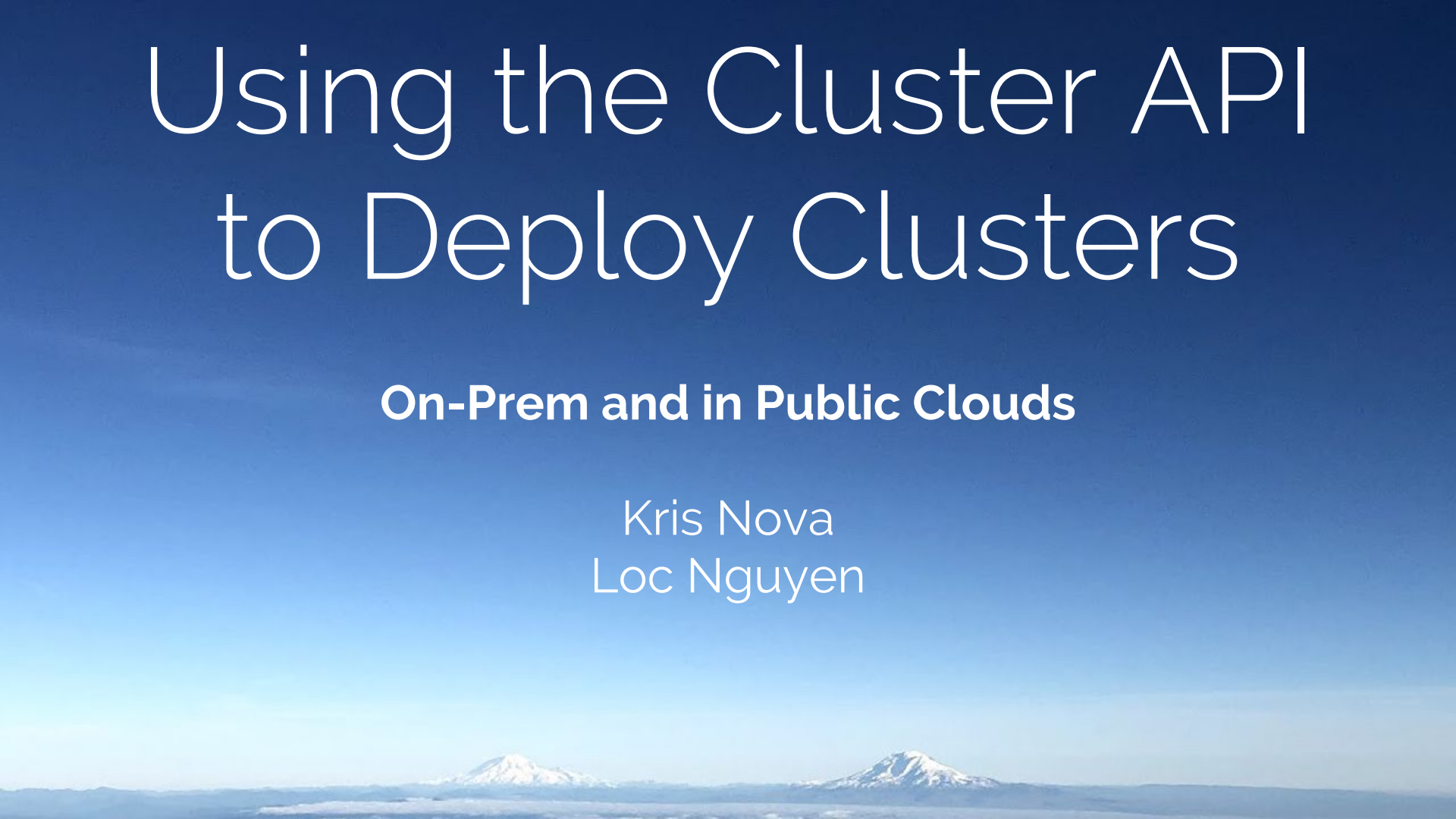
# Using the Cluster API to Deploy Clusters

