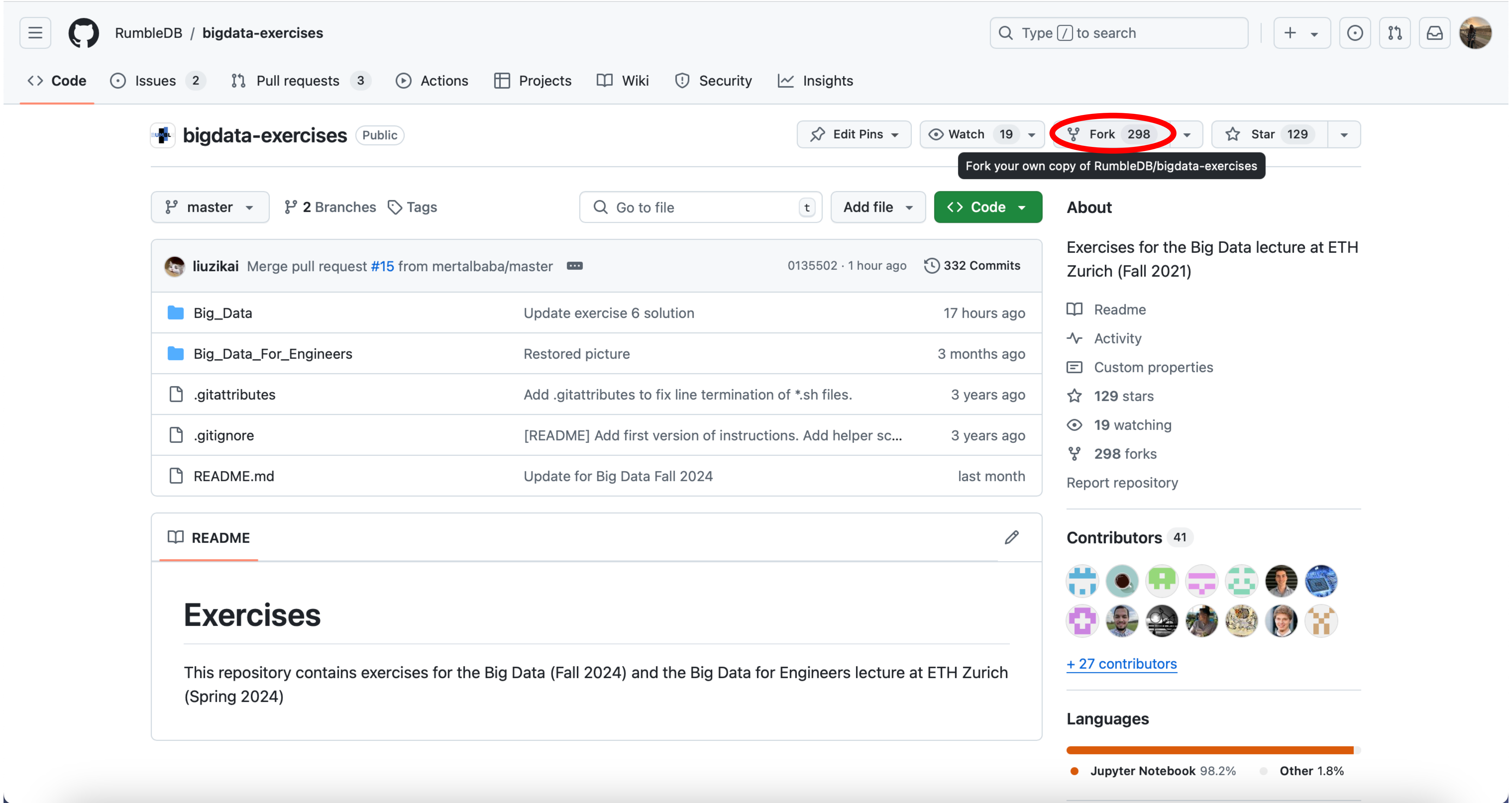
How to work around Docker

Using GitHub Codespaces

1) Fork the repository



Fork your own copy of RumbleDB/bigdata-exercises

liuzikai 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

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

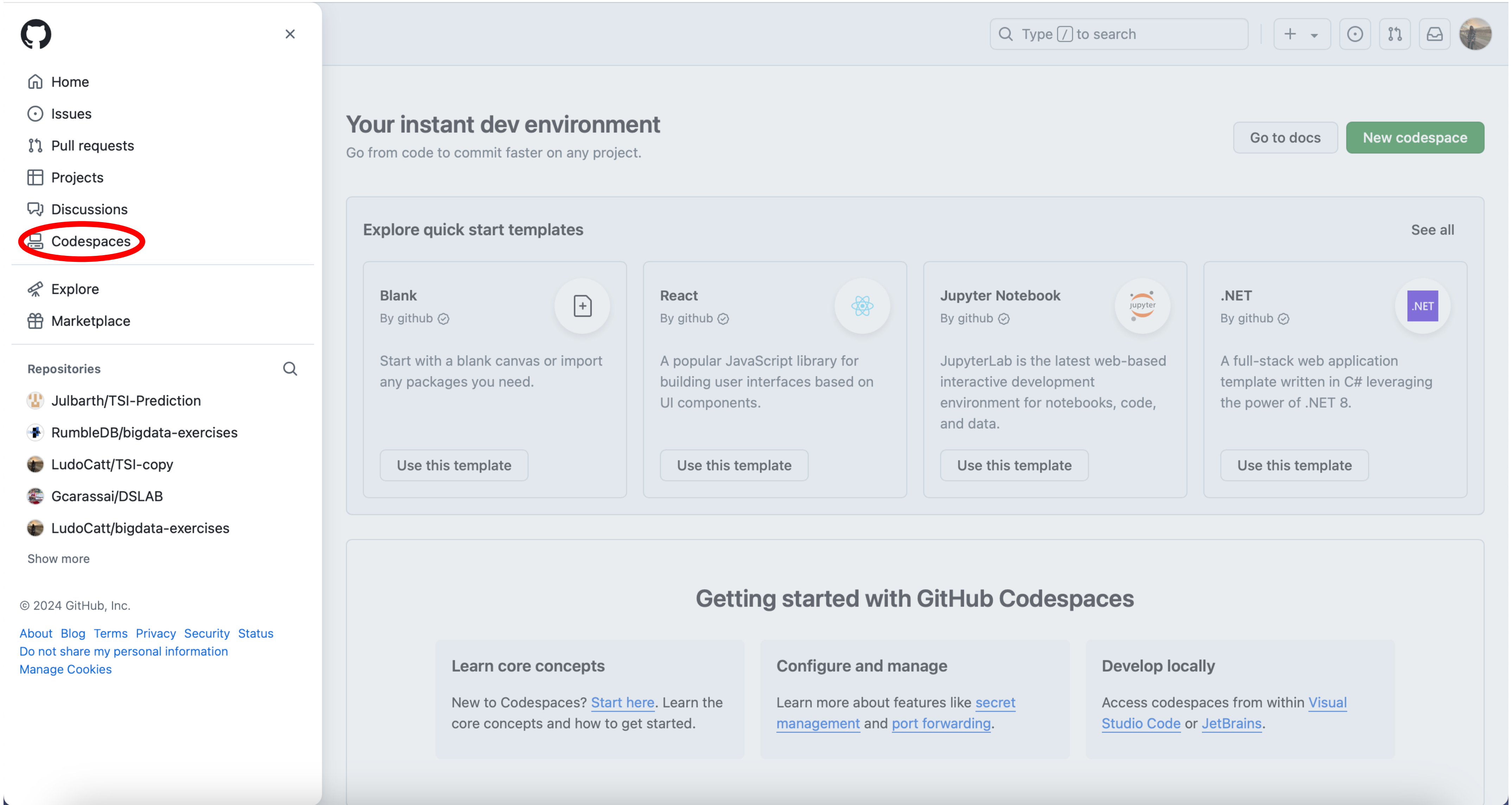
Contributors 41

+ 27 contributors

Languages

● Jupyter Notebook 98.2% ● Other 1.8%

2) Go to GitHub Codespaces



© 2024 GitHub, Inc.

About Blog Terms Privacy Security Status
Do not share my personal information
Manage Cookies

Your instant dev environment

Go from code to commit faster on any project.

Explore quick start templates

Blank
By github

Start with a blank canvas or import any packages you need.

