**Building a Docker image to host an RShiny app on BlueBRIDGE**

*Online accounts required:*

Github: <https://github.com/>

DockerHub: <https://hub.docker.com/>

*Step 1: Basic shiny app.*

Develop your shiny app and host in a Github repository. The structure of the repository should be an RShiny application file either as a single file app.R or paired files server.R and ui.R and folders to contain any additional files or data used by your app (Figure 1).

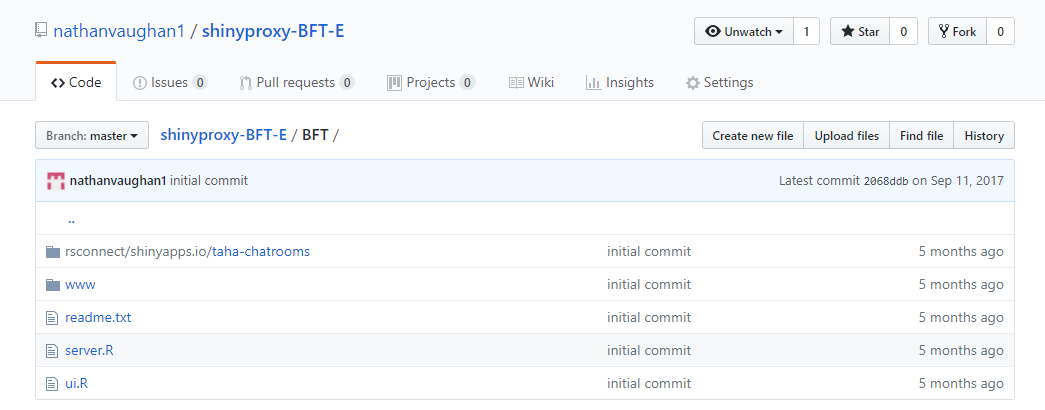


Figure 1: Example of a RShiny app Github repository.

Step 2: *Docker wrapper*.

Wrap all app files/folders in a single folder in this example named BFT then add a plain text Rprofile.site file to set the visible shiny port and host details and a plain text file called Dockerfile which will contain the instructions for building a Docker container.

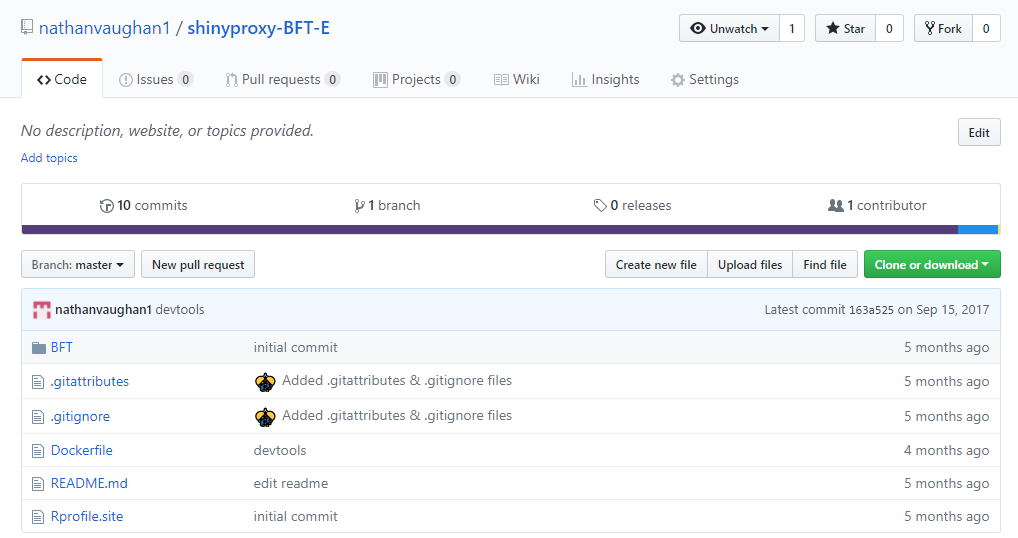
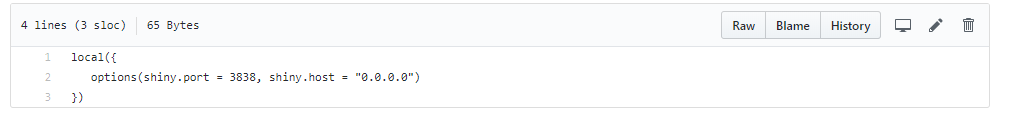


Figure 2: Folder structure for Docker ready RShiny app.

