Streamlining Kubernetes Application CI/CD with Bazel

Gregg Donovan - Staff Software Engineer, Etsy
Christopher Love - lionkube.com - chrislovecnm.com

March 2019

Bazel and Kubernetes at Etsy

Etsy

The global marketplace for unique and creative goods

About Etsy

- 39.4m active buyers
- 2.1m sellers
- 60m+ listings
- $3.9b 2018 GMS
- 874 employees
- Headquartered in Brooklyn, NY
- Offices in SF, Dublin

etsy.com/shop/SimplyCreatedForYou6

Etsy

unicorn paintings | Search

Sell on Etsy | Register | Sign in | Discover | Cart

Jewelry & Accessories | Clothing & Shoes | Home & Living | Wedding & Party | Toys & Entertainment | Art & Collectibles | Craft Supplies | Vintage | Gifts

All categories › "unicorn paintings" (6,357 Results)

Sort by: Relevancy

Special offers
- On sale

All categories
- Art & Collectibles
- Craft Supplies & Tools
- Bath & Beauty
- Home & Living

+ Show more

Shipping
- Free shipping
- Ready to ship in 1 business day
- Ready to ship within 3 business days

Subject
- Abstract & geometric
- Animal
- Anime & cartoon
- Architecture & cityscape
- Beach & tropical

+ Show more

Orientation
- Horizontal

Ad

Unicorn Waterc

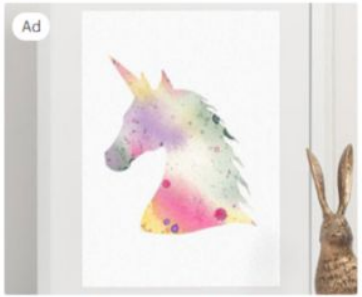
AudreyZombiesAr

★★★★★ (1)

$20.00

Ad

Ad



ıt, Cut...

Ad

Unicorn Watercolor Print Nursery ...

SuziBlueDesigns

★★★★★ (21)

$17.00

It's a fun problem

More colors

Unicorn canvas

Framed Magical Rainbow Haired U...

LoveBumble

Framed Magical Rainbow Haired U...

LoveBumble

Framed Magical Rainbow Haired, ...

