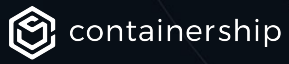
Kubernetes Manages More Than Containers

What are CCMs and how do you build one?

Ashley Schuett



About Me

What is a CCM?

Running a CCM

Features of a CCM

Creating your own

Questions?

01

About Me

What is a CCM?

Running a CCM

Features of a CCM

Creating your own

Questions?

About me

- Engineer at **Containership**
- Cluster lifecycle management:

  provisioning, plugins, upgrades, ...
- Controllers and anything running in-cluster

02

About Me

What is a CCM?

Running a CCM

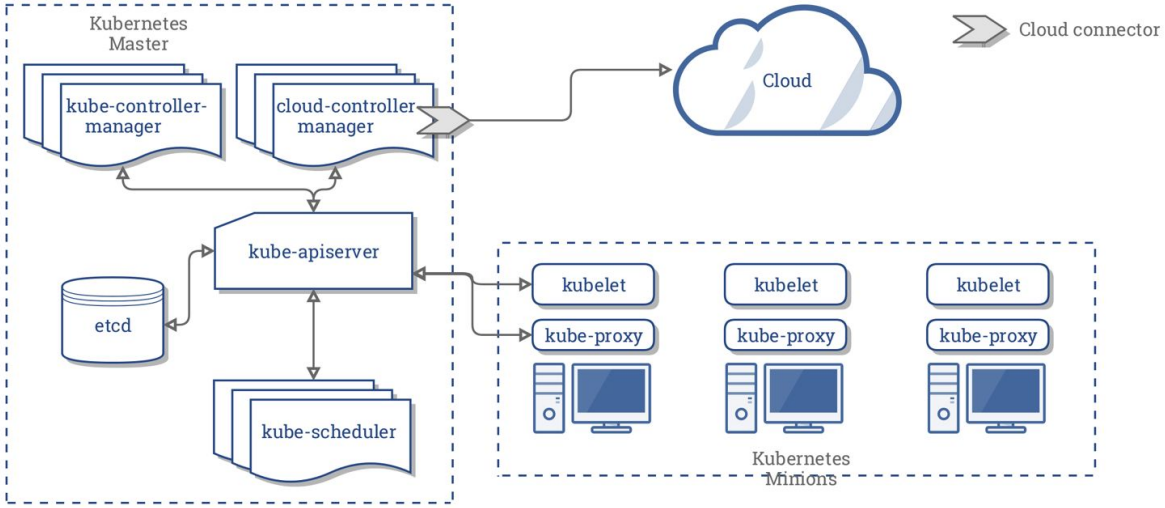
Features of a CCM

Creating your own

Questions?

What is a Cloud Controller Manager (CCM)?

- Daemon
- Cloud Specific Control loops
- Evolve independently of Kubernetes



Reference: [Concepts Underlying the Cloud Controller Manager]

03

About Me

What is a CCM?

Running a CCM

Features of a CCM

Creating your own

Questions?

Choose your adventure...

Add flag to kubelet (if in tree)

```
--cloud-provider=<provider>
```

Create a Deployment or DaemonSet

```
command: [
  '/usr/local/bin/cloud-controller-manager',
  '--cloud-provider=<provider>',
],
image:
k8s.gcr.io/cloud-controller-manager:v1.12.3
```

When you specify a cloud-provider...

```
$ kubectl get nodes -o json | jq '.items[].spec'
  {
    "taints": [{
        "effect": "NoSchedule",
        "key": "node.cloudprovider.kubernetes.io/uninitialized",
        "value": "true"
      } ]
  }
```

04

About Me

What is a CCM?

Running a CCM

Features of a CCM

Creating your own

Questions?

