CUMULUS

# If (Network == Server)
{
Magic happens
}

November 9th, 2017

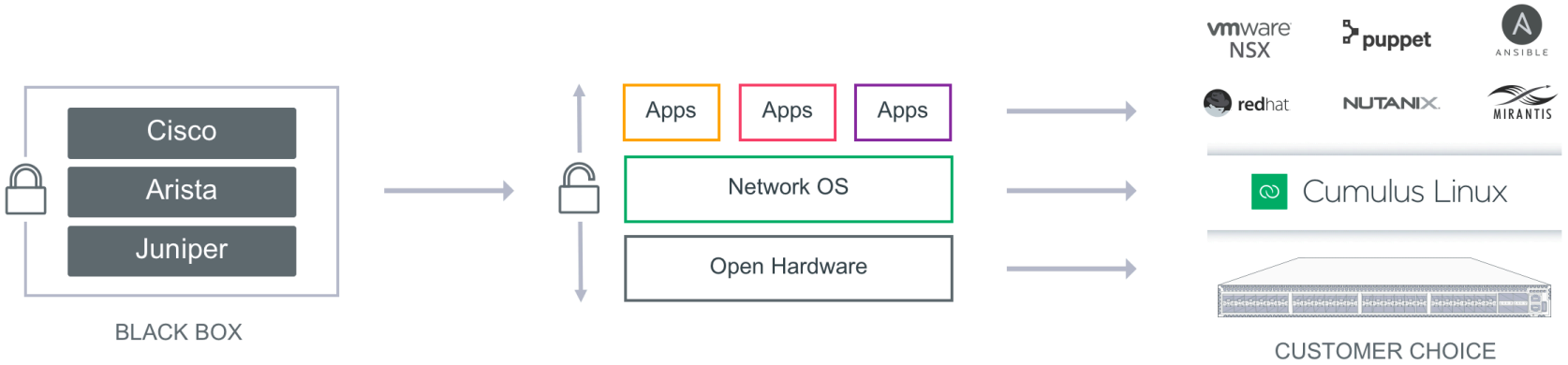
Attilla de Groot | SE, HCIE #3494 | Cumulus Networks

Agenda

- Cumulus
- NetQ & Hostpack
- Network CI / CD
- Demo

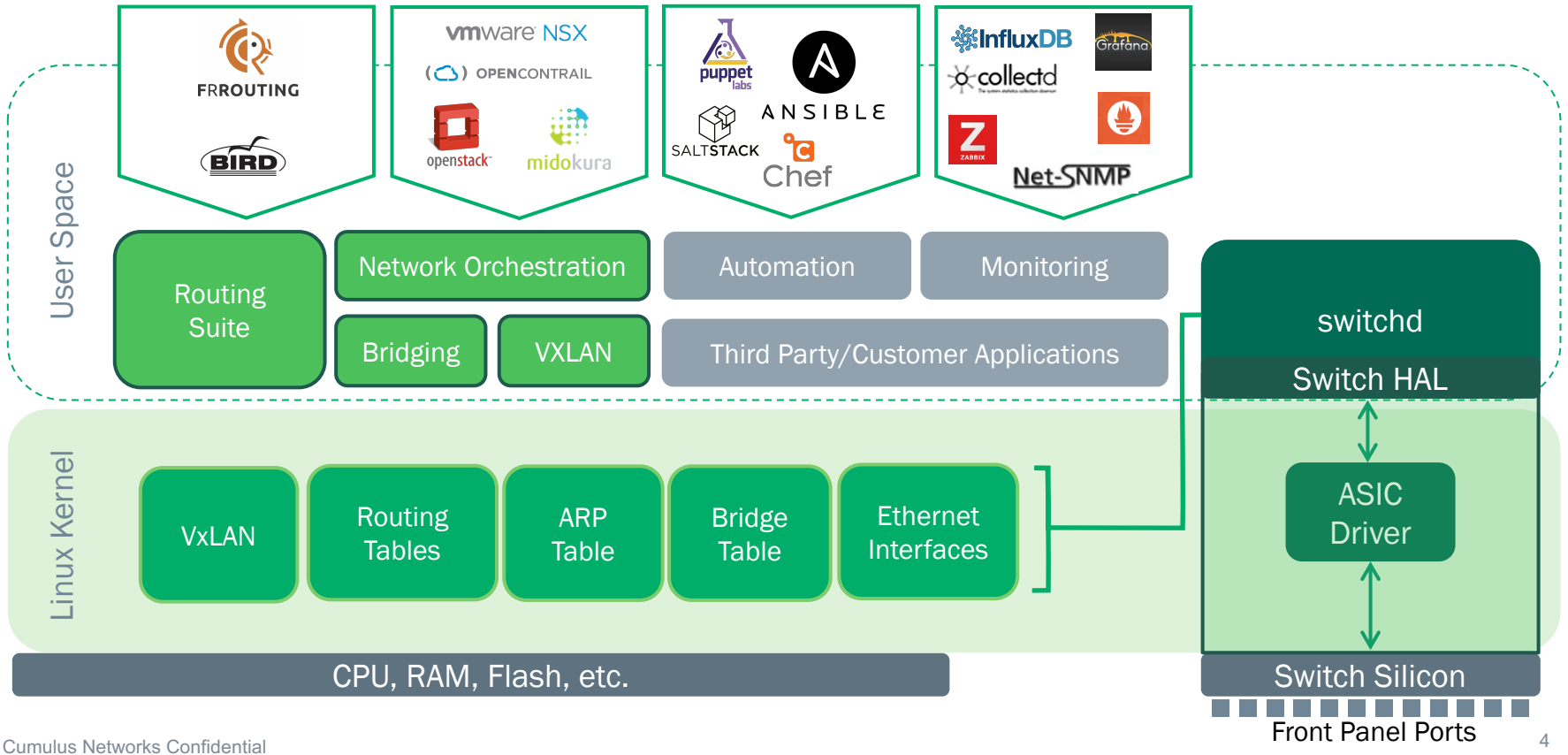
Cumulus Networks brings Web-Scale Networking

