








How to work around Docker


Using GitHub Codespaces

1) Fork the repository

 RumbleDB / bigdata-exercises




[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#)






 **bigdata-exercises** Public



[Edit Pins](#) [Watch](#) **19** [Fork](#) **298** [Star](#) **129**

Fork your own copy of RumbleDB/bigdata-exercises

[master](#) [2 Branches](#) [Tags](#) [Add file](#) [Code](#)

 Merge pull request [#15](#) from mertalbaba/master 0135502 · 1 hour ago 332 Commits

 Big_Data	Update exercise 6 solution	17 hours ago
 Big_Data_For_Engineers	Restored picture	3 months ago
 .gitattributes	Add .gitattributes to fix line termination of *.sh files.	3 years ago
 .gitignore	[README] Add first version of instructions. Add helper sc...	3 years ago
 README.md	Update for Big Data Fall 2024	last month







 **README** 

Exercises

This repository contains exercises for the Big Data (Fall 2024) and the Big Data for Engineers lecture at ETH Zurich (Spring 2024)

About

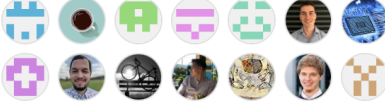
Exercises for the Big Data lecture at ETH Zurich (Fall 2021)

-  Readme
-  Activity
-  Custom properties
-  129 stars
-  19 watching
-  298 forks

[Report repository](#)

Contributors

41



[+ 27 contributors](#)

