



KubeCon



CloudNativeCon

North America 2019

# Intro: Kubernetes SIG Apps

*Adnan Abdulhussein  
Matt Farina*



# SIG Chairs



KubeCon



CloudNativeCon

North America 2019



**Matt Farina**  
Samsung SDS  
@mattfarina



**Ken Owens**  
Brex  
@kow3ns



**Adnan Abdulhussein**  
VMware  
@prydonius



**Janet Kuo**  
Google  
@janetkuo



KubeCon



CloudNativeCon

North America 2019



# In The Beginning



KubeCon



CloudNativeCon

North America 2019

Why GitHub? ▾ Enterprise Explore ▾ Marketplace Pricing ▾

Search / Sign in Sign up

kubernetes / kubernetes

Code Issues 2,273 Pull requests 1,092 Projects 9 Security Insights

Branch: release-1.0 kubernetes / pkg / controller / Create new file Find file History

This branch is 544 commits ahead, 70882 commits behind master. Pull request Compare

mikedanese run gofmt on everything we touched Latest commit 0496a2f on Aug 17, 2015

..

framework	rewrite go imports	4 years ago
controller_utils.go	run gofmt on everything we touched	4 years ago
doc.go	Make copyright ownership statement generic	5 years ago
replication_controller.go	run gofmt on everything we touched	4 years ago
replication_controller_test.go	rewrite go imports	4 years ago

# Way Too Early Helm



KubeCon



CloudNativeCon

North America 2019

This repository has been archived by the owner. It is now read-only.

[helm / helm-classic](#) Archived

Watch 31 Star 583 Fork 54

Code Issues 42 Pull requests 0 Projects 0 Security Insights

**⚠ (OBSOLETE) Helm Classic v1 <https://github.com/helm/helm>**

158 commits 8 branches 12 releases 26 contributors View license

Tag: 0.0.1 ▾ Find file Clone or download ▾

Author	Commit Message	Date
	Merge pull request #120 from technosophos/feat/git-logger ...	Latest commit da7874b on Nov 2, 2015
	feat(ci): adjust include/upload patterns	4 years ago
	Merge pull request #59 from gabrtv/publish	4 years ago
	Merge pull request #120 from technosophos/feat/git-logger	4 years ago
	doc(skel) make the skeleton Chart generic	4 years ago
	feat(test): add ci test infrastructure scaffolding	4 years ago
	feat(ci): add publishing of binaries	4 years ago
	test(travis): speed up glide pkg handling	4 years ago

# SIG Apps Is Born



KubeCon



CloudNativeCon

North America 2019

 mattfarina Adding SIG Apps dce5eb4 on May 11, 2016

1 contributor

22 lines (17 sloc) | 928 Bytes Raw Blame History   

## (SIG Apps)

A Special Interest Group for deploying and operating applications in Kubernetes.

### Goals:

- Discuss running applications in k8s
- Discuss how to define and run apps in k8s (APIs, CLIs, SDKs, package management tools, etc.)
- Suggest k8s features where we see friction
- Be the voice of the people running applications into the k8s development (developers and devops)
- Help people get involved in the kubernetes community
- Show early features/demos of tools that make running apps easier

## SIG Apps Charter

This charter adheres to the conventions described in the [Kubernetes Charter README](#) and uses the Roles and Organization Management outlined in [sig-governance](#).

### Scope

SIG Apps covers developing, deploying, and operating applications on Kubernetes with a focus on the application developer and application operator experience.

#### In scope

##### Code, Binaries and Services

- APIs used for running applications (e.g., Workloads API)
- Tools and documentation to aid in ecosystem tool interoperability around apps (e.g., Application CRD/Controller)
- Grandfathered in tools used to aid in development of and management of workloads (e.g., Kompose)

##### Cross-cutting and Externally Facing Processes

- A discussion platform for solving app development and management problems
- Represent the needs and persona of application developers and operators

