So You Want To Run a Data-Intensive System On Kubernetes

@lenadroid

#SeattleScalability

Agenda

- Distributed systems for big data
- Running them on Kubernetes?
- Getting them to “work”
- Challenges
- Work in progress and solutions
- Future
- Summary


cassandra


kafka


Spark


elastic


redis

Data systems on Kubernetes

@lenadroid

#SeattleScalability

How do we run systems like Kafka, Cassandra, Spark, etc. on Kubernetes?

@lenadroid

#SeattleScalability

Interpretation #1

What Kubernetes abstractions can we use to run these distributed systems?

Interpretation #2

What are the steps to simply get those distributed systems up and running based on these Kubernetes abstractions?

Interpretation #3

How to get them to run and work according to what is considered correct behavior for those systems?

Answer #1:

Kubernetes abstractions

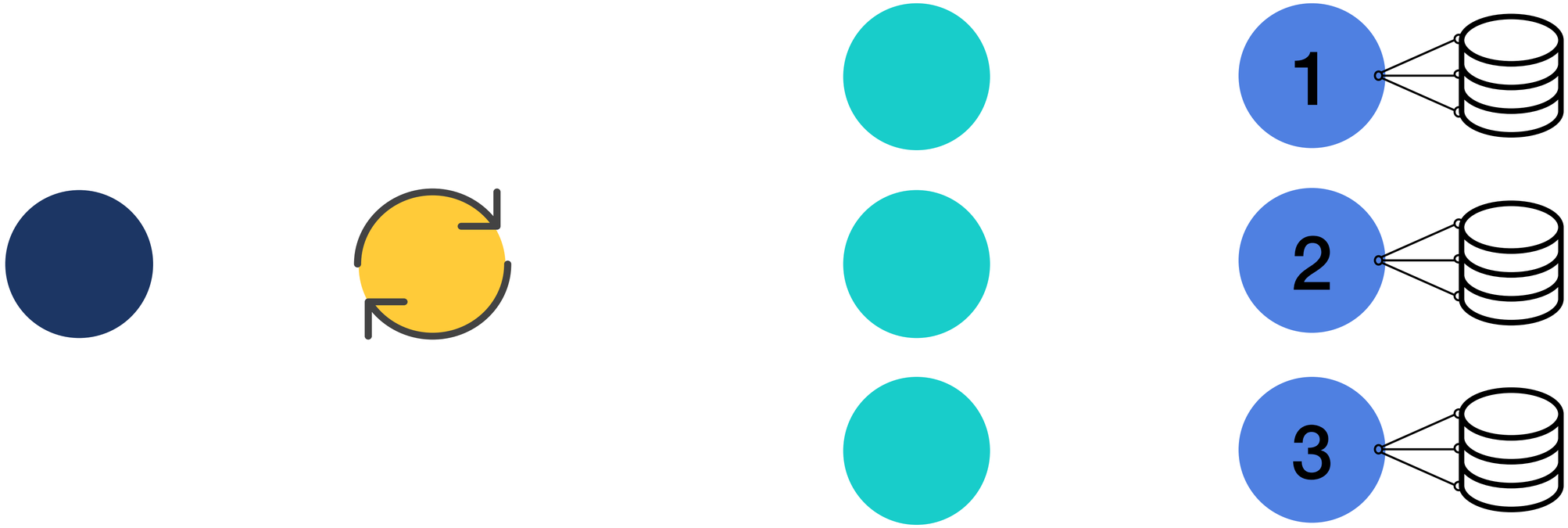
@lenadroid

Pod

Job

Replica Set

Stateful Set



And many more…

Answer #2:

Getting it up and running using built-in K8S abstractions

@lenadroid

@lenadroid

Yaml files

Example:

Getting Cassandra up and running with Stateful Sets and YAMLs on AKS

cassandra-statefulset.yaml

```yaml
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: cassandra
  labels:
    app: cassandra
spec:
  serviceName: cassandra
  replicas: 5
  selector:
    matchLabels:
      app: cassandra
  template:
    metadata:
      labels:
        app: cassandra
    spec:
      terminationGracePeriodSeconds:
      1800
      containers:
      - name: cassandra
        image:
        gcr.io/google-samples/cassand
        ra:v13
        imagePullPolicy: Always
        ports:
        - containerPort: 7000
          name: intra-node
        - containerPort: 7001
          name: tls-intra-node
        - containerPort: 7199
          name: jmx
        - containerPort: 9042
          name: cql
        resources:
```

