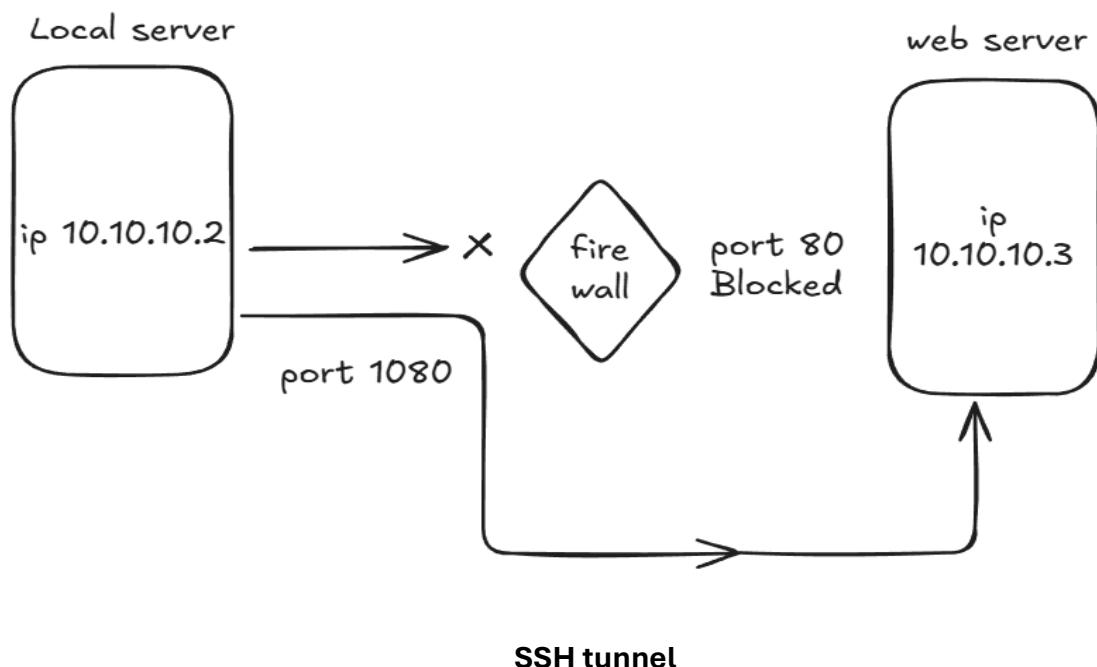


Dynamic Port Forwarding with SSH

Purpose:

Create a SOCKS proxy using SSH to allow applications (Browser, Curl) to route traffic dynamically to any destination through an encrypted SSH tunnel.

