Version 2

Apr 08, 2021

How to increase the number of simultaneous users allowed in an R Shiny App V.2

Sonia García-Ruiz[1]

[1]University College London, University of London

1 *Works for me* This protocol is published without a DOI.

Ryten Laboratory

Sonia García-Ruiz

ABSTRACT

There are multiple ways of hosting a Shiny App in production. When it is required to do so over a specific URL domain, RStudio Server combined with APACHE HTTP server might be one of the most widely used solutions. However, the open-source edition of RStudio Server presents a drawback: it only allows one simultaneous user browsing your web app at the same time. This protocol presents an open-source solution to allow multiple users to browse your Shiny App in an enterprise context.

vizER app was released as part of the publication Incomplete annotation has a disproportionate impact on our understanding of Mendelian and complex neurogenetic disorders

PROTOCOL CITATION

Sonia García-Ruiz 2021. How to increase the number of simultaneous users allowed in an R Shiny App. **protocols.io** https://protocols.io/view/how-to-increase-the-number-of-simultaneous-users-a-bt3vnqn6 Version created by Sonia García-Ruiz

WHAT'S NEW

A dockerized version of vizER, which can be found on: https://hub.docker.com/r/soniaruiz/vizer

KEYWORDS

ShinyProxy, Shiny App, ShinyApp, web development, .yml, Docker, Dockerfile, apache, apache server, HTTP

CREATED

Apr 08, 2021

LAST MODIFIED

Apr 08, 2021

PROTOCOL INTEGER ID

48981

MATERIALS TEXT

The code of the Shiny App vizER can be fully downloaded from GitHub on https://github.com/SoniaRuiz/vizer/tree/master/vizER.

A dockerized version of vizER can be found on: https://hub.docker.com/r/soniaruiz/vizer

BEFORE STARTING

Dependencies:

- Operating System: CentOS 7 or similar.

- Java 8 or higher.
- Docker: to install Docker on CentOS 7: https://docs.docker.com/engine/install/centos/
- ShinyProxy can be downloaded here: https://www.shinyproxy.io/downloads/
- APACHE HTTP Server installed (optional)

1 Install ShinyProxy.

ShinyProxy is a java-based open-source tool to deploy Shiny apps in a production or enterprise context https://www.shinyproxy.io/.
To install it on CentOS 7:

```
sudo yum install shinyproxy_2.3.0_x86_64.rpm
```

2 Configure ShinyProxy.

Before executing any docker image, ShinyProxy first needs to be able to connect to the Docker daemon. By default, ShinyProxy will do so on port 2375. To allow for ShinyProxy connections on port 2375 of the Docker host, the docker startup file needs to be edited.

- Edit the Docker configuration file:

```
sudo systemctl edit docker
```

- Add the following lines at the end of the config file:

```
[Service]
ExecStart=
ExecStart=/usr/bin/dockerd -H unix:// -D -H tcp://127.0.0.1:2375
```

- Restart the Docker service:

```
sudo systemctl daemon-reload
sudo systemctl restart docker
```

3 Generate a Dockerfile and build the Docker image.

This step has been previously detailed within the protocol https://www.protocols.io/view/how-to-dockerize-an-r-shiny-app-btxvnpn6.html. Please, follow the link for more info about this step.

4 Generate an *'application.yml'* file.

The application.yml file describes the docker image and how ShinyProxy should execute it. In this step, we are specifying the details of the vizER docker image generated in Step 3 (more details also here).

In addition, the *'application.yml'* file below contains the following blocks of information:
- server: this block contains info about the server environment. In this case, it only specifies the URL path in which vizER should be executed.
- proxy: details about the ShinyProxy execution.
- docker: details about the docker engine execution.
- specs: details about the docker image to be executed. This block also specifies the path of the folder that will be mounted during the image execution (this is the folder containing the data dependencies detailed here).
- logging: details about the log register.

```
server:
  servlet:
    context-path: /browser/
```

```
proxy:
  hide-navbar: true
  heartbeat-rate: 10000
  heartbeat-timeout: 60000
  container-wait-time: 100000
  authentication: none
  port: 2525

  docker:
    url: http://localhost:2375
    port-range-start: 20000
    port-range-max: 20020

  specs:
  - id: vizER
    display-name: vizER
    container-cmd: ["R", "-e shiny::runApp('/root/vizER')"]
    container-image: vizer
    container-volumes: [ "/home_2/gsit/vizER_data:/root/vizER/vizER_data" ]

logging:
  file:
    shinyproxy.log
```

5 Configuring the APACHE HTTP server.

(Please, skip this step in case you are not serving your Shiny app through the APACHE HTTP server).

To server the output of ShinyProxy execution through APACHE, we need to add the following lines into the Apache configuration file (this file is usually located on the path /etc/httpd/conf.d/):

```
<Proxy *>
Allow from localhost
</Proxy>

RedirectMatch permanent ^/browser$ /browser/
RewriteEngine on
RewriteCond %{HTTP:Upgrade} =websocket
RewriteRule /browser/(.*) ws://0.0.0.0:2525/browser/$1 [P,L]
RewriteCond %{HTTP:Upgrade} !=websocket
RewriteRule /browser/(.*) http://0.0.0.0:2525/browser/$1 [P,L]
ProxyPass /browser/ http://0.0.0.0:2525/browser/ connectiontimeout=3000 timeout=3000
ProxyPassReverse /browser/ http://0.0.0.0:2525/browser/
Header edit Location ^/ /browser/
ProxyRequests Off
```

6 Running the docker image.

To run the Docker image of vizER through ShinyProxy, type the following command from within the same folder in which the *'application.yml'* file has been placed:

```
java -jar shinyproxy-2.0.5.jar
```

The command above will start the execution of ShinyProxy. Any HTTP request received on the URL:

- http://yourserverdomain/browser/ (in case you have configured APACHE).
- http://localhost:2525/browser/ (in case you have not configured APACHE)

will redirect the ShinyProxy output as a response to the beforementioned request.