Cloud Native Buildpacks

A simple *commit* to sail to *production*

Kubernetes Native (kpack) - https://github.com/pivotal/kpack

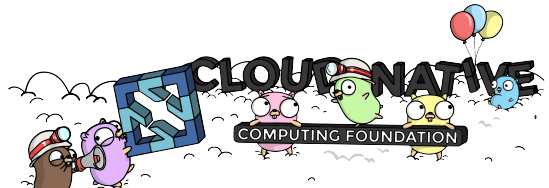
https://github.com/buildpacks

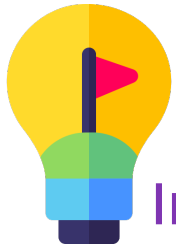
https://github.com/paketo-buildpacks

Open-Source Enthusiast & Contributor
Love Travel & Photography

Srinivasa Vasu
Solution Engineer @ VMware

@humourmind

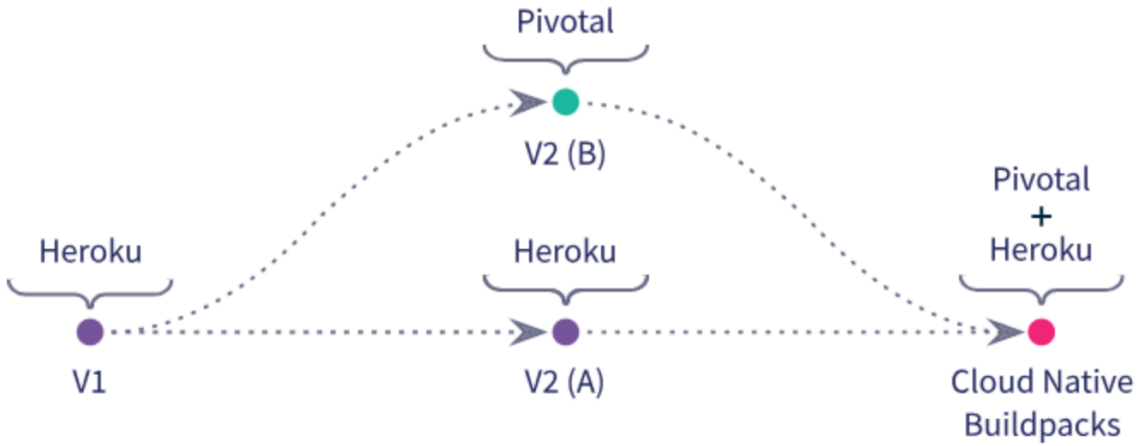




Incubated

Cloud Native Buildpacks

transform your application **source code** into **images** that can run on any cloud



Source: https://buildpacks.io/

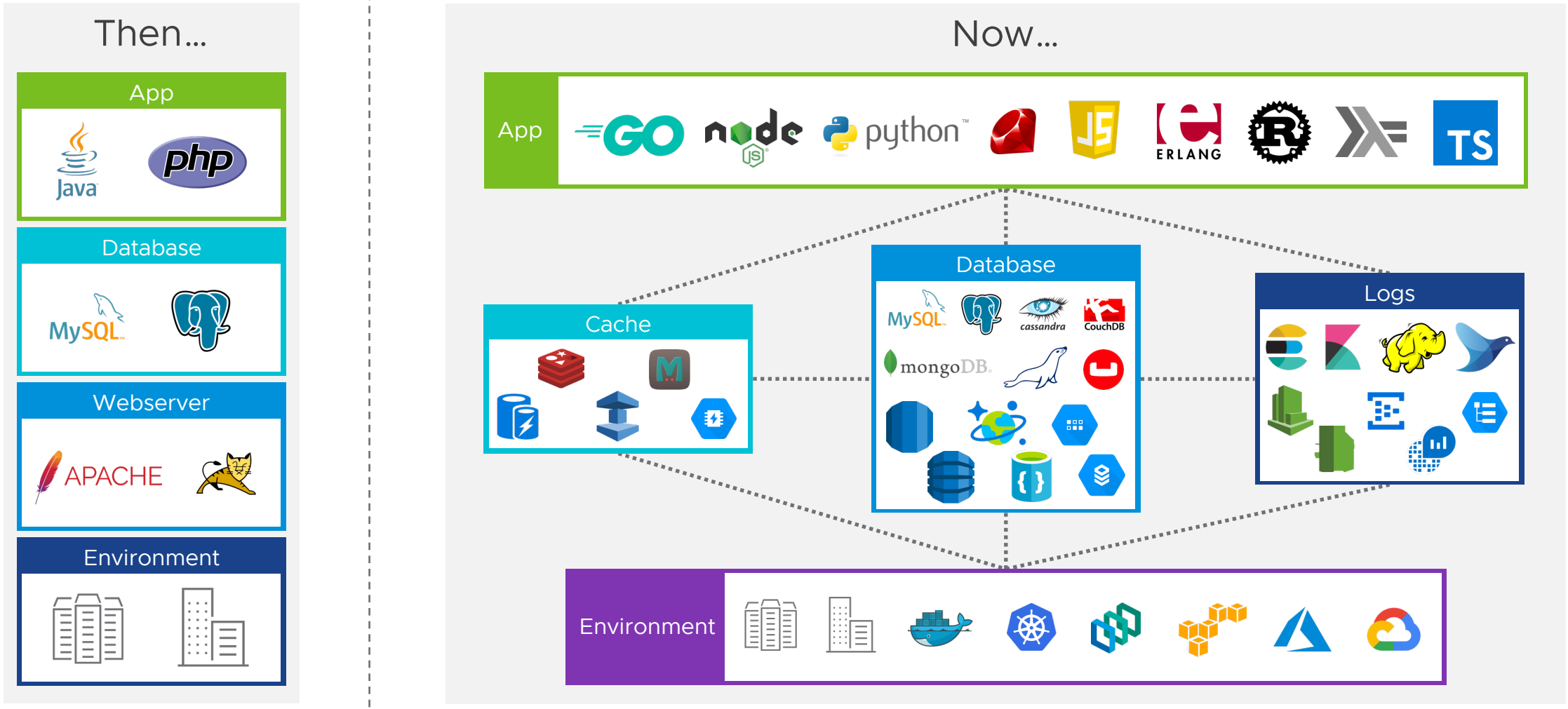
What problem does cloud native buildpacks solve?

