UTOPIOS®
TRANSFORMATION DIGITALE INTELLIGENTE

# Formation PCA/PRA

## Outils Open Source

# Programme de la Formation

- Introduction : Pourquoi l'open source
- Module 1 : Sauvegarde et restauration
  - BorgBackup
  - Restic
  - Bareos
  - Proxmox Backup Server
- Module 2 : Haute disponibilite
  - DRBD
  - Pacemaker / Corosync
  - HAProxy / Keepalived
- Module 3 : Detection et reponse
  - Wazuh (SIEM/XDR)
  - ELK / OpenSearch
  - Velociraptor
  - Outils forensic
- Module 4 : Supervision et documentation
  - Prometheus / Grafana
  - Uptime Kuma
  - Wiki.js / Netbox
- TP Final : Infrastructure complete

UTOPIOS®
TRANSFORMATION DIGITALE INTELLIGENTE

# Introduction

## Pourquoi l'open source pour le PCA/PRA

# Avantages de l'open source

**Cout**

- Pas de licence a payer
- Reduction du TCO (Total Cost of Ownership)
- Budget reallouable vers expertise et formation

**Flexibilite**

- Personnalisation possible du code
- Integration facilitee via APIs ouvertes
- Pas de vendor lock-in

**Transparence**

- Code auditable (securite)
- Communaute active pour les corrections
- Documentation communautaire abondante

# Inconvenients a considerer

**Support**

- Pas de support editeur garanti (sauf offres commerciales)
- Dependance aux competences internes
- Communaute parfois lente a repondre

**Maturite variable**

- Certains projets instables ou abandonnes
- Documentation parfois insuffisante
- Interface utilisateur souvent spartiate

**Competences**

- Necessite des equipes formees
- Courbe d'apprentissage parfois raide
- Integration a faire soi-meme

# Criteres de choix d'un outil

| Critere | Questions a se poser |
|---|---|
| Activite projet | Derniere release ? Frequence commits ? |
| Communaute | Taille ? Reactivite sur issues ? |
| Documentation | Complete ? A jour ? Tutoriels ? |
| Securite | CVE recentes ? Politique disclosure ? |
| Integration | APIs ? Plugins ? Compatibilite ? |
| Offre commerciale | Support pro disponible si besoin ? |

**Conseil** : Verifier le projet sur GitHub avant adoption

UTOPIOS®
TRANSFORMATION DIGITALE INTELLIGENTE

# Module 1

## Sauvegarde et Restauration

# Les fondamentaux - Regle 3-2-1-1-0

| Chiffre | Signification |
|---|---|
| 3 | 3 copies des donnees (prod + 2 sauvegardes) |
| 2 | Sur 2 types de supports differents |
| 1 | 1 copie hors site (ou cloud) |
| 1 | 1 copie offline/immuable (air-gapped) |
| 0 | 0 erreur lors des tests de restauration |

**Point cle**

Une sauvegarde non testee n'est pas une sauvegarde.
Planifier des tests de restauration reguliers (mensuel minimum).

# Types de sauvegarde

**Sauvegarde complete (Full)**

- Copie integrale des donnees
- Restauration simple et rapide
- Consomme beaucoup d'espace et de temps

**Sauvegarde incrementale**

- Uniquement les modifications depuis derniere sauvegarde
- Rapide et econome en espace
- Restauration plus complexe (chaine de dependances)

**Sauvegarde differentielle**

- Modifications depuis derniere sauvegarde complete
- Compromis entre les deux approches
- Restauration : full + derniere differentielle

# RTO et RPO - Rappel

**RPO (Recovery Point Objective)**

- Perte de donnees maximale acceptable
- Determine la frequence des sauvegardes
- Exemple : RPO 4h = sauvegarde toutes les 4h minimum

**RTO (Recovery Time Objective)**

- Temps maximal de restauration acceptable
- Influence le choix de la solution
- Exemple : RTO 2h = restauration complete en moins de 2h

**Impact sur le choix des outils**

- RPO court : replication temps reel, sauvegardes frequentes
- RTO court : infrastructure de secours, restauration automatisee

# BorgBackup - Presentation

Solution de sauvegarde avec deduplication et compression.

