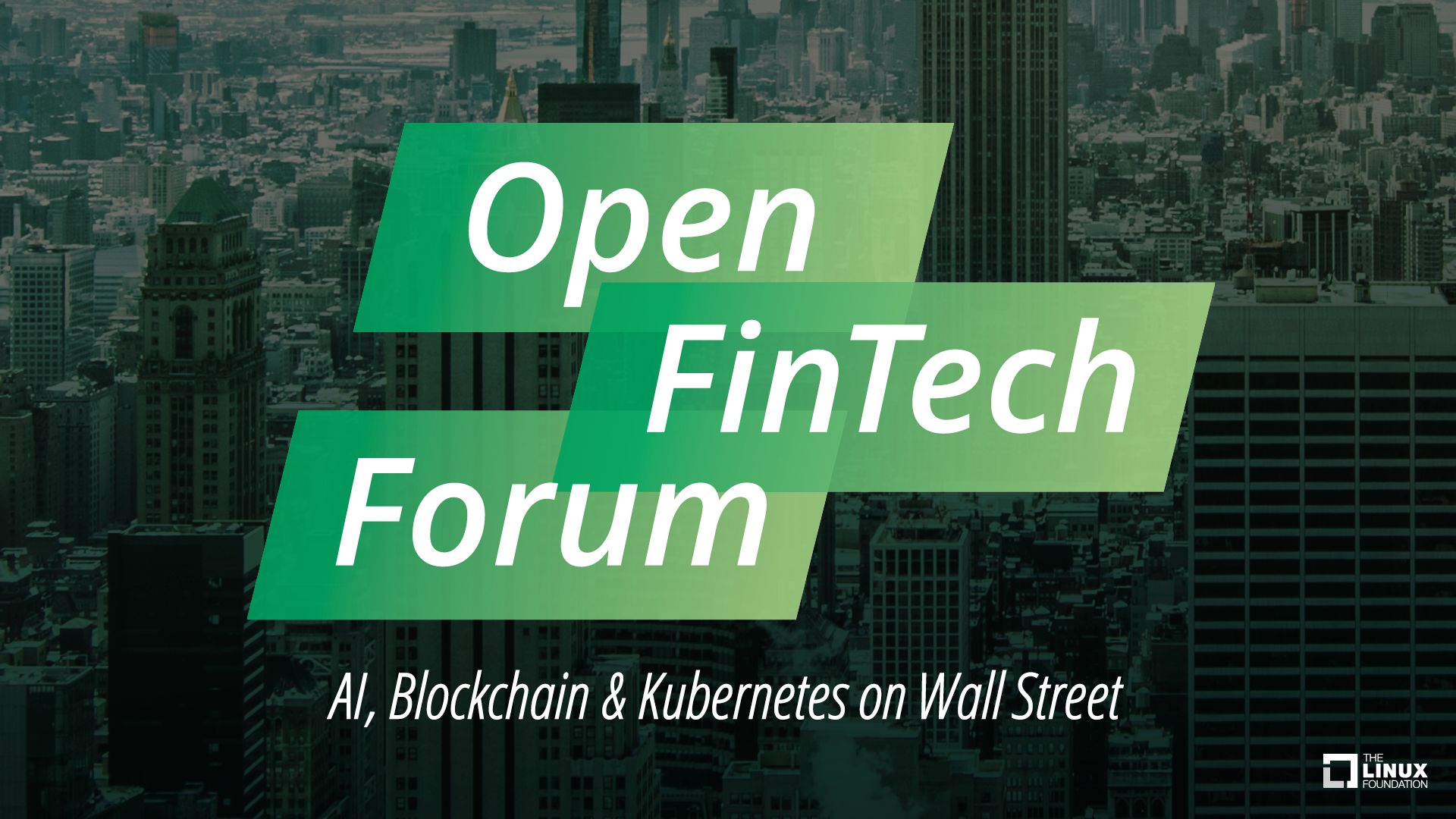
