



# DNS Anycast Stack Virtual vs Bare Metal

Denic eG  
Dr. Benjamin Schönbach

Team DNS

Jens Finkhäuser  
Patrick Fedick  
Jürgen Geinitz  
Sven Wefels  
Marco Prause  
Frank Ulbricht  
Riyad Islam  
Manuel Stausberg  
Daniel Brandtner



# Anycast Stack (AS)

## AS

*„... the basic runtime environment of a nameserver (NS) and its (sub)processes upon OS-level including observer and other subsystems (i.e monitoring, networking).“*

## Virtual

- Multiple NS processes within isolated **namespaces** (aka containers)
  - isolated path resolution, PID, UID and network stack
  - restricted compute resources

## Bare Metal

- Multiple NS processes within isolated **chroot(s)** (aka filesystem)
  - isolated path resolution only
  - shared compute resources



# Requirements & Motivation

## Why stacking?

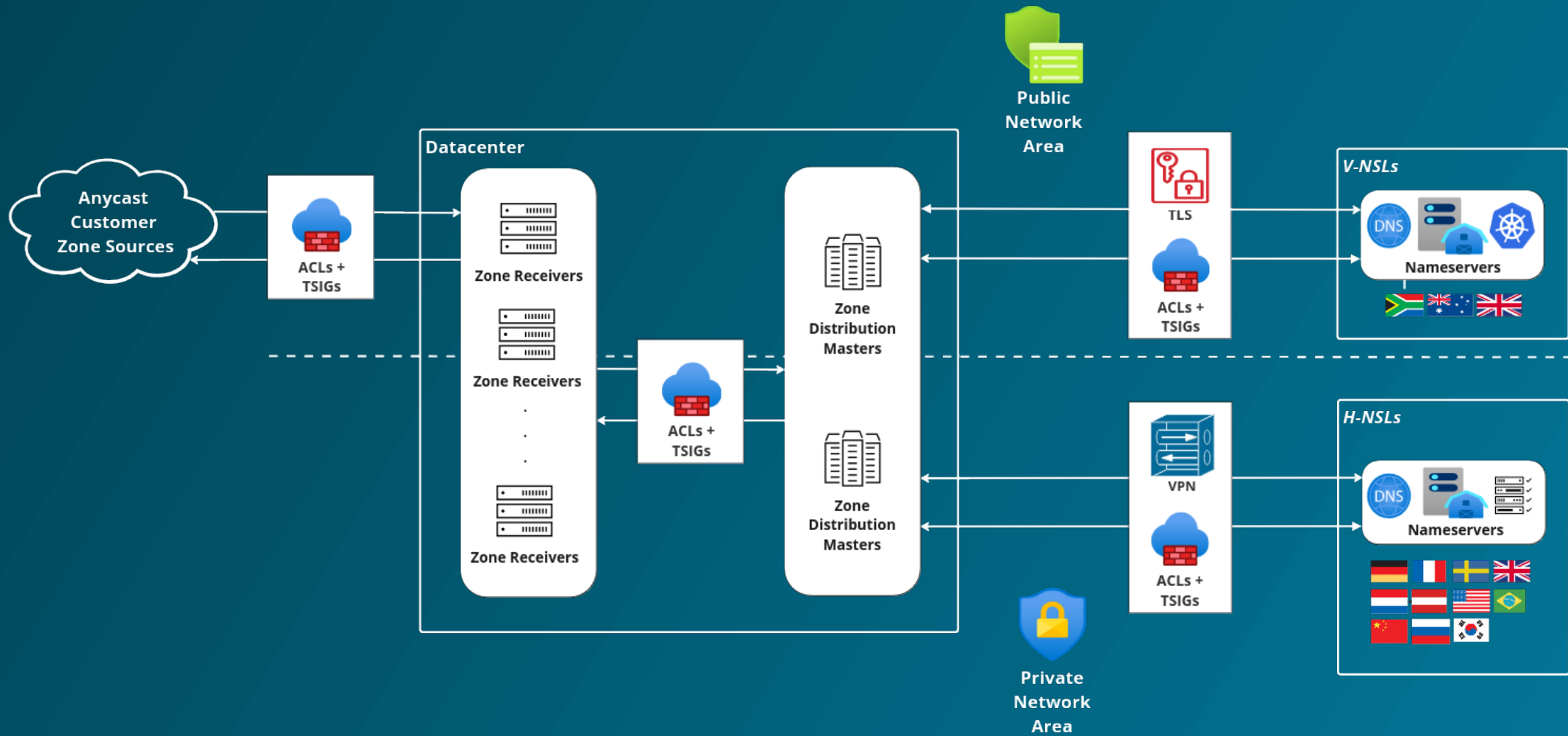
- Cloud computing era leads to more flexibility, less system administration but also managed restriction by providers
- Node redundancy vs. DNS traffic vs. Marketing → more nodes with complex routing and high customer awareness
- Increasing geopolitical crises impacting global network and node infrastructure
- Data science & AI with BIG hunger on DNS related data