Lab



Components

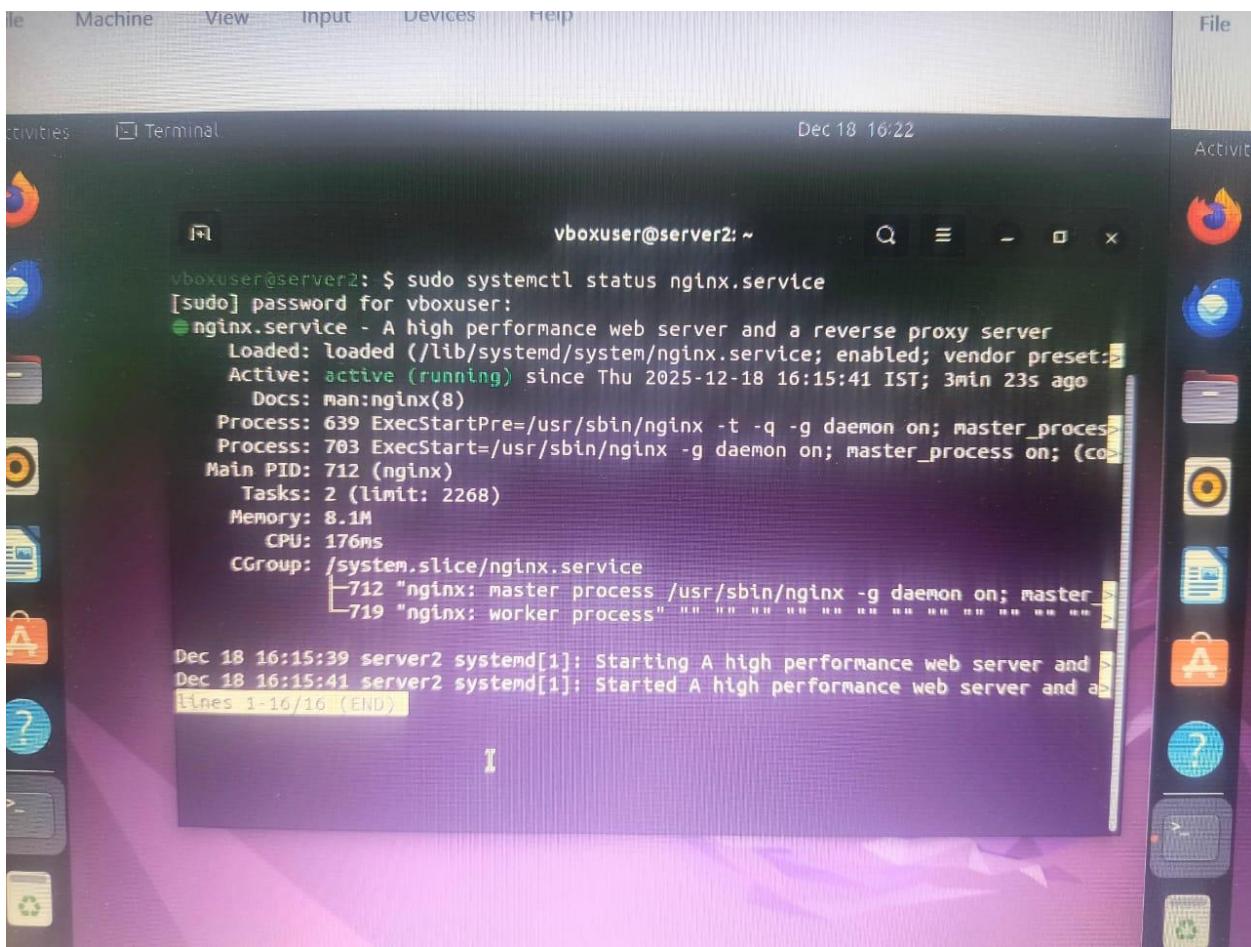
Component	Description
VM1	Local server (Client)
VM2	Remote Web server
Nginx	Web service on port 80
SSH	Service on port 22
Firewall	Port 80 blocked on Web server

Prerequisites

- SSH access to remote server
- Web service running on remote server
- SSH service running
- Port 80 blocked on Web server

Steps

Step 1: Verify Web Service on Client VM (10.10.1.3)



```
sudo systemctl status nginx  
# OR  
sudo systemctl status apache2
```

Check that the web service is running.

