

Lab 7

BTI525 NCC

Fall 2025

INSTRUCTOR: ROBIN HUANG

Title: *Adding Docker to CI*

Course: Web Programming for the Cloud

Student Name: Minhaz Abedin

Student ID: 171424229

Email: mabedin1@myseneca.ca

Date: November 3rd, 2025

Github Link: https://github.com/minhazabedin53/bti525_lab7

Docker Hub Link: <https://hub.docker.com/r/abedin53/fragments>

Screenshot of the first time I ran Hadolint on your Dockerfile, showing any warnings or errors I had to correct:

The screenshot shows a GitHub CI pipeline for a repository. The summary indicates a failure in the "Dockerfile Lint" job. The logs for this job show the command being run and an error message from Hadolint:

```
1 ► Run hadolint/hadolint-action@v3.1.0
11   /usr/bin/docker run --name e6679b821cf8439f86a50ce7a4fbdc33_e7b050 --label 859912 --workdir /github/workspace --rm -e
    "INPUT_DOCKERFILE" -e "INPUT_CONFIG" -e "INPUT_RECURSIVE" -e "INPUT_OUTPUT_FILE" -e "INPUT_NO_COLOR" -e "INPUT_NO_FAIL"
    -e "INPUT_VERBOSE" -e "INPUT_FORMAT" -e "INPUT_FAILURE_THRESHOLD" -e "INPUT_OVERRIDE_ERROR" -e "INPUT_OVERRIDE_WARNING"
    -e "INPUT_OVERRIDE_INFO" -e "INPUT_OVERRIDE_STYLE" -e "INPUT_IGNORE" -e "INPUT_TRUSTED_REGISTRIES" -e "NO_COLOR" -e
    "HADOLINT_NOFAIL" -e "HADOLINT_VERBOSE" -e "HADOLINT_FORMAT" -e "HADOLINT_FAILURE_THRESHOLD" -e
    "HADOLINT_OVERRIDE_ERROR" -e "HADOLINT_OVERRIDE_WARNING" -e "HADOLINT_OVERRIDE_INFO" -e "HADOLINT_OVERRIDE_STYLE" -e
    "HADOLINT_IGNORE" -e "HADOLINT_TRUSTED_REGISTRIES" -e "HADOLINT_CONFIG" -e "HADOLINT_RECURSIVE" -e "HADOLINT_OUTPUT" -e
    "HOME" -e "GITHUB_JOB" -e "GITHUB_REF" -e "GITHUB_SHA" -e "GITHUB_REPOSITORY" -e "GITHUB_REPOSITORY_OWNER" -e
    "GITHUB_REPOSITORY_OWNER_ID" -e "GITHUB_RUN_ID" -e "GITHUB_RUN_NUMBER" -e "GITHUB_RETENTION_DAYS" -e
    "GITHUB_RUN_ATTEMPT" -e "GITHUB_ACTOR_ID" -e "GITHUB_ACTOR" -e "GITHUB_WORKFLOW" -e "GITHUB_HEAD_REF" -e
    "GITHUB_BASE_REF" -e "GITHUB_EVENT_NAME" -e "GITHUB_SERVER_URL" -e "GITHUB_API_URL" -e "GITHUB_GRAPHQL_URL" -e
    "GITHUB_REF_NAME" -e "GITHUB_REF_PROTECTED" -e "GITHUB_REF_TYPE" -e "GITHUB_WORKFLOW_REF" -e "GITHUB_WORKFLOW_SHA" -e
    "GITHUB_REPOSITORY_ID" -e "GITHUB_TRIGGERING_ACTOR" -e "GITHUB_WORKSPACE" -e "GITHUB_ACTION" -e "GITHUB_EVENT_PATH" -e
    "GITHUB_ACTION_REPOSITORY" -e "GITHUB_ACTION_REF" -e "GITHUB_PATH" -e "GITHUB_ENV" -e "GITHUB_STEP_SUMMARY" -e
    "GITHUB_STATE" -e "GITHUB_OUTPUT" -e "RUNNER_OS" -e "RUNNER_ARCH" -e "RUNNER_NAME" -e "RUNNER_ENVIRONMENT" -e
    "RUNNER_TOOL_CACHE" -e "RUNNER_TEMP" -e "RUNNER_WORKSPACE" -e "ACTIONS_RUNTIME_URL" -e "ACTIONS_RUNTIME_TOKEN" -e
    "ACTIONS_CACHE_URL" -e "ACTIONS_RESULTS_URL" -e "GITHUB_ACTIONS=true" -e "CI=true" -v
    "/var/run/docker.sock": "/var/run/docker.sock" -v "/home/runner/work/_temp/_github_home": "/github/home" -v
    "/home/runner/work/_temp/_github_workflow": "/github/workflow" -v
    "/home/runner/work/_temp/_runner_file_commands": "/github/file_commands" -v
    "/home/runner/work/bti525_lab6/bti525_lab6": "/github/workspace" 859912:e6679b821cf8439f86a50ce7a4fbdc33 "Dockerfile"
12 Error: Dockerfile:31 DL3018 warning: Pin versions in apk add. Instead of `apk add <package>` use `apk add <package>= <version>`
13 Hadolint output saved to: /dev/stdout
```

The log entry at line 12 is highlighted in red, indicating an error related to pinning apk versions.

