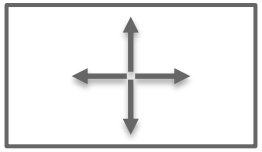
Extending Kubernetes with CRDs and Controllers/Operators

Chris Hein | Developer Advocate | AWS | CNCF Ambassador
heichris@amazon.com
@christopherhein
christopherhein

What is Kubernetes?

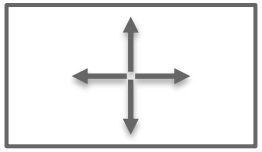
Open source container management platform



Helps you run containers at scale

Gives you primitives for building modern applications

Open source ~~container~~ management platform



Helps you run containers at scale

Gives you primitives for building modern applications

What does all of that mean

Let’s see it

```
$ kubectl get pods -v=7
GET https://apiserver/api/v1/namespaces/default/pods?limit=500
Request Headers:
    Accept: application/json;as=Table;v=v1beta1;g=meta.k8s.io,
application/json
    User-Agent: kubectl/v1.12.1 (linux/amd64) kubernetes/4ed3216
Response Status: 200 OK in 145 milliseconds
Response Headers:
    Content-Type: application/json
    Content-Length: 1909
    Date: Wed, 28 Nov 2018 00:23:05 GMT
    Audit-Id: 7b949a88-f3d2-429d-9b19-889c01f2c634
Response Body: ...
```

API Layer

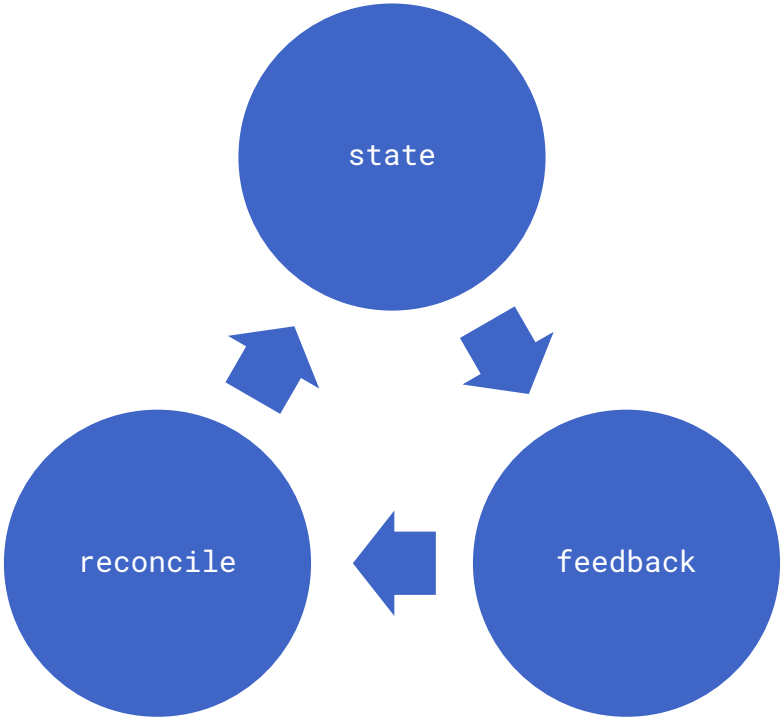
```
{
  "paths": [
    "/api",
    "/api/v1",
    "/apis",
    "/apis/",
    "/apis/admissionregistration.k8s.io",
    "/apis/admissionregistration.k8s.io/v1beta1",
    ...
  ]
}
```

Kubernetes objects

- **Record of intent**
- Persistent entities
- Marshal the desired state of your cluster
- Examples: Pods, Services, Ingress, NetworkPolicies, ConfigMaps, Secrets, etc.

How do they work?

