

## 网络组Network Teaming

网络组是将多块网卡逻辑地连接到一起从而允许故障转移或者提高吞吐率的方法。提高服务器网络可用性的一个方式是使用多个网卡。Linux 绑定驱动程序提供了一种将多个网卡聚合到一个逻辑的绑定接口的方法。这是个新的实现绑定的方法，并不影响linux内核中旧绑定驱动。

- 网络组：是将多个网卡聚合在一起方法，从而实现冗错和提高吞吐量
- 网络组不同于旧版中bonding技术，提供更好的性能和扩展性
- 网络组由内核驱动和teamd守护进程实现。
- 多种方式runner：

broadcast：传输来自所有端口的每个包

roundrobin：以轮询方式传输来自每个端口的包

activebackup：故障转移，监视链接更改并选择活动端口进行数据传输

loadbalance：监控流量并使用哈希函数以尝试在为包传输选择端口时达到完美均衡

lacp(implements the 802.3ad Link Aggregation Control Protocol)

实施802.3链路聚合控制协议

- 启动网络组接口不会自动启动网络组中的port接口
- 启动网络组接口中的port接口始终会启动合作接口
- 禁用网络组接口会自动禁用网络组中的port接口
- 没有port接口的网络组接口可以启动静态IP连接
- 启用DHCP连接时，没有port接口的网络组会等待port接口的加入

## 创建网络组接口

nmcli命令可用于创建和管理组和端口接口。以下4个步骤用于创建和激活网络组接口：

- 1、创建组接口
- 2、确定组接口的IPv4和/或IPv6属性
- 3、分配端口接口
- 4、启动/关闭组接口和端口接口

创建组接口命令格式：组接口默认使用dhcp获取IP参数

```
nmcli con add type team con-name CNAME ifname INAME [configJSON]
```

CNAME：连接名， INAME：接口名

JSON指定runner方式

格式： '{"runner": {"name": "METHOD"}}'

METHOD可以是broadcast,roundrobin,  
activebackup,loadbalance, lacp

分配端口接口命令格式：

```
nmcli con add type team-slave con-name CNAME ifname INAME master TEAM
```

CNAME: 连接名

INAME: 网络接口名

TEAM: 网络组接口名

连接名若不指定, 默认为team-slave-IFACE

启动/关闭组接口和端口接口

```
nmcli dev dis INAME
```

```
nmcli con up CNAME
```

INAME设备名 CNAME网络组接口名或port接口

### 实验:创建网络组

1、创建网络组team0, 会在/etc/sysconfig/network-scripts/目录下生成ifcfg-team0文件

```
# nmcli connection add con-name team0 type team ifname team0 config  
'{"runner":{"name":"activebackup"}}' ip4 10.1.252.60/16 gw4 10.1.0.1
```

2、添加DNS

```
# nmcli connection modify team0 ipv4.dns 8.8.8.8
```

3、加入组成员

```
# nmcli connection add con-name team0-eth0 type team-slave  
ifname eth0 master team0
```

```
# nmcli connection add con-name team0-eth1 type team-slave  
ifname eth1 master team0
```

4、启动

```
nmcli connection up team0
```

```
nmcli connection up team0-eth0
```

```
nmcli connection up team0-eth1
```

5、查看状态

```
# teamdctl team0 stat
```

```
[root@CentOS 7 ~]# teamdctl team0 state
setup:
  runner: activebackup
ports:
  eth0
    link watches:
      link summary: up
      instance[link_watch_0]:
        name: ethtool
        link: up
        down count: 0
  eth1
    link watches:
      link summary: up
      instance[link_watch_0]:
        name: ethtool
        link: up
        down count: 0
runner:
  active port: eth0
```

会自动创建配置文件:

team组配置文件

```
[root@localhost network-scripts]# cat ifcfg-team1
```

**DEVICE=team1**

**TEAM\_CONFIG="{\"runner\":{\"name\": \"activebackup\"}}"**

**BOOTPROTO=none**

**DEFROUTE=yes**

**IPV4\_FAILURE\_FATAL=no**

**IPV6INIT=yes**

**IPV6\_AUTOCONF=yes**

**IPV6\_DEFROUTE=yes**

**IPV6\_FAILURE\_FATAL=no**

**IPV6\_ADDR\_GEN\_MODE=stable-privacy**

**NAME=team1**

**UUID=346f1ea9-20e8-4798-b162-0cddbееef375**

**ONBOOT=yes**

**DEVICETYPE=Team** //标示这个是一个team组接口

**IPADDR=192.168.1.105**

**PREFIX=24**

**GATEWAY=192.168.1.1**

**IPV6\_PEERDNS=yes**

**IPV6\_PEERROUTES=yes**

组中端口的配置文件:

```
[root@localhost network-scripts]# cat ifcfg-team1-port1
```

**NAME=team1-port1**

UUID=01ff2a34-121a-46b0-bd4a-0f303577984f

DEVICE=ens37

ONBOOT=yes

TEAM\_MASTER=team1 //定义端口所属的组设备

DEVICETYPE=TeamPort //标示这是一个组端口接口

可以模拟设备接口down掉

nmcli device disconnect ens38

常用的team配置排障工具命令：

teamnl 和 teamdctl

显示team1接口的组端口：

```
[root@localhost ~]# teamnl team1 ports
```

```
2: ens33: up 1000Mbit FD
```

```
3: ens37: up 1000Mbit FD
```

显示team1的当前活动端口：

```
[root@localhost ~]# teamnl team1 getoption activeport
```

```
2
```

设置team1的当前活动端口：

```
[root@localhost ~]# teamnl team1 setoption activeport 3
```

```
[root@localhost network-scripts]# teamnl team1 getoption activeport
```

```
3
```