

Big Data in Economics

Lecture 12: Docker

Grant McDermott

University of Oregon | EC 510

Table of contents

1. Prologue
2. Examples
 - Base R
 - RStudio+
3. Sharing files with a container

Prologue

Install Docker

- Linux
- Mac
- Windows (varies by version)
 - Windows 10 Pro / Education / Enterprise
 - Windows 10 Home
 - Windows 7 / 8

Note: These lecture slides are mostly intended as a companion to the [Rocker Wiki](#) and the [ROpenSci Docker tutorial](#). Use them as a quick-start reference for common Docker commands.

Examples

Base R container

For our first example, let's fire up a simple container that contains little more than a base R installation.

```
$ docker run --rm -ti rocker/r-base
```

This will take a little while to download the first time. But afterwards the container will be ready and waiting for immediate deployment ("instantiation") on your system.

A quick note on these `docker run` flags:

- `--rm` Automatically remove the container once it exits (i.e. clean up).
- `-ti` Launch with interactive (`i`) shell/terminal (`t`).
- Type `man docker run` to see a full list of flag options.

To see a list of running containers on your system, in a new terminal window type:

```
$ docker ps
```

Base R container (cont.)

Your base R container should have launched directly into R. To exit the container, simply quit R.

```
R> q()
```

Check that it worked:

```
$ docker ps
```

BTW, if you don't want to launch directly into your container's R console, you can instead start it in the bash shell.

```
$ docker run --rm -ti rocker/r-base /bin/bash
```

This time to close and exit the container, you need to exit the shell, e.g.

```
root@09dda673a187:/# exit
```

RStudio+ container

The **Rocker Project** works by layering Docker images on top of each other in a **grouped stack**.

An important group here is the **versioned stack** that includes RStudio.

- For example, the "rstudio" image builds on top of the "r-ver" image (which itself is a versioned "r-base" image).

I'm going to collectively refer to images in this stack as **RStudio+**.

- It's not that each image in the stack contains exactly the same things...
- But they do share some important common features that will make the RStudio+ shorthand convenient for these lecture slides.

Everyone clear on what I mean by "RStudio+"? Good.

RStudio+ container (cont.)

Let's try the "tidyverse" image, which is basically base R + RStudio + tidyverse packages.

Again, this will take a minute or three to download and extract the first time. But it will be ready for immediate deployment in the future.

```
$ docker run -d -p 8787:8787 -e PASSWORD=mypassword rocker/tidyverse
```

If you run this... nothing seems to happen. Don't worry, I'll explain on the next slide.

But first, two quick asides on usernames/passwords:

- All RStudio+ images in the Rocker stack require a password. Pretty much anything you want except "rstudio", which is the default username...
- If you don't like the default "rstudio" username, you can choose your own by adding `-e USER=myusername` to the above command (the extra `-e` flag is required).

RStudio+ container (cont.)

Unlike, the "r-base" container, this time we aren't immediately taken to our R environment.

Reason: Our container is running RStudio Server, which needs to be opened up in a browser.

