Réseaux et Télécommunications
SAÉ201
2023/2024

IUT1 Grenoble
Université Grenoble Alpes

Construire un réseau informatique pour une petite structure

1 INTRODUCTION

Le but de cette SAÉ est de mettre en place un réseau informatique avec la configuration de différents services sur des machines Centos8. Nous utiliserons pour cela la plateforme Openstack vue dans le TP d'introduction.

- Le nombre d'heures de projet de cette SAÉ est de 12H, répartie en 3 séances de 4H.
- La mise en place du réseau s'effectuera en binôme ; vous devrez donc collaborer pour mettre en place les éléments qui sont demandés dans le cahier des charges.
- Vous devrez vous appuyer sur les connaissances acquises lors des TPs réalisés dans la ressource R203 ainsi que lors du TP d'introduction à l'environnement Openstack.
- Une évaluation individuelle de TP sera réalisée à la suite de ces trois séances : cette évaluation portera sur le travail réalisé lors des 3 séances de projet.

Le planning ainsi que les éléments évalués et échéances sont sur Chamilo (Ressource SAÉ201).

2 CAHIER DES CHARGES POUR UN BINÔME

Attention ; le travail à réaliser ne commence qu'à la **partie 3**. Cette partie 2 constitue seulement une description du cahier des charges.
Les deux membres d'un binôme sont membres d'un projet nommé **proj-gxby**. Les logins agalan des deux membres du binôme sont notés **agalan1** et **agalan2** dans la suite du cahier des charges. Il faudra donc les remplacer par vos logins agalan respectifs en respectant l'ordre alphabétique (exemple : agalan**1** : **m**ontaigm et agalan**2** : **r**abelaif)
ATTENTION : il est indispensable de respecter (à la lettre près) les noms imposés dans la suite du cahier des charges et qui utilisent **agalan1**, **agalan2** et **proj-gxby**. Des tests de validation automatiques seront effectués avec ces noms et la validation ne sera pas accordée en cas d'erreur de saisie.

2.1 Instances

La structure à réaliser dans l'environnement Openstack, sera la suivante :

- 2 instances **« Windows 10 »**, une par personne, sur un réseau nommé **NET0 : les FQDNs seront w10a-agalan1.proj-gxby et w10b-agalan2.proj-gxby.**
- 1 instance de type « CentosStream8 CLI », sur un réseau nommé **NET1 et de FQDN serva-agalan1.proj-gxby.**
- 1 instance de type « CentosStream8 CLI », sur un réseau nommé **NET2 et de FQDN servb-agalan2.proj-gxby.**

Les deux instances « Windows 10 » présentes sur le réseau **NET0** seront accessibles par RDP depuis la salle de TP via des IP flottantes, à prendre sur le réseau **EXT_UGA**.
L'accès SSH aux instances Linux présentes sur les réseaux **NET1** et **NET2** s'effectuera ensuite depuis les instances « Windows 10 » présentes quant à elles sur le réseau **NET0**.
Sur l'instance **serva-agalan1.proj-gxby** présente sur **NET1,** on installera un serveur FTP (logiciel **vsftpd**) et un serveur DNS (logiciel **dnsmasq**).
Sur l'instance **servb-agalan2.proj-gxby** présente sur **NET2**, on installera un serveur WEB (logiciel **apache**) hébergeant, 1 site de test, 2 sites personnels (un par membre du binôme) et un site NextCloud hébergeant des fichiers (images, documents) partagés entre des utilisateurs.

Les sites web seront accessibles depuis vos instances « Windows 10 » avec des noms enregistrés dans votre DNS : **servb-agalan2.proj-gxby, www-agalan1.proj-gxby**, **www-agalan2.proj-gxby**, **nextcloud.proj-gxby**.
Le serveur FTP sera quant à lui accessible avec le nom : **ftp.proj-gxby**.
Si tous les services précédents fonctionnent, on installera sur **serva-agalan1.proj-gxby** les logiciels Prometheus et Grafana permettant d'obtenir un tableau de bord affichant les ressources utilisées sur les deux serveurs Linux