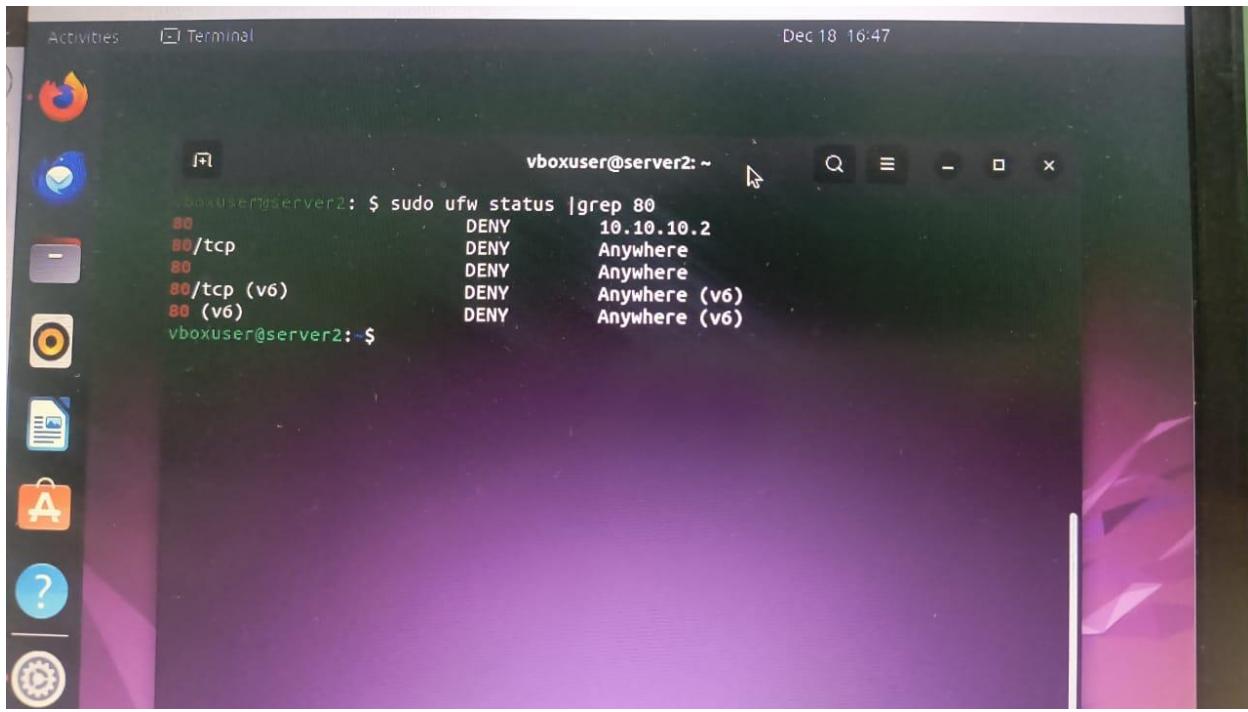
Step 2: Verify SSH Service on VM (10.10.10.2)

The screenshot shows a terminal window titled "vboxuser@server1: ~". The window contains the following command and its output:

```
vboxuser@server1: $ sudo systemctl status ssh  
sshd.service ssh.service ssh.socket  
vboxuser@server1: $ sudo systemctl status sshd.service  
[sudo] password for vboxuser:  
Sorry, try again.  
[sudo] password for vboxuser:  
● ssh.service - OpenBSD Secure Shell server  
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)  
  Active: active (running) since Thu 2025-12-18 16:11:10 IST; 14min ago  
    Docs: man:sshd(8)  
           man:sshd_config(5)  
   Process: 748 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)  
 Main PID: 799 (sshd)  
    Tasks: 1 (limit: 6223)  
   Memory: 3.5M  
      CPU: 90ms  
     CGroup: /system.slice/ssh.service  
             └─799 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"  
  
Dec 18 16:11:09 server1 systemd[1]: Starting OpenBSD Secure Shell server...  
Dec 18 16:11:09 server1 sshd[799]: Server listening on 0.0.0.0 port 22.  
Dec 18 16:11:10 server1 sshd[799]: Server listening on :: port 22.  
Dec 18 16:11:10 server1 systemd[1]: Started OpenBSD Secure Shell server.  
[lines 1-17/17 (END)]
```