Unlocking the vertical network stack to build the modern data center

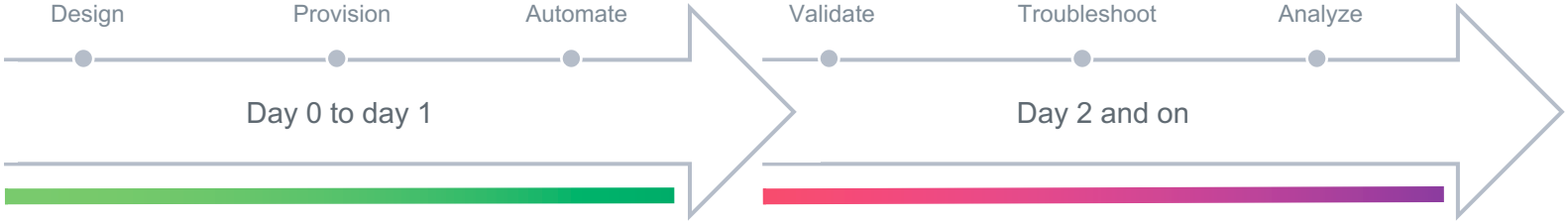


Cumulus Linux architecture

Uniform operating model – write any tools, use any apps



NetQ brings web-scale efficiencies to network operations



Cumulus Linux

Open Network Operating System

Open OS foundation of network
flexibility & choice of apps

Disaggregation of SW & HW
supply chain freedom

Speed of provisioning workloads
with automation at scale

Cumulus NetQ

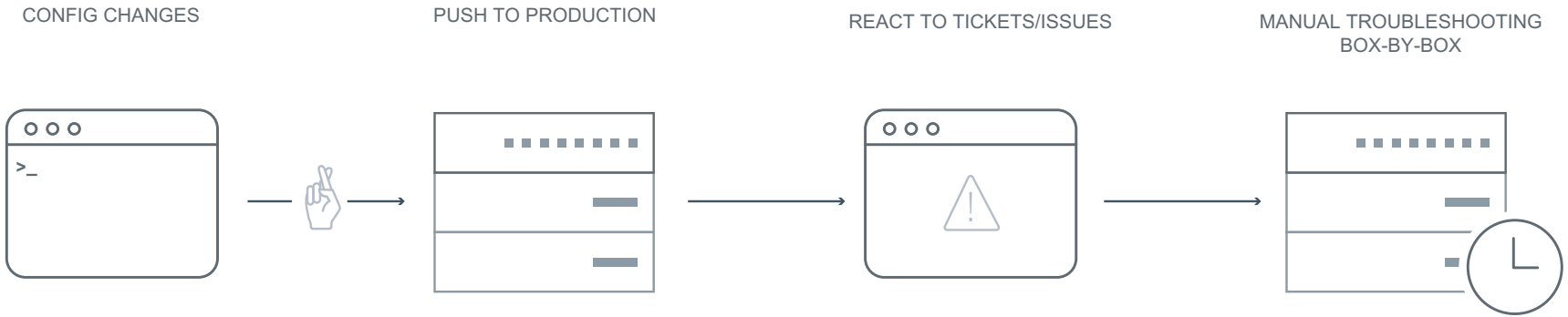
Telemetry-Based Fabric validation application

Validate changes
before & during production rollout

Precise fault location alerts
for rapid problem resolution

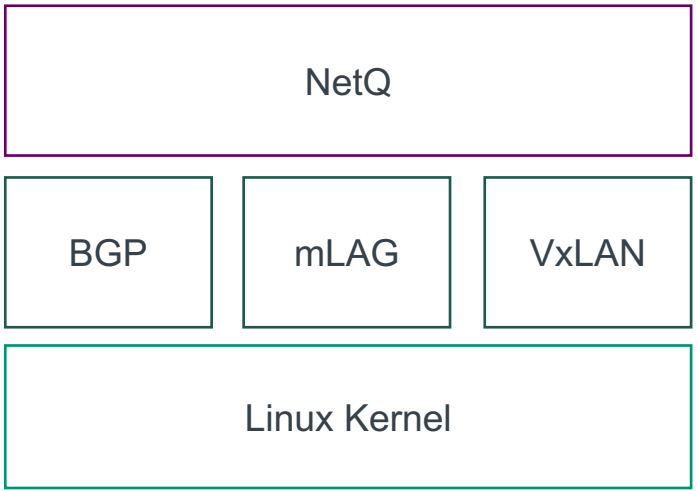
Time-machine diagnosis & troubleshooting

