kops

Justin Santa Barbara, Google

Overview

Introduction to kops / demo

Terraform / Declarative management in git

Project Overview

Kops "under the hood"

Kops roadmap

UX - creation

```
kops create cluster --zones us-east-1c cluster.example.com

kops update cluster cluster.example.com --yes

kubectl create -f app.yaml
```
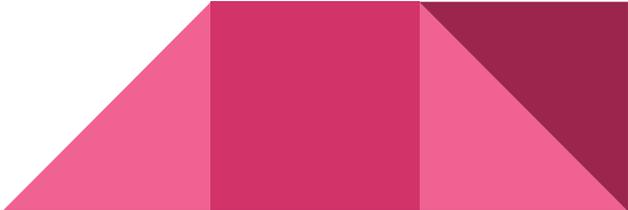
UX - upgrades

```
kops upgrade cluster

kops update cluster --yes

kops rolling-update cluster --yes
```
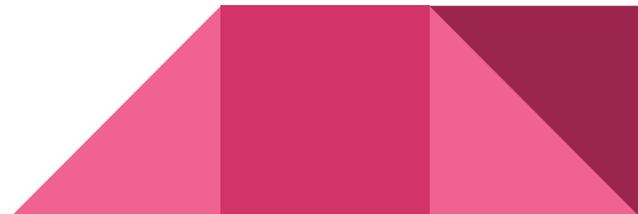
UX - terraform

```
kops update cluster --target=terraform

cd out; tf plan; tf apply
```
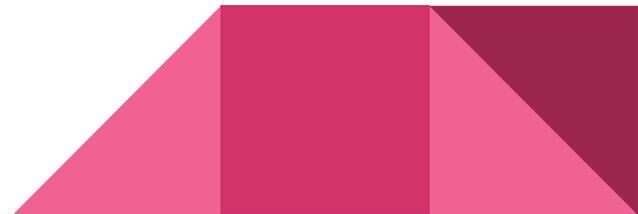
UX - declarative management

```
kops get cluster.example.com -oyaml

git add .; git commit

...⌛⌛⌛...

git pull

kops replace -f all.yaml
```
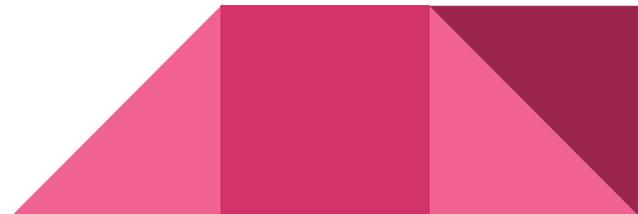
More clouds!

Today

AWS

GCE

In progress

Digital Ocean

OpenStack

VSphere

Bare Metal

Project growth

~400 contributors

> 5k commits

> 5k github stars

How it works - architecture

api-machinery for core **Objects** (Clusters, InstanceGroups)

Model layer maps objects to Tasks

Tasks reliably create / change cloud objects
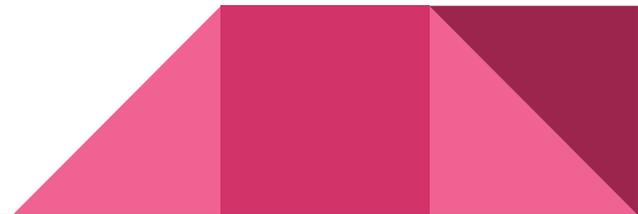
How it works - on the CLI

Tasks are run and create/update cloud objects

     Kubernetes-inspired reconciliation of actual & desired state

Kops creates the instances and then exits the picture

    +   can export to terraform / cloudformation

    +   self-healing

    -   we should check success
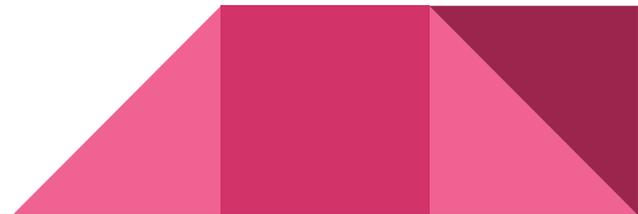
How it works - on the node

Instance boots & runs startup script

Downloads nodeup agent

Nodeup reads state from S3/GCS bucket; installs core components

    packages, docker, protokube, kubelet, control plane manifests

Protokube mounts volumes, starts etcd, installs system manifests (kube-dns etc)

DNS for discovery

One of the big design decisions was to use DNS for discovery

Proved good, but a big hurdle for some

Gossip DNS added to make that easier
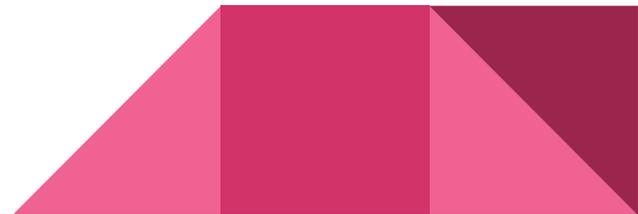
Break up kops into reusable pieces

Protokube

etcd-manager

addon-management

dns / discovery manager

Nodeup

Move more logic out of nodeup

Integrate with kubeadm

kops-cli

Machines API

etcd-manager

etcd2 -> etcd3 migration has been a big effort for kops

kops is doing HA etcd2 -> etcd3 migration

Championing backups & reliable automation

Etcd-manager splits code out of protokube

1.9 backups, 1.10 opt-in to etcd-manager & etcd3 default, 1.11 opt-out
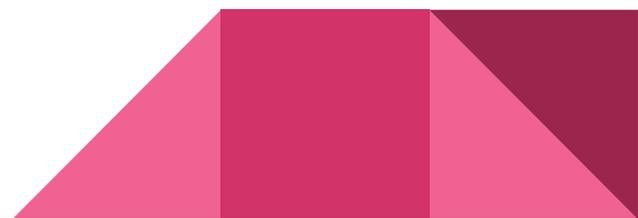
Cluster API / Machines API

- A declarative way to create, configure, and manage a cluster
  - apiVersion: "cluster.k8s.io/v1alpha1"
  - kind: Cluster, Machine, MachineSet, MachineDeployment

- Cluster: General cluster configuration (e.g. networking)
- Machine: A single physical or virtual machine
- MachineSet / MachineDeployment
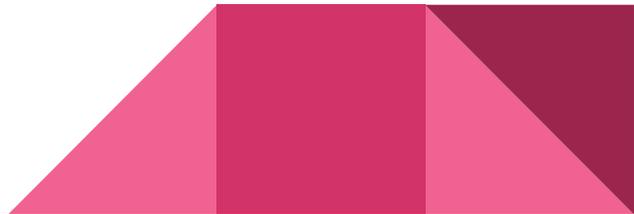
dns / discovery

Move to external-dns

Other mechanisms for discovery

    iptables?
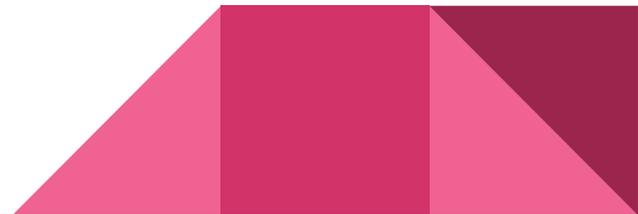
addon-management

kops includes some "extras"

    DNS, CNI, kube-dns, cloud-controller-managers

Currently these are baked into kops

Ideally: maintained by projects

Thank you!

Q & A