Simplify Your Cloud Native Application Packaging and Deployments

Chris Crone

Chris Crone

Developer tooling

docker

Agenda

- What is a cloud native application?
- Challenges with packaging and deploying apps
- Introduction to Cloud Native Application Bundles
- Demo
- CNAB for application packaging and deployment
- Where to learn more
- Questions?

What is a cloud native application?

"*A program or piece of software designed to fulfil a particular purpose*" – Oxford English Dictionary

A cloud native application?

- Compute
  - Containers
  - Functions (AWS Lambda, Azure Functions, etc.)
  - Virtual machines
- Storage
  - Databases
  - Object storage
  - Volumes
- Networking

CNCF Cloud Native Landscape
2020-07-25T00:25:36Z e181c216

Overwhelmed? Please see the CNCF Trail Map. That and the interactive landscape are at l.cncf.io

Greyed logos are not open source

App Definition and Development

Database

Streaming & Messaging

Application Definition & Image Build

Continuous Integration & Delivery

Orchestration & Management

Scheduling & Orchestration

Coordination & Service Discovery

Remote Procedure Call

Service Proxy

API Gateway

Service Mesh

Runtime

Cloud Native Storage

Container Runtime

Cloud Native Network

Provisioning

Automation & Configuration

Container Registry

Security & Compliance

Key Management

Platform

Certified Kubernetes - Distribution

Certified Kubernetes - Hosted

Certified Kubernetes - Installer

PaaS/Container Service

Observability and Analysis

Monitoring

Logging

Tracing

Chaos Engineering

Serverless

Special

Kubernetes Certified Service Provider

Kubernetes Training Partner

Members

CLOUD NATIVE Landscape

CLOUD NATIVE COMPUTING FOUNDATION

Redpoint

Amplify PARTNERS

l.cncf.io

This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF Projects representing a particularly well-traveled path.



KubeCon | CloudNativeCon

Europe 2020

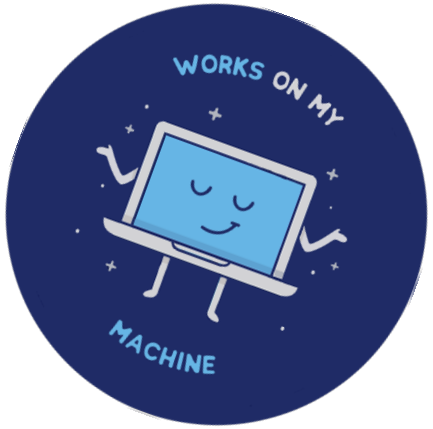
Virtual

Deploying applications

```
$ less README.md
$ bash ./deploy.sh
Error: command not found: terraform
$ curl -Lo terraform https://…
$ bash ./deploy.sh
Error: unknown option --deprecated-option
$ curl -Lo terraform-old https://…
$ bash ./deploy.sh
Error: command not found: jq
```

Deployment tooling

- Often need more than one tool to deploy an application
- Is the README up to date?
- Which version of the tools?
- What if I'm using Windows and not Linux?
- Difficult coordination problem between team members, CI, users



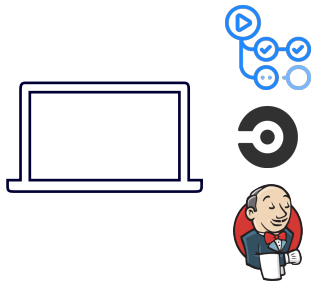
Ideal deployment tooling

- Defined as code: tools, versions, options
- Same deployment environment everywhere

Packaging applications

Different parts, different places

GitHub



docker hub

app/compose.yaml

```
services:
  front:
    image: user/front:1.19
    ports:
      - "80:80"
  back:
    image: user/myapp:1.1
```

```
$ docker

$ terraform

$ helm
```

user/front:1.19

user/myapp:1.1

Application definition

Deployment tooling

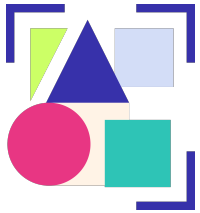
Application components

Ideal application packaging

- Immutable application artifact
- Store the whole application in a registry
- Ability to store application artifact offline