LoveBumble

Bazel: A Modern Build *and test* System

Bazel.build

Fast, reproducible build and test

Hermetic and Deterministic Builds

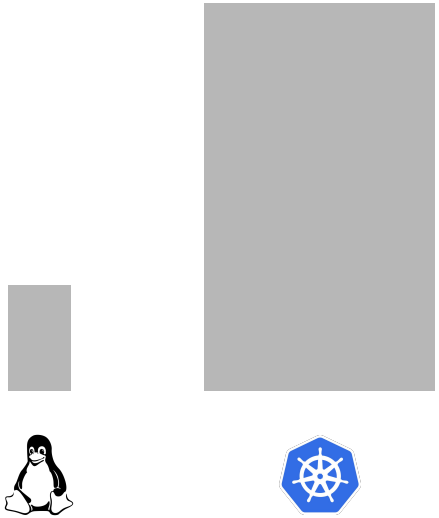
Cloud accelerated

Google OSS

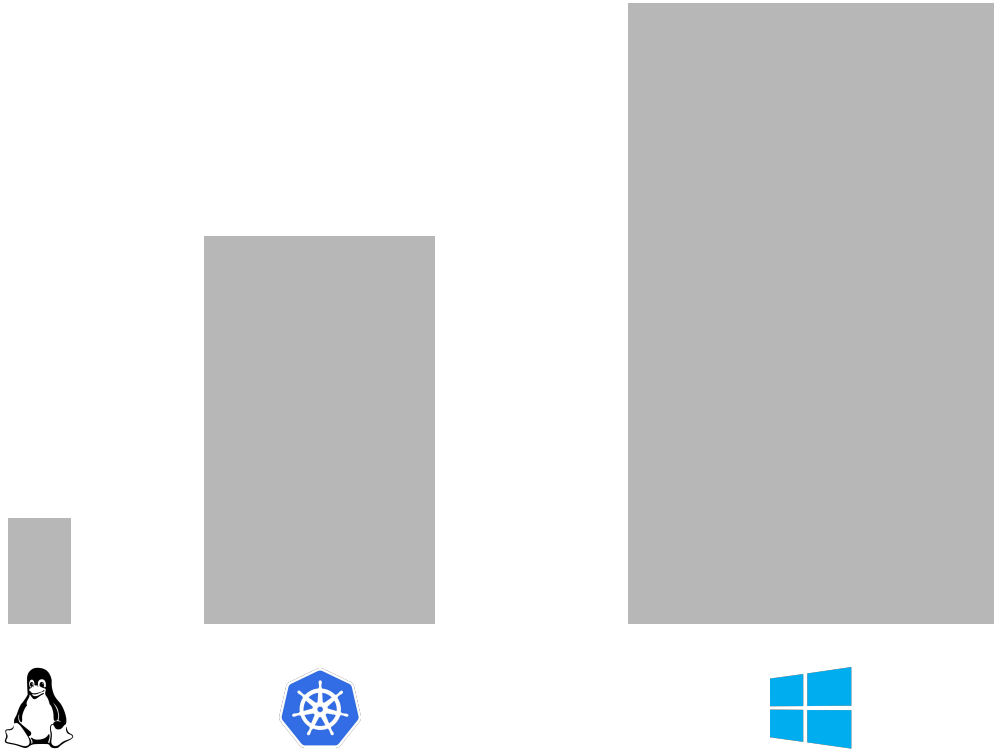
BIG CODE

LoC

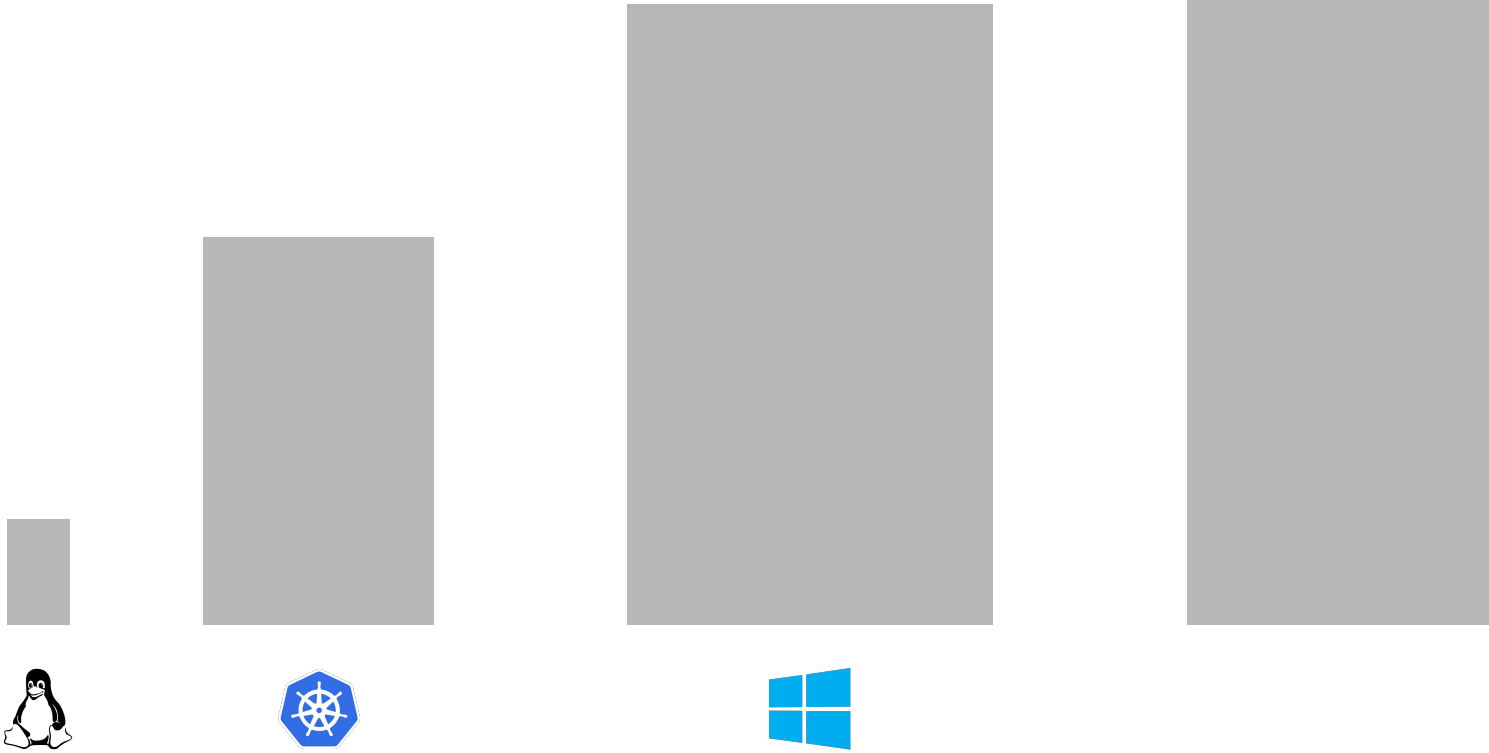
LoC



LoC

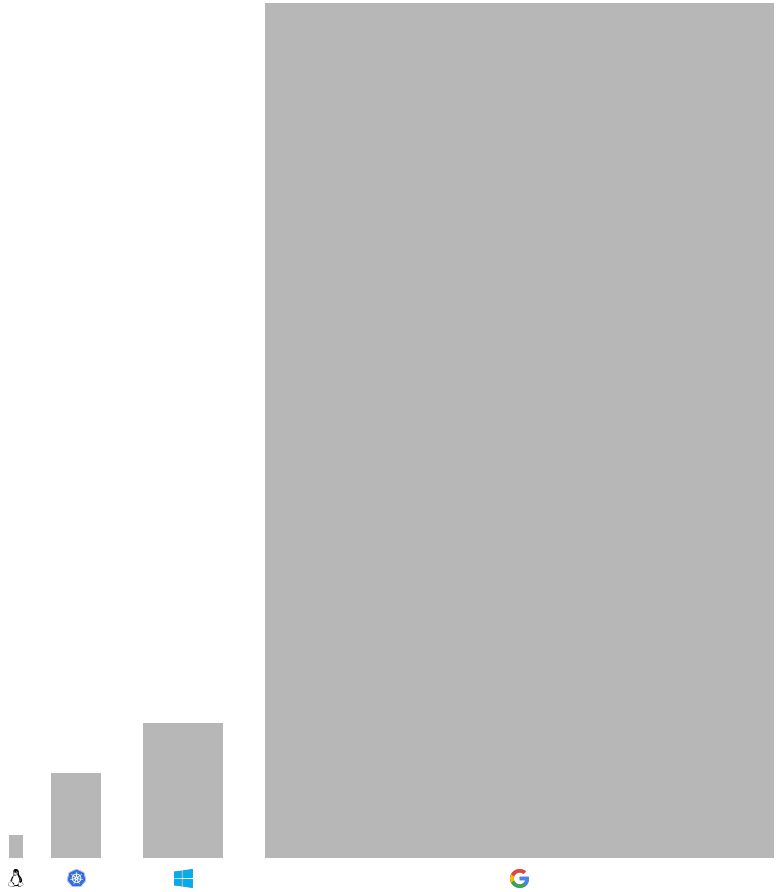


LoC



LoC

LoC



2,000,000,000

BIG CODE →

BIG BUILD →

BIG TEST

I don’t have that much code

Deterministic

In mathematics, computer science and physics, a **deterministic** system is a system in which no randomness is involved in the development of future states of the system. A **deterministic** model will thus always produce the same output from a given starting condition or initial state.

en.wikipedia.org/wiki/Deterministic_system

Hermetic

A **hermetic seal** is any type of sealing that makes a given object **airtight** (excludes the passage of air, oxygen, or other gases). The term originally applied to airtight glass containers, but as technology advanced it applied to a larger category of materials, including rubber and plastics.

en.wikipedia.org/wiki/Hermetic_seal

D.R.Y. Tests

Only retest when necessary

Fan Out

Execute build and tests in parallel

Remote Builds

RBE and other remote build farms

Bazel builds ~all the things

Android
C and C++
C#
D
Docker
Go
Groovy
Haskell
Kotlin
iOS
Java
JavaScript
Jsonnet
Objective C
Perl
PHP
Protobuf
Python
Ruby
Rust
Sass
Scala
Shell
Swift
TypeScript

WORKSPACE

A *workspace* is a directory on your filesystem that contains the source files for the software you want to build, as well as symbolic links to directories that contain the build outputs. Each workspace directory has a text file named `WORKSPACE` which may be empty, or may contain references to external dependencies required to build the outputs. See also the Workspace Rules section in the Build Encyclopedia.

