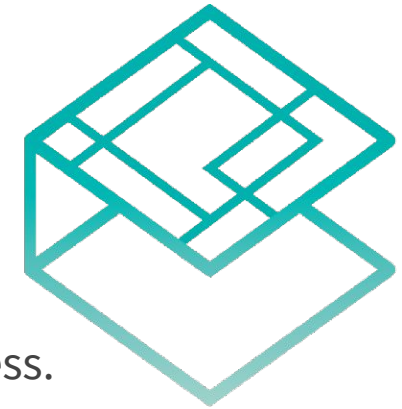
# Network Services on Kubernetes on-premise

Hans Duedal
Network Services Engineer @ NPF
CTO @ Visma e-conomic

NPF
VISMA

# Networking in Kubernetes



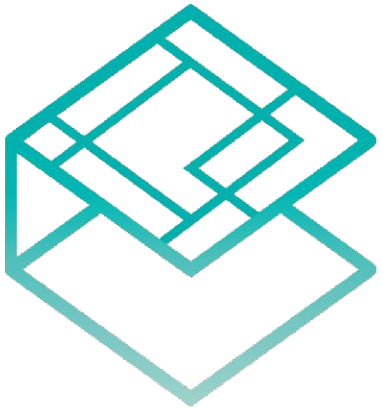
Kubernetes assumes that pods can communicate with other pods, regardless of which host they land on. Every pod gets its own IP address.

Fundamental requirements

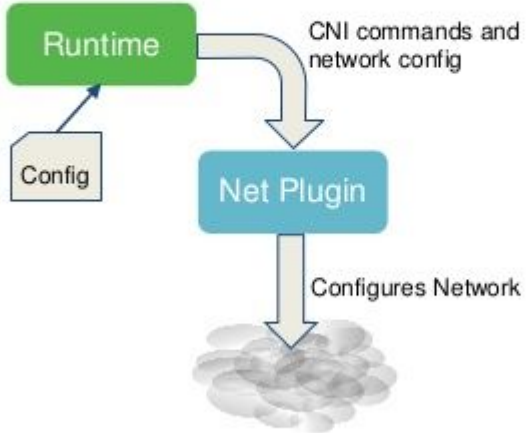
1. all containers can communicate with all other containers without NAT
2. all nodes can communicate with all containers (and vice-versa) without NAT
3. the IP that a container sees itself as is the same IP that others see it as

NPF VISMA

# Container Network Interface



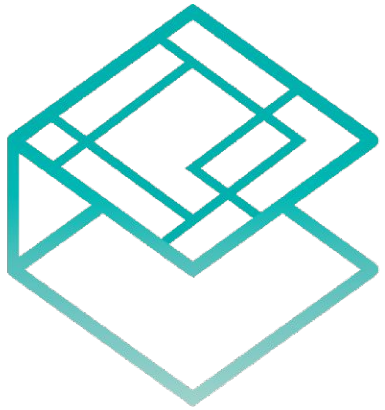
- Originated at CoreOS as part of rkt
- Cloud Native project
- Many container runtimes, ie. kubernetes, rkt, mesos
- If your network solution adheres to the CNI Spec, the runtimes can use it
- Sometimes the choice of CNI plugins are (almost) made for you, ie. on public clouds
- For an on-premise project, choosing the right CNI solution is critical to success




NPF

VISMA

# Kubernetes Network Solutions



CISCO
Cisco ACI

Contiv

big switch networks

Google Cloud Platform

vmware NSX

Juniper CONTRAIL

cilium

WEAVE

Bridge

PROJECT CALICO

nuagenetworks

flannel

MULTUS

KUBE-ROUTER

OvS
Open vSwitch

OVN
(Open Virtual Network)

