

# Dossier de Projet

Titre: Administrateur Système et Réseau

Nom: RODRIGUES DOS SANTOS

Prénom: Mylène

En formation à l'école de La Plateforme

## 1.1 Présentation Personnelle

Je m'appelle Mylène Rodrigues Dos Santos et j'ai commencé ma reconversion dans le monde de l'informatique, plus précisément la cybersécurité en 2022. Ma reconversion fait suite à quelques années d'expérience dans le monde de la restauration.

Durant ces années, je me suis rendue compte que je ne voulais pas continuer en restauration jusqu'à ma retraite, j'ai donc commencé à me poser des questions et à réfléchir à ce qui m'intéressait et à ce que je pourrais faire durablement.

Enfant, j'étais passionnée par les ordinateurs et les logiciels attractifs, je me suis donc tournée vers l'informatique en général et la cybersécurité étant en plein essor, je me suis renseignée et documentée sur ce domaine. J'ai trouvé des domaines en cybersécurité tellement intéressants et passionnants que j'ai su que je voulais travailler dans ce milieu.

Curieuse par l'étendue du monde de la cybersécurité et des nouvelles technologies et avec une grande envie d'apprendre, je me suis donc dirigée vers l'école de La Plateforme afin de reprendre mes études et faire carrière dans la cybersécurité.

J'ai donc commencé une première année à l'école de la plateforme en IT puis je me suis spécialisée en cybersécurité la seconde année. C'est également cette année-là que j'ai trouvé et commencé mon alternance chez CortexEra, une entreprise de cybersécurité qui aide les entreprises à rester aux normes à respecter dans la cybersécurité mais également des pentests, et à protéger les entreprises de cybercriminel et des procédures à suivre en cas de cyberattaque.

## **1.2 Présentation de l'entreprise**

CortexEra est une jeune entreprise spécialisée dans la cybersécurité pour aider les petites et moyennes entreprises à se défendre contre divers cyberattaques.

CortexEra intervient pour plusieurs types de services: audits, audits sur IoT, pentests, réponse sur incident, gestion de crise, analyse de risques, récupération des données, installation et maintenance en informatique générale ... . Nous traitons la demande du client et intervenons si nécessaire selon la demande de service du et/ou des clients.

CortexEra est également certifié Qualiopi et référencé par Cybermalveillance et Dynabuy et nous participons également à des CTF et des conférences.

## **1.3 Summary In English**

During my internship, I had the opportunity to work at Cortexera, a cybersecurity company that specializes in supporting organizations targeted by cyberattacks. The company also conducts security audits to help businesses strengthen their digital defenses.

In addition to its operational services, Cortexera is involved in educational initiatives, providing cybersecurity training and awareness programs in various schools and academic institutions. This experience allowed me to gain valuable insight into both the technical and strategic aspects of cybersecurity.

## **1.4. Résumé entreprise**

J'ai travaillé chez CortexEra, une petite entreprise de quatre personnes dont je fais partie et où j'ai la chance d'avoir un patron qui a plus de 25 ans d'expérience dans le domaine de la cybersécurité qui m'a énormément appris.

Mon patron m'a fait revoir toutes les bases de l'école:

- en réseau: VirtualBox et réseau des entreprises à auditer,
- en code: HTML, CSS, Javascript, C/C++,
- en virtualisation,
- en analyse de log: parsing de log,
- en cloud: Acronis,
- les normes ISO 27xxx/RGPD
- ...

J'ai également appris les bases en soudure et électronique et en reverse engineering.

Je prépare également l'examen du CEH (Certified Ethical Hacker) qui est une certification reconnue, délivrée par l'organisation EC-Council qui forme sur la sécurité offensive comme les techniques de piratage utilisées par les cybercriminels (dans un cadre légal et éthique). Le CEH a pour but d'enseigner les compétences nécessaires pour tester les défenses d'un système informatique, prévenir les intrusions, renforcer les mesures de sécurité et connaître les techniques, outils et méthodes d'attaque les plus utilisées dans le monde réel .

J'ai aussi pu participer au renouvellement et à la préparation de plusieurs certifications de mon entreprise.

Concernant le cloud, mon entreprise travaille avec Acronis, une application Cloud qui permet de protéger son infrastructure avec un choix de services tiers à activer ou non. Elle est spécialisée dans la sauvegarde, la récupération après sinistre, la cybersécurité et la protection des données. Cette solution permet entre autres de:

- faire des backups du systèmes,
- planifier des sauvegardes,
- de créer des images complètes du système afin de pouvoir restaurer l'environnement en cas de perte ou de corruption,
- de sécuriser ses données, de pouvoir avoir un plan de remédiation si les données sont chiffrées,
- d'isoler un programme ou logiciel lorsque l'agent d'Acronis détecte un code malveillant
- protéger les fichiers critiques avec des fonctions de chiffrement, d'authentification et de contrôle d'intégrité
- effectuer des tests de restauration pour valider l'efficacité du plan de reprise d'activité (PRA)

Ce logiciel est constamment mis-à-jour. On peut visualiser tout cela avec un dashboard qui nous permet de voir rapidement ce qu'il se passe et s' il y a un problème.

On peut également via Acronis synchroniser les mails et ainsi pouvoir bloquer des attaques de type phishing, spear phishing, business email compromise ou encore des malspams.

Nous gérons donc la cybersécurité de plusieurs clients via ce logiciel.

J'ai également eu l'opportunité d'intervenir avec mon entreprise directement chez des clients pour faire des audits de sécurité, voir comment ça pouvait se passer sur

le terrain et comprendre comment et pourquoi des cyberattaques peuvent se produire que ce soit en distanciel ou directement sur le terrain. J'ai également vu quels équipements les entreprises possédaient et quels moyens elles utilisaient pour sécuriser leur système. J'ai donc pu participer à la rédaction de rapports en incluant aussi plusieurs schéma réseau des dites entreprises.

A la suite de ces interventions chez les clients pour des audits de sécurité, j'ai travaillé sur un code en HTML, CSS et Javascript pour faire une application qui automatise le schéma réseau d'une entreprise. Sur la première page, on peut mettre toutes les informations de l'entreprise. Sur la deuxième page nous avons un rack avec plusieurs boutons pour choisir quels équipements (marque et model) on veut rajouter dans le rack, y compris des boutons pour modifier la place de l'équipement dans le rack. Sur la dernière page on peut rajouter des photos des racks présent directement chez le client et ajouter des informations supplémentaires si nécessaire.

J'ai également pu assister à des conférences sur la cybersécurité comme Barbhack et assisté à des meetings où je présentais mon entreprise à d'autres entreprises.

J'ai été formé sur plusieurs logiciels comme wireshark, ghidra, openstreetmap, Acronis, Visual Studio Code etc.

## **1.5 Présentation de mon projet**

Une entreprise fictive souhaite moderniser et sécuriser son infrastructure IT grâce à la conteneurisation pour faciliter le déploiement, la maintenance et la sécurité système et des différents services. Cette infrastructure a pour but de fournir un environnement SIEM ( Security Information and Event Management) robuste, modulaire et sécurisé. Cela intègre une supervision et une politique de sécurité adaptée.

L'entreprise souhaite également renforcer sa sécurité grâce à une surveillance accrue de la supervision des données en temps réel, un système de détection d'intrusion et un accès distant chiffré et la protection des données. Les services doivent être accessibles via VPN et les métriques doivent être collectées et analysées pour garantir la disponibilité, la performance et la sécurité du système.

Pour répondre à cette demande, je vais mettre en place plusieurs containers Docker pour isoler et déployer rapidement les services applicatifs. je vais utiliser les services suivants:

- Grafana pour la visualisation et les dashboards de supervision

- Prometheus pour la collecte et la surveillance des métriques système
- Wazuh pour la détection des intrusions, l'analyse de logs et la supervision de la sécurité
- OpenVPN pour sécuriser l'accès distant à l'infrastructure
- Nginx utilisé en tant que reverse proxy sécurisé (HTTPS) afin de centraliser et sécuriser les accès aux services

Chaque service sera isolé dans son propre environnement docker et relié à des réseaux docker dédiés. La configuration repose un usage des dockerfiles, du fichier .env et des certificats SSL auto-signés. Plusieurs scripts bash seront mis en place pour déployer et stopper l'ensemble des conteneurs proprement.

Parallèlement, pour pouvoir valider la partie réseau, j'ai conçu et documenté une architecture réseau sécurisée pour une entreprise fictive sur Cisco Packet Tracer en:

- configurant et segmentant les sous-réseaux,
- les pare-feux,
- les VLANs,
- mise en place de protocole
- les routes Inter-VLAN
- serveur de sauvegarde TFTP
- simulation d'accès à Internet
- DMZ

Chaque équipement sera configuré et testé pour assurer le bon fonctionnement de l'infrastructure.

## **1.6 Présentation des outils utilisés dans ce projet:**

### **Docker**

Docker est un outil open-source qui permet de créer, déployer et exécuter des applications dans des conteneurs. On peut donc lancer un ou plusieurs services simultanément avec une simple commande. Il simplifie le processus de gestion des applications en isolant les applications les unes des autres et en utilisant des images pour créer des conteneurs.

### **Grafana**

Grafana est une plateforme open-source pour visualiser et donc pouvoir analyser des données. Prometheus est un système de surveillance et d'alerte également open-source. Ces deux services se complètent que ce soit pour la surveillance ou l'analyse des données.

Je commence par prometheus et grafana, je me connecte sur les deux interfaces et je peux directement synchroniser Prometheus à Grafana.

## **OpenVPN**

OpenVPN est une solution open source qui permet de créer un tunnel chiffré entre un client et un réseau distant comme Docker. Il chiffre donc les données et permet d'être protégé contre des attaques comme le sniffing, man-in-the-middle ou encore des fuites de données. Les services configurés avec OpenVPN ne seront accessibles que par les personnes autorisées et seront accessibles à distance et sécurisé.

## **Wazuh**

Wazuh est une plateforme opensource de détection d'intrusion et de surveillance. Elle aidera notre entreprise à détecter les menaces, à surveiller l'intégrité des fichiers mais également pour la collecte et l'analyse des logs et de la gestion des configurations système. Vous l'aurez compris, Wazuh est une solution complète et puissante et nécessaire pour la sécurité de l'entreprise.

J'intègre Wazuh dans docker en 3 services distincts mais partageant le même réseau: un wazuh Indexer, un wazuh manager et un wazuh dashboard.

### **Wazuh Indexer**

Wazuh Indexer est une solution d'indexation, de recherche et d'analyse basée sur OpenSearch. Il est utile car il facilite la recherche et l'analyse des logs et événements. Il permet la gestion de grande quantité de données et est scalable (il s'adapte donc pour les infrastructures en croissances).

### **Wazuh Manager**

Wazuh Manager est responsable de la gestion des agents, de la collecte des données et de l'analyse des événements de sécurité. Il permet une gestion centralisée des agents de Wazuh et la configuration et le déploiement des politiques de sécurité.

### **WazuhDashboard**

Wazuh Dashboard est une interface utilisateur basée sur OpenSearch Dashboards, conçue pour visualiser et analyser les données de sécurité collectées par Wazuh.

On peut y créer des dashboards personnalisables pour la surveillance des métriques et est optimisé pour fonctionner avec Wazuh Manager et Wazuh Indexer.

### **Nginx(proxy de Wazuh)**

Nginx est un logiciel open source conçu pour le service Web, le reverse proxy, l'équilibrage de charge (...). Nous utiliserons le reverse proxy de nginx dans ce projet afin d'apporter une couche de sécurité supplémentaire à Wazuh. Nginx peut filtrer et bloquer les requêtes malveillantes avant qu'elles n'atteignent les services Wazuh. Il peut également gérer la gestion des certificats SSL/TLS. Nginx améliore donc la performance et la disponibilité des services, la gestion des accès et des certificats, la répartition des charges et la sécurité. Je vais donc l'intégrer à Wazuh.

### **OpenSearch**

OpenSearch est un projet créé par Amazon mais open source basé sur les forks d'anciennes versions d'Elasticsearch et de kibana. Il est utilisé pour la surveillance des applications et l'analyse des journaux en temps réel. OpenSearch Dashboards permet aux clients d'explorer facilement leurs données et prend en charge un nombre important de fonctionnalités de recherches d'analytiques.

