NAME

ovn-nb − OVN_Northbound database schema

This database is the interface between OVN and the cloud management system (CMS), such as OpenStack, running above it. The CMS produces almost all of the contents of the database. The **ovn−northd** program monitors the database contents, transforms it, and stores it into the **OVN_Southbound** database.

We generally speak of ‘‘the’’ CMS, but one can imagine scenarios in which multiple CMSes manage different parts of an OVN deployment.

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.

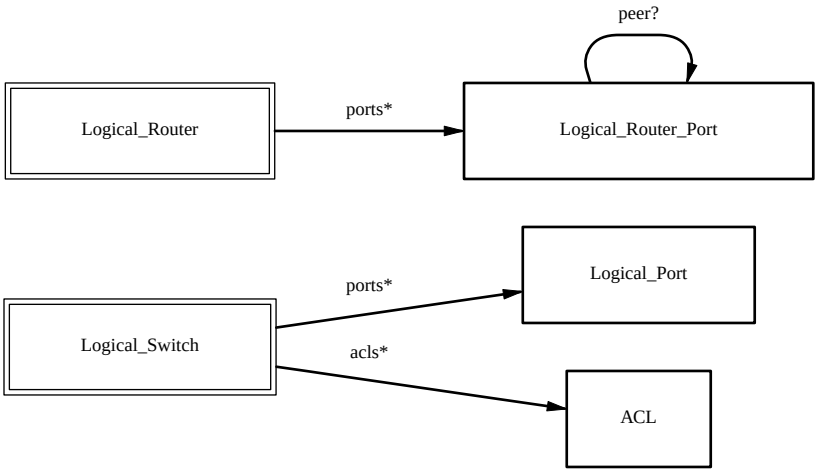
TABLE SUMMARY

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

Table	Purpose
Logical_Switch	L2 logical switch
Logical_Port	L2 logical switch port
ACL	Access Control List (ACL) rule
Logical_Router	L3 logical router
Logical_Router_Port	L3 logical router port

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.



Logical_Switch TABLE

Each row represents one L2 logical switch.

Summary:

name	string
ports	set of **Logical_Port**s
acls	set of **ACL**s
Common Columns:	
external_ids	map of string-string pairs

Details:

name: string

A name for the logical switch. This name has no special meaning or purpose other than to provide convenience for human interaction with the ovn-nb database. There is no requirement for the name to be unique. The logical switch's UUID should be used as the unique identifier.

ports: set of **Logical_Port**s

The logical ports connected to the logical switch.

It is an error for multiple logical switches to include the same logical port.

acls: set of **ACL**s

Access control rules that apply to packets within the logical switch.

Common Columns:

external_ids: map of string-string pairs

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

Logical_Port TABLE

A port within an L2 logical switch.

Summary:

Core Features:	
name	string (must be unique within table)
type	string
Options:	
options	map of string-string pairs
Options for router ports:	
options : router-port	optional string
Options for localnet ports:	
options : network_name	optional string
Options for vtep ports:	
options : vtep-physical-switch	optional string
options : vtep-logical-switch	optional string
Containers:	
parent_name	optional string
tag	optional integer, in range 1 to 4,095
Port State:	
up	optional boolean
enabled	optional boolean
Addressing:	
addresses	set of strings
port_security	set of strings
Common Columns:	
external_ids	map of string-string pairs

Details:

Core Features:

name: string (must be unique within table)

The logical port name.

For entities (VMs or containers) that are spawned in the hypervisor, the name used here must match those used in the **external_ids:iface-id** in the **Open_vSwitch** database's **Interface** table, because hypervisors use **external_ids:iface-id** as a lookup key to identify the network interface of that entity.

For containers that share a VIF within a VM, the name can be any unique identifier. See **Containers**, below, for more information.

type: string

Specify a type for this logical port. Logical ports can be used to model other types of connectivity into an OVN logical switch. The following types are defined:

(empty string)
: A VM (or VIF) interface.

router A connection to a logical router.

localnet
: A connection to a locally accessible network from each **ovn−controller** instance. A logical switch can only have a single **localnet** port attached and at most one regular logical port. This is used to model direct connectivity to an existing network.

vtep A port to a logical switch on a VTEP gateway.

Options:

options: map of string-string pairs
This column provides key/value settings specific to the logical port **type**. The type-specific options are described individually below.

