Cloud Native Buildpacks

Sridhar Vennela
Kevin Takenaga

Cloud Native Buildpacks (CNB) Bring Developer Productivity to K8s

Pluggable, modular tools that translate source code into OCI images.

- Portability via the OCI standard
- Greater modularity
- Faster builds
- Reproducible image builds
- Unprivileged containers
- Collaboration with Heroku
- **CNCF Sandbox project**

OCI Image Specification

Established in June 2015 by Docker and other leaders in the container industry, the OCI currently contains two specifications: the Runtime Specification (runtime-spec) and the Image Specification (image-spec). The Runtime Specification outlines how to run a "filesystem bundle" that is unpacked on disk. At a high-level an OCI implementation would download an OCI Image then unpack that image into an OCI Runtime filesystem bundle. At this point the OCI Runtime Bundle would be run by an OCI Runtime.

https://www.opencontainers.org/

History of Buildpacks

- Buildpacks were first created by Heroku in 2012 (*v1*)

- Cloud Foundry adopted buildpacks with modifications (*v2b*) and Heroku modified their original spec (*v2a*)

- In 2018, buildpack teams from both Pivotal and Heroku came together to create a new open and formal standard for buildpacks called **Cloud Native Buildpacks** and submitted the project for inclusion into the Cloud Native Computing Foundation. It's currently a sandbox project.
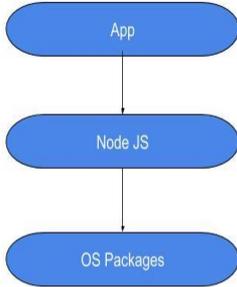
Case Study