Problem

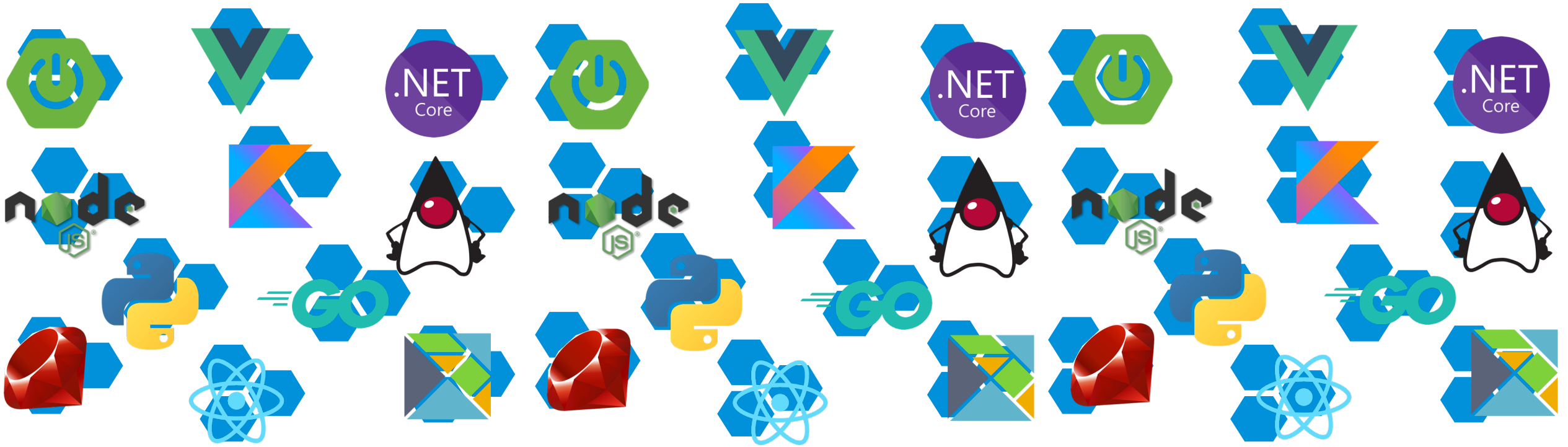
Source (Character art): https://phippy.io/

Application architectures are changing

We've moved from simple and centralized, to complex and decentralized.



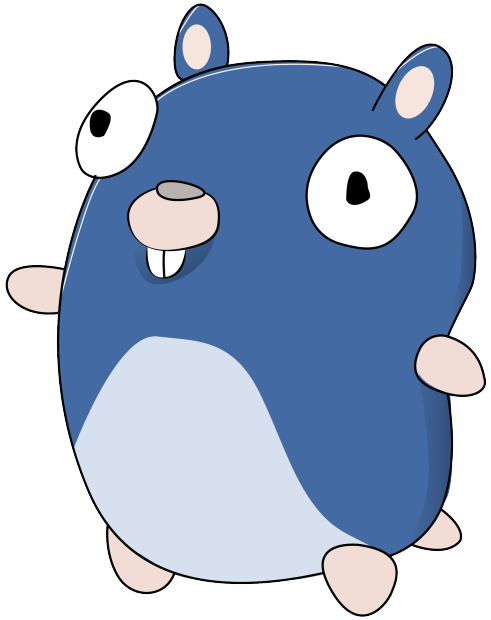
Fleet of Polyglot purpose-built apps



How can we package and distribute these apps?

Problem

Containers have become the de-facto standard packaging format to package and distribute apps



