1. **-Criar um user**

adduser rstudio

gpasswd -a rstudio sudo

1. **trocar o user**

su - rstudio

1. **instalar nginx**

sudo apt-get update

sudo apt-get -y install nginx

O arquivo padrão que é veiculado está localizado em /var/www/html/index.nginx-debian.html; portanto, se você quiser alterar o que a página da Web está mostrando, edite esse arquivo com sudo nano / var / www / html / index.nginx-debian.html. Por exemplo, eu apenas coloquei um pouco de texto redirecionando para outras páginas na minha página de índice. O arquivo de configuração está localizado em /etc/nginx/nginx.conf.

Ao editar um arquivo HTML, você poderá ver as alterações imediatamente quando atualizar a página, mas se fizer alterações na configuração, precisará reiniciar o nginx. No futuro, você pode parar / iniciar / reiniciar o nginx com

sudo service nginx stop

sudo service nginx start

sudo service nginx restart

1. **instalar R**

Para garantir que obtemos a versão mais recente do R, precisamos primeiro adicionar biônico (o nome do lançamento do Ubuntu 18.04) ao nosso sources.list:

sudo sh -c 'echo "deb <http://cran.rstudio.com/bin/linux/ubuntu> bionic-cran35/" >> /etc/apt/sources.list'

Agora adicione as chaves públicas:

gpg --keyserver [keyserver.ubuntu.com](http://keyserver.ubuntu.com/) --recv-key E298A3A825C0D65DFD57CBB651716619E084DAB9

gpg -a --export E298A3A825C0D65DFD57CBB651716619E084DAB9 | sudo apt-key add -

Agora estamos prontos para instalar o R

sudo apt-get update

sudo apt-get -y install r-base

Agora você deve poder executar o R ​​e, esperançosamente, ser recebido com uma mensagem contendo a versão mais recente do R.

R

para sair R (quit ())

Precisamos adicionar 1 GB de espaço de swap para poder instalar pacotes:

sudo /bin/dd if=/dev/zero of=/var/swap.1 bs=1M count=1024

sudo /sbin/mkswap /var/swap.1

sudo /sbin/swapon /var/swap.1

sudo sh -c 'echo "/var/swap.1 swap swap defaults 0 0 " >> /etc/fstab'

Agora, a instalação da maioria dos pacotes funcionará. Mas antes de instalar qualquer pacote, é importante ter o devtools disponível para poder instalar pacotes R que estão no GitHub. No momento, o devtools não poderá ser instalado porque possui algumas dependências. Vamos instalar essas dependências:

sudo apt-get -y install libcurl4-gnutls-dev libxml2-dev libssl-dev

Ok, agora podemos começar a instalar pacotes R, tanto do CRAN quanto do GitHub!

sudo su - -c "R -e \"install.packages('devtools', repos='<http://cran.rstudio.com/>')\""

sudo su - -c "R -e \"devtools::install\_github('daattali/shinyjs')\""

1. **Instalar Rstudio**

sudo apt-get -y install gdebi-core

wget <https://download2.rstudio.org/rstudio-server-1.1.442-amd64.deb>

sudo gdebi rstudio-server-1.1.442-amd64.deb

Por padrão, o RStudio usa a porta 8787, portanto, para acessar o RStudio, vá para http://123.456.1.2:8787 e você deve ser recebido com uma página de login do RStudio. (Se você esqueceu o IP da sua gota, pode descobrir executando o nome do host -I)

1. **Instalar Shiny**

Para instalar o Shiny Server, primeiro instale o pacote brilhante:

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

Fazer download do shiny

wget <https://download3.rstudio.org/ubuntu-12.04/x86_64/shiny-server-1.5.6.875-amd64.deb>

sudo gdebi shiny-server-1.5.6.875-amd64.deb

O Shiny Server está agora instalado e em execução. Supondo que não houve problemas, se você acessar http://123.456.1.2:3838/, deverá ver a página inicial padrão do Shiny Server, que inclui algumas instruções e dois aplicativos Shiny:

Se você vir um erro no aplicativo Shiny inferior, provavelmente é porque não possui o pacote rmarkdown R instalado (as instruções na página padrão do Shiny Server mencionam isso). Depois de instalar o rmarkdown no R, o aplicativo Shiny inferior também deve funcionar. Não se esqueça de instalar o rmarkdown para que fique disponível para todos os usuários, conforme descrito acima. Eu sugiro que você leia a página de instruções em <http://123.456.1.2:3838/>.

