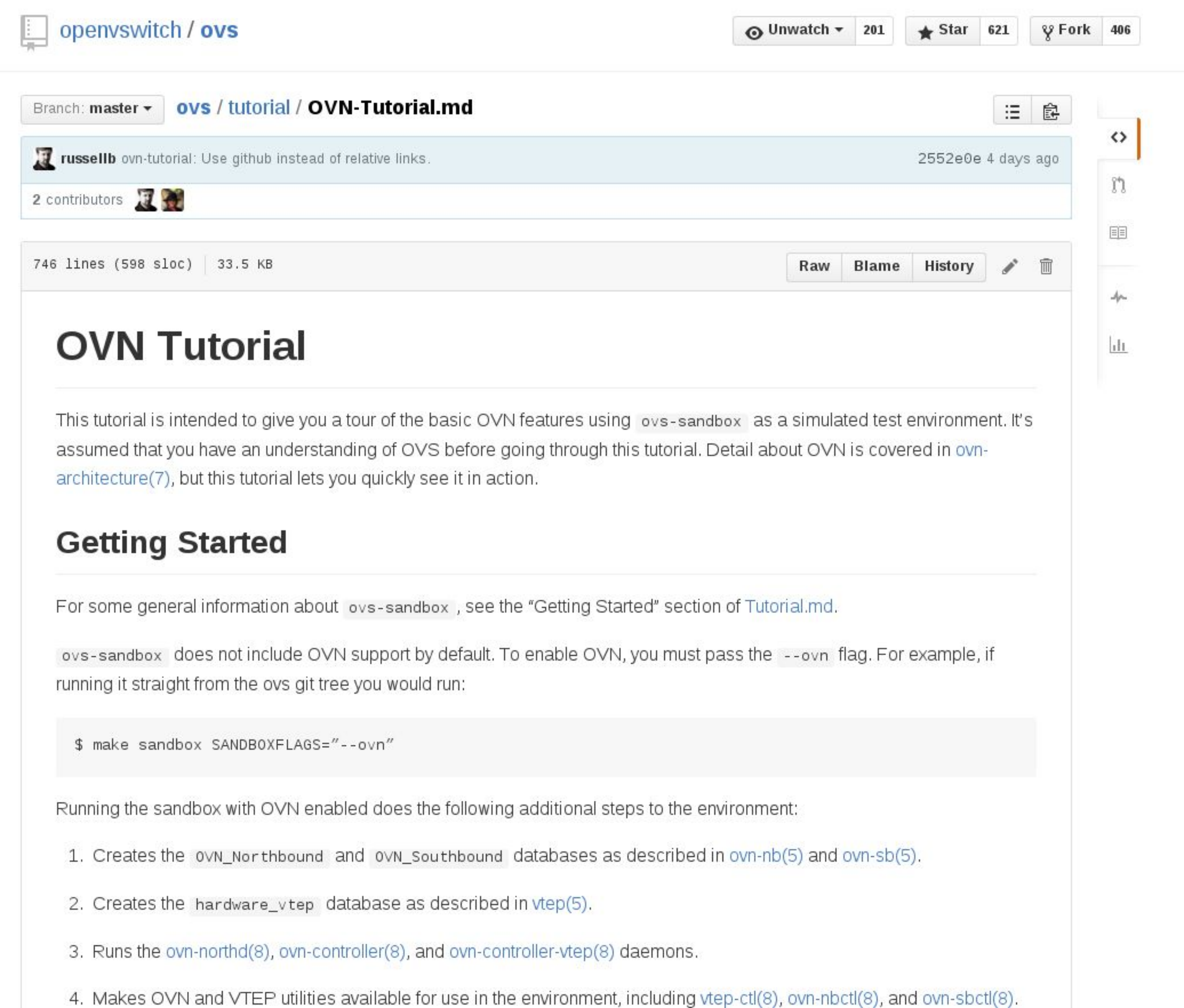
OVN in a sandbox

Russell Bryant (@russellbryant)



ovs / tutorial / OVN-Tutorial.md

russellb ovn-tutorial: Use github instead of relative links. 2552e0e 4 days ago

2 contributors

746 lines (598 sloc) 33.5 KB

OVN Tutorial

This tutorial is intended to give you a tour of the basic OVN features using `ovs-sandbox` as a simulated test environment. It's assumed that you have an understanding of OVS before going through this tutorial. Detail about OVN is covered in ovn-architecture(7), but this tutorial lets you quickly see it in action.

Getting Started

For some general information about `ovs-sandbox`, see the "Getting Started" section of Tutorial.md.

