KubeCon | CloudNativeCon

OPEN SOURCE SUMMIT

China 2019

KubeCon CloudNativeCon

OPEN SOURCE SUMMIT

China 2019

Intro: SIG-OpenStack

Chris Hoge, OpenStack Foundation

SIG-OpenStack Governance

SIG-OpenStack

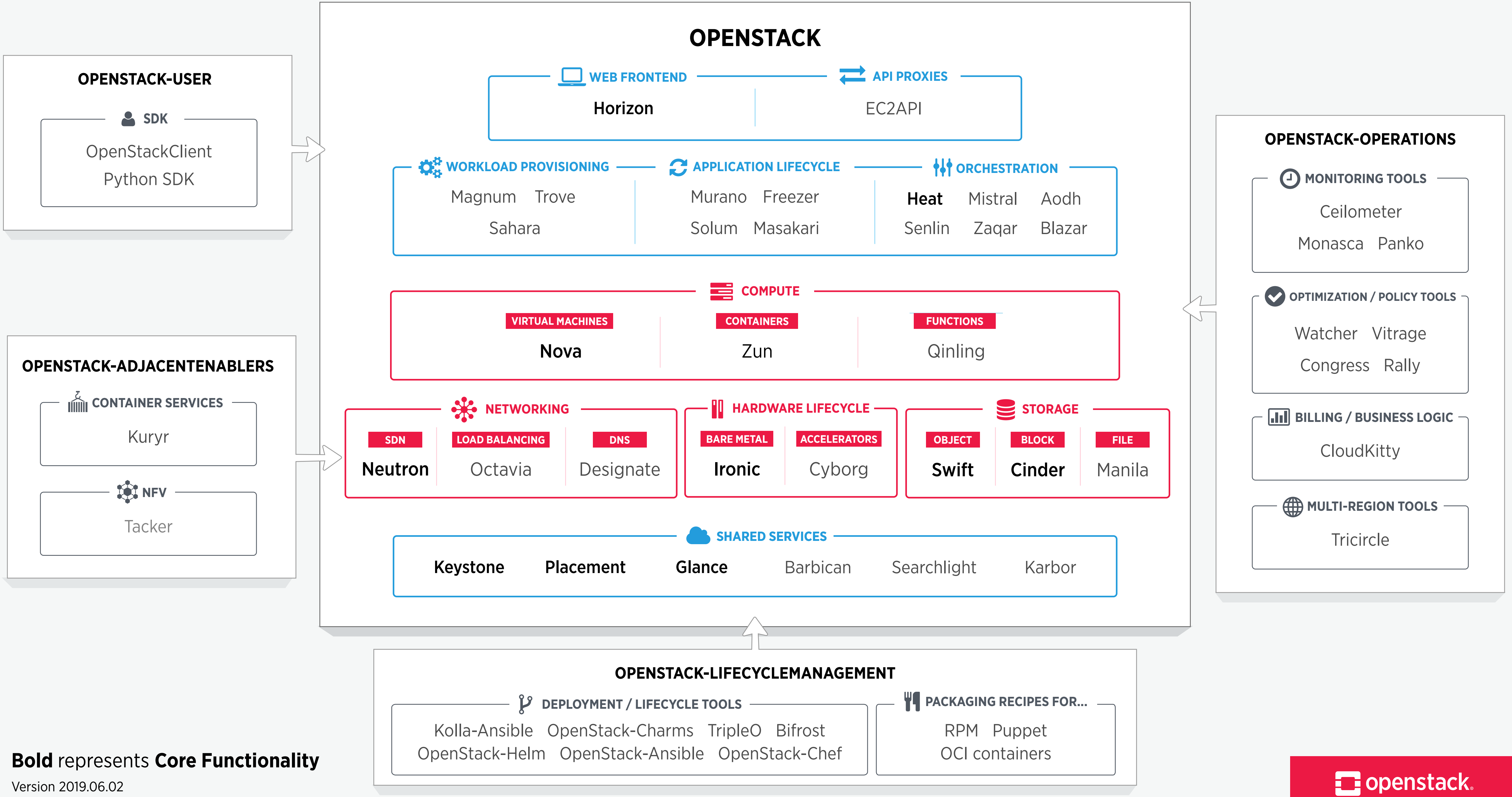
Coordinates the cross-community efforts of the OpenStack and Kubernetes communities.