**On-Prem and in Public Clouds**

Kris Nova
Loc Nguyen



# Appendix

# j.hept.io/cluster-api-kubecon-2018


Loc Nguyen
- OSS Engineer at VMware focused on the container space
- github.com/sflxn

Kris Nova
- Senior developer advocate at Heptio
- CNCF Ambassador
- github.com/kris-nova/me


# Cluster API Today

- A management framework to handle day 1 and 2 operations for kubernetes cluster
  - Day 1) Bringing up a cluster
    - Solves from 0 to Kubernetes
  - Day 2) Managing a cluster
    - Managing in an idiomatic kubernetes way
    - Upgrades
    - Scaling
  - Standardizing a fragmented ecosystem
    - Many tools, all with varying scope and user experience
- Experimenting still!

# Cluster API Today

- Core is in Alpha
- Many provider breakouts
- Thank you




Google Cloud

vmware

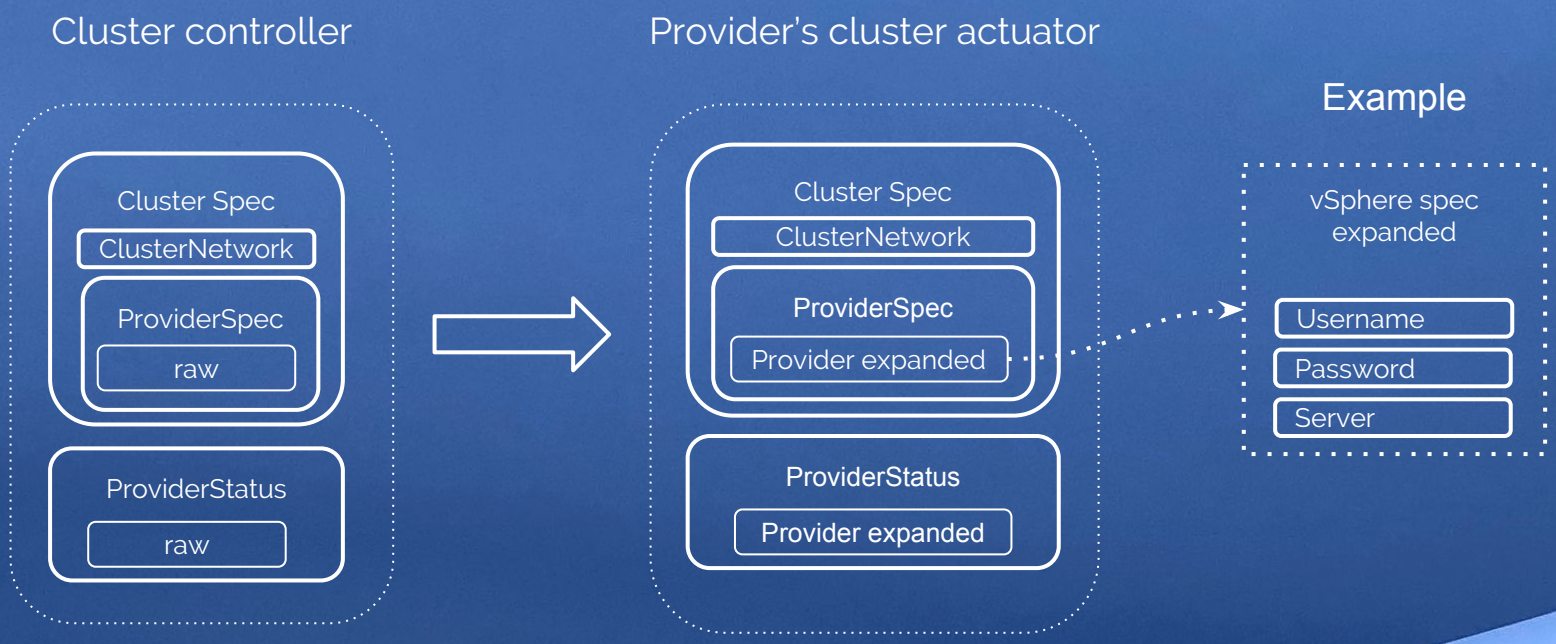
# Declared State of Cluster API

- Project was started in 2017
  - History of cluster api blog found in Appendix (j.hept.io/cluster-api-kubecon-2018)
- Project Goals:
  - Atomic transactions of cluster management operations
