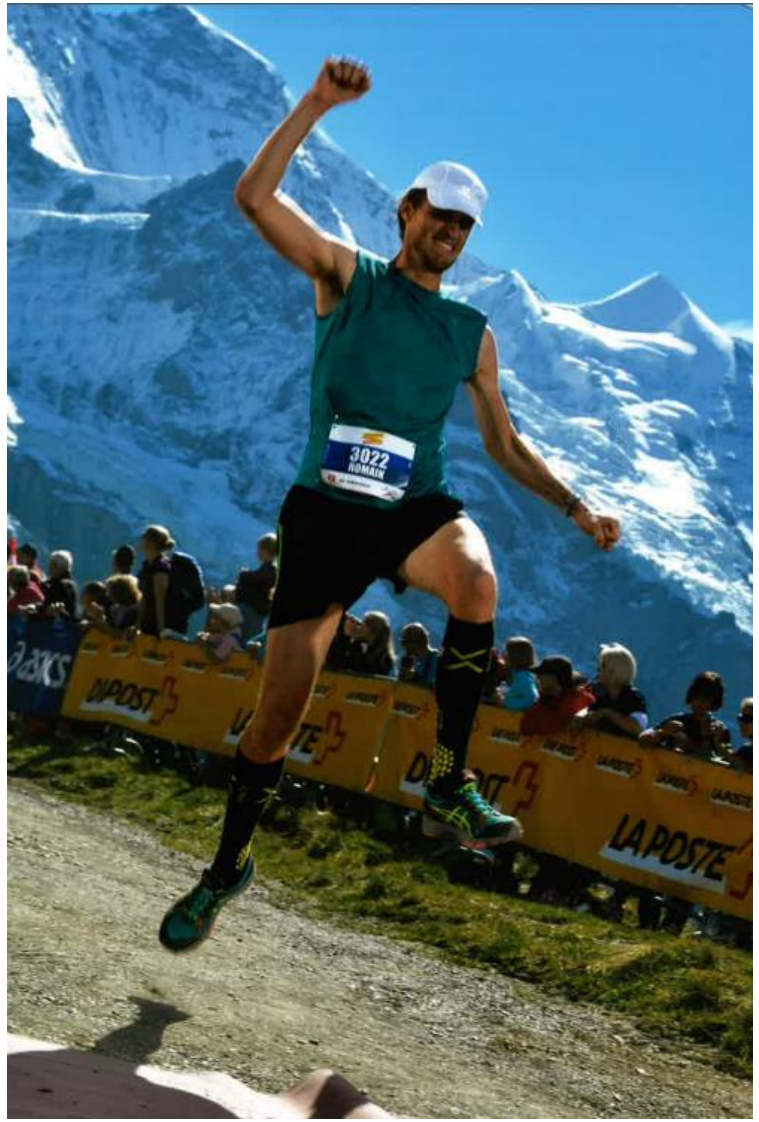
AnsibleFest 2017.06.22 - London

Network CI/CD Using Ansible and GitLab – Romain Aviolat

Romain Aviolat

Cloud Infrastructure Expert

- Swiss product
- Mountains
- Reading, Cinema, Music
- Open Source advocate
- Void warranties
- Hardware hacking



Agenda

- Some context
- Whitebox networking
- Zero-Touch-Provisioning
- Ansible
- GitLab
- Ansible and Gitlab
- Virtual network environment
- Live demo (:

Kudelski Group

- +60 years
- +3K employees on 5 continents
- 200M+ annual R&D investment

- DigitalTV (Content protection)
- Public Access
- Cyber Security

en.wikipedia.org/wiki/Kudelski_Group

Goals

- Design / Build / Operate the foundation of our new IT infrastructure
  - Private-cloud
  - Data-lake
- Must be flexible and aaS as possible for R&D teams
- Silo-less mode
- Stable enough for production workloads

The team

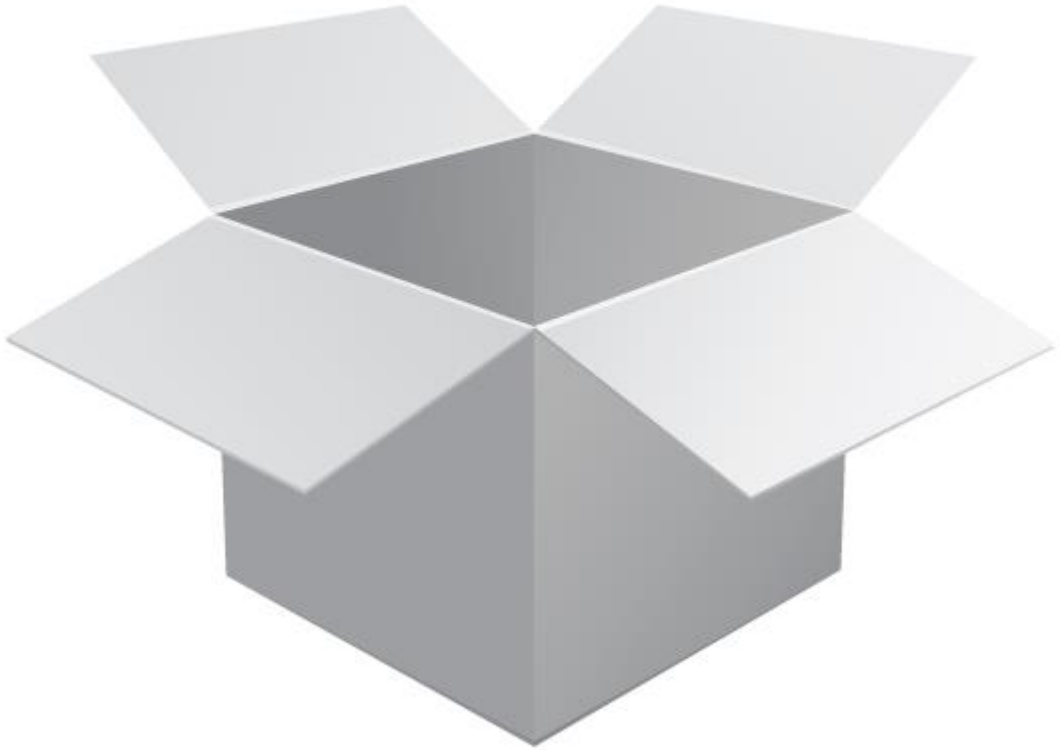
- I’m presenting this on behalf of my team
- Initially a 2.5 person initiative (tech-side)
  - Benoît Knecht (amongst others our DevOPs / Linux / OpenSource Evangelist)
  - François Deppierraz (aka **the** OpenStack guy) (50%)
  - Me
- Has grown a bit since

Self-Imposed challenges

- We had the opportunity to
  - design /
  - build /
  - operate something from scratch, with no history
- Go further than traditional configuration management
- Apply this model also to the network equipment
  - Get away from the traditional model

Where are we

- Started the project 1y ago
- Deployed this model in two DCs around the world
- We will replicate this setup few times more
- ~1y of experience



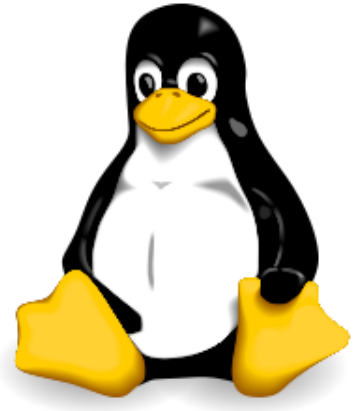
Whitebox networking

- Keystone of our infrastructure
- The network revolution **finally** arrived (like for the servers 15y ago)
- Hardware decoupled from the SW
  - No that’s not weird, remember Oracle, IBM, … ?
- Allows us to provision the network equipment like any other servers
- Demystifies networking by using simple and standards designs

What / How ?