Includes OpenStack-related contributions to Kubernetes.

OpenStack can be:

- A deployment platform for Kubernetes.
- A service provider for Kubernetes.
- A collection of applications to run on Kubernetes

https://github.com/kubernetes/community/tree/master/sig-openstack



OPENSTACK

WEB FRONTEND: **Horizon**

API PROXIES: EC2API

WORKLOAD PROVISIONING: Magnum, Trove, Sahara

APPLICATION LIFECYCLE: Murano, Freezer, Solum, Masakari

ORCHESTRATION: **Heat**, Mistral, Aodh, Senlin, Zaqar, Blazar

COMPUTE

- VIRTUAL MACHINES: **Nova**
- CONTAINERS: Zun
- FUNCTIONS: Qinling

NETWORKING

- SDN: **Neutron**
- LOAD BALANCING: Octavia
- DNS: Designate

HARDWARE LIFECYCLE

- BARE METAL: **Ironic**
- ACCELERATORS: Cyborg

STORAGE

- OBJECT: **Swift**
- BLOCK: **Cinder**
- FILE: Manila

SHARED SERVICES: **Keystone**, **Placement**, **Glance**, Barbican, Searchlight, Karbor

OPENSTACK-USER

SDK: OpenStackClient, Python SDK

OPENSTACK-ADJACENTENABLERS

CONTAINER SERVICES: Kuryr

NFV: Tacker

OPENSTACK-OPERATIONS

MONITORING TOOLS: Ceilometer, Monasca, Panko

OPTIMIZATION / POLICY TOOLS: Watcher, Vitrage, Congress, Rally

BILLING / BUSINESS LOGIC: CloudKitty

MULTI-REGION TOOLS: Tricircle

OPENSTACK-LIFECYCLEMANAGEMENT

DEPLOYMENT / LIFECYCLE TOOLS: Kolla-Ansible, OpenStack-Charms, TripleO, Bifrost, OpenStack-Helm, OpenStack-Ansible, OpenStack-Chef