**Referências do Shiny Server:**

- O log do Shiny Server está em /var/log/shiny-server.log.

- A página inicial padrão do Shiny Server que você está vendo está localizada em /srv/shiny-server/index.html - você pode editá-la ou removê-la.

- Qualquer diretório de aplicativo brilhante que você colocar em / srv / shiny-server / será servido como um aplicativo brilhante. Por exemplo, há um aplicativo padrão em / srv / shiny-server / sample-apps/ hello /, o que significa que você pode executar o aplicativo acessando http://123.456.1.2:3838/sample-apps/hello/.

- O arquivo de configuração do Shiny Server está em /etc/shiny-server/shiny-server.conf.

- Para recarregar o servidor após editar a configuração, use o serviço sudo service shiny-server restart

- Ao hospedar um arquivo Rmarkdown, nomeie o arquivo index.rmd e adicione runtime: shiny

ao frontmatter do documento.

Importante! Se você procurar no arquivo de configuração, verá que, por padrão, os aplicativos são executados como usuários shiny. É importante entender qual usuário está executando um aplicativo porque coisas como permissões de arquivo e bibliotecas pessoais R serão diferentes para cada usuário e isso pode causar dores de cabeça (ou muitos dias de dores de cabeça na minha experiência) até você perceber que é porque o aplicativo deve Não seja tão shiny. Mantenha isso em mente. O fato de os aplicativos serem executados com o usuário shiny significa que qualquer pacote necessário em um aplicativo shiny precisa estar na biblioteca global ou na biblioteca do shiny. Como mencionei acima, pode ser necessário instalar pacotes R de uma maneira especial para garantir que o usuário shiny possa acessá-los.

1. **Atribuir permissões ao usuário rstudio e Shiny a mesma pasta**

sudo groupadd shiny-apps

sudo usermod -aG shiny-apps rstudio

sudo usermod -aG shiny-apps shiny

cd /srv/shiny-server

sudo chown -R rstudio:shiny-apps .

sudo chmod g+w .

sudo chmod g+s .

1. **Configurar as URLs**

sudo nano /etc/nginx/sites-enabled/default

Adicione as seguintes linhas acima da linha que lê o server {

map $http\_upgrade $connection\_upgrade {

default upgrade;

'' close;

}

E adicione as seguintes linhas logo após a linha que lê server\_name \_;

location /shiny/ {

proxy\_pass <http://127.0.0.1:3838/>;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection $connection\_upgrade;

rewrite ^(/shiny/[^/]+)$ $1/ permanent;

}

location /rstudio/ {

proxy\_pass <http://127.0.0.1:8787/>;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection $connection\_upgrade;

}

Desde que alteramos a configuração do nginx, precisamos reiniciar o nginx para que ele entre em vigor.

sudo service nginx restart

**LINUX**

Criar directorio

mkdir dirname

Copiar ficheiro para directorio

cp filename /tmp

Se o comando descompactar ainda não estiver instalado no seu sistema, execute:

sudo apt-get install unzip

Após instalar o utilitário descompactar, se você deseja extrair para uma pasta de destino específica, poderá usar:

unzip file.zip -d destination\_folder

Se os diretórios de origem e destino forem os mesmos, você pode simplesmente:

unzip file.zip

//

‘geoaxe’, ‘rgbif’, ‘rgeos’, ‘rgdal’

You have to install gdal library

sudo apt install gdal-bin proj-bin libgdal-dev libproj-dev

‘units’, ‘sf’, ‘rnaturalearth’

sudo apt-get install libudunits2-dev