2.2 Réseaux et routeurs

- Le réseau **Ext_UGA** a comme adresse **152.77.180.0/24** et la passerelle par défaut pour la sortie sur « Internet » depuis ce réseau est un routeur d'adresse **152.77.180.1 (déjà configuré pour Ext_UGA dans Openstack).**
- Pour **NET0,** le sous-réseau **subNET0** sera un réseau **/24** d'adresse **10.x.y.0/24**.
- Un routeur **GW0** faisant de la NAT permettra de relier **NET0** au réseau externe **EXT_UGA.**
- Pour **NET1 et NET2,** les 2 sous-réseaux **subNET1** et **subNET2** seront des réseaux **/24** d'adresse **192.168.(10x+y).0/24 et 192.168.(20x+y).0/24**. (**10x** signifiant 10 multiplié par **x**).
- 2 routeurs nommés **GW1** et **GW2** feront le lien entre le réseau **NET0** et les 2 réseaux **NET1** et **NET2**. Les deux routeurs seront configurés **sans NAT** (Pas de translation d'adresse entre les trois réseaux privés)

3 TRAVAIL A REALISER

Le projet se décompose en plusieurs étapes à réaliser dans l'ordre. Les TPs de R203 « Bases des services Réseaux » permettent la mise en place des services jusqu'à l'étape 6. Les étapes 7 et 8 demandent quant à elles de se documenter par ailleurs.

3.1 Etape 1

- Création des 2 instances « Windows 10 » dans Openstack et mise en place provisoire de celles-ci sur le réseau **EXT_TPRT**. Suivre la méthodologie vue dans le TP d'introduction pour la création d'instances. Il faudra aussi penser impérativement à finaliser l'installation (supprimer la partition de récupération, agrandir la partition C:, activer Windows, désactiver les mises à jour…) et mettre les machines à l'heure (NTP + fuseau horaire)
- Création de groupe de sécurité et réalisation de la connexion RDP depuis les postes de TP.
- Réalisation du schéma (sur papier) **du réseau final** du cahier des charges, comprenant les instances (avec leur FQDN), les routeurs, les réseaux, les adresses des réseaux : **ce schéma est obligatoire et nécessite la validation d'un enseignant avant la fin de la première séance de projet.**

3.2 Etape 2

- Création d'une paire de clefs SSH par chaque étudiant (nommées **keypair-SSH-agalan1** et **keypair-SSH-agalan2**).
- Création des 2 instances **Linux « CentosStream8 CLI »** et mise en place provisoire de celles-ci sur le réseau **EXT_TPRT.** Suivre la méthodologie vue dans le TP d'introduction pour la création d'instances. On modifiera le fichier **centos-cli-cloudinit.yml** afin de créer pour chaque instance deux utilisateurs **agalan1** et **agalan2** utilisant chacun sa propre clef SSH créée précédemment. Ceci permettra aux deux membres du binôme d'accéder à distance aux deux instances.
- Création d'un groupe de sécurité et réalisation de la connexion SSH vers les deux instances depuis chaque poste de TP.
- Configuration de base de toutes les machines : **FQDN**, date et heure (**NTP**), désactivation éventuelle des services inutiles.
- Création dans **Openstack** des réseaux et des routeurs.
- Mise à jour des adresses des routeurs sur le schéma papier.
- Création du schéma réseau sur (https://www.draw.io) : on créera un nouveau « **Diagramme vierge** » et on utilisera les bibliothèques existantes pour obtenir des symboles normalisés (« **Cisco 19/Routing WAN** » et « **Network** » notamment). On pourra s'inspirer du schéma présent dans la partie 4 : « Quelques éléments de méthodologie »

- Représentation des tables de routage des routeurs **GW0**, **GW1** et **GW2** et de l'instance **w10a-agalan1.proj-gxby** du schéma précédent (à représenter sur la même feuille que le schéma réseau).
- Configuration des routes statiques dans les routeurs conformément aux tables de routage (onglet « routes statiques » de la configuration de chaque routeur). **Attention, pour le routeur NAT GW0 (et seulement pour lui), la route par défaut est déjà configurée automatiquement dans Openstack, il NE FAUT DONC PAS la renseigner manuellement.**