ROMANA

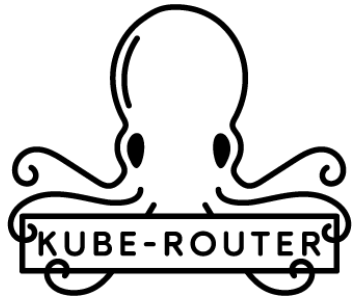
NPF

VISMA

# Project Calico

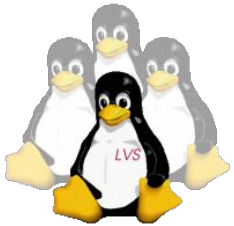

- IP-in-IP with option for BGP Based
- Independent of Kubernetes (ships with it's own etcd cluster)
- Backbone of Github.com's metal cloud
  https://thenewstack.io/github-goes-kubernetes-tells/
- Policy based access control
- Fully featured for Datacenter Level Kubernetes Networking
- Doesn't make much sense to use without BGP

# Kube-router




Three components

1. IPVS/LVS (Part of Linux kernel since 1999)
2. GoBGP
3. Kuberouter itself




- BGP Based
  https://cloudnativelabs.github.io/post/2017-05-22-kube-pod-networking/
- Heavy lifting by IPVS/LVS (in-kernel load balancer) and the routing tables
- Kubernetes specific
- Small and lightweight, with excellent community on slack
- Network Policies (but not as advanced as Calico)

NPF
VISMA

# flannel


- Full userspace software based solution, no hardware or kernel involved
- IP-in-IP based
- Often the goto/default choice for on-premise deployments
- Only network, no policy
- Networking: “Developer Edition”
- CoreOS project, but plays extremely well with Kubernetes
- The reference plugin for CNI

# How Rancher does it


- Magically network nodes in AWS with GCE or on-premise
- It’s all an illusion, it sets up a full mesh of IPSec (VPN) tunnels between each node
- High overhead of IPSec stack and encryption
- Easy to use, and most of the time “just works”

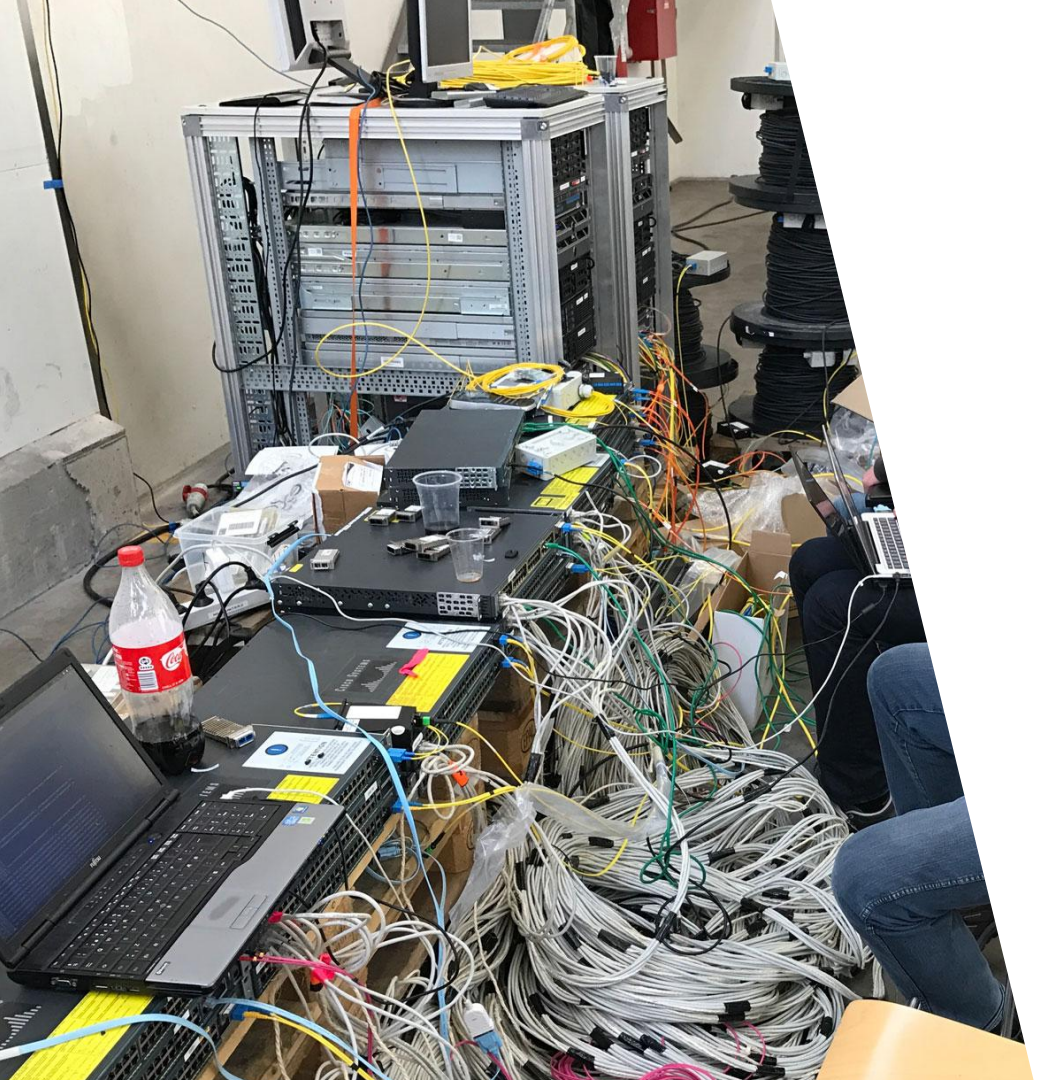
That’s all fine, but you said something about e-sports?


