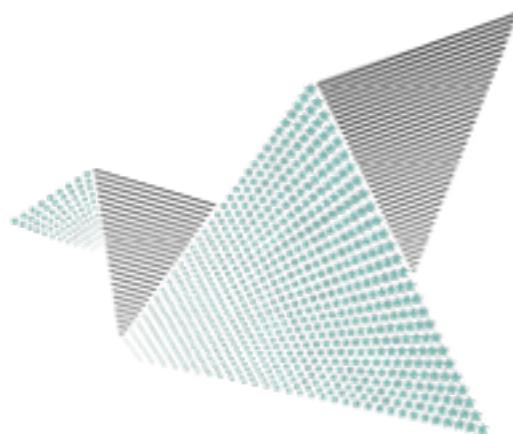


Kubernetes volume plugins

Flocker

@agonzalezro





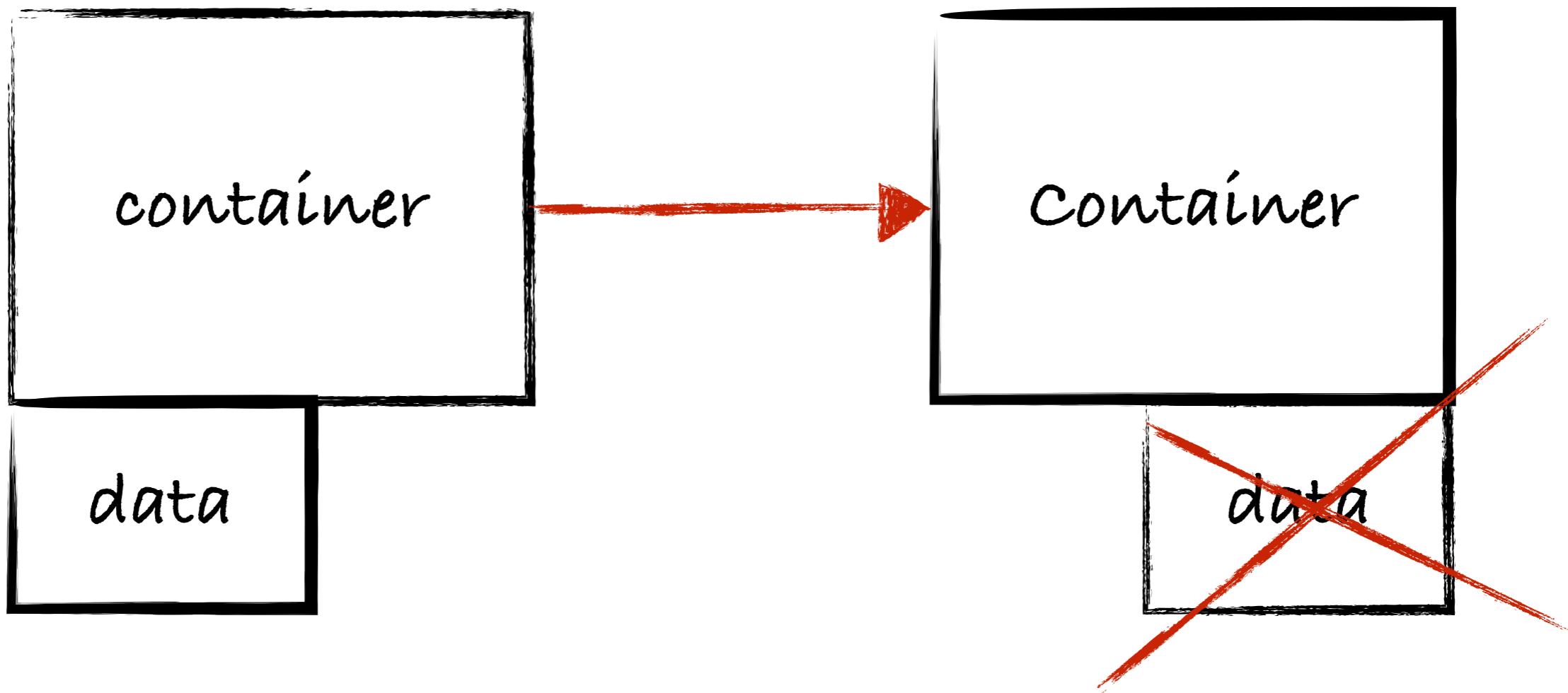
Flocker™
by ClusterHQ™

JETSTACK®

TOC

1. Introduction to Flocker
2. Introduction to Kubernetes
3. Introduction to Flocker plugin
4. Code
5. Summary

What Flocker offers?



What else does it offer?

