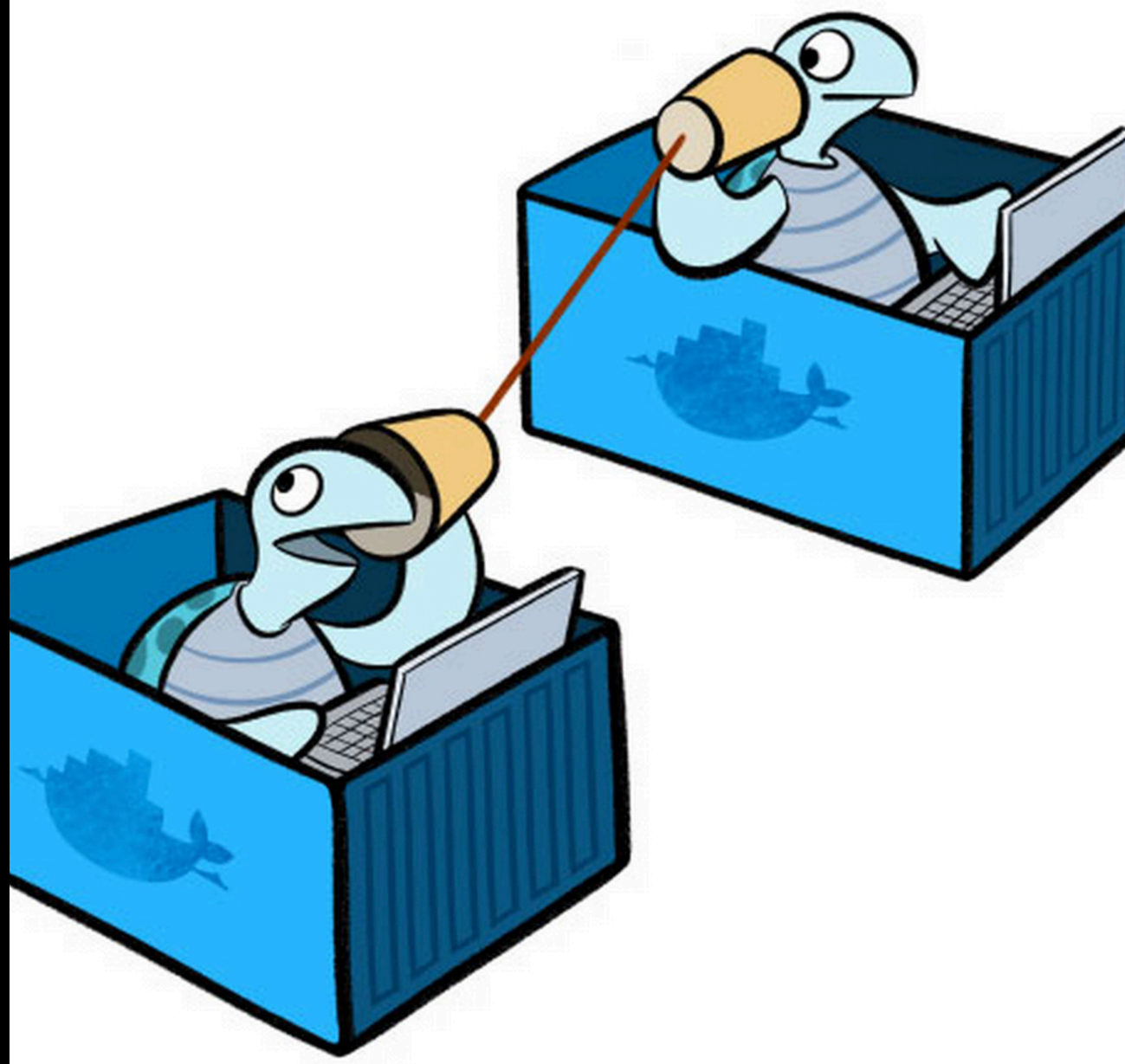
Docker

Container Linking

for production

From a NetOps dude view

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



Vicente De Luca

About me

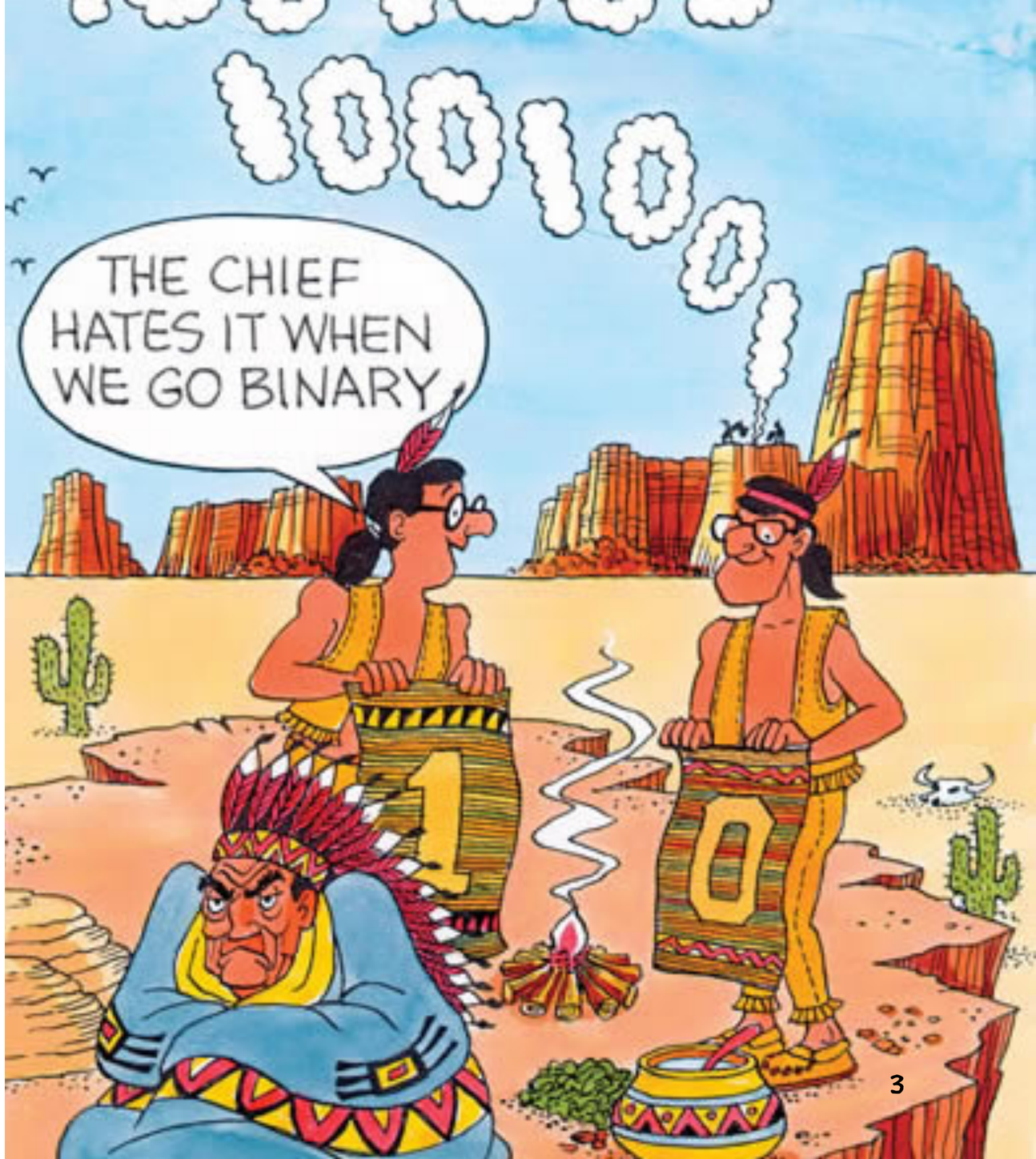
- Network Engineer @ Zendesk
- jack of all trades
- hacking since 90's
- IPv6 Addicted and Evangelist

http://nethero.org
http://github.com/zenvdeluca
vdeluca@zendesk.com
@zenvdeluca

Docker engine

Network support

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



Bridge mode

Characteristics:

- Docker default
- Sequential IP assignment
- Linux Worker = gateway 😟

Port mapping (NAT)

Characteristics:

- Add operations overhead 😟
- Linux Worker = gateway 😟
- Add difficulty for logging 😟
- Resource exhaustion risks 😟

Don't scale!

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

Container namespace

Characteristics:

- share other container net namespace
- recent feature

Host mapping

Characteristics:

- Bind at worker net namespace 😐
- Mostly one IP, all containers 😟
- No network ns isolation 😟
- Concurrent apps: some risks 😟

A good start, but...

as you grow...

bad things can happen

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

Common issues

- Bind TCP conflicts at same port
  Non stop code deployment?