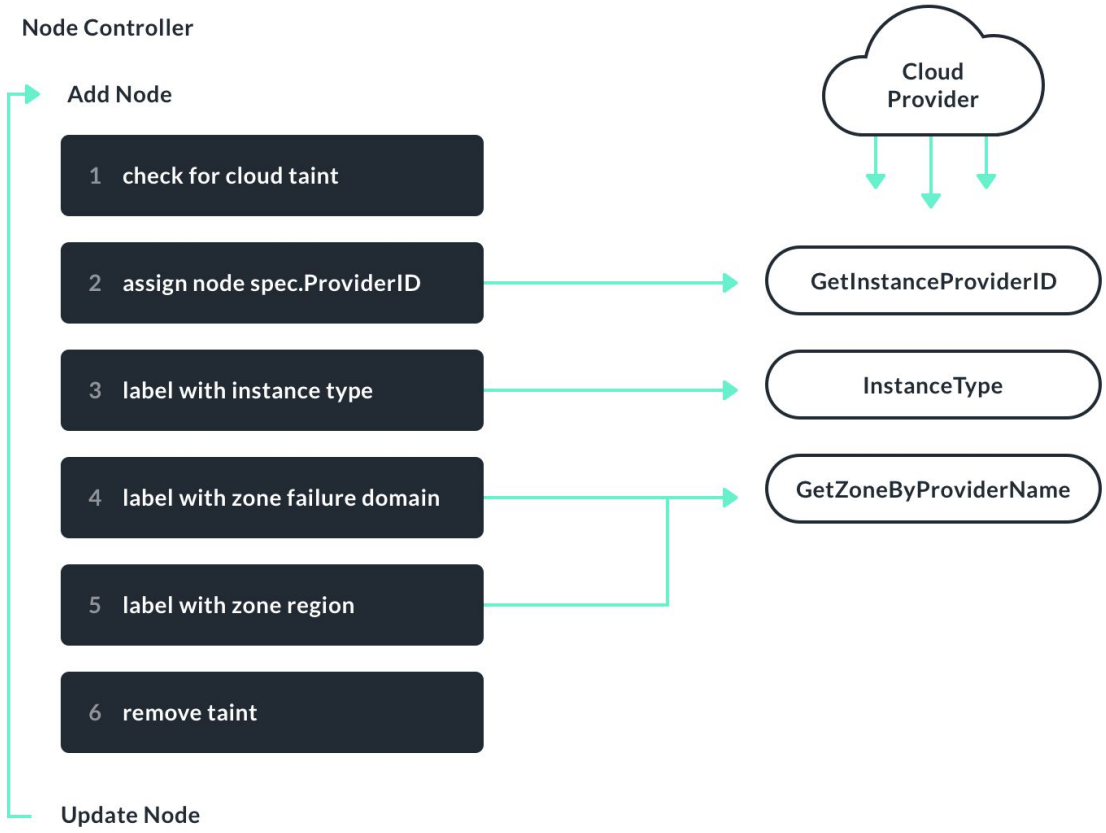
Features of a CCM

- Kubernetes Node Management
- Load balancer Management
- Routing Management
- Any other feature you would like to add if you are running out-of-tree

