Lab: Docker Compose

Introduction:

Docker-Compose is a tool for defining and running multi-container Docker applications. With Compose, you use a YAML file to configure your application's services. Then with a single command you create and start all the services from your configuration.

Compose works in all environments: production, staging, development, testing, as well as CI workflows.

Objective:

- **Install Docker Compose**
- **Create a Ghost Blog and MySQL Service.**
- **Bring Up the Ghost Blog Service.**
- **Cleanup**

1. Run this command to **install** the latest version of **Docker Compose**.

```
# dnf install -y docker-compose-plugin
```

Output:

```
[root@docker-engine ~]#dnf install -y docker-compose-plugin
Last metadata expiration check: 0:01:58 ago on Tue 07 Feb 2023 09:28:43 AM IST.
Dependencies resolved.
================================================================================
 Package                  Architecture      Version                Repository
================================================================================
Installing:
 docker-compose-plugin    x86_64            2.15.1-3.el9           docker-ce-stable

Transaction Summary
================================================================================
Install  1 Package

Total download size: 10 M
Installed size: 43 M
Downloading Packages:
docker-compose-plugin-2.15.1-3.el9.x86_64.rpm                   31 MB/s |  10 MB
--------------------------------------------------------------------------------
Total                                                           30 MB/s |  10 MB
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :
  Installing       : docker-compose-plugin-2.15.1-3.el9.x86_64
  Running scriptlet: docker-compose-plugin-2.15.1-3.el9.x86_64
  Verifying        : docker-compose-plugin-2.15.1-3.el9.x86_64

Installed:
  docker-compose-plugin-2.15.1-3.el9.x86_64
```

1.1 Let's **verify** the **version** has been installed.

```
# docker compose version
```

Student Material – Do Not Re-distribute. For any queries contact:
training@eyesoncloud.com or https://www.linkedin.com/in/eyesoncloud/

Output:

```
[root@docker-engine ~]#docker compose version
Docker Compose version v2.16.0
```

2. Let's Create a Ghost Blog and MySQL Service.

2.1 Let's **create** a directory named **~/ghost-app.**

```
# mkdir ~/ghost-app
```

2.2 Let's get **inside** the directory ~/ghost-app.

```
# cd ~/ghost-app
```

Output:

```
[root@docker-engine ~]#cd ~/ghost-app/
[root@docker-engine ghost-app]#
```

2.3 Let's download "**docker-compose.yml**" for deploying two-tier app.

```
# wget
https://raw.githubusercontent.com/EyesOnCloud/docker/main/doc
ker-compose.yml
```

Output:

```
root@docker-engine ghost-app #wget https://raw.githubusercontent.com/EyesOnCloud/docker/main/docker-compose.yml
--2023-02-07 09:42:58--  https://raw.githubusercontent.com/EyesOnCloud/docker/main/docker-compose.yml
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.111.133, 185.199.110.133, 185.199.109.133, ..
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 662 [text/plain]
Saving to: 'docker-compose.yml'

docker-compose.yml              100%[=========================================================>]     662  --.-KB/s

2023-02-07 09:42:58 (1.98 MB/s) - 'docker-compose.yml' saved [662/662]
```

2.4 Let's **view** the manifest.

```
# cat -n docker-compose.yml
```

Eyes On Cloud

Output:

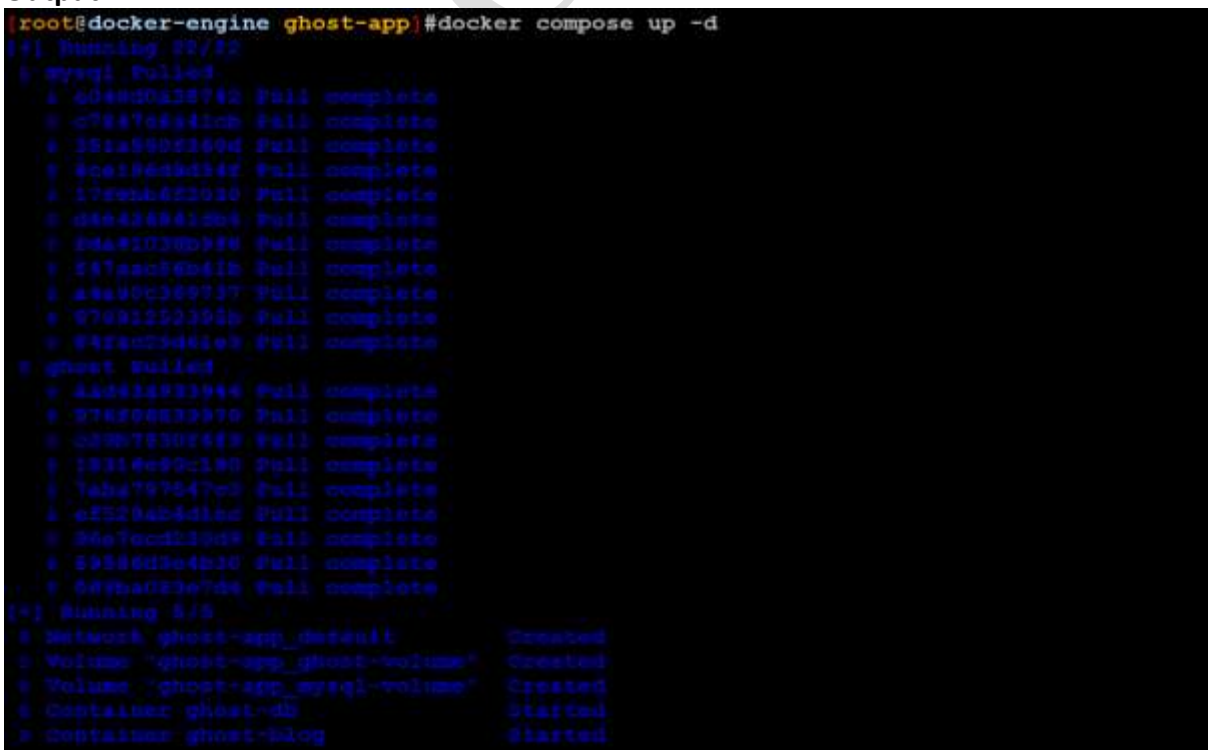
```
[root@docker-engine ghost-app]#cat -n docker-compose.yml
     1  version: '3.3'
     2  services:
     3    ghost:
     4      image: ghost:1-alpine
     5      container_name: ghost-blog
     6      restart: always
     7      ports:
     8        - 80:2368
     9      environment:
    10        database__client: mysql
    11        database__connection__host: mysql
    12        database__connection__user: root
    13        database__connection__password: P4sSw0rd0!
    14        database__connection__database: ghost
    15      volumes:
    16        - ghost-volume:/var/lib/ghost
    17      depends_on:
    18        - mysql
    19
    20    mysql:
    21      image: mysql:5.7
    22      container_name: ghost-db
    23      restart: always
    24      environment:
    25        MYSQL_ROOT_PASSWORD: P4sSw0rd0!
    26      volumes:
    27        - mysql-volume:/var/lib/mysql
    28
    29  volumes:
    30    ghost-volume:
    31    mysql-volume:
```

3 Let's Bring Up the Ghost Blog Service.

3.1 Let's **Start up** the Docker Compose service.

```
# docker compose up -d
```

Output:

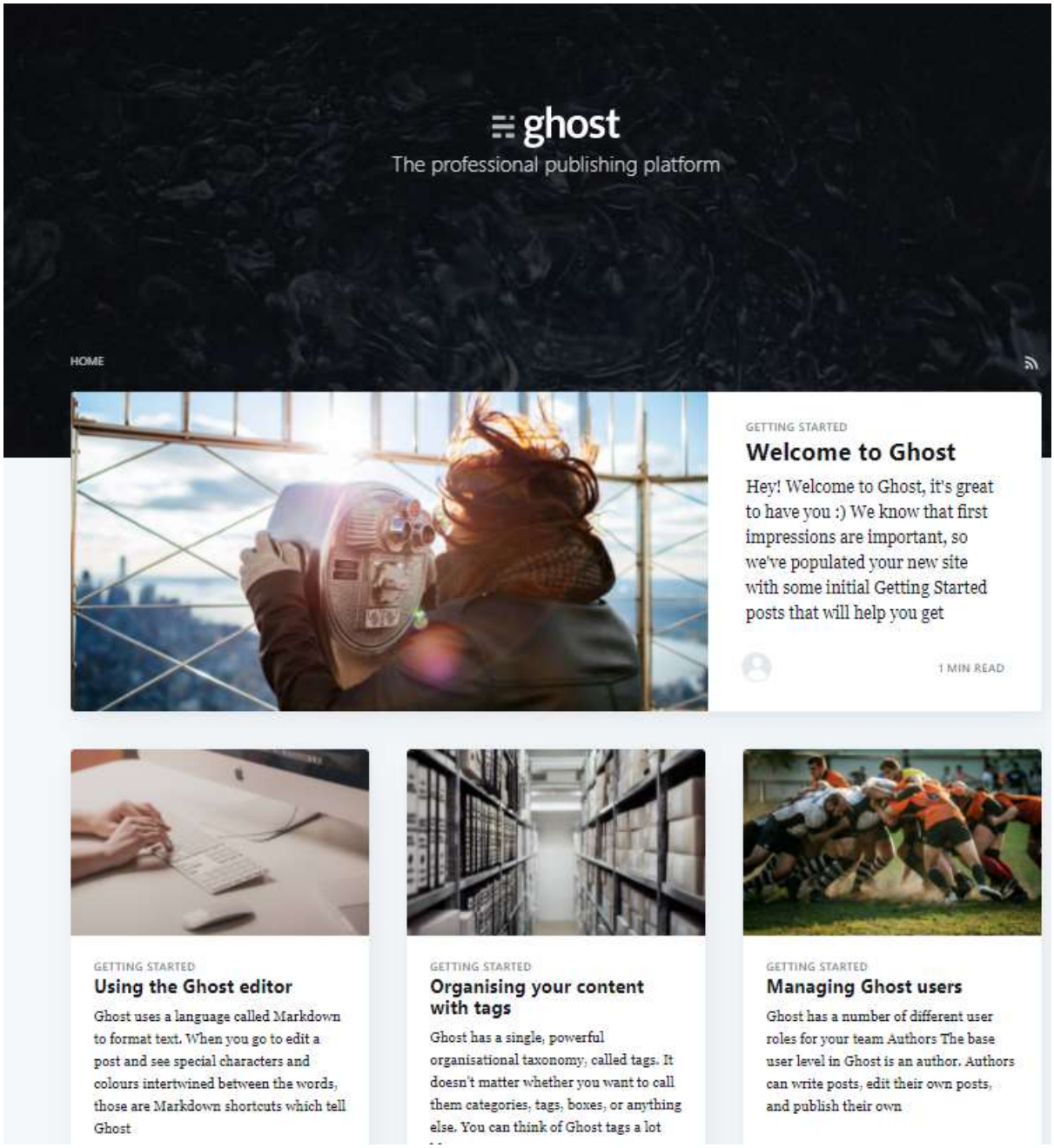


Student Material – Do Not Re-distribute. For any queries contact:
training@eyesoncloud.com or https://www.linkedin.com/in/eyesoncloud/

Conclusion

Confirm by access the below URL from the web browser.

http://PublicIP:80



Student Material – Do Not Re-distribute. For any queries contact:
training@eyesoncloud.com or https://www.linkedin.com/in/eyesoncloud/

4 Cleanup.

4.1 Let's **stop** the Application.

```
# docker compose stop
```

Output:

```
[root@docker-engine ghost-app]#docker compose stop
[+] Running 2/2
 ⠿ Container ghost-blog  Stopped
 ⠿ Container ghost-db    Stopped
```

4.2 Let's **remove** the Application.

```
# docker compose rm -f
```

Output:

```
[root@docker-engine ghost-app]#docker compose rm -f
Going to remove ghost-blog, ghost-db
[+] Running 2/0
 ⠿ Container ghost-db    Removed
 ⠿ Container ghost-blog  Removed
```

4.3 Let's **cleanup** the dependent objects.

```
# docker compose down
```

Output:

```
[root@docker-engine ghost-app]#docker compose down
[+] Running 1/1
 ⠿ Network ghost-app_default  Removed
```

