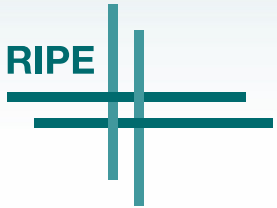
RIPE

Simulating networks at scale with Clabernetes and OVHcloud

Carl Montanari <carl.r.montanari@gmail.com>
Simon Peccaud <simon.peccaud@ovhcloud.com>



Agenda

- Clabernetes
- Clabernetes at OVHcloud
- Q&A

Claber… What?

Start at the… start: Containerlab

- Declarative, Repeatable, Share-able network labs
- Containerized NOS support, but *also* VMs[1]
- FOSS, batteries included, very awesome

CONTAINERlab

Big ol' huge labs? Resource hungry NOS(es)?

- Sure, but… containerlab runs on one host
- Multi-node topologies supported w/ VxLAN interconnects
  - Doable, yes, but... tedious/error prone unless fully automated

What if…

- There was some sort of scale out compute system we could use for network labs….

1: Package up « non » containerized NOS with hellt/vrnetlab

RIPE

Kubernoodles Enters The Chat

Whats a Kubernetes

- Container orchestration platform
- Intelligent workload scheduling, scale, and manage software
- Extensible by writing *controllers* and *custom resources*

Why Kubernetes

- Pervasive, and getting more pervasive
- We already have containers, we just need to spread ‘em around
  - Resource requests help the scheduler decide where to put workloads
- Cool points?
- Deployment freedom: self-hosted, managed, on-prem, in cloud, anywhere!

Hello, Clabernetes

Goals

- Containerlab, just scaled out
- Install in any k8s cluster, no drama
- No need for k8s PhD
- « just works » with existing clab topologies

Design

- Be dumb… smartly?
  - Use containerlab, it already does most of the stuff we need
- Standard k8s « things »
  - Simple helm chart install
  - Few, simple CRDs
  - No CNI/cluster requirements