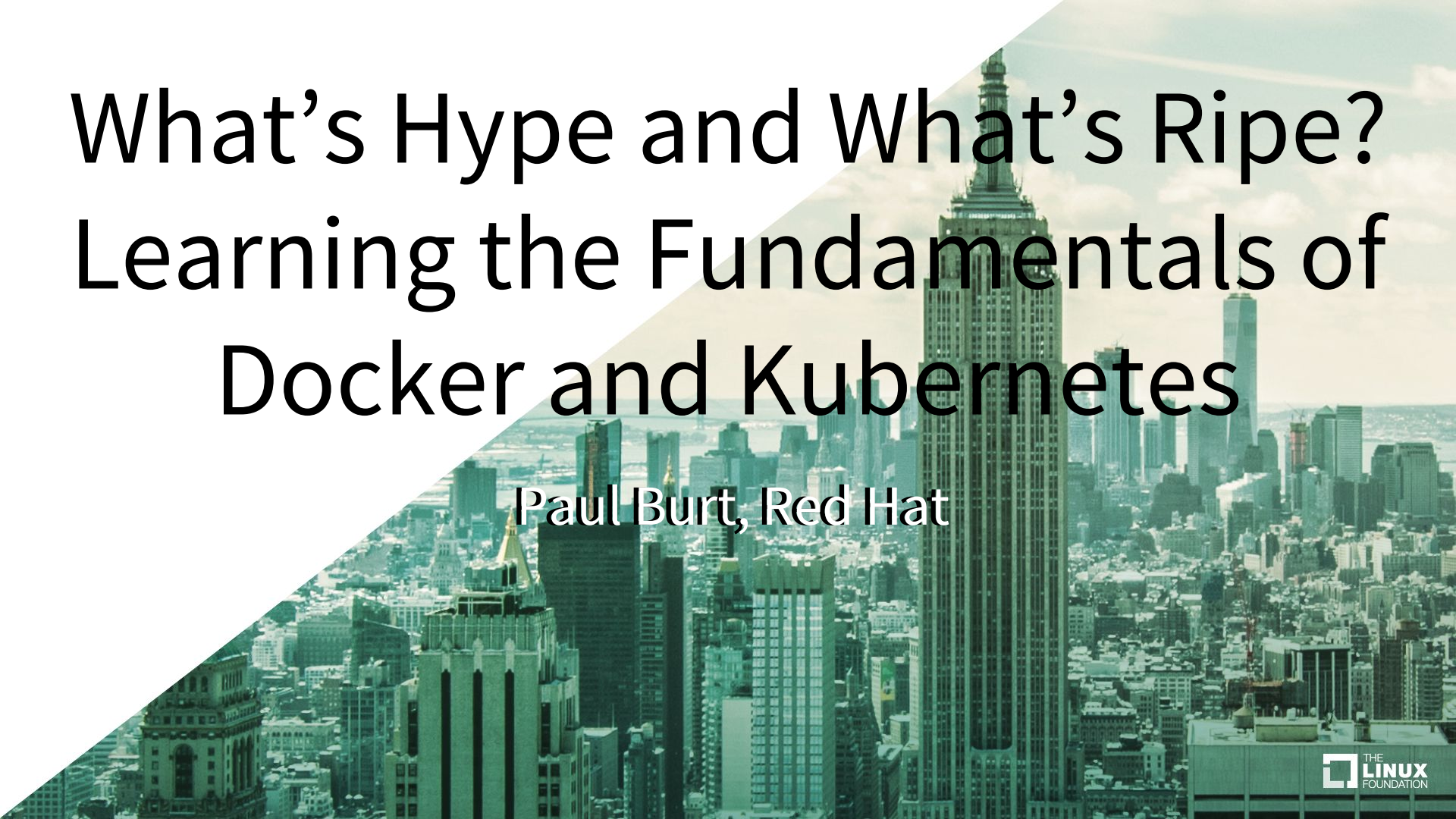

