MuL SDN Controller - HOWTO

Document Revision History

Rev. No.	Date	Revised By	Comments
1.0	30.10.2013	MUL development team	Doc to describe how-to use mul controller

Contents

1 Installing MuL controller

1.1 Getting MuL Code

The latest MuL code can be downloaded using git as follows:

```
$ git clone git://git.code.sf.net/p/mul/code mul-sf-code
```

1.2 Building MuL Code from source

1. Get necessary packages

   a) For Ubuntu 12.04.3 LTS, the following packages should be installed:

```
$ sudo apt-get install flex bison libwxgtk2.6-dev build-essential expect g++-multilib
tofrodos zlib1g-dev gawk libffi-dev
```

   Not verified with recent ubuntu versions but should work fine.

   b) Install core packages glib-2.0 (>= 2.32) and libevent (>= 2.0.18):

   i) There is a utility script which should configure and build these packages(but will not install it to the system), just, do enough for mul compilation. (This is our preferred way)

```
$ cd mul-sf-code
$ cd SCRIPTS
$ ./configure_ext_libs
```

   OR,

   ii) One can also download and build/install them separately or by apt-get.

2. Configure and make MuL

```
$  cd ../
$ ./autogen.sh
$  ./configure  --with-glib=`pwd`/common-libs/3rd-party/glib-2.32.0/ --with-
libevent=`pwd`/common-libs/3rd-party/libevent-2.0.21-stable/
$ make
```

Please note that if you build and install glib and lib-event separately (not using the script), you can simply use the following as the penultimate step above:

```
$ ./configure
```

You might need to pass LDFLAGS and CFLAGS to “configure” if these are installed in non-standard directories.

1.3 Running Mul

Before running Mul, one to understand that MuL has many components, each running as a separate application. MuL is basically a suite of co-operative applications comprising of:

1) Mul director/core
    a. Major component of mul
    b. Handles all low level connections and does openflow processing.
    c. Location : $(mul-code)/mul

2) Mul services
    a. These provide basic infra services built on top of mul director
    b. Currently available :
        i. Topology discovery service
        ii. Path finding service
    c. Location : $(mul-code)/services/loadable/topo_routing/

3) MuL system apps
    a. System apps are built using a common api provided by mul-director and mul-services
    b. These are hardly aware of Openflow and hence designed to work across different openflow versions provided switches support common requirement of these apps
    c. Currently available :
        i. L2 learning app
            - A simple learning application
            - Location : $(mul-code/application/l2switch

        ii. Fabric app
            - Enables multi-tenant aware end to end P2P connections between end-hosts across a mesh of Openflow switches
            - Dynamically recalculates paths on various network events
            - Location : $(mul-code/application/fabric

        iii. CLI app

- This provides a common cli based provisioning tool for all MuL components.
- Location : $(mul-code/application/cli

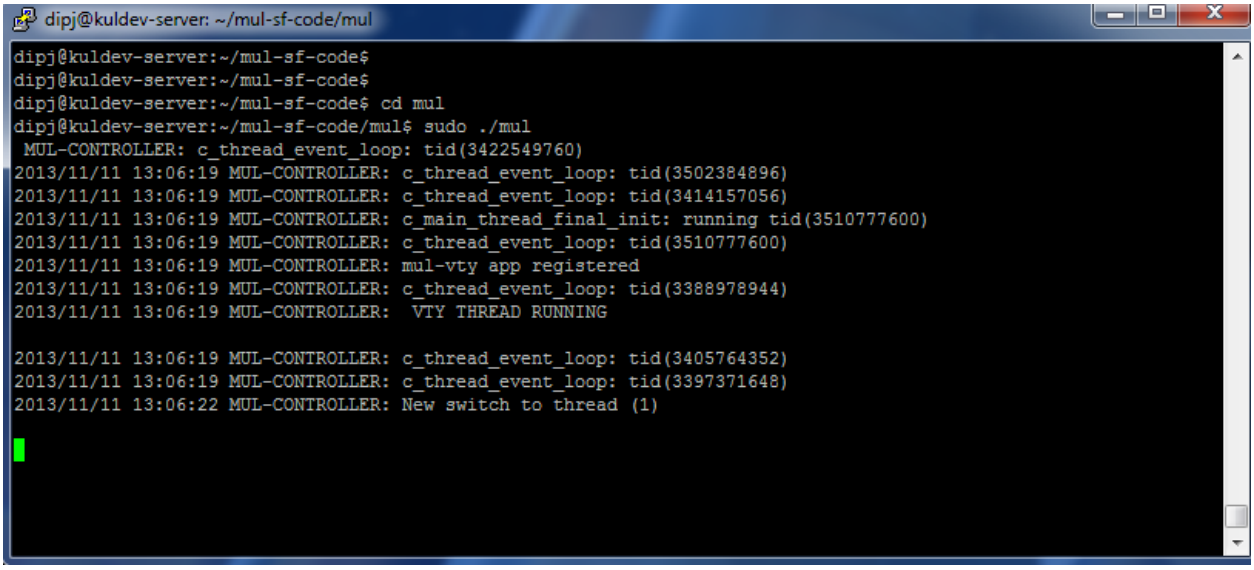
1.3.1 Running Mul director/core

```
$ cd mul-sf-code/mul
$ sudo ./mul
```

