

NAME

ovn-controller – Open Virtual Network local controller

SYNOPSIS

ovn-controller [*options*] [*ovs-database*]

DESCRIPTION

ovn-controller is the local controller daemon for OVN, the Open Virtual Network. It connects up to the OVN Southbound database (see **ovn-sb**(5)) over the OVSDB protocol, and down to the Open vSwitch database (see **ovs-vswitchd.conf.db**(5)) over the OVSDB protocol and to **ovs-vswitchd**(8) via OpenFlow. Each hypervisor and software gateway in an OVN deployment runs its own independent copy of **ovn-controller**; thus, **ovn-controller**'s downward connections are machine-local and do not run over a physical network.

ACL LOGGING

ACL log messages are logged through **ovn-controller**'s logging mechanism. ACL log entries have the module **acl_log** at log level **info**. Configuring logging is described below in the **Logging Options** section.

OPTIONS**Daemon Options**

--pidfile[=*pidfile*]

Causes a file (by default, *program.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 **/usr/local/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, the daemon 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 this program 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 this program. If it 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, the daemon changes its current working directory to the root directory after it detaches. Otherwise, invoking the daemon 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 the daemon 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 this daemon will try to self-confine itself to work with files under well-known directories whitelisted at build time. 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=*user:group*

Causes this program 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 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 two additional capabilities, namely **CAP_NET_ADMIN** 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.

Logging Options

-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, the daemon 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 `/usr/local/var/log/openvswitch/program.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* as how syslog messages should be sent to syslog daemon. The following forms are supported:

- **libc**, to use the libc `syslog()` function. This is the default behavior. 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**, to use a 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**, to use a 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.

PKI Options

PKI configuration is required in order to use SSL for the connections to the Northbound and Southbound databases.

-p *privkey.pem*

--private-key=*privkey.pem*

Specifies a PEM file containing the private key used as 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 for verifying certificates presented to this program 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.

Other Options

-h

--help Prints a brief help message to the console.

-V
--version

Prints version information to the console.

CONFIGURATION

ovn-controller retrieves most of its configuration information from the local Open vSwitch's `ovsdb-server` instance. The default location is **db.sock** in the local Open vSwitch's "run" directory. It may be overridden by specifying the *ovs-database* argument in one of the following forms:

ssl:ip:port

The specified SSL *port* on the host at the given *ip*, which must be expressed as an IP address (not a DNS name) in IPv4 or IPv6 address format. If *ip* is an IPv6 address, then wrap *ip* with square brackets, e.g.: **ssl:[::1]:6640**. The **--private-key**, **--certificate**, and **--ca-cert** options are mandatory when this form is used.

tcp:ip:port

Connect to the given TCP *port* on *ip*, where *ip* can be an IPv4 or IPv6 address. If *ip* is an IPv6 address, then wrap *ip* with square brackets, e.g.: **tcp:[::1]:6640**.

unix:file

On POSIX, connect to the 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.

pssl:port:ip

Listen on the given SSL *port* for a connection. By default, connections are not bound to a particular local IP address and it listens only on IPv4 (but not IPv6) addresses, but specifying *ip* limits connections to those from the given *ip*, either IPv4 or IPv6 address. If *ip* is an IPv6 address, then wrap *ip* with square brackets, e.g.: **pssl:6640:[::1]**. The **--private-key**, **--certificate**, and **--ca-cert** options are mandatory when this form is used.

ptcp:port:ip

Listen on the given TCP *port* for a connection. By default, connections are not bound to a particular local IP address and it listens only on IPv4 (but not IPv6) addresses, but *ip* may be specified to listen only for connections to the given *ip*, either IPv4 or IPv6 address. If *ip* is an IPv6 address, then wrap *ip* with square brackets, e.g.: **ptcp:6640:[::1]**.

punix:file

On POSIX, listen on the Unix domain server socket named *file* for a connection.

On Windows, listen on a local named pipe. A file is created in the path *file* to mimic the behavior of a Unix domain socket.

ovn-controller assumes it gets configuration information from the following keys in the **Open_vSwitch** table of the local OVS instance:

external_ids:system-id

The chassis name to use in the Chassis table.

external_ids:hostname

The hostname to use in the Chassis table.

external_ids:ovn-bridge

The integration bridge to which logical ports are attached. The default is **br-int**. If this bridge does not exist when `ovn-controller` starts, it will be created automatically with the default configuration suggested in **ovn-architecture(7)**.

external_ids:ovn-remote

The OVN database that this system should connect to for its configuration, in one of the same forms documented above for the *ovs-database*.

external_ids:ovn-remote-probe-interval

The inactivity probe interval of the connection to the OVN database, in milliseconds. If the value is zero, it disables the connection keepalive feature.

If the value is nonzero, then it will be forced to a value of at least 1000 ms.

external_ids:ovn-encap-type

The encapsulation type that a chassis should use to connect to this node. Multiple encapsulation types may be specified with a comma-separated list. Each listed encapsulation type will be paired with **ovn-encap-ip**.

