univention
be open.

Univention Corporate Server

Extended computer management documentation

univention
be open.

Table of Contents

Chapter 1. Advanced networking configuration

UCS supports advanced network configurations using bridging, bonding and virtual networks (VLAN).

- Bridging is often used with virtualization to connect multiple virtual machines running on a host through one shared physical network interface.
- Bondings allows failover redundancy for hosts with multiple physical network interfaces to the same network.
- VLANs can be used to separate network traffic logically while using only one (or more) physical network interface.

Configuration of bridges, bondings and VLANs is performed through Univention Configuration Registry variables. The following sections provide an example for each configuration.

Caution

Changing the network configuration on a running system is dangerous, since any misconfiguration can make the host unreachable. Make sure you can access the host out-of-band in case of errors! It is important to set all related Univention Configuration Registry variables at once, since the interfaces are re-configured after each change unles the Univention Configuration Registry variable `interfaces/restart/auto` is set to `false`.

The following examples only configure IPv4, but all examples can be used with IPv6 as well.

1.1. Bridging

Feedback

Bridging allows a physical network interface to be shared by multiple virtual machines running on a single host. Instead of using one physical interface for each virtual machine and the host itself, all hosts are connected through only one (or more) uplink(s), which is then split up virtually just like a physical hub or switch would be used to connect multiple downstream hosts to one uplink.

Any physical network interface must not have an IP addresses configured and must not be started automatically. This can be achieved by setting the Univention Configuration Registry variables `interfaces/$INTERFACENAME/type` to `manual` and `interfaces/$INTERFACENAME/start` to `false`.

An IP address can be assigned to the bridge device (`br0`) instead, which can then be used as the primary interface for services on the host itself.

The Spanning Tree Protocol (STP) prevents bridge loops, i.e. multiple network paths from a source to a destination host. If bridging is used for virtual machines and the host is an edge node (in other words the bridge does not really connect two physical network interfaces and is not supposed to forward traffic between those networks), STP should be disabled and the so-called *forwarding delay* (`bridge_fd`) should be set to 0. This is needed for virtual machines to boot from the network via PXE, since otherwise the initial DHCP packets will be dropped and won't reach the network.

1.1.1. Software requirements

Feedback

The Linux kernel only implements basic STP but neither the extensions Rapid STP (RSTP) nor Multiple STP (MSTP).

To configure bridges the package ***bridge-utils*** needs to installed and the kernel module `bridge` must be loaded. To ensure automatic loading of the module `;bridge` can be appended to the Univention Configuration Registry variable `kernel/modules`.

1.1.2. Example configuration

Feedback

The following example uses a single physical network interface, which is sufficient for connecting virtual machines. For building a real bridge to forward packets between two physical interfaces, all these interfaces would be mentioned in the *`bride_ports`* stanza.

```
grep -v '^#' <<END_CONFIG | xargs -d '\n' ucr set
# Configure physical interface
interfaces/eth0/type=manual
interfaces/eth0/start=false
# Configure IP on bridge
interfaces/br0/start=true
interfaces/br0/address=192.168.122.13
interfaces/br0/broadcast=192.168.122.255
interfaces/br0/netmask=255.255.255.0
interfaces/br0/network=192.168.122.0
interfaces/primary=br0
# Connect physical interface to bridge
interfaces/br0/options/1=bridge_ports eth0
# Set forwarding delay to 0
interfaces/br0/options/2=bridge_fd 0
END_CONFIG
```

For more informations on configuring bridging see the bridge-utils-interfaces(5) manual page.

1.2. Bonding

Feedback

Bonding allows two (or more) physical network interfaces to be coupled to increase network throughput or to improve redundancy in failover scenarios. While various modes are supported by the bonding driver, the following example only describes an active-backup failover scenario.

Any physical network interface must not have an IP addresses configured and must not be started automatically. This can be achieved by setting the Univention Configuration Registry variables `interfaces/`*`$INTERFACENAME`*`/type` to `manual` and `interfaces/`*`$INTERFACENAME`*`/start` to `false`.

The IP address is assigned to the bonding device (*`bond0`*) and is used as the primary interface for the services on the host itself.

1.2.1. Hardware requirements

Feedback

Active-backup setups can be configured with any network switch, while active-active setups and channel aggregation must be supported by the switch. Using the MII to detect active links must be supported by the switch, too.