Possible options to use -

```
-d        : Daemon mode
-S <n>    : Number of switch threads
-A <n>    : Number of app threads
```

Sample:



Note - If all you need to do is statically configure flows/groups etc in the Openflow switches, then you don't need to run any other component. This has to run all the time in order to run all other services/applications

So, let's see how to add a few flows/groups to a OF switch. We encourage people to make use of mininet http://mininet.org/ or LINC switch https://github.com/FlowForwarding/LINC-Switch (*which supports OF1.3.1*) for simulating virtual test environment.

```
## Open a separate terminal
$ telnet localhost 7000
```

```
dipj@kuldev-server: ~/mul-sf-code/mul
^Z
[1]+  Stopped                 sudo ./mul
dipj@kuldev-server:~/mul-sf-code/mul$ fg
sudo ./mul
^Cdipj@kuldev-server:~/mul-sf-code/mul$ fg
-bash: fg: current: no such job
dipj@kuldev-server:~/mul-sf-code/mul$ sudo ./mul
2013/11/11 13:23:52 2013/11/11 13:23:52 MUL-CONTROLLER: MUL-CONTROLLER: c_thread_event_loop: tid(3335956224)
c_thread_event_loop: tid(3327563520)
2013/11/11 13:23:52 MUL-CONTROLLER: c_thread_event_loop: tid(3310778112)
2013/11/11 13:23:52 MUL-CONTROLLER: c_thread_event_loop: tid(3302385408)
2013/11/11 13:23:52 MUL-CONTROLLER: c_thread_event_loop: tid(2952787712)
2013/11/11 13:23:52 MUL-CONTROLLER: mul-vty app registered
2013/11/11 13:23:52 MUL-CONTROLLER:  VTY THREAD RUNNING

2013/11/11 13:23:52 MUL-CONTROLLER: c_main_thread_final_init: running tid(3344348928)
2013/11/11 13:23:52 MUL-CONTROLLER: c_thread_event_loop: tid(3344348928)
2013/11/11 13:23:52 MUL-CONTROLLER: c_thread_event_loop: tid(3319170816)
2013/11/11 13:23:54 MUL-CONTROLLER: New switch to thread (1)

2013/11/11 13:23:57 unknown: Vty connection from 127.0.0.1
```

MUL Controller Core

```
dipj@kuldev-server: ~/mul-sf-code
dipj@kuldev-server:~/mul-sf-code$ telnet localhost 7000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

MUL-Controller (version 0.99).
Copyright (C) 2012 KULCLOUD LTD

kuldev-server> en
kuldev-server#
```

CLI shell for MUL controller core

One can check various switch connections and status using this cli shell:

```
dipj@kuldev-server: ~/mul-sf-code
dipj@kuldev-server:~/mul-sf-code$ telnet localhost 7000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

MUL-Controller (version 0.99).
Copyright (C) 2012 KULCLOUD LTD

kuldev-server> en
kuldev-server# sho
kuldev-server# show of-sw
kuldev-server# show of-switch
kuldev-server# show of-switch all

Switch-DP-id    |   State     |  Peer                 | Ports
-------------------------------------------------------------------------------

0x782bcb684d8d    Registered  127.0.0.1:41984            2
-------------------------------------------------------------------------------

kuldev-server# show of-switch de
kuldev-server# show of-switch 0x782bcb684d8d detail
-------------------------------------------------------------------------------

Datapath-id : 0x782bcb684d8d
OFP-Version : 0x4
Buffers     : 0
Tables      : 255
Actions     : 0x0
Capabilities: 0x4f(FLOW_STATS TABLE_STATS PORT_STATS STP QUEUE_STATS )
Num Ports   : 2
-------------------------------------------------------------------------------
                              Port info
-------------------------------------------------------------------------------
2      Port2      78:2b:cb:68:4d:91  PORT_UP LINK_UP
1      Port1      78:2b:cb:68:4d:8f  PORT_UP LINK_UP
-------------------------------------------------------------------------------

kuldev-server#
```