`ovs-sandbox` does not include OVN support by default. To enable OVN, you must pass the `--ovn` flag. For example, if running it straight from the ovs git tree you would run:

```
$ make sandbox SANDBOXFLAGS="--ovn"
```

Running the sandbox with OVN enabled does the following additional steps to the environment:

1. Creates the `OVN_Northbound` and `OVN_Southbound` databases as described in ovn-nb(5) and ovn-sb(5).
2. Creates the `hardware_vtep` database as described in vtep(5).
3. Runs the ovn-northd(8), ovn-controller(8), and ovn-controller-vtep(8) daemons.
4. Makes OVN and VTEP utilities available for use in the environment, including vtep-ctl(8), ovn-nbctl(8), and ovn-sbctl(8).

ovs-sandbox

```
$ make sandbox

…

You are running in a dummy Open vSwitch environment.  You can use
ovs-vsctl, ovs-ofctl, ovs-appctl, and other tools to work with the
dummy switch.

Log files, pidfiles, and the configuration database are in the
"sandbox" subdirectory.

Exit the shell to kill the running daemons.
```

ovs-sandbox with OVN enabled

```
$ make sandbox SANDBOXFLAGS=”--ovn”
```

Enables:

- DBs: OVN_Northbound, OVN_Southbound, hardware_vtep
- Daemons: ovn-northd, ovn-controller, ovn-controller-vtep
- Utilities: ovn-nbctl, ovn-sbctl, vtep-ctl

ovs-sandbox OVN Tutorial Environments

- Environments with setup scripts, sample packets, and discussion
- Environments in the tutorial:
    1. Simple two-port setup
    2. Two logical switches, Four logical ports
    3. Two hypervisors
    4. Locally attached networks (Flat)
    5. Locally attached networks (VLAN)
    6. Stateful ACLs

OVN - Simple 2 Port Setup

```
$ make sandbox SANDBOXFLAGS=”--ovn”
$ ovn/env1/setup.sh
+ ovn-nbctl lswitch-add sw0
+ ovn-nbctl lport-add sw0 sw0-port1
+ ovn-nbctl lport-add sw0 sw0-port2
+ ovn-nbctl lport-set-addresses sw0-port1 00:00:00:00:00:01
+ ovn-nbctl lport-set-addresses sw0-port2 00:00:00:00:00:02
+ ovn-nbctl lport-set-port-security sw0-port1 00:00:00:00:00:01
+ ovn-nbctl lport-set-port-security sw0-port2 00:00:00:00:00:02
+ ovs-vsctl add-port br-int lport1 -- set Interface lport1
external_ids:iface-id=sw0-port1
+ ovs-vsctl add-port br-int lport2 -- set Interface lport2
external_ids:iface-id=sw0-port2
```

OVN - Simple 2 Port setup

```
$ ovn-nbctl show

    lswitch bf558515-cb97-4fd3-842a-d20a1b9a98e9 (sw0)

        lport sw0-port2

            addresses: 00:00:00:00:00:02

        lport sw0-port1

            addresses: 00:00:00:00:00:01
```

OVN - Simple 2 Port Setup

```
$ ovn-sbctl show
Chassis "56b18105-5706-46ef-80c4-ff20979ab068"
    Encap geneve
        ip: "127.0.0.1"
    Port_Binding "sw0-port1"
    Port_Binding "sw0-port2"
```

OVN - Simple 2 Port Setup

```
$ ovn-sbctl lflow-list
Datapath: 37914a82-30be-41d7-b8d4-f6cbda8478bd  Pipeline: ingress
  table=0(  ls_in_port_sec), priority=  100, match=(eth.src[40]), action=(drop;)
  table=0(  ls_in_port_sec), priority=  100, match=(vlan.present), action=(drop;)
  table=0(  ls_in_port_sec), priority=   50, match=(inport == "sw0-port1" && eth.src == {00:00:
00:00:00:01}), action=(next;)
  table=0(  ls_in_port_sec), priority=   50, match=(inport == "sw0-port2" && eth.src == {00:00:
00:00:00:02}), action=(next;)
  table=1(   ls_in_pre_acl), priority=    0, match=(1), action=(next;)
  table=2(       ls_in_acl), priority=    0, match=(1), action=(next;)
  table=3(   ls_in_l2_lkup), priority=  100, match=(eth.mcast), action=(outport = "_MC_flood";
output;)
  table=3(   ls_in_l2_lkup), priority=   50, match=(eth.dst == 00:00:00:00:00:01), action=
(outport = "sw0-port1"; output;)
  table=3(   ls_in_l2_lkup), priority=   50, match=(eth.dst == 00:00:00:00:00:02), action=
(outport = "sw0-port2"; output;)
Datapath: 37914a82-30be-41d7-b8d4-f6cbda8478bd  Pipeline: egress
  table=0(  ls_out_pre_acl), priority=    0, match=(1), action=(next;)
  table=1(      ls_out_acl), priority=    0, match=(1), action=(next;)
  table=2( ls_out_port_sec), priority=  100, match=(eth.mcast), action=(output;)
  table=2( ls_out_port_sec), priority=   50, match=(outport == "sw0-port1" && eth.dst == {00:00:
00:00:00:01}), action=(output;)
  table=2( ls_out_port_sec), priority=   50, match=(outport == "sw0-port2" && eth.dst == {00:00:
00:00:00:02}), action=(output;)
```

OVN - Simple 2 Port Setup

```
$ ovs-ofctl -O OpenFlow13 show br-int

…

$ ovs-ofctl -O OpenFlow13 dump-flows br-int

…
```

OVN - Simple 2 Port Setup

```
$ ovn/env1/packet1.sh
… unicast packet

$ ovn/env1/packet2.sh
… broadcast packet

$ ovn/env1/add-third-port.sh
… configure a 3rd port

$ ovn/env1/packet3.sh
… broadcast packet
```

Thank you

Russell Bryant (@russellbryant)