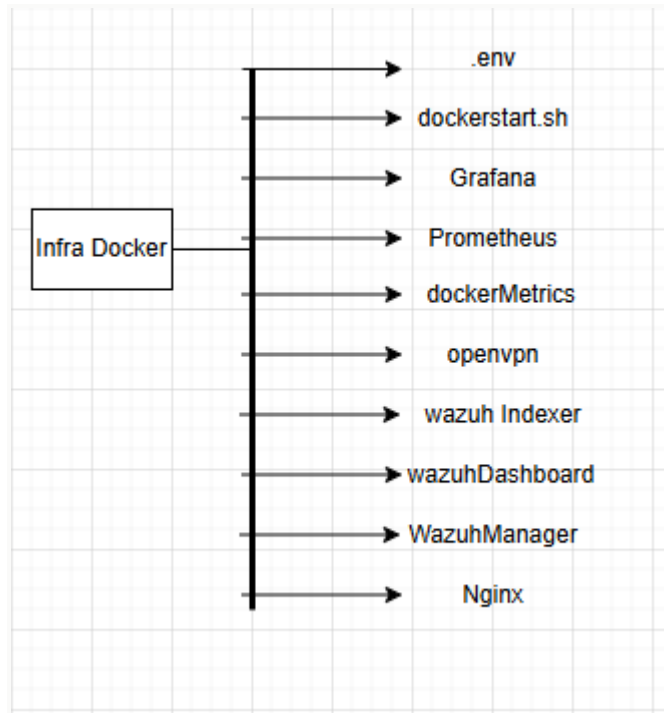
### **Cisco Packet Tracer**

Cisco Packet Tracer est un logiciel de simulation réseau développé par Cisco Systems. Il permet de concevoir, configurer, tester, modifier et dépanner des réseaux informatiques dans un environnement virtuel. Ce logiciel est très utile car il nous permet d'éviter des erreurs de configuration, mais aussi de compatibilité entre différents composants ou machines qu'il peut y avoir dans le monde réel. C'est donc un logiciel qui nous permet de gagner du temps et de l'argent.

Cisco Packet Tracer permet entre autres de créer des topologies de réseaux, de configurer des équipements Cisco (qui se trouve majoritairement dans les entreprises) avec une interface CLI (ligne de commande), de simuler la communication et connexion réseau grâce à des pings, transfert de fichier, routage (...), de mettre en place différents protocoles comme FTP, DNS, DHCP, Bluetooth (...) et on peut l'utiliser hors ligne.

## 2. Projet SIEM avec docker

### 2.0 Architecture simplifié de l'infrastructure:



Chaque dossier de service aura ses propres sous-dossiers.

### 2.1 Installation Docker

Tout d'abord, j'installe Docker pour pouvoir lancer nos containers grâce à un script d'installation.



```
nerds@vbox: ~/Entreprise
GNU nano 7.2 dockerInstall
#uninstall all conflicting packages

for pkg in docker.io docker-doc docker-compose podman-docker containerd runc; do

# Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/debian/gpg -o /etc/apt/keyrin
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.as
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update

#install the latest version

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^_ Replace   ^U Paste     ^J Justify   ^_ Go To Line

#install the latest version

sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

#test

sudo docker run hello-world

#add group

sudo groupadd docker

#add user

sudo usermod -aG docker $nerds

#Add docker system boot

sudo systemctl enable docker.service
sudo systemctl enable containerd.service

[ Wrote 37 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  ^U Undo      ^M-A Set Mark  ^M-J To
^X Exit      ^R Read File ^_ Replace   ^U Paste     ^J Justify   ^_ Go To Line ^M-B Undo     ^M-C Copy     ^M-L To
```

## 2.2 Création de réseaux Docker

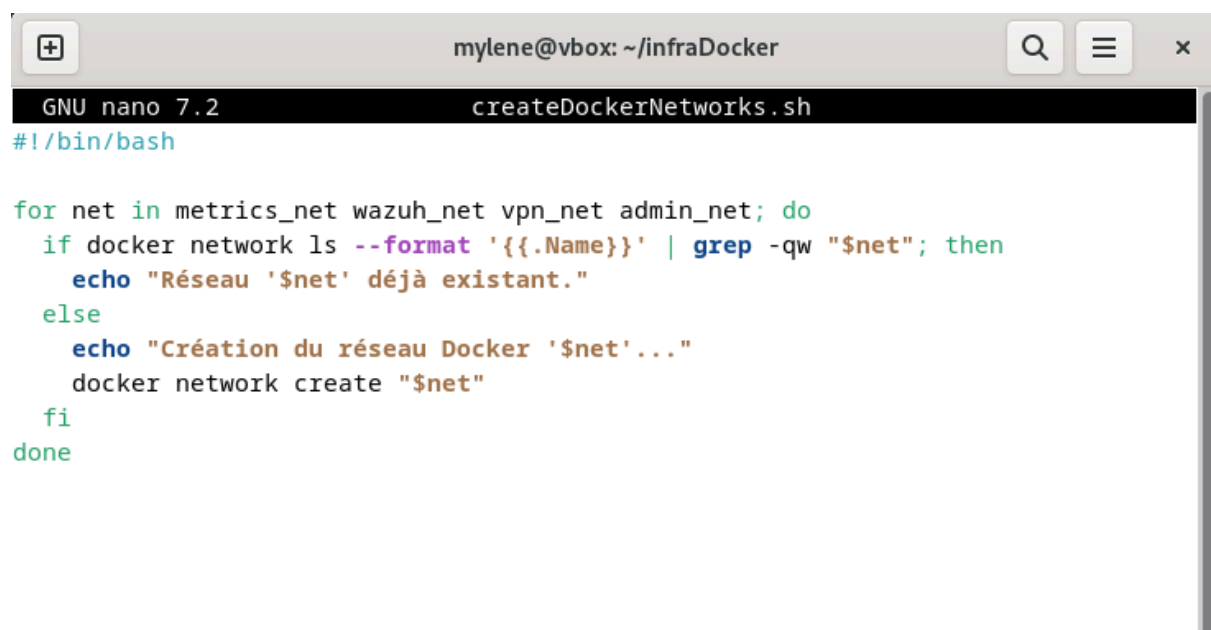
Ensuite, pour des raisons de sécurité, de modularité, de lisibilité et de maintenance, pour chaque services listés ci-dessus nous allons créer un fichier yaml (Docker Compose) en séparant les réseaux.

Je vais alors créer 4 réseaux distincts:

- un réseau metrics\_net pour la supervision et les métriques qu'on recueillera grâce à prometheus et grafana et aux métriques de docker.
- un réseau wazuh\_net pour la surveillance et la supervision des données grâce au SIEM de Wazuh
- un réseau vpn\_net pour les services VPN
- un réseau admin\_net pour centraliser les vues.

Pour des raisons de sécurité et de confidentialité, je vais également utiliser un fichier .env pour regrouper toutes les valeurs comme les ports et mot de passe pour ne pas qu'il s'affiche directement dans les fichiers de dockers compose.

Pour la création des réseaux Docker:



The screenshot shows a terminal window titled 'mylene@vbox: ~/infraDocker'. The terminal is running GNU nano 7.2 editing a file named 'createDockerNetworks.sh'. The script content is as follows:

```
#!/bin/bash

for net in metrics_net wazuh_net vpn_net admin_net; do
  if docker network ls --format '{{.Name}}' | grep -qw "$net"; then
    echo "Réseau '$net' déjà existant."
  else
    echo "Création du réseau Docker '$net'..."
    docker network create "$net"
  fi
done
```

## 2.3 Configuration des services avec Docker compose et configuration post-lancement.

Je vais créer et configurer un fichier yaml pour chaque service.

### Docker Métriques:

```

services:
  docker_metrics:
    image: docker
    command: docker --metrics-addr 0.0.0.0:9323 --experimental
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
    networks:
      - ${METRICS_NET}
      - ${VPN_NET}

networks:
  metrics_net:
    name: ${METRICS_NET}
    driver: bridge
    external: true
  vpn_net:
    name: ${VPN_NET}
    driver: bridge
    external: true

```

## Grafana

```

GNU nano 7.2
apiVersion: 1

datasources:
  -name: Prometheus
    type: prometheus
    access: proxy
    url: http://prometheus:9090
    isDefault: true

```

---

```
services:
  grafana:
    image: grafana/grafana
    container_name: grafana
    expose:
      - "3000:3000"
    volumes:
      - grafana_data:/var/lib/grafana #persistance des users,dashboards...
      - ./provisioning:/etc/grafana/provisioning #config locale monté ds le container
    networks:
      - ${METRICS_NET}
      - ${ADMIN_NET}
      - ${VPN_NET}
    restart: unless-stopped
    environment:
      - GF_SECURITY_ADMIN_USER=${PROXY_USER}
      - GF_SECURITY_ADMIN_PASSWORD=${PROXY_PASSWORD}

volumes:
  grafana_data:

networks:
  vpn_net:
    name: ${VPN_NET}
    driver: bridge
    external: true
  admin_net:
    name: ${ADMIN_NET}

  metrics_net:
    name: ${METRICS_NET}
    driver: bridge
    external: true
```

## Prometheus

En complément du docker compose pour prometheus, je vais rajouter deux fichiers:

- un fichier de scrape pour pouvoir collecter des métriques à des points précis du réseau
- un fichier de règles d'alerte pour détecter automatiquement des anomalies dans le trafic réseau. En cas de dépassement de seuils prédéfinis et inhabituel, des alertes seront générées.

```
services:
  prometheus:
    image: prom/prometheus
    container_name: prometheus
    ports:
      - "9090:9090"
    volumes:
      - prometheus_data:/prometheus
      - ./prometheus.yml:/etc/prometheus/prometheus.yml
      - ./alert.rules.yml:/etc/prometheus/alert.rules.yml
    networks:
      - ${METRICS_NET}
      - ${VPN_NET}
    restart: unless-stopped

volumes:
  prometheus_data:

networks:
  vpn_net:
    name: ${VPN_NET}
    driver: bridge
    external: true
  metrics_net:
    name: ${METRICS_NET}
    driver: bridge
    external: true

global:
  scrape_interval: 15s

rule_files:
  - "alert.rules.yml"

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

  - job_name: 'docker'
    static_configs:
      - targets: ['172.17.0.1:9323']
```

```

GNU nano 7.2                                alert.rules.yml
groups:
- name: alertsCPU
  rules: HighCPUUsage
  expr: 100 - (avg by (instance) (rate(node_cpu_seconds_total{mode="idle"}[2m])) * 100) > 80
  for: 1m
  labels:
    severity: warning
  annotations:
    summary: "CPU usage is high on instance {{ $labels.instance }}"
    description: "CPU usage above 80% for more than 1 minute"

- name: conteneurAlerte
  rules:
  - alert: ConteneurDown
    expr: up{job="docker"} == 0
    for: 30s
    labels:
      severity: critical
    annotations:
      summary: "Un conteneur Docker est inactif"
      description: "Le conteneur ne répond plus"

```

## Wazuh

### Indexer:

```

services:
  wazuh.indexer:
    image: wazuh/wazuh-indexer:4.7.3
    container_name: wazuh
    restart: unless-stopped
    ports:
      - "9200:9200"
    environment:
      - OPENSEARCH_INITIAL_ADMIN_PASSWORD=${OPENSEARCH_INITIAL_ADMIN_PASSWORD}
      - OPENSEARCH_PASSWORD=${INDEXER_PASSWORD}
      - DISCOVERY_TYPE=single-node
      - DISABLE_SECURITY_PLUGIN=false
    volumes:
      - wazuh-indexer-data:/usr/share/opensearch/data
      - ./config/opensearch-security:/usr/share/opensearch/plugins/opensearch-security/securityconfig
      - ./config/wazuh_indexer_ssl_certs/root-ca.pem:/usr/share/opensearch/config/certs/root-ca.pem:ro
      - ./config/wazuh_indexer_ssl_certs/certs/admin.pem:/usr/share/opensearch/config/certs/admin.pem:ro
      - ./config/wazuh_indexer_ssl_certs/certs/admin-key.pem:/usr/share/opensearch/config/certs/admin-key.pem:ro

    networks:
      - ${WAZUH_NET}
      - ${VPN_NET}

```

```
volumes:
  indexer_data:
  wazuh-indexer-data:
```

```
networks:
  wazuh_net:
    name: ${WAZUH_NET}
    driver: bridge
    external: true
  vpn_net:
    name: ${VPN_NET}
    driver: bridge
    external: true
```

Manager:

```
services:
  wazuh.manager:
    image: wazuh/wazuh-manager:4.7.3
    container_name: wazuh_manager
    ports:
      - "1514:1514/udp" #agent
      - "1515:1515" # agent
      - "5500:5500" #API
    volumes:
      - wazuh_data:/var/ossec/data
      - ../wazuh/config/certs:/etc/ssl/certs/indexer:ro
      - wazuh-manager-data:/var/ossec/data
    networks:
      - ${WAZUH_NET}
      - ${VPN_NET}
    restart: unless-stopped
    environment:
      - INDEXER_URL=https://wazuh:9200
      - INDEXER_USERNAME=${INDEXER_USERNAME}
      - INDEXER_PASSWORD=${INDEXER_PASSWORD}
      - SSL_CERTIFICATE_AUTHORITIES=/etc/ssl/certs/indexer/ca.crt

volumes:
  wazuh_data:
  wazuh-manager-data:
```

```

networks:
  wazuh_net:
    name: ${WAZUH_NET}
    driver: bridge
    external: true
  vpn_net:
    name: ${VPN_NET}
    driver: bridge
    external: true