Can't keep up with the speed of automation



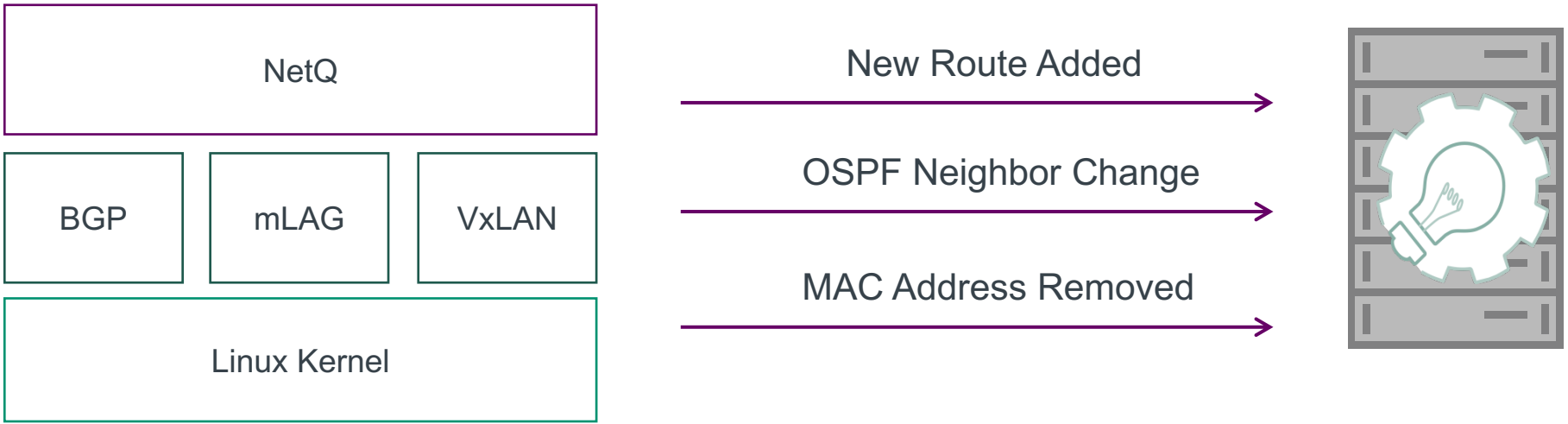
Existing tools do not offer closed-loop feedback

NetQ: How it Works



- NetQ Agent Subscribes to Linux Kernel Events
  - Interface State
  - MTU
  - Routes, MACs
- NetQ Agent Polls Routing Information
  - BGP and OSPF
  - Neighbor States
  - Error Conditions

