FUJITSU
shaping tomorrow with you

VietKubers

Metal3 – Bare Metal Host Provision for Kubernetes

Speakers Kim Bao Long <longkb@vn.fujitsu.com>
Dao Cong Tien <tiendc@vn.fujitsu.com>

Address: Fujitsu Vietnam Limited
Vietkubers

Outline

- Metal3 101
- Architecture
- Deployment model
- Workflow
- Demo

Metal3 101

What is Metal3

- **Metal3** (pronounced **MetalKube**) is **bare metal provisioning** tool for **Kubernetes**.
  - Working as a Kubernetes application
  - Running on Kubernetes and is managed through Kubernetes's style interfaces

Metal3 community

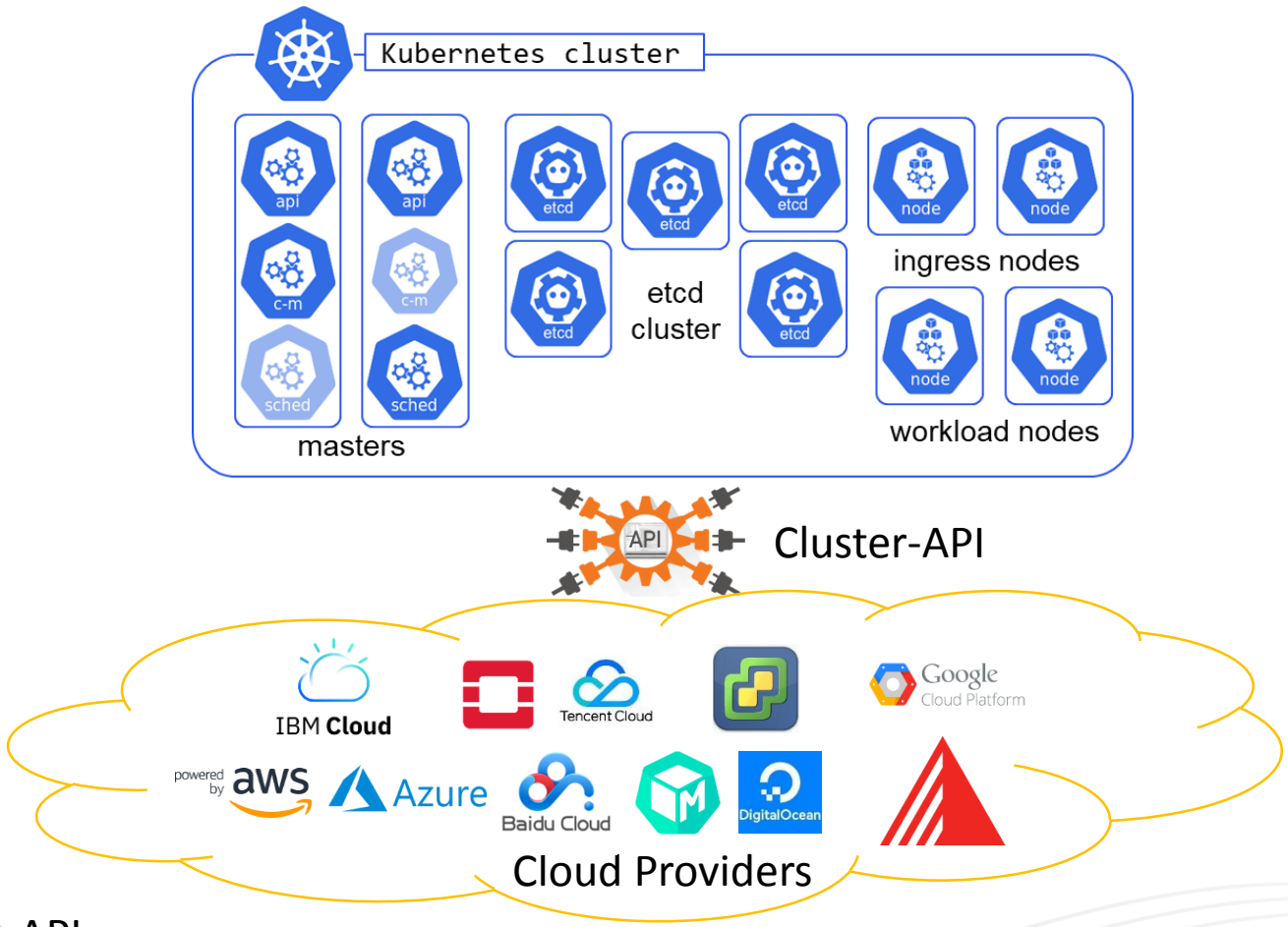
https://github.com/metal3-io

#cluster-api-baremetal channel on Kubernetes slack

https://groups.google.com/forum/#!forum/metal3-dev

Architecture

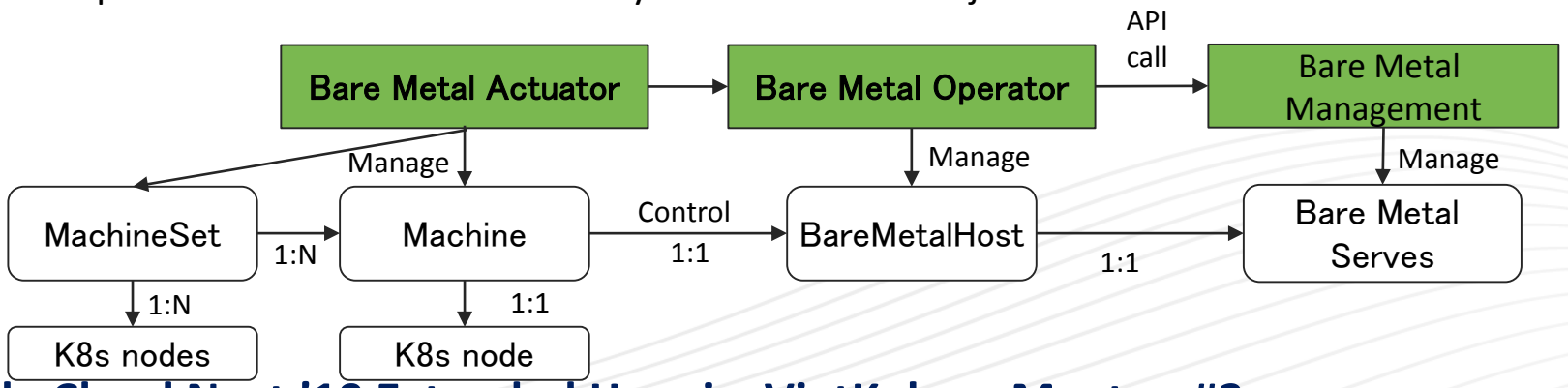
Architecture



- Cluster-API
  - Running K8s cluster on top of Cloud providers
  - Providing Kubernetes-style API for cluster creation, configuration and management.
  - Using Cloud Providers to provision support infrastructure

Architecture

- Baremetal Actuator
  - An implementation of **Cluster-API**