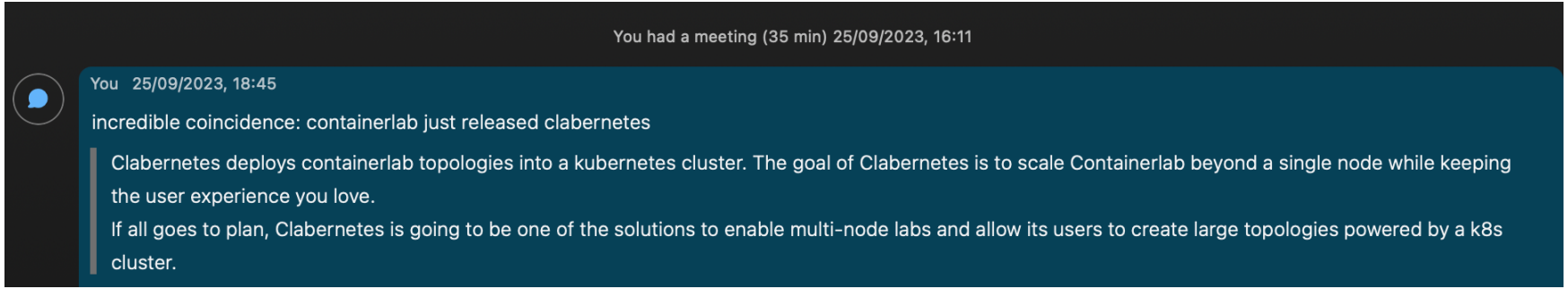
How it Works

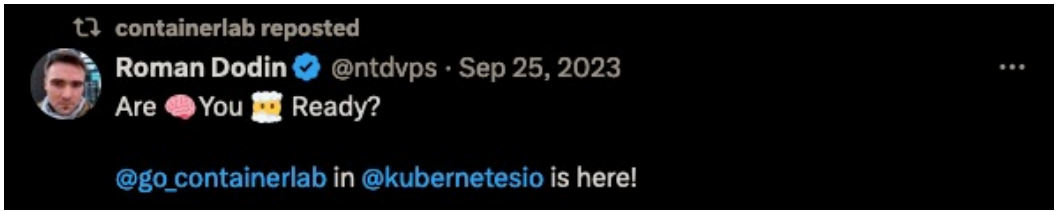
TL:DR

- Toplogy CRD holds « normal » containerlab topology
  - And some other goodies
  - Can get to this by using the « clabverter » tool
- K8s pod per containerlab node
- Plumbed together with VxLAN tunnels
- Nodes exposed via LoadBalancer service
- Topology updates cause reconcile event
  - Only restart nodes that have had things change
- ???
- Profit! J/K its OSS

```yaml
---
apiVersion: clabernetes.containerlab.dev/v1alpha1
kind: Topology
metadata:
  name: hi-ripe-people
  namespace: clabernetes
spec:
  definition:
    containerlab: |-
      name: does-it-work
      topology:
        nodes:
          srl1:
            kind: srl
            image: ghcr.io/nokia/srlinux
          srl2:
            kind: srl
            image: ghcr.io/nokia/srlinux
        links:
          - endpoints: ["srl1:e1-1", "srl2:e1-1"]
          - endpoints: ["srl1:e1-2", "srl2:e1-2"]
```

Clabernetes @OVH: the backstory





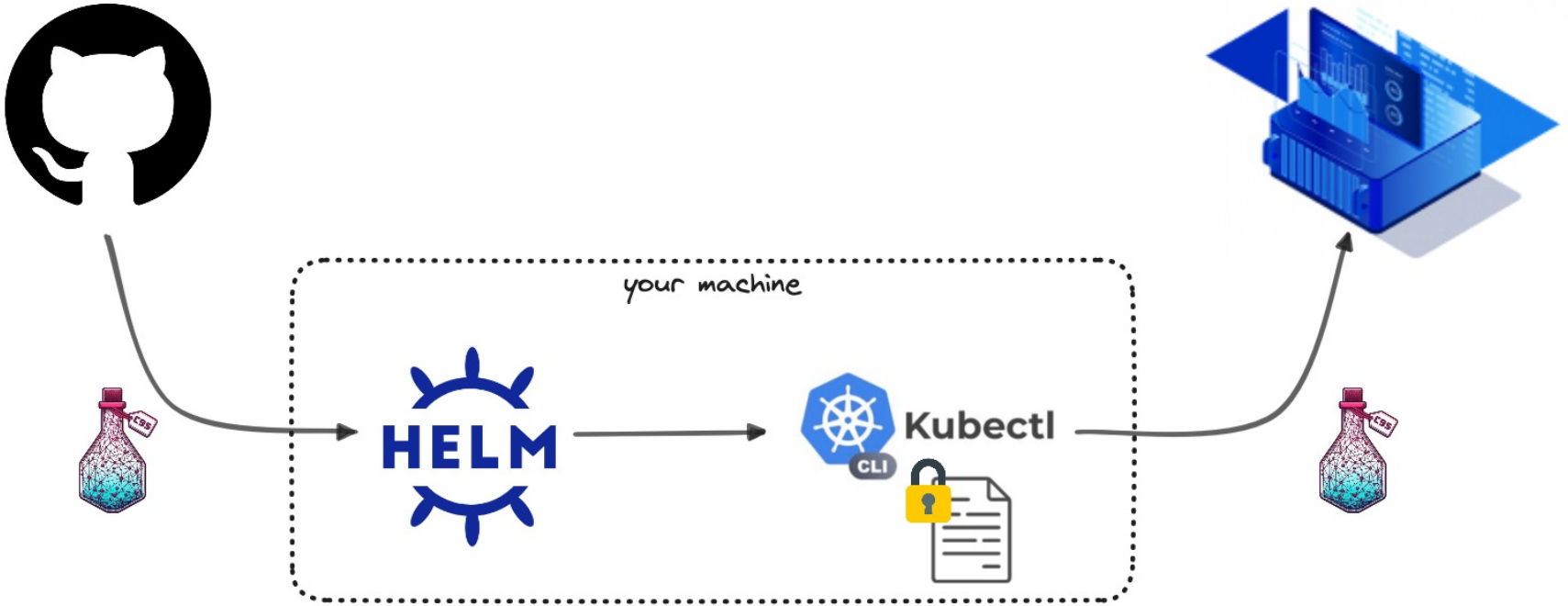
Managed Kubernetes Service

+ Create a Kubernetes cluster

1. Select a location
2. Select the minor version of Kubernetes you want
3. Choose a private network for this cluster
4. Configure your node type
5. Configure node pool size
6. Billing and anti-affinity
7. Name your cluster

