

## Outil de cartographie du datacenter

### Sommaire:

- 1) *DCIM, c'est quoi?*
- 2) *Nécessité d'un outil de cartographie*
- 3) *Présentation des 3 logiciels de cartographie*
  - a) *Ralph*
  - b) *Netbox*
  - c) *Racktables*
- 4) *Installation Ralph*
- 5) *Installation Netbox*
- 6) *Comparaison des logiciels et choisir le meilleur*
- 7) *Mise en production du Racktables*
  - a) *Création d'un vm sous esxi (debian 8.0)*
  - b) *Installation de racktables*
  - c) *Création du certificat SSL*
- 8) *Présentation d'itop*
- 9) *Synchronisation des serveurs entre Itop et Racktables*
  - a) *Conception de synchronisation*
  - b) *Script de synchronisation*
- 10) *Synchronisation des vm entre Itop et Racktables*
  - a) *Conception de synchronisation*
  - b) *Script de synchronisation pour les Vm*
- 11) *Connexion LDAP*
- 12) *OpenManage Essentials API*
- 13) *Ajout de la machine Racktables (machine client) dans la supervision*
- 14) *Guide d'utilisation du Racktables*

## 1) DCIM, c'est quoi?

Le DCIM (datacenter infrastructure management) est une solution de gestion de l'infrastructure d'un centre de données. Il offre une vision en temps réel du datacenter et des salles IT, grâce à laquelle l'exploitant peut s'assurer du bon fonctionnement des équipements, anticiper les défaillances, optimiser les infrastructures et ainsi améliorer en permanence son efficacité énergétique.

## 2) Nécessité d'un outil de cartographie

Aujourd'hui, dans les deux salles de machines, le niveau de service attendu a fortement augmenté. En outre, le nombre de contraintes physiques dans le datacenter n'a cessé de croître. Le DCIM, c'est comme une console centralisée, afin d'obtenir une vision unique, simple et complète de l'état de santé de nos datacenter. Et en tant que administrateur

Dans le marché, il existe plusieurs solutions logicielles comme Ralph, Netbox, Racktables etc ... Il est même possible de lier cet outil à d'autres logiciels et donc d'intégrer le DCIM dans la cartographie applicative de la DSI.

La documentation est l'élément le plus important dans la gestion d'un système d'information.

C'est grâce à une bonne documentation que l'on va pouvoir:

- assurer la continuité des services que l'on délivre aux utilisateurs finaux
- assurer un support réactif et de qualité
- gérer le patrimoine informatique du DSI

## 3) Présentation des 3 logiciels de cartographie

Ralph, Netbos et Racktables sont des 3 meilleurs logiciels de cartographie dans le marché. Chacun a sa propre avantage et inconvénient donc on va analyser les 3 logiciels pour afin choisir le plus adéquat à nos infrastructures des salles de machines.

### a) Ralph

Ralph est une solution de gestion d'actifs orientée Datacenter, on peut le comparer à une CMDB (Configuration Management DataBase).

Il permet de:

- Scanner le réseau automatiquement de manière périodique ou manuelle
- Déployer des serveurs en générant les configurations DNS/DHCP et en utilisant PXE
- Voir les relations entre les CI (configuration Items) en utilisant l'interface de visualisation de la CMDB
- 

Ce sous-module permet une gestion avancées des actifs qui a les caractéristiques suivantes :

- Inventaire manuel
- Peut être utilisé conjointement avec la découverte Ralph de base en utilisant une technique de rapprochement
- Couvre le cycle de vie complet des actifs de l'achat jusqu'au retrait.
- Permet de générer des rapport PDF personnalisés
- Gestion des licences intégrés
- Support matériel de base et gestion des contrats
- Module facile et utilisable pour des tâches d'inventaire génériques

Ralph	
Avantages	Inconvénient
<ul style="list-style-type: none"> <li>-Gérer plusieurs data center</li> <li>-Distingue les ressources virtuels et physiques (Cluster, hyperviseur...)</li> <li>-une interface facile et rapide</li> <li>-Gestion d'incident (ITIL)</li> <li>-Informations sur les composants d'un élément (RAM, processeur, disque...)</li> <li>-Assigner plusieurs @ ip à un serveur</li> <li>-Scanne le réseau dynamiquement à travers le DNS (pro's or con's)</li> <li>-Service Cloud</li> <li>-Déploiement de fichier de démarrage</li> <li>-Informations financières sur des éléments (date d'achat, dépréciation, budget, coût, n° de facture...)</li> <li>-Gérer les @ réseau en fonction des baies (avoir le même @ réseau dans différents baie et différents sites)</li> <li>-Gérer les licences des logiciels et les contrats (inventaire des logiciels, coût, date,)</li> <li>-Hiérarchisation des @ ip</li> <li>-Un rapport des éléments du data center que l'on peut télécharger (avec leur ID, nom_fournisseur, modèle, status, ...)</li> <li>-Création des Vlans</li> <li>-Gérer le patrimoine du datacenter</li> </ul>	<ul style="list-style-type: none"> <li>-Scanne le réseau dynamiquement à travers le DNS (pro's or con's)</li> <li>-Absence d'inventaire des câbles</li> <li>-Absence de stockage pour les données sensibles</li> <li>-ne propose pas les différents types des éléments (switch, routeur dc obligé de créer le template pour chaque type d'élément avant d'avoir créer les éléments)</li> <li>-on doit préciser dans le template pour qu'il prend en compte. (ex : si un serveur il a comme hauteur de rack 2U alors on ne peut pas assigner directement ,passer par le template)</li> <li>-manque d'information sur les Vlan</li> <li>-on peut attribuer une @ ip à un élément même si son @ réseau n'existe pas</li> <li>-manque d'information sur les switches</li> </ul>

#### b) Netbox

NetBox est une application Web open source conçue pour s'aider à gérer et documenter les équipements informatiques. NetBox a été développé spécifiquement pour aider les administrateur système et réseau à la gestion des assets du datacenter et des adresses ip.

Les différents fonctions de netbox:

- IPAM : adresse réseau, @ ip, VRFs and VLAN
- Racks : Organiser par groupe et site
- Equipement : types des équipements et leur emplacement dans le baie
- Connections : Réseau, console et connexion d'alimentation entre les équipements
- Fichier : Stocker des fichiers sensibles propres à des équipements (fichier crypter)

Netbox	
Avantages	Inconvénient
<ul style="list-style-type: none"> <li>-On peut gérer plusieurs sites et salles de machines.</li> <li>-une interface facile et rapide</li> <li>-on peut assigner plusieurs @ ip à une interface</li> </ul>	<ul style="list-style-type: none"> <li>- Absence de tickets d'incidents</li> <li>- ne distingue pas des ressources virtuels et physiques (ip SLB, ...)</li> </ul>

<ul style="list-style-type: none"> <li>-possibilité d'exporter et d'importer les données</li> <li>-Ne Découvert pas dynamiquement le réseau car seul humain peut définir correctement le réseau.</li> <li>-Stocker des données sensibles, les données sont chiffrés avec RSA</li> <li>-une interface admin pour gérer l'ensemble de fonctionnalité (facile et rapide)</li> <li>-attribuer un objet à un service (création de différent pôle)</li> <li>-Hiérarchisation des @ ip</li> <li>-Informations sur le FAI</li> <li>-VRF (virutal routing and forwading tables)</li> <li>-Informations financières des éléments (date d'achat,dépréciation,facture ...)</li> <li>-une visualisation par baie</li> </ul>	<ul style="list-style-type: none"> <li>- Absence d'inventaire des câbles</li> <li>- Api en mode lecture seulement</li> <li>- Pas de la supervision du réseau</li> <li>- Pas de service DNS (live ptr)</li> <li>- Pas de connexion ldap</li> </ul>
--	---

### c) Racktables

Racktables est une DCIM qui permet de gérer les patrimoines du datacenter et les adresses ip. C'est une application Web open source conçue pour s'aider aux administrateurs systèmes et réseaux pour la gestion des patrimoines du datacenter.

Racktables	
Avantages	Inconvénient
<ul style="list-style-type: none"> <li>-Assigner plusieurs @ ip à une interface</li> <li>-Exporter et d'importer les données via la bdd</li> <li>-Ne découvert pas dynamiquement le réseau car seul humain peut définir correctement le réseau.</li> <li>-Hiérarchisation des @ ip</li> <li>-attribuer une adresse sous-réseau à un VLAN (visualiser les serveurs d'un VLAN)</li> <li>-Création des VLANS (informations sur les éléments du Vlan)</li> <li>-distingue les ressources virtuels et physiques (hyperviseur,cluster,Ip SLB...)</li> <li>-Inventaire des câbles</li> <li>-une visualisation par baie (interface facile et rapide)</li> <li>-on peut gérer plusieurs salles de machines.</li> <li>-Propose les différents types des éléments (switch,routeur, kVM ...)</li> <li>-Connexion Ldap</li> <li>-Live ptr (remplie la plage d'@ réseau avec les infos du DNS)</li> <li>-informations sur les redirections des ports</li> <li>-Rapport du datacenter (système,IPV4,interfaces,Vlan)</li> <li>-gérer les ports des équipements et la liaison entre eux</li> </ul>	<ul style="list-style-type: none"> <li>-ne fournit pas d'API</li> <li>-Absence de tickets d'incidents</li> <li>-un système pour stocker des fichiers mais non-sécurisé.</li> </ul>

#### 4) Installation Ralph

### **I- Ajouter le dépôt de ralph**

Ralph est seulement compatible avec Ubuntu 14.04.

// Tout d'abord installer le paquet transport-https avec la commande suivante :

```
root@ralph:# apt-get install apt-transport-https
```

// Une fois installer, ajouter le dépôt officiel de ralph :

```
root@ralph : apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys  
379CE192D401AB61  
root@ralph :sh -c "echo 'deb https://dl.bintray.com/vi4m/ralph wheezy main' >  
/etc/apt/sources.list.d/vi4m_ralph.list"
```

### **II- Installation du serveur ralph depuis le dépôt**

//Puis installer le paquet ralph, redis-server et le serveur mysql

```
sudo apt-get update  
sudo apt-get install ralph-core redis-server mysql-server
```

// Attention, c'est possible de rencontrer des problèmes lors la procédure d'installation...

// Dans mon cas, j'ai eu le problème suivant :

```
Changement de support : veuillez insérer le disque  
  
« Debian GNU/Linux 8.7.1 _Jessie_ - Official amd64 CD Binary-1 20170116-11:01 »  
  
dans le lecteur « /media/cdrom/ » et appuyez sur la touche Entrée
```

*Pour résoudre ce problème :*

sudo nano /etc/apt/sources.list et mettre en commentaire le deuxième ligne.

```
admin01srv@SRV-Ralph: ~
#
# deb cdrom:[Debian GNU/Linux 8.7.1 _Jessie_ - Official amd64 CD Binary-1 201701
16-11:01]/ jessie main
#deb cdrom:[Debian GNU/Linux 8.7.1 _Jessie_ - Official amd64 CD Binary-1 2017011
6-11:01]/ jessie main
deb http://ftp.fr.debian.org/debian/ jessie main
deb-src http://ftp.fr.debian.org/debian/ jessie main
deb http://security.debian.org/ jessie/updates main
deb-src http://security.debian.org/ jessie/updates main
# jessie-updates, previously known as 'volatile'
deb http://ftp.fr.debian.org/debian/ jessie-updates main
deb-src http://ftp.fr.debian.org/debian/ jessie-updates main
~
~
~
~
~
"/etc/apt/sources.list" 15 lines, 590 characters
```

//Puis ajoutez a chaque ligne « contrib non-free » comme ci-dessous :

```
admin01srv@SRV-Ralph: ~
GNU nano 2.2.6      Fichier : /etc/apt/sources.list
#
# deb cdrom:[Debian GNU/Linux 8.7.1 _Jessie_ - Official amd64 CD Binary-1 20170$
#deb cdrom:[Debian GNU/Linux 8.7.1 _Jessie_ - Official amd64 CD Binary-1 201701$
deb http://ftp.fr.debian.org/debian/ jessie main contrib non-free
deb-src http://ftp.fr.debian.org/debian/ jessie main contrib non-free
deb http://security.debian.org/ jessie/updates main contrib non-free
deb-src http://security.debian.org/ jessie/updates main contrib non-free
# jessie-updates, previously known as 'volatile'
deb http://ftp.fr.debian.org/debian/ jessie-updates main contrib non-free
deb-src http://ftp.fr.debian.org/debian/ jessie-updates main contrib non-free
Lecture de 15 lignes
^G Aide      ^O Écrire    ^R Lire fich.^Y Page préc.^K Couper     ^C Pos. cur.
^X Quitter   ^J Justifier ^W Chercher  ^V Page suiv.^U Coller    ^T Orthograp.
```

### III- Configuration de la base de donnée

Pour naviguer dans le serveur mysql, utiliser la commande « mysql -u root -p »

```
root@srv-Debian:/# mysql -u root -p
//le mot de passe est P@ssword/91
```

Une fois entrer dans le serveur mysql, on crée la bdd s'intitulant ralph\_db avec les commande ci-dessous :

```
mysql> CREATE DATABASE ralph_db CHARACTER SET utf8 COLLATE utf8_general_ci;
Query OK, 1 row affected (0.00 sec)

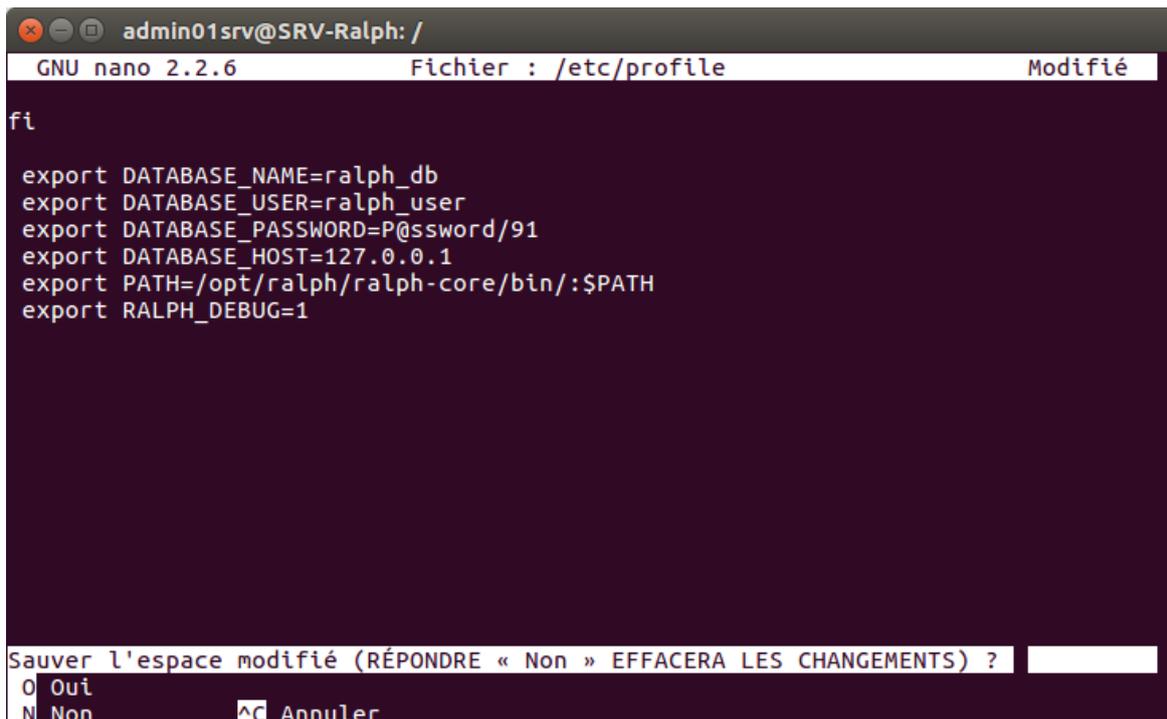
mysql> GRANT ALL PRIVILEGES ON ralph_db.* TO ralph_user@127.0.0.1 IDENTIFIED BY 'admin';
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)
```

Actuellement, le serveur ralph a besoin de lire certains variables d'environnements, donc coller le code (les variables d'environnement) ci-dessous dans le fichier ~/.profile en personnalisant avec les informations que vous voulez.

```
cat ~/.profile
```

```
export DATABASE_NAME=ralph
export DATABASE_USER=someuser
export DATABASE_PASSWORD=somepassword
export DATABASE_HOST=127.0.0.1
export PATH=/opt/ralph/ralph-core/bin/:$PATH
export RALPH_DEBUG=1
```



```
admin01srv@SRV-Ralph: /
GNU nano 2.2.6      Fichier : /etc/profile      Modifié
ft
export DATABASE_NAME=ralph_db
export DATABASE_USER=ralph_user
export DATABASE_PASSWORD=P@ssword/91
export DATABASE_HOST=127.0.0.1
export PATH=/opt/ralph/ralph-core/bin/:$PATH
export RALPH_DEBUG=1
Sauver l'espace modifié (RÉPONDRE « Non » EFFACERA LES CHANGEMENTS) ?
O Oui
N Non      ^C Annuler
```

Pour vérifier si la bdd a bien été créée :

```
mysql> SHOW DATABASES;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| ralph          |
+-----+
4 rows in set (0.00 sec)
```

Initialisation :

1. Tapez 'ralph migrate' pour créer les tables dans votre bdd.
2. Tapez 'ralph sitetree resync\_apps' pour relancer le menu.
3. Tapez 'ralph createsuperuser' pour ajouter un utilisateur

Des commandes :

// Pr appliquer les variables qu'on a exporté sur /etc/profile :

```
source /etc/profile
```

```
// pr verifier la version du mysql
```

```
# mysql -v -p
```

```
Server version: 5.5.53-0+deb8u1 (Debian)
```

```
root@SRV-Ralph:/opt/ralph/ralph-core/bin# ./ralph migrate
```

**PB :**

```
File "/opt/ralph/ralph-core/lib/python3.4/site-packages/cffi/ffiplatform.py", line 65, in _build  
raise VerificationError('%s: %s' % (e.__class__.__name__, e))  
cffi.ffiplatform.VerificationError: CompileError: command 'x86_64-linux-gnu-gcc' failed with exit  
status 1
```

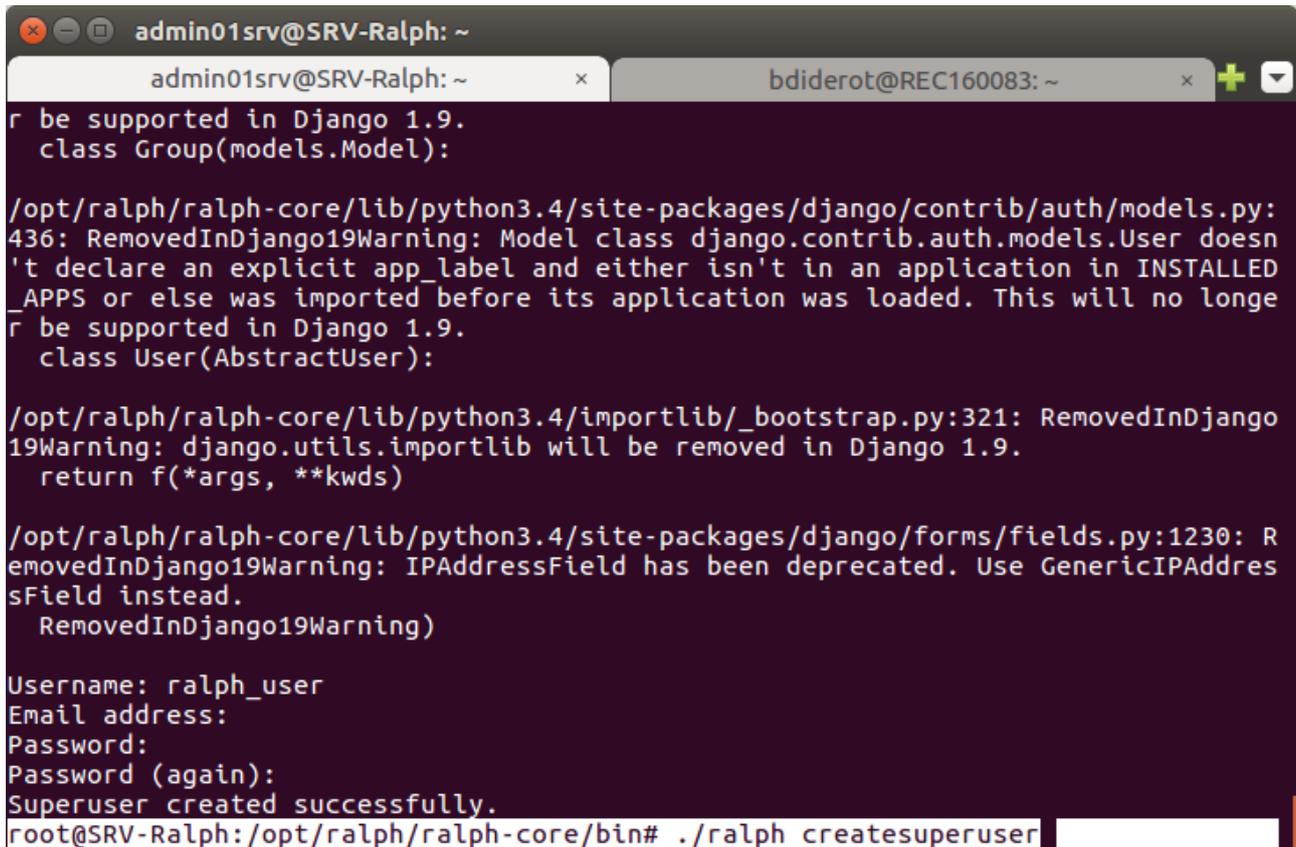
**// pour resoudre ce probleme telechargez le paquet suivante :**

```
apt-get install build-essential autoconf libtool pkg-config python-opengl python-imaging python-pyrex python-  
pyside.qtopengl idle-python2.7 qt4-dev-tools qt4-designer libqtgui4 libqtcore4 libqt4-xml libqt4-test libqt4-script libqt4-  
network libqt4-dbus python-qt4 python-qt4-gi libgle3 python-dev
```

```
root@SRV-Ralph:/opt/ralph/ralph-core/bin# ./ralph migrate
```

//normalement ça dois marcher

```
root@SRV-Ralph:/opt/ralph/ralph-core/bin# ./ralph sitetree_resync_apps
root@SRV-Ralph:/opt/ralph/ralph-core/bin# ./ralph createsuperuser
```



```
admin01srv@SRV-Ralph: ~
admin01srv@SRV-Ralph: ~
bdiderot@REC160083: ~
r be supported in Django 1.9.
class Group(models.Model):

/opt/ralph/ralph-core/lib/python3.4/site-packages/django/contrib/auth/models.py:
436: RemovedInDjango19Warning: Model class django.contrib.auth.models.User doesn
't declare an explicit app_label and either isn't in an application in INSTALLED
_APPS or else was imported before its application was loaded. This will no longe
r be supported in Django 1.9.
class User(AbstractUser):

/opt/ralph/ralph-core/lib/python3.4/importlib/_bootstrap.py:321: RemovedInDjango
19Warning: django.utils.importlib will be removed in Django 1.9.
return f(*args, **kwds)

/opt/ralph/ralph-core/lib/python3.4/site-packages/django/forms/fields.py:1230: R
emovedInDjango19Warning: IPAddressField has been deprecated. Use GenericIPAdres
sField instead.
RemovedInDjango19Warning)

Username: ralph_user
Email address:
Password:
Password (again):
Superuser created successfully.
root@SRV-Ralph:/opt/ralph/ralph-core/bin# ./ralph createsuperuser
```

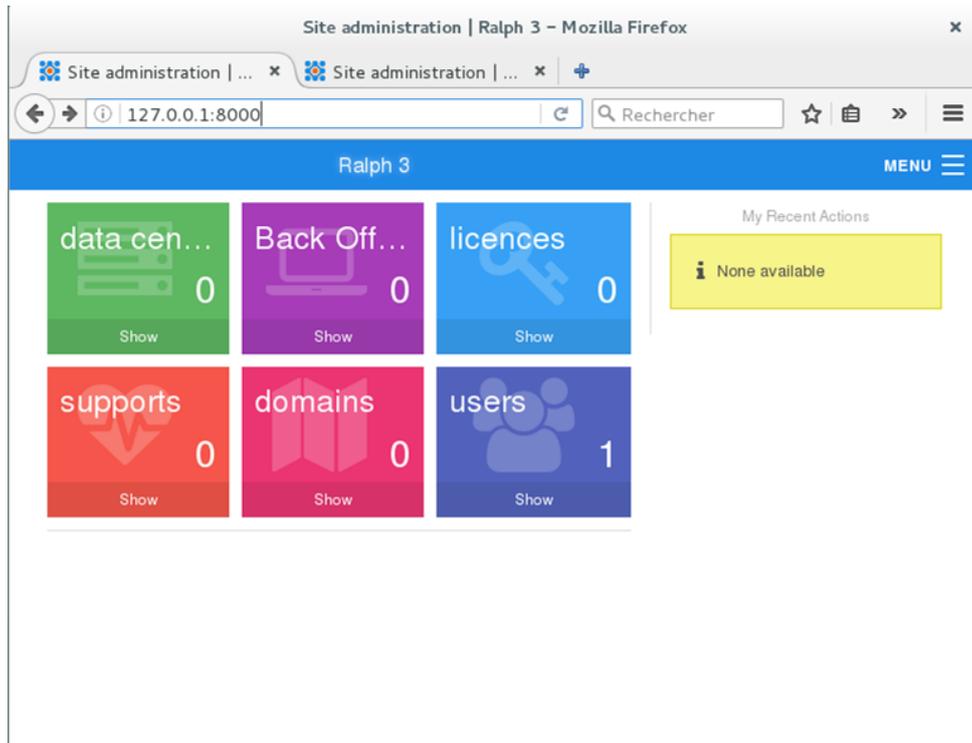
//le mot de passe est admin01srv

#### **IV- Lancer le serveur ralph**

Lancez le serveur ralph avec :

```
root@ralph:# ralph runserver 0.0.0.0:8000
```

Il vous reste à pointer votre navigateur à 'http://localhost:8000' et connecter avec vos identifiants...  
Happy ralphing ....



V-

### Installation du serveur web Nginx (reverse proxy)

```
# aptitude search nginx
root@SRV-Ralph:/opt/ralph/ralph-core/bin# aptitude install nginx-full
# vim /etc/nginx/sites-enabled/default

nginx -t //permet de verifier si le serveur web est bien installé
Nano /etc/hosts
//puis ajoutez une ligne comme ceci : 127,0,0,1 ralph.in.ac-versailles.fr
```

```
admin01srv@SRV-Ralph: ~
admin01srv@srv-Debian... x admin01srv@srv-Debian: ~ x admin01srv@SRV-Ralph: ~ x
GNU nano 2.2.6 Fichier : /etc/hosts
127.0.0.1 localhost
127.0.1.1 SRV-Ralph
127.0.0.1 ralph.in.ac-versailles.fr
# The following lines are desirable for IPV6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

```

# Default server configuration
#
server {
    #listen 80 default_server;

    # SSL configuration
    #
    listen 443 ssl default_server;
    #
    listen [::]:443 default_server;
    #
    # Self signed certs generated by the ssl-cert package
    # Don't use them in a production server!
    #
    # include snippets/snakeoil.conf;
    ssl on;

    ssl_certificate      /etc/ssl/certs/ssl-cert-snakeoil.pem;
    ssl_certificate_key  /etc/ssl/private/ssl-cert-snakeoil.key;

    ssl_session_timeout 5m;

    ssl_protocols        TLSv1 TLSv1.1 TLSv1.2;

    access_log  /var/log/nginx/ralph-access.log;
    error_log   /var/log/nginx/ralph-error.log;
    root        /var/www/html;

    # Add index.php to the list if you are using PHP
    index index.html index.htm index.nginx-debian.html;

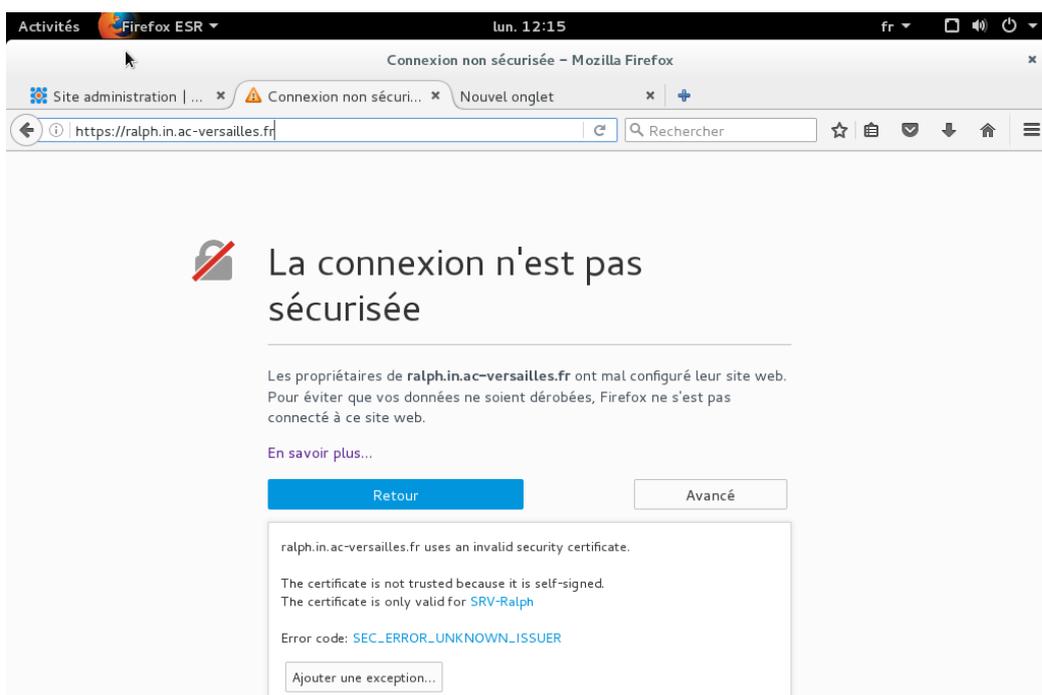
    server_name ralph.in.ac-versailles.fr;

    location / {
        proxy_pass      http://localhost:8000;
        include          /etc/nginx/proxy_params;

        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        try_files $uri $uri/ =404;
    }
}

```

//tapez l'url suivante dans le navigateur



Activités | Navigateur Web Chromium | mar. 09:17 | fr

Site administration | Not Secure | https://ralph.in.ac-versailles.fr

Ralph 3

Data Center | Back Office | Networks | Licenses | Domains | Supports | Reports | Operations | Dashboards | Settings

ralph\_user

data center a...

237

Show

Back Office ...

0

Show

licences

0

Show

supports

0

Show

domains

0

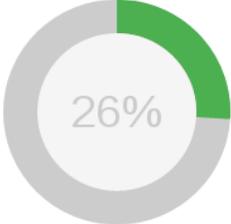
Show

users

2

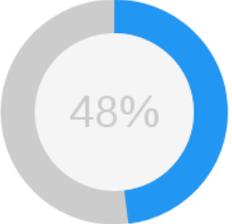
Show

Bât A



26%

Bât B



48%

My Recent Actions

- 🔑 SRV-SQLSERVER1-SMB (BC: 199 / SN: 199) Data center asset 1 week, 4 days ago
- 🚫 test (0.0.0.0/32 | VLAN: None) Network 1 week, 4 days ago
- + test (192.61.231.0/26 | VLAN: None) Network 1 week, 4 days ago
- 🔑 SRV-SQLSERVER1-SMB (BC: 199 / SN: 199) Data center asset 1 week, 4 days ago
- + black screen Incident 1 week, 4 days ago
- 🔑 DHCP Network environment 1 week, 4 days ago
- 🔑 Salle de machine A (Bât A) Server room 1 week, 4 days ago
- 🔑 Switch fibre channel (BC: 189 / SN: 189) Data center asset 1 week, 4 days ago
- 🔑 Rcade ethernet SMA Baie B (BC: 190 / SN: 190) Data center asset 1 week, 4 days ago
- 🔑 Rcade FC vers salle machine (BC: 192 / SN: 192) Data center asset 1 week, 4 days ago
- 🚫 i (BC: 182 / SN: 182) Data center asset 1 week, 4 days ago
- 🚫 ij (BC: 181 / SN: 181) Data center asset 1 week, 4 days ago
- 🚫 i (BC: 180 / SN: 180) Data center asset 1 week, 4 days ago

**VI- Redirection Http vers https :**

```

admin01srv@SRV-Ralph: ~
admin01srv@srv-Debian: /var/www... x admin01srv@srv-Debian: ~ x admin01srv@SRV-Ralph: ~ x
GNU nano 2.2.6 Fichier : default
#
server {
    listen 80 default_server;
    #
    listen [::]:80;

    server_name ralph.in.ac-versailles.fr;
    #Redirige le http vers https
    #
    return 301 https://$ralph.in.ac-versailles.fr$ralph.in.ac-versailles.fr?;
    rewrite ^ https://ralph.in.ac-versailles.fr$1 permanent;

    #
    #
    root /var/www/example.com;
    #
    index index.html;
    #
    #
    location / {
    #
        try_files $uri $uri/ =404;
    #
    }
}

^G Aide ^O Écrire ^R Lire fich. ^Y Page préc. ^K Couper ^C Pos. cur.
^X Quitter ^J Justifier ^W Chercher ^V Page suiv. ^U Coller ^T Orthograp.

```

// accéder au page web du server non sécurisée, il doit rediriger automatiquement vers la page web sécurisée.  
 (http://ralph.in.ac-versailles.fr)

The screenshot displays the Ralph 3 administration web interface. At the top, there's a navigation bar with menus for Data Center, Back Office, Networks, Licenses, Domains, Supports, Reports, Operations, and Dashboards. Below this is a dashboard with several colored tiles: 'data center a...' (237), 'Back Office ...' (0), 'licences' (0), 'supports' (0), 'domains' (0), and 'users' (2). At the bottom, there are two donut charts: 'Bât A' showing 26% and 'Bât B' showing 48%. On the right side, a 'My Recent Actions' sidebar lists various system events and user actions, such as 'SRV-SQLSERVER1-SMB' and 'black screen Incident'.

## VII-Ralph API

//pr generer la clé api

```
root@SRV-Ralph# curl -H "Content-Type:application/json" -X POST http://10.0.2.15:8000/api-token-auth/ -d '{"username": "ralph_user", "password": "admin"}'
```

