

NAME

ovn-ic-nb – OVN_IC_Northbound database schema

This database is the interface for cloud management system (CMS), such as OpenStack, to configure OVN interconnection settings. The CMS produces almost all of the contents of the database. The **ovn-ic** program monitors the database contents, transforms it, and stores it into the **OVN_IC_Southbound** database.

We generally speak of “the” CMS, but one can imagine scenarios in which multiple CMSes manage different parts of OVN interconnection.

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.

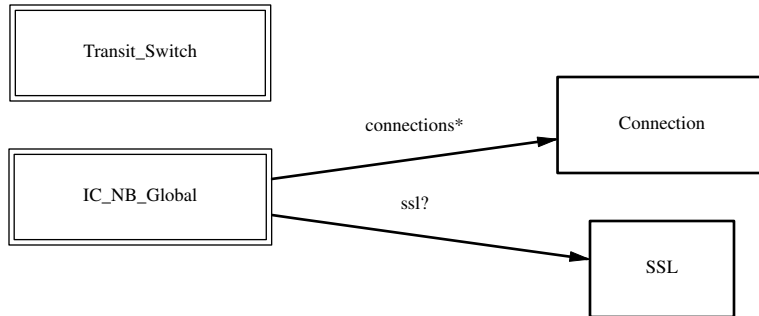
TABLE SUMMARY

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

Table	Purpose
IC_NB_Global	IC Northbound configuration
Transit_Switch	Transit logical switch
SSL	SSL configuration.
Connection	OVSDDB client connections.

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.



IC_NB_Global TABLE

Northbound configuration for OVN interconnection. This table must have exactly one row.

Summary:

Common Columns:

external_ids map of string-string pairs

Common options:

options map of string-string pairs

Connection Options:

connections set of **Connections**

ssl optional **SSL**

Details:

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 **Connections**

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 configuration.

Transit_Switch TABLE

Each row represents one transit logical switch for interconnection between different OVN deployments (availability zones).

Summary:

Naming:

name string (must be unique within table)

Common Columns:

other_config map of string-string pairs

external_ids map of string-string pairs

Details:

Naming:

name: string (must be unique within table)

A name that uniquely identifies the transit logical switch.

Common Columns:

other_config: map of string-string pairs

external_ids: map of string-string pairs

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

SSL TABLE

SSL 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
<i>Common Columns:</i>	
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 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 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 connections must be authenticated by a certificate signed by the CA certificate thus obtained. **This option exposes the SSL connection to a man-in-the-middle attack obtaining the initial CA certificate.** It may still be useful for bootstrapping.
- ssl_protocols:** string
List of SSL protocols to be enabled for SSL connections. The default when this option is omitted is **TLSv1,TLSv1.1,TLSv1.2**.
- ssl_ciphers:** string
List of ciphers (in OpenSSL cipher string format) to be supported for SSL connections. The default when this option is omitted is **HIGH:!aNULL:!MD5**.

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

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 *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 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 support is an optional feature that is not always built as part of 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 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 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 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 support is an optional feature that is not always built as part of 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 OVSDDB 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 OVSDDB 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 OVSDDB locks that the connection has had stolen by another OVSDDB 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 OVSDDB 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