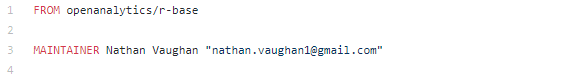
Step 3: Rprofile.site

This is a simple file that sets the shiny port and host values

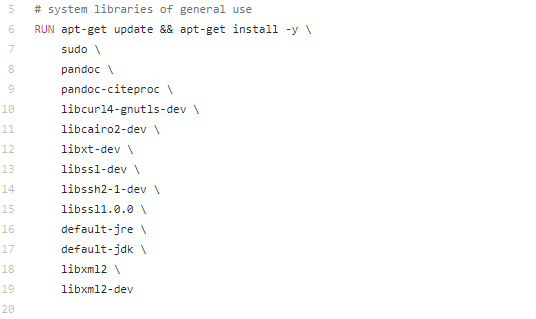


Step 3: Dockerfile.

The Dockerfile required to build a shiny app consists of several distinct sections: 1) Declaring a base image from which to build upon a simple choice is “openanalytics/r-base”.



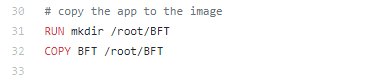
2) Installing important Linux system libraries.



3) Installing required R packages such as ‘shiny’ and any dependencies for your specific app.



4) Copy the shiny app to the root directory of the linux image.



5) Copy the Rprofile.sit file and expose correct port (3838).



