

How our Cloudy Mindsets Approached Physical Routers

SNMP was not an option

Steffen Gebert

DENOG12, 09.11.2020

Abstract

After the latest project, EMnify became a 99% only cloud company. To meet growing scalability and reliability requirements of the interconnection between our AWS-based deployments and multiple carriers, BGP peerings had to be moved out of AWS. Therefore, a pair of Juniper routers were put into place. For a company fully relying on cloud services so far, this alien technology resulted in several challenges.

We want to share, how we solved the integration puzzle of this physical equipment into our existing workflows and tools. The use of CI/CD systems for applying changes, AWS CloudWatch, Prometheus and Grafana for monitoring as well as the reluctance to run applications that require a lot of shepherding lead our research to find the right glue - the glue between these pieces of iron and our cloud infrastructure.

Being used to CI/CD processes backed by automated tests, we wanted to adapt these practices here as well. As a result, configuration changes are rolled out by an automated pipeline using Ansible. Efforts for automated testing were made, where we failed. We explain why and what we did instead as well as what we envision for the future.

As every other part of our system, we want its monitoring data accessible via Grafana.

With the help of pmacct and fluentbit, we can treat IPFIX flow records as they were logs. With the help of jtimon, Prometheus stores the routers' metrics as we are used to do, in doubt tickled out through few custom YANG models.

In summary, the integration worked very well, while we still have several learnings and pain points to share.

| Thanks to our Sponsors!

Diamond Sponsor



Platin Sponsor



**CORE
BACKBONE**

EMnify

*Virtual Support
Sponsor*



Streaming Location
EXARING AG

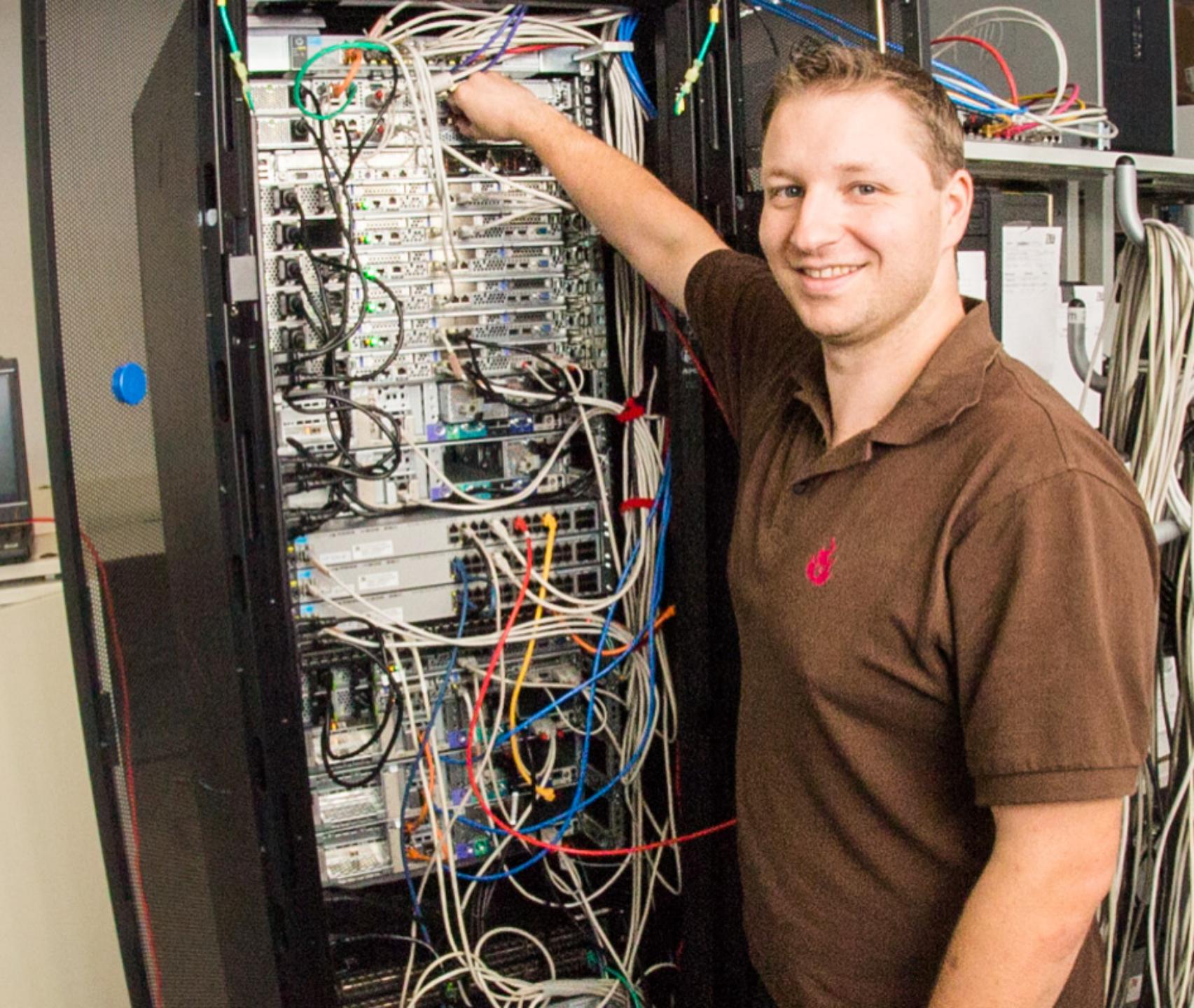


Cloudy Mindset?

| 5 years ago



EMnify



| 3-10 years ago



I Since 2017

EMnify

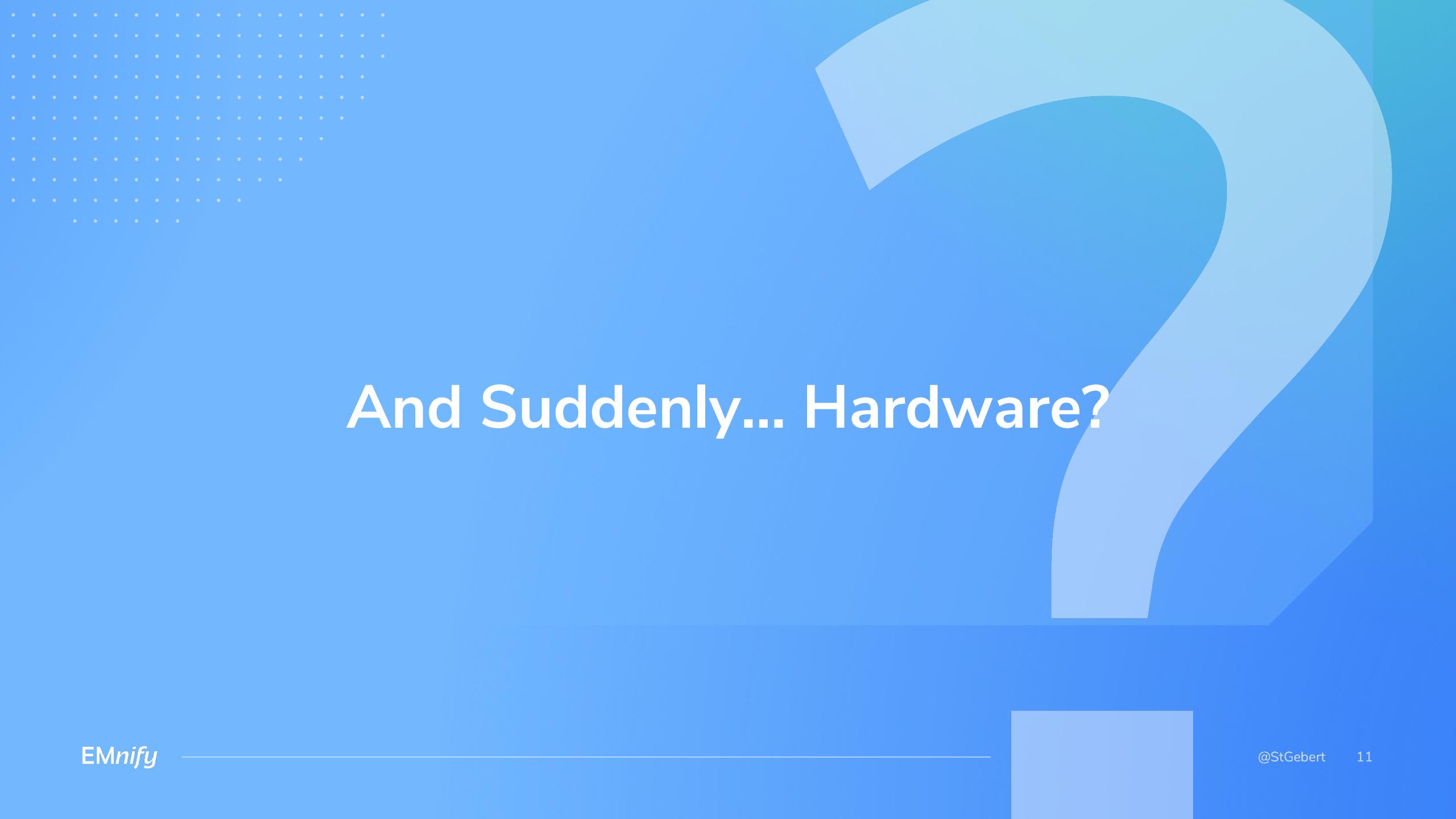


Is This a Better World?



Focus on Business Value

Prefer Managed Services



And Suddenly... Hardware?

| Agenda



EMnify's IoT Connectivity Platform



**Cellular connectivity
in 500+ networks in
185 countries**

RESTful APIs

**Pay as you go
pricing**

**SMS/USSD to REST
bridge**

**Secure connectivity
via VPN and AWS
natively**

**Implemented using
own virtualized
mobile core network**

Supporting Global IoT Deployments

Traditional Operators



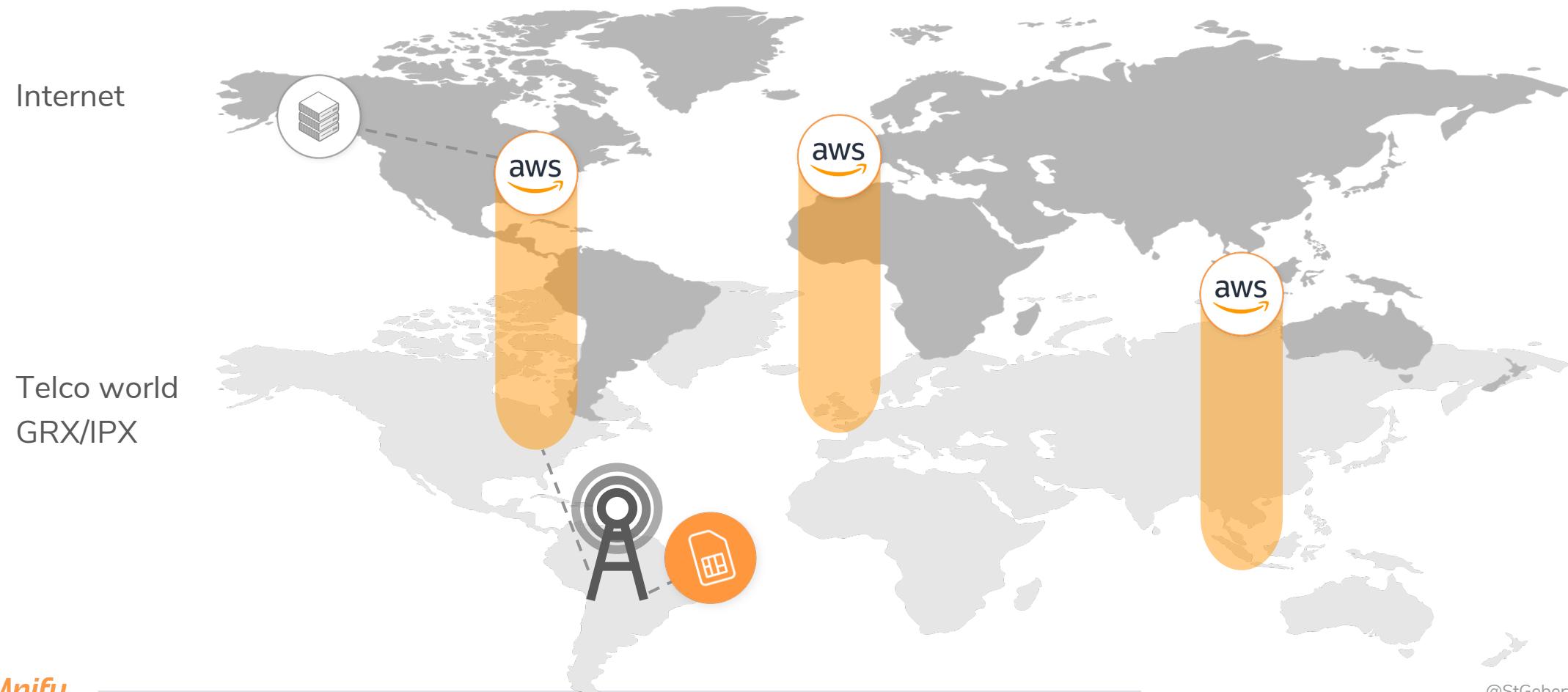
Home-routing of roaming SIM data
prevents distributed architecture

