

Mãos à obra: Configurando arquivo para Amazon

Agora que realizamos o teste local, conseguimos verificar que temos o NGINX realizando o cacheamento dos dados da aplicação da Casa do Código, agora devemos montar essa imagem do NGINX para enviar para o Docker Hub, para que depois o *Elastic Beanstalk* consiga pegar essa imagem para executar o container.

Para isso, vá até o [Docker Hub \(https://hub.docker.com/\)](https://hub.docker.com/) e crie um novo repositório chamado **nginx_casadocodigo** clicando no botão **Create Repository** e posteriormente clicando no botão **Create**.

Feito isso, caso esteja no Windows, abra o **Docker Quickstart Terminal** ou o terminal se estiver no Linux ou no Mac e faça o upload da imagem customizada do NGINX para o repositório do Docker Hub. Para isso, é necessário referenciar essa imagem por uma nova **tag** de identificação que deverá seguir a sintaxe **Docker_ID/repositório:[tag]**, coloque o comando:

```
docker tag nginx_casadocodigo:v1-dev [Docker_ID]/nginx_casadocodigo:v1-prod
```

E posteriormente envie essa imagem para o repositório no Docker Hub com o comando:

```
docker push [Docker_ID]/nginx_casadocodigo:v1-prod
```

Agora que já temos a imagem customizada do NGINX no Docker Hub, vamos começar a montar a configuração necessária para levar esse nosso projeto que agora terá dois containers, para o ambiente de produção da Amazon através do serviço do *Elastic Beanstalk*. Assim como fizemos na primeira parte do curso, o *Elastic Beanstalk* espera receber um arquivo de configuração chamado **Dockerrun.aws.json** o qual vemos a sintaxe na documentação fornecida pela Amazon, que já está editada abaixo para você:

```
{
  "AWSEBDockerrunVersion": 2,
  "containerDefinitions": [
    {
      "name": "container_casadocodigo",
      "image": "[Imagem de produção da Casa do Código que está no Docker Hub]"
    }
  ]
}
```

```

    "image": "[Imagem de produção da casa do código que está no Docker Hub]",
    "memory": 256
  },
  {
    "name": "container_nginx",
    "image": "[Imagem de produção do NGINX que está no Docker Hub]",
    "memory": 256,
    "portMappings": [
      {
        "hostPort": 80,
        "containerPort": 80
      }
    ],
    "links": [
      "container_casadocodigo"
    ]
  }
]
}

```

OBS: Cuidado com os nomes dos containers, o arquivo de configuração do NGINX está configurado para redirecionar as requisições para o container com nome **container_casadocodigo** que havíamos especificado através do comando `proxy_pass http://container_casadocodigo:8080`

Agora que temos o arquivo de configuração **Dockerrun.aws.json**, salve esse arquivo dentro de uma pasta com o nome de sua preferência, por exemplo **multicontainers** dentro da pasta **Beanstalk_producao**. Feito isso, precisamos realizar a configuração do ambiente no *Elastic Beanstalk* para que seja possível trabalhar com múltiplos containers Docker. Faremos isso na próxima etapa.