

# Brigade

Scripting container workflows on Kubernetes

Radu Matei  
@matei\_radu  
Software Engineer, Microsoft Azure



# Who am I?

- Radu Matei ([@matei\\_radu](https://twitter.com/matei_radu), <https://radu-matei.com>)
- open source maintainer for Brigade, CNAB (hear me ramble about developer tools, distributed systems, OCI, security protocols)
- really like music and bicycles
- software engineer at Microsoft Azure (Deis Labs )





Docs: <https://docs.brigade.sh>

GitHub: <https://github.com/brigadecore>



# What is an OS?

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



# Kubernetes as a “cloud OS”

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

Kubernetes manages **containers on a cluster.**



# OS shell scripting

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



# OS shell scripting

```
#!/bin/bash
```

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



# Cluster shell scripting?

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





# Cluster shell scripting?

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



# How have we done this so far?

- bash scripts
- abusing YAML





# BRIGADE



**CLOUD NATIVE**  
**SANDBOX**



# What is Brigade?

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



# Cluster shell scripting

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



# Cluster shell scripting

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

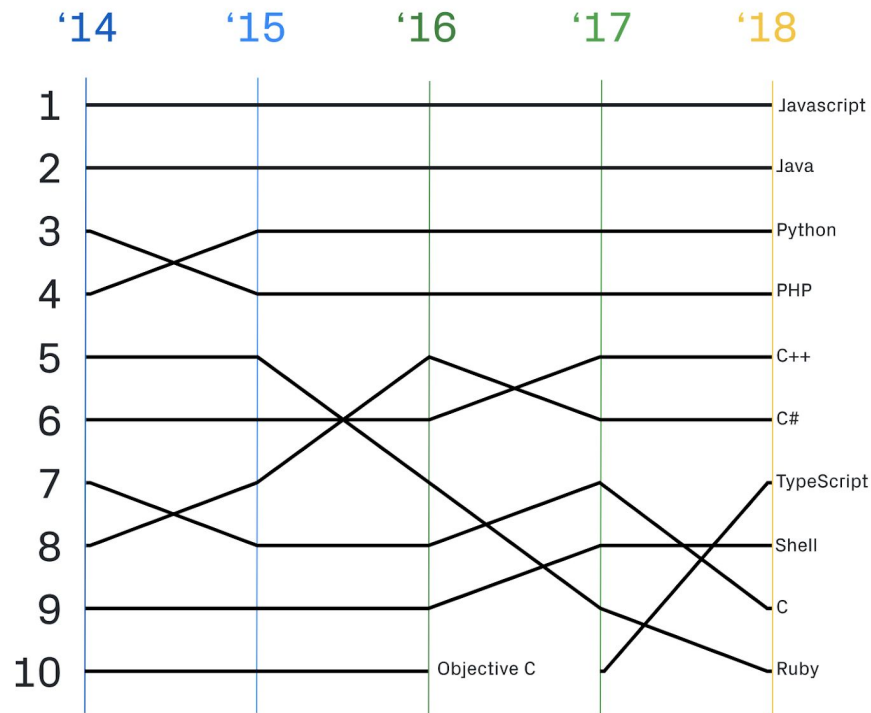


# Choose a language, don't write one

## 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. \*



# Why JavaScript?

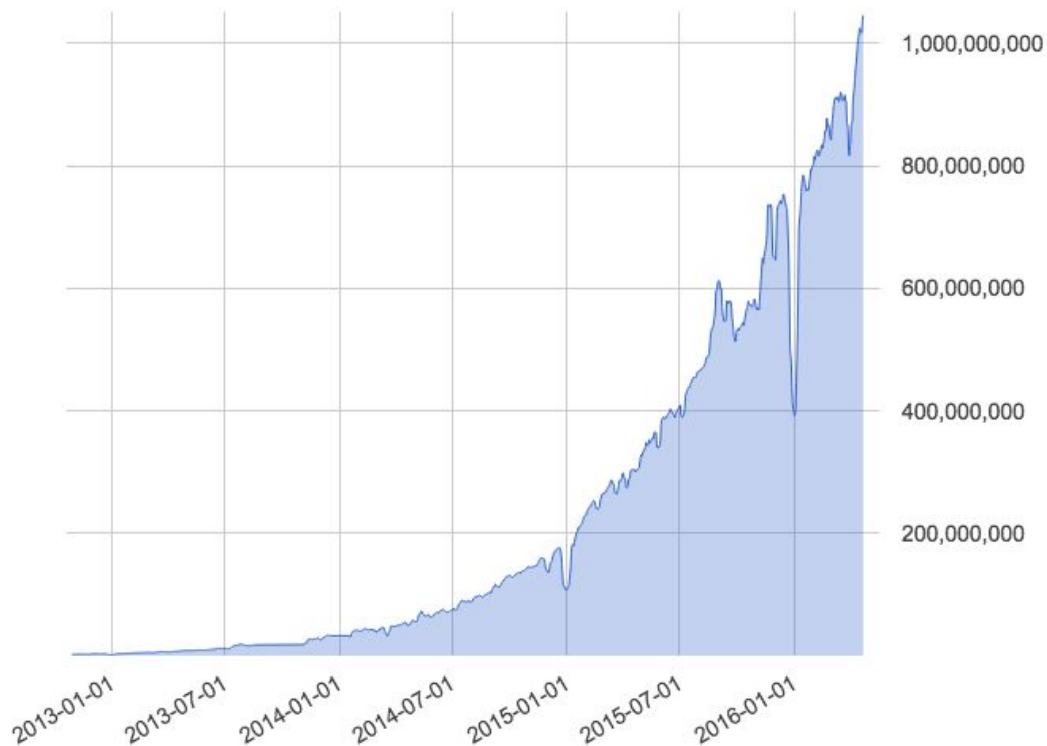
- most popular programming language
- rich ecosystem of tools
- extremely flexible





# 1 Billion weekly NPM package pulls

Rolling weekly downloads of npm packages



# How to share data between containers?

- pass information *TO* the containers (commands, environment variables, files)
  
  
  
  
  
  
  
  
  
  
- get information *FROM* the containers (STDOUT, exit codes, files)



# How to share data between containers?

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



# It's just JavaScript

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

program.run()
```



# When do we execute scripts?

- reacting to triggers / events
  
  
  
  
  
  
  
  
  
  
- event handlers become the entry points for Brigade scripts



# Where do events come from?

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



# 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();
});
```



