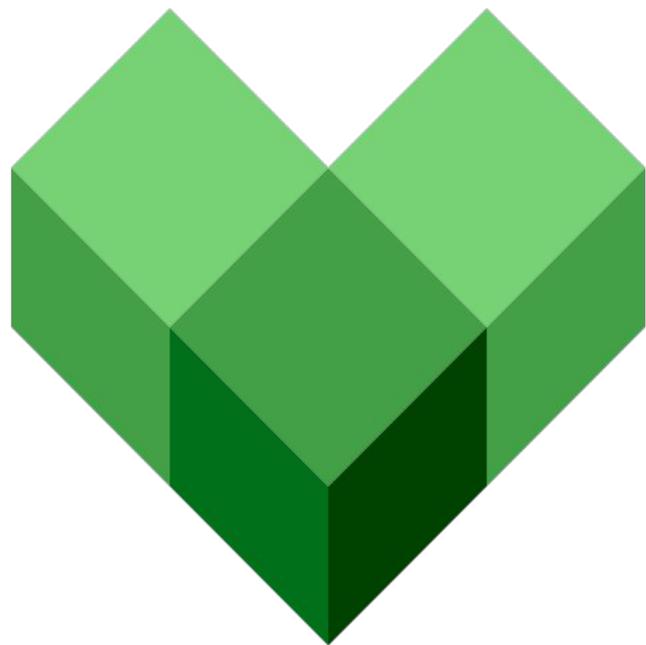


# Kube + Bazel + Helm

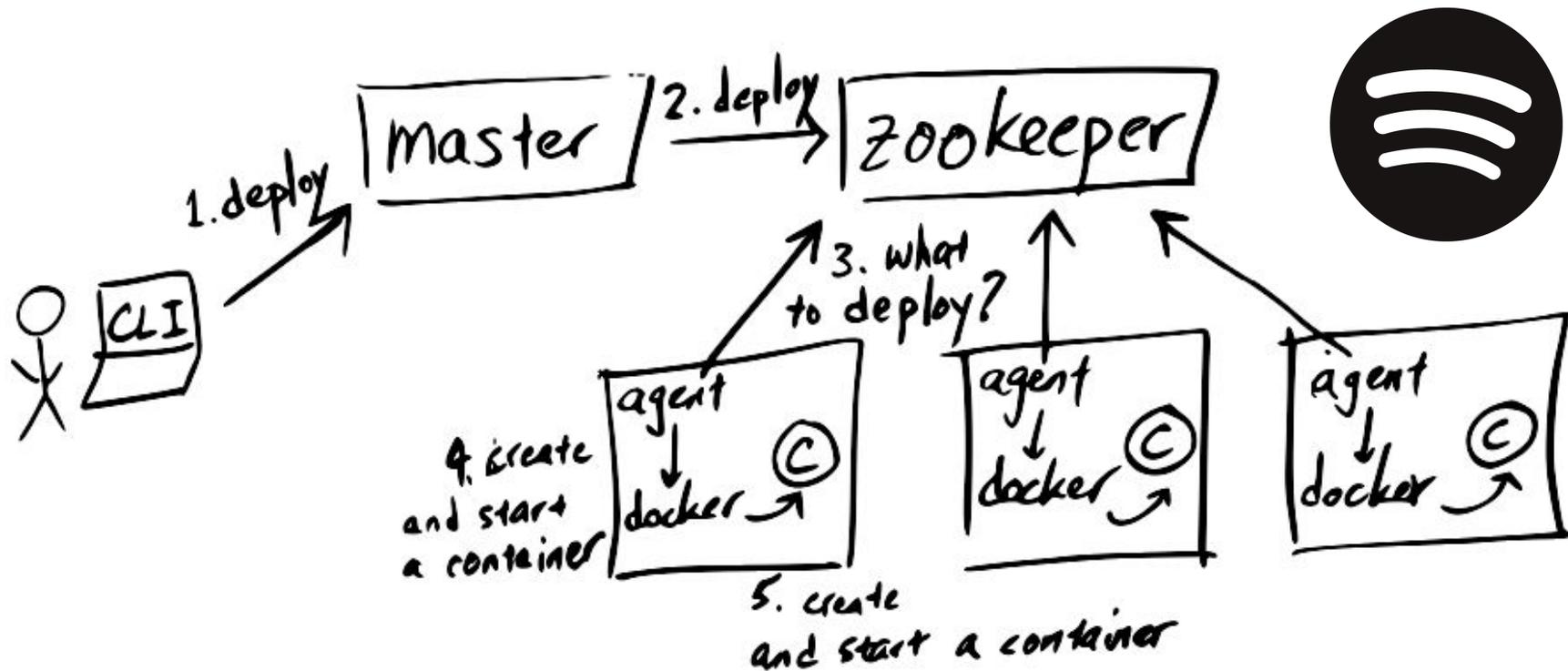
“I’ll take Cloud Buzzwords for \$300, Alex”