# Personas / Roles

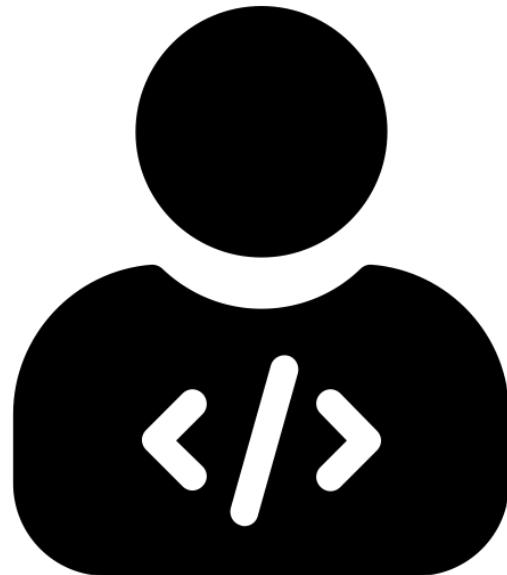


KubeCon

CloudNativeCon

North America 2019

## Application Developer



Created by Shastry  
from Noun Project

## Application Operator



Created by Adrien Coquet  
from Noun Project

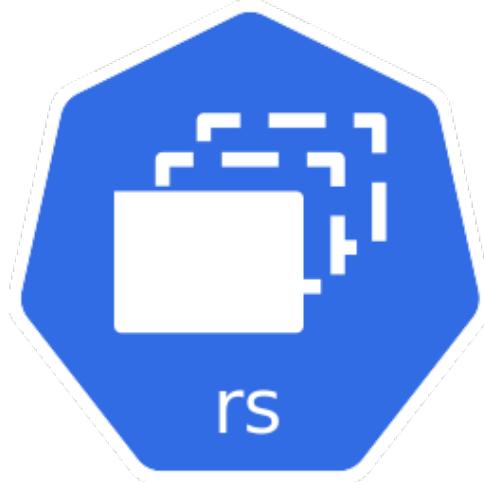
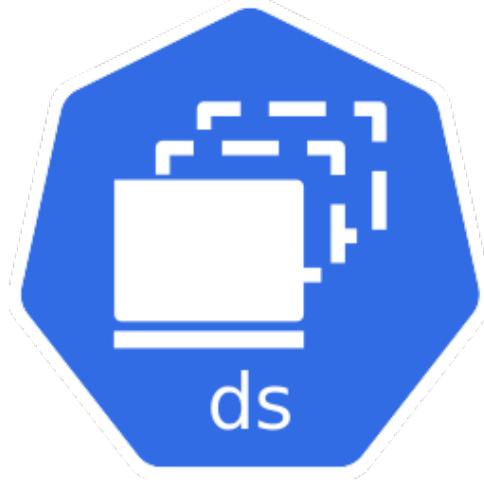
# Some Core Controllers...



KubeCon

CloudNativeCon

North America 2019



And more...

# Application CRD/Controller



KubeCon



CloudNativeCon

North America 2019

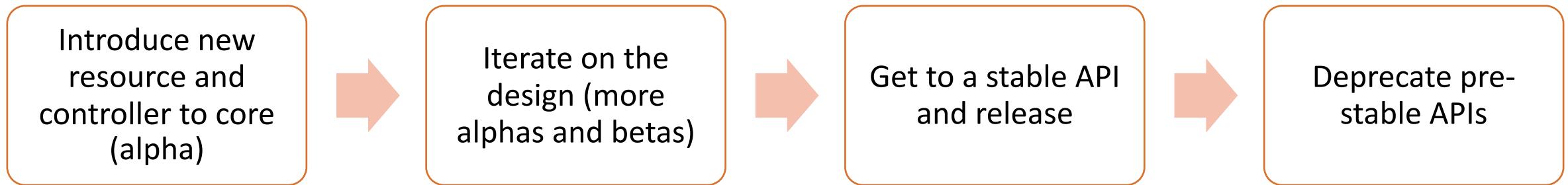
Screenshot of the GitHub repository for `kubernetes-sigs/application`.

The repository has 129 commits, 2 branches, 0 packages, 1 release, 21 contributors, and is licensed under Apache-2.0.