React
By github

A popular JavaScript library for building user interfaces based on UI components.

Jupyter Notebook
By github

JupyterLab is the latest web-based interactive development environment for notebooks, code, and data.

.NET
By github

A full-stack web application template written in C# leveraging the power of .NET 8.

Getting started with GitHub Codespaces

Learn core concepts

New to Codespaces? Start here. Learn the core concepts and how to get started.

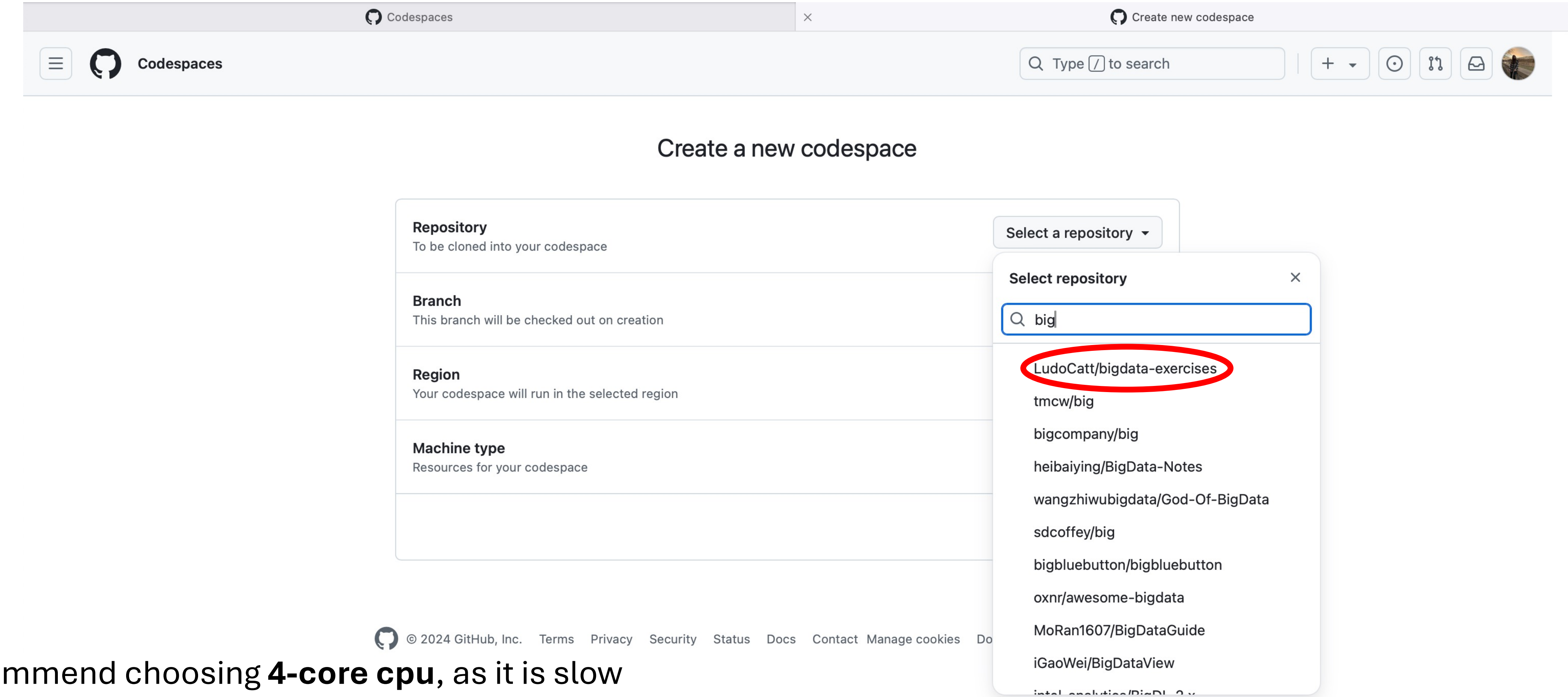
Configure and manage

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



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 repository

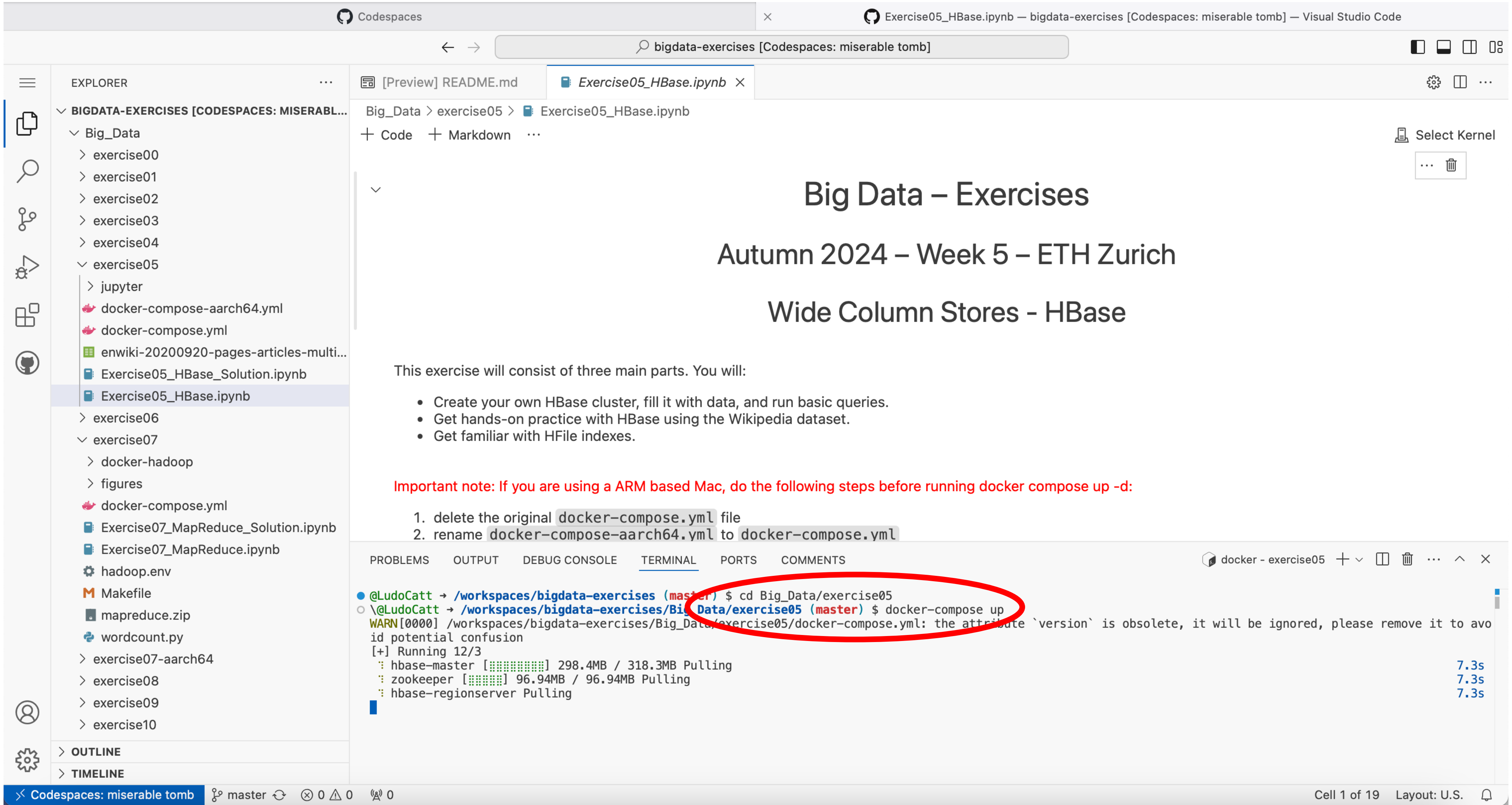
big

- 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



Big Data – Exercises

Autumn 2024 – Week 5 – ETH Zurich

Wide Column Stores - HBase

This exercise will consist of three main parts. You will:

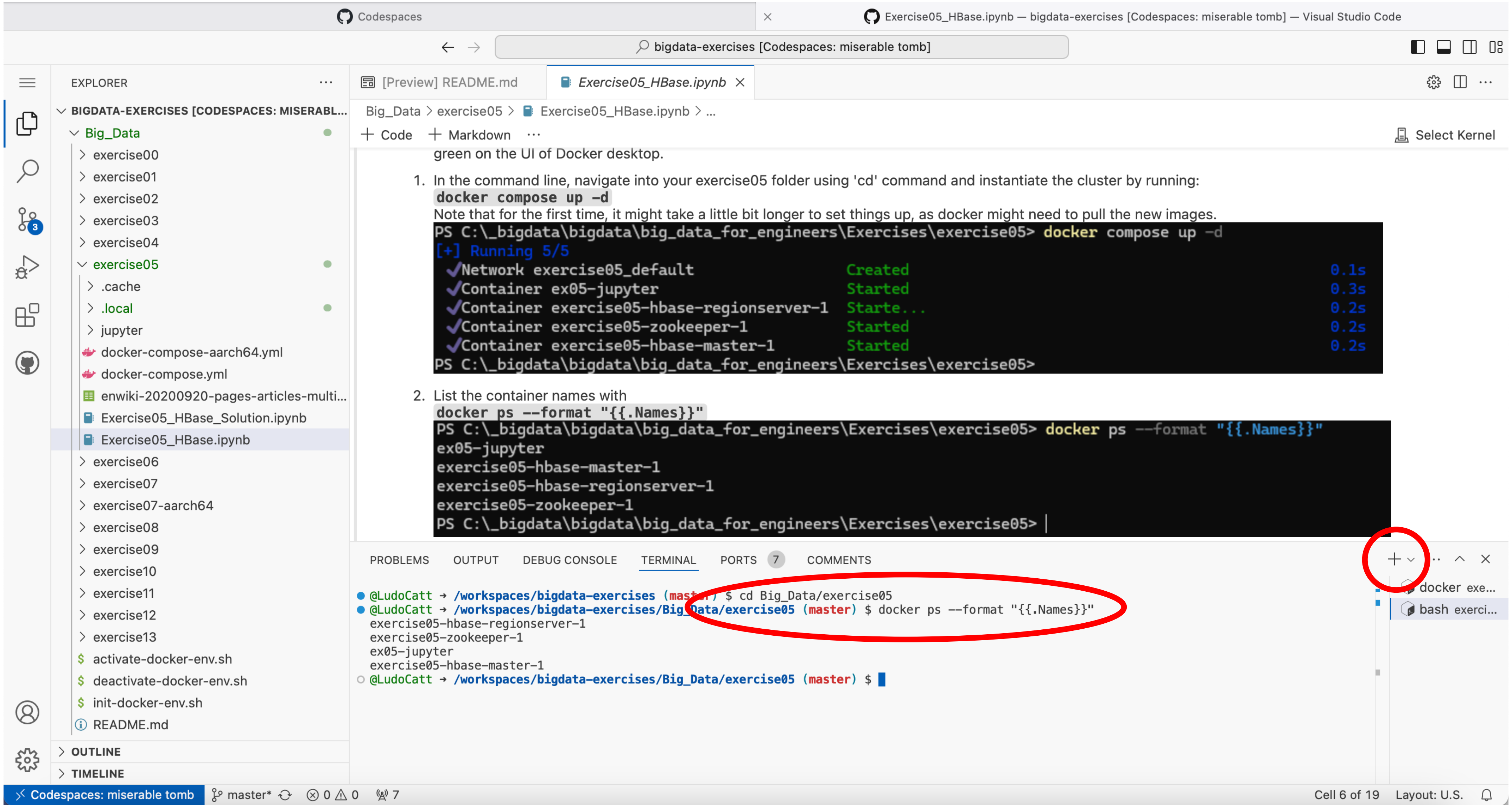
- Create your own HBase cluster, fill it with data, and run basic queries.
- Get hands-on practice with HBase using the Wikipedia dataset.
- Get familiar with HFile indexes.

