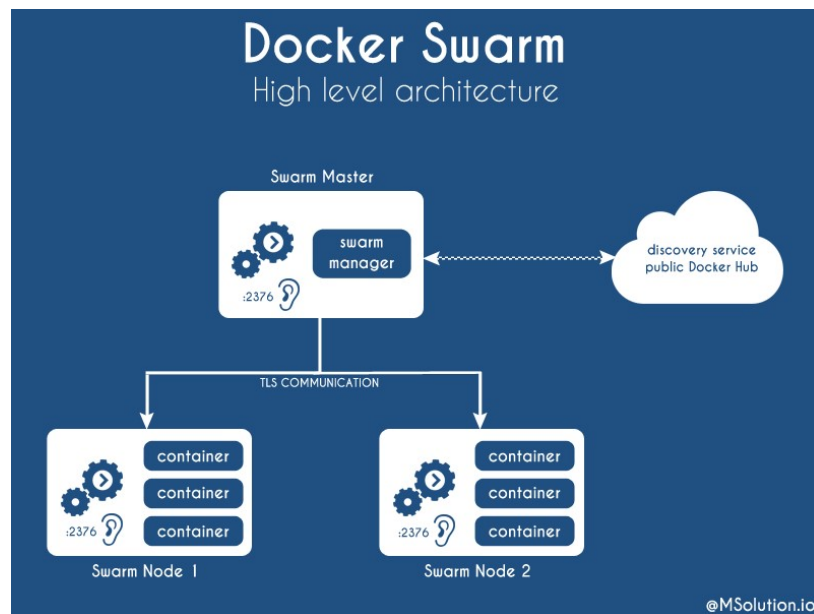


Docker swarm



27

27

Docker swarm command

Usage: `docker swarm COMMAND`

Manage Swarm

Options:

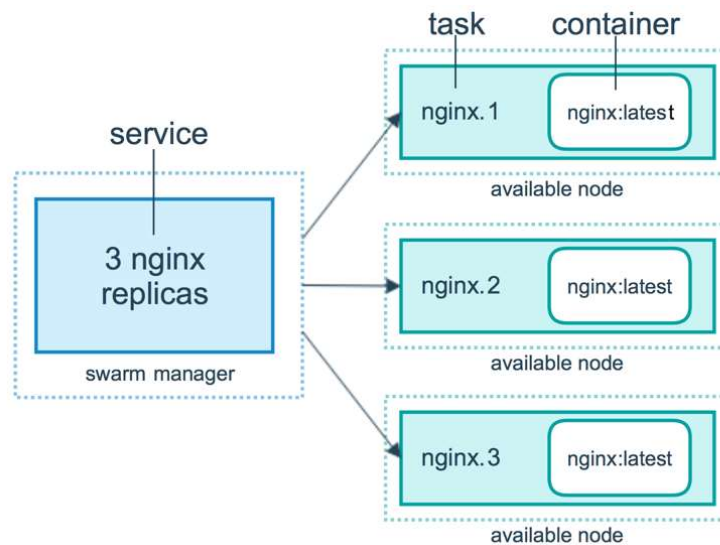
Commands:

<code>ca</code>	Display and rotate the root CA
<code>init</code>	Initialize a swarm
<code>join</code>	Join a swarm as a node and/or manager
<code>join-token</code>	Manage join tokens
<code>leave</code>	Leave the swarm
<code>unlock</code>	Unlock swarm
<code>unlock-key</code>	Manage the unlock key
<code>update</code>	Update the swarm

28

28

Docker service



29

29

Docker service command

```
Usage: docker service COMMAND

Manage services

Options:

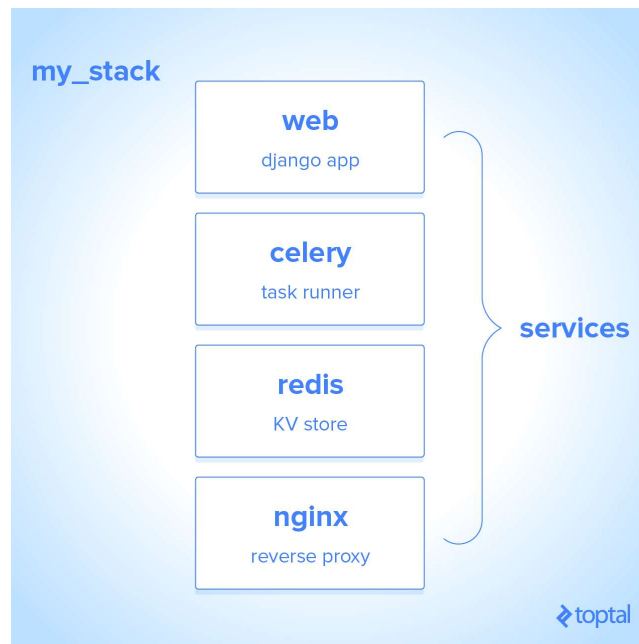
Commands:
  create      Create a new service
  inspect     Display detailed information on one or more services
  logs        Fetch the logs of a service or task
  ls          List services
  ps          List the tasks of one or more services
  rm          Remove one or more services
  rollback    Revert changes to a service's configuration
  scale       Scale one or multiple replicated services
  update      Update a service
```

30

30

2

Docker Stack



31

31

docker Stack command

```
Usage:  docker stack COMMAND

Manage Docker stacks

Options:

Commands:
  deploy  Deploy a new stack or update an existing stack
  ls      List stacks
  ps      List the tasks in the stack
  rm      Remove one or more stacks
  services List the services in the stack
```

32

32

docker-compose.yml

```
version: '3'

services:
  web:
    image: henaras/simpleweb:2
    build: .
    ports:
      - "8000:8000"
    deploy:
      resources: #Note
        limits:
          cpus: '0.25'
          memory: 256M
  redis:
    image: redis:alpine
    deploy:
      resources:
        limits:
          cpus: '0.25'
          memory: 256M
```

33

33

Dockerfile

```
# Use an official Python runtime as a parent image
FROM python:2.7-slim

# Set the working directory to /app
WORKDIR /app

# Copy the current directory contents into the container at /app
ADD . /app

# Install any needed packages specified in requirements.txt
RUN pip install --trusted-host pypi.python.org -r requirements.txt

# Make port 80 available to the world outside this container
EXPOSE 80

# Define environment variable
ENV NAME World

# Run app.py when the container launches
CMD ["python", "app.py"]
```

34

34

4