# amcat-book

2025-09-09

# Table of contents

# Welcome

Welcome to the amcat manual.

## Acknowledgments

This version of the book was built with the following packages and versions:

| package | version |
|---|---|
| quarto | 1.2.313 |
| pandoc | 2.19.2 |
| R | 4.2.2 |
| Python | 3.10 |
| amcat4 | |
| amcat4r | 0.0.1.9000 |
| amcat4apiclient | 0.9 |
| operating system | Linux 6.1.3-arch1-1 |

# 1 Why amcat

An example workflow

Say, there has been a media frenzy recently about a political scandal. You know that it started with a revelation by an anonymous poster on Reddit, spread through social media, and was investigated by respectable media outlets, which created public pressure until the topic was discussed in your country's parliament and a minister or two had to step down. You quickly realise that the case is a treasure trove for investigating your political system and testing several media theories in the process.

You spin up a new instance of the amcat suite on your research server or a new cloud instance and start to collect data through APIs of social media websites, media monitoring sites and scrapers and store everything in the amcat database. You share your data with collaborators, using fine-grained data access control since some of the scraped content is copyrighted. A new research assistant with some knowledge about the case but no technical training starts digging through the data using amcat's user interface and publicly available dashboard to search for potentially relevant terms and queries. One of your collaborators builds on this by producing time-series models and plots in `R`.

You decide to dig deeper into the content and start training coders and deploy coding tasks through amcat's annotation tool. Coders get through the task quickly as they can access the interface on their phones and code on their bus commutes or whenever else they feel like it. You use your amcat server to preprocess the data and train and validate a heap of advanced machine learning models that complete the coding task of all documents. Your analysis reveals new mechanisms and confirms some of the theories you worked with.

After writing up a paper, you submit your results to a journal. The editor asks for replication code and data, so you simply share your R or Python scripts and grant temporary access to your amcat server for a researcher tasked with replicating your results.

After publication, a newspaper picks up your results, leading some interested citizens to play with your dashboard. Even though users do not have access to the full text for copyright reasons, they can query different combinations of keywords, which makes your research transparent for a wider audience. Since the annotations and the preprocessed texts are also available, someone finds they get even better validation scores using a newly created algorithm.

## 1.1 What is amcat?

The `amcat`-suite consists of several packages for text analysis. It has two main goals: to **standardize** text analysis tasks with **easy to use** software, while offering quality-of-life features for power users. It consists of several different software packages, which are usually used together:

- `amcat4` takes care of **document storage**, provides fine-grained **data access control** (e.g., to restrict access to the parts of a dataset which are copyrighted, proprietary or (privacy-)sensitive) and supports fast queries using Elasticsearch
- `middlecat` provides **authentication methods** to support the fine-grained **data access control** built into `amcat4` (e.g., to make datasets available for which data owners have restricted full text access)
- `amcat4client` offers a user interface, which makes it easy to **query documents from `amcat4` via a web interface**, share data with collaborators or the public and present your corpora to stakeholders, the community or the public
- `amcat4apiclient` provides bindings to **manage and query corpora from the `Python` programming language** via `amcat4`'s REST API
- `amcat4r` provides bindings to **manage and query corpora from the `R` programming language** via `amcat4`'s REST API

These core packages can be extended by powerful addons which provide additional features:

- `annotinder` which let's you **manually annotate documents** with an appealing web interface (which also looks great on mobile!) and the possibility to deploy it to the web. There is also an R client!
- `nlpipe` can be used for **advanced document pre-processing and machine learning** tasks. You can't share your full text? How about letting `nlpipe` apply word embeddings on your corpus and share the embedding instead of the full text!

You can use many of these packages individually or you create a full setup, which would look something like this:

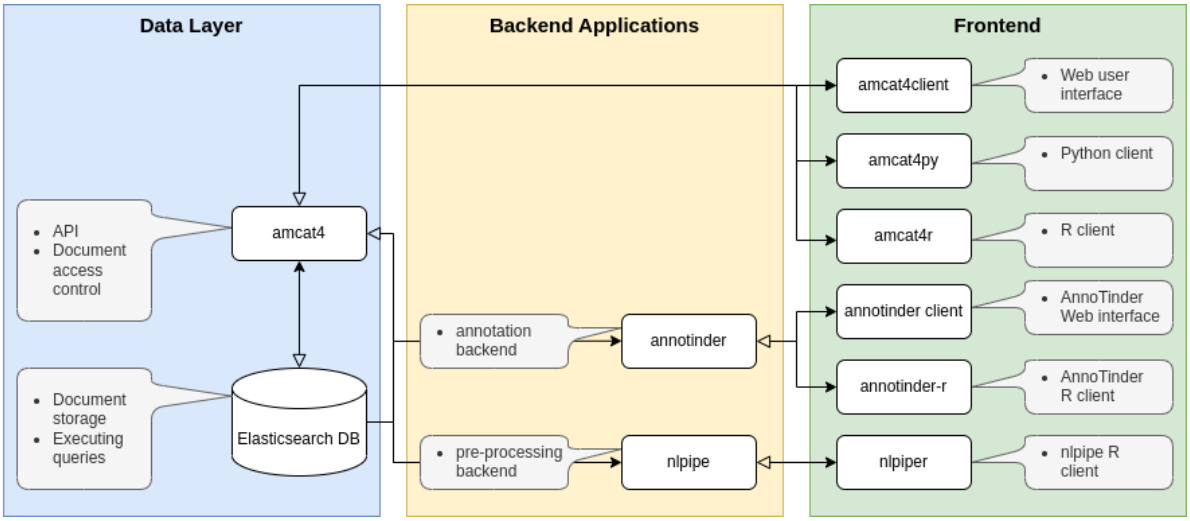


Figure 1.1: amcat-framework

It might seem like this is overly complicated, given that all of the features are also available in other software packages, some of which you will also be familiar with already. However, the main reason that functions are split between different software modules is to make development easier and more transparent. **If you are only interested in amcat's capabilities, you can use it on our servers or conveniently install everything at once through docker.**

If you want to learn more about the project, have a look at the about chapter.

# 2 Getting started

> ! Will change soon
>
> **This is work in progress.** Passages highlighted in red are likely to change soon.

In this chapter, we show you how to download, set up, and start AmCAT.