Recent activity includes:

- `k8s-ci-robot` Merge pull request #126 from janetkuo/patch-1
- `cmd/manager` add comment for reconcile time
- `config` Merge pull request #103 from barney-s/issue\_102
- `docs` Merge pull request #56 from tossmilestone/fix-example-error
- `e2e` WIP: Upgrade to version 1.0.5 of Kubebuilder
- `hack` WIP: Upgrade to version 1.0.5 of Kubebuilder
- `pkg` Merge pull request #106 from barney-s/issue\_105
- `vendor` Absorb vendor/kubesdk into application/pkg. Since kubesdk is vendored...
- `.gitignore` WIP: Upgrade to version 1.0.5 of Kubebuilder
- `CONTRIBUTING.md` Upgrade resource to v1beta1

# Old Process For New To Core



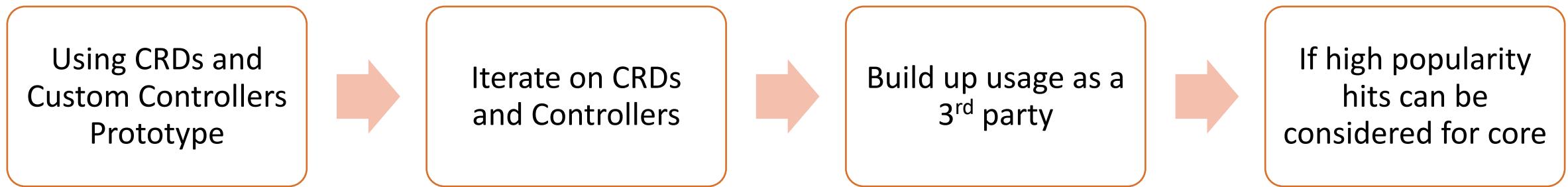
# Current Process



KubeCon

CloudNativeCon

North America 2019



\* This process can happen as a SIG sponsored project like the Application CRD / Controller

# Execution Hook



KubeCon

CloudNativeCon

North America 2019

## Summary

---

This proposal is to introduce an API (`ExecutionHook`) for dynamically executing user's commands in a pod/container or a group of pods/containers and a controller (`ExecutionHookController`) to manage the hook lifecycle. `ExecutionHook` provides a general mechanism for users to trigger hook commands in their containers for their different use cases. Different options have been evaluated to decide how this `ExecutionHook` should be managed and executed. The preferred option is described in the `Proposal` section. The other options are discussed in the `Alternatives` section.

## Motivation

---

The volume snapshot feature allows creating/deleting volume snapshots, and the ability to create new volumes from a snapshot natively using the Kubernetes API. However, application consistency is not guaranteed. An user has to figure out how to quiesce an application before taking a snapshot and unquiesce it after taking the snapshot.

So we want to introduce an `ExecutionHook` to facilitate the quiesce and unquiesce actions when taking a snapshot. There is an existing lifecycle hook in the `Container` struct. The lifecycle hook is called immediately after a container is created or immediately before a container is terminated. The proposed execution hook is not tied to the start or termination time of the container. It can be triggered on demand by callers (users or controllers) and the status will be updated dynamically.

\* Notice: This is very new

# Documentation



A screenshot of the Kubernetes Concepts page for Deployments. The page has a dark header with the Kubernetes logo and navigation links: Documentation, Blog, Partners, Community, Case Studies, English, v1.16. The main content area has a title "Deployments" with a blue edit icon. It contains text about what Deployments are and how they work, followed by a note about managing ReplicaSets. A sidebar on the left lists various Kubernetes concepts, with "Deployments" highlighted.

A screenshot of the Kubernetes Tutorials page for the PHP Guestbook application. The page has a dark header with the Kubernetes logo and navigation links: Documentation, Blog, Partners, Community, Case Studies, English, v1.16. The main content area has a title "Example: Deploying PHP Guestbook application with Redis" with a blue edit icon. It contains text about the tutorial's purpose and components, followed by a list of objectives and steps. A sidebar on the left lists various Kubernetes tutorials, with "Stateless Applications" highlighted.