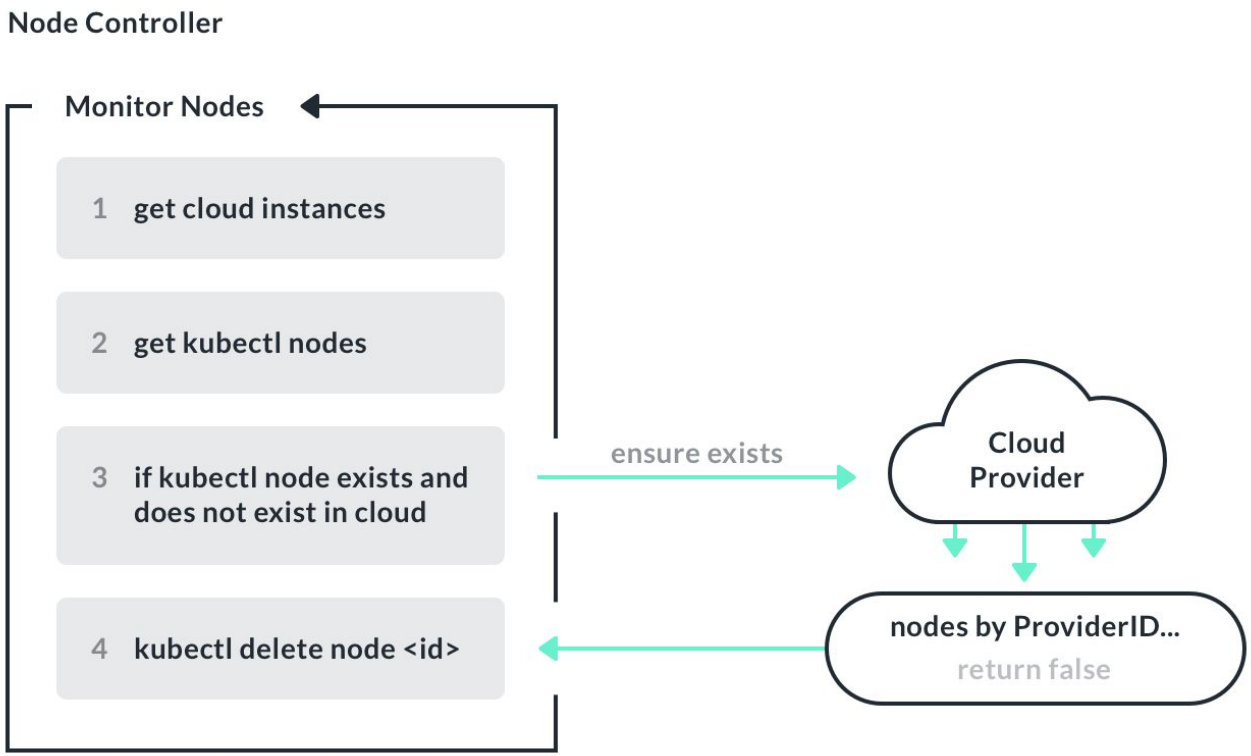
Kubernetes Node Management

- Responsible for initializing nodes
  - Adding providerID and removing taint
    - node.spec.providerID: **<provider>://<instance-id>**
  - Add zone/region labels
  - Writes the nodes network addresses and hostname to the node status
  - Node monitoring

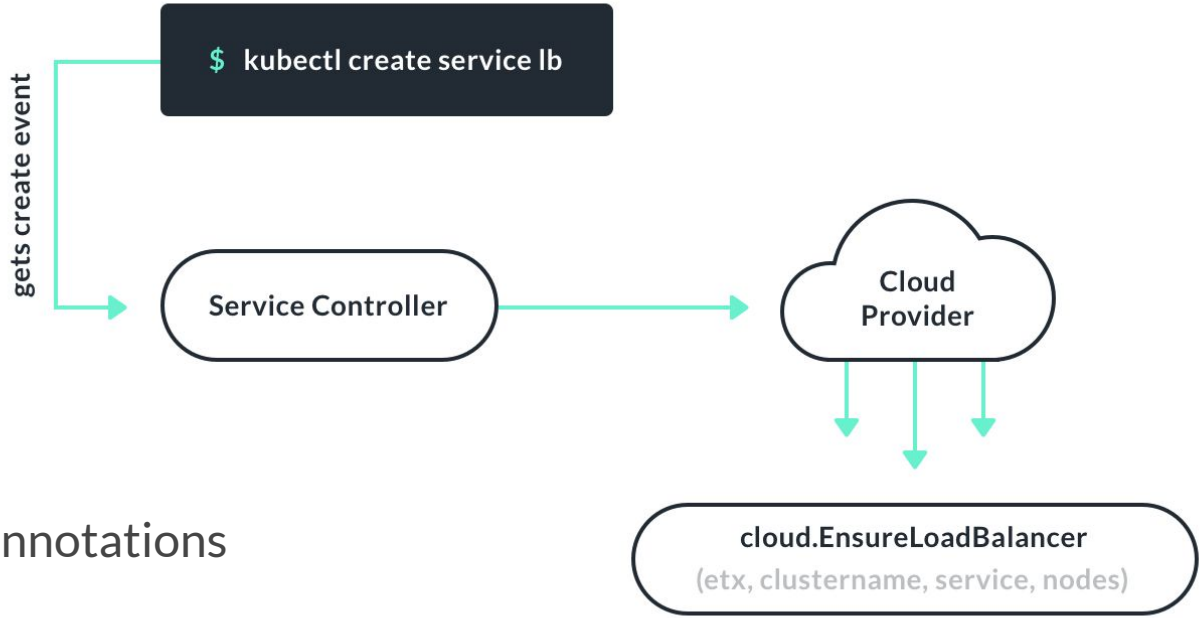
Node Controller



Node Monitoring



Load balancer Management



- Services Controller
- Special features with annotations

Route Management

- Required for GCE
- Pod networking between nodes

Pain Points

- Documentation is sporadic
- Sometimes you have to dig into the code
- Configuration options not documented
- Hard to set up

Pain Points

- CCM’s not standardized between providers
- CCM’s not standardized when running from kubelet vs. deployment
- Container Storage Interface (CSI) /CCM discrepancy

Past, Present, and Future of the CCM

- Cloud specific logic in core
- Pulled into own module in core
- Moving in-tree out of core

Read more: Kep 0002 & 0013

05

About Me

What is a CCM?