```

## Dashboard:

---

```

services:
  wazuh.dashboard:
    image: wazuh/wazuh-dashboard:4.7.3
    container_name: wazuh.dashboard
    restart: unless-stopped
    ports:
      - "5601:5601" # web API
    environment:
      - DASHBOARD_PASSWORD=${DASHBOARD_PASSWORD}
      - DASHBOARD_USERNAME=${DASHBOARD_USERNAME}
      - INDEXER_USERNAME=${INDEXER_USERNAME}
      - INDEXER_PASSWORD=${INDEXER_PASSWORD}
      - WAZUH_API_URL=https://wazuh-manager:55000
      - SSL_CERTIFICATE_AUTHORITIES=/etc/ssl/certs/indexer/ca.crt
    networks:
      - ${WAZUH_NET}
      - ${ADMIN_NET}
      - ${VPN_NET}

    volumes:
      - ../wazuh/config/certs:/etc/ssl/certs/indexer:ro

networks:
  wazuh_net:
    name: ${WAZUH_NET}
    driver: bridge
    external: true
  admin_net:

```

---



```
networks:
  wazuh_net:
    name: ${WAZUH_NET}
    driver: bridge
    external: true
  admin_net:
    name: ${ADMIN_NET}
    driver: bridge
    external: true
  vpn_net:
    name: ${VPN_NET}
    driver: bridge
    external: true
```

## OpenVPN

---

```
services:
  openvpn:
    image: kylemanna/openvpn
    container_name: openvpn
    cap_add:
      - NET_ADMIN
    volumes:
      - ./data:/etc/openvpn
    ports:
      - "1194:1194/udp"
    restart: unless-stopped
    networks:
      - ${VPN_NET}
    environment:
      - DEBUG=1
      - OVPN_SERVER_CN=${VPN_SERVER_CN}

networks:
  vpn_net:
    name: ${VPN_NET}
    driver: bridge
```

Afin de renforcer la sécurité des services Wazuh, je vais mettre en place un reverse proxy NGINX. Bien que les services soient déjà protégés par un tunnel chiffré via OpenVPN, le proxy jouera un rôle supplémentaire en assurant :

- la redirection des requêtes vers les bons services internes (comme le dashboard Wazuh)
- le chiffrement des échanges via HTTPS pour éviter que les données ne transitent en clair même en interne
- l'ajout d'une authentification sécurisée en amont

Le reverse proxy ajoute alors une couche de sécurité supplémentaire

## Nginx

---

```
services:
  wazuh_proxy:
    image: nginx:alpine
    container_name: wazuh_proxy
    expose:
      - "443:443"
      - "80:80"
    volumes:
      - ./nginx.conf:/etc/nginx/nginx.conf:ro
      - ./htpasswd:/etc/nginx/.htpasswd:ro
      - ./certs:/etc/nginx/certs:ro
    networks:
      - ${WAZUH_NET}
      - ${VPN_NET}
    restart: unless-stopped

networks:
  wazuh_net:
    name: ${WAZUH_NET}
    driver: bridge
    external: true
  vpn_net:
    name: ${VPN_NET}
    driver: bridge
    external: true
```

## Fichier de configuration nginx

---

```
event {}

http {
    include      mime.types
    default_type application/octet-stream

    server {
        listen 80;
        server_name _;

        return 301 https://$host$request_uri;
    }

    server {
        listen 443 ssl;
        server_name localhost;

        ssl_certificate      /etc/nginx/certs/cert.pem;
        ssl_certificate_key  /etc/nginx/certs/key.pem;

        ssl_protocols       TLSv1.2 TLSv1.3
        ssl_ciphers          HIGH:!aNULL:!MD5;

        location / {
            proxy_http_version 1.1;
            proxy_pass http://wazuh_dashboard:5601;
            proxy_set_header Host $host;
```

## Génération de clés et certificats

[illegible]

## fichier .env

Les fichiers .env est un moyen souvent utilisé pour centraliser certaines configurations dans notre fichier de Docker Compose comme les secrets ou les variables sensibles (ex: mot de passe). Ca nous évite de modifier directement le docker Compose et on peut le déployer sur plusieurs environnements.

```
mylene@vbox:~/infraDocker$ cat .env
#.env

##Wazuh indexer
INDEXER_USERNAME=admin
INDEXER_PASSWORD=admin
OPENSEARCH_INITIAL_ADMIN_PASSWORD=admin

#Wazuh Manager
WAZUH_MANAGER_USER=admin
WAZUH_MANAGER_PASSWORD=admin

#DASHBOARD
DASHBOARD_USERNAME=admin
DASHBOARD_PASSWORD=admin
PROXY_USER=admin
PROXY_PASSWORD=monpass

#Réseau Docker
METRICS_NET=metrics_net
WAZUH_NET=wazuh_net
ADMIN_NET=admin_net
VPN_NET=vpn_net

#Grafana
GF_SECURITY_ADMIN_USER=admin
GF_SECURITY_ADMIN_PASSWORD=admin
```

Avant de lancer tous les conteneurs, nous avons d'autres configurations à mettre en place.

## Wazuh

Pour Wazuh, il est nécessaire de pré configurer le serveur et donc de générer les certificats et clés nécessaires à la sécurisation des communications entre les composants (Indexer, Manager, Dashboard). Cette étape garantit que les échanges internes sont authentifiés et chiffrés. Une fois ces éléments mis en place, les conteneurs peuvent être lancés en toute sécurité.

```
GNU nano 7.2 generate-indexer-certs.yml
Services:
  generator:
    image: wazuh/wazuh-certs-generator:0.0.2
    hostname: wazuh-certs-generator
    volumes:
      - ./config/wazuh_indexer_ssl_certs:/certificates/
      - ./config/certs.yml:/config/certs.yml
      - ./config/certs:/config
```

```
mylene@vbox:~/infraDocker/wazuh$ nano generate-indexer-certs.yml
mylene@vbox:~/infraDocker/wazuh$ sudo docker compose -f generate-indexer-certs.yml run --rm generator
The tool to create the certificates exists in the in Packages bucket
24/07/2025 14:57:42 INFO: Generating the root certificate.
24/07/2025 14:57:42 INFO: Generating Admin certificates.
24/07/2025 14:57:42 INFO: Admin certificates created.
Moving created certificates to the destination directory
Changing certificate permissions
Setting UID indexer and dashboard
Setting UID for wazuh manager and worker
mylene@vbox:~/infraDocker/wazuh$
```

## OpenVPN

En ce qui concerne OpenVPN, il faut l'initialiser sinon il ne pourra pas démarrer. Il faut ensuite le sécuriser pour avoir une transmission chiffrée.

```
mylene@vbox:~/infraDocker/openvpn$ sudo docker run -v $PWD/data:/etc/openvpn --rm kylemanna/openvpn ovpn_genconfig -u udp://localhost
[sudo] password for mylene:
Processing PUSH Config: 'block-outside-dns'
Processing Route Config: '192.168.254.0/24'
Processing PUSH Config: 'dhcp-option DNS 8.8.8.8'
Processing PUSH Config: 'dhcp-option DNS 8.8.4.4'
Processing PUSH Config: 'comp-lzo no'
Successfully generated config
Cleaning up before Exit ...
```

```
mylene@vbox:~/infraDocker/openvpn$ sudo docker run -v $PWD/data:/etc/openvpn --rm -it kylemanna/openvpn ovpn_initpki

init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /etc/openvpn/pki
```

Using SSL: openssl OpenSSL 1.1.1g 21 Apr 2020

```
Enter New CA Key Passphrase:
Re-Enter New CA Key Passphrase:
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name (eg: your user, host, or server name) [Easy-RSA CA]:mylene

CA creation complete and you may now import and sign cert requests.
Your new CA certificate file for publishing is at:
/etc/openvpn/pki/ca.crt
```

```
.....
writing new private key to '/etc/openvpn/pki/easy-rsa-73.GIGeGh/tmp.fEKLjD'
```

```
-----
Using configuration from /etc/openvpn/pki/easy-rsa-73.GIGeGh/tmp.flcdBn
Enter pass phrase for /etc/openvpn/pki/private/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName          :ASN.1 12:'localhost'
Certificate is to be certified until Oct 28 11:01:45 2027 GMT (825 days)
```

```
Write out database with 1 new entries
Data Base Updated
```

```
Using SSL: openssl OpenSSL 1.1.1g 21 Apr 2020
Using configuration from /etc/openvpn/pki/easy-rsa-148.hhGo0h/tmp.kCkeje
Enter pass phrase for /etc/openvpn/pki/private/ca.key:
```

```
An updated CRL has been created.
CRL file: /etc/openvpn/pki/crl.pem
```

Puis il faut créer un utilisateur qui pourra se connecter:

```
mylene@vbox:~/infraDocker/openvpn$ sudo docker run -v $PWD/data:/etc/openvpn --rm -it kylemanna/openvpn easyrsa build-client-full mylene
Using SSL: openssl OpenSSL 1.1.1g  21 Apr 2020
Generating a RSA private key
.....+++++
.....+++++
writing new private key to '/etc/openvpn/pki/easy-rsa-1.CoNaIi/tmp.GBGhae'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
Using configuration from /etc/openvpn/pki/easy-rsa-1.CoNaIi/tmp.KKcpnE
Enter pass phrase for /etc/openvpn/pki/private/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :ASN.1 12:'mylene'
Certificate is to be certified until Oct 28 11:04:13 2027 GMT (825 days)

Write out database with 1 new entries
Data Base Updated
```

```
docker run -v $PWD/data:/etc/openvpn --rm kylemanna/openvpn ovpn_getclient mylene > mylene.ovpn
```

Pour la sécurité du réseau, je vais utiliser iptables qui est un outil pour le renforcement de la sécurité des conteneurs et du réseau de diverses menaces. On l'utilise pour autoriser le trafic entrant et sortant des conteneurs Docker en définissant des règles précises pour restreindre l'accès aux ports.

Il peut être également utilisé pour isoler les conteneurs et réduire les impacts d'attaques de type DoS par exemple. Je vais donc seulement autoriser les ports du conteneur du VPN à être accessibles du réseau.

En complément d'Nginx, cette configuration permet de réduire fortement la surface d'attaque, car seul le reverse proxy est directement exposé.



```
#etape 1: Réinitialiser
```

```
iptables -F  
iptables -X  
iptables -t nat -F  
iptables -t nat -X
```

```
###tout bloquer
```

```
iptables -P INPUT DROP  
iptables -P FORWARD DROP  
iptables -P OUTPUT ACCEPT
```

```
#autoriser le loopback
```

```
iptables -A INPUT -i lo -j ACCEPT
```

```
#autoriser les connexions établies:::
```

```
iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
```

```
#Autoriser openvpn:
```

```
iptables -A INPUT -p udp --dport 1194 -j ACCEPT
```

```
#autoriser le trafic Docker
```

```
iptables -A INPUT -i docker0 -j ACCEPT
```

```
#autoriser la communication entre les conteneurs docker
```

```
iptables -A FORWARD -i docker0 -o docker0 -j ACCEPT
```

Il faut ensuite activer le routage VPN pour que l'utilisateur puisse accéder aux services de docker.

```
mylene@vbox:/etc$ sudo sysctl -p  
net.ipv4.ip_forward = 1  
mylene@vbox:/etc$ █
```

Les différents fichiers yaml étant dans des répertoires différents, je lance un script bash pour pouvoir tous les lancer en même temps.

```
#!/bin/bash
echo "Démarrage des services"

cd prometheus && docker compose up -d && cd ../grafana/ && docker compose up -d && cd ../openvpn/ && docker compose up -d && cd ../wazuh/ &&
echo "fin des services"
```

```
[+] Running 19/74
  : wazuh.indexer [██████████] 338.1MB / 1.522GB Pulling 95.3s
  : openvpn [ ] Pulling 95.3s
  : wazuh.dashboard [██████████] Pulling 95.3s
  : grafana [ ] Pulling 95.3s
  : prometheus [ ] Pulling 95.3s
  : wazuh.manager [ ] Pulling 95.3s
```

Pour des raisons de gestion, je vais également créer un script pour stopper les conteneurs lancés.

```
#!/bin/bash
echo "Arrêt des services"

cd prometheus && docker compose down && cd ../grafana/ && docker compose down && cd ../wazuh/ && docker compose down && cd ../wazuhManager/ &&
echo "Tous les services sont arrêtés."
```

