Docker Installation Document

Table of Contents

[1 About this document 4](#_Toc58315515)

[1.1 Purpose of this document 4](#_Toc58315516)

[1.2 Audience 4](#_Toc58315517)

[1.3 Abbreviations 4](#_Toc58315518)

[1.4 Assumptions 4](#_Toc58315519)

[1.5 Open items 4](#_Toc58315520)

[2 Overview 5](#_Toc58315521)

[3 Introduction 6](#_Toc58315522)

[3.1 Pre – Requisites 6](#_Toc58315523)

[3.2 Steps to create a user 6](#_Toc58315524)

[4 Approvals and Revision History 8](#_Toc58315525)

List of figures

[Figure 1: User view page 7](#_Toc55292983)

[Figure 2: Create user page 8](#_Toc55292984)

# About this document

## Purpose of this document

The purpose of this document is to guide on how to install the Docker Enterprise Edition on centos 7.

## Audience

All the Devops and network administrator level users of MDAP are the audience of this document.

## Abbreviations

|  |  |
| --- | --- |
| **Abbreviation** | **Expansion** |
| UCP | Universal Control Panel |
| DTR | Docker Trusted Repository |

# Overview

Docker Enterprise provides many capabilities to schedule and manage your container workloads and offers a private docker image repository. The challenge though is while it is fairly easy to install Docker Desktop it isn’t quite the same to install Docker Enterprise. Docker Enterprise invariably requires installing Docker Enterprise Engine, installing Universal Control Plane (UCP) with multiple manager and worker nodes, Docker Trusted Registry (DTR) – a private image repo (like public Docker Hub), and other components.

**Hardware Requirements**

I have mentioned hardware specs based on our Aigilx Environment. Because it will differ based on the container.

**Development Environments**

**Number of servers - 1**

|  |  |  |
| --- | --- | --- |
| **DISK** | **RAM** | **CPU** |
| 200G | 32G | 8 |

**Production Environments**

**Number of servers - 20**

|  |  |  |
| --- | --- | --- |
| **DISK** | **RAM** | **CPU** |
| 50GB | 60GB | 16 |

# Install Docker Enterprise Engine

* The docker-ee rpm file should be loaded via .repo file or repository manager

**yum-config-manager --add-repo "webserver /location/ /docker-ee.repo"**

* Then install the docker EE engine, CLI & containerd packages.

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

# Install UCP (Universal Control Plane)

UCP is the heart of Docker Enterprise. It allows for running containers at scale, across multiple nodes, let us you choose an orchestrator, provides RBAC enabled web UI for operations, and much more.

Use ssh to log in to the host where you want to install UCP.

Run the following command:

**# Pull the latest version of UCP**

**$ docker image pull docker/ucp:2.2.3**

**# Install UCP**

**$ docker container run --rm -it --name ucp \**

**-v /var/run/docker.sock:/var/run/docker.sock \**

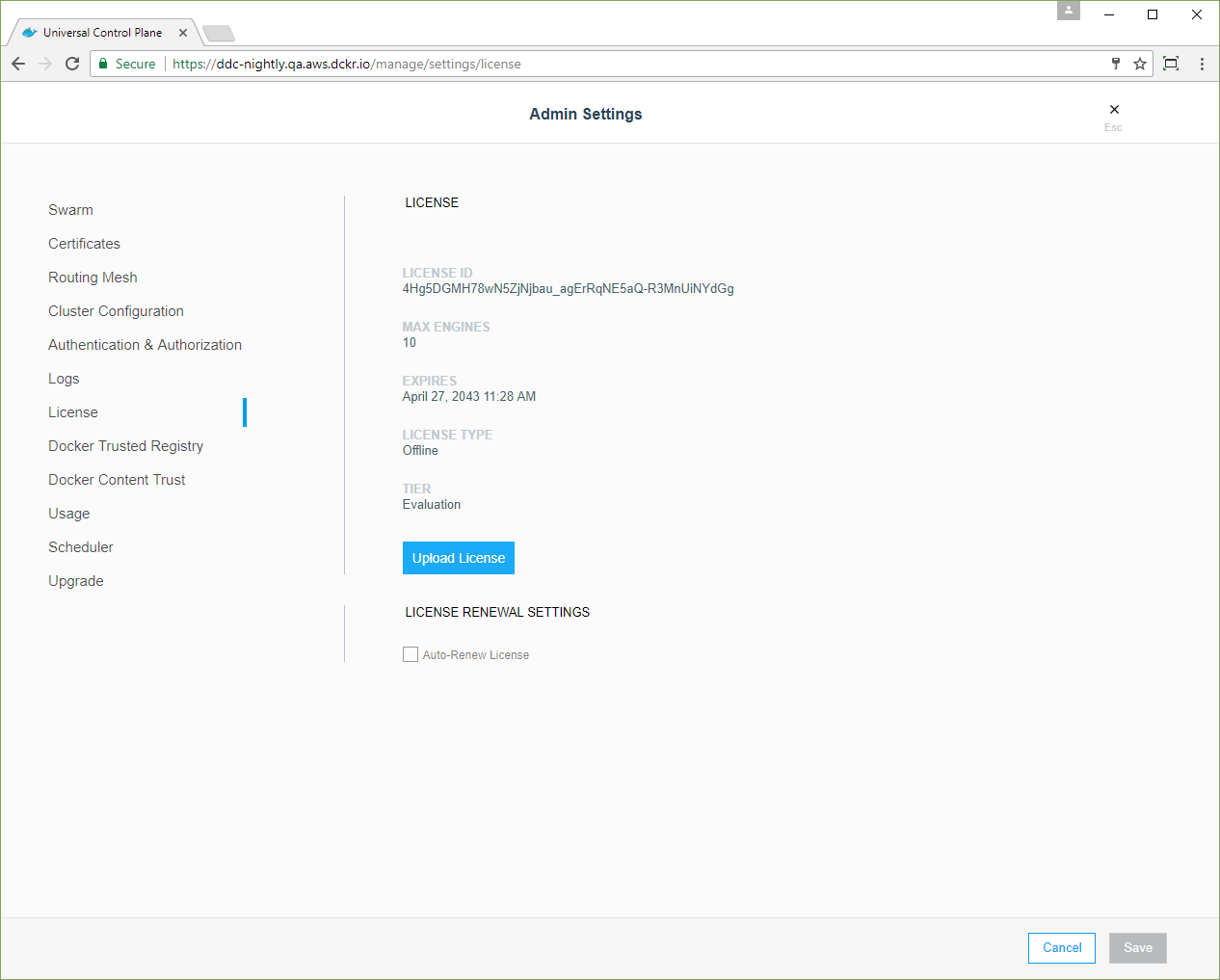
**docker/ucp:2.2.3 install \**

**--host-address <node-ip-address> \**

**--interactive**

## License your installation

In your browser, navigate to the UCP web UI, log in with your administrator credentials and upload your license. Navigate to the **Admin Settings** page and in the left pane, click **License**.