- AWS EBSRackspace Cloud Block Storage
- Anything that supports the OpenStack Cinder API
- EMC ScaleIO
- EMC XtremIO
- Local storage using our ZFS driver (currently Experimental)

Flocker Docker plugin

HTTP Rest API with twisted

Docker plugins

/VolumeDriver.Create

/VolumeDriver.Mount

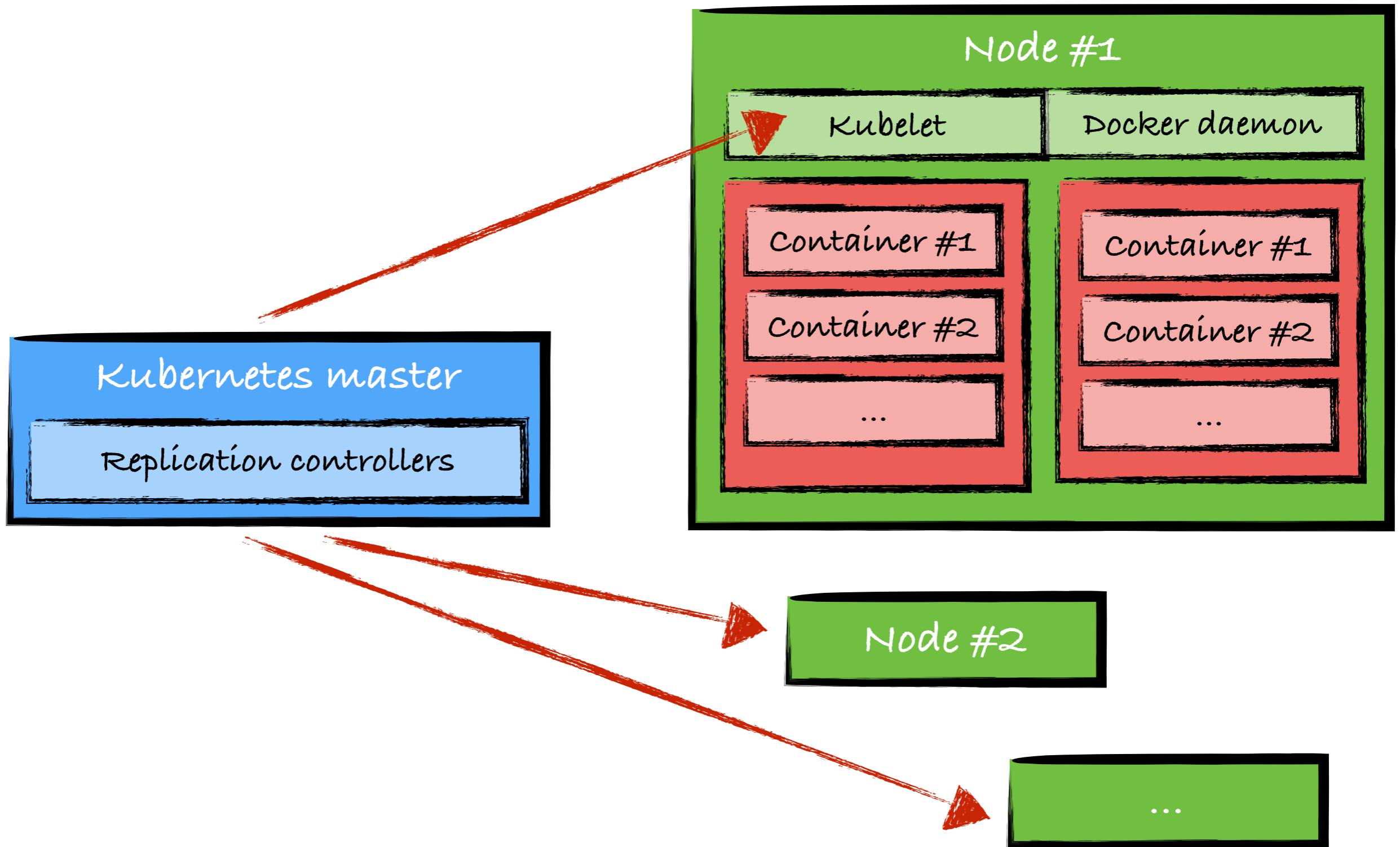
/VolumeDriver.Path

/VolumeDriver.Unmount

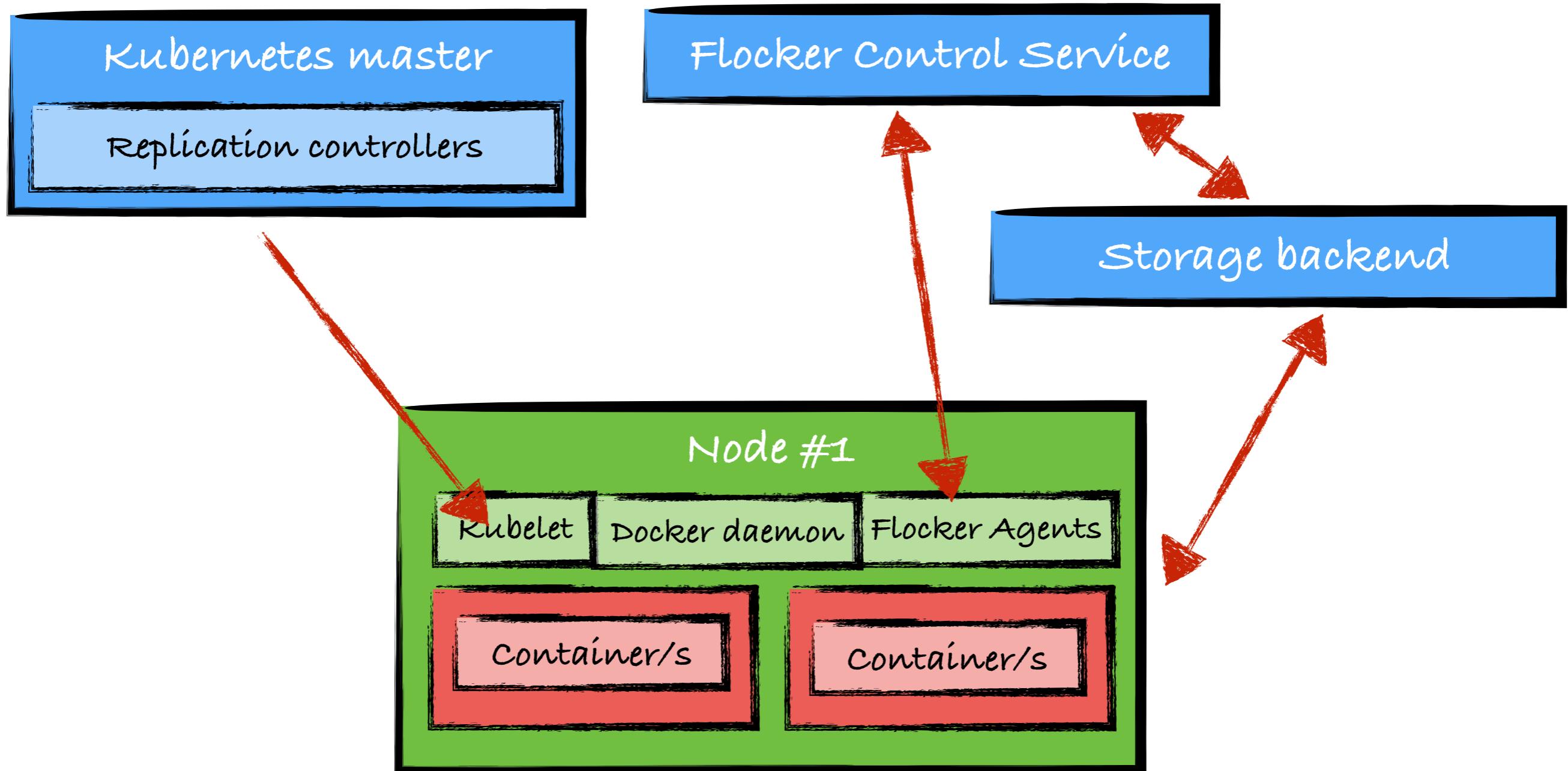
/VolumeDriver.Remove



k8s architecture



How does Flocker fit in?



How to use it (1/2):

Create volume with the flocker-cli:

```
$ uft-flocker-volumes \  
  --control-service=172.16.255.250  
  create --node=43a06d55 \  
  -m name=my-flocker-vol -s 10G
```

```
$ uft-flocker-volumes \  
  --control-service=172.16.255.250  
  list-nodes
```

How to use it (2/2):

```
spec:  
  containers:  
    - name: web  
      image: nginx  
      ports:  
        - name: web  
          containerPort: 80  
  volumeMounts:  
    - name: www-root  
      mountPath: "/usr/share/nginx/html"  
  volumes:  
    - name: www-root  
  flocker:  
    datasetName: my-flocker-vol
```



Before going to code!

Volume vs PersistentVolume

Examples: gcePersistentDisk,
awsElasticBlockStore, gitRepo
(cool example!), etc...

How was it made?

bit.ly/k8smeetup

API: pkg/api/{,v1/}types.go

```
type VolumeSource struct {
    Flocker *FlockerVolumeSource `json:"flocker,omitempty"`
    ...
}

type PersistentVolumeSource struct {
    Flocker *FlockerVolumeSource `json:"flocker,omitempty"`
    ...
}

type FlockerVolumeSource struct {
    // Required: the volume name. This is going to be stored
    // on metadata -> name on the payload for Flocker
    DatasetName string `json:"datasetName"`
}
```

VolumePlugin interface

```
type VolumePlugin interface {
    Init(host VolumeHost)
    Name() string
    CanSupport(spec *Spec) bool
    NewBuilder(
        s *Spec, pod *api.Pod, opts VolumeOptions
    ) (Builder, error)
    NewCleaner(
        name string, podUID types.UID
    ) (Cleaner, error)
}
```

<https://github.com/kubernetes/kubernetes/blob/master/pkg/volume/plugins.go#L55-L81>

How was it used for Flocker?