NPF

# NPF in Numbers

- 3600 Amps of mains, 3 phase power
- Almost 14 kilometers of network cables (2.4 km fiber, 11.5 km copper)
- 400+ volunteers
- 300+ network switches
- Built up in five days, 47 hour event, and then torn down in two days
- Almost 6000 gamers with their computers
- Internet consumption the size of Aalborg, so internet wise we become the fourth largest “city” in Denmark for a weekend

NPF
VISMA



# Network at NPF

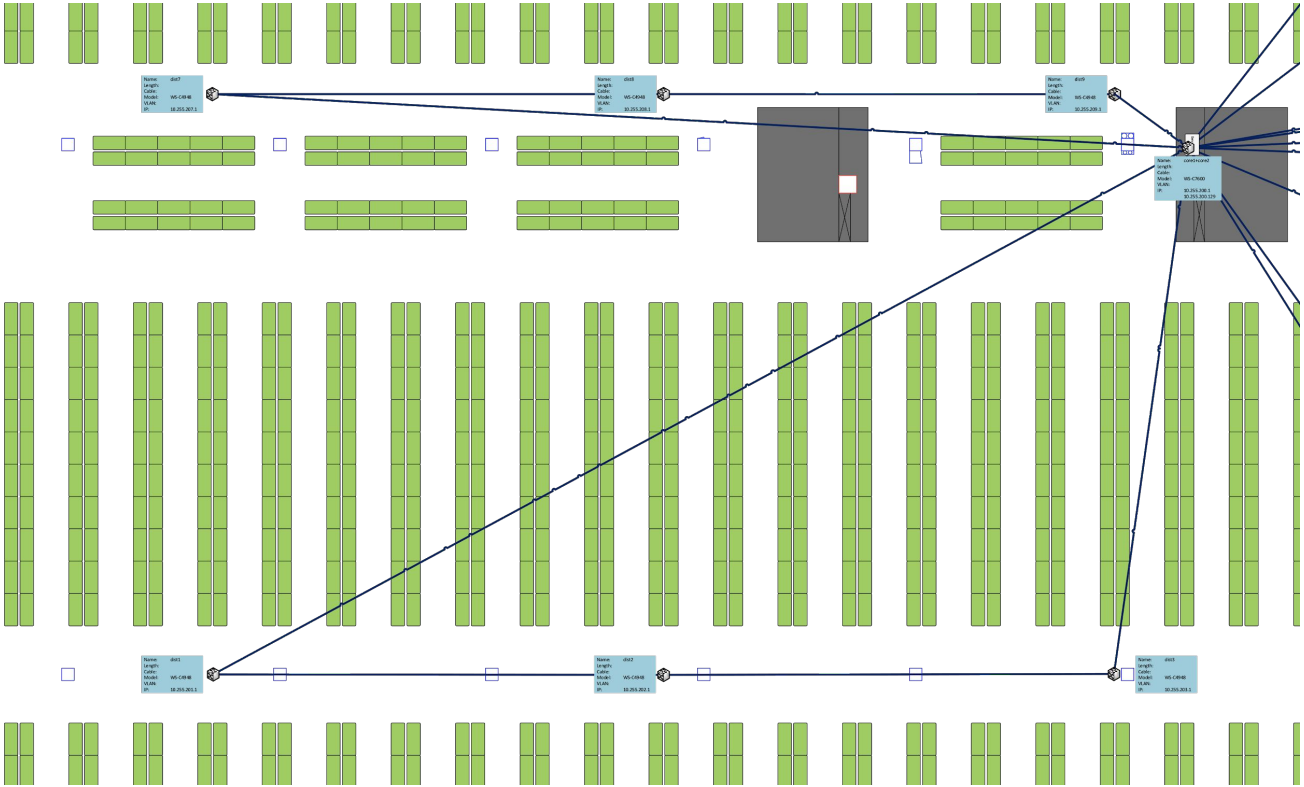
NPF

