Docker Images

Business Science

11/26/2019

Contents

Docker Images

This document covers the using `docker image` to create custom images that you can store on Docker Hub and install on your EC2 Server. By the end of this, you will be able to build your own custom docker images using a DockerFile.

Prerequisites

Use `docker container ls` and visit the EC2 Server's port 8787 to confirm that RStudio server is running with linked volumes to the `/home/ubuntu/rstudio_docker`.

If a container is NOT running RStudio, use this command to setup the RStudio Server with linked volumes using Docker:

```
sudo docker run -e PASSWORD=your_password -d -p 8787:8787 \
-v /home/ubuntu/rstudio_docker:/home/rstudio/rstudio_docker rocker/tidyverse
```

Definitions

- **Dockerfile** - A text document that contains all the commands a user could call on the command line to assemble an image.
- **Building an Image** - Using `docker build`, users can create an automated build that executes several command-line instructions in succession that are recorded in a Dockerfile.

Example - Adding shinyWidgets to an Image

Making a Dockerfile

Make the following text file in your RStudio IDE.

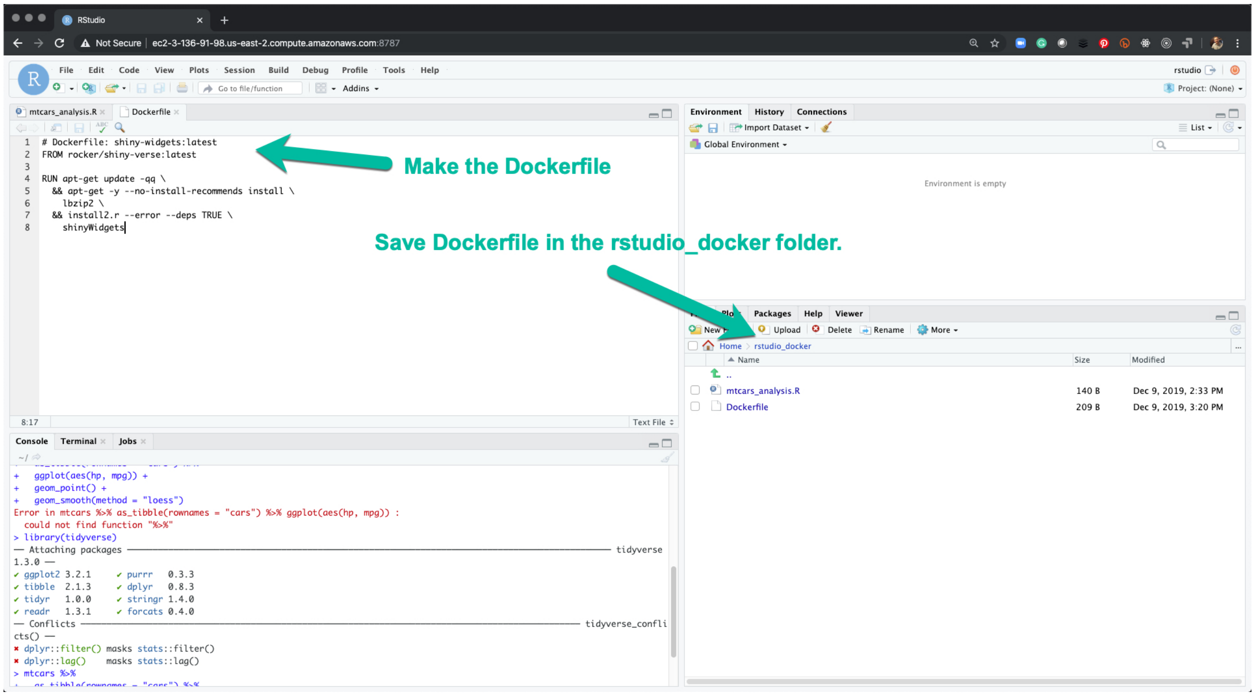
```
# File: Dockerfile
FROM rocker/shiny-verse:latest

RUN apt-get update -qq \
  && apt-get -y --no-install-recommends install \
    lbzip2 \
  && install2.r --error --deps TRUE \
    shinyWidgets
```

Here's what is happening in the Dockerfile:

- `FROM rocker/shiny-verse:latest` - Uses `shiny-verse` as the starting point for the image.
- `RUN apt-get update -qq` - Updates the installation software
- `&& apt-get -y --no-install-recommends install` - Installs Linux libraries that R Packages depend on (e.g. `lbzip2` is a linux library)
- `&& install2.r --error --deps TRUE` - Installs R libaries from CRAN. This is where we list the R packages we want.