Bazel.io

docs.bazel.build/versions/master/build-ref.html#workspace

WORKSPACE Rules

```
workspace(name = "gke_bazel_example")

# "http_archive" is a Bazel rule that loads Bazel repositories &
# makes its targets available for execution.
# It is deprecated however, so we need to manually load it to use it.
# See
https://docs.bazel.build/versions/master/be/workspace.html#http_archive
load("@bazel_tools//tools/build_defs/repo:http.bzl", "http_archive")

# Load go rules to build kubectl
http_archive(
  name = "io_bazel_rules_go",
  urls =
["https://github.com/bazelbuild/rules_go/releases/download/0.18.5/rules_go-0
.18.5.tar.gz"],
  sha256 =
"a82a352bffae6bee4e95f68a8d80a70e87f42c4741e6a448bec11998fcc82329",
)
```

BUILD File

```
# The package rule declares this directory as a Bazel package
# which scopes targets defined in this Build file under this package.
# The visibility parameter declares which packages can call targets
# in this package. In this case, we're saying anyone can call these targets.
# See https://docs.bazel.build/versions/master/be/functions.html#package
package(default_visibility = ["//visibility:public"])

load("@io_bazel_rules_k8s//k8s:objects.bzl", "k8s_objects")


# ts_library and ng_module use the `//:tsconfig.json` target
# by default. This alias allows omitting explicit tsconfig
# attribute, and uses //js-client/src:tsconfig.json whenever
# //:tsconfig.json is called
# See https://docs.bazel.build/versions/master/be/general.html#alias
alias(
    name = "tsconfig.json",
    actual = "//js-client/src:tsconfig.json",
)

k8s_objects(
   name = "bazel_demo_k8s",
   objects = [
      "//java-spring-boot:k8s",
      "//js-client:k8s",
   ]
)
```

Search Monorepo

20+ services

One CI/CD pipeline

Bazel

rules_k8s (Bazel)

rules_docker (Bazel)

