Proxmox VE Networking

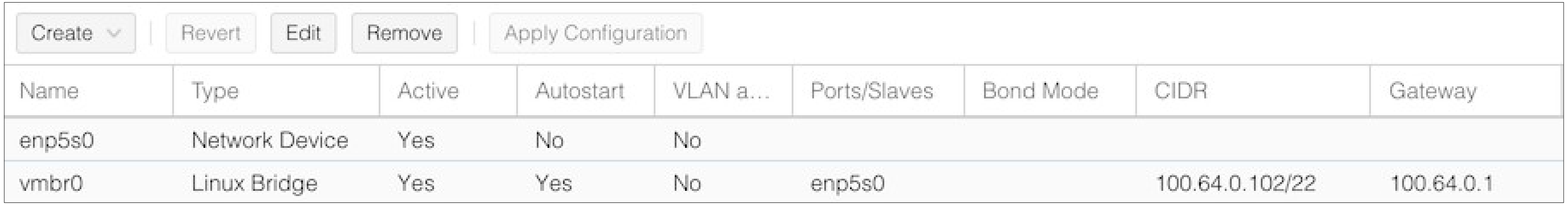
Cloud and Virtualization Workshop

UNIVERSITY OF OREGON

NSRC
Network Startup Resource Center

Default Networking Configuration

- This is the initial config from the ISO installer
- A bridge named "vmbr0"
  - Configured with your node's chosen static IP address and gateway
  - With one physical port as bridge member
- GUI view:



Name	Type	Active	Autostart	VLAN a...	Ports/Slaves	Bond Mode	CIDR	Gateway
enp5s0	Network Device	Yes	No	No				
vmbr0	Linux Bridge	Yes	Yes	No	enp5s0		100.64.0.102/22	100.64.0.1

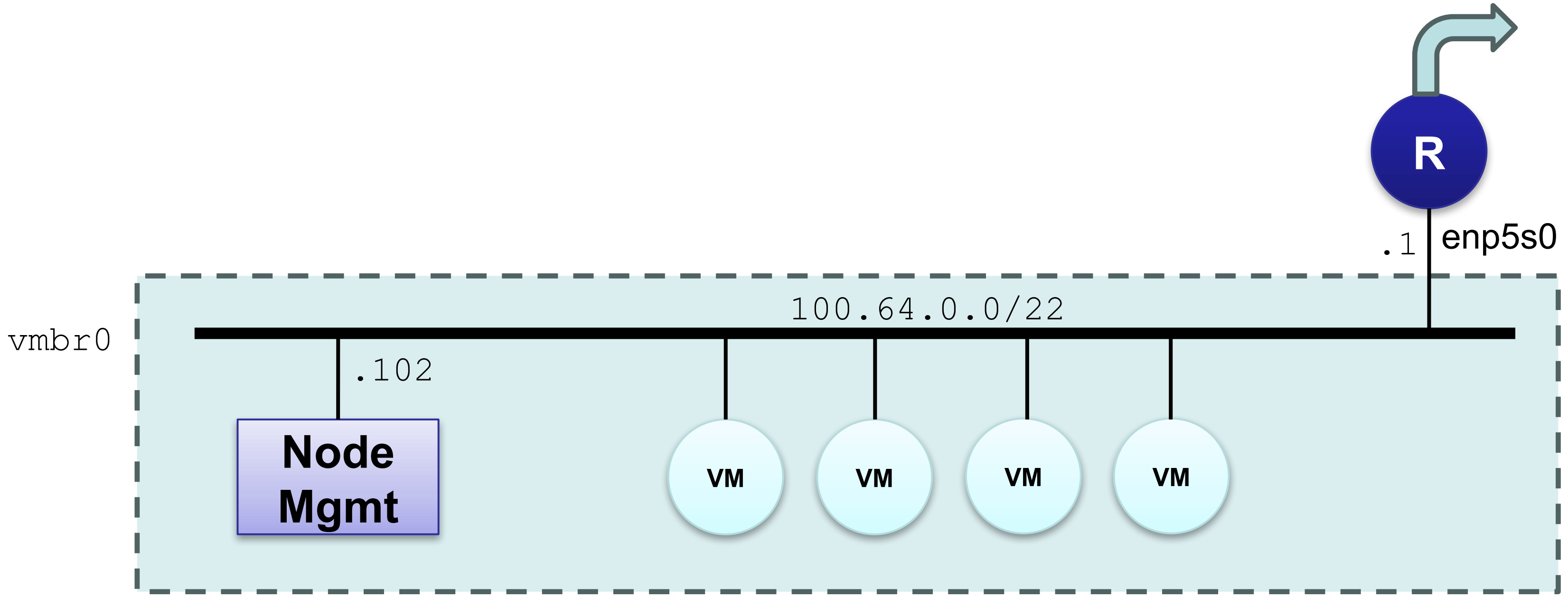
/etc/network/interfaces

```
auto lo
iface lo inet loopback

iface enp5s0 inet manual

auto vmbr0
iface vmbr0 inet static
        address 100.64.0.102/22
        gateway 100.64.0.1
        bridge-ports enp5s0
        bridge-stp off
        bridge-fd 0
```