## Whats needed?

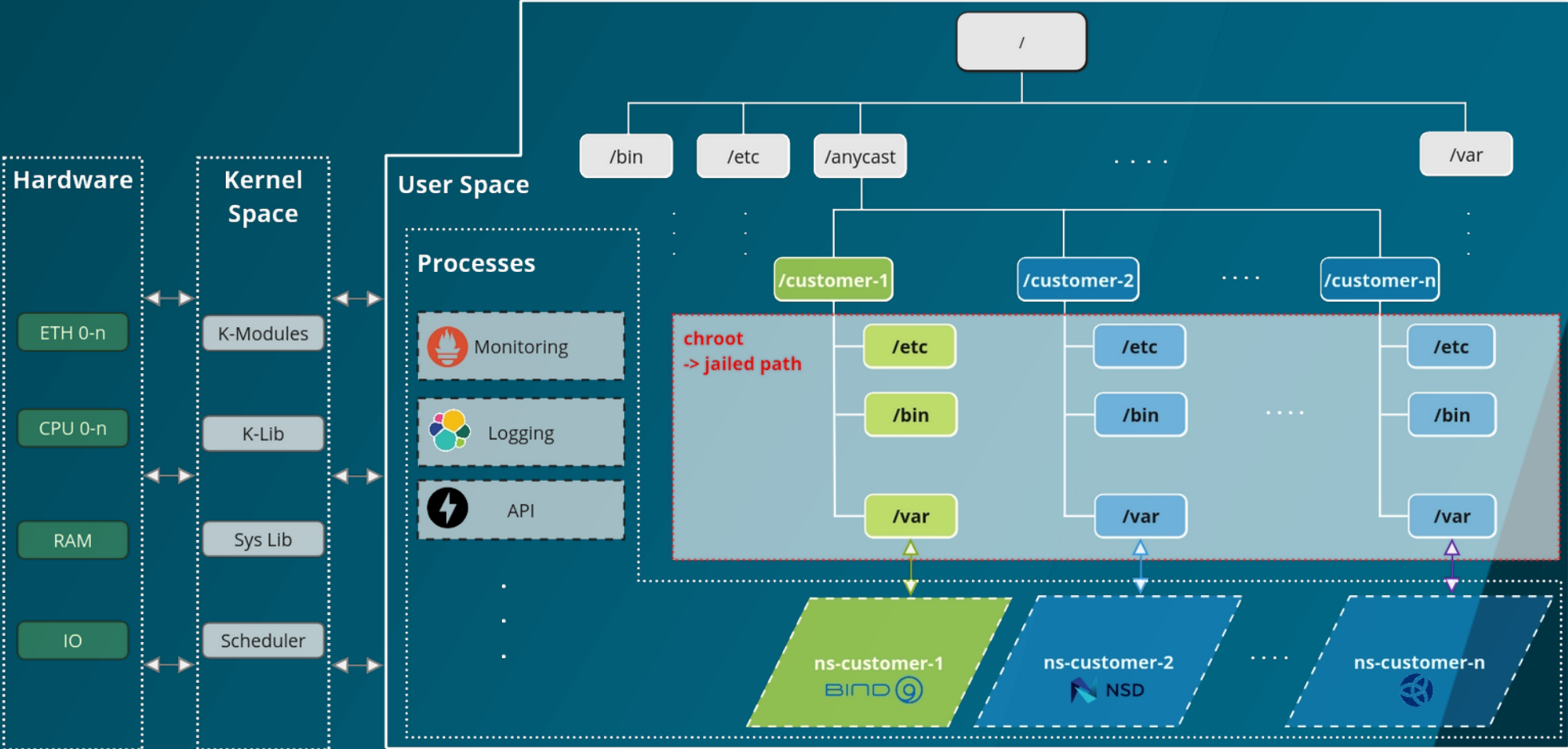
- Easy & fast setup
  - one click install & teardown
- Flex routing
  - many IXP connects
  - hard vs. soft routing
- Adaptive platform
  - on-prem, cloud, hybrid
- Multi-Tenancy
  - traffic isolation
  - high scalability



