

# Connecting to a Controller

Use the OVSDB protocol to connect to a controller. The `ovs-vsctl` command requires an IP address and a port number on the OVS database server. In the example below, the switch connects to an OF controller with an IP address of 10.10.53.50, and port number of 6633.

```
root@PicOS-OVS# ovs-vsctl set-controller br0 tcp:10.10.53.50:6633
root@PicOS-OVS#
```

## Verify connectivity with the controller:

```
admin@PicOS-OVS$ ovs-vsctl show

101c4a95-2973-4aeb-9c0a-a04380950b4d
  Bridge "br0"
    Controller "tcp:10.10.53.50:6633"
      is_connected: true
    Port "te-1/1/50"
      tag: 1
      Interface "te-1/1/50"
        type: "pica8"
    Port "ge-1/1/48"
      tag: 1
      Interface "ge-1/1/48"
        type: "pica8"
    Port "br0"
      Interface "br0"
        type: internal
```

## Check detailed controller status. A properly working controller configuration would appear as:

```
admin@PicOS-OVS$ ovs-vsctl list controller

_uuid                : f35735f3-1d62-45ba-967e-59e324d1e150
auxiliary             : {}
connection_mode      : []
controller_burst_limit: []
controller_rate_limit: []
enable_async_messages: []
external_ids         : {}
inactivity_probe     : []
is_connected         : true
local_gateway        : []
local_ip             : []
local_netmask        : []
max_backoff          : []
other_config         : {}
role                 : other
status               : {current_version="OpenFlow13", sec_since_connect="20", state=ACTIVE}
target               : "tcp:10.10.53.50:6633"
```

In the event of an error in the bridge configuration, such as a mismatch in the open flow version, the output would appear as:

```

admin@PicOS-OVS$ovs-vsctl list controller

 _uuid          : f35735f3-1d62-45ba-967e-59e324d1e150
 auxiliary      : {}
 connection_mode : []
 controller_burst_limit: []
 controller_rate_limit: []
 enable_async_messages: []
 external_ids   : {}
 inactivity_probe : []
 is_connected   : true
 local_gateway  : []
 local_ip       : []
 local_netmask  : []
 max_backoff    : []
 other_config   : {}
 role           : other
 status         : {last_error="Connection refused", sec_since_connect="7713",sec_since_disconnect="7721",
 state=ACTIVE}
 target         : "tcp:10.10.53.50:6633"

```

## Modify Openflow Protocol

PicOS supports openflow1.0, openflow1.2, openflow1.3, openflow1.4 default, if user wants special configuration openflow protocol number, the command is below.

**ovs-vsctl set bridge br0 protocol=<OpenFlow protocol number>**

```

admin@PicOS-OVS$ovs-vsctl set bridge br0 protocol=OpenFlow13
admin@PicOS-OVS$
admin@PicOS-OVS$ovs-vsctl set bridge br0 protocol=OpenFlow10,OpenFlow13

```

## Connection Mode between Bridge and Controller

From PicOS 2.4, the default connection between one bridge and all the controllers is in-band mode. User can configure to disable it, using the command as shown below.

**ovs-vsctl set bridge br0 other\_config=disable-in-band=<true|false>**

Disable in-band mode. After disable in-band is done, all the controllers must use out-of-band mode connection to bridge, which means only the management port can be used to connect.

```

ovs-vsctl set bridge br0 other_config=disable-in-band=true

```

Enable in-band mode, all the controllers can use in-band connect to bridge.

```

ovs-vsctl set bridge br0 other_config=disable-in-band=false

```

If enabling in-band in bridge, user also can configure in-band or out-of-band in one single controller. That mean that this controller only uses out-of-band, Others still use in-band connect to bridge.

**ovs-vsctl set controller [\_uuid] connection\_mode=out-of-band**

```

admin@PicOS-OVS$ovs-vsctl list controller
_uuid          : 7f651b3b-4b0d-4d9b-b6e5-fe67499be1c4
auxiliary      : {}
connection_mode : in-band
controller_burst_limit: []
controller_rate_limit: []
enable_async_messages: []
external_ids   : {}
inactivity_probe : []
is_connected   : false
local_gateway  : []
local_ip       : []
local_netmask  : []
max_backoff    : []
other_config   : {}
role           : other
status         : {last_error="Network is unreachable", state=BACKOFF}
target         : "tcp:10.10.50.42:6653"
admin@PicOS-OVS$
admin@PicOS-OVS$ovs-vsctl set controller 7f651b3b-4b0d-4d9b-b6e5-fe67499be1c4 connection_mode=out-of-band
admin@PicOS-OVS$
admin@PicOS-OVS$ovs-vsctl list controller
_uuid          : 7f651b3b-4b0d-4d9b-b6e5-fe67499be1c4
auxiliary      : {}
connection_mode : out-of-band
controller_burst_limit: []
controller_rate_limit: []
enable_async_messages: []
external_ids   : {}
inactivity_probe : []
is_connected   : false
local_gateway  : []
local_ip       : []
local_netmask  : []
max_backoff    : []
other_config   : {}
role           : other
status         : {last_error="Network is unreachable", state=BACKOFF}
target         : "tcp:10.10.50.42:6653"

```

## Auxiliary Connections

PicOS OVS supports Auxiliary connections to the controller. Auxiliary connections configuration is based on the OVS controller. When the user wants to use it, they must first configure auxiliary in the controller list.

```
ovs-vsctl set controller {uuid} auxiliary:{id}={udp | tcp}
```

Auxiliary configuration has two parameters:

- The first is Auxiliary ID, an integer that identifies auxiliary connections.

And the range is [1, 255].

- The second is a string that specifies the transport of auxiliary connection.

And now we only support “udp” and “tcp”.

## Example:

**Step 1: Assume there is a controller “tcp:10.10.51.16:6633” on Bridge br0 using this:**

Firstly, get uuid of the controller using this:

```
ovs-vsctl set-controller br0 tcp:10.10.51.50:6633
```

**Step 2: Get controller uuid:**

```
ovs-vsctl list controller
admin@PicOS-OVS$ovs-vsctl list controller
_uuid           : 6eb5d036-87af-44ca-aa58-2b916ae126f4
auxiliary       : {}
connection_mode : []
controller_burst_limit: []
controller_rate_limit: []
enable_async_messages: []
external_ids    : {}
inactivity_probe : []
is_connected    : false
local_gateway   : []
local_ip        : []
local_netmask   : []
max_backoff     : []
other_config    : {}
role            : other
status          : {last_error="Connection refused", sec_since_disconnect="1", state=BACKOFF}
target          : "tcp:10.10.51.50:6633"
```

### Step 3:

```
admin@PicOS-OVS$ovs-vsctl set controller 6eb5d036-87af-44ca-aa58-2b916ae126f4 auxiliary:1=udp
```

## Configure Flow Table Flush once set or delete controller

**ovs-vsctl set bridge br0 other\_config:enable-flush=<true|false>**

In PicOS OVS switch, the flow table will be cleared when user uses the **set-controller** command for the bridge. Starting with PicOS 2.6, user can define if the flow table is flushed or not by the **set-controller** and **del-controller** commands.

The flow table is flushed when **enable-flush=true**. The flow table is not flushed when **enable-flush=false**. The default value is **enable-flush=true**.

```
ovs-vsctl set bridge br0 other_config:enable-flush=true
ovs-vsctl set bridge br0 other_config:enable-flush=false
```

Display the configuration.

```
ovs-vsctl list Bridge
ovs-vsctl list bridge br0
```