  - React with update from of Machine objects
    - Mapping 1:1 wit K8s node
    - Control the **BareMetalHost** resources
- Bare Metal Operator
  - Define/Manage **BareMetalHost** as **Custom Resource** in K8s
  - Handles reconciling the **BareMetalHost** with Ironic API calls underneath
- Bare Metal Management: Make use of **Ironic**
- Bare Metal Servers
  - The servers that will join into K8s cluster
  - Represented in K8s environment by **BareMetalHost** objects



Architecture

Ironic

- *OpenStack Ironic* grew out of the *Nova* project as a bare metal compute driver
- *Ironic* can also be used as a standalone bare-metal infrastructure management tool

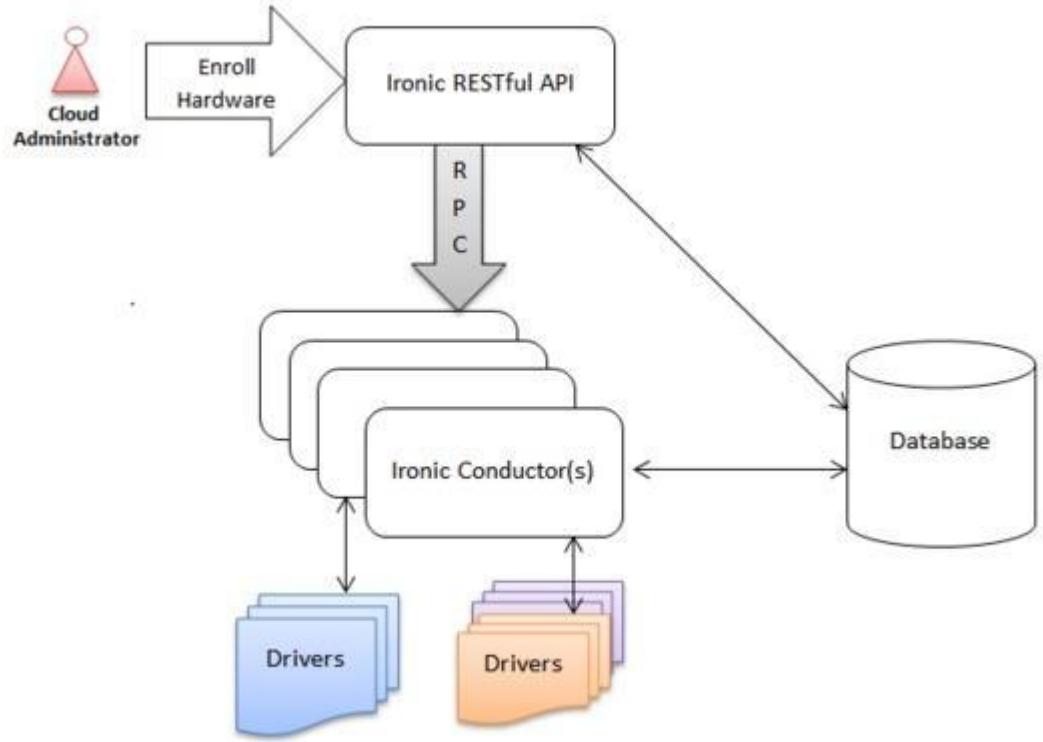
Architecture

Why need Ironic?

- *OpenStack Co*mpute service can provide computing infrastructure via virtual machines. This works for most of the cases. ***But…***
- Some applications require
  - high computing performance
  - low latency
  - hardware-level optimization
  - hardware compatibility
  - strong isolation from others
- And Ironic can help build *dynamic* K8s clusters