Name	Flavor	Billing type	Status	
c9s-nodepool-node-2eaf7f	C2-30 (8 CPU 30 GB RAM 200 GB disk)	Hourly	OK	•••
c9s-nodepool-node-48b95c	C2-30 (8 CPU 30 GB RAM 200 GB disk)	Hourly	OK	•••
c9s-nodepool-node-af1da0	C2-30 (8 CPU 30 GB RAM 200 GB disk)	Hourly	OK	•••
c9s-nodepool-node-a11135	C2-30 (8 CPU 30 GB RAM 200 GB disk)	Hourly	OK	•••
c9s-nodepool-node-7e7e7c	C2-30 (8 CPU 30 GB RAM 200 GB disk)	Hourly	OK	•••
c9s-nodepool-node-eee2c5	C2-30 (8 CPU 30 GB RAM 200 GB disk)	Hourly	OK	•••

Install clabernetes

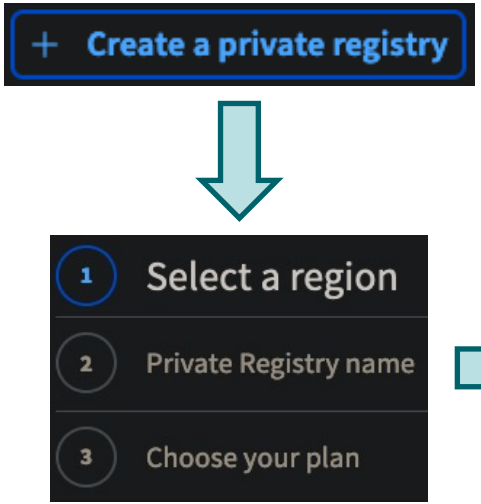


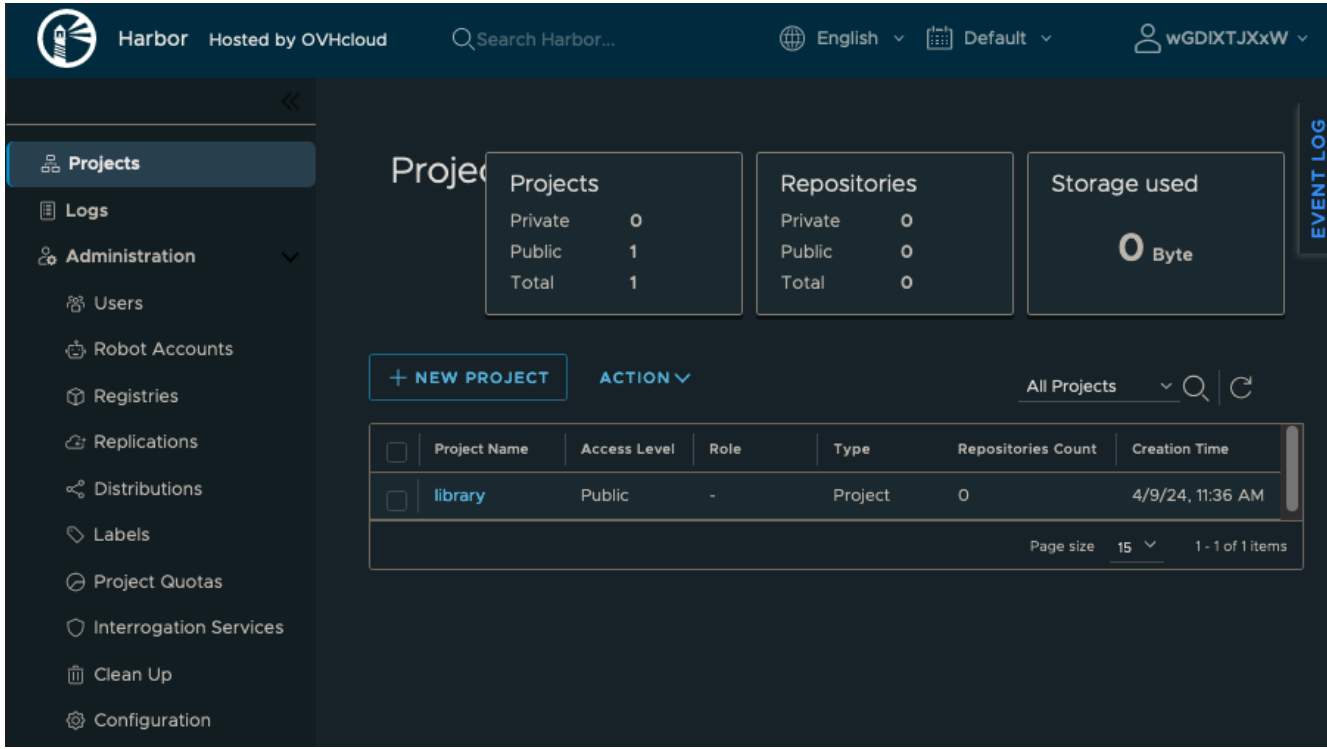
Install clabernetes - command

install clabernetes

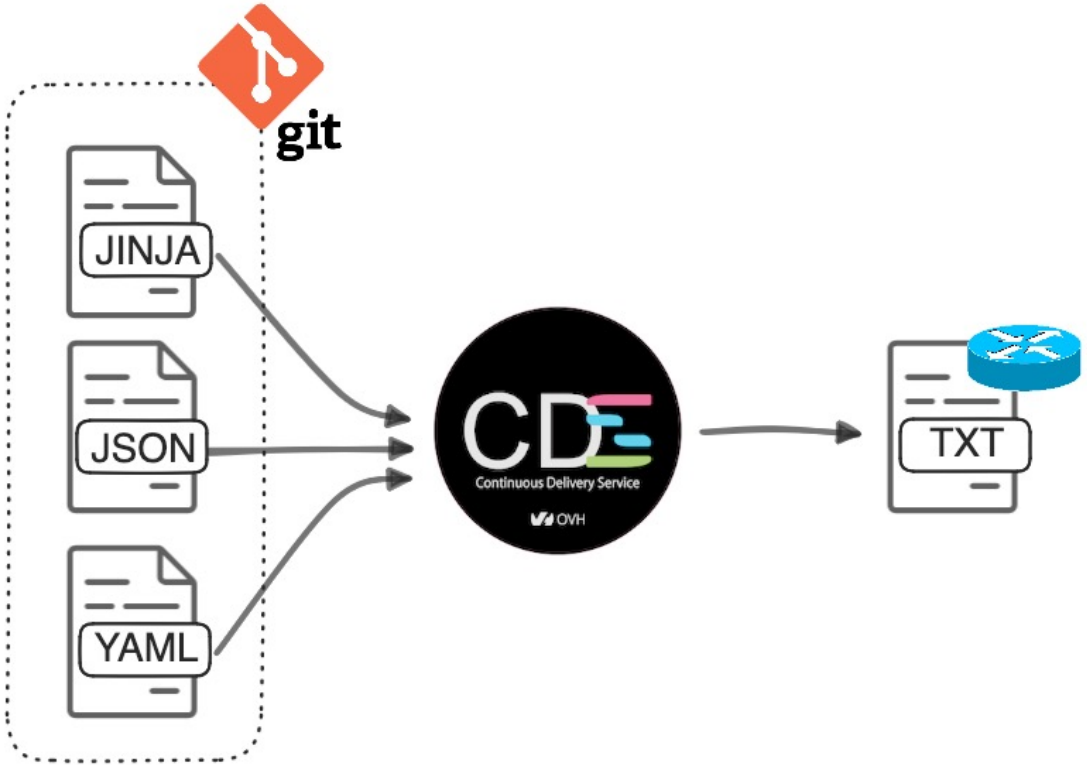
```
~/dev/clabernetes-ovh master*
❯ helm upgrade --install --namespace clabernetes -f clabernetes-values.yml clabernetes oci://ghcr.io/srl-labs/clabernetes/clabernetes
Release "clabernetes" does not exist. Installing it now.
Pulled: ghcr.io/srl-labs/clabernetes/clabernetes:0.1.0
...
STATUS: deployed
REVISION: 1
TEST SUITE: None

~/dev/clabernetes-ovh master*
❯ kubectl get pods --namespace clabernetes
NAME                                  READY   STATUS      RESTARTS   AGE
clabernetes-clicker-28466384-tss8q    0/1     Completed   0          15s
clabernetes-manager-6f7b9f4cb8-7q4l6  1/1     Running     0          24s
clabernetes-manager-6f7b9f4cb8-g68wz  1/1     Running     0          24s
clabernetes-manager-6f7b9f4cb8-nfqlp  1/1     Running     0          24s
```