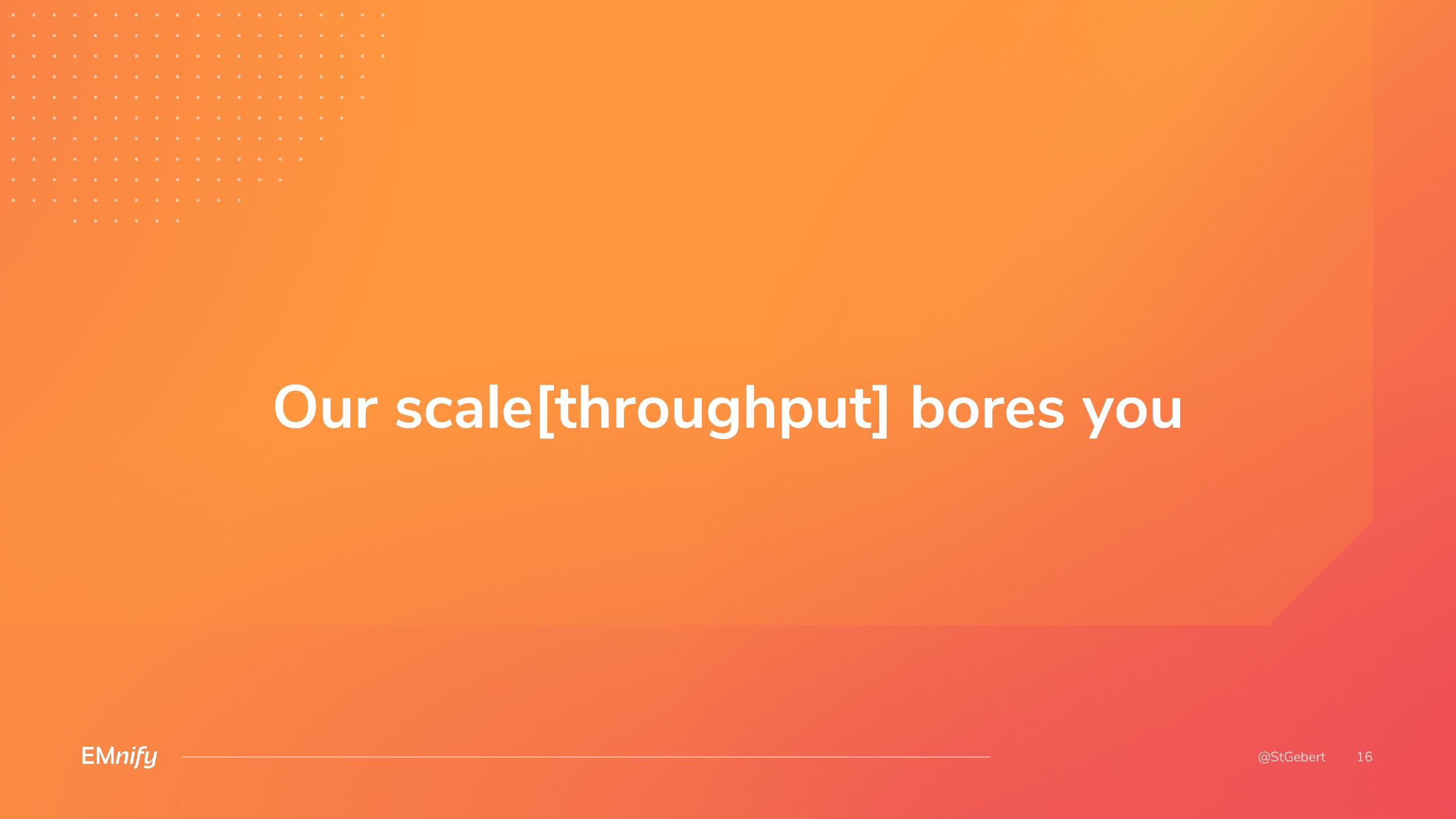
EMnify Connectivity



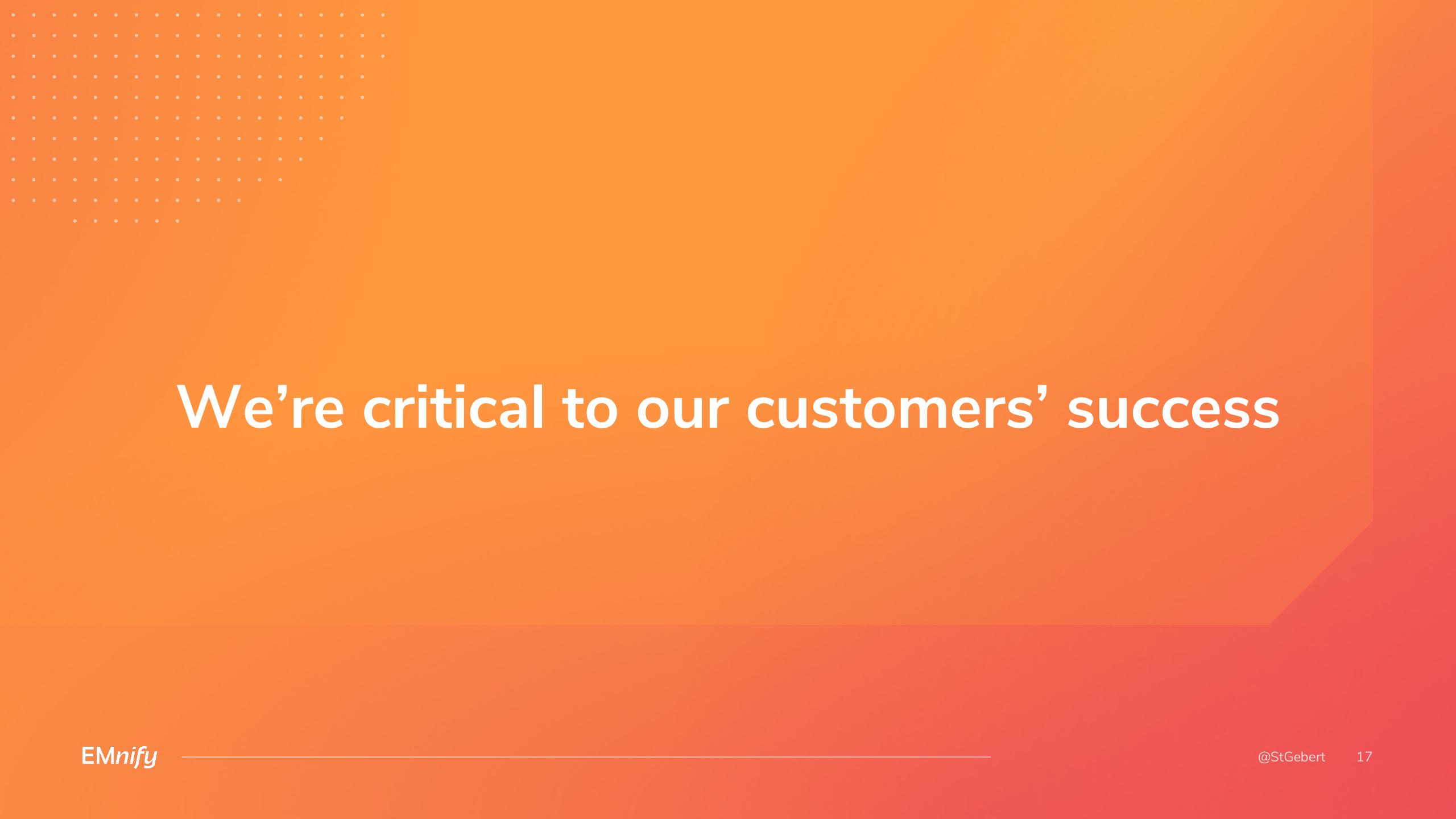
EMnify's mobile core network is
deployed in multiple AWS regions
– keeping data local

| GRX/IPX Network (GPRS Roaming Exchange)