- Choose a HW vendor
  - DELL, Mellanox,
  - Penguin Computing, HP
  - …

- Choose a SW distribution
  - Cumulus Linux
  - SONiC (Microsoft Azure)
  - PICA8
  - IPinFusion



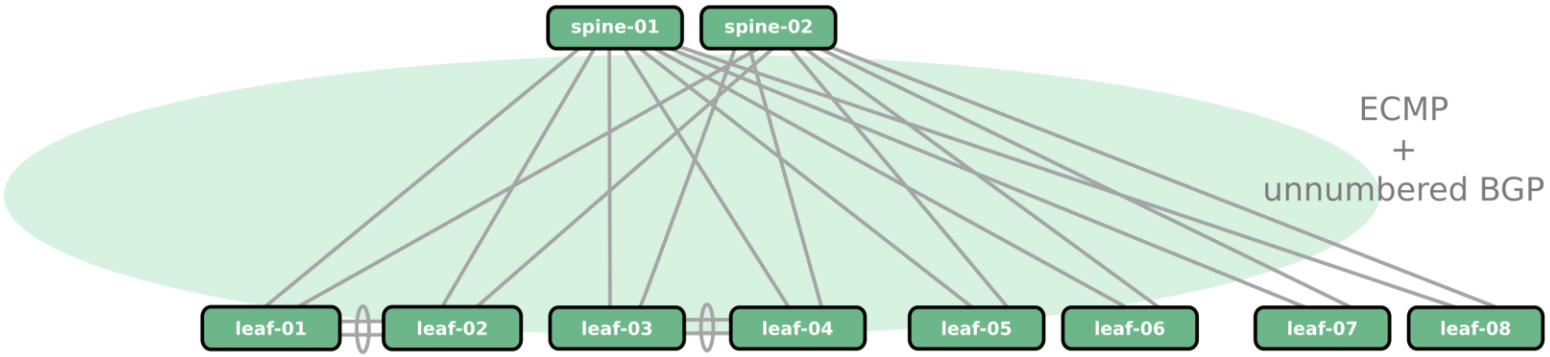
Why CumulusLinux in our setup

- Most advanced distro at that time
- Similar to the OS running on our servers
  - Debian-based
- Awesome and creative engineers (binary compatibility)
  - Evangelize the DevOPs model