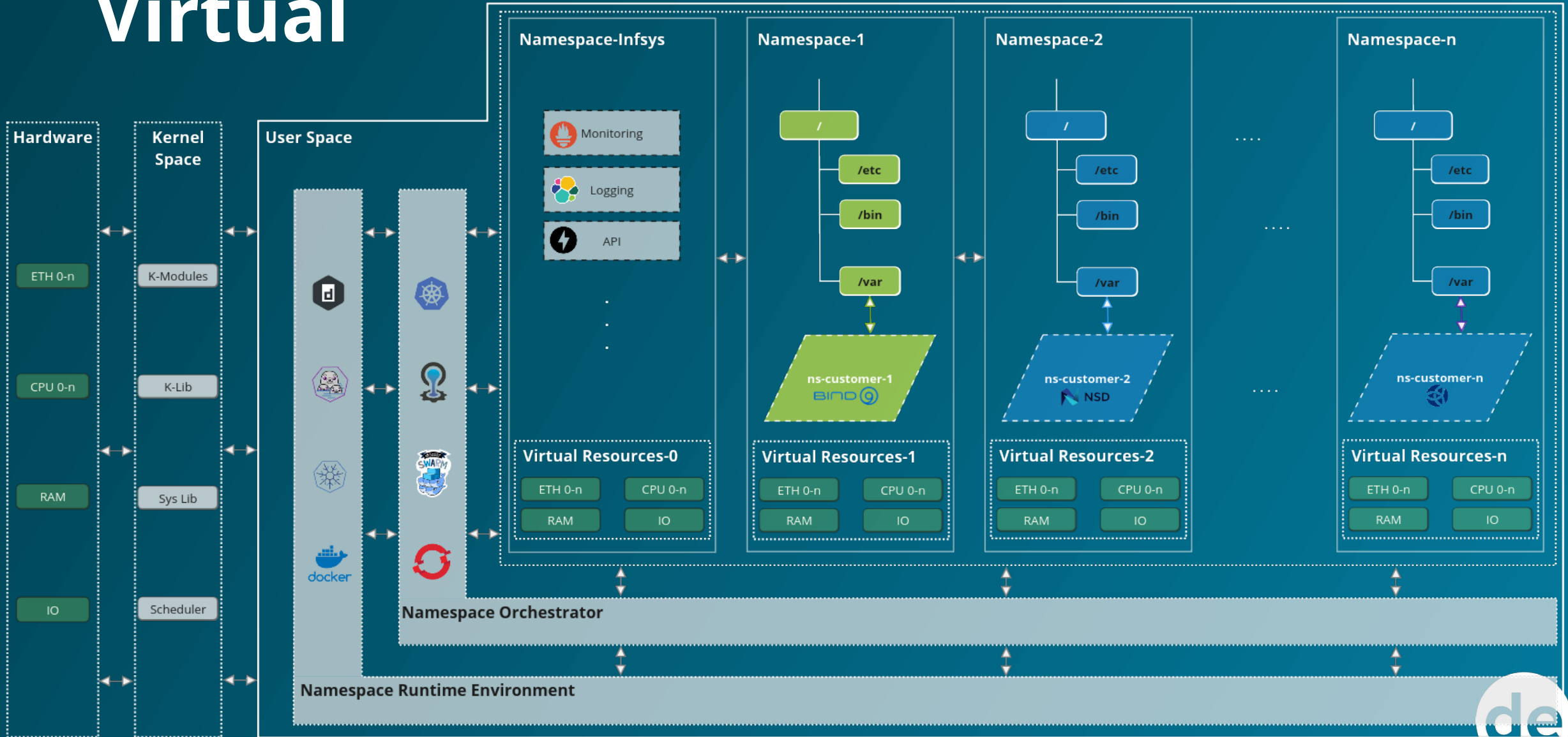
# Big Picture



# Bare Metal



# Virtual



# Demo Time :)

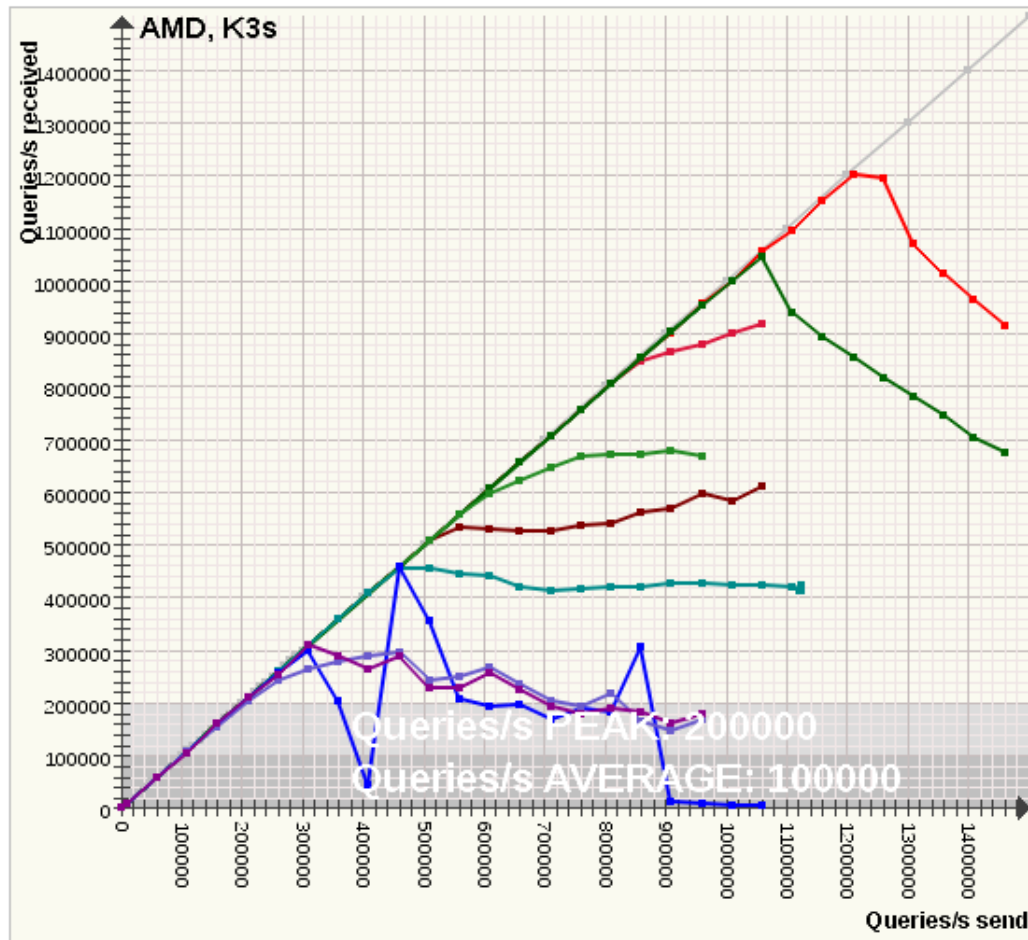
Lets setup a virtual anycast stack in less than 5 minutes with.....