**Points forts**

- Deduplication au niveau bloc (tres efficace)
- Chiffrement AES-256 cote client
- Compression (lz4, zstd, lzma)
- Verification d'integrite
- Montage FUSE des archives

**Cas d'usage**

- Sauvegarde de serveurs fichiers
- Sauvegarde de bases de donnees (dumps)
- Sauvegarde de configurations

# BorgBackup - Installation

```
# Debian/Ubuntu
sudo apt update
sudo apt install borgbackup

# CentOS/RHEL
sudo yum install epel-release
sudo yum install borgbackup

# Via pip (version recente)
pip install borgbackup
```

### Initialisation d'un depot

```
# Depot local chiffre
borg init --encryption=repokey /chemin/vers/backup

# Depot distant (SSH)
borg init --encryption=repokey ssh://user@serveur/chemin/repo
```

# BorgBackup - Creer une sauvegarde

```
# Sauvegarde basique
borg create /backup::archive-{now} /donnees/a/sauvegarder

# Sauvegarde avec exclusions et stats
borg create --stats --progress \
    --exclude '*.tmp' \
    --exclude '/home/*/.cache' \
    /backup::srv-{hostname}-{now:%Y-%m-%d} \
    /etc /home /var/www

# Sauvegarde distante
borg create ssh://backup@serveur/repo::archive-{now} /donnees
```

**Bonne pratique** : Nommer les archives avec date et hostname

# BorgBackup - Restauration

```
# Lister les archives
borg list /backup

# Voir le contenu d'une archive
borg list /backup::archive-2024-01-15

# Restaurer une archive complete
cd /restore
borg extract /backup::archive-2024-01-15

# Restaurer un fichier specifique
borg extract /backup::archive-2024-01-15 home/user/document.pdf

# Monter une archive (lecture seule)
mkdir /mnt/backup
borg mount /backup::archive-2024-01-15 /mnt/backup
```

# BorgBackup - Politique de retention

```
# Appliquer une politique de retention
borg prune --stats \
    --keep-daily=7 \
    --keep-weekly=4 \
    --keep-monthly=6 \
    /backup

# Liberer l'espace disque
borg compact /backup
```

**Options de retention**

| Option | Description |
|---|---|
| --keep-daily | Nombre de sauvegardes journalieres |
| --keep-weekly | Nombre de sauvegardes hebdomadaires |
| --keep-monthly | Nombre de sauvegardes mensuelles |
| --keep-yearly | Nombre de sauvegardes annuelles |

# BorgBackup - Script automatise

```
#!/bin/bash
# /usr/local/bin/backup-borg.sh

set -e

export BORG_REPO="ssh://backup@nas.local/volume1/backups/serveur-web"
export BORG_PASSPHRASE="MotDePasseComplexe"

BACKUP_NAME="backup-{hostname}-{now:%Y-%m-%d_%H%M%S}"

# Creer la sauvegarde
borg create --stats --compression zstd \
    --exclude '/var/cache/*' \
    --exclude '*.log' \
    ::$BACKUP_NAME /etc /home /var/www

# Appliquer la retention
borg prune --stats --keep-daily=7 --keep-weekly=4 --keep-monthly=12

# Verifier l'integrite
borg check --last 1
```

Crontab : `0 2 * * * /usr/local/bin/backup-borg.sh`

# Restic - Presentation

Solution moderne de sauvegarde multi-destinations.