Rohan Singh / Infra Engineer

*Hustle*

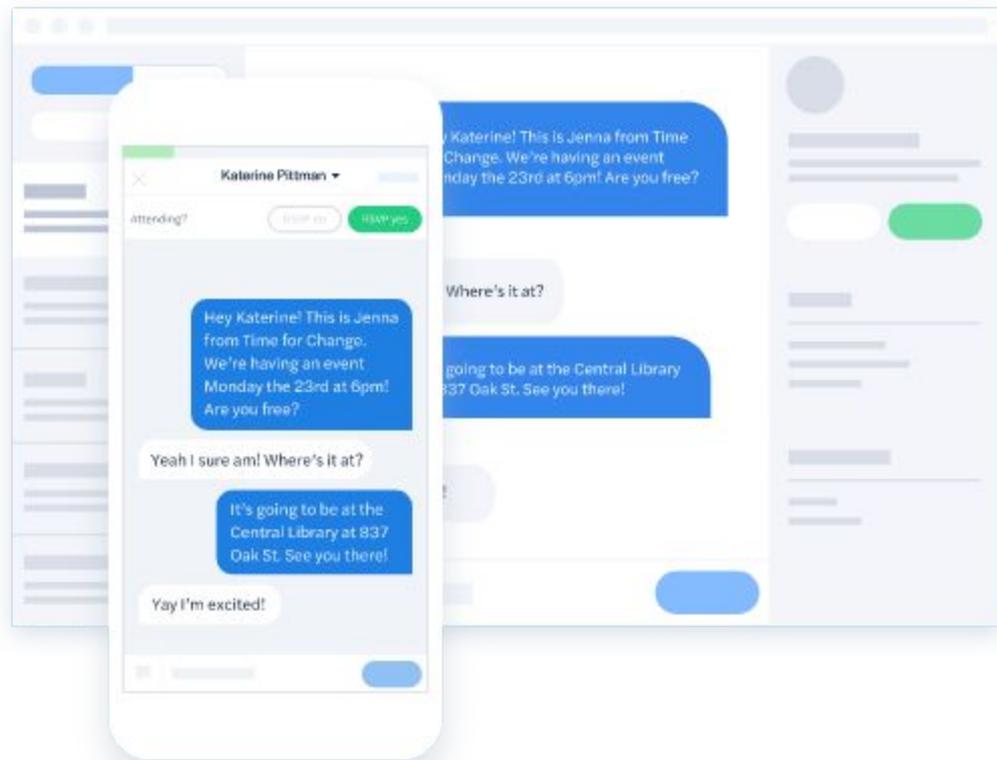


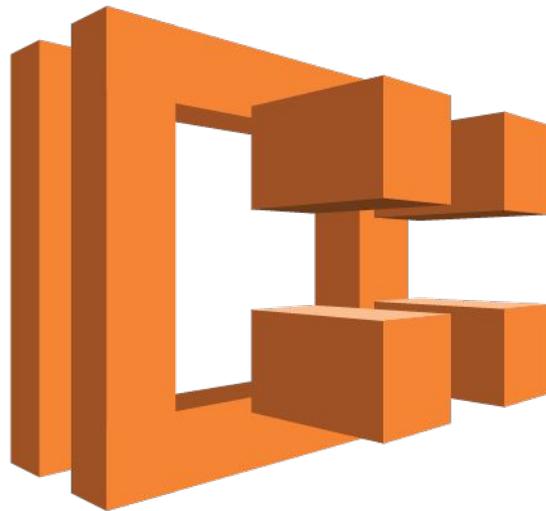
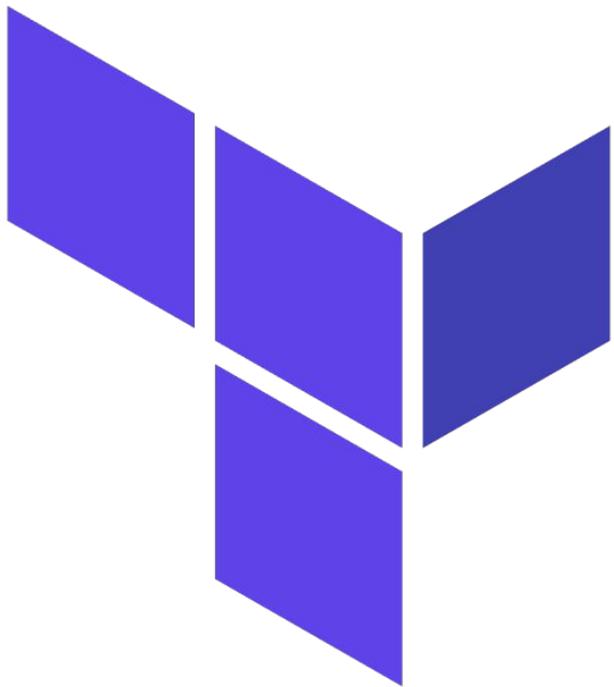
*Hustle*





