## NAME

ovs−testcontroller − simple OpenFlow controller for testing

## SYNOPSIS

**ovs−testcontroller** [*options*] *method* **[***method***]...**

## DESCRIPTION

**ovs−testcontroller** is a simple OpenFlow controller that manages any number of switches over the OpenFlow protocol, causing them to function as L2 MAC-learning switches or hubs. It is suitable for initial testing of OpenFlow networks. It is not a necessary or desirable part of a production OpenFlow deployment.

**ovs−testcontroller** controls one or more OpenFlow switches, specified as one or more of the following OpenFlow connection methods:

**pssl:**[*port*][**:***host*]
**ptcp:**[*port*][**:***host*]
Listens for OpenFlow connections on *port*. The default *port* is 6653. By default, connections are allowed from any IPv4 address. Specify *host* as an IPv4 address or a bracketed IPv6 address (e.g. **ptcp:6653:[::1]**). On Linux, use **%***device* to designate a scope for IPv6 link-level addresses, e.g. **ptcp:6653:[fe80::1234%eth0]**. DNS names can be used if built with unbound library. For **pssl**, the **−−private−key**,**−−certificate**, and **−−ca−cert** options are mandatory.

**punix:***file*
Listens for OpenFlow connections on the Unix domain server socket named *file*.

**ssl:***host*[**:***port*]
**tcp:***host*[**:***port*]
The specified *port* on the given *host*, which can be expressed either as a DNS name (if built with unbound library) or an IP address in IPv4 or IPv6 address format. Wrap IPv6 addresses in square brackets, e.g. **tcp:[::1]:6653**. On Linux, use **%***device* to designate a scope for IPv6 link-level addresses, e.g. **tcp:[fe80::1234%eth0]:6653**. For **ssl**, the **−−private−key**, **−−certificate**, and **−−ca−cert** options are mandatory.

If *port* is not specified, it defaults to 6653.

**unix:***file*
On POSIX, a Unix domain server socket named *file*.

On Windows, connect to a local named pipe that is represented by a file created in the path *file* to mimic the behavior of a Unix domain socket.

## OPTIONS

**−n**
**−−noflow**
By default, **ovs−testcontroller** sets up a flow in each OpenFlow switch whenever it receives a packet whose destination is known due through MAC learning. This option disables flow setup, so that every packet in the network passes through the controller.

This option is most useful for debugging. It reduces switching performance, so it should not be used in production.

**−−max−idle=***secs***|permanent**
Sets *secs* as the number of seconds that a flow set up by the controller will remain in the switch's flow table without any matching packets being seen. If **permanent** is specified, which is not recommended, flows will never expire. The default is 60 seconds.

This option has no effect when **−n** (or **−−noflow**) is in use (because the controller does not set up flows in that case).

**−H**
**−−hub** By default, the controller acts as an L2 MAC-learning switch. This option changes its behavior to that of a hub that floods packets on all but the incoming port.

If **−H** (or **−−hub**) and **−n** (or **−−noflow**) are used together, then the cumulative effect is that every packet passes through the controller and every packet is flooded.

This option is most useful for debugging. It reduces switching performance, so it should not be used in production.

**−w**[*wildcard_mask*]
**−−wildcards**[**=***wildcard_mask*]
By default, **ovs−testcontroller** sets up exact-match flows. This option allows it to set up wildcarded flows, which may reduce flow setup latency by causing less traffic to be sent up to the controller.

The optional *wildcard_mask* is an OpenFlow wildcard bitmask in hexadecimal that specifies the fields to wildcard. If no *wildcard_mask* is specified, the default value 0x2820F0 is used which specifies L2-only switching and wildcards L3 and L4 fields. Another interesting value is 0x2000EC, which specifies L3-only switching and wildcards L2 and L4 fields.

This option has no effect when **−n** (or **−−noflow**) is in use (because the controller does not set up flows in that case).

**−N**
**−−normal**
By default, **ovs−testcontroller** directs packets to a particular port or floods them. This option causes it to direct non-flooded packets to the OpenFlow **OFPP_NORMAL** port. This allows the switch itself to make decisions about packet destinations. Support for **OFPP_NORMAL** is optional in OpenFlow, so this option may not well with some non-Open vSwitch switches.

