



Install Cisco iNode Manager

This chapter describes how to install the SMI Cluster Deployer and iNode Manager clusters for single-node and multinode deployment in an offline environment.

- [Install Cisco iNode Manager with Autodeployer, on page 1](#)
- [Troubleshooting, on page 19](#)
- [iNode Manager Web Interface Access, on page 21](#)
- [Post Installation/Upgrade Checklist, on page 22](#)
- [Log Examples, on page 24](#)

Install Cisco iNode Manager with Autodeployer

Prerequisite: Install Intelligent Node Software on GS7000 iNode

As a prerequisite for iNode Manager 23.1 installation or migration, ensure that the GS7000 iNode runs 3.1.1 and above OIB Image by performing the following steps.

-
- | | |
|---------------|---|
| Step 1 | Download GS7000 iNode image from Cisco Software Download page. |
| Step 2 | Set iNode version in the DHCP configuration file as <i>03.01.01</i> or above. |
| Step 3 | Enable the force upgrade option in the DHCP configuration. See GS7000 iNode Release Notes . |
| Step 4 | Reboot iNode. |
-

Background

To install an SMI cluster, the following setup is necessary:

- *Staging server*: a physical or virtual machine to run the installation script.
- *Hypervisor (VMware ESXi 7.0)*
- *vCenter* (version 7.0 or above): manager for the vSphere infrastructure that hosts the VMs for the SMI clusters.



Note We recommend that you use VMware vCenter Server 7.0 with VMFS 6 Datastore type.

The installation process creates the following:

- *SMI Cluster Manager* (or *Deployer*): a controller to configure and deploy SMI cluster.
- *SMI Cluster*: the cluster on which the target product application runs

A *release image bundle* is a compressed tarball file that contains all the scripts, helm charts, and docker images necessary to install the Deployer and the SMI cluster. It also contains copies of these instruction and configuration examples.

You can use the *Autodeploy* script that is in the bundle to set up the Deployer and the SMI clusters.

Cisco iNode manager supports two cluster sizes:

- *Single-node cluster* (also called *All-In-One cluster*, or *AIO*): Runs on a single VM.
- *Multinode cluster*: Runs on three UCS servers; each with a *Control-Plane*, *ETCD*, *Infra*, and *App* VMs, giving a total of 12 VMs. Multinode cluster supports two sizes `small` and `normal`.

The multinode cluster provides full high-availability support and is the only recommended cluster size for production environment. The default size is `small` when `size` field is not provided..

The following tables show the minimum requirements for AIO cluster:

| Node Type | CPUs | RAM Size (GB) | Disk Size (GB) |
|-----------|------|---------------|----------------|
| Deployer | 8 | 16 | 320 |
| AIO | 18 | 96 | 1529 |

The following tables show the minimum requirements for each of the VM types deployed in a multimode cluster:

Table 1: Small Multimode Cluster

| Node Type | CPUs | RAM Size (GB) | Disk Size (GB) |
|---------------|------|---------------|----------------|
| Deployer | 8 | 16 | 320 |
| Control-plane | 2 | 16 | 125 |
| etcd | 2 | 16 | 125 |
| Infra | 8 | 64 | 1000 |
| Ops | 8 | 64 | 320 |

Table 2: Normal Multimode Cluster

| Node Type | CPUs | RAM Size (GB) | Disk Size (GB) |
|-----------|------|---------------|----------------|
| Deployer | 8 | 16 | 320 |

| Node Type | CPUs | RAM Size (GB) | Disk Size (GB) |
|---------------|------|---------------|----------------|
| Control-plane | 2 | 16 | 125 |
| etcd | 2 | 16 | 125 |
| Infra | 14 | 98 | 1500 |
| Ops | 16 | 180 | 320 |

High-Level Overview of Installation Workflow

1. [Prepare Staging Server, on page 3](#)
2. [Prepare a Cluster Configuration File, on page 4](#)
3. [Deploy the Cluster, on page 16](#)

You can prepare and deploy multiple clusters, if necessary.

Prepare Staging Server

The staging server can be any type of host: physical server, virtual machine, or even a laptop. However, the server must be able to connect to the target VMware vSphere Infrastructure, vCenter Server, and cluster nodes with correct credentials.

Prerequisites

The Staging Server must have the following software installed:

- [Docker \(18.09.7 or later\)](#)
- [Python \(3.6 or later\)](#)



Note Ensure that the staging server has internet connectivity to download the iNode Manager release bundle from the Cisco software downloads page.

Unpack Cisco iNode Manager Release Bundle

The iNode Manager release bundle image is a compressed tarball file that is self-sufficient for the Deployer and the iNode Manager cluster installation. It contains the following files:

- Installation script
- All relevant product images
- Sample configuration files
- Copy of the README file

Step 1 Download the signed iNode Manager release bundle image to the Staging Server. Extract the content with the following command:

```
tar -xzovf inode-manager-installer-<version>.SPA.tgz
```

This command untars the signed bundle.

Step 2 Run the following command to extract all the individual images of SMI, Cisco Operations Hub, and iNode Manager.

```
tar -xzovf inode-manager-installer-<version>.tgz
```

This extraction creates the installation directory `inode-manager-installer-<version>` with the following content:

```
host# tree -a
.
├── cluster-deployer-airgap.vmdk
├── cluster-deployer-airgap.vmdk.signature
├── deploy
├── deploy.signature
├── examples
│   ├── aio-inode-manager-cli-config
│   ├── aio-inode-manager-config.yaml
│   ├── aio-inode-manager-standby-config.yaml
│   ├── deployer-sample-config-autodeploy.yaml
│   ├── deployer-sample-config.yaml
│   ├── multinode-inode-manager-cli-config
│   ├── multinode-inode-manager-config.yaml
│   └── multinode-inode-manager-standby-config.yaml
├── offline-products
│   ├── cee-<version>.tar
│   ├── cee-<version>.tar.signature
│   ├── opshub-<version>.tar
│   ├── opshub-<version>.tar.signature
│   ├── inode-manager-<version>.tar
│   └── inode-manager-<version>.tar.signature
├── README.md
├── utility-images
│   ├── autodeploy_<version>.tar.gz
│   ├── autodeploy_<version>.tar.gz.signature
│   ├── cluster-manager-docker-deployer_<version>.tar
│   └── cluster-manager-docker-deployer_<version>.tar.signature
```

We call this directory the staging directory in this document.

Prepare a Cluster Configuration File

VMware vCenter Details

To contact the VMware vCenter server, the deploy script and the deployer require the following details:

- Server name or IP address
- Username and password
- Datacenter and cluster name
- Host server and data store names

For the Deployer and single-node cluster, one host server is necessary. For multinode clusters, three host servers are necessary.

The Deployer and the SMI Clusters can run on different vCenters.

IP Addresses for Deployer and Cluster

Deploying the iNode Manager software offline requires the following IP addresses:

- One management IP address for the Deployer
- Management IP addresses for cluster nodes (1 for single-node, 12 for multinode clusters)
- Converged Interconnect Network (CIN) network IP addresses for iNode Manager (1 per CIN interfaces per APP node)
- For multinode clusters, 1 virtual IP for management and 1 for each CIN network

Cluster Configuration File

Place the configuration under the staging directory. This configuration file is in the standard [YAML language format](#), with the following three sections:

- Environments
- Deployers
- Clusters (iNode Manager multinode or single-node)

Each section can contain multiple items.



Note Replace all the fields marked with <...> in the following sections with actual values.

VMware vCenter Environment Configuration

This section provides details of the VMware vCenter access and network access for creating and provisioning the deployers and cluster virtual machines.

```
environments:
  <environment name>:
    server: <vCenter name or IP address>
    username: <vCenter user name>
    datacenter: <vCenter datacenter name>
    cluster: <vCenter cluster name>
    nics: [ <LIST of vCenter management networks> ]
    host: <UCS host>
    datastore: <Datastore name>
    nameservers: [ <LIST of DNS servers> ]
    search-domains: [ <LIST of search domains> ]
    ntp: <ntp server name or IP address>
```

Guidelines for configuring the VMware vCenter Environment:

- The environment name can have only lowercase letters, digits, and hyphens (-).

- The NIC's list must have only one network, although the NIC configuration allows multiple networks. The deployer or cluster that refers to this environment uses this network as the management network.
- Configure multiple environments for this vCenter if your vCenter has more than one network that serves as a management network. Configure an environment for each network. Use the corresponding environment in the deployer or cluster, based on the management network it uses.
- Configure the NIC's `nameservers` and `search-domains` fields as lists.



Note If there are special characters in the username, update the configuration from the deployer CLI. Add double quotes (") around the username value and rerun the `sync` command.

Deployer Configuration

Before creating and deploying a deployer, define a minimum of one environment.

FQDN disabled:

```

deployers:
  <deployer name>:
    environment: <environment of vCenter hosting the deployer>
    address: <deployer VM IP address in CIDR format>
    gateway: <gateway IP address>
    username: <user name for deployer>
    # SSH private-key-file with path relative to the staging directory
    # If the line is missing, ssh private key will be auto-generated and saved inside
    .sec/
    private-key-file: <path and filename for ssh private key>
    host: <ESXi host IP address>
    datastore: <vCenter datastore name for host>

```

FQDN enabled:

```

deployers:
  <deployer name>:
    environment: <environment of vCenter hosting the deployer>
    address: <deployer VM IP address in CIDR format>
    gateway: <gateway IP address>
    username: <user name for deployer>
    # SSH private-key-file with path relative to the staging directory
    # If the line is missing, ssh private key will be auto-generated and saved inside
    .sec/
    private-key-file: <path and filename for ssh private key>
    host: <ESXi host IP address>
    datastore: <vCenter datastore name for host>
    # ingress-hostname only supports valid FQDN
    ingress-hostname: "deployer.example.com"

```

Guidelines for configuring the deployer:

- The name of the deployer can have only lowercase letters, digits, and hyphens (-).
- The `private-key-file` field, when present, must refer to the SSH private key file. This file must be in the staging directory and must not be accessible (read, write, or execute) to other users.

If the `private-key-file` line is missing, the deploy script generates an SSH private key for the deployer (or SMI cluster) and places it in the `.sec` subdirectory under the staging directory. The filename is `<deployer-name>_auto.pem`.

- To avoid resource-contention, do not run the deployer in an ESXi server that serves any iNode Manager clusters.

Cluster Configuration

Before creating and deploying a cluster, configure a minimum of one environment and one deployer. A cluster has an environment field to reference to its corresponding environment.

```
clusters:
"multinode-blr":
  type: "opshub"
  size: "normal"
  environment: "sj-mn-inf"
  username: "cloud-user"
  # "true" for dual-stack, otherwise "none"
  ipv6-mode: "true"
  # private-key-file must exist in the path of staging/install directory
  # file path is relative to the staging/install directory
  private-key-file: "mncmtsb.pem"
  primary-vip: "10.64.98.219/25"
  primary-vip-ipv6: "2001:420:54FF:24:0000:0000:655:0017/112"
  gateway: "10.64.98.129"
  # You can configure the optional parameter ingress-hostname to enable FQDN for ingress
  access.
  # If you do not configure ingress-hostname, ingress can be accessed via
  <primary-vip>.nip.io
  #ingress-hostname: "blrmn.opsdev.com"
  ipv6-gateway: "2001:420:54FF:24:0000:0000:655:1"
  # 'ingress-hostname only supports '.' and alphanumeric characters
  nodes:
    - host: "10.64.98.171"
      datastore: "datastore1 (2)"
      addresses: [ "10.64.98.220", "10.64.98.221", "10.64.98.222", "10.64.98.223" ]
      addresses-v6: [ "2001:420:54FF:24:0000:0000:655:b/112",
"2001:420:54FF:24:0000:0000:655:c/112", "2001:420:54FF:24:0000:0000:655:d/112",
"2001:420:54FF:24:0000:0000:655:e/112" ]
    - host: "10.64.98.172"
      datastore: "datastore1 (3)"
      addresses: [ "10.64.98.224", "10.64.98.225", "10.64.98.226", "10.64.98.227" ]
      addresses-v6: [ "2001:420:54FF:24:0000:0000:655:f/112",
"2001:420:54FF:24:0000:0000:655:0017/112", "2001:420:54FF:24:0000:0000:655:0011/112",
"2001:420:54FF:24:0000:0000:655:0012/112" ]
    - host: "10.64.98.173"
      datastore: "datastore1 (4)"
      addresses: [ "10.64.98.228", "10.64.98.229", "10.64.98.230", "10.64.98.231" ]
      addresses-v6: [ "2001:420:54FF:24:0000:0000:655:0013/112",
"2001:420:54FF:24:0000:0000:655:0014/112", "2001:420:54FF:24:0000:0000:655:0015/112",
"2001:420:54FF:24:0000:0000:655:0016/112" ]
  apps:
    - inode-manager:
      nodes:
        - host: 10.64.98.171
          # nics and ops->interfaces nodes are array object they are mapped by array
index.
  nics:
    - 7.29.9.x Network
  ops:
    interfaces:
      -
      addresses:
        - 7.29.9.20/16
      # vip - Virtual IP address of the southbound interface
      vip:
```

```

        - 7.29.9.23/16
        vrouter-id: 20
- host: 10.64.98.172
  # nics and ops->interfaces nodes are array object they are mapped by array
index.
  nics:
  - 7.29.9.x Network
  ops:
    interfaces:
    -
      addresses:
      - 7.29.9.21/16
      vip:
      - 7.29.9.23/16
      vrouter-id: 20
- host: 10.64.98.173
  # nics and ops->interfaces nodes are array object they are mapped by array
index.
  nics:
  - 7.29.9.x Network
  ops:
    interfaces:
    -
      addresses:
      - 7.29.9.22/16
      vip:
      - 7.29.9.23/16
      vrouter-id: 20

# For Single-Node cluster only
clusters:
  "cicd-aio-nodes":
    type: opshub
    environment: "chn-smi-inodemgr-lab"
    username: "inodemgruser"
    gateway: "10.78.229.1"
    private-key-file: "inodemgr.pem"
    #pod-subnet is an optional field if not given by default "192.168.0.0/16" will be
assigned.
    pod-subnet: "192.168.120.0/24"
    # service-subnet is an optional field if not given by default "10.96.0.0/12" will be
assigned.
    service-subnet: "10.96.120.0/24"
    # docker-bridge-subnet is an optional field if not given by default 172.17.0.0/16"
will be assigned.
    docker-bridge-subnet: ["172.17.0.0/16"]
    nodes:
    - host: 10.78.229.151
      datastore: DatastoreSSD-229-151
      datastore-folder: "ClusterDataStore"
      addresses: ["10.78.229.229/24"]

  apps:
  - inode-manager:
    nodes:
    - host: 10.78.229.151
      nics:
      - "VLAN 175"
      control-plane:
      interfaces:
      -
        addresses:
        - 175.175.255.229/16
        - "2002::afaf:ffd6/112"

```



```

routes:
-
  dest:
  - 192.175.175.0/24
  nhop: "175.175.254.254"
-
  dest:
  - "2002::C0af:af00/120"
  nhop: "2002::afaf:fefe"

```

Guidelines for configuring a cluster:

- The name of the cluster can have only lowercase letters, digits, and hyphens (-).
- The `private-key-file` field, when present, must refer to the SSH private key file. This file must be in the staging directory and must not be accessible (read, write, or execute) to other users.

If the `private-key-file` line is missing, the deploy script generates an SSH private key for the deployer (or SMI cluster) and places it in the `.sec` subdirectory under the staging directory. The filename is `<deployer-name>_auto.pem`.

- Configure the virtual IP address (`master-vip`) and VRRP ID (`vrouter-id` at the cluster level) for the management network for multinode clusters. The management network supports only IPv4. The `vrouter-id` parameter can take values 1–254.
- If multiple clusters share the same management subnet, the VRRP ID for each cluster must be unique in the management subnet.
- The `ingress-hostname` field, when present, only supports valid DNS name, i.e., a fully qualified domain name (FQDN). If `ingress-hostname` is specified, for example `inodemgr.cisco.com`, then the following FQDNs are used:

```

- inodemgr.cisco.com
- restconf.cee-data-ops-center.inodemgr.cisco.com
- cli.cee-data-ops-center.inodemgr.cisco.com
- restconf.opshub-data-ops-center.inodemgr.cisco.com
- cli.opshub-data-ops-center.inodemgr.cisco.com
- restconf.inode-manager-data-ops-center.inodemgr.cisco.com
- cli.inode-manager-data-ops-center.inodemgr.cisco.com
- grafana.inodemgr.cisco.com
- show-tac-manager.cee-data-smi-show-tac.inodemgr.cisco.com

```



Note It's recommended to register a wildcard DNS record, such as `*.inodemgr.cisco.com`, so that all the sub domains resolve to the same IP. Otherwise all the above FQDNs must be configured in the DNS server.

The IP address used for the DNS record/FQDN is the `ingress-ip` (for Multi-Node clusters)/ AIO VM IP address (for AIO clusters).

If `ingress-hostname` is not specified, then the specified `ingress-ip` is used to create a FQDN. For example, if `ingress-ip` is `1.2.3.4`, then the following FQDNs are used. For AIO installation(s), the `ingress-ip` is the IP assigned to the AIO/Ops node.

```

- 1.2.3.4.nip.io
- restconf.cee-data-ops-center.1.2.3.4.nip.io
- cli.cee-data-ops-center.1.2.3.4.nip.io
- restconf.opshub-data-ops-center.1.2.3.4.nip.io
- cli.opshub-data-ops-center.1.2.3.4.nip.io
- restconf.inode-manager-data-ops-center.1.2.3.4.nip.io

```

```
- cli.inode-manager-data-ops-center.1.2.3.4.nip.io
- grafana.1.2.3.4.nip.io
- show-tac-manager.cee-data-smi-show-tac.1.2.3.4.nip.io
```



Note The DNS server must allow the resolution of `nip.io` domain names (corporate DNS resolution policies must not block the resolution of `nip.io` domain names) for this approach to work.

iNode Manager CIN Configuration

Configure Converged Interconnect Network (CIN) for the iNode Manager cluster. One or more CIN networks can be present. Configure CIN under each node.

Guidelines for configuring CIN:

- CIN must contain the network names (`nics`) and the IP addresses (`addresses`).
- The routing table (`routes`) is optional.
- Use the virtual IP addresses (`vip`) and the VRRP ID (`vrouter-id`) fields only in multinode clusters. Configure them on the first node.
- The virtual IP addresses are mandatory. You can configure up to one IPv4 and one IPv6 address per CIN network.
- If multiple iNode Manager clusters share a CIN subnet, the VRRP ID must be unique for each cluster.
- For multinode cluster, all nodes must have the same number of CIN interfaces. If the NICs or route fields are missing for the second or third nodes, use the corresponding value from the first node.
- You can also set up a iNode Manager cluster as a backup cluster. For backup clusters, do not include any CIN configuration. The configuration must not have operations and interfaces under the nodes.

Sample Configuration Files

The `examples` directory contains sample configuration files for automatic deployment:

- `deployer-sample-config-autodeploy.yaml`: Configuration file with only the deployer configuration.
- `aio-inode-manager-config.yaml`: Configuration file with the deployer and the single-node iNode Manager cluster configuration.
- `multinode-inode-manager-config.yaml`: Configuration file with the deployer and multinode iNode Manager cluster configuration.
- `aio-inode-manager-standby-config.yaml`: Configuration file with the standby deployer and single-node iNode Manager cluster configuration (without CIN config).
- `multinode-inode-manager-standby-config.yaml`: Configuration file with the standby deployer and multinode iNode Manager cluster configuration (without CIN config).



Note You can find `ingress-hostname` parameter in the sample configuration files only when the FQDN is enabled.

deployer-sample-config-autodeploy.yaml

```

deployers:
  smi-deployer-147:
    address: 10.78.229.147/24
    datastore: DatastoreSSD-229-151
    environment: chn-smi-inodemgr-lab
    gateway: "10.78.229.1"
    host: "10.78.229.151"
    private-key-file: inodemgr.pem
    username: cloud-user
    #ingress-hostname only supports valid FQDN
    ingress-hostname: deployer.example.com
    #Optional configuration
    docker-subnet-override:
      - pool-name: pool1
        base: 172.17.0.0/16
        size: 16

```

```

environments:
  chn-smi-inodemgr-lab:
    cluster: smi
    datacenter: CABU-VC65
    datastore: "DatastoreSSD-229-150 (1)"
    nameservers:
      - "172.30.131.10"
      - "172.16.128.140"
    nics:
      - "VM Network"
    ntp:
      - 8.ntp.esl.cisco.com
      - 2.ntp.esl.cisco.com
    search-domains:
      - cisco.com
    server: "10.78.229.250"
    username: administrator@CABU.VCENTER60
apps:
  - inode-manager

```

aio-inode-manager-config.yaml

```

deployers:
  smi-deployer-147:
    address: 10.78.229.147/24
    datastore: DatastoreSSD-229-151
    datastore-folder: "ClusterDataStore"
    environment: chn-smi-inodemgr-lab
    gateway: "10.78.229.1"
    host: "10.78.229.151"
    private-key-file: inodemgr.pem
    username: cloud-user
    #ingress-hostname only supports valid FQDN
    ingress-hostname: deployer.example.com
    #Optional configuration
    docker-subnet-override:
      - pool-name: pool1
        base: 172.17.0.0/16

```

```

        size: 16

environments:
  chn-smi-inodemgr-lab:
    cluster: smi
    datacenter: CABU-VC65
    datastore: DatastoreSSD-229-150
    nameservers:
      - "172.30.131.10"
      - "172.16.128.140"
    nics:
      - "VM Network"
    ntp:
      - 8.ntp.esl.cisco.com
      - 2.ntp.esl.cisco.com
    search-domains:
      - cisco.com
    server: "10.78.229.250"
    username: administrator@CABU.VCENTER60

clusters:
  "cicd-aio-229":
    type: opshub
    environment: "chn-smi-inodemgr-lab"
    username: "inodemgruser"
    gateway: "10.78.229.1"
    private-key-file: "inodemgr.pem"
    ipv6-mode: "true"
    ipv6-gateway: "2001:0000:0000:0000:0000:0000:655:1"
    #pod-subnet is an optional field if not given by default "192.168.0.0/16" will be
assigned.
    pod-subnet: "192.168.121.0/24"
    # service-subnet is an optional field if not given by default "10.96.0.0/12" will be
assigned.
    service-subnet: "10.96.130.0/24"
    # docker-bridge-subnet is an optional field if not given by default 172.17.0.0/16"
will be assigned.
    docker-bridge-subnet: ["172.20.0.0/16"]
    nodes:
      - host: 10.78.229.151
        datastore: DatastoreSSD-229-151
        datastore-folder: "ClusterDataStore"
        addresses: ["10.78.229.229/24"]
        addresses-v6: [ "2001:0000:0000:0000:0000:0000:afaf:e5e5/112"]
    apps:
      - inode-manager:
          nodes:
            - host: 10.78.229.151
              nics:
                - "VLAN 175"
              control-plane:
                interfaces:
                  -
                    addresses:
                      - 172.17.255.229/16
                      - "2002::afaf:ffe5/112"
                routes:
                  -
                    dest:
                      - 192.168.174.0/24
                    nhop: "172.17.254.254"
                  -
                    dest:
                      - "2002::C0af:af00/120"

```

```

                                nhop: "2002::afaf:fefe"
apps:
  - inode-manager

```

multinode-inode-manager-config.yaml

```

deployers:
  smi-deployer-147:
    address: 10.78.229.147/24
    datastore: DatastoreSSD-229-151
    environment: chn-smi-inodemgr-lab
    gateway: "10.78.229.1"
    host: "10.78.229.151"
    private-key-file: inodemgr.pem
    username: cloud-user
    #ingress-hostname only supports valid FQDN
    ingress-hostname: deployer.example.com
    #Optional configuration
    docker-subnet-override:
      - pool-name: pool1
        base: 172.17.0.0/16
        size: 16

environments:
  chn-smi-inodemgr-lab:
    cluster: smi
    datacenter: CABU-VC65
    datastore: "DatastoreSSD-229-150 (1)"
    nameservers:
      - "172.30.131.10"
      - "172.16.128.140"
    nics:
      - "VM Network"
    ntp:
      - 8.ntp.esl.cisco.com
      - 2.ntp.esl.cisco.com
    search-domains:
      - cisco.com
    server: "10.78.229.250"
    username: administrator@CABU.VCENTER60

clusters:
  "cicd-multi-node-211":
    type: opshub
    environment: "chn-smi-inodemgr-lab"
    username: "inodemgruser"
    gateway: "10.78.229.1"
    primary-vip: "10.78.229.211/23"
    vrouter-id: 78
    private-key-file: "inodemgr.pem"
    ipv6-mode: "true"
    primary-vip-ipv6: "2001:0000:0000:0000:0000:0000:655:9/112"
    ipv6-gateway: "2001:0000:0000:0000:0000:0000:655:1"
    ingress-hostname: "inodemgr-chn08-dev01.cisco.com"
    enable-http-redirect: "true"
    #pod-subnet is an optional field if not given by default "192.168.0.0/16" will be
assigned.
    pod-subnet: "192.168.121.0/24"
    # service-subnet is an optional field if not given by default "10.96.0.0/12" will be
assigned.
    service-subnet: "10.96.130.0/24"
    # docker-bridge-subnet is an optional field if not given by default 172.17.0.0/16"
will be assigned.
    docker-bridge-subnet: ["172.20.0.0/16"]

```

```

nodes:
- host: 10.78.229.150
  datastore: "DatastoreSSD-229-150 (1)"
  addresses: ["10.78.229.217", "10.78.229.214", "10.78.229.224", "10.78.229.221"]
  addresses-v6: [ "2001:0000:0000:0000:0000:0000:655:5/112",
"2001:0000:0000:0000:0000:0000:655:6/112", "2001:0000:0000:0000:0000:0000:655:7/112",
"2001:0000:0000:0000:0000:0000:655:8/112" ]
- host: 10.78.229.151
  datastore: DatastoreSSD-229-151
  addresses: ["10.78.229.213", "10.78.229.216", "10.78.229.219", "10.78.229.223"]
  addresses-v6: [ "2001:0000:0000:0000:0000:0000:655:4/112",
"2001:0000:0000:0000:0000:0000:655:9/112", "2001:0000:0000:0000:0000:0000:655:a/112",
"2001:0000:0000:0000:0000:0000:655:b/112" ]
- host: 10.78.229.196
  datastore: DatastoreSSD-229-196
  addresses: ["10.78.229.222", "10.78.229.218", "10.78.229.215", "10.78.229.212"]
  addresses-v6: [ "2001:0000:0000:0000:0000:0000:655:c/112",
"2001:0000:0000:0000:0000:0000:655:d/112", "2001:0000:0000:0000:0000:0000:655:e/112",
"2001:0000:0000:0000:0000:0000:655:f/112" ]
apps:
- inode-manager:
  nodes:
  - host: 10.78.229.150
    nics:
    - "VLAN 175"
    ops:
    interfaces:
    -
      addresses:
      - 192.168.255.214/16
      - "2002::afaf:ffd6/112"
      vip: [ 192.168.255.211/16, "2002::afaf:ffd3/112" ]
      vrouter-id: 78
      routes:
      -
        dest:
        - 192.168.175.0/24

apps:
- inode-manager

```

aio-inode-manager-standby-config.yaml

```

deployers:
smi-deployer-147:
  address: 10.78.229.147/24
  datastore: DatastoreSSD-229-151
  datastore-folder: "ClusterDataStore"
  environment: chn-smi-inodemgr-lab
  gateway: "10.78.229.1"
  host: "10.78.229.151"
  private-key-file: inodemgr.pem
  username: cloud-user
  #ingress-hostname only supports valid FQDN
  ingress-hostname: deployer.example.com
  #Optional configuration
  docker-subnet-override:
  - pool-name: pool1
    base: 172.17.0.0/16
    size: 16

environments:
  chn-smi-inodemgr-lab:
  cluster: smi
  datacenter: CABU-VC65

```

```

datastore: DatastoreSSD-229-150
nameservers:
  - "172.30.131.10"
  - "172.16.128.140"
nics:
  - "VM Network"
ntp:
  - 8.ntp.esl.cisco.com
  - 2.ntp.esl.cisco.com
search-domains:
  - cisco.com
server: "10.78.229.250"
username: administrator@CABU.VCENTER60

clusters:
  "cicd-aio-229":
    type: opshub
    environment: "chn-smi-inodemgr-lab"
    username: "inodemgruser"
    gateway: "10.78.229.1"
    private-key-file: "inodemgr.pem"
    nodes:
      - host: 10.78.229.151
        datastore: DatastoreSSD-229-151
        datastore-folder: "ClusterDataStore"
        addresses: ["10.78.229.229/24"]
apps:
  - inode-manager

```

multinode-inode-manager-standby-config.yaml

```

deployers:
  smi-deployer-147:
    address: 10.78.229.147/24
    datastore: DatastoreSSD-229-151
    environment: chn-smi-inodemgr-lab
    gateway: "10.78.229.1"
    host: "10.78.229.151"
    private-key-file: inodemgr.pem
    username: cloud-user
    #ingress-hostname only supports valid FQDN
    ingress-hostname: deployer.example.com
    #Optional configuration
    docker-subnet-override:
      - pool-name: pool1
        base: 172.17.0.0/16
        size: 16

environments:
  chn-smi-inodemgr-lab:
    cluster: smi
    datacenter: CABU-VC65
    datastore: "DatastoreSSD-229-150 (1)"
    nameservers:
      - "172.30.131.10"
      - "172.16.128.140"
    nics:
      - "VM Network"
    ntp:
      - 8.ntp.esl.cisco.com
      - 2.ntp.esl.cisco.com
    search-domains:
      - cisco.com
    server: "10.78.229.250"

```

```

username: administrator@CABU.VCENTER60

clusters:
  "cicd-multi-node-211":
    type: opshub
    environment: "chn-smi-inodemgr-lab"
    username: "inodemgruser"
    gateway: "10.78.229.1"
    primary-vip: "10.78.229.211/23"
    vrouter-id: 78
    private-key-file: "inodemgr.pem"
    ingress-hostname: "inodemgr-chn08-dev01.cisco.com"
    enable-http-redirect: "true"
    pod-subnet: "192.168.120.0/24"
    service-subnet: "10.96.120.0/24"
    docker-bridge-subnet: ["172.17.0.0/16"]
    nodes:
      - host: 10.78.229.150
        datastore: "DatastoreSSD-229-150 (1)"
        addresses: ["10.78.229.217", "10.78.229.214", "10.78.229.224", "10.78.229.221"]
      - host: 10.78.229.151
        datastore: DatastoreSSD-229-151
        addresses: ["10.78.229.213", "10.78.229.216", "10.78.229.219", "10.78.229.223"]
      - host: 10.78.229.196
        datastore: DatastoreSSD-229-196
        addresses: ["10.78.229.222", "10.78.229.218", "10.78.229.215", "10.78.229.212"]
    apps:
      - inode-manager

```

Deploy the Cluster

Use the deploy script to deploy both the deployer and the cluster. Run the deploy command without any parameters to get the available options:

```

./deploy -c <config_file> [-v]
-c <config_file> : Configuration File, <Mandatory Argument>
-v              : Config Validation Flag, [Optional]
-f              : Day0: Force VM Redeploy Flag [Optional]
                : Day1: Force iNode Manager Update Flag [Optional]
-u              : Cluster Upgrade Flag [Optional]
-s              : Skip Compare Flag [Optional]
-i <install_opt> : Cluster installation options: deploy, redeploy, or upgrade [Optional]

```

The deploy script takes a configuration file with the '-c' option.

The deploy script uses the -u flag to update the deployer. When this flag is present, the script processes all the deployers in the `deployers` section in the config yml. The deploy script ignores the clusters in the `clusters` section.

For cluster installations, use one of the three options for the -i flag:

- **deploy**: this option is active when the -i <install_option> parameter is absent. In this mode, the deploy script first pings the cluster. If it is not pingable, the script deploys the cluster. Otherwise, the script does not perform any operations on the cluster.
- **redeploy**: In this mode, the deploy script first uninstalls the cluster, if it is already available. Then the script redeploys the new cluster.
- **upgrade**: In this mode, the deploy script upgrades the cluster with the software in the package.



Caution With the redeploy option, you lose all data in the original cluster.

For example, the following command installs the cluster using the configuration file `config.yaml`, assuming it does not exist:

```
$ ./deploy -c config.yaml
```



-
- Note**
- The deploy script invokes the docker command that requires the root permission to run. Depending on your setting, you may have to prepend `sudo` to the preceding command.
 - At once, either `deployer (-u)` / `cluster (-i)` – only one of the options can work. Both the options do not work in tandem.
-

The deploy script does the following operations:

If you are running the deploy script for the first time, it prompts you to enter all the passwords required for installation.

- For VMware vCenter environment:
 - vCenter password for the user specified in the environment config
- For deployer:
 - SSH password for the deployer's ops-center, for the user `cloud-user`
- For an iNode Manager cluster:
 - SSH password for all VMs in the cluster, for the user in the cluster's config (`inodemgruser` is the default user)
 - SSH passwords for the three ops-centers (iNode Manager, Operations Hub, and CEE), for the user `admin`



Note The deploy script prompts you twice to enter each password. The deploy script saves the passwords in the staging directory in encrypted form for future use.

- Passwords for the deployer, the cluster, and the Operation Centers must be eight characters long. The passwords must have a minimum of one lowercase letter, one uppercase letter, one numeric character, and one special character.
- The deploy script generates an SSH key pair when the `private-key-file` line is missing for the deployer or the cluster in the configuration file. The generated private key files are in the `.sec` sub directory under the staging directory, with `<cluster-name>_auto.pem` as the filename.
- The root-user owns the generated private keys. When logging in using SSH and these private key files, make sure that you run it with `sudo`.
- If the deployer is not running, the deploy script installs the deployer.

- The deploy script checks if the deployer is missing any of the product packages in the `offline-images` directory. If it finds any missing, it uploads them to the deployer.
- The script also generates the configuration for each cluster and pushes them to the deployer.
- The deploy script triggers the deployer to perform the sync operation for the cluster. The sync operation applies the configuration to the cluster. If you have not set up the cluster, it installs the cluster. Or the sync operation updates the cluster with the configuration.
- If the sync operation times out, the deploy script triggers the sync operation again. The script waits for the sync operation to complete. Then, it continues to monitor the cluster to ensure the deployment of all helm charts and creation of all pods.

You can repeat the deploy script to deploy more than one cluster by providing the corresponding configuration files. Alternatively, you can run this command appending a `-v` flag. The `-v` flag forces the deploy script to skip the sync operation and the remaining operations. Use this option to push the configuration of a cluster to the deployer without deploying or updating the cluster.

Sample Logs

The following example shows logs for the autodeployer.

```
[host]$ ./deploy -c examples/deployer-sample-config.yaml -v

Running autodeployer...

Day0 Configuration Detected
Validating config [environments]
Validating config [deployers]
Config Validated...
[vCenter:cabu-sdn-vc.cisco.com]$ Enter Password for cvideo.gen@cisco.com :
Re-Enter Password :

Create credentials for the deployer...inode-manager-deployer-1
Enter password for cloud-user@192.0.2.28 :
Re-Enter Password :

Gathering Product Images Info !!!

--- : Product Info : ---
cee                               : http://charts.192.0.2.28.nip.io/cee-2020-01-1-11
inode                              :
http://charts.192.0.2.28.nip.io/inode-manager-3.1.0-release-2007142325
opshub                             : http://charts.192.0.2.28.nip.io/opshub-release-2007150030

--- : cnBR Images : ---
cluster-manager-docker-deployer : cluster-manager-docker-deployer:1.0.3-0079-01a50dd
autodeploy                       : autodeploy:0.1.0-0407-2e073f8

--- : vCenter Info : ---
atl-smi-inodemgr-lab            : Cloud Video Datacenter, iNodeManager

--- : Deployer Info : ---
inode-manager-deployer-1       : IP -> 192.0.2.28/24, host -> 192.0.2.7

PING 192.0.2.28 (192.0.2.28) 56(84) bytes of data.

--- 192.0.2.28 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms

2020-08-03 12:08:51.753 INFO deploy: Parsing config file: .gen/tmp7n19eqji.json
```

```

2020-08-03 12:08:51.842 INFO deploy: Created ansible inventory yaml file

2020-08-03 12:08:51.842 INFO deploy: Config Directory is /opt/deployer/work and vmk file
is /opt/deployer/work/cluster-deployer-airgap.vmk:

2020-08-03 12:08:51.842 INFO deploy: Ansible inventory file:

    /tmp/tmpsetosj02/output_inventory.yaml

2020-08-03 12:08:51.842 INFO deploy: Running ansible to deploy and update VM. See vsphere
for progress: .gen/tmp7n19eqji.json

```

Troubleshooting

When you deploy a new deployer or a new Cisco iNode Manager cluster, make sure that the IP addresses and virtual machine (VM) names in the configuration file are not currently in use.

For deployers, the VM name is the same as the deployer name.

For single-node clusters, the VM name is the cluster-name with `-ops` appended.

For a multinode cluster, there are 12 VMs. Their names are cluster names with `-master-n`, `-etcd-n`, `-infra-n`, and `-ops-n` appended, where `n` is 1, 2, and 3.

Deploying a New Deployer

- Check if the VM is created on a vCenter.
- Log into the deployer VM using SSH with the correct username and private key file.


```
ssh -i <private-key-file> <deployer-user>@<deployer-address>
```
- Log into the deployer using deployer credentials setup during installation via the below command:


```
ssh admin@localhost -p 2022
```
- Check whether the product tar files available in the `offline-products` directory have been downloaded to the deployer:


```
software-package list
```

Overcoming Ansible Timeout Errors

To work around the ansible timeout errors and to avoid rerunning `sync` multiple times, perform the following steps.

1. SSH into the deployer VM and login to the `cluster-sync` container in the `ops-center-smi-cluster-deployer` pod

```
kubectrl exec -it ops-center-smi-cluster-deployer-xxxxxxxx-yyyy --container cluster-sync
-n smi bash
```

2. Increase the 30-second timeout to 120 seconds.

```
cd /opt/run/server/ansible
sed -i 's/timeout = 30/timeout = 120/' ansible.cfg
```

3. Open a new deployer CLI session and run the sync command.

```
clusters <cluster> actions sync run
```



Note If you recreate the deployer or restart the pod, you may lose the preceding configuration changes. In such cases, reapply the changes as necessary.

Deploying the Cisco iNode Manager

- Check if the configuration for iNode Manager clusters has been pushed to the deployer:

```
show running-config
```

- Monitor the deployment status from the deployer:

```
monitor sync-logs <cluster>
```

(Press Ctrl+C to quit monitoring)

- Check whether the VMs of the cluster are created on vCenter.
- Log into the cluster VMs using SSH to see if they are accessible.

For a single-node cluster, log into the `-ops` VM. For a multinode cluster, log into one of the control plane VMs using SSH with the correct username and the SSH private key file.

```
ssh -i <private-key-file> <cluster-user>@<vm-ip-address>
```

- Check the Kubernetes cluster using the `kubectl` command. For example, to check the status of all pods, use the following command:

```
kubectl get pod --all-namespaces
```

- When all pods are in `Running` state, you can log in to the iNode Manager user interface.

Helm chart deployment missing

After installation, check the output of the `helm ls -A` command.

| NAME | CHART | REVISION | UPDATED | APP VERSION |
|--------------------------------|---|----------|--------------------------|--------------|
| cee-data-cnat-monitoring | cnat-monitoring-0.6.0-0-6-0023-201013174135-9a71d44 | 3 | Wed Dec 16 06:08:17 2020 | 2020.01.1-16 |
| DEPLOYED | cee-data | | | |
| cee-data-ops-center | cee-ops-center-0.6.0-0-6-0256-201008225538-5470620 | 5 | Wed Dec 16 06:04:40 2020 | 2020.01.1-16 |
| DEPLOYED | cee-data | | | |
| cee-data-product-documentation | product-documentation-0.6.0-0-6-0038-200910021447-5adb52c | 3 | Wed Dec 16 06:08:15 2020 | 2020.01.1-16 |
| DEPLOYED | cee-data | | | |
| cee-data-pv-manager | pv-manager-0.2.0-0-3-0011-200913183355-60e70dd | 3 | Wed Dec 16 06:08:15 2020 | 2020.01.1-16 |
| DEPLOYED | cee-data | | | |
| cee-data-smi-autoheal | | 2 | Wed Dec 16 06:08:16 2020 | |

```

DEPLOYED      smi-autoheal-0.2.0-master-0009-201001205725-0b34f83      2020.01.1-16
                cee-data
cee-data-smi-show-tac      3      Wed Dec 16 06:08:16 2020
DEPLOYED      smi-show-tac-0.2.0-0-2-0115-200909130841-b3cd71b      2020.01.1-16
                cee-data
cee-data-storage-provisioner      3      Wed Dec 16 06:08:15 2020
DEPLOYED      storage-provisioner-0.3.0-0-3-0083-201001203003-3922e70      2020.01.1-16
                cee-data
inode-manager-data-inode-manager-app      5      Wed Dec 16 06:07:59 2020
DEPLOYED      inode-manager-app-3.1.0-main-0010-201124065212-8fedd73
inodemanager-3.1.0-release      inode-manager-data
inode-manager-data-ops-center      5      Wed Dec 16 06:04:50 2020
DEPLOYED      inode-manager-ops-center-0.1.0-main-0022-201118083544-881...
inodemanager-3.1.0-release      inode-manager-data
kubernetes-dashboard      3      Wed Dec 16 06:04:23 2020
DEPLOYED      kubernetes-dashboard-1.10.1-master-0013-190605174754-8d7080d      1.10.1
                kube-system
nginx-ingress      3      Wed Dec 16 06:04:20 2020
DEPLOYED      nginx-ingress-1.5.0-master-0078-200417033703-5484f87      0.26.1
                nginx-ingress
opshub-data-ops-center      5      Wed Dec 16 06:04:59 2020
DEPLOYED      opshub-ops-center-0.5.3-smartphy-0052-201029172812-4a0b973
opshub-3.0.4-release      opshub-data
opshub-data-opshub-infra-app      1      Wed Dec 16 06:25:44 2020
DEPLOYED      opshub-infra-app-0.1.0-main-0048-201029234800-2eb8f1f
opshub-3.0.4-release      opshub-data
smi-cluster-maintainer      3      Wed Dec 16 06:04:16 2020
DEPLOYED      smi-cluster-maintainer-1.1.0-master-0005-200324060503-218...
                kube-system
smi-keepalived-vips      4      Wed Dec 16 06:04:26 2020
DEPLOYED      smi-keepalived-1.0.0-master-0061-200414235846-e656df5
                smi-vips
ss-cert-prov      3      Wed Dec 16 06:04:12 2020
DEPLOYED      self-signed-cert-provisioner-1.0.0-master-0018-2004091602...
                smi-certs

```

If any of the above chart is not listed/failed to deploy, try to re-create the cluster freshly using the following command.

```
clusters <<cluster-name>> actions sync run debug true force-vm-redeploy true purge-data-disks true
```

If the problem persists upon re-creation of the cluster, it requires a manual installation of the failed helm chart(s).

Rename the Configuration Profile with the Special Character

Before iNode manager 3.2.0 release, you can create a configuration profile with the special character '+' in the name. But the support for the special character is removed in iNode manager 3.2.0 release.

If the special character '+' is present in the name of the configuration profile, it will be replaced by the special character '-' in the background when you open the Config Profiles tab in the iNode manager UI.

iNode Manager Web Interface Access

To access the iNode Manager web interface, use one of the following URLs:

- With **FQDN enabled**: `https://ingress-hostname`

`ingress-hostname` is the DNS name (FQDN) if configured. For example:

```
https://inodemgr.example.com
```

- With **FQDN disabled**: `https://ingress-ip.nip.io`

For AIO - the `ingress IP` is the management IP of the `-ops` VM.

For Multinode - the `ingress IP` is the primary-virtual IP configured for the management network.

Use the following credentials to log in.

```
Username: admin
Password: <password configured for "inode-manager" ops-center>
```



Note The default password to access UI expires after 6 months. Ensure the password is renewed before expiry or reset the expired password by following the steps in [Reset Admin Password](#).

Post Installation/Upgrade Checklist

Step 1 Check NTP Time Sync.

- For multinode deployment, check if all the 12 nodes in the cluster are in time-sync. If not, execute the following command on the nodes that are not in sync.

```
sudo service chronyd restart
```

Step 2 Check helm charts.

- Check if all helm charts are in "DEPLOYED" state.

Example:

```
helm ls -A
APP VERSION
cee-data-cnat-monitoring cee-data 1 2021-08-09
10:18:45.953440052 +0000 UTC deployed
cnat-monitoring-0.7.1-2020-02-3-0034-210617032059-af5a8ce 2020.02.3.i14
cee-data-ops-center cee-data 2 2021-08-09
12:02:11.540037018 +0000 UTC deployed
cee-ops-center-0.7.1-2020-02-3-0431-210625205136-664b1b4 2020.02.3.i14
cee-data-product-documentation cee-data 1 2021-08-09
10:18:06.497866693 +0000 UTC deployed
product-documentation-0.7.1-2020-02-3-0048-210617012614-7619b76 2020.02.3.i14
cee-data-pv-manager cee-data 1 2021-08-09
10:18:06.497416732 +0000 UTC deployed
pv-manager-0.2.1-2020-02-3-0015-210617012645-e25720c
2020.02.3.i14
cee-data-smi-autoheal cee-data 1 2021-08-09
10:18:06.548581132 +0000 UTC deployed
smi-autoheal-0.2.1-2020-02-3-0021-210617012113-c0a7c37
2020.02.3.i14
cee-data-smi-show-tac cee-data 1 2021-08-09
10:18:06.592350238 +0000 UTC deployed
smi-show-tac-0.3.1-2020-02-3-0162-210621135943-3b74a43
2020.02.3.i14
cee-data-storage-provisioner cee-data 1 2021-08-09
10:18:06.56561449 +0000 UTC deployed
storage-provisioner-0.3.1-2020-02-3-0089-210617012245-1cef60 2020.02.3.i14
distributed-registry registry 1 2021-08-09
10:08:32.456224738 +0000 UTC deployed
distributed-registry-0.2.0-2020-02-3-0046-210616223348-dc09e48
```

```

inode-manager-data-inode-manager-app      inode-manager-data      1      2021-08-09
10:17:59.8987974 +0000 UTC deployed      inode-manager-app-3.2.0-3-3-0-0013-210809045220-7391cff
      inodemanager-3.3.0-release
inode-manager-data-ops-center             inode-manager-data      2      2021-08-09
12:02:21.636540821 +0000 UTC deployed
inode-manager-ops-center-0.1.0-3-3-0-0031-210805133056-ace980d      inodemanager-3.3.0-release
inode-manager-disk-util                   inode-manager-data      1      2021-08-24
06:20:28.150553921 +0000 UTC deployed
inode-manager-utils-1.0.0-disk-util-0003-210824061029-657e2d4      3.0.0
nginx-ingress                             nginx-ingress           1      2021-08-09
10:08:06.434442445 +0000 UTC deployed    nginx-ingress-1.6.0-2020-02-3-0092-210616222705-cb12ba6
      0.47.0
opshub-data-ops-center                   opshub-data            1      2021-08-09
12:02:32.163637776 +0000 UTC deployed
opshub-ops-center-0.5.3-stable-0113-210624222226-41ff839      21.3.0.6
opshub-data-opshub-infra-app             opshub-data            1      2021-08-09
12:03:06.551311613 +0000 UTC deployed    opshub-infra-app-0.1.0-stable-0087-210623165712-a9b6662
      21.3.0.6
smi-cluster-maintainer                   kube-system             1      2021-08-09
10:08:01.416040644 +0000 UTC deployed
smi-cluster-maintainer-1.1.1-2020-02-3-0022-210625135311-a6514dc
smi-keepalived-vips                      smi-vips                1      2021-08-09
10:08:12.135698028 +0000 UTC deployed
smi-keepalived-1.1.0-2020-02-3-0090-210624055703-3ce7966
smi-secure-access                        smi-secure-access      1      2021-08-09
10:08:16.124812544 +0000 UTC deployed
smi-secure-access-0.1.0-2020-02-3-0013-210616214456-4641407
ss-cert-prov                             smi-certs               1      2021-08-09
10:07:56.792775786 +0000 UTC deployed
self-signed-cert-provisioner-1.1.0-2020-02-3-0040-210616220407-a528f9c

```

- b) If any of the charts are in "FAILED" state, delete the charts and run the sync command from the deployer CLI again. The failure is due to a temporary timeout issue which resolves on retrying.

```

helm delete <failed-chart-name> -n <namespace of the failed chart>
(from deployer cli) clusters <cluster> actions sync run

```

Step 3 Enable SMI Log Forwarder.

For multinode deployment, you can enable centralized logging on Elasticsearch.

To enable log-forwarding (which is disabled by default on deployment), login to CEE Ops center - <https://cli.cee-data-ops-center.<ingress-ip>.nip.io/>. Use the username (admin) and password that is configured during deployment and perform the following steps.

- a) Enter config terminal.

```

[inode-manager-multinode/data] cee# config terminal
Entering configuration mode terminal

```

- b) Set the following logging config.

```

[inode-manager-multinode/data] cee(config)# logging fluent host fluentd.opshub-data port 24224
disable-tls true

```

- c) Commit and exit.

```

[inode-manager-multinode/data] cee(config)# commit
Commit complete.
[inode-manager-multinode/data] cee(config)# exit

```

When the log forwarder is enabled, you see the following messages on the CLI.

```

[inode-manager-multinode/data] cee#
Message from confd-api-manager at 2020-08-06 16:02:16...

```

```
Helm update is STARTING. Trigger for update is STARTUP.
[inode-manager-multinode/data] cee#
Message from confd-api-manager at 2020-08-06 16:02:16...
System is current running at 98.85
[inode-manager-multinode/data] cee#
Message from confd-api-manager at 2020-08-06 16:02:18...
Helm update is SUCCESS. Trigger for update is STARTUP.
[inode-manager-multinode/data] cee#
Message from confd-api-manager at 2020-08-06 16:02:18...
System is current running at 98.86
[inode-manager-multinode/data] cee#
```

Step 4 Update ARP cache.

Note For large iNode Manager deployments - that manages more than 500 iNodes, we recommend updating the default ARP cache size to avoid connectivity errors with iNodes.

We recommend the following set of values for managing 30k iNodes.

```
net.ipv4.neigh.default.gc_thresh3 = 32768
net.ipv4.neigh.default.gc_thresh2 = 16384
net.ipv4.neigh.default.gc_thresh1 = 4096
net.ipv6.neigh.default.gc_thresh3 = 32768
net.ipv6.neigh.default.gc_thresh2 = 16384
net.ipv6.neigh.default.gc_thresh1 = 4096
net.core.somaxconn = 65535
```

Update ARP cache by appending the preceding values in the `/etc/sysctl.conf` file and executing the following command.

```
sudo sysctl -p
```

Note Update ARP cache after you recreating/updating the VM.

Log Examples

This section contains the example logs that you can use as references in your iNode Manager installation.

Log Example for Deployer Installation

```
13:31:00-29-root-INFO: Start logging for cnBR/Opshub automatic offline installation:
13:31:37-857-AUTO-DEPLOY-INFO: [32m
--- : Product Info : ---[0m
13:31:37-857-AUTO-DEPLOY-INFO: [34mcee :
http://charts.10.90.154.28.nip.io/cee-2020-01-1-11[0m
13:31:37-858-AUTO-DEPLOY-INFO: [34minode :
http://charts.10.90.154.28.nip.io/inode-manager-3.0.0-release-2007142325[0m
13:31:37-858-AUTO-DEPLOY-INFO: [34mopshub :
http://charts.10.90.154.28.nip.io/opshub-release-2007150030[0m
13:31:37-858-AUTO-DEPLOY-INFO: [32m
--- : cnBR Images : ---[0m
13:31:37-858-AUTO-DEPLOY-INFO: [34mcluster-manager-docker-deployer :
cluster-manager-docker-deployer:1.0.3-0079-01a50dd[0m
13:31:37-858-AUTO-DEPLOY-INFO: [34mautodeploy :
autodeploy:0.1.0-0407-2e073f8[0m
13:31:37-859-AUTO-DEPLOY-INFO: [32m
--- : vCenter Info : ---[0m
```



```
13:31:37-859-AUTO-DEPLOY-INFO: [34matl-smi-inodemgr-lab          : Cloud Video Datacenter,
iNodeManager[0m
13:31:37-859-AUTO-DEPLOY-INFO: [32m
--- : Deployer Info : ---[0m
13:31:37-859-AUTO-DEPLOY-INFO: [34minode-manager-deployer-1    : IP -> 10.90.154.28/24,
host -> 10.90.154.7[0m
13:31:37-859-AUTO-DEPLOY-INFO:
13:31:49-102-AUTO-DEPLOY-INFO: 2020-08-03 13:31:49.102 INFO deploy: Parsing config file:
.gen/tmp7_la094u.json

13:31:49-136-AUTO-DEPLOY-INFO: 2020-08-03 13:31:49.136 INFO deploy: Created ansible inventory
yaml file

13:31:49-137-AUTO-DEPLOY-INFO: 2020-08-03 13:31:49.136 INFO deploy: Config Directory is
/opt/deployer/work and vmdk file is /opt/deployer/work/cluster-deployer-airgap.vmdk:

13:31:49-137-AUTO-DEPLOY-INFO: 2020-08-03 13:31:49.136 INFO deploy: Ansible inventory file:

13:31:49-137-AUTO-DEPLOY-INFO: /tmp/tmpy6huxl8r/output_inventory.yaml

13:31:49-137-AUTO-DEPLOY-INFO: 2020-08-03 13:31:49.136 INFO deploy: Running ansible to
deploy and update VM. See vsphere for progress: .gen/tmp7_la094u.json

13:56:20-963-AUTO-DEPLOY-INFO:

13:56:20-963-AUTO-DEPLOY-INFO: PLAY [Create VM]
*****

13:56:20-963-AUTO-DEPLOY-INFO:

13:56:20-963-AUTO-DEPLOY-INFO: TASK [Gathering Facts]
*****

13:56:20-964-AUTO-DEPLOY-INFO: Monday 03 August 2020 13:31:50 +0000 (0:00:00.220)
0:00:00.220 *****

13:56:20-964-AUTO-DEPLOY-INFO: ok: [cluster_manager]

13:56:20-964-AUTO-DEPLOY-INFO:

13:56:20-964-AUTO-DEPLOY-INFO: TASK [vm-vsphere : set common variables]
*****

13:56:20-964-AUTO-DEPLOY-INFO: Monday 03 August 2020 13:31:51 +0000 (0:00:01.039)
0:00:01.259 *****

13:56:20-964-AUTO-DEPLOY-INFO: ok: [cluster_manager]

13:56:20-964-AUTO-DEPLOY-INFO:

13:56:20-964-AUTO-DEPLOY-INFO: TASK [vm-vsphere : Set hostname fact (override)]
*****

13:56:20-964-AUTO-DEPLOY-INFO: Monday 03 August 2020 13:31:51 +0000 (0:00:00.067)
0:00:01.327 *****

13:56:20-964-AUTO-DEPLOY-INFO: ok: [cluster_manager]

13:56:20-964-AUTO-DEPLOY-INFO:

13:56:20-965-AUTO-DEPLOY-INFO: TASK [vm-vsphere : Set hostname fact (other)]
*****
```

```
13:56:20-965-AUTO-DEPLOY-INFO: Monday 03 August 2020 13:31:51 +0000 (0:00:00.061)
0:00:01.388 *****

13:56:20-965-AUTO-DEPLOY-INFO: skipping: [cluster_manager]

13:56:20-965-AUTO-DEPLOY-INFO:

13:56:20-965-AUTO-DEPLOY-INFO: TASK [vm-vsphere : Debug]
*****

13:56:20-965-AUTO-DEPLOY-INFO: Monday 03 August 2020 13:31:51 +0000 (0:00:00.030)
0:00:01.419 *****

13:56:20-965-AUTO-DEPLOY-INFO: ok: [cluster_manager] =>

13:56:20-965-AUTO-DEPLOY-INFO: msg: |-

13:56:20-965-AUTO-DEPLOY-INFO: user_id: root

13:56:20-965-AUTO-DEPLOY-INFO: server: cabu-sdn-vc.cisco.com

13:56:20-965-AUTO-DEPLOY-INFO: port: 443

13:56:20-966-AUTO-DEPLOY-INFO: allow-self-signed-cert: True

13:56:20-966-AUTO-DEPLOY-INFO: user: cvideo.gen@cisco.com

13:56:20-966-AUTO-DEPLOY-INFO: datastore: datastore1 (1)

13:56:20-966-AUTO-DEPLOY-INFO: cluster: iNodeManager

13:56:20-966-AUTO-DEPLOY-INFO: nics: [{'network-name': 'VM Network'}]

13:56:20-966-AUTO-DEPLOY-INFO: datacenter: Cloud Video Datacenter

13:56:20-966-AUTO-DEPLOY-INFO: host: 10.90.154.7

13:56:20-966-AUTO-DEPLOY-INFO:

13:56:20-966-AUTO-DEPLOY-INFO: TASK [vm-vsphere : Test vCenter credentials are valid]
*****

13:56:20-966-AUTO-DEPLOY-INFO: Monday 03 August 2020 13:31:51 +0000 (0:00:00.060)
0:00:01.479 *****

13:56:20-966-AUTO-DEPLOY-INFO: ok: [cluster_manager]

13:56:20-966-AUTO-DEPLOY-INFO:

13:56:20-967-AUTO-DEPLOY-INFO: TASK [vm-vsphere : Get VM Update needed]
*****

13:56:20-967-AUTO-DEPLOY-INFO: Monday 03 August 2020 13:31:53 +0000 (0:00:02.026)
0:00:03.506 *****

13:56:20-967-AUTO-DEPLOY-INFO: ok: [cluster_manager]

13:56:20-967-AUTO-DEPLOY-INFO:

13:56:20-967-AUTO-DEPLOY-INFO: TASK [vm-vsphere : set vm_update_needed set_fact]
*****

13:56:20-967-AUTO-DEPLOY-INFO: Monday 03 August 2020 13:32:06 +0000 (0:00:12.980)
0:00:16.487 *****
```

```
13:56:20-967-AUTO-DEPLOY-INFO: ok: [cluster_manager]
13:56:20-967-AUTO-DEPLOY-INFO:
13:56:20-967-AUTO-DEPLOY-INFO: TASK [vm-vsphere : Ensure temp directory exists]
*****
13:56:20-967-AUTO-DEPLOY-INFO: Monday 03 August 2020 13:32:06 +0000 (0:00:00.055)
0:00:16.543 *****
13:56:20-967-AUTO-DEPLOY-INFO: changed: [cluster_manager]
13:56:20-967-AUTO-DEPLOY-INFO:
13:56:20-967-AUTO-DEPLOY-INFO: TASK [vm-vsphere : create netplan Template]
*****
13:56:20-968-AUTO-DEPLOY-INFO: Monday 03 August 2020 13:32:06 +0000 (0:00:00.125)
0:00:16.668 *****
13:56:20-968-AUTO-DEPLOY-INFO: changed: [cluster_manager]
13:56:20-968-AUTO-DEPLOY-INFO:
13:56:20-968-AUTO-DEPLOY-INFO: TASK [vm-vsphere : create ssh public key file]
*****
13:56:20-968-AUTO-DEPLOY-INFO: Monday 03 August 2020 13:32:07 +0000 (0:00:00.148)
0:00:16.816 *****
13:56:20-968-AUTO-DEPLOY-INFO: changed: [cluster_manager]
13:56:20-968-AUTO-DEPLOY-INFO:
13:56:20-968-AUTO-DEPLOY-INFO: TASK [vm-vsphere : Create user data ISO]
*****
13:56:20-968-AUTO-DEPLOY-INFO: Monday 03 August 2020 13:32:07 +0000 (0:00:00.082)
0:00:16.899 *****
13:56:20-968-AUTO-DEPLOY-INFO: changed: [cluster_manager]
13:56:20-968-AUTO-DEPLOY-INFO:
13:56:20-968-AUTO-DEPLOY-INFO: TASK [vm-vsphere : Check if VMs Folder exists]
*****
13:56:20-968-AUTO-DEPLOY-INFO: Monday 03 August 2020 13:32:07 +0000 (0:00:00.341)
0:00:17.241 *****
13:56:20-969-AUTO-DEPLOY-INFO: skipping: [cluster_manager]
13:56:20-969-AUTO-DEPLOY-INFO:
13:56:20-969-AUTO-DEPLOY-INFO: TASK [vm-vsphere : Check if VM Template exists]
*****
13:56:20-969-AUTO-DEPLOY-INFO: Monday 03 August 2020 13:32:07 +0000 (0:00:00.036)
0:00:17.278 *****
13:56:20-969-AUTO-DEPLOY-INFO: ok: [cluster_manager]
13:56:20-969-AUTO-DEPLOY-INFO:
```

```
13:56:20-969-AUTO-DEPLOY-INFO: TASK [vm-vsphere : Upload VM Template]
*****

13:56:20-969-AUTO-DEPLOY-INFO: Monday 03 August 2020 13:32:20 +0000 (0:00:12.540)
0:00:29.818 *****

13:56:20-969-AUTO-DEPLOY-INFO: changed: [cluster_manager]

13:56:20-969-AUTO-DEPLOY-INFO:

13:56:20-969-AUTO-DEPLOY-INFO: TASK [vm-vsphere : Create VM]
*****

13:56:20-969-AUTO-DEPLOY-INFO: Monday 03 August 2020 13:47:51 +0000 (0:15:31.814)
0:16:01.633 *****

13:56:20-969-AUTO-DEPLOY-INFO: changed: [cluster_manager]

13:56:20-970-AUTO-DEPLOY-INFO:

13:56:20-970-AUTO-DEPLOY-INFO: TASK [vm-vsphere : Wait for ssh]
*****

13:56:20-970-AUTO-DEPLOY-INFO: Monday 03 August 2020 13:48:48 +0000 (0:00:56.169)
0:16:57.803 *****

13:56:20-970-AUTO-DEPLOY-INFO: ok: [cluster_manager]

13:56:20-970-AUTO-DEPLOY-INFO:

13:56:20-970-AUTO-DEPLOY-INFO: PLAY [Init K3s]
*****

13:56:20-970-AUTO-DEPLOY-INFO:

13:56:20-970-AUTO-DEPLOY-INFO: TASK [init-k3s : Ensure /data folder exists]
*****

13:56:20-970-AUTO-DEPLOY-INFO: Monday 03 August 2020 13:50:44 +0000 (0:01:56.187)
0:18:53.991 *****

13:56:20-970-AUTO-DEPLOY-INFO: changed: [cluster_manager]

13:56:20-970-AUTO-DEPLOY-INFO:

13:56:20-970-AUTO-DEPLOY-INFO: TASK [init-k3s : Copy config]
*****

13:56:20-970-AUTO-DEPLOY-INFO: Monday 03 August 2020 13:50:44 +0000 (0:00:00.629)
0:18:54.620 *****

13:56:20-971-AUTO-DEPLOY-INFO: changed: [cluster_manager]

13:56:20-971-AUTO-DEPLOY-INFO:

13:56:20-971-AUTO-DEPLOY-INFO: TASK [init-k3s : Init k3s]
*****

13:56:20-971-AUTO-DEPLOY-INFO: Monday 03 August 2020 13:50:44 +0000 (0:00:00.095)
0:18:54.716 *****

13:56:20-971-AUTO-DEPLOY-INFO: ok: [cluster_manager]
```

```
13:56:20-971-AUTO-DEPLOY-INFO:
13:56:20-971-AUTO-DEPLOY-INFO: PLAY [Install NTP]
*****
13:56:20-971-AUTO-DEPLOY-INFO:
13:56:20-971-AUTO-DEPLOY-INFO: TASK [install-ntp : set chrony ntp server facts]
*****
13:56:20-971-AUTO-DEPLOY-INFO: Monday 03 August 2020 13:51:07 +0000 (0:00:22.967)
0:19:17.683 *****
13:56:20-971-AUTO-DEPLOY-INFO: ok: [cluster_manager]
13:56:20-971-AUTO-DEPLOY-INFO:
13:56:20-971-AUTO-DEPLOY-INFO: TASK [install-ntp : Check smi ingresses]
*****
13:56:20-972-AUTO-DEPLOY-INFO: Monday 03 August 2020 13:51:07 +0000 (0:00:00.061)
0:19:17.744 *****
13:56:20-972-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (300 retries left).
13:56:20-972-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (299 retries left).
13:56:20-972-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (298 retries left).
13:56:20-972-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (297 retries left).
13:56:20-972-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (296 retries left).
13:56:20-972-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (295 retries left).
13:56:20-972-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (294 retries left).
13:56:20-972-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (293 retries left).
13:56:20-972-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (292 retries left).
13:56:20-972-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (291 retries left).
13:56:20-972-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (290 retries left).
13:56:20-973-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (289 retries left).
13:56:20-973-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (288 retries left).
13:56:20-973-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (287 retries left).
13:56:20-973-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (286 retries left).
13:56:20-973-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (285 retries left).
13:56:20-973-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (284 retries left).
13:56:20-973-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (283 retries left).
13:56:20-973-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (282 retries left).
13:56:20-973-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (281 retries left).
13:56:20-973-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (280 retries left).
```

```
13:56:20-973-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (279 retries left).
13:56:20-973-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (278 retries left).
13:56:20-973-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (277 retries left).
13:56:20-974-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (276 retries left).
13:56:20-974-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (275 retries left).
13:56:20-974-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (274 retries left).
13:56:20-974-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (273 retries left).
13:56:20-974-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (272 retries left).
13:56:20-974-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (271 retries left).
13:56:20-974-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (270 retries left).
13:56:20-974-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (269 retries left).
13:56:20-974-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (268 retries left).
13:56:20-974-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (267 retries left).
13:56:20-974-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (266 retries left).
13:56:20-975-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (265 retries left).
13:56:20-975-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (264 retries left).
13:56:20-975-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (263 retries left).
13:56:20-975-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (262 retries left).
13:56:20-975-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (261 retries left).
13:56:20-975-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (260 retries left).
13:56:20-975-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (259 retries left).
13:56:20-975-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (258 retries left).
13:56:20-975-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (257 retries left).
13:56:20-975-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (256 retries left).
13:56:20-975-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (255 retries left).
13:56:20-975-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (254 retries left).
13:56:20-975-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (253 retries left).
13:56:20-976-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (252 retries left).
13:56:20-976-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (251 retries left).
13:56:20-976-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (250 retries left).
13:56:20-976-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (249 retries left).
13:56:20-976-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (248 retries left).
```

```
13:56:20-976-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (247 retries left).
13:56:20-976-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (246 retries left).
13:56:20-976-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (245 retries left).
13:56:20-976-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (244 retries left).
13:56:20-976-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (243 retries left).
13:56:20-976-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (242 retries left).
13:56:20-976-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (241 retries left).
13:56:20-976-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (240 retries left).
14:07:06-354-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (239 retries left).
14:07:06-355-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (238 retries left).
14:07:06-355-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (237 retries left).
14:07:06-355-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (236 retries left).
14:07:06-355-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (235 retries left).
14:07:06-355-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (234 retries left).
14:07:06-355-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (233 retries left).
14:07:06-355-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (232 retries left).
14:07:06-355-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (231 retries left).
14:07:06-355-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (230 retries left).
14:07:06-355-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (229 retries left).
14:07:06-355-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (228 retries left).
14:07:06-355-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (227 retries left).
14:07:06-356-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (226 retries left).
14:07:06-356-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (225 retries left).
14:07:06-356-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (224 retries left).
14:07:06-356-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (223 retries left).
14:07:06-356-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (222 retries left).
14:07:06-356-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (221 retries left).
14:07:06-356-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (220 retries left).
14:07:06-356-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (219 retries left).
14:07:06-356-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (218 retries left).
14:07:06-356-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (217 retries left).
14:07:06-356-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (216 retries left).
```

```
14:07:06-356-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (215 retries left).
14:07:06-356-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (214 retries left).
14:07:06-357-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (213 retries left).
14:07:06-357-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (212 retries left).
14:07:06-357-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (211 retries left).
14:07:06-357-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (210 retries left).
14:07:06-357-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (209 retries left).
14:07:06-357-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (208 retries left).
14:07:06-357-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (207 retries left).
14:07:06-357-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (206 retries left).
14:07:06-357-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (205 retries left).
14:07:06-357-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (204 retries left).
14:07:06-357-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (203 retries left).
14:07:06-357-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (202 retries left).
14:07:06-358-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (201 retries left).
14:07:06-358-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (200 retries left).
14:07:06-358-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (199 retries left).
14:07:06-358-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (198 retries left).
14:07:06-358-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (197 retries left).
14:07:06-358-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (196 retries left).
14:07:06-358-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (195 retries left).
14:07:06-358-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (194 retries left).
14:07:06-358-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (193 retries left).
14:07:06-358-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (192 retries left).
14:07:06-358-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (191 retries left).
14:07:06-358-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (190 retries left).
14:07:06-359-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (189 retries left).
14:07:06-359-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (188 retries left).
14:07:06-359-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (187 retries left).
14:07:06-359-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (186 retries left).
14:07:06-359-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (185 retries left).
14:07:06-359-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (184 retries left).
```



```
14:07:06-359-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (183 retries left).
14:07:06-359-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (182 retries left).
14:07:06-359-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (181 retries left).
14:07:06-359-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (180 retries left).
14:07:06-359-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (179 retries left).
14:07:06-359-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (178 retries left).
14:07:06-359-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (177 retries left).
14:07:06-359-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (176 retries left).
14:07:06-360-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (175 retries left).
14:07:06-360-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (174 retries left).
14:07:06-360-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (173 retries left).
14:07:06-360-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (172 retries left).
14:07:06-360-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (171 retries left).
14:07:06-360-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (170 retries left).
14:07:06-360-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (169 retries left).
14:07:06-360-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (168 retries left).
14:07:06-360-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (167 retries left).
14:07:06-360-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (166 retries left).
14:07:06-360-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (165 retries left).
14:07:06-360-AUTO-DEPLOY-INFO: RETRYING: Check smi ingresses (164 retries left).
14:07:06-361-AUTO-DEPLOY-INFO: ok: [cluster_manager]
14:07:06-361-AUTO-DEPLOY-INFO:
14:07:06-361-AUTO-DEPLOY-INFO: TASK [install-ntp : Add the url to the hosts file]
*****
14:07:06-361-AUTO-DEPLOY-INFO: Monday 03 August 2020 14:02:53 +0000 (0:11:45.761)
0:31:03.506 *****
14:07:06-361-AUTO-DEPLOY-INFO: changed: [cluster_manager]
14:07:06-361-AUTO-DEPLOY-INFO:
14:07:06-361-AUTO-DEPLOY-INFO: TASK [install-ntp : Check ingress url]
*****
14:07:06-361-AUTO-DEPLOY-INFO: Monday 03 August 2020 14:02:53 +0000 (0:00:00.179)
0:31:03.686 *****
14:07:06-361-AUTO-DEPLOY-INFO: RETRYING: Check ingress url (120 retries left).
14:07:06-361-AUTO-DEPLOY-INFO: RETRYING: Check ingress url (119 retries left).
```

```
14:07:06-361-AUTO-DEPLOY-INFO: RETRYING: Check ingress url (118 retries left).
14:07:06-361-AUTO-DEPLOY-INFO: RETRYING: Check ingress url (117 retries left).
14:07:06-361-AUTO-DEPLOY-INFO: RETRYING: Check ingress url (116 retries left).
14:07:06-362-AUTO-DEPLOY-INFO: RETRYING: Check ingress url (115 retries left).
14:07:06-362-AUTO-DEPLOY-INFO: RETRYING: Check ingress url (114 retries left).
14:07:06-362-AUTO-DEPLOY-INFO: RETRYING: Check ingress url (113 retries left).
14:07:06-362-AUTO-DEPLOY-INFO: RETRYING: Check ingress url (112 retries left).
14:07:06-362-AUTO-DEPLOY-INFO: RETRYING: Check ingress url (111 retries left).
14:07:06-362-AUTO-DEPLOY-INFO: RETRYING: Check ingress url (110 retries left).
14:07:06-362-AUTO-DEPLOY-INFO: RETRYING: Check ingress url (109 retries left).
14:07:06-362-AUTO-DEPLOY-INFO: RETRYING: Check ingress url (108 retries left).
14:07:06-362-AUTO-DEPLOY-INFO: RETRYING: Check ingress url (107 retries left).
14:07:06-362-AUTO-DEPLOY-INFO: RETRYING: Check ingress url (106 retries left).
14:07:06-362-AUTO-DEPLOY-INFO: RETRYING: Check ingress url (105 retries left).
14:07:06-362-AUTO-DEPLOY-INFO: RETRYING: Check ingress url (104 retries left).
14:07:06-362-AUTO-DEPLOY-INFO: RETRYING: Check ingress url (103 retries left).
14:07:06-363-AUTO-DEPLOY-INFO: RETRYING: Check ingress url (102 retries left).
14:07:06-363-AUTO-DEPLOY-INFO: RETRYING: Check ingress url (101 retries left).
14:07:06-363-AUTO-DEPLOY-INFO: RETRYING: Check ingress url (100 retries left).
14:07:06-363-AUTO-DEPLOY-INFO: changed: [cluster_manager]
14:07:06-363-AUTO-DEPLOY-INFO:
14:07:06-363-AUTO-DEPLOY-INFO: TASK [install-ntp : Remove "ntp" package]
*****
14:07:06-363-AUTO-DEPLOY-INFO: Monday 03 August 2020 14:05:02 +0000 (0:02:08.790)
0:33:12.476 *****
14:07:06-363-AUTO-DEPLOY-INFO: ok: [cluster_manager]
14:07:06-363-AUTO-DEPLOY-INFO:
14:07:06-363-AUTO-DEPLOY-INFO: TASK [install-ntp : Cleaning cache]
*****
14:07:06-363-AUTO-DEPLOY-INFO: Monday 03 August 2020 14:05:04 +0000 (0:00:01.451)
0:33:13.928 *****
14:07:06-363-AUTO-DEPLOY-INFO: changed: [cluster_manager]
14:07:06-363-AUTO-DEPLOY-INFO:
14:07:06-364-AUTO-DEPLOY-INFO: TASK [install-ntp : Install offline APT repo GPG key]
```

14:07:06-364-AUTO-DEPLOY-INFO: Monday 03 August 2020 14:05:04 +0000 (0:00:00.309)
0:33:14.237 *****

14:07:06-364-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (60 retries left).
14:07:06-364-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (59 retries left).
14:07:06-364-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (58 retries left).
14:07:06-364-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (57 retries left).
14:07:06-364-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (56 retries left).
14:07:06-364-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (55 retries left).
14:07:06-364-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (54 retries left).
14:07:06-364-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (53 retries left).
14:07:06-364-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (52 retries left).
14:07:06-365-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (51 retries left).
14:07:06-365-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (50 retries left).
14:07:06-365-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (49 retries left).
14:07:06-365-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (48 retries left).
14:07:06-365-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (47 retries left).
14:07:06-365-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (46 retries left).
14:07:06-365-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (45 retries left).
14:07:06-365-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (44 retries left).
14:07:06-365-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (43 retries left).
14:07:06-365-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (42 retries left).
14:07:06-365-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (41 retries left).
14:07:06-365-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (40 retries left).
14:07:06-365-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (39 retries left).
14:07:06-366-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (38 retries left).
14:31:18-724-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (37 retries left).
14:31:18-724-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (36 retries left).
14:31:18-725-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (35 retries left).
14:31:18-725-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (34 retries left).
14:31:18-725-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (33 retries left).
14:31:18-725-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (32 retries left).
14:31:18-725-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (31 retries left).

```
14:31:18-725-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (30 retries left).
14:31:18-725-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (29 retries left).
14:31:18-725-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (28 retries left).
14:31:18-725-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (27 retries left).
14:31:18-725-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (26 retries left).
14:31:18-725-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (25 retries left).
14:31:18-725-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (24 retries left).
14:31:18-726-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (23 retries left).
14:31:18-726-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (22 retries left).
14:31:18-726-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (21 retries left).
14:31:18-726-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (20 retries left).
14:31:18-726-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (19 retries left).
14:31:18-726-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (18 retries left).
14:31:18-726-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (17 retries left).
14:31:18-726-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (16 retries left).
14:31:18-726-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (15 retries left).
14:31:18-726-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (14 retries left).
14:31:18-726-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (13 retries left).
14:31:18-726-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (12 retries left).
14:31:18-726-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (11 retries left).
14:31:18-727-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (10 retries left).
14:31:18-727-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (9 retries left).
14:31:18-727-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (8 retries left).
14:31:18-727-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (7 retries left).
14:31:18-727-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (6 retries left).
14:31:18-727-AUTO-DEPLOY-INFO: RETRYING: Install offline APT repo GPG key (5 retries left).
14:31:18-727-AUTO-DEPLOY-INFO: changed: [cluster_manager]
14:31:18-727-AUTO-DEPLOY-INFO:
14:31:18-727-AUTO-DEPLOY-INFO: TASK [install-ntp : Create sources.list file]
*****
14:31:18-727-AUTO-DEPLOY-INFO: Monday 03 August 2020 14:10:11 +0000 (0:05:06.712)
0:38:20.950 *****
14:31:18-727-AUTO-DEPLOY-INFO: changed: [cluster_manager]
```

```
14:31:18-727-AUTO-DEPLOY-INFO:
14:31:18-727-AUTO-DEPLOY-INFO: TASK [install-ntp : Disable SRV records so apt-update is
faster.] *****
14:31:18-728-AUTO-DEPLOY-INFO: Monday 03 August 2020 14:10:11 +0000 (0:00:00.100)
0:38:21.050 *****
14:31:18-728-AUTO-DEPLOY-INFO: changed: [cluster_manager]
14:31:18-728-AUTO-DEPLOY-INFO:
14:31:18-728-AUTO-DEPLOY-INFO: TASK [install-ntp : apt_update]
*****
14:31:18-728-AUTO-DEPLOY-INFO: Monday 03 August 2020 14:10:11 +0000 (0:00:00.078)
0:38:21.129 *****
14:31:18-728-AUTO-DEPLOY-INFO: ok: [cluster_manager]
14:31:18-728-AUTO-DEPLOY-INFO:
14:31:18-728-AUTO-DEPLOY-INFO: TASK [install-ntp : Install chrony]
*****
14:31:18-728-AUTO-DEPLOY-INFO: Monday 03 August 2020 14:10:12 +0000 (0:00:01.159)
0:38:22.288 *****
14:31:18-728-AUTO-DEPLOY-INFO: changed: [cluster_manager]
14:31:18-728-AUTO-DEPLOY-INFO:
14:31:18-728-AUTO-DEPLOY-INFO: TASK [install-ntp : Comment out server lines from
/etc/chrony/chrony.conf] *****
14:31:18-728-AUTO-DEPLOY-INFO: Monday 03 August 2020 14:10:12 +0000 (0:00:00.114)
0:38:22.403 *****
14:31:18-728-AUTO-DEPLOY-INFO: changed: [cluster_manager]
14:31:18-729-AUTO-DEPLOY-INFO:
14:31:18-729-AUTO-DEPLOY-INFO: TASK [install-ntp : enable chrony ntp]
*****
14:31:18-729-AUTO-DEPLOY-INFO: Monday 03 August 2020 14:10:12 +0000 (0:00:00.119)
0:38:22.523 *****
14:31:18-729-AUTO-DEPLOY-INFO: ok: [cluster_manager]
14:31:18-729-AUTO-DEPLOY-INFO:
14:31:18-729-AUTO-DEPLOY-INFO: TASK [install-ntp : Add the ntp servers in chrony]
*****
14:31:18-729-AUTO-DEPLOY-INFO: Monday 03 August 2020 14:10:13 +0000 (0:00:00.277)
0:38:22.801 *****
14:31:18-729-AUTO-DEPLOY-INFO: changed: [cluster_manager] => (item={'url':
'8.ntp.esl.cisco.com'})
14:31:18-729-AUTO-DEPLOY-INFO:
```

```
14:31:18-729-AUTO-DEPLOY-INFO: TASK [install-ntp : Remove the apt file]
*****

14:31:18-729-AUTO-DEPLOY-INFO: Monday 03 August 2020 14:10:13 +0000 (0:00:00.111)
0:38:22.912 *****

14:31:18-729-AUTO-DEPLOY-INFO: ok: [cluster_manager]

14:31:18-730-AUTO-DEPLOY-INFO:

14:31:18-730-AUTO-DEPLOY-INFO: RUNNING HANDLER [install-ntp : restart_chrony]
*****

14:31:18-730-AUTO-DEPLOY-INFO: Monday 03 August 2020 14:10:13 +0000 (0:00:00.064)
0:38:22.976 *****

14:31:18-730-AUTO-DEPLOY-INFO: changed: [cluster_manager]

14:31:18-730-AUTO-DEPLOY-INFO:

14:31:18-730-AUTO-DEPLOY-INFO: RUNNING HANDLER [install-ntp : force_time_sync]
*****

14:31:18-730-AUTO-DEPLOY-INFO: Monday 03 August 2020 14:10:13 +0000 (0:00:00.275)
0:38:23.251 *****

14:31:18-730-AUTO-DEPLOY-INFO: RETRYING: force_time_sync (60 retries left).

14:31:18-730-AUTO-DEPLOY-INFO: RETRYING: force_time_sync (59 retries left).

14:31:18-730-AUTO-DEPLOY-INFO: RETRYING: force_time_sync (58 retries left).

14:31:18-730-AUTO-DEPLOY-INFO: RETRYING: force_time_sync (57 retries left).

14:31:18-730-AUTO-DEPLOY-INFO: RETRYING: force_time_sync (56 retries left).

14:31:18-730-AUTO-DEPLOY-INFO: RETRYING: force_time_sync (55 retries left).

14:31:18-731-AUTO-DEPLOY-INFO: RETRYING: force_time_sync (54 retries left).

14:31:18-731-AUTO-DEPLOY-INFO: RETRYING: force_time_sync (53 retries left).

14:31:18-731-AUTO-DEPLOY-INFO: RETRYING: force_time_sync (52 retries left).

14:31:18-731-AUTO-DEPLOY-INFO: changed: [cluster_manager]

14:31:18-731-AUTO-DEPLOY-INFO:

14:31:18-731-AUTO-DEPLOY-INFO: RUNNING HANDLER [install-ntp : verify_chrony_status]
*****

14:31:18-731-AUTO-DEPLOY-INFO: Monday 03 August 2020 14:10:23 +0000 (0:00:09.833)
0:38:33.085 *****

14:31:18-731-AUTO-DEPLOY-INFO: changed: [cluster_manager]

14:31:18-731-AUTO-DEPLOY-INFO:

14:31:18-731-AUTO-DEPLOY-INFO: RUNNING HANDLER [install-ntp : check_system_time]
*****

14:31:18-731-AUTO-DEPLOY-INFO: Monday 03 August 2020 14:10:23 +0000 (0:00:00.196)
0:38:33.282 *****
```

```
14:31:18-731-AUTO-DEPLOY-INFO: changed: [cluster_manager]

14:31:18-731-AUTO-DEPLOY-INFO:

14:31:18-732-AUTO-DEPLOY-INFO: PLAY [Docker load]
*****

14:31:18-732-AUTO-DEPLOY-INFO:

14:31:18-732-AUTO-DEPLOY-INFO: TASK [docker-image-load : Ensure directory]
*****

14:31:18-732-AUTO-DEPLOY-INFO: Monday 03 August 2020 14:10:24 +0000 (0:00:00.726)
0:38:34.008 *****

14:31:18-732-AUTO-DEPLOY-INFO: changed: [cluster_manager]

14:31:18-732-AUTO-DEPLOY-INFO:

14:31:18-732-AUTO-DEPLOY-INFO: TASK [docker-image-load : Copy docker tars]
*****

14:31:18-732-AUTO-DEPLOY-INFO: Monday 03 August 2020 14:10:24 +0000 (0:00:00.085)
0:38:34.094 *****

14:31:18-732-AUTO-DEPLOY-INFO: [WARNING]: Unable to find '/opt/deployer/work/docker-images'
in expected paths

14:31:18-732-AUTO-DEPLOY-INFO: (use -vvvvv to see paths)

14:31:18-732-AUTO-DEPLOY-INFO:

14:31:18-732-AUTO-DEPLOY-INFO: TASK [docker-image-load : Load docker images]
*****

14:31:18-732-AUTO-DEPLOY-INFO: Monday 03 August 2020 14:10:24 +0000 (0:00:00.029)
0:38:34.123 *****

14:31:18-733-AUTO-DEPLOY-INFO: changed: [cluster_manager]

14:31:18-733-AUTO-DEPLOY-INFO:

14:31:18-733-AUTO-DEPLOY-INFO: PLAY [Offline Products load]
*****

14:31:18-733-AUTO-DEPLOY-INFO:

14:31:18-733-AUTO-DEPLOY-INFO: TASK [offline-products-load : Ensure directory]
*****

14:31:18-733-AUTO-DEPLOY-INFO: Monday 03 August 2020 14:10:24 +0000 (0:00:00.093)
0:38:34.217 *****

14:31:18-733-AUTO-DEPLOY-INFO: ok: [cluster_manager]

14:31:18-733-AUTO-DEPLOY-INFO:

14:31:18-733-AUTO-DEPLOY-INFO: TASK [offline-products-load : Copy offline product tars]
*****

14:31:18-733-AUTO-DEPLOY-INFO: Monday 03 August 2020 14:10:24 +0000 (0:00:00.076)
0:38:34.293 *****

14:31:18-733-AUTO-DEPLOY-INFO: changed: [cluster_manager] =>
```

```

(item=/opt/deployer/work/offline-products/cee-2020-01-1-11.tar)

14:31:18-733-AUTO-DEPLOY-INFO: changed: [cluster_manager] =>
(item=/opt/deployer/work/offline-products/inode-manager-3.0.0-release-2007142325.tar)

14:31:18-733-AUTO-DEPLOY-INFO: changed: [cluster_manager] =>
(item=/opt/deployer/work/offline-products/opshub-release-2007150030.tar)

14:31:18-733-AUTO-DEPLOY-INFO:

14:31:18-734-AUTO-DEPLOY-INFO: PLAY RECAP
*****

14:31:18-734-AUTO-DEPLOY-INFO: cluster_manager          : ok=41   changed=24   unreachable=0
failed=0

14:31:18-734-AUTO-DEPLOY-INFO:

14:31:18-734-AUTO-DEPLOY-INFO: Monday 03 August 2020  14:31:18 +0000 (0:20:54.167)
0:59:28.461 *****

14:31:18-734-AUTO-DEPLOY-INFO:
=====

14:31:18-734-AUTO-DEPLOY-INFO: offline-products-load : Copy offline product tars
----- 1254.17s

14:31:18-734-AUTO-DEPLOY-INFO: vm-vsphere : Upload VM Template
----- 931.81s

14:31:18-734-AUTO-DEPLOY-INFO: install-ntp : Check smi ingresses
----- 705.76s

14:31:18-734-AUTO-DEPLOY-INFO: install-ntp : Install offline APT repo GPG key
----- 306.71s

14:31:18-734-AUTO-DEPLOY-INFO: install-ntp : Check ingress url
----- 128.79s

14:31:18-734-AUTO-DEPLOY-INFO: vm-vsphere : Wait for ssh
----- 116.19s

14:31:18-734-AUTO-DEPLOY-INFO: vm-vsphere : Create VM
----- 56.17s

14:31:18-735-AUTO-DEPLOY-INFO: init-k3s : Init k3s
----- 22.97s

14:31:18-735-AUTO-DEPLOY-INFO: vm-vsphere : Get VM Update needed
----- 12.98s

14:31:18-735-AUTO-DEPLOY-INFO: vm-vsphere : Check if VM Template exists
----- 12.54s

14:31:18-735-AUTO-DEPLOY-INFO: install-ntp : force_time_sync
----- 9.83s

14:31:18-735-AUTO-DEPLOY-INFO: vm-vsphere : Test vCenter credentials are valid
----- 2.03s

14:31:18-735-AUTO-DEPLOY-INFO: install-ntp : Remove "ntp" package
----- 1.45s

14:31:18-735-AUTO-DEPLOY-INFO: install-ntp : apt_update

```



```

----- 1.16s
14:31:18-735-AUTO-DEPLOY-INFO: Gathering Facts
----- 1.04s
14:31:18-735-AUTO-DEPLOY-INFO: install-ntp : check_system_time
----- 0.73s
14:31:19-93-AUTO-DEPLOY-INFO: 2020-08-03 14:31:19.092 INFO deploy: Success
14:31:19-94-AUTO-DEPLOY-INFO: 2020-08-03 14:31:19.093 INFO deploy:
14:31:19-94-AUTO-DEPLOY-INFO:
14:31:19-95-AUTO-DEPLOY-INFO: Environment Information:
14:31:19-95-AUTO-DEPLOY-INFO: =====
14:31:19-95-AUTO-DEPLOY-INFO: SSH: ssh cloud-user@10.90.154.28
14:31:19-96-AUTO-DEPLOY-INFO: Deployer CLI: cli.smi-cluster-deployer.10.90.154.28.nip.io
14:31:19-96-AUTO-DEPLOY-INFO: Deployer User/Pass: admin/CiscoChn123*
14:31:19-96-AUTO-DEPLOY-INFO: -----
14:31:19-96-AUTO-DEPLOY-INFO:
14:31:19-97-AUTO-DEPLOY-INFO: init-k3s : Ensure /data folder exists
----- 0.63s
14:31:19-97-AUTO-DEPLOY-INFO: vm-vsphere : Create user data ISO
----- 0.34s
14:31:19-97-AUTO-DEPLOY-INFO: install-ntp : Cleaning cache
----- 0.31s
14:31:19-97-AUTO-DEPLOY-INFO: install-ntp : enable chrony ntp
----- 0.28s

14:31:20-357-paramiko.transport-INFO: Connected (version 2.0, client OpenSSH_7.6p1)
14:31:20-575-paramiko.transport-INFO: Authentication (publickey) successful!
14:31:20-576-AUTO-DEPLOY-INFO:
Host connected successfully
14:31:20-576-AUTO-DEPLOY-INFO:
Checking deployer VM status
14:31:20-576-AUTO-DEPLOY-INFO:
Checking for software package...
14:31:22-809-AUTO-DEPLOY-INFO: [34m
Software package list: {
  "Packages": {
    "package": ["cee-2020-01-1-11"],
    "package": ["inode-manager-3.0.0-release-2007142325"],
    "package": ["opshub-release-2007150030"],
    "package": ["sample"]
  }
}
[0m
14:31:22-810-AUTO-DEPLOY-INFO:
All software packages are loaded to smi deployer successfully
14:31:22-810-AUTO-DEPLOY-INFO:
Checking smi-cluster-deployer pod status...
14:31:30-606-AUTO-DEPLOY-INFO:
Waiting for Deployer VM Services...

```

```

14:31:30-607-AUTO-DEPLOY-INFO:
Waiting for Deployer VM Services...
14:31:30-607-AUTO-DEPLOY-INFO:
Waiting for Deployer VM Services...
14:31:30-607-AUTO-DEPLOY-INFO:
Waiting for Deployer VM Services...
14:31:30-607-AUTO-DEPLOY-INFO:
Waiting for Deployer VM Services...
14:31:30-607-AUTO-DEPLOY-INFO:
Waiting for Deployer VM Services...
14:31:30-608-AUTO-DEPLOY-INFO: [32m
Deployer VM is ready[0m
14:31:30-608-AUTO-DEPLOY-INFO: [32m

Completed cnBR/Opshub automatic offline installation

```

Log Example for iNode Manager AIO Cluster Installation

```

08:47:40-295-root-INFO: Start logging for cnBR/Opshub automatic offline installation:
08:48:14-527-AUTO-DEPLOY-INFO: [32m
--- : Product Info : ---[0m
08:48:14-527-AUTO-DEPLOY-INFO: [34mcee :
http://charts.10.90.154.28.nip.io/cee-2020-01-1-11[0m
08:48:14-527-AUTO-DEPLOY-INFO: [34mopshub :
http://charts.10.90.154.28.nip.io/opshub-release-2007150030[0m
08:48:14-527-AUTO-DEPLOY-INFO: [34minodemanager :
http://charts.10.90.154.28.nip.io/inodemanager-3.0.0-release-2007142325[0m
08:48:14-527-AUTO-DEPLOY-INFO: [32m
--- : cnBR Images : ---[0m
08:48:14-527-AUTO-DEPLOY-INFO: [34mcluster-manager-docker-deployer :
cluster-manager-docker-deployer:1.0.3-0079-01a50dd[0m
08:48:14-527-AUTO-DEPLOY-INFO: [34mautodeploy :
autodeploy:0.1.0-0408-f8blfe6[0m
08:48:14-528-AUTO-DEPLOY-INFO: [32m
--- : vCenter Info : ---[0m
08:48:14-528-AUTO-DEPLOY-INFO: [34matl-smi-inodemgr-lab : Cloud Video Datacenter,
iNodeManager[0m
08:48:14-528-AUTO-DEPLOY-INFO: [32m
--- : Deployer Info : ---[0m
08:48:14-528-AUTO-DEPLOY-INFO: [34minode-manager-deployer-1 : IP -> 10.90.154.28/24,
host -> 10.90.154.7[0m
08:48:14-528-AUTO-DEPLOY-INFO:
08:48:15-169-AUTO-DEPLOY-INFO:
Reuse an existing deployer with IP 10.90.154.28, running pre-check
08:48:15-175-paramiko.transport-INFO: Connected (version 2.0, client OpenSSH_7.6p1)
08:48:15-237-paramiko.transport-INFO: Authentication (publickey) successful!
08:48:15-237-AUTO-DEPLOY-INFO:
Host connected successfully
08:48:15-238-AUTO-DEPLOY-INFO:
Checking deployer VM status
08:48:15-238-AUTO-DEPLOY-INFO:
Checking for software package...
08:48:15-935-AUTO-DEPLOY-INFO: [34m
Software package list: {
  "Packages": {
    "package": ["opshub-release-2007150030"],
    "package": ["inodemanager-3.0.0-release-2007142325"],
    "package": ["cee-2020-01-1-11"],
    "package": ["sample"]
  }
}
[0m

```

```

08:48:15-936-AUTO-DEPLOY-INFO:
All software packages are loaded to smi deployer successfully
08:48:15-936-AUTO-DEPLOY-INFO:
Checking smi-cluster-deployer pod status...
08:48:19-189-AUTO-DEPLOY-INFO:
Waiting for Deployer VM Services...
08:48:19-189-AUTO-DEPLOY-INFO:
Waiting for Deployer VM Services...
08:48:19-190-AUTO-DEPLOY-INFO:
Waiting for Deployer VM Services...
08:48:19-190-AUTO-DEPLOY-INFO:
Waiting for Deployer VM Services...
08:48:19-190-AUTO-DEPLOY-INFO:
Waiting for Deployer VM Services...
08:48:19-190-AUTO-DEPLOY-INFO:
Waiting for Deployer VM Services...
08:48:19-190-AUTO-DEPLOY-INFO: [32m
Deployer VM is ready[0m
08:48:19-191-AUTO-DEPLOY-INFO:
Skipping creation of deployer...
08:48:19-191-AUTO-DEPLOY-INFO: [32m
--- : Cluster Info : ---[0m
08:48:19-191-AUTO-DEPLOY-INFO: [34minode-manager-aioc : Type -> inode-manager,
master-IP -> 10.90.154.29[0m
08:48:20-319-AUTO-DEPLOY-INFO: [32mSuccess: Configuring Resource
https://restconf.smi-cluster-deployer.10.90.154.28.nip.io/api/config/environments/[0m
08:48:21-184-AUTO-DEPLOY-INFO: [32mSuccess: Configuring Resource
https://restconf.smi-cluster-deployer.10.90.154.28.nip.io/api/config/feature-gates[0m
08:48:22-419-AUTO-DEPLOY-INFO: [32mSuccess: Configuring Resource
https://restconf.smi-cluster-deployer.10.90.154.28.nip.io/api/config/clusters/[0m
08:48:22-419-AUTO-DEPLOY-INFO: [32m

Completed cnBR/Opshub automatic offline installation

```

Log Example for iNode Manager Cluster Multi-Node Installation

```

09:10:26-111-root-INFO: Start logging for cnBR/Opshub automatic offline installation:
09:10:39-345-AUTO-DEPLOY-INFO: [32m
--- : Product Info : ---[0m
09:10:39-345-AUTO-DEPLOY-INFO: [34mcee :
http://charts.10.90.154.28.nip.io/cee-2020-01-1-11[0m
09:10:39-345-AUTO-DEPLOY-INFO: [34mopshub :
http://charts.10.90.154.28.nip.io/opshub-release-2007150030[0m
09:10:39-346-AUTO-DEPLOY-INFO: [34minodemanager :
http://charts.10.90.154.28.nip.io/inodemanager-3.0.0-release-2007142325[0m
09:10:39-346-AUTO-DEPLOY-INFO: [32m
--- : cnBR Images : ---[0m
09:10:39-346-AUTO-DEPLOY-INFO: [34mcluster-manager-docker-deployer :
cluster-manager-docker-deployer:1.0.3-0079-01a50dd[0m
09:10:39-346-AUTO-DEPLOY-INFO: [34mautodeploy :
autodeploy:0.1.0-0409-b5ec500[0m
09:10:39-346-AUTO-DEPLOY-INFO: [32m
--- : vCenter Info : ---[0m
09:10:39-346-AUTO-DEPLOY-INFO: [34matl-smi-inodemgr-lab : Cloud Video Datacenter,
iNodeManager[0m
09:10:39-346-AUTO-DEPLOY-INFO: [32m
--- : Deployer Info : ---[0m
09:10:39-346-AUTO-DEPLOY-INFO: [34minode-manager-deployer-1 : IP -> 10.90.154.28/24,
host -> 10.90.154.7[0m
09:10:39-347-AUTO-DEPLOY-INFO:
09:10:39-358-AUTO-DEPLOY-INFO:
Reuse an existing deployer with IP 10.90.154.28, running pre-check

```

```

09:10:39-364-paramiko.transport-INFO: Connected (version 2.0, client OpenSSH_7.6p1)
09:10:39-425-paramiko.transport-INFO: Authentication (publickey) successful!
09:10:39-425-AUTO-DEPLOY-INFO:
Host connected successfully
09:10:39-426-AUTO-DEPLOY-INFO:
Checking deployer VM status
09:10:39-426-AUTO-DEPLOY-INFO:
Checking for software package...
09:10:41-39-AUTO-DEPLOY-INFO: [34m
Software package list: {
  "Packages": {
    "package": ["opshub-release-2007150030"],
    "package": ["inodemanager-3.0.0-release-2007142325"],
    "package": ["cee-2020-01-1-11"],
    "package": ["sample"]
  }
}
[0m
09:10:41-39-AUTO-DEPLOY-INFO:
All software packages are loaded to smi deployer successfully
09:10:41-39-AUTO-DEPLOY-INFO:
Checking smi-cluster-deployer pod status...
09:10:45-344-AUTO-DEPLOY-INFO:
Waiting for Deployer VM Services...
09:10:45-345-AUTO-DEPLOY-INFO:
Waiting for Deployer VM Services...
09:10:45-346-AUTO-DEPLOY-INFO:
Waiting for Deployer VM Services...
09:10:45-347-AUTO-DEPLOY-INFO:
Waiting for Deployer VM Services...
09:10:45-347-AUTO-DEPLOY-INFO:
Waiting for Deployer VM Services...
09:10:45-348-AUTO-DEPLOY-INFO:
Waiting for Deployer VM Services...
09:10:45-348-AUTO-DEPLOY-INFO: [32m
Deployer VM is ready[0m
09:10:45-348-AUTO-DEPLOY-INFO:
Skipping creation of deployer...
09:10:45-348-AUTO-DEPLOY-INFO: [32m
--- : Cluster Info : ---[0m
09:10:45-349-AUTO-DEPLOY-INFO: [34minode-manager-multinode      : Type -> inode-manager,
master-IP -> 10.90.154.30[0m
09:10:46-413-AUTO-DEPLOY-INFO: [32mSuccess: Configuring Resource
https://restconf.smi-cluster-deployer.10.90.154.28.nip.io/api/config/environments/[0m
09:10:47-76-AUTO-DEPLOY-INFO: [32mSuccess: Configuring Resource
https://restconf.smi-cluster-deployer.10.90.154.28.nip.io/api/config/feature-gates[0m
09:10:48-450-AUTO-DEPLOY-INFO: [32mSuccess: Configuring Resource
https://restconf.smi-cluster-deployer.10.90.154.28.nip.io/api/config/clusters/[0m
09:10:48-451-AUTO-DEPLOY-INFO: [32m

Completed cnBR/Opshub automatic offline installation

```