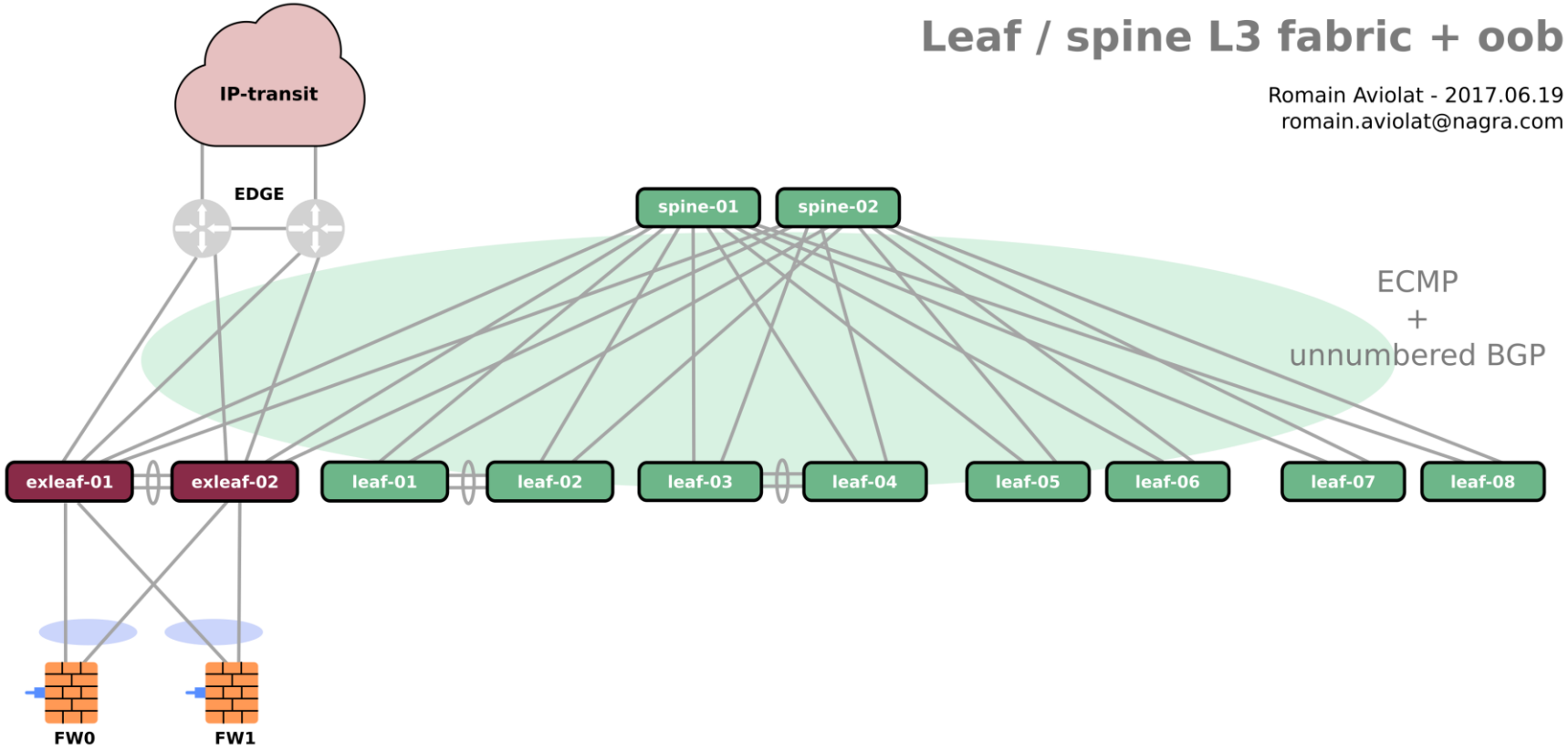
Infrastructure big picture

Leaf / spine L3 fabric + oob

Romain Aviolat - 2017.06.19
romain.aviolat@nagra.com



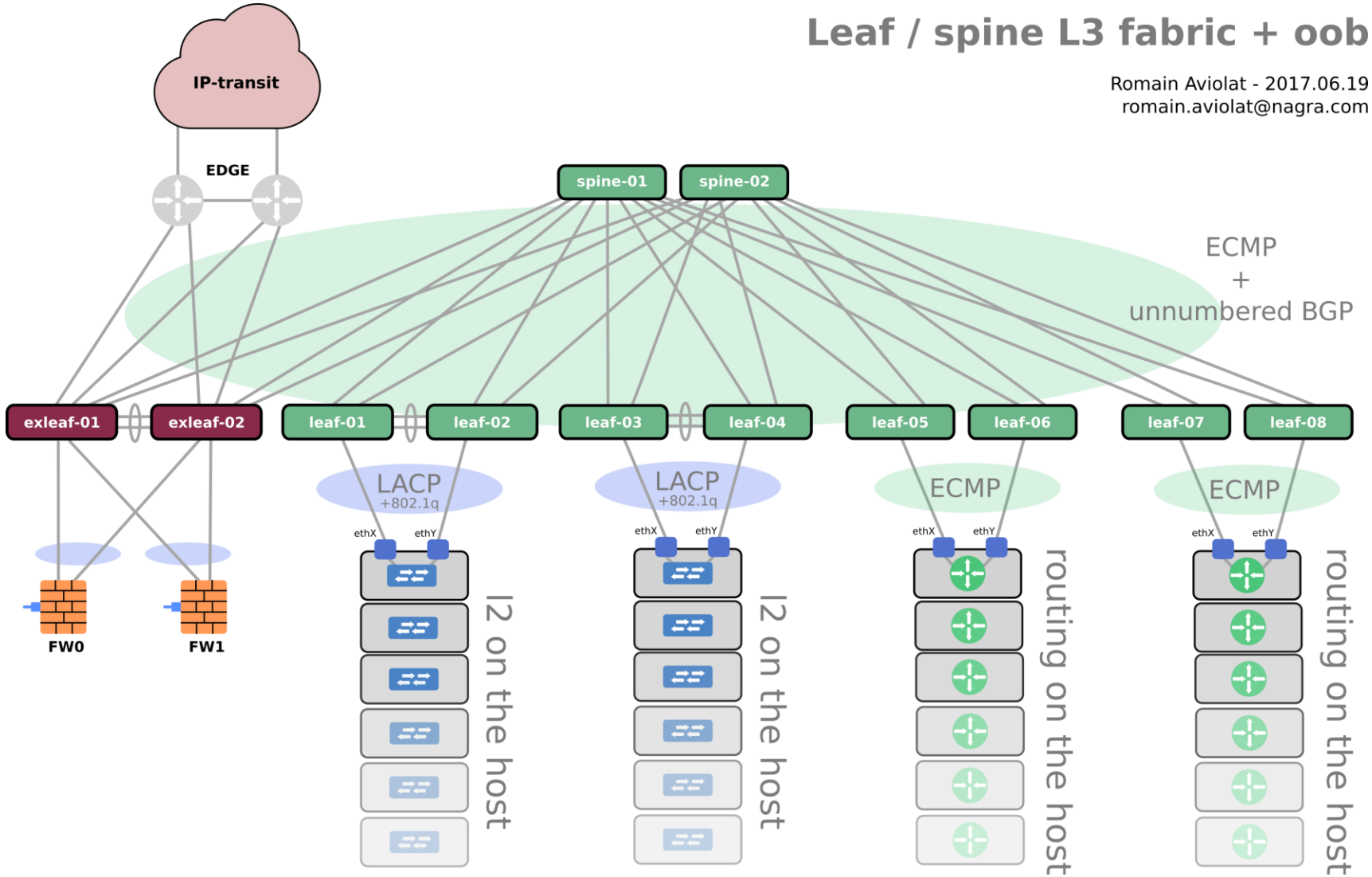
Infrastructure big picture



Infrastructure big picture

Leaf / spine L3 fabric + oob

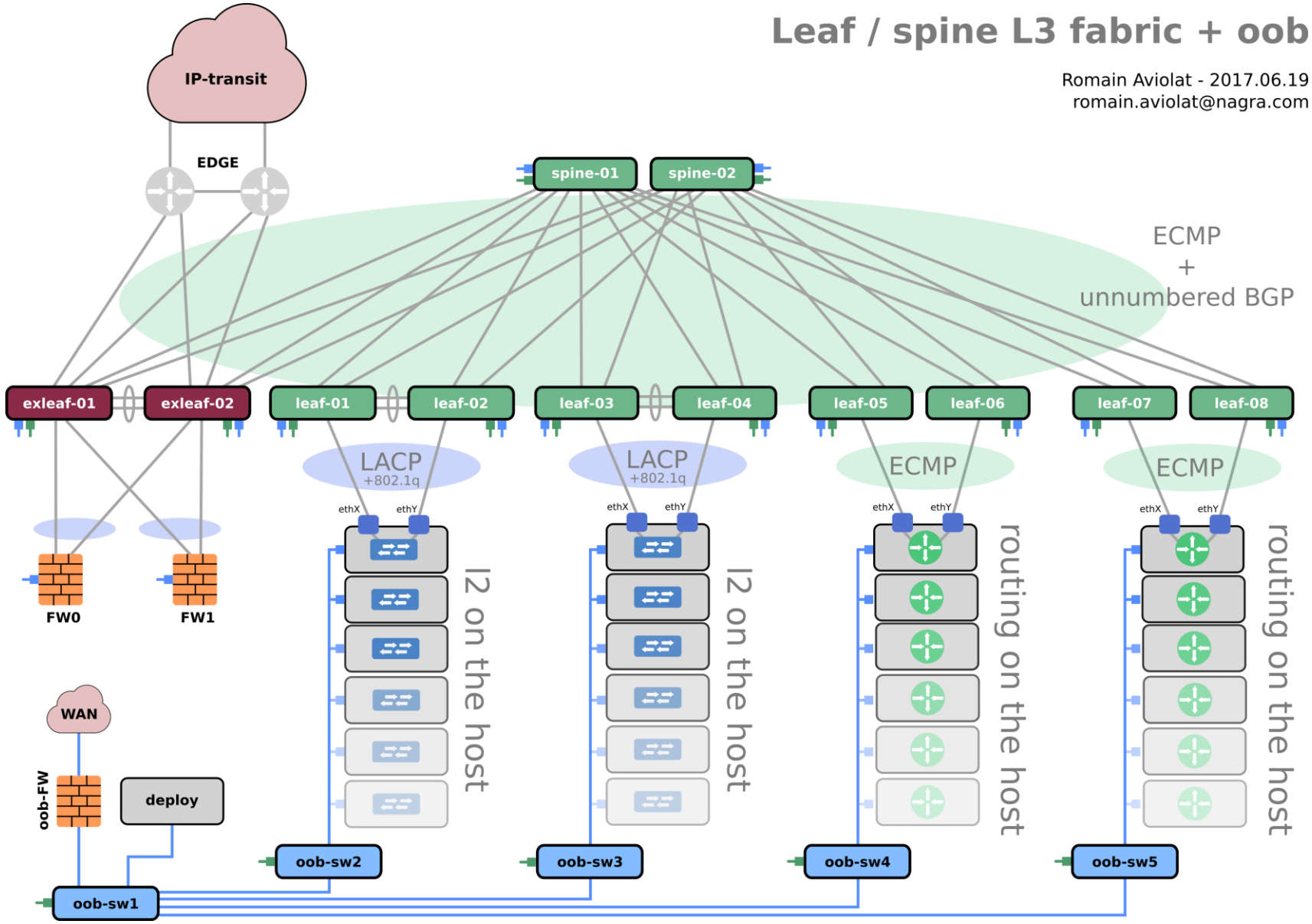
Romain Aviolat - 2017.06.19
romain.aviolat@nagra.com



