PAQUELET Etienne Groupe 11

KUT Suha

Jalon 10 : Déploiement VLAN Switches

La simulation étant rendue, nous vous présentons dans ce document les commandes nous ayant permis la mise en place des fonctionnalités suivantes :

* VTP
* Routage inter VLAN
* Configuration par défaut
* STP

Nous vous présenterons aussi des preuves de leur fonctionnement.

La première tâche à réaliser sur les switches a été la mise en place du protocole VTP. Le protocole VTP (VLAN Trunk Protocol) a pour objectif de centraliser la gestion des VLANs dans un réseau et de distribuer les VLANs du serveur VTP aux clients. Le serveur VTP peut être un switch quelconque. Nous avons donc configuré le serveur VTP sur le switch CORESW1, qui est un switch de couche 3. Un switch de couche 3 est un switch qui a des fonctionnalités de routage que les switches de couche 2 ne possèdent pas.

Voici les commandes sur CORESW1 à exécuter afin de créer un serveur VTP version 2 opérationnel.

**CORESW1#conf t** //pour entrer en mode configuration globale

**CORESW1(config)#vtp mode server** // définir le switch comme serveur vtp

**CORESW1(config)#vtp version 2** //spécification de la version de vtp utilisé. Il existe 3 versions de VTP actuellement

**CORESW1(config)#vtp domain wsl2024.org** //definition du domaine vtp

**CORESW1(config)#vtp password P@ssw0rd** //Sécurisation des communications vtp. Un switch client où le mot de passe n’est pas défini ne traitera pas les opérations transmises par le serveur VTP.

Afin que les clients puissent recevoir les VLANs qui seront crées sur le serveur, il faut mettre les interfaces de CORESW1 reliées aux clients VTP en mode trunk. Cela permettra ainsi le transport des trames des différents VLANs sur une seule interface.

**CORESW1(config)#interface Fa0/1**

**CORESW1(config-if)#switchport mode trunk**

**CORESW1(config-if)#switchport trunk native vlan 666** //Le VLAN 666 sera crée ultérieurement.

**CORESW1(config-if)#switchport trunk allowed vlan 10,20,30,40,50,99**

**CORESW1(config-if)#switchport trunk encapsulation dot1q**

**CORESW1(config-if)#switchport nonegotiate**

Les commandes ci-dessus indique que l’encapsulation des trames respectera la norme 802.q, que seul les VLANs 10,20,30,40,50,99 pourront être transportés sur l'interface, et que le VLAN natif sera le VLAN 666. Ainsi, les VLANs crée sur le serveur pourront être transmis aux switches clients. Il faut cependant effectuer cette même opération d’assignation en mode trunk de l’interface du client reliée au serveur pour que le client puisse communiquer avec le serveur.

Voici les commandes pour créer un VLAN :

**CORESW1(config)#vlan 10**

**CORESW1(config-vlan)#name Servers**

Afin que le client vtp puisse recevoir les différents VLANs crée par le serveur, voici les commandes réalisées sur un switch client. L’ordre des commandes est important. En effet, une fois que le client a été configuré en mode client VTP, la version de VTP utilisée par le client n’est plus modifiable. Pour des questions de compatibilité, il est important de modifier la version de VTP sur le client avant que celui-ci soit déclaré en tant que client VTP.

**ACCSW1(config)#vtp version 2**

**ACCSW1(config)#vtp mode client**

**ACCSW1(config)#vtp domain wsl2024.org**

**ACCSW1(config)#vtp password P@ssw0rd**

Il faut que le domaine et le mot de passe soient les mêmes que ceux configurés sur le serveur. Nous pouvons voir dans la capture suivante le résumé de la bonne configuration du serveur VTP.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Vérification de la configuration du client VTP et confirmation de la présence des vlans :

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Pour configurer le routage inter VLAN avec des switches de couche 3, la solution de Router on a Stick ne peut être mise en place. Il faut donc créer des interfaces VLANs et leurs attribuer une adresse IP.