**Points forts**

- Multi-destinations natives (local, SFTP, S3, Azure, GCS, Backblaze)
- Deduplication et chiffrement par defaut
- Tres rapide
- Pas de serveur requis

**Cas d'usage**

- Sauvegarde vers le cloud (S3, Backblaze B2)
- Environnements multi-cloud
- Sauvegardes postes de travail

# Restic - Installation et configuration

```
# Installation
sudo apt install restic

# Initialiser un depot local
restic init --repo /backup/restic-repo

# Initialiser vers S3
export AWS_ACCESS_KEY_ID="votre_access_key"
export AWS_SECRET_ACCESS_KEY="votre_secret_key"
restic init --repo s3:s3.amazonaws.com/bucket-backup

# Initialiser vers Backblaze B2
export B2_ACCOUNT_ID="votre_account_id"
export B2_ACCOUNT_KEY="votre_account_key"
restic init --repo b2:nom-du-bucket
```

# Restic - Utilisation

```
# Definir les variables
export RESTIC_PASSWORD="MotDePasseComplexe"
export RESTIC_REPOSITORY="/backup/restic-repo"

# Creer une sauvegarde
restic backup /donnees/importantes

# Avec exclusions
restic backup --exclude="*.tmp" --exclude-caches /home

# Lister les snapshots
restic snapshots

# Restaurer
restic restore latest --target /restore
```

### Retention

```
restic forget --keep-daily 7 --keep-weekly 4 --keep-monthly 12 --prune
```

# Bareos - Presentation

Fork de Bacula, solution entreprise complete.

**Architecture**

- Director : Orchestration des sauvegardes
- Storage Daemon : Gestion du stockage
- File Daemon : Agent sur les machines
- Catalog : Base de donnees (PostgreSQL/MySQL)

**Points forts**

- Solution entreprise complete
- Interface web (Bareos WebUI)
- Gestion des bandes magnetiques
- Plugins (MySQL, PostgreSQL, VMware)

**Cas d'usage** : Sauvegarde centralisee multi-serveurs

# Proxmox Backup Server

Solution dediee a la sauvegarde des environnements Proxmox VE.

**Points forts**

- Integration native avec Proxmox VE
- Deduplication et chiffrement
- Sauvegarde incrementale efficace
- Interface web complete
- Verification integrite automatique

**Installation**

```
# Ajouter le depot
echo "deb http://download.proxmox.com/debian/pbs bookworm pbs-no-subscription" \
    > /etc/apt/sources.list.d/pbs.list
apt update && apt install proxmox-backup-server
```

# Comparatif solutions sauvegarde

| Critere | BorgBackup | Restic | Bareos | PBS |
|---|---|---|---|---|
| Deduplication | Excellente | Excellente | Basique | Excellente |
| Chiffrement | AES-256 | AES-256 | Optionnel | AES-256 |
| Multi-destination | SFTP | S3/Cloud | Disque/bande | Local |
| Interface web | Non | Non | Oui | Oui |
| Complexite | Faible | Faible | Elevee | Moyenne |
| Cas d'usage | Serveurs Linux | Multi-cloud | Entreprise | Proxmox |

**Recommandations**

- Nouveau projet Linux : BorgBackup ou Restic
- Besoin cloud : Restic
- Environnement Proxmox : PBS

UTOPIOS®
TRANSFORMATION DIGITALE INTELLIGENTE

# Module 2

## Haute Disponibilite et Replication

# Concepts fondamentaux

**Actif/Passif**

- Un noeud actif, un ou plusieurs en attente
- Basculement en cas de panne du noeud actif
- Ressources du passif non utilisees en temps normal

**Actif/Actif**

- Tous les noeuds traitent des requetes
- Repartition de charge
- Plus complexe a mettre en oeuvre

**SPOF (Single Point of Failure)**

- Element dont la panne entraine l'arret du service
- Objectif : eliminer tous les SPOF

# Metriques de disponibilite

| Disponibilite | Indisponibilite/an | Indisponibilite/mois |
|---|---|---|
| 99% | 3,65 jours | 7,3 heures |
| 99,9% | 8,76 heures | 43,8 minutes |
| 99,99% | 52,6 minutes | 4,38 minutes |
| 99,999% | 5,26 minutes | 26,3 secondes |

**Comment atteindre 99,9% et plus ?**

- Redondance des composants
- Basculement automatique
- Monitoring et alerting
- Procedures testees

# DRBD - Presentation

DRBD replique les donnees au niveau bloc entre serveurs.
Souvent decrit comme "RAID 1 sur le reseau".