VISMA

Distribution Network

- Layer 3 Distribution
- Approx. 1100 Participants pr ring
- OSPF Routing

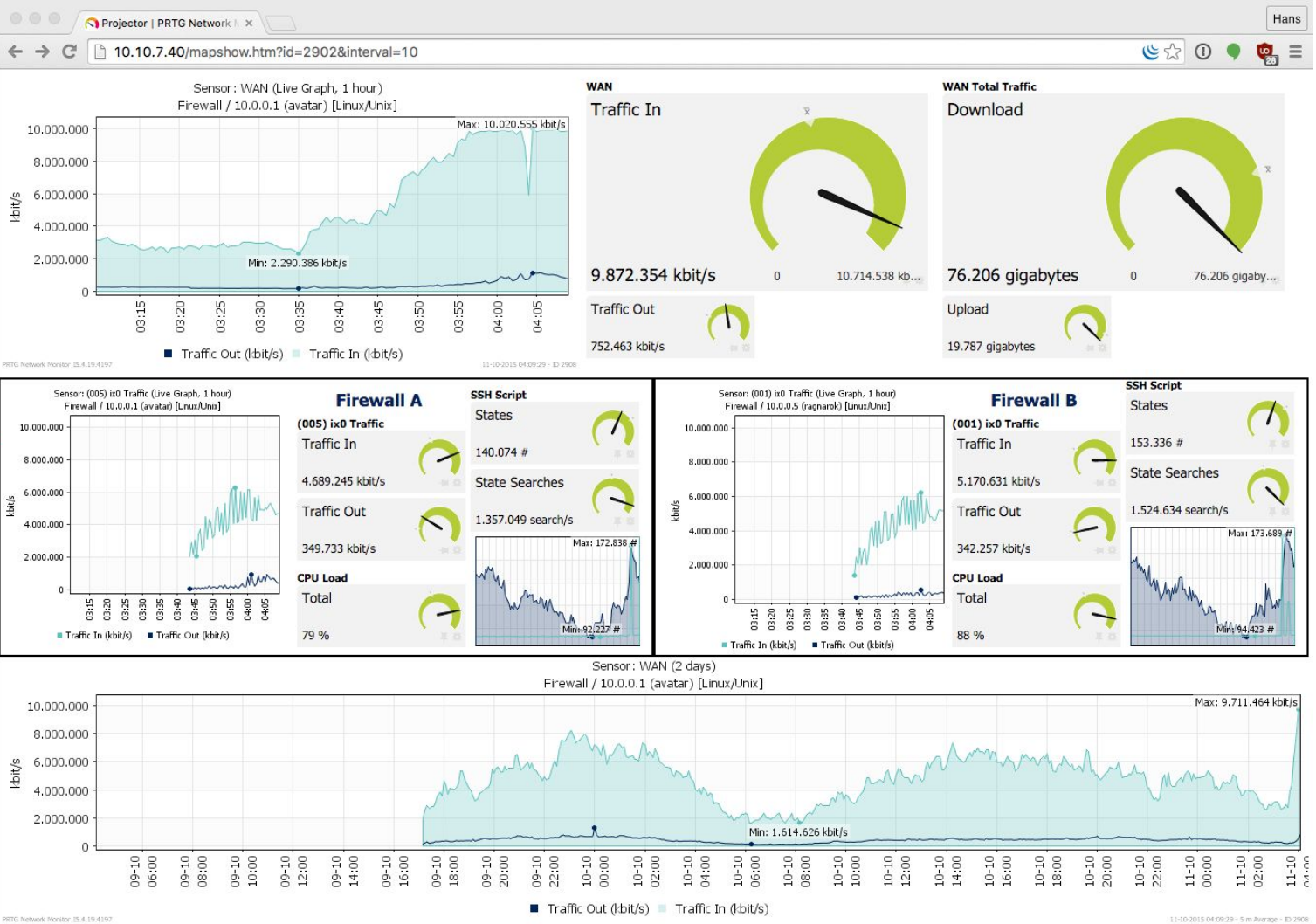
# Rings

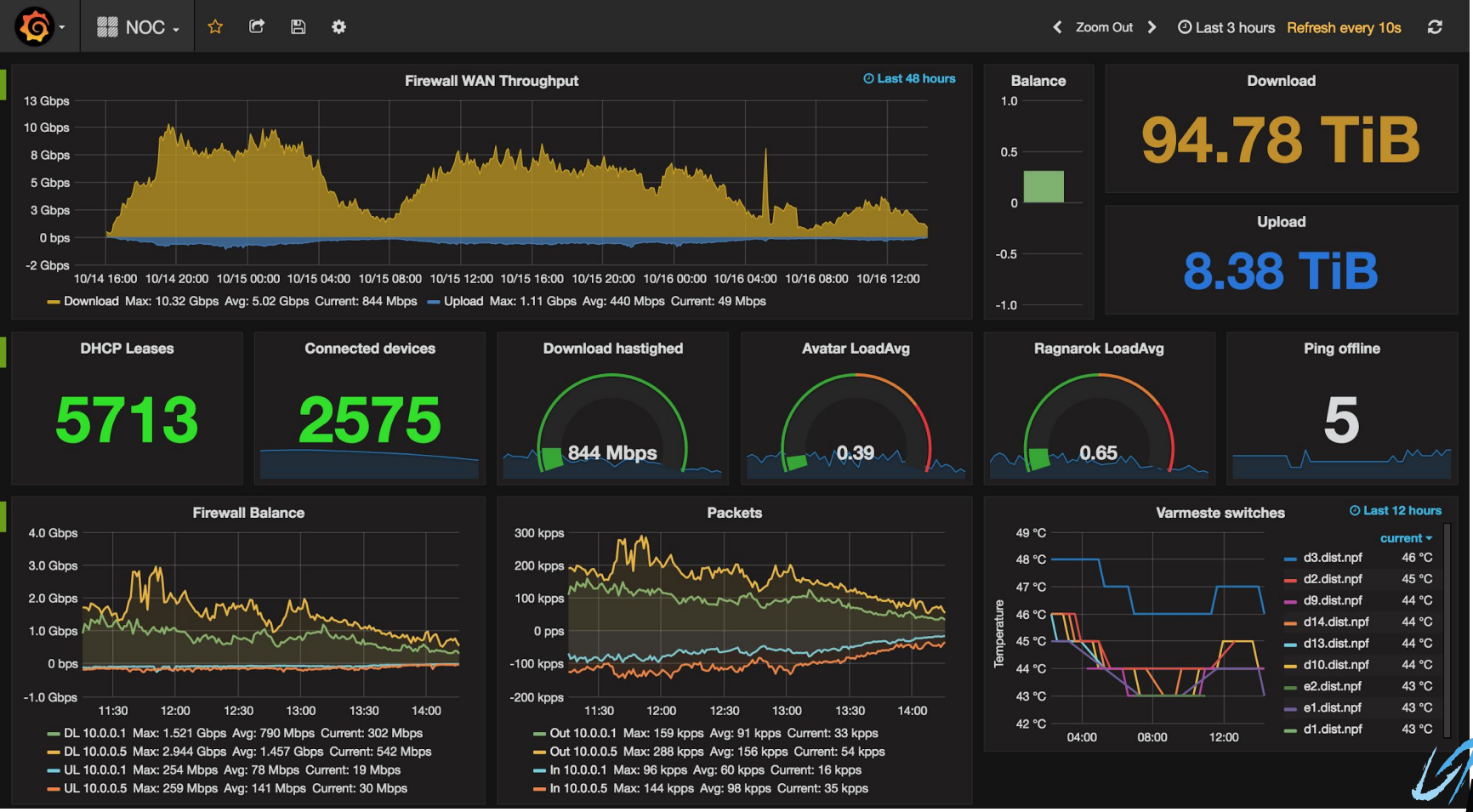


# Network “Services”?

- DHCP
- DNS
- Prometheus Monitoring
- Grafana
- SMNP Polling (Blackbox exporter)
- SNMP Traps (ELK Cluster)

NPF
VISMA


Projector | PRTG Network
Hans
10.10.7.40/mapshow.htm?id=2902&interval=10
Sensor: WAN (Live Graph, 1 hour)
Firewall / 10.0.0.1 (avatar) [Linux/Unix]
