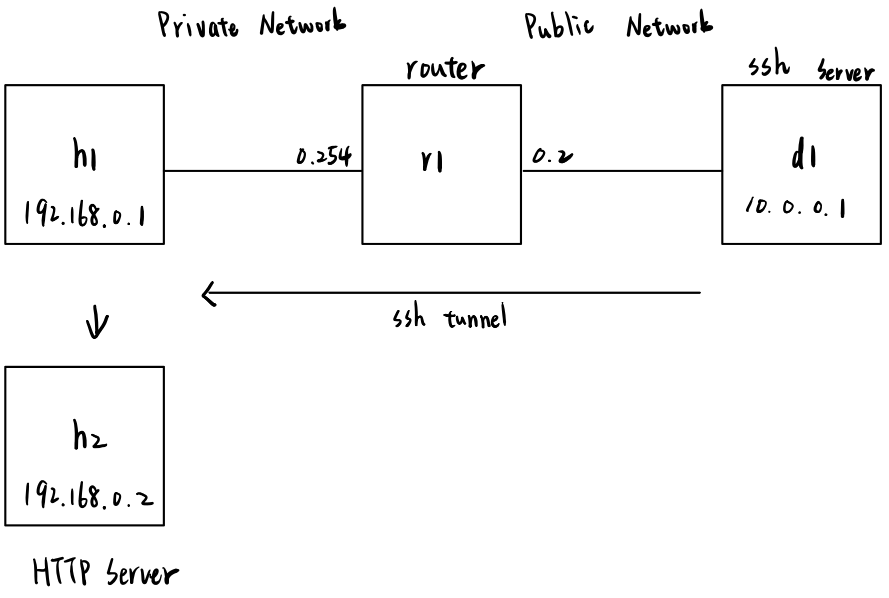
Remote Port Forwarding



```
# gedit lab4.py
# python3 lab4.py
  > xterm h1 h2
```

h2

```
# python -m SimpleHTTPServer 80
```

h1 連到 d1 建立ssh通到連回來,然後跳接到 h2

```
# ssh -Nf -R 10.0.0.1:5555:192.168.0.2:80 user@10.0.0.1
```

返向　　跳接的位置

user

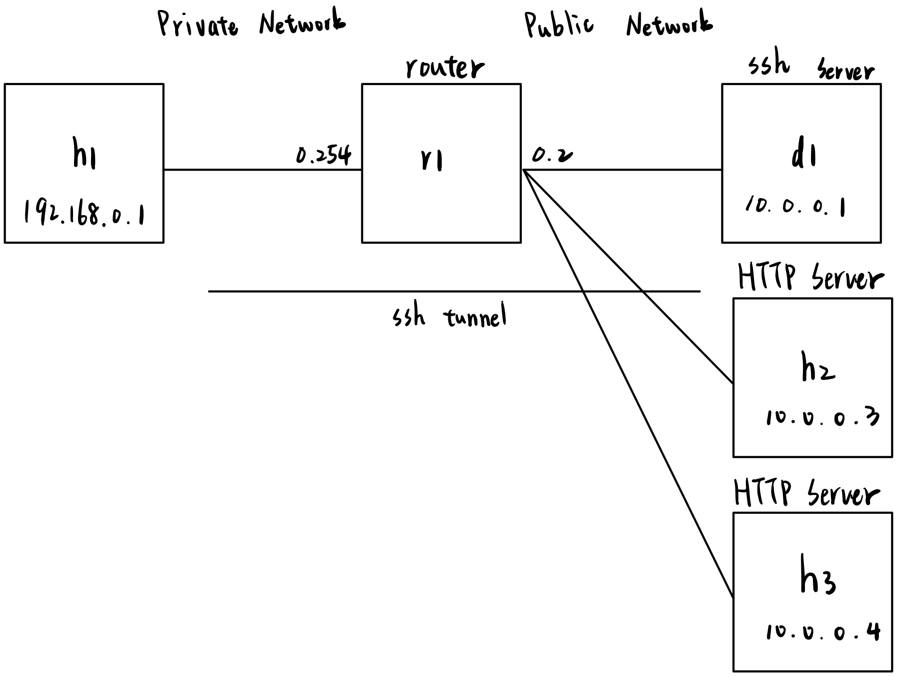
terminal d1

```
# docker exec -it mn.d1 bash
  /# curl 127.0.0.1:5555
```

ssh 指定 port

```
# ssh root@192.168.0.1 -p25
```

Dynamic Port Forwarding



```
# gedit lab5.py

# python3 lab5.py

  > xterm h1 h2 h3
```

h2

```
# python -m SimpleHTTPServer 80
```

h3

```
# python -m SimpleHTTPServer 8080
```

h1

动態

```
# ssh -Nf -D 127.0.0.1:8080 user@10.0.0.1
```

user

```
# su -user
# firefox
```

firefox - 三 - Preference - Network Settings - OK

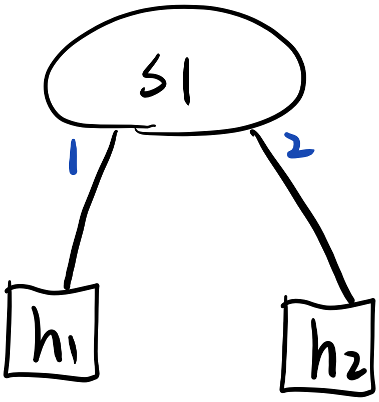
h2-ip h3-ip

firefox search 10.0.0.3 , 10.0.0.4:8080

SDN - Mininet、OVS、P4 switch

data plan

Control plan



```
# mn --topo single,2
```

new terminal

```
# ps -ef | grep controller
```

查詢交換机內的控制表格

```
> sh ovs-ofctl dump-flows s1
> h1 ping -c 3 h2
  X
                      ovs-ofctl
```

設定規則讓 h1 . h2 可以互ping

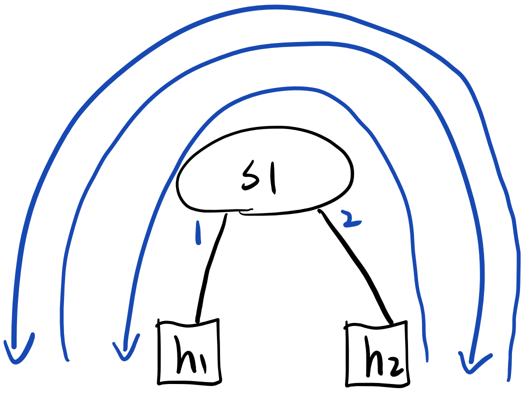
1號進來 執行动作 2號出去

```
> sh ovs-ofctl add-flow s1 in_port=1, actions=output:2
> sh ovs-ofctl add-flow s1 in_port=2, actions=output:1
```

```
> sh ovs-ofctl dump-flows s1
> h1 ping -c 3 h2
  √
> sh ovs-ofctl dump-flows s1
```

n_packet=3

符合規則的封包数



1. arp request
2. arp reply
3. icmp request

刪除全部規則

```
> sh ovs-ofctl del-flows s1
> sh ovs-ofctl dump-flows s1
> h1 ping -c 3 h2
  X
```

控制器在遠端，沒啟动

```
# mn --topo single,2 --controller =remote
# mn --topo single,2
```

一樣的

只是第一個就不用再kill了