3.3 Etape 3

- Test de connexion SSH depuis chaque instance W10 dans Openstack vers les 2 instances Linux sur le réseau **EXT_TPRT** (Installer Mobaxterm sur les instances W10 pour la connexion SSH).
- Placement des quatre instances dans les réseaux définitifs, accès depuis les machines de TP aux instances Windows par RDP via l'IP flottante puis accès depuis celles-ci aux instances linux par SSH. Attention : vérifier les groupes de sécurité qui sont supprimés lors du changement de réseau d'une instance (n'autoriser que les services nécessaires) et vérifier la configuration du serveur DNS joignable via le réseau **EXT_UGA.** Pour la prise en compte d'un nouveau DNS sur les machines Linux Centos8 dans cet environnement « Openstack », il est nécessaire de modifier manuellement le fichier **/etc/resolv.conf**).
- Le routage doit être optimisé pour les instances (notamment les instances W10) : nombre de sauts minimum vers chaque réseau destination ; ce routage doit être permanent, donc opérationnel au redémarrage de chaque instance.
- Mise à jour des adresses des instances sur le schéma (https://www.draw.io).
- Dépôt du schéma final (avec les tables de routage) dans l'espace « Travaux » de la SAE201 sous le nom : **schema-proj-gxby.drawio. (format .drawio, avec le nom correct)**

Pour les étapes suivantes, la mise en place des services s'effectuera toujours en prenant en compte l'aspect sécurité : Firewall et groupes de sécurité **autorisant seulement les services utilisés.** De plus, les services doivent être configurés pour se lancer automatiquement au démarrage.

3.4 Etape 4 : service DNS

- Mise en place du service DNS avec résolution des noms des 4 machines et redirection vers le serveur DNS de l'UGA pour les autres noms. (Rappel : les requêtes DNS se font sur le port **53 en UDP). Les quatre instances doivent utiliser ce serveur DNS. Cette étape est indispensable et doit fonctionner avant de poursuivre le projet.**

3.5 Etape 5 : service FTP

- Mise en place du service FTP en mode anonyme, accessible depuis les instances Windows avec résolution du nom par le DNS interne. Les fichiers devront se trouver obligatoirement à la racine du site (dans le répertoire **/var/ftp)** et correspondront à une copie de tous les fichiers présents dans le répertoire ftp.rtgrenoble.fr/TpRes/OUTILS.
- Installation de **Filezilla Client** sur les instances « Windows 10 » pour tester facilement le serveur FTP.

3.6 Etape 6 : service Web

Mise en place du service Web : site de test et sites personnels, accessibles depuis les instances Windows avec résolution des noms par le DNS interne.

- Création d'un site web de test pour http://servb-agalan2.proj-gxby avec une simple page d'accueil affichant un message du type « Site Web de test de **proj-gxby** »
- Création de deux sites virtuels personnels **www-agalan1.proj-gxby** et **www-agalan2.proj-gxby** reprenant le contenu du site développé pour le CV Web de chaque membre du binôme lors de la **SAE104.**

Les trois sites Web devront obligatoirement être testés avec **Egde** depuis chaque instance « Windows 10 ».

3.7 Etape 7 : Nextcloud

Installation de NEXTCLOUD, basée sur **Apache** avec comme base de données **mysql**, permettant un partage de fichiers avec gestion des droits utilisateurs (création de deux utilisateurs **user1** et **user2**). L'accès au serveur s'effectuera via le site virtuel **nextcloud.proj-gxby**.

3.8 Etape 8 : Prometheus

Si tous les services précédents fonctionnent, installation progressive de « **Prometheus** », « **Node_exporter** » puis « **Grafana** » sur **serva-agalan1.proj-gxby**. Ceci doit permettre dans une configuration finale d'obtenir un tableau de bord affichant les ressources utilisées sur ce serveur.
Réaliser enfin un second tableau de bord dans Grafana permettant d'afficher les ressources du second serveur linux **servb-agalan2.proj-gxby**. Attention, il ne s'agit pas de réinstaller « **Prometheus** », « **Node_exporter** » et « **Grafana** » sur **servb-agalan2.proj-gxby** mais bien d'afficher les ressources de **servb-agalan2.proj-gxby** sur le tableau de bord de **serva-agalan1.proj-gxby**…
Faire en sorte de mettre en favori votre tableau de bord afin qu'il soit facilement visible dès l'ouverture de la page Web.