- Port exhaustion risk when:
  TCP conns to same dst_ip:port
  > = ~500 new req/s
  default TIME_WAIT=60s

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 - @zenvdeluca

DOOOEEET RIGHT

FROM THE BEGINNING!

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

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

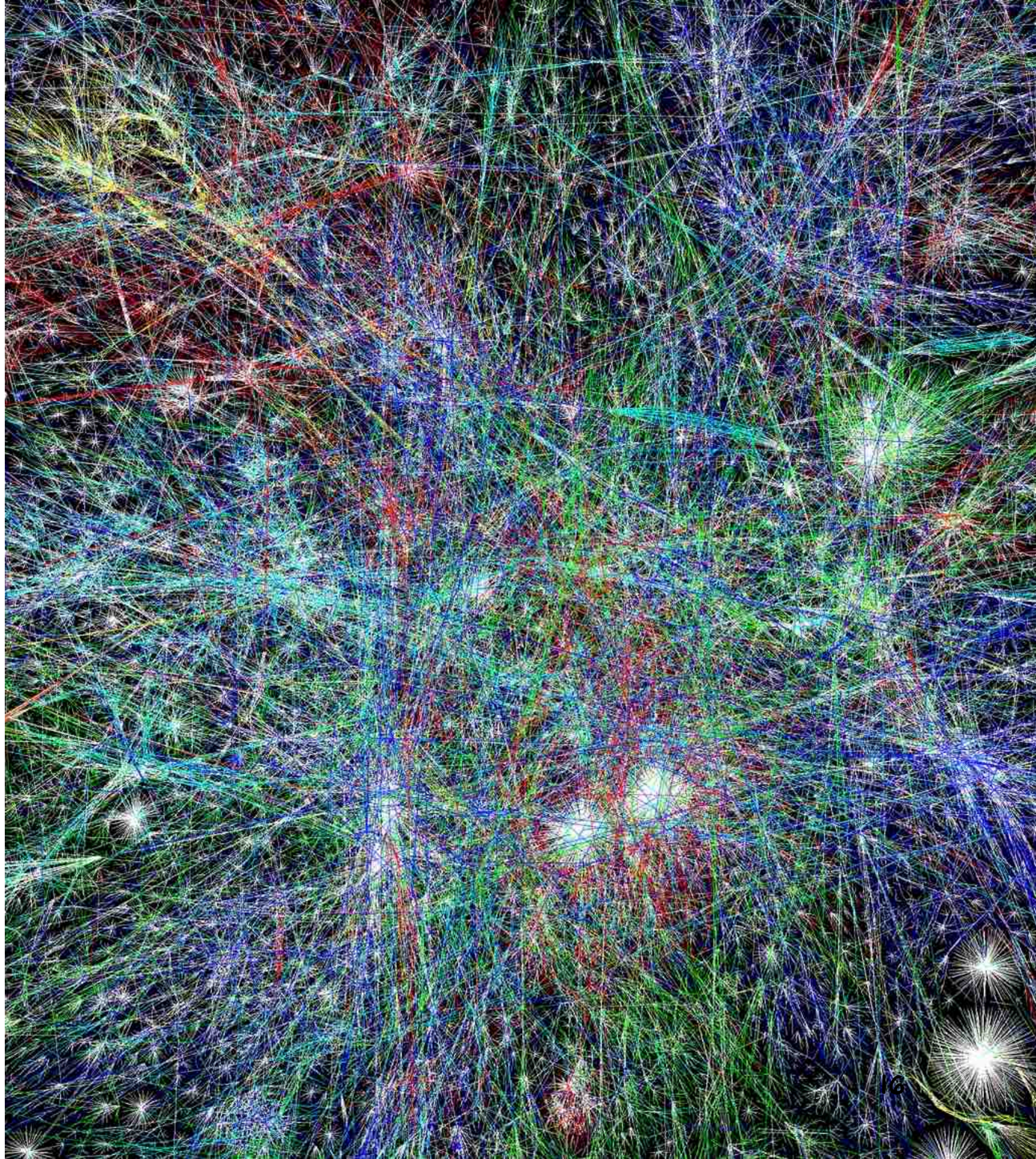
www.projectcalico.com

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 - @zenvdeluca

Stop chit chat, please?