"ifupdown2" config

- Changes made in the GUI edit this underlying file
- You'll be asked to confirm the changes (diff)

Flat network



VMs have addresses on the same subnet as Proxmox node (either assigned statically, or from an upstream DHCP server)

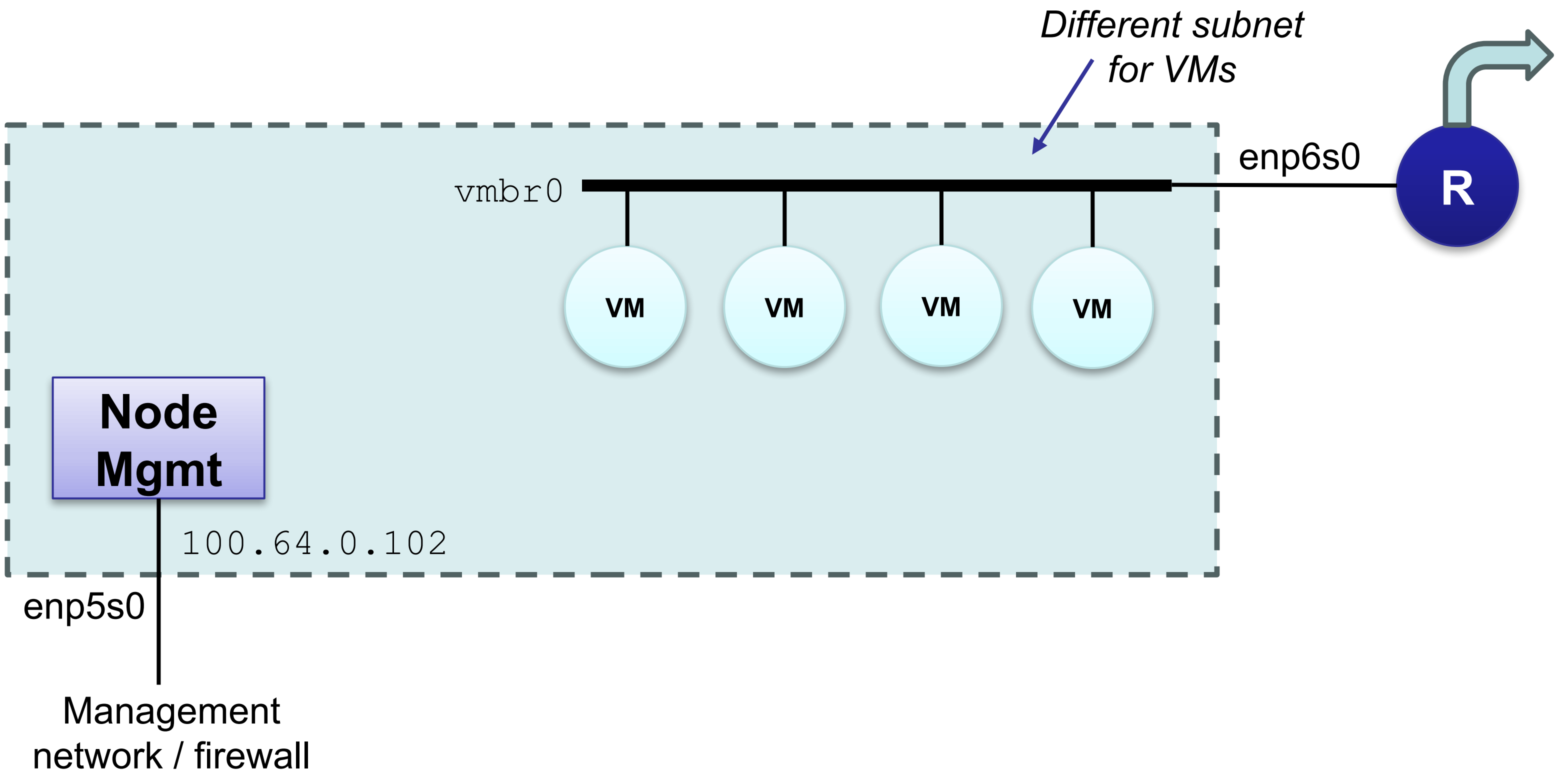
This works, but…

- If VMs are exposed to the outside world, then so is the Proxmox management interface and API
  - Probably not a good idea
  - How strong are your admin passwords? Is everyone using 2FA?
  - How much do you trust that there are no security flaws in Proxmox?
  - However, you can restrict access using firewall rules (see later)
- VMs can potentially attack each other, and Proxmox itself
  - e.g. ARP spoofing: one VM takes over the IP address of another