Screenshot of a successful CI workflow, showing your new Dockerfile Lint and Docker Hub jobs succeeding.

The screenshot shows a GitHub CI pipeline for a repository. The summary indicates success for all jobs, including "Dockerfile Lint". The logs for the "Dockerfile Lint" job show the command being run and its success:

```
> ✓ Set up job
> ✓ Build hadolint/hadolint-action@v3.1.0
> ✓ Run actions/checkout@v4
> ✓ Run hadolint/hadolint-action@v3.1.0
> ✓ Post Run actions/checkout@v4
> ✓ Complete job
```

Screenshot of your main and latest Tags in Docker Hub, created by your GitHub Actions ci.yml workflow.

The screenshot shows the Docker Hub interface for the repository `docker/abedin53/fragments`. The `Tags` tab is selected. Three tags are listed:

- latest**: Last pushed 3 minutes ago by `abedin53`.
Digest: `f75532845912`, OS/ARCH: linux/amd64, Last pull: less than 1 day, Compressed size: 73.21 MB.
Command: `docker pull abedin53/fragments:latest`.
- main**: Last pushed 3 minutes ago by `abedin53`.
Digest: `f75532845912`, OS/ARCH: linux/amd64, Last pull: less than 1 day, Compressed size: 73.21 MB.
Command: `docker pull abedin53/fragments:main`.
- sha-6c0951597753997a6b775e8...**: Last pushed 3 minutes ago by `abedin53`.
Digest: `f75532845912`, OS/ARCH: linux/amd64, Last pull: less than 1 day, Compressed size: 73.21 MB.
Command: `docker pull abedin53/fragments:sha-6c0951597753997a6b775e8b1520fd04d3ca162e`.

making ecr fragments repo

The screenshot shows the AWS ECR console with a green success message: "Successfully created fragments". The repository list shows one private repository named `fragments`. The details for this repository are:

Repository name	URI	Created at	Tag immutability	Encryption type
<code>fragments</code>	<code>637423375311.dkr.ecr.us-east-1.amazonaws.com/fragments</code>	November 03, 2025, 22:38:46 (UTC-05)	Mutable	AES-256

repo secrets done

Repository secrets		New repository secret
Name	Last updated	
AWS_ACCESS_KEY_ID	21 minutes ago	
AWS_REGION	19 minutes ago	
AWS_SECRET_ACCESS_KEY	20 minutes ago	
AWS_SESSION_TOKEN	20 minutes ago	
DOCKERHUB_TOKEN	4 hours ago	
DOCKERHUB_USERNAME	4 hours ago	

Screenshot of a successful CD workflow, showing your AWS login, build and push jobs succeeding.

The screenshot shows a GitHub Actions workflow run for a release named 'Release v0.7.0 #1'. The workflow has completed successfully. The main summary indicates the job succeeded 8 minutes ago in 45s. The detailed log for the 'Build & Push to Amazon ECR' job is shown, listing the following steps and their execution times:

Step	Description	Time
1	Set up job	3s
2	Check out code	3s
3	Set up Docker Buildx	10s
4	Configure AWS Credentials	0s
5	Login to Amazon ECR	1s
6	Build and push to Amazon ECR	21s
7	Post Build and push to Amazon ECR	2s
8	Post Login to Amazon ECR	0s
9	Post Configure AWS Credentials	0s
10	Post Set up Docker Buildx	2s
11	Post Check out code	0s
12	Complete job	0s

Screenshot of my Amazon Elastic Container Registry showing my fragments image and tags (v0.7.0, latest), built and pushed via GitHub Actions CD workflow

Screenshot of Docker Hub image (:main) running locally and responding to check.

```
PS C:\Users\minha\Downloads\BTI\bt1525_lab6> docker pull abedin53/fragments:main
main: Pulling from abedin53/fragments
1cdfe5a93b96: Pull complete
bb87725ac545: Pull complete
85b237fcf654: Pull complete
8e48a0e23d52: Pull complete
7d3dcf32738b: Pull complete
Digest: sha256:c262fe5c098f41d4c2f7b80afe650511a584e9820606e3a828f12c2c18cdf064
Status: Downloaded newer image for abedin53/fragments:main
docker.io/abedin53/fragments:main
PS C:\Users\minha\Downloads\BTI\bt1525_lab6> docker run -d -p 8080:8080 abedin53/fragments:main
cfc3099736a21c231f587f45a0b904ed36e4d8f7f30afdd9f8919e6ec577dac6
```

