Kubernetes APIs Under the Hood
Who Am I?

Phillip Wittrock (@pwittrock)

Software Engineer at Google working on GKE and OSS Kubernetes

My mission is to make using Kubernetes simple and enjoyable

You might have come across me through...

- Kubectl
- Kubebuilder
- Kubernetes Steering Committee
Kubernetes Refresher

- **Nodes** are machines in a cluster that run Containers in **Pods**
- **Pods** are created and managed by higher level abstractions such as **ReplicaSets**
- **ReplicaSets** managed by higher level abstractions such as **Deployments**
- **Deployments** (and all other user owned objects) defined in files and created / updated with `kubectl apply`
- **APIs** (deployments, replicaset, pods, nodes) == Resource Types and Objects == Resources

@pwittrock
Kubernetes APIs Are...

Declarative, Asynchronous, Level-Triggered, Observable, Discoverable, Versioned, Access Controlled, Extensible, ...
Kubernetes APIs are… Declarative

Configuration for a **Deployment** that manages 3 **Pods** each running an nginx container.

Deployment resource is declared in a file.

Create or update the resource in the cluster by run `kubectl apply` on a file or directory.

```
kubectl apply -f deploy.yaml
```
Create Deployment Example

1. **Discover** EndPoints
2. HTTP POST
3. ???

---

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels: {app: nginx}
spec:
  replicas: 3
  selector:
    matchLabels: {app: nginx}
template:
  metadata:
    labels: {app: nginx}
  spec:
    containers: [
      {name: nginx,
       image: 'nginx:1.7.9'}
    ]
```

*deploy.yaml*

- kubectl apply -f deploy.yaml

---

@pwittrock
Lifecycle - Resources, Controllers and Webhooks

- APIs **declared** as **Resources** - provide storage and endpoints
- APIs **actuated** by **Controllers** - execute the business logic
- APIs **admitted** by **Webhooks** - defaulting, validation, conversion
Kubernetes APIs are... Asynchronous & Observable

Resource:
Stores Stuff

Controller:
Does Stuff

Loose coupling between Controller and API endpoints:
Storage doesn’t know about Controllers

Foo Resources stored in etcd by the apiserver

Asynchronous watch notification on object create / update / delete
Kubernetes APIs are... Level Triggered

**Object**

- **default/baz**

  Watch events:

  - Create
  - Update

**Controller**

- **Reconcile Queue**
  - default / baz

- **Reconcile Function**

**Tips:**

- Internal cleanup with ownerReferences
- External cleanup with finalizers

Batch events together into single Reconcile call

Reconcile on namespace/name only, not the event

@pwittrock
Controller Workflow

- apiserver streams **Watch Event** to Controller
- Controller **Reads Object + Related Objects** (e.g. Deployment + ReplicaSets)
- Controller **Creates new owned objects, updates owned objects, updates object status**
Kubectl Apply: Create Deployment

- **deploy. yaml**
- **Deployment Controller**
- **Node(s)**
- **Scheduler**
- **apiServer**
- **create ReplicaSet**
- **create Pod(s)**
- **watch evt**
- **update Pod(s)**

- **apply**
- **create**
Kubectl Apply: Update Deployment

- apply deploy.yaml
- patch
- update Pod(s)
- Node(s)
- update Pod(s)
- Deployment Controller
  - watch evt
  - Scale up new ReplicaSet / Scale down old ReplicaSet
- ReplicaSet Controller
  - create Pod(s)
  - delete Pod(s)
- (Pod) Scheduler

@pwittrock
Resources

- Resource Types **declare APIs**
- Resources provide **storage for objects**
- Standardized schema structure
- Discoverable API endpoints and schema
- “Just work” with declarative tooling - e.g. `kubectl apply`

```
apiVersion: v1
group: some-project
kind: Pod
metadata:
  name: my-app
  namespace: default
spec:
  containers:
    - name: client
      image: gcr.io/some-project/udptest
      imagePullPolicy: Always
      args: [sh]
  dnsPolicy: ClusterFirst
status:
  podIP: 10.8.3.11
```
**TypeMeta**

- **Kind (Deployment)**
  - Name of the API (e.g. Deployment)
- **Group (apps)**
  - Like a package in go, java, etc (e.g. apps)
- **Version (v1)**
  - Ensures backwards compatibility of: Defaulted Fields & Schema

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels: {app: nginx}
spec:
  replicas: 3
  selector:
    ... 
  template:
    ... 
```
ObjectMeta