Options for router ports:
These options apply when **type** is **router**.

If a given logical switch has multiple **router** ports, the **Logical_Router_Port** rows that they reference must be all on the same **Logical_Router** (for different subnets).

options : router-port: optional string
Required. The **name** of the **Logical_Router_Port** to which this logical switch port is connected.

Options for localnet ports:
These options apply when **type** is **localnet**.

options : network_name: optional string
Required. The name of the network to which the **localnet** port is connected. Each hypervisor, via **ovn−controller**, uses its local configuration to determine exactly how to connect to this locally accessible network.

Options for vtep ports:
These options apply when **type** is **vtep**.

options : vtep-physical-switch: optional string
Required. The name of the VTEP gateway.

options : vtep-logical-switch: optional string
Required. A logical switch name connected by the VTEP gateway.

Containers:
When a large number of containers are nested within a VM, it may be too expensive to dedicate a VIF to each container. OVN can use VLAN tags to support such cases. Each container is assigned a VLAN ID and each packet that passes between the hypervisor and the VM is tagged with the appropriate ID for the container. Such VLAN IDs never appear on a physical wire, even inside a tunnel, so they need not be unique except relative to a single VM on a hypervisor.

These columns are used for VIFs that represent nested containers using shared VIFs. For VMs and for containers that have dedicated VIFs, they are empty.

parent_name: optional string
The VM interface through which the nested container sends its network traffic. This must match the **name** column for some other **Logical_Port**.

tag: optional integer, in range 1 to 4,095
The VLAN tag in the network traffic associated with a container's network interface.

When **type** is set to **localnet**, this can be set to indicate that the port represents a connection to a specific VLAN on a locally accessible network. The VLAN ID is used to match incoming traffic and is also added to outgoing traffic.

Port State:

up: optional boolean
This column is populated by **ovn−northd**, rather than by the CMS plugin as is most of this database. When a logical port is bound to a physical location in the OVN Southbound database **Binding** table, **ovn−northd** sets this column to **true**; otherwise, or if the port becomes unbound later, it sets it to **false**. This allows the CMS to wait for a VM's (or container's) networking to become active before it allows the VM (or container) to start.

enabled: optional boolean
This column is used to administratively set port state. If this column is empty or is set to **true**, the port is enabled. If this column is set to **false**, the port is disabled. A disabled port has all ingress and egress traffic dropped.

Addressing:

addresses: set of strings

Addresses owned by the logical port.

Each element in the set must take one of the following forms:

xx:*xx*:*xx*:*xx*:*xx*:*xx*

An Ethernet address owned by the logical port. Like a physical Ethernet NIC, a logical port ordinarily has a single fixed Ethernet address.

When a OVN logical switch processes a unicast Ethernet frame whose destination MAC address is in a logical port's **addresses** column, it delivers it only to that port, as if a MAC learning process had learned that MAC address on the port.

xx:*xx*:*xx*:*xx*:*xx*:*xx a*.*b*.*c*.*d*

This form has all the effects of the previous form. It also indicates that the logical port owns the given IPv4 address.

The OVN logical switch uses this information to synthesize responses to ARP requests without traversing the physical network. The OVN logical router connected to the logical switch, if any, uses this information to avoid issuing ARP requests for logical switch ports.

Note that the order here is important. The Ethernet address must be listed before the IP address.

unknown

This indicates that the logical port has an unknown set of Ethernet addresses. When an OVN logical switch processes a unicast Ethernet frame whose destination MAC address is not in any logical port's **addresses** column, it delivers it to the port (or ports) whose **addresses** columns include **unknown**.

port_security: set of strings

A set of L2 (Ethernet) addresses from which the logical port is allowed to send packets and to which it is allowed to receive packets. If this column is empty, all addresses are permitted. Logical ports are always allowed to receive packets addressed to multicast and broadcast addresses.

Each member of the set is an Ethernet address in the form *xx*:*xx*:*xx*:*xx*:*xx*:*xx*.

This specification will be extended to support L3 port security.

Common Columns:

external_ids: map of string-string pairs

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

ACL TABLE

Each row in this table represents one ACL rule for a logical switch that points to it through its **acls** column. The **action** column for the highest-**priority** matching row in this table determines a packet's treatment. If no row matches, packets are allowed by default. (Default-deny treatment is possible: add a rule with **priority** 0, **0** as **match**, and **deny** as **action**.)

