White Paper

Cisco UCS with Docker Enterprise Edition

Contents

Introduction

Docker has become an efficient platform for developers to use to build, ship, and run distributed applications anywhere. With microservices architecture shaping the next generation of applications, enterprises with large investments in monolithic applications are finding ways to adopt Docker containers to stay competitive. Containerization provides the agility, control, and portability that developers and operations require to build and deploy applications across any infrastructure.

Docker allows distributed applications to be composed easily into lightweight application containers that can change dynamically and without disruption. Containers make the applications portable across development, test, and production environments running on physical and virtual machines locally, in data centers, and across clouds from different cloud service providers. Docker Enterprise Edition (DEE) provides native container management tools, including Docker Engine (DE), Docker Trusted Registry (DTR), and Docker Universal Control Plane (UCP). It can be deployed on premises or in a virtual private cloud (vPC) and connected to the existing infrastructure and systems, such as storage, Microsoft Active Directory (AD), and Lightweight Directory Access Protocol (LDAP) services.

Cisco Unified Computing System™ (Cisco UCS®) servers adapt to meet rapidly changing business needs, including just-in-time deployment of new computing resources to meet requirements and improve business outcomes. The combination of Docker container technology and Cisco UCS server hardware supports highly scalable, resilient, elastic application deployment with the simplicity of the cloud and a full set of enterprise capabilities.

Docker and Cisco have developed a Cisco® Validated Design for Docker Datacenter on Cisco UCS. The validated design presents a defined process for provisioning and configuring the solution. The solution is tested in a lab setting to measure performance, scalability, availability, and failure using workloads that simulate actual production deployments. As a result, you can achieve faster, more reliable, and more predictable implementations.

This document demonstrates the benefits of using Cisco UCS with Docker Enterprise Edition to deploy, scale, and manage a production-ready application container environment. Cisco UCS is fully programmable, and it provides automated infrastructure lifecycle management. This document also describes how to deploy and run an application container using Docker Enterprise Edition components so that you can deploy Docker containerized applications in your production environment with confidence. It also explains the unique and advanced capabilities of the solution to orchestrate the application container lifecycle using Docker Enterprise Edition on Cisco UCS infrastructure.

Reference Architecture

This section provides an overview of the Cisco UCS and Docker Enterprise Edition architectures.

Cisco UCS Programmable Infrastructure

Cisco UCS is infrastructure as code (IaC). It was designed with four main technology innovations, so you can more easily define the desired state of the infrastructure and what you want to do with it. These four innovations provide the foundation for automated infrastructure management by making the infrastructure programmable:

- **Software object model:** Hardware is not configured manually in Cisco UCS. Instead, every identity and configuration setting for every device in the system is defined in software through policies and service profiles. This data model helps ensure that configurations are consistent and allows simple implementation of changes at scale.

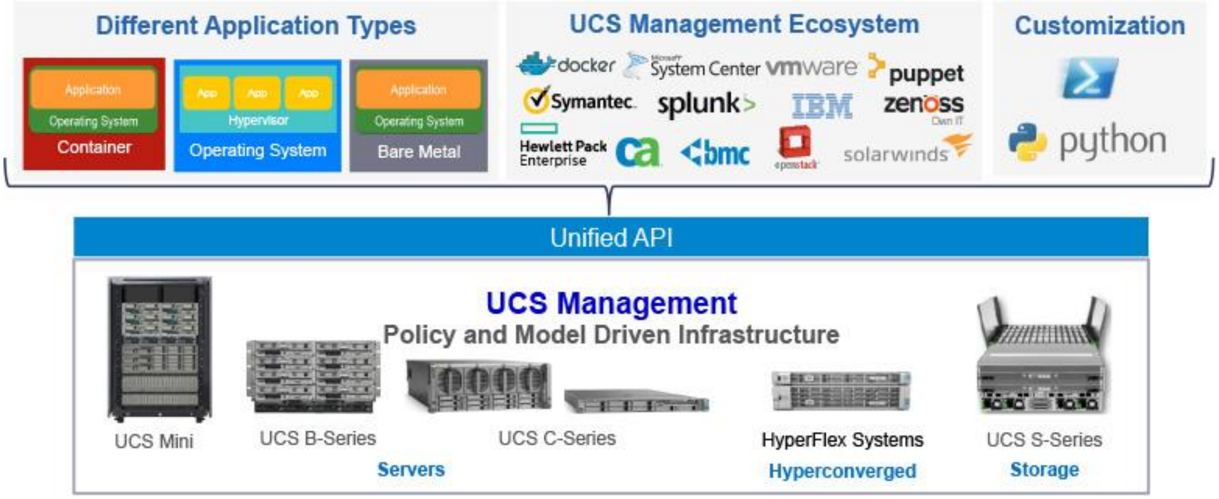
- **API-centric approach:** The unified system control plane is accessible through a fully documented and open API.
- **Virtual interface card (VIC):** All network and SAN adapters are software defined, but they present themselves to the bare-metal OS or hypervisor as physical devices.
- **Service profiles and templates:** The infrastructure policies needed to deploy applications are encapsulated in service profiles templates, which are collections of policies needed for the specific applications. The service profile templates are then used to create one or more service profiles, which provide the complete definition of the server, storage, and fabric. Service profiles and service profile templates also help eliminate configuration drift and ensure a standardized environment for the applications.

The software object model and unified API in the Cisco UCS management framework work in conjunction with the Cisco® fabric interconnects and the VICs to facilitate IaC. As a result of this programmatic capability, Cisco UCS simplifies and accelerates application and service deployment in bare-metal, virtualized, and containerized environments. Unified, model-based management, end-to-end provisioning, and migration support further ease deployment and enhance reliability and security.

Cisco UCS Manager automates the provisioning, configuration, and monitoring of the infrastructure. It includes a unified API that serves as a unified control plane for integration with Puppet and a wide range of independent software vendor (ISV) configuration, orchestration, and monitoring tools. You can also use Cisco UCS PowerTool for Microsoft Windows PowerShell and a Python software development kit (SDK) for customization and further integration.

Figure 1 provides an overview of the Cisco UCS ecosystem.

Figure 1. Cisco UCS Ecosystem: High-Level View



For more information about Cisco UCS, see Cisco Unified Computing System.