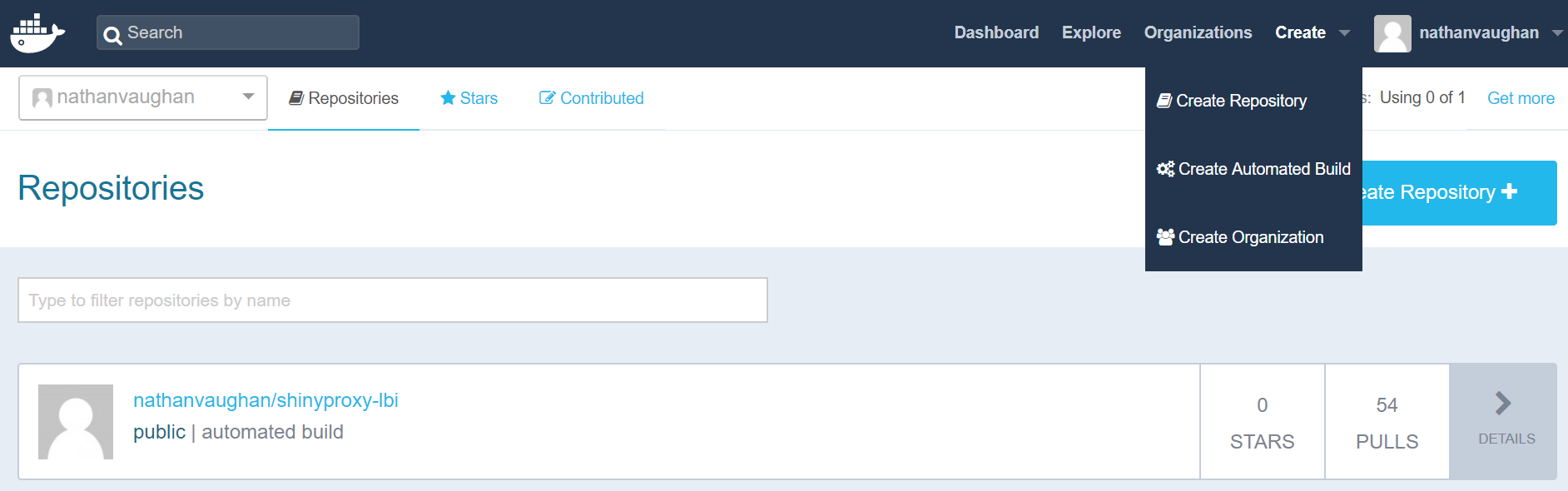
6) Run R application



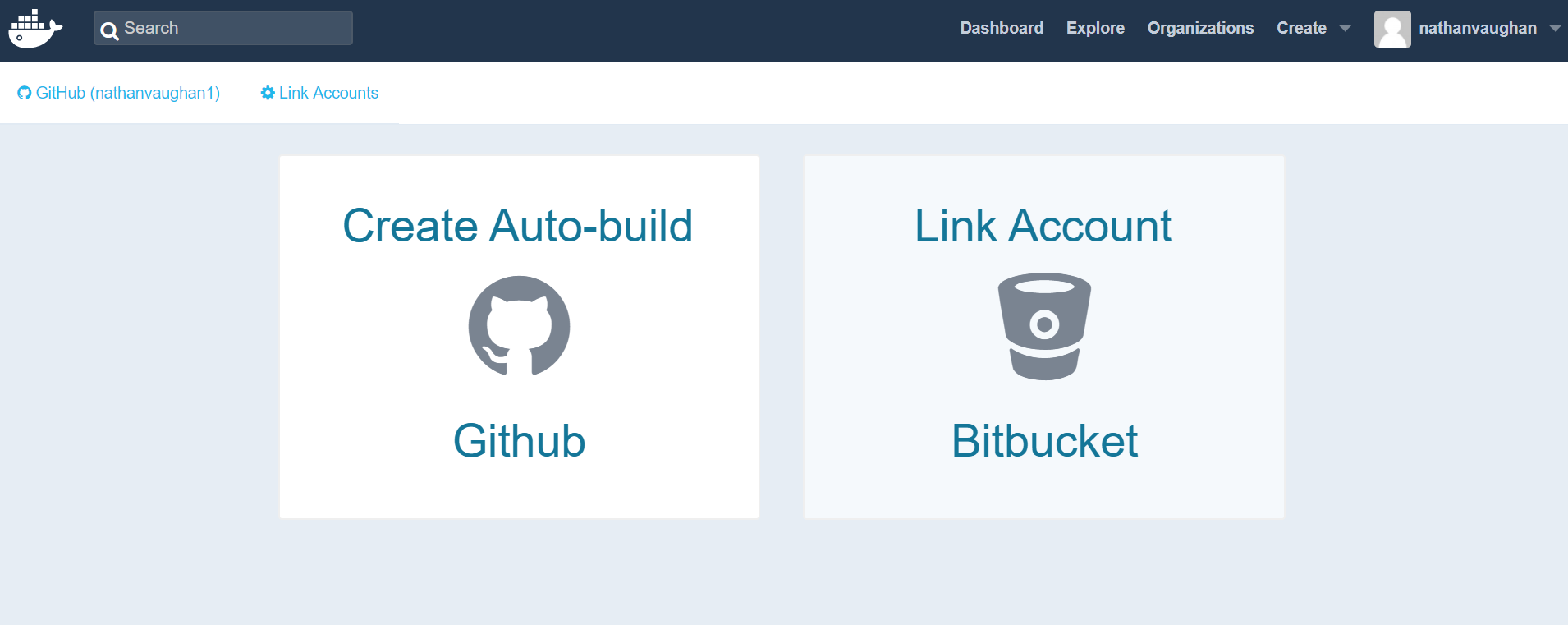
Step 4: Create DockerHub autobuild package.

1) Go to <https://hub.docker.com/> and sign in or sign up for an account.

2) From the create dropdown menu select create automated build.