**−−mute**
Prevents ovs−testcontroller from replying to any OpenFlow messages sent to it by switches.

This option is only for debugging the Open vSwitch implementation of ‘‘fail open’’ mode. It must not be used in production.

**−q** *id*
**−−queue=***id*
By default, **ovs−testcontroller** uses the default OpenFlow queue for sending packets and setting up flows. Use one of these options, supplying *id* as an OpenFlow queue ID as a decimal number, to instead use that specific queue.

This option is incompatible with **−N** or **−−normal** and with **−H** or **−−hub**. If more than one is specified then this option takes precedence.

This option may be useful for testing or debugging quality of service setups.

**−Q** *port-name***:***queue-id*

**−−port−queue** *port-name***:***queue-id*
Configures packets received on the port named *port-name* (e.g. **eth0**) to be output on OpenFlow queue ID *queue-id* (specified as a decimal number). For the specified port, this option overrides the default specified on **−q** or **−−queue**.

This option may be specified any number of times with different *port-name* arguments.

This option is incompatible with **−N** or **−−normal** and with **−H** or **−−hub**. If more than one is specified then this option takes precedence.

This option may be useful for testing or debugging quality of service setups.

**−−with−flows** *file*
When a switch connects, push the flow entries as described in *file*. Each line in *file* is a flow entry in the format described for the **add−flows** command in the **Flow Syntax** section of the **ovs−ofctl**(8) man page.

Use this option more than once to add flows from multiple files.

### Public Key Infrastructure Options

**−p** *privkey.pem*
**−−private−key=***privkey.pem*
Specifies a PEM file containing the private key used as **ovs−testcontroller**'s identity for outgoing SSL connections.

**−c** *cert.pem*
**−−certificate=***cert.pem*
Specifies a PEM file containing a certificate that certifies the private key specified on **−p** or **−−private−key** to be trustworthy. The certificate must be signed by the certificate authority (CA) that the peer in SSL connections will use to verify it.

**−C** *cacert.pem*
**−−ca−cert=***cacert.pem*
Specifies a PEM file containing the CA certificate that **ovs−testcontroller** should use to verify certificates presented to it by SSL peers. (This may be the same certificate that SSL peers use to verify the certificate specified on **−c** or **−−certificate**, or it may be a different one, depending on the PKI design in use.)

**−C none**
**−−ca−cert=none**
Disables verification of certificates presented by SSL peers. This introduces a security risk, because it means that certificates cannot be verified to be those of known trusted hosts.

**−−peer−ca−cert=***peer-cacert.pem*
Specifies a PEM file that contains one or more additional certificates to send to SSL peers. *peer-cacert.pem* should be the CA certificate used to sign **ovs−testcontroller**'s own certificate, that is, the certificate specified on **−c** or **−−certificate**. If **ovs−testcontroller**'s certificate is self-signed, then **−−certificate** and **−−peer−ca−cert** should specify the same file.

This option is not useful in normal operation, because the SSL peer must already have the CA certificate for the peer to have any confidence in **ovs−testcontroller**'s identity. However, this offers a way for a new installation to bootstrap the CA certificate on its first SSL connection.

### Daemon Options

The following options are valid on POSIX based platforms.

**−−pidfile**[**=***pidfile*]
Causes a file (by default, **ovs−testcontroller.pid**) to be created indicating the PID of the running process. If the *pidfile* argument is not specified, or if it does not begin with **/**, then it is created in **/var/run/openvswitch**.

If **−−pidfile** is not specified, no pidfile is created.

**−−overwrite−pidfile**
By default, when **−−pidfile** is specified and the specified pidfile already exists and is locked by a running process, **ovs−testcontroller** refuses to start. Specify **−−overwrite−pidfile** to cause it to instead overwrite the pidfile.

When **−−pidfile** is not specified, this option has no effect.

**−−detach**
Runs **ovs−testcontroller** as a background process. The process forks, and in the child it starts a new session, closes the standard file descriptors (which has the side effect of disabling logging to the console), and changes its current directory to the root (unless **−−no−chdir** is specified). After the child completes its initialization, the parent exits.

