Synchronisation des données entre Netdisco et Itop

Présentation Netdisco

Netdisco est un outil de gestion de réseau basé sur le Web qui convient aux réseaux de petite à très grande taille. Les données sont collectées dans une base de données PostgreSQL à l'aide de SNMP. Netdisco est écrit en Perl et est indépendant de la base de données PostgreSQL. Il fonctionne bien sur n'importe quel système Linux ou Unix. Il comprend un serveur Web léger pour l'interface, un démon backend pour collecter les données sur votre réseau et une interface de ligne de commande pour le dépannage. Le fichier de configuration est au format YAML.

Netdisco utilise le protocole CDP (Cisco Discovery Protocol) et LLDP (Link Layer Discovery Protocol) pour découvrir des voisins.

Le Cisco Discovery Protocol (CDP) est un protocole de découverte de réseau de niveau 2 développé par Cisco Systems qui permet, avec SNMP, de trouver d'autres périphériques voisins directement connectés (routeur, switch, pont, etc.).

Le Link Layer Discovery Protocol (LLDP) est un protocole standardisé (IEEE 802.1AB) de découverte de réseau ayant pour vocation de suppléer la multitude de protocoles propriétaires du même type (Cisco Discovery Protocol, Nortel Discovery Protocol, ...) et ainsi de permettre aux équipements de différents fabricants de se découvrir mutuellement.

Les fonctionnalités de Netdisco:

- Localiser une machine sur le réseau par MAC ou IP et retrouver le port du switch dans lequel elle est connectée
- Désactiver un port de commutateur ou modifier l'affectation d'un VLAN ou PoE d'un port
- Inventaire de votre matériel réseau par modèle, fournisseur, logiciel et système d'exploitation
- une image complète de votre réseau

URL: <http://netdisco-2:5000>

Installation Netdisco

Le logiciel Netdisco est installé sur une vm debian (netdisco-2.in.ac-versailles.fr/172.31.131.223).

Dépendances

Netdisco a plusieurs dépendances de bibliothèque Perl qui seront automatiquement installées.

Cependant, il est fortement recommandé d'installer "DBD :: Pg", "SNMP" et un compilateur en utilisant vos paquets de système d'exploitation.

Créez un utilisateur sur votre système appelé netdisco. Nous allons installer Netdisco et ses dépendances dans la zone d'accueil de cet utilisateur, ce qui prendra environ 250 Mo, y compris les fichiers MIB.

```
root:~# useradd -m -p x -s /bin/bash netdisco
```

On Ubuntu/Debian:

```
root:~# apt-get install libdbd-pg-perl libsnmp-perl libssl-dev build-essential
root:~# apt-get install cpanm
root:~# cpanm DBD::Pg
```

Installation snmp

SNMP est un protocole réseau qui permet le monitoring et la supervision d'éléments systèmes et réseau. Généralement, un serveur de supervision utilise SNMP pour connaître rapidement l'état du parc informatique (switch, routeur, serveurs) comme l'occupation RAM, CPU, Disque, etc.

```
root:~# apt-get install snmpd snmp
```

Nous pourrions ensuite voir si le service est actif:

```
root:~# systemctl status snmpd
```

Si ce n'est pas le cas, nous pourrions le démarrer:

```
root:~# systemctl start snmpd
```

Enfin, une fois celui-ci démarré, nous verrons qu'il est actif sur le port UDP 161, cela avec la commande "ss" qui permet de lister les ports en écoute :

```
root:~# ss -ulnp
```

Pour détailler cette commande :

- l'option "u" permet de lister les ports UDP uniquement
- l'option "l" permet de lister les ports en écoute ("Listening")
- l'option "n" permet d'afficher les numéros de ports et non leur correspondance (exemple : afficher "22" plutôt que "SSH")
- l'option "p" permet d'afficher les processus correspondants aux ports en écoute

