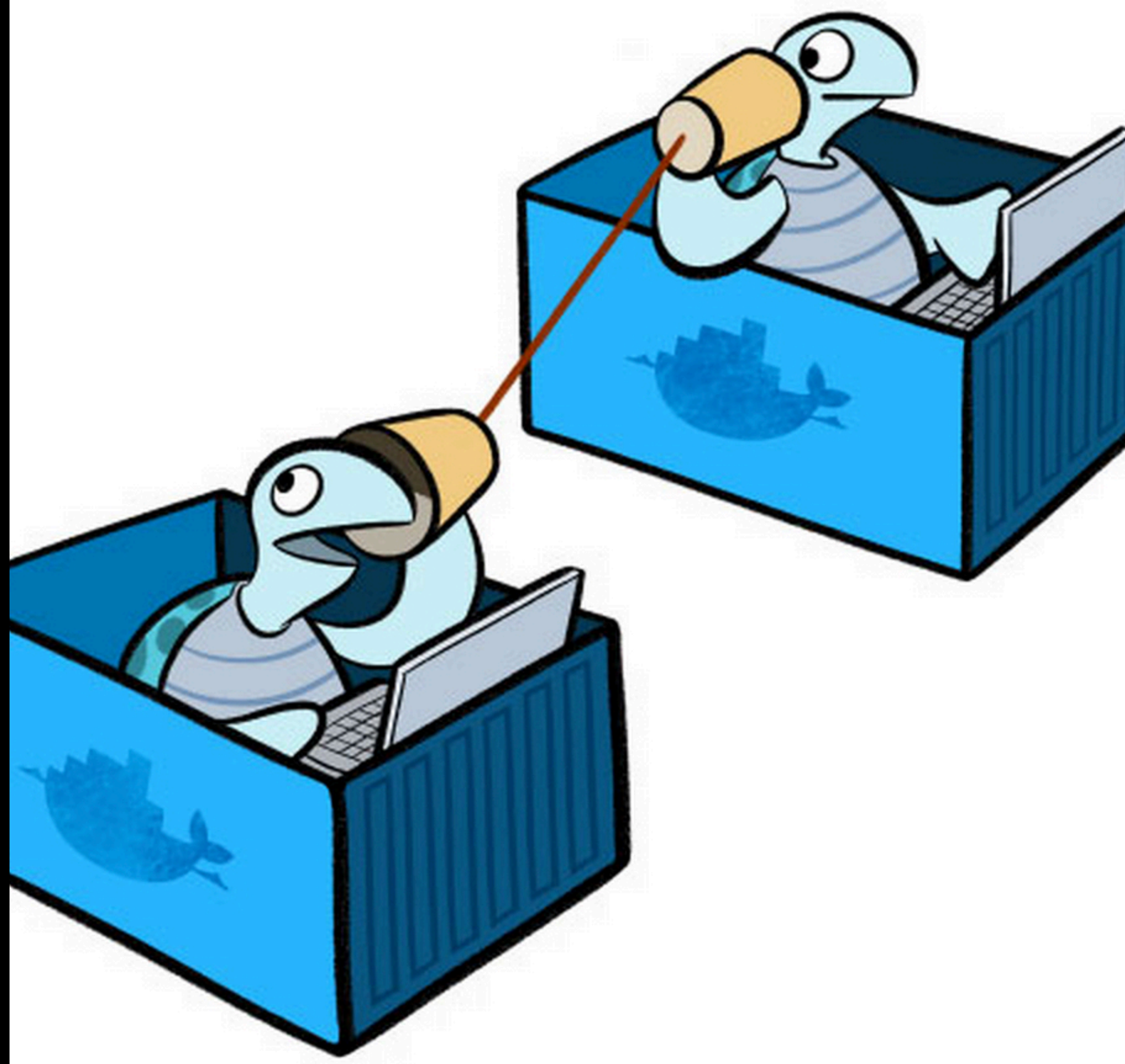
Docker Networking

From NetOps dude view

© Vicente De Luca, 2015 - http://www.nethero.org



Vicente De Luca

About me

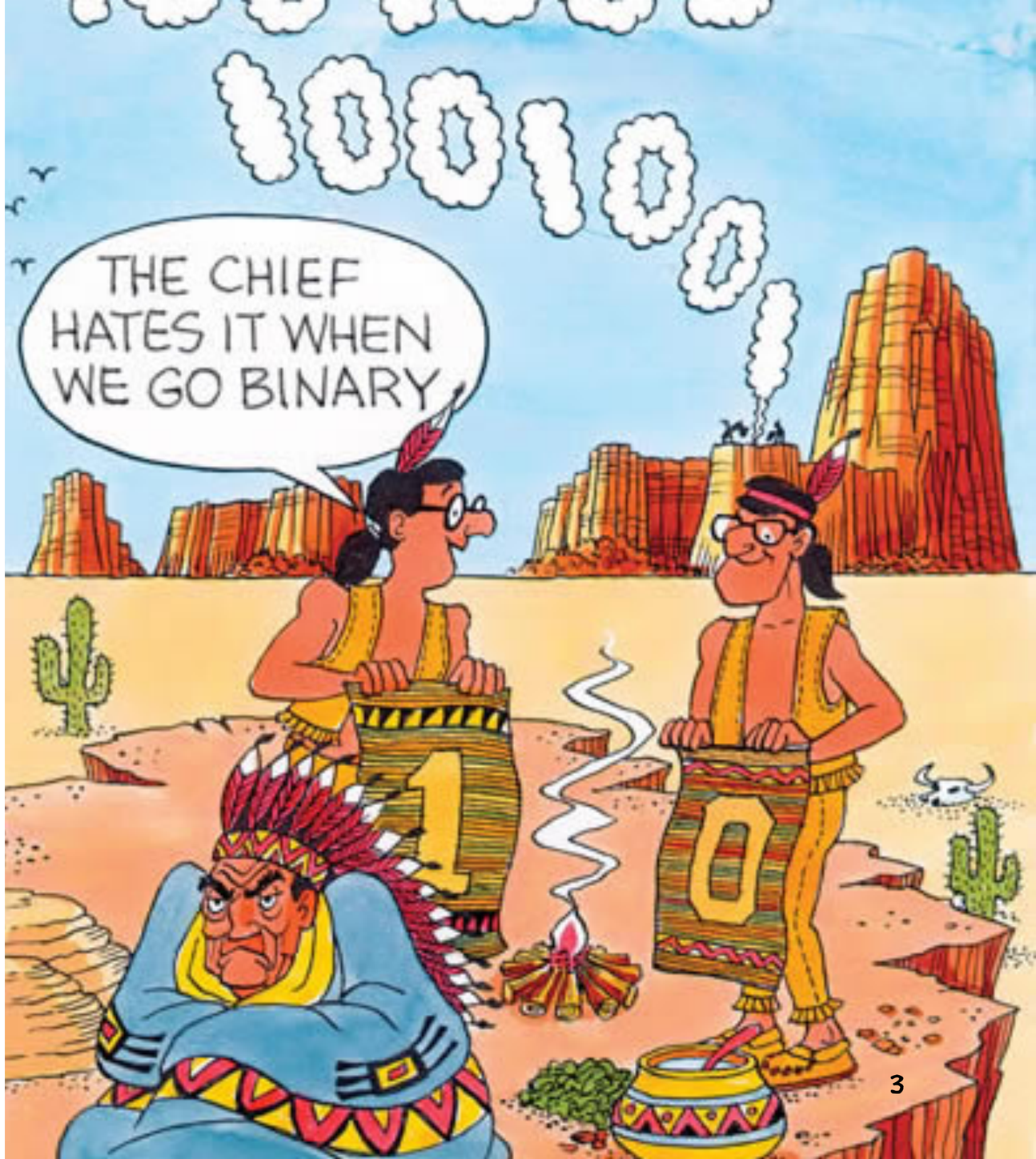
- Formal Network Engineer
- Jack of all trades
- Works for Zendesk
- vdeluca@zendesk.com
- http://nethero.org
- http://github.com/zenvdeluca
- @zenvdeluca
- Hacking since the 90's
- IPv6 Evangelist