Cloud Native Application Bundles

CNAB

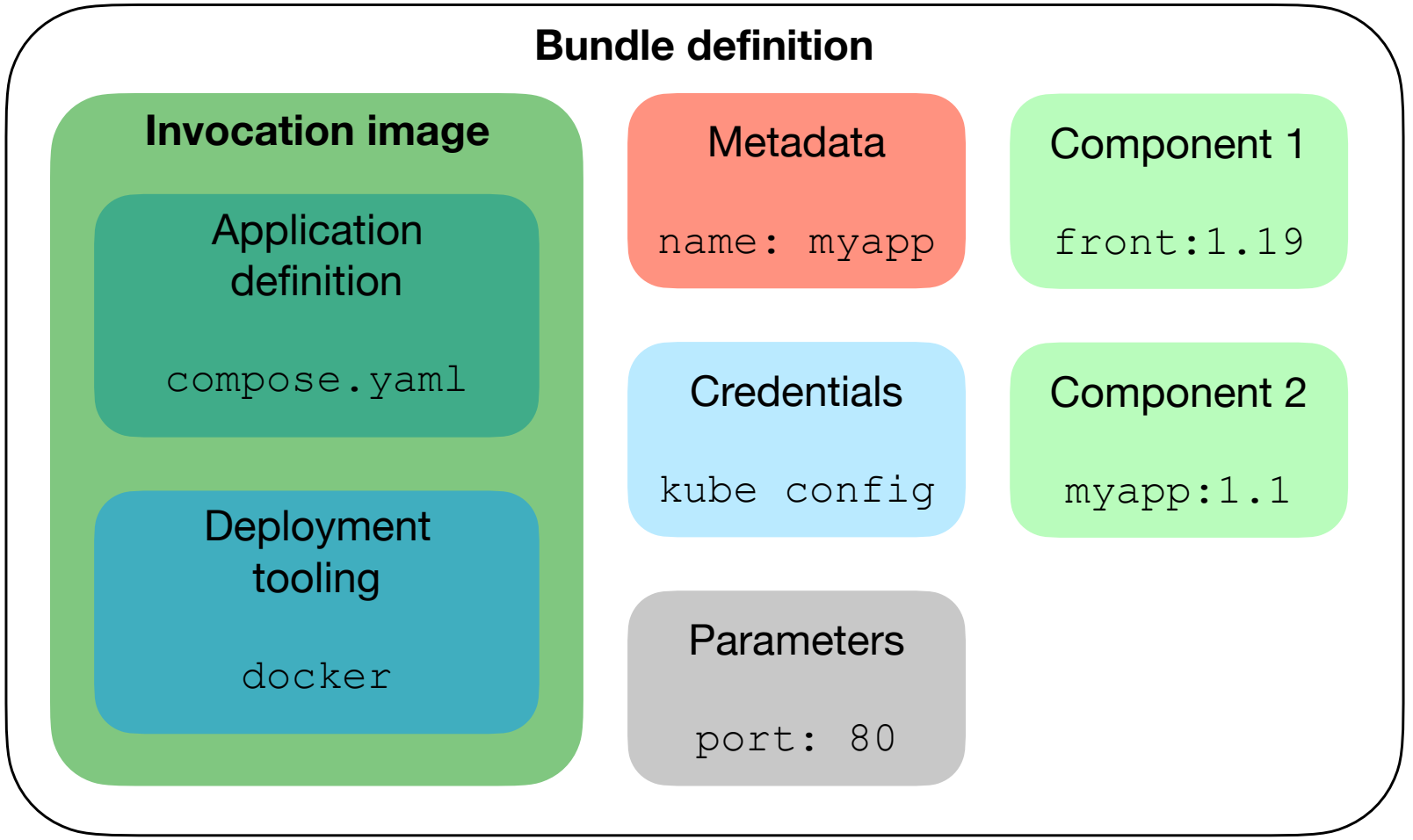


Cloud Native Application Bundles (CNAB) are a package format specification that describes a technology for bundling, installing, and managing distributed applications, that are by design, cloud agnostic.

CNAB specification

- Target is tooling developers
- Packaging specification (bundle)
- Bundle runtime (actions)
  - Install, upgrade, uninstall
- Optionally
  - Lifecycle tracking (claims)
  - Registry storage
  - Security
  - Dependencies

Bundle structure



CNAB runtime

- Standard actions: install, upgrade, uninstall
- Custom actions
  - e.g.: status, logs
  - Stateful/stateless
- Application lifecycle tracked by “claims”
  - Keep track of state of installations
  - Keep record of parameters, outputs, etc.
  - Only data structure defined in specification

