

Lappeenrannan teknillinen yliopisto  
School of Engineering

Software Development Skills DevOps, Online course

<Thong Dang >, <q7954>

**LEARNING DIARY, <Introduction to DevOps> MODULE**

## **Table of Contents**

<b>1</b>	<b>Version Control Learning</b>	<b>3</b>
<b>2</b>	<b>Container Technologies</b>	<b>12</b>
<b>3</b>	<b>Cloud Environments</b>	<b>18</b>
<b>4</b>	<b>CI/CD</b>	<b>20</b>

# LEARNING DIARY

## 1 VERSION CONTROL LEARNING

07.09.2024

In the section of “Version Control”, I learned how to deal with Git issues in file directories. Whenever I started with files using Git, I created a repository in a parent file with the command **git init**. Inside the parent file, we got various children files inside the parent file, whenever we updated or modified the children files, to keep track of these files, we needed to stage these files in the repository by running the command **git add .** in the parent file. Then, we could use **git commit -m commit-name** to create snapshots of commits of these files along with messages what you did for these files. At the initial, you created the first commit of the head of the master branch. In addition, whenever you committed the new changes of the files into the repository, the head moved forward but we could move back because we got references from the previous commits. The basic Git workflow describes a fundamental technique for version control in software development. It starts with editing files in the working directory, which is your local project space. Once modifications are made, they are staged and marked for inclusion in the following commit. Staging allows you to commit to particular changes while maintaining flexibility. Finally, the staged modifications are committed, resulting in a new Git snapshot that records the project's present state. This technique is critical for keeping a systematic history of modifications and facilitating collaboration in development initiatives. However, in a working life, whenever we conducted a professional software development process, we usually did in a non-linear ways when many people worked in the same codebase with working with different features simultaneously. Simply, you could do and develop own features with a branching method. To be precise, you simply created your own branch with the command **git branch branch-name** and used **git checkout** to go into that branch. In this working branch, you were free to update new features without affecting the files or codes of the master branch with own history. After you had done own feature updates and would like to merge the history of own working branch into the history of the master branch, you simply went back to the master branch with **git checkout master** and ran the command **git merge branch-name**. Now, the tip of the feature branch became the head of the master branch.

I started with the implementations of Git demonstration videos. At first, I did the following ways as illustrated in Basic Operations in the Version Control demonstration video. In this video, I learned how to stage and commit the README file into the repository as below.

```
[thongdang@thongs-MacBook-Pro git_practice % git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /Users/thongdang/Desktop/dev_ops_course/git_
practice/.git/
[thongdang@thongs-MacBook-Pro git_practice % git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
[thongdang@thongs-MacBook-Pro git_practice % nano README
[thongdang@thongs-MacBook-Pro git_practice % ls -la
total 8
drwxr-xr-x 4 thongdang staff 128 Sep  7 10:07 .
drwxr-xr-x 4 thongdang staff 128 Sep  7 10:03 ..
drwxr-xr-x@ 9 thongdang staff 288 Sep  7 10:06 .git
-rw-r--r--  1 thongdang staff     6 Sep  7 10:07 README
[thongdang@thongs-MacBook-Pro git_practice % git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    README

nothing added to commit but untracked files present (use "git add" to track)
[thongdang@thongs-MacBook-Pro git_practice % git add README
[thongdang@thongs-MacBook-Pro git_practice % git commit -m "added a README file"
[master (root-commit) 0723c2a] added a README file
  1 file changed, 2 insertions(+)
  create mode 100644 README
[thongdang@thongs-MacBook-Pro git_practice % git status
On branch master
nothing to commit, working tree clean
```

In the second demonstration video, I followed to learn how to modify files and committed with new changes for README and Express files as below.

```

[thongdang@thongs-MacBook-Pro git_practice % nano app.js
[thongdang@thongs-MacBook-Pro git_practice % git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    app.js

nothing added to commit but untracked files present (use "git add" to track)
[thongdang@thongs-MacBook-Pro git_practice % git add app.js
[thongdang@thongs-MacBook-Pro git_practice % nano README
[thongdang@thongs-MacBook-Pro git_practice % git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   app.js

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   README

[thongdang@thongs-MacBook-Pro git_practice % nano app.js
[thongdang@thongs-MacBook-Pro git_practice % git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   app.js

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   README
    modified:   app.js

[thongdang@thongs-MacBook-Pro git_practice % git add app.js
[thongdang@thongs-MacBook-Pro git_practice % git add README
[thongdang@thongs-MacBook-Pro git_practice % git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   README
    new file:   app.js

[thongdang@thongs-MacBook-Pro git_practice % git commit -m "added app.js and modified README file"
[master 1f080b4] added app.js and modified README file
  2 files changed, 21 insertions(+), 1 deletion(-)
  create mode 100644 app.js
[thongdang@thongs-MacBook-Pro git_practice % git log
commit 1f080b4fc9e5c77d86b1d39f6e1e6219ac00242b (HEAD -> master)
Author: Thong Dang <duythong2708@gmail.com>
Date:   Sat Sep 7 11:18:08 2024 +0300

    added app.js and modified README file

commit 0723c2a2f8f54cb9d1d45c3382b9516f42cd8778
Author: Thong Dang <duythong2708@gmail.com>
Date:   Sat Sep 7 10:08:18 2024 +0300

    added a README file

```

In the third video, I learned how to see changes in more details in README file with **git diff** command as below.

```
[thongdang@thongs-MacBook-Pro git_practice % nano README
[thongdang@thongs-MacBook-Pro git_practice % git diff README
diff --git a/README b/README
index 0e90bad..524612f 100644
--- a/README
+++ b/README
@@ -2,3 +2,5 @@ This is a readme file.

The project now includes a app.js file.

+Now, it is possible to see diff.
+
```

