Cloning Remote Repo

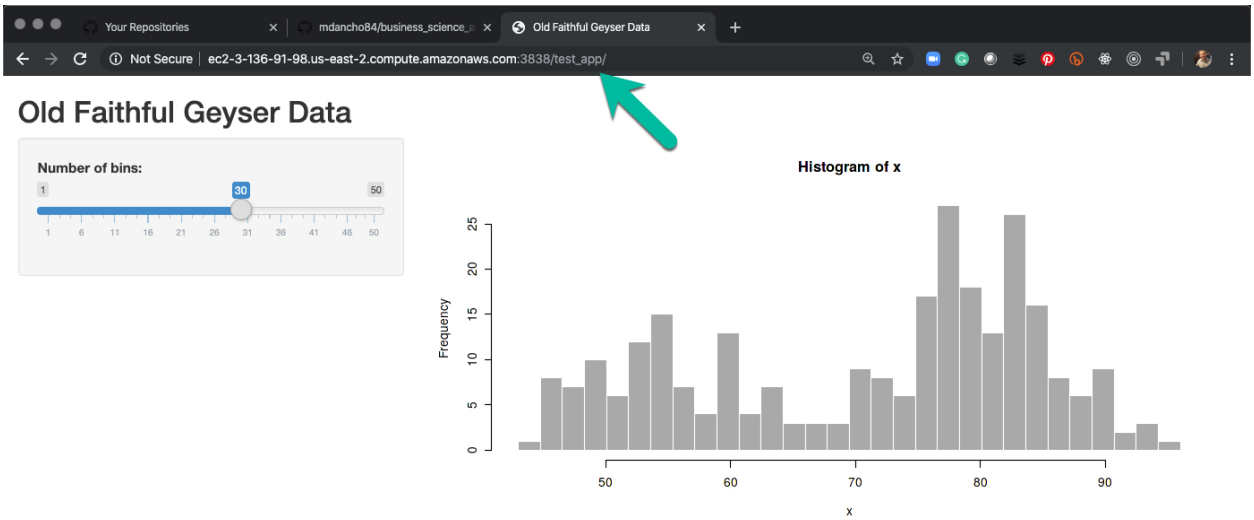
Business Science

11/26/2019

Contents

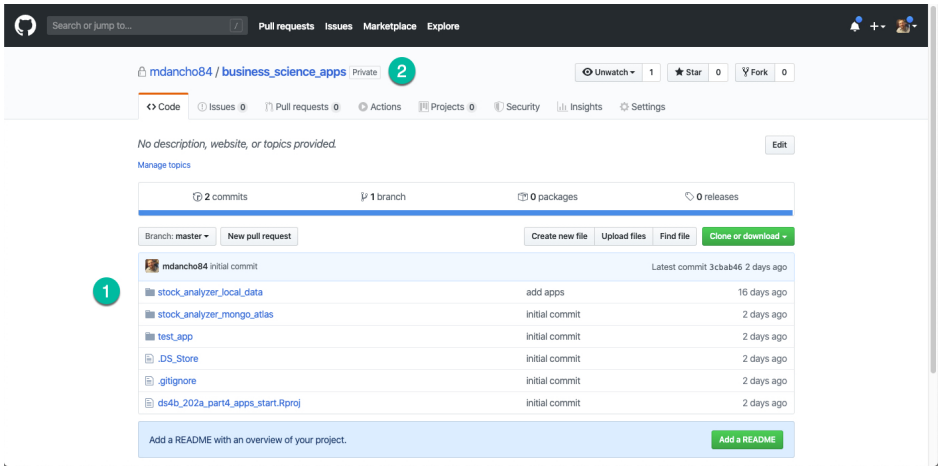
Cloning Remote Repo on EC2 Server

This document covers getting the `shiny` app files onto your AWS EC2 Server. By the end of this document, you will have used `git clone` to clone your files and used `docker run` to run a Test Application publically on you EC2 Server.