Open
FinTech
Forum
AI, Blockchain & Kubernetes on Wall Street
THE LINUX FOUNDATION



What's Hype and What's Ripe? Learning the Fundamentals of Docker and Kubernetes

Paul Burt, Red Hat

THE LINUX FOUNDATION

What is a container? Why are they catching on? In this introductory tutorial, we'll explore how to run a container on your laptop, in addition to answering these questions.

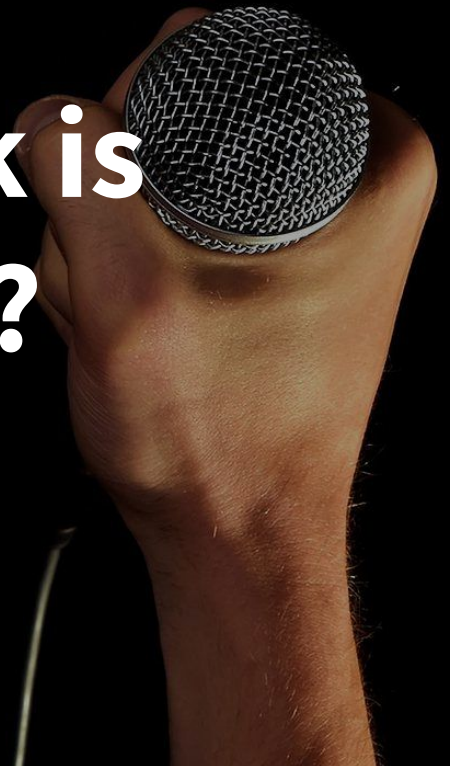
We'll look at how to build a container and the importance of building your own or using trusted containers. We'll look at recent studies on the health of "public registries", discuss what they are, and how you can easily avoid their security risks with a bit of pragmatism.

After that, we'll get our hands dirty looking at the tools required to build and run containers at a small scale. We'll answer questions like why do container deployments often grow to such a large scale, and what changes about the way we should run them when looking at a large scale? We'll explore the importance of automation, using the story of Knight Capital as a cautionary tale.

We'll look at common architectures, work through some tutorials for Docker and Kubernetes on katacoda, and map out the dizzying landscape that is the CNCF Landscape. The goal of this tutorial is to introduce you to containers if you're new to the scene and fill in gaps-in-knowledge for veterans of the container ecosystem.

What the heck is Kubernetes?

Slides: **goo.gl/oPUmcZ**



This talk is
Containers +
Kubernetes

puppy
@duckinator

Following

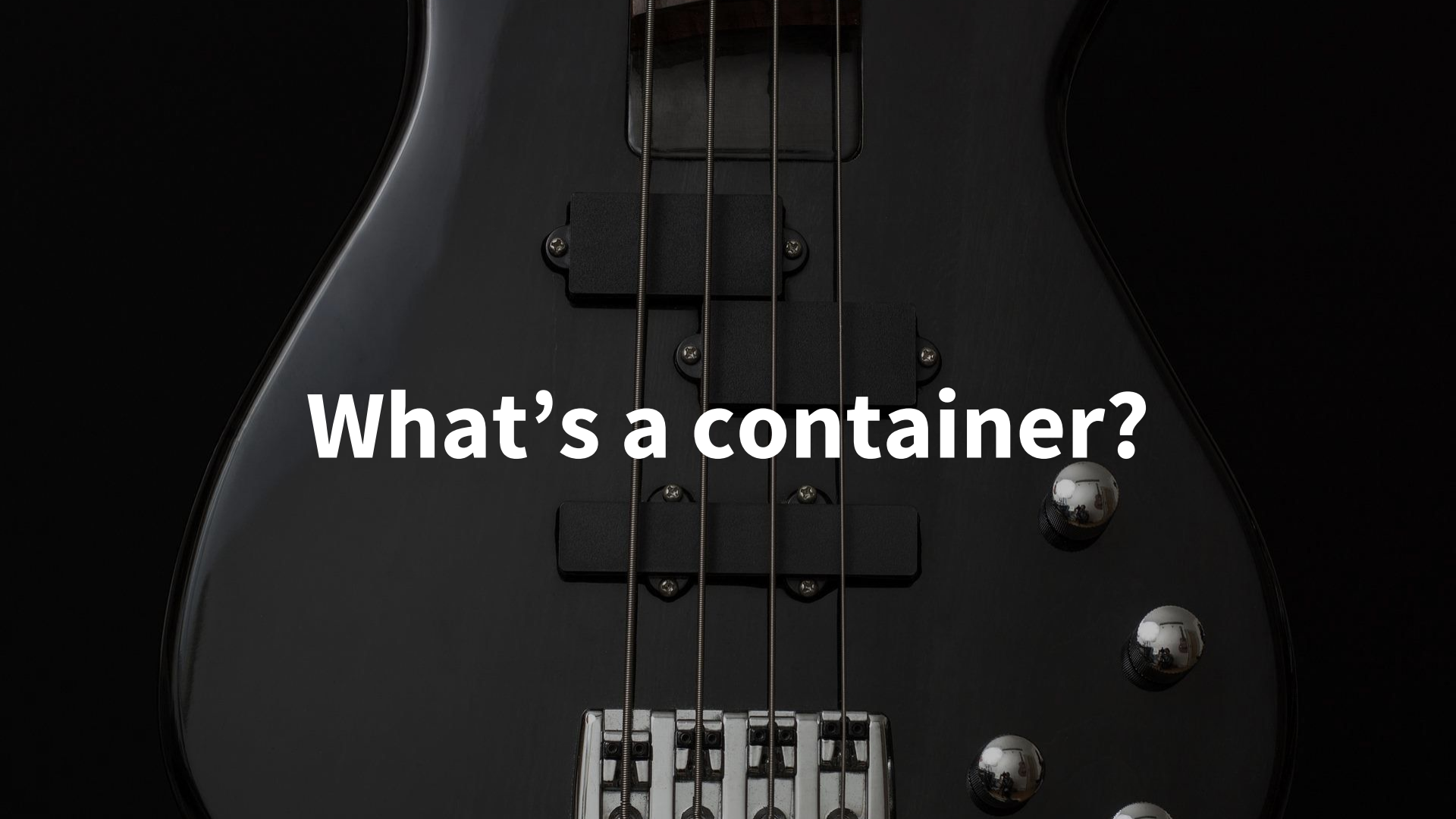
bashes their fists against the table while chanting

