NAME

ovn-bridge-controller − OVN_Bridge_Controller database schema

This database is the interface between OVN Bridge Controller and the cloud management system (CMS) to program and control the OVS bridges using OVN logical flows. The CMS produces almost all of the contents of the database. The **ovn−bridge−controller** program monitors the database contents and programs the OVS bridges with the OpenFlow rules.

External IDs

Each of the tables in this database contains a special column, named **external_ids**. This column has the same form and purpose each place it appears.

external_ids: map of string-string pairs

Key-value pairs for use by the CMS. The CMS might use certain pairs, for example, to identify entities in its own configuration that correspond to those in this database.

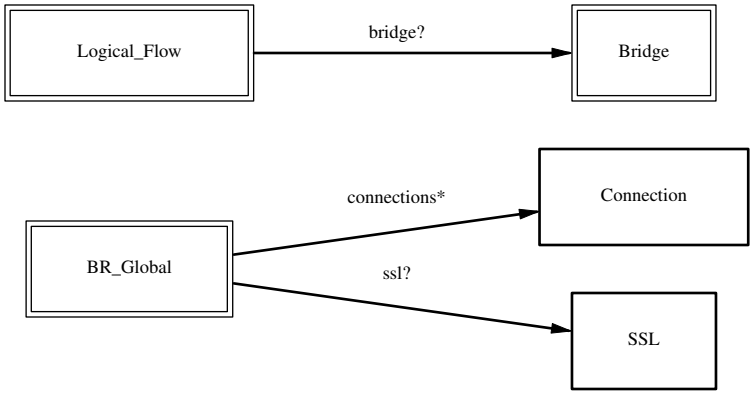
TABLE SUMMARY

The following list summarizes the purpose of each of the tables in the **OVN_Bridge_Controller** database. Each table is described in more detail on a later page.

Table	Purpose
BR_Global	Bridge Controller configuration
Connection	OVSDB client connections.
SSL	SSL configuration.
Bridge	OVS Bridge to control
Logical_Flow	Logical flow

TABLE RELATIONSHIPS

The following diagram shows the relationship among tables in the database. Each node represents a table. Tables that are part of the ‘‘root set’’ are shown with double borders. Each edge leads from the table that contains it and points to the table that its value represents. Edges are labeled with their column names, followed by a constraint on the number of allowed values: **?** for zero or one, ***** for zero or more, **+** for one or more. Thick lines represent strong references; thin lines represent weak references.



BR_Global TABLE

Summary:

br_cfg	integer
Common Columns:	
external_ids	map of string-string pairs
Common options:	
options	map of string-string pairs
Connection Options:	
connections	set of **Connection**s
ssl	optional **SSL**

Details:

br_cfg: integer

Sequence number for client to increment. When a client modifies any part of the bridge Controller database configuration and wishes to wait for **ovn−br−controller** to finish applying the changes, it may increment this sequence number.

Common Columns:

external_ids: map of string-string pairs

See **External IDs** at the beginning of this document.

Common options:

options: map of string-string pairs

This column provides general key/value settings. The supported options are described individually below.

Connection Options:

connections: set of **Connection**s

Database clients to which the Open vSwitch database server should connect or on which it should listen, along with options for how these connections should be configured. See the **Connection** table for more information.

ssl: optional **SSL**

Global SSL/TLS configuration.

Connection TABLE

Configuration for a database connection to an Open vSwitch database (OVSDB) client.

This table primarily configures the Open vSwitch database server (**ovsdb−server**).

The Open vSwitch database server can initiate and maintain active connections to remote clients. It can also listen for database connections.

Summary:

Core Features:	
target	string (must be unique within table)
Client Failure Detection and Handling:	
max_backoff	optional integer, at least 1,000
inactivity_probe	optional integer
Status:	
is_connected	boolean
status : last_error	optional string
status : state	optional string, one of **ACTIVE**, **BACKOFF**, **CONNECTING**, **IDLE**, or **VOID**
status : sec_since_connect	optional string, containing an integer, at least 0
status : sec_since_disconnect	optional string, containing an integer, at least 0
status : locks_held	optional string
status : locks_waiting	optional string
status : locks_lost	optional string
status : n_connections	optional string, containing an integer, at least 2
status : bound_port	optional string, containing an integer
Common Columns:	
external_ids	map of string-string pairs
other_config	map of string-string pairs

Details:

Core Features:

target: string (must be unique within table)

Connection methods for clients.

The following connection methods are currently supported:

ssl:*host*[**:***port*]

The specified SSL/TLS *port* on the host at the given *host*, which can either be a DNS name (if built with unbound library) or an IP address. A valid SSL/TLS configuration must be provided when this form is used, this configuration can be specified via command-line options or the **SSL** table.

If *port* is not specified, it defaults to 6640.