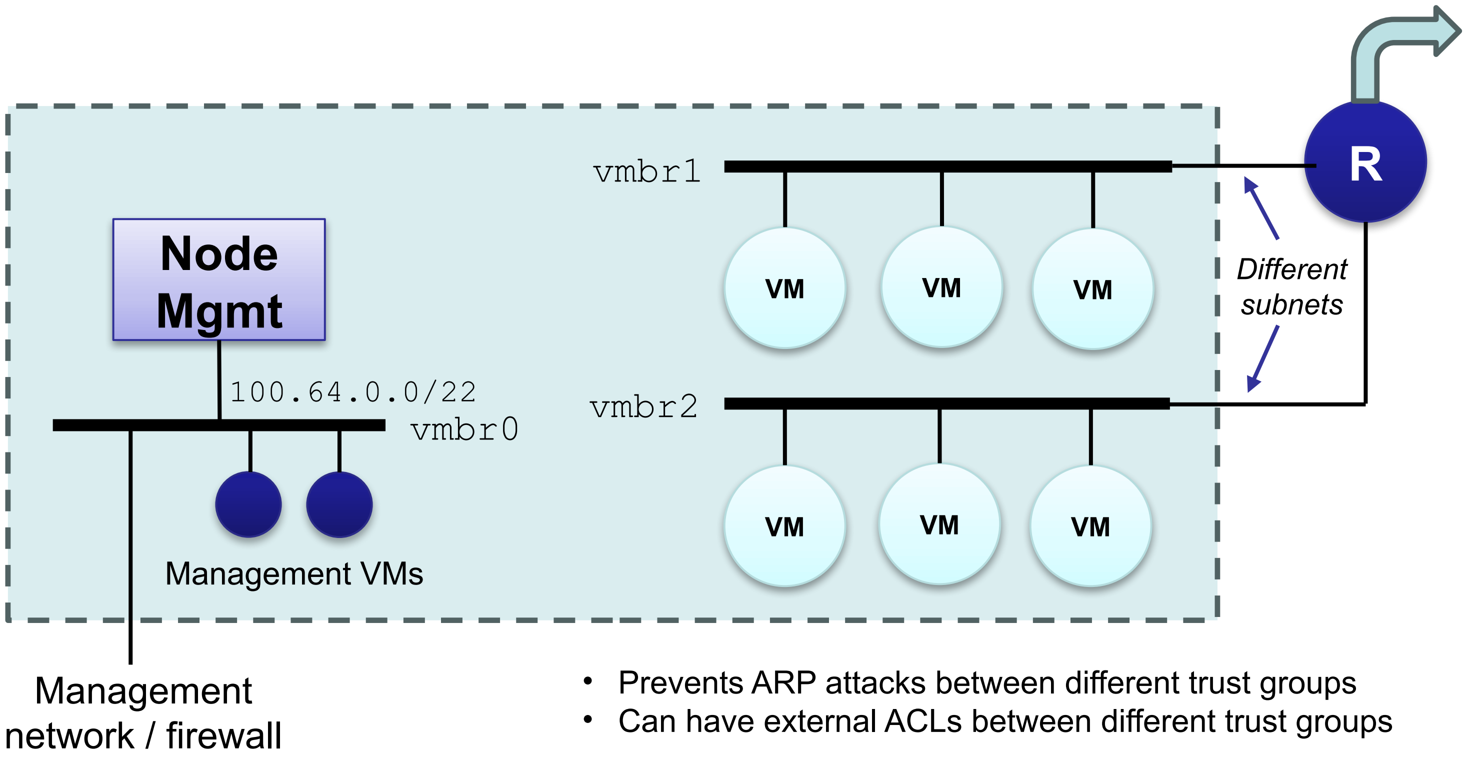
Suggest: separate management network

- Move the management address to a different network
- Can be directly assigned to management NIC, or to another bridge which connects to management NIC
- On the bridge used by VMs, *do not assign any IP address*
