# Syslog Proxy Server Setup on Ubuntu

## Objective

The goal is to create a server with two network interfaces: one connected to a local network to collect syslog messages from devices, and one connected to a remote Syslog server.

## Architecture

- rsyslog receives syslog messages via UDP on port 6666, updates the timestamp, and writes them to a buffer file.  
- The server is synchronized via NTP.  
- A Python script reads the buffer, deletes outdated messages, attempts to send the messages via TCP to the remote server (port 6666), and deletes them upon successful transmission.  
- The script is scheduled to run at regular intervals.

## rsyslog Installation and Configuration

### Install rsyslog:

```bash  
sudo apt update  
sudo apt install rsyslog  
```

Configure rsyslog to listen on UDP port 6666, replace the timestamp, and store messages in a buffer file at /opt/syslog-relay/buffer-relay.log

### Edit the configuration file:

```bash  
sudo nano /etc/rsyslog.d/66-udp-buffer-relay.conf  
```

Insert the following:

###############################

# Proxy syslog UDP vers buffer

# - Remplace la date par celle de réception

# - Enlève le timestamp original de l’émetteur

###############################

# Charge le module UDP

module(load="imudp")

input(type="imudp" port="6666" name="udp-buffer")

# Template personnalisée : format BSD classique

template(name="CleanBufferFormat" type="string"

string="<%pri%>%timegenerated:::date-bsd% %hostname% %app-name%: %msg%\n")

# Application : uniquement pour les messages venant du port 6666 UDP

if $inputname == 'udp-buffer' then {

action(type="omfile"

file="/opt/syslog-relay/buffer-relay.log"

template="CleanBufferFormat")

stop

}

Restart rsyslog:

```bash  
sudo systemctl restart rsyslog  
```

### Create buffer directory and set permissions:

```bash  
sudo mkdir -p /opt/syslog-relay  
sudo nano /opt/syslog-relay/buffer-relay.log  
sudo chown syslog:syslog /opt/syslog-relay /opt/syslog-relay/buffer-relay.log  
sudo chmod 775 /opt/syslog-relay  
sudo chmod 664 /opt/syslog-relay/buffer-relay.log  
```

Check if buffer receives logs:

```bash  
tail -f /opt/syslog-relay/buffer-relay.log  
```

## Copy and Configure the Python Script

Install TFTP server on a Windows machine with the Python files and download them on the Ubuntu server:

```bash  
sudo apt install tftp  
cd /opt/syslog-relay/  
tftp <TFTP\_SERVER\_IP>  
binary  
get syslog-relay.py  
get syslog-relay-config.txt  
quit  
```

Edit the configuration file to set the remote server IP and TTL:

```bash  
nano syslog-relay-config.txt  
```

Make the script executable system-wide:

```bash  
sudo ln -s /opt/syslog-relay/syslog-relay.py /usr/local/bin/syslog-relay  
```

Test the script manually:

```bash  
sudo python3 /opt/syslog-relay/syslog-relay.py  
```

## Automate Execution with Cron

Edit crontab:

```bash  
sudo crontab -e  
```

Add the following line to run the script every 2 minutes:

```bash  
\*/2 \* \* \* \* /usr/bin/python3 /opt/syslog-relay/syslog-relay.py >> /var/log/syslog-relay-cron.log 2>&1  
```

## NTP Time Synchronization

Install chrony:

```bash  
sudo apt update && sudo apt install chrony  
```

Edit chrony configuration file:

```bash  
sudo nano /etc/chrony/chrony.conf  
```

Example content with local NTP server:

# chrony.conf — configuration de Chrony avec priorité au serveur local NTP

# Inclure les fichiers de configuration supplémentaires (conservé par défaut)

confdir /etc/chrony/conf.d

# Serveur NTP local — utilisé en priorité

server 10.10.1.91 iburst prefer

# Désactiver l'ajout dynamique de sources via DHCP ou fichiers

# sourcedir /run/chrony-dhcp

# sourcedir /etc/chrony/sources.d

# Fichier de clés d'authentification NTP (optionnel ici)

keyfile /etc/chrony/chrony.keys

# Fichier de dérive pour ajustement du taux d'horloge

driftfile /var/lib/chrony/chrony.drift

# Répertoire de sauvegarde pour les clés NTS

ntsdumpdir /var/lib/chrony

# Activer le logging si nécessaire

# log tracking measurements statisticsRestart chrony and verify sync:

```bash  
sudo systemctl restart chrony  
chronyc tracking  
chronyc sources  
```