In the fourth demonstration video, I learned how to rename and delete the file, and how to commit those changes in the repository as below.

```
[thongdang@thongs-MacBook-Pro git_practice % nano new_test_file
[thongdang@thongs-MacBook-Pro git_practice % git add new_test_file
[thongdang@thongs-MacBook-Pro git_practice % git commit -m "Updated new test file"
[master 9fefdf52] Updated new test file
 1 file changed, 1 insertion(+)
[thongdang@thongs-MacBook-Pro git_practice % git status
On branch master
nothing to commit, working tree clean
[thongdang@thongs-MacBook-Pro git_practice % git mv new_test_file test_file
[thongdang@thongs-MacBook-Pro git_practice % git commit -m "Updated name of test file"
[master f612886] Updated name of test file
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename new_test_file => test_file (100%)
[thongdang@thongs-MacBook-Pro git_practice % git diff test_file
[thongdang@thongs-MacBook-Pro git_practice % git log
commit f612886303612968ff22fd48f92232ea6c1b4cee (HEAD -> master)
Author: Thong Dang <duythong2708@gmail.com>
Date:   Sat Sep 7 11:38:59 2024 +0300

        Updated name of test file

commit 9fefdf52ed8a7e1e70a2320a9b93f8883670e9af7
Author: Thong Dang <duythong2708@gmail.com>
Date:   Sat Sep 7 11:38:31 2024 +0300

        Updated new test file

commit 694e88e6c2df003e88435126590992dbe48cb380
Author: Thong Dang <duythong2708@gmail.com>
Date:   Sat Sep 7 11:37:40 2024 +0300

        Updated new README

commit c381bf026272205be04dc1e4d26dd045f2b8ca0e
Author: Thong Dang <duythong2708@gmail.com>
Date:   Sat Sep 7 11:35:01 2024 +0300

        Rename the test file

commit 7b855b80afff0d14b7f1cd3c2ca0b188ffab8b33
Author: Thong Dang <duythong2708@gmail.com>
Date:   Sat Sep 7 11:33:58 2024 +0300

        Added a new test file

commit 1f080b4fc9e5c77d86b1d39f6e1e6219ac00242b
Author: Thong Dang <duythong2708@gmail.com>
Date:   Sat Sep 7 11:18:08 2024 +0300

        added app.js and modified README file

commit 0723c2a2f8f54cb9d1d45c3382b9516f42cd8778
Author: Thong Dang <duythong2708@gmail.com>
Date:   Sat Sep 7 10:08:18 2024 +0300

        added a README file
[thongdang@thongs-MacBook-Pro git_practice % git rm test_file
rm 'test_file'
[thongdang@thongs-MacBook-Pro git_practice % git commit -m "Deleted test_file"
[master 82cb94c] Deleted test_file
 1 file changed, 2 deletions(-)
 delete mode 100644 test_file
[thongdang@thongs-MacBook-Pro git_practice % git log
commit 82cb94cc5491fd9d7f1733ff62e638f2e0ca1d89 (HEAD -> master)
Author: Thong Dang <duythong2708@gmail.com>
Date:   Sat Sep 7 11:40:09 2024 +0300

        Deleted test_file

commit f612886303612968ff22fd48f92232ea6c1b4cee
```

In the fifth demonstration video, I learned how to make changes for files in a sub-branch instead of the master branch as below.

```
[thongdang@thongs-MacBook-Pro git_practice % git branch
* master
[thongdang@thongs-MacBook-Pro git_practice % git branch feature_1
[thongdang@thongs-MacBook-Pro git_practice % git branch
  feature_1
* master
[thongdang@thongs-MacBook-Pro git_practice % git checkout feature_1
Switched to branch 'feature_1'
[thongdang@thongs-MacBook-Pro git_practice % git branch
* feature_1
  master
[thongdang@thongs-MacBook-Pro git_practice % nano app.js
[thongdang@thongs-MacBook-Pro git_practice % git add app.js
[thongdang@thongs-MacBook-Pro git_practice % git commit -m "Made changes for app.js in branch"
[feature_1 5f7649d] Made changes for app.js in branch
  1 file changed, 4 insertions(+)
[thongdang@thongs-MacBook-Pro git_practice % git log
commit 5f7649d4b45130b85d20568e419f29c07e78553b (HEAD -> feature_1)
Author: Thong Dang <duythong2708@gmail.com>
Date:   Sat Sep 7 11:53:13 2024 +0300

  Made changes for app.js in branch
```

In the sixth demonstration video, I learned how to merge changes from sub-branch to the master branch and delete that sub-branch after merging as below.

```
added a README file
[thongdang@thongs-MacBook-Pro git_practice % git branch
* feature_1
  master
[thongdang@thongs-MacBook-Pro git_practice % git checkout master
Switched to branch 'master'
[thongdang@thongs-MacBook-Pro git_practice % git merge feature_1
Updating 82cb94c..5f7649d
Fast-forward
  app.js | 4 +++
  1 file changed, 4 insertions(+)
[thongdang@thongs-MacBook-Pro git_practice % nano app.js
[thongdang@thongs-MacBook-Pro git_practice % git branch -d feature_1
Deleted branch feature_1 (was 5f7649d).
[thongdang@thongs-MacBook-Pro git_practice % git branch
* master
```

