

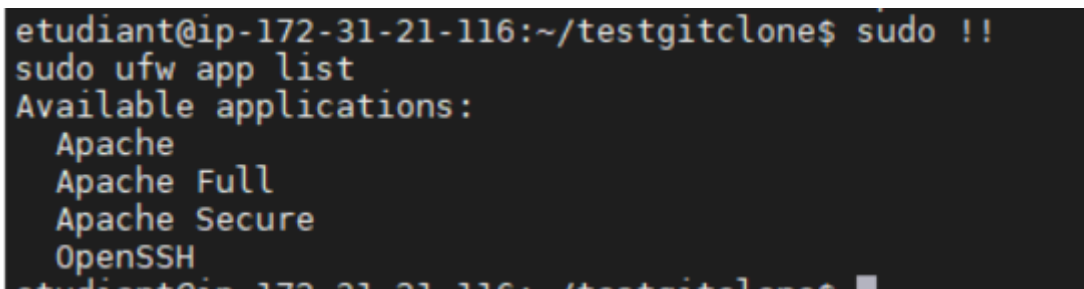
Préparation :

1. Dans une connexion SSH, le rôle du fichier `id_rsa` : elle sert de clé privée, c'est l'équivalent de la serrure qui forme le couple avec sa clé qui est dédiée dans notre cas elle correspond à la clé publique `id_rsa.pub`
2. Le NAT statique : correspond à la translation d'une adresse IP publique par une adresse IP privée (*à vérifier*)
3. Le port TCP par défaut pour le protocole SSH est le port 22.

- créer user avec sudo

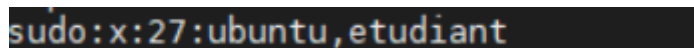
```
sudo adduser etudiant  
sudo usermod -aG sudo etudiant
```

Sudo et ufw fonctionne avec le compte etudiant :



```
etudiant@ip-172-31-21-116:~/testgitclone$ sudo !!  
sudo ufw app list  
Available applications:  
  Apache  
  Apache Full  
  Apache Secure  
  OpenSSH  
etudiant@ip-172-31-21-116:~/testgitclone$
```

En effet etudiant fait partie du groupe sudo : [cat /etc/sudoers](#)



```
sudo:x:27:ubuntu,etudiant
```

On a exécuté cette commande :

```
sudo rsync --archive --chown=etudiant:etudiant ~/.ssh /home/etudiant
```

Voici les paquets installés :

```
etudiant@ip-172-31-21-116:~/testgitclone$ sudo dpkg -l | grep "apache2"
ii  apache2                2.4.29-1ubuntu4.14
ii  apache2-bin            2.4.29-1ubuntu4.14
ii  apache2-data           2.4.29-1ubuntu4.14
ii  apache2-utils          2.4.29-1ubuntu4.14
ii  libapache2-mod-php     1:7.2+60ubuntu1
)
ii  libapache2-mod-php7.2  7.2.24-0ubuntu0.18.04.7
etudiant@ip-172-31-21-116:~/testgitclone$ sudo dpkg -l | grep "mysql"
ii  mysql-client-5.7        5.7.34-0ubuntu0.18.04.1
ii  mysql-client-core-5.7   5.7.34-0ubuntu0.18.04.1
ii  mysql-common            5.8+1.0.4
ii  mysql-server            5.7.34-0ubuntu0.18.04.1
ii  mysql-server-5.7        5.7.34-0ubuntu0.18.04.1
ii  mysql-server-core-5.7   5.7.34-0ubuntu0.18.04.1
ii  php-mysql               1:7.2+60ubuntu1
ii  php7.2-mysql            7.2.24-0ubuntu0.18.04.7
etudiant@ip-172-31-21-116:~/testgitclone$ sudo dpkg -l | grep "php"
ii  libapache2-mod-php     1:7.2+60ubuntu1
)
ii  libapache2-mod-php7.2  7.2.24-0ubuntu0.18.04.7
ii  php                    1:7.2+60ubuntu1
ii  php-bz2                1:7.2+60ubuntu1
ii  php-cli                 1:7.2+60ubuntu1
```