Architecture

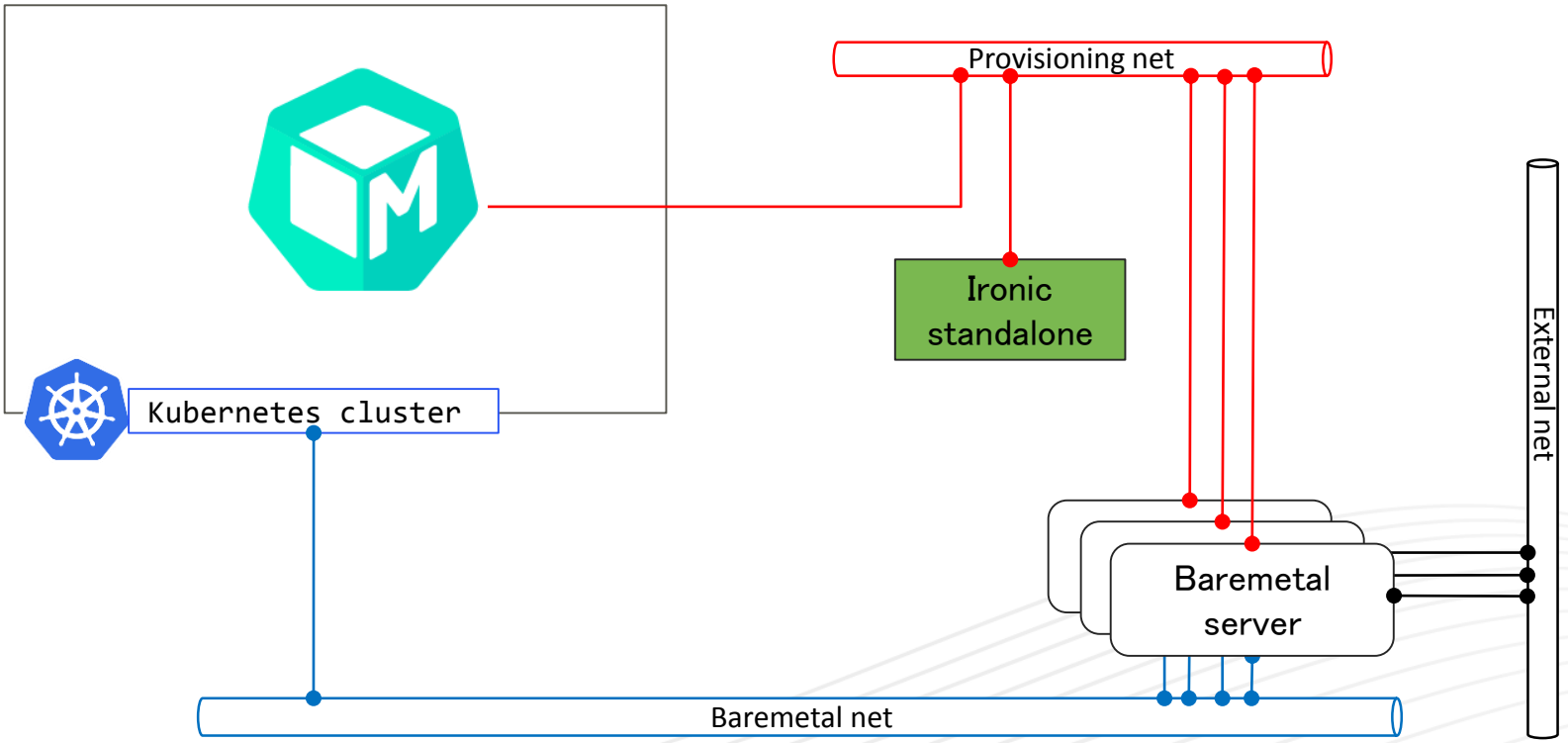
Ironic architecture



Deployment Model

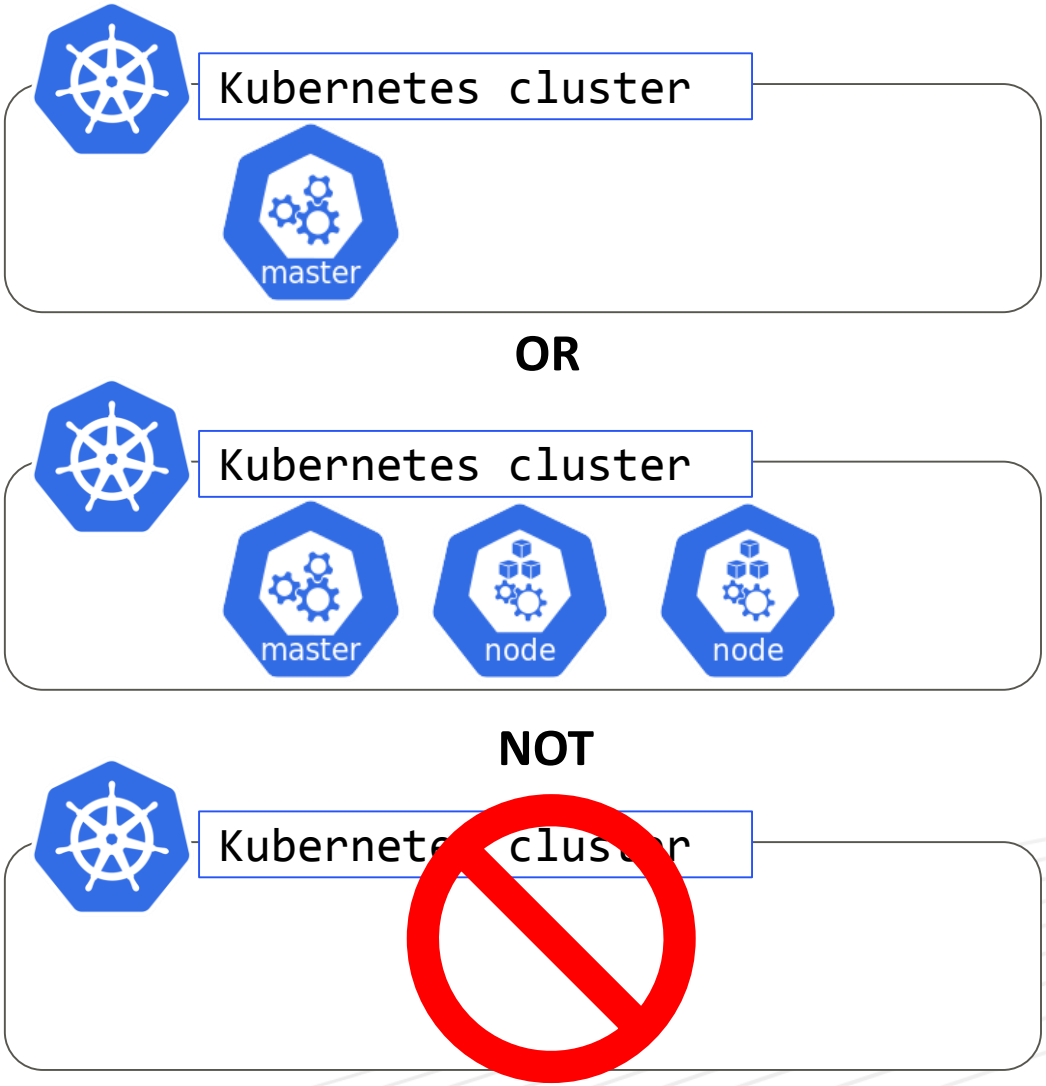
Deployment model

- Metal3 deployment on top of K8s cluster
- Ironic standalone
- Baremetal servers
- Networks