Dans tous les cas, nous devrions avoir quelque chose qui ressemble à cela :

```
UNCONN      0      0      *:161      *:*
users: ( ("snmpd", pid=406, fd=9) )
```

Pour ouvrir l'écoute sur l'extérieur, il faut aller modifier le fichier de configuration de snmp: **/etc/snmp/snmpd.conf**. On va lors voir dès les premières lignes, celles qui nous intéressent:

```
# Listen for connections from the local system only
agentAddress  udp:127.0.0.1:161

# Listen for connections on all interfaces (both IPv4 *and* IPv6)
#agentAddress udp:161,udp6:[::1]:161
```

Nous allons commenter la ligne suivante :

```
agentAddress  udp:127.0.0.1:161
```

Puis ajouter celle-ci:

```
agentAddress  udp:161
```

Dans le fichier **/etc/snmp/snmpd.conf** à la ligne 41, nous allons changer la **"community"** . La community est une sorte de "mot de passe" qui va permettre de restreindre l'accès aux informations fournis par les serveurs SNMP. Par défaut, cette community est généralement "public" , c'est pourquoi il est recommandé de changer, autrement, n'importe qui dans votre réseau pourra alors questionner vos serveurs sur leurs états de santé :

```
rocommunity ctodnb default -V systemonly
```

Une fois tout cela fait, on pourra redémarrer SNMP pour recharger la configuration modifiée

```
systemctl restart snmpd
```

PostgreSQL

PostgreSQL est un puissant système de gestion de base de données. Il est réputé pour sa fiabilité, son intégrité des données, et son exactitude. Il fonctionne sur tous les principaux systèmes d'exploitation, y compris Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS, Solaris, Tru64) et Windows. Netdisco utilise le serveur de base de données PostgreSQL. Installez PostgreSQL (au moins la version 8.4), puis passez au superutilisateur PostgreSQL (généralement postgres). On va aussi installer le paquet phppgadmin qui permet d'administrer le serveur PostgreSQL depuis une interface graphique.

URL: <http://netdisco-2/phppgadmin/>

Installation et configuration de PostgreSQL

Exécutez la commande suivante :

```
apt-get install postgresql postgresql-client postgresql-contrib
apt-get install phppgadmin
```

Dans le fichier /etc/phppgadmin/config.inc.php,

```
remplacer le 'localhost' en 127.0.0.1
```

A l'installation de PostgreSQL, un utilisateur à la fois du système et de PostgreSQL, nommé postgres, est créé (sans mot de passe) avec sa propre base de données portant son nom. Cet utilisateur est également administrateur de PostgreSQL ;

Vous pouvez déjà accéder à PostgreSQL en étant connecté en tant que postgres via les commandes suivantes :

```
su - postgres
psql
```

Créez une nouvelle base de donnée et un utilisateur PostgreSQL pour l'application Netdisco.

```
root:~# su - postgres
postgres:~$ createuser -DRSP netdisco
Enter password for new role:
Enter it again:
postgres:~$ createdb -O netdisco netdisco
```

Pour attribuer les droits d'administrateur au utilisateur Netdisco, tapez les commandes suivantes:

```
postgres@netdisco-2:~$ psql
psql (9.4.15)
Saisissez « help » pour l'aide.
```

```

postgres=# SELECT usesuper FROM pg_user;
-----
t
f
(2 lignes)

postgres=# SELECT usesuper FROM pg_user WHERE username='netdisco';
usesuper
-----
f
(1 ligne)

postgres=# ALTER ROLE netdisco WITH superuser;
ALTER ROLE
postgres=# SELECT usesuper FROM pg_user WHERE username='netdisco';
usesuper
-----
t

```

Installation et configuration de Netdisco

Pour éviter toute confusion avec votre système, utilisez le script suivant pour télécharger et installer Netdisco et ses dépendances dans la zone d'accueil de l'utilisateur netdisco (~ netdisco / perl5):

```

su - netdisco
curl -L https://cpanmin.us/ | perl - --notest --local-lib ~/perl5 App::Netdisco