In the seventh demonstration video, I learned how to connect with remote repository. However, it took me a little bit of time to connect to Gitlab and set up the SSH key to connect as below. Finally, I could push my code to the Gitlab remote repository.

```

thongdang@thongs-MacBook-Pro git_practice_test % ssh-keygen -t rsa -b 4096 -C "t1daph00@students.oamk.fi"
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/thongdang/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/thongdang/.ssh/id_rsa
Your public key has been saved in /Users/thongdang/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:RoPm+/SRHo2d1Z12iHxHP7GoB3mDHEUujvpWlpffmZE t1daph00@students.oamk.fi
The key's randomart image is:
+---[RSA 4096]---+
|          o..o |
|         . o+. . |
|        . ++*.+ + |
|       +oBoX =.|
|      S...B.=oB|
|     . =,=oEo|
|    .. o ooB|
|     .. .=o|
|      .. |
+---[SHA256]---+
thongdang@thongs-MacBook-Pro git_practice_test % eval "$(ssh-agent -s)"
Agent pid 2353
thongdang@thongs-MacBook-Pro git_practice_test % ssh-add ~/ssh/id_rsa
Identity added: /Users/thongdang/.ssh/id_rsa (t1daph00@students.oamk.fi)
thongdang@thongs-MacBook-Pro git_practice_test % cat ~/ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAQADQABAAACAOQDmJ3e0u/NF1a5TwKu1YLfJf4o0kMNCLoSj04ThUWFTx0Dm9+6xFSHQX4u317wnS24fsqfwghSgANzbX091YHURfPQHU
V6B/AB8D0o0aRpj0dGdQfZxvLPkwXkegONfQetQ8wtwua56dzXjWQsNDdkZMM584zjetk/pstfAR1Wd0yRaUNQKz/kHffzKdbXf4FBMWZHGX7cqIyi2bmGjy5c42sM+5Gm0lLMfGPEZ53L0yK/GWgALe/oPJ1
CHwSGDdKTJd233KVUomYTikbslhG1NRUnUL22GOsUOxF6tEymVPxUwEms+KR980E6p4aDt3LQwR163Q0U05gdK+3iUMNPnCPFCIfJB9tZB1bnNq2Fzk67buLa7ac60gRvEs+Nx1QEkoLHwtHPSNHvSzV7
4r3Kt1+EmmHy84ngLO6GJBFB1jb5+s4qBqE9FG9x8jKwLT2UGySGcXj2g7EBhm8CbH1SmAXT9HU6h6z1GwG04Mytl37pwK1020wvfzmgh8FZwaYjM6yDb0soB2F/2r9+Z5tneeZ9eCf+J6MwPpYLz/7ypn
y89MSWctMC2aTSPPgciu3X4q/gsD2MXN092Ttdd8WAsgL/4fFQQF+HPoJTqzRlabAznVtvkg/BCj/2a9LR7jw== t1daph00@students.oamk.fi
thongdang@thongs-MacBook-Pro git_practice_test % git push -u origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 1.31 KiB | 1.31 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To gitlab.com:iut-study/version-control-learning-test.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

```

## V Version Control Learning Test

 master  +    

 Commit all files  
Thong Dang authored 6 minutes ago   

Name	Last commit	Last update
 README	Commit all files	6 minutes ago
 app.js	Commit all files	6 minutes ago
 test_file	Commit all files	6 minutes ago

 README

This is a readme file.

The project now includes a app.js file.

Now, it is possible to see diff. Check log!

In the eighth demonstration video, I learned how to create a merge request for the sub-branch with new changes to the remote repository and delete the sub-branch in the local repository after merging as below. It is a valuable lesson.

```

thongdang@thongs-MacBook-Pro git_practice_test % git branch
* master
thongdang@thongs-MacBook-Pro git_practice_test % git branch app_js_feature
thongdang@thongs-MacBook-Pro git_practice_test % git branch
  app_js_feature
* master
thongdang@thongs-MacBook-Pro git_practice_test % git checkout app_js_feature
Switched to branch 'app_js_feature'
thongdang@thongs-MacBook-Pro git_practice_test % ls -la
total 24
drwxr-xr-x  6 thongdang  staff  192 Sep  7 14:18 .
drwxr-xr-x  6 thongdang  staff  192 Sep  7 16:01 ..
drwxr-xr-x 12 thongdang  staff  384 Sep  8 10:39 .git
-rw-r--r--  1 thongdang  staff  109 Sep  7 11:35 README
-rw-r--r--  1 thongdang  staff  485 Sep  7 12:07 app.js
-rw-r--r--  1 thongdang  staff  21 Sep  7 14:18 test_file
thongdang@thongs-MacBook-Pro git_practice_test % nano app.js
thongdang@thongs-MacBook-Pro git_practice_test % git add app.js
thongdang@thongs-MacBook-Pro git_practice_test % git commit -m "Added feature to appjs"
[app_js_feature dd94ae7] Added feature to appjs
 1 file changed, 4 insertions(+)
thongdang@thongs-MacBook-Pro git_practice_test % git push --set-upstream origin app_js_feature
Enumerating objects: 100%, 5/5, done.
Counting objects: 100%, 5/5, done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1018 bytes | 1018.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote:
remote: To create a merge request for app_js_feature, visit:
remote: https://gitlab.com/lut-study/version-control-learning-test/-/merge_requests/new?merge_request%5Bsource_branch%5D=app_js_feature
remote:
To gitlab.com:lut-study/version-control-learning-test.git
 * [new branch]      app_js_feature -> app_js_feature
branch 'app_js_feature' set up to track 'origin/app_js_feature'.
thongdang@thongs-MacBook-Pro git_practice_test % git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
thongdang@thongs-MacBook-Pro git_practice_test % git pull
remote: Enumerating objects: 1, done.
remote: Counting objects: 100%, (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (1/1), 292 bytes | 292.00 KiB/s, done.
From gitlab.com:lut-study/version-control-learning-test
 c43b8de..66fe744  master    -> origin/master
Updating c43b8de..66fe744
Fast-forward
 app.js | 4 +---
 1 file changed, 4 insertions(+)
thongdang@thongs-MacBook-Pro git_practice_test % git branch
  app_js_feature
* master
thongdang@thongs-MacBook-Pro git_practice_test % git branch -d app_js_feature

```

You pushed to [app\\_js\\_feature](#) just now

[Create merge request](#)

gitlab.com:lut-study/version-control-learning-test /

Compare History Find file Edit Code

**Added feature to appjs**  
Thong Dang authored 1 minute ago

Unverified dd94ae70

Name	Last commit	Last update
README	Commit all files	20 hours ago
app.js	Added feature to appjs	1 minute ago
test_file	Commit all files	20 hours ago

**README**

This is a readme file.

The project now includes a app.js file.

