

# **Docker Swarm Setup**

## **for**

## **DiCRA**

mistEO Private Limited

Submitted to



**India**

17.02.2023

## TABLE OF CONTENTS

<b>1. Installation using Docker Framework</b>	<b>3</b>
<b>1.1. Purpose</b>	<b>3</b>
<b>2. Installation Prerequisites</b>	<b>3</b>
2.1. Hardware Requirements	3
2.2. Software Requirements	3
<b>3. Pre-Deployment Steps</b>	<b>4</b>
3.1. Docker Installation (with Internet Connection)	4
3.2. Create Docker Swarm Manager	4
<b>4. Join Docker Swarm as WORKER</b>	<b>5</b>
<b>5. Join Docker Swarm as MANAGER</b>	<b>5</b>
5.1. Docker Swarm Visualiser	6
<b>6. Verify the swarm visualiser</b>	<b>6</b>
<b>7. Fail-over Scenarios</b>	<b>7</b>
7.1. Manager Fail-over Scenario	7
7.2. Worker Node Fail-over scenario	7

## 1. Installation using Docker Framework

### 1.1. Purpose

This document captures the steps involved in installing Docker and Setup Docker SWARM Cluster to install and configure DiCRA Data Automation Platform components.

The setup would primarily address 4 node production configuration - 2 nodes of Master and 2 nodes of Worker setup. The SWARM cluster is designed to scale up or scale down based on the workload requirements.

## 2. Installation Prerequisites

### 2.1. Hardware Requirements

The recommended hardware configuration for Installing DiCRA platform is Linux VMs with the below configurations:

VM Spec (Production) 4 Instances	
RAM	16 GB
vCPU	4 Core
Hard Disk	32 GB (System) + 100 GB (Application) - 2 mount points
OS	Ubuntu Linux

### 2.2. Software Requirements

For docker based installation of DiCRA components, we need the below software to be installed.

Software & Version	Description	Installation Information	File
Docker version 18.0 or above	For Docker based deployment	Open Source Software	

### 3. Pre-Deployment Steps

#### 3.1. Docker Installation (with Internet Connection)

Before starting the installation of DiCRA components, Docker should be installed in all the nodes where DiCRA components are to be deployed. All the steps in this document should be executed by a user with Root or sudo privileges.

Login to each of the 4 Linux nodes and check if the docker is pre-installed and a supported version exists.

Verification Step - Type the below command in the terminal.

```
docker -version
```

It should output the version of the Docker installed. It should be above 18.0.

```
Docker version 20.10.16, build aa7e414
```

If the supported docker is not installed, then install docker using the steps mentioned in the docker setup instructions given in the below link or Refer Appendix as per the OS image you have selected for these VMs.

<https://docs.docker.com/install/linux/docker-ce/ubuntu/#docker-ee-customers/>

Note: The Docker Swarm needs to be started in one of the nodes (Designated Manager Node) and then all the other nodes (Designated Worker Nodes or Additional Manager nodes) should join the Swarm cluster before starting the deployment of the DiCRA Components.

#### 3.2. Create Docker Swarm Manager

Login to the Manager Node (VM1) , or select the first VM as the master node in the production VM instances.

In the OS / Linux terminal type the below command: #Initialize Swarm

```
docker swarm init --advertise-addr <Manager Node IP Address>
```

Use the "ifconfig" command to get the IP address of the manager node. If multiple IP addresses found use an IP address which is accessible

from all the other 3 Linux nodes. All 4 Linux nodes of the production server should be able to communicate with each other. Use “ping <Manager IP>” in all the 3 nodes to confirm the communication and connectivity.

**Verification Step:** #Listing the nodes participating in the swarm

```
docker node ls
```

#### 4. Join Docker Swarm as WORKER

Tokens are required for the worker to join the swarm. To get the worker joining token, login to the Manager Swarm node (VM1) and in the terminal, type the below command:

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

Then use the token given by the above command in the worker nodes to join the Swarm. Login to the Worker nodes (VM 2 & VM3) and in the terminal, type the below command:

```
docker swarm join --token <<Token from Manager Node>>  
<<Manager IP>>:2377
```

**Verification Step:** In the Manager Node (VM1) terminal, type the below command. #Listing the nodes participating in the swarm

```
docker node ls
```

#### 5. Join Docker Swarm as MANAGER

Tokens are required for the manager to join the swarm. To get the manager joining token, login to the Manager Swarm node (VM1) and in the terminal, type the below command:

```
docker swarm join-token -q manager
```

Then use the token given by the above command to join the Swarm. Login to the Manager (VM4) node and type the below command in the terminal.

```
docker swarm join --token <<Token from Manager Node>>  
<<Manager IP>>:2377
```

**Verification Step:** In the Manager Node (VM1) terminal, type the below command. # Listing the nodes participating in the swarm

```
docker node ls
```

The above command should list 4 nodes, VM1 & VM4 would be listed as Manager / Leader nodes and VM2 & VM3 should be listed as worker nodes.

### 5.1. Docker Swarm Visualiser

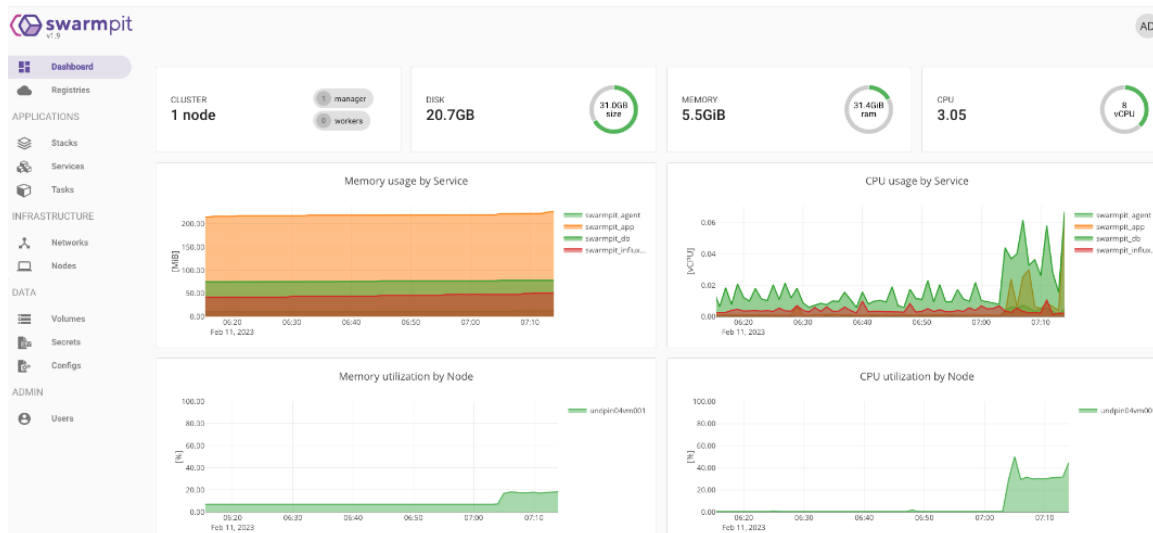
Install a SWARM visualizer to manage the containers with in swarm cluster

```
docker run -it --rm \
--name swarmpit-installer \
--volume /var/run/docker.sock:/var/run/docker.sock \
swarmpit/install:1.9
```

This will install swarmpit visualizer, provide a username and password for swarmpit access during the setup.

## 6. Verify the swarm visualiser

Access the Swarmpit web console at <http://localhost:888>. Use the admin credentials configured during the setup. Below screenshot is from the single node setup for development and testing.



## **7. Fail-over Scenarios**

### **7.1. Manager Fail-over Scenario**

All the Managers need to have access to the yml files and docker image files, so that they can restart any of the service. Ensure that all the relevant yml files are copied or the shared drive is mounted in both master nodes. Both master nodes should be configured to the same docker registry to pull the relevant container images.

### **7.2. Worker Node Fail-over scenario**

If any of the worker nodes fails, then the manager node will take care of starting additional services in the currently available nodes based on the load distribution among them. No manual intervention is required in this case.