Running a CCM

Features of a CCM

Creating your own

Questions?

Building your own

- Implement the Cloud Interface
- Implement each features interface
- Why create your own?

Modules needed

Include core CCM

Import your API

```
import (
  "k8s.io/kubernetes/cmd/cloud-controller-manager/app"
  "k8s.io/kubernetes/cmd/cloud-controller-manager/app/options"

  _ "github.com/organization/custom-ccm/custom" // your custom implementation
)
```

Then run the CCM

```
func main() {
     s, err := options.NewCloudControllerManagerOptions()
     ...
     config, err := s.Config()
     ...
     if err := app.Run(config.Complete()); err != nil {
          fmt.Fprintf(os.Stderr, "%v\n", err)
          os.Exit(1)
     }
}
```

Interface

```go
// Interface is an abstract, pluggable interface for cloud providers.
type Interface interface {
	// Initialize provides the cloud with a kubernetes client builder and may spawn goroutines
	// to perform housekeeping or run custom controllers specific to the cloud provider.
	// Any tasks started here should be cleaned up when the stop channel closes.
	Initialize(clientBuilder ControllerClientBuilder, stop <-chan struct{})
	// LoadBalancer returns a balancer interface. Also returns true if the interface is supported, false otherwise.
	LoadBalancer() (LoadBalancer, bool)
	// Instances returns an instances interface. Also returns true if the interface is supported, false otherwise.
	Instances() (Instances, bool)
	// Zones returns a zones interface. Also returns true if the interface is supported, false otherwise.
	Zones() (Zones, bool)
	// Clusters returns a clusters interface.  Also returns true if the interface is supported, false otherwise.
	Clusters() (Clusters, bool)
	// Routes returns a routes interface along with whether the interface is supported.
	Routes() (Routes, bool)
	// ProviderName returns the cloud provider ID.
	ProviderName() string
	// HasClusterID returns true if a ClusterID is required and set
	HasClusterID() bool
}
```

Reference: [kubernetes/cloud-provider/cloud.go]

Initializing your provider

- Initialize provider
- Registering provider in init()

```
func init() {
        cloudprovider.RegisterCloudProvider(
                providerName,
                func(config io.Reader),
        (cloudprovider.Interface, error) {
                return newCloud(config)
        })
}
```

Return the cloud provider interfaces

- Include the implemented interfaces

```
return &cloud{
        client:        client,
        instances:     newInstances(client, region),
        zones:         newZones(client, region),
        loadbalancers: newLoadbalancers(client, region),
}, nil
```

Examples:

- packet-ccm
- digital-ocean-cloud-controller-manager

Zones

```
// Zone represents the location of a particular machine.
type Zone struct {
        FailureDomain string
        Region        string
}
```

- Used for syncing metadata
- Synced by the Node Controller
- Adds consistency
    - **failure-domain.beta.kubernetes.io/region: nyc1**

Implementing the Zone Interface

Example: [packet/packet-ccm/packet/facilities.go]

```
// Zones is an abstract, pluggable interface for zone enumeration.
type Zones interface {
	// GetZone returns the Zone containing the current failure zone and locality region that the program is running in
	// In most cases, this method is called from the kubelet querying a local metadata service to acquire its zone.
	// For the case of external cloud providers, use GetZoneByProviderID or GetZoneByNodeName since GetZone
	// can no longer be called from the kubelets.
	GetZone(ctx context.Context) (Zone, error)

	// GetZoneByProviderID returns the Zone containing the current zone and locality region of the node specified by providerID
	// This method is particularly used in the context of external cloud providers where node initialization must be done
	// outside the kubelets.
	GetZoneByProviderID(ctx context.Context, providerID string) (Zone, error)

	// GetZoneByNodeName returns the Zone containing the current zone and locality region of the node specified by node name
	// This method is particularly used in the context of external cloud providers where node initialization must be done
	// outside the kubelets.
	GetZoneByNodeName(ctx context.Context, nodeName types.NodeName) (Zone, error)
}
```

CCM & You

- Contribute!
  - sig cloud-provider
- Help out with docs!
  - There is currently a plan waiting for approval with sig-docs
- Build and open source your own
- In tree cloud providers

06

About Me

What is a CCM?

Running a CCM

Features of a CCM

Creating your own

Questions?

Thank you!

Ashley Schuett

@victimnvillain

ashleyschuett

containership