Python for YAML

Per k8s context config

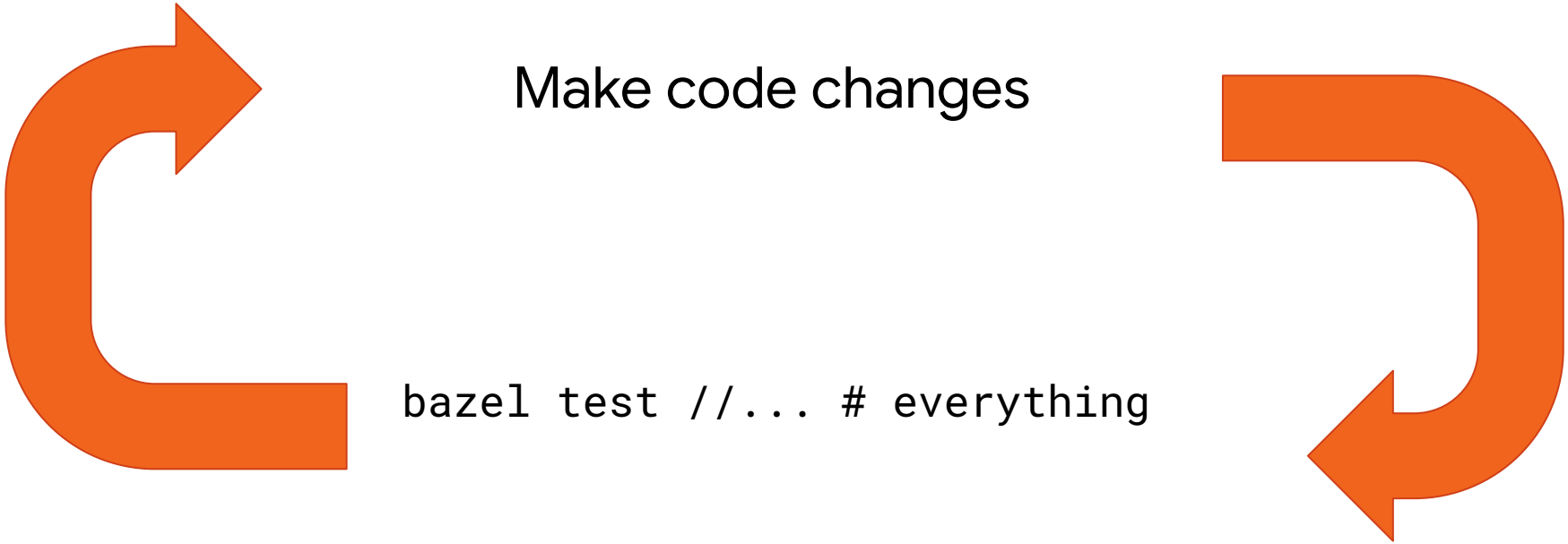
etsy.com/shop/RossiVArt

- Why and how Etsy adopted Bazel, **rules_k8s,and rules_docker**

- How they work to yield fast, correct deployments

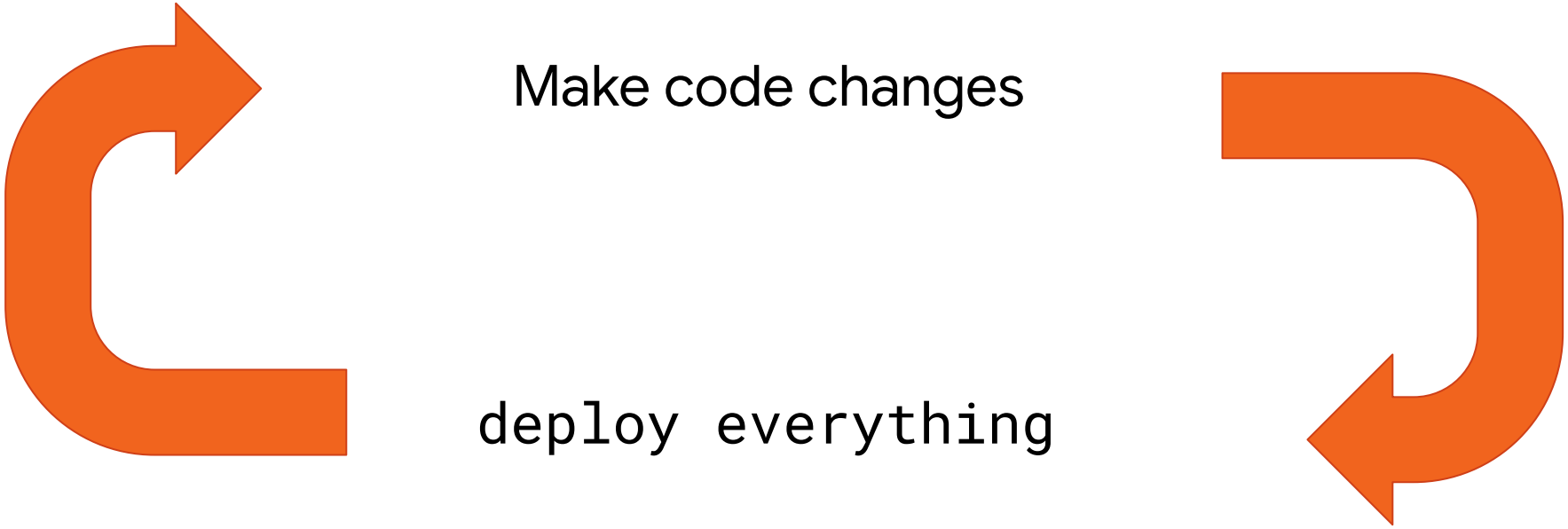
- Bazel and Kubernetes learnings from our GKE migration

Bazel development cycle



Let Bazel figure out what actually needs to be retested.