Get started with your Dockerfile

Source (Character art): https://phippy.io/

```
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#      https://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#

FROM ubuntu:bionic
     You, 2 days ago • cnb native image support
ENV LANG='en_US.UTF-8' LANGUAGE='en_US:en' LC_ALL='en_US.UTF-8'

RUN apt-get update \
    && apt-get install -y --no-install-recommends curl ca-certificates fontconfig loca
    && echo "en_US.UTF-8 UTF-8" >> /etc/locale.gen \
    && locale-gen en_US.UTF-8 \
    && rm -rf /var/lib/apt/lists/*
```

```
FROM humourmind/ubuntu:bionic

ENV JAVA_VERSION jdk-11.0.7+10

RUN set -eux; \
    ARCH="$(dpkg --print-architecture)"; \
    case "${ARCH}" in \
       aarch64|arm64) \
         ESUM='cfe504e9e9621b831a5cfd800a2005dafe90a1d11aa14ee35d7b674d68685698'; \
         BINARY_URL='https://github.com/AdoptOpenJDK/openjdk11-binaries/releases/download/jdk-11.0.
         ;; \      You, 2 days ago • cnb native image support
       armhf|armv7l) \
         ESUM='581bae8efcaa40e209a780baa6f96b7c8c9397965bc6d54533f4fd8599d5c742'; \
         BINARY_URL='https://github.com/AdoptOpenJDK/openjdk11-binaries/releases/download/jdk-11.0.
         ;; \
       ppc64el|ppc64le) \
         ESUM='364a030a4c5c10660d33988bd1d40e6a34f49ca6a2e04ff44b8cf809da75423a'; \
         BINARY_URL='https://github.com/AdoptOpenJDK/openjdk11-binaries/releases/download/jdk-11.0.
         ;; \
       s390x) \
         ESUM='8e14d11a84c67da457695cbc20a9d70628c741ffc09087d11b40a52f2e11dddc'; \
         BINARY_URL='https://github.com/AdoptOpenJDK/openjdk11-binaries/releases/download/jdk-11.0.
         ;; \
       amd64|x86_64) \
         ESUM='74b493dd8a884dcbee29682ead51b182d9d3e52b40c3d4cbb3167c2fd
         BINARY_URL='https://github.com/AdoptOpenJDK/openjdk11-binaries/
         ;; \
       *) \
         echo "Unsupported arch: ${ARCH}"; \
         exit 1; \
         ;; \
    esac; \
    curl -LfsSo /tmp/openjdk.tar.gz ${BINARY_URL}; \
    echo "${ESUM} *tmp/openjdk.tar.gz" | sha256sum -c -; \
    mkdir -p /opt/java/openjdk; \
    cd /opt/java/openjdk; \
    tar -xf /tmp/openjdk.tar.gz --strip-components=1; \
    rm -rf /tmp/openjdk.tar.gz;

ENV JAVA_HOME=/opt/java/openjdk \
```

```
FROM humourmind/adoptjre:11.0.7

LABEL MAINTAINER="Srinivasa Vasu"

ENV SPRING_OUTPUT_ANSI_ENABLED=ALWAYS

EXPOSE 8080

COPY ./build/libs/kloudnative-1.0.jar app.jar
     You, 2 days ago • cnb native image support
ENTRYPOINT ["java", "-jar", "/app.jar", "${JAVA_OPTS}", "-
```

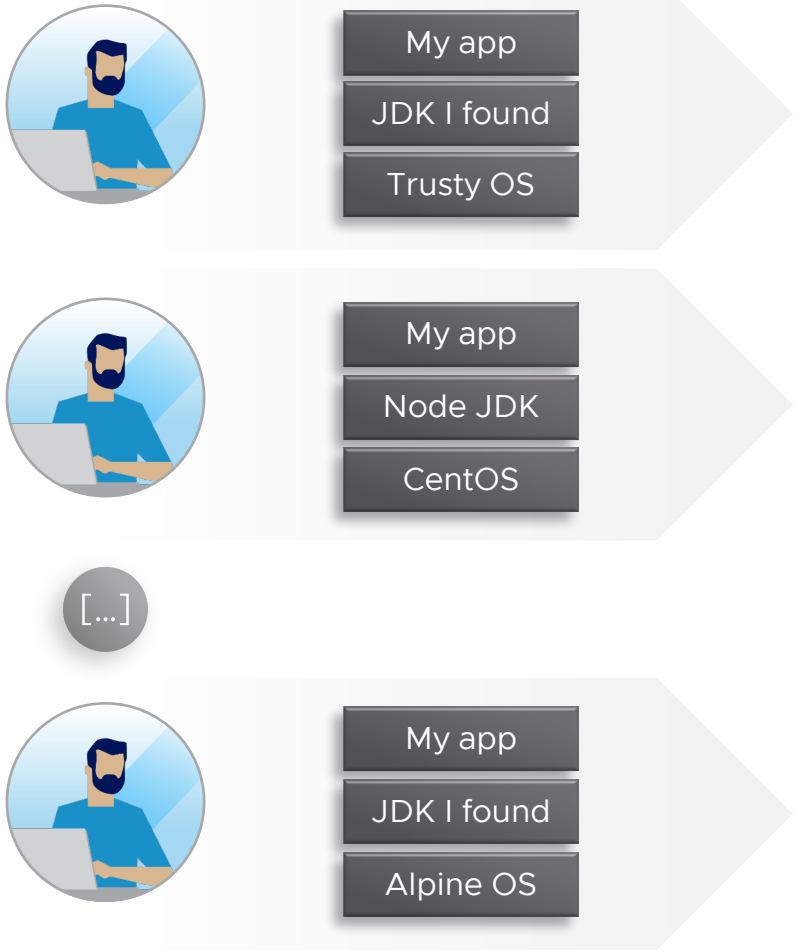
Yeah, Dockerfiles are great, but writing a production grade, secure, hardened images?