Security agent in the container

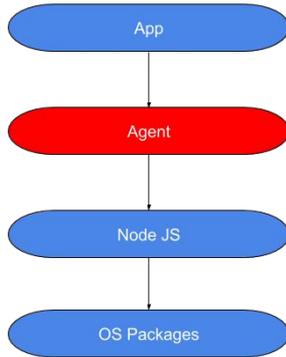# Node.js App

FROM nodejs:latest
COPY myapp .
RUN npm install

...
FROM ubuntu:trusty
RUN wget http://nodejs.org

FROM ubuntu:16.04
…

Node.js App

App

↓

Agent

↓

Node JS

↓

OS Packages

FROM nodejs:latest
COPY myapp .
RUN npm install
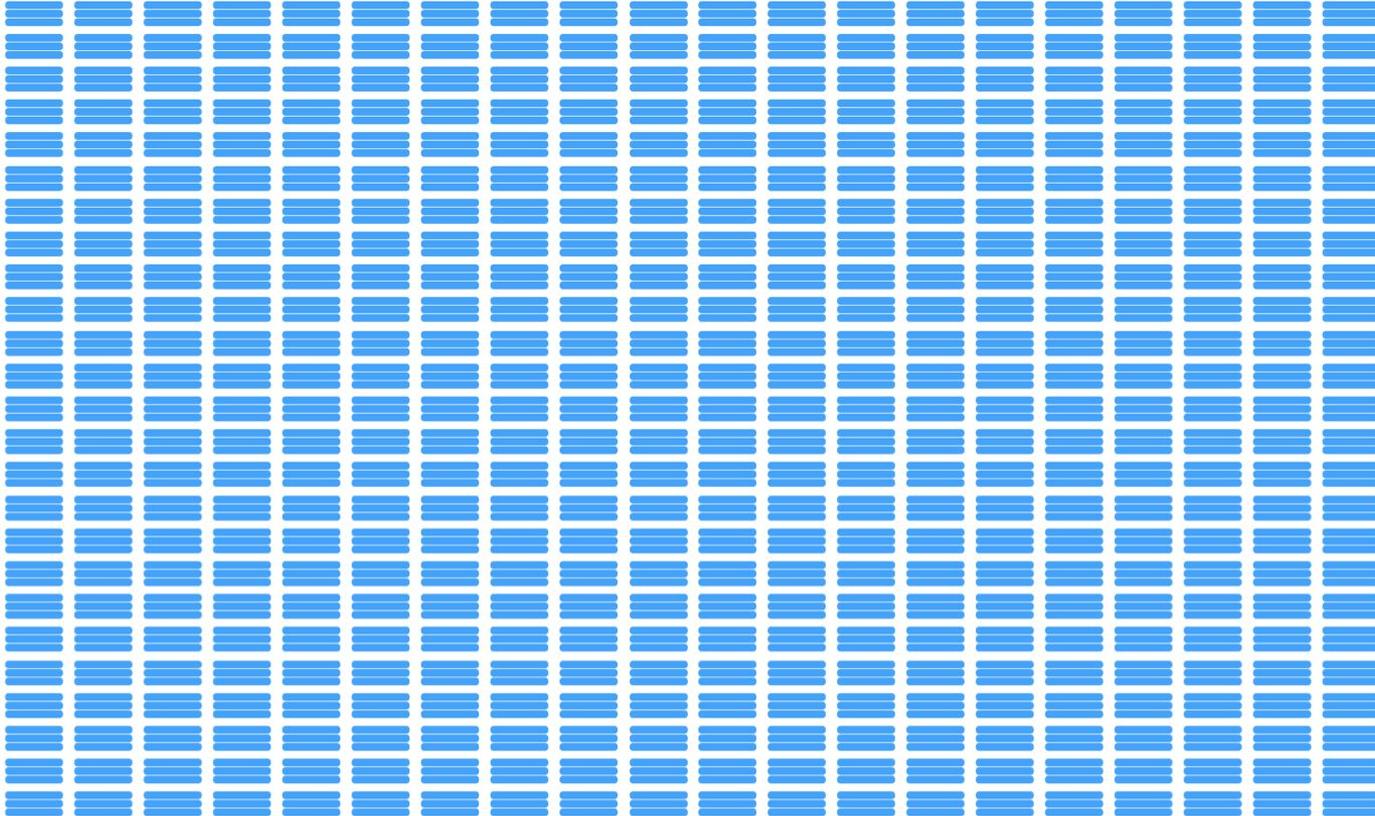
...

RUN wget http://securityagent.com
RUN Install securityagent

...