1.2.2. Software requirements

Feedback

To configure bonding the package ***ifenslave-2.6*** needs to be installed and the kernel module `bonding` must be loaded. To ensure automatic loading of the module `;bonding` can be appended to the Univention Configuration Registry variable `kernel/modules`.

1.2.3. Example configuration

Feedback

In the following example the bonding device is configured to check the link state of the physical interfaces every 100 ms using the MII monitor of the network interface. In case of a link failure the host switches the active link over to the other interface and remains there until that link fails too.

```
grep -v '^#' <<END_CONFIG | xargs -d '\n' ucr set
# Configure physical interfaces
interfaces/eth0/type=manual
interfaces/eth0/start=false
interfaces/eth1/type=manual
interfaces/eth1/start=false
# Configure IP on bond
interfaces/bond0/start=true
interfaces/bond0/address=192.168.122.13
interfaces/bond0/broadcast=192.168.122.255
interfaces/bond0/netmask=255.255.255.0
interfaces/bond0/network=192.168.122.0
interfaces/primary=bond0
# Connect physical interfaces to bond
interfaces/bond0/options/1=bond-slaves eth0 eth1
# Use active-backup mode
interfaces/bond0/options/2=bond-mode 1
# Check MII link status very 100 ms
interfaces/bond0/options/3=bond-miimon 100
# Do not prefer any interface over the other
interfaces/bond0/options/4=bond-primary eth0 eth1
END_CONFIG
```

For more informations on bonding see the Linux Ethernet Bonding Driver HOWTO [https://www.kernel.org/doc/Documentation/networking/bonding.txt] documentation.

1.3. Virtual LANs

Feedback

Virtual LANs (VLANs) can be used to separate network traffic into different virtual networks each representing their own broadcast domain. Instead of using multiple physically separated networks, the network packets are instead tagged with a VLAN ID, which is used to partition the traffic. Valid VLANs are numbered from 1 to 4095. The same VLAN ID must be configured on all switches.

A link between two network devices can either transport packets of only a single VLAN (which are normally untagged) or can transport packets of multiple VLANs (a so-called *trunk link*), where each packet [1] is tagged with a VLAN ID. Switches are responsible for adding and removing the tags when forwarding between trunk links and links dedicated to only one VLAN.

To connect a host to multiple VLANs, the physical interface (ethX) must be connected to a trunk link. The physical interface represents the packets as is, that is the interface sees all packets with the tag added. For each VLAN *Y* on the physical interface *ethX* a virtual interface *ethX.Y* must be created. It handles only the packets belonging to that VLAN and should be configures with an IP address matching that VLAN.

The physical interface itself should not have an IP address configured and should not be started automatically. This can be achieved by setting the Univention Configuration Registry variables `interfaces/$INTERFACENAME/type` to `manual` and `interfaces/$INTERFACENAME/start` to `false`.

[1] Some switches also support mixing tagged with un-tagged packets on a single link, but for sake of simplicity that scenario is not discussed here.

1.3.1. Hardware requirements

Feedback

The network switches must support 802.1q VLANs.

1.3.2. Software requirements

Feedback

To configure VLANs the package ***vlan*** needs to installed and the kernel module `8021q` must be loaded. To ensure automatic loading of the module `;8021q` can be appended to the Univention Configuration Registry variable `kernel/modules`.

1.3.3. Example configuration

Feedback

The following example configures eth0 as a trunk link, which carries tagged packets of VLAN 2 and VLAN 3.

The IP addresses are assigned to the untagged virtual interfaces eth0.2 and eth0.3. eth0.2 is used as the primary interface for all host related services.

```
grep -v '^#' <<END_CONFIG | xargs -d '\n' ucr set
# Configure physical interfaces
interfaces/eth0/type=manual
interfaces/eth0/start=false
# Configuration for VLAN 2
interfaces/eth0.2/start=true
interfaces/eth0.2/address=192.168.122.13
interfaces/eth0.2/broadcast=192.168.122.255
interfaces/eth0.2/netmask=255.255.255.0
interfaces/eth0.2/network=192.168.122.0
# Configuration for VLAN 3
interfaces/eth0.3/start=true
interfaces/eth0.3/address=10.200.17.1
interfaces/eth0.3/broadcast=10.200.17.255
interfaces/eth0.3/netmask=255.255.255.0
interfaces/eth0.3/network=10.200.17.0
# Use VLAN 2 for host services
interfaces/primary=eth0.2
END_CONFIG
```

For more informations on configuring VLANs see the vlan-interfaces(5) manual page.

