A declarative approach to Linux networking with Netplan

DebConf23, Kochi/India

Lukas Märdian – slyon@debian.org | slyon@ubuntu.com
Netplan – Declarative network configuration

Table of contents

• About myself
• What is Netplan?
  • But why?
  • Some History
• Netplan usage
  • Status quo
• Future work
  • QA
About myself
About myself – Lukas ‘slyon’ Märdian

- Based in Germany
  - First time at DebConf, please be kind ;-) 
- Contributor since 10+ years
  - Debian pkg-fso team, Openmoko anyone?
- Ubuntu Foundations
  - At Canonical since 2020
- DM since 2021, DD since 2023

---

Netplan Maintainer  Ubuntu Core-Dev  Debian Developer
early 2020   late 2020   mid 2023

slyon@debian.org | slyon@ubuntu.com
What is Netplan?
What is Netplan?

Common interface for networking daemons

**YAML Frontend**

Some sugar on top: SR-IOV, OVS, regdom, Interactive CLI

Implemented as systemd-generator

**Slim one-off binary**

systemd-networkd & NetworkManager Backend support
What is Netplan?

Declarative network config

Netplan reads network configuration from `/etc/netplan/*.yaml` written by admins, installers, cloud image metadata or other OS deployments.

During early boot, Netplan generates backend specific configuration files in `/run` to hand off control of interfaces to a particular backend:

- systemd-networkd
- NetworkManager
Section #1.1

But ... why?
Why Netplan?

Keep simple things simple, but allow for custom setups of any complexity.

- IPv6, bridges, bonds, ifupdown, VRFs, VXLANs, ifupdown-ng, tunnels, declarative, iproute2, NetworkManager, access-points, dummy, routing policy, veth, WWAN, ifupdown2, VLANs, ethernet, imperative, InfiniBand, DNS, systemd-networkd, hardware offloading, DHCP, sysctl, router-advertisement

- Network ecosystem is scattered
- Active upstream & feature development
- Modern codebase
- Test automation
- Combining the best of two worlds
Why Netplan?

Keep simple things simple, but allow for custom setups of any complexity.

- **systemd-networkd**
  - Slim, static, text-based configuration
  - Cloud-init integration

- **NetworkManager**
  - Interactive UI, dynamic WiFi, cellular
  - Desktop integration

- Unify config file path
- Good test automation
- Detailed documentation, `netplan(5)`, [netplan.rtfd.io](https://netplan.rtfd.io)
Some History
Some History

Ubuntu’s default networking tool

- Default Ubuntu networking **since 2016**
  - 18.04 LTS, 20.04 LTS & 22.04 LTS …
  - Dependency based boot
  - Unify config files across server & desktop
  - Unify cloud metadata and & installers
- Used by millions of users
- Available on AWS, GCP and other public clouds, through cloud-init
- Available on Debian, Arch Linux, Fedora, EPEL/RHEL, Ubuntu, …
Section #2

Netplan usage
Netplan usage: simple

Simple example & drop-in configuration

/etc/netplan/30-user.yaml

<table>
<thead>
<tr>
<th>network:</th>
</tr>
</thead>
<tbody>
<tr>
<td>version: 2</td>
</tr>
<tr>
<td>ethernets:</td>
</tr>
<tr>
<td>enp3s0:</td>
</tr>
<tr>
<td>dhcp4: true</td>
</tr>
<tr>
<td>dhcp6: true</td>
</tr>
</tbody>
</table>

Output generated:

/run/systemd/network/10-netplan-enp3s0.network
Netplan usage: simple

Simple example & drop-in configuration

/etc/netplan/30-user.yaml

```yaml
network:
  version: 2
  ethernets:
    enp3s0:
      dhcp4: true
      dhcp6: true
```

Output generated:

/run/systemd/network/10-netplan-enp3s0.network

/usr/lib/netplan/00-nm-all.yaml

```yaml
network:
  version: 2
  renderer: NetworkManager
```

/run/NetworkManager/system-connections/netplan-enp3s0.nmconnection
Netplan usage: complex

Open vSwitch example

```yaml
network:
  version: 2
  ethernets:
  - eth0: {}
  - eth1: {}
  bonds:
    ovsbond:
      interfaces: [eth0, eth1]
      openvswitch:
        lACP: active
  bridges:
    ovsbr:
      addresses: [192.170.1.1/24]
      interfaces: [ovsbond]
```

