Agenda / Content:

• History
• Motivation
• The Scope

• Linux Networking
• Open Networking
• Container Networking
• Automation/Orchestration
a quick private history lesson

RS-6000/AIX
SPARC/SunOS
3745
CIP/OSA
OS/2
Novell
NT
DOS/WIN3.1
MAU/Hub
Bridge
Router

SDN Controller

Same CLI

Same CLI

Grossartig!
Motivation

Layer 3 & EVPN

CI/CD   Virtual Environment   Production Environment
Motivation

Let's have a look at the OS architecture
Architecture (Linux / Cumulus Linux)

- FRR
- iproute2
- brctl
- ifupdown2
- snmpd
- Lldpd
- ssh
- iptable
- ip6table
- ethtool
- lldpd
- develop network synchronisation via netlink/switchd
- ARP Table
- Routing Tables
- Bridge FDB
- Filter Tables
Automatisation / Orchestration

Control-Plane

- FRR
  - mstpd
  - ethtool
  - ifupdown2
  - snmpd
  - lldpd
  - ssh
  - iptable
  - ip6table
  - ebtable
  - switchd
  - netq

/etc
/daemon
/networks/interfaces
/cumulus/ports.conf
/daemon/frr
/frr.conf

Package
User/Groups/etc
Side-Note

• Options: Network CLI

$ net add interface swp1 ip address 10.1.3.11/24
$ net pending
$ net commit

- hosts: leaf01
tasks:
  - name: Phys.Schnittstelle
    nclu:
      commands:
        - add interface swp1 ip address 10.1.3.11/24
“What“ do we REALLY need to automate?

- **Use-Case: Server network-interfaces**
  place-holder packages

- **Use-Case: Switch network-interfaces routing-protocol(s)** packages

```yaml
hosts: server-switch
  name: example
  become: yes
  tasks:
    - name: Ifs
c      copy:
        src: /...
d      dst: /etc/network/interfaces
```

and a bit of magic
Interface Manager

- New implementation in Python
- Backward compatibility with debian ifupdown
- Pluggable architecture with python add-on modules
- Ordered Network Interface dependency relationship handling
- Incremental changes and query live configuration

# git clone git://github.com/CumulusNetworks/ifupdown2
# sudo apt-get install make
# cd ifupdown2 && git checkout master-next && make deb
# sudo dpkg -i ./ifupdown2_1.2.1_all.deb
$ sudo ifquery

Dependencies: demo leaf01

cumulus@leaf01:mgmt-vrf:~$ ifquery -a -p list
lo : []
eth0 : []
mgmt : ['eth0']
swp1 : []
swp2 : []
TEST : ['swp2']
swp51 : []
bridge : ['swp1', 'vni-10010']
vni-10010 : []
vlan10 : ['bridge']
Interface Manager

- hosts: spine01
  name: Konfiguration der IF
  become: yes
  tasks:
    - name: IFs for BGP unnumbered
      copy:
        src: /home/cumulus/on/on-001/on-spine01-001
        dest: /etc/network/interfaces
        # aktivieren
    - name: Durchstarten der Schnittstelle
      shell: /sbin/ifreload -a
stolen from Eric

---
- name: Load Interface Configuration
copy: src=:/config/{{ansible_hostname}}/interfaces dest=/etc/network/

- name: Apply Interface Configuration Delta
  shell: nohup bash -c 'sleep 2 && /sbin/ifreload -a > /tmp/ifreload.out 2>&1' &
  async: 1
  poll: 0
  changed_when: true

---

stolen from Attilla

```
auto lo
iface lo inet loopback
  address {{node[inventory_hostname]["routing"]["lo"]["addr"]}}
{% if node[inventory_hostname]["overlay"]["anycast-ip"] is defined %}
  claqd-vxlan-anycast-ip {{ node[inventory_hostname]["overlay"]["anycast-ip"]["addr"]}}
{% endif %}
{% endif %}

auto eth0
iface eth0 inet dhcp
  vrf mgmt

auto mgmt
iface mgmt
  address 127.0.0.1/8
  vrf-table auto

{% if node[inventory_hostname]["overlay"] is defined %}
{% set vsrs = [] %}
{% for irb in node[inventory_hostname]["overlay"]["irb"] %}
  {% if node[inventory_hostname]["overlay"]["irb"]["vrf"] is defined %}
    auto {{node[inventory_hostname]["overlay"]["irb"]["vrf"]}}
   {% endif %}
  {% endif %}
{% endfor %}
{% endif %}

iface {{node[inventory_hostname]["overlay"]["irb"]["vrf"]}}
  vrf-table auto
```
Layer 2 and Layer 3 (a quick private history)

Once upon a time I had a CAT…

telnet 10.0.0.1 or session 3

> enable
# show run
# conf t

router bgp 65000
router-id 1.1.1.1
Open source routing software suite

- Linux Foundation project
- [https://frrouting.org](https://frrouting.org)
- [https://github.com/FRRouting/frr](https://github.com/FRRouting/frr)
- Presents an industry-standard CLI
- Robust protocol set
FRRouting Architecture

