Module-7: Containerization using Docker Part -II

1. **Download Company's Website Files.**

```
root@docker-serv:/opt/dockerContent-# ls
Dockerfile     docker-compose.yml  prepros-6.config              work-grid.html
about.html     fonts               sass                          work.html
blog.html      images              services.html
contact.html   index.html          shop.html
css            js                  work-grid-without-text.html
root@docker-serv:/opt/dockerContent-#
```

2. **2. Written a Dockerfile for the Website.**

```
root@docker-serv:/opt/dockerContent-# cat Dockerfile
# Use the official Apache image as the base image
FROM httpd:latest

# Copy the website files into the container's Apache directory
COPY ./ /usr/local/apache2/htdocs/

# Expose port 80 for the web service
EXPOSE 80

# Configure Apache to run as the foreground process
CMD ["httpd-foreground"]

root@docker-serv:/opt/dockerContent-#
```

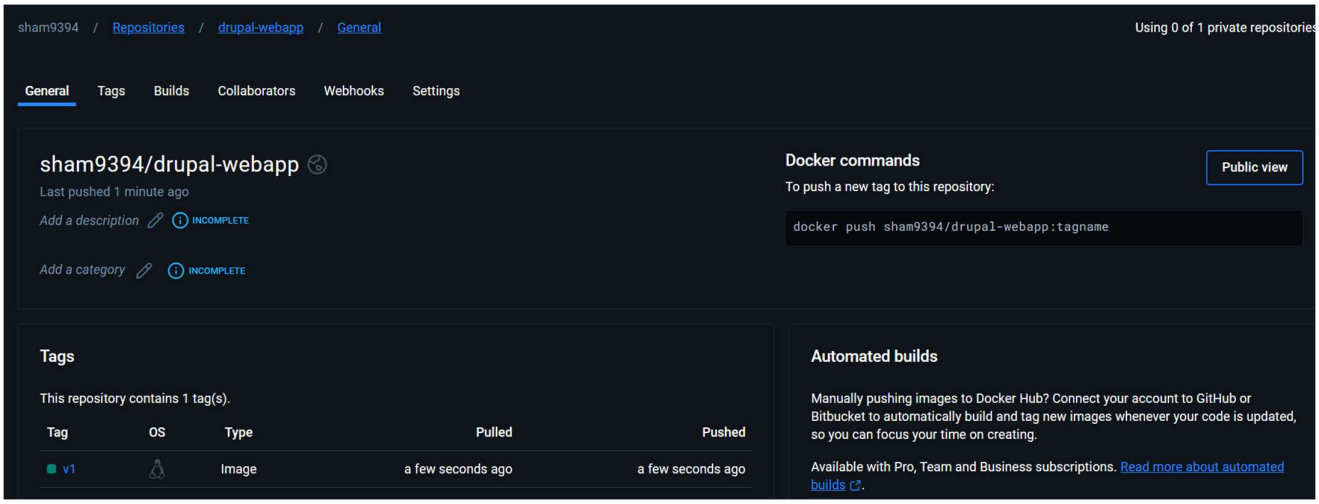
3. **Used Volumes to Store Website Data Outside the Container using docker-compose.**

```
root@docker-serv:/opt/dockerContent-# cat docker-compose.yml
version: "3.8"

services:
  website:
    image: sham9394/drupal-webapp:v1
    deploy:
      replicas: 3  # Number of replicas (containers)
      restart_policy:
        condition: on-failure
    ports:
      - "80:80"  # Expose port 80 to the host
    volumes:
      - website_data:/opt/website-data

volumes:
  website_data:
```

4. **4. Pushed the Docker Image to Docker Hub.**

```
root@docker-serv:/opt/dockerContent-# docker images
REPOSITORY                TAG       IMAGE ID       CREATED          SIZE
sham9394/drupal-webapp    v1        42cee64239d1   3 minutes ago    184MB
hello-world               latest    d2c94e258dcb   20 months ago    13.3kB
root@docker-serv:/opt/dockerContent-#
```

```
root@docker-serv:/opt/dockerContent-# docker push sham9394/drupal-webapp:v1
The push refers to repository [docker.io/sham9394/drupal-webapp]
c6d5f13bf097: Pushed
52168ee29b83: Pushed
ca565a60a706: Mounted from library/httpd
5dceadbf1901: Mounted from library/httpd
5f70bf18a086: Pushed
fa084c5dde25: Pushed
8b296f486960: Mounted from library/httpd
v1: digest: sha256:26b7bd543f5fcecd8c6d423ee94b70367867d76fe4fbe368d7679e30d2d7bce2 size:
 1784
```



5. **Create a Docker Swarm Cluster.**

```
root@docker-serv:/opt/dockerContent-# docker swarm init
Swarm initialized: current node (olxx8g3fqe0odcrod5yc3c6f2) is now a manager.

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

    docker swarm join --token SWMTKN-1-6a2i1oob5ftq9pxvhsr2sf86iwp2w0budyxar5tw6fyrthe14n
-2asvwyjmqdat07grqeu9a61sd 10.128.0.15:2377

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

Created node server.

```
root@node1:~# whoami
root
root@node1:~#
```

Added Nodes: Using the docker swarm join token to add worker nodes.

