Accessing NHANES with Docker

Below you can find instructions for installing and running the Docker container for NHANES data, created by the CCB.

Note that these instructions go into how to:

1. Install Docker/Docker Desktop.
2. Download the docker image.
3. Create and run the docker container.
4. Connect to the running docker container.

However, they do not go into how to actually access the NHANES data once connected to the container. We will be learning how to do this in class.

Installing Docker

We will be installing Docker Desktop, a graphical application which provides useful additional functionality and ease of access for using Docker containers. You can go to the documentation for Docker Desktop, or go straight to the installation instructions for Mac or Windows. It is recommended for Linux users to instead install Docker Engine (see below).

Notes for Windows Users

It is recommended to use the Windows Subsystem for Linux (WSL) backend as opposed to the Hyper-V backend. WSL is official Microsoft software which lets you install a Linux distribution on your Windows machine. Installing WSL will also allow you to easily run other software designed for Linux systems and directly use Bash command-line tools.

You can /install WSL by following the instructions here. Feel free to choose any of the possible distributions (we like Ubuntu).

Notes for Mac Users

You will need to know whether your Mac computer is using an Intel or Apple silicon processor. You can follow the instructions here to find this information.

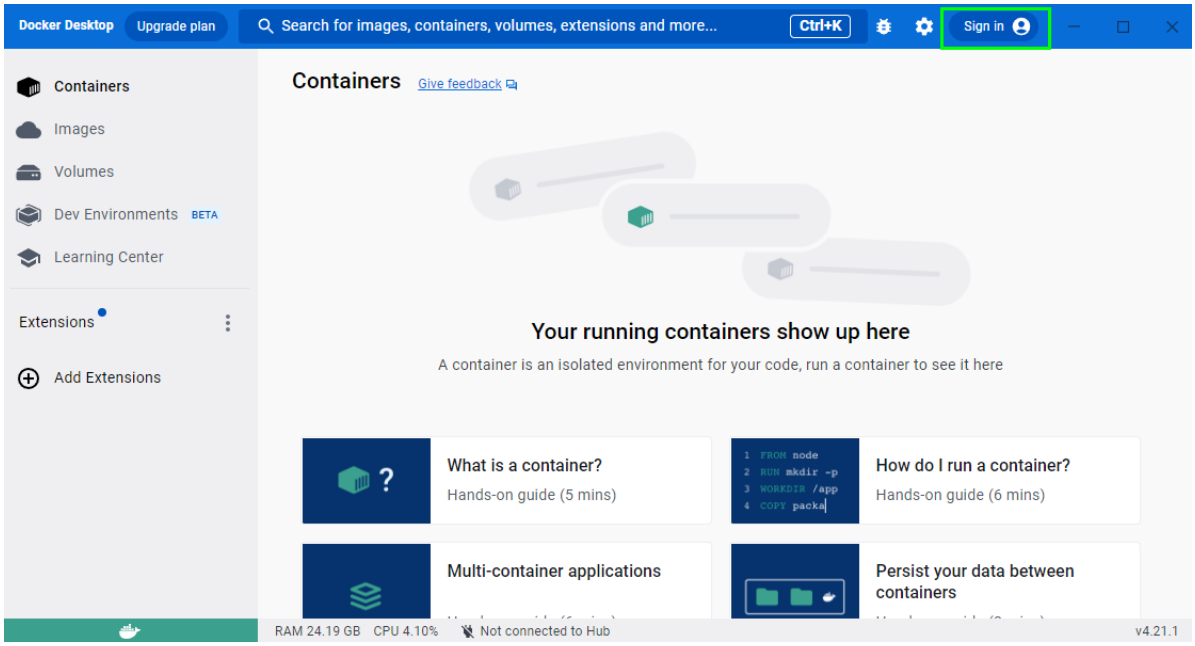
Notes for Linux Users

While Docker Desktop can be run on Linux, support is limited. This is because the core Docker containerization software (now called Docker Engine) is originally designed for Linux systems, and Docker Desktop enables Windows and Mac users to easily interact with containers as Linux users are already able to with the original software. You can install Docker Engine by following the instructions here.

Downloading the container

Option 1: Using Docker Desktop

1. Make a free DockerHub account at this link.
2. Login to your DockerHub account in Docker Desktop:



3. Search for `hmsccb/nhanes-workbench` in the search bar, to the left of where you previously clicked `Sign In` in the top blue bar. You should get one image as the result.

4. Make sure the `Tag` dropdown is set to latest and click `Pull`.

Option 2: Using the command line

Open a terminal (if in Mac or Linux) or the command prompt (in Windows). Type the command

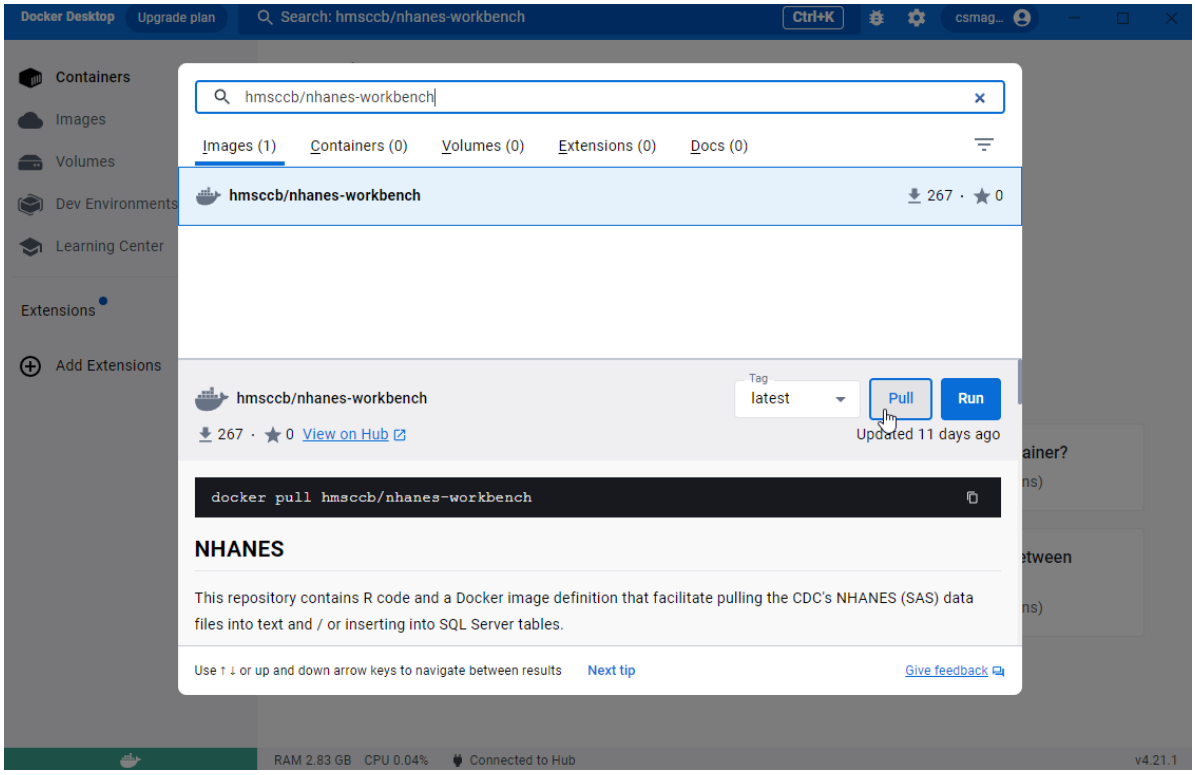


Figure 1: Search result

```
docker pull hmsccb/nhanes-workbench:latest
```

into the terminal.

After you've downloaded the container, you should see it when you look at the `Images` tab in Docker Desktop.

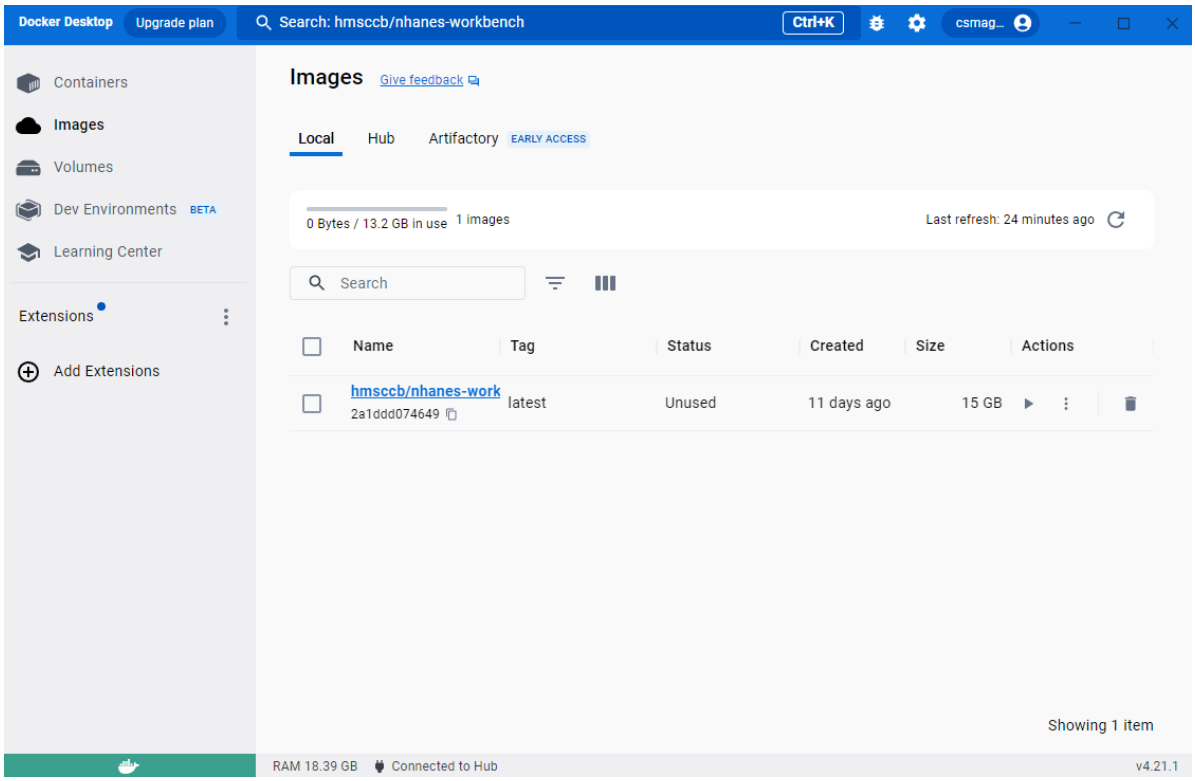


Figure 2: Downloaded NHANES Image

Running the container

Unfortunately, right now the container cannot be run from Docker Desktop due to Docker Desktop not allowing us to setup multiple ports. However, we can simply run the container from the command line.

On Mac or Linux, copy and paste the following command into a terminal:

```
docker \
    run \
        --rm \
```

```
--name nhanes-workbench \
-d \
-v LOCAL_DIRECTORY:/HostData \
-p 8787:8787 \
-p 2200:22 \
-p 1433:1433 \
-e 'CONTAINER_USER_USERNAME=USER' \
-e 'CONTAINER_USER_PASSWORD=PASSWORD' \
-e 'ACCEPT_EULA=Y' \
-e 'SA_PASSWORD=yourStrong(!)Password' \
hmsccb/nhanes-workbench:hmsccb/nhanes-workbench:latest
```

If you're using the WSL backend on Windows (recommended), you can open a linux terminal by hitting the windows key and typing in `WSL`:

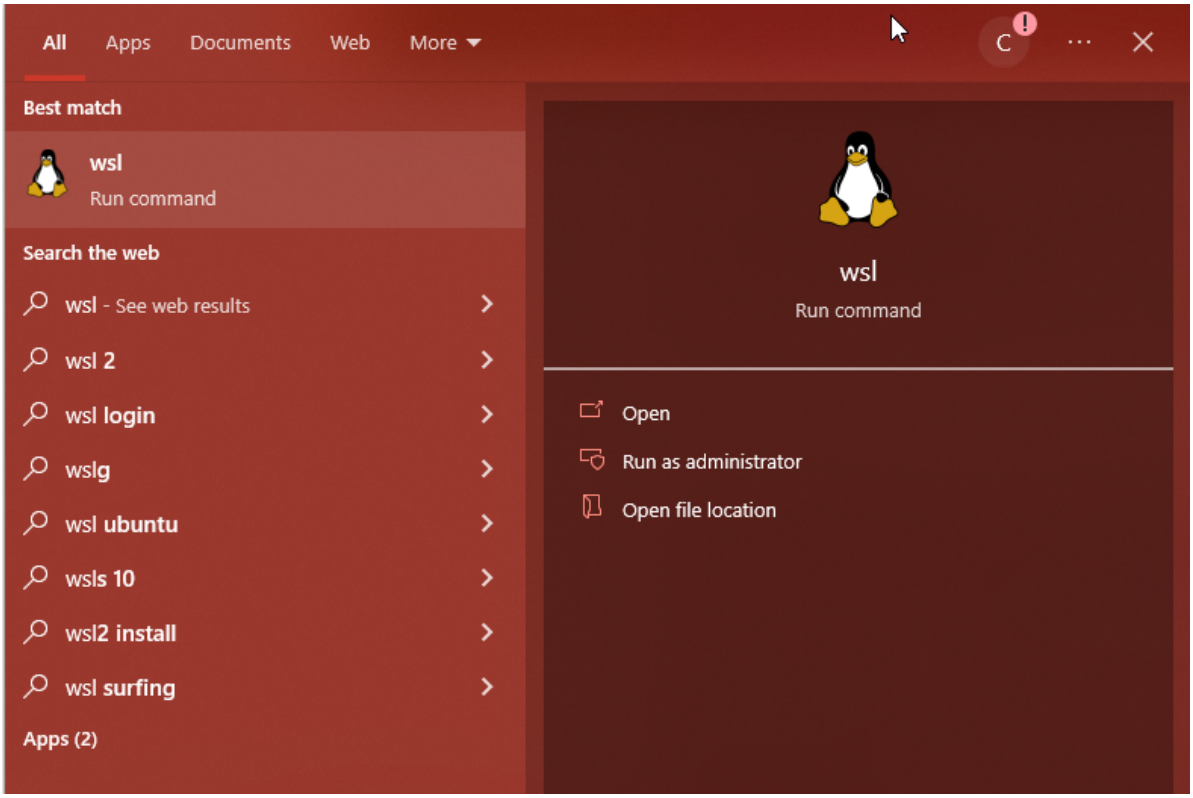


Figure 3: Running WSL

Paste the run command above into your terminal to run the container.

If you're using the Hyper-V backend on Windows, copy and paste this command into the command prompt (this is the same as the Mac/Linux command but doesn't use the \ character

to make it multiline):

```
docker run  --rm --name nhanes-workbench -d  -v LOCAL_DIRECTORY:/HostData -p 8787:8787 -p 220
```

You can learn more about what each part of this command does below.

Details

Parameters

`LOCAL_DIRECTORY` is a directory on the host that you would like mounted at /HostData in the container. Can be omitted.

`CONTAINER_USER_USERNAME` is the name of the user in the container that will be created at runtime. You can connect via `ssh` or `RStudio Server` with this user name.

`CONTAINER_USER_PASSWORD` is the password of the user in the container that will be created at runtime. You can connect via `ssh` or `RStudio Server` with this password.

`ACCEPT_EULA` is required for SQL Server to successfully start

`SA_PASSWORD` is the password for the SQL Server `sa` account. See here for complexity requirements.

Port Forwarding

These options control port forwarding from the container to the host:

```
-p 8787:8787 \
-p 2200:22 \
-p 1433:1433 \
```

Port 8787 is used for access to the RStudio Server HTTP server. Port 2200 on the host provides access to ssh server in the container. Port 1433 provides access to SQL Server running in the container.

Confirming that the container is running

Inside Docker Desktop, in the `Containers` tab you should now see the `nhanes-workbench` container.

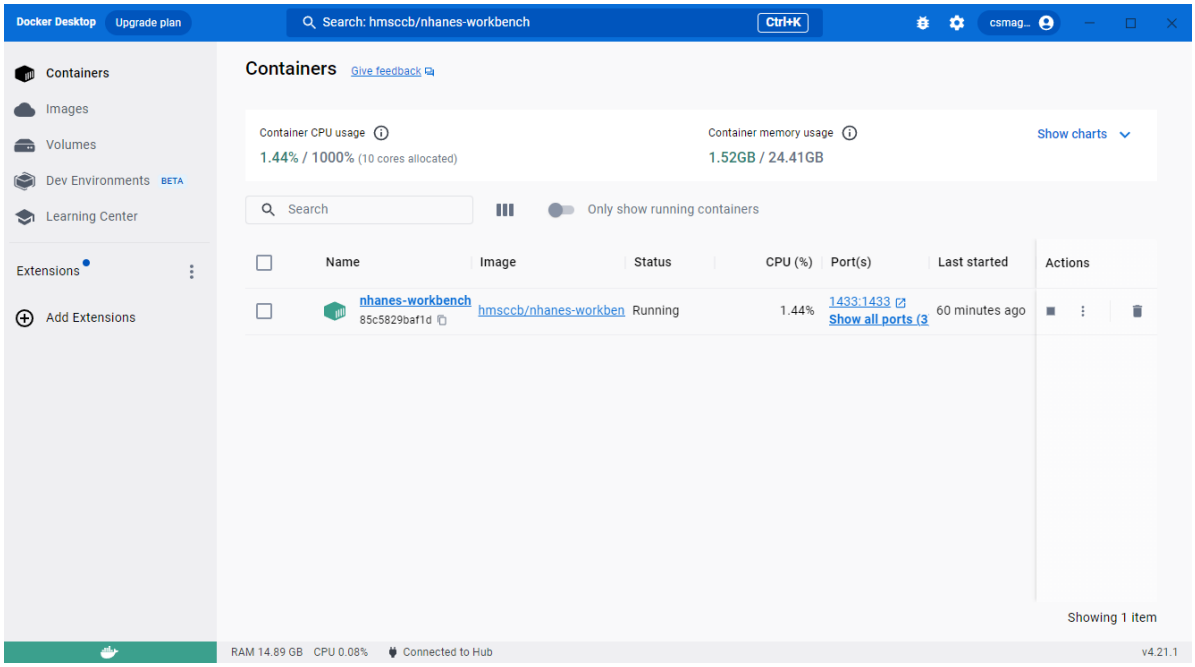


Figure 4: The running container

Inspecting the container

If you click on the `nhanes-workbench` entry, you can see and interact with the running container in a number of tabs.

Logs

This tab shows all of the output of the running container. If you are having problems with the container, the logs are the place to try to figure out what is going wrong.

Inspect

This tab gives you the general container information. You can check here to see where mounted files in the container live under `Mounts` or double check what you set your password to under `Environment`.

Terminal

This tab gives you a command-line interface into the container while it is running. Here you can manipulate files, run command-line tools, or anything else you'd need to do at the terminal inside the container.

Files

This tab lets you easily explore all of the files in the container.

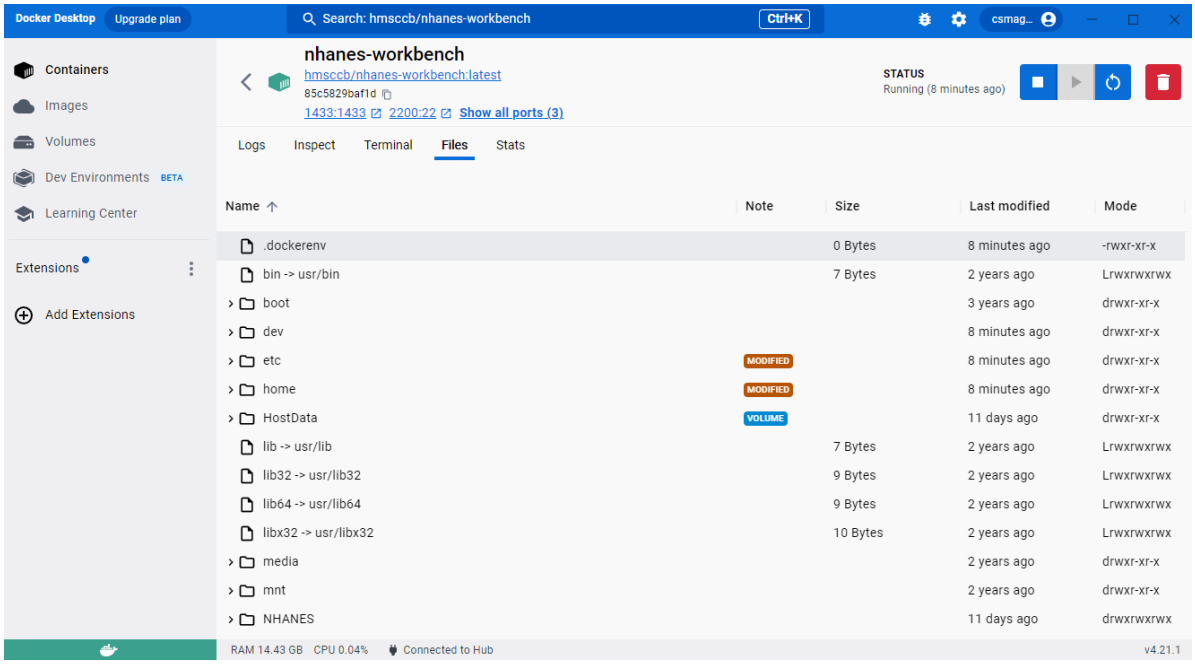


Figure 5: Inspecting the container's files

You can select files or folders by right/command clicking on them, then select `Save` to save a copy of that file somewhere on your computer outside of the container.

Connecting to the container

Running RStudio

Once the Docker container is running, you can connect to it by pointing your web browser to port 8787. To do this, simply type `http://localhost:8787` into your browser's address bar. You'll then need to login using `USER` and `PASSWORD` or whatever you set `CONTAINER_USER_USERNAME` and `CONTAINER_USER_PASSWORD` to when creating the container. You will then be in an RStudio instance inside the container.

Connecting to the SQL server directly

Database connectivity is enabled on TCP port 1433 on the host. Any standard tools that work with SQL Server (Azure Data Studio, SSMS, ODBC, JDBC) can be aimed at this port on the host to work with the DBs in the container.

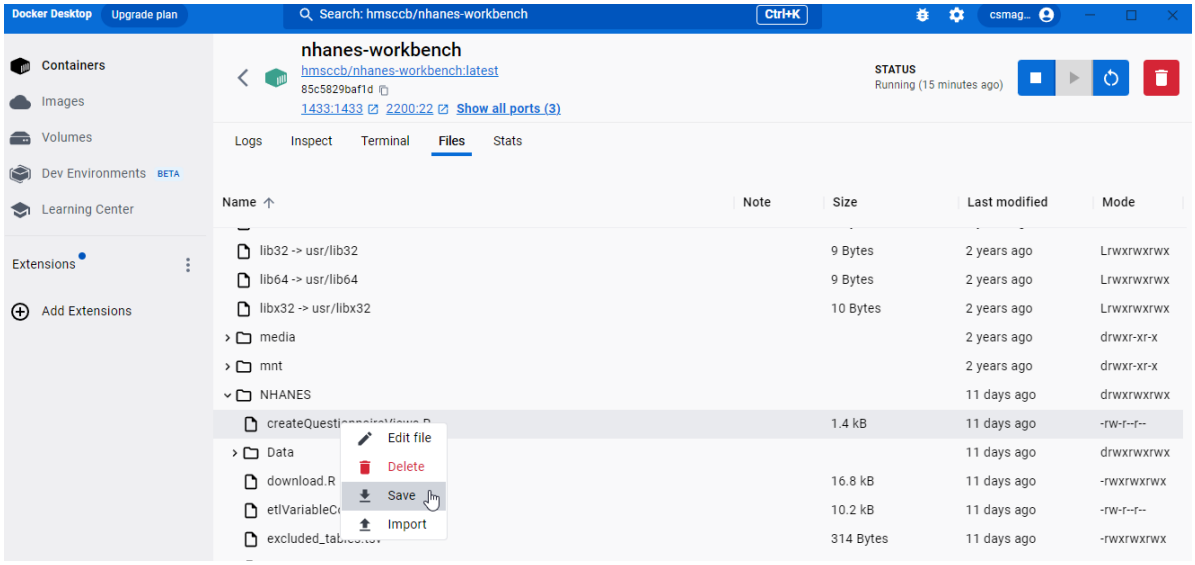


Figure 6: Saving a file to outside docker

If you are already familiar with relational databases, this can be a great option for accessing the NHANES data.

