

Guia de Implantação

Versão 1

JOB SEA – MAR DE OPORTUNIDADES

Histórico de Revisões

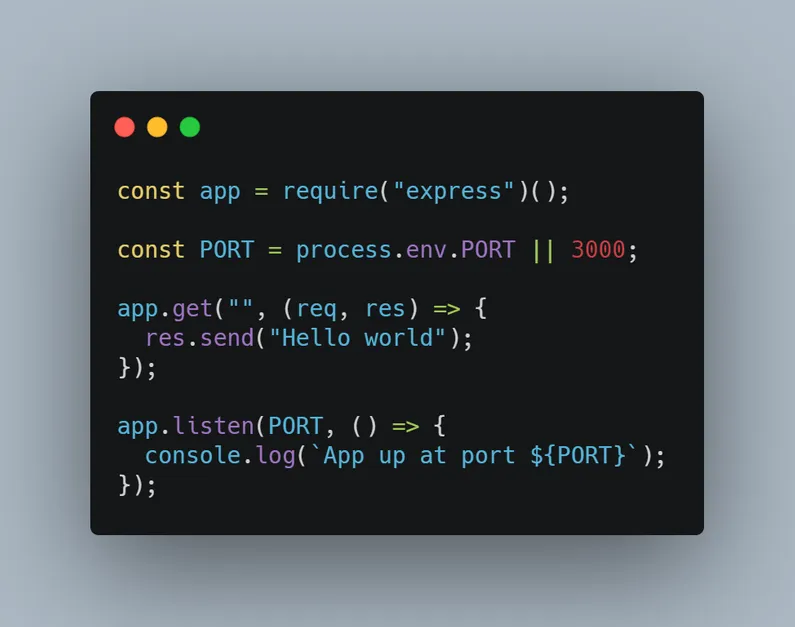
|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 24/01/2021 | 01 | Projeto foi implantado na plataforma Heroku, conforme dados abaixo. | Aluisio José Galvão dos Santos. |

**GUIA DE IMPLANTAÇÃO**

**Etapa 1 – Crie o projeto**

Em uma nova pasta, vou abrir um terminal e executar o comando npm init -y para criar um projeto. O servidor fictício será escrito em expressar, então precisamos executar o npm install express comando para instalar este módulo.

Após a instalação desta biblioteca, podemos criar um arquivo para o nosso projeto, chamado app.js. Dentro dele, escreveremos o código para o nosso servidor simples:



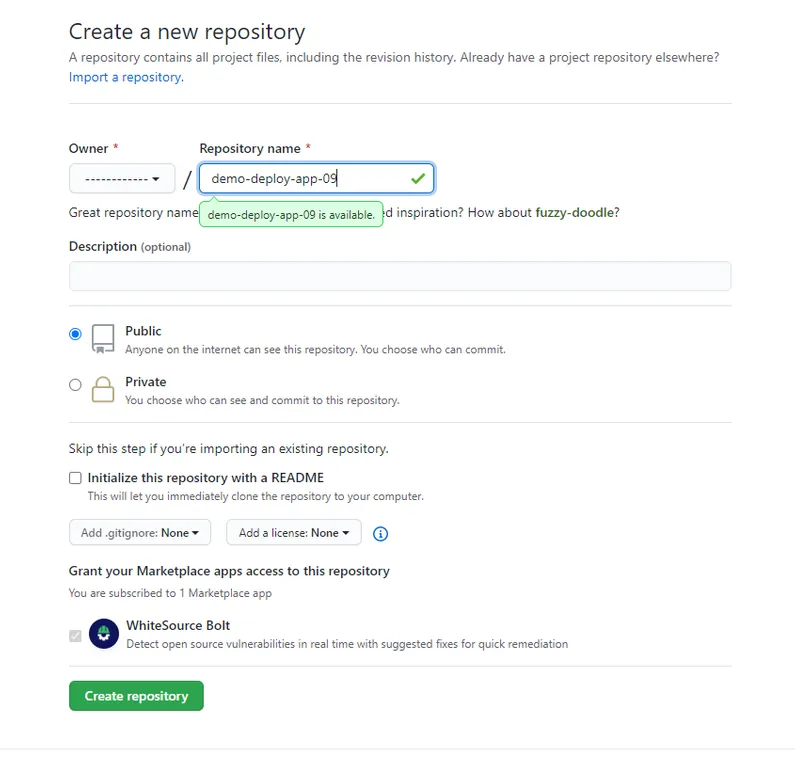
Podemos iniciar o aplicativo executando node app.js. Então podemos experimentá-lo no seguinte URL http://localhost:3000. Neste ponto, você deverá ver a mensagem Hello World no navegador.

