BUILDING GO WITH BAZEL

PAUL BELLAMY

@PYRHHO

WHAT IS BAZEL

CORE CONCEPTS

2 LANGUAGES

BUILD

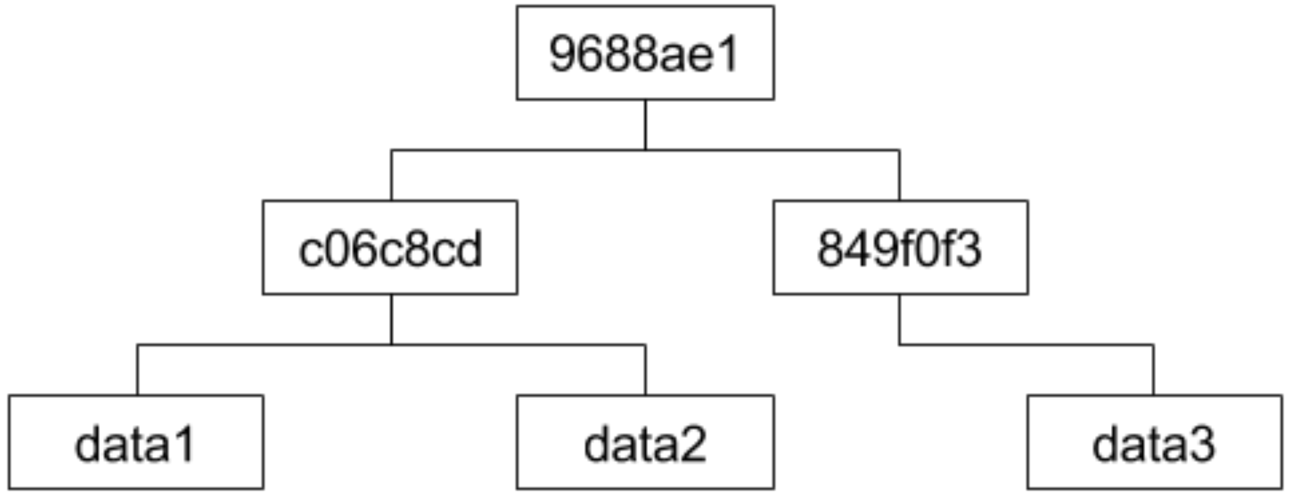
```
genrule(
  name = "hello",
  outs = ["hello_world.txt"],
  cmd = "echo Hello World > $@",
)
```

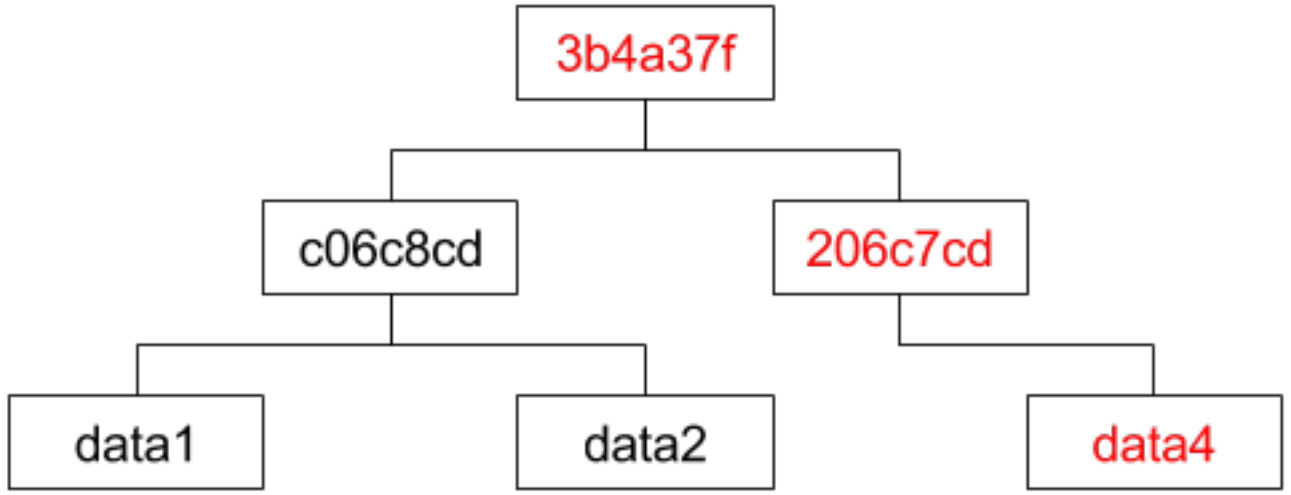
SKYLARK

```
def _impl(ctx):
  # You may use print for debugging.
  print("Rule name = %s, package = %s" % (ctx.label.name, ctx.label.package))

  # This prints the labels of the deps attribute.
  print("There are %d deps" % len(ctx.attr.deps))
  for i in ctx.attr.deps:
    print("- %s" % i.label)
    # A label can represent any number of files (possibly 0).
    print("  files = %s" % [f.path for f in i.files])

printer = rule(
    implementation=_impl,
    attrs={
      # Do not declare "name": It is added automatically.
      "number": attr.int(default = 1),
      "deps": attr.label_list(allow_files=True),
    })
```

REPEATABILITY VENDORING

INCREMENTAL


9688ae1
c06c8cd
849f0f3
data1
data2
data3



3b4a37f

c06c8cd

206c7cd

data1

data2

data4

SHARED CACHING

REMOTE WORKERS

SHARDING TESTS

```
go_test(
    name = "mytest",
    srcs = ["file_test.go"],
    library = ":go_default_library"
)
```

POLYGLOT

ALTERNATIVES

LET'S BUILD A STARTUP

WORKSPACES

RULES_DOCKER

RULES_GO

GAZELLE

```
$ gazelle \
  -go_prefix github.com/paulbellamy/example \
  -external vendored
```

or, add

```
gazelle(
  name = "gazelle",
  external = "vendored",
)

# then

$ bazel run //:gazelle
```

WEBPACK

TARGET SYNTAX

@workspace//foo/bar:wiz

foo/...

:*

REPEATABLE BUILDS

IS IT WORTH IT?

THANKS

- Big thanks to: @tom_wilkie

HTTP://GARLIC.BUILD

MORE READING

- Building Software at Google Scale https://www.youtube.com/watch?v=2qv3fcXW1mg
- Watch and rebuild (or run tests) https://github.com/bazelbuild/bazel-watcher
- Dave Cheney - Reproducible Builds http://go-talks.appspot.com/github.com/davecheney/presentations/reproducible-builds.slide#1