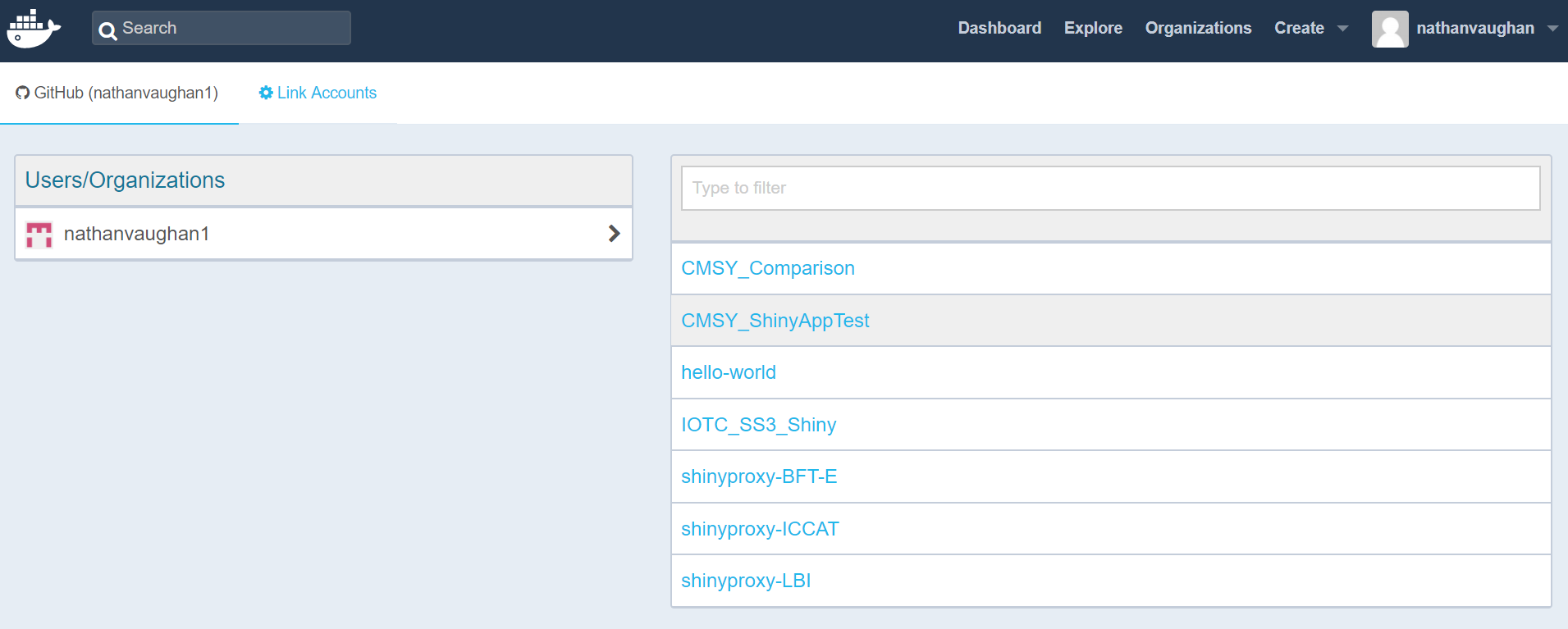


3) On the next page select Create Auto-Build Github.