Le mysql_native_password au lieu des sockets classiques sont mis en place (phpmyadmin pourrait être utilisé) :

```
mysql> SELECT user,authentication_string,plugin,host FROM mysql.user;
+-----+-----+-----+-----+
| user          | authentication_string | plugin           | host      |
+-----+-----+-----+-----+
| root          | *2FF63E0FF8B165484127006F60C79702E1317E14 | mysql_native_password | localhost |
| mysql.session | *THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE | mysql_native_password | localhost |
| mysql.sys     | *THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE | mysql_native_password | localhost |
| debian-sys-maint | *E64F5DA8CE611DD0602EB263D46D8F25D08001EC1 | mysql_native_password | localhost |
| owncloud      | *E881F6EE79B111481490E19C3FB64CA5D35B61AA | mysql_native_password | localhost |
+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```

Contenu du modules apache *dir.conf* :

```
etudiant@ip-172-31-21-116:~/testgitclone$ sudo cat /etc/apache2/mods-enabled/dir.conf
<IfModule mod_dir.c>
    DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm
</IfModule>
```

Contenu de /etc/apache2/conf-available/ssl-params.conf :

```
etudiant@ip-172-31-21-116:~/testgitclone$ sudo cat /etc/apache2/conf-available/ssl-params.conf
SSLCipherSuite EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH
SSLProtocol All -SSLv2 -SSLv3 -TLSv1 -TLSv1.1
SSLHonorCipherOrder on
# Disable preloading HSTS for now. You can use the commented out header line that includes
# the "preload" directive if you understand the implications.
# Header always set Strict-Transport-Security "max-age=63072000; includeSubDomains; preload"
Header always set X-Frame-Options DENY
Header always set X-Content-Type-Options nosniff
# Requires Apache >= 2.4
SSLCompression off
SSLUseStapling on
SSLStaplingCache "shmcb:logs/stapling-cache(150000)"
# Requires Apache >= 2.4.11
SSLSessionTickets Off
etudiant@ip-172-31-21-116:~/testgitclone$
```

Modification de /etc/apache2/sites-available/default-ssl.conf :

```
etudiant@ip-172-31-21-116:~/testgitclone$ sudo cat /etc/apache2/sites-available/default-ssl.conf
<IfModule mod_ssl.c>
    <VirtualHost _default_:443>
        ServerAdmin zemail@u-pec.fr
        ServerName 54.167.179.185

        #
        DocumentRoot /var/www/html
        DocumentRoot /var/www/owncloud

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
    </VirtualHost>
</IfModule>
```

Modification de /etc/apache2/sites-available/000-default.conf :

```
etudiant@ip-172-31-21-116:~/testgitclone$ sudo cat /etc/apache2/sites-available/000-default.conf
<VirtualHost *:80>
    # 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 www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/owncloud

    Redirect "/" "https://54.167.179.185/"
    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