Idealized Bazel deploy cycle

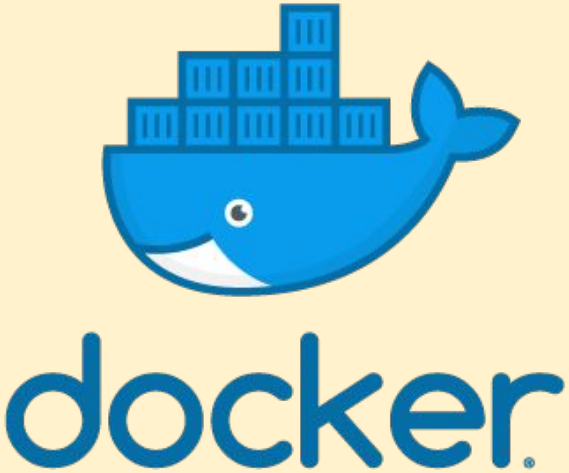


Let Bazel and K8s figure out what actually needs redeploying.

Reproducibility and Determinism are Challenging

- Timestamps
- Timestamps in archive metadata (JAR, TAR)
- Timestamps in Docker images
- RUN commands in Dockerfiles
- Unstable inputs
- Unstable outputs
- etc.

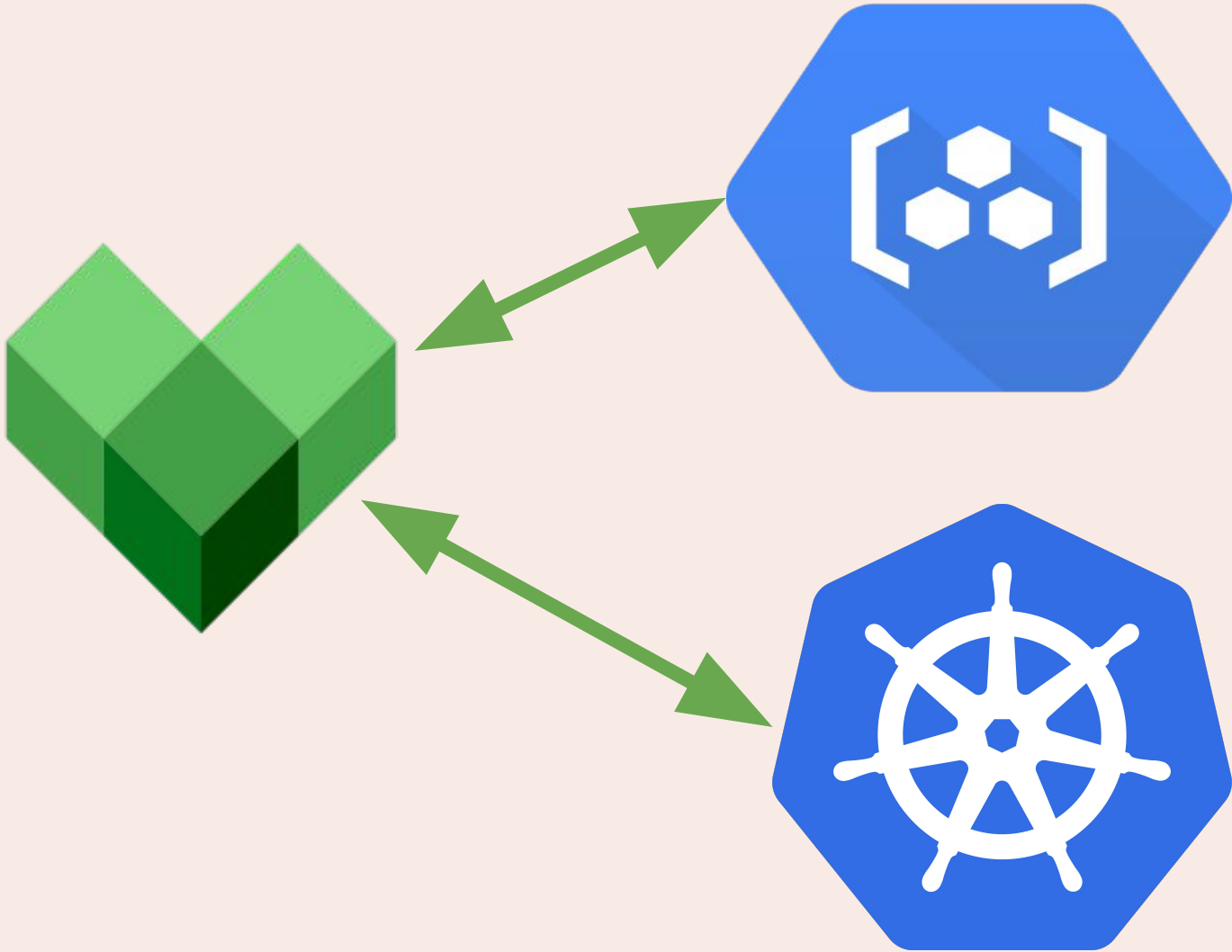
A set of rules for pulling down base images, augmenting them with build artifacts and assets



