draw_line
    echo -e "${GREEN}[1]${NC} Zabbix   ${GREEN}[2]${NC} Asterix"
    echo -e "${GREEN}[3]${NC} Glpi     ${GREEN}[4]${NC} Cloudflare"
    echo -e "${YELLOW}[5]${NC} Retour"
    read -p "Choix : " srv
    case "$srv" in
      1) connect_host "Zabbix"      "$ZABBIX_HOST"    ;;
      2) connect_host "Asterix"     "$ASTERIX_HOST"   ;;
      3) connect_host "Glpi"        "$GLPI_HOST"      ;;
      4) connect_host "Cloudflare"  "$CLOUFLARE_HOST" ;;
      5) break ;;
      *) echo -e "${RED}Choix invalide.${NC}"; sleep 1 ;;
    esac
  done
}

# ==========================
# BOUCLE PRINCIPALE
# ==========================
while true; do
  show_header; show_info_panel; show_footer
  read -p "Choix : " choix
  case "$choix" in
    1) connect_host "Routeur" "$ROUTER_HOST"   ;;
    2) connect_host "Switch"  "$SWITCH_HOST"   ;;
    3) connect_host "Proxmox" "$PROXMOX_HOST"  ;;
    4) linux_menu ;;
    5) exec /bin/bash ;;
    6) clear; log_action "deconnexion"; exit 0 ;;
    *) echo -e "${RED}Choix invalide.${NC}"; sleep 1 ;;
  esac
```

```
done
```

6.4 Déploiement et activation du script

Étape 1 — Dépôt du script sur le bastion

Copier le script dans le répertoire système approprié :

```
nano /usr/local/bin/bastion-menu.sh
```

Coller le contenu du script, sauvegarder (Ctrl+O) et quitter (Ctrl+X).

Étape 2 — Attribution des droits d'exécution

```
chmod +x /usr/local/bin/bastion-menu.sh
```

REMARQUE : Sans ce droit d'exécution, le script ne pourra pas être lancé et OpenSSH retournera une erreur à la connexion.

Étape 3 — Configuration de ForceCommand dans sshd_config

Éditer le fichier de configuration du service OpenSSH :

```
nano /etc/ssh/sshd_config
```

Ajouter ou modifier la ligne suivante :

```
ForceCommand /usr/local/bin/bastion-menu.sh
```

⚠ ATTENTION : La directive ForceCommand remplace la commande demandée par l'utilisateur lors de la connexion SSH. Même si l'utilisateur tente de passer une commande directe (ex. : ssh bastion ls), le menu sera toujours affiché. Veillez à conserver un accès de secours (console Proxmox) avant d'activer cette directive.

Étape 4 — Redémarrage du service SSH

```
systemctl restart sshd
```

Vérifier l'état du service après redémarrage :

```
systemctl status sshd
```

BONNE PRATIQUE : Tester la connexion depuis une session SSH distincte avant de fermer la session courante. En cas d'erreur de configuration, la session en cours reste ouverte et permet de corriger sans perte d'accès.

7. Intégration Active Directory

L'intégration du bastion au domaine Active Directory (domaine : projet.com) permet à chaque administrateur de se connecter avec ses identifiants de domaine. Cela offre deux avantages majeurs :

- Traçabilité individuelle : chaque connexion est associée à un compte nominatif, facilitant l'audit
- Gestion centralisée : la révocation d'un compte AD coupe immédiatement l'accès au bastion

7.1 Installation des paquets nécessaires

13. Se connecter au bastion en root ou avec un compte sudo.
14. Exécuter la commande d'installation suivante :

```
apt install realmd sssd sssd-tools libnss-sss libpam-sss adcli samba-common-bin oddjob oddjob-mkhomedir packagekit -y
```

Paramètre	Valeur / Description
`realmd`	Découverte et jonction au domaine Active Directory
`sssd / sssd-tools`	Gestion des authentifications et sessions utilisateur
`libnss-sss`	Résolution des noms d'utilisateurs via SSSD
`libpam-sss`	Module PAM pour l'authentification SSSD
`adcli`	Outil d'intégration Active Directory en ligne de commande
`samba-common-bin`	Outils Samba pour la communication avec AD
`oddjob / oddjob-mkhomedir`	Création automatique du répertoire utilisateur à la connexion

7.2 Vérification de la détection du domaine

15. Avant de joindre le domaine, vérifier qu'il est bien détecté depuis le bastion :

