To LACP or NOT to LACP?

We have gotten quite a few customer inquiries lately about using LACP with Nimble Storage – and I was fortunate enough to secure an HP Procurve network switch to dig deeper. If you have LACP in mind for your virtualized environment, this post could be useful to you (from both VM traffic and iSCSI storage traffic perspectives)

First of all, let's discuss what are the use cases/benefits for LACP for VM traffic:

What is link aggregation, LACP?

In a nutshell, it allows one to aggregate multiple network connections in parallel to increase throughput beyond what a single connection could sustain, and to provide redundancy in case one link goes down. LACP is a vendor independent standard term which stands for Link Aggregation Control Protocol, defined in IEEE 802.1ax or 802.3ad. LACP links need to be manually configured on the physical network switch, to allow both links to appear as one logical aggregated link. MAC address(es) from the host side could appear on both links simultaneously, and the switch will not freak out and thinking there's a loop on the network.

What are some use cases for LACP?

Here are some that we have seen from customer environments:

1)use LACP to ensure ESX host NICs are connected correctly (each NIC going to separate network switches): in this scenario, a LACP trunk is typically configured between two physical switches, in dynamic mode. The trunk would not come to live until both NICs are connected to the correct ports on both switches. This is a good practice to ensure the NICs are connected to the right port for ease of operational management and traceability, and more importantly, no user error of connecting both NICs to the same switch causing a single point of failure at the switch layer

2)for VMs that host applications needing access to multiple target IP addresses, LACP links combined with IP hash load balance algorithm provide good balance of traffic across all connections. Compared to traditional NIC teaming, all links get utilized simultaneously. While traditional NIC teaming is simple to configure, without any extra steps needed on the physical switch, a given VM could only be active on one link at a time (as the MAC appearing on two ports on the switch that are not LACP configured would cause one of the ports to be shutdown)

"Gotchas" with LACP

Compared to traditional NIC teaming, LACP does require a few extra steps:

1. ports need to be configured to be part of a LACP trunk (by default, switches do not have this enabled)
2. dynamic LACP typically uses the default VLAN on the network switch; general practice is to NOT use default VLAN on the switch. In such a case, you need to manually tag VLAN(s) for the LACP trunk
3. from ESX side, you need to explicitly use "out-ip" load balance algorithm for your vSwitch (you could also define this at the port group level); if you use vSphere 5.1, the ONLY supported LACP configuration is through the use of vDS (vNetwork Distributed Switch); refer to the following VMW KB for further details

LACP for iSCSI Traffic?? NOT!

Now, moving away from VM traffic to iSCSI storage traffic – the recommended practice is to use MPIO (multipathing I/O). For block storage, there is no reason to configure LACP for ports facing storage target side. Path availability and load distribution are handled by ESX storage MPIO stack (PSA SATP & PSP), refer to my joint webcast with VMware's Mostafa Khalil for further details. In a nutshell, SATP (storage array type plugin) would monitor the health status of all active paths, and perform switch over if necessary. PSP(path selection plugin) chooses the best path to issue I/O – for Nimble Storage specifically, we recommend setting iops=0 & bytes=0 to achieve the best load distribution possible. VMkernel does not have to wait for an IO to be issued or certain number of bytes to be issued prior to using all available paths. Don't believe me, check out the result from one of our customers in the Nimble community. If you are not using our storage, check with your vendor on their specific recommendations.

Now here's an interesting scenario that customers have brought to our attention – there are two, and ONLY two 10G NICs available on the ESX server, therefore, VM traffic and storage traffic need to co-exist between the two connections. In this case, LACP is the preferred method for VM traffic for customer, but they also want to leverage the same set of uplinks for iSCSI, and use VMkernel MPIO for load distribution – can this be done? Answer is – hellz yeah! If this scenario applies to you, here are the high level steps/considerations to keep in mind:

1. Use ESX 5.1 vDS as that's the only supported method for LACP
2. Use NetIOC (Network IO Control) feature in vSphere to configure quality of service for both VM traffic and storage traffic: you certainly want to put a cap on regular VM traffic to prevent bursty network access traffic to impact storage traffic, causing extra latency to your applications
3. Configure LACP for the dvUplinks (dynamic or static)
4. Use "out-ip" load balance algorithm for EVERY portgroup (including the one for iSCSI)
5. For the vmk NICs for iSCSI, override the failover order for the vmnics
6. Configure LACP properly on the network switch, with VLAN tagging

