

# IMPLEMENTATION BY USING DOCKER SWARM

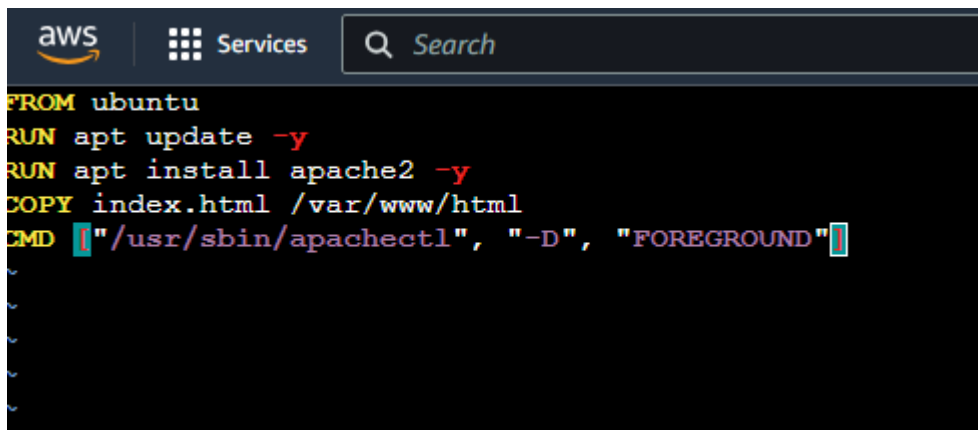
Step 1: Launch one instance and connect with the created instance.

- Switch to the root user by using the command **“sudo su -”**.

Step 2: Install the docker in a terminal and start the docker and check the status by using the following commands.

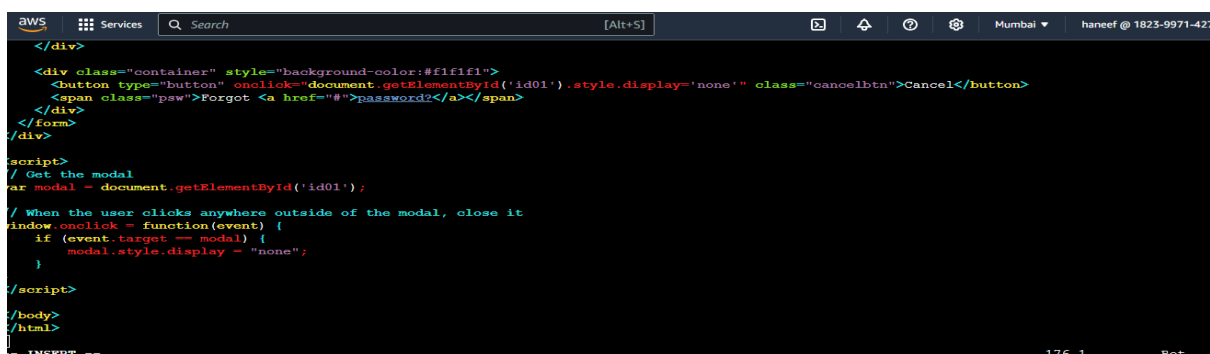
- **yum install docker** - Install docker
- **systemctl start docker** -To start docker
- **systemctl status docker** -To check the status of docker

Step 3: Create the docker file by using the command **“vi Dockerfile”** in that you may perform the following instruction as shown in the below figure.



```
FROM ubuntu
RUN apt update -y
RUN apt install apache2 -y
COPY index.html /var/www/html
CMD ["/usr/sbin/apachectl", "-D", "FOREGROUND"]
```

Step 4: Perform the command called **“vi index.html”**. Here insert the html code in that as shown in the below figure.



```
</div>
<div class="container" style="background-color:#f1f1f1">
  <button type="button" onclick="document.getElementById('id01').style.display='none'" class="cancelbtn">Cancel</button>
  <span class="psw">Forgot <a href="#">password?</a></span>
</div>
</form>
</div>
</div>
<script>
// Get the modal
var modal = document.getElementById('id01');
// When the user clicks anywhere outside of the modal, close it
window.onclick = function(event) {
  if (event.target == modal) {
    modal.style.display = "none";
  }
}
</script>
</body>
</html>
```

Step 5: Build the image by using the command.

