Cahier des Charges Technique - Projet Réseaux Avancés & SDN

Cahier des Charges Technique - Projet Réseaux Avancés Cloud & SDN

**Objectif du projet**

Concevoir, implémenter et tester une architecture réseau avancée dans un contexte Cloud privé ou hybride. Les étudiants travailleront en équipe, en mode projet agile, avec une infrastructure virtualisée incluant des technologies SDN, routage dynamique, overlay, et outils de supervision.

**Livrables principaux**

- Cahier des charges fonctionnel et technique (initial)  
- Planning (Gantt/Kanban)  
- Dossier de conception (architecture logique et physique)  
- Scripts et fichiers de configuration (Ansible, Vagrant, Netplan, Ryu, FRR, etc.)  
- Rapport de tests et tableau de bord  
- Rapport final avec retour d'expérience  
- Soutenance orale + démo technique (optionnelle : vidéo)

**Infrastructure à mettre en place (lab Technique)**

**Minimum requis :  
- 3 à 4 VM provisionnées via Vagrant  
- Routage dynamique via FRRouting (BGP ou OSPF)  
- Switch SDN avec Open vSwitch et contrôleur Ryu  
- Monitoring avec Prometheus + Grafana  
- Scripts d’automatisation (Bash, Ansible)**

**Scénario 1 : SDN + OSPF + Vagrant (automatisation)**

**Objectif pédagogique**

Mettre en place une infrastructure réseau automatisée avec Vagrant intégrant :

* Un plan de données programmable via SDN (Open vSwitch + OpenFlow)
* Un plan de contrôle dynamique avec OSPF (FRRouting)
* Une automatisation complète des déploiements et configurations

**Objectifs techniques**

* Créer 3 à 4 machines virtuelles via Vagrant :
  + 1 contrôleur SDN (Ryu)
  + 2 routeurs OSPF (FRRouting ou VyOS)
  + 1 ou 2 hôtes clients Linux
* Utiliser Vagrant pour déployer automatiquement :
  + L’ensemble des VM
  + La configuration initiale d’Open vSwitch et des daemons FRR
  + Les fichiers de routage (frr.conf) et scripts de test
* Implémenter un échange OSPF entre les routeurs
* Connecter les switches OVS à un contrôleur Ryu et y injecter des règles OpenFlow
  + Exemple : rediriger le trafic HTTP vers une route spécifique

**Livrables attendus**

* Fichier Vagrantfile + scripts provisionning (Ansible/Bash)
* Configurations frr.conf OSPF sur chaque nœud
* Script Python Ryu de règles OpenFlow
* Capture de ip route, vtysh, ovs-ofctl
* Journal de bord technique + schéma réseau

**Scénario 2 : pfSense + OSPF + VPN (NVA complète et interconnexion sécurisée)**

**Objectif pédagogique**

Déployer une appliance réseau virtuelle (NVA) basée sur pfSense, intégrant :

* OSPF pour l’échange de routes dynamiques avec un autre routeur
* Un tunnel VPN site-à-site sécurisé (Wireguard ou IPSec)

Infrastructure cible

| VM | Rôle | Contenu principal |
| --- | --- | --- |
| pfSense | NVA (Firewall + OSPF + VPN) | FRRouting intégré, configuration via GUI |
| Site A | Client / réseau local | Linux, routage , monitoring |
| Site B | Réseau distant (VPN peer) | Routeur Linux (FRR) ou pfSense secondaire |
| (Optionnel) | Serveur Prometheus/Grafana | Supervision réseau |

**Étapes clés**

* Déployer pfSense et les hôtes avec Vagrant
* Activer FRRouting dans pfSense pour gérer OSPF
* Créer un tunnel VPN sécurisé entre Site A ↔ Site B
* Ajouter des règles de firewall/NAT pour contrôler les flux
* Intégrer Prometheus ou NetFlow pour superviser les routes/tunnels

**Livrables attendus**

* Export de config XML pfSense
* Captures de la GUI (OSPF, VPN, Firewall)
* Tableau de test : routage, VPN, failover
* Schéma réseau clair
* Rapport d’analyse : sécurité, convergence, politique de routage

**Thématiques intégrées :  
-** IP statique et netplan  
- Overlay VXLAN ou GRE  
- Service Mesh (optionnel)  
- VPN entre VMs ou avec l’extérieur (Wireguard ou IPSec)

**Liste de tâches par équipe (rôles suggérés)**

Chef de projet :  
- Création et suivi du planning (outil : Trello/Notion) Lamya/Ibrahima  
- RACI des rôles Lamya/Yve

Architecte réseau :  
- Conception réseau logique et physique Dimitri  
- Choix des protocoles (BGP/OSPF, VXLAN, etc.) Dimitri

Intégrateur :  
- Mise en place de l’infra avec Vagrant/Proxmox Dimitri   
- Configuration Netplan, IP route Dimitri

DevOps Réseau :  
- Déploiement FRR, Open vSwitch, Ryu Dimitri- Lamya en backup  
- Automatisation via Ansible Vagrant etc .. Dimitri

Analyste/Testeur :  
- Configuration de tests de performance (iperf, ping, latence) Dimitri Lamya Mohamed  
- Wireshark, tcpdump, analyse de routage (vtysh) Dimitri Lamya Ibrahima

Rédacteur/documentaliste :  
- Synthèse des choix techniques Dimitri Ibrahima   
- Rédaction du rapport final Ibrahima/Lamya

Questions à traiter (dans le rapport ou en soutenance)

1. Quelle est la différence entre un switch SDN et un switch classique ?  
2. Pourquoi préférer BGP à OSPF dans une interconnexion multi-cloud ?  
3. Quels sont les risques de convergence lente en routage dynamique ?  
4. Comment monitorer l’état du réseau efficacement ?  
5. Quelle est la place de la sécurité dans un réseau SDN ?  
6. Quelles alternatives open source à Istio connaissez-vous ?  
7. Peut-on déployer cette architecture sur une offre cloud publique (AWS/Azure) ?  
8. Comment faire évoluer cette maquette vers un POC réel d’entreprise ?

**Extensions possibles**

- Intégration de Linkerd ou Istio avec un cluster Kubernetes  
- Simulation de défaillance de lien et convergence automatique  
- CI/CD d’infra avec GitLab CI ou Jenkins  
- Déploiement d’une application web distribuée entre plusieurs zones  
- Tunnel chiffré entre 2 réseaux via Wireguard ou OpenVPN