Infrastructure big picture

Leaf / spine L3 fabric + oob

Romain Aviolat - 2017.06.19
romain.aviolat@nagra.com

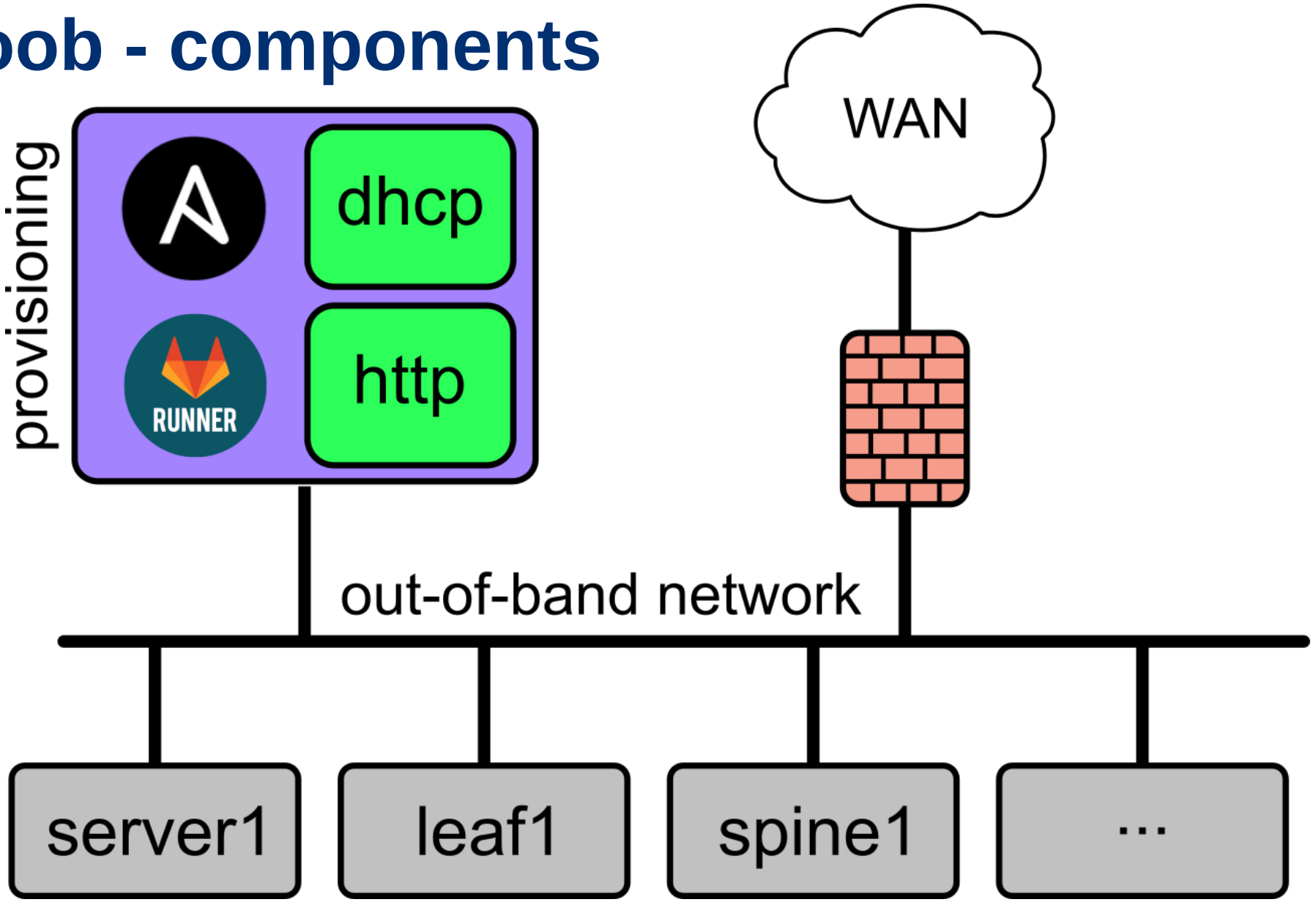


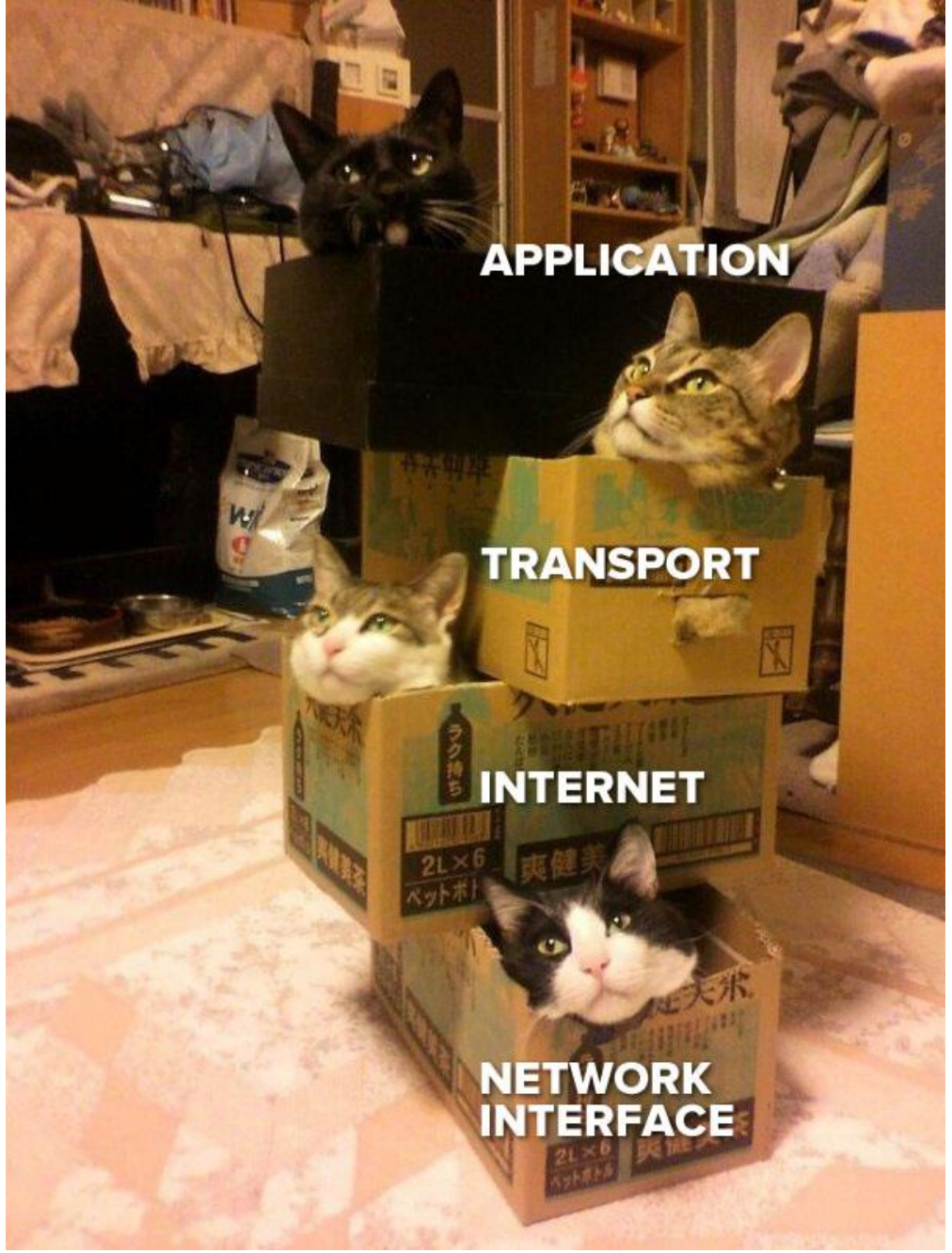
ZTP - ZeroTouchProvisioning

ZTP

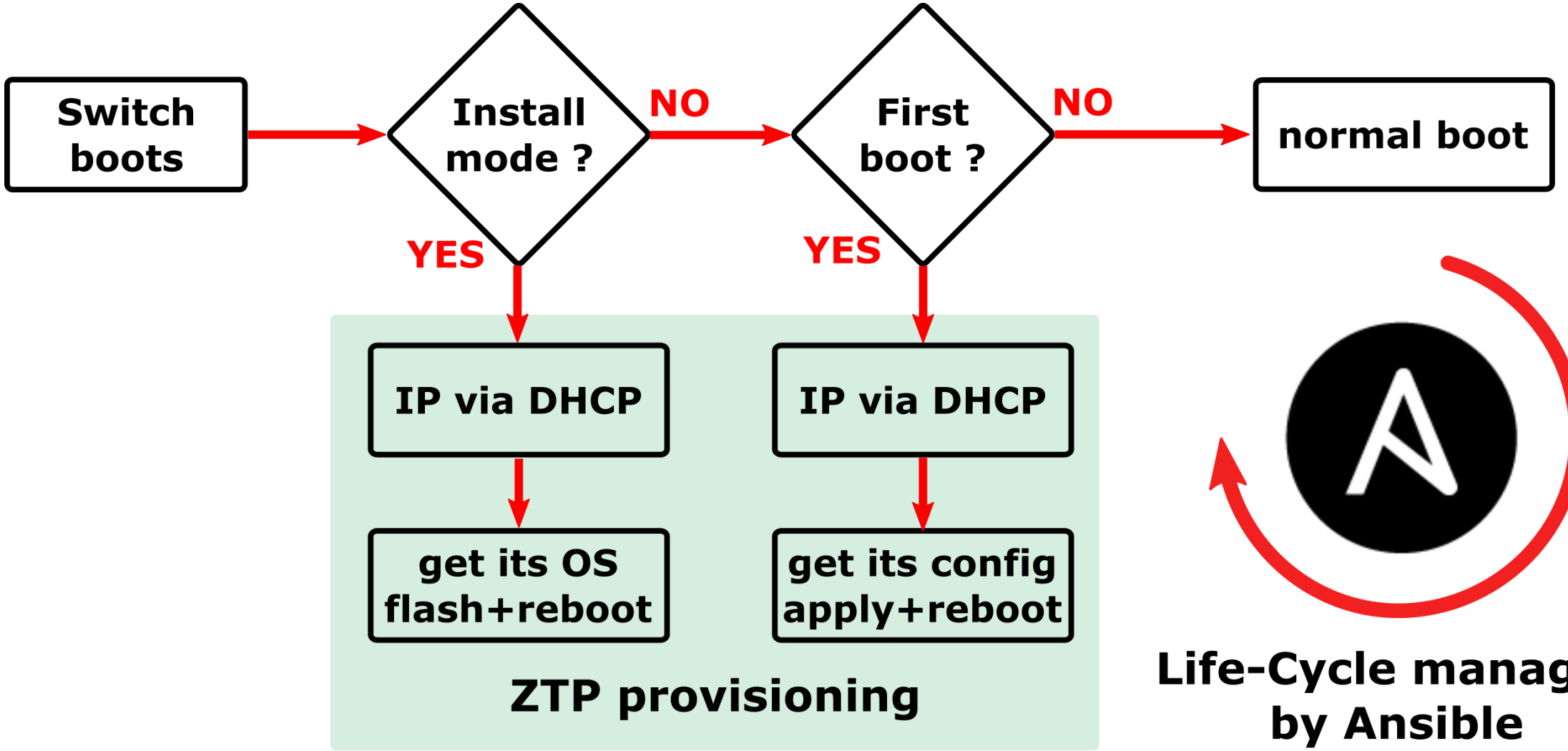
- Building block of our network infrastructure
- Replace initial manual provisioning
  - Install latest image
  - Deploy base config
- It’s **not** a config management tool
- Does not yet support unboxing and racking

