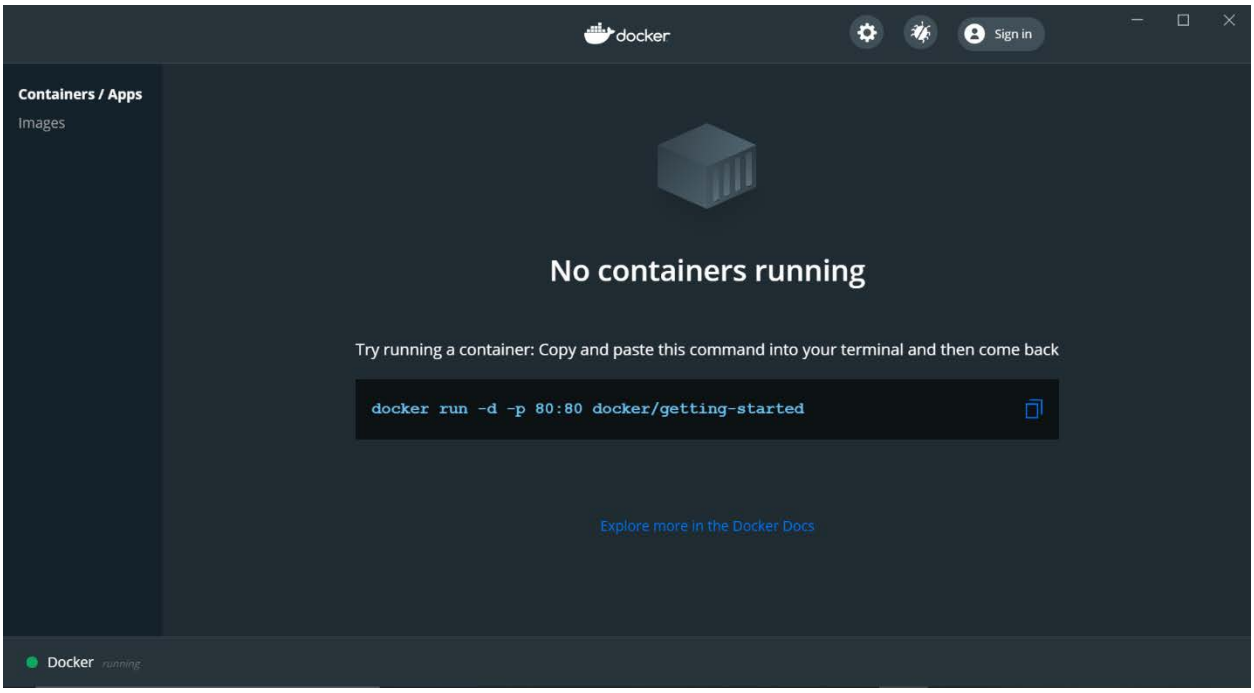
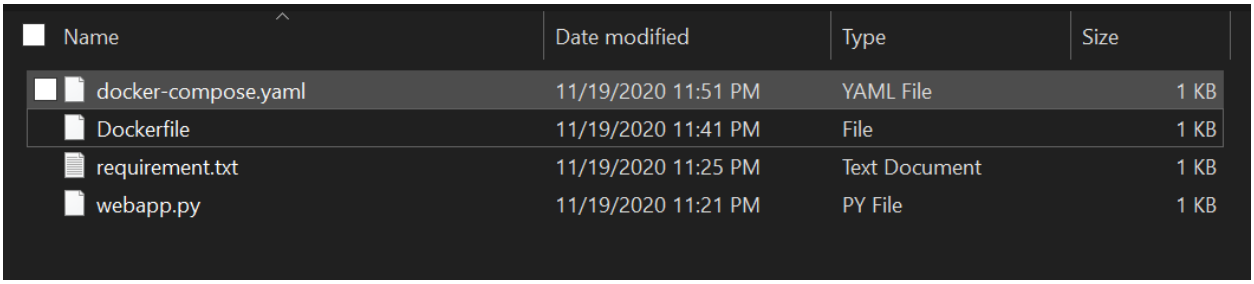
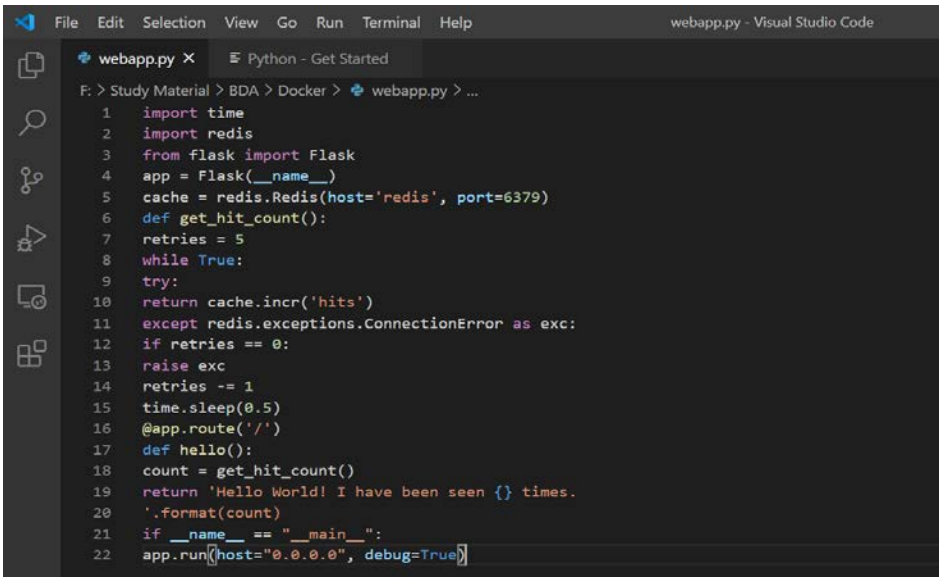
Docker Installation on Windows