  - That is: the bridge carries traffic to/from the VMs, but the node itself has no address on this subnet, so cannot be attacked

Separate VM subnet



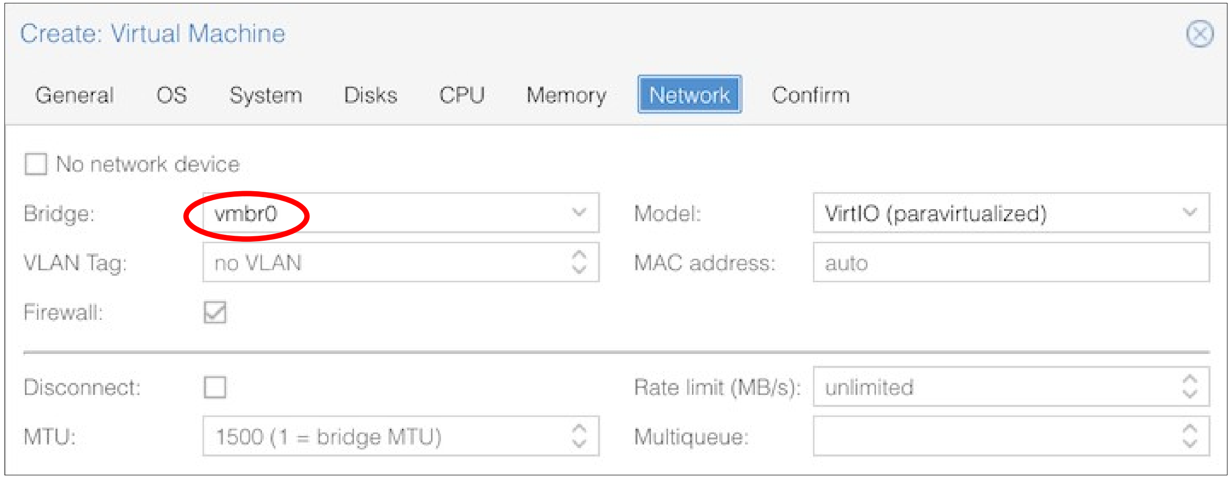
Multiple VM networks



- Prevents ARP attacks between different trust groups
- Can have external ACLs between different trust groups