1. Authentication
2. Publish Containers
3. Mananage Container Digests
4. Manifest Templating
5. Deploying Manifests
6. Full Application CRUD

Deploy just the right amount, every time

Let Bazel work it out with the Container Registry and K8s



Kubernetes: Hashing & Caching

SHA256

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: redis-master
  labels:
    app: redis
spec:
  selector:
    matchLabels:
      app: redis
      role: master
      tier: backend
  replicas: 1
  template:
    metadata:
      labels:
        app: redis
        role: master
        tier: backend
    spec:
      containers:
      - name: master
        image: k8s.gcr.io/redis:e2e
        resources:
          requests:
            cpu: 100m
            memory: 100Mi
        ports:
        - containerPort: 6379
```

rules_docker > Dockerfile

```
$ docker images
REPOSITORY                          TAG                       IMAGE ID       CREATED          SIZE
search/apps/mmx                     mmx_docker                e2a1d55be23d   48 years ago     932 MB
search_data_docker                  intermediate              cbefdae46002   48 years ago     460.4 MB
search/apps/spell_correction        spell_correction_docker   91653e8b5207   48 years ago     448.8 MB
search/apps/etsy-search1            etsy-search1_docker       167736f9b424   48 years ago     569.1 MB
search/apps/slv2                    slv2_docker               3aa5a41625c5   48 years ago     935.3 MB
search/apps/elastic2/kubernetes     elastic2_gke_docker       eb56b8285cad   48 years ago     125.9 MB
...
```

rules_k8s

```
load("@io_bazel_rules_k8s//k8s:object.bzl", "k8s_object")

k8s_object(
  name = "dev",
  kind = "deployment",

  # A template of a Kubernetes Deployment object yaml.
  template = ":deployment.yaml",

  # An optional collection of docker_build images to publish
when this target is bazel run. The digest of the published image
is substituted as a part of the resolution process.
  images = {
    "gcr.io/rules_k8s/server:dev": //server:image"
  },
)
```

What is a container?

```
$ docker inspect bb1efd443479
[
    {
        "Id": "sha256:bb1efd443479d95d959c990f268a6bb3d06bfafb82ce2200c45d0a24262e0c1d",
        "RepoTags": [ "bazel/grafana:grafana_docker" ],
        "Created": "1970-01-01T00:00:00Z",
        "Author": "Bazel",
        "Config": {
            "User": "grafana",
            "ExposedPorts": { "3000/tcp": {} },
            "Env": [

