Docker

Assignment 1: Build Java Application with Maven and Deploy Using GitLab CI/CD (Using Shell & Docker Executors)

Dockerfile

FROM openjdk:17-jdk-slim

WORKDIR /app

COPY target/java-docker-app-1.0-SNAPSHOT.jar app.jar

CMD ["java", "-jar", "app.jar"]

Build & Test Locally

Build image: -

docker build -t docker-app:latest .

```
Vishnu Raj@LAPTOP-KSAOBH37 MINGW64 ~/devops/app_build_deploy (main)
$ docker build -t docker-app:latest .
[+] Building 1.4s (8/8) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 164B
 => [internal] load metadata for docker.io/library/openjdk:17-jdk-slim
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [1/3] FROM docker.io/library/openjdk:17-jdk-slim@sha256:aaa3b3cb27e3e520b8f116863d0580c438ed55ecfa0bc126b41f68c3f62f9774
 => => resolve docker.io/library/openjdk:17-jdk-slim@sha256:aaa3b3cb27e3e520b8f116863d0580c438ed55ecfa0bc126b41f68c3f62f9774
 => [internal] load build context
 => => transferring context: 2.51kB
 => CACHED [2/3] WORKDIR /app
 => [3/3] COPY target/java-docker-app-1.0-SNAPSHOT.jar app.jar
 => exporting to image
 => => exporting layers
 => => exporting manifest sha256:71ccaf3fcbd981b25768a1683bc5f9e5dfed8f6ce36422a71ffca11abbaf5973
 => => exporting config sha256:d1e0e2aab136b1cd55e8bee1ee710065599f757ffe9aae7a07ef7db5df215d28
 => => exporting attestation manifest sha256:6e4e8d312ca002087832efbf601eef71fd425343a8088482b758ba168933d71b
 => => exporting manifest list sha256:2dc774b5c3897fb48018b588448e3ebde97fa54133a40838d514a3b2671e8e20
 => => naming to docker.io/library/docker-app:latest
 => => unpacking to docker.io/library/docker-app:latest
```

Run container: -

docker run --rm docker-app:latest

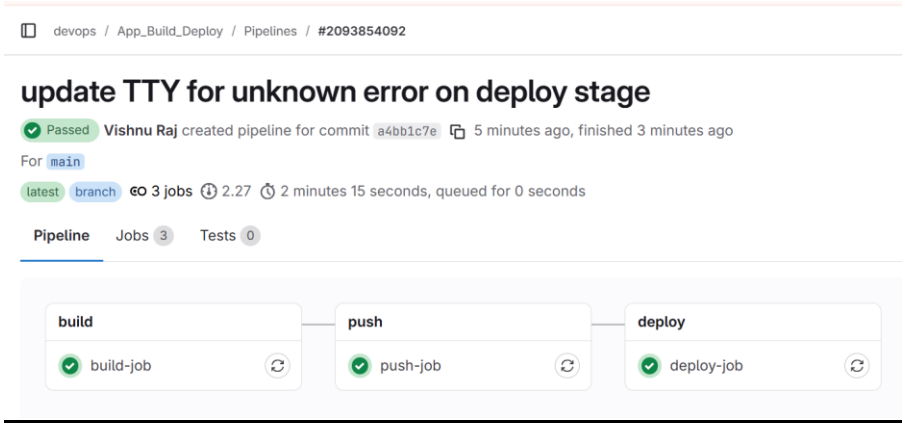
```
Vishnu Raj@LAPTOP-KSAOBH37 MINGW64 ~/devops/app_build_deploy (main)
$ docker run --rm docker-app:latest
Java Application with Maven and Deploy Using GitLab CI/CD

Vishnu Raj@LAPTOP-KSAOBH37 MINGW64 ~/devops/app_build_deploy (main)
```

GitLab CI/CD Configuration

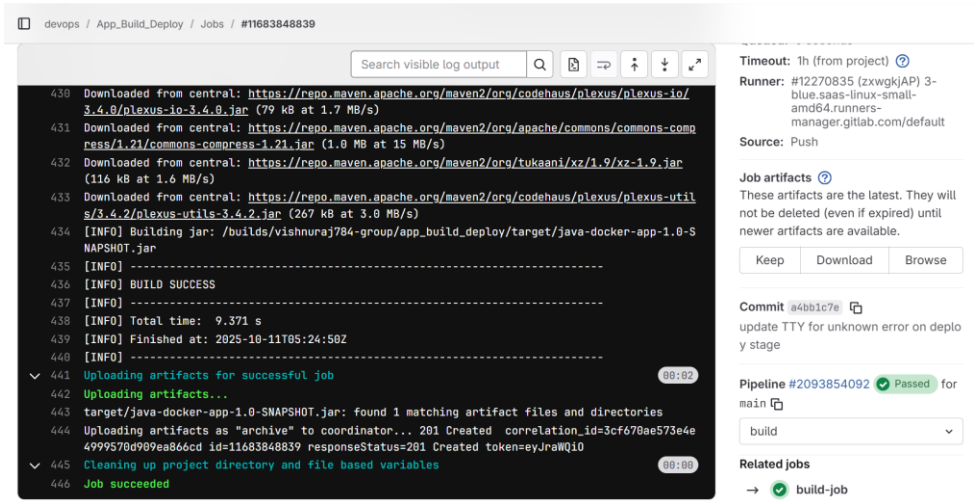
```
#yaml
stages:
  - build
  - push
  - deploy

variables:
  DOCKER_DRIVER: overlay2
  DOCKER_TLS_CERTDIR: ""
  DOCKER_HOST: tcp://docker:2375
  APP_NAME: java-docker-app
  IMAGE_TAG: $CI_REGISTRY_IMAGE:$CI_COMMIT_SHORT_SHA

build-job:
  stage: build
  image: maven:latest
  script:
    - echo "Building the Maven project..."
    - mvn clean package
  artifacts:
    paths:
      - target/java-docker-app-1.0-SNAPSHOT.jar

push-job:
  stage: push
  image: docker:24.0.7
  services:
    - docker:24.0.7-dind
```

```
  script:
    - echo "Login to GITLAB container..."
    - echo "$CI_REGISTRY_PASSWORD" | docker login -u "$CI_REGISTRY_USER"
"$CI_REGISTRY" --password-stdin
    - echo "Building Docker image..."
    - docker build -t $IMAGE_TAG .
    - echo "Pushing the image to GITLAB..."
    - docker push $IMAGE_TAG

deploy-job:
  stage: deploy
  image: docker:24.0.7
  services:
    - docker:24.0.7-dind
  script:
    - echo "Deploying container..."
    - echo "$CI_REGISTRY_PASSWORD" | docker login -u "$CI_REGISTRY_USER"
"$CI_REGISTRY" --password-stdin
    - docker pull $IMAGE_TAG
    - docker run --rm $IMAGE_TAG
    - docker ps
```

