ICN Installation Guide

- 1 Introduction
- 2 License
- 3 Deployment Architecture
- 4 Pre-installation Requirements
    - 4.1.1 Local controller:
    - 4.1.2 Compute k8s cluster:
  - 4.2 Hardware Requirements
    - 4.2.1 Minimum Hardware Requirement:
    - 4.2.2 Recommended Hardware Requirements
  - 4.3 Software Prerequisites
  - 4.4 Database Prerequisites
  - 4.5 Other Installation Requirements
    - 4.5.1 Jump Host Requirements <--- SM comment "Is jump host same as jump server? if yes, better to use same nomenclature thru out the doc" >>>
      - 4.5.1.1 Jump server Hardware Requirements
      - 4.5.1.2 Jump server Software Requirements:
    - 4.5.2 Network Requirements
    - 4.5.3 Bare Metal Node Requirements
    - 4.5.4 Compute k8s cluster:
      - 4.5.4.1 Compute server Hardware Requirements:
      - 4.5.4.2 Compute server Software Requirements:
    - 4.5.5 Execution Requirements (Bare Metal Only)
- 5 Installation High-Level Overview
  - 5.1 Baremetal Deployment Guide
    - 5.1.1 Install Bare Metal Jump Host <--- SM comment "Is Jump Host same as local Controller? The high level installation overview never mentions installing jump host" >>>
      - 5.1.1.1 Creating a Node Inventory File
        - 5.1.1.1.1 Preconfiguration for the local controller. <--- SM comment "local Controller or jump host? Please use one nomenclature. changing names is confusing" >>>
        - 5.1.1.1.2 Local controller Metal3 configuration Reference: <--- SM comment "Since we are doing a 3 compute node installation; it will be better to include 3 nodes in the json sample file above." >>>
      - 5.1.1.2 Creating the Settings Files
        - 5.1.1.2.1 Local controller network configuration Reference:
      - 5.1.1.3 Running
  - 5.2 Virtual Deployment Guide
    - 5.2.1 Standard Deployment Overview
    - 5.2.2 Snapshot Deployment Overview
    - 5.2.3 Special requirements for virtual deployment
      - 5.2.3.1 Install Jump Host
      - 5.2.3.2 Verifying the Setup - VMs
- 6 Verifying the Setup
- 7 Developer Guide and Troubleshooting
  - 7.1 Utilization of Images
  - 7.2 Post-deployment Configuration
  - 7.3 Debugging Failures
  - 7.4 Reporting a Bug
- 8 Uninstall Guide
  - 8.1 Baremetal deployment
  - 8.2 Virtual deployment
- 9 Troubleshooting
  - 9.1 Error Message Guide
- 10 Maintenance
  - 10.1 Blue Print Package Maintenance
  - 10.2 Software maintenance
  - 10.3 Hardware maintenance
  - 10.4 BluePrint Deployment Maintenance
- 11 Frequently Asked Questions
- 12 License
- 13 References
- 14 Definitions, acronyms and abbreviations

Introduction

ICN strives to automate the process of installing the local cluster controller to the greatest degree possible–"zero touch installation". Most of the work is done simply by booting up the jump host (Local Controller). Once booted, the controller is fully provisioned and begins to inspect and provision baremetal servers, until the cluster is entirely configured.

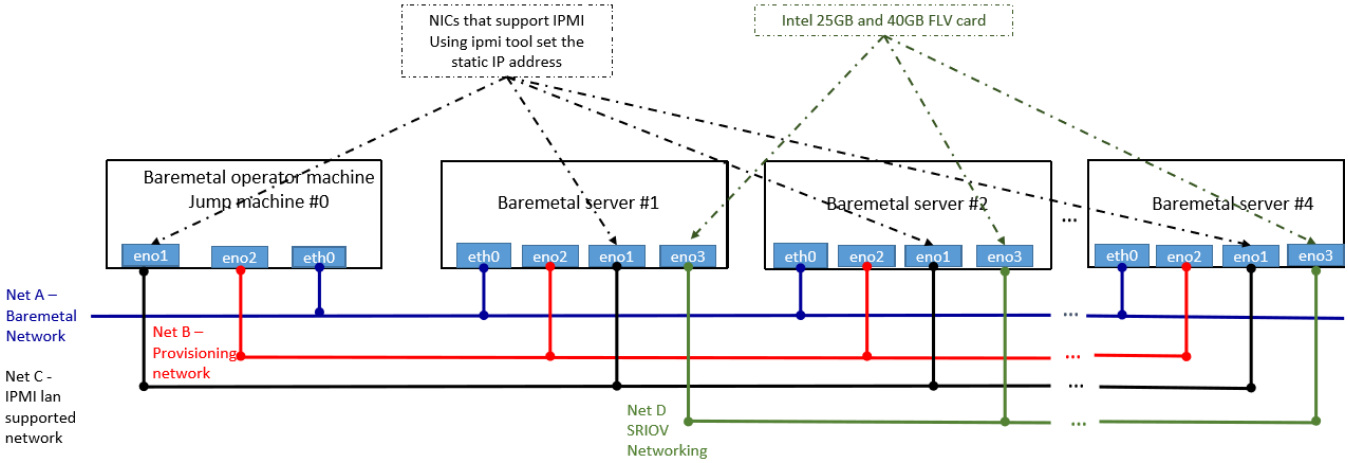
This document show step by step to configure the network, and deployment architecture for ICN BP.