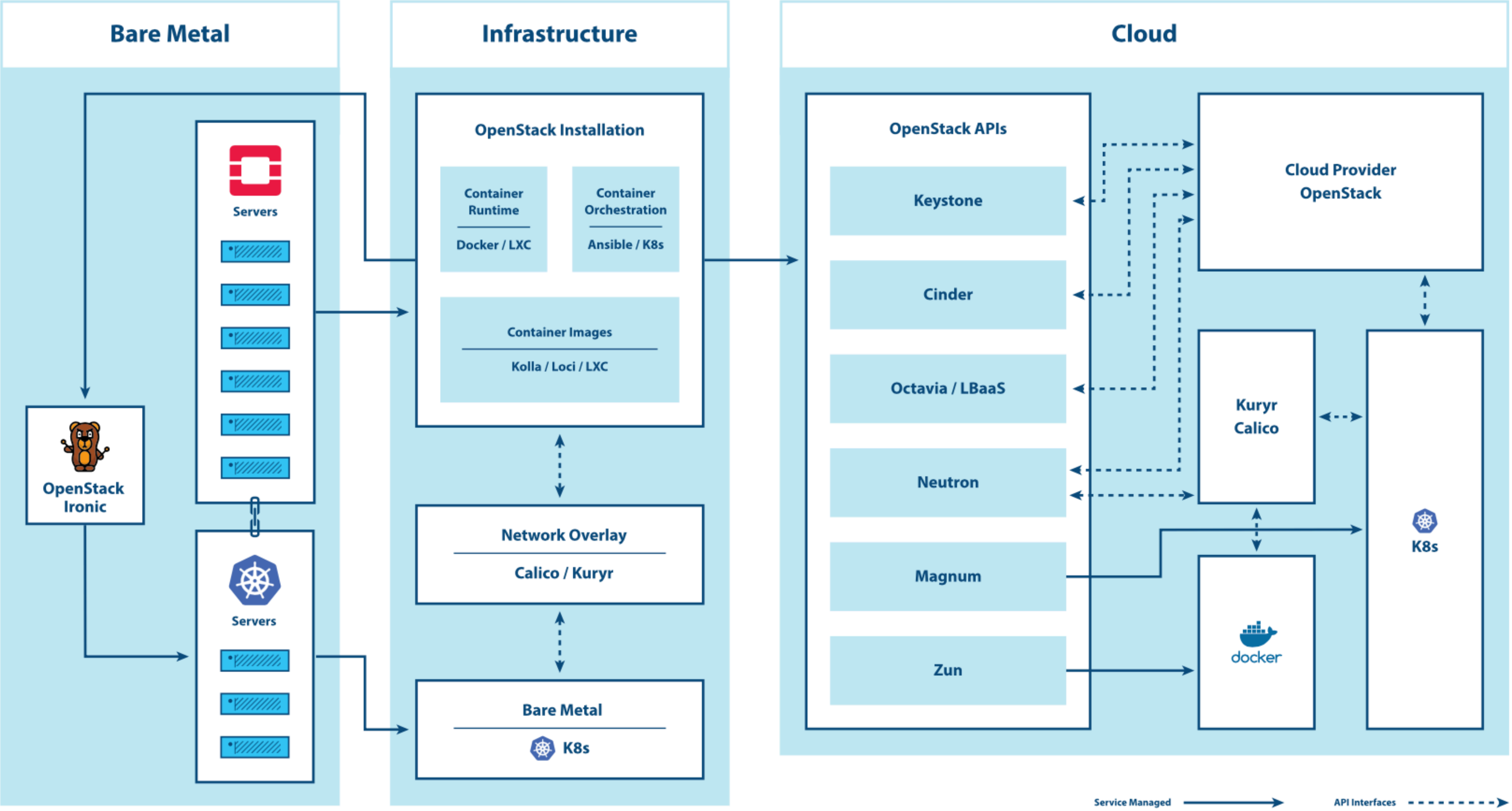
PACKAGING RECIPES FOR...: RPM, Puppet, OCI containers

Bold represents **Core Functionality**

Version 2019.06.02

openstack.

openstack.



Bare Metal

Servers

OpenStack Ironic

Servers

Infrastructure

OpenStack Installation

Container Runtime

Docker / LXC

Container Orchestration

Ansible / K8s

Container Images

Kolla / Loci / LXC

Network Overlay

Calico / Kuryr

Bare Metal

K8s

Cloud

OpenStack APIs

Keystone

Cinder

Octavia / LBaaS

Neutron

Magnum

Zun

Cloud Provider OpenStack

Kuryr Calico

K8s

docker

Service Managed ⟶

API Interfaces - - - - - - →

Cloud Provider OpenStack

Cloud Provider OpenStack

Repository hosts several Kubernetes and OpenStack integrations:

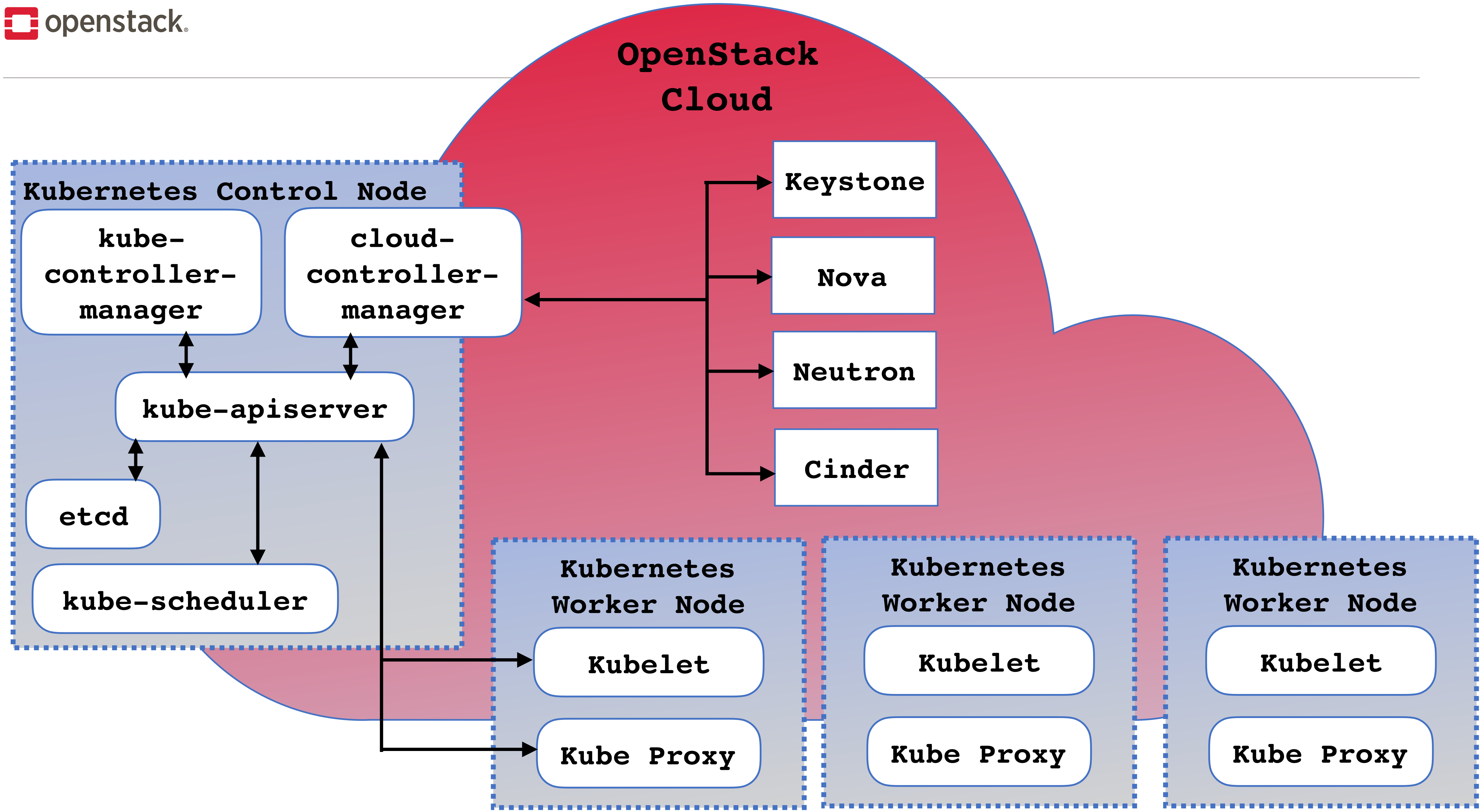
- Cloud Controller Manager for OpenStack
- Cinder CSI Plugin
- Octavia Ingress Controller
- Barbican KMS Plugin
- Keystone Authentication, Authorization

Cloud Provider OpenStack

- **Cloud Provider OpenStack**: It implements the cloud-provider interface, runs the cloud provider specific control loops that needs services/information from cloud provider(zones, node details, load balancer etc.)

- **Cinder CSI/ Manila CSI:** Cinder is the Storage Driver in OpenStack, Manila is the Â Shared Filesystem Service. Cinder and Manila CSI Drivers implements container storage Interface for the underlying services. Kubernetes specific deployment files are also maintained with the drivers. With the deprecations of intree volume drivers CSI Drivers has to be used with Kubernetes for volume managements.