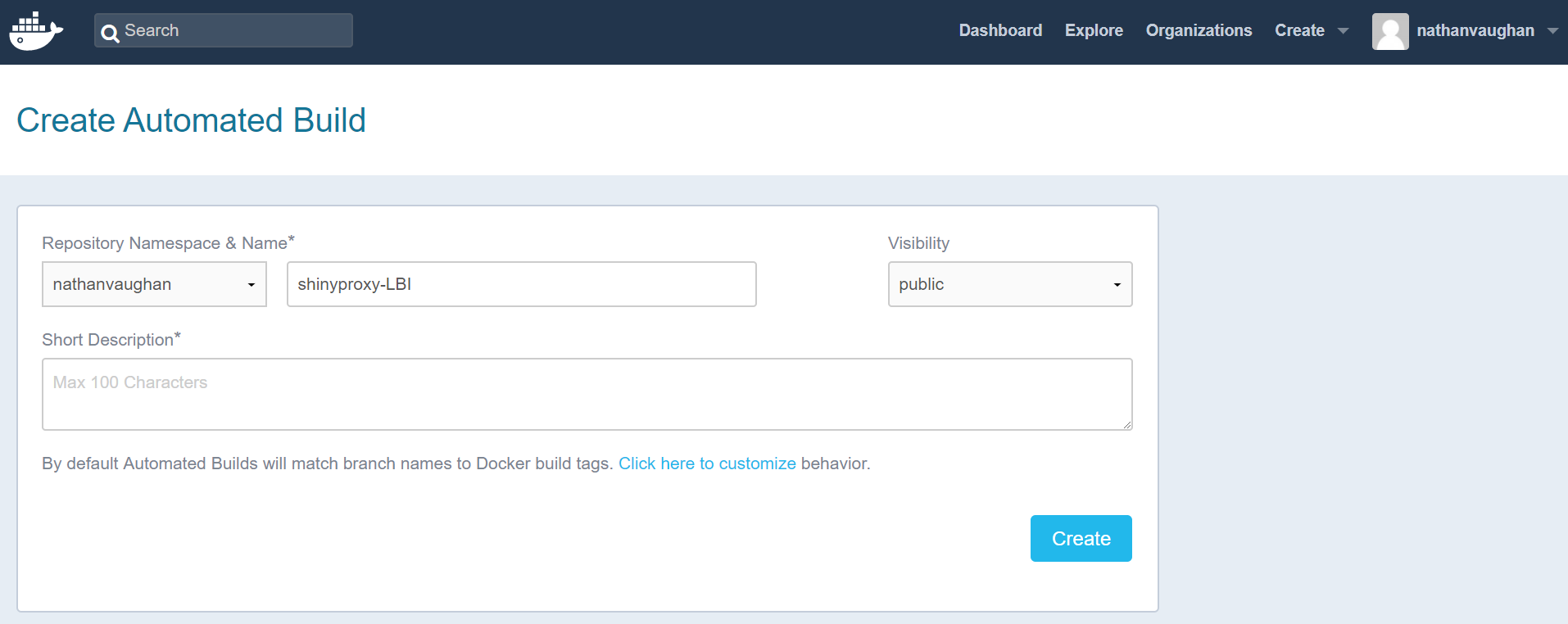


4) If your github account is not already linked to DockerHub select link new account.