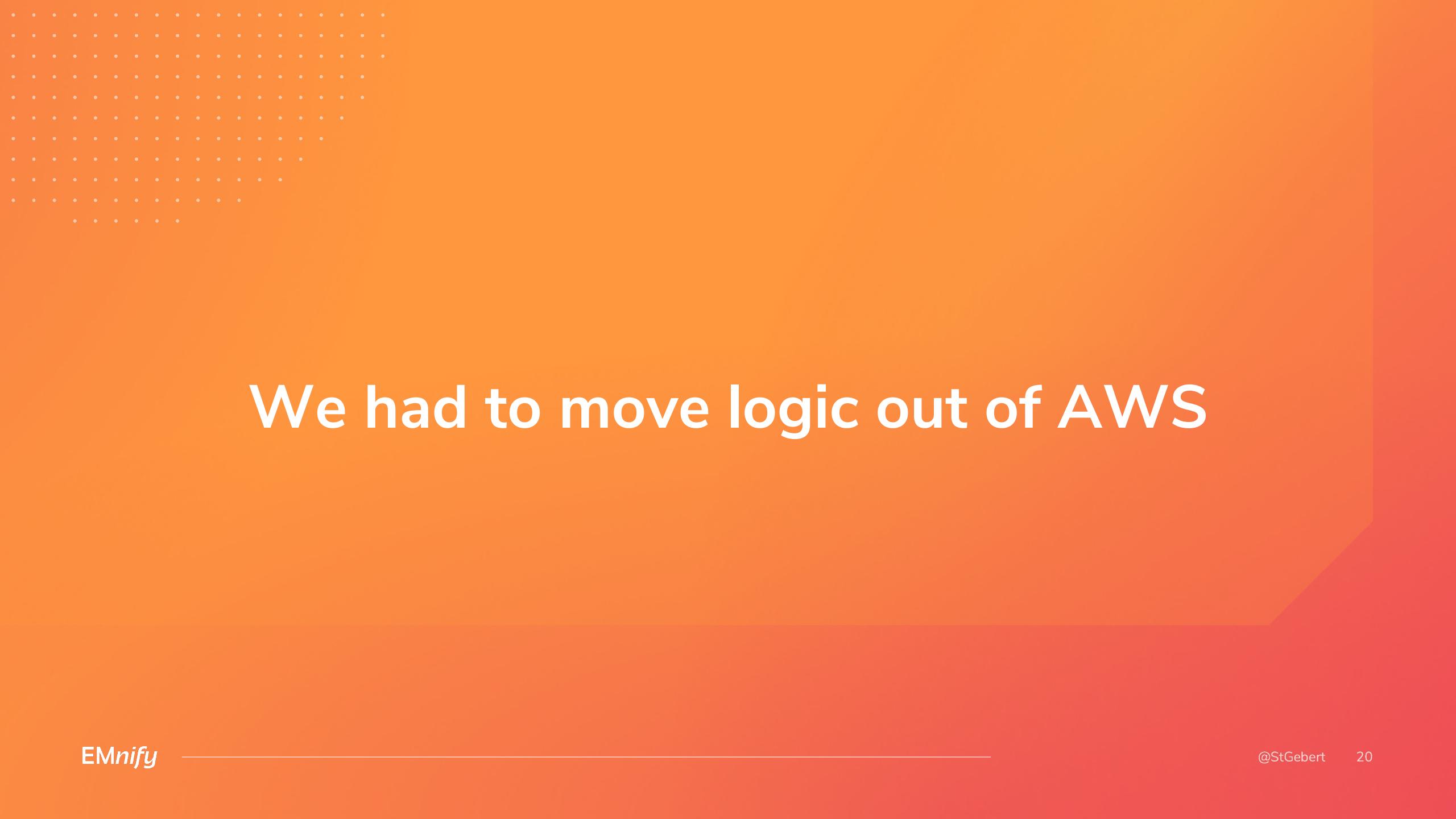
Our scale[throughput] bores you



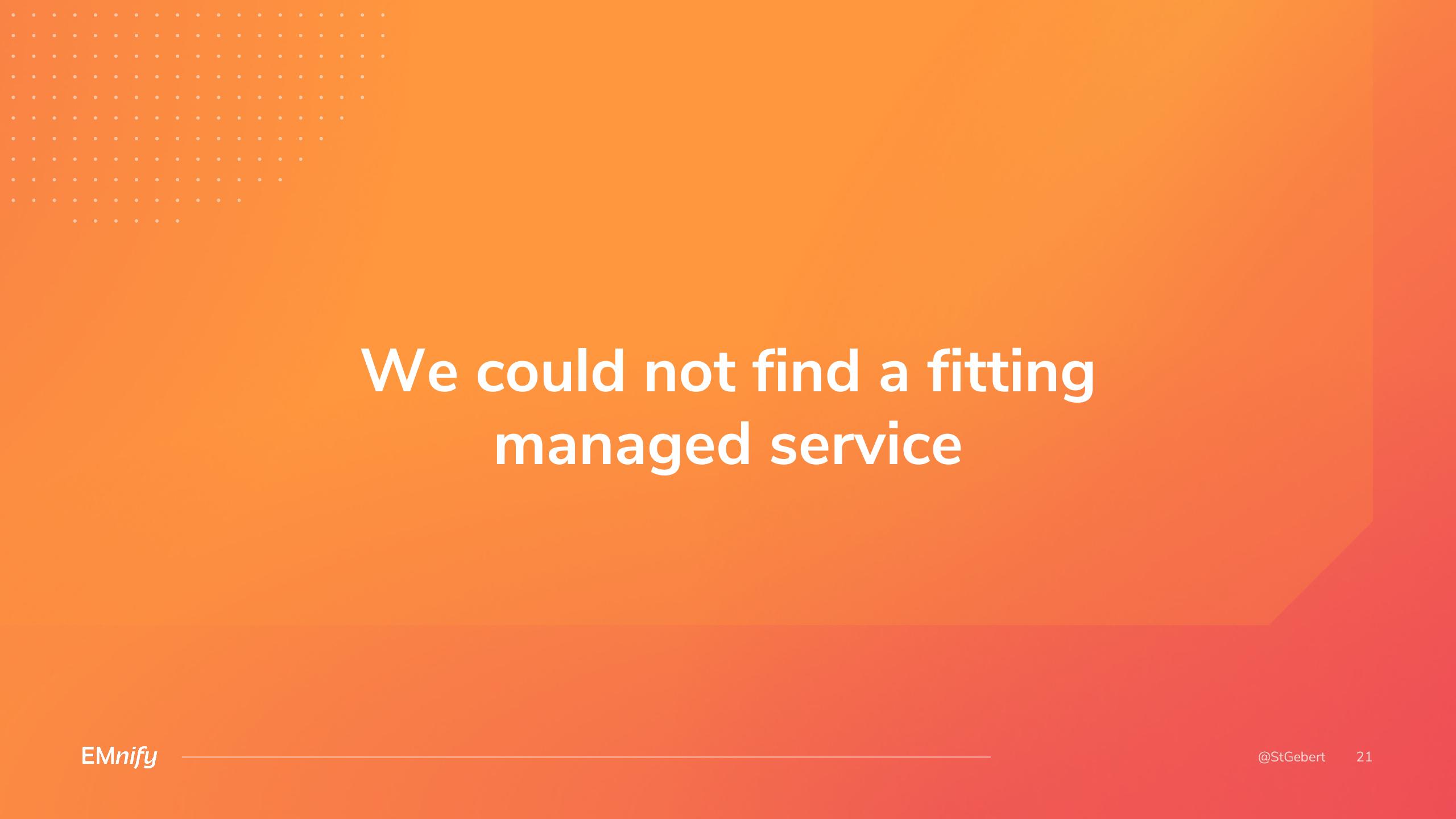
We're critical to our customers' success

Increased demands vs. AWS as “General Purpose Cloud”

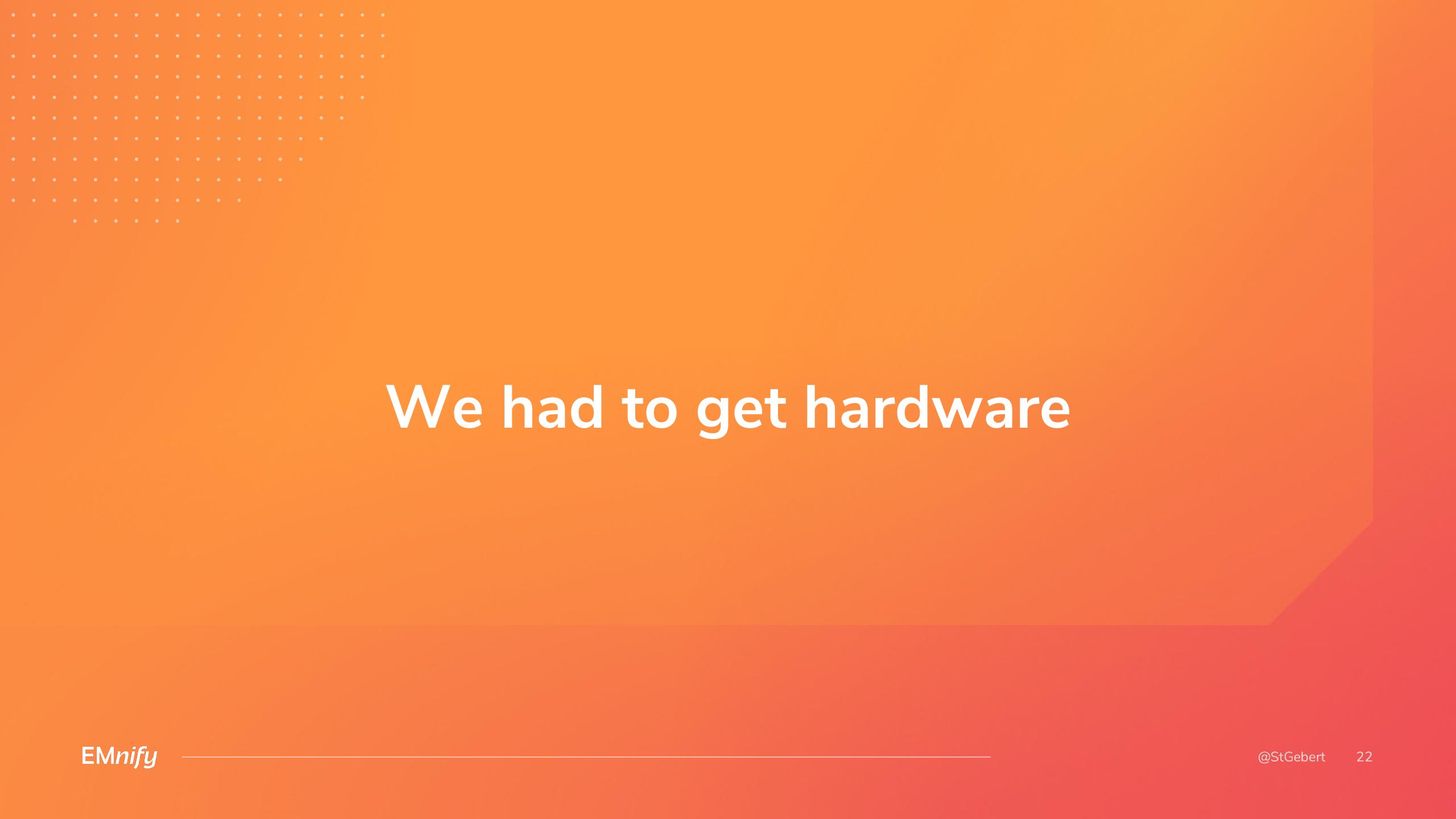
Running BGP on AWS?