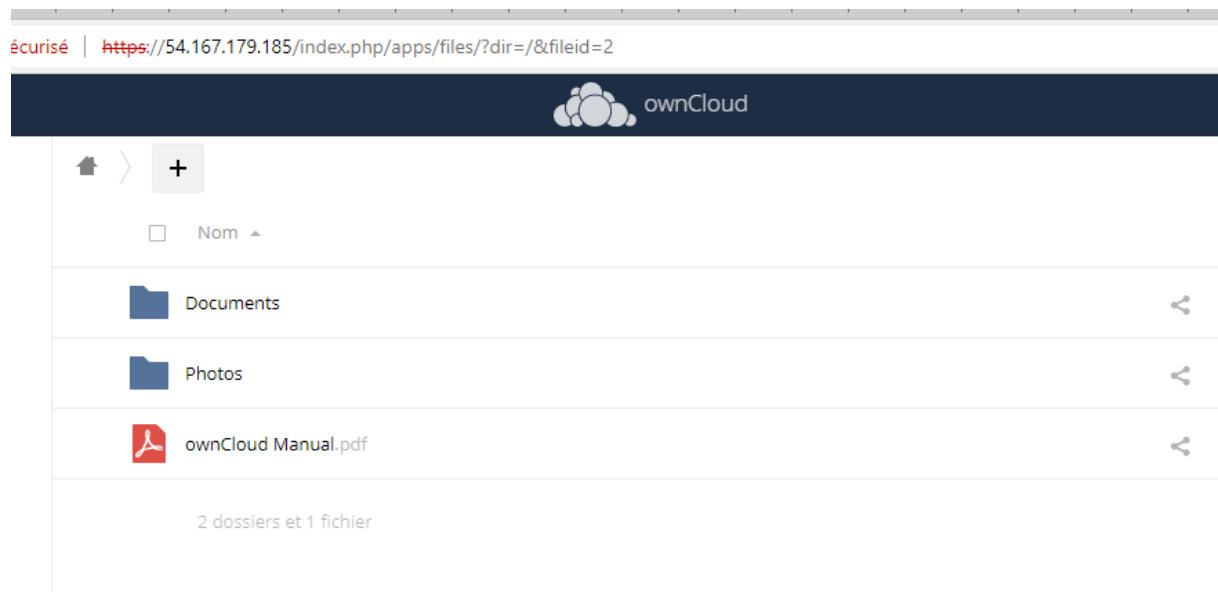
Activation des modules apache avec a2enmod pour SSL et headers ainsi que a2ensite pour le default-ssl

On a aussi après à débattu la config d'APACHE via `sudo apache2ctl configtest && sudo systemctl restart apache2`

On a installé Owncloud : Voici l'interface de connexion



Voici les fichiers de notre Cloud :

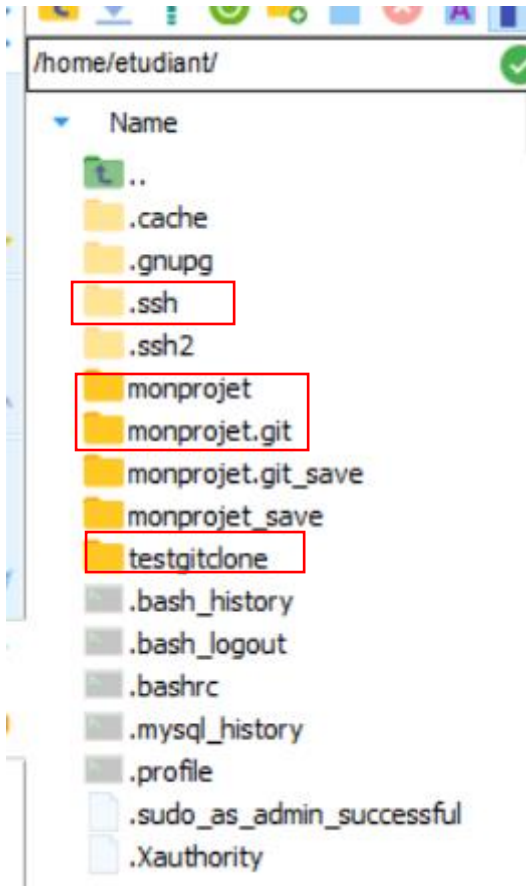


Nous avons configuré GIT :

```
git init --bare /home/etudiant/monprojet.git
```

```
cd /home/etudiant/.ssh/ && chmod 600 *.pem
```

```
git clone etudiant@172.31.21.116:/home/etudiant/monprojet.git GIT_SSH_COMMAND='sudo ssh -i  
/home/etudiant/.ssh/*.pem -o IdentitiesOnly=yes' git clone  
etudiant@172.31.21.116:/home/etudiant/monprojet.git
```