ZTP / oob - components

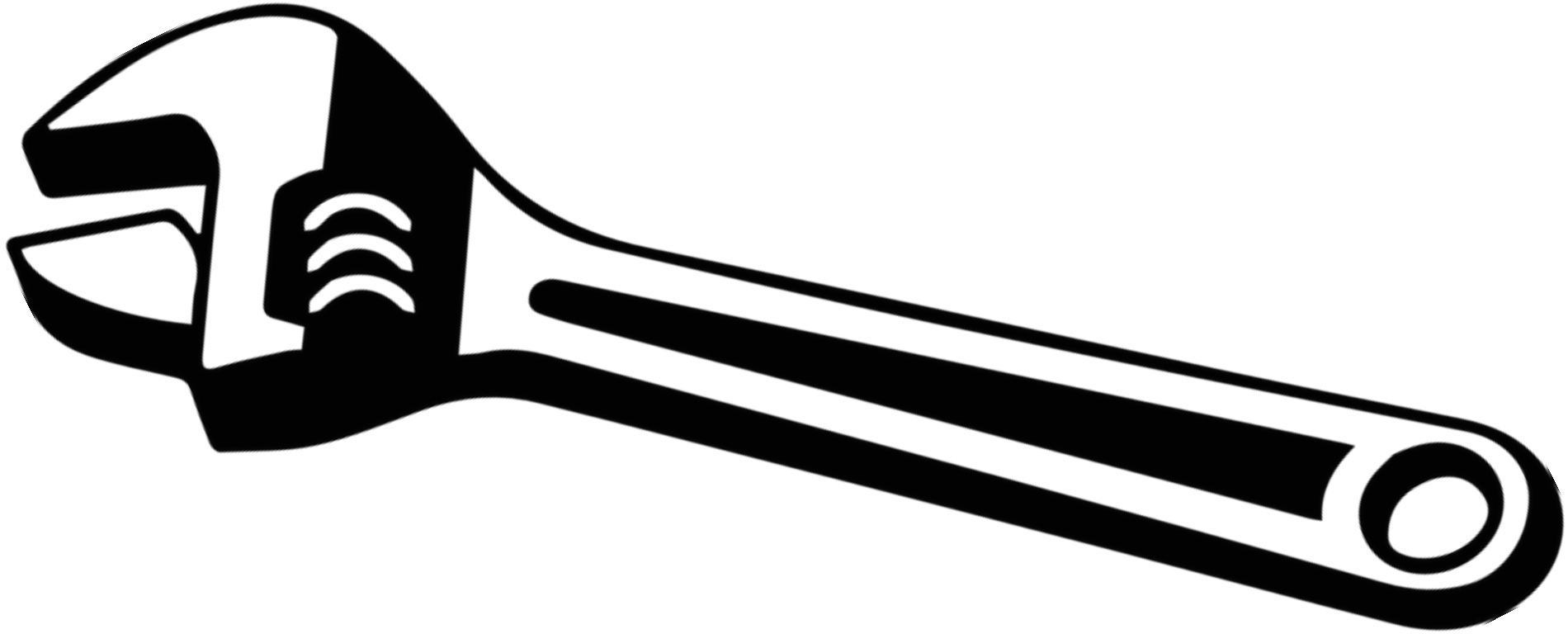




Provisioning work-flow



Life-Cycle managed by Ansible



Tooling

Infrastructure as code + Automation

- Infrastructure is deployed using automation
- Code is versioned
- Enforce infrastructure compliance
- Common practice in the sysadmin world for years (Not that much for the network infrastructure)

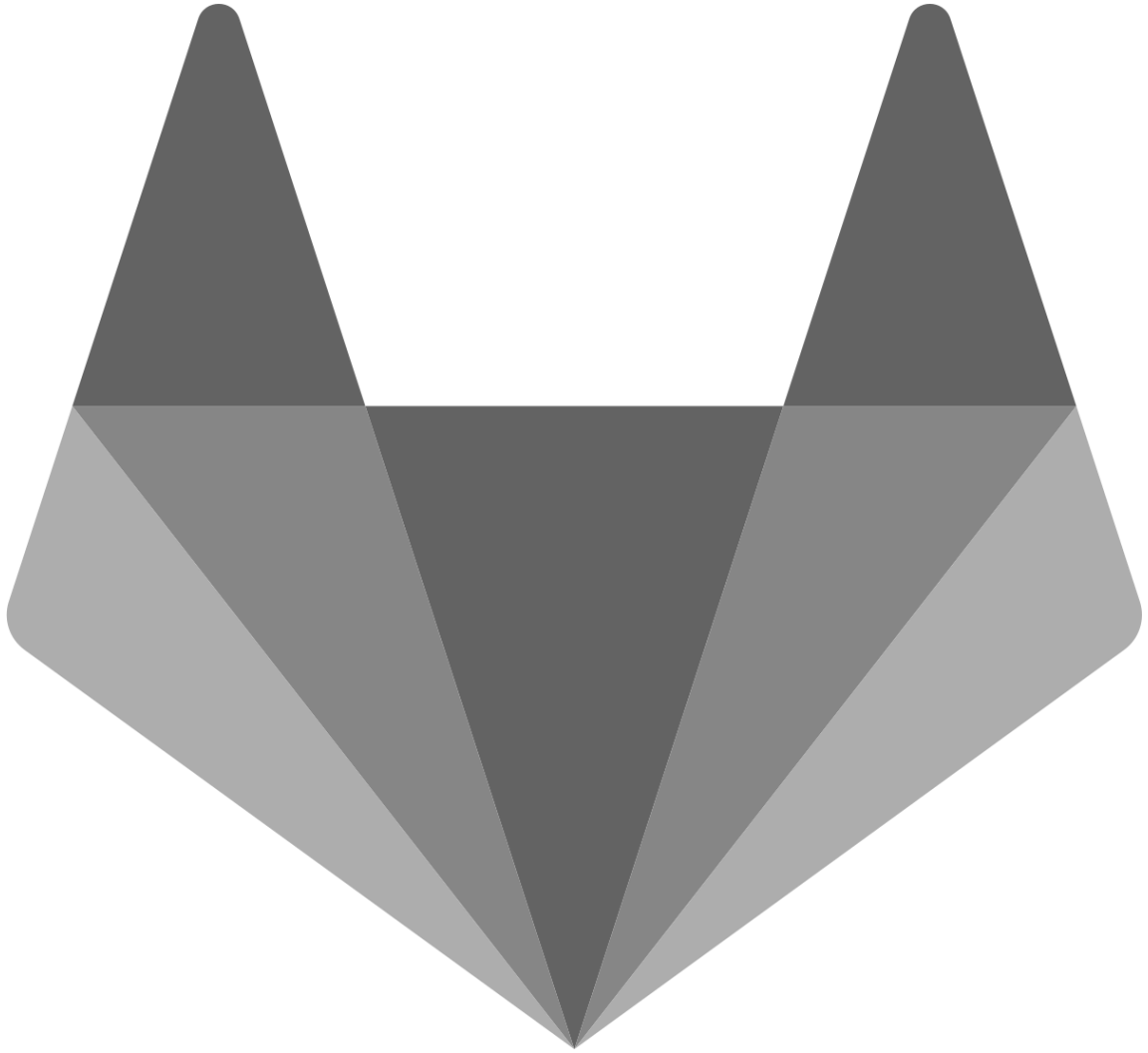
Ansible role in our setup

- Jinja templates for text-file configurations
  - Network interfaces
  - Quagga (routing engine)
- Manage all the OS settings
  - Users, dns, ntp, sshd, ...
- Idempotent setup
  - Our code is reapplied continuously

```
{% for vrfname, vrfinfo in fabric.vrfs.items() %}
router bgp {{ fabric.asn }} vrf {{ vrfname }}
{% if bgp_networks is defined %}
{% for net in bgp_networks %}
 network {{ net }}
{% endfor %}
{% endif %}
 bgp router-id {{ fabric.router_id }}
 bgp bestpath as-path multipath-relax no-as-set
 neighbor fabric peer-group
 neighbor fabric remote-as external
 neighbor fabric description Internal Fabric Network
```