SERVERLESS BLOCKCHAIN CLOUD
SERVERLESS BLOCKCHAIN CLOUD
SERVERLESS BLOCKCHAIN CLOUD
SERVERLESS BLOCKCHAIN CLOUD
SERVERLESS BLOCKCHAIN CLOUD
SERVERLESS BLOCKCHAIN CLOUD
SERVERLESS BLOCKCHAIN CLOUD
SERVERLESS BLOCKCHAIN CLOUD

9:47 PM - 8 Feb 2018

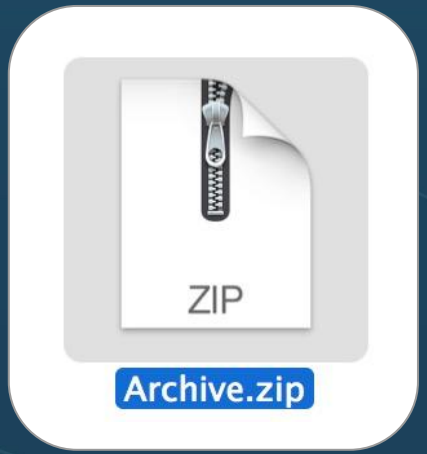
“Did anyone make a large purchase recently? A house, an expensive car? **How many of you made that purchase by just looking at the benefits, and not looking at the price?**”

– R. Meshenberg


What’s a container?

What is a container?

It’s a TAR file.



AKA

Containers are . . .

TAR files + Cgroups
Chroot
Unshare
Nsenter
Bind mounts


Linux
Magic
redhat.

For More on WHAT containers are

Containers From Scratch

By Eric Chiang

Best Practices for Containerized Environments

By Brian “Redbeard”

For History on containers see B. Cantrill's talk

goo.gl/1m1G3b

Why containers?

idk, why do ducks float?