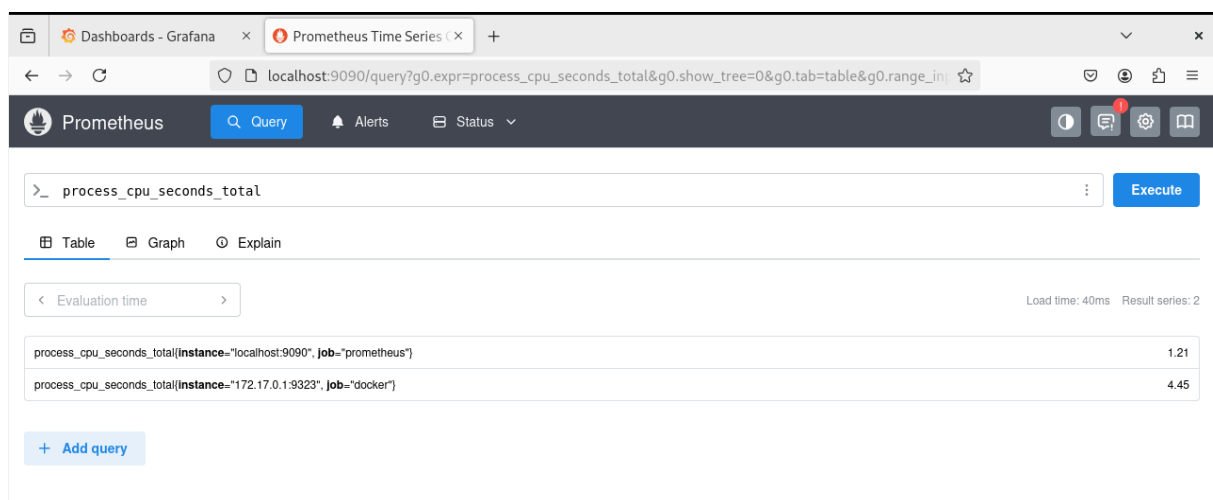
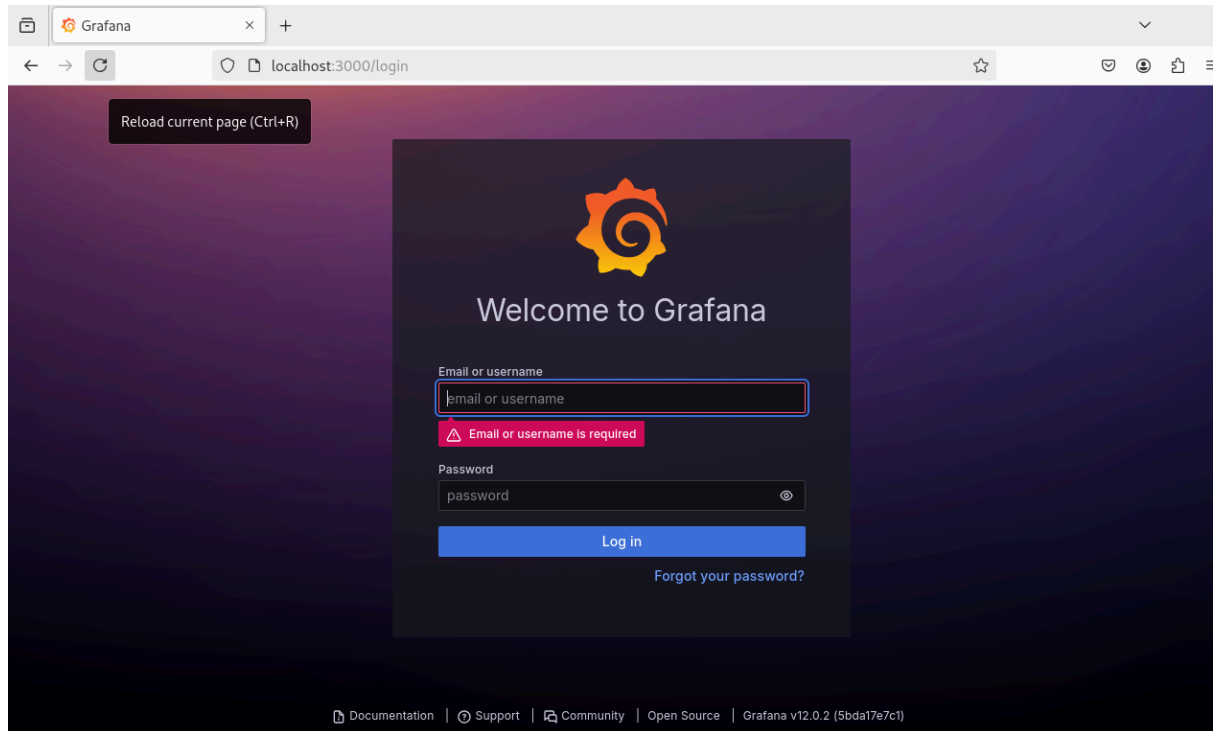
  

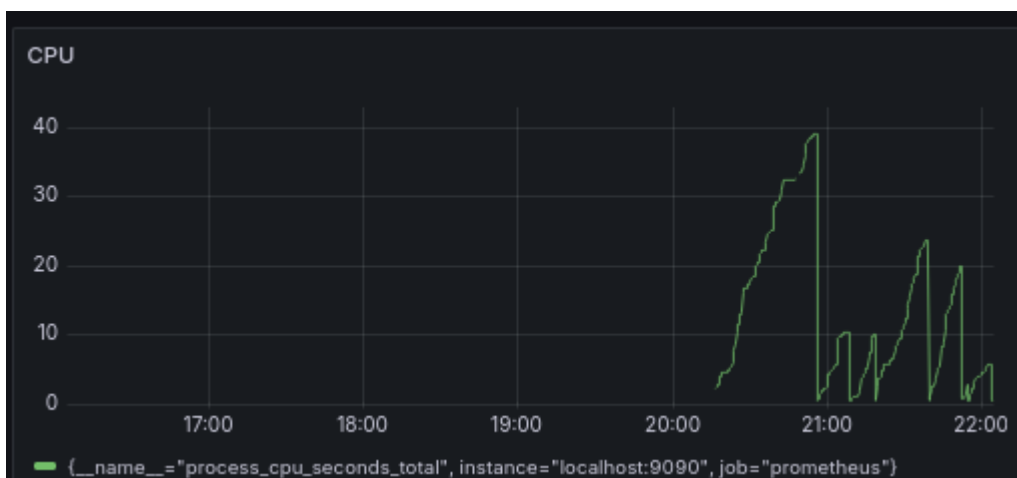
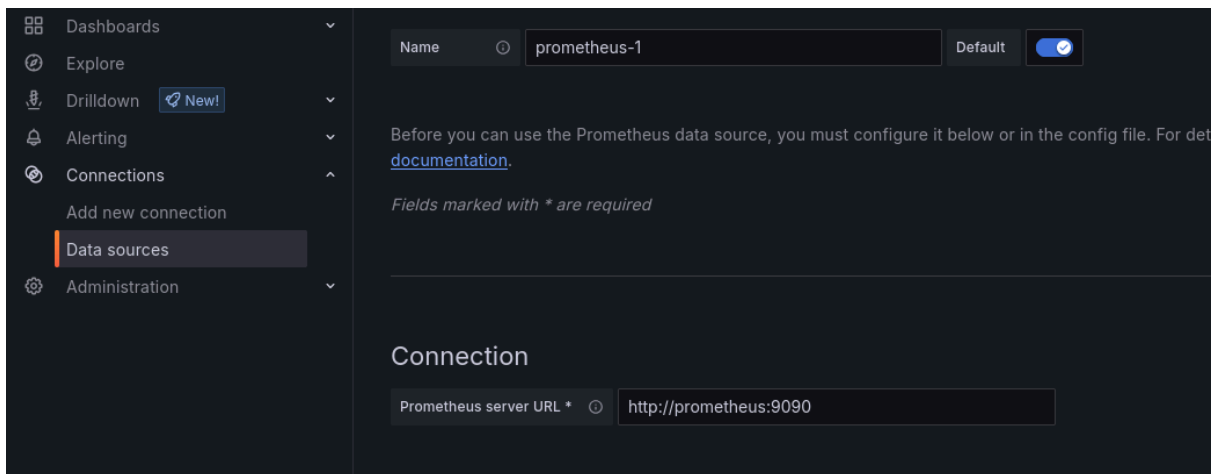
```
mylene@vbox:~/infraDocker$ sudo ./stop.sh
Arrêt des services
[+] Running 1/1
  ✓ Container prometheus Removed 0.0s
[+] Running 1/1
  ✓ Container grafana Removed 0.0s
[+] Running 1/1
  ✓ Container wazuh Removed 0.0s
[+] Running 1/1
  ✓ Container wazuh_manager Removed 0.3s
[+] Running 1/1
  ✓ Container wazuh.dashboard Removed 0.0s
```

Une fois les scripts lancés et le chargement terminé, je peux directement aller sur les interfaces que j'ai lancé.

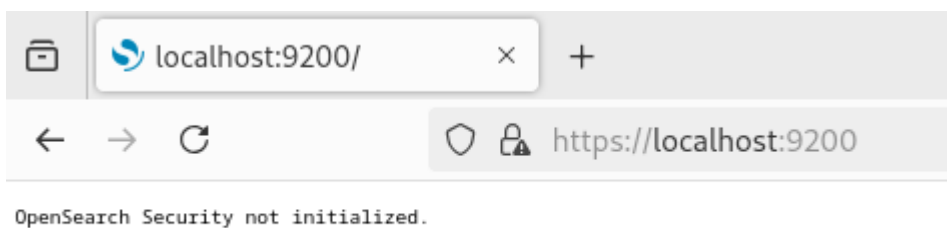
## 2.4 Configuration et application après lancement des conteneurs

Je peux commencer à faire des visuels par rapport aux données transitant sur le trafic réseau et sur la consommation des ressources de la machine (RAM, CPU, disque ...). Je met également en place des seuil d'alertes sur Prometheus pour qu'on soit prévenu en cas de dépassement d'un seuil non autorisé.





Pour wazuh:



Pour initialiser le serveur d'opensearch afin de pouvoir utiliser wazuh, je dois aller dans le conteneur de wazuh:

```
wazuh-indexer@5aeb0e4933ba:~/plugins/opensearch-security/tools$ ./securityadmin.sh \ -cd /usr/share/opensearch/plugins/opensearch-security/s
ecurityconfig/ \ -icl -nhnv \ cacert /usr/share/opensearch/config/root-ca.pem \ -cert /usr/share/opensearch/config/indexer.pem \ -key /usr/s
hare/opensearch/config/indexer-key.pem
*****
** This tool will be deprecated in the next major release of OpenSearch **
** https://github.com/opensearch-project/security/issues/1755 **
*****
WARNING: nor OPENSEARCH_JAVA_HOME nor JAVA_HOME is set, will use
wazuh-indexer@5aeb0e4933ba:~/plugins/opensearch-security/tools$
```

Une fois le script `security.admin` exécuté, je peux directement aller dans l'interface de Wazuh.

Je passe ensuite à l'installation d'agents de Wazuh sur mes conteneurs les plus sensibles comme OpenVPN et NGINX. L'agent de Wazuh me servira pour détecter des connexions suspectes et des attaques par brute-force pour mon service OpenVPN. Pour Nginx, il me servira dans la supervision des logs d'accès et pour détecter des anomalies. Je ne l'installe pas sur mes autres conteneurs car je ne veux pas alourdir mon infrastructure.

L'installation de l'agent se fait via un `dockerfile` qui sera dans le dossier du service qu'il doit protéger. Une fois notre `Dockerfile` en place, on ajoute dans le `docker` compose au service *environment*: l'IP de mon manager Wazuh.

Agent Wazuh pour OpenVPN:

```
FROM debian:12

ENV WAZUH_MANAGER_IP=localhost
ENV AGENT_NAME=myOpenVPNAgent

RUN apt-get update && \
    apt-get install -y curl gnupg && \
    curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | apt-key add - && \
    echo "deb https://packagees.wazuh.com/4.x/apt/ stable main" > /etc/apt/sources.list.d/wazuh.list && \
    apt-get update && \
    apt-get install -y wazuh-agent

#Configuration agent

RUN sed -i "s|<address>.*</address>|<address>${WAZUH_MANAGER_IP}</address>|" /var/ossec/etc/ossec.conf && \
    sed -i "s|<name>.*</name>|<name>${AGENT_NAME}</name>|" /var/ossec/etc/ossec/conf

CMD["/var/ossec/bin/wazuh-control", "start"]
```

### 3. Cisco Packet Tracer



#### 3.1 Les besoins de l'entreprise

***Une infrastructure réseau se repose sur les besoins techniques et fonctionnels du client (l'entreprise) et permet une réponse structurée, organisée adaptée aux attentes.***

Mon entreprise fictive à besoin d'un réseau segmenté en plusieurs sous-réseaux selon les services dont elle dispose:

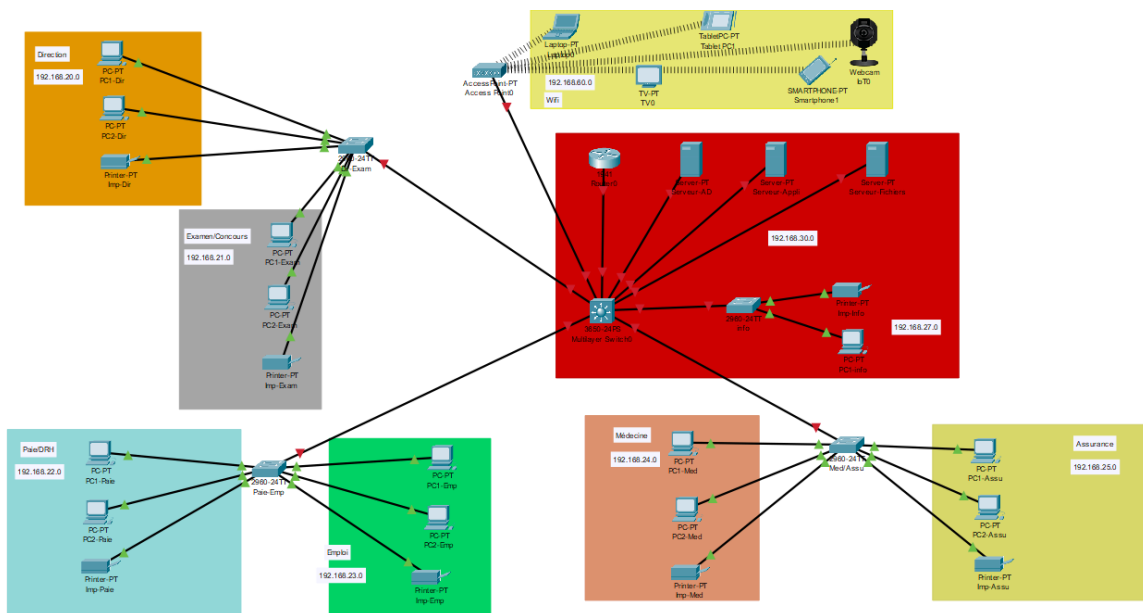
- Direction
- Examen/Concour
- Wifi
- Salle Serveur
- Paie/ DRH
- Emploie
- Médecine
- Assurance
- VPN
- Téléphonie
- Imprimerie

