Travaux Diriges

Administration Linux Avancee

Fil rouge : Mise en production de l'infrastructure NovaTech

8 TD progressifs - de l'installation a la supervision

Le fil rouge : NovaTech

Contexte general

Vous etes administrateur systeme junior chez **NovaTech**, une PME de 50 personnes qui developpe des applications web.

L'entreprise migre son infrastructure vers des serveurs Linux autogeres.

Votre mission : **construire, securiser, maintenir et superviser** l'infrastructure de production, etape par etape.

> Chaque TD correspond a une phase du projet. Le travail de chaque TD s'appuie sur le precedent.

Infrastructure cible

Machine	Role	OS
`srv-prod`	Serveur applicatif principal	Rocky Linux / AlmaLinux
`srv-backup`	Serveur de sauvegarde	Rocky Linux / AlmaLinux
`srv-nagios`	Supervision	Rocky Linux / AlmaLinux

> Environnement : 3 VMs (KVM/VirtualBox/VMware). Chaque VM : 2 vCPU, 2 Go RAM, 2 disques de 20 Go minimum.

Prerequis techniques

- Hyperviseur installe (KVM + virt-manager, VirtualBox ou VMware)
- ISO Rocky Linux 9 / AlmaLinux 9 telechargee
- Acces reseau entre les VMs (reseau host-only ou bridge)
- Connaissance de base de Linux (navigation, edition, services)

Configuration reseau des VMs

Machine	IP	Passerelle
`srv-prod`	192.168.56.10/24	192.168.56.1
`srv-backup`	192.168.56.11/24	192.168.56.1
`srv-nagios`	192.168.56.12/24	192.168.56.1

Ajoutez dans `/etc/hosts` de chaque machine :

```
192.168.56.10  srv-prod     srv-prod.novatech.lan
192.168.56.11  srv-backup   srv-backup.novatech.lan
192.168.56.12  srv-nagios   srv-nagios.novatech.lan
```

TD 1 - Installation et deploiement

TD 1 - Objectif

Contexte NovaTech : La direction a valide le projet. Vous devez installer le premier serveur (`srv-prod`) de maniere **reproductible** et avec un stockage **resilient**.

A la fin de ce TD, vous aurez :

- Installe `srv-prod` manuellement une premiere fois
- Cree un fichier Kickstart a partir de cette installation
- Configure un stockage RAID1 + LVM
- Securise le chargeur de demarrage GRUB

TD 1 - Etape 1 : Installation initiale

1. Creez une VM `srv-prod` avec **2 disques virtuels** de 20 Go chacun
2. Installez Rocky Linux 9 en mode **minimal** (pas d'interface graphique)
3. Pendant l'installation, configurez :
   - Nom d'hote : `srv-prod.novatech.lan`
   - Utilisateur admin : `ntadmin`
   - Reseau : IP statique `192.168.56.10/24`

TD 1 - Etape 2 : Recuperer et analyser le Kickstart

4. Connectez-vous a `srv-prod` et recuperez le fichier Kickstart genere automatiquement par l'installeur :

```
cat /root/anaconda-ks.cfg
```

Ce fichier contient l'ensemble des reponses donnees pendant l'installation. Vous devriez y trouver :

- `%packages` : la liste des paquets installes (ici `@core` pour une install minimale)
- `network` : la configuration reseau choisie
- `part` / `logvol` : le schema de partitionnement

5. Copiez ce fichier sur votre poste hote :

```
scp root@192.168.56.10:/root/anaconda-ks.cfg ./ks-original.cfg
```

TD 1 - Etape 3 : Creer un Kickstart RAID1 + LVM

6. Creez un nouveau fichier `ks-novatech.cfg` base sur l'original. Commencez par les parametres generaux :

```
# Langue, clavier, fuseau
lang fr_FR.UTF-8
keyboard fr-latin1
timezone Europe/Paris --utc

# Reseau
network --bootproto=static --ip=192.168.56.10 --netmask=255.255.255.0 \
        --gateway=192.168.56.1 --nameserver=8.8.8.8 \
        --hostname=srv-prod.novatech.lan --activate
```

TD 1 - Etape 3 : Kickstart - Partitionnement

Ajoutez le schema de partitionnement RAID1 + LVM :

```
# Partitionnement RAID1 + LVM
clearpart --all --initlabel
part /boot --fstype=ext4 --size=1024 --ondisk=sda
part raid.01 --size=1 --grow --ondisk=sda
part raid.02 --size=1 --grow --ondisk=sdb
raid pv.01 --level=1 --device=md0 raid.01 raid.02
volgroup vg_prod pv.01
logvol / --vgname=vg_prod --name=lv_root --fstype=ext4 --size=10240
logvol /var --vgname=vg_prod --name=lv_var --fstype=ext4 --size=5120
logvol /home --vgname=vg_prod --name=lv_home --fstype=ext4 --size=3072
logvol swap --vgname=vg_prod --name=lv_swap --size=2048
```

TD 1 - Etape 3 : Kickstart (suite)

Ajoutez la section paquets et post-installation :

```
# Mot de passe root (genere avec : openssl passwd -6 'Mon_mot_de_passe')
rootpw --iscrypted <hash>

# Utilisateur
user --name=ntadmin --groups=wheel --password=<hash> --iscrypted

# Paquets
%packages
@core
vim-enhanced
rsync
%end

# Post-installation
%post
echo "192.168.56.11  srv-backup srv-backup.novatech.lan" >> /etc/hosts
echo "192.168.56.12  srv-nagios srv-nagios.novatech.lan" >> /etc/hosts
%end
```

TD 1 - Etape 4 : Valider le Kickstart

7. Validez la syntaxe du fichier avec `ksvalidator` :

```
dnf install pykickstart
ksvalidator ks-novatech.cfg
```

> Si aucune erreur ne s'affiche, le fichier est syntaxiquement correct.

TD 1 - Etape 4 : Tester le Kickstart

8. Rendez le fichier accessible en HTTP depuis votre poste hote :

```
# Sur le poste hote, dans le repertoire contenant ks-novatech.cfg :
python3 -m http.server 8080
```

9. Creez une nouvelle VM `srv-prod-v2` avec 2 disques, bootez sur l'ISO et au menu d'installation appuyez sur `Tab` (BIOS) ou `e` (UEFI), puis ajoutez :

```
inst.ks=http://192.168.56.1:8080/ks-novatech.cfg
```

L'installation se deroule alors sans aucune intervention.

TD 1 - Etape 5 : Verifier le RAID et le LVM

10. Une fois `srv-prod` installe (via Kickstart ou manuellement avec RAID+LVM), verifiez :

```
cat /proc/mdstat
```

Resultat attendu : Le RAID1 est actif avec les deux disques synchronises :

```
md0 : active raid1 sdb1[1] sda2[0]
      19921920 blocks super 1.2 [2/2] [UU]
```

> `[UU]` signifie que les deux disques sont operationnels. `[U_]` indiquerait un disque manquant.

TD 1 - Etape 5 : Verification (suite)

11. Verifiez les Logical Volumes :

```
lvs
```

Resultat attendu :

```
  LV      VG      Attr       LSize
  lv_home vg_prod -wi-ao----  3.00g
  lv_root vg_prod -wi-ao---- 10.00g
  lv_swap vg_prod -wi-ao----  2.00g
  lv_var  vg_prod -wi-ao----  5.00g
```

12. Verifiez le montage :

```
df -h
```

Vous devez voir `/`, `/var`, `/home` et `/boot` montes sur les bons volumes.

TD 1 - Etape 6 : Securiser le boot

13. Protegez GRUB par mot de passe :

```
grub2-setpassword
```

Cette commande cree le fichier `/boot/grub2/user.cfg` contenant le hash PBKDF2 du mot de passe.

14. Redemarrez et tentez de modifier une entree GRUB avec `e` : le mot de passe est maintenant demande.

> Le demarrage normal (sans edition) ne demande pas de mot de passe. Seule la modification des parametres de boot est protegee. Cela empeche un attaquant d'ajouter `init=/bin/bash` pour obtenir un shell root.