  - Cluster automation
  - Resilient infrastructure
  - Improve user experience
  - Cluster upgrade
  - Create a community
- Additional goal - cross infrastructure deployment

# Cluster API Basics

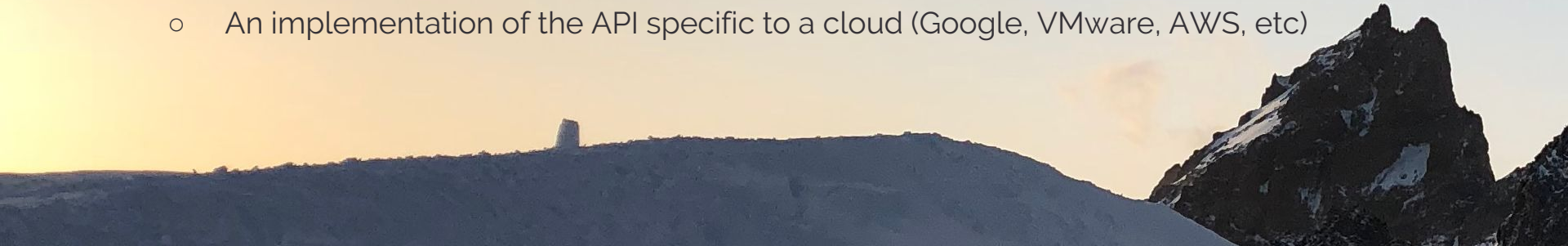
- clusterctl
  - Generic CLI tool for the project
  - Each cloud provider forks
- Deploys controller that reads cluster api CRD objects
  - Provider specific
- Deploys CRD objects to **some** kubernetes cluster
  - Cluster{}
  - Machine {}
  - MachineSet{}, MachineDeployment{}
- Controllers reconciles new cluster accordingly
  - Can create new cluster
  - Can mutate existing cluster

# Controller Basics



# Terms

- Target cluster
  - The declared cluster we intend to create and manage
- Bootstrap/Management cluster
  - The cluster that manages the target cluster
  - Possibly the same cluster
- clusterctl
  - Community CLI tool that favors a provider implementation for creating and managing a cluster
- Provider implementation
  - An implementation of the API specific to a cloud (Google, VMware, AWS, etc)