NetQ: Fabric Change Log



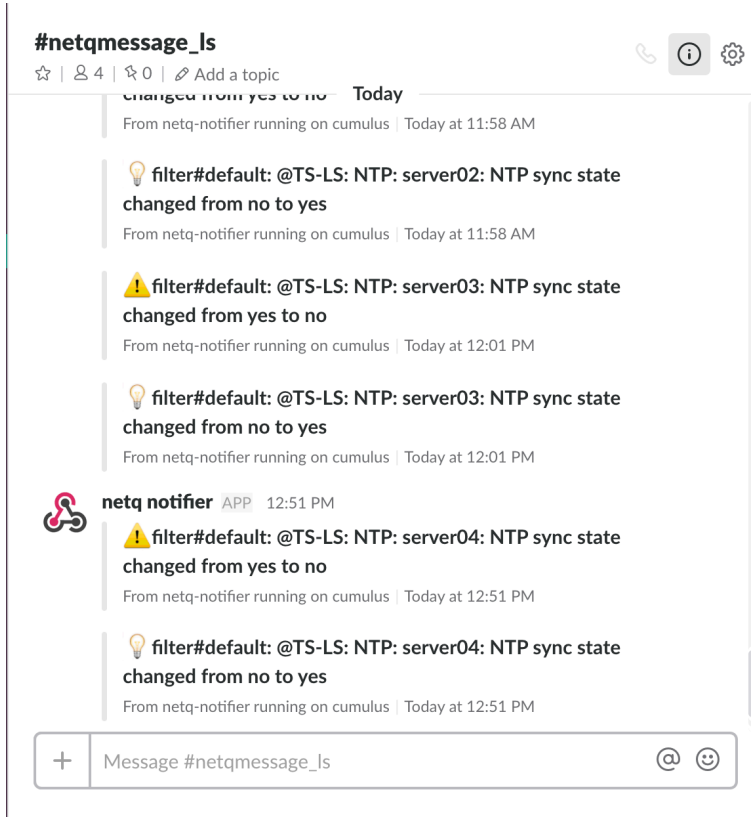
See state now or any point in the past

NetQ: Advanced Notification

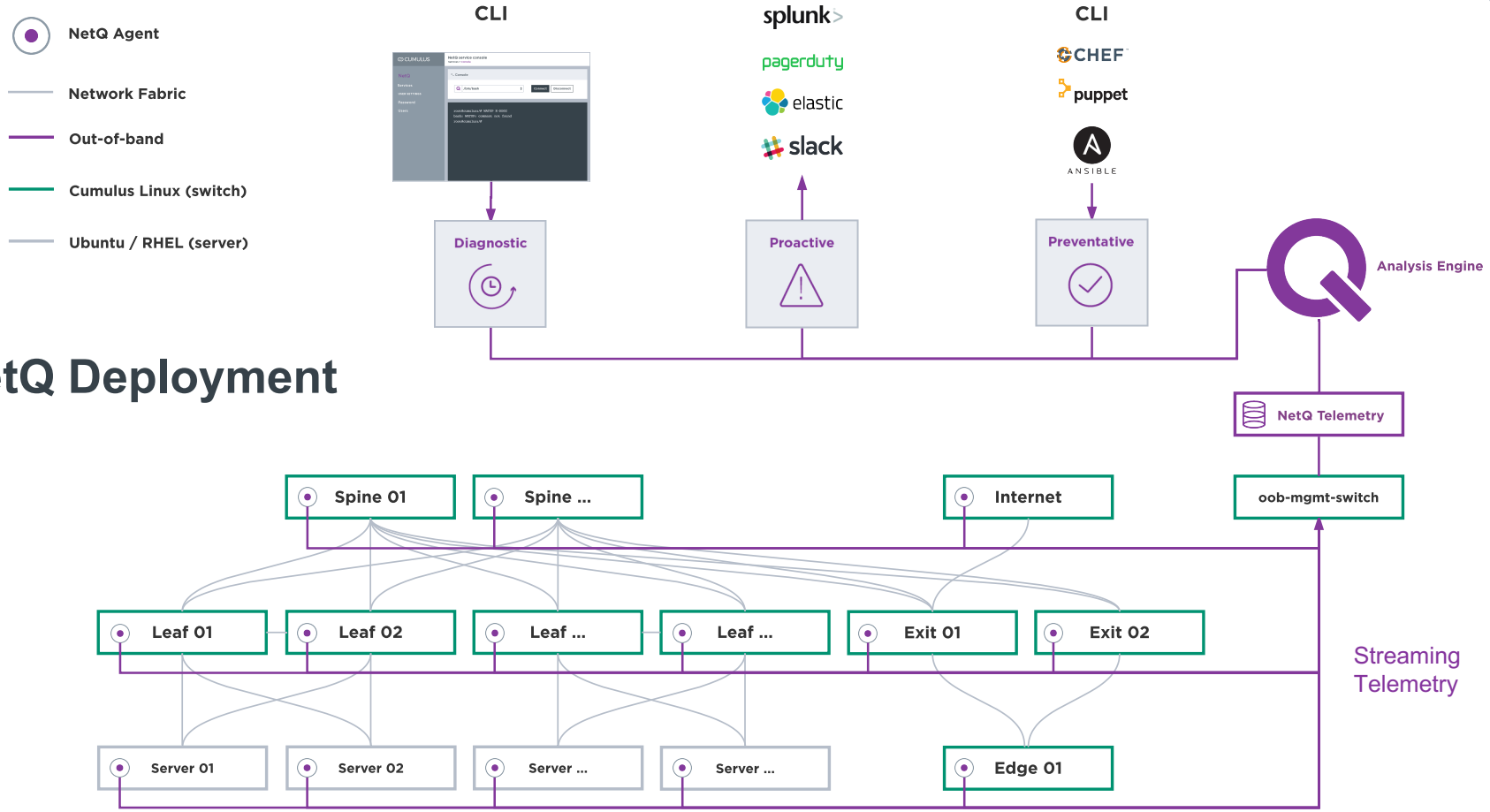
NetQ Notifier Service

Automatically Alert on Check Failures

- Syslog
- ChatOps (Slack)
- ELK
- Splunk



NetQ Deployment



Question

Who recently made a network change?

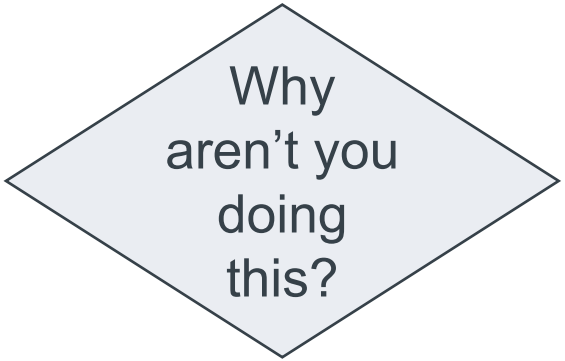
Question

Who had no issues during that change?

What is CI / CD?

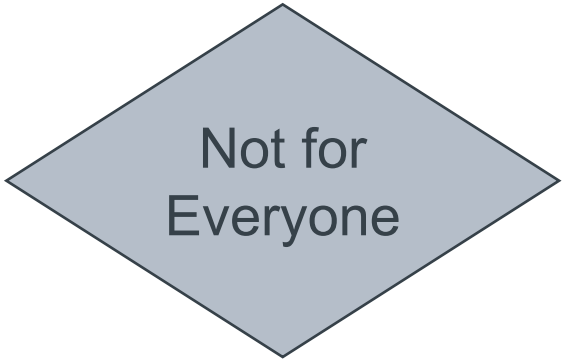
Continuous Integration (CI):

A system where all changes are automatically tested before being pushed to production or seen by others



Continuous Delivery (CD):

Built on a CI system where changes are automatically pushed to production after tests past, often multiple times per day



Types of tests

Linting

- Basic syntax checking

Unit Test

- Testing specific functions or chunks of code

Systems Test

- End to end testing
- Did the program do what I want?

Linting tests

Great tools for software, not so much for network

Mainly test yaml (Ansible)

Enforce a style

Create good practices

Find dumb errors early

```
./.gitlab-ci.yml
  6:4       error    wrong indentation: expected 2 but found 3  (indentation)
  10:4      error    wrong indentation: expected 2 but found 3  (indentation)
  12:7      error    wrong indentation: expected 5 but found 6  (indentation)
  15:4      error    wrong indentation: expected 2 but found 3  (indentation)
  17:7      error    wrong indentation: expected 5 but found 6  (indentation)
  20:1      error    too many blank lines (1 > 0)  (empty-lines)
```

