1. Docker Service is the set of operations used to create Docker services from given docker images having applications with embedded run time environment.

```
osgdev@TG-DevOps-OS004:~/dockerlab$ docker service

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

Run 'docker service COMMAND --help' for more information on a command.
```

Note: All docker service operations runs only on the swarm manager node. Hence a swarm must be creted.

```
osgdev@TG-DevOps-OS004:~/dockerlab$ docker service ls
Error response from daemon: This node is not a swarm manager. Use "docker
swarm init" or "docker swarm join" to connect this node to swarm and try
again.

osgdev@TG-DevOps-OS004:~/dockerlab$ docker swarm init
Swarm initialized: current node (ynebhavl86bcio1ff2vd94m67) is now a
manager.

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

    docker swarm join --token SWMTKN-1-
5md523rebs2ssuhg34re26n9m1l1uelz92hq7c1ffe3t9n5kig-
7zy3tfky3qhdfd8avfgqavc9h 10.199.0.104:2377
```

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

```
osgdev@TG-DevOps-OS004:~/dockerlab$ docker service ls
ID                  NAME                MODE                REPLICAS
IMAGE               PORTS
```

---

2. Create a service:

```
osgdev@TG-DevOps-OS004:~/dockerlab$ docker service create --name sayhello
--replicas 2 sayhello
image sayhello:latest could not be accessed on a registry to record
its digest. Each node will access sayhello:latest independently,
possibly leading to different nodes running different
versions of the image.

r0vuvj5v1oau63i9l33pvblcw
overall progress: 2 out of 2 tasks
1/2: running
2/2: running
verify: Service converged


osgdev@TG-DevOps-OS004:~/dockerlab$ docker service ls
ID                  NAME                MODE                REPLICAS
IMAGE               PORTS
r0vuvj5v1oau        sayhello            replicated          2/2
sayhello:latest

osgdev@TG-DevOps-OS004:~/dockerlab$ docker service ps sayhello
ID                  NAME                IMAGE               NODE
DESIRED STATE       CURRENT STATE            ERROR               PORTS
x8tix7mjro77        sayhello.1          sayhello:latest     TG-DevOps-
OS004     Running             Running 29 seconds ago
2nn9ai1nneif        sayhello.2          sayhello:latest     TG-DevOps-
OS004     Running             Running 25 seconds ago

osgdev@TG-DevOps-OS004:~/dockerlab$ docker container ls
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS               NAMES
958253c413a1        sayhello:latest     "python app.py"     About a
minute ago   Up About a minute   80/tcp
sayhello.2.2nn9ai1nneifjvrp1521vsqnq
73946d27b081        sayhello:latest     "python app.py"     About a
minute ago   Up About a minute   80/tcp
sayhello.1.x8tix7mjro77ttzq1oswwqj8b
```

---

3. Inspecting the service:

```
osgdev@TG-DevOps-OS004:~/dockerlab$ docker service inspect --pretty
sayhello

ID:         r0vuvj5v1oau63i9l33pvblcw
Name:       sayhello
Service Mode:    Replicated
 Replicas:  2
Placement:
UpdateConfig:
 Parallelism:    1
 On failure:     pause
 Monitoring Period: 5s
 Max failure ratio: 0
 Update order:      stop-first
RollbackConfig:
 Parallelism:    1
 On failure:     pause
 Monitoring Period: 5s
 Max failure ratio: 0
 Rollback order:    stop-first
ContainerSpec:
 Image:          sayhello:latest
Resources:
Endpoint Mode:   vip
```