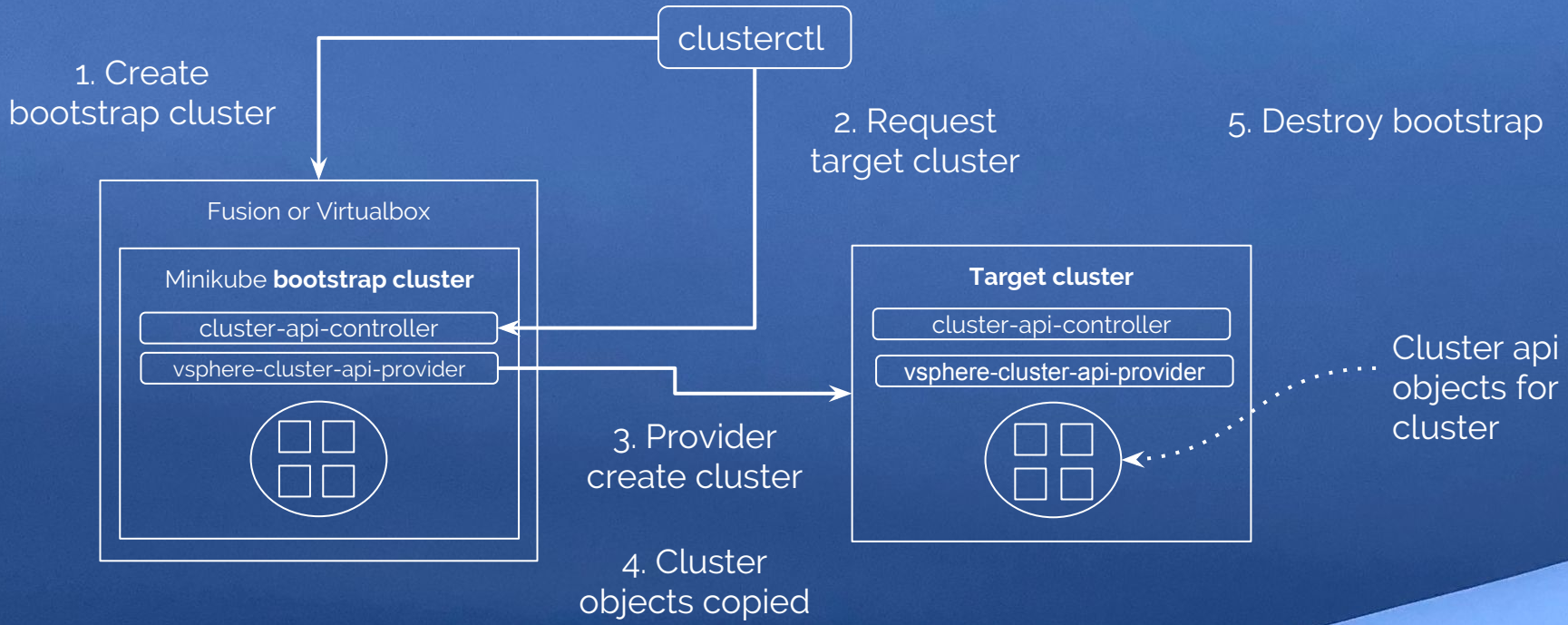
# Usage Modes

- Self Service (original)
  - Kubernetes managing itself
    - Bootstrap problem
    - Bootstrap cluster
  - Source of truth managing self
  - Simpler
- Management Cluster
  - Separation of concerns (admin vs cluster creators)
  - Some **secondary** management cluster
  - A little more complex
  - Flexible