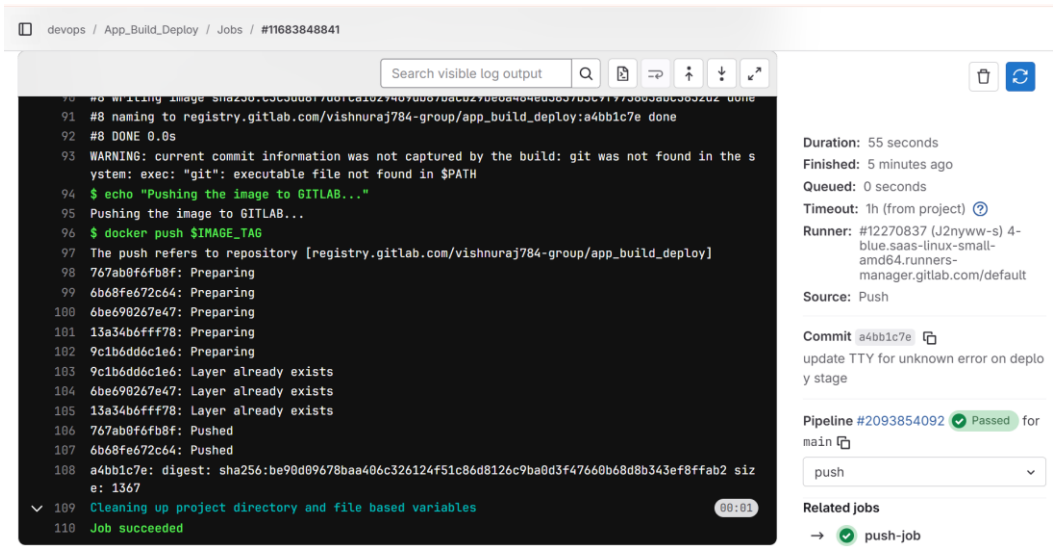
OUTPUT



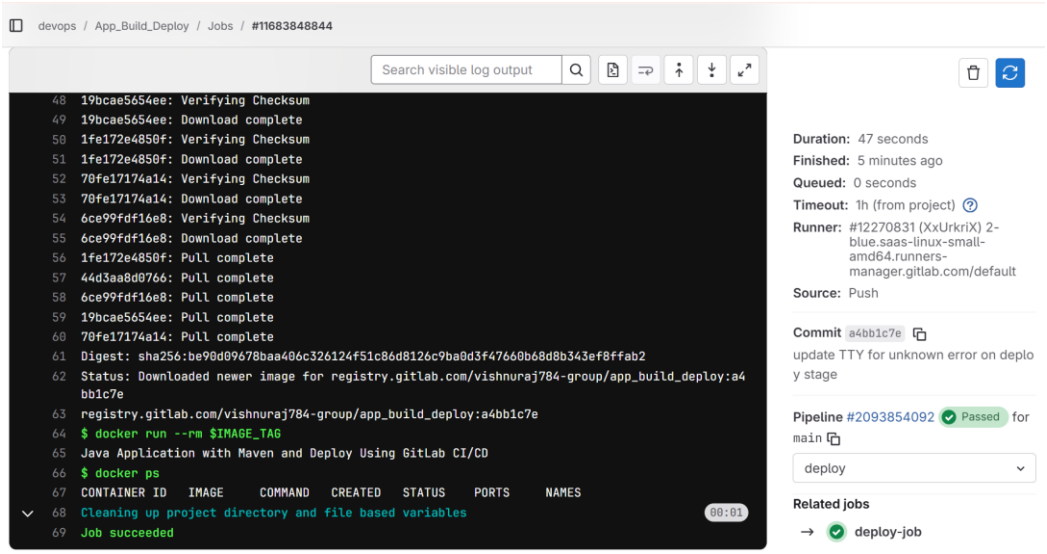
build-job



push-job



deploy-job



Assignment 2: Manage Docker Build Locally and Using GitLab CI/CD

Dockerfile

FROM openjdk:26-slim

WORKDIR /app

COPY HelloWorld.java .

RUN javac HelloWorld.java

CMD ["java", "HelloWorld"]

