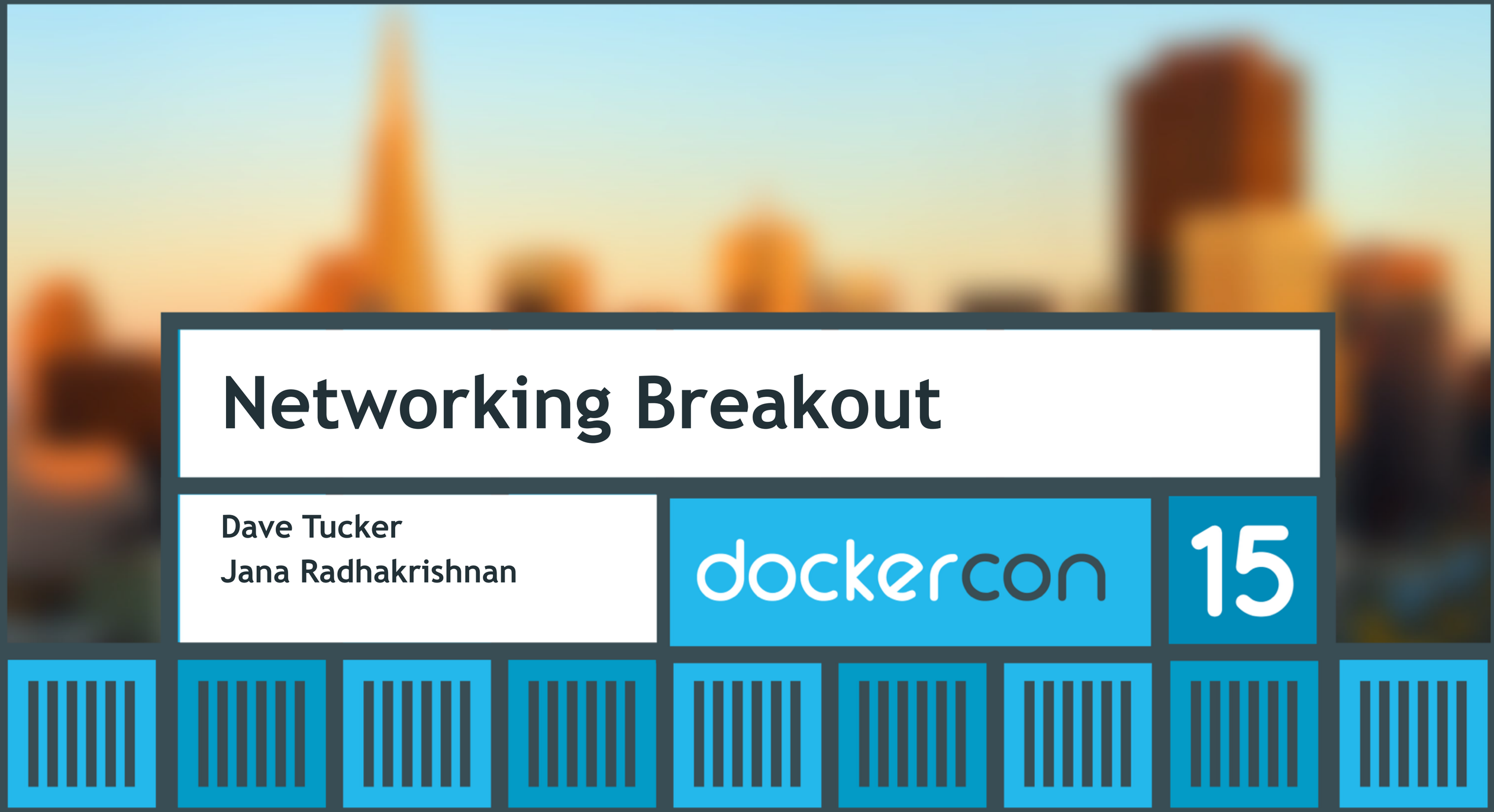


Networking Breakout

Dave Tucker
Jana Radhakrishnan

dockercon 15

Agenda

- Introduction
