

Getting started with swarm mode

Estimated reading time: 3 minutes

This tutorial introduces you to the features of Docker Engine Swarm mode. You may want to familiarize yourself with the key concepts (</engine/swarm/key-concepts/>) before you begin.

The tutorial guides you through the following activities:

- initializing a cluster of Docker Engines in swarm mode
- adding nodes to the swarm
- deploying application services to the swarm
- managing the swarm once you have everything running

This tutorial uses Docker Engine CLI commands entered on the command line of a terminal window.

If you are brand new to Docker, see [About Docker Engine \(/engine/\)](/engine/).

Set up

To run this tutorial, you need the following:

- three Linux hosts which can communicate over a network, with Docker installed
- the IP address of the manager machine
- open ports between the hosts

Three networked host machines

This tutorial requires 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. Check out [Getting started - Swarms \(/get-started/swarm-deploy/#prerequisites\)](/get-started/swarm-deploy/#prerequisites) for one possible set-up for the hosts.

One of these machines is a manager (called `manager1`) and two of them are workers (`worker1` and `worker2`).

Note: You can follow many of the tutorial steps to test single-node swarm as well, in which case you need only one host. Multi-node commands do not work, but you can initialize a swarm, create services, and scale them.

INSTALL DOCKER ENGINE ON LINUX MACHINES

If you are using Linux based physical computers or cloud-provided computers as hosts, simply follow the Linux install instructions ([/engine/install/](#)) for your platform. Spin up the three machines, and you are ready. You can test both single-node and multi-node swarm scenarios on Linux machines.

USE DOCKER DESKTOP FOR MAC OR DOCKER DESKTOP FOR WINDOWS

Alternatively, install the latest Docker Desktop for Mac ([/docker-for-mac/](#)) or Docker Desktop for Windows ([/docker-for-windows/](#)) application on one computer. You can test both single-node and multi-node swarm from this computer, but you need to use Docker Machine to test the multi-node scenarios.

- You can use Docker Desktop for Mac or Windows to test *single-node* features of swarm mode, including initializing a swarm with a single node, creating services, and scaling services.
- Currently, you cannot use Docker Desktop for Mac or Docker Desktop for Windows alone to test a *multi-node* swarm, but many examples are applicable to a single-node Swarm setup.

The IP address of the manager machine

The IP address must be assigned to a network interface available to the host operating system. All nodes in the swarm need to connect to the manager at the IP address.

Because other nodes contact the manager node on its IP address, you should use a fixed IP address.

You can run `ifconfig` on Linux or macOS to see a list of the available network interfaces.

If you are using Docker Machine, you can get the manager IP with either `docker-machine ls` or `docker-machine ip <MACHINE-NAME>` — for example, `docker-machine ip manager1` .

The tutorial uses `manager1` : `192.168.99.100` .

Open protocols and ports between the hosts

The following ports must be available. On some systems, these ports are open by default.

- **TCP port 2377** for cluster management communications
- **TCP and UDP port 7946** for communication among nodes
- **UDP port 4789** for overlay network traffic

If you plan on creating an overlay network with encryption (`--opt encrypted`), you also need to ensure **ip protocol 50 (ESP)** traffic is allowed.

What's next?

After you have set up your environment, you are ready to create a swarm ([/engine/swarm/swarm-tutorial/create-swarm/](#)).

[tutorial \(/search/?q=tutorial\)](#), cluster management ([/search/?q=cluster management](#)), swarm mode ([/search/?q=swarm mode](#))