- Orchestrator
  - k3s
- Dual network stack (ipv4/6)
  - calico (bird)
- 3 customers
  - de, nic ,enum

```
root@hv-5:~# kubectl get pods -A
NAMESPACE   NAME                                                                 READY   STATUS    RESTARTS   AGE
kube-system  local-path-provisioner-6c86858495-f9l59  1/1     Running   0           57d
kube-system  metrics-server-54fd9b65b-f9dtb          1/1     Running   0           57d
kube-system  calico-node-q46gv                       1/1     Running   0           57d
kube-system  coredns-6799fbc5-h5xsw                   1/1     Running   0           57d
kube-system  calico-kube-controllers-7ddc4f45bc-hn2jt 1/1     Running   0           57d
monitoring   node-exporter-zhzjx                      1/1     Running   0           73m
monitoring   blackbox-exporter-75fc876cfc-zzdqr       1/1     Running   0           73m
monitoring   kube-state-metrics-59845c6ddb-qp57n      1/1     Running   0           73m
de           ac-c9df674b-wt9b8                       1/1     Running   0           73m
enum         ac-7c4fdb5b46-ddhn8                     1/1     Running   0           73m
monitoring   filebeat-h6m94                           1/1     Running   0           73m
nic          ac-bfc8cb4dd-pd2w2                      1/1     Running   0           73m
monitoring   prometheus-0                             1/1     Running   0           73m
monitoring   nginx-5dc4679fb8-xllgp                   1/1     Running   2 (73m ago) 73m
enum         ns.enum.test-tt3-5-87f49dbc7-5mx66       3/3     Running   0           73m
nic          ns.nic.test-tt3-5-95b9b5cdb-qzkfj       3/3     Running   0           73m
de           ns.de.test-tt3-5-7c89b7799b-jqr7r       3/3     Running   0           73m
```