4 QUELQUES ELEMENTS DE METHODOLOGIE

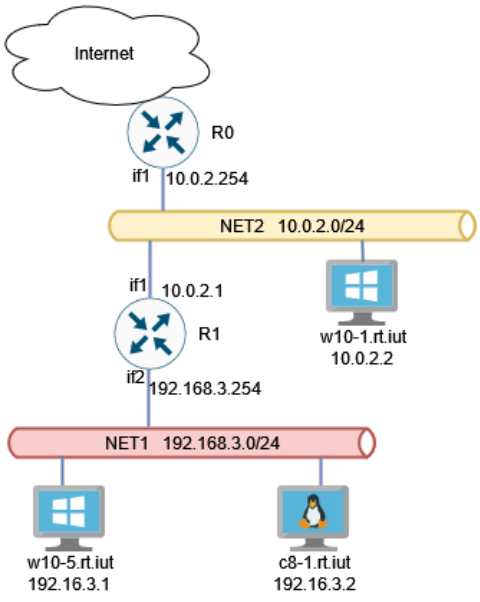
4.1 Schéma du réseau

En réseau, il est impossible de travailler sans un schéma de l'infrastructure, des réseaux, de l'emplacement de ces machines, sans un plan d'adressage fonctionnel.
L'élaboration de ce schéma sera donc pour vous une étape obligatoire, indispensable pour une bonne réalisation. Il vous permettra d'avoir une vision globale des éléments mis en place, et de pouvoir expliquer ce que vous faites.
Il existe plusieurs outils gratuits ou payants pour réaliser ces schémas.
Le plus connu est Visio de Microsoft. Il n'est pas fourni avec la suite Office. C'est un outil de référence.

Il existe des alternatives gratuites, mais leurs possibilités et leurs rendus font qu'ils sont loin d'atteindre les possibilités de Visio.

Il existe une autre solution : les sites en ligne. Parmi ceux-ci, un site est vraiment intéressant car il est gratuit et ne nécessite aucune inscription. C'est celui que nous vous demanderons d'utiliser : https://www.draw.io.

Ci-contre un exemple de réalisation.



R1		
Destination	GW	Interface
10.0.2.0/24	*	if1
192.168.3.0/24	*	if2
0.0.0.0/0	10.0.2.254	if1

4.2 Méthodes de résolution des problèmes

En cas de problème, en réseau, il faut être méthodique

- Est-ce que la configuration réseau est correcte, à savoir IP, masque, passerelle par défaut, DNS ?
- Est-ce qu'on arrive à joindre la passerelle par défaut ?

- Est-ce qu'on peut sortir de son réseau local ? (ping 8.8.8.8). 8.8.8.8 est une machine en Californie, et entre la salle réseau et la Californie, il peut y avoir d'autres problèmes...)
- Est-ce qu'on arrive à joindre une machine par son nom ? (ping www.toto.fr). Si ce n'est pas le cas, a-t-on renseigné un serveur DNS fonctionnel depuis le réseau sur lequel se trouve notre instance ? Est-ce que le serveur DNS renseigné connait les noms qu'il doit résoudre ? Par exemple, si on a des noms « privés », comme serveur.mydom.local, est-ce qu'on utilise bien un DNS connaissant les enregistrements du domaine « mydom.local » ?
- Est-ce qu'un ping fonctionne entre les instances présentes sur les réseaux internes ? Si ce n'est pas le cas, d'où peut provenir le problème ?
    - un problème de routage des paquets dans les réseaux internes ?
    - un blocage lié au firewall de la machine destinataire ?
    - un blocage lié au groupe de sécurité de l'instance dans Openstack ?
- Dans le cas où le ping fonctionne mais que la connexion à un service est impossible, d'où peut provenir le problème ?
    - le service est-il autorisé par le firewall de l'instance destinataire ?
    - le service est-il autorisé par le groupe de sécurité de l'instance destinataire dans Openstack ?
    - le service est-il bien démarré ?
    - le service écoute-t-il sur l'interface réseau adéquate ? (netstat)
    - le service est-il configuré correctement ?

4.3 Wireshark, tcpdump, logs

Il peut être nécessaire de mettre en place des outils pour se rendre compte que tout se passe bien au fur et à mesure de l'avancement, ou pour aider en cas de difficultés. Vous connaissez déjà Wireshark, outil gratuit, fonctionnel sur tous les systèmes d'exploitation à condition d'avoir une interface graphique.
Pour cette SAE, vous devez utiliser des instances Linux ne possédant qu'une console texte ; il faut dans ce cas utiliser l'utilitaire « tcpdump ».
« tcpdump » est très pratique, en de nombreuses circonstances, mais il est souvent nécessaire de filtrer les paquets affichés sinon c'est inexploitable. Comme vous ne connaissez pas cet outil, cette mise en place d'un réseau est une excellente occasion de le découvrir et de vous en servir.
Pour vous aider, en annexe, vous avez une synthèse des commandes dans un document appelé « Cheat Sheet », soit « aide-mémoire ».
A vous d'utiliser cet outil sur les instances qui le nécessitent et de penser à lui plus tard, lors de TPs ou SAÉ à venir.
Enfin, que ce soit sous Linux ou sous Windows, vous avez des « logs », c'est à dire des informations sur l'activité des programmes, que ce soit à propos de dysfonctionnements ou non.
Sous Windows, les logs sont nombreux, organisés d'une certaine façon, avec des numéros d'événements propre à l'événement enregistré. Ces « logs » sont accessibles dans l'Observateur d'événements (commande « eventvwr »).

