Cisco iNode Manager Installation Guide, Release 3.2.0

First Published: 2021-06-30

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

CHAPTER 1

Prepare UCS Servers

This chapter describes how to configure and prepare UCS servers for iNode Manager software installation.

- Configure the servers using the Cisco Integrated Management Controller (CIMC).
- Install the ESXi Hypervisor.
- Add the ESXi hosts to a vSphere cluster using VMware vCenter.

Install and Configure iNode Manager Cluster

To install and configure the Cisco iNode Manager cluster, do the following:

1. UCS Server Installation, on page 1
2. Update Firmware, on page 2
3. Configure Boot Drives, on page 2
4. Configure Data Drives, on page 2
5. Install the VMware ESXi Hypervisor, on page 2
6. Reboot ESXi Host and Set Boot Device, on page 3

UCS Server Installation

Step 1 Rack mount the servers.

For more details, refer the Cisco UCS C220 M5 Server Installation and Service Guide.

Step 2 Ensure that both power supplies are connected on each server and power on the servers.

Step 3 Connect the network cables.

- For CIMC, use the 1-Gb Ethernet dedicated management port.
- For ESXi Host Management, use the Ethernet Port 1 of the Dual 1Gb/10Gb Intel X550T onboard NIC.

- For iNode Manager data, connect port 1 of the Intel XL710 40G NIC in PCIe Slot 1 to the SP Router/Leaf Switch using Cisco QSFP-40G-SR4.

Step 4 Connect the UCS KVM console adapter or connect a keyboard and a monitor directly to the server.

Step 5 Configure the Cisco IMC through the KVM console and update the Network Settings

Update Firmware

Download the latest Hardware Update Utility for the UCS C220 M5 server from the Cisco Software Download site. The Utility helps you to update the CIMC, BIOS, and device firmware for storage controllers, network adapters, SSDs, and other components.

Configure Boot Drives

Step 1 Enable the Cisco MSTOR Boot Optimized M.2 RAID Controller.

Step 2 Create RAID 1 virtual drive from 2 x M.2 SSD drives.

Step 3 Set Stripe Size to 64 KB.

Configure Data Drives

Step 1 Enable Cisco 12G SAS Modular RAID Controller.

Step 2 Create RAID 5 enabled virtual drive using 4 x SSDs.

Step 3 Set Stripe Size to 64 KB.

Step 4 Set the Write Cache Policy to **Write Back with Good BBU**.

Install the VMware ESXi Hypervisor

Step 1 Install VMware ESXi 6.5 Update 3 on the M.2 RAID 1 Virtual Drive (Boot Drive).

Use the Cisco Custom ISO: `VMware_ESXi_6.5.0_13932383_Custom_Cisco_6.5.3.1.iso`

Step 2 Set a password for the root user per the installation process.

Step 3 Reboot the VMware ESXi host according to the installation process.

Reboot ESXi Host and Set Boot Device

Step 1 Interrupt the boot process with F2 after the host resets and boot into the BIOS.

Step 2 Under the **Boot Options** tab set the **Boot Option #1** to the UEFI target: `VMWARE ESXi`

Step 3 Disable all other boot options.

Step 4 Save changes and exit.

Step 5 Confirm whether the host boots directly into VMware ESXi.

Add ESXi Hosts to vSphere Virtual Infrastructure

Step 1 Configure ESXi host management networking.

a) Log in to the ESXi host through the DCUI with the root account.

b) Configure the Management Network: Update IP configuration, DNS configuration, custom DNS suffixes, and VLAN ID (if necessary).

Step 2 Add ESXi hosts to the VMware vCenter server.

a) In VMware vCenter, create a new, dedicated cluster for iNode Manager.

Do not enable DRS or any HA features.

b) Add each new iNode Manager ESXi host to the new iNode Manager cluster.

Step 3 Configure and enable required ESXi host features.

a) Configure time on the host: Enable NTP.

b) Apply ESXi host licenses.

c) Create a new data store on the data drive storage device.

Step 4 Configure VM networking.

a) Ensure VMware vSwitch connectivity to the physical switch.

b) Create a PortGroup and vSwitch for the K8s cluster node VM management network.

Note For Multinode deployment, ensure that the network name setup is similar across UCS hosts for ease of use and configuration.

CHAPTER 2

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 without Autodeployer

Prerequisite: Install Intelligent Node Software on GS7000 iNode

As a prerequisite for iNode Manager 3.x installation or migration, ensure that the GS7000 iNode runs 2.0.8 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 *02.00.08* 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 6.5.0)*
- *vCenter* (version 6.7.0 or above): manager for the vSphere infrastructure that hosts the VMs for the SMI clusters.

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.

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.

The multinode cluster provides full high-availability support and is the only recommended cluster size for production.

The following is the resource allocation for various node (VM) types:

Node type	CPUs	RAM size (GB)	Disk size - root (GB)	Disk size - home (GB)	Disk size - data (GB)
Deployer	4	16	5	64	40
AIO	16	64	100	10	200
Control-Plane	2	16	20	5	20
etcd	2	16	20	5	20
infra	14	64	100	10	200
ops	4	16	100	10	100

High-Level Overview of Installation Workflow

1. Prepare Staging Server, on page 6
2. Prepare SMI Cluster Deployer Configuration File, on page 8
3. Install the SMI Cluster Deployer, on page 9
4. Prepare iNode Manager Cluster Configuration, on page 11
5. Deploy iNode Manager Cluster, on page 24
6. Monitor iNode Manager Cluster Deployment, on page 24

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-config.yaml
│   ├── aio-inode-manager-standby-config.yaml
│   ├── deployer-sample-config.yaml
│   ├── 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 SMI Cluster Deployer 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 IP: Host on which you install the nodes.
- Data store names

In addition, provide an available IP address for the Deployer.

SMI Cluster Deployer Configuration File

The SMI Cluster Deployer configuration file is a yaml file that provides vCenter access and network access details. The installation script uses the vCenter access information to: communicate with the VMware vCenter. The script instructs vCenter to create the Virtual Machines (VMs) for the SMI Cluster Manager and configure the VM using the network access information in the configuration file.

You can find a sample of the deployer configuration file (deployer-sample-config.yaml) in the examples subdirectory. Use this file to create the SMI cluster deployer configuration file:

1. Copy the sample configuration file.
2. Remove the comments starting with # to end of each line.
3. Replace the fields marked by the <...> placeholders with the correct information.

Note

- The file is in yaml format. Follow the conventions of yaml specification (https://yaml.org/spec/). In general, you do not need double quotes (") around a string. However, if a field-value is a string with space, do enclose it within double quotes.
- The deployer configuration file contains an SSH public and private key pair. You can use existing SSH key pair, or generate new one. The following link has instructions on how to generate an SSH key pair: https://www.ssh.com/ssh/keygen
- If your vcenter has multiple hosts and data stores, make sure that the specified data store is available to the specified host.
- Make sure the deployer IP address is not in use. Shut down or remove any VM that is using the deployer IP first if needed, before installing the deployer.

Install the SMI Cluster Deployer

Step 1 Load the cluster-manager-docker-deployer_.tar file using the following command:

```
docker load -i utility-images/cluster-manager-docker-deployer_<ver>.tar
```

Step 2 After the deployer configuration file is ready, install the deployer using the following command. Replace *deployer_config.yaml* with the deployer configuration file.

```
docker run --rm -v ~/SMI/staging:/opt/deployer/work -it
dockerhub.cisco.com/smi-fuse-docker-internal/smi-apps/cluster-manager-docker-deployer/master/cluster-manager-docker-deployer:1.0.3-0079-01a50dd
 ./deploy  --vmdk-name cluster-deployer-airgap.vmdk --config-file-name deployer_config.yaml
```

Note

- The configuration file must reside in the installation directory.
- Depending on your settings, you may have to prepend 'sudo' to the preceding command.

If the deployer is not currently running, this command sets up the new deployer. If your deployer is already running, this command adds the new offline software packages to the deployer.

Note If you want to remove the old Deployer and install it afresh, use this command instead:

```
docker run --rm -v ~/SMI/staging:/opt/deployer/work -it
dockerhub.cisco.com/smi-fuse-docker-internal/smi-apps/cluster-manager-docker-deployer/master/cluste
  ./deploy  --vmdk-name cluster-deployer-airgap.vmdk --config-file-name deployer_config.yaml --over
```

Wait for the installation process to complete.

Example

The following is a sample log of the installation process.

```
PLAY [Offline Products load]
*******************************************************************************************************************************************************************

TASK [offline-products-load : Ensure directory]
```

```
****************************************************************************************************************************************************************
Wednesday 13 May 2020  05:11:21 +0000 (0:00:07.395)       0:12:45.411 *********
ok: [cluster_manager]

TASK [offline-products-load : Copy offline product tars]
****************************************************************************************************************************************************************
Wednesday 13 May 2020  05:11:21 +0000 (0:00:00.086)       0:12:45.498 *********

PLAY RECAP
****************************************************************************************************************************************************************
cluster_manager            : ok=40   changed=23   unreachable=0    failed=0

Wednesday 13 May 2020  05:11:21 +0000 (0:00:00.023)       0:12:45.521 *********
===============================================================================
vm-vsphere : Upload VM Template
------------------------------------------------------------------------------------------
 264.53s
install-ntp : Check smi ingresses
------------------------------------------------------------------------------------------
 219.96s
docker-image-load : Copy docker tars
------------------------------------------------------------------------------------------
 80.73s
vm-vsphere : Wait for ssh
------------------------------------------------------------------------------------------
 66.27s
install-ntp : Check ingress url
------------------------------------------------------------------------------------------
 58.27s
install-ntp : Install offline APT repo GPG key
------------------------------------------------------------------------------------------
 27.25s
install-ntp : force_time_sync
------------------------------------------------------------------------------------------
 8.55s
vm-vsphere : Create VM
------------------------------------------------------------------------------------------
 7.88s
docker-image-load : Load docker images
------------------------------------------------------------------------------------------
 7.40s
init-k3s : Init k3s
------------------------------------------------------------------------------------------
 7.12s
install-ntp : restart_chrony
------------------------------------------------------------------------------------------
 5.58s
install-ntp : apt_update
------------------------------------------------------------------------------------------
 2.05s
install-ntp : Cleaning cache
------------------------------------------------------------------------------------------
 1.30s
Gathering Facts
------------------------------------------------------------------------------------------
 1.23s
install-ntp : Remove "ntp" package
------------------------------------------------------------------------------------------
 1.01s
vm-vsphere : Get VM Update needed
------------------------------------------------------------------------------------------
 0.72s
vm-vsphere : Check if VM Template exists
------------------------------------------------------------------------------------------
```

```
 0.66s
vm-vsphere : Test vCenter credentials are valid
-------------------------------------------------------------------------------------------
 0.54s
vm-vsphere : Create user data ISO
-------------------------------------------------------------------------------------------
 0.49s
install-ntp : Install chrony
-------------------------------------------------------------------------------------------
 0.46s
2020-05-13 05:11:22.101 INFO deploy: Success
2020-05-13 05:11:22.101 INFO deploy:

Environment Information:
========================
SSH: ssh cloud-user@192.0.2.202
Deployer CLI: cli.smi-cluster-deployer.192.0.2.202.nip.io
Deployer User/Pass: admin/xxxxxx
------------------------
```

Prepare iNode Manager Cluster Configuration

Accessing the SMI Cluster Deployer

Perform iNode Manager cluster deployment on the SMI Cluster Deployer. Access the deployer using a web browser or a terminal.

Accessing the Deployer Using a Web Browser:

To access the Deployer from a web browser, point the browser to the following url:

```
https://cli.smi-cluster-deployer.<deployer-ip>.nip.io/
```

Log in with the username *admin* and the password in the init-k3s section in your deployer configuration file.

Accessing the Deployer Using a Terminal Program:

Log in to the Deployer VM using SSH:

```
ssh -i <ssh-private-key-file> <deployer-user>@<depolyer-ip>
```

Note The private key file must have the ssh-private-key in your deployer configuration file.

Next, use **kubectl** command to find the internal IP address of the Operations-Center service:

```
kubectl get svc ops-center-smi-cluster-deployer -n smi
```

Obtain the CLUSTER-IP field from the output of the **kubectl** command. Use it to SSH into the deployer:

```
ssh admin@<cluster-ip> -p 2024
```

The first time you log in to the deployer, use the initial password in init-k3 section of the deployer configuration file. The deployer prompts you to change the password.

The following example is a sample log of this process:

```
$ ssh -i id_rsa cloud-user@192.0.2.234

Last login: Mon May 11 02:55:02 2020 from 192.0.2.156
cloud-user@smi-deployer-inodemgr:~$
```

```
cloud-user@smi-deployer-inodemgr:~$ kubectl get svc ops-center-smi-cluster-deployer -n smi
NAME                              TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)
                    AGE
ops-center-smi-cluster-deployer   ClusterIP   192.0.2.35   <none>
8008/TCP,2024/TCP,2022/TCP,7681/TCP   12d
cloud-user@smi-deployer-inodemgr:~$ ssh admin@192.0.2.35 -p 2024
admin@10.43.120.106's password:
Welcome to the Cisco SMI Cluster Deployer on smi-deployer-inodemgr
admin connected from 192.0.2.1 using ssh on ops-center-smi-cluster-deployer-7b979b5548-6pl85
[smi-deployer-inodemgr] SMI Cluster Deployer#
```

Note Accessing the deployer (using either a web browser or a terminal program) grants you access to the Operations center of the SMI deployer. The Operations center provides a Cisco IOS like environment. For example, you can use "show run" to show the running configuration.

Adding Configuration for iNode Manager Cluster

Enter the configuration mode using the **config** command:

```
[smi-deployer-inodemgr] SMI Cluster Deployer# config
Entering configuration mode terminal
[smi-deployer-inodemgr] SMI Cluster Deployer(config)#
```

Note You must save changes to the configuration using the **commit** command. Alternately, come out of config mode with **exit** and input ‘yes’ at the prompt.

To deploy AIO or Single Node cluster, you must change the default VMware sizing parameters. To change these parameters, add the following configuration:

```
feature-gates alpha true
feature-gates test true
```

Next, add the VMware vCenter environment to the configuration. There is a sample vCenter environment configuration file (sample-vcenter-env.cfg) in the examples subdirectory.

Note You can have multiple VMware vCenter environments in the configuration.

Next, add the iNode Manager cluster configuration. There are sample configuration files for the AIO (sample-aio.cfg) and multinode (sample-multinode.cfg) clusters in the examples subdirectory.

Note

- The cluster name must contain only lowercase letters, digits, and underscore (_).
- You can have multiple cluster configurations in a deployer.
- Changes to the configuration of an SMI cluster will not take effect until you apply it to the cluster using the **sync** command.

Special Handling for Secret Fields

To protect user data, the deployer displays the sensitive fields like passwords and SSH private key (not SSH public key) of the **show run** command output (running configuration) in AES encrypted format.

To configure these fields, enter the commands up to the secret field, hit Enter, and then type or paste the secret value. For example:

```
[smi-deployer-inodemgr] SMI Cluster Deployer(config-environments-test-vcenter)# vcenter
password
(<AES encrypted string>): **********
[smi-deployer-inodemgr] SMI Cluster Deployer(config-environments-test-vcenter)#
```

Note

- The encryption includes a secret key which is different for each deployer installation. So never copy the encrypted field from one deployer and paste it over to another deployer.
- For the SSH private keys that span multiple lines, paste the entire private key (including the BEGIN and END marker lines), and hit Control+D to exit the AES encrypted string mode.

Netplan Configuration

Netplan is the network configuration utility in Ubuntu Linux distribution from 18.04 release. It uses a YAML file to describe the networking-related configuration like interfaces, routes, DNS servers, and so on. On an Ubuntu host, the netplan configuration file is in the directory /etc/netplan.

In the SMI cluster configuration, you must provide a template for the netplan configuration in the VMs. The deployer applies this single template to all the VMs for the cluster. This template is in jinja2 templating language, with variables defined that may be different from VM to VM. The Jinja2 templating engine renders the template to provide the netplan configuration for each VM, using variables defined for each VM.

You can enter the netplan template in CLI as follows:

```
node-defaults netplan template
< enter all the lines of your netplan template here >
< enter control-D to end >
```

Note The netplan template is a long string, with '\n' replacing line breaks.

The following is a simple netplan template example:

```
network:
  ethernets:
    ens192:
      addresses: [ {{addr}} ]
      gateway4: {{gateway}}
      nameservers:
        addresses: [ {{nameserveraddr}} ]
```

In this template, three variables are present: *addr*, *gateway*, and *nameserveraddr*. Define these variables accordingly:

```
default-node netplan variable-definition addr
exit
default-node netplan variable-definition gateway
exit
```

```
default-node netplan variable-definition nameserver
exit
```

In the nodes section, define values for these variables: The following is an example for the configuration of a node in AIO:

```
nodes ops
  ...
  netplan variable addr
    value 192.0.2.207/25
  exit
  netplan variable gateway
    value 192.0.2.129
  exit
  netplan variable nameserver
    value 192.0.2.3,192.0.2.10
  exit
exit
```

Note If the value for a variable contains spaces, place the value inside double quotes (").

The following example is the rendered netplan configuration:

```
network:
    ethernets:
        ens192:
            addresses:
            - 192.0.2.207/25
            gateway4: 192.0.2.129
            nameservers:
                addresses:
                - 192.0.2.83
                - 192.0.2.10
```

Find the recommended netplan template for iNode Manager in the examples subdirectory:

```
network:
    version: 2
    ethernets:
        ens192:
            addresses: [ {{K8S_SSH_IP}}/{{prefixlen}} ]
            {%- if gateway4 %}
            gateway4: {{ gateway4 }}
            {%- endif %}
            {%- if gateway6 %}
            gateway6: {{ gateway6 }}
            {%- endif %}
            nameservers:
                {%- if nameserveraddr %}
                addresses: [ {{nameserveraddr}} ]
                {%- endif %}
                {%- if nameserversearch %}
                search: [ {{nameserversearch}} ]
                {%- endif %}
            {%- if route1 %}
            routes: [ {{route1}} ]
            {%- endif %}
        {%- if addr2 %}
        ens224:
            addresses: [ {{addr2}} ]
            {%- if route2 %}
            routes: [ {{route2}} ]
            {%- endif %}
        {%- endif %}
        {%- if addr3 %}
        ens256:
            addresses: [ {{addr3}} ]
            {%- if route3 %}
            routes: [ {{route3}} ]
            {%- endif %}
        {%- endif %}
```

This template supports up to two CIN networks, with optional routing tables in the primary and the CIN networks.

The template uses the following variable definitions:

- *K8S_SSH_IP:* SSH IP address. The installer sets this IP address automatically. You do not need to define it explicitly.
- *preflxlen:* The length of the netmask for the primary IP address
- *addr2:* List of IP addresses with netmask length for the first CIN network. If you do not set this address, the installer does not use the first CIN interface.
- *addr3:* List of IP addresses with netmask length for the second CIN network. If you do not set this address, the installer does not use the second CIN interface.
- *gateway4:* IP4 gateway
- *gateway6:* IPv6 gateway
- *nameserveraddr:* List of name server addresses. The installer uses these addresses in the primary network.
- *nameserversearch:* List of name server search domains. The installer uses these search domains in the primary network.
- *route1:* Routing table of the primary network
- *route2:* Routing table of the first CIN network
- *route3:* Routing table of the second CIN network

The following example shows how to set the netplan variables for an AIO cluster:

```
netplan variables gateway4
 value 203.0.113.129
exit
netplan variables nameserveraddr
 value 209.165.200.225,203.0.113.1
exit
netplan variables nameserversearch
 value cisco.com
exit
netplan variables addr2
 value 10.40.5.167/24,2001:DB8:10:40:5::167/48
exit
netplan variables route2
 value "{ to: 192.168.0.0/16, via: 10.40.5.1 },  { to: 172.16.0.0/12, via: 10.40.5.1 }"
exit
```

The following example shows a rendered netplan configuration:

```
network:
    ethernets:
        ens192:
            addresses:
            - 203.0.113.207/25
            gateway4: 203.0.113.129
            nameservers:
                addresses:
                - 209.165.200.225
                - 203.0.113.1
                search:
                - cisco.com
        ens224:
            addresses:
            - 10.40.5.167/24
            - 2001:DB8:10:40:5::167/48
            routes:
            -   to: 192.168.0.0/16
                via: 10.40.5.1
            -   to: 172.16.0.0/12
                via: 10.40.5.1
    version: 2
```

Note The sample cluster configuration files use the recommended netplan template.

Operations Center Configuration

The iNode Manager cluster requires services from the following three products:

- **CEE** provides monitoring and logging.
- **Operations Hub** provides infrastructure services.
- **iNode Manager** provides application-specific services for iNode Manager.

You can configure and manage each product from their corresponding Operations Centers. The Operations Center configuration has the following format:

FQDN disabled:

```
ops-centers <product> <instance-name>
 repository <chart-url>
 initial-boot-parameters first-boot-password <encrypted-password>
 initial-boot-parameters auto-deploy { true | false }
 initial-boot-parameters single-node { true | false }
exit
```

FQDN enabled:

```
ops-centers <product> <instance-name>
 repository <chart-url>
 ingress-hostname <FQDN>
 initial-boot-parameters first-boot-password <encrypted-password>
 initial-boot-parameters auto-deploy { true | false }
 initial-boot-parameters single-node { true | false }
exit
```

In this configuration, *<product>* can be either *cee*, *opshub*, or *inode-manager*.

The <chart-url> must be in the following form: *http://charts.<deployer-ip>.nip.io/<product-chart>*

<FQDN> is the fully qualified domain name that is registered with the DNS server.

To obtain the exact name of the product charts in the installation file, run the following command in the SMI Deployer:

```
software-package list
```

If you set *auto-deploy* to *true*, Operations Center installation and automatic deployment take place.

Set *single-node* to *true* for AIO clusters, and false for multinode clusters.

The following example shows an Operations Center configuration for an AIO cluster:

FQDN disabled:

```
[deployer] SMI Cluster Deployer# software-packages list
[ cee-<version> ]
[ inode-manager-<version> ]
[ sample ]
[ opshub-<version> ]
[deployer] SMI Cluster Deployer# show run cluster test-aio ops-centers
clusters test-aio
 ops-centers cee data
  repository https://charts.209.165.200.225.nip.io/cee-2020-1-2
  initial-boot-parameters use-volume-claims true
  initial-boot-parameters first-boot-password $8$x3HvIjGBnzHv27exE5vC3ozNmL+5h4ulTEsG3D8+WJ0=

  initial-boot-parameters auto-deploy true
  initial-boot-parameters single-node true
 exit
 ops-centers opshub data
  repository https://charts.209.165.200.225.nip.io/opshub-<version>
  initial-boot-parameters use-volume-claims true
  initial-boot-parameters first-boot-password $8$x3HvIjGBnzHv27exE5vC3ozNmL+5h4ulTEsG3D8+WJ0=

  initial-boot-parameters auto-deploy true
  initial-boot-parameters single-node true
 exit
 ops-centers inode-manager data
  repository http://charts.209.165.200.225.nip.io/inode-manager-<version>
  initial-boot-parameters use-volume-claims true
  initial-boot-parameters first-boot-password $8$x3HvIjGBnzHv27exE5vC3ozNmL+5h4ulTEsG3D8+WJ0=

  initial-boot-parameters auto-deploy true
  initial-boot-parameters single-node true
 exit
exit
```

FQDN enabled:

```
[deployer] SMI Cluster Deployer# software-packages list
[ cee-<version> ]
[ inode-manager-<version> ]
[ sample ]
[ opshub-<version> ]
[deployer] SMI Cluster Deployer# show run cluster test-aio ops-centers
clusters test-aio
 ops-centers cee data
  repository https://charts.209.165.200.225.nip.io/cee-2020-1-2
  ingress-hostname inodemgr.cisco.com
  initial-boot-parameters use-volume-claims true
  initial-boot-parameters first-boot-password $8$x3HvIjGBnzHv27exE5vC3ozNmL+5h4ulTEsG3D8+WJ0=

  initial-boot-parameters auto-deploy true
  initial-boot-parameters single-node true
 exit
 ops-centers opshub data
  repository https://charts.209.165.200.225.nip.io/opshub-<version>
  ingress-hostname inodemgr.cisco.com
  initial-boot-parameters use-volume-claims true
  initial-boot-parameters first-boot-password $8$x3HvIjGBnzHv27exE5vC3ozNmL+5h4ulTEsG3D8+WJ0=

  initial-boot-parameters auto-deploy true
  initial-boot-parameters single-node true
 exit
 ops-centers inode-manager data
  repository http://charts.209.165.200.225.nip.io/inode-manager-<version>
  ingress-hostname inodemgr.cisco.com
  initial-boot-parameters use-volume-claims true
  initial-boot-parameters first-boot-password $8$x3HvIjGBnzHv27exE5vC3ozNmL+5h4ulTEsG3D8+WJ0=

  initial-boot-parameters auto-deploy true
  initial-boot-parameters single-node true
 exit
exit
```

Sample Configuration Files

The `examples` directory contains sample configuration files for the deployment without Autodeployer:

- `aio-inode-manager-cli-config`: CLIs for the single-node iNode Manager cluster configuration.
- `multinode-inode-manager-cli-config`: CLIs for the multinode iNode Manager cluster configuration.

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

aio-inode-manager-cli-config

```
clusters inode-manager-aio
 environment atl-smi-inodemgr-lab
 addons ingress bind-ip-address 10.90.154.29
 configuration size      functional-test-aio
 configuration allow-insecure-registry true
 node-defaults initial-boot default-user inodemgruser
 node-defaults initial-boot default-user-ssh-public-key "ssh-rsa
[illegible]"

 node-defaults initial-boot default-user-password
$8$wWv1014jVkdJNMPVDgEKTXC+pVQkYPuvCsdaocPwg7E=
 node-defaults netplan template "network:\n    version: 2\n    ethernets:\n    {%- if
pci_passthrough is defined and pci_passthrough == 'true' %}\n        ens192:\n
dhcp4: no\n        ens224:\n            addresses:\n                - {{K8S_SSH_IP}}/24\n
           gateway4: 10.90.154.1\n            dhcp4: no\n            nameservers:\n
         addresses: ['173.36.131.10', '72.163.128.140']\n                search:
['cisco.com']\n        {%- if ipaddr1 %}\n        ens256:\n            addresses: {{ ipaddr1
 }}\n            {%- if route1 %}\n            routes: {{ route1 }}\n            {%- endif
 %}\n        {%- endif %}\n    {%- else %}\n        ens192:\n            addresses:\n
           - {{K8S_SSH_IP}}/24\n            gateway4: 10.90.154.1\n            dhcp4: no\n
            nameservers:\n                addresses: ['173.36.131.10', '72.163.128.140']\n
                search: ['cisco.com']\n        {%- if ipaddr1 %}\n        ens224:\n
     addresses: {{ ipaddr1 }}\n            {%- if route1 %}\n            routes: {{ route1
 }}\n            {%- endif %}\n        {%- endif %}\n    {%- endif %}"
 node-defaults netplan variable-definitions ipaddr1
 exit
 node-defaults netplan variable-definitions route1
 exit
 node-defaults k8s ssh-username inodemgruser
 node-defaults k8s ssh-connection-private-key
[illegible]

 node-defaults os proxy https-proxy http://proxy-wsa.esl.cisco.com:80
 node-defaults os proxy no-proxy
127.0.0.1,localhost,dockerhub.cisco.com,devhub-docker.cisco.com,nip.io,172.22.0.0/16,10.96.0.0/16

 node-defaults os ntp enabled
 node-defaults os ntp servers 8.ntp.esl.cisco.com
 exit
 nodes ops
  k8s node-type master
  k8s ssh-ip 10.90.154.29
  k8s node-labels smi.cisco.com/node-type oam
  exit
  k8s node-labels type_infra yes
  exit
  vmware datastore "datastore1 (1)"
  vmware host 10.90.154.7
  vmware performance memory-reservation false
  vmware performance cpu-reservation false
  vmware sizing ram-mb 65536
  vmware sizing cpus 16
  vmware sizing disk-root-gb 100
  vmware sizing disk-data-gb 250
  vmware sizing disk-home-gb 10
  vmware nics "VM Network"
  exit
  vmware nics "iNode Network"
  exit
  netplan variables ipaddr1
   value "['175.175.255.29/16', '2002::afaf:ff1d/112']"
  exit
 exit
 ops-centers cee data
  repository http://charts.10.90.154.28.nip.io/cee-2020-01-1-11
  ingress-hostname inodemgr.cisco.com
  initial-boot-parameters use-volume-claims true
 initial-boot-parameters first-boot-password $8$Svqj/rduuSg0vgPEsoQbLtq41MNJqtNtekFZG2S+vs8=

  initial-boot-parameters auto-deploy true
  initial-boot-parameters single-node true
 exit
 ops-centers inode-manager data
  repository http://charts.10.90.154.28.nip.io/inodemanager-3.0.0-release-2007142325
  ingress-hostname inodemgr.cisco.com
  initial-boot-parameters use-volume-claims true
 initial-boot-parameters first-boot-password $8$n21R6mgTCIlco1xOsDa7cHaSoHChfITfT2t1jjY6VpI=

  initial-boot-parameters auto-deploy true
  initial-boot-parameters single-node true
 exit
 ops-centers opshub data
  repository http://charts.10.90.154.28.nip.io/opshub-release-2007150030
  ingress-hostname inodemgr.cisco.com
  initial-boot-parameters use-volume-claims true
 initial-boot-parameters first-boot-password $8$CYlY6zsMh4+L/0G+WQxwrHJX2A1CaAm5PizK/Y2UpGQ=

  initial-boot-parameters auto-deploy true
  initial-boot-parameters single-node true
 exit
exit
```

multinode-inode-manager-cli-config

```
clusters inode-manager-multinode
 environment atl-smi-inodemgr-lab
 addons ingress bind-ip-address 10.90.154.30
 configuration master-virtual-ip 10.90.154.30
 configuration master-virtual-ip-interface ens192
 configuration virtual-ip-vrrp-router-id 100
 configuration size          functional-test-ha
 configuration allow-insecure-registry true
 node-defaults initial-boot default-user inodemgruser
 node-defaults initial-boot default-user-ssh-public-key "ssh-rsa
[illegible]

 node-defaults initial-boot default-user-password
$8$ZuSg2Wce926IpXWdk9F2WluO+OHZbX+2I2+24Of4pPc=
 node-defaults netplan template "network:\n    version: 2\n    ethernets:\n    {%- if
pci_passthrough is defined and pci_passthrough == 'true' %}\n        ens192:\n
dhcp4: no\n        ens224:\n            addresses:\n                - {{K8S_SSH_IP}}/24\n
           gateway4: 10.90.154.1\n            dhcp4: no\n            nameservers:\n
         addresses: ['173.36.131.10', '72.163.128.140']\n                search:
['cisco.com']\n        {%- if ipaddr1 %}\n        ens256:\n            addresses: {{ ipaddr1
 }}\n            {%- if route1 %}\n            routes: {{ route1 }}\n            {%- endif
 %}\n        {%- endif %}\n    {%- else %}\n        ens192:\n            addresses:\n
           - {{K8S_SSH_IP}}/24\n            gateway4: 10.90.154.1\n            dhcp4: no\n
            nameservers:\n                addresses: ['173.36.131.10', '72.163.128.140']\n
                search: ['cisco.com']\n        {%- if ipaddr1 %}\n        ens224:\n
     addresses: {{ ipaddr1 }}\n            {%- if route1 %}\n            routes: {{ route1
 }}\n            {%- endif %}\n        {%- endif %}\n    {%- endif %}"
 node-defaults netplan variable-definitions ipaddr1
 exit
 node-defaults netplan variable-definitions route1
 exit
 node-defaults k8s ssh-username inodemgruser
 node-defaults k8s ssh-connection-private-key
[illegible]

 node-defaults vmware performance memory-reservation false
 node-defaults vmware performance cpu-reservation false
 node-defaults os proxy https-proxy http://proxy-wsa.esl.cisco.com:80
 node-defaults os proxy no-proxy
127.0.0.1,localhost,dockerhub.cisco.com,devhub-docker.cisco.com,nip.io,172.22.0.0/16,10.96.0.0/16

 node-defaults os ntp enabled
 node-defaults os ntp servers 8.ntp.esl.cisco.com
 exit
 nodes app-1
  k8s node-type worker
  k8s ssh-ip 10.90.154.40
  k8s node-labels smi.cisco.com/node-type app
  exit
  vmware datastore "datastore1 (1)"
  vmware host 10.90.154.7
  vmware sizing ram-mb 16384
  vmware sizing cpus 4
  vmware sizing disk-root-gb 50
  vmware sizing disk-data-gb 50
  vmware sizing disk-home-gb 10
  vmware nics "VM Network"
  exit
  vmware nics "iNode Network"
  exit
  netplan variables ipaddr1
   value "['175.175.255.40/16', '2002::afaf:ff28/112']"
  exit
  netplan variables route1
   value "[{to: 192.175.175.0/16, via: 175.175.254.254}, {to: 2002::C0af:af00/120, via:
2002::afaf:fefe}]"
  exit
 exit
 nodes app-2
  k8s node-type worker
  k8s ssh-ip 10.90.154.41
  k8s node-labels smi.cisco.com/node-type app
  exit
  vmware datastore "datastore1 (2)"
  vmware host 10.90.154.9
  vmware sizing ram-mb 16384
  vmware sizing cpus 4
  vmware sizing disk-root-gb 50
  vmware sizing disk-data-gb 50
  vmware sizing disk-home-gb 10
  vmware nics "VM Network"
  exit
  vmware nics "iNode Network"
  exit
  netplan variables ipaddr1
   value "['175.175.255.41/16', '2002::afaf:ff29/112']"
  exit
  netplan variables route1
   value "[{to: 192.175.175.0/16, via: 175.175.254.254}, {to: 2002::C0af:af00/120, via:
2002::afaf:fefe}]"
  exit
 exit
 nodes app-3
  k8s node-type worker
  k8s ssh-ip 10.90.154.42
  k8s node-labels smi.cisco.com/node-type app
  exit
  vmware datastore "datastore1 (3)"
  vmware host 10.90.154.11
  vmware sizing ram-mb 16384
  vmware sizing cpus 4
  vmware sizing disk-root-gb 50
  vmware sizing disk-data-gb 50
  vmware sizing disk-home-gb 10
  vmware nics "VM Network"
  exit
  vmware nics "iNode Network"
  exit
  netplan variables ipaddr1
   value "['175.175.255.42/16', '2002::afaf:ff2a/112']"
  exit
  netplan variables route1
   value "[{to: 192.175.175.0/16, via: 175.175.254.254}, {to: 2002::C0af:af00/120, via:
2002::afaf:fefe}]"
  exit
 exit
 nodes control-plane-1
  k8s node-type master
  k8s ssh-ip 10.90.154.31
  vmware datastore "datastore1 (1)"
  vmware host 10.90.154.7
  vmware sizing ram-mb 16384
  vmware sizing cpus 2
  vmware sizing disk-data-gb 20
  vmware sizing disk-home-gb 5
 exit
```

```
nodes control-plane-2
 k8s node-type master
 k8s ssh-ip 10.90.154.32
 vmware datastore "datastore1 (2)"
 vmware host 10.90.154.9
 vmware sizing ram-mb 16384
 vmware sizing cpus 2
 vmware sizing disk-data-gb 20
 vmware sizing disk-home-gb 5
exit
nodes control-plane-3
 k8s node-type master
 k8s ssh-ip 10.90.154.33
 vmware datastore "datastore1 (3)"
 vmware host 10.90.154.11
 vmware sizing ram-mb 16384
 vmware sizing cpus 2
 vmware sizing disk-data-gb 20
 vmware sizing disk-home-gb 5
exit
nodes etcd-1
 k8s node-type etcd
 k8s ssh-ip 10.90.154.34
 vmware datastore "datastore1 (1)"
 vmware host 10.90.154.7
 vmware sizing ram-mb 16384
 vmware sizing cpus 2
 vmware sizing disk-data-gb 20
 vmware sizing disk-home-gb 5
exit
nodes etcd-2
 k8s node-type etcd
 k8s ssh-ip 10.90.154.35
 vmware datastore "datastore1 (2)"
 vmware host 10.90.154.9
 vmware sizing ram-mb 16384
 vmware sizing cpus 2
 vmware sizing disk-data-gb 20
 vmware sizing disk-home-gb 5
exit
nodes etcd-3
 k8s node-type etcd
 k8s ssh-ip 10.90.154.36
 vmware datastore "datastore1 (3)"
 vmware host 10.90.154.11
 vmware sizing ram-mb 16384
 vmware sizing cpus 2
 vmware sizing disk-data-gb 20
 vmware sizing disk-home-gb 5
exit
nodes infra-1
 k8s node-type worker
 k8s ssh-ip 10.90.154.37
 k8s node-labels smi.cisco.com/node-type oam
 exit
 vmware datastore "datastore1 (1)"
 vmware host 10.90.154.7
 vmware performance cpu-reservation false
 vmware sizing ram-mb 65536
 vmware sizing cpus 14
 vmware sizing disk-root-gb 90
 vmware sizing disk-data-gb 200
 vmware sizing disk-home-gb 10
exit
```

```
nodes infra-2
 k8s node-type worker
 k8s ssh-ip 10.90.154.38
 k8s node-labels smi.cisco.com/node-type oam
 exit
 vmware datastore "datastore1 (2)"
 vmware host 10.90.154.9
 vmware performance cpu-reservation false
 vmware sizing ram-mb 65536
 vmware sizing cpus 14
 vmware sizing disk-root-gb 90
 vmware sizing disk-data-gb 200
 vmware sizing disk-home-gb 10
exit
nodes infra-3
 k8s node-type worker
 k8s ssh-ip 10.90.154.39
 k8s node-labels smi.cisco.com/node-type oam
 exit
 vmware datastore "datastore1 (3)"
 vmware host 10.90.154.11
 vmware performance cpu-reservation false
 vmware sizing ram-mb 65536
 vmware sizing cpus 14
 vmware sizing disk-root-gb 90
 vmware sizing disk-data-gb 200
 vmware sizing disk-home-gb 10
exit
virtual-ips vg1
 vrrp-interface ens224
 vrrp-router-id 101
 ipv4-addresses 175.175.255.30
  mask      16
  broadcast 175.175.255.255
  device    ens224
 exit
 ipv6-addresses 2002::afaf:ff1e
  mask   112
  device ens224
 exit
 hosts app-1
 exit
 hosts app-2
 exit
 hosts app-3
 exit
exit
ops-centers cee data
 repository http://charts.10.90.154.28.nip.io/cee-2020-01-1-11
 ingress-hostname inodemgr.cisco.com
 initial-boot-parameters use-volume-claims true
 initial-boot-parameters first-boot-password $8$g2AHhNEmz6I4SirSyaetIKZWEuZRRdC0jNy5S3364bM=

 initial-boot-parameters auto-deploy true
 initial-boot-parameters single-node false
exit
ops-centers inode-manager data
 repository http://charts.10.90.154.28.nip.io/inodemanager-3.0.0-release-2007142325
 ingress-hostname inodemgr.cisco.com
 initial-boot-parameters use-volume-claims true
 initial-boot-parameters first-boot-password $8$okH+v5L5m+7WC3FnjewKfElMD2DG9PFuXrFf5EBw5MA=

 initial-boot-parameters auto-deploy true
 initial-boot-parameters single-node false
 exit
 ops-centers opshub data
  repository
https://engci-maven-master.cisco.com/artifactory/smi-fuse-internal-snapshot/inode-manager/opshub-product/main/

  ingress-hostname inodemgr.cisco.com
  username   inode-manager-deployer
  password   $8$U8Rrbte++xl6cT90xJ56FGGf/OCeK6prLOWy7J8u3JbBjPdKlYXda2uYX/nOjo2W
  initial-boot-parameters use-volume-claims true
  initial-boot-parameters first-boot-password $8$J52Obr71gVDFvH2HtI7s65K6pFozZxgAbxZKhW4BSoc=

  initial-boot-parameters auto-deploy true
  initial-boot-parameters single-node false
 exit
exit
```

Deploy iNode Manager Cluster

Deploying a New iNode Manager Cluster

To deploy a new iNode Manager cluster, run the sync command as follows to deploy a new SMI cluster, or to update an existing cluster.

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

Enter 'yes' at the prompt. This command launches the deployment as a background sync job.

Redeploying iNode Manager Cluster

To remove and redeploy a cluster after setting up the cluster, run the following command:

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

This command removes the VMs of the cluster and purges the data disks before redeploying the cluster.

Caution Make sure you back up the data before redeploying the cluster. This command wipes the configuration data and persistent user data permanently from your iNode Manager cluster.

Updating an Existing iNode Manager Cluster

To modify the cluster configuration after setting up a cluster, update the configuration and run the following command:

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

Note This command works if you change the ops-center configuration, such as changing the repository field to use a different product version. Some changes, however, will not work with resync. These changes include network configuration change or node level changes. In such cases, you must redeploy the cluster.

Monitor iNode Manager Cluster Deployment

To monitor the status of the sync job, use the following command:

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

This command shows the logs from the background sync job continuously. Press Ctrl+C to quit monitoring, without impacting the underlying sync job. You can rerun the monitor command at any time to see the latest status.

The sync job runs an Ansible script to install the cluster. The following example shows a sample monitor log.

```
LAY RECAP *********************************************************************
localhost                  : ok=1    changed=0    unreachable=0    failed=0
ops                        : ok=250  changed=116  unreachable=0    failed=0

Wednesday 08 April 2020  23:46:53 +0000 (0:00:00.268)       0:05:24.420 *******
===============================================================================
2020-04-08 23:46:53.821 DEBUG cluster_sync.inodemgr-aio: Cluster sync successful
2020-04-08 23:46:53.822 DEBUG cluster_sync.inodemgr-aio: Ansible sync done
```

Note If the sync job times out at any step, rerun the sync command. It continues from where it left previously.

If you modify a cluster's configuration after deployment, run the sync command to update the cluster.

To remove the VMs and start from scratch, run the sync command with the additional options as follows:

```
clusters <cluster> actions sync run force-vm-redeploy true purge-data-disk true
```

Install Cisco iNode Manager with Autodeployer

Prerequisite: Install Intelligent Node Software on GS7000 iNode

As a prerequisite for iNode Manager 3.x installation or migration, ensure that the GS7000 iNode runs 2.0.8 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 *02.00.08* 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 6.5.0)*
- *vCenter* (version 6.7.0 or above): manager for the vSphere infrastructure that hosts the VMs for the SMI clusters.

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.

The multinode cluster provides full high-availability support and is the only recommended cluster size for production.

The following is the resource allocation for various node (VM) types:

Node type	CPUs	RAM size (GB)	Disk size - root (GB)	Disk size - home (GB)	Disk size - data (GB)
Deployer	4	16	5	64	40
AIO	16	64	100	10	200
Control-Plane	2	16	20	5	20
etcd	2	16	20	5	20
infra	14	64	100	10	200
ops	4	16	100	10	100

High-Level Overview of Installation Workflow

1. Prepare Staging Server, on page 26
2. Prepare a Cluster Configuration File, on page 28
3. Deploy the Cluster, on page 38

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-config.yaml
│   ├── aio-inode-manager-standby-config.yaml
│   ├── deployer-sample-config.yaml
│   ├── 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> ]
      nameservers: [ <LIST of DNS servers> ]
      search-domains: [ <LIST of search domains ]
      ntp: <ntp server name or IP address>
      https-proxy: <HTTP proxy server>
      no-proxy: <list of domains not using proxy>
```

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.

FQDN disabled:

```
clusters:
  <SMI cluster name>:
      type: "inode-manager"
      environment: <environment of vCenter hosting the SMI cluster>
      gateway: <gateway IP address>
      ingress-ip: <virtual ip of the management network>
     # Recommended to set the below field to true for Multi-Node and false for AIO deployment

      enable_logs_fwd: <"true"/"false">
      username: <user name for the SMI cluster>
      # 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>
      # The following two fields are for multi-node cluster only
      primary-vip: <virtual IP address for the management network in CIDR format>
      vrouter-id: <VRRP ID for the management network>

      # For Multi-Node cluster only
      nodes:
        -  host: <ESXi host 1 IP address>
           addresses: [ <Control-Plane 1 IP>, <ETCD 1 IP>, <INTRA 1 IP>, <APP 1 IP> ]
           datastore: <vCenter datastore for host 1>
           ops:
             interfaces:
               - vip: [ <LIST of virtual IP for CIN network in CIDR format> ]
                 vrouter-id: <VRRP ID for CIN network>
                 addresses: [ <LIST of IP addresses for CIN network in CIDR format> ]
                 nics: <vCenter network for CIN>
                 # CIN routing table (optional)
                 routes:
                   - { dest: [ <LIST of destination subnets> ], nhop: <next hop IP> }
                   - { dest: [ <LIST of destination subnets> ], nhop: <next hop IP> }
        -  host: <ESXi host 2 IP address>
           addresses: [ <Control-Plane 2 IP>, <ETCD 2 IP>, <INTRA 2 IP>, <APP 2 IP> ]
           datastore: <vCenter datastore for host 2>
           ops:
             interfaces:
               - addresses: [ <LIST of IP addresses for CIN network in CIDR format> ]
        -  host: <ESXi host 3 IP address>
           addresses: [ <Control-Plane 3 IP>, <ETCD 3 IP>, <INTRA 3 IP>, <APP 3 IP> ]
           datastore: <vCenter datastore for host 3>
           ops:
             interfaces:
               - addresses: [ <LIST of IP addresses for CIN network in CIDR format> ]

      # For Single-Node cluster only
      nodes:
        -  host: <ESXi host IP address>
           addresses: [ <AIO VM IP address> ]
           datastore: <vCenter datastore for host>
           ops:
             interfaces:
               - addresses: [ <LIST of IP addresses for CIN network in CIDR format> ]
                 nics: <vCenter network for CIN>
                 routes:
                   - { dest: [ <LIST of destination subnets> ], nhop: <next hop IP> }
                   - { dest: [ <LIST of destination subnets> ], nhop: <next hop IP> }
```

FQDN enabled:

```
clusters:
  <SMI cluster name>
      type: "inode-manager"
      environment: <environment of vCenter hosting the SMI cluster>
      gateway: <gateway IP address>
      # ingress-hostname only supports valid FQDN
      ingress-hostname: "inodemgr.example.com"
     # Recommended to set the below field to true for Multi-Node and false for AIO deployment

      enable_logs_fwd: <"true"/"false">
      username: <user name for the SMI cluster>
      # 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>
      # The following two fields are for multi-node cluster only
      primary-vip: <virtual IP address for the management network in CIDR format>
      vrouter-id: <VRRP ID for the management network>

      # For Multi-Node cluster only
      nodes:
        -  host: <ESXi host 1 IP address>
           addresses: [ <Control-Plane 1 IP>, <ETCD 1 IP>, <INTRA 1 IP>, <APP 1 IP> ]
           datastore: <vCenter datastore for host 1>
           ops:
             interfaces:
               - vip: [ <LIST of virtual IP for CIN network in CIDR format> ]
                 vrouter-id: <VRRP ID for CIN network>
                 addresses: [ <LIST of IP addresses for CIN network in CIDR format> ]
                 nics: <vCenter network for CIN>
                 # CIN routing table (optional)
                 routes:
                   - { dest: [ <LIST of destination subnets> ], nhop: <next hop IP> }
                   - { dest: [ <LIST of destination subnets> ], nhop: <next hop IP> }
        -  host: <ESXi host 2 IP address>
           addresses: [ <Control-Plane 2 IP>, <ETCD 2 IP>, <INTRA 2 IP>, <APP 2 IP> ]
           datastore: <vCenter datastore for host 2>
           ops:
             interfaces:
               - addresses: [ <LIST of IP addresses for CIN network in CIDR format> ]
        -  host: <ESXi host 3 IP address>
           addresses: [ <Control-Plane 3 IP>, <ETCD 3 IP>, <INTRA 3 IP>, <APP 3 IP> ]
           datastore: <vCenter datastore for host 3>
           ops:
             interfaces:
               - addresses: [ <LIST of IP addresses for CIN network in CIDR format> ]

      # For Single-Node cluster only
      nodes:
        -  host: <ESXi host IP address>
           addresses: [ <AIO VM IP address> ]
           datastore: <vCenter datastore for host>
           ops:
             interfaces:
               - addresses: [ <LIST of IP addresses for CIN network in CIDR format> ]
                 nics: <vCenter network for CIN>
                 routes:
                   - { dest: [ <LIST of destination subnets> ], nhop: <next hop IP> }
                   - { dest: [ <LIST of destination subnets> ], nhop: <next hop IP> }
```

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

```
environments:
  atl-smi-inodemgr-lab:
    server: "cabu-sdn-vc.cisco.com"
    username: "cvideo.gen@cisco.com"
    datacenter: "Cloud Video Datacenter"
    cluster: "iNodeManager"
    nics: [ "VM Network" ]
    nameservers: [ 173.36.131.10, 72.163.128.140 ]
    search-domains: [ cisco.com ]
    ntp: 8.ntp.esl.cisco.com
    https-proxy: "http://proxy-wsa.esl.cisco.com:80"
    no-proxy:
"127.0.0.1,localhost,dockerhub.cisco.com,devhub-docker.cisco.com,nip.io,172.22.0.0/16,10.96.0.0/16"

deployers:
  inode-manager-deployer-1:
      environment: atl-smi-inodemgr-lab
      address: 10.90.154.28/24
      gateway: 10.90.154.1
      username: cloud-user
      private-key-file: inodemgr.pem
      host: 10.90.154.7
      datastore: "datastore1 (1)"
      # ingress-hostname only supports valid FQDN
      ingress-hostname: "deployer.example.com"
```

aio-inode-manager-config.yaml

```
environments:
  atl-smi-inodemgr-lab:
    server: "cabu-sdn-vc.cisco.com"
    username: "cvideo.gen@cisco.com"
    datacenter: "Cloud Video Datacenter"
    cluster: "iNodeManager"
    nics: [ "VM Network" ]
    nameservers: [ 173.36.131.10, 72.163.128.140 ]
    search-domains: [ cisco.com ]
    ntp: 8.ntp.esl.cisco.com
    https-proxy: "http://proxy-wsa.esl.cisco.com:80"
    no-proxy:
"127.0.0.1,localhost,dockerhub.cisco.com,devhub-docker.cisco.com,nip.io,172.22.0.0/16,10.96.0.0/16"

deployers:
  inode-manager-deployer-1:
      environment: atl-smi-inodemgr-lab
      address: 10.90.154.28/24
      gateway: 10.90.154.1
      username: cloud-user
      private-key-file: inodemgr.pem
      host: 10.90.154.7
      datastore: "datastore1 (1)"
      # ingress-hostname only supports valid FQDN
      ingress-hostname: "deployer.example.com"

clusters:
  inode-manager-aio:
      type: inode-manager
      enable_logs_fwd: "true"
      environment: atl-smi-inodemgr-lab
      username: inodemgruser
      private-key-file: inodemgr.pem
      gateway: 10.90.154.1
      # ingress-hostname only supports valid FQDN
      ingress-hostname: "inodemgr.example.com"
      nodes:
        -  host: 10.90.154.7
           datastore: "datastore1 (1)"
           addresses: [ 10.90.154.29/24 ]
           ops:
             interfaces:
               - nics: "iNode Network"
                 addresses: [ 175.175.255.29/16, "2002::afaf:ff1d/112" ]
```

multinode-inode-manager-config.yaml

```
environments:
  atl-smi-inodemgr-lab:
    server: "cabu-sdn-vc.cisco.com"
    username: "cvideo.gen@cisco.com"
    datacenter: "Cloud Video Datacenter"
    cluster: "iNodeManager"
    nics: [ "VM Network" ]
    nameservers: [ 173.36.131.10, 72.163.128.140 ]
    search-domains: [ cisco.com ]
    ntp: 8.ntp.esl.cisco.com
    https-proxy: "http://proxy-wsa.esl.cisco.com:80"
    no-proxy:
"127.0.0.1,localhost,dockerhub.cisco.com,devhub-docker.cisco.com,nip.io,172.22.0.0/16,10.96.0.0/16"

deployers:
  inode-manager-deployer-1:
      environment: atl-smi-inodemgr-lab
      address: 10.90.154.28/24
      gateway: 10.90.154.1
      username: cloud-user
      private-key-file: inodemgr.pem
      host: 10.90.154.7
      datastore: "datastore1 (1)"
      # ingress-hostname only supports valid FQDN
      ingress-hostname: "deployer.example.com"

clusters:
  inode-manager-multinode:
      type: inode-manager
      environment: atl-smi-inodemgr-lab
      username: inodemgruser
      private-key-file: inodemgr.pem
      gateway: 10.90.154.1
      primary-vip: 10.90.154.30/24
      # ingress-hostname only supports valid FQDN
      ingress-hostname: "inodemgr.example.com"
      vrouter-id: 100
      nodes:
        -  host: 10.90.154.7
           datastore: "datastore1 (1)"
           addresses: [ 10.90.154.31, 10.90.154.34, 10.90.154.37, 10.90.154.40 ]
           ops:
             interfaces:
               - addresses: [ 175.175.255.40/24, "2002::afaf:ff28/112" ]
                 nics: "iNode Network"
                 vip: [ 175.175.255.30/24, "2002::afaf:ff1e/112" ]
                 vrouter-id: 101
                 # Below is just a route example (Applicable for CHN lab, not ATL Lab)
                 routes:
                   - {dest: [ 192.175.175.0/24 ], nhop: 175.175.254.254 }
                   - {dest: [ "2002::C0af:af00/120" ], nhop: "2002::afaf:fefe" }
        -  host: 10.90.154.9
           datastore: "datastore1 (2)"
           addresses: [ 10.90.154.32, 10.90.154.35, 10.90.154.38, 10.90.154.41 ]
           ops:
             interfaces:
               - addresses: [ 175.175.255.41/24, "2002::afaf:ff29/112" ]
        -  host: 10.90.154.11
           datastore: "datastore1 (3)"
           addresses: [ 10.90.154.33, 10.90.154.36, 10.90.154.39, 10.90.154.42 ]
           ops:
             interfaces:
               - addresses: [ 175.175.255.42/24, "2002::afaf:ff2a/112"  ]
```

aio-inode-manager-standby-config.yaml

```
environments:
  atl-smi-inodemgr-lab:
    server: "cabu-sdn-vc.cisco.com"
    username: "cvideo.gen@cisco.com"
    datacenter: "Cloud Video Datacenter"
    cluster: "iNodeManager"
    nics: [ "VM Network" ]
    nameservers: [ 173.36.131.10, 72.163.128.140 ]
    search-domains: [ cisco.com ]
    ntp: 8.ntp.esl.cisco.com
    https-proxy: "http://proxy-wsa.esl.cisco.com:80"
    no-proxy:
"127.0.0.1,localhost,dockerhub.cisco.com,devhub-docker.cisco.com,nip.io,172.22.0.0/16,10.96.0.0/16"

deployers:
  inode-manager-deployer-1:
      environment: atl-smi-inodemgr-lab
      address: 10.90.154.28/24
      gateway: 10.90.154.1
      username: cloud-user
      private-key-file: inodemgr.pem
      host: 10.90.154.7
      datastore: "datastore1 (1)"
      # ingress-hostname only supports valid FQDN
      ingress-hostname: "deployer.example.com"

clusters:
  inode-manager-aio:
      type: inode-manager
      environment: atl-smi-inodemgr-lab
      username: inodemgruser
      private-key-file: inodemgr.pem
      gateway: 10.90.154.1
      # ingress-hostname only supports valid FQDN
      ingress-hostname: "inodemgr.example.com"
      nodes:
        -  host: 10.90.154.7
           datastore: "datastore1 (1)"
           addresses: [ 10.90.154.29/24 ]
```

multinode-inode-manager-standby-config.yaml

```
environments:
  atl-smi-inodemgr-lab:
    server: "cabu-sdn-vc.cisco.com"
    username: "cvideo.gen@cisco.com"
    datacenter: "Cloud Video Datacenter"
    cluster: "iNodeManager"
    nics: [ "VM Network" ]
    nameservers: [ 173.36.131.10, 72.163.128.140 ]
    search-domains: [ cisco.com ]
    ntp: 8.ntp.esl.cisco.com
    https-proxy: "http://proxy-wsa.esl.cisco.com:80"
    no-proxy:
"127.0.0.1,localhost,dockerhub.cisco.com,devhub-docker.cisco.com,nip.io,172.22.0.0/16,10.96.0.0/16"

deployers:
  inode-manager-deployer-1:
      environment: atl-smi-inodemgr-lab
      address: 10.90.154.28/24
      gateway: 10.90.154.1
      username: cloud-user
      private-key-file: inodemgr.pem
      host: 10.90.154.7
      datastore: "datastore1 (1)"
      # ingress-hostname only supports valid FQDN
      ingress-hostname: "deployer.example.com"

clusters:
  inode-manager-multinode:
      type: inode-manager
      environment: atl-smi-inodemgr-lab
      username: inodemgruser
      private-key-file: inodemgr.pem
      gateway: 10.90.154.1
      primary-vip: 10.90.154.30/24
      # ingress-hostname only supports valid FQDN
      ingress-hostname: "inodemgr.example.com"
      vrouter-id: 100
      nodes:
        -  host: 10.90.154.7
           datastore: "datastore1 (1)"
           addresses: [ 10.90.154.31, 10.90.154.34, 10.90.154.37, 10.90.154.40 ]
        -  host: 10.90.154.9
           datastore: "datastore1 (2)"
           addresses: [ 10.90.154.32, 10.90.154.35, 10.90.154.38, 10.90.154.41 ]
        -  host: 10.90.154.11
           datastore: "datastore1 (3)"
           addresses: [ 10.90.154.33, 10.90.154.36, 10.90.154.39, 10.90.154.42 ]
```

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 yaml. 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 vmdk file
is /opt/deployer/work/cluster-deployer-airgap.vmdk:

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 public key file.

  ```
  ssh -i <private-key-file> <deployer-user>@<deployer-address>
  ```

- Use **kubectl** command to find the internal IP address of the Operation Center service:

  ```
  kubectl get svc ops-center-smi-cluster-deployer -n smi
  ```

- Look for the CLUSTER-IP field from the output of the preceding step. Use it to SSH into the deployer:

  ```
  ssh admin@<cluster-ip> -p 2024
  ```

- 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

   ```
   kubectl exec -it ops-center-smi-cluster-deployer-xxxxxxxxx-yyyyy --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.

Handling deployment failure on ESXI versions above 6.5

With ESXI versions above 6.5, there is an OVFtool version compatibility issue with the iNode Manager release bundle. This requires the user to manually download ovftool version 4.4.0 and work around the failure.

Error logs

```
  \- '2021-02-04 19:43:21.995 ERROR root: Opening VMX source:
/opt/deploy/ubuntu-18.04-template.vmx'

  \- 'Opening VI target: vi://username@vsphere-host.com:443/lab-data-center/host/iNodeManager'


  \- 'Deploying to VI:  vi://username@vsphere-host.com:443/lab-data-center/host/iNodeManager'


  \- Transfer Failed

  \- Transfer Failed

  \- 'Error: Failed to send http data'

  \- Completed with errors
```

Cause

The ovftool CLI version that is bundled with iNode Manager is 4.2.0 and ESXI versions above 6.5 requires the latest ovftool. Reference: https://kb.vmware.com/s/article/71039.

Workaround

To solve this problem, download the ovftool version 4.4.0 to SMI deployer VM and perform the following steps.

1. SSH to the deployer VM.
2. Get the SMI ops-center pod name using the following command.

```
kubectl get pod -n smi | grep ops-center
ops-center-smi-cluster-deployer-6c5d899458-jrrkm                  6/6     Running   8
        132d
```

3. Copy the VMX file and VMdk file locally to the VM from the pod.

```
kubectl cp -c cluster-sync
smi/ops-center-smi-cluster-deployer-6c5d899458-jrrkm:/opt/deploy/ubuntu-18.04-template.vmdk
kubectl cp -c cluster-sync
smi/ops-center-smi-cluster-deployer-6c5d899458-jrrkm:/opt/deploy/ubuntu-18.04-template.vmx
```

4. Run the following command with the ovftool version 4.4.0 downloaded earlier on the current directory where the above command was executed from.

```
ovftool --X:skipContentLength --overwrite --name='.smi-base-image-20200730'
--diskMode=thin --acceptAllEulas -ds='<<datastore-name>>' --noSSLVerify
'./ubuntu-18.04-template.vmx'
'vi://<<username>>:<<password>>@<<vcenter-server-ip:port>>/<<datacenter-name>>/host/<<vcenter-cluster-name>>'
```

Note Replace the place-holders << >> with appropriate values.

5. Upon successful upload, delete the existing VM(s) that failed to deploy from the Vsphere web console.

6. Login to SMI deployer CLI *https://cli.smi-cluster-deployer.\<deployer-ip\>.nip.io* and run the command **clusters** *cluster_name* **actions sync run debug true** to re-trigger the deployment.

Helm chart deployment missing

After installation, check the output of the **helm ls** command.

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

Post Installation/Upgrade Checklist

Step 1 Check NTP Time Sync.

a) 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.

a) Check if all helm charts are in "DEPLOYED" state.

Example:

```
helm ls -a
cee-data-cnat-monitoring                1              Wed Aug 19 07:29:00 2020        DEPLOYED
        cnat-monitoring-0.6.0-0-6-0023-200806180041-b7ceaaa             2020.01.1-14
          cee-data
cee-data-ops-center                     1              Wed Aug 19 07:23:50 2020        DEPLOYED
        cee-ops-center-0.6.0-0-6-0249-200730182303-49fe2c7              2020.01.1-14
          cee-data
cee-data-product-documentation          1              Wed Aug 19 07:28:59 2020        DEPLOYED
        product-documentation-0.6.0-0-6-0037-200730201839-360af73       2020.01.1-14
          cee-data
cee-data-pv-manager                     1              Wed Aug 19 07:28:59 2020        DEPLOYED
        pv-manager-0.2.0-0-3-0010-200731040356-7e38fd8                  2020.01.1-14
          cee-data
cee-data-smi-autoheal                   1              Wed Aug 19 07:28:59 2020        DEPLOYED
        smi-autoheal-0.1.1-master-0007-200730204255-8c59ad4             2020.01.1-14
          cee-data
cee-data-smi-show-tac                   1              Wed Aug 19 07:28:59 2020        DEPLOYED
        smi-show-tac-0.2.0-0-2-0114-200731061558-c07b9a9                2020.01.1-14
          cee-data
cee-data-storage-provisioner            1              Wed Aug 19 07:28:59 2020        DEPLOYED
        storage-provisioner-0.2.0-0-3-0081-200731040450-66bbb04         2020.01.1-14
          cee-data
inode-manager-data-inode-manager-app    1              Wed Aug 19 07:29:18 2020        DEPLOYED
       inode-manager-app-0.1.0-main-0009-200814062935-faa704f        inodemanager-3.0.1-release
      inode-manager-data
inode-manager-data-ops-center           1              Wed Aug 19 07:24:07 2020        DEPLOYED
       inode-manager-ops-center-0.1.0-main-0019-200805232831-1c9...    inodemanager-3.0.1-release
      inode-manager-data
kubernetes-dashboard                    2              Tue Aug 18 16:44:55 2020        DEPLOYED
        kubernetes-dashboard-1.10.1-master-0013-190605174754-8d7080d    1.10.1
          kube-system
nginx-ingress                           2              Tue Aug 18 16:44:50 2020        DEPLOYED
        nginx-ingress-1.5.0-master-0078-200417033703-5484f87            0.26.1
          nginx-ingress
opshub-data-ops-center                  1              Wed Aug 19 07:24:18 2020        DEPLOYED
        opshub-ops-center-0.5.3-smartphy-0049-200805231738-906d44c      opshub-main-release
          opshub-data
opshub-data-opshub-infra-app            1              Wed Aug 19 07:29:12 2020        DEPLOYED
        opshub-infra-app-0.1.0-main-0048-200806120249-7dc6157           opshub-main-release
          opshub-data
smi-cluster-maintainer                  2              Tue Aug 18 16:44:44 2020        DEPLOYED
        smi-cluster-maintainer-1.1.0-master-0005-200324060503-218...
          kube-system
smi-keepalived-vips                     2              Tue Aug 18 16:44:59 2020        DEPLOYED
        smi-keepalived-1.0.0-master-0061-200414235846-e656df5
          smi-vips
```

```
ss-cert-prov                           2              Tue Aug 18 16:44:29 2020        DEPLOYED
        self-signed-cert-provisioner-1.0.0-master-0018-2004091602...
          smi-certs
```

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 --purge <failed-chart-name>
(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 Setup firewall rules.

Note Set up firewall rules to ensure the inode-service-manager application running on port 30628 is not accessible externally. Access to inode-service-manager APIs should be available only by using the nginx gateway.

a) For AIO Deployment, add the following rules on the "ops" node.

```
sudo iptables -A INPUT -p tcp --destination-port 30628 -m iprange --src-range
192.168.0.0-192.168.255.255 -j ACCEPT
sudo iptables -A INPUT -p tcp --destination-port 30628 -m iprange --src-range
<ops_node_ip>-<ops_node_ip> -j ACCEPT
sudo iptables -A INPUT -p tcp --dport 30628 -j DROP
```

b) For Multi-Node Deployment, add the following rules on all the "app-x" nodes.

```
sudo iptables -A INPUT -p tcp --destination-port 30628 -m iprange --src-range
192.168.0.0-192.168.255.255 -j ACCEPT
sudo iptables -A INPUT -p tcp --destination-port 30628 -m iprange --src-range
<cluster_start_ip>-<cluster_end_ip> -j ACCEPT
sudo iptables -A INPUT -p tcp --dport 30628 -j DROP
```

Note The <cluster_start_ip> and <cluster_end_ip> refers to the range of management IP chosen for the all the 12 nodes in the cluster.

Step 5 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 20k 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.

Step 6 Disable Logstash service

This is a mandatory step to perform after the installation/upgrade. Logstash service is vulnerable to the recent **log4j jndi** exploitation. Since this service is not required for iNode Manager, disable it by following the below steps.

a) Edit the logstash stateful set.

```
kubectl edit statefulset logstash -n opshub-data
```

b) Set **replicas** to 0 in the editor.

```
spec:
  podManagementPolicy: OrderedReady
  replicas: 0
```

c) Save and quit the editor (Press 'Esc' and :wq).

Step 7 Fix the postgres pod crash.

Note When upgrading from an older release, postgres pod may go into crash loop state due to which other dependent pods also crash leading to an OOM issue. To fix this issue, after the installation or upgrade make sure that all pods are running.

a) To check if all the pods are running, execute the following command.

```
kubectl get pods -n cee-data
```

b) If postgres pods are not running, execute the following commands to fix this error and get all the services back up and running.

```
sudo rm -rf /data/cee-data/data-postgres-*
kubectl delete pod postgres-0 postgres-1 postgres-2 pgpool-xxxxx pgpool-yyyyy grafana -n cee-data
```

Step 8 Fix the kube controller manager crash.

When using HDD, the disk writes slowly and the kube controller manager's lease to acquire a leader times out. So the following pods restarts frequently.

```
kubectl get pods -n kube-system
NAME                                                   READY   STATUS    RESTARTS    AGE
kube-controller-manager-inode-mgr-mn-183-control-plane-1   1/1     Running   10          39m21s
kube-controller-manager-inode-mgr-mn-183-control-plane-2   1/1     Running   11          36m37s
kube-controller-manager-inode-mgr-mn-183-control-plane-3   1/1     Running   11          35m59s
```

To fix this issue, perform the following steps on all control-plane-x nodes (for Multi Node)/ the ops node (for AIO).

a) Execute the following command.

```
sudo vim /etc/kubernetes/manifests/kube-controller-manager.yaml
```

b) Add the following entries after the line `- --use-service-account-credentials=true`. Make sure to follow the existing indentation.

```
- --leader-elect-lease-duration=60s
- --leader-elect-renew-deadline=30s
```

c) Save and exit the file.

Now the controller manager pods run with a longer lease timeout and do not restart frequently.

Note Perform these steps after each upgrade/install and `sync` command executed from the deployer CLI, because the `sync` command will revert any manual changes to the defaults.

Step 9 **(Optional)** Configure authentication via an LDAP server.

Access the iNode Manager application CLI `https://cli.inode-manager-data-ops-center.<cluster-IP>.nip.io` with the username `admin` and password that is configured during deployment.

```
config terminal
ldap-security ldap-server-url <URL>/<IP addr>
ldap-security ldap-username-domain <domain name for user>
ldap-security base-dn <base DN for user>
ldap-security ldap-filter <LDAP search filter>
ldap-security group-attr <Attribute string for this user>
ldap-security group-mapping <group name of this user> <NACM group>
```

In the following example, the application user is authenticated with UID@cisco.com. Create the user in an LDAP server with attribute `memberOf` and group `inode-mgr-chn` for authentication to succeed. On successful authentication, this user is given `admin` privileges on this application.

```
ldap-security ldap-server-url ldap://ds.cisco.com
ldap-security ldap-username-domain cisco.com
ldap-security base-dn DC=cisco,DC=com
ldap-security ldap-filter userPrincipalName=%s@cisco.com
ldap-security group-attr memberOf
ldap-security group-mapping inode-mgr-chn admin
```

Note Local authentication co-exists when the LDAP authentication is configured.

Note Open LDAP is currently not supported. Configure LDAP with AD (Active directory) for LDAP support.

Step 10 **(Optional)** Check ingress-hostname.

Perform the following sanity checks after each installation only if you have set the FQDN (Fully Qualified Domain Name) as the ingress-hostname. In this step, we use `inodemgr.cisco.com` as an example ingress-hostname.

```
ingress-hostname inodemgr.cisco.com
```

a) Get the ingress from `inode-manager-data` namespace.

```
kubectl get ingress -n inode-manager-data
```

b) Check if the following ingress are available.

```
NAME                                                  HOSTS
                                     ADDRESS                           PORTS     AGE
cli-ingress-inode-manager-data-ops-center
cli.inode-manager-data-ops-center.inodemgr.cisco.com
               80, 443   36d
documentation-ingress
documentation.inode-manager-data-ops-center.inodemgr.cisco.com
               80, 443   36d
inode-manager-data-inode-manager-app-robot-ui         inodemgr.cisco.com
                                                                     80, 443   36d
inode-manager-data-inode-manager-app-robot-ui-login   inodemgr.cisco.com
                                                                     80, 443   36d
inode-manager-data-inode-manager-app-sockio           inodemgr.cisco.com
                                                                     80, 443   36d
restconf-ingress-inode-manager-data-ops-center
restconf.inode-manager-data-ops-center.inodemgr.cisco.com
               80, 443   36d
smartphy-apis                                         *
                                     10.0.0.2,10.0.0.3,10.0.0.4        80        36d
```

c) If all the above ingress has the FQDN as the suffix, it confirms that the ingress is applied without any issue. If the FQDN is applied only for `ops-center` ingress, continue with the following steps.

d) Access the CLI of the `inode-manager-data` ops-center using the URL `https://cli.inode-manager-data-ops-center.inodemgr.cisco.com`. The username is admin and the password is the one configured during the deployment.

e) Set the ingress-hostname with the FQDN.

```
[cluster-name/data] inode-manager# config terminal
Entering configuration mode terminal
[cluster-name/data] inode-manager(config)# k8s ingress-host-name inodemgr.cisco.com
[cluster-name/data] inode-manager(config)# commit
[cluster-name/data] inode-manager(config)# exit
```

f) Repeat substep b after 2-3 mins and check if all the ingress has the FQDN as the suffix.

g) Repeat substeps a to f with namespaces `cee-data` and `opshub-data`.

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-f8b1fe6[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:
```

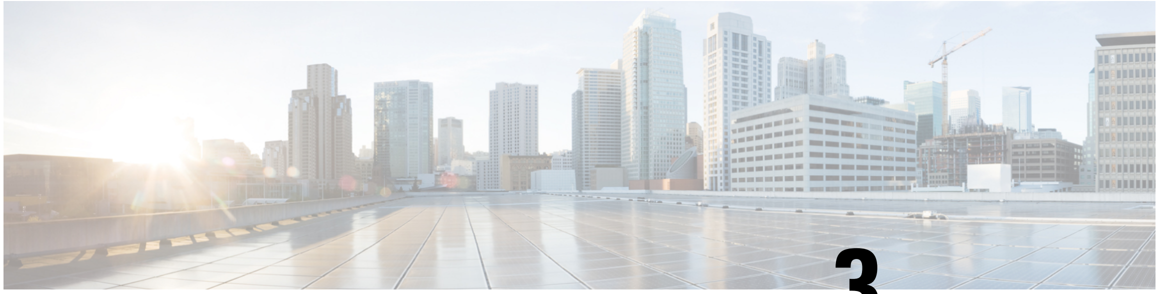
```
Host connected successfully
08:48:15-238-AUTO-DEPLOY-INFO:
Checking deployer VM status
08:48:15-238-AUTO-DEPLOY-INFO:
Checking for software package...
08:48:15-935-AUTO-DEPLOY-INFO: [34m
Software package list: {
  "Packages": {
    "package": ["opshub-release-2007150030"],
    "package": ["inodemanager-3.0.0-release-2007142325"],
    "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-aio               : 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
```



CHAPTER 3

Migrate and Upgrade Cisco iNode Manager

This chapter describes how to migrate and upgrade Cisco iNode Manager.

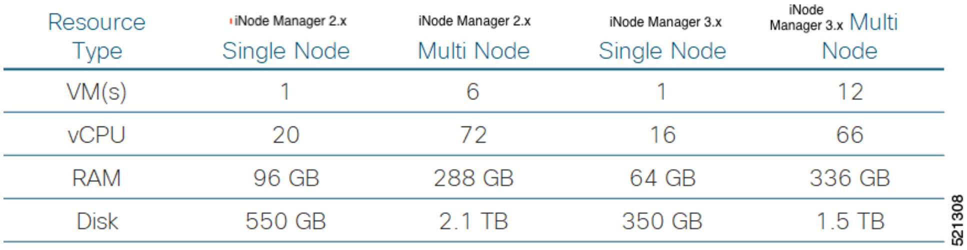
Migrate Cisco iNode Manager

Before you begin

Before migrating the iNode manager from 2.x to 3.x release, make sure the following requirements are met:

- The current iNode Manager installation is 2.2.0 or later release.
- Export the database from 2.x installation into a remote location.
- Following recommended hardware resource is available.

Figure 1: Hardware Requirement Comparison



Resource Type	iNode Manager 2.x Single Node	iNode Manager 2.x Multi Node	iNode Manager 3.x Single Node	iNode Manager 3.x Multi Node
VM(s)	1	6	1	12
vCPU	20	72	16	66
RAM	96 GB	288 GB	64 GB	336 GB
Disk	550 GB	2.1 TB	350 GB	1.5 TB

Step 1 Prepare the configuration file for the standby iNode Manager cluster.

A standby cluster is an iNode Manager cluster without CIN network configuration.

Step 2 Deploy the standby cluster as described in the Install Cisco iNode Manager with Autodeployer, on page 25.

Step 3 Log on to the UI and import database to the cluster.

Step 4 Add back the CIN network configuration to the cluster's configuration file.

Step 5 Push the new configuration to the deployer.

```
./deploy -c <config.yaml> -v
```

The `-v` option can push the cluster configuration to the deployer without triggering the sync.

Step 6 Log on to the deployer ops-center from a browser using this URL:

```
https://cli.smi-cluster-deployer.<deployer-ip>.nip.io/
```

Use `admin` as username, and the password you chose for the deployer in the deployer configuration file.

Step 7 Issue the following CLI to trigger the sync:

```
cluster <clutser-name> actions sync run force-vm-redeploy true
```

Note The `force-vm-redeploy true` option causes the VMs of the cluster to be redeployed, which is required due to the network configuration change.

If the previous CLI times out, try the following CLI:

```
cluster <cluster-name> actions sync run
```

Note Once the migration is completed successfully, update the DHCP IP of iNode Manager with the virtual IP (IPv4/IPv6) configured in the cluster configuration. This is required for iNode Auto Registration. For AIO, this will be the CIN Network IPv4/IPv6 of the "ops" node.

Upgrade Cisco iNode Manager

Before you begin

Before upgrading the iNode manager, make sure the following requirements are met:

- The cluster is running with iNode Manager 3.0.1 or later releases.
- The original installation directory `inode-manager-installer-<orig_version>` is available, with the original configuration file used.
- A deployer is available.

Step 1 Copy the iNode Manager release bundle image to the Staging Server.

Note Ensure that the new image is copied to a different location to the original installation directory.

Step 2 Extract the image content.

```
tar xvfz inode-manager-installer-<new_version>.tgz
```

Step 3 Go to the new installation directory.

```
cd inode-manager-installer-<new_version>
```

Step 4 Run the upgrade preperation script to copy new software packages to the original installation directory.

```
./upgrade-prep absolute-path-to-inode-manager-installer-<orig_version>
```

For example:

```
./upgrade-prep /home/inodemgruser/autodeployer/original-installation/inode-manager-installer-v3.1.0
```

Note This step overwrites the existing software packages in the original installation directory.

Step 5 Go to the original installation directory. Note that the version number in original installation directory is changed to the new version number.

```
cd inode-manager-installer-<orig_version>
```

For example:

```
cd /home/inodemgruser/autodeployer/original-installation/inode-manager-installer-v3.2.0
```

Step 6 Upgrade the deployer with new software packages.

```
./deploy -c <config.yaml> -u
```

This step downloads the new software packages to the deployer. The original software packages in the deployer remain untouched.

Step 7 Upgrade the iNode Manager cluster.

```
./deploy -c <config.yaml> -i upgrade
```

If the original installation directory is not available, you can skip Step 4 and 5, and perform Step 6 and 7 inside the new installation directory instead. You need to add the configure file and the SSH private files to the new installation directory.

The deploy script will prompt you for the passwords. Make sure you use the original passwords picked when you installed the deployer and the iNode Manager cluster.

If a deployer is not available, you can skip Step 6. So in Step 7, a new deployer will be installed before upgrading the iNode Manager cluster.

CHAPTER 4

Troubleshoot Cisco iNode Manager

This chapter captures some of the common commands that are used for troubleshooting Cisco iNode Manager.

Reset Admin Password

The admin user password expires after a few months from installation. Starting from Cisco iNode manager release 3.3.0, the user is prompted to update the password before expiry when log in. If the user is failed to update the password before expiry, use the following commands to reset admin password.

Step 1 Connect to one of the control-plane nodes using SSH and get ops-center pod name.

Note If SSH login with username and password fails, use the PEM file from the staging server (used for deployment) to connect to the node using SSH.

```
kubectl get pods -n inode-manager-data | grep "ops-center-inode-manager-data-ops-center"

Sample output:
kubectl get pods -n inode-manager-data | grep "ops-center-inode-manager-data-ops-center"
ops-center-inode-manager-data-ops-center-7b648ffdc6-5cvx8    5/5     Running     0          6d19h
```

Step 2 Execute the **reset-admin** command with the pod name from previous command.

```
kubectl exec -it ops-center-inode-manager-data-ops-center-7b648ffdc6-5cvx8 -n inode-manager-data
reset-admin
Defaulting container name to confd.
Use 'kubectl describe pod/ops-center-inode-manager-data-ops-center-7b648ffdc6-5cvx8 -n
inode-manager-data' to see all of the containers in this pod.
Enter new password:
```

Renew Kubernetes Certificate

The kubernetes certificates expires 365 days/1 year after installation. The operator must renew the certificates before expiry. Use the following steps to renew the certificate.

Note This is required only for Cisco iNode Manager release 3.1.0 and 3.2.0. Starting from release 3.3.0, the certificates are auto-renewed by a certificate maintainer service. If you are failed to renew the certificate before expiry, the only option is to delete the cluster and re-deploy. Kindly backup the Database via the iNode Manager DB Export functionality from the UI frequently.

Step 1 Check expiry time of the kubernetes certificate.

```
inodemgruser@inodemgrmulti-control-plane-3:~$ sudo kubeadm alpha certs check-expiration
[check-expiration] Reading configuration from the cluster...
[check-expiration] FYI: You can look at this config file with 'kubectl -n kube-system get cm
kubeadm-config -oyaml'

CERTIFICATE               EXPIRES                 RESIDUAL TIME   CERTIFICATE AUTHORITY  EXTERNALLY
 MANAGED
admin.conf                 Oct 18, 2022 09:05 UTC   260d                                    no
apiserver                  Oct 18, 2022 09:05 UTC   260d            ca                      no
apiserver-kubelet-client   Oct 18, 2022 09:05 UTC   260d            ca                      no
controller-manager.conf    Oct 18, 2022 09:05 UTC   260d                                    no
front-proxy-client         Oct 18, 2022 09:05 UTC   260d            front-proxy-ca          no
scheduler.conf             Oct 18, 2022 09:05 UTC   260d                                    no

CERTIFICATE AUTHORITY   EXPIRES                  RESIDUAL TIME   EXTERNALLY MANAGED
ca                      Oct 16, 2031 09:01 UTC   9y              no
front-proxy-ca          Oct 16, 2031 09:01 UTC   9y              no
```

Step 2 Login to the deployer CLI using admin user name and password and run **sync** command.

Note This command must be run before the certificate expiry date.

```
[smi-deployer-202] SMI Cluster Deployer# clusters <<cluster-name>> actions sync run
This will run sync.  Are you sure? [no,yes] yes
message accepted
[smi-deployer-202] SMI Cluster Deployer#
```

Step 3 Repeat step 1 to confirm that the certificates are renewed.

```
inodemgruser@inodemgrmulti-control-plane-3:~$ sudo kubeadm alpha certs check-expiration
[check-expiration] Reading configuration from the cluster...
[check-expiration] FYI: You can look at this config file with 'kubectl -n kube-system get cm
kubeadm-config -oyaml'

CERTIFICATE               EXPIRES                 RESIDUAL TIME   CERTIFICATE AUTHORITY  EXTERNALLY
 MANAGED
admin.conf                 Jan 31, 2023 08:51 UTC   364d                                    no
apiserver                  Jan 31, 2023 08:51 UTC   364d            ca                      no
apiserver-kubelet-client   Jan 31, 2023 08:51 UTC   364d            ca                      no
controller-manager.conf    Jan 31, 2023 08:51 UTC   364d                                    no
front-proxy-client         Jan 31, 2023 08:51 UTC   364d            front-proxy-ca          no
scheduler.conf             Jan 31, 2023 08:51 UTC   364d                                    no

CERTIFICATE AUTHORITY   EXPIRES                  RESIDUAL TIME   EXTERNALLY MANAGED
ca                      Oct 16, 2031 09:01 UTC   9y              no
front-proxy-ca          Oct 16, 2031 09:01 UTC   9y              no
inodemgruser@inodemgrmulti-control-plane-3:~$
```