BTS SIO · OPTION SISR · SESSION 2026
BAGARRY Matthias

DOSSIER TECHNIQUE

Industrialisation du déploiement Linux par clonage de VM — VMware Workstation · Debian 12

0 CONTEXTE ET OBJECTIFS

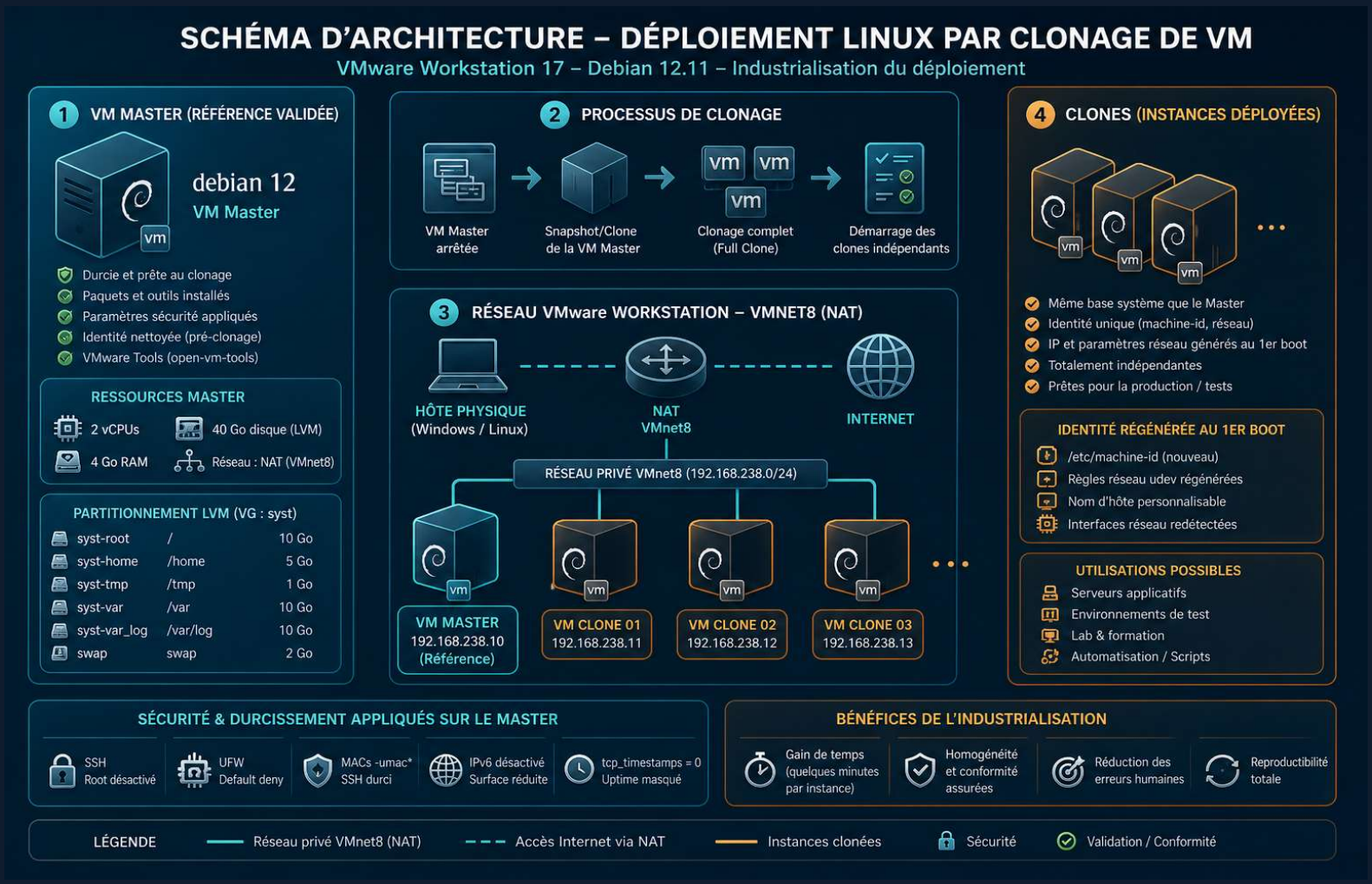
Ce projet s'inscrit dans une démarche personnelle de montée en compétences sur la virtualisation et l'industrialisation des déploiements Linux. L'objectif est de passer d'une installation manuelle unitaire à un mode de déploiement reproductible basé sur une machine de référence validée, appelée Master. L'environnement utilisé est VMware Workstation 17 avec Debian 12.11.