Problem

Bloated Containers Are Inefficient and Hard to Secure

Problem

Excess code and dependencies make container images larger files

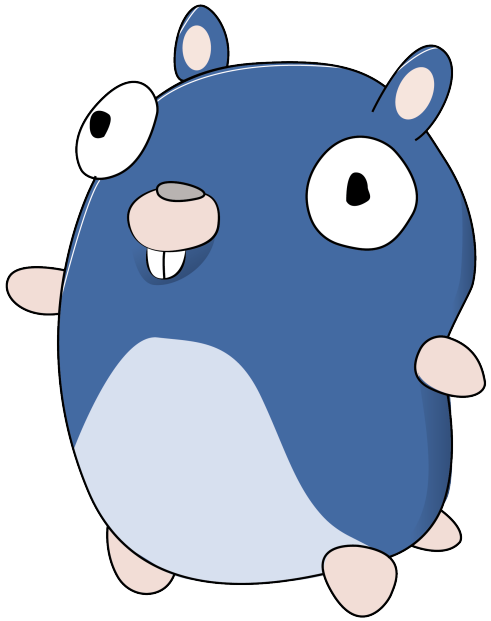


Find the CVE in production

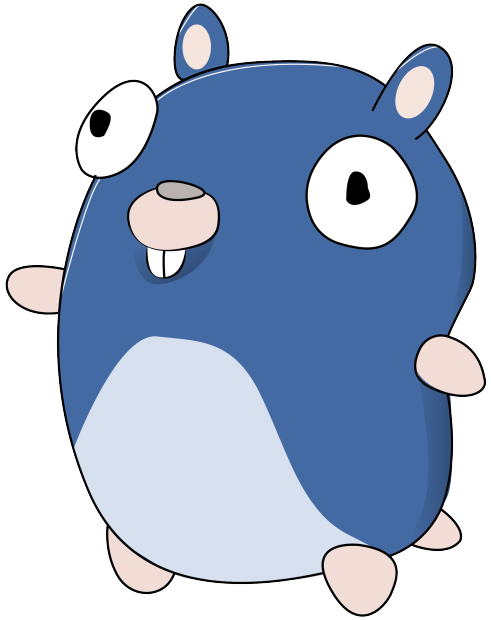
Aye, it's never easy. That's where cloud native buildpacks simplifies & eases the job of both dev & ops!

Let's get started with CNB

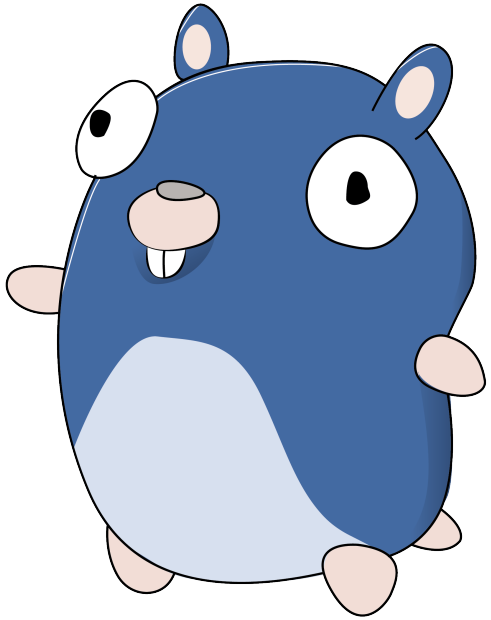
CNB is an open standard spec that defines how to transform your application source code into images that can run on any cloud



3 things to know about CNB spec

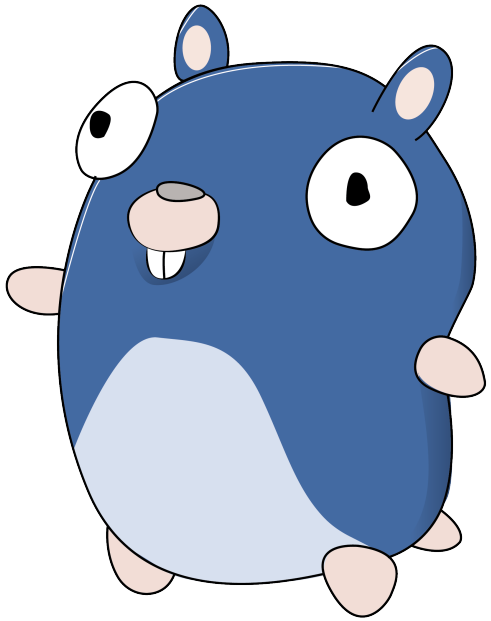


A **buildpack** is software that partially or completely transforms application source code into runnable artifacts



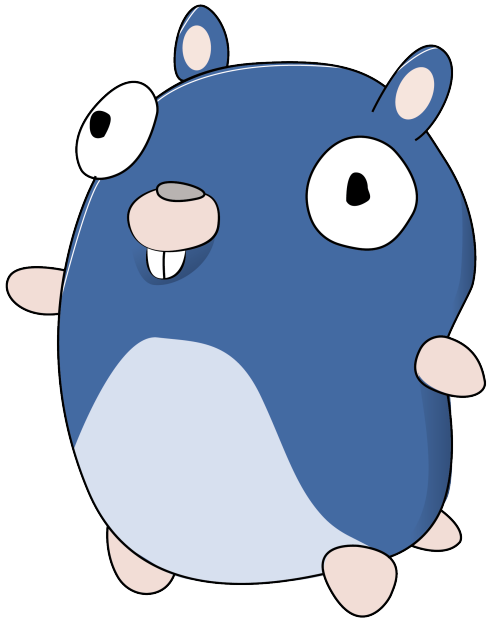
Source: https://buildpacks.io/

A lifecycle is software that orchestrates buildpacks and transforms the resulting artifacts into an OCI image