Now, it is possible to see diff. Check log!

## Added feature to appjs

Open Pham Duy Thong Dang requested to merge [app\\_js\\_feature](#) into [master](#) just now

Edit Code :

Overview 0 Commits 0 Pipelines 0 Changes 1

Add a to do

Test sub-branch

0 0

① Before you can run pipelines, we need to verify your account.

We won't ask you for this information again. It will never be used for marketing purposes.

[Verify my account](#)

8<sup>+</sup> [Approve](#) Approval is optional

Ready to merge!

Delete source branch  Squash commits  Edit commit message

1 commit and 1 merge commit will be added to master.

[Merge](#)

0 Assignees

Edit

None - assign yourself

0 Reviewers

Edit

None - assign yourself

Labels

Edit

None

Milestone

Edit

None

Time tracking

+

No estimate or time spent

1 Participant



## Activity

All activity

Preview

Write a comment or drag your files here...

Switch to rich text editing

## Added feature to appjs

Merged Pham Duy Thong Dang requested to merge [app\\_js\\_feature](#) into [master](#) just now

Edit Code :

Overview 0 Commits 0 Pipelines 0 Changes 1

Add a to do

Test sub-branch

0 0

① Before you can run pipelines, we need to verify your account.

We won't ask you for this information again. It will never be used for marketing purposes.

[Verify my account](#)

8<sup>+</sup> Approval is optional

Merged by Pham Duy Thong Dang just now

[Revert](#) [Cherry-pick](#)

Merge details

- Changes merged into master with [66fe7444](#).
- Deleted the source branch.

0 Assignees

Edit

None - assign yourself

0 Reviewers

Edit

None - assign yourself

Labels

Edit

None

Milestone

Edit

None

Time tracking

+

No estimate or time spent

1 Participant



## Activity

All activity

• Pham Duy Thong Dang mentioned in commit [66fe7444](#) just now

Pham Duy Thong Dang merged just now

Preview

Write a comment or drag your files here...

Switch to rich text editing

In the last demonstration video, I learned how to deal with conflicts when we committed new changes in the sub-branch and merged into the master branch which included the new changes as the same line in the sub-branch. Actually, I had to go into the changed files to decide which to keep and which not to keep in the changed files to resolve the conflicts before committing them into the repository.

```
[thongdang@thongs-MacBook-Pro git_practice_test % git branch feature_4_readme
[thongdang@thongs-MacBook-Pro git_practice_test % git checkout feature_4_readme
Switched to branch 'feature_4_readme'
[thongdang@thongs-MacBook-Pro git_practice_test % nano README
[thongdang@thongs-MacBook-Pro git_practice_test % nano README
[thongdang@thongs-MacBook-Pro git_practice_test % nano README
[thongdang@thongs-MacBook-Pro git_practice_test % git add README
[thongdang@thongs-MacBook-Pro git_practice_test % git commit -m "README subbranch change"
[feature_4_readme 8c84c76] README subbranch change
  1 file changed, 2 insertions(+)
[thongdang@thongs-MacBook-Pro git_practice_test % nano README
[thongdang@thongs-MacBook-Pro git_practice_test % git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 3 commits.
  (use "git push" to publish your local commits)
[thongdang@thongs-MacBook-Pro git_practice_test % nano README
[thongdang@thongs-MacBook-Pro git_practice_test % git add README
[thongdang@thongs-MacBook-Pro git_practice_test % git commit -m "Change in readme in master"
[master 7a139e1] Change in readme in master
  1 file changed, 2 insertions(+)
[thongdang@thongs-MacBook-Pro git_practice_test % git merge feature_4_readme
Auto-merging README
CONFLICT (content): Merge conflict in README
Automatic merge failed; fix conflicts and then commit the result.
[thongdang@thongs-MacBook-Pro git_practice_test % nano README
[thongdang@thongs-MacBook-Pro git_practice_test % git merge feature_4_readme
error: Merging is not possible because you have unmerged files.
hint: Fix them up in the work tree, and then use 'git add/rm <file>'
hint: as appropriate to mark resolution and make a commit.
fatal: Exiting because of an unresolved conflict.
[thongdang@thongs-MacBook-Pro git_practice_test % git add README
[thongdang@thongs-MacBook-Pro git_practice_test % git commit -m "resolve conflicts"
[master 1cff830] resolve conflicts
```

In conclusion, the section “Version Control” gave me a good review for the knowledge how to deal with Git even though I used them quite a lot at my university projects and at my workplace. There were many scenarios how to deal with Git when working as software developers in a team.

## 2 CONTAINER TECHNOLOGIES

9.9.2024

In this second topic relating to Docker, I used Docker many times during my studies and this section is also a good chance to revise the knowledge. In the first demonstration video about “Overview of Docker”, I revised that Docker was mainly used to pack software applications as containers including source code, runtime environments, and dependencies that could be shared to run in any other runtime environments. Then, coming to the second demonstration video about “Pulling and running 3<sup>rd</sup> party container images”, I learned how to pull images from remote repositories in Docker Hub and run them in my local environments. Below are examples of pulling and running images Hello World and ubuntu in my local environments on Docker playground.

```
$ docker pull
"docker pull" requires exactly 1 argument.
See 'docker pull --help'.

Usage: docker pull [OPTIONS] NAME[:TAG|@DIGEST]

Download an image from a registry
[node1] (local) root@192.168.0.28 ~
$ docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
hello-world    latest    d2c94e258dcb  16 months ago  13.3kB
[node1] (local) root@192.168.0.28 ~
$ docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