► OBJECTIFS PÉDAGOGIQUES

Maîtriser la configuration avancée d'une VM sous VMware Workstation, construire un système Debian 12 durci et prêt à la duplication, nettoyer son identité avant clonage, et déployer des instances opérationnelles indépendantes. En référence, les mécanismes équivalents sous vSphere (Templates, Customization Specifications) sont identifiés pour intégrer la démarche dans un contexte professionnel.

► POURQUOI INDUSTRIALISER LE DÉPLOIEMENT

Un parc de 10 serveurs représente déjà plusieurs heures de travail répétitif à risque d'erreur. Le clonage à partir d'un Master validé ramène ce temps à quelques minutes par instance, avec la garantie que chaque machine déployée part du même état : mêmes paquets installés, mêmes paramètres de sécurité, même base système.



Ce schéma présente l'architecture de déploiement par clonage de VM mise en œuvre dans le cadre de ce PoC. La VM Master, configurée et durcie sous Debian 12.11, sert de référence unique à partir de laquelle les clones sont générés.

1 CONFIGURATION DE LA VM MASTER

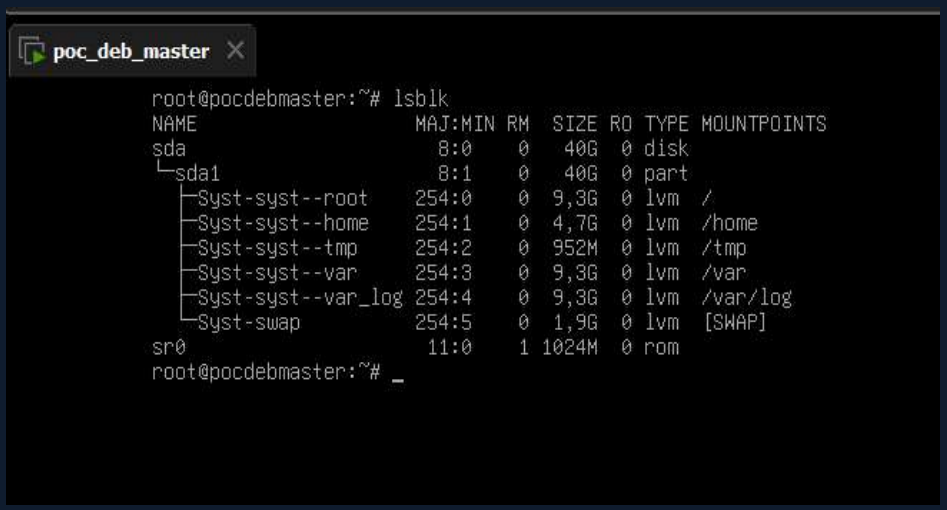
1.1 Ressources et partitionnement LVM

La VM Master est créée sous VMware Workstation 17. Les ressources sont dimensionnées pour représenter un serveur Linux minimal de production. Le partitionnement LVM est appliqué dès l'installation pour isoler les volumes critiques et permettre une extension à chaud si nécessaire.

Ressource	Valeur
Système d'exploitation	Debian 12.11 Bookworm - Installation minimale
Processeurs	2 vCPUs - VT-x/AMD-V activé, EPT/RVI activé
Mémoire vive	4 Go RAM
Disque	40 Go VMDK - partitionné en LVM (VG : syst)
Réseau	NAT (VMnet8) - DHCP, accès Internet via l'hôte

Volume (LV)	Point de montage	Taille	Justification
`syst-root`	`/`	10 Go	Système + binaires installés
`syst-home`	`/home`	5 Go	Répertoires utilisateurs
`syst-tmp`	`/tmp`	1 Go	Isolé pour limiter les attaques via /tmp
`syst-var`	`/var`	10 Go	Caches APT, données services
`syst-var_log`	`/var/log`	10 Go	Isolé — protège le système contre la saturation
`swap`	`swap`	2 Go	Débordement mémoire

> ► POURQUOI LVM PLUTÔT QU'UN PARTITIONNEMENT STANDARD
>
> Le partitionnement LVM sépare la gestion du système de fichiers du matériel physique. Contrairement au partitionnement standard (MBR/GPT) qui fige définitivement la taille des partitions, LVM permet de redimensionner un volume à chaud avec lvextend, sans redémarrer le serveur. Cette flexibilité est cruciale pour /var et /var/log, sujets à des remplissages rapides et imprévisibles.