Managed Private Registry





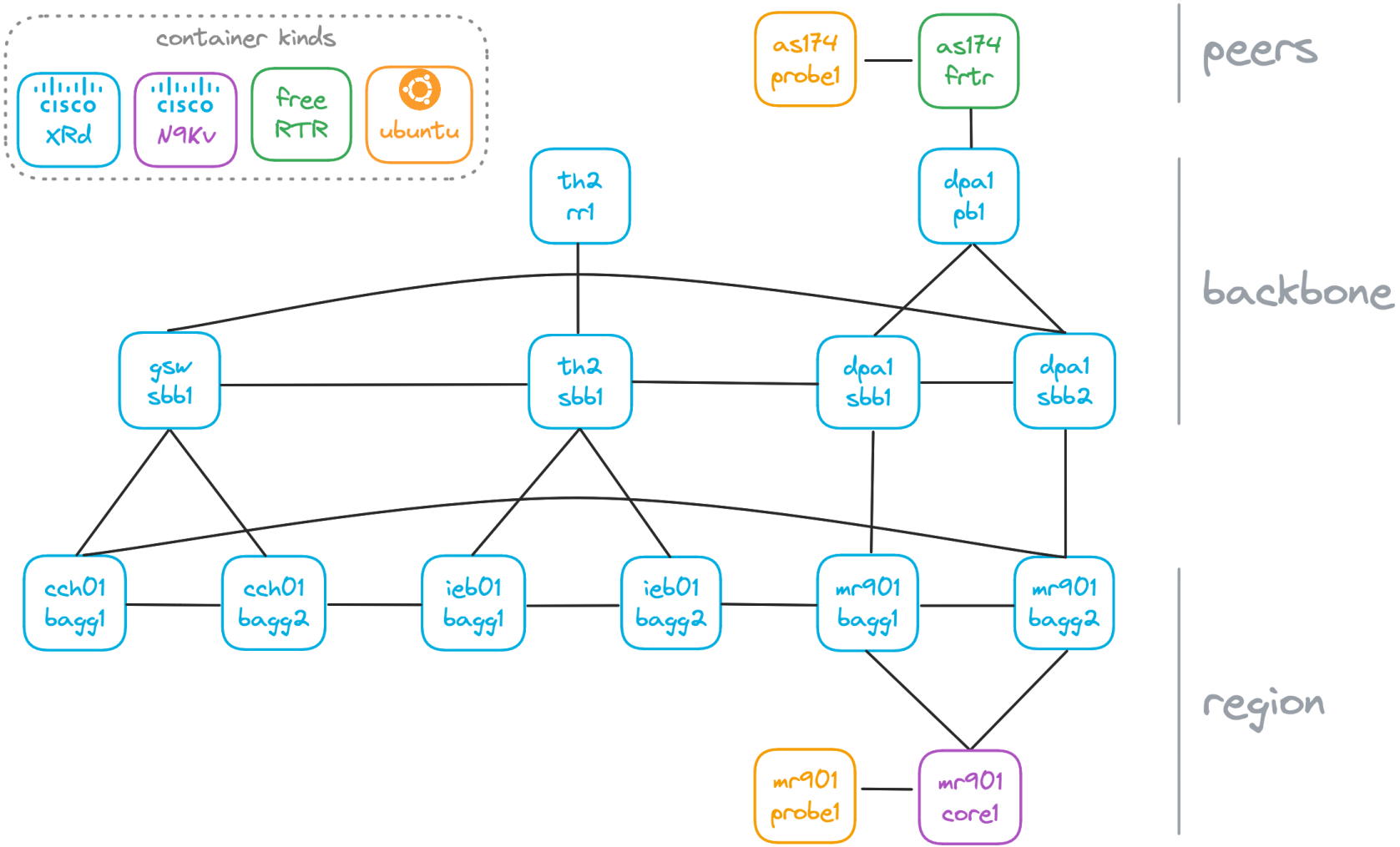
Network configuration



Yes but

… is the configuration syntax valid ?

… am I breaking anything ?

