Brigade

Scripting container workflows on Kubernetes

Radu Matei
@matei_radu
Software Engineer, Microsoft Azure

CLOUD NATIVE
COMPUTING FOUNDATION

Docs: https://docs.brigade.sh

GitHub: https://github.com/brigadecore

© 2019 Cloud Native Computing Foundation

What is an OS?

An **operating system** is the program that, after being initially loaded into the computer, manages all the other programs.

© 2019 Cloud Native Computing Foundation

Kubernetes as a “cloud OS”

An operating system manages **processes on a machine**.

Kubernetes manages **containers on a cluster**.

© 2019 Cloud Native Computing Foundation

OS shell scripting

Flow control that wraps the execution of **processes**, but is not opinionated about what the **processes** do, or how they run.

© 2019 Cloud Native Computing Foundation

OS shell scripting

```
#!/bin/bash


for i in $( ls -1 ); do
   if [[ -d $i ]]; then
      echo "$i is a directory"
   fi
done
```


© 2019 Cloud Native Computing Foundation

Cluster shell scripting?

Flow control that wraps the execution of **containers**, but is not opinionated about what the **containers** do, or how they run.

© 2019 Cloud Native Computing Foundation

Cluster shell scripting?

```
for i in $( run_some_container ); do
  if $( run_some_other_container );
then
    $( run_yet_another_container )
  fi
done
```


© 2019 Cloud Native Computing Foundation

BRIGADE

CLOUD NATIVE SANDBOX

© 2019 Cloud Native Computing Foundation

What is Brigade?

- framework for event-driven scripting on Kubernetes
- extremely lightweight, Kubernetes native
- chain together containers to create workflows

© 2019 Cloud Native Computing Foundation

Cluster shell scripting

```
for i in $( run_some_container ); do
  if $( run_some_other_container );
then
    $( run_yet_another_container )
  fi
done
```

 | © 2019 Cloud Native Computing Foundation

Cluster shell scripting

- what programming language?
- how to share data between containers?
- when should we execute scripts?

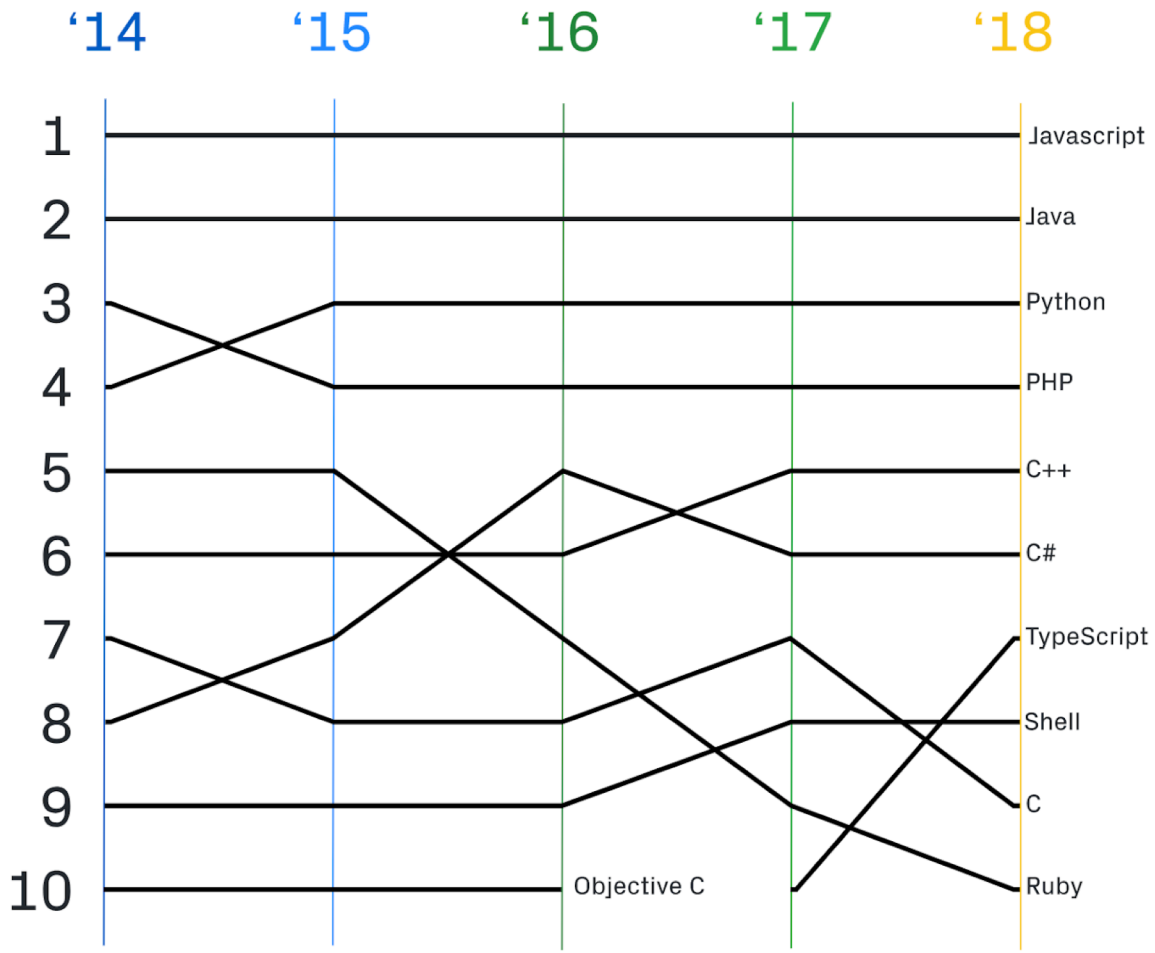
© 2019 Cloud Native Computing Foundation