**Etapa 2 – Sistema de controle de versão**

O próximo passo é escolher um sistema de controle de versão e colocar nosso código em uma plataforma de desenvolvimento em um repositório.

O sistema de controle de versão mais popular é Git junto com Github como uma plataforma de desenvolvimento, é isso que usaremos aqui.

No GitHub, vá em frente e crie um repositório para o seu aplicativo, assim:



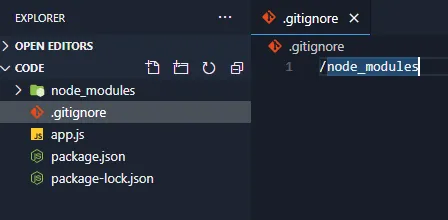
Para fazer upload do seu código local em um repositório, você precisa executar os comandos listados no Github depois de clicar em Create repository botão:



Comandos para fazer upload de nosso código no repositório do Github

Antes de fazer isso, devemos ignorar alguns arquivos. Queremos fazer upload para o repositório apenas do código que escrevemos, sem as dependências (os módulos instalados).

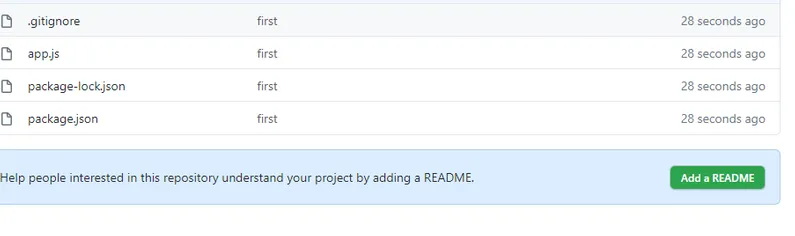
Para isso, precisamos criar um arquivo .gitignore e dentro dele escreva o arquivo que queremos ignorar.



Estrutura de pastas e arquivo .gitignore

Agora, podemos escrever os comandos listados na figura acima (a do GitHub).

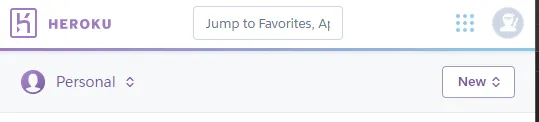
Se você executou os comandos corretamente, ele estará na página do seu repositório. Se você o atualizar, deverá ver seus arquivos, exceto o que você ignorou explicitamente, a saber node modules.



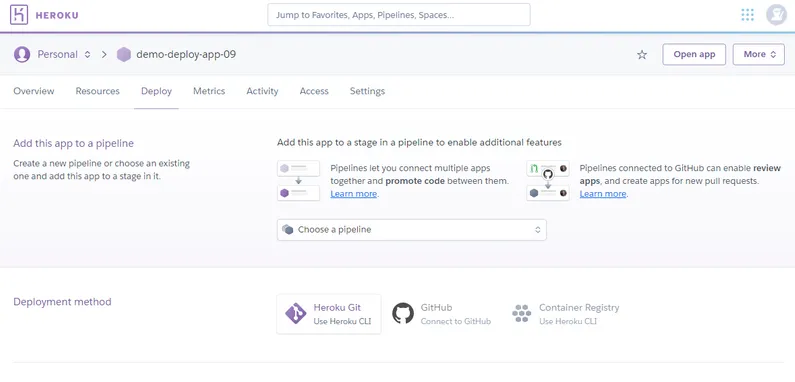
**Etapa 3 – Vincular o repositório ao Heroku**

Nesta etapa, podemos vincular o repositório do Github ao nosso aplicativo Heroku.

