Creating a custom Cluster API Provider

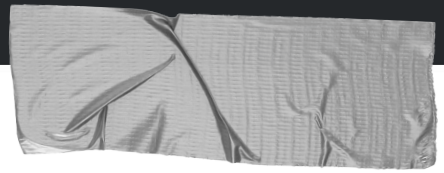
Himani Agrawal
@himani_93

Giri Kuncoro
@girikuncoro

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
JAPAN

History

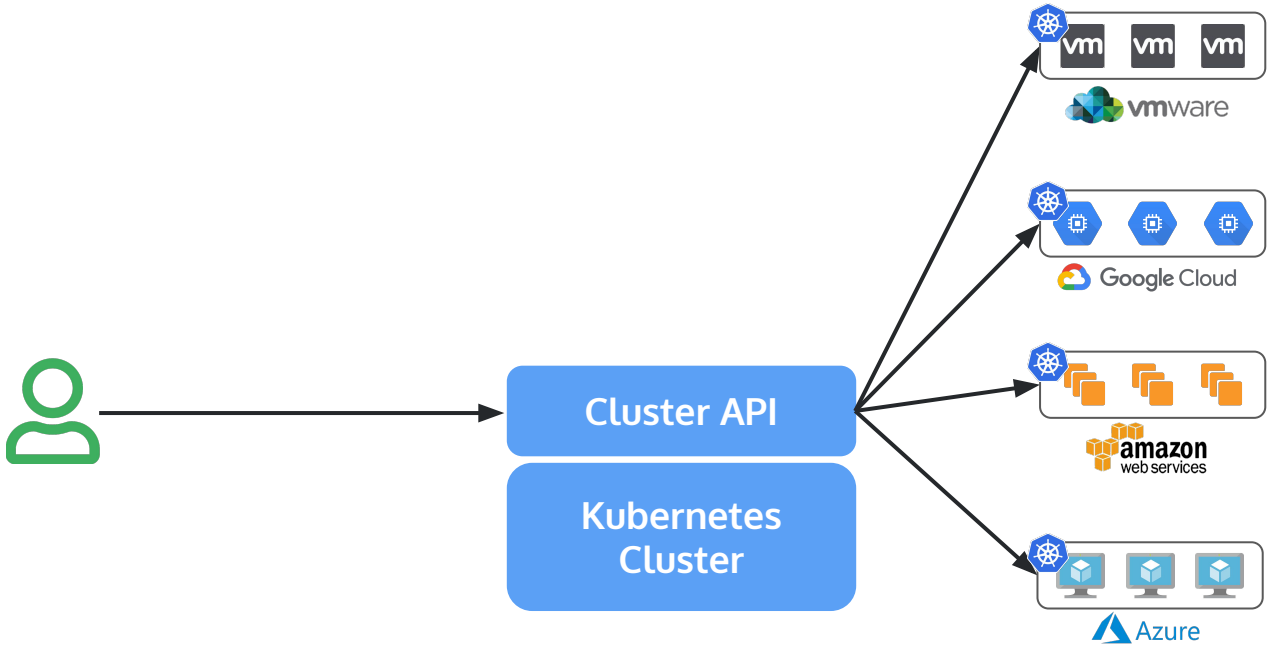