**CORESW1(config)#interface vlan 10**

**CORESW1(config-if)#description IP VLAN 10**

**CORESW1(config-if)#ip address 10.11.10.60 255.255.255.192**

**CORESW1(config-if)#exit**

Une fois cela répété pour chaque VLAN, il faut indiquer au switch de faire du routage et que les interfaces reliées aux autres switches soient bien en mode trunk :

**CORESW1(config)#ip routing**

Nous pouvons vérifier la bonne configuration du routage inter VLAN ainsi :

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

![Une image contenant capture d’écran, noir

Description générée automatiquement]()

![Une image contenant capture d’écran, noir

Description générée automatiquement]()

![Une image contenant capture d’écran, noir

Description générée automatiquement]()

![Une image contenant capture d’écran, noir

Description générée automatiquement]()

![Une image contenant capture d’écran, noir

Description générée automatiquement]()

![Une image contenant capture d’écran, noir

Description générée automatiquement]()

Nous pouvons observer les différentes interfaces trunk avec leurs caractéristiques du switch CORESW2, dont la configuration est identique à celle de CORESW1, afin de permettre le bon fonctionnement du protocole de redondance de passerelle HSRP.

Une image contenant texte, capture d’écran, Police

Description générée automatiquementUne image contenant capture d’écran, noir

Description générée automatiquementUne image contenant texte, capture d’écran, menu, Police

Description générée automatiquementUne image contenant texte, capture d’écran

Description générée automatiquementVoici un résumé de la configuration des interfaces VLANs et interfaces physiques reliées aux switches ACCSW1 et 2.

Nous pouvons y observer la présence d’interfaces pour les VLANs 200,300 et 30. En effet, les switches CORE doivent faire la liaison via l’Ether Channel entre les deux routeurs Edge pour que la liaison BGP soit fonctionnelle entre ces routeurs EDGES. De plus, l’interface VLAN 200 permet aux réseaux locaux d’être routés vers le cœur de réseaux.

* Configuration de l’Ether Channel

Nous avons ensuite configuré l’Ether Channel entre CORESW1 et CORESW2. L’Ether Channel est une technologie permettant la redondance de lien trunk. Il faut au minimum 2 liens trunks entre 2 switches pour que l’Ether Channel soit fonctionnel. Ainsi, si un des liens trunk venait à être défaillant, la communication entre les deux switches sera toujours opérationnelle.

Voici comment configurer l’Ether Channel :

**CORESW1(config)#interface port-channel 1**

**CORESW1(config-if)#description LAG (LACP)**

**CORESW1(config-if)#switchport trunk native vlan 666**

**CORESW1(config-if)#switchport trunk allowed vlan 10,20,30,40,50,99,300,666**

**CORESW1(config-if)#switchport trunk encapsulation dot1q**

**CORESW1(config-if)#switchport mode trunk**

**CORESW1(config-if)#switchport nonegotiate**

**CORESW1(config)interface fa0/3**

**CORESW1(config-if)#channel-group 1 mode active** //on décalre que l’interface fait partie du groupe port-channel 1

La commande **channel-group 1 mode active** permet l’ajout de l’interface fa0/3 au port channel. Ainsi, si l’interface fa0/4 est configuré de même et qu’elle devient défaillante, le lien entre CORESW1 et CORESW2 sera toujours actif.

Nous pouvons vérifier la bonne configuration de l’ether channel ainsi :

Une image contenant texte, capture d’écran, menu, Police

Description générée automatiquement

Nous pouvons constater que les ports Fa1/0/22 sont en mode *active* et utilisent des LACPDU lentes, ces dernières étant des paquets échangés pour négocier et maintenir l'agrégation des liens dans le cadre du protocole LACP.

* Configuration par défaut

Concernant la configuration par défaut des appareils réseaux demandés, voici les commandes à réaliser. Nous avons configuré une adresse IP sur les switches afin de les rendre joignables par SSH.

**Switch(config)#hostname CORESW1**

**CORESW1(config)ip domain-name wsl2024.org**

**CORESW1(config)#no ip domain-lookup** //pas de résolution de domaine

**CORESW1(config)#username admin privileges 15 password P@ssw0rd** //Définition d’un utilisateur pour les connexions SSH.

**CORESW1(config)#crypto key generate rsa** // Définition d’une clé rsa pour la sécurisation de la connexion SSH

**modulus 2048** //Définition de la longueur de la clé.

**CORESW1(config)#ip ssh version 2** //Définition de la version de SSH

**CORESW1(config)#access-list 99 permit 10.11.99.0 0.0.0.31** //Autorisation des IP en 10.11.99.0/27 de se connecter en ssh sur le switch

**CORESW1(config)#line vty 0 4** //Configuration de la ligne vty 0 4 pour les connexions ssh

**CORESW1(config-line)#access-class 99 in** //Application de l’ACL 99

**CORESW1(config-line)#transport input ssh** // Définition du traffic qui sera accepté sur cette ligne

**CORESW1(config-line)#login local** //Authentification locale

**CORESW1(config-line)#exec-timeout 5 0** //Time out après 5 min d’inactivité

**CORESW1(config-line)#absolute-timeout 20** //Timeout absolue de 20 minutes

**CORESW1(config)#banner login #** // bannière de login

**/!\ Restricted access. Only for authorized people /!\**

**#**

**CORESW1(config)#service password-encryption** // Chiffrement des mots de passes

* Configuration du STP

Le STP, Spanning Tree Protocole, est un protocole dont le but est d’éviter les boucles de commutation dans un réseau local (LAN). STP garantit qu'il n'y a qu'un seul chemin actif entre deux équipements du réseau à un instant donné, tout en maintenant des chemins redondants en mode passif pour assurer la tolérance aux pannes.