## Concepts

- ▶ Overview
- ▶ Cluster Architecture
- ▶ Containers
- ▶ Workloads
  - ▶ Pods
  - ▶ Controllers
    - ReplicaSet
    - ReplicationController
    - ▶ Deployments
    - StatefulSets
    - DaemonSet
    - Garbage Collection
    - TTL Controller for Finished Resources
    - Jobs - Run to Completion
    - CronJob
  - ▶ Services, Load Balancing, and Networking
  - ▶ Storage
  - ▶ Configuration
  - ▶ Security
  - ▶ Policies
  - ▶ Scheduling
  - ▶ Cluster Administration
  - ▶ Extending Kubernetes

## Deployments



A Deployment provides declarative updates for Pods and ReplicaSets.

You describe a *desired state* in a Deployment, and the Deployment Controller changes the actual state to the desired state at a controlled rate. You can define Deployments to create new ReplicaSets, or to remove existing Deployments and adopt all their resources with new Deployments.

**Note:** Do not manage ReplicaSets owned by a Deployment. Consider opening an issue in the main Kubernetes repository if your use case is not covered below.

- Use Case
- Creating a Deployment
- Updating a Deployment
- Rolling Back a Deployment
- Scaling a Deployment
- Pausing and Resuming a Deployment
- Deployment status
- Clean up Policy
- Canary Deployment
- Writing a Deployment Spec
- Alternative to Deployments

## Use Case

The following are typical use cases for Deployments:

- Create a Deployment to rollout a ReplicaSet. The ReplicaSet creates Pods in the background. Check the status of the rollout to see if it succeeds or not.
- Declare the new state of the Pods by updating the PodTemplateSpec of the Deployment. A new ReplicaSet is created and the Deployment manages moving the Pods from the old ReplicaSet to the new one at a controlled rate. Each new ReplicaSet updates the revision of the Deployment.

## Tutorials

- ▶ Hello Minikube
- ▶ Learn Kubernetes Basics
- ▶ Online Training Courses
- ▶ Configuration
- ▶ Stateless Applications
  - Exposing an External IP Address to Access an Application in a Cluster
  - Example: Deploying PHP Guestbook application with Redis
  - Example: Add logging and metrics to the PHP / Redis Guestbook example
- ▶ Stateful Applications
- ▶ Clusters
- ▶ Services

## Example: Deploying PHP Guestbook application with Redis

This tutorial shows you how to build and deploy a simple, multi-tier web application using Kubernetes and Docker. This example consists of the following components:

- A single-instance Redis master to store guestbook entries
- Multiple replicated Redis instances to serve reads
- Multiple web frontend instances
- Objectives
  - Before you begin
  - Start up the Redis Master
  - Start up the Redis Slaves
  - Set up and Expose the Guestbook Frontend
  - Scale the Web Frontend
  - Cleaning up
  - What's next

## Objectives

- Start up a Redis master.
- Start up Redis slaves.
- Start up the guestbook frontend.
- Expose and view the Frontend Service.
- Clean up.

## Before you begin

You need to have a Kubernetes cluster, and the kubectl command-line tool must be configured

