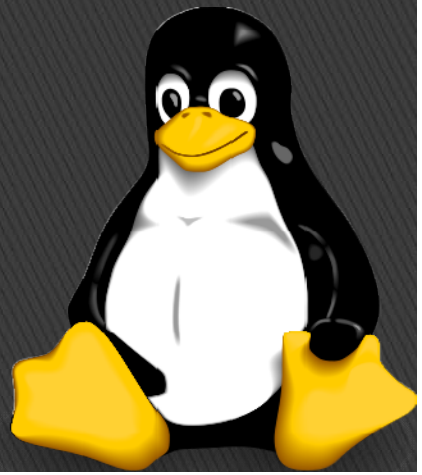
NIC Teaming (Bonding) /Link aggregation

What is NIC Teaming

Network interface bonding is called by many names: Port Trunking, Channel Bonding, Link Aggregation, NIC teaming, and others. It combines or aggregates multiple network connections into a single channel bonding interface. This allows two or more network interfaces to act as one, to increase throughput and to provide redundancy or failover.

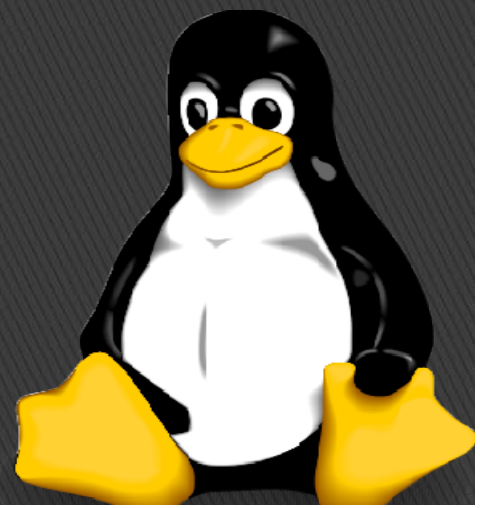
The Linux kernel comes with the bonding driver for aggregating multiple physical network interfaces into a single logical interface (for example, aggregating eth0 and eth1 into bond0).



