

FIAP GRADUAÇÃO

TECNOLOGIA EM DESENVOLVIMENTO DE SISTEMAS

DevOps Tools & Cloud Computing

Instalar Plug-in/Extension da Azure e Realizar Testes no Eclipse e VSC

PROF. João Menk profjoao.menk@fiap.com.br

PROF. Sálvio Padlipskas salvio@fiap.com.br

PROF. Antonio Figueiredo profantonio.figueiredo@fiap.com.br

PROF. Marcus Leite profmarcus.leite@fiap.com.br

PROF. Thiago Rocha profthiago.rocha@fiap.com.br

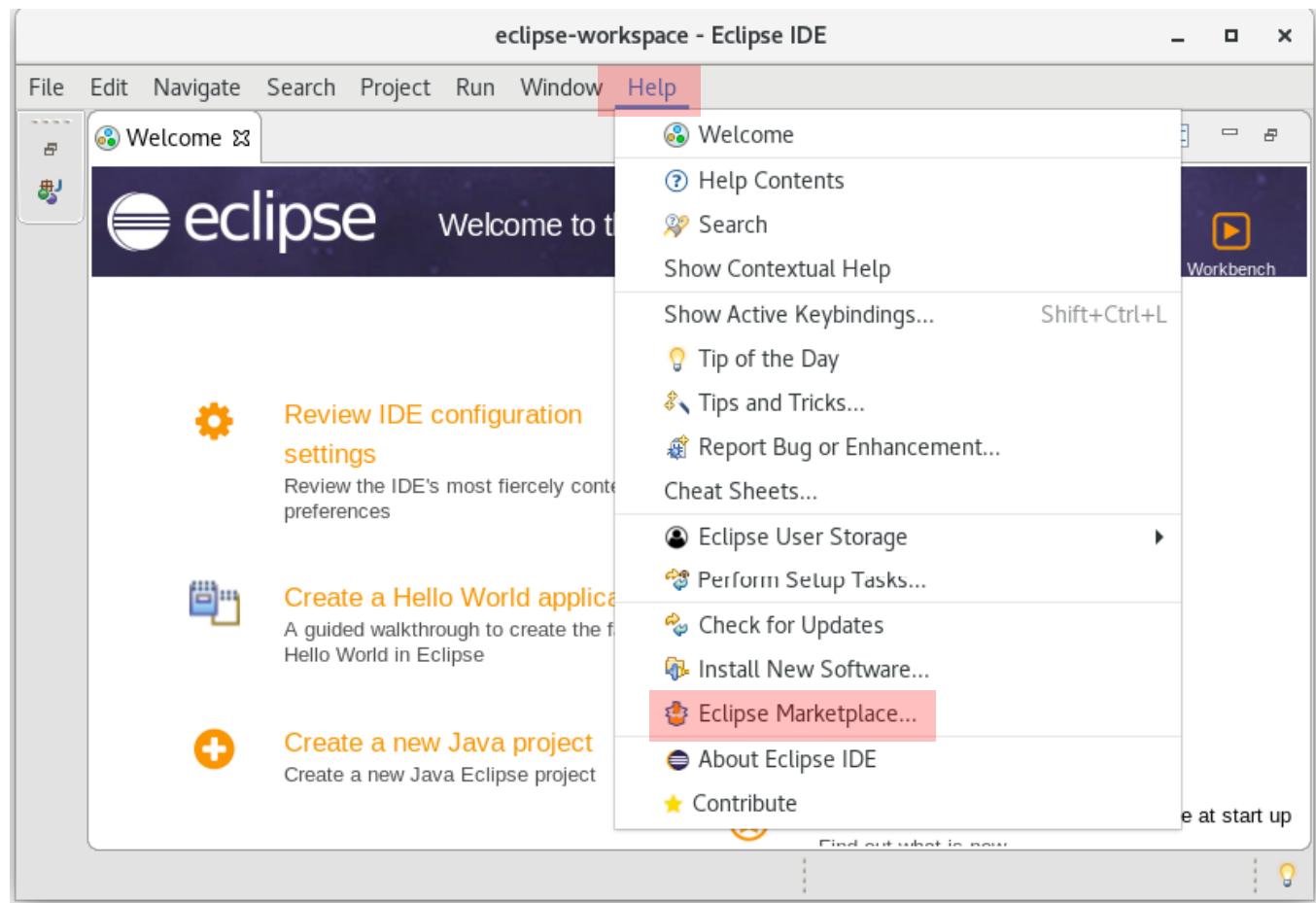


Para podermos realizar um Deploy em um Serviço de Aplicativo na Azure (Web App) através do Eclipse, precisamos realizar o Download do Plugin no Marketplace

01) Abra o Eclipse



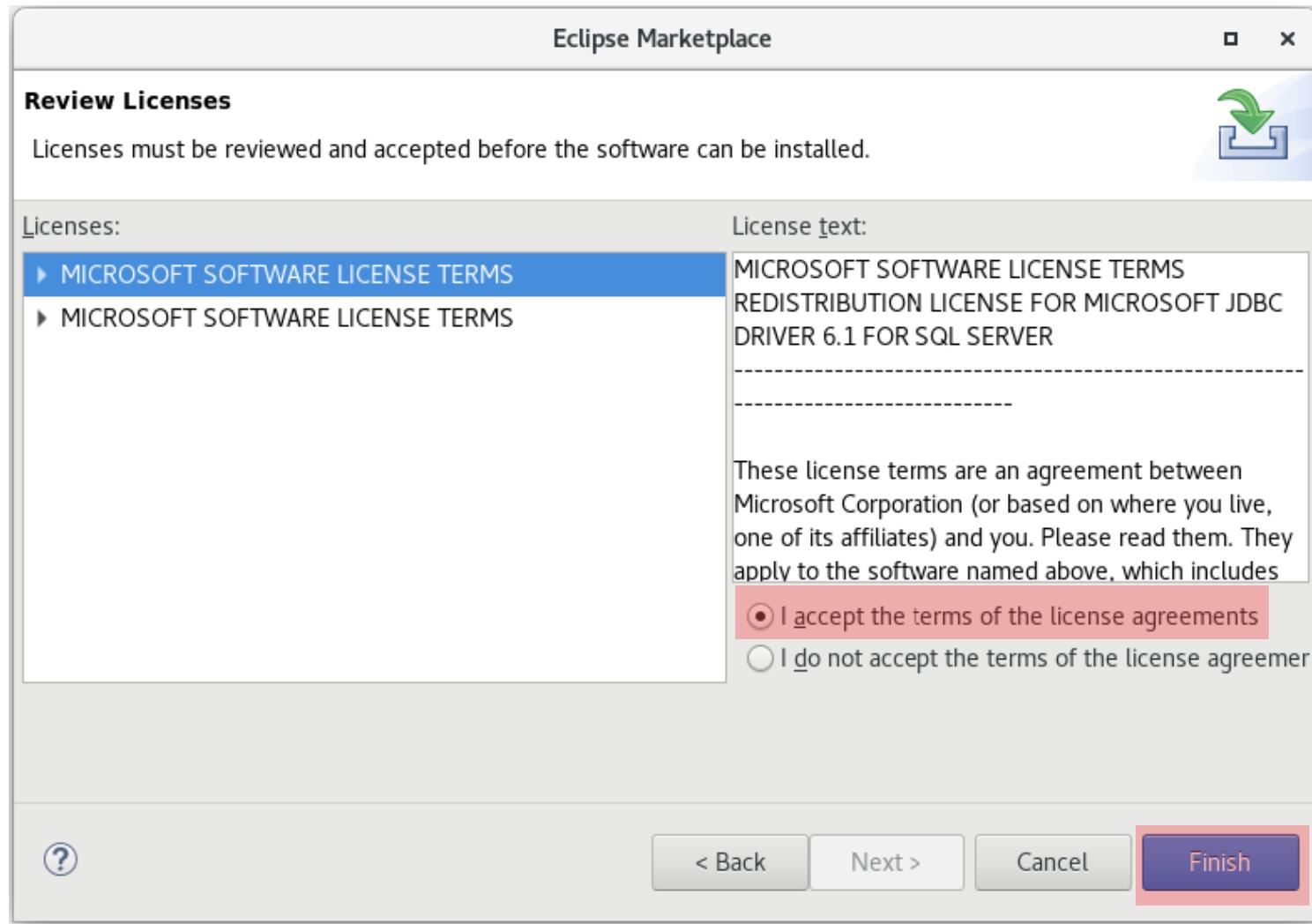
02) No Menu do Eclipse, clique em **Help** e depois em **Eclipse Marketplace...**



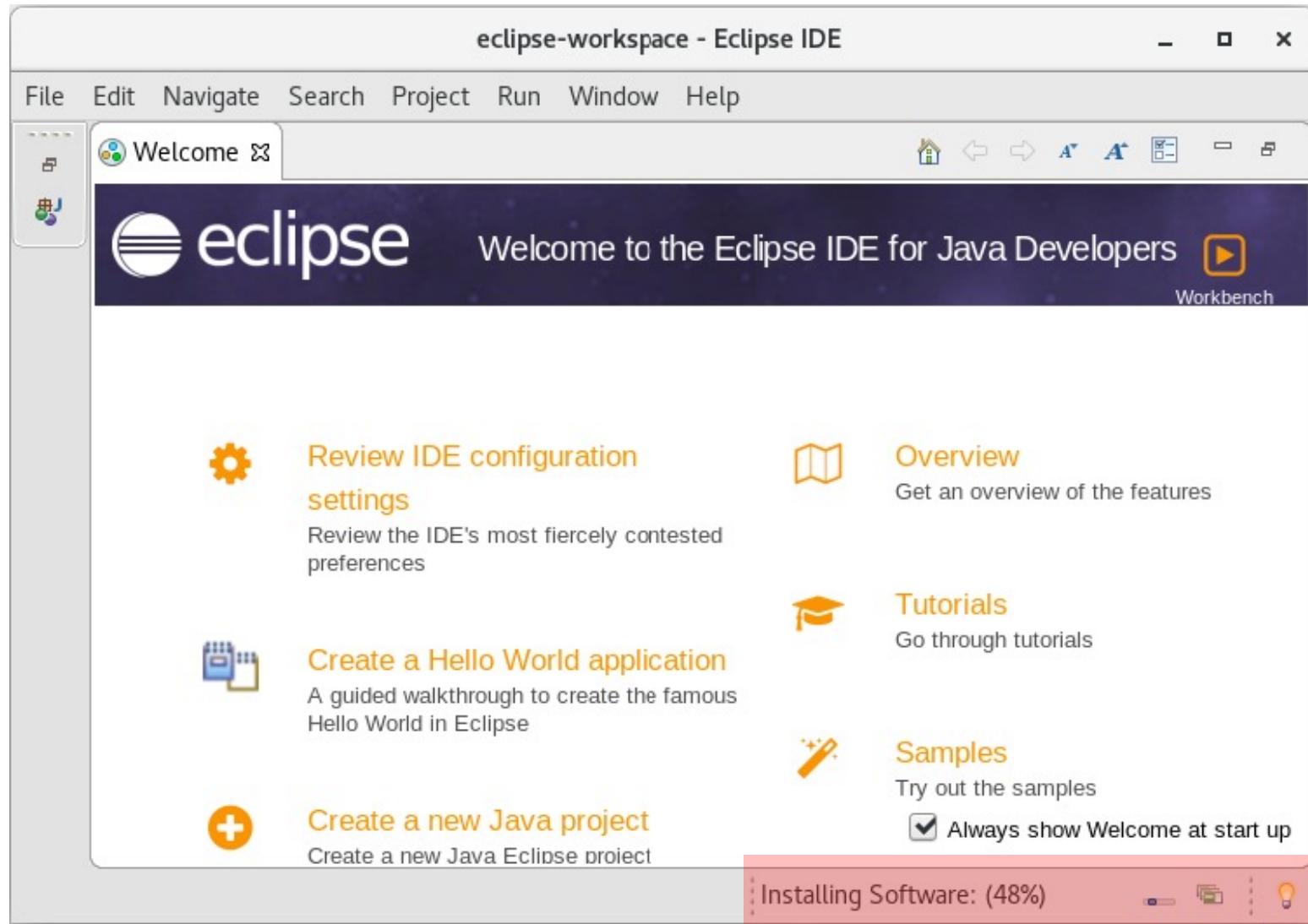
03) No campo de busca digite: **azure**, clique no botão **Go**, escolha o Plugin abaixo e clique em **Install**

The screenshot shows the Eclipse Marketplace window. At the top, it says "Eclipse Marketplace". Below that, there's a search bar with the text "Search Recent Popular Favorites Installed" and a search icon. A red box highlights the search bar with the text "Find: azure". To the right of the search bar are buttons for "All Markets" and "All Categories", and a red "Go" button. Below the search bar, a pink box highlights the "Azure Toolkit for Eclipse 3.31.0" entry. It features a large blue letter "A" icon, the text "The Azure Toolkit for Eclipse provides functionality that allow you to easily create, develop, configure, test, and deploy lightweight, highly available and... [more info](#)", and "by Microsoft Corp., Other". Below this, it lists categories: "Cloud Azure microsoft java WebApp Development". Underneath the entry, there are two grey buttons: one with a star and the number "61" and another with a circular arrow. To the right of these is a red "Install" button. At the bottom of the window, there's a "Marketplaces" section with three icons: a blue square with white lines, a blue hexagon with white dots, and a red circle with a white "M". At the very bottom, there are four buttons: a question mark icon, "[? < Back](#)", "[Install Now >](#)", "[Cancel](#)", and "[Finish](#)".

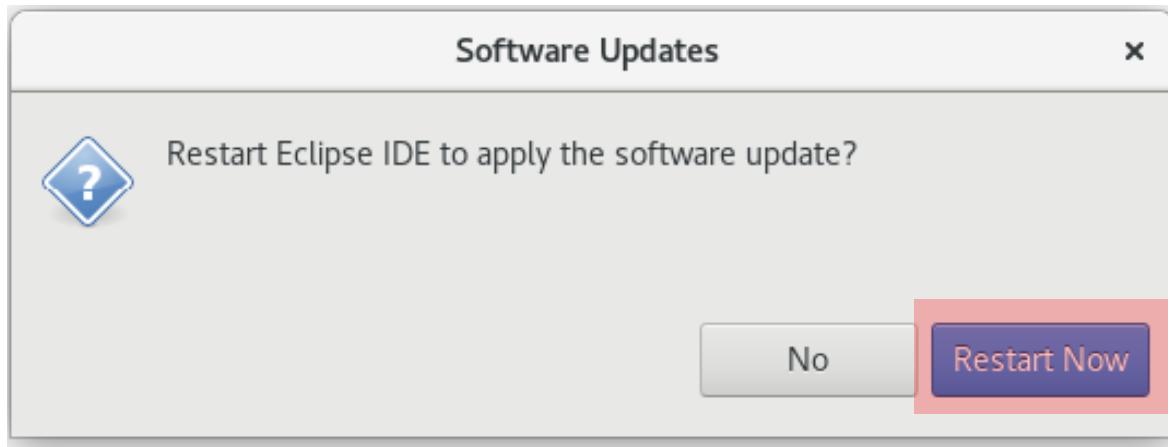
04) Após o Download a seguinte tela será exibida. Clique na opção abaixo para **aceitar os termos de uso** e depois em **Finish**



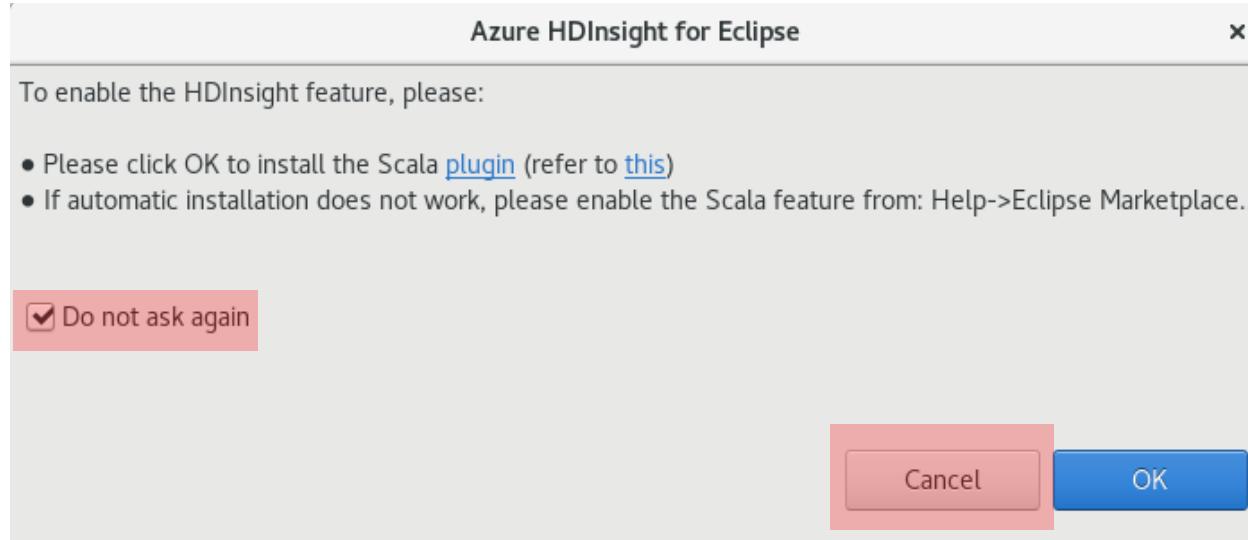
05) Aguarde a instalação do Plugin



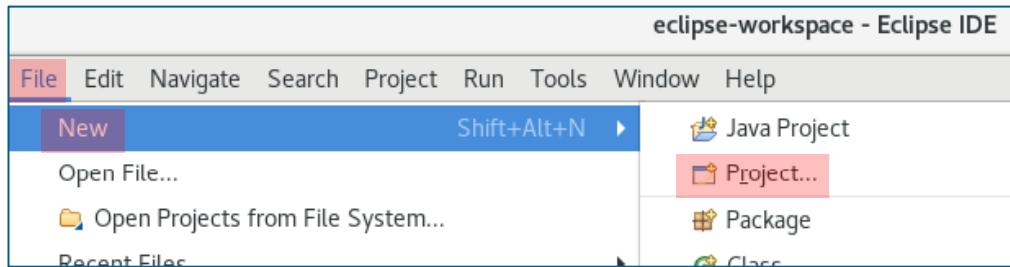
06) Reinicie o Eclipse



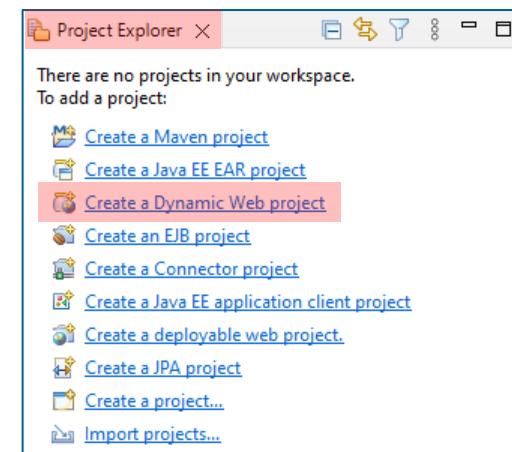
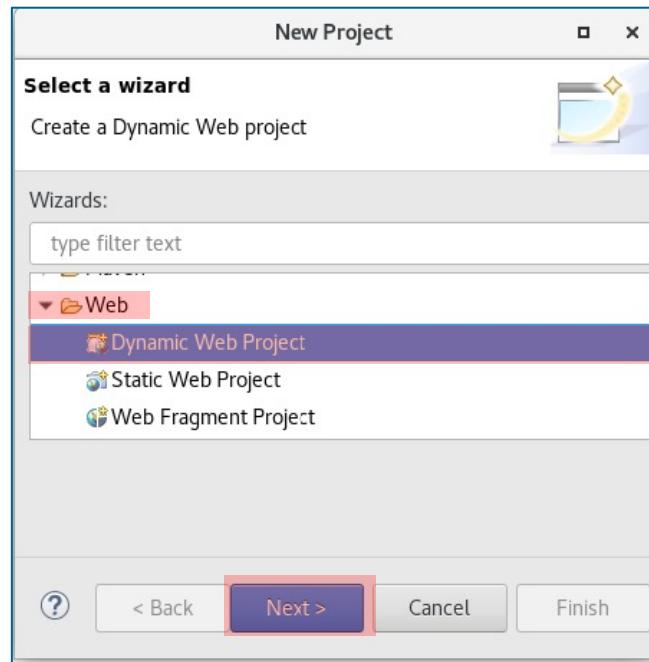
07) Nesse momento podemos ignorar o **HDInsight for Eclipse**



08) No menu do Eclipse escolha o caminho: **File -> New -> Project**

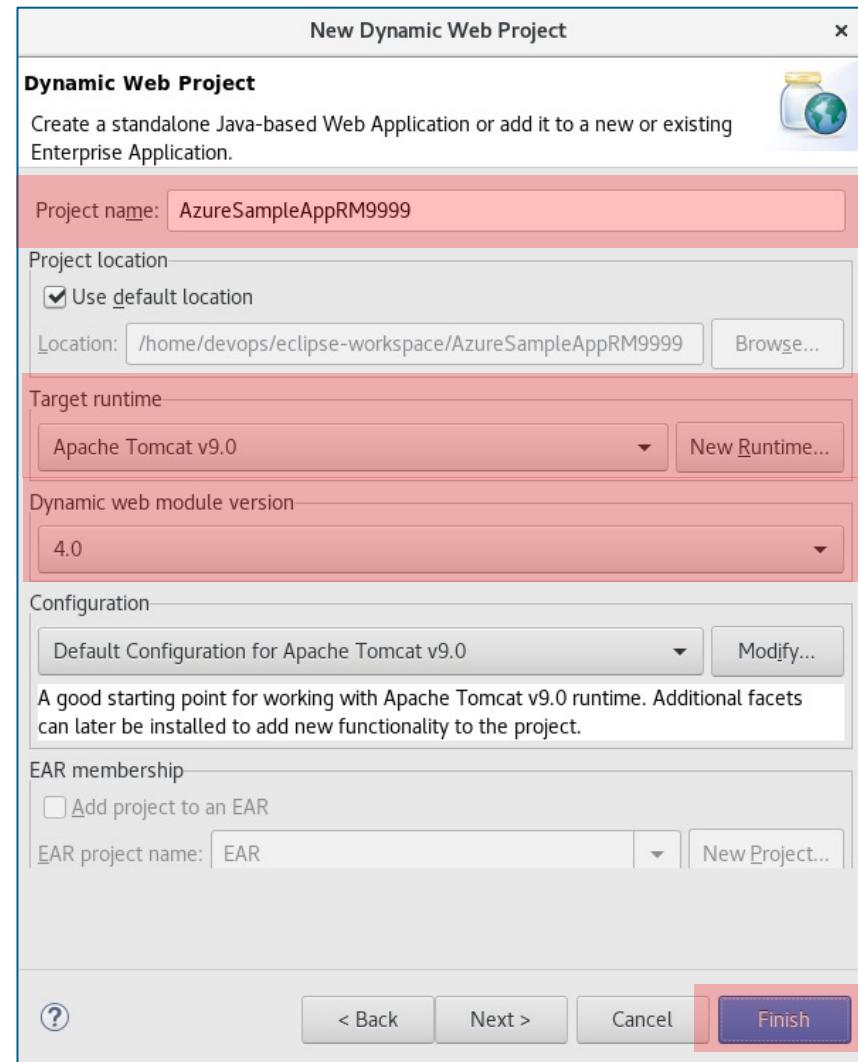


09) Na nova janela aberta escolha **Web** e **Dynamic Web Project**

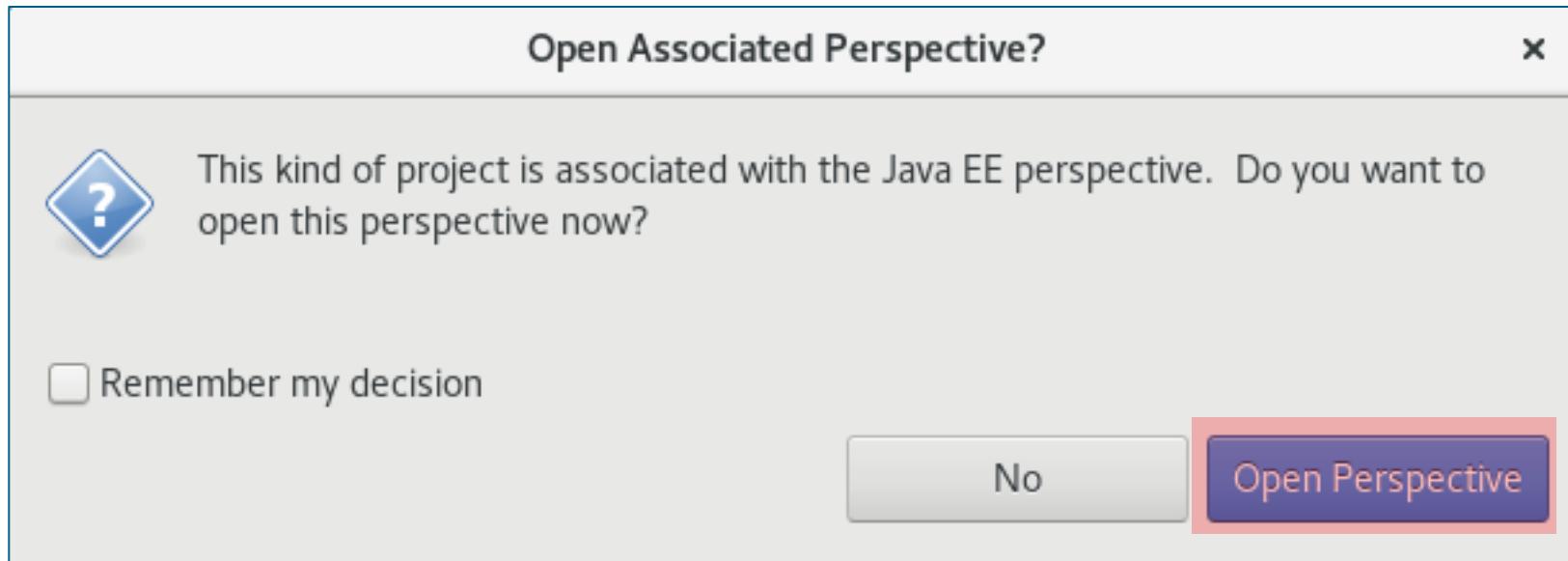


Instalar Plug-in Azure no Eclipse e Realizar testes

10) Informe o nome do Projeto (altere o nome com seu RM), escolha a versão 3.1 para Dynamic web e clique no botão **Finish**



