Lab 6 – Docker-swarm
Manage cluster of machines

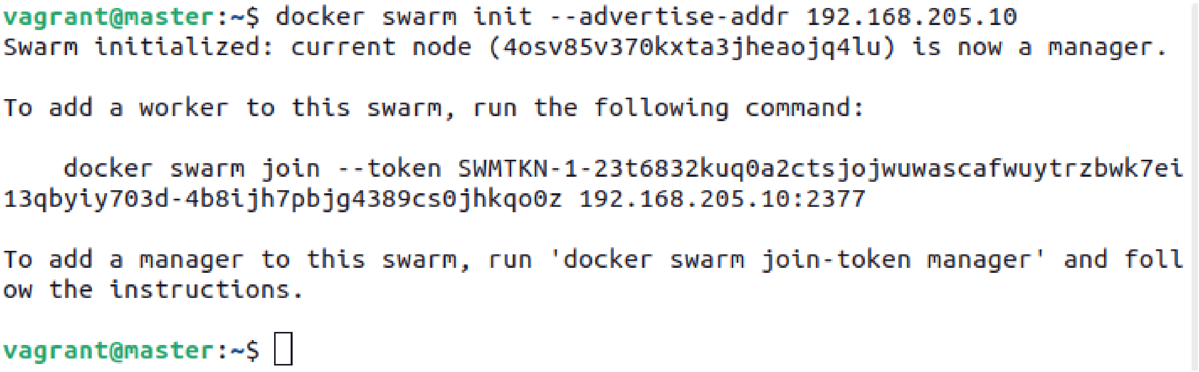
We're going to build a cluster of Docker servers using Docker Swarm. You have 3 Docker servers:

- 1 manager
- 2 workers

Cluster initialization

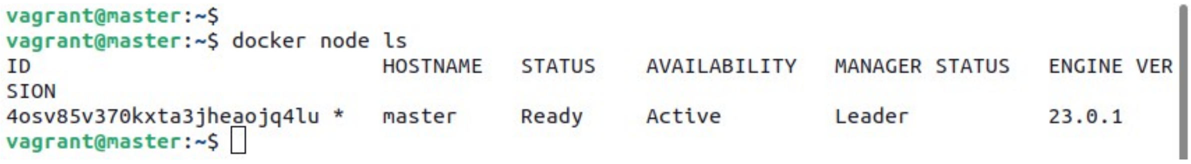
1. Initialize the swarm cluster from the manager host using its IP from the same range than the 2 workers:

   docker swarm init --advertise-addr = 192.168.205.10



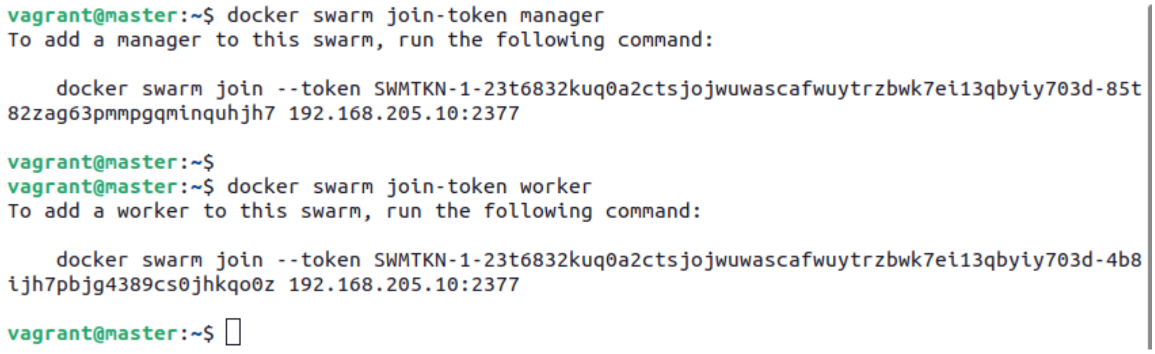
```
vagrant@master:~$ docker swarm init --advertise-addr 192.168.205.10
Swarm initialized: current node (4osv85v370kxta3jheaojq4lu) is now a manager.

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

    docker swarm join --token SWMTKN-1-23t6832kuq0a2ctsjojwuwascafwuytrzbwk7ei
13qbyiy703d-4b8ijh7pbjg4389cs0jhkqo0z 192.168.205.10:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and foll
ow the instructions.

vagrant@master:~$ 
```

- What does this command generate?
- Check that the cluster has been created and that the manager is the single member.



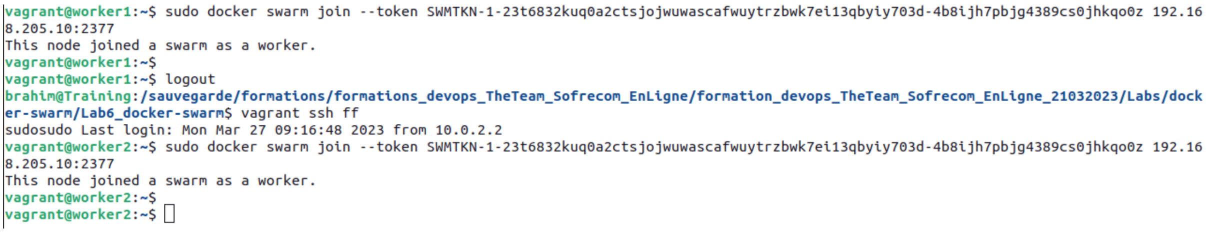
```
vagrant@master:~$
vagrant@master:~$ docker node ls
ID                            HOSTNAME   STATUS    AVAILABILITY   MANAGER STATUS   ENGINE VER
SION
4osv85v370kxta3jheaojq4lu *   master     Ready     Active         Leader           23.0.1
vagrant@master:~$ 
```

- What is the token generated by this command used for? Display the 2 tokens.