# 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





# 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)



# Getting started with Brigade

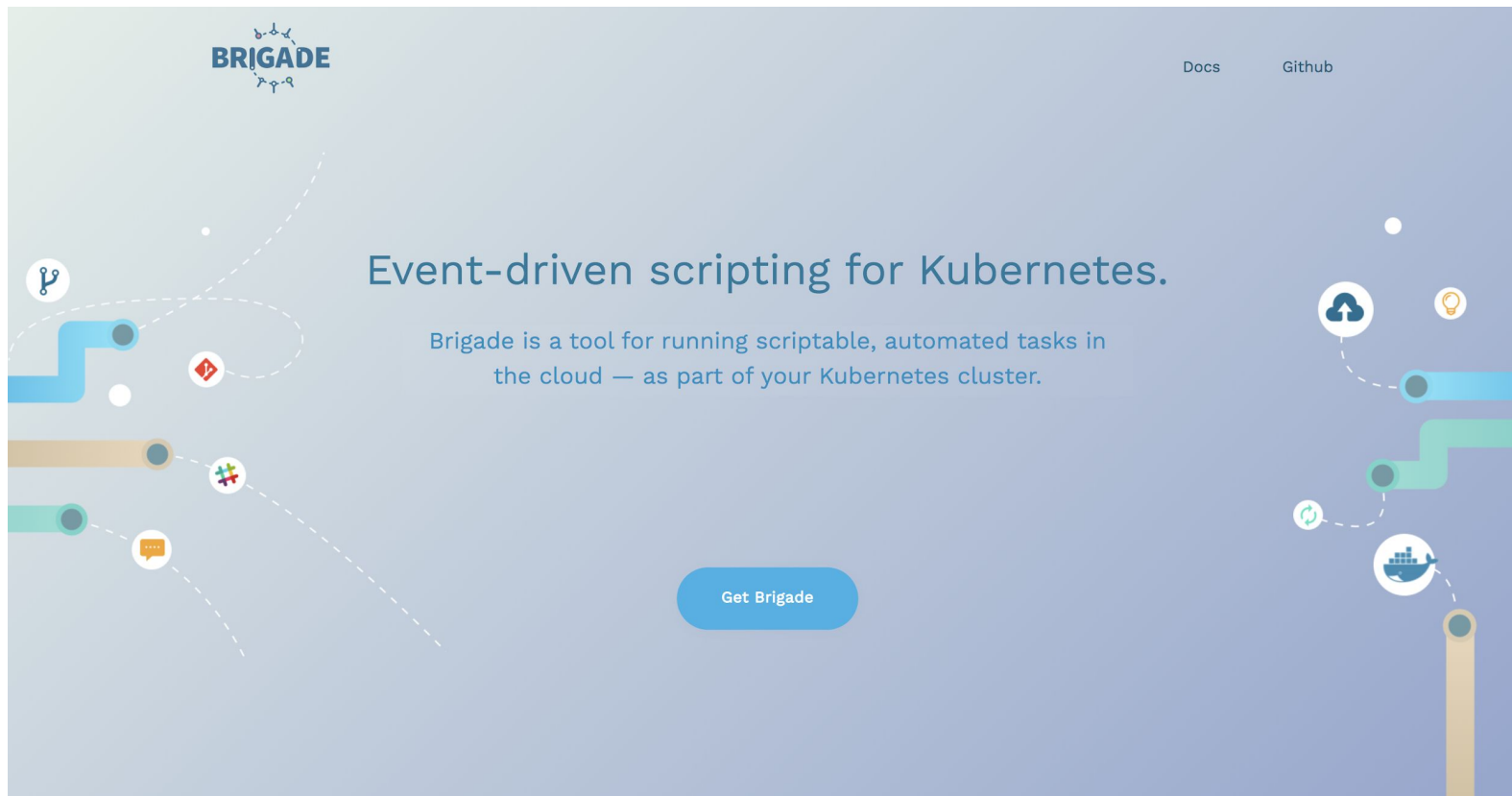
```
$ helm repo add brigade  
https://brigadecore.github.io/charts
```

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

```
$ brig check
```



# docs.brigade.sh



# Demo

## Getting started with Brigade

# Components

- `controller` - listens for events and creates worker pods
- `worker pods` - execute scripts and schedule jobs
- `API (optional)` - used to interact with Brigade from web or CLI
- `gateways` - map outside events into Brigade



# Gateways

(<https://docs.brigade.sh/topics/gateways>)

- translate events from outside Brigade
- out of the box - generic gateway (CloudEvents or arbitrary JSON), container registry, GitHub
- they can be webhooks, or any application that can create a Kubernetes secret (`brig run` acts like a gateway)
- examples for Go, NodeJS, Bash, Python



# Community integrations

- built-in support for CloudEvents (CNCF)
- integration with the GitHub Checks API
- Integrations with Prometheus and Virtual Kubelet (CNCF)
- documentation on using Linkerd (CNCF) soon