Saving a Dockerfile

Save the file as `rstudio_docker/Dockerfile`.



```
# Dockerfile: shiny-widgets:latest
FROM rocker/shiny-verse:latest

RUN apt-get update -qq \
  && apt-get -y --no-install-recommends install \
    lbzip2 \
  && install2.r --error --deps TRUE \
    shinyWidgets
```

Make the Dockerfile

Save Dockerfile in the rstudio_docker folder.

Home > rstudio_docker

Name	Size	Modified
..		
mtcars_analysis.R	140 B	Dec 9, 2019, 2:33 PM
Dockerfile	209 B	Dec 9, 2019, 3:20 PM

```
+   ggplot(aes(hp, mpg)) +
+   geom_point() +
+   geom_smooth(method = "loess")
Error in mtcars %>% as_tibble(rownames = "cars") %>% ggplot(aes(hp, mpg)) :
  could not find function "%>%"
> library(tidyverse)
── Attaching packages ─────────────────────────────────────────────── tidyverse
1.3.0 ──
✔ ggplot2 3.2.1     ✔ purrr   0.3.3
✔ tibble  2.1.3     ✔ dplyr   0.8.3
✔ tidyr   1.0.0     ✔ stringr 1.4.0
✔ readr   1.3.1     ✔ forcats 0.4.0
── Conflicts ──────────────────────────────────────────────────── tidyverse_confli
cts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
> mtcars %>%
+   as_tibble(rownames = "cars") %>%
```

Docker Build

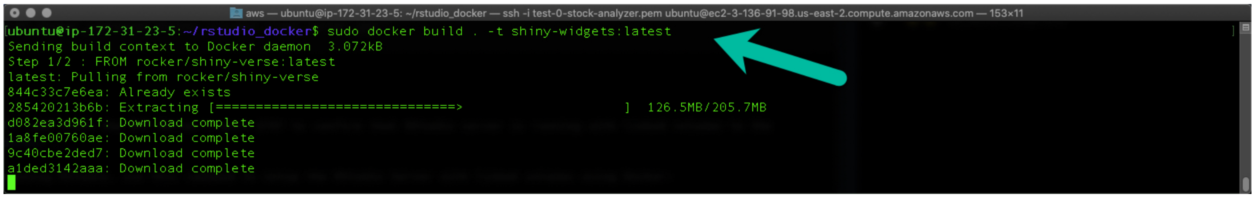
The `docker build` command is how to build docker images that extend current images using software that you need for your applications to run.

Navigate into the `rstudio_docker` folder. Run the following command:

```
sudo docker build . -t shiny-widgets:latest
```

Here's what is happening in the Dockerfile:

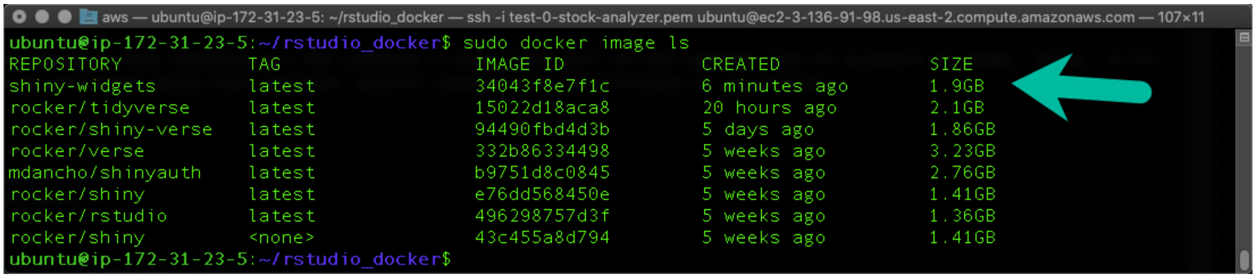
- `docker build` - Used to build an image from a Dockerfile (can also use `Docker image build`)
- `.` - The directory that contains the Dockerfile named "Dockerfile". If your dockerfile is located somewhere else or named differently than Dockerfile, use tab completion to locate it (i.e. `docker build path/to/your/file/dockerfile_name`)
- `-t` - Adds an image name and tag.
  - name - The name of the image. The name we chose is `shiny-widgets`.
  - tag - The version of the image. `latest` is used by default.


