

# Linux tools

Bunch of linux tools installing process

- [Connect linux to AD](#)
- [Nala](#)
- [Rudder](#)
- [Ansible](#)
- [Illumio VEN](#)
- [Cheatsheet](#)
- [Bashtop](#)

# Connect linux to AD

First you need to install some lib

```
sudo apt install sssd-ad sssd-tools realmd adcli krb5-user
```

version kerberos → 5 et ne pas remplir le reste seulement cliquer sur “OK” puis redémarrer le package

Run la commande suivante pour voir si l’AD est joignable

```
sudo realm -v discover <domain>
```

vous aurez ce retour si l’AD est contactée

```
* Resolving: _ldap._tcp.<domain>
* Performing LDAP DSE lookup on: x.x.x.x
* Performing LDAP DSE lookup on: x.x.x.x
* Successfully discovered: <domain>
<domain>
type: kerberos
realm-name: <domain>
domain-name: <domain>
configured: kerberos-member
server-software: active-directory
client-software: sssd
required-package: sssd-tools
required-package: sssd
required-package: libnss-sss
required-package: libpam-sss
required-package: adcli
required-package: samba-common-bin
login-formats: %U@<domain>
login-policy: allow-realm-logins
```

Joignez l’AD avec votre compte administrateur (xxxxx\_adm)

```
sudo realm join <domain> -v --computer-  
ou='OU=xxxxx,OU=xxxxx,OU=xxxxx,OU=xxxxxx,OU=xxxxx,DC=xxxxx,DC=xxxxx' --user=xxx_adm  
Password for xxxxx_adm:
```

et vous aurez se retour si la connexion se fait bien

```
* Resolving: _ldap._tcp.ad1.example.com  
* Performing LDAP DSE lookup on: x.x.x.x  
* Successfully discovered: ad1.example.com  
Password for Administrator:  
* Unconditionally checking packages  
* Resolving required packages  
* LANG=C /usr/sbin/adcli join --verbose --domain ad1.example.com --domain-realm AD1.EXAMPLE.COM --  
domain-controller x.x.x.x --login-type user --login-user Administrator --stdin-password  
* Using domain name: ad1.example.com  
* Calculated computer account name from fqdn: AD-CLIENT  
* Using domain realm: ad1.example.com  
* Sending NetLogon ping to domain controller: x.x.x.x  
* Received NetLogon info from: SERVER1.ad1.example.com  
* Wrote out krb5.conf snippet to /var/cache/realmd/adcli-krb5-hUfTUg/krb5.d/adcli-krb5-conf-hv2kzi  
* Authenticated as user: xxxxxx_adm@AD1.EXAMPLE.COM  
* Looked up short domain name: AD1  
* Looked up domain SID: S-1-5-21-2660147319-831819607-3409034899  
* Using fully qualified name: ad-client.ad1.example.com  
* Using domain name: ad1.example.com  
* Using computer account name: AD-CLIENT  
* Using domain realm: ad1.example.com  
* Calculated computer account name from fqdn: AD-CLIENT  
* Generated 120 character computer password  
* Using keytab: FILE:/etc/krb5.keytab  
* Found computer account for AD-CLIENT$ at: CN=xxxxx,CN=xxxxx,DC=ad1,DC=example,DC=com  
* Sending NetLogon ping to domain controller: x.x.x.x  
* Received NetLogon info from: SERVER1.ad1.example.com  
* Set computer password  
* Retrieved kvno '3' for computer account in directory: CN=AD-  
CLIENT,CN=Computers,DC=ad1,DC=example,DC=com  
* Checking RestrictedKrbHost/ad-client.ad1.example.com  
* Added RestrictedKrbHost/ad-client.ad1.example.com  
* Checking RestrictedKrbHost/AD-CLIENT  
* Added RestrictedKrbHost/AD-CLIENT
```

```
* Checking host/ad-client.ad1.example.com
*   Added host/ad-client.ad1.example.com
* Checking host/AD-CLIENT
*   Added host/AD-CLIENT
* Discovered which keytab salt to use
* Added the entries to the keytab: AD-CLIENT$@AD1.EXAMPLE.COM: FILE:/etc/krb5.keytab
* Added the entries to the keytab: host/AD-CLIENT@AD1.EXAMPLE.COM: FILE:/etc/krb5.keytab
* Added the entries to the keytab: host/ad-client.ad1.example.com@AD1.EXAMPLE.COM: FILE:/etc/krb5.keytab
* Added the entries to the keytab: RestrictedKrbHost/AD-CLIENT@AD1.EXAMPLE.COM: FILE:/etc/krb5.keytab
* Added the entries to the keytab: RestrictedKrbHost/ad-client.ad1.example.com@AD1.EXAMPLE.COM:
FILE:/etc/krb5.keytab
* /usr/sbin/update-rc.d sssd enable
* /usr/sbin/service sssd restart
* Successfully enrolled machine in realm
```