License

Apache license v2.0

Deployment Architecture

The local controller is provisioned with the Metal3 Baremetal Operator and Ironic, which enable provisioning of Baremetal servers. The controller has three network connections to the baremetal servers: network A connects baremetal servers, network B is a private network used for provisioning the baremetal servers, and network C is the IPMI network, used for control during provisioning. In addition, the baremetal hosts connect to the network D, the SRIOV network.



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

In some deployment model, you can combine Net C and Net A to be the same networks, but developer should take care of IP Address management between Net A and IPMI address of the server.

Pre-installation Requirements

There are two main components in ICN Infra local controller - Local controller and Compute K8s cluster

Local controller:

Local controller will reside in the jump server to run the Metal3 operator, Binary provisioning agent operator and Binary provisioning agent restapi controller.

Compute k8s cluster:

Compute K8s cluster will actually run the workloads and it installed on Baremetal nodes

Hardware Requirements

Minimum Hardware Requirement:

All-in-one VM based deployment required at least 32 GB RAM and 32 CPU servers

Recommended Hardware Requirements

Recommended Hardware requirements 64GB Memory and 32 CPU servers, SRIOV network cards

Software Prerequisites

Jump server required to be pre-installed with Ubuntu 18.04

Database Prerequisites

No Prerequisites for ICN BP

Other Installation Requirements

Jump Host Requirements <--- SM comment "Is jump host same as jump server? if yes, better to use same nomenclature thru out the doc" >>>

Jump server required to be installed with Ubuntu 18.04 server, and have 3 distinguished networks as shown in figure 1 **<--- SM comment "This is redundant, is captured under HW and SW requirements" >>>**

Jump server Hardware Requirements

Local controller: at least three network interfaces.

Baremetal hosts: four network interfaces, including one IPMI interface.

Four or more hubs, with cabling, to connect four networks.

Hostname	CPU Model	Memory	Storage	1GbE: NIC#, VLAN, (Connected extreme 480 switch)	10GbE: NIC# VLAN, Network (Connected with IZ1 switch)
Jump	Intel 2xE5-2699	64GB	3TB (Sata) 180 (SSD)	IF0: VLAN 110 (DMZ) IF1: VLAN 111 (Admin)	IF2: VLAN 112 (Private) **VLAN 114** (Management) IF3: VLAN 113 (Storage) **VLAN 1115** (Public)

<--- SM comment "DMZ may not mean anything to customer. Lets be consistent. Can we replace IF0, IF1 etc with Net A, Net BThis is redundant, is captured under HW and SW requirements" >>>

Jump server Software Requirements:

ICN R2 release support Ubuntu 18.04 - ICN BP install all required software during "make install"

Network Requirements

Please refer the figure 1, for all the network requirement in ICN BP

Please make sure you have 3 distinguished networks - Net A, Net B and Net C as mentioned in figure 1. Local controller uses the Net B and Net C to provision the Baremetal servers to do the OS provisioning.