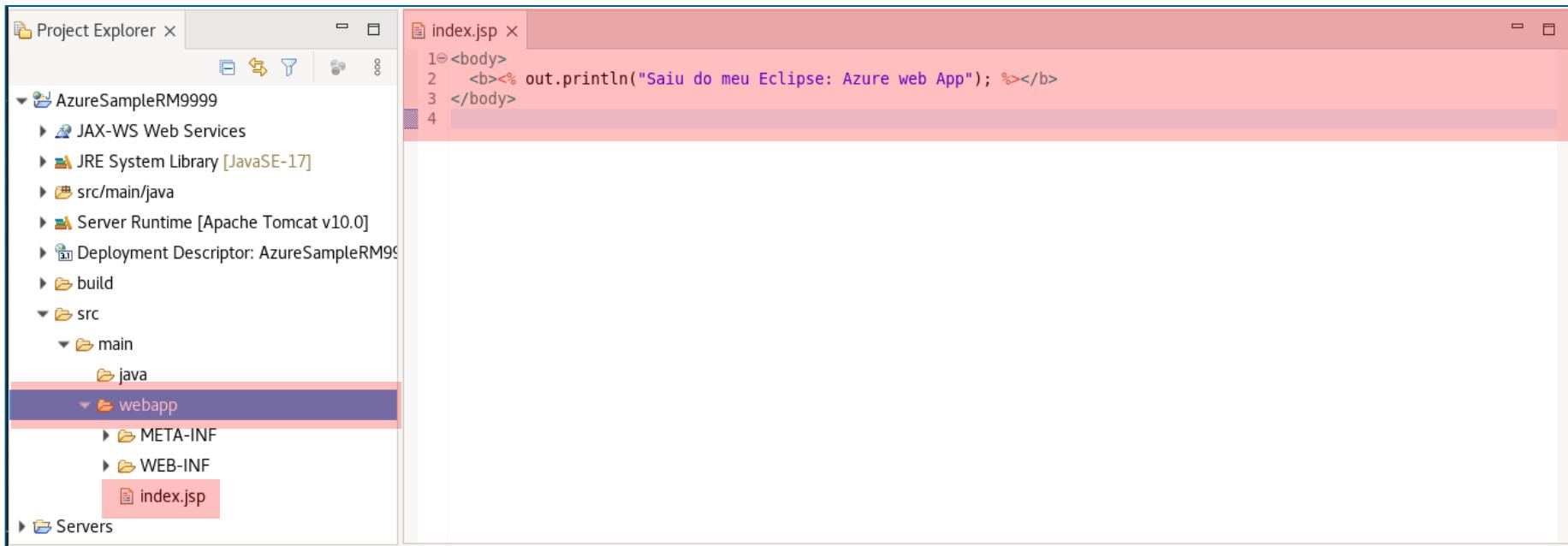
11) Abra a Perspectiva do Java EE se for necessário



12) Expanda a árvore de diretórios do projeto e acesse a pasta **webapp**. Crie um arquivo com o nome **index.jsp** (file -> new -> JSP file). Abra o arquivo e cole o conteúdo abaixo

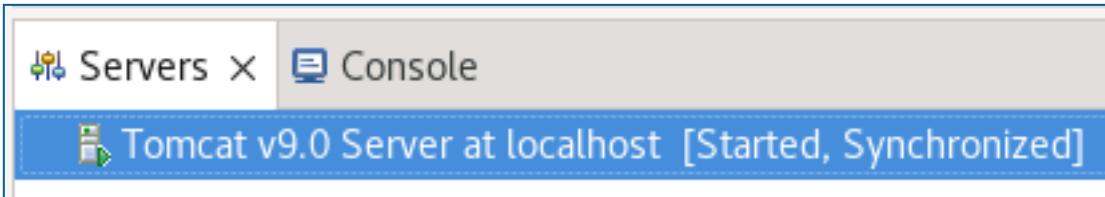
```
<body>
<b><% out.println("Saiu do meu Eclipse: Azure web App"); %></b>
</body>
```

Salve seu Arquivo

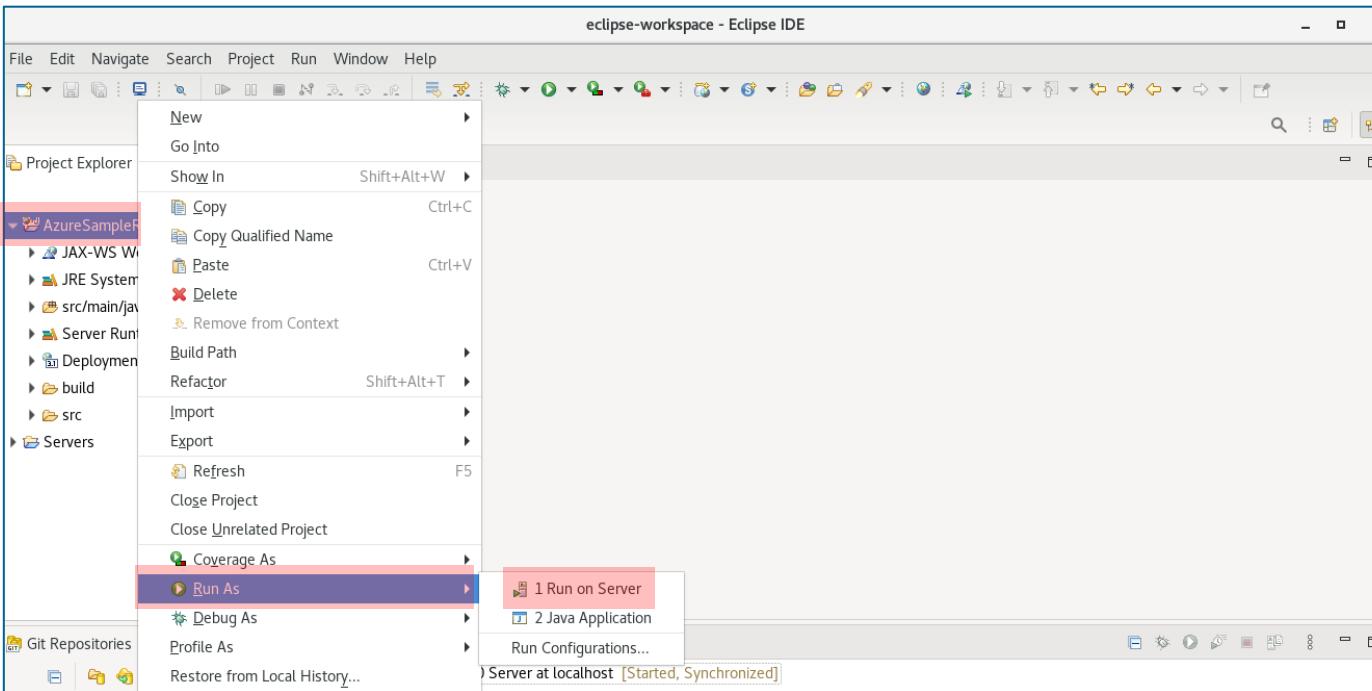


13) Vamos rodar esse projeto localmente antes de realizar um Deploy em Nuvem

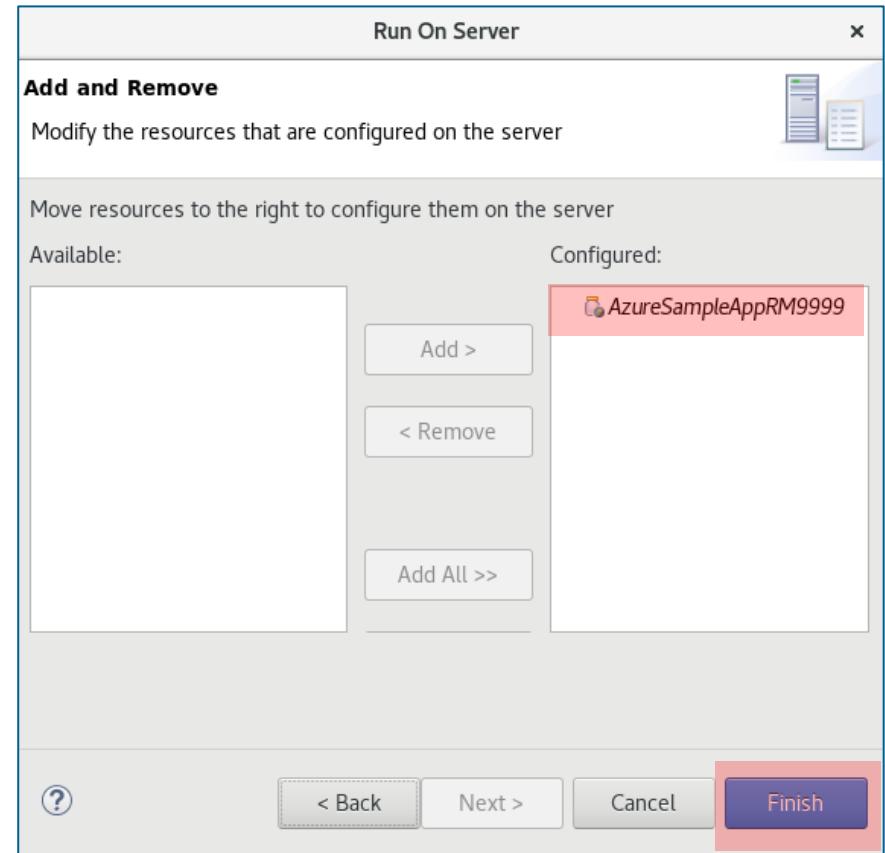
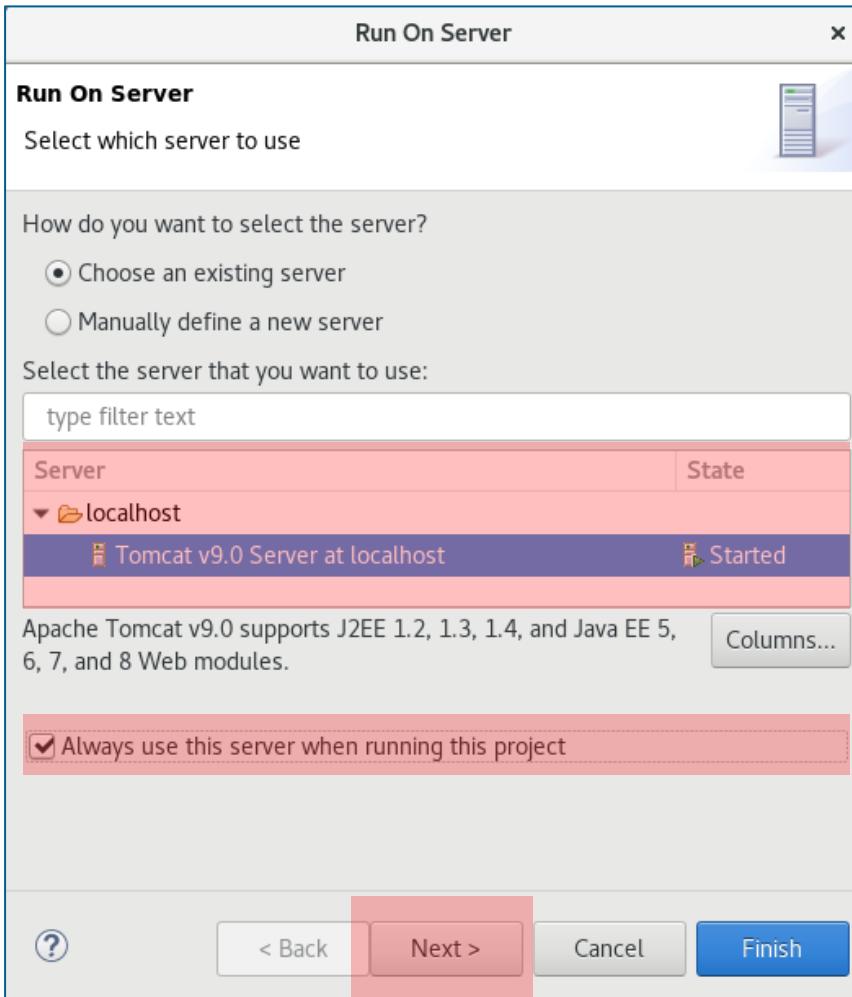
Primeiro, se o Tomcat não estiver iniciado, inicie



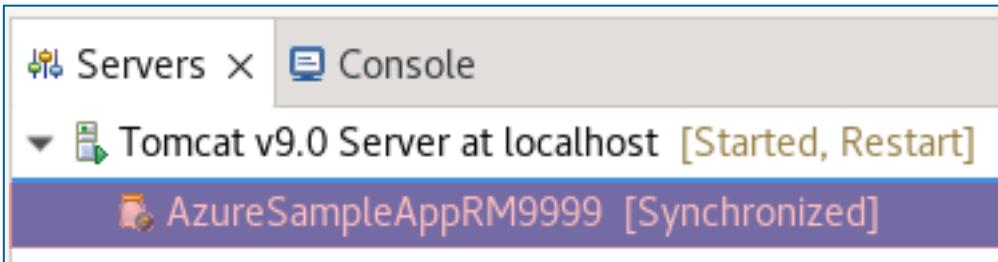
Agora, clique com o botão direito do mouse no Projeto e execute no Tomcat



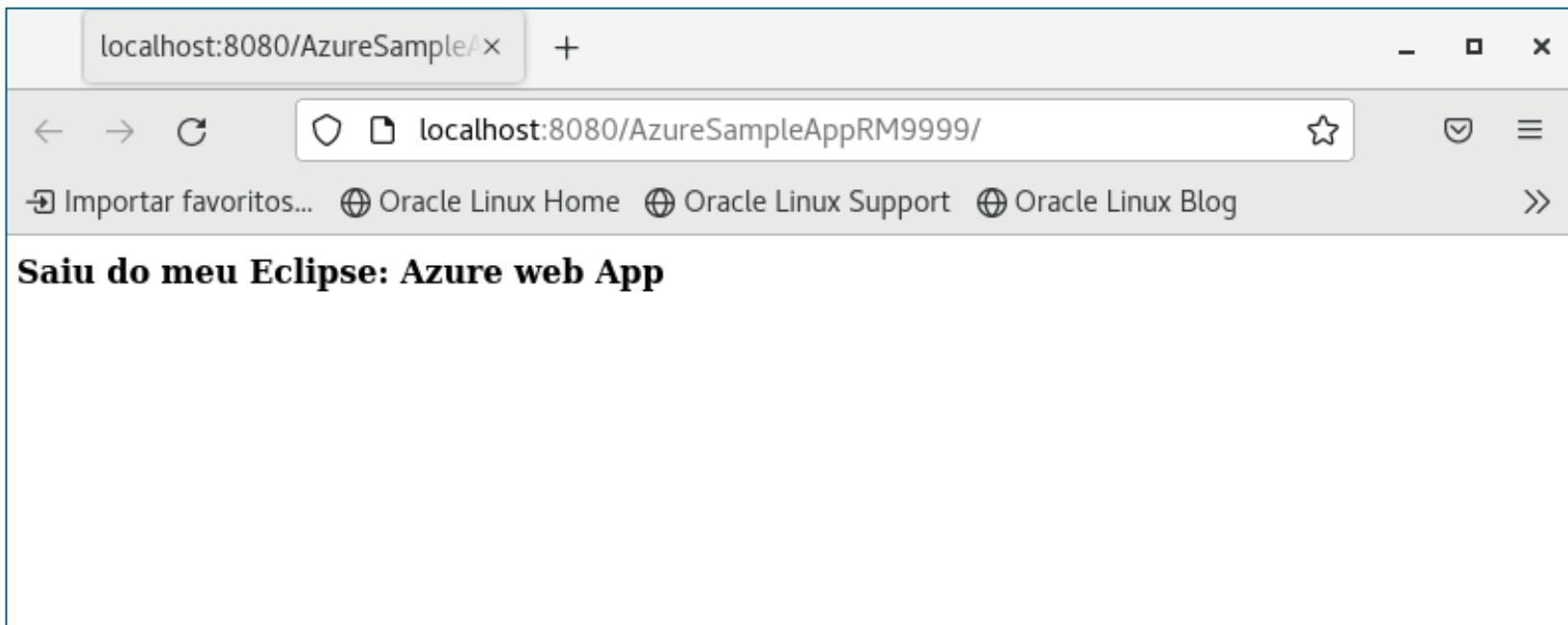
13) Escolha o Servidor que o Projeto irá ser rodar. Em nosso caso Tomcat 9



13) Projeto no Tomcat 9



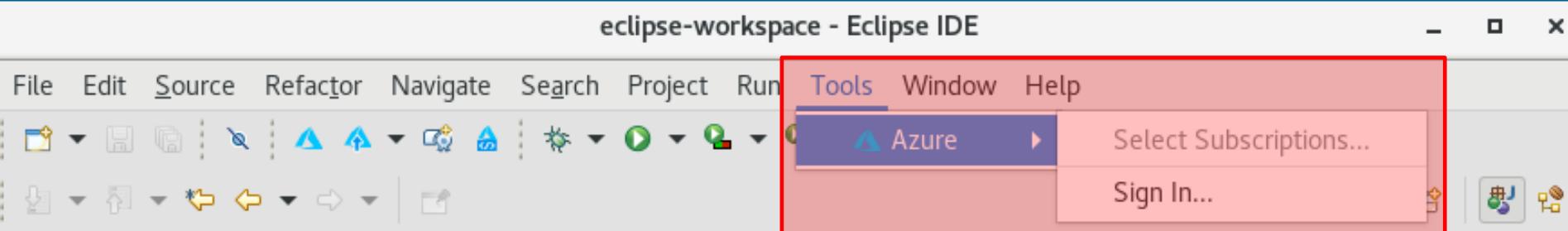
Teste no endereço **localhost:8080/AzureSampleAppRM9999**



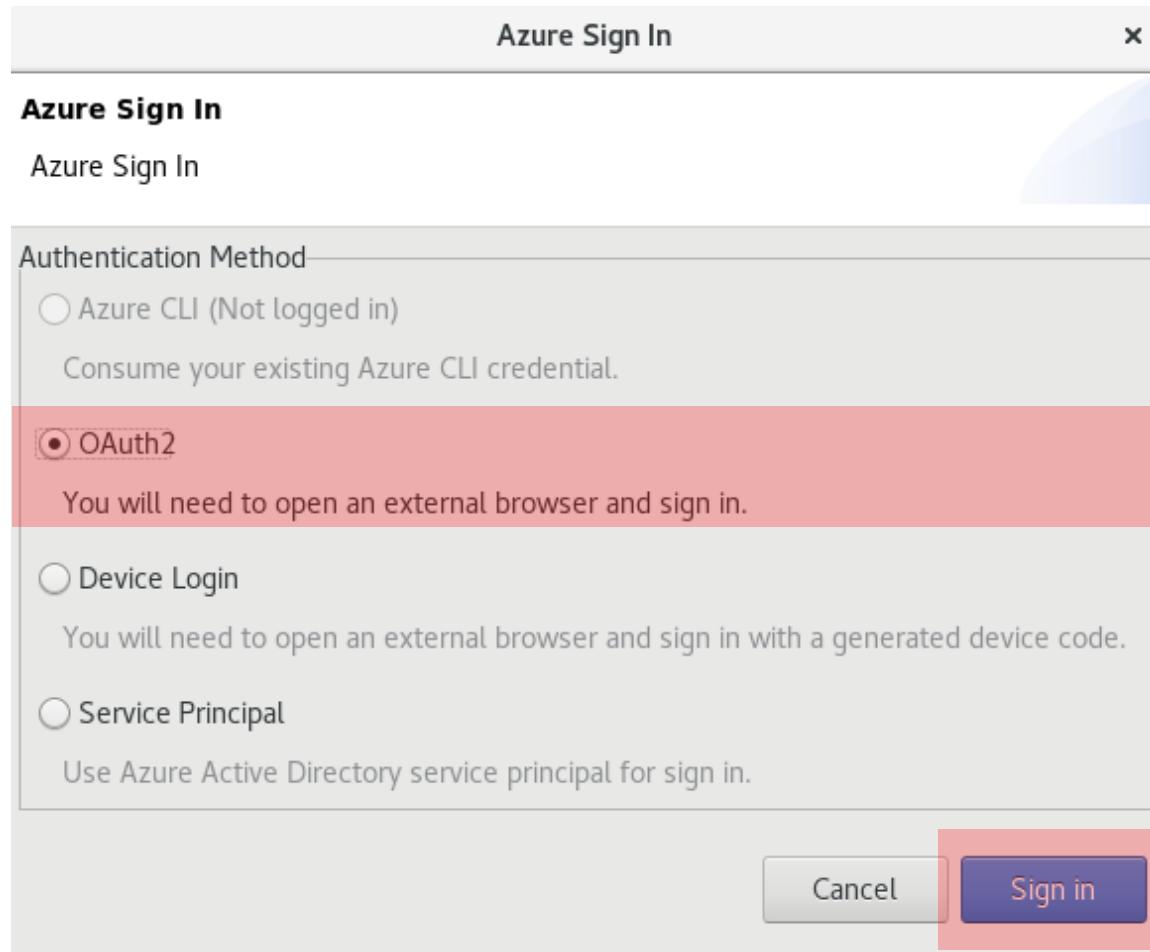
14) Note que o **Plugin do Azure** criou alguns atalhos na barra de ações do Eclipse



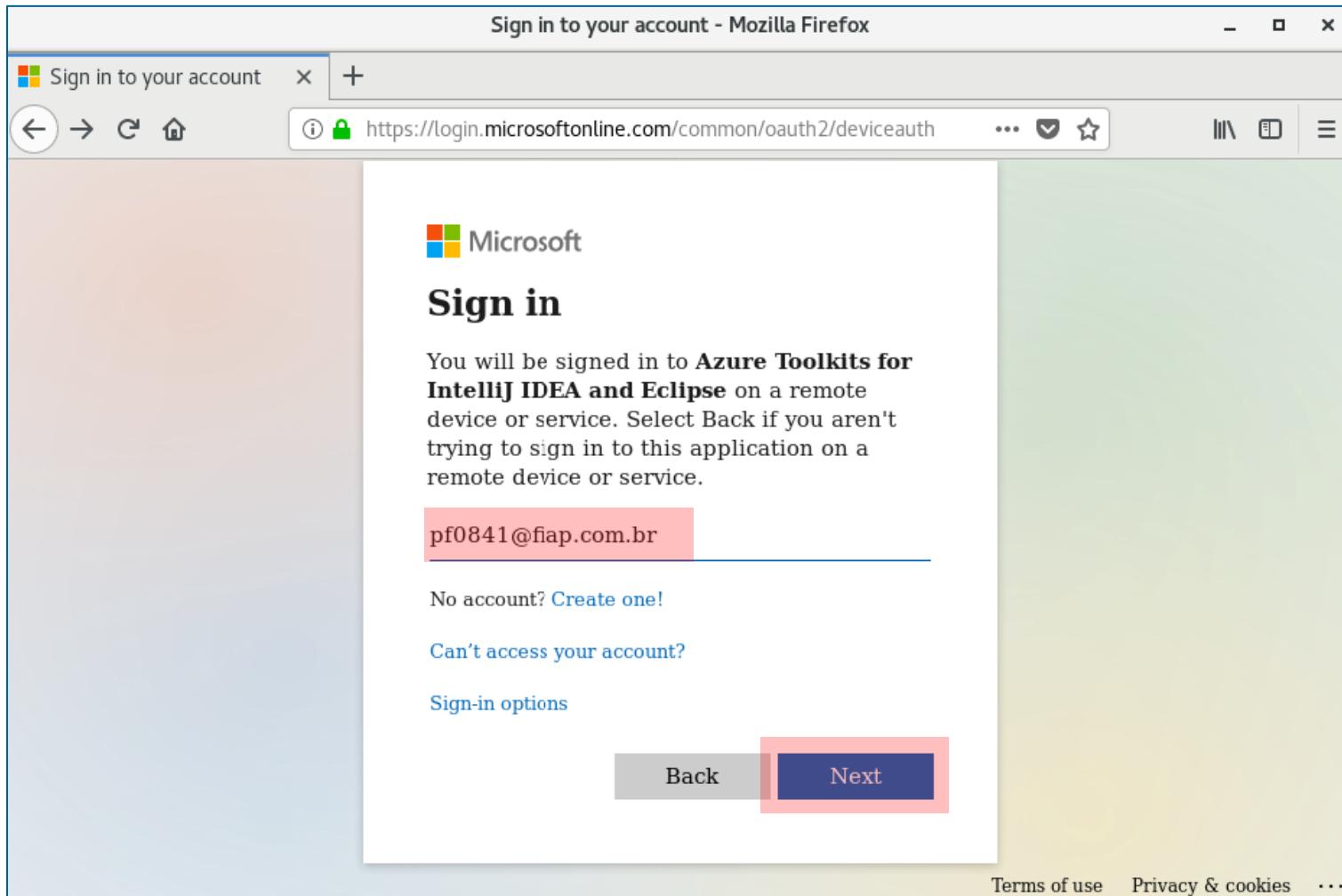
14) No meu **Tools -> Azure**, clique em **Sign in...**