TD 1 - Etape 7 : Installer srv-backup

15. Dupliquez le Kickstart en adaptant pour `srv-backup` :

```
sed -e 's/192.168.56.10/192.168.56.11/g' \
    -e 's/srv-prod/srv-backup/g' \
    -e 's/vg_prod/vg_backup/g' \
    ks-novatech.cfg > ks-backup.cfg
```

16. Installez `srv-backup` avec ce Kickstart.

17. Depuis chaque machine, verifiez la connectivite :

```
# Depuis srv-prod :
ping -c 3 srv-backup.novatech.lan
# Depuis srv-backup :
ping -c 3 srv-prod.novatech.lan
```

> **Resultat attendu** : Les pings repondent. Si ce n'est pas le cas, verifiez le firewall (`firewall-cmd --list-all`) et `/etc/hosts`.

TD 2 - Configuration logicielle

TD 2 - Objectif

Contexte NovaTech : Les serveurs sont installes. Avant de deployer les applications, vous devez maitriser la gestion des paquets, comprendre les dependances, et mettre en place un depot local pour controler les mises a jour.

A la fin de ce TD, vous aurez :

- Inspecte et disseque des paquets RPM
- Analyse les dependances de binaires en production
- Cree un depot RPM local sur `srv-backup`
- Configure `srv-prod` pour utiliser ce depot

TD 2 - Etape 1 : Anatomie d'un RPM

1. Sur `srv-prod`, telechargez un RPM sans l'installer :

```
dnf download nginx
```

2. Inspectez-le :

```
file nginx-*.rpm
```

Resultat attendu : `RPM v3.0 bin i386/x86_64 nginx-1.xx.x-...` - on voit qu'il s'agit d'un binaire RPM avec l'architecture cible.

```
rpm -qpi nginx-*.rpm
```

Resultat attendu : Les metadonnees du paquet s'affichent (nom, version, release, architecture, taille, license, description, URL upstream, etc.)

TD 2 - Etape 1 : Anatomie (suite)

3. Listez les fichiers contenus dans le RPM :

```
rpm -qpl nginx-*.rpm
```

Resultat attendu : La liste des fichiers qui seront installes, par exemple :

```
/etc/nginx/nginx.conf
/usr/sbin/nginx
/usr/lib/systemd/system/nginx.service
/usr/share/nginx/html/index.html
...
```

4. Affichez les dependances requises :

```
rpm -qp --requires nginx-*.rpm
```

Resultat attendu : Une liste de bibliotheques (`libssl.so` , `libcrypto.so` , `libpthread.so.0` ...) et de paquets (`openssl-libs` , `pcre2` ...) necessaires au fonctionnement.

TD 2 - Etape 2 : Extraction sans installation

5. Extrayez le contenu du RPM sans l'installer :

```
mkdir /tmp/nginx-extract && cd /tmp/nginx-extract
rpm2cpio ~/nginx-*.rpm | cpio -idmv
```

Resultat attendu : L'arborescence est recreee localement :

```
./etc/nginx/
./usr/sbin/nginx
./usr/lib64/nginx/
./usr/share/nginx/
...
```

> C'est utile pour examiner un paquet avant installation, ou pour recuperer un fichier de configuration par defaut sans installer le paquet.

6. Revenez au repertoire home :

```
cd ~ && rm -rf /tmp/nginx-extract
```

TD 2 - Etape 3 : Dependances et bibliotheques

7. Installez `nginx` puis identifiez ses dependances dynamiques :

```
dnf install -y nginx
ldd /usr/sbin/nginx
```

Resultat attendu :

```
linux-vdso.so.1 (0x00007ffd...)
libdl.so.2 => /lib64/libdl.so.2 (0x00007f...)
libpthread.so.0 => /lib64/libpthread.so.0 (0x00007f...)
libcrypt.so.2 => /lib64/libcrypt.so.2 (0x00007f...)
libssl.so.3 => /lib64/libssl.so.3 (0x00007f...)
libcrypto.so.3 => /lib64/libcrypto.so.3 (0x00007f...)
libc.so.6 => /lib64/libc.so.6 (0x00007f...)
...
```

> `ldd` montre chaque bibliotheque dynamique et son chemin resolu. `linux-vdso.so.1` est un pseudo-fichier injecte par le noyau.

TD 2 - Etape 4 : Le linker dynamique

8. Examinez la configuration du linker :

```
cat /etc/ld.so.conf
```

Resultat attendu : Typiquement un `include ld.so.conf.d/*.conf` qui charge tous les fichiers du sous-repertoire.

```
ls /etc/ld.so.conf.d/
```

Resultat attendu : Des fichiers comme `mariadb-x86_64.conf`, etc. Chacun ajoute un chemin de recherche pour les bibliotheques.

9. Recherchez `libssl` dans le cache du linker :

```
ldconfig -p | grep libssl
```

Resultat attendu :

```
libssl.so.3 (libc6,x86-64) => /lib64/libssl.so.3
```

> C'est ce cache (mis a jour par `ldconfig`) que le systeme consulte au lancement de chaque programme pour trouver les bibliotheques.

TD 2 - Etape 5 : Creer un depot local

10. Sur `srv-backup`, installez les outils necessaires :

```
dnf install -y createrepo_c httpd
mkdir -p /var/www/html/repo/novatech
```

11. Telechargez des paquets utiles avec leurs dependances :

```
dnf download --resolve --destdir=/var/www/html/repo/novatech \
    nginx vim-enhanced tmux htop rsync
```

> L'option `--resolve` telechargee aussi toutes les dependances. Sans elle, seuls les paquets nommes seraient recuperes.

TD 2 - Etape 5 : Generer les metadonnees

12. Generez les metadonnees du depot :

```
createrepo_c /var/www/html/repo/novatech
```

Resultat attendu :

```
Directory walk started
Spawning worker 0 with 42 pkgs
Workers Finished
Saving Primary metadata
Saving file lists metadata
Saving other metadata
```

TD 2 - Etape 5 : Depot local (suite)

13. Demarrez Apache et ouvrez le firewall :

```
systemctl enable --now httpd
firewall-cmd --permanent --add-service=http && firewall-cmd --reload
```

14. Testez l'acces depuis `srv-backup` lui-meme :

```
curl -s http://localhost/repo/novatech/repodata/repomd.xml | head -5
```

Resultat attendu : Du XML valide s'affiche, confirmant que le depot est accessible.

TD 2 - Etape 6 : Configurer le client

15. Sur `srv-prod`, creez le fichier de configuration du depot :

```
cat > /etc/yum.repos.d/novatech.repo << 'EOF'
[novatech-local]
name=NovaTech Local Repository
baseurl=http://srv-backup.novatech.lan/repo/novatech
enabled=1
gpgcheck=0
priority=10
EOF
```

TD 2 - Etape 6 : Tester le depot

16. Testez :

```
dnf clean all
dnf repolist
```

Resultat attendu : Le depot `novatech-local` apparait dans la liste avec le nombre de paquets disponibles.

```
dnf install --repo=novatech-local tmux
```

Resultat attendu : `tmux` s'installe depuis le depot local (pas depuis Internet).

TD 2 - Etape 7 : Mises a jour de securite

17. Verifiez les mises a jour de securite disponibles :

```
dnf updateinfo list security
dnf updateinfo summary
```

Resultat attendu : Un tableau recapitulatif indiquant le nombre d'avis de securite par severite (Critical, Important, Moderate, Low).

18. Appliquez uniquement les correctifs de securite :

```
dnf update --security
```

> En production, on applique les correctifs de securite en priorite, et les mises a jour fonctionnelles apres validation sur un environnement de test. Le depot local permet de controler exactement quels paquets sont disponibles pour les serveurs.

TD 3 - Systemes de fichiers et stockage

TD 3 - Objectif

Contexte NovaTech : Le serveur applicatif va heberger des donnees critiques. Vous devez preparer le stockage de maniere reflechie : choisir le bon systeme de fichiers, gerer les volumes dynamiquement, et etre capable de recuperer des donnees en cas d'incident.

A la fin de ce TD, vous aurez :

- Compare ext4 et XFS sur des cas concrets
- Etendu un volume logique a chaud
- Cree et utilise des snapshots LVM
- Realise une recuperation de fichiers supprimes

TD 3 - Etape 1 : Creer des LV de test

1. Sur `srv-prod`, verifiez l'espace disponible dans le VG :

```
vgs
```

Resultat attendu :

```
  VG      #PV #LV #SN Attr   VSize   VFree
  vg_prod   1   4   0 wz--n- <19.00g <0.95g
```

> `VFree` indique l'espace non alloue. S'il n'est pas suffisant, reduisez un LV existant ou ajoutez un disque.