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

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

```
● @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 avo
id 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



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.

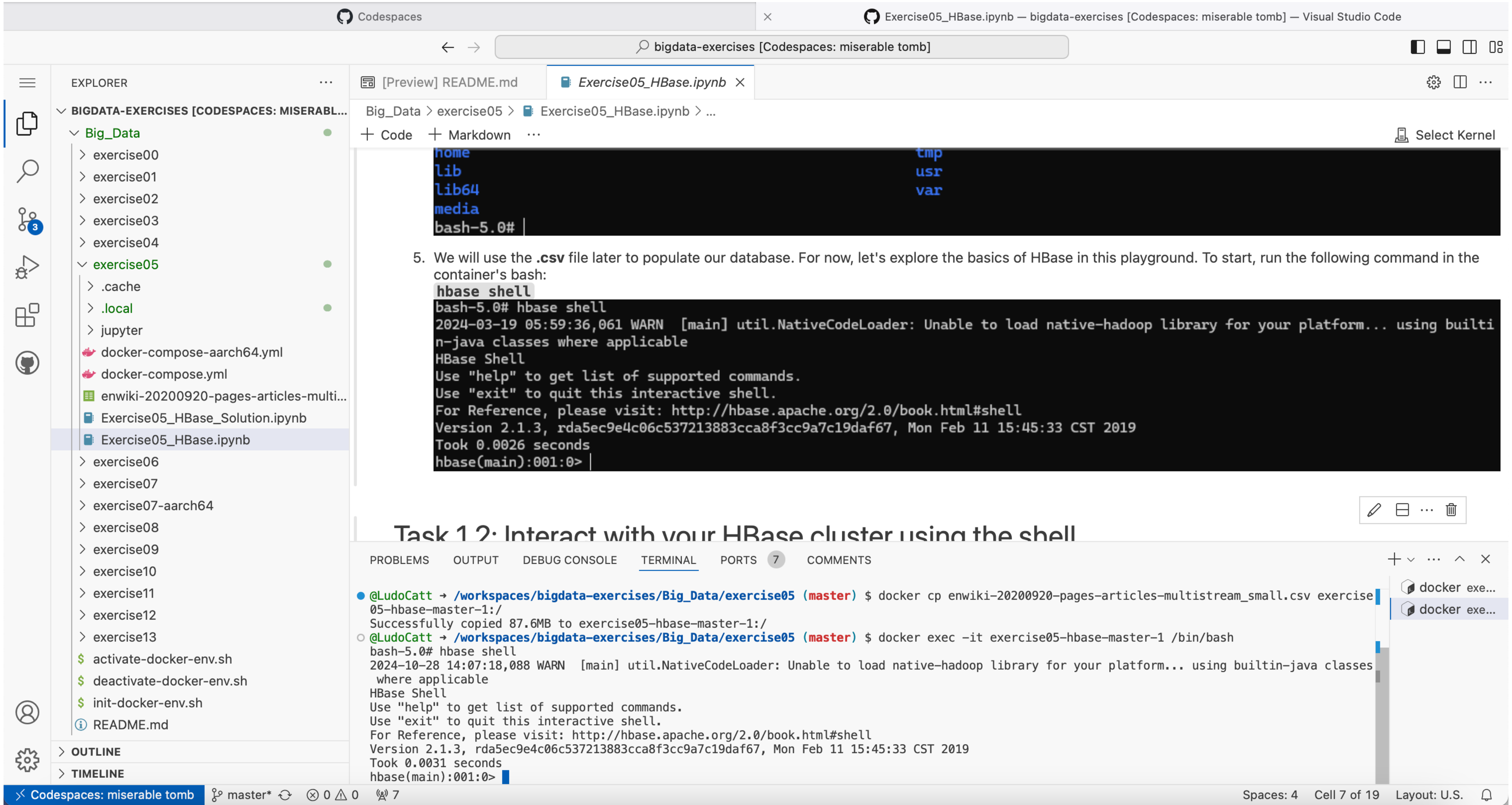
```
PS C:\_bigdata\bigdata\big_data_for_engineers\Exercises\exercise05> docker compose up -d
[+] Running 5/5
 ✓Network exercise05_default                  Created                                   0.1s
 ✓Container ex05-jupyter                      Started                                   0.3s
 ✓Container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

```
● @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) $ █
```

5.2) Open a new terminal and continue



```
home                                                        tmp
lib                                                         usr
lib64                                                       var
media
bash-5.0#
```