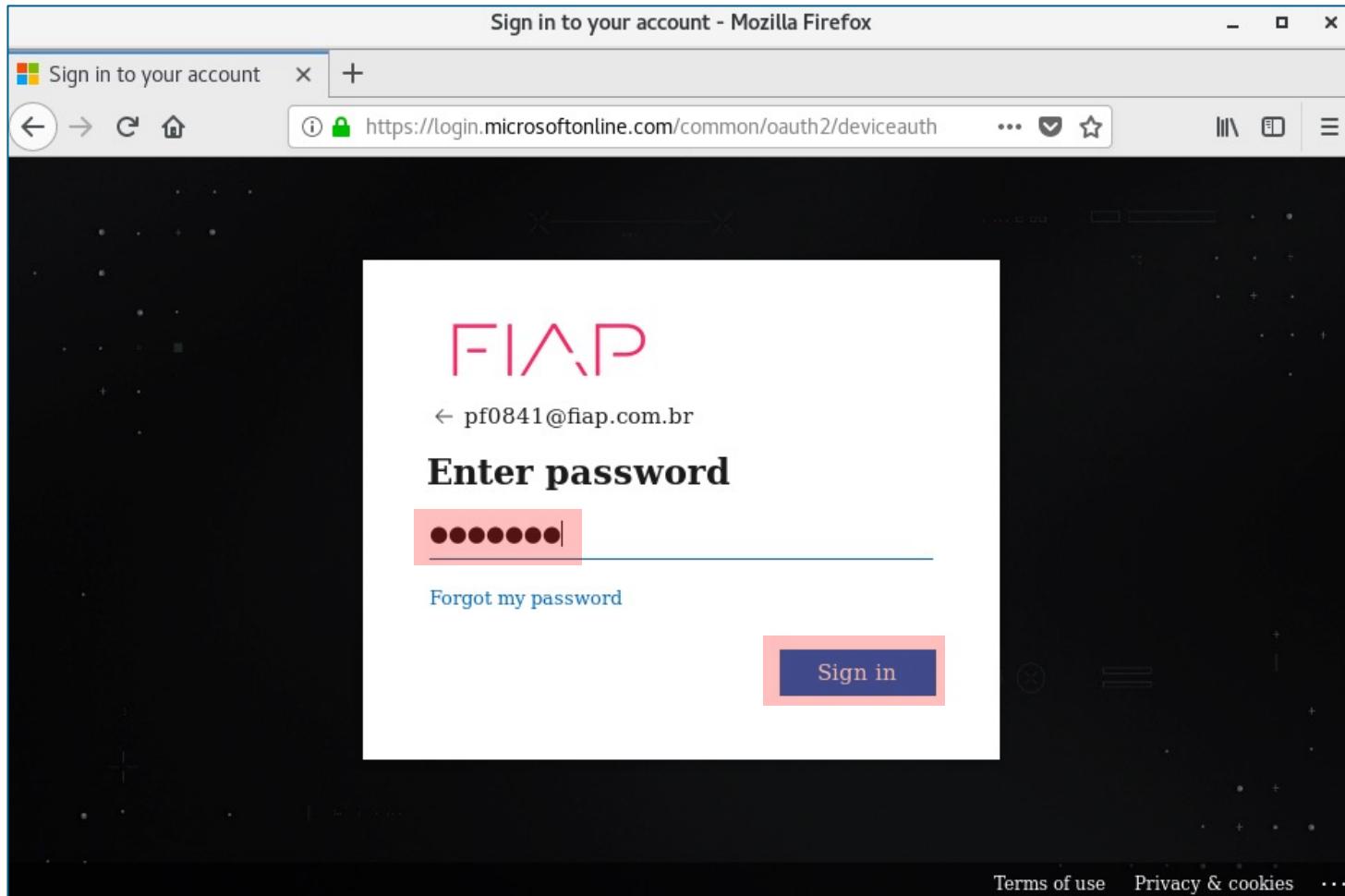
15) Escolha realizar o Login através do Browser (OAuth2)



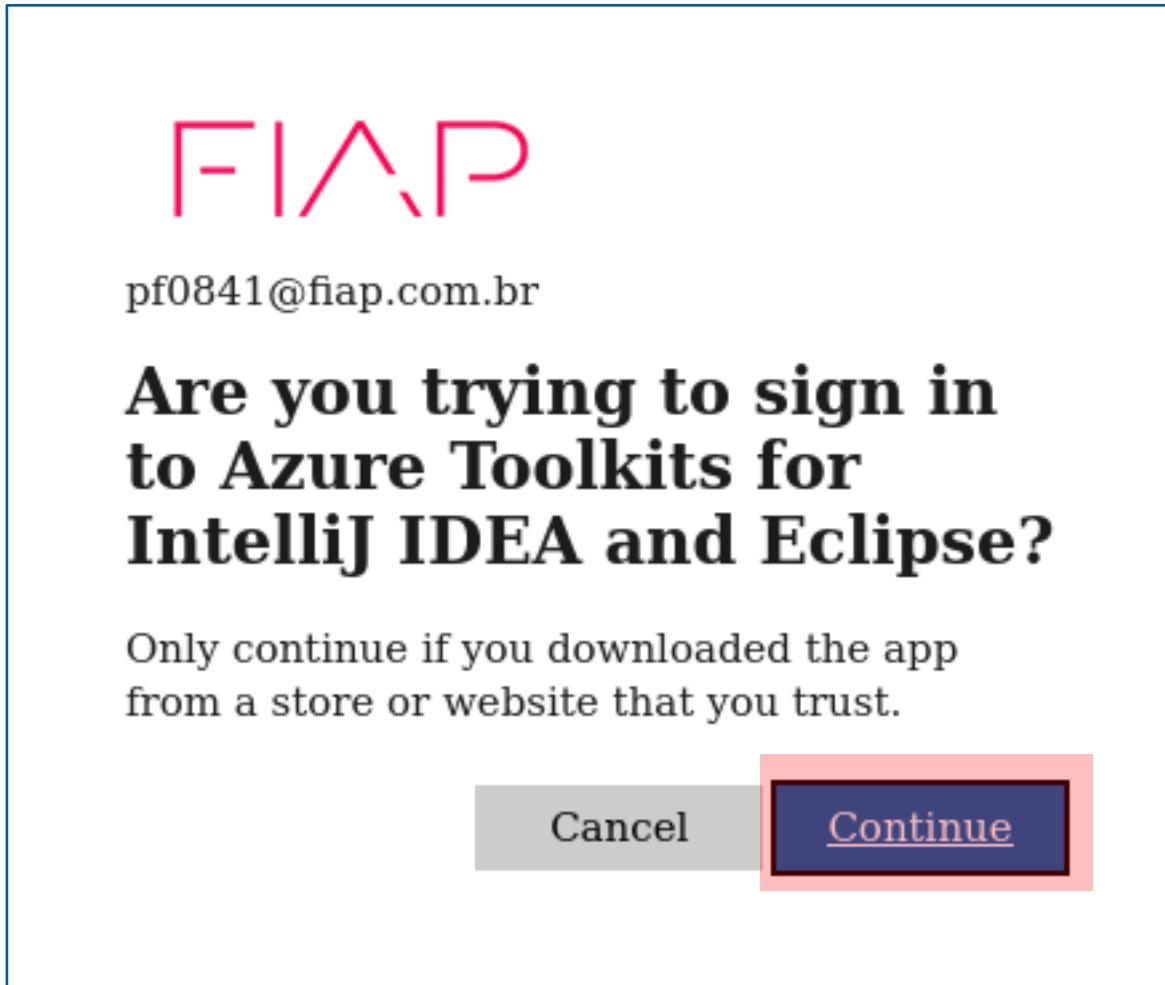
16) Realize sua autenticação



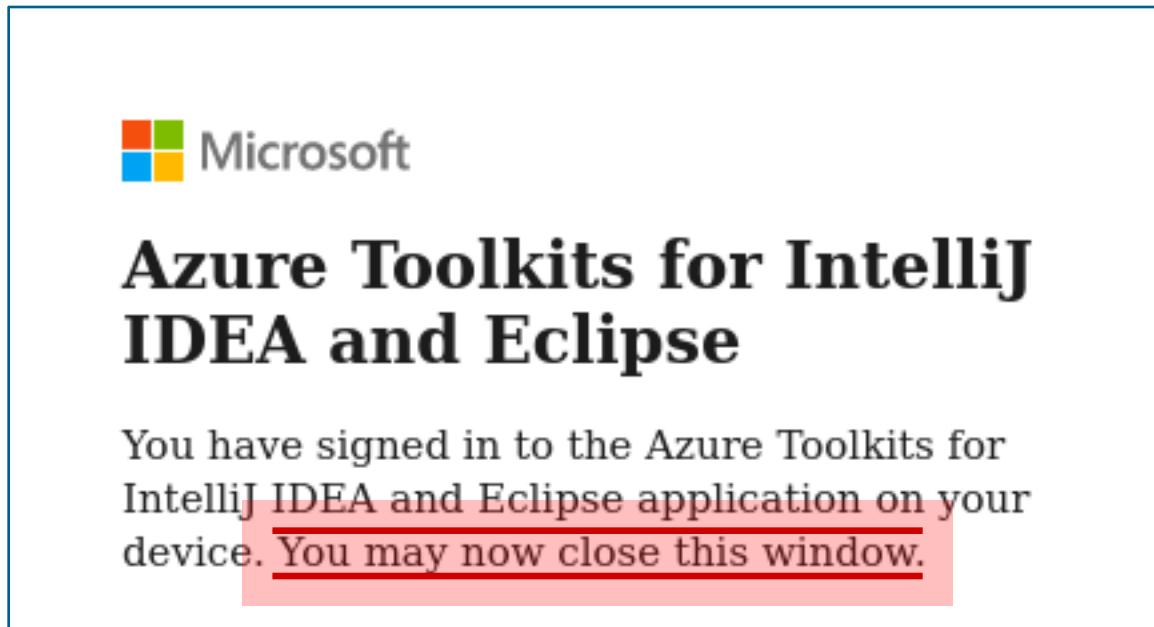
16) Realize sua autenticação



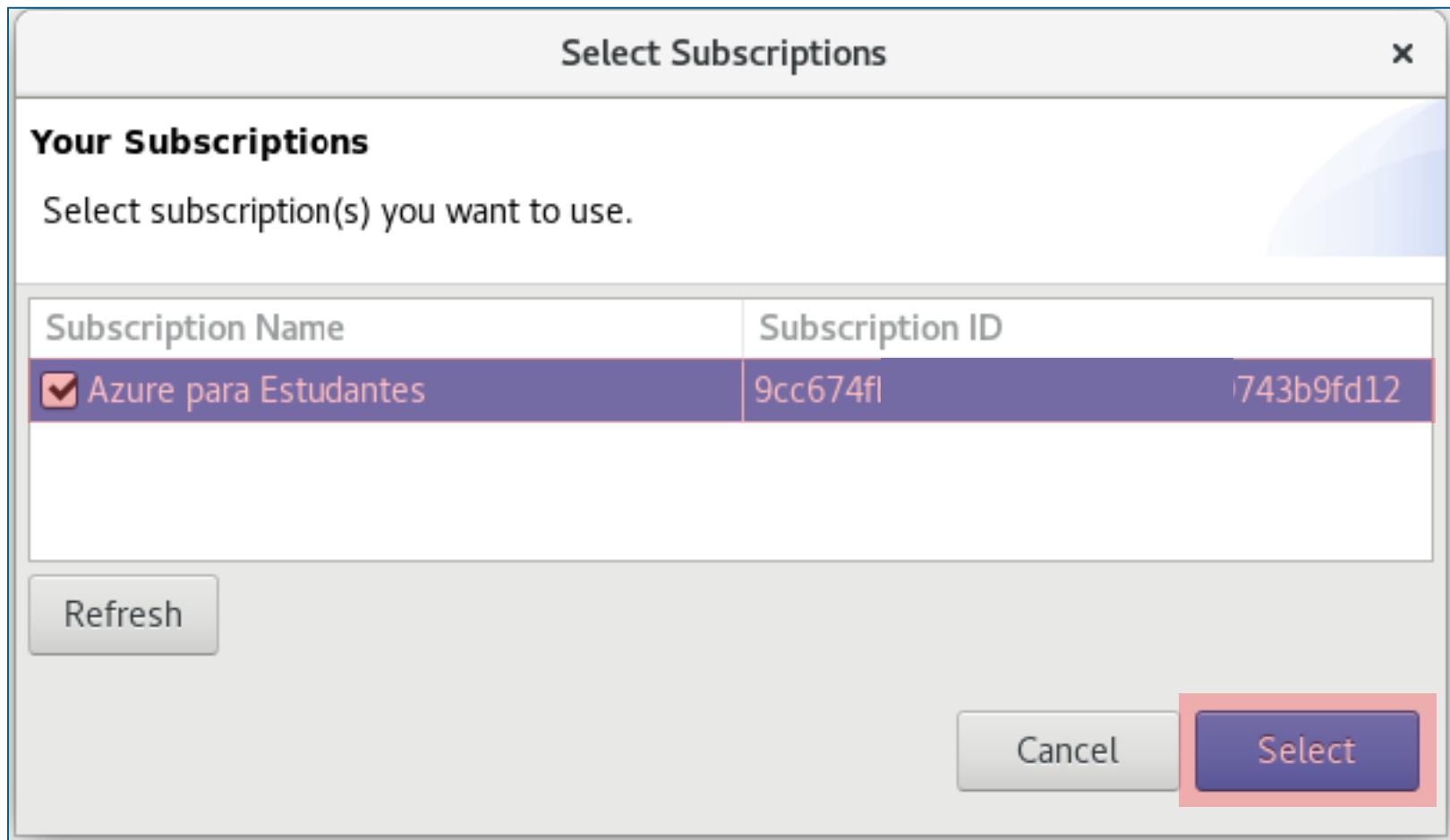
16) Realize sua autenticação



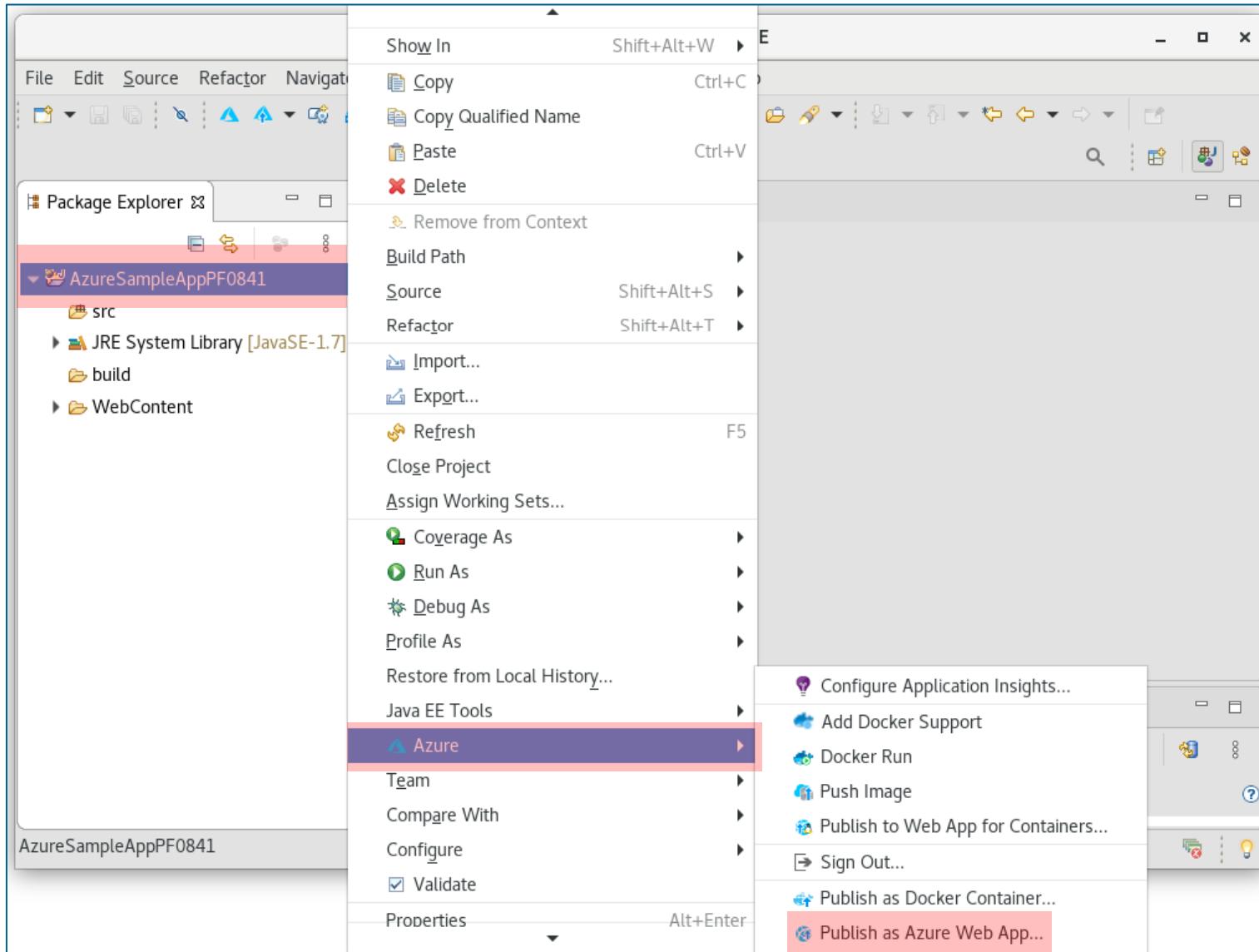
16) Realize sua autenticação



17) Escolha sua assinatura

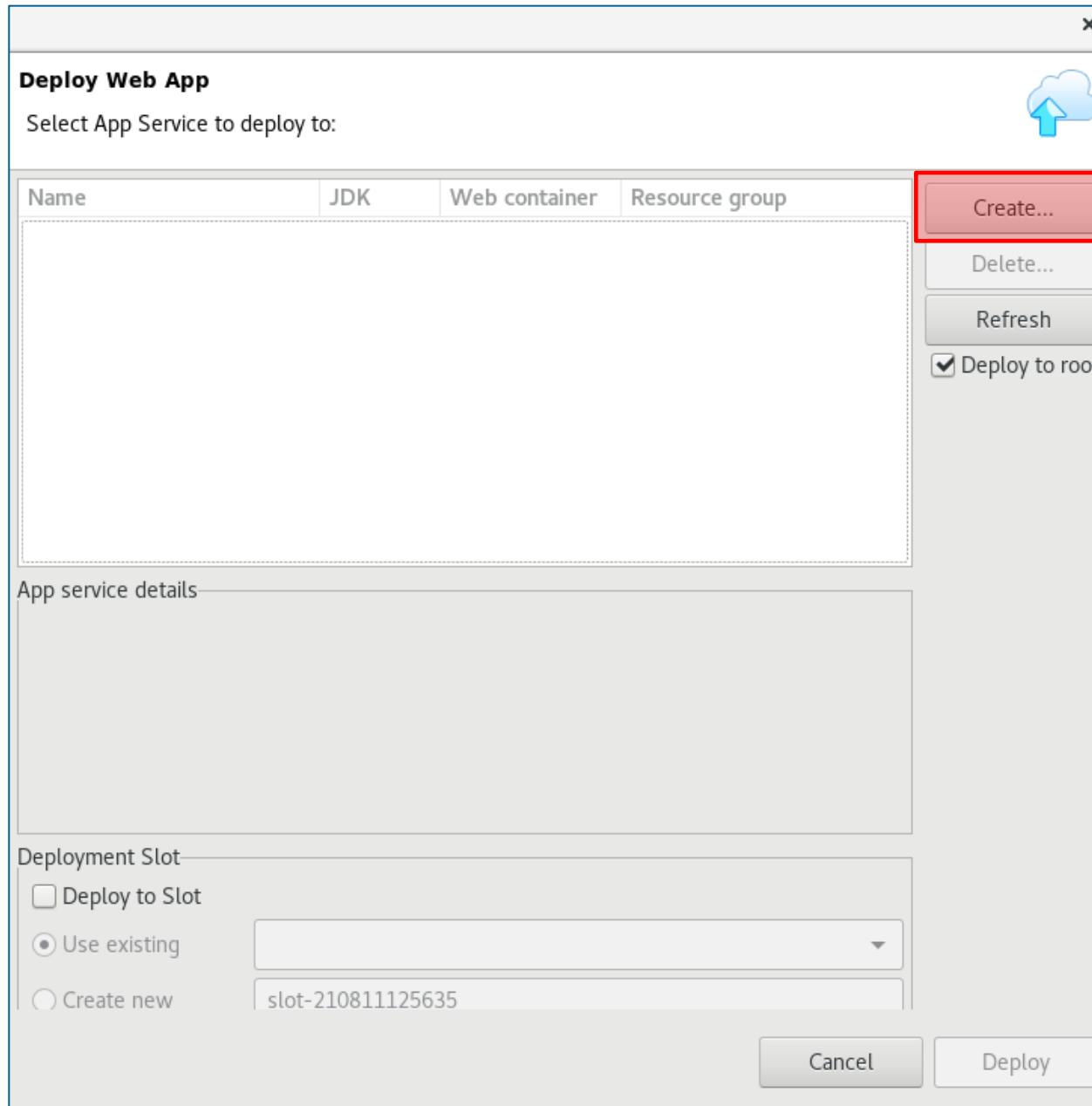


18) Vamos criar nosso Web App em nuvem: Infra, Build e Deploy



Instalar Plug-in Azure no Eclipse e Realizar testes

FIAP



Create App Service

Create Azure App Service

Enter name .azurewebsites.net

Subscription ▾

Runtime

Linux Windows

Linux Runtime ▾

Java version ▾

Web container ▾

App service plan

Use existing

▼

Location N/A

Pricing tier N/A

Create new

PlanoServico

Location

Brazil South

Pricing tier

Free_F1

[App service pricing details](#)

Resource group

Use existing

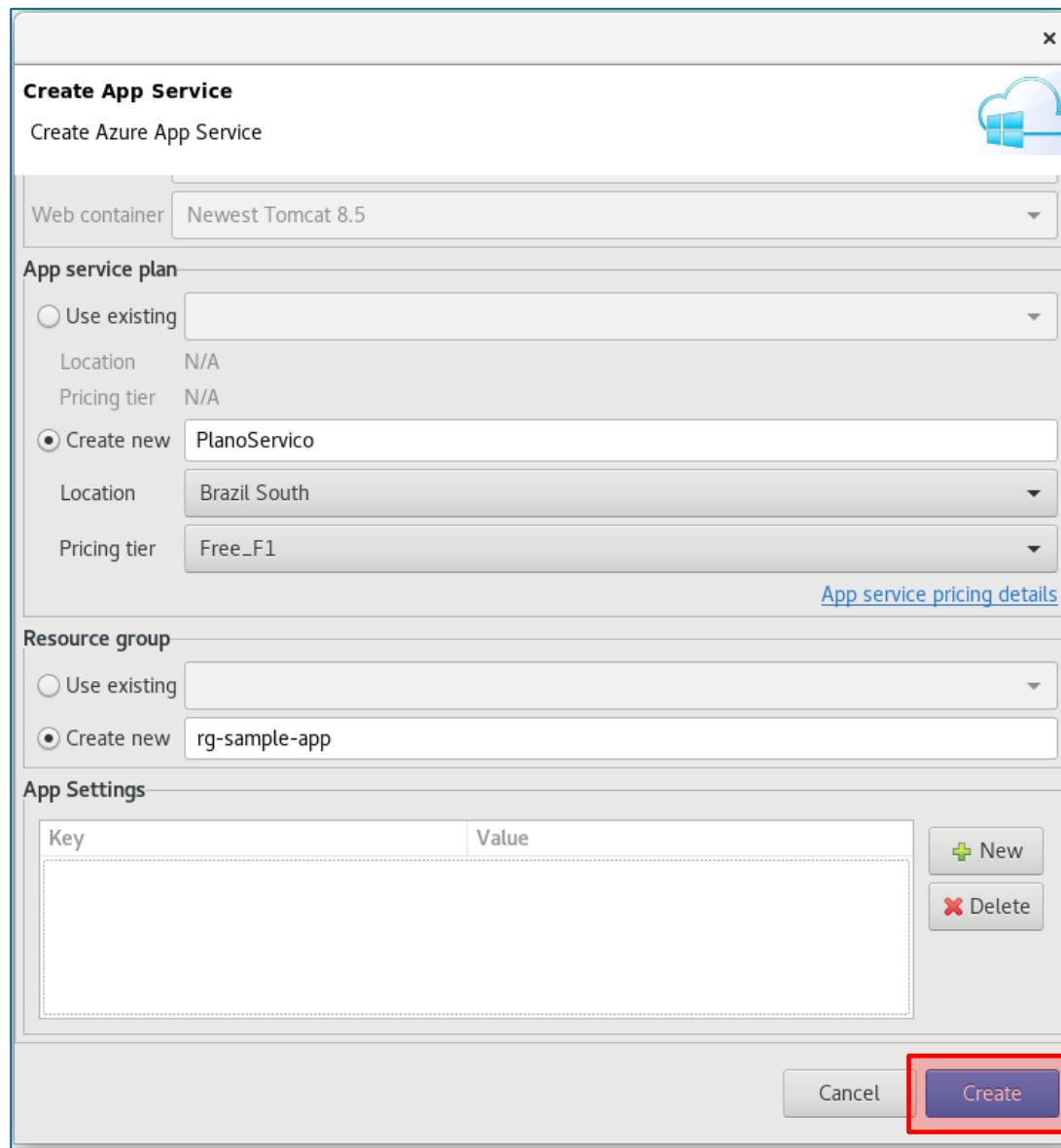
▼

Create new

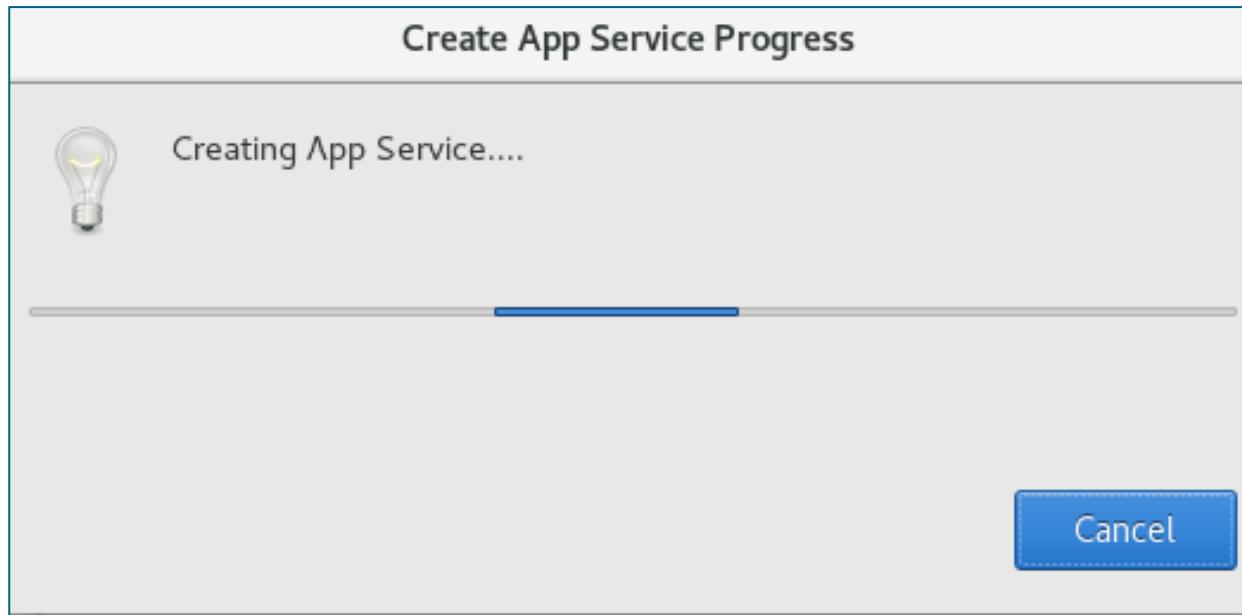
rg-sample-app

Instalar Plug-in Azure no Eclipse e Realizar testes

FIAP



Aguarde a criação da nossa infra



Instalar Plug-in Azure no Eclipse e Realizar testes

Antes de criar em **Deploy** o Serviço de Aplicativo (Web App), certifique-se de ter selecionado **Deploy to Root** (**Implementar na raiz**)