© Vicente De Luca, 2015 - http://www.nethero.org

Docker engine

Network support

- Bridge (default)
- Port mapping (NAT)
- Host mapping (--net host)
- Container (--net container)
- No network (--net none)



© Vicente De Luca, 2015 - http://www.nethero.org

Bridge mode

Characteristics:

- Default mode
- Sequential IP assignment
- Linux Worker = gateway

© Vicente De Luca, 2015 - http://www.nethero.org

Port mapping (NAT)

Characteristics:

- Add more Ops overhead
- Linux Worker = gateway
- Add difficulty for logging
- Resource exhaustion risks

Don't scale!

- at 15 containers =~ 2000 ports
- Avoid NAT444
- Docker NAT != CGN

© Vicente De Luca, 2015 - http://www.nethero.org

Container namespace

Characteristics:

- share container net namespace

© Vicente De Luca, 2015 - http://www.nethero.org

Host mapping

Characteristics:

- Bind at worker net namespace :|
- Usually One IP for all containers
- No network ns isolation
- Concurrent apps: some risks

Where it usually starts

But as you grow...

Bad things can happen..

© Vicente De Luca, 2015 - http://www.nethero.org

Limits and Risks

- LISTEN conflicts if same port
  - Non stop code deploy?
- TCP connections to same dst_ip:port
  - Port exhaustion risk, if:
  - ~500 new req/s
  - default time_wait = 60s

© Vicente De Luca, 2015 - http://www.nethero.org

Complicate Troubleshooting

- netflow
  - all apps run at worker IP
- i.e: Which app is starving?
  - easy for inbound
  - hard for outbound

© Vicente De Luca, 2015 - http://www.nethero.org

DOOOEEET RIGHT

FROM THE BEGINNING!

© Vicente De Luca, 2015 - http://www.nethero.org

Proposed approach

One IP per container:

- containers <=> containers, no NAT
- nodes <=> containers, no NAT
- IP that sees itself as, is the same others see it
- != containers *CAN* expose same service port
- Cluster managers requirement, like Kubernetes
- Supports IPv4 (DHCP) and IPv6 (SLAAC)

One IP per container

Different tracks

Dynamic Routing

Building complicated solutions to simple problems

- Advertise running container IPs to network routers
- Uses BGP protocol
  - NetOps OK, DevOps not so