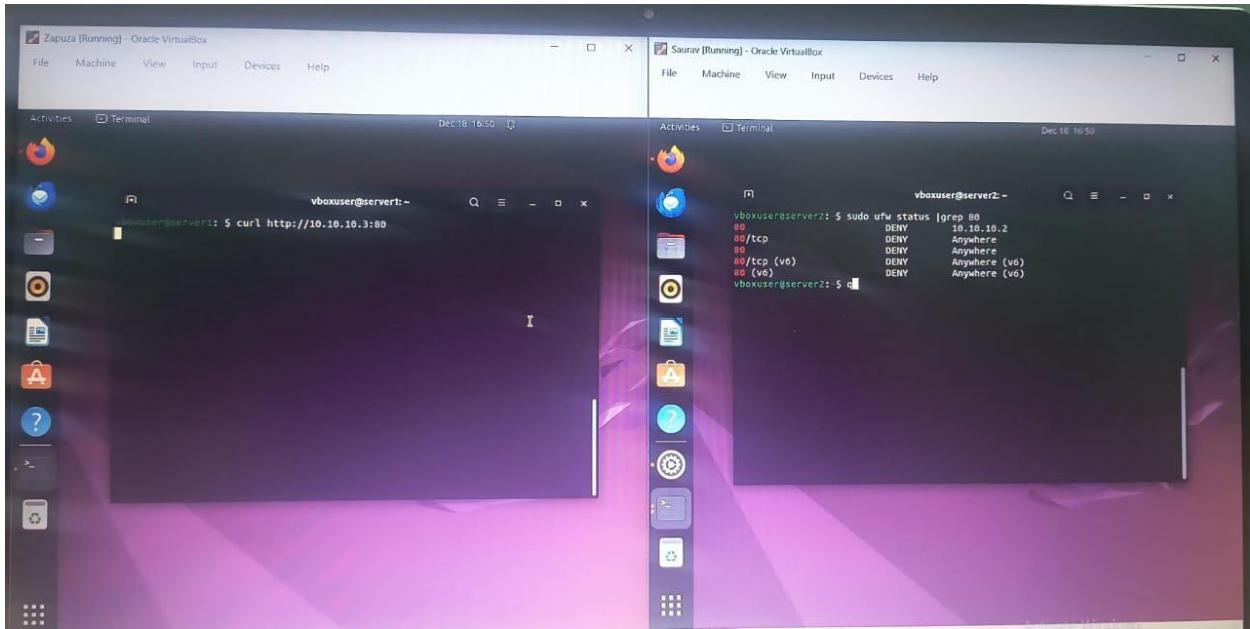
```
sudo systemctl status ssh  
# Check listening port  
sudo ss -tuln | grep :22
```

Step 3: Verify Port 80 Blocked on Web Server



```
sudo ufw status | grep 80
```

Step 4: Direct Access Test (Expected to Fail)