NewBuilder

just creates the struct that will
allow us to setup the datasets

NewCleaner

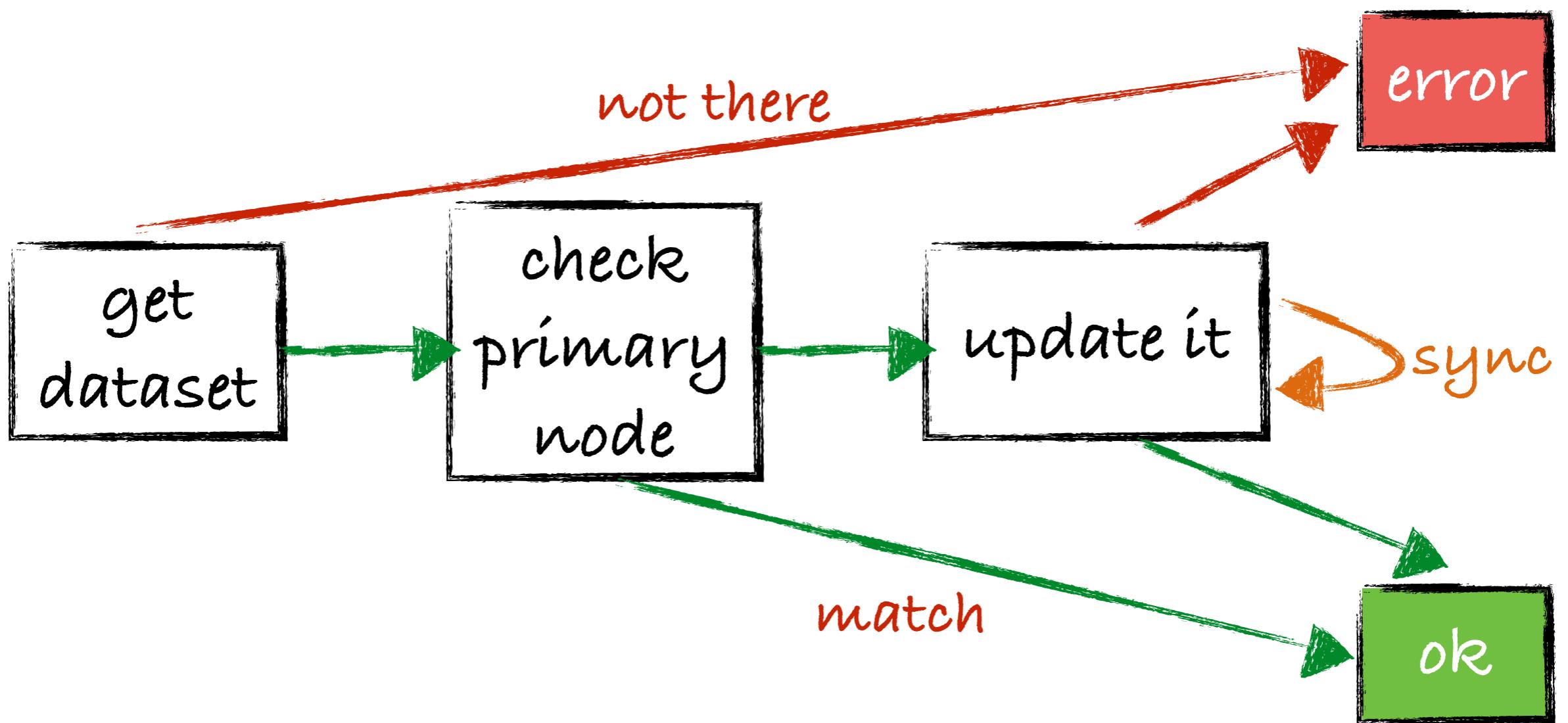
doing nothing

Builder interface

```
type Builder interface {
    Volume // gives GetPath()
    SetUp() error
    SetUpAt(dir string) error
    IsReadOnly() bool
}
```

How was it used for Flocker?

Setup -> Calls SetUpAt the datasetName received by API





Things that didn't go well

- OMG, CLAs! Sorry Matt!
- Shippable and Jenkins
- Creating the volume on SetUp
- hack/ scripts

Future!

- 1.2 has:
`CreatableVolumePlugin`
- Run Flocker in a Pod
- Probably use `VolumeConfig`

What did we talk about?

1. Flocker & Kubernetes
2. Plugin for Flocker
3. Something you have seen in the code

Thanks!

@agonzalezro