Unit and system tests

Again, not a lot of great options

- Software light years ahead of us

Likely to be combined tests for networking

Multiple options, none perfect

- Stackstorm
- Roll your own (behave, python, ansible)
- Cumulus NetQ

Important to validate *state* not just config

- Is the interface up
- AND do I see an LLDP peer

Build tools

Build tools enable CI/CD Pipelines

- Magic duct tape

Common ones include

- Jenkins
- TravisCI
- Atlassian Bamboo
- GitLab CI

How it works

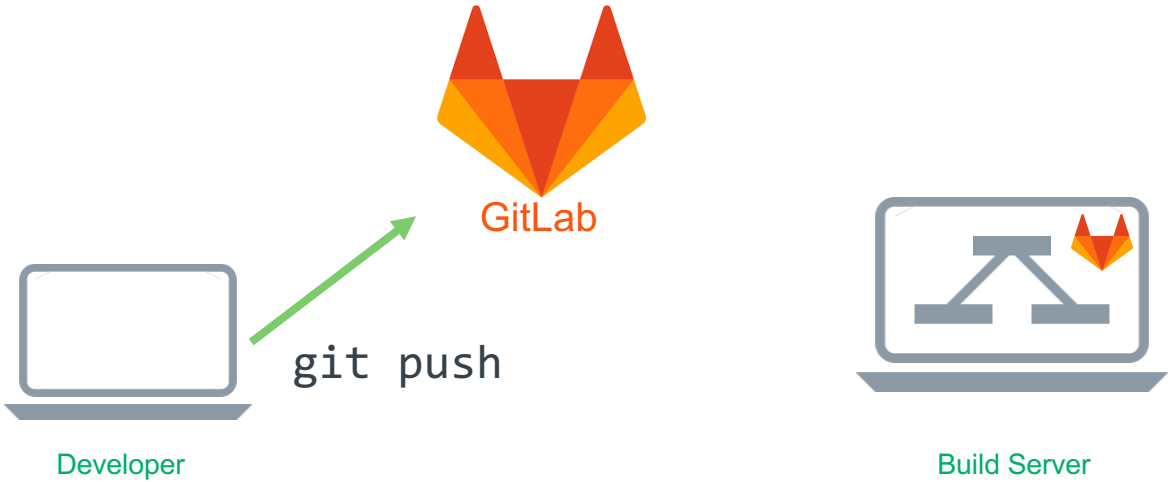
`git push` signals build tool

- Either github REST call (Jenkins) or agent (Gitlab)

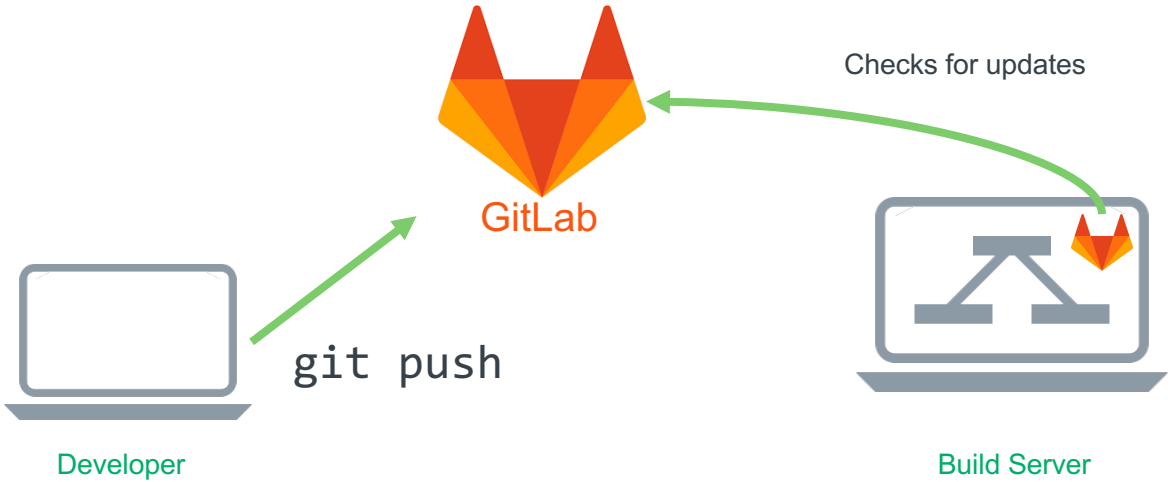
Build tool runs things

- Some built in macros for programming languages
- Custom scripts for the rest (networking)