Hell is other people's development environment

COMMUNITY

The Worst Computer Bugs in History: Losing $460m in 45 minutes

JAMIE LYNCH

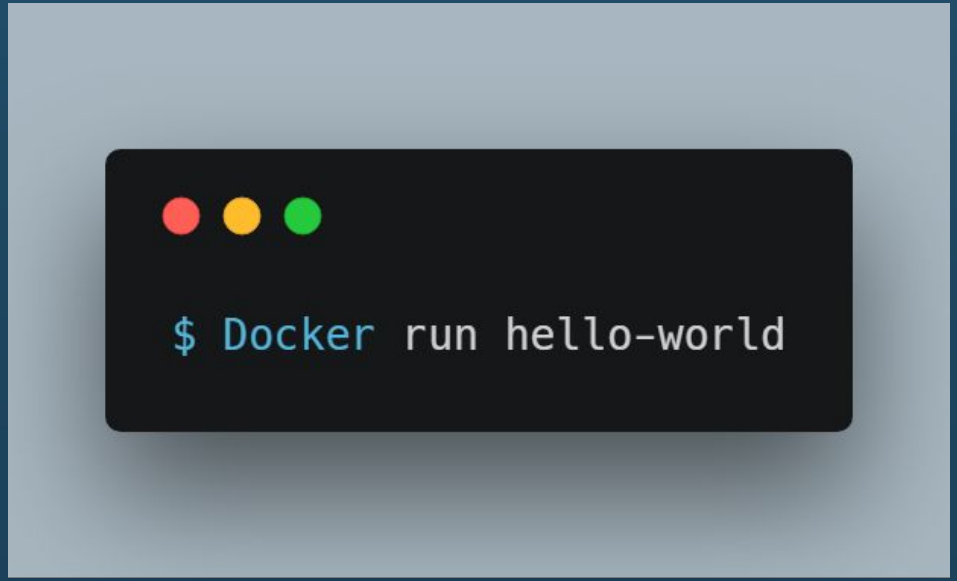
September 14th, 2017

Knight Capital Group

On August 1st, 2012, Knight Capital deployed a new software update to their production servers. At around 08:01AM, staff in the firm received 97 email notifications stating that *Power Peg*, a defunct internal system that was last used in 2003, was configured incorrectly.

TL;DR of how it works

Ain’t nobody got time to read specs



redhat

```
# Use an official Python runtime as a parent image
FROM python:2.7-slim

# Set the working directory to /app
WORKDIR /app

# Copy the current directory contents into the container at /app
COPY . /app

# Install any needed packages specified in requirements.txt
RUN pip install --trusted-host pypi.python.org -r requirements.txt

# Make port 80 available to the world outside this container
EXPOSE 80

# Define environment variable
ENV NAME World

# Run app.py when the container launches
CMD ["python", "app.py"]
```

The **runtime** manages the lifecycle of a container (including plugins)

An image format
A container runtime
A log collection daemon
An init system and process babysitter
A container image build system
A remote management API

redhat.

The OCI standards

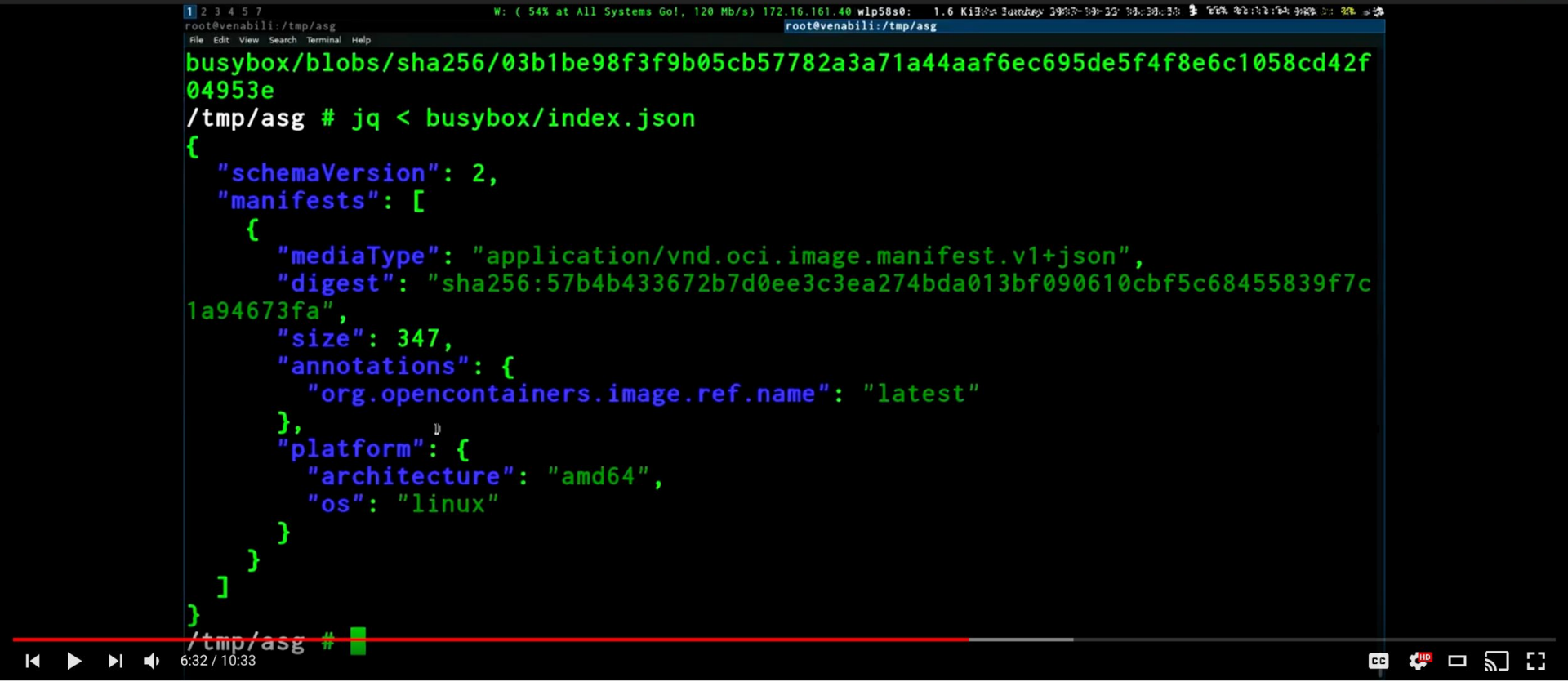
Two separate but connected specifications