---

4. Scaling the service:

```
osgdev@TG-DevOps-OS004:~/dockerlab$ docker service scale sayhello=5
sayhello scaled to 5
overall progress: 5 out of 5 tasks
1/5: running
2/5: running
3/5: running
4/5: running
5/5: running
verify: Service converged

osgdev@TG-DevOps-OS004:~/dockerlab$ docker service ps sayhello
ID                  NAME                IMAGE               NODE
DESIRED STATE       CURRENT STATE           ERROR               PORTS
x8tix7mjro77        sayhello.1          sayhello:latest     TG-DevOps-
OS004     Running             Running 5 minutes ago
2nn9ai1nneif        sayhello.2          sayhello:latest     TG-DevOps-
OS004     Running             Running 5 minutes ago
u40090dcc3n6        sayhello.3          sayhello:latest     TG-DevOps-
OS004     Running             Running 14 seconds ago
1yl5eyboy1lz        sayhello.4          sayhello:latest     TG-DevOps-
OS004     Running             Running 12 seconds ago
c9oon6h16usm        sayhello.5          sayhello:latest     TG-DevOps-
OS004     Running             Running 15 seconds ago

osgdev@TG-DevOps-OS004:~/dockerlab$ docker service scale sayhello=3
sayhello scaled to 3
overall progress: 3 out of 3 tasks
1/3: running
2/3: running
3/3: running
verify: Service converged
```

```
osgdev@TG-DevOps-OS004:~/dockerlab$ docker service ps sayhello
ID                  NAME                IMAGE               NODE
DESIRED STATE       CURRENT STATE                   ERROR
PORTS
2nn9ai1nneif        sayhello.2          sayhello:latest     TG-DevOps-
OS004     Running             Running 6 minutes ago
1yl5eyboy1lz        sayhello.4          sayhello:latest     TG-DevOps-
OS004     Running             Running about a minute ago
c9oon6h16usm        sayhello.5          sayhello:latest     TG-DevOps-
OS004     Running             Running about a minute ago

osgdev@TG-DevOps-OS004:~/dockerlab$ docker container ls
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS               NAMES
82acda6220ae        sayhello:latest     "python app.py"     2 minutes ago
Up 2 minutes        80/tcp
sayhello.4.1yl5eyboy1lztc7ysio1p0ebl
ed43ffe43240        sayhello:latest     "python app.py"     2 minutes ago
Up 2 minutes        80/tcp
sayhello.5.c9oon6h16usmh8l767tfptsuz
958253c413a1        sayhello:latest     "python app.py"     7 minutes ago
Up 7 minutes        80/tcp
sayhello.2.2nn9ai1nneifjvrp1521vsqnq
```