Control Loop



What do we get out of this complexity

Immutable

{kind}

Declarative

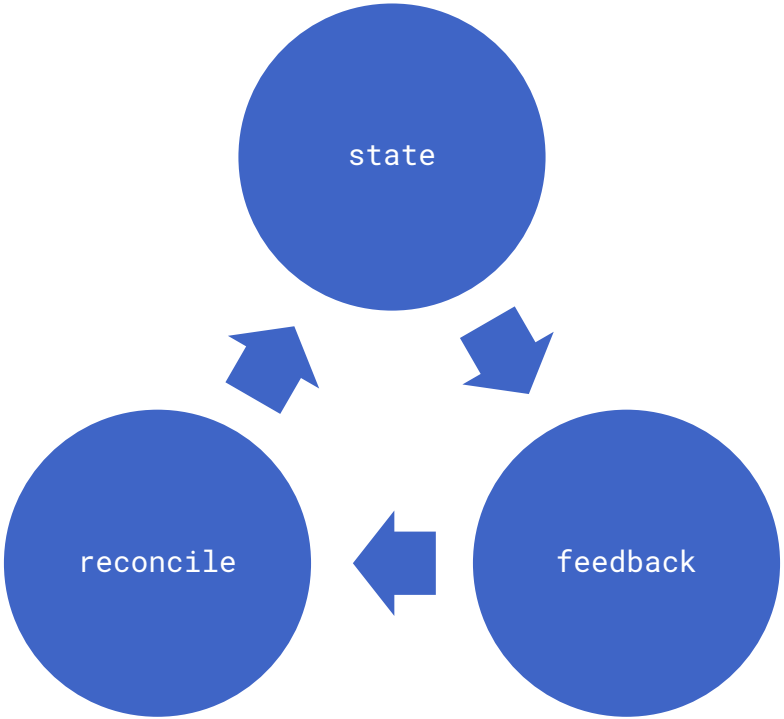
Self-healing

How does it connect?

Controllers

attachdetach, bootstrapsigner, clusterrole-aggregation, cronjob, csrapproving, csrcleaner, csrsigning, daemonset, deployment, disruption, endpoint, garbagecollector, job, namespace, horizontalpodautoscaling, nodeipam, nodelifecycle, persistentvolume-binder, persistentvolume-expander, podgc, pv-protection, pvc-protection, replicaset, replicationcontroller, resourcequota, route, service, serviceaccount, serviceaccount-token, statefulset, tokencleaner, ttl, ttl-after-finished bootstrapsigner, tokencleaner

Control Loop



How do you extend it?

Extending Kubernetes

- Built to be highly configurable
- Uses the **control loop** pattern
- Extensions incl. Storage, Device, Network plugins, Aggregate apiservers, CRDs and more
- Awesome Operators list - **https://git.io/vh6Qm**

CustomResourceDefinitions (CRDs)

CustomResourceDefinitions (CRDs)

- Dynamically creates RESTful routes in the apiserver
- Defined with OR without a Pod to use them
- Uses YAML to tell Kubernetes to create them
- Built-in Mechanisms for .Status and sub-resources
- Ability to validate at the apiserver

CRD YAML

```yaml
apiVersion: apiextensions.k8s.io/v1beta1
kind: CustomResourceDefinition
metadata:
  name: crontabs.stable.example.com
spec:
  group: stable.example.com
  version: v1
  scope: Namespaced
  names:
    plural: crontabs
    singular: crontab
    kind: CronTab
    shortNames:
    - ct
```

Standard K8s primitives style

Defined Custom Resource

```yaml
apiVersion: "stable.example.com/v1"
kind: CronTab
metadata:
  name: my-new-cron-object
spec:
  cronSpec: "* * * * */5"
  image: my-awesome-cron-image
```

Again…

Standard K8s primitives style

What do you do with CRDs?