Demo


CNAB for deployment

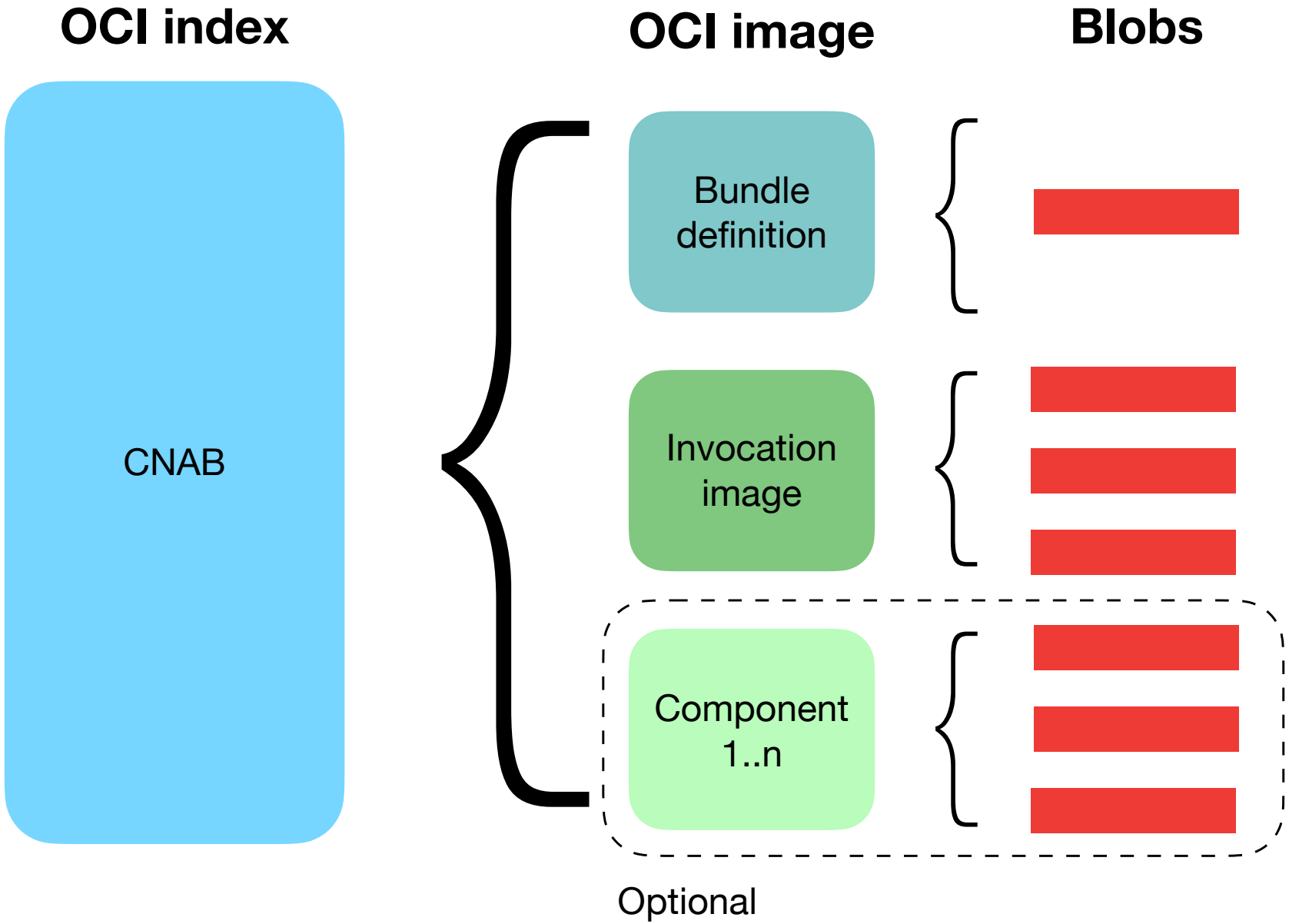
```
$ less porter.yaml
$ porter install myapp --tag acme/app:v0.1.0
```

Ideal deployment tooling

- Defined as code: tools, versions, options
  - porter.yaml
  - Stored in CNAB invocation image
- Same deployment environment everywhere
  - Containers!

CNAB for packaging

CNAB in registries



Different parts, same place

app/compose.yaml

```
services:
  front:
    image: user/front:1.19
    ports:
      - "80:80"
  back:
    image: user/myapp:1.1
```

Application definition

```
$ docker
$ terraform
$ helm
```

Deployment tooling

user/front:1.19

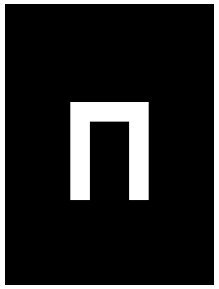
user/myapp:1.1

Application components

Ideal application packaging

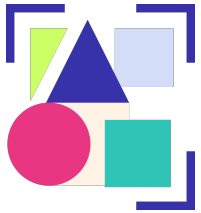
- Immutable application artifact
  - Hashes for components
  - Leverage OCI image specification
- Store the whole application in a registry
  - Any OCI compliant container registry
- Ability to store application artifact offline
  - OCI image layout

CNAB security



- Leverage same mechanisms as containers
  - TUF
  - In-toto
  - Notary
- Reference tooling: signy
  - github.com/cnabio/signy

Learn more



- CNAB website: cnab.io
- Demo code: github.com/chris-crone/kubecon-eu-20
- Porter: porter.sh
- Interested in storing things in registries?
  - Sharing Is Caring! Push Your Cloud Application to an OCI Registry (sched.co/Zemr)
    - Silvin Lubecki and Djordje Lukic, Docker
  - Where to Put All That YAML: Secure Content Management for Cloud Native Apps (sched.co/Zeiq)
    - Ryan Abrams, Stripe



KubeCon | CloudNativeCon

Europe 2020

Virtual

Questions?

Thank you!


KubeCon
CloudNativeCon
Europe 2020
Virtual
HELM
KEEP CLOUD NATIVE
CONNECTED