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.

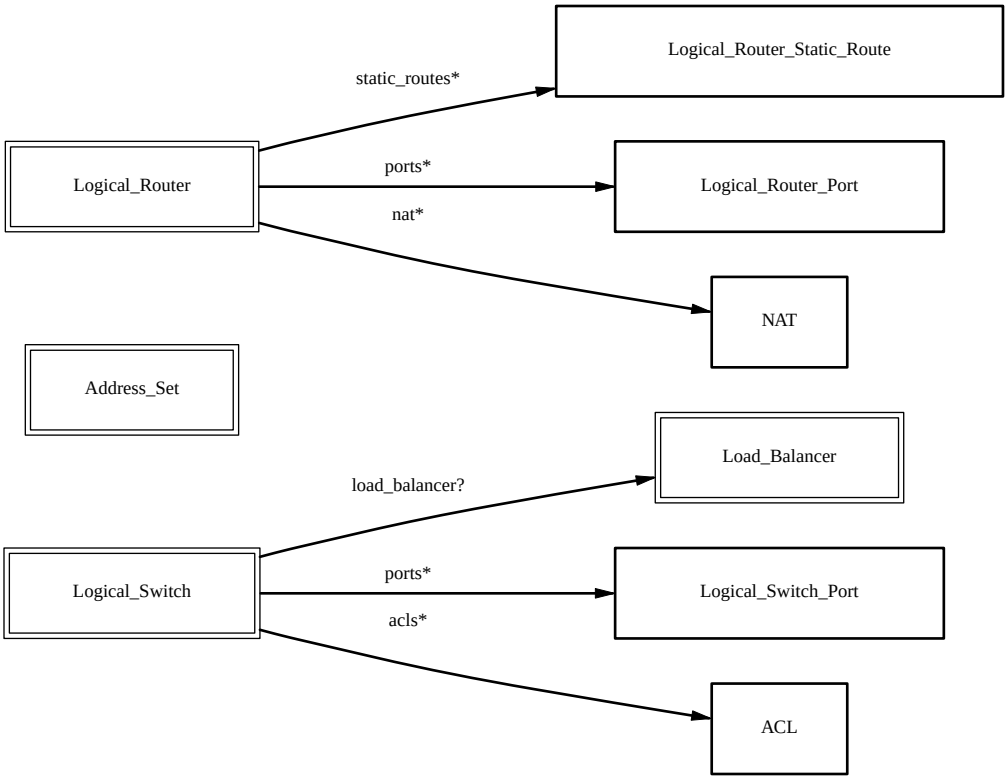
TABLE SUMMARY

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_Switch_Port	L2 logical switch port
Address_Set	Address Sets
Load_Balancer	load balancer
ACL	Access Control List (ACL) rule
Logical_Router	L3 logical router
Logical_Router_Port	L3 logical router port
Logical_Router_Static_Route	Logical router static routes
NAT	NAT rules for a Gateway router.

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.

There are two kinds of logical switches, that is, ones that fully virtualize the network (overlay logical switches) and ones that provide simple connectivity to a physical network (bridged logical switches). They work in the same way when providing connectivity between logical ports on same chasis, but differently when connecting remote logical ports. Overlay logical switches connect remote logical ports by tunnels, while bridged logical switches provide connectivity to remote ports by bridging the packets to directly connected physical L2 segment with the help of **localnet** ports. Each bridged logical switch has one and only one **localnet** port, which has only one special address **unknown**.

Summary:

name	string
ports	set of **Logical_Switch_Port**s
load_balancer	optional **Load_Balancer**
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_Switch_Port**s

The logical ports connected to the logical switch.

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

load_balancer: optional **Load_Balancer**

Load balance a virtual ipv4 address to a set of logical port endpoint ipv4 addresses.

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_Switch_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 l2gateway ports:	
options : network_name	optional string
Options for vtep ports:	
options : vtep-physical-switch	optional string
options : vtep-logical-switch	optional string
VMI (or VIF) Options:	
options : policing_rate	optional string
options : policing_burst	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. This is used to model direct connectivity to an existing network.

l2gateway
A connection to a physical 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**.

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 l2gateway ports:
These options apply when **type** is **l2gateway**.

options : network_name: optional string
Required. The name of the network to which the **l2gateway** port is connected. The L2 gateway, via **ovn−controller**, uses its local configuration to determine exactly how to connect to this 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.

VMI (or VIF) Options:
These options apply to logical ports with **type** having (empty string)

options : policing_rate: optional string
If set, indicates the maximum rate for data sent from this interface, in kbps. Data exceeding this rate is dropped.

options : policing_burst: optional string
If set, indicates the maximum burst size for data sent from this interface, in kb.