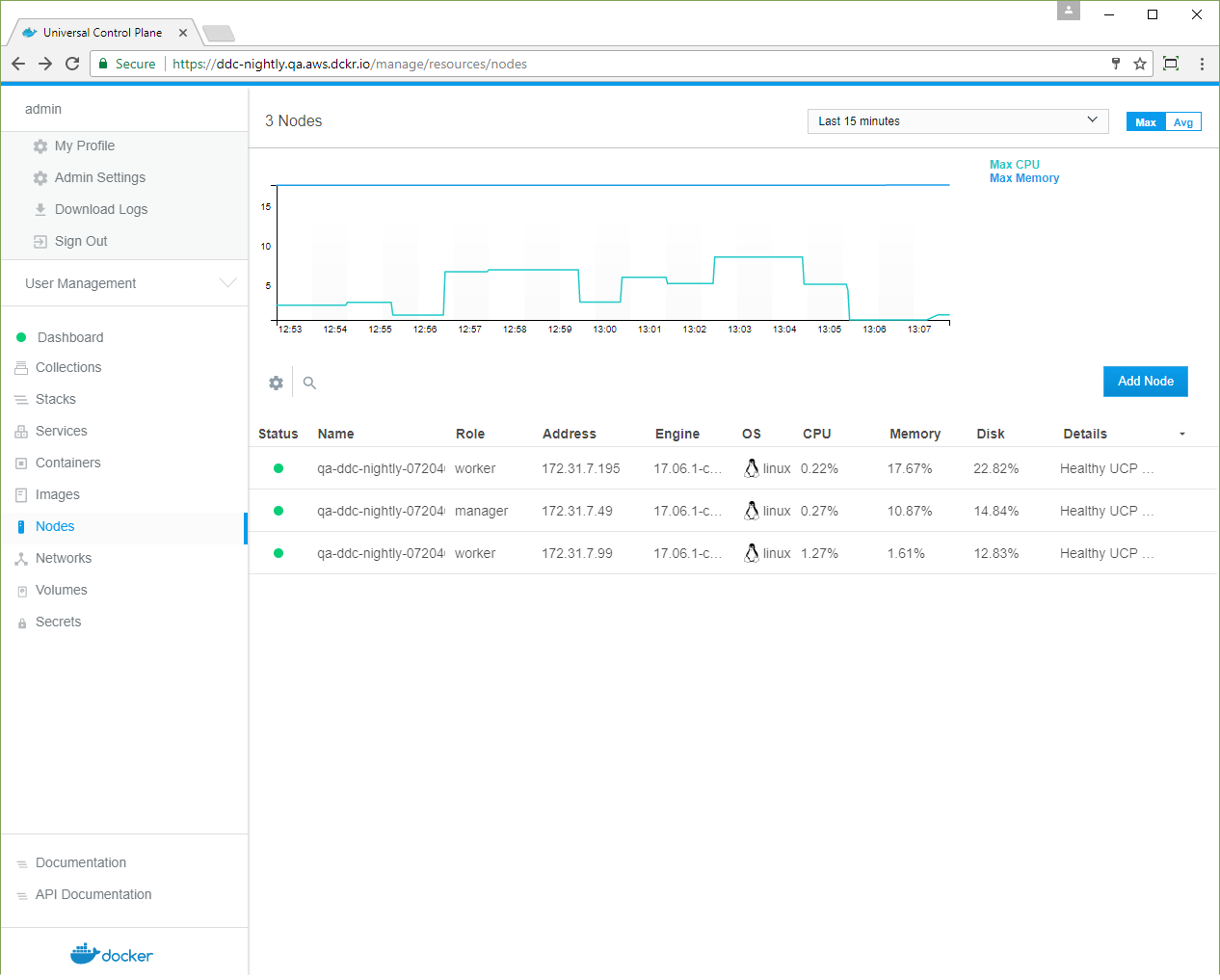


## Join manager nodes

To make your Docker swarm and UCP fault-tolerant and highly available, you can join more manager nodes to it. Manager nodes are the nodes in the swarm that perform the orchestration and swarm management tasks, and dispatch tasks for worker nodes to execute.

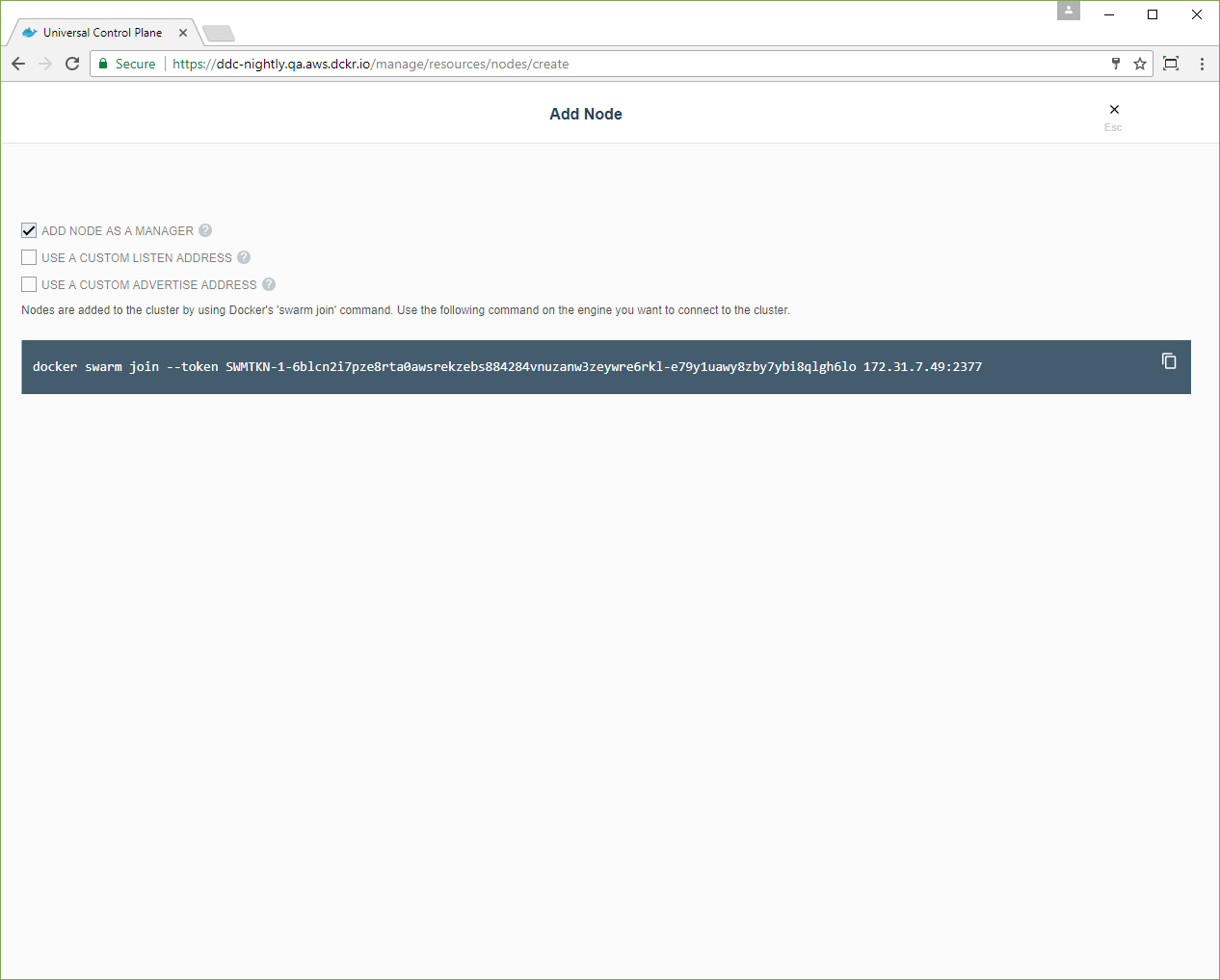
To join manager nodes to the swarm,

1. In the UCP web UI, navigate to the **Nodes** page, and click the **Add Node** button to add a new node.



# 

# To achieve the docker swarm node, copy the docker swarm join command that nodes use to join the swarm



## Join worker nodes

To add more computational resources to your swarm, you can join worker nodes. These nodes execute tasks assigned to them by the manager nodes. Follow the same steps as before, but do not check the **Add node as a manager** option.

# Install DTR (DOCKEr trusted registry)

DTR installation takes a similar route to UCP. It uses a container image ‘dtr’ with install command.

To install DTR you use the docker/dtr image. This image has commands to install, configure, and backup DTR.

Run the following command to install DTR

**# Pull the latest version of DTR**

**$ docker pull docker/dtr:2.3.3**

**# Install DTR**

**$ docker run -it --rm \**

**docker/dtr:2.3.3 install \**

**--ucp-node <ucp-node-name> \**

**--ucp-insecure-tls**

# CHanged parameters

## JSON File logging driver

By default, Docker captures the standard output (and standard error) of all your containers and writes them in files using the JSON format. The JSON format annotates each line with its origin (stout or stderr) and its timestamp. Each log file contains information about only one container.

Add the below mentioned line in /etc/docker/daemon.json file for efficient log collections on running containers. The json file is looks like below.

**{  
 "log-driver": "json-file",  
 "log-opts": {  
   "max-size": "10m",  
   "max-file": "3"  
 }  
}**

There are no default values used in daemon. json. We must create the .json file under /etc/docker for log optimizations.

| **Option** | **Description** | **Example value** |
| --- | --- | --- |
| max size | The maximum size of the log before it is rolled. A positive integer plus a modifier representing the unit of measure (k, m, or g). Defaults to 20m. | --log-opt max-size=10m |
| max file | The maximum number of log files that can be present. If rolling the logs creates excess files, the oldest file is removed. A positive integer. Defaults to 5. | --log-opt max-file=3 |

NOTE: By default, the local driver preserves 100MB of log messages per container and uses automatic compression to reduce the size on disk. The 100MB default value is based on a 20M default size for each file and a default count of 5 for the number of such files (to account for log rotation).