```
{"token":"1084427f22a4b48edcad084b29dc50470727dd44"}
```

### Get sample resource

Use HTTP GET method to get details of the resource. Example:

//ds le serveur netbox,tapez l'url suivante http://10,0,2,15:8000/api

Django REST framework v 3.2.2 ralph\_user

# Api Root

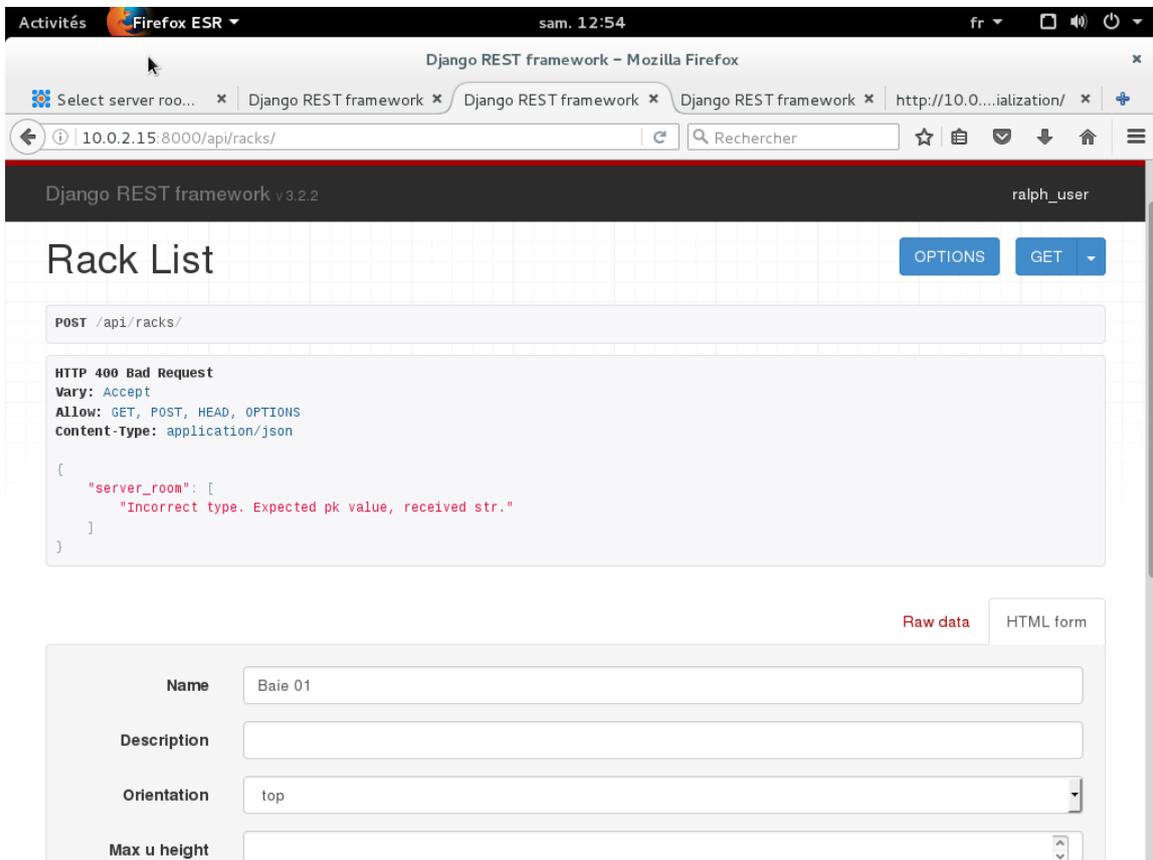
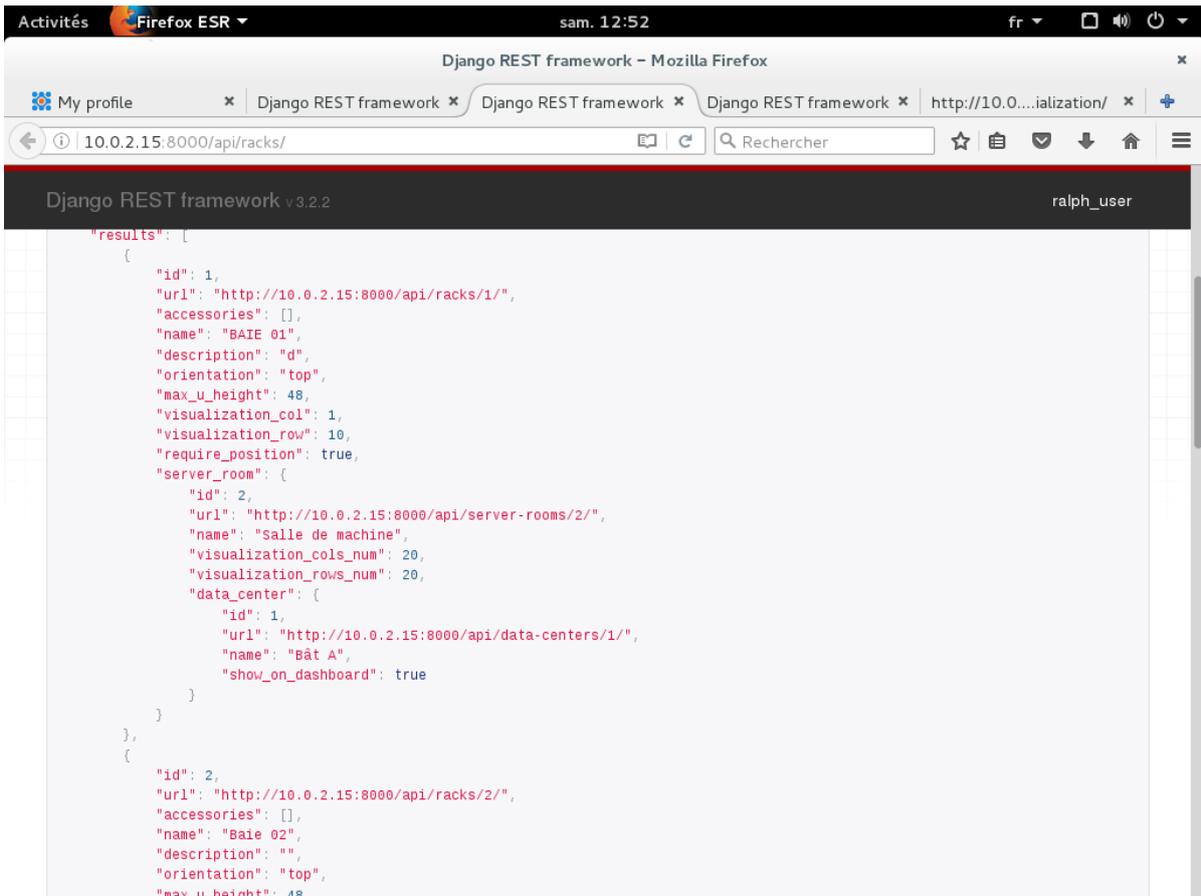
[OPTIONS](#) [GET](#)

GET /api/

**HTTP 200 OK**  
**Vary:** Accept  
**Allow:** GET, HEAD, OPTIONS  
**Content-Type:** application/json

```
{
  "accessories": "http://10.0.2.15:8000/api/accessories/",
  "assetholders": "http://10.0.2.15:8000/api/assetholders/",
  "assetmodels": "http://10.0.2.15:8000/api/assetmodels/",
  "back-office-assets": "http://10.0.2.15:8000/api/back-office-assets/",
  "base-object-clusters": "http://10.0.2.15:8000/api/base-object-clusters/",
  "base-objects": "http://10.0.2.15:8000/api/base-objects/",
  "base-objects-licences": "http://10.0.2.15:8000/api/base-objects-licences/",
  "base-objects-supports": "http://10.0.2.15:8000/api/base-objects-supports/",
  "budget-info": "http://10.0.2.15:8000/api/budget-info/",
  "business-segments": "http://10.0.2.15:8000/api/business-segments/",
  "categories": "http://10.0.2.15:8000/api/categories/",
  "cloud-flavors": "http://10.0.2.15:8000/api/cloud-flavors/",
  "cloud-hosts": "http://10.0.2.15:8000/api/cloud-hosts/",
  "cloud-projects": "http://10.0.2.15:8000/api/cloud-projects/",
  "cloud-providers": "http://10.0.2.15:8000/api/cloud-providers/",
  "cluster-types": "http://10.0.2.15:8000/api/cluster-types/",
  "clusters": "http://10.0.2.15:8000/api/clusters/",
  "configuration-classes": "http://10.0.2.15:8000/api/configuration-classes/",
  "configuration-modules": "http://10.0.2.15:8000/api/configuration-modules/",
  "custom-fields": "http://10.0.2.15:8000/api/custom-fields/",
```

//url: 10.0.2.15:8000/api/racks et on peut meme faire des query (filter)



## 5) Installation Netbox

## I- Installation PostgreSQL

Netbox a requiert PostgreSQL base de donnée pour stocker les données. Attention, le Mysql n'est pas compatible avec netbox.

```
# apt-get update
# apt-get install -y postgresql libpq-dev
```

Une fois télécharger, lancer le service postgresql

```
# systemctl start postgresql
```

## II- Création bdd

Au minimum, on a besoin de créer une base de donnée pour Netbox et assigner un utilisateur et un mot de passe pour l'authentification. Ceci est fait avec les commandes suivantes :

```
# su - postgres
postgres@SRV-Netbox:~$ psql
psql (9.4.10)
Saisissez « help » pour l'aide.

postgres=# CREATE DATABASE netbox_db;
ERREUR: la base de données « netbox_db » existe déjà
postgres=# CREATE USER netbox_user WITH PASSWORD 'admin';
ERREUR: le rôle « netbox_user » existe déjà
postgres=# GRANT ALL PRIVILEGES ON DATABASE netbox_db TO netbox_user;
GRANT
postgres=# \quit
```

Pour vérifier si l'authentification se fonctionne, Utiliser la commande suivante avec le mot de passe configuré :

```
#psql -u netbox -h localhost -W
```

## III-Installation Netbox

```
# apt-get install -y python2.7 python-dev python-pip libxml2-dev libxslt1-dev libffi-dev graphviz libp
# apt-get install -y git
```

Cloner le Git dépôt:

Crée une répertoire pour l'installation de Netbox :

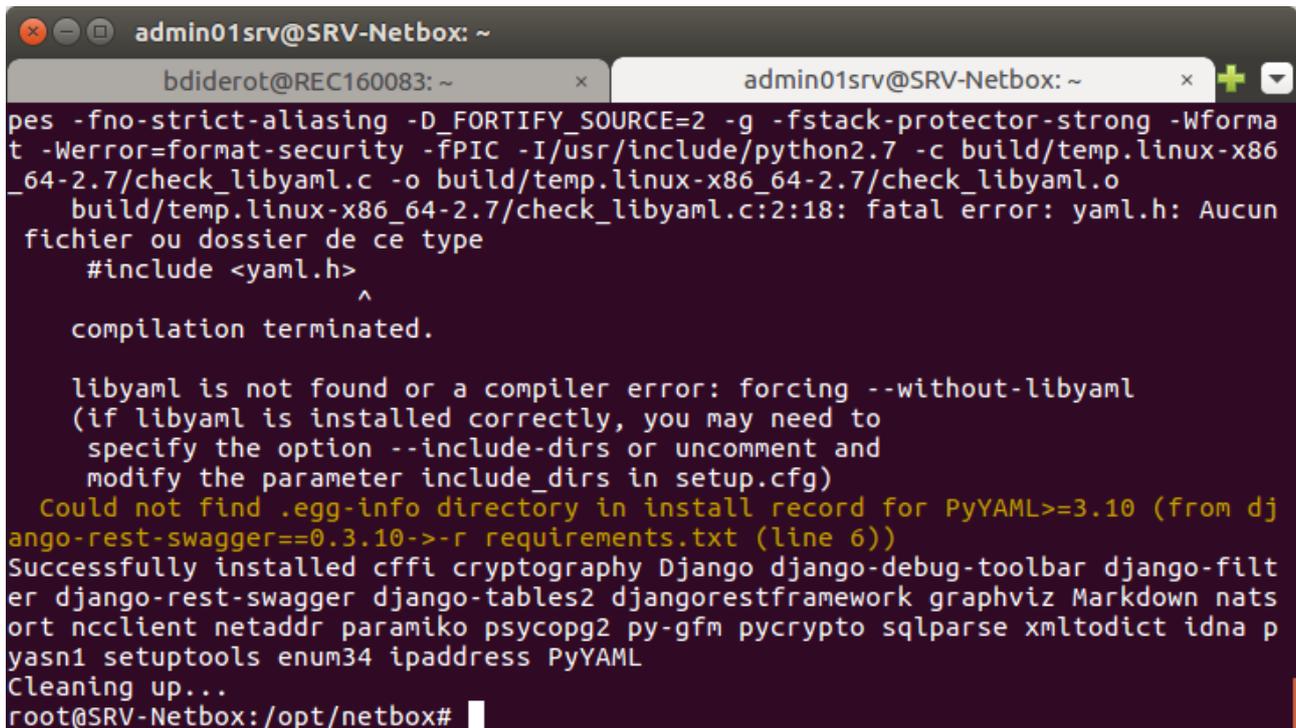
```
mkdir -p /opt/netbox/ && cd /opt/netbox/
```

```
root@SRV-Netbox:/opt/netbox# git clone -b master https://github.com/digitalocean/netbox.git .
Clonage dans '.'...
remote: Counting objects: 7682, done.
remote: Compressing objects: 100% (186/186), done.
remote: Total 7682 (delta 111), reused 0 (delta 0), pack-reused 7496
Réception d'objets: 100% (7682/7682), 3.16 MiB | 1.49 MiB/s, fait.
Résolution des deltas: 100% (5917/5917), fait.
Vérification de la connectivité... fait.
```

## IV- Installer les paquets python

Installer les paquets python requiert avec la commande pip

```
# pip install -r requirements.txt
```



```
admin01srv@SRV-Netbox: ~  
bdiderot@REC160083: ~ x admin01srv@SRV-Netbox: ~ x  
pes -fno-strict-aliasing -D_FORTIFY_SOURCE=2 -g -fstack-protector-strong -Wforma  
t -Werror=format-security -fPIC -I/usr/include/python2.7 -c build/temp.linux-x86  
_64-2.7/check_libyaml.c -o build/temp.linux-x86_64-2.7/check_libyaml.o  
build/temp.linux-x86_64-2.7/check_libyaml.c:2:18: fatal error: yaml.h: Aucun  
fichier ou dossier de ce type  
#include <yaml.h>  
^  
compilation terminated.  
  
libyaml is not found or a compiler error: forcing --without-libyaml  
(if libyaml is installed correctly, you may need to  
specify the option --include-dirs or uncomment and  
modify the parameter include_dirs in setup.cfg)  
Could not find .egg-info directory in install record for PyYAML>=3.10 (from dj  
ango-rest-swagger==0.3.10->-r requirements.txt (line 6))  
Successfully installed cffi cryptography Django django-debug-toolbar django-filt  
er django-rest-swagger django-tables2 djangorestframework graphviz Markdown nats  
ort ncclient netaddr paramiko pycppg2 py-gfm pycrypto sqlparse xmldict idna p  
yasni setuptools enum34 ipaddress PyYAML  
Cleaning up...  
root@SRV-Netbox:/opt/netbox#
```

### V-Configuration

Allez dans le répertoire de configuration de netbox et fait un copie de 'configuration.example.py' en nommant configuration.py

```
root@SRV-Netbox:/opt/netbox# cd /opt/netbox/netbox/netbox  
root@SRV-Netbox:/opt/netbox/netbox/netbox# cp configuration.example.py configuration.py
```

Ouvrir le fichier 'configuration.py' et assigner les variables suivantes :

- Allowed\_HOSTS
- DATABASE
- SECRET\_KEY

a) Allowed\_hosts

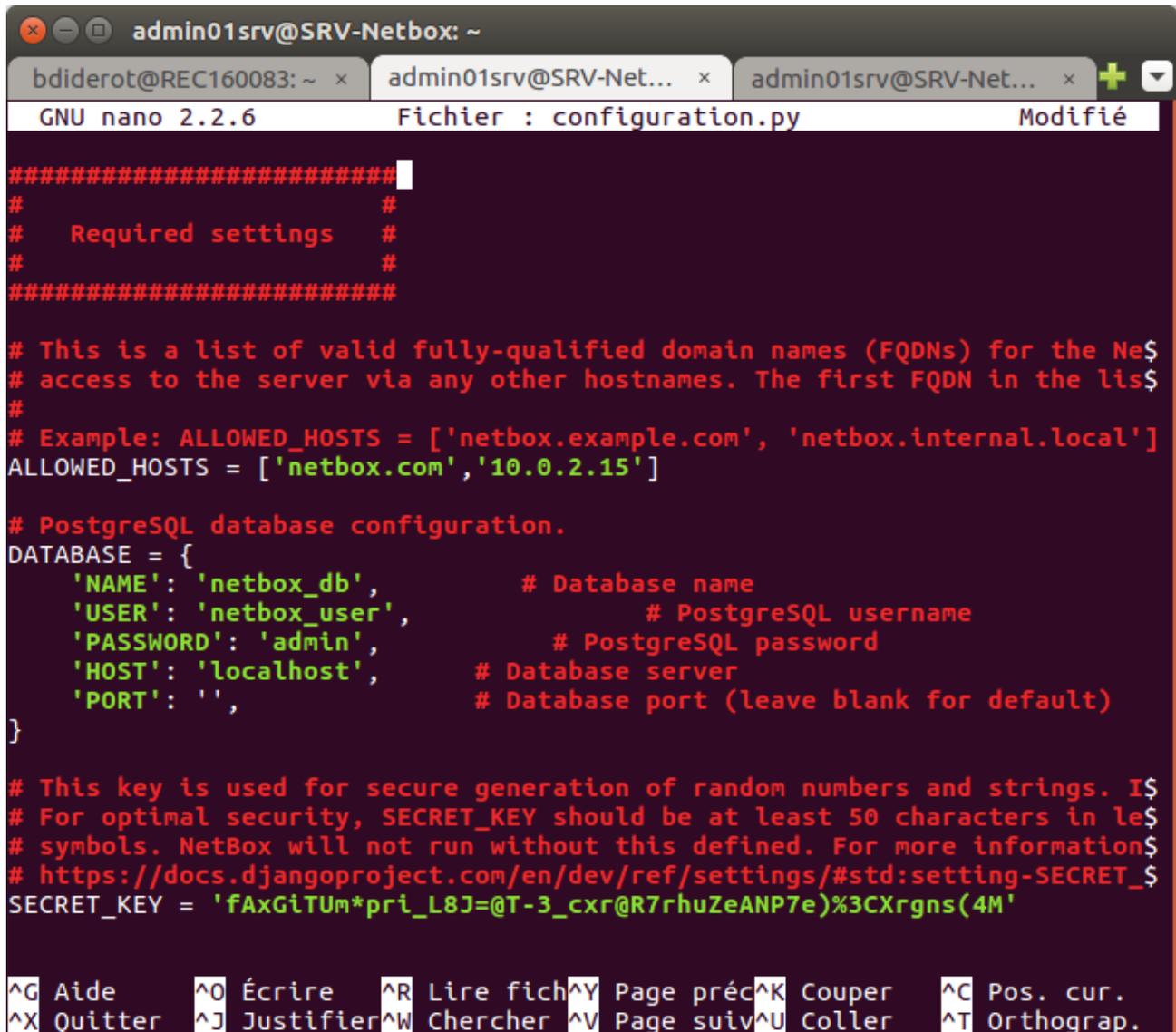
Ceci est une liste des hôtes par lequel le serveur est joignable. On devez spécifier le nom ou l'adresse ip.

Example:

```
ALLOWED_HOSTS = ['netbox.com', '10,0,2,15']
```

b) Base de donnée

On doit définir l'identifiant et le mot de passe utilisé lors de la configuration de postgresql.



```
admin01srv@SRV-Netbox: ~
bdiderot@REC160083: ~ x admin01srv@SRV-Net... x admin01srv@SRV-Net... x
GNU nano 2.2.6 Fichier : configuration.py Modifié
#####
# Required settings
#####

# This is a list of valid fully-qualified domain names (FQDNs) for the Ne$
# access to the server via any other hostnames. The first FQDN in the lis$
#
# Example: ALLOWED_HOSTS = ['netbox.example.com', 'netbox.internal.local']
ALLOWED_HOSTS = ['netbox.com', '10.0.2.15']

# PostgreSQL database configuration.
DATABASE = {
    'NAME': 'netbox_db',          # Database name
    'USER': 'netbox_user',       # PostgreSQL username
    'PASSWORD': 'admin',        # PostgreSQL password
    'HOST': 'localhost',        # Database server
    'PORT': '',                 # Database port (leave blank for default)
}

# This key is used for secure generation of random numbers and strings. I$
# For optimal security, SECRET_KEY should be at least 50 characters in le$
# symbols. NetBox will not run without this defined. For more informatio$
# https://docs.djangoproject.com/en/dev/ref/settings/#std:setting-SECRET_$
SECRET_KEY = 'fAxGiTUm*pri_L8J=@T-3_cxr@R7rhuZeANP7e)%3CXrgns(4M'
```

c) Secret\_Key

Générer une clé aléatoire de minimum de 50 caractère alphanumeric. La clé doit être unique et ne doit pas être partager à l'extérieur de la machine locale.

On pouvez utiliser le script situé dans le répertoire suivant pour générer une clé unique :

```
# netbox/generate_secret_key.py
```

## VI- Migration de la bdd

Avant qu'on lance le Netbox, il faut installer le schéma de la bdd. Utiliser la commande suivante:

```
# cd /opt/netbox/netbox/  
# ./manage.py migrate  
Operations to perform:  
  Apply all migrations: dcim, sessions, admin, ipam, utilities, auth, circuits, contenttypes, extras,  
secrets, users  
Running migrations:  
  Rendering model states... DONE  
  Applying contenttypes.0001_initial... OK  
  Applying auth.0001_initial... OK  
  Applying admin.0001_initial... OK
```

### VII- Création d'un utilisateur global

Netbox a aucune compte utilisateur prédéfini. On doit créer un utilisateur pour se connecter à Netbox.

```
root@SRV-Netbox:/opt/netbox/netbox# ./manage.py createsuperuser  
Username (leave blank to use 'root'): netbox_user  
Email address:  
Password: //admin  
Password (again): //admin  
Superuser created successfully.
```

### VIII-Collecter les fichiers statiques

```
root@SRV-Netbox:/opt/netbox/netbox# ./manage.py collectstatic  
  
You have requested to collect static files at the destination  
location as specified in your settings:  
  
  /opt/netbox/netbox/static  
  
This will overwrite existing files!  
Are you sure you want to do this?  
  
Type 'yes' to continue, or 'no' to cancel: yes
```

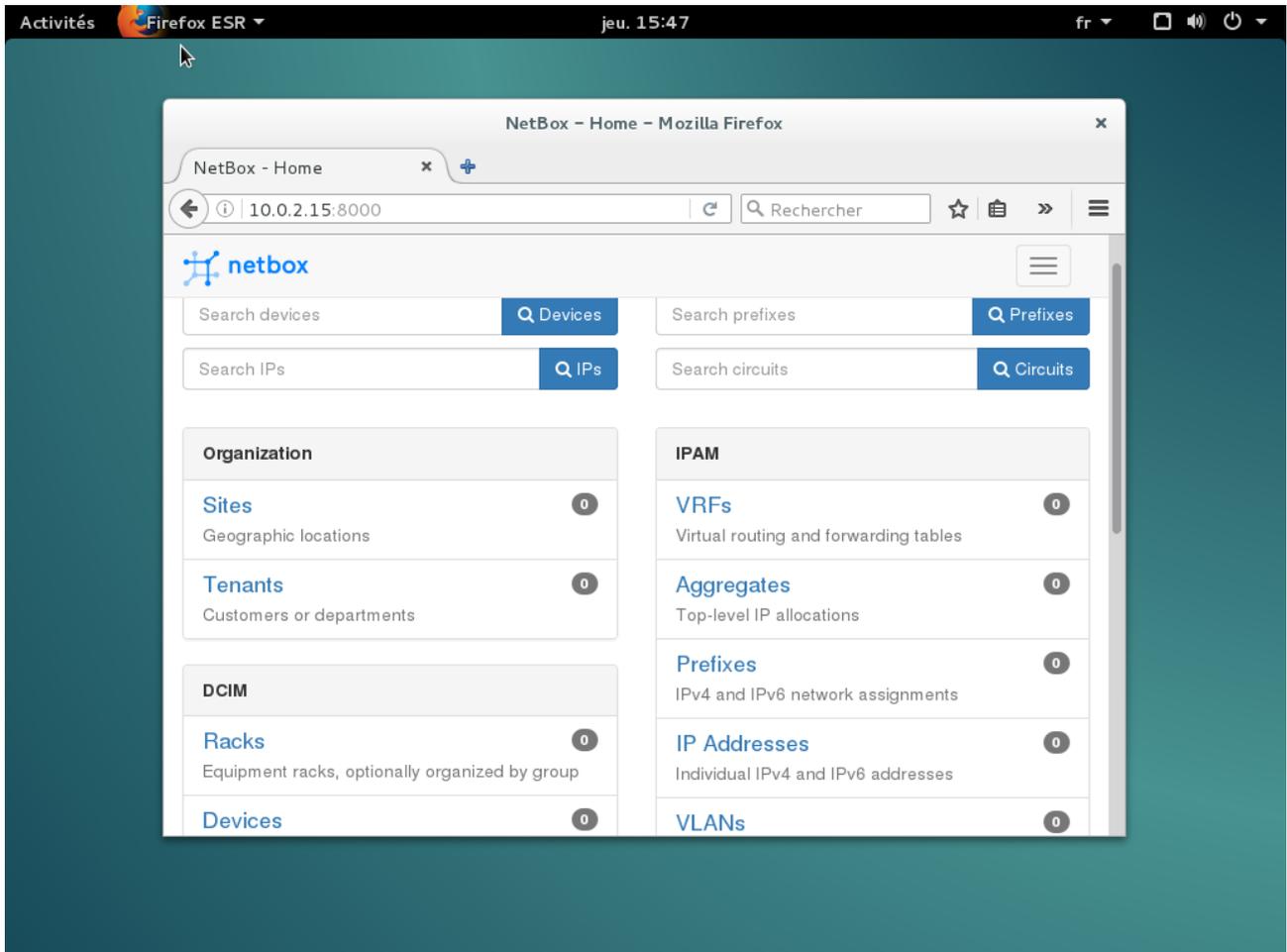
### IX- Tester l'application

Pour lancer le serveur Netbox, utiliser la commande ci-dessous :

```
# ./manage.py runserver 0.0.0.0:8000 --insecure
Performing system checks...

System check identified no issues (0 silenced).
June 17, 2016 - 16:17:36
Django version 1.9.7, using settings 'netbox.settings'
Starting development server at http://0.0.0.0:8000/
Quit the server with CONTROL-C.
```

Pointer votre navigateur avec l'@ip du serveur:8000  
ex : 10,0,2,15:8000



X-Installation du serveur web

```
# apt-get install -y gunicorn supervisor
# apt-get install -y nginx
```

Une fois qu'on installe le nginx, sauvegarder les configurations suivantes dans le répertoire /etc/nginx/sites-available/netbox.

```
server {
    listen 80;

    server_name 10.0.2.15;

    access_log off;

    location /static/ {
        alias /opt/netbox/netbox/static/;
    }

    location / {
        proxy_pass http://127.0.0.1:8001;
        proxy_set_header X-Forwarded-Host $server_name;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-Proto $scheme;
        add_header P3P 'CP="ALL DSP COR PSAa PSDa OUR NOR ONL UNI COM NAV"';
    }
}

Then
```

Puis, supprimer le fichier /etc/nginx/sites-enabled/default/ et créer une liaison symbolique dans 'sites-enabled' au fichier de configuration qu'on vient de créer.

```
# cd /etc/nginx/sites-enabled/
# rm default
# ln -s /etc/nginx/sites-available/netbox

Redémarrer le service nginx
# service nginx restart
```

## XI- Installation gunicorn

Enregistrer les configurations suivantes dans le répertoire de configuration de netbox (/opt/netbox/gunicorn\_config.py).

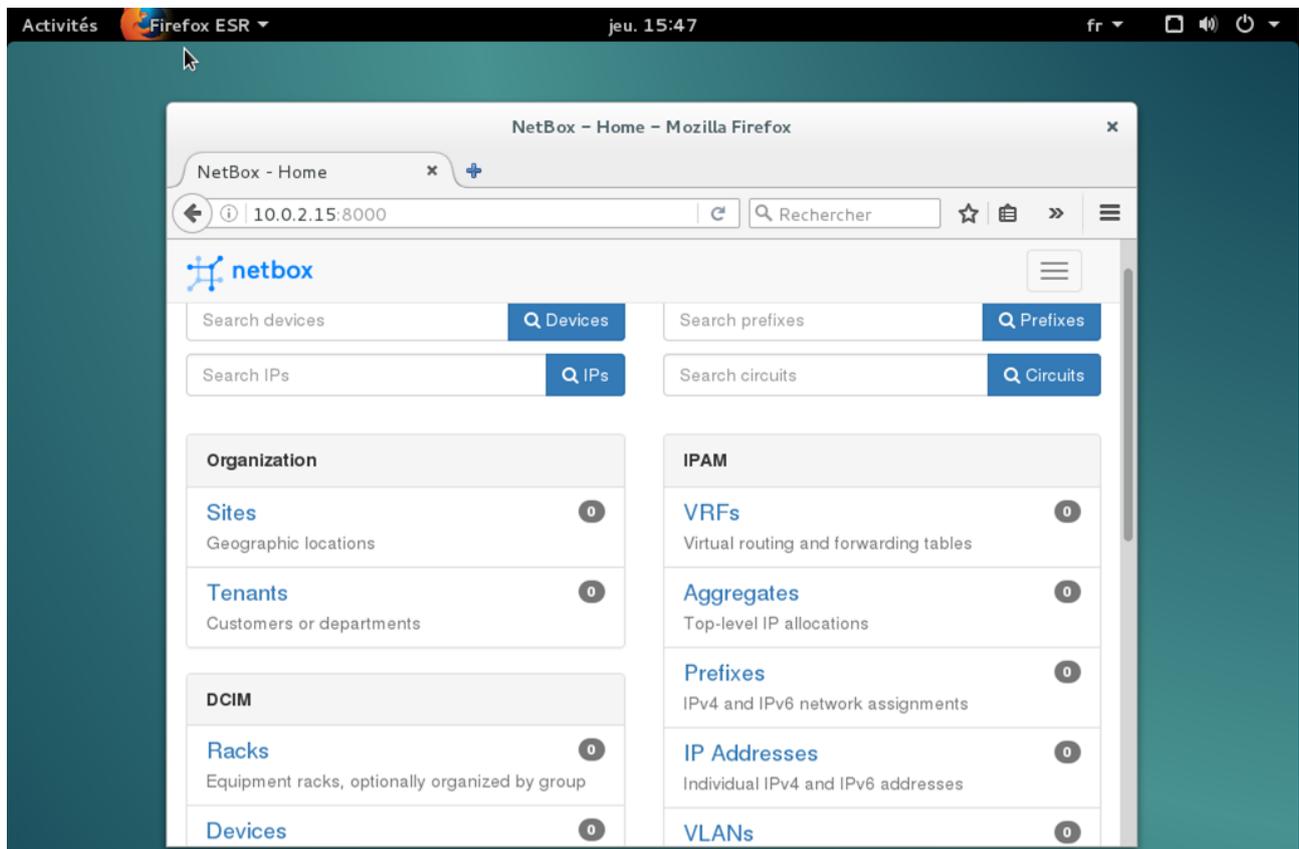
```
command = '/usr/bin/gunicorn'  
pythonpath = '/opt/netbox/netbox'  
bind = '127.0.0.1:8001'  
workers = 3  
user = 'www-data'  
  
//journalctl -xn //pr voir le journal
```

### XII-Installation supervisord

Enregistrer les configurations suivantes tels que /etc/supervisor/conf.d/netbox.conf.