Choose a language, don't write one

Top languages over time

You're coding on GitHub in hundreds of programming languages, but JavaScript still has the most contributors in public and private repositories, organizations of all sizes, and every region of the world.

This year, TypeScript shot up to #7 among top languages used on the platform overall, after making its way in the top 10 for the first time last year. TypeScript is now in the top 10 most used languages across all regions GitHub contributors come from—and across private, public, and open source repositories. *



[State of the Octoverse, 2018]

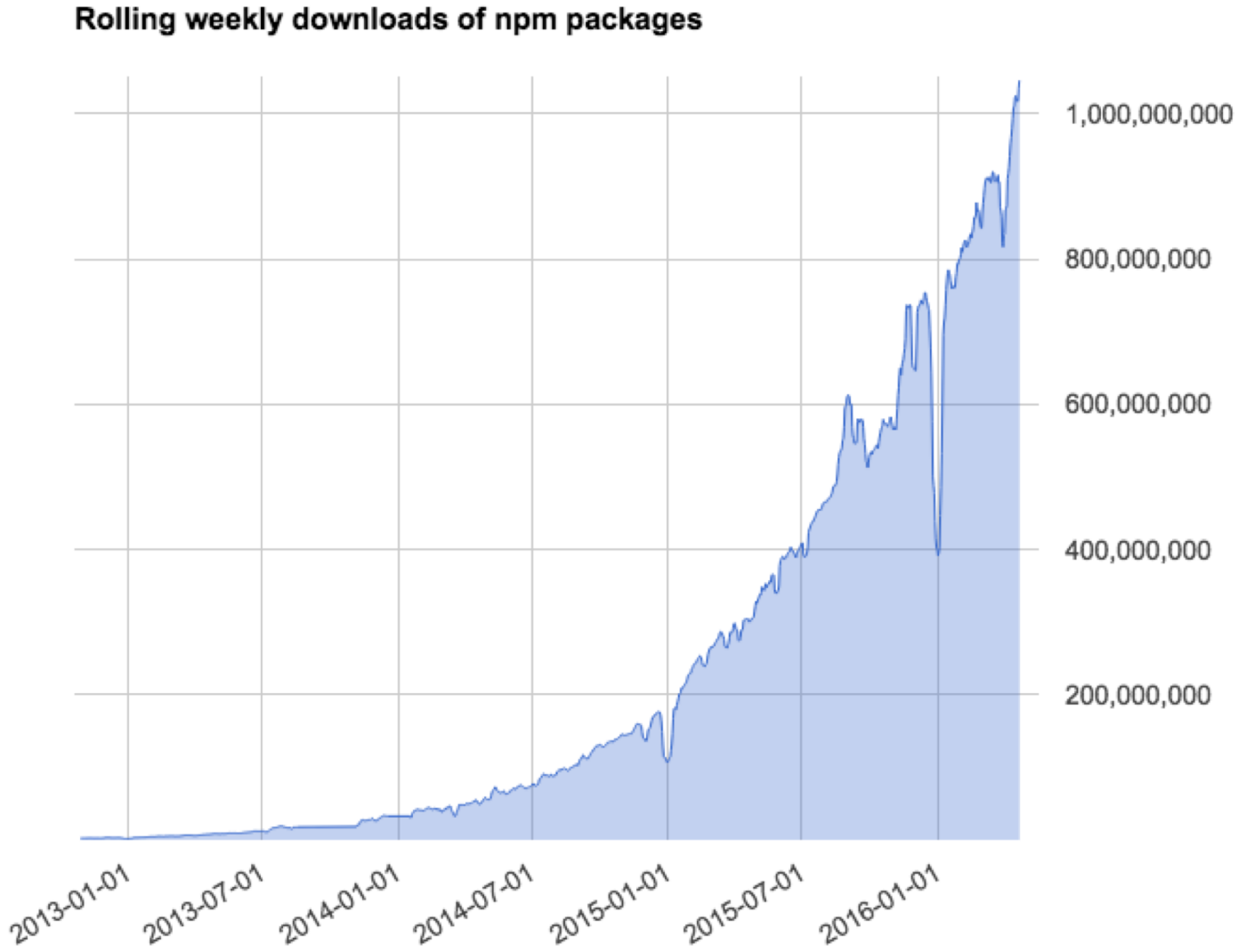
© 2019 Cloud Native Computing Foundation

Why JavaScript?

- most popular programming language
- rich ecosystem of tools
- extremely flexible

© 2019 Cloud Native Computing Foundation

1 Billion weekly NPM package pulls



blog.npmjs.org

© 2019 Cloud Native Computing Foundation

How to share data between containers?

- pass information *TO* the containers (commands, environment variables, files)

- get information *FROM* the containers (STDOUT, exit codes, files)

© 2019 Cloud Native Computing Foundation

How to share data between containers?

- JavaScript native

- preserve the existing paradigms for passing data to and from containers

© 2019 Cloud Native Computing Foundation

It's just JavaScript

```
var program = new Job("one", "alpine:3.5")