2. Creez deux LV de test :

```
lvcreate -L 200M -n lv_test_ext4 vg_prod
lvcreate -L 200M -n lv_test_xfs vg_prod
```

TD 3 - Etape 2 : Comparer ext4 et XFS

3. Formatez et montez :

```
mkfs.ext4 /dev/vg_prod/lv_test_ext4
mkfs.xfs /dev/vg_prod/lv_test_xfs
mkdir -p /mnt/test_{ext4,xfs}
mount /dev/vg_prod/lv_test_ext4 /mnt/test_ext4
mount /dev/vg_prod/lv_test_xfs /mnt/test_xfs
```

4. Generez 1000 petits fichiers sur chaque FS et comparez :

```
for i in $(seq 1 1000); do echo "data$i" > /mnt/test_ext4/file_$i; done
for i in $(seq 1 1000); do echo "data$i" > /mnt/test_xfs/file_$i; done
df -h /mnt/test_ext4 /mnt/test_xfs
df -i /mnt/test_ext4 /mnt/test_xfs
```

Resultat attendu : ext4 reserve plus d'espace pour ses structures (env. 5% par defaut) et pre-alloue un nombre fixe d'inodes. XFS alloue les inodes dynamiquement, donc `df -i` montre des valeurs tres differentes.

TD 3 - Etape 2 : Comparaison (suite)

5. Inspectez les parametres de chaque FS :

```
tune2fs -l /dev/vg_prod/lv_test_ext4 | grep -E "Block size|Inode count|Reserved"
```

Resultat attendu :

```
Inode count:              51200
Block size:               1024 ou 4096
Reserved block count:     XXX (5% par defaut)
```

```
xfs_info /dev/vg_prod/lv_test_xfs
```

Resultat attendu : Details sur les sections data, log, realtime, la taille de bloc (generalement 4096), et les allocation groups.

> **Synthese** : ext4 est fiable et polyvalent, redimensionnable dans les deux sens. XFS excelle sur les gros fichiers et les serveurs, mais **ne peut pas etre reduit** (uniquement agrandi). Pour `/var` en production (logs, donnees variables), les deux conviennent ; XFS est souvent le choix par defaut de Red Hat.

TD 3 - Etape 3 : Extension a chaud

6. Le volume `/var` de `srv-prod` se remplit. Etendez-le a chaud :

```
vgs   # Verifier l'espace libre
lvextend -L +1G /dev/vg_prod/lv_var
```

7. Redimensionnez le systeme de fichiers (la commande depend du FS) :

```
# Si /var est en ext4 :
resize2fs /dev/vg_prod/lv_var

# Si /var est en XFS :
xfs_growfs /var
```

TD 3 - Etape 3 : Verification de l'extension

8. Verifiez :

```
df -h /var
```

Resultat attendu : L'espace disponible a augmente de 1 Go, **sans aucune interruption de service**. Les processus ecrivant dans `/var` n'ont rien remarque.

> C'est l'un des avantages majeurs de LVM : l'extension a chaud sans demontage.

TD 3 - Etape 4 : Snapshots LVM

9. Creez quelques fichiers de test dans `/home/ntadmin/` :

```
mkdir -p /home/ntadmin/Documents
echo "Rapport Q1 NovaTech" > /home/ntadmin/Documents/rapport.txt
echo "Budget 2025" > /home/ntadmin/Documents/budget.txt
```

10. Creez un snapshot du volume `/home` :

```
lvcreate -L 500M -s -n snap_home /dev/vg_prod/lv_home
lvs
```

Resultat attendu : Le snapshot apparait dans la liste avec l'attribut `s` (snapshot) et un pourcentage d'utilisation (initialement ~0%) :

```
  snap_home vg_prod swi-a-s--- 500.00m      lv_home 0.00
```

TD 3 - Etape 4 : Snapshots (suite)

11. Simulez une erreur humaine :

```
rm -rf /home/ntadmin/Documents/
ls /home/ntadmin/Documents/
```

Resultat attendu : `ls: cannot access '/home/ntadmin/Documents/': No such file or directory` - les fichiers sont perdus.

12. Restaurez depuis le snapshot :

```
umount /home
lvconvert --merge /dev/vg_prod/snap_home
mount /home
```

> Si le merge ne peut pas se faire immediatement (volume en cours d'utilisation), il sera effectue au prochain `mount`. Un reboot peut etre necessaire si `/home` ne peut pas etre demonte.

13. Verifiez la restauration :

```
cat /home/ntadmin/Documents/rapport.txt
```

Resultat attendu : `Rapport Q1 NovaTech` - le fichier est restaure a l'etat du snapshot.

TD 3 - Etape 5 : Snapshot pour sauvegarde coherente

14. Utilisez un snapshot pour une sauvegarde coherente de `/var` vers `srv-backup` :

```
lvcreate -L 500M -s -n snap_var_backup /dev/vg_prod/lv_var
mkdir -p /mnt/snap_var
mount -o ro /dev/vg_prod/snap_var_backup /mnt/snap_var
```

> Le snapshot est monte en **lecture seule** (`ro`). Les donnees sont figees au moment de la creation du snapshot, meme si `/var` continue de recevoir des ecritures.

TD 3 - Etape 5 : Sauvegarde et nettoyage

15. Sauvegardez vers `srv-backup` :

```
rsync -avz /mnt/snap_var/ ntadmin@srv-backup:/home/ntadmin/backup-var/
```

16. Nettoyez :

```
umount /mnt/snap_var
lvremove -f /dev/vg_prod/snap_var_backup
```

> **Rappel** : Un snapshot n'est PAS une sauvegarde. Il reside sur le meme stockage physique. Si le disque tombe en panne, le snapshot est perdu aussi.

TD 3 - Etape 6 : Recuperation de donnees

17. Sur le LV de test ext4, creez puis supprimez un fichier :

```
echo "DONNEE CRITIQUE NOVATECH - NE PAS SUPPRIMER" > /mnt/test_ext4/important.txt
sync
rm /mnt/test_ext4/important.txt
```

18. Tentez une recuperation directe avec `grep` sur le device :

```
umount /mnt/test_ext4
grep -a "DONNEE CRITIQUE" /dev/vg_prod/lv_test_ext4
```

Resultat attendu : La chaine `DONNEE CRITIQUE NOVATECH - NE PAS SUPPRIMER` apparait. Le contenu du fichier est encore sur le disque, seule la reference dans le systeme de fichiers a ete supprimee.

> **Principe fondamental** : Quand un fichier est supprime, les blocs de donnees ne sont pas effaces immediatement. Ils sont marques comme libres. Tant qu'ils n'ont pas ete reecrits, les donnees sont recuperables. D'ou la regle : **ne plus ecrire sur le volume** des qu'une perte est detectee.

TD 3 - Etape 7 : Surveillance de l'espace disque

19. Creez un script de surveillance sur `srv-prod` :

```
cat > /usr/local/bin/check_disk_usage.sh << 'SCRIPT'
#!/bin/bash
SEUIL=80
df -h --output=pcent,target | tail -n +2 | while read usage mount; do
    pct=${usage%%%}
    if [ "$pct" -gt "$SEUIL" ]; then
        logger -t disk-alert "ALERTE: $mount a ${usage} d'utilisation"
    fi
done
SCRIPT
chmod +x /usr/local/bin/check_disk_usage.sh
```

TD 3 - Etape 7 : Test et planification

20. Testez-le et planifiez l'execution :

```
/usr/local/bin/check_disk_usage.sh
journalctl -t disk-alert    # Verifier si des alertes sont apparues
echo "0 * * * * root /usr/local/bin/check_disk_usage.sh" > /etc/cron.d/disk-check
```

21. Nettoyez les LV de test :

```
umount /mnt/test_ext4 /mnt/test_xfs 2>/dev/null
lvremove -f /dev/vg_prod/lv_test_ext4 /dev/vg_prod/lv_test_xfs
```

TD 4 - Noyau et peripheriques

TD 4 - Objectif

Contexte NovaTech : Le serveur de production doit supporter un peripherique USB specifique (un lecteur de code-barres pour l'entrepot) et vous devez ajuster des parametres noyau pour la production. Vous allez aussi explorer l'initramfs pour comprendre le processus de demarrage.