FROM ubuntu:trusty
RUN wget http://nodesjs.org
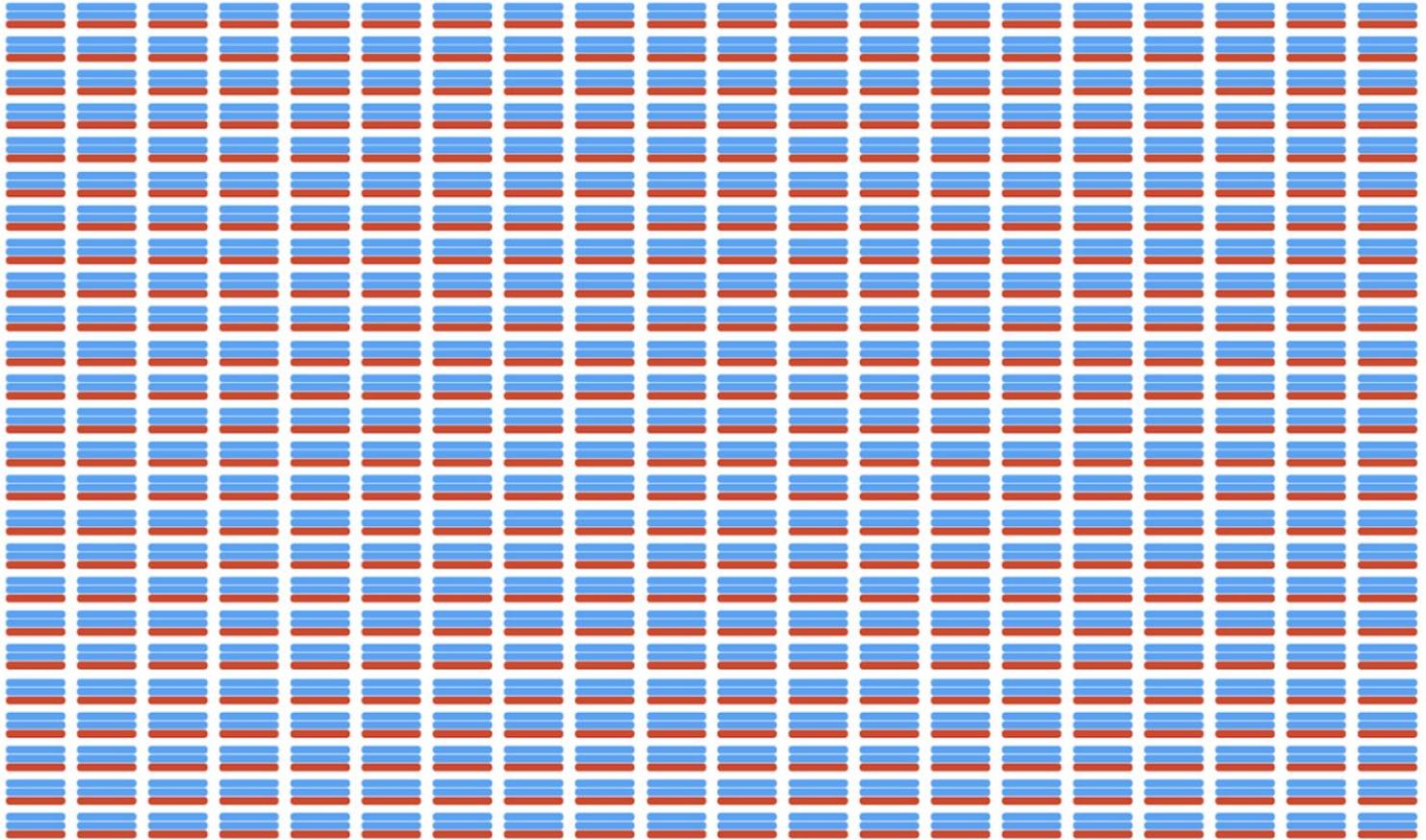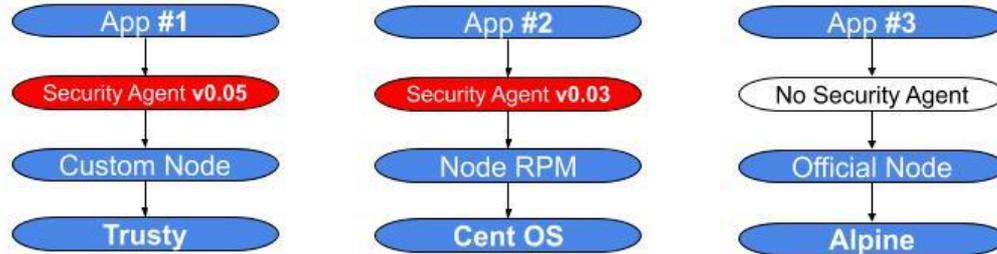
FROM ubuntu:16.04

...

500 Node.js Apps

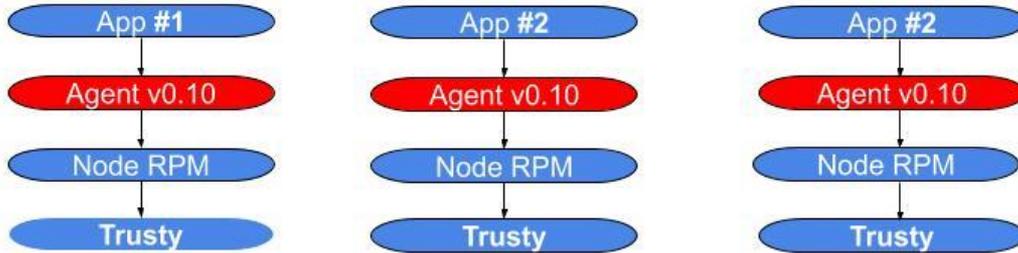# 500 Node.js Apps that need a Security Agent!

Dockerfiles: Done Wrong (individually managed)

ETA to mitigate

Months? Years?

