



## **Cisco ACI and Docker EE Integration**

**New and Changed Information 2**

Cisco ACI and Docker EE Integration 2

System Requirements for Cisco ACI Docker EE Integration 2

Hardware Requirements 3

Workflow for Cisco ACI Docker EE Integration 3

Deploying Applications 9

Revised: February 4, 2020

## New and Changed Information

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

**Table 1: New and changed information**

| Cisco APIC Release Version | Feature                               | Description  |
|----------------------------|---------------------------------------|--|
| Cisco APIC 4.2(1)          | Docker EE 2.1 support with Kubernetes | Beginning with this release, Docker EE is supported with Kubernetes.     |
| Cisco APIC 4.2(2)          | Docker EE 3.0 support with Kubernetes | Beginning with this release, Docker EE 3.0 is supported with Kubernetes. |

## Cisco ACI and Docker EE Integration

Docker Enterprise Edition (EE) is a containers as a service (CaaS) platform that enables workload deployment for high availability using the Kubernetes orchestrator. Beginning with Cisco Application Policy Infrastructure Controller (APIC) Release 4.2(1), Cisco Application Centric Infrastructure (ACI) supports integration with Docker EE.

Docker EE includes the Docker Universal Control Plane (UCP), a cluster-management solution that supports the Cisco ACI Container Network Interface (CNI) plug-in. The Cisco ACI CNI plug-in is required for integration with Cisco ACI Docker EE provides technical support.



---

**Note** This document refers to Docker EE 3.0, Docker Enterprise Engine 19.03.x, and Docker UCP 3.2.x on Red Hat 7.6 and 7.7. See the section "Docker Enterprise Edition 3.0" in the Compatibility Matrix article on the Docker website.

---

## System Requirements for Cisco ACI Docker EE Integration

You need at least one manager Kubernetes node and one Kubernetes worker node. This section lists the requirements for the nodes.

- Manager node:
  - CPU: 16 core
  - RAM: 16 GB
  - OS: Red Hat Enterprise Linux Server release 7.6 (Maipo)
- Worker node:
  - CPU: 8 core
  - RAM: 8 GB

- OS: Red Hat Enterprise Linux Server release 7.6 (Maipo)



---

**Note** The use of symmetric policy-based routing (PBR) feature for load-balancing external services requires the use of Cisco Nexus 9300-EX or FX leaf switches.

---

## Hardware Requirements

This section provides the hardware requirements:

- Connecting the servers to Gen1 hardware or Cisco Fabric Extenders (FEXes) is not supported and results in a nonworking cluster.
- The use of symmetric policy-based routing (PBR) feature for load balancing external services requires the use of Cisco Nexus 9300-EX or -FX leaf switches.

For this reason, the Cisco ACI CNI Plug-in is only supported for clusters that are connected to switches of those models.



---

**Note** UCS-B is supported as long as the UCS Fabric Interconnects are connected to Cisco Nexus 9300-EX or -FX leaf switches.

---

## Workflow for Cisco ACI Docker EE Integration

This section provides a high-level description of the tasks that you must perform to integrate Docker Enterprise Edition (EE) into the Cisco Application Centric Infrastructure (ACI) fabric.

- Configure the manager and worker nodes.  
See the section [Configure the Manager and Worker Nodes, on page 4](#).
- Install Docker EE.  
See the section [Install Docker EE, on page 4](#).
- Generate the deployment file for the Cisco ACI Container Network Interface (CNI) plug-in.  
See the section [Generate the Deployment File for Cisco ACI CNI Plug-in, on page 6](#).
- Install the Docker Universal Control Plane (UCP).  
See the section [Install the Docker UCP, on page 6](#).
- Install the Cisco ACI CNI plug-in  
See the section [Install the Cisco ACI CNI Plug-in, on page 7](#).
- Verify the Docker EE installation.  
See the section [Verify the Docker EE Installation, on page 8](#).
- Configure the worker nodes to join the swarm cluster.  
See the section [Configure Worker Nodes to Join the Swarm Cluster, on page 7](#).

- Verify the installation.  
See the section [Verify the Docker EE Installation, on page 8](#).
- Connect to the Docker UCP Dashboard.  
See the section [Connect to the Docker UCP Dashboard, on page 8](#).

## Configure the Manager and Worker Nodes

You must configure the manager and worker nodes before you can integrate Docker Enterprise Edition (EE) with the Cisco Application Centric Infrastructure (ACI) Perform the following steps on all manager and worker nodes.

### Procedure

---

- Step 1** Configure the firewall to allow the ports mentioned in the section "Ports Used," in the article *UCP System requirements*, on the Docker website.
- Step 2** Install the following packages:
- ```
yum -y update
yum install python27-python-pip
yum install -y yum-utils device-mapper-persistent-data lvm2
```
- 

## Install Docker EE

### Procedure

---

- Step 1** Get the Docker Enterprise Edition (EE) image through your Docker subscription.

**Example:**

```
export DOCKERURL=" https://storebits.docker.com/ee/rhel/from-your-subscription"
sudo -E sh -c 'echo "$DOCKERURL/rhel" > /etc/yum/vars/dockerurl'
sudo sh -c 'echo "7" > /etc/yum/vars/dockerosversion'
```

By default, the Docker version is 19.03.4. It is backward compatible.

- Step 2** Disable firewall.

**Example:**

```
systemctl disable firewalld
systemctl stop firewalld
```

- Step 3** Enable the repository.

**Example:**

```
sudo -E yum-config-manager \
  --add-repo \
  "$DOCKERURL/rhel/docker-ee.repo"
```

- Step 4** Install Docker:

**Example:**

```
yum -y install docker-ee docker-ee-cli containerd.io
```

## Step 5 Start Docker:

### Example:

```
systemctl enable docker
systemctl start docker
```

## Step 6 Verify the Docker version and installation on all nodes:

### Example:

```
$ docker version
Client: Docker Engine - Enterprise
Version:      19.03.4
API version:  1.40
Go version:   go1.12.10
Git commit:   9e27c76fe0
Built:        Thu Oct 17 23:25:45 2019
OS/Arch:      linux/amd64
Experimental: false

Server: Docker Engine - Enterprise
Engine:
  Version:      19.03.4
  API version:  1.40 (minimum version 1.12)
  Go version:   go1.12.10
  Git commit:   9e27c76fe0
  Built:        Thu Oct 17 23:24:10 2019
  OS/Arch:      linux/amd64
  Experimental: false
containerd:
  Version:      1.2.10
  GitCommit:    b34a5c8af56e510852c35414db4c1f4fa6172339
runc:
  Version:      1.0.0-rc8+dev
  GitCommit:    3e425f80a8c931f88e6d94a8c831b9d5aa481657
docker-init:
  Version:      0.18.0
  GitCommit:    fec3683

$ docker run hello-world
Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

y) device needs to be vmotioned, force a failover so primary becomes active, then initiate vmotion of primary(active) to minimize traffic loss.

---

## Generate the Deployment File for Cisco ACI CNI Plug-in

You use the `acc_provision` tool to generate the deployment for the Cisco Application Centric Infrastructure (ACI) Container Network Interface (CNI) plug-in.

The tool supports `docker-ucp-3.x` to generate the relevant deployment file for the Docker Enterprise Edition (EE) platform. The version results in the creation of Cisco ACI contracts to allow traffic for the following TCP ports that are required for Docker Universal Control Plane (UCP):

- `ucp-ui-port`: 443
- `ucp-tls-port`: 12376
- `ucp-kubelet-port`: 10250
- `ucp-miscellaneous-ports`: 12378 to 12388

### Before you begin

- Make sure that your manager node is able to reach Cisco Application Policy Infrastructure Controller (APIC).
- Install the `acc-provision` file on the manager node (either from the package or from PyPI).

### Procedure

---

**Step 1** Generate the input file that you need for `acc-provision` from the generated sample file:

**Example:**

```
acc-provision --sample > acc_provision_input.yaml
```

Update the `acc_provision_input.yaml` file as required.

**Step 2** Run `acc-provision` to configure Cisco APIC and generate the Kubernetes deployment file:

**Example:**

```
acc-provision -a -c acc_provision_input.yaml -u apic-user -p apic-password -f docker-ucp-3.0 --debug  
-o aci_deployment.yaml
```

---

## Install the Docker UCP

This procedure installs the latest version of the Docker Universal Control Plane (UCP) on our machine.



**Note** The output of the command in this procedure displays the URL for the Docker UCP browser, administrator username, and password. Record this information for use in the procedure [Verify the Docker EE Installation, on page 8](#).

---

## Procedure

On the manager node, run the following command:

**Note** The following example uses Docker UCP 3.14.

**Example:**

```
docker container run --rm -it --name ucp \
-v /var/run/docker.sock:/var/run/docker.sock \
docker/ucp:3.2.3 install \
--host-address X.X.X.X \
--admin-username xxxx \
--admin-password xxxxxxxx \
--san Y.Y.Y.Y \
--unmanaged-cni \
--interactive
```

The output displays a command for each node to join the cluster, as shown in the following example. Note the information; you will use it to verify the Docker installation after the Cisco Application Centric Infrastructure (ACI) Container Network Interface (CNI) plug-in is installed.

```
docker swarm join --token token IP_address:2377
```

## Install the Cisco ACI CNI Plug-in

After you install the Docker Unified Control Plane (UCP), install the Cisco Application Centric Infrastructure (ACI) Container Network Interface (CNI) plug-in.

## Procedure

- 
- Step 1** Install the kubectl command-line interface on the manager node by using yum or by following Docker documentation. See the document *Install the Kubernetes CLI* on the Docker website.
  - Step 2** Run the following command on the manager node using the `aci_deployment.yaml` file that you generated earlier:  

```
kubectl apply -f aci_deployment.yaml
```
  - Step 3** Verify the status of pods `aci-*` and `kube-dns-*`:

**Example:**

```
$ kubectl get pods -n kube-system
aci-containers-controller-74b57ccf68-jm656    1/1      Running    0          5d
aci-containers-host-86j5b                    3/3      Running    0          5d
aci-containers-openvswitch-bzxwc            1/1      Running    0          5d
compose-64cf857c47-nbrfv                     1/1      Running    0          5d
compose-api-7cc55fdb6-k7wp6                 1/1      Running    0          5d
kube-dns-6d79cfd8c-97rfh                    3/3      Running    0          5d
ucp-metrics-76qpl                            3/3      Running    0          5d
```

Wait until all the pods are up before proceeding.

---

## Configure Worker Nodes to Join the Swarm Cluster

After you complete installation, you configure the worker nodes to join the storm cluster.

## Before you begin

You have noted the command generated in the procedure [Install the Docker UCP, on page 6](#) in this guide, or you can log in to the master node and regenerate it using the following command:

```
docker swarm join-token worker
```

## Procedure

Log in to your worker nodes and on each one run the following command:

```
docker swarm join --token token IP_address:2377
```

## Verify the Docker EE Installation

After installing Docker Enterprise Edition (EE), verify that the Docker EE is functional and ready to use. Use the administrator username and password that you captured in the procedure "Install Docker UCP" to log in to UCP. There you can see your swarm cluster and details of the Kubernetes pods, nodes, and Cisco Application Centric Infrastructure (ACI) Container Network Interface (CNI) plug-in.

## Before you begin

You must have captured the URL for the Universal Control Plane (UCP) browser, the administrator username, and password from the procedure [Install the Docker UCP, on page 6](#).

## Procedure

---

**Step 1** Verify the swarm cluster, making sure that the state is `Running` and that the status is `Active`.

### Example:

```
docker node ls
```

| ID                          | HOSTNAME               | STATUS | AVAILABILITY | MANAGER STATUS | ENGINE VERSION |
|-----------------------------|------------------------|--------|--------------|----------------|----------------|
| qfma9oxt8emf16qlmn19hirul * | kube-ee4               | Ready  | Active       | Leader         | 19.03.4        |
| xf6kp099v6odswtts0lo7f8c9   | kube-ee5               | Ready  | Active       |                | 19.03.4        |
| ntyje82r9aouic3v6vh77cvuc   | kube-ee6.sys.cisco.com | Ready  | Active       |                | 19.03.4        |

**Step 2** Use the `kubectl` command-line interface to get detailed information for all the pods running on your system.

### Example:

```
kubectl get pods --all-namespaces
```

| NAMESPACE   | NAME                                       | READY | STATUS  | RESTARTS | AGE |
|-------------|--------------------------------------------|-------|---------|----------|-----|
| kube-system | aci-containers-controller-79f9476556-2bwv2 | 2/2   | Running | 2        | 3d  |
| kube-system | aci-containers-host-6fxrs                  | 4/4   | Running | 4        | 3d  |
| kube-system | aci-containers-openvswitch-58dt6           | 1/1   | Running | 1        | 3d  |
| kube-system | compose-64cf857c47-rbgmt                   | 1/1   | Running | 1        | 3d  |
| kube-system | compose-api-6bddf48f67-krq55               | 1/1   | Running | 2        | 3d  |
| kube-system | kube-dns-6d79cfd8c-rc7df                   | 3/3   | Running | 3        | 3d  |

---

## Connect to the Docker UCP Dashboard

After you install Docker Enterprise Edition (EE), you connect to the Docker Universal Control Plane (UCP) dashboard.



## Procedure

---

**Step 1** Open a web browser and point to the host IP address that was provided during Docker UCP installation in the section [Install the Docker UCP, on page 6](#).

**Example:**

```
https://host-address
```

**Step 2** Accept the exception and continue.  
The Docker Enterprise login page appears.

**Step 3** Log in with the credentials that you provided when you installed the Docker UCP.

**Step 4** Upload your license or skip uploading it for now.  
The Docker UCP dashboard appears. You can browse the namespaces, pods, and other Kubernetes resources from the left navigation panel.

---

## Deploying Applications

After you integrate Docker Enterprise Edition (EE) with Cisco Application Centric Infrastructure (ACI), you can deploy applications.

This document includes procedures for deploying two popular applications. You do not need to deploy the applications: They are examples and can help you validate your cluster.

### Deploy NGINX

Use the kubectl CLI to apply the NGINX .yaml file, which contains the NGINX specification. Then verify the deployment using kubectl and the Docker Universal Control Plane (UCP) dashboard.

The following is an example of an NGINX deployment .yaml file:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.7.9
          ports:
            - containerPort: 80
```

## Procedure

---

**Step 1** Apply the `.yaml` file to deploy NGINX.

**Example:**

```
kubectl create -f nginx_deploy.yaml
```

**Step 2** Verify the deployment using `kubectl`.

**Example:**

```
$ kubectl get deployment
```

| NAME             | DESIRED | CURRENT | UP-TO-DATE | AVAILABLE | AGE |
|------------------|---------|---------|------------|-----------|-----|
| nginx-deployment | 3       | 3       | 3          | 3         | 3d  |

**Step 3** Verify the deployment pm the Docker UCP dashboard.

a) Log in to the Docker UCP dashboard.

b) In the left navigation pane, click **Pods**.

The central pane displays information about each pod: Its state, status, name, namespace, node, and date and time of creation.

---

## Deploy Guestbook

Complete this procedure to deploy the guestbook application.

### Before you begin

- Make sure that the final "frontend" service is up.
- Download the deployment files from the following location:

```
https://kubernetes.io/docs/tutorials/stateless-application/guestbook/
```

### Procedure

---

**Step 1** Run the `kubectl get svc` command and note the external IP address from the output.

**Step 2** Use a browser to reach the dashboard of the guestbook service.

**Step 3** Log in to the Docker Universal Control Plane (UCP) dashboard.

**Step 4** Go to **Kubernetes > Load Balancers > frontend**, and in the central pane verify the deployed service.

**Step 5** Go to **Kubernetes > Pods**, and in the central pane, and verify the pods.

Look for pods with names that begin with **frontend-**.

---





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

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

**Europe Headquarters**  
CiscoSystemsInternationalBV  
Amsterdam,TheNetherlands

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