**Modes de replication**

- Protocol A : Asynchrone (performance)
- Protocol B : Semi-synchrone
- Protocol C : Synchrone (securite maximale)

**Cas d'usage**

- Cluster de bases de donnees
- Stockage partage pour cluster HA
- Replication temps reel entre sites

# DRBD - Configuration

```
# /etc/drbd.d/data.res
resource data {
    protocol C;

    on serveur1 {
        device    /dev/drbd0;
        disk      /dev/sdb1;
        address   192.168.1.10:7789;
        meta-disk internal;
    }

    on serveur2 {
        device    /dev/drbd0;
        disk      /dev/sdb1;
        address   192.168.1.11:7789;
        meta-disk internal;
    }
}
```

```
# Initialisation
drbdadm create-md data && drbdadm up data
drbdadm primary --force data  # Sur le primaire
```

# Pacemaker et Corosync

**Corosync** : Couche de communication du cluster
**Pacemaker** : Gestionnaire de ressources du cluster

### Installation

```
apt install pacemaker corosync pcs
systemctl enable --now pcsd
passwd hacluster
```

### Creation du cluster

```
pcs host auth serveur1 serveur2 -u hacluster
pcs cluster setup moncluster serveur1 serveur2
pcs cluster start --all
pcs cluster enable --all
```

# Pacemaker - Ressources

### IP virtuelle (VIP)

```
pcs resource create VIP ocf:heartbeat:IPaddr2 \
    ip=192.168.1.100 cidr_netmask=24 \
    op monitor interval=30s
```

### Service

```
pcs resource create WebServer systemd:nginx \
    op monitor interval=10s
```

### Contraintes

```
# Colocalisation
pcs constraint colocation add WebServer with VIP INFINITY

# Ordre de demarrage
pcs constraint order VIP then WebServer
```

# HAProxy - Presentation

Load balancer et reverse proxy haute performance.

**Cas d'usage**

- Repartition de charge entre serveurs web
- Terminaison SSL
- Basculement automatique (health checks)
- Point d'entree unique pour un cluster

**Installation**

```
apt install haproxy
```

**Algorithmes de repartition**

- roundrobin : Tour a tour
- leastconn : Moins de connexions
- source : Hash IP source (affinite session)

# HAProxy - Configuration

```
# /etc/haproxy/haproxy.cfg

frontend http_front
    bind *:80
    default_backend http_back

backend http_back
    balance roundrobin
    option httpchk GET /health
    http-check expect status 200
    server web1 192.168.1.11:80 check
    server web2 192.168.1.12:80 check
    server web3 192.168.1.13:80 check backup

listen stats
    bind *:8080
    stats enable
    stats uri /stats
    stats auth admin:MotDePasse
```

# Keepalived - VIP entre HAProxy

```
# /etc/keepalived/keepalived.conf - MASTER

vrrp_script chk_haproxy {
    script "/usr/bin/killall -0 haproxy"
    interval 2
    weight 2
}

vrrp_instance VI_1 {
    state MASTER
    interface eth0
    virtual_router_id 51
    priority 101

    authentication {
        auth_type PASS
        auth_pass secret123
    }

    virtual_ipaddress {
        192.168.1.100/24
    }

    track_script { chk_haproxy }
}
```

# HA Bases de donnees - Patroni

Patroni automatise la gestion d'un cluster PostgreSQL.

**Fonctionnalites**

- Failover automatique
- Election de leader
- Gestion des replicas
- Integration etcd/Consul/ZooKeeper

**Verification**

```
patronictl -c /etc/patroni/config.yml list
```

| Membre | Role | Etat |
|---|---|---|
| node1 | Leader | running |
| node2 | Replica | streaming |
| node3 | Replica | streaming |

# HA Bases de donnees - Galera

Cluster multi-maitre synchrone pour MariaDB/MySQL.

