Docker Ecosystem

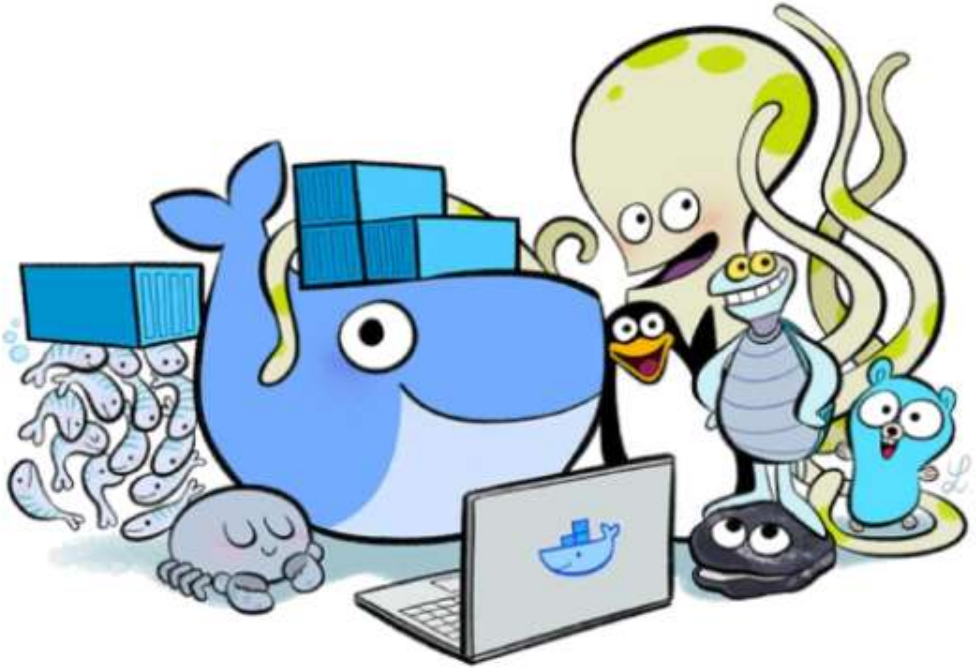
Agenda

- Introduction to Docker Ecosystem
- Docker Compose
- Docker Swarm
- Managing Containers
- Running Containers

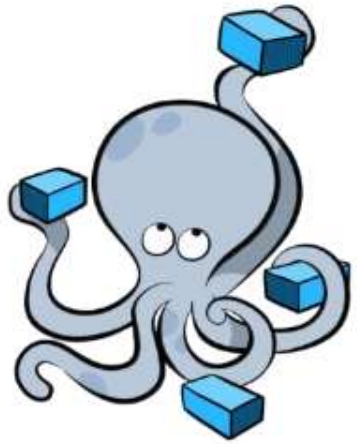
INTRODUCTION TO DOCKER ECOSYSTEM

Docker Ecosystem

- Docker in itself is extremely powerful.
- Docker is even more powerful because of its large and vibrant ecosystem.
- Let's learn about a large set of tools in the Docker ecosystem.