2 INSTALLATION ET DURCISSEMENT DU MASTER

2.1 Mise à jour et installation des outils système

Après l'installation minimale (toutes les options décochées), le système est mis à jour et un socle d'outils d'administration est installé en une seule commande. L'objectif est d'avoir une base identique, reproductible, sans paquets superflus.

```
# Mise à jour complète
apt-get update && apt-get full-upgrade -y
# Socle d'outils système en une seule commande
apt-get install -y curl wget vim-tiny sudo openssh-server ufw \
    open-vm-tools net-tools iproute2 htop lsof psmisc \
    rsync ncdu screen dos2unix man-db ca-certificates
# Nettoyage des paquets inutiles hérités de l'installation
apt-get purge tasksel tasksel-data && apt-get autoremove --purge -y
# Timezone
timedatectl set-timezone Europe/Paris
```

2.2 Durcissement SSH, UFW et sysctl

```
# --- SSH : durcissement de /etc/ssh/sshd_config ---
sed -i 's/^#PermitRootLogin.*/PermitRootLogin no/' /etc/ssh/sshd_config
# Restriction des algorithmes MAC faibles (vuln. SSH weak MAC)
echo 'MACs -umac*' >> /etc/ssh/sshd_config
systemctl restart ssh
# Vérification des MACs actifs
sshd -T | grep macs
# --- UFW : stratégie default deny ---
ufw default deny incoming && ufw default allow outgoing
ufw allow ssh && ufw enable
# --- sysctl : durcissement réseau ---
# Ajout dans /etc/sysctl.conf
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
# Correctif TCP timestamps (fuite d'information sur l'uptime)
net.ipv4.tcp_timestamps = 0
sysctl -p
```

► PRINCIPE DE DÉFENSE EN PROFONDEUR

Quatre mesures complémentaires sont appliquées. (1) Root SSH désactivé : élimine la moitié des attaques brute-force. (2) MACs -umac* : corrige une vulnérabilité connue sur les algorithmes HMAC faibles. (3) UFW default deny : tout trafic entrant non explicitement autorisé est bloqué. (4) IPv6 désactivé et tcp_timestamps à 0 : réduit la surface d'attaque réseau et évite la divulgation de l'uptime du serveur.

► MÉCANIQUE DES RESTRICTIONS RÉSEAU (sysctl)

La désactivation d'IPv6 répond au principe de réduction de la surface d'attaque. Si IPv6 n'est pas routé sur le réseau, le laisser activé expose la machine à des attaques locales comme l'interception de trafic via de fausses annonces SLAAC. La mise à zéro de tcp_timestamps neutralise la divulgation de l'uptime : un attaquant utilise cette donnée pour déduire la date du dernier redémarrage et identifier si le noyau a été patché contre des vulnérabilités critiques (CVE).

2.3 Intégration VMware : open-vm-tools vs VMware Tools

L'intégration avec l'hyperviseur est assurée par les VMware Tools. Deux méthodes coexistent selon le contexte : en lab sous VMware Workstation, open-vm-tools s'installe via apt et suffit amplement.

En environnement vSphere d'entreprise, les VMware Tools propriétaires peuvent être imposés par la politique système :

Critère	open-vm-tools (apt)	VMware Tools (ISO)
Installation	`apt-get install open-vm-tools`	`ISO montée + ./vmware-install.pl`
Maintenance	Via apt-get upgrade	Réinstallation manuelle à chaque version
Contexte recommandé	Lab, vSphere moderne (≥ 6.5)	Anciens ESXi, politique d'entreprise spécifique