Prerequisites

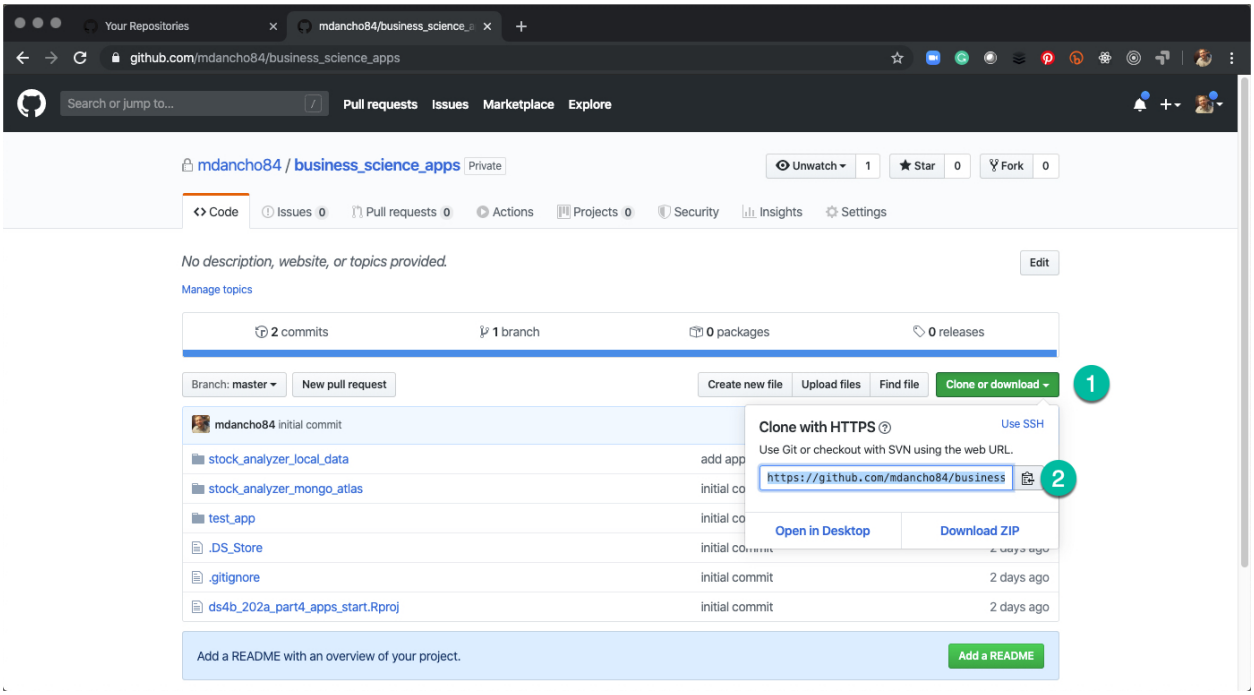
You must have a remote repository on GitHub to clone. This repository contains the 3 shiny app directories with supporting files.



GitHub Clone URL

On you GitHub Repo, get the URL string for cloning the repo.