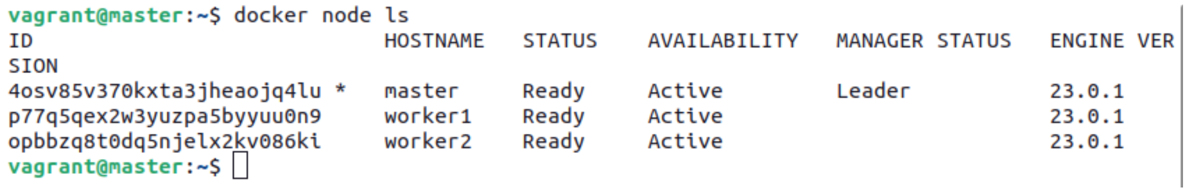
```
vagrant@master:~$ docker swarm join-token manager
To add a manager to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-23t6832kuq0a2ctsjojwuwascafwuytrzbwk7ei13qbyiy703d-85t
82zag63pmmpgqminquhjh7 192.168.205.10:2377

vagrant@master:~$
vagrant@master:~$ docker swarm join-token worker
To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-23t6832kuq0a2ctsjojwuwascafwuytrzbwk7ei13qbyiy703d-4b8
ijh7pbjg4389cs0jhkqo0z 192.168.205.10:2377

vagrant@master:~$
```

2. Join worker1 and worker2 VMs to the cluster (as workers).



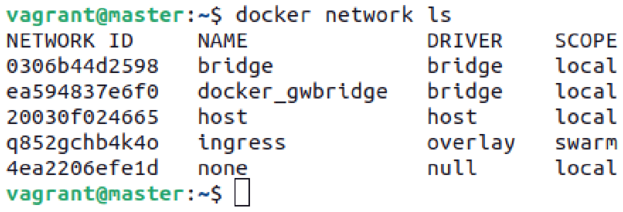
```
vagrant@worker1:~$ sudo docker swarm join --token SWMTKN-1-23t6832kuq0a2ctsjojwuwascafwuytrzbwk7ei13qbyiy703d-4b8ijh7pbjg4389cs0jhkqo0z 192.16
8.205.10:2377
This node joined a swarm as a worker.
vagrant@worker1:~$
vagrant@worker1:~$ logout
brahim@Training:/sauvegarde/formations/formations_devops_TheTeam_Sofrecom_EnLigne/formation_devops_TheTeam_Sofrecom_EnLigne_21032023/Labs/dock
er-swarm/Lab6_docker-swarm$ vagrant ssh ff
sudosudo Last login: Mon Mar 27 09:16:48 2023 from 10.0.2.2
vagrant@worker2:~$ sudo docker swarm join --token SWMTKN-1-23t6832kuq0a2ctsjojwuwascafwuytrzbwk7ei13qbyiy703d-4b8ijh7pbjg4389cs0jhkqo0z 192.16
8.205.10:2377
This node joined a swarm as a worker.
vagrant@worker2:~$
vagrant@worker2:~$
```

- Check that the 2 workers are in the cluster and that they are ready.



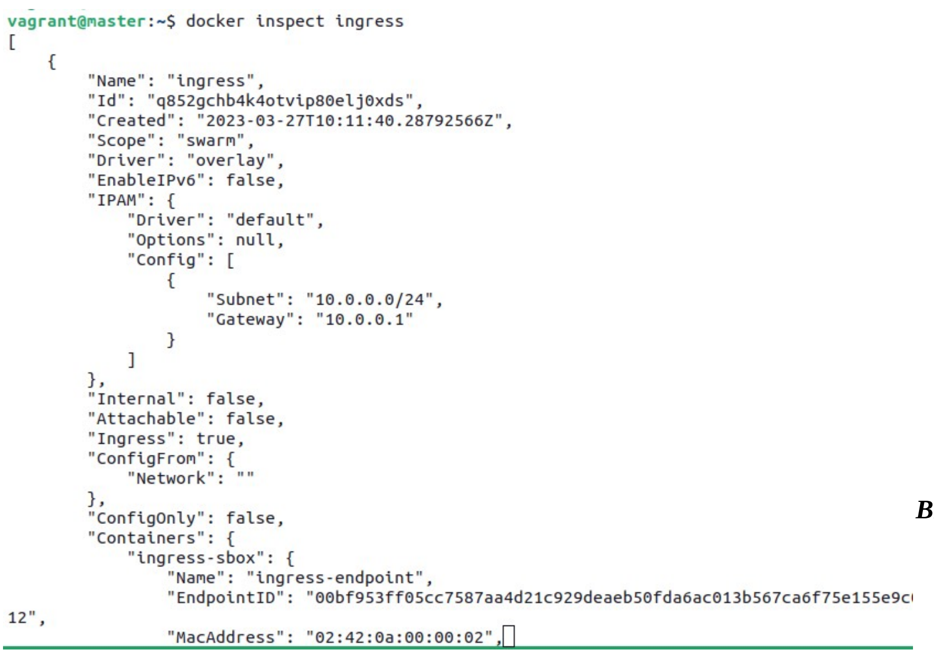
```
vagrant@master:~$ docker node ls
ID                            HOSTNAME   STATUS    AVAILABILITY   MANAGER STATUS   ENGINE VER
SION
4osv85v370kxta3jheaojq4lu *   master     Ready     Active         Leader           23.0.1
p77q5qex2w3yuzpa5byyuu0n9     worker1    Ready     Active                          23.0.1
opbbzq8t0dq5njelx2kv086ki     worker2    Ready     Active                          23.0.1
vagrant@master:~$
```