Max: 10.020.555 kbit/s
Min: 2.290.386 kbit/s
kbit/s
Traffic Out (kbit/s)
Traffic In (kbit/s)
WAN
Traffic In
9.872.354 kbit/s
0
10.714.538 kb...
Traffic Out
752.463 kbit/s
WAN Total Traffic
Download
76.206 gigabytes
0
76.206 gigaby...
Upload
19.787 gigabytes
Sensor: (005) ix0 Traffic (Live Graph, 1 hour)
Firewall / 10.0.0.1 (avatar) [Linux/Unix]
Firewall A
(005) ix0 Traffic
Traffic In
4.689.245 kbit/s
Traffic Out
349.733 kbit/s
CPU Load
Total
79 %
SSH Script
States
140.074 #
State Searches
1.357.049 search/s
Max: 172.838 #
Min: 92.227 #
Sensor: (001) ix0 Traffic (Live Graph, 1 hour)
Firewall / 10.0.0.5 (ragnarok) [Linux/Unix]
Firewall B
(001) ix0 Traffic
Traffic In
5.170.631 kbit/s
Traffic Out
342.257 kbit/s
CPU Load
Total
88 %
SSH Script
States
153.336 #
State Searches
1.524.634 search/s
Max: 173.689 #
Min: 94.423 #
Sensor: WAN (2 days)
Firewall / 10.0.0.1 (avatar) [Linux/Unix]
Max: 9.711.464 kbit/s
Min: 1.614.626 kbit/s
Traffic Out (kbit/s)
Traffic In (kbit/s)