So far we have…

- All our Ansible code versioned inside Git
- The ability to redeploy from scratch the infrastructure using
  - ZTP
  - Ansible



Pushing things further

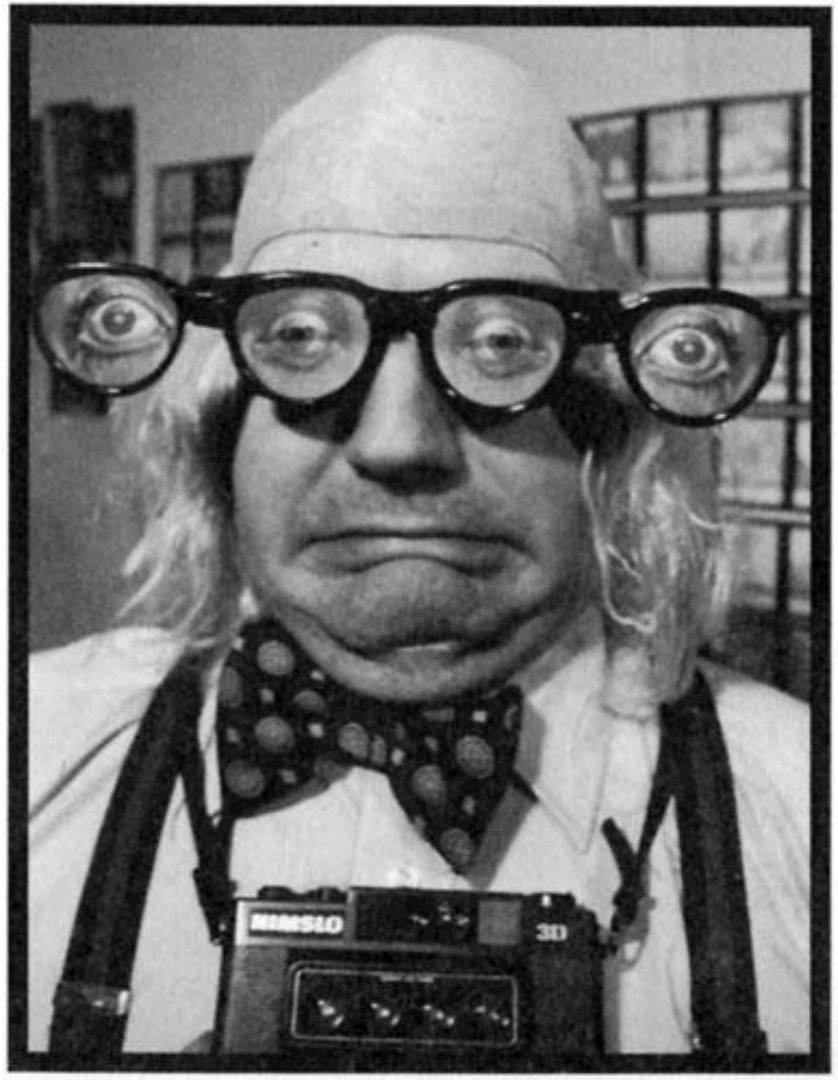
That’s a good start but …

- Where to execute my Ansible code from ?
- What happens if someone executes an older code version ?
- How do I manage code-changes ?

SW dev. best-practices

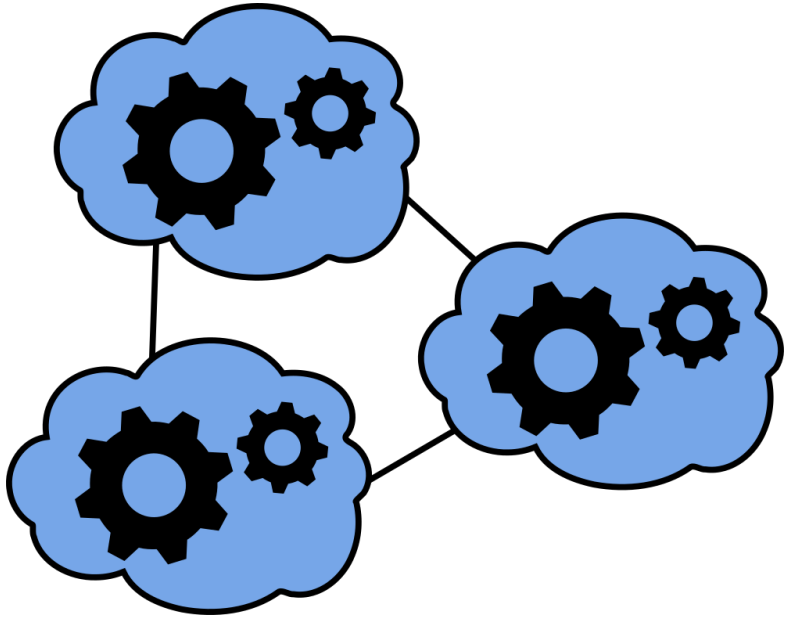
We solve these problems by applying the same paradigms as in software dev.

- Multiple environments
  - Dev, Staging, Production
- Code review / Four-eyes review
- CI / CD pipeline



Staging / LABs

- Complicated to achieve with physical hardware
- With virtualization it’s now “easy” to simulate a complete network environment
- KVM, Virtualbox / Vagrant
- Some vendors directly provide a VM for their OS



GitLab

- Open source GitHub-like
- Git repositories
- On or Off-premises
- Built-in CI/CD (à la Jenkins)
- Issue tracking