aws — ubuntu@ip-172-31-23-5: ~/rstudio_docker — ssh -i test-0-stock-analyzer.pem ubuntu@ec2-3-136-91-98.us-east-2.compute.amazonaws.com — 153×11

```
ubuntu@ip-172-31-23-5:~/rstudio_docker$ sudo docker build . -t shiny-widgets:latest
Sending build context to Docker daemon  3.072kB
Step 1/2 : FROM rocker/shiny-verse:latest
latest: Pulling from rocker/shiny-verse
844c33c7e6ea: Already exists
285420213b6b: Extracting [==============================>                    ]  126.5MB/205.7MB
d082ea3d961f: Download complete
1a8fe00760ae: Download complete
9c40cbe2ded7: Download complete
a1ded3142aaa: Download complete
```

Docker Image List (Local)

The image is now stored locally on our EC2 Server. We can verify this using `sudo docker image ls` to list the images. We can now use this image to build containers that


aws — ubuntu@ip-172-31-23-5: ~/rstudio_docker — ssh -i test-0-stock-analyzer.pem ubuntu@ec2-3-136-91-98.us-east-2.compute.amazonaws.com — 107×11

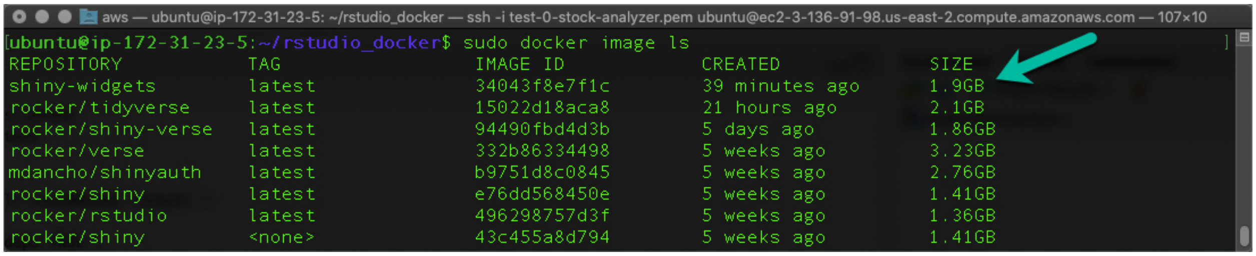
```
ubuntu@ip-172-31-23-5:~/rstudio_docker$ sudo docker image ls
REPOSITORY           TAG        IMAGE ID       CREATED          SIZE
shiny-widgets        latest     34043f8e7f1c   6 minutes ago    1.9GB
rocker/tidyverse     latest     15022d18aca8   20 hours ago     2.1GB
rocker/shiny-verse   latest     94490fbd4d3b   5 days ago       1.86GB
rocker/verse         latest     332b86334498   5 weeks ago      3.23GB
mdancho/shinyauth    latest     b9751d8c0845   5 weeks ago      2.76GB
rocker/shiny         latest     e76dd568450e   5 weeks ago      1.41GB
rocker/rstudio       latest     496298757d3f   5 weeks ago      1.36GB
rocker/shiny         <none>     43c455a8d794   5 weeks ago      1.41GB
ubuntu@ip-172-31-23-5:~/rstudio_docker$
```

Docker Hub

If we'd like to be able to share the image, we can push the image to **Docker Hub.**

