https://devcontent.com.br/artigos/git-e-github/guia-de-comandos

Passo a passo:

1. Construir aplicação Streamlit;
2. Criar arquivos requirements.txt, setup.sh e Procfile;
3. Configurar repositório Git;
4. Configurar Heroku;
5. Deploy.

# Construir aplicação Streamlit

Antes de mais nada precisamos ter uma aplicação Streamlit rodando localmente.

Criar arquivos requirements.txt, setup.sh e Procfile

O Heroku exige alguns arquivos para conseguir instalar as dependências e rodar nossa aplicação Streamlit. Eles devem constar na pasta raíz de seu projeto. Vamos analisar como construir cada um.

**requirements.txt**

Este arquivo deve conter todas as bilbiotecas necessárias para rodar seu projeto. Ele pode ser criado manualmente conferindo todas bibliotecas utilizadas em seus arquivos Python, ou pode ser gerado automaticamente utilizando o pipreqs (recomendo). Você deve substituir <caminho\_projeto> pelo caminho até seu projeto (por exemplo /home/guma/Desktop/Projetos/Tutorial-Streamlit) ou se você rodar já de dentro do projeto basta substituir por**./**.

pipreqs <caminho\_projeto>

Se você utilizou o *pipreqs, o*arquivo requirements.txt (deve ter esse nome e essa extensão) já foi gerado e deve ter ficado mais ou menos assim:

numpy==1.18.3  
pandas==1.0.3  
streamlit==0.58.0

**setup.sh**

Este arquivo ajuda a construir as configurações de ambiente necessárias, criando a pasta ~/.streamlit e colocando dentro dela os arquivos credentials.toml e config.toml. Copie o conteúdo abaixo substituindo apenas o seu email.

mkdir -p ~/.streamlit/echo "\  
[general]\n\  
email = \"[seu-email@dominio.com](mailto:gustavomdom@gmail.com)\"\n\  
" > ~/.streamlit/credentials.tomlecho "\  
[server]\n\  
headless = true\n\  
enableCORS=false\n\  
port = $PORT\n\  
" > ~/.streamlit/config.toml

**Procfile**

O Procfile (deve ser escrito assim mesmo, sem extensão!) é quem vai chamar para rodar o setup.sh e seu arquivo python.

web: sh setup.sh && streamlit run sistemavendas.py

# Configurar repositório Git

O Heroku permite fazer o deploy utilizando diferentes tecnologias como o Git e o Docker. Mas optamos pelo Git pela simplicidade.

Dentro da pasta raíz de seu projeto rode o comando git init e então já terá criado seu repositório.

guma@Gunote:~/Desktop/Projetos/Tutorial-Streamlit$ git init  
Initialized empty Git repository in /home/guma/Desktop/Projetos/Tutorial-Streamlit/.git/

# Configurar Heroku

Crie uma conta gratuita no [site do Heroku](http://heroku.com/), ela será suficiente para fazer deploy de alguns projetos.

Também terá que instalar o CLI (Heroku Command Line Interface), basta executar o comando adequado descrito[nesta página](https://devcenter.heroku.com/articles/getting-started-with-python#set-up).

Depois de concluída a instalação do CLI, abra um terminal e navegue até a pasta do seu projeto. Lá, execute o comando heroku login.

~/Desktop/Projetos/Tutorial-Streamlit$ heroku login  
heroku: Press any key to open up the browser to login or q to exit:   
Opening browser to [https://cli-auth.heroku.com/auth/cli/browser/123](https://cli-auth.heroku.com/auth/cli/browser/a3a28a99-b835-42e8-aec8-f86e5826d085)  
Logging in... done  
Logged in as [seu-email@dominio.com](mailto:gustdomar@gmail.com)

# Deploy

Primeiro passo é criar sua aplicação no Heroku, basta digitar o comando heroku create <nome\_da\_sua\_aplicação>. O nome que você escolher vai aparecer no link do site, então escolha algo legal.

~/Desktop/Projetos/Tutorial-Streamlit$ heroku create tutorial-streamlit  
Creating ⬢ tutorial-streamlit… done  
<https://tutorial-streamlit.herokuapp.com/> | <https://git.heroku.com/tutorial-streamlit.git>

Sempre que quiser atualizar ou enviar sua primeira versão do projeto basta seguir os passos abaixo.

git status  
git add .  
git commit -m "some message"  
git push heroku master

O git status indica quais arquivos foram alterados desde sua última atualização no repositório. A partir desta lista, utilize git add . para adicionar todos arquivos modificados ou substitua o ponto pelo nome dos arquivos que quer atualizar.

O git commit prepara os arquivos para serem enviados para o Heroku e permite adicionar uma mensagem através do -m.

Se o git reclamar que não conseguiu identificar seu email quando você rodou git commit, adicione as linhas abaixo no arquivo config que está na pasta .git dentro do seu projeto.

[user]  
 name = Seu Nome  
 email = [seu-email@dominio.com](mailto:gustdomar@gmail.com)

Para finalizar, dê o git push heroku master para enviar sua aplicação de vez. Ao final da execução você terá o link com sua aplicação rodando na nuvem.

remote: — — -> Discovering process types  
remote: Procfile declares types -> web  
remote:   
remote: — — -> Compressing…  
remote: Done: 125.2M  
remote: — — -> Launching…  
remote: Released v3  
remote: <https://tutorial-streamlit.herokuapp.com/> deployed to Heroku  
remote:   
remote: Verifying deploy… done.  
To <https://git.heroku.com/tutorial-streamlit.git>  
 \* [new branch] master -> master

Reiniar heroku

First delete the git repo on the server:

> cd /my-project/

> heroku plugins:install heroku-repo

> heroku repo:reset

Then re-initialise your local git repo by deleting and recreating it:

> cd /my-project/

> rm -rf .git

> git init

> heroku git:remote -a <appname>