L'entreprise à besoin de configurer et sécuriser les différents équipements de connexion afin que seul les personnes autorisées puissent accéder à ce réseau. L'entreprise doit avoir:

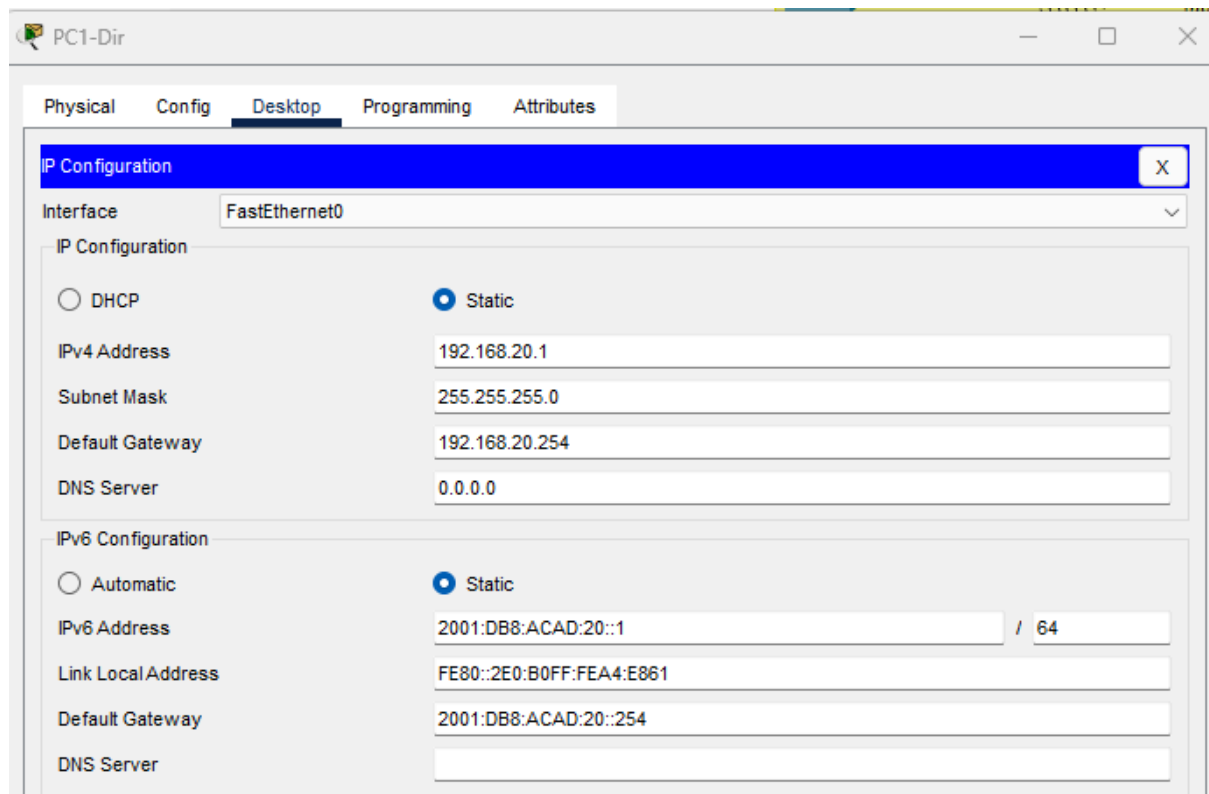
- adresse IPv4 et IPV6
- commutateurs qui puisse être gérée à distance et sécurisé de couche 2 et 3
- configuration d'un nom de domaine, des routeurs
- mots de passe chiffrés
- sous-réseau distant VPN
- VLANs
- connexion WAN
- 2 routeurs

### 3.2 Mise en place, configuration et sécurité

Dans chacune de ses salles doit se trouver au minimum de plusieurs PC et d'une imprimante (je rajoute les téléphone VoIP plus tard) excepté pour la salle WiFi qui sera composé d'un laptop, d'une Télé, d'une tablette, d'un smartphone et d'une webcam. Une fois ces sous-réseaux agencés, je les ai reliés avec un commutateur Cisco de couche 2 sauf pour la salle info qui dispose d'un commutateur de couche 3 et la salle wifi qui dispose d'un appareil sans fil.



Je commence par configurer tous les PC en IPv4 et IPv6 dans leurs sous-réseaux respectifs. Je fais de même pour l'imprimante qui aura son propre sous-réseau.



Je configure ensuite les interfaces SVI des commutateurs de couche 2 et lui ajoute un VLAN différent du VLAN par défaut qu'on créera plus tard et également une adresse IPv4 et IPv6. Je configure ensuite la gateway du commutateur et je répète ensuite l'opération pour tous les commutateurs de couches 2.

Commandes utilisées:

Configuration VLANs:

```
en
conf t
hostname Paie-Emp
int vlan 10
ip address 192.168.100.1/24
exit
sdm prefer dual-ipv4-and-ipv6 default
exit
reload
yes
```



```
en
conf t
int vlan 10
ipv6 address 2001:db8:acad:100::1/64
no shutdown
exit
exit
copy running-config startup-config
```

Configuration gateway:

```
en
conf t
ip-default gateway 192.168.100.254
end
copy run start
```

Med/Assu

Physical Config **CLI** Attributes

IOS Command Line Interface

```
%LINK-5-CHANGED: Interface FastEthernet0/6, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/6, changed state to up

Switch>en
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#hostname Med-Assu
Med-Assu(config)#int vlan 100
Med-Assu(config-if)#ip address 192.168.100.4 255.255.255.0
Med-Assu(config-if)#exit
Med-Assu(config)#sdm prefer dual-ipv4
Med-Assu(config)#sdm prefer dual-ipv4-and-ipv6
% Incomplete command.
Med-Assu(config)#exit
Med-Assu#
%SYS-5-CONFIG_I: Configured from console by console

Med-Assu#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Med-Assu(config)#int vlan 100
Med-Assu(config-if)#exit
Med-Assu(config)#sdm prefer dual
Med-Assu(config)#sdm prefer dual-ipv4-and-ipv6 default
Changes to the running SDM preferences have been stored, but cannot take effect until
the next reload.
Use 'show sdm prefer' to see what SDM preference is currently active.
Med-Assu(config)#exit
Med-Assu#
%SYS-5-CONFIG_I: Configured from console by console

Med-Assu#reload
System configuration has been modified. Save? [yes/no]:
% Please answer 'yes' or 'no'.
System configuration has been modified. Save? [yes/no]:yes
Building configuration...
```

Copy Paste

☐ Top

```

Dir-Exam>en
Dir-Exam#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Dir-Exam(config)#ip default-gateway 192.168.100.254
Dir-Exam(config)#end
Dir-Exam#
%SYS-5-CONFIG_I: Configured from console by console

Dir-Exam#copy run start
Destination filename [startup-config]?
Building configuration...
[OK]
Dir-Exam#

```

Copy

Paste

☐ Top

Dir-Exam

Physical
Config
**CLI**
Attributes

IOS Command Line Interface

```

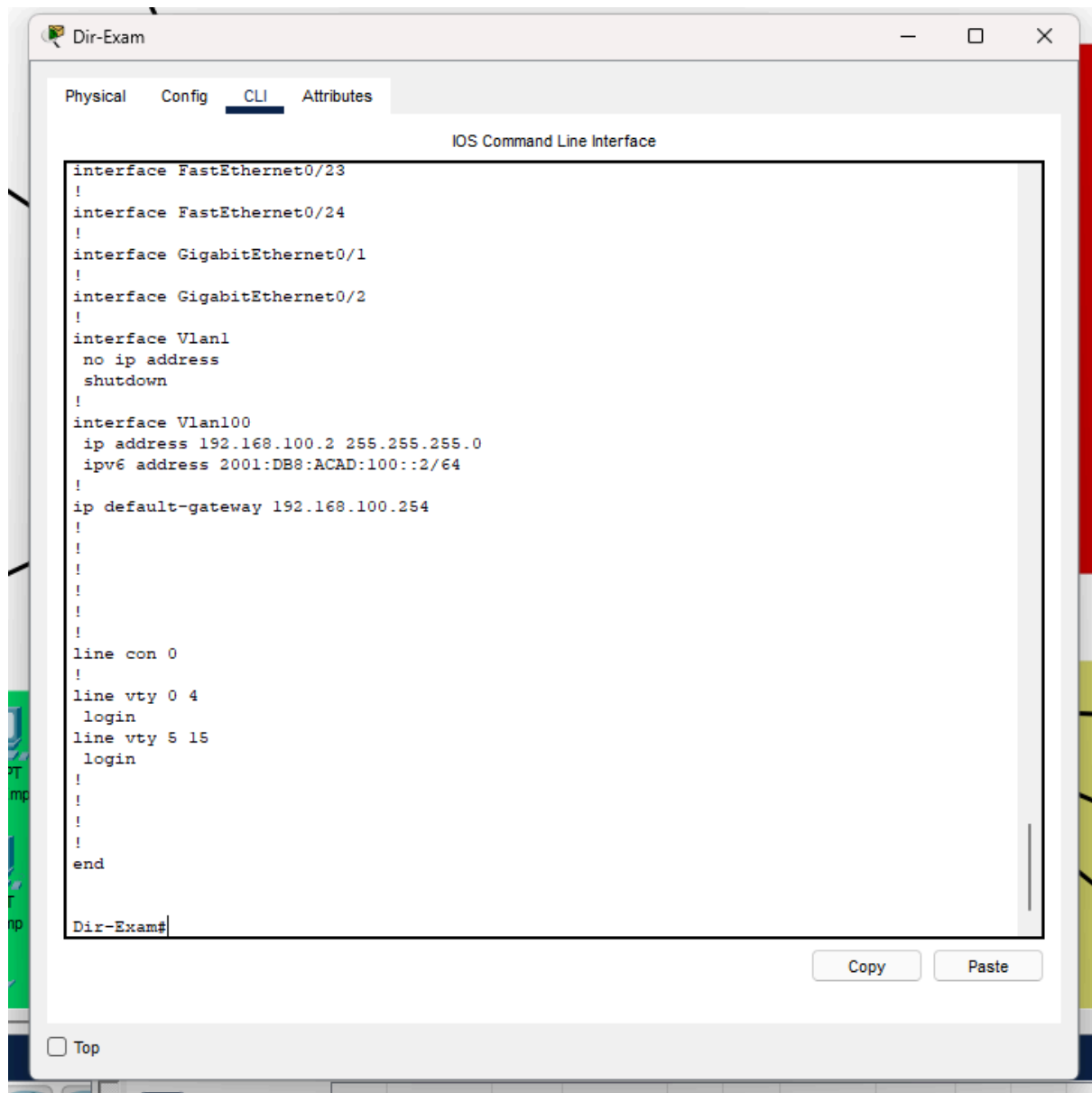
Dir-Exam>en
Dir-Exam#show ip int brief
Interface                IP-Address      OK? Method Status        Protocol
FastEthernet0/1          unassigned      YES manual  up             up
FastEthernet0/2          unassigned      YES manual  up             up
FastEthernet0/3          unassigned      YES manual  up             up
FastEthernet0/4          unassigned      YES manual  up             up
FastEthernet0/5          unassigned      YES manual  up             up
FastEthernet0/6          unassigned      YES manual  up             up
FastEthernet0/7          unassigned      YES manual  down           down
FastEthernet0/8          unassigned      YES manual  down           down
FastEthernet0/9          unassigned      YES manual  down           down
FastEthernet0/10         unassigned      YES manual  down           down
FastEthernet0/11         unassigned      YES manual  down           down
FastEthernet0/12         unassigned      YES manual  down           down
FastEthernet0/13         unassigned      YES manual  down           down
FastEthernet0/14         unassigned      YES manual  down           down
FastEthernet0/15         unassigned      YES manual  down           down
FastEthernet0/16         unassigned      YES manual  down           down
FastEthernet0/17         unassigned      YES manual  down           down
FastEthernet0/18         unassigned      YES manual  down           down
FastEthernet0/19         unassigned      YES manual  down           down
FastEthernet0/20         unassigned      YES manual  down           down
FastEthernet0/21         unassigned      YES manual  down           down
FastEthernet0/22         unassigned      YES manual  down           down
FastEthernet0/23         unassigned      YES manual  down           down
FastEthernet0/24         unassigned      YES manual  down           down
GigabitEthernet0/1       unassigned      YES manual  up             up
GigabitEthernet0/2       unassigned      YES manual  down           down
Vlan1                    unassigned      YES manual  administratively down down
Vlan100                  192.168.100.2  YES manual  down           down
Dir-Exam#
Dir-Exam#
Dir-Exam#
Dir-Exam#

```

Copy

Paste

☐ Top



Pour sécuriser le commutateur, il faut tout d'abord vérifier sa version d'iOS grâce à la commande *show version*, si l'on y voit les caractères K9, il peut être sécurisé en SSH. Je le configure en générant une clé RSA de 1024 octets puis coupant les transmission des 15 premiers ports pour éviter une connexion avec le protocole Telnet qui ne sécurise plus aujourd'hui les données.

