Providing HTTP Service on a Port

**Goal:** Configure and activate a http server bound to a specific port in LANforge.

This is useful if you want to provide an application layer target to interact with. This cookbook assumes you can access a shell prompt on the LANforge computer; the command-line instructions can be done at the system console, in a terminal over a remote desktop connection, or over ssh. The nginx service is only available on the Linux version of LANforge server.

**Note:** There are two web servers installed in a LANforge computer. The default web server is a stock version of Apache HTTPD that responds to all port 80 requests on all interfaces. For testing we recommend running Candela Technologies' version of Nginx on specific ports as covered in this cookbook. In contrast, Apache cannot bind to a network-device therefore we do not suggest running it.

Nginx can also be configured to listen to IPv6 traffic. Jump to the end of the cookbook for that technique.

---

1. Stop and disable LANforge system Apache instance.
   A. Log into the LANforge computer as user lanforge

   ![Terminal]
   - jreynolds@jed-shuttle:~ $ sudo ssh lanforge@192.168.100.40

   ![Terminal]
   - Become super-user using the command su - (or sudo -s)

   ![Terminal]
   - root@jed-f20:/home/lanforge:~ $ sudo -s
   - [root@jed-f20 lanforge]#
C. Stop the Apache service with `systemctl stop httpd`.

D. If you want this setting to persist after a reboot, disable the service: `systemctl disable httpd`.

E. Set this as system default with: `systemctl daemon-reload`.

2. (Optional) There are other options for running Apache if you want to have both web servers available. You would not need stop and disable Apache, just restart it. You can change Apache to:
   
   A. ...listen to a different port (like 81). Edit `/etc/httpd/conf/httpd.conf` and change the `Listen` option.
3. The demonstration test we'll create is making two ports, one for making requests and the other one for serving the protocol. Let's create two ports:
   A. Create a redirect device:
      A. In the Ports tab, click the Create button
   B. Select Redirect
   C. Enter rd0a for #1 Redirect name.
   D. and rd0b for #2 Redirect name.
   E. Click Apply and then Cancel to close the window.
B. Configure rd0a as the service port:
   A. In the Ports tab, double-click the row for port rd0a

B. The Configure Settings window will appear
C. Set an IP of 10.2.0.1,
D. a netmask of 255.255.255.0
E. and the report time to faster [1 s].
F. Click OK
C. Configure rd0b as a client port

A. In the Ports tab, double-click the row for port rd0b

<table>
<thead>
<tr>
<th>Port</th>
<th>Phy</th>
<th>Down</th>
<th>IP</th>
<th>SEC</th>
<th>Alias</th>
<th>Parent Dev</th>
<th>RX Bytes</th>
<th>RX Pkt</th>
<th>TX Bytes</th>
<th>TX Pkt</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td></td>
<td>192.168.100.40</td>
<td>0</td>
<td>eth1</td>
<td></td>
<td>1,857,907</td>
<td>18,473</td>
<td>7</td>
<td>5,938</td>
<td>3,696,697</td>
</tr>
<tr>
<td>1.1</td>
<td></td>
<td>0.0.0</td>
<td>0</td>
<td>eth2</td>
<td></td>
<td>1,073,234</td>
<td>12,067</td>
<td>3</td>
<td>2,052</td>
<td>2,842</td>
</tr>
<tr>
<td>1.2</td>
<td></td>
<td>0.0.0</td>
<td>0</td>
<td>eth3</td>
<td></td>
<td>87,773</td>
<td>353</td>
<td>0</td>
<td>0</td>
<td>2,842</td>
</tr>
<tr>
<td>1.3</td>
<td></td>
<td>1.0.0.0</td>
<td>0</td>
<td>rd0a</td>
<td></td>
<td>728</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1,296</td>
</tr>
<tr>
<td>1.4</td>
<td></td>
<td>0.0.0</td>
<td>0</td>
<td>rd0b</td>
<td></td>
<td>1,366</td>
<td>17</td>
<td>0</td>
<td>54</td>
<td>738</td>
</tr>
</tbody>
</table>

B. The Configure Settings window will appear

C. Set an IP of 10.2.0.3,

D. a netmask of 255.255.255.0

E. and the report time to faster [1 s],

F. Click OK

4. Enable HTTP service on your upstream port using the following steps.
A. Enable the nginx service on port rd0a:
   A. In the Ports tab, double-click the row for port rd0a

   ![LANForge Manager](image)

   B. The Configure Settings window will appear
   C. In the lower left column of the window, enable HTTP option.
   D. Click OK and then close the window.