```

Associez les applications nouvellement installées à un emplacement pratique:

```

mkdir ~/bin
ln -s ~/perl5/bin/{localenv,netdisco-*} ~/bin/

```

Testez l'installation en exécutant la commande suivante, qui ne devrait produire qu'un message d'état (c'est juste un test – on va démarrer les démons plus tard):

```

~/bin/netdisco-backend status

```

Créez un répertoire pour votre configuration locale et copiez le modèle de configuration à partir de cette distribution:

```

mkdir ~/environments
cp ~/perl5/lib/perl5/auto/share/dist/App-Netdisco/environments/deployment.yml
~/environments
chmod 600 ~/environments/deployment.yml

```

Modifiez le fichier ("~/environments/deployment.yml") et modifiez les paramètres de connexion de

la base de données pour qu'ils correspondent à ceux de votre système local. Dans le même fichier, décommentez et modifiez le paramètre `domain_suffix` pour qu'il soit adapté à votre site local. Modifiez aussi le paramètre de chaîne de communauté.

```
# NETDISCO 2 CONFIGURATION FILE
#
# Settings in this file override share/config.yml
# See App::Netdisco::Manual::Configuration for more info.
# -----
# ESSENTIAL SETTINGS
# -----

database:
  name: 'netdisco'
  user: 'netdisco'
  pass: 'EPPve254'
  #host: 'localhost'

# -----
# RECOMMENDED SETTINGS
# -----

# SNMP community string(s)
# ~~~~~

device_auth:
  - tag: 'default_v2_readonly'
    community: 'ctodnb'
    read: true
    write: false
  # - tag: 'default_v2_for_write'
  #   community: 'private'
  #   read: false
  #   write: true

# will be stripped from fqdn when displayed in the web UI
# also, do not forget the leading dot.
# ~~~~~
domain_suffix: 'in.ac-versailles.fr'
```

La base de données doit être configurée si elle est nouvelle ou mettre à jour à partir de la version actuelle de Netdisco (1.x). Vous avez également besoin des préfixes d'adresse MAC du fournisseur (données OUI) et de certaines MIB si vous voulez exécuter le démon backend. Le script suivant prendra soin de tout cela pour vous:

```
~/bin/netdisco-deploy
```

Si c'est une nouvelle installation, répondez 'yes' pour toutes les questions.

Exécutez la commande suivante pour démarrer le serveur d'application Web en tant que démon en

arrière-plan (en écoute sur le port 5000):

```
~/bin/netdisco-web start
```

Exécutez la commande suivante pour démarrer le démon de contrôle du travail (contrôle de port, etc.):

```
~/bin/netdisco-backend start
```

Les sous-réseaux et les équipements réseaux qu'on va découvrir sont les suivants:

```
192.168.231.0 ----> out_band_smb
192.168.233.0 via 192.168.233.245 ----> in_band_sma, passerelle de 4510
192.168.237.0 ----> hardware
192.168.235.0 ----> in_bound_smb
192.168.231.204 ----> load_balancer(répartiteur de charge),f5 / BIG-IP 2000
192.168.25.108 ----> parefeu interne, fortinet / 800C
```

Commande pour lancer la découverte sur les sous-réseaux et les équipements réseaux, cités ci-dessous:

```
netdisco@netdisco-2:~$ ~/bin/netdisco-do -D discover -d 192.168.231.0
netdisco@netdisco-2:~$ ~/bin/netdisco-do -D discover -d 192.168.233.245
netdisco@netdisco-2:~$ ~/bin/netdisco-do -D discover -d 192.168.237.0
netdisco@netdisco-2:~$ ~/bin/netdisco-do -D discover -d 192.168.235.0
netdisco@netdisco-2:~$ ~/bin/netdisco-do -D discover -d 192.168.231.204
netdisco@netdisco-2:~$ ~/bin/netdisco-do -D discover -d 192.168.25.108
```

Fonctionnement du script

Comme vu précédemment, les données de Netdisco sont stockées dans une base de données PostgreSQL.