SSL/TLS support is an optional feature that is not always built as part of OVN or Open vSwitch.

tcp:*host*[**:***port*]

The specified TCP *port* on the host at the given *host*, which can either be a DNS name (if built with unbound library) or an IP address. If *host* is an IPv6 address, wrap it in square brackets, e.g. **tcp:[::1]:6640**.

If *port* is not specified, it defaults to 6640.

pssl:[*port*][**:***host*]

Listens for SSL/TLS connections on the specified TCP *port*. Specify 0 for *port* to have the kernel automatically choose an available port. If *host*, which can either be a DNS name (if built with unbound library) or an IP address, is specified, then connections are restricted to the resolved or specified local IPaddress (either IPv4 or IPv6 address). If *host* is an IPv6 address, wrap in square brackets, e.g. **pssl:6640:[::1]**. If *host* is not specified

then it listens only on IPv4 (but not IPv6) addresses. A valid SSL/TLS configuration must be provided when this form is used, this can be specified either via command-line options or the **SSL** table.

If *port* is not specified, it defaults to 6640.

SSL/TLS support is an optional feature that is not always built as part of OVN or Open vSwitch.

ptcp:[*port*][**:***host*]

Listens for connections on the specified TCP *port*. Specify 0 for *port* to have the kernel automatically choose an available port. If *host*, which can either be a DNS name (if built with unbound library) or an IP address, is specified, then connections are restricted to the resolved or specified local IP address (either IPv4 or IPv6 address). If *host* is an IPv6 address, wrap it in square brackets, e.g. **ptcp:6640:[::1]**. If *host* is not specified then it listens only on IPv4 addresses.

If *port* is not specified, it defaults to 6640.

When multiple clients are configured, the **target** values must be unique. Duplicate **target** values yield unspecified results.

Client Failure Detection and Handling:

max_backoff: optional integer, at least 1,000

Maximum number of milliseconds to wait between connection attempts. Default is implementation-specific.

inactivity_probe: optional integer

Maximum number of milliseconds of idle time on connection to the client before sending an inactivity probe message. If Open vSwitch does not communicate with the client for the specified number of seconds, it will send a probe. If a response is not received for the same additional amount of time, Open vSwitch assumes the connection has been broken and attempts to reconnect. Default is implementation-specific. A value of 0 disables inactivity probes.

Status:

Key-value pair of **is_connected** is always updated. Other key-value pairs in the status columns may be updated depends on the **target** type.

When **target** specifies a connection method that listens for inbound connections (e.g. **ptcp:** or **punix:**), both **n_connections** and **is_connected** may also be updated while the remaining key-value pairs are omitted.

On the other hand, when **target** specifies an outbound connection, all key-value pairs may be updated, except the above-mentioned two key-value pairs associated with inbound connection targets. They are omitted.

is_connected: boolean

true if currently connected to this client, **false** otherwise.

status : last_error: optional string

A human-readable description of the last error on the connection to the manager; i.e. **strerror(errno)**. This key will exist only if an error has occurred.

status : state: optional string, one of **ACTIVE**, **BACKOFF**, **CONNECTING**, **IDLE**, or **VOID**

The state of the connection to the manager:

VOID Connection is disabled.

BACKOFF

Attempting to reconnect at an increasing period.

CONNECTING
Attempting to connect.

ACTIVE
Connected, remote host responsive.

IDLE Connection is idle. Waiting for response to keep-alive.

These values may change in the future. They are provided only for human consumption.

status : sec_since_connect: optional string, containing an integer, at least 0
The amount of time since this client last successfully connected to the database (in seconds). Value is empty if client has never successfully been connected.

status : sec_since_disconnect: optional string, containing an integer, at least 0
The amount of time since this client last disconnected from the database (in seconds). Value is empty if client has never disconnected.

status : locks_held: optional string
Space-separated list of the names of OVSDB locks that the connection holds. Omitted if the connection does not hold any locks.

status : locks_waiting: optional string
Space-separated list of the names of OVSDB locks that the connection is currently waiting to acquire. Omitted if the connection is not waiting for any locks.

status : locks_lost: optional string
Space-separated list of the names of OVSDB locks that the connection has had stolen by another OVSDB client. Omitted if no locks have been stolen from this connection.

status : n_connections: optional string, containing an integer, at least 2
When **target** specifies a connection method that listens for inbound connections (e.g. **ptcp:** or **pssl:**) and more than one connection is actually active, the value is the number of active connections. Otherwise, this key-value pair is omitted.

status : bound_port: optional string, containing an integer
When **target** is **ptcp:** or **pssl:**, this is the TCP port on which the OVSDB server is listening. (This is particularly useful when **target** specifies a port of 0, allowing the kernel to choose any available port.)

Common Columns:

The overall purpose of these columns is described under **Common Columns** at the beginning of this document.

external_ids: map of string-string pairs

other_config: map of string-string pairs

SSL TABLE

SSL/TLS configuration for ovn-nb database access.