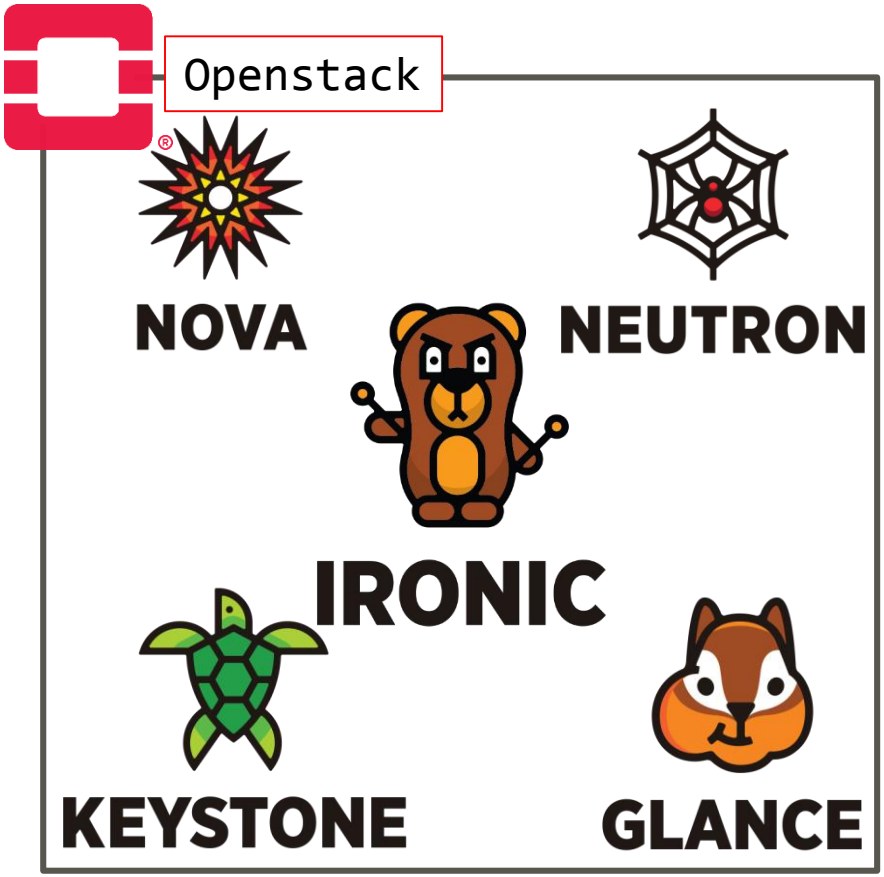
Deployment model

- Kubernetes cluster deployment with **kubeadm**

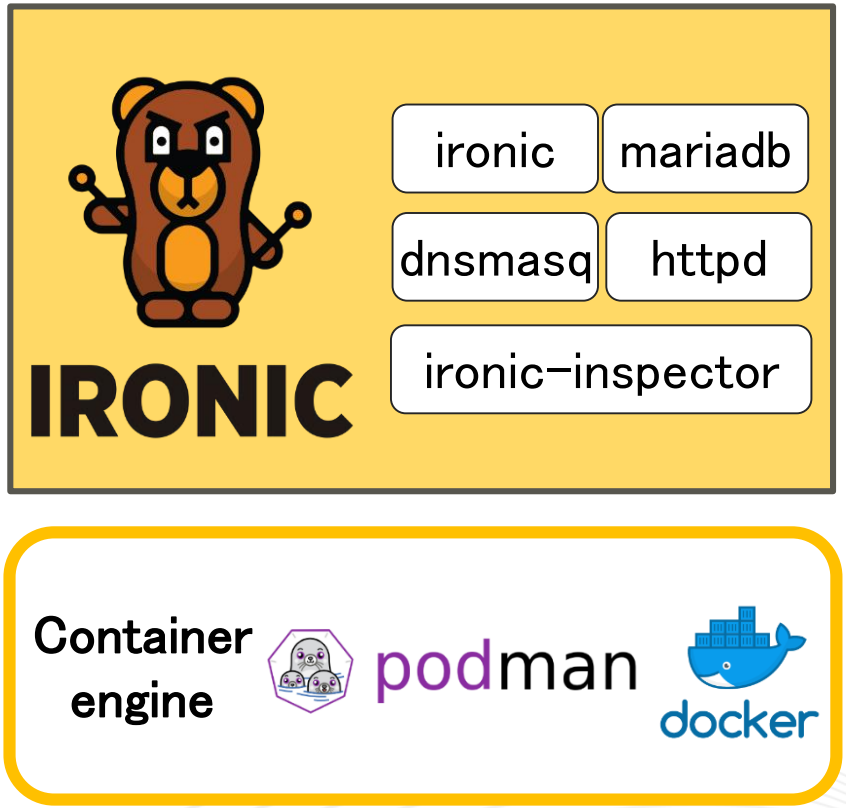


Deployment model

- Deploy Ironic standalone



Integration among Ironic