Languages

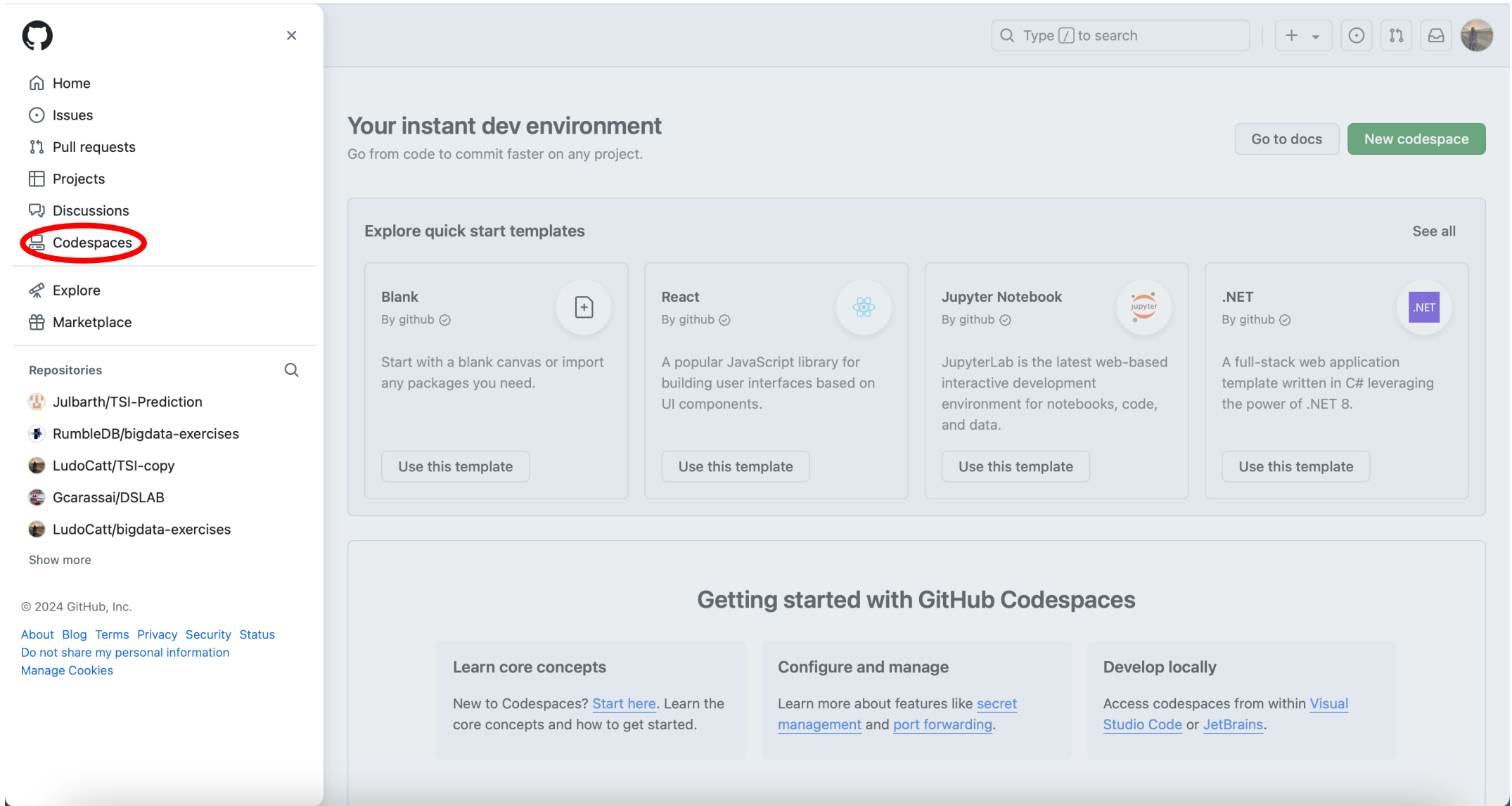
Jupyter Notebook

98.2%

Other

1.8%

2) Go to GitHub Codespaces



The screenshot displays the GitHub Codespaces web interface. On the left, a sidebar contains navigation links: Home, Issues, Pull requests, Projects, Discussions, and Codespaces (which is circled in red). Below these are links for Explore and Marketplace, followed by a list of repositories including Julbarth/TSI-Prediction, RumbleDB/bigdata-exercises, LudoCatt/TSI-copy, Gcarassai/DSLALB, and LudoCatt/bigdata-exercises. The main content area features a header with a search bar and navigation icons. Below this, a section titled 'Your instant dev environment' includes a 'Go to docs' button and a prominent green 'New codespace' button. The 'Explore quick start templates' section offers four options: Blank, React, Jupyter Notebook, and .NET, each with a 'Use this template' button. At the bottom, a 'Getting started with GitHub Codespaces' section provides links for learning core concepts, configuring and managing the environment, and developing locally.

GitHub Codespaces Interface

Left Sidebar:

- Home
- Issues
- Pull requests
- Projects
- Discussions
- Codespaces**
- Explore
- Marketplace
- Repositories
 - Julbarth/TSI-Prediction
 - RumbleDB/bigdata-exercises
 - LudoCatt/TSI-copy
 - Gcarassai/DSLALB
 - LudoCatt/bigdata-exercises

Main Content Area:

Your instant dev environment
Go from code to commit faster on any project.

[Go to docs](#) [New codespace](#)

Explore quick start templates

- Blank**
By github
Start with a blank canvas or import any packages you need.
[Use this template](#)
- React**
By github
A popular JavaScript library for building user interfaces based on UI components.
[Use this template](#)
- Jupyter Notebook**
By github
JupyterLab is the latest web-based interactive development environment for notebooks, code, and data.
[Use this template](#)
- .NET**
By github
A full-stack web application template written in C# leveraging the power of .NET 8.
[Use this template](#)

[See all](#)

Getting started with GitHub Codespaces

- Learn core concepts**
New to Codespaces? [Start here](#). Learn the core concepts and how to get started.
- Configure and manage**
Learn more about features like [secret management](#) and [port forwarding](#).
- Develop locally**
Access codespaces from within [Visual Studio Code](#) or [JetBrains](#).

3) Create new Codespace with forked repo

Codespaces

Create new codespace

Search Type to search

Create a new codespace

Repository
To be cloned into your codespace

Branch
This branch will be checked out on creation

Region
Your codespace will run in the selected region

Machine type
Resources for your codespace

Select a repository

Select repository

big

LudoCatt/bigdata-exercises

tmcw/big

bigcompany/big

heibaiying/BigData-Notes

wangzhiwubigdata/God-Of-BigData

sdcoffey/big

bigbluebutton/bigbluebutton

oxnr/awesome-bigdata

MoRan1607/BigDataGuide

iGaoWei/BigDataView

intel-analytics/BigDL-2.x

© 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do

- I recommend choosing **4-core cpu**, as it is slow otherwise
- You only have limited time in free plan, but should be enough: ~120 Core hours per month