3 NETTOYAGE DE L'IDENTITÉ (PRÉ-CLONAGE)

C'est l'étape la plus critique du processus. Avant d'éteindre la VM pour la cloner, tous les identifiants uniques générés à l'installation doivent être supprimés. Sans cette étape, les clones partagent la même identité système : conflits d'adresse IP sur le réseau, logs indéchiffrables, baux DHCP en conflit.

```
# 1. Vider le machine-id (pas rm — systemd recrée au 1er boot)
truncate -s 0 /etc/machine-id
rm -f /var/lib/dbus/machine-id
# 2. Supprimer les règles réseau persistantes
rm -f /etc/udev/rules.d/70-persistent-net.rules
# 3. Forcer le DHCP (chaque clone reçoit sa propre IP)
printf 'auto ens33\niface ens33 inet dhcp\n' > /etc/network/interfaces
# 4. Nettoyer logs et cache apt
apt-get clean && rm -rf /var/log/*.log /var/log/apt/*
# 5. Éteindre proprement — NE JAMAIS cloner une VM allumée
shutdown -h now
```

► POURQUOI TRUNCATE ET PAS RM SUR /ETC/MACHINE-ID

Supprimer le fichier avec rm provoquerait une erreur au démarrage de certains services systemd qui s'attendent à ce que le fichier existe, même vide. La commande truncate -s 0 vide le contenu sans supprimer le fichier. Au premier boot du clone, systemd détecte un machine-id vide, en génère un nouveau unique, et le client DHCP l'utilise pour obtenir un bail distinct du Master.

► CONSÉQUENCES D'UN MAUVAIS NETTOYAGE

Le client DHCP de Debian utilise le machine-id pour formuler sa requête de bail. Si truncate n'est pas appliqué, deux clones possèdent le même identifiant : le serveur DHCP leur attribue alors la même adresse IP, provoquant une tempête de paquets ARP sur le réseau et des coupures de connectivité intermittentes pour les deux machines. Sur le plan de la supervision, un serveur Syslog centralisé écrase ou mélange les logs des différents clones en les considérant comme une seule entité, rendant tout audit de sécurité impossible.

4 CLONAGE ET DÉPLOIEMENT

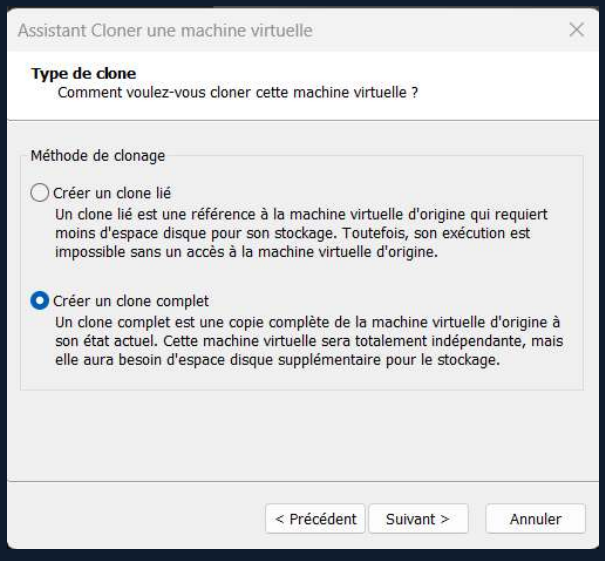
4.1 Full Clone sous VMware Workstation

Le Full Clone génère une copie complète et entièrement indépendante du VMDK. Après la création, les deux VMs n'ont plus aucun lien : chacune peut être déplacée, modifiée ou supprimée sans affecter l'autre. La procédure : VM Master éteinte → clic droit → Manage → Clone → Full Clone → nommer le clone.

Critère	Full Clone ✓	Linked Clone
Dépendance Master	Aucune — totalement indépendant	Dépend du snapshot Master
Performances I/O	Optimales — disque propre	Dégradées (overhead copy-on-write)
Usage recommandé	Déploiement final, production	Tests rapides, économie de stockage

► MÉCANIQUE DES LINKED CLONES (COPY-ON-WRITE)

Les clones liés utilisent un mécanisme de disque Copy-on-Write (CoW). Le clone lit les données depuis le disque en lecture seule du Master, mais écrit ses propres modifications dans un fichier delta séparé. Plus le clone est utilisé, plus ce fichier delta grossit. Cela ralentit les opérations I/O car le contrôleur doit en permanence consolider les données réparties entre le disque de base et le delta. Le Full Clone, disposant de ses propres blocs de stockage dédiés, élimine complètement cet overhead et est donc indispensable en production.



4.2 Infrastructure vSphere et hyperviseur ESXi

Dans un environnement professionnel, VMware vSphere remplace VMware Workstation. vSphere est une plateforme composée de deux éléments principaux : l'hyperviseur ESXi, installé directement sur le matériel physique (bare-metal), et vCenter Server, le système de gestion centralisé de toute l'infrastructure.

