No Packet Left Behind

AWS' journey to running its own hardware and software end-to-end across a global network

Lincoln Dale

Senior Principal Engineer
AWS – AS16509

Agenda

AWS Global Infrastructure

Our journey to reinventing our network infrastructure

- our hardware, software and how we put systems together

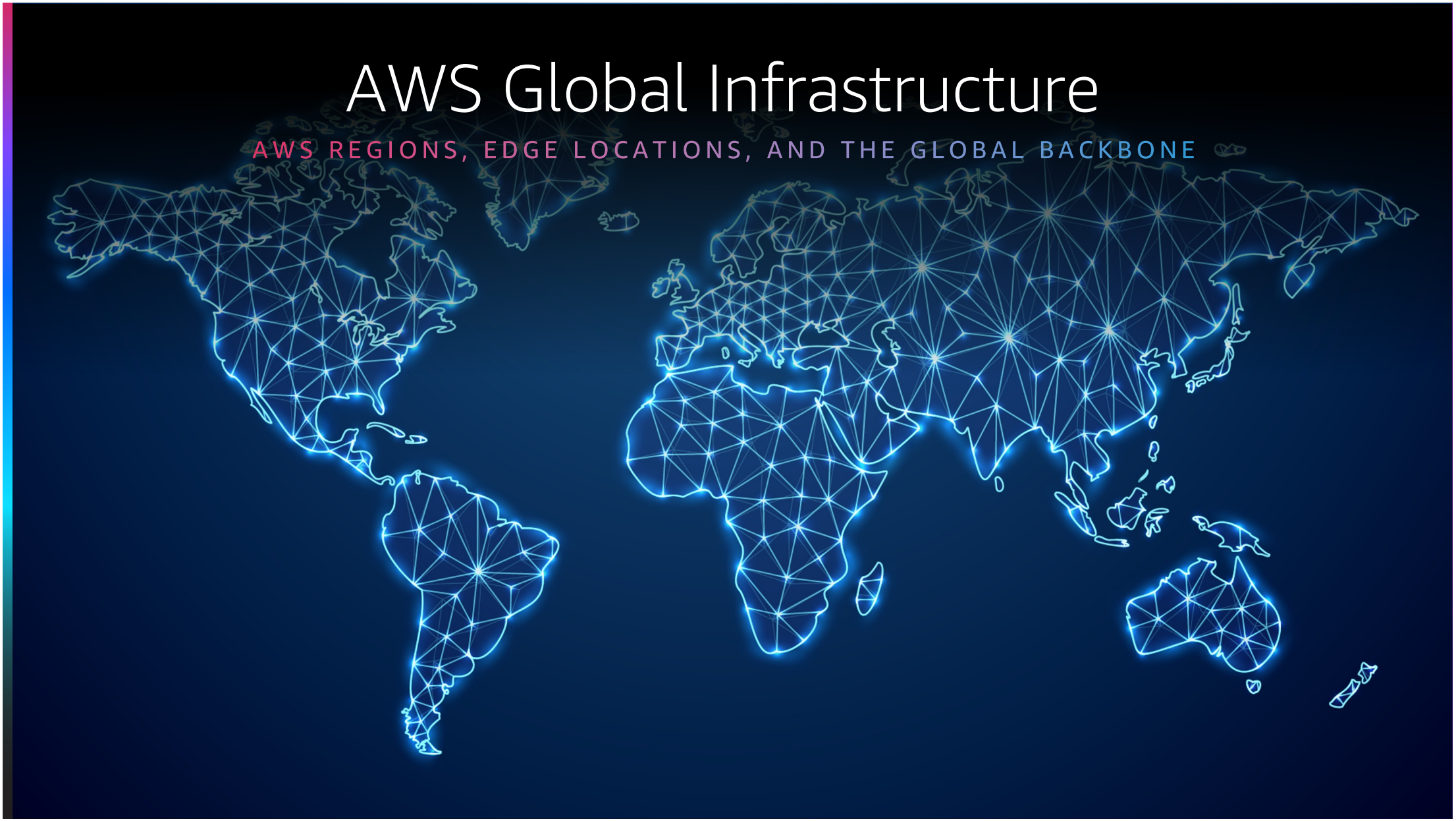
Network architecture and software, tools and controllers

How we build and automate our network

AWS Global Infrastructure

AWS Global Infrastructure

AWS REGIONS, EDGE LOCATIONS, AND THE GLOBAL BACKBONE



AWS Global Infrastructure

AWS REGIONS, EDGE LOCATIONS, AND THE GLOBAL BACKBONE

REGIONAL EXPANSION

- Available Today: 31 Regions
- Coming soon 5 Regions

AWS Global Infrastructure

AWS REGIONS, EDGE LOCATIONS, AND THE GLOBAL BACKBONE

LOCAL ZONES

- 4 recently launched
- 21 available today
- 30 coming soon

AWS Global Infrastructure

AWS REGIONS, EDGE LOCATIONS, AND THE GLOBAL BACKBONE

EDGE LOCATIONS

- 450+ CloudFront PoPs
- 115+ Direct Connect Locations

AWS Global Infrastructure

AWS REGIONS, EDGE LOCATIONS, AND THE GLOBAL BACKBONE

AWS Global Infrastructure

AWS REGIONS, EDGE LOCATIONS, AND THE GLOBAL BACKBONE

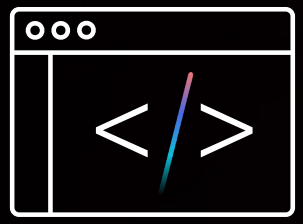
AWS NETWORK BACKBONE

- Redundant 400 Gbps links
- 245+ Countries & Territories
- Between all Regions, Local Zones, and Edge Locations

aws


NFL
prime video
THURSDAY
NIGHT
FOOTBALL

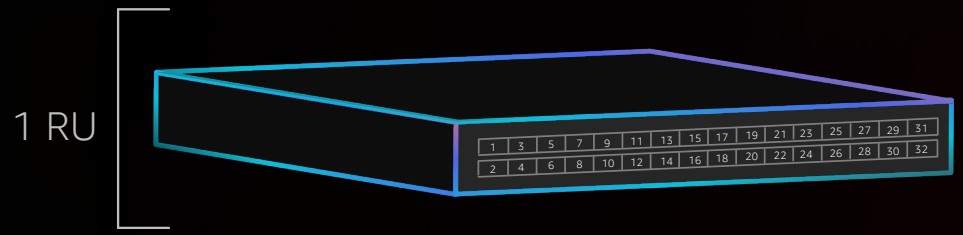
Reinventing our network infrastructure



CUSTOM SOFTWARE

CUSTOM HARDWARE

- Simplicity Scales
- Focus on the benefits
- Freedom to examine trade-offs



12.8

TERABITS PER SECOND

DEVICE: 1 x Switch

HEIGHT: 1 x Rack Unit (RU)