70+ ways to deploy
Kubernetes



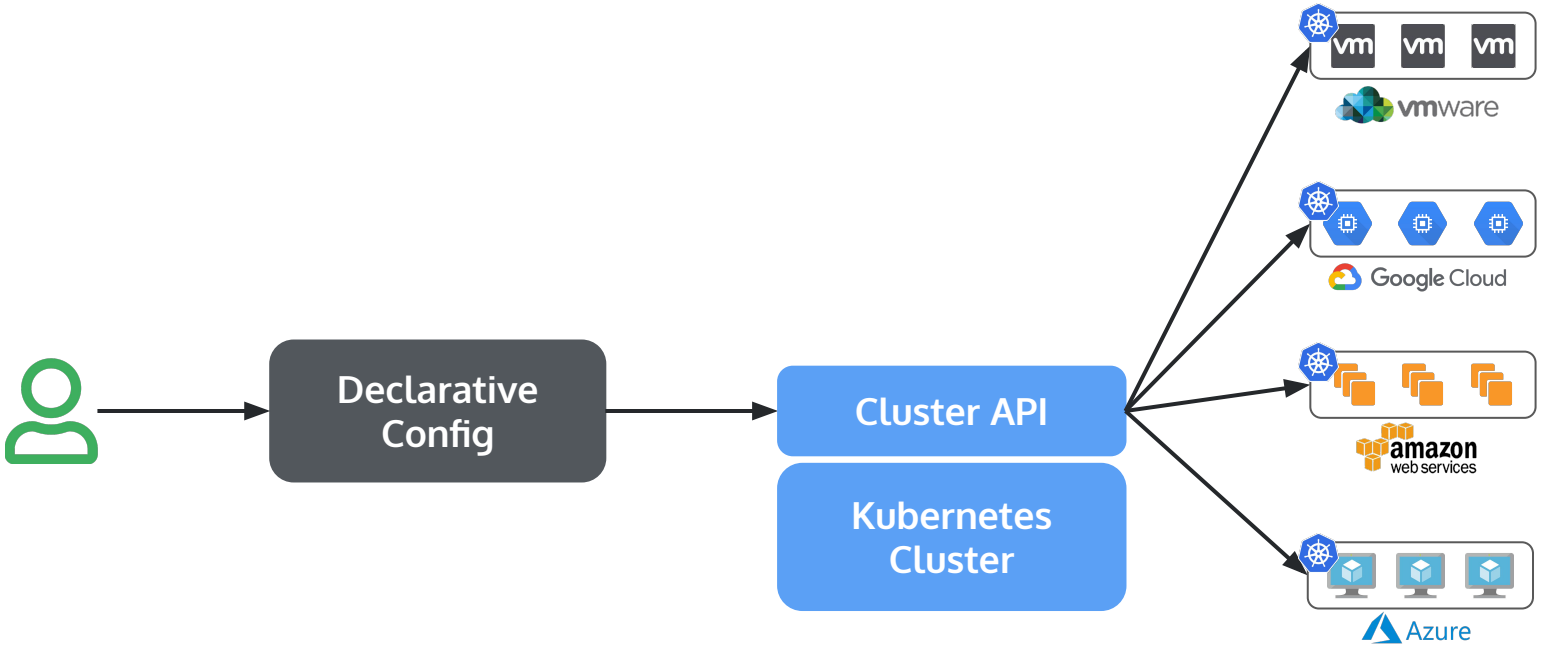
What is missing?

- ➔ Declarative, **Kubernetes-style** API
- ➔ **Compatibility** with tooling
- ➔ **Cloud agnostic**
- ➔ **Single interface** for infra/cluster