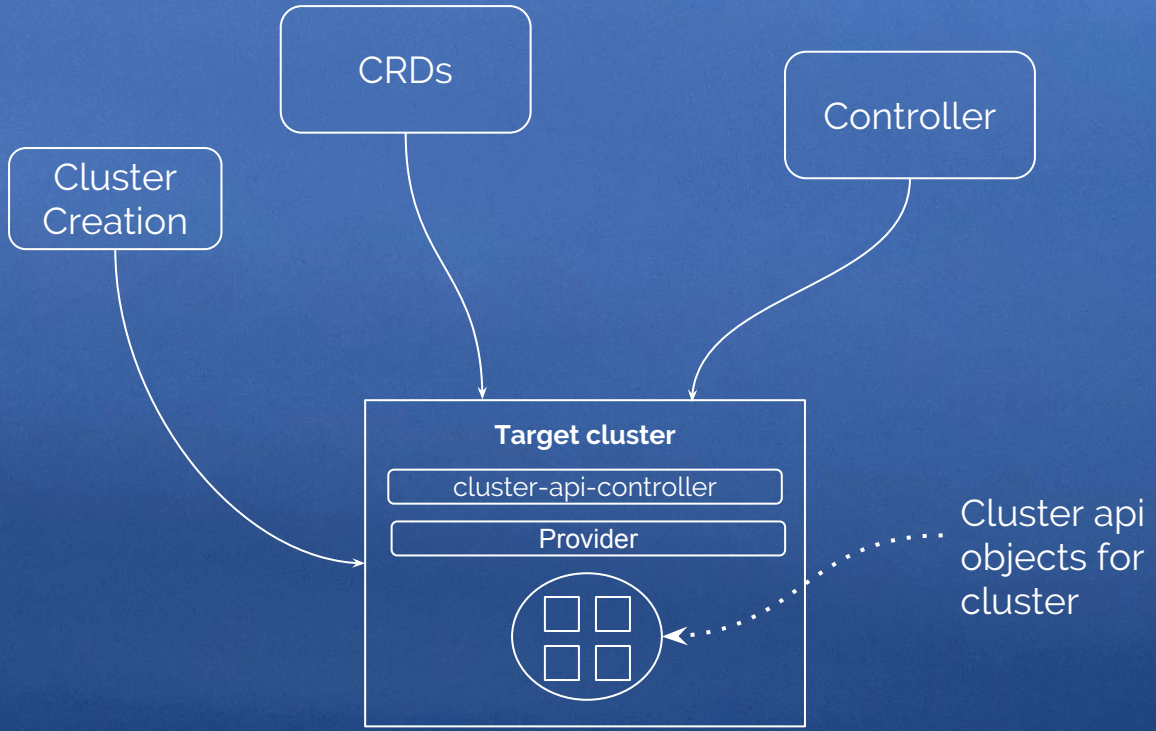
# Usage Mode:  Self-service

- When kubernetes manages itself
  - Either by itself via solving the **bootstrap problem** or via a bootstrap cluster
  - Clusterctl or some CLI drives the process
- Traits
  - Have to solve original cluster problem
    - Using minikube requires a hypervisor