Je rajoute également un mot de passe pour que seules les personnes autorisées puissent accéder à la console CLI du commutateur. Je répète l'opération pour les autres commutateurs.

Commandes utilisées:

```

en
show ip ssh
conf t
en secret 1234-Metropole:1234
ip domain-name metropolecg.com
ip ssh version 2
crypto key generate rsa
username admin secret 1234-Metropole:1234
line vty 0 15
login local
transport input ssh
exit
copy run start

```

```

24 FastEthernet interfaces
2 Gigabit Ethernet interfaces
The password-recovery mechanism is enabled.

64K bytes of flash-simulated non-volatile configuration memory.
Base ethernet MAC Address       : 00:D0:BC:8A:6D:A4
Motherboard assembly number     : 73-10390-03
Power supply part number        : 341-0097-02
Motherboard serial number       : FOC10093R12
Power supply serial number      : AZS1007032H
Model revision number           : B0
Motherboard revision number     : B0
Model number                    : WS-C2960-24TT-L
System serial number            : FOC1010X104
Top Assembly Part Number        : 800-27221-02
Top Assembly Revision Number    : A0
Version ID                     : V02
CLEI Code Number                : COM3L00BRA
Hardware Board Revision Number  : 0x01

Switch Ports Model          SW Version  SW Image
-----
*    1 26    WS-C2960-24TT-L  15.0(2)SE4  C2960-LANBASEK9-M

Configuration register is 0xF

Dir-Exam#

```

Copy

Paste

```

Dir-Exam(config)#crypto key generate rsa
The name for the keys will be: Dir-Exam.metropolecg.com
Choose the size of the key modulus in the range of 360 to 4096 for your
General Purpose Keys. Choosing a key modulus greater than 512 may take
a few minutes.

How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]

Dir-Exam(config)#username admin secret 1234-Metropole:1234
*Mar 1 3:56:26.141: %SSH-5-ENABLED: SSH 2 has been enabled
Dir-Exam(config)#line vty 0 15
Dir-Exam(config-line)#login local
Dir-Exam(config-line)#transport input ssh
Dir-Exam(config-line)#exit
Dir-Exam(config)#copy run st
Dir-Exam(config)#copy run start
^
% Invalid input detected at '^' marker.

Dir-Exam(config)#exit
Dir-Exam#
%SYS-5-CONFIG_I: Configured from console by console

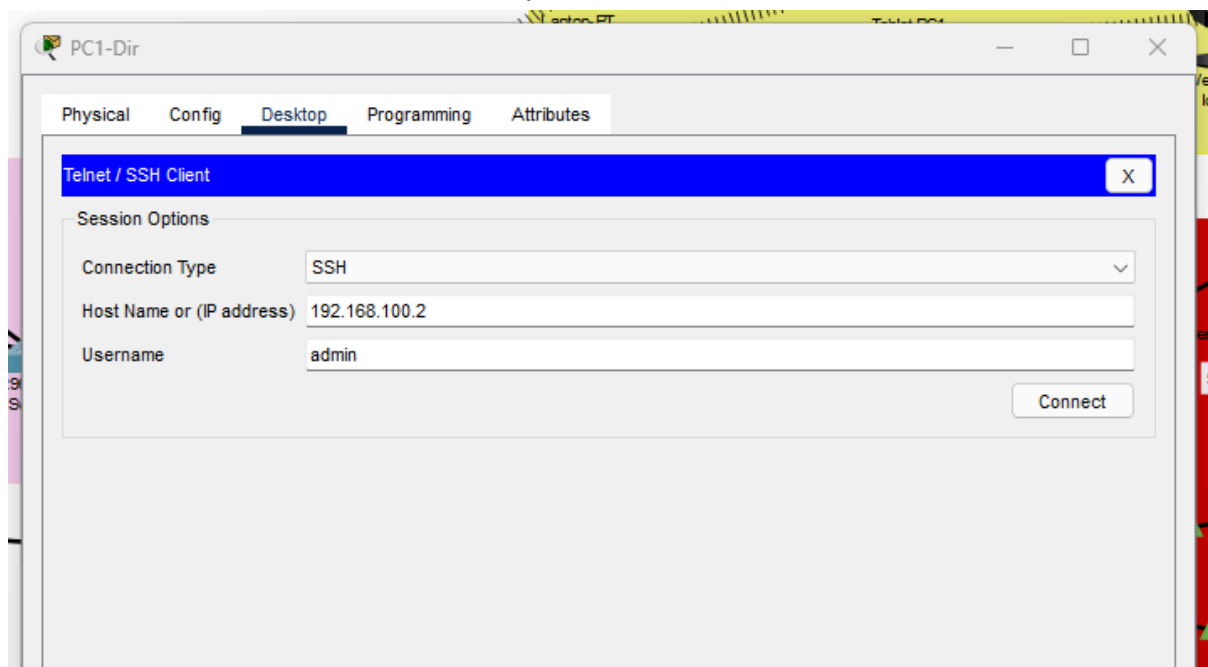
Dir-Exam#copy run start
Destination filename [startup-config]?
Building configuration...
[OK]
Dir-Exam#
Dir-Exam#dis
% Ambiguous command: "dis"
Dir-Exam#disable
Dir-Exam>en
Password:
Dir-Exam#

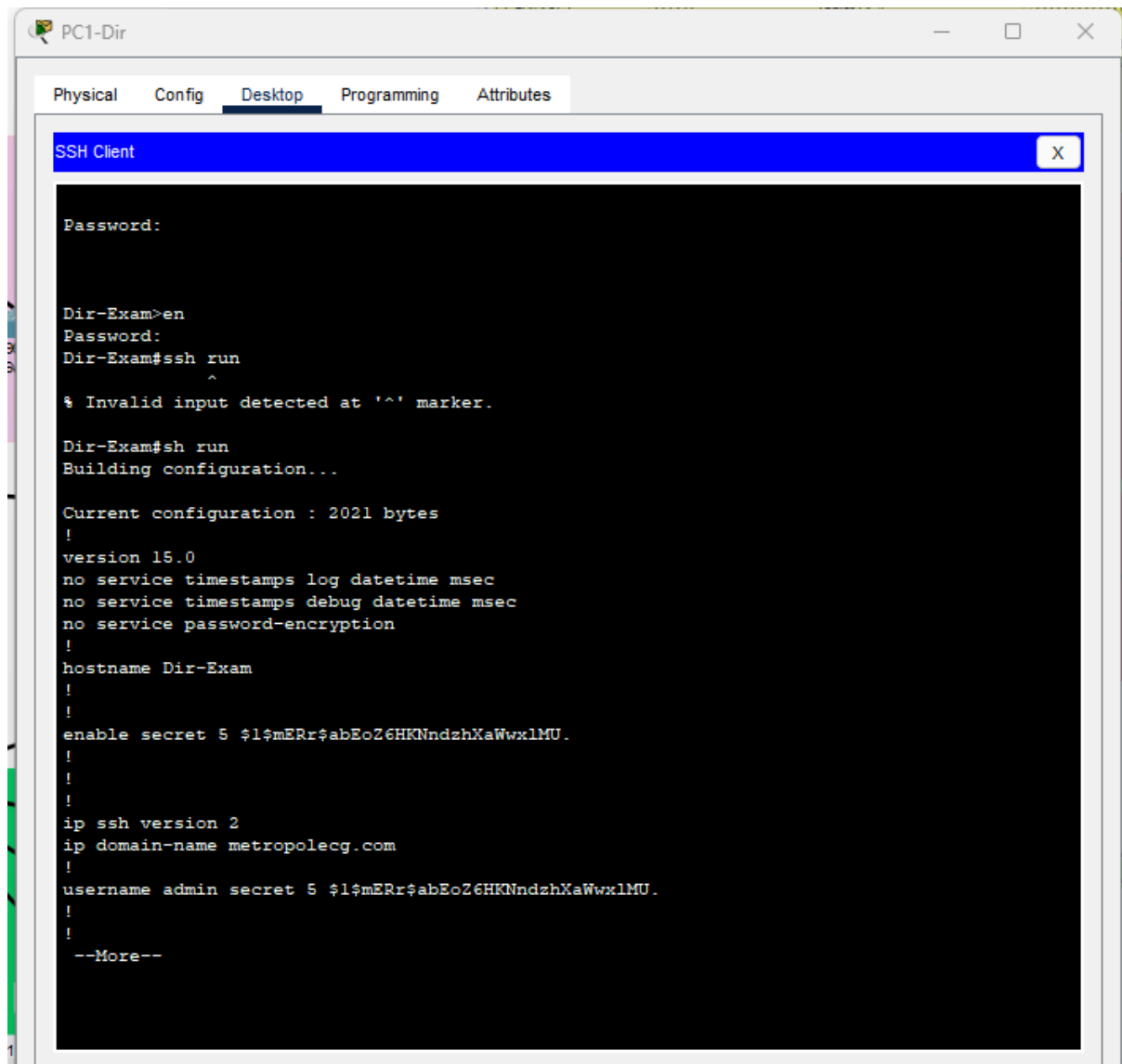
```

Copy

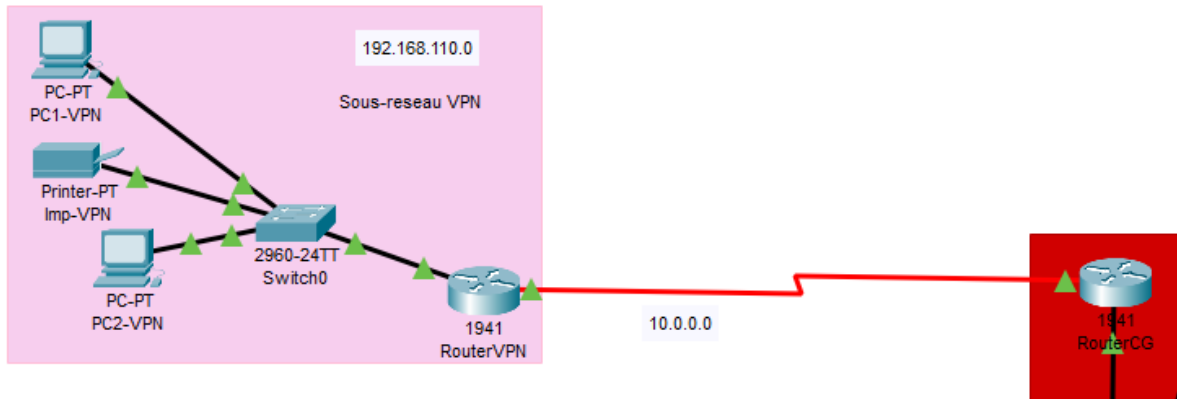
Paste

La connexion SSH fonctionnera lorsque le LAN 100 sera créé.





Je rajoute ensuite le sous-réseau VPN équipé d'un routeur, d'un switch, de deux PC et d'une imprimante et les configure en IPv4 et IPv6. Pour relier le routeur de la salle distante VPN à celui de la salle info, je rajoute une carte WIC afin d'avoir deux ports séries ce qui va me permettre de réaliser des connexions de type WAN entre le routeur VPN et celui de la salle info. J'utilise un câble DCE pour relier les routeurs en série.



Je configure ensuite le routeur VPN et je le sécurise en lui administrant un mot de passe pour le mode privilégié, une configuration du protocole SSH, un autre mot de passe pour l'accès console, un mot de passe pour les lignes VTY pour l'accès en SSH, le chiffrement des mots de passe et la copie de la configuration dans la mémoire non volatile. Sans ces étapes, n'importe qui pourrait avoir accès à la configuration du routeur.