- **image-spec**: what's in a container
- **runtime-spec**: how to run a container

redhat.

OCI Image Spec

- Portable archive format
- Composed of:
  - image manifest
  - image index (optional)
  - filesystem layers
  - configuration



2017 What's in a container? The OCI Answer

52 views

1 0 SHARE

Liked videos

CNI - the Container Network Interface

What is CNI?

CNI (*Container Network Interface*), a Cloud Native Computing Foundation project, consists of a specification and libraries for writing plugins to configure network interfaces in Linux containers, along with a number of supported plugins. CNI concerns itself only with network connectivity of containers and removing allocated resources when the container is deleted. Because of this focus, CNI has a wide range of support and the specification is simple to implement.

As well as the specification, this repository contains the Go source code of a library for integrating CNI into applications and an example command-line tool for executing CNI plugins. A separate repository contains reference plugins and a template for making new plugins.

CNI

CNI - the Container Network Interface

Zoom,

Enhance!

What is CNI?

CNI (*Container Network Interface*), a Cloud Native Computing Foundation project, consists of a specification and libraries for writing plugins to configure network interfaces in Linux containers, along with a number of supported plugins. CNI concerns itself only with network connectivity of containers and removing allocated resources when the container is deleted. Because of this focus, CNI has a wide range of support and the specification is simple to implement.

As well as the specification, this repository contains the Go source code of a library for integrating CNI into applications and an example command-line tool for executing CNI plugins. A separate repository contains reference plugins and a template for making new plugins.

CNI

CNI concerns itself only with network connectivity of containers and removing allocated resources when the container is deleted.

kubernetes

An open source system for automating deployment, scaling, and operations of applications.

Learn about Kubernetes

Monday, December 19, 2016

Introducing Container Runtime Interface (CRI) in Kubernetes

Subscribe To Blog

Posts

Comments

@Kubernetesio

View on GitHub

#kubernetes-users

Stack Overflow

Download Kubernetes

Blog Archive

▶ 2017 (49)

kubernetes

An open source system for automating deployment, scaling, and operations of applications.

Learn about Kubernetes

Monday, December 19, 2016

Introducing Container Runtime Interface (CRI) in Kubernetes

Subscribe To Blog

Posts

Comments

> “Docker and rkt were integrated directly and deeply into the kubelet source code through an internal and volatile interface.”

▶ 2017 (49)

POWER CUBE 1910/USRU4P
POWER CUBE USB by POWER C

Write a Review

Free shipping over $35 and Free Retu

~~$34.49~~

$15.95

Size

4 Outlets/2 USB

Buy 1	Buy 2	
$15.95/ea	**$15.15**/ea	**$1**

Add to Cart

CRI Timeline

Dec 12

1.5 alpha out

2017

Mar 28

1.6 Docker CRI gets beta + enabled by default

Jun 30

1.7 Docker CRI goes GA

That’s exciting...

In a Mom & Dad got me socks for X-mas kind of way

README.md

CRI-O - OCI-based implementation of Kubernetes Container Runtime Interface

Status: Stable

The speed of containers, the security of VMs

Kata Containers is a new open source project building extremely lightweight virtual machines that seamlessly plug into the containers ecosystem.

GET KATA CONTAINERS 1.3

Kata Containers 1.3 is here

Kata Containers 1.3 is here. Explore Kata Containers on GitHub.

