Installing OpenShift 4.13 on a Bare Metal Server

Revised: October 10, 2023

New and Changed Information

The following table provides an overview of the significant changes up to this current release. The table does not provide an exhaustive list of all changes or of the new features up to this release.

Cisco ACI CNI plug-in Release Version	Feature
5.2(7)	Cisco Application Centric Infrastructure (ACI) supports Red Hat OpenShift 4.13 on a bare metal server.

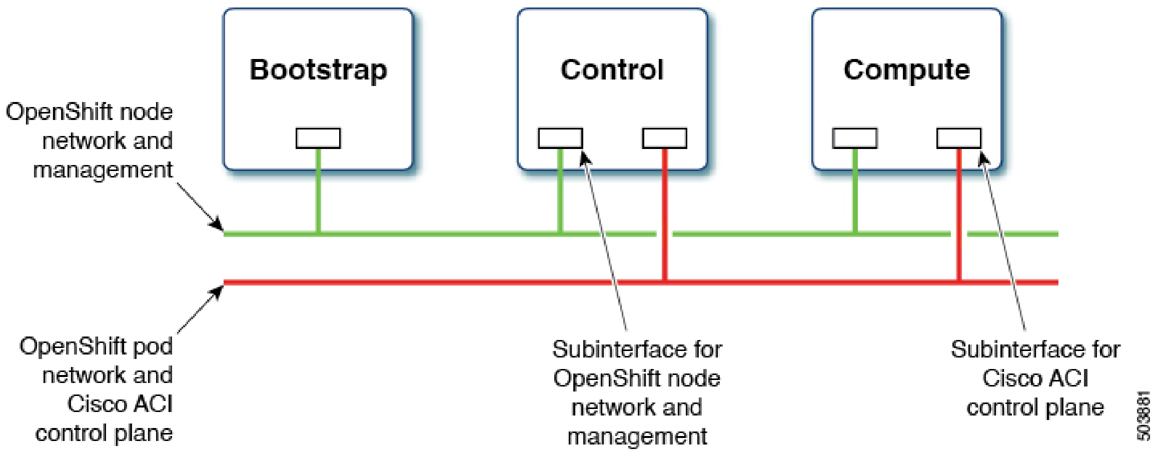
Openshift 4.13 on Bare Metal

This document pertains to installing OCP with the ACI CNI. However, to identify and resolve issues in your infrastructure not related to the ACI CNI, see the relevant installation guide to first install OCP on your bare metal nodes using the default OpenShift SDN CNI. *You can check the OpenShift 4.13 container platform documentation.*

Note This document can not be used standalone. This document should be used along with the *Red Hat OpenShift 4.13 Installing a Cluster on Bare Metal with Network Customizations* document to perform the OpenShift cluster installation.

Requirements for supporting OpenShift 4.13 on a Bare Metal Server

At least two network interfaces are required for bare metal nodes, one for the node network, and the second for the pod network. The design separates OpenShift node traffic from the pod traffic. The separation results in the control and compute machines having two network interfaces, as shown in the following illustration:


Bootstrap
Control
Compute
OpenShift node network and management
OpenShift pod network and Cisco ACI control plane
Subinterface for OpenShift node network and management
Subinterface for Cisco ACI control plane
503881

The first interface is used for the node network and the second one is used for the pod network. The second interface also carries Cisco ACI control plane traffic. A VLAN tagged subinterface is configured on the second interface to carry the cluster's pod traffic and also the Cisco ACI control plane traffic.

Installation Process

The following sections detail the steps required to install the OpenShift cluster using the ACI CNI.

Configuring the OpenShift Installer

Use this procedure to configure the OpenShift installer.

Before you begin

Download the OpenShift installer and OC client.