5. Demonstrating Self-healing. Since each instance of service is running inside a container, try stopping a container.

```
osgdev@TG-DevOps-OS004:~/dockerlab$ docker container ls
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS               NAMES
82acda6220ae        sayhello:latest     "python app.py"     2 minutes ago
Up 2 minutes        80/tcp
sayhello.4.1yl5eyboy1lztc7ysio1p0ebl
ed43ffe43240        sayhello:latest     "python app.py"     2 minutes ago
Up 2 minutes        80/tcp
sayhello.5.c9oon6h16usmh8l767tfptsuz
958253c413a1        sayhello:latest     "python app.py"     7 minutes ago
Up 7 minutes        80/tcp
sayhello.2.2nn9ai1nneifjvrp1521vsqnq

osgdev@TG-DevOps-OS004:~/dockerlab$ docker container stop 82acda6220ae
82acda6220ae

osgdev@TG-DevOps-OS004:~/dockerlab$ docker service ps sayhello
ID                  NAME                IMAGE               NODE
DESIRED STATE       CURRENT STATE                ERROR
PORTS
2nn9ai1nneif        sayhello.2          sayhello:latest     TG-DevOps-
OS004     Running             Running 9 minutes ago
m9ie39au4uqe        sayhello.4          sayhello:latest     TG-DevOps-
OS004     Ready               Preparing 2 seconds ago
```

```
1yl5eyboy1lz         \_ sayhello.4      sayhello:latest     TG-DevOps-
OS004     Shutdown            Failed 3 seconds ago      "task: non-zero
exit (137)"
c9oon6h16usm        sayhello.5          sayhello:latest     TG-DevOps-
OS004     Running             Running 4 minutes ago

osgdev@TG-DevOps-OS004:~/dockerlab$ docker service ps sayhello
ID                  NAME                IMAGE               NODE
DESIRED STATE       CURRENT STATE             ERROR
PORTS
2nn9ai1nneif        sayhello.2          sayhello:latest     TG-DevOps-
OS004     Running             Running 9 minutes ago
m9ie39au4uqe        sayhello.4          sayhello:latest     TG-DevOps-
OS004     Running             Preparing 8 seconds ago
1yl5eyboy1lz         \_ sayhello.4      sayhello:latest     TG-DevOps-
OS004     Shutdown            Failed 8 seconds ago      "task: non-zero
exit (137)"
c9oon6h16usm        sayhello.5          sayhello:latest     TG-DevOps-
OS004     Running             Running 4 minutes ago

osgdev@TG-DevOps-OS004:~/dockerlab$ docker container ls
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS               NAMES
722596fa9c31        sayhello:latest     "python app.py"     51 seconds
ago      Up 50 seconds       80/tcp
sayhello.4.m9ie39au4uqeapyr2lpkagb1o
ed43ffe43240        sayhello:latest     "python app.py"     5 minutes ago
Up 5 minutes        80/tcp
sayhello.5.c9oon6h16usmh8l767tfptsuz
958253c413a1        sayhello:latest     "python app.py"     10 minutes
ago      Up 10 minutes       80/tcp
sayhello.2.2nn9ai1nneifjvrp1521vsqnq

osgdev@TG-DevOps-OS004:~/dockerlab$ docker container ls -a
CONTAINER ID        IMAGE               COMMAND                  CREATED
STATUS                        PORTS               NAMES
722596fa9c31        sayhello:latest     "python app.py"          4
minutes ago       Up 4 minutes                  80/tcp
sayhello.4.m9ie39au4uqeapyr2lpkagb1o
82acda6220ae        sayhello:latest     "python app.py"          8
minutes ago       Exited (137) 4 minutes ago
sayhello.4.1yl5eyboy1lztc7ysio1p0ebl
ed43ffe43240        sayhello:latest     "python app.py"          8
minutes ago       Up 8 minutes                  80/tcp
sayhello.5.c9oon6h16usmh8l767tfptsuz
958253c413a1        sayhello:latest     "python app.py"          13
minutes ago      Up 13 minutes                 80/tcp
sayhello.2.2nn9ai1nneifjvrp1521vsqnq
```

6. Made a small change to application file "app.py" to get an updated image. Adding additional line to html "<h1>Adding a small change</h1>"

```
osgdev@TG-DevOps-OS004:~/dockerlab/python$ cat app.py
from flask import Flask
from redis import Redis, RedisError
import os
import socket

# Connect to Redis
redis = Redis(host="redis", db=0, socket_connect_timeout=2,
socket_timeout=2)

app = Flask(__name__)

@app.route("/")
def hello():
    try:
        visits = redis.incr("counter")
    except RedisError:
        visits = "<i>cannot connect to Redis, counter disabled</i>"

    html = "<h3>Hello {name}!</h3>" \
           "<h1>Adding a small change</h1>" \
           "<b>Hostname:</b> {hostname}<br/>" \
           "<b>Visits:</b> {visits}"
    return html.format(name=os.getenv("NAME", "world"),
hostname=socket.gethostname(), visits=visits)

if __name__ == "__main__":
    app.run(host='0.0.0.0', port=80)
```