Name	Date modified	Type	Size
docker-compose.yaml	11/19/2020 11:51 PM	YAML File	1 KB
Dockerfile	11/19/2020 11:41 PM	File	1 KB
requirement.txt	11/19/2020 11:25 PM	Text Document	1 KB
webapp.py	11/19/2020 11:21 PM	PY File	1 KB



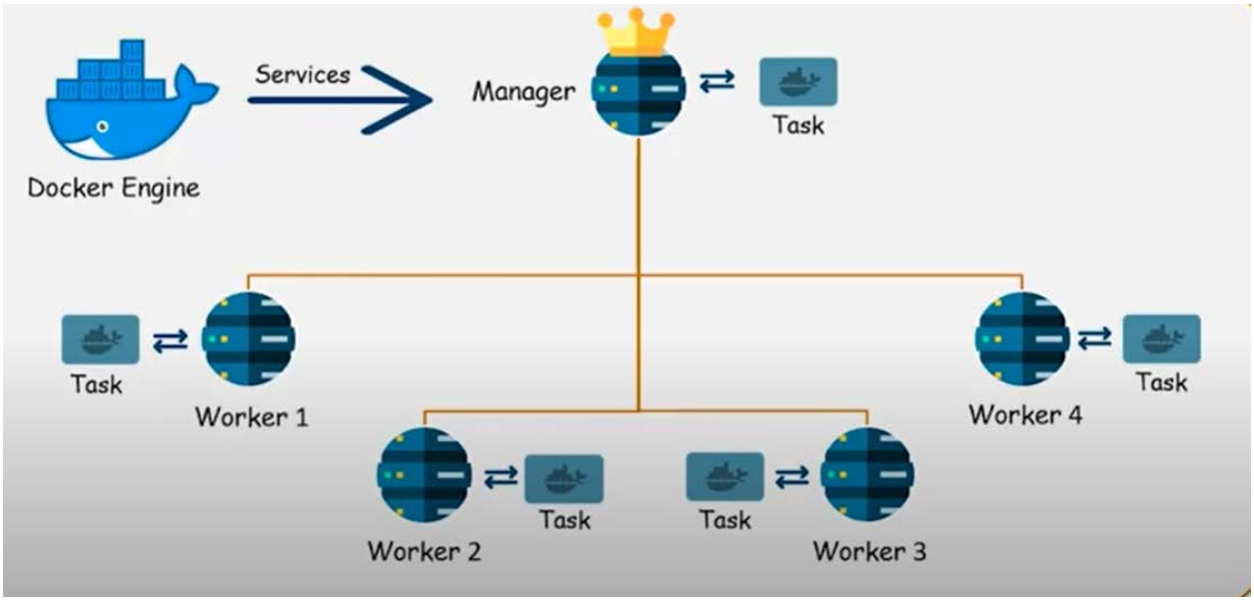
Files for running docker compose from edureka tutorial

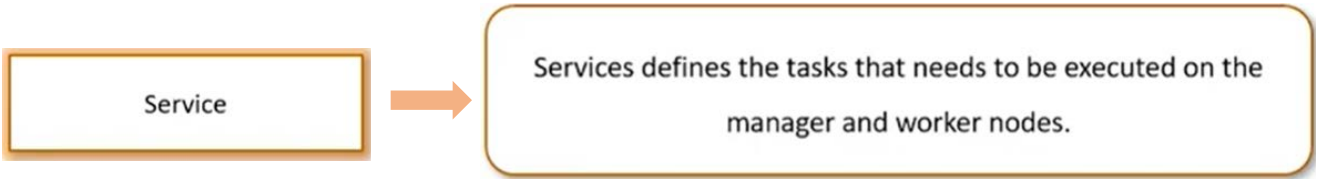
Methods of Creating a Docker Swarm

Q: What are the methods for creating a docker swarm?

Docker swarm is a container orchestration tool part of the Docker Engine. With it, developers and IT administrators can deploy and mange a cluster of Docker nodes as a virtual system.