  - Decentralized storage of cluster declaration
- Strength
  - Simple.  User can deploy from desktop with little oversight.
- Weakness
  - Decentralize nature makes central accounting of clusters harder
  - May not be allowed in some enterprises

# Usage Mode:  Self-service (bootstrap)



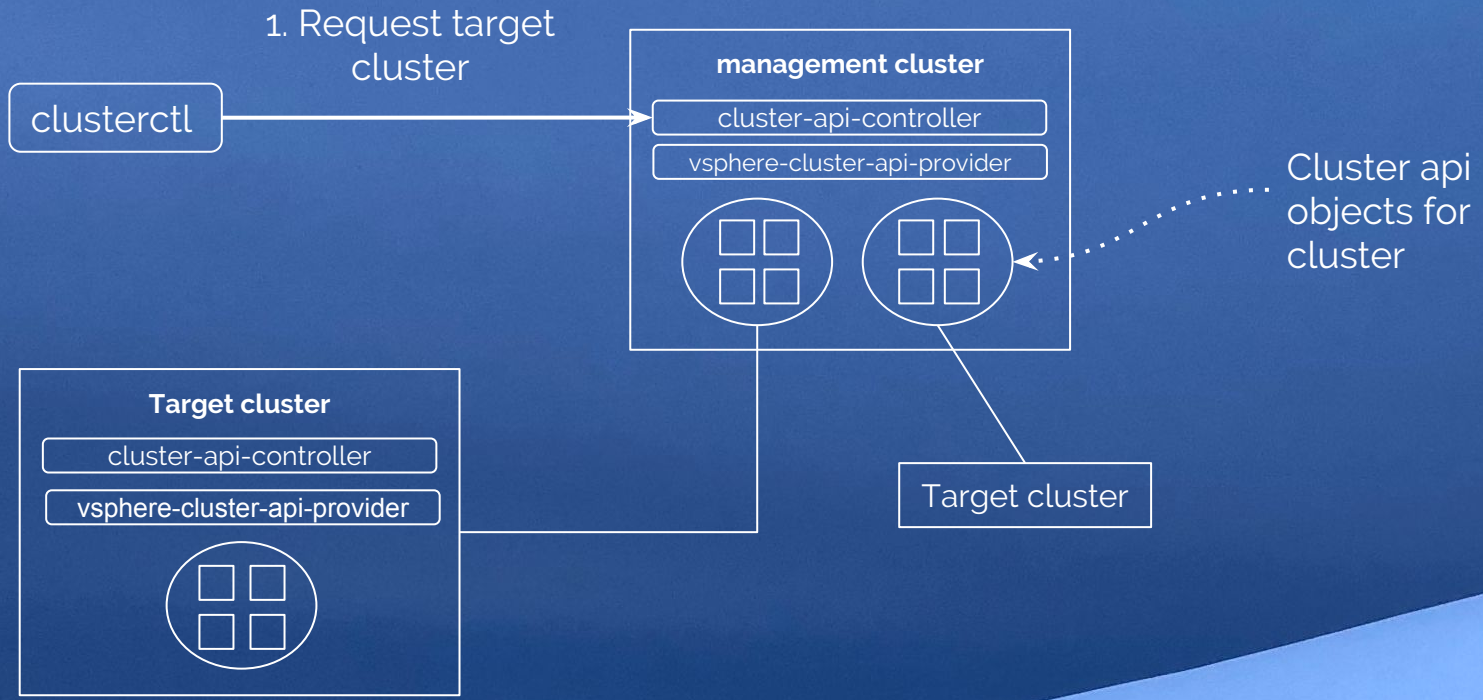
# Usage Mode:  Self-service (self)



