Fiddling with FreeBSD - Jails

In the previous post, when we created the demo3 jail, we encountered an ERROR related to /dev/random. Let's resolve this issue and streamline our syntax in jail.conf to keep things neat and efficient.

```
demo3 {
   host.hostname="${name}jail4";
   ip4.addr=192.168.15.13/24;
   interface=em0;
   path="/jail/demo3";
   exec.start="/bin/sh /etc/rc";
   exec.stop="/bin/sh /etc/rc.shutdown";
}
```

1

Changes highlighted in green indicate comments, three different methods for adding comments to our jail.conf file.

Changes highlighted in red indicate modifications made to our existing jail.conf. Additionally, we have separated parameter related to the each jail.

The first change we made is substituting the jail path with the variable ${name}.

Second change to mount devfs under jail "dev" directory, we also specify that what devices should be visible to jail with devfs_ruleset=4

Third change we made is overriding the common parameter to mount.devfs=0, note you can use TRUE OR FALSE, 0 OR 1 for boolean parameters, this jail will complain about /dev/random

/etc/jail.conf

```
# Common parameters for my jails

host.hostname="${name}.home.arpa";
interface=em0;
path="/jail/${name}";
mount.devfs=true;
devfs_ruleset=4;
exec.start="/bin/sh /etc/rc";
exec.stop="/bin/sh /etc/rc.shutdown";

// jail related parameters

demo3 {
   ip4.addr=192.168.15.13/24;
}

demo2 {
   ip4.addr=192.168.15.12/24;
   mount.devfs=0;
   devfs_ruleset="";
}

/* Let's add one more ip to
 * my demo1 jail
 */

demo1 {
   ip4.addr=192.168.15.11/24;
   ip4.addr+=192.168.15.14/24;
}
```

The last change, indicated by "+=", signifies that we can add different values for the same parameter. Alternatively, you can use comma-separated values as well.

2

Starting with 14.0-RELEASE, you can now use the ".include" directive inside jail.conf. This allows you to have separate configuration files for each jail.

We have added two more parameters here: exec.clean=true. Note that you can exclude "true" here and it will still work. this parameter denotes that commands are executed in a clean environment.

The second parameter, exec.consolelog, helps you log command output to a file.

How about having dedicated interface for my demo2 jail, hence interface=em1.

```
#jail -vc demo1 <-- create one jail
#jail -vc       <-- create all jails
```

OR

```
#sysrc jail_enable="YES"
#service jail start demo1
#service jail start
```

/etc/jail.conf

```
# Common parameters for my jails
.include "/etc/jail.conf.d/*.conf";
exec.clean=true;
exec.consolelog="/var/log/j_${name}.log
host.hostname="${name}.home.arpa";
interface=em0;
path="/jail/${name}";
mount.devfs=true;
devfs_ruleset=4;
exec.start="/bin/sh /etc/rc";
exec.stop="/bin/sh /etc/rc.shutdown";
```