Voici le dossier cloner après la configuration du serveur Git (côté client) :

```
Last login: Thu May 27 17:27:57 2021 from 91.163.139.165  
etudiant@ip-172-31-21-116:~$ cd testgitclone/  
etudiant@ip-172-31-21-116:~/testgitclone$ ls  
monprojet  
etudiant@ip-172-31-21-116:~/testgitclone$ ls monprojet/
```


Synthèse 1 :

Nous sommes en charge de la gestion site web comme serveur de fichier (own)Cloud, on a géré le serveur git pour versionner en local et en privée.

Tout d'abord, pour cela, nous nous sommes connectées chez AWS EC2 : avec des adresses élastiques.

Nous avons géré les règles du firewall, effectuer en spécifiant la clé publique.

On a configuré Owncloud, en installant les prérequis LAMP et avec un certificat SSL autosigné.

On a installé Git en SSH, gestion des accès par un groupe, on initialise un dépôt en précisant --bare pour le partage via le groupe.

Après l'installation de Python3 et Python3-pip ainsi que [pip install awscli](#) :

```
sudo apt-get update -y && sudo apt-get install python3 python3-pip -y
```

```
ii python3-idna 2.6-1 all Python IDNA2008 (RFC 5891) handling (Python 3)
ii python3-incremental 16.10.1-3 all Library for versioning Python projects,
ii python3-jinja2 2.10-1ubuntu0.18.04.1 all small but fast and easy to use stand-alone template engine
ii python3-json-pointer 1.10-1 all resolve JSON pointers - Python 3.x
ii python3-jsonpatch 1.19+really1.16-1fakesync1 all library to apply JSON patches - Python 3.x
ii python3-jsonschema 2.6.0-2 all An(other) implementation of JSON Schema (Draft 3 and 4) - Pyt
ii python3-jwt 1.5.3+ds1-1 all Python 3 implementation of JSON Web Token
ii python3-keyring 10.6.0-1 all store and access your passwords safely - Python 3 version of
ii python3-keyrings.alt 3.0-1 all alternate backend implementations for python3-keyring
ii python3-lib2to3 3.6.9-1-18.04 all Interactive high-level object-oriented language (2to3, versio
ii python3-markupsafe 1.0-1build1 amd64 HTML/XHTML/XML string library for Python 3
ii python3-minimal 3.6.7-1-18.04 amd64 minimal subset of the Python language (default python3 versio
ii python3-netifaces 0.10.4-0.1build4 amd64 portable network interface information - Python 3.x
ii python3-newt:amd64 0.52.20-1ubuntu1 amd64 NEWT module for Python3
ii python3-oauthlib 2.0.6-1 all generic, spec-compliant implementation of OAuth for Python3
ii python3-openssl 17.5.0-1ubuntu1 all Python 3 wrapper around the OpenSSL library
ii python3-pam 0.4.2-13.2ubuntu4 amd64 Python interface to the PAM library
ii python3-pexpect 4.2.1-1 all Python 3 module for automating interactive applications
ii python3-pip 9.0.1-2.3-ubuntu1.18.04.4 all Python package installer
ii python3-pkg-resources 39.0.1-2 all Package Discovery and Resource Access using pkg_resources
ii python3-problem-report 2.20.9-0ubuntu7.24 all Python 3 library to handle problem reports
ii python3-ptyprocess 0.5.2-1 all Run a subprocess in a pseudo terminal from Python 3
ii python3-pyasn1 0.4.2-3 all ASN.1 library for Python (Python 3 module)
ii python3-pyasn1-modules 0.2.1-0.2 all Collection of protocols modules written in ASN.1 language (Py
ii python3-requests 2.18.4-2ubuntu0.1 all elegant and simple HTTP library for Python3, built for human
ii python3-requests-unixsocket 0.1.5-3 all Use requests to talk HTTP via a UNIX domain socket - Python 3
ii python3-secretstorage 2.3.1-2 all Python module for storing secrets - Python 3.x version
ii python3-serial 3.4-2 all pyserial - module encapsulating access for the serial port
ii python3-service-identity 16.0.0-2 all Service identity verification for pyOpenSSL (Python 3 module)
ii python3-setuptools 39.0.1-2 all Python3 Distutils Enhancements
ii python3-six 1.11.0-2 all Python 2 and 3 compatibility library (Python 3 interface)
ii python3-software-properties 0.96.24-32.14 all manage the repositories that you install software from
ii python3-systemd 234-1build1 amd64 Python 3 bindings for systemd
ii python3-twisted 17.9.0-2ubuntu0.1 all Event-based framework for internet applications
ii python3-twisted-bin:amd64 17.9.0-2ubuntu0.1 amd64 Event-based framework for internet applications
ii python3-update-manager 1:18.04.11.13 all python 3.x module for update-manager
ii python3-urllib3 1.22-1ubuntu0.18.04.2 all HTTP library with thread-safe connection pooling for Python3
ii python3-wheel 0.30.0-0.2 all built-package format for Python
ii python3-xdg 0.25-4ubuntu1.1 all Python 3 library to access freedesktop.org standards
ii python3-yaml 3.12-1build2 amd64 YAML parser and emitter for Python3
ii python3-zope.interface 4.3.2-1build2 amd64 Interfaces for Python3
ii python3.6 3.6.9-1-18.04ubuntu1.4 amd64 Interactive high-level object-oriented language (version 3.6)
ii python3.6-dev 3.6.9-1-18.04ubuntu1.4 amd64 Header files and a static library for Python (v3.6)
ii python3.6-minimal 3.6.9-1-18.04ubuntu1.4 amd64 Minimal subset of the Python language (version 3.6)
ubuntu@ip-172-31-27-215:~$
```

