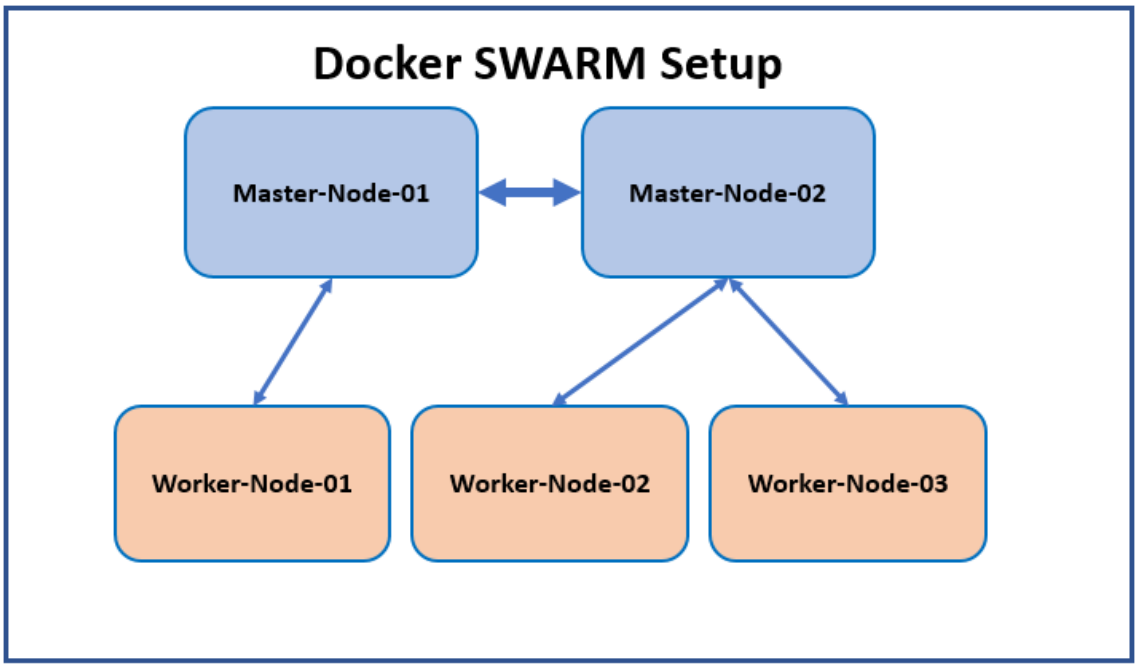
Objective

Setup a Docker swarm Env and run the Dockerfile.

Diagram



Steps to follow

Step1.

Create a Manager/Leader Node.

On the first Master/Manager node, install the docker and run the below command

 # docker swarm init –advertise-addr *<ip add of Manager node>*

```
[root@Docker-Mas01 ~]# ifconfig enp0s8
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.131  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::9378:b7d5:2a1a:646  prefixlen 64  scopeid 0x20<link>
        inet6 fe80::903d:5bb9:8de:7cef  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:d8:70:e8  txqueuelen 1000  (Ethernet)
        RX packets 226  bytes 23815 (23.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 207  bytes 28921 (28.2 KiB)
```

This is the ip of this Manager node.

```
[root@Docker-Mas01 ~]# docker swarm init --advertise-addr 192.168.56.131
Swarm initialized: current node (wxsq743pffyyzea9790a1pf7n) is now a manager.

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

    docker swarm join --token SWMTKN-1-5uduzfokyhfue4lxayx80mm8hcbqptec7p6uqtl5xml8fpm7dt-6zpf7zg
131:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions
```

In the above output, “docker swarm join” is for the worker node and NOT for another Manager node.

Now for adding another manager node to this Leader Manager node, run the below command

```
[root@Docker-Mas01 ~]# docker swarm join-token manager
To add a manager to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-5uduzfokyhfue4lxayx80mm8hcbqptec7p6uqtl5xml8fpm7dt-du3of
09aba2lpelujit7m40yx 192.168.56.131:2377

[root@Docker-Mas01 ~]#
```

$ docker swarm join-token manager

Now we have both the tokens that are required, to join the nodes to this SWARM cluster.

Also, by running command “docker node ls” will show the number of nodes

```
[root@Docker-Mas01 ~]# docker node ls
ID                            HOSTNAME            STATUS              AVAILABILITY        MANAGER STATUS
      ENGINE VERSION
0woeuw4jib5xi2jmg7tzmcg0f *   Docker-Mas01        Ready               Active              Leader
      18.09.0
[root@Docker-Mas01 ~]#
```

Currently only one node is available.

Step2:

Adding another manager node to this First Manager node.