Keeping the otherfiles in the folder as is, create a new image "sayhello_new"

```
osgdev@TG-DevOps-OS004:~/dockerlab/python$ docker build -t sayhello_new .
Sending build context to Docker daemon  4.608kB
Step 1/7 : FROM python:2.7-slim
 ---> b16fde09c92c
Step 2/7 : WORKDIR /app
 ---> Using cache
 ---> 77fae6c6b4a3
Step 3/7 : ADD . /app
 ---> 71c992d0e5b7
Step 4/7 : RUN pip install --trusted-host pypi.python.org -r
requirements.txt
 ---> Running in 360d4e8e00a0
Collecting Flask (from -r requirements.txt (line 1))
  Downloading
https://files.pythonhosted.org/packages/77/32/e3597cb19ffffe724ad4bf0beca
4153419918e7fa4ba6a34b04ee4da3371/Flask-0.12.2-py2.py3-none-any.whl
(83kB)
Collecting Redis (from -r requirements.txt (line 2))
  Downloading
https://files.pythonhosted.org/packages/3b/f6/7a76333cf0b9251ecf49efff635
015171843d9b977e4ffcf59f9c4428052/redis-2.10.6-py2.py3-none-any.whl
(64kB)
```

```
Collecting itsdangerous>=0.21 (from Flask->-r requirements.txt (line 1))
  Downloading
https://files.pythonhosted.org/packages/dc/b4/a60bcdba945c00f6d608d897513
1ab3f25b22f2bcfe1dab221165194b2d4/itsdangerous-0.24.tar.gz (46kB)
Collecting Jinja2>=2.4 (from Flask->-r requirements.txt (line 1))
  Downloading
https://files.pythonhosted.org/packages/7f/ff/ae64bacdfc95f27a016a7bed8e8
686763ba4d277a78ca76f32659220a731/Jinja2-2.10-py2.py3-none-any.whl
(126kB)
Collecting Werkzeug>=0.7 (from Flask->-r requirements.txt (line 1))
  Downloading
https://files.pythonhosted.org/packages/20/c4/12e3e56473e52375aa29c4764e7
0d1b8f3efa6682bef8d0aae04fe335243/Werkzeug-0.14.1-py2.py3-none-any.whl
(322kB)
Collecting click>=2.0 (from Flask->-r requirements.txt (line 1))
  Downloading
https://files.pythonhosted.org/packages/34/c1/8806f99713ddb993c5366c362b2
f908f18269f8d792aff1abfd700775a77/click-6.7-py2.py3-none-any.whl (71kB)
Collecting MarkupSafe>=0.23 (from Jinja2>=2.4->Flask->-r requirements.txt
(line 1))
  Downloading
https://files.pythonhosted.org/packages/4d/de/32d741db316d8fdb7680822dd37
001ef7a448255de9699ab4bfcbdf4172b/MarkupSafe-1.0.tar.gz
Building wheels for collected packages: itsdangerous, MarkupSafe
  Running setup.py bdist_wheel for itsdangerous: started
  Running setup.py bdist_wheel for itsdangerous: finished with status
'done'
  Stored in directory:
/root/.cache/pip/wheels/2c/4a/61/5599631c1554768c6290b08c02c72d7317910374
ca602ff1e5
  Running setup.py bdist_wheel for MarkupSafe: started
  Running setup.py bdist_wheel for MarkupSafe: finished with status
'done'
  Stored in directory:
/root/.cache/pip/wheels/33/56/20/ebe49a5c612fffe1c5a632146b16596f9e646767
68661e4e46
Successfully built itsdangerous MarkupSafe
Installing collected packages: itsdangerous, MarkupSafe, Jinja2,
Werkzeug, click, Flask, Redis
Successfully installed Flask-0.12.2 Jinja2-2.10 MarkupSafe-1.0 Redis-
2.10.6 Werkzeug-0.14.1 click-6.7 itsdangerous-0.24
You are using pip version 9.0.3, however version 10.0.1 is available.
You should consider upgrading via the 'pip install --upgrade pip'
command.
Removing intermediate container 360d4e8e00a0
 ---> 7a35d70c8a98
Step 5/7 : EXPOSE 80
 ---> Running in a460af9afe98
Removing intermediate container a460af9afe98
 ---> 0f2cae6842d6
Step 6/7 : ENV NAME World
 ---> Running in acaa9eda48a7
Removing intermediate container acaa9eda48a7
 ---> 7d485b62b746
```

```
Step 7/7 : CMD ["python", "app.py"]
 ---> Running in 7deea850c91c
Removing intermediate container 7deea850c91c
 ---> 9dfab9a7d6a7
Successfully built 9dfab9a7d6a7
Successfully tagged sayhello_new:latest

osgdev@TG-DevOps-OS004:~/dockerlab/python$ docker image ls
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE
sayhello_new        latest              9dfab9a7d6a7        24 seconds
ago      151MB
sayhello            latest              02b691805eb2        10 days ago
150MB
```