```
[program:netbox]  
command = gunicorn -c /opt/netbox/gunicorn_config.py netbox.wsgi  
directory = /opt/netbox/netbox/  
user = www-data  
  
Puis redémarrer le service supervisor  
# service supervisor restart
```

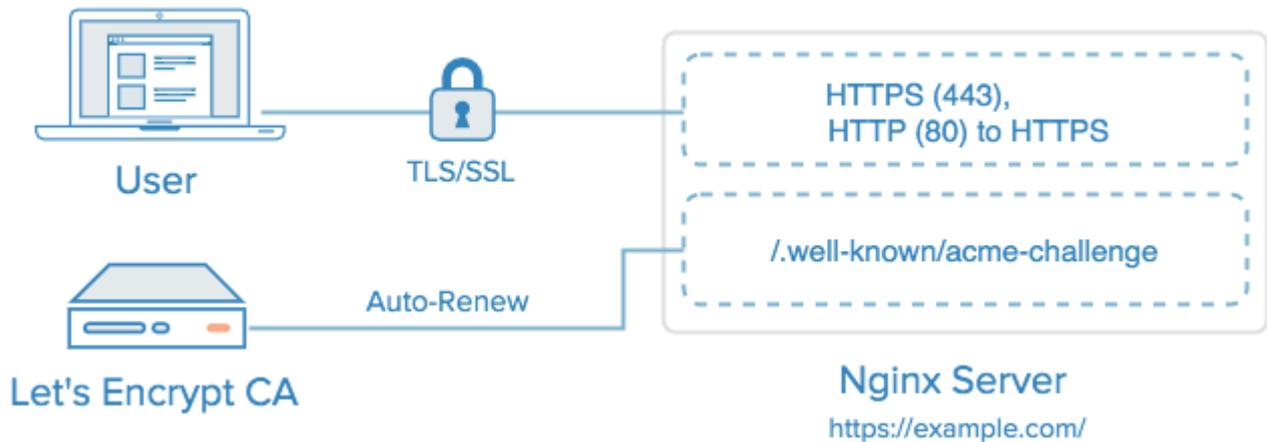
À ce point, on peut se connecter au serveur netbox avec le service nginx HTTP.



### XIII- Certificat SSL

Let's Encrypt est une nouvelle Autorité de Certificat (CA) qui fournit une façon facile d'obtenir et installer des certificats TLS/SSL libres, permettant ainsi des HTTP cryptés sur des serveurs Web. Il simplifie le processus en fournissant un client logiciel, certbot, qui essaye d'automatiser la plupart (si pas tout) des pas nécessaires. Actuellement, le processus entier d'obtention et d'installation d'un certificat est entièrement automatisé seulement sur des serveurs Web Apache.

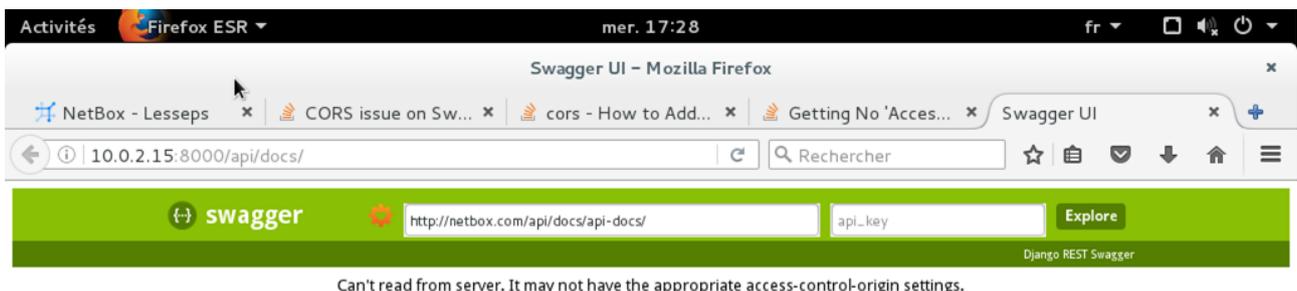
Cependant, Let's encrypt peut utiliser pour obtenir des certificats gratuit, lequel peut être installer manuellement.



### XIV- Netbox API

Netbox fournit une API de type rest en mode lecture seule qui peut être utilisée pour intégrer avec d'autres applications.

Si vous voulez construire un nouveau client d'API ou explorer simplement l'API de NetBox, la documentation Swagger peut être trouvée à l'URL/api/docs/sur un serveur de NetBox.



//Pb: il n'arrive pas à se connecter au serveur. Les cors ne sont pas activées. Ce dernier permet de protéger les données personnels contre certains sites web.

#### **Testing CORS Support**

You can verify CORS support with one of three techniques:

- Curl your API and inspect the headers. For instance:

```
root@SRV-Netbox:/etc/nginx/sites-available# curl -I "http://10.0.2.15:8000/api/docs/api-docs/"
```

HTTP/1.0 200 OK

Server: WSGIServer/0.1 Python/2.7.9

X-Frame-Options: SAMEORIGIN

Allow: GET, HEAD, OPTIONS

Date: Thu, 26 Jan 2017 15:28:44 GMT

Vary: Cookie

Content-Type: application/json

// on constate qu'il manque cette paramètre tels que options et patch dans le access-control-allow-headers et dans l'en tete (headers) Content-Type,api\_key, Authorization

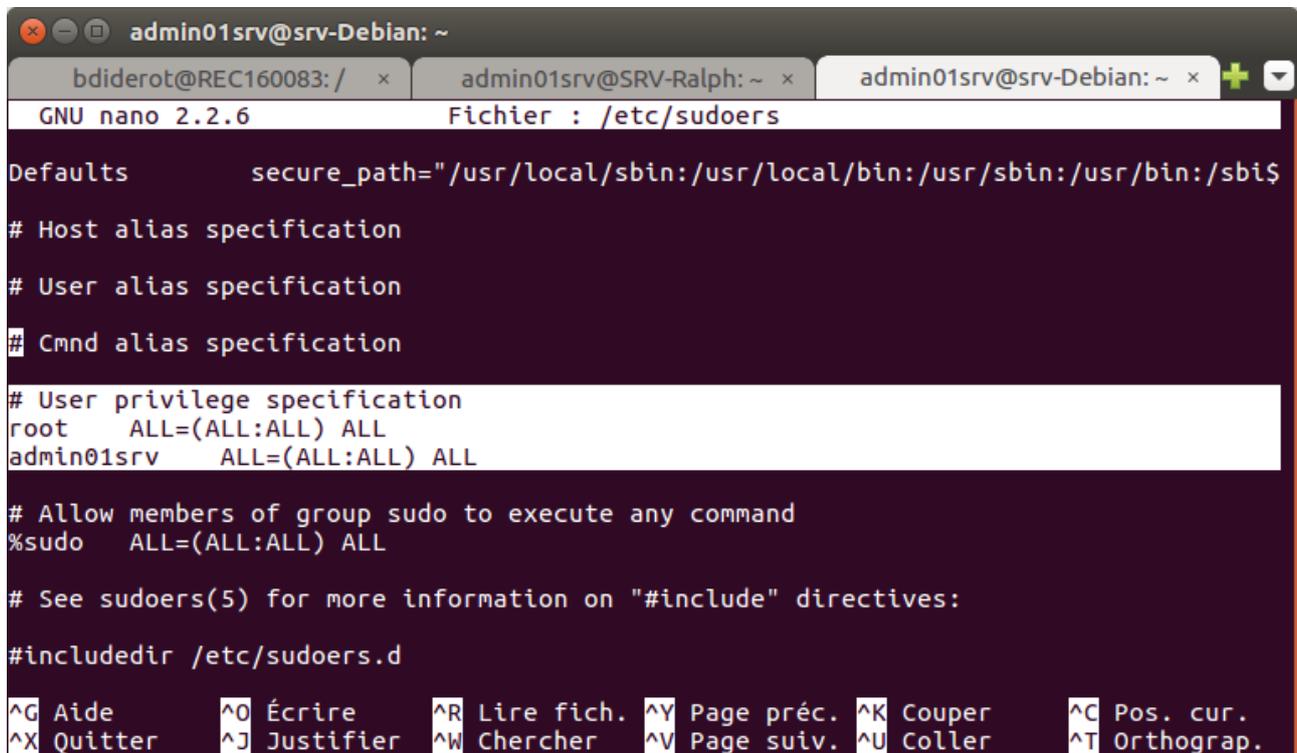
- 6) Comparaison des logiciels et choisir le meilleur
- 7) Mise en production du Racktables
  - a) Création d'un vm sous esxi (debian 8.0)

b) Installation de racktables

1. Installation des paquets

Il faut installer un serveur http qui s'occupent d'afficher les différents pages, le php version 5 qui permet de gérer des contenus dynamiques des pages web, et puis le mariadb qui est un système de gestion de base de données.

vim /etc/sudoers //pr autoriser le compte admin01srv a faire des commandes de root



```
admin01srv@srv-Debian: ~
GNU nano 2.2.6 Fichier : /etc/sudoers
Defaults secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
# Host alias specification
# User alias specification
## Cmnd alias specification
# User privilege specification
root    ALL=(ALL:ALL) ALL
admin01srv  ALL=(ALL:ALL) ALL
# Allow members of group sudo to execute any command
%sudo  ALL=(ALL:ALL) ALL
# See sudoers(5) for more information on "#include" directives:
#include_dir /etc/sudoers.d
^G Aide      ^O Écrire    ^R Lire fich. ^Y Page préc. ^K Couper    ^C Pos. cur.
^X Quitter   ^J Justifier ^W Chercher   ^V Page suiv. ^U Coller    ^T Orthograp.
```

```
sudo apt-get install software-properties-common
```

```
sudo apt-key adv --recv-keys --keyserver keyserver.ubuntu.com 0xc9cb082a1bb943db
```

```
sudo add-apt-repository 'deb [arch=amd64,i386,ppc64el] http://ftp.igh.cnrs.fr/pub/mariadb/repo/10.1/debian jessie main'
```

```
admin01srv@srv-Debian:~$ sudo vim /etc/apt/sources.list
```

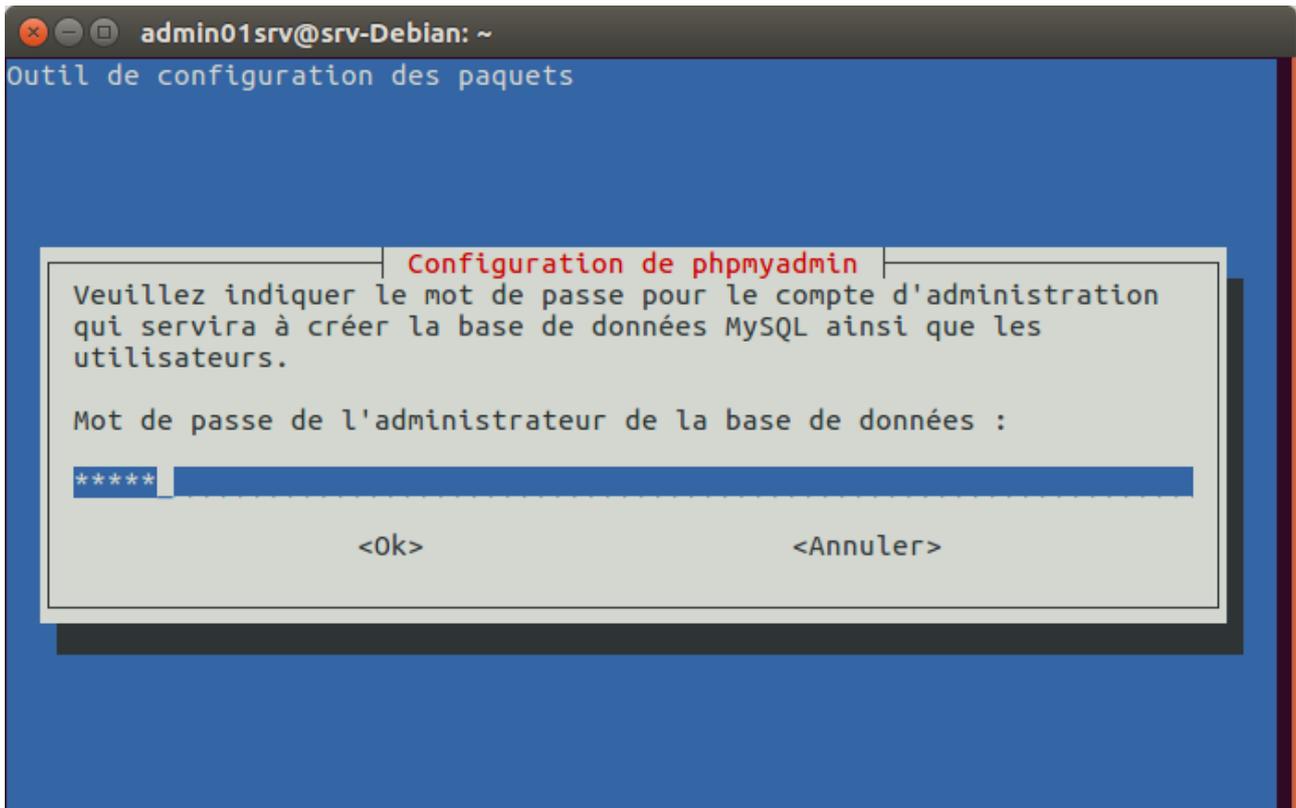
```
admin01srv@srv-Debian:~$ sudo vim /etc/apt/sources.list
```

```
admin01srv@srv-Debian:~$ sudo vim /etc/apt/sources.list.d/mariadb.list
```

/\*Saisie de mot de passe pr la base de donnée est admin

lien utile :

[https://downloads.mariadb.org/mariadb/repositories/#mirror=cnrs&distro=Debian&distro\\_release=jessie--jessie&version=10.1](https://downloads.mariadb.org/mariadb/repositories/#mirror=cnrs&distro=Debian&distro_release=jessie--jessie&version=10.1) \*/



```
admin01srv@srv-Debian:~$ sudo apt-get install python-software-properties
```

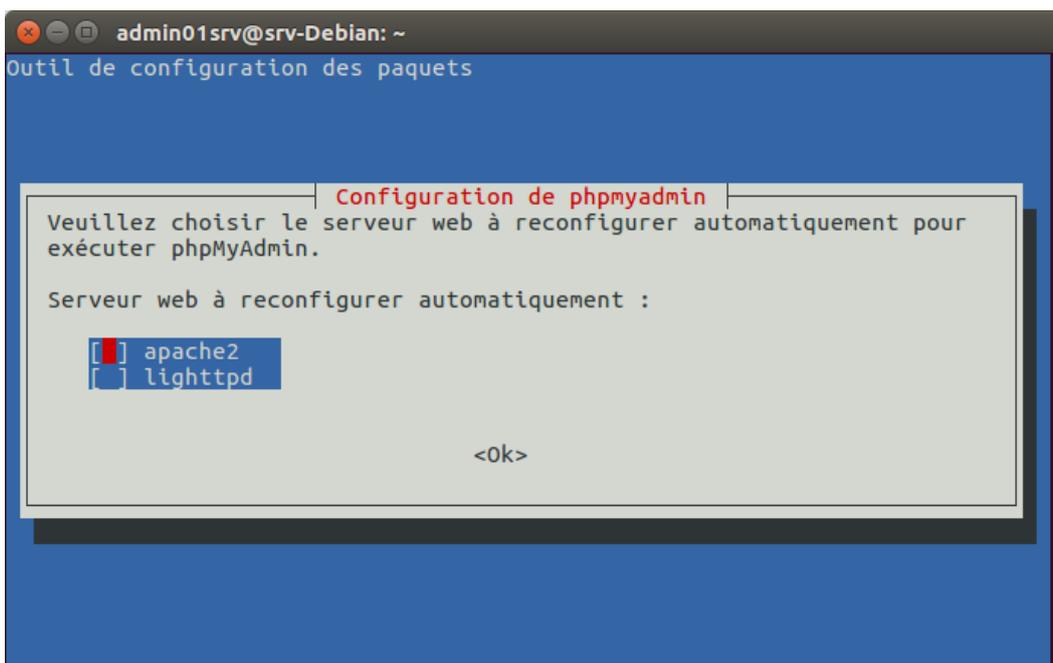
```
admin01srv@srv-Debian:~$ sudo add-apt-repository ppa:ondrej/php
```

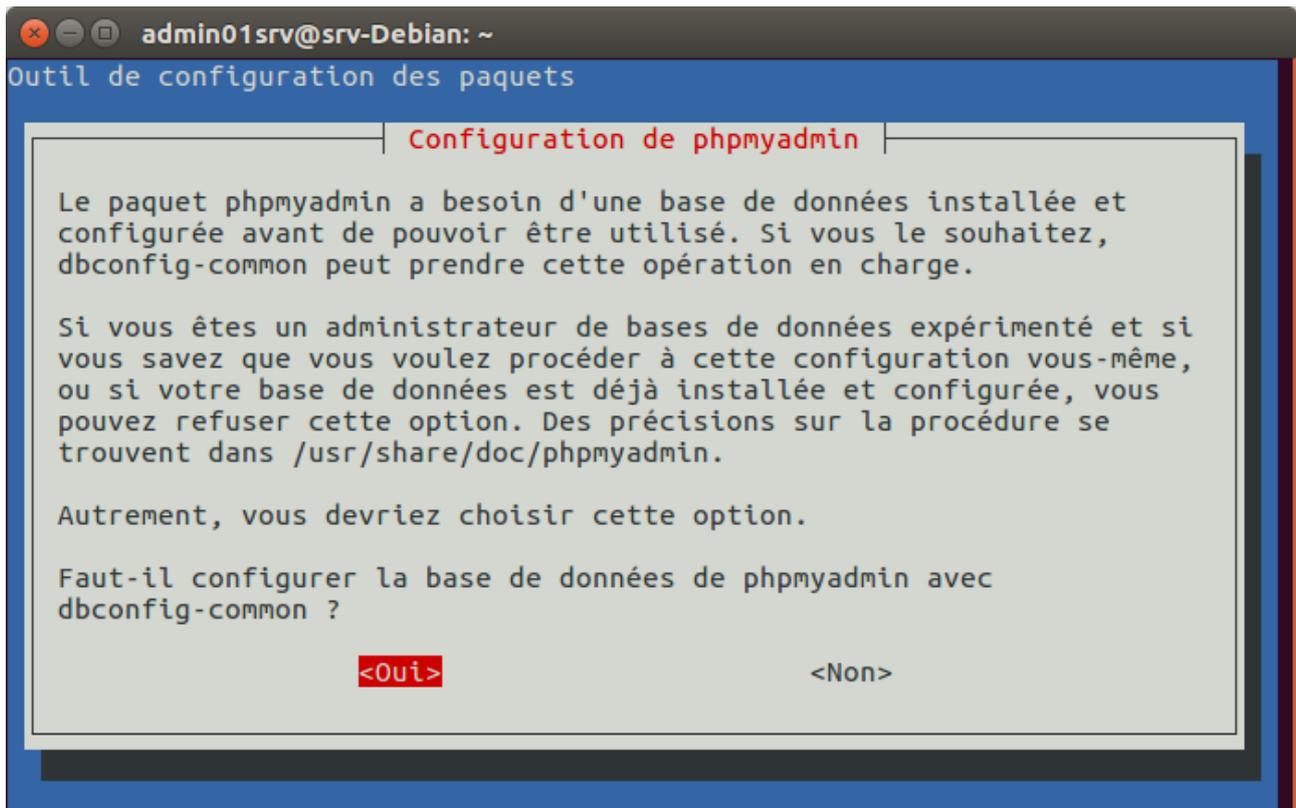
```
admin01srv@srv-Debian:~$ sudo apt-get update
```

```
admin01srv@srv-Debian:~$ sudo apt-get install -y php7.0
```

```
admin01srv@srv-Debian:~$ sudo apt-get install php7.0-mysql php7.0-curl php7.0-json php7.0-cgi
```

```
admin01srv@srv-Debian:~$ sudo apt-get install phpmyadmin
```



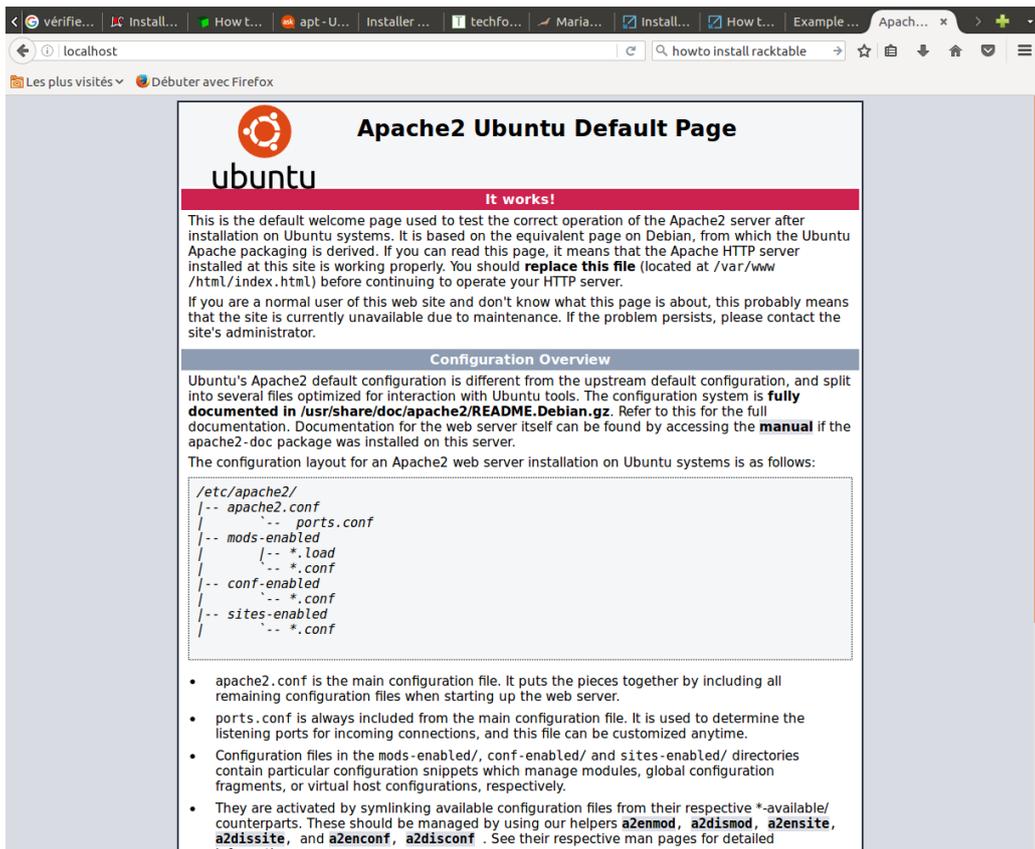


// le mot de passe est admin pr administrer la bdd, mysql

Vérification d'installation des paquets :

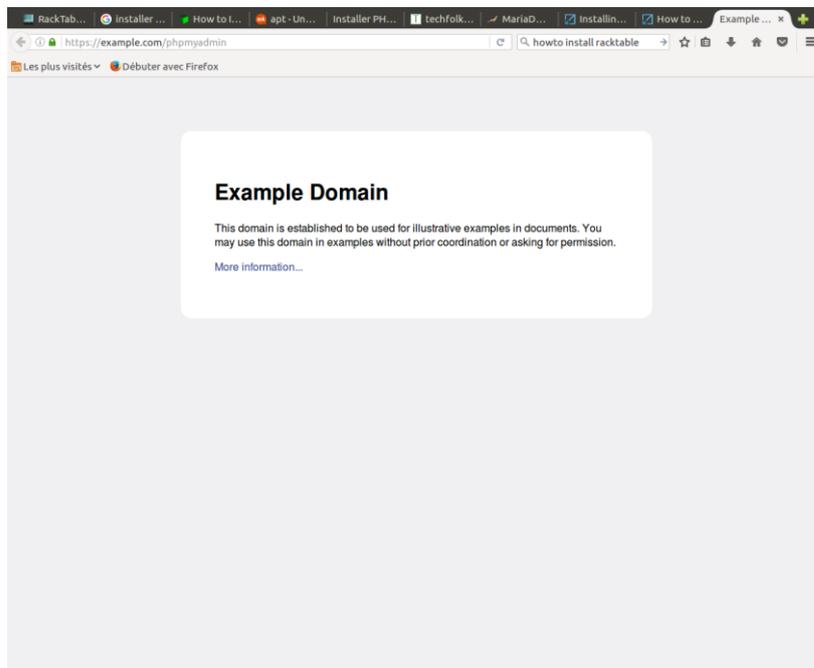
Pour tester l'apache, lancez le navigateur et tapez l'url <http://localhost> ou <http://172,31,130,38>. Vous devez également pouvoir accéder à partir des autres machines de la salle, si tout est bon. Vous devrez tomber sur cette page.

Pour  
le



tester

phpmyadmin, lancez le navigateur et tapez l'url <https://example.com/phpmyadmin>.



Pour sécuriser la base de données, tapez cette commande ci-dessous :

```
# mysql_secure_installation  
//puis il va vous demander le mot de passe du bdd qui est admin.Puis suivez les etapes
```

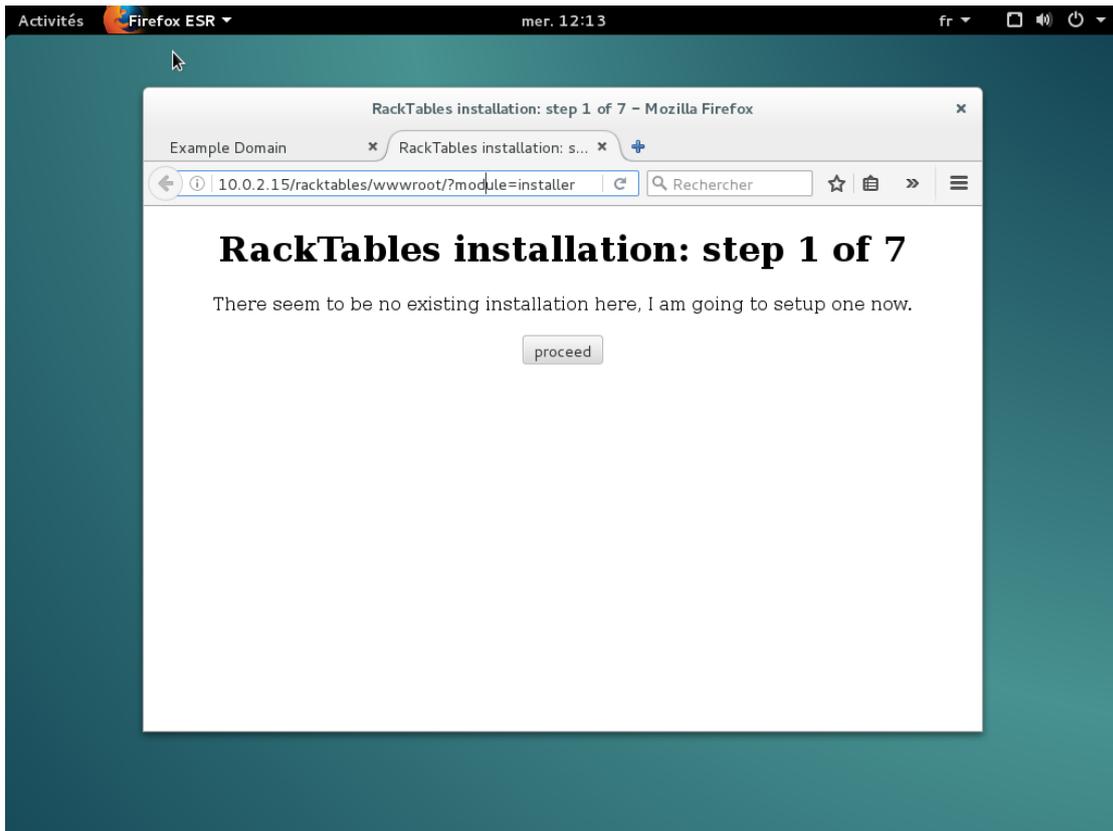
## 2. Téléchargement du Racktables Tarball

```
admin01srv@srv-Debian:~$ # wget
https://sourceforge.net/projects/racktables/files/RackTables-0.20.11.tar.gz
admin01srv@srv-Debian:~$ # tar xzvf RackTables-0.20.11.tar.gz
admin01srv@srv-Debian:~$ # sudo mkdir /var/www/html/racktables
admin01srv@srv-Debian:~$ # sudo cp -r RackTables-0.20.11/wwwroot
/var/www/html/racktables
```

### 3.Installation de Racktables en debian 8(jessie)

Lancez le navigateur et tapez l'url suivante

<http://192.168.0.29/racktables/wwwroot/?module=installer>



si

certaines modules n'existent pas, retournez dans l'invite de commande et installez-les.

Dans

l'invite de



### RackTables installation: step 2 of 7

PHP version >= 5.2.10	PASSED
PDO extension	PASSED
PDO-MySQL extension	PASSED
PCRE extension	PASSED
PCRE compiled with --enable-unicode-properties	PASSED
SNMP extension	NOT PRESENT
GD extension	NOT PRESENT
Multibyte string extension	NOT PRESENT
LDAP extension	NOT PRESENT
PCNTL extension	NOT PRESENT
JSON extension	PASSED
cURL extension	PASSED
BC Math extension	NOT PRESENT
accessed over HTTPS	No! (all your passwords will be transmitted in cleartext)

retry

commande, tapez les commandes suivantes pour telecharger les extensions:

```
# sudo apt-get install snmpd snmp
# sudo systemctl start snmpd
# sudo systemctl status snmpd //pr verifier
# sudo apt-get install php7.0-gd
# sudo apt-get install php7.0-mbstring
# sudo apt-get install php7.0-mbstring
# sudo systemctl restart apache2.service
```



### RackTables installation: step 2 of 7

PHP version >= 5.2.10	PASSED
PDO extension	PASSED
PDO-MySQL extension	PASSED
PCRE extension	PASSED
PCRE compiled with --enable-unicode-properties	PASSED
SNMP extension	NOT PRESENT
GD extension	PASSED
Multibyte string extension	PASSED
LDAP extension	NOT PRESENT
PCNTL extension	NOT PRESENT
JSON extension	PASSED
cURL extension	PASSED
BC Math extension	PASSED
accessed over HTTPS	No! (all your passwords will be transmitted in cleartext)

proceed

Faire le fichier de config inscriptible par le serveur web et désactiver le Selinux pendant l'installation :

```
# sudo touch /var/www/html/racktables/wwwroot/inc/secret.php
# sudo chmod 666 /var/www/html/racktables/wwwroot/inc/secret.php
# setenforce 0
```

#### 4. Création de base de donnée

Ouvrir le mariadb shell avec les commandes suivantes :

```
# root@srv-Debian:/# mysql -u root -p
//le mot de passe est admin
```

```
//vous etes dans la bdd
```

```
MariaDB [(none)]> CREATE DATABASE racktables_db CHARACTER SET utf8 COLLATE utf8_general_ci;
Query OK, 1 row affected (0.00 sec)
```

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON racktables_db.* TO racktables_user@localhost IDENTIFIED BY
'MY_SECRET_PASSWORD';
Query OK, 0 rows affected (0.03 sec)
```

```
MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)
```

```
//le mot de passe du table est MY_SECRET_PASSWORD.
```



## RackTables installation: step 3 of 7

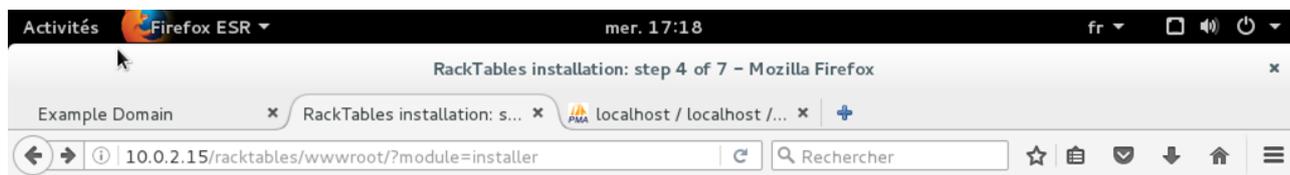
The configuration file has been written successfully.

## 5. Configuration du racktables

Dans l'invite de commande, Tapez les commandes suivantes pour créer des permissions par rapport à l'autorité du fichier secret.php :

```
# cd /var/www/html  
# ls -la  
# sudo chown -R www-data:www-data .
```

//cliquez sur retry



### **RackTables installation: step 4 of 7**

The configuration file ownership/permissions seem to be OK.

proceed

Activités Firefox ESR mer. 17:25 fr

Example Domain PDO Exception localhost / localhost / ...

10.0.2.15/racktables/wwwroot/?module=installer Rechercher

## Pdo exception: PDOException

SQLSTATE[HY000]: General error: 1419 You do not have the SUPER privilege and binary logging is enabled (you \*might\* want to use the less safe log\_bin\_trust\_function\_creators variable) (HY000)

at file /var/www/html/racktables/wwwroot/inc/dictionary.php, line 43

```

/var/www/html/racktables/wwwroot/inc/dictionary.php:43 query('CREATE TRIGGER `trigger_test` BEFORE INSERT ON `innodb_test` FOR EACH ROW BEGIN E
/var/www/html/racktables/wwwroot/inc/install.php:364 isInnoDBSupported()
/var/www/html/racktables/wwwroot/inc/install.php:66 init_database_static()
/var/www/html/racktables/wwwroot/index.php:286 renderInstallerHTML()

```

### Error info:

```

Array
(
    [0] => HY000
    [1] => 1419
    [2] => You do not have the SUPER privilege and binary logging is enabled (you *might* want to use the less safe log_bin_trust_function_crea
)

```

### Parameters:

**GET**

module installer

**POST**

step 5

**COOKIE**

//pb de privilege et de binary login

```

root@srv-Debian:/# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 65
Server version: 10.1.20-MariaDB-1~jessie mariadb.org binary distribution

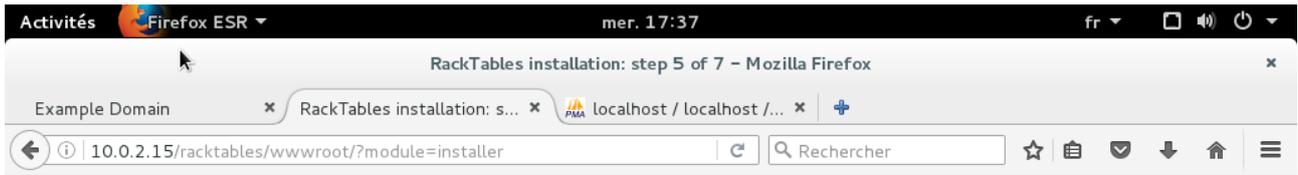
Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> set global log_bin_trust_function_creators=1;
Query OK, 0 rows affected (0.02 sec)

```

//actualiser la page



## RackTables installation: step 5 of 7

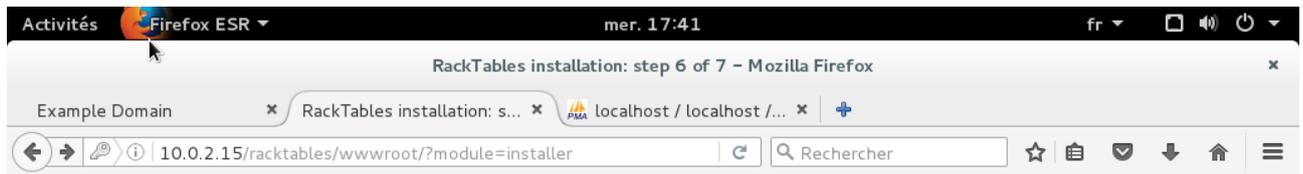
Initializing the database...

done

section	queries	errors
structure	85	0
dictbase	16	0
dictionary	92	0

proceed

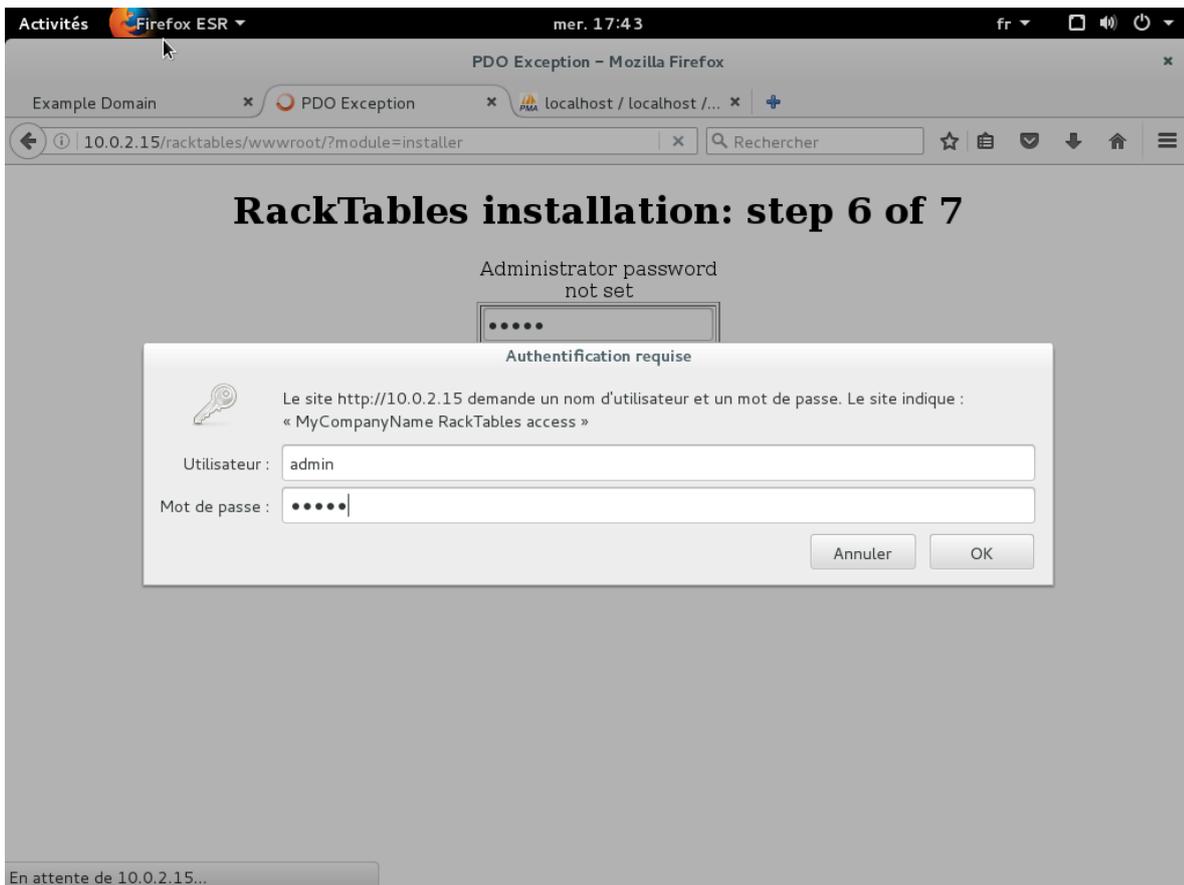
//vous devez entrer une mot de passe pour le compte admin du racktables. Et ce mot de passe va etre utiliser pr se connecter au interface web. Dans notre cas, on a choisit admin comme mot de passe.