program.tasks = [ "echo hello", "echo goodbye"]
program.env = { "KEY": "value" }

program.run()
```


© 2019 Cloud Native Computing Foundation

When do we execute scripts?

- reacting to triggers / events

- event handlers become the entry points for Brigade scripts

© 2019 Cloud Native Computing Foundation

Where do events come from?

- HTTP webhooks (internal, 3rd party)
- Git events, container registry events
- Kubernetes events
- any event source can be configured for Brigade - gateways

© 2019 Cloud Native Computing Foundation

Reacting to events with Brigade

```
const { events, Job } = require("brigadier");

events.on("someEvent", ( eventData ) => {
  var program = Job("one", "alpine:3.5");
  program.tasks = [ "echo hello", "echo goodbye"];
  program.env = { "KEY": "value" };

  program.run();
});
```


© 2019 Cloud Native Computing Foundation

Use cases

- foundation for opinionated CI/CD systems
- application security scanning
- aggregating + analyzing data from multiple systems and building reports
- creating preview environments on Kubernetes for pull requests

© 2019 Cloud Native Computing Foundation

Use cases

- processing orders, connecting to external services
- actual CI/CD, integration with GitHub Checks API, BitBucket, GitLab
- any potential container workflow on Kubernetes that would benefit from actual language features (as opposed to being constrained by YAML)