Summary:

private_key	string
certificate	string
ca_cert	string
bootstrap_ca_cert	boolean
ssl_protocols	string
ssl_ciphers	string
ssl_ciphersuites	string
Common Columns:	
external_ids	map of string-string pairs

Details:

private_key: string

Name of a PEM file containing the private key used as the switch's identity for SSL/TLS connections to the controller.

certificate: string

Name of a PEM file containing a certificate, signed by the certificate authority (CA) used by the controller and manager, that certifies the switch's private key, identifying a trustworthy switch.

ca_cert: string

Name of a PEM file containing the CA certificate used to verify that the switch is connected to a trustworthy controller.

bootstrap_ca_cert: boolean

If set to **true**, then Open vSwitch will attempt to obtain the CA certificate from the controller on its first SSL/TLS connection and save it to the named PEM file. If it is successful, it will immediately drop the connection and reconnect, and from then on all SSL/TLS connections must be authenticated by a certificate signed by the CA certificate thus obtained. **This option exposes the SSL/TLS connection to a man−in−the−middle attack obtaining the initial CA certificate.** It may still be useful for bootstrapping.

ssl_protocols: string

Range or a comma- or space-delimited list of the SSL/TLS protocols to enable for SSL/TLS connections.

Supported protocols include **TLSv1.2** and **TLSv1.3**. Ranges can be provided in a form of two protocol names separated with a dash (**TLSv1.2−TLSv1.3**), or as a single protocol name with a plus sign (**TLSv1.2+**). The value can be a list of protocols or exactly one range. The range is a preferred way of specifying protocols and the configuration always behaves as if the range between the minimum and the maximum specified version is provided, i.e., if the value is set to **TLSv1.X,TLSv1.(X+2)**, the **TLSv1.(X+1)** will also be enabled as if it was a range. Regardless of order, the highest protocol supported by both sides will be chosen when making the connection.

The default when this option is omitted is **TLSv1.2+**.

ssl_ciphers: string

List of ciphers (in OpenSSL cipher string format) to be supported for SSL/TLS connections with TLSv1.2. The default when this option is omitted is **DEFAULT:@SECLEVEL=2**.

ssl_ciphersuites: string

List of ciphersuites (in OpenSSL ciphersuites string format) to be supported for SSL/TLS connections with TLSv1.3 and later. Default value from OpenSSL will be used when this option is omitted.

Common Columns:

The overall purpose of these columns is described under **Common Columns** at the beginning of this document.

external_ids: map of string-string pairs

Bridge TABLE

Summary:

name	string (must be unique within table)
options	map of string-string pairs
external_ids	map of string-string pairs

Details:

name: string (must be unique within table)

Name of the OVS bridge. This bridge should exist in the local OVS database.

options: map of string-string pairs

Reserved for future use.

external_ids: map of string-string pairs

See **External IDs** at the beginning of this document.

Logical_Flow TABLE

Summary:

bridge	optional **Bridge**
table_id	integer, in range 0 to 100
priority	integer, in range 0 to 65,535
match	string
actions	string
external_ids	map of string-string pairs

Details:

bridge: optional **Bridge**

The bridge to which the logical flow belongs to.

table_id: integer, in range 0 to 100

The stage in the logical pipeline, analogous to an OpenFlow table number.

priority: integer, in range 0 to 65,535

The flow's priority. Flows with numerically higher priority take precedence over those with lower. If two logical flows with the same priority both match, then the one actually applied to the packet is undefined.

match: string

A matching expression. OVN provides a superset of OpenFlow matching capabilities, using a syntax similar to Boolean expressions in a programming language.

Please see the documentation of the **match** column of the **Logical_Flow** table in the **OVN_Southbound** database.

actions: string

Logical datapath actions, to be executed when the logical flow represented by this row is the highest-priority match.

Actions share lexical syntax with the **match** column. An empty set of actions (or one that contains just white space or comments), or a set of actions that consists of just **drop;**, causes the matched packets to be dropped. Otherwise, the column should contain a sequence of actions, each terminated by a semicolon.

Please see the documentation of the **actions** column of the **Logical_Flow** table in the **OVN_Southbound** database.

While **ovn−br−controller** technically supports all OVN logical flow actions defined in the **actions** column of the **Logical_Flow** table in the **OVN_Southbound**, some actions are not intended for use by the CMS. These include controller OpenFlow actions (e.g., put_dhcp_opts, dns_lookup) and OVN-specific actions (e.g., look_arp, look_fdb).

It is the CMS's responsibility to avoid using such actions. When the CMS defines the packet pipeline in the OVS bridge and adds logical flows using **ovn−br−controller**, it is expected to be mindful of using only relevant OVN actions.

external_ids: map of string-string pairs

See **External IDs** at the beginning of this document.