Netplan usage: CLI & API

Interactive runtime configuration

- Netplan CLI
  - Written in Python (optional)
  - “netplan apply/try”
  - “netplan get/set”
  - “netplan status”
- Making use of libnetplan.so
  - YAML parser
  - Logical validation of configuration
  - Python bindings available in python3-netplan
Section #2.1

Status quo
Status quo: Features

Supported technologies

- Ethernet, InfiniBand
- WiFi + AP, cellular
- Bridge, Bond
- VLAN, veth, dummy
- Routing, PBR, VRF
- OVS, SR-IOV
- Tunnels (WireGuard, IPIP, VXLAN, GRE, …)
- DNS, regulatory-domain
Status quo: Extensions & Side-by-Side

Native extensions

**Systemd-networkd (override)**
/run/systemd/network/10-netplan-eth0.network
/etc/systemd/network/10-netplan-eth0.network.d/override.conf

**NetworkManager (keyfile)**

YAML setting: `networkmanager.passthrough`

https://netplan.io/reference#properties-for-device-type-nm-devices

Plays nice with 3rd parties

- Custom scripts
- iproute2
- Open vSwitch
- ifupdown{2,-ng}
Status quo: d-i & cloud-images

- Netplan is used in Debian cloud-images since Bookworm (Kudos!)
  - Cloud-init integration
  - Playground for hands-on experience
    - Netplan and systemd-networkd on Debian Bookworm – SlyBlog
- ifupdown & NetworkManager in debian-installer
  - Many using systemd-networkd
  - Plenty of discussions: ifupdown{2,-ng} vs NetworkManager vs systemd-networkd
    - DebConf’21 Cloud BoF
    - MiniDebConf’23 Hamburg
    - Re: Bug#995189: RFH: isc-dhcp
    - Re: proposal: dhcpcd-base as standard DHCP client starting with Trixie
Section #3

Future work
Future work

• Netplan 1.0 release due 2024
  • stable API & ABI for 3rd party integrations
• Documentation improvements
  • Salsa: debian/debian-reference (!14)
• Debian-installer changes?
  • Salsa: installer-team/netcfg (!9)
Thank you. Questions?

Lukas Märdian – slyon@debian.org | slyon@ubuntu.com
Ubuntu Foundations, Software Engineer @ Canonical
FAQ

- Does it have a Python dependency?
  - No. Just the optional CLI, not the `netplan-generator` package.
- How are interactive changes in NetworkManager handled?
  - Bi-directional NetworkManager integration patch available.
- What about hooks?
  - Networkd-dispatcher, NM dispatcher (https://netplan.io/faq#use-pre-up-post-up-etc-hook-scripts)
- Is it just another competing standard?
  - No. It integrates nicely with native configuration & can be used in parallel.
Netplan usage: CLI

```
lukas@abacony:~$ netplan status
  Online state: online
  DNS Addresses: 127.0.0.53 (stub)
  DNS Search: ff-sw.net

● 1: lo ethernet UNKNOWN/UP (unmanaged)
  MAC Address: 00:00:00:00:00:00
  Addresses: 127.0.0.1/8
  ::1/128
  Routes: ::1 metric 256

● 9: wlan0 wifi"freifunk-weinstrasse.de" UP (NetworkManager: NM-b6b7a):
  MAC Address: f4:a4:75:a2:6c:5f (Intel Corporation)
  Addresses: 10.210.20.75/20
  2a03:2260:300c:100::3418:ec08:7f99:a525/64
  2a03:2260:300c:100::bfc5:b4af:a09e:a44b/64
  fe80::1b29:d4a5:c7a0:22ed/64 (link)
  DNS Addresses: 10.210.16.4
  2a03:2260:300c:100::4
  2a03:2260:300c:100::5
  2a03:2260:300c:100::6
  DNS Search: ff-sw.net
  Routes: default via 10.210.16.4 metric 600 (dhcp)
  10.210.16.0/24 from 10.210.20.75 metric 600 (link)
  2a03:2260:300c:100::/64 metric 600 (ra)
  fe80::/64 metric 1024

7 inactive interfaces hidden. Use "--all" to show all.
```