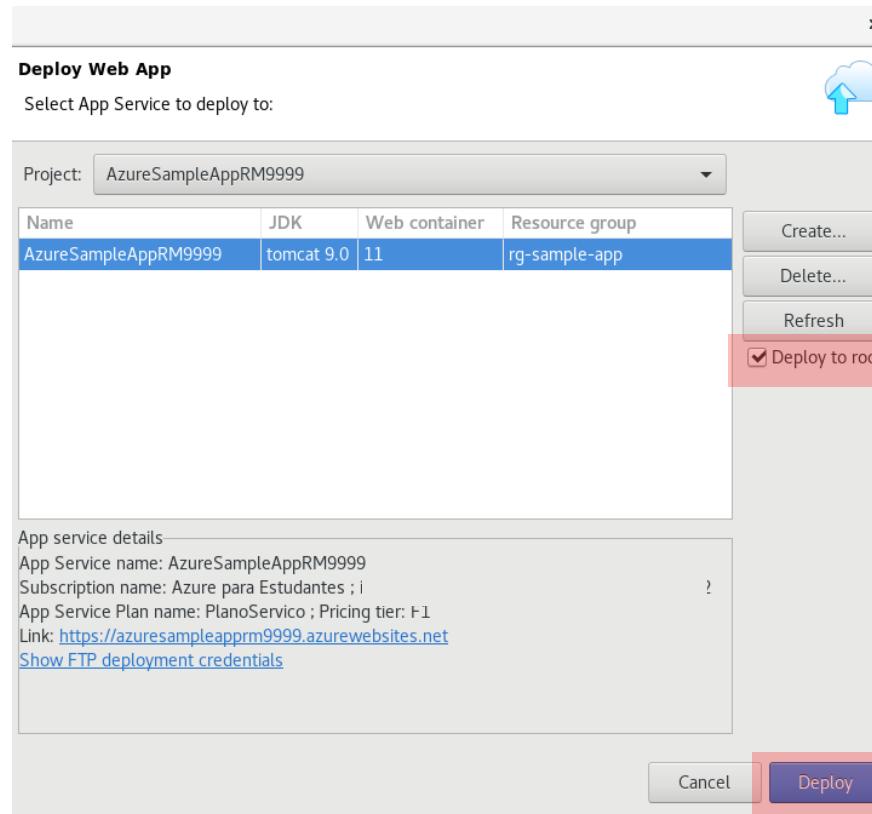
Tendo exemplo o nome do Aplicativo: **AzureSampleAppPF0841**

Se você não selecionar Deploy to Root, seu aplicativo será implementado em:

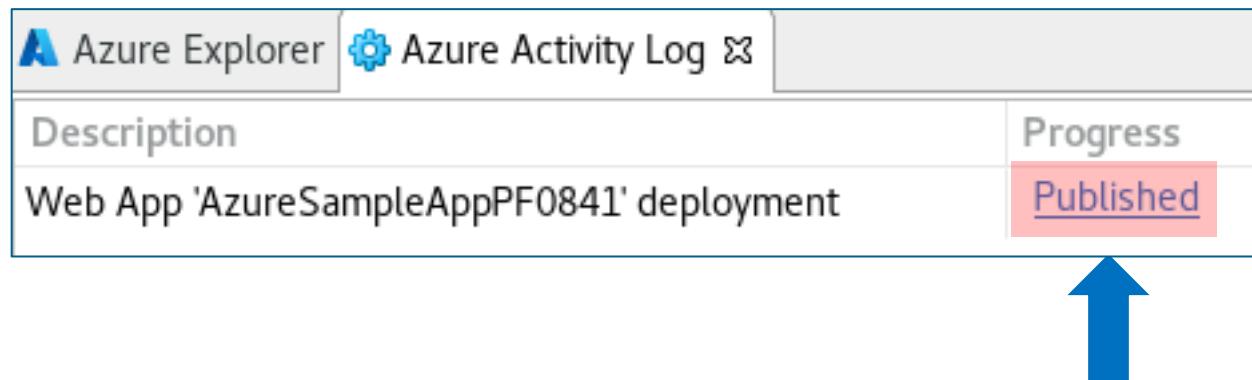
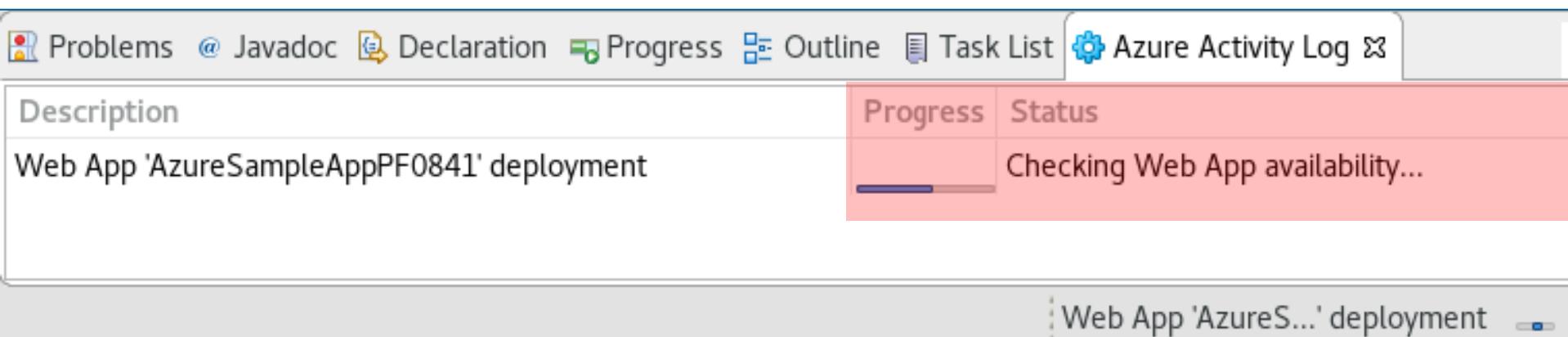
<https://azuresampleapppf0841.azurewebsites.net/azuresampleapppf0841>, tendo o nome do aplicativo no caminho da URL

Selecionando o aplicativo com implantação ao nível da raiz, a URL do aplicativo seria

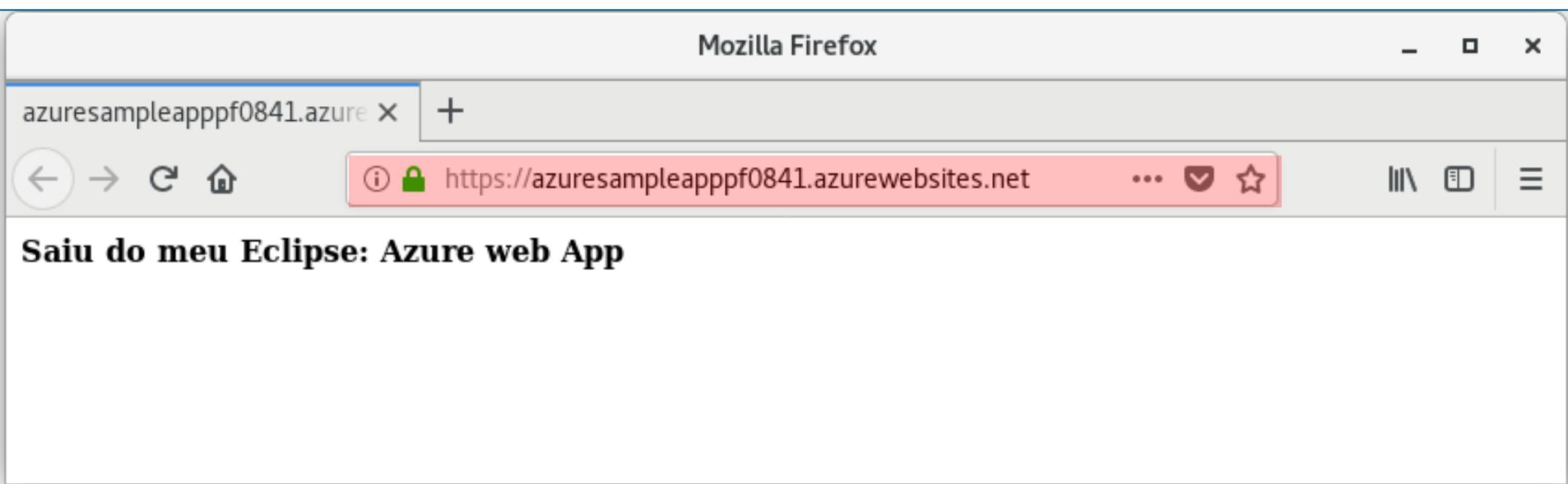
<https://azuresampleapppf0841.azurewebsites.net>



19) Aguarde o Deploy ser efetuado



20) Acesse a URL e verifique o resultado



21) Acesse o portal do Azure em sua conta e verifique os recursos criados

The screenshot shows the Azure portal interface for a resource group named 'rg-sample-app'. The left sidebar has a red box around it, highlighting the 'Visão geral' (Overview) section. The main content area displays the following details:

Nome	Tipo	Localização	Ações
AzureSampleAppPF0841	Serviço de Aplicativo	Sul do Brasil	...
PlanoServico	Plano do Serviço de Aplicativo	Sul do Brasil	...

Below the table, there are filter options: 'Filtrar por qualquer campo...', 'Tipo == tudo', 'Localização == tudo', and 'Adicionar filtro'. There are also buttons for 'Mostrar os tipos ocultos' (Show hidden types), 'Nenhum agrupamento' (No grouping), and 'Exibição de lista' (List view).

Acesse o Serviço de Aplicativo criado

Ferramentas de Desenvolvimento

Clonar Aplicativo

SSH ← **Essa opção aparece para Servidores Linux**

Ferramentas Avançadas

Extensões



Ferramentas Avançadas

As Ferramentas Avançadas oferecem uma variedade de recursos para gerenciar seu aplicativo.

[Saiba mais](#) →



Azure App Service

Environment SSH Bash Log stream



PF0841@fiap.com.br

Environment

Build 1.0.0.7 ([e59ed50ca2](#))
Site up time 00:00:08:56
Site folder /home
Temp folder /tmp/

REST API

 (works best when using a JSON viewer extension)

- [App Settings](#)
- [Deployments](#)
- [Source control info](#)
- [Files](#)
- [Current Docker logs \(Download as zip\)](#)

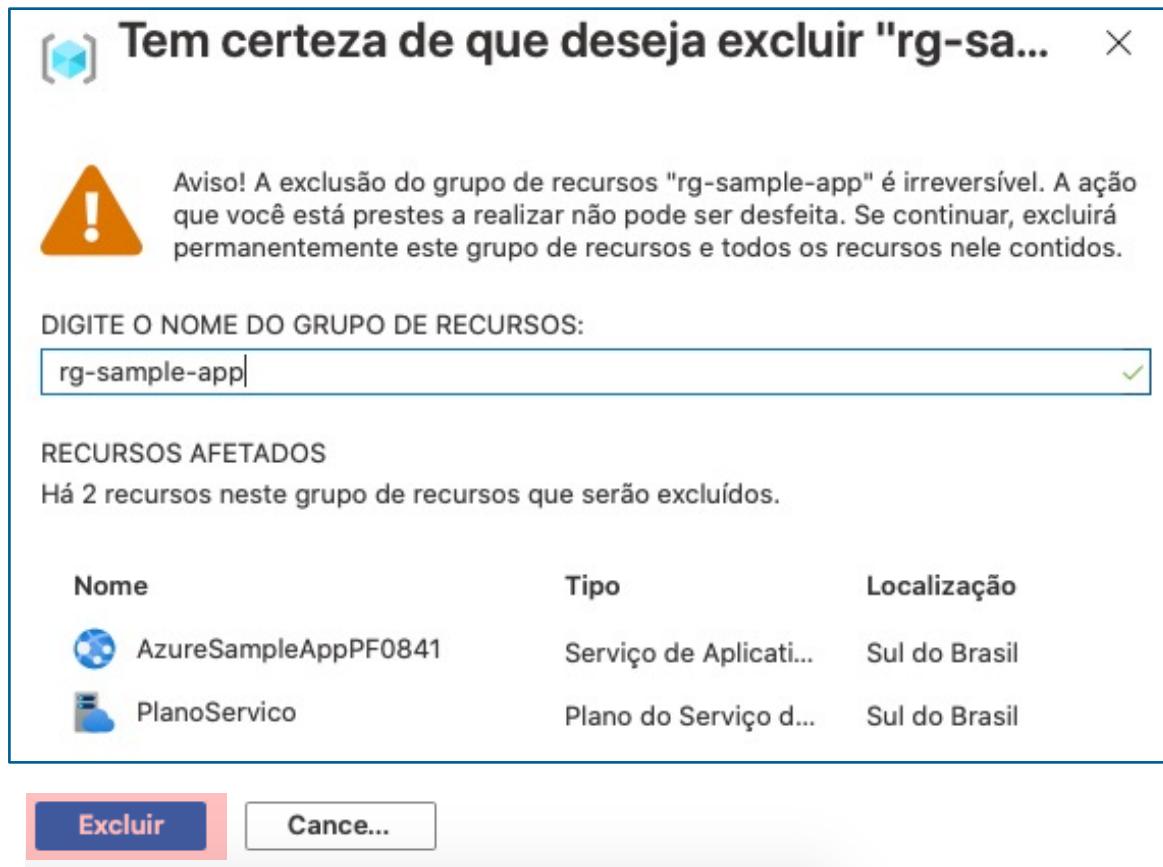
Browse Directory

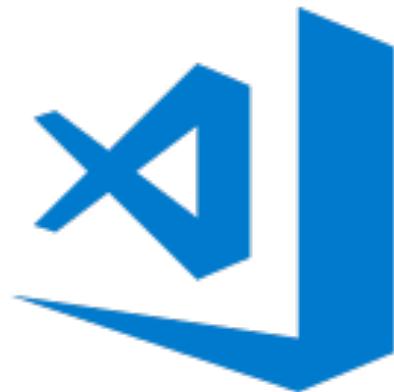
- [Deployment Logs](#)
- [Site wwwroot](#)

Acesse o Serviço de Aplicativo criado

Index of /wwwroot/		
Name	Size	Last Modified
app.war	972	00:39:04 +00:00
hostingstart.html	3,269	00:38:11 +00:00

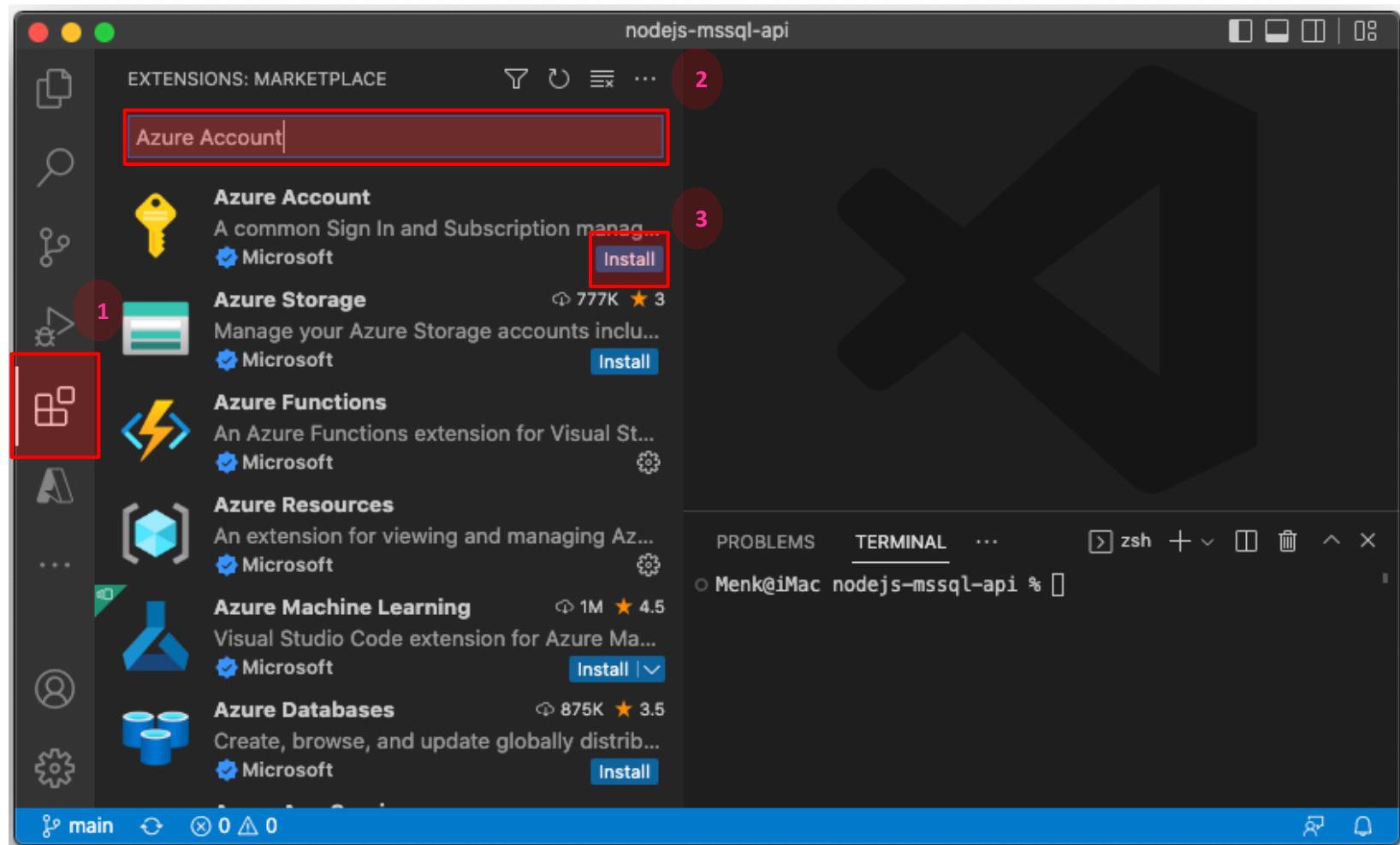
22) Após as verificações, exclua o Grupo de Recursos criado em nosso exemplo



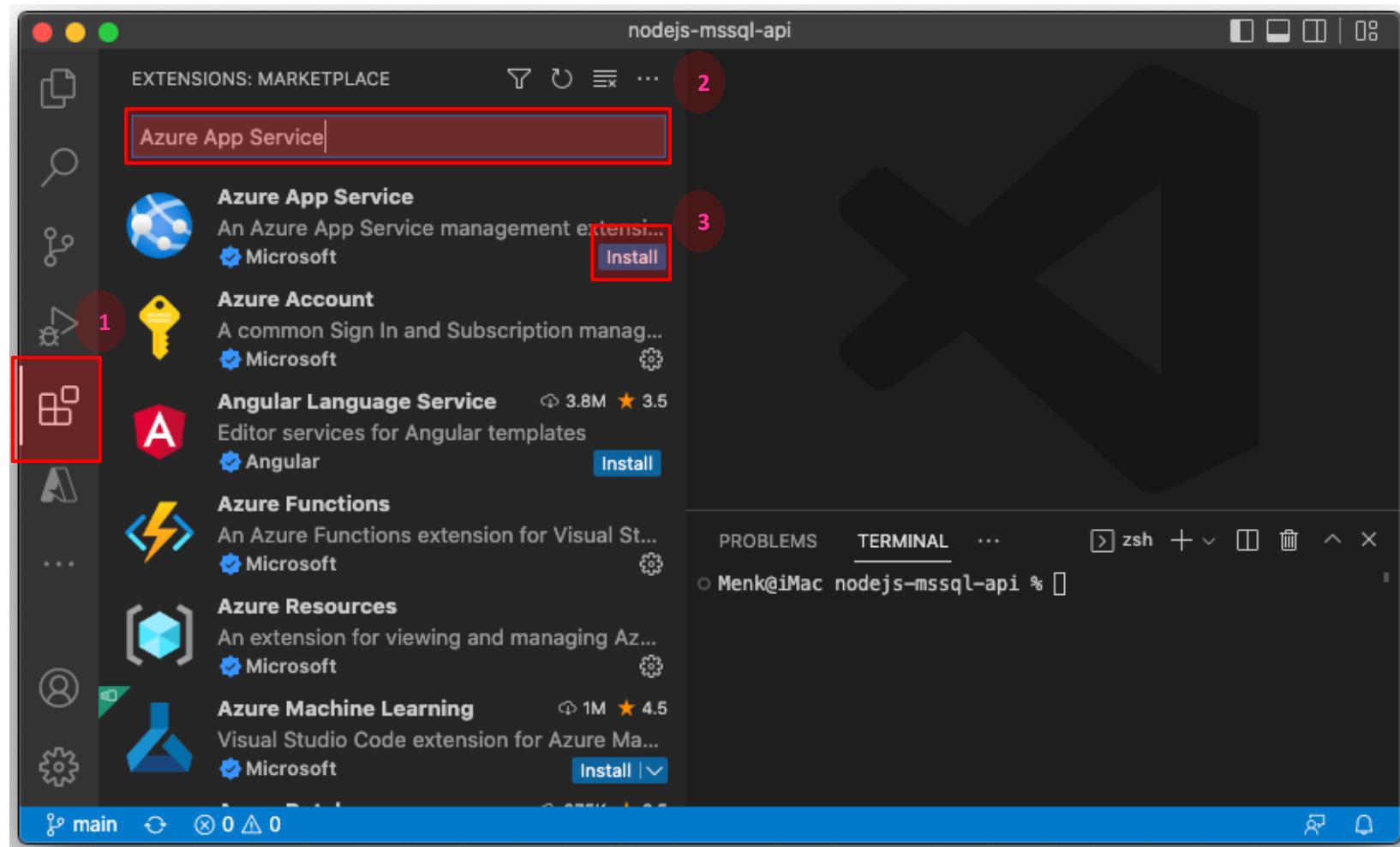


Visual Studio Code

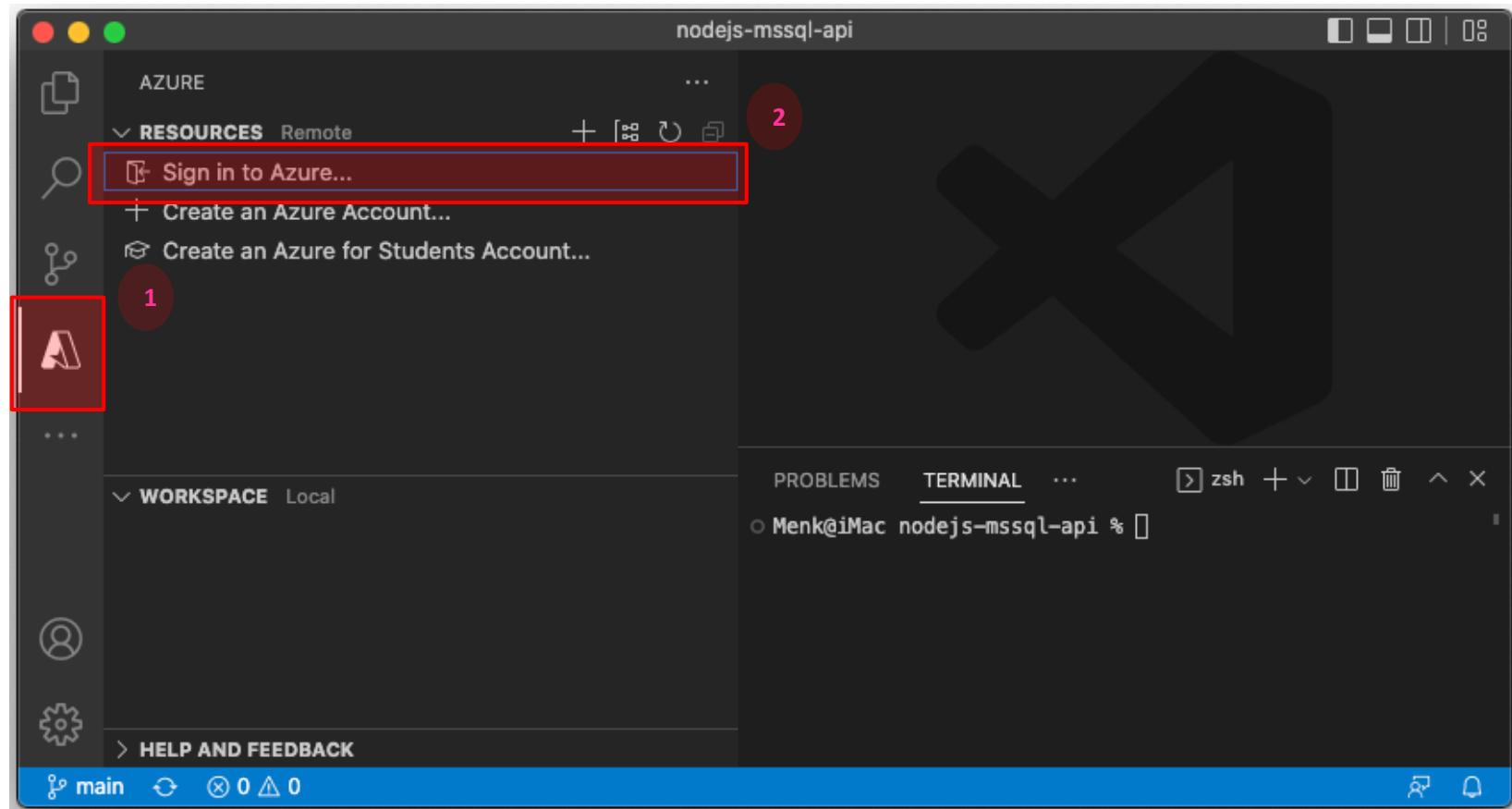
Em extensões, procure por **Azure Account** no Marketplace e instale