**Caracteristiques**

- Tous les noeuds en lecture/ecriture
- Replication synchrone
- Pas de perte de donnees

**Configuration**

```
# /etc/mysql/mariadb.conf.d/60-galera.cnf
[galera]
wsrep_on = ON
wsrep_cluster_name = "mon_cluster"
wsrep_cluster_address = "gcomm://192.168.1.10,192.168.1.11,192.168.1.12"
wsrep_node_address = "192.168.1.10"
wsrep_sst_method = mariabackup
```

# Architecture HA type

UTOPIOS®
TRANSFORMATION DIGITALE INTELLIGENTE

# Module 3

## Detection, SIEM et Reponse a Incident

# Wazuh - Presentation

Plateforme de securite open source complete.

**Fonctionnalites**

- SIEM : Collecte et analyse des logs
- XDR : Detection et reponse etendue
- FIM : Surveillance integrite fichiers
- Vulnerability detection : Scan de vulnerabilites
- Compliance : Conformite (PCI-DSS, GDPR, HIPAA)

**Architecture**

- Wazuh Manager : Analyse des evenements
- Wazuh Indexer : Stockage (OpenSearch)
- Wazuh Dashboard : Interface web
- Wazuh Agent : Collecte sur endpoints

# Wazuh - Installation

## Installation all-in-one

```
curl -sO https://packages.wazuh.com/4.7/wazuh-install.sh
chmod +x wazuh-install.sh
./wazuh-install.sh -a
```

## Installation agent

```
apt install wazuh-agent

# Configuration
# /var/ossec/etc/ossec.conf
<client>
  <server>
    <address>192.168.1.50</address>
  </server>
</client>

# Enregistrement
/var/ossec/bin/agent-auth -m 192.168.1.50
systemctl enable --now wazuh-agent
```

# Wazuh - FIM (File Integrity Monitoring)

```
<!-- /var/ossec/etc/ossec.conf sur l'agent -->
<syscheck>
  <disabled>no</disabled>
  <frequency>43200</frequency>
  <scan_on_start>yes</scan_on_start>

  <!-- Repertoires a surveiller -->
  <directories check_all="yes" realtime="yes">/etc</directories>
  <directories check_all="yes" realtime="yes">/bin</directories>
  <directories check_all="yes">/var/www</directories>

  <!-- Fichiers a ignorer -->
  <ignore>/etc/mtab</ignore>
  <ignore type="sregex">.log$</ignore>
</syscheck>
```

**Alertes** : Modification, creation, suppression de fichiers

# Wazuh - Regles personnalisees

```
<!-- /var/ossec/etc/rules/local_rules.xml -->
<group name="custom,">

  <!-- Alerte sur tentatives SSH echouees multiples -->
  <rule id="100001" level="10">
    <if_matched_sid>5710</if_matched_sid>
    <same_source_ip />
    <description>Multiple SSH failed logins from same IP</description>
  </rule>

  <!-- Alerte sur modification fichier critique -->
  <rule id="100002" level="12">
    <if_sid>550</if_sid>
    <match>/etc/passwd|/etc/shadow|/etc/sudoers</match>
    <description>Critical system file modified</description>
  </rule>

</group>
```

# ELK / OpenSearch

**ELK Stack**

- Elasticsearch : Moteur de recherche et stockage
- Logstash : Collecte et transformation des logs
- Kibana : Interface de visualisation

**OpenSearch** : Fork open source d'Elasticsearch (licence Apache 2.0)

**Installation Docker**

```
docker run -d --name opensearch \
    -p 9200:9200 \
    -e discovery.type=single-node \
    opensearchproject/opensearch:latest
```

**Cas d'usage** : Centralisation logs, analyse, dashboards

# Filebeat - Collecte de logs

```
# /etc/filebeat/filebeat.yml
filebeat.inputs:
  - type: log
    enabled: true
    paths:
      - /var/log/auth.log
      - /var/log/syslog
    fields:
      type: syslog

  - type: log
    enabled: true
    paths:
      - /var/log/nginx/access.log
    fields:
      type: nginx-access

output.elasticsearch:
  hosts: ["https://192.168.1.50:9200"]
  username: "admin"
  password: "Admin123!"
```

# Velociraptor - Presentation

Outil de forensic et hunting endpoint.