- **Name** and **Namespace** uniquely identify an object for a given Resource
- **Annotations** are arbitrary key-value pairs that cannot be queried
- **Labels** are key-value pairs that may be queried (selected)

```yaml
apiVersion: apps/v1
class: Deployment
metadata:
  name: nginx-deployment
  namespace: default
labels: {app: nginx}
spec:
  replicas: 3
  selector:
  template:
```
Spec and Status

- **Spec**
  - *Object Desired State* (e.g. how many replicas to run, template for Pods, etc)

- **Status (not shown)**
  - Defines the *observed* state for an object (e.g. how many replicas are running)

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels: {app: nginx}
spec:
  replicas: 3
  selector:
    matchLabels: {...}
template:
  metadata:
    labels: {...}
  spec:
    ...
Resource Wiring

- **Labels/Selectors** locate objects
- **Label** - generated objects
- **Selector** - find labeled objects
- **OwnerReference** on generated objects

Tip: Objects with owner references are automatically garbage collected when all of their owners have been deleted.

Deployment
- name: nginx
- labels: run=nginx
- selector: run=nginx

ReplicaSet
- name: nginx-65899c769f
- labels: run=nginx
- selector: run=nginx
- owner: Deployment nginx

Pod
- name: nginx-65899c769f-6slpx
- labels: run=nginx
- owner: ReplicaSet nginx-65899c769f

Pod
- name: nginx-65899c769f-fbgcv
- labels: run=nginx
- owner: ReplicaSet nginx-65899c769f

@pwittrock
Synchronous Defaulting and Validation

- Unspecified optional fields may be defaulted by the apiserver before the object is stored
- Simple Schema validation performed through OpenAPI
- Complex validation performed by the apiserver before the object is stored

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels: {app: nginx}

spec:
  # server defaults this value
  # to 1 if unset
  replicas: 1

  # make sure these match the
  # template labels
  selector:
    matchLabels: {...}

  template:
    metadata:
      labels: {...}
```

@pwittrock
Kubernetes APIs are... Extensible

Mutating Webhook + Service + Deployment (or Pod)

Foo Admission

Defaulting, Validation, Version Conversion

CustomResourceDefinition (CRD)

Foo Resource

Storage, Schema, Display, etc

Deployment (or Pod)

Foo Controller

Actuation: level-triggered, asynchronous

Tip: build your own APIs in go using kubebuilder
Updating Resources

Gotchas

- Spec has shared ownership across multiple parts of the system
- Controllers or other actors may update the Spec with new fields which must be retained across updates to the object
- Both an issue for Controllers and for users managing Resources using config
- Need to either read-update-write or apply

```yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx
  labels: {app: nginx}
spec:
  selector:
    app: nginx
  ports:
  - protocol: TCP
    port: 80
# not set by owner!
# don’t overwrite!
  clusterIp: 10.0.171.239
# not set by owner!
# don’t overwrite!
  loadBalancerIp: 78.1124.19
  type: LoadBalancer
```
Observing Objects - Status and Events

- Actuation performed asynchronously
- Status published to users, tools and other controllers through Status field
- Conditions: key/value pairs that communicate status (current) to other tools (part of Status field)
- Events: separate objects that communicate past events to users
Converting API versions

- Different versions of an API may have different representations
  - Changing default values and field names / field types requires a new version
- All versions of the same API are logically equivalent
- The same object may be read or written in any version -- the underlying object remains the same -- but the endpoints are different.
Classes of APIs

Composites

Operators
Spark, Airflow

Cloud Native Abstractions
Tekton, Knative

Decorators

Autoscalers, Resource Tuners

@pwittrock
Kubernetes APIs Are...

Declarative, Asynchronous, Level-Triggered, Observable, Discoverable, Versioned, Access Controlled, Extensible, ...
Kubebuilder Workshop

Rate today’s session
Questions?