# Cluster Lifecycle:  Management Cluster

- Secondary design
- Solves the bootstrap problem of how to create the initial cluster
- Traits
  - Has two primary roles:  admin and cluster creator
  - Central storage of clusters declarations
- Strength
  - Central accounting of clusters declarations makes monitoring easier
  - User do not need any extra software (e.g. minikube or desktop hypervisor)
- Weakness
  - Requires admin to deploy the initial cluster
  - Until these projects are mature, getting logs, tracing maybe difficult

# Cluster Lifecycle:  Management Cluster



# vSphere Provider

- Supports self-service and management cluster models
- Admin workflow
  - Deploys OVA to vSphere
    - OVA deploys initial ***management cluster***
  - SCP kubeconfig from the management cluster
  - Deploys Cluster API to the management cluster (via clusterctl or kubectl)
  - Creates a VM template from a cloud image OS ISO (e.g. Ubuntu)
  - Provide kubeconfig to cluster creators
- Cluster creators workflow
  - Uses clusterctl to create target cluster using the kubeconfig

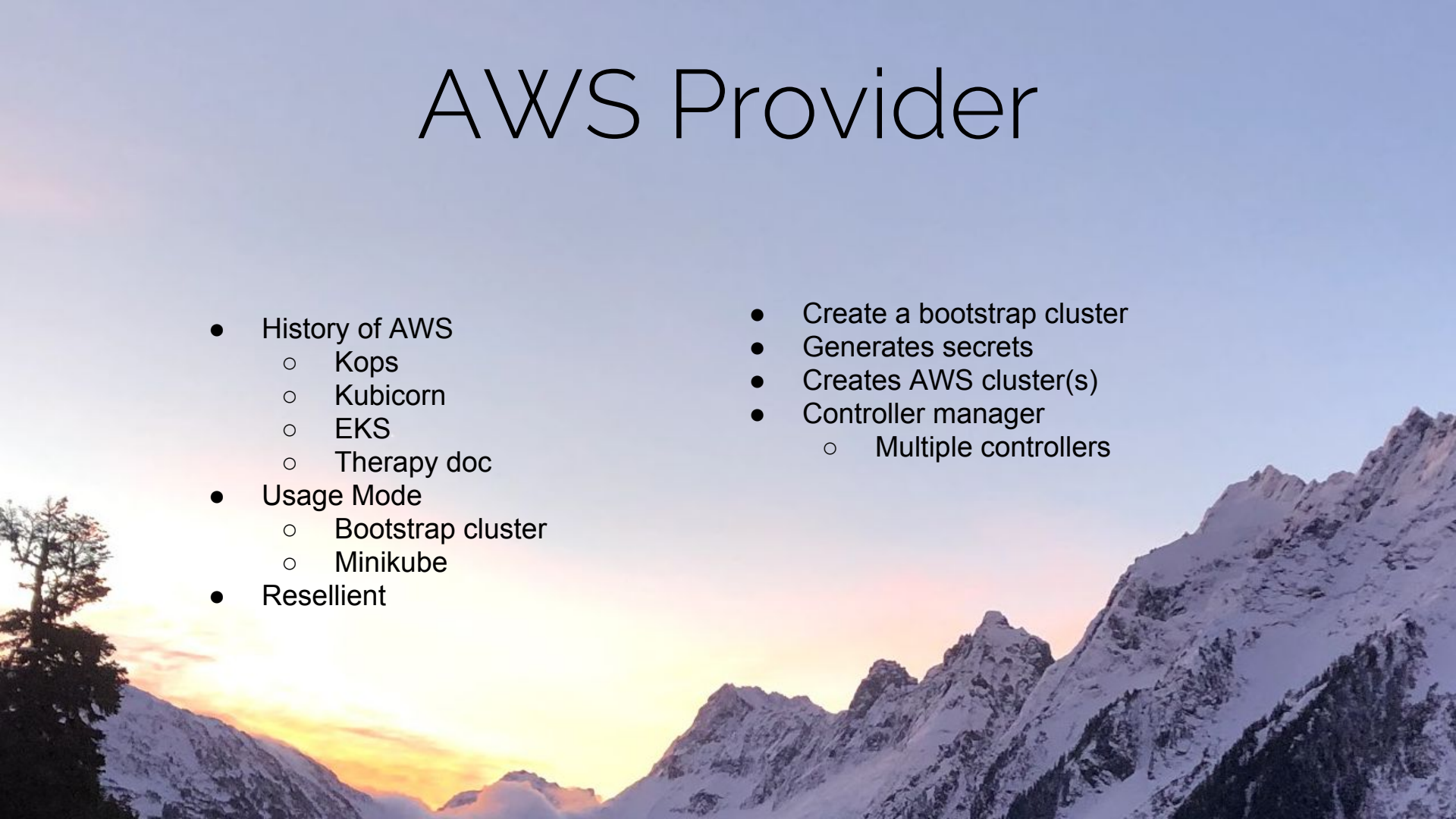
# vSphere Provider

- Why a management cluster?
  - Many users of vSphere are administrators and devops
  - Visibility into resources used by the created clusters
  - An OVA is a preferred deployment model for vSphere users
    - Server driven and not desktop driven
    - Removes requirements for additional software (minikube and hypervisor)
    - Less stack to debug if things go wrong (minikube and hypervisor)
  - OVA can present UI wizard that can customize the management cluster
    - CNI (future)
    - Backup/restore (future)

# AWS Provider

- History of AWS
  - Kops
  - Kubicorn
  - EKS
  - Therapy doc
- Usage Mode
  - Bootstrap cluster
  - Minikube
- Resellient

- Create a bootstrap cluster
- Generates secrets
- Creates AWS cluster(s)
- Controller manager
  - Multiple controllers