# KEPs

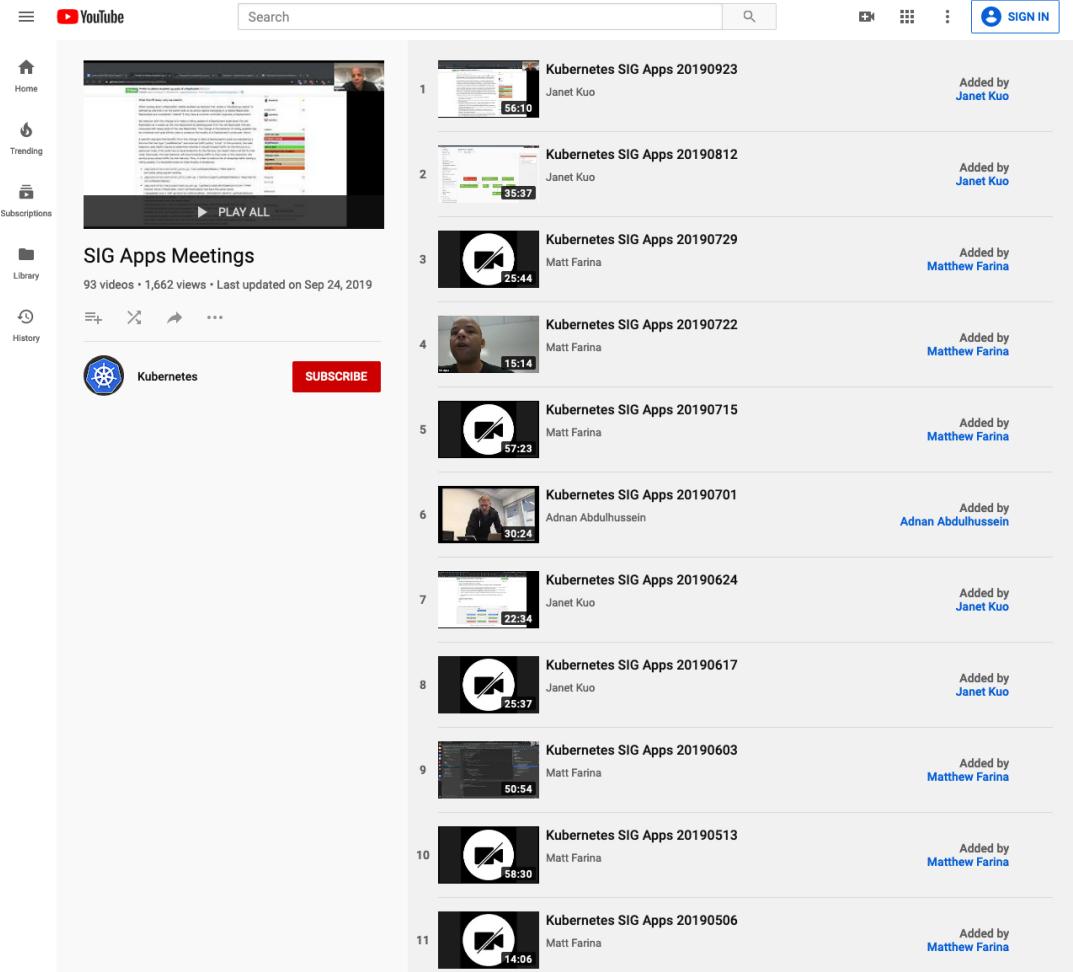
- “A Kubernetes Enhancement Proposal (KEP) is a way to propose, communicate and coordinate on new efforts for the Kubernetes project.”
- More details at  
<https://github.com/kubernetes/enhancements/tree/master/keps>

Author	Labels	Projects	Milestones	Reviews	Assignee	Sort
	11 Open ✓ 23 Closed					
	needs-ok-to-test sig/apps size/XL	cncf-cla: yes kind/kep				2
	#1246 opened on Sep 18 by barney-s					
	lifecycle/stale ok-to-test sig/apps size/L	cncf-cla: yes kind/kep				23
	#1145 opened on Jul 16 by kolorful					
	[WIP] Enhance daemonset on rollingupdate.	cncf-cla: yes do-not-merge/work-in-progress kind/kep sig/apps size/L				26
	#1140 opened on Jul 14 by zhangxiaoju-zidif • Changes requested					
	Supports PreSidecars and PostSidecars	cncf-cla: yes kind/kep sig/apps size/M				5
	#1135 opened on Jul 10 by dixudx					
	needs-ok-to-test sig/apps size/M	cncf-cla: yes kind/kep				17
	#1072 opened on May 27 by fudali113					
	Added a KEP for the Stateful Application Data Management API	cncf-cla: yes kind/kep lifecycle/stale sig/apps sig/storage size/XL				31
	#1051 opened on May 7 by liyinan926					
	KEP for Graduating CronJob to GA	cncf-cla: yes kind/kep lifecycle/stale ok-to-test sig/apps size/L				41
	#978 opened on Apr 20 by barney-s • Changes requested					
	ConfigMap / Secret Orchestration	cncf-cla: yes kind/kep ok-to-test sig/apps size/L				7
	#948 opened on Apr 10 by kfox111					
	Add KEP for application pod graceful update	cncf-cla: yes kind/kep lifecycle/stale ok-to-test sig/apps size/L				20
	#905 opened on Mar 19 by zhan849					
	KEP for adding ProgressDeadlineSeconds to job	cncf-cla: yes kind/feature kind/kep ok-to-test sig/apps size/L				26
	#875 opened on Mar 5 by goodluckbot • Changes requested					
	Add StatefulSet Volume Expansion Kep	api-review approved cncf-cla: yes do-not-merge/hold kind/api-change kind/feature kind/kep lgtm sig/apps sig/storage size/L tide/merge-method-squash				124
	#660 opened on Dec 20, 2018 by SidakM • Changes requested					