cassandra-statefulset.yaml

```yaml
        resources:
          limits:
            cpu: "2"
            memory: 3.5Gi
          requests:
            cpu: "2"
            memory: 3.5Gi
        securityContext:
          capabilities:
            add:
              - IPC_LOCK
        lifecycle:
          preStop:
            exec:
              command:
              - /bin/sh
              - -c
              - nodetool drain
        env:
          - name: MAX_HEAP_SIZE
            value: 512M
          - name: HEAP_NEWSIZE
            value: 100M
          - name: CASSANDRA_SEEDS
            value:
            "cassandra-0.cassandra.
            default.svc.cluster.
            local"
          - name:
          CASSANDRA_CLUSTER_NAME
            value: "chicago"
          - name: CASSANDRA_DC
            value: "DC1-chicago"
          - name: CASSANDRA_RACK
            value: "Rack1-chicago"
```

cassandra-statefulset.yaml

```yaml
          - name: POD_IP
            valueFrom:
              fieldRef:
                fieldPath:
                status.podIP
        readinessProbe:
          exec:
            command:
            - /bin/bash
            - -c
            - /ready-probe.sh
          initialDelaySeconds: 15
          timeoutSeconds: 5
        volumeMounts:
        - name: cassandra-data
          mountPath: /cassandra_data
  volumeClaimTemplates:
  - metadata:
      name: cassandra-data
    spec:
      accessModes: [
      "ReadWriteOnce" ]
      storageClassName:
      managed-premium
      resources:
        requests:
          storage: 1Gi
```

```
repeat while (true) {
```

UP AND RUNNING

!=

OPERATING CORRECTLY

```
}
```

@lenadroid

#SeattleScalability

HELM

Get Helm Blog Docs

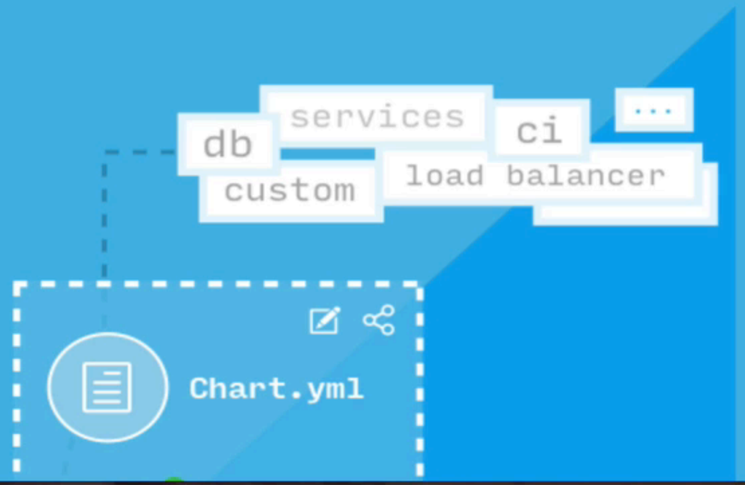
The package manager for Kubernetes

Helm is the best way to find, share, and use software built for Kubernetes.

What is Helm?

Helm helps you manage Kubernetes applications — Helm Charts helps you define, install, and upgrade even the most complex Kubernetes application.

Charts are easy to create, version, share, and publish — so start using Helm and stop the copy-and-paste madness.



Helm

- Charts and Chart.yaml
- Values and Values.yaml
- Chart repositories
- Tiller, Releases and Rollbacks

kubernetes / charts

Watch 164 | Star 3,390 | Fork 2,899

Code | Issues 397 | Pull requests 280 | Projects 0 | Wiki | Insights

Branch: master ▾ charts / incubator / kafka / Create new file | Upload files | Find file | History

Dean-Coakley and **k8s-ci-robot** [incubator/kafka] Fix heapOptions template reference (#5916) ··· Latest commit 3ba5451 a day ago

..		
templates	[incubator/kafka] Fix heapOptions template reference (#5916)	a day ago
.helmignore	adds chart for kafka (#144)	a year ago
Chart.yaml	[incubator/kafka] Fix heapOptions template reference (#5916)	a day ago
OWNERS	[incubator/kafka] Allow configurationOverride of zookeeper.connect (#...	3 months ago
README.md	Add "helm repo add incubator" to README (#5611)	20 days ago
requirements.lock	Update zookeeper dependency to latest chart (#3916)	3 months ago
requirements.yaml	Update zookeeper dependency to latest chart (#3916)	3 months ago
values.yaml	Issue_5426: Add default antiAffinity example to Kafka (#5428)	27 days ago