Summary:

priority	integer, in range 0 to 32,767
direction	string, either **to−lport** or **from−lport**
match	string
action	string, one of **allow−related**, **drop**, **allow**, or **reject**
log	boolean
Common Columns:	
external_ids	map of string-string pairs

Details:

priority: integer, in range 0 to 32,767

The ACL rule's priority. Rules with numerically higher priority take precedence over those with lower. If two ACL rules with the same priority both match, then the one actually applied to a packet is undefined.

Return traffic from an **allow−related** flow is always allowed and cannot be changed through an ACL.

direction: string, either **to−lport** or **from−lport**

Direction of the traffic to which this rule should apply:

- **from−lport**: Used to implement filters on traffic arriving from a logical port. These rules are applied to the logical switch's ingress pipeline.
- **to−lport**: Used to implement filters on traffic forwarded to a logical port. These rules are applied to the logical switch's egress pipeline.

match: string

The packets that the ACL should match, in the same expression language used for the **match** column in the OVN Southbound database's **Logical_Flow** table. The **outport** logical port is only available in the **to−lport** direction (the **inport** is available in both directions).

By default all traffic is allowed. When writing a more restrictive policy, it is important to remember to allow flows such as ARP and IPv6 neighbor discovery packets.

Note that you can not create an ACL matching on a port with type=router.

action: string, one of **allow−related**, **drop**, **allow**, or **reject**

The action to take when the ACL rule matches:

- **allow**: Forward the packet.
- **allow−related**: Forward the packet and related traffic (e.g. inbound replies to an outbound connection).
- **drop**: Silently drop the packet.
- **reject**: Drop the packet, replying with a RST for TCP or ICMP unreachable message for other IP-based protocols. **Not implemented−−currently treated as drop**

log: boolean

If set to **true**, packets that match the ACL will trigger a log message on the transport node or nodes that perform ACL processing. Logging may be combined with any **action**.

Logging is not yet implemented.

Common Columns:

external_ids: map of string-string pairs
See **External IDs** at the beginning of this document.

Logical_Router TABLE

Each row represents one L3 logical router.

Summary:

name	string
ports	set of **Logical_Router_Port**s
default_gw	optional string
Common Columns:	
external_ids	map of string-string pairs

Details:

name: string

A name for the logical router. This name has no special meaning or purpose other than to provide convenience for human interaction with the ovn-nb database. There is no requirement for the name to be unique. The logical router's UUID should be used as the unique identifier.

ports: set of **Logical_Router_Port**s

The router's ports.

default_gw: optional string

IP address to use as default gateway, if any.

Common Columns:

external_ids: map of string-string pairs

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

Logical_Router_Port TABLE

A port within an L3 logical router.

Exactly one **Logical_Router** row must reference a given logical router port.

Summary:

name	string (must be unique within table)
network	string
mac	string
enabled	optional boolean
Attachment:	
peer	optional **Logical_Router_Port**
Common Columns:	
external_ids	map of string-string pairs

Details:

name: string (must be unique within table)

A name for the logical router port.

In addition to provide convenience for human interaction with the ovn-nb database, this column is used as reference by its patch port in **Logical_Port** or another logical router port in **Logical_Router_Port**.

network: string

The IP address of the router and the netmask. For example, **192.168.0.1/24** indicates that the router's IP address is 192.168.0.1 and that packets destined to 192.168.0.*x* should be routed to this port.

mac: string

The Ethernet address that belongs to this router port.

enabled: optional boolean

This column is used to administratively set port state. If this column is empty or is set to **true**, the port is enabled. If this column is set to **false**, the port is disabled. A disabled port has all ingress and egress traffic dropped.

Attachment:

A given router port serves one of two purposes:

- To attach a logical switch to a logical router. A logical router port of this type is referenced by exactly one **Logical_Port** of type **router**. The value of **name** is set as **router−port** in column **options** of **Logical_Port**. In this case **peer** column is empty.
- To connect one logical router to another. This requires a pair of logical router ports, each connected to a different router. Each router port in the pair specifies the other in its **peer** column. No **Logical_Switch** refers to the router port.

peer: optional **Logical_Router_Port**

For a router port used to connect two logical routers, this identifies the other router port in the pair by **name**.

For a router port attached to a logical switch, this column is empty.

Common Columns:

external_ids: map of string-string pairs

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