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_Switch_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:

Ethernet address followed by zero or more IPv4 or IPv6 addresses (or both)
An Ethernet address defined is 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.

If IPv4 or IPv6 address(es) (or both) are defined, it indicates that the logical port owns the given IP addresses.

If IPv4 address(es) are defined, 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(es) if defined.

Examples:

80:fa:5b:06:72:b7
This indicates that the logical port owns the above mac address.

80:fa:5b:06:72:b7 10.0.0.4 20.0.0.4
This indicates that the logical port owns the mac address and two IPv4 addresses.

80:fa:5b:06:72:b7 fdaa:15f2:72cf:0:f816:3eff:fe20:3f41
This indicates that the logical port owns the mac address and 1 IPv6 address.

80:fa:5b:06:72:b7 10.0.0.4 fdaa:15f2:72cf:0:f816:3eff:fe20:3f41
This indicates that the logical port owns the mac address and 1 IPv4 address and 1 IPv6 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

This column controls the addresses from which the host attached to the logical port (''the host'') is allowed to send packets and to which it is allowed to receive packets. If this column is empty, all addresses are permitted.

Each element in the set must begin with one Ethernet address. This would restrict the host to sending packets from and receiving packets to the ethernet addresses defined in the logical port's **port_security** column. It also restricts the inner source MAC addresses that the host may send in ARP and IPv6 Neighbor Discovery packets. The host is always allowed to receive packets to multicast and broadcast Ethernet addresses.

Each element in the set may additionally contain one or more IPv4 or IPv6 addresses (or both), with optional masks. If a mask is given, it must be a CIDR mask. In addition to the restrictions described for Ethernet addresses above, such an element restricts the IPv4 or IPv6 addresses from which the host may send and to which it may receive packets to the specified addresses. A masked address, if the host part is zero, indicates that the host is allowed to use any address in the subnet; if the host part is nonzero, the mask simply indicates the size of the subnet. In addition:

- If any IPv4 address is given, the host is also allowed to receive packets to the IPv4 local broadcast address 255.255.255.255 and to IPv4 multicast addresses (224.0.0.0/4). If an IPv4 address with a mask is given, the host is also allowed to receive packets to the broadcast address in that specified subnet.

  If any IPv4 address is given, the host is additionally restricted to sending ARP packets with the specified source IPv4 address. (RARP is not restricted.)

- If any IPv6 address is given, the host is also allowed to receive packets to IPv6 multicast addresses (ff00::/8).

  If any IPv6 address is given, the host is additionally restricted to sending IPv6 Neighbor Discovery Solicitation or Advertisement packets with the specified source address or, for solicitations, the unspecified address.

If an element includes an IPv4 address, but no IPv6 addresses, then IPv6 traffic is not allowed. If an element includes an IPv6 address, but no IPv4 address, then IPv4 and ARP traffic is not allowed.

This column uses the same lexical syntax as the **match** column in the OVN Southbound database's **Pipeline** table. Multiple addresses within an element may be space or comma separated.

This column is provided as a convenience to cloud management systems, but all of the features that it implements can be implemented as ACLs using the **ACL** table.

Examples:

80:fa:5b:06:72:b7

The host may send traffic from and receive traffic to the specified MAC address, and to receive traffic to Ethernet multicast and broadcast addresses, but not otherwise. The host may not send ARP or IPv6 Neighbor Discovery packets with inner source Ethernet addresses other than the one specified.

80:fa:5b:06:72:b7 192.168.1.10/24

This adds further restrictions to the first example. The host may send IPv4 packets from or receive IPv4 packets to only 192.168.1.10, except that it may also receive IPv4 packets to 192.168.1.255 (based on the subnet mask), 255.255.255.255, and any address in 224.0.0.0/4. The host may not send ARPs with a source Ethernet address other than

80:fa:5b:06:72:b7 or source IPv4 address other than 192.168.1.10. The host may not send or receive any IPv6 (including IPv6 Neighbor Discovery) traffic.

"80:fa:5b:12:42:ba", "80:fa:5b:06:72:b7 192.168.1.10/24"
The host may send traffic from and receive traffic to the specified MAC addresses, and to receive traffic to Ethernet multicast and broadcast addresses, but not otherwise. With MAC 80:fa:5b:12:42:ba, the host may send traffic from and receive traffic to any L3 address. With MAC 80:fa:5b:06:72:b7, the host may send IPv4 packets from or receive IPv4 packets to only 192.168.1.10, except that it may also receive IPv4 packets to 192.168.1.255 (based on the subnet mask), 255.255.255.255, and any address in 224.0.0.0/4. The host may not send or receive any IPv6 (including IPv6 Neighbor Discovery) traffic.