For details of the location from where you can download the installer , see the OpenShift 4.13 document titled, *Installing a Cluster on Bare Metal with Network Customizations.*

Procedure

Create the `install-config.yaml` file.

```
$ ~/openupi$ pwd
/home/<user>/openupi

$ ~/openupi$ cat install-config.yaml
apiVersion: v1
baseDomain: noiro.local
compute:
- hyperthreading: Enabled
  name: worker
  replicas: 0
controlPlane:
  hyperthreading: Enabled
  name: master
  replicas: 3
metadata:
  name: openupi
networking:
  clusterNetwork:
  - cidr: 10.2.0.0/16
    hostPrefix: 23
  networkType: CiscoACI
  serviceNetwork:
  - 172.30.0.0/16
platform:
  none: {}
fips: false
proxy:
  httpsProxy: <http-proxy>
  httpProxy: <https-proxy>
  noProxy: <no-proxy>
publish: External
pullSecret: <RH-account-pull-secret>
sshKey: <host-ssh-key>
```

Configuring ACI Infra and CNI

Use this procedure for configuring ACI infra and CNI using acc-provision.

Procedure

Sample ACI configuration:

```
# Configuration for ACI Fabric
#
aci_config:
  system_id: openupi                       # Every opflex cluster on the same fabric must have a
distict ID
  apic_hosts:                              # List of APIC hosts to connect to for APIC API access
    - <APIC-IP>
  apic_login:
    username: <username>
    password: <password>
  vmm_domain:                     # Kubernetes VMM domain configuration
    encap_type: vxlan             # Encap mode: vxlan or vlan
    mcast_range:                  # Every vxlan VMM on the same fabric must use a distinct range
        start: 225.115.1.1
        end: 225.115.255.255
    nested_inside:
       installer_provisioned_lb_ip: <loadbalancer_ip>
  # This IP should match the one configured in the loadbalancer during installation.
  # The following resources must already exist on the APIC,
  # this is a reference to use them
  aep: <AAEP_NAME>            # The attachment profile for ports/VPCs connected to this cluster
  vrf:                         # VRF used to create all subnets used by this Kubernetes cluster
    name: <VRF_NAME>           # This should exist, the provisioning tool does not create it
    tenant: <TENANT>            # This can be tenant for this cluster (system-id) or common
  l3out:                       # L3out to use for this kubernetes cluster (in the VRF above)
    name:<L3OUT_NAME>        # This is used to provision external service IPs/LB
    external_networks:
        <EXTERNAL_EPG_NAME>    # This should also exist, the provisioning tool does not create it
#
# Networks used by Kubernetes
#
net_config:
  node_subnet: 172.253.3.1/24         # Subnet to use for nodes
  pod_subnet: 10.2.0.1/16           # Subnet to use for Kubernetes Pods
  extern_dynamic: 10.3.0.1/16       # Subnet to use for dynamically allocated external services
  extern_static: 10.4.0.1/16        # Subnet to use for statically allocated external services
  node_svc_subnet: 10.5.0.1/16      # Subnet to use for service graph
  kubeapi_vlan: 3                   # The VLAN used by the internal physdom for nodes
  service_vlan: 202                   # The VLAN used for external LoadBalancer services
  infra_vlan: 4093
```

Note The `*.apps.<cluster_name>.<base_domain>` records in the user-provisioned DNS should refer to the same IP address used in the installer_provisioned_lb_ip.

Customize the sample `acc-provision` input file shown above as per your requirements. Then, install the latest `acc-provision` package from https://pypi.org/project/acc-provision/ and run `pip install acc-provision` .

Run the `acc-provision` as follows:

```
$ ~/openupi$ pwd
/home/<user>/openupi

$ ~/openupi$ acc-provision -a -c acc_provision_input.yaml -f openshift-4.13-baremetal -u <user> -p
<password> -o aci_deployment.yaml -z aci deployment.yaml.tar.gz
```

This generates a new `aci_deployment.yaml.tar.gz` file which contains the ACI CNI manifests, and is used later during the OpenShift installation.

Preparing Custom Network Configuration for OpenShift Nodes

ACI CNI requires additional VLANs to be extended towards each OpenShift node. Additional VLANS are required for master and worker nodes, but not required for the bootstrap node.

You can configure additional VLANs on the interface that will be configured with the node network subnet, or can be configured on an additional physical interface on the hosts.

The two options available for configuring additional VLANs are listed below, you can use either of these:

- A Cisco script that modifies ignition files before the installation. This method is preferred for environments with uniform hardware, where the interface names on all master nodes should be the same, and likewise, all interfaces on worker nodes should be the same. However, it is important to note that the interfaces on master nodes and worker nodes can differ. See the Ignition Files Modification, on page 5 section. In this method you must ensure DHCP server for node network.
- Network configuration using kernel arguments in the PXE boot configuration. This option enables you to create individual network configuration(s) per node server. See the Additional Network Configuration Using PXE, on page 8 section.