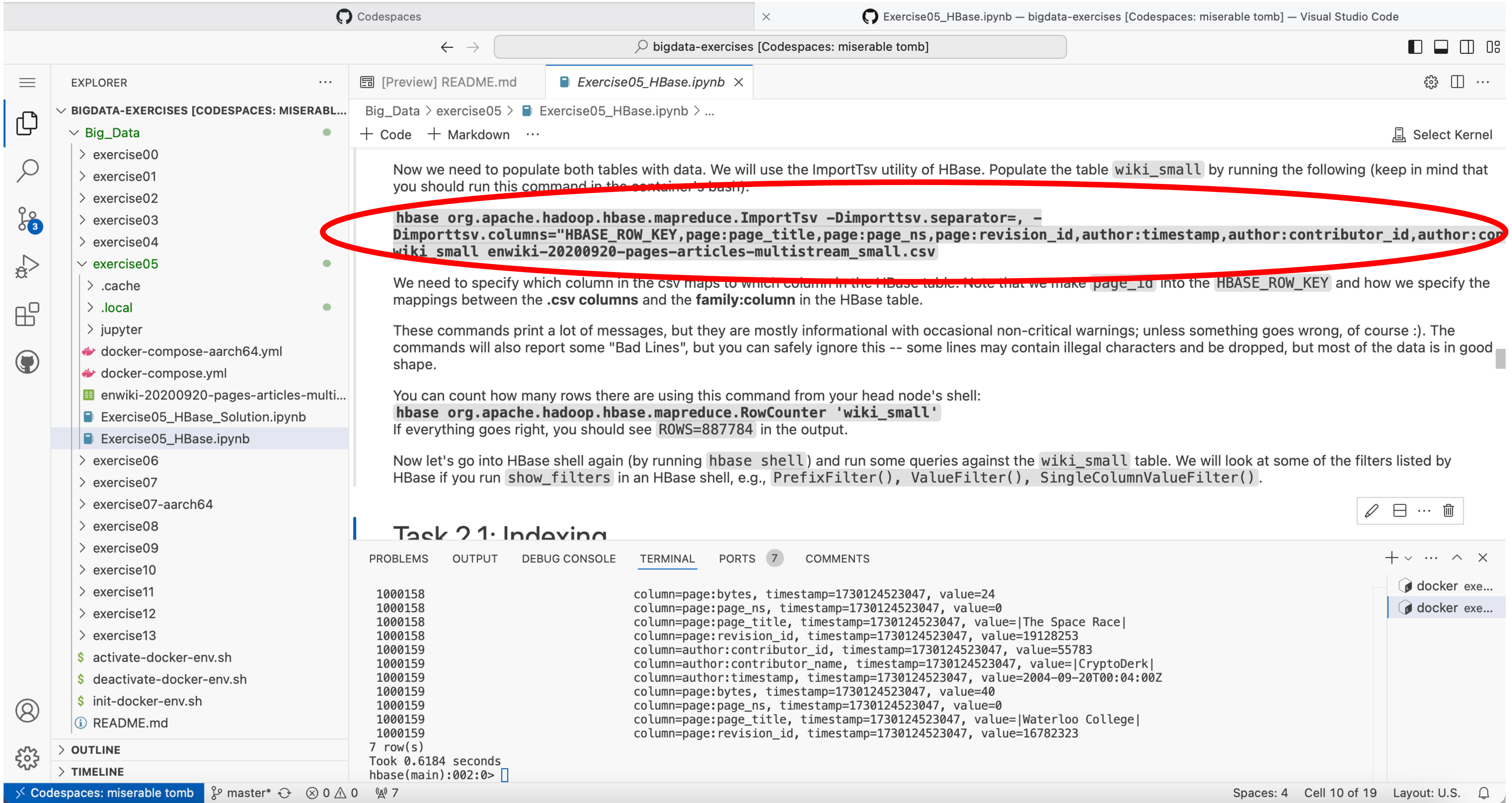
5. We will use the **.csv** file later to populate our database. For now, let's explore the basics of HBase in this playground. To start, run the following command in the container's bash:
   `hbase shell`

```
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 builti
n-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 1.2: Interact with your HBase cluster using the shell

```
@LudoCatt → /workspaces/bigdata-exercises/Big_Data/exercise05 (master) $ docker cp enwiki-20200920-pages-articles-multistream_small.csv exercise
05-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>
```

6) Copy the dataset and get started



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:con
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

```
 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>
```

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