When creating a VM…

- You select which bridge its virtual NIC is attached to
- Hence assign the VM an IP address from the correct subnet


Create: Virtual Machine

☐ No network device

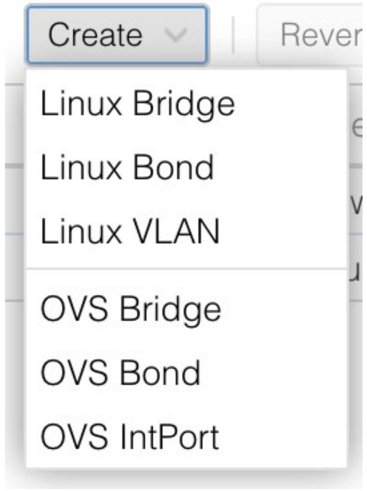
Bridge:	vmbr0	Model:	VirtIO (paravirtualized)
VLAN Tag:	no VLAN	MAC address:	auto
Firewall:	☑		
Disconnect:	☐	Rate limit (MB/s):	unlimited
MTU:	1500 (1 = bridge MTU)	Multiqueue:	

Bridge naming

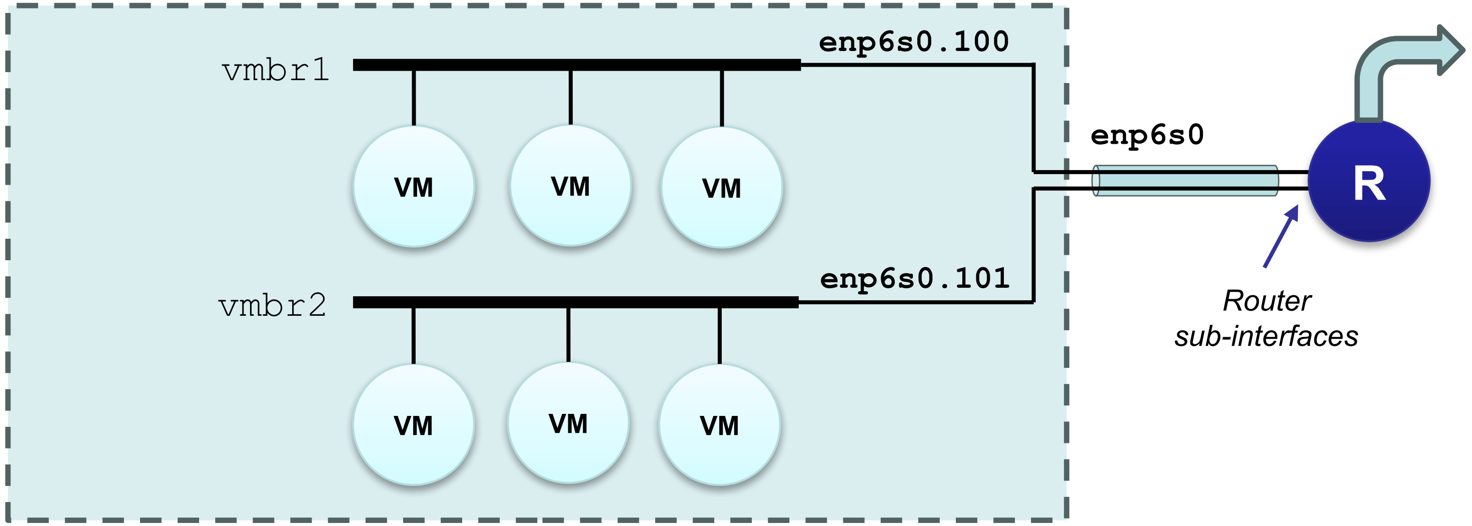
- You don't need to call the bridges "vmbr0", "vmbr1" etc
- You can give them meaningful names, e.g.
  - the customer name (if separate subnet per customer)
  - the department name, the application name, …
- Makes it easier to select the correct bridge