Ignition Files Modification

Use this procedure for configuring ignition files for the bare metal nodes.

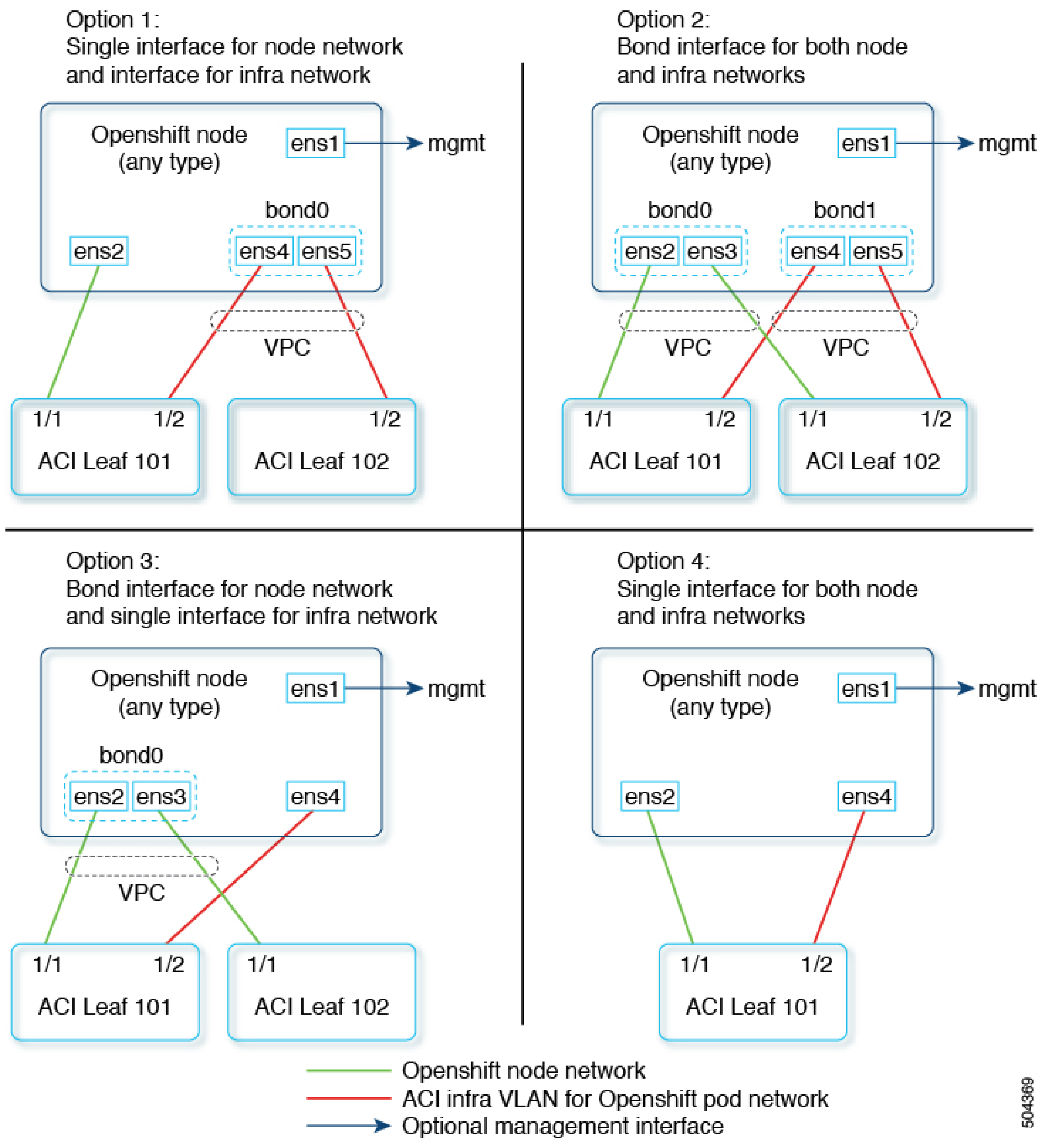
Before you begin

Prepare the Ignition Files

From Github download the `config.yaml` and `update_ign.py` files. This script updates the CoreOS ignition file, with additional NIC configuration, required to extend the Cisco ACI internal network (Infra VLAN) up to the server level. This interface is used to carry VxLAN traffic from OVS towards the ACI leaf switch with an appropriate tag for the pod network. The script offers four options for node and pod network configuration:

- Single interface for node network and bond interface for infra network.
- Bond interface for both node and infra networks.
- Bond interface for node network and single interface for infra network.
- Single interface for both node and infra networks.