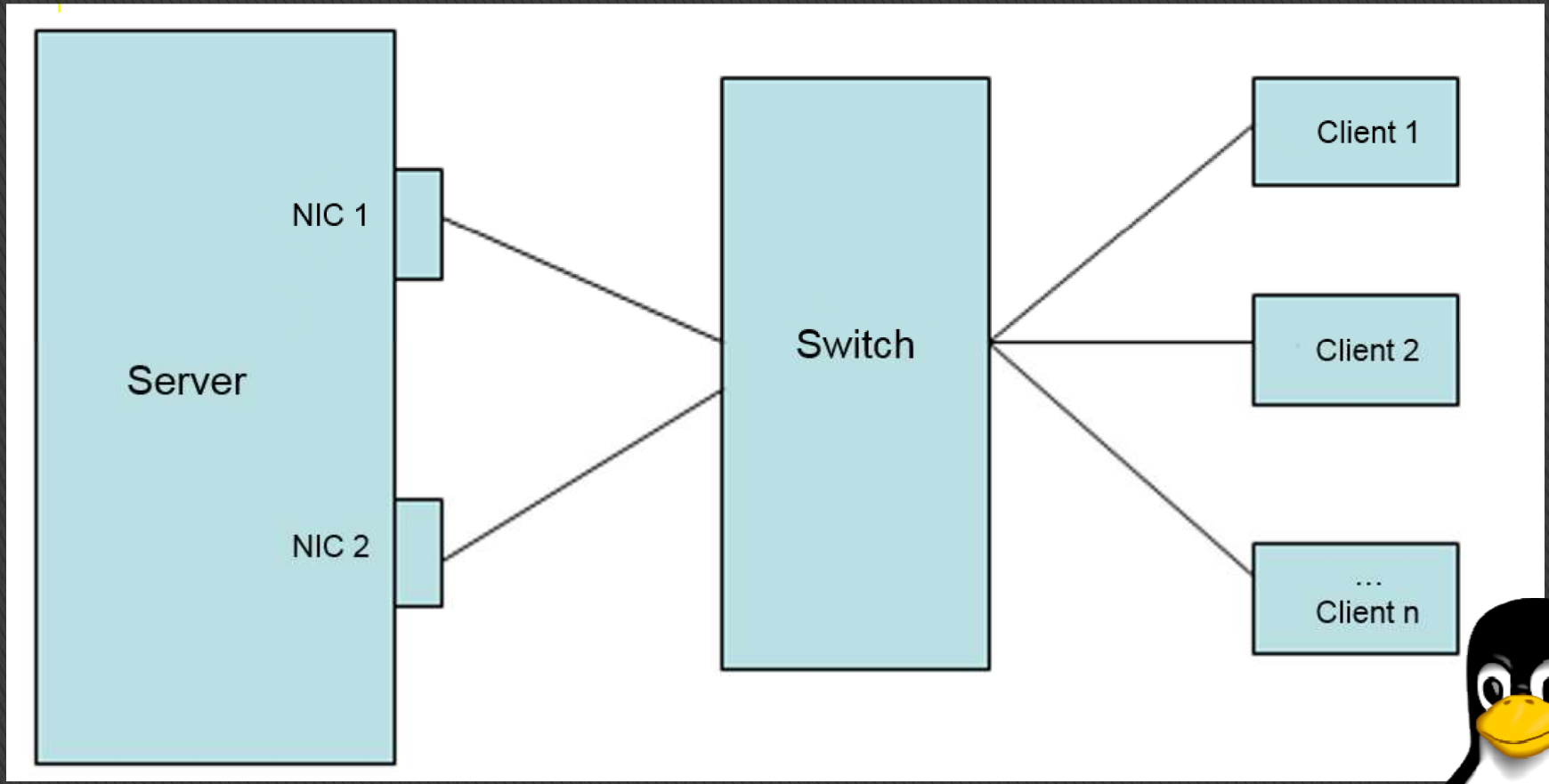
What is NIC Teaming

- NIC teaming is the concept of combining or bonding 2 or more network interfaces into one logical interface to provide high throughput and redundancy. This practice is popular especially with critical servers where high availability is expected at all times.

- In a server with 2 or more NIC cards, the concept of NIC teaming is critical in the event where one NIC card fails. With NIC teaming, the logical network interface will ensure that the remaining NIC will continue functioning and serving the purpose of the defective NIC. In this guide, we take you through the configuration of NIC teaming in CentOS 8 and RHEL 8.



NIC Teaming



Available runners (Team running modes)

Runners:

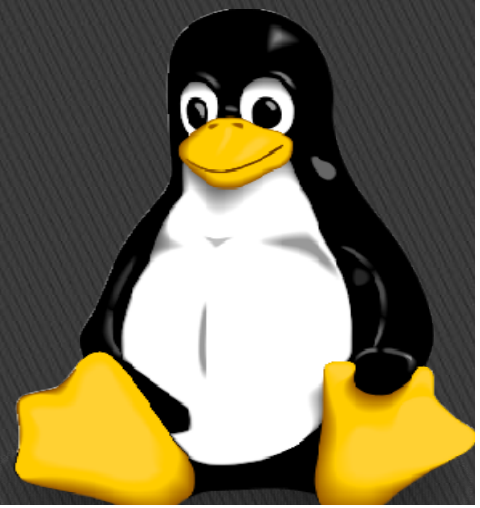
These are distinct units of implement NIC teaming in different mode.

broadcast – data is transmitted over all ports

active-backup – one port or link is used while others are kept as a backup

round-robin – data is transmitted over all ports in turn

loadbalance – Traffic is distributed across all NIC's.



Advantage of NIC Teaming

Load balancing

In the case of NIC teaming, the network traffic is balanced across all active NICs equally.

Fault tolerance

Another benefit offered by NIC teaming is higher fault tolerance. If one of the underlying physical NICs is broken down or if the cable of the corresponding NIC is unplugged, the host/server detects the fault condition and moves the traffic to another NIC automatically. This reduces the possibility of a breakdown of the entire network, thus improving the fault tolerance of the system.

Configure network teaming daemon

#yum install teamd

Create the teaming interface:

#nmcli con add type team con-name Team1 ifname Team1 config '{"runner“: {"name": "activebackup"}}'

Add an IPv4 configuration

#nmcli con modify Team1 ipv4.add 192.168.1.10/24 gw4 192.168.1.1 ipv4.dns 192.168.1.1 connection.autoconnect yes ipv4.method manual

Add the eth0 & eth1 interface to the teaming interface

#nmcli con add type team-slave con-name Team1-slave1 ifname enp0s1 master
Team1

#nmcli con add type team-slave con-name Team1-slave2 ifname enp0s2 master
Team1

Activate the teaming interface:

```
#nmcli con up Team1
#nmcli con up Team1-slave1
#nmcli con up Team1-slave2
```

For Check :

#ping 192.168.1.10
#teamdctl Team1 stat