---

7. Update the existing service sayhello which is using the image sayhello to sayhello_new. Since the port forwarding is not done in the existing service let us remove the existing service and create new service with port forwarding.

```
osgdev@TG-DevOps-OS004:~/dockerlab$ docker service ls
ID                  NAME                MODE                REPLICAS
IMAGE               PORTS
r0vuvj5v1oau        sayhello            replicated          3/3
sayhello:latest

osgdev@TG-DevOps-OS004:~/dockerlab$ docker service ps sayhello
ID                  NAME                IMAGE               NODE
DESIRED STATE       CURRENT STATE            ERROR
PORTS
2nn9ai1nneif        sayhello.2          sayhello:latest     TG-DevOps-
OS004     Running             Running 40 minutes ago
m9ie39au4uqe        sayhello.4          sayhello:latest     TG-DevOps-
OS004     Running             Running 30 minutes ago
1yl5eyboy1lz         \_ sayhello.4      sayhello:latest     TG-DevOps-
OS004     Shutdown            Failed 30 minutes ago    "task: non-zero
exit (137)"
c9oon6h16usm        sayhello.5          sayhello:latest     TG-DevOps-
OS004     Running             Running 34 minutes ago

osgdev@TG-DevOps-OS004:~/dockerlab$ docker service rm sayhello
sayhello

osgdev@TG-DevOps-OS004:~/dockerlab$ docker service create --name
sayhello2 --replicas 1 --publish 11022:80 sayhello
image sayhello:latest could not be accessed on a registry to record
its digest. Each node will access sayhello:latest independently,
possibly leading to different nodes running different
versions of the image.

m51sey9wnzmaeulvueab9dz92
overall progress: 1 out of 1 tasks
```

```
1/1: running
verify: Service converged

osgdev@TG-DevOps-OS004:~/dockerlab$ docker service ls
ID                  NAME                MODE                REPLICAS
IMAGE               PORTS
m51sey9wnzma        sayhello2           replicated          1/1
sayhello:latest     *:11022->80/tcp

osgdev@TG-DevOps-OS004:~/dockerlab$ docker service update --image
sayhello_new sayhello2
image sayhello_new:latest could not be accessed on a registry to record
its digest. Each node will access sayhello_new:latest independently,
possibly leading to different nodes running different
versions of the image.

sayhello2
overall progress: 1 out of 1 tasks
1/1: running
verify: Service converged

osgdev@TG-DevOps-OS004:~/dockerlab$ docker service ls
ID                  NAME                MODE                REPLICAS
IMAGE                 PORTS
m51sey9wnzma        sayhello2           replicated          1/1
sayhello_new:latest   *:11022->80/tcp

Check the output on browser:
http://<IP address>:11022
```