PORTS: 32 x 400G

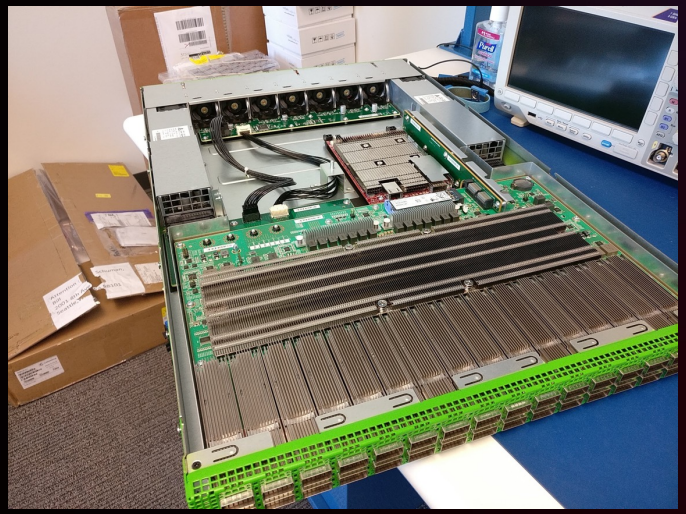
12.8

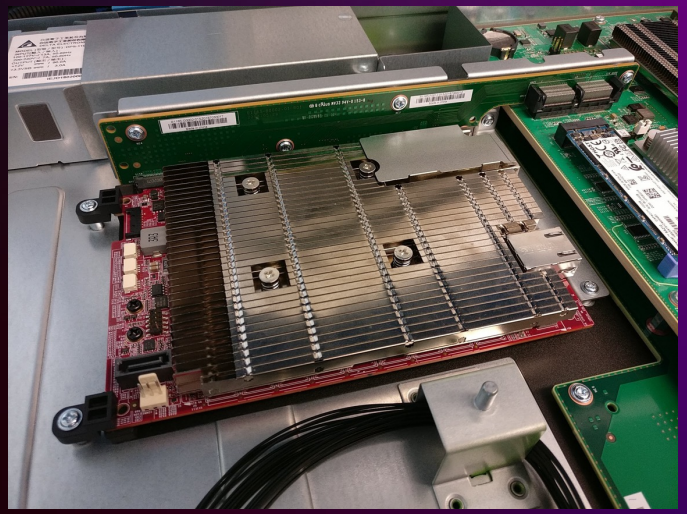
TERABITS PER SECOND

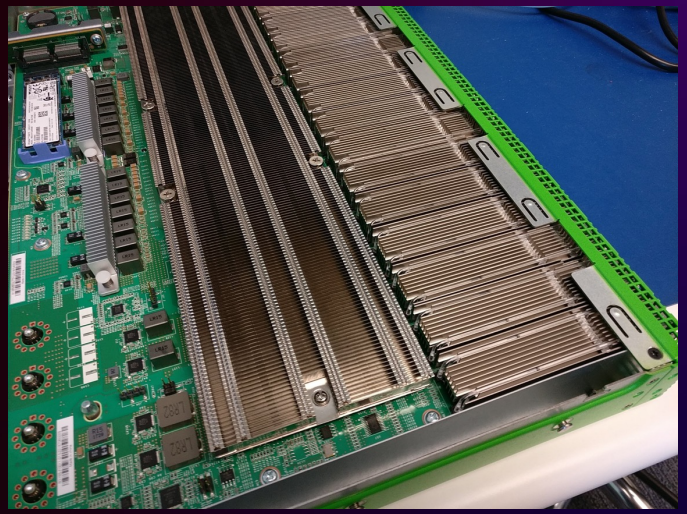
DEVICE: 1 x Switch

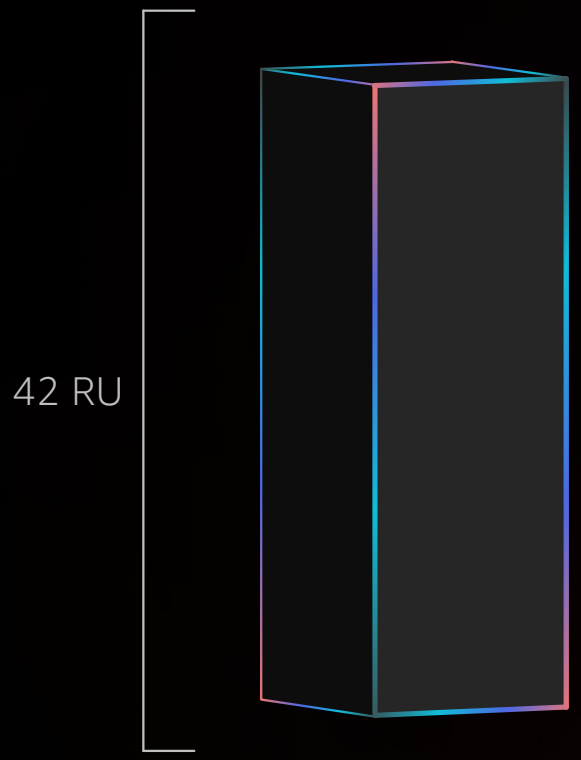
HEIGHT: 1 x Rack Unit (RU)

PORTS: 32 x 400G









100

TERABITS PER SECOND

DEVICE: 1 rack (32 x switches)

HEIGHT: 42 x Rack Unit (RU)

