DOCKER-02

1) Create a tomcat container on 8081 and deploy sample application in tomcat.

docker container run -itd -p 8081:8080 tomcat:latest

docker ps

```
[root@ip-10-0-0-169 ~]# docker ps
CONTAINER ID   IMAGE           COMMAND             CREATED             STATUS              PORTS
                                 NAMES
05436ced9517   tomcat:latest   "catalina.sh run"   About a minute ago   Up About a minute   0.0.0.0:808
1->8080/tcp, :::8081->8080/tcp   competent_turing
```

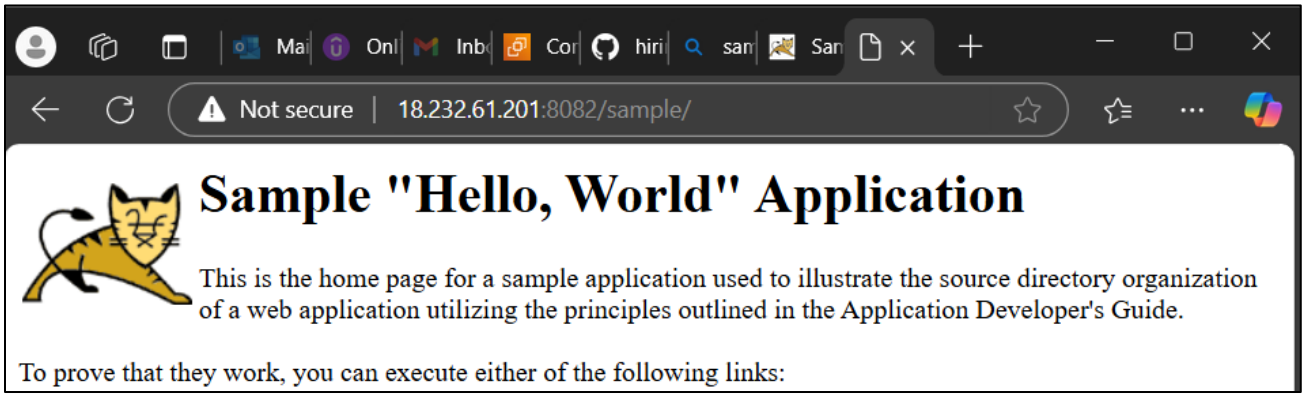
docker exec -it 0543 /bin/bash

2) Create volume and deploy tomcat container on port 8082.

docker container run -itd -p 8082:8080 -v /var/lib/docker/volumes/tomcat_app/_data: /usr/local/tomcat/webapps tomcat:latest

```
[root@ip-10-0-0-169 ~]# docker container run -itd -p 8082:8080 -v /var/lib/docker/volumes/tomcat_app/_d
ata:/usr/local/tomcat/webapps tomcat:latest
5bfe8e10c7ea886ba25c6979c825a9b6027c28455f7e336cdd3f5149ab88ae3d
[root@ip-10-0-0-169 ~]# docker ps
CONTAINER ID   IMAGE           COMMAND             CREATED          STATUS          PORTS
                                 NAMES
5bfe8e10c7ea   tomcat:latest   "catalina.sh run"   30 seconds ago   Up 29 seconds   0.0.0.0:8082->8080/
tcp, :::8082->8080/tcp   hopeful_bartik
05436ced9517   tomcat:latest   "catalina.sh run"   22 minutes ago   Up 22 minutes   0.0.0.0:8081->8080/
tcp, :::8081->8080/tcp   competent_turing
```

→checking on browser



3) Limit the nginx container to 500 MB.

```
[root@ip-10-0-0-169 ~]# docker container run -itd -m 500m nginx
e320d5ac5a43ea3752c8231543e810523a7a9699f32fa52b2f87dcbddce5d4b0
[root@ip-10-0-0-169 ~]# docker stats
CONTAINER ID   NAME                CPU %     MEM USAGE / LIMIT     MEM %     NET I/O           BLOCK I/O         PIDS
e320d5ac5a43   intelligent_shamir  0.00%     6.109MiB / 500MiB     1.22%     500B / 0B         11.6MB / 15.9kB   3
5bfe8e10c7ea   hopeful_bartik      0.18%     92.11MiB / 7.748GiB   1.16%     4.66kB / 7.05kB   0B / 0B           35
05436ced9517   competent_turing    0.32%     99.98MiB / 7.748GiB   1.26%     71.9kB / 66kB     12.3MB / 18.9kB   35
```

4) Create a sample docker file using below instructions.

- **Base module as amazonlinux:latest**
- **Maintainer you name**
- **Install nginx**
- **COPY one index.html file to image**
- **Expose on port 80**
- **Command to start the nginx container**

vi index.html (**add the sample content)**

vi Dockerfile

```
# Use the latest Amazon Linux as the base image
FROM amazonlinux:latest

# Maintainer information
MAINTAINER Farsaan

# Install nginx
RUN yum -y update && \
   yum -y install nginx && \
   yum clean all

# Copy the index.html file to the nginx html directory
COPY index.html /usr/share/nginx/html/

# Expose port 80
EXPOSE 80

# Command to start the nginx container
CMD ["nginx", "-g", "daemon off;"]
```

docker build -t nginx:v1 .