Each of the above options have been illustrated as shown below:


Option 1:
Single interface for node network and interface for infra network
Openshift node (any type)
ens1
mgmt
bond0
ens2
ens4
ens5
VPC
1/1
1/2
ACI Leaf 101
1/2
ACI Leaf 102
Option 2:
Bond interface for both node and infra networks
Openshift node (any type)
ens1
mgmt
bond0
bond1
ens2
ens3
ens4
ens5
VPC
VPC
1/1
1/2
ACI Leaf 101
1/1
1/2
ACI Leaf 102
Option 3:
Bond interface for node network and single interface for infra network
Openshift node (any type)
ens1
mgmt
bond0
ens2
ens3
ens4
VPC
1/1
1/2
ACI Leaf 101
1/1
ACI Leaf 102
Option 4:
Single interface for both node and infra networks
Openshift node (any type)
ens1
mgmt
ens2
ens4
1/1
1/2
ACI Leaf 101
Openshift node network
ACI infra VLAN for Openshift pod network
Optional management interface
504369

Note Node network is configured as VLAN subinterface of either bond0 or physical NIC. You can configure the server with additional VLAN(s) for management purpose or use the node network for management network. The design might be dependent on the server provisioning method (PXE or manual ISO boot).

Read the instructions provided at the top of the `update_ign.py` script and accordingly update the `config.yaml` file to match your environment.

Update the `config.yaml` file to match your environment. In the config file, specify the pair of physical interfaces used as a bond or single interface for standalone. The script generates three ignition files: *boostrap.ign*, *master.ign* and *worker.ign*. Hostname should be configured from DHCP during the boot. The script does not add any static IP addresses. The node network is assumed to use DHC and the sub-interfaces for ACI infra VLAN is configured to get IP address from the DHCP server running on the APICs.

```
all:
  infra_vlan: 4093
  kubeapi_vlan: 3
  service_vlan: 202
  network_interfaces:
   node:
     mtu: 1500
   opflex:
     mtu: 1700
  os_compute_nodes:
   node_network_interface:
     - ens2
     - ens3
   aci_infra_network_interface:
     - ens4
     - ens5
  os_cp_nodes:
   node_network_interface:
     - ens1f0
     - ens1f1
   aci_infra_network_interface:
     - ens1f2
     - ens1f3
```

Procedure

Step 1 Create a root folder for your cluster.

```
cd /home/<user>/openupi
mkdir upi
```

Step 2 Copy the `install-config.yaml`, `config.yaml`, and the `update_ign.py` in the newly created upi folder.

Step 3 Create the manifests.

```
openshift-install create manifests --log-level debug --dir=upi
```

Step 4 Extract all the ACI manifest files in `upi/manifests/`.

```
tar -xvf aci_deployment.yaml.tar.gz -C upi/manifests/
```

Step 5 Create the ignition configurations.

```
openshift-install create ignition-configs  --log-level debug --dir=upi
```

Step 6 Update the ignition files with the ACI CNI specific configuration(s).

```
cd upi
export INFRA_ID=$(jq -r .infraID metadata.json)
python3 update_ign.py
```

The ignition files are now ready and can be copied to your HTTP server, so they can be served to your nodes. The `update_ign.py` will generate one ignition file per node following the *ocpbm-<cluster-id>-<node-name>-ignition.json* format.

What to do next

Proceed with the installation of the cluster; see the *Redhat OpenShift 4.13 document* (mentioned earlier in the chapter).

Additional Network Configuration Using PXE

Use this procedure as an alternate procedure to prepare custom network configuration for OpenShift nodes.

Follow the instructions in the *Installing RHCOS by using PXE or iPXE booting for Openshift 4.13 Baremetal User-Provisioned cluster on bare metal* document which is available under the official OpenShift documentation.