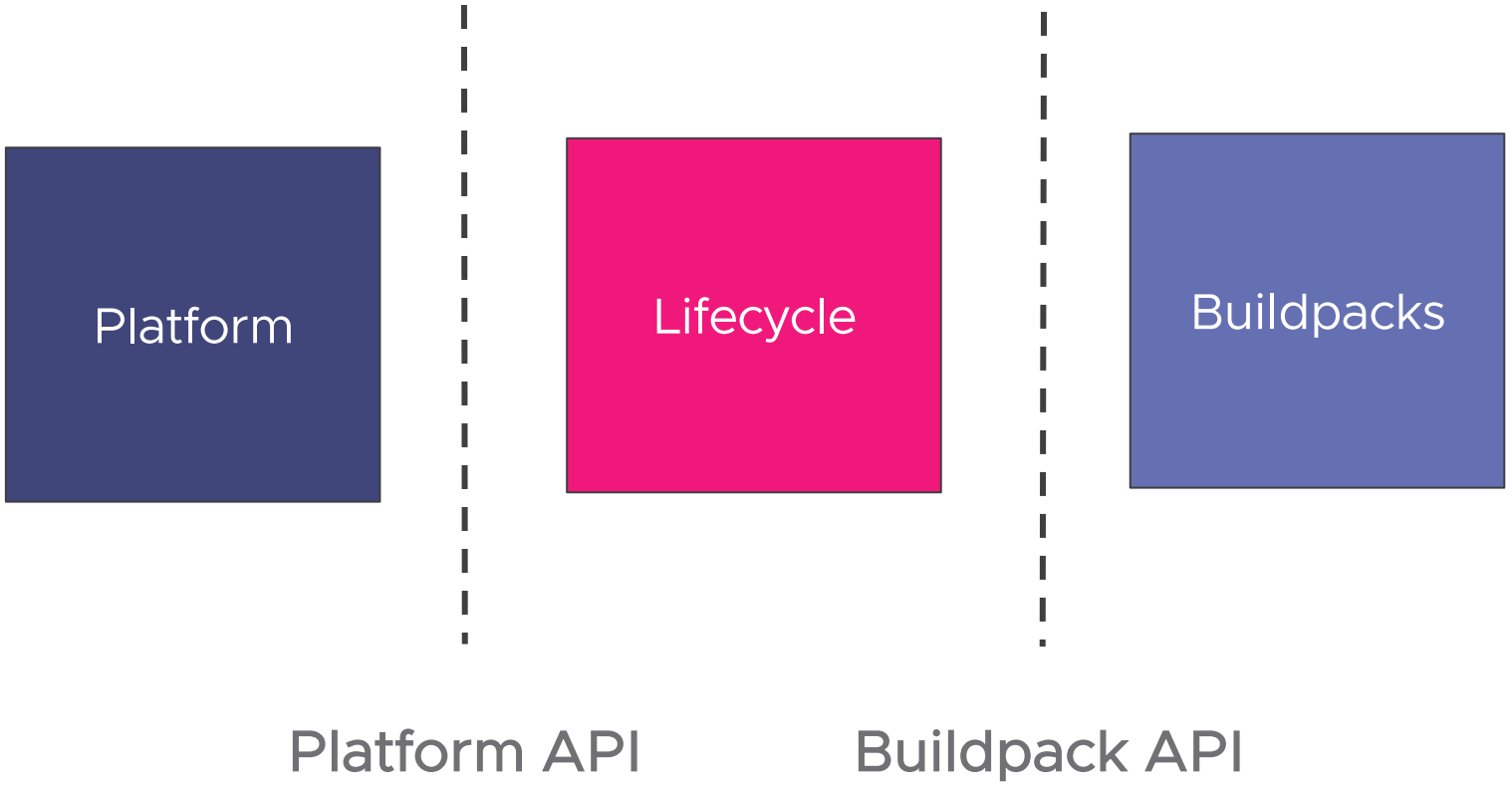
Source: https://buildpacks.io/

A platform is software that orchestrates a lifecycle to make buildpack functionality available to end-users such as application developers