**−−monitor**
Creates an additional process to monitor the **ovs−testcontroller** daemon. If the daemon dies due to a signal that indicates a programming error (**SIGABRT**, **SIGALRM**, **SIGBUS**, **SIGFPE**, **SIGILL**, **SIGPIPE**, **SIGSEGV**, **SIGXCPU**, or **SIGXFSZ**) then the monitor process starts a new copy of it. If the daemon dies or exits for another reason, the monitor process exits.

This option is normally used with **−−detach**, but it also functions without it.

**−−no−chdir**
By default, when **−−detach** is specified, **ovs−testcontroller** changes its current working directory to the root directory after it detaches. Otherwise, invoking **ovs−testcontroller** from a carelessly chosen directory would prevent the administrator from unmounting the file system that holds that directory.

Specifying **−−no−chdir** suppresses this behavior, preventing **ovs−testcontroller** from changing its current working directory. This may be useful for collecting core files, since it is common behavior to write core dumps into the current working directory and the root directory is not a good directory to use.

This option has no effect when **−−detach** is not specified.

**−−no−self−confinement**
By default daemon will try to self-confine itself to work with files under well-known directories determined during build. It is better to stick with this default behavior and not to use this flag unless some other Access Control is used to confine daemon. Note that in contrast to other access control implementations that are typically enforced from kernel-space (e.g. DAC or MAC), self-confinement is imposed from the user-space daemon itself and hence should not be considered as a full confinement strategy, but instead should be viewed as an additional layer of security.

**−−user** Causes **ovs−testcontroller** to run as a different user specified in "user:group", thus dropping most of the root privileges. Short forms "user" and ":group" are also allowed, with current user or group are assumed respectively. Only daemons started by the root user accepts this argument.

On Linux, daemons will be granted CAP_IPC_LOCK and CAP_NET_BIND_SERVICES before dropping root privileges. Daemons that interact with a datapath, such as **ovs−vswitchd**, will be granted three additional capabilities, namely CAP_NET_ADMIN, CAP_NET_BROADCAST and CAP_NET_RAW. The capability change will apply even if the new user is root.

On Windows, this option is not currently supported. For security reasons, specifying this option will cause the daemon process not to start.

**−v**[*spec*]
**−−verbose=**[*spec*]
Sets logging levels. Without any *spec*, sets the log level for every module and destination to **dbg**. Otherwise, *spec* is a list of words separated by spaces or commas or colons, up to one from each category below:

- A valid module name, as displayed by the **vlog/list** command on **ovs−appctl**(8), limits the log level change to the specified module.
- **syslog**, **console**, or **file**, to limit the log level change to only to the system log, to the console, or to a file, respectively. (If **−−detach** is specified, **ovs−testcontroller** closes its standard file descriptors, so logging to the console will have no effect.)

  On Windows platform, **syslog** is accepted as a word and is only useful along with the **−−syslog−target** option (the word has no effect otherwise).
- **off**, **emer**, **err**, **warn**, **info**, or **dbg**, to control the log level. Messages of the given severity or higher will be logged, and messages of lower severity will be filtered out. **off** filters out all messages. See **ovs−appctl**(8) for a definition of each log level.

Case is not significant within *spec*.

Regardless of the log levels set for **file**, logging to a file will not take place unless **−−log−file** is also specified (see below).

For compatibility with older versions of OVS, **any** is accepted as a word but has no effect.

**−v**

**−−verbose**
Sets the maximum logging verbosity level, equivalent to **−−verbose=dbg**.

**−vPATTERN:***destination***:***pattern*
**−−verbose=PATTERN:***destination***:***pattern*
Sets the log pattern for *destination* to *pattern*. Refer to **ovs−appctl**(8) for a description of the valid syntax for *pattern*.

**−vFACILITY:***facility*
**−−verbose=FACILITY:***facility*
Sets the RFC5424 facility of the log message. *facility* can be one of **kern**, **user**, **mail**, **daemon**, **auth**, **syslog**, **lpr**, **news**, **uucp**, **clock**, **ftp**, **ntp**, **audit**, **alert**, **clock2**, **local0**, **local1**, **local2**, **local3**, **local4**, **local5**, **local6** or **local7**. If this option is not specified, **daemon** is used as the default for the local system syslog and **local0** is used while sending a message to the target provided via the **−−syslog−target** option.