Add the following modification(s) to the PXE menu configuration files:

```
DEFAULT pxeboot
TIMEOUT 20
PROMPT 0
LABEL pxeboot
    KERNEL http://<HTTP_server>/rhcos-<version>-live-kernel-<architecture>
    APPEND initrd=http://<HTTP_server>/rhcos-<version>-live-initramfs.<architecture>.img
        coreos.live.rootfs_url=http://<HTTP_server>/rhcos-<version>-live-rootfs.<architecture>.img
        coreos.inst.install_dev=/dev/sda coreos.inst.ignition_url=http://<HTTP_server>/bootstrap.ign
    ip=bond1:off:9000 ip=bond1.4093:dhcp:9000 bond=bond1:ens1f0,ens1f1:mode=802.3ad,miimon=100,lacp_rate=slow


    vlan=bond1.4093:bond0 rd.route=224.0.0.0/4::bond1.4093
```

In the above example, hosts are configured with an additional `bond` interface with two member physical interfaces, `bond0` and `bond1`. `bond1` uses VLAN tagging for ACI “Infra VLAN”. You also need a multicast route, which should be attached to the new `bond` subinterface.

Note You can also configure the primary interface (for node network), using kernel parameters. PXE Boot works in the native VLAN (no tagging), and does not support LACP bonding. If you need to use VPC / Port Channel configuration for node network interfaces, you need to configure ACI VPC Policy Group without **Suspend Individual** on LACP policy.

Procedure

Step 1 Create a root folder for your cluster using the **mkdir** command.

```
cd /home/<user>/openupi mkdir upi
```

Step 2 Copy the install-config.yaml, config.yaml, and the update_ign.py in the newly created upi folder.

Step 3 Create the manifests using the **openshift-install create manifests** command.

```
openshift-install create manifests --log-level debug --dir=upi
```

Step 4 Extract all the ACI manifest files in `upi/manifests/`.

```
tar -xvf aci_deployment.yaml.tar.gz -C upi/manifests/
```

Step 5 Create the ignition configurations.

```
openshift-install create ignition-configs --log-level debug --dir=upi
```

Step 6 Prepare PXE configuration(s) for all the server(s) and start the bare metal servers.

What to do next

Proceed with the installation of the cluster; see the *Redhat OpenShift 4.13* document.

Updating the Default Ingress Controller

For updating the default Ingress Controller publish strategy to use the ACI Loadbalancer, log in as a user with cluster-admin privileges and run the following:

```
oc replace --force --wait --filename - <<EOF
apiVersion: operator.openshift.io/v1
kind: IngressController
metadata:
  namespace: openshift-ingress-operator
  name: default
spec:
  endpointPublishingStrategy:
    type: LoadBalancerService
    loadBalancer:
      scope: External
EOF
```

For more details, see the *Configuring the Default Ingress Controller for your Cluster to be Internal* section in the *Ingress Operator in OpenShift Container Platform Red Hat* guide.

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS REFERENCED IN THIS DOCUMENTATION ARE SUBJECT TO CHANGE WITHOUT NOTICE. EXCEPT AS MAY OTHERWISE BE AGREED BY CISCO IN WRITING, ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS DOCUMENTATION ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

The Cisco End User License Agreement and any supplemental license terms govern your use of any Cisco software, including this product documentation, and are located at: http://www.cisco.com/go/softwareterms.Cisco product warranty information is available at http://www.cisco.com/go/warranty. US Federal Communications Commission Notices are found here http://www.cisco.com/c/en/us/products/us-fcc-notice.html.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any products and features described herein as in development or available at a future date remain in varying stages of development and will be offered on a when-and if-available basis. Any such product or feature roadmaps are subject to change at the sole discretion of Cisco and Cisco will have no liability for delay in the delivery or failure to deliver any products or feature roadmap items that may be set forth in this document.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

The documentation set for this product strives to use bias-free language. For the purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com go trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2023 Cisco Systems, Inc. All rights reserved.

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA 95134-1706
USA

Asia Pacific Headquarters
CiscoSystems(USA)Pte.Ltd.
Singapore

Europe Headquarters
CiscoSystemsInternationalBV
Amsterdam,TheNetherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.