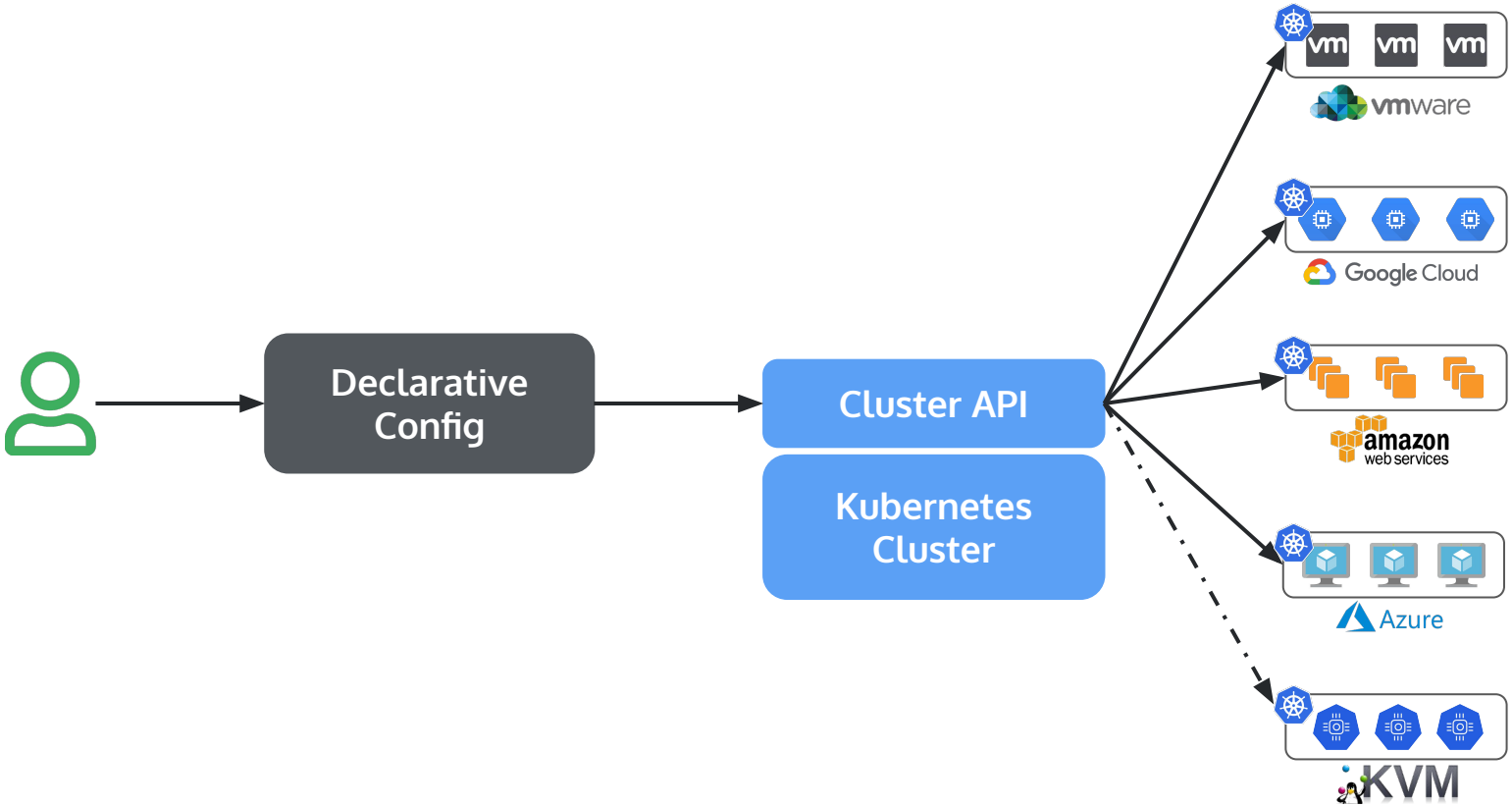
Cluster API



Reference: Deep Dive: Cluster API KubeCon EU 2019



Reference: Deep Dive: Cluster API KubeCon EU 2019



Reference: Deep Dive: Cluster API KubeCon EU 2019

Cluster API CRDs

Cluster

Cluster

```yaml
apiVersion: "cluster-api.k8s.io/v1alpha1"
kind: Cluster
metadata:
  name: fujisan
spec:
  providerSpec:
    ...
  clusterNetwork:
    services:
      cidrBlocks: ["10.96.0.0/12"]
    pods:
      cidrBlocks: ["192.168.0.0/16"]
      serviceDomain: "cluster.local"
```

