

## **RAPPORT DE PROJET**

---

*PROJET UF*

---

*Projet réalisé par :*

ABOUBAKAR ALI

&

FAKIR Youssef

*Projet encadré par :*

AARAM Anas

anas.aaram@ynov.com

## REMERCIEMENTS

Au moment où ce travail prend sa forme définitive, nous tenons à témoigner toute notre reconnaissance à notre encadreur, Monsieur AARAM Anas, pour le temps qu'il a consacré à nous apporter les outils méthodologiques indispensables à la conduite de cette recherche, d'avoir eu la patience de répondre à nos innombrables questions, de nous avoir aussi aidé dans notre travail par son expérience. Son exigence nous a grandement stimulé.

# **SOMMAIRE**

## **I- OBJECTIFS PROJET**

- a) Objectifs Techniques
- b) Délais Du Projet

## **II- GESTION PROJET**

- a) Planification du projet
- b) Objectifs du projet

## **III- CONFIGURATION RESEAUX ET SYSTEM**

- a) Configuration réseaux
- b) Configuration system

## I-BESOINS ET OBJECTIFS DU PROJET

### **a) Les objectifs techniques**

Contacté par la société LIPS basée à Taiwan, il sera question pour nous mettre en place un système de sauvegarde qui serait placé dans les salles de serveurs de production.

### **b) Les Délais**

Le projet débute le jeudi 17 Mars et s'achève le vendredi 15 Avril 2022, soit le jour de la présentation ; une durée de 28 jours. Afin de terminer ce projet ambitieux à temps, il est important de correctement le gérer et de le tenir à jour grâce aux outils de gestion adéquats.

## II- GESTION DE PROJET

### **a- La planification du projet**

Pour accompagner le développement du projet dès le stade initial, des cours sur l'Infrastructure et Système d'information nous ont été transmis par notre encadreur au sein du Campus. Ces cours nous définissent les notions et introduisent des méthodologies à suivre.

### **b- Les objectifs du projet**

Après une analyse approfondie du projet nous avons pu fixer les objectifs du projet qui sont cités ci-dessous :

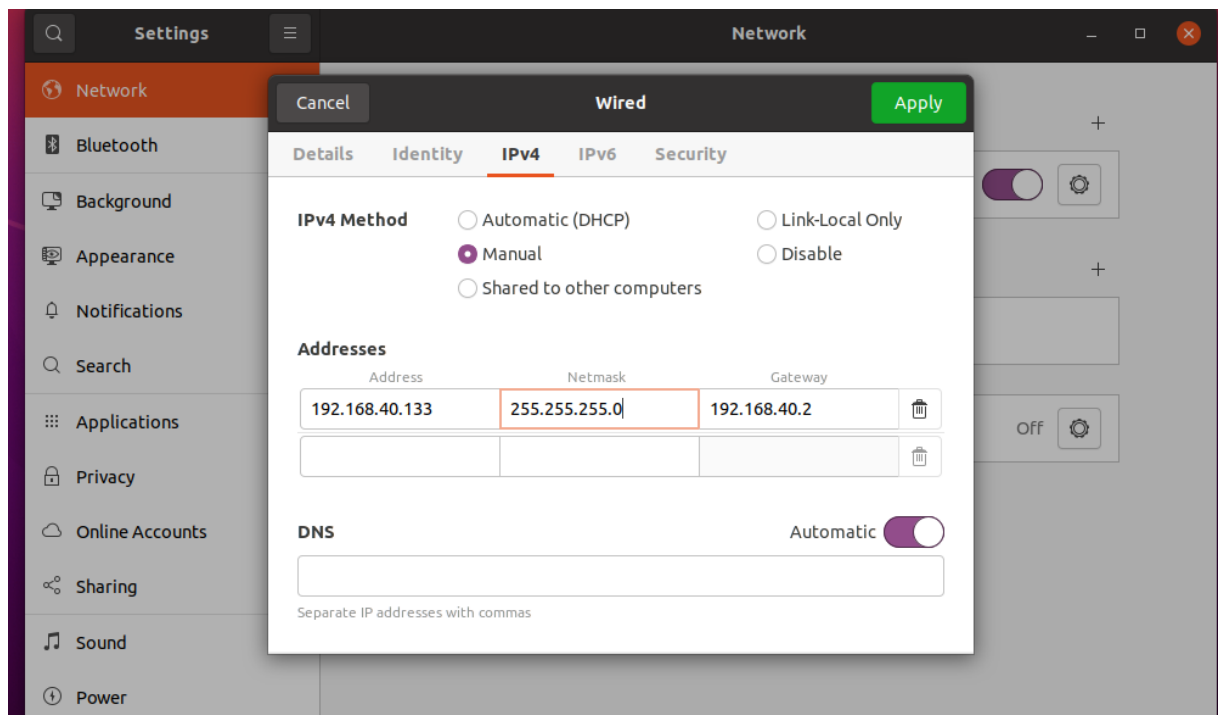
- ADMINISTRER UN POSTE.
- CONFIGURER UN RÉSEAU SIMPLE.
- GERER UN ENVIRONNEMENT VIRTUEL.
- METTRE EN PLACE UNE INTERACTION CLIENT-SERVEUR.
- APPRÉHENDER LA SÉCURITÉ.

