ICN Metal3 Baremetal Operator

Goal:

This wiki describes the specifications for integration of Baremetal operator required for the Integrated Cloud Native Akraino project.

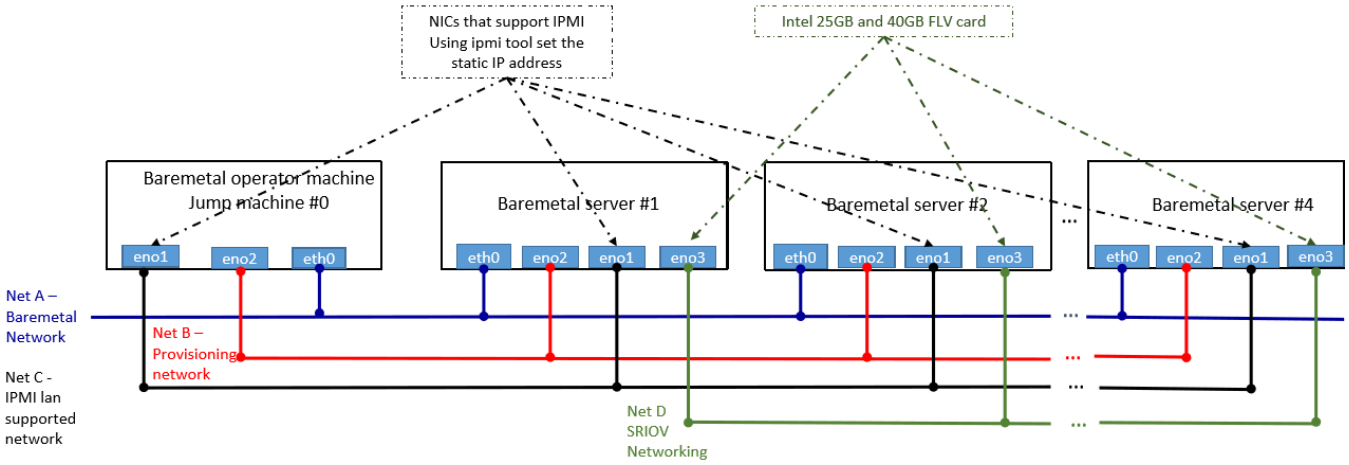
Overview of Baremetal Provisioning:

ICN architecture has a bootstrap cluster in all the edge location, this k8s cluster is used to do provisioning of compute nodes in the edge location.

Feature requirement of Baremetal provisioning

1. Bootstrap cluster should maintain the under cloud structure - adding new node and remove of node from the compute cluster
2. Bootstrap cluster should be aware of hardware platform awareness(HPA) of the compute node cluster to make intelligence decision of allocating the nodes in the compute cluster
3. Bootstrap cluster should keep the nodes in the ready state for provisioning and de-provisioning

Each bootstrap cluster has 3 distinguished networks one for bare-metal networking, provisioning network and ipmi lan network as show below:



Net A – Baremetal Network, lab networking for SSH, It is used as the control plane for Kubernetes, used by OVN and Flannel for the overlay networking with Internet access
Net B(Internal networking) – Provisioning Networking used by Ironic to do inspection
Net C(Internal networking) – IPMI lan to do IPMI protocols for the OS provisioning
Net D(Internal networking) - Data plan networking for the Akraino application. Using the SRIOV Networking and fiber cables

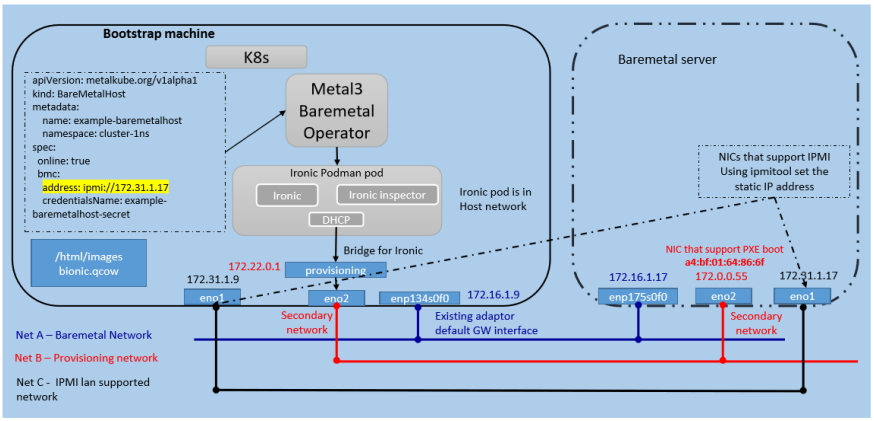
Bootstrap cluster will be in the jump machine, it has 3 interfaces, eno1 interface NIC support IPMI, eno2 for provisioning and eht0 for the bare-metal networking.