1.4. Combined setup

Feedback

Bonding, VLANs and bridging can also be combined.

The following example assumes the host is used to run several virtual machines. While the host itself is considered trusted, the virtual machines are considered untrusted, since the host administrator has no control over them and cannot guarentee their proper (network) configuration.

The host in this example has three physical network interfaces: One interface (eth2) is dedicated to the host itself and is the only interface with a IP address configured on the host. The other two interfaces (eth0, eth1) are used to connect the virtual machines to the VLANs 2 and 3. These physical interfaces use bonding (bond0) to provide high availability. For each VLAN (2 and 3) a bridge is created on the host, to which the virtual

machines are selectively connected to get access to the corresponding VLAN. This ensures that the virtual machines only see that traffic and cannot use the VLAN tools themselves to access other (prohibited) VLANs.

The interfaces must be ordered bottom-up, because otherwise the interfaces can not be brought up or down automatically. For that the correct order of the interfaces in `/etc/network/interfaces` is explicitly specified using the Univention Configuration Registry variables `interfaces/*$INTERFACENAME*/order`.

```
grep -v '^#' <<END_CONFIG | xargs -d '\n' ucr set
# Configure dedicated eth2 for the host
interfaces/eth2/start=true
interfaces/eth2/order=1
interfaces/eth2/address=192.168.122.13
interfaces/eth2/broadcast=192.168.122.255
interfaces/eth2/netmask=255.255.255.0
interfaces/eth2/network=192.168.122.0
interfaces/primary=eth2
# Configure low-level eth0 and eth1
interfaces/eth0/start=false
interfaces/eth0/order=2
interfaces/eth0/type=manual
interfaces/eth1/start=false
interfaces/eth1/order=2
interfaces/eth1/type=manual
# Configure bonding on eth0 and eth1
interfaces/bond0/start=true
interfaces/bond0/order=3
interfaces/bond0/type=manual
interfaces/bond0/options/1=bond-slaves eth0 eth1
interfaces/bond0/options/2=bond-mode 1
interfaces/bond0/options/3=bond-miimon 100
interfaces/bond0/options/4=bond-primary eth0 eth1
# Configure VLAN 2 and VLAN 3
interfaces/bond0.2/start=false
interfaces/bond0.2/order=4
interfaces/bond0.2/type=manual
interfaces/bond0.3/start=false
interfaces/bond0.3/order=4
interfaces/bond0.3/type=manual
# Configure bridges for VM
interfaces/br2/start=true
interfaces/br2/order=5
interfaces/br2/type=manual
interfaces/br2/options/1=bridge_ports bond0.2
interfaces/br2/options/2=bridge_fd 0
interfaces/br3/start=true
interfaces/br3/order=5
interfaces/br3/type=manual
interfaces/br3/options/1=bridge_ports bond0.3
interfaces/br3/options/2=bridge_fd 0
END_CONFIG
```

1.5. Setup for UCS Virtual Machine Manager

Feedback

UCS Virtual Machine Manager normally replaces eth0 with a bridge interface to connect virtual machines to a network. This was achieved by two init scripts renaming the physical interface eth0 to peth0 and creating a bridge called eth0, into which the physical interface was connected.

If you configure your own bridges, bonds or VLANs using the Univention Configuration Registry variables described above you need to deactivate these scripts by setting the following UCR variables:

```
# for KVM:
ucr set uvmm/kvm/bridge/autostart=no
# for Xen:
ucr set xen/bridge/interface=none
```

1.6. Troubleshooting

Feedback

The name of an interface depends on the loading order of kernel modules and on the timing the hardware needs to reach an active state. `udev` tries to assign persistent names to the interfaces based on their MAC address. The status is stored in `/etc/udev/rules.d/70-persistent-net.rules`. If interfaces get removed or replaced by other interfaces with a different MAC address, old named won't get reused until that file is reset.