Table Of Contents

Table Of Contents 1
How do I load lagg driver? 2
A note about custom FreeBSD kernels 2
How do I create a lagg interface? 2
An Example - Create link aggregation using LACP 3
How do I create an active failover interface? 3
How do I make link aggregation configuration persistent? 3
References: 4

nixCraft: Linux Tips, Hacks, Tutorials, And Ideas In Blog Format
http://www.cyberciti.biz/

Home > Faq > FreeBSD

FreeBSD: NIC Bonding / Link Aggregation / Trunking / Link Failover Tutorial

Posted by Vivek Gite <vivek@nixcraft.com>

[1]

I've two Intel gigabit network card installed in HP server. I know how to setup bounding under CentOS Linux [2], but I'd like to do same under FreeBSD. How do I setup link aggregation of multiple network interfaces as one virtual trunk interface for the purpose of providing fault-tolerance and high-speed links under FreeBSD 7.x server?

FreeBSD has lagg - link aggregation and link failover interface. The lagg interface allows aggregation of multiple network interfaces as one virtual lagg interface for the purpose of providing fault-tolerance and high-speed links.

How do I load lagg driver?

First, you need to load the lagg driver into the memory. To load the driver as a module at boot time, place the following line in loader.conf:

```
if_lagg_load="YES"
```

Type the following command to add line to /boot/loader.conf file, enter:

```
# echo 'if_lagg_load="YES"' >> /boot/loader.conf
```

To load driver for current session use kldload command [3], type:

```
# kldload if_lagg
# kldstat
```

A note about custom FreeBSD kernels

If you have custom compiled kernel, you need to compile this driver into the kernel, place the following line in your FreeBSD kernel configuration file:

```
device lagg
```

How do I create a lagg interface?

A lagg interface can be created using the following command:

```
# ifconfig laggN create
# ifconfig lagg0 create
```

It can use different link aggregation protocols specified using the laggproto proto option. The driver currently supports the following aggregation protocols:

Aggregation Protocols	Description
failover	Sends and receives traffic only through the master port. If the master port becomes unavailable, the next active port is used. The first interface added is the master port; any interfaces added after that are used as failover devices.
fec	Supports Cisco EtherChannel. This is a static setup and does not negotiate aggregation with the peer or exchange frames to monitor the link.
lacp	Supports the IEEE 802.3ad Link Aggregation Control Protocol (LACP) and the Marker Protocol. LACP will negotiate a set of aggregable links with the peer in to one or more Link Aggregated Groups. Each LAG is composed of ports of the same speed, set to full-duplex operation. The traffic will be balanced across the ports in the LAG with the greatest total speed, in most cases there will only be one LAG which contains all ports. In the event of changes in physical connectivity, Link Aggregation will quickly converge to a new configuration.

loadbalance	Balances outgoing traffic across the active ports based on hashed protocol header information and accepts incoming traffic from any active port. This is a static setup and does not negotiate aggregation with the peer or exchange frames to monitor the link. The hash includes the Ethernet source and destination address, and, if available, the VLAN tag, and the IP source and destination address.
roundrobin	Distributes outgoing traffic using a round-robin scheduler through all active ports and accepts incoming traffic from any active port.
none	This protocol is intended to do nothing: it disables any traffic without disabling the lagg interface itself.

An Example - Create link aggregation using LACP

To create a 802.3ad link aggregation using LACP with two em Intel PRO/1000 Gigabit Ethernet adapter driver Gigabit Ethernet interfaces use the following procedure. First, you do not need to assign any IPs to em0 and em1 interfaces. Do not configure them via /etc/rc.conf. Next, bring up both interfaces, enter:

```
# ifconfig em0 up
# ifconfig em1 up
```

Create a lagg interface called lagg0, enter:

```
# ifconfig lagg0 create
```

Finally, assign an IP address to lagg0:

```
# ifconfig lagg0 laggproto lacp laggport em0 laggport em1 10.24.116.2 netmask
255.255.255.192
```

If you need to create failover between NIC instead of lacp , enter:

```
# ifconfig lagg0 up laggproto failover laggport em0 laggport em1 10.24.116.2 netmask
255.255.255.192
```

To verify status of your links, enter:

```
# ifconfig
```

You may also need to add a default gateway [4], enter:

```
# route add default 10.24.116.100
# netstat -nr
```

Where,

- **lagg0** : Bounding interface name.
- **laggproto lacp** : Bounding protocol. See above tables for possible values.
- **laggport em0** : Your first Ethernet interface name.
- **laggport em1** : Your second Ethernet interface name.
- **10.24.116.2** : IP address for lagg0 interface.
- **netmask 255.255.255.192** : Netmask address for lagg0 interface.

How do I create an active failover interface?

The following example uses an active failover interface to set up roaming between wired and wireless networks using two network devices. Whenever the wired master interface is unplugged, the wireless failover device will be used:

```
# ifconfig em0 up
# ifconfig ath0 nwid my_ssid up
# ifconfig lagg0 laggproto failover laggport em0 laggport ath0 192.168.1.1 netmask
255.255.255.0
```

How do I make link aggregation configuration persistent?

To make link aggregation configuration persistent use cloned_interfaces variable in /etc/rc.conf. Open /etc/rc.conf file, enter:

```
# vi /etc/rc.conf
```

Append the following configuration:

```
ifconfig_em0="up"
ifconfig_em1="up"
cloned_interfaces="lagg0"
ifconfig_lagg0="laggproto failover laggport em0 laggport em1"
ipv4_addrs_lagg0="10.24.116.2/32"
defaultrouter="10.24.116.100"
```

Save and close the file. Now, FreeBSD will remember your configuration after the system reboot.

References:

The research for this FAQ is based upon the following man pages:

- rc.conf, ifconfig, and lagg

Article printed from Frequently Asked Questions About Linux / UNIX: **http://www.cyberciti.biz/faq/**

URL to article: **http://www.cyberciti.biz/faq/freebsd-network-link-aggregation-trunking/**

URLs in this post:

[1] Image: **http://www.cyberciti.biz/faq/category/freebsd/**
[2] bounding under CentOS Linux: **http://www.cyberciti.biz/tips/linux-bond-or-team-multiple-network-interfaces-nic-into-single-interface.html**
[3] kldload command: **http://www.cyberciti.biz/tips/freebsd-how-to-load-device-driver-module-in-kernel.html**
[4] default gateway: **http://www.cyberciti.biz/faq/freebsd-setup-default-routing-with-route-command/**