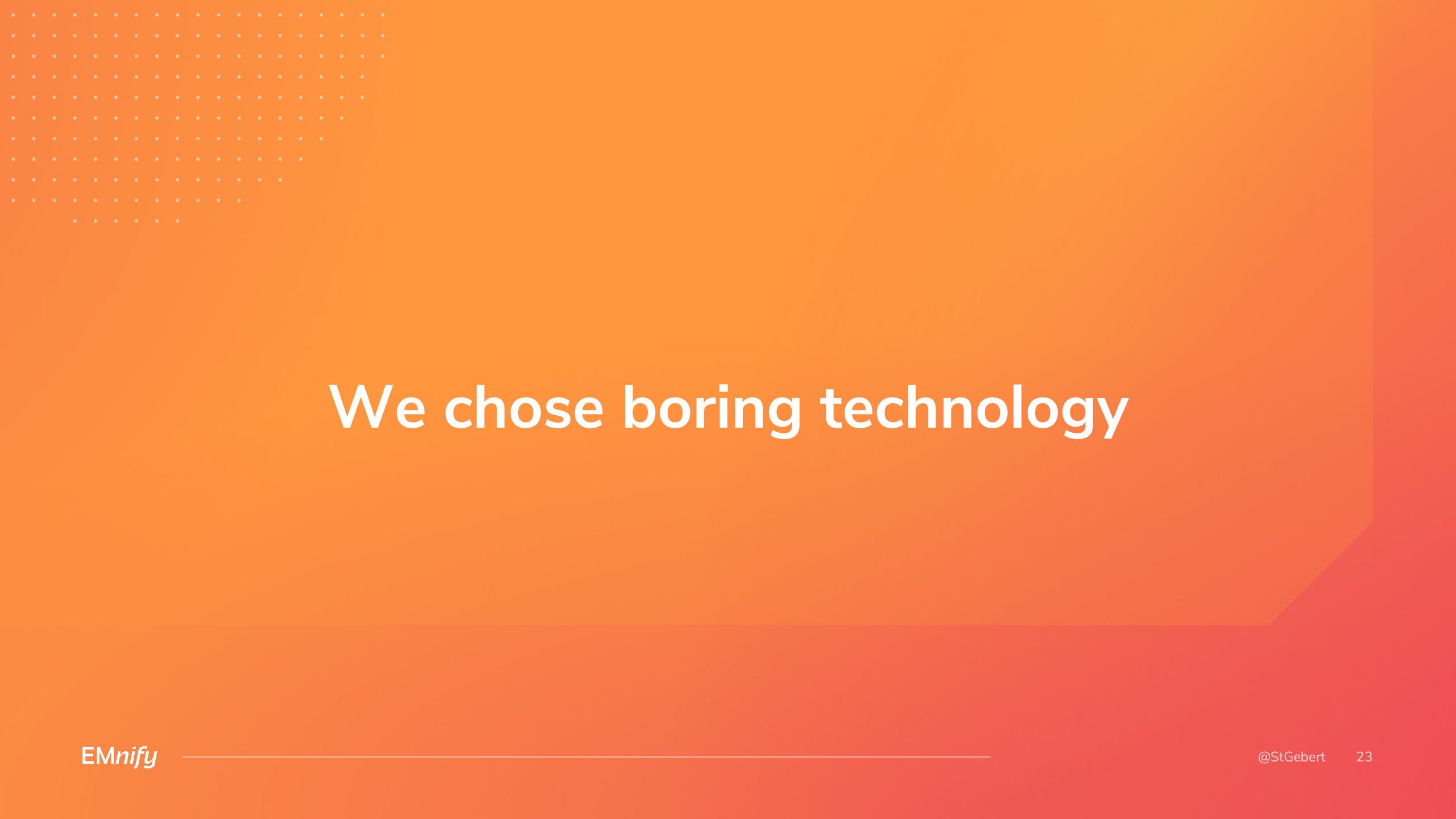
We had to move logic out of AWS



We could not find a fitting
managed service



We had to get hardware



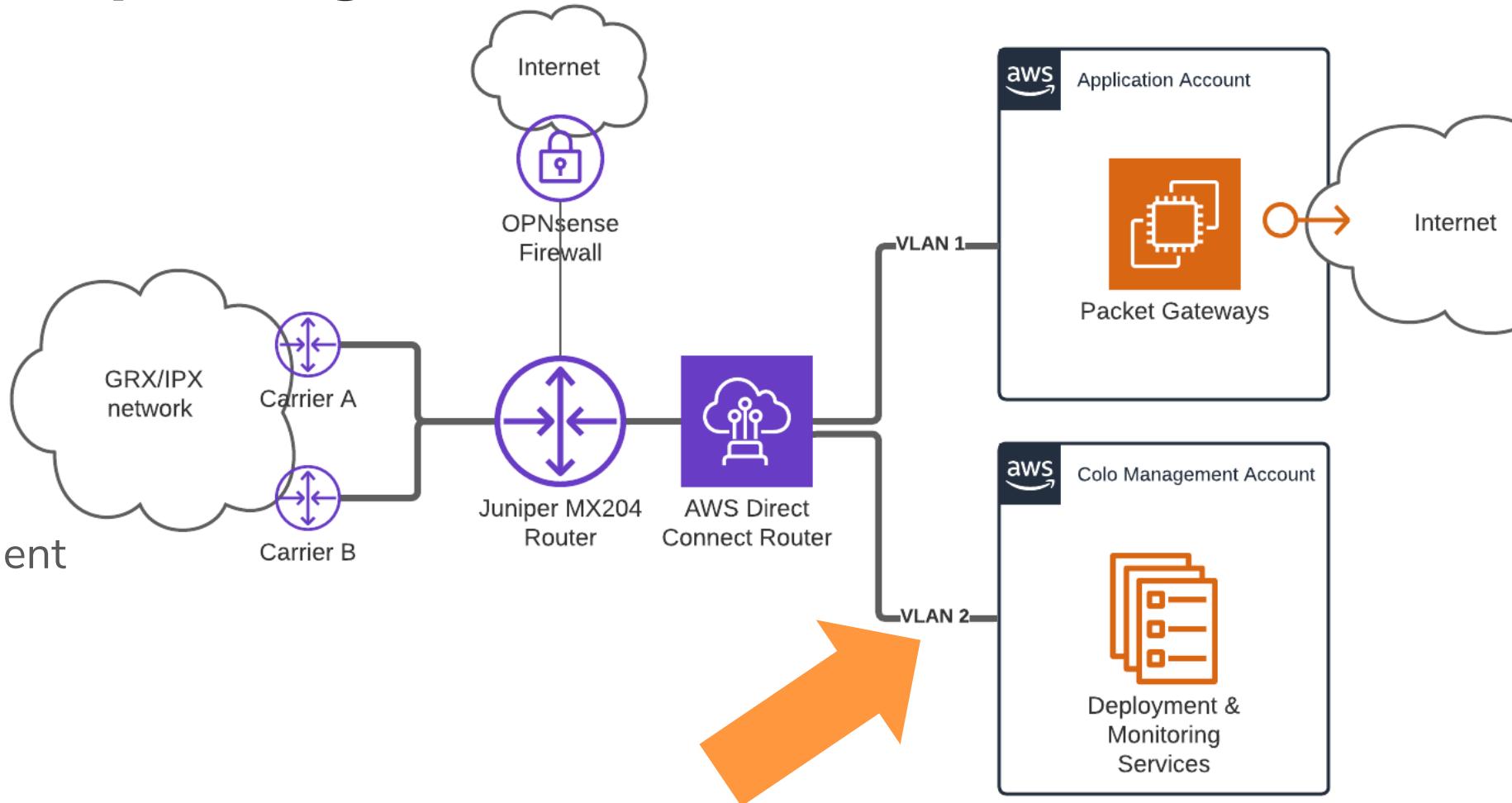
We chose boring technology



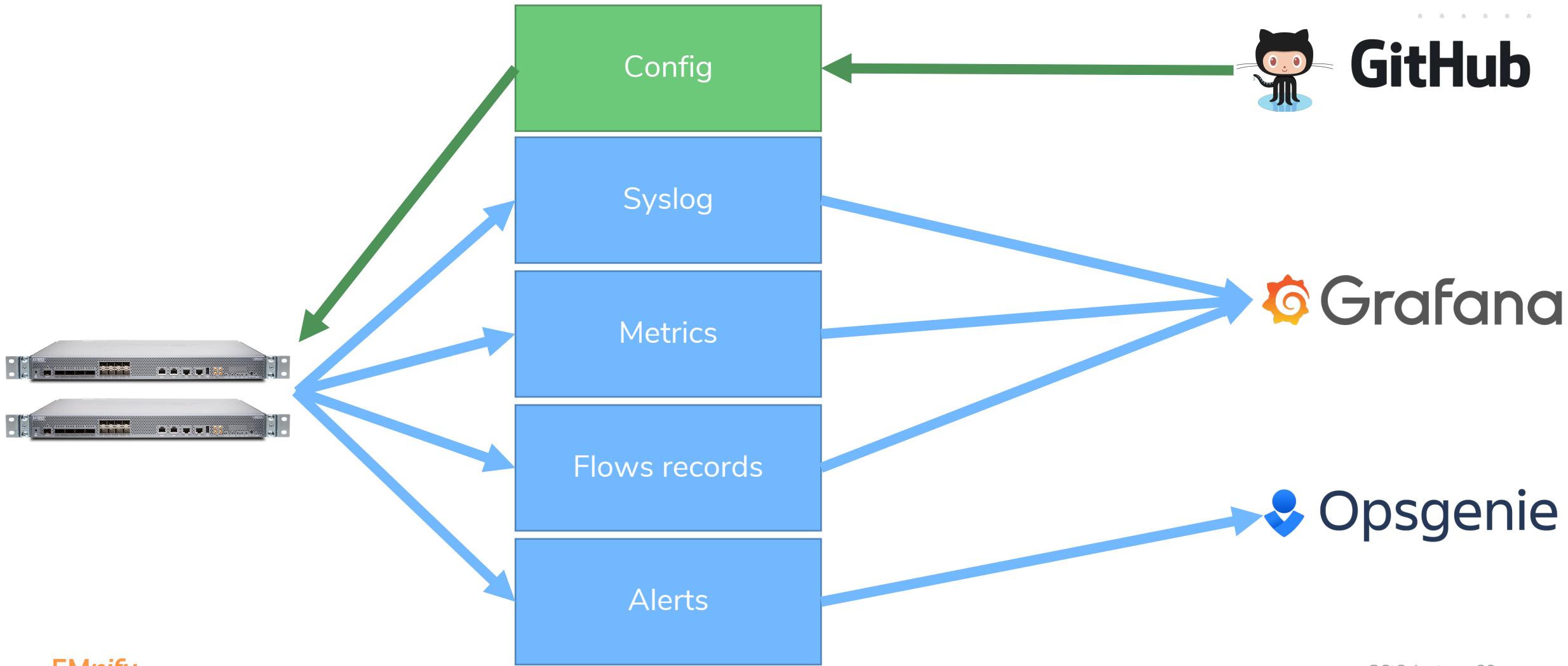
Greenfield project

Setup – Twice per region

- Juniper MX204
- Colocation rack space
- Fiber links towards 2 carriers
- AWS
- Out-of-band management access via OPNsense