4) Docker-compose up in exercise05

The screenshot shows a Visual Studio Code window with the following components:

- EXPLORER:** A file tree on the left showing the project structure. The 'Big_Data' folder is expanded, showing subfolders 'exercise00' through 'exercise10'. The 'exercise05' folder is selected, showing files like 'docker-compose-aarch64.yml', 'docker-compose.yml', 'enwiki-20200920-pages-articles-multi...', 'Exercise05_HBase_Solution.ipynb', and 'Exercise05_HBase.ipynb'.
- EDITOR:** The main workspace shows the 'Exercise05_HBase.ipynb' file. It contains a title 'Big Data – Exercises', a subtitle 'Autumn 2024 – Week 5 – ETH Zurich', and a section 'Wide Column Stores - HBase'. Below this, it states 'This exercise will consist of three main parts. You will:' followed by a list of tasks: 'Create your own HBase cluster, fill it with data, and run basic queries.', 'Get hands-on practice with HBase using the Wikipedia dataset.', and 'Get familiar with HFile indexes.'
- TERMINAL:** A terminal window at the bottom shows the command prompt for a user named 'LudoCatt'. The command executed is `cd Big_Data/exercise05` followed by `docker-compose up`. The output shows the process of pulling Docker images for 'hbase-master', 'zookeeper', and 'hbase-regionserver'.

Important note: If you are using a ARM based Mac, do the following steps before running docker compose up -d:

- delete the original `docker-compose.yml` file
- rename `docker-compose-aarch64.yml` to `docker-compose.yml`

The terminal output shows the following commands and results:

```
@LudoCatt → /workspaces/bigdata-exercises (master) $ cd Big_Data/exercise05
@LudoCatt → /workspaces/bigdata-exercises/Big_Data/exercise05 (master) $ docker-compose up
WARN[0000] /workspaces/bigdata-exercises/Big_Data/exercise05/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 12/3
  ⚙️ hbase-master [#####] 298.4MB / 318.3MB Pulling 7.3s
  ⚙️ zookeeper [#####] 96.94MB / 96.94MB Pulling 7.3s
  ⚙️ hbase-regionserver Pulling 7.3s
```

5) Open a new terminal and continue

The screenshot shows the Visual Studio Code interface with a Jupyter Notebook open. The Explorer panel on the left shows the file structure of the 'BIGDATA-EXERCISES' workspace, with 'exercise05' selected. The Jupyter Notebook editor displays instructions for setting up a Docker cluster. The terminal window at the bottom shows the execution of commands to start the cluster and list the containers. A red circle highlights the terminal window's title bar, which includes a '+' button for opening a new terminal.

Codespaces

Exercise05_HBase.ipynb — bigdata-exercises [Codespaces: miserable tomb] — Visual Studio Code

bigdata-exercises [Codespaces: miserable tomb]

EXPLORER

BIGDATA-EXERCISES [CODESPACES: MISERABLE TOMB]

Big_Data

- exercise00
- exercise01
- exercise02
- exercise03
- exercise04
- exercise05
 - .cache
 - .local
 - jupyter
 - docker-compose-aarch64.yml
 - docker-compose.yml
 - enwiki-20200920-pages-articles-multi...
 - Exercise05_HBase_Solution.ipynb
 - Exercise05_HBase.ipynb
- exercise06
- exercise07
- exercise07-aarch64
- exercise08
- exercise09
- exercise10
- exercise11
- exercise12
- exercise13

activate-docker-env.sh

deactivate-docker-env.sh

init-docker-env.sh

README.md

OUTLINE

TIMELINE

[Preview] README.md

Exercise05_HBase.ipynb

Big_Data > exercise05 > Exercise05_HBase.ipynb > ...

+ Code + Markdown ...

Select Kernel

green on the UI of Docker desktop.