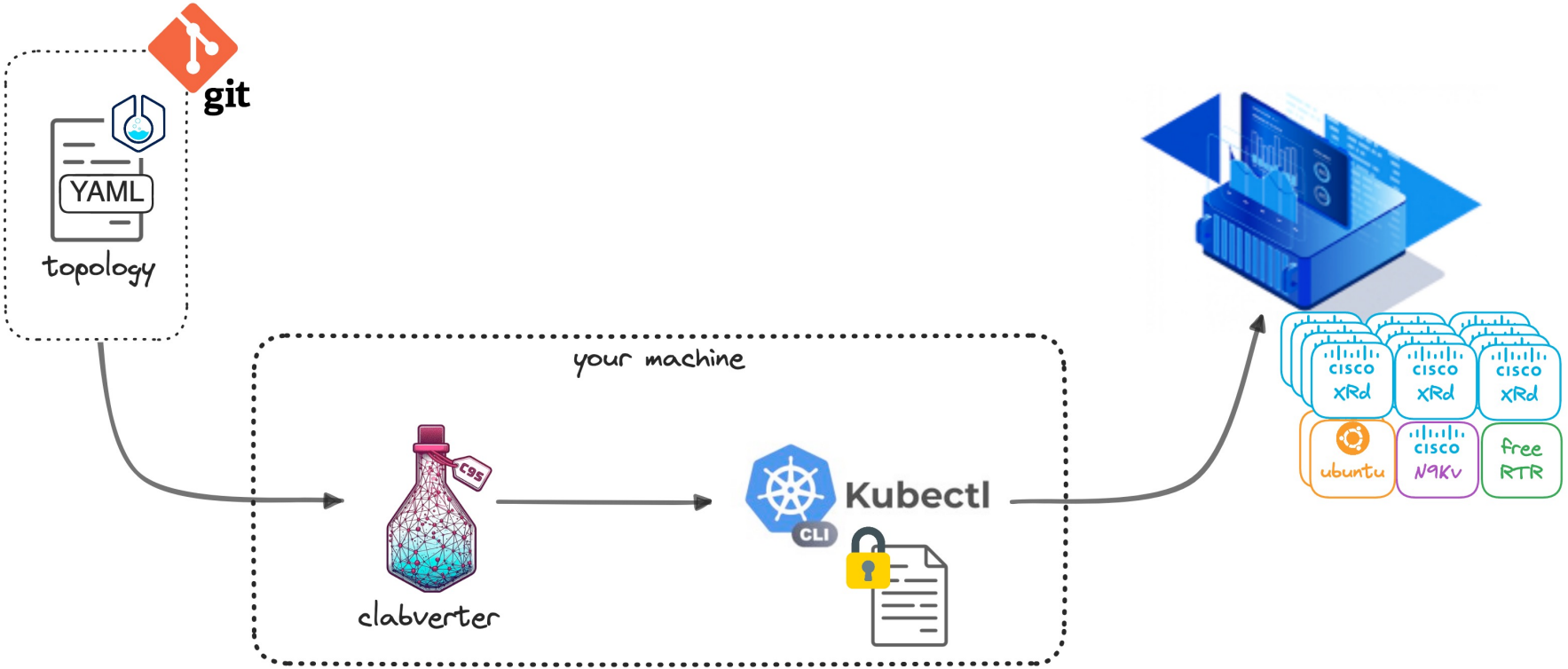

container kinds
cisco XRd
cisco N9Kv
free RTR
ubuntu
as174 probe1
as174 frtr
th2 rr1
dpa1 pb1
gsw sbb1
th2 sbb1
dpa1 sbb1
dpa1 sbb2
cch01 bagg1
cch01 bagg2
ieb01 bagg1
ieb01 bagg2
mr901 bagg1
mr901 bagg2
mr901 probe1
mr901 core1
peers
backbone
region

Paris area – topology file

topology.clab.yml - shortened for legibility

```yaml
nodes:
  par3-mr901-bagg2-8k:
    kind: xrd
    image: my.container-registry.ovh.net/cisco/xrd-control-plane:7.7.1
    startup-config: configurations/par3-mr901-bagg2-8k/output.txt
  par3-mr901-core1-n94:
    kind: cisco_n9kv
    image: my.container-registry.ovh.net/cisco/vr-n9kv-9300:10.3.4a
    startup-config: configurations/par3-mr901-core1-n94/output.txt
  par-gsw-tier1-ubuntu:
    kind: linux
    image: my.container-registry.ovh.net/ovh/netprobe:0.1.11
    binds:
      - configurations/par-gsw-tier1-ubuntu/output.txt:/root/entrypoint.sh

links:
  - endpoints: ["par3-mr901-bagg2-8k:Gi0-0-0-1", "par3-mr901-core1-n94:eth16"]
  - endpoints: ["par3-mr901-srv1-ubuntu:eth1", "par3-mr901-core1-n94:eth1"]
```