Primeiro, crie um aplicativo no Heroku e siga as etapas listadas na plataforma.

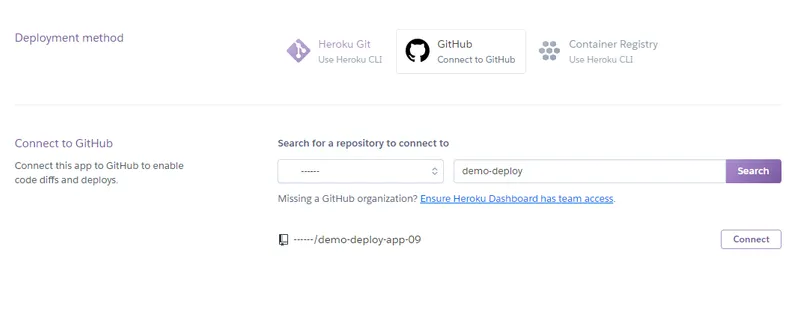


Depois que o aplicativo for criado, uma janela semelhante a esta deve aparecer:

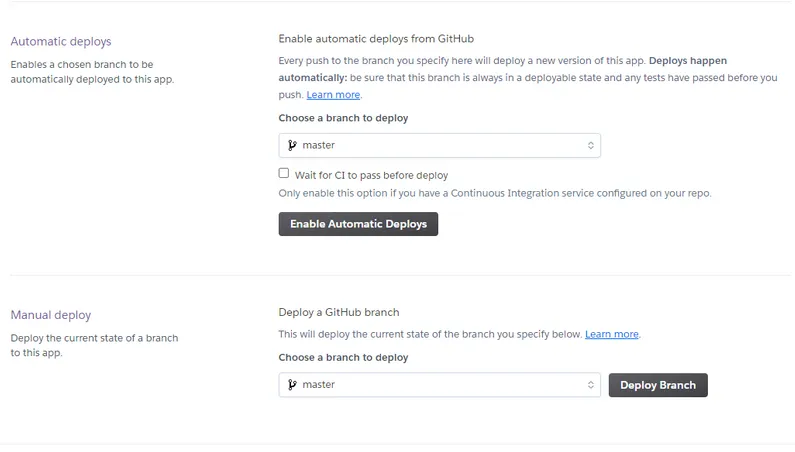


**Painel de aplicativos**

Agora, se você olhar para a navegação na parte superior, verá Overview, Resources, Deploy, Metrics e assim por diante. Tenha certeza disso Deploy é selecionado. Em seguida, na segunda linha, clique no ícone GitHub.



**Clique em conectar**

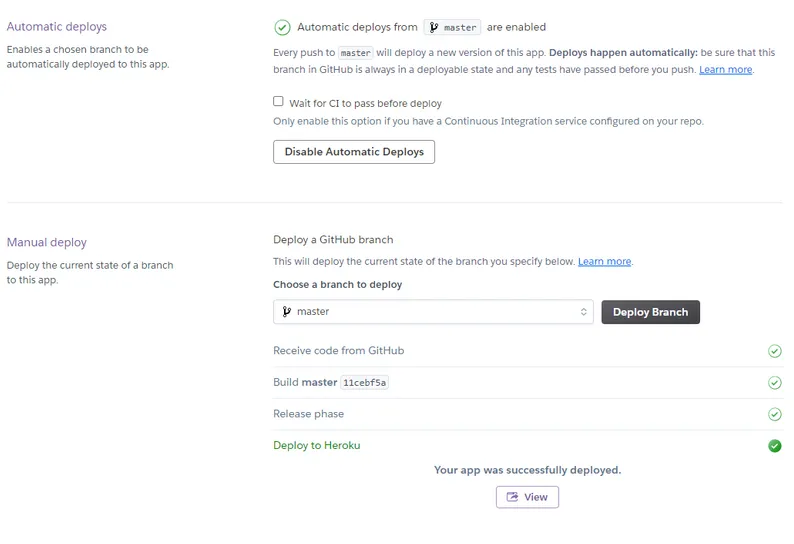
Procure o aplicativo desejado, que é demo-deploy-app-09 no nosso caso. Então clique Connect.