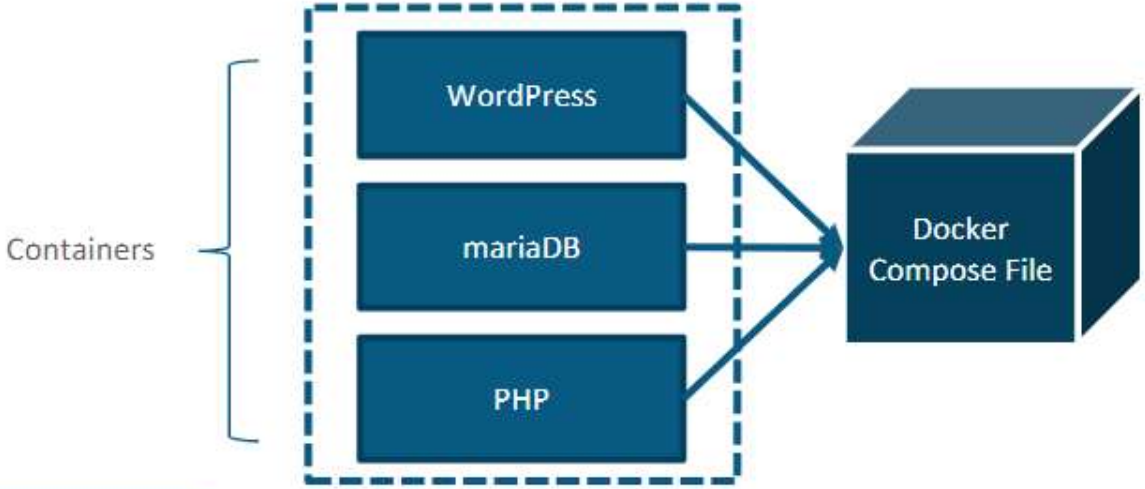


Docker Compose



Docker Compose makes it easier to configure and run applications made up of multiple containers.

For example: imagine being able to define three containers— wordpress, mysql and php—all in one YAML file and then running those three connected containers with a single command `$ docker compose up -d`



You can run these three containers with a single command

WORDPRESS USING CONTAINER

What is WordPress?

- WordPress is a free and open source blogging tool and a content management system (CMS) based on PHP and MySQL, which runs on a web hosting service it include a plugin architecture and a template system.
- It is used by more than 22.0% of the top 10 million websites as of August 2013.
- WordPress is the most popular blogging system in use on the Web, at more than 60 million websites.
- The most popular languages used are English, Spanish and Bahasa Indonesia.

Installing Docker Compose

```
root@edureka: ~
root@edureka:~# apt install python-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libexpat1-dev libpython-all-dev libpython-dev libpython2.7-dev python-all
  python-all-dev python-dev python-pip-whl python-pkg-resources
  python-setuptools python-wheel python2.7-dev
Suggested packages:
  python-setuptools-doc
The following NEW packages will be installed:
  libexpat1-dev libpython-all-dev libpython-dev libpython2.7-dev python-all
  python-all-dev python-dev python-pip python-pip-whl python-pkg-resources
  python-setuptools python-wheel python2.7-dev
0 upgraded, 13 newly installed, 0 to remove and 135 not upgraded.
Need to get 29.8 MB of archives.
After this operation, 45.1 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Step 1: Installing python-pip

Installing Docker Compose

```
root@edureka:~# pip install docker-compose
Collecting docker-compose
  Downloading docker_compose-1.13.0-py2.py3-none-any.whl (94kB)
    100% |                                        | 102kB 505kB/s
Collecting six<2,>=1.3.0 (from docker-compose)
  Downloading six-1.10.0-py2.py3-none-any.whl
Collecting dockerpty<0.5,>=0.4.1 (from docker-compose)
  Downloading dockerpty-0.4.1.tar.gz
Collecting texttable<0.9,>=0.8.1 (from docker-compose)
  Downloading texttable-0.8.8.tar.gz
Collecting cached-property<2,>=1.2.0 (from docker-compose)
  Downloading cached_property-1.3.0-py2.py3-none-any.whl
Collecting jsonschema<3,>=2.5.1 (from docker-compose)
  Downloading jsonschema-2.6.0-py2.py3-none-any.whl
Collecting ipaddress>=1.0.16; python_version < "3.3" (from docker-compose)
  Downloading ipaddress-1.0.18-py2-none-any.whl
Collecting backports.ssl-match-hostname>=3.5; python_version < "3.5" (from docke
r-compose)
  Downloading backports.ssl_match_hostname-3.5.0.1.tar.gz
Collecting docker<3.0,>=2.2.1 (from docker-compose)
  Downl
```

Step 2: Installing docker compose

Editing Docker-Compose