README.md

Apache Kafka Helm Chart

Chart Details

This chart will do the following:

- Implement a dynamically scalable kafka cluster using Kubernetes StatefulSets
- Implement a dynamically scalable zookeeper cluster as another Kubernetes StatefulSet required for the Kafka cluster above
- Expose Kafka protocol endpoints via NodePort services (optional)

Other examples of deployment

Example: Spark on Kubernetes (AKS)

Is this really enough?

Answer #3:

Making sure our systems operate according to what is considered correct behavior for those systems

Things that need special care

Kubernetes is smart.

P.S. When we tell it how to treat our systems.

Kafka

Configuration

Rolling cluster restarts and upgrades

Scaling cluster up and down

Data rebalancing

Kafka configuration

Pod specific vs general configurations

Broker ID and rack assignments

Kafka cluster restarts/upgrades

No under-replicated partitions

Restart one broker at a time

Always wait for restarted broker to catch up to a leader

Scaling Kafka cluster

Partition assignment

Data rebalancing

Managing configuration


OPERATORS
@lenadroid

Operators

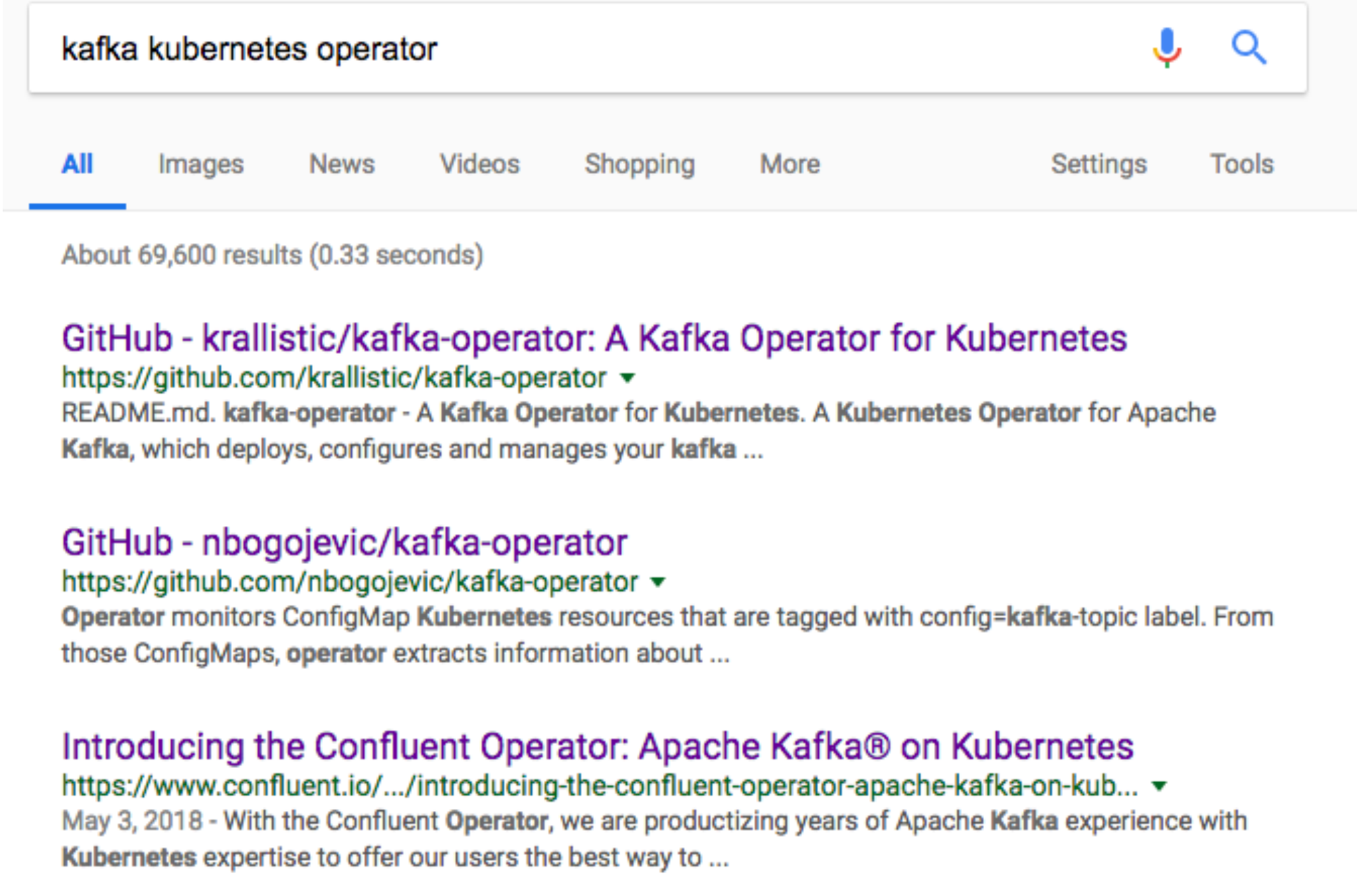
Custom Resource Definitions

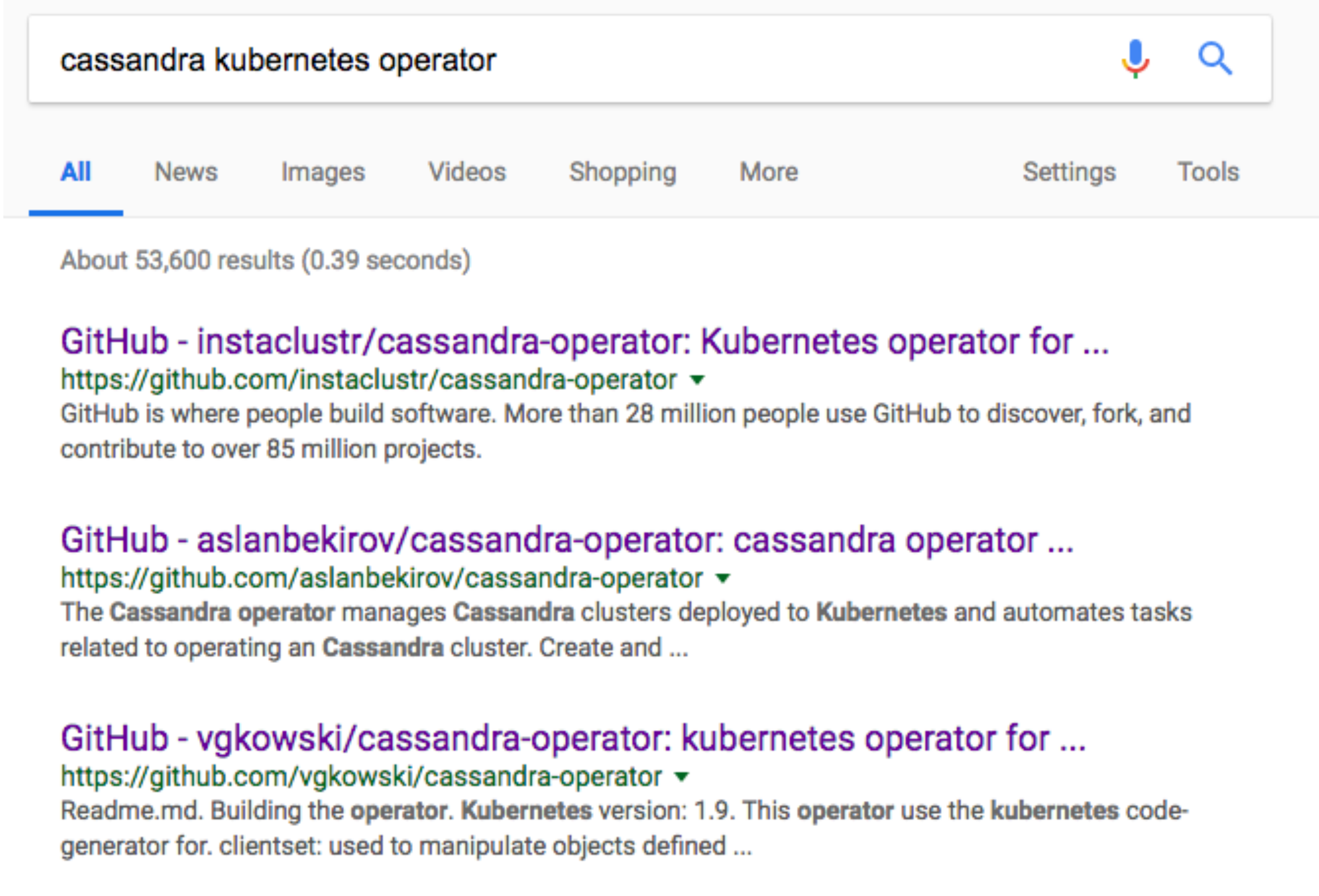
+

Custom Controllers

=

More Control


kafka kubernetes operator
All
Images
News
Videos
Shopping
More
Settings
Tools
About 69,600 results (0.33 seconds)
GitHub - krallistic/kafka-operator: A Kafka Operator for Kubernetes
https://github.com/krallistic/kafka-operator
README.md. kafka-operator - A Kafka Operator for Kubernetes. A Kubernetes Operator for Apache Kafka, which deploys, configures and manages your kafka ...