PORTS: 32 x 32 x 400G

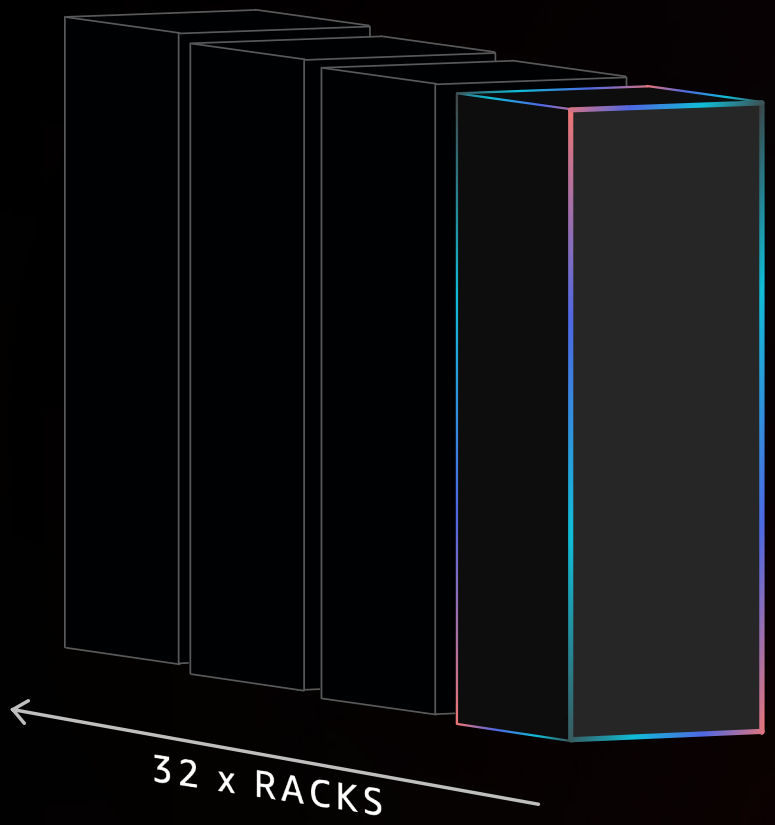
100

TERABITS PER SECOND

DEVICE: 1 rack (32 x switches)

HEIGHT: 42 x Rack Unit (RU)

PORTS: 32 x 32 x 400G



3,200

TERABITS PER SECOND

DEVICE: 32 racks (32 x switches)

HEIGHT: 42 x Rack Unit (RU)

THROUGHPUT/RACK: 100 Tbps

How we do it – In rack

Direct-attach copper (DAC) cabling

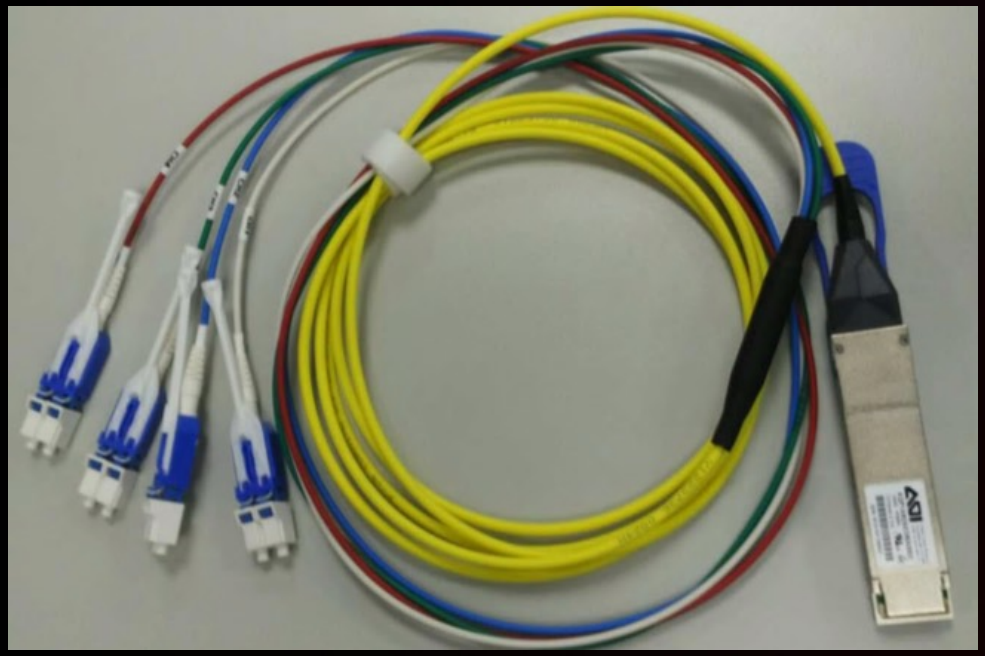
- 100G 6.7mm OD at 2.5m
- 400G 11mm OD at 2.5m

Our Biggest enemy? Cable diameter.

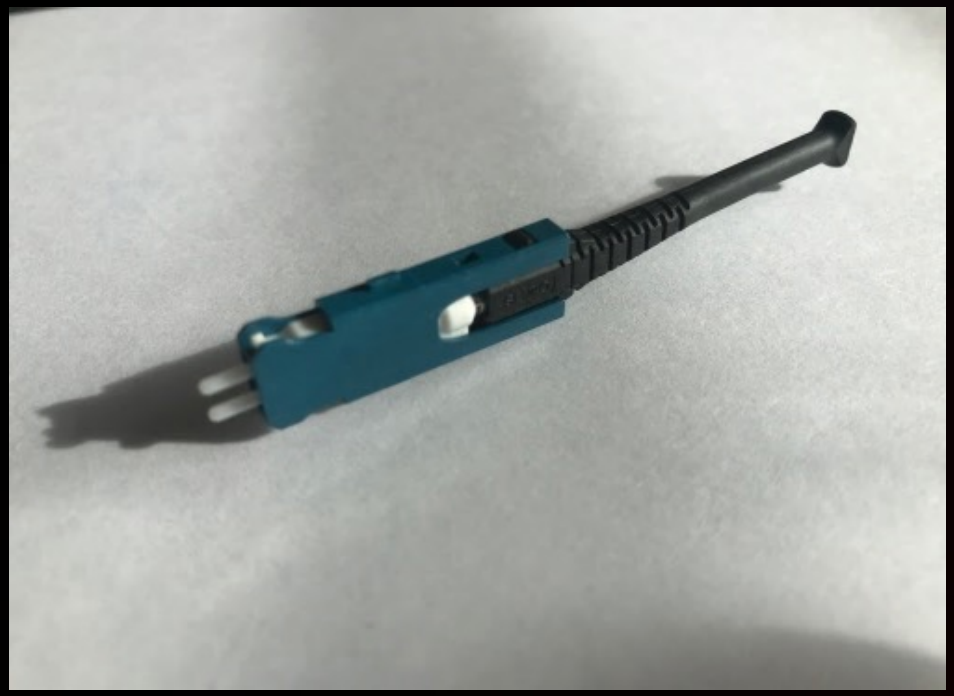
Active DAC with retimers to reduce cable area


POWER
ON/OFF
RESET
LUG TAG
BBU COVER
T1-R1-P27::T2-R11-P17
T1-R3-P27::T2-R11-P19
T1-R3-P25::T2-R9-P19
T1-R3-P23::T2-R7-P19
T1-R3-P17::T2-R1-P19
T1-R3-P19::T2-R3-P19
T1-R3-P31::T2-R