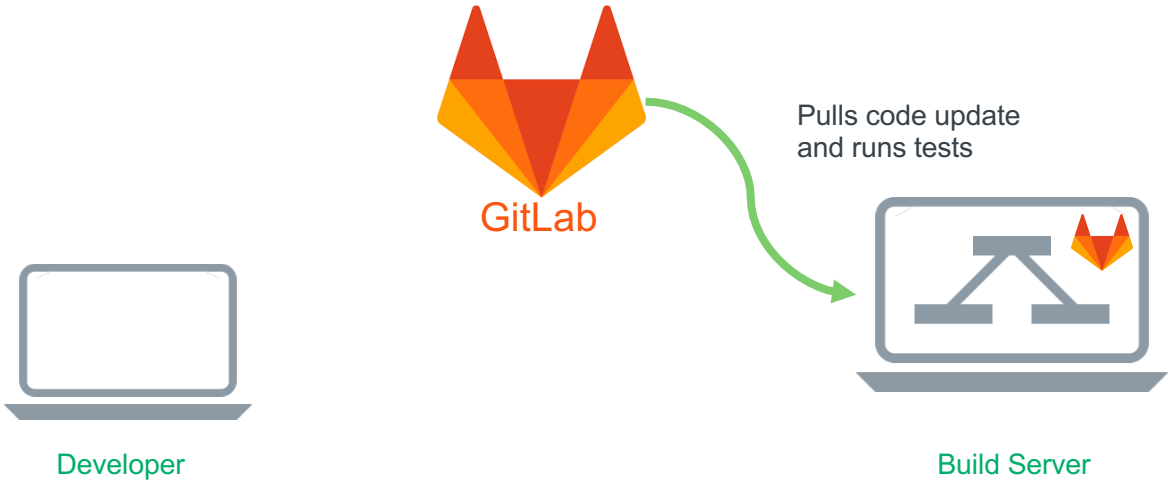
Gitlab build process



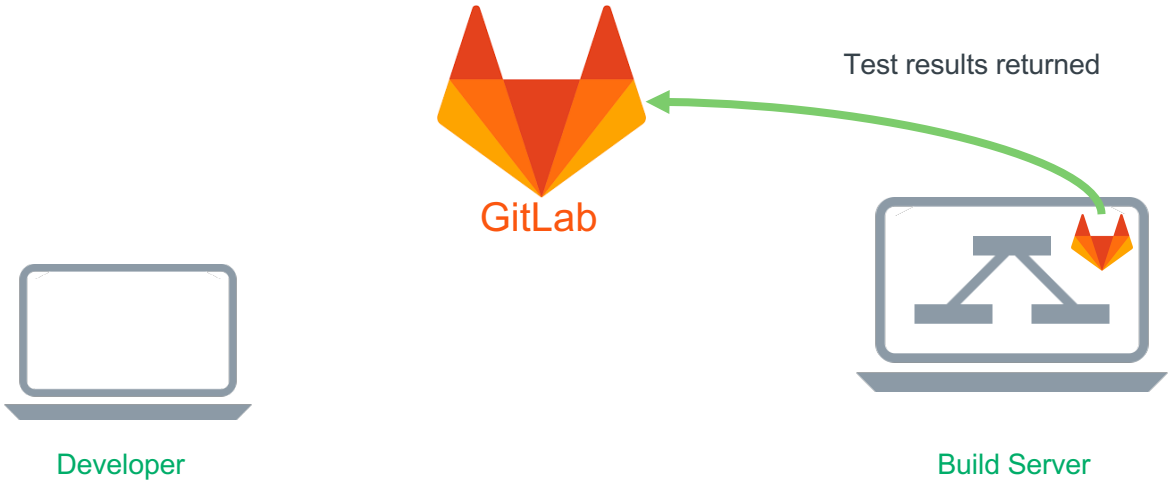
Gitlab build process



Gitlab build process



Gitlab build process



.gitlab-ci.yml
Virtualizing your network

Stages, before_script, after_script

- Stages run in order on success
- *_script run at each stage

Script defines things to do

- Each script step is run on the build server
- Each step is a unique session
- Simple bash scripts are like duct tape

Before/After scripts setup or cleanup

- `vagrant destroy -f`

```
---
stages:
  - staging
  - production

staging:
  tags:
    - staging
  before_script:
    - cd evpn
  stage: staging
  script:
    - 'ansible-playbook deploy.yml'
    - sleep 25
    - netq check bgp
    - netq check mtu
    - netq check vxlan

production:
  tags:
    - production
  before_script:
    - cd evpn
  stage: production
  when: manual
  script:
    - 'ansible-playbook deploy.yml'
    - sleep 10
    - netq check bgp
---
```

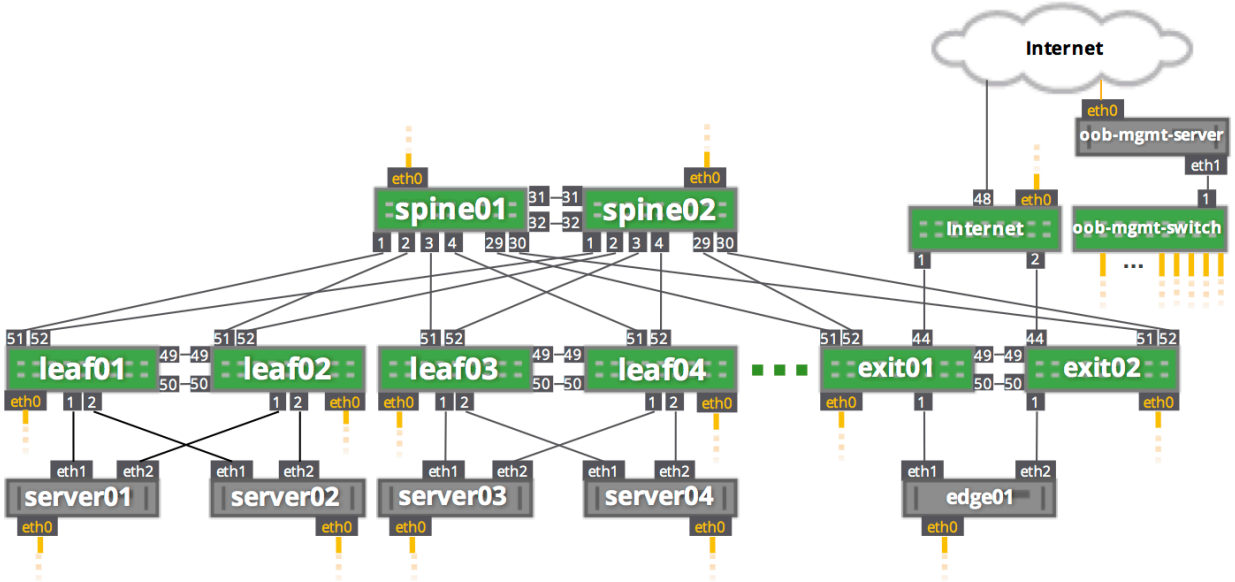
Testing and experimentation
Virtualizing your network