© 2019 Cloud Native Computing Foundation

Getting started with Brigade

```
$ helm repo add brigade
https://brigadecore.github.io/charts
```

```
$ helm install brigade brigade/brigade
```

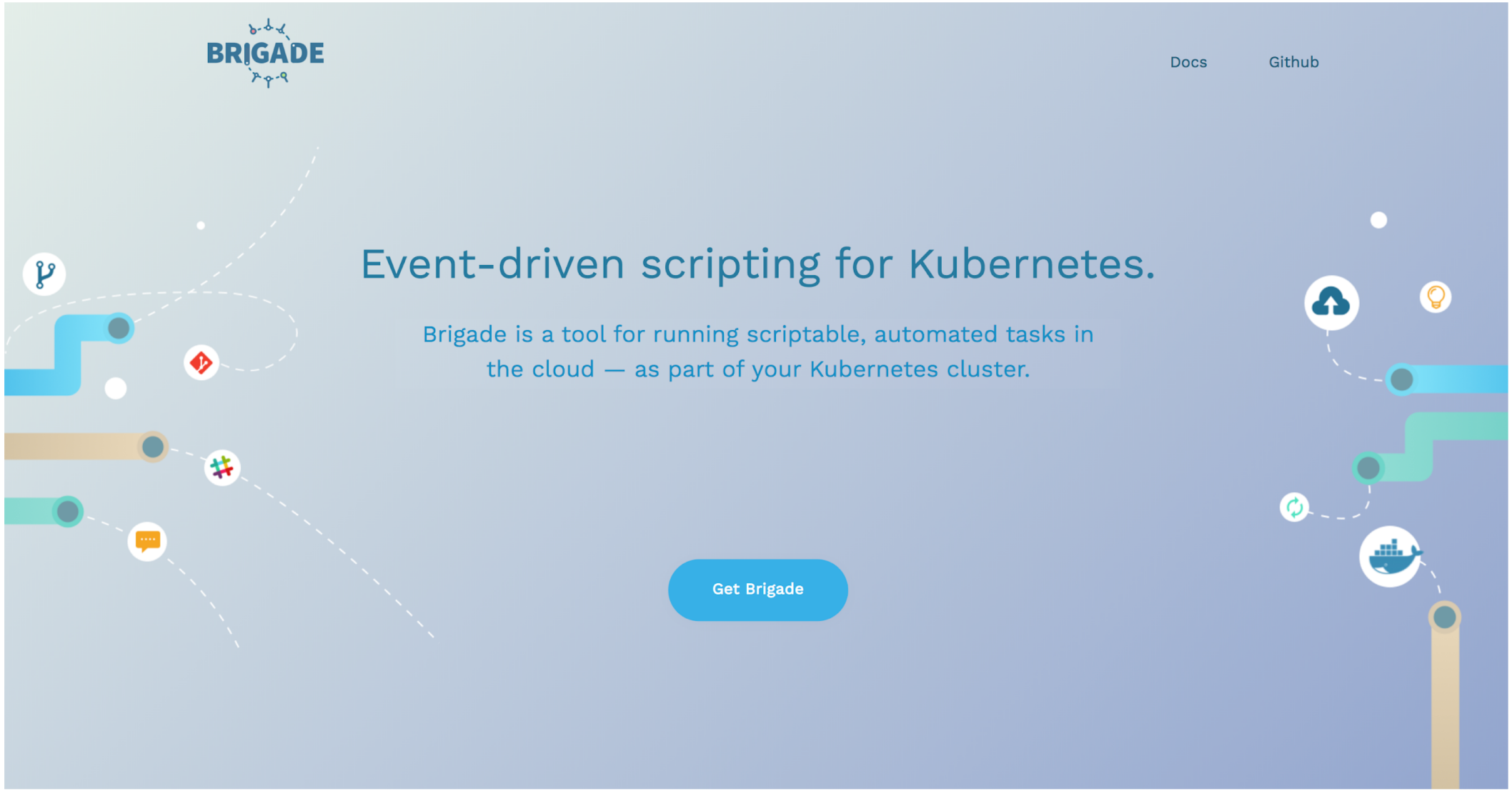
```
$ brig check
```