The following example adds a simple flow to redirect any flow from port 1 to port 2.

```
dipj@kuldev-server: ~/mul-sf-code
dipj@kuldev-server:~/mul-sf-code$
dipj@kuldev-server:~/mul-sf-code$
dipj@kuldev-server:~/mul-sf-code$
dipj@kuldev-server:~/mul-sf-code$ telnet localhost 7000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

MUL-Controller (version 0.99).
Copyright (C) 2012 KULCLOUD LTD

kuldev-server> en
kuldev-server# conf t
kuldev-server(config)# of-flow add switch 0x782bcb684d8d smac * dmac * eth-type * vid * vlan-pcp * dip 0.0.0.0/0
 sip 0.0.0.0/0 proto * tos * dport * sport * in-port 1 table 0
(config-flow-action)# action-add output 2
(config-flow-action)# com
(config-flow-action)# commit
kuldev-server(config)# exit
kuldev-server#
kuldev-server# show of-flow switch 0x782bcb684d8d
-------------------------------------------------------------------------------
Flow:  in-port:0x1
Actions: instructions: write-act: act-out-port(2):max-len(0x0),
Stats: Bytes 0 Packets 0  Bps 0.000000 Pps 0.000000
Prio:0 Table:0 Flags:static no-clone non-local Owner: mul-vty
-------------------------------------------------------------------------------
kuldev-server#
```

Flow delete is similar to the above example. One can discover many other commands by pressing “tab” in the cli.

Here is another example to add a static flow using groups. First add a group

```
dipj@kuldev-server: ~/mul-sf-code
dipj@kuldev-server:~/mul-sf-code$
dipj@kuldev-server:~/mul-sf-code$ telnet localhost 7000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

MUL-Controller (version 0.99).
Copyright (C) 2012 KULCLOUD LTD

kuldev-server> en
kuldev-server# conf te
kuldev-server(config)# of-group add switch 0x782bcb684d8d  group 1 type all
(config-grp-act-vectors)# action-add output 1
(config-grp-act-vectors)# group-act-vector-finish
(config-grp-act-vectors)# action-add output 2
(config-grp-act-vectors)# commit-group
kuldev-server(config)#
kuldev-server(config)# exit
kuldev-server# show of-switch 0x782bcb684d8d groups
-------------------------------------------------------------------------------
Group: 1 Type: all
Action bucket 0: act-out-port(1):max-len(0x0),
Action bucket 1: act-out-port(2):max-len(0x0),

-------------------------------------------------------------------------------
kuldev-server#
```

Now, add a flow with action as group output to group 1.

```
kuldev-server(config)#
kuldev-server(config)#
kuldev-server(config)#
kuldev-server(config)#
kuldev-server(config)#
kuldev-server(config)# of-flow add switch 0x782bcb684d8d smac * dmac * eth-type * vid * vlan-pcp * dip 0.0.0.0/0
 sip 0.0.0.0/0 proto * tos * dport * sport * in-port 1 table 0
(config-flow-action)# action-add group-id 1
(config-flow-action)# com
(config-flow-action)# commit
kuldev-server(config)#
```

```
kuldev-server(config)# do show of-flow switch 0x782bcb684d8d
------------------------------------------------------------------------------
Flow:  in-port:0x1
Actions: instructions: write-act: act-group:group-id(1),
Stats: Bytes 0 Packets 0  Bps 0.000000 Pps 0.000000
Prio:0 Table:0 Flags:static no-clone non-local Owner: mul-vty
------------------------------------------------------------------------------
kuldev-server(config)#
```