Build & Test Locally

Build image: -

docker build -t sample-app .

```
Vishnu Raj@LAPTOP-KSAOBH37 MINGW64 ~/devops/manage_docker_build (main)
$ docker build -t sample-app .
[+] Building 3.6s (10/10) FINISHED                          docker:desktop-linux
 => [internal] load build definition from Dockerfile                        0.0s
 => => transferring dockerfile: 151B                                        0.0s
 => [internal] load metadata for docker.io/library/openjdk:26-slim          3.3s
 => [auth] library/openjdk:pull token for registry-1.docker.io              0.0s
 => [internal] load .dockerignore                                           0.0s
 => => transferring context: 2B                                             0.0s
 => [internal] load build context                                           0.0s
 => => transferring context: 37B                                            0.0s
 => [1/4] FROM docker.io/library/openjdk:26-slim@sha256:5be22161570041c14   0.0s
 => => resolve docker.io/library/openjdk:26-slim@sha256:5be22161570041c14   0.0s
 => CACHED [2/4] WORKDIR /app                                               0.0s
 => CACHED [3/4] COPY HelloWorld.java .                                     0.0s
 => CACHED [4/4] RUN javac HelloWorld.java                                  0.0s
 => exporting to image                                                      0.1s
 => => exporting layers                                                     0.0s
 => => exporting manifest sha256:42e6eab5a6f28abeb00fee10515a3e6754634c51   0.0s
 => => exporting config sha256:a27c88ff8bca400b55ec17b24ac992cbe1e370dcb3   0.0s
 => => exporting attestation manifest sha256:c5ed22e1c4bc3ec9f695717ec8d3   0.0s
 => => exporting manifest list sha256:09e7715661d328fa2944c4382a75f0651d2   0.0s
 => => naming to docker.io/library/sample-app:latest                        0.0s
 => => unpacking to docker.io/library/sample-app:latest                     0.0s

Vishnu Raj@LAPTOP-KSAOBH37 MINGW64 ~/devops/manage_docker_build (main)
```

Run container: -

docker run -it sample-app

```
Vishnu Raj@LAPTOP-KSAOBH37 MINGW64 ~/devops/manage_docker_build (main)
$ docker run -it sample-app
Hello from Java in Docker!

Vishnu Raj@LAPTOP-KSAOBH37 MINGW64 ~/devops/manage_docker_build (main)
```

GitLab CI/CD Configuration

yaml file

stages:

- build

- push

variable

variables:

DOCKER_DRIVER: overlay2

DOCKER_TLS_CERTDIR: ""

DOCKER_HOST: tcp://docker:2375

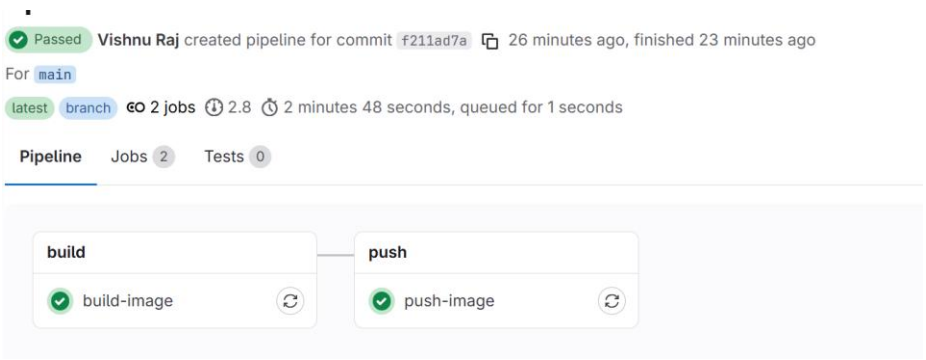
services:

- name: docker:20.10.24-dind

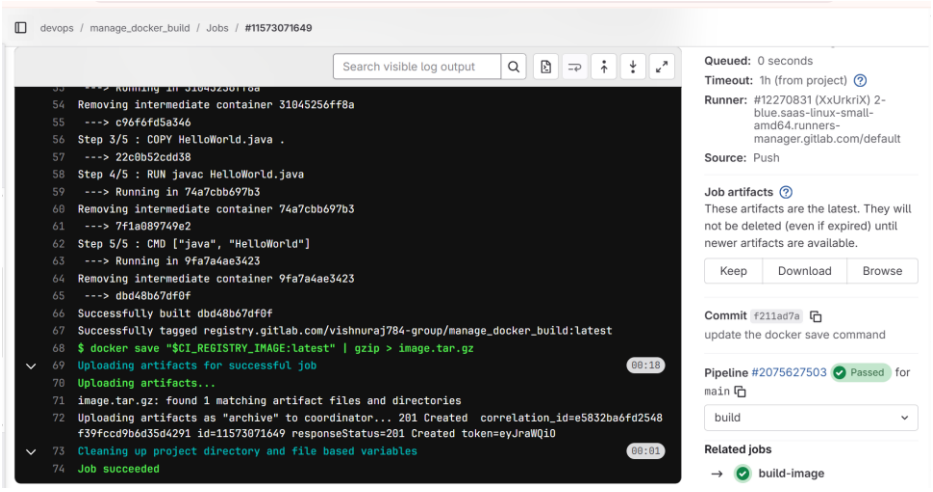
  alias: docker

```
#docker image
image: docker:20.10.24-git
# Login to GitLab Container Registry
before_script:
  - echo $CI_REGISTRY_PASSWORD | docker login -u $CI_REGISTRY_USER --password-stdin
$CI_REGISTRY

#build image stage
build-image:
  stage: build
  script:
    - docker build -t "$CI_REGISTRY_IMAGE:latest" .
    - docker save "$CI_REGISTRY_IMAGE:latest" | gzip > image.tar.gz
  artifacts:
    paths:
      - image.tar.gz

# push image stage
push-image:
  stage: push
  script:
    - docker load < image.tar.gz
    - docker push "$CI_REGISTRY_IMAGE:latest"
```