© 2019 Cloud Native Computing Foundation

Demo

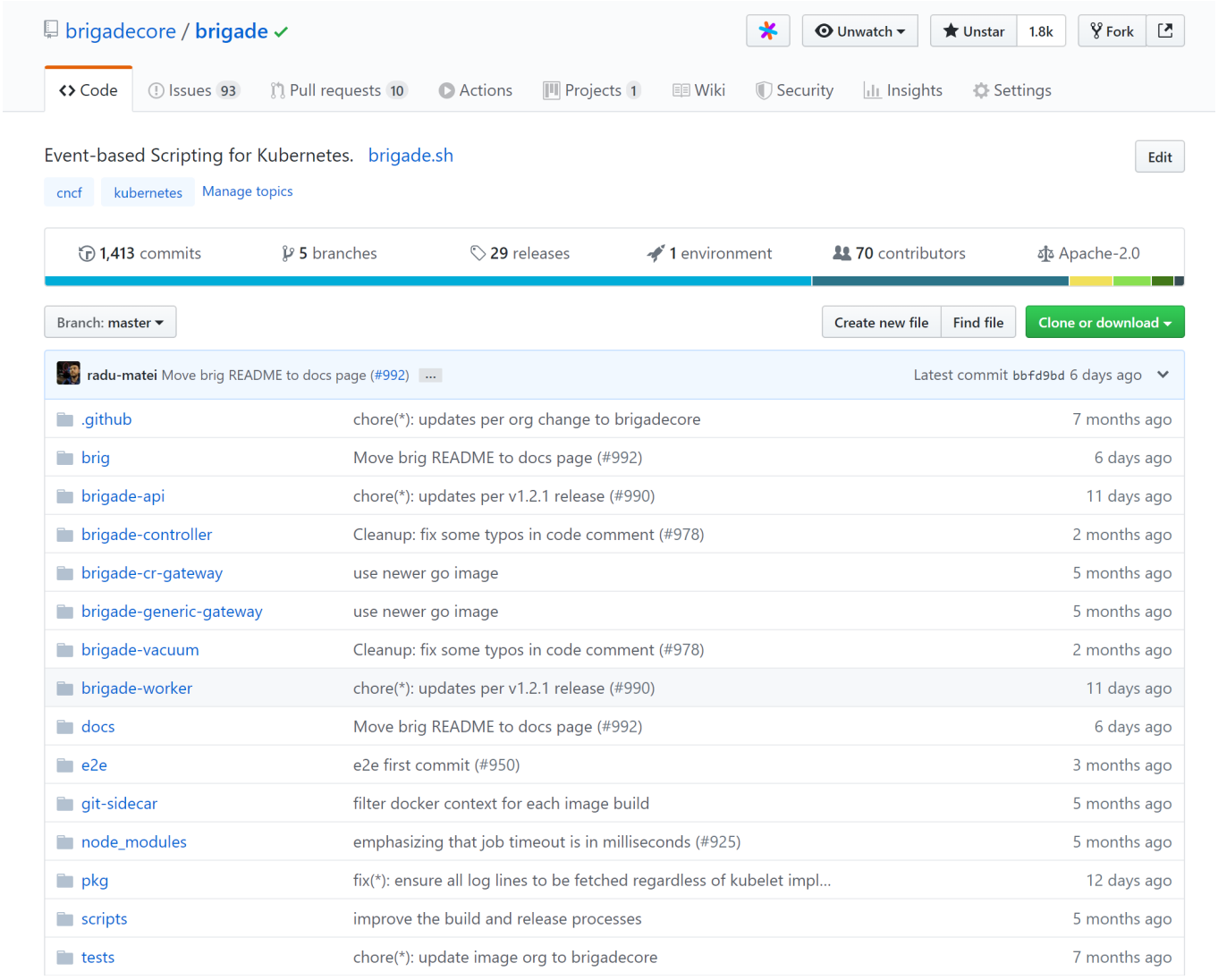
Getting started with Brigade

docs.brigade.sh



© 2019 Cloud Native Computing Foundation

Contribute - #good-first-issue



© 2019 Cloud Native Computing Foundation

Get involved!