How we do it – Short reach



How we do it – SN connector



Metal boxes and a lot of cables

Small number of rack variations

Rack and cable switches for burn-in

Collect inventory and compare with bill of materials

Reprogram with AWS controlled binaries

Network Architecture and Software

Automate everything

Config generation

Deployment coordination

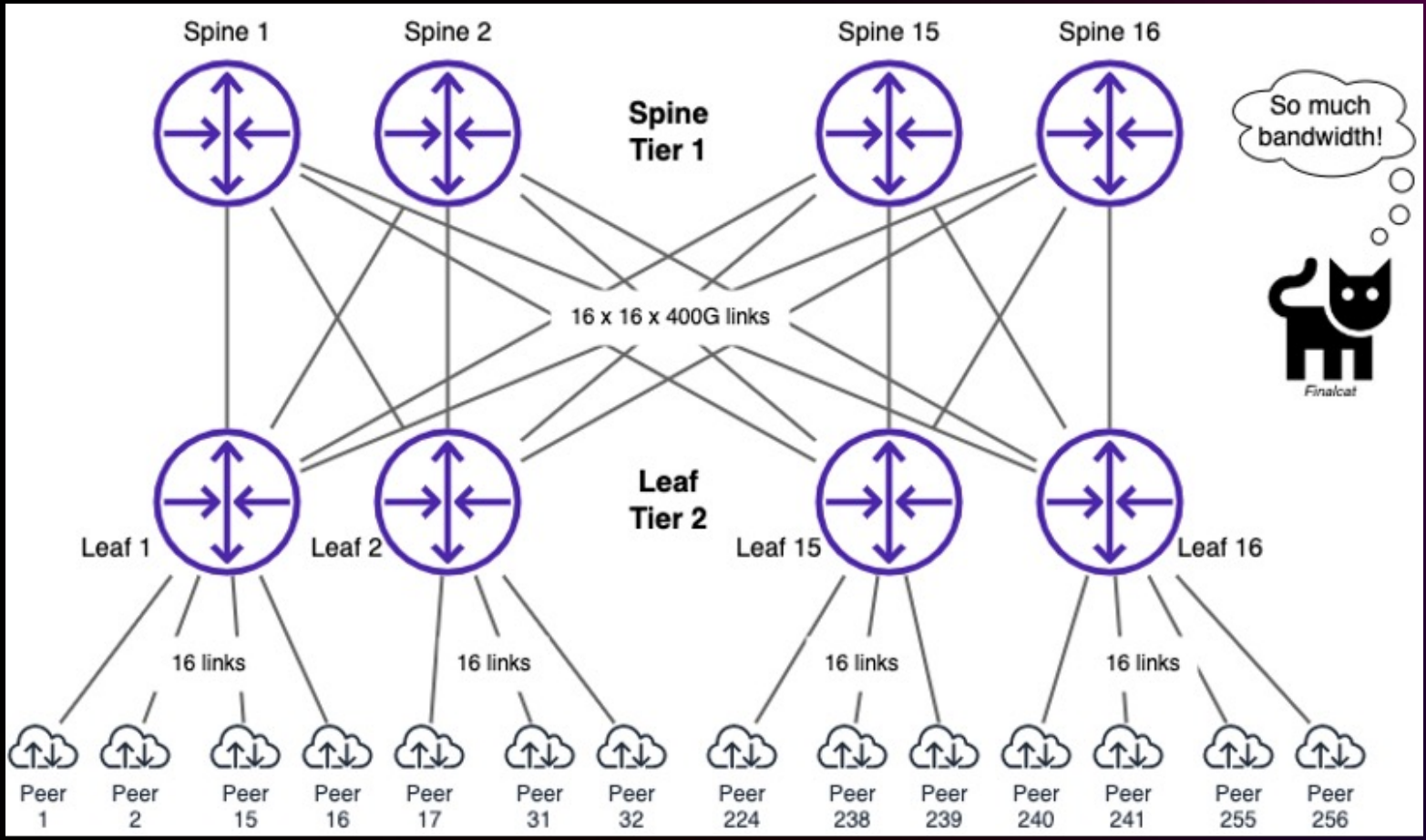
Active telemetry

Auto-remediation

NOC-less

2 tier Clos

NON OVERSUBSCRIBED ANY PORT TO ANY PORT



How we do it

MEDIUM HAUL

Data center interconnect (DCI)