```
root@edureka: ~/WordPress
root@edureka:~# mkdir WordPress
root@edureka:~# cd WordPress
root@edureka:~/WordPress# gedit docker-compose.yml
```

Step 3: Editing Docker-Compose

Docker-compose.yml

Open ▾ docker-compose.yml ~/ Save

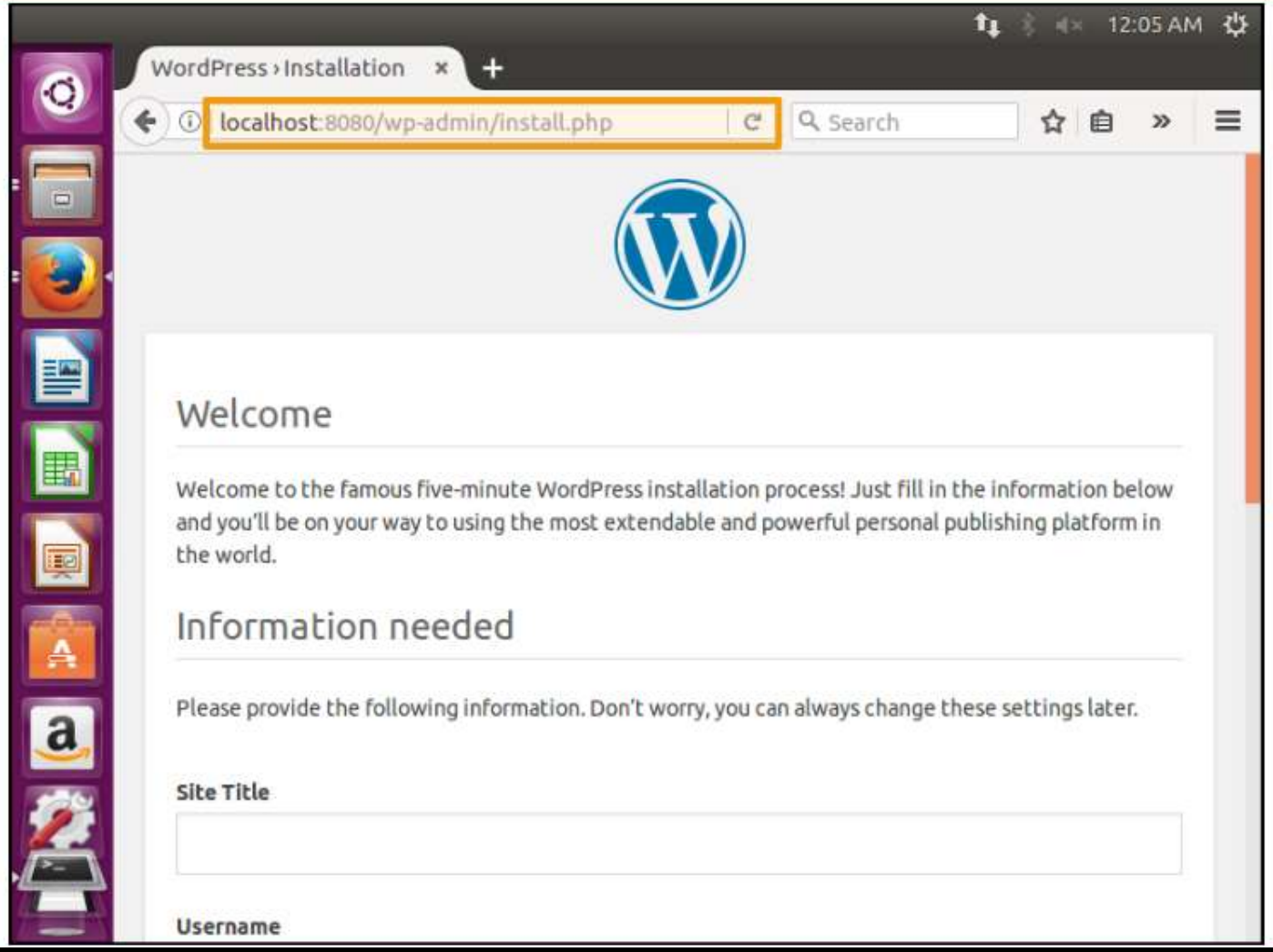
```
wordpress:
     image: wordpress
     links:
       - wordpress_db:mysql
     ports:
       - "8080:80"