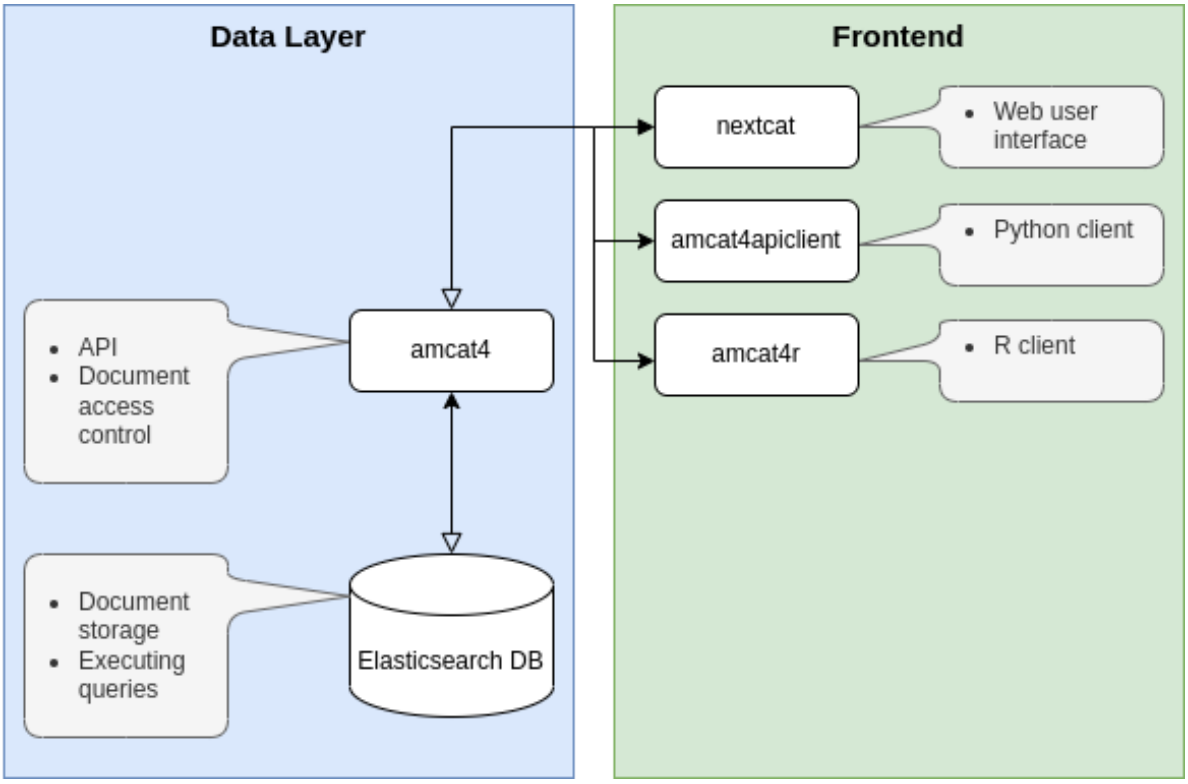


Figure 2.1: AmCAT instance after following this chapter

In the Installing and accessing AmCAT section, it is sufficient if you choose one of the subsections to follow. We explain how you can run AmCAT

- on our servers, which we recommend for testing purposes;

- through a Docker image, which we recommend for most people who want to conduct a research project and/or share data online;
- or install AmCAT directly on your system, which we only recommend for advanced users, who want a customised setup.

In the Frontend section, it makes sense to cover the amcat4client, which provides a react web interface to query data. Then you can select to either install the R or Python client.

## 2.1 Installing and accessing AmCAT

### 2.1.1 Run on our servers

> ❗ Coming soon...

### 2.1.2 AmCAT in your personal computer

Let's first install AmCAT in your computer. For this, you need to: 1.Install Docker, 2.Copy the "docker-compose.yml" file in your computer, and 3.Run the file via terminal. It sounds hard, but do not worry, I will visually walk you through each step.

#### 2.1.2.1 Install Docker

> 💡 Why do we use Docker for installation?
>
> Functionally, Docker containers are a cross-platform installation format that makes it trivially easy to install software packages on Linux, MacOS and Windows without needing users to deal with dependency issues or installation quirks on different systems.[1] A side effect is that we can easily develop AmCAT for all operating systems at once and you can be sure that we do not fall behind on developing AmCAT for your operating system of choice.

Docker is a software that allows easy creation and portability of software in something called containers. AmCAT is stored in one of such containers. To install it, you need to first install Docker: https://www.docker.com/get-started/. To install Docker, you can head over to the Docker website to get Docker Desktop or the Docker Engine and Docker Compose to use

[1] Technically, it is a little more complicated, as Docker containers have many similarities to virtual machines. However, for most users that technical background is not really important. If you want to learn more, have a look here.

Docker from the command line.[2] For a visual guide, you can check the video Johannes Gruber (2024) made: https://www.youtube.com/watch?v=iMyCdd5nP5U. Remember that to use Docker, you do NOT need to create an account.

Docker Desktop makes installation easier if you do not use a package manager (which many Windows and MacOS users do not). Docker Desktop looks like this:

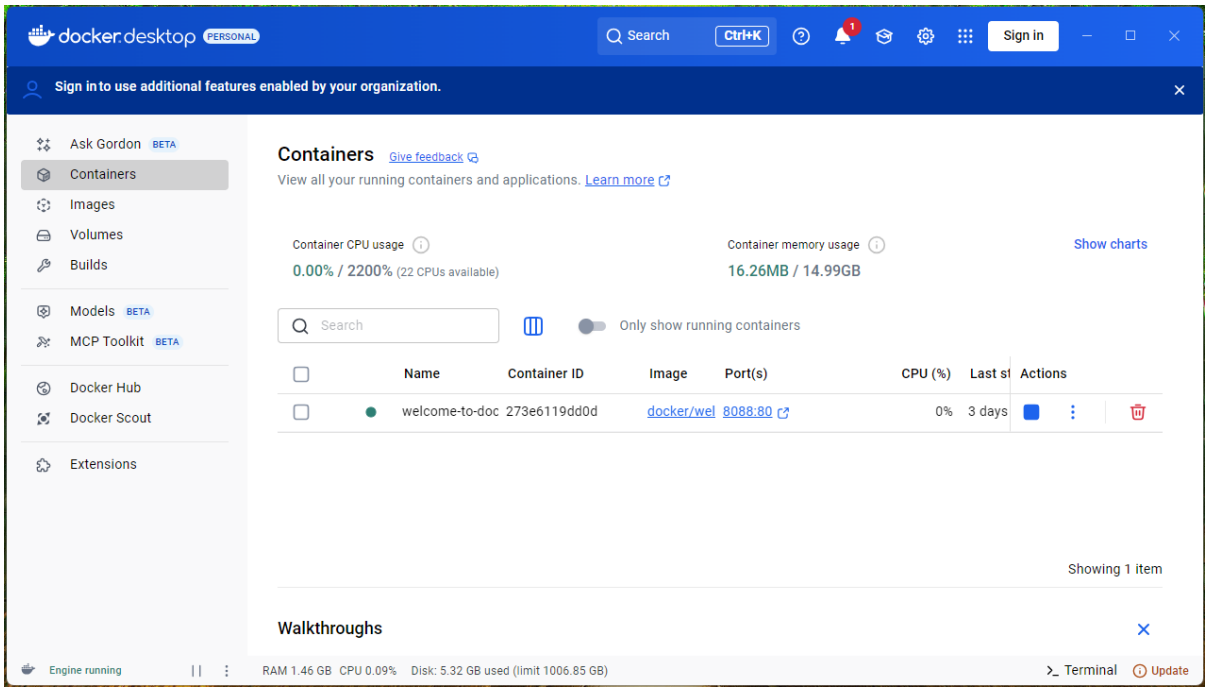


Figure 2.2: Docker_Desktop

#### 2.1.2.2 Copy the docker-compose.yml file in your computer

To install the AmCAT data layer, you should use our Docker Compose file. You can get it from here (save the file as "docker-compose.yml").