Integration Points



| Design Principles

80/20 rule
aka
MVP

**Don't get out
of our comfort
zone**

**Don't setup
anythat that
requires lot of
handholding**

EMnify

Deployment

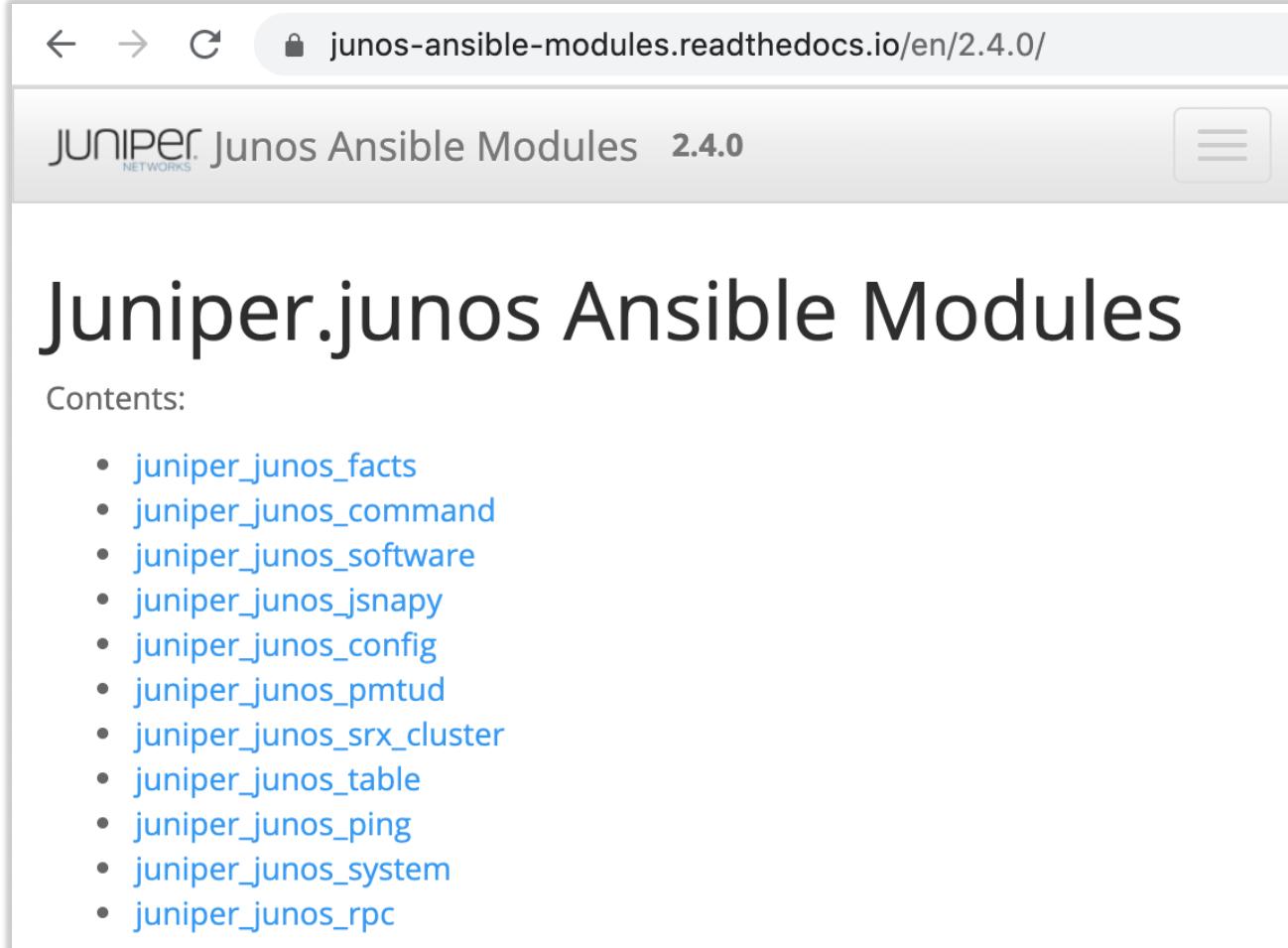


“

A human shall not SSH into
something

MY INNER SELF

| juniper_junos Ansible Modules

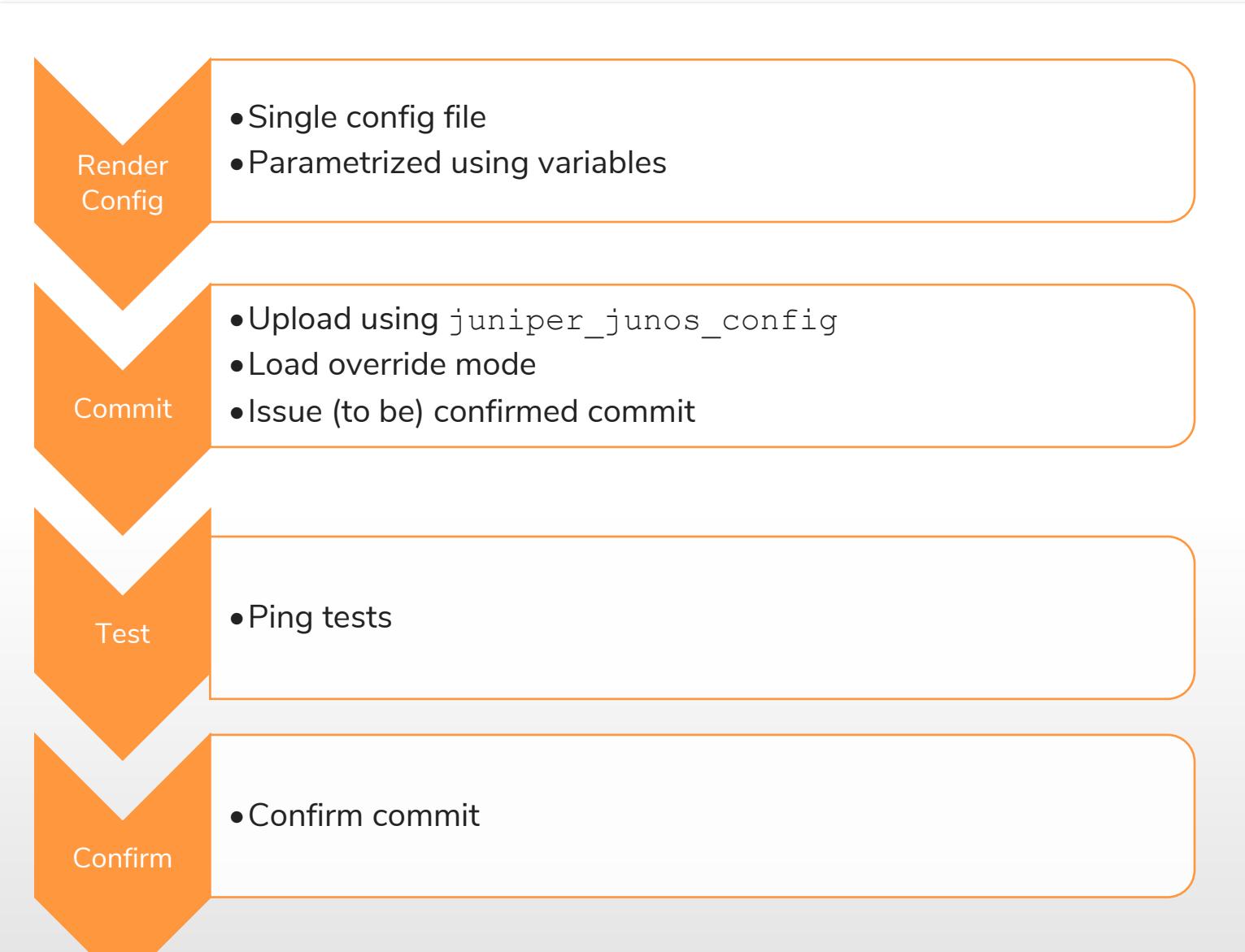


The screenshot shows a web browser displaying the Juniper Junos Ansible Modules documentation. The URL in the address bar is `junos-ansible-modules.readthedocs.io/en/2.4.0/`. The page title is "JUNIPER Junos Ansible Modules 2.4.0". Below the title, the main heading is "Juniper.junos Ansible Modules". A "Contents:" section lists the following modules:

- [juniper_junos_facts](#)
- [juniper_junos_command](#)
- [juniper_junos_software](#)
- [juniper_junos_jsnapy](#)
- [juniper_junos_config](#)
- [juniper_junos_pmtud](#)
- [juniper_junos_srx_cluster](#)
- [juniper_junos_table](#)
- [juniper_junos_ping](#)
- [juniper_junos_system](#)
- [juniper_junos_rpc](#)



Configuration Deployment

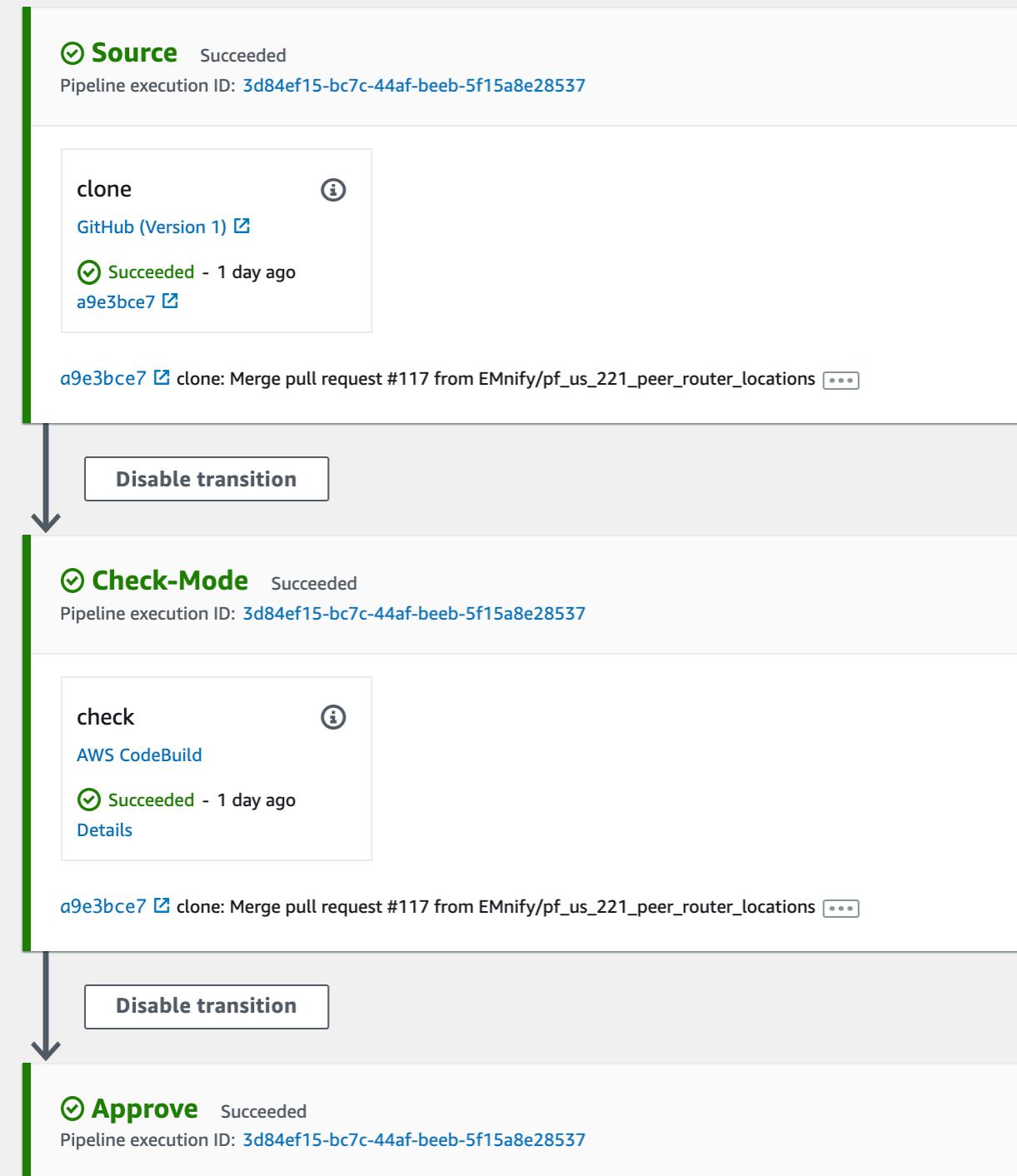
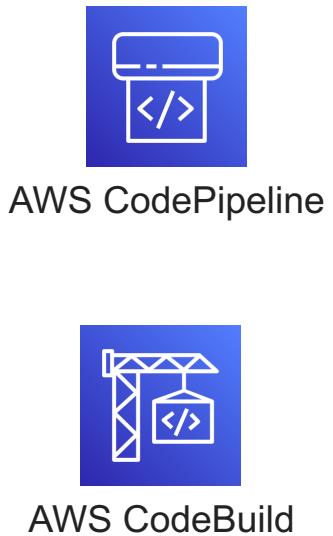


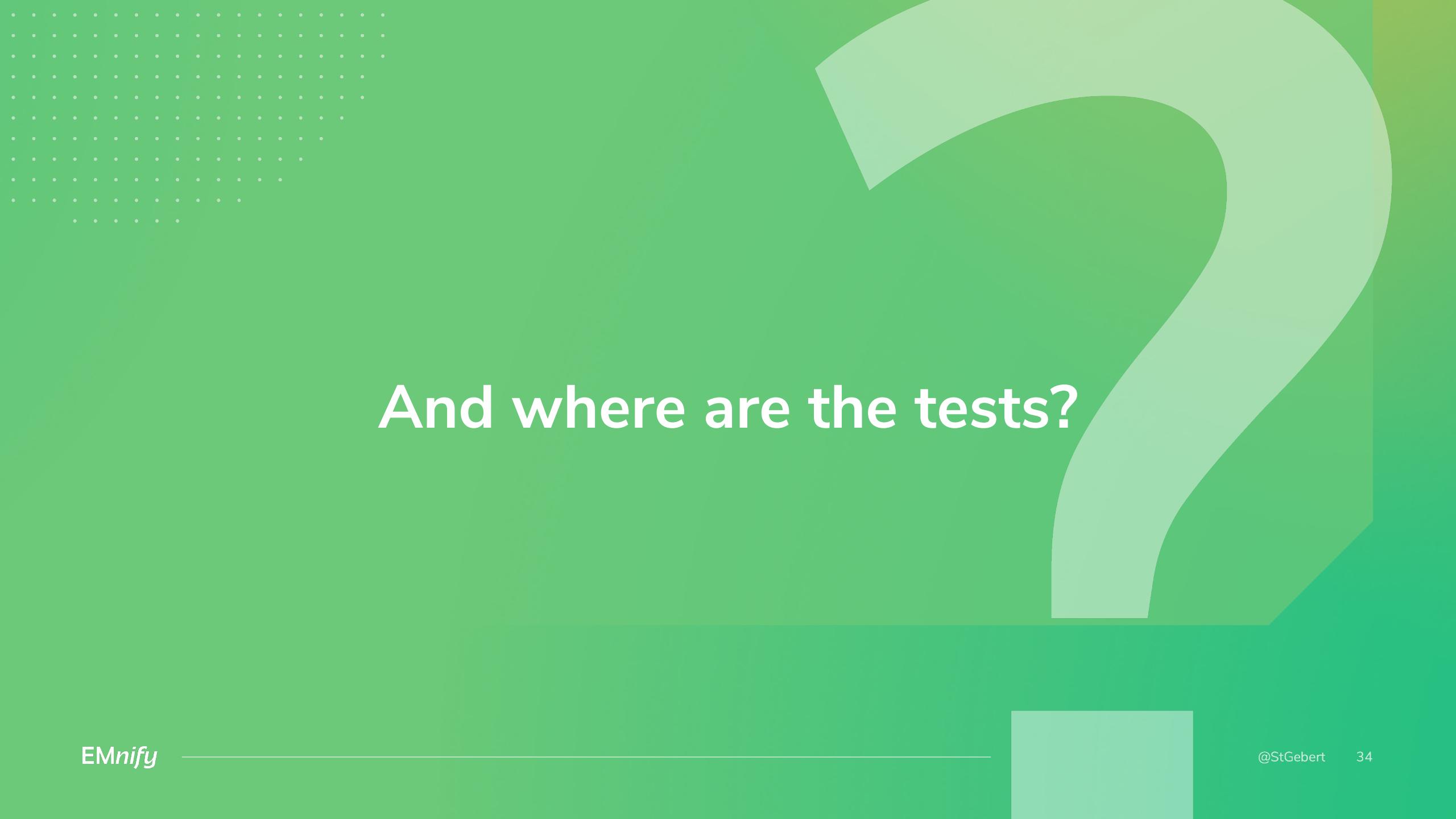
Ansible Playbook - Code Example

```
- name: install generated configuration file onto device
  juniper_junos_config:
    provider: "{{ juniper_connection_settings }}"
    src: "{{ conf_file }}"
    load: override
    comment: "playbook execution, commit confirmed"
    confirmed: 3 # wait X minutes until rollback
    diff: yes
    ignore_warning: yes
    register: config_results
    notify: confirm previous commit
```