## RackTables installation: step 6 of 7

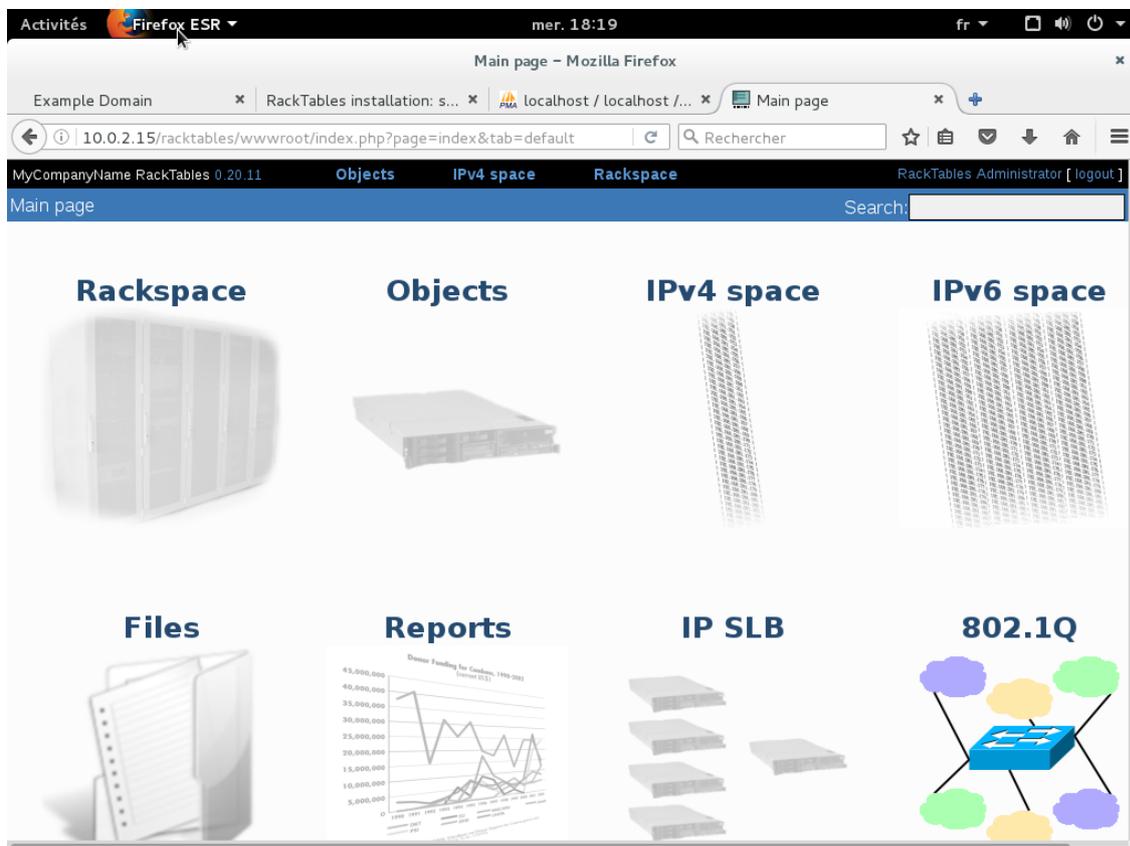
Administrator password  
not set

retry



//connecter à l'url suivante :

<http://10,0,2,15/racktables/wwwroot/index.php?page=index&tab=default>



//ajouter cette à l'invite de commandes

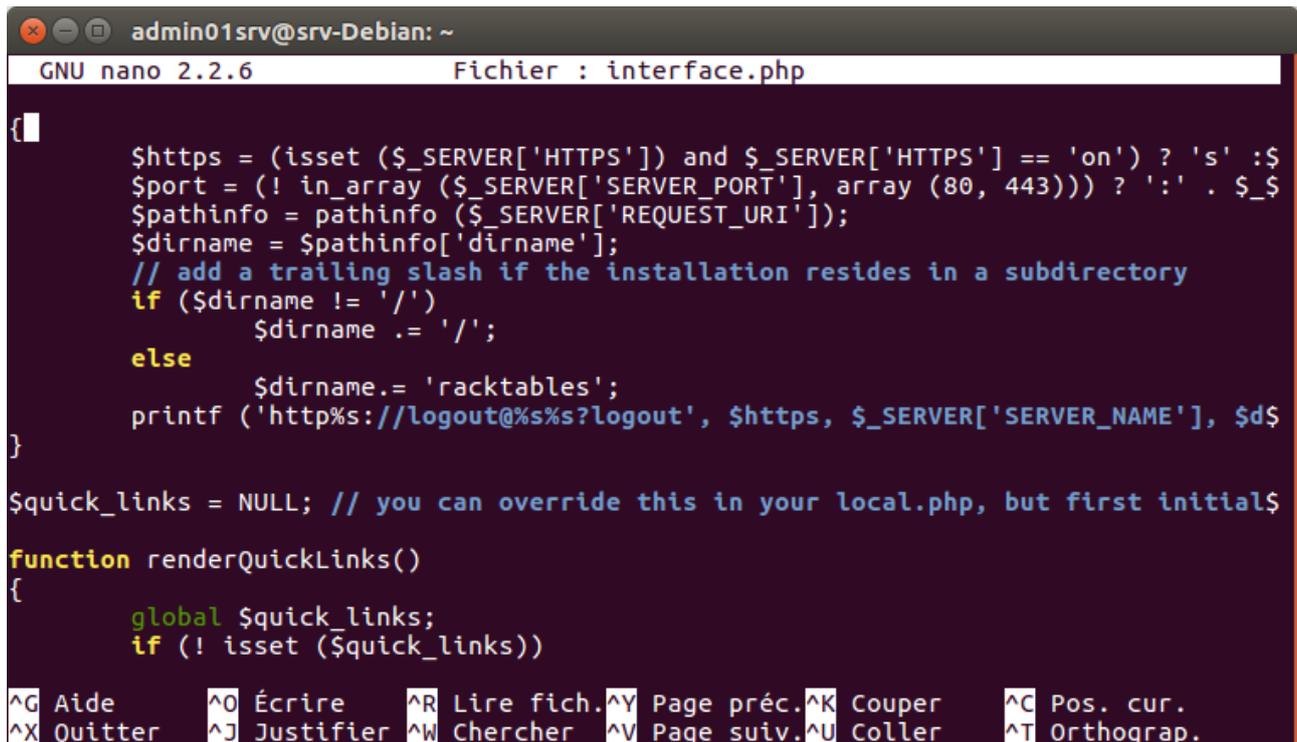
```
# sudo ln -s /var/www/html/racktables/wwwroot/index.php
/var/www/html/racktables/index.php
6.Déconnexion du session Racktables
```

```
root@srv-Debian:/home/admin01srv# cd /var/www/html/racktables/wwwroot/inc
root@srv-Debian:/var/www/html/racktables/wwwroot/inc# ls

ajax-interface.php  install.php          secret.php
auth.php            interface-cables.php slb2-interface.php
caching.php         interface-config.php slb-interface.php
code.php            interface-lib.php    slb.php
config.php          interface.php        slbv2.php
database.php        interface-reports.php snmp.php
deviceconfig.php    navigation.php       solutions.php
dictionary.php       ophandlers.php      triggers.php
exceptions.php       popup.php            upgrade.php
functions.php        pre-init.php
init.php             remote.php

root@srv-Debian:/var/www/html/racktables/wwwroot/inc# nano interface.php
```

//dans le fichier de 'interface.php' à la ligne logout, ajoutez un else



```
admin01srv@srv-Debian: ~
GNU nano 2.2.6 Fichier : interface.php
{
    $https = (isset ($_SERVER['HTTPS']) and $_SERVER['HTTPS'] == 'on') ? 's' :$
    $port = (! in_array ($_SERVER['SERVER_PORT'], array (80, 443))) ? ':' . $_$
    $pathinfo = pathinfo ($_SERVER['REQUEST_URI']);
    $dirname = $pathinfo['dirname'];
    // add a trailing slash if the installation resides in a subdirectory
    if ($dirname != '/')
        $dirname .= '/';
    else
        $dirname.= 'racktables';
    printf ('http%s://logout@%s%s?logout', $https, $_SERVER['SERVER_NAME'], $d$
}

$quick_links = NULL; // you can override this in your local.php, but first initial$

function renderQuickLinks()
{
    global $quick_links;
    if (! isset ($quick_links))

```

Il vous reste plus qu'à connecter à l'interface web du racktables avec son @ ip / son nom DNS ...  
Bonne visite...

c) Création d'un certificat SSL sous apache

```
root@srv-Debian:/# a2enmod ssl
```

```
Considering dependency setenvif for ssl:  
Module setenvif already enabled  
Considering dependency mime for ssl:  
Module mime already enabled  
Considering dependency socache_shmcb for ssl:  
Module socache_shmcb already enabled  
Module ssl already enabled
```

```
root@srv-Debian:/# systemctl restart apache2
```

```
root@srv-Debian:/# pwd
```

```
/
```

```
root@srv-Debian:/# openssl genrsa -out racktable.key 2048
```

```
Generating RSA private key, 2048 bit long modulus
```

```
.....+++
```

```
.....+++
```

```
e is 65537 (0x10001)
```

```
root@srv-Debian:/# openssl req -new -key racktable.key -out racktable.csr
```

```
-----
```

```
Country Name (2 letter code) [AU]:FR
```

```
State or Province Name (full name) [Some-State]:
```

```
Locality Name (eg, city) []:Versailles
```

```
Organization Name (eg, company) [Internet Widgits Pty Ltd]:REC
```

```
Organizational Unit Name (eg, section) []:DSI
```

```
Common Name (e.g. server FQDN or YOUR name) []:racktable.com
```

```
Email Address []:bhuva.diderot@ac-versailles.fr
```

```
Please enter the following 'extra' attributes  
to be sent with your certificate request
```

```
A challenge password []:
```

```
An optional company name []:
```

```
root@srv-Debian:/# openssl x509 -req -days 365 -in racktable.csr -signkey racktable.key -out  
racktable.crt
```

```
Signature ok
```

```
subject=/C=FR/ST=Some-
```

```
State/L=Versailles/O=REC/OU=DSI/CN=racktable.com/emailAddress=bhuva.diderot@ac-  
versailles.fr
```

```
Getting Private key
```

```
root@srv-Debian:/etc/ssl# mv racktable.crt /etc/ssl/certs/
```

```
root@srv-Debian:/etc/ssl# mv racktable.key /etc/ssl/certs/
```

```
root@srv-Debian:/etc/apache2/sites-available# ls
```

```
000-default.conf default-ssl.conf
```

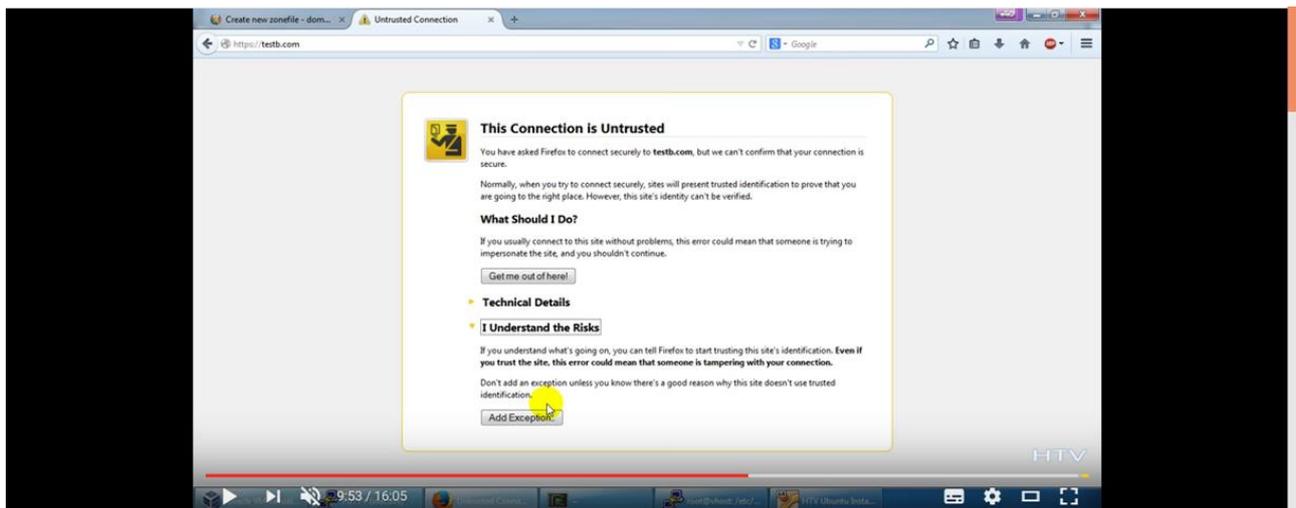
```
root@srv-Debian:/etc/apache2/sites-available# nano 000-default.conf
```

```
admin01srv@srv-Debian: ~
admin01srv@srv-Debian: /var/www/html/... x admin01srv@srv-Debian: ~ x +
GNU nano 2.2.6 Fichier : 000-default.conf Modifié
<VirtualHost *:443>
# The ServerName directive sets the request scheme, hostname and port to
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.

ServerName racktable.com
ServerAlias 10.0.2.15/racktables
SSLEngine on
SSLCertificateFile /etc/ssl/certs/racktable.crt
SSLCertificateKeyFile /etc/ssl/certs/racktable.key

^G Aide      ^O Écrire    ^R Lire fich.^Y Page préc.^K Couper    ^C Pos. cur.
^X Quitter   ^J Justifier ^W Chercher  ^V Page suiv.^U Coller    ^T Orthograp.
```

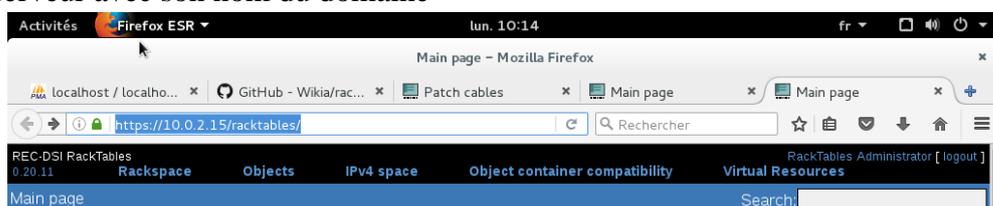
//tapez l'url suivante dans le navigateur <https://10.0.2.15/racktables>



//et puis ajoutez une exception et voilà l'interface web du racktables

//on change le nom du serveur avec son nom du domaine

//puis le document  
root pour qu'il renvoi  
directement au sous  
dossier racktable



```
admin01srv@srv-Debian: ~
admin01srv@srv-Debian: /var/w... x  admin01srv@srv-Debian: ~ x  admin01srv@SRV-Ralph: ~ x
GNU nano 2.2.6 Fichier : 000-default.conf

<VirtualHost *:443>
# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.

ServerName racktable.in.ac-versailles.fr
ServerAlias racktables
SSLEngine on
SSLCertificateFile /etc/ssl/certs/racktable.crt
SSLCertificateKeyFile /etc/ssl/certs/racktable.key

ServerAdmin webmaster@localhost
DocumentRoot /var/www/html/racktables

^G Aide      ^O Écrire    ^R Lire fich. ^Y Page préc. ^K Couper    ^C Pos. cur.
^X Quitter   ^J Justifier ^W Chercher   ^V Page suiv. ^U Coller    ^T Orthograp.
```

//quand on accède le serveur avec son nom , on ne peut pas accéder donc :

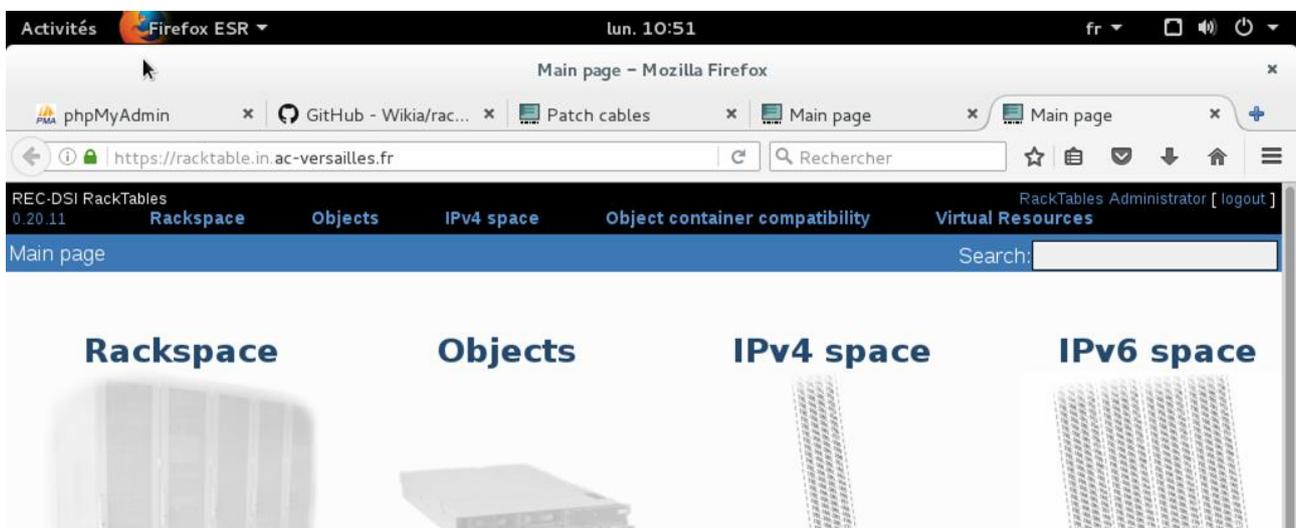
//pr ceci on ajoute une entrée au fichier hosts en précisant son nom fqdn (racktable.in.ac-versailles.fr)

```
admin01srv@srv-Debian: ~
admin01srv@srv-Debian: /var/w... x admin01srv@srv-Debian: ~ x admin01srv@SRV-Ralph: ~ x
GNU nano 2.2.6 Fichier : hosts

127.0.0.1 localhost
127.0.1.1 srv-Debian
127.0.0.1 racktable.in.ac-versailles.fr racktables
# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

[ Lecture de 7 lignes ]
^G Aide      ^O Écrire    ^R Lire fich. ^Y Page préc. ^K Couper    ^C Pos. cur.
^X Quitter   ^J Justifier ^W Chercher  ^V Page suiv. ^U Coller    ^T Orthograp.
```

Et voilà, on crée le certificat ssl pour nos sites racktables. Pour accéder à l'interface web du racktables, tapez l'url suivante : <https://racktables-server.in.ac-versailles.fr>



```
root@srv-Debian:/var/www/html# apachectl configtest
```

// ce commande sert a vérifier si le serveur web fonctionne bien

## 8) Présentation d'itop

Itop est une solution libre visant à couvrir l'ensemble des bonnes pratiques ITIL. Itop fournit une interface web permettant la gestion d'une CMDB, la gestion d'incidents, la gestion des demandes, la gestion du changement ... La documentation est l'élément le plus important dans la gestion d'un système d'information.

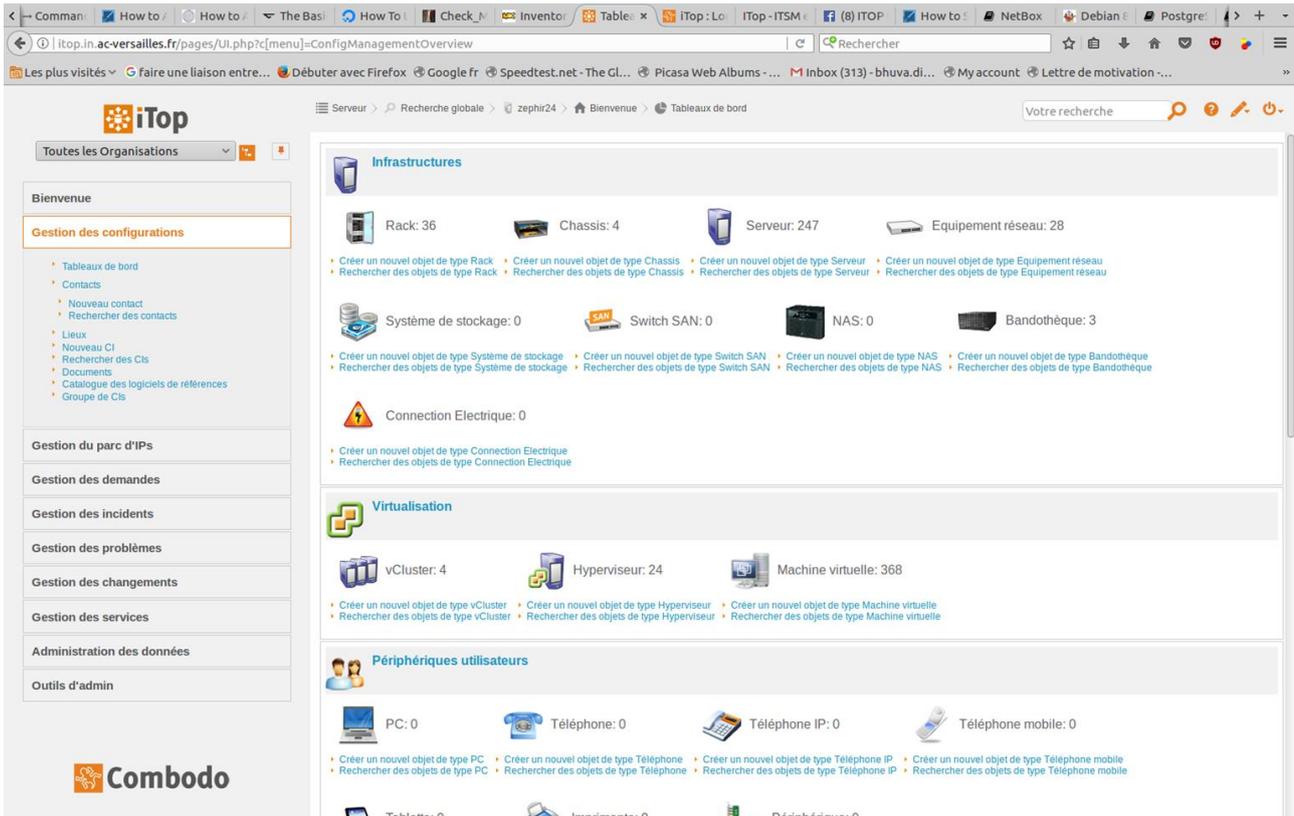
C'est grâce à une bonne documentation que l'on va pouvoir :

- assurer la continuité des services que l'on délivre aux utilisateurs finaux
- assurer un support réactif et de qualité
- gérer le patrimoine informatique du DSI

Voici une liste non-exhaustive de ses fonctionnalités :

- Un système de CMDB complet
- Un module de gestion de services IT
- Un module de gestion d'incidents
- Un module de gestion des changement
- Un module de gestion de contrats
- Une knowledge base
- Un module de coupure permettant de documenter ces coupures et de notifier les personnes concernées
- Un tableau de bord permettant une vision complète
- Outils de reporting
- Intégration facile avec des outils tierces comme par exemple Nagios, Rudder

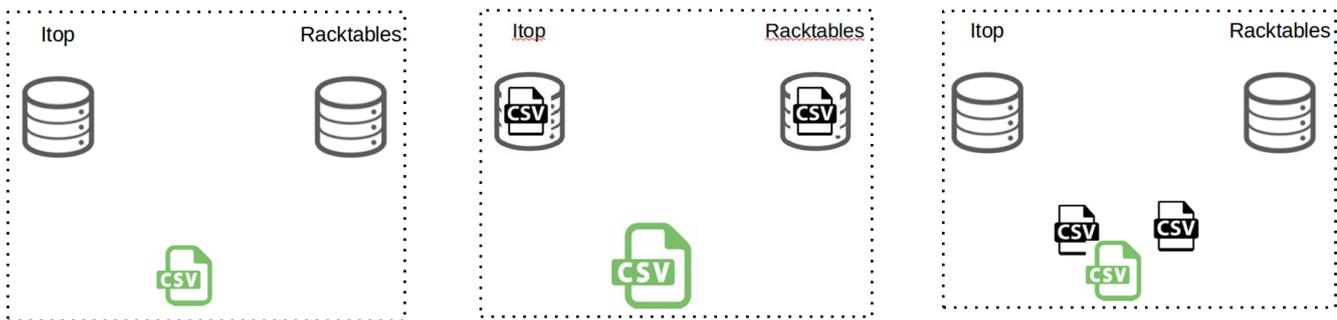
Voici le tableau de bord du Itop:

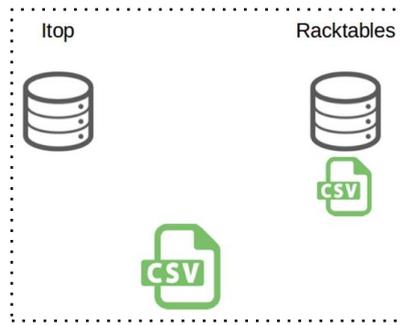


## 9) Synchronisation des serveurs entre Itop et Racktables

Dans nos infrastructures, il existe déjà un inventaire des serveurs (Itop) mais il n'est pas à jour. Et en outre dans racktables, il y a aussi des serveurs que l'on a ajoutés manuellement. Donc aucun des deux systèmes de gestion n'est à jour. Pour conclure, l'idée est de fusionner les données Itop avec celles de Racktables et d'avoir une réplication des données dans les deux systèmes de gestion.

### a) Conception de la synchronisation





## b) Script de synchronisation

Le script est codé en langage python. Le script de synchronisation des serveurs entre itop et racktables est fait en trois partie pour des raisons de syntaxe du python et pour simplifier les tâches.

1<sup>er</sup> script :

description: Le premier script permet de remettre à zéro les données et d'exporter à nouveau les données depuis le serveur Itop et Racktables.

----- Début 1<sup>er</sup> script -----

```
# -*- coding: UTF-8 -*-

import csv
import os,sys
import subprocess
import hashlib
from getpass import getpass

#chaîne_mot_de_passe = "racktables_@dmin"

#mot_de_passe_chiffre = hashlib.sha1(chaine_mot_de_passe).hexdigest()
#verrouille = True

#while verrouille:

# entre = getpass("\033[36mTapez le mot de passe :\033[0m ")
# entre = entre.encode()
# entre_chiffre = hashlib.sha1(entre).hexdigest()
# if entre_chiffre == mot_de_passe_chiffre:
#     verrouille = False
# else:
#     print("\033[31mMot de passe incorrect\033[0m")

#print("\033[32mMot de passe accepté...\033[0m")

#path="/var/lib/mysql/racktables_db/Object.csv"
#path2="/var/lib/mysql/racktables_db/server_itop.csv"
#path3="/var/lib/mysql/racktables_db/load_file.csv"
```

```
##### Suppression des fichiers s'ils existent #####
```

```
def remove():
```

```
    path="/var/lib/mysql/racktables_db/Object.csv"  
    path2="/var/lib/mysql/racktables_db/server_itop.csv"  
    path3="/var/lib/mysql/racktables_db/load_file.csv"  
    path4="/var/lib/mysql/racktables_db/server_only.csv"
```

```
    if os.path.isfile(path):  
        os.remove(path) # supprime le fichier Object.csv  
        print "\033[34mObject.csv DELETED\033[0m"
```

```
    else:  
        print "\033[34mObject.csv NOT FOUND\033[0m"
```

```
    if os.path.isfile(path2):  
        os.remove(path2) #supprime le fichier server_itop.csv  
        print "\033[34mserver_itop.csv DELETED\033[0m"
```

```
    else:  
        print "\033[34mserver_itop.csv NOT FOUND\033[0m"
```

```
    if os.path.isfile(path3):  
        os.remove(path3) #supprime le fichier load_file.csv  
        print "\033[34mload_file.csv DELETED\033[0m"
```

```
    else:  
        print "\033[34mload_file.csv NOT FOUND\033[0m"
```

```
    if os.path.isfile(path4):  
        os.remove(path4)  
        print "\033[34mFILE DELETED\033[0m"
```

```
    else:  
        print "\033[34mFILE NOT FOUND\033[0m"
```

```
#remove(path,path2)
```

```
##### Exportation des données du racktable et Itop en format csv #####
```

```
    itop_export= subprocess.Popen('mysql -h mysql-msg1.in.ac-versailles.fr -u racktables --  
password="R@cqtables" -D base_itop -e "SELECT name FROM functionalci WHERE  
finalclass=\'Server\'" > server_itop.csv',shell=True)
```

```
    #itop_export= subprocess.Popen('mysql -h \'mysql-msg1.in.ac-versailles.fr\' -u racktables --  
password="R@cqtables" -D base_itop -e "SELECT name FROM functionalci WHERE  
finalclass=\'Server\'" > server_itop.csv"',shell=True)
```

```
    print itop_export, "\033[35mItop Export success\033[0m"
```

```
    move_csv= subprocess.Popen('mv /home/administrateur/server_itop.csv  
/var/lib/mysql/racktables_db/',shell=True)
```

```
    # print move_csv, "\033[35mItop file Successfully moved\033[0m"
```

```
rack_export = subprocess.Popen('mysql -u root --password="admin" -D racktables_db -e
"SELECT * FROM Object INTO OUTFILE \'Object.csv\' FIELDS TERMINATED BY \',\' LINES
TERMINATED BY \\n\\',shell=True)
print rack_export,"\033[35mRack Export success\033[0m"
```

```
ad=subprocess.Popen('mysql -u root --password=\'admin\' -D racktables_db -e "SELECT
name FROM Object WHERE objtype_id=4" > server_only.csv',shell=True)
print ad,"\033[32mLISTE DES SERVEURS RECUPERER\033[0m"
```

```
move2= subprocess.Popen('mv /home/administrateur/server_only.csv
/var/lib/mysql/racktables_db',shell=True)
print move2,"\033[32mMOVED SUCCESSFULLY\033[0m"
```

```
print "\033[031mMAINTENANT EXECUTER LE 2eme SCRIPT!!!\033[0m"
remove()
```

----- Fin 1<sup>er</sup> script -----

2<sup>ème</sup> script :

description: Ce script permet de comparer les serveurs d'Itop avec celle du Racktables et si un serveur existe dans Itop et pas dans Racktables, alors il crée ce serveur dans Racktables. En un mot, ce script permet d'ajouter un serveur dans Racktables.

-----Début 2<sup>ème</sup> script -----

```
# -*- coding: UTF-8 -*-
```

```
import csv
import os,sys
import subprocess
import hashlib
from getpass import getpass
```

```
print "\033[31mATTENTION VEUILLEZ EXECUTER LE SCRIPT 1 AVANT D'EXECUTER CE
SCRIPT !!! \033[0m"
```

```
##### Lecture des fichiers csv #####
```

```
rack= list()
load_data= list()
itope=list()
```

```
cr=csv.reader(open("/var/lib/mysql/racktables_db/server_itop.csv","rb"))
#cr=csv.reader(open("/var/lib/mysql/racktables_db/server.csv","rb"))
```

```
itop= list()
for row in cr:
    itop.append(row) # [1])
```

```

itop.remove(itop[0])
#print itop

len= len(itop)
#print len

for i in range(0,len):
    x=itop[i][0]
    #print itop[46][0]
#    itope=list()
    itope.append(x)
#print itope

read=csv.reader(open("/var/lib/mysql/racktables_db/Object.csv","rb"))
for row in read:
#    print row[1]
    rack.append(row[1])
for i in rack:
    racke=i
#print rack

##### Fusion des données #####

rackee=['aragon0','aragon3',"aragon1","test"]
#    load_data=[]
for x in itope:
    if x not in rack:
        load_data.append(x)

print "\033[34mCes éléments vont être ajoutés dans racktables:\033[0m",load_data
#    return(load_data)

##### création d'un fichier csv pour ajouter les données #####
#####qui ont été fusionner depuis les données Itop et racktables

#print load_data

#entetes = [
#    u'id',
#    u'name',
#    u'label',
#    u'objtype_id',
#    u'asset_no',
#    u'has_problems',
#    u'comment'
#]

fichier = open('load_file.csv','wb')
#ligneEntete = ",".join(entetes) + "\n"
#fichier.write(ligneEntete)

```

```

for i in load_data:
#   dyc = [[""] + i + ['NULL', 4, 'NULL', 'NO', 'NULL']]
#   dyc = { i : ['NULL',4,'NULL','NO','NULL']}
#   print dyc

#   mywriter = csv.writer(fichier, delimiter=',', dialect='excel', lineterminator='\n')
#   mywriter.writerow(dyc[0])

#   for k,[a,b,c,d,e] in dyc.items():
#       mywriter.writerow(["k,a,b,c,d,e"])
fichier.close()

##### Déplacer le fichier vers le répertoire du bdd /var/lib/mysql/racktables_db #####

move = subprocess.Popen('cp /home/administrateur/load_file.csv
/var/lib/mysql/racktables_db',shell=True)
delete = subprocess.Popen('rm /home/administrateur/load_file.csv',shell=True)

##### Importation des données au bdd rack #####

query2='mysql -u root --password="admin" -D racktables_db -e "LOAD DATA INFILE
\'load_file.csv\' IGNORE INTO TABLE Object FIELDS TERMINATED BY \',\' ENCLOSED BY
\\\'\'\'\'
#print query2

import_rack= subprocess.Popen(query2,shell=True)
print import_rack,'\033[35mmaj succès\033[0m'

#query='touch toto.py'
#print query

```

-----Fin 2ème script -----

3ème script :

description : Ce script permet de supprimer un serveur dans Racktables si ce serveur n'existe pas dans itop.