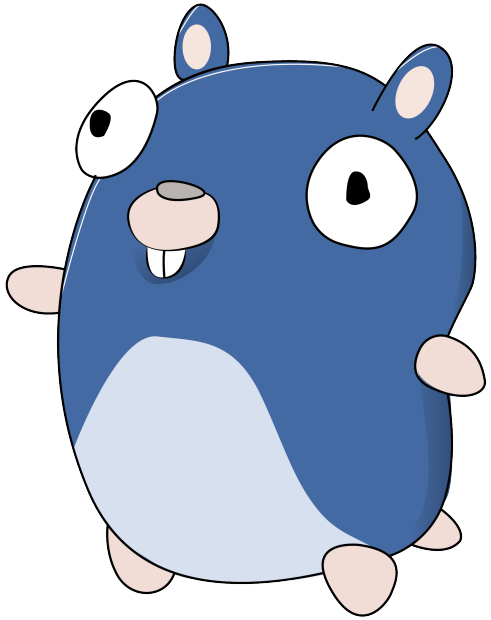


Source: https://buildpacks.io/

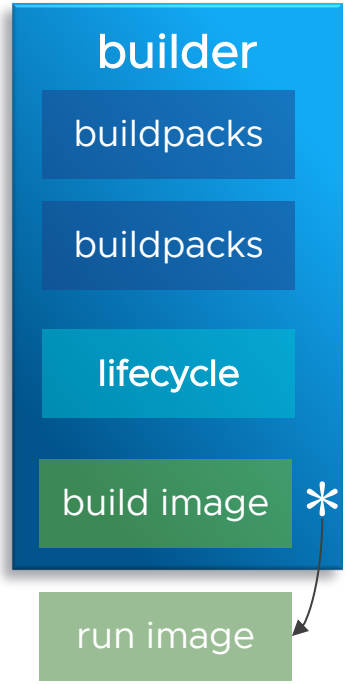
CNB Specification

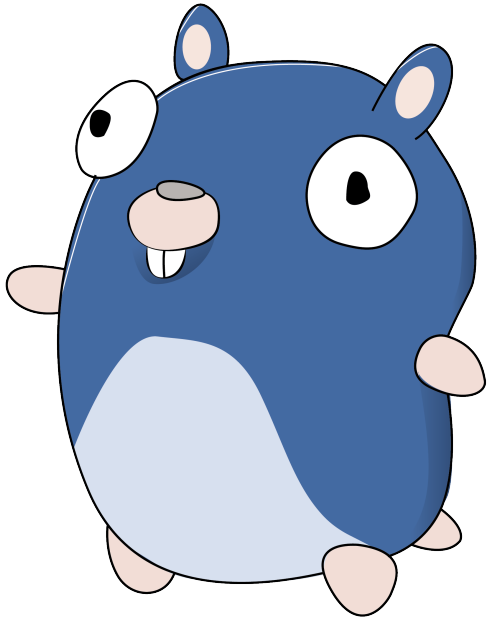


Two more things to know about before we get into live actions



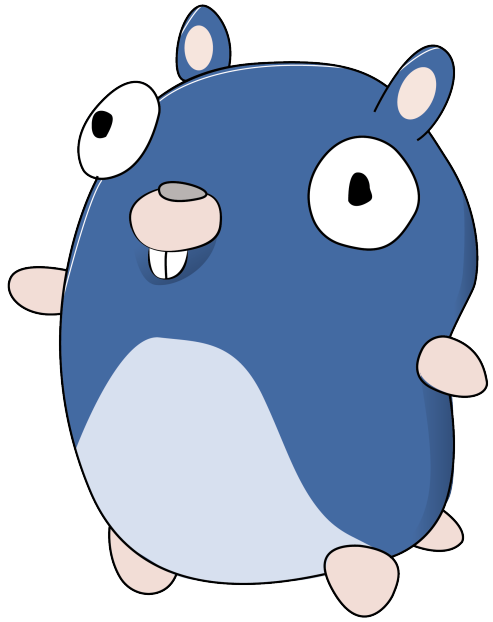
A builder is an image that bundles all the bits and information on how to build your apps

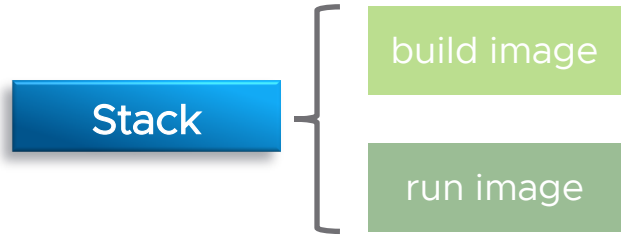




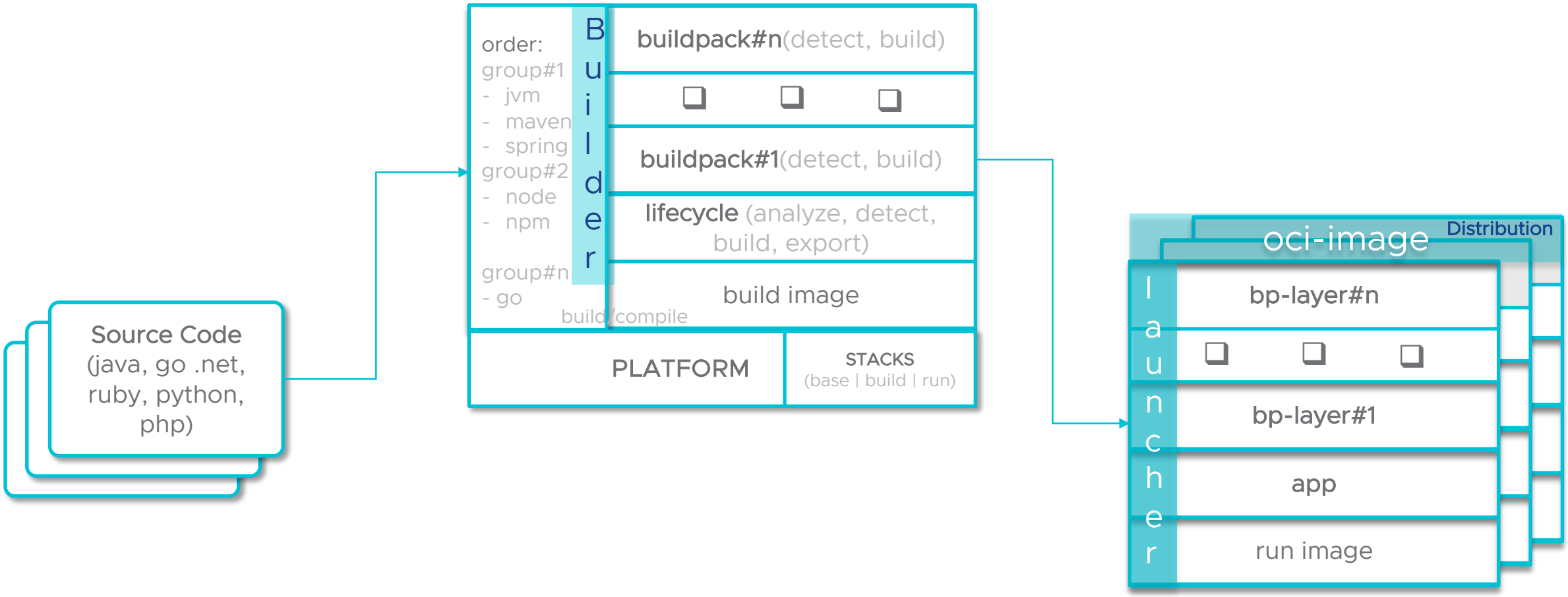
Source: https://buildpacks.io/

A stack provides the buildpack lifecycle with build-time and run-time environments in the form of images