**Implantar Filial**

Depois que o aplicativo for conectado com sucesso à sua conta Heroku, você poderá clicar em Deploy Branch para implantar seu aplicativo.

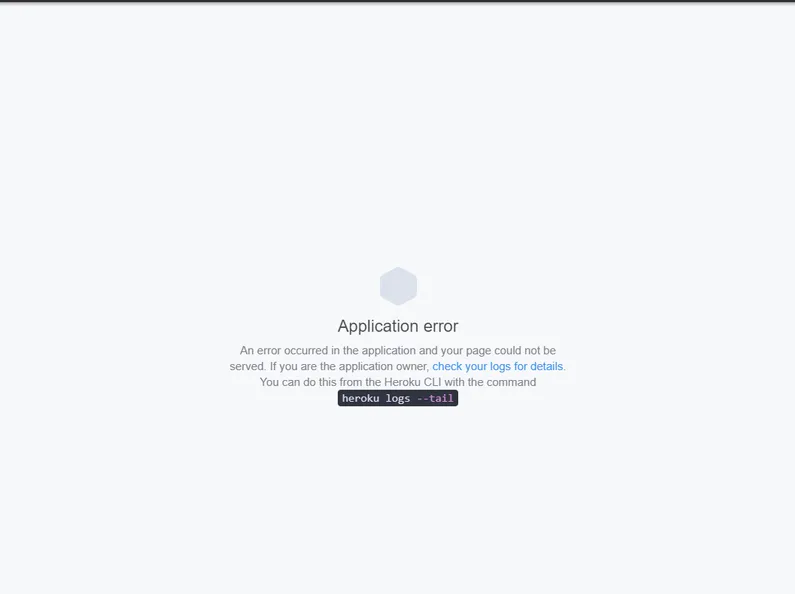
Se você quiser, também pode selecionar a opção Enable Automatic Deploys que extrairá automaticamente o código do seu repositório do Github toda vez que você enviar um push para esse repositório.

Depois que o aplicativo foi implantado, você pode clicar em Exibir para abrir seu aplicativo.



**Etapa 4 – Configure o Heroku para executar corretamente o aplicativo**

Se você abrir o aplicativo neste momento, deverá ver algo assim:



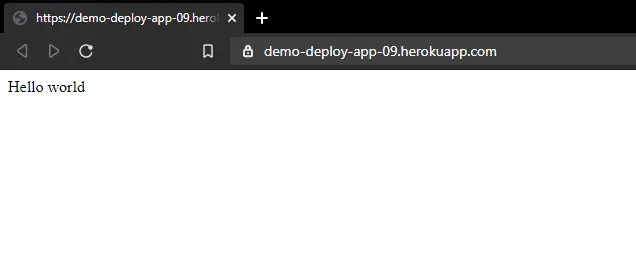
Isso mesmo, um erro. Isso porque o Heroku não sabe como iniciar nosso aplicativo.

Se você se lembra, executamos o comando node app.js para iniciar o aplicativo localmente.  
O Heroku não tem como saber quais comandos ele precisa executar para iniciar o aplicativo, e é por isso que gerou um erro.

Para resolver esse problema, precisamos criar um arquivo chamado Procfile com o seguinte conteúdo: web: node ./app.js.

Para atualizar nosso aplicativo, tudo o que precisamos fazer é enviar um novo commit ao GitHub. Se tivermos ativado o Automatic Deploys opção, o código será automaticamente puxado para o Heroku. Caso contrário, precisamos clicar em Deploy Branch novamente.

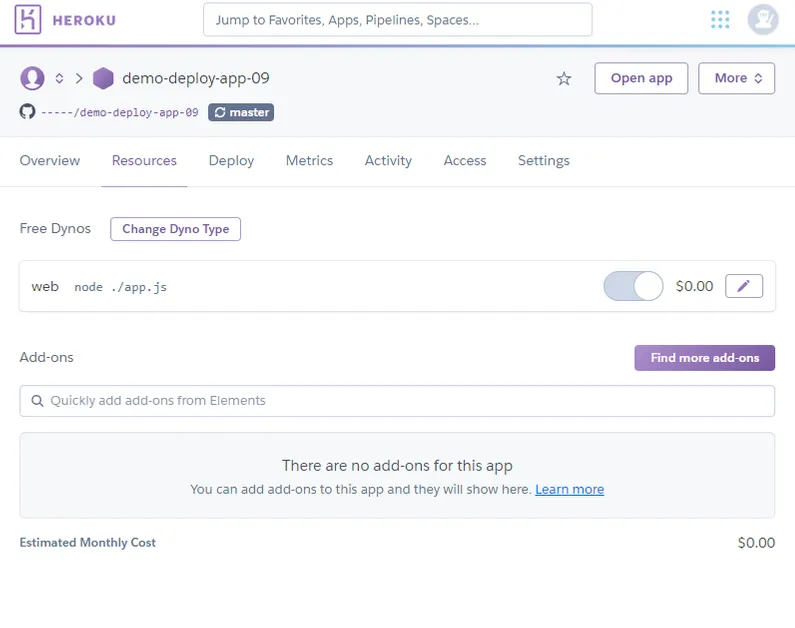
Depois que o aplicativo for reconstruído, deveremos vê-lo funcionando da seguinte maneira:



**Etapa 5 – Como adicionar um complemento**

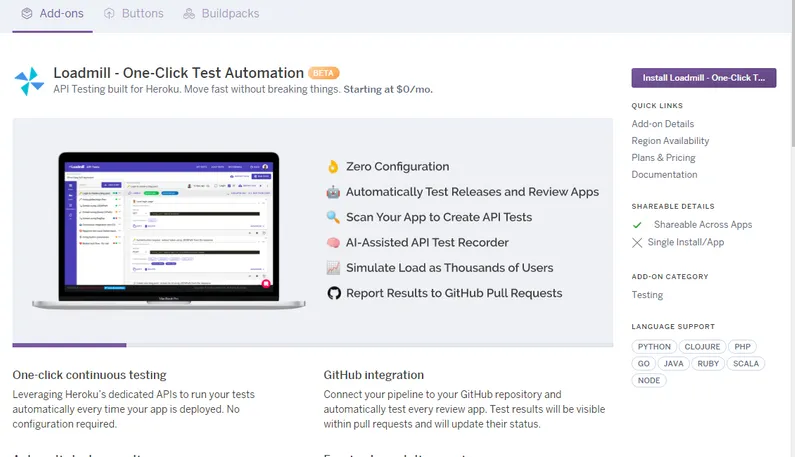
Um dos principais benefícios que o Heroku oferece é o fato de que você pode adicionar recursos facilmente na forma de add-ons ao seu projeto. Esses recursos externos vêm na forma de bancos de dados, ferramentas de registro e monitoramento, ferramentas de CI e CD ou ferramentas de teste.

Então agora vamos ver como adicionar um novo recurso ao seu projeto. Primeiro, iremos para Recursos e, a partir daí, adicionaremos uma nova ferramenta para teste.