- Networking Deep Dive
- Ecosystem
- Q&A

Introduction

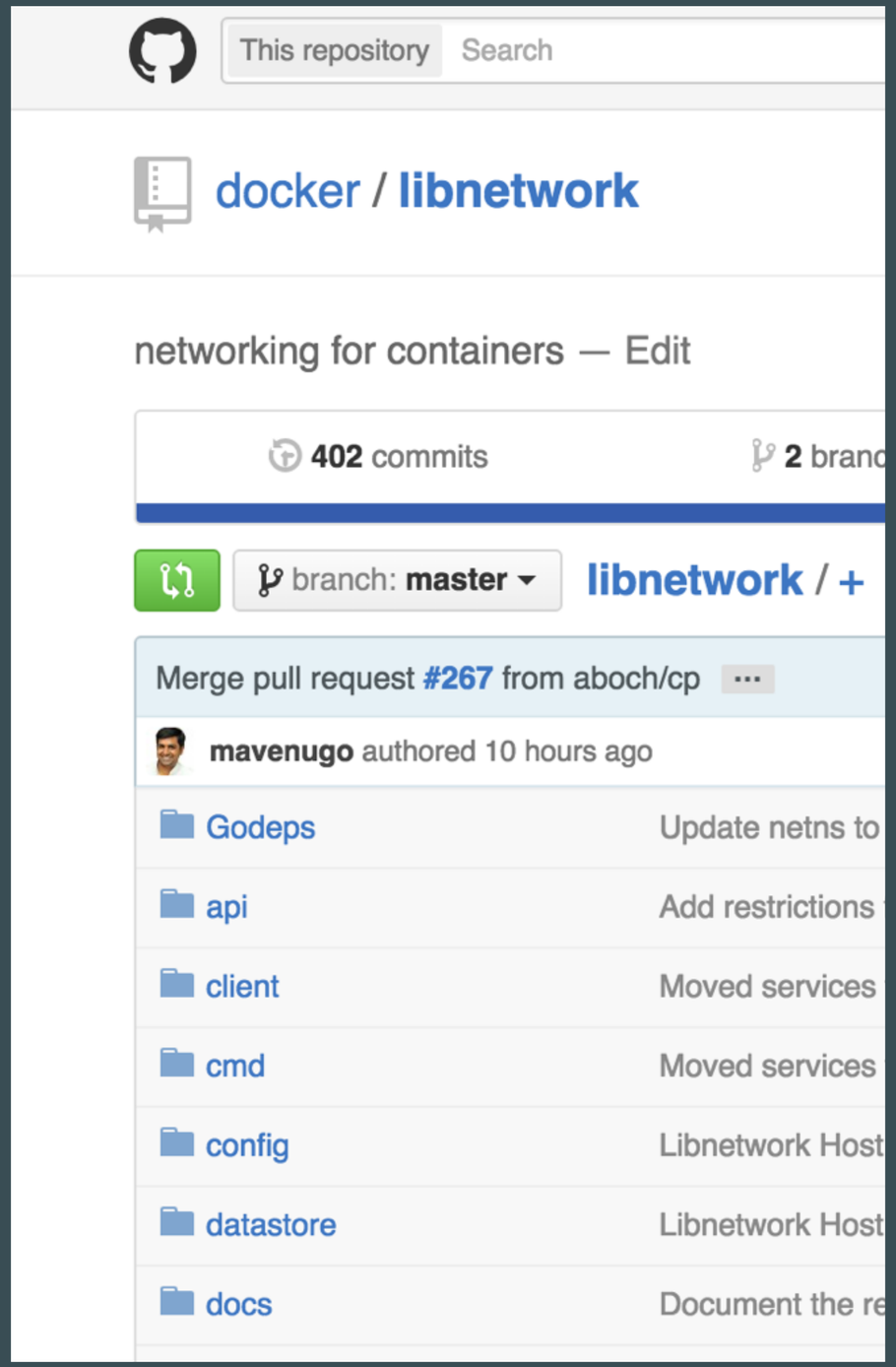
Why is Networking important?