List the Images

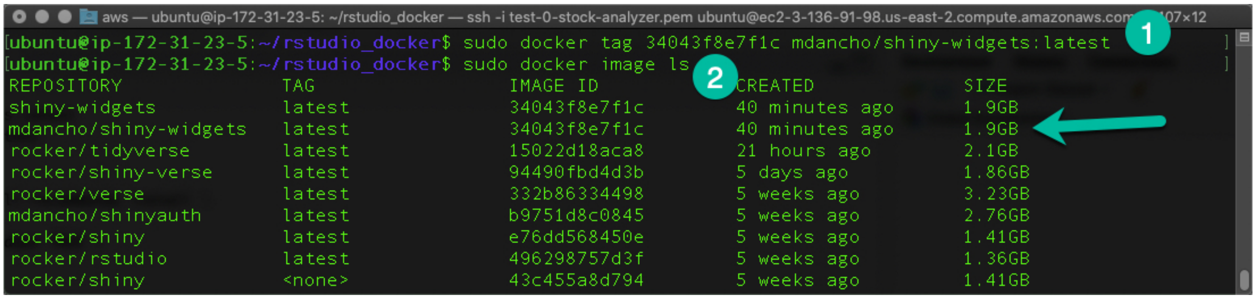
```
sudo docker image ls
```


aws — ubuntu@ip-172-31-23-5: ~/rstudio_docker — ssh -i test-0-stock-analyzer.pem ubuntu@ec2-3-136-91-98.us-east-2.compute.amazonaws.com — 107×10

```
ubuntu@ip-172-31-23-5:~/rstudio_docker$ sudo docker image ls
REPOSITORY           TAG        IMAGE ID       CREATED          SIZE
shiny-widgets        latest     34043f8e7f1c   39 minutes ago   1.9GB
rocker/tidyverse     latest     15022d18aca8   21 hours ago     2.1GB
rocker/shiny-verse   latest     94490fbd4d3b   5 days ago       1.86GB
rocker/verse         latest     332b86334498   5 weeks ago      3.23GB
mdancho/shinyauth    latest     b9751d8c0845   5 weeks ago      2.76GB
rocker/shiny         latest     e76dd568450e   5 weeks ago      1.41GB
rocker/rstudio       latest     496298757d3f   5 weeks ago      1.36GB
rocker/shiny         <none>     43c455a8d794   5 weeks ago      1.41GB
```

Prep for upload with tag command

1. `sudo docker tag [IMAGE ID] your_dockerhub_user_name/shiny-widgets:latest` - Replace the Image ID with the correct Image ID and your docker hub User Name.
2. `sudo docker image ls` - Make sure the repository now matches your user name.


aws — ubuntu@ip-172-31-23-5: ~/rstudio_docker — ssh -i test-0-stock-analyzer.pem ubuntu@ec2-3-136-91-98.us-east-2.compute.amazonaws.com — 107×12
1
2

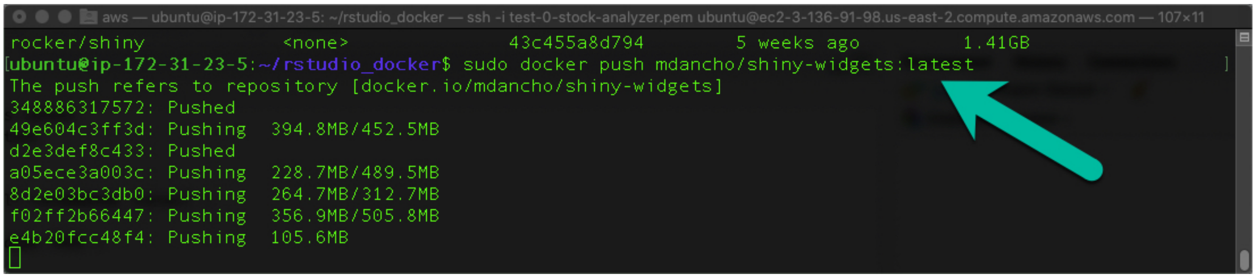
```
[ubuntu@ip-172-31-23-5:~/rstudio_docker$ sudo docker tag 34043f8e7f1c mdancho/shiny-widgets:latest
[ubuntu@ip-172-31-23-5:~/rstudio_docker$ sudo docker image ls
REPOSITORY              TAG                 IMAGE ID            CREATED             SIZE
shiny-widgets           latest              34043f8e7f1c        40 minutes ago      1.9GB
mdancho/shiny-widgets   latest              34043f8e7f1c        40 minutes ago      1.9GB
rocker/tidyverse        latest              15022d18aca8        21 hours ago        2.1GB
rocker/shiny-verse      latest              94490fbd4d3b        5 days ago          1.86GB
rocker/verse            latest              332b86334498        5 weeks ago         3.23GB
mdancho/shinyauth       latest              b9751d8c0845        5 weeks ago         2.76GB
rocker/shiny            latest              e76dd568450e        5 weeks ago         1.41GB
rocker/rstudio          latest              496298757d3f        5 weeks ago         1.36GB
rocker/shiny            <none>              43c455a8d794        5 weeks ago         1.41GB
```