**Fonctionnalites**

- Collecte d'artefacts a distance
- Requetes en temps reel sur les endpoints
- Hunting sur tout le parc
- Reponse a incident

**Installation**

```
wget https://github.com/Velocidex/velociraptor/releases/latest
chmod +x velociraptor
velociraptor config generate -i
velociraptor frontend -v
```

# Velociraptor - Requetes VQL

```
-- Lister les processus
SELECT Pid, Name, Exe, CommandLine
FROM pslist()

-- Connexions reseau etablies
SELECT Pid, Name, Laddr, Raddr, Status
FROM netstat()
WHERE Status = 'ESTABLISHED'

-- Fichiers modifies recemment (24h)
SELECT FullPath, Mtime, Size
FROM glob(globs='/etc/**')
WHERE Mtime > now() - 86400

-- Recherche dans les logs Windows
SELECT * FROM parse_evtx(filename='Security.evtx')
WHERE EventID = 4625  -- Echecs authentification
```

# TheHive - Gestion d'incidents

Plateforme de gestion d'incidents de securite.

**Workflow**

1. Alerte recue (Wazuh, SIEM, email)
2. Creation d'un Case dans TheHive
3. Ajout des Observables (IP, hash, domaine)
4. Analyse automatique via Cortex
5. Investigation manuelle
6. Actions de remediation
7. Cloture et rapport

**Outils associes**

- Cortex : Analyseurs automatises
- MISP : Partage d'indicateurs de compromission

# Outils forensic - Acquisition

### Image disque

```
# Copie avec hash pour integrite
dd if=/dev/sda | tee /evidence/disk.img | sha256sum > hash.txt

# Version forensic
dcfldd if=/dev/sda of=/evidence/disk.img hash=sha256
```

### Capture memoire Linux

```
./avml /evidence/memory.lime
```

### Capture memoire Windows

```
winpmem_mini_x64.exe memory.raw
```

# Outils forensic - Analyse

## Volatility 3 - Analyse memoire

```
# Lister les processus
vol -f memory.lime linux.pslist.PsList

# Connexions reseau
vol -f memory.lime linux.sockstat.Sockstat

# Processus caches/malveillants
vol -f memory.lime linux.malfind.Malfind
```

## Autopsy - Analyse disque

```
apt install autopsy
autopsy  # Interface web sur port 9999
```

UTOPIOS®
TRANSFORMATION DIGITALE INTELLIGENTE

# Module 4

## Supervision, Alerting et Documentation

# Prometheus - Presentation

Systeme de monitoring base sur les metriques.

**Caracteristiques**

- Modele pull (scraping des endpoints)
- Base de donnees time-series
- Langage de requete PromQL
- Alerting integre

**Installation**

```
wget https://github.com/prometheus/prometheus/releases/latest
tar xvfz prometheus-*.tar.gz
./prometheus --config.file=prometheus.yml
```

**Port par defaut** : 9090

# Prometheus - Configuration

```yaml
# prometheus.yml
global:
  scrape_interval: 15s

alerting:
  alertmanagers:
    - static_configs:
        - targets: ['localhost:9093']

rule_files:
  - "alerts.yml"

scrape_configs:
  - job_name: 'prometheus'
    static_configs:
      - targets: ['localhost:9090']

  - job_name: 'node'
    static_configs:
      - targets: ['192.168.1.10:9100', '192.168.1.11:9100']
```

# Node Exporter - Metriques systeme

### Installation

```
wget https://github.com/prometheus/node_exporter/releases/latest
tar xvfz node_exporter-*.tar.gz
./node_exporter &
```

### Metriques collectees

- CPU : node_cpu_seconds_total
- Memoire : node_memory_MemAvailable_bytes
- Disque : node_filesystem_avail_bytes
- Reseau : node_network_receive_bytes_total