Some details for each of the steps above:

Let's start from switch side as that's where you typically would start configuring this stuff (HP Procurve switch was used in our lab):

#config (go into configuration terminal)

#trunk 24-25 trk2 (create trunk group called 'trk2')

#int 24-25 lacp active (enable lacp for both ports)

#vlan 528 tagged trk2 (enable tagging for the VLAN that we want to use, and tell the LACP trunk NOT to use default VLAN)

After the above steps are completed, confirm the new lacp group is up:

#show lacp

You should see something like the following:

```
C28-SW3(config)# show lacp

                          LACP

  PORT   LACP      TRUNK     PORT      LACP      LACP
  NUMB   ENABLED   GROUP     STATUS    PARTNER   STATUS
  ----   -------   -------   -------   -------   -------
  1      Active    Trk1      Up        Yes       Success
  2      Active    Trk1      Up        Yes       Success
  24     Active    Trk2      Up        Yes       Success
  25     Active    Trk2      Up        Yes       Success
```

#show vlan 528

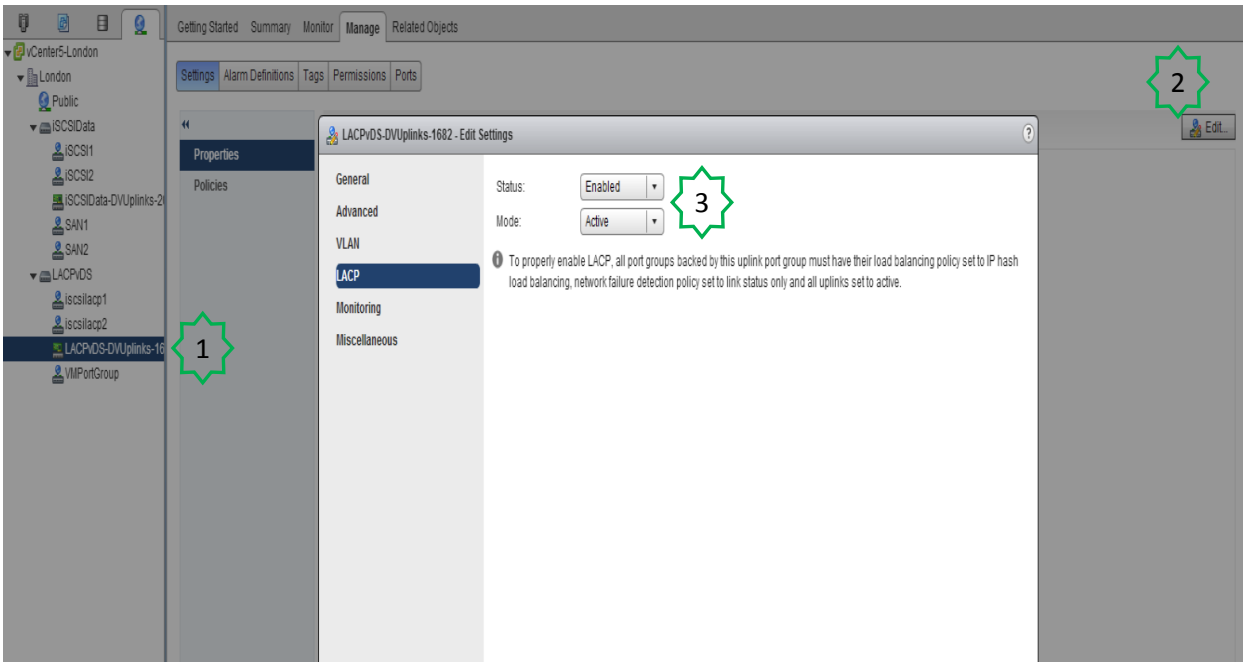
You should see 'tagged' mode for the VLAN that you want your LACP group to be on, with the trunk group listed as a member:

```
 Trk1             Tagged   Learn        Up
 Trk2             Tagged   Learn        Up
```

Your switch ports are now ready for ESXi side – here are the steps to take from ESXi5.1 with vDS (**you have to use the web client for this to be configured properly**):