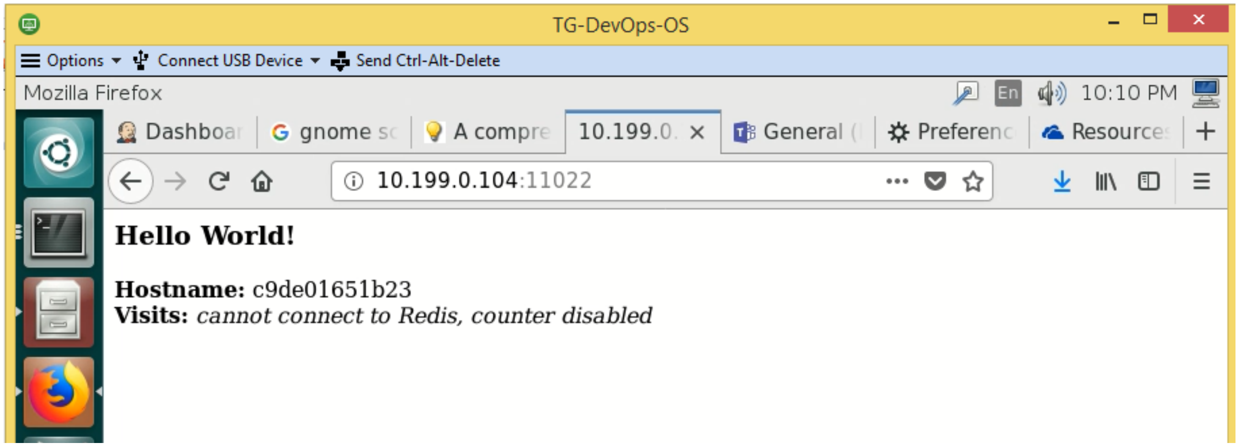
---

8. Rolling back the changes. Observe the change made look ackward on the screen, hence we decided to rollback.

```
osgdev@TG-DevOps-OS004:~/dockerlab$ docker service rollback sayhello2
sayhello2
rollback: manually requested rollback
overall progress: rolling back update: 1 out of 1 tasks
1/1: running
verify: Service converged

osgdev@TG-DevOps-OS004:~/dockerlab$ docker service ls
ID                  NAME                MODE                REPLICAS
IMAGE               PORTS
m51sey9wnzma        sayhello2           replicated          1/1
sayhello:latest     *:11022->80/tcp

Check the output on browser:
http://<IP address>:11022
```



Hello World!

Hostname: c9de01651b23
Visits: *cannot connect to Redis, counter disabled*

9. Docker Stack can hold multiple services in multiple containers and can connect them using overlay network.

```
osgdev@TG-DevOps-OS004:~/dockerlab$ docker stack

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

Run 'docker stack COMMAND --help' for more information on a command.
```

10. Create a service stack with a single service as we did in previous exercise. This is better to be used with multiple services listed in docker compose YAML file.

```
osgdev@TG-DevOps-OS004:~/dockerlab$ docker stack deploy -c docker-
compose-sayhello.yaml sayhello_stack
Creating network sayhello_stack_webnet
Creating service sayhello_stack_web

osgdev@TG-DevOps-OS004:~/dockerlab$ docker stack ls
NAME                SERVICES
sayhello_stack      1

osgdev@TG-DevOps-OS004:~/dockerlab$ docker stack ps sayhello_stack
```

```
ID                  NAME                   IMAGE               NODE
DESIRED STATE       CURRENT STATE            ERROR               PORTS
ta9qkz2d7alm        sayhello_stack_web.1   sayhello:latest     TG-DevOps-
OS004     Running             Running 41 seconds ago
qnj9a9gdrgxb        sayhello_stack_web.2   sayhello:latest     TG-DevOps-
OS004     Running             Running 40 seconds ago
ymnmg4wh6ydm        sayhello_stack_web.3   sayhello:latest     TG-DevOps-
OS004     Running             Running 41 seconds ago
wtxw5mxoeuz9        sayhello_stack_web.4   sayhello:latest     TG-DevOps-
OS004     Running             Running 41 seconds ago
p6o2ov88v7zt        sayhello_stack_web.5   sayhello:latest     TG-DevOps-
OS004     Running             Running 42 seconds ago

osgdev@TG-DevOps-OS004:~/dockerlab$ docker stack services sayhello_stack
ID                  NAME                MODE                REPLICAS
IMAGE               PORTS
navdfrg3nw5z        sayhello_stack_web   replicated          5/5
sayhello:latest     *:11055->80/tcp
osgdev@TG-DevOps-OS004:~/dockerlab$ docker stack rm sayhello_stack
Removing service sayhello_stack_web
Removing network sayhello_stack_webnet
```