1. In the command line, navigate into your exercise05 folder using 'cd' command and instantiate the cluster by running:

docker compose up -d

Note that for the first time, it might take a little bit longer to set things up, as docker might need to pull the new images.

```
PS C:\_bigdata\bigdata\big_data_for_engineers\Exercises\exercise05> docker compose up -d
[+] Running 5/5
 ✓ Network exercise05_default          Created      0.1s
 ✓ Container ex05-jupyter              Started      0.3s
 ✓ Container exercise05-hbase-regionserver-1 Started      0.2s
 ✓ Container exercise05-zookeeper-1    Started      0.2s
 ✓ Container exercise05-hbase-master-1 Started      0.2s
PS C:\_bigdata\bigdata\big_data_for_engineers\Exercises\exercise05>
```

2. List the container names with

docker ps --format "{{.Names}}"

```
PS C:\_bigdata\bigdata\big_data_for_engineers\Exercises\exercise05> docker ps --format "{{.Names}}"
ex05-jupyter
exercise05-hbase-master-1
exercise05-hbase-regionserver-1
exercise05-zookeeper-1
PS C:\_bigdata\bigdata\big_data_for_engineers\Exercises\exercise05>
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 7 COMMENTS

```
@LudoCatt → /workspaces/bigdata-exercises (master) $ cd Big_Data/exercise05
@LudoCatt → /workspaces/bigdata-exercises/Big_Data/exercise05 (master) $ docker ps --format "{{.Names}}"
exercise05-hbase-regionserver-1
exercise05-zookeeper-1
ex05-jupyter
exercise05-hbase-master-1
@LudoCatt → /workspaces/bigdata-exercises/Big_Data/exercise05 (master) $
```

Cell 6 of 19 Layout: U.S.

5.2) Open a new terminal and continue

The screenshot displays the Visual Studio Code interface with a Jupyter Notebook open. The Explorer sidebar on the left shows a project structure for 'BIGDATA-EXERCISES [CODESPACES: MISERABLE TOMB]'. The main editor area shows the 'Exercise05_HBase.ipynb' notebook. The notebook content includes a code cell with a directory listing and a text cell with instructions for using the HBase shell.

EXPLORER

- BIGDATA-EXERCISES [CODESPACES: MISERABLE TOMB]
- Big_Data
 - exercise00
 - exercise01
 - exercise02
 - exercise03
 - exercise04
 - exercise05
 - .cache
 - .local
 - jupyter
 - docker-compose-aarch64.yml
 - docker-compose.yml
 - enwiki-20200920-pages-articles-multi...
 - Exercise05_HBase_Solution.ipynb
 - Exercise05_HBase.ipynb
 - exercise06
 - exercise07
 - exercise07-aarch64
 - exercise08
 - exercise09
 - exercise10
 - exercise11
 - exercise12
 - exercise13
- activate-docker-env.sh
- deactivate-docker-env.sh
- init-docker-env.sh
- README.md

EXERCISE05_HBase.ipynb

Big_Data > exercise05 > Exercise05_HBase.ipynb > ...

Code | Markdown

Select Kernel

home tmp
lib usr
lib64 var
media
bash-5.0#

5. We will use the .csv file later to populate our database. For now, let's explore the basics of HBase in this playground. To start, run the following command in the container's bash:

```
hbase shell
bash-5.0# hbase shell
2024-03-19 05:59:36,061 WARN [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using built-in java classes where applicable
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
For Reference, please visit: http://hbase.apache.org/2.0/book.html#shell
Version 2.1.3, rda5ec9e4c06c537213883cca8f3cc9a7c19daf67, Mon Feb 11 15:45:33 CST 2019
Took 0.0026 seconds
hbase(main):001:0>
```

Task 12: Interact with your HBase cluster using the shell

PROBLEMS | OUTPUT | DEBUG CONSOLE | **TERMINAL** | PORTS | COMMENTS

• @LudoCatt → /workspaces/bigdata-exercises/Big_Data/exercise05 (master) \$ docker cp enwiki-20200920-pages-articles-multistream_small.csv exercise05-hbase-master-1:/

Successfully copied 87.6MB to exercise05-hbase-master-1:/

○ @LudoCatt → /workspaces/bigdata-exercises/Big_Data/exercise05 (master) \$ docker exec -it exercise05-hbase-master-1 /bin/bash