```
root@node1:~# docker swarm join --token SWMTKN-1-6a2i1oob5ftq9pxvhsr2sf86iwp2w0budyxar5tw
6fyrthe14n-2asvwyjmqdat07grqeu9a61sd 10.128.0.15:2377
This node joined a swarm as a worker.
root@node1:~#
```

```
root@docker-serv:/opt/dockerContent-# docker node ls
ID                            HOSTNAME      STATUS    AVAILABILITY   MANAGER STATUS   ENG
INE VERSION
olxx8g3fqe0odcrod5yc3c6f2 *   docker-serv   Ready     Active         Leader           27.
4.1
5ssvv650q1igyy3cyjhopnmdf     node1         Ready     Active                          27.
4.1
```

6. **Deploy Drupal Website on Swarm.**

```
root@docker-serv:/opt/dockerContent-# docker stack deploy -c docker-compose.yml drupal-st
ack
Since --detach=false was not specified, tasks will be created in the background.
In a future release, --detach=false will become the default.
Creating network drupal-stack_default
Creating service drupal-stack_website
```

7. **Verify the Deployment:**

```
root@docker-serv:/opt/dockerContent-# docker service ls
ID             NAME                    MODE         REPLICAS   IMAGE
 PORTS
vykif832azmp   drupal-stack_website    replicated   3/3        sham9394/drupal-webapp:v1
 *:80->80/tcp
```

On master:

```
root@docker-serv:/opt/dockerContent-# docker ps
CONTAINER ID   IMAGE                        COMMAND              CREATED          STATUS
      PORTS      NAMES
1700f862f3f7   sham9394/drupal-webapp:v1    "httpd-foreground"   3 minutes ago    Up 3 minu
tes    80/tcp     drupal-stack_website.2.9xz67es5uvi53uzpsavruoxdw
root@docker-serv:/opt/dockerContent-#
```

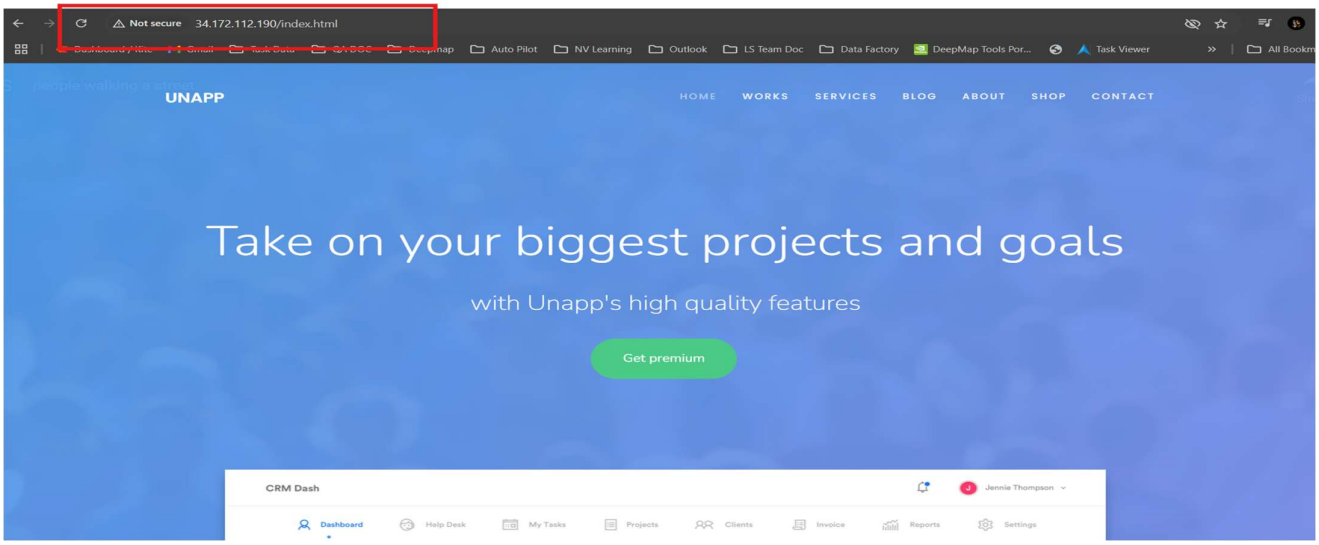
On Worker node:

```
root@node1:~# docker ps
CONTAINER ID   IMAGE                        COMMAND              CREATED          STATUS
      PORTS      NAMES
52d25fd7c675   sham9394/drupal-webapp:v1    "httpd-foreground"   3 minutes ago    Up 3 minu
tes    80/tcp     drupal-stack_website.3.9j4xg116gqxc7e7q6zlkb2xqt
2261734f2858   sham9394/drupal-webapp:v1    "httpd-foreground"   3 minutes ago    Up 3 minu
tes    80/tcp     drupal-stack_website.1.nmc8yqcn91q7wt4ex5pukbefz
```

```
root@docker-serv:/opt/dockerContent-# docker service ps drupal-stack_website
ID             NAME                    IMAGE                        NODE         DESIRED STATE   CURRENT STATE            ERROR     PORTS
nmc8yqcn91q7   drupal-stack_website.1   sham9394/drupal-webapp:v1    node1        Running         Running 6 minutes ago
9xz67es5uvi5   drupal-stack_website.2   sham9394/drupal-webapp:v1    docker-serv  Running         Running 6 minutes ago
9j4xg116gqxc   drupal-stack_website.3   sham9394/drupal-webapp:v1    node1        Running         Running 6 minutes ago
root@docker-serv:/opt/dockerContent-#
```

8. **Access the Website: Once deployed, we can access the website by navigating to the IP address of any node in the Swarm cluster and port 80**

On Master IP:



On Node IP:

Status	Name	Zone	Recommendations	In use by	Internal IP	External IP	Connect
✓	docker-serv	us-central1-a			10.128.0.15 (nic0)	34.172.112.190 (nic0)	SSH
✓	node1	us-central1-a			10.128.0.16 (nic0)	34.29.36.129 (nic0)	SSH