- #brigade channel in the Kubernetes Slack
- bi-weekly community meetings, Tuesdays, 5PM PST
- github.com/brigadecore
- join us at KubeCon (intro + deep dive)

© 2019 Cloud Native Computing Foundation

TL; DR

- Brigade is a cluster scripting environment that allows you to chain multiple containers and create workflows and pipelines
- lightweight, and works on **any Kubernetes cluster**
- write scripts in basic JavaScript
- Brigade is a Kubernetes application - you can manage and monitor it the same way as any other application
- Brigade is a CNCF Sandbox Project, stable at v1.2

© 2019 Cloud Native Computing Foundation

FAQ – Where do I learn about X?

- advanced language features guide (async/await, try/catch) - https://docs.brigade.sh/topics/scripting_advanced/
- gateways - https://docs.brigade.sh/topics/gateways/
- security - https://docs.brigade.sh/topics/security/
- how to package up dependencies - https://docs.brigade.sh/topics/dependencies/
- testing Brigade scripts - https://docs.brigade.sh/topics/testing/

© 2019 Cloud Native Computing Foundation

FAQ – Declarative?

- Brigade is a cluster scripting environment - so a real programming language is required
- but… it is easy to create pipelines with Brigade
- there are already multiple implementations of declarative support on top of Brigade (at least 3 that we know of)

© 2019 Cloud Native Computing Foundation

FAQ – Windows containers?

[WIP] Experimental Windows containers support #838

Draft **radu-matei** wants to merge 1 commit into `brigadecore:master` from `radu-matei:windows-entrypoint`