wordpress_db:
 image: mariadb
 environment:
  MYSQL_ROOT_PASSWORD: edureka
phpmyadmin:
 image: corbinu/docker-phpmyadmin
 links:
  - wordpress_db:mysql
 ports:
  - 8181:80
 environment:
   MYSQL_USERNAME: root
   MYSQL_ROOT_PASSWORD: edureka
```

Step 4: Editing Docker-Compose

Executing Docker-Compose

```
root@ubuntu:~# docker-compose up -d
Pulling wordpress_db (mariadb:latest)...
latest: Pulling from library/mariadb
cd0a524342ef: Already exists
d9c95f06c17e: Already exists
46b2d578f59a: Already exists
10fbc2bcc6e9: Already exists
c5b600f068c4: Pull complete
4bca4fbb56d8: Pull complete
b9e6f929873a: Pull complete
d6a107bfa79e: Pull complete
535efcd897c3: Downloading [==================================================> ] 72.99 M
B/74.24 MB73: Download complete
cfe3cdae1993: Download complete
69f22759c937: Download complete
B/119 B
```

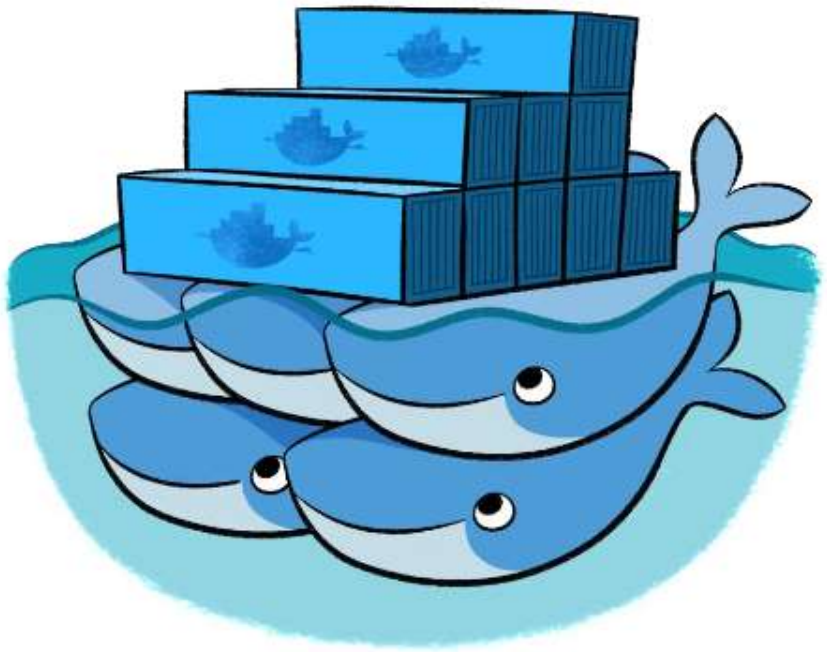
WordPress : LocalHost



DOCKER SWARM

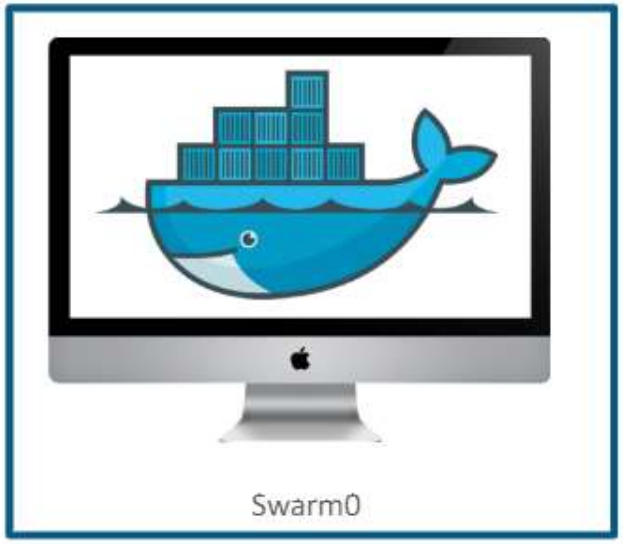
Docker Swarm

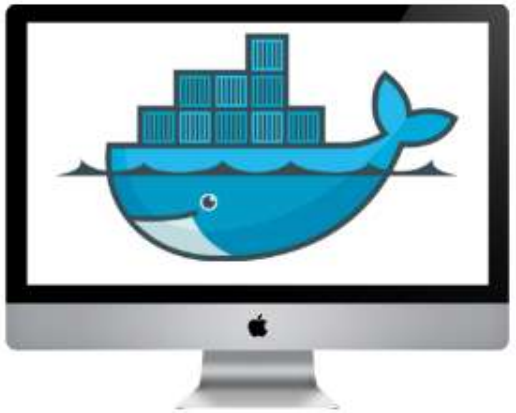
- Docker Swarm is a cluster of machine all running docker which provide scalable and reliable platform to run many containers.
- With Swarm, IT administrators and developers can establish and manage a cluster of Docker nodes as a single virtual system.