*Hustle*





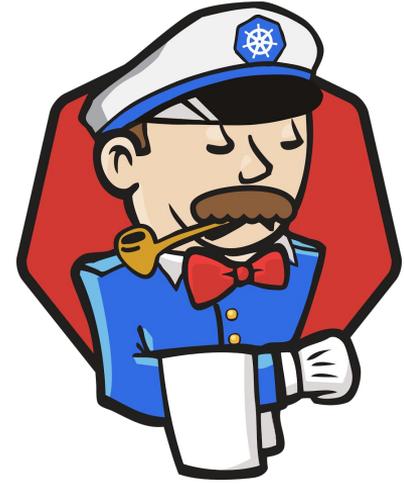
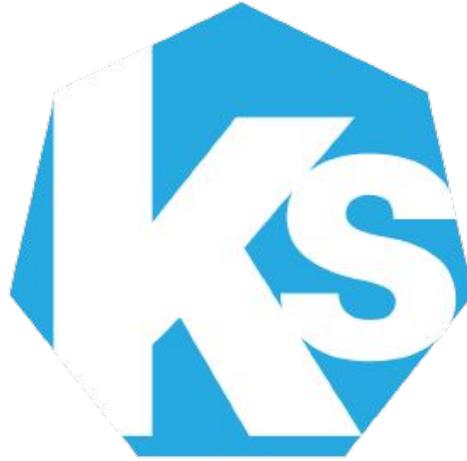
**AWS ECS**

