

Agora que fizemos as alterações no projeto da Casa do Código para que a sessão do usuário seja armazenada no Redis, precisamos levar nossa aplicação para a Amazon. O primeiro passo que devemos alterar no projeto é utilizar o Endpoint do Redis para realizar a conexão no método **JedisConnectionFactory** da classe **RedisConfiguration**

@Bean

```
public JedisConnectionFactory jedisConnectionFactory() {  
    JedisPoolConfig pool = new JedisPoolConfig();  
    JedisConnectionFactory factory = new JedisConnectionFactory(pool);  
    factory.setHostName("[ENDPOINT REDIS]");  
    factory.setPort(6379);  
    return factory;  
}
```

COPIAR CÓDIGO

Agora como vamos levar o projeto para a Amazon devemos utilizar o banco de dados de produção, para isso vá até a classe **ServletSpringMVC** e no método **onStartup** coloque o perfil de produção:

@Override

```
public void onStartup(ServletContext servletContext) throws ServletException {  
    super.onStartup(servletContext);  
    servletContext.setInitParameter("spring.profiles.active", "prod");  
}
```

COPIAR CÓDIGO

Uma vez que fizemos esses ajustes, devemos realizar a exportação para o arquivo **.war**. Vamos salvá-lo no mesmo diretório onde salvamos a chave para realizarmos a autenticação com os servidores na Amazon (Caso precise, remova o arquivo **.war** salvo previamente). Feito isso, vamos utilizar o protocolo secure copy (scp) para enviar esse arquivo para os servidores da Amazon. Coloque no terminal:

```
scp -i [chave de autenticação] [arquivo.war] [usuário@DNS público do
```

COPIAR CÓDIGO

Faça esse processo para os dois servidores que temos rodando na Amazon. Feito isso, conecte remotamente com os servidores com o Putty se estiver no Windows ou diretamente no terminal se estiver no Linux ou no Mac. Vamos parar o Tomcat para que façamos as alterações:

```
sudo service tomcat8 stop
```

[COPIAR CÓDIGO](#)

Uma vez conectado nos servidores altere a permissão para que posteriormente o Tomcat consiga descompactar o arquivo:

```
chmod 644 [aplicacao.war]
```

[COPIAR CÓDIGO](#)

Na sequência vá até o diretório **webapps** e remova o projeto existente previamente. Coloque:

```
cd /var/lib/tomcat8/webapps  
sudo rm -r [aplicacao.war]  
sudo rm -r [aplicacao]
```

[COPIAR CÓDIGO](#)

E para finalizar leve a aplicação para a pasta webapps:

```
sudo mv ~/[aplicacao.war] .
```

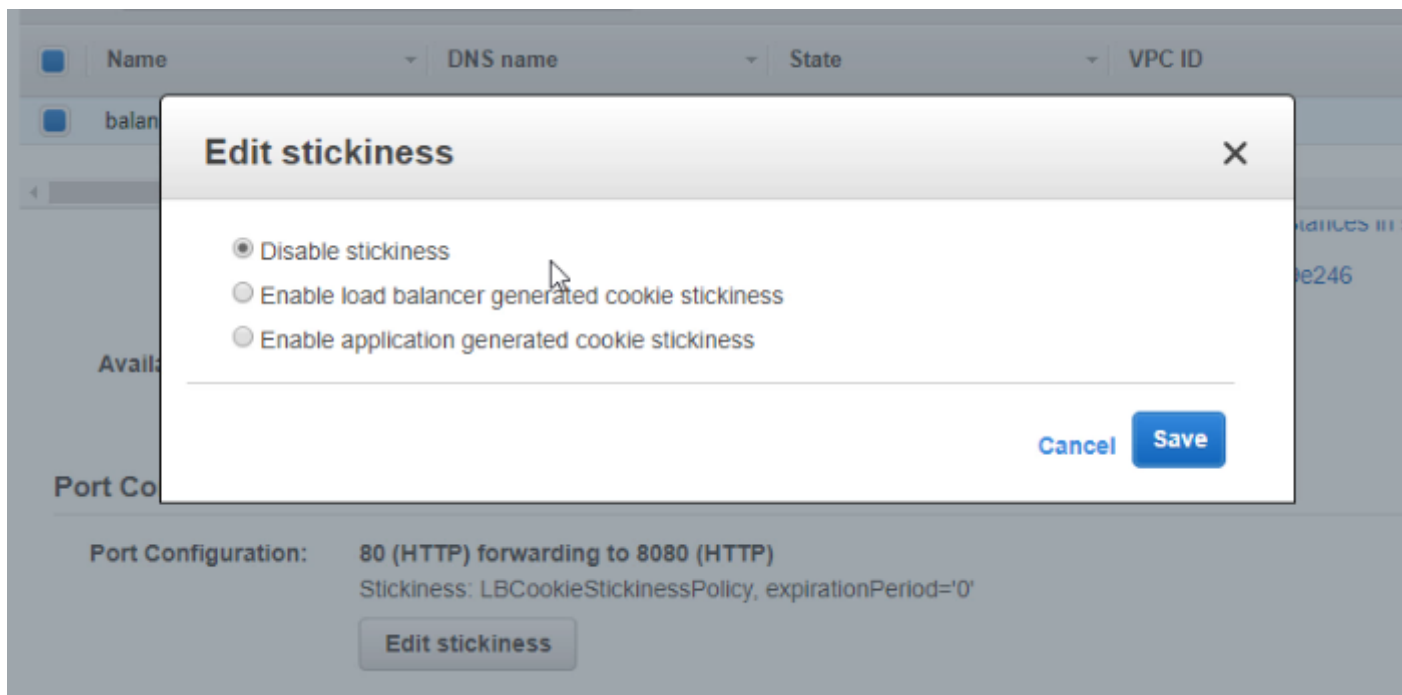
[COPIAR CÓDIGO](#)

Para inicializar o Tomcat coloque:

```
sudo service tomcat8 start
```

[COPIAR CÓDIGO](#)

Uma vez que fizemos essas mudanças, confirme que tais alterações foram refletidas e o balanceador apresente como status as 2 instâncias ativas. Na sequência, desative a opção **Sticky Session** clicando no botão **Edit Stickiness** e posteriormente **Disable stickiness**



Posteriormente acesse a aplicação e coloque um livro no carrinho de compras, qual é o resultado? Posteriormente acesse o endereço **loadbalancer** e confirme que o balanceamento continua sendo feito entre as duas instâncias.