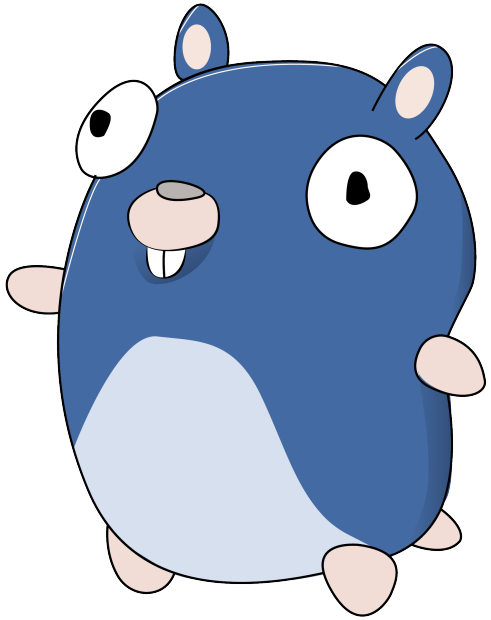




Workflow

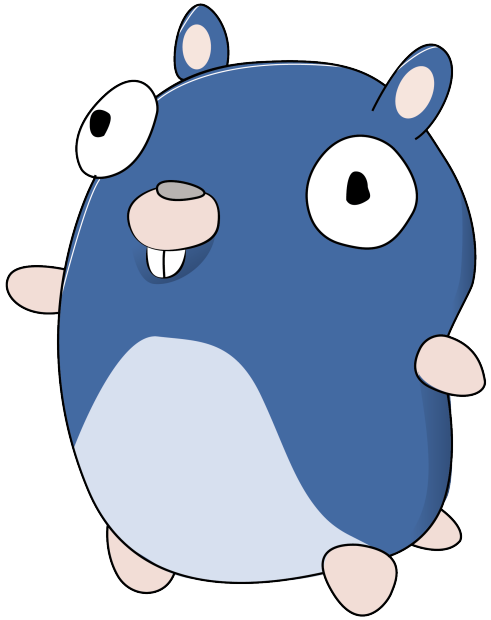

Source Code
(java, go .net, ruby, python, php)
order:
group#1
- jvm
- maven
- spring
group#2
- node
- npm
group#n
- go
Builder
buildpack#n(detect, build)
buildpack#1(detect, build)
lifecycle (analyze, detect, build, export)
build image
build/compile
PLATFORM
STACKS
(base | build | run)
oci-image
Distribution
launcher
bp-layer#n
bp-layer#1
app
run image

Ok, it's time to see it in action



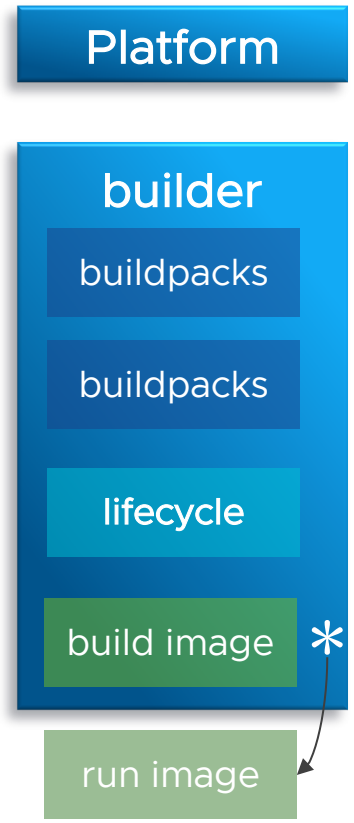
Wait, I remember seeing few other things like **kpack**, **paketo-buildpacks** etc. What are those?

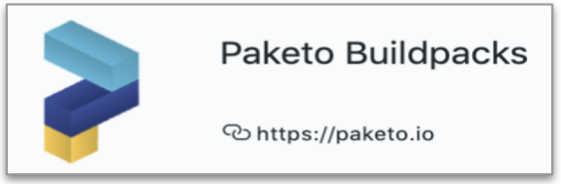
Aye, let's quickly look at the ecosystem