GitHub - nbogojevic/kafka-operator
https://github.com/nbogojevic/kafka-operator
Operator monitors ConfigMap Kubernetes resources that are tagged with config=kafka-topic label. From those ConfigMaps, operator extracts information about ...
Introducing the Confluent Operator: Apache Kafka® on Kubernetes
https://www.confluent.io/.../introducing-the-confluent-operator-apache-kafka-on-kub...
May 3, 2018 - With the Confluent Operator, we are productizing years of Apache Kafka experience with Kubernetes expertise to offer our users the best way to ...


cassandra kubernetes operator
All
News
Images
Videos
Shopping
More
Settings
Tools
About 53,600 results (0.39 seconds)
GitHub - instaclustr/cassandra-operator: Kubernetes operator for ...
https://github.com/instaclustr/cassandra-operator
GitHub is where people build software. More than 28 million people use GitHub to discover, fork, and contribute to over 85 million projects.
GitHub - aslanbekirov/cassandra-operator: cassandra operator ...
https://github.com/aslanbekirov/cassandra-operator
The Cassandra operator manages Cassandra clusters deployed to Kubernetes and automates tasks related to operating an Cassandra cluster. Create and ...
GitHub - vgkowski/cassandra-operator: kubernetes operator for ...
https://github.com/vgkowski/cassandra-operator
Readme.md. Building the operator. Kubernetes version: 1.9. This operator use the kubernetes code-generator for. clientset: used to manipulate objects defined ...

Spark (>=2.3)

Automatic job resubmission after specification update

Configurable restart policy

Automatic retries on failed submissions

Mounting specific ConfigMaps and volumes

Example: Spark Operator on Kubernetes (AKS)