Commandes utilisées:

```

en
conf t
hostname RouteurVPN
enable secret 1234-Metropole:1234
ip ssh version 2
ip domain-name metropolecg.com
username admin secret 1234-Metropole:1234
crypto key generate rsa
line console 0
password 1234-Metropole:1234
line vty 0 15
transport input ssh
login local
exit
service password-encryption

```



```
banner motd #Acces aux Personnes autorisées seulement!#
exit
copy run start
```

Je configure ensuite les interfaces du routeur VPN en lui attribuant une adresse IPV4 et IPv6.

```
en
conf t
int g0/0
ip address 192.168.110.254 255.255.255.0
ipv6 address 2001:db8:acad:110::254/64
no shut
exit

int s0/1/1
ip address 10.0.0.2 255.255.255.0
ipv6 address 2001:db8:acad:1001::2/64
no shutdown
```

Pour simuler Internet, j'utilise une adresse de bouclage sur mon routeur VPN et lui attribue une adresse IPv4.

```
adresse bouclage routeur
interface loopback 0
ip address
exit
```

Je répète ces opérations pour mon routeur de la salle Info en lui attribuant une adresse de bouclage et IPv4/IPv6 différente de mon routeur VPN.

RouterVPN

Physical
Config
**CLI**
Attributes

### IOS Command Line Interface

```

The name for the keys will be: RouteurVPN.metropolecg.com
Choose the size of the key modulus in the range of 360 to 4096 for your
  General Purpose Keys. Choosing a key modulus greater than 512 may take
  a few minutes.

How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]

RouteurVPN(config)#line console 0
*Mar 1 0:11:22.224: %SSH-5-ENABLED: SSH 2 has been enabled
RouteurVPN(config-line)#password 1234-Metropole:1234
RouteurVPN(config-line)#login
RouteurVPN(config-line)#exit
RouteurVPN(config)#line vty 0 15
RouteurVPN(config-line)#transport input ssh
RouteurVPN(config-line)#login local
RouteurVPN(config-line)#exit
RouteurVPN(config)#service password-
RouteurVPN(config)#service password-encryption
RouteurVPN(config)#banner motd "#Acces aux Personnes Autorisees seulement!#
Enter TEXT message. End with the character '#'.
banner motd #Acces aux Personnes Autorisees seulement!#

RouteurVPN(config)#banner motd #Acces aux Personnes Autorisees Seulement!#
RouteurVPN(config)#exit
RouteurVPN#
%SYS-5-CONFIG_I: Configured from console by console

RouteurVPN#copy run start
Destination filename [startup-config]?
Building configuration...
[OK]
RouteurVPN#exit

```

CopyPaste

```

Press RETURN to get started!

Acces aux Personnes Autorisees Seulement!

User Access Verification

Password:

```

CopyPaste

```

RouteurVPN>en
Password:
RouteurVPN#conf t
Enter configuration commands, one per line. End with CNTL/Z.
RouteurVPN(config)#int g0/0
RouteurVPN(config-if)#ip address 192.168.110.254 255.255.255.0
RouteurVPN(config-if)#ipv6 address 2001:db8:acad:110::254/64
RouteurVPN(config-if)#no shut

RouteurVPN(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up

```

```

RouteurVPN#show ip interface brief

```

Interface	IP-Address	OK?	Method	Status	Protocol
GigabitEthernet0/0	192.168.110.254	YES	manual	up	up
GigabitEthernet0/1	unassigned	YES	unset	administratively down	down
Serial0/1/0	unassigned	YES	unset	administratively down	down
Serial0/1/1	10.0.0.2	YES	manual	up	up
Loopback0	192.168.200.2	YES	manual	up	up
Vlan1	unassigned	YES	unset	administratively down	down

```

RouteurVPN#show ipv6 interface brief
GigabitEthernet0/0      [up/up]
    FE80::20C:85FF:FE0C:B701
    2001:DB8:ACAD:110::254
GigabitEthernet0/1      [administratively down/down]
    unassigned
Serial0/1/0              [administratively down/down]
    unassigned
Serial0/1/1              [up/up]
    FE80::20C:85FF:FE0C:B701
    2001:DB8:ACAD:1001::2
Loopback0                [up/up]
    unassigned
Vlan1                    [administratively down/down]
    unassigned
RouteurVPN#

```

Copy

Paste

```

version 15.1
no service timestamps log datetime msec
no service timestamps debug datetime msec
service password-encryption
!
hostname RouteurVPN
!
!
!
enable secret 5 $1$mERr$abEoZ6HKNndzhXaWwx1MU.
!
!
!
!
!
!
!

```

```
username admin secret 5 $1$mERr$abEoZ6HKNndzhXaWwx1MU.  
!  
!  
license udi pid CISC01941/K9 sn FTX15248Z9K-  
!  
!|  
!  
!  
!  
!  
!  
!  
!  
!  
ip ssh version 2  
ip domain-name metropolecg.com  
!  
!  
spanning-tree mode pvst  
!  
!  
!  
!  
!  
!  
interface GigabitEthernet0/0  
description Lien Sous-Reseau VPN  
ip address 192.168.110.254 255.255.255.0  
duplex auto  
speed auto
```

```

ipv6 address 2001:DB8:ACAD:110::254/64
!
interface GigabitEthernet0/1
  no ip address
  duplex auto
  speed auto
  shutdown
!
interface Serial0/1/0
  no ip address
  clock rate 2000000
  shutdown
!
interface Serial0/1/1
  description Lien RouteurVPN-RouteurCG
  ip address 10.0.0.2 255.255.255.0
  ipv6 address 2001:DB8:ACAD:1001::2/64
!
interface Vlan1
  no ip address
  shutdown
!
ip classless
!
ip flow-export version 9
!
!
!
banner motd #Acces aux Personnes Autorisees Seulement!#
!
!
!
!
line con 0
  password 7 08701E1D5D542812061903142527217262677147
  login
!
line aux 0
!
line vty 0 4
  login local
  transport input ssh
line vty 5 15
  login local
  transport input ssh
!
!
!
end

```

## Routage du Routeur VPN et du Routeur de l'entreprise

Etant donné que mon routeur VPN se situe en extrémité du réseau, pour aller ailleurs que dans le sous-réseau VPN, je dois passer par une passerelle donc rajouter une route par défaut. Je rajoute une route en IPv4 et IPv6.

Accès aux Personnes Autorisées seulement!

```
RouteurCG>en
Password:
RouteurCG#conf t
Enter configuration commands, one per line. End with CNTL/Z.
RouteurCG(config)#ip route 192.168.110.0 255.255.255.0 10.0.0.2
RouteurCG(config)#ip route 0.0.0.0 0.0.0.0 192.168.10.2
RouteurCG(config)#ipv6 unicast-routing
RouteurCG(config)#ipv6 route 2001:db8:acad:110::0/64 2001:db8:acad:1001::2
RouteurCG(config)#ipv6 route ::/0 2001:db8:acad:10::2
RouteurCG(config)#
```

Copy

Paste

```
RouteurVPN#show running-config | include up
duplex auto
duplex auto
RouteurVPN#show ip interface brief | include up
GigabitEthernet0/0    192.168.110.254 YES manual up
Serial0/1/1          10.0.0.2 YES manual up
Loopback0            192.168.200.2 YES manual up
RouteurVPN#show ip interface brief | exclude unassigned
Interface            IP-Address      OK? Method Status Protocol
GigabitEthernet0/0    192.168.110.254 YES manual up
Serial0/1/1          10.0.0.2 YES manual up
Loopback0            192.168.200.2 YES manual up
RouteurVPN#show ip route | begin Gateway
Gateway of last resort is 10.0.0.1 to network 0.0.0.0

    10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.0.0.0/24 is directly connected, Serial0/1/1
L       10.0.0.2/32 is directly connected, Serial0/1/1
    192.168.110.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.110.0/24 is directly connected, GigabitEthernet0/0
L       192.168.110.254/32 is directly connected, GigabitEthernet0/0
    192.168.200.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.200.0/24 is directly connected, Loopback0
L       192.168.200.2/32 is directly connected, Loopback0
S*     0.0.0.0/0 [1/0] via 10.0.0.1
RouteurVPN#
```

Pour la configuration des routes du routeur de l'entreprise, je dois rajouter les IP du réseau distant.

commandes utilisées:

```
ip route 192.168.110.0 255.255.255.0 10.0.0.2
ip route 0.0.0.0 0.0.0.0 192.168.10.2
```

```
ipv6 unicast-routing
ipv6 route 2001:db8:acad:110::0/64 2001:db8:acad:1001::2
ipv6 route ::0 2001:db8:acad:10::2
```

```
RouteurCG#show ip route | begin Gateway
Gateway of last resort is 192.168.10.2 to network 0.0.0.0

 10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C    10.0.0.0/24 is directly connected, Serial0/1/0
L    10.0.0.1/32 is directly connected, Serial0/1/0
 192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
C    192.168.10.0/24 is directly connected, GigabitEthernet0/0
L    192.168.10.1/32 is directly connected, GigabitEthernet0/0
S    192.168.110.0/24 [1/0] via 10.0.0.2
 192.168.200.0/24 is variably subnetted, 2 subnets, 2 masks
C    192.168.200.0/24 is directly connected, Loopback0
L    192.168.200.1/32 is directly connected, Loopback0
S*   0.0.0.0/0 [1/0] via 192.168.10.2

RouteurCG#
```

Conv

Paste

Et enfin, pour le routage du commutateur de niveau 3, je configure les adresses IPv4 et IPv6 et j'active le routage IPv4. Le routage avec les différents réseaux de l'entreprise sera assuré grâce au routage inter-VLAN.

Commandes utilisées:

```
en
conf t
hostname SwitchL3
interface g1/0/1
no switchport
ip address 192.168.10.2 255.255.255.0
ipv6 address 2001:db8:acad:10::2/64
exit
ip route 0.0.0.0 0.0.0.0 192.168.10.1
ipv6 unicast-routing
```

```

SwitchL3>en
SwitchL3#show ip route | begin Gateway
Gateway of last resort is 192.168.10.1 to network 0.0.0.0

C    192.168.10.0/24 is directly connected, GigabitEthernet1/0/1
C    192.168.20.0/24 is directly connected, Vlan20
C    192.168.21.0/24 is directly connected, Vlan21
C    192.168.22.0/24 is directly connected, Vlan22
C    192.168.23.0/24 is directly connected, Vlan23
C    192.168.24.0/24 is directly connected, Vlan24
C    192.168.25.0/24 is directly connected, Vlan25
C    192.168.27.0/24 is directly connected, Vlan27
C    192.168.30.0/24 is directly connected, Vlan30
C    192.168.40.0/24 is directly connected, Vlan40
C    192.168.50.0/24 is directly connected, Vlan50
C    192.168.60.0/24 is directly connected, Vlan60
C    192.168.100.0/24 is directly connected, Vlan100
S*   0.0.0.0/0 [1/0] via 192.168.10.1

SwitchL3#

```

VLAN mis en place (pour éviter les domaines de collision):

Pour éviter des collisions et réduire la congestion du réseau, je dois mettre en place des VLANs pour segmenter le réseau afin de minimiser les risques de collisions de paquets et une bande passante ralentie. Cette mise-en-place permet également une gestion simplifiée du réseau car elle regroupe de façon logique un ensemble de machine et/ou d'équipements, une réduction de coût et la réduction de périphériques dans le domaine de diffusion.

Commandes utilisées: (je répète l'opération pour tous les VLANs du réseau).

```

configure terminal
vlan 20
name Direction
end

```

Je commence donc par paramétrer les VLAN du commutateur de niveau 3, puis configurer les VLAN des commutateurs de niveau 2 sans oublier de les attribuer à une interface.

Pour la configuration, je dois entrer dans l'interface que je souhaite configurer, puis j'autorise l'accès à LAN et spécifie à quel LAN je veux relié cette interface.



