

---

# 《虚拟化与云计算实践》

(2024-2025 学年第 2 学期)

## 实验三

---



姓名	学号	班级
江欣怡	2022337621242	22 计科 2 班
周洪蕊	2022337621245	22 计科 2 班
祝云佳	2022337621086	22 计科 3 班

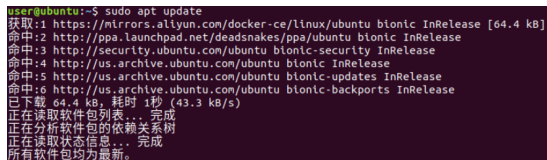
2025 年 6 月 6 日

---

### 1 使用 Python 3.8 创建虚拟环境并安装 Ryu

#### 1. 安装系统依赖:

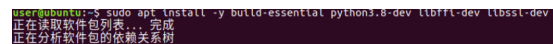
```
sudo apt update  
sudo apt install -y build-essential python3.8-dev libffi-dev  
libssl-dev
```



user@ubuntu:~\$ sudo apt update  
获取:1 https://mirrors.aliyun.com/docker-ce/linux/ubuntu bionic InRelease [64.4 kB]  
命中:2 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu bionic InRelease  
命中:3 http://security.ubuntu.com/ubuntu bionic-security InRelease  
命中:4 http://us.archive.ubuntu.com/ubuntu bionic InRelease  
命中:5 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease  
命中:6 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease  
已下载 64.4 kB, 耗时 1秒 (43.3 kB/s)  
正在读取软件包列表... 完成  
正在分析软件包的依赖关系树  
正在读取状态信息... 完成  
所有软件包均均为最新。

图 1: \*

执行 sudo apt update



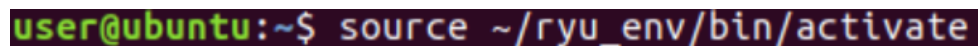
user@ubuntu:~\$ sudo apt install -y build-essential python3.8-dev libffi-dev libssl-dev  
正在读取软件包列表... 完成  
正在分析软件包的依赖关系树

图 2: \*

执行 apt install 安装依赖

#### 2. 进入虚拟环境:

```
source ~/ryu_env/bin/activate
```

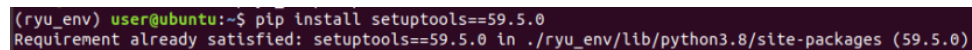


user@ubuntu:~\$ source ~/ryu\_env/bin/activate

图 3: 激活虚拟环境的终端截图

#### 3. 降级 setuptools 到兼容版本

```
pip install setuptools==59.5.0
```

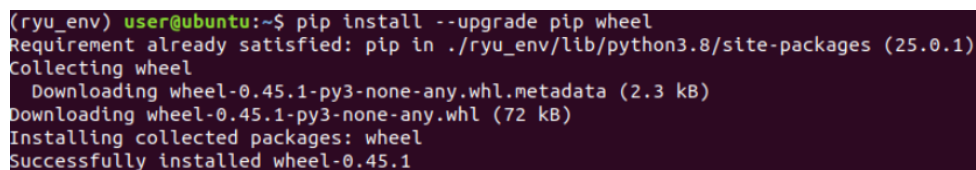


(ryu\_env) user@ubuntu:~\$ pip install setuptools==59.5.0  
Requirement already satisfied: setuptools==59.5.0 in ./ryu\_env/lib/python3.8/site-packages (59.5.0)

图 4: 降级 setuptools 到 59.5.0

#### 4. 升级 pip 和 wheel

```
pip install --upgrade pip wheel
```



(ryu\_env) user@ubuntu:~\$ pip install --upgrade pip wheel  
Requirement already satisfied: pip in ./ryu\_env/lib/python3.8/site-packages (25.0.1)  
Collecting wheel  
 Downloading wheel-0.45.1-py3-none-any.whl.metadata (2.3 kB)  
 Downloading wheel-0.45.1-py3-none-any.whl (72 kB)  
Installing collected packages: wheel  
Successfully installed wheel-0.45.1