- Networking itself is incredible vast and complex
- It's necessary to define networking for distributed applications consisting of micro-services
- BUT, it needs to be developer-friendly

“We'll do for Networking, What Docker did for Compute.”

Goals

- Make "**network**" a first class object
- Provide a **pluggable** networking stack
- Span networks across **multiple hosts**
- Support **multiple platforms**



Libnetwork

- Open Sourced in April
- Over 200 Pull Requests
- Over 200 GitHub Stars
- Windows and FreeBSD ports in progress

Whats New?

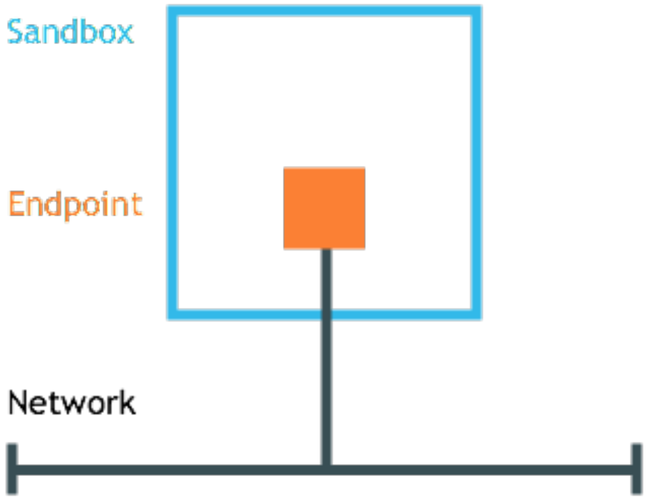
- Updated Networking Stack in Docker
- Create Networks using the Docker CLI
- Multi-host Networking
- Services UI

blue = experimental

Networking Deep Dive

What is Libnetwork

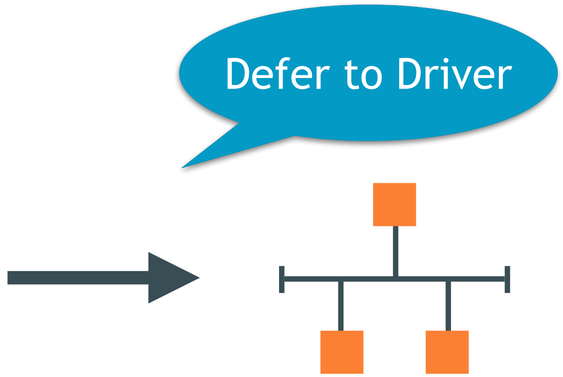
- Library for creating and managing network stacks for containers
- Test daemon/client called "dnet"
- Driver-based networking
- Implements the Container Network Model



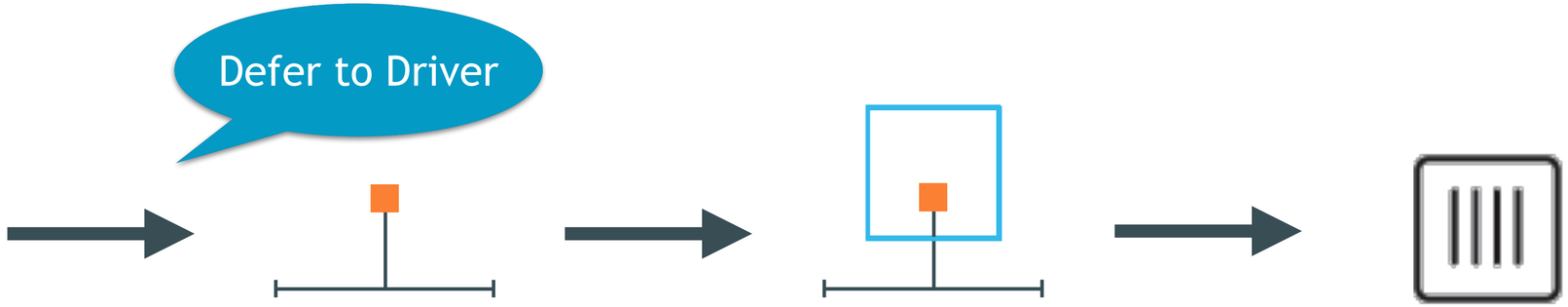
Container Network Model (CNM)