Bare Metal Node Requirements

Compute k8s cluster:

Compute server Hardware Requirements:

(Tested as below)

Hostname	CPU Model	Memory	Storage	1GbE: NIC#, VLAN, (Connected extreme 480 switch)	10GbE: NIC# VLAN, Network (Connected with IZ1 switch)
node1	Intel 2xE5-2699	64GB	3TB (Sata) 180 (SSD)	IF0: VLAN 110 (DMZ) IF1: VLAN 111 (Admin)	IF2: VLAN 112 (Private) **VLAN 114** (Management) IF3: VLAN 113 (Storage) **VLAN 1115** (Public)
node2	Intel 2xE5-2699	64GB	3TB (Sata) 180 (SSD)	IF0: VLAN 110 (DMZ) IF1: VLAN 111 (Admin)	IF2: VLAN 112 (Private) **VLAN 114** (Management) IF3: VLAN 113 (Storage) **VLAN 1115** (Public)
node3	Intel 2xE5-2699	64GB	3TB (Sata) 180 (SSD)	IF0: VLAN 110 (DMZ) IF1: VLAN 111 (Admin)	IF2: VLAN 112 (Private) **VLAN 114** (Management) IF3: VLAN 113 (Storage) **VLAN 1115** (Public)

Compute server Software Requirements:

The local controller will install all the software in compute servers right from OS, the software required to bring up the Kubernetes cluster **<--- SM comment "local controller installs OS on compute servers?" >>>**

Execution Requirements (Bare Metal Only)

ICN BP check all the precondition and execution requirements for both Baremetal and VM deployment **<--- SM comment "Heading indicates only Baremetal but the text here mentions both types of deployments including VM" >>>**

Installation High-Level Overview

Installation is two-step process and everything starts with one command "make install"

- Installation of the local controller.
- Installation of Compute cluster.

Baremetal Deployment Guide

Install Bare Metal Jump Host <--- SM comment "Is Jump Host same as local Controller? The high level installation overview never mentions installing jump host" >>>

Creating a Node Inventory File

Preconfiguration for the local controller. <--- SM comment "local Controller or jump host? Please use one nomenclature. changing names is confusing" >>>

User required to provide the IPMI information of the edge server they required to connect to the local controller by editing node JSON sample file in the directory icn/deploy/metal3/scripts/nodes.json.sample as below. If you want to increase nodes, just add another array

node.json.sample

```
{
  "nodes": [
    {
      "name": "edge01-node01",
      "ipmi_driver_info": {
        "username": "admin",
        "password": "admin",
        "address": "10.10.10.11"
      },
      "os": {
        "image_name": "bionic-server-cloudimg-amd64.img",
        "username": "ubuntu",
        "password": "mypasswd"
      }
    },
     {
      "name": "edge01-node02",
      "ipmi_driver_info": {
        "username": "admin",
        "password": "admin",
        "address": "10.10.10.12"
      },
      "os": {
        "image_name": "bionic-server-cloudimg-amd64.img",
        "username": "ubuntu",
        "password": "mypasswd"
      }
    }
  ]
}
```

Local controller Metal3 configuration Reference: **<--- SM comment "Since we are doing a 3 compute node installation; it will be better to include 3 nodes in the json sample file above." >>>**

- *node*: The array of nodes required to add to local controller
- *name*: Name of the Baremetal to be provisioned by Metal3, and this name will be the hostname for the machine, once it is provisioned
- *ipmi_driver_info*: IPMI driver info is a json field, currently holds the IPMI information required for Ironic to send the IPMI tool command
    - *username*: BMC username required to be provided for Ironic
    - *password*: BMC password required to be provided for Ironic
    - *address*: BMC server IPMI LAN IP address
- *os*: Baremetal machine OS information is a json field, currently holds the image name to be provisioned, username name and password for the login.
    - *image_name*: images name should be in qcow2 format
    - *username*: login username for the OS provisioned
    - *password*: login password for the OS provisioned

Creating the Settings Files

Local controller network configuration Reference:

User will find the network configuration file named as "user_config.sh" in the icn parent folder

user_config.sh

```
#!/bin/bash

#Local controller - Bootstrap cluster DHCP connection
#BS_DHCP_INTERFACE defines the interfaces, to which ICN DHCP deployment will bind
#e.g. export BS_DHCP_INTERFACE="ens513f0"
export BS_DHCP_INTERFACE=

#BS_DHCP_INTERFACE_IP defines the IPAM for the ICN DHCP to be managed.
#e.g. export BS_DHCP_INTERFACE_IP="172.31.1.1/24"
export BS_DHCP_INTERFACE_IP=

#Edge Location Provider Network configuration
#Net A - Provider Network
#If provider having specific Gateway and DNS server details in the edge location
#export PROVIDER_NETWORK_GATEWAY="10.10.110.1"
export PROVIDER_NETWORK_GATEWAY=
#export PROVIDER_NETWORK_DNS="8.8.8.8"
export PROVIDER_NETWORK_DNS=

#Ironic Metal3 settings for provisioning network
#Interface to which Ironic provision network to be connected
#Net B - Provisioning Network
#e.g. export IRONIC_INTERFACE="enp4s0f1"
export IRONIC_INTERFACE=

#Ironic Metal3 setting for IPMI LAN Network
#Interface to which Ironic IPMI LAN should bind
#Net C - IPMI LAN Network
#e.g. export IRONIC_IPMI_INTERFACE="enp4s0f0"
export IRONIC_IPMI_INTERFACE=

#Interface IP for the IPMI LAN, ICN verfiy the LAN Connection is active or not
#e.g. export IRONIC_IPMI_INTERFACE_IP="10.10.110.20"
#Net C - IPMI LAN Network
export IRONIC_IPMI_INTERFACE_IP=
```