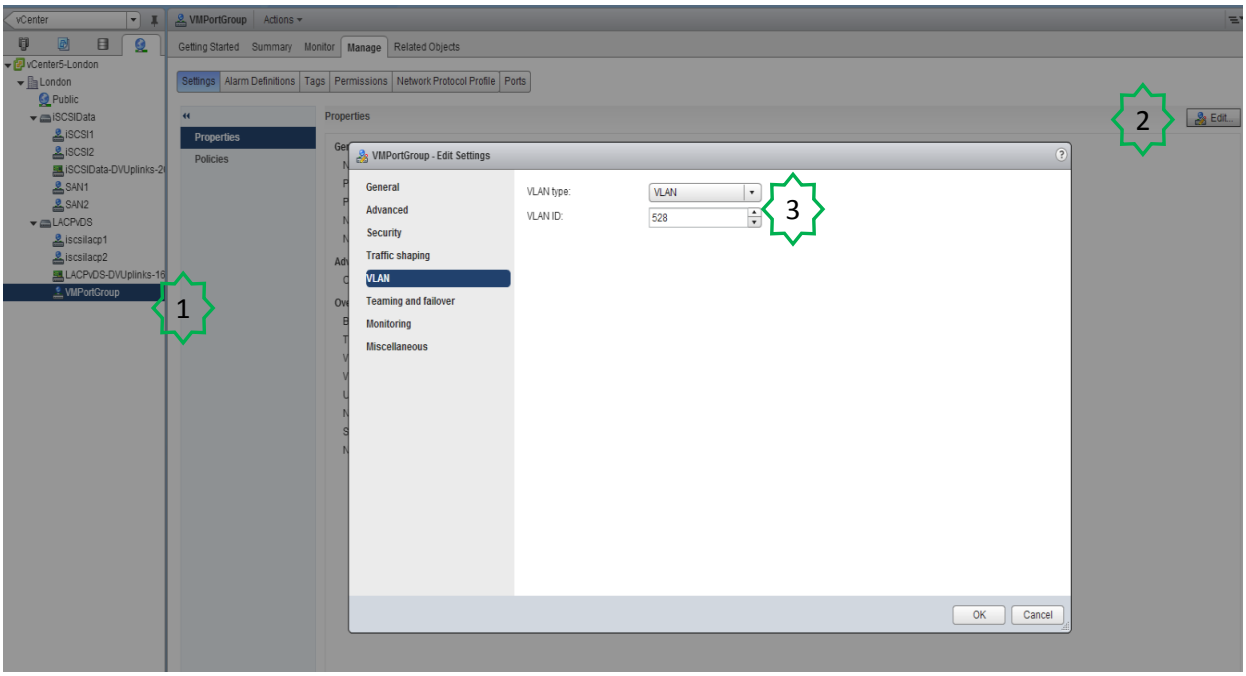
-create a vDS and add your host(s) of interest + select the uplinks