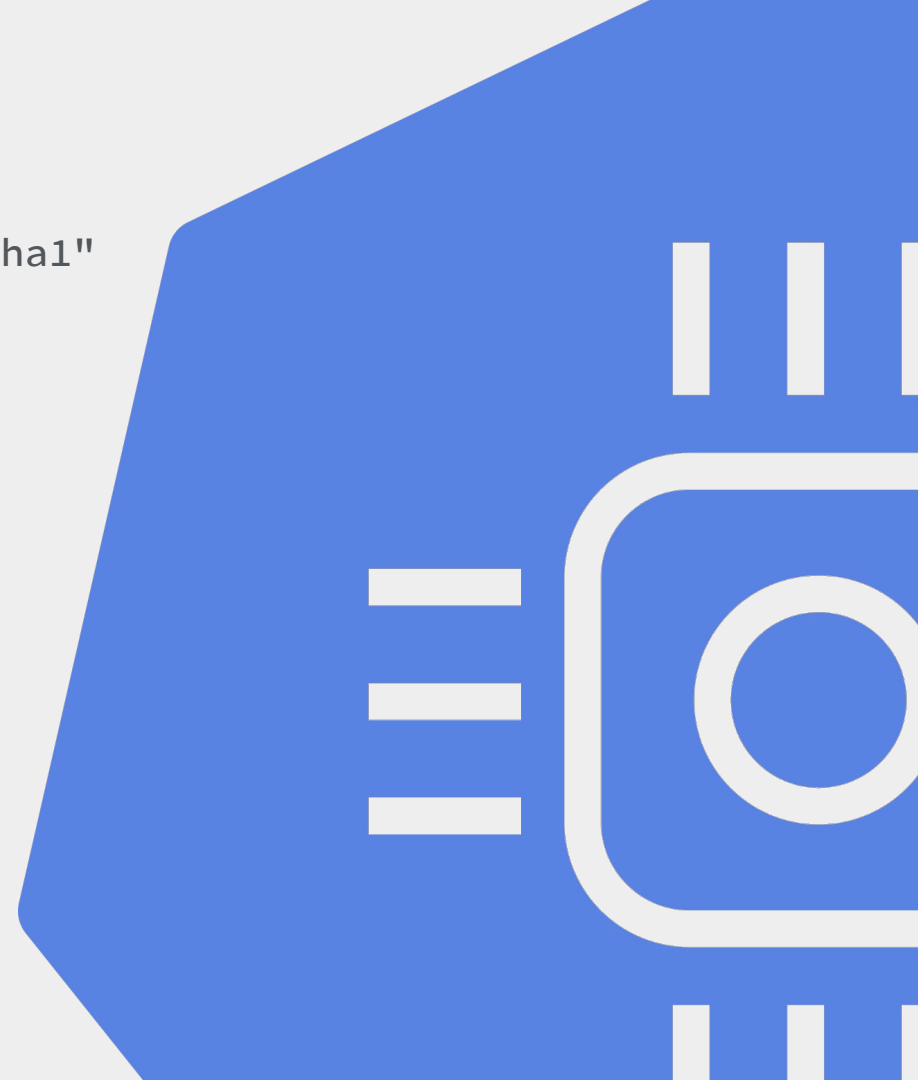
Cluster API CRDs

Cluster

Machine

Machine

```yaml
apiVersion: "cluster-api.k8s.io/v1alpha1"
kind: Machine
metadata:
  name: kube-cp
spec:
  providerSpec:
    vcpu: 2
    ...
  versions:
    kubelet: 1.15.0
```