Docker Enterprise Edition

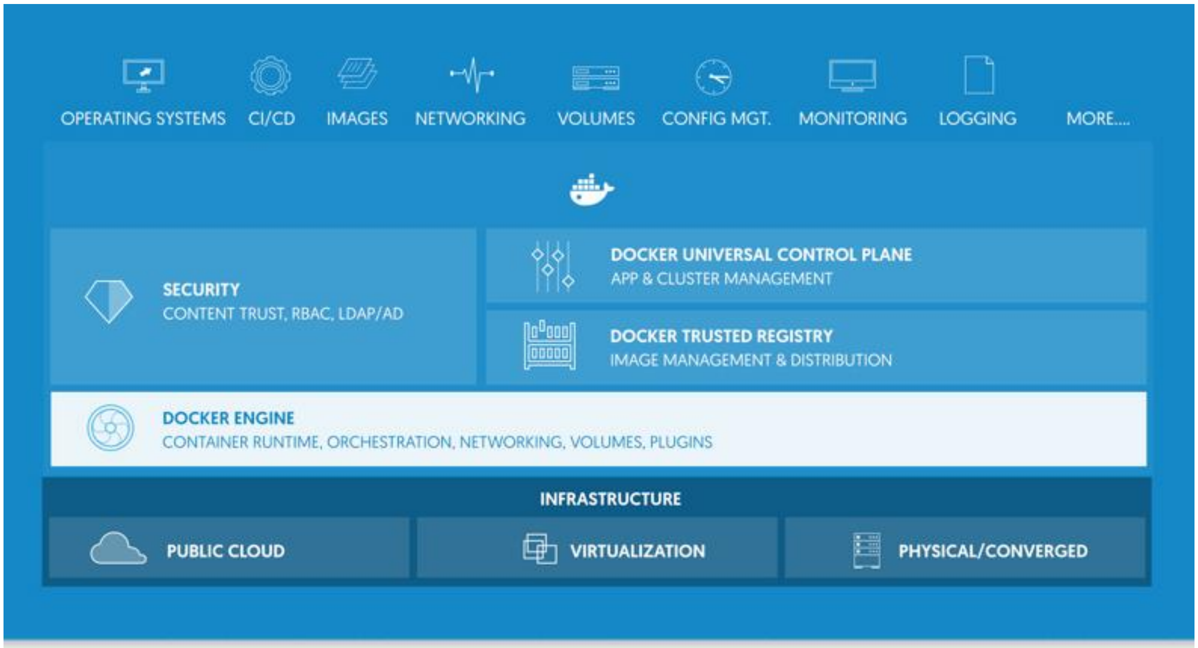
Docker containers encapsulate all application components, such as dependencies and services. When all dependencies are encapsulated, applications become portable and can be dependably moved among development, test, and production environments. Docker makes container creation and management simple and integrates it with many open-source projects.

Docker Enterprise Edition includes leading Docker open-source projects, commercial software, and integration with validated and supported configurations:

- Docker Universal Control Plane, or UCP, provides an embedded Swarm scheduler for integrated management and orchestration of the Docker environment.
- Docker Trusted Registry, or DTR, supports Docker image management, security, and collaboration.
- Docker Content Trust Security provides a multilayered approach to security, with the capability to sign images with digital keys and then verify the signature of those images.
- Docker Enterprise Edition, or Docker EE, provides for a robust container runtime environment.

Figure 2 provides an overview of the Docker architecture.

Figure 2. Docker Architecture: High-Level View



Solution Components

This section introduces the Cisco and Docker components used in the solution.

Cisco UCS Manager

Cisco UCS Manager provides unified, embedded management for all software and hardware components in Cisco UCS servers. It manages, controls, and administers multiple chassis for thousands of virtual machines as a single logical entity through an intuitive GUI, a command-line interface (CLI), or an XML API. Cisco UCS Manager resides on a pair of Cisco UCS fabric interconnects using a clustered, active-standby configuration for high availability. It provides an embedded management interface that integrates server, network, and storage resources. The manager performs autodiscovery to detect, manage, and provision system components that are added or changed. The API exposes 9000 points of integration and facilitates integration with third-party operations management tools and custom development for automation, orchestration, and monitoring.

Service profiles benefit both virtualized and nonvirtualized environments and increase the mobility of nonvirtualized servers: for instance, when moving workloads from server to server or taking a server offline for maintenance or upgrade. Profiles can also be used in conjunction with virtualization clusters to bring new resources online easily, complementing existing virtual machine mobility.

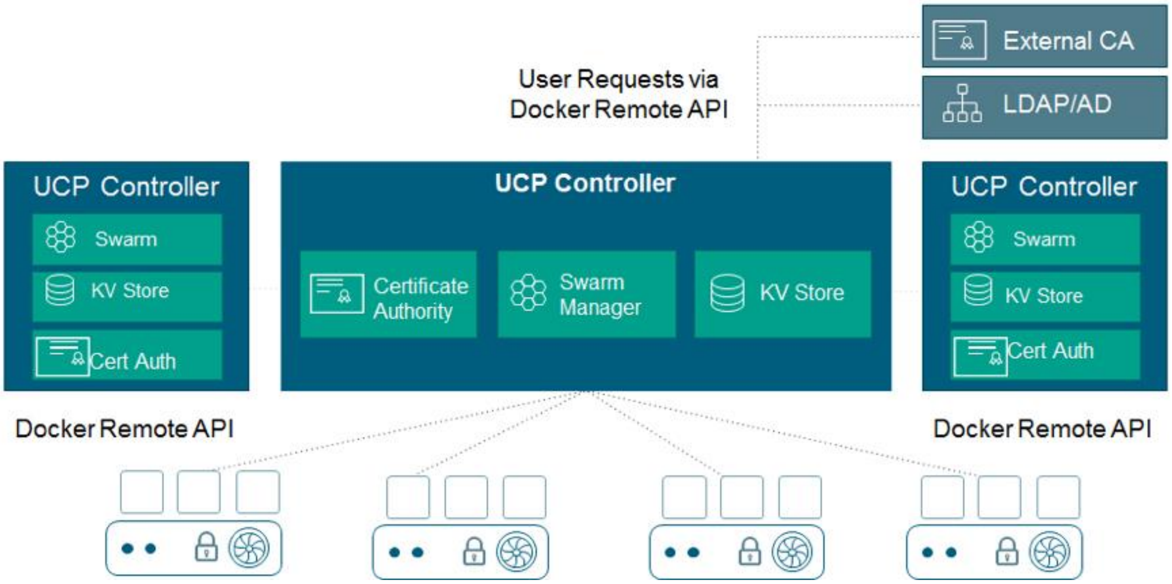
For more information about Cisco UCS Manager, see http://www.cisco.com/c/en/us/products/servers-unified-computing/ucs-manager/index.html.