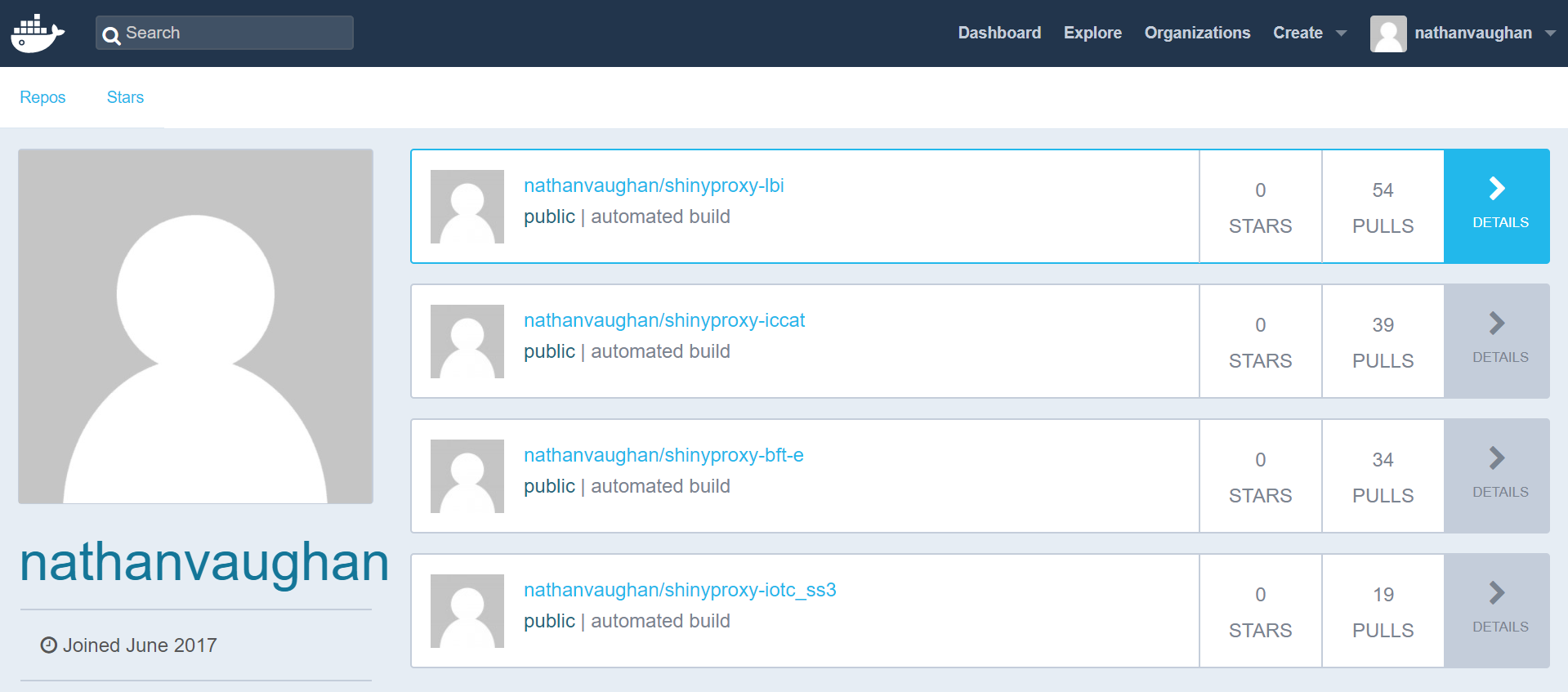
5) Select the Github package with dockerfile that you created in steps 1-3.



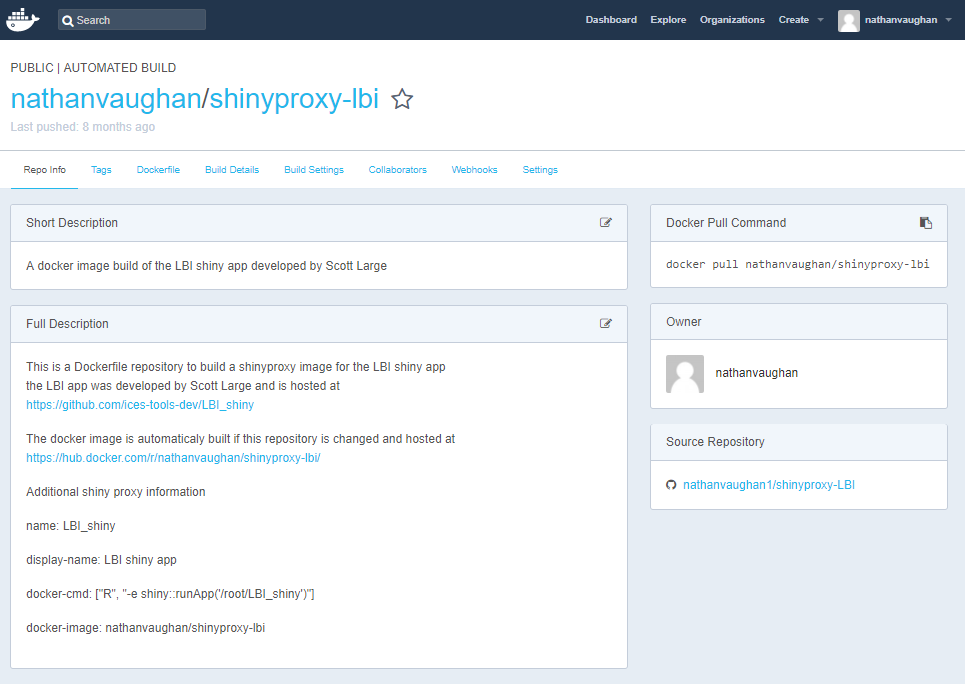
6) Add short description of app and select create.