Config Pipeline

- Separate AWS account
- Isolated connectivity

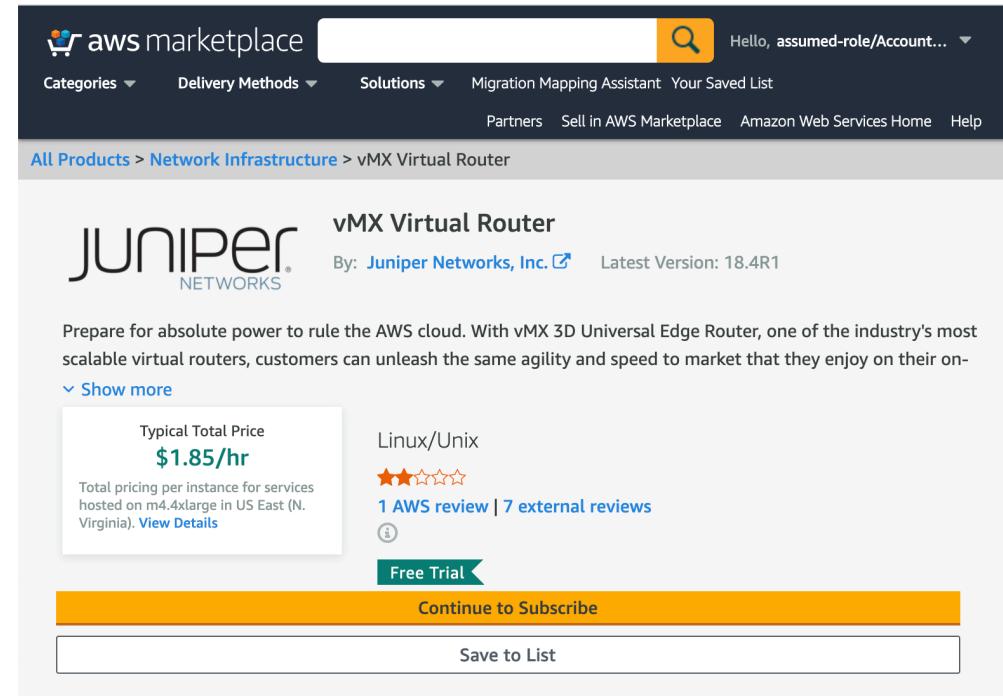




And where are the tests?

I In a Perfect World..

- 2-star review could have been mine :D
- Latest version: 18.4R1
- Takes ~30min to be ready
- AWS does not support VLANs!
- Only for manual testing
- Maybe eve-ng or GNS3 could help?



I On My Bucket List



- Start virtualized topology in network emulator
- Apply configuration pipeline
- Emulate BGP peers
- Execute end-to-end connectivity tests
- Emulate link failures
- Verify connectivity

- AWS: run on bare metal host (b/c CPU VMX)

Routine Operations (Runbooks)

Firmware Update - Checks

```
/workspace/prod # ansible-playbook upgrade_check.yaml -u steffen.gebert
...
TASK [Validate result] *****
[ mx204-am3 ] Chassis Alarms
-----
Expect:
No alarms currently active
Actual:
No alarms currently active
...
[ mx204-am3 ] Core Dumps
-----
Expect:
/var/crash/*core*: No such file or directory
Actual:
/var/crash/*core*: No such file or directory
```

[mx204-am3] ! Proceed? !

:

Press 'C' to continue the play or 'A' to abort

Firmware Update - Draining

- **name:** Drain traffic
 - juniper_junos_config:**
 - provider:** "{{ juniper_connection_settings }}"
 - load:** 'set'
 - lines:**
 - 'activate policy-options policy-statement OUT-OF-SERVICE-SWITCH term as-path-p'
 - comment:** 'Drain traffic to router for upgrade'

- **name:** Traffic drained
 - pause:**
 - prompt:** |
 - [{{item}}] Traffic is draining.
 - Verify that traffic is completely drained on the following dashboard before proceeding.
 - [{{item}}] ! Proceed with the JunOS upgrade !?
 - loop:** "{{ ansible_play_hosts }}"

| Firmware Update – Execute!

```
- name: Install Junos OS package
  juniper_junos_software:
    provider:
      host: "{{ ansible_host }}"
      timeout: 3600
      remote_package: "{{ junos_vm_file }}"
      validate: True
      cleanfs: False
      vmhost: True
      reboot: True
  ignore_errors: yes # rpc times out when upgrading, despite the provider timeout setting
  register: output
```

Challenges

Deploy a file
↖(ツ)↗

Max length
of file
copy URLs

Feedback for
invalid config

Amount of
boilerplate
code

EMnify

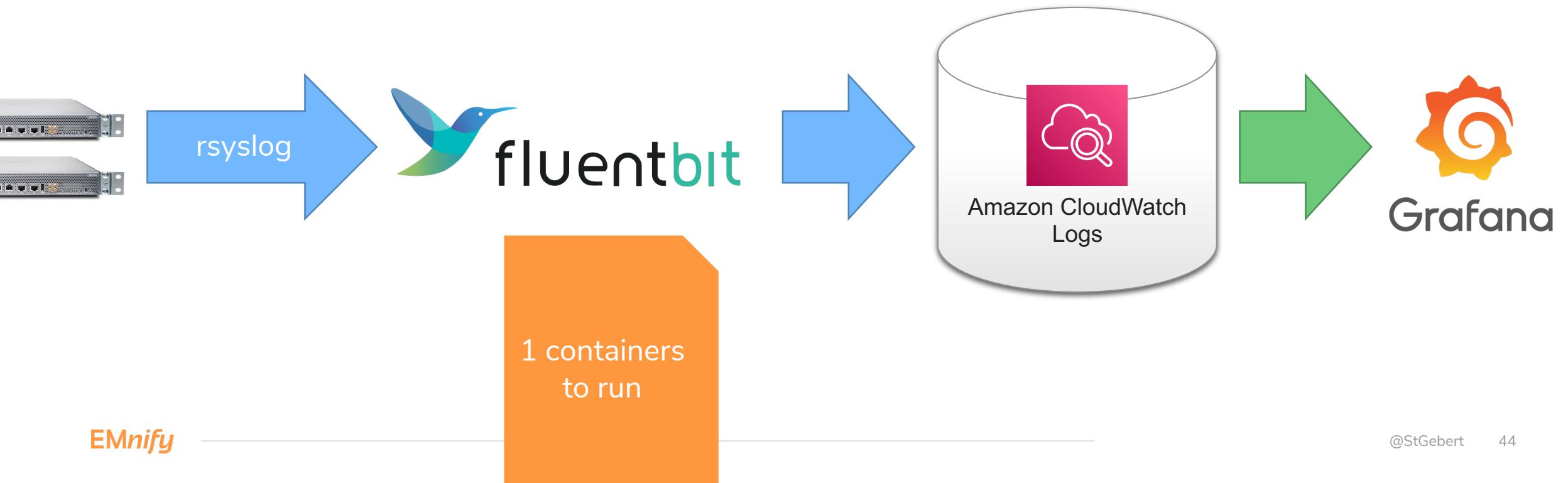
Monitoring



Syslogs

Syslog Implementation

- Who logged into the router?
- What's happening in the router?



Flow Records

Flow Records

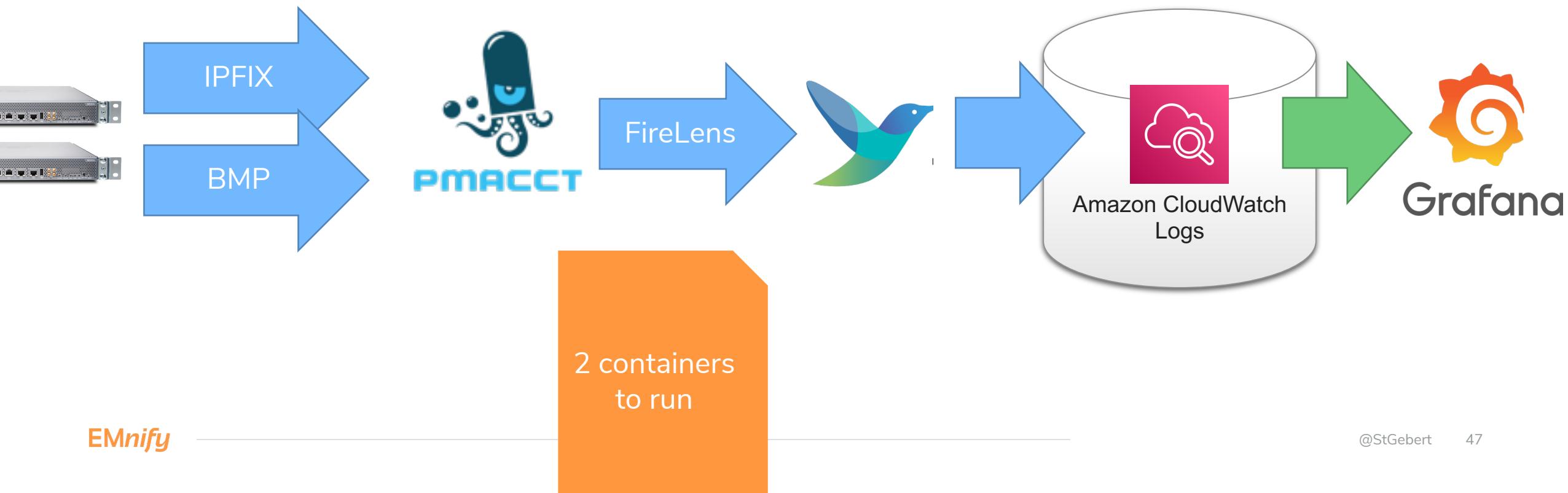
**Network-to-
Network
Interface
(GTP traffic)**

**~20k parallel
flows**

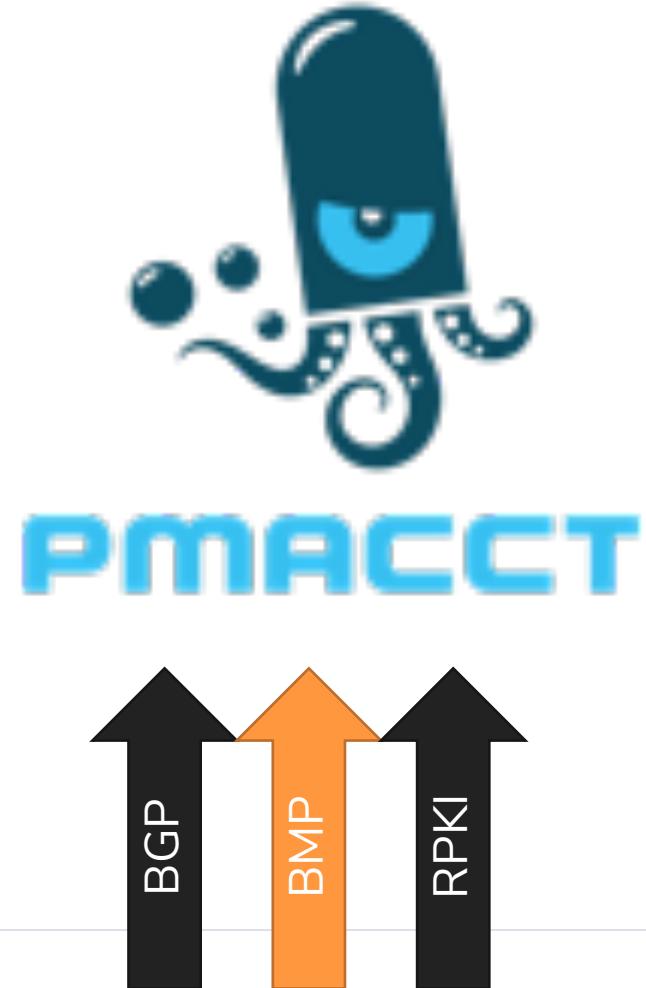
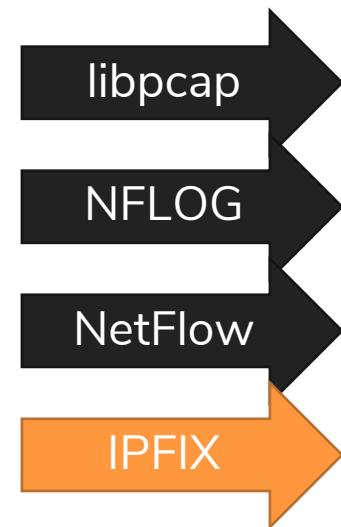
**1 flow =>
1 log line?!**