Components big picture

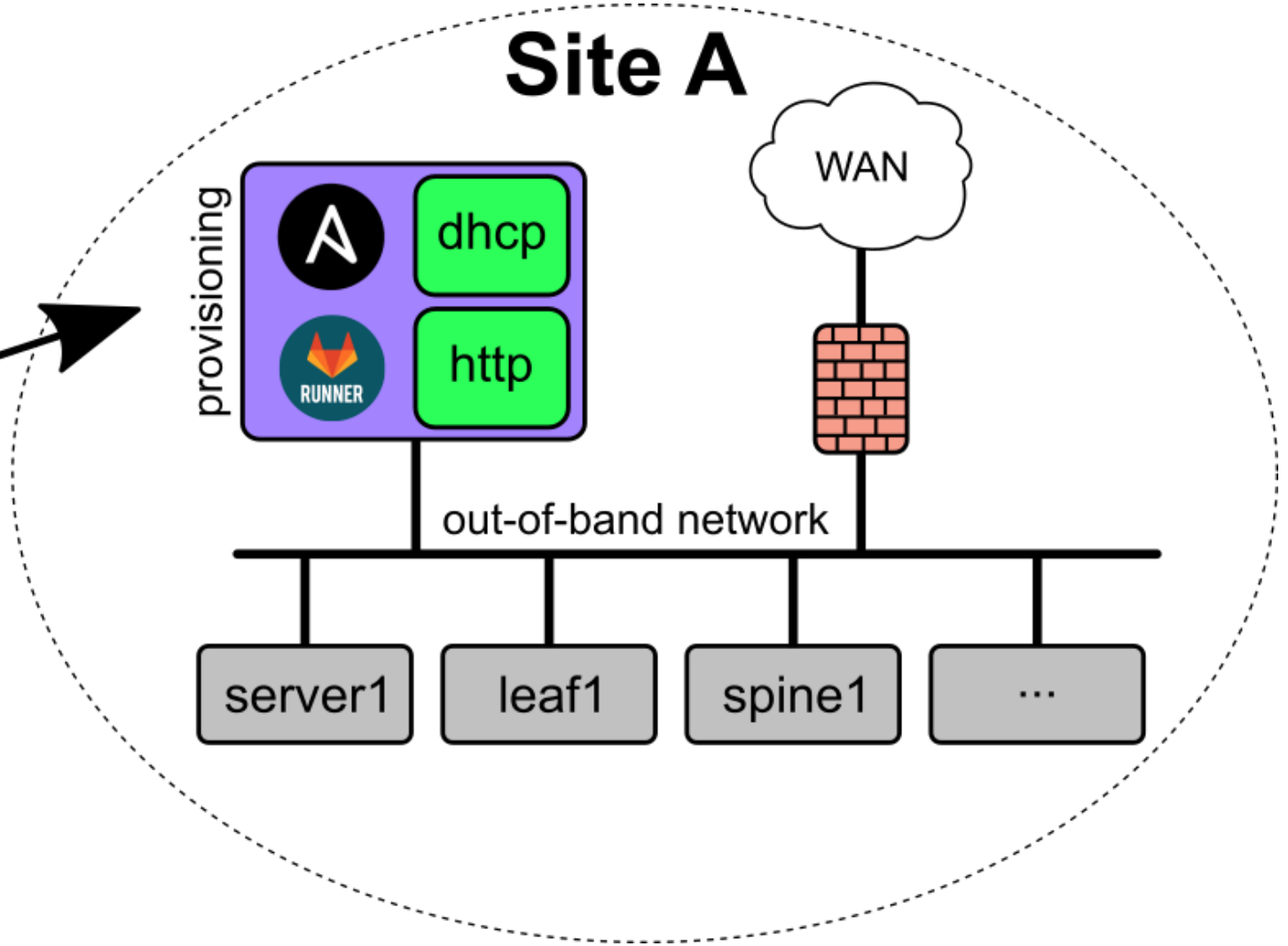
Git repository

Image registry

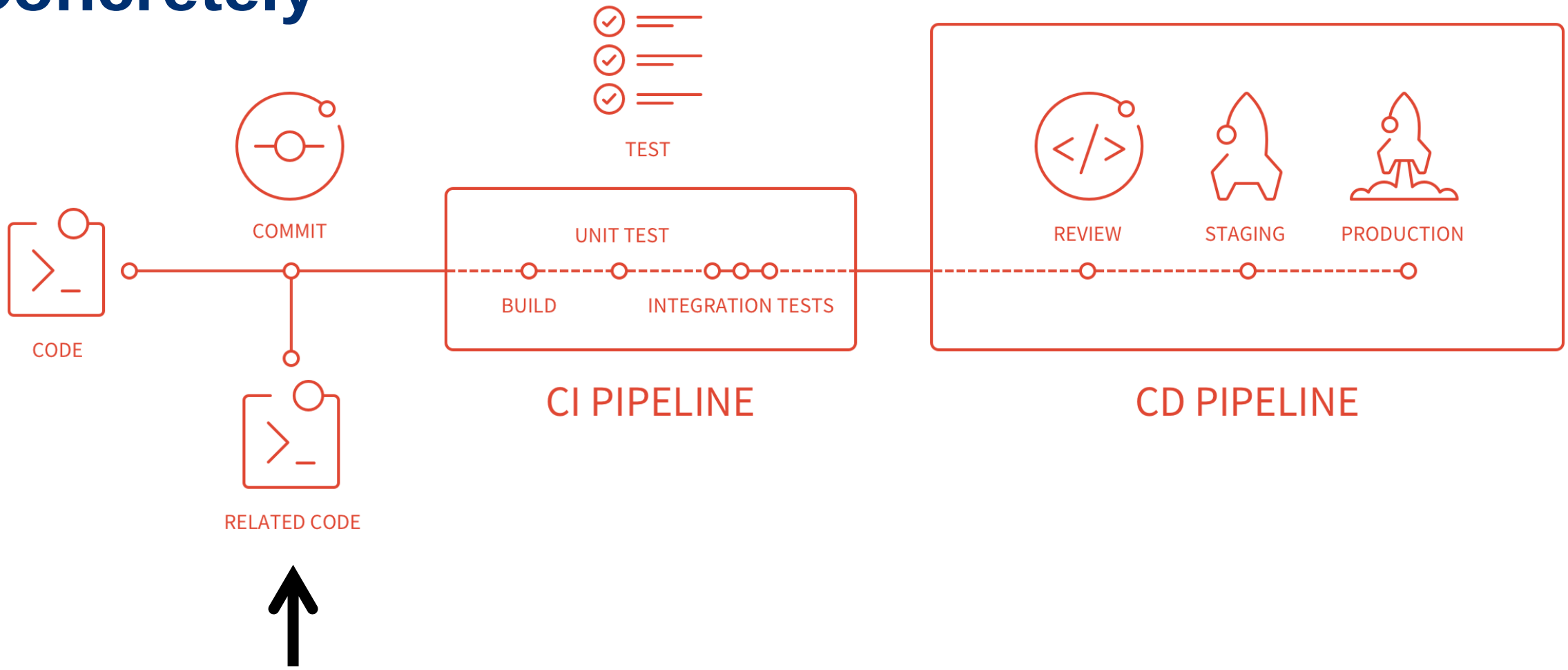
Issue tracking

Code review

CI/CD pipeline

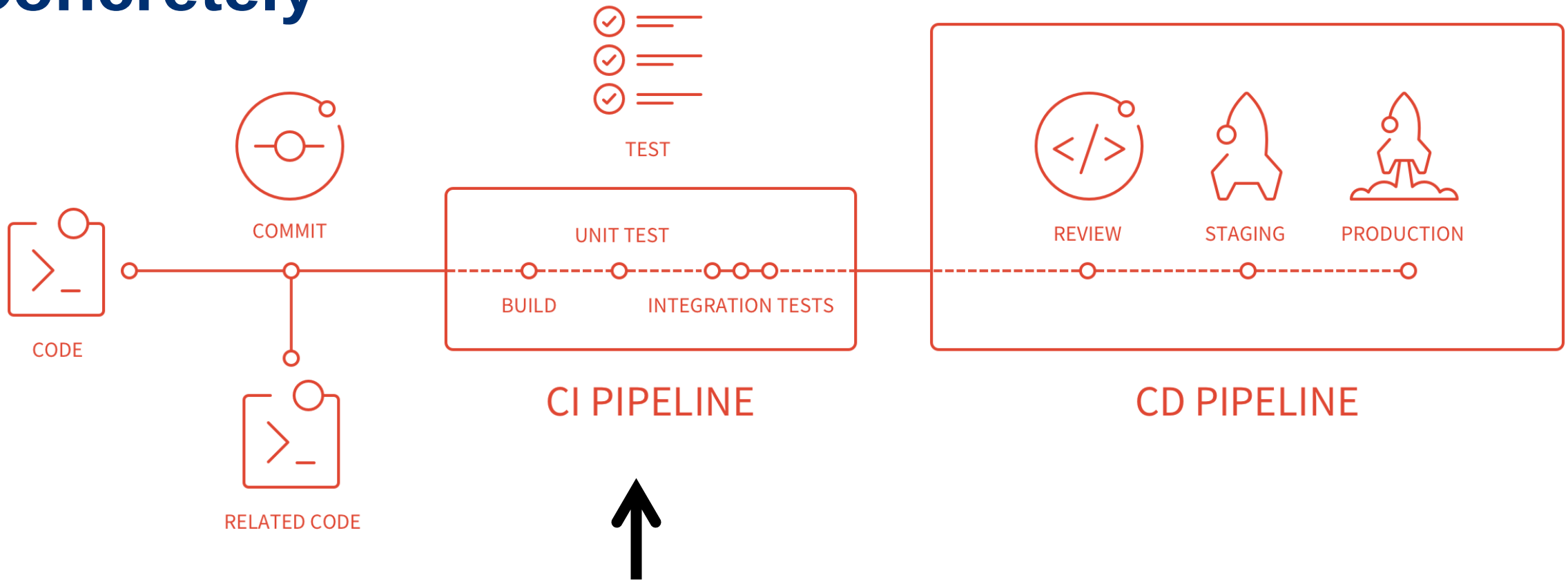


Concretely



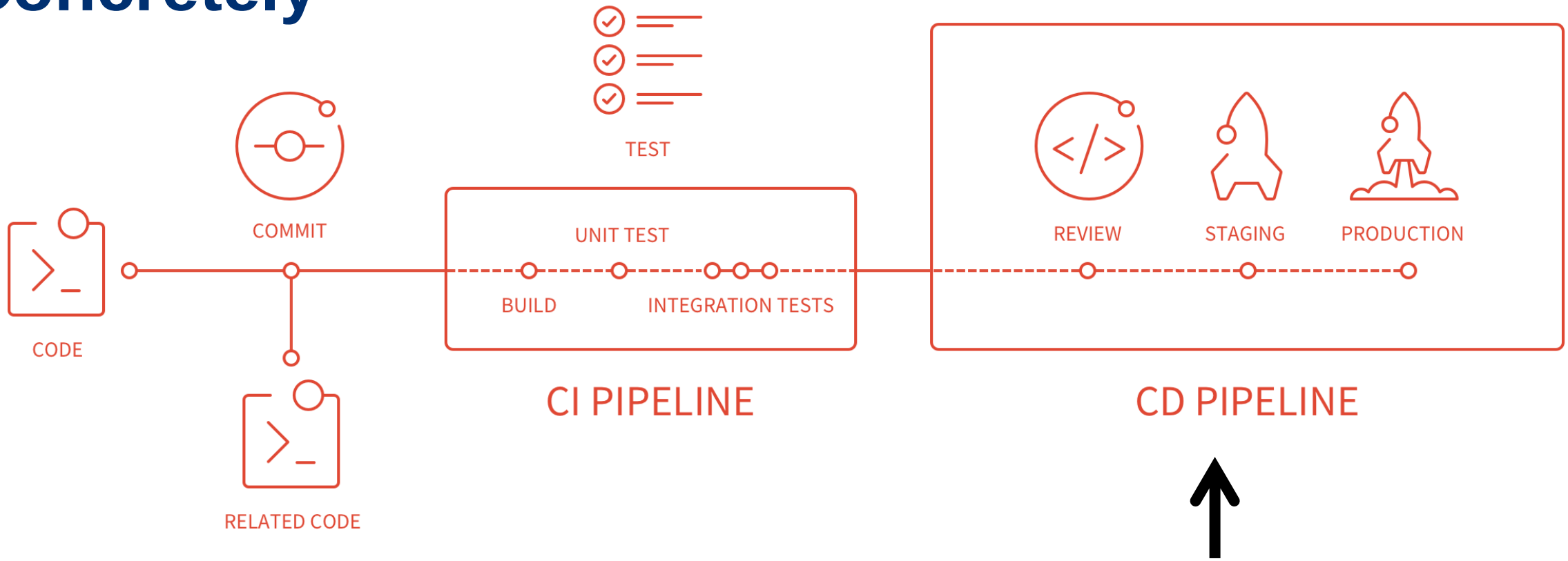
1. Commit a new feature (add a new VLAN for example)

Concretely



2. GitLab CI pipeline can do multiple checks on your code
   - Apply your new playbooks in –check mode to validate that you code can run
   - Check the current state of your infrastructure, ensure that things aren't up and down

Concretely



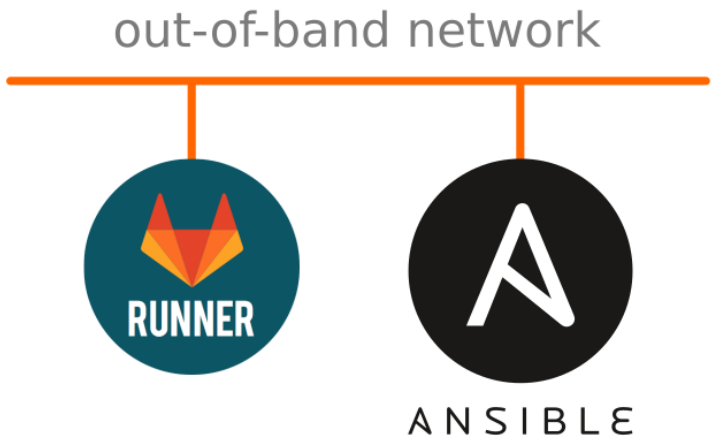
3. Validate your change in a staging environment (can be virtual)
4. Assign to a colleague for review, deploy your change in production