La synchronisation des données entre Netdisco et Itop se fait en plusieurs scripts.

Le script **switch_add_rm_number.py** permet d'automatiser la création, la suppression et la synchronisation du numéro de série des équipements réseaux dans Itop.

Le script **switch_ip_gestion.py** permet de synchroniser l'adresse ip de gestion des équipements réseaux de façon dynamique avec les données récupérées depuis Netdisco.

Le script **switch_client.py** permet de retrouver toutes les machines clients qui sont connectées dans des équipements réseaux avec l'adresse MAC et l'adresse ip du machine client.

Dans la base de données d'Itop, les tables concernées par la synchronisation des équipements réseaux sont les suivantes:

- * connectableci (id)
- * datacenterdevice (id,rack_id,...,managementip_id,redundancy)
- * networkdevice (id,networkdevicetype_id,iosversion_id,ram,position)
- * physicaldevice (id,serialnumber,location_id,...)
- * ipaddressv4 (id,subnet_id,range_id,ip)
- * ipaddress (id,short_name,domain_id,fqdn)
- * ipobject (id,org_id, ... , finalclass)
- * ipsubnet (id,name, ...)
- * ipsubnetv4 (id,block_id,ip,mask,gatewayip,broadcastip)

Fonctionnement du script 'switch_add_rm_number.py'

Dans un premier temps, on va faire une requête psql pour afin récupérer les données (nom,ip,serial,mac des équipements réseaux) de Netdisco sous forme d'un fichier csv, que l'on nomme *switch_netdisco.csv*. Pour la suite de ce tuto, le fichier **switch_netdisco.csv** va être la source d'alimentation des données pour Itop. Ces données sont exportées depuis la table **device** de la bdd de Netdisco.

Pour éviter des conflits des données, on supprime tous les équipements réseaux présents dans Itop (dans les tables networkDevice,connectableci,datacenterdevice,physicaldevice) puisque la source d'alimentation des équipements réseaux est Netdisco. Une fois qu'on a remis à zéro les données concernant les équipements réseaux, on va importer les données du fichier **switch_netdisco.csv** dans la base d'Itop, en respectant la structure et la syntaxe des tables.

Pour conclure, ce script permet d'automatiser la création, la suppression et la synchronisation du numéro de série des équipements réseaux dans Itop, avec les données de Netdisco.

Fonctionnement du script 'switch_ip_gestion.py'

Comme vu précédemment, le fichier csv **switch_netdisco.csv** contient toutes les adresses ip de gestion des équipements réseaux. L'idée, c'est de récupérer toutes les adresses ip de gestion est de vérifier dans la base de données d'Itop, s'il existe une entrée pour cette adresse ip. Si il existe des entrées pour ces adresses ip alors on les suppriment pour afin importer les données récupérer depuis le Netdisco.

Cette tâche se fait en plusieurs étapes puisque les données concernant des adresses ip sont stockées dans plusieurs tables qui sont dépendent de l'un et l'autre.

Step 1

L'ajout des adresses ip se fait dans la table ipadressv4 dans laquel on doit préciser l'id du sous-réseau correspondant à cette adresse ip. Donc, on va commence par récupérer les données de la table *ipsubnetv4* que l'on va stocké dans le fichier **subnet.csv**. Le fichier **subnet.csv** contient les informations suivantes: id, name, block_id, ip, broadcastip

Le concept pour récupérer l'id du sous-réseaux correspondant à des adresses ip de mgmt des équipements réseaux, c'est que pour chaque adresse ip de gestion, on vérifie si cette adresse ip est compris entre l'adresse sous-réseaux et le broadcast, si oui, alors on récupère l'id du sous-réseaux pour afin ajouter l'adresse ip dans la table *ipadressv4*.

Note: On ne va pas s'intéresser sur le gateway pour récupérer l'id du sous-réseaux puisque certains sous-réseaux ne possèdent pas de gateway. Par exemple les sous-réseaux d'interconnexion.