Login to Docker Hub

`sudo docker login` - Provide your Docker Hub User Name and Password.

Push to Docker Hub

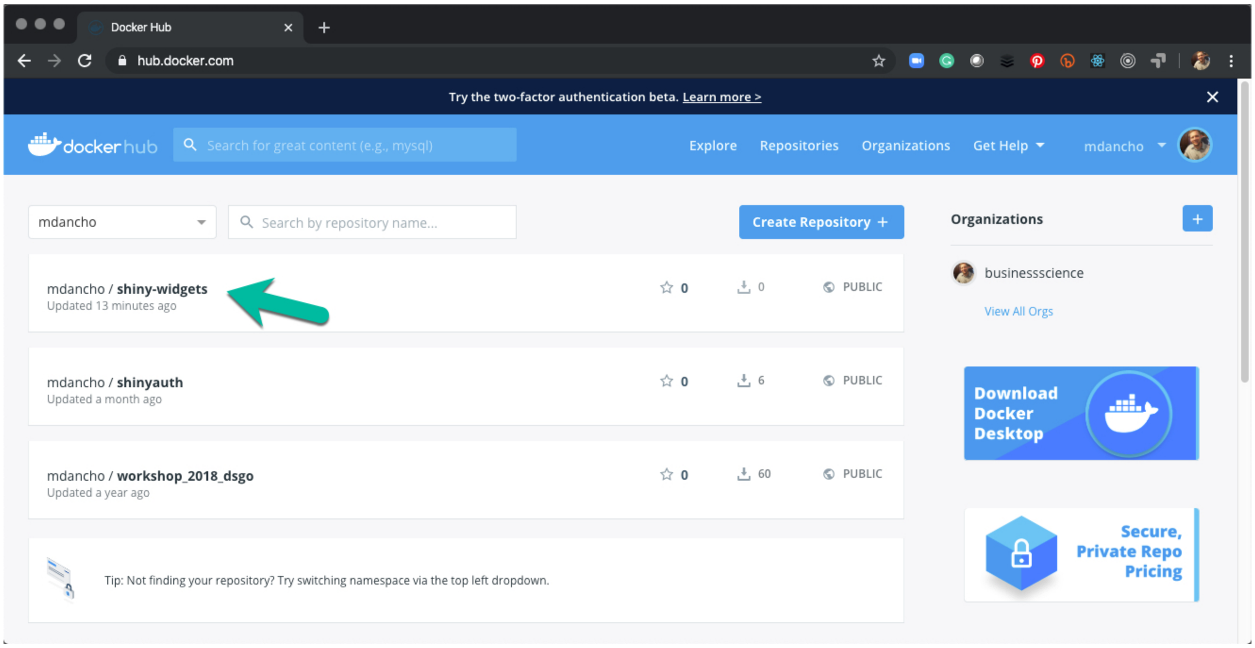
`sudo docker push your_docker_hub_user_name/shiny-widgets:latest` - Pushes the Dockerfile to DockerHub.