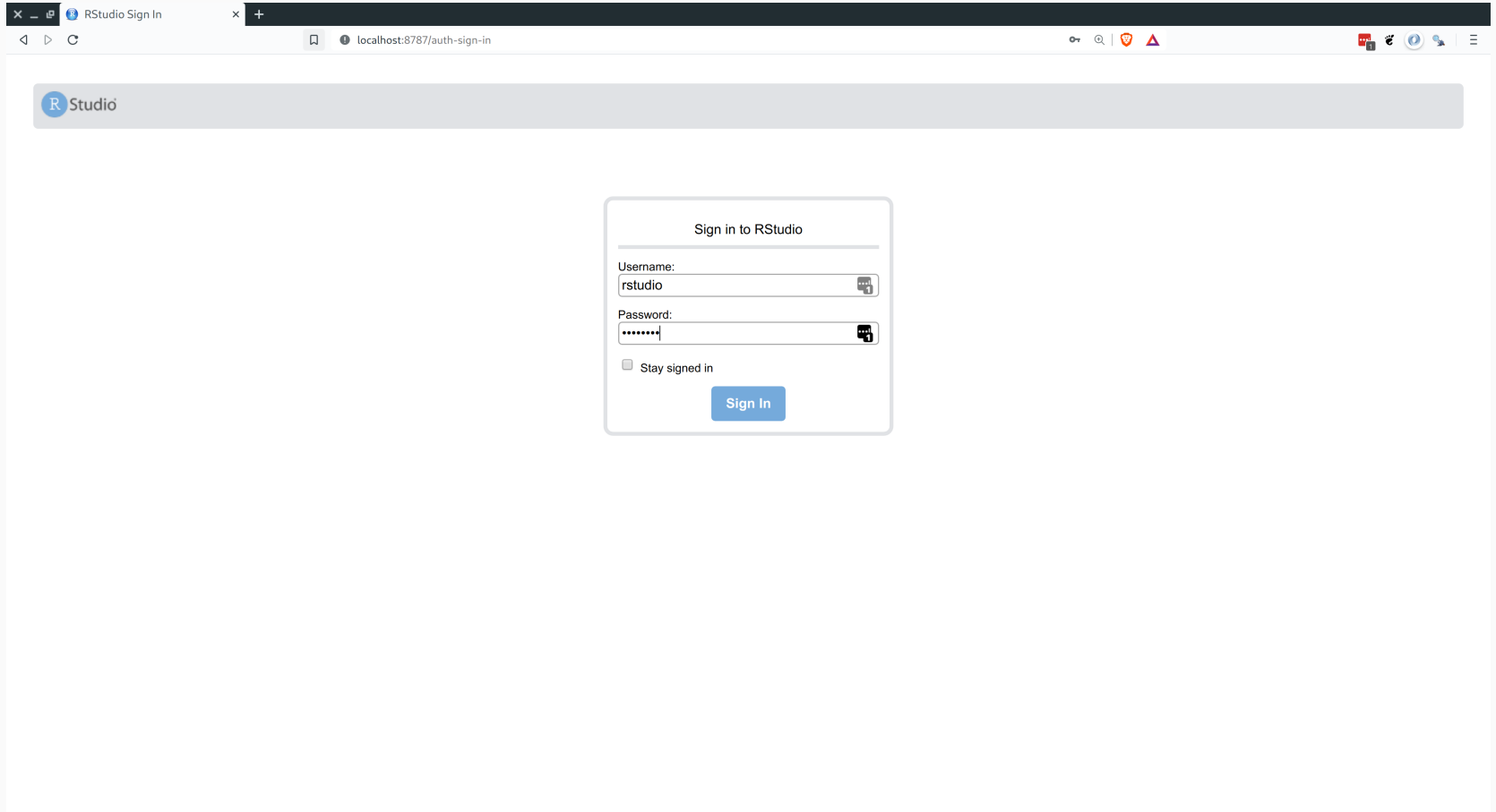
So we need to point our browsers to the relevant IP address (plus the assigned 8787 port):