Conversation 0 | Commits 1 | Checks 2 | Files changed 1

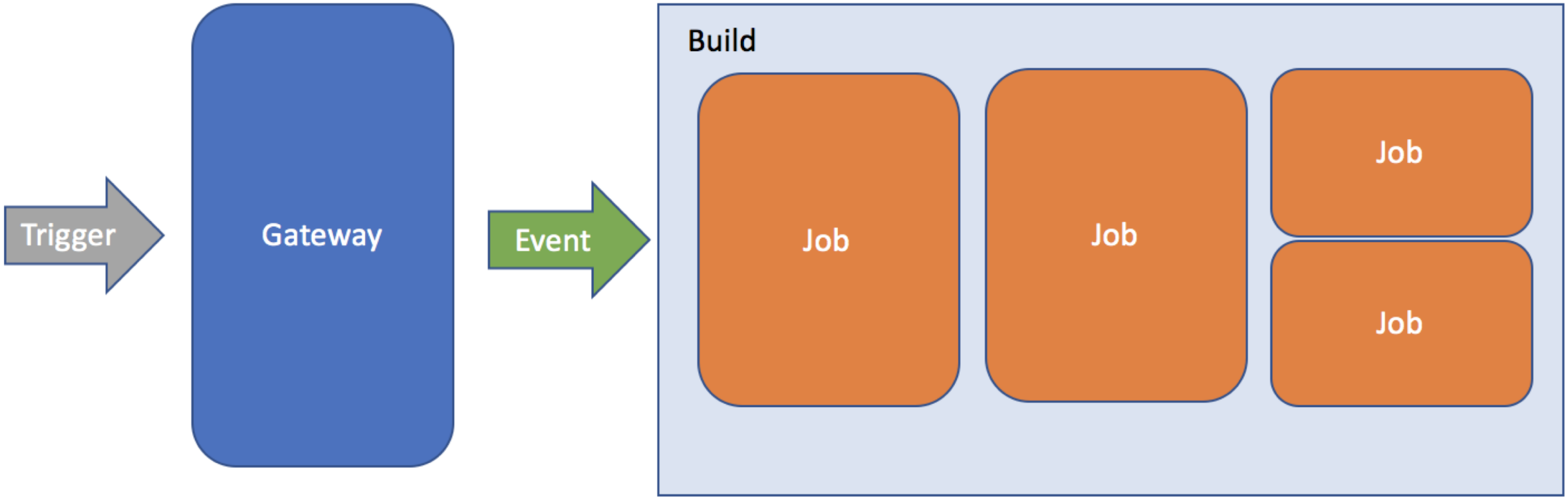
radu-matei on 8 Mar — Member

What this PR does / why we need it:
This PR adds very early experimental Windows Server containers support.
It has been tested with a Virtual Kubelet node backed by Azure Container Instances, so the following are current known limitations:

- job and build caches are not supported (limitation of volumes in ACI)
- the Git sidecar is not supported (limitation of volumes in ACI)
- the entrypoint is passed directly to the command arguments, not constructed as a script (limitation of volumes in ACI)

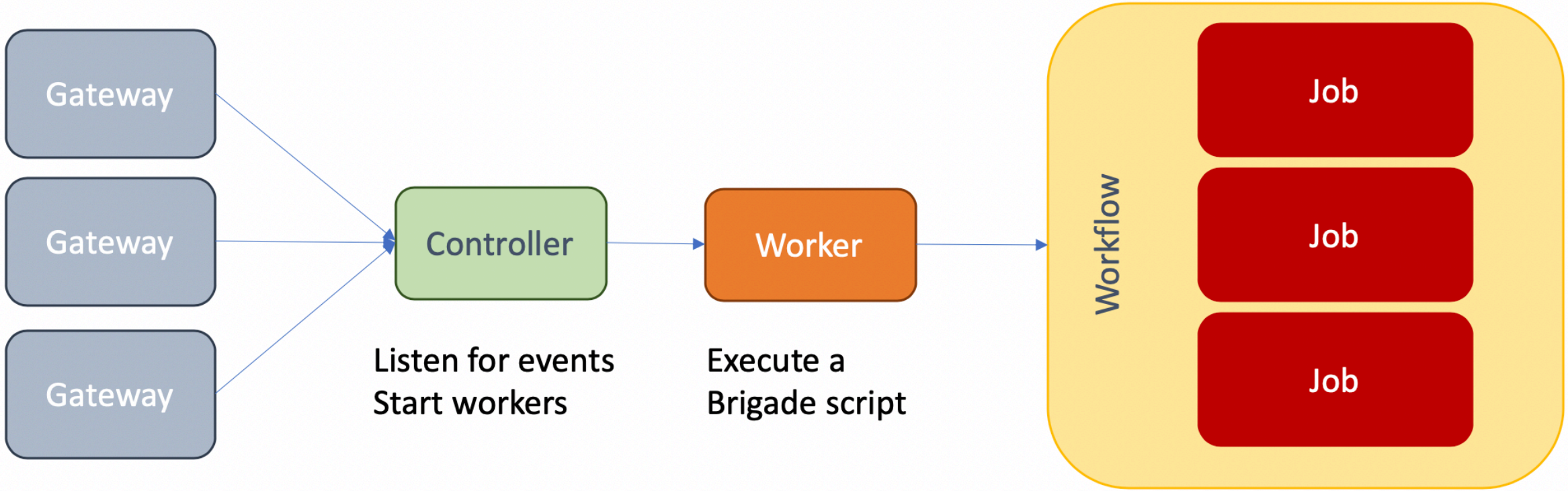
© 2019 Cloud Native Computing Foundation

FAQ – how does Brigade work?



© 2019 Cloud Native Computing Foundation

FAQ – how does Brigade work?



Based on something external create an event.