- **Octavia Ingress Controller:** In kubernetes a Ingress Resource can be configured for external access to services, load balancing, filtering etc. for an Ingress Resource to work a ingress controller must be running in kubernetes cluster. In an OpenStack cloud it is provided by Octavia Service via Octavia Ingress Controller.

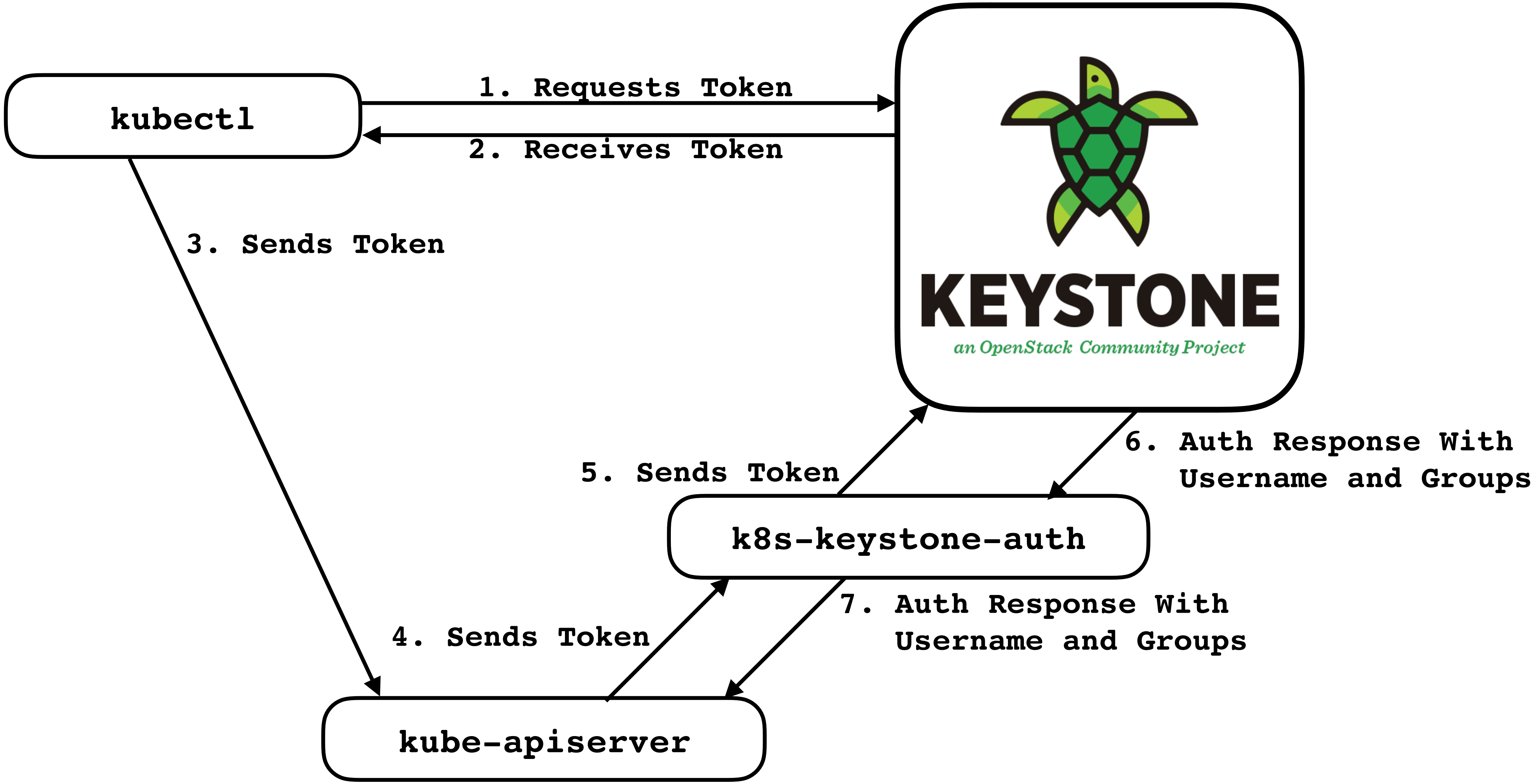

OpenStack Cloud
Keystone
Nova
Neutron
Cinder
Kubernetes Control Node
kube-controller-manager
cloud-controller-manager
kube-apiserver
etcd
kube-scheduler
Kubernetes Worker Node
Kubelet
Kube Proxy
Kubernetes Worker Node
Kubelet
Kube Proxy
Kubernetes Worker Node
Kubelet
Kube Proxy

Cloud Provider OpenStack

- **Keystone Auth and client keystone:** k8s-keystone-auth implements Kubernetes webhook authentication and authorization. Client keystone allows client side integrations with authentication using Keystone API, it allows exchange of user credentials with bearer token via kubectl.

  A typical flow would be:

  - The user issues a kubectl command.
  - Credential plugin prompts the user for Keystone credentials, exchanges credentials with external service for a token.