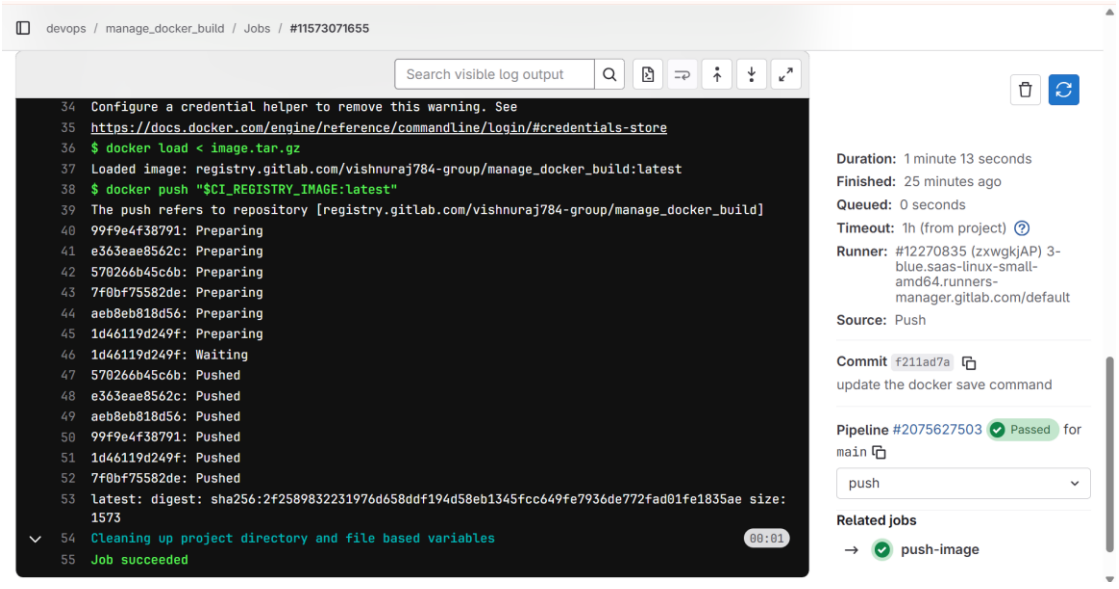
OUTPUT



build-image job



push-image job



Assignment 3: Manage Docker Build Locally and Using GitLab CI/CD with Essential Dockerfile Instructions

Dockerfile

FROM openjdk:26-slim

ARG APP_ENV=production

RUN echo "Building image for environment: $APP_ENV"

WORKDIR /app

COPY target/java-docker-app-1.0-SNAPSHOT.jar app.jar

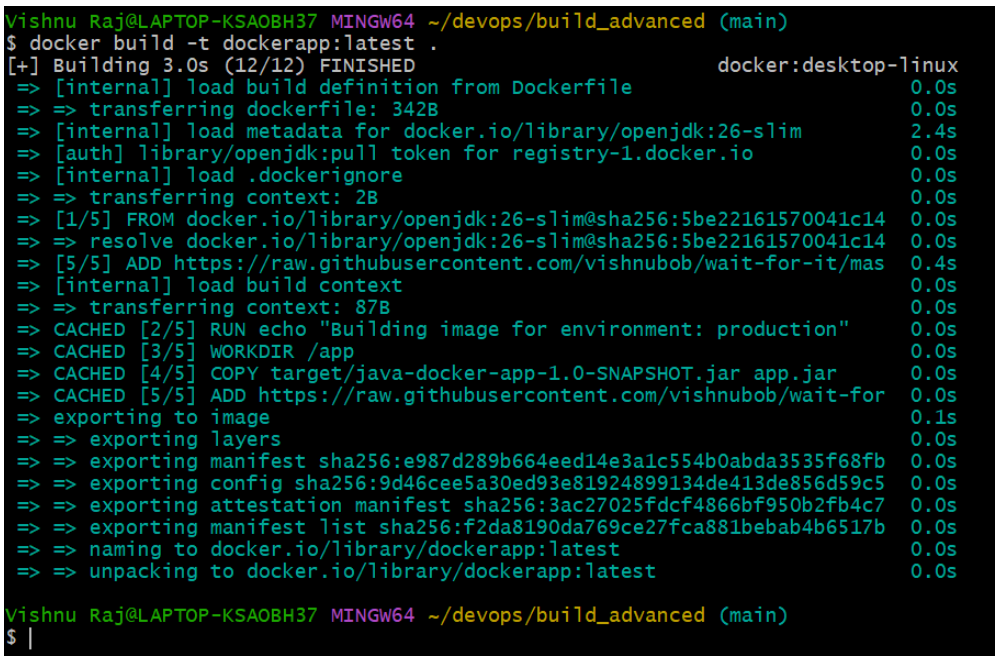
ADD https://raw.githubusercontent.com/vishnubob/wait-for-it/master/wait-for-it.sh /app/wait-for-it.sh

CMD ["java", "-jar", "app.jar"]

Build & Test Locally

Build image: -

```
docker build -t dockerapp:latest .
```



Run container: -

docker run -it dockerapp:latest

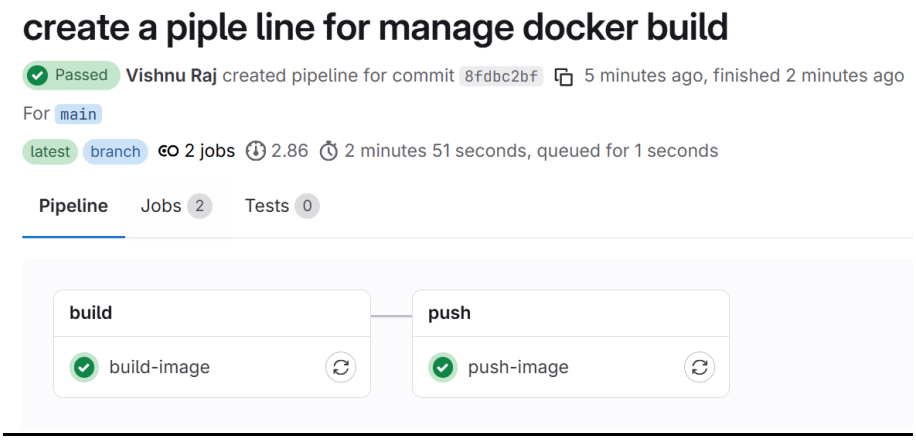
```
Vishnu Raj@LAPTOP-KSAOBH37 MINGW64 ~/devops/build_advanced (main)
$ docker run -it dockerapp:latest
Hello Docker!

Vishnu Raj@LAPTOP-KSAOBH37 MINGW64 ~/devops/build_advanced (main)
```