Virtual Environment

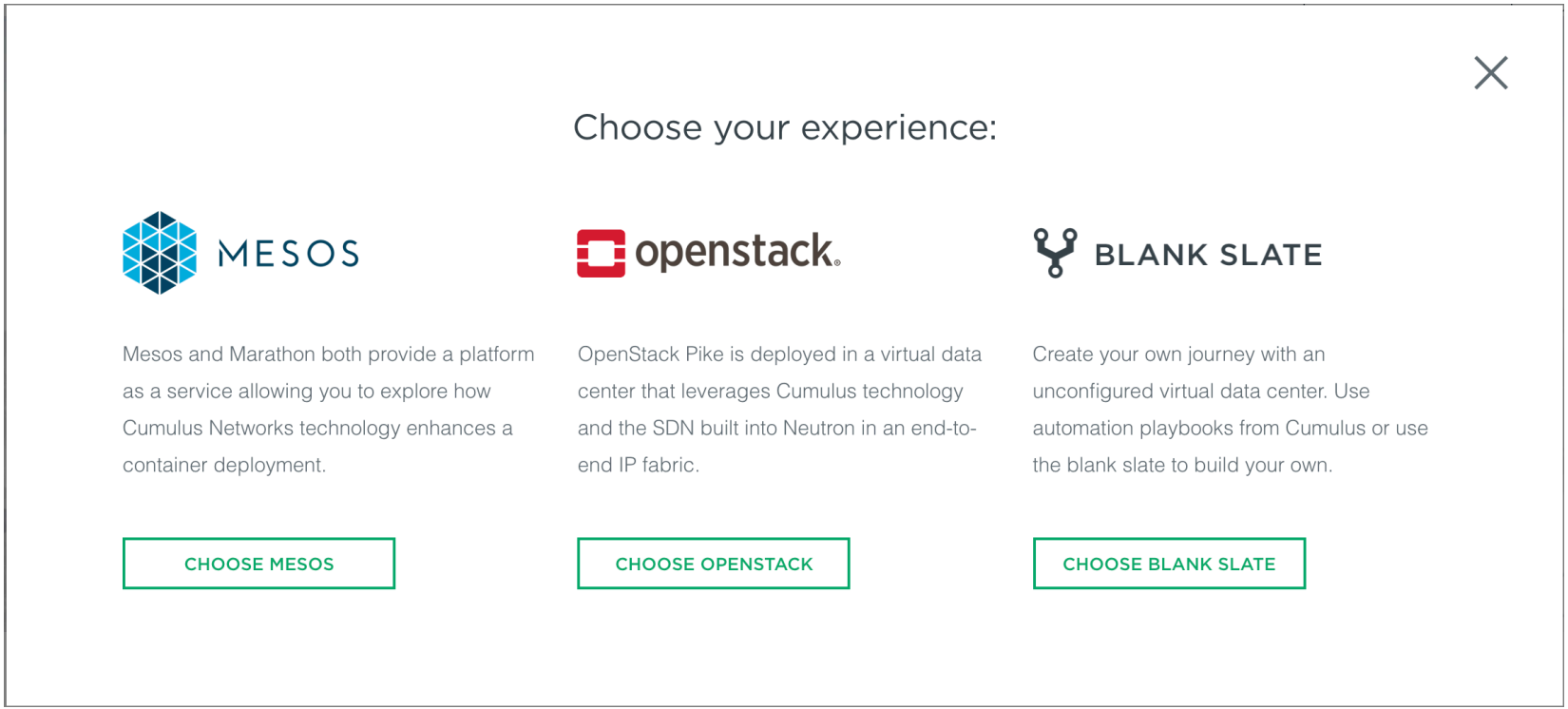
- Cumulus VX
- Familiarizing with the OS

Larger scale

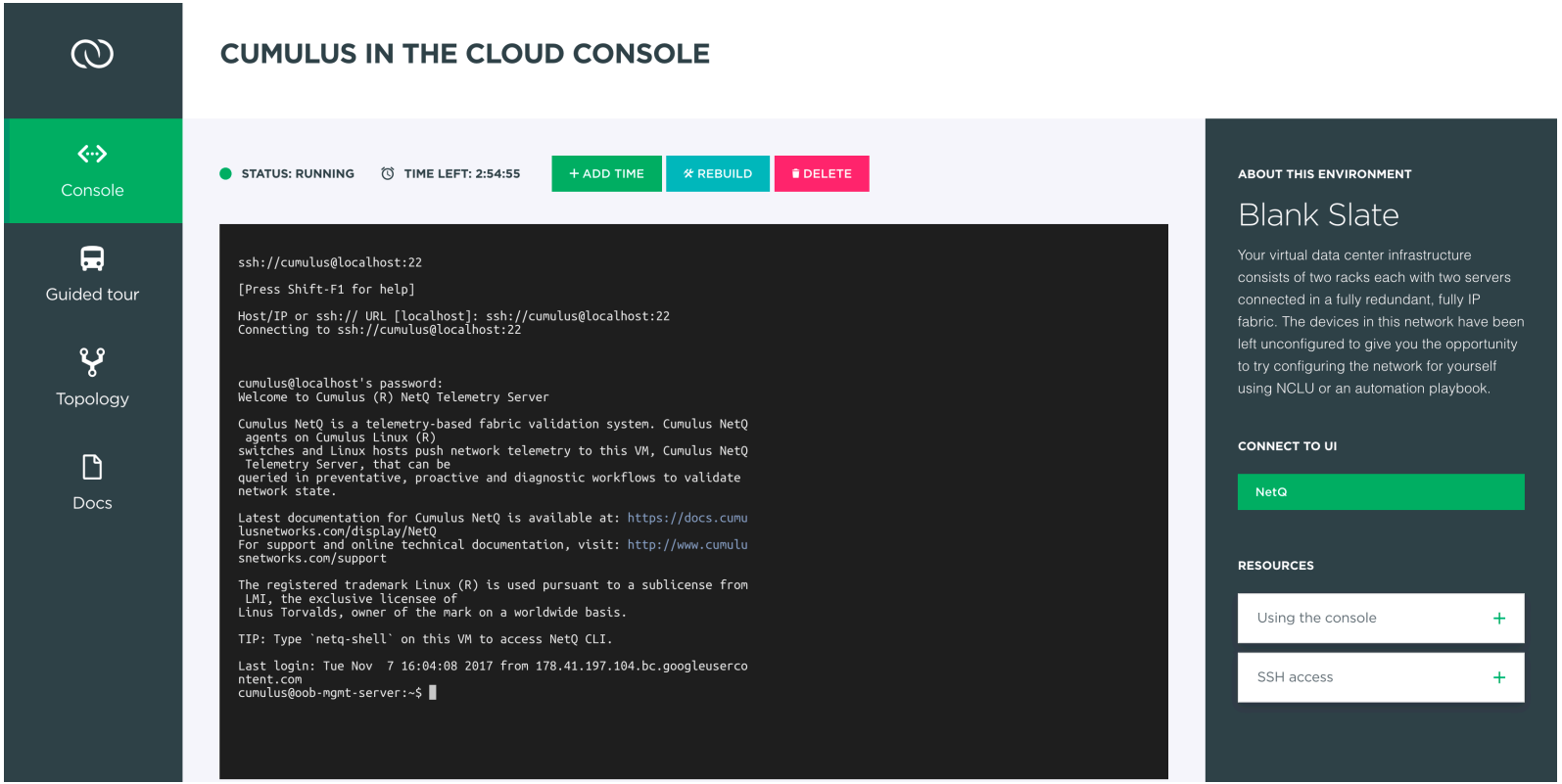
- Vagrant
- Reference Topology
- Topology Generator