Common Columns:

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

Address_Set TABLE

Each row in this table represents a named set of addresses. An address set may contain Ethernet, IPv4, or IPv6 addresses with optional bitwise or CIDR masks. Address set may ultimately be used in ACLs to compare against fields such as **ip4.src** or **ip6.src**. A single address set must contain addresses of the same type. As an example, the following would create an address set with three IP addresses:

ovn−nbctl create Address_Set name=set1 addresses=’10.0.0.1 10.0.0.2 10.0.0.3’

Address sets may be used in the **match** column of the **ACL** table. For syntax information, see the details of the expression language used for the **match** column in the **Logical_Flow** table of the **OVN_Southbound** database.

Summary:

name	string (must be unique within table)
addresses	set of strings
Common Columns:	
external_ids	map of string-string pairs

Details:

name: string (must be unique within table)

A name for the address set. This must be unique among all address sets.

addresses: set of strings

The set of addresses in string form.

Common Columns:

external_ids: map of string-string pairs

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

Load_Balancer TABLE

Each row represents one load balancer.

Summary:

vips	map of string-string pairs
protocol	optional string, either **udp** or **tcp**
Common Columns:	
external_ids	map of string-string pairs

Details:

vips: map of string-string pairs

A map of virtual IPv4 addresses (and an optional port number with **:** as a separator) associated with this load balancer and their corresponding endpoint IPv4 addresses (and optional port numbers with **:** as separators) separated by commas. If the destination IP address (and port number) of a packet leaving a container or a VM matches the virtual IPv4 address (and port number) provided here as a key, then OVN will statefully replace the destination IP address by one of the provided IPv4 address (and port number) in this map as a value. Examples for keys are "192.168.1.4" and "172.16.1.8:80". Examples for value are "10.0.0.1, 10.0.0.2" and "20.0.0.10:8800, 20.0.0.11:8800".

protocol: optional string, either **udp** or **tcp**

Valid protocols are **tcp** or **udp**. This column is useful when a port number is provided as part of the **vips** column. If this column is empty and a port number is provided as part of **vips** column, OVN assumes the protocol to be **tcp**.

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.

Note that when **localnet** port exists in a lswitch, for **to−lport** direction, the **inport** works only if the **to−lport** is located on the same chassis as the **inport**.

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
static_routes	set of **Logical_Router_Static_Route**s
default_gw	optional string
enabled	optional boolean
nat	set of **NAT**s
Options:	
options : chassis	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.

static_routes: set of **Logical_Router_Static_Route**s

One or more static routes for the router.

default_gw: optional string

IP address to use as default gateway, if any.

enabled: optional boolean

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

nat: set of **NAT**s

One or more NAT rules for the router. NAT rules only work on the Gateway routers.

Options:

Additional options for the logical router.

options : chassis: optional string

If set, indicates that the logical router in question is a Gateway router (which is centralized) and resides in the set chassis. The same value is also used by **ovn−controller** to uniquely identify the chassis in the OVN deployment and comes from **external_ids:system−id** in the **Open_vSwitch** table of Open_vSwitch database.

The Gateway router can only be connected to a distributed router via a switch if SNAT and DNAT are to be configured in the Gateway router.

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 string
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_Switch_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_Switch_Port** of type **router**. The value of **name** is set as **router−port** in column **options** of **Logical_Switch_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 string

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.

Logical_Router_Static_Route TABLE

Each record represents a static route.

Summary:

ip_prefix	string
nexthop	string
output_port	optional string

Details:

ip_prefix: string

IP prefix of this route (e.g. 192.168.100.0/24).

nexthop: string

Nexthop IP address for this route. Nexthop IP address should be the IP address of a connected router port or the IP address of a logical port.

output_port: optional string

The name of the **Logical_Router_Port** via which the packet needs to be sent out. This is optional and when not specified, OVN will automatically figure this out based on the **nexthop**.

NAT TABLE

Each record represents a NAT rule in a Gateway router.

Summary:

type	string, one of **snat**, **dnat**, or **dnat_and_snat**
external_ip	string
logical_ip	string

Details:

type: string, one of **snat**, **dnat**, or **dnat_and_snat**

Type of the NAT rule.

- When **type** is **dnat**, the externally visible IP address **external_ip** is DNATted to the IP address **logical_ip** in the logical space.
- When **type** is **snat**, IP packets with their source IP address that either matches the IP address in **logical_ip** or is in the network provided by **logical_ip** is SNATed into the IP address in **external_ip**.
- When **type** is **dnat_and_snat**, the externally visible IP address **external_ip** is DNATted to the IP address **logical_ip** in the logical space. In addition, IP packets with the source IP address that matches **logical_ip** is SNATed into the IP address in **external_ip**.

external_ip: string

An IPv4 address.

logical_ip: string

An IPv4 network (e.g 192.168.1.0/24) or an IPv4 address.