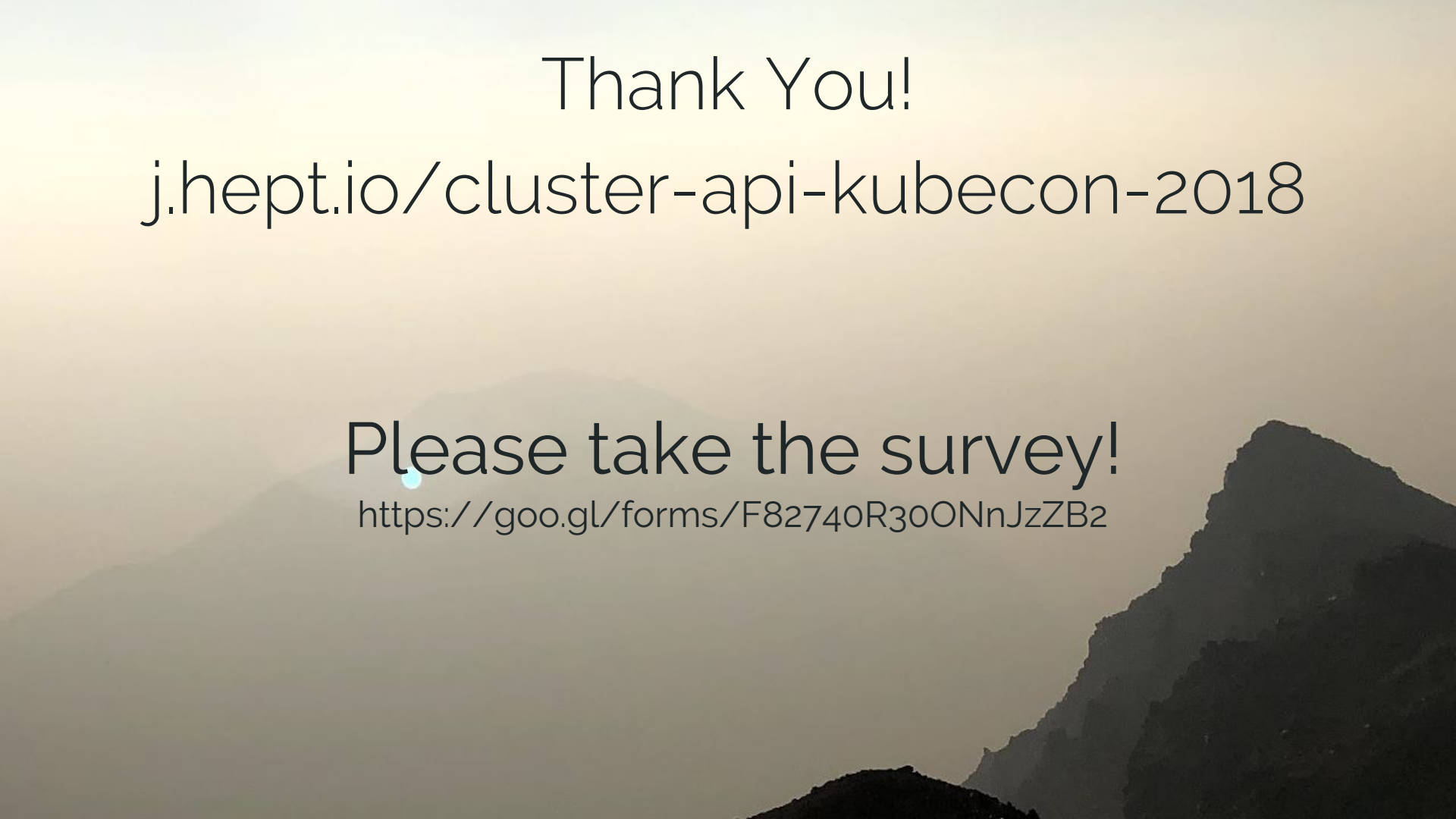


# Q&A

- Does audience want to see demos?
  - Less time for Q&A if we do demos
  - Show of hands

# Related Sessions

- Cluster API Deep Dive, Robert Bailey and David Watson
  - Thurs, 2:35-3:10


Thank You!
j.hept.io/cluster-api-kubecon-2018
Please take the survey!
https://goo.gl/forms/F82740R30ONnJzZB2