- Endpoint
- Network
- Sandbox

Create Network



Create Container



Libnetwork API

- libnetwork.New
- controller.ConfigureNetworkDriver
- controller.NewNetwork
- network.CreateEndpoint
- endpoint.Join

RESTful API

- Provides CRUD for Networks and Endpoints

- */network*
- */network/<network_id>/endpoints*
- */network/<network_id>/endpoints/<endpoint_id>*
- */network/<network_id>/endpoints/<endpoint_id>/containers*
- */services*
- */services/<service_id>*
- */services/<service_id>/backends*

Drivers

- Drivers implement the Driver API

- They provide the specifics of how a network and endpoint are implemented

Bridge Driver

- Creates a Linux Bridge for each network

- Creates a veth pair for each endpoint
  - One end is attached to the bridge
  - The other appears as eth0 inside the containers

- iptables rules created for NAT

Overlay Driver

- Creates a separate network namespace for every network
  - Facilitates overlapping IP address space across networks
- Creates a Linux Bridge and VXLAN tunnels to every other discovered host
- Creates a veth pair for each endpoint
  - One end is attached to the bridge
  - The other appears as eth0 inside the container
- Network namespace connected to host network using NAT
  - Facilitates exiting the overlay network at every host(for external connectivity)

Network Plugins

- Implemented using libnetwork's remote driver
- Uses JSON-RPC transport
- Can be written in any language
- Can be deployed as a container

Networking Ecosystem

“One size never fits all.”

– R. Callon, RFC 1925 - The Twelve Networking Truths

Microsoft

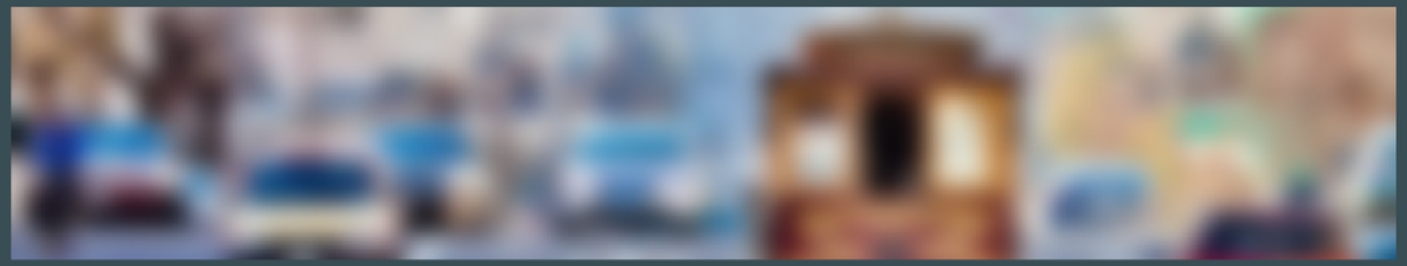
CISCO

vmware®

midokura

Call to Action!

- Try the Docker Experimental Channel!
  - https://experimental.docker.com
- Contribute to libnetwork
  - Raise an Issue or Submit a Pull Request
- Chat with us on IRC
  - #docker-network on Freenode
- Stop by at the booth for a demo



Q&A

Thank you

Dave Tucker

dt@docker.com
@dave_tucker

Jana Radhakrishnan

mrjana@docker.com
@mrjana