/etc/jail.conf.d/demo3.conf

```
demo3 {
   ip4.addr=192.168.15.13/24;
}
```

/etc/jail.conf.d/demo2.conf

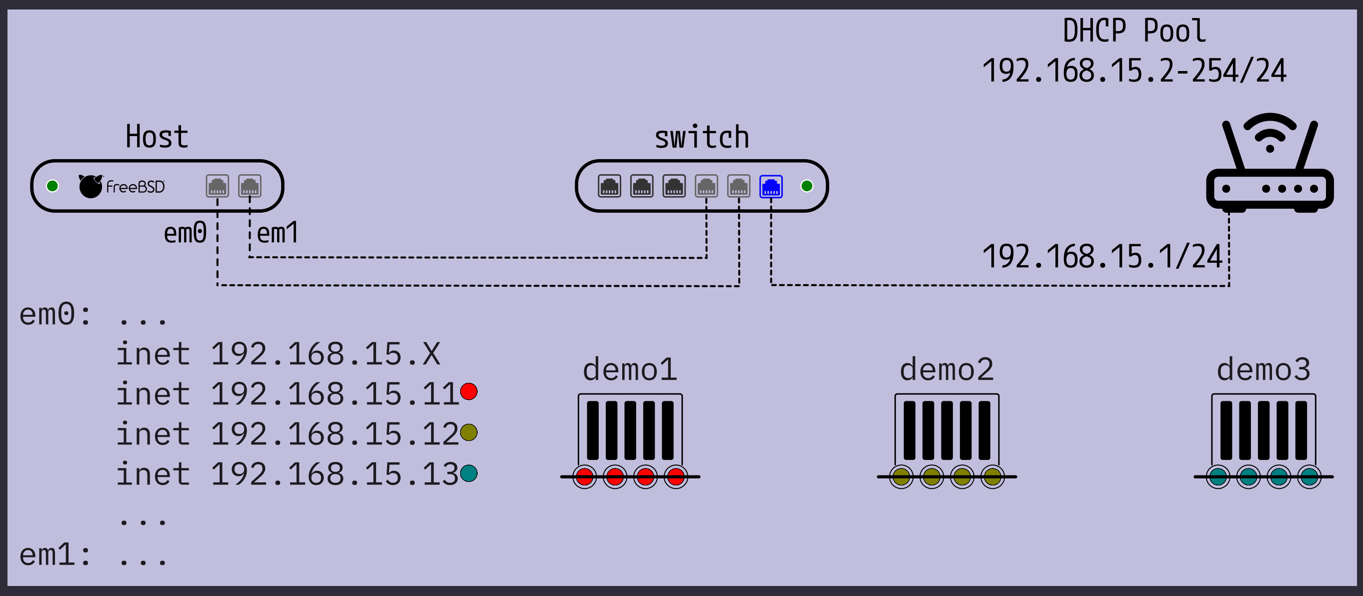
```
demo2 {
   ip4.addr=192.168.15.12/24;
   interface=em1;
}
```

/etc/jail.conf.d/demo1.conf

```
demo1 {
   ip4.addr=192.168.15.11/24;
   ip4.addr+=192.168.15.14/24;
}
```

3

Currently, we are sharing the network stack with the host system, but FreeBSD offers a virtualized network stack called VNET that allows each jail to have its own independent network stack. The kernel must be compiled with the VIMAGE option.



/etc/jail.conf

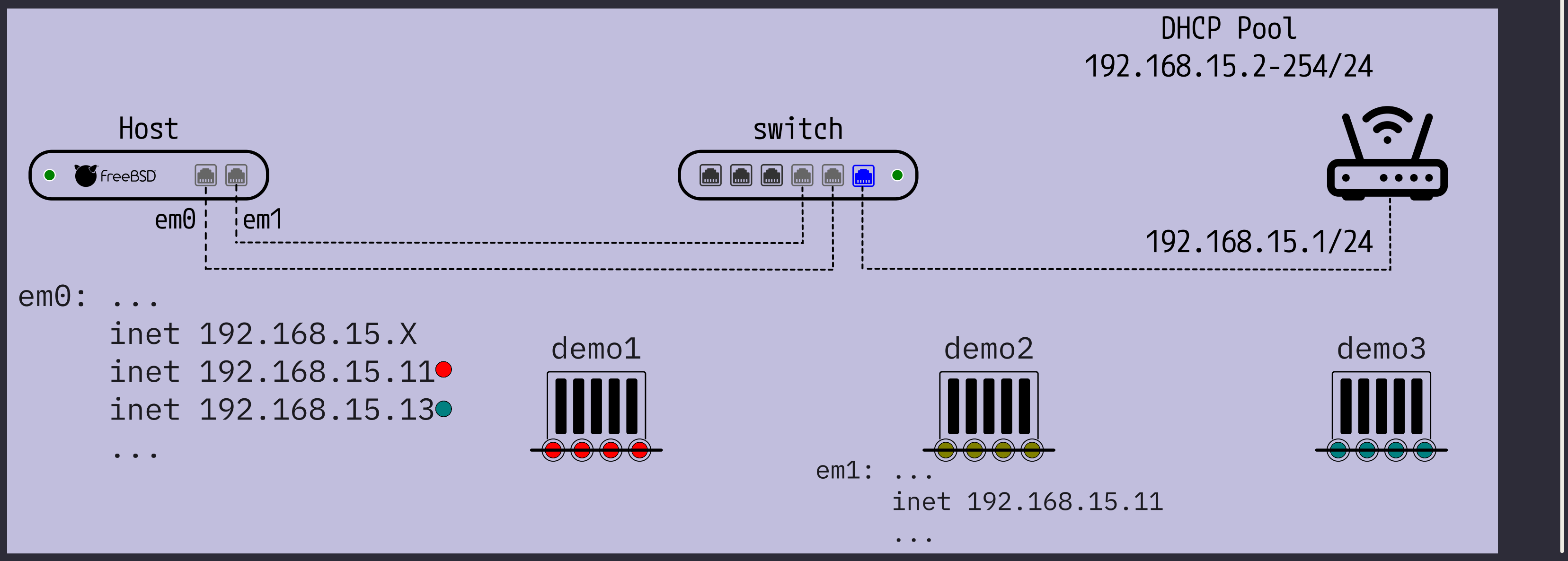
```
# Common parameters for my jails

...; # <- syntax excluded for space

demo3 {
   ip4.addr=192.168.15.13/24;
}

demo2 {
   vnet=new;
   vnet.interface=em1;
}

demo1 {
   ip4.addr=192.168.15.11/24;
}
```