Baremetal operator:

ICN stack uses metal3 baremetal operator to do node provisioning in the bootstrap cluster. Baremetal operator runs as deployments in bootstrap cluster, gets the OS image details and baremetal server ipmi details in the edge location to do the provisioning. Baremetal operator uses Ironic as a provisioning agent.

Baremetal operator has the following components

- Baremeta host Custom resource definition(CRD)
- Baremetal host CRD controller
- Ironic
- Ironic Inspector
- Ironic internal DHCP server



Baremetal Host Custom Resource Definition(CRD)

The baremetal operator abstract the baremetal server hardware features and store the hardware profile details in the baremetal host. It hold key information such as CPU information, NIC, FPGA, QAT card and disk details, Baremetalhost CR act as template by a user to send the ipmi username and password encode as k8s secret to the Baremetal operator. And a refer to that K8s secret is refered as CredentialName in Baremetal operator API. Baremetal API defines various baremetal server details that are required to manage and provision the server.

BMC play a key in Baremetalhost CRD object. BMC spec has address and image and userData.

- BMC address define control plane that has a url to communicate to the BMC controller. ICN uses IPMI for communication with BMC controller and has Net C for that control plane traffic
- image field usually has image ins .img and .qcow2 format with their md5sum details
- User Data field is used give k8s secret that hold key information for the OS such as SSH authorization key, hostname or any start-up scripts

```
---
apiVersion: v1
kind: Secret
metadata:
  name: demo-bmc-secret
type: Opaque
data:
  username: cnllbGVzd2E=
  password: Y2hhbmdlbWUx

---
apiVersion: metal3.io/v1alpha1
kind: BareMetalHost
metadata:
  name: demo
spec:
  online: true
  bmc:
    address: ipmi://172.31.1.17
    credentialsName: demo-bmc-secret
  image:
    url: "http://172.22.0.1/images/bionic-server-cloudimg-amd64.img"
    checksum: "http://172.22.0.1/images/bionic-server-cloudimg-amd64.md5sum"
  userData:
    name: demo-user-data
    namespace: metal3
```

Baremetal host CRD controller

Baremetal host controller is CRD implementation that list and watch for the creation of the BMH CR in the booststrap cluster. Once the CR is created or applied with patches, this event triggers the CRD controller and invoke the ironic with ipmi address, image and userdata. In order to run the baremetal CRD controller to communicate with Ironic, user has pass down following information.

- Deploy ramdisk url to Ironic agent
- Kernel details to deploy ramdisk
- Ironic endpoint url
- Ironic inspector url

Baremetal CRD controller basically act as a client with Ironic endpoint and Ironic inspector endpoint to send the crd information and to retrieve the hardware details from the Ironic inspector to store the details in the etcd controller

Ironic

Ironic as standalone open source has a lot of capability to control a remote BMC in a server. In ICN architecture, Ironic boot the baremetal server through PXE, it order to assist it. We have a lightweight DHCP server running in the provisioning network. Currently, in metal3 project, the provisioning network is required to boot the ramdisk and receive the hardware details from ramdisk to the ironic inspector. Ram disk gives the information regarding the PXE boot information to the ironic inspector and Ironic uses this information to initiates the deployment of OS images

ICN has L2 network for the BMC connectivity using net-c, DHCP server communication using provisioning network net-a.

Provisioning above Baremetalhost CRD controller can be directly debugged with openstack as follows

```
# export OS_TOKEN=fake-token
# export OS_URL=http://localhost:6385/
# openstack baremetal node list
+--------------------------------------+------+--------------------------------------+-------------
+--------------------+-------------+
| UUID                                 | Name | Instance UUID                        | Power State |
Provisioning State | Maintenance |
+--------------------------------------+------+--------------------------------------+-------------
+--------------------+-------------+
| 36b462c1-02a0-499e-b4f1-cc6087a1c574 | demo | 36b462c1-02a0-499e-b4f1-cc6087a1c574 | power on    |
active             | False       |
+--------------------------------------+------+--------------------------------------+-------------
+--------------------+-------------+
```