A la fin de ce TD, vous aurez :

- Realise un inventaire materiel complet
- Ecrit une regle udev personnalisee
- Explore et compris l'initramfs
- Ajuste des parametres noyau via sysctl

TD 4 - Etape 1 : Inventaire materiel

1. Sur `srv-prod`, realisez un inventaire complet :

```
lscpu
```

Resultat attendu : Architecture (x86_64), nombre de CPU, modele, flags supportes (vmx/svm pour la virtualisation, aes pour le chiffrement materiel, etc.)

```
lspci -nn
```

Resultat attendu : Chaque peripherique PCI avec ses identifiants vendeur:produit entre crochets, par ex. `[8086:100e]` pour une carte Intel.

```
lspci -k
```

Resultat attendu : Pour chaque peripherique PCI, le module noyau associe (`Kernel driver in use:`). C'est ainsi que l'on sait quel driver gere quel materiel.

TD 4 - Etape 1 : Inventaire (suite)

```
lsusb
```

Resultat attendu : La liste des peripheriques USB (au minimum le hub racine de la VM).

```
lsblk
```

Resultat attendu : L'arborescence des disques, partitions, RAID et LV. On voit la chaine complete : `sda` -> `sda2` -> `md0` -> `vg_prod-lv_root`.

2. Comptez les modules charges :

```
lsmod | wc -l
```

Resultat attendu : Typiquement entre 50 et 100 modules sur un serveur minimal.

TD 4 - Etape 2 : Gestion des modules

3. Choisissez un module (par exemple le beeper PC speaker) et pratiquez :

```
modinfo pcspkr
```

Resultat attendu : Description, licence (GPL), alias, dependances eventuelles du module.

4. Chargez/dechargez un module :

```
lsmod | grep pcspkr        # Est-il charge ?
modprobe pcspkr             # Charger
lsmod | grep pcspkr         # Verifier : il apparait
modprobe -r pcspkr          # Decharger
lsmod | grep pcspkr         # Verifier : il a disparu
```

5. Bloquez un module inutile de maniere persistante :

```
echo "blacklist pcspkr" > /etc/modprobe.d/blacklist-beep.conf
```

> Apres reboot, le module ne sera plus charge automatiquement. C'est utile pour les modules problematiques ou inutiles en production.

TD 4 - Etape 3 : Explorer /sys et udev

6. Explorez l'arborescence `/sys` pour un disque :

```
ls /sys/block/sda/
cat /sys/block/sda/size
cat /sys/block/sda/queue/scheduler
```

Resultat attendu : `size` donne la taille en secteurs de 512 octets. `scheduler` montre le planificateur d'I/O actif (entre crochets).

7. Listez les attributs d'un peripherique (ceux que udev peut utiliser dans ses regles) :

```
udevadm info --attribute-walk --name=/dev/sda | head -60
```

Resultat attendu : Une cascade d'attributs hierarchiques (ATTR{size}, ATTR{model}, SUBSYSTEM, DRIVER...) remontant du peripherique jusqu'au bus.

TD 4 - Etape 3 : udev monitor

8. Dans un premier terminal, lancez la surveillance udev :

```
udevadm monitor --property
```

9. Dans un second terminal (ou via l'hyperviseur), ajoutez un disque virtuel a chaud a la VM.

Resultat attendu : Une serie d'evenements `UDEV` et `KERNEL` s'affiche, montrant les attributs du nouveau peripherique (ACTION=add, SUBSYSTEM=block, DEVNAME=/dev/sdb, etc.)

> Ces attributs sont exactement ceux que l'on utilise dans les regles udev pour declencher des actions.

TD 4 - Etape 4 : Ecrire une regle udev

10. Creez une regle udev qui journalise l'insertion de tout peripherique de stockage bloc :

```
cat > /etc/udev/rules.d/99-novatech-usb.rules << 'EOF'
ACTION=="add", SUBSYSTEM=="block", ENV{ID_BUS}=="usb", \
    SYMLINK+="novatech-usb-%n", \
    RUN+="/usr/bin/logger -t udev-novatech 'Peripherique USB connecte: %k'"
ACTION=="remove", SUBSYSTEM=="block", ENV{ID_BUS}=="usb", \
    RUN+="/usr/bin/logger -t udev-novatech 'Peripherique USB deconnecte: %k'"
EOF
```

TD 4 - Etape 4 : Tester la regle udev

11. Rechargez les regles :

```
udevadm control --reload-rules
udevadm trigger
```

12. Branchez une cle USB (ou ajoutez un disque virtuel USB) et verifiez :

```
ls -la /dev/novatech-usb-*
journalctl -t udev-novatech --no-pager
```

Resultat attendu : Un symlink `/dev/novatech-usb-1` pointe vers le vrai device, et le journal contient l'evenement.

TD 4 - Etape 5 : Explorer l'initramfs

13. Identifiez et examinez l'initramfs :

```
ls -lh /boot/initramfs-$(uname -r).img
```

Resultat attendu : Un fichier de 30 a 80 Mo selon les modules inclus.

```
lsinitrd /boot/initramfs-$(uname -r).img | head -50
```

Resultat attendu : Une liste de fichiers (scripts, modules, binaires) necessaires au demarrage.

14. Verifiez que les modules pour votre RAID et LVM sont presents :

```
lsinitrd /boot/initramfs-$(uname -r).img | grep -E "md|raid|lvm|dm-"
```

Resultat attendu : Des fichiers comme `md-mod.ko`, `raid1.ko`, `dm-mod.ko`, `lvm` sont presents. Sans eux, le noyau ne pourrait pas assembler le RAID ni activer le LVM au boot, et le systeme ne demarrerait pas.

TD 4 - Etape 6 : Tuning noyau avec sysctl

15. Consultez les parametres actuels :

```
sysctl net.core.somaxconn
sysctl net.ipv4.tcp_max_syn_backlog
sysctl net.ipv4.conf.all.rp_filter
sysctl kernel.sysrq
```

Resultat attendu : Les valeurs par defaut, par ex. `somaxconn = 4096` (taille de la file d'attente TCP), `sysrq = 16` (touche magique partiellement activee).

TD 4 - Etape 6 : sysctl - Profil production

16. Appliquez un profil de tuning production :

```
cat > /etc/sysctl.d/99-novatech-prod.conf << 'EOF'
# NovaTech - Tuning production srv-prod

# File d'attente TCP : plus de connexions simultanees
net.core.somaxconn = 1024

# SYN backlog : protection contre les SYN floods
net.ipv4.tcp_max_syn_backlog = 2048

# Reverse path filtering : rejete les paquets IP incoherents
net.ipv4.conf.all.rp_filter = 1

# Ignorer les pings broadcast (prevention smurf attack)
net.ipv4.icmp_echo_ignore_broadcasts = 1

# Desactiver SysRq en production (securite)
kernel.sysrq = 0
EOF
```

TD 4 - Etape 6 : sysctl - Application et verification

17. Appliquez et verifiez :

```
sysctl --system
sysctl net.core.somaxconn
```

Resultat attendu : `net.core.somaxconn = 1024` - la valeur a change. Elle sera persistante apres reboot grace au fichier dans `/etc/sysctl.d/`.

TD 5 - Maintenance et metrologie

TD 5 - Objectif

Contexte NovaTech : Le serveur est en production. Vous devez mettre en place les outils de surveillance : journalisation centralisee, verification d'integrite des fichiers critiques, et suivi des metriques systeme.