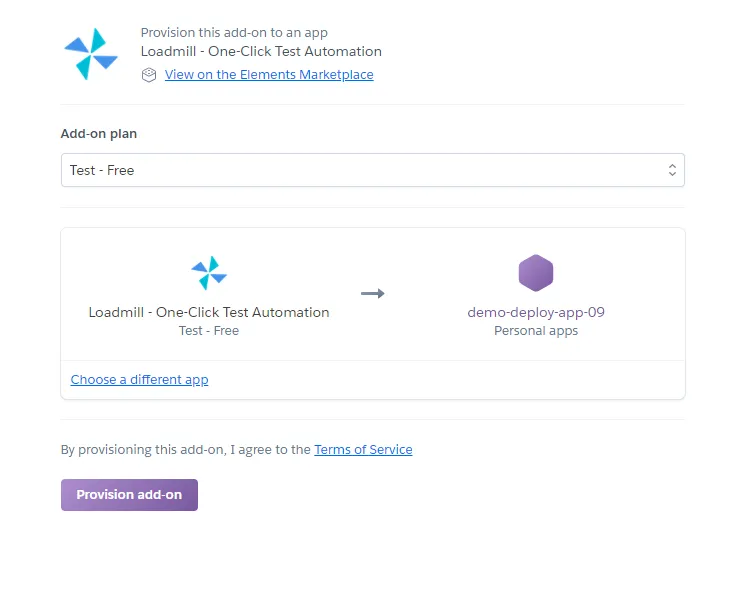


**Vá em frente e clique em Find more add-ons e depois procure Loadmill.**

Loadmill é uma ferramenta de teste realmente excelente para testes de regressão e testes de carga.

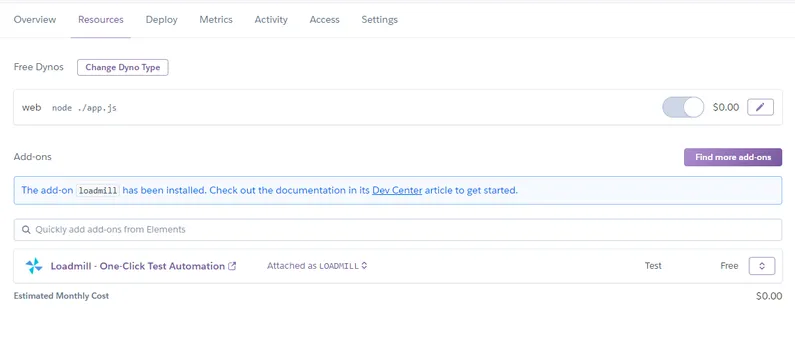


**Vá em frente e clique em Install…. Em seguida, escolha o aplicativo que você deseja vincular.**

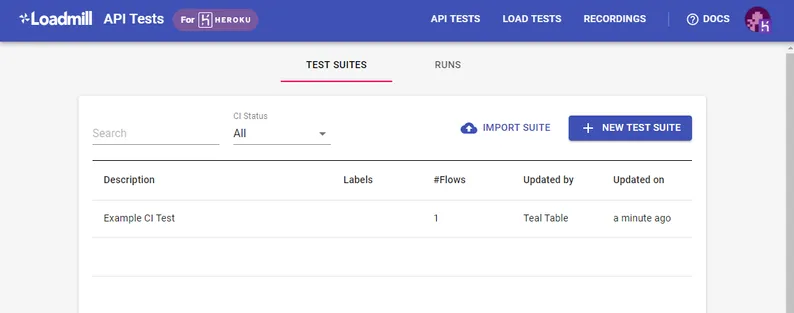


Nesta etapa, o Heroku criará automaticamente uma conta para você na plataforma provisionada.

Na guia recursos, você pode ver o recurso adicionado recentemente:



Se você acessar esse complemento, deverá ver o painel com um tutorial de introdução e um teste de demonstração criado para você.



**Painel do moinho de carga**

O Heroku permite que os desenvolvedores implantem um aplicativo de maneira rápida e quase indolor em um servidor da web.

Ele também fornece muitos plugins que você pode integrar ao seu aplicativo.

Uma solução PaaS sempre permitirá que você se mova mais rápido do que a solução com um VPS, no qual você precisa configurar tudo desde o início.

Link do projeto:

<https://jobsea-front.herokuapp.com/>