► ESXi : HYPERVISEUR DE TYPE 1

Contrairement à VMware Workstation (hyperviseur type 2 qui s'installe sur un OS hôte), ESXi est un hyperviseur de type 1 : il s'installe directement sur le serveur physique, sans système d'exploitation intermédiaire. Cette architecture réduit la surcharge système et améliore les performances des VMs. Chaque ESXi gère ses propres VMs et ses ressources CPU, RAM et stockage réseau.

► ORDONNANCEMENT ET VMKERNEL

La supériorité d'ESXi sur Workstation réside dans la gestion de l'ordonnancement CPU. Un hyperviseur de type 2 subit les interruptions de l'OS hôte (Windows ou Linux). ESXi, en s'installant en bare-metal, communique directement avec le matériel via son propre micro-noyau (VMkernel). Cela garantit une allocation stricte et ininterrompue des cycles CPU et de la RAM aux machines virtuelles, éliminant les latences induites par un système d'exploitation intermédiaire. C'est ce qui rend vSphere adapté aux charges de production, là où Workstation reste un outil de lab.

4.3 vCenter Server : gestion centralisée

vCenter Server est le point de contrôle unique de toute l'infrastructure vSphere. Là où VMware Workstation gère des VMs localement machine par machine, vCenter offre une vue unifiée sur l'ensemble du parc : tous les hôtes ESXi, toutes les VMs, les datastores (espaces de stockage partagés) et les réseaux virtuels, depuis une seule interface web.

► CE QUE PERMET VCENTER

Depuis vCenter, un administrateur peut : déployer simultanément plusieurs VMs depuis un même Template sur n'importe quel ESXi du cluster, migrer une VM à chaud d'un ESXi à un autre sans interruption de service (vMotion), surveiller en temps réel la consommation CPU et RAM, gérer les alertes et seuils, et appliquer des politiques de sécurité uniformes. Tout ce qui est fait manuellement dans ce dossier (nettoyage identité, clonage, hostname) est automatisé et centralisé dans vCenter.

L'interface vCenter n'était pas disponible pour être ajoutée à ce dossier, mais en production l'inventaire des VMs et hôtes ESXi serait visible dans vCenter Web Client (vSphere Client).

4.4 Templates et Customization Specifications

En environnement vSphere, la même logique de clonage s'applique à l'échelle de l'infrastructure. La VM Master validée est convertie en Template depuis vCenter : clic droit → Cloner vers un modèle. Ce Template est stocké sur un datastore centralisé, accessible depuis tous les ESXi du cluster. La convention de nommage en production est du type tmpldeb12_11_AAAAMMJJ pour tracer la version de l'OS et la date de création.

> **► CUSTOMIZATION SPECIFICATION : L'AUTOMATISATION DU POST-CLONAGE**
>
> Une Customization Specification est un profil de personnalisation attaché au déploiement d'un Template. Elle remplace toutes les étapes manuelles de la section 5 : elle redéfinit automatiquement le hostname, régénère le machine-id, configure l'adresse réseau (statique ou DHCP) et remet la clé SSH à zéro. Chaque clone déployé depuis vCenter est immédiatement opérationnel, sans aucune intervention manuelle post-clonage.

5 VALIDATION TECHNIQUE (RECETTE)

Après le premier démarrage du clone, systemd génère automatiquement un nouveau machine-id. La recette valide que le clone est bien indépendant du Master sur les trois points critiques : identité système, adresse réseau et connectivité SSH.

```
# Individualisation du clone
hostnamectl set-hostname debclone01
# Vérifications post-clonage
cat /etc/machine-id          # doit être différent du Master
ip a show ens33              # IP distincte du Master
hostname                     # debclone01
systemctl status open-vm-tools ssh ufw
# Test SSH depuis l'hôte
ssh user@<IP-clone>
```