4.4 Let's **remove** all the **images** run the below commands.

```
# docker image rm `docker image ls -q` -f
```

Eyes On Cloud

Student Material – Do Not Re-distribute. For any queries contact:
training@eyesoncloud.com or https://www.linkedin.com/in/eyesoncloud/

Output:

```
[root@docker-engine ghost-app]#docker image rm `docker image ls -q` -f
Untagged: mysql:5.7
Untagged: mysql@sha256:8cf035b14977b26f4a47d98e85949a7dd35e641f88fc24aa4b466b36beecf9d6
Deleted: sha256:be16cf2d832a9a54ce42144e25f5ae7cc66bccf0e003837e7b5eb1a455dc742b
Deleted: sha256:2dd6e094d35f48086adcdff89f36c9be8166fbbda0775b72efd9b12788a400a0
Deleted: sha256:01de218cde8a6eef7c19083f1a9dfdcc10e656651137c5083c8e8b293561e674
Deleted: sha256:84e3163c39fc29722fbea1461bc279bafc9d2da5937b1f4e5db6e08c6f08e3b4
Deleted: sha256:2b39b403c72a380afa25629a8b0af21e5bd33b071985740242e63856e6453359
Deleted: sha256:bf76a072ee83eef013d21ef4aed92cea89f1d15edd62ce0d4204268f513a2658
Deleted: sha256:8112761750f4100814858c647189c97db4129852fec2060580625254fa9b3440
Deleted: sha256:a527ebda1c7ee77668ecfec84a62ba7cc46769946fb565ff7895f2a0aecfd082
Deleted: sha256:ba8f679e76a9faba19756857e6f5d6809d8ba0a90507c6a815f3823244636e64
Deleted: sha256:51b0a4dc7d8cdf87950abd409fe3155df9118e4b33a1bbe91d6367d62557db87
Deleted: sha256:f3277814332978b36b309d241f692a182e37564d96a94fdde0382cb471953483
Deleted: sha256:c233345f327a72c79e2e19c75fcf9089a0c1488dfb66dd00d49fa2a5d1f76057
Untagged: ghost:1-alpine
Untagged: ghost@sha256:0a9957f8831db9fe6a87fe95217053939601fdcd1db047cb8b106f0ec4b7506b
Deleted: sha256:efcd4044e3a0e05f85560a6c8111096511fbc17812474f8c871b77844a68d92d
Deleted: sha256:c00d5d9ae18eb43e37cce7140d0aa2a1b8de2b75bf5a603814515bf34cf45520
Deleted: sha256:186d738eda8736f8a5e4aed6c0aa50959b1e1a643d0f3365745b2c8f38532bf8
Deleted: sha256:4160f9fd1c7b85a0fe338be0544ccf834eec883982b4976f531e91c8041910d2
Deleted: sha256:2b216a78326e73e7f5c704c9905f576c2499f117670caf5b416eb8a88b616a45
Deleted: sha256:b2d984eb28da3d83ec4259c552bb70b7eec2017c1b93e03a5276389cae2c0e76
Deleted: sha256:697ff4d37a7b82b48adbe739972ae7cbdaefe721c82829933a86d5e6b787f179
Deleted: sha256:15ea3998b3cf3293aa2fc0c0cbb8605921742ca17d6457e1fe065ff6ed0d7aeb
Deleted: sha256:5382df0b73a6712d4018d9778149ea8915bf8ee43733bce9988beae42d87e082
Deleted: sha256:beee9f30bc1f711043e78d4a2be0668955d4b761d587d6f60c2c8dc081efb203
```

4.5 Let's **remove** the **Volumes** for complete cleanup.

```
# docker volume prune -f
```

4.6 Let's remove the Network for complete cleanup.

```
# docker network prune -f
```

4.7 Let's **verify** that **containers** are removed.

```
# docker container ls
```

Output:

```
[root@docker-engine ghost-app]#docker container ls
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

```
# cd ~
```

4.8 Let's **verify** that docker **images** are removed.

```
# docker image ls
```

Output:

```
[root@docker-engine ~]#docker image ls
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
```

Note: For more information refer below link.

https://docs.docker.com/compose/gettingstarted/

Student Material – Do Not Re-distribute. For any queries contact:
training@eyesoncloud.com or https://www.linkedin.com/in/eyesoncloud/