VLAN tagging (trunks)

- Rather than a separate physical NIC for each subnet, you can use VLAN tags on a trunk link
- Create a "Linux VLAN" interface
  - You can name it <interface>.<vlan> e.g. enp6s0.100
  - OR use arbitrary name, in which case you have to specify the parent interface and vlan tag
- Create a "Linux Bridge"
  - with "bridge ports" containing the VLAN interface
- Repeat for each subnet



Bridges and VLAN trunk

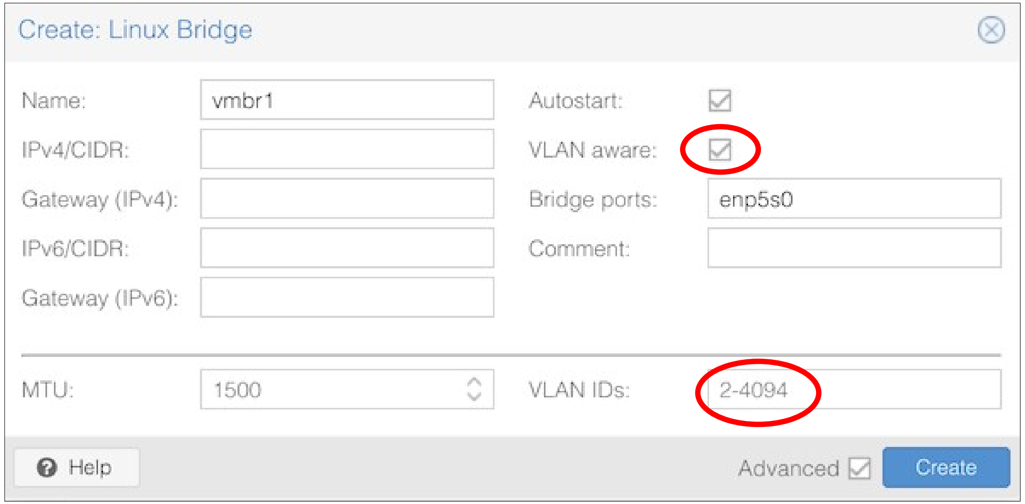


Limitations of this approach

- If you need a large number of subnets, you have to create a large number of VLAN interfaces and a large number of bridges
- Becomes difficult to manage (since you have to repeat this configuration on each node)

Alternative: "VLAN-aware bridge"

- Create a single bridge
- Select "VLAN-aware" and a range of allowed VLANs



Create: Linux Bridge

Name:	vmbr1	Autostart:	✓
IPv4/CIDR:		VLAN aware:	✓
Gateway (IPv4):		Bridge ports:	enp5s0
IPv6/CIDR:		Comment:	
Gateway (IPv6):			
MTU:	1500	VLAN IDs:	2-4094