**−−log−file**[**=***file*]
Enables logging to a file. If *file* is specified, then it is used as the exact name for the log file. The default log file name used if *file* is omitted is **/var/log/openvswitch/ovs−testcontroller.log**.

**−−syslog−target=***host***:***port*
Send syslog messages to UDP *port* on *host*, in addition to the system syslog. The *host* must be a numerical IP address, not a hostname.

**−−syslog−method=***method*
Specify *method* how syslog messages should be sent to syslog daemon. Following forms are supported:

- **libc**, use libc **syslog()** function. Downside of using this options is that libc adds fixed prefix to every message before it is actually sent to the syslog daemon over **/dev/log** UNIX domain socket.
- **unix:***file*, use UNIX domain socket directly. It is possible to specify arbitrary message format with this option. However, **rsyslogd 8.9** and older versions use hard coded parser function anyway that limits UNIX domain socket use. If you want to use arbitrary message format with older **rsyslogd** versions, then use UDP socket to localhost IP address instead.
- **udp:***ip*:*port*, use UDP socket. With this method it is possible to use arbitrary message format also with older **rsyslogd**. When sending syslog messages over UDP socket extra precaution needs to be taken into account, for example, syslog daemon needs to be configured to listen on the specified UDP port, accidental iptables rules could be interfering with local syslog traffic and there are some security considerations that apply to UDP sockets, but do not apply to UNIX domain sockets.
- **null**, discards all messages logged to syslog.

The default is taken from the **OVS_SYSLOG_METHOD** environment variable; if it is unset, the default is **libc**.

**−−unixctl=***socket*
Sets the name of the control socket on which **ovs−testcontroller** listens for runtime management commands (see **RUNTIME MANAGEMENT COMMANDS**, below). If *socket* does not begin with **/**, it is interpreted as relative to **/var/run/openvswitch**. If **−−unixctl** is not used at all, the default socket is **/var/run/openvswitch/ovs−testcontroller.***pid***.ctl**, where *pid* is **ovs−testcontroller**'s process ID.

On Windows a local named pipe is used to listen for runtime management commands. A file is created in the absolute path as pointed by *socket* or if **−−unixctl** is not used at all, a file is created as **ovs−testcontroller.ctl** in the configured *OVS_RUNDIR* directory. The file exists just to mimic the behavior of a Unix domain socket.

Specifying **none** for *socket* disables the control socket feature.

**−h**
**−−help** Prints a brief help message to the console.

**−V**
**−−version**
Prints version information to the console.

**−O** [*version*[**,***version*]...]
**−−protocols=**[*version*[**,***version*]...]
Sets the OpenFlow protocol versions that are allowed when establishing an OpenFlow session.

These protocol versions are enabled by default:

- **OpenFlow10**, for OpenFlow 1.0.

The following protocol versions are generally supported, but for compatibility with older versions of Open vSwitch they are not enabled by default:

- **OpenFlow11**, for OpenFlow 1.1.
- **OpenFlow12**, for OpenFlow 1.2.
- **OpenFlow13**, for OpenFlow 1.3.
- **OpenFlow14**, for OpenFlow 1.4.
- **OpenFlow15**, for OpenFlow 1.5.

## EXAMPLES

To bind locally to port 6653 (the default) and wait for incoming connections from OpenFlow switches:

**% ovs−testcontroller ptcp:**

## BUGS

Configuring a Citrix XenServer to connect to a particular controller only points the remote OVSDB management connection to that controller. It does not also configure OpenFlow connections, because the manager is expected to do that over the management protocol. **ovs−testcontroller** is not an Open vSwitch manager and does not know how to do that.

As a stopgap workaround, **ovs−vsctl** can wait for an OVSDB connection and set the controller, e.g.:

**% ovs−vsctl −t0 −−db=pssl: −−certificate=cert.pem −−ca−cert=none −−private−key=privkey.pem −−peer−ca−cert=cacert.pem set−controller ssl:***ip*

## SEE ALSO

**ovs−appctl**(8), **ovs−ofctl**(8), **ovs−dpctl**(8)