GitLab CI/CD Configuration

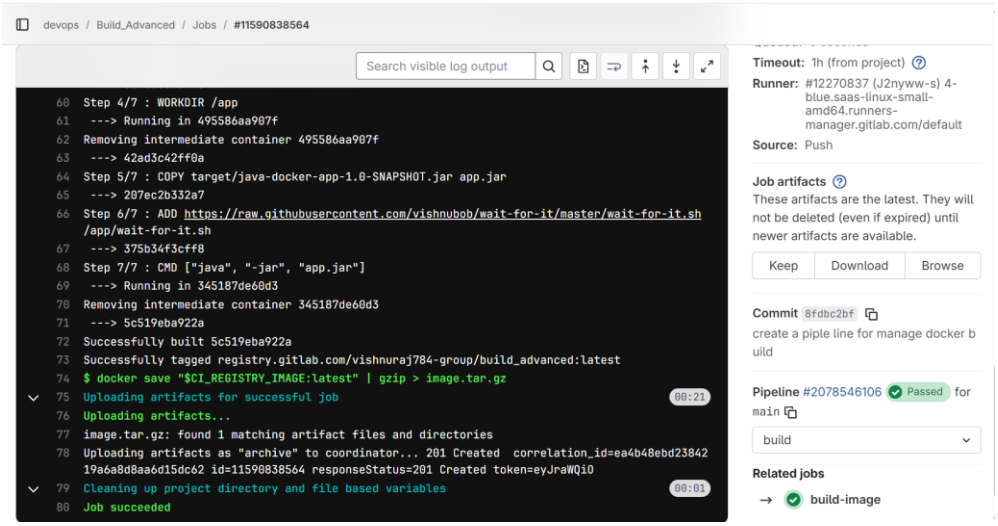
```
# yaml file
stages:
  - build
  - push

# variable
variables:
  DOCKER_DRIVER: overlay2
  DOCKER_TLS_CERTDIR: ""
  DOCKER_HOST: tcp://docker:2375
services:
  - name: docker:20.10.24-dind
    alias: docker

#docker image
image: docker:20.10.24-git

# Login to GitLab Container Registry
before_script:
  - echo $CI_REGISTRY_PASSWORD | docker login -u $CI_REGISTRY_USER --password-stdin
$CI_REGISTRY

#build image stage
```

```
build-image:
  stage: build
  script:
    - docker build -t "$CI_REGISTRY_IMAGE:latest" .
    - docker save "$CI_REGISTRY_IMAGE:latest" | gzip > image.tar.gz
  artifacts:
    paths:
      - image.tar.gz

# push image stage
push-image:
  stage: push
  script:
    - docker load < image.tar.gz
    - docker push "$CI_REGISTRY_IMAGE:latest"
```

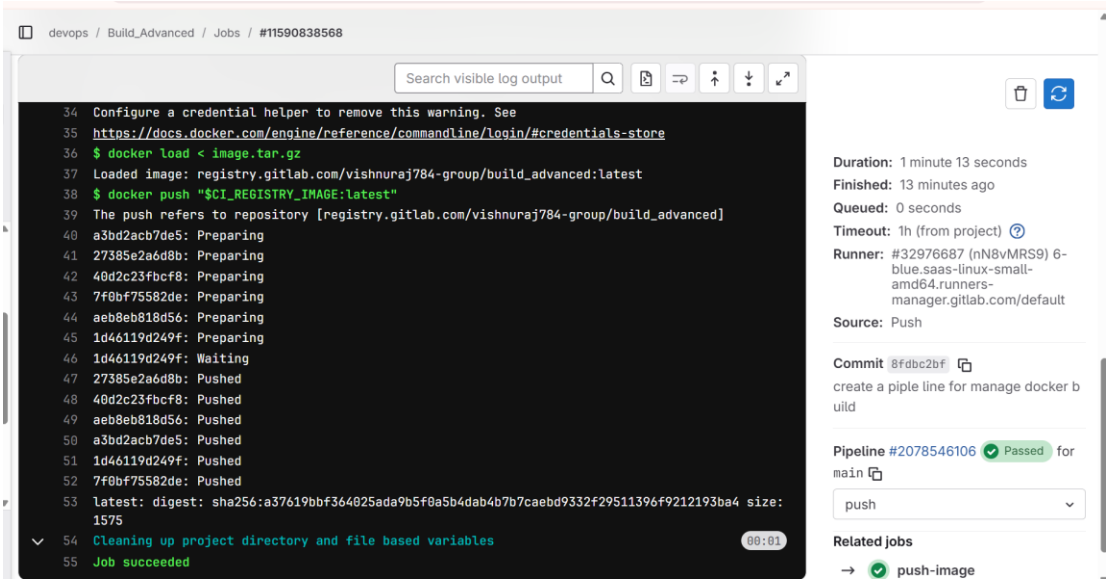
OUTPUT



build-image job



push-image job



Assignment 4: Manage Docker Build Locally and Using GitLab CI/CD with Shell and Docker Executors (Push to GitLab Container Registry)

Dockerfile

FROM openjdk:17-jdk-slim

WORKDIR /app

COPY target/java-docker-sample-1.0-SNAPSHOT.jar app.jar

CMD ["java", "-jar", "app.jar"]

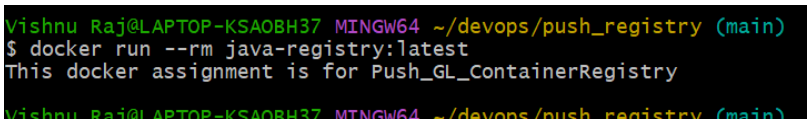
Build & Test Locally

Build image: -

```
docker build -t java-registry:latest .
```

Run container: -

```
docker run --rm java-registry:latest
```



GitLab CI/CD Configuration

```
stages:
  - build
  - docker_build

# Build using Shell Executor
build_job:
  stage: build
  tags:
    - shell
  script:
    - echo "Running Maven build locally with Shell executor..."
    - mvn clean package -DskipTests
  artifacts:
    paths:
      - target/*.jar

# Build Docker Image (Docker Executor)
docker_build:
  stage: docker_build
  image: docker:24.0.7
  services:
    - docker:24.0.7-dind
  variables:
    DOCKER_HOST: tcp://docker:2375/
    DOCKER_TLS_CERTDIR: ""
  tags:
    - docker
  script:
```

- echo "Logging in to GitLab Container Registry..."