OIF 400G ZR

400G – ZR+ to 400km,
Bright ZR over 1000km

Integrated routing, DWDM, encryption

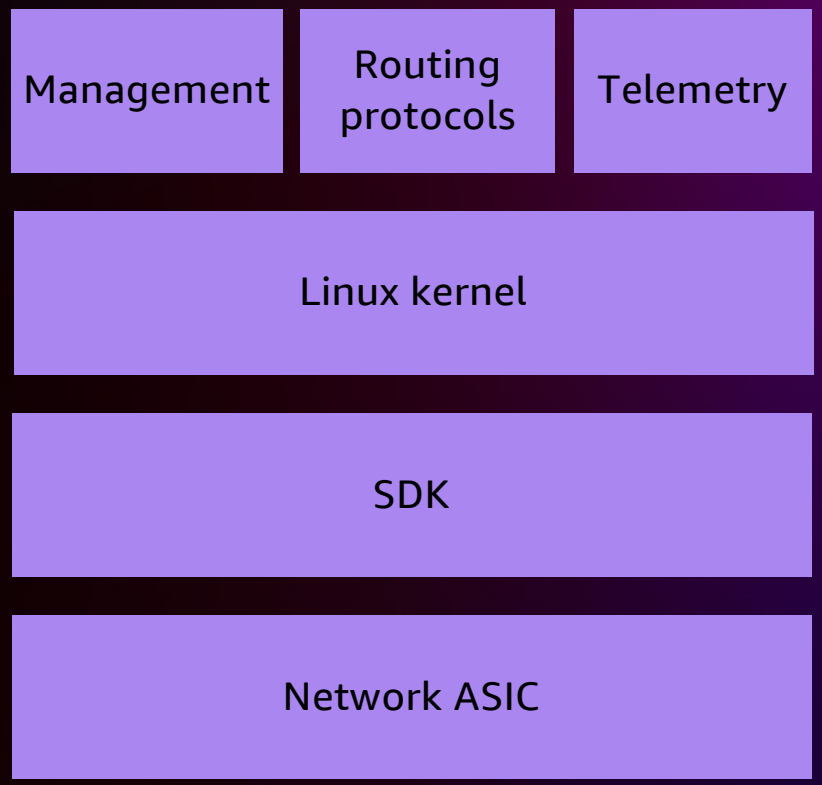
Create

NETWORK OPERATING SYSTEM

Linux-based

Multi-sourced manufacturing

Multi-ASIC



Create

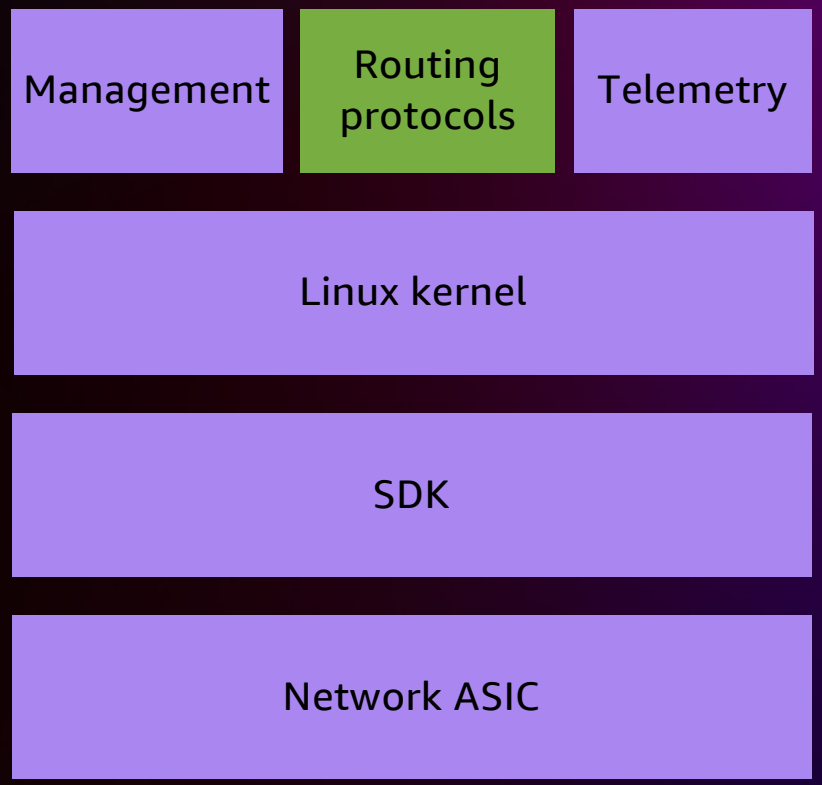
NETWORK OPERATING SYSTEM

Linux-based

Multi-sourced manufacturing

Multi-ASIC

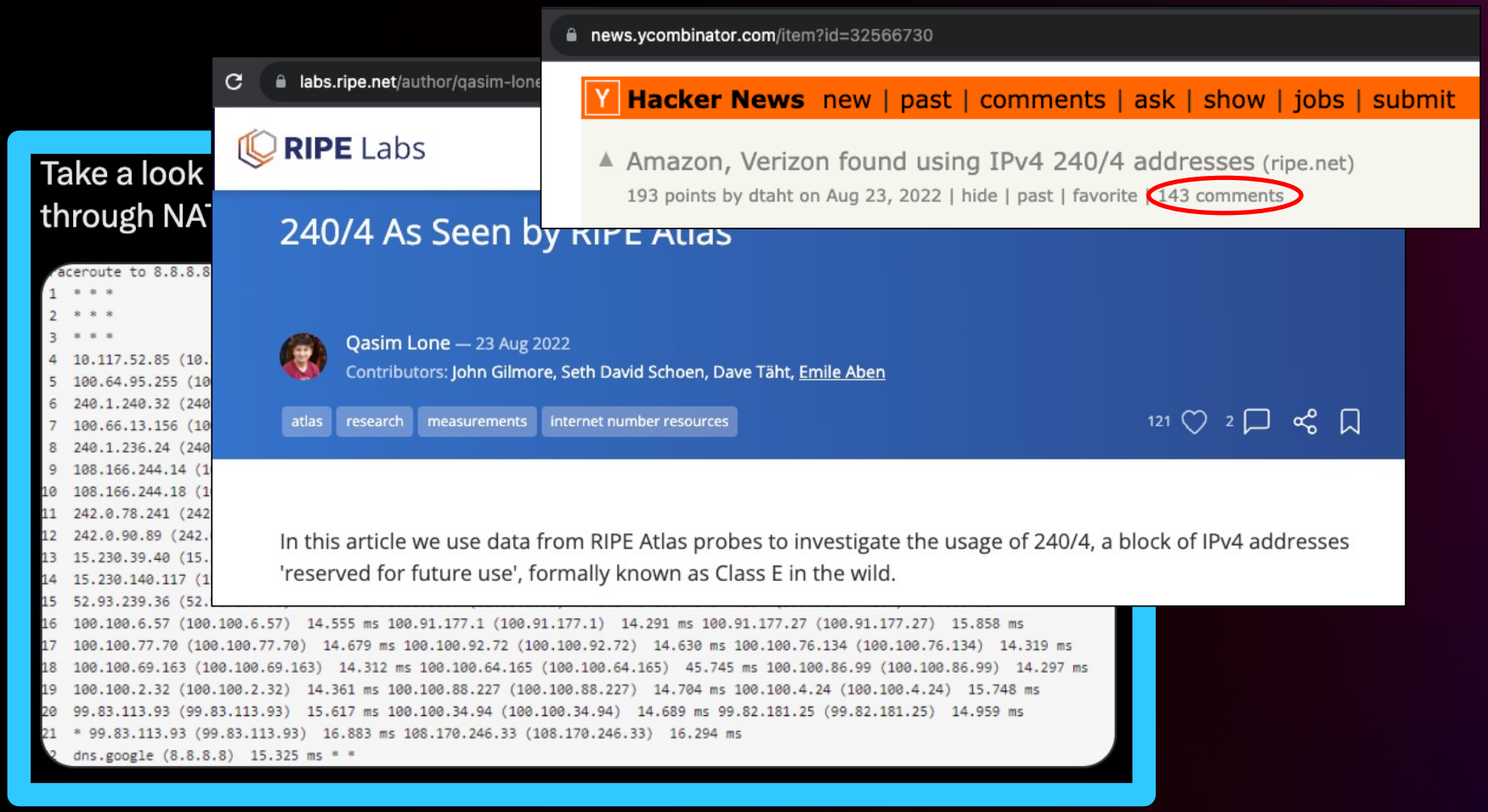
OSPF/BGP ++



Traceroute from outside AWS

```
% traceroute www.amazon.com
...
 5  * * *
 6  * * *
 7  52.93.33.77 (52.93.33.77)  1.984 ms 1.789 ms 1.983 ms
 8  52.93.33.130 (52.93.33.130)  2.316 ms 2.362 ms  2.891 ms
 9  150.222.72.105 (150.222.72.105)  3.682 ms  3.044 ms  3.002 ms
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  server-65-8-32-17.mel50.r.cloudfront.net (65.8.32.17)  3.650 ms  4.866 ms  3.033 ms
```

