

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.

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 either of **--ca-cert** or **--bootstrap-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 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 localhost TCP port whose value is written in *file*.

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.

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**.