Pre-Requisite → install Docker and get the services up.

```
[root@DS-Mgr02 ~]# service docker status
Redirecting to /bin/systemctl status docker.service
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled;
   Active: active (running) since Wed 2019-07-31 23:04:09 EDT; 6s a
     Docs: http://docs.docker.com
 Main PID: 13771 (dockerd-current)
   CGroup: /system.slice/docker.service
           ├─13771 /usr/bin/dockerd-current --add-runtime docker-ru
           └─13777 /usr/bin/docker-containerd-current -l unix:///va

Jul 31 23:04:07 DS-Mgr02 dockerd-current[13771]: time="2019-07-31T2
Jul 31 23:04:08 DS-Mgr02 dockerd-current[13771]: time="2019-07-31T2
Jul 31 23:04:08 DS-Mgr02 dockerd-current[13771]: time="2019-07-31T2
Jul 31 23:04:08 DS-Mgr02 dockerd-current[13771]: time="2019-07-31T2
Jul 31 23:04:08 DS-Mgr02 dockerd-current[13771]: time="2019-07-31T2
Jul 31 23:04:08 DS-Mgr02 dockerd-current[13771]: time="2019-07-31T2
Jul 31 23:04:09 DS-Mgr02 dockerd-current[13771]: time="2019-07-31T2
```

On the 2nd manager node run the command that has the token as manager.

Below is the ip of the 2nd manager node

```
[root@DS-Mgr02 ~]# ifconfig enp0s8
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.132  netmask 255.255.255.0  broadcast 19
        inet6 fe80::905d:5bb9:8de:7cef  prefixlen 64  scopeid 0x
        ether 08:00:27:46:49:a6  txqueuelen 1000  (Ethernet)
        RX packets 2052  bytes 136706 (133.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 834  bytes 194193 (189.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions
```

Note:-- "enp0s8" is specific to my system. With ifconfig command you can get to know your interface.

Now running the below command on the 2nd manager.

docker swarm join --token <token-ID of the master node> <ip of the master node>:2377

```
[root@DS-Mgr02 ~]# docker swarm join --token SWMTKN-1-5uduzfokyhfue4lxayx80mm8hcbqptec7p6uqtl5x
ml8fpm7dt-du3of09aba2lpelujit7m40y 192.168.56.131:2377
Error response from daemon: rpc error: code = 14 desc = grpc: the connection is unavailable
[root@DS-Mgr02 ~]#
```

There is an error, this is because the firewall port **"2377"** is not opened on the Master01 node.

In fact the port has to be opened on both **Master01 and mnager02**.

1. **$ firewall-cmd --list-all**

```
[root@DS-Mgr02 ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3 enp0s8
  sources:
  services: ssh dhcpv6-client
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

[root@DS-Mgr02 ~]#
```

The port "2377" is not listed here, which means its blocked on this instance.

Run the below command to open the port **"2377"**

2. **$ firewall-cmd --zone=public --permanent --add-port=2377/tcp**

```
[root@DS-Mgr02 ~]# firewall-cmd --zone=public --permanent --add-port=2377/tcp
success
[root@DS-Mgr02 ~]#
```

Here, we are opening the port "2377" permanentely.

We need to reload the firewall to get this affected on the system.

3. **$ firewall-cmd --reload**

```
[root@DS-Mgr02 ~]# firewall-cmd --reload
success
[root@DS-Mgr02 ~]#
```

```
[root@DS-Mgr02 ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3 enp0s8
  sources:
  services: ssh dhcpv6-client
  ports: 2377/tcp
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

[root@DS-Mgr02 ~]#
```

Carry the above 3 sub steps on the "Master" node as well.

```
[root@Docker-Mas01 ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3 enp0s8
  sources:
  services: ssh dhcpv6-client http
  ports: 8080/tcp 90/tcp 95/tcp 2377/tcp
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

[root@Docker-Mas01 ~]#
```

Now running the join-swarm command

```
[root@DS-Mgr02 ~]# docker swarm join --token SWMTKN-1-5uduzfokyhfue4lxayx80mm8hcbqptec7p6uqtl5x
ml8fpm7dt-du3of09aba2lpelujit7m40yx 192.168.56.131:2377
This node joined a swarm as a manager.
[root@DS-Mgr02 ~]#
```

The Manager02 is now successfully joined the Master node.