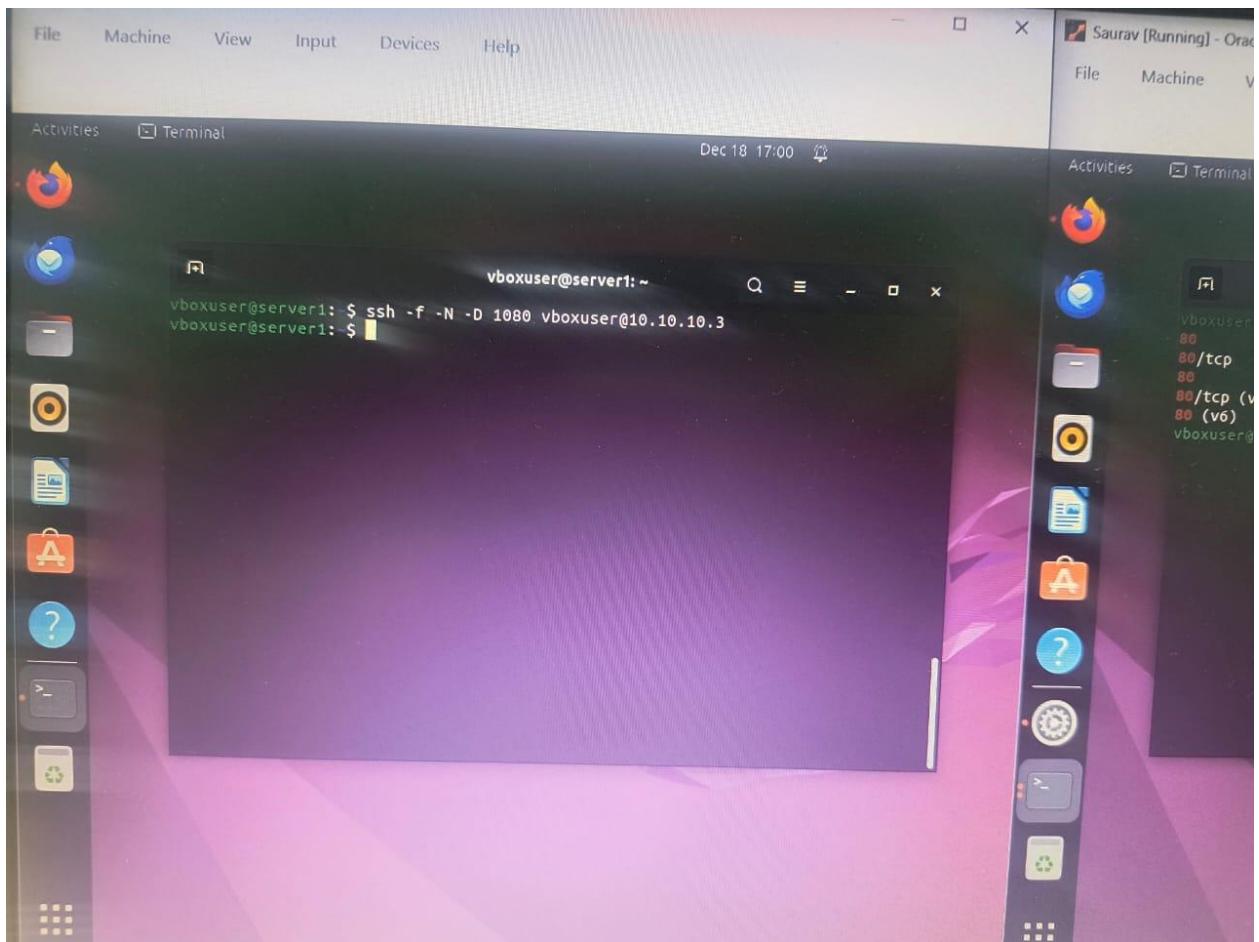


From Server VM (10.10.10.2):

```
curl http://10.10.10.3:80
```

Fails due to firewall restrictions or localhost binding.

Step 5: Create SSH Dynamic Port Forwarding



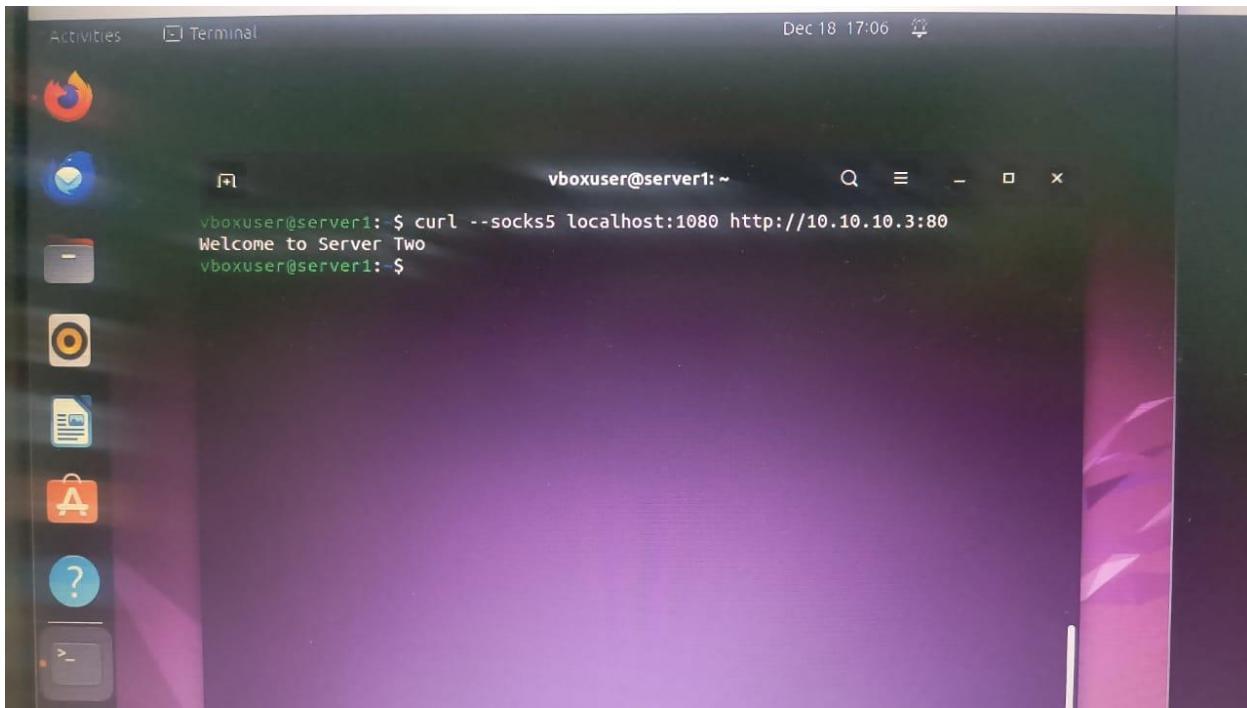
On Client VM (10.10.10.2):

```
ssh -f -N -D 1080 username@Server_IP
```