- echo $CI_REGISTRY_PASSWORD | docker login -u $CI_REGISTRY_USER $CI_REGISTRY --password-stdin

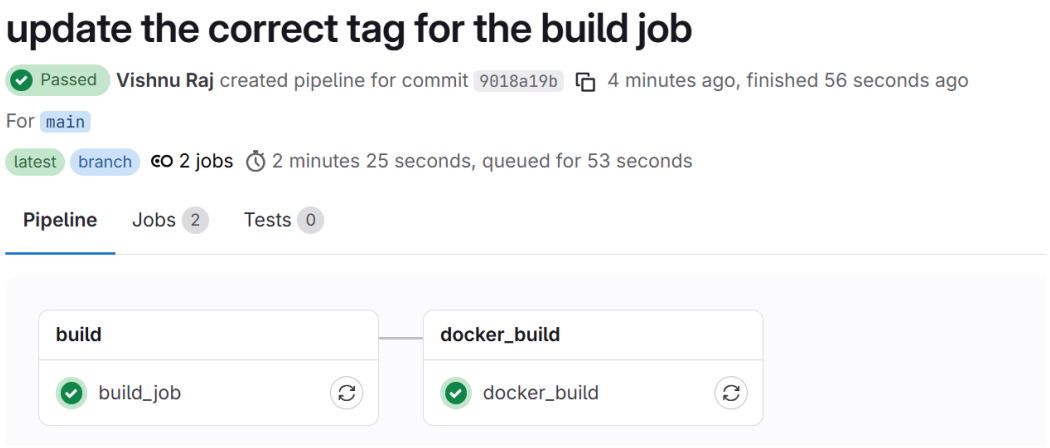
- echo "Building Docker image..."

- docker build -t $CI_REGISTRY_IMAGE:latest .

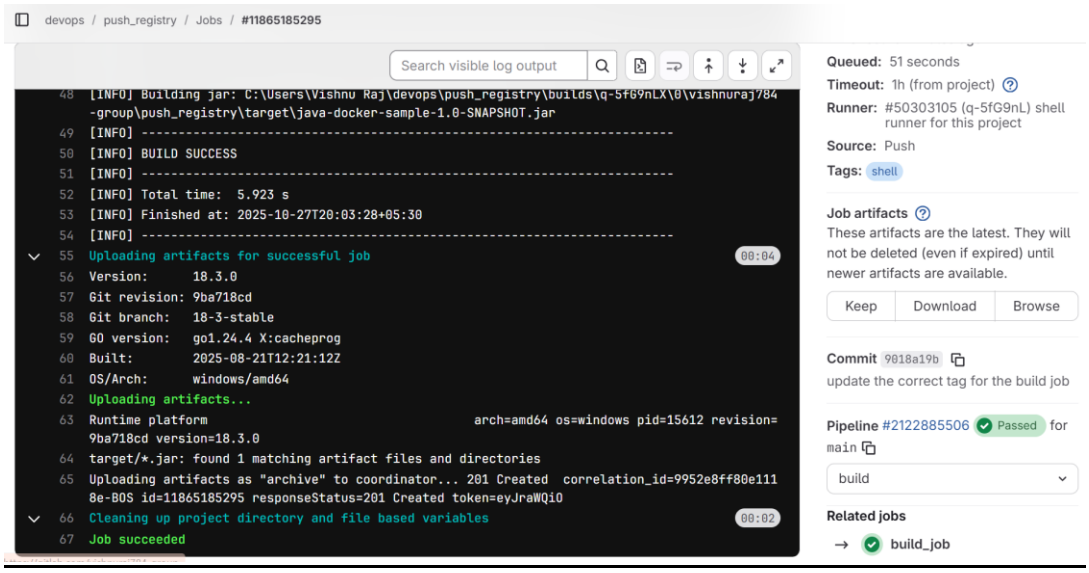
- echo "Pushing Docker image to GitLab Registry..."

- docker push $CI_REGISTRY_IMAGE:latest

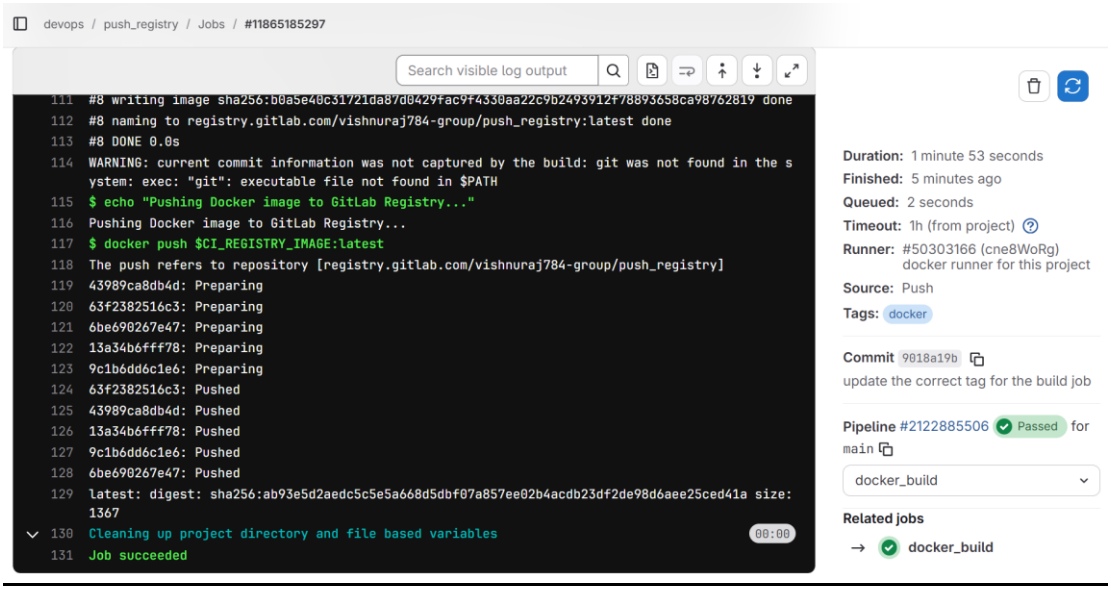
OUTPUT



Build job



Docker_build job



Assignment 5: Manage Docker Build Locally and Using GitLab CI/CD with Shell and Docker Executors