Changes highlighted in red indicate that we will be using vnet=new paramter, note that we can exclude 'new' altogether and still it will work.

We have also specified the second parameter, vnet.interface=em1, to designate which interface should be used for this jail.

Make sure that you configure IP address in '/jail/demo2/etc/rc.conf' of demo2 jail and create the jail with 'jail -vc demo2'

Note that once the jail is created, em1 will be removed from the host network stack, as illustrated in the diagram below.



4

As you may have noticed, with VNET, the interface no longer belongs to the host but instead to the associated jail. It will return to the host's network stack once the jail is destroyed.

There is also a virtual interface called 'epair(4)'. As the name suggests, it is a pair of Ethernet interfaces that connect two ends together, similar to how a physical Ethernet cable connects two computers.

Epairs will resolve our issue by linking one end of the pair to the jail and the other end to the host, thus enabling communication between the two.

```
# ifconfig epair create
```

You should have now two interfaces epair0a and epair0b, you can use same name or rename them as per your jail application.

Let us assign ip address to the epair0a and assign epair0b to demo1 jail.

```
# ifconfig epair0a 172.16.15.1/24 up
```

Changes highlighted in red indicates that we have assigned **epair0b** to demo1 jail. Let us create the demo1 jail.

```
# jail -vc demo1
```

Let us go inside the demo1 jail and assign ip address to **epair0b**.

```
# jexec demo1 /bin/sh
```

```
# ifconfig epair0b 172.16.15.2/24
```

You could also have assigned ip adress from host with `# ifconfig -j demo1 epair0b <ipaddress>`

Run the ifconfig command again and verify ip address is assigned, try ping to 172.16.15.1 from jail.

Exit from the jail and try ping to 172.16.15.2 from host.

/etc/jail.conf

```
# Common parameters for my jails
.include "/etc/jail.conf.d/*.conf";
exec.clean=true;
exec.consolelog="/var/log/j_${name}.log
host.hostname="${name}.home.arpa";
interface=em0;
path="/jail/${name}";
mount.devfs=true;
devfs_ruleset=4;
exec.start="/bin/sh /etc/rc";
exec.stop="/bin/sh /etc/rc.shutdown";
```

/etc/jail.conf.d/demo3.conf

```
demo3 {
   ip4.addr=192.168.15.13/24;
}
```

/etc/jail.conf.d/demo2.conf

```
demo2 {
   vnet=new;
   vnet.interface=em1;
}
```

/etc/jail.conf.d/demo1.conf

```
demo1 {
   vnet=new;
   vnet.interface="epair0b";
}
```