1.3.2 Running Mul topology/routing service

```
$ cd mul-sf-code/services/loadable/topo_routing/
$ sudo ./multr
```

Possible options to use -

```
-d                  : Daemon mode
-s <server-ip>      : Controller server ip address to connect
-V <vty-port>       : vty port address. (enables vty shell)
```

Please note that this service also provides its own cli-shell as the mul director/core component. To enable it, you have to specify any port number in command line using –V option.

```
$ sudo ./multr –V 8000
$ telnet localhost 8000          ## Access to topo-routing service cli
```

Sample:

```
dipj@kuldev-server: ~/mul-sf-code/services/loadable/topo_routing
dipj@kuldev-server:~/mul-sf-code/services/loadable/topo_routing$
dipj@kuldev-server:~/mul-sf-code/services/loadable/topo_routing$ sudo ./multr -V 8000
2013/11/11 14:22:04 MUL-CONTROLLER: tr_module_init
2013/11/11 14:22:04 MUL-CONTROLLER: mul_route_init
2013/11/11 14:22:04 MUL-CONTROLLER: mul_lldp_init
2013/11/11 14:22:04 MUL-CONTROLLER: Service Create mul-tr:12345
2013/11/11 14:22:04 MUL-CONTROLLER: lldp_port_add: adding 2 to switch 0x782bcb684d8d
2013/11/11 14:22:04 MUL-CONTROLLER: lldp_port_add: adding 1 to switch 0x782bcb684d8d
2013/11/11 14:22:04 MUL-CONTROLLER: lldp_vty_init: installing vty command
2013/11/11 14:22:04 MUL-CONTROLLER: route_vty_init: installing route vty command
2013/11/11 14:22:04 MUL-CONTROLLER:  App vty thread running

2013/11/11 14:22:08 MUL-CONTROLLER: mul_route_apsp_init_state:
```

Access service cli and check neighbor status of each switch (note the port num 8000 should be same as the command-line –V option given before hand) :

```
dipj@kuldev-server: ~/mul-sf-code
dipj@kuldev-server:~/mul-sf-code$
dipj@kuldev-server:~/mul-sf-code$
dipj@kuldev-server:~/mul-sf-code$ telnet localhost 8000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

MUL-Controller (version 0.99).
Copyright (C) 2012 KULCLOUD LTD

kuldev-server> en
kuldev-server# show lldp switch 0x782bcb684d8d detail
-------------------------------------------------------------------------------
      port # |     status | neighbor # | neighbor port #
-------------------------------------------------------------------------------
           2 |  LINK INIT | 0x         |
           1 |  LINK INIT | 0x         |
-------------------------------------------------------------------------------

kuldev-server#
```

1.3.3 Running Applications: l2switch

L2switch application provides bare-bones forwarding scheme based on l2 learning.

```
$ cd mul-sf-code/application/l2switch
```

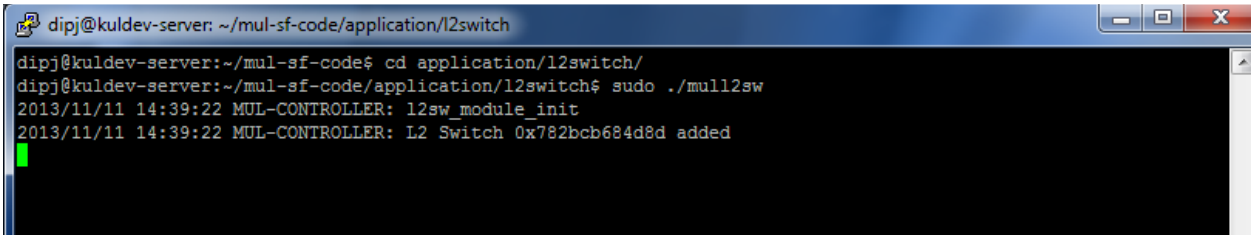
```
$ sudo ./mull2sw
```

Possible options to use -

```
    -d                  : Daemon mode
    -s <server-ip>      : Controller server ip address to connect
    -V <vty-port>       : vty port address. (enables vty shell)
```

Note that this application does not provide any meaningful cli so instead we can use director cli for getting any necessary info.

Sample :



Once we run l2switch app, network wide l2-switching takes place so if you have any hosts connected to an Openflow network they will start “pinging” each other.

Note – This does not support STP or similar algorithm yet. So, if you have loopy network, you should know you are in trouble.