Si votre serveur ne joins que un seul AD vous pouvez retirer la partie FQDN des utilisateurs dans `/etc/sss/sss.conf`

```
[sss]
domains = ad1.example.com
config_file_version = 2
services = nss, pam

[domain/ad1.example.com]
default_shell = /bin/bash
krb5_store_password_if_offline = True
cache_credentials = True
krb5_realm = AD1.EXAMPLE.COM
realmd_tags = manages-system joined-with-adcli
id_provider = ad
fallback_homedir = /home/%u
ad_domain = ad1.example.com
use_fully_qualified_names = False
ldap_id_mapping = True
access_provider = ad
```

Ensuite activez la création automatique des homedir et redémarrer le service `sss`

```
sudo pam-auth-update --enable mkhomedir
sudo service sssd restart
```

## Se login avec un utilisateur de l'AD

```
sudo login
```

```
ad-client login: john@ad1.example.com
```

```
Password:
```

```
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-24-generic x86_64)
```

```
...
```

```
Creating directory '/home/john@ad1.example.com'.
```

```
john@ad1.example.com@ad-client:~$
```

On peut aussi voir notre ticket kerberos avec la commande suivante:

```
john@ad1.example.com@ad-client:~$ klist
```

```
Ticket cache: FILE:/tmp/krb5cc_1725801106_9UxVlz
```

```
Default principal: john@AD1.EXAMPLE.COM
```

Valid starting	Expires	Service principal
----------------	---------	-------------------

04/16/20 21:32:12	04/17/20 07:32:12	krbtgt/AD1.EXAMPLE.COM@AD1.EXAMPLE.COM
-------------------	-------------------	--

```
⌘renew until 04/17/20 21:32:12
```

⚠ In case the server crashed the LDAP connexion can fail so just restart the `sssd` service and next the server:

```
sudo service sssd restart
```

```
reboot
```

⚠ After restart if the connexion always don't initiate check the space available on the `/` partition:

```
sudo df -h
```

#	Filesystem	Size	Used	Avail	Use%	Mounted on
---	------------	------	------	-------	------	------------

#	/dev/mapper/ubuntu--vg-ubuntu--lv	47G	11G	34G	25%	/
---	-----------------------------------	-----	-----	-----	-----	---

# Nala

Install nala:

```
sudo apt update && sudo apt install nala
```

Install a package with nala:

```
sudo nala install <package> -y
```

Display nala history:

```
sudo nala history
```

Delete an history entry to revert it:

```
sudo nala history undo <history-id>
```

# Rudder

## Installation rudder server

```
wget --quiet -O /etc/apt/trusted.gpg.d/rudder_apt_key.gpg "https://repository.rudder.io/apt/rudder_apt_key.gpg"
echo "deb http://repository.rudder.io/apt/7.2/ $(lsb_release -cs) main" > /etc/apt/sources.list.d/rudder.list
echo "machine download.rudder.io login LOGIN password PASSWORD" > /etc/apt/auth.conf.d/rudder.conf
chmod 640 /etc/apt/auth.conf.d/rudder.conf
apt-get update
apt-get install rudder-server
```

Change `LOGIN` and `PASSWORD` by yours

Then create an admin user:

```
rudder server create-user -u USERNAME
```

Replace `USERNAME` by what you want to name the admin account

---

## Installation rudder agent

```
wget --quiet -O /etc/apt/trusted.gpg.d/rudder_apt_key.gpg "https://repository.rudder.io/apt/rudder_apt_key.gpg"
echo "deb http://repository.rudder.io/apt/7.2/ $(lsb_release -cs) main" > /etc/apt/sources.list.d/rudder.list
echo "machine download.rudder.io login LOGIN password PASSWORD" > /etc/apt/auth.conf.d/rudder.conf
chmod 640 /etc/apt/auth.conf.d/rudder.conf
apt-get update
apt-get install rudder-agent
```