Traceroute from inside AWS



Traceroute from inside AWS (2)

EC2 INSTANCE IN MELBOURNE TRACEROUTE TO ELASTIC IP OF EC2 INSTANCE IN SYDNEY

```
% traceroute -n -q1 3.24.0.0
traceroute to 3.24.0.0 (3.24.0.0), 30 hops max, 60 byte packets
 1  244.5.0.1  1.647 ms
 2  240.1.72.6  0.185 ms
 3  240.1.192.13  10.690 ms
 4  15.230.210.36  13.923 ms
 5  15.230.210.45  19.009 ms
 6  15.230.210.96  10.888 ms
 7  15.230.211.4  11.825 ms
 8  240.1.184.15  11.448 ms
 9  240.1.184.30  11.345 ms
10  242.4.106.53  16.677 ms
11  3.24.0.0  11.696 ms
%
```

Disaggregated control plane

COMBINATION OF ON-DEVICE AND OFF-DEVICE

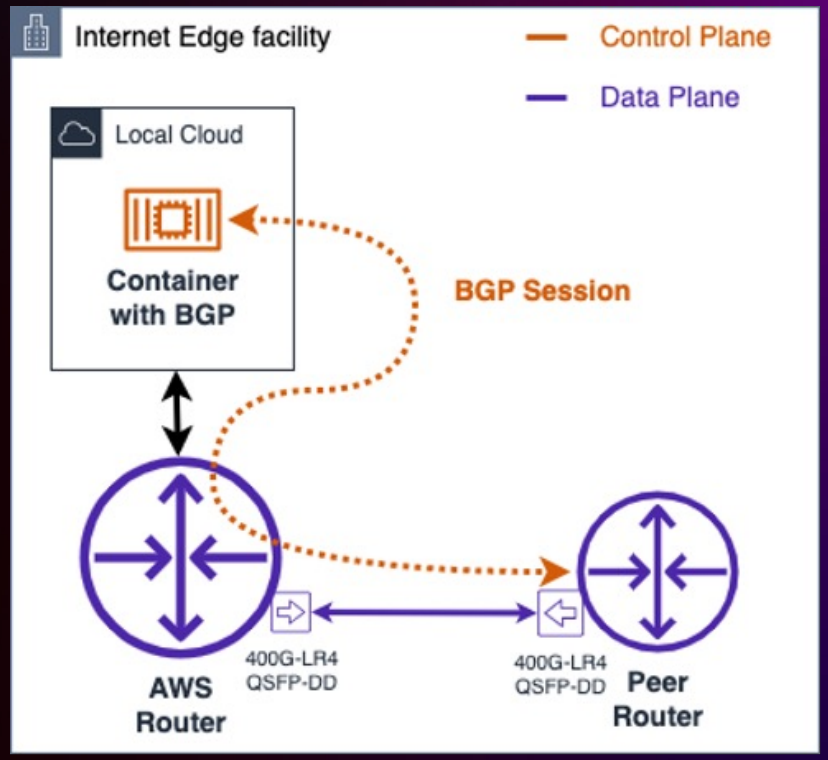
On-device handles local things like LACP, ARP/ND and all aspects of physical connectivity

BGP speaker runs elsewhere

Faster convergence and higher scale than would otherwise be possible

Enables us to iterate/evolve each part separately

Peer doesn't see anything different, TTL1 or TTL255 BGP still works the same way



How you see us / how we see you

COMBINATION OF ON-DEVICE AND OFF-DEVICE

```
re0> ping 189.233.232.1 source 189.233.232.0 size 1472 do-not-fragment
PING 189.233.232.1 (189.233.232.1) from 189.233.232.0 : 1472(1500) bytes of data.
1480 bytes from 189.233.232.1: icmp_seq=1 ttl=255 time=0.901 ms
1480 bytes from 189.233.232.1: icmp_seq=1 ttl=254 time=0.919 ms (DUP!)
1480 bytes from 189.233.232.1: icmp_seq=2 ttl=255 time=0.555 ms
1480 bytes from 189.233.232.1: icmp_seq=2 ttl=254 time=0.614 ms (DUP!)
```

```
% ip -d link show bond1
166: bond1: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 1500 qdisc noqueue master TMS-VRF
state UP mode DEFAULT group default qlen 1000
    link/ether a0:d0:dc:94:52:42 brd ff:ff:ff:ff:ff:ff promiscuity 0 minmtu 68 maxmtu 65535
    bond mode 802.3ad miimon 100 updelay 5000 downdelay 1100 peer_notify_delay 0 use_carrier 1
arp_interval 0 arp_validate none arp_all_targets any primary_reselect always fail_over_mac none
xmit_hash_policy layer2 resend_igmp 1 num_grat_arp 1 all_slaves_active 1 min_links 1
lp_interval 1 packets_per_slave 1 lacp_rate fast ad_select count ad_aggregator 1 ad_num_ports 1
ad_actor_key 12345 ad_partner_key 12345 ad_partner_mac 94:ae:f0:c8:38:dd tlb_dynamic_lb 1
    vrf_slave table 14 addrgenmode eui64 numtxqueues 16 numrxqueues 16 gso_max_size 65536
gso_max_segs 65535
    alias _EXTPEER-LAG_ AUSSIE_BROADBAND #1 AS4764 PS_ID:96445 BGP:bne50-br-fnc-f1-b1-bgp-r2-c1
```