- **Linux:** `http://localhost:8787`
- **Mac/Windows:** Get your IP address by typing `$ docker inspect <containerid> | grep IPAddress` (see [here](#)). Alternatively, this IP address was also displayed when you first launched your Docker Quickstart Terminal. E.g. The below user would navigate their browser to `192.168.99.100:8787`.

```
      ##          .
    ##  ##  ##      =
  ##  ##  ##  ##  ##  ==
/""""""""""""""""""""\___/ ===
 ~~~ { ~ ~~~~ ~~~ ~~~~ ~~~ ~ /  === - ~~~
      \____ o  _____/
         \    \  _____/
```

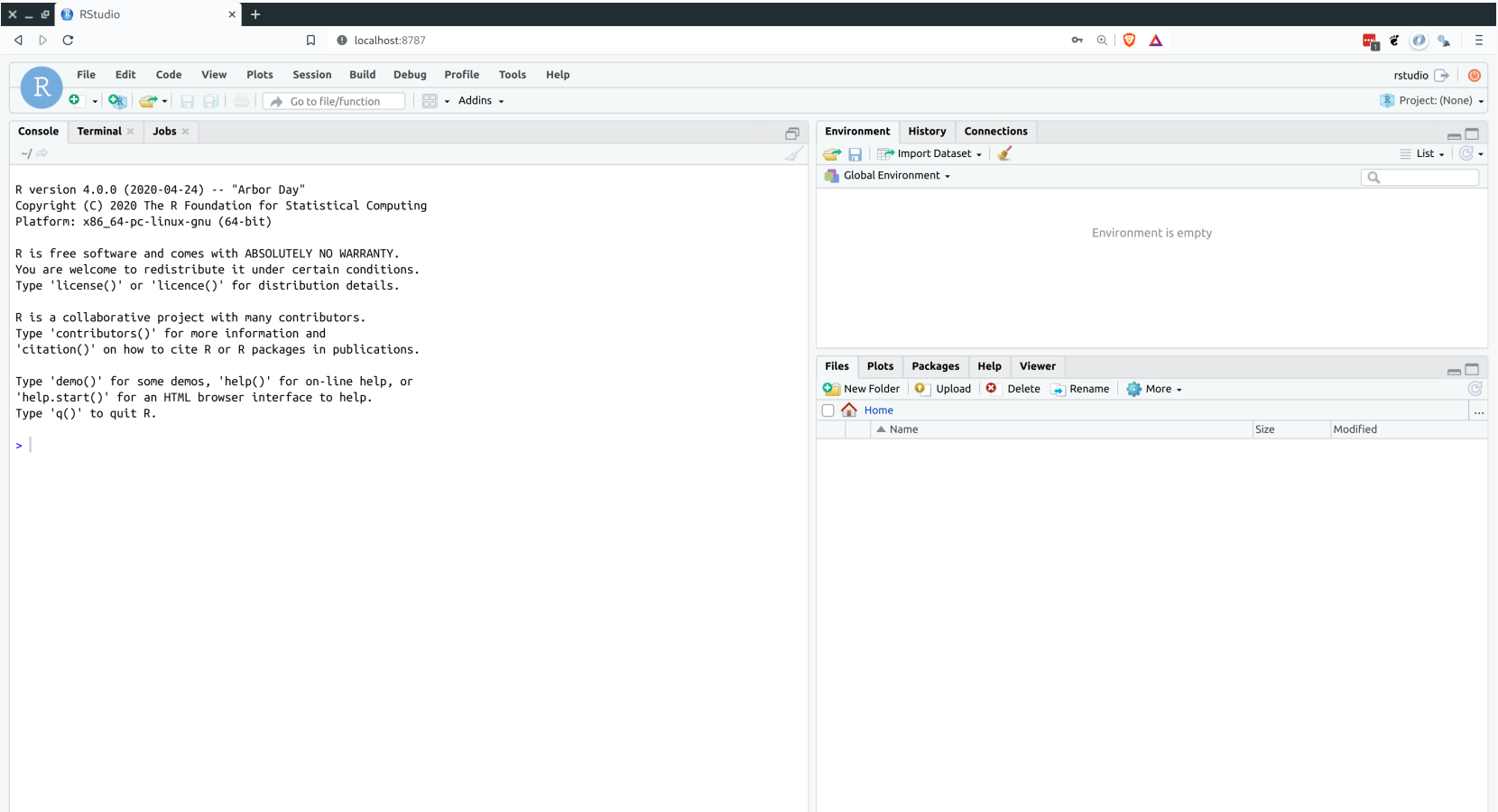
RStudio+ container (cont.)

Here's the login-in screen that I see when I navigate my browser to the relevant URL.



RStudio+ container (cont.)

And here I am in RStudio Server running through Docker! (Pro-tip: Hit F11 to go full-screen.)



RStudio+ container (cont.)

To stop this container, open up a new terminal window and grab the container ID with `$ docker ps` (it will be the first column). Then run:

```
$ docker stop <containerid>
```

Note: We actually ran this container as a background process, because we used the `-d` flag when we first instantiated it, i.e.