### Service systemd

```
systemctl enable --now node_exporter
```

# Prometheus - Regles d'alerte

```yaml
# alerts.yml
groups:
  - name: availability
    rules:
      - alert: InstanceDown
        expr: up == 0
        for: 1m
        labels:
          severity: critical
        annotations:
          summary: "Instance {{ $labels.instance }} down"

      - alert: HighCPUUsage
        expr: 100 - (avg by(instance) (irate(node_cpu_seconds_total{mode="idle"}[5m])) * 100) > 80
        for: 5m
        labels:
          severity: warning
        annotations:
          summary: "High CPU on {{ $labels.instance }}"
```

# Grafana - Visualisation

### Installation

```
apt install -y software-properties-common
wget -q -O - https://packages.grafana.com/gpg.key | apt-key add -
echo "deb https://packages.grafana.com/oss/deb stable main" \
    > /etc/apt/sources.list.d/grafana.list
apt update && apt install grafana
systemctl enable --now grafana-server
```

**Acces** : http://localhost:3000 (admin/admin)

### Dashboards utiles

- ID 1860 : Node Exporter Full
- ID 12708 : HAProxy
- ID 9628 : PostgreSQL

# Uptime Kuma - Monitoring simple

Alternative legere pour monitoring de disponibilite.

**Installation**

```
docker run -d --name uptime-kuma \
    -p 3001:3001 \
    -v uptime-kuma:/app/data \
    louislam/uptime-kuma
```

**Types de monitors**

- HTTP(s) : Code retour, temps reponse, recherche texte
- TCP : Port ouvert
- Ping : ICMP
- DNS : Resolution
- Docker : Etat des conteneurs

# Documentation - Wiki.js

### Installation

```
docker run -d --name wikijs \
    -p 3000:3000 \
    -e DB_TYPE=sqlite \
    -v wikijs-data:/wiki/data \
    requarks/wiki
```

### Structure recommandee PCA/PRA

- 1-Gouvernance (politique, organisation)
- 2-Analyse (BIA, cartographie)
- 3-PCA (procedures mode degrade)
- 4-PRA (runbooks restauration)
- 5-Communication (templates, contacts)
- 6-Exercices (planning, RETEX)

# Documentation - Netbox

Documentation reseau et datacenter (IPAM + DCIM).

**Installation**

```
git clone https://github.com/netbox-community/netbox-docker.git
cd netbox-docker
docker-compose up -d
```

**Fonctionnalites**

- Gestion des adresses IP
- Inventaire equipements reseau
- Racks et datacenters
- Cables et connexions
- API REST complete

**Acces** : http://localhost:8000

# Communication de crise - Mattermost

Messagerie d'equipe open source (alternative Slack).

**Installation**

```
docker run -d --name mattermost \
    -p 8065:8065 \
    -v mattermost-data:/mattermost/data \
    mattermost/mattermost-team-edition
```

**Usage PCA/PRA**

- Canal dedie cellule de crise
- Canal par incident
- Integrations webhooks (alertes)
- Historique des echanges
- Fonctionne hors SI principal (cloud ou serveur dedie)

# Communication de crise - Element

Messagerie chiffree decentralisee (protocole Matrix).

**Avantages**

- Chiffrement de bout en bout
- Decentralise (pas de SPOF)
- Federation possible entre serveurs
- Open source

**Installation serveur Synapse**

```
docker run -d --name synapse \
    -p 8008:8008 \
    -v synapse-data:/data \
    matrixdotorg/synapse:latest
```

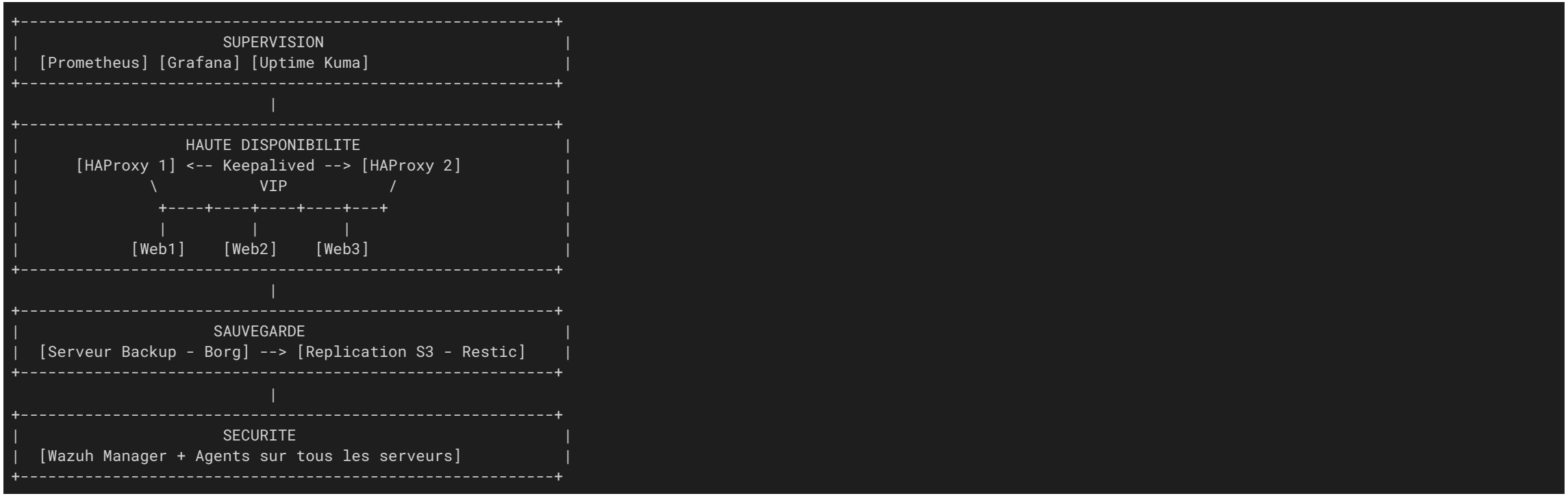
**Usage** : Canal de secours ultra-securise

UTOPIOS®
TRANSFORMATION DIGITALE INTELLIGENTE

# TP Final

# Infrastructure PCA/PRA Complete

# Architecture cible



# Etapes du TP

| Etape | Duree | Contenu |
|---|---|---|
| 1 | 30 min | Infrastructure de base (5 VMs) |
| 2 | 30 min | Haute disponibilite (HAProxy + Keepalived) |
| 3 | 20 min | Sauvegarde (BorgBackup) |
| 4 | 20 min | Supervision (Prometheus + Grafana) |
| 5 | 20 min | Securite (Wazuh) |
| 6 | 30 min | Tests de resilience |

**VMs a deployer**

- 2x HAProxy
- 2x Web (nginx)
- 1x Backup

# Tests de resilience

**Scenario 1 : Panne HAProxy**

- Arreter le HAProxy master
- Verifier le basculement automatique
- Observer les alertes dans Grafana

**Scenario 2 : Panne serveur web**

- Arreter un serveur web
- Verifier que HAProxy le retire du pool
- Observer la continuite de service

**Scenario 3 : Restauration**

- Simuler une corruption de donnees
- Restaurer depuis sauvegarde Borg
- Verifier le retour a la normale

# Checklist outils PCA/PRA

| Categorie | Verification |
|---|---|
| Sauvegarde | Solution deployee, chiffree, testee |
| Sauvegarde | Regle 3-2-1-1-0 respectee |
| Sauvegarde | Alertes echec configurees |
| HA | Load balancer en HA |
| HA | Base de donnees repliquee |
| HA | Pas de SPOF |
| Supervision | Metriques systeme collectees |
| Supervision | Alertes configurees |
| Securite | SIEM/collecte logs centralisee |
| Securite | FIM active |

# Ressources complementaires

**Documentation officielle**

| Outil | URL |
|---|---|
| BorgBackup | borgbackup.readthedocs.io |
| Restic | restic.readthedocs.io |
| Wazuh | documentation.wazuh.com |
| Prometheus | prometheus.io/docs |
| Grafana | grafana.com/docs |
| HAProxy | haproxy.org |
| Pacemaker | clusterlabs.org |
| Velociraptor | docs.velociraptor.app |

UTOPIOS®
TRANSFORMATION DIGITALE INTELLIGENTE

# Questions ?