  - Credential plugin returns token to client-go, which uses it as a bearer token against the API server.
  - API server uses the webhook token authenticator to submit the token to the Keystone service.
  - Keystone service verifies the token and returns the user's username and groups.



kubectl

1. Requests Token

2. Receives Token

KEYSTONE

an OpenStack Community Project

3. Sends Token

5. Sends Token

6. Auth Response With Username and Groups

k8s-keystone-auth

4. Sends Token

7. Auth Response With Username and Groups

kube-apiserver

Cloud Provider OpenStack

- **Barbican KMS Plugin:** Kubernetes supports encrypting secret data at REST with various providers one of which is Key Manager Service.

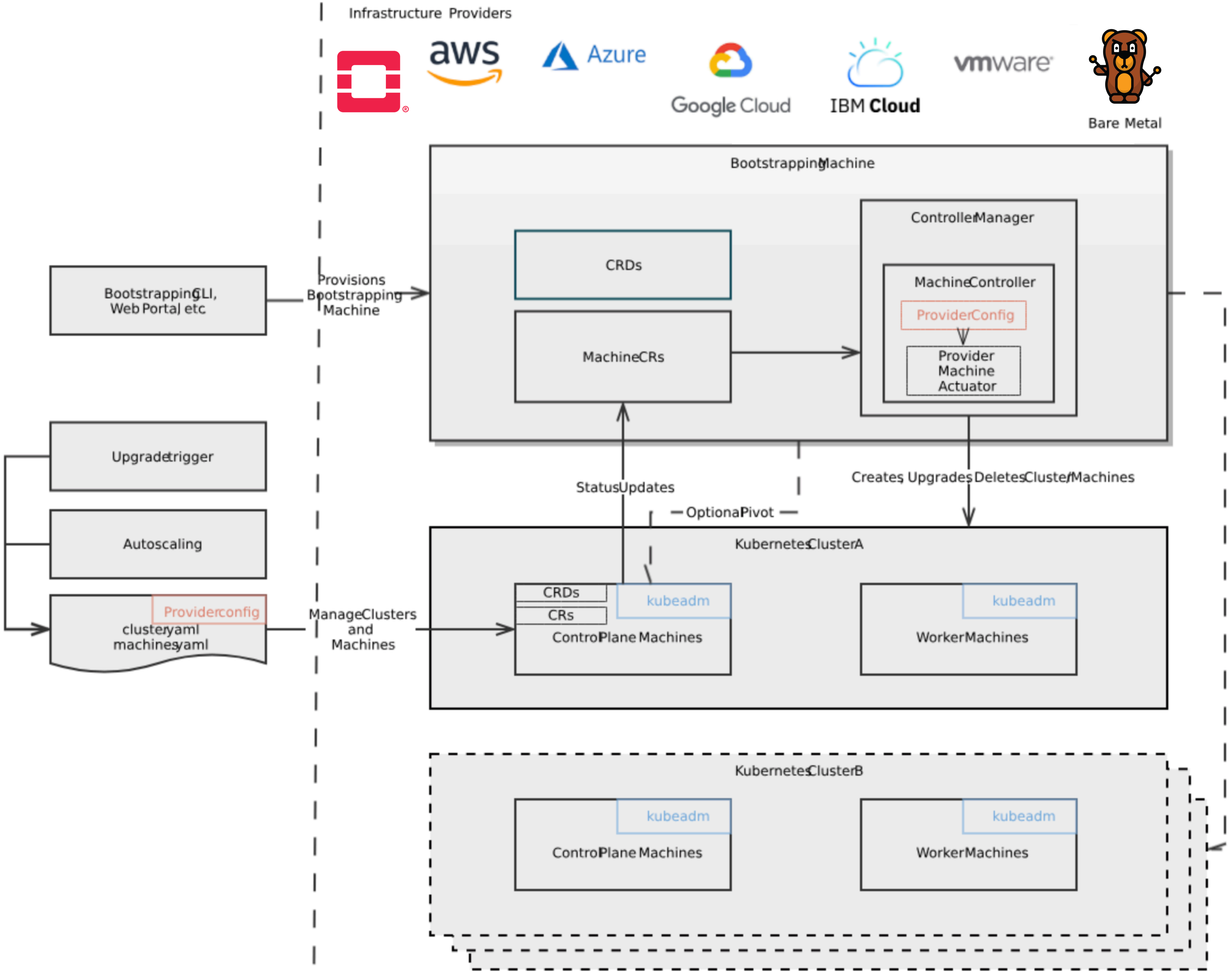
- The API Server requests the KMS plugin to encrypt/decrypt the secret data
- It encrypts/decrypt the data with the key from Barbican Service running in an OpenStack Cloud.
- Using *envelope encryption*, the Kubernetes API server requests to encrypt/decrypt the Data Encryption Key.