The curious case of flaky IPv6 NS

LINUX MCAST_RESOLICIT (NON-DEFAULT) REQUIRED FOR TO TRIGGER SRC+DST LINK-LOCAL IPV6 NS

src FE80 dst 2620 fails, src FE80 dst FF80 WORKS **TL;DR: Many people get FE80 ACLs wrong**

```
% ip -ts monitor neigh dev bond1
[2023-01-13T02:58:15.544747] 2620:107:4008:xxx::2 dev bond1 lladdr d4:6a:35:35:4c:92 router PROBE
[2023-01-13T02:58:15.649269] fe80::d66a:35ff:fe25:4c92 dev bond1 lladdr d4:6a:35:35:4c:92 router PROBE
[2023-01-13T02:58:15.650764] fe80::d66a:35ff:fe25:4c92 dev bond1 lladdr d4:6a:35:35:4c:92 router REACHABLE
[2023-01-13T02:58:45.852977] 99.83.1xx.xx dev bond1 lladdr d4:6a:35:35:4c:92 PROBE
[2023-01-13T02:58:45.854469] 99.83.1xx.xx dev bond1 lladdr d4:6a:35:35:4c:92 REACHABLE
[2023-01-13T02:58:46.112645] fe80::d66a:35ff:fe25:4c92 dev bond1 lladdr d4:6a:35:35:4c:92 router PROBE
[2023-01-13T02:58:46.114825] fe80::d66a:35ff:fe25:4c92 dev bond1 lladdr d4:6a:35:35:4c:92 router REACHABLE
[2023-01-13T02:58:52.413809] 2620:107:4008:xxx::2 dev bond1  router FAILED
[2023-01-13T02:59:07.779235] 2620:107:4008:xxx::2 dev bond1 lladdr d4:6a:35:35:4c:92 router REACHABLE
[2023-01-13T02:59:16.305279] 99.83.1xx.xx dev bond1 lladdr d4:6a:35:35:4c:92 PROBE
[2023-01-13T02:59:16.306371] 99.83.1xx.xx dev bond1 lladdr d4:6a:35:35:4c:92 REACHABLE
[2023-01-13T02:59:16.473164] 2620:107:4008:xxx::2 dev bond1 lladdr d4:6a:35:35:4c:92 router PROBE
[2023-01-13T02:59:16.570665] fe80::d66a:35ff:fe25:4c92 dev bond1 lladdr d4:6a:35:35:4c:92 router PROBE
[2023-01-13T02:59:16.574393] fe80::d66a:35ff:fe25:4c92 dev bond1 lladdr d4:6a:35:35:4c:92 router REACHABLE
[2023-01-13T02:59:46.767019] 99.83.1xx.xx dev bond1 lladdr d4:6a:35:35:4c:92 PROBE
[2023-01-13T02:59:46.770263] 99.83.1xx.xx dev bond1 lladdr d4:6a:35:35:4c:92 REACHABLE
[2023-01-13T02:59:47.025611] fe80::d66a:35ff:fe25:4c92 dev bond1 lladdr d4:6a:35:35:4c:92 router PROBE
[2023-01-13T02:59:47.026513] fe80::d66a:35ff:fe25:4c92 dev bond1 lladdr d4:6a:35:35:4c:92 router REACHABLE
[2023-01-13T02:59:53.341824] 2620:107:4008:xxx::2 dev bond1  router FAILED
[2023-01-13T03:00:07.779211] 2620:107:4008:xxx::2 dev bond1 lladdr d4:6a:35:35:4c:92 router REACHABLE
```

36 IPv6 NS sent, none answered

As soon as we age out the entry..

..it then answers

..rinse/repeat..

```
% tcpdump -i bond1 -n -p --direction=out 'icmp6'
04:15:57.597793 IP6 fe80::a2d0:dcff:fefc:8ed6 > 2620:107:4008:xxx::2: ICMP6, neigh solicitation, who has 2620:107:4008:xxx::2
04:16:02.717802 IP6 fe80::a2d0:dcff:fefc:8ed6 > 2620:107:4008:xxx::2: ICMP6, neigh solicitation, who has 2620:107:4008:xxx::2
04:16:07.837808 IP6 fe80::a2d0:dcff:fefc:8ed6 > 2620:107:4008:xxx::2: ICMP6, neigh solicitation, who has 2620:107:4008:xxx::2
04:16:10.407026 IP6 fe80::a2d0:dcff:fefc:8ed6 > fe80::d66a:35ff:fe35:4c92: ICMP6, neighbor advertisement, tgt is fe80::a2d0:d
04:16:12.957792 IP6 fe80::a2d0:dcff:fefc:8ed6 > ff02::1:ff00:2: ICMP6, neighbor solicitation, who has 2620:107:4008:xxx::2
```

Cisco handling of link-local IPv6 nexthops

NEIGH SOLICITATION ON UNRESOLVED V6 NEXTHOP (VERSUS NOT USING THE ONE THAT DID RESOLVE!)

```
# tcpdump on container terminating BGP, we see route announced to peer
...
21:20:10.427367 IP6 (flowlabel 0x7236d, hlim 1, next-header TCP (6) payload length: 286)
2620:107:XXXX:YYYY::1.41991 > 2620:107:XXXX:YYYY::2.179: Flags [P.], seq 1660:1926, ack 38, win 15745, length 266: BGP
    Update Message (2), length: 117
      Origin (1), length: 1, Flags [T]: IGP
        0x0000:  00
      AS Path (2), length: 6, Flags [T]: 16509
        0x0000:  0201 0000 407d
      Multi Exit Discriminator (4), length: 4, Flags [O]: 1000
        0x0000:  0000 03e8
 Multi-Protocol Reach NLRI (14), length: 44, Flags [OE]:
        AFI: IPv6 (2), SAFI: Unicast (1)
        nexthop: 2620:107:XXXX:YYYY::1, fe80::f040:4861, nh-length: 32, no SNPA
          2605:a7c0:12a::/48
          2605:a7c0:10a::/48
...
```

... because we provided a link-local nexthop

... alongside a unicast nexthop

```
# tcpdump on physical device terminating peering session
% sudo tcpdump -i bond13 -n -vve icmp6
tcpdump: listening on bond13, link-type EN10MB (Ethernet), capture size 262144 bytes
21:52:17.498480 94:ae:f0:c3:a0:d9 > 33:33:ff:40:48:61, ethertype IPv6 (0x86dd), length 86: (class 0xe0, hlim 255, next-header
ICMPv6 (58) payload length: 32) fe80::96ae:f0ff:fec3:a0d9 > ff02::1:ff40:4861: [icmp6 sum ok] ICMP6, neighbor solicitation,
length 32, who has fe80::f040:4861
      source link-address option (1), length 8 (1): 94:ae:f0:c3:a0:d9
...
```

Why is the peer sending us a NS for an address we don't have?