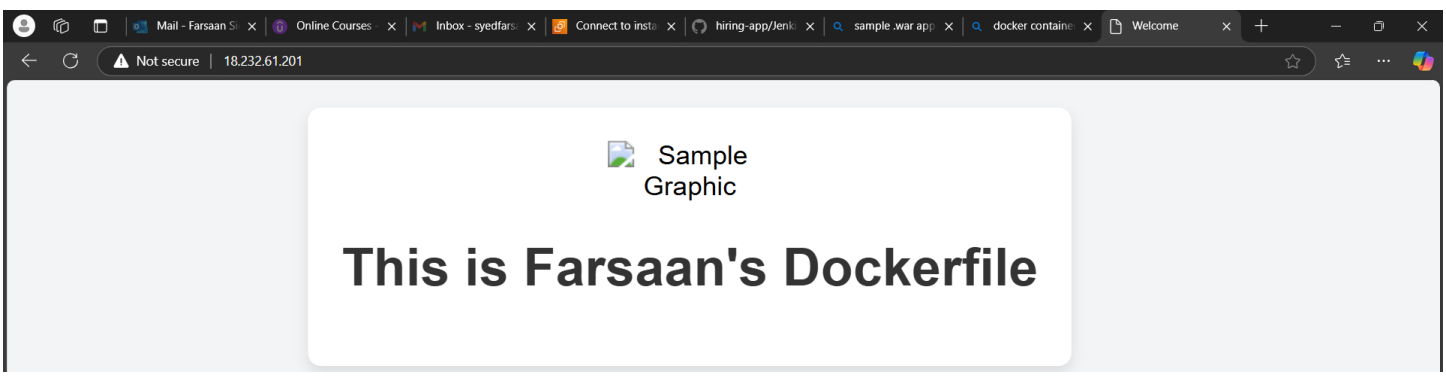
```
[root@ip-10-0-0-169 ~]# vi Dockerfile
[root@ip-10-0-0-169 ~]# vi index.html
[root@ip-10-0-0-169 ~]# docker build -t nginx:v1
ERROR: "docker buildx build" requires exactly 1 argument.
See 'docker buildx build --help'.

Usage:  docker buildx build [OPTIONS] PATH | URL | -

Start a build
[root@ip-10-0-0-169 ~]# docker build -t nginx:v1 .
[+] Building 17.5s (8/8) FINISHED                                                                                   docker:default
 => [internal] load build definition from Dockerfile                                                                          0.0s
 => => transferring dockerfile: 444B                                                                                          0.0s
 => [internal] load metadata for docker.io/library/amazonlinux:latest                                                         0.3s
 => [internal] load .dockerignore                                                                                             0.0s
 => => transferring context: 2B                                                                                               0.0s
 => [1/3] FROM docker.io/library/amazonlinux:latest@sha256:5cb6ab1a1a13f33425b6c660a45f06298e204a22df1f9eeefe69cda3735d9757   3.8s
 => => resolve docker.io/library/amazonlinux:latest@sha256:5cb6ab1a1a13f33425b6c660a45f06298e204a22df1f9eeefe69cda3735d9757   0.0s
 => => sha256:5cb6ab1a1a13f33425b6c660a45f06298e204a22df1f9eeefe69cda3735d9757 2.38kB / 2.38kB                                0.0s
 => => sha256:2d884271a93133a33d91fcf93c042156800dd572a59090f8d035f87da55dbeb7 1.02kB / 1.02kB                                0.0s
 => => sha256:4e24a4e616bb9928209da85eb4d989dc91cbf785c4687947d51fd75fde578dd9 556B / 556B                                    0.0s
 => => sha256:4a665eb63bc8cddb90e1e74e3ec745a1bab733c919dc4b2d648b43459295464a 52.38MB / 52.38MB                              0.6s
 => => extracting sha256:4a665eb63bc8cddb90e1e74e3ec745a1bab733c919dc4b2d648b43459295464a                                     3.0s
 => [internal] load build context                                                                                             0.0s
 => => transferring context: 1.00kB                                                                                           0.0s
 => [2/3] RUN yum -y update &&     yum -y install nginx &&     yum clean all                                                 12.7s
 => [3/3] COPY index.html /usr/share/nginx/html/                                                                              0.0s
 => exporting to image                                                                                                        0.5s
 => => exporting layers                                                                                                       0.5s
 => => writing image sha256:1ccd5204a2882576b748542f5338be03ad673211cc6c75f6498770939a89f48b                                  0.0s
 => => naming to docker.io/library/nginx:v1                                                                                   0.0s
```

docker ps

docker container run -itd -p 80:80 nginx:v1

```
[root@ip-10-0-0-169 ~]# docker ps
CONTAINER ID   IMAGE           COMMAND                  CREATED          STATUS          PORTS                                       NAMES
e320d5ac5a43   nginx           "/docker-entrypoint.…"   21 minutes ago   Up 21 minutes   80/tcp                                      intelligent_shamir
5bfe8e10c7ea   tomcat:latest   "catalina.sh run"        28 minutes ago   Up 28 minutes   0.0.0.0:8082->8080/tcp, :::8082->8080/tcp   hopeful_bartik
05436ced9517   tomcat:latest   "catalina.sh run"        50 minutes ago   Up 50 minutes   0.0.0.0:8081->8080/tcp, :::8081->8080/tcp   competent_turing
[root@ip-10-0-0-169 ~]# docker run -itdd -p 80:80 nginx:v1
fa08e19167ed57a22685b1a73db45eea40251a4c30029fe22dac602e4753c9c5
```