A la fin de ce TD, vous aurez :

- Configure la journalisation centralisee (rsyslog)
- Mis en place AIDE pour la detection d'intrusion
- Installe et utilise les outils de metrologie (vmstat, iostat, sar)

TD 5 - Etape 1 : Centraliser les logs (serveur)

1. Sur `srv-backup`, configurez rsyslog pour recevoir les logs distants. Editez `/etc/rsyslog.conf` et decommentez :

```
module(load="imudp")
input(type="imudp" port="514")
module(load="imtcp")
input(type="imtcp" port="514")
```

TD 5 - Etape 1 : Template de separation par hote

2. Ajoutez une regle pour separer les logs par machine source :

```
cat > /etc/rsyslog.d/novatech-remote.conf << 'EOF'
template(name="RemoteLogs" type="string"
    string="/var/log/remote/%HOSTNAME%/%PROGRAMNAME%.log")
if $fromhost-ip != '127.0.0.1' then {
    action(type="omfile" dynaFile="RemoteLogs")
    stop
}
EOF
```

3. Redemarrez et ouvrez le firewall :

```
systemctl restart rsyslog
firewall-cmd --permanent --add-port=514/tcp --add-port=514/udp
firewall-cmd --reload
```

TD 5 - Etape 2 : Centraliser les logs (client)

4. Sur `srv-prod`, configurez l'envoi vers `srv-backup` :

```
cat > /etc/rsyslog.d/novatech-forward.conf << 'EOF'
*.* @@srv-backup.novatech.lan:514
EOF
```

> `@@` = TCP (fiable, avec acquittement). `@` = UDP (plus leger, sans garantie de livraison). En production, on prefere TCP.

5. Redemarrez et testez :

```
systemctl restart rsyslog
logger -t test-novatech "Message de test depuis srv-prod"
```

6. Sur `srv-backup`, verifiez la reception :

```
ls /var/log/remote/
cat /var/log/remote/srv-prod/test-novatech.log
```

Resultat attendu : Le repertoire `srv-prod` a ete cree automatiquement, et le message `Message de test depuis srv-prod` apparait dans le fichier log.

TD 5 - Etape 3 : AIDE - Initialisation

7. Sur `srv-prod`, installez AIDE :

```
dnf install -y aide
```

8. Examinez la configuration par defaut :

```
head -80 /etc/aide.conf
```

Resultat attendu : Des regles definissant quels fichiers surveiller et quels attributs verifier (hash, permissions, owner, size, mtime, etc.). Par exemple : `/usr/sbin` est surveille avec des checksums SHA256.

9. Initialisez la base de reference :

```
aide --init
```

Resultat attendu : Le scan prend 1-2 minutes. Il cree `/var/lib/aide/aide.db.new.gz`.

```
cp /var/lib/aide/aide.db.new.gz /var/lib/aide/aide.db.gz
```

> En production, cette base devrait etre copiee sur un support en lecture seule ou un serveur distant pour empecher un attaquant de la modifier.

TD 5 - Etape 4 : AIDE - Simuler des modifications

10. Simulez des modifications suspectes :

```
touch /usr/bin/curl                      # Modification du timestamp
chmod 777 /tmp                           # Changement de permissions
echo "10.0.0.99 malware.bad" >> /etc/hosts   # Ajout suspect
```

11. Lancez une verification :

```
aide --check
```

TD 5 - Etape 4 : AIDE - Analyser le rapport

Resultat attendu : AIDE detecte les trois modifications :

```
Changed entries:
---------------------------------------------------
f ... .C... : /usr/bin/curl          (mtime change)
d ...p.... : /tmp                    (permissions)
f ... ..S.. : /etc/hosts             (size, hash)
```

> AIDE detecte **que** quelque chose a change, mais ne dit pas si c'est legitime. C'est a l'administrateur d'analyser.

12. Annulez les modifications :

```
touch -r /usr/bin/wget /usr/bin/curl   # Restaurer timestamp
chmod 1777 /tmp                         # Restaurer permissions (sticky bit)
sed -i '/malware/d' /etc/hosts          # Supprimer la ligne ajoutee
```

TD 5 - Etape 5 : Automatiser AIDE

13. Creez un cron quotidien pour AIDE :

```
cat > /etc/cron.daily/aide-check << 'SCRIPT'
#!/bin/bash
RAPPORT="/var/log/aide/aide-$(date +%Y%m%d).log"
mkdir -p /var/log/aide
aide --check > "$RAPPORT" 2>&1
if [ $? -ne 0 ]; then
    logger -t aide-alert "AIDE a detecte des modifications - voir $RAPPORT"
fi
SCRIPT
chmod +x /etc/cron.daily/aide-check
```

> Grace a la centralisation des logs (TD 5 etape 1-2), l'alerte `aide-alert` sera aussi transmise a `srv-backup`.

14. Mettez a jour la base apres les modifications legitimers :

```
aide --update
cp /var/lib/aide/aide.db.new.gz /var/lib/aide/aide.db.gz
```

TD 5 - Etape 6 : Metrologie - Outils temps reel

15. Installez sysstat :

```
dnf install -y sysstat
systemctl enable --now sysstat
```

16. Observez la memoire :

```
free -h
```

Resultat attendu :

```
               total        used        free      shared  buff/cache   available
Mem:           1.8Gi       350Mi       1.0Gi       10Mi       450Mi       1.3Gi
Swap:          2.0Gi          0B       2.0Gi
```

> `available` est la metrique importante : c'est la memoire reellement disponible pour les applications (free + cache recuperable). Ne pas confondre `free` (strictement inutilisee) et `available`.

TD 5 - Etape 6 : Metrologie - Generer de la charge

17. Observez l'activite systeme sous charge :

```
# Terminal 1 : monitoring
vmstat 1

# Terminal 2 : generer de la charge CPU
dd if=/dev/urandom of=/dev/null bs=1M &

# Terminal 3 : generer de la charge I/O
dd if=/dev/zero of=/tmp/testfile bs=1M count=500
```

TD 5 - Etape 6 : Interpreter vmstat

Resultat attendu de `vmstat` sous charge :

```
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 1  1      0 800000  50000 300000    0    0     0 500000 1500 3000 50 20 10 20  0
```

- `r` : processus en attente de CPU (ici 1, normal)
- `b` : processus bloques en I/O (ici 1, a cause du dd)
- `us` : CPU utilisateur, `sy` : CPU systeme, `wa` : CPU en attente I/O
- `bo` : blocs ecrits par seconde (eleve a cause du dd)

18. Nettoyez la charge :

```
kill %1          # Stopper le dd en arriere-plan
rm /tmp/testfile
```

TD 5 - Etape 7 : Metrologie - iostat et sar

19. Observez les I/O disque :

```
iostat -x 1 5
```

Resultat attendu : Pour chaque disque, les metriques de lecture/ecriture, latence (`await`), saturation (`%util`). Un `%util` proche de 100% indique un goulot d'etranglement I/O.

20. Consultez l'historique avec sar (les donnees s'accumulent au fil des jours) :

```
sar -u          # Historique CPU
sar -r          # Historique memoire
sar -b          # Historique I/O
```

Resultat attendu : Des lignes horodatees montrant l'evolution des metriques. Si sysstat vient d'etre installe, seules les dernieres minutes sont disponibles.

> `sar` est l'outil de reference pour les analyses post-incident : "que s'est-il passe hier a 3h du matin ?"

TD 6 - Boot et depannage

TD 6 - Objectif

Contexte NovaTech : Un stagiaire a fait une fausse manipulation sur `srv-prod` : le mot de passe root est perdu, et apres un reboot force, le systeme ne demarre plus correctement. C'est a vous de reparer.

A la fin de ce TD, vous aurez :

- Sauvegarde et restaure le MBR / la configuration GRUB
- Recupere un mot de passe root perdu
- Repare un systeme qui ne boot plus
- Documente une procedure de reprise apres incident

TD 6 - Etape 1 : Sauvegarder le MBR

1. Sur `srv-prod`, sauvegardez le MBR complet (bootloader + table de partitions) :

```
dd if=/dev/sda of=/root/mbr-srv-prod.bin bs=512 count=1
```

2. Sauvegardez uniquement le code du bootloader (sans la table de partitions) :

```
dd if=/dev/sda of=/root/bootloader-srv-prod.bin bs=1 count=440
```

TD 6 - Etape 1 : Structure du MBR

> Les 440 premiers octets contiennent le code GRUB stage 1. Les octets 441-446 sont la signature disque. Les octets 447-510 contiennent la table de partitions. Les octets 511-512 sont la signature de boot `55 AA`.