- Installer CLI AWS :

```
sudo pip3 install awscli
```

```
ubuntu@ip-172-31-27-215:~$ pip3 list
DEPRECATION: The default format will switch to columns in the future. You can disable this warning with
--list-format=json. (Use --help for more information.)
asn1crypto (0.24.0)
attrs (17.4.0)
Automat (0.6.0)
awscli (1.19.82)
```

Après la création du dossier .aws

Fichier .aws/config :

```
sudo cat > ~/.aws/config << EOF
# Datacenter de Virginie
region=us-east-1
EOF
```

Synthèse 2 :

Nous avons installé les paquetages linux : Python3, Python, Python3-pip.

Nous avons installé le paquet Python : awscli.

Nous avons récupéré le credential sur AwsEducate puis créé le dossier .aws et enfin nous avons configuré la région par défaut de la CLI AWS.

Nous avons manipulé le firewall en lignes de commandes et identifier les paramètres nécessaires.

Nous avons tenté d'utiliser les commandes. Mais elles sont parfois obsolètes.

Synthèse 3 :

```
C: > Users > Administrateur > Documents > Cdocuments > Eagain > UE36-PRIVATE > {} ec2_modified-token-protected.json > ...
1  {
2      "builders": [
3      {
4          "type": "amazon-ebs",
5          "region": "us-east-1",
6          "source_ami": "ami-0747bdcabd34c712a",
7          "instance_type": "t2.micro",
8          "ssh_username": "ubuntu",
9          "ami_name": "ubuntu/images/hvm-ssd/ubuntu-bionic-18.04-amd64-server-20210415",
10         "access_key": "ASIA4A3Q4IWM*****NMGV",
11         "secret_key": "zHAG2/1/tcJ3998AWjt0Xf1F*****j/EaTki5/1h"
12     }
13 ]
14 }
15 }
```

Nous avons utilisé le builder PACKER pour AMAZON EC2.

On a géré les identifiants AWS. On a automatisé la création d'une AMI.

[ue36/recap_commands.md at main · amineAUPEC/ue36 · GitHub](#)