→check on browser with <public-ip>:80



5) Push image to dockerhub

docker login

```
[root@ip-10-0-0-169 ~]# docker login
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, he
ad over to https://hub.docker.com/ to create one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security a
nd is required for organizations using SSO. Learn more at https://docs.docker.com/go/access-tokens/

Username: syedfs97@outlook.com
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

→create a repo in docker hub and change the tag in docker host for the image we created from dockerfile

docker tag nginx:v1 syedfs97/nginx-repo:v1

docker push syedfs97/nginx-repo:v1

```
[root@ip-10-0-0-169 ~]# docker images
REPOSITORY          TAG       IMAGE ID       CREATED          SIZE
nginx               v1        1ccd5204a288   13 minutes ago   208MB
jenkins/jenkins     latest    76242b94846e   2 days ago       468MB
tomcat              latest    f62f518e5c5c   10 days ago      467MB
sonarqube           latest    6bada21f1230   2 weeks ago      1.12GB
nginx               latest    66f8bdd3810c   3 weeks ago      192MB
ubuntu              latest    b1d9df8ab815   4 weeks ago      78.1MB
jenkins             2.60.3    cd14cecfdb3a   6 years ago      696MB
[root@ip-10-0-0-169 ~]# docker tag nginx:v1 syedfs97/nginx-repo:v1
[root@ip-10-0-0-169 ~]# docker images
REPOSITORY              TAG       IMAGE ID       CREATED          SIZE
nginx                   v1        1ccd5204a288   15 minutes ago   208MB
syedfs97/nginx-repo     v1        1ccd5204a288   15 minutes ago   208MB
jenkins/jenkins         latest    76242b94846e   2 days ago       468MB
tomcat                  latest    f62f518e5c5c   10 days ago      467MB
sonarqube               latest    6bada21f1230   2 weeks ago      1.12GB
nginx                   latest    66f8bdd3810c   3 weeks ago      192MB
ubuntu                  latest    b1d9df8ab815   4 weeks ago      78.1MB
jenkins                 2.60.3    cd14cecfdb3a   6 years ago      696MB
[root@ip-10-0-0-169 ~]# docker push syedfs97/nginx-repo:v1
The push refers to repository [docker.io/syedfs97/nginx-repo]
9ad539e34e54: Pushed
2f7f3a4c651f: Pushed
0b7270d4d52c: Mounted from library/amazonlinux
v1: digest: sha256:c8a8158143d03f5b73c55d668e359ff2d53bd07148c045531d5592e120ed4159 size: 948
```

→ check the dockerhub

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
v1		Image	3 minutes ago	3 minutes ago

6) push image to aws ecr

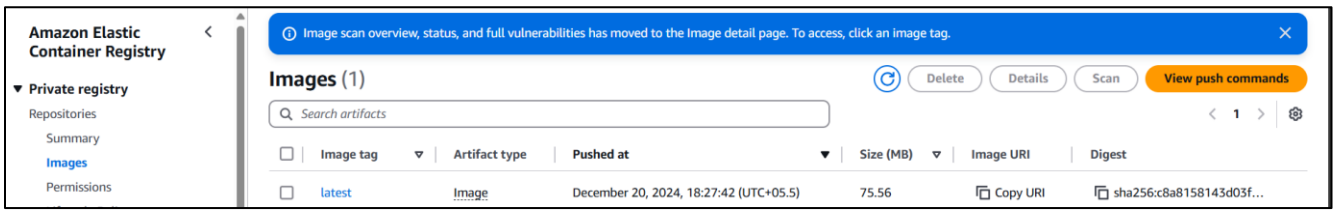
→ create a repo in aws ecr and follow the instructions in view push commands

aws configure

aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 390403872402.dkr.ecr.us-east-1.amazonaws.com

docker tag nginx:v1 390403872402.dkr.ecr.us-east-1.amazonaws.com/nginx-repo:latest

docker push 390403872402.dkr.ecr.us-east-1.amazonaws.com/nginx-repo:latest

```
[root@ip-10-0-0-169 ~]# docker push 390403872402.dkr.ecr.us-east-1.amazonaws.com/nginx-repo:latest
The push refers to repository [390403872402.dkr.ecr.us-east-1.amazonaws.com/nginx-repo]
9ad539e34e54: Pushed
2f7f3a4c651f: Pushed
0b7270d4d52c: Pushed
latest: digest: sha256:c8a8158143d03f5b73c55d668e359ff2d53bd07148c045531d5592e120ed4159 size: 948
```

→checking the ecr