SCALEOUT

Configuration Management Tools - Software-Defined Networks

Max Andersson

- Primary goal of SDN
  - Open & Programmable
    - Traffic Engineering
    - Security
    - Qos
    - Routing
    - Switching
    - Virtualization
    - Monitoring
    - Load Balancing
    - New applications

- 3 Layer Model OS
  - South side
  - North side

- 3 Layer SDN model
  - Forwarding Device Southside
  - Net applications
  - NEtwork operating system
- Fast path , caches paths. Can ask operation system

- Forwarding devices / Data Plane
    - Can be hardware switches if they support openflow etc, but can also be software devices.

- Southbound Interface
    - OpenFlow
    - OVSDB -(Open vSwitch)
    - NETCONF
    - SNMP

- Application interfaces
  - Java Api
  - Northbound (e.g RESTConf)

- Layer details
  - Slicing, forwarding to different network operating systems
  - Logically centralized vs physically centralized
  - Clustered Network operating system.
  - Regions with EAST/WEST protocol
  - Hierarchies

- Availability and scalability
- SDN vs Traditional networks
    - Traditional networks has a control plane and data plane in one physical system

- BGP, Multiprotocol Label Switching (MPLS), Open Shortest Path First (OSPF)
    - Control plane pushes policies down to the fast path of the data plane.

- Openstack
    - Networks
    - Segmentation Methods
        - VLAN’s
        - VXLAN
        - GRE
        - Network Namespaces
        - OpenFlow Rules
        - Subnet
            - Default DHCP service (dnsmasq)’

- **Kubernetes**
  - Flannel
    - Why
      - Layer 2 solution
      - Simple & Mature
      - Overlays are useful when network address space is limite
      - Overlays auto-configureate
    - Scenario
      - On-Prem or custom cloud where native routing isn’t possible
    - Why not
      - Native routing is faster and easier to debug.
      - You need Calico if you want network policies.

Calico

- Why
    - Layer 3 solution
    - Good Network policy support
    - Default on most kubernetes distributions
    - Easy to debug on hosts by looking on route tables. BGP allows access inside and outside the cluster.

- Screnario
    - On-Prem with native routing or cloud kubernetes services

- Why Not
    - IP-in-IP mode is needed when routing between subnets, which negates some of the performance benefits vs an overlay
    - BGP can be scary.

Calico

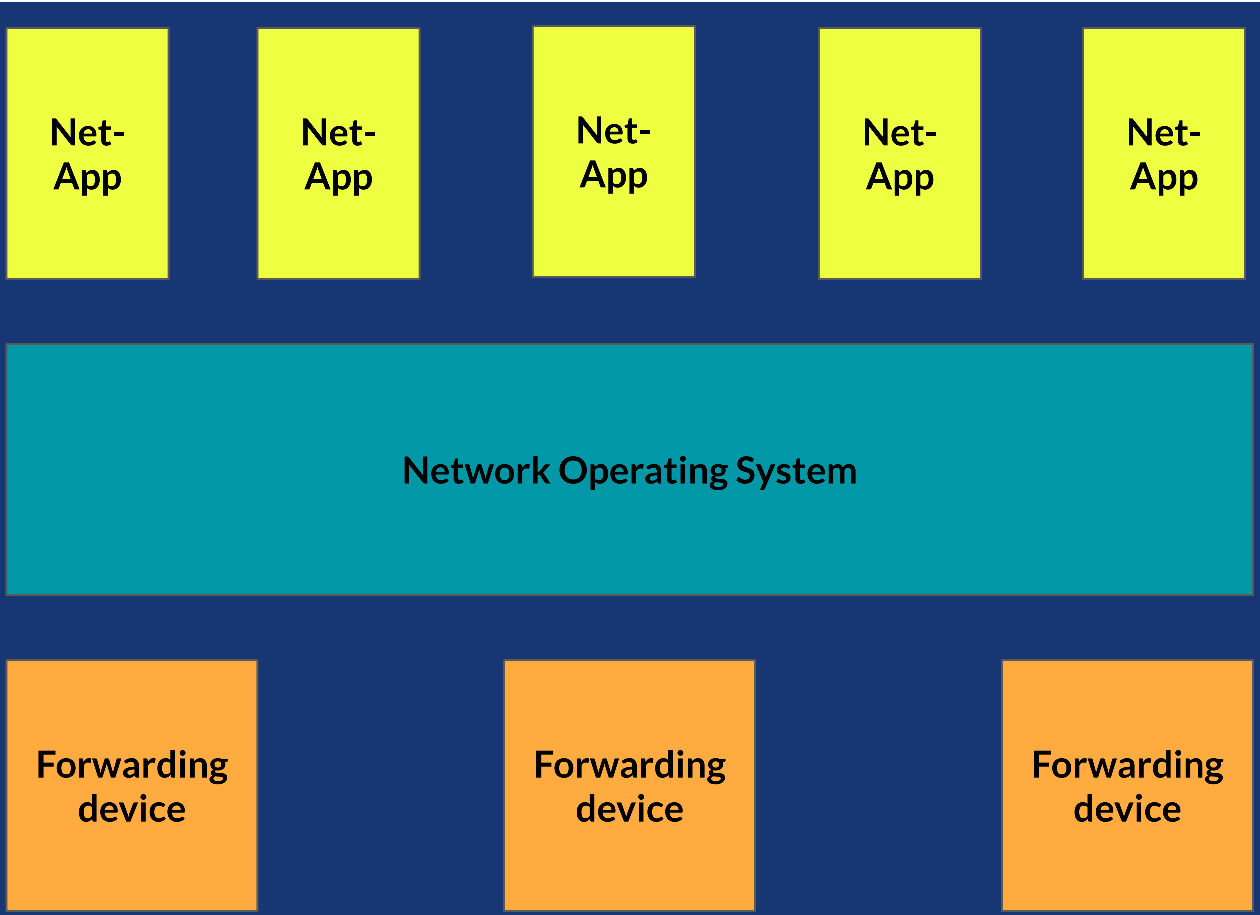
- Why
  - Layer 3 solution
  - BGP
  - Good Network policy support
  - Default on most kubernetes distributions
  - Easy to debug on hosts by looking on route tables. BGP allows access inside and outside the cluster.

- Screnario
  - ○ On-Prem with native routing or cloud kubernetes services

- Why Not
  - IP-in-IP mode is needed when routing between subnets, which negates some of the performance benefits vs an overlay
  - BGP can be scary.

- Kube-Router
    - Why
        - Layer 3 solution
        - BGP
        - Single Go binary built from the ground up for Kubernetes.
        - Uses new IPVS/LVS kernel features to improve service load balancing performance.
        - Also does direct server return to improve latency.

    - Scenario
        - On-Prem or custom cloud latency focus.
    - Why not
        - Similar to Calico in that it uses IP-in-IP by default to encapsulate traffic between subnets.
        - Quite a new project and although it's in use in production at some companies it's still not v1.



Net-App

Net-App

Net-App

Net-App

Net-App

Network Operating System

Forwarding device

Forwarding device

Forwarding device