3. What networks are created following cluster initialization.

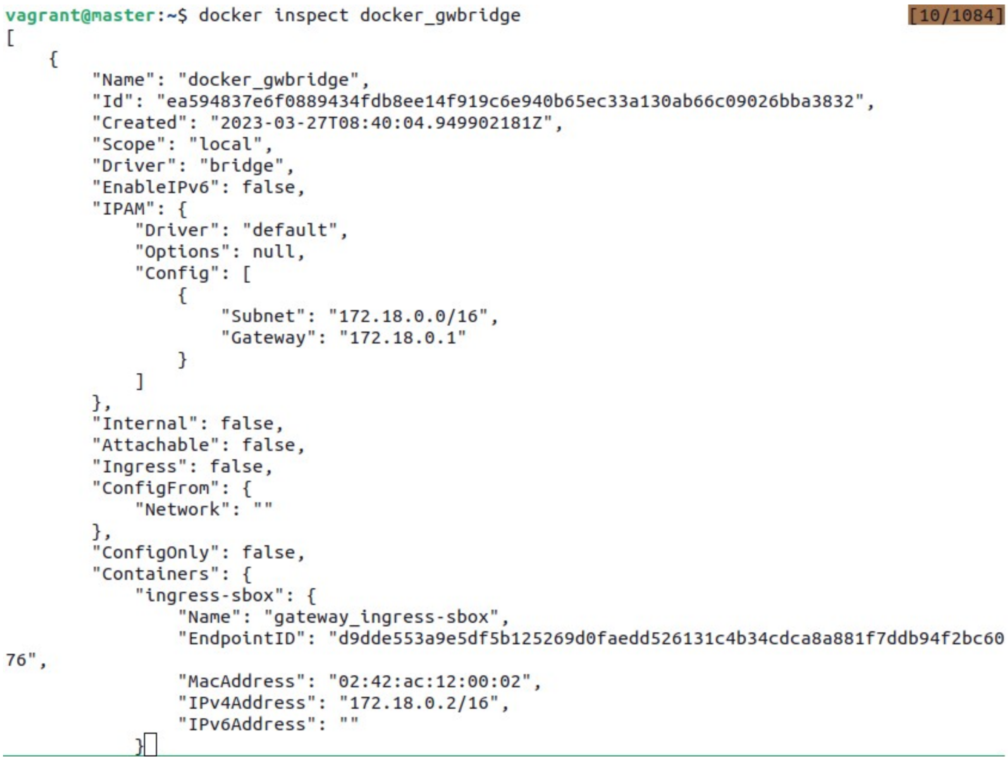


```
vagrant@master:~$ docker network ls
NETWORK ID     NAME              DRIVER    SCOPE
0306b44d2598   bridge            bridge    local
ea594837e6f0   docker_gwbridge   bridge    local
20030f024665   host              host      local
q852gchb4k4o   ingress           overlay   swarm
4ea2206efe1d   none              null      local
vagrant@master:~$
```

- Describe them.



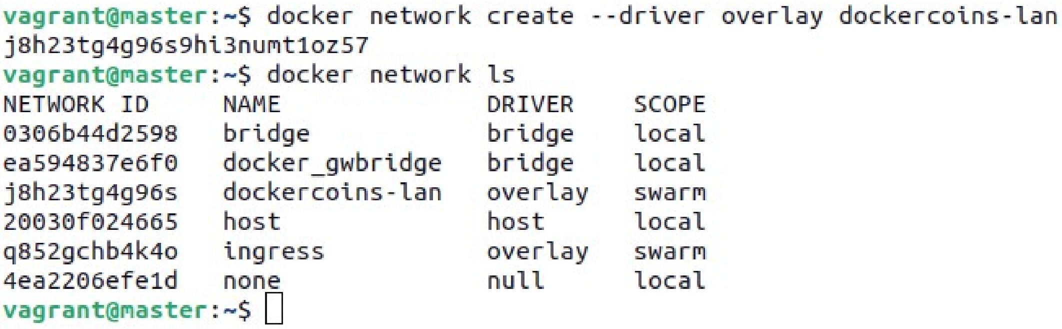
```
vagrant@master:~$ docker inspect ingress
[
    {
        "Name": "ingress",
        "Id": "q852gchb4k4otvip80elj0xds",
        "Created": "2023-03-27T10:11:40.28792566Z",
        "Scope": "swarm",
        "Driver": "overlay",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "10.0.0.0/24",
                    "Gateway": "10.0.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": true,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "ingress-sbox": {
                "Name": "ingress-endpoint",
                "EndpointID": "00bf953ff05cc7587aa4d21c929deaeb50fda6ac013b567ca6f75e155e9c
12",
                "MacAddress": "02:42:0a:00:00:02",
```



```
vagrant@master:~$ docker inspect docker_gwbridge                                  [10/1084]
[
    {
        "Name": "docker_gwbridge",
        "Id": "ea594837e6f0889434fdb8ee14f919c6e940b65ec33a130ab66c09026bba3832",
        "Created": "2023-03-27T08:40:04.949902181Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "172.18.0.0/16",
                    "Gateway": "172.18.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "ingress-sbox": {
                "Name": "gateway_ingress-sbox",
                "EndpointID": "d9dde553a9e5df5b125269d0faedd526131c4b34cdca8a881f7ddb94f2bc60
76",
                "MacAddress": "02:42:ac:12:00:02",
                "IPv4Address": "172.18.0.2/16",
                "IPv6Address": ""
            }
```