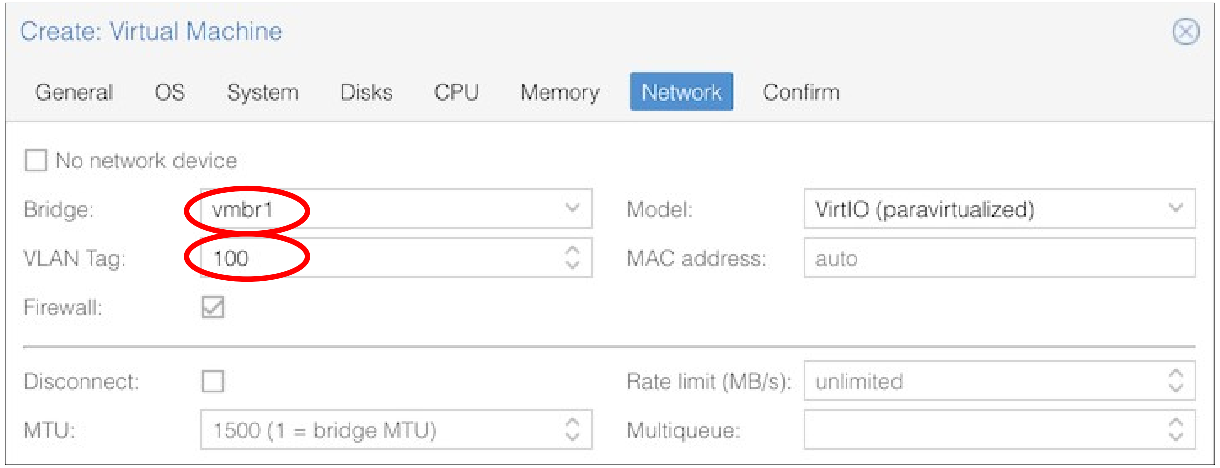
```
bridge-vlan-aware yes
bridge-vids 2-4094
```

UNIVERSITY OF OREGON

NSRC Network Startup Resource Center

Using VLAN-aware bridge on a VM

- Select the bridge and the required VLAN number
- You lose the ability to have meaningful names for each subnet


Create: Virtual Machine

☐ No network device

Bridge:	vmbr1	Model:	VirtIO (paravirtualized)
VLAN Tag:	100	MAC address:	auto
Firewall:	☑		
Disconnect:	☐	Rate limit (MB/s):	unlimited
MTU:	1500 (1 = bridge MTU)	Multiqueue:	

Alternative: "Open vSwitch"

- Separate software implementation of bridging
- Similar to kernel VLAN-aware bridging, but with more features
  - e.g. VXLAN: lets you extend a subnet across multiple locations using tunnelling
- Powerful but complex
- If you want this, read the documentation and test it out
- https://pve.proxmox.com/wiki/Open_vSwitch

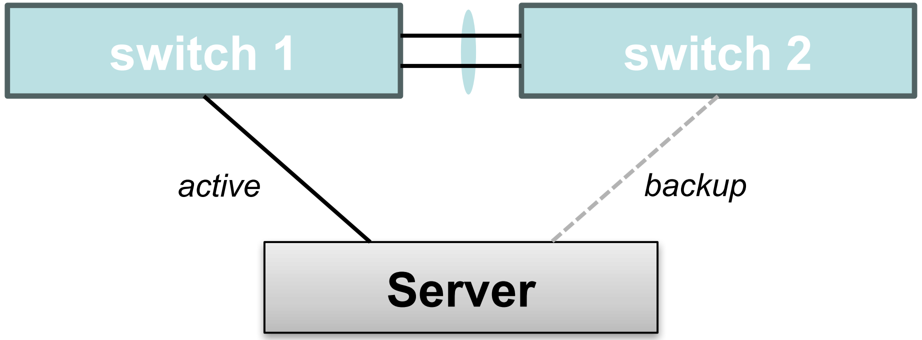
How many NICs should you have?

- You don't want very busy VM or storage traffic to starve out important management or intra-cluster traffic
- Suggestion:
  - 10G for VM external subnets (trunk)
  - 10G for storage replication (e.g. Ceph OSD, Linstor)
  - 1G for management (Proxmox API, Ceph API, Linstor API)
  - 1G for cluster (corosync)
  - Server may also have 1G out-of-band management port: use it!