图 5: 升级 pip 与 wheel

## 1 使用 PYTHON 3.8 创建虚拟环境并安装 RYU

### 5. 卸载当前 eventlet，安装兼容的 eventlet 版本

```
pip uninstall eventlet  
pip install eventlet==0.30.2
```

```
(ryu_env) user@ubuntu:~$ pip uninstall eventlet  
Found existing installation: eventlet 0.39.1  
Uninstalling eventlet-0.39.1:  
  Would remove:  
    /home/user/ryu_env/lib/python3.8/site-packages/eventlet-0.39.1.dist-info/*  
    /home/user/ryu_env/lib/python3.8/site-packages/eventlet/*  
  Proceed (Y/n)? Y  
  Successfully uninstalled eventlet-0.39.1  
(ryu_env) user@ubuntu:~$ pip install eventlet==0.30.2  
Collecting eventlet==0.30.2  
  Downloading eventlet-0.30.2-py2.py3-none-any.whl.metadata (4.1 kB)  
Collecting dnspython<2.0.0,>=1.15.0 (from eventlet==0.30.2)  
  Downloading dnspython-1.16.0-py2.py3-none-any.whl.metadata (1.8 kB)  
Requirement already satisfied: greenlet>=0.3 in ./ryu_env/lib/python3.8/site-packages (from eventlet==0.30.2) (3.1.1)  
Requirement already satisfied: six>=1.10.0 in ./ryu_env/lib/python3.8/site-packages (from eventlet==0.30.2) (1.17.0)  
Downloading eventlet-0.30.2-py2.py3-none-any.whl (224 kB)  
Downloading dnspython-1.16.0-py2.py3-none-any.whl (188 kB)  
Installing collected packages: dnspython, eventlet  
  Attempting uninstall: dnspython  
    Found existing installation: dnspython 2.6.1  
    Uninstalling dnspython-2.6.1:  
      Successfully uninstalled dnspython-2.6.1  
  Successfully installed dnspython-1.16.0 eventlet-0.30.2
```

图 6: 卸载并安装兼容的 eventlet 版本

### 6. 安装 ryu

```
pip install ryu
```

```
(ryu_env) user@ubuntu:~$ pip install ryu  
Collecting ryu  
  Using cached ryu-4.34.tar.gz (1.1 MB)  
  Preparing metadata (setup.py) ... done  
Collecting eventlet!=0.18.3,!0.20.1,!0.21.0,!0.23.0,>=0.18.2 (from ryu)  
  Downloading eventlet-0.39.1-py3-none-any.whl.metadata (5.5 kB)  
Collecting msgpack>=0.3.0 (from ryu)  
  Downloading msgpack-1.1.0-cp38-cp38-manylinux_2_17_x86_64_manylinux2014_x86_64.whl.metadata (8.4 kB)  
Collecting netaddr (from ryu)  
  Downloading netaddr-1.3.0-py3-none-any.whl.metadata (5.0 kB)  
Collecting oslo.config>=2.5.0 (from ryu)  
  Downloading oslo.config-9.6.0-py3-none-any.whl.metadata (3.0 kB)  
Collecting ovs>=2.6.0 (from ryu)  
  Downloading ovs-3.5.1.tar.gz (161 kB)  
  Preparing metadata (setup.py) ... done  
Collecting routes (from ryu)  
  Downloading Routes-2.5.1-py2.py3-none-any.whl.metadata (25 kB)  
Collecting six>=1.4.0 (from ryu)  
  Downloading six-1.17.0-py2.py3-none-any.whl.metadata (1.7 kB)  
Collecting tinyrpc (from ryu)  
  Downloading tinyrpc-1.1.7-py3-none-any.whl.metadata (6.7 kB)  
Collecting webob>=1.2 (from ryu)  
  Downloading WebOb-1.8.9-py2.py3-none-any.whl.metadata (11 kB)  
Collecting dnspython>=1.15.0 (from eventlet!=0.18.3,!0.20.1,!0.21.0,!0.23.0,>=0.18.2->ryu)  
  Downloading dnspython-2.6.1-py3-none-any.whl.metadata (5.8 kB)  
Collecting greenlet>=1.0 (from eventlet!=0.18.3,!0.20.1,!0.21.0,!0.23.0,>=0.18.2->ryu)  
  Downloading greenlet-3.1.1-cp38-cp38-manylinux_2_24_x86_64_manylinux2014_x86_64.whl.metadata (3.8 kB)  
Collecting PyYAML>=5.1 (from oslo.config>=2.5.0->ryu)  
  Downloading PyYAML-6.0.2-cp38-cp38-manylinux_2_17_x86_64_manylinux2014_x86_64.whl.metadata (2.1 kB)  
Collecting debtcollector>=1.2.0 (from oslo.config>=2.5.0->ryu)
```

图 7: 安装 Ryu 控制器

### 7. 后续用完，退出虚拟环境:

```
deactivate
```

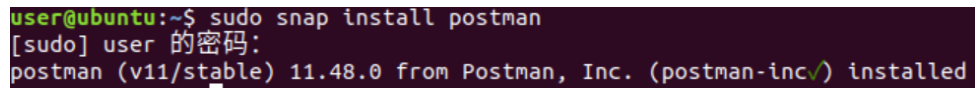
再次打开虚拟环境 (ryu):

```
source ~/ryu_env/bin/activate
```