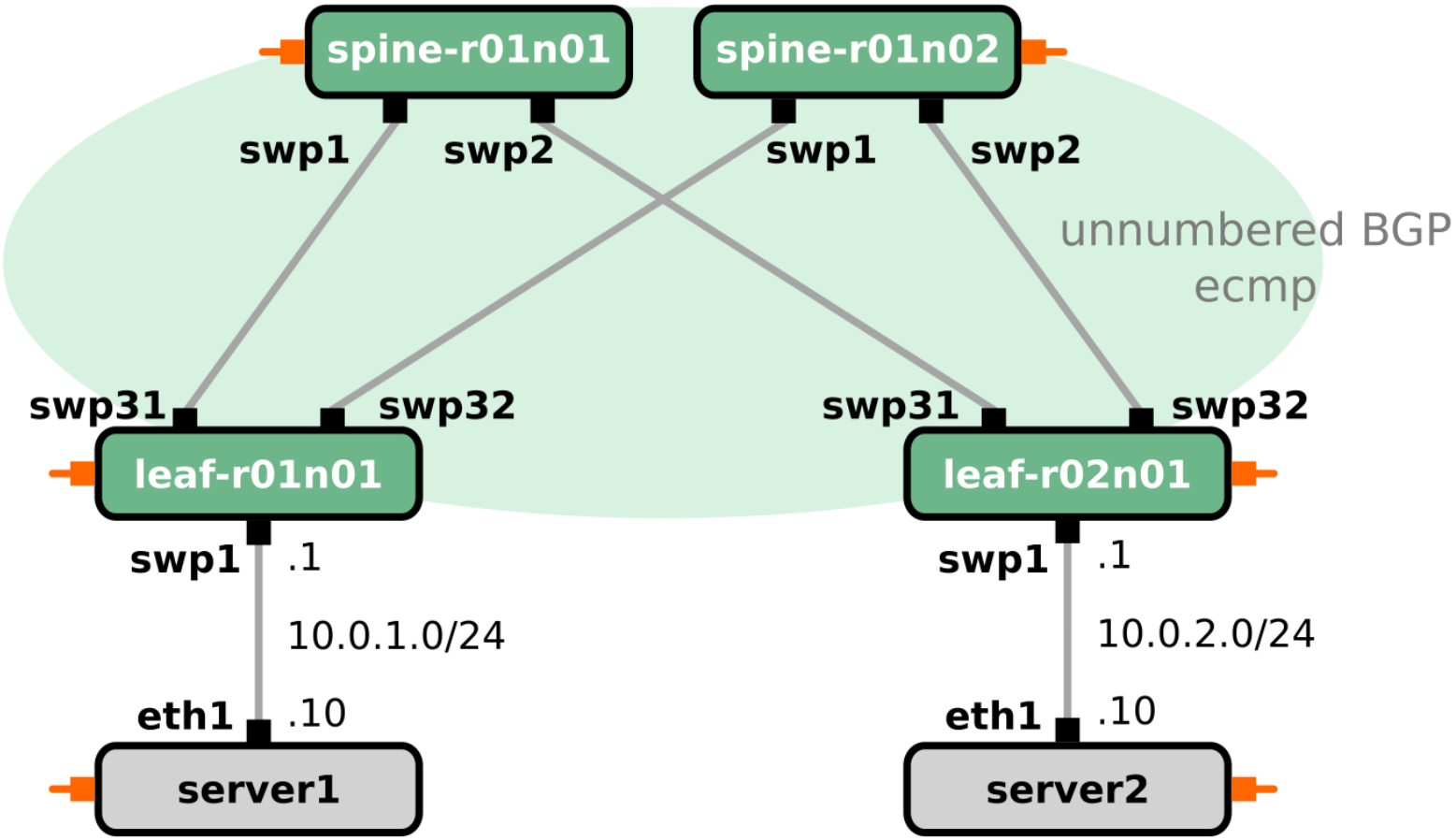
Live-demo time

leaf-spine staging env.

2017.06.19 - Romain Aviolat

github.com/xens





Busting myths

- This model doesn’t fit for small infrastructures
  - That’s wrong, this model is applicable to any infrastructure size
  - The later you dig into it the more it’ll cost (time, money)
  - for very small setup the short-term benefits will be mitigated by the time to put this model in place

- Traditional IT teams won’t be able to operate such model
  - people are eager to learn and immediately catch the benefits of operating IT this way
  - Some guidance and help may be needed to ramp-up people

References

- https://en.wikipedia.org/wiki/Kudelski_Group
- https://gitlab.com
- https://github.com/xens
- https://cumulusnetworks.com