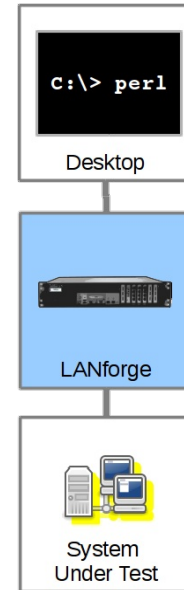


## Learn CLI commands used create Generic endpoints.

**Goal:** Compare and learn script and CLI commands used when creating and operating Generic endpoints.

Similar to the Layer3 perl script, `1f_firemod.pl`, the `1f_generic_ping.pl` script has been enhanced to use curl or other commands with parameter expansions. The `1f_curl.sh` script is a helper script that wraps curl commands and reports success or failure.

Introduced in LANforge 5.3.8.



---

## Example Commands

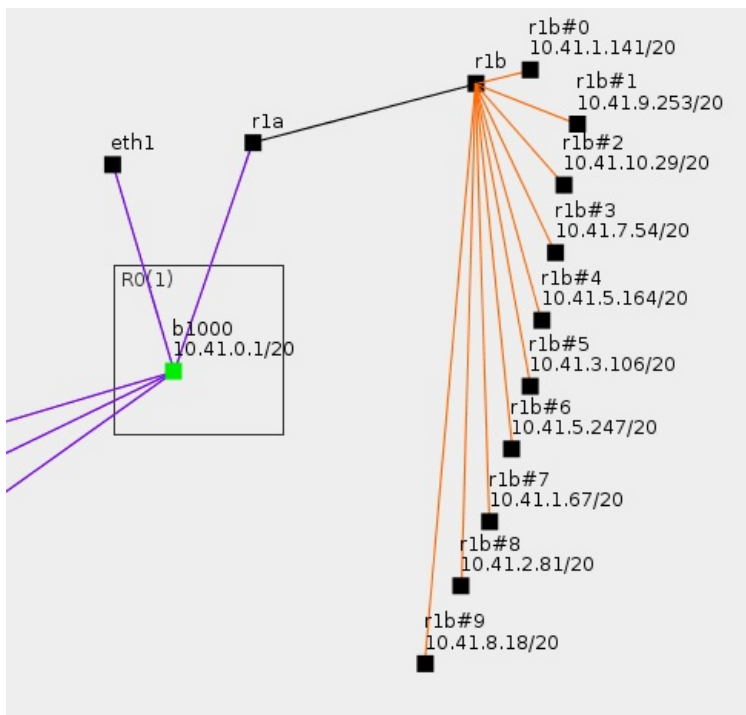
1. Creating ports and MAC VLANs
2. Creating ping endpoints
3. Creating curl endpoints
4. Using parameters for endpoint options
5. Using the `1f_curl.sh` script

## Setting for Examples

Generic endpoints are effectively one-legged connections. You can ping an IP or use curl to download web content. Both of these types of connections can be pointed back at the LANforge itself. We can operate these connections from redirect interfaces. The same techniques can apply to WiFi stations, of course.

### Create Redirects

1. Create a bridge `br0` including `eth1`
2. In Netsmith, select `br0`→Modify and enable DHCP
3. Create a redirect `r1a`, `r1b`
4. Add `r1a` to `br0`
5. Create 10 MAC VLANs based off `r1b` with DHCP enabled



6. After the MAC VLAN ports have addresses, you can verify that you can ping and download pages from LANforge webserver:

1. In a terminal, begin by sourcing `lanforge.profile`:

```
/home/lanforge$ . lanforge.profile
```

2. Ping the bridge from a MAC VLAN:

```
/home/lanforge$ ping -I 10.41.1.141 10.41.0.1
```

3. Grab the web page. (Apache is listening on all ports by default.)

```
/home/lanforge$ curl -sq --interface 10.41.1.141 http://10.41.0.1/
```

## Creating ping endpoints

In the `/home/lanforge/scripts` directory, the `lf_generic_ping.pl` script creates a wrapped ping command by default. There is another script, `lfping`, that reports ping results to LANforge. Here is an example of creating 1 ping endpoint:

```
/home/lanforge/scripts$ ./lf_generic_ping.pl --mgr localhost --resource 1 \
--dest 10.41.0.1 --interface r1b#0 --name pingtest
```

Create a ping endpoint for every MAC VLAN parented by `r1b`:

```
$ ./lf_generic_ping.pl --mgr localhost --resource 1 \
--dest 10.41.0.1 --parent r1b
```

Create a ping endpoint for every virtual station parented by `wiphy0`:

```
$ ./lf_generic_ping.pl --mgr localhost --resource 1 \
--dest 10.41.0.1 --radio wiphy0
```

Create a ping endpoint for every MAC VLAN beginning with `r1b#1` (`r1b#1`, `r1b#10`):

```
$ ./lf_generic_ping.pl --mgr localhost --resource 1 \
--dest 10.41.0.1 --match 'r1b#1'
```

## Creating curl endpoints

To use other commands with the script, you can create a `--cmd` parameter. You can use `curl` directly if desired, but `curl`'s output is not formatted well for LANforge to understand. By default, commands do not understand what port or IP they

should be interacting as. We need to provide special parameters to help.

## Parameter expansion

The `lf_generic_ping.pl` script will look for these tokens in the `--cmd` parameter:

- `%i` Expands to the IPv4 address of the port.
- `%d` Expands to the destination hostname or address
- `%p` Expands to the port name

## The curl wrapper script

The `scripts/lf_curl.sh` script is a wrapper for curl that detects success or failure, and can operate the request in a loop. Expandable parameters are expanded by `lf_generic_ping.pl`, not `lf_curl.sh`.

You can use `lf_curl.sh` from the command line to test it out:

```
$ ./lf_curl.sh -i 10.41.1.141 -p r1b#1 -o /tmp/output -d http://example.com/
```

Executes:

```
curl -sqLk --interface 10.0.0.1 -o /tmp/output_r1b#1 http://example.com/
```

So it is best used from `lf_generic_ping.pl` to construct commands referencing this script:

```
./lf_generic_ping.pl --mgr localhost --resource 1 \  
  --name curl_ex_ --match 'r1b#' --dest http://10.41.0.1/ \  
  --cmd 'lf_curl.sh -o /tmp/curl_%p.out -i %i -d %d -p %p'
```