----- Début 3ème script -----

```
# -*- coding: UTF-8 -*-
```

```
import csv
import os,sys
import subprocess
```

```
rack= list()
load_data= list()
itope=list()
```

```

racke=list()

cr=csv.reader(open("/var/lib/mysql/racktables_db/server_itop.csv","rb"))
#cr=csv.reader(open("/var/lib/mysql/racktables_db/server.csv","rb"))

itop= list()
for row in cr:
    itop.append(row) # [1])
itop.remove(itop[0])
#print itop

len1= len(itop)
#print len

for i in range(0,len1):
    x=itop[i][0]
    #print itop[46][0]
#    itope=list()
    itope.append(x)
#print itope

#ad=subprocess.Popen('mysql -u root --password=\admin\' -D racktables_db -e "SELECT name
FROM Object WHERE objtype_id=4" > server_only.csv',shell=True)
#print ad,"\033[32mLISTE DES SERVEURS RECUPERER\033[0m"
#move2= subprocess.Popen('mv /home/administrateur/server_only.csv
/var/lib/mysql/racktables_db',shell=True)
#print move2,"\033[32mMOVED SUCCESSFULLY\033[0m"

read1 = csv.reader(open("/var/lib/mysql/racktables_db/server_only.csv","rb"))
for row in read1:
#    print row[1]
    rack.append(row)
rack.remove(rack[0])
    #print rack

lenght = len(rack)
#print lenght

#print rack[2][0]

for i in range(0,lenght):
    x=rack[i][0]
    racke.append(x)
#print racke

remove_data=list()

for x in racke:
#    print x
    if x not in itope:

```

```

#         print x
         remove_data.append(x)
print "\033[34mCes éléments vont être supprimer du racktable:\033[0m",remove_data

for i in remove_data:
    x=""+i+""+""
#     print x

    exampl='mysql -u root --password=admin -D racktables_db -e "DELETE FROM Object
WHERE name='

    ter= exampl+x
#     print ter

    dele=subprocess.Popen(ter,shell=True)

#dele=subprocess.Popen('mysql -u root --password='\admin\' -D racktables_db -e "DELETE FROM
Object WHERE name='\test bhuva\'",shell=True)
    print dele,"\033[31mDELETED SUCCESSFULLY\033[0m"

```

----- Fin 3<sup>ème</sup> script -----

## 10) Synchronisation des vm entre Itop et Racktables

- a) Conception de synchronisation
- b) Script de synchronisation pour les Vm

Ce script est codé en langage python. Le script de synchronisation des vm entre itop et racktables est fait en trois partie pour des raisons de syntaxe du python et pour simplifier les tâches.

1<sup>er</sup> script :

description: Le premier script permet de remettre à zéro tous les données et d'exporter à nouveau les données depuis le serveur Itop et Racktables.

----- Début 1<sup>er</sup> script -----

```
# -*- coding: UTF-8 -*-
```

```
import csv
```

```

import os,sys
import subprocess
import hashlib
from getpass import getpass

def remove():

    path="/var/lib/mysql/racktables_db/itop_vm.csv"
    path2="/var/lib/mysql/racktables_db/rack_vm.csv"
    path3="/var/lib/mysql/racktables_db/load_vm.csv"

    if os.path.isfile(path):
        os.remove(path) # supprime le fichier Object.csv
        print "\033[34mitop_vm.csv DELETED\033[0m"
    else:
        print "\033[34mitop_vm.csv NOT FOUND\033[0m"

    if os.path.isfile(path2):
        os.remove(path2) #supprime le fichier server_itop.csv
        print "\033[34mrack_vm.csv DELETED\033[0m"
    else:
        print "\033[34mrack_vm.csv NOT FOUND\033[0m"

    if os.path.isfile(path3):
        os.remove(path3) #supprime le fichier server_itop.csv
        print "\033[34mload_vm.csv DELETED\033[0m"
    else:
        print "\033[34mload_vm.csv NOT FOUND\033[0m"

    vm= subprocess.Popen('mysql -h mysql-msg1.in.ac-versailles.fr -u racktables --
password="R@cqtables" -D base_itop -e "SELECT name FROM functionalci WHERE
finalclass=\'VirtualMachine\'" > itop_vm.csv',shell=True)
    print vm, "\033[35mVM Export success\033[0m"

    deplacer= subprocess.Popen('mv /home/administrateur/itop_vm.csv
/var/lib/mysql/racktables_db/',shell=True)

    ad=subprocess.Popen('mysql -u root --password=\'admin\' -D racktables_db -e "SELECT
name FROM Object WHERE objtype_id=1504" > rack_vm.csv',shell=True)
    print ad,"\033[32mLISTE DES Vm de racktables RECUPERER\033[0m"

    move2= subprocess.Popen('mv /home/administrateur/rack_vm.csv
/var/lib/mysql/racktables_db',shell=True)
    print move2,"\033[32mMOVED SUCCESSFULLY\033[0m"

remove()

```

----- Fin 1<sup>er</sup> script -----

2<sup>ème</sup> script :

description: Ce script permet de comparer les vm d'Itop avec celle du Racktables et si une vm existe dans Itop et pas dans Racktables, alors il crée cette virtual machine dans Racktables. En un mot, ce script permet d'ajouter des vm dans Racktables.

-----Début 2ème script -----

```
# -*- coding: UTF-8 -*-
```

```
import csv
import os,sys
import subprocess
import hashlib
from getpass import getpass
```

```
##### Lecture des fichiers csv
#####
```

```
cr=csv.reader(open("/var/lib/mysql/racktables_db/itop_vm.csv","rb"))
```

```
virtual= list()
for row in cr:
#     print row
    virtual.append(row)
virtual.remove(virtual[0])
```

```
#print virtual
```

```
len2=len(virtual)
```

```
virtuel= list()
```

```
for i in range (0,len2):
    x=virtual[i][0]
    virtuel.append(x)
#print virtuel
```

```
rack=list()
```

```
read1=csv.reader(open("/var/lib/mysql/racktables_db/Object.csv","rb"))
```

```
for row in read1:
#     print row[1]
    rack.append(row[1])
for i in rack:
    racke=i
#print rack
```

```
load2=list()
```

```
for x in virtuel:
    if x not in rack:
```

```

load2.append(x)

print "\033[34mVirtual machine added to the racktables:\033[0m",load2

fichier= open('load_vm.csv',"wb")

for i in load2:
#   dyc = ["" + i + ['NULL', 4, 'NULL', 'NO', 'NULL']]
#   dyc = { i : ['NULL',1504,'NULL','NO','NULL']}
#   print dyc

#   mywriter = csv.writer(fichier, delimiter=',', dialect='excel', lineterminator='\n')
#   mywriter.writerow(dyc[0])

#   for k,[a,b,c,d,e] in dyc.items():
#       mywriter.writerow(["k,a,b,c,d,e"])
fichier.close()

##### Déplacer le fichier vers le répertoire du bdd /var/lib/mysql/racktables_db #####

move = subprocess.Popen('cp /home/administrateur/load_vm.csv
/var/lib/mysql/racktables_db',shell=True)
delete = subprocess.Popen('rm /home/administrateur/load_vm.csv',shell=True)

##### Importation des données au bdd rack
#####

query2='mysql -u root --password="admin" -D racktables_db -e "LOAD DATA INFILE
\'load_vm.csv\' IGNORE INTO TABLE Object FIELDS TERMINATED BY \',\' ENCLOSED BY
\'\"\'"'
#print query2

import_rack= subprocess.Popen(query2,shell=True)
print import_rack,'\033[35mmaj succès\033[0m'

```

-----Fin 2ème script -----

3ème script :

descripton : Ce script permet de supprimer un vm dans Racktables si cette vm n'existe pas dans itop.

----- Début 3ème script -----

```
# -*- coding: UTF-8 -*-
```

```
import csv
import os,sys
import subprocess
```

```
##### les variables #####
```

```
vm_rm=list()
vmi=list()
rack_vm=list()
vmr=list()
remove_data=list()
```

```
##### lecture du fichier csv d'itop #####
```

```
reader= csv.reader(open("/var/lib/mysql/racktables_db/itop_vm.csv", "rb"))
```

```
for row in reader:
    vm_rm.append(row)
vm_rm.remove(vm_rm[0])
#print vm_rm
```

```
longueur=len(vm_rm)
#print longueur
```

```
for i in range(0,longueur):
    y= vm_rm [i] [0]
    vmi.append(y)
#print vmi
```

```
##### lecture du fichier csv de rack #####
```

```
reader1= csv.reader(open("/var/lib/mysql/racktables_db/rack_vm.csv", "rb"))
```

```
for row in reader1:
    rack_vm.append(row)
rack_vm.remove(rack_vm[0])
#print rack_vm
```

```
length= len(rack_vm)
#print length
```

```
for row in range(0,length):
    x= rack_vm [row] [0]
    vmr.append(x)
#print vmr
```

```
for x in vmr:
    if x not in vmi:
        remove_data.append(x)
print "\033[34mCes éléments vont être supprimer du racktable:\033[0m",remove_data
```

```

for i in remove_data:
    x=""+i+""+""
#    print x

    exampl='mysql -u root --password=admin -D racktables_db -e "DELETE FROM Object
WHERE name='

    ter= exampl+x
#    print ter

    dele=subprocess.Popen(ter,shell=True)

#dele=subprocess.Popen('mysql -u root --password=\'admin\' -D racktables_db -e "DELETE FROM
Object WHERE name=\'test bhuva\'"',shell=True)
    print "\033[31mDELETED SUCCESSFULLY\033[0m",dele

----- Fin 3ème script -----

```

## 11) Connexion LDAP

Il existe plusieurs méthodes pour reconnaître les comptes LDAP dans Racktables au lieu du compte local. La méthode qu'on va utiliser est de configurer le serveur Racktables pour communiquer avec le serveur ldap directement.

//Modifiez le fichier secret.php de façon suivante :

```

GNU nano 2.2.6                                Fichier : secret.php
#?php
# This file has been generated automatically by RackTables installer.
$pdo_dsn = 'mysql:host=localhost;dbname=racktables_db';
$db_username = 'racktables_user';
$db_password = 'MY_SECRET_PASSWORD';

# Setting MySQL client buffer size may be required to make downloading work for
# larger files, but it does not work with mysqlnd.
# $pdo_bufsize = 50 * 1024 * 1024;
# Setting PDO SSL key, cert, and CA will allow a SSL/TLS connection to the MySQL
# DB. Make sure the files are readable by the web server
# $pdo_ssl_key = '/path/to/ssl/key'
# $pdo_ssl_cert = '/path/to/ssl/cert'
# $pdo_ssl_ca = '/path/to/ssl/ca'

$user_auth_src = 'ldap';
$require_local_account = FALSE;
#$user_auth_src = 'database';
#$require_local_account = TRUE;

```

Note : à tout moment si la configuration ldap ne fonctionne pas, vous pouvez changer la méthode d'authentification en changeant : « user\_auth\_src= 'ldap' » en « user\_auth\_src='database' » et la ligne suivante en 'TRUE'

Cette étape permet de faire accepter Racktables toutes les comptes valides du LDAP.

Configurer OpenLDAP dans « search » :

```

GNU nano 2.2.6                                Fichier : secret.php
#?php
# This file has been generated automatically by RackTables installer.
$pdo_dsn = 'mysql:host=localhost;dbname=racktables_db';
$db_username = 'racktables_user';
$db_password = 'MY_SECRET_PASSWORD';

# Setting MySQL client buffer size may be required to make downloading work for
# larger files, but it does not work with mysqlnd.
# $pdo_bufsize = 50 * 1024 * 1024;
# Setting PDO SSL key, cert, and CA will allow a SSL/TLS connection to the MySQL
# DB. Make sure the files are readable by the web server
# $pdo_ssl_key = '/path/to/ssl/key'
# $pdo_ssl_cert = '/path/to/ssl/cert'
# $pdo_ssl_ca = '/path/to/ssl/ca'

$user_auth_src = 'ldap';
$require_local_account = FALSE;
#$user_auth_src = 'database';
#$require_local_account = TRUE;

# Default setting is to authenticate users locally, but it is possible to
# employ existing LDAP or Apache user accounts. Check RackTables wiki for
# more information, in particular, this page for LDAP configuration details:
# http://wiki.racktables.org/index.php?title=LDAP

$LDAP_options = array
(
    'server' => 'ldaps://ldapm.in.ac-versailles.fr',
    'search_attr' => 'uid',
    'search_dn' => 'ou=ac-versailles,ou=education,o=gouv,c=fr',
    // The following credentials will be used when searching for the user's DN:
    'group_attr' => 'personnels',
    'displayname_attrs' => 'cn',
    'options' => array (LDAP_OPT_PROTOCOL_VERSION => 3),
);

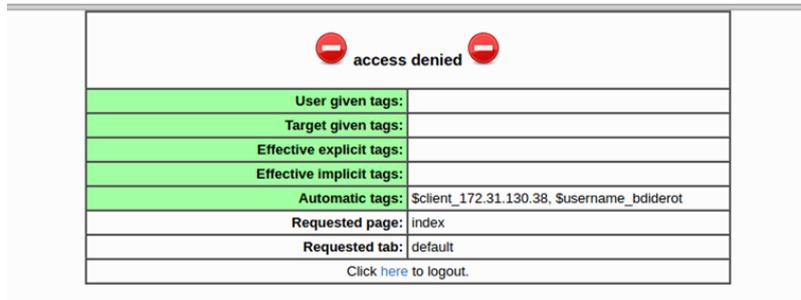
# For SAML configuration details:
# http://wiki.racktables.org/index.php?title=SAML

```

Et si  
essaye

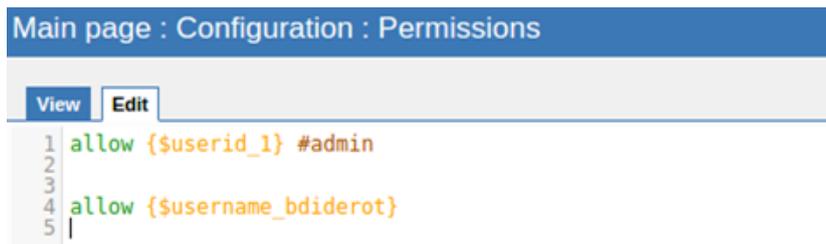
on  
de se

connecter à l'interface web avec notre ldap, il doit afficher une page en disant que on n'a pas les droits pour accéder. Contacter l'administrateur pour attribuer les droits à cette utilisateur dans Racktables. Dans cette exemple, l'identifiant est bdiderot.

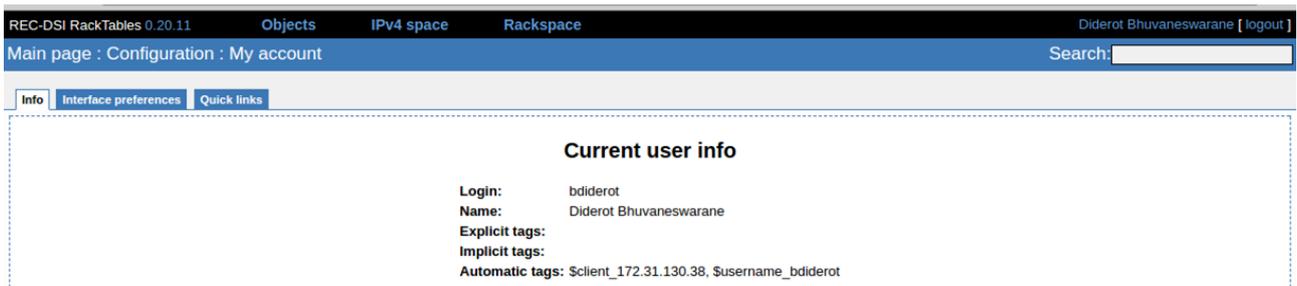


Dans racktables, cliquez sur l'onglet configuration puis dans permissions :

Ajoutez l'utilisateur bdiderot comme ci-dessous :



Maintenant, on peut accéder à l'interface web avec l'utilisateur bdiderot.



## 12) OpenManage Essentials API- synchronisation des dates de fin de garanties

### I. Présentation d'api

API signifie Application Programming Interface. Il permet d'établir des connexions entre plusieurs logiciels pour échanger des données.

Pour pouvoir accéder aux données reçues via une API, il faut bien sûr faire des requête et on a deux type de format de requête le **REST** et La **SAOP**, en réponse les formats sont soit **XML** soit du **JSON** ça dépend bien sûr de l'utilisateur. OpenManage essentials fournit une API de type REST en format xml donc, on n'est obligé de faire du parsing pour obtenir les données.

## II. Base URL

La base url d'api pour une requête REST est de format suivant:

`https://<HOST_ADDRESS>:<PORT_NUM>/api/<PRODUCT_NAMESPACE>`

Exemple:

`https://192.168.231.15:2607/api/OME.svc/`

Le reste du document a pour référence d'exemple la d'URL ci-dessus.

## III. Garantie

L'URL pour récupérer les informations d'un serveur est la suivante:

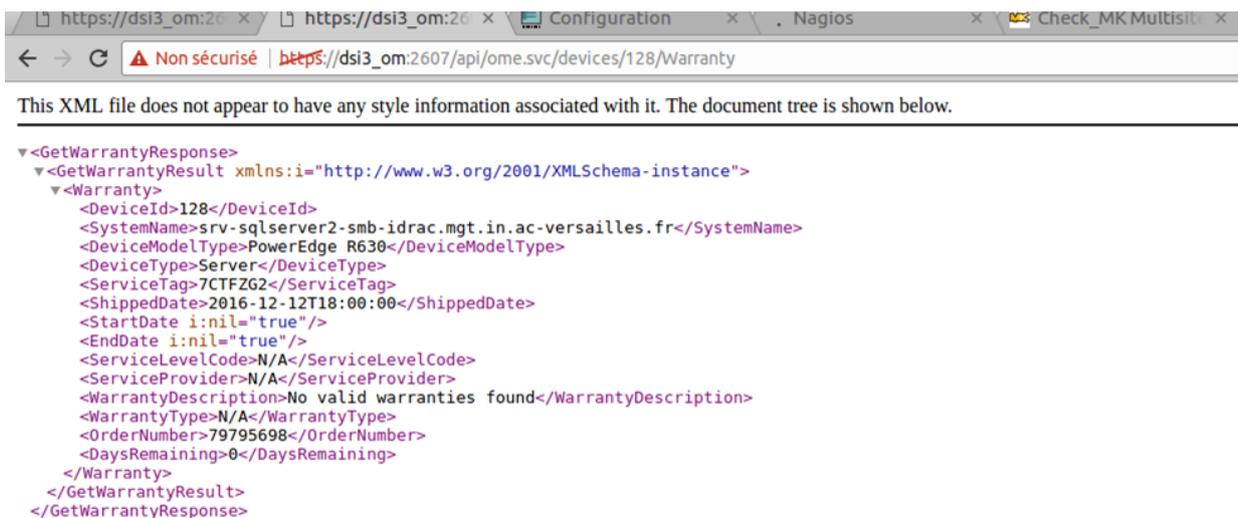
`https://dsi3_om:2607/api/ome.svc/devices/182`

note : Le numéro 182 représente l'id de l'équipement(serveur,blade-cmc, etc...). L'id de chaque équipement peut être récupérer dans l'URL suivante :

`https://dsi3_om:2607/api/ome.svc/devices`

Cette URL liste tout les équipements avec les informations de ces composants tels que l'adresse ip, service tag, adresse mac etc. ...mais dans cette page, on ne trouve pas les informations de garantie donc, on n'est obligé de trouver l'id de chaque équipement est de remplacer l'URL ci-dessous avec l'id correspondant:

`https://dsi3_om:2607/api/ome.svc/devices/128/Warranty`



## IV. Les informations de garantie

Les informations de garantie sont définies dans les tables suivantes:

Nom d'attribut	Description
----------------	-------------

Device id	L'identifiant de l'équipement
System Name	Nom de l'équipement
Device Model Type	Modèle de l'équipement
Service Tag	Service tag
End Date	<i>Date de Fin de garantie</i>
Start Date	Date de début
Shipped date	Date de livraison
Service Level Code	
Service Provider	Nom du fournisseur
Warranty Description	Description du paquet de garantie
Warranty Type	Type de garantie
OrderNumber	Numéro de commande
DaysRemainind	Reste du jour

**Note :** Attention un équipement peut avoir plusieurs paquet de garantie qui sont des extensions des autre paquets de garantie. Donc, pour retrouver la date de fin de garantie, il faut comparer les différents dates de fin de garantie et de retrouver la date la plus éloignée.

```

<?xml version="1.0"?>
- <GetWarrantyInformationResponse>
  - <GetWarrantyInformationResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
    - <Warranties>
      - <Warranty>
        <DeviceId>34</DeviceId>
        <SystemName>idrac</SystemName>
        <DeviceModelType>PowerEdge R920</DeviceModelType>
        <DeviceType>SERVER</DeviceType>
        <ServiceTag>1234567</ServiceTag>
        <ShippedDate>2009-01-05T00:00:00</ShippedDate>
        <StartDate>2009-01-05T00:00:00</StartDate>
        <EndDate>2010-01-05T23:59:59</EndDate>
        <ServiceLevelCode>ND</ServiceLevelCode>
        <ServiceProvider>QLX</ServiceProvider>
        <WarrantyDescription>Next Business Day Support</WarrantyDescription>
        <WarrantyType>INITIAL</WarrantyType>
      </Warranty>
    - <Warranty>
      <DeviceId>34</DeviceId>
      <SystemName>idrac</SystemName>
      <DeviceModelType>PowerEdge R920</DeviceModelType>
      <DeviceType>SERVER</DeviceType>
      <ServiceTag>1234567</ServiceTag>
      <ShippedDate>2009-01-05T00:00:00</ShippedDate>
      <StartDate>2010-01-06T00:00:00</StartDate>
      <EndDate>2012-01-05T23:59:59</EndDate>
      <ServiceLevelCode>NP</ServiceLevelCode>
      <ServiceProvider>DELL</ServiceProvider>
      <WarrantyDescription>Next Business Day Parts Delivery</WarrantyDescription>
      <WarrantyType>EXTENDED</WarrantyType>
    </Warranty>
  </Warranties>
</GetWarrantyInformationResult>
</GetWarrantyInformationResponse>

```

## V. Conception de synchronisation entre openManage et racktables (système de gestion de configuration)

L'idée est de récupérer la date de fin de garantie de chaque équipement est d'ajouter au logiciel racktables qui est un DCIM (outil de management du datacenter).

a) Récupération d'id de chaque équipement

Comme vu précédemment, dans l'api de openmanage Essentials, les équipements sont identifiés par son id. Et l'id de chaque équipement peut être retrouver dans l'url suivante:

[https://dsi3\\_om:2607/api/ome.svc/devices](https://dsi3_om:2607/api/ome.svc/devices)

b) Information de garantie

Une fois qu'on a récupérer l'id de chaque équipement, on peut retrouver les infos de garantie en accédant à l'url suivante en remplaçant l'id par l'id de l'équipement qu'on voudrai récupérer les infos de fin de garantie:

[https://dsi3\\_om:2607/api/ome.svc/devices/id/Warranty](https://dsi3_om:2607/api/ome.svc/devices/id/Warranty)

exemple : [https://dsi3\\_om:2607/ome.svc/devices/128/Warranty](https://dsi3_om:2607/ome.svc/devices/128/Warranty)

c) Comparaison des dates de garanties

Un équipement (plutôt un serveur) peut avoir plusieurs paquets de garantie donc cela revient à dire qu'il y a plusieurs date de fin de garantie. Donc pour retrouver la date exacte de fin de garantie, on doit prendre la date la plus éloignée entre les différents paquets.

d) Conversion en timestamp

Pour insérer les données dans racktables, il faut le convertir en format timestamp. (conception du base de donnée)

e) Liste des serveurs dans Racktables

On récupère le nom de tous les serveurs du Racktables pour afin les comparer avec les données de openmanage Essentials.

f) Insertion des données

Puis au final dans Racktables, on ajoute la date de fin de garantie au serveur correspondant.

The screenshot shows the RackTables web interface for a server named 'zephir24'. The interface includes a navigation bar with tabs for 'View', 'Properties', 'Log', 'Rackspace', 'Ports', 'IP', 'Tags', 'Files', and '802.1Q order'. The main content area is titled 'zephir24' and contains a 'summary' section and a 'ports and links' section.

**summary**

Common name: zephir24  
Object type: Server  
Asset tag is missing.  
support contract expiration: 2020-11-13  
Explicit tags: Serveur

**ports and links**

Local name	Visible label	Interface	L2 address	Remote object and port	Cable ID
kvm		KVM (host)			
eth0		1000Base-T			
eth1		1000Base-T			

## VI. Script de synchronisation

Le script de synchronisation est codé en langage python. Ce script est fait en trois parties pour des raisons de syntaxe et de sécurité:

### 1<sup>er</sup> script:

```
-----Début-----
# -*- coding: UTF-8 -*-

import os,sys
import subprocess
import csv

import requests
import xml.etree.ElementTree as ET

from requests.auth import HTTPBasicAuth
from datetime import datetime
import datetime

##### Suppression du fichier #####

def remove():

    path="/home/bdiderot/devices.xml"

    if os.path.isfile(path):
        os.remove(path) ##### supprime le fichier devices.xml
        print "\033[34mdevices.xml DELETED\033[0m"
    else:
        print "\033[34mdevices.xml NOT FOUND\033[0m"

        b=subprocess.Popen('rm devices_id_*.xml',shell=True)
        print b,"Successfully removed"
remove()

def remove2():

    path="/var/lib/mysql/racktables_db/Object_id.csv"

    if os.path.isfile(path):
        os.remove(path)
        print "\033[34mObject_id.csv DELETED\033[0m"
    else:
        print "\033[34mObject_id.csv Not found\033[0m"

remove2()

##### Récupération des données de l'api en format xml
#####

a=subprocess.Popen('wget "https://dsi3_om:2607/api/ome.svc/devices" --no-check-certificate --
user=Administrator --password=EPPve254 --output-document=devices.xml',shell=True)
print a
-----FIN 1er script-----
```

2ème script:

-----Début 2ème script-----

```
# -*- coding: UTF-8 -*-
```

```
import os,sys
import subprocess
import csv
```

```
import requests
import xml.etree.ElementTree as ET
```

```
from requests.auth import HTTPBasicAuth
```

```
from datetime import datetime
import time
import datetime
```

```
##### les variables #####
```

```
#### Le deuxième script permet de récupérer l'id de chaque élément présent dans openmanage puis avec cette id il va télécharger la page correspondant du device pour afin récupérer la date de fin de garantie(il sera faite dans le 3ème script)
```

```
id=list()
```

```
##### Lecture du fichier xml #####
```

```
global_url='https://dsi3_om:2607/api/ome.svc/devices'
```

```
#requete = requests.get(global_url, auth=HTTPBasicAuth('Administrator','EPPve254'),verify=False)
#print "Page status:",requete.status_code
```

```
tree = ET.parse('devices.xml')
root = tree.getroot()
```

```
for child in root:
    for childr in child:
        for petit in childr:
            for i in petit:
                for Id in i.iter('Id'):
                    id.append(Id.text)
                    #print Id.text
```

```
#print id
```

```
#if requete.status_code==200:
#    print 'root'
#    server_id.append(root.Id)
#else:
#    print 'Something went wrong'
```

```
##### Ajoute dans une liste tous les id qu'il récupérait avec l'api de om #####
```

```
length=len(id)
#print length
```

```
count_device=0
count_nbOfDates=0
```

```
url1='wget "https://dsi3_om:2607/api/ome.svc/devices/'
url2='/Warranty" --no-check-certificate --user=Administrator --password=EPPve254 --output-
document=/home/administrateur/script_sync_om/devices_id_'
```

```
for i in id:
```

```
    url3='_xml'
    url4=str(i)+url3
```

```
#    print url4
```

```
    count_device=count_device+1
#    print "Device no:",count_device
```

```
    warranty_url=url1+str(i)+url2+url4
#    print warranty_url
```

```
##### Requête URL #####
```

```
    v=subprocess.Popen(warranty_url,shell=True)
    print v
```

```
#print itop_warranty
```

```
requete_sub=subprocess.Popen('mysql -u root --password=admin -D racktables_db -e "SELECT
id,name INTO OUTFILE \'Object_id.csv\' FIELDS TERMINATED BY \',\' LINES TERMINATED
BY \''\n\' FROM Object WHERE objtype_id=4"',shell=True)
print requete_sub
```

-----FIN 2ème script-----

3ème script:

-----Début 3ème script -----

```
# -*- coding: UTF-8 -*-
```

```
import os,sys
import subprocess
import csv
```

```
import requests
import xml.etree.ElementTree as ET
```

```
from requests.auth import HTTPBasicAuth
```

```
from datetime import datetime
import time
import datetime
```

```
##### lire tous les fichiers du dossier
script_sync_om ki se termine _.xml
```

```
repertoire=os.listdir('/home/administrateur/script_sync_om')
#print repertoire
```

```
##### Retrouve tous les fichiers qui se terminent par _.xml et les ajoutent ds
la liste xml_file #####
```

```
xml_file=list()
```

```
for element in repertoire: #os.listdir('/home/bdiderot/test/'):
    if element.endswith('_.xml'):
        xml_file.append(element)
```

```
#print "devices_wa file",xml_file
#print len(xml_file) ##### total nmb of devices in open manage
```

```
##### Lecture des fichiers xml
#####
```

```
warranty= list()
count_device=0
```

```
for i in xml_file:
```

```
    count_device=count_device+1
```

```
#    arbre=ET.parse('devices_id_69_.xml')
    arbre=ET.parse(i)
    root=arbre.getroot()
```

```
    for child in root:
        garranty=list()
```

```

for x in child:

    # garranty=list()
    for enddate in x.iter('EndDate'):
        #garranty=list()
        for name in x.iter('SystemName'):
            # print enddate.text,name.text

            garranty.append(enddate.text)
            #print garranty,name.text

            # endDate=sorted(garranty)
            # print endDate
            # lenght=len(endDate)
            # print lenght
            #x=lenght-1
            #FinDeGarantie=endDate[x]

            #print x,name.text
##### Récupère toutes les dates de garantie d'un équipement et les
ajoutes ds la var garranty
            #print garranty,name.text

            endDate=sorted(garranty)
            #print endDate
            lenght=len(endDate)
            #print lenght
            x=lenght-1
            #print x
            #FinDeGarantie=endDate[x]

            if x>0:
                FinDeGarantie=endDate[x]
                data=FinDeGarantie,name.text
                warranty.append(data)
                #print 'test',warranty
            #else:
                #print 'not found',name.text

# print warranty
# print len(warranty)

##### création d'un fichier csv et on ajoute la liste warranty ds
ce fichier #####

# for i in warranty:
#     print 'test',i

with open("om_warranty.csv","wb") as f:
    writer= csv.writer(f)
    writer.writerows(warranty)

##### Timestamp #####

#export_itop=subprocess.Popen('mysql -h mysql-msg1.in.ac-versailles.fr -u racktables --
password="R@cqtables" -D base_itop -e "SELECT p.id, f.name, p.end_of_warranty FROM

```

```

physicaldevice AS p, functionalci AS f WHERE f.finalclass='\Server\' AND p.id=f.id" >>
itop_warranty.csv',shell=True)
#print export_itop

read_date=csv.reader(open("/home/administrateur/script_sync_om/om_warranty.csv"))

om=list()
timestamp_warranty=list()

for row in read_date:
    om.append(row[0])
#    print row
#print om ##### cette variable affiche toutes les dates
#print len(om)

##### conversion des dates en format timestamp

for timestamp in om:

    b=time.mktime(datetime.datetime.strptime(timestamp,"%Y-%m-%dT%H:%M:%S").timetuple())
    #print timestamp,'in timestamp format:',b

    timestamp_warranty.append(b)
#print timestamp_warranty
#print len(timestamp_warranty)

read_om_1=csv.reader(open("/home/administrateur/script_sync_om/om_warranty.csv"))

device_date=list(read_om_1)
#print device_date

compteur=len(device_date)
#print compteur

for i in range(compteur):
    device_date[i].append(timestamp_warranty[i])
#    timestamp_warranty[i].append(device_date[i])
#print device_date ##### la var device_data affiche: date, nom du serveur, date en timestamp
#print timestamp_warranty
#print len(device_date)

war_stamp=list()

for g in device_date:
##### Pr chaque élément de device_data, on récupère le nom du serveur et la date en timestamp
    s= g[1],g[2]
    war_stamp.append(s)
#print war_stamp ##### cette variable affiche le nom dns et sa date de fin de garantie
#print len(war_stamp)