# How You Can Get Involved

# Attend A Meeting

- Meetings every other Monday at 9am Pacific Time
- Recordings on YouTube
- More information at <https://github.com/kubernetes/community/tree/master/sig-apps>



# Contribute Documentation



This screenshot shows the "Deployments" page under the "Concepts" section of the Kubernetes documentation. The sidebar lists various Kubernetes concepts, and the main content explains what a Deployment is and how it manages Pods and ReplicaSets. A note at the bottom of the content area states: "Note: Do not manage ReplicaSets owned by a Deployment. Consider opening an issue in the main Kubernetes repository if your use case is not covered below." A red arrow points from this note to the "No controller intro" text below.

This screenshot shows the same "Deployments" page as the previous one, but with a red arrow pointing from the "No controller intro" text to the "Controllers" section in the sidebar, highlighting the lack of introductory content for controllers.

No controller intro

This screenshot shows the "Deployments" page again, with a red arrow pointing from the "No controller intro" text to the "Controllers" section in the sidebar, emphasizing the absence of introductory content for controllers.

Not Recommended Labels

3 ½ year old image version

```
apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2
kind: Deployment
metadata:
  name: frontend
  labels:
    app: guestbook
spec:
  selector:
    matchLabels:
      app: guestbook
      tier: frontend
  replicas: 3
  template:
    metadata:
      labels:
        app: guestbook
        tier: frontend
    spec:
      containers:
        - name: php-redis
          image: gcr.io/google-samples/gb-frontend:v4
          resources:
            requests:
              cpu: 100m
              memory: 100Mi
          env:
            - name: GET_HOSTS_FROM_DNS
              value: dns
              # Using 'GET_HOSTS_FROM_DNS=dns' requires your cluster to
              # provide a dns service. As of Kubernetes 1.3, DNS is a built-in
              # service launched automatically. However, if the cluster you are using
              # does not have a built-in DNS service, you can instead
              # access an environment variable to find the master
              # service's host. To do so, comment out the 'value: dns' line above,
              # uncomment the line below:
              # value: env
          ports:
            - containerPort: 80
```