```
/usr/sbin/realm discover projet.com
```

La sortie attendue confirme la détection du domaine Kerberos :

```
projet.com
  type: kerberos
  realm-name: PROJET.COM
  domain-name: projet.com
  configured: kerberos-member
  server-software: active-directory
  client-software: sssd
  required-package: sssd-tools
  required-package: sssd
```

```
    required-package: libnss-sss
    required-package: libpam-sss
    required-package: adcli
    required-package: samba-common-bin
    login-formats: %U@projet.com
    login-policy: allow-realm-logins
```

⚠ **ATTENTION :** Si la commande realm discover ne retourne rien ou affiche une erreur, vérifier que le bastion peut résoudre le domaine projet.com via DNS et que le contrôleur de domaine est joignable (ping, port 88/TCP Kerberos).

7.3 Jonction au domaine Active Directory

16. Exécuter la commande de jonction au domaine avec le compte Administrateur AD :

```
/usr/sbin/realm join projet.com -U Administrateur
```

17. Saisir le mot de passe du compte Administrateur lorsqu'il est demandé.
18. Si aucun message d'erreur n'apparaît, la machine est désormais membre du domaine.

7.4 Vérification de la jonction au domaine

19. Vérifier que la machine est bien membre du domaine :

```
/usr/sbin/realm list
```

La sortie doit afficher le domaine avec le statut configured: kerberos-member. La même sortie que realm discover est attendue, confirmant la jonction réussie.

☑ **BONNE PRATIQUE :** Après jonction, tester immédiatement une connexion avec un compte du domaine (ex. : ssh utilisateur@projet.com@bastion) depuis une machine du VLAN administrateur. Cela confirme que l'intégration est fonctionnelle.

7.5 Création automatique des répertoires utilisateurs

Par défaut, les utilisateurs du domaine n'ont pas de répertoire home sur le bastion. La configuration suivante crée automatiquement ce répertoire lors de la première connexion.

Étape 1 — **Installation du paquet PAM complémentaire**