## 2 安装 Postman

通过 Snap 安装:

```
sudo snap install postman
```



```
user@ubuntu:~$ sudo snap install postman
[sudo] user 的密码:
postman (v11/stable) 11.48.0 from Postman, Inc. (postman-inc✓) installed
```

图 8: 通过 Snap 成功安装 Postman 的终端截图

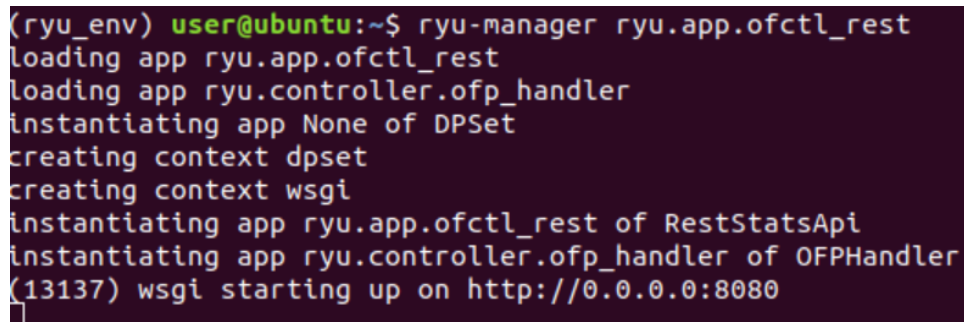
安装后点击“Show Applications”，搜索“Postman”启动应用。

## 3 运行控制器与拓扑

1. 启动 Ryu 控制器（终端 1）:

```
ryu-manager ryu.app.ofctl_rest
```

此应用提供了对交换机流表的 REST 控制接口，默认监听端口为 127.0.0.1:8080。



```
(ryu_env) user@ubuntu:~$ ryu-manager ryu.app.ofctl_rest
loading app ryu.app.ofctl_rest
loading app ryu.controller.ofp_handler
instantiating app None of DPSet
creating context dpset
creating context wsgi
instantiating app ryu.app.ofctl_rest of RestStatsApi
instantiating app ryu.controller.ofp_handler of OFPHandler
(13137) wsgi starting up on http://0.0.0.0:8080
```

图 9: 启动 Ryu 控制器的终端截图

2. 启动 Mininet 拓扑（终端 2）:

```
sudo mn --controller=remote,ip=127.0.0.1,port=6653 --switch ovs,
      protocols=OpenFlow13
```

```
user@ubuntu:~$ sudo mn --controller=remote,ip=127.0.0.1,port=6653 --switch ovs,protocols=OpenFlow13
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> sh ovs-ofctl -O OpenFlow13 dump-flows s1
mininet> h1 ping h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data:
From 10.0.0.1 icmp_seq=1 Destination Host Unreachable
From 10.0.0.1 icmp_seq=2 Destination Host Unreachable
From 10.0.0.1 icmp_seq=3 Destination Host Unreachable
^C
--- 10.0.0.2 ping statistics ---
6 packets transmitted, 0 received, +3 errors, 100% packet loss, time 5109ms
pipe 4
```

图 10: 启动 Mininet 网络拓扑的终端截图

此时尝试在 CLI 中执行：

```
h1 ping h2
```

会发现 h1 与 h2 无法通信。

## 4 使用 Postman 控制流表

### 1. 添加流表项

打开 Postman，配置如下请求：

**Method:** POST

**URL:** `http://127.0.0.1:8080/stats/flowentry/add`

**Body** → raw → JSON:

```
{
  "dpid": 1,
  "priority": 200,
  "match": {
    "in_port": 1
  },
  "actions": [
    {
      "type": "OUTPUT",
```

```
    "port": 2
  }
]
}
```

继续发送另一条规则 (in\_port 2 → out\_port 1):

```
{
  "dpid": 1,
  "priority": 200,
  "match": {
    "in_port": 2
  },
  "actions": [
    {
      "type": "OUTPUT",
      "port": 1
    }
  ]
}
```