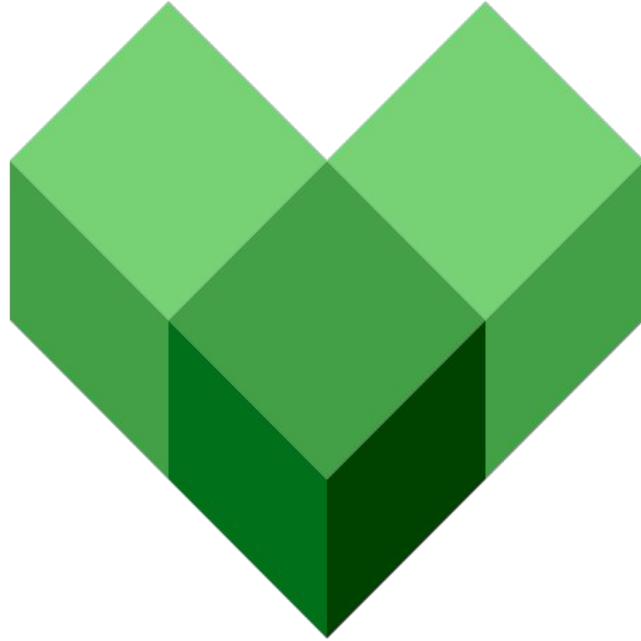


*Hustle*

**YA  
ML**

*Hustle*





*Hustle*

**DETERMINISM**

```
# BUILD file
```

```
container_image(  
    name = "image",  
    files = ["app.js"],  
)
```

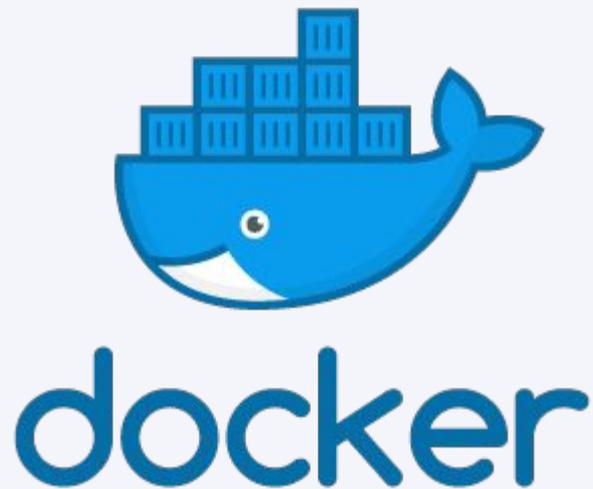


docker

```
# BUILD file

container_image(
    name = "image",
    files = [":app.jar"],
)

java_binary(
    name = "app",
    srcs = ["Foo.java"],
)
```



```
# BUILD file

container_image(
  name = "image",
  files = ["app.js"],
)

k8s_object(
  name = "k8s",
  images = {
    "hustle/app": ":image",
  },
  template = "app.yaml",
)
```



```
def k8s_set(name):  
    for c in ["dev", "prod"]:  
        k8s_object(  
            name = name + "-" + c,  
            template = c + ".yaml",  
        )
```

## Macros

Handy, simple

# Rules

*‘A rule defines a series of actions that Bazel should perform on inputs to get a set of outputs.’*

container\_image

java\_library

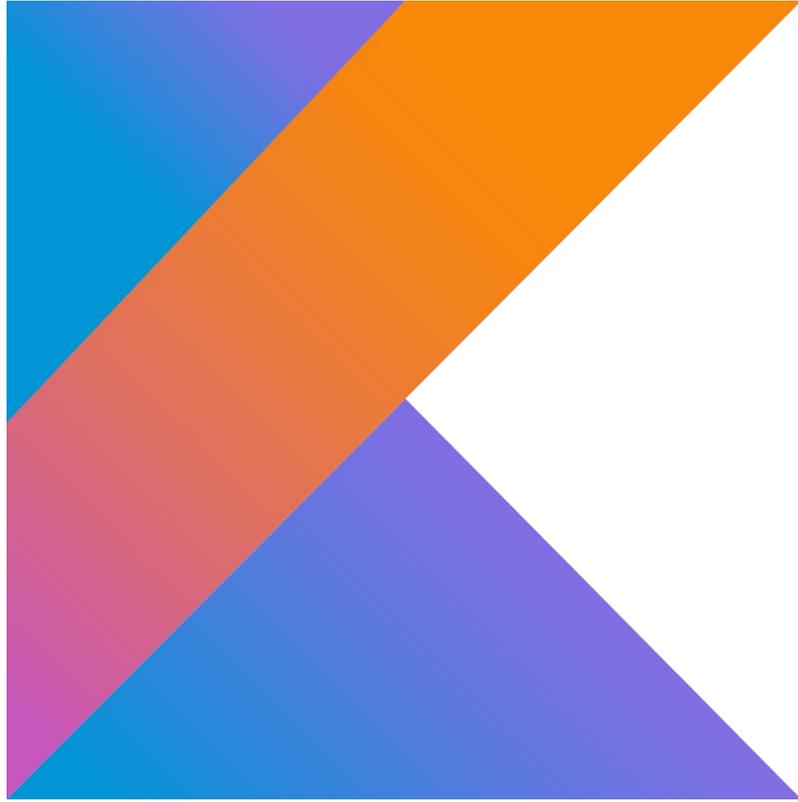
nodejs\_binary

sh\_test

# What do the docs say?

*‘Rules are for advanced users: we expect that most people will never have to write one, they will only load and call existing rules.’*