Em extensões, procure por Azure App Service no Marketplace e instale



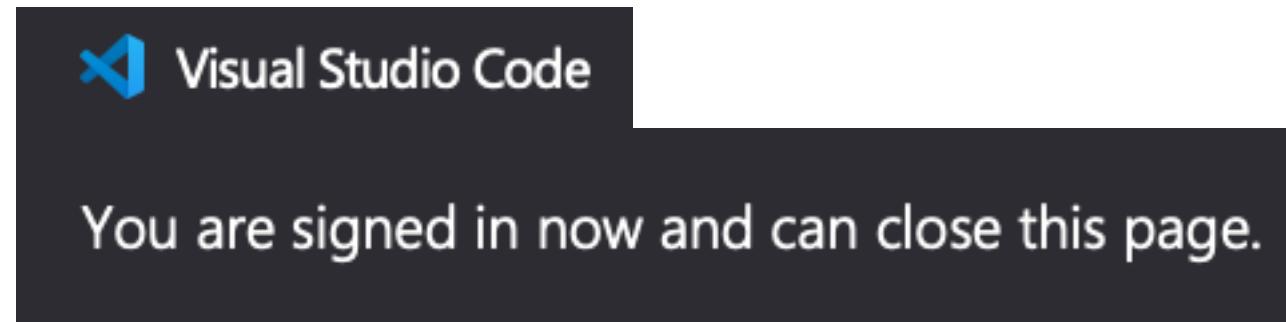
Navegue até a nova aba de extensão da Azure e faça seu Login em sua conta



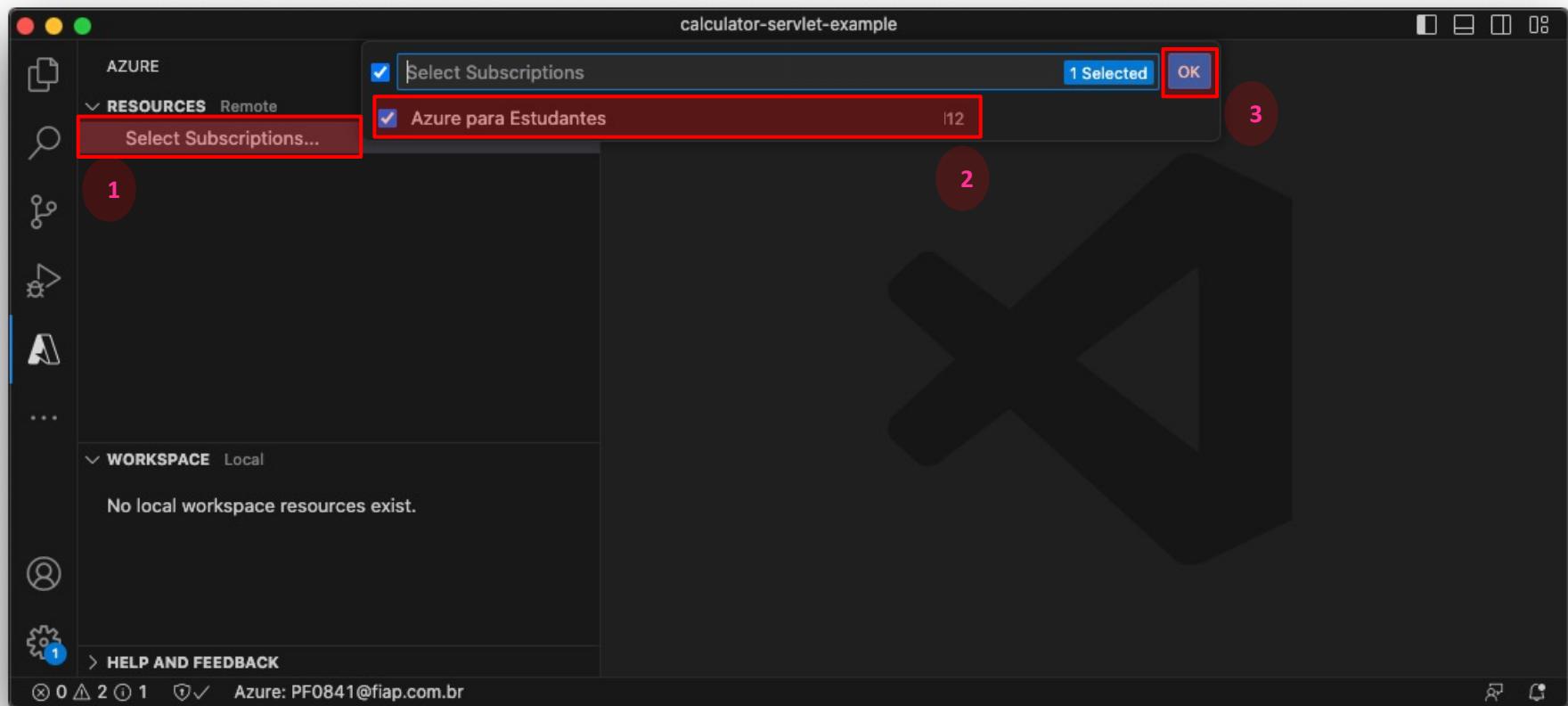
Uma nova aba de seu navegador padrão irá se abrir solicitando as credenciais



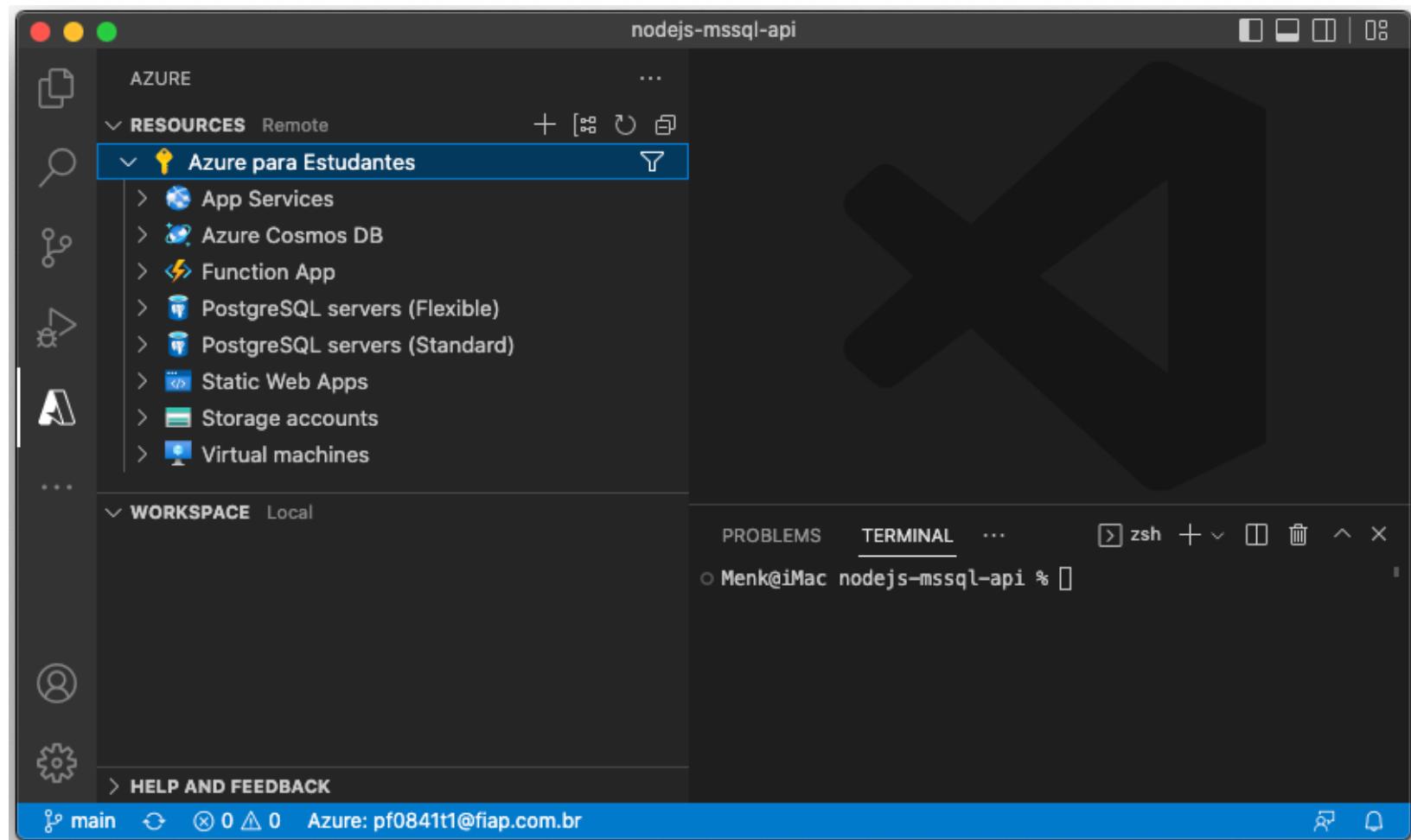
Assim que as credenciais forem aceitas você poderá fechar essa nova aba aberta



Escolha a assinatura



Agora que estamos com as extensões necessárias e logado na Azure, vamos criar nosso Web App em nuvem: Infra, Build e Deploy



Para o VSC vamos utilizar um exemplo de uma Aplicação Java Web com o Maven

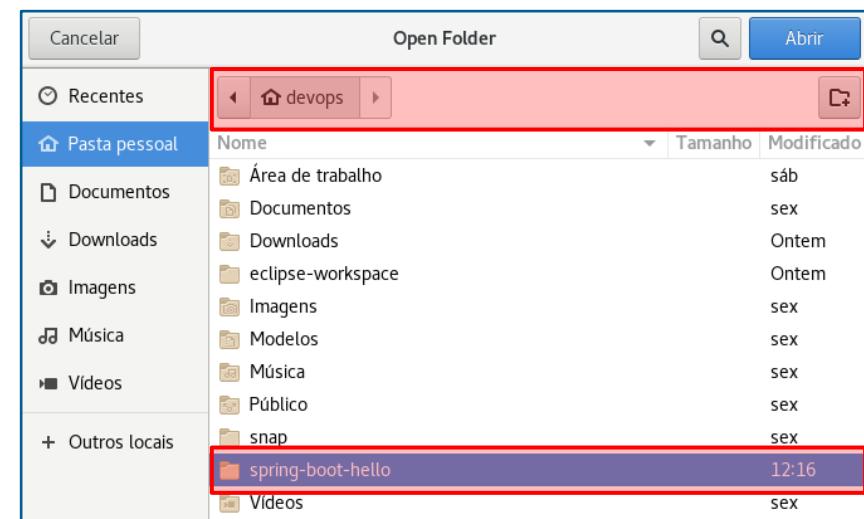
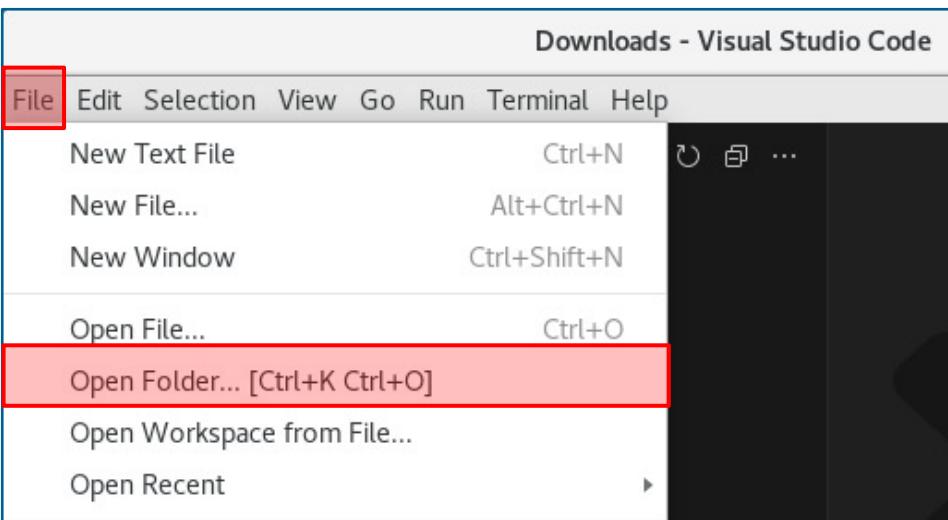
Primeiro realize o clone o repositório abaixo em seu diretório de trabalho

git clone <https://github.com/profjoaomenk/spring-boot-hello.git>

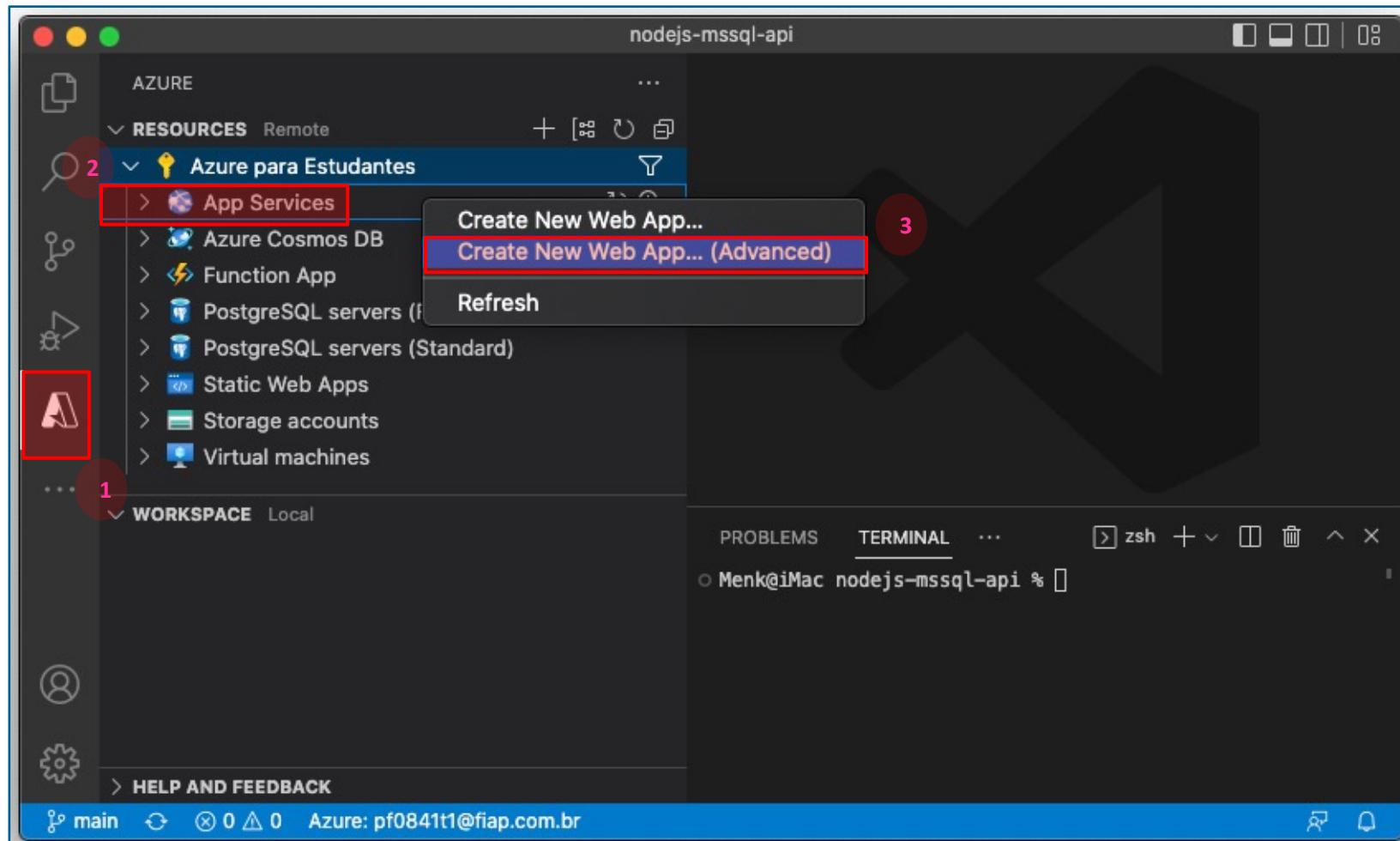
The screenshot shows a terminal window with a dark theme. The title bar reads "devops@orlinux8:~". The menu bar includes "Arquivo", "Editar", "Ver", "Pesquisar", "Terminal", and "Ajuda". The command line shows the user's path as "/home/devops" and the command "git clone https://github.com/profjoaomenk/spring-boot-hello.git". The output of the command is displayed below, showing the progress of cloning the repository, including object enumeration, counting, compressing, and receiving objects.

```
devops@orlinux8:~$ pwd
/home/devops
[devops@orlinux8 ~]$ git clone https://github.com/profjoaomenk/spring-boot-hello.git
Cloning into 'spring-boot-hello'...
remote: Enumerating objects: 29, done.
remote: Counting objects: 100% (29/29), done.
remote: Compressing objects: 100% (19/19), done.
remote: Total 29 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (29/29), 66.87 KiB | 1.81 MiB/s, done.
[devops@orlinux8 ~]$
```

Agora abra a pasta no VSC

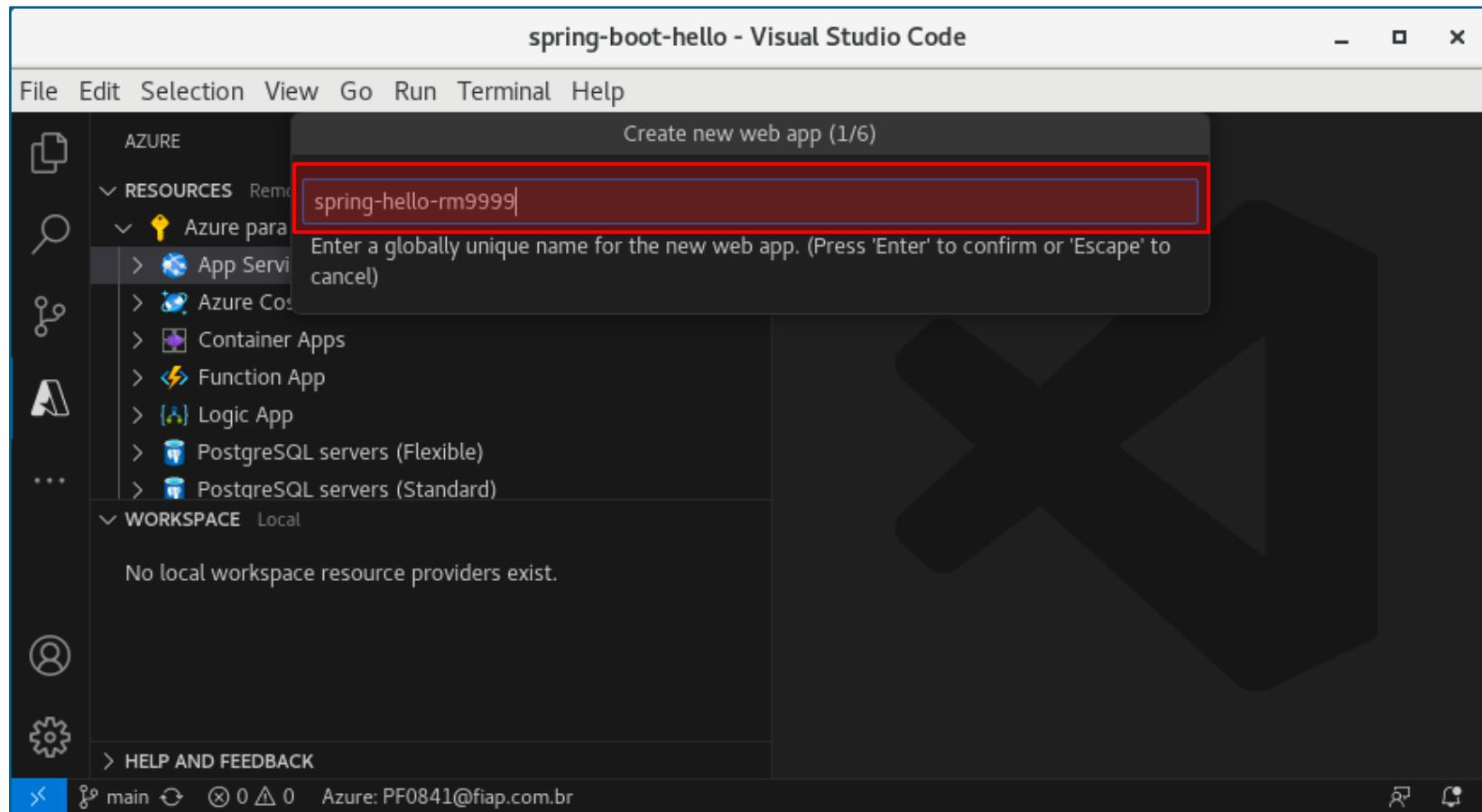


Clique com o botão direito do Mouse em App Services e escolha **Create New Web App... (Advanced)**