### III – CONFIGURATION RESEAUX ET SYSTEM

#### a) Configuration réseaux

Dans cette partie, il sera question pour nous de présenter les étapes qui nous ont permis de configurer notre réseau.

➔ Mise en place du réseau privé et attribution de l'IP



## ➔ Configuration du server

```
GNU nano 4.8 /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    ens33:
      addresses: [192.168.40.130/24]
      gateway4: 192.168.40.129
      nameservers:
        addresses: [4.2.2.2, 8.8.8.8]
  version: 2
```

## ➔ Interconnexion entre les deux machines virtuelles

```
client@ubuntu:~$ ping 192.168.40.130
PING 192.168.40.130 (192.168.40.130) 56(84) bytes of data.
64 bytes from 192.168.40.130: icmp_seq=1 ttl=64 time=1.02 ms
64 bytes from 192.168.40.130: icmp_seq=2 ttl=64 time=1.03 ms
64 bytes from 192.168.40.130: icmp_seq=3 ttl=64 time=0.970 ms
64 bytes from 192.168.40.130: icmp_seq=4 ttl=64 time=1.25 ms
64 bytes from 192.168.40.130: icmp_seq=5 ttl=64 time=0.976 ms
64 bytes from 192.168.40.130: icmp_seq=6 ttl=64 time=0.946 ms
64 bytes from 192.168.40.130: icmp_seq=7 ttl=64 time=1.04 ms
^C
```

## b) Configuration réseaux

### ➔ Automatisation des CRON JOBS

```
GNU nano 4.8 /tmp/crontab.G0UtdE/crontab Modified
# 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
0 6 * * 1,5 /home/client/make_backup.sh
```

### ➔ Etablissement de la sauvegarde entre les machines

```
client@ubuntu:~$ rsync -av -e ssh /make_backup/ server@192.168.40.130:/home/client
The authenticity of host '192.168.40.130 (192.168.40.130)' can't be established.
ECDSA key fingerprint is SHA256:nILQmPjpCcISr4TF4qgBGeor80Dcis5sJbW0th5AOZQ.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.40.130' (ECDSA) to the list of known hosts.
server@192.168.40.130's password:
sending incremental file list
rsync: change_dir "/make_backup" failed: No such file or directory (2)

sent 20 bytes  received 12 bytes  1.49 bytes/sec
total size is 0  speedup is 0.00
rsync error: some files/attrs were not transferred (see previous errors) (code 23) at main.c(1207) [sender=3.1.3]
client@ubuntu:~$
```



## ➔ Script Backup de sauvegarde

```
client@ubuntu: ~  
GNU nano 4.8 make_backup.sh  
function perform_backup {  
    #get backup location  
    backup_path=$(cat /backup_loc.conf)  
  
    echo "Starting backup..." > $LOG_LOC  
    #for each dir , archive and compress to backup location  
    while read dir_path  
    do  
        #get backup dir name  
        dir_name=$(basename $dir_path)  
  
        #create filename for compressed backup  
        filename=$backup_path$dir_name.tar.gz  
  
        #archive dirs and compress archive  
        tar -zcf $filename $dir_path 2>> $LOG_LOC  
  
        #change ownership of backup files  
        chown client:client $filename  
  
        echo "Backing up of $dir_name completed." >> $LOG_LOC  
    done < /backup_dirs.conf  
  
    echo "Backup completed at :" >> $LOG_LOC  
    date >> $LOG_LOC  
}
```

```
client@ubuntu: ~  
GNU nano 4.8 make_backup.sh  
#!/bin/bash  
#  
## create weekly backup of selected directories  
#  
## variables  
LOG_LOC="/var/log/mybackup.log"  
##  
  
function check_dir_loc {  
    #check for dir list file  
    if [ ! -s "/backup_dirs.conf" ]  
    then  
        echo "Please create a list of directories to backup by creating a backups_dir.conf file in the root directory"  
        exit 1  
    fi  
}  
  
function check_backup_loc {  
    if [ ! -s "/backup_loc.conf" ]  
    then  
        echo "Please specify the full path of where to send the backup to, by creating the backup_loc.conf file in the root directory"  
        exit 1  
    fi  
}
```