Change `LOGIN` and `PASSWORD` by yours

Then join the rudder server

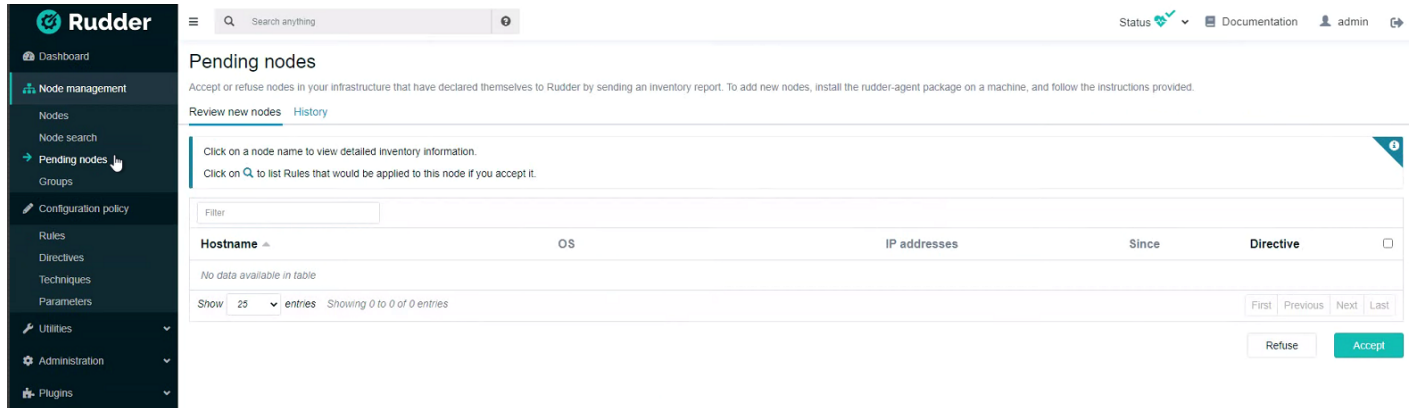
```
rudder agent policy-server <IP>
```

force the agent to run

```
rudder agent inventory
```

```
rudder agent run
```

Go on the rudder web GUI in **Pending nodes** tab and accept the new node



# Use Rudder as ansible inventory provider

Get the conf file and pt it in the ansible dir

```
cd /usr/share/ansible
```

```
wget https://raw.githubusercontent.com/ansible-community/contrib-scripts/main/inventory/rudder.py
```

```
wget https://raw.githubusercontent.com/ansible-community/contrib-scripts/main/inventory/rudder.ini
```

open **rudder.py** and change the first line to

```
#!/usr/bin/env python
```

```
#!/usr/bin/env python3
```

chmod **rudder.py** to 777 and then modify **rudder.ini**

```
uri = https://rudder.local/rudder/api
```

```
uri = https://<ip srv root rudder>/rudder/api
```

```
token = aaabbbcccc
```



```
token = <token readonly>
```

add script plugin in **ansible.cfg**

```
enable_plugins=script, auto, yaml, ini, toml, host_list
```

run the list command with **rudder.py** as inventory

```
ansible-inventory -i rudder.py --list
```

---

# Change root policy on node

```
rudder agent factory-rest  
rudder agent policy-server <ip>  
rudder agent inventory  
rudder agent run
```

Then accept the node on the root UI

# Ansible

## Ansible installation

Log in where you want to make your ansible server then run this commands:

```
sudo apt update && sudo apt install nala  
sudo nala install software-properties-common  
sudo add-apt-repository --yes --update ppa:ansible/ansible  
sudo nala install ansible -y
```

You can create a shared folder where you can put all your ansible script if you need to share it with other people in the team

```
sudo mkdir /usr/share/ansible  
sudo chmod g+w -R /usr/share/ansible/  
sudo chown -R root:<dev group> /usr/share/ansible/
```

For more safety you can replace the `root` user by an system user like for exemple `sys_ansible`

---

## Ansible commands

To execute playbook

```
ansible-playbook -K -i <inventaire> <playbook>
```

# Illumio VEN

1. retrieve the CA-root on the ansible srv connect to the ansible srv and execute the following commands:

```
cd /usr/share/illumio  
python3 -m http.server
```

2. install the CA-root on the VEN srv connect to the srv where the VEN must be installed and execute the following commands:

```
wget http://10.0.1.160:8000/bf-root-ca.crt  
sudo mv bf-root-ca.crt /usr/local/share/ca-certificates/  
sudo update-ca-certificates
```

3. install the VEN execute the following commands (replacing the parts between **\*\* \*\*** with the equivalents):

```
rm -fr /opt/illumio_ven_data/tmp && umask 026 && mkdir -p /opt/illumio_ven_data/tmp && curl --tlsv1  
"https://illumio-pce.businessfrance.fr:8443/api/v25/software/ven/image?pair_script=pair.sh&profile_id=**5**" -o  
/opt/illumio_ven_data/tmp/pair.sh && chmod +x /opt/illumio_ven_data/tmp/pair.sh &&  
/opt/illumio_ven_data/tmp/pair.sh --management-server illumio-pce.businessfrance.fr:8443 --activation-code  
**activation key**
```

4. close the python srv on ansible with the keys `ctrl+c`

# Cheatsheet

SOON

# Bashtop

```
sudo add-apt-repository ppa:bashtop-monitor/bashtop
```

```
sudo apt update
```

```
sudo apt install bashtop
```