```
$ docker run -d -p 8787:8787 -e PASSWORD=mypassword rocker/tidyverse
```

If you dropped the `-d` flag and re-ran the above command, your terminal would stay open as an ongoing process. (Try this yourself.)

- Everything else would remain the same. You'd still navigate to `<IPADDRESS>:8787` to log in, etc.
- However, I wanted to mention this non-background process version because it offers another way to shut down the container: Simply type `CTRL+C` in the (same, ongoing process) Terminal window. Again, try this yourself.
- Confirm that the container is stopped by running `$ docker ps`.

Sharing files with a container

Share files by mounting volumes

Each container runs in a sandboxed environment and cannot access the files and directories on your computer unless you give it explicit permission.

To share files with a container, the `-v` (mount volume) flag is your friend.

- Adopts a `LHS:RHS` convention, where `LHS` = `path/on/your/computer/` and `RHS` = `path/on/the/container`.

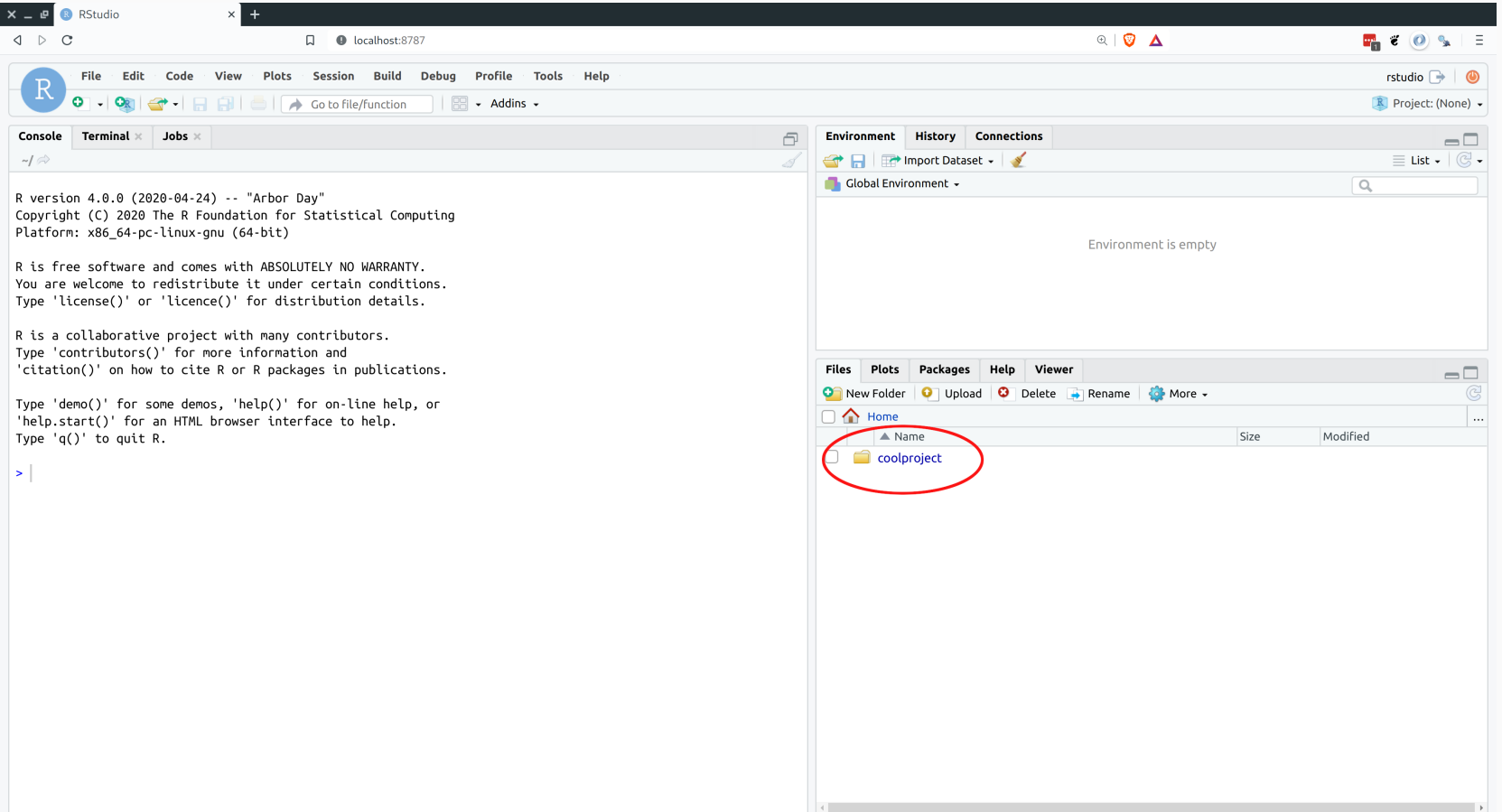
For example, say I have a folder on my computer located at `/home/grant/coolproject`. I can make this available to my "tidyverse" container by running:

```
$ docker run -v /home/grant/coolproject:/home/rstudio/coolproject -d -p 8787:8787
```

PS — I'll get back to specifying the correct RHS path in a couple slides.

coolproject

The coolproject directory is now available from RStudio running on the container.



Choosing the RHS mount point

In the previous example, I specified the RHS mount point as `/home/rstudio/coolproject`.

How did I know this would work?

The short answer is that `/home/rstudio` is the default user's home directory for images in the RStudio+ stack. If you're running a container from this stack, you should almost always start your RHS with this path root.

- Exception: If you assigned a different default user than "rstudio" ([back here](#)).

OTOH, the `/coolproject` directory name is entirely optional. You could call it anything you want. Giving it the same name as the directory on your computer obviously helps to avoid confusion, though.

- Similarly, nothing is stopping you from adding a couple of parent directories. I could have used `-v /home/grant/coolproject:/home/rstudio/parentdir/coolproject` and it would have worked fine.

Choosing the RHS mount point (cont.)

So, the RHS mount point for RStudio+ containers is (almost) always `/home/rstudio`.

What about other containers?

- E.g. For r-base containers there's no "rstudio" user so the above definitely won't work. (Fun fact: When you run an r-base container you are actually logged in as root.)

In truth, the specific RHS mount point is less important for non-RStudio+ containers.¹ Still, I recommend a general strategy of mounting external volumes on the dedicated `/mnt` directory that is standard on Linux.² For example:

```
$ docker run -it --rm -v /home/grant/coolproject:/mnt/coolproject r-base
```

¹ It only really matters for the first group because RStudio Server limits how and where users can access files. This is a security feature we'll revisit in the next lecture on cloud computing.

² Remember, Docker images are basically just miniaturized, portable Linux OSs.

Further reading

- [Rocker Wiki](#)
- [ROpenSci Docker Tutorial](#)

Next class: Cloud computing with Google Compute Engine