For instance, in Azure Data Studio you can connect to `localhost` using the username `SA` and password `yourStrong(!)Password` or whatever `SA_PASSWORD` was set to when you created the container. Note that you'll have to set the authentication to `SQL Login`.

SSH

You can can ssh into the container, e.g.:

```
ssh USER@HOST_ADDRESS -p 2200 -o GlobalKnownHostsFile=/dev/null -o UserKnownHostsFile=/dev/nu
```

If you are running SSH on the same host where the container is running:

```
ssh USER@localhost -p 2200 -o GlobalKnownHostsFile=/dev/null -o UserKnownHostsFile=/dev/null
```

More Details

You can find more details and some example scripts for running the SQL server on its Github page.

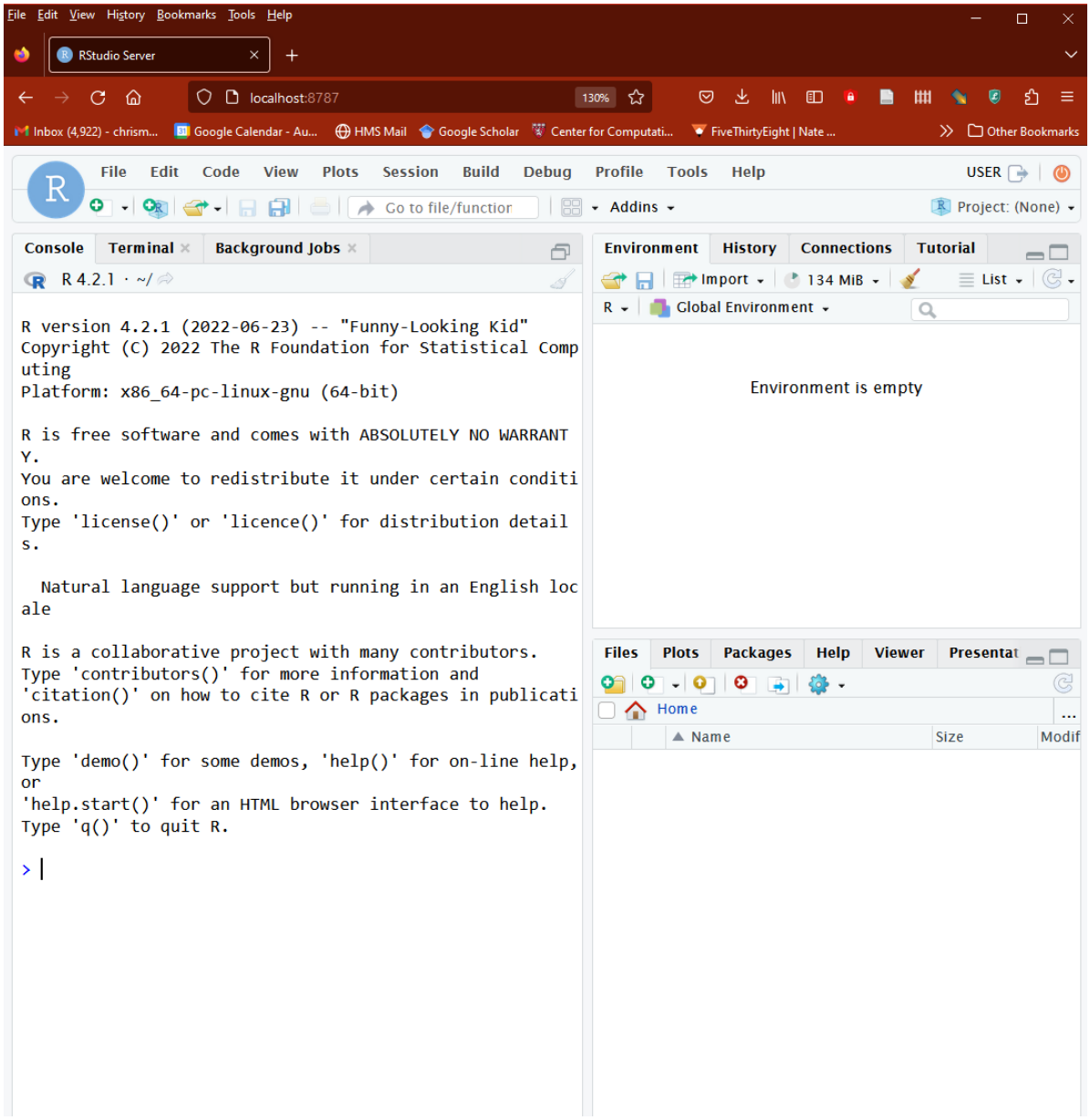


Figure 7: Running RStudio in your browser

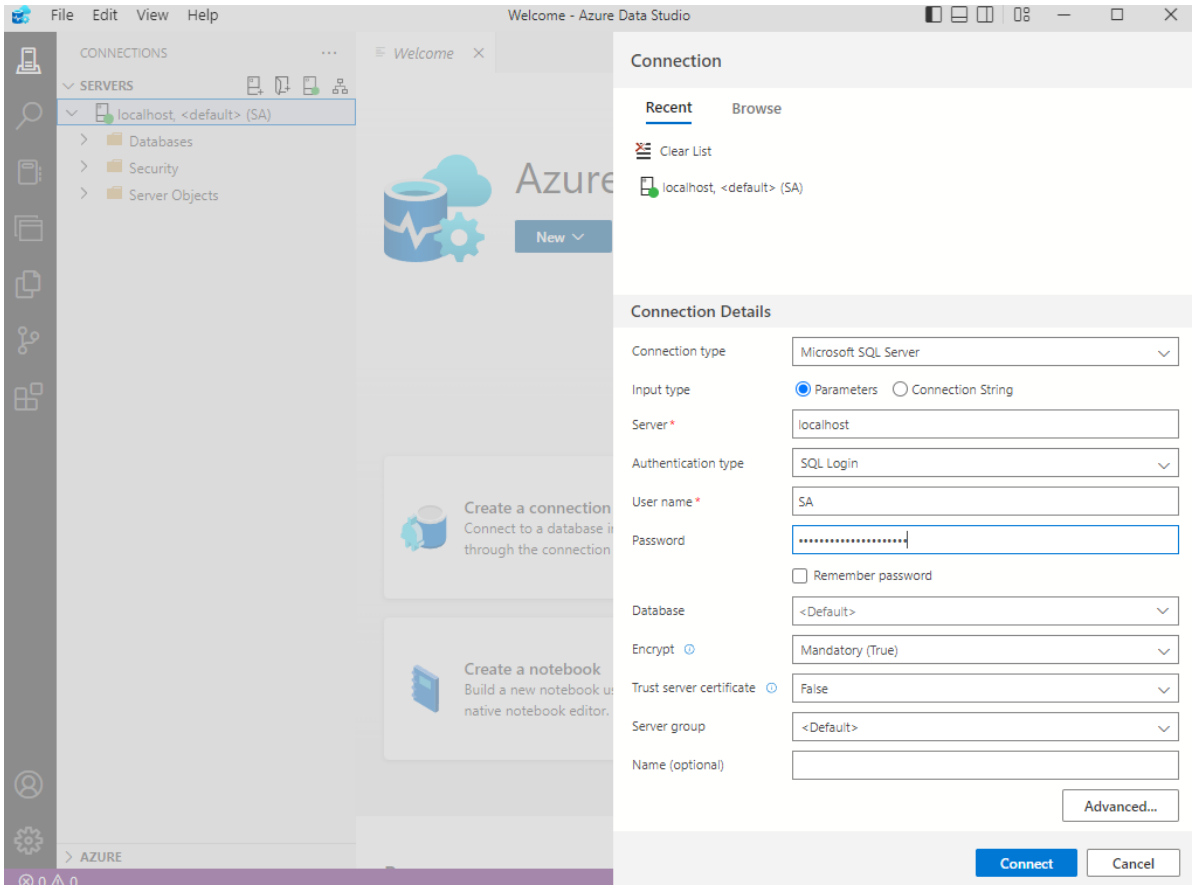


Figure 8: Connecting to the SQL server