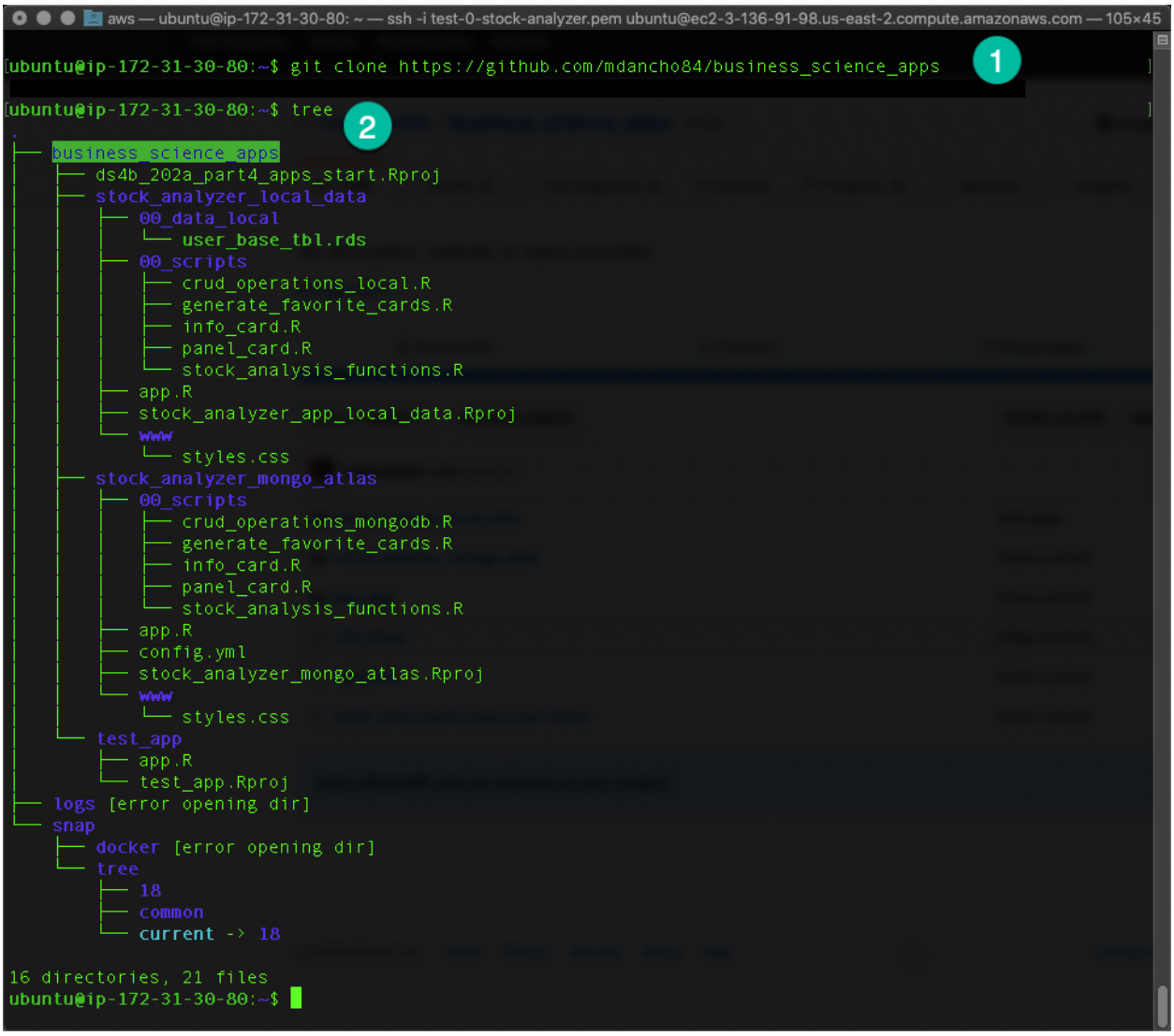
1. Click Clone or Download
2. Copy the URL for your GitHub Repo



Clone the Directory on EC2 Server

This creates a local copy that is linked to your remote.

1. Use `git clone your_remote_repo_url`
2. Use `tree` to verify a folder has been added with contents of your github repo.



Test an App

Finally, we'll test that we can get an app running in development mode combining `docker` and the `/test_app`.