# Ecosystem

## Related Projects

---

- [Kashti](#) - a dashboard for your Brigade pipelines.
- [Brigadeterm](#) - a simple terminal ui for brigade pipelining system.
- [Brigade exporter](#) - a [Prometheus](#) exporter to gather metrics from Brigade.
- Gateways
  - [BitBucket events](#): Gateway Support for BitBucket repositories
  - [GitLab events](#): Gateway Support for GitLab repositories
  - [Kubernetes events](#): Gateway that listens to Kubernetes event stream
  - [Event Grid gateway](#): Gateway for Azure Event Grid events
  - [Cron Gateway](#): Schedule events to run at a particular time
  - [Trello and Generic Webhooks](#): Experimental gateway for Trello and for generic webhooks
  - [Draft Pack for Building Custom Gateways](#): Build your own gateway [in 5 minutes](#)
  - [Azure DevOps / VSTS gateway](#): Gateway for Azure DevOps / VSTS events





# Contribute - #good-first-issue

The screenshot shows the GitHub repository page for 'brigadecore / brigade'. At the top, there are navigation links for Code, Issues (93), Pull requests (10), Actions, Projects (1), Wiki, Security, Insights, and Settings. Below this, the repository name 'brigadecore / brigade' is displayed with a checkmark, along with 'Unwatch', 'Unstar', '1.8k', 'Fork', and 'Share' buttons. The main heading is 'Event-based Scripting for Kubernetes. brigade.sh', with an 'Edit' button. Below the heading are tags for 'cncf' and 'kubernetes', and a 'Manage topics' link. A summary bar shows '1,413 commits', '5 branches', '29 releases', '1 environment', '70 contributors', and 'Apache-2.0' license. Below this is a 'Branch: master' dropdown and buttons for 'Create new file', 'Find file', and 'Clone or download'. The main content is a list of files and directories with their commit messages and dates. The latest commit is by 'radu-matei' moving the brig README to docs page (#992) 6 days ago.

brigadecore / brigade ✓

Unwatch Unstar 1.8k Fork

Code Issues 93 Pull requests 10 Actions Projects 1 Wiki Security Insights Settings

Event-based Scripting for Kubernetes. brigade.sh Edit

cncf kubernetes Manage topics

1,413 commits 5 branches 29 releases 1 environment 70 contributors Apache-2.0

Branch: master Create new file Find file Clone or download

radu-matei	Move brig README to docs page (#992)	Latest commit bbfd9bd 6 days ago
.github	chore(*): updates per org change to brigadecore	7 months ago
brig	Move brig README to docs page (#992)	6 days ago
brigade-api	chore(*): updates per v1.2.1 release (#990)	11 days ago
brigade-controller	Cleanup: fix some typos in code comment (#978)	2 months ago
brigade-cr-gateway	use newer go image	5 months ago
brigade-generic-gateway	use newer go image	5 months ago
brigade-vacuum	Cleanup: fix some typos in code comment (#978)	2 months ago
brigade-worker	chore(*): updates per v1.2.1 release (#990)	11 days ago
docs	Move brig README to docs page (#992)	6 days ago
e2e	e2e first commit (#950)	3 months ago
git-sidecar	filter docker context for each image build	5 months ago
node_modules	emphasizing that job timeout is in milliseconds (#925)	5 months ago
pkg	fix(*): ensure all log lines to be fetched regardless of kubelet impl...	12 days ago
scripts	improve the build and release processes	5 months ago
tests	chore(*): update image org to brigadecore	7 months ago



# Get involved!

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



# 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



# FAQ – Where do I learn about X?

- advanced language features guide (async/await, try/catch) – [https://docs.brigade.sh/topics/scripting\\_advanced/](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/>