Flow Records Collection

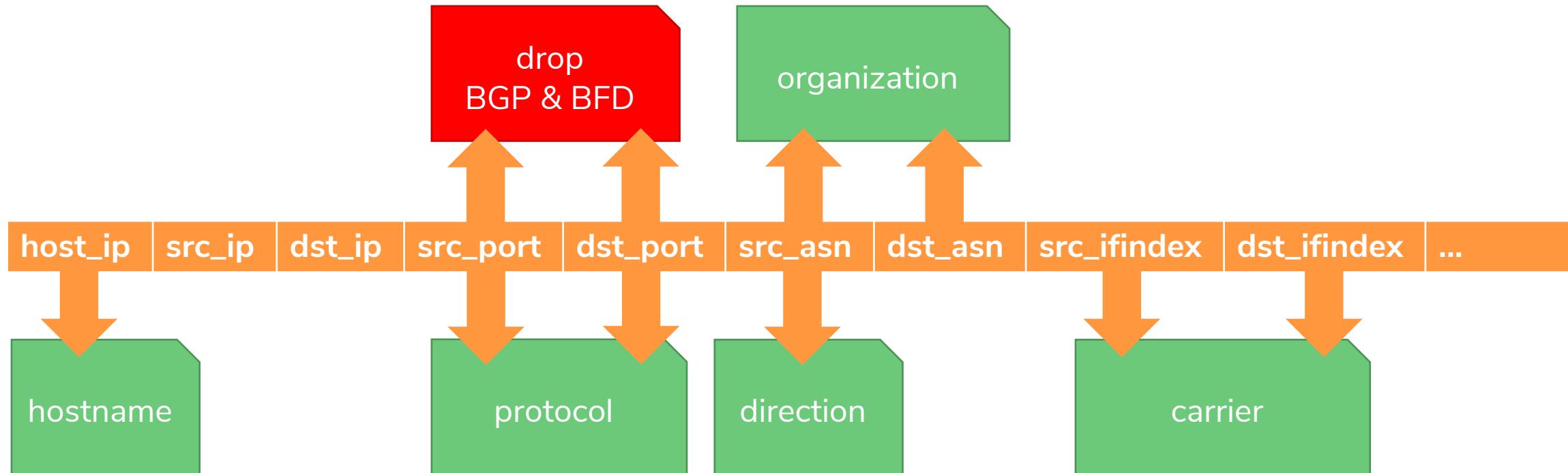
- How much traffic per AS?
- Did we receive any signaling from XYZ and did we really respond?



| pmacct / nfacct



Enrichment



| Lua Magic

```
-- Sets GTP-c or GTP-u protocol depending on port numbers
function setGTPProtocol(tag, timestamp, record)
    local code = 0
    local gtp_ports = {
        ["GTP-c"] = 2123,
        ["GTP-u"] = 2152,
        ["GTP'"] = 3386,
    }
    local new_record = record
    for protocol, port in pairs(gtp_ports)
    do
        if record["source.port"] == port or record["destination.port"] == port then
            new_record["network.application"] = "GTP"
            new_record["network.protocol"] = protocol
            code = 2
        end
    end
    return code, timestamp, new_record
```

Datasource

CloudWatch-colo-mgmt ▾

Operator

Chile

Carrier

All ▾

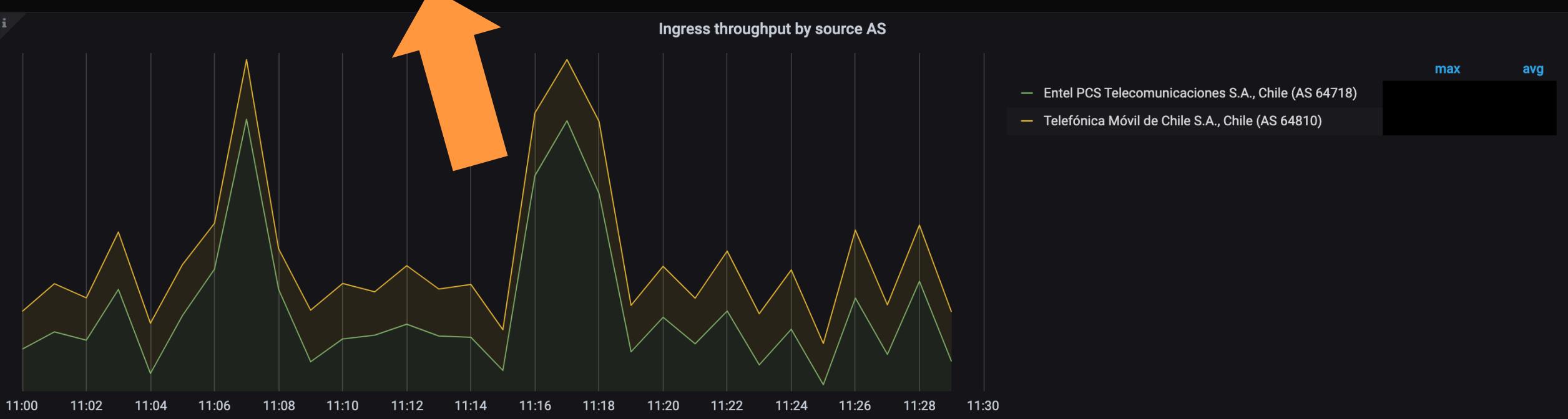
Host

All ▾

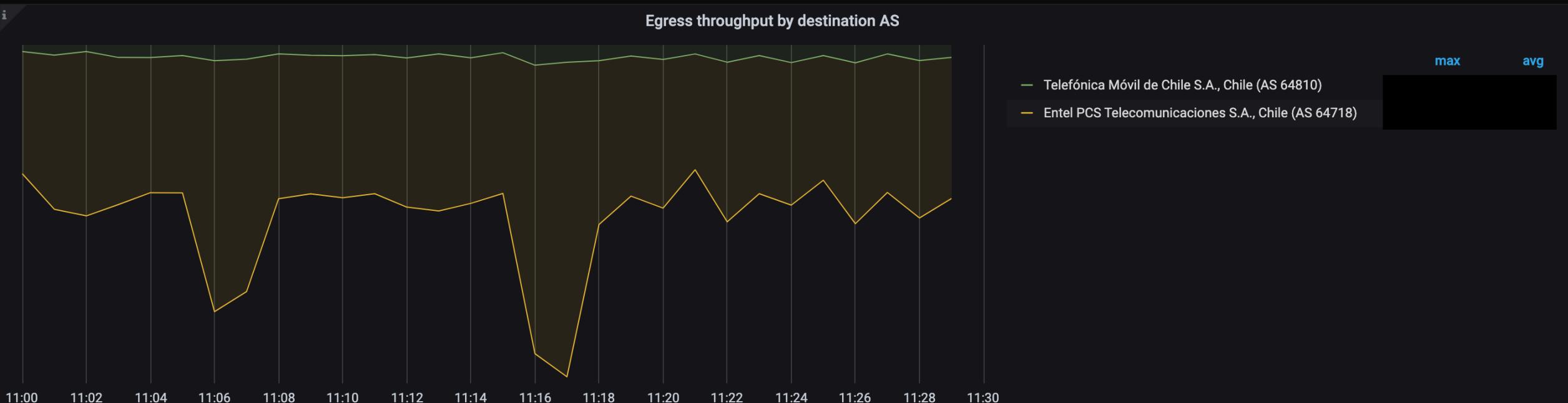
Protocol

All ▾

Ingress throughput by source AS



Egress throughput by destination AS



datasource	CloudWatch-col0-mgmt	operator	Chile	carrier	All	host	All	Protocol	GTP-c	Flows Records																			
Time		direction	protocol	src.org	src.country	src.asn	dst.org	dst.country	dst.tadig	bytes	packets	rate	src.ip	src.port	dst.ip	dst.port	transport	applic	hostname	dst.carrier	src.carrier	dst.as_path	src.tadig						
2020-10-19 11:29:10		inbound	GTP-c	Entel PCS Telecomu...	Chile	64718	EMnify	Ireland		114	34352	103	2123	udp	GTP	mx204-fr7					65001->>>	CHLMV							
2020-10-19 11:29:10		outbound	GTP-c	EMnify	Ireland	65001	Entel PCS Telecomu...	Chile	CHLMV	117	2123	13	34352	udp	GTP	mx204-am3					6774->64718->>								
2020-10-19 11:29:10		inbound	GTP-c	Telefónica Móvil de ...	Chile	64810	EMnify	Ireland		254	34416	72	2123	udp	GTP	mx204-am3					65001->>>	CHLTM							
2020-10-19 11:29:10		outbound	GTP-c	EMnify	Ireland	65001	Entel PCS Telecomu...	Chile	CHLMV	64718	182	2123	240	33904	udp	GTP	mx204-am3					6774->64718->>							
2020-10-19 11:29:10		inbound	GTP-c	Telefónica Móvil de ...	Chile	64810	EMnify	Ireland		236	34032	82	2123	udp	GTP	mx204-am3					65001->>>	CHLTM							
2020-10-19 11:29:10		inbound	GTP-c	Entel PCS Telecomu...	Chile	64718	EMnify	Ireland		213	33968	103	2123	udp	GTP	mx204-fr7					65001->>>	CHLMV							
2020-10-19 11:29:10		inbound	GTP-c	Entel PCS Telecomu...	Chile	64718	EMnify	Ireland		213	35184	103	2123	udp	GTP	mx204-fr7					65001->>>	CHLMV							
2020-10-19 11:29:10		outbound	GTP-c	EMnify	Ireland	65001	Entel PCS Telecomu...	Chile	CHLMV	64718	103	2123	12	34480	udp	GTP	mx204-am3					6774->64718->>							
2020-10-19 11:29:10		outbound	GTP-c	EMnify	Ireland	65001	Telefónica Móvil de ...	Chile	CHLTM	64810	172	2123	204	35120	udp	GTP	mx204-am3					6774->12956->65140->64810							
2020-10-19 11:29:10		outbound	GTP-c	EMnify	Ireland	65001	Entel PCS Telecomu...	Chile	CHLMV	64718	103	2123	8	35248	udp	GTP	mx204-am3					6774->64718->>							
2020-10-19 11:29:10		outbound	GTP-c	EMnify	Ireland	65001	Entel PCS Telecomu...	Chile	CHLMV	64718	172	2123	13	35504	udp	GTP	mx204-am3					6774->64718->>							
2020-10-19 11:29:10		outbound	GTP-c	EMnify	Ireland	65001	Entel PCS Telecomu...	Chile	CHLMV	64718	117	2123	12	33968	udp	GTP	mx204-fr7					6774->64718->>							
2020-10-19 11:29:10		outbound	GTP-c	EMnify	Ireland	65001	Entel PCS Telecomu...	Chile	CHLMV	64718	117	2123	193	2123	udp	GTP	mx204-fr7					6774->64718->>							
2020-10-19 11:29:10		inbound	GTP-c	Entel PCS Telecomu...	Chile	64718	EMnify	Ireland		112	33904	103	2123	udp	GTP	mx204-fr7					65001->>>	CHLMV							
2020-10-19 11:29:10		outbound	GTP-c	EMnify	Ireland	65001	Entel PCS Telecomu...	Chile	CHLMV	64718	182	2123	103	2123	udp	GTP	mx204-fr7					6774->64718->>							
2020-10-19 11:29:10		inbound	GTP-c	Telefónica Móvil de ...	Chile	64810	EMnify	Ireland		254	34992	103	2123	udp	GTP	mx204-fr7					65001->>>	CHLTM							
2020-10-19 11:29:10		outbound	GTP-c	EMnify	Ireland	65001	Entel PCS Telecomu...	Chile	CHLMV	64718	117	2123	8	34928	udp	GTP	mx204-fr7					6774->64718->>							
2020-10-19 11:29:10		inbound	GTP-c	Entel PCS Telecomu...	Chile	64718	EMnify	Ireland		213	34928	103	2123	udp	GTP	mx204-fr7					65001->>>	CHLMV							
2020-10-19 11:29:10		outbound	GTP-c	EMnify	Ireland	65001	Entel PCS Telecomu...	Chile	CHLMV	64718	172	2123	8	35376	udp	GTP	mx204-fr7					6774->64718->>							
2020-10-19 11:29:10		inbound	GTP-c	Entel PCS Telecomu...	Chile	64718	EMnify	Ireland		112	33968	103	2123	udp	GTP	mx204-fr7					65001->>>	CHLMV							
2020-10-19 11:29:10		outbound	GTP-c	EMnify	Ireland	65001	Telefónica Móvil de ...	Chile	CHLTM	64810	117	2123	204	35024	udp	GTP	mx204-am3					6774->12956->65140->64810							
2020-10-19 11:29:11		outbound	GTP-c	EMnify	Ireland	65001	Telefónica Móvil de ...	Chile	CHLTM	64810	117	2123	254	36528	udp	GTP	mx204-fr7					6774->12956->65140->64810							
2020-10-19 11:29:11		outbound	GTP-c	EMnify	Ireland	65001	Entel PCS Telecomu...	Chile	CHLMV	64718	117	2123	12	2123	udp	GTP	mx204-am3					6774->64718->>							
2020-10-19 11:29:11		inbound	GTP-c	Telefónica Móvil de ...	Chile	64810	EMnify	Ireland		204	35216	72	2123	udp	GTP	mx204-am3					65001->>>	CHLTM							
2020-10-19 11:29:11		outbound	GTP-c	EMnify	Ireland	65001	Entel PCS Telecomu...	Chile	CHLMV	64718	117	2123	13	34032	udp	GTP	mx204-am3					6774->64718->>							
2020-10-19 11:29:11		outbound	GTP-c	EMnify	Ireland	65001	Entel PCS Telecomu...	Chile	CHLMV	64718	117	2123	201	2123	udp	GTP	mx204-am3					6774->64718->>							
2020-10-19 11:29:11		inbound	GTP-c	Telefónica Móvil de ...	Chile	64810	EMnify	Ireland		204	35568	82	2123	udp	GTP	mx204-am3					65001->>>	CHLTM							