NOC
Zoom Out
Last 3 hours
Refresh every 10s
Firewall WAN Throughput
Last 48 hours
13 Gbps
10 Gbps
8 Gbps
5 Gbps
3 Gbps
0 bps
-2 Gbps
10/14 16:00 10/14 20:00 10/15 00:00 10/15 04:00 10/15 08:00 10/15 12:00 10/15 16:00 10/15 20:00 10/16 00:00 10/16 04:00 10/16 08:00 10/16 12:00
Download Max: 10.32 Gbps Avg: 5.02 Gbps Current: 844 Mbps
Upload Max: 1.11 Gbps Avg: 440 Mbps Current: 49 Mbps
Balance
1.0
0.5
0
-0.5
-1.0
Download
94.78 TiB
Upload
8.38 TiB
DHCP Leases
5713
Connected devices
2575
Download hastighed
844 Mbps
Avatar LoadAvg
0.39
Ragnarok LoadAvg
0.65
Ping offline
5
Firewall Balance
4.0 Gbps
3.0 Gbps
2.0 Gbps
1.0 Gbps
0 bps
-1.0 Gbps
11:30 12:00 12:30 13:00 13:30 14:00
DL 10.0.0.1 Max: 1.521 Gbps Avg: 790 Mbps Current: 302 Mbps
DL 10.0.0.5 Max: 2.944 Gbps Avg: 1.457 Gbps Current: 542 Mbps
UL 10.0.0.1 Max: 254 Mbps Avg: 78 Mbps Current: 19 Mbps
UL 10.0.0.5 Max: 259 Mbps Avg: 141 Mbps Current: 30 Mbps
Packets
300 kpps
200 kpps
100 kpps
0 pps
-100 kpps
-200 kpps
11:30 12:00 12:30 13:00 13:30 14:00
Out 10.0.0.1 Max: 159 kpps Avg: 91 kpps Current: 33 kpps
Out 10.0.0.5 Max: 288 kpps Avg: 156 kpps Current: 54 kpps
In 10.0.0.1 Max: 96 kpps Avg: 60 kpps Current: 16 kpps
In 10.0.0.5 Max: 144 kpps Avg: 98 kpps Current: 35 kpps
Varmeste switches
Last 12 hours
Temperature
49 °C
48 °C
47 °C
46 °C
45 °C
44 °C
43 °C
42 °C
04:00 08:00 12:00
current
d3.dist.npf 46 °C
d2.dist.npf 45 °C
d9.dist.npf 44 °C
d14.dist.npf 44 °C
d13.dist.npf 44 °C
d10.dist.npf 44 °C
e2.dist.npf 43 °C
e1.dist.npf 43 °C
d1.dist.npf 43 °C

NPF

VISMA

# Kube-router in NPF

Key learnings

- Think in terms of ToR (top-of-rack) design
- Often it’s not the new fancy kubernetes network that fails
- Kube-router toolbox is essential
  - You just kubectl exec -it into the kube-router box

Why kube-router

- We tried calico and got overwhelmed
- Very low overhead, must of it is done in-kernel
- Minimal code base, easily understood
- We run full kube-router, so no kubenet or kubeproxy on the nodes

NPF
VISMA

# DHCP Server Troubles

Modern DHCP server Kea failed, so we had to go with isc-dhcpd

ISC DHCPd needs to know the IP of all availability peers on startup

- ISC DHCPd can take a hostname rather than the IP
- Headless Services expose the pod ip to kube-dns
- Kubernetes DNS resolution to the rescue!
- Stateful Sets could be a possible non-hacky solution, but not sure ISC DHCPd will play nice with stateful sets.

NPF

VISMA

So how did it go?

thank you!

e-conomic.com/joinus

npf.dk/om-npf-crew/

NPF

VISMA