Deploy time


git
YAML
topology
your machine
clabverter
Kubectl
CLI
cisco XRd
cisco XRd
cisco XRd
ubuntu
cisco N9Kv
free RTR

Laying back a bit

deploy topology - output shortened for legibility

```
~/dev/containerlab-topologies/topology master*
❯ clabverter --stdout | kubectl apply -f -
    INFO |               clabverter | starting clabversion!
    INFO |               clabverter | attempting to find topology file in the working directory...
    INFO |               clabverter | found topology file "topology.clab.yml"
 ...
    INFO |               clabverter | clabversion complete!
namespace/c9s-topology created
...
topology.clabernetes.containerlab.dev/topology created

~/dev/containerlab-topologies/topology master*
❯ kubectl get pods --namespace c9s-topology
NAME                                                READY   STATUS              RESTARTS   AGE
topology-par3-mr901-bagg2-8k-7c6c75f56c-s77rc       1/1     Running             0          12s
topology-par3-mr901-core1-n94-575f4c4fcd-cr97v      0/1     ContainerCreating   0          12s
...
topology-par-gsw-tier1-ubuntu-758b7556d5-tvbt7      1/1     Running             0          13s
```

Sure but is my design working ?

- containerlab automatically provision management interfaces
- clabernetes embeds sshin binary to swiftly ssh into the pod container

```
connecting to routers

❯ kubectl exec -ti speccaud-1b-par3-par-gsw-sbb1-nc5-7c54946858-nd85q sshin
(clab@par-gsw-sbb1-nc5) Password:
RP/0/RP0/CPU0:par-gsw-sbb1-nc5# sh isis neighbors
IS-IS BB neighbors:
System Id      Interface        SNPA           State Holdtime Type IETF-NSF
par-dpa1-sbb2-8k BE112            *PtoP*         Up    12       L2   Capable
par-th2-sbb1-nc5 BE10             *PtoP*         Up    13       L2   Capable
par-gsw-pb1-nc5 BE300             *PtoP*         Up    10       L2   Capable
```

Nicer way

netprobe fping loop

```
fping --addr --name --period=5000 --squiet=5 --loop {{ destinations }} | sed {{ json formatting }}
```

netprobe json output

```
{
  "netprobe_dst_name": "par3-cch01-bagg1-8k_loopback0_v6",
  "netprobe_ping_rcv": 1,
  "netprobe_success": true
}
```

fluentbit

OVHcloud

Logs Data Platform

OpenSearch

par3-mr901-srv1-ubuntu: probing results

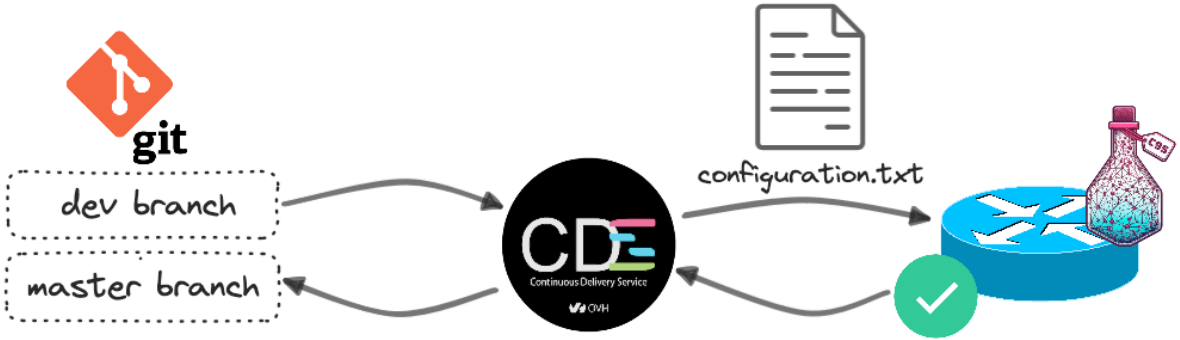
destination	reachable
par3-mr901-bagg2-8k_loopback0_v6	true
par3-mr901-bagg2-8k_loopback0_v4	true
par3-mr901-bagg1-8k_loopback0_v6	true
par3-mr901-bagg1-8k_loopback0_v4	true