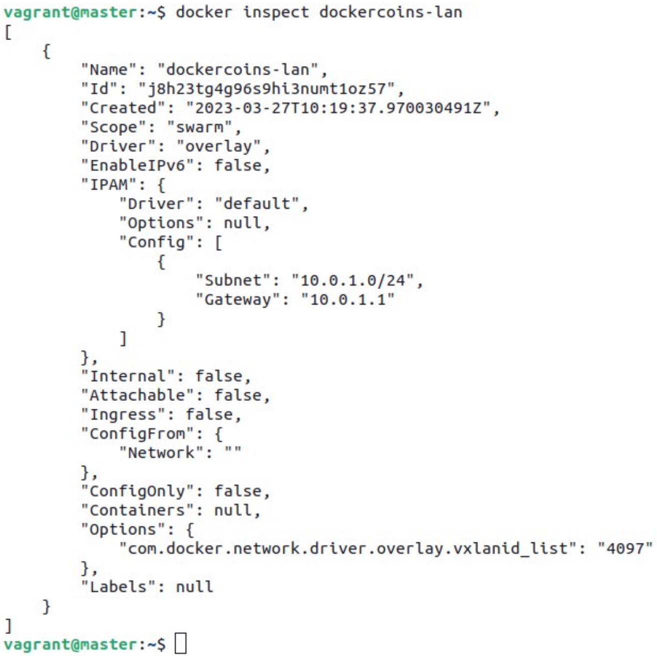
Launch of the Dockercoins application on Swarm

4. Create an overlay network to isolate all services of the application.



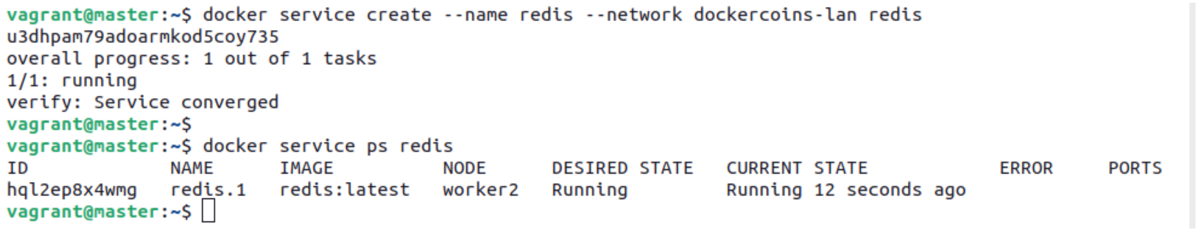
```
vagrant@master:~$ docker network create --driver overlay dockercoins-lan
j8h23tg4g96s9hi3numt1oz57
vagrant@master:~$ docker network ls
NETWORK ID     NAME              DRIVER    SCOPE
0306b44d2598   bridge            bridge    local
ea594837e6f0   docker_gwbridge   bridge    local
j8h23tg4g96s   dockercoins-lan   overlay   swarm
20030f024665   host              host      local
q852gchb4k4o   ingress           overlay   swarm
4ea2206efe1d   none              null      local
vagrant@master:~$
```

- View details of our network



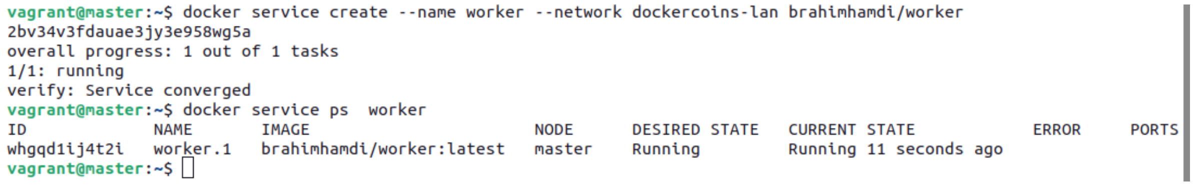
```
vagrant@master:~$ docker inspect dockercoins-lan
[
    {
        "Name": "dockercoins-lan",
        "Id": "j8h23tg4g96s9hi3numt1oz57",
        "Created": "2023-03-27T10:19:37.970030491Z",
        "Scope": "swarm",
        "Driver": "overlay",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "10.0.1.0/24",
                    "Gateway": "10.0.1.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": null,
        "Options": {
            "com.docker.network.driver.overlay.vxlanid_list": "4097"
        },
        "Labels": null
    }
]
vagrant@master:~$
```

5. Create the service running the redis database.

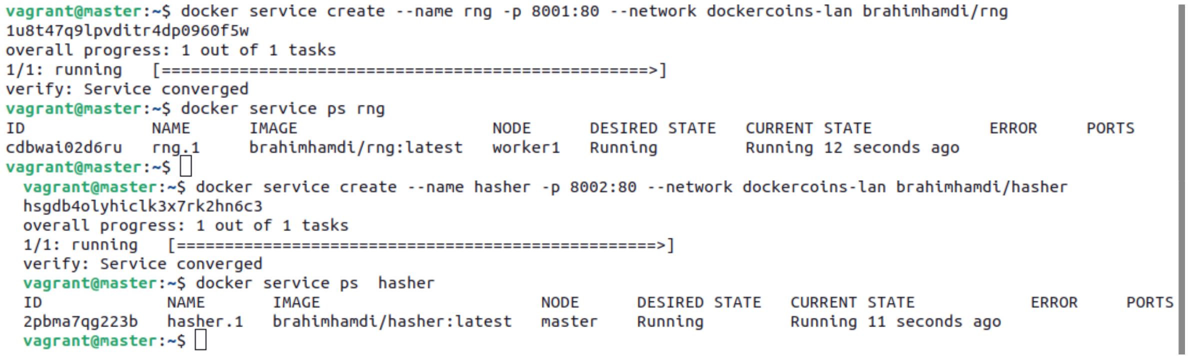


```
vagrant@master:~$ docker service create --name redis --network dockercoins-lan redis
u3dhpam79adoarmkod5coy735
overall progress: 1 out of 1 tasks
1/1: running
verify: Service converged
vagrant@master:~$
vagrant@master:~$ docker service ps redis
ID             NAME      IMAGE          NODE      DESIRED STATE   CURRENT STATE            ERROR     PORTS
hql2ep8x4wmg   redis.1   redis:latest   worker2   Running         Running 12 seconds ago
vagrant@master:~$
```