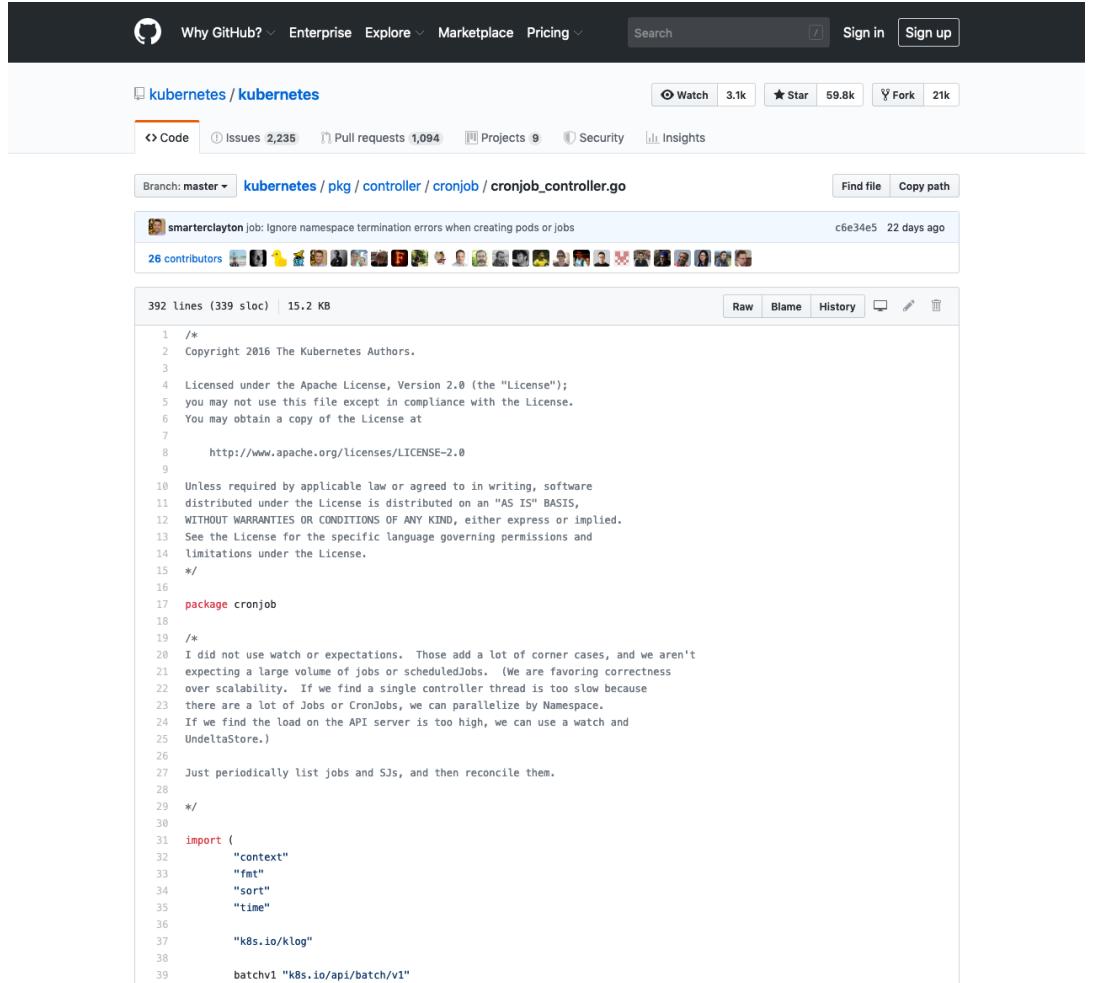
1. Apply the frontend Deployment from the `frontend-deployment.yaml` file:

```
kubectl apply -f https://k8s.io/examples/application/guestbook/frontend-deployment.yaml
```

2. Query the list of Pods to verify that the three frontend replicas are running:

# Code!

- There's a lot of opportunity to code
- Bugs need to be fixed
- CronJob needs a stable release (e.g., needs to switch to shared informers)
- Application CRD/Controller and Execution Hook need work



The screenshot shows a GitHub repository page for the file `kubernetes/pkg/controller/cronjob/cronjob_controller.go`. The page includes navigation links for Why GitHub?, Enterprise, Explore, Marketplace, Pricing, and a search bar. It displays statistics such as 2,235 issues, 1,094 pull requests, 9 projects, and 21k forks. A commit by `smarterclayton` is shown, dated 22 days ago, with a commit message about ignoring namespace termination errors. The code listing shows 392 lines of Go code, starting with a license header and imports for context, fmt, sort, time, k8s.io/klog, and batchv1.

```
/*
Copyright 2016 The Kubernetes Authors.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

*/
package cronjob

/*
I did not use watch or expectations. Those add a lot of corner cases, and we aren't
expecting a large volume of jobs or scheduledJobs. (We are favoring correctness
over scalability. If we find a single controller thread is too slow because
there are a lot of Jobs or CronJobs, we can parallelize by Namespace.
If we find the load on the API server is too high, we can use a watch and
UndeltaStore.)

Just periodically list jobs and SJs, and then reconcile them.

*/
import (
    "context"
    "fmt"
    "sort"
    "time"
    "k8s.io/klog"
    batchv1 "k8s.io/api/batch/v1"
```

# Build More Tools

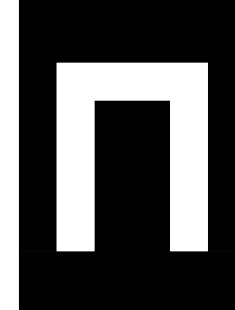


KubeCon



CloudNativeCon

North America 2019



More are needed...

# Questions / Discussion?