

Project 4: Automated Provisioning, Centralized Logging & Disk Quota Management

System Administration on CentOS 9 Stream

Nmar Yassir

Contents

1	Introduction and Environment Preparation	2
1.1	Virtual Machine Setup	2
1.2	Network Topology	2
1.3	Network Configuration, IP Addressing, DHCP and DNS	2
1.3.1	Static IP Configuration on centosServer	2
1.3.2	Static IP Configuration on centosClient	3
1.3.3	DHCP Server Installation and Configuration	3
1.3.4	Manual DNS Configuration	3
1.3.5	Connectivity Verification	3
2	Use Case Diagram	3
3	Part 1: Automated Provisioning	4
3.1	Script Content	4
4	Part 2: Centralized Logging with Rsyslog	5
4.1	Server Configuration (Receiving Logs)	5
4.2	Client Configuration (Sending Logs)	5
4.3	Verification	5
5	Part 3: XFS Quota Management	6
5.1	Setting Limits	6
5.2	Testing Limits	6
6	Conclusion	6
7	References	7

1 Introduction and Environment Preparation

The goal of this project is to simulate a professional Linux environment using CentOS 9 Stream. We automated server provisioning, centralized system logging using Rsyslog, and implemented disk quotas using XFS.

1.1 Virtual Machine Setup

Two virtual machines were created using Oracle VirtualBox:

- **centosServer**: central log server and quota manager.
- **centosClient**: client machine sending logs to the server.

1.2 Network Topology

Each virtual machine uses two network adapters:

1. **NAT**: for Internet access.
2. **Host-Only**: internal network for direct communication.

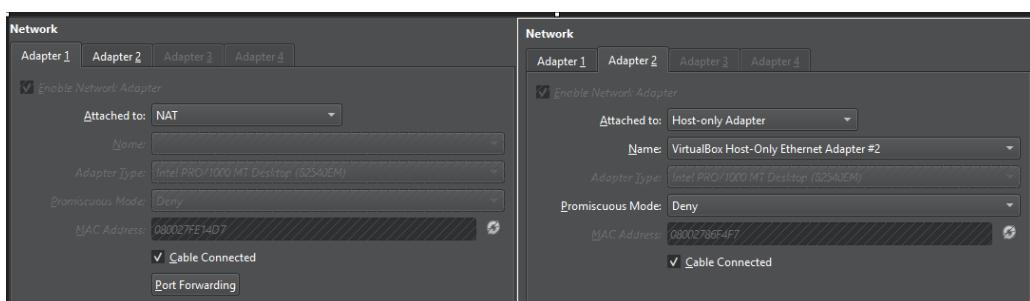


Figure 1: VirtualBox Network Configuration

1.3 Network Configuration, IP Addressing, DHCP and DNS

To ensure stable communication between both CentOS machines, a manual network configuration was applied. Each machine uses the `enp0s8` (Host-Only) interface.

1.3.1 Static IP Configuration on centosServer

The server uses a static IP address to ensure stable Rsyslog and DHCP operation.

```
1 sudo nmcli con mod "System enp0s8" ipv4.method manual \
2 ipv4.address 192.168.56.10/24 \
3 ipv4.gateway 192.168.56.1 \
4 ipv4.dns "8.8.8.8 1.1.1.1"
5
6 sudo nmcli con up "System enp0s8"
```

Listing 1: Server IP Configuration

Verification:

```
1 ip a
2 nmcli con show "System enp0s8"
```

1.3.2 Static IP Configuration on centosClient

```
1 sudo nmcli con mod "System enp0s8" ipv4.method manual \
2 ipv4.address 192.168.56.20/24 \
3 ipv4.gateway 192.168.56.1 \
4 ipv4.dns "8.8.8.8 1.1.1.1"
5
6 sudo nmcli con up "System enp0s8"
```

Listing 2: Client IP Configuration

1.3.3 DHCP Server Installation and Configuration

The DHCP server is configured on **centosServer** to dynamically assign IPs to future clients.

```
1 sudo dnf install dhcp-server -y
```

Listing 3: DHCP Installation

Configuration file: /etc/dhcp/dhcpd.conf

```
1 default-lease-time 600;
2 max-lease-time 7200;
3
4 subnet 192.168.56.0 netmask 255.255.255.0 {
5   range 192.168.56.50 192.168.56.100;
6   option routers 192.168.56.1;
7   option broadcast-address 192.168.56.255;
8   option domain-name-servers 8.8.8.8, 1.1.1.1;
9 }
```

Start the service:

```
1 sudo systemctl enable --now dhcpd
2 sudo systemctl status dhcpd
```

1.3.4 Manual DNS Configuration

The file /etc/resolv.conf was modified to ensure stable DNS resolution:

```
1 sudo nano /etc/resolv.conf
```

Add:

```
1 nameserver 8.8.8.8
2 nameserver 1.1.1.1
```

1.3.5 Connectivity Verification

```
1 ping -c 3 192.168.56.10    # Test from the client to the server
2 ping -c 3 google.com        # DNS resolution test
```

This configuration guarantees reliable communication between the machines for provisioning, centralized logging, and quota management.