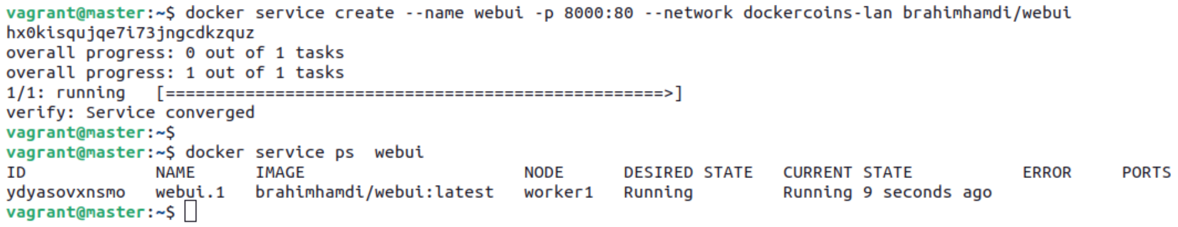
- ○ Wait until the server is in "running"
- ○ Once redis is ready, run the hasher, rng, and worker services.



```
vagrant@master:~$ docker service create --name worker --network dockercoins-lan brahimhamdi/worker
2bv34v3fdauae3jy3e958wg5a
overall progress: 1 out of 1 tasks
1/1: running
verify: Service converged
vagrant@master:~$ docker service ps  worker
ID             NAME       IMAGE                       NODE      DESIRED STATE   CURRENT STATE            ERROR     PORTS
whgqd1ij4t2i   worker.1   brahimhamdi/worker:latest   master    Running         Running 11 seconds ago
vagrant@master:~$
```

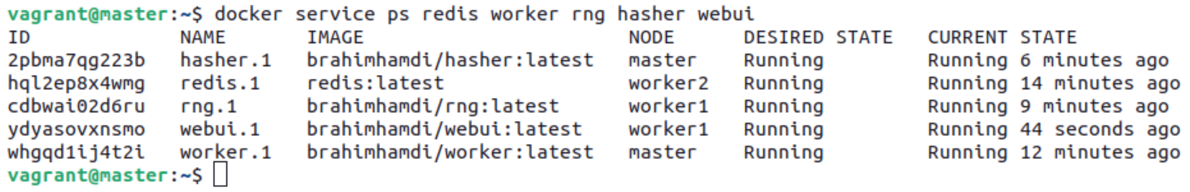


```
vagrant@master:~$ docker service create --name rng -p 8001:80 --network dockercoins-lan brahimhamdi/rng
1u8t47q9lpvditr4dp0960f5w
overall progress: 1 out of 1 tasks
1/1: running   [==================================================>]
verify: Service converged
vagrant@master:~$ docker service ps rng
ID             NAME      IMAGE                    NODE      DESIRED STATE   CURRENT STATE            ERROR     PORTS
cdbwai02d6ru   rng.1     brahimhamdi/rng:latest   worker1   Running         Running 12 seconds ago
vagrant@master:~$
  vagrant@master:~$ docker service create --name hasher -p 8002:80 --network dockercoins-lan brahimhamdi/hasher
  hsgdb4olyhiclk3x7rk2hn6c3
  overall progress: 1 out of 1 tasks
  1/1: running   [==================================================>]
  verify: Service converged
  vagrant@master:~$ docker service ps  hasher
  ID             NAME       IMAGE                       NODE      DESIRED STATE   CURRENT STATE            ERROR     PORTS
  2pbma7qg223b   hasher.1   brahimhamdi/hasher:latest   master    Running         Running 11 seconds ago
  vagrant@master:~$
```



```
vagrant@master:~$ docker service create --name webui -p 8000:80 --network dockercoins-lan brahimhamdi/webui
hx0kisqujqe7i73jngcdkzquz
overall progress: 0 out of 1 tasks
overall progress: 1 out of 1 tasks
1/1: running   [==================================================>]
verify: Service converged
vagrant@master:~$
vagrant@master:~$ docker service ps  webui
ID             NAME      IMAGE                      NODE      DESIRED STATE   CURRENT STATE           ERROR     PORTS
ydyasovxnsmo   webui.1   brahimhamdi/webui:latest   worker1   Running         Running 9 seconds ago
vagrant@master:~$
```

- ○ Check that these services are "running".

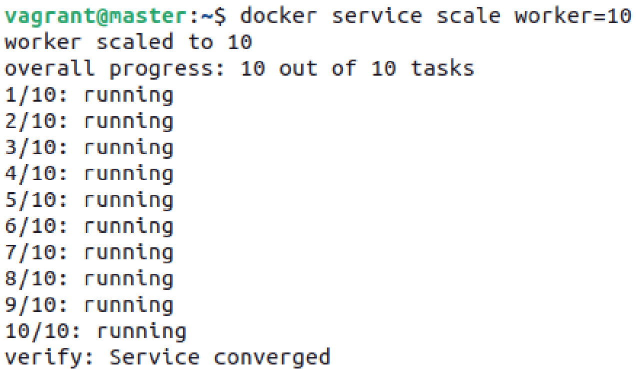


```
vagrant@master:~$ docker service ps redis worker rng hasher webui
ID             NAME       IMAGE                       NODE      DESIRED STATE   CURRENT STATE
2pbma7qg223b   hasher.1   brahimhamdi/hasher:latest   master    Running         Running 6 minutes ago
hql2ep8x4wmg   redis.1    redis:latest                worker2   Running         Running 14 minutes ago
cdbwai02d6ru   rng.1      brahimhamdi/rng:latest      worker1   Running         Running 9 minutes ago
ydyasovxnsmo   webui.1    brahimhamdi/webui:latest    worker1   Running         Running 44 seconds ago
whgqd1ij4t2i   worker.1   brahimhamdi/worker:latest   master    Running         Running 12 minutes ago
vagrant@master:~$
```

- ○ Give the IP address of each container as well as the node on which it is run.