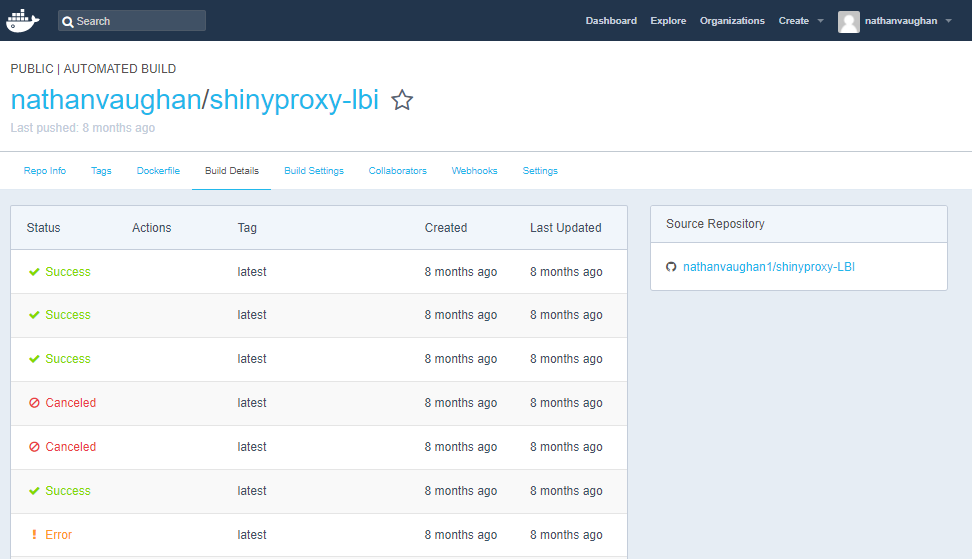
Step 5: Now go to your profile and find your package then click details.



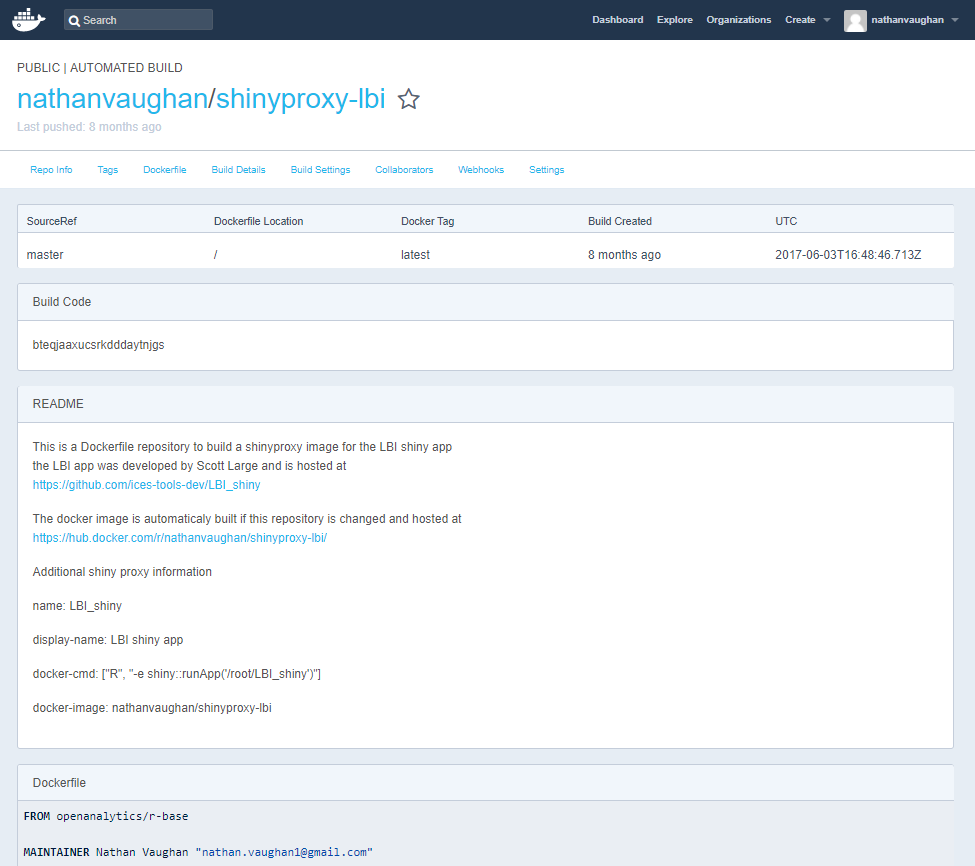
Step 6: Your main package page will include all the important details for your package including its pull command, description, and Github source package.