Step 2

Une fois qu'on a ajouté les adresses ip de gestion dans la table *ipadressv4*, dans la table *ipadress*, on doit ajouter le nom court et le fqdn des adresses ip (qu'on vient d'ajouter dans la table *ipadressv4*). Pour ceci, on commence par récupérer l'id des adresses ip qui viennent d'être insérer dans la table *ipadressv4*.

Dans notre infrastructure, le nom de domaine des adresses ip de management est: *mgt.in.ac-versailles.fr*. Donc, pour créer le fqdn, on va faire une concaténation entre le nom de l'équipement réseau (récupérer depuis le fichier **switch_netdisco.csv**) et le nom de domaine. Puis, il nous reste que de saisir ces informations (id,nom_court,fqdn) dans la table *ipadress*.

Step 3

Puis en se basant sur les données du fichier **switch-netdisco.csv** et le fqdn, on va ajouter l'adresse ip de gestion au serveur approprié. Si le nom court d'une adresse ip est identique au nom du serveur alors il va ajouter l'adresse ip de management à cet serveur. Cette étape se fait dans la table *datacenterdevice*.

Fonctionnement du script 'switch_client.py'

Les informations concernant les machines clients connectées se trouvent dans les tables suivantes:

- * node
- * node_ip

La liaison entre ces deux tables se fait avec l'adresse Mac.

Step 1

Premièrement, on va établir une connexion psql pour exécuter la requête suivante:

```
SELECT n.mac, n.switch, n.port, n.vlan, ni.mac, ni.ip, ni.dns FROM node as n, node_ip as ni WHERE n.mac= ni.mac AND ni.dns is not NULL
```

Le champ *switch* représente l'adresse ip de l'équipement réseau.

Le champ *port* affiche le port dans lequel la machine client est connectée.

Le champ *ni.mac* et *ni.ip* représente l'adresse Mac et l'adresse ip du machine client.

Ces informations sont stockées dans le fichier **client_connecte.csv**.

```
ex: 00:50:56:a0:23:61 | 192.168.237.177 | Gi2/0/9 | 36 | 00:50:56:a0:23:61 |  
172.31.33.143 | db2-13.in.ac-versailles.fr
```

Step 2

Du côté Itop, on fait une requête mysql pour récupérer l'*id*, *managementip_id*, *nom*, *adresse ip de gestion* des équipements réseaux que l'on va stockée dans le fichier **switch_ip.csv**.

```
requête: SELECT datacenterdevice.id, datacenterdevice.managementip_id,  
functionalci.name, ipaddressv4.ip FROM datacenterdevice , ipaddressv4, functionalci  
WHERE datacenterdevice.id=functionalci.id AND  
datacenterdevice.managementip_id=ipaddressv4.id AND  
functionalci.finalclass='NetworkDevice' > switch_ip.csv
```

```
ex: 9208 | 9583 | blade-05-sw-02 | 192.168.237.177
```

Dans l'exemple du 'Step 1', on voit bien que dans le champ *switch* il nous affiche l'adresse ip suivante: 192.168.237.177 qui correspond au *blade-05-sw-02*.

La liaison entre l'équipement réseau et les machines clients se fait avec l'adresse ip de gestion.

Step 3

Puis au final, dans la table **Inkconnectablecitonetnetworkdevice**, on ajoute les données en respectant la structure et la syntaxe des tables.

Liens utiles:

- <https://metacpan.org/pod/App::Netdisco>
- <https://metacpan.org/pod/DBIx::Class>
- <https://metacpan.org/pod/distribution/App-Netdisco/bin/netdisco-do>
- http://search.cpan.org/~oliver/App-Netdisco-2.033004/lib/App/Netdisco/Manual/Configuration.pod#Web_Frontend
- <http://search.cpan.org/~oliver/App-Netdisco-2.033006/lib/App/Netdisco/Manual/Troubleshooting.pod>