- PS C:\Users\minha\Downloads\BTI\bt1525_lab6> docker ps

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
cfc3099736a2	abedin53/fragments:main	"docker-entrypoint.s..."	3 minutes ago	Up 3 minutes (healthy)	0.0.0.0:8080->8080/tcp, [::]:8080->8080/tcp focused_dhawan
- PS C:\Users\minha\Downloads\BTI\bt1525_lab6> curl.exe http://localhost:8080/
{"status": "ok", "author": "Minhaz Abedin mabedin1@myseneca.ca", "githubUrl": "https://github.com/minhazabedin53/fragments-service", "version": "0.7.0"}
- PS C:\Users\minha\Downloads\BTI\bt1525_lab6> []

setting up EC2 ECR

```
[ec2-user@ip-172-31-29-218 ~]$ docker login -u AWS -p $(aws ecr get-login-password --region us-east-1) 637423375311.dkr.ecr.us-eas
t-1.amazonaws.com
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
WARNING! Your password will be stored unencrypted in /home/ec2-user/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[ec2-user@ip-172-31-29-218 ~]$ docker pull 637423375311.dkr.ecr.us-east-1.amazonaws.com/fragments:v0.7.0
v0.7.0: Pulling from fragments
2d35ebdb57d9: Already exists
9682177f5dda: Already exists
e910c7b62cc7: Already exists
eb9e06b66aae: Already exists
7897658e7a6c: Full complete
a0839a77c9de: Full complete
bafa007ff489: Full complete
aadbf38f1f97: Full complete
ee71f35a55f3: Full complete
Digest: sha256:6b49fbded430c50e260219f49612426494e62da9f48e179764d7fba5eec2ec57
Status: Downloaded newer image for 637423375311.dkr.ecr.us-east-1.amazonaws.com/fragments:v0.7.0
637423375311.dkr.ecr.us-east-1.amazonaws.com/fragments:v0.7.0
```

EC2 running ECR image :v0.7.0 and passing health check.

```
[ec2-user@ip-172-31-29-218 ~]$ docker run -d --name fragments \
-e AUTH_STRATEGY=basic \
-e BASIC_AUTH_REALM=Fragments \
-e BASIC_AUTH_HTPASSWD=/app/tests/.htpasswd \
-e LOG_LEVEL=info \
-p 8080:8080 \
637423375311.dkr.ecr.us-east-1.amazonaws.com/fragments:v0.7.0
9fdcf2206bb8aa4dce555823d7718e4ec9ebdfbac12b4ecb693bbc2ce3619dfb
[ec2-user@ip-172-31-29-218 ~]$ curl http://localhost:8080/
{"status": "ok", "author": "Minhaz Abedin mabedin1@myseneca.ca", "githubUrl": "https://github.com/minhazabec
docker run -d --name fragments -e AUTH_STRATEGY=basic -e BASIC_AUTH_REALM=Fragments -e BASIC_AUTH_
HTPASSWD=/app/tests/.htpasswd -e LOGUN -d --name fragments -e AUTH_STRATEGY=basic -e BASIC_AUTH_
REALM=Fragments -e BASIC_AUTH
[ec2-user@ip-172-31-29-218 ~]$ }
```

protected route works

```
-user@ip-172-31-29-218 ~]$ curl -u user:pass123 -H "Content-Type: text/markdown" --data "# Hello from EC2" http://localhost:8080/v1/fragments
[ec2-user@ip-172-31-29-218 ~]$
[ec2-user@ip-172-31-29-218 ~]$ curl -u user:pass123 -H "Content-Type: text/markdown" \
--data "# Hello from EC2" \
http://localhost:8080/v1/fragments
{"status": "ok", "fragment": {"id": "ee42ac47-cd4b-4665-919e-91a74703e017", "ownerId": "04f8996da763b7a969b1028ee3007569eaf3a635486ddab2
11d512c85b9df8fb", "created": "2025-11-04T04:33:54.714Z", "updated": "2025-11-04T04:33:54.714Z", "type": "text/markdown", "size": 16}} [ec2
-user@ip-172-31-29-218 ~]$ curl -u user:pass123 http://localhost:8080/v1/fragments?expand=1
{"status": "ok", "fragments": [{"id": "f74c3752-5d56-4374-9b32-e1a84347f117", "ownerId": "04f8996da763b7a969b1028ee3007569eaf3a635486dda"}]
```