"PATH=/usr/share/grafana/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
                "GF_PATHS_CONFIG=/etc/grafana/grafana.ini",
            ],
            "Image": "sha256:ea9f0ca0dc5d538ab046a8618af1aaf0d3df05e89dc3a0420fabd9b46c4a0261",
            "WorkingDir": "/",
            "Entrypoint": ["/run.sh"],
        },
        "Architecture": "amd64",
        "Os": "linux",
        "Size": 238231783,
        "RootFS": {
            "Type": "layers",
            "Layers": [
                "sha256:d626a8ad97a1f9c1f2c4db3814751ada64f60aed927764a3f994fcd88363b659",
                "sha256:fe145ea19a267f67c106d3bf3df09a14d0d02c0f93e2c14df2f32f28562b954c",
                "sha256:d580759d14dac7f636711d0901258b1b22ae4c1bb046e06d1801c031192e52b5",
                "sha256:7d59735eaa9f4b2c5da8dc576540d1903a9db46fcbf867453cf95b6466f2ceab",
                "sha256:fd0c81ee3761fc31e63a56793e9baaa3744f1bc26077f63480bde878cc819b53",
                "sha256:f874fe8e2453b568a50fc6072edc1dd75c6ab568dbd658fe9978588411abad20",
                "sha256:9dd3209f58e05896460aac252bb068e1a59d107eabf7ffb7faf25f2cebae70cd"
            ]
        }
    }
]
```

`rules_docker`: Docker without docker or a Dockerfile

Container Registry v2 API

`HEAD /v2/<image-name>/manifests/<sha256>`

Check for the existence of an image manifest.

`HEAD /v2/<name>/blobs/<digest>`

Check for the existence of a layer.

Kubernetes pod-template-hash

SHA256

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: grafana
  labels:
    app: grafana
spec:
  selector:
    matchLabels:
      app: grafana
  replicas: 1
  template:
    metadata:
      labels:
        app: grafana
    spec:
      containers:
      - name: grafana
        image:
gcr.io/etsy-gcr/grafana@sha256:99b8c7ac7fdb1e04ccbd5609
0f91f3eeb0ed21a77abb5bb2a25532fca7026dbb
        resources:
          requests:
            cpu: 100m
            memory: 100Mi
        ports:
        - containerPort: 3000
```

ther ules_k8s eatures

- Yaml templating and variable resolving
- Multiple Objects definitions
- Kubectl versioning and compiling from source for specific builds
- Different cluster definitions without changing the build

ther ules_docker eatures

- Fine grained reusable container layer definitions
- Various distroless language based runtimes. (Java, Go, Python, Rust, etc.)
- Load and run container locally

Tip #1:
Use "Distroless" Containers

- Smaller size
- No package manager
- Fewer CVEs

github.com/GoogleContainerTools/distroless

```
load("@io_bazel_rules_docker//java:image.bzl",
"java_image")


java_image(
    name = "hello",
    srcs = ["HelloJava.java"],
    base = "//java:java8",
    main_class = "examples.HelloJava",
)
```

Tip #2:
Use SHA256 image references

Tip #3:
Build complex YAML with the K8s Client APIs

ip #4:
ilt for dev
workflow

lt.dev

```
def bazel_build(image, target):
  custom_build(
    image,
    'bazel run ' + target,
    [],
    tag="image",
  )

k8s_yaml(bazel_k8s(":snack-server"))
bazel_build('bazel/snack', '//snack:image')
```

ip #5:
se CRDs to
odel cloud
esources

```
apiVersion: redis.cnrm.cloud.google.com/v1alpha2
kind: RedisInstance
metadata:
 name: redisinstance-sample
spec:
 displayName: Sample Redis Instance
 region: us-central1
 tier: BASIC
 memorySizeGb: 16
---------
apiVersion: service-operator.aws/v1alpha1
kind: ElastiCache
spec:
 cacheSubnetGroupName: "loadtest-cluster-k8s"
 vpcSecurityGroupIds: "sg-0581b94aa3c0db58c, sg-02b6d0034e8c2fa1b"
 autoMinorVersionUpgrade: true
 engine: redis
 cacheNodeType: "cache.m4.large"
```

See:

github.com/GoogleCloudPlatform/k8s-config-connector

Demo Architecture

http://bit.ly/k8s-bazel

Frameworks
- Springboot rest services
- Angular front end

Multiple Languages
- Java
- Typescript
- Javascript

Unit Testing
- JUnit
- Jasmine
- Protractor
- Run Headless Chrome testing

DEMO

Contact Us

@chrislovecnm
lionkube.com

@greggdonovan
gregg@etsy.com

etsy.com/shop/IrinaRedineArt

Project Helmsman

Workshops and matching open-source PoCs to guide customers and partners through using Kubernetes Engine in production

Helmsman is a project to build and release open-source examples of how to run common patterns in Google Kubernetes Engine along with workshops for Google's partners to deliver, to teach their customers how to move to a containerized world.

Shortlink to the code: **https://goo.gl/uD5sAM**

Thank you!

etsy.com/shop/CharlenePrecious