Cluster API Provider OpenStack

Cluster API

- “The Cluster API is a Kubernetes project to bring declarative, Kubernetes-style APIs to cluster creation, configuration, and management. It provides optional, additive functionality on top of core Kubernetes.”

- “Note that Cluster API effort is still in the prototype stage while we get feedback on the API types themselves. All of the code here is to experiment with the API and demo its abilities, in order to drive more technical feedback to the API design. Because of this, all of the prototype code is rapidly changing.”



Infrastructure Providers

Bare Metal

Bootstrapping Machine

CRDs

Machine CRs

Controller Manager

Machine Controller

Provider Config

Provider Machine Actuator

Bootstrapping CLI, Web Portal etc

Provisions Bootstrapping Machine

Upgrade trigger

Autoscaling

Provider config

cluster.yaml
machines.yaml

Manage Clusters and Machines

Status Updates

Optional Pivot

Creates, Upgrades, Deletes Cluster/Machines

Kubernetes Cluster A

CRDs

CRs

kubeadm

Control Plane Machines

kubeadm

Worker Machines

Kubernetes Cluster B

kubeadm

Control Plane Machines

kubeadm

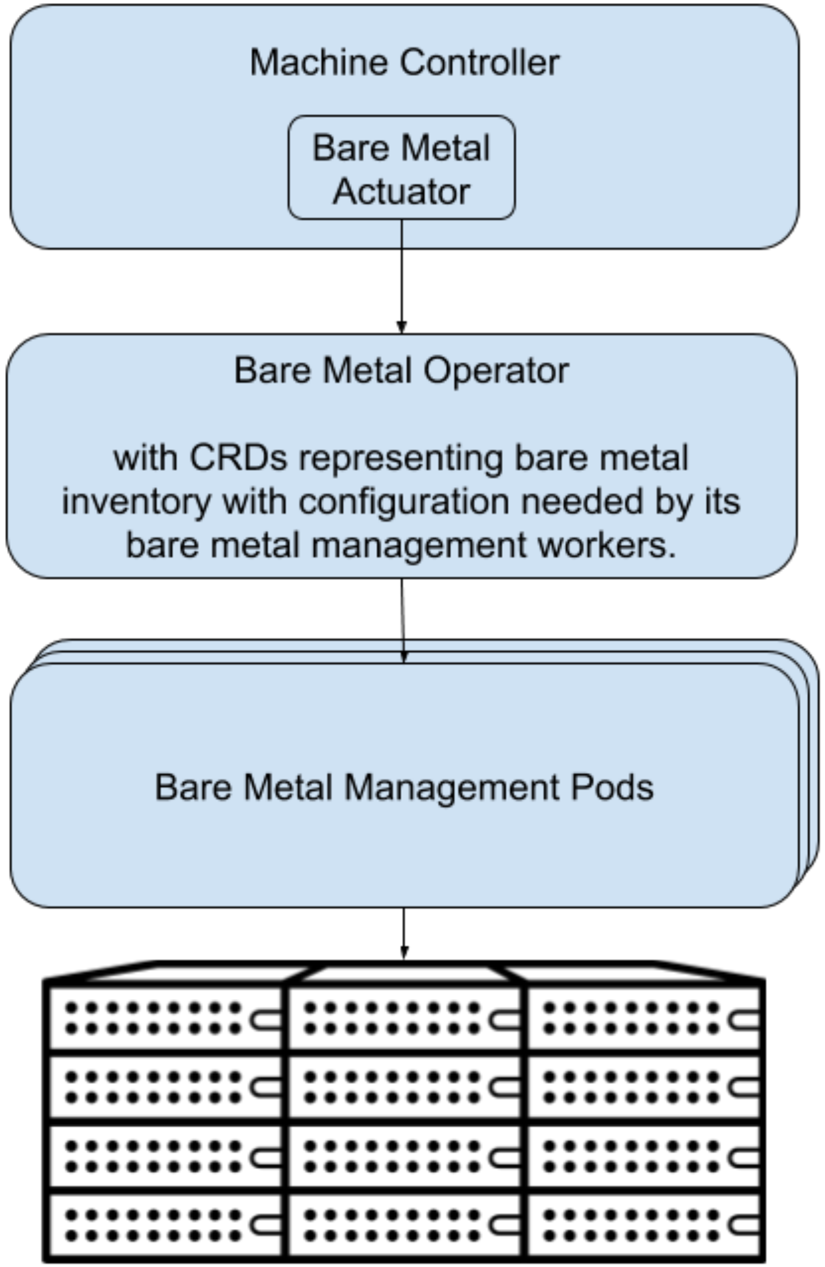
Worker Machines

cluster-api-provider-openstack

- It’s more of a concrete implementation than a true pluggable provider.
- Note that Cluster API effort is still in the prototype stage while we get feedback on the API types themselves. All of the code here is to experiment with the API and demo its abilities, in order to drive more technical feedback to the API design. Because of this, all of the prototype code is rapidly changing.
- Fast paced development effort, looking for more developers and more robust implementations.
- Cloud credentials are configured differently than for cloud-provider.
- Potential avenue for new work, abstracting credential loading.

Metal3 - Bare Metal Cluster API Provider

- The Metal^3 project (pronounced: Metal Kubed) exists to provide components that allow you to do bare metal host management for Kubernetes. Metal^3 works as a Kubernetes application, meaning it runs on Kubernetes and is managed through Kubernetes interfaces.



Machine Controller

Bare Metal Actuator

Bare Metal Operator

with CRDs representing bare metal inventory with configuration needed by its bare metal management workers.

Bare Metal Management Pods

SIG-OpenStack

Slack Channel: #SIG-OpenStack

Bi-Weekly Meetings: Wednesday @ 15:00 UTC

Primary GitHub Repository:
https://github.com/kubernetes/cloud-provider-openstack

openstack

Thank you!

Questions?

openstack   @OpenStack   openstack   OpenStackFoundation