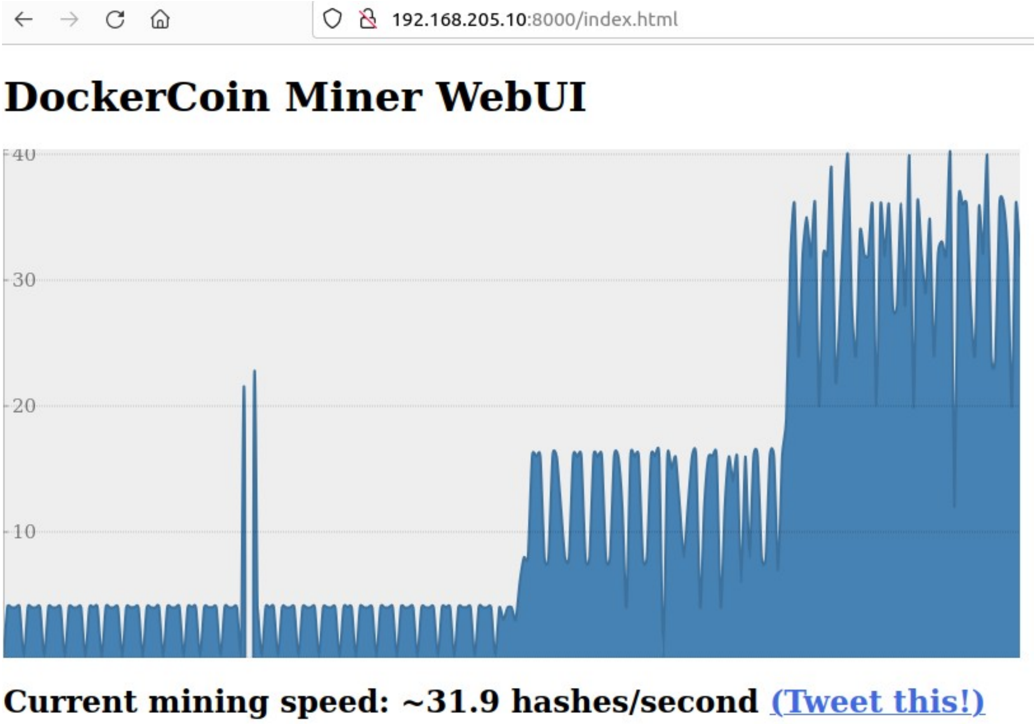
Scaling on the swarm

6. scale out the worker service to 10 replicas.



```
vagrant@master:~$ docker service scale worker=10
worker scaled to 10
overall progress: 10 out of 10 tasks
1/10: running
2/10: running
3/10: running
4/10: running
5/10: running
6/10: running
7/10: running
8/10: running
9/10: running
10/10: running
verify: Service converged
```

- Once all the replicas are "running", display the webui. What is the rate of generation of dockercoins?

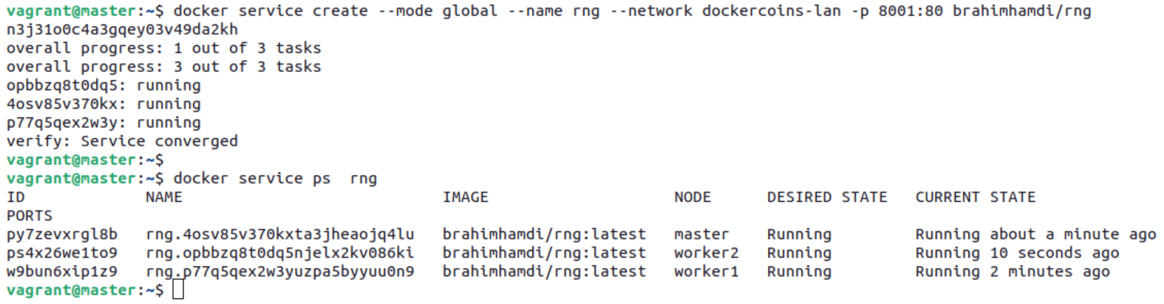


192.168.205.10:8000/index.html

DockerCoin Miner WebUI

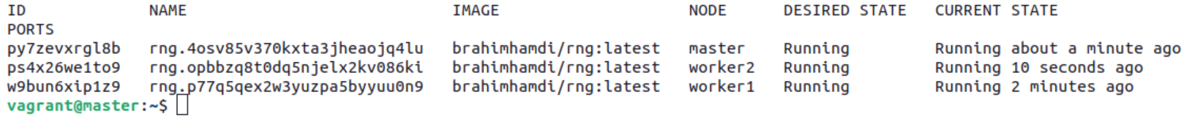
Current mining speed: ~31.9 hashes/second (Tweet this!)

7. We want to run exactly one *rng* instance on each host. Unfortunately, this scheduling mode cannot be activated / deactivated for a existing service.
So, we have to start by removing the existing rng service, then recreate it with global scheduling