Inbound traffic by AS query

```
fields concat(source.as.organization.name, ' ',  
             source.as.organization.country, ' (AS ', source.as.number, ')') as org  
| filter @logStream = "flows"  
| filter host.name like /$host$/  
| filter concat(source.as.number, ' ', source.as.organization.name, ' ',  
                source.as.organization.country, ' ', source.as.organization.tadig) like /$operator/  
OR concat(destination.as.number, ' ', destination.as.organization.name, ' ',  
          destination.as.organization.country, ' ', destination.as.organization.tadig)  
          like /$operator/  
| filter network.peer.destination.as.organization.name like /$carrier$/  
| filter network.direction = "inbound"  
| filter network.protocol like /$protocol/  
| filter 10000  
| stats sum(network.bytes)/60*8 as `` by org,bin($time_interval)  
| sort `` desc
```

Challenges

CloudWatch
Read Limits

CloudWatch
Write Limits

pmacct
config
“creativity”

Metrics

Metrics Demand



Temperature, light
levels, etc.



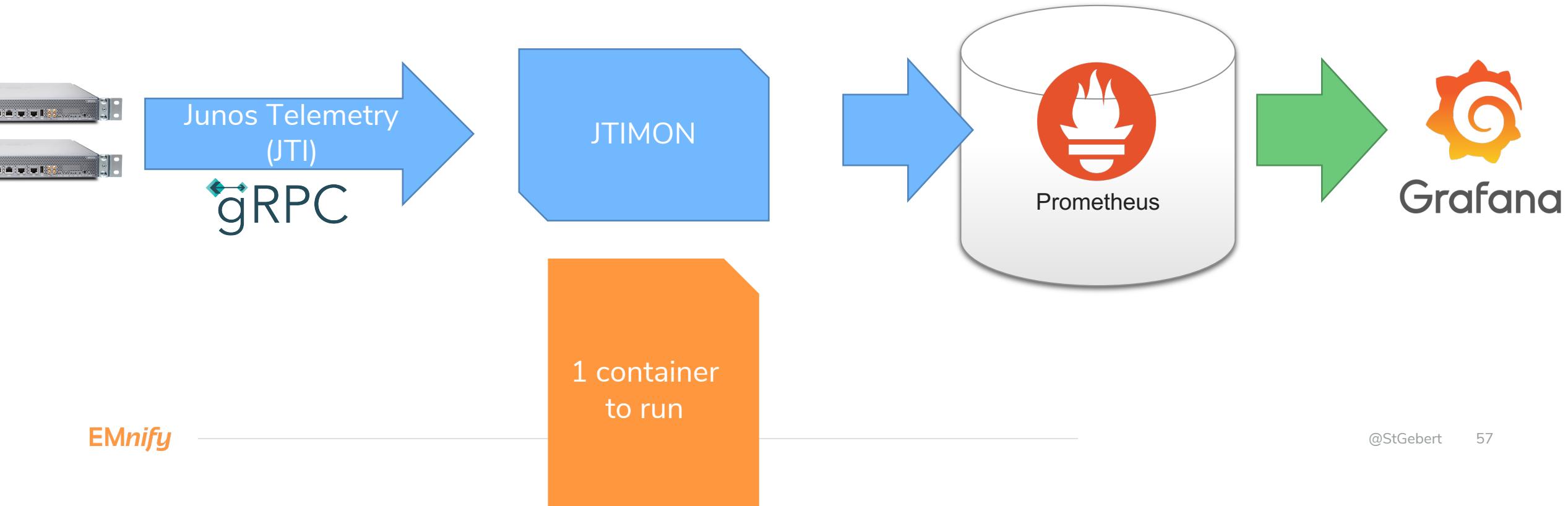
State, throughput,
errors, etc.



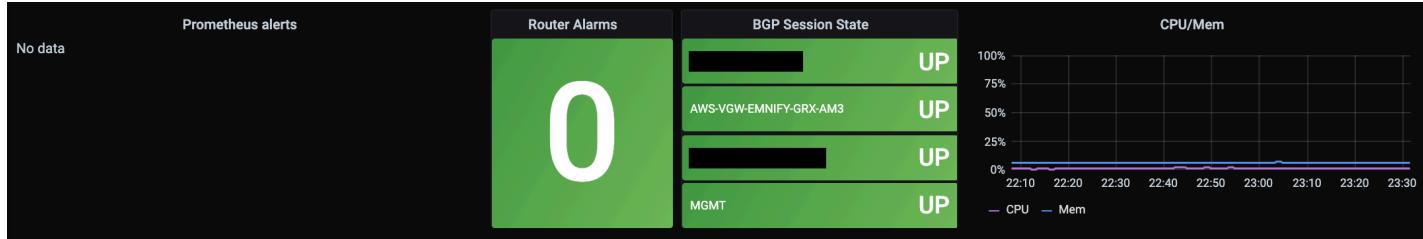
State, prefixes
received/accepted/installed

Metrics Implementation

- High cardinality, high frequency metrics collection



Metrics Examples



✓ BGP - details

BGP Information											
Name	Neighbor ↓	device	UP	Established	Transitions	Received Prefixes	Accepted Prefixes	Installed Prefixes	Sent Prefixes		
EMNIFY-GRX	172.23.94.33	mx204-am3.c	UP	2020-10-13 16:34:17	8	10853	10853	217	2		
EMNIFY-GRX	172.22.94.33	mx204-fr7.col	UP	2020-10-15 21:14:22	10	10853	10853	217	2		
EMNIFY-GRX	10.246.176.217	mx204-fr7.col	UP	2020-10-08 18:19:14	5	10736	10733	10733	2		
EMNIFY-GRX	10.246.176.17	mx204-am3.c	UP	2020-10-08 18:19:07	5	10736	10733	10733	2		
EMNIFY-GRX	10.90.1.13	mx204-fr7.col	UP	2020-10-16 12:36:13	8	3	3	3	4		
EMNIFY-GRX	10.90.1.9	mx204-am3.c	UP	2020-10-09 16:01:21	4	3	3	3	4		
master	10.90.1.5	mx204-fr7.col	UP	2020-10-16 12:36:12	7	1	1	1	3		
master	10.90.1.1	mx204-am3.c	UP	2020-10-09 16:04:38	4	1	1	1	3		

Challenges

JTI Sensor
availability

JTIMON
config file
duplication

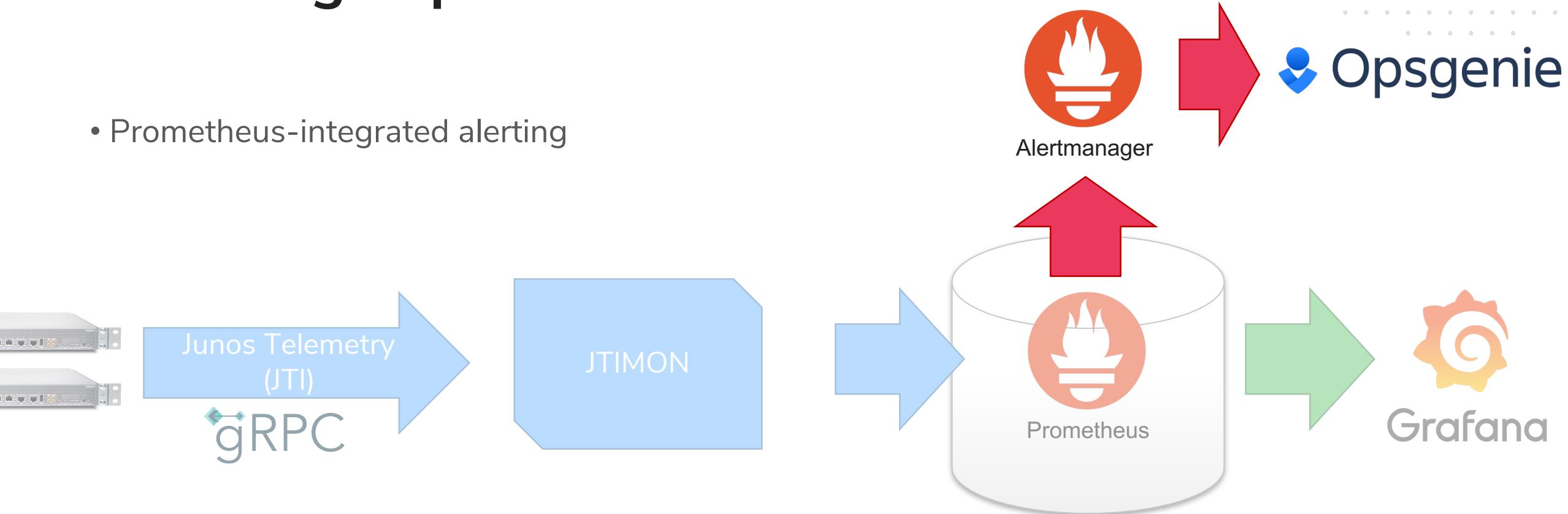
JTIMON ENUM
support

PKI setup

Alerting

Alerting Implementation

- Prometheus-integrated alerting



| Summary & Conclusion

- Integrated hardware into an otherwise fully cloud-based environment
 - Avoid new processes
 - Avoid new (user-facing) tooling
- Found tooling to bridge gaps to “what we’re comfortable with”
 - 1-2 containers running existing open source tooling
 - No guarantee that this scales to 10s of devices
- Please contact me, if you want details (configs etc.) or have suggestions!

EMnify

Need a Lockdown Project?

Go to emnify.com/devs



EMnify

Develop future-proof IoT solutions with seamless integration

We partner with your business to deliver smart IoT and M2M connectivity solutions. Build efficient and innovative applications with our REST-ful API, programmable SIMs and support from our technical experts.

- Transparent pricing
- Efficient monitoring
- Fast and secure

[Request a Free Evaluation](#)