CNI, CRI, and OCI - Oh My!

goo.gl/fK8kFS

So, that’s it?

But... How do we run containers?

I know! This is where we talk about Kubernetes.

I know! This is where we talk about Kubernetes.

Nope

Systemd is sufficient management for a single host

Unit file

On Container Linux, unit files are located at `/etc/systemd/system`. Let's create a simple unit named `hello.service`:

```
[Unit]
Description=MyApp
After=docker.service
Requires=docker.service

[Service]
TimeoutStartSec=0
ExecStartPre=-/usr/bin/docker kill busybox1
ExecStartPre=-/usr/bin/docker rm busybox1
ExecStartPre=/usr/bin/docker pull busybox
ExecStart=/usr/bin/docker run --name busybox1 busybox /bin/sh -c "trap 'exit 0' INT TERM; while true; do echo Hello
orld; sleep 1; done"

[Install]
WantedBy=multi-user.target
```

The `Description` shows up in the systemd log and a few other places. Write something that will help you understand exactly what this does later on.

`After=docker.service` and `Requires=docker.service` means this unit will only start after `docker.service` is active. You can define as many of these as you want.


So, why the excitement
about Kubernetes?

Distributed systems,
Immutable infrastructure,
CAP Theorem, Pets vs Cattle,
etc...

Fallacies of distributed computing

1. The network is reliable.
2. Latency is zero.
3. Bandwidth is infinite.
4. The network is secure.
5. Topology doesn't change.
6. There is one administrator.
7. Transport cost is zero.
8. The network is homogeneous.

Distributed systems are hard


So, what is Kubernetes?
(specifically)

```
$ cat <<EOF | kubectl apply -f -
apiVersion: "cache.example.com/v1alpha1"
kind: "Memcached"
metadata:
  name: "memcached-for-wordpress"
spec:
  size: 1
EOF
$ cat <<EOF | kubectl apply -f -
apiVersion: "cache.example.com/v1alpha1"
kind: "Memcached"
metadata:
  name: "memcached-for-drupal"
spec:
  size: 1
EOF
$ kubectl get Memcached
NAME                    AGE
memcached-for-drupal    22s
memcached-for-wordpress   27s
$ kubectl get pods
NAME                                    READY   STATUS    RESTARTS   AGE
memcached-app-operator-66b5777b79-pnsfj   1/1     Running   0          14m
memcached-for-drupal-5476487c46-qbd66     1/1     Running   0          3s
memcached-for-wordpress-65b75fd8c9-7b9x7  1/1     Running   0          8s
```

For nuts and bolts of K8s see

How Heptio engineers ace the certified kubernetes admin exam

By Ross Kukulinski

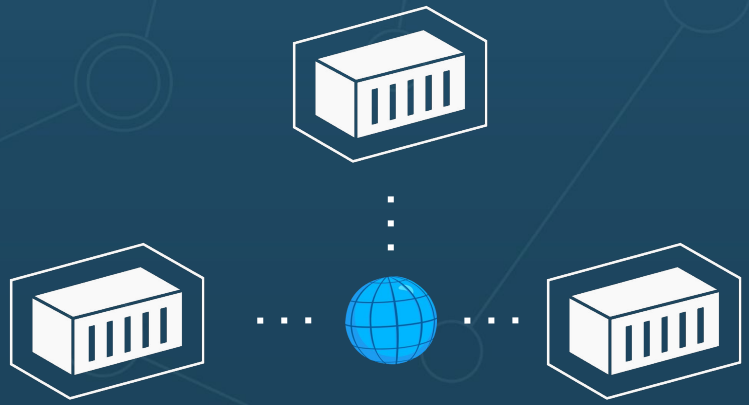
Kubernetes certified administrator GitHub

