1. **建立instance**

两个instance: og\_teamXX\_vm1 og\_teamXX\_vm2 ,建立instanc和修改config，加上这两个host，运行两个VM

Screens screenshot of a computer screen

Description automatically generated

1. **完成Communicate**

* 设置好vxlan
* 然后根据vxlan0当中的设置，在输入sudo ip route add 10.10.2.0/24 via xxx.xxx.xxx.xxx/24 dev vxlan0，然后才能h1 ping 进行测试
* 根据week10 lecture- Creating VXlan tunnel.，设置VxLan，然后才能在VM1中的mininet 中h1 ping 10.10.2.2，然后另一边h2 ping 10.10.1.1，这个ping成功，那后面client和server，只需要在vm1输入sudo mn - -nat -i 10.10.2.0/24，在进入的mininet中输入h1 xx.py -a 10.10.2.2 -p 5555就能communicate

图中这步骤不要做

A screenshot of a computer screen

Description automatically generated

Milestone 1 report

So far, we finished some parts of the milestone 1.

- [x] 2 VMs are created on Chameleon

- [x] VxLANs are set for each VM

- [x] a very simple topology (the triangle in the document) is created using Mininet programming APIs

Other steps to finish (not sure how to do these):

- [ ] test and confirm the connectivity between two VMs through VxLANs

- [ ] "mesh the Mininet topology in one VM with the Mininet topology in the second VM"

Commands used for Milestone 1

On VM1:

sudo ip link add vxlan0 type vxlan id 100 local 192.168.5.115 remote 192.168.5.84 dev ens3 dstport 4789

sudo ip addr add 192.168.100.1/24 dev vxlan0

sudo ip link set vxlan0 up

ifconfig

sudo ip route add 10.10.2.0/24 via 192.168.100.2/24 dev vxlan0

sudo ip link set vxlan0 up

sudo mn --nat -i 10.10.2.0/24

On VM2:

sudo ip link add vxlan0 type vxlan id 100 local 192.168.5.84 remote 192.168.5.115 dev ens3 dstport 4789

sudo ip addr add 192.168.100.2/24 dev vxlan0

sudo ip link set vxlan0 up

ifconfig

sudo ip route add 10.10.1.0/24 via 192.168.100.1/24 dev vxlan0

sudo ip link set vxlan0 up

sudo mn --nat -i 10.10.1.0/24

# Other information

Reference for running custom python api for mininets

https://stackoverflow.com/questions/44921474/how-to-run-python-code-in-mininet