Running

After configuring, Node inventory file and setting files. Please run "make install" from the ICN parent directory as shown below:

icn terminal

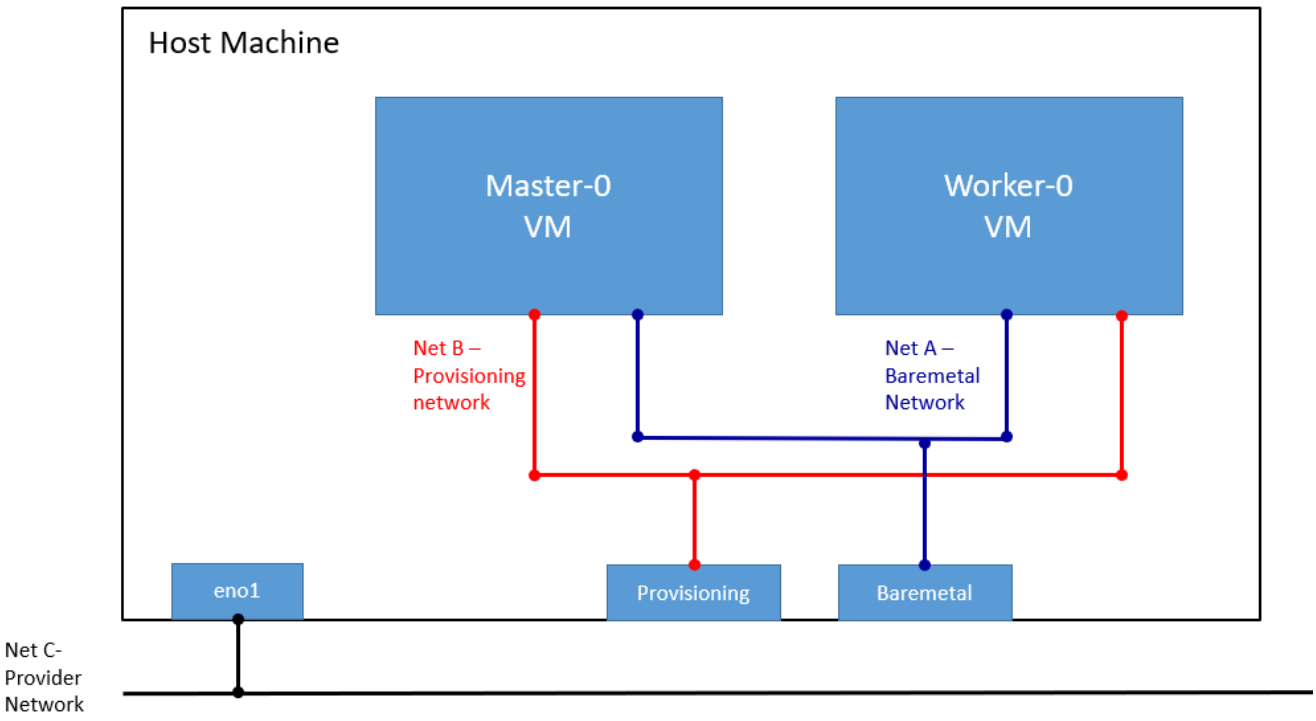
```
root@pod11-jump:# git clone "https://gerrit.akraino.org/r/icn"
Cloning into 'icn'...
remote: Counting objects: 69, done
remote: Finding sources: 100% (69/69)
remote: Total 4248 (delta 13), reused 4221 (delta 13)
Receiving objects: 100% (4248/4248), 7.74 MiB | 21.84 MiB/s, done.
Resolving deltas: 100% (1078/1078), done.
root@pod11-jump:# cd icn/
root@pod11-jump:# vim Makefile
root@pod11-jump:# make install
```

Following steps occurs once the "make install" command is given.