## 4 使用 POSTMAN 控制流表

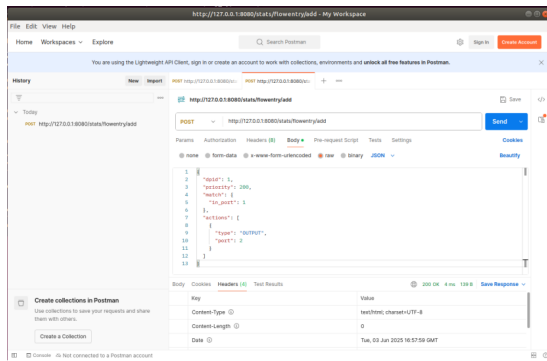


图 11: \*

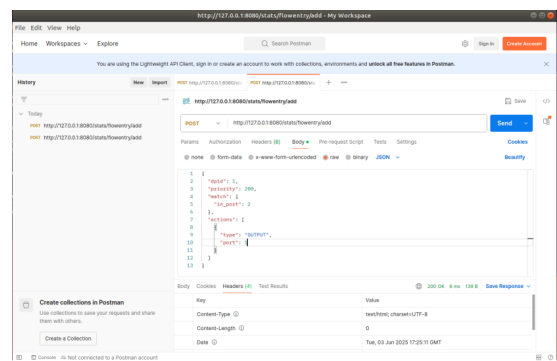


图 12: \*

下发 in\_port 1 → out\_port 2

下发 in\_port 2 → out\_port 1

```
(ryu_env) user@ubuntu:~$ ryu-manager ryu.app.ofctl_rest
loading app ryu.app.ofctl_rest
loading app ryu.controller.ofp_handler
instantiating app None of DPSet
creating context dpset
creating context wsgi
instantiating app ryu.app.ofctl_rest of RestStatsApi
instantiating app ryu.controller.ofp_handler of OFPHandler
(13137) wsgi starting up on http://0.0.0.0:8080
(13137) accepted ('127.0.0.1', 57466)
127.0.0.1 - - [03/Jun/2025 10:25:11] "POST /stats/flowentry/add HTTP/1.1" 200 139 0.003313
127.0.0.1 - - [03/Jun/2025 10:25:28] "POST /stats/flowentry/add HTTP/1.1" 200 139 0.000458
```

图 13: \*

Ryu 返回 200 OK

## 2. 再次测试通信

回到 Mininet CLI:

```
pingall
```

可以看到主机之间已能通信，说明流表生效。

## 4 使用 POSTMAN 控制流表

```
user@ubuntu:~$ sudo mn --controller=remote,ip=127.0.0.1,port=6653 --switch ovs,protocols=OpenFlow13
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> sh ovs-ofctl -O OpenFlow13 dump-flows s1
mininet> h1 ping h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
From 10.0.0.1 icmp_seq=1 Destination Host Unreachable
From 10.0.0.1 icmp_seq=2 Destination Host Unreachable
From 10.0.0.1 icmp_seq=3 Destination Host Unreachable
^C
--- 10.0.0.2 ping statistics ---
6 packets transmitted, 0 received, +3 errors, 100% packet loss, time 5109ms
pipe 4
mininet> sh ovs-ofctl -O OpenFlow13 dump-flows s1
cookie=0x0, duration=37.978s, table=0, n_packets=0, n_bytes=0, priority=200,in_port="s1-eth1" actions=0
output:"s1-eth2"
cookie=0x0, duration=20.718s, table=0, n_packets=0, n_bytes=0, priority=200,in_port="s1-eth2" actions=0
output:"s1-eth1"
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
```