```

[node1] (local) root@192.168.0.28 ~
$ docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
31e907dcc94a: Pull complete
Digest: sha256:8a37d68f4f73ebf3d4efafbcf66379bf3728902a8038616808f04e34a9ab63ee
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
[node1] (local) root@192.168.0.28 ~
$ docker run -it ubuntu bash
root@161cc29911c8:/# ls -la
total 0
drwxr-xr-x    1 root root   6 Sep  9 07:07 .
drwxr-xr-x    1 root root   6 Sep  9 07:07 ..
-rw xr-xr-x    1 root root   0 Sep  9 07:07 .dockerenv
lrwxrwxrwx    1 root root   7 Apr 22 13:08 bin -> usr/bin
drwxr-xr-x    2 root root   6 Apr 22 13:08 boot
drwxr-xr-x    5 root root 360 Sep  9 07:07 dev
drwxr-xr-x    1 root root   66 Sep  9 07:07 etc
drwxr-xr-x    3 root root  20 Aug  1 12:03 home
lrwxrwxrwx    1 root root   7 Apr 22 13:08 lib -> usr/lib
lrwxrwxrwx    1 root root   9 Apr 22 13:08 lib64 -> usr/lib64
drwxr-xr-x    2 root root   6 Aug  1 11:59 media
drwxr-xr-x    2 root root   6 Aug  1 11:59 mnt
drwxr-xr-x    2 root root   6 Aug  1 11:59 opt
dr-xr-xr-x 1066 root root   0 Sep  9 07:07 proc
drwx----- 2 root root  37 Aug  1 12:03 root
drwxr-xr-x    4 root root  33 Aug  1 12:03 run
lrwxrwxrwx    1 root root   8 Apr 22 13:08 sbin -> usr/sbin
drwxr-xr-x    2 root root   6 Aug  1 11:59 srv
dr-xr-xr-x 13 root root   0 Aug 22 08:00 sys
drwxrwxrwt  2 root root   6 Aug  1 12:03 tmp
drwxr-xr-x 12 root root 133 Aug  1 11:59 usr
drwxr-xr-x 11 root root 139 Aug  1 12:03 var
root@161cc29911c8:/# exit
exit

```

In the third demonstration video, I learned about how to create a Docker file for Express application. To be precise, at first, I created the Express project and pushed changes into the Gitlab remote repository for cloning later on from Docker playground as below.

```

git: 'ignore' is not a git command. See 'git --help'.
thongdang@thongs-MacBook-Pro docker_practice % git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initial empty Git repository in /Users/thongdang/Desktop/dev_ops_course/docker_practice/.git/
thongdang@thongs-MacBook-Pro docker_practice % git remote -v
thongdang@thongs-MacBook-Pro docker_practice % git remote add origin git@gitlab.com:lut-study/container-technologies.git
thongdang@thongs-MacBook-Pro docker_practice % git add .
thongdang@thongs-MacBook-Pro docker_practice % git commit -m "Initial commit"
[master (root-commit) f755744] Initial commit
 5 files changed, 736 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 README.md
 create mode 100644 app.js
 create mode 100644 package-lock.json
 create mode 100644 package.json
thongdang@thongs-MacBook-Pro docker_practice % git push -u origin master
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 8.55 KiB | 4.28 MiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To gitlab.com:lut-study/container-technologies.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

```

Then, as mentioned, I cloned the Express project from Gitlab. There were some problems with my Gitlab authentication, but it was still done as below.

```
$ git clone https://gitlab.com/lut-study/container-technologies.git
Cloning into 'container-technologies'...
Username for 'https://gitlab.com': ThongDang27081998
Password for 'https://ThongDang27081998@gitlab.com':
remote: HTTP Basic: Access denied. The provided password or token is incorrect or your account has 2FA enabled and you must use a personal access token instead
of a password. See https://gitlab.com/help/topics/git/troubleshooting_git#error-on-git-fetch-http-basic-access-denied
fatal: Authentication failed for 'https://gitlab.com/lut-study/container-technologies.git/'
[node1] (local) root@192.168.0.28
```

Then, I added contents into the Dockerfile in the Express project to build an Docker image. However, I got a problem with the command **npm ci** in **RUN** command in the Dockerfile. Then, I changed into **npm install** and it worked to build the docker image for the Express project as you can see from below images.

```
[node1] (local) root@192.168.0.28 ~/container-technologies
$ cat Dockerfile

FROM node:12

WORKDIR /app

COPY package*.json .

RUN npm ci

COPY . .

EXPOSE 3000

CMD [ "npm", "start" ]
```

```
[node1] (local) root@192.168.0.28 ~/container-technologies
$ docker build .
[+] Building 101.5s (8/9)
=> [internal] load build definition from Dockerfile                               docker:default
=> => transferring dockerfile: 145B                                              0.1s
=> [internal] load .dockerignore                                                 0.0s
=> => transferring context: 2B                                                 0.1s
=> [internal] load metadata for docker.io/library/node:12                      0.0s
=> [1/5] FROM docker.io/library/node:12@sha256:01627afeb110b3054ba4a140      97.2s
=> => resolve docker.io/library/node:12@sha256:01627afeb110b3054ba4a1405   0.0s
=> => sha256:3a69ea1270dbf4ef20477361be4b7a43400e559c6ab 2.21kB / 2.21kB  0.0s
=> => sha256:f5196cdf25181bc7e4411865a2e002932b7b6b0ff 45.43MB / 45.43MB  1.6s
=> => sha256:01627afeb110b3054ba4a1405541ca095c8bfcalcb6f2be 776B / 776B  0.0s
=> => sha256:6c8de432fc7f7d8c58899f61982d1662ec6b73fb3ef 7.68kB / 7.68kB  0.0s
=> => sha256:f44e4bdb3a6c1325cc4d40e585ed7a759127c0c87b0 4.34MB / 4.34MB  0.4s
=> => sha256:9bed1e86f01ee95c76d2c8b4385a47ae336e6d293 11.30MB / 11.30MB  0.7s
=> => sha256:2f75d131f4060950dd6cc1f580e2fa5504ece8d69 49.77MB / 49.77MB  3.5s
=> => sha256:07dff4ad21ebdb3ce3e329699663b2f81af701 214.48MB / 214.48MB  16.4s
=> => sha256:e0ac4f13b766d321acc3b650d3d23b82828995711f6 4.19kB / 4.19kB  1.7s
=> => sha256:df2c3b2eb7cc63351bb32f26457bbe0402af80825 23.70MB / 23.70MB  3.2s
=> => extracting sha256:f5196cdf25181bc7e4411865a2e002932b7b6b0ffce787c 18.3s
=> => sha256:efe636eac583776a8a114d50fef15bc65b648f3d2bb 2.34MB / 2.34MB  3.5s
=> => sha256:fe17849545bb51455d3f7c8773ded2dbb1d6668a85bd005 464B / 464B  3.6s
=> => extracting sha256:9bed1e86f01ee95c76d2c8b4385a47ae336e6d293afade93 3.1s
=> => extracting sha256:f44e4bdb3a6c1325cc4d40e585ed7a759127c0c87b0388ec 1.0s
=> => extracting sha256:2f75d131f4060950dd6cc1f580e2fa5504ece8d692113a9 14.9s
=> => extracting sha256:07dff4ad21ebdb3ce3e329699663b2f81af70152453025f 46.2s
=> => extracting sha256:e0ac4f13b766d321acc3b650d3d23b82828995711f6247f 0.0s
=> => extracting sha256:df2c3b2eb7cc63351bb32f26457bbe0402af8082548f2697 8.8s
=> => extracting sha256:efe636eac583776a8a114d50fef15bc65b648f3d2bb53326 0.4s
=> => extracting sha256:fe17849545bb51455d3f7c8773ded2dbb1d6668a85bd0056 0.0s
=> [internal] load build context                                                 0.2s
=> => transferring context: 59.44kB                                            0.1s
=> [2/5] WORKDIR /app                                                       0.1s
=> [3/5] COPY package*.json .                                                 0.1s
=> ERROR [4/5] RUN npm ci                                                    3.3s
-----
```