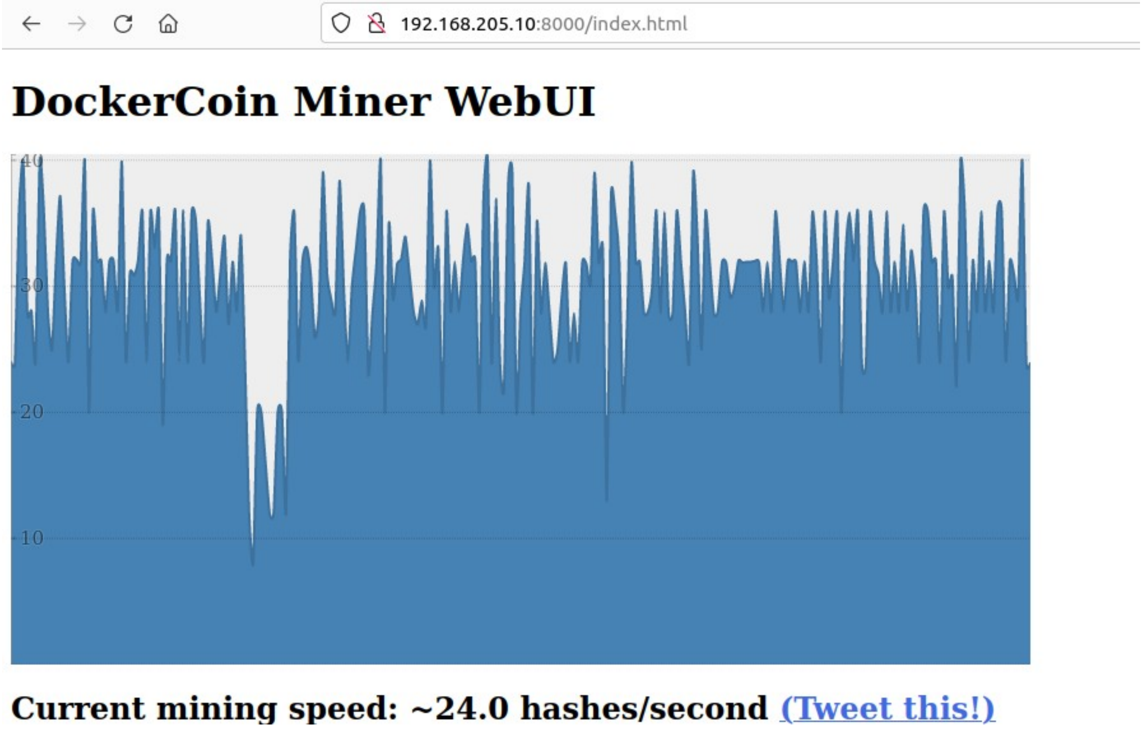
```
vagrant@master:~$ docker service create --mode global --name rng --network dockercoins-lan -p 8001:80 brahimhamdi/rng
n3j31o0c4a3gqey03v49da2kh
overall progress: 1 out of 3 tasks
overall progress: 3 out of 3 tasks
opbbzq8t0dq5: running
4osv85v370kx: running
p77q5qex2w3y: running
verify: Service converged
vagrant@master:~$
vagrant@master:~$ docker service ps  rng
ID             NAME                           IMAGE                   NODE      DESIRED STATE   CURRENT STATE
PORTS
py7zevxrgl8b   rng.4osv85v370kxta3jheaojq4lu   brahimhamdi/rng:latest   master    Running         Running about a minute ago
ps4x26we1to9   rng.opbbzq8t0dq5njelx2kv086ki   brahimhamdi/rng:latest   worker2   Running         Running 10 seconds ago
w9bun6xip1z9   rng.p77q5qex2w3yuzpa5byyuu0n9   brahimhamdi/rng:latest   worker1   Running         Running 2 minutes ago
vagrant@master:~$ 
```

8. Check that only one instance of rng is running on each node.



```
ID             NAME                           IMAGE                   NODE      DESIRED STATE   CURRENT STATE
PORTS
py7zevxrgl8b   rng.4osv85v370kxta3jheaojq4lu   brahimhamdi/rng:latest   master    Running         Running about a minute ago
ps4x26we1to9   rng.opbbzq8t0dq5njelx2kv086ki   brahimhamdi/rng:latest   worker2   Running         Running 10 seconds ago
w9bun6xip1z9   rng.p77q5qex2w3yuzpa5byyuu0n9   brahimhamdi/rng:latest   worker1   Running         Running 2 minutes ago
vagrant@master:~$ 
```

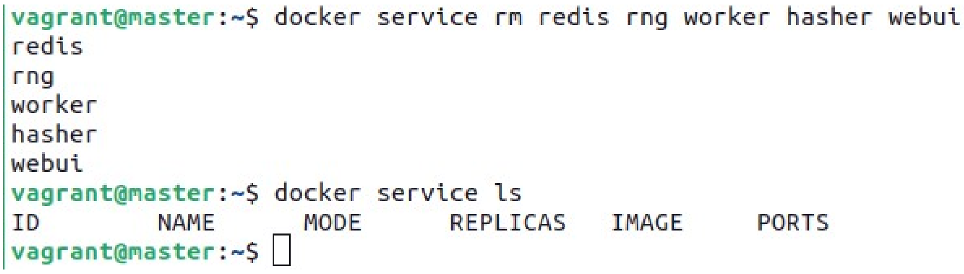
- ○ Open the webui, what is the Dockercoins generation speed?



DockerCoin Miner WebUI

Current mining speed: ~24.0 hashes/second (Tweet this!)