# Virtual

## DNS Performance tests



### Best Results:

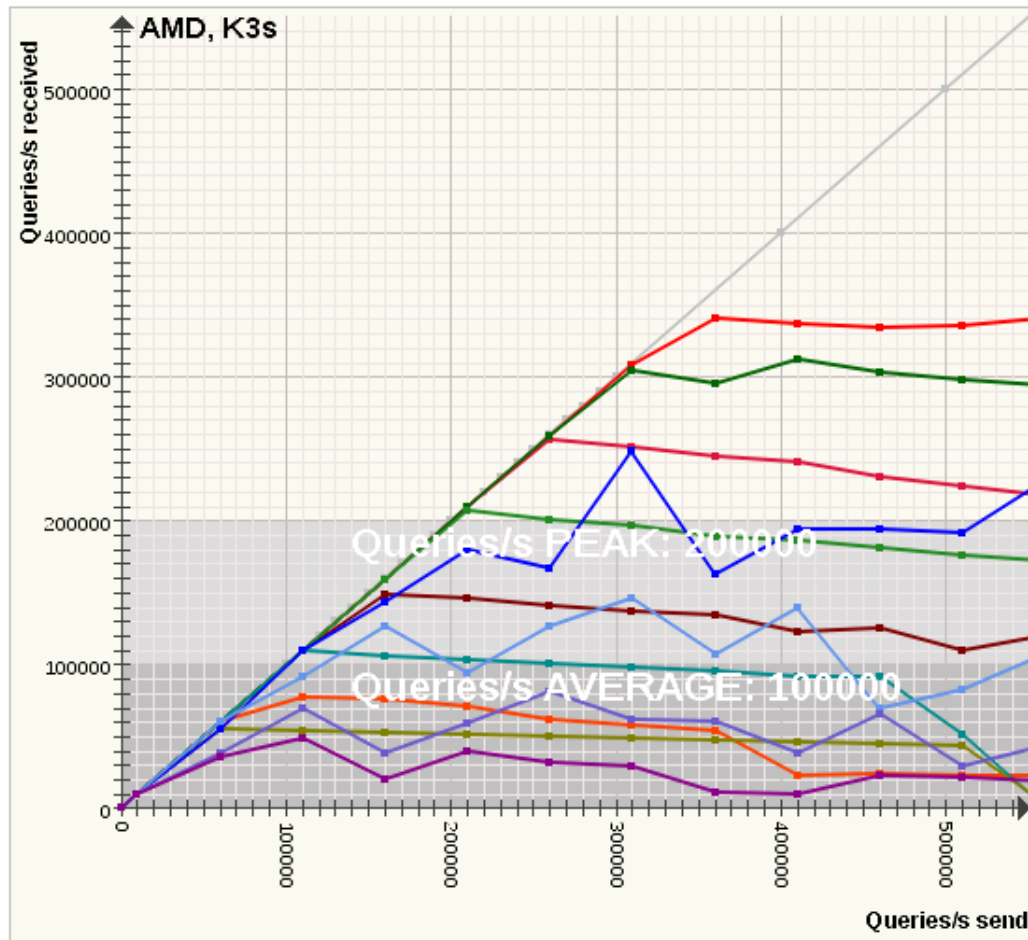
Threshold Packetloss: 1.0%