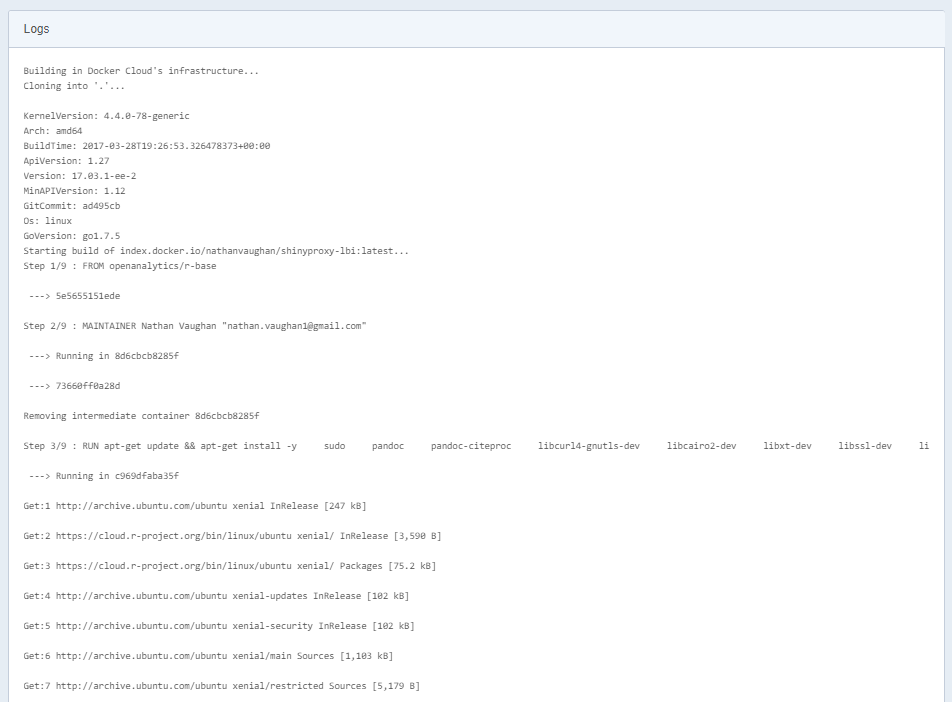
Step 7: Now go to Build Details to see the status of your new docker build. For around the first hour after creating a build or making a change to your Github source package the status will be displays as building or queued. After this it will show as either success or error.



Step 8: Even if the build shows success there may still be errors in the shiny app due to issues such as missing R dependencies. To check for this or to debug a status of error, click on the success or error status label for your latest build. This will display all details for that build.



Step 9: Scroll down on this page until you get to the Log panel.



Step 10: This section will be very long but will describe all the step by step details of the linux build, R installation, package loading, and shiny application run. The cause of an error status or any missing packages can be identified by warnings and error messages in this log file.

Step 11: Once the Docker build is completed successfully and has been checked for missing R dependencies it can be uploaded by maintainers of the BlueBRIDGE infrastructure. To direct them in this process they require the following details to inform shinyproxy.