1.3.4 Running Applications: fabric

The fabric app provides a slightly complex forwarding scheme wherein it provides a loop and learning free point to point virtual network connectivity between two or more hosts. Furthermore, it is multi-tenant aware. This application works wonders in tandem with orchestration software like Openstack. (Alas, the Openstack plugins are not yet open-sourced ☹). But never mind, fabric specific cli can make it up for this.

```
$ cd mul-sf-code/application/fabric
$ sudo ./mulfab –V 9000            ## Access to fabric cli on telnet port 9000
```

Possible options to use -

```
    -d                  : Daemon mode
    -s <server-ip>      : Controller server ip address to connect
    -V <vty-port>       : vty port address. (enables vty shell)
```

NOTE – This application needs topo-routing service to be also running.

Sample:

```
dipj@kuldev-server: ~/mul-sf-code/application/fabric
dipj@kuldev-server:~/mul-sf-code/application/fabric$ sudo ./mulfab -V 9000
2013/11/11 14:52:55 MUL-CONTROLLER: fabric_module_init
2013/11/11 14:52:55 MUL-CONTROLLER: Service Create mul-fab-cli:12346
2013/11/11 14:52:55 MUL-CONTROLLER: mul_route_service_get:
shm_open: Success
2013/11/11 14:52:55 MUL-CONTROLLER: fab_switch_add:switch (0x782bcb684d8d) added
2013/11/11 14:52:55 MUL-CONTROLLER: fab_port_add:switch (0x782bcb684d8d) port(2) added
2013/11/11 14:52:55 MUL-CONTROLLER: fab_port_add:switch (0x782bcb684d8d) port(1) added
2013/11/11 14:52:55 MUL-CONTROLLER: fabric_module_vty_init:
2013/11/11 14:52:55 MUL-CONTROLLER: fabric_vty_init: installing fabric vty command
2013/11/11 14:52:55 MUL-CONTROLLER:  App vty thread running
```

Following example shows how to add a host connected to an edge port of an OF switch which is a part of a OF switch mesh network or FAT tree or anything else. It works as long as there is a physical connectivity between any given pair of hosts.

```
dipj@kuldev-server: ~/mul-sf-code
dipj@kuldev-server:~/mul-sf-code$ telnet localhost 9000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

MUL-Controller (version 0.99).
Copyright (C) 2012 KULCLOUD LTD

kuldev-server> en
kuldev-server# add fabric-host tenant  0 network 0 host-ip 1.1.1.2 host-mac 00:01:02:03:05:07 swit
ch 0x782bcb684d8d port 1 non-gw
kuldev-server# add fabric-host tenant  0 network 0 host-ip 1.1.1.1 host-mac 00:01:03:04:05:ac swit
ch 0x1ebf546712ac port 2 non-gw
kuldev-server#
```

There you are!!! The two hosts can talk to each other now.

1.3.5 Running Applications: cli

So, far we have seen that each of application/service/core can start its own cli. So, one might guess what is this cli app all about. Basically, this provides a unified cli which

hooks in with each of the MuL controller components thereby giving us a single point of management. (That's one of the goals of SDN anyway to keep management simple)