$ docker node ls

This will show the list of all the nodes part of the swarm cluster.

```
[root@DS-Mgr02 ~]# docker node ls
ID                            HOSTNAME      STATUS  AVAILABILITY  MANAGER STATUS
s6d5hbkahhjuxv1azranux38z *   DS-Mgr02      Ready   Active        Reachable
wxsq743pffyyzea9790a1pf7n     Docker-Mas01  Ready   Active        Leader
[root@DS-Mgr02 ~]#
```

Step3:

Setup up the worker node01 and connect it to the Master01

Note:-- Docker needs to be installed.

```
[root@Docker-WN01 ~]# service docker status
Redirecting to /bin/systemctl status docker.service
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor pre
   Active: active (running) since Thu 2019-08-01 00:29:07 EDT; 9s ago
     Docs: http://docs.docker.com
 Main PID: 13604 (dockerd-current)
   CGroup: /system.slice/docker.service
           ├─13604 /usr/bin/dockerd-current --add-runtime docker-runc=/usr/lib
           └─13610 /usr/bin/docker-containerd-current -l unix:///var/run/docke

Aug 01 00:29:04 Docker-WN01 dockerd-current[13604]: time="2019-08-01T00:29:04
Aug 01 00:29:05 Docker-WN01 dockerd-current[13604]: time="2019-08-01T00:29:05
Aug 01 00:29:05 Docker-WN01 dockerd-current[13604]: time="2019-08-01T00:29:05
```

```
[root@Docker-Mas01 ~]# docker swarm join-token worker
To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-5uduzfokyhfue4lxayx80mm8hcbqptec7p6uqtl5xml8fpm7dt-
131:2377

[root@Docker-Mas01 ~]#
```

This is the token that would be used to join the worker node to the master

```
[root@Docker-WN01 ~]# docker swarm join --token SWMTKN-1-5uduzfokyhfue4lxayx80mm8hcbqptec7p6u
217c82 192.168.56.131:2377
This node joined a swarm as a worker.
[root@Docker-WN01 ~]#
```

Note:-- The firewall port opening is NOT a requirement for the worker node, as it is only receiving commands from Swarm Master.

```
[root@Docker-Mas01 ~]# docker node ls
ID                            HOSTNAME            STATUS              AVAILABILITY        MANAGER STATUS
s6d5hbkahhjuxv1azranux38z     DS-Mgr02            Ready               Active              Reachable
wxsg743nffyyze2g709a1pf7n *   Docker-Mas01        Ready               Active              Leader
p0vhlhic1alycjkbxijfmzt5v     Docker-WN01         Ready               Active
[root@Docker-Mas01 ~]#
```

The worker node is added successfully and Active.

Similarly – Lets add the Worker node 02 to the manager 02.

Make sure the docker is running on the manager 02.

```
[root@Docker-WN02 ~]# service docker status
Redirecting to /bin/systemctl status docker.service
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; ve
   Active: active (running) since Thu 2019-08-01 00:46:10 EDT; 5min ag
     Docs: http://docs.docker.com
 Main PID: 14060 (dockerd-current)
   CGroup: /system.slice/docker.service
           ├─14060 /usr/bin/dockerd-current --add-runtime docker-runc=
           └─14066 /usr/bin/docker-containerd-current -l unix:///var/r

Aug 01 00:46:08 Docker-WN02 dockerd-current[14060]: time="2019-08-01T0
Aug 01 00:46:09 Docker-WN02 dockerd-current[14060]: time="2019-08-01T0
Aug 01 00:46:09 Docker-WN02 dockerd-current[14060]: time="2019-08-01T0
Aug 01 00:46:09 Docker-WN02 dockerd-current[14060]: time="2019-08-01T0
Aug 01 00:46:09 Docker-WN02 dockerd-current[14060]: time="2019-08-01T0
```

Taken the join token from the **manager02 (DS-Mgr02)**

```
[root@DS-Mgr02 ~]# docker swarm join-token worker
To add a worker to this swarm, run the following command:

    docker swarm join \
    --token SWMTKN-1-5uduzfokyhfue4lxayx80mm8hcbqptec7p6uqtl5xml8fpm7dt-6zpf7zgfwory6b
    192.168.56.132:2377

[root@DS-Mgr02 ~]#
```

```
[root@Docker-WN02 ~]# docker swarm join \
>     --token SWMTKN-1-5uduzfokyhfue4lxayx80mm8hcbqptec7p6uqtl5xml8fpm7dt-6zpf7zgfwory6biyhjk217c82 \
>     192.168.56.132:2377
This node joined a swarm as a worker.
[root@Docker-WN02 ~]#
```

The worker node 02 is added successfully to the mgr02

$ docker node ls (run this on either of the Manager node)

```
[root@Docker-Mas01 ~]# docker node ls
ID                            HOSTNAME            STATUS              AVAILABILITY        MANAGER STATUS
s6d5hbkahhjuxv1azranux38z     DS-Mgr02            Ready               Active              Reachable
wxsq743pffyyzea9790a1pf7n *   Docker-Mas01        Ready               Active              Leader
p0vhlhic1alycjkbxijfmzt5v     Docker-WN01         Ready               Active
7gqtxo275ug3ey0oyxcu97zyj     Docker-WN02         Ready               Active
[root@Docker-Mas01 ~]#
```

This completes the setup of the Docker Swarm and the Worker nodes.

Next → we will see how to run and manage the containers on the Swarm cluster.