By Walid Shaari

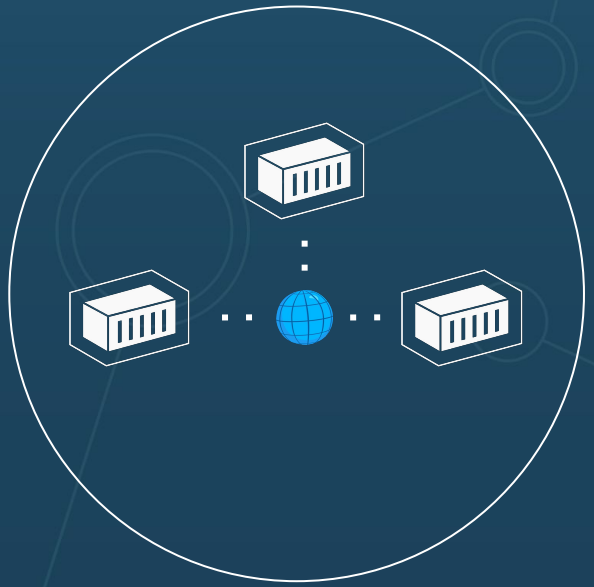
For our purposes, we can think of the machinery as the *control plane* and the *kubelet*

Popping the hood is different than driving

Pods and Services



redhat.



Give this a name

Design patterns for container-based distributed systems

goo.gl/MrJEj8

NIST Application Container Security Guide

goo.gl/bkmG7i

DIY Kubernetes

I assure you, Lord Vader,
my men are working as fast they can.

BEDROOM FURNITURE INSPIRATION

Where do you want to start your day?

Hello Darkness, my old friend...

Like, they can build buildings, but they fall apart after 6 months. JUST LIKE YOUR FURNITURE IKEA.

10 mins

Every country has a little tyranny, a little imperialism, Sweden's is IKEA.

Daniel Heinrich

They're laughing at our inability to pronounce the names of their furniture AND our inability to assemble simple bookshelves.

Daniel Heinrich

Like they have a TV channel: Americans assembling IKEA furniture.

Daniel Heinrich · 8 mins

To get started you need an **image registry + cluster**

Image Registries

redhat.

Tainted, crypto-mining containers pulled from Docker Hub

John Biggs @johnbiggs / 4 months ago

Comment

Security companies Fortinet and Kromtech found seventeen tainted Docker containers that were essentially downloadable images containing programs that had been designed to mine cryptocurrencies. Further investigation found that they had been downloaded 5 million times, suggesting that hackers were able to inject commands into insecure containers to download this code into otherwise healthy web applications. The researchers found the containers on **Docker** Hub, a repository for user images.

“Of course, we can safely assume that these had not been deployed manually. In fact, the attack seems to be fully automated. Attackers have most probably developed a script to find misconfigured Docker

Clusters

\+ Every Cloud
and Linux vendor

redhat.

NullPointerException on Kubernetes 1.6.1 #1219

Closed **lkysow** opened this issue on Apr 17, 2017 · 9 comments

lkysow commented on Apr 17, 2017 • edited

Linkerd Versions: 0.9.0 and 0.9.1 have same error
Kubernetes Versions: 1.6.1, 1.6.0

Stack trace on linkerd 0.9.0

```
E 0417 21:47:41.726 UTC THREAD21: k8s failed to list endpoints
java.lang.NullPointerException
        at io.buoyant.k8s.EndpointsNamer$.io$buoyant$k8s$EndpointsNamer$$getAddrs(Endpoints
        at io.buoyant.k8s.EndpointsNamer$.io$buoyant$k8s$EndpointsNamer$$mkPorts(EndpointsN
        at io.buoyant.k8s.EndpointsNamer$NsCache$$anonfun$io$buoyant$k8s$EndpointsNamer$NsC
```

CRYPTOCURRENCY JACKING —

Tesla cloud resources are hacked to run cryptocurrency-mining malware

Crooks find poorly secured access credentials, use them to install stealth miner.

DAN GOODIN - 2/20/2018, 2:21 PM

Let’s get our hands dirty with Kubernetes

learn.openshift.com

Slides: **goo.gl/oPUmcZ**

Learn Docker & Containers using Interactive Browser-Based Scenarios

By Ben Hall

Solve real problems and enhance your skills with browser based hands on labs without any downloads or configuration

Get Started!

Scenarios Completed
0 of 21

Progress
0%

Points
0

Create Your Free Account

Deploying Your First Docker Container

Learn how to launch containers using Docker

Start Scenario

Deploy Static HTML Website as Container

Learn how to run a static HTML website using Nginx

Start Scenario

Building Container Images

Learn how to build and launch your own container images

Dockerizing Node.js

Learn how to deploy Node.js applications as containers

Optimise Builds With Docker OnBuild

Ignoring Files During Build

Learn how to ignore files being sent to the Docker Build