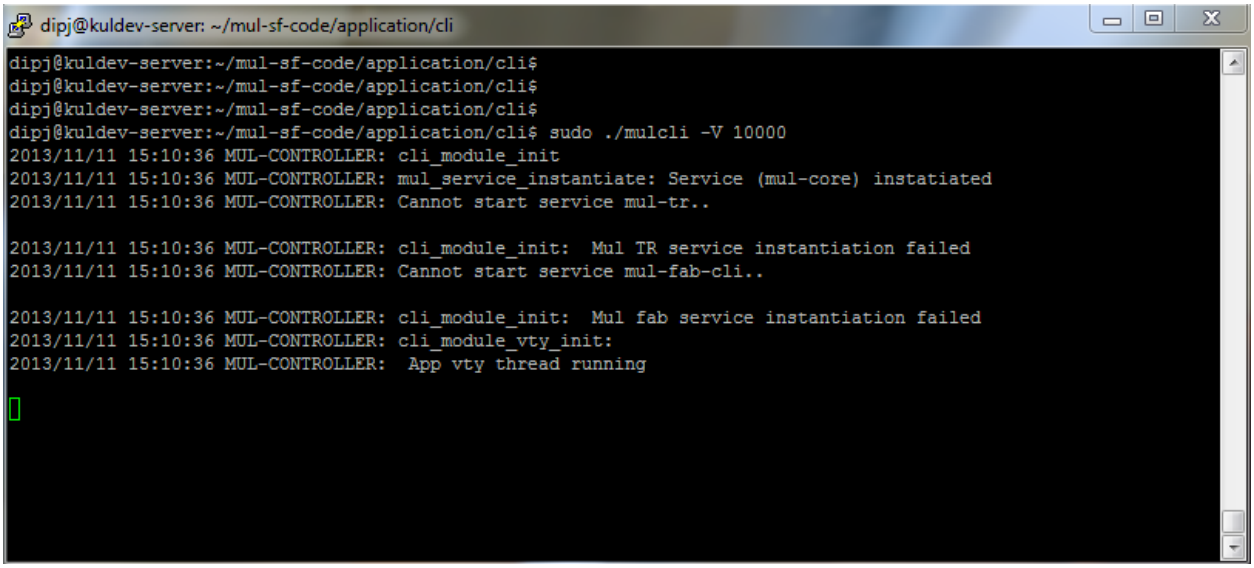
```
$ cd mul-sf-code/application/cli
$ sudo ./mulcli –V 10000            ## Access to fabric cli on telnet port 10000
```

Possible options to use -

```
    -d                  : Daemon mode
    -s <server-ip>      : Controller server ip address to connect
    -V <vty-port>       : vty port address. (enables vty shell)
```

NOTE – This application needs will auto-detect which of the mul's application are present.

Sample:



Following example shows how cli app is used to check switch's neighbor status (from topo-routing service) and then add a static flow to mul director/core module all from one place.

```
dipj@kuldev-server: ~/mul-sf-code
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

MUL-Controller (version 0.99).
Copyright (C) 2012 KULCLOUD LTD

kuldev-server> en
kuldev-server# sho
kuldev-server# show neigh switch 0x782bcb684d8d  deta
kuldev-server# show neigh switch 0x782bcb684d8d  detail
-----------------------------------------------------------------------------
      port # |     status | neighbor # | neighbor port #
-----------------------------------------------------------------------------
           2 |        EXT | 0x         0 | 0
           1 |        EXT | 0x         0 | 0
-----------------------------------------------------------------------------

kuldev-server# conf te
kuldev-server(config)# mul-conf
(mul-main)# of-flow add switch 0x782bcb684d8d smac 00:01:02:03:04:05 dmac 00:01:02:03:04:07 eth-ty
pe 2048 vid 2 vlan-pcp * dip 1.1.1.1/32 sip 2.2.2.2/24 proto 17 tos * dport 100 sport 200 in-port
1 table 0
(config-flow-action)# action-add output 2
(config-flow-action)# commit
(mul-main)# exit
kuldev-server# show of-fl
kuldev-server# show of-flow sw
kuldev-server# show of-flow switch
kuldev-server# show of-flow switch
kuldev-server# show of-flow switch 0x782bcb684d8d
kuldev-server# show of-flow switch 0x782bcb684d8d static
-----------------------------------------------------------------------------
Flow:  smac:00:01:02:03:04:05: dmac:00:01:02:03:04:07: eth-type:0x800  vlan-id:0x2  dst-ip:1.1.1.1
 (0xffffffff)  src-ip:2.2.2.0 (0xffffff00)  ip-proto:0x11  src-port:0xc8  dst-port:0x64  in-port:0
x1
instructions: write-act: act-out-port(2):max-len(0x0),
Prio: 0 Flags: static no-clone non-local Datapath-id: 0x782bcb684d8d
-----------------------------------------------------------------------------
kuldev-server# conf te
```

Note – You can type "tab" or "?" to list out all available cli commands

1.3.6 Running Applications: All at once

At times it is desirable to run a combination of a set of applications for a particular purpose and just have cli app do any management needs. So, we also have a utility script which makes it easier.

```
$ cd mul-sf-code
$ ./mul.sh start l2switch        ## Runs mul-core, l2switch and cli apps

 OR,

$ ./mul.sh start fabric          ## Run in mul-core, topo-route service, fabric and cli apps
```

If you are using this script to run mul components, please note that cli app is accessible using “telnet localhost 10000”.