Network redundancy

- You can use *bonding* to combine NICs
- For failover and/or additional capacity

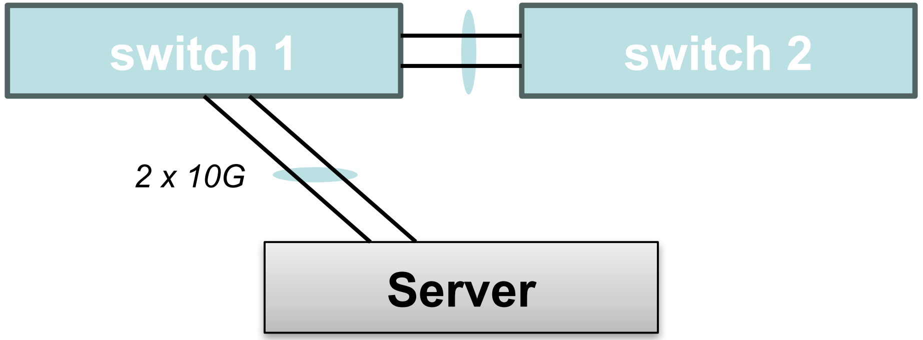
Active/backup failover



Handles some failures of port, or entire switch

But depends on L1 link-down to detect failure

LACP Link Aggregation Group (LAG)

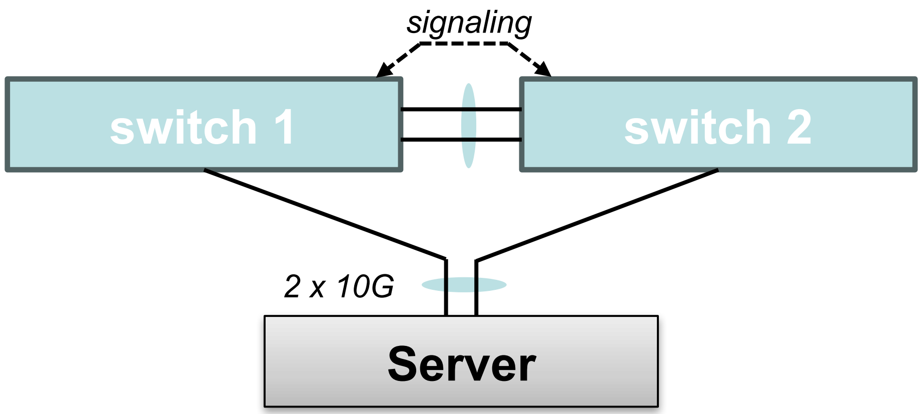


Load-shares between links – extra capacity

Active testing – detects more failure modes

But requires both links into same switch

Multi-chassis LAG (MLAG)



- Gives load-sharing *and* active failover
- From server's point of view, it's standard LACP
- But switches use vendor-proprietary signaling

Proxmox Firewalling

UNIVERSITY OF OREGON

NSRC
Network Startup Resource Center

Firewall config

- Firewall must be enabled at multiple levels
  - Globally - Datacenter (default: no)
  - On each node (default: yes)
  - On each instance (default: yes)
  - On each network interface (default: no)
- Is able to restrict traffic to/from each VM separately
  - Using ebtables, works at both layer 2 and layer 3
  - Can restrict traffic between VMs on same subnet
  - Similar to EC2 security groups

Enabling the firewall globally

- Firewalling restricts access to Proxmox admin interface
  - Create an IPSet called "management" (exact name) containing the subnet(s) you want to allow access from
  - Otherwise, the default is only to allow from the local subnet
- You can lock yourself out! If so, login at console and run:

```
pve-firewall stop
```

- The firewall is still "enabled", but it's "stopped"
- When you've fixed the firewall rules:

```
pve-firewall start
```

Enabling the firewall globally

- Default ruleset allows corosync cluster traffic, but not Linstor, or Ceph OSD via separate storage network – you'll need to add rules for those yourself
- e.g. create an ipset containing all your node IPs, and allow all traffic with this ipset as source and destination
- or a single rule with a subnet covering your management IPs
  - if you're being really careful you'd limit this to just required ports

Firewall rules

- Cluster-level rules
  - Affect all traffic, including node management
- Instance-level rules
  - Affect traffic to/from this VM only
- Stored in files under /etc/pve/firewall
  - Easy to read, audit, backup, etc.
- An "IPSet" is a list of IP addresses and/or subnets that can be used in a rule
  - simplifies rules and makes them easier to maintain