```

```
##### Récupération des serveurs depuis racktable
#####
```

```
##### décommenter cette ligne si ce script se trouve dans la machine racktables et le fichier
de sortie se trouve /var/lib$mysql/racktables_db
```

```
#export=subprocess.Popen('mysql -h racktables-server.in.ac-versailles.fr -u root -D racktables_db -e
"SELECT name FROM Object WHERE objtype_id=4 INTO OUTFILE \'rack_list_server.csv\'
FIELDS TERMINATED BY \',\' LINES TERMINATED BY \''\n\'"',shell=True)
```

```
export=subprocess.Popen('mysql -u root --password='admin' -D racktables_db -e 'SELECT name
FROM Object WHERE objtype_id=4'>rack_list_server.csv",shell=True)
#print export,"list of server successfully exported from racktables"
```

```
rack=csv.reader(open('/home/administrateur/script_sync_om/rack_list_server.csv','rb'))
```

```
rack_list=list()
element_rack=list()
element_om=list()
results=list()
```

```
for y in rack:
    rack_list.append(y)
#print rack_list
```

```
for itemRack in rack_list:
#    print itemRack
    for item in itemRack:
#        print item
        element_rack.append(item)
element_rack.remove(element_rack[0])
#print "nom de tous les serveurs dans racktables",element_rack
#print len(element_rack)
```

```
for itemOm in war_stamp:
#    print itemOm
    element_om.append(itemOm[0])
#print "nom dns des equipemnts présents dans openmanage",element_om
#print len(element_om)
```

```
##### Pr chaque serveur de racktables cherche s'il existe ds openmange si oui, alors il récupère son
emplacement
```

```
for rack_i in range(len(element_rack)):
    for om_i in range(len(element_om)):
        if element_rack[rack_i] in element_om[om_i]:
            results.append((rack_i,om_i))
#print results,"emplacement dans la liste element_om et element_rack"
```

```
load_data=list()
```

```
for item in results:
    #print item
    data=list()
```

```

    rack_indice,om_indice= item
#    print rack_indice ##### indice ds la liste element_rack
#    print om_indice ##### indice ds la liste element_om
    i_rack=element_rack[rack_indice] ##### nom du serveur du racktables ki existe ds om
    i_om=war_stamp[om_indice] ##### sa date de fin de garantie correspondant
    data.append(i_rack)
    data.append(i_om[1])

    #print data
    load_data.append(data)
#print len(load_data)
#print "Total nb of devices in racktables where warranty info exist:",load_data

#print len(load_data)
#print "\033[31mTotal nb of devices in openmanage:\033[0m",count_device
#print "\033[31mTotal nb of devices where warranty info exist in openmanage:\033[0m",compteur
#print "\033[34mTotal nb of devices in racktables where warranty info exist\033[0m",len(load_data)

##### Insertion des données dans racktable
#####

#requete_sub=subprocess.Popen('mysql -u root --password=admin -D racktables_db -e "SELECT
id,name INTO OUTFILE \'Object_id.csv\' FIELDS TERMINATED BY \',\' LINES TERMINATED
BY \'\n\' FROM Object WHERE objtype_id=4"',shell=True)
#print requete_sub

rack_object_name=list()
rack_object_id=list()

##### récupere le nom des serveurs

##### le fichier csv Object_id.csv contient l'id et le nom de chaque serveur

lecteurForName=csv.reader(open('/var/lib/mysql/racktables_db/Object_id.csv','rb'))

for row in lecteurForName:
    #print row
    rack_object_name.append(row[1])
#print "name",rack_object_name

##### récupère l'id de tous les serveurs

lecteurForId=csv.reader(open('/var/lib/mysql/racktables_db/Object_id.csv','rb'))

for row in lecteurForId:
    rack_object_id.append(row[0])
#print "id",rack_object_id

longueur1=len(rack_object_name)
#print longueur1

##### récupère le nom des serveurs qui ont les infos de garantie

load_data1=list()

```

```

for item in load_data:
    #print item[0]
    load_data1.append(item[0])
#print load_data1

##### la variable load_data1 affiche le nom des équipements de om qui ont la
date de fin de garantie

#print "load_data1",len(load_data1)
#print "rack_object_name",len(rack_object_name)
#print "rack_object_id",len(rack_object_id)
#print "compteur",compteur

##### compare les éléments pr afin retrouver l'id des serveurs qui
vt être insérer ds racktable

testG=list()
test2=list()
countT=0

for rack_indice in range(len(rack_object_name)):
    testL=list()
    #print rack_indice
    #print rack_object_name[rack_indice]

    element=rack_object_name[rack_indice]
    ##### la variable element affiche le nom des serveurs
    #print element

##### si un élément de racktables existe dans

    if element in load_data1:
        #print "cet élément de racktables est trouvé ds om",element,rack_indice,countT
        #print element,rack_object_id[rack_indice]
        w=rack_object_id[rack_indice]
#        print w
        testL.append(element)
        testL.append(w)
#        print testL

        testG.append(testL)
        countT=countT+1
    #else:
    #    print element,"not found"

#print "testG",len(testG)

#print "listeG",sorted(testG)
conca_id=sorted(testG)
#print len(conca_id)

#print "liste load_data",sorted(load_data)
conca_timestamp=sorted(load_data)
#print len(conca_timestamp)

```

```

final_id=list()

for row in conca_id:
    #print row[1]
    final_id.append(row[1])
#print final_id
#print len(final_id)

#for row in range(len(conca_timestamp)):
compteur1=0

final_data=list()

for item in conca_timestamp:

#    print compteur1
    #print "test",final_id[compteur1]

    #print item,compteur1
    item.append(final_id[compteur1])
    #print item
    final_data.append(item)

    compteur1=compteur1+1

#print "\033[34mLa date de fin de garantie de ces serveurs vont être ajoutés:\033[0m",final_data

print "\033[31mTotal nb of devices in openmanage:\033[0m",count_device
print "\033[31mTotal nb of devices where warranty info exist in openmanage:\033[0m",len(warranty)
print "\033[34mTotal nb of devices in racktables where warranty info exist\033[0m",len(load_data)

for ligne in final_data:

    id=ligne[2]
    #print id
    name=ligne[0]
    #print name
    date=ligne[1]
    #print date

    query1='mysql -u root --password=admin -D racktables_db -e "INSERT INTO AttributeValue
(object_id, object_tid, attr_id, string_value, uint_value, float_value) VALUES ('
    #print query1

    query2=",4,21,\'NULL\',"

    query3=",\'NULL\')'"

    query4=""

    query_global=query1+id+query2+str(date)+query3+query4
    #print query_global

```

```
del_query1='mysql -u root --password=admin -D racktables_db -e "DELETE FROM
AttributeValue WHERE object_id='
```

```
del_query2=' and attr_id=21''
```

```
del_query_global=del_query1+str(id)+del_query2
#print del_query_global
```

```
##### Suppression des données ds la table attributeValue du bdd
racktable #####
delete=subprocess.Popen(del_query_global,shell=True)
```

```
##### Injection des données dans la bdd du racktables
#####
```

```
insert=subprocess.check_output(query_global,stderr=subprocess.STDOUT,shell=True)
#insert=subprocess.Popen(query_global,shell=True)
#print insert
s=not insert
#print s
```

```
if s==True:
    print "\033[32mLes infos de garantie du serveur \033[0m",name,"\033[32mont bien été
ajoutés.\033[0m"
```

```
else:
    print "\033[34mLes infos de garantie du serveur \033[0m",name,"\033[32mn'ont pas
été ajoutés.\033[0m"
```

```
#fe=subprocess.Popen('mysql -u root --password=admin -D racktables_db -e "INSERT INTO
AttributeValue (object_id, object_tid, attr_id, string_value, uint_value, float_value) VALUES
(2249,4,21,\NULL\',1499128400,\NULL\')"'',shell=True )
```

----- Fin 3ème script -----

## VII. Exécution du script et la sortie du script

L'ordre d'exécution des scripts :

- om\_rack\_sync1.py
- om\_rack\_sync2.py
- om\_rack\_sync3.py

Pour info, les scripts sont placés dans la machine racktables, dans le répertoire /home/administarteur/script\_sync\_om

La sortie du script affichera tous les serveurs dont les infos de garantie ont été ajouté dans Racktables. (voir ci-dessous)

```
root@racktables-server:/home/administrateur/script_sync_om# python om_rack_sync3.py
Total nb of devices in openmanage: 110
Total nb of devices where warranty info exist in openmanage: 95
Total nb of devices in racktables where warranty info exist 86
Les infos de garantie du serveur app-admin ont bien été ajoutés.
Les infos de garantie du serveur app-loc-3 ont bien été ajoutés.
Les infos de garantie du serveur app-nat-1 ont bien été ajoutés.
Les infos de garantie du serveur app-nat-2 ont bien été ajoutés.
Les infos de garantie du serveur app1-sconet ont bien été ajoutés.
Les infos de garantie du serveur app2-sconet ont bien été ajoutés.
Les infos de garantie du serveur app3-sconet ont bien été ajoutés.
Les infos de garantie du serveur app4-sconet ont bien été ajoutés.
Les infos de garantie du serveur aragon0 ont bien été ajoutés.
Les infos de garantie du serveur aragon2 ont bien été ajoutés.
Les infos de garantie du serveur aragon3 ont bien été ajoutés.
Les infos de garantie du serveur aragon4 ont bien été ajoutés.
Les infos de garantie du serveur aragon5 ont bien été ajoutés.
Les infos de garantie du serveur aragon6 ont bien été ajoutés.
Les infos de garantie du serveur aragon7 ont bien été ajoutés.
Les infos de garantie du serveur axm-3 ont bien été ajoutés.
Les infos de garantie du serveur bi4-aca ont bien été ajoutés.
Les infos de garantie du serveur bv10 ont bien été ajoutés.
Les infos de garantie du serveur bv9 ont bien été ajoutés.
Les infos de garantie du serveur cml1-ctl1 ont bien été ajoutés.
Les infos de garantie du serveur cml1-ctl2 ont bien été ajoutés.
Les infos de garantie du serveur db2-5 ont bien été ajoutés.
Les infos de garantie du serveur db2-6 ont bien été ajoutés.
Les infos de garantie du serveur db2-7 ont bien été ajoutés.
Les infos de garantie du serveur db2-8 ont bien été ajoutés.
Les infos de garantie du serveur demact-app-1 ont bien été ajoutés.
Les infos de garantie du serveur demact-db-1 ont bien été ajoutés.
Les infos de garantie du serveur demact-db-2 ont bien été ajoutés.
Les infos de garantie du serveur esx-blade-1.mgt.in.ac-versailles.fr ont bien été ajoutés.
Les infos de garantie du serveur esx-blade-10.mgt.in.ac-versailles.fr ont bien été ajoutés.
Les infos de garantie du serveur esx-blade-11.mgt.in.ac-versailles.fr ont bien été ajoutés.
Les infos de garantie du serveur esx-blade-12.mgt.in.ac-versailles.fr ont bien été ajoutés.
Les infos de garantie du serveur esx-blade-13.mgt.in.ac-versailles.fr ont bien été ajoutés.
Les infos de garantie du serveur esx-blade-14.mgt.in.ac-versailles.fr ont bien été ajoutés.
Les infos de garantie du serveur esx-blade-15.mgt.in.ac-versailles.fr ont bien été ajoutés.
Les infos de garantie du serveur esx-blade-16.mgt.in.ac-versailles.fr ont bien été ajoutés.
Les infos de garantie du serveur esx-blade-2.mgt.in.ac-versailles.fr ont bien été ajoutés.
Les infos de garantie du serveur esx-blade-3.mgt.in.ac-versailles.fr ont bien été ajoutés.
Les infos de garantie du serveur esx-blade-4.mgt.in.ac-versailles.fr ont bien été ajoutés.
Les infos de garantie du serveur esx-blade-5.mgt.in.ac-versailles.fr ont bien été ajoutés.
Les infos de garantie du serveur esx-blade-6.mgt.in.ac-versailles.fr ont bien été ajoutés.
Les infos de garantie du serveur esx-blade-7.mgt.in.ac-versailles.fr ont bien été ajoutés.
Les infos de garantie du serveur esx-blade-8.mgt.in.ac-versailles.fr ont bien été ajoutés.
Les infos de garantie du serveur esx-blade-9.mgt.in.ac-versailles.fr ont bien été ajoutés.
Les infos de garantie du serveur esx-sma-toip-1.mgt.in.ac-versailles.fr ont bien été ajoutés.
Les infos de garantie du serveur esx-smb-1.mgt.in.ac-versailles.fr ont bien été ajoutés.
Les infos de garantie du serveur esx-smb-2.mgt.in.ac-versailles.fr ont bien été ajoutés.
Les infos de garantie du serveur esx-smb-3.mgt.in.ac-versailles.fr ont bien été ajoutés.
Les infos de garantie du serveur esx-smb-4.mgt.in.ac-versailles.fr ont bien été ajoutés.
Les infos de garantie du serveur esx-smb-5.mgt.in.ac-versailles.fr ont bien été ajoutés.
Les infos de garantie du serveur esx-smb-6.mgt.in.ac-versailles.fr ont bien été ajoutés.
```

### VIII. Automatisation de l'exécution du script

Pour automatiser l'exécution des script, on utilise **crontab** qui est un gestionnaire des tâches devant être exécutées à un moment précis.

Pour regarder le contenu de votre **crontab**, tapez :

```
crontab -l
```

Pour éditer le fichier de votre **crontab**, tapez :

```
crontab -e
```

Dans notre cas, on planifie l'exécution du script tous les 5 minutes. Voici le fichier de crontab dont on préciser d'exécuter les scripts tous les 5 minutes. (note : utiliser les chemins absolus)

```

GNU nano 2.2.6 Fichier : /tmp/crontab.zuYHh0/crontab Modifié
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command

### Le script de synchronisation entre itop et racktables est executé toutes les 2 minutes
*/2 * * * * /home/administrateur/rack.sh >> /var/log/racktables_syncho_log/rack.log 2>&1

## le fichier de log du script de synchro est supprimé toutes les mercredis à minuit (dans la nuit du mardi à mercredi)
0 0 * * 3 rm /var/log/racktables_syncho_log/rack_del.log 2>&1

#### Le script de synchronisation entre itop et racktables pour les vm est executé toutes les 2 minutes
*/2 * * * * /home/administrateur/sync_vm.sh >> /var/log/racktables_syncho_log/sync_vm.log 2>&1

##### Le script de synchro entre openmanage essentials et racktables #####
*/1 * * * * /home/administrateur/script_sync_om/sync_om1.sh >> /var/log/racktables_syncho_log/sync_om1.log 2>&1
*/2 * * * * /home/administrateur/script_sync_om/sync_om2.sh >> /var/log/racktables_syncho_log/sync_om2.log 2>&1
*/3 * * * * /home/administrateur/script_sync_om/sync_om3.sh >> /var/log/racktables_syncho_log/sync_om3.log 2>&1

^G Aide          ^O Écrire        ^R Lire fich.    ^V Page préc.    ^K Couper        ^C Pos. cur.
^X Quitter      ^D Justifier    ^W Chercher     ^N Page suiv.    ^U Coller       ^T Orthograp.

```

## IX. Fichier de log des scripts

Pour diagnostiquer un problème dans un script, c'est toujours mieux d'avoir un fichier de log au lieu de casser la tête pour trouver le problème dans un fichier pleins de codes.

On crée un fichier de log respective pour chaque script de synchronisation, on se nomme les suivants:

- sync\_om1.log
- sync\_om2.log
- sync\_om3.log

Pour info, les fichiers de logs sont placés dans le répertoire suivant :  
/var/log/racktables\_syncho\_log

### 13) Ajout de la machine Racktables (machine client) dans la supervision

Sur notre machine Racktables:

La procédure est un peu plus longue. Il faut d'abord installer le daemon NRPE et les plugins Nagios (qui vont être lancés localement par le daemon NRPE):

Sous Ubuntu/Debian:

```
# sudo apt-get install nagios-nrpe-server
# sudo apt-get install nagios-plugins
# sudo apt-get install xinetd
#apt-get install check-mk-agent
```

Puis éditer le fichier /etc/nagios/nrpe.cfg pour modifier la ligne suivante:



```
GNU nano 2.2.6 Fichier : check_mk
#
# -----
#
# Copyright Mathias Kettner 2014 mk@mathias-kettner.de
# -----
#
# This file is part of Check_MK.
# The official homepage is at http://mathias-kettner.de/check_mk.
#
# check_mk is free software; you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by
# the Free Software Foundation in version 2. check_mk is distributed
# in the hope that it will be useful, but WITHOUT ANY WARRANTY; with-
# out even the implied warranty of MERCHANTABILITY or FITNESS FOR A
# PARTICULAR PURPOSE. See the GNU General Public License for more de-
# tails. You should have received a copy of the GNU General Public
# License along with GNU Make; see the file COPYING. If not, write
# to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor,
# Boston, MA 02110-1301 USA.
#
service check_mk
{
    type          = UNLISTED
    port          = 6556
    socket_type   = stream
    protocol      = tcp
    wait          = no
    user          = root
    server        = /usr/bin/check_mk_agent

    # If you use fully redundant monitoring and poll the client
    # from more then one monitoring servers in parallel you might
    # want to use the agent cache wrapper:
    #server        = /usr/bin/check_mk_caching_agent

    # configure the IP address(es) of your Nagios server here:
    #only from    = 127.0.0.1 10.0.20.1 10.0.20.2
    only from    = 127.0.0.1 172.31.131.135

    # Don't be too verbose. Don't log every check. This might be
    # commented out for debugging. If this option is commented out
    # the default options will be used for this service.
    #log_on_success = /var/log/check_mk_debug.log

    disable      = no
}

^G Aide          ^O Écrire        ^R Lire fich.    ^Y Page préc.    ^K Couper
^X Quitter      ^J Justifier     ^M Chercher      ^V Page suiv.    ^U Coller
                  ^C Pos. cur.
                  ^T Orthograp.
```

...  
allowed\_hosts = *Mettre ici l'adresse IP de votre serveur Nagios*  
...

// Vérifier si le service nagios-nrpe-server est en fonctionne avec la commande ci-dessous :

```

root@racktables-server:/etc/nagios# systemctl status nagios-nrpe-server.service
● nagios-nrpe-server.service - LSB: Start/Stop the Nagios remote plugin execution daemon
   Loaded: loaded (/etc/init.d/nagios-nrpe-server)
   Active: active (running) since mar. 2017-07-04 12:01:59 CEST; 1 weeks 1 days ago
   Process: 4558 ExecStop=/etc/init.d/nagios-nrpe-server stop (code=exited, status=0/SUCCESS)
   Process: 4564 ExecStart=/etc/init.d/nagios-nrpe-server start (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/nagios-nrpe-server.service
           └─4567 /usr/sbin/nrpe -c /etc/nagios/nrpe.cfg -d

```

On automatise le lancement du daemon au démarrage du serveur avec la commande:

```

root@racktables-server:/etc/nagios# chkconfig --add nagios-nrpe-server
nagios-nrpe-server    0:off 1:off 2:on 3:on 4:on 5:on 6:off

```

On ajoute une règle pour autoriser le Firewall IPtable à laisser passer les requêtes NRPE (à adapter selon vos règles):

```

root@racktables-server:/etc/xinetd.d# iptables -A INPUT -p tcp -m tcp --dport 5666 -j ACCEPT

root@racktables-server:/etc/xinetd.d# iptables -L
Chain INPUT (policy ACCEPT)
target prot opt source destination
ACCEPT tcp -- anywhere anywhere tcp dpt:nrpe

Chain FORWARD (policy ACCEPT)
target prot opt source destination

Chain OUTPUT (policy ACCEPT)
target prot opt source destination

```

// Vérifier si la machine écoute bien dans le port 5666 avec la commande ci-dessous :

```

root@racktables-server:/etc/xinetd.d# netstat -tulpn
Connexions Internet actives (seulement serveurs)
Proto Recv-Q Send-Q Adresse locale Adresse distante Etat PID/Program name
tcp 0 0 0.0.0.0:22 0.0.0.0:* LISTEN 6551/sshd
tcp 0 0 127.0.0.1:25 0.0.0.0:* LISTEN 15119/exim4
tcp 0 0 0.0.0.0:6556 0.0.0.0:* LISTEN 13149/xinetd
tcp 0 0 0.0.0.0:5666 0.0.0.0:* LISTEN 13341/nrpe
tcp 0 0 0.0.0.0:111 0.0.0.0:* LISTEN 1808/rpcbind
tcp6 0 0 :::22 :::* LISTEN 6551/sshd
tcp6 0 0 :::1:25 :::* LISTEN 15119/exim4
tcp6 0 0 :::443 :::* LISTEN 4648/apache2
tcp6 0 0 :::5666 :::* LISTEN 13341/nrpe

```

Etat de la machine racktables-server dans la supervision avant les manipes:

The screenshot shows the Check\_MK web interface for the host 'racktables-server'. The main table displays the following services and their statuses:

State	Service	Icons	Status detail	Age	Checked	Perf-O-Meter
CRIT	Check_MK		CRIT - Cannot get data from TCP port 172.31.131.37:6556: [Errno 111] Connection refused, execution time 0.0 sec	17 sec	17 sec	0.0 s
WARN	Check_MK inventory		WARN - 2 unchecked services (ps:2)	2017-05-10 14:26:47	19 min	
OK	[SYS] CPU utilise		OK - user: 0.3%, system: 0.1%, wait: 0.0%, total: 0.5%	14 min	5 min	0%
OK	[SYS] FS /		OK - 11.1% used (6.29 of 56.49 GB), (levels at 80.00/90.00%), trend: +8.30 kB / 24 hours, inodes available 3687k/97.63%	14 min	5 min	11.14 %
UNKN	[SYS] FS /boot		UNKNOWN - filesystem not found	14 min	5 min	
UNKN	[SYS] FS /home		UNKNOWN - filesystem not found	14 min	5 min	
UNKN	[SYS] FS /opt		UNKNOWN - filesystem not found	14 min	5 min	
UNKN	[SYS] FS /utils		UNKNOWN - filesystem not found	14 min	5 min	
UNKN	[SYS] FS /tmp		UNKNOWN - filesystem not found	14 min	5 min	
UNKN	[SYS] FS /usr		UNKNOWN - filesystem not found	14 min	5 min	
UNKN	[SYS] FS /var		UNKNOWN - filesystem not found	14 min	5 min	
OK	[SYS] IO Disque SUMMARY		OK - 575.74 B/sec read, 3.58 kB/sec write, IOs: 0.42/sec	9 min	5 min	0.00 M/s 0.00 M/s
OK	[SYS] Kernel Context Switches		OK - 84/s	9 min	5 min	83.6/s
OK	[SYS] Kernel Major Page Faults		OK - 0/s	9 min	5 min	0.0/s
OK	[SYS] Kernel Process Creations		OK - 1/s	9 min	5 min	0.6/s
OK	[SYS] Load		OK - 15min load 0.05	14 min	5 min	0.0
OK	[SYS] Memoire utilisee		OK - 0.41 GB used (0.40 RAM + 0.00 SWAP + 0.01 Pagetables, this is 20.7% of 1.96 RAM (2.47 total SWAP)), 0.0 mapped, 0.8 committed, 0.0 shared	14 min	5 min	20%
OK	[SYS] Nombre de Threads		OK - 107 threads	14 min	5 min	107
OK	[SYS] Process RSYSLOG		OK - 1 processes 252.6 MB virtual, 3.7 MB resident, 0.0% CPU	14 min	5 min	0.0%
OK	[SYS] Process SYSLOG		OK - 1 processes 252.6 MB virtual, 3.7 MB resident, 0.0% CPU	14 min	5 min	0.0%

Maintenant, lancez le service nagios-nrpe avec la commande suivante:

```
systemctl restart nagios-nrpe-server.service
systemctl restart xinetd.service
```

Dans certains distributions, il faut activer le script de lancement de xinetd. Sinon le service xinetd ne sera pas lancé automatiquement lors du prochain reboot.

On RedHat, SUSE and FEDORA this is done by:

```
root@racktables-server:~# chkconfig xinetd on
```

```
root@racktables-server:~# netstat -ltn
Connexions Internet actives (seulement serveurs)
Proto Recv-Q Send-Q Adresse locale      Adresse distante    Etat
tcp      0      0 0.0.0.0:22          0.0.0.0:*           LISTEN
tcp      0      0 127.0.0.1:25       0.0.0.0:*           LISTEN
```

Lorsqu'on essaye de se connecter à la machine client avec le check-mk, la sortie depuis l'agent sur le port 6556 est vide :

```
bdiderot@REC160083:~$ sudo check_mk -d 172.31.131.37
Problem contacting agent: Empty output from agent at TCP port 6556
```

La machine client écoute bien sur le port 6556... Ceci est vérifié avec la commande netstat -an | grep 6556

- *Vérification de check\_mk en local*

1/- ///On checke en local si le check-mk sort bien les infos ... Pour ceci, on lance le client directement dans la machine hôte et vérifiez si on récupère les données...

```
root@racktables-server:/usr/bin# check_mk_agent
```

2/- lancer le client en local avec localhost :

```
root@racktables-server:/# telnet localhost 6556
```

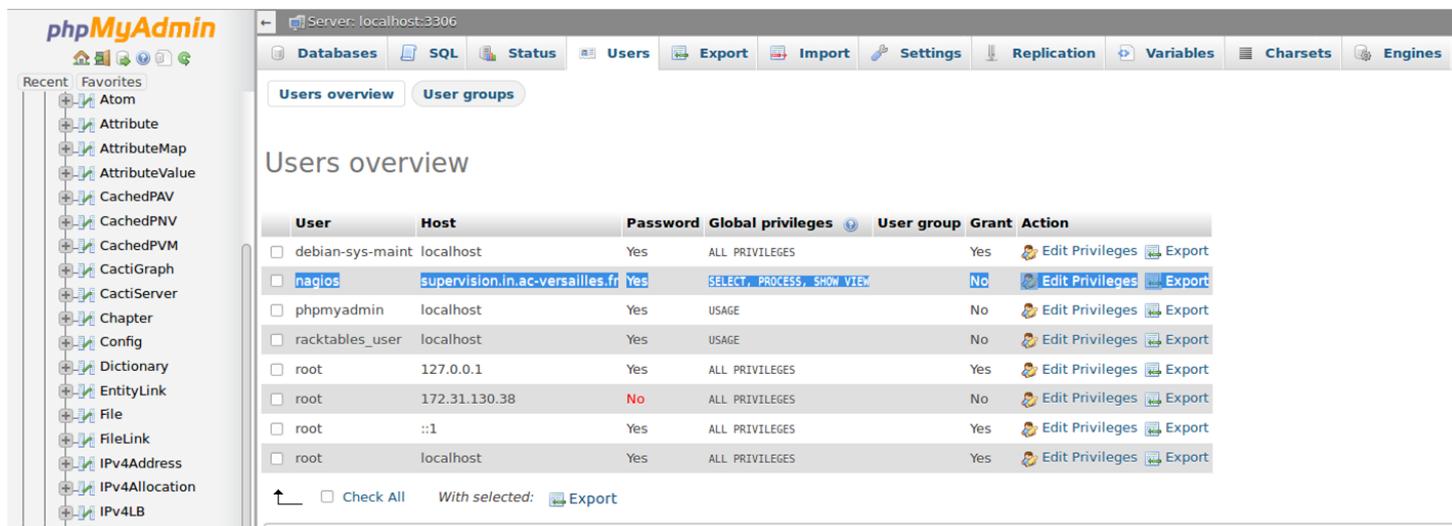
3/- lancer le client en local avec son fqdn :

```
root@racktables-server:/# telnet racktables-server 6556
```

- On autorise le serveur supervision à accéder au machine client pour récupérer les données. Pour ceci, il faut ajouter le serveur dans le fichier /etc/hosts.allow du machine client :

// créer l'utilisateur nagios ds le mysql pour qu'il accède au bdd et attribuer tous les droits

*nagios'@'supervision.in.ac-versailles.fr'*



Voici le fichier /etc/hosts.allow ci-dessous :

```
# /etc/hosts.allow: list of hosts that are allowed to access the system.
# See the manual pages hosts_access(5) and hosts_options(5).
#
# Example:  ALL: LOCAL @some_netgroup
#          ALL: .foobar.edu EXCEPT terminalserver.foobar.edu
#
# If you're going to protect the portmapper use the name "rpcbind" for the
# daemon name. See rpcbind(8) and rpc.mountd(8) for further information.
#
ALL:172.31.131.135
##@ ip du machine supervision
```

Etat de la machine racktables-server dans la supervision après les manipules :

The screenshot shows the Nagios Check\_MK interface for the host 'racktables-server'. The top navigation bar includes 'Services of Host racktables-server', '41 rows', 'admin (admin)', and '10:56'. The left sidebar contains a 'Tactical Overview' showing 638 hosts and 11993 services, along with a 'Quicksearch' bar and a 'Views' menu. The main content area displays a table of services for 'racktables-server'.

State	Service	Icons	Status detail	Age	Checked	Perf-O-Meter
OK	Check_MK		OK - Agent version 1.2.6p12, execution time 0.2 sec	18 min	10 min	0.2 s
OK	Check_MK inventory		OK - no unchecked services found	13 min	8 min	
OK	[APACHE] Check active_threads		APACHE OK - active_threads 2	8 min	8 min	
OK	[APACHE] Check cpu_load		APACHE OK - cpu_load 0.001%	7 min	7 min	
OK	[APACHE] Check cpu_usage_user_children		APACHE OK - cpu_usage_user_children 0.00000%	5 min	3 min	
OK	[APACHE] Check idle_threads		APACHE OK - idle_threads 4	2 min	2 min	
PEND	[APACHE] Check requests_per_second			-	-	
OK	[APACHE] Check slots_open		APACHE OK - slots_open 145	10 min	6 min	
OK	[APACHE] Check slots_total		APACHE OK - slots_total 150	8 min	8 min	
OK	[APACHE] Check threads_closing_connection		APACHE OK - threads_closing_connection 0	8 min	8 min	
OK	[APACHE] Check threads_gracefully_finishing		APACHE OK - threads_gracefully_finishing 0	7 min	7 min	
OK	[APACHE] Check threads_idly_cleaning		APACHE OK - threads_idly_cleaning 0	5 min	3 min	
OK	[APACHE] Check threads_keepalive		APACHE OK - threads_keepalive 0	2 min	2 min	
PEND	[APACHE] Check threads_starting_up			-	-	
OK	[APACHE] Check threads_waiting_for_connection		APACHE OK - threads_waiting_for_connection 3	10 min	6 min	
OK	[APACHE] Check url		HTTP OK: HTTP/1.1 200 OK - 6481 octets en 0,003 secondes de temps de réponse	8 min	8 min	
PEND	[MYSQL] connection-time			-	-	
PEND	[MYSQL] connects-aborted			-	-	
OK	[MYSQL] log-waits		OK - 0 innodb log waits in 122 seconds (0.0000/sec)	5 min	3 min	
OK	[MYSQL] long-running-procs		OK - 0 long running processes	2 min	2 min	
PEND	[MYSQL] open-files			-	-	
OK	[MYSQL] process		Uptime: 1286591 Threads: 1 Questions: 813071 Slow queries: 0 Opens: 1177 Flush tables: 1 Open tables: 199 Queries per second avg: 0.631	8 min	8 min	

// créer l'utilisateur nagios ds le mysql pour qu'il accède au bdd et attribuer tous les droits  
*nagios'*@'supervision.in.ac-versailles.fr'

The screenshot shows the phpMyAdmin interface for 'Server: localhost:3306'. The 'Users overview' page displays a table of database users and their privileges.

User	Host	Password	Global privileges	User group	Grant	Action
<input type="checkbox"/>	debian-sys-maint	localhost	Yes	ALL PRIVILEGES	Yes	
<input type="checkbox"/>	nagios	supervision.in.ac-versailles.fr	Yes	SELECT, PROCESS, SHOW VIEW	No	
<input type="checkbox"/>	phpmyadmin	localhost	Yes	USAGE	No	
<input type="checkbox"/>	racktables_user	localhost	Yes	USAGE	No	
<input type="checkbox"/>	root	127.0.0.1	Yes	ALL PRIVILEGES	Yes	
<input type="checkbox"/>	root	172.31.130.38	No	ALL PRIVILEGES	No	
<input type="checkbox"/>	root	:::1	Yes	ALL PRIVILEGES	Yes	
<input type="checkbox"/>	root	localhost	Yes	ALL PRIVILEGES	Yes	

Les commandes nécessaires :

- cmk -II : permet d'inventorier les hôtes et de les configurer automatiquement
- cmk -o : ceci va créer le fichier de configuration nagios pour les hôtes et les services

option : transfert de fichier d'une machine à une autre avec la connexion ssh

```
scp check-mk-raw-1.2.8p24_0.jessie_amd64.deb root@racktables-server:/home/administrateur
```

## **Guide d'utilisation Racktables**

### **Sommaire :**

- I. Ajouter un serveur**
- II. Ajouter une machine virtuelle**
- III. Gestion des adresses sous-réseaux**
- IV. Assigner une adresse ip à un objet**
- V. Gestion des baies**
- VI. Emplacement d'un objet dans une baie**
- VII. Gestion d'un blade et des lames**
- VIII. Gestion d'un hyperviseur**
- IX. Rapport du datacenter avec système d'étiquetage**
- X. Date de fin de garantie**
- XI. Création et gestion d'un switch**
- XII. Lier un port du switch à un objet**
- XIII. Authentification LDAP**

## I. Ajouter un serveur

Note: Les serveurs ne peuvent pas être créés directement dans Racktables. Les serveurs présents dans Racktables sont synchronisés avec la liste des serveurs d'Itop (système de gestion).

Pour créer un serveur, il faut d'abord créer ce serveur dans Itop comme ci-dessous :

The screenshot shows the iTop web interface for creating a new server object. The browser address bar indicates the URL: `itop.in.ac-versailles.fr/pages/UI.php?operation=new&class=Server`. The page title is "Création d'un objet de type Serveur".

The interface includes a sidebar on the left with navigation options such as "Bienvenue", "Mes raccourcis", "Gestion des configurations", "Gestion du parc d'IPs", "Gestion des demandes", "Gestion des incidents", "Gestion des problèmes", "Gestion des changements", "Gestion des services", "Administration des données", and "Outils d'admin".

The main content area is titled "Création d'un objet de type Serveur" and features a top navigation bar with tabs: "Propriétés", "Logiciels", "Contacts", "Documents", "Solutions applicatives", "Interfaces réseaux", "FC ports", "Equipements réseaux", "SANs", and "Volumes logiques". Below this, there are sub-tabs for "Contrats fournisseur" and "Tickets en cours".

The form is divided into several sections:

- Informations générales:** Includes fields for "Nom" (test), "Organisation" (INFRA), "Statut" (production), "Criticité" (basse), "Site", "Rack", and "Chassis".
- Dates:** Includes fields for "Date de mise en production", "Date d'achat", and "Date de fin de garantie".
- Alimentation électrique:** Includes fields for "Source électrique A" and "Source électrique B", and a "Redondance" section with a note: "Le Serveur est alimenté si au moins une source électrique (A ou B) est opérationnelle".
- Informations complémentaires:** Includes fields for "Marque", "Modèle", "Famille OS", "Version OS", "IP de gestion", "Licence OS", "CPU", "RAM", "NB Unité", "Numéro de série", and "Numéro Asset".
- Autres informations:** Includes a "Description" text area.

The form has "Annuler" and "Créer" buttons at the top and bottom.