with other OpenStack services



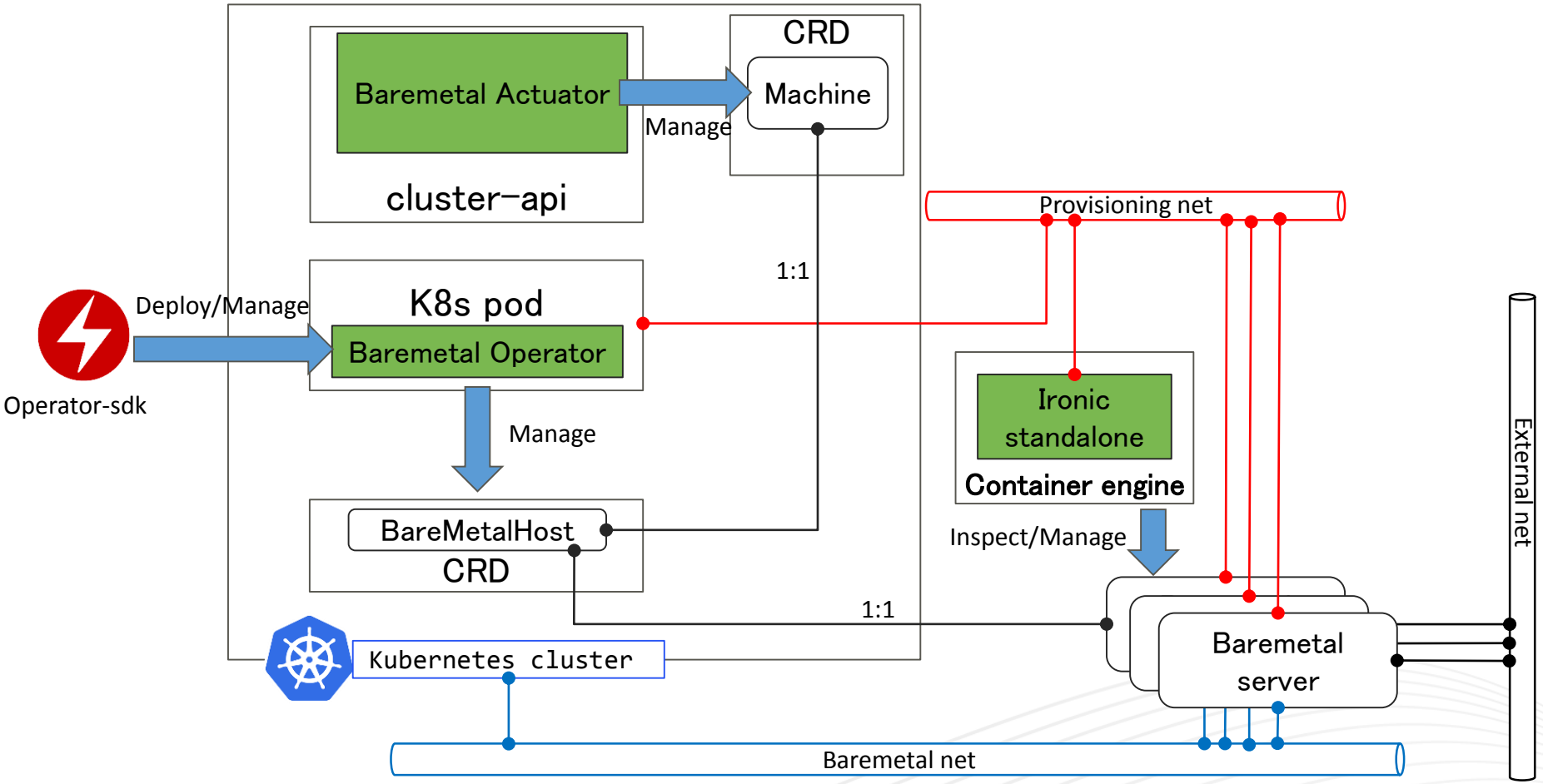
Ironic standalone

Deployment model

- Prepare power, cables, and BMC for your Bare Metal servers
  - BMC integration
  - Plug your cables
  - Plug your power cables