aws — ubuntu@ip-172-31-23-5: ~/rstudio_docker — ssh -i test-0-stock-analyzer.pem ubuntu@ec2-3-136-91-98.us-east-2.compute.amazonaws.com — 107×11

```
rocker/shiny            <none>              43c455a8d794        5 weeks ago         1.41GB
[ubuntu@ip-172-31-23-5:~/rstudio_docker$ sudo docker push mdancho/shiny-widgets:latest
The push refers to repository [docker.io/mdancho/shiny-widgets]
348886317572: Pushed
49e604c3ff3d: Pushing  394.8MB/452.5MB
d2e3def8c433: Pushed
a05ece3a003c: Pushing  228.7MB/489.5MB
8d2e03bc3db0: Pushing  264.7MB/312.7MB
f02ff2b66447: Pushing  356.9MB/505.8MB
e4b20fcc48f4: Pushing  105.6MB
```

Check Docker Hub

Congrats. Your image is now available on Docker Hub. You can share it with others or use it on other EC2 Servers that you create.



mdancho / shiny-widgets
Updated 13 minutes ago
☆ 0 0 PUBLIC

mdancho / shinyauth
Updated a month ago
☆ 0 6 PUBLIC

mdancho / workshop_2018_dsgo
Updated a year ago
☆ 0 60 PUBLIC

Tip: Not finding your repository? Try switching namespace via the top left dropdown.

shinyauth - Complex Dockerfile & Image for Stock Analyzer

The `shiny-widgets` image that we created is much too simplistic for our Stock Analyzer application. In addition to `shinywidgets`, we need:

- CRAN Libraries: `shinythemes`, `shinyjs`, `mongolite`, `jsonlite`, `config`, `remotes`, `tidyquant`, and `plotly`.
- GitHub Libraries: `business-science/shinyauthr`

To make this image, I created `mdancho/shinyauth` - A Docker Image that contains the necessary R Packages and Linux Dependencies that we need.

Dockerfile

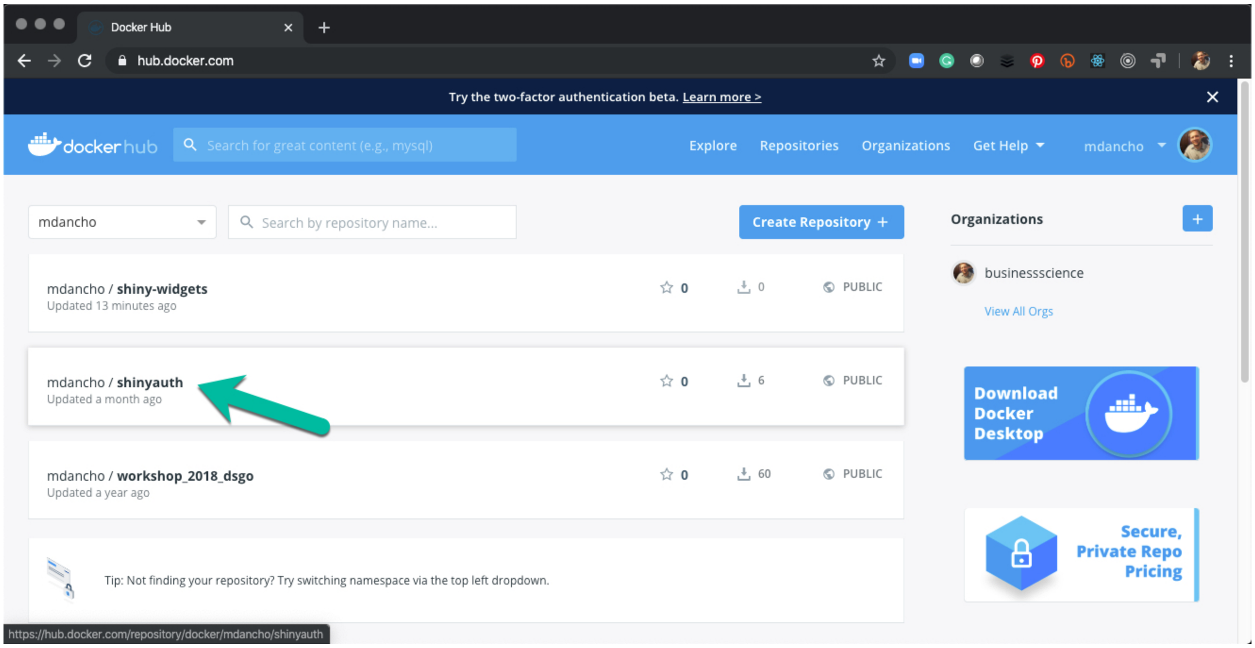
The Dockerfile looks like this:

```
FROM rocker/shiny-verse:latest

RUN apt-get update -qq \
    && apt-get -y --no-install-recommends install \
        lbzip2 \
        libfftw3-dev \
        libgdal-dev \
        libgeos-dev \
        libgsl0-dev \
        libgl1-mesa-dev \
        libglu1-mesa-dev \
        libhdf4-alt-dev \
        libhdf5-dev \
        libjq-dev \
        liblwgeom-dev \
        libpq-dev \
        libproj-dev \
        libprotobuf-dev \
        libnetcdf-dev \
        libsqlite3-dev \
        libssl-dev \
        libudunits2-dev \
        netcdf-bin \
        postgis \
        protobuf-compiler \
        sqlite3 \
        tk-dev \
        unixodbc-dev \
        libsasl2-dev \
        libv8-dev \
        libsodium-dev \
    && install2.r --error --deps TRUE \
        shinyWidgets \
        shinythemes \
        shinyjs \
        mongolite \
        jsonlite \
        config \
        remotes \
        tidyquant \
        plotly \
    && installGithub.r business-science/shinyauthr
```

