

Guia Rápido de Implantação de Modelos no AWS Elastic Beanstalk usando Python

Revisão1- 29/5/2019- itens:

5- Advertência na criação do arquivo zip

6- Incluído o método alternativo para upload durante o debugging

O AWS Elastic Beanstalk é um serviço para a implantação de aplicações e serviços em Java, .NET, PHP, Node.js, Python, Ruby, Go e Docker. O serviço tem um certo grau de automatização, mas mesmo assim pode ser desafiador para iniciantes na plataforma. Este guia tem o propósito de ser uma referência para implantação de aplicações em python com Flask, já que a documentação disponível no site da AWS tem várias deficiências.

1- Criando sua conta

Crie uma conta gratuita em <https://aws.amazon.com/pt/free/> .Esta conta inclui gratuitamente durante um ano os seguintes limites:

- 2000 requisições *put,copy,post* ou *list* por mês no Amazon S3
- 20000 requisições *get* por mês no Amazon S3
- 30 GB no *Amazon Elastic Block Storage*
- 750 horas de uso por mês do *Amazon EC2 Microsoft Windows Server t2.micro*
- 750 horas de uso por mês do *Amazon EC2 Linux t2.micro*

2- Ajustando o código python/flask

Alguns ajustes são necessários para que o código funcione no AWS Elastic Beanstalk. Nas linhas em que define a aplicação Flask e roteamento , oque se tem geralmente é:

```
app = Flask(__name__)
```

```
@app.route('/')
```

```
.
```

```
.
```

```
.
```

```
@app.route('/index')
```

No AWS Elastic o nome da aplicação deve ser **o mesmo nome do seu arquivo .py**, além disto é preciso definir um caminho para a pasta 'templates', onde o Flask procura os arquivos .html, a maneira mais simples é **colocar seus arquivos html na pasta 'static'**.

Altere seu código assim:

```
application = Flask(__name__, template_folder='static')
```

```
your_file_name = application
```

Agora altere em todo o código **todas as linhas** `@app.route` para `@your_file_name.route`, resultando em:

```
@your_file_name.route('/')  
  
.  
  
.  
  
.  
  
@your_file_name.route('/index')
```

Agora vá para o final do seu código, geralmente o que se tem é:

```
if __name__ == "__main__":
```

```
    app.run()
```

Altere o nome do aplicativo flask e ative a o estado debugger, para facilitar a localização de erros no AWS.

```
if __name__ == "__main__":
```

```
    your_file_name.debug = True
```




```
    your_file_name.run()
```

Se a sua aplicação abre algum arquivo de dados altere o caminho para que o arquivo possa ser localizado no AWS Elastic:


```
with open(join(dirname(realpath(__file__)), 'static/')+"data.csv") as file1
```

3- Criar a pasta .ebextensions

É necessário criar dentro da pasta local do seu arquivo a pasta `.ebextensions`, com o arquivo `python.config` definindo as configurações de arquivos estáticos e de WSGI para o AWS Elastic:

Nome	Data de modificaç...	Tipo	Tamanho
 .ebextensions	25/05/2019 11:33	Pasta de arquivos	
 static	25/05/2019 12:33	Pasta de arquivos	
 your_file_name.py	28/05/2019 10:01	Arquivo PY	1 KB

Conteúdo da pasta .ebextensions:

Nome	Data de modificaç...	Tipo	Tamanho
 python.config	25/05/2019 11:45	Arquivo CONFIG	1 KB

O conteúdo do arquivo python.config deve ter o seguinte conteúdo :





```
option_settings:
  "aws:elasticbeanstalk:container:python:staticfiles":
    "/static/": "static/"
  "aws:elasticbeanstalk:container:python":
    WSGIPath: your_file_name.py
```

Copiar e colar é a melhor forma de evitar erros, pois o espaçamento deve idêntico ao que está mostrado abaixo:

```
option_settings:
  "aws:elasticbeanstalk:container:python:staticfiles":
    "/static/": "static/"
  "aws:elasticbeanstalk:container:python":
    WSGIPath: your_file_name.py
```

4- Criar o arquivo requirements.txt

Na pasta do seu arquivo crie o arquivo 'requirements.txt' com as bibliotecas python e respectivas versões que a sua aplicação necessita.

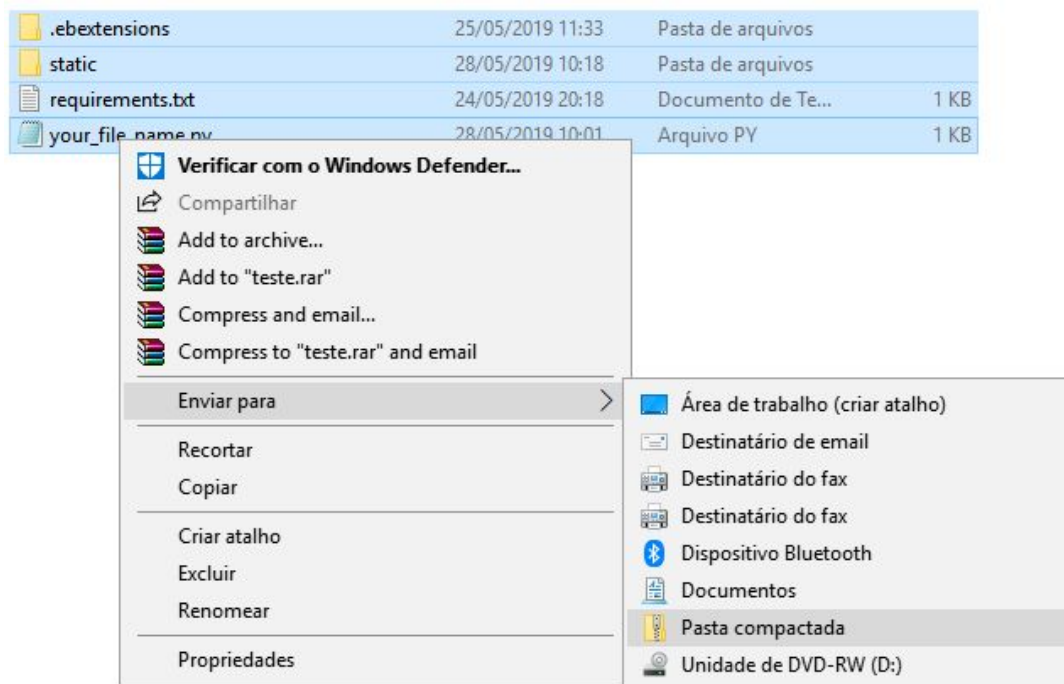
 .ebextensions	25/05/2019 11:33	Pasta de arquivos	
 static	28/05/2019 10:18	Pasta de arquivos	
 requirements.txt	24/05/2019 20:18	Documento de Te...	1 KB
 your_file_name.py	28/05/2019 10:01	Arquivo PY	1 KB

Conteúdo do arquivo 'requirements.txt', o exemplo abaixo pode ser utilizado para aplicações básicas que só utilizam pandas e numpy:

```
pandas==0.23.4
numpy==1.14.2
click==6.7
Flask==1.0.2
itsdangerous==0.24
Jinja2==2.10
MarkupSafe==1.0
Werkzeug==0.14.1
```

5- Criando o arquivo zip

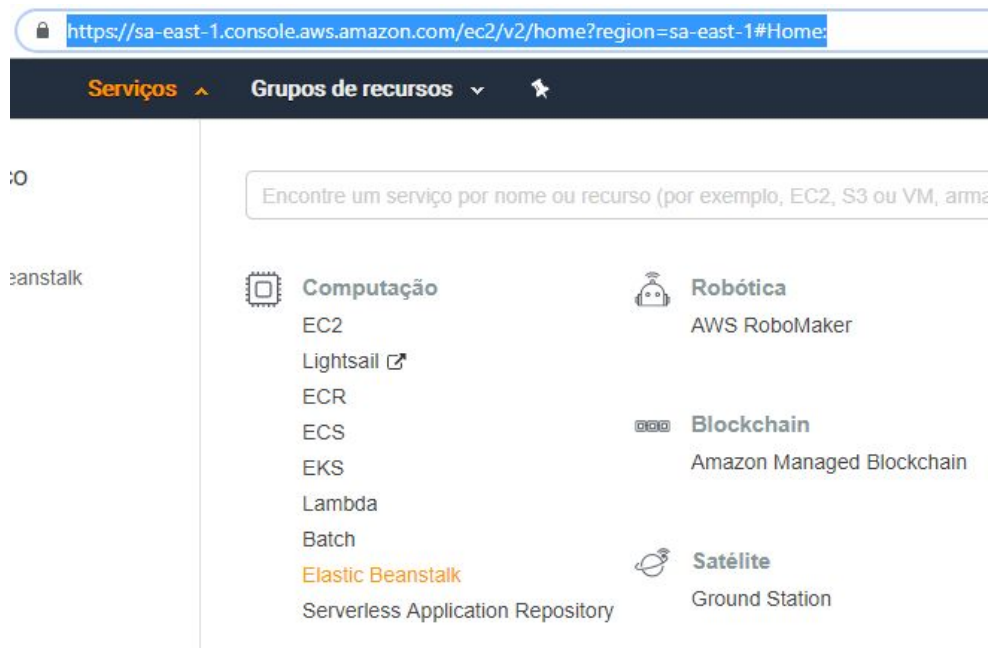
O AWS Elastic precisa de um arquivo zipado do seu projeto, ele extrai o arquivo automaticamente e instala as bibliotecas. Faça exatamente como mostrado abaixo, **não inclua o diretório que contém todo o seu projeto**, senão um erro ocorrerá na hora de implantação na AWS.



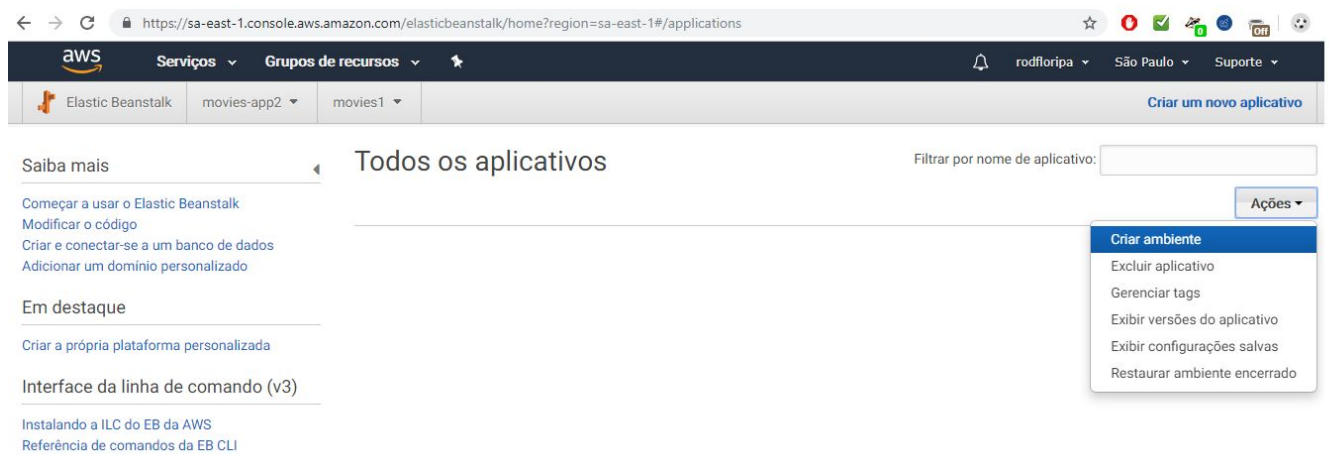
6- Fazendo o upload no AWS Elastic Beanstalk

O site da AWS possui vários serviços, um iniciante pode ficar perdido com a quantidade de links. Recomenda-se a criação de Bookmarks dos dois links que você utilizará: o painel geral de serviços AWS e o Elastic Beanstalk, que serão mostrados a seguir. Vá para o painel geral de serviços AWS em:

<https://sa-east-1.console.aws.amazon.com/ec2/v2/home?region=sa-east-1#Home>; e clique em Elastic Beanstalk.



No site AWS Elastic Beanstalk clique em 'Ações' e depois em 'criar ambiente'



Selecione 'Ambiente de Servidor Web':



Coloque um nome de ambiente qualquer:



Criar um ambiente de servidor Web

Inicie um ambiente com um aplicativo de exemplo ou seu próprio código. Ao criar um ambiente, você permite que o AWS Elastic Beanstalk gerencie os recursos e as permissões da AWS em seu nome. [Saiba mais](#)

Informações do ambiente

Escolha o nome, o subdomínio e a descrição para o ambiente. Esses itens não podem ser alterados posteriormente.

Nome do aplicativo movies-app2

Nome do ambiente

Domínio Verificar disponibilidade

Descrição

Em Configuração Básica - Plataforma, selecione Python:

Configuração básica

Plataforma ☒ Plataforma pré-configurada

Plataformas publicadas e mantidas pelo AWS Elastic Beanstalk.

Código do aplicativo ☒

- Choose a platform –

– Choose a platform –

Genérico

Docker

Multi-container Docker

Pré-configurado

Elastic Beanstalk Packer Builder

Go

.NET (Windows/IIS)

Java

Node.js

Ruby

PHP

Python

Tomcat

Pré-configurado - Docker

GlassFish

Go

Python

id para movies-app2

Em 'Código do aplicativo' selecione 'Fazer upload do código' e clique em 'Upload'

Código do aplicativo

☐ Aplicativo de exemplo
Comece imediatamente com o código de exemplo.

☐ Versão existente
As versões do aplicativo que você enviou por upload para **movies-app2**.

– Escolha uma versão –

☒ Fazer upload do código
Faça upload de um pacote de origem do seu computador ou copie um do Amazon S3.

Upload ZIP ou WAR

► Tags de código de aplicativo

Clique em 'selecionar arquivo' e depois em 'fazer upload':

Faça upload do seu código

Faça upload de um pacote de origem do seu computador ou copie um do Amazon S3.

Origem do código-fonte ☒ Arquivo local
(Tamanho máximo 512 MB)

Escolher arquivo Nenhum arquivo selecionado

☐ URL **Nenhum arquivo selecionado**

https://s3.sa-east-1.amazonaws.com

Rótulo da versão **movies-app2-source**
Nome exclusivo desta versão do código do aplicativo.

Cancelar **Fazer upload**

Clique em 'Criar ambiente'

Código do aplicativo

☐ Aplicativo de exemplo
Comece imediatamente com o código de exemplo.

☐ Versão existente
As versões do aplicativo que você enviou por upload para **movies-app2**.

– Escolha uma versão –

☒ Fazer upload do código
Faça upload de um pacote de origem do seu computador ou copie um do Amazon S3.

test-app2-source

► Tags de código de aplicativo

O AWS Elastic iniciará a instalação, após cerca de dois minutos exibirá a seguinte tela com a url da aplicação no canto direito:

aws Serviços Grupos de recursos rodfloripa


Elastic Beanstalk movies-app2 movies1

Todos os aplicativos > movies-app2 > ilovemovies2 (ID do ambiente: e-ujvpqhbych, URL: ilovemovies2.ezp2nffwbs.sa-east-1.elasticbeanstalk.com)


Painel

- Configuração
- Logs
- Integridade
- Monitoramento
- Alertas
- Atualizações gerenciadas
- Eventos

Visão geral

 **Integridade**
OK

Versão em execução
movies-app2-source2



Eventos recentes

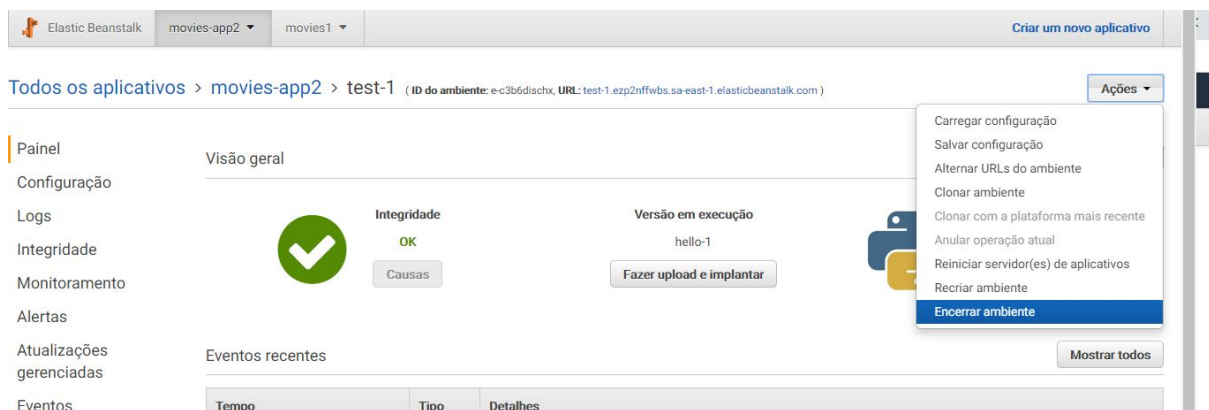
Tempo	Tipo	Detalhes
-------	------	----------

Se a aplicação não estiver funcionando corretamente o log de erros pode ser visto na opção 'Logs' na mesma página:



No caso da aplicação não estar funcionando não esqueça de encerrar o ambiente que foi criado. Para evitar o incômodo de ter que fazer o upload de todo o projeto zipado várias vezes durante o processo de debug de erros existe um post no stackoverflow que apresenta uma solução para este problema: <https://stackoverflow.com/questions/42582384/is-there-a-way-to-push-changed-to-a-aws-beanstalk-instead-of-uploading-an-entire-z>

Depois de arrumar, crie um ambiente com um nome diferente.



Em EC2 Dashboard (<https://sa-east-1.console.aws.amazon.com/ec2/v2/home?region=sa-east-1#Home>) é possível ver as instâncias que estão rodando clicando em 'Running Instances'