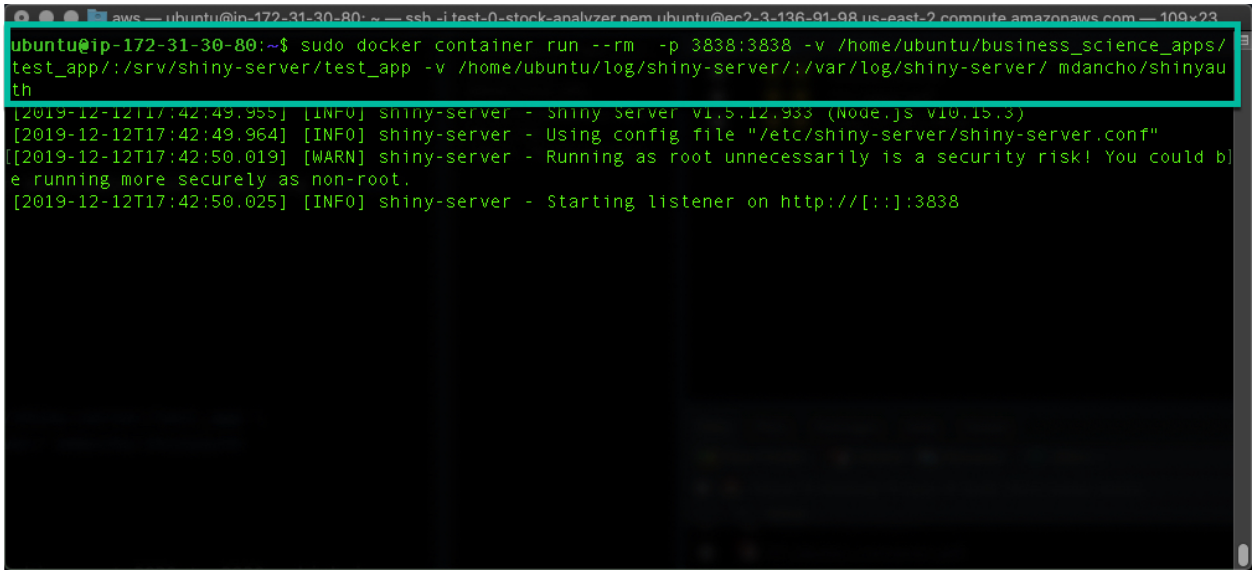
Step 1 - Run Docker with the Test App

Run the following command (modify as necessary to fit your ubuntu directory setup):

```
sudo docker container run --rm  -p 3838:3838 \
-v /home/ubuntu/business_science_apps/test_app/:/srv/shiny-server/test_app \
-v /home/ubuntu/log/shiny-server/:/var/log/shiny-server/ \
mdancho/shinyauth
```

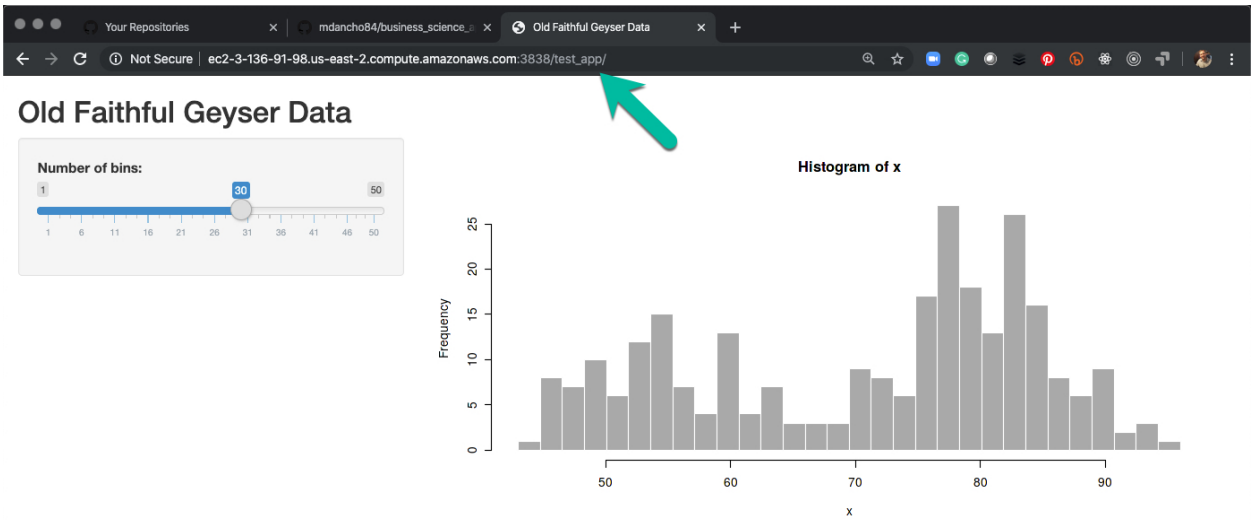
What's happening?

- `sudo docker container run --rm  -p 3838:3838` - We are setting up a development docker container linking port 3838 (EC2) to 3838 (Container), which is what Shiny Server runs on.
- There are two volumes that need to be linked for shiny apps:
    1. **App Files** - The Test App files are being linked to a volume inside the container at `/srv/shiny-server/test_app/`
    2. **Log Files** - These track if anything goes wrong. We link a directory at `/home/ubuntu/log/shiny-server/` to the container at `/var/log/shiny-server/`
- We use `mdancho/shinyauth` image to load the R libraries and shiny server inside the container so the Test App can run



Step 2 - Navigate to the EC2 Server port to see your app in action

You should now see an app at the URL of your EC2 Server by appending `:3838/test_app` to the URL.



Step 3 - Shutdown the Test App

Use `Ctrl+C` to shutdown the Test App.

Wrapup

Congratulations. You've just launched and shutdown your first Shiny App on AWS EC2. You now know how to:

1. Clone your apps using `git clone`
2. Leverage `docker` and your Shiny App files to stand-up a simple shiny app

We'll start adding more complexity once we get into Shiny App Deployment, but next we will cover making app changes.