Deployment model

- Install metal3: Actuator, baremetal-operator, ironic standalone

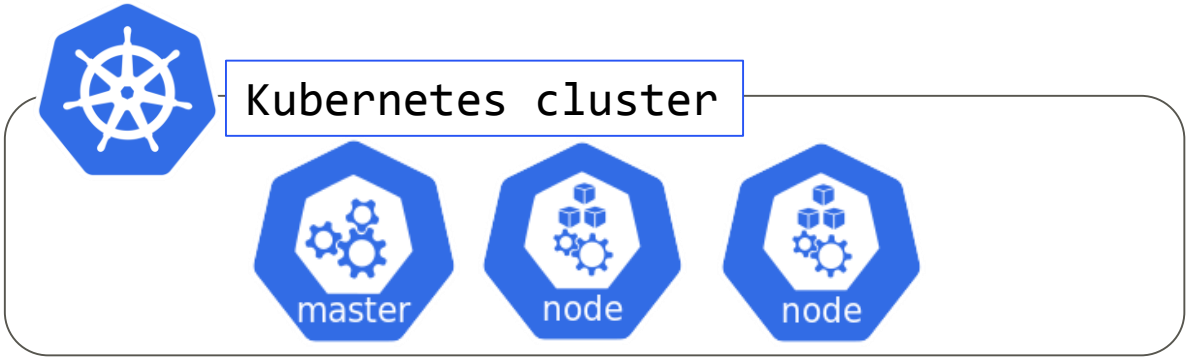


Workflow

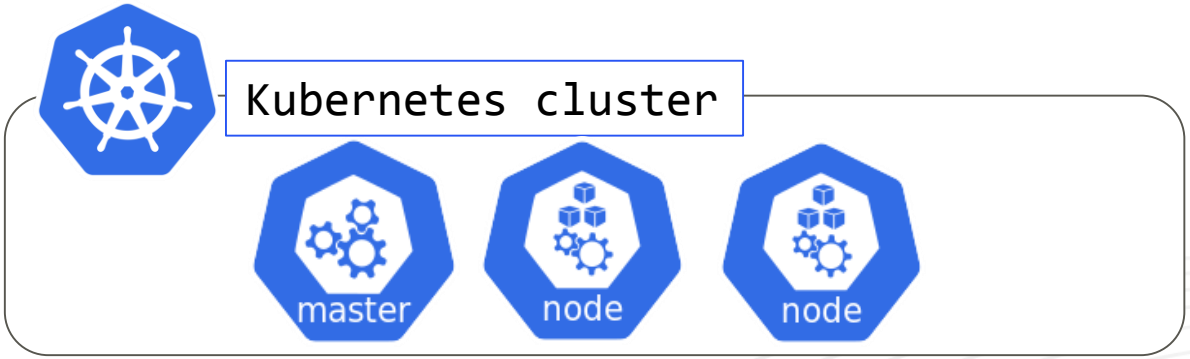
- Provisioning
- Deprovisioning

Workflow

- Provisioning - Growing the Cluster



- Deprovisioning - Shrinking the Cluster



Provisioning

- Admin register hosts to be part of the available inventory
  - Using **kubectl** to define a new **BareMetalHost** object with information to access the host's management controller (**BMC**)
- After registration, we have **1:1** mapping between **baremetal server** and **BareMetalHost** object (NIC, storage, CPUs, RAM, etc)
  - **BareMetalHost**: REGISTERING -> INSPECTING -> READY
  - **Ironic node:** Enroll → Verifying → Manageable → Inspecting → Inspect wait → Manageable

1:1

BaremetalHost

```
Hardware:
  Cpu:
    Count:       4
    Model:       Intel Core Processor (Broadwell, IBRS)
    Speed G Hz:  2.808
    Type:        x86_64
  Nics:
    Ip:          172.22.0.17
    Mac:         52:54:00:c3:40:2e
    Model:       0x1af4 0x0001
    Name:        eth0
    Network:     Pod Networking
    Speed Gbps:  0
    Ip:
    Mac:         52:54:00:fd:e6:78
    Model:       0x10ec 0x8139
    Name:        eth1
    Network:     Pod Networking
    Speed Gbps:  0
  Ram Gi B:      7
  Storage:
    Name:            /dev/vda
    Serial Number:
    Size Gi B:       50
    Type:            HDD
    Vendor:          0x1af4
    Name:            /dev/vdb
    Serial Number:
    Size Gi B:       50
    Type:            HDD
    Vendor:          0x1af4
  System Vendor:
    Manufacturer:     QEMU
    Product Name:     Standard PC (i440FX + PIIX, 1996)
    Serial Number:
```

Provisioning

- Admin create a new **Machine** object (a request) to grow the cluster
  - Create the **Machine** directly
  - Increase the replica count on a **MachineSet**
- The **Actuator** will match profile in **Machine** to available hosts with the same profile.
- The Actuator add information to the host to tell baremetal-operator which image to provision to the host.
- The host is provisioned and rebooted
- Kubelet connects to the cluster, triggering the creation of Node object.

Deprovisioning

- To shrink the cluster, admin can delete **Machine** object
  - Breaking the link to the host
  - The host is powered off and placed back into the inventory
- To completely remove a host, the **BareMetalHost** object is deleted

Demo

Model

- KVM machine with vBMC
- Kubernetes with 01 master node
- Metal3
  - Baremetal Operator
  - Baremetal Actuator – Cluster-API
- Ironic standalone
- Images:
  - Ironic Python Agent – IPA
  - Ubuntu 16.04 server: xenial-server-cloudimg-amd64-disk1.img