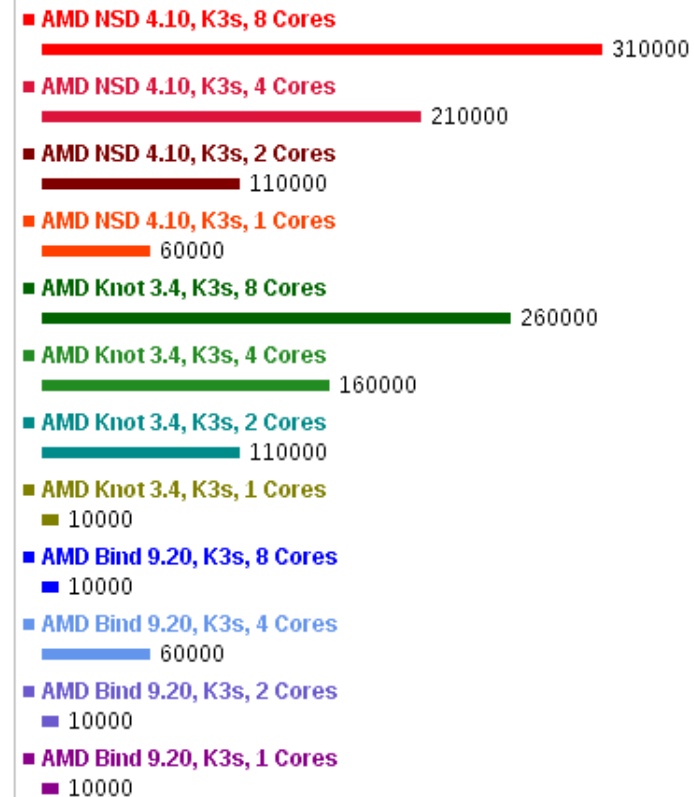
# Virtual

## DNS Performance tests



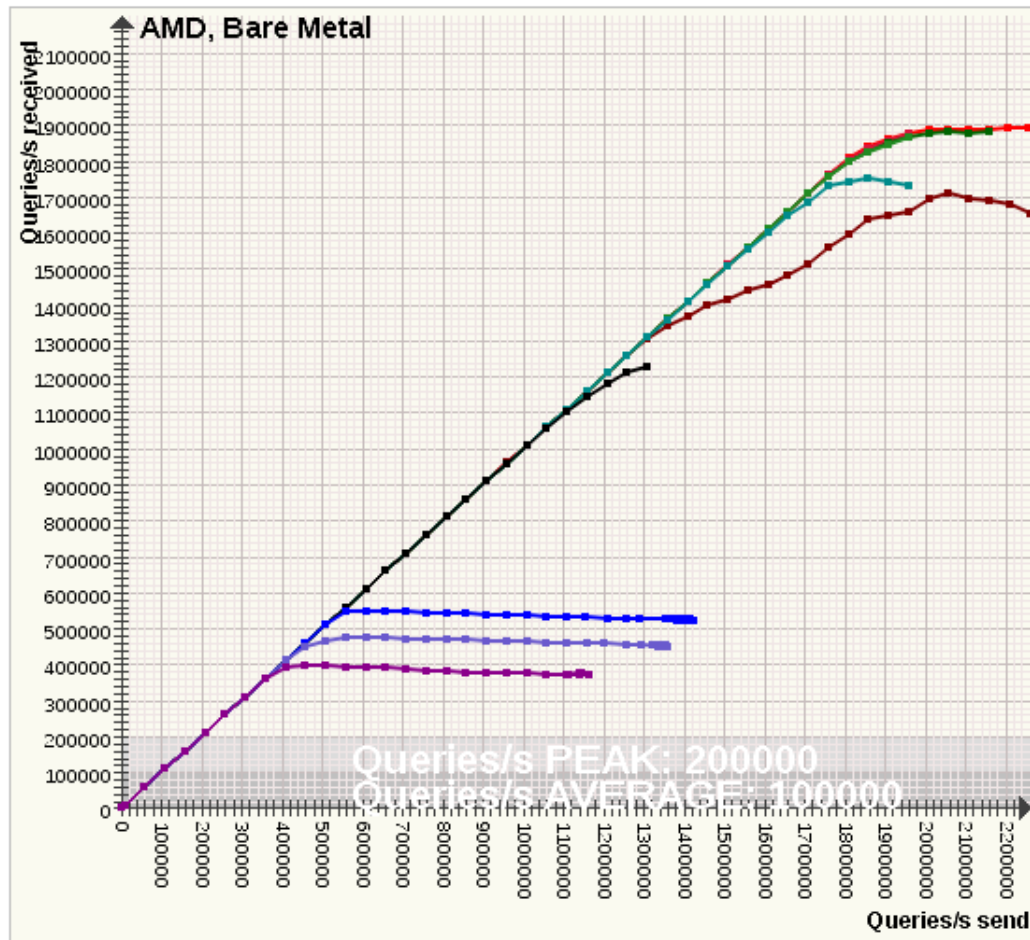
### Best Results:

Threshold Packetloss: 1.0%



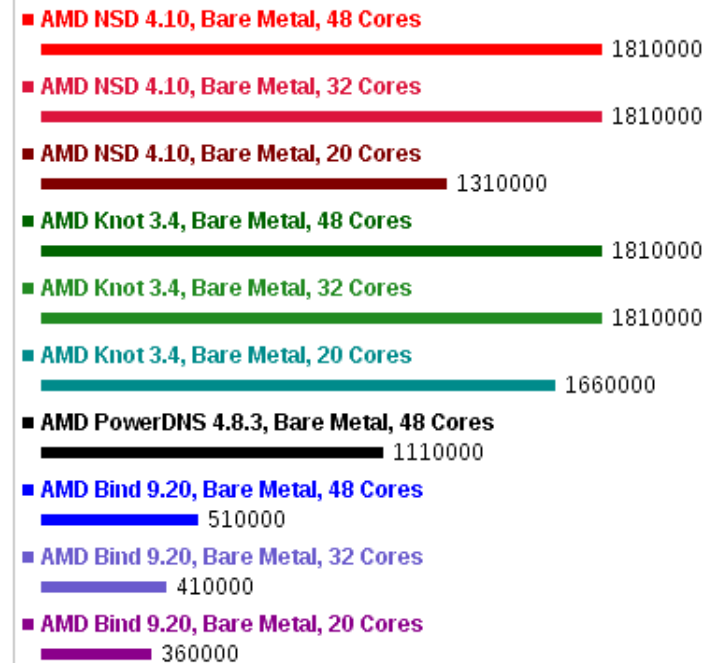
# Bare Metal

## DNS Performance tests



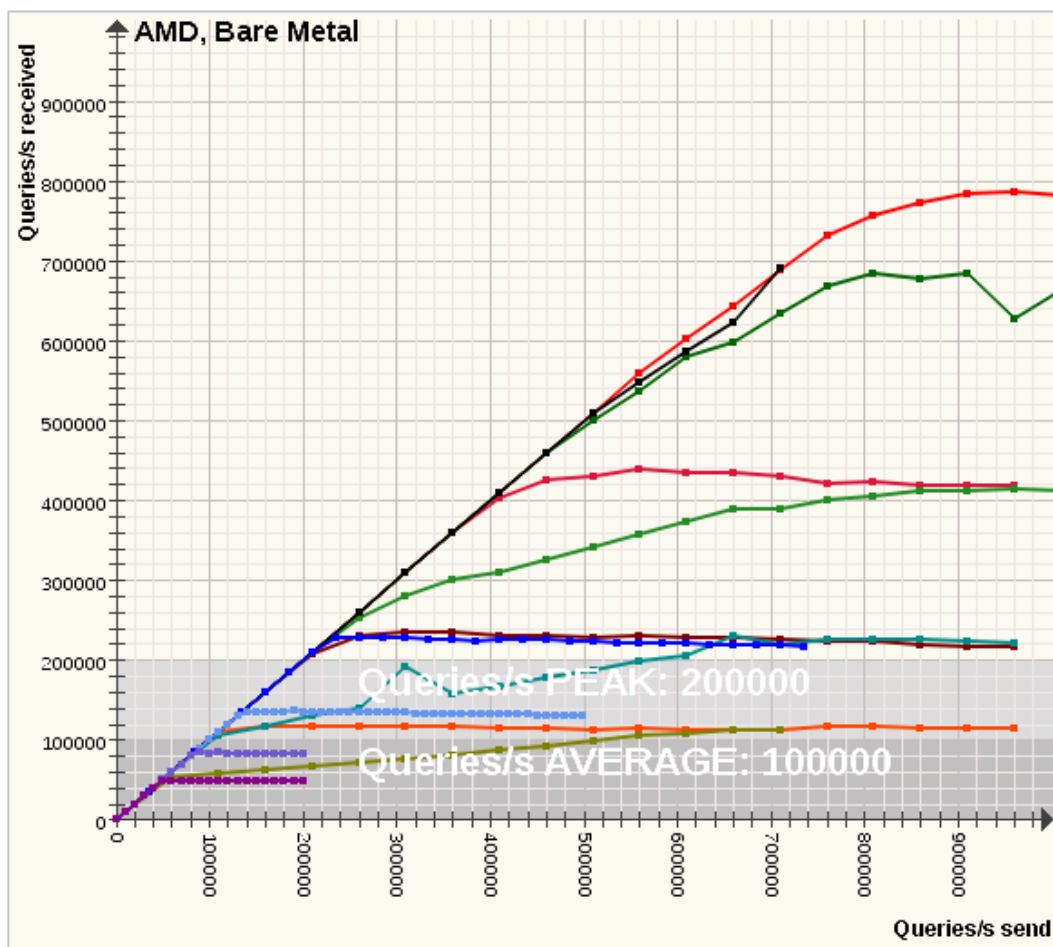
### Best Results:

Threshold Packetloss: 1.0%



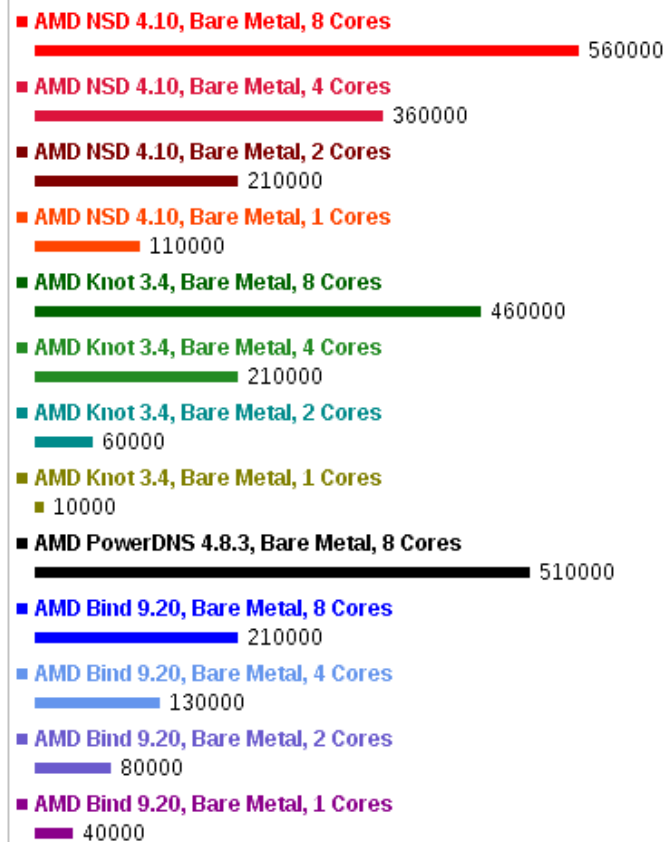
# Bare Metal

## DNS Performance tests



### Best Results:

Threshold Packetloss: 1.0%



# Conclusion

Which stack fits better?

- Trade off between....
  - query performance
  - administration effort
  - customer scalability
  - fast setup
  - OS portability

Capability	Stack	
	Bare Metal	Virtual
Performance	😊😊😊	😊
Administration	😊	😊😊😊
Scalability	😊😊😊	😊😊
Setup	😊😊😊	😊😊😊
Portability	😊	😊😊😊

Which stack to choose?

- High performance & scalability → Bare Metal
  - huge number of customers on same node
- High portability & less administration → Virtual
  - easy setup and strong OS adaptiveness

# Q&A





**THANK YOU!**