AWS Global Infrastructure

AWS REGIONS, EDGE LOCATIONS, AND THE GLOBAL BACKBONE

AWS NETWORK BACKBONE

- Redundant 400 Gbps links
- 245+ Countries & Territories
- Between all Regions, Local Zones, and Edge Locations

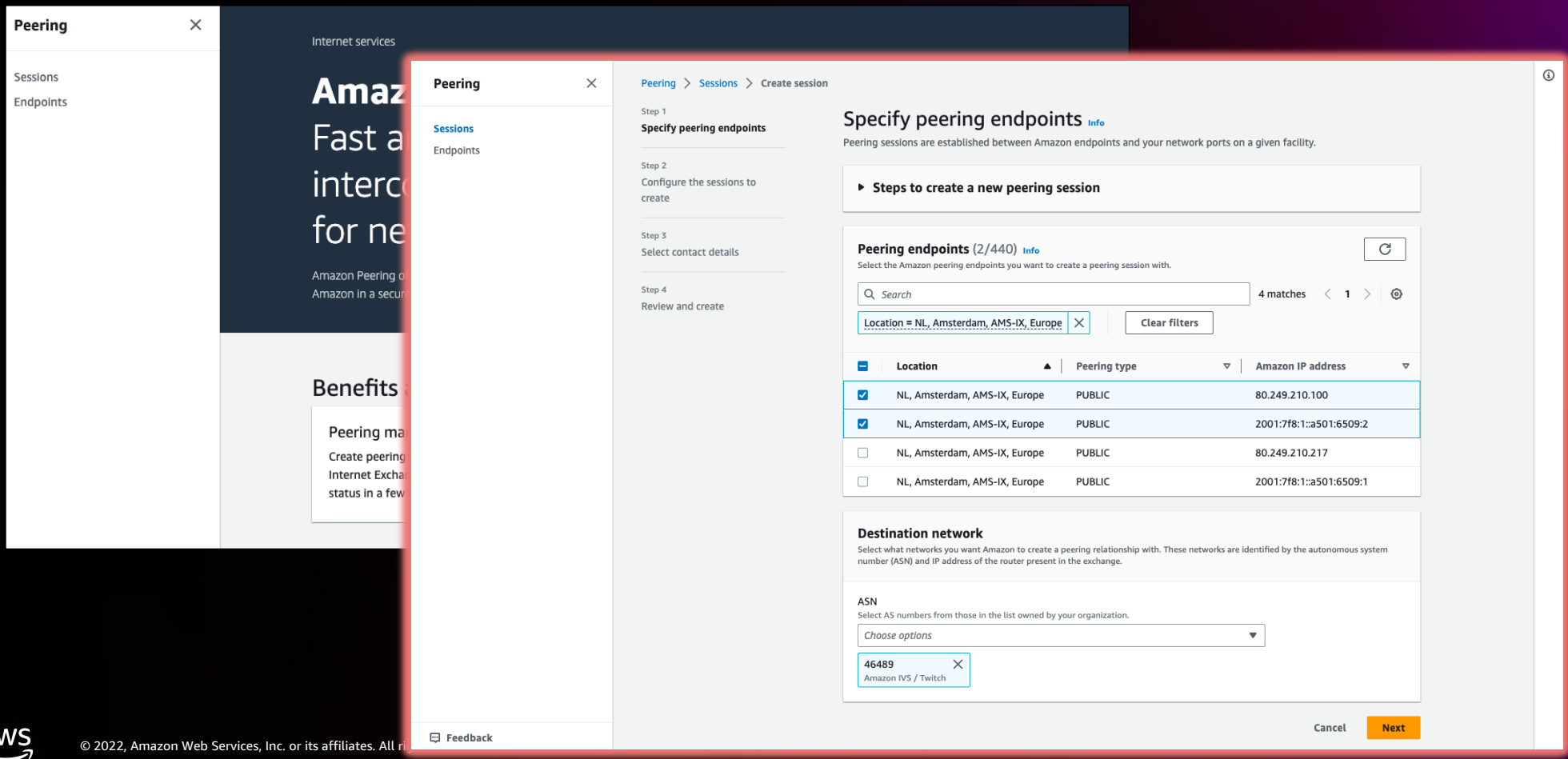
aws

http://ec2-reachability.amazonaws.com/

- Sydney ping/traceroute 3.24.0.0
- Melbourne ping/traceroute 16.50.164.255
- Perth ping/traceroute 96.0.1.218
- (see http://ec2-reachability.amazonaws.com/ for other IPv4/v6 locations)
- `curl https://blue.diagnostics.globalaccelerator.aws/api/stack`
  `{"ip":"13.248.102.33","stack":"MEL50-1"}`
- Under 2msec (ideally <1msec) RTT wired ethernet CPE on NBN FTTP -> you -> us

https://interconnect.amazon/

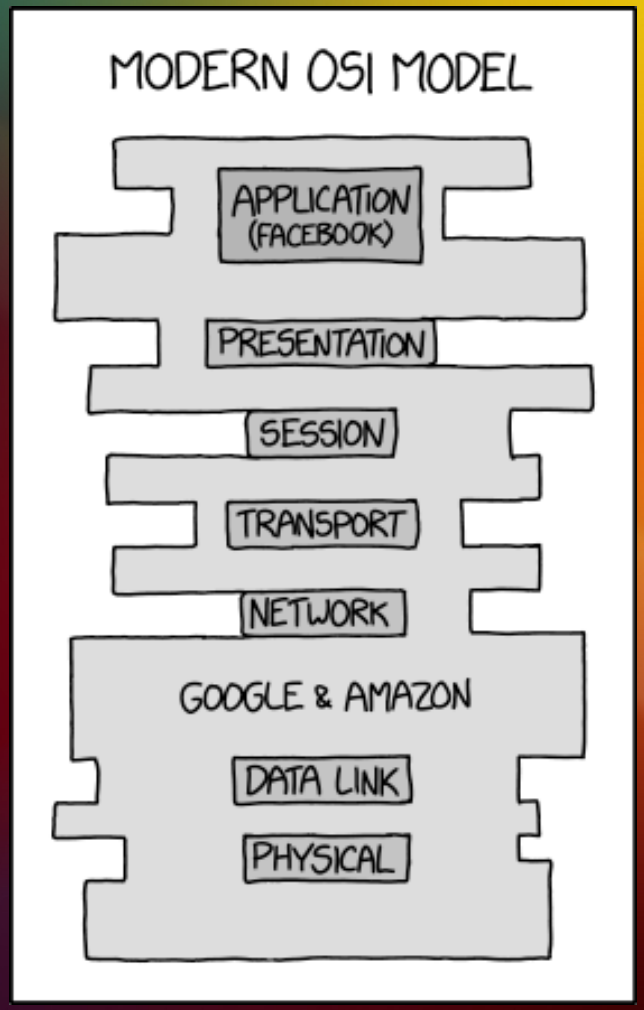
SELF-SERVICE IX PEERING VIA PEERINGDB CREDENTIALS (FOR ALL ELSE PEERING-APAC@AMAZON.COM)



Thank You

Lincoln Dale

Senior Principal Engineer
AWS – AS16509



Source: https://xkcd.com/2105, Randall Munroe