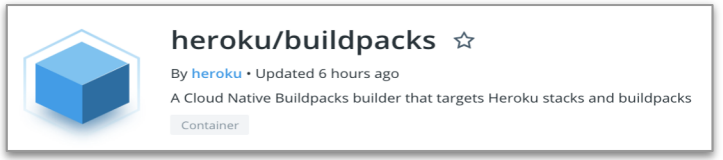


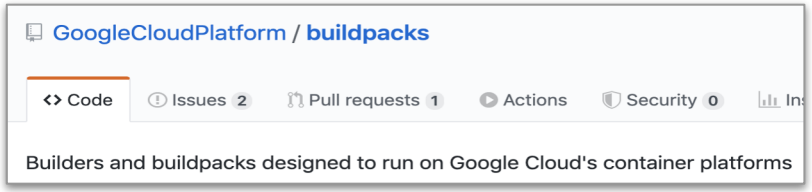
Platform and Buildpacks Eco system

pack
kpack
Spring Boot
Tekton
Tanzu Build Service
az acr pack
Google Cloud Run

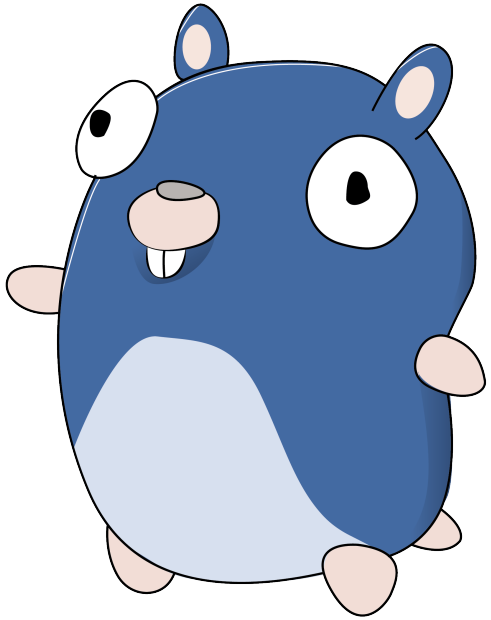






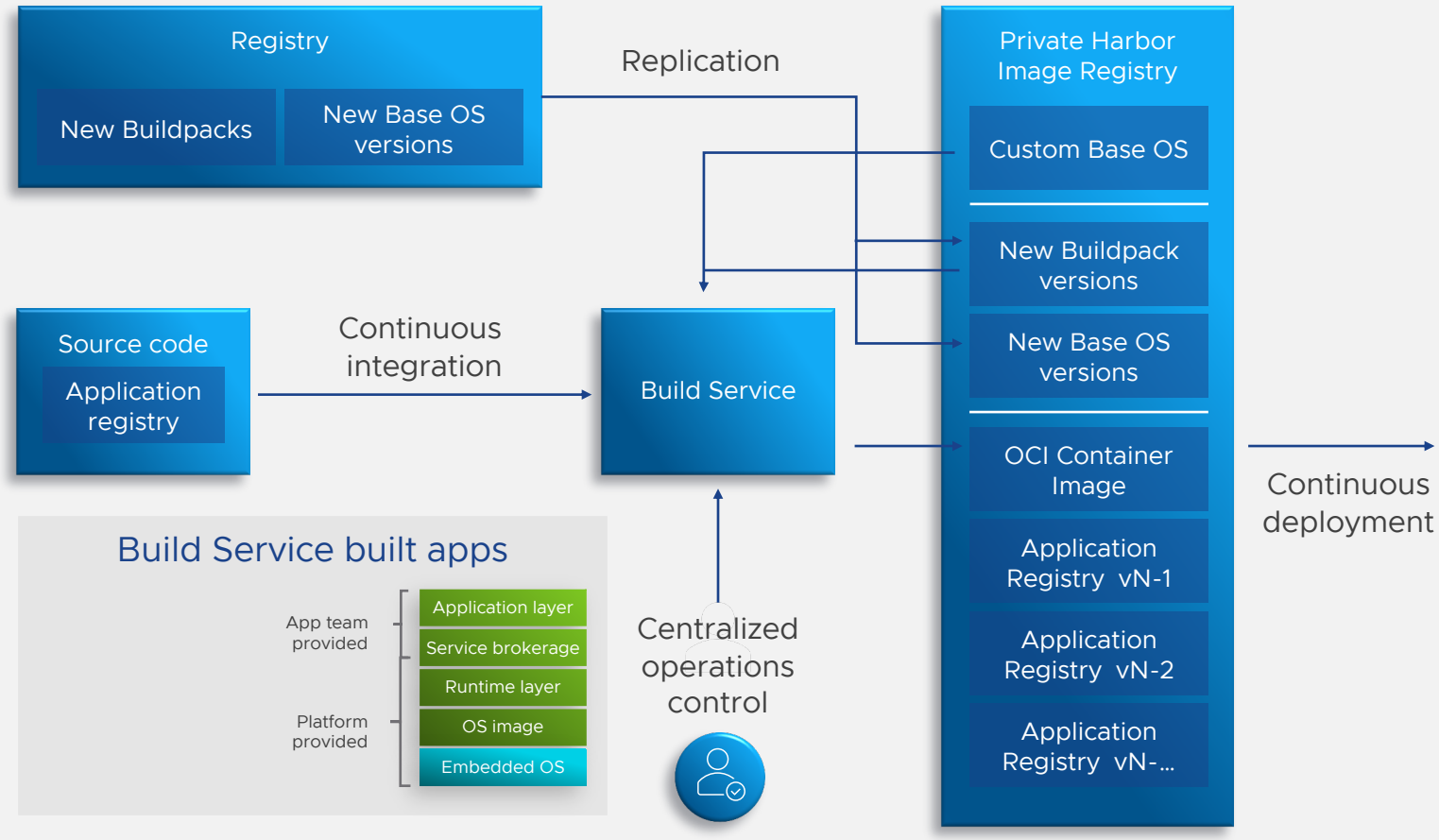


time to see it in action



Developing Cloud Native Applications

VMware Tanzu Build Service



Build Service

Continuously monitor image configurations for changes

Automatically rebuild containers based on image layer changes

Offer operations teams a centralized location for updating OS libraries and code dependencies

Creates OCI compliant container images within private registries ready to be deployed

Harbor

Automatically scan containers for vulnerabilities

Restrict containers based on vulnerability threat level

Thank You