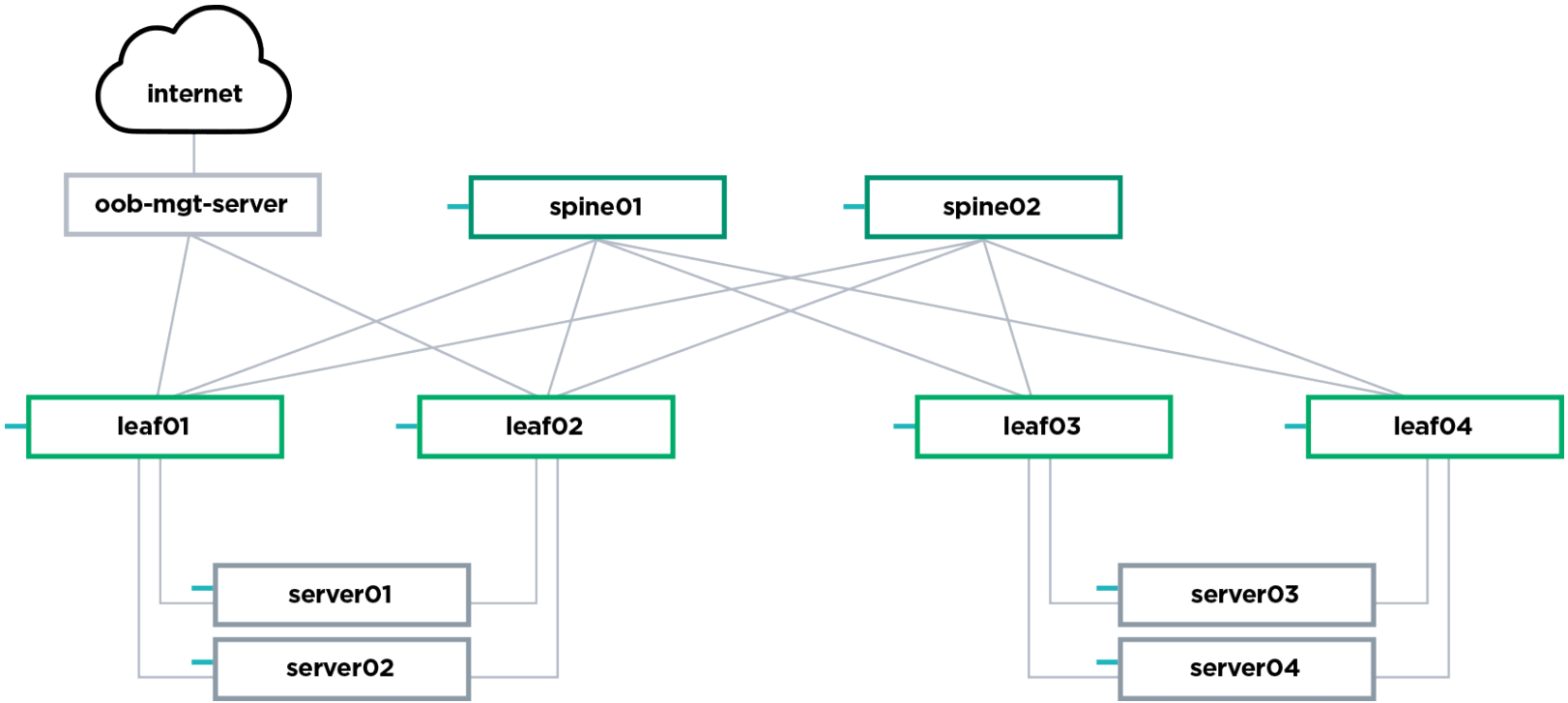
Cumulus in the Cloud



Cumulus in the Cloud

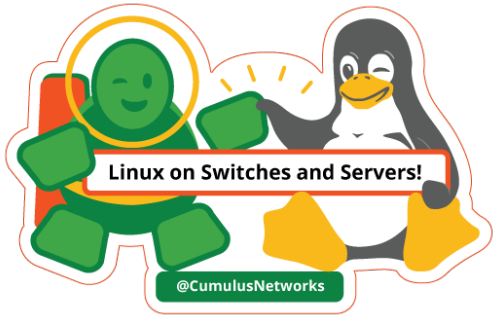


Cumulus in the Cloud



CI / CD Demo

CUMULUS



Thank you!