Une fois que le serveur a été bien créé dans Itop, dans le prochain cinq minutes, ce serveur va être créé automatiquement dans racktables. Vous pouvez le voir dans la rubrique « Objects » de l'interface de racktables :

The screenshot shows the RackTables web interface. The main content area displays a table titled "Objects (991)". The table has the following columns: Common name, Type, Visible label, Asset tag, and Row/Rack or Container. The table lists various objects, including network chassis, VMs, network switches, and servers. A sidebar on the right titled "Tag filters" shows a list of filters with checkboxes, such as "Blade (4)", "Lame (51)", "KVM", "Console Kvm (9)", "Switch Kvm (8)", "Routeur (2)", "Serveur (157)", "Switch (8)", "Switch 2960 (10)", "Switch du management SMB (1)", "Switch nexus 2000 (10)", "Switch nexus 5000 (1)", "Tape Library (2)", "Virtual Switch (3)", "Vm (337)", and "VTL (1)".

Common name	Type	Visible label	Asset tag	Row/Rack or Container
0025WG79 WAN-ONS-1	Network chassis		Allée 3/Baie 20	
1XATTD NEW VM	VM		esx-smb-toip-1.mgt.in.ac-versailles.fr	
1XATTD NEW VM	VM		Unmounted	
3750	Network switch	SWITCH	SMA-BA-3	Coeur du réseau/Baie A
5 view svc code:39105854047	Server		Allée 3/Baie 22	
5 vieix svc code:39114580015	Server		Allée 3/Baie 22	
78858 KAZC VERS DPM06	Network security		Allée 3/Baie 20	
aca-pic VM	VM		esx-blade-10.mgt.in.ac-versailles.fr	
aca-pic VM	VM		Unmounted	
ACM1A VM	VM		esx-sma-toip-1.mgt.in.ac-versailles.fr	
ACM1A VM	VM		Unmounted	
ACM1B VM	VM		Unmounted	
ACM1B VM	VM		esx-smb-toip-1.mgt.in.ac-versailles.fr	
adhara VM	VM		esx-blade-9.mgt.in.ac-versailles.fr	
adhara VM	VM		Unmounted	
admin-fim VM	VM		Unmounted	
admin-fim VM	VM		Unmounted	
admin-jb-1 VM	VM		esx-blade-2.mgt.in.ac-versailles.fr	
admin-jb-1 VM	VM		Unmounted	
admin-wt12 VM	VM		esx-blade-15.mgt.in.ac-versailles.fr	
admin-wt12 VM	VM		Unmounted	
admin-wt12-2 VM	VM		esx-blade-12.mgt.in.ac-versailles.fr	

Voici la liste de tous les objets (serveur,vm,switch,routeur etc ...) du racktables :

Dans ce liste, vous devez trouver le serveur 'test' qu'on vient de créer dans itop. Si le serveur n'est pas encore créé dans Racktables, cela veut dire qu'il existe déjà un serveur avec ce nom. En considérant que le serveur a bien été créé on pouvez remarqué que le status du serveur est « Unmounted », cela s'explique que le serveur n'a pas encore été monter dans le baie.

tablettes-amon-78-2 Vm	VM	esx-blade-8.mgt.in.ac-versailles.fr
tablettes-amon-95-1	VM	Unmounted
tablettes-amon-95-1 Vm	VM	esx-blade-5.mgt.in.ac-versailles.fr
tablettes-amon-95-2	VM	Unmounted
tablettes-amon-95-2 Vm	VM	esx-blade-1.mgt.in.ac-versailles.fr
tablettes-amon-pere Vm	VM	esx-blade-13.mgt.in.ac-versailles.fr
tablettes-amon-pere	VM	Unmounted
tablettes-scribe	VM	Unmounted
tablettes-scribe Vm	VM	esx-blade-11.mgt.in.ac-versailles.fr
tablettes-syslog	VM	Unmounted
tablettes-syslog Vm	VM	esx-blade-16.mgt.in.ac-versailles.fr
taxation Vm	VM	esx-blade-8.mgt.in.ac-versailles.fr
taxation	VM	Unmounted
test	Server	Unmounted
Test_Centreon Vm	VM	esx-blade-8.mgt.in.ac-versailles.fr
Test_Centreon	VM	Unmounted
test_supervision Vm	VM	esx-blade-3.mgt.in.ac-versailles.fr

Cliquez sur le nom du serveur...

The screenshot shows the Nagios RackTables web interface. The browser address bar displays the URL: [https://racktables-server/index.php?page=object&tab=default&object\\_id=2807](https://racktables-server/index.php?page=object&tab=default&object_id=2807). The page title is "test". The interface includes a navigation menu with options: View, Properties, Log, Rackspace, Ports, IP, AutoPorts, Tags, Files, and 802.1Q order. The main content area is divided into two sections: "summary" and "rackspace allocation". The "summary" section contains the following information:

- Common name: test
- Object type: Server
- Asset tag is missing.

Pour personnaliser (étiquette,nom,date de garantie,hardware...) le serveur, cliquez sur l'onglet «Properties» :

The screenshot shows the 'Attributes' form in the RackTables web interface. The form is titled 'Attributes' and contains the following fields and options:

- Type: Server (dropdown menu)
- Common name: test (text input)
- Visible label: (text input)
- Asset tag: (text input)
- Tags: new tags here... (text input with a plus icon)
- Select container: Select a container (button)
- contact person: (text input)
- FQDN: (text input)
- HW type: -- NOT SET -- (dropdown menu)
- HW warranty expiration (YYYY-MM-DD): (text input)
- Hypervisor: -- NOT SET -- (dropdown menu)
- OEM S/N 1: (text input)
- Slot number: (text input)
- support contract expiration (YYYY-MM-DD): (text input)
- SW type: -- NOT SET -- (dropdown menu)
- SW warranty expiration (YYYY-MM-DD): (text input)
- UUID: (text input)
- Has problems:
- Actions:
- Comment:

A yellow highlight is visible on the 'HW type' dropdown menu, which is currently set to '-- NOT SET --'.

Dans cette fenêtre, on peut personnaliser le serveur.

Par exemple pour cet objet, on va attribué les informations ci-dessous :

tag : Serveur  
 FQDN : test.in.ac-versailles.fr  
 HW type : Dell PowerEdge R730  
 Hypervisor : NO  
 Software : Vmware ESXI 6.0  
 etc...

N'oublier d'enregistrer les modifications en cliquant sur le



bouton.

### Attributes

Type:

Common name:

Visible label:

Asset tag:

Tags:

Tags:

Select container: [Select a container](#)

contact person:

FQDN:

HW type:

HW warranty expiration (YYYY-MM-DD):

Hypervisor:

OEM S/N 1:

Slot number:

support contract expiration (YYYY-MM-DD):

SW type:

SW warranty expiration (YYYY-MM-DD):

UUID:

Has problems:

Actions:

Comment:

## II- Ajouter une machine virtuelle

### Note:

Les vm ne peuvent pas être créés directement dans Racktables. Les vm du Racktables sont synchronisés avec les vm d'Itop qui est lui-même synchronisé avec ESXI. Donc, on ne peut pas concrètement créer des machines virtuelles dans Racktables. Lorsqu'on crée une Vm dans Esxi, il va automatiquement être créé dans Itop et Racktables...

Et pour personnaliser une Vm, c'est la même méthode du serveur qui s'applique. (voir ci-dessus)

Comme les machines virtuelles n'ont pas une substance physique, on ne peut pas les placer dans une baie, on va les placer dans un hyperviseur. (cf. VIII- Gestion d'un hyperviseur)

## III- Gestion des adresses sous-réseaux

Pour ajouter une adresse réseau, cliquez « Add »

REC-DSI RackTables 0.20.11 Rackspace Objects IPv4 space 802.1Q Object container compatibility Virtual Resources

Main page : IPv4 space

Browse Add Delete

**networks (157)**

auto-collapsing at threshold 25 (expand all / collapse all)

prefix	name/tags	capacity	routed by
10.78.0.0/24 VLAN 8	VLAN-AGRIATES	.256	
172.27.128.0/24 VLAN 217	TOIP-OSN-DSDEN95	.256	
172.27.129.0/25 VLAN 202	ToIP-BAT-F	.128	
172.27.129.128/25 VLAN 203	ToIP-BAT-E	.128	
172.27.130.0/24 VLAN 204	ToIP-BAT-A	.256	
172.27.131.0/24 VLAN 205	ToIP-BAT-D-sub1	.256	
172.27.132.0/24 VLAN 206	ToIP-BAT-D-sub2	.256	
172.27.133.0/25 VLAN 207	ToIP-BAT-C	.128	
172.27.133.128/27 VLAN 208	ToIP-BAT-B	.32	
172.27.133.160/27 VLAN 209	ToIP-Entree-Hors-niveau	.32	
172.27.134.0/24 VLAN 210	ToIP-Lescot	.256	
172.27.135.0/25 VLAN 211	ToIP-Ceinture	.128	
172.27.136.0/24 VLAN 213	TOIP-GUY-DSDEN78	.256	
172.27.137.0/24 VLAN 214	TOIP-GUY-RECT	.256	

REC-DSI RackTables 0.20.11 Rackspace Objects IPv4 space 802.1Q Object container compatibility Virtual Resources RackTables Administrator [logout]

Main page : IPv4 space Search: \_\_\_\_\_

Browse Add Delete

**Add new**

Prefix:

VLAN:

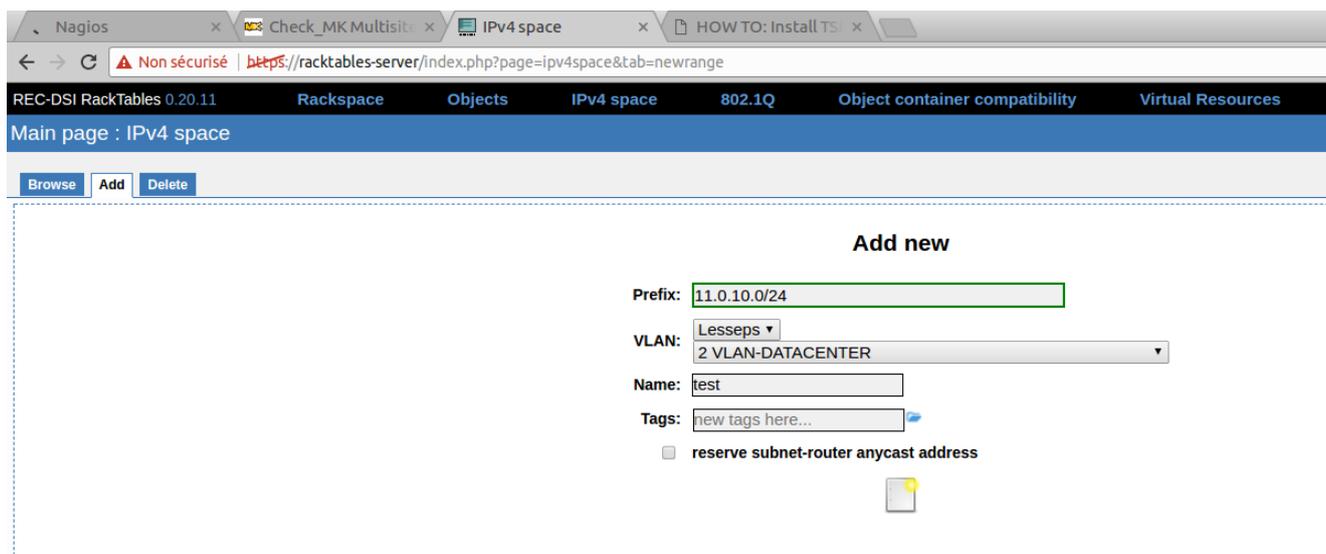
Name:

Tags:

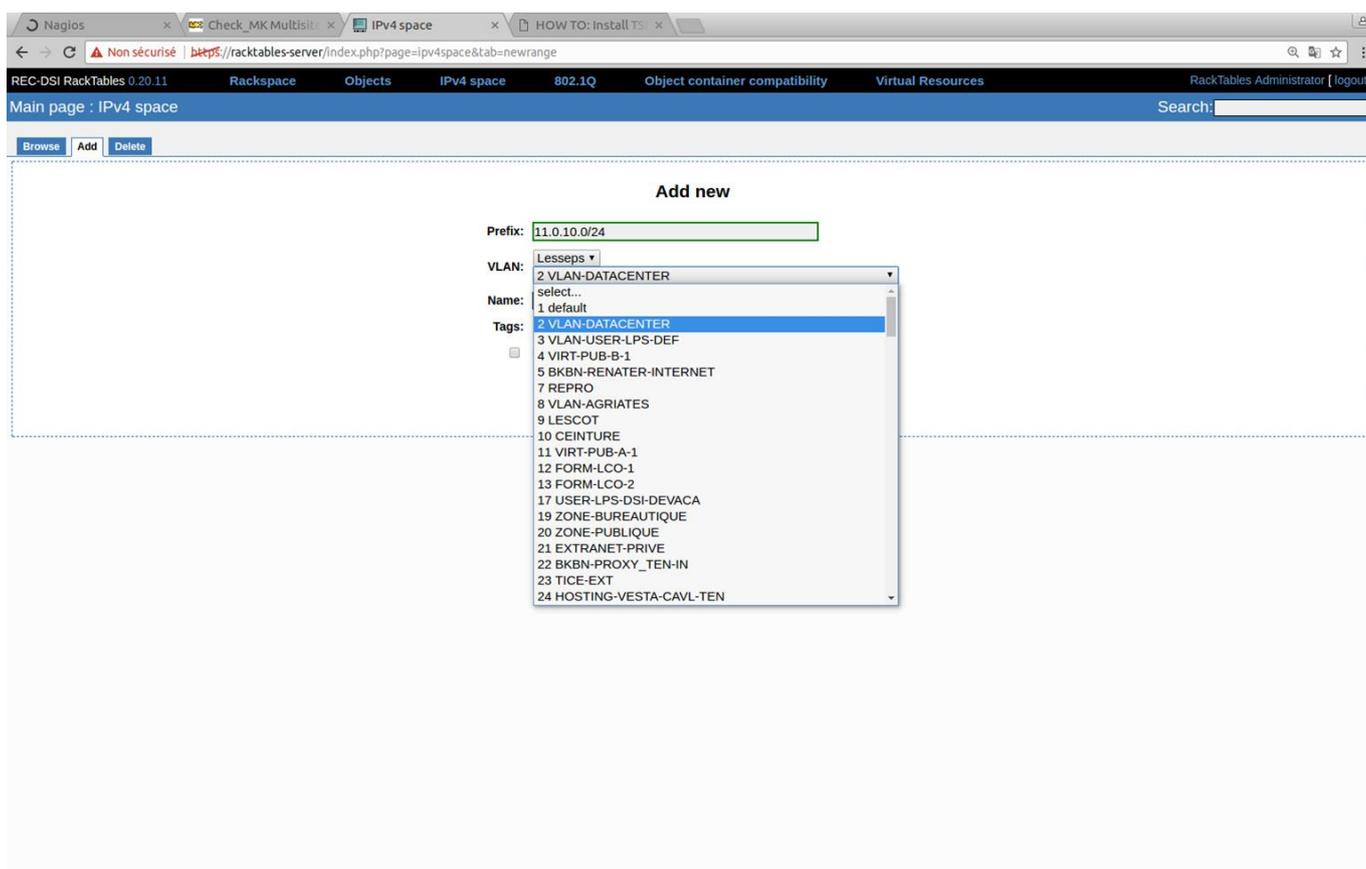
reserve subnet-router anycast address

Par exemple, on crée une adresse sous-réseau avec les informations suivantes :

adresse réseau : 11.0.10.0/24  
 Vlan : 2 Vlan-datacenter  
 Nom : test



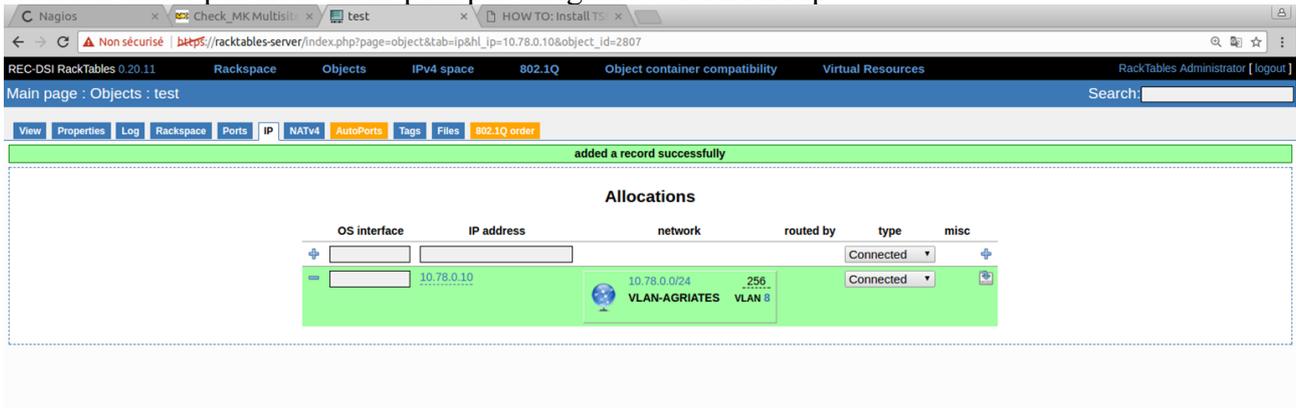
Les Vlans sont déjà créés, ils vous restent que à choisir le vlan correspondant dans le menu déroulant...



#### IV- Assigner une adresse ip à un objet

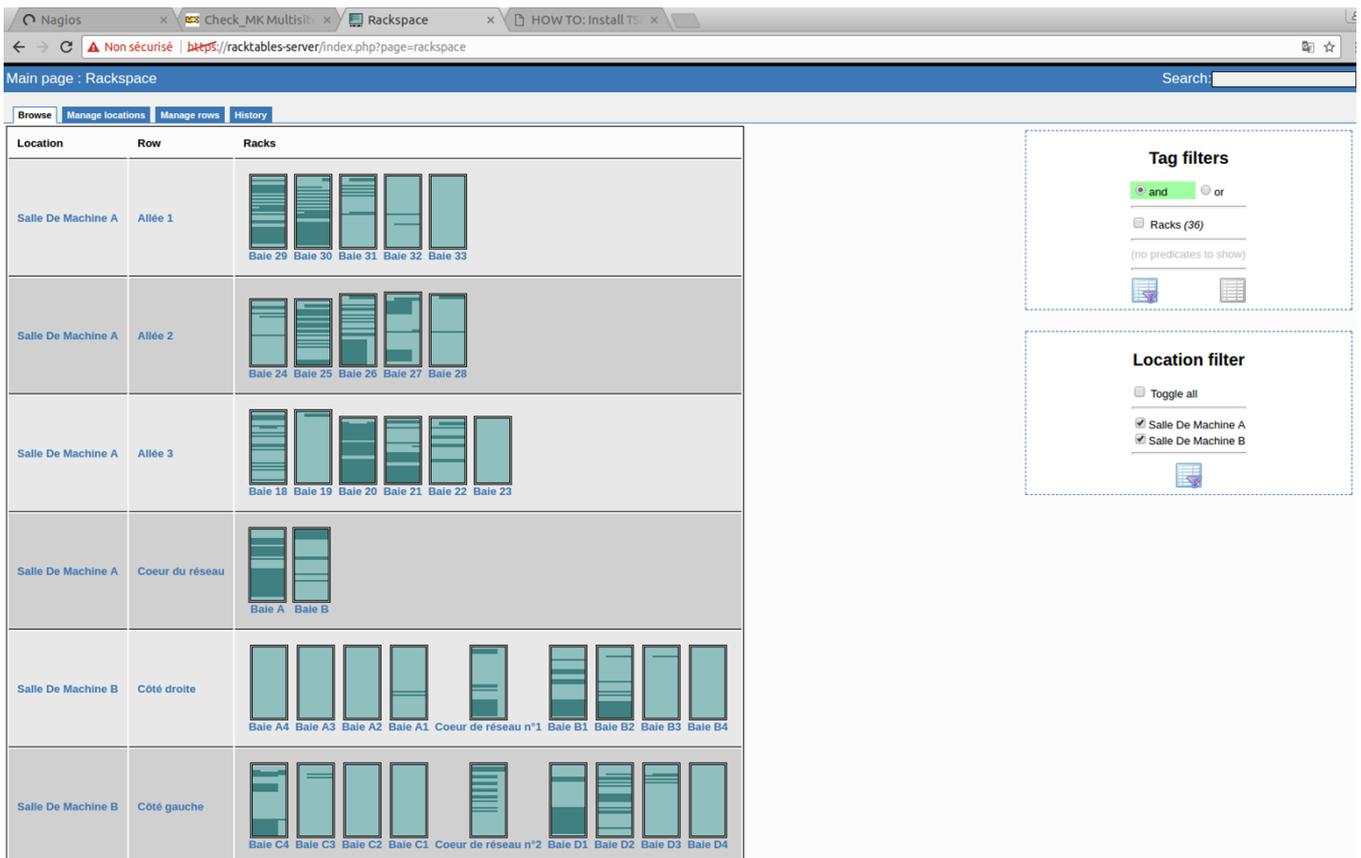
Par exemple, on va assigner l'adresse ip (10.78.0.10) suivante au serveur 'Test' :

Cliquez sur la rubrique « Ip » puis saisissez l'adresse ip comme ci-dessous. Attention si l'adresse réseau n'existe pas alors on ne peut pas assigner cette adresse ip.



## V- Gestion des baies

Pour gérer les baies, cliquez sur la rubrique « Rackspace ».



Pour ajouter un nouveau baie, cliquez le « Row » dans lequel vous voulez installer le baie. Par exemple, on prend le row 2 c-à-d « Allée 1 ».

REC-DSI RackTables 0.20.11 Rackspace Objects IPv4 space 802.1Q Object container compatibility Virtual Resources RackTables Administrator [logout]

Main page : Rackspace : Salle De Machine A : Allée 1 Search:

View Properties Manage racks Add new rack Log Tag roller Files

**Summary**

Name: Allée 1  
 Location: Salle De Machine A  
 Racks: 5  
 Units: 210  
 % used:

**Tag filters**

and  or

Racks (5)

(no predicates to show)

**Racks**

Baie 29 Baie 30 Baie 31 Baie 32 Baie 33

Puis cliquez sur « Add new rack »

REC-DSI RackTables 0.20.11 Rackspace Objects IPv4 space 802.1Q Object container compatibility Virtual Resources

Main page : Rackspace : Salle De Machine A : Allée 1

View Properties Manage racks Add new rack Log Tag roller Files

**Add one**

Name (required):

Height in units (required):

Asset tag:

Tags:

**Add many**

Height in units (\*):

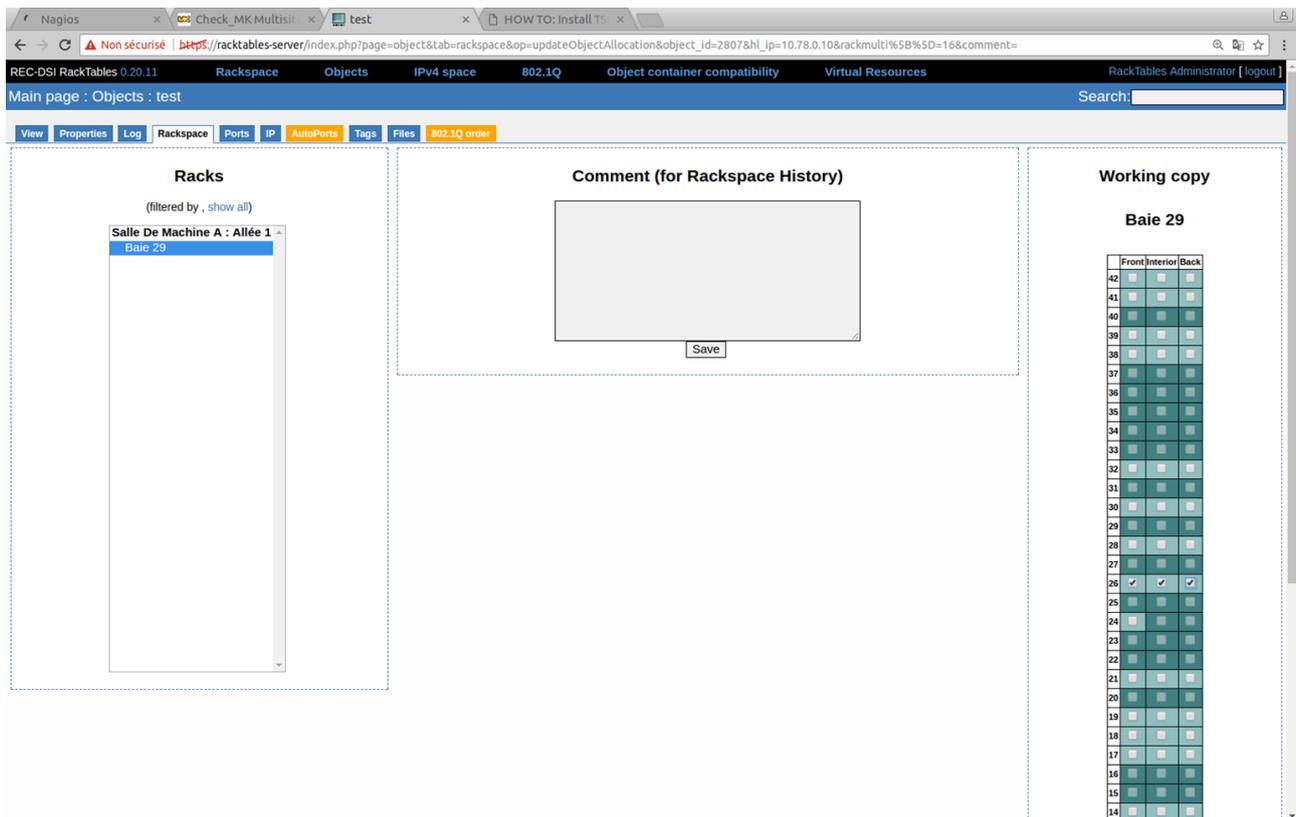
Assign tags:

Rack names (required):

**VI-**

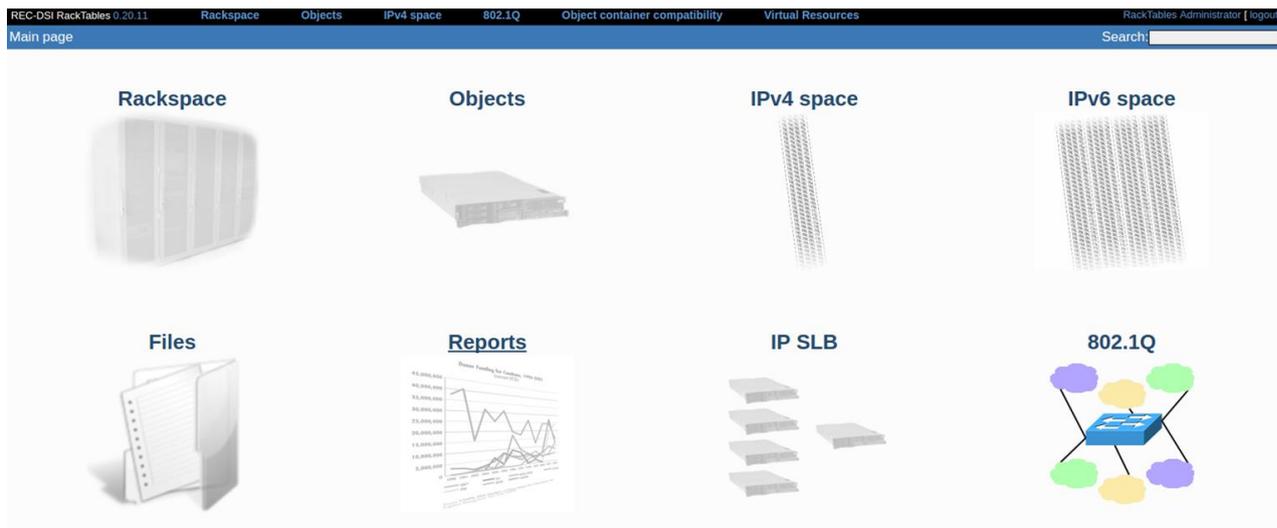
## Emplacement des Objets dans le baie

Pour définir l'emplacement d'un objet dans le baie, il faut d'abord aller dans les paramètres du serveur que vous souhaitez puis dans la rubrique « Rackspace » et cochez les cases que vous voulez allouer au serveur.



## VII- Gestion d'u blade et des lames(serveur)

Pour créer une blade (chassis de serveur), cliquez sur la rubrique « Object » puis « Add more »



Lors de la création du blade, on précise que le type de l'objet est un «Server chassis », comme ci-dessous :

**Distinct types, same tags**

Object type	Common name	Visible label	Asset tag	Tags
Server chassis ▼	test de blade			new tags here... Blade x
select type... ▼				
select type... ▼				
select type... ▼				
select type... ▼				
select type... ▼				
select type... ▼				
select type... ▼				

---

**Same type, same tags**

names

type

Server ▼

Tags

new tags here...

Note : Les blades sont considérés comme des conteneurs. On va voir dans la suite comment assigner un serveur dans un conteneur. Les emplacements dans le chassis de serveur sont gérés par le numéro de slot que l'on va indiquer dans le « Properties » des serveurs.

Par exemple, on placera le serveur 'Test' dans le slot 8 du Blade «test de blade », pour ceci dirigé vous vers le serveur « Test » puis dans la partie « Properties »

D'abord, on choisit le conteneur qui est «test de blade » :

REC-DSI RackTables 0.20.11    Rackspace    Objects    IPv4 space    802.1Q    Object container compatibility    Virtual Resources

Main page : Objects : test

View   Properties   Log   Rackspace   Ports   IP   AutoPorts   Tags   Files   802.1Q order

### Attributes

Type:

Common name:

Visible label:

Asset tag:

Tags:

Select container:

contact person:

FODN:

Puis, on choisit le blade parmi les propositions :

### Choose a container:

- blade1
- blade3
- blade5
- smb-blade01
- test de blade
- VersaillesA
- VersaillesB
- Vrs-test

Dans la partie « Slot number », on indique le numéro d'emplacement du serveur dans le blade.

## Attributes

Type: Server

Common name: test

Visible label:

Asset tag:

Tags: new tags here...

Server chassis container: test de blade

Select container: Select a container

contact person:

FQDN:

HW type: -- NOT SET --

HW warranty expiration (YYYY-MM-DD):

Hypervisor: -- NOT SET --

OEM S/N 1:

Slot number: 9

support contract expiration (YYYY-MM-DD):

SW type: -- NOT SET --

SW warranty expiration (YYYY-MM-DD):

UUID:

Has problems:

Actions:  

## Visualisation

34	db2-6
33	
32	
31	shaula
30	
29	supervision
28	
27	mx3
26	
25	mx4
24	SW-KVM-B29
23	sphinx-ng 2
22	
21	KVM-B29
19	
18	
17	
16	sphinx-ng 1
15	
14	
13	
	blade3
12	app2-sconet
11	jupiter
10	sympa6
9	app4-sconet
8	app-nat-1
7	srv-varonis1
6	app-admin
5	app-loc-3
4	app-nat-1
3	app-nat-1
2	
1	

d'un blade :

Dans ce

conteneur, on peut trouver les serveurs suivants :

- sympa
- jupiter
- app2-sconet
- bv7.in.ac-versailles.fr
- app-loc-3
- app-admin
- jcmsacac-1.in.ac-versailles.fr
- srv-varonis
- app-nat-1
- app4-sconet

## VIII- Gestion d'un hyperviseur

Un hyperviseur est en base un serveur. Par exemple, on va configurer le serveur «test » en tant qu'un hyperviseur.

**Attributes**

Type:

Common name:

Visible label:

Asset tag:

Tags:

Server chassis container:

Select container: [Select a container](#)

contact person:

FQDN:

HW type:

HW warranty expiration (YYYY-MM-DD):

Hypervisor:

OEM S/N 1:

Slot number:

support contract expiration (YYYY-MM-DD):

SW type:

SW warranty expiration (YYYY-MM-DD):

UUID:

Has problems:

Actions:

Dans la partie « Hypervisor », sélectionner « yes ».  
note : les hyperviseurs sont aussi traités comme des conteneurs.

### Attributes

Type: Server

Common name: test

Visible label:

Asset tag:

Tags: new tags here...

Server chassis container: test de blade

Select container: Select a container

contact person:

FQDN:

HW type: -- NOT SET --

HW warranty expiration (YYYY-MM-DD):

Hypervisor: Yes

OEM S/N 1:

Slot number:

support contract expiration (YYYY-MM-DD):

SW type: -- NOT SET --

SW warranty expiration (YYYY-MM-DD):

UUID:

Has problems:

Actions:  

N'oublier pas d'enregistrer les modifications...

### **Pour ajouter une Vm à un hyperviseur :**

Par exemple, on va ajouter le vm « admin-fim » à l'hyperviseur « test ».