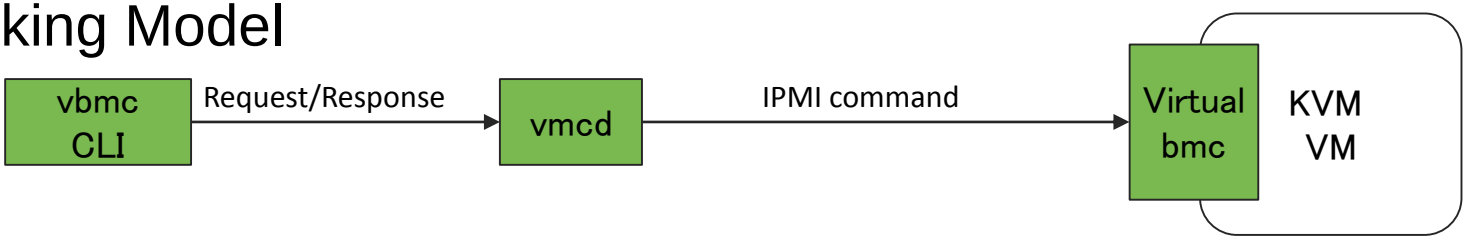
Baremetal simulation

- vBMC [1]
  - A tools simulates a Baseboard Management Controller (BMC) by exposing IPMI responder to the network and talking to libvirt at the host vBMC is running at to manipulate virtual machines which pretend to be bare metal servers.
- Supported IPMI commands

```
# Power the virtual machine on, off, graceful off, NMI and reset
ipmitool -I lanplus -U admin -P password -H 127.0.0.1 -p 6230 power on|off|soft|diag|reset

# Check the power status
ipmitool -I lanplus -U admin -P password -H 127.0.0.1 -p 6230 power status
```

- Working Model



[1] https://docs.openstack.org/virtualbmc/latest/

Baremetal simulation

- Setting up vBMC
  - In order to use VMs as hosts, they need to be connected to **vbmc**
  - Adding a new vBMC to control libvirt domain called **bm0** that will listen for IPMI commands on port **6230**

```
vbmc add bm0 --port 6230 --username admin --password password
```

  - Starting the **vbmc** to control libvirt domain **bm0**

```
vbmc start bm0
```

  - Check **vbmc** status

```
longkb@metal:~$ vbmc show bm0
+-----------------------+----------------+
| Property              | Value          |
+-----------------------+----------------+
| active                | True           |
| address               | ::             |
| domain_name           | bm0            |
| libvirt_sasl_password | ***            |
| libvirt_sasl_username | None           |
| libvirt_uri           | qemu:///system |
| password              | ***            |
| port                  | 6230           |
| status                | running        |
| username              | admin          |
+-----------------------+----------------+
```

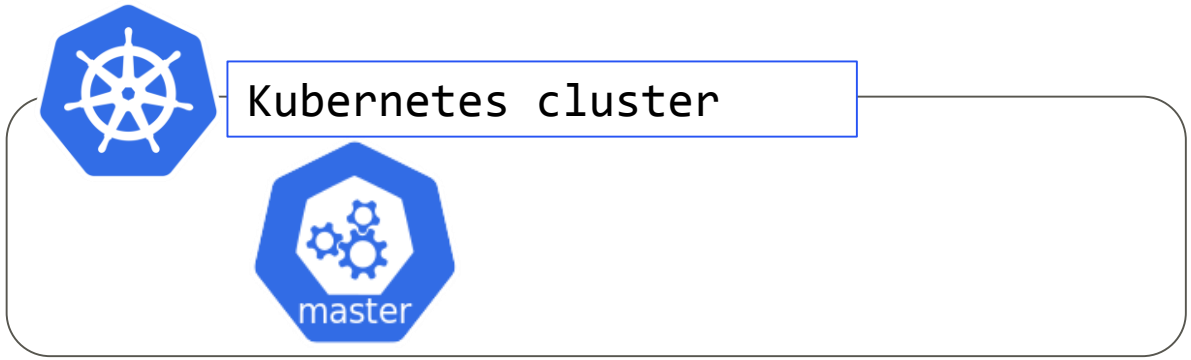
  - Power the virtual machine on, off

```
ipmitool -I lanplus -U admin -P password -H 127.0.0.1 -p 6230 power on|off
```

Demo

- Prepare 01 node Kubernetes cluster deployed with kubeadm

```
longkb@metal:~$ kubectl get nodes
NAME    STATUS   ROLES    AGE   VERSION
metal   Ready    master   19d   v1.14.2
```



- Prepare Ironic standalone with Docker

```
longkb@metal:~/go/src/github.com/metal3-io/baremetal-operator$ ironic node-list
The "ironic" CLI is deprecated and will be removed in the S* release. Please use the "openstack baremetal" CLI instead.
+------+------+---------------+-------------+--------------------+-------------+
| UUID | Name | Instance UUID | Power State | Provisioning State | Maintenance |
+------+------+---------------+-------------+--------------------+-------------+
+------+------+---------------+-------------+--------------------+-------------+
```