3. Copiez sur `srv-backup` :

```
scp /root/*-srv-prod.bin ntadmin@srv-backup:/home/ntadmin/
```

TD 6 - Etape 1 : MBR (suite)

4. Examinez le contenu avec hexdump :

```
hexdump -C /root/mbr-srv-prod.bin | head -5
hexdump -C /root/mbr-srv-prod.bin | tail -5
```

Resultat attendu : Les derniers octets montrent la signature de boot :

```
000001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa
```

> `55 AA` en position 510-511 est la signature MBR. Sans elle, le BIOS considere le disque comme non bootable.

TD 6 - Etape 2 : Comprendre la configuration GRUB

5. Examinez la configuration :

```
cat /etc/default/grub
```

Resultat attendu :

```
GRUB_TIMEOUT=5
GRUB_DEFAULT=saved
GRUB_CMDLINE_LINUX="crashkernel=1G-4G:192M,4G-64G:256M rd.lvm.lv=vg_prod/lv_root ..."
```

TD 6 - Etape 2 : Entrees de boot et scripts

6. Listez les entrees de boot :

```
grubby --info=ALL
```

Resultat attendu : Chaque noyau installe avec son initramfs, ses parametres, et son index.

7. Identifiez les scripts de generation :

```
ls -la /etc/grub.d/
```

Resultat attendu : Des scripts numerotes (00_header, 10_linux, 40_custom...) qui sont executes dans l'ordre pour generer `/boot/grub2/grub.cfg`. On ne modifie **jamais** `grub.cfg` directement.

TD 6 - Etape 3 : Recuperer un mot de passe root perdu

8. Redemarrez `srv-prod`. Au menu GRUB :
   - Appuyez sur `e` pour editer (entrez le mot de passe GRUB du TD 1)
   - Trouvez la ligne commencant par `linux` ou `linuxefi`
   - Ajoutez `rd.break` a la fin de cette ligne
   - Appuyez sur `Ctrl+X` pour demarrer

TD 6 - Etape 3 : Shell d'urgence rd.break

9. Vous arrivez dans un shell d'urgence. Le vrai systeme est monte en lecture seule sur `/sysroot` :

```
mount -o remount,rw /sysroot
chroot /sysroot
passwd root
```

> Entrez le nouveau mot de passe deux fois.

```
touch /.autorelabel
exit
reboot
```

> `touch /.autorelabel` est **necessaire** si SELinux est actif (mode enforcing). Sans cela, le fichier `/etc/shadow` modifie aurait un contexte SELinux incorrect et le login echouerait. Le reboot avec relabel prend quelques minutes supplementaires.

TD 6 - Etape 4 : Reparer GRUB apres corruption

10. **Sur une VM de test** (pas `srv-prod` !), corrompez volontairement GRUB :

```
dd if=/dev/zero of=/dev/sda bs=1 count=440
reboot
```

Resultat attendu : La VM ne demarre plus. L'ecran reste noir ou affiche un message d'erreur du BIOS/UEFI.

11. Demarrez la VM sur l'ISO Rocky Linux et choisissez **Troubleshooting > Rescue a Rocky Linux system**.

12. L'installeur detecte le systeme installe et propose de le monter. Choisissez **1) Continue** :

```
chroot /mnt/sysimage
grub2-install /dev/sda
grub2-mkconfig -o /boot/grub2/grub.cfg
exit
reboot
```

Resultat attendu : Le systeme redemarre normalement. GRUB a ete reinstalle dans le MBR et sa configuration regeneree.

TD 6 - Etape 5 : Methode alternative avec init=/bin/bash

13. Au menu GRUB, editez l'entree et remplacez `rhgb quiet` par :

```
init=/bin/bash
```

14. Appuyez sur `Ctrl+X`. Vous obtenez un shell root minimal :

```
mount -o remount,rw /
passwd root
mount -o remount,ro /
reboot -f
```

Resultat attendu : Le mot de passe est change. Le `-f` force le reboot car `systemd` n'est pas en cours d'execution (init a ete remplace par bash).

> **Difference entre les deux methodes** :
>
> - `rd.break` : arrete le boot dans l'initramfs, avant le pivot vers le vrai root. Plus propre.
> - `init=/bin/bash` : remplace le processus init par un shell. Plus brutal, pas de services.
>
> Les deux demontrent pourquoi la protection physique du serveur et le chiffrement LUKS sont importants.

TD 6 - Etape 6 : Documenter la procedure de reprise

15. Redigez un document de procedure de reprise :

```
cat > /root/procedure-reprise-novatech.txt << 'EOF'
=== PROCEDURE DE REPRISE - NovaTech ===

1. SAUVEGARDE MBR
   - Frequence : apres chaque modification de GRUB ou noyau
   - Stockage : srv-backup:/home/ntadmin/
   - Commande : dd if=/dev/sda of=mbr.bin bs=512 count=1
```

TD 6 - Etape 6 : Procedure de reprise (suite)

```
2. PERTE MOT DE PASSE ROOT
   - Reboot > GRUB > e > ajouter rd.break > Ctrl+X
   - mount -o remount,rw /sysroot && chroot /sysroot
   - passwd root && touch /.autorelabel && reboot

3. GRUB CORROMPU
   - Boot ISO > Rescue > chroot /mnt/sysimage
   - grub2-install /dev/sda
   - grub2-mkconfig -o /boot/grub2/grub.cfg

4. ESCALADE
   - Niveau 1 : Administrateur systeme (vous)
   - Niveau 2 : Responsable infrastructure
   - Niveau 3 : Prestataire externe
EOF
```

> Ce document fait partie du **Plan de Reprise d'Activite (PRA)** de NovaTech.

TD 7 - Optimisation des performances

TD 7 - Objectif

Contexte NovaTech : Le site web heberge sur `srv-prod` est lent. La direction demande un audit de performances. Vous devez analyser methodiquement le systeme, identifier les goulots d'etranglement, et appliquer des optimisations mesurees.

A la fin de ce TD, vous aurez :

- Analyse la consommation memoire des processus
- Optimise les systemes de fichiers (noatime)
- Analyse et optimise le temps de boot
- Mis en place une baseline de performances

TD 7 - Etape 1 : Analyse memoire des processus

1. Assurez-vous que nginx est demarre :

```
dnf install -y nginx
systemctl enable --now nginx
```

2. Analysez la memoire avec smem :

```
dnf install -y smem
smem -t -k -P nginx
```

Resultat attendu :

```
  PID User     Command                         Swap      USS      PSS      RSS
 1234 root     nginx: master process              0     1.2M     1.5M     5.8M
 1235 nginx    nginx: worker process              0     1.8M     2.5M     6.2M
 1236 nginx    nginx: worker process              0     1.7M     2.4M     6.1M
                                                ---     -----    -----    -----
    3                                             0     4.7M     6.4M    18.1M
```

TD 7 - Etape 1 : Memoire (suite)

> **Comprendre les metriques** :
>
> - **RSS** (Resident Set Size) : memoire physique totale utilisee, **mais compte les bibliotheques partagees en double** pour chaque processus. Ici, les 3 processus nginx partagent `libssl`, `libc`, etc. Le RSS total (18 Mo) surestime la realite.
> - **PSS** (Proportional Set Size) : divise equitablement la memoire partagee entre les processus qui l'utilisent. C'est la metrique **la plus realiste**.
> - **USS** (Unique Set Size) : memoire utilisee **uniquement** par ce processus. C'est ce qui serait libere si on tuait ce processus.

3. Examinez les zones memoire d'un processus :

```
NGINX_PID=$(pgrep -o nginx)
cat /proc/$NGINX_PID/status | grep -E "Vm|Threads"
```

Resultat attendu :

```
VmPeak:    10240 kB    (pic de memoire virtuelle)
VmSize:     9800 kB    (memoire virtuelle actuelle)
VmRSS:      5800 kB    (memoire physique)
Threads:        1
```

TD 7 - Etape 2 : Optimisation filesystem - noatime

4. Verifiez les options de montage actuelles :

```
mount | grep vg_prod
```

Resultat attendu : Les volumes sont montes avec `relatime` (valeur par defaut) : le systeme met a jour le `atime` (access time) seulement si le fichier a ete modifie ou si le dernier access time date de plus de 24h.