```
-----> [4/5] RUN npm ci:
3.093 npm ERR! Cannot read property 'express' of undefined
3.133
3.136 npm ERR! A complete log of this run can be found in:
3.143 npm ERR!     /root/.npm/_logs/2024-09-09T09_03_26_444Z-debug.log
-----Dockerfile:8
-----6 |     COPY package*.json .
7 |
8 | >>> RUN npm ci
9 |
10 |    COPY . .
-----ERROR: failed to solve: process "/bin/sh -c npm ci" did not complete successfully: exit code: 1
```

```
[node1] (local) root@192.168.0.28 ~/container-technologies
$ nano Dockerfile
[node1] (local) root@192.168.0.28 ~/container-technologies
$ docker build .
[+] Building 15.6s (10/10) FINISHED                                            docker:default
  => [internal] load build definition from Dockerfile                      0.0s
  => => transferring dockerfile: 150B                                         0.0s
  => [internal] load .dockerignore                                         0.0s
  => => transferring context: 2B                                           0.0s
  => [internal] load metadata for docker.io/library/node:12                  0.1s
  => [1/5] FROM docker.io/library/node:12@sha256:01627afeb110b3054ba4a1405  0.0s
  => [internal] load build context                                         0.1s
  => => transferring context: 2.10kB                                       0.1s
  => CACHED [2/5] WORKDIR /app                                              0.0s
  => CACHED [3/5] COPY package*.json .                                      0.0s
  => [4/5] RUN npm install                                                 14.0s
  => [5/5] COPY . .                                                       0.1s
  => exporting to image                                                 1.1s
  => => exporting layers                                                 1.1s
  => => writing image sha256:4e80061f89fc8396d5056c83694697e6a85d9e31e2020 0.0s
```

Then, it comes to the learning about “Tagging and running own image”, I learned how to get a name for my created image and then, it was quite easy for me to run the image. Then, I learned how to delete the container and image, I had to keep in mind that I had to delete the container first and then, the image as below.

```
[node1] (local) root@192.168.0.28 ~/container-technologies
docker images
REPOSITORY      TAG      IMAGE ID      CREATED        SIZE
<none>          <none>    4e80061f89fc  38 minutes ago  923MB
ubuntu          latest    edbfe74c41f8  5 weeks ago   78.1MB
hello-world     latest    d2c94e258dcb  16 months ago  13.3kB
[node1] (local) root@192.168.0.28 ~/container-technologies
$ docker tag 4e80061f89fc app
[node1] (local) root@192.168.0.28 ~/container-technologies
$ docker images
REPOSITORY      TAG      IMAGE ID      CREATED        SIZE
app             latest   4e80061f89fc  39 minutes ago  923MB
ubuntu          latest   edbfe74c41f8  5 weeks ago   78.1MB
hello-world     latest   d2c94e258dcb  16 months ago  13.3kB
[node1] (local) root@192.168.0.28 ~/container-technologies
$ docker run app

> docker_practice@1.0.0 start /app
> node app.js

Server is running on http://localhost:3000
```

```

$ docker container ls --all
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
0afcd6f168bb app "docker-entrypoint.s..." 2 minutes ago Exited (0) About a minute ago
161cc29911c8 ubuntu "bash" 3 hours ago Exited (0) 3 hours ago
9f8680435d19 hello-world "/hello" 3 hours ago Exited (0) 3 hours ago
[node1] (local) root@192.168.0.28 ~/container-technologies
$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
app latest 4e80061f89fc 41 minutes ago 923MB
ubuntu latest edbfe74c41f8 5 weeks ago 78.1MB
hello-world latest d2c94e258dcb 16 months ago 13.3kB
[node1] (local) root@192.168.0.28 ~/container-technologies
$ docker image rm app
Error response from daemon: conflict: unable to remove repository reference "app" (must force) - container 0afcd6f168bb is using its referenced image 4e80061f89fc
[node1] (local) root@192.168.0.28 ~/container-technologies
$ docker container rm 0afcd6f168bb
0afcd6f168bb
[node1] (local) root@192.168.0.28 ~/container-technologies
$ docker container ls --all
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
161cc29911c8 ubuntu "bash" 3 hours ago Exited (0) 3 hours ago
9f8680435d19 hello-world "/hello" 3 hours ago Exited (0) 3 hours ago
[node1] (local) root@192.168.0.28 ~/container-technologies
$ docker image rm app
Untagged: app:latest
Deleted: sha256:4e80061f89fc8396d5056c83694697e6a85d9e31e2020bebff00d4d5edb118557
[node1] (local) root@192.168.0.28 ~/container-technologies
$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
ubuntu latest edbfe74c41f8 5 weeks ago 78.1MB
hello-world latest d2c94e258dcb 16 months ago 13.3kB

```

In the demonstration video about “Creating a container - useful options”, I learned how to run docker image in another way that I could access the endpoint in my local environment. At first, I built the image again after deleting it and ran the command **docker run -d -p 3003:3000 app** to run the app at a localhost port 3003. Then, when I ran the command **curl** with the localhost endpoint, I could access the Express application as below.