Docker Universal Control Plane

Docker UCP is an on-premises enterprise solution that includes user management, resource management, clustering, and orchestration capabilities. It integrates with existing enterprise LDAP and Active Directory services for high availability, security, and compliance. UCP enables IT operations teams to deploy and manage containerized applications in production environments.

Figure 3 shows the Docker UCP architecture and illustrates the built-in high availability of UCP.

Figure 3. Docker UCP Architecture



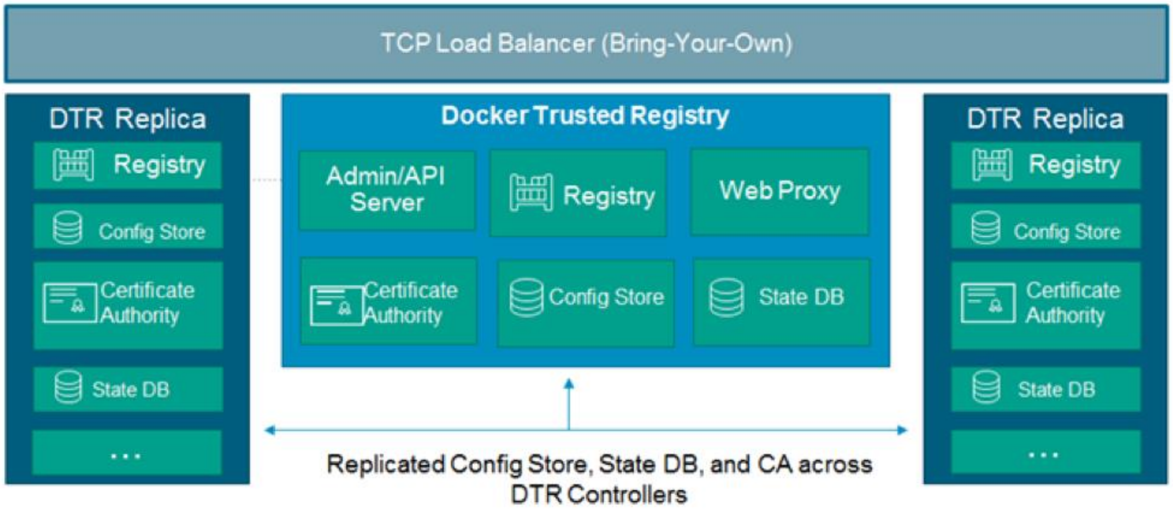
Docker Trusted Registry

DTR is the enterprise-class image storage solution from Docker. DTR gives enterprises the security and compliance they need to store and manage their Docker images on premises or in their VPC. It has a built-in authentication mechanism, supports role-based access control (RBAC), and can integrate with LDAP and Active Directory.

DTR is part of the Docker Enterprise Edition Subscription, which also includes Docker UPC, a commercially supported engine (Docker CS engine), and support. DTR is easy to deploy, configure, and integrate with your existing infrastructure and application delivery workflows.

Figure 4 shows the built-in high availability of DTR in the control plane. High availability helps ensure continuous availability for DTR. If the main instance fails, a replica instance takes over nondisruptively.

Figure 4. Docker Trusted Registry Architecture



Solution Components

Table 1 lists the Cisco UCS infrastructure components used in this solution.

Table 1. Solution Components

Component	Model	Quantity	Comments
UCP controller, UCP node, and DTR node	Cisco UCS B200 M4 B-Series Server and Cisco UCS C220 M4 Server	8 B-Series or 4 C-Series Servers	• **CPU:** 2 x Intel Xeon processor E5-2630 v3 • **Memory:** 8 x 16-GB 2133 DIMMs, for a total of 128 GB • **Local disks:** 2 x 300-GB SAS disks for OS boot and Docker volume • **Network card:** 1 x Cisco UCS VIC 1340 • **RAID controller:** Cisco MegaRAID 12-GB SAS controller (Or) • **CPU:** 2 x E5-2669 v4 • **Memory:** 16 x 16GB 2133 DIMMs, for a total of 256GB • **Local disks:** 6 x 1.2 TB SAS disks for OS Boot & Docker Engine • **Network card:** 1x1227 VIC • **RAID controller:** Cisco MegaRAID SAS Controller
Chassis	Cisco UCS 5108 Blade Server Chassis	1	
I/O module	Cisco UCS 2208XP Fabric Extender	2	
Fabric interconnect	Cisco UCS 6248UP 48-Port Fabric Interconnect	2	
Switch	Cisco Nexus® 9372PX Switch	2	

Physical Topology

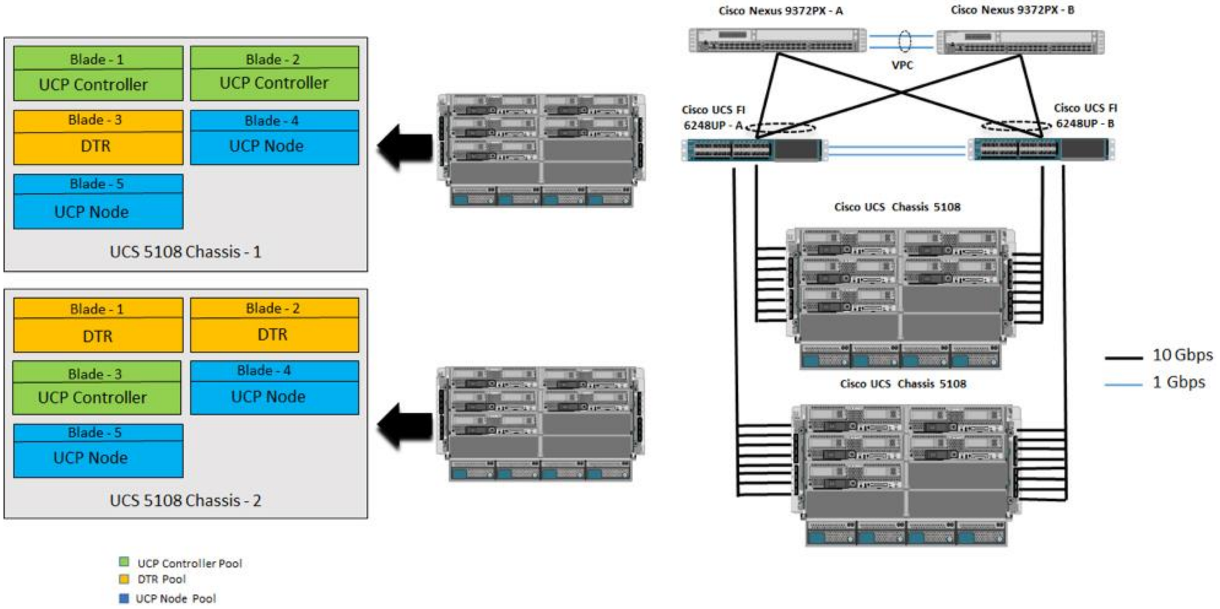
Figure 5 shows the topology with Cisco Nexus 9372PX top-of-rack (ToR) switches, Cisco UCS 6248UP fabric interconnects, and Cisco UCS B200 M4 servers forming the Cisco UCS infrastructure for Docker Enterprise Edition. Each Cisco UCS blade server in the chassis has a different function, as shown in the figure.

A Docker UCP cluster consists of three types of nodes:

- UCP nodes are primarily for running containers.
- UCP controller nodes are for running containerized UCP services. The UCP controller node acts as the primary node for the UCP cluster.
- UCP replica nodes are other UCP services cluster nodes that work in high-availability mode. They are ready to take over the UCP controller primary role in the event of controller node failure.

The UCP controller and replica nodes persistently manage the cluster and cluster configurations. All UCP services nodes can handle application container workloads and can be configured to run only UCP services such as scheduler and orchestration tasks. UCP nodes act as computing nodes and handle application container workloads. The DTR node provides DTR services offered by the application containers running on Docker UCP.

Figure 5. Topology: Cisco UCS Infrastructure for Docker Enterprise Edition with B-Series Servers



Note: Instead of blade chassis, we have an alternate topology with Cisco UCS C-Series servers as shown in the topology below. For further details, see: http://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/UCS_CVDs/ucs_docker.html

Figure 6. Alternate Topology: Cisco UCS Infrastructure for Docker Enterprise Edition with C-Series Servers

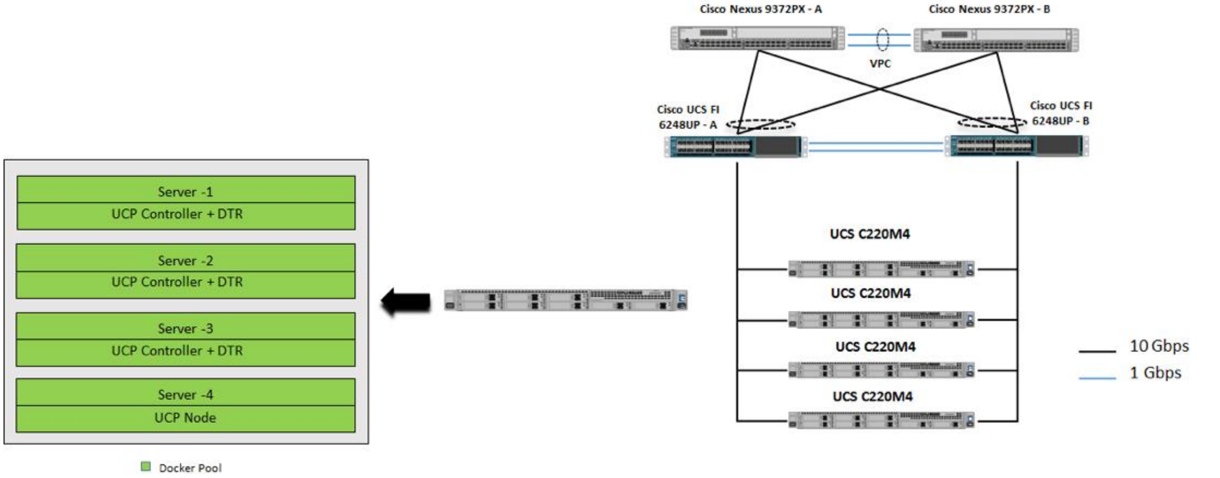


Table 2 lists the hardware and software versions used in this solution.

Table 2. Hardware and Software Versions

Layer	Device	Image	Comments
Computing	Cisco UCS B200 M4 server or C220 M4 servers	Version 3.1 (1e)	Cisco UCS server
Network adapter	Cisco UCS VIC 1340or Cisco UCS VIC 1227	Version 3.1 (1e)	Cisco VIC
Network	Cisco UCS 6248UP fabric interconnect	Version 3.1 (1e)	Cisco UCS fabric interconnect
	Cisco Nexus 9372PX Switch	Version 7.0(3)I1(3)	Cisco Nexus 9000 Series ToR switch
Cisco software	Cisco UCS Manager	Version 3.1 (1e)	Cisco UCS Manager
Docker Enterprise Edition	Docker CS engine	Version 1.12.1-cs1	Docker commercially supported engine
	Docker Swarm	Version 1.2.5	Docker Swarm scheduler is embedded in UCP
	Docker UCP	Version 1.1.3	Docker environment orchestrator and management interface
	Docker DTR	Version 2.0.3	Docker image store for Docker EE
Operating system	Red Hat Enterprise Linux	Version 7.2	Red Hat Linux for bare-metal OS

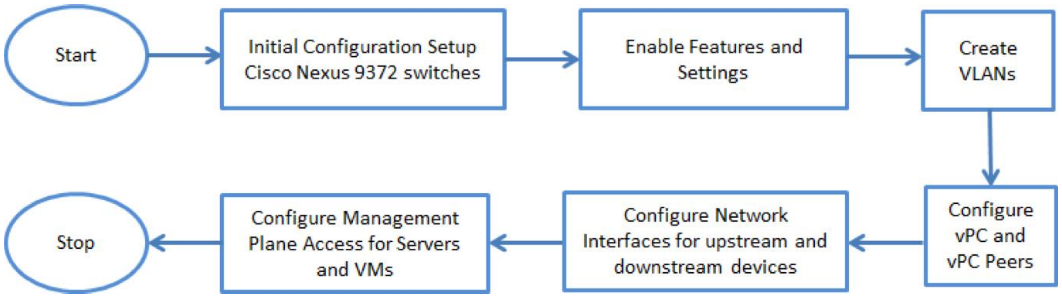
Installation Workflow

The solution as a whole consists of Cisco UCS blade servers, Cisco UCS fabric interconnects, and Cisco Nexus ToR switches as hardware components. The Docker Enterprise Edition and Cisco UCS Manager are part of the software components of the solution. The Docker software stack runs on the Red Hat Enterprise Linux OS. Cisco UCS servers provide a converged and highly available hardware platform centrally managed by Cisco UCS Manager software residing on Cisco UCS fabric interconnects. An important component of the Docker Enterprise Edition product, the UCP provides the redundancy and high availability of the Docker CS engine and management interface. This solution holistically offers a container-as-a-service model that supports deployment of diverse application environments in DevOps and production use cases.

Cisco Nexus 9372PX Configuration

Figure 7 shows the configuration process for the Cisco Nexus 9372PX Switches.

Figure 7. Cisco Nexus 9372PX Configuration Process

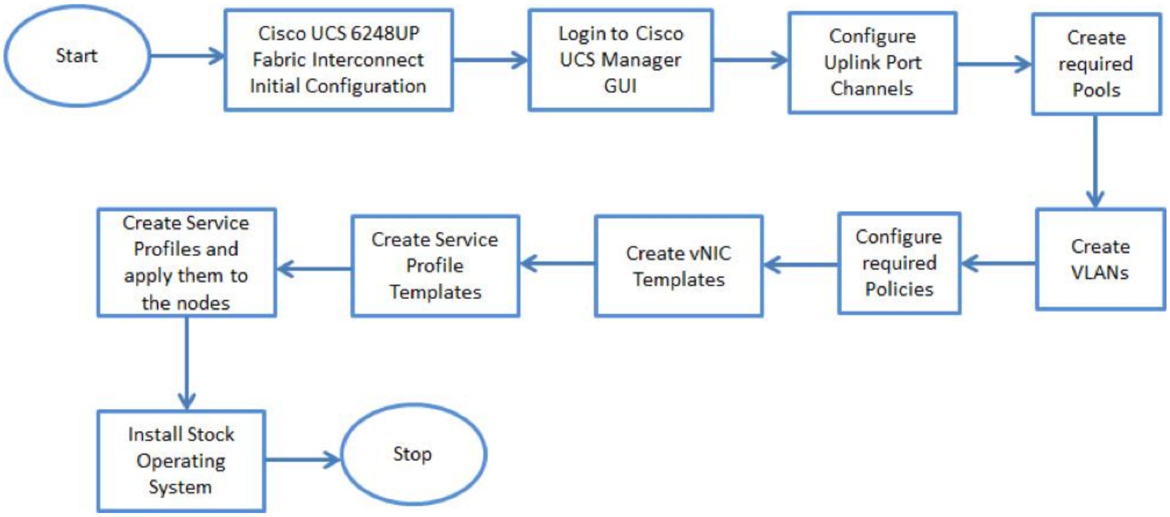

Start
Initial Configuration Setup Cisco Nexus 9372 switches
Enable Features and Settings
Create VLANs
Configure vPC and vPC Peers
Configure Network Interfaces for upstream and downstream devices
Configure Management Plane Access for Servers and VMs
Stop

Cisco Nexus 9372PX Switches include ToR or middle-of-row (MoR) fiber-based server connectivity suitable for deployment in small business, enterprise, and service provider environments. Enhanced Cisco NX-OS Software is designed to provide a robust feature set, including performance, resiliency, scalability, manageability, and programmability features. In this solution, a pair of Cisco Nexus 9372PX upstream switches is deployed to provide northbound network connectivity for the application containers. Furthermore, these switches provide redundancy for the application container data path, using a Cisco virtual port-channel (vPC) configuration between the Cisco Nexus 9372PX Switches and Cisco UCS 6248UP fabric interconnects.

Cisco UCS Manager Configuration

Figure 8 shows the configuration process for Cisco UCS Manager.

Figure 8. Cisco UCS Manager Configuration Process


Start
Cisco UCS 6248UP Fabric Interconnect Initial Configuration
Login to Cisco UCS Manager GUI
Configure Uplink Port Channels
Create required Pools
Create VLANs
Configure required Policies
Create vNIC Templates
Create Service Profile Templates
Create Service Profiles and apply them to the nodes
Install Stock Operating System
Stop

Cisco UCS management software delivers policy-based automation and role-based access capabilities to help you effectively administer and manage data center infrastructure at scale. This tightly integrated management solution uses a DevOps-friendly architecture that treats infrastructure as lines of code, with every component programmable. It provides a model-based foundation to simplify the day-to-day processes of provisioning, monitoring, and managing computing and local storage resources and storage and network connections. These novel features coupled with an open XML API facilitates integration with third-party infrastructure automation tools such as Ansible, Puppet, and Chef for day-zero provisioning, management, and monitoring tasks.

The Cisco UCS converged infrastructure model emphasizes the use logical servers rather than traditional physical servers (blade or rack). Service profiles, which are software definitions of servers, shift Cisco UCS from conventional servers to logical servers. The service profile contains all the server hardware identifier, firmware, state, configuration, connectivity, and behavior information, but is totally abstracted from the physical server.

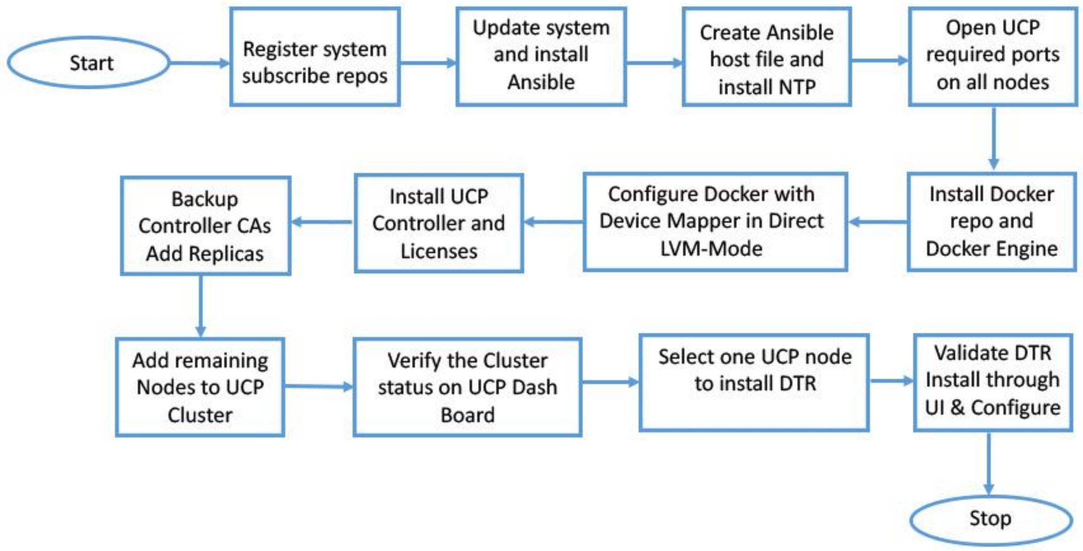
Service profiles form the foundation for the stateless, utility computing model in Cisco UCS. Virtualization of MAC addresses and World Wide Port Name (WWPN) identifiers has been evolving in the industry for years, but Cisco UCS extends the logical service profile definition to include the hardware, BIOS, CPU, and I/O adapter configuration; versions; and settings through policies, pools, isolation, and templates.

One of the main benefits of application containers is software portability. Cisco UCS infrastructure extends the concept of portability to include the operating system and underlying hardware in addition to the deployed applications. Stateless computing, a distinguishing feature of Cisco UCS servers, provides the capability to move workloads from one sever to another in the event of hardware failure with little downtime.

Docker Enterprise Edition Deployment

Figure 9 shows the Docker Enterprise Edition deployment process.

Figure 9. Docker Enterprise Edition Deployment Process


Start
Register system subscribe repos
Update system and install Ansible
Create Ansible host file and install NTP
Open UCP required ports on all nodes
Install Docker repo and Docker Engine
Configure Docker with Device Mapper in Direct LVM-Mode
Install UCP Controller and Licenses
Backup Controller CAs Add Replicas
Add remaining Nodes to UCP Cluster
Verify the Cluster status on UCP Dash Board
Select one UCP node to install DTR
Validate DTR Install through UI & Configure
Stop

Host setup day-zero automation tasks are run on all the participating server nodes in the cluster using the Ansible automation tool. This tool does not use a client-server model and does not need an additional build server. Any node can be initiated to run some limited host setup tasks even before the Docker Enterprise Edition is deployed on the nodes.

Validation

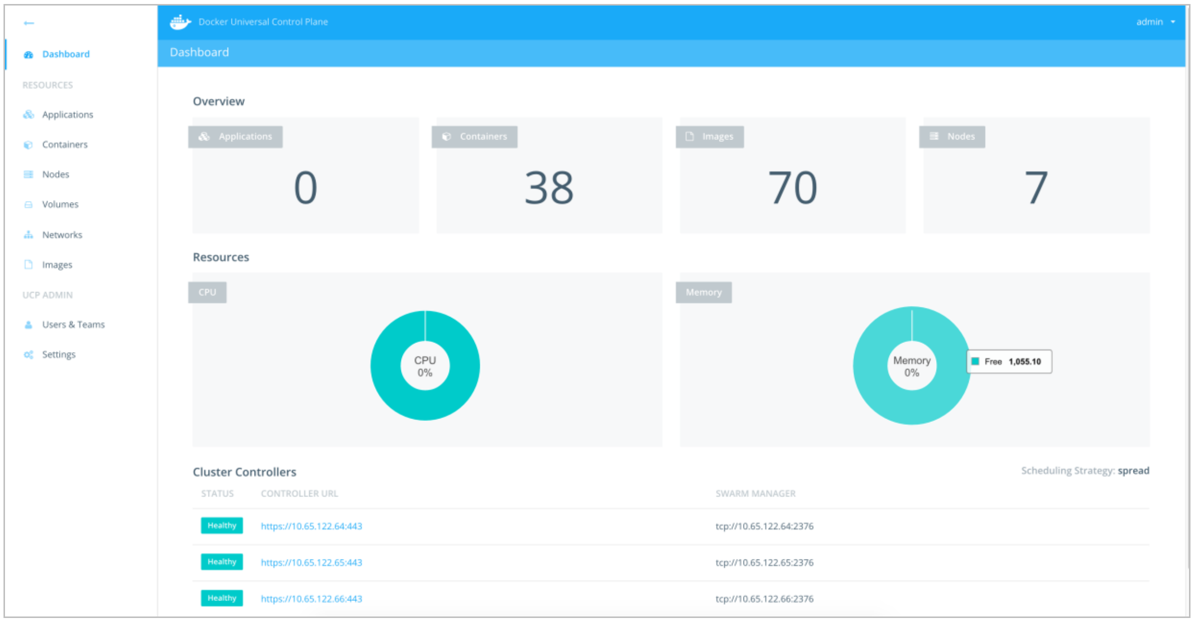
After the stack is up and running, you can validate the stack by installing a sample WordPress two-tier application on the stack.

The WordPress application is an open-source blogging tool and content-management system based on PHP and MySQL. The application runs on a web hosting service. It has two components: a front-end web interface and a back-end MySQL or MariaDB database. Both components run as application containers on Docker UCP nodes as scheduled by the UCP services nodes. This scheduling is performed through either the UCP dashboard user interface or the UCP CLI.

Figures 10 through 13 show the various stages of containerized application deployment using the Docker Enterprise Edition platform.

- **Cluster status:** Docker UCP shows all the controllers with their status. The dashboard in Figure 10 also shows the number of containers spawned and the number of nodes used in this solution.

Figure 10. Cluster Status



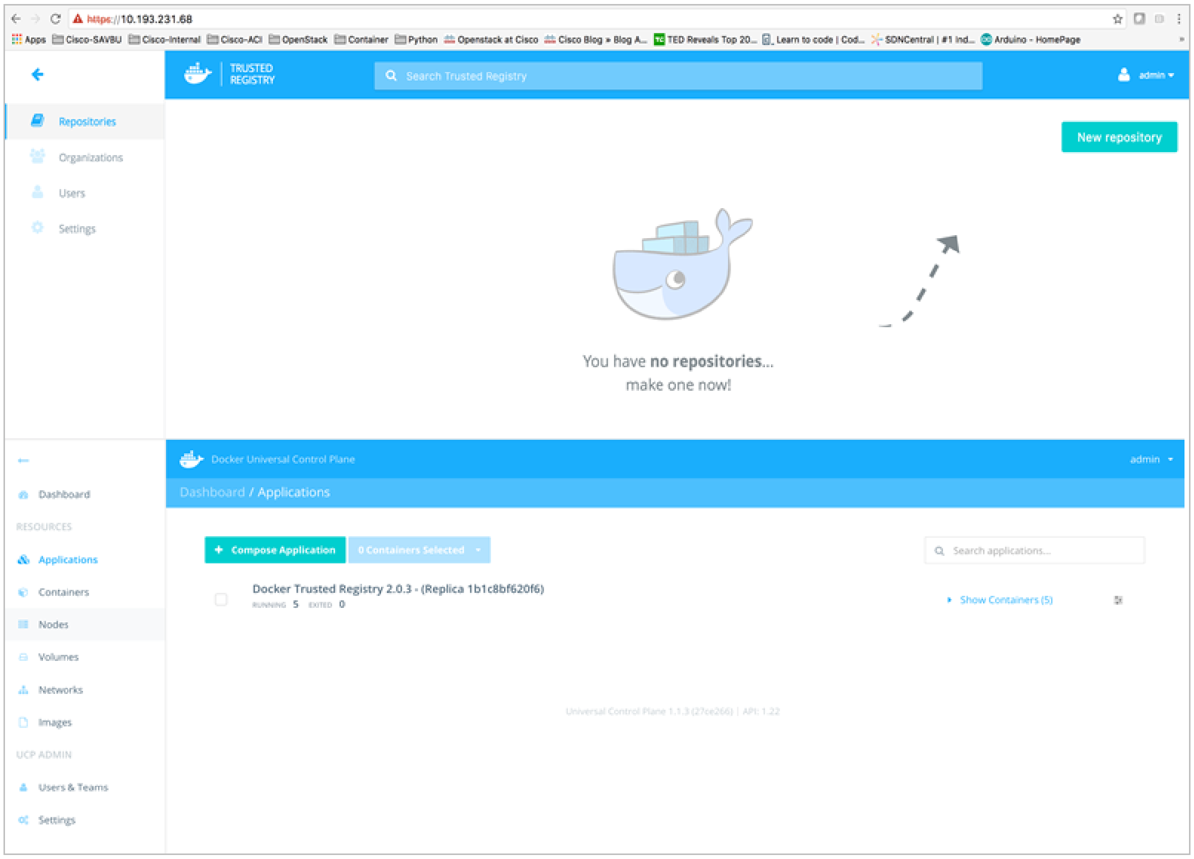
Cluster Controllers

Scheduling Strategy: spread

STATUS	CONTROLLER URL	SWARM MANAGER
Healthy	https://10.65.122.64:443	tcp://10.65.122.64:2376
Healthy	https://10.65.122.65:443	tcp://10.65.122.65:2376
Healthy	https://10.65.122.66:443	tcp://10.65.122.66:2376

- **DTR status:** The dashboard in Figure 11 shows the configured DTR and the number of instances running. DTR is responsible for Docker image management, security, and collaboration.

Figure 11. Docker Trusted Registry Status



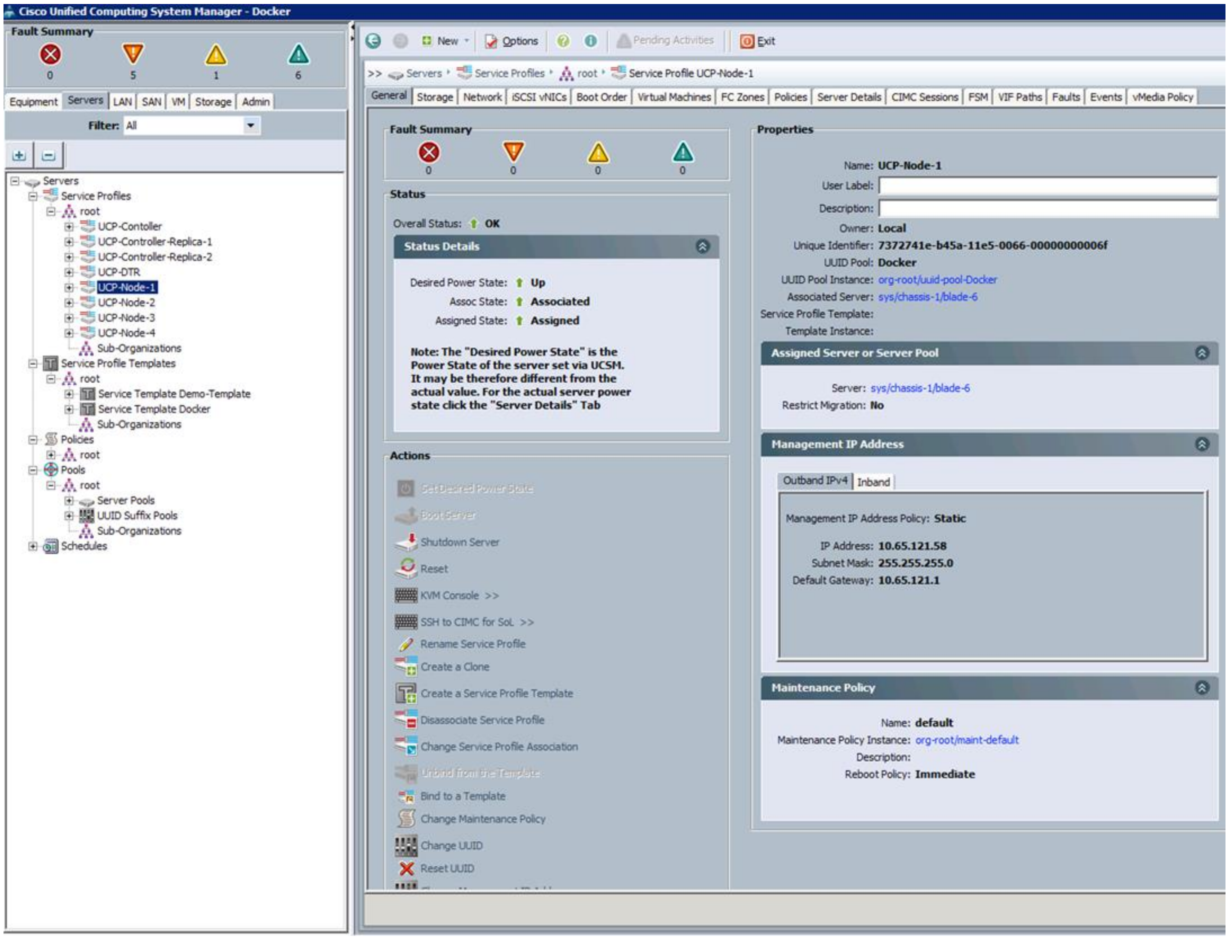
You have **no repositories**...
make one now!

Docker Trusted Registry 2.0.3 - (Replica 1b1c8bf620f6)
RUNNING 5 EXITED 0

Universal Control Plane 1.1.3 (27ce266) | API: 1.22

- **Cisco UCS Manager:** The Cisco UCS Manager GUI shows eight service profiles configured for Cisco UCS B200 M4 servers housed in the Cisco UCS 5108 chassis. The status details in Figure 12 screenshot show the assignment and association status of the blade server with the service profile.

Figure 12. Cisco UCS Manager Details



Status

Overall Status: OK

Status Details

Desired Power State: Up
Assoc State: Associated
Assigned State: Assigned

Note: The "Desired Power State" is the Power State of the server set via UCSM. It may be therefore different from the actual value. For the actual server power state click the "Server Details" Tab

Properties

Name: UCP-Node-1
User Label:
Description:
Owner: Local
Unique Identifier: 7372741e-b45a-11e5-0066-00000000006f
UUID Pool: Docker
UUID Pool Instance: org-root/uuid-pool-Docker
Associated Server: sys/chassis-1/blade-6
Service Profile Template:
Template Instance:

Assigned Server or Server Pool

Server: sys/chassis-1/blade-6
Restrict Migration: No

Management IP Address

Outband IPv4 | Inband

Management IP Address Policy: Static

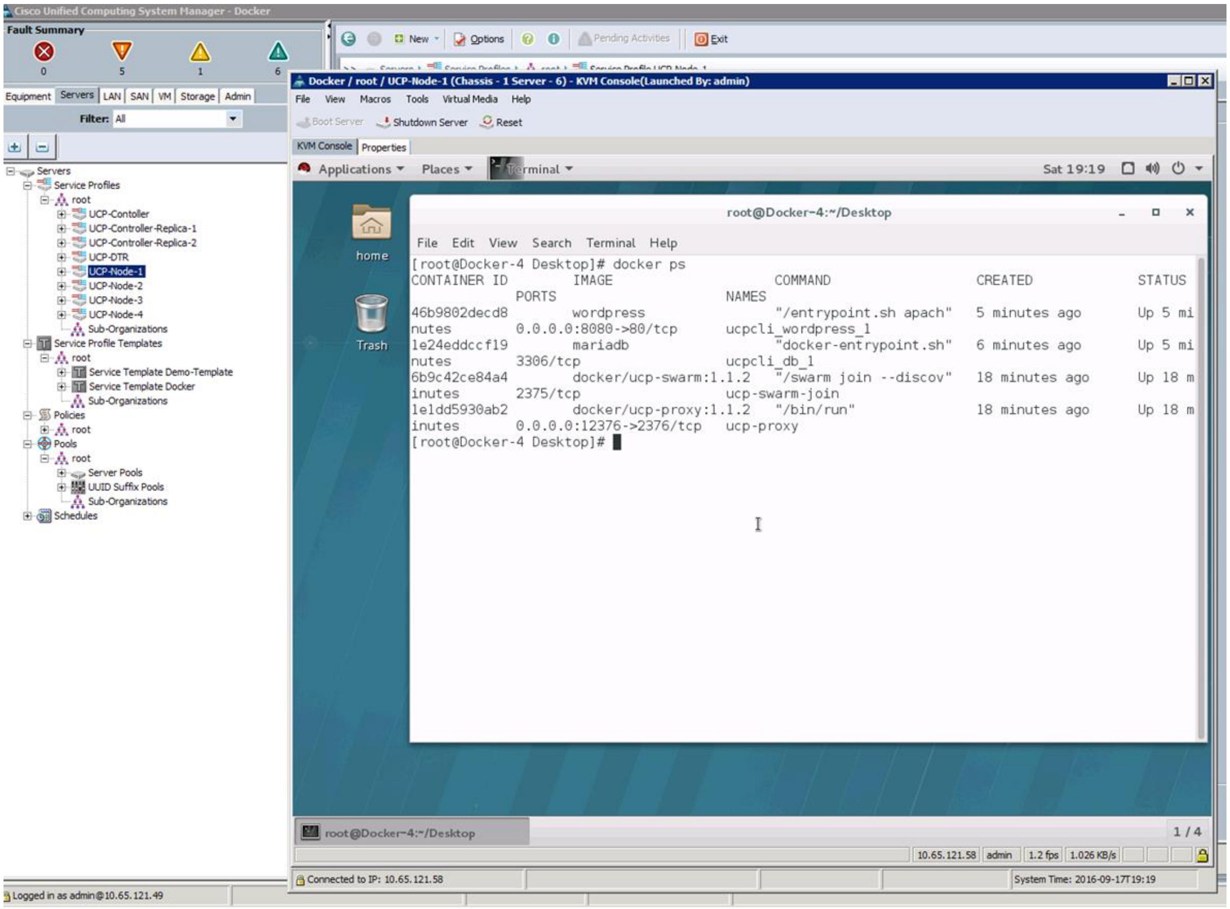
IP Address: 10.65.121.58
Subnet Mask: 255.255.255.0
Default Gateway: 10.65.121.1

Maintenance Policy

Name: default
Maintenance Policy Instance: org-root/maint-default
Description:
Reboot Policy: Immediate

- **Application container:** In this solution, the WordPress application is deployed on the Cisco UCS blade servers. Figure 13 shows the successful deployment of the WordPress application using Docker Enterprise Edition on a Cisco UCS blade server. You can see the status of the containerized application on the Docker Enterprise Edition platform by entering the **docker ps** command on all the configured server nodes.

Figure 13. Application Container Details



```
[root@Docker-4 Desktop]# docker ps
CONTAINER ID        IMAGE                    COMMAND                  CREATED             STATUS
             PORTS                     NAMES
46b9802decd8        wordpress                "/entrypoint.sh apach"   5 minutes ago       Up 5 mi
nutes        0.0.0.0:8080->80/tcp      ucpcli_wordpress_1
1e24eddccf19        mariadb                  "docker-entrypoint.sh"   6 minutes ago       Up 5 mi
nutes        3306/tcp                  ucpcli_db_1
6b9c42ce84a4        docker/ucp-swarm:1.1.2   "/swarm join --discov"   18 minutes ago      Up 18 m
inutes       2375/tcp                  ucp-swarm-join
1e1dd5930ab2        docker/ucp-proxy:1.1.2   "/bin/run"               18 minutes ago      Up 18 m
inutes       0.0.0.0:12376->2376/tcp   ucp-proxy
[root@Docker-4 Desktop]#
```

Conclusion

The integration of Docker Datacenter with Cisco UCS is intuitive. This integration enables enterprises to deploy and manage highly scalable and fast-evolving application containers to the data center environment. This document provides insight into deploying the Docker Enterprise Edition on Cisco UCS. It explains the deployment, management, and scalability aspects of Docker containers. Large numbers of nodes can easily be deployed, and they can be easily made availability to the cluster by using infrastructure automation tools such as Ansible. Docker UCP provides a single control plane for operations from runtime container processing through container lifecycle management, and Cisco UCS provides the converged computing, network, and storage platform needed to run the entire stack.

For More Information

- FlexPod Datacenter with Docker Datacenter for Container Management
- Cisco UCS Infrastructure with Docker Datacenter for Container Management

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

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.**

Printed in USA

C11-737985-01 04/17