Supported tunnel types for connecting hypervisors are **geneve** and **stt**. Gateways may use **geneve**, **vxlan**, or **stt**.

Due to the limited amount of metadata in **vxlan**, the capabilities and performance of connected gateways will be reduced versus other tunnel formats.

external_ids:ovn-encap-ip

The IP address that a chassis should use to connect to this node using encapsulation types specified by **external_ids:ovn-encap-type**.

external_ids:ovn-bridge-mappings

A list of key-value pairs that map a physical network name to a local ovs bridge that provides connectivity to that network. An example value mapping two physical network names to two ovs bridges would be: **physnet1:br-eth0,physnet2:br-eth1**.

external_ids:ovn-encap-csum

ovn-encap-csum indicates that encapsulation checksums can be transmitted and received with reasonable performance. It is a hint to senders transmitting data to this chassis that they should use checksums to protect OVN metadata. Set to **true** to enable or **false** to disable. Depending on the capabilities of the network interface card, enabling encapsulation checksum may incur performance loss. In such cases, encapsulation checksums can be disabled.

ovn-controller reads the following values from the **Open_vSwitch** database of the local OVS instance:

datapath-type from **Bridge** table

This value is read from local OVS integration bridge row of **Bridge** table and populated in **external_ids:datapath-type** of the **Chassis** table in the OVN_Southbound database.

iface-types from **Open_vSwitch** table

This value is populated in **external_ids:iface-types** of the **Chassis** table in the OVN_Southbound database.

private_key, **certificate**, **ca_cert**, and **bootstrap_ca_cert** from **SSL** table

These values provide the SSL configuration used for connecting to the OVN southbound database server when an SSL connection type is configured via **external_ids:ovn-remote**. Note that this SSL configuration can also be provided via command-line options, the configuration in the database takes precedence if both are present.

OPEN VSWITCH DATABASE USAGE

ovn-controller uses a number of **external_ids** keys in the Open vSwitch database to keep track of ports and interfaces. For proper operation, users should not change or clear these keys:

external_ids:ovn-chassis-id in the **Port** table

The presence of this key identifies a tunnel port within the integration bridge as one created by **ovn-controller** to reach a remote chassis. Its value is the chassis ID of the remote chassis.

external_ids:ct-zone-* in the **Bridge** table

Logical ports and gateway routers are assigned a connection tracking zone by **ovn-controller** for stateful services. To keep state across restarts of **ovn-controller**, these keys are stored in the integration bridge's Bridge table. The name contains a prefix of

ct-zone- followed by the name of the logical port or gateway router's zone key. The value for this key identifies the zone used for this port.

external_ids:ovn-localnet-port in the **Port** table

The presence of this key identifies a patch port as one created by **ovn-controller** to connect the integration bridge and another bridge to implement a **localnet** logical port. Its value is the name of the logical port with **type** set to **localnet** that the port implements. See **external_ids:ovn-bridge-mappings**, above, for more information.

Each **localnet** logical port is implemented as a pair of patch ports, one in the integration bridge, one in a different bridge, with the same **external_ids:ovn-localnet-port** value.

external_ids:ovn-l2gateway-port in the **Port** table

The presence of this key identifies a patch port as one created by **ovn-controller** to connect the integration bridge and another bridge to implement a **l2gateway** logical port. Its value is the name of the logical port with **type** set to **l2gateway** that the port implements. See **external_ids:ovn-bridge-mappings**, above, for more information.

Each **l2gateway** logical port is implemented as a pair of patch ports, one in the integration bridge, one in a different bridge, with the same **external_ids:ovn-l2gateway-port** value.

external_ids:ovn-l3gateway-port in the **Port** table

This key identifies a patch port as one created by **ovn-controller** to implement a **l3gateway** logical port. Its value is the name of the logical port with type set to **l3gateway**. This patch port is similar to the OVN logical patch port, except that **l3gateway** port can only be bound to a particular chassis.

external_ids:ovn-logical-patch-port in the **Port** table

This key identifies a patch port as one created by **ovn-controller** to implement an OVN logical patch port within the integration bridge. Its value is the name of the OVN logical patch port that it implements.

RUNTIME MANAGEMENT COMMANDS

ovs-appctl can send commands to a running **ovn-controller** process. The currently supported commands are described below.

exit Causes **ovn-controller** to gracefully terminate.

ct-zone-list

Lists each local logical port and its connection tracking zone.

inject-pkt *microflow*

Injects *microflow* into the connected Open vSwitch instance. *microflow* must contain an ingress logical port (**inport** argument) that is present on the Open vSwitch instance.

The *microflow* argument describes the packet whose forwarding is to be simulated, in the syntax of an OVN logical expression, as described in **ovn-sb(5)**, to express constraints. The parser understands prerequisites; for example, if the expression refers to **ip4.src**, there is no need to explicitly state **ip4** or **eth.type == 0x800**.