In general, Gruber (2024) explains how to copy a "docker-compose.yml" file depending on the Operating System (OS). Here, we visually follow the steps for the AmCAT "docker-compose.yml" using a Windows OS. Access the AmCAT "docker-compose.yml" here: https://raw.githubusercontent.com/ccs-amsterdam/amcat4docker/main/docker-compose.yml.

[2]On Ubuntu and Debian, the Docker Compose version in the repos is too old. You can use this gist to update it: https://gist.github.com/JBGruber/db4de49ee106889b969a277f800c05b8

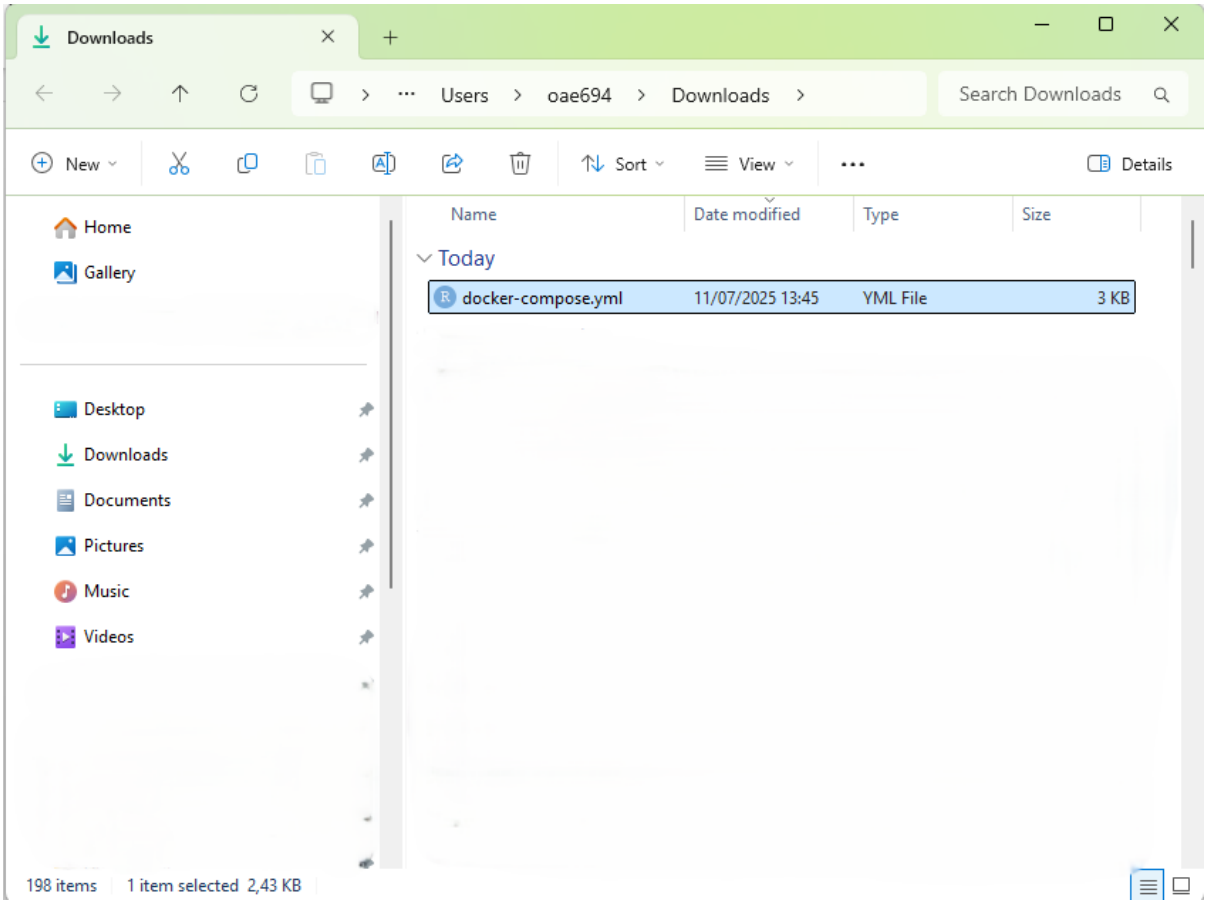


Figure 2.3: AmCAT_DockerCompose1

Now, press the right button of your mouse. That will display the following menu:

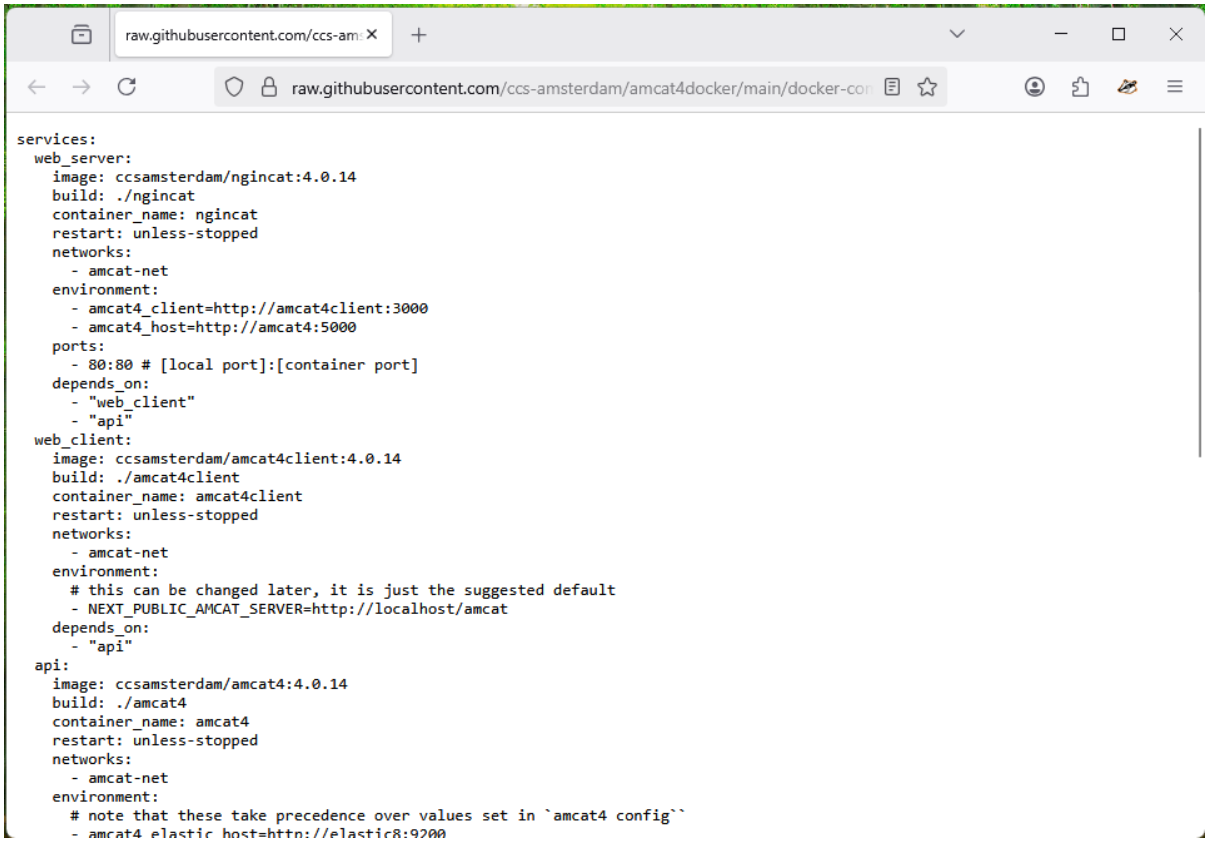


Figure 2.4: AmCAT_DockerCompose2

From there, press the option “Save Page As…” and save it in any location you want:

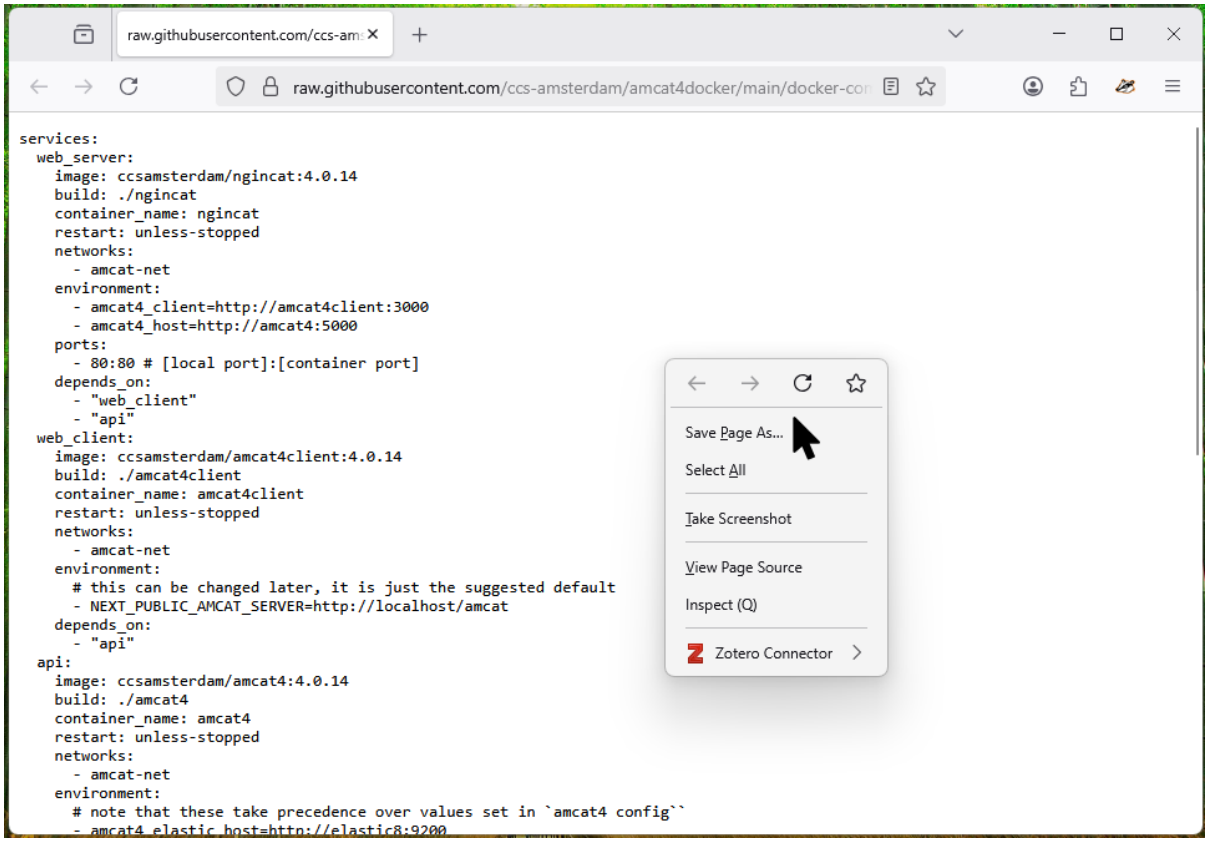


Figure 2.5: AmCAT_DockerCompose3

By default, Windows saves any “weird” file as “txt”:

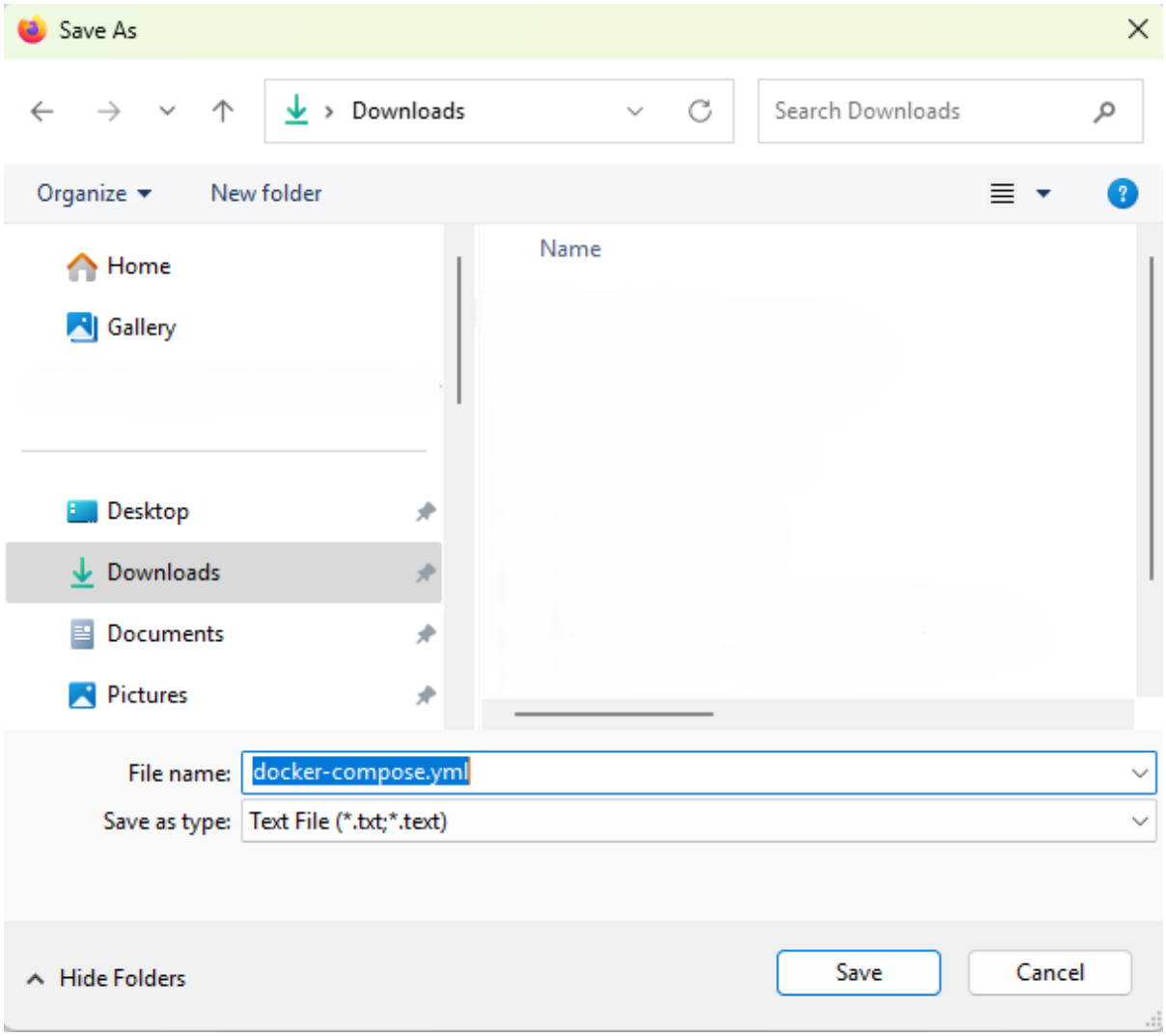


Figure 2.6: AmCAT_DockerCompose4

So, we need to explicitly tell Windows that our file is a “yml” type. To do so, you need to press View/Show/File name extensions:

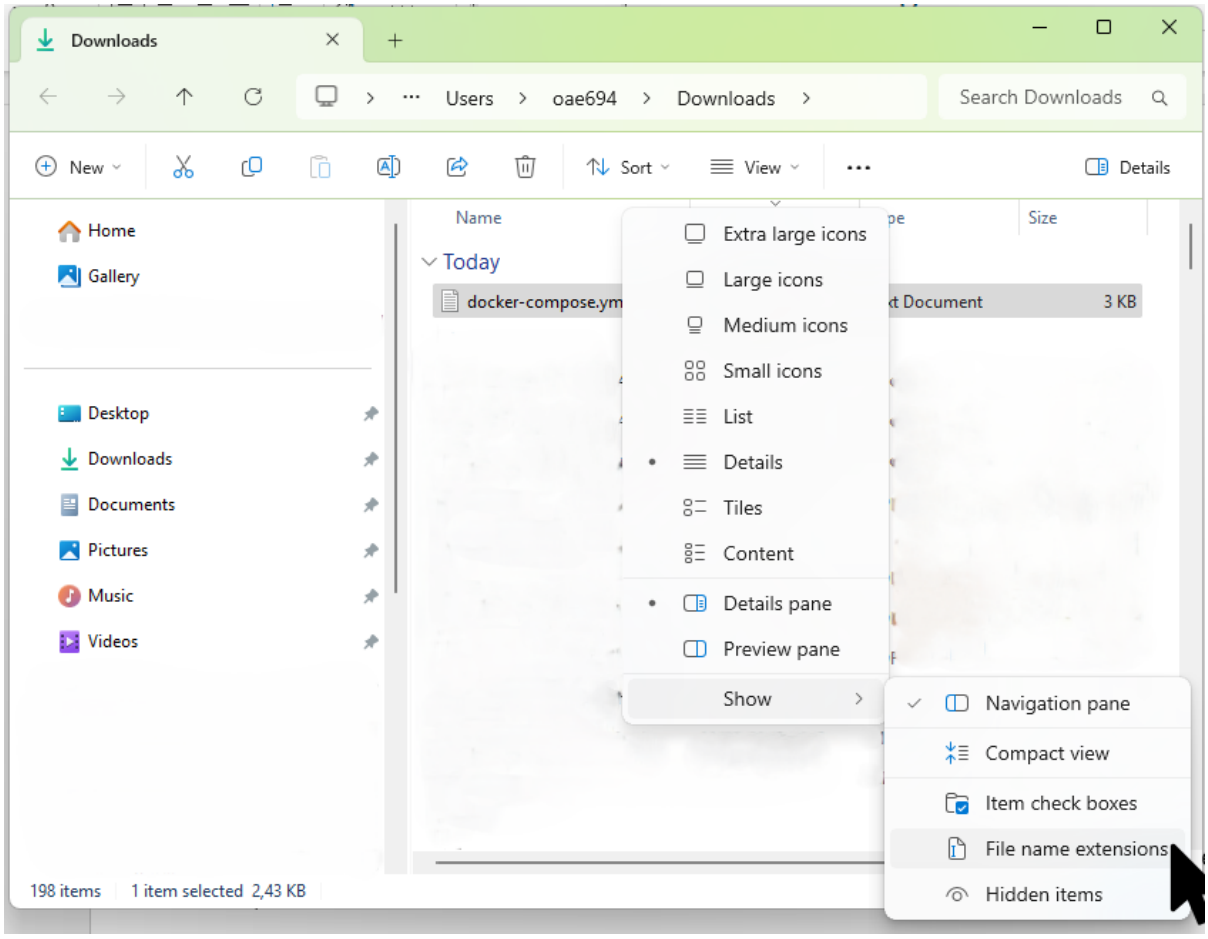


Figure 2.7: AmCAT_DockerCompose5

Now, you can see that Windows added the “txt” extension to the file:

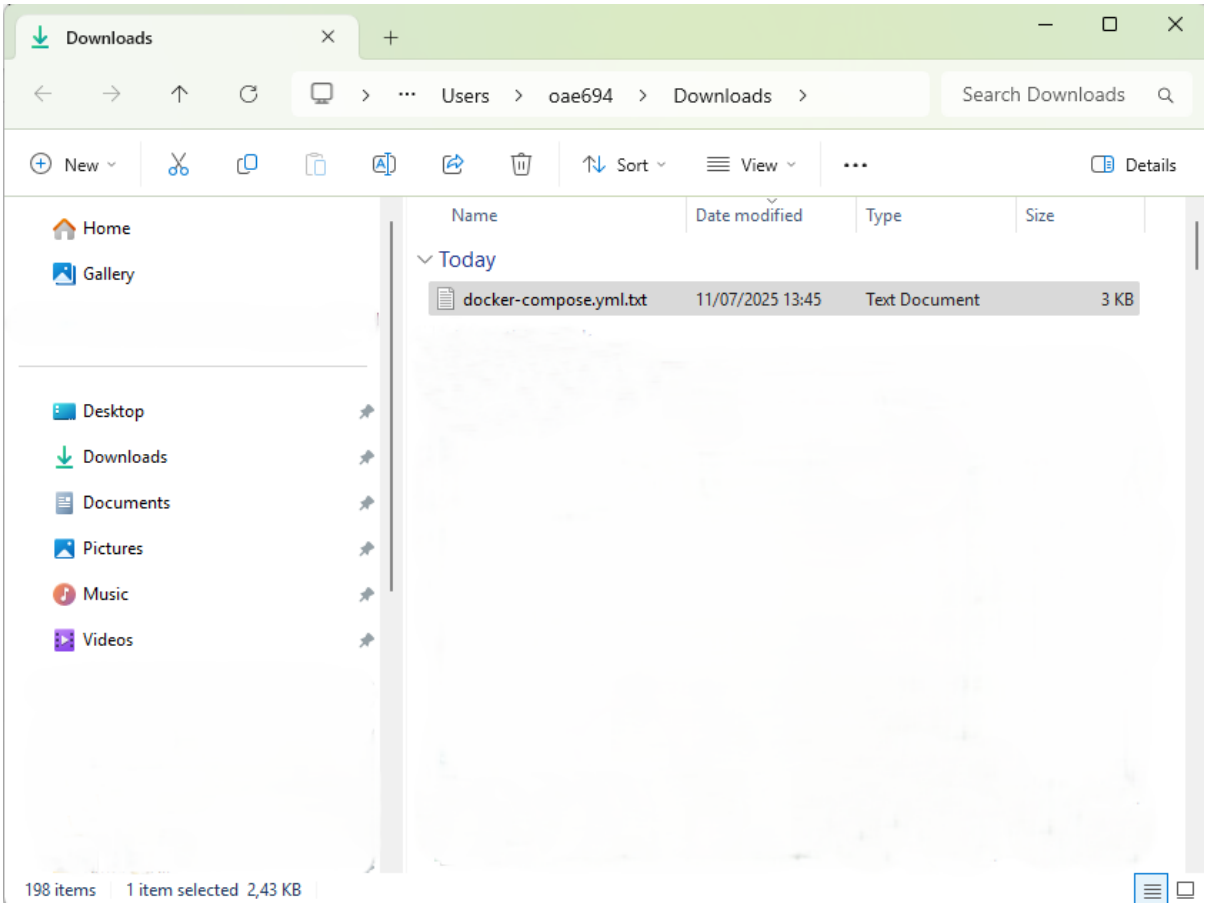


Figure 2.8: AmCAT_DockerCompose6

All we need to do is to delete the extra ".txt":

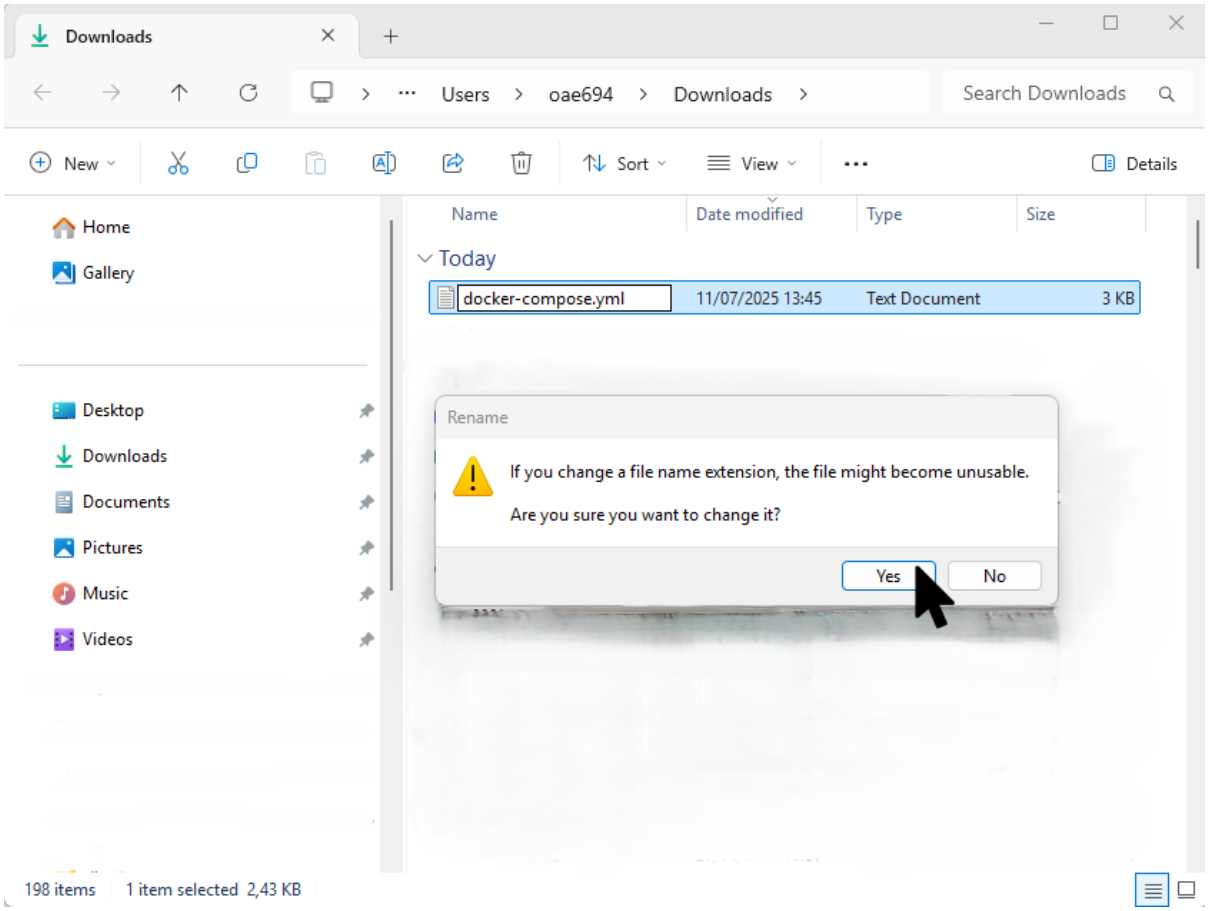


Figure 2.9: AmCAT_DockerCompose7

Now the file type changes to “YML file”.

#### 2.1.2.3 Run the file via terminal

Now, we will run the file via terminal. Gruber (2024) https://www.youtube.com/watch?v=iMyCdd5nP5U shows how to do so depending on the OS. Here, we will do it for Windows. So, let’s open “Windows PowerShell”:

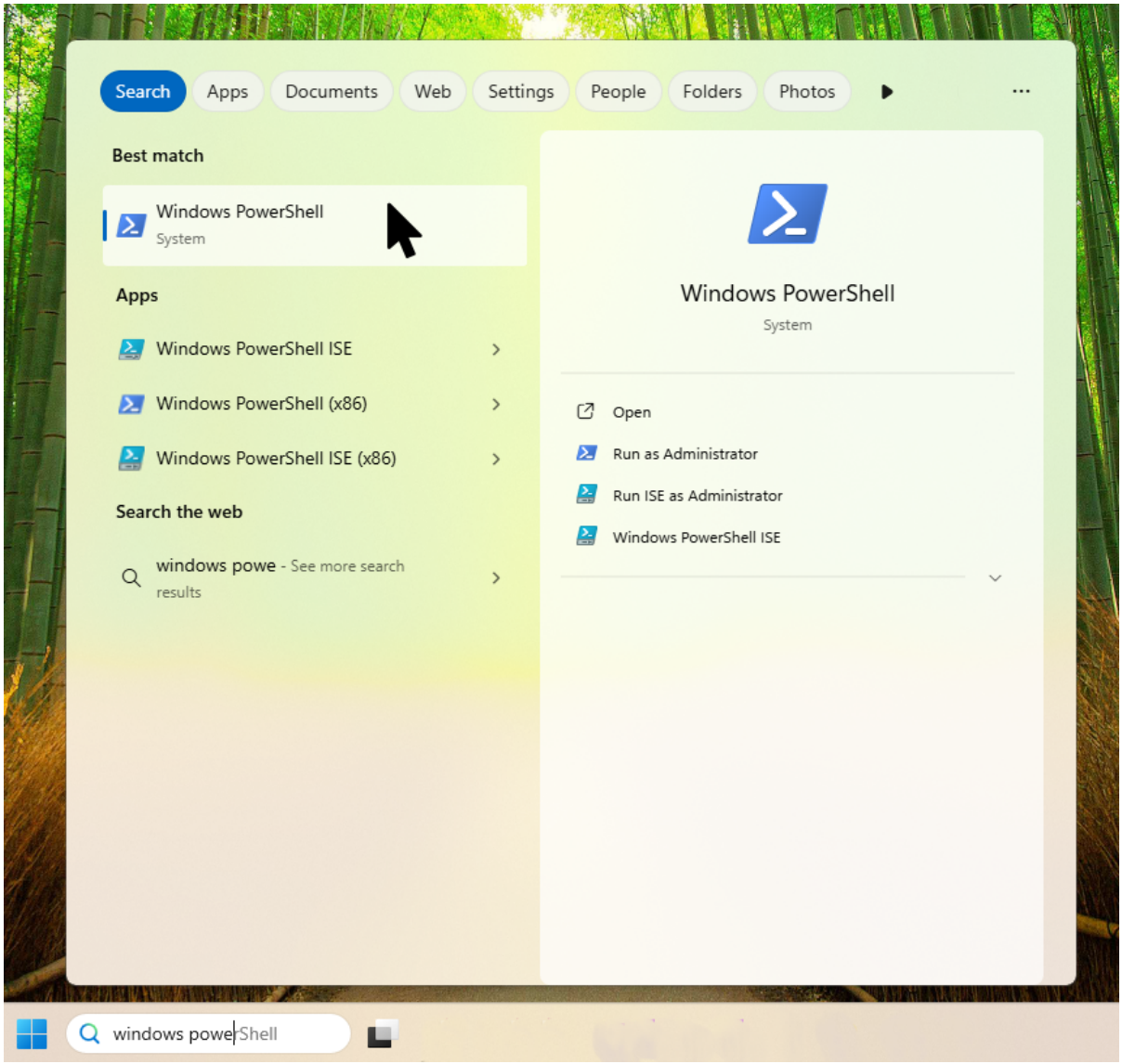


Figure 2.10: Windows Power Shell

This will open the following window:

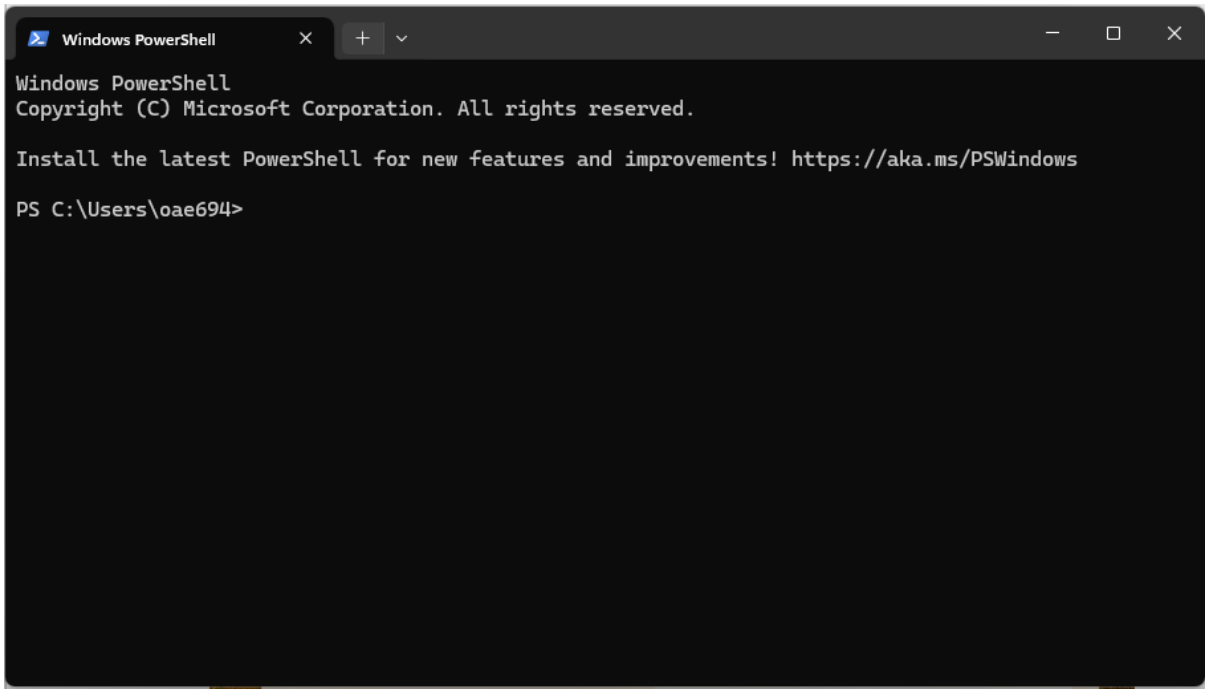


Figure 2.11: Windows Power Shell

In here, we will: (a) move to the folder where the "docker-compose.yml" is, and (b) run the "docker-compose.yml" file.

**(a) Move to the folder where the "docker-compose.yml" is**

For that, you will use the command `cd` followed by the path where the file is. For example, mine is in the "Download" folder, therefore, my command looks like:

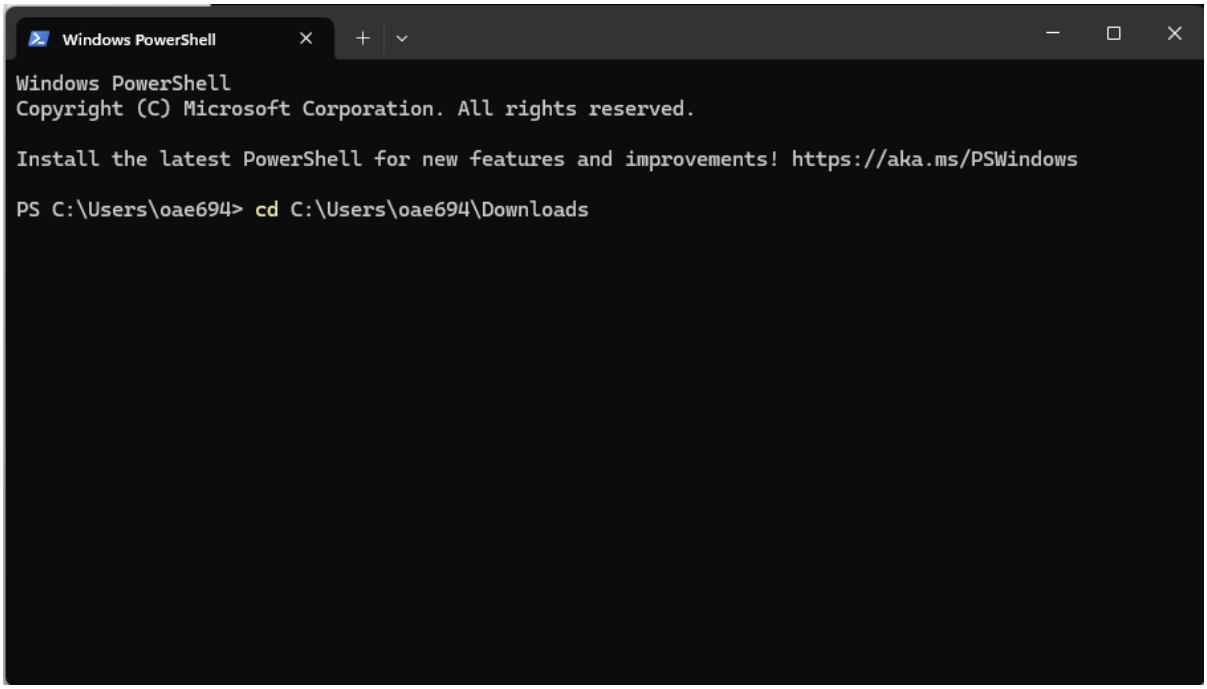


Figure 2.12: Change Directory

Now press enter. This will modify how the previous link looks like:

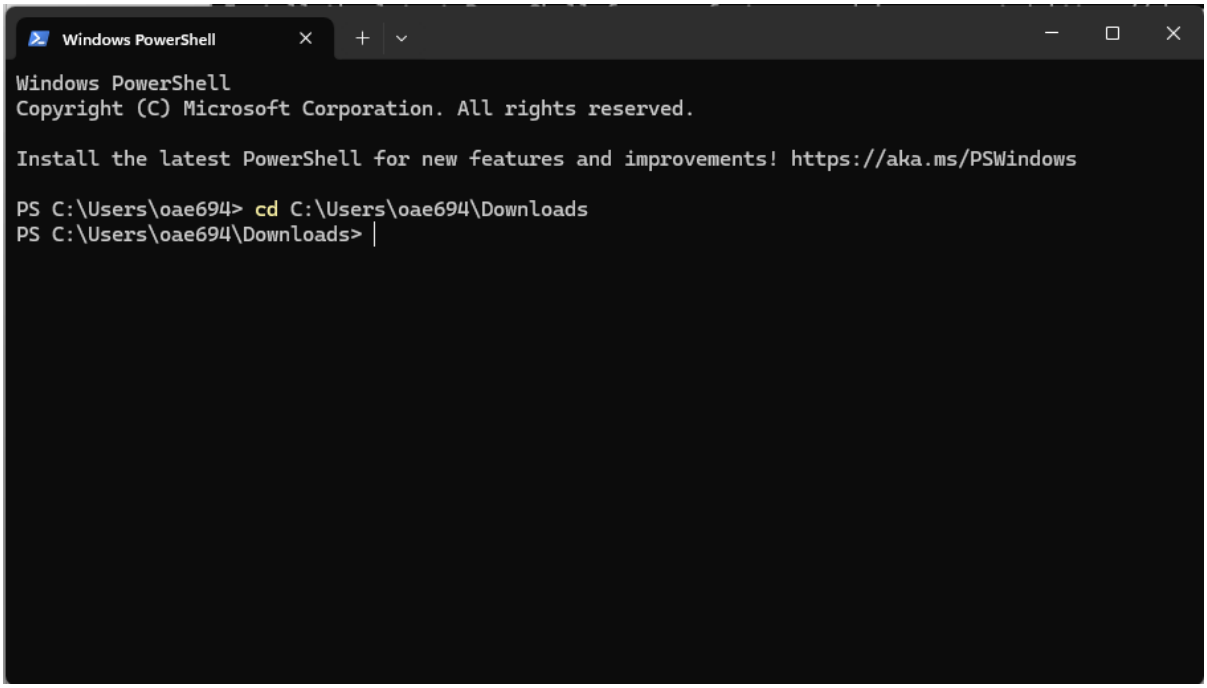


Figure 2.13: Change Directory2

**(b) Run the "docker-compose.yml" file**

Now, you need to run the following command:

```
# docker-compose up --pull="missing" -d
```

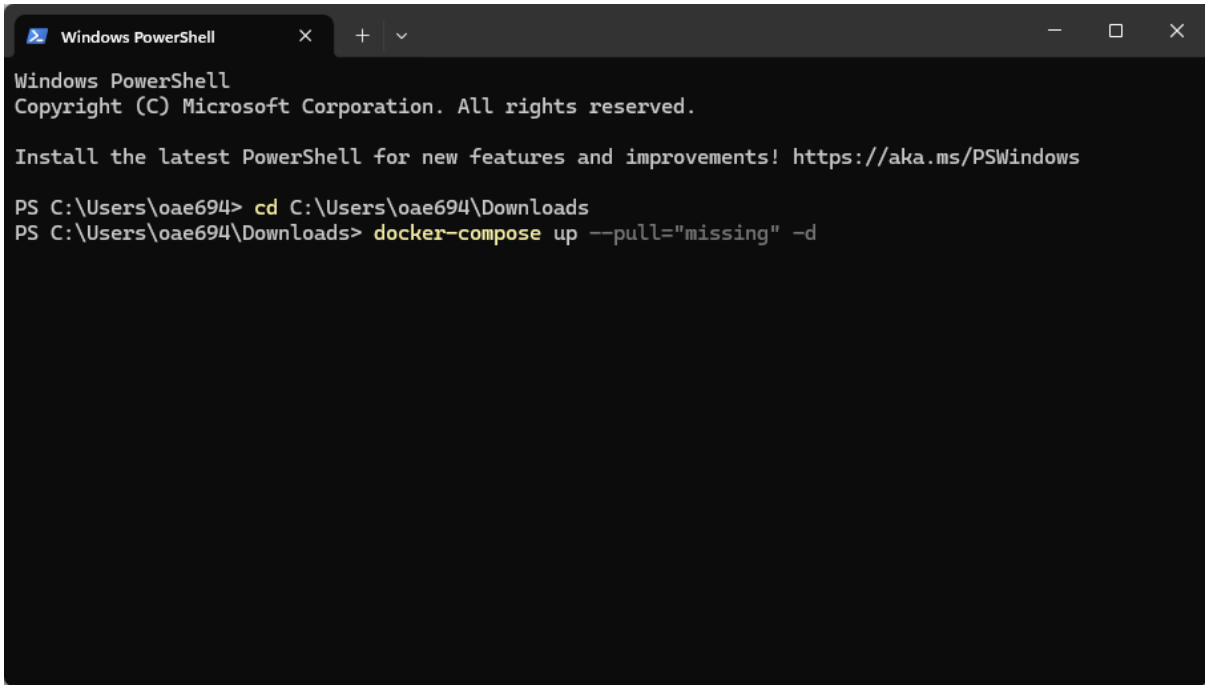


Figure 2.14: Docker compose

Once it finishes running, it will show the following: