Docker Swarm Configuration



Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Doc Version | Date | Create By | Description of the Revision | Reviewed By |
| Draft (V.0) | 24-May-2021 | Rajeev Ranjan | Docker Swarm configuration | Suresh Kumar |

**Contents**

[1 Introduction 3](#__RefHeading___Toc2271_3757422114)

[2 Purpose 3](#__RefHeading___Toc2273_3757422114)

[2.1 Scope 3](#__RefHeading___Toc2275_3757422114)

[3 Prerequisites](#__RefHeading___Toc2279_3757422114)[.](#__RefHeading___Toc2277_3757422114) [3](#__RefHeading___Toc2279_3757422114)

4. Setup of Docker CE and Docker Swarm……………………………………………………………...4

1. Introduction

A Docker Swarm is a group of either physical or virtual machines that are running the Docker application and that have been configured to join together in a cluster. Once a group of machines have been clustered together, you can still run the Docker commands that you're used to, but they will now be carried out by the machines in your cluster. The activities of the cluster are controlled by a swarm manager, and machines that have joined the cluster are referred to as nodes.

2. Purpose

**Docker swarm** is a container orchestration tool, meaning that it allows the user to manage multiple containers deployed across multiple host machines. One of the key benefits associated with the operation of a **docker swarm** is the high level of availability offered for applications.

2.1 Scope

By default **Docker Swarm** uses a default address pool 10.0. 0.0/8 for global **scope** (overlay) networks. Every network that does not have a subnet specified will have a subnet sequentially allocated from this pool. In some circumstances it may be desirable to use a different default IP address pool for networks

3. [Prerequisites](#__RefHeading___Toc2279_3757422114)

We have taken three machines and installed with amazon linux2 OS flavor.. We have completed basic network security with docker machine,  timezone and IP addresses allocated on each of servers and server details is.

|  |  |  |  |
| --- | --- | --- | --- |
| **Hostname** | **Port** | **Static Ip Address** | **Local IP Address** |
| Docker Server-1 | 2377,4789,7946 | 3.108.85.174 | 172.31.21.28 |
| Docker Server-1 | 2377,4789,7946 | 3.108.31.87 | 172.31.29.154 |
| Docker Server-1 | 2377,4789,7946 | 13.235.47.187 | 172.31.19.42 |

4. Setup of Docker CE and Docker Swarm

Note:- We choose the three node for Docker swarm and install Docker CE service on Three node.

Step-1:- Docker provides a repository where you can fetch the stable Docker CE version. Install it with this command:

#amazon-linux-extras install epel

#amazon-linux-extras enable docker

#yum install docker-19.03.6ce-4.amzn2

#service docker restart

Step-2:- Create the Manager node in single instance.

#docker swarm init --advertise-addr <MANAGER-IP> (run on master)

Note:- output is display in terminal for add worker in swarm.

Step-3:- To add a worker to this swarm, run the following command:

docker swarm join \

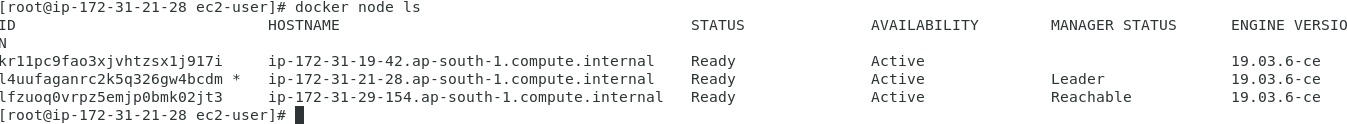
--token SWMTKN-1-49nj1cmql0jkz5s954yi3oex3nedyz0fb0xx14ie39trti4wxv-8vxv8rssmk743ojnwacrr2e7c \

second instance-IP:2377 -----run on slaves

Note:- Allow the 2377,,4789,7946 port with iptables.

Step-4: Check the docker node on cloud

# docker node ls



Step-5: Create the single Network for three nodes.

#docker network create --driver overlay --attachable nvkn-network

Step5:-Create Service of portainer agent for two nodes

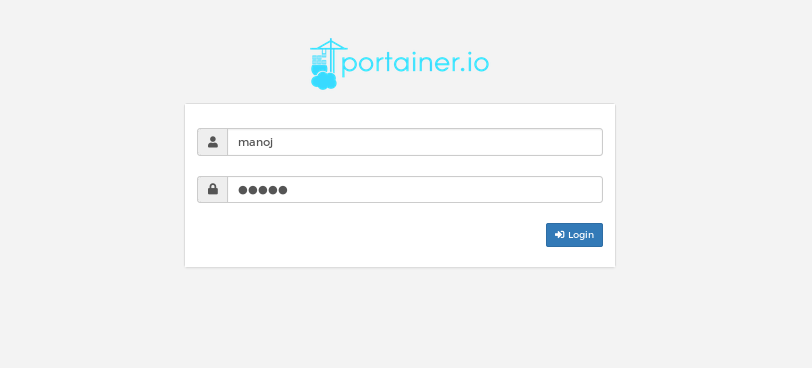
#docker service create --name portainer\_agent --network portainer\_agent\_network --publish mode=host,target=9001,published=9001 --mode global --mount type=bind,src=//var/run/docker.sock,dst=/var/run/docker.sock --mount type=bind,src=//var/lib/docker/volumes,dst=/var/lib/docker/volumes portainer/agent----->run on master

#Step-6:Run the Portainer on local system for Access UI.-----→no need to run as portainer ui is already deployed on 172.16.0.68

#docker run -d -p 8000:8000 -p 9100:9000 -v /var/run/docker.sock:/var/run/docker.sock -v portainer\_data:/data portainer/portainer

Step7- login the portainer UI and set endpoints of docker swarm

#[http://127.0.0.1:9100](http://127.0.0.1:9100/)



Step8:- After login Go to endpoint (endpoint is environment in latest portainer) option and click endpoint.

Step9:- click the add endpoint and give some information.

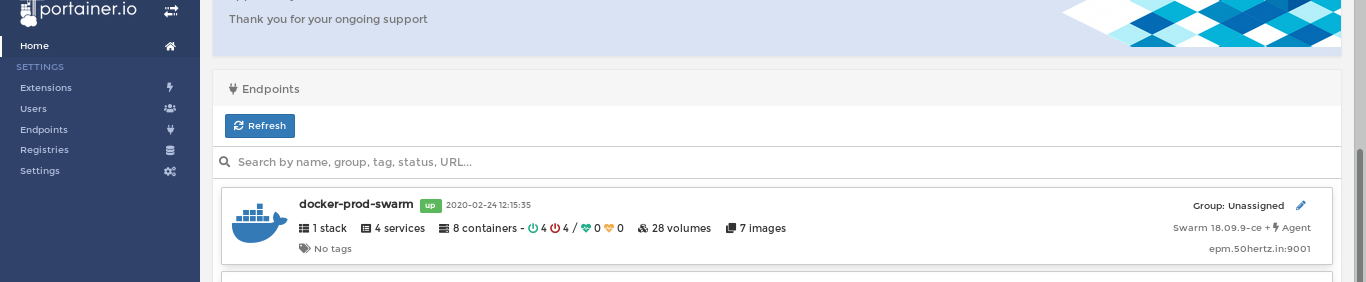
Name:- Docker swarm Prod

Endpoint URL:- Docker swarm IP:9001

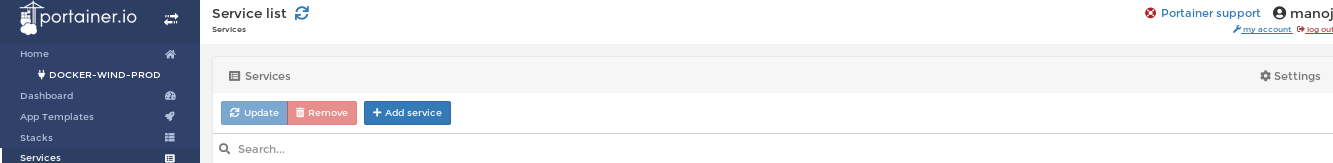
Public IP:- Docker swarm Public IP

=> Click the add endpoint.

Step10:- Again go to home tab and show this page.



Step11:- Click the docker prod-swarm and create service for Deploy jar image on cloud docker swarm.



Step12:- Click the Add Service and give the some information for create docker image.

Name:- Deploy Application name

Registry:- Docker Hub(docker hub credential set in registries option)

Image:- 50hertz/energy:export-actual-forecast-batch

Sceduling-mode:- Replicated

Network :- application-network

config:- if you required the create config file under config option.

Volume:- bind volume source to destination path if you have required.

Step13:- Again Create the Service option and show the image.