11. Docker swarm manages a cluster of servers to deploy Docker Stacks, Services in the form of containers.

```
osgdev@TG-DevOps-OS004:~/dockerlab$ docker swarm

Usage:      docker swarm COMMAND

Manage Swarm

Options:


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

Run 'docker swarm COMMAND --help' for more information on a command.
```

12. You need additional machines which can be pinged from this Topgear machine to create the cluster. You would need join-token to add worker nodes to the swarm.

```
osgdev@TG-DevOps-OS004:~/dockerlab$ docker swarm leave --force
Node left the swarm.
```

FOLLOWING HANDSON CANNOT BE DONE UNLESS YOU HAVE ADDITIONAL MACHINES CONNECTED TO YOUR TOPGEAR MACHINE.
THIS PART MAY BE USED AS REFERENCE ONLY IF YOU HAVE SINGLE TOPGEAR MACHINE.
IP ADDRESS of the TOPGEAR Machine used in this exercise is: 10.199.0.104

Command below initializes the current machine as Swarm Manager and provide the token for worker nodes to join.

```
osgdev@TG-DevOps-OS004:~/dockerlab$ docker swarm init --advertise-addr
10.199.0.104
Swarm initialized: current node (kdmjazcolih0w01uzp0z11dqf) is now a
manager.

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

    docker swarm join --token SWMTKN-1-
3kdv2bpwdkjv840avaojkwopesjgm1t6x6yqk9bw3rujo2x3kf-
dqq49eo8dwwh9d30uh6x7dxo9 10.199.0.104:2377

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

Command specified in the response of above command must be used by the worker nodes to join the swarm:

```
docker swarm join --token SWMTKN-1-
3kdv2bpwdkjv840avaojkwopesjgm1t6x6yqk9bw3rujo2x3kf-
dqq49eo8dwwh9d30uh6x7dxo9 10.199.0.104:2377
```

In case if the command is missed out, you may again generate the token even after the swarm is initialized.

```
osgdev@TG-DevOps-OS004:~/dockerlab$ docker swarm join-token worker
To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-
3kdv2bpwdkjv840avaojkwopesjgm1t6x6yqk9bw3rujo2x3kf-
dqq49eo8dwwh9d30uh6x7dxo9 10.199.0.104:2377
```

---

13. Docker Secret is used to make any data available in encrypted form

```
osgdev@TG-DevOps-OS004:~/dockerlab$ docker secret
```

```
Usage:     docker secret COMMAND

Manage Docker secrets

Options:


Commands:
  create      Create a secret from a file or STDIN as content
  inspect     Display detailed information on one or more secrets
  ls          List secrets
  rm          Remove one or more secrets

Run 'docker secret COMMAND --help' for more information on a command.
```

14. Create a secret, which may be a password, private key or any important data to be accessible only to containers/services

```
osgdev@TG-DevOps-OS004:~/dockerlab$ echo "This is a secret" | docker
secret create my_secret_data -
r0g9sncatb0tws0xn50jnzair
```

15. Create a service using tomcat:8 image and add the secret to the service.

```
osgdev@TG-DevOps-OS004:~/dockerlab$ docker service create --name
webservice --secret="my_secret_data" tomcat:8
image tomcat:8 could not be accessed on a registry to record
its digest. Each node will access tomcat:8 independently,
possibly leading to different nodes running different
versions of the image.

8o2kfr9hprdsr4xeov0w748uh
overall progress: 1 out of 1 tasks
1/1: running
verify: Service converged

osgdev@TG-DevOps-OS004:~/dockerlab$ docker service ls
ID                  NAME                MODE                REPLICAS
IMAGE               PORTS
8o2kfr9hprds        webservice          replicated          1/1
tomcat:8

osgdev@TG-DevOps-OS004:~/dockerlab$ docker service ps webservice
ID                  NAME                IMAGE               NODE
DESIRED STATE       CURRENT STATE            ERROR               PORTS
f2ye108r7jzr        webservice.1        tomcat:8            TG-DevOps-
OS004     Running             Running 32 seconds ago

osgdev@TG-DevOps-OS004:~/dockerlab$ docker container ls
```

```
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS               NAMES
669235a8eec6        tomcat:8            "catalina.sh run"   55 seconds
ago      Up 52 seconds       8080/tcp
webservice.1.f2ye108r7jzrz0xx48r8nmwq2
```

16. Interact with the container to check accessibility to the secret:

```
osgdev@TG-DevOps-OS004:~/dockerlab$ docker container exec 669235a8eec6 ls
-l /run/secrets
total 4
-r--r--r-- 1 root root 17 Apr 24 17:19 my_secret_data

osgdev@TG-DevOps-OS004:~/dockerlab$ docker container exec 669235a8eec6
cat /run/secrets/my_secret_data
This is a secret
```

17. Check whether this secret goes into image when the container is committed to extract the image

```
osgdev@TG-DevOps-OS004:~/dockerlab$ docker container commit 669235a8eec6
newtomcat
sha256:a7b131389e8349debf276590691c1f452d99e5f00f144dad92cec97ce61fb01d

osgdev@TG-DevOps-OS004:~/dockerlab$ docker container run -d newtomcat
29794907fcb1676daccf192df6fbe7831e1da8eb4731cabf45547d0f90fa2611

osgdev@TG-DevOps-OS004:~/dockerlab$ docker container ls
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS               NAMES
29794907fcb1        newtomcat           "catalina.sh run"   7 seconds ago
Up 7 seconds        8080/tcp            adoring_minsky
669235a8eec6        tomcat:8            "catalina.sh run"   3 minutes ago
Up 3 minutes        8080/tcp
webservice.1.f2ye108r7jzrz0xx48r8nmwq2

osgdev@TG-DevOps-OS004:~/dockerlab$ docker container exec 29794907fcb1 ls
-l /run/secrets
total 0
-rwxr-xr-x 1 root root 0 Apr 24 17:19 my_secret_data

osgdev@TG-DevOps-OS004:~/dockerlab$ docker container exec 29794907fcb1
cat /run/secrets/my_secret_data
```

Note: Secret is missing in the newtomcat image. Hence it is accessible only to that container.

18. Secret cannot be removed as long as the service having connectivity to secret is running.

```
osgdev@TG-DevOps-OS004:~/dockerlab$ docker secret ls
ID                          NAME                DRIVER
CREATED             UPDATED
r0g9sncatb0tws0xn50jnzair   my_secret_data                          8
minutes ago       8 minutes ago

osgdev@TG-DevOps-OS004:~/dockerlab$ docker secret rm my_secret_data
Error response from daemon: rpc error: code = InvalidArgument desc =
secret 'my_secret_data' is in use by the following service: webservice
```

19. Remove the secret connected with the service

```
osgdev@TG-DevOps-OS004:~/dockerlab$ docker service update --secret-
rm="my_secret_data" webservice
webservice
overall progress: 1 out of 1 tasks
1/1: running
verify: Service converged

osgdev@TG-DevOps-OS004:~/dockerlab$ docker container ls
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS               NAMES
40d947afeb3e        tomcat:8            "catalina.sh run"   17 seconds
ago      Up 14 seconds       8080/tcp
webservice.1.io3mtyj2gvvs04dq5mbvsuiyp
29794907fcb1        newtomcat           "catalina.sh run"   5 minutes ago
Up 5 minutes        8080/tcp            adoring_minsky

osgdev@TG-DevOps-OS004:~/dockerlab$ docker container exec 40d947afeb3e ls
/run/secrets/
ls: cannot access '/run/secrets/': No such file or directory
```

20. Once the secret is not associated with any service it can be removed.

```
osgdev@TG-DevOps-OS004:~/dockerlab$ docker service rm webservice
webservice

osgdev@TG-DevOps-OS004:~/dockerlab$ docker secret rm my_secret_data
my_secret_data
```

21.

22.

23.

24.

25.