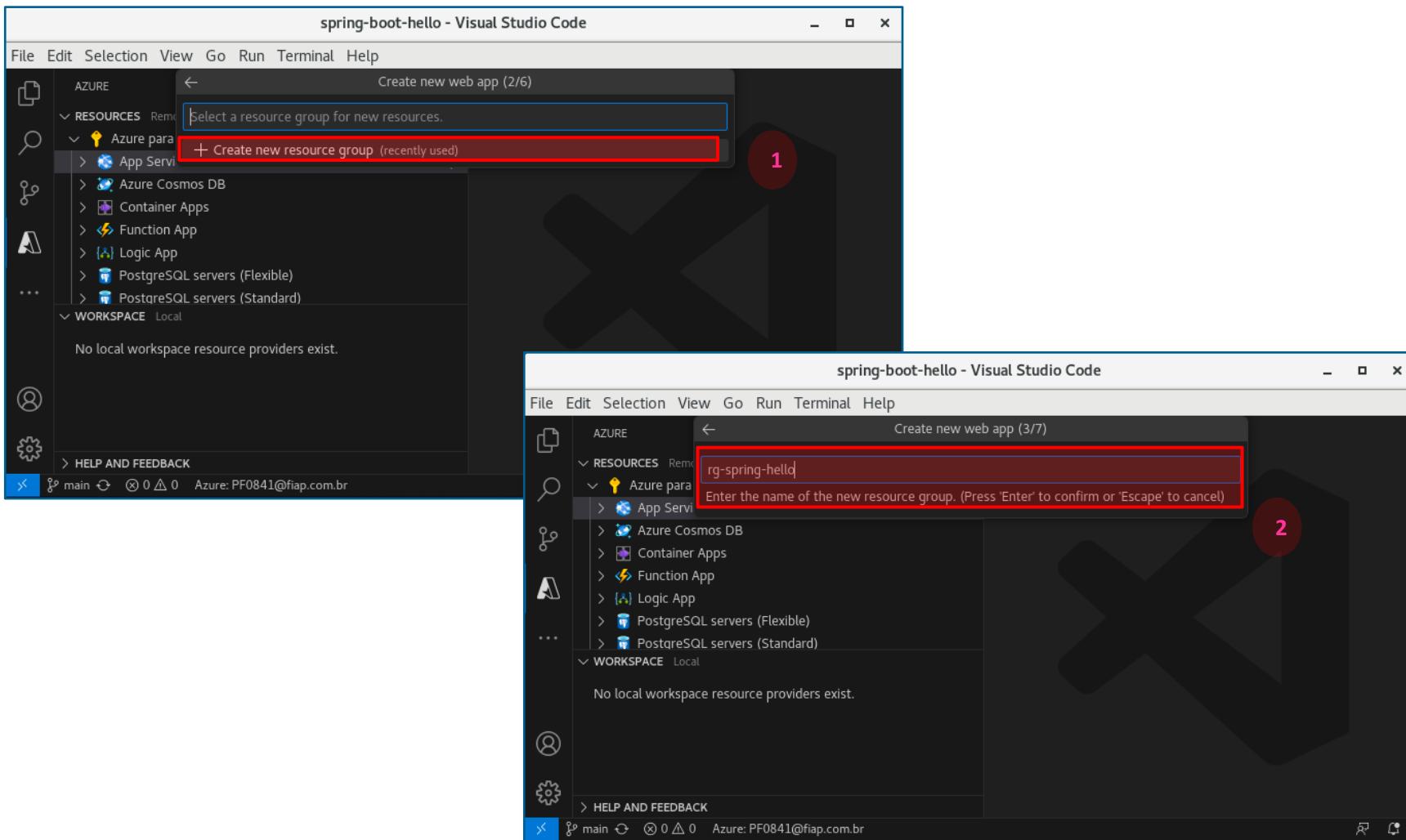


Entre com o **Nome da Aplicação** (nome único) e tecle Enter

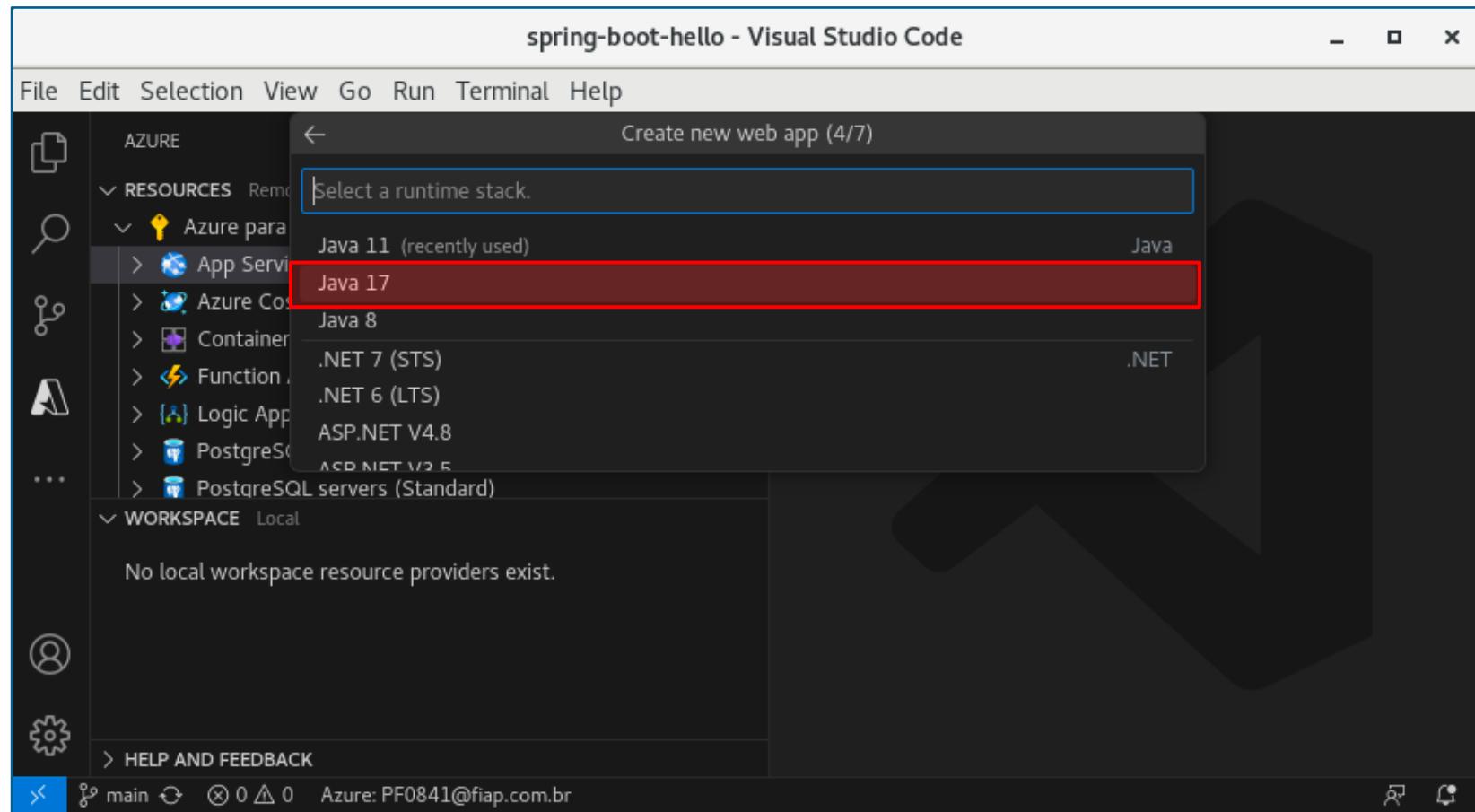
Altere para o número do seu RM



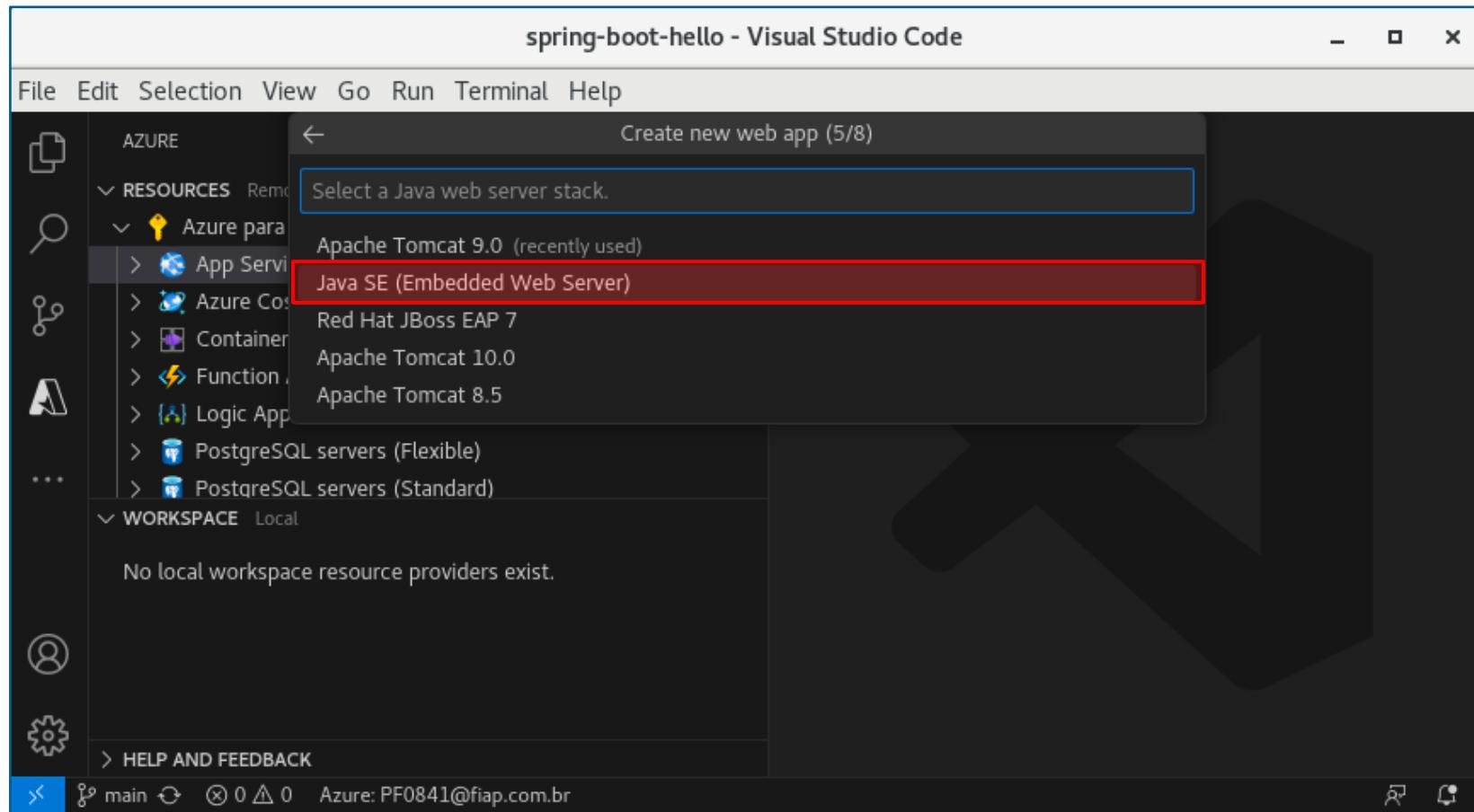
Escolha **Create new resource group**, tecle Enter, informe um **Nome** para o Grupo de Recursos e tecle Enter novamente



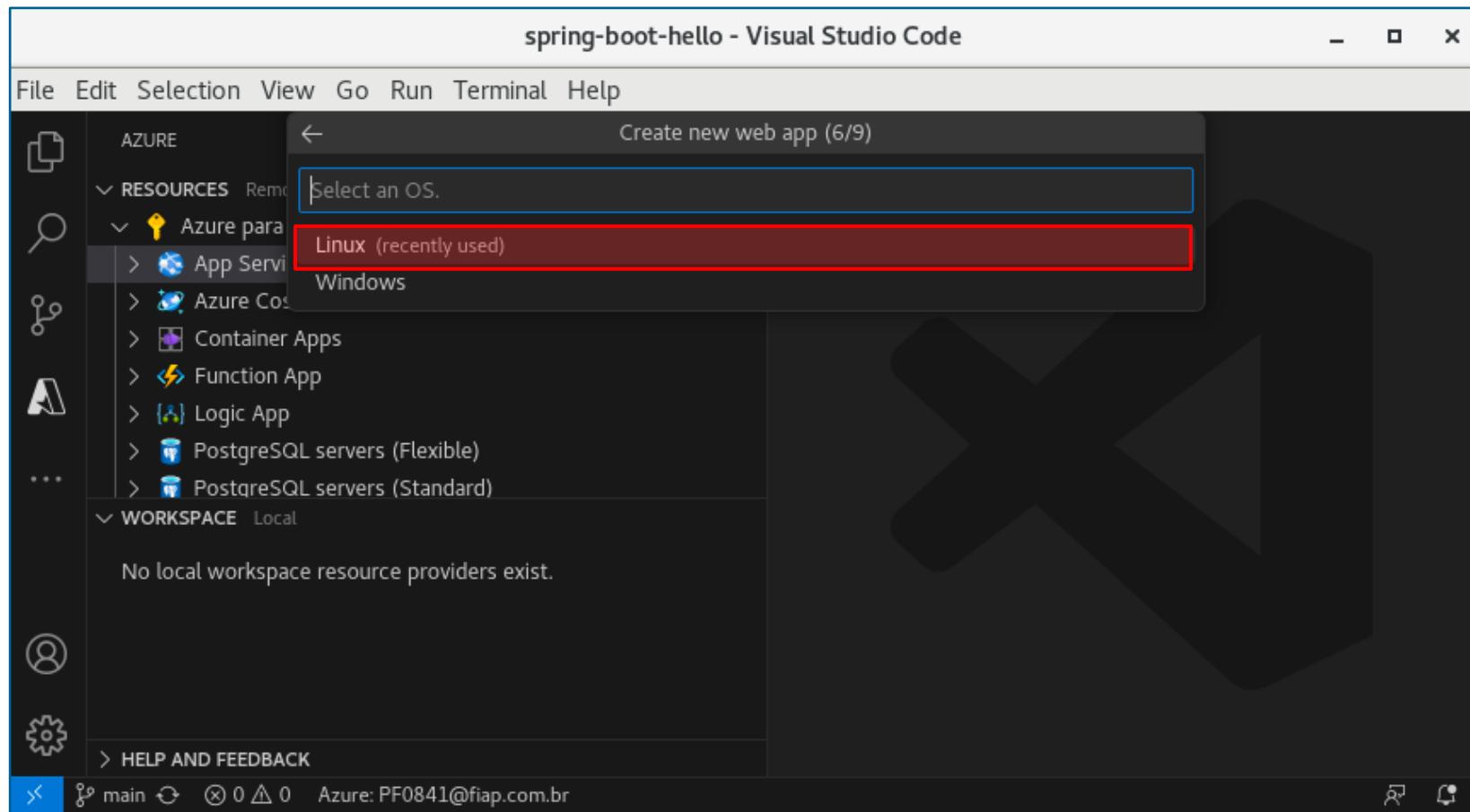
Selecione **Java 17** como a tecnologia a ser utilizada



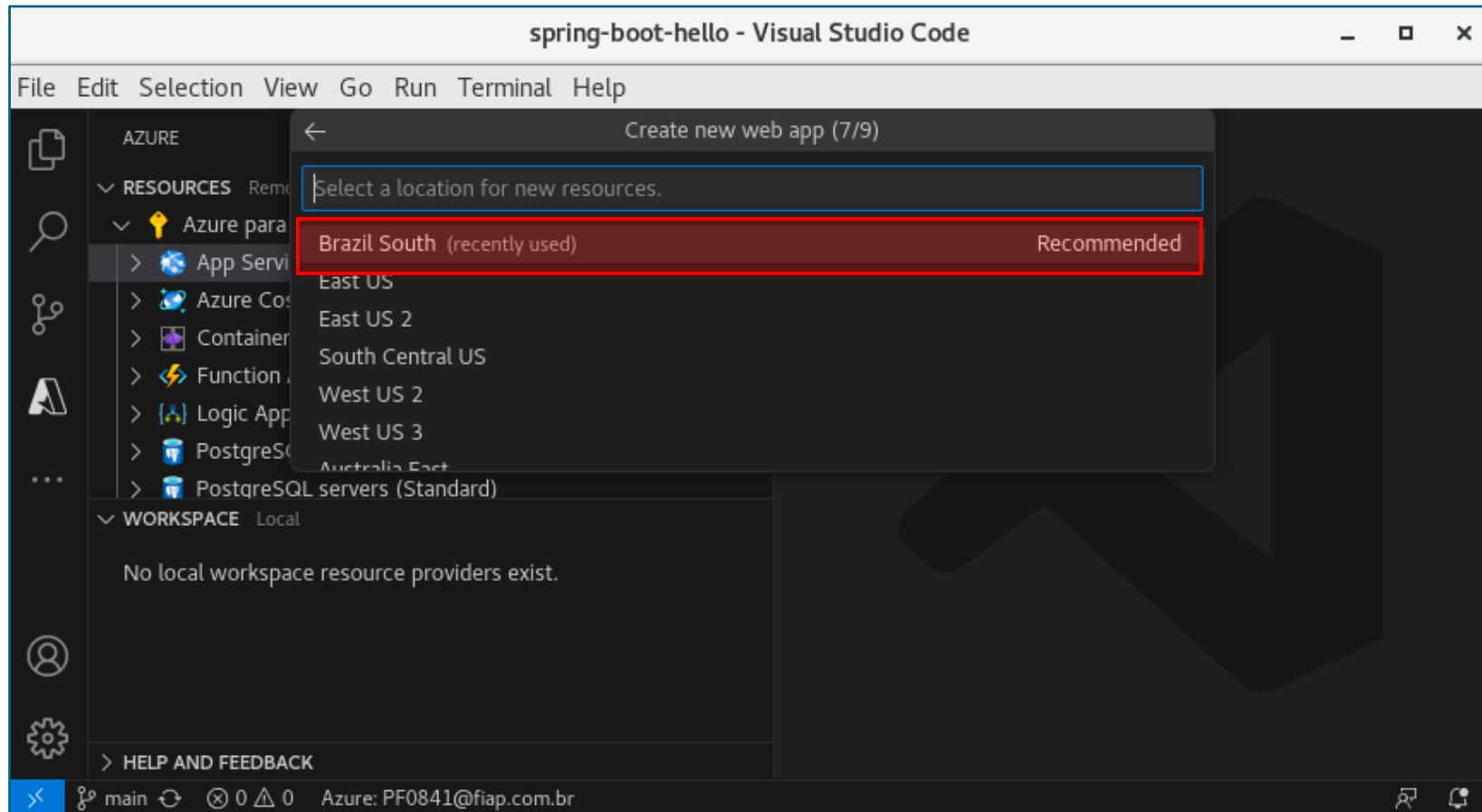
Selecione Java SE (Embedded Web Server)



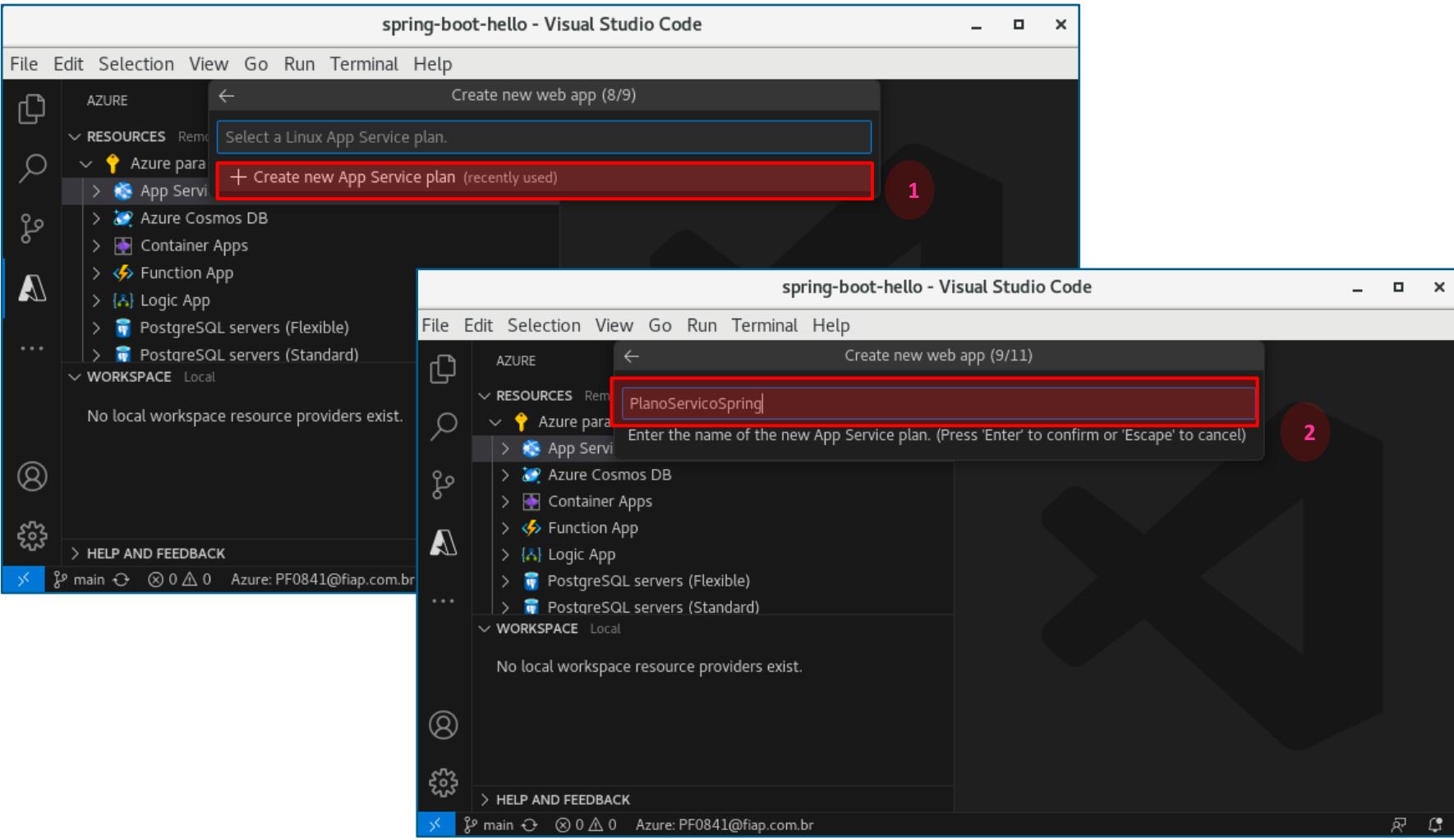
Selecione Linux como Sistema Operacional



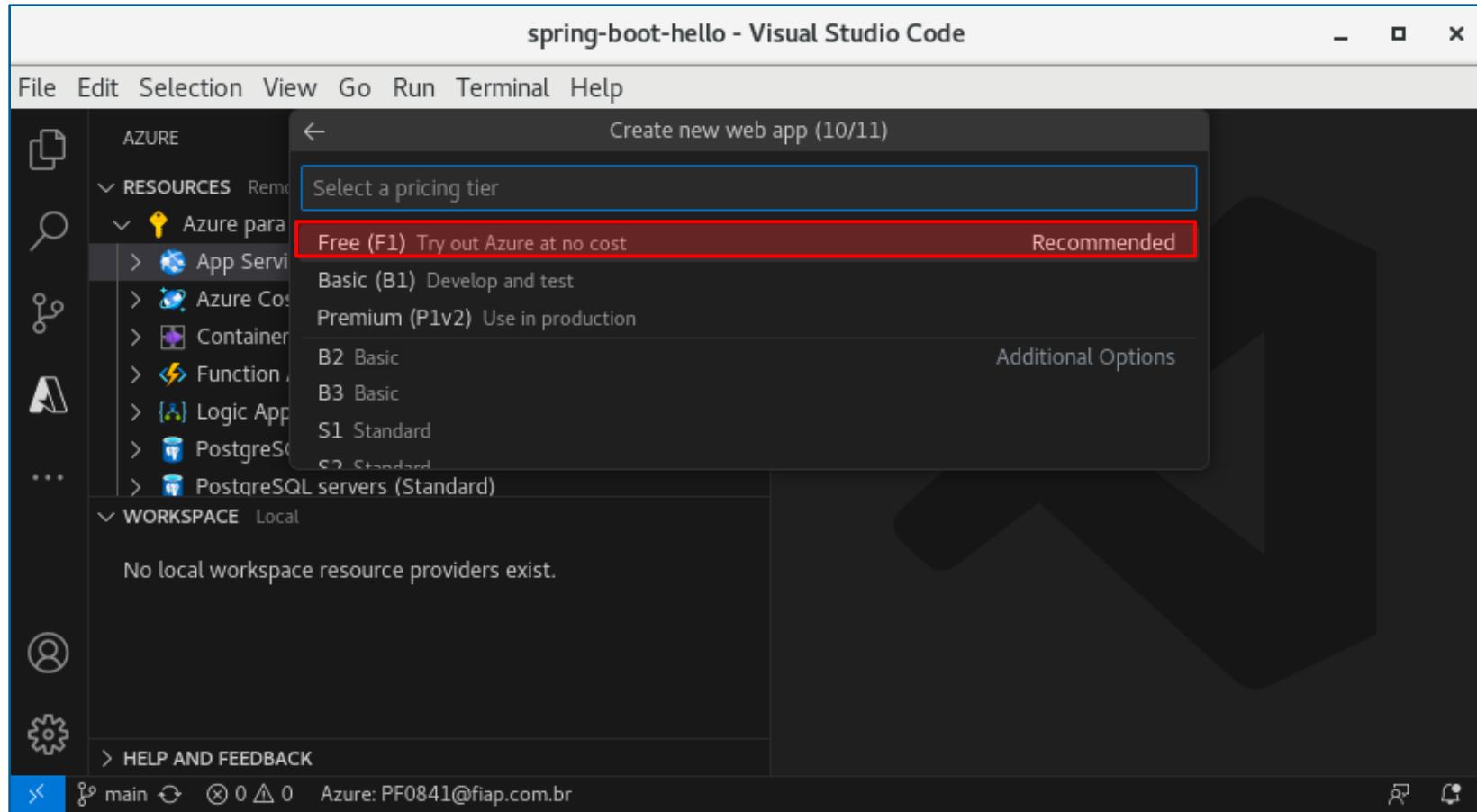
Escolha Brazil South como Datacenter



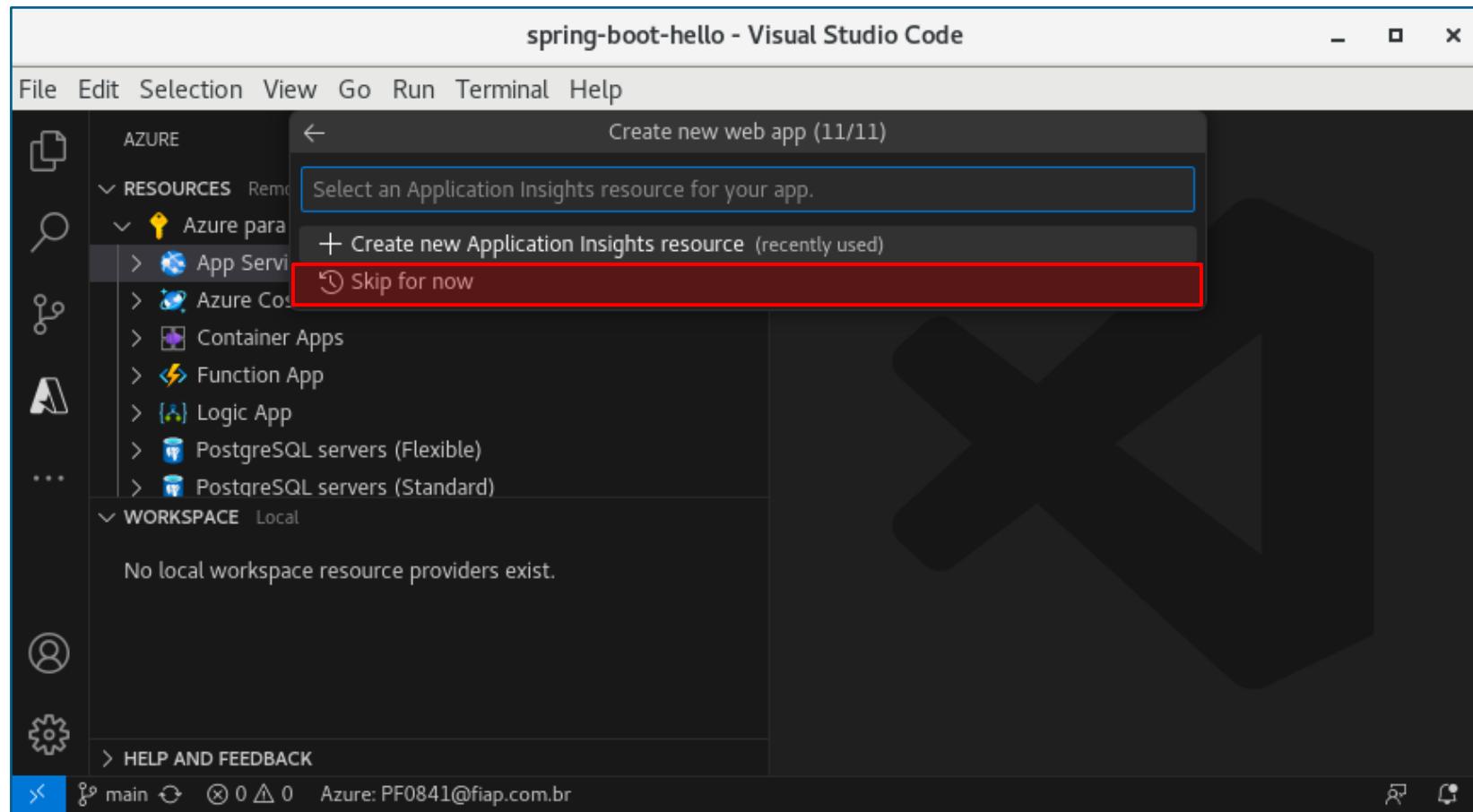
Selecione **Create new App Service plan**, defina o nome do **Plano de Serviço** e tecle Enter



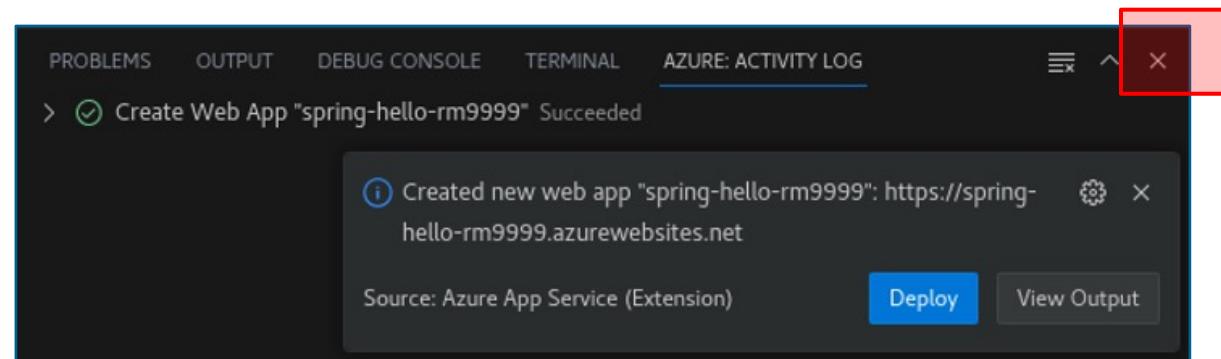
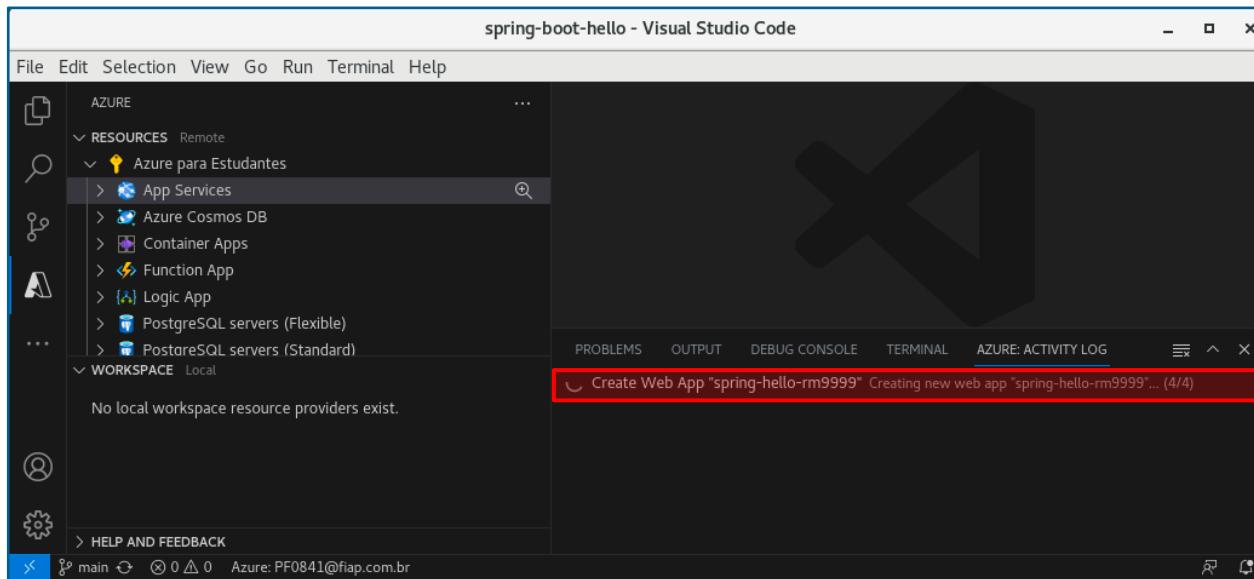
Selecione a camada gratuita Free (F1)



Nesse exemplo vamos pular a criação do Application Insights, selecione **Skip for now**

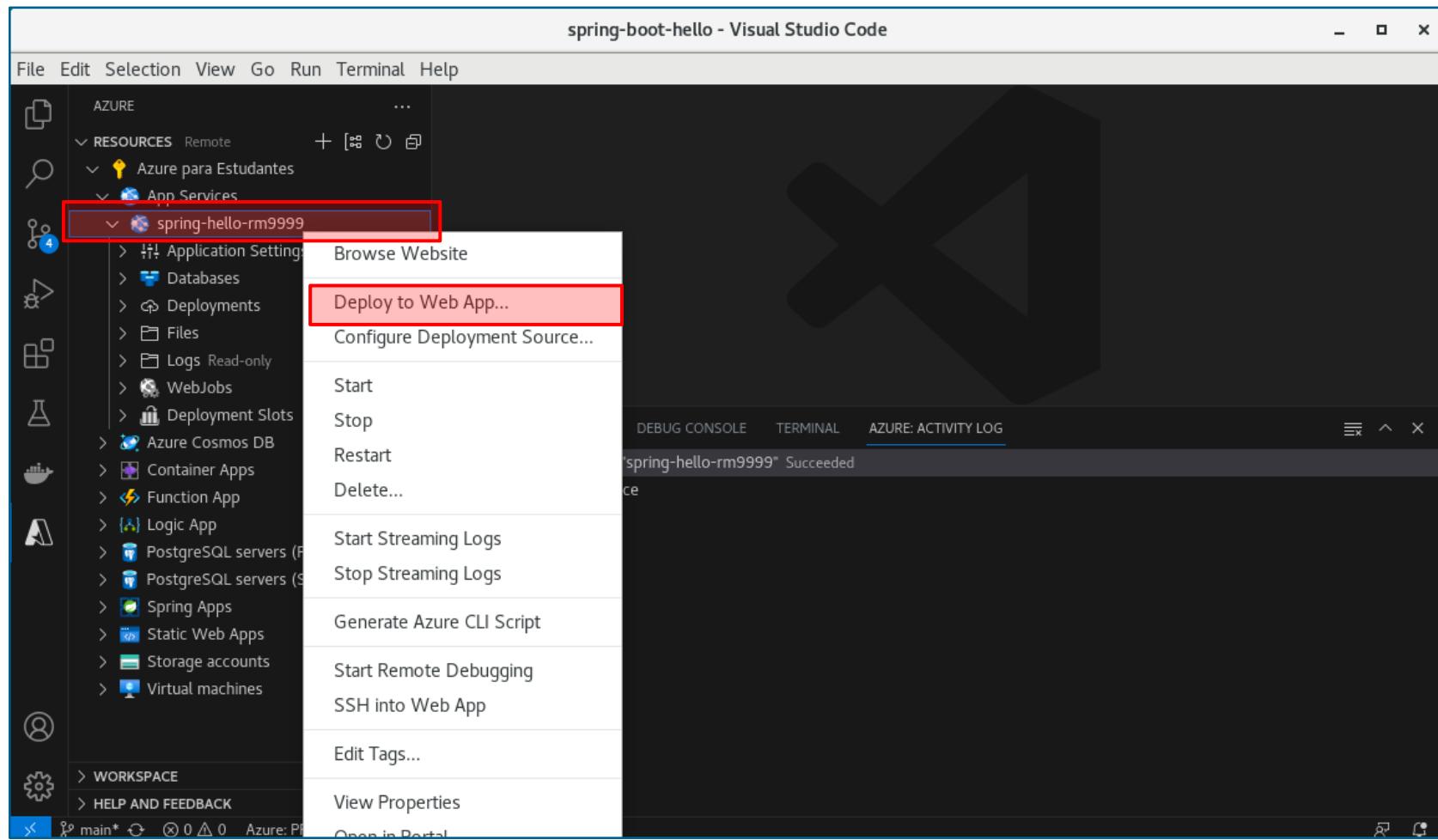


Aguarde a criação dos recursos na Azure. Nesse primeiro exemplo **ignore o botão Deploy**



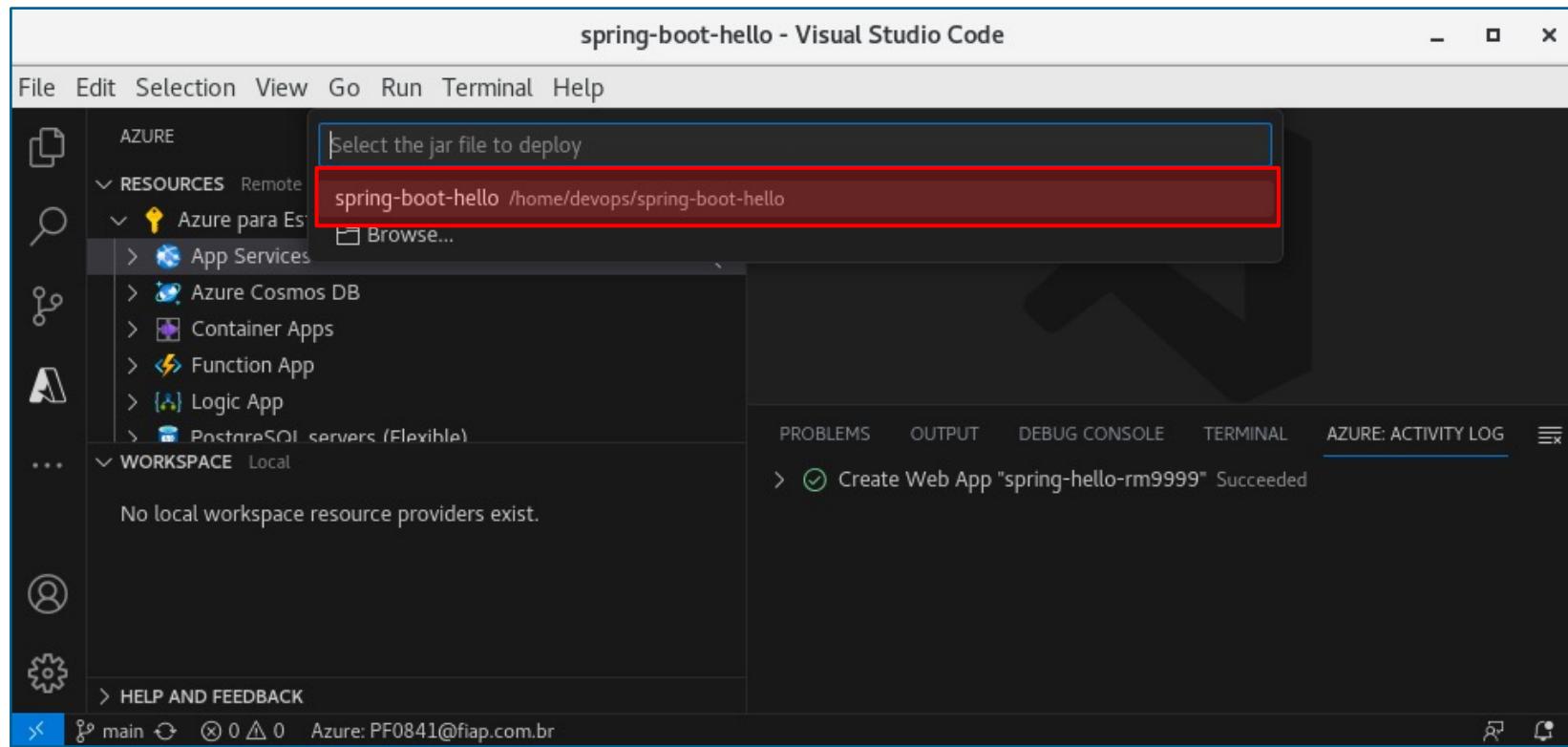
→ Ignore o botão Deploy

Expanda a sessão **App Services** e clique com o Botão direito do mouse no **Serviço de Aplicativo criado** e escolha **Deploy to Web App...**



Perceba que existem várias opções de gerenciamento do seu Web App: Iniciar, parar, deletar etc

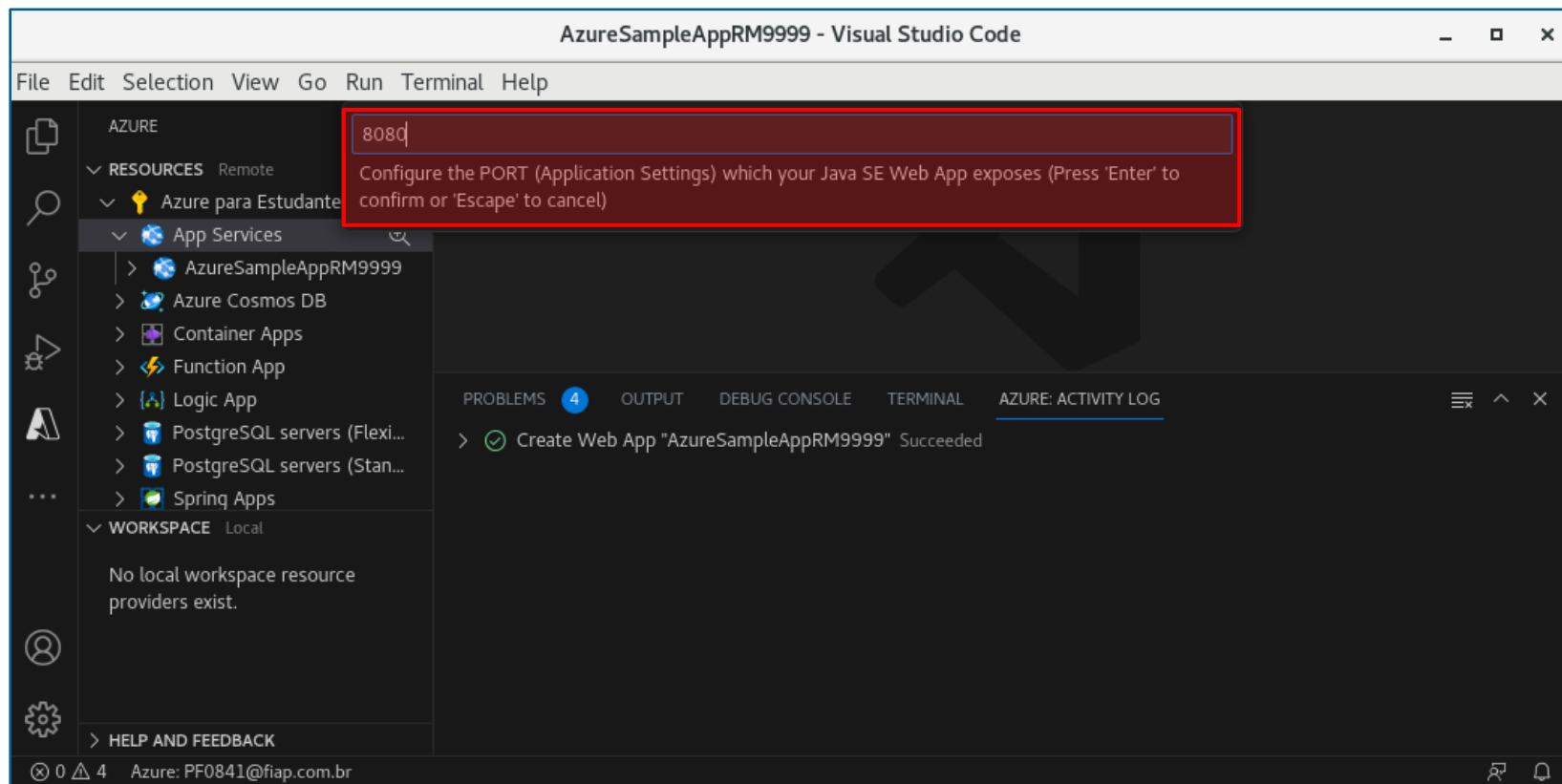
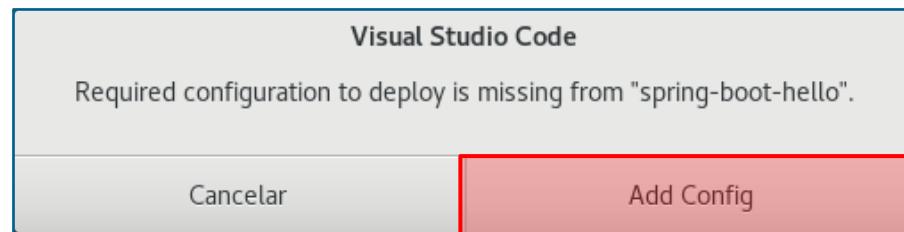
Clique na pasta do Projeto aberto



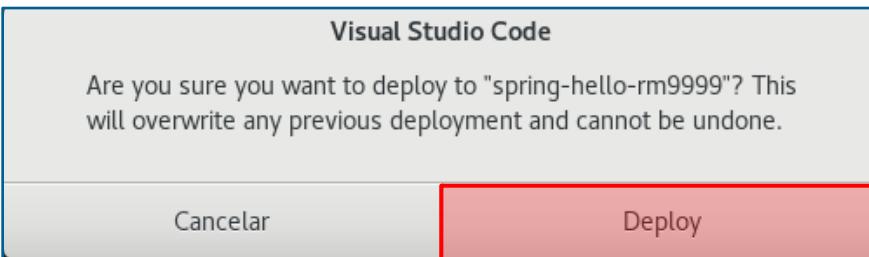
Instalar Plug-in Azure no VSC e Realizar testes

FIAP

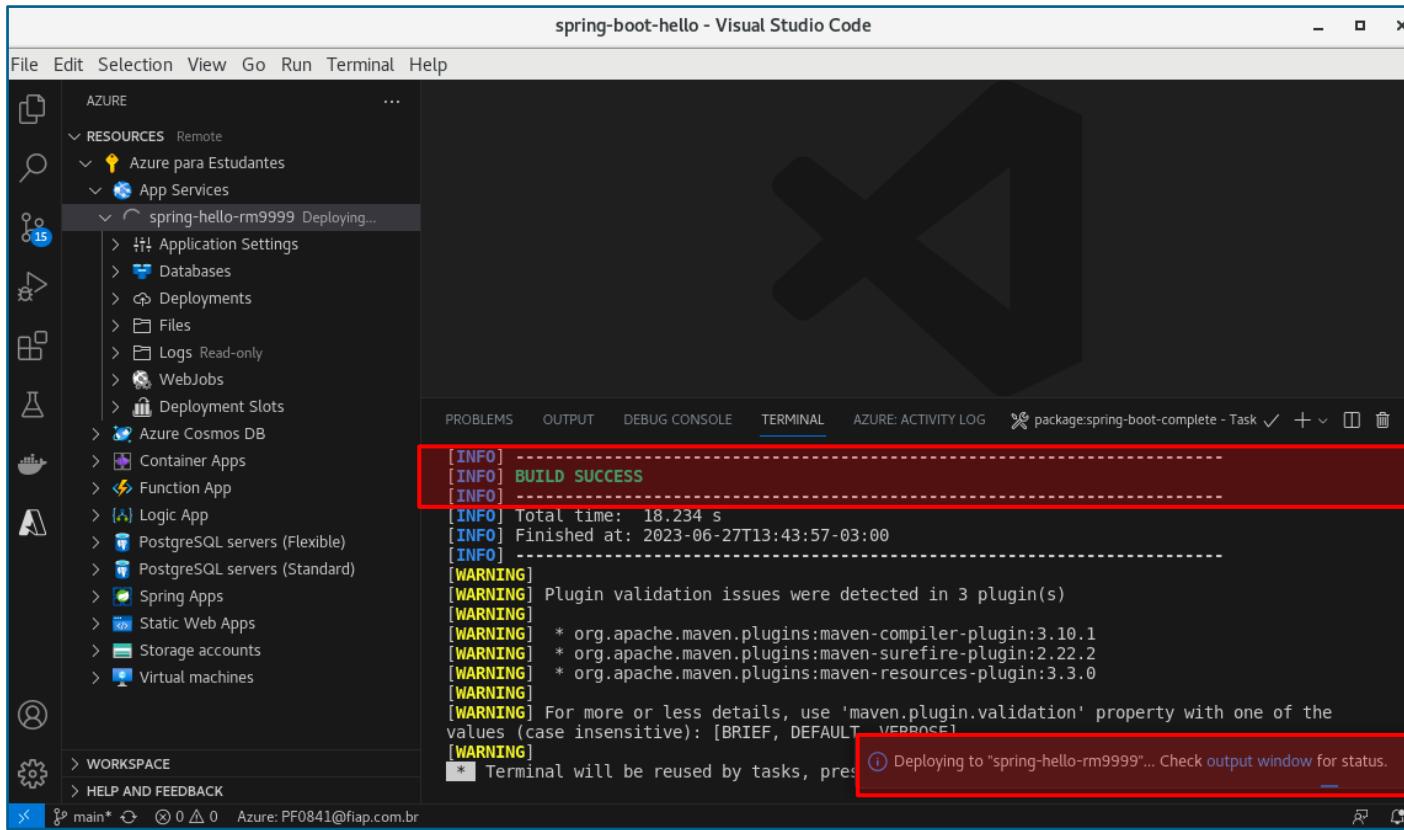
Selecione a **porta** do Projeto
e clique **ENTER**



Aguarde o Deploy



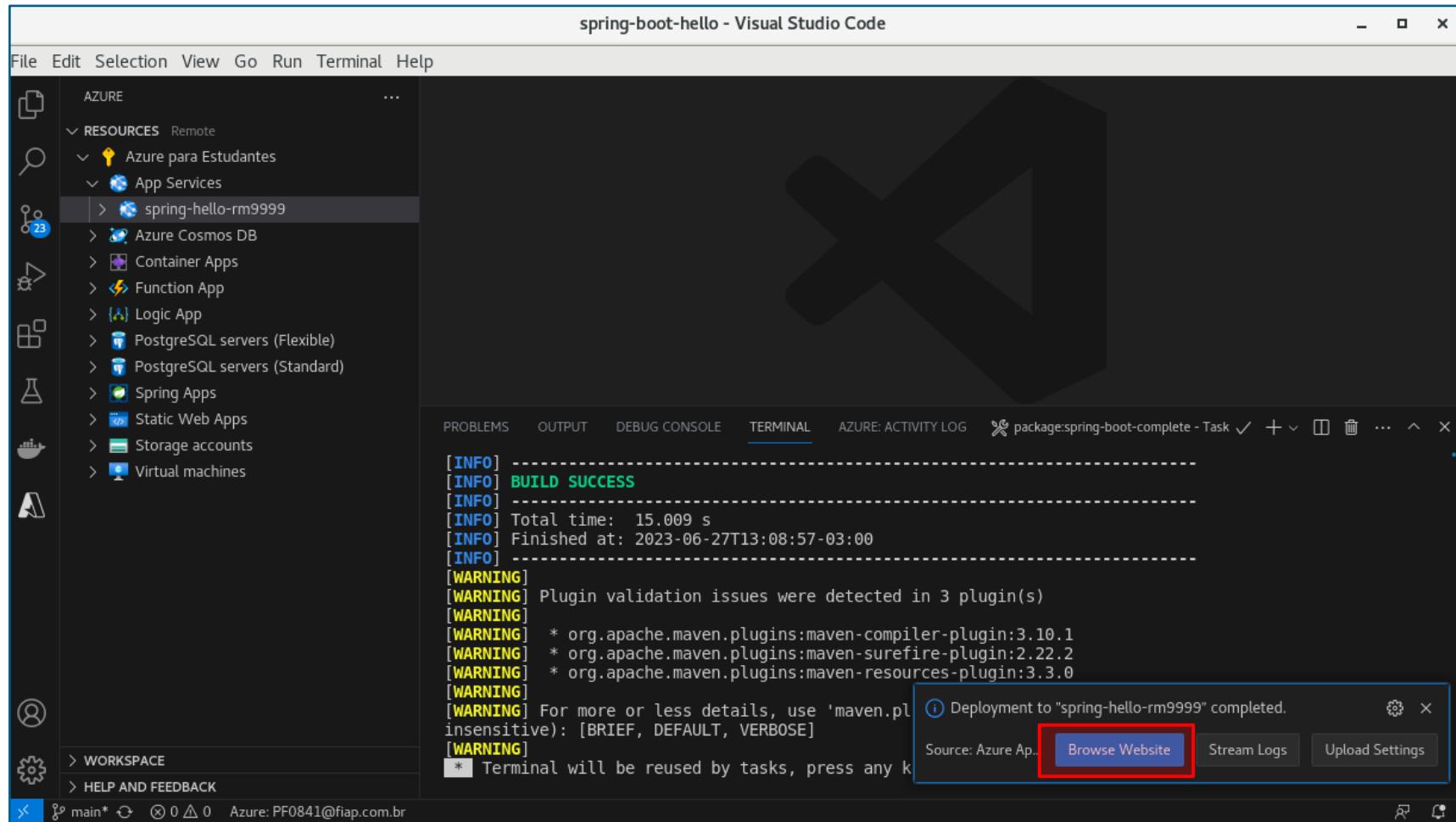
Caso essa janela apareça, escolha
Deploy



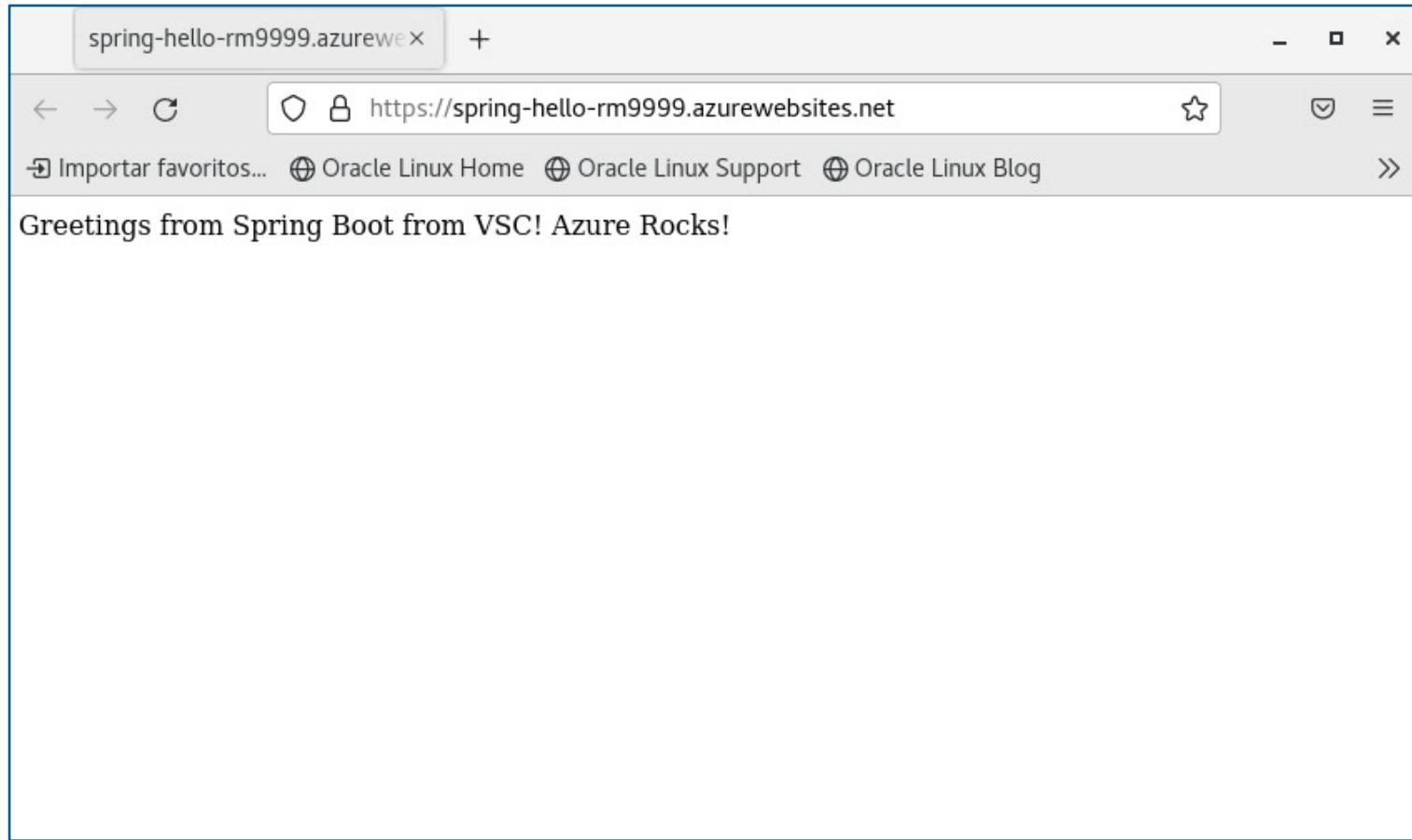
Instalar Plug-in Azure no VSC e Realizar testes

FIAP

Após o término do Deploy clique em **Browse Website**



Teste a Aplicação Web em Nuvem



<https://spring-hello-rm9999.azurewebsites.net/>

Instalar Plug-in Azure no VSC e Realizar testes



FIAP

Podemos realizar o Deploy somente indicando o pacote (previamente gerado) que desejamos levar para a Nuvem e qual será o Serviço de Aplicativo para hospedar

- 1) Realize alguma alteração no código fonte
- 2) Realize o Build manual
- 3) Copie o novo arquivo JAR gerado para outro diretório, Downloads por exemplo

The screenshot shows two instances of Visual Studio Code side-by-side. The left instance is titled 'HelloController.java - spring-boot-hello - Visual Studio Code' and contains Java code for a Spring Boot controller. The right instance is titled 'HelloControllerIT.java - spring-boot-hello - Visual Studio Code' and contains Java code for a Spring Boot test. Both code snippets have certain lines highlighted with red boxes, likely indicating changes made during the process.

File Edit Selection View Go Run Terminal Help

File Edit Selection View Go Run Terminal Help

```
• HelloController.java - spring-boot-hello - Visual Studio Code
HelloController.java
src > main > java > com > example > springboot > J HelloController.java > ...
1 package com.example.springboot;
2
3 import org.springframework.web.bind.annotation.GetMapping;
4 import org.springframework.web.bind.annotation.RestController;
5
6 @RestController
7 public class HelloController {
8
9     @GetMapping("/")
10    public String index() {
11        return "Greetings from Spring Boot!";
12    }
13
14
15 }
```

```
• HelloControllerIT.java - spring-boot-hello - Visual Studio Code
HelloControllerIT.java
src > test > java > com > example > springboot > J HelloControllerIT.java > ...
9
10    import static org.assertj.core.api.Assertions.assertThat;
11
12    @SpringBootTest(webEnvironment = SpringBootTest.WebEnvironment.RANDOM_PORT)
13    public class HelloControllerIT {
14
15        @Autowired
16        private TestRestTemplate template;
17
18        @Test
19        public void hello() throws Exception {
20            ResponseEntity<String> response = template.getForEntity("/", String.class);
21            assertThat(response.getBody()).isEqualTo("Greetings from Spring Boot!");
22        }
23
24 }
```

As mensagens devem ser as mesmas, caso contrário os Testes irão falhar

Instalar Plug-in Azure no VSC e Realizar testes



FIAP

```
File Edit Selection View Go Run Terminal Help
File Explorer View Test Terminal Help
src > test > java > com > example > springboot > HelloControllerTest.java > ...
17
18 @SpringBootTest
19 @AutoConfigureMockMvc
20 public class HelloControllerTest {
21
22     @Autowired
23     private MockMvc mvc;
24
25     @Test
26     public void getHello() throws Exception {
27         mvc.perform(MockMvcRequestBuilders.get("/"))
28             .andExpect(status().isOk())
29             .andExpect(content().string(equalTo("Greetings from Spring Boot!")));
30     }
31 }
32 
```

Abra um terminal no VSC

```
File Edit Selection View Go Run Terminal Help
File Explorer View Test Terminal Help
New Terminal Ctrl+Shift+` Split Terminal Ctrl+Shift+5 
```

Empacote de forma manual com o comando abaixo

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL AZURE
[INFO] BUILD SUCCESS
[INFO]
-----
[INFO] Total time: 13.522 s
[INFO] Finished at: 2023-06-27T15:00:15-03:00
[INFO]
-----
[WARNING] Plugin validation issues were detected in 3 plugin(s)
[WARNING] * org.apache.maven.plugins:maven-compiler-plugin:3.10.1
[WARNING] * org.apache.maven.plugins:maven-surefire-plugin:2.22.2
[WARNING] * org.apache.maven.plugins:maven-resources-plugin:3.3.0
[WARNING]
[WARNING] For more or less details, use 'maven.plugin.validation' property with one of the values (case insensitive): [BRIEF, DEFAULT, VERBOSE]
[WARNING]
[devops@orinux8 spring-boot-hello]$ 
```

mvn clean package



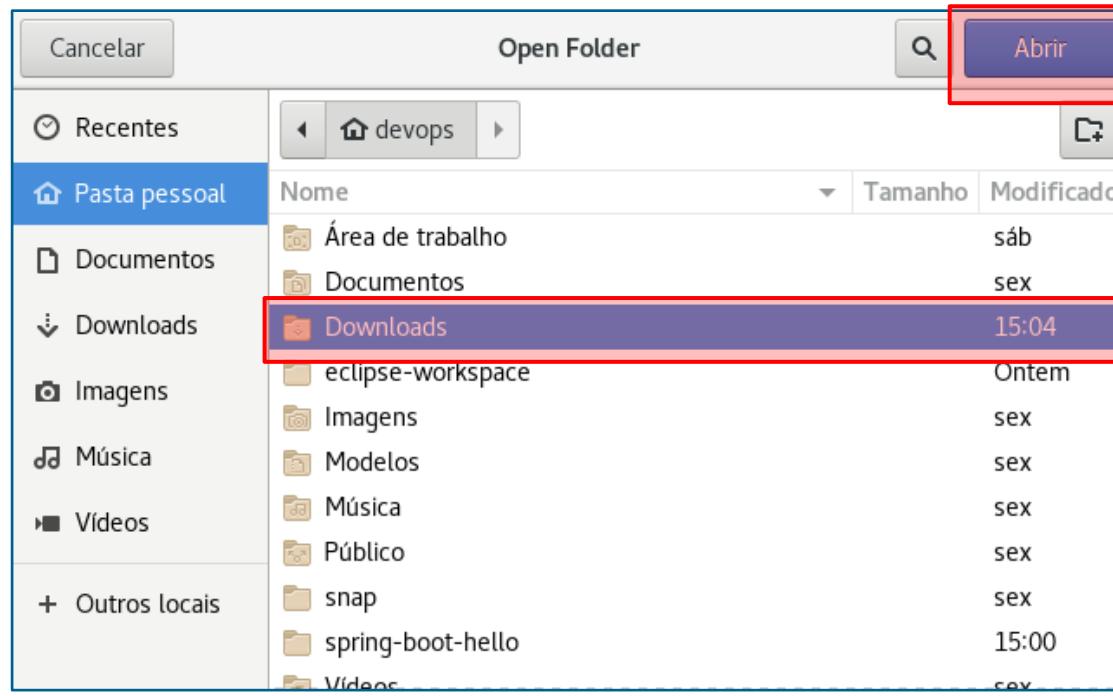
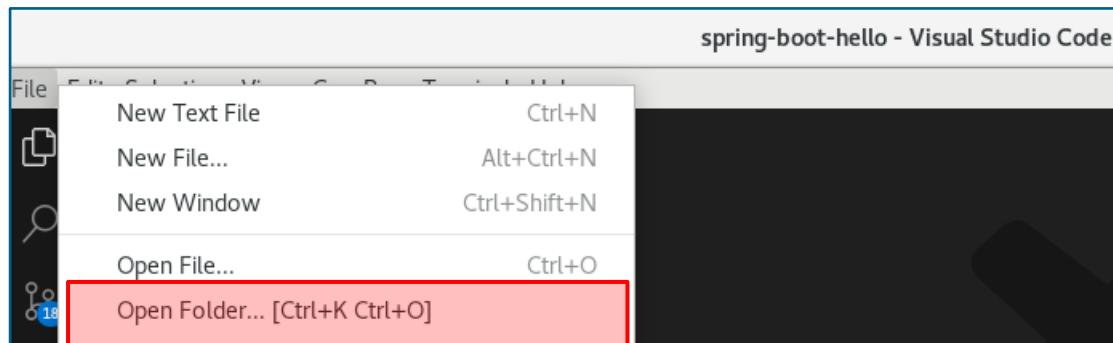
Aproveite que já está no terminal e copie o novo arquivo JAR gerado para a pasta Downloads

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL AZURE ⚡ bash + □ 🗑 ...
```

- [devops@orlinux8 spring-boot-hello]\$ ls -l target
total 20332
drwxrwxr-x. 3 devops devops 17 jun 27 15:00 classes
drwxrwxr-x. 3 devops devops 25 jun 27 15:00 generated-sources
drwxrwxr-x. 3 devops devops 30 jun 27 15:00 generated-test-sources
drwxrwxr-x. 2 devops devops 28 jun 27 15:00 maven-archiver
drwxrwxr-x. 3 devops devops 35 jun 27 15:00 maven-status
-rw-rw-r--. 1 devops devops 20815105 jun 27 15:00 spring-boot-complete-0.0.1-SNAPSHOT.jar
-rw-rw-r--. 1 devops devops 3820 jun 27 15:00 spring-boot-complete-0.0.1-SNAPSHOT.jar.original
drwxrwxr-x. 2 devops devops 119 jun 27 15:00 surefire-reports
drwxrwxr-x. 3 devops devops 17 jun 27 15:00 test-classes
- [devops@orlinux8 spring-boot-hello]\$ cp target/spring-boot-complete-0.0.1-SNAPSHOT.jar ~/Downloads
- [devops@orlinux8 spring-boot-hello]\$
- [devops@orlinux8 spring-boot-hello]\$



Abra o diretório **Downloads** pelo VSC

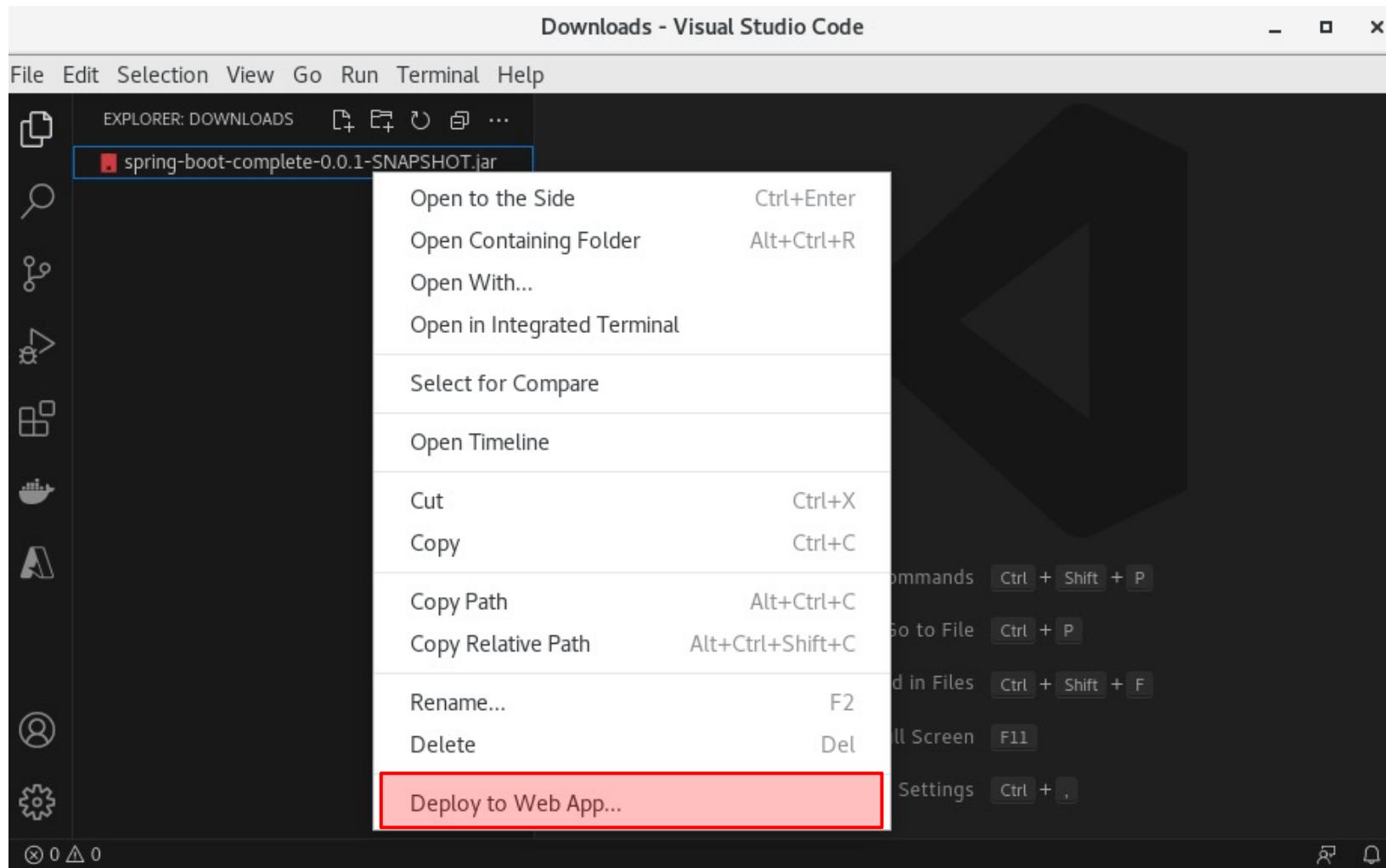


Instalar Plug-in Azure no VSC e Realizar testes



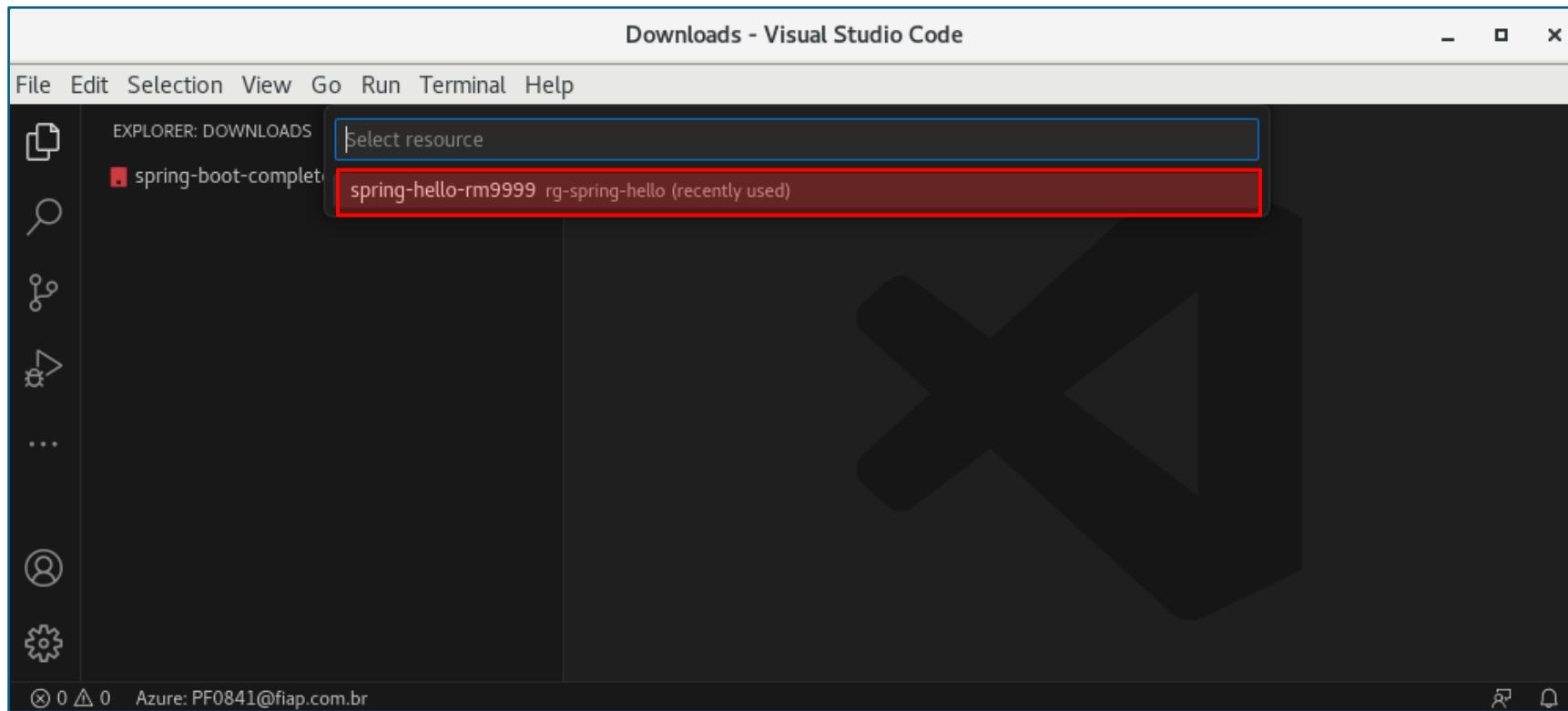
FIAP

Clique com o botão direito do mouse no arquivo JAR e escolha **Deploy to Web App...**



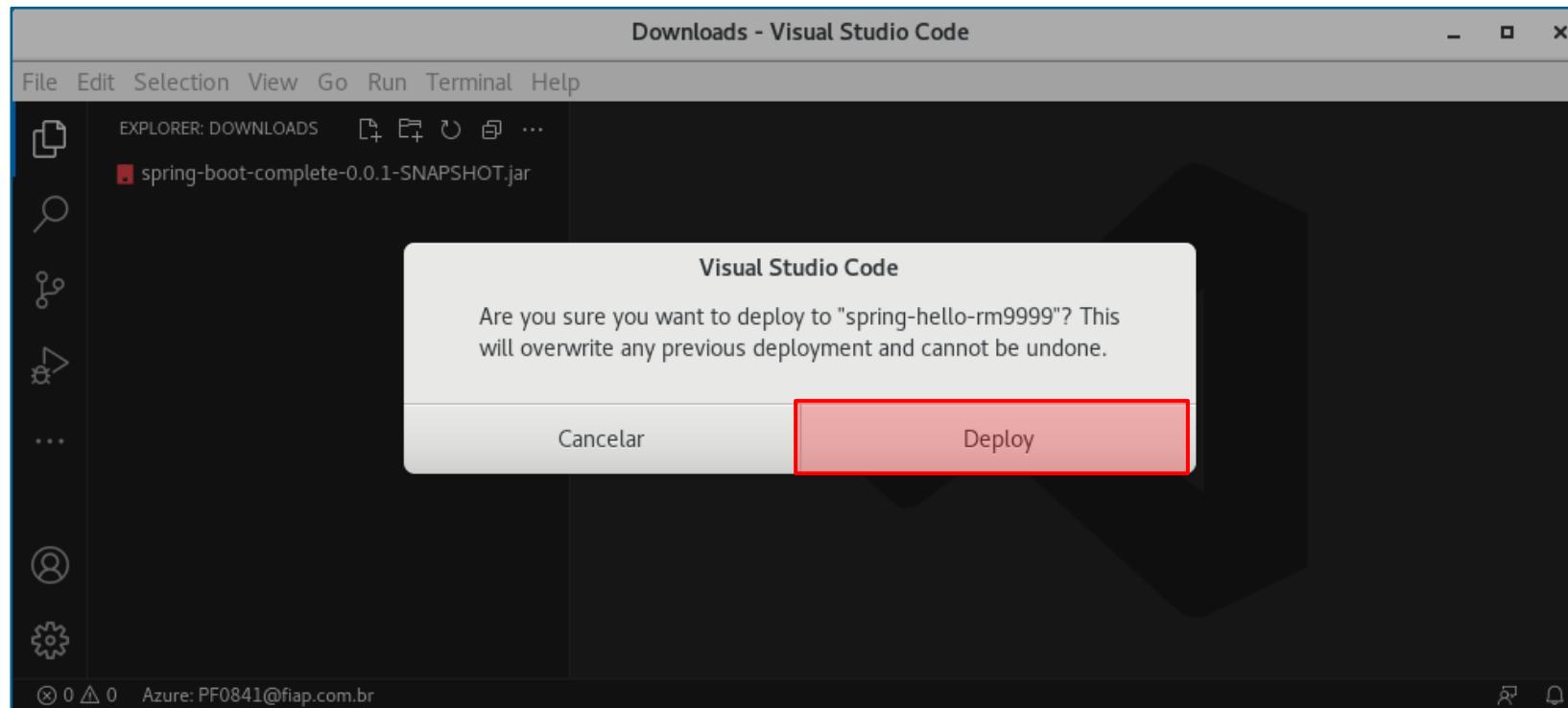


Escolha um Serviço de Aplicativo na lista para o Deploy desse Pacote
Em nosso caso só irá aparecer o que acabamos de criar





Se já existir algum Deploy efetuado anteriormente clique em **Deploy**

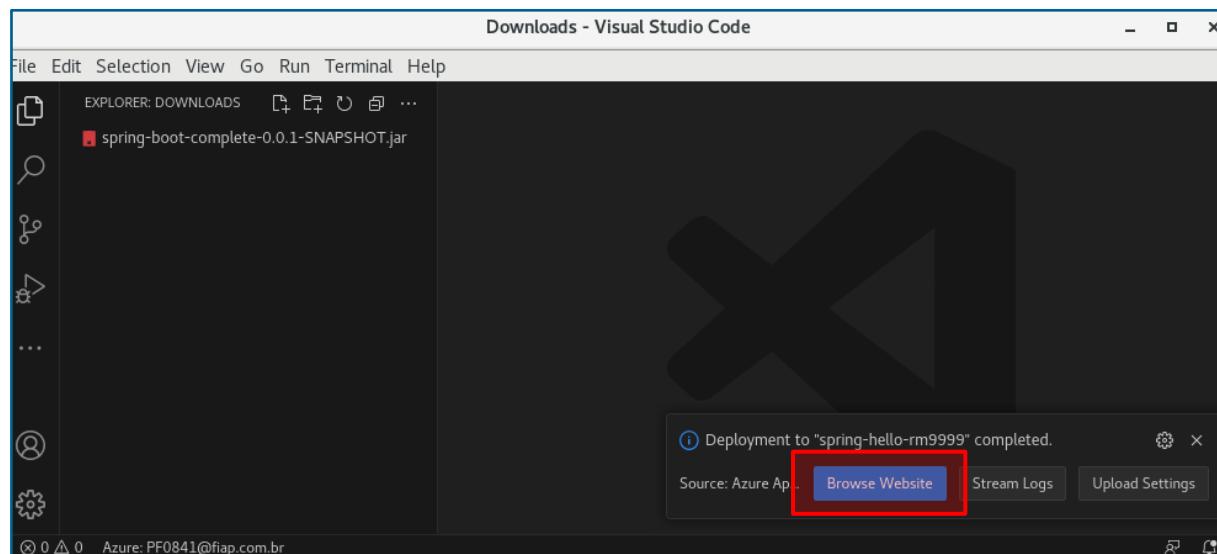
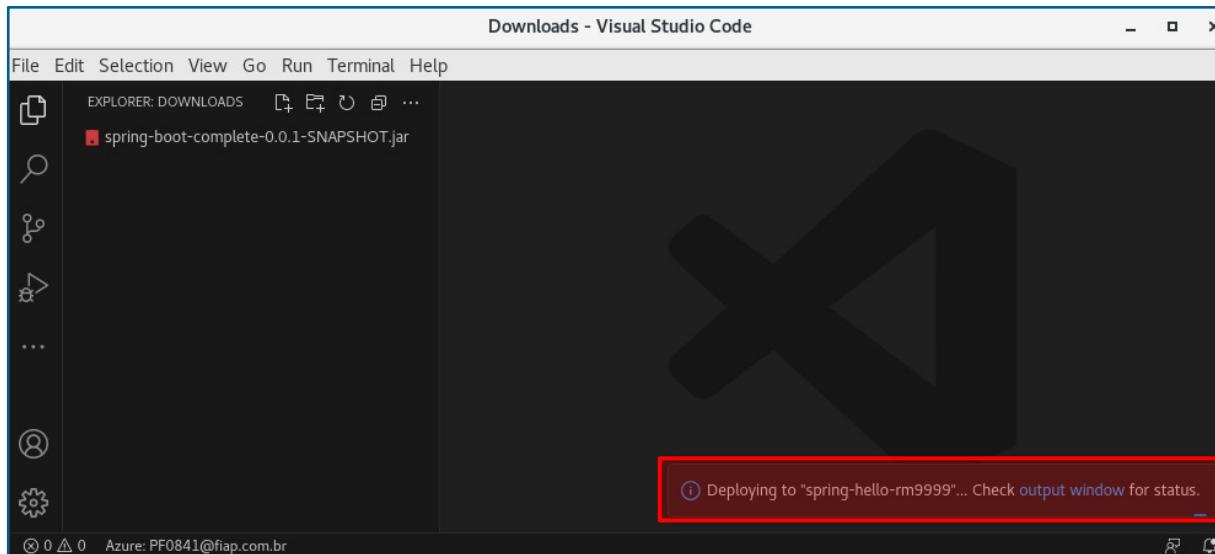


Instalar Plug-in Azure no VSC e Realizar testes



FIAP