# 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)



# FAQ – Windows containers?

## [WIP] Experimental Windows containers support #838

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

 Conversation 0

 Commits 1

 Checks 2

 Files changed 1



radu-matei on 8 Mar

Member



### What this PR does / why we need it:

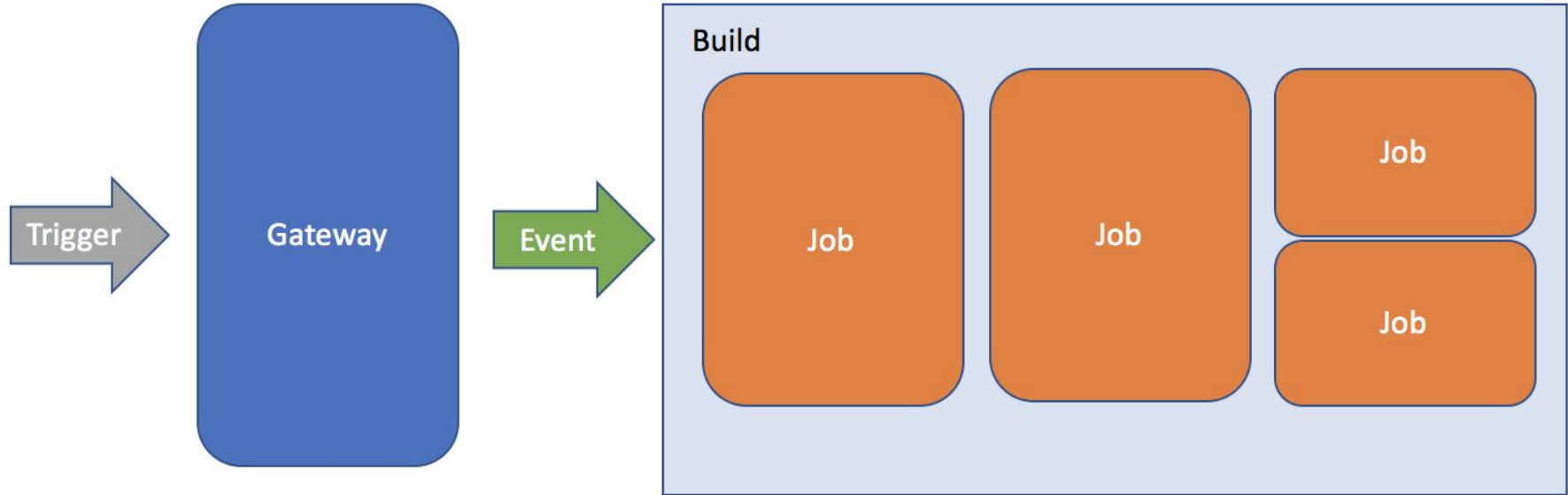
This PR adds very early experimental Windows Server containers support.

It has been tested with a Virtual Kubelet node backed by Azure Container Instances, so the following are current known limitations:

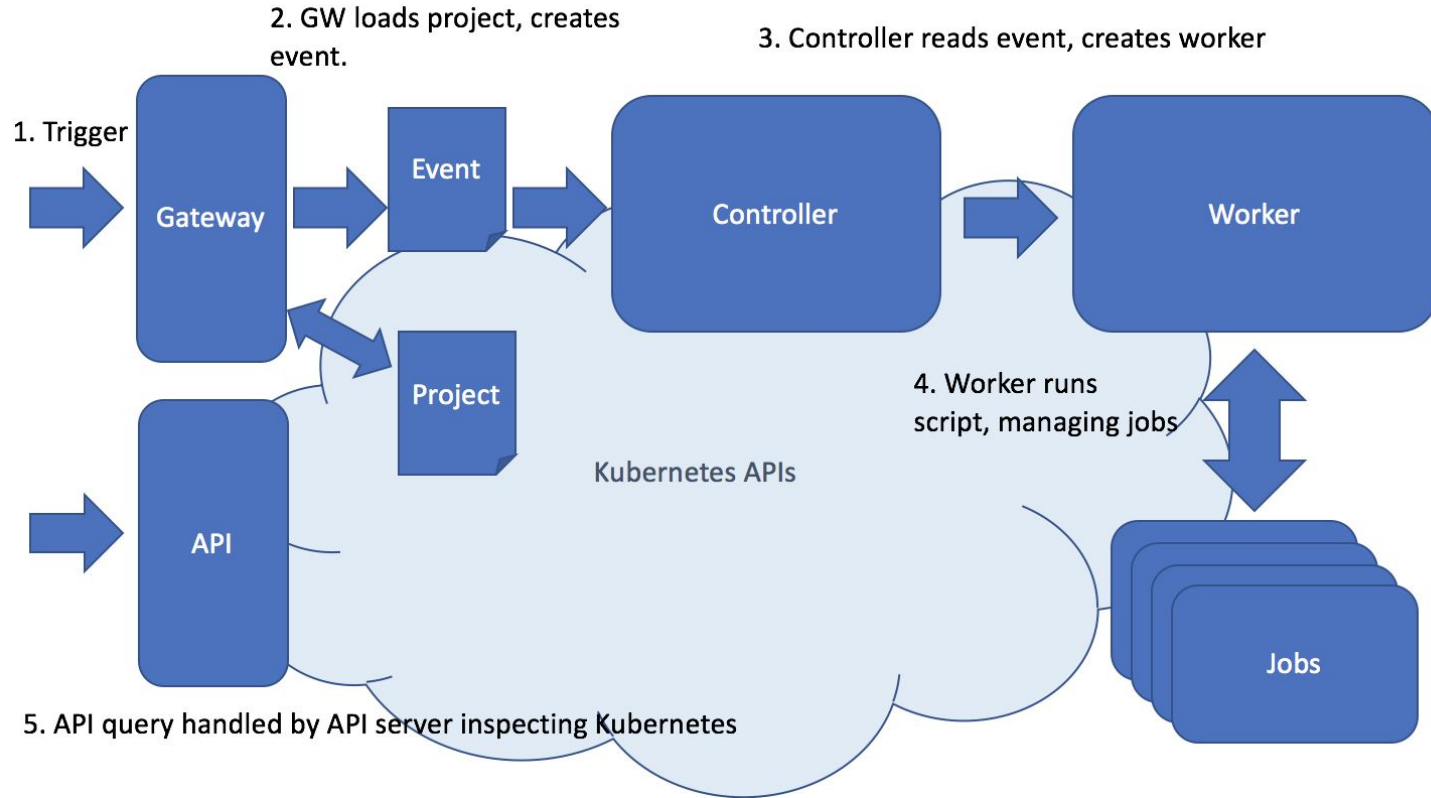
- job and build caches are not supported (limitation of volumes in ACI)
- the Git sidecar is not supported (limitation of volumes in ACI)
- the entrypoint is passed directly to the command arguments, not constructed as a script (limitation of volumes in ACI)



# FAQ – how does Brigade work?



# FAQ – how does Brigade work?





# Securing Brigade

(<https://docs.brigade.sh/topics/security>)

- isolate Brigade in a namespace
- use Kubernetes RBAC (enabled by default)
- always use HTTPS for exposed services  
(<https://docs.brigade.sh/topics/ingress>)
- always use trusted container images in jobs
- always validate and authenticate events



# Dependencies in brigade.js

(<https://docs.brigade.sh/topics/dependencies>)

- `import` local JavaScript modules from the repo
- pull packages from NPM
- build a custom worker image with added dependencies (<https://docs.brigade.sh/topics/workers>)
- BUT - avoid putting too much logic in `brigade.js`



# Testing Brigade scripts

(<https://docs.brigade.sh/topics/testing>)

- brigetest

(<https://github.com/technosophos/brigetest>)

- checks if brigade.js is properly formatted, execute unit tests
- does not run jobs, does not need Kubernetes

- Brigade Integration Test

(<https://github.com/blimmer/brigade-project-integration-test>)

- integration testing using Minikube
- automatically configures Brigade, creates a project, invokes events, tests output