Simple is Beautiful

DHCP:

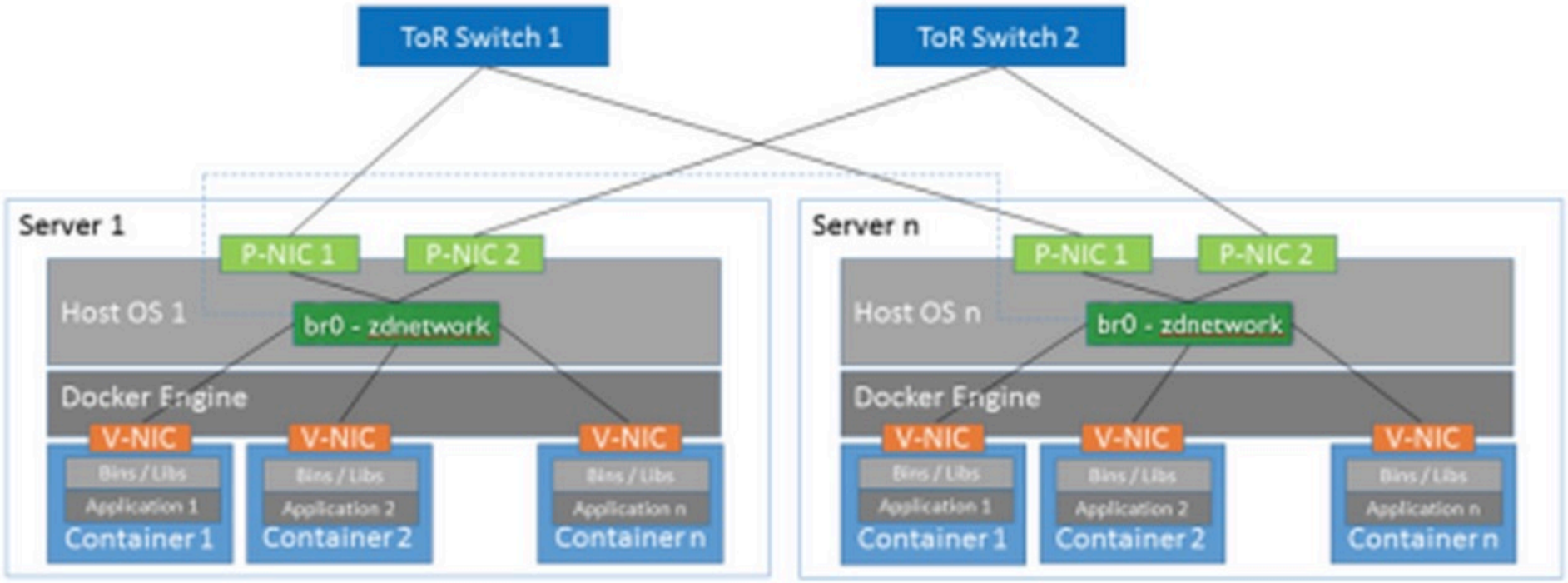
- reliable (RFC2131)
- everybody knows how it works

Bridge:

- Part of kernel since middle ages
- Replaceable by fancy vSwitches
- Network gears does L3 router 👍

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

Network Architecture



One IP per container

Ingredients:

- DHCP Server providing dynamic leases
- Bridge interface (br0) with eth0 attached
- One DHCP client per "RPM package on steroids"
- DHCP processes running at worker userspace
- A Network wrapper will do the magic

Network wrapper - #!/bin/bash

My Simplified fork from github.com/jpetazzo/pipework

-generate unique locally MAC
-Creates a new network namespace
-Associate the new veth with the our br0 interface
-Run and maintain the DHCP client
-Garbage collection for orphan DHCP processes

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

```
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

TL;DR

- no one architecture fits all needs
- currently everything is a trade off,
- but reliability / scaling is non-negotiable core value
- Learned (often the hard way) that significant floats in network architectures are not easy problems to solve.

More info about

- Tricking Docker daemon to provide IP per container
- Auto scaling load balancing using consul-template

check it out at my blog http://nethero.org
(or invite me for the next meetup 😄)

Codeship @codeship · 17m

"Even refrigerators and toasters have their own IP, why not my containers?" – @zendeluca

bit.ly/1FPBaZK

OPEN CHAT & QUESTIONS

THANK YOU!

2nd round starts now at the Barge

just across the street!

Sorry about our streaming fellows :(