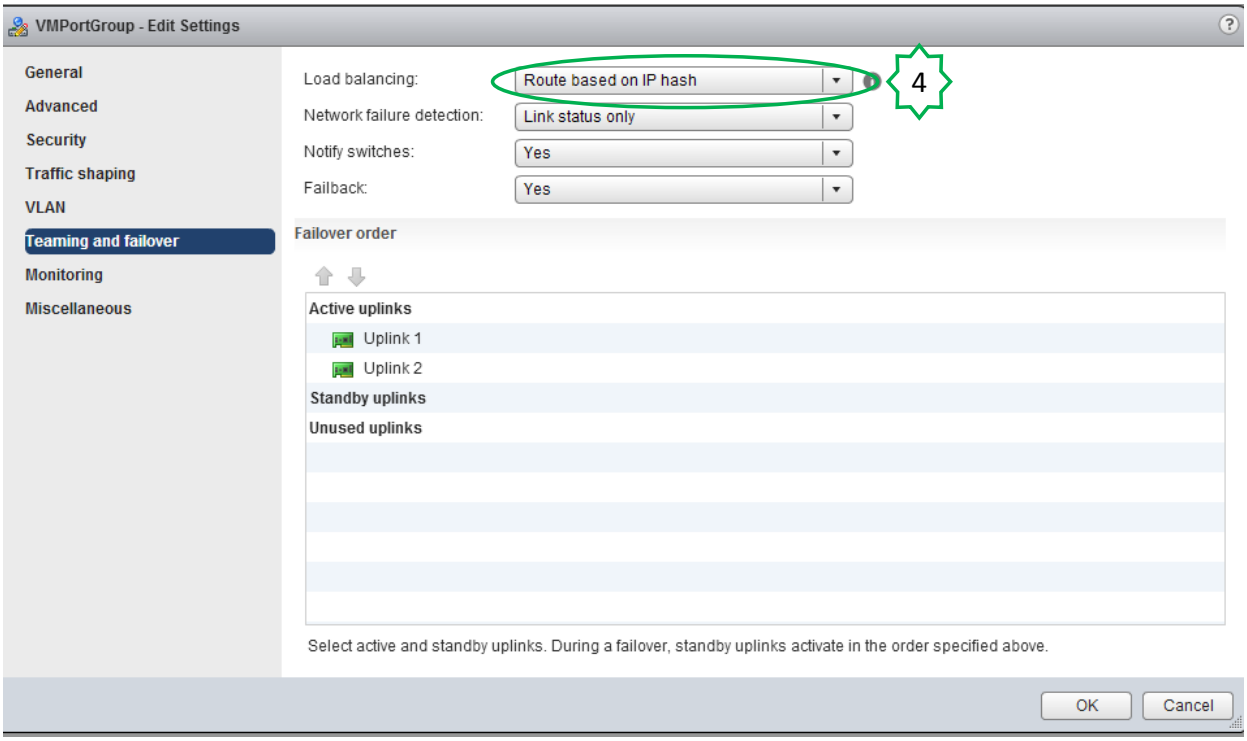
-once your vDS has been created with uplink group, go into the vDS uplinks group that has been created, and following the steps below to enable LACP:



-now create a VM dvPortgroup for your VM traffic

-after that has been done, configure VLAN 528 for the portgroup, AND configure load balancing to use “Route based on IP hash”