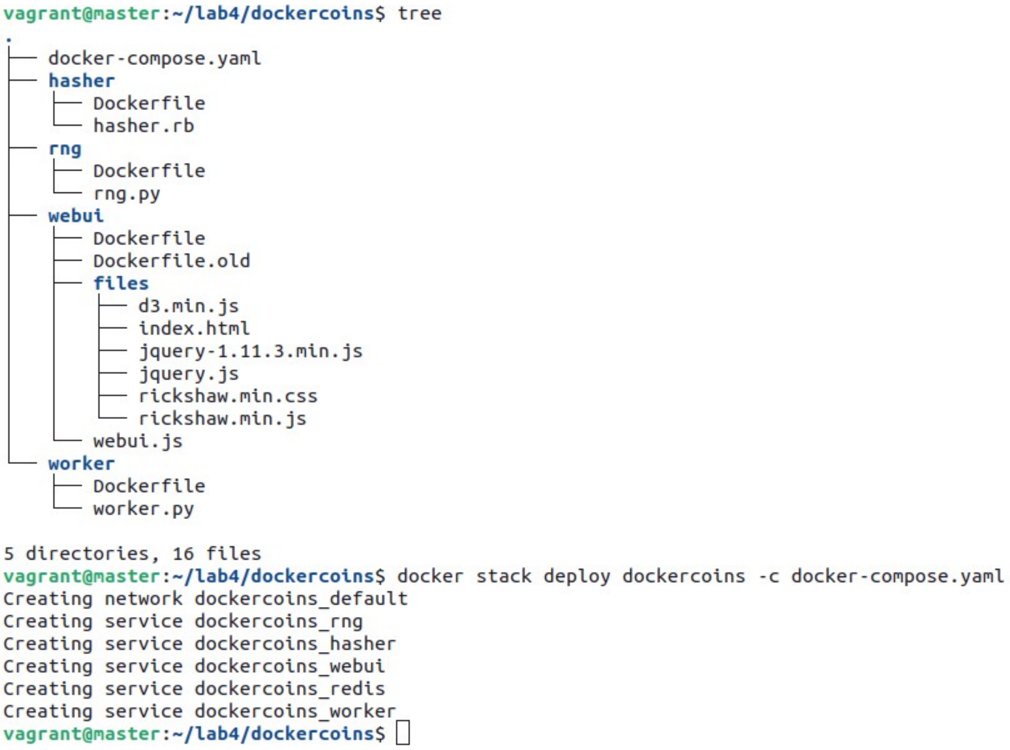
- ○ Remove all services of the application.



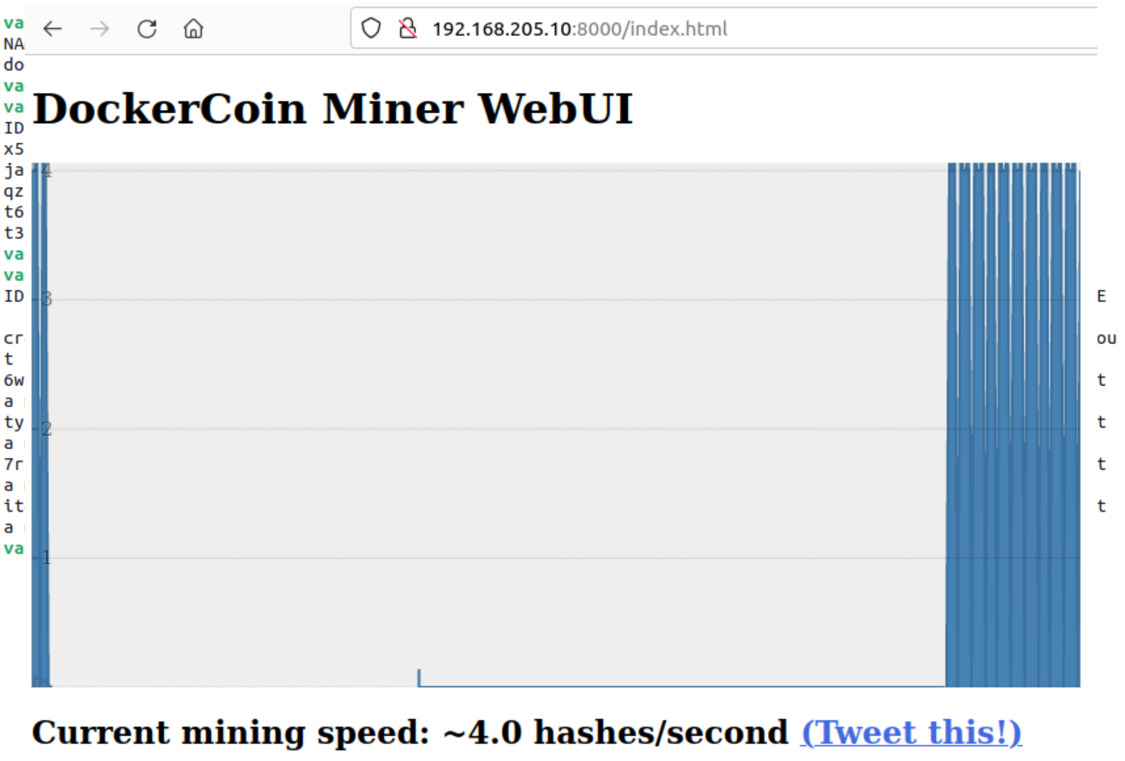
```
vagrant@master:~$ docker service rm redis rng worker hasher webui
redis
rng
worker
hasher
webui
vagrant@master:~$ docker service ls
ID        NAME      MODE      REPLICAS   IMAGE     PORTS
vagrant@master:~$ 
```

Docker stack

9. Now we will combine docker swarm and docker compose to be able to deploy on the cluster using docker-compose.yml file.
    - Check that all services running



```
vagrant@master:~/lab4/dockercoins$ tree
.
├── docker-compose.yaml
├── hasher
│   ├── Dockerfile
│   └── hasher.rb
├── rng
│   ├── Dockerfile
│   └── rng.py
├── webui
│   ├── Dockerfile
│   ├── Dockerfile.old
│   ├── files
│   │   ├── d3.min.js
│   │   ├── index.html
│   │   ├── jquery-1.11.3.min.js
│   │   ├── jquery.js
│   │   ├── rickshaw.min.css
│   │   └── rickshaw.min.js
│   └── webui.js
└── worker
    ├── Dockerfile
    └── worker.py

5 directories, 16 files
vagrant@master:~/lab4/dockercoins$ docker stack deploy dockercoins -c docker-compose.yaml
Creating network dockercoins_default
Creating service dockercoins_rng
Creating service dockercoins_hasher
Creating service dockercoins_webui
Creating service dockercoins_redis
Creating service dockercoins_worker
vagrant@master:~/lab4/dockercoins$ 
```



192.168.205.10:8000/index.html

DockerCoin Miner WebUI

Current mining speed: ~4.0 hashes/second (Tweet this!)