```

[root@node1 ~]
[node1] (local) root@192.168.0.28 ~/container-technologies
$ docker build .
[+] Building 0.5s (10/10) FINISHED                                            docker:default
=> [internal] load build definition from Dockerfile                         0.0s
=> => transferring dockerfile: 150B                                         0.0s
=> [internal] load .dockerignore                                         0.0s
=> => transferring context: 2B                                         0.0s
=> [internal] load metadata for docker.io/library/node:12                  0.3s
=> [1/5] FROM docker.io/library/node:12@sha256:01627afeb110b3054ba4a1405  0.0s
=> [internal] load build context                                         0.0s

[node1] (local) root@192.168.0.28 ~/container-technologies
$ docker run -d -p 3003:3000 app
20d455d448bcde3b624846683cccd348bf82f11cf98a28a308b081cdcb3c2d4c
[node1] (local) root@192.168.0.28 ~/container-technologies
$ curl http://localhost:3003/
Hello, Node.js and Express.js![node1] (local) root@192.168.0.28 ~/container-technologies
$ ^C

```

In the last video, I learned how to push image to Container Registry on Gitlab based on instructions of a teacher. I successfully did it from Docker Playground and image was registered on Gitlab as below.

The screenshot shows a container registry interface. At the top, there are buttons for 'CLI Commands' and a gear icon. Below that, it says '1 Image repository' and 'Cleanup is not scheduled. Set up cleanup'. A note indicates 'Container Scanning for Registry: Off'. There is a search bar and filters for 'Updated' and 'Published'. A single image entry is listed: 'docker-registry-learning' with one tag, published 2 minutes ago.

In conclusion, in the section Container Technologies, I learned a lot how to dockerize the application as the image, run the image to access the application, and push it to the Container Registry on Gitlab. The knowledge was good and supported a lot for my insights about software development.

### 3 CLOUD ENVIRONMENTS

**10.09.2024**

In this course, I learned about how to use Azure Cloud Environments. Actually, I had learned about AWS cloud in the course named AWS Academy Cloud Foundations and in this course, it was quite to revise cloud knowledge and get deeper into another Azure Cloud Environments. Thanks to the support of teacher, I could create an Azure account for student to start learning this section as below.

The screenshot shows the Microsoft Azure portal's 'Subscriptions' page. The URL is 'portal.azure.com'. The top navigation bar includes 'Log in to the site | LUT-korkeakoulut', 'Email - Thong P Dang - Outlook', and 'Subscriptions - Microsoft Azure'. The main menu has 'Microsoft Azure' and 'Copilot'. The search bar says 'Search resources, services, and docs (G+)'. The 'Subscriptions' section shows a single entry: 'Azure for Students'. The table details include:

Subscription name	Subscription ID	My role	Current cost	Secure S
Azure for Students	955aae4e-965a-4602-a07a-204cd9b50ac5	Owner	Not available	-

In the demonstration video about “Creating, managing and removing a web app service”, I learned how to deploy the Express application to Azure Cloud Environment. However, I constantly got a 401 error regarding authentication. I tried to look up on the Internet and finally, found a solution that I should set up **External Git** in **Settings** part on the Azure

Cloud platform that I should choose **Private for Repository Type** instead of **Public** type and filled my Gitlab username and password as Gitlab required strict requirements for fetching Gitlab repositories in any environments and platforms. I also sent an email to notice the teacher about this issue. Eventually, I could deploy as below images.

**lut-devops-application-learning | Deployment Center**

**Logs**

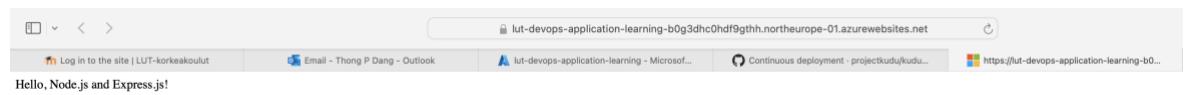
Deploy and build code from your preferred source and build provider. [Learn more](#)

Source	External Git
	<a href="#">Disconnect</a>
<b>External Git</b>	
Username	ThongDang27081998
Repository	<a href="https://gitlab.com/lut-study/docker-registry-learning.git">https://gitlab.com/lut-study/docker-registry-learning.git</a>
Branch	master
<b>Build</b>	
Build provider	App Service Build Service
Runtime stack	Node
Version	~18

**lut-devops-application-learning | Deployment Center**

**Logs**

Time	Commit ID	Commit Author	Status	Message
Tuesday, September 10, 2024 (1)	12d12f9	Thong Dang	Success (Active)	Update node version



## 4 CI/CD

10.09.2024

In this section about CI/CD, I started to learn new knowledge about continuous integration and continuous deployment. It is nice to learn about this.

In the first demonstration video about “Running an example pipeline”, I learned how to create a pipeline to run in test stage and it gave me the first overview about how to configure the pipeline in a yaml file on Gitlab as below images. There were some errors but finally, it still worked well.

All 4	Finished	Branches	Tags	Clear runner caches	Run pipeline
Filter pipelines				<input type="text"/>	Show Pipeline ID
Status	Pipeline	Created by	Stages		
<span>Passed</span>	Fixed pipeline #1451279052 ↗ master ➔ d0e11bb4 (latest)				
<span>Failed</span>	Test pipeline #1451278318 ↗ master ➔ ba982e93 (yaml invalid error)				
<span>Passed</span>	Added a yaml file #1451276369 ↗ master ➔ 0b44426e				