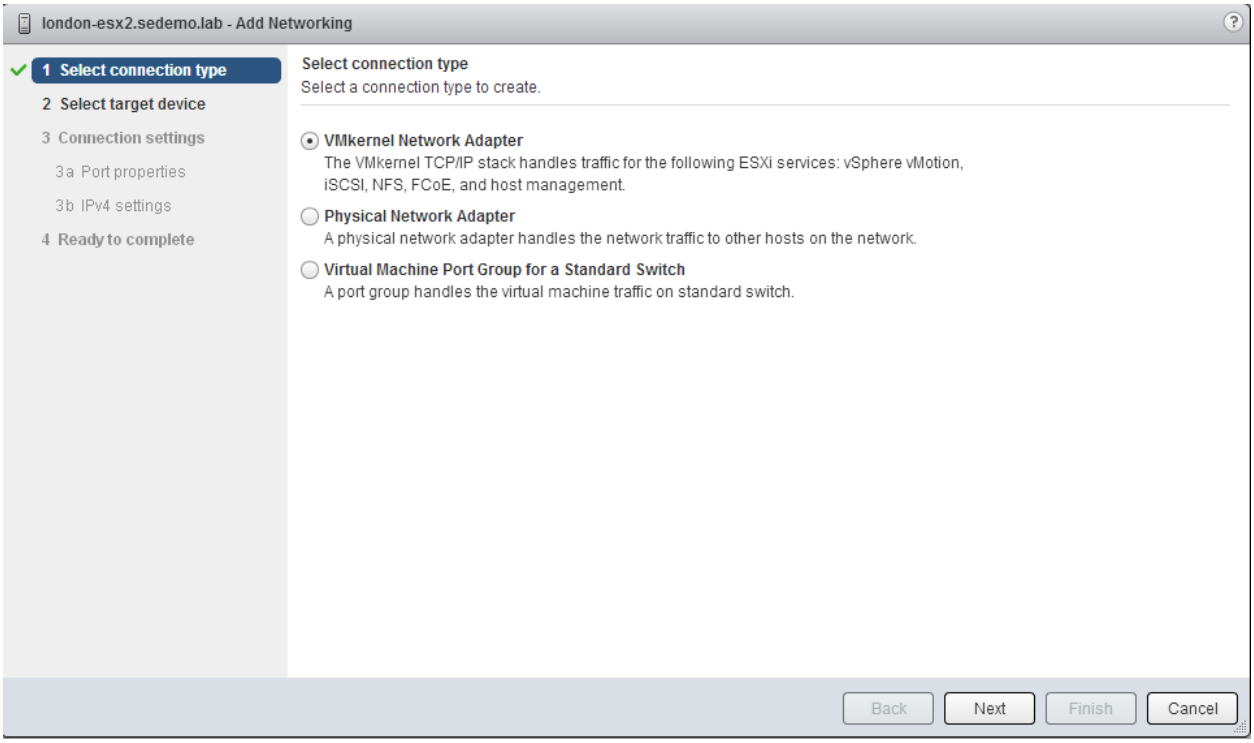




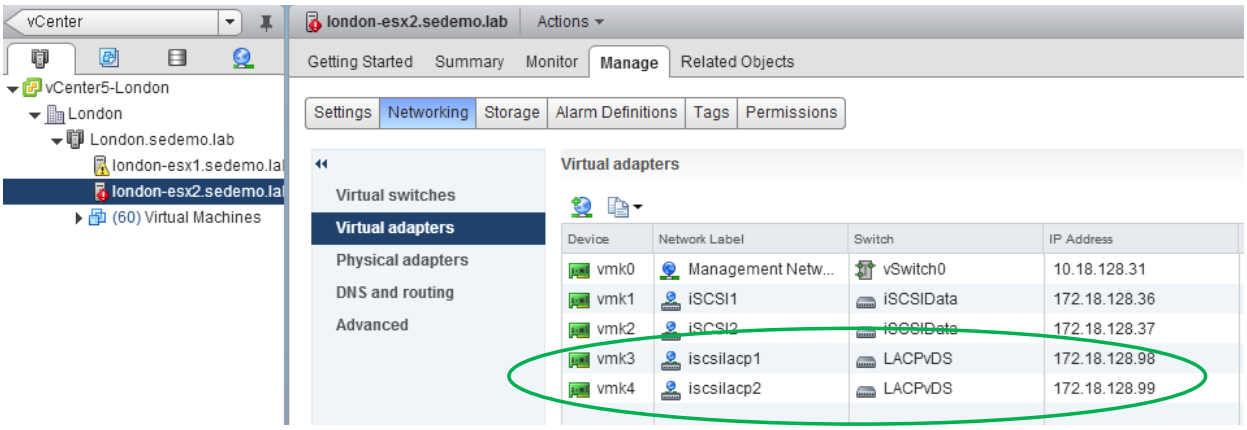
Now, here's the important part if you want your iSCSI traffic to share the same uplinks on the vDS:

-create two dvPortgroups for VMkernel

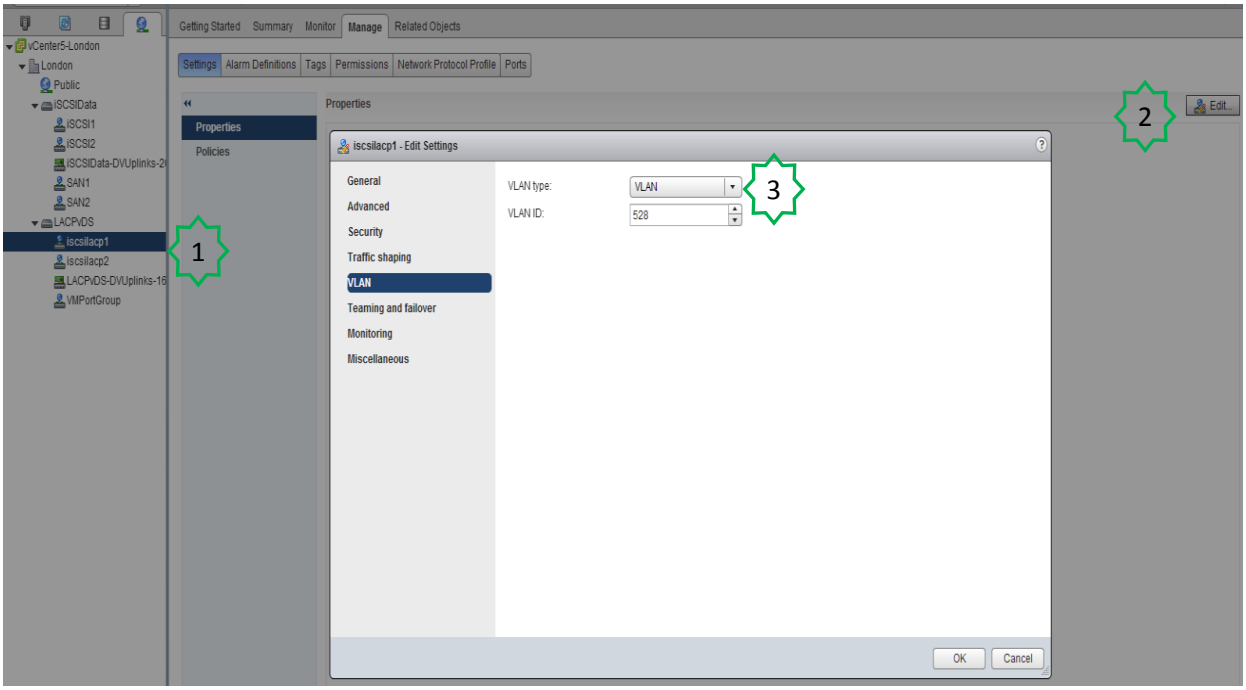
-create two virtual adapters respective (VMK1 & VMK2) and associate each with its respective dvPortgroups (you have to do this from each ESX host – there is no option to create this from the networking pane of the web client)



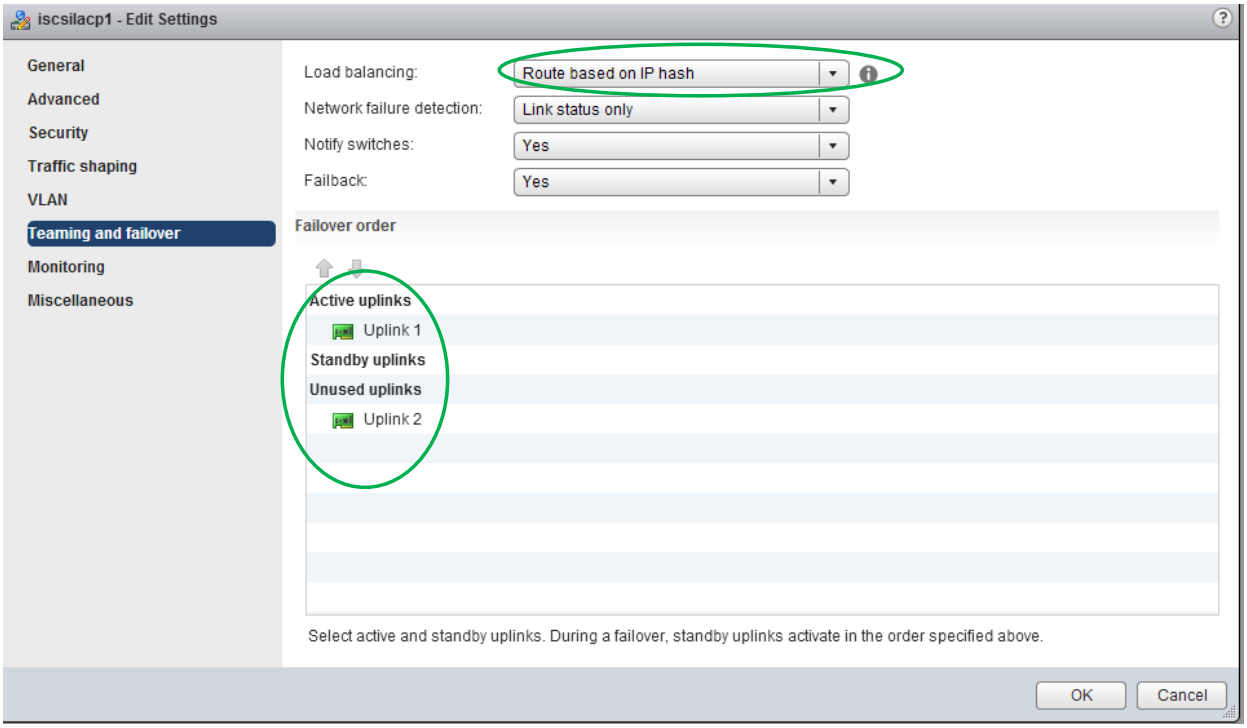
The finished product should look something like this:



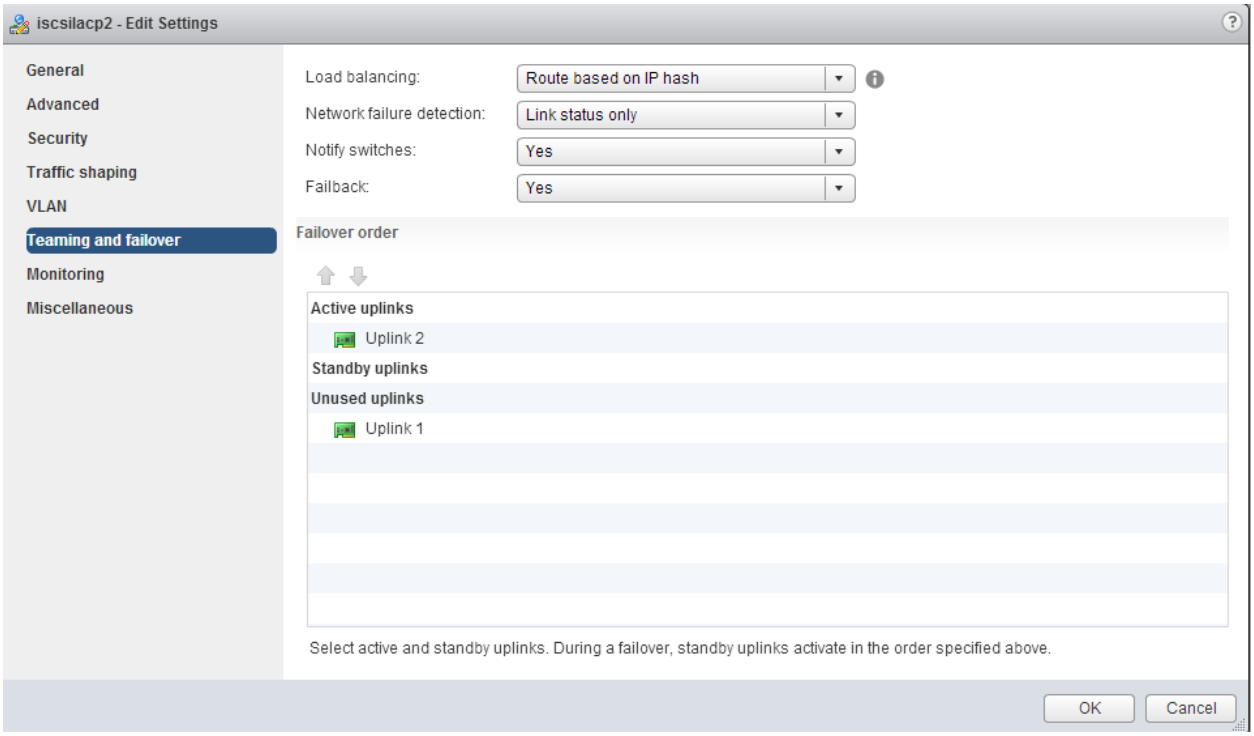
Update the VLAN ID for each of the dvPortgroup for iSCSI:



Now update the load balance algorithm to "Route based on IP hash" AND override the "Failover order" for the uplink adapters! This step is crucial to iSCSI configuration. If the step below is not done, the sw/iscsi initiator in ESXi would NOT be able to bind to this vmkernel port!



Do the same for the second dvPortgroup for iSCSI, and make sure the active uplink is the second vmnic, and the unused uplink is the first:



After the above steps are completed, you can now bind your sw/iscsi initiator to both vmk virtual NICs that have been created.

That's it, you have now successfully configured your ESXi environment to share a two 10G connection, configured with LACP for VM traffic, and iSCSI traffic with ESXi MPIO. I'd like to call it best of both worlds without spending additional $$ for dedicated 10G cards.