Shiny Server setup

Installation of app on dedicated server

<http://rstudio.github.io/shiny-server/latest/#configuration-settings>

<https://github.com/rstudio/shiny-server/blob/master/README.md>

# Shiny information

http://rstudio.github.io/shiny/tutorial/#scoping

# Reduce the dependencies of the app

Minimal source code and imports. But everything together in a conceice app.

# Setup R & requiered libraries/packages on server

Create script for server setup:

Libraries have to be available for root

$ sudo apt-get install r-base

sudo R -e \"install.packages('shiny', repos='http://cran.rstudio.com/')\"

Important that the libraries are available to root so that the server can use them

sudo su -

R

install.packages(‘shiny’)

install.packages(‘rmarkdown’)

library('methods')

library('plyr')

library('data.table')

library('gamlss')

library('matrixStats');

library('ggplot2')

library('libSBML')

Install the libSBML as root

Build the ‘MultiscaleAnalysis’ package and install from source.

RStudio -> Build -> More -> Build Source package

located in ~/multiscale-galactose/R/

R CMD INSTALL MultiscaleAnalysis ...

I.e. install.packages which are required

Test that all the libraries can be imported.

Install shiny server

See <http://www.rstudio.com/products/shiny/download-server/> for download links

sudo apt-get install gdebi-core

wget http://download3.rstudio.org/ubuntu-12.04/x86\_64/shiny-server-1.3.0.403-amd64.deb

sudo gdebi shiny-server-1.3.0.403-amd64.deb

Copy the app to the server location

sudo cp -R ~/MY-APP /srv/shiny-server/

**~~sudo rm -R /srv/shiny-server/gec\_app~~**

**~~sudo cp -R ~/multiscale-galactose/R/shiny/gec\_app /srv/shiny-server/~~**

**sudo ln -s /home/mkoenig/multiscale-galactose/R/shiny/gec\_app /srv/shiny-server/gec\_app**

## Restart shiny server

sudo stop shiny-server

sudo start shiny-server

status shiny-server

## Test app at the following location

Install text based browser for testing

sudo apt-get install lynx

sudo apt-get install links

lynx <http://localhost:3838>

lynx http://localhost:3838/gec\_app

# Configuration of shiny server

Necessary to make changes to the server configuration. Mainly run the server as mkoenig so that the standard R library locations can be used and set the port to 80 so that normal http requests from browser work.

sudo gedit /etc/shiny-server/shiny-server.conf

server {

**listen 80;**

# Define a location at the base URL

location / {

# User to run shiny server

# **run\_as mkoenig;**

# Provide a default/global GAID  
 # google\_analytics\_id "UA-12345-1";

# Host the directory of Shiny Apps stored in this directory

site\_dir /srv/shiny-server;

# Log all Shiny output to files in this directory

log\_dir /var/log/shiny-server;

# When a user visits the base URL rather than a particular application,

# an index of the applications available in this directory will be shown.

directory\_index on;

}

}

## Setting debug levels

sudo gedit /etc/init/shiny-server.conf

### SHINY\_LOG\_LEVEL

Defines the verbosity of logging which will be used by Shiny Server. Valid options -- in level of increasing verbosity -- are TRACE, DEBUG, INFO, WARN, and ERROR. More verbose levels of logging (such as TRACE) may be helpful when debugging an issue or trying to understand exactly what Shiny Server is doing, but will likely be far too much information for a system with even moderate load.

The default if this environment variable is not found is to use INFO.

## Apache2 proxy forwarding to shiny

<https://httpd.apache.org/docs/2.2/mod/mod_proxy.html>

<http://sharadchhetri.com/2013/08/02/how-to-install-mod_proxy-and-setup-reverse-proxy-in-apache-ubuntu/>

The module name is mod\_proxy ,it is core module which supports Forward and Reverse Proxy settings in Apache WebServer.

Install module with dependencies

sudo apt-get install libapache2-mod-proxy-html

sudo apt-get install libxml2-dev

Load the module

sudo a2enmod proxy proxy\_http

Change the virtual host settings in

/etc/apache2/sites-enabled/000-default

<VirtualHost \*:80>  
 ProxyPass /gec\_app/ [http://localhost:3838/gec\_app/](http://localhost:3838/app/)

ProxyPassReverse /gec\_app/<http://localhost:3838/app/>

RedirectMatch permanent ^/gec\_app$ /gec\_app/  
</VirtualHost>

<https://groups.google.com/forum/#!searchin/shiny-discuss/apache/shiny-discuss/q4JtGxkCaSk/GiYhNfgbWvoJ>