TD 7 - Etape 2 : Appliquer noatime

5. Passez en `noatime` sur `/var` :

```
# Editez /etc/fstab, ajoutez noatime aux options du volume /var :
# /dev/vg_prod/lv_var  /var  ext4  defaults,noatime  0 2
```

```
mount -o remount,noatime /var
mount | grep lv_var
```

Resultat attendu : `noatime` apparait dans les options de montage. Chaque lecture de fichier dans `/var` n'entrainera plus d'ecriture sur le disque pour mettre a jour le timestamp d'acces.

> **Attention** : Certains outils dependent de `atime` (Mutt pour detecter les mails non lus, certains scripts de sauvegarde incrementale). Verifiez avant d'appliquer en production.

TD 7 - Etape 3 : Analyse du temps de boot

6. Analysez le temps de demarrage :

```
systemd-analyze
```

Resultat attendu :

```
Startup finished in 1.5s (kernel) + 2.8s (initramfs) + 12.4s (userspace) = 16.7s
```

7. Identifiez les services les plus lents :

```
systemd-analyze blame | head -15
```

Resultat attendu :

```
  5.2s  NetworkManager-wait-online.service
  2.1s  firewalld.service
  1.8s  kdump.service
  ...
```

TD 7 - Etape 3 : Boot - Chaine critique

8. Visualisez la chaine critique :

```
systemd-analyze critical-chain
```

Resultat attendu : Un arbre montrant les dependances bloquantes. Le chemin le plus long determine le temps total de boot.

9. Generez un graphique SVG :

```
systemd-analyze plot > /tmp/boot-analysis.svg
```

> Transferez ce fichier sur votre poste pour l'ouvrir dans un navigateur. Chaque barre horizontale represente un service, avec son temps de demarrage.

TD 7 - Etape 3 : Boot - Optimiser les services

10. Desactivez les services inutiles en production :

```
systemctl disable --now kdump.service         # Si pas besoin de crash dump
systemctl disable --now NetworkManager-wait-online.service  # Pas utile sur serveur IP fixe
```

11. Redemarrez et mesurez a nouveau :

```
reboot
# Apres reboot :
systemd-analyze
```

Resultat attendu : Le temps de boot a diminue de plusieurs secondes.

TD 7 - Etape 4 : Creer une baseline de performances

12. Creez un script qui collecte une baseline complete :

```
cat > /usr/local/bin/novatech-baseline.sh << 'SCRIPT'
#!/bin/bash
DIR="/var/log/novatech/baseline"
DATE=$(date +%Y%m%d-%H%M)
mkdir -p "$DIR"
OUT="$DIR/baseline-$DATE.txt"

echo "=== BASELINE NOVATECH - $DATE ===" > "$OUT"
echo -e "\n--- MEMOIRE ---" >> "$OUT"
free -h >> "$OUT"
echo -e "\n--- CPU (5 echantillons) ---" >> "$OUT"
vmstat 1 5 >> "$OUT"
echo -e "\n--- DISQUES (3 echantillons) ---" >> "$OUT"
iostat -x 1 3 >> "$OUT"
```

TD 7 - Etape 4 : Script baseline (suite)

```
echo -e "\n--- BOOT ---" >> "$OUT"
systemd-analyze >> "$OUT" 2>&1
echo -e "\n--- TOP 10 PROCESSUS (memoire) ---" >> "$OUT"
ps aux --sort=-%mem | head -11 >> "$OUT"
echo -e "\n--- ESPACE DISQUE ---" >> "$OUT"
df -h >> "$OUT"
echo -e "\n--- SERVICES ACTIFS ---" >> "$OUT"
systemctl list-units --type=service --state=running --no-pager >> "$OUT"

echo "Baseline enregistree : $OUT"
SCRIPT
chmod +x /usr/local/bin/novatech-baseline.sh
```

TD 7 - Etape 4 : Baseline (suite)

13. Executez la baseline :

```
/usr/local/bin/novatech-baseline.sh
cat /var/log/novatech/baseline/baseline-*.txt
```

Resultat attendu : Un rapport complet avec toutes les metriques du systeme a cet instant.

> **Principe d'optimisation** : On ne peut optimiser que ce que l'on mesure. Toujours :
>
> 1. Mesurer l'etat initial (baseline)
> 2. Modifier **un seul parametre** a la fois
> 3. Mesurer a nouveau
> 4. Comparer objectivement
>
> Sans cette methode, on ne sait jamais quelle modification a eu un effet positif (ou negatif).

TD 8 - Supervision avec Nagios

TD 8 - Objectif

Contexte NovaTech : L'infrastructure est en production. La derniere piece du puzzle : mettre en place une supervision proactive pour detecter les problemes avant les utilisateurs.

A la fin de ce TD, vous aurez :

- Installe et configure Nagios sur `srv-nagios`
- Supervise `srv-prod` et `srv-backup` (ping, disque, CPU, services)
- Configure NRPE pour les checks distants
- Mis en place des alertes et des templates

TD 8 - Etape 1 : Installer srv-nagios

1. Installez la VM `srv-nagios` (via Kickstart adapte ou manuellement) avec l'IP `192.168.56.12`.

2. Installez Nagios et ses plugins :

```
dnf install -y epel-release
dnf install -y nagios nagios-plugins-all nagios-plugins-nrpe httpd php
```

3. Configurez le mot de passe de l'interface web :

```
htpasswd -c /etc/nagios/passwd nagiosadmin
```

> Ce mot de passe protege l'acces a l'interface web Nagios. L'utilisateur par defaut est `nagiosadmin`.

TD 8 - Etape 1 : Demarrage et verification

4. Demarrez les services :

```
systemctl enable --now nagios httpd
firewall-cmd --permanent --add-service=http && firewall-cmd --reload
```

5. Testez l'acces : ouvrez `http://192.168.56.12/nagios/` dans un navigateur.

Resultat attendu : La page de login Nagios s'affiche. Connectez-vous avec `nagiosadmin` et le mot de passe choisi.

TD 8 - Etape 2 : Creer les templates

6. Creez des templates NovaTech pour eviter la repetition de parametres :

```
cat > /etc/nagios/objects/novatech-templates.cfg << 'EOF'
define host {
    name                    novatech-server
    use                     linux-server
    check_interval          5
    retry_interval          1
    max_check_attempts      3
    notification_interval   30
    register                0
}

define service {
    name                    novatech-service
    use                     generic-service
    check_interval          5
    retry_interval          1
    max_check_attempts      3
    register                0
}
EOF
```

> `register 0` signifie que ce sont des **templates** (modeles), pas des objets reels. Ils servent de base via la directive `use`.

TD 8 - Etape 3 : Declarer les hotes

7. Creez les definitions d'hotes :

```
cat > /etc/nagios/objects/novatech-hosts.cfg << 'EOF'
define hostgroup {
    hostgroup_name  novatech-servers
    alias           Serveurs NovaTech
    members         srv-prod,srv-backup
}

define host {
    use             novatech-server
    host_name       srv-prod
    alias           Serveur Production
    address         192.168.56.10
}

define host {
    use             novatech-server
    host_name       srv-backup
    alias           Serveur Backup
    address         192.168.56.11
}
EOF
```

> `use novatech-server` herite de tous les parametres du template defini a l'etape precedente. On ne redefinit que ce qui est specifique (nom, IP).

TD 8 - Etape 4 : Declarer les services de base

8. Ajoutez les checks reseaux (sans NRPE) :

```
cat > /etc/nagios/objects/novatech-services.cfg << 'EOF'
define service {
    use                 novatech-service
    hostgroup_name      novatech-servers
    service_description PING
    check_command       check_ping!200.0,20%!500.0,60%
}

define service {
    use                 novatech-service
    host_name           srv-prod
    service_description HTTP
    check_command       check_http
}

define service {
    use                 novatech-service
    host_name           srv-prod
    service_description SSH
    check_command       check_ssh
}
EOF
```

> `check_ping!200.0,20%!500.0,60%` signifie : WARNING si latence > 200ms ou > 20% de perte, CRITICAL si > 500ms ou > 60% de perte.