REC-DSI RackTables 0.20.11 Rackspace Objects IPv4 space 802.1Q Object container compatibility Virtual Resources RackTables Administrator [logout]

Main page : Objects : admin-fim Search:

View Properties Log Ports IP Tags Files 802.1Q order

### Attributes

Type: VM

Common name: admin-fim

Visible label:

Asset tag:

Tags: new tags here...  
Vm

Select container: Select a container

contact person:

FQDN:

SW type: -- NOT SET --

SW warranty expiration (YYYY-MM-DD):

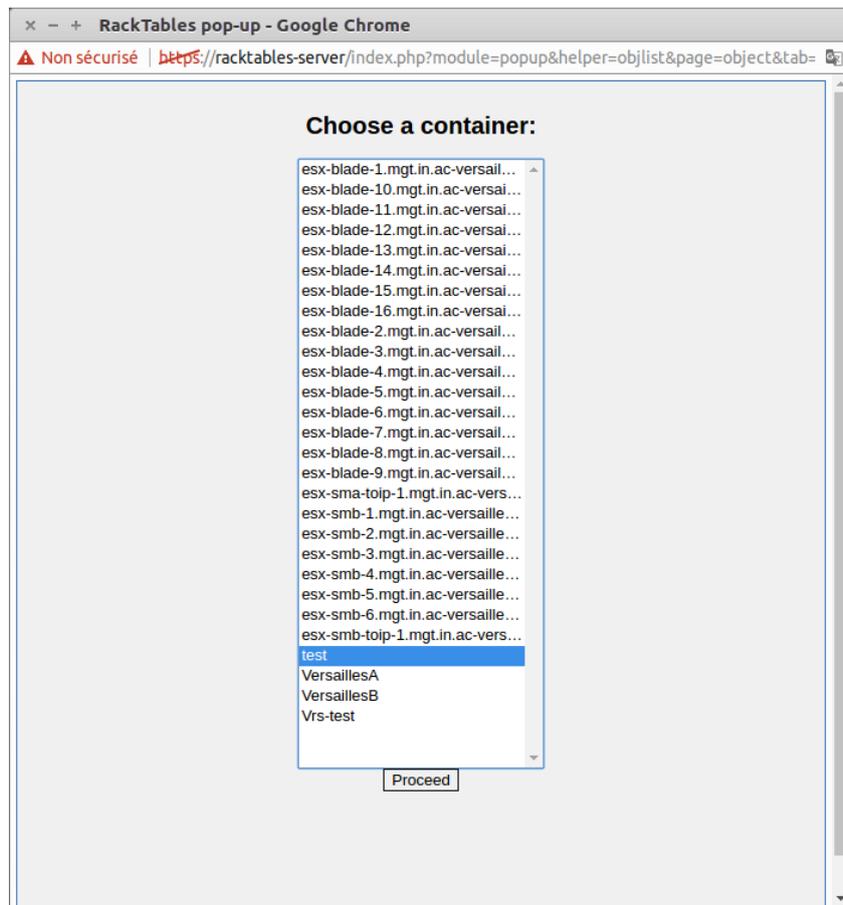
Has problems:

Actions:  

Comment:



Choisissez un conteneur



## **IX- Rapport du datacenter avec systèmes d'étiquetage**

Cliquer sur la rubrique « Reports »

REC-DSI RackTables 0.20.11    Rackspace    Objects    IPv4 space    802.1Q    Object container compatibility    Virtual Resources    RackTables Administrator [logout]

Main page : Reports    Search:

System    RackCode    IPv4    IPv6    Ports    802.1Q    Expirations    Data Integrity

---

**Dictionary/objects**

Total chapters in dictionary: 27  
 Total words in dictionary: 2267  
 User chapters: 0  
 Words in user chapters: 0  
 Total objects: 649  
 Objects with stickers: 120  
 Total stickers attached: 252

---

**Rackspace**

Rows: 6  
 Racks: 36  
 Average rack height: 41.2778  
 Total rack units in field: 1486

---

**Files**

Links in realm 'rack': 1  
 Unlinked files: 0  
 Total files: 1

---

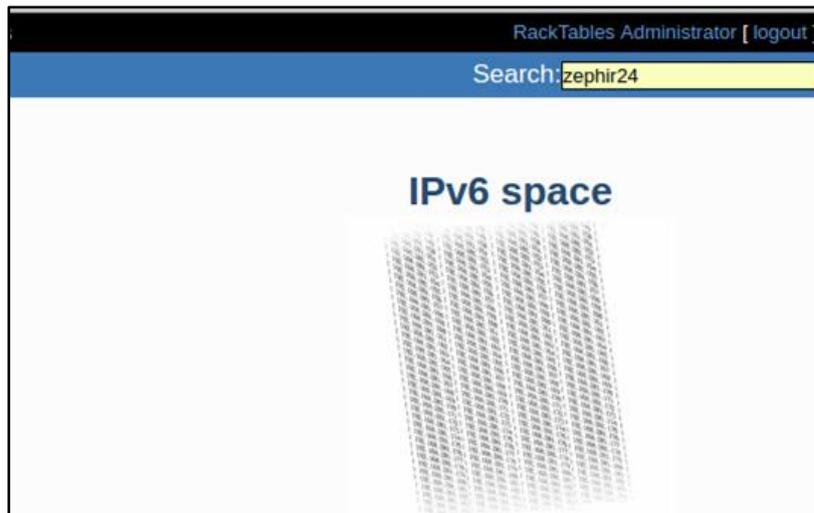
**Tags top list**

tag	total	objects	IPv4 nets	IPv6 nets	racks	IPv4 VS	IPv4 RS pools	users	files
Vm	370	370							
Serveur	157	157							
Lame	51	51							
Racks	36				36				
Switch nexus 2000	10	10							
Switch 2960	10	10							
Console Kvm	9	9							
Switch Kvm	8	8							
Switch	8	8							
Blade	5	5							
Virtual Switch	3	3							
Tape Library	2	2							
Routeur	2	2							
Switch du management SMB	1	1							
VTL	1	1							

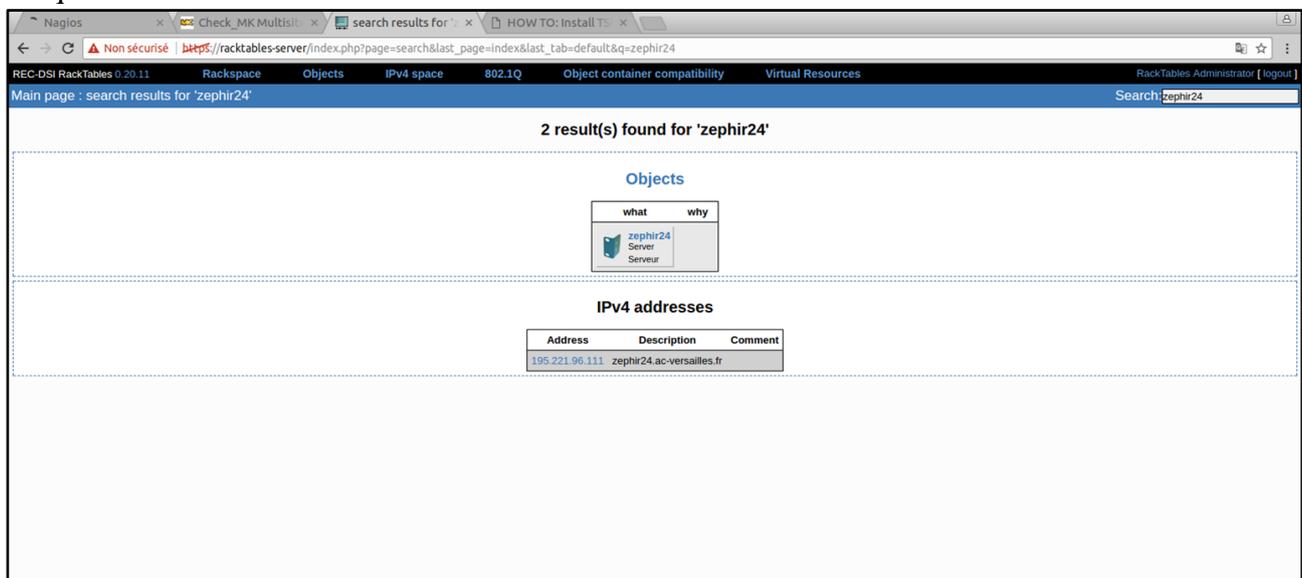
**X- Date de fin de garantie**

Pour retrouver la date de fin de garantie d'un serveur DELL:

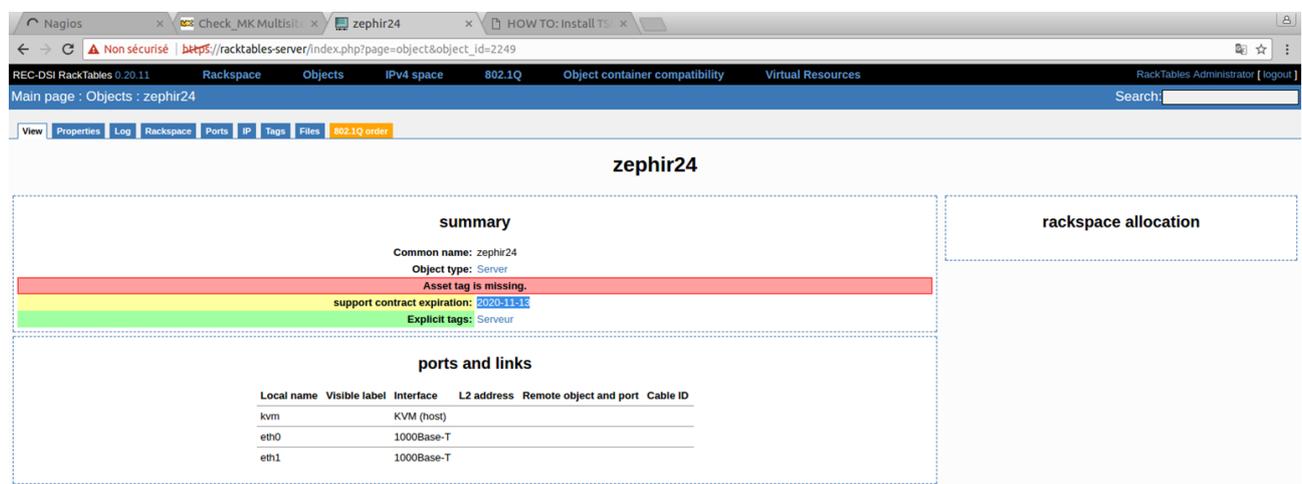
Faire une recherche avec le nom du serveur souhaité, dans notre cas on va s'intéresser sur le serveur « zephir24 » :



Cliquer sur le serveur

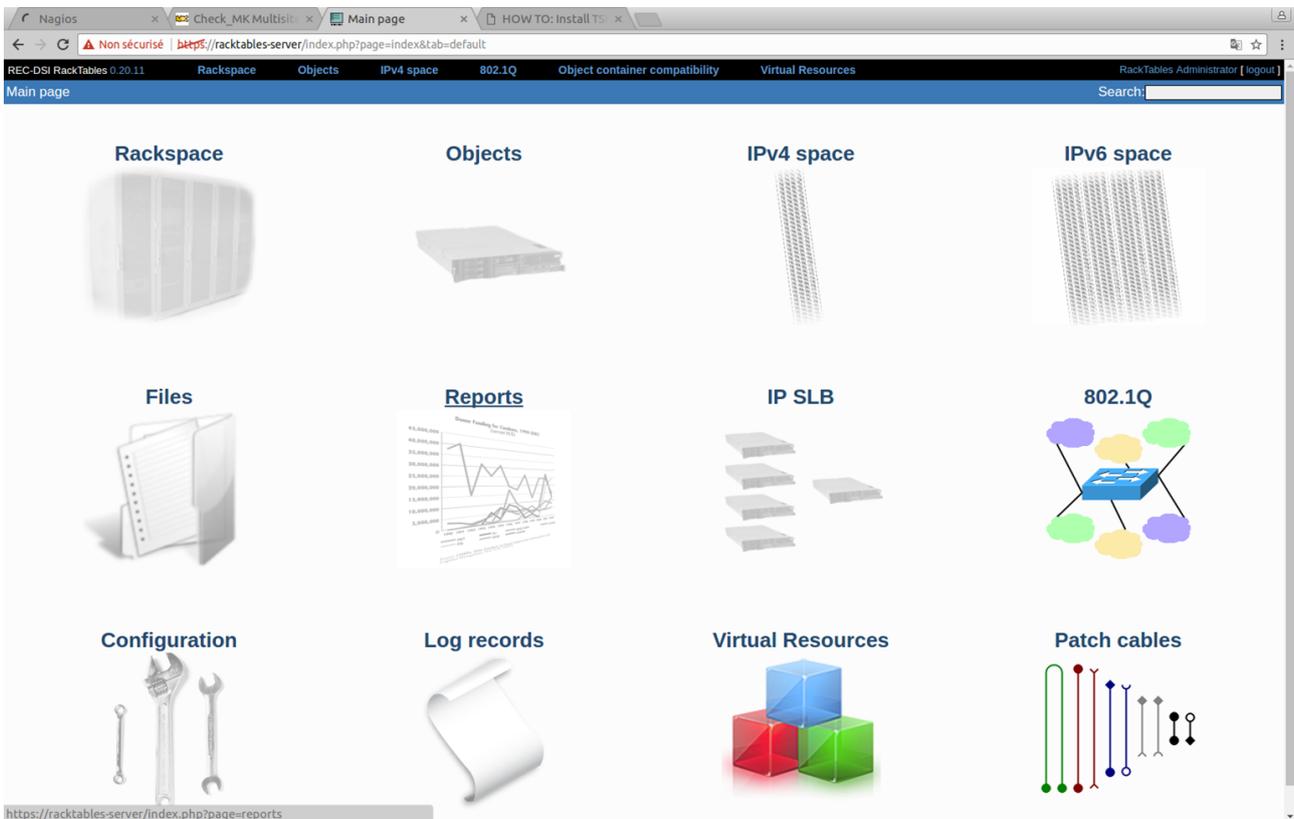


Dans la partie «Support contract expiration », on pouvez trouver la date de fin de garantie. Les dates de fin de garantie sont synchronisé avec OpenManage Essentials. Si un serveur n'a pas les infos de garantie, ça veut dire qu'il n'est pas remonté dans OME(OpenManage Essentials).



Pour retrouver tous les serveurs qui ne sont plus sous garanties et les futures serveurs qui vont être plus sous garantie.

Dans le tableau du bord, cliquez sur la rubrique «Reports ».



Puis dans la rubrique « Expirations » :

REC-DSI RackTables 0.20.11 Rackspace Objects IPv4 space 802.1Q Object container compatibility Virtual Resources RackTables Administrator [logout]

Main page: Reports Search:

System RackCode IPv4 IPv6 Ports 802.1Q Expirations Data Integrity

### support contract expiration

has expired within last year

Count	Name	Asset Tag	OEM S/N 1	Date Warranty Expires
1	srv-varonis2			2017-05-31
2	sympa6			2017-05-31

expires within 30 days  
(none)

expires within 60 days  
(none)

expires within 90 days

Count	Name	Asset Tag	OEM S/N 1	Date Warranty Expires
1	ldapi5			2017-10-19
2	mysql-msg2			2017-10-19

### HW warranty expiration

has expired within last year  
(none)

expires within 30 days  
(none)

expires within 60 days  
(none)

expires within 90 days  
(none)

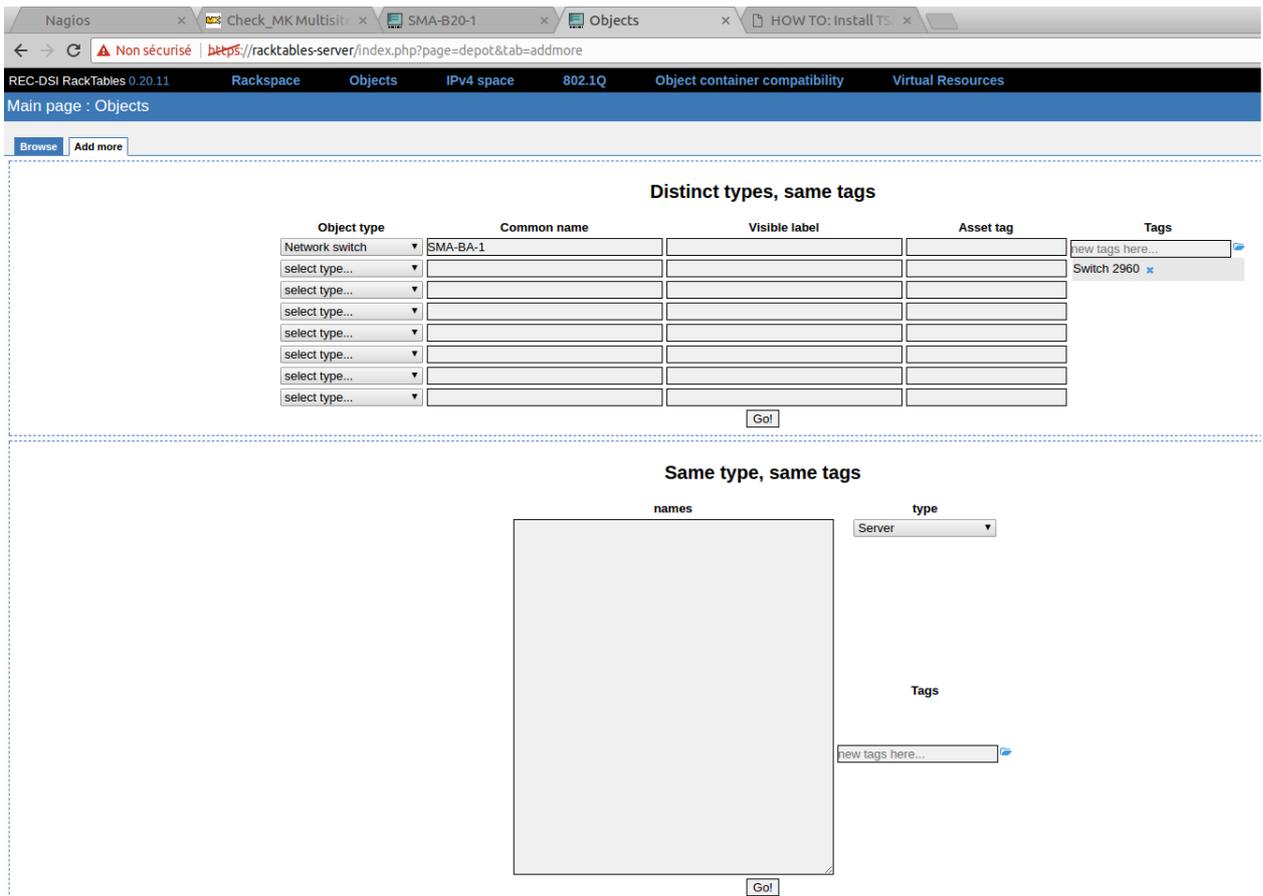
### SW warranty expiration

has expired within last year  
(none)

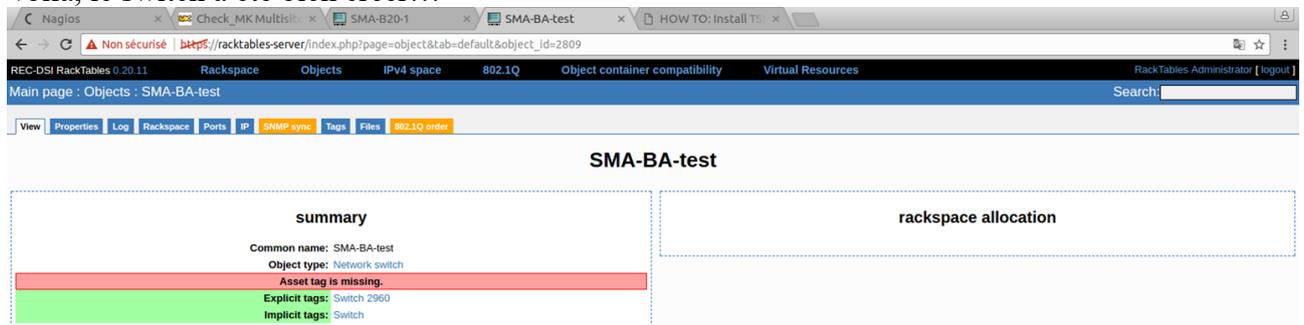
expires within 30 days

## **XI- Création et gestion d'un switch**

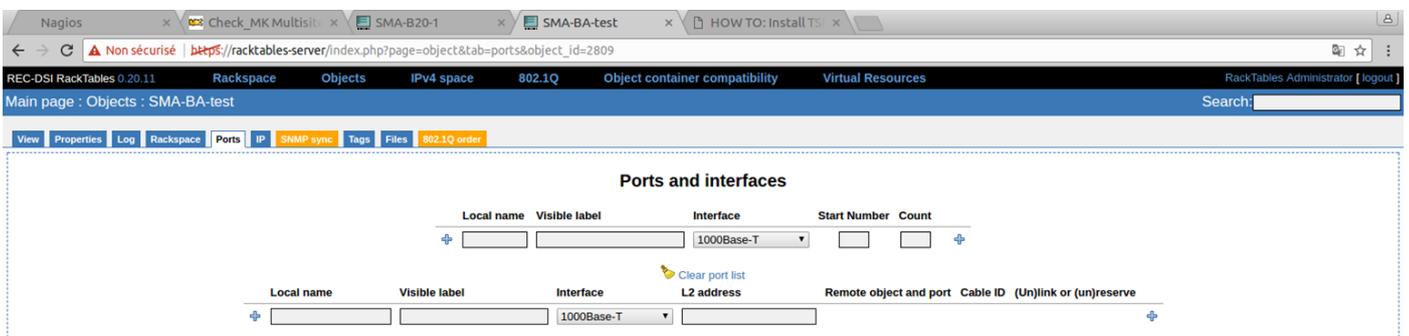
Pour créer un switch, c'est la même méthode classique (« Objects »----> « Add more »). Mais en précise bien que le type d'objet est un « Network switch ».



Voilà, le switch a été bien créer...



Pour ajouter les ports, allez dans la rubrique « Ports »...



Pour ajouter un port, il faut donner un nom pour chaque interface, moi je l'ai nommée Gigabit Ethernet1,2,3... jusqu'à 44 puisque le switch possède 44 ports.

Ports and interfaces

Local name	Visible label	Interface	Start Number	Count
<input type="text"/>	<input type="text"/>	1000Base-T	<input type="text"/>	<input type="text"/>

Clear port list

Local name	Visible label	Interface	L2 address	Remote object and port	Cable ID	(Un)link or (un)reserve
<input type="text"/>	<input type="text"/>	1000Base-T	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Gigabit Ethernet1	Not connected	1000Base-T	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Gigabit Ethernet2	Not connected	1000Base-T	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Gigabit Ethernet3	Not connected	1000Base-T	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Gigabit Ethernet4	Not connected	1000Base-T	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Gigabit Ethernet5	Not connected	1000Base-T	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Gigabit Ethernet6	Not connected	1000Base-T	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Gigabit Ethernet7	API-LNK SPD 2	1000Base-T	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Gigabit Ethernet8	API-LNK SPD 1	1000Base-T	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Gigabit Ethernet9	Not connected	1000Base-T	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Gigabit Ethernet10	Not connected	1000Base-T	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Gigabit Ethernet11	PIX ADRIATIC- Eth 1	1000Base-T	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Gigabit Ethernet12	PIX ADRIATIC- Eth 0	1000Base-T	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Gigabit Ethernet13	Not connected	1000Base-T	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Gigabit Ethernet14	Not connected	1000Base-T	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Gigabit Ethernet15	Not connected	1000Base-T	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Gigabit Ethernet16	Not connected	1000Base-T	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Gigabit Ethernet17	Blade5-sw1-17	1000Base-T	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Gigabit Ethernet18	ENC78-1 GE0/1	1000Base-T	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Gigabit Ethernet19	Not connected	1000Base-T	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Gigabit Ethernet20	Not connected	1000Base-T	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

### Visualisation du switch :

SMA-B20-1

#### summary

Common name: SMA-B20-1  
 Object type: Network switch  
 Visible label: Switch  
Asset tag is missing.  
 Explicit tags: Switch 2960  
 Implicit tags: Switch

#### ports and links

Local name	Visible label	Interface	L2 address	Remote object and port	Cable ID
Gigabit Ethernet1	Not connected	1000Base-T			
Gigabit Ethernet2	Not connected	1000Base-T			
Gigabit Ethernet3	Not connected	1000Base-T			
Gigabit Ethernet4	Not connected	1000Base-T			
Gigabit Ethernet5	Not connected	1000Base-T			
Gigabit Ethernet6	Not connected	1000Base-T			
Gigabit Ethernet7	API-LNK SPD 2	1000Base-T			
Gigabit Ethernet8	API-LNK SPD 1	1000Base-T			
Gigabit Ethernet9	Not connected	1000Base-T			
Gigabit Ethernet10	Not connected	1000Base-T			
Gigabit Ethernet11	PIX ADRIATIC- Eth 1	1000Base-T			
Gigabit Ethernet12	PIX ADRIATIC- Eth 0	1000Base-T			
Gigabit Ethernet13	Not connected	1000Base-T			
Gigabit Ethernet14	Not connected	1000Base-T			
Gigabit Ethernet15	Not connected	1000Base-T			
Gigabit Ethernet16	Not connected	1000Base-T			
Gigabit Ethernet17	Blade5-sw1-17	1000Base-T			
Gigabit Ethernet18	ENC78-1 GE0/1	1000Base-T			
Gigabit Ethernet19	Not connected	1000Base-T			
Gigabit Ethernet20	Not connected	1000Base-T			

#### rackspace allocation

Allée 3 : ← Baie 20 →

	Front	Interior	Back
38			
37			
36	7858 KALC VERS DPM05		SMA-B20-1
35	COMPLETEL LIEN PRINCIPAL	FIBRES 1/2 VV638 345	SMA-B20-2
34		COMPLETEL	
33		ENC78-COVAGE	
32		VV510613 FIBRES 7 et 8	
31			
30		ENC78-1	
29		CG82XX.VPN-EFFT	
28		CG82XX.VPN-EFPL	
27		0025WG79 WAN-ONS-1	
26		FIBRE 5 0034DW09	
25			
24			
23		PIX ADRIATIC	
22		API	
21		RACINE2-BKB-CES-RH	
20		RACINE2-BKB-VERSAILLES	
19			
18		RACINE2-BKB-POLE-VERSAILLES	
17			
16			
15			
14		VERS UFR VERSAILLES	
13		RTR WAN	
12		MASTER VRRP	
11			
10			
9			
8			
7			
6			
5			
4			
3			
2			
1			

## XII- Lier un port du switch à un objet

Pour lier un port du switch à un objet :

Allez dans les paramètres du switch... on va prendre le switch SMA-B20-1...

The screenshot shows the Nagios Rackspace interface for the switch SMA-B20-1. The browser address bar shows the URL: [https://racktables-server/index.php?page=object&tab=default&object\\_id=71&hl\\_ip=10.78.0.19](https://racktables-server/index.php?page=object&tab=default&object_id=71&hl_ip=10.78.0.19). The interface includes a navigation menu with options like Properties, Log, Rackspace, Ports, IP, NATv4, Tags, Files, 802.1Q order, and 802.1Q sync. The main content area is titled "SMA-B20-1" and is divided into three sections:

### summary

Common name: SMA-B20-1  
Object type: Network switch  
Visible label: Switch  
**Asset tag is missing.**  
Explicit tags: Switch 2950  
Implicit tags: Switch

### ports and links

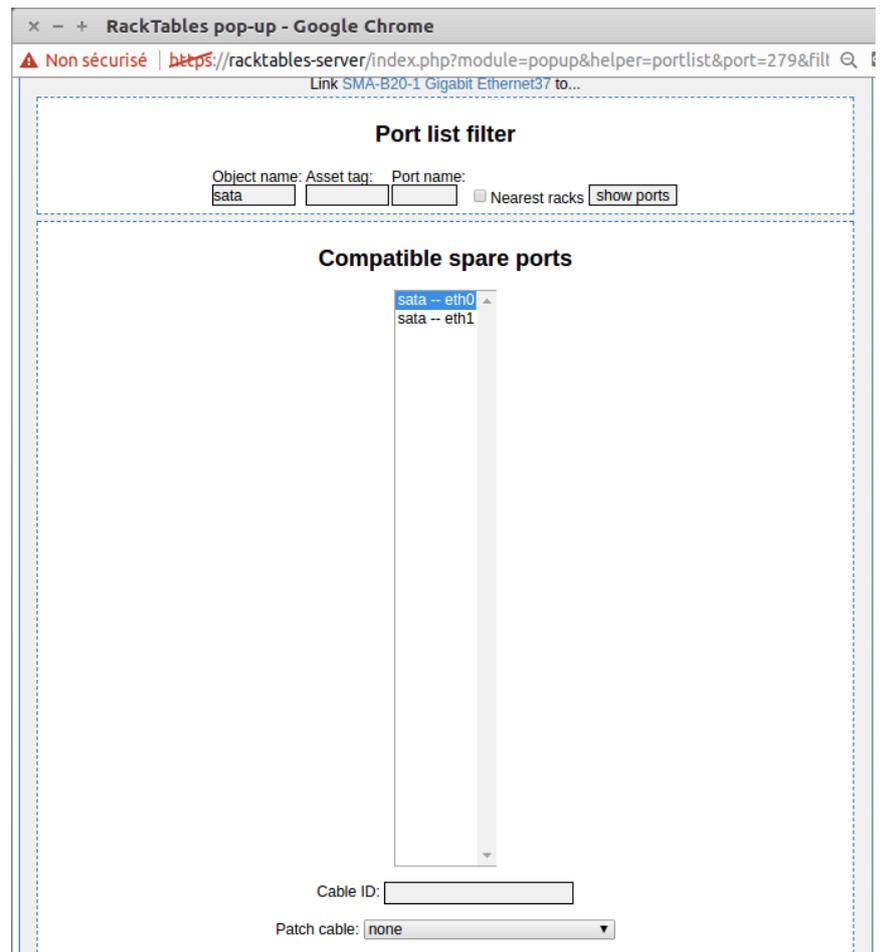
Local name	Visible label	Interface	L2 address	Remote object and port	Cable ID
Gigabit Ethernet1	Not connected	1000Base-T			
Gigabit Ethernet2	Not connected	1000Base-T			
Gigabit Ethernet3	Not connected	1000Base-T			
Gigabit Ethernet4	Not connected	1000Base-T			
Gigabit Ethernet5	Not connected	1000Base-T			
Gigabit Ethernet6	Not connected	1000Base-T			
Gigabit Ethernet7	API-LNK SPD 2	1000Base-T			
Gigabit Ethernet8	API-LNK SPD 1	1000Base-T			
Gigabit Ethernet9	Not connected	1000Base-T			
Gigabit Ethernet10	Not connected	1000Base-T			
Gigabit Ethernet11	PIX ADRIATIC- Eth 1	1000Base-T			
Gigabit Ethernet12	PIX ADRIATIC- Eth 0	1000Base-T			
Gigabit Ethernet13	Not connected	1000Base-T			
Gigabit Ethernet14	Not connected	1000Base-T			
Gigabit Ethernet15	Not connected	1000Base-T			
Gigabit Ethernet16	Not connected	1000Base-T			
Gigabit Ethernet17	Blade5-sw1-17	1000Base-T			
Gigabit Ethernet18	ENC78-1 GE0/1	1000Base-T			
Gigabit Ethernet19	Not connected	1000Base-T			
Gigabit Ethernet20	Not connected	1000Base-T			

### rackspace allocation

Allée 3 : Baie 20

	Front	Interior	Back
38			
37			
36	7888 KALC VERS DPM05		SMA-B20-1
35	COMPLETEL LIEN PRINCIPAL	FIBRES 1/2 VV638 345	SMA-B20-2
34		COMPLETEL	
33		ENC78-COVAGE	
32		VV510613 FIBRES 7 et 8	
31		ENC78-1	
30		CG92XX.VPN-EFFT	
29		CG92XX.VPN-EFPL	
28		0025WG79 WAN-ONS-1	
27		FIBRE 5 00340WC9	
26			
25			
24		PIX ADRIATIC	
23		API	
22			
21		RACINE2-BKB-CES-RH	
20		RACINE2-BKB-VERSAILLES	
19			
18		RACINE2-BKB-VERSAILLES	
17			
16			
15			
14		VERS UFR VERSAILLES	
13		RTR WAN	
12		MASTER VRRP	
11			
10			
9			
8			
7			
6			
5			
4			
3			
2			
1			

Puis dans la rubrique « Ports »...  
 On va assigner le port 37 à l'interface eth0 du serveur « sata ».  
 Attention, vérifier bien que les ports sont déjà créés dans le serveur que vous souhaitez utiliser.  
 Pour lier, il faut juste cliquer sur ce symbole  et choisir l'interface...

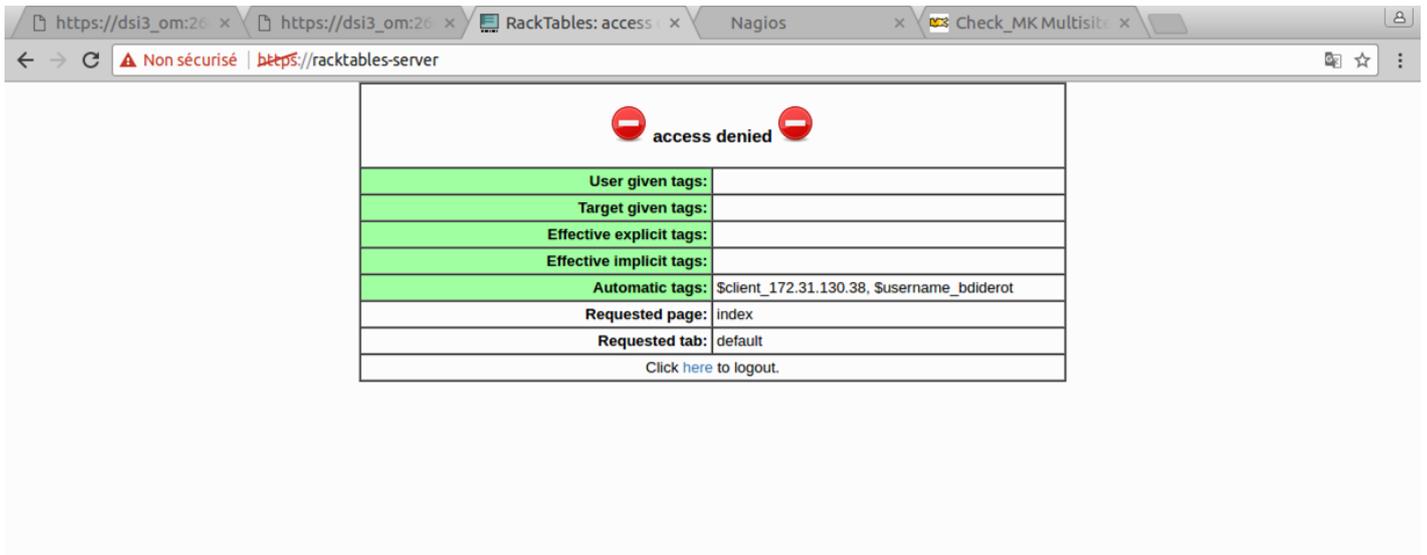


### XIII-Authentification

#### LDAP

Racktables est configuré pour reconnaître les comptes LDAP.

Donc, tous les utilisateurs du groupe DSI peuvent accéder à l'interface web de Racktables. Mais, ils n'auront pas tous les droits...



Pour attribuer les droits, connecter vous en tant que administrateur. Dans la rubrique « configuration » → « Permissions », ajouter l'utilisateur avec les droits que vous voulez attribuer. Par exemple, je vais attribuer tous les droits à l'utilisateur bdiderot.

View Edit

```
1 allow {$userid_1} #admin  
2  
3  
4 allow {$username_bdiderot}  
5
```

Verify Save