SIEnergiaViewer - Instalação no servidor

Projeto R6/PTI/EPE

2023-12-15

Índice

Atualizado em: 2023-12-15

Introdução

O objetivo deste documento é descrever o processo de instalação do aplicativo "SIEnergiaViewer" no servidor.

Dependências do sistema

A seguir, listamos as dependências do aplicativo, que deverão ser instaladas no servidor antes dos passos descritos neste documento.

Shiny Server

O Shiny Server é um servidor web que permite a execução de aplicativos Shiny em um servidor Linux, acessado a partir de um navegador web.

Dependências do pacote sf

O pacote sf é usado no otimizadorApp e possui diversas dependências, que podem ser instaladas usando o seguinte comando no terminal:

```
sudo dnf install gdal-devel proj-devel geos-devel sqlite-devel udunits2-devel
```

Leia mais em: https://r-spatial.github.io/sf/#installing

Dependências do pacote tidyverse

O pacote tidyverse é usado no otimizadorApp e possui diversas dependências, que podem ser instaladas usando o seguinte comando no terminal:

```
sudo dnf install libcurl-devel libxml2-devel openssl-devel libsodium-devel
```

Saiba mais em: https://tidyverse.tidyverse.org/

Solver do otimizador

O otimizador está preparado para utilizar os seguintes solvers: glpk, cplex, cbc, e gurobi.

glpk

O glpk é um solver de código aberto, que pode ser instalado no servidor usando o seguinte comando no terminal:

```
sudo dnf install glpk-devel
```

CPLEX

O aplicativo utiliza o otimizador CPLEX para resolver o problema de otimização.

O CPLEX é um software de otimização matemática desenvolvido pela IBM.

É possível fazer o download e ler as instruções de instalação neste link: https://www.ibm.com/products/ilog-cplex-optimization-studio/cplex-optimizer.

Deploy no Shiny Server

O usuário do servidor onde os pacotes são instalados precisa ser o mesmo onde o app será instalado.

Os códigos a seguir devem ser executados no console do R, exceto quando citado o contrário.

Configurando as variáveis de ambiente

As variáveis de ambiente são variáveis que podem ser configuradas no sistema operacional e que podem ser acessadas pelo R. Elas são utilizadas para armazenar informações que são utilizadas por diferentes programas.

Após configurar as variáveis de ambiente, **é necessário reiniciar o R para que as alterações tenham efeito**.

Caso esteja no RStudio Server, é necessário clicar no botão “Session” (localizado no menu superior) e depois em “Restart R”.

Caminho de instalação do CPLEX

Faça essa configuração apenas se for utilizar o solver CPLEX.

O caminho de instalação do CPLEX deve ser configurado na variável de ambiente `CPLEX_BIN`. O caminho a seguir foi o utilizado no servidor de homologação. Caso necessário, alterar o caminho da instalação do CPLEX de acordo com o servidor de produção.

Em caso de dúvidas, entrar em contato com Elisabete ou outra pessoa da equipe de Infra/TI, responsável pela instalação do CPLEX no servidor.

```
Sys.setenv(
  CPLEX_BIN = "/opt/ibm/ILOG/CPLEX_Studio1210/cplex/bin/x86-64_linux/cplex"
  )
```

GitHub PAT

O GitHub PAT (Personal Access Token) é um token de acesso pessoal que permite o acesso à repositórios privados do GitHub, útil por exemplo no caso da instalação de pacotes diretamente do GitHub, acesso à arquivos disponíveis nos repositórios, etc.

O GitHub PAT deve ser configurado na variável de ambiente `GITHUB_PAT`. O PAT tem um prazo de validade que é definido no momento da criação do mesmo. Caso necessário, gerar um novo token de acordo e atualizar o valor da variável de ambiente.

O anexo "Criação do GitHub PAT" descreve o passo-a-passo para a criação de um novo PAT (Seção).

```
Sys.setenv(
  GITHUB_PAT = "ghp_xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
  )
```

Instalando pacotes necessários

Os pacotes necessários para a execução do aplicativo são:

RCplex: Pacote que permite a integração do CPLEX com o R.

- INSTALE APENAS SE FOR UTILIZAR O SOLVER CPLEX.

Website: https://cran.r-project.org/web/packages/Rcplex/index.html

```
devtools::install_github("curso-r/Rcplex")
```

Rglpk: Pacote que permite a integração do glpk com o R.

- INSTALE APENAS SE FOR UTILIZAR O SOLVER glpk.