   ![rd0a (jed-f20) Configure Settings](image)

   B. (Optional) Modify the nginx config file for port rd0a. This example assumes you are logged into the LANforge system console, or have connected to the desktop using a remote-desktop client like vncviewer. If you are familiar with editing from the command-line, you will likely know how to do this via ssh.
   A. From the desktop Accessories menu, select Terminal Emulator,
   B. Change to the LANforge nginx directory: cd /home/lanforge/vr_conf

   ```bash
   [lanforge@jed-f20 ~]$ cd
   [lanforge@jed-f20 -]$ cd vr_conf
   [lanforge@jed-f20 vr_conf]$ ls -l conf
   nginx_rd0a.conf  radvd.conf
   [lanforge@jed-f20 vr_conf]$ ```
C. Edit the file `nginx_rd0a.conf`, if you do not see the file, you might have connected to the wrong LANforge resource, or you might not have clicked OK/Apply when enabling the Configure Settings window for port rd0a.

D. You will see that the listen directive is already set to `10.2.0.1:80 bind_dev=rd0a;`

E. If you want to change settings, delete the first line of the config file as part of your changes. This will signal LANforge not to overwrite the file.
C. To apply the changes to the nginx service on this port:
   A. Disable the HTTP service in the rd0a Configure Settings window.

   ![rd0a Configure Settings](image)

   B. Click the **Apply** button.

   C. Enable HTTPD service.

   ![rd0a Configure Settings](image)

   D. Click the **Apply** button.

   E. Click the **Cancel** button to close the window if you are done.
D. A quick way to add a file into the nginx document root folder is to symlink the system dictionary there. You will need to be in a terminal or unix shell of the resource running nginx (192.168.100.40 in this example).

   A. Change the ownership of the document root directory to user lanforge:

   B. `sudo chown lanforge:lanforge /usr/local/lanforge/nginx/html`

   C. `cd /usr/local/lanforge/nginx/html`

   D. `ln -s /usr/share/dict/linux.words .`

E. (Optional) To load a larger payload (such as a DVD file), you can use Filezilla (or scp) to copy the file over. You will want to have changed the nginx document root directory owner to lanforge (as above):
1. Quick connect to sftp://192.168.100.40 (also named jed-f20 in this example)

2. Using username lanforge and password lanforge

B. In the left column, choose your local directory (Y:\downloads)
C. In the right column, type in the nginx document root, `/usr/local/lanforge/nginx/html` and hit enter.

D. In the left column, double-click your DVD image (Fedora-20-x86_64-netinst.iso)
5. Configure application-layer traffic using the Layer 4-7 tab tools:
   A. In the Layer 4-7 tab, click the Create button.
   
   B. You will see the Create/Modify L4Endpoint window.
   
   C. Create a connection named web-requests.
   
   D. With a Report timer of fast (1 s)
   
   E. from Resource (jed-20) port rd0b.
   
   F. Let's make ten requests a second. Enter 6000 for URLs per 10m.
   
   G. Jump down to the URL field, and enter http://10.2.0.1/index.html.
   
   H. and to avoid saving the result, set the Dest File to /dev/null.
6. Running the traffic is simple:
   A. Select the row named web-requests in the Layer 4-7 table.
   B. Click the Start button.
C. Click the Display button to see more detail on the traffic:

7. It is also possible to verify your connection setup from the command-line:

A. To verify the Layer 3 connection, you can use ping:
   ```
   ping -I 10.2.0.3 10.2.0.1
   ```

B. To see that nginx is listening on port 80, use netstat on the resource running nginx and see what is listening on port 80:
   ```
   sudo netstat -ntlp | grep ':80'
   ```

C. To see the output of the web request, you can use LANforge's version of curl located in `/home/lanforge/local/bin/curl`. Since it uses bundled libraries, first type:
   ```
   cd /home/lanforge && ./lanforge.profile
   ```
   This sets up all environment variables and paths to work with the binaries and libraries shipped with LANforge.
D. Verify it runs without complaint:

```bash
/home/lanforge/local/bin/curl --help
```

E. If you scroll up in your terminal, you will see options available in the output not found in a normal version of curl, like --dns_server, --dns_interface, --dns_ip4_addr, --dns_ip6_addr, --interface and --localaddr. We will use --interface next.

F. Craft your download command:

```bash
cd /home/lanforge/local/bin
./curl --interface 10.2.0.3 'http://10.2.0.1/index.html'
```
G. You will see the html output in the terminal:

```
[lanforge@jed-f20 -] $ cd /home/lanforge/local/bin
[lanforge@jed-f20 bin]$ ./curl --interface 10.2.6.3 http://10.2.0.1/index.html

<!DOCTYPE html>
<html>
<head>
<style>
  body {
    width: 35em;
    margin 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and working. Further configuration is required.</p>
<p>For online documentation and support please refer to:
</p>
<p>Commercial support is available at:
</p>
<p>Thank you for using nginx.</p>
</body>
</html>
```

H. There will be a record of connections and errors in the nginx logs directory: cd /usr/local/lanforge/nginx/logs

```
[lanforge@jed-f20 logs]$ cd /usr/local/lanforge/nginx/logs
[lanforge@jed-f20 logs]$ pwd
/usr/local/lanforge/nginx/logs
[lanforge@jed-f20 logs]$ ls -lta
total 604
-rw-r--r--  1 root   root   0 Sep 30 14:24 error.log
-rw-r--r--  1 root   root  82 Sep 30 14:24 rd8a_error.log
-rw-r--r--  1 root   root  83 Sep 30 14:24 rd8a_access.log
drwxr-xr-x 11  500  500  4096 Sep 30 14:24 ...
-rw-r--r--  1 root   root   0 Sep 30 14:24 ...
-rw-r--r--  2  500  500  4096 Sep 30 14:24 ...
-rw-r--r--  1 root   root  807684 Sep 30 15:20 rd8a_host.access.log
[lanforge@jed-f20 logs]$ 
```

8. [Advanced] Creating IPv6 traffic to nginx. We will assume a scenario where eth1 will serve nginx requests and eth2 will generate requests.

   A. Add IPv6 addresses to your ports. We will add
   - `ee::2:1/120` for eth1,
   - `ee::2:2/120` for eth2,
   - and set the gateways for them to `ee::2:fe`. We won’t actually use a gateway, but our port configuration requires it.
A. Here is eth1, and we will enable the HTTP service as well.

```
eth1 (jed-21.candelaitech.com) Configure Settings
```

8. Here is eth2, notice how we added an IPv4 address to the port. Do that so that the Create Layer 4-7 dialog does not complain.

```
eth2 (jed-21.candelaitech.com) Configure Settings
```

C. Here is a picture of the Port Mgr tab, with ports on two resources configured.

```
LANforge Manager Version 3.4
```

D. Verify your connection between ports using `ping6`: `ping6 -I ee::2:2 eth2 ee::2:1`
B. Edit your `/home/lanforge/vr_conf/nginx_eth1.conf` file. You will remove the header comment and alter the `listen` statements to include IPv6 addresses. You can listen for `[:1]` or you could listen for `[ee::2:1]`

```bash
worker_processes 1;
error_log logs/eth1_error.log;
pid /home/lanforge/vr_conf/nginx_eth1.pid;

events {
  worker_connections 1024;
}

http {
  include /usr/local/lanforge/nginx/conf/mime.types;
  default_type application/octet-stream;
  access_log logs/eth1_access.log;
  sendfile on;
  keepalive_timeout 65;

  server {
    listen 0.0.0.0:80 bind dev=eth1;
    listen [:1]:80 bind dev=eth1;
    Server_name www-eth1;
    access_log logs/eth1_host.access.log;

    location / {
      root html;
      index index.html index.htm;
    }

    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
      root html;
    }
  }
}
```

C. If you provide an IPv6 address, please watch the nginx log file for errors; listening on an IPv6 address is easy to misconfigure. The log files can be found in `/usr/local/lanforge/nginx/logs`.

D. Re-start the `http` service on `eth1` by un-checking HTTP, clicking Apply, checking HTTP, clicking OK.

E. Configure your Layer 4-7 endpoint:

A. Name your endpoint
B. Set your URLs per Minute
C. Specify the URL with brackets around the address: `http://[ee::2:1]/index.html`
D. Set your output file to `/dev/null`

F. You can watch traffic on `eth1` or `eth2` to verify the web requests.