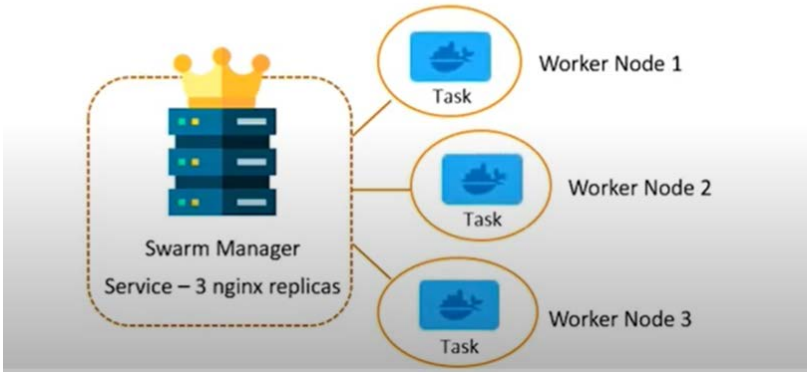
Docker Swarm Architecture





Task

Tasks refer to the docker containers that execute the commands defined in the service.



Manager Node

The manager node is responsible for:

- Accepting commands and creating service objects
- Allocating IP addresses to tasks
- Assigning tasks to nodes
- Instructing a worker to run a task

Worker Node

Worker Nodes are responsible for checking assigned tasks and executing containers.

Creating a Swarm

➢ Setup

You need three Linux hosts which have Docker installed and can communicate over a network. These can be physical machines, virtual machines, Amazon EC2 instances, or hosted in some other way. You can even use Docker Machine from a Linux, Mac, or Windows host.

➢ Inititalizing a Swarm

1. Make sure the Docker Engine daemon is started on the host machines.
2. Open a terminal and ssh into the machine where you want to run your manager node. This tutorial uses a machine named manager1. If you use Docker Machine, you can connect to it via SSH using the following command:

   $ docker-machine ssh manager1

3. Run the following command to create a new swarm:

   $ docker swarm init --advertise-addr <MANAGER-IP>

```
$ docker swarm init --advertise-addr 192.168.99.100
Swarm initialized: current node (dxn1zf6l61qsb1josjja83ngz) is now a manager.

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

    docker swarm join \
    --token SWMTKN-1-49nj1cmql0jkz5s954yi3oex3nedyz0fb0xx14ie39trti4wxv-8vxv8rssmk74
    192.168.99.100:2377

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

The *--advertise-addr* flag configures the manager node to publish its address as *192.168.99.100*. The other nodes in the swarm must be able to access the manager at the IP address.

The output includes the commands to join new nodes to the swarm. Nodes will join as managers or workers depending on the value for the *--token* flag.

4. Run *docker info* to view the current state of the swarm:

```
$ docker info

Containers: 2
Running: 0
Paused: 0
Stopped: 2
  ...snip...
Swarm: active
  NodeID: dxn1zf6l61qsb1josjja83ngz
  Is Manager: true
  Managers: 1
  Nodes: 1
  ...snip...
```

5. Run the *docker node ls* command to view information about nodes:

```
$ docker node ls

ID                           HOSTNAME  STATUS  AVAILABILITY  MANAGER STATUS
dxn1zf6l61qsb1josjja83ngz *  manager1  Ready   Active        Leader
```

The * next to the node ID indicates that you're currently connected on this node.

Once you've created a swarm with a manager node, you're ready to add worker nodes.

6. Open a terminal and ssh into the machine where you want to run a worker node. This tutorial uses the name worker1.

7. Run the command produced by the *docker swarm init* output from the Create a swarm tutorial step to create a worker node joined to the existing swarm:

```
$ docker swarm join \
  --token  SWMTKN-1-49nj1cmql0jkz5s954yi3oex3nedyz0fb0xx14ie39trti4wxv-8vxv8rssmk743ojn
  192.168.99.100:2377

This node joined a swarm as a worker.
```

If you don't have the command available, you can run the following command on a manager node to retrieve the join command for a worker:

```
$ docker swarm join-token worker

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

    docker swarm join \
    --token SWMTKN-1-49nj1cmql0jkz5s954yi3oex3nedyz0fb0xx14ie39trti4wxv-8vxv8rssmk743
    192.168.99.100:2377
```

8. Open a terminal and ssh into the machine where you want to run a second worker node. This tutorial uses the name worker2.
9. Run the command produced by the *docker swarm init* output from the Create a swarm tutorial step to create a second worker node joined to the existing swarm:

```
$ docker swarm join \
  --token SWMTKN-1-49nj1cmql0jkz5s954yi3oex3nedyz0fb0xx14ie39trti4wxv-8vxv8rssmk743oj
  192.168.99.100:2377

This node joined a swarm as a worker.
```

10. Open a terminal and ssh into the machine where the manager node runs and run the *docker node ls* command to see the worker nodes:

```
ID                            HOSTNAME   STATUS   AVAILABILITY   MANAGER STATUS
03g1y59jwfg7cf99w4lt0f662     worker2    Ready    Active
9j68exjopxe7wfl6yuxml7a7j     worker1    Ready    Active
dxn1zf6l61qsb1josjja83ngz *   manager1   Ready    Active         Leader
```

The MANAGER column identifies the manager nodes in the swarm. The empty status in this column for worker1 and worker2 identifies them as worker nodes.

Swarm management commands like *docker node ls* only work on manager nodes.