Website: https://cran.r-project.org/web/packages/Rglpk/index.html

```
install.packages("Rglpk")
```

- **otimizadorLinear**: Pacote desenvolvido neste projeto, que apresenta as funções de otimização linear.

```
devtools::install_github("curso-r/otimizadorLinear")
```

- **sienergiaViewer**: Pacote desenvolvido neste projeto, que apresenta o código do aplicativo em Shiny.

```
devtools::install_github("curso-r/sienergiaViewer")
```

- **piggyback**: Pacote que permite o download de arquivos diretamente do GitHub.

```
install.packages("piggyback")
```

- **pak** e **pkgdepends**: Pacote que permite o download de arquivos diretamente do GitHub.

```r
install.packages("pkgdepends")
install.packages("pak")
```

Instalando o aplicativo pela primeira vez

O aplicativo deve ser instalado no servidor em um diretório específico. O diretório padrão é `/home/NOME.USUARIO/ShinyApps`. Caso necessário, alterar o diretório de acordo com o servidor de produção, em todos os códigos abaixo onde o diretório é citado.

Neste exemplo, utilizaremos o diretório:

`/home/andrea.pinto/ShinyApps/`

Baixando o arquivo Zip do aplicativo

O primeiro passo é fazer o download do arquivo zip do aplicativo, que está disponível no repositório do GitHub.

Esse aplicativo zip contém:

- Um arquivo `app.R`
- Pasta contendo dados.

Exemplo:

```r
piggyback::pb_download(
  file = "deploy-sienergiaViewer.zip",
  repo = "curso-r/sienergiaViewer",
  tag = "zip-deploy",
  dest = "/home/andrea.pinto/ShinyApps/",
  show_progress = TRUE,
)
```

Descompactando o arquivo Zip

Os arquivos do aplicativo devem ser descompactados no diretório a seguir, através do argumento `exdir`.

`` `/home/NOME.USUARIO/ShinyApps/NOME-DO-APP` ``

Exemplo:

```
unzip(
  zipfile = "/home/andrea.pinto/ShinyApps/deploy-sienergiaViewer.zip",
  exdir = "/home/andrea.pinto/ShinyApps/deploy-sienergiaViewer"
  )
```

Para acessar o aplicativo no navegador, é necessário acessar o endereço:

`http://vsvr-shinyhtml/users/NOME.USUARIO/NOME-DO-APP/`

Exemplo: `vsvr-shinyhtml/users/andrea.pinto/deploy-sienergiaViewer/`.

Banco de dados

O banco de dados do aplicativo é atualizado periodicamente.

Para baixar a versão mais recente dos dados, é necessário executar o seguinte comando no R:

```
piggyback::pb_download(
  file = "SIEnergia_dados.sqlite",
  repo = "curso-r/sienergiaViewer",
  tag = "bd-sienergia",
  dest = "/home/andrea.pinto/ShinyApps/deploy-sienergiaViewer/",
  show_progress = TRUE
)
```

Alteração do solver

Para alterar o solver utilizado, abra o arquivo `app.R`, e altere na função `sienergiaviewer::run_app()` o argumento `motor_otimizador` para o solver desejado.

Atualização do aplicativo

Para atualizar o aplicativo, é necessário atualizar os pacotes otimizadorLinear e sienergiaViewer. Para isso, é necessário executar os seguintes comandos no R:

```r
# Atualizar o pacote otimizadorLinear
pak::pkg_install("curso-r/otimizadorLinear")


# Atualizar o pacote sienergiaViewer
pak::pkg_install("curso-r/sienergiaViewer")
```

Reiniciar o Shiny-server

Para reiniciar o Shiny-server, é necessário executar o seguinte comando **no terminal**:

```
sudo systemctl restart shiny-server
```

Atenção: para que isso seja possível, é necessário que o usuário tenha permissão de super usuário (sudo). Caso não tenha, é necessário solicitar ao responsável pela infraestrutura do servidor.

Anexos

Criação do GitHub PAT

O passo-a-passo a seguir descreve como criar um novo GitHub PAT. É necessário ter uma conta no GitHub (https://github.com/), com acesso aos repositórios privados do projeto.

Utilizaremos o pacote `usethis`:

```r
install.packages("usethis")
```

A função `usethis::create_github_token()` abrirá uma janela no navegador, onde é possível criar um novo token. É necessário dar um nome para o token e definir a data de expiração. Após a criação do token, copie-o e salve-o em um local seguro.

```r
usethis::create_github_token()
```

Após a criação do token, é necessário salvá-lo usando a função `gitcreds::gitcreds_set()`. Essa função solicitará que você cole o token criado anteriormente no console do R

```
gitcreds::gitcreds_set()
```

Também é necessário atualizar o token nas variáveis de ambiente:

```
Sys.setenv(
  GITHUB_PAT = "ghp_xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
  )
```

Importante: após essas etapas, reinicie o R.