Cluster API CRDs

Cluster

Machine

Machine Set

Cluster API CRDs

Cluster

Machine

Machine Set

Machine Deployment

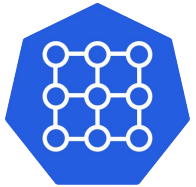
Cluster API CRDs

Cluster

Machine

Machine Set

Machine Deployment



Machine Class

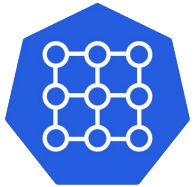
Cluster API CRDs

Cluster

Machine

Machine Set

Machine Deployment



Machine Class

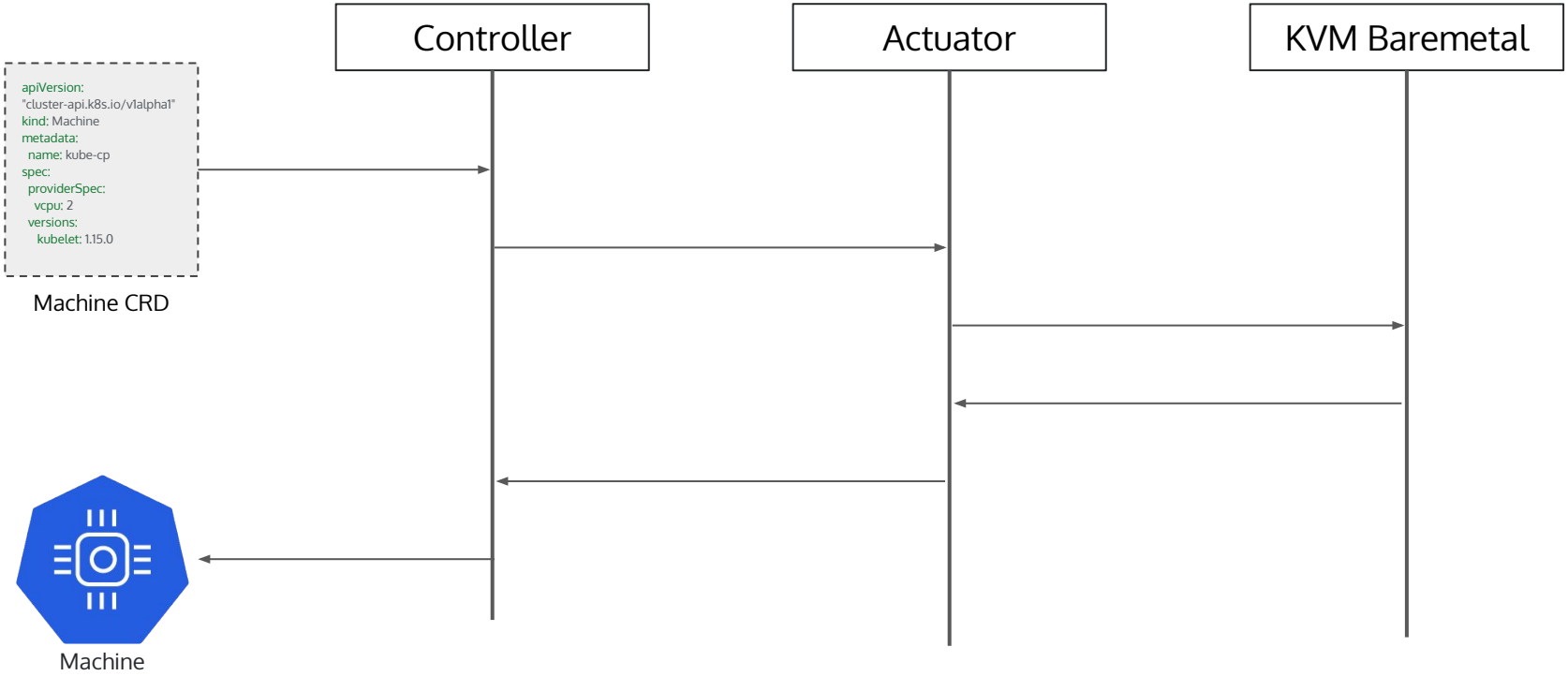
Pod

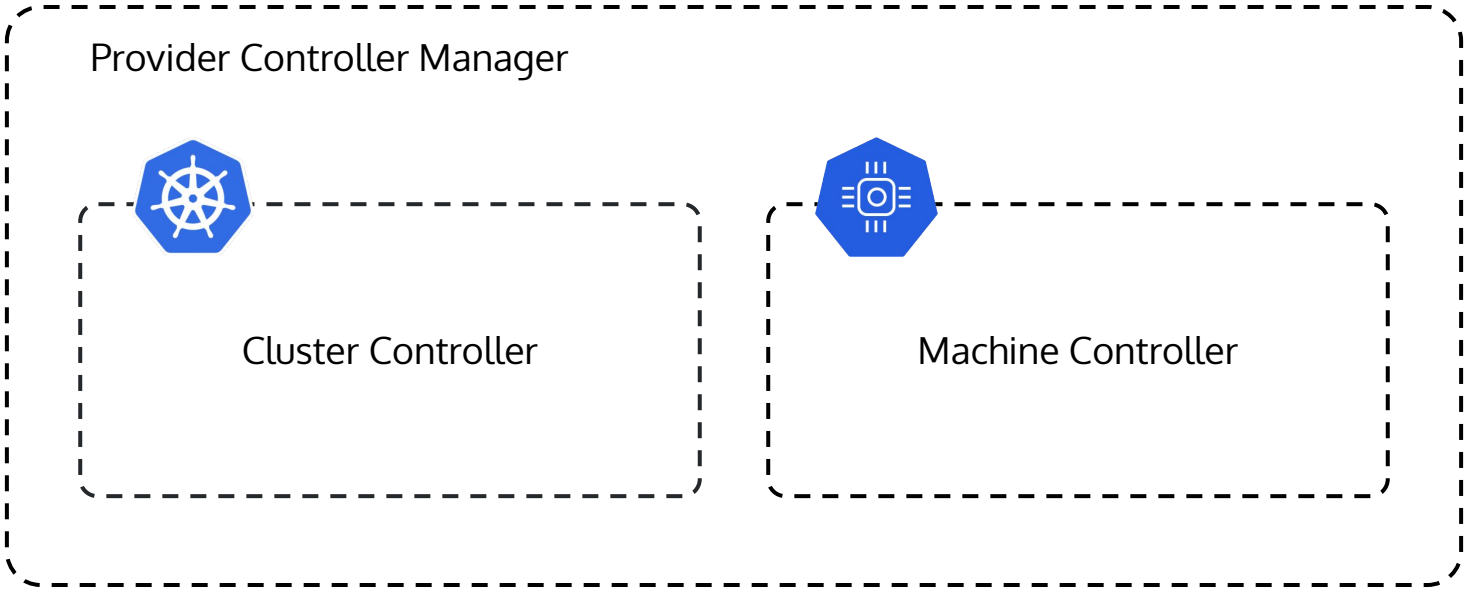
Replica Set

Deployment

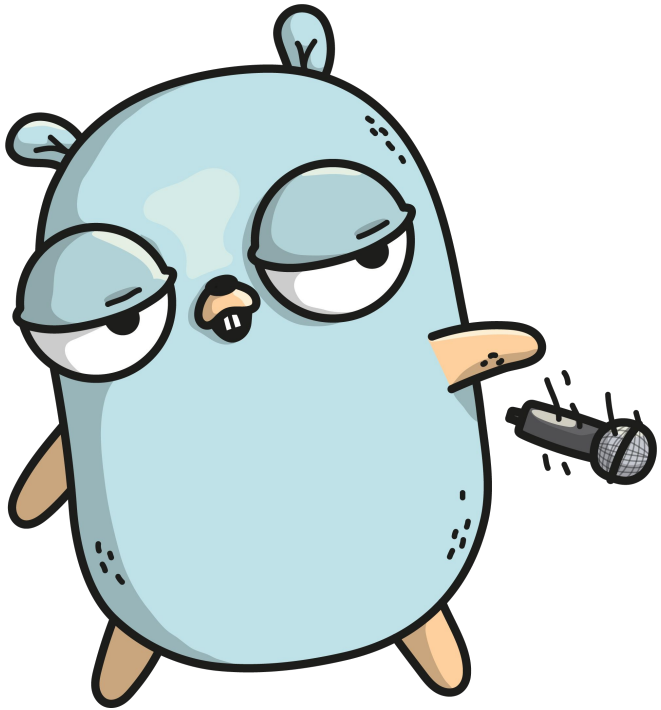
Storage Class

How Cluster API works?




Provider Controller Manager
Cluster Controller
Machine Controller

Demo



Picture Credits: Ashley McNamara

Code

cmd/manager/main.go

```
func main() {

  mgr, _ := manager.New(cfg, opts)

  machineActuator, _ := machine.NewActuator(machine.ActuatorParams{
      Client: mgr.GetClient(),
  })

  apis.AddToScheme(mgr.GetScheme())

  clusterapis.AddToScheme(mgr.GetScheme())

  capimachine.AddWithActuator(mgr, machineActuator)

  mgr.Start(signals.SetupSignalHandler())

}
```

Code

pkg/controller/add_machine_controller.go

```
func init() {

  // AddToManagerFuncs is a list of functions to create controllers
  // and add them to a manager.
  AddToManagerFuncs = append(AddToManagerFuncs, func(m manager.Manager) error {
    return capimachine.AddWithActuator(m, &machine.Actuator{})
  })
}
```