ANNEXE

Packet Capturing Options

Switch	Syntax	Description
-i any	tcpdump -i any	Capture from all interfaces
-i eth0	tcpdump -i eth0	Capture from specific interface (Ex Eth0)
-c	tcpdump -i eth0 -c 10	Capture first 10 packets and exit
-D	tcpdump -D	Show available interfaces
-A	tcpdump -i eth0 -A	Print in ASCII
-w	tcpdump -i eth0 -w tcpdump.txt	To save capture to a file
-r	tcpdump -r tcpdump.txt	Read and analyze saved capture file
-n	tcpdump -n -I eth0	Do not resolve host names
-nn	tcpdump -n -i eth0	Stop Domain name translation and lookups (Host names or port names)
tcp	tcpdump -i eth0 -c 10 -w tcpdump.pcap tcp	Capture TCP packets only
port	tcpdump -i eth0 port 80	Capture traffic from a defined port only
host	tcpdump host 192.168.1.100	Capture packets from specific host
net	tcpdump net 10.1.1.0/16	Capture files from network subnet
src	tcpdump src 10.1.1.100	Capture from a specific source address
dst	tcpdump dst 10.1.1.100	Capture from a specific destination address
<service>	tcpdump http	Filter traffic based on a port number for a service
<port>	tcpdump port 80	Filter traffic based on a service
port range	tcpdump portrange 21-125	Filter based on port range
-S	tcpdump -S http	Display entire packet
ipv6	tcpdunp -IPV6	Show only IPV6 packets
-d	tcpdump -d tcpdump.pcap	display human readable form in standard output
-F	tcpdump -F tcpdump.pcap	Use the given file as input for filter
-I	tcpdump -I eth0	set interface as monitor mode
-L	tcpdump -L	Display data link types for the interface
-N	tcpdump -N tcpdump.pcap	not printing domian names
-K	tcpdump -K tcpdump.pcap	Do not verify checksum
-p	tcpdump -p -i eth0	Not capturing in promiscuous mode

Logical Operators

Operator	Syntax	Example	Description
AND	and, &&	tcpdump -n src 192.168.1.1 and dst port 21	Combine filtering options
OR	or, ||	tcpdump dst 10.1.1.1 && !icmp	Either of the condition can match
EXCEPT	not, !	tcpdump dst 10.1.1.1 and not icmp	Negation of the condition
LESS	<	tcpdump <32	Shows packets size less than 32
GREATER	>	tcpdump >=32	Shows packets size greater than 32

Installation Commands

CENT OS and REDHAT	$ sudo yum install tcpdump
Fedora	$ dnf install tcpdump
Ubuntu, Debian and Linux Mint	#apt-get install tcpdump

Display / Output Options

Switch	Description
-q	Quite and less verbose mode display less details
-t	Do not print time stamp details in dump
-v	Little verbose output
-vv	More verbose output
-vvv	Most verbose output
-x	Print data and headers in HEX format
-xx	Print data with link headers in HEX format
-X	Print output in HEX and ASCII format **excluding** link headers
-XX	Print output in HEX and ASCII format **including** link headers
-e	Print Link (Ethernet) headers
-S	Print sequence numbers in exact format

Protocols

Ether, fddi, icmp ,ip, ip6 , ppp, radio, rarp, slip, tcp , udp, wlan

Common Commands with Protocols for Filtering Captures

src/ dst host (host name or IP)	Filter by source or destination IP address or host
ether src/ dst host (ethernet host name or IP)	Ethernet host filtering by source or destination
src/ dst net (subnet mask in CIDR)	Filter by subnet
tcp/udp src/dst port (port number)	Filter TCP or UDP packets by source or destination port
tcp/udp src/dst port range (port number range)	Filter TCP or UDP packets by source or destination port range
ether/ip broadcast	Filter for Ethernet or IP broadcasts
ether/ip multicast	Filter for Ethernet or IP multicasts