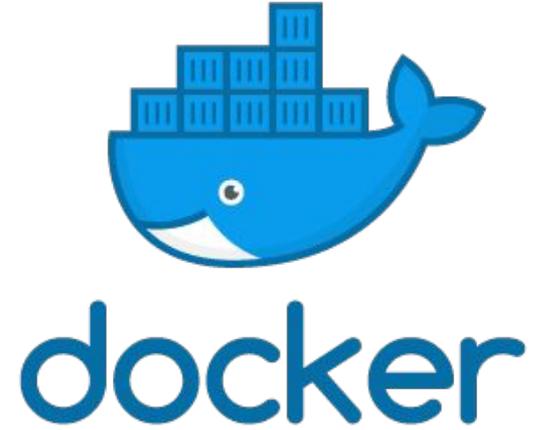




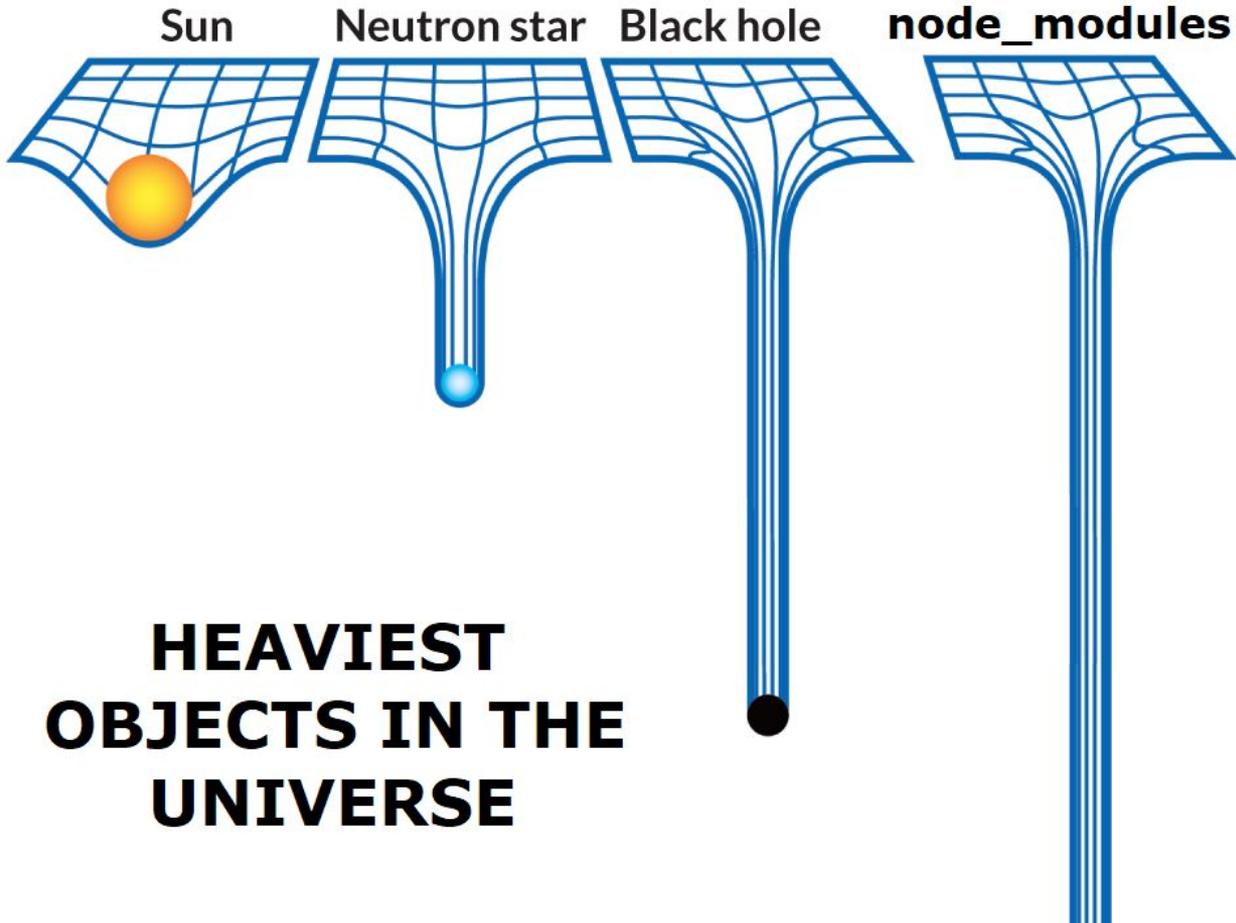
*Hustle*



*Hustle*



*Hustle*



**HEAVIEST  
OBJECTS IN THE  
UNIVERSE**

would i do it again?

-----  
< thanks! >  
-----

\ ^ ^  
 \_ \_  
 \ (oo) \-----  
 (--) \ ) \ / \  
 ||-----w ||  
 || ||

*Hustle*

```
genrule(  
  name = "my_app",  
  srcs = [  
    "my-chart.tgz",  
    "Values.yaml",  
  ],  
  outs = ["manifests"],  
  cmd = """  
    helm template \  
    --values $(location values.yaml) \  
    --output $@ \  
    $(location my-chart.tgz)  
    """,  
)
```