图 14: 成功配置流表后，主机通信正常

### 3. 查看当前流表

Method: GET

URL: <http://127.0.0.1:8080/stats/flow/1>

可以看到当前交换机中的所有流表项信息。

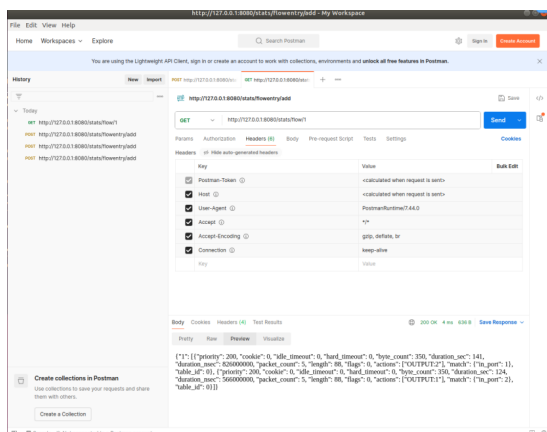


图 15: \*  
请求查看流表

```
(ryu_emu) user@ubuntu:~$ ryu-manager ryu.app.ofctl_rest
loading app ryu.app.ofctl_rest
loading app ryu.controller.ofp_handler
instantiating app None of DPSet
creating context dpset
instantiating app ryu.app.ofctl_rest of RestStatsApi
instantiating app ryu.controller.ofp_handler of OFHandler
(13137) wsgiref starting up on http://0.0.0.0:8080
(13137) accepted ('127.0.0.1', 57466)
127.0.0.1 - - [03/Jun/2025 10:25:11] "POST /stats/flowentry/add HTTP/1.1" 200 139 0.003113
127.0.0.1 - - [03/Jun/2025 10:25:28] "POST /stats/flowentry/add HTTP/1.1" 200 139 0.000458
(13137) accepted ('127.0.0.1', 54826)
127.0.0.1 - - [03/Jun/2025 10:27:33] "GET /stats/flow/1 HTTP/1.1" 200 636 0.001134
(13137) accepted ('127.0.0.1', 54912)
127.0.0.1 - - [03/Jun/2025 10:28:56] "DELETE /stats/flowentry/clear/1 HTTP/1.1" 200 139 0.000380
```

图 16: \*  
Ryu 返回 200 OK

### 4. 清除流表



### Method: DELETE

URL: `http://127.0.0.1:8080/stats/flowentry/clear/1`

流表已清空。

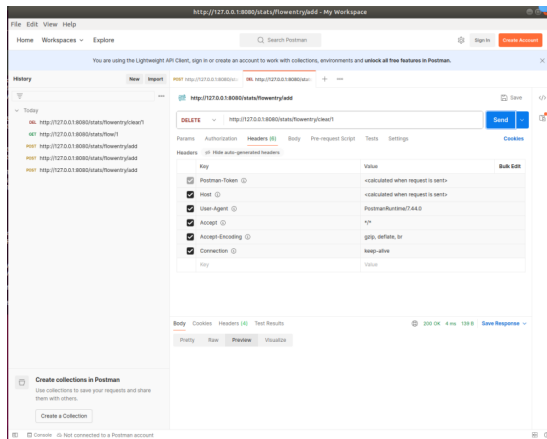


图 17: \*

向交换机发送清除流表请求

```
(ryu_env) user@ubuntu:~$ ryu-manager ryu.app.ofctl_rest
loading app ryu.app.ofctl_rest
loading app ryu.controller.ofp_handler
Instantiating app None of DPSet
creating context dpset
Instantiating app ryu.app.ofctl_rest of RestStatsApi
Instantiating app ryu.controller.ofp_handler of OFPHandler
(13137) wsgi starting up on http://0.0.0.0:8080
(13137) accepted ('127.0.0.1', 57466)
127.0.0.1 - - [03/Jun/2025 10:25:11] "POST /stats/flowentry/add HTTP/1.1" 200 139 0.003313
127.0.0.1 - - [03/Jun/2025 10:25:28] "POST /stats/flowentry/add HTTP/1.1" 200 139 0.000458
(13137) accepted ('127.0.0.1', 54826)
127.0.0.1 - - [03/Jun/2025 10:27:33] "GET /stats/flow/1 HTTP/1.1" 200 636 0.001134
(13137) accepted ('127.0.0.1', 32972)
127.0.0.1 - - [03/Jun/2025 10:28:50] "DELETE /stats/flowentry/clear/1 HTTP/1.1" 200 139 0.000300
```

图 18: \*

Ryu 返回 200 OK

发送后再次执行 pingall，发现通信中断。

```
user@ubuntu:~$ sudo mn --controller=remote,ip=127.0.0.1,port=6653 --switch ovs,protocols=OpenFlow13
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> sh ovs-ofctl -O OpenFlow13 dump-flows s1
mininet> h1 ping h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
From 10.0.0.1 icmp_seq=1 Destination Host Unreachable
From 10.0.0.1 icmp_seq=2 Destination Host Unreachable
From 10.0.0.1 icmp_seq=3 Destination Host Unreachable
^C
--- 10.0.0.2 ping statistics ---
6 packets transmitted, 0 received, +3 errors, 100% packet loss, time 5109ms
pipe 4
mininet> sh ovs-ofctl -O OpenFlow13 dump-flows s1
 cookie=0x0, duration=37.978s, table=0, n_packets=0, n_bytes=0, priority=200,in_port="s1-eth1" actions=output:"s1-eth2"
 cookie=0x0, duration=20.718s, table=0, n_packets=0, n_bytes=0, priority=200,in_port="s1-eth2" actions=output:"s1-eth1"
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet> pingall
*** Ping: testing ping reachability
h1 -> X
h2 -> X
*** Results: 100% dropped (0/2 received)
mininet>
```

图 19: 通过 Postman 清空交换机流表后通信中断