Dockerfile

```
FROM alpine:latest

COPY app.sh /usr/local/bin/app.sh

RUN chmod +x /usr/local/bin/app.sh

CMD ["app.sh"]
```

Build & Test Locally

Build image: -

```
docker build -t myapp:latest .
```

```
Vishnu Raj@LAPTOP-KSAOBH37 MINGW64 ~/devops/push_dockerhub (main)
$ docker build -t myapp:latest .
[+] Building 0.6s (8/8) FINISHED                                docker:desktop-linux
 => [internal] load build definition from Dockerfile                              0.0s
 => => transferring dockerfile: 143B                                              0.0s
 => [internal] load metadata for docker.io/library/alpine:latest                  0.0s
 => [internal] load .dockerignore                                                 0.0s
 => => transferring context: 2B                                                   0.0s
 => [internal] load build context                                                 0.0s
 => => transferring context: 129B                                                 0.0s
 => CACHED [1/3] FROM docker.io/library/alpine:latest@sha256:4bcff63911fcb444     0.0s
 => => resolve docker.io/library/alpine:latest@sha256:4bcff63911fcb4448bd4fda      0.0s
 => [2/3] COPY app.sh /usr/local/bin/app.sh                                       0.0s
 => [3/3] RUN chmod +x /usr/local/bin/app.sh                                      0.2s
 => exporting to image                                                            0.2s
 => => exporting layers                                                           0.1s
 => => exporting manifest sha256:306a7a1ef6633979e97756eb6190ff8163212f7545ae     0.0s
 => => exporting config sha256:dea7f2eb1e01ea47c4b0d605000bb10a69ed8088e1ff44     0.0s
 => => exporting attestation manifest sha256:855275ac4166c582771d022d4fdde67e     0.0s
 => => exporting manifest list sha256:ae30cdcc88a17f1c46e528a85a2cd3db53cb51c     0.0s
 => => naming to docker.io/library/myapp:latest                                   0.0s
 => => unpacking to docker.io/library/myapp:latest                                0.0s
```

Run container: -

docker run --rm myapp:latest

```
Vishnu Raj@LAPTOP-KSAOBH37 MINGW64 ~/devops/push_dockerhub (main)
$ docker run --rm myapp:latest
Hello from inside the Docker container!
Current date and time: Mon Oct  6 14:14:01 UTC 2025
```

GitLab CI/CD Configuration

stages:

- test
- build
- push

variables:

IMAGE_NAME: "myapp"

PUBLIC_REPO: "docker.io/vishnuraj784/myapp"

PRIVATE_REPO: "docker.io/vishnuraj784/myapp-private"

DOCKER_DRIVER: overlay2

DOCKER_TLS_CERTDIR: ""

DOCKER_HOST: tcp://docker:2375

test_script:

```
  stage: test
  tags:
    - shell-runner
  script:
    - echo "Running shell tests..."
    - bash app.sh
    - echo "Shell test passed!"

build_image:
  stage: build
  image: docker:24.0
  services:
    - name: docker:24.0-dind
      alias: docker
  tags:
    - docker-runner
  script:
    - echo "Building Docker image..."
    - docker build -t $IMAGE_NAME:latest .
    - docker images
    - echo "Build complete."

push_dockerhub:
  stage: push
  image: docker:24.0
  services:
    - docker:24.0-dind
  tags:
    - docker-runner
```

script:

- echo "Logging into Docker Hub..."

- echo "$DOCKER_HUB_PASSWORD" | docker login -u "$DOCKER_HUB_USERNAME" --password-stdin

- echo "Rebuilding Docker image for push..."

- docker build -t $IMAGE_NAME:latest .

- docker tag $IMAGE_NAME:latest $PUBLIC_REPO:latest

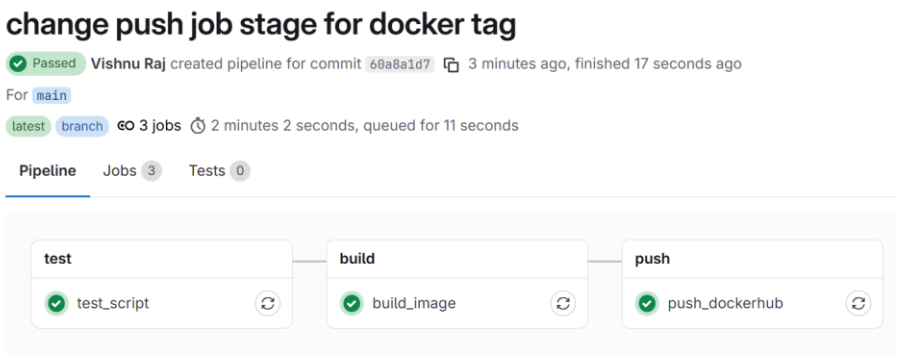
- docker tag $IMAGE_NAME:latest $PRIVATE_REPO:latest

- echo "Pushing images..."