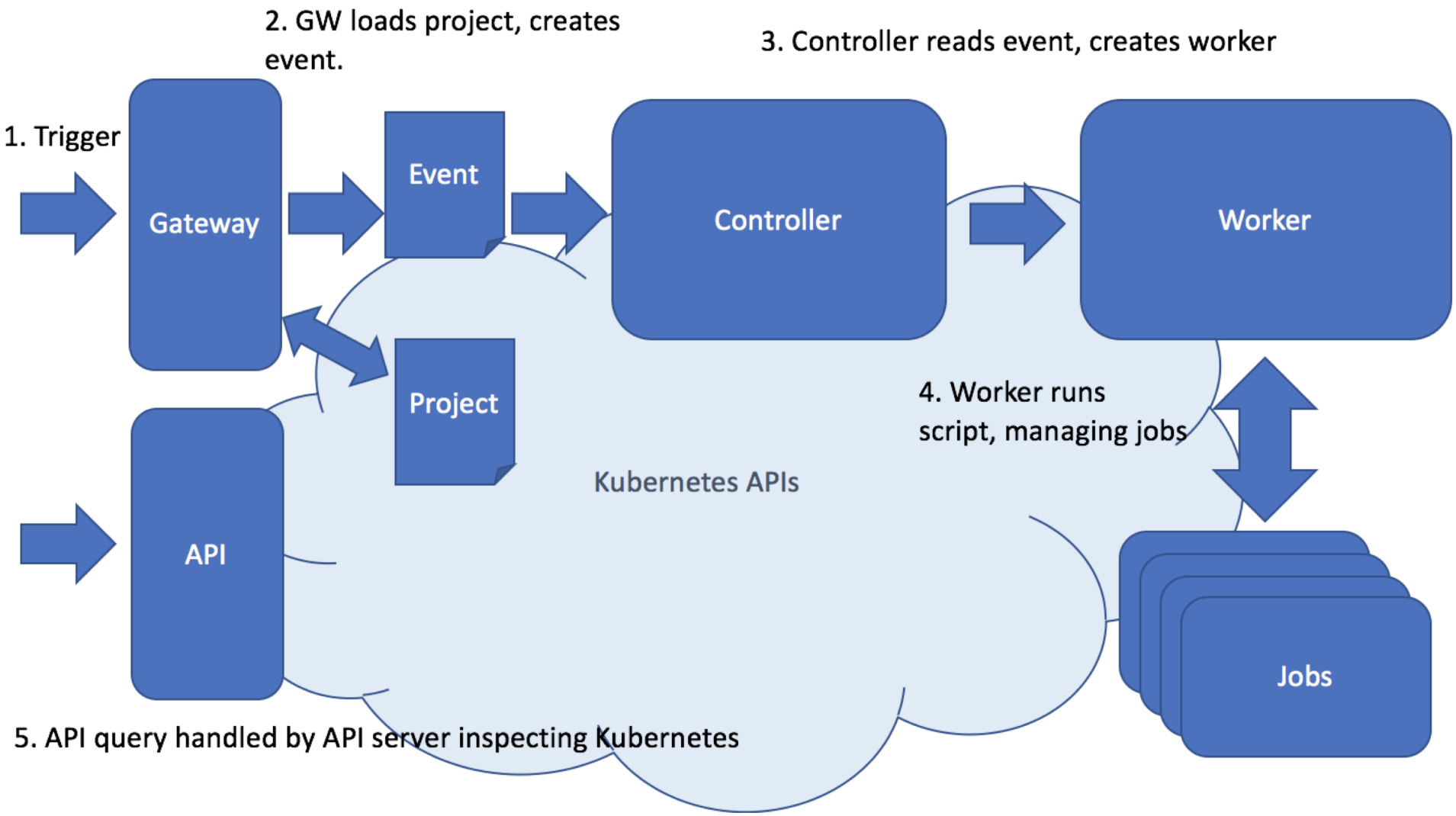
Listen for events
Start workers

Execute a Brigade script

Run jobs to completion, where each job is a step in a workflow.

© 2019 Cloud Native Computing Foundation

FAQ – how does Brigade work?



© 2019 Cloud Native Computing Foundation