UDM nSpawn Setup

Links

How to Create nSpawn Container GitHub
Setting Up Containers with Systemd-nSpawn

Instructions

nSpawn Container Setup

1. Enable SSH on UDM and SSH to UDM.
2. Run **apt full-upgrade**.(Default accept UPnP to be not installed and disabled)
3. **apt remove -y unifi-talk unifi-access unifi-protect uid-agent**
4. **apt -y install systemd-container debootstrap**
5. **mkdir -p /data/custom/machines**
6. **cd /data/custom/machines**
7. **debootstrap --include=systemd,dbus unstable debian-custom**
8. **systemd-nspawn -M debian-custom -D /data/custom/machines/debian-custom**
9. **passwd root**
10. **systemctl enable systemd-networkd**
11. **echo "nameserver 1.1.1.1" > /etc/resolv.conf**
12. **echo "debian-custom" > /etc/hostname**
13. **exit**

Container Configuration

1. **mkdir -p /data/on_boot.d && cd /data/on_boot.d**

2. **curl -LO** https://raw.githubusercontent.com/peacey/unifios-utilities/nspawn/nspawn-container/scripts/10-setup-network.sh

3. **vi 10-setup-network.sh**

10-setup-network.sh

```
#!/bin/bash
# This script will create a macvlan bridge interface to allow communication
# between container networks and host networks.
# An interface called brX.mac will be created, where X = $VLAN configured
below.
# The interface will be assigned an IP of $IPV4_GW, and $IPV6_GW configured
below.
# Routes will be added for the container IP $IPV4 and $IPV6.
# Script is based on 10-dns.sh from unifios-utilities.

## CONFIGURATION VARIABLES

# VLAN ID network container will be on. This VLAN has to first be configured as
a
# network in Unifi Network settings with a unique IP/subnet. Do not use the
same
# IP in the unifi network settings as you will use below for IPV4_IP or
IPV4_GW.
VLAN=10

# IP addresses of container.
IPV4_IP="10.1.10.2"
# Gateway IP address of macvlan interface. IP above should be in this subnet.
IPV4_GW="10.1.10.1/24"

# IPv6 container and gateway addresses. These can be empty if not using IPv6.
# Preferably generate your own ULA instead of using the default one below.
# A public IPv6 prefix based on your ISP's prefix can be used too, but any
# prefix changes for dynamic IPv6 prefixes have to be modified manually.

## END OF CONFIGURATION

# set VLAN bridge promiscuous
ip link set "br${VLAN}" promisc on

# create macvlan bridge and add IPv4 IP
ip link add "br${VLAN}.mac" link "br${VLAN}" type macvlan mode bridge
```

```
ip addr add "${IPV4_GW}" dev "br${VLAN}.mac" noprefixroute

# (optional) add IPv6 IP to VLAN bridge macvlan bridge
if [ -n "${IPV6_GW}" ]; then
  ip -6 addr add "${IPV6_GW}" dev "br${VLAN}.mac" noprefixroute
fi

# set macvlan bridge promiscuous and bring it up
ip link set "br${VLAN}.mac" promisc on
ip link set "br${VLAN}.mac" up

# add IPv4 route to container
ip route add "${IPV4_IP}/32" dev "br${VLAN}.mac"

# (optional) add IPv6 route to container
if [ -n "${IPV6_IP}" ]; then
  ip -6 route add "${IPV6_IP}/128" dev "br${VLAN}.mac"
fi
```

4. **mkdir -p /etc/systemd/nspawn && cd /etc/systemd/nspawn**

5. **vi debian-custom.nspawn**

debian-custom.nspawn

```
[Exec]
Boot=on

[Network]
MACVLAN=br10
ResolvConf=off
```

6. **cd /data/custom/machines/debian-custom/etc/systemd/network**

7. **vi mv-br10.network**

mv-br10.network

```
[Match]
Name=mv-br10

[Network]
IPForward=yes
Address=10.1.10.2/24
Gateway=10.1.10.1
```

8. **chmod +x /data/on_boot.d/10-setup-network.sh**

9. **/data/on_boot.d/10-setup-network.sh**
10. **ln -s /data/custom/machines/debian-custom /var/lib/machines/debian-custom**
11. **systemctl enable --now systemd-nspawn@debian-custom**
12. **machinectl start debian-custom**
13. **machinectl shell debian-custom**
14. **systemctl enable systemd-networkd**
15. **echo "nameserver 1.1.1.1" > /etc/resolv.conf**
16. **ip addr show** (Should now see the vlan and IP address assignment from 10-setup-network.sh)
17. **ping -c4 1.1.1.1**
18. Exit container to host OS: **vi /etc/systemd/system/setup-network.service**

setup-network.service

```
[Unit]
Description=Setup custom container network service
Wants=network-online.target
After=network-online.target

[Service]
Type=oneshot
ExecStart=/data/on_boot.d/10-setup-network.sh
RemainAfterExit=yes

[Install]
WantedBy=multi-user.target
```

19. **systemctl enable setup-network**

Setup nSpawn Persistence Across Firmware Updates

1. **mkdir -p /data/on_boot.d && cd /data/on_boot.d**

2. **curl -LO** https://raw.githubusercontent.com/peacey/unifios-utilities/nspawn/nspawn-container/scripts/0-setup-system.sh

3. **chmod +x /data/on_boot.d/0-setup-system.sh**

4. **mkdir -p /data/custom/dpkg && cd /data/custom/dpkg**

5. **sudo apt download systemd-container libnss-mymachines debootstrap arch-test**

6. **vi /etc/systemd/system/setup-system.service**

   setup-system.service

   ```
   [Unit]
   Description=Setup custom container service
   Wants=network-online.target
   After=network-online.target

   [Service]
   Type=oneshot
   ExecStart=/data/on_boot.d/0-setup-system.sh
   RemainAfterExit=yes

   [Install]
   WantedBy=multi-user.target
   ```

7. **systemctl enable setup-system**