1. All the software required to run the bootstrap cluster is being downloaded and installed
2. Kubernetes cluster to maintain the Bootstrap cluster and all the servers in the edge location is installed
3. Metal3 specific network configuration such as local DHCP server networking for each edge location, Ironic networking for both provisioning network and IPMI LAN network are identified and created
4. Metal3 is launched with IPMI configuration as configured in "user_config.sh" and provision the Baremetal servers using IPMI LAN network. For more information refer the Debugging Failure section
5. Metal3 launch verification run without a timeout of 60 mins, by checking the status of all the servers being provisioned or not,
    a. All servers are provisioned parallelly. For example, if your deployment is having 10 servers in the edge location. All the 10 servers are provisioned at the same time
    b. Metal3 launch verification take care of checking all the servers are provisioned, the network interfaces are up and provisioned with a provider network gateway and DNS server
    c. Metal3 launch verification checks the status of all servers given in user_config.sh to make sure all the servers are provisioned. For example, if 8 servers are provisioned and 2 servers are not provisioned, Launch verification make sure all servers are provisioned before launch Kubernetes clusters on those servers
6. BPA Baremetal components are invoked with the mac address of the servers provisioned by metal3, BPA Baremetal components decide the cluster size and also the number of clusters required in the edge location
7. BPA Baremetal runs the containerized KUD as a job for each cluster. KUD install the kubernetes cluster on the slice of servers and install ONAP4k8s and all other default plugins such as Multus, OVN, OVN4NFV, NFD, Virtlet, SRIOV
8. BPA rest agent installed in the bootstrap cluster or jump server, and this install rest-api, rook/ceph, Mimio as the cloud storage. This provides a way for user to upload their own software, container images or os image to jump server

Virtual Deployment Guide

Standard Deployment Overview



Virtual deployment is used for the dev env using metal3 virtual deployment to create VM with PXE boot. VM Ansible scripts the node inventory file in the /opt/ironic. No setting is required from the user to deploy the virtual deployment. Virtual deployment is used for dev works.

Snapshot Deployment Overview

no snapshot is implemented in ICN R2

Special requirements for virtual deployment

Install Jump Host

Host server or Jump host required to install with ubuntu 18.04. This install all the VMs and install the k8s clusters. Same as Baremetal deployment use "make vm_install" to install Virtual deployment

Verifying the Setup - VMs

"make verify_all" install two VMs with name master-0 and worker-0 with 8GB RAM and 8vCPUs, And install k8s cluster on the VMs using the ICN - BPAoperator and install the ICN - BPA rest API verifier. BPA operator installs the Multi cluster KUD to bring up the kubernetes with all addons and plugins.

Verifying the Setup

ICN blueprint checks all the setup in both baremetal and VM deployment. Verify script will check the metal3 provision the OS in each baremetal nodes by checking with a timeout period of 60 sec and interval of 30. BPA operator verifies will check, whether the KUD is installation is complete by doing plain curl command to the Kubernetes cluster installed in baremetal and VM setup.

Baremetal Verifier: Run the "make bm_verifer", it will verify the bare-metal deployment

VM Verifier: Run the "make vm_verifier", it will verify the Virtual deployment

Developer Guide and Troubleshooting

For development uses the virtual deployment, it take up 10 mins to bring up the setup virutal BMC VMs with pxeboot.

Virtual deployment works well for the BPA operator development for metal3 installation scripts.