2 Use Case Diagram

This section presents the use case diagram showing the interactions between the actors (*Administrator*, *Server*, *Client*) and the main features of the project: automated provisioning, centralized logging, and disk quota management.

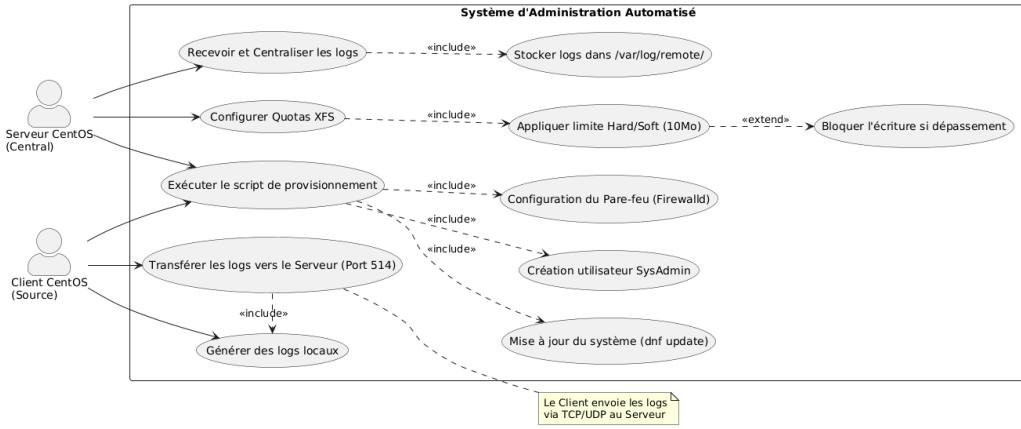


Figure 2: System Use Case Diagram

3 Part 1: Automated Provisioning

To avoid configuration errors and speed up deployment, we created a Bash script: `provision.sh`. It is executed on both machines.

3.1 Script Content

```
1 #!/bin/bash
2 if [ "$EUID" -ne 0 ]; then
3     echo "Please run as root"
4     exit
5 fi
6
7 echo "--- Updating system ---"
8 dnf update -y
9 echo "--- Installing tools ---"
10 dnf install -y vim curl net-tools epel-release
11
12 USERNAME="sysadmin"
13 if id "$USERNAME" &>/dev/null; then
14     echo "User already exists."
15 else
16     useradd -m $USERNAME
17     echo "$USERNAME:StrongPass123" | chpasswd
18     usermod -aG wheel $USERNAME
19 fi
20
21 systemctl start firewalld
22 systemctl enable firewalld
23 firewall-cmd --permanent --add-service=ssh
24 firewall-cmd --reload
```

Listing 4: provision.sh Script

```

client1@vbox ~]$ ./setup.sh
Please run as root
client1@vbox ~]$ sudo -i
root@vbox ~]# cd ..
root@vbox /]# cd home/client1
root@vbox client1]# ls
Desktop Documents Downloads Music Pictures Public setup.sh Templates Videos
root@vbox client1]# ./setup.sh
-- Starting Provisioning Process ---
-- Updating System (this may take time) ---
Last metadata expiration check: 4:31:05 ago on Wed 10 Dec 2025 08:38:04 AM UTC.
Dependencies resolved.
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction

```

Figure 3: Provisioning Script Execution

4 Part 2: Centralized Logging with Rsyslog

4.1 Server Configuration (Receiving Logs)

In `/etc/rsyslog.conf`, we enabled TCP/UDP reception on port 514 and defined a log storage template:

```

1 module(load="imudp")
2 input(type="imudp" port="514")
3 module(load="imtcp")
4 input(type="imtcp" port="514")
5
6 $template RemoteLogs, "/var/log/remote/%HOSTNAME%/%PROGRAMNAME%.log"
7 *.* ?RemoteLogs
8 &

```

4.2 Client Configuration (Sending Logs)

```

1 *.* @192.168.56.10:514

```

4.3 Verification

```

1 [root@centosServer ~]# ls -R /var/log/remote/
2 centosClient/
3 root.log
4
5 cat /var/log/remote/centosClient/root.log

```

```
6 Dec 10 14:45:22 centosClient root: Connection test
7 Dec 10 14:45:35 centosClient root: Log sent to central server
```

5 Part 3: XFS Quota Management

CentOS 9 uses XFS by default, so we used `xfs_quota`. To avoid modifying the root partition, we created a 500MB virtual disk.

5.1 Setting Limits

```
1 xfs_quota -x -c 'limit -u bsoft=9m bhard=10m junior_user' /mnt/myshare
```

5.2 Testing Limits

```
1 dd if=/dev/zero of=testfile bs=1M count=20
2 dd: error writing 'testfile': Disk quota exceeded
```

6 Conclusion

This project allowed us to deploy a complete Linux environment including:

- automated provisioning,
- a professional centralized logging system,
- advanced resource management using XFS quotas.

These skills are essential for administering secure and reliable enterprise infrastructures.

7 References

- Official Rsyslog Documentation: <https://www.rsyslog.com/doc/>
- Red Hat – XFS Quota Management
- CentOS Stream 9 Administration Guide
- Linux Manuals: `man rsyslog`, `man xfs_quota`, `man firewalld`
- VirtualBox Documentation