OvS

Open vSwitch

November 8-10, 2022

OVSDB: A database to configure your database?

Ilya Maximets, Red Hat

Ways to configure ovsdb-server process

- Command line arguments:
  - `--remote=ptcp:6641 --unixctl=db.ctl`

- `UnixCtl commands:`
  - `ovs-appctl -t ovsdb-server ovsdb-server/add-remote ptcp:6641`
  - `ovs-appctl -t ovsdb-server ovsdb-server/add-db my_conf.db`

- Via rows in a served database:
  - `--remote=db:Open_vSwitch,Open_vSwitch,manager_options`

What needs to be configurable

- Database file path.

What needs to be configurable

- Database file path.

- Remote address to listen on:
  - Globally for all servers in a cluster.
  - Individually for each server in a cluster.

What needs to be configurable

- Database file path.

- Remote address to listen on:
  - Globally for all servers in a cluster.
  - Individually for each server in a cluster.

- SSL configuration.

What needs to be configurable

- Database file path.

- Remote address to listen on:
  - Globally for all servers in a cluster.
  - Individually for each server in a cluster.

- SSL configuration.

- Connection parameters:
  - Inactivity probe interval.
  - ‘Backoff’ settings.
  - Role for the access control

What needs to be configurable

- Database file path.
- Remote address to listen on:
    - Globally for all servers in a cluster.
    - Individually for each server in a cluster.
- SSL configuration.

- Connection parameters:
    - Inactivity probe interval.
    - ‘Backoff’ settings.
    - Role for the access control
- Relay server parameters:
    - Target ovsdb-server
    - Schema name

What needs to be configurable

- Database file path.
- Remote address to listen on:
  - Globally for all servers in a cluster.
  - Individually for each server in a cluster.
- SSL configuration.
- Replication:
  - Target ovsdb-server
  - List of excluded tables

- Connection parameters:
  - Inactivity probe interval.
  - ‘Backoff’ settings.
  - Role for the access control
- Relay server parameters:
  - Target ovsdb-server
  - Schema name

What is actually configurable

		Command Line	UnixCtl	Served database
Database file path		✅	✅	❌
Remote address to listen on	For all servers in a cluster	❌	❌	✅
	Individually for each server	✅	✅	❌
SSL configuration		✅	❌	✅
Connection parameters	Inactivity probe interval	❌ (✅ for replication)	❌	✅
	Max. backoff	❌	❌	✅
	Role (RBAC)	❌	❌	✅
Relay server parameters	Target ovsdb-server	✅	❌	❌
	Schema name	✅	❌	❌
Replication	Target ovsdb-server	✅	✅	❌
	Tables to exclude	✅	✅	❌

What is actually configurable

- **There is no method to cover all cases!**

Various proposals

- Extend command line arguments since the only missing bits are connection method parameters.
  - Proposal [Terry Wilson]:
    - `--remote="pssl:6640;inactivity_probe=10000,max_backoff=4000"`

      https://patchwork.ozlabs.org/project/openvswitch/patch/20220722191126.2112869-1-twilson@redhat.com/
  - Limited to command line specification of ‘remote’.
  - Doesn’t cover relay case, which already has a bit overloaded configuration:
    - `relay:OVN_Southbound:ssl:127.0.0.1:6642`

Various proposals

- Extend command line arguments since the only missing bits are connection method parameters.
    - Maybe extend the generic ‘connection method’ to include all possible options? E.g.:
        - `--remote=`**`[inactivity_probe:10000,max_backoff:4000]`**`pssl:6640`
    - Relay still looks overloaded:
        - `relay:OVN_Southbound:`**`[inactivity_probe:10000]`**`ssl:127.0.0.1:6642`
    - Possibility to configure the same thing in 2 different (conflicting) ways via database.

Various proposals

- Add a local database to store the configuration of the database server.
  - Can cover all the cases.
    - Except for a remote address for all servers at once.
  - No need for overly complicated argument lists.

Little known fact:

- There is already a small database that is created by the monitor thread and holds parts of the current database configuration, so it can be restored in case of a crash. This concept can be expanded to be an actual database.

What is already done

- There is a Local_Config database schema in the OVS repository:
  - `ovsdb/local-config.ovsschema`

- It can be used today by manually crafting transactions.

- Contains some minimal configuration for connection methods.

Next steps?

- User-friendly tools to work with Local_Config database both online and offline
  - Some work is started here:

    `“Add ovs-confctl utility for Local_Config db ”`

    `https://patchwork.ozlabs.org/project/openvswitch/patch/20220713030250.2634491-1-twilson@redhat.com/`

- Extension of a schema to allow more things to be configured.

- Potential conversion to another internal database alongside with the _Server database, or potential merge of these databases:
  - Main difference is that _Server is read-only.
  - May need protection from the remote access to the configuration database?

Open vSwitch

Thanks!

Email: i.maximets@ovn.org