Example: Redis Operator on Kubernetes (AKS)

Anatomy of an Operator

CRD

Resource Structs/Types

Informer

Controller

Queue

TensorFlow Operator

CRD

```yaml
apiVersion: apiextensions.k8s.io/v1beta1
kind: CustomResourceDefinition
metadata:
  name: tfjobs.kubeflow.org
spec:
  group: kubeflow.org
  version: v1alpha1
  names:
    kind: TFJob
    singular: tfjob
    plural: tfjobs
```

TFJOB

```go
// TFJob represents the configuration of signal TFJob
type TFJob struct {
        metav1.TypeMeta `json:",inline"`

        // Standard object's metadata.
        metav1.ObjectMeta `json:"metadata,omitempty"`

        // Specification of the desired behavior of the TFJob.
        Spec TFJobSpec `json:"spec,omitempty"`

        // Most recently observed status of the TFJob.
        // This data may not be up to date.
        // Populated by the system.
        // Read-only.
        Status TFJobStatus `json:"status,omitempty"`
}
```

Takeaways

Built-in Kubernetes abstractions don’t solve all the issues

Tools like Helm really help structure and manage deployments

Operators = Custom Controllers + CRDs & …

…. a way to teach Kubernetes understand our needs

There’re many ways to deploy and use Operators…

Thank You!

My blog - **lenadroid.github.io/posts.html**

My talk on Stateful Sets, Persistent Volumes, Persistent Volime Claims, Storage Classes - **aka.ms/gotochgo**

Thomas Stringer's post on Custom Controllers – **aka.ms/custom-controllers**

Alena Hall - lenadroid

- ✓ Works on Azure at Microsoft
- ✓ Lives in Seattle
- ✓ F# Software Foundation Board of Trustees
- ✓ Organizes @ML4ALL
- ✓ Program Committee for Lambda World
- ✓ Has a channel: YouTube /c/AlenaHall