

OpenvSwitch Production Ready Install & Commands

Step 1: Update your Raspberry Pi OS

```
sudo apt update && sudo apt upgrade -y
```

Step 2: Install Dependencies

```
sudo apt install -y git build-essential fakeroot \  
  libssl-dev python3 python3-pip python3-six \  
  python3-setuptools python3-wheel autoconf \  
  automake libtool
```

```
sudo apt install -y dkms
```

Step 3: Install Open vSwitch from Official Repository (Simpler Method)

```
sudo apt install -y openvswitch-switch openvswitch-common
```

```
ovs-vsctl --version
```

or

<https://www.openvswitch.org/releases/openvswitch-3.3.5.tar.gz>

Step 4: Start and Enable the Open vSwitch Service

```
sudo systemctl start openvswitch-switch  
sudo systemctl enable openvswitch-switch
```

Optional: Build Open vSwitch from Source (if needed)

```
git clone https://github.com/openvswitch/ovs.git  
cd ovs  
git checkout <stable_version> # e.g., branch: v3.2.1
```

```
./boot.sh  
./configure  
make  
sudo make install  
sudo /usr/local/share/openvswitch/scripts/ovs-ctl start
```

Final Checks

```
sudo ovs-vsctl show  
ip addr
```

```
sudo systemctl status openvswitch-switch
```

1. Basic OVS Control Commands

```
sudo ovs-vsctl show
sudo ovs-vsctl list-br
sudo ovs-vsctl add-br br0
sudo ovs-vsctl del-br br0
sudo ovs-vsctl add-port br0 eth0
sudo ovs-vsctl del-port br0 eth0
sudo ovs-vsctl list-ports br0
sudo ovs-vsctl list interface
```

2. Interface and Port Management

Makes an interface internal (used for local VM traffic).

```
sudo ovs-vsctl add-port br0 veth1 -- set interface veth1 type=internal
```

Get OpenFlow port number for interface.

```
sudo ovs-vsctl get interface eth0 ofport
```

3. IP Configuration (Linux-level)

Assign static IP to bridge

```
sudo ip addr flush dev eth0
sudo ip addr add 192.168.1.100/24 dev br0
sudo ip link set br0 up
```

Request IP via DHCP

```
sudo dhclient br0
```

4. OpenFlow Commands

Show current OpenFlow rules.

```
sudo ovs-ofctl dump-flows br0
```

Add a manual flow rule.

```
sudo ovs-ofctl add-flow br0 "in_port=1,actions=output:2"
```

Deletes all flows on the bridge.

```
sudo ovs-ofctl del-flows br0
```

5. VLAN Tagging and Trunking

Access Port (untagged VLAN)

```
sudo ovs-vsctl set port eth0 tag=10
```

Trunk Port (multiple VLANs)

```
sudo ovs-vsctl set port eth0 trunks=10,20,30
```

6. Bonding Interfaces (LACP)

Create Bond

```
sudo ovs-vsctl add-bond br0 bond0 eth0 eth1
```

Enable LACP (Link Aggregation Control Protocol)

```
sudo ovs-vsctl set port bond0 lacp=active
```

7. Persistent Configuration (Debian/RPi OS)

To persist across reboots, add in:

```
sudo nano /etc/network/interfaces
```

```
auto br0
iface br0 inet dhcp
    bridge_ports eth0
```

8. Monitoring and Debugging

Check Open vSwitch status

```
sudo systemctl status openvswitch-switch
```

Restart OVS

```
sudo systemctl restart openvswitch-switch
```

Log output (if something goes wrong)

```
sudo journalctl -u openvswitch-switch
```

```
ifconfig eth0 0
```

```
dhclient br0
```

```
ifconfig
```

```
route -n
```

adding virtual port

```
ip tuntap add mode tap vport1
```

```
ip tuntap add mode tap vport2
```

add that port to br0

```
ovs-vsctl add-port br0 vport1 -- add-port br0 vport2
```

```
ovs-vsctl show
```

check mac address on br0

```
ovs-appctl fdb/show
```

check openflow br0

```
ovs-ofctl show br0
```

```
ovs-ofctl dump-flows br0
```

```
ovs-vsctl list br0
```

```
ovs-vsctl list Port | more
```

```
ovs-vsctl list Interface | more
```