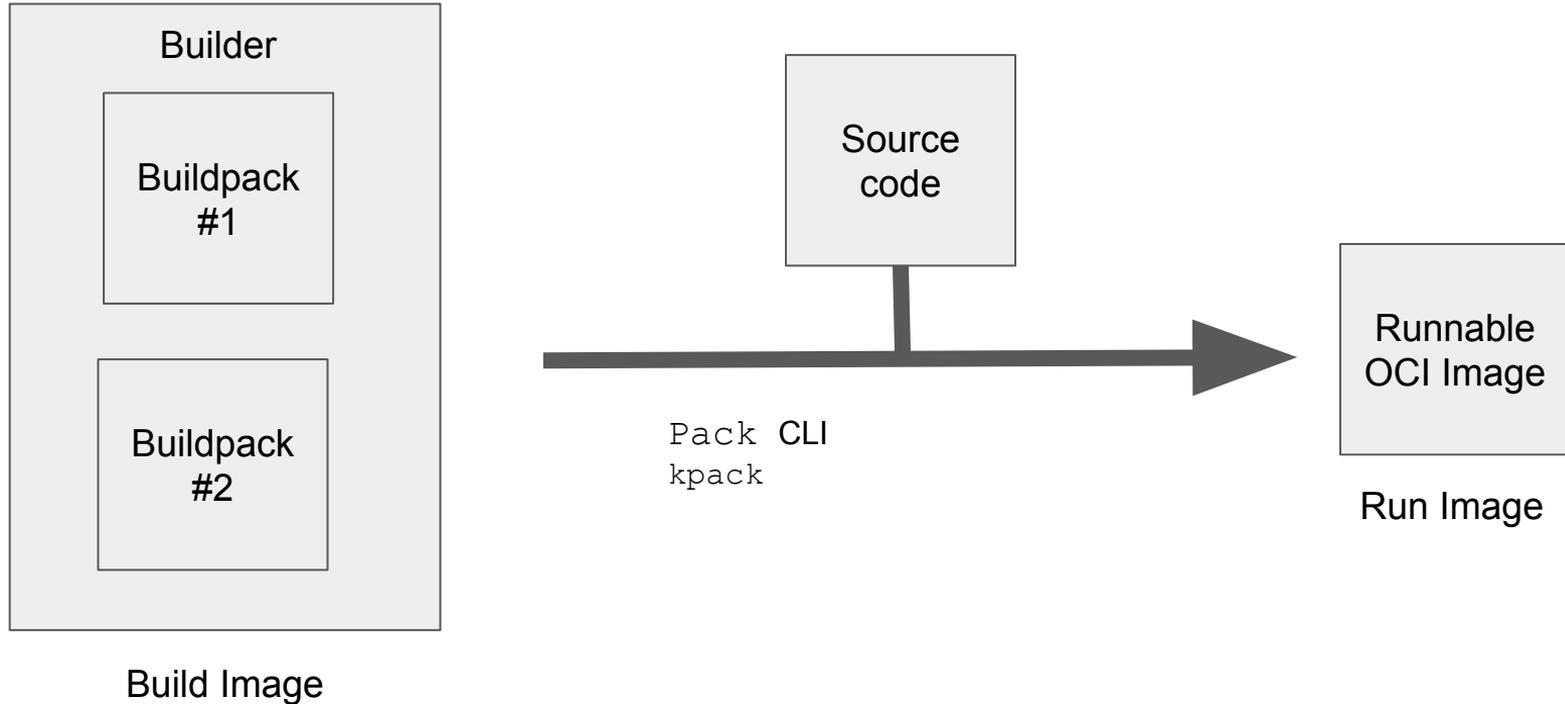# Dockerfiles: Done Right (Operator managed)

ETA to mitigate

Months? Years?

What's the benefit of Cloud Native Buildpacks?

Cloud Native Buildpacks allow you to package your app in a container without using a Dockerfile.

The following demos will show how this is done.

How Cloud Native Buildpacks Work

Builder

Buildpack #1

Buildpack #2

Build Image

Source code

Pack CLI
kpack

Runnable OCI Image

Run Image

Demos

Demo #1

Build a Node JS app using the `pack` cli

Demo #2

Build a Buildpack & Builder for Security Agent

Demo #3

Use Builder to Build Node JS App