Afin de configurer le STP sur les switches, voici la configuration réalisée sur CORESW1

**CORESW1(config)#spanning-tree mode rapid-pvst**

**CORESW1(config)#spanning-tree extend system-id**

**CORESW1(config)#spanning-tree vlan 10,20,30,40,50,99,300 priority 24576** //définition de la priorité sur CORESW1

Configuration des switches clients

**ACCSW1(config)#interface Gi1/0/2**

**ACCSW1(config-if)#description LIAISON ACCSW1-CORESW2**

**ACCSW1(config-if)# spanning-tree cost 100**

**ACCSW1(config-if)#spanning-tree portfast**

**ACCSW1(config-if)#spanning-tree bpduguard enable**

Configuration des sécurités :

**ACCSW1(config-if) switchport port-security maximum 3**

**ACCSW1(config-if)#switchport port-security aging time 30**

**ACCSW1(config-if)#switchport port-security**

Voici quelques screenshots montrant un résumé des configurations effectués pour le STP :

Sur le switch ACCSW1

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Sur le CORESW1 :

Une image contenant capture d’écran, noir

Description générée automatiquementUne image contenant capture d’écran, noir

Description générée automatiquement![Une image contenant capture d’écran, noir

Description générée automatiquement]()Une image contenant capture d’écran, noir

Description générée automatiquementUne image contenant texte, capture d’écran

Description générée automatiquementUne image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Nous avons également réalisé un script permettant d’abaisser la priorité HSRP des interfaces afin que le CORESW2 devient la passerelle par défaut si EDGE1 était down. Ce script peut être réalisé sur le routeur grâce à l’Embedded Event Manager (EEM). Cette fonctionnalité des produits cisco permet d’automatiser certaines tâches en fonction de certains paramètres. Ainsi, dans notre script, que nous n’avons pu tester suite à une erreur mais qui globalement est correcte et suit la logique suivante « Si le CORESW1 arrive pas à ping EDGE1 🡺 baisser priorité HSRP des interfaces vlan », nous avons automatisé grâce à l’EEM l’envoi d’un ping et l’abaissement des priorités HSRP du CORESW1 en fonction du résultat du ping envoyé. Voici le script réalisé :

event manager applet HSRP-Priority-Monitor

event timer watchdog time 10

action 1.0 cli command "enable"

action 2.0 cli command "ping 10.11.254.242 source Vlan100 repeat 2"

action 3.0 regexp "Success rate is 0 percent" "$\_cli\_result" match result

action 4.0 if $match eq "1"

action 4.1 cli command "config terminal"

action 4.2 cli command "interface Vlan10"

action 4.3 cli command "standby 1 priority 90"

action 4.4 cli command "interface Vlan20"

action 4.5 cli command "standby 1 priority 90"

action 4.6 cli command "interface Vlan40"

action 4.7 cli command "standby 1 priority 90"

action 4.8 cli command "interface Vlan50"

action 4.9 cli command "standby 1 priority 90"

action 5.0 cli command "interface Vlan99"

action 5.1 cli command "standby 1 priority 90"

action 5.2 cli command "end"

action 5.3 syslog msg "Interface injoignable, priorité HSRP abaissée."

action 5.4 else

action 5.5 cli command "config terminal"

action 5.6 cli command "interface Vlan10"

action 5.7 cli command "standby 1 priority 110"

action 5.8 cli command "interface Vlan20"

action 5.9 cli command "standby 1 priority 110"

action 6.0 cli command "interface Vlan40"

action 6.1 cli command "standby 1 priority 110"

action 6.2 cli command "interface Vlan50"

action 6.3 cli command "standby 1 priority 110"

action 6.4 cli command "interface Vlan99"

action 6.5 cli command "standby 1 priority 110"

action 6.6 cli command "end"

action 6.7 syslog msg "Interface joignable, priorité HSRP restaurée."

action 6.8 end

could not find environnement variable