- “**docker build -t image1 .**” as shown in the below figure.

```
[root@ip-172-31-4-167 ~]# docker build -t image1 .
[+] Building 26.3s (9/9) FINISHED                                docker:default
=> [internal] load build definition from Dockerfile              0.0s
=> => transferring dockerfile: 234B                               0.0s
=> [internal] load metadata for docker.io/library/ubuntu:latest 2.2s
=> [internal] load .dockerignore                                  0.0s
=> => transferring context: 2B                                       0.0s
=> [1/4] FROM docker.io/library/ubuntu:latest@sha256:278628f08d4979fb9af9ead44277dbc9c92c2465922310916ad0c46ec9999295 2.5s
=> => resolve docker.io/library/ubuntu:latest@sha256:278628f08d4979fb9af9ead44277dbc9c92c2465922310916ad0c46ec9999295 0.0s
=> => sha256:278628f08d4979fb9af9ead44277dbc9c92c2465922310916ad0c46ec9999295 6.69kB / 6.69kB 0.0s
=> => sha256:f470988096c4d77efac9740a1b6700823681af518a17fad30111430b95dfbffa 424B / 424B 0.0s
=> => sha256:fec8b9d95b54439b934c5033dc62d79b946291c327814f2d4df181e1d7536806 2.30kB / 2.30kB 0.0s
=> => sha256:afad30e59d72d5c8df4023014c983e457f21818971775c4224163595ec20b69f 29.75MB / 29.75MB 0.6s
=> => extracting sha256:afad30e59d72d5c8df4023014c983e457f21818971775c4224163595ec20b69f 1.7s
=> [internal] load build context                                  0.0s
=> => transferring context: 3.98kB                                0.0s
=> [2/4] RUN apt update -y                                       6.2s
=> [3/4] RUN apt install apache2 -y                             13.6s
```

Step 6: Switch to the docker hub and create on repository.

- Switch to the terminal perform the command called “**docker tag image1 haneef945/repo1**”.

Step 7: Perform the command “**docker login**”.

- Here have to provide the username and password of a dockerhub account, Example.
- Username : \*\*\*\*\*
- Password :\*\*\*\*\*

Step 8 : Push the image into your docker hub repository by using the following command.

- “**docker tag image1 haneef945/repo1**”.

Step 9 : Initialize the swarm docker by using the following command.

- “**docker swarm init**” - Used to create the Manager node and it generates the token as shown in the below figure.

```
[root@ip-172-31-4-167 ~]# docker swarm init
Swarm initialized: current node (kw364ph1k1ih0y5q5va3e9e66) is now a manager.

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

    docker swarm join --token SWMTKN-1-1n3h4tjnxdsfw3qzeo9n1a9ze6bszi24poe737z17rj6yqsxlw-0ykcwbhai1hgqoa50x114ve4q 172.31.4.167:2377

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

Step 10: Create the service by using the following command.

- **“docker service create --name cont1 --publish 8000:80 image1”.**
- Here the service is created as well as the container also created as shown in the given figure.

```
root@ip-172-31-4-167 ~]# docker service create --name cont1 --publish 8000:80 image1
image image1:latest could not be accessed on a registry to record
its digest. Each node will access image1:latest independently,
possibly leading to different nodes running different
versions of the image.

7ymhtx4mxixf1mzz84wsp8d
overall progress: 1 out of 1 tasks
/1: running [=====>]
verify: Service converged
```

- Copy the public IP and enter in a google with the port number you can access output as shown in the figure.

The screenshot displays a web application interface with a 'Modal Login Form'. The form is a white box with a grey border, set against a dark grey background. At the top left of the modal is a small 'Avatar' icon and a close button 'X'. The form contains two input fields: 'Username' with a placeholder 'Enter Username' and 'Password' with a placeholder 'Enter Password'. Below these fields is a large green 'Login' button. Under the button is a checkbox labeled 'Remember me'. At the bottom left is a red 'Cancel' button, and at the bottom right is a blue link that says 'Forgot password?'. In the top left corner of the dark grey background, there is a green 'Login' button.

# Register

Please fill in this form to create an account.

Email

Enter Email

Password

Enter Password

Repeat Password

Repeat Password

By creating an account you agree to our [Terms & Privacy](#).

Register