Feedback

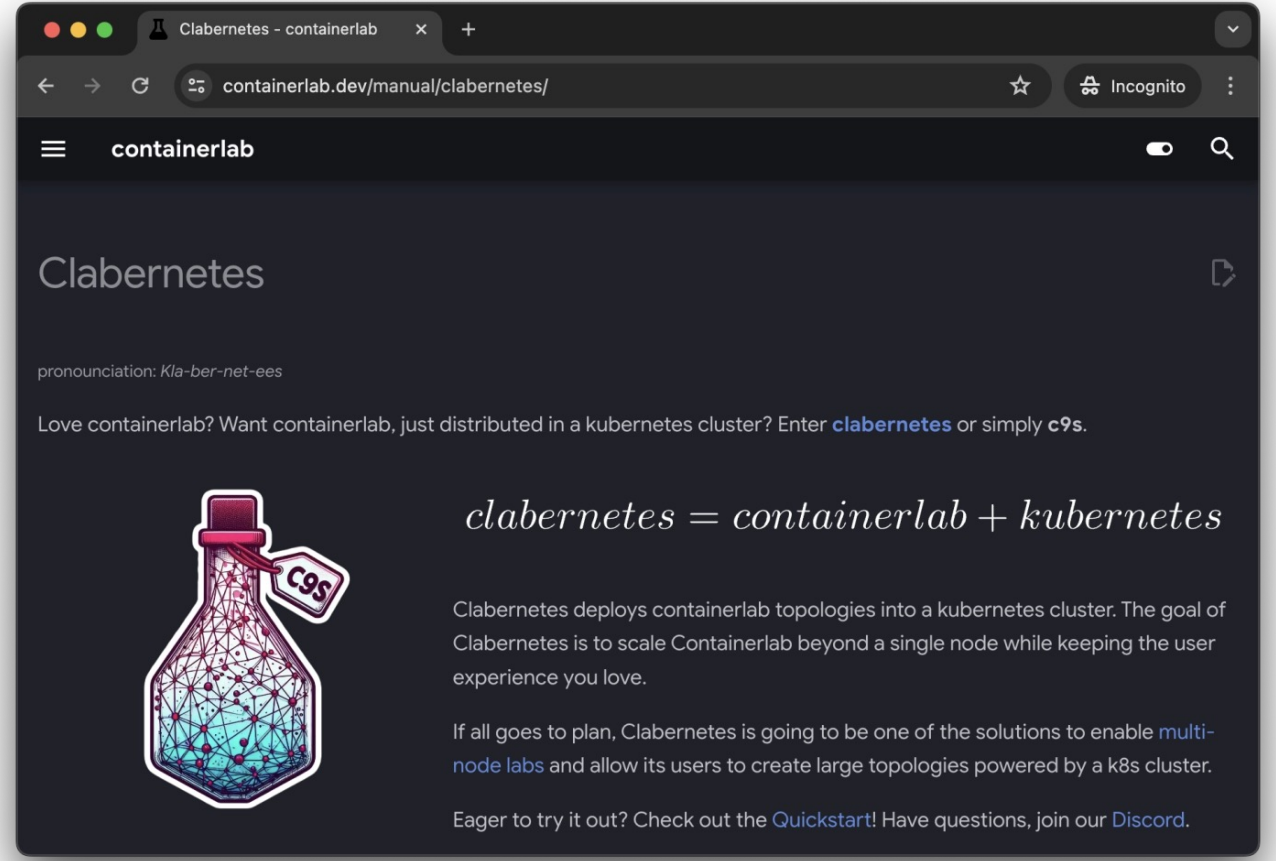
- We can now run as much production replicates as we like
- Clabernetes + cloud-managed kubernetes cluster = virtually no pain
- More container-native network OSes would be nice (NX-OS)

Where we're headed at

- Simulate the whole backbone/regions !
- CI/CD one-shot labs validation



Get started


Clabernetes - containerlab
containerlab.dev/manual/clabernetes/
Incognito
containerlab
Clabernetes
pronounciation: Kla-ber-net-ees
Love containerlab? Want containerlab, just distributed in a kubernetes cluster? Enter clabernetes or simply c9s.
c9s
clabernetes = containerlab + kubernetes
Clabernetes deploys containerlab topologies into a kubernetes cluster. The goal of Clabernetes is to scale Containerlab beyond a single node while keeping the user experience you love.
If all goes to plan, Clabernetes is going to be one of the solutions to enable multi-node labs and allow its users to create large topologies powered by a k8s cluster.
Eager to try it out? Check out the Quickstart! Have questions, join our Discord.

Thanks for your attention.

Questions ?

RIPE