```
apt install libpam-runtime -y
```

Étape 2 — **Configuration via pam-auth-update**

20. Exécuter la commande de configuration PAM :
    `/usr/sbin/pam-auth-update`

21. Dans le menu interactif qui s'affiche, cocher la case Create home directory on login à l'aide de la barre espace, puis valider avec OK.

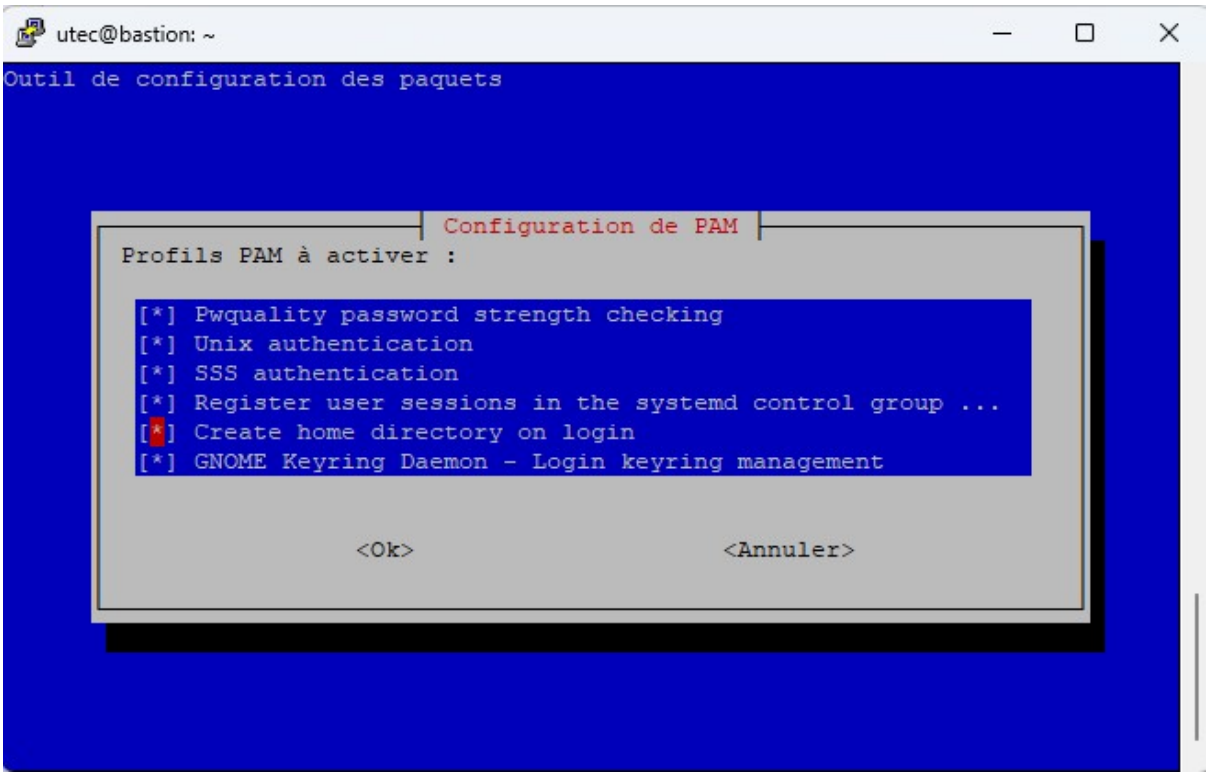

Figure 11 — Menu pam-auth-update : activation de la création automatique du répertoire utilisateur

REMARQUE : Une fois cette option activée, un répertoire /home/utilisateur@projet.com est créé automatiquement lors de la première connexion de chaque utilisateur du domaine. Ce répertoire persiste entre les sessions.

8. Vérification et tests

8.1 Test du filtrage SSH par ACL

- Depuis une machine du VLAN client (ex. : VLAN 10), tenter une connexion SSH vers un serveur : la connexion doit être refusée (timeout ou connexion refusée).
- Depuis le bastion (VLAN 70), tenter une connexion SSH vers un serveur : la connexion doit aboutir.

8.2 Test du menu interactif

- Se connecter en SSH au bastion depuis le VLAN 70 : le menu doit s'afficher automatiquement.

- Sélectionner une cible (ex. : Proxmox) et vérifier que la connexion SSH s'établit.
- Tester la déconnexion (option 6) et le retour au shell local (option 5).

8.3 Test de l'authentification Active Directory

- Se connecter au bastion avec un compte du domaine (ex. : ssh jdupont@projet.com@bastion-ip) : la connexion doit réussir.
- Vérifier que le répertoire /home/jdupont@projet.com/ a bien été créé.
- Contrôler les logs de connexion :
  ```
  journalctl -u sshd -f
  ```

8.4 Vérification des journaux d'audit

- Consulter les actions tracées par le script :
  ```
  grep BASTION-MENU /var/log/syslog
  ```

☑ **BONNE PRATIQUE :** Planifier des tests de connexion réguliers (hebdomadaires) pour s'assurer que les ACL, le menu et l'authentification AD fonctionnent toujours après des mises à jour système ou des changements réseau.

9. Conclusion

Cette documentation a présenté le déploiement complet d'un bastion SSH sécurisé, depuis la création du VLAN administrateur jusqu'à l'intégration Active Directory. Les composants mis en œuvre sont :

- VLAN 70 dédié : isolement du réseau d'administration
- ACL Cisco : blocage du trafic SSH inter-VLAN, seul le bastion est autorisé
- Machine virtuelle Bastion : déployée sur Proxmox avec interface VLAN 70
- Menu interactif : script Bash avec ForceCommand pour une navigation simplifiée et auditée
- Authentification AD : jonction au domaine projet.com via realmd et SSSD
- Création automatique des répertoires home : via pam-auth-update

Cette architecture respecte les bonnes pratiques de sécurité en matière d'administration système : accès centralisé, traçabilité nominative, principe de moindre privilège et réduction de la surface d'attaque.

☑ **Améliorations possibles**

• Mettre en place des clés SSH par utilisateur AD pour supprimer l'authentification par mot de passe

• Intégrer un outil de journalisation centralisée (ex. : envoi des logs vers Zabbix ou un SIEM)

• Configurer le MFA (authentification à deux facteurs) pour les connexions au bastion

• Restreindre les utilisateurs AD autorisés à se connecter au bastion via la directive AllowUsers dans sshd_config

Historique des révisions

Version	Date	Auteur	Description
1.0	—	S. Dos Santos	Version initiale brute
1.1	23/03/2026	—	Révision complète : corrections, restructuration et captures d'écran