Demo

- Launch **baremetal-operator** and **cluster-api-provider-baremetal**

```
longkb@metal:~/go/src/github.com/metal3-io/baremetal-operator$ operator-sdk up local --namespace metal3
INFO[0000] Running the operator locally.
INFO[0000] Using namespace metal3.
{"level":"info","ts":1560418695.2099662,"logger":"cmd","msg":"Go Version: go1.12.5"}
{"level":"info","ts":1560418695.2100139,"logger":"cmd","msg":"Go OS/Arch: linux/amd64"}
{"level":"info","ts":1560418695.2100246,"logger":"cmd","msg":"Version of operator-sdk: v0.4.0+git"}
{"level":"info","ts":1560418695.2112577,"logger":"leader","msg":"Trying to become the leader."}
{"level":"info","ts":1560418695.2113082,"logger":"leader","msg":"Skipping leader election; not running in a cluster."}
{"level":"info","ts":1560418695.2517176,"logger":"cmd","msg":"Registering Components."}
{"level":"info","ts":1560418695.2520394,"logger":"kubebuilder.controller","msg":"Starting EventSource","controller":"metal3-baremetalhost-con
troller","source":"kind source: /, Kind="}
{"level":"info","ts":1560418695.2522407,"logger":"kubebuilder.controller","msg":"Starting EventSource","controller":"metal3-baremetalhost-con
troller","source":"kind source: /, Kind="}
{"level":"info","ts":1560418695.2523263,"logger":"cmd","msg":"Starting the Cmd."}
{"level":"info","ts":1560418695.3529656,"logger":"kubebuilder.controller","msg":"Starting Controller","controller":"metal3-baremetalhost-cont
roller"}
{"level":"info","ts":1560418695.4534373,"logger":"kubebuilder.controller","msg":"Starting workers","controller":"metal3-baremetalhost-control
ler","worker count":1}
```

```
longkb@metal:~/go/src/github.com/metal3-io/cluster-api-provider-baremetal$ make run
go generate ./pkg/... ./cmd/...
go fmt ./pkg/... ./cmd/...
go vet ./pkg/... ./cmd/...
go run ./cmd/manager/main.go
{"level":"info","ts":1560418752.804999,"logger":"baremetal-controller-manager","msg":"Found API group metal3.io/v1alpha1"}
{"level":"info","ts":1560418752.8348546,"logger":"kubebuilder.controller","msg":"Starting EventSource","controller":"machine-controller","sou
rce":"kind source: /, Kind="}
{"level":"info","ts":1560418752.9367163,"logger":"kubebuilder.controller","msg":"Starting Controller","controller":"machine-controller"}
{"level":"info","ts":1560418753.0370405,"logger":"kubebuilder.controller","msg":"Starting workers","controller":"machine-controller","worker
count":1}
```

Demo

- Prepare **YAML** file for **BaremetalHost**

```yaml
---
apiVersion: v1
kind: Secret
metadata:
  name: bm0-bmc-secret
type: Opaque
data:
  username: YWRtaW4=
  password: cGFzc3dvcmQ=

---
apiVersion: metal3.io/v1alpha1
kind: BareMetalHost
metadata:
  name: bm0
spec:
  online: true
  bmc:
    address: libvirt://192.168.122.1:6230/
    credentialsName: bm0-bmc-secret
  bootMACAddress: 52:54:00:01:92:68
```

Secret object that contains credentials for BMC access

MAC address of booting NIC

- Create BaremetalHost **bm0** with **bm0.yaml**

```
longkb@metal:~/go/src/github.com/metal3-io$ kubectl create -f bm0.yaml -n metal3
secret/bm0-bmc-secret created
baremetalhost.metal3.io/bm0 created
```

- Check **bm0** status

```
longkb@metal:~/go/src/github.com/metal3-io$ kubectl get baremetalhosts -n metal3
NAME   STATUS   PROVISIONING STATUS   MACHINE   BMC                         HARDWARE PROFILE   ONLINE   ERROR
bm0    OK       ready                           libvirt://192.168.122.1:6230/   libvirt            true
```

Demo

- Prepare **YAML** file for **Machine**

```yaml
apiVersion: "cluster.k8s.io/vlalphal"
kind: Machine
metadata:
  name: w0
  generateName: baremetal-machine-
spec:
  providerSpec:
    value:
      apiVersion: "baremetal.cluster.k8s.io/vlalphal"
      kind: "BareMetalMachineProviderSpec"
      image:
        url: http://172.22.0.1/images/xenial-server-cloudimg-amd64-disk1.img
        checksum: http://172.22.0.1/images/xenial-server-cloudimg-amd64-disk1.img.md5sum
      userData:
        name: w0-user-data
        namespace: metal3
```

Provisioning OS and it's own checksum

Secret that contains base64 encode of user-data for worker deployment

- Create Machine **w0** with **w0.yaml**

```
longkb@metal:~/go/src/github.com/metal3-io/metal3-dev-env$ kubectl create -f w0.yaml -n metal3
machine.cluster.k8s.io/w0 created
```

- Check **w0** status

```
longkb@metal:~/go/src/github.com/metal3-io$ kubectl get machine -n metal3
NAME   PROVIDERID   PHASE
w0
```

Demo

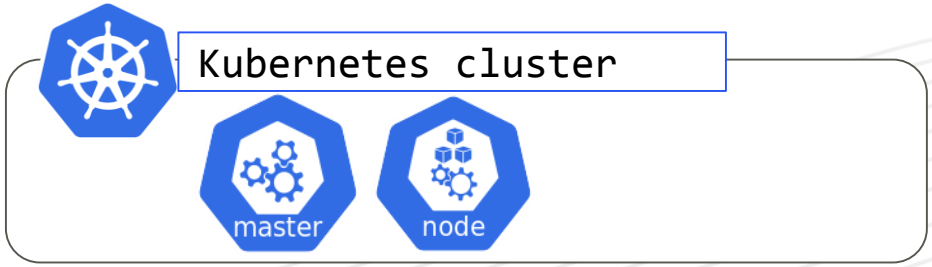
- Check **bm0** status

```
longkb@metal:~/go/src/github.com/metal3-io$ kubectl get baremetalhosts -n metal3
NAME   STATUS   PROVISIONING STATUS   MACHINE   BMC                              HARDWARE PROFILE   ONLINE   ERROR
bm0    OK       provisioned           w0        libvirt://192.168.122.1:6230/   libvirt            true
```

Demo

- Let's check kubenetes cluster => The new worker has appeared as a Node

```
longkb@metal:~/go/src/github.com/metal3-io$ kubectl get nodes
NAME      STATUS     ROLES    AGE     VERSION
metal     Ready      master   19d     v1.14.2
worker    NotReady   <none>   3m36s   v1.14.3
```




?
Q
&
&
A
!

Thank you!