Flags Explained:

- `-D` → Dynamic port forwarding
- `-f` → Run SSH in background
- `-N` → Do not open remote shell

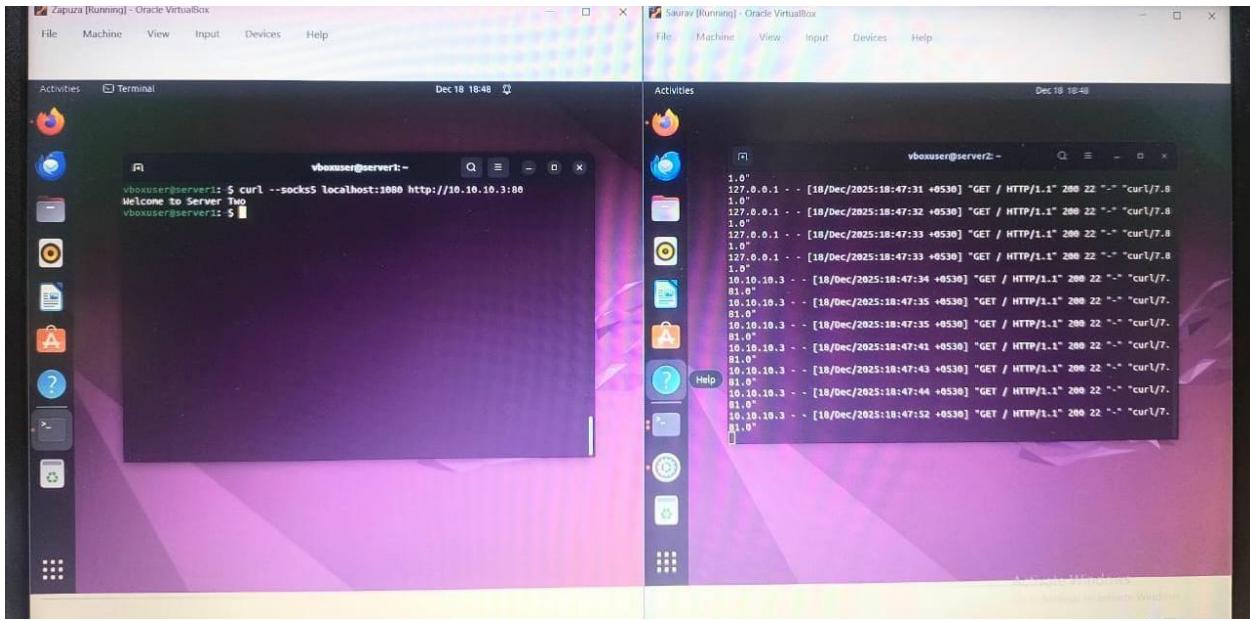
Step 6: Access the Service via Proxy



```
curl --socks5 localhost:1080 http://10.10.10.3:80
```

Or open the browser and set SOCKS5 proxy to localhost:1080.

Step 7: Check Web Server Logs on Client VM



```
sudo tail -f /var/log/nginx/access.log
```

Why 127.0.0.1?

Traffic reaches Nginx through the SSH tunnel, which forwards requests locally on the client VM.