Commande	Critère de succès
`cat /etc/machine-id`	Valeur différente du Master (32 hex générés)
`ip a show ens33`	IP distincte, bail DHCP obtenu
`sshd -T | grep macs`	Aucun algorithme umac* dans la liste active
`ufw status`	Active, seul SSH autorisé
`sysctl net.ipv4.tcp_timestamps`	Retourne 0
`ssh user@<IP-clone>`	Connexion réussie, root refusé

poc_deb_master | poc_deb_clone01

```
adminsys@pocdebmaster:~$ cat /etc/machine-id
896f72fdaa0345ddb895b4e536e5814f
adminsys@pocdebmaster:~$ ip a show ens33
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:ae:92:81 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.40.129/24 brd 192.168.40.255 scope global dynamic ens33
       valid_lft 1297sec preferred_lft 1297sec
adminsys@pocdebmaster:~$ hostname
pocdebmaster
adminsys@pocdebmaster:~$ systemctl status open-vm-tools ssh ufw
● open-vm-tools.service - Service for virtual machines hosted on VMware
     Loaded: loaded (/lib/systemd/system/open-vm-tools.service; enabled; preset: enabled)
     Active: active (running) since Sun 2026-05-03 04:45:44 CEST; 8min ago
       Docs: http://open-vm-tools.sourceforge.net/about.php
   Main PID: 645 (vmtoolsd)
      Tasks: 3 (limit: 4607)
     Memory: 12.2M
        CPU: 2.917s
     CGroup: /system.slice/open-vm-tools.service
             └─645 /usr/bin/vmtoolsd

Warning: some journal files were not opened due to insufficient permissions.

● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; preset: enabled)
     Active: active (running) since Sun 2026-05-03 04:45:46 CEST; 8min ago
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 987 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 990 (sshd)
      Tasks: 1 (limit: 4607)
     Memory: 3.2M
        CPU: 79ms
     CGroup: /system.slice/ssh.service
             └─990 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Warning: some journal files were not opened due to insufficient permissions.

● ufw.service - Uncomplicated firewall
     Loaded: loaded (/lib/systemd/system/ufw.service; enabled; preset: enabled)
     Active: active (exited) since Sun 2026-05-03 04:45:45 CEST; 8min ago
       Docs: man:ufw(8)
    Process: 524 ExecStart=/lib/ufw/ufw-init start quiet (code=exited, status=0/SUCCESS)
   Main PID: 524 (code=exited, status=0/SUCCESS)
        CPU: 112ms

Warning: some journal files were not opened due to insufficient permissions.
adminsys@pocdebmaster:~$
```

poc_deb_master | poc_deb_clone01

```
adminsys@debclone01:~$ cat /etc/machine-id
896f72fdaa0345ddb895b4e536e5814f
adminsys@debclone01:~$ ip a show ens33
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:ad:9c:33 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.40.130/24 brd 192.168.40.255 scope global dynamic ens33
       valid_lft 1664sec preferred_lft 1664sec
adminsys@debclone01:~$ hostname
debclone01
adminsys@debclone01:~$ systemctl status open-vm-tools ssh ufw
● open-vm-tools.service - Service for virtual machines hosted on VMware
     Loaded: loaded (/lib/systemd/system/open-vm-tools.service; enabled; preset: enabled)
     Active: active (running) since Sun 2026-05-03 04:50:09 CEST; 2min 49s ago
       Docs: http://open-vm-tools.sourceforge.net/about.php
   Main PID: 635 (vmtoolsd)
      Tasks: 3 (limit: 4607)
     Memory: 11.2M
        CPU: 1.221s
     CGroup: /system.slice/open-vm-tools.service
             └─635 /usr/bin/vmtoolsd

Warning: some journal files were not opened due to insufficient permissions.

● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; preset: enabled)
     Active: active (running) since Sun 2026-05-03 04:50:11 CEST; 2min 47s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 985 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 988 (sshd)
      Tasks: 1 (limit: 4607)
     Memory: 3.2M
        CPU: 78ms
     CGroup: /system.slice/ssh.service
             └─988 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Warning: some journal files were not opened due to insufficient permissions.

● ufw.service - Uncomplicated firewall
     Loaded: loaded (/lib/systemd/system/ufw.service; enabled; preset: enabled)
     Active: active (exited) since Sun 2026-05-03 04:50:10 CEST; 2min 49s ago
       Docs: man:ufw(8)
    Process: 520 ExecStart=/lib/ufw/ufw-init start quiet (code=exited, status=0/SUCCESS)
   Main PID: 520 (code=exited, status=0/SUCCESS)
        CPU: 133ms

Warning: some journal files were not opened due to insufficient permissions.
adminsys@debclone01:~$ _
```