Routing Database
- bgpd
- ospfd
- ospf6d

Routing Information Database (RIB)
- zebra

Forwarding Information Database (FIB)
- Kernel Routing Table
  - Switchd

Hardware Acceleration Of FIB

netlink
- hosts: switches
  name: Configure /etc/frr/daemon
  become: yes
  tasks:
    - copy:
        src: /home/cumulus/on/on-001/daemons
        dest: /etc/frr/daemon
    notify:
      - frr_restart
        - copy:
            src: /home/cumulus/on/on-001/frr_spine01
            dest: /etc/frr/frr.conf
            notify:
              - frr_reload

reload vs restart
BGP Configuration Evolution

Reference

- eBGP
- IP addresses
- AS numbers
- Slow timers
- Unique info for each node
  - "external" information to this node's config needed

Cumulus Linux (+ upstream FRR)
- Simple configuration
  - Ideal for automating
- Enable BGP on an interface
- Removes IP addressing and AS numbers

Traditional/Incumbent Vendor
- Complex configuration
  - IP addresses
  - AS numbers
- Slow timers
- Unique info for each node
- "external" information to this node's config needed

```
router bgp 65002
bgp log-neighbor-changes
bgp router-id 10.0.0.17
neighbor swp2 remote-as external

interface eth1
address 1.1.1.33 255.255.255.252
```
IPv6 link local address for BGP sessions (not really unnumbered)
- RFC 5549 advertises IPv4 addresses over IPv6 session
- IPv6 router advertisement to learn neighbor’s link local address
Real World Data-Center Fabric

- Network Admin
- NetOps
- Cloud Admin
- Admin 2.0
- Admin NG
- ... Securing your Future
- Compute Admin
- DevOps

Out-Of-Band Network

Superspine

Spine

EVPN

ToR

K8S

Border ToR (summarize)
Live and/or "outside..."
Example-01: EVPN decentralized symmetric routing

ansible-playbook clean-leaf01.yml
ip route
net show route

ansible-playbook evpn-leaf01.yml
ip route
net show route
net show route vrf vrf1

ping 10.1.3.1
ping 10.1.3.11
ping 10.1.3.13
ping 10.1.3.103
Example-01: EVPN decentralized symmetric routing

```yaml
---
- hosts: leaf01
  name: clean it up
  become: yes
  gather_facts: no
  tasks:
    - name: clean e/n/i
      copy:
        src: /home/cumulus/on3/leaf01-clean
        dest: /etc/network/interfaces
    - name: activate
      shell: /sbin/ifreload -a

---
- hosts: leaf01
  name: evpn leaf01
  become: yes
  gather_facts: no
  tasks:
    - name: configure evpn e/n/i
      copy:
        src: /home/cumulus/on3/leaf01-evpn
        dest: /etc/network/interfaces
    - name: activate
      shell: /sbin/ifreload -a
```
Example-02: EVPN decentralized multi-tenant symmetric routing

```bash
ansible server02 -a "sudo ifdown eth2"
ansible leaf02 -a "sudo ifdown swp2"
```

```bash
ansible server02 -a ip route
ansible leaf02 -a ip route
ansible leaf02 -a ip route show vrf vrf1
```

```bash
ansible-playbook leaf02-server02.yml
ip route
net show route
net show route vrf vrf1
```

```bash
ping 10.2.4.102
```

```bash
ifupdown2
```

```bash
ping 10.2.4.102
```
Example-01: EVPN decentralized symmetric routing

```yaml
---
- hosts: server02
  name: set it up
  become: yes
  become_method: sudo
  gather_facts: no
  tasks:
    - name: setup e/n/i
      copy:
        src: /home/cumulus/on3/server02
        dest: /etc/network/interfaces
    - name: activate
      shell: /sbin/ifreload -a

- hosts: leaf02
  name: set ip up
  become: yes
  gather_facts: no
  tasks:
    - name: setup e/n/i
      copy:
        src: /home/cumulus/on3/leaf02-evpn
        dest: /etc/network/interfaces
    - name: activate
      shell: /sbin/ifreload -a
    - name: frr
      copy:
        src: /home/cumulus/on3/leaf02-frr.conf
        dest: /etc/frr/frr.conf
    - name: frr_restart
      service:
        name=frr
        state=restarted
        enabled=yes
```
Example-03: NS (Server-01) + Leaf01 (Static + Redistribution)

net show route vrf vrf1 | grep S
S>* 172.16.0.0/24 [1/0] via 10.1.3.101, vlan13, 1d02h13m
Example-04: K8S (Server-01) + Leaf01 (Static + Redistribution)

```plaintext
kubectl run nginx --image nginx --replicas 3

ip netns

sudo ip netns exec cni-78e7f747-1bfa-8431-3b8d-3a06bf0e178e ip route
sudo ip netns exec cni-78e7f747-1bfa-8431-3b8d-3a06bf0e178e ping 10.1.3.1/11/13/103
sudo ip netns exec cni-78e7f747-1bfa-8431-3b8d-3a06bf0e178e ping 10.2.4.102
sudo ip netns exec cni-78e7f747-1bfa-8431-3b8d-3a06bf0e178e traceroute 10.2.4.102
```
Example-05: FRR (Server-01)

BGP unnumbered
Works well to the server,
For illustrative/education purposes the legacy option with remote IPs and remote ASN
(== the more complicated variant) is shown
Open Networking

- Late 2012, brainstormed initial features
- Early 2013, evangelized with HW partners
- May 2013, first public demo at OCP MIT Workshop
- Summer 2013, first products available from multiple vendors
- Summer 2013, project incubated by OCP
- June 2014, project fully adopted by OCP
Open Networking

One tool-set to rule them all  One skill-set to rule them all

Simple:
Server == Network
Thank you!

Visit us at cumulusnetworks.com or follow us @cumulusnetworks