<span>Failed</span>	Added a yaml file #1451269287 ↗ master ➔ 0b44426e (error)				

In the second demonstration video, I learned how to refactor the Express application to run tests with Jest and Supertest successfully. This thing did not relate to contents of the course but I learned how to refactor the structures of the Express application and it was good to understand. I made changes in **tests** branch and then merged into the master branch as below images. In addition, I also delete the tests branch in my local environment after merging successfully.

In the third demonstration video, I learned how to set up the first stage of the pipeline and it was **build\_and\_test**. However, I got so many failing incidents in the pipeline with **npm run test** even though in my local environment, there were no problems. Finally, I found that there

was an irrelevant import in a test file and it caused constant fails in the pipeline. I removed that import and the pipeline worked well.

Status	Pipeline	Created by	Stages	
<span>Passed</span> ⌚ 00:00:36 ⌚ Just now	Added echo to pipeline for test #1451409767 ⌚ master → 8cb6f463 [latest]	●	●	⤵ ⏺
<span>Passed</span> ⌚ 00:00:43 ⌚ 2 minutes ago	Remove redundant import #1451407026 ⌚ master → 35117c28 [latest]	●	●	⤵ ⏺
<span>Failed</span> ⌚ 00:00:36 ⌚ 13 minutes ago	Remove echo #1451397201 ⌚ master → e8307ced [latest]	●	○	⤵ ⏺
<span>Failed</span> ⌚ 00:00:37 ⌚ 14 minutes ago	Remove echo #1451396078 ⌚ master → e8307ced [latest]	●	○	⤵ ⏺
<span>Failed</span> ⌚ 00:00:36 ⌚ 15 minutes ago	Added test to pipeline #1451395059 ⌚ master → 31376d8b [latest]	●	○	⤵ ⏺
<span>Failed</span> ⌚ 00:00:36 ⌚ 17 minutes ago	Added test to pipeline #1451393311 ⌚ master → 31376d8b [latest]	●	○	⤵ ⏺

In this demonstration video about how to containerization stage in the pipeline, I learned how to add into the pipeline to containerize the application after building and testing successfully. There were some mistakes in writing the script and after fixing, the pipeline worked well as below images. I also learned several authentication terms to log into the docker from the CI pipeline safely without giving correct user credentials such as **\$CI\_REGISTRY\_USER**, **\$CI\_REGISTRY\_PASSWORD**, **\$CI\_REGISTRY**.

Status	Pipeline	Created by	Stages	
<span>Passed</span> ⌚ 00:01:53 ⌚ Just now	Fixed spelling #1451469725 ⌚ master → 695dca9b [latest]	●	● ●	⤵ ⏺
<span>Failed</span> ⌚ 00:00:44 ⌚ 4 minutes ago	Added echos #1451466870 ⌚ master → 1da8738c [latest]	●	● ○	⤵ ⏺
<span>Passed</span> ⌚ 00:02:26 ⌚ 11 minutes ago	Fixed pipeline #1451458489 ⌚ master → 21b0542d [latest]	●	● ●	⤵ ⏺
<span>Failed</span> ⌚ 00:01:15 ⌚ 18 minutes ago	Added a second stage containerization #1451452619 ⌚ master → 965231ae [latest]	●	● ○	⤵ ⏺

In the following demonstration videos, I learned how to configure the yaml file with the final deployment stage. At first, I had to set up on Microsoft Azure to pull the container image from Gitlab registry. Then, I was given with WebHook url to set up WebHook on Gitlab, and I also needed to set up production environment on Gitlab. Based on learnings, in WebHook configuration on Gitlab, I needed to set up to trigger WebHook in **Deployment Events** after running **build\_and\_test** and **containerization** stages successfully in the pipeline before the **deployment** stage instead of **Push Events**, which triggered the WebHook immediately after there were committed changes in the master branch. Actually, I got two

specific problems regarding deployment to Azure that I needed to allow **Continuous Deployment** in my Web App Service on Azure and set up to avoid **REDACTED:REDACTED** in the beginning of the WebHook url by enabling **SCM basic auth publishing credentials** on Azure. After fixes, I could deploy my application well to Azure and whenever I updated new changes, a process ran automatically and the application was automatically deployed with new changes.

The screenshot shows the Azure Deployment Center interface for a Web App named "cicdlutlearningcourse". The top navigation bar includes a blue icon, the app name, a star icon, and a three-dot menu. Below the navigation, there are buttons for "Active 1" (highlighted in blue), "Stopped 0", "Clean up environments", "Enable review apps", and "New environment". A search bar labeled "Search by environment name" is present. Under the search bar, a link to "production environment" is shown with "Open" and "Stop" buttons. A deployment event titled "Test webhook again" is listed, showing a green "Passed" status, duration "00:02:19", and timestamp "10 minutes ago". The pipeline details include a pull request "#1453483246" from "master" to "fa4fea0d" (latest). The stages section shows three green checkmarks. At the bottom, a URL "https://cicdlutlearningcourse-dbawfue6atfyh6bj.eastus-01.azurewebsites.net" is displayed with a back arrow, a refresh icon, and a lock icon.

Active 1   Stopped 0   Clean up environments   Enable review apps   New environment

Search by environment name

> production environment   Open   Stop

Deployment events   SSL Verification: enabled

Status	Pipeline	Created by	Stages
Passed	Test webhook again #1453483246 master fa4fea0d latest		

←   ↕   https://cicdlutlearningcourse-dbawfue6atfyh6bj.eastus-01.azurewebsites.net

Check again to test Web Hook works well!