Code

pkg/apis/libvirt/v1alpha1/libvirtmachineproviderspec_types.go

```
type LibvirtMachineProviderSpecSpec struct {
  // Number of virtual CPU
  VCPU int `json:"vcpu"`

  // Amount of RAM in GBs
  MemoryInGB int `json:"memoryInGB"`

  // Image URL to be provisioned
  ImageURI string `json:"imageURI"`

  // UserData URI of cloud-init image
  UserDataURI string `json:"userDataURI"`
}
```

Code

pkg/cloud/libvirt/domain.go

```
// CreateDomain connects to libvirt daemon and creates a new domain
// as per domainXML.
func CreateDomain(domainXML string) error {
  ...
}


// DomainExists checks if the domain with the given name exists.
func DomainExists(domainName string) bool {
  ...
}


// defineDomain returns the XML representation of the domain to be created.
// Output of this func is passed into CreateDomain() to create a new domain.
func defineDomain() string {
  ...
}
```

Code

pkg/cloud/libvirt/actuators/machine/actuator.go

```go
func (a *Actuator) Create() error {
  return libvirt.CreateDomain()
}


func (a *Actuator) Exists() (bool, error) {
  return libvirt.DomainExists()
}


func (a *Actuator) Delete() { ... }


func (a *Actuator) Update() { ... }
```

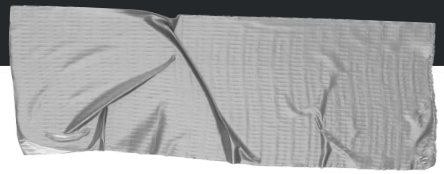
/himani93/cluster-api-provider-libvirt

Dependency Versions

- Cluster API v0.1.0
- Kubebuilder v1.0.8
- Kustomize v1.0.11
- Kubectl v1.13
- Minikube v1.2.0

Discussion

- Cluster API is **super cool**!
- **Early work** for production version



Getting Involved

- ➔ Github Repo
  github.com/kubernetes-sigs/cluster-api
- ➔ Mailing List: kubernetes-sig-cluster-lifecycle
  https://groups.google.com/forum/#!forum/kubernetes-sig-cluster-lifecycle
- ➔ Slack: #cluster-api
  https://kubernetes.slack.com/messages/C8TSNPY4T

Provider Implementations

- AWS, https://github.com/kubernetes-sigs/cluster-api-provider-aws
- Azure, https://github.com/kubernetes-sigs/cluster-api-provider-azure
- Baidu Cloud, https://github.com/baidu/cluster-api-provider-baiducloud
- Bare Metal, https://github.com/metal3-io/cluster-api-provider-baremetal
- DigitalOcean, https://github.com/kubernetes-sigs/cluster-api-provider-digitalocean
- Exoscale, https://github.com/exoscale/cluster-api-provider-exoscale
- GCP, https://github.com/kubernetes-sigs/cluster-api-provider-gcp
- IBM Cloud, https://github.com/kubernetes-sigs/cluster-api-provider-ibmcloud
- OpenStack, https://github.com/kubernetes-sigs/cluster-api-provider-openstack
- Talos, https://github.com/talos-systems/cluster-api-provider-talos
- Tencent Cloud, https://github.com/TencentCloud/cluster-api-provider-tencent
- vSphere, https://github.com/kubernetes-sigs/cluster-api-provider-vsphere

API Adoption

Following are the implementations managed by third-parties adopting the standard cluster-api and/or machine-api being developed here.

- Kubermatic machine-controller, https://github.com/kubermatic/machine-controller/tree/master
- Machine API Operator, https://github.com/openshift/machine-api-operator/tree/master
- Machine-controller-manager, https://github.com/gardener/machine-controller-manager/tree/cluster-api

References

- https://kccna18.sched.com/event/Greh
- https://kccnceu19.sched.com/event/MPkR
- https://blogs.vmware.com/cloudnative/2019/03/14/what-and-why-of-cluster-api/
- https://blog.heptio.com/the-kubernetes-cluster-api-de5a1ff870a5
- https://github.com/kubernetes-sigs/cluster-api
- https://cluster-api.sigs.k8s.io/

ありがとうございます。