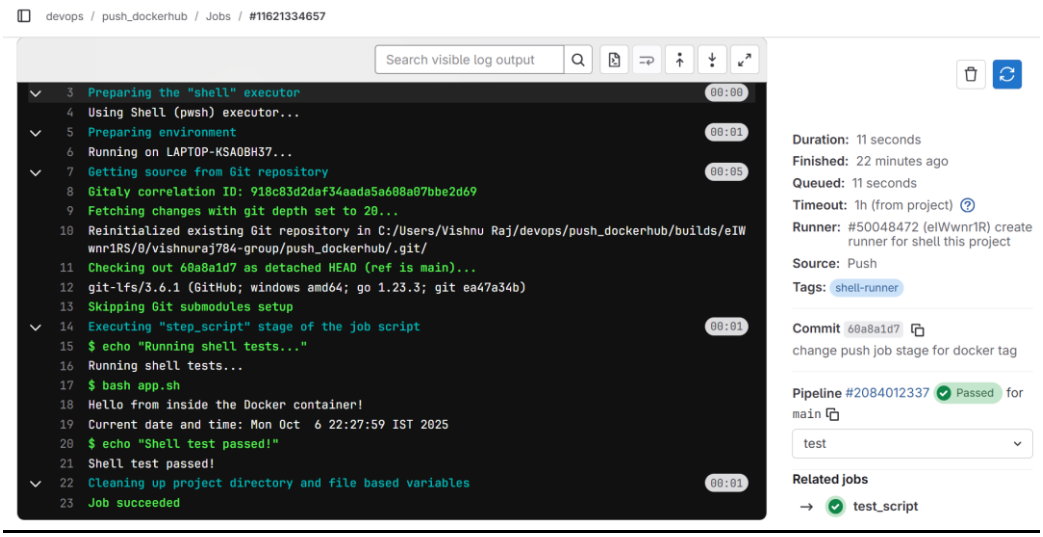
- docker push $PUBLIC_REPO:latest

- docker push $PRIVATE_REPO:latest

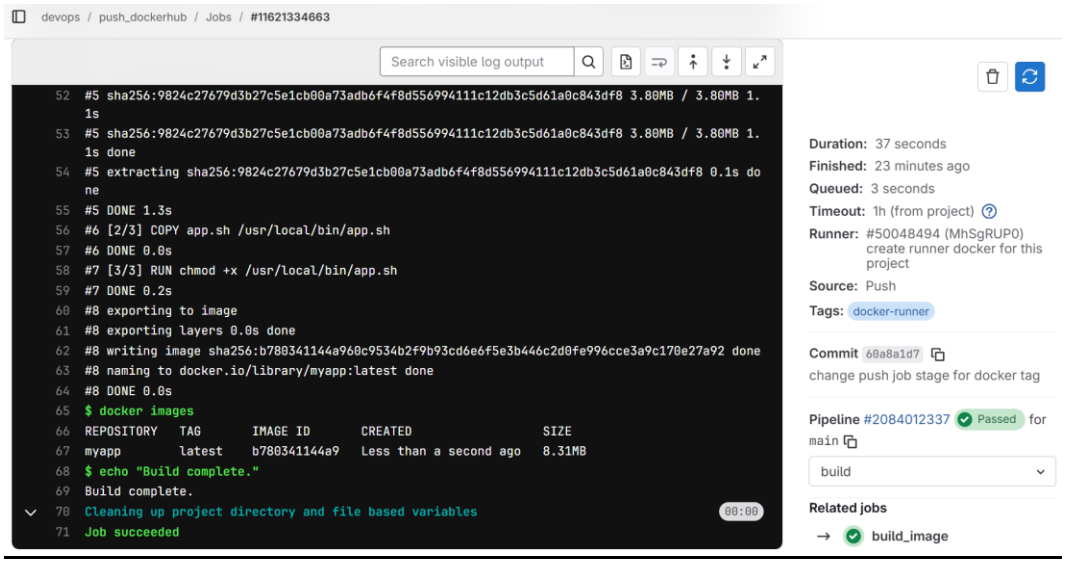
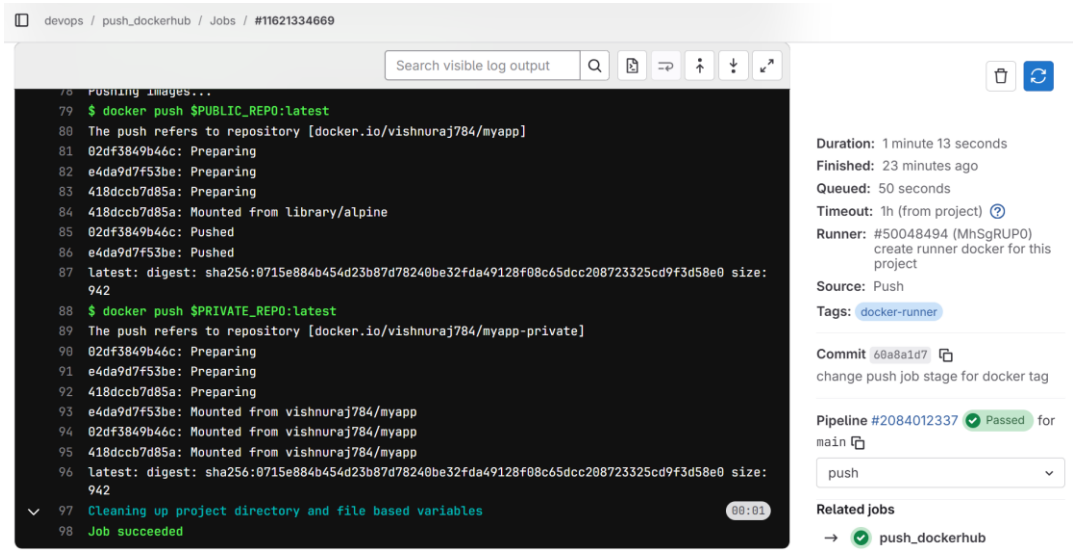
OUTPUT



test_script job



build_image job



push_dockerhub job