www.projectcalico.com

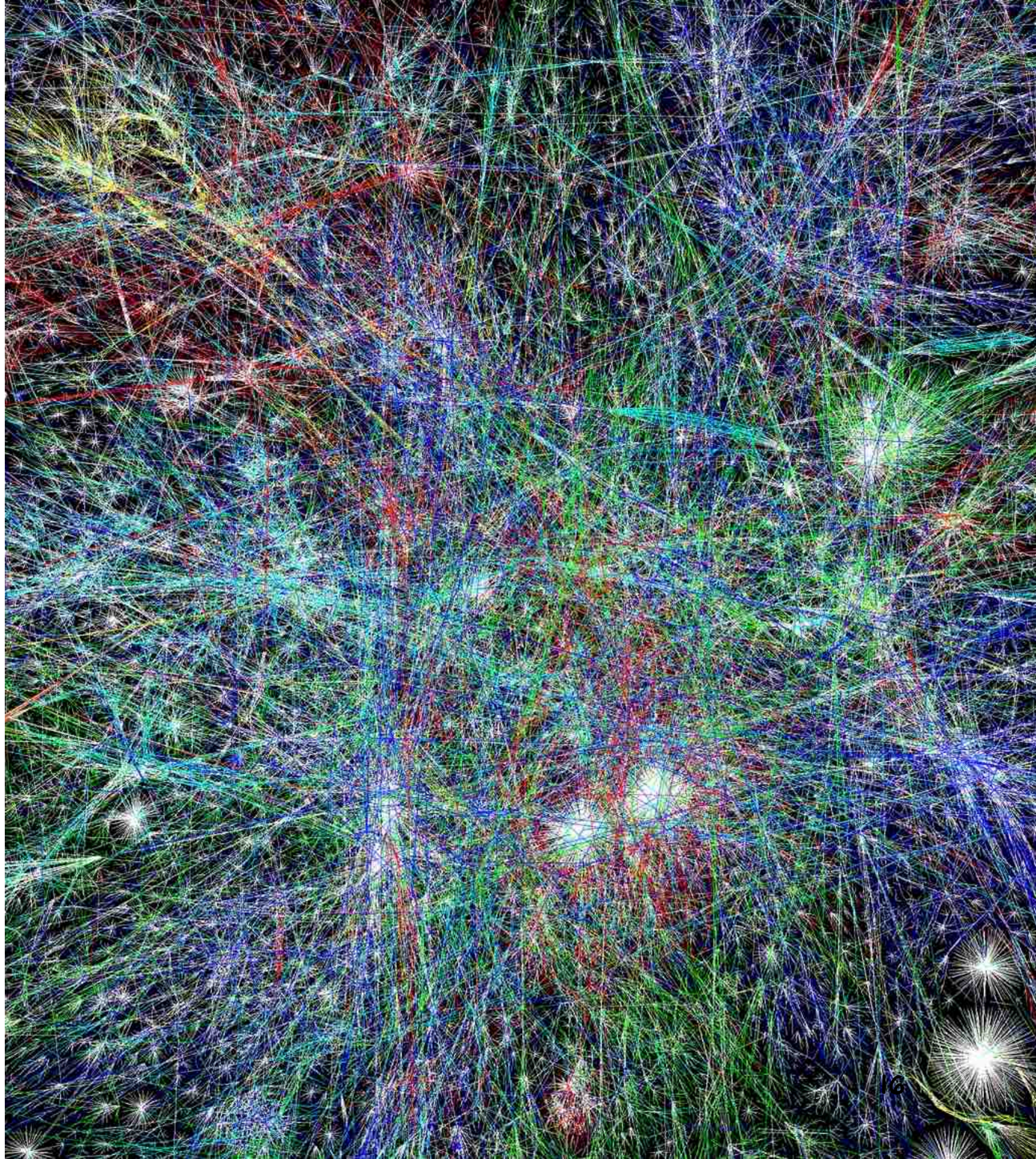
© Vicente De Luca, 2015 - http://www.nethero.org

Overlay Network

For Cloud Providers or Staging:

Flannel / Weaver

- overhead increase
- reduces the effective MTU size
- may impact TTL
- tunnel = lower performance
- reduces network visibility



© Vicente De Luca, 2015 - http://www.nethero.org

Stop chit chat, please?

Simple is Beautiful

DHCP:

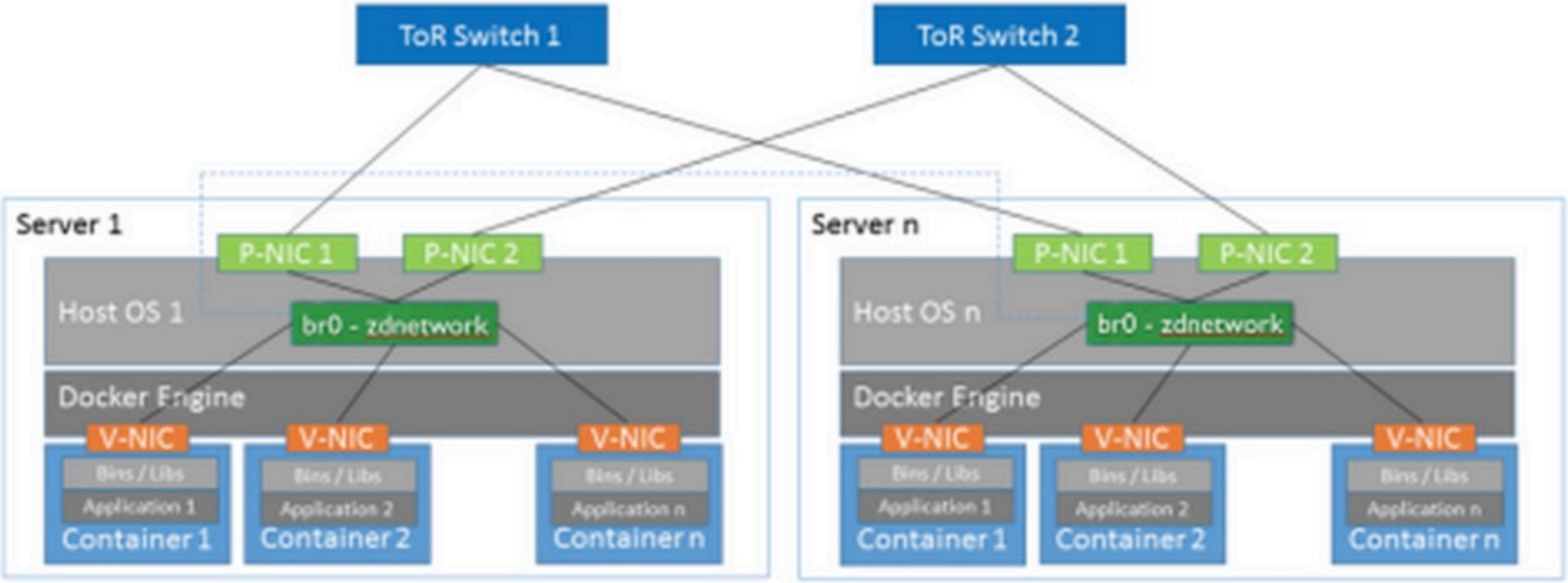
- reliable (RFC2131)
- everybody knows how to tshoot

Bridge:

- Part of kernel since middle ages
- Easily replaceable by vSwitches
- Network gears = L3 router

© Vicente De Luca, 2015 - http://www.nethero.org

Architecture Overview


ToR Switch 1
ToR Switch 2
Server 1
P-NIC 1
P-NIC 2
Host OS 1
br0 - zdnetwork
Docker Engine
V-NIC
V-NIC
V-NIC
Bins / Libs
Application 1
Container 1
Bins / Libs
Application 2
Container 2
Bins / Libs
Application n
Container n
Server n
P-NIC 1
P-NIC 2
Host OS n
br0 - zdnetwork
Docker Engine
V-NIC
V-NIC
V-NIC
Bins / Libs
Application 1
Container 1
Bins / Libs
Application 2
Container 2
Bins / Libs
Application n
Container n

One IP per container

How to ?

- One DHCP client per container
- Facing the container as a RPM on steroids
- DHCP Client runs at worker userspace
- Need a network wrapper for handling the config

MAC Address Generation Algorithm

Generate a locally unique MAC:
XX:IP:IP:IP:YY:ZZ

- XX = restricted random least significant bits = 10
- IP:IP:IP = worker IPv4 3 last octets in HEX
- YY:ZZ = random portion

Network wrapper - bash script

-to be executed after spin up

-Creates a new net namespace

-Associate veth with br0 interface

-Run and maintain the DHCP client

-Garbage collect orphan DHCP

My Simplified fork from github.com/jpetazzo/pipework,
with some hacks to DHCP be RFC compliance.

```
root@docker1 vdeluca]# network-wrapper
Syntax:
network-wrapper inspect[-all] <guestID>
network-wrapper <HostOS-interface> [-i Container-interface] <guestID> dhcp [macaddr]

[root@docker1 vdeluca]# network-wrapper br0 $(docker run --net none -d nginx) dhcp
{
    "Container": "4b94d0da2047",
        "NetworkSettings": {
          "HWAddr": "c6:64:50:63:3b:86",
          "Bridge": "br0",
          "IPAddress": "10.100.80.190",
          "IPPrefixLen": "24",
          "Gateway": "10.100.80.1",
          "DNS": "10.100.64.9, 10.100.66.9",
          "DHCP_PID": "23384"
} }
[root@docker1 vdeluca]# curl 4b94d0da2047.stage.lab.zdsys.com
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
```

QUICK LIVE DEMO

OPEN CHAT QUESTIONS?

THANK YOU!

We are starting the 2nd round at the Barge Pub, just across the street!

Sorry for that streaming fellows :(