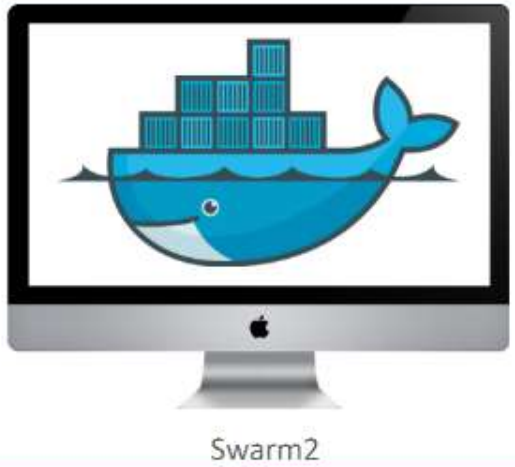


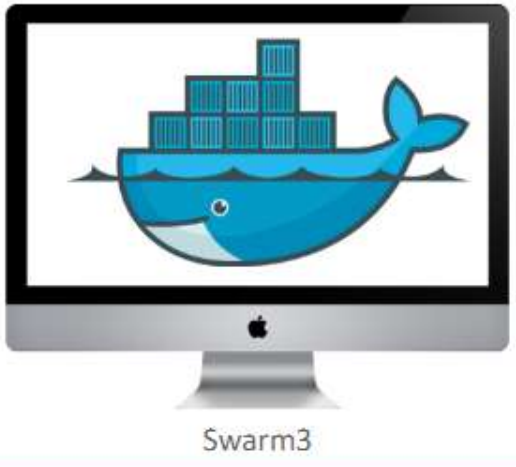
Docker Swarm : Managers and Workers

- Every Swarm has atleast one manger (Generally, the one which is initialized first)
- Port 2377 is the default port.
- Managers:
  - Swarm 0
- Workers :
  - Swarm 1
  - Swarm 2
  - Swarm 3









Starting Container on a Cluster with Docker Swarm

```
root@docker:~# docker swarm init --advertise-addr 10.0.0.1
Swarm initialized: current node (ufhxx65x5tkzthl9t3vt2ph8g) is now a manager.

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

    docker swarm join \
    --token SWMTKN-1-5ornalooqp74x5woorv6dfhe3k7rj5xf0nuyr3qh83pfu8juey-9o0nc8kmqt3r4ldz6u9ocedp7 \
    10.0.0.1:2377

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

root@docker:~#
```

- Open another VM on the same host and paste this command to join it as a worker node
- You can also join as worker mode by using the command "docker join <master-ip:port>"

- While creating the Swarm Cluster, we have to designate a node manager. For the above example, we'll be using a host by the name of **docker** as a node manager
- You can see the status of connected node by the command **`docker node ls`**