Utilization of Images

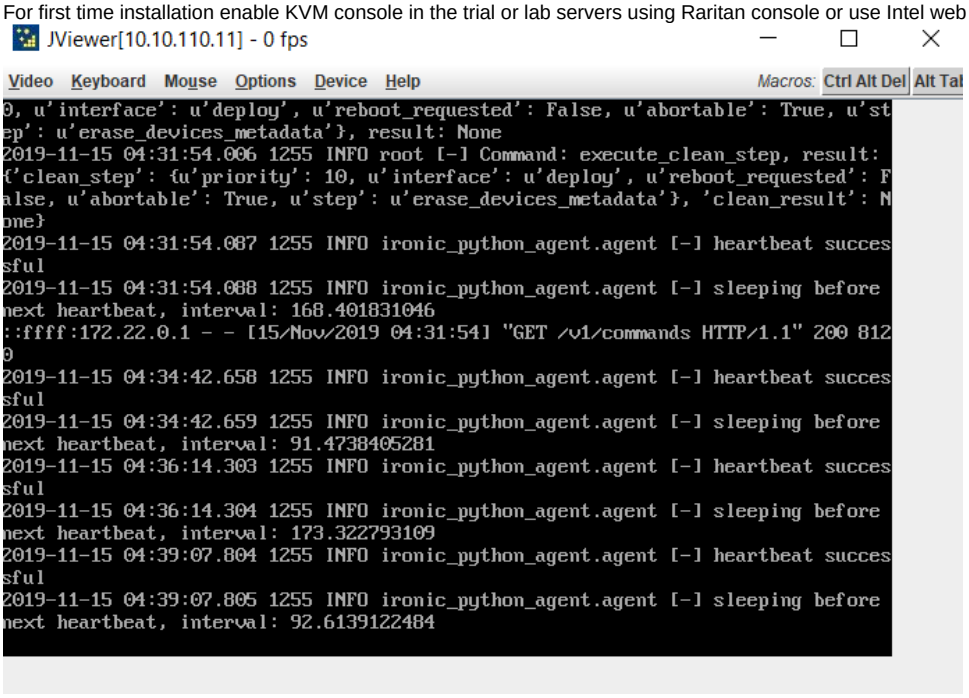
No images provided in ICN R2 release

Post-deployment Configuration

no Post-deployment configuration required in ICN R2 release

Debugging Failures

- For first time installation enable KVM console in the trial or lab servers using Raritan console or use Intel web bmc console



- Deprovision state will result ironic agent to sleeping before next heartbeat - It is not a error message. It Baremetal without OS and installed with ramdisk
- Deprovision in metal3 is not straight forward - Metal3 follow various stages from provisioned deprovisioning ready. ICN BP take care navigating the deprovisioning and removing the BMH CR in case of clean
- Manual BMH cleaning of bmh or force cleaning of bmh resource result in hang state - Use make bmh_clean to remove the bmh state.
- Logs of ironic, openstack baremetal command to see the state of the node.
- Logs of Baremetal operator gives failure related to images or images md5sum errors
- It is not possible to change the state from provision to deprovision or deprovision to provision without completing that state. All the issues are handled in ICN scripts
- Kubernetes cluster failure can be debugged by KUD pod logs

Reporting a Bug

Required Linux Foundation ID to launch bug in ICN: https://jira.akraino.org/projects/ICN/issues

Uninstall Guide

Baremetal deployment

- The command **make clean_all** uninstall all the components installed by "make install"
    - It de-provision all the servers provisioned and remove them from Ironic database
    - Baremetal operator is deleted followed by Ironic database and container
    - Network configuration such internal dhcp server, provisioning interfaces and IPMI LAN interfaces are deleted
    - docker images build during the "make install" are deleted , such all Ironic images, baremetal operator images, bpa operator images and KUD images
    - KUD will reset the bootstrap cluster - Kubernetes cluster is teardown in the jump server and all the associated docker images are removed
    - All software packages installed by "make install_all" are removed, such Ironic , openstack utility tool, docker packages and basic prerequisite packages

Virtual deployment

The command **"make vm_clean_all"** uninstall all the components for the virtual deployments

Troubleshooting

Error Message Guide

The error message is explicit, all messages are captured in logs folder

Maintenance

Blue Print Package Maintenance

no packages is maintained in ICN R2

Software maintenance

not applicable

Hardware maintenance

not applicable

BluePrint Deployment Maintenance

not applicable

Frequently Asked Questions

How to setup IPMI?

First, make sure the IPMI tool is installed in your servers, if not install them using apt install ipmitool

Then, check for the ipmitool information of each servers using the command "ipmitool lan print 1"

If the above command doesn't show the IPMI information, then setup the IPMI static IP address using following instruction

- Mostly easy way to set up IPMI topology in your lab setup is by using IPMI tool.
- Using IPMI tool - https://www.thomas-krenn.com/en/wiki/Configuring_IPMI_under_Linux_using_ipmitool
- IPMI information can be considered during the BIOS setting as well.

BMC web console url is not working?

It is hard to find issues or reason. Check the ipmitool bmc info to find the issues, if the url is not available

No change in bmh state - provisioning state is for more than 40min?

Generally metal3 provision for bare metal takes 20 - 30 mins. Look at the ironic logs and bare-metal operator to look at the state of nodes. Openstack baremetal node shows all state of the node right from power, storage.

Why provide network is required?

Generally, provider network DHCP servers in lab provide the router and dns server details. In some lab setup DHCP server don't provide this information.

License

/*
* Copyright 2019 Intel Corporation, Inc
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

References

Definitions, acronyms and abbreviations