```
bash-5.0# hbase shell
2024-10-28 14:07:18,088 WARN [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
For Reference, please visit: http://hbase.apache.org/2.0/book.html#shell
Version 2.1.3, rda5ec9e4c06c537213883cca8f3cc9a7c19daf67, Mon Feb 11 15:45:33 CST 2019
Took 0.0031 seconds
hbase(main):001:0>
```

Codespaces: miserable tomb | master* | 0 | 7

Spaces: 4 | Cell 7 of 19 | Layout: U.S.

6) Copy the dataset and get started

Codespaces

Exercise05_HBase.ipynb — bigdata-exercises [Codespaces: miserable tomb] — Visual Studio Code

bigdata-exercises [Codespaces: miserable tomb]

EXPLORER

BIGDATA-EXERCISES [CODESPACES: MISERABLE TOMB]

Big_Data

- exercise00
- exercise01
- exercise02
- exercise03
- exercise04
- exercise05
 - .cache
 - .local
 - jupyter
 - docker-compose-aarch64.yml
 - docker-compose.yml
 - enwiki-20200920-pages-articles-multi...
 - Exercise05_HBase_Solution.ipynb
 - Exercise05_HBase.ipynb
- exercise06
- exercise07
- exercise07-aarch64
- exercise08
- exercise09
- exercise10
- exercise11
- exercise12
- exercise13

activate-docker-env.sh

deactivate-docker-env.sh

init-docker-env.sh

README.md

OUTLINE

TIMELINE

[Preview] README.md

Exercise05_HBase.ipynb

Big_Data > exercise05 > Exercise05_HBase.ipynb > ...

+ Code + Markdown ...

Select Kernel

Now we need to populate both tables with data. We will use the ImportTsv utility of HBase. Populate the table `wiki_small` by running the following (keep in mind that you should run this command in the container's bash).

```
hbase org.apache.hadoop.hbase.mapreduce.ImportTsv -Dimporttsv.separator=, -Dimporttsv.columns='HBASE_ROW_KEY,page:page_title,page:page_ns,page:revision_id,author:timestamp,author:contributor_id,author:contributor_name' wiki_small enwiki-20200920-pages-articles-multistream_small.csv
```

We need to specify which column in the csv maps to which column in the HBase table. Note that we make `page_id` into the `HBASE_ROW_KEY` and how we specify the mappings between the `.csv` columns and the `family:column` in the HBase table.

These commands print a lot of messages, but they are mostly informational with occasional non-critical warnings; unless something goes wrong, of course :). The commands will also report some "Bad Lines", but you can safely ignore this -- some lines may contain illegal characters and be dropped, but most of the data is in good shape.

You can count how many rows there are using this command from your head node's shell:

```
hbase org.apache.hadoop.hbase.mapreduce.RowCounter 'wiki_small'
```

If everything goes right, you should see `ROWS=887784` in the output.

Now let's go into HBase shell again (by running `hbase shell`) and run some queries against the `wiki_small` table. We will look at some of the filters listed by HBase if you run `show_filters` in an HBase shell, e.g., `PrefixFilter()`, `ValueFilter()`, `SingleColumnValueFilter()`.

Task 2.1: Indexing

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 7 COMMENTS

```
1000158 column=page:bytes, timestamp=1730124523047, value=24
1000158 column=page:page_ns, timestamp=1730124523047, value=0
1000158 column=page:page_title, timestamp=1730124523047, value=|The Space Race|
1000158 column=page:revision_id, timestamp=1730124523047, value=19128253
1000159 column=author:contributor_id, timestamp=1730124523047, value=55783
1000159 column=author:contributor_name, timestamp=1730124523047, value=|CryptoDerk|
1000159 column=author:timestamp, timestamp=1730124523047, value=2004-09-20T00:04:00Z
1000159 column=page:bytes, timestamp=1730124523047, value=40
1000159 column=page:page_ns, timestamp=1730124523047, value=0
1000159 column=page:page_title, timestamp=1730124523047, value=|Waterloo College|
1000159 column=page:revision_id, timestamp=1730124523047, value=16782323
7 row(s)
Took 0.6184 seconds
hbase(main):002:0>
```

docker exe...

docker exe...

Codespaces: miserable tomb master* 0 0 7

Spaces: 4 Cell 10 of 19 Layout: U.S.

Additional notes

- Even if you have an ARM architecture this all will happen within the Codespace so you do not have to touch the docker-compose files
- Every time you create a new Codespace you will have to start from scratch
- Be mindful of the instructions in the exercise: when you have to be in the shell and when you instead cannot