TD 8 - Etape 4 : Validation

9. Ajoutez les fichiers de configuration dans `nagios.cfg` :

```
# Editez /etc/nagios/nagios.cfg et ajoutez :
# cfg_file=/etc/nagios/objects/novatech-templates.cfg
# cfg_file=/etc/nagios/objects/novatech-hosts.cfg
# cfg_file=/etc/nagios/objects/novatech-services.cfg
```

10. Validez la configuration :

```
nagios -v /etc/nagios/nagios.cfg
```

Resultat attendu : A la fin de la sortie :

```
Total Warnings: 0
Total Errors:   0
Things look okay - No serious problems were detected during the pre-flight check
```

> Si des erreurs apparaissent, le message indique la ligne et le fichier concernes. Corrigez avant de continuer.

```
systemctl reload nagios
```

TD 8 - Etape 5 : NRPE - Installation

11. Sur `srv-prod` **et** `srv-backup`, installez NRPE :

```
dnf install -y epel-release
dnf install -y nrpe nagios-plugins-all
```

12. Configurez NRPE pour autoriser `srv-nagios`. Editez `/etc/nagios/nrpe.cfg` :

```
# Trouvez la ligne allowed_hosts et modifiez-la :
allowed_hosts=127.0.0.1,::1,192.168.56.12
```

TD 8 - Etape 5 : NRPE - Commandes locales

13. Definissez les commandes NRPE locales :

```
cat > /etc/nrpe.d/novatech.cfg << 'EOF'
command[check_disk_root]=/usr/lib64/nagios/plugins/check_disk -w 20% -c 10% -p /
command[check_disk_var]=/usr/lib64/nagios/plugins/check_disk -w 20% -c 10% -p /var
command[check_load]=/usr/lib64/nagios/plugins/check_load -w 2,1.5,1 -c 4,3,2
command[check_mem]=/usr/lib64/nagios/plugins/check_swap -w 30% -c 10%
command[check_zombie]=/usr/lib64/nagios/plugins/check_procs -w 5 -c 10 -s Z
command[check_total_procs]=/usr/lib64/nagios/plugins/check_procs -w 200 -c 300
EOF
```

TD 8 - Etape 5 : NRPE (suite)

> **Comment NRPE fonctionne** :
>
> 1. Nagios execute `check_nrpe -H srv-prod -c check_disk_root`
> 2. Le plugin contacte le daemon NRPE sur `srv-prod:5666`
> 3. NRPE execute localement `/usr/lib64/nagios/plugins/check_disk -w 20% -c 10% -p /`
> 4. Le resultat (code retour 0/1/2/3 + message) est renvoye a Nagios
>
> Codes retour : **0** = OK, **1** = WARNING, **2** = CRITICAL, **3** = UNKNOWN

14. Demarrez NRPE et ouvrez le firewall :

```
systemctl enable --now nrpe
firewall-cmd --permanent --add-port=5666/tcp && firewall-cmd --reload
```

15. Testez depuis `srv-nagios` :

```
/usr/lib64/nagios/plugins/check_nrpe -H 192.168.56.10
```

Resultat attendu : `NRPE v4.x.x` - confirme que NRPE repond.

```
/usr/lib64/nagios/plugins/check_nrpe -H 192.168.56.10 -c check_disk_root
```

Resultat attendu : `DISK OK - free space: / 8500 MB (85% inode=99%);`

TD 8 - Etape 6 : Services NRPE - Disques

16. Sur `srv-nagios`, ajoutez les services NRPE. Commencez par les disques :

```
cat >> /etc/nagios/objects/novatech-services.cfg << 'EOF'

define service {
    use                 novatech-service
    hostgroup_name      novatech-servers
    service_description Disk /
    check_command       check_nrpe!check_disk_root
}

define service {
    use                 novatech-service
    hostgroup_name      novatech-servers
    service_description Disk /var
    check_command       check_nrpe!check_disk_var
}
EOF
```

TD 8 - Etape 6 : Services NRPE - Systeme

Ajoutez les checks systeme :

```
cat >> /etc/nagios/objects/novatech-services.cfg << 'EOF'

define service {
    use                 novatech-service
    hostgroup_name      novatech-servers
    service_description CPU Load
    check_command       check_nrpe!check_load
}

define service {
    use                 novatech-service
    hostgroup_name      novatech-servers
    service_description Swap Usage
    check_command       check_nrpe!check_mem
}

define service {
    use                 novatech-service
    hostgroup_name      novatech-servers
    service_description Zombie Processes
    check_command       check_nrpe!check_zombie
}
EOF
```

TD 8 - Etape 6 : Validation NRPE

17. Validez et rechargez :

```
nagios -v /etc/nagios/nagios.cfg
systemctl reload nagios
```

18. Dans l'interface web Nagios, verifiez :

- **Hosts** : `srv-prod` et `srv-backup` sont en vert (UP)
- **Services** : Tous les services passent progressivement au vert (OK)

Resultat attendu dans l'interface :

Host	Service	Status
srv-prod	PING	OK
srv-prod	HTTP	OK
srv-prod	SSH	OK
srv-prod	Disk /	OK
srv-prod	Disk /var	OK
srv-prod	CPU Load	OK
srv-backup	PING	OK

TD 8 - Etape 7 : Simuler une panne de service

19. Simulez un arret de service sur `srv-prod` :

```
# Sur srv-prod :
systemctl stop nginx
```

20. Attendez le prochain check (~5 min) ou forcez-le dans l'interface Nagios (Re-schedule check).

Resultat attendu : Le service HTTP passe en **CRITICAL** (rouge) avec le message `Connection refused`.

TD 8 - Etape 7 : Simuler un disque plein

21. Simulez un disque plein :

```
# Sur srv-prod :
dd if=/dev/zero of=/var/tmp/fill_disk bs=1M count=4000
```

Resultat attendu : Apres le prochain check, le service `Disk /var` passe en **WARNING** ou **CRITICAL** selon le seuil atteint.

22. Nettoyez et verifiez le retour a la normale :

```
rm /var/tmp/fill_disk
systemctl start nginx
```

Resultat attendu : Apres quelques minutes, tous les services repassent en **OK** (vert).

Synthese du projet NovaTech

Recapitulatif de l'infrastructure

A l'issue des 8 TD, vous avez construit une infrastructure complete :

Composant	TD	Ce qui a ete fait
Installation reproductible	TD 1	Kickstart + RAID1 + LVM
Gestion logicielle	TD 2	Depot local + politique MaJ
Stockage resilient	TD 3	Snapshots + sauvegardes + recuperation
Tuning systeme	TD 4	udev + sysctl + modules
Surveillance	TD 5	Logs centralises + AIDE + metrologie
Plan de reprise	TD 6	MBR + GRUB + procedures
Performances	TD 7	Baseline + optimisations mesurees
Supervision	TD 8	Nagios + NRPE + alerting

Competences acquises

- **Deployer** : Installation automatisee et reproductible
- **Stocker** : Choix de FS eclaire, LVM dynamique, snapshots
- **Comprendre** : Le noyau, les modules, le boot, l'initramfs
- **Securiser** : GRUB, AIDE, sysctl, logs centralises
- **Diagnostiquer** : vmstat, iostat, sar, smem, systemd-analyze
- **Reparer** : Recuperation GRUB, mot de passe, fichiers
- **Optimiser** : Mesure avant/apres, approche scientifique
- **Superviser** : Nagios, NRPE, alerting proactif

Pour aller plus loin

- **Automatisation** : Ansible pour deployer la configuration sur N serveurs
- **Conteneurisation** : Docker/Podman sur `srv-prod`
- **Haute disponibilite** : Cluster Pacemaker/Corosync
- **Chiffrement** : LUKS pour les disques, Let's Encrypt pour HTTPS
- **Monitoring moderne** : Prometheus + Grafana en complement de Nagios

> L'infrastructure NovaTech est une base solide. A vous de la faire evoluer !

Checklist globale du projet

Progression

- [ ] **TD 1** : 2 serveurs installes, RAID+LVM, GRUB securise
- [ ] **TD 2** : Depot local, politique de MaJ
- [ ] **TD 3** : Snapshots, sauvegarde, recuperation
- [ ] **TD 4** : Inventaire, udev, sysctl
- [ ] **TD 5** : Logs centralises, AIDE, metrologie
- [ ] **TD 6** : Procedures de reprise documentees
- [ ] **TD 7** : Audit de performances avec baseline
- [ ] **TD 8** : Supervision complete Nagios

> Chaque TD valide = une brique de plus dans l'infrastructure de production.