Build Operators

“Operators are domain specific”

- Kris Nova

Operators

- Coined by CoreOS
- Standalone Pod in a Kubernetes cluster
- Watches kube-apiserver for updates
- Can watch any Kubernetes resource
- Allows you to take action on new or updates to existing resources changes

Simple use case

- **https://git.io/fjOZ3**
- Uses client-go and exposes a CRD for managing authorized_key files on your nodes
- Deploy the operator + CRDs then deploy an AuthorizedKey Resource for each employee

!!! DEMO USE ONLY !!!

Demo

Code Generation FTW! 🎉

apis/node/types.go

```
// +genclient
// +genclient:noStatus
// +genclient:nonNamespaced
// +k8s:deepcopy-gen:interfaces=k8s.io/apimachinery/pkg/runtime.Object
type AuthorizedKey struct {
     metav1.TypeMeta   `json:",inline"`
     metav1.ObjectMeta `json:"metadata"`
     Data              AuthorizedKeyData `json:"data"`
}

type AuthorizedKeyData struct {
     Key               string `json:"key"`
}
```

hack/update-codegen.sh

```
scriptdir="$( cd "$( dirname "${BASH_SOURCE[0]}" )" && pwd )"
projectdir="$(pwd | sed "s#$GOPATH\/src\/##g")"
cd ${scriptdir}/vendor/k8s.io/code-generator && ./generate-groups.sh \
  all \
  ${projectdir}/generated \
  ${projectdir}/apis \
  "node.chrishein.com:v1alpha1" \
```

generated/...

```
generated/
├── clientset
|   └── versioned
|       ├── clientset.go
|       ├── doc.go
|       ├── fake
|       |   ├── clientset_generated.go
|       |   ├── doc.go
|       |   └── register.go
|       ├── scheme
|       |   ├── doc.go
|       |   └── register.go
|       └── typed
...
```

main.go

```
cfg, _ := clientcmd.BuildConfigFromFlags(masterURL, kubeconfig)
nodeClient, _ := clientset.NewForConfig(cfg)

clientInformerFactory := informers.NewSharedInformerFactory(nodeClient,
time.Minute*30)

controller := ctrl.New(nodeClient,
clientInformerFactory.Node().V1alpha1().AuthorizedKeys())

clientInformerFactory.Start(stopCh)

if err = controller.Run(1, stopCh); err != nil {
    klog.Fatalf("error running controller error=%s", err.Error())
}
```

controller/controller.go#New

```
authKeyInformer.Informer().AddEventHandler(cache.ResourceEventHandlerFuncs
{
    AddFunc: controller.enqueueAuthKey,
    UpdateFunc: func(old, new interface{}) {
        newKey := new.(*nodev1alpha1.AuthorizedKey)
        oldKey := old.(*nodev1alpha1.AuthorizedKey)
        if newKey.ResourceVersion == oldKey.ResourceVersion {
            return
        }
        controller.enqueueAuthKey(new)
    },
    DeleteFunc: controller.deleteAuthKey,
})
```

controller/controller.go#syncHander

```
_, name, _ := cache.SplitMetaNamespaceKey(key)

authKey, _ := c.authKeyLister.Get(name)

authorizedKeyFile = authorizedkey.File{
    UID: authKey.Name,
    Key: authKey.Data.Key,
}

err = authorizedKeyFile.Sync(false)
return nil
```

Other use cases


aws service
operator


kubernetes
OPERATOR
etcd


kubernetes
OPERATOR
Prometheus

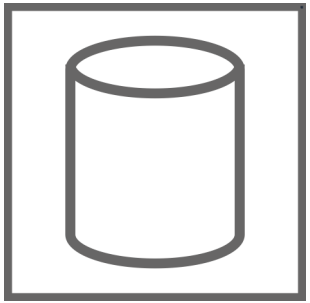
Demo

AWS Service Operator

https://git.io/fhWu3

Others…

storage solution



databases

out of cluster tools

Thanks!

Chris Hein | Developer Advocate | AWS | CNCF Ambassador
heichris@amazon.com
@christopherhein
christopherhein

kubernetes