Docker Hub Image

You can find the `shinyauth` image several ways.

Docker Hub



Try the two-factor authentication beta. Learn more >

mdancho / **shiny-widgets** ☆ 0 ⤓ 0 PUBLIC
Updated 13 minutes ago

mdancho / **shinyauth** ☆ 0 ⤓ 6 PUBLIC
Updated a month ago

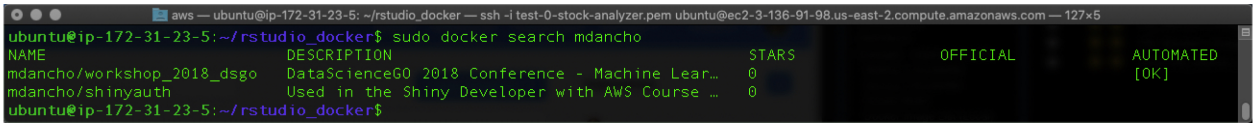
mdancho / **workshop_2018_dsgo** ☆ 0 ⤓ 60 PUBLIC
Updated a year ago

Tip: Not finding your repository? Try switching namespace via the top left dropdown.

https://hub.docker.com/repository/docker/mdancho/shinyauth

Docker Search

`sudo docker search mdancho` - Searches for any public images that "mdancho" has pushed to Docker Hub.


aws — ubuntu@ip-172-31-23-5: ~/rstudio_docker — ssh -i test-0-stock-analyzer.pem ubuntu@ec2-3-136-91-98.us-east-2.compute.amazonaws.com — 127×5

```
ubuntu@ip-172-31-23-5:~/rstudio_docker$ sudo docker search mdancho
NAME                         DESCRIPTION                                     STARS               OFFICIAL            AUTOMATED
mdancho/workshop_2018_dsgo   DataScienceGO 2018 Conference - Machine Lear…   0                                       [OK]
mdancho/shinyauth            Used in the Shiny Developer with AWS Course …   0
ubuntu@ip-172-31-23-5:~/rstudio_docker$
```

Docker pull to install shinyauth

`sudo docker pull mdancho/shinyauth` - Will `pull` the `shinyauth:latest` image onto your EC2 Server.

Docker image list

`sudo docker image ls` - List the images. Verify that you've installed the `shinyauth` image.

Wrapup

1. You now are able to `build` your own docker images. Simply follow the Shiny Widgets Example.
2. You know the basic structure of a `Dockerfile`. Simply modify as necessary to build more complex docker images.