SwitchL3#show vlan brief

VLAN Name	Status	Ports
1 default	active	Gig1/0/10, Gig1/0/11, Gig1/0/12, Gig1/0/13 Gig1/0/17 Gig1/0/21 Gig1/1/1
20 Direction	active	Gig1/1/2, Gig1/1/3, Gig1/1/4
21 Examen/Concours	active	
22 Paie/DRH	active	
23 Emploi	active	
24 Medecine	active	
25 Assurance	active	
27 Info/RGPD	active	
30 Serveurs	active	Gig1/0/6, Gig1/0/7, Gig1/0/8
40 Impression	active	
50 Telephonie	active	
60 Wifi	active	Gig1/0/9
100 Administration	active	
1002 fddi-default	active	
1003 token-ring-default	active	
--More--		

Copy

Paste

Multilayer Switch0

Physical Config CLI Attributes

IOS Command Line Interface

```
%LINK-5-CHANGED: Interface Vlan20, changed state to up
%LINK-5-CHANGED: Interface Vlan21, changed state to up
%LINK-5-CHANGED: Interface Vlan22, changed state to up
%LINK-5-CHANGED: Interface Vlan23, changed state to up
%LINK-5-CHANGED: Interface Vlan24, changed state to up
%LINK-5-CHANGED: Interface Vlan25, changed state to up
%LINK-5-CHANGED: Interface Vlan27, changed state to up
%LINK-5-CHANGED: Interface Vlan30, changed state to up
%LINK-5-CHANGED: Interface Vlan40, changed state to up
%LINK-5-CHANGED: Interface Vlan50, changed state to up
%LINK-5-CHANGED: Interface Vlan60, changed state to up
%LINK-5-CHANGED: Interface Vlan100, changed state to up
%LINK-5-CHANGED: Interface GigabitEthernet1/0/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0/1, changed state to up
%LINK-5-CHANGED: Interface GigabitEthernet1/0/2, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0/2, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan20, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan21, changed state to up
```

Copy Paste

```
Dir-Exam#show vlan brief
```

VLAN	Name	Status	Ports
1	default	active	Fa0/1, Fa0/2, Fa0/3, Fa0/4 Fa0/5, Fa0/6, Fa0/7, Fa0/8 Fa0/9, Fa0/10, Fa0/11, Fa0/12 Fa0/13, Fa0/14, Fa0/15, Fa0/16 Fa0/17, Fa0/18, Fa0/19, Fa0/20 Fa0/21, Fa0/22, Fa0/23, Fa0/24 Gig0/1, Gig0/2
20	Direction	active	
21	Examen/Concours	active	
40	Impression	active	
50	Telephonie	active	
100	Administration	active	
1002	fddi-default	active	
1003	token-ring-default	active	
1004	fddinet-default	active	
1005	trnet-default	active	

```
Dir-Exam#
```

Copy

Paste

☐ Top

## Attribution des VLANs:

```
configure terminal
```

```
int fa0/1
```

```
switchport mode access
```

```
switchport access vlan 20
```

```
end
```

```
%LINK-3-UPDOWN: Interface Vlan100, changed state to down
%LINK-5-CHANGED: Interface Vlan100, changed state to up
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan100, changed state to up
%LINK-5-CHANGED: Interface FastEthernet0/3, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/3, changed state to up
%LINK-5-CHANGED: Interface FastEthernet0/6, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/6, changed state to up
%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
%LINK-5-CHANGED: Interface FastEthernet0/2, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/2, changed state to up
%LINK-5-CHANGED: Interface FastEthernet0/4, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/4, changed state to up
%LINK-5-CHANGED: Interface FastEthernet0/5, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/5, changed state to up
```

En ce qui concerne la Téléphonie, je vais utiliser des téléphones VoIP. Je vais ainsi devoir ajouter un VLAN voix, et des ports Trunk car avec les ports Trunk, plusieurs VLAN peuvent être autorisés sur un port.

Pour activer le VLAN Téléphonie, je dois activer la qualité de service pour assurer un bon trafic sur les interfaces du commutateur puis j'ajoute le VLAN correspondant à la Téléphonie afin d'attribuer cette interface au VLAN.

Configuration Téléphonie (sur les commutateurs):

```
% Invalid input detected at '^' marker.

Dir-Exam(config)#int f0/1
Dir-Exam(config-if)#mls qos trust cos
Dir-Exam(config-if)#switchport voice vlan 50
Dir-Exam(config-if)#exit
Dir-Exam(config)#
```

Copy

Paste

☐ Top

L'ajout d'un Trunk de VLAN permet d'acheminer le trafic pour tous les VLANs autorisés. Je dois donc configurer le port d'interconnexion pour activer la liaison Trunk.

```
int g0/1
switchport mode trunk
switchport trunk native vlan 100
switchport trunk allowed vlan 20,21,40,50,100
end
```

```
show int g0/1 switchport
```

```
Dir-Exam#show interface g0/1 switchport
Name: Gig0/1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 100 (Administration)
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk private VLANs: none
Operational private-vlan: none
Trunking VLANs Enabled: 20-21,40,50,100
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL
Protected: false
--More--
```

## Routage Inter-VLAN:

Pour faire le routage inter VLAN, je dois avoir des VLAN créé dans le commutateur de niveau 3, affecter ces VLAN aux différentes interfaces et autoriser ces VLAN sur ces liaisons Trunk.

```
vlan 20
name Direction
exit
```

```
interface g1/0/2
switchport trunk encapsulation dot1q
switchport mode trunk
switchport trunk native vlan 100
switchport trunk allowed vlan 20,21,40,50,100
```

```

int g1/0/6
switchport mode access
switchport access vlan 30
exit

```

```

SwitchL3#show interfaces trunk

```

Port	Mode	Encapsulation	Status	Native vlan
Gig1/0/2	on	802.1q	trunking	100
Gig1/0/3	on	802.1q	trunking	100
Gig1/0/4	on	802.1q	trunking	100
Gig1/0/5	on	802.1q	trunking	100

```

Port      Vlans allowed on trunk
Gig1/0/2  20-21,40,50,100
Gig1/0/3  22-23,40,50,100
Gig1/0/4  24-25,40,50,100
Gig1/0/5  27,40,50,100

```

```

Port      Vlans allowed and active in management domain
Gig1/0/2  20,21,40,50,100
Gig1/0/3  22,23,40,50,100
Gig1/0/4  24,25,40,50,100
Gig1/0/5  27,40,50,100

```

```

Port      Vlans in spanning tree forwarding state and not pruned
Gig1/0/2  20,21,40,50,100
Gig1/0/3  22,23,40,50,100
Gig1/0/4  24,25,40,50,100
--More--

```

Enfin, je vais créer des SVI (interfaces virtuelles) pour terminer le routage:

```

interface vlan 20
description Passerelle SVI Direction
ip address 192.168.20.254 255.255.255.0
ipv6 address 2001:db8:acad:20::254/64
no shutdown
exit

```

Sauvegarde de Configuration Cisco Packet Tracer avec TFTP:

```

RouteurVPN#copy running-config tftp
Address or name of remote host []? 192.168.30.3
Destination filename [RouteurVPN-config]?

Writing running-config.....!!
[OK - 1412 bytes]

1412 bytes copied in 7.036 secs (200 bytes/sec)
RouteurVPN#

```

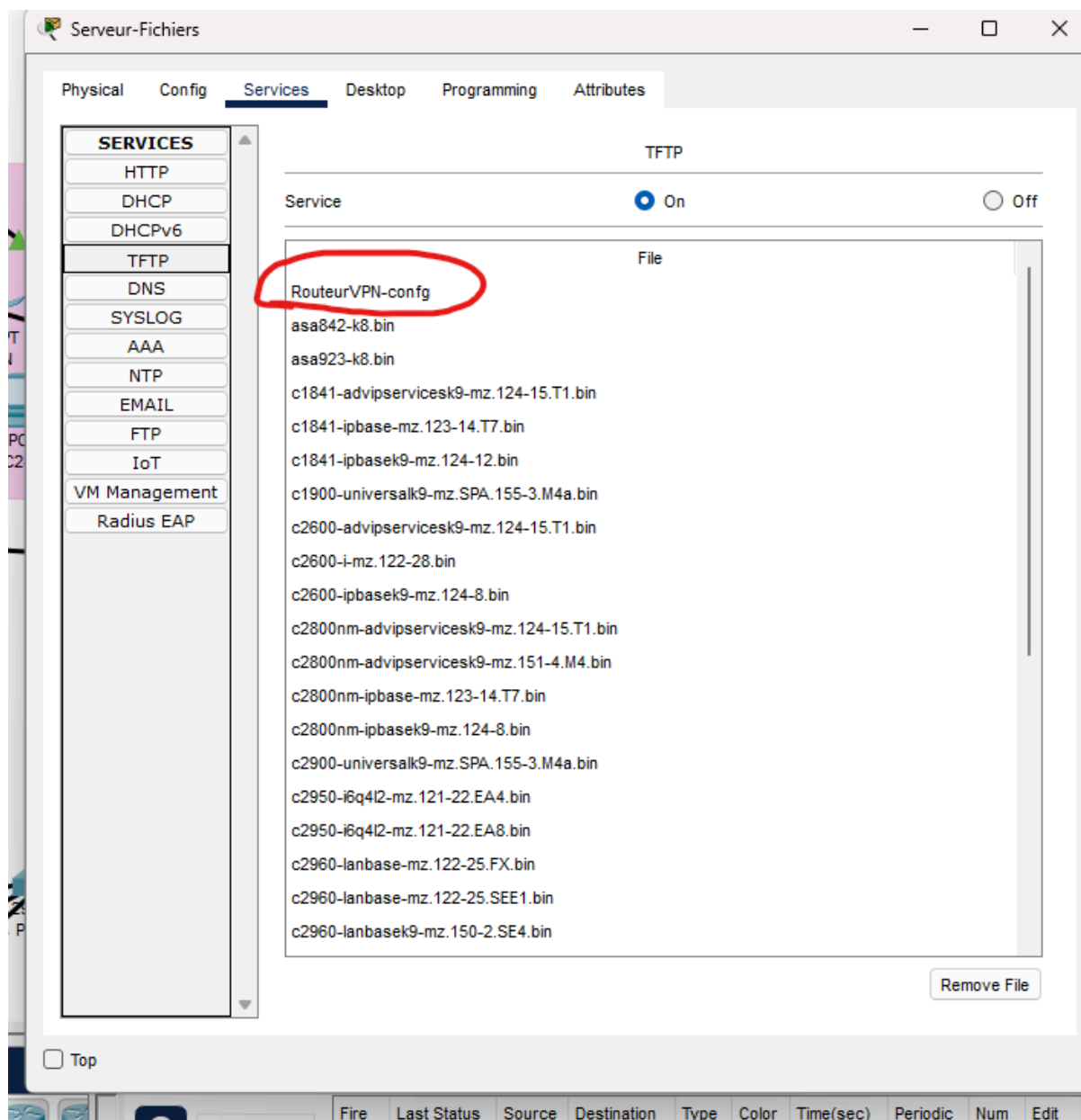
Copy

Paste

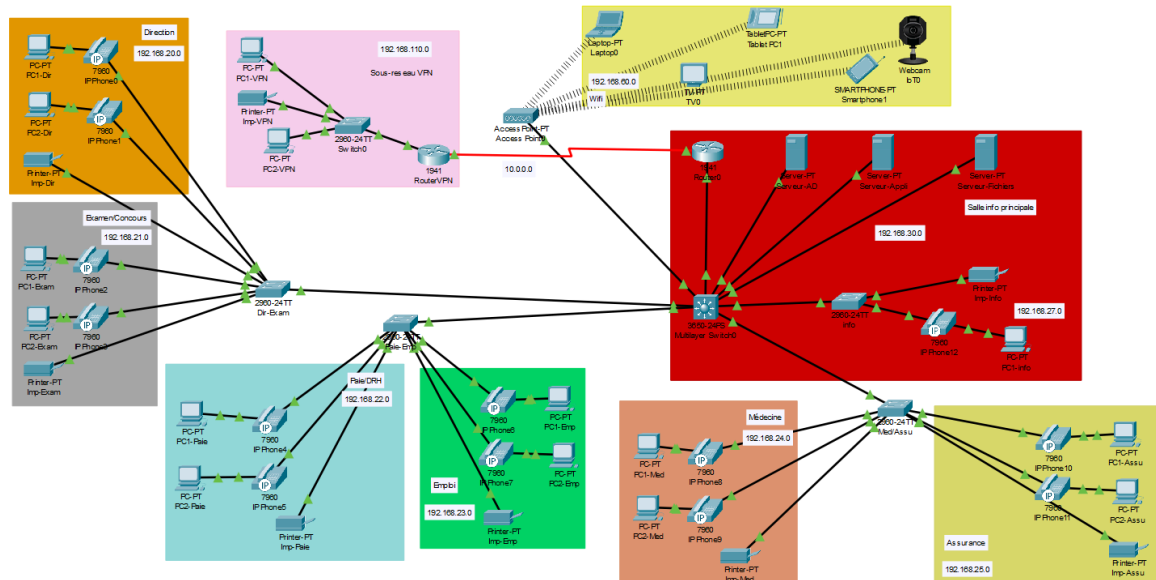
Sauvegarde de configuration sur un serveur TFTP:

Pour sauvegarder et restaurer nos configurations dans Cisco, nous pouvons le faire soit en ligne de commande où une seule commande suffit pour sauvegarder même lorsque le routeur est éteint, soit avec l'interface config de la machine où il y a un bouton avec marqué export.

On peut également sauvegarder et restaurer nos configurations avec un serveur TFTP. Pour copier nos informations dans un serveurs, il suffit d'aller dans la CLI du routeur où l'on veut copier nos info et utiliser la commande `copy running-config tftp` et ensuite d'écrire l'adresse ipv4 où l'on veut sauvegarder ces données.



Je vous présente donc mon schéma réseau finale:



Plan d'adressage réseau mis en place:

Groupe	VLAN	Adresse Réseau	Passerelle Réseau
Direction	20	192.168.20.0/24	192.168.20.254
Examen/concours	21	192.168.21.0/24	192.168.21.254
Paie/DRH	22	192.168.22.0/24	192.168.22.254
Emploi	23	192.168.23.0/24	192.168.23.254
Médecine	24	192.168.24.0/24	192.168.24.254
Assurance	25	192.168.25.0/24	192.168.25.254
Info/RGPD	27	192.168.27.0/24	192.168.27.254
Serveurs	30	192.168.30.0/24	192.168.30.254
Impression	40	192.168.40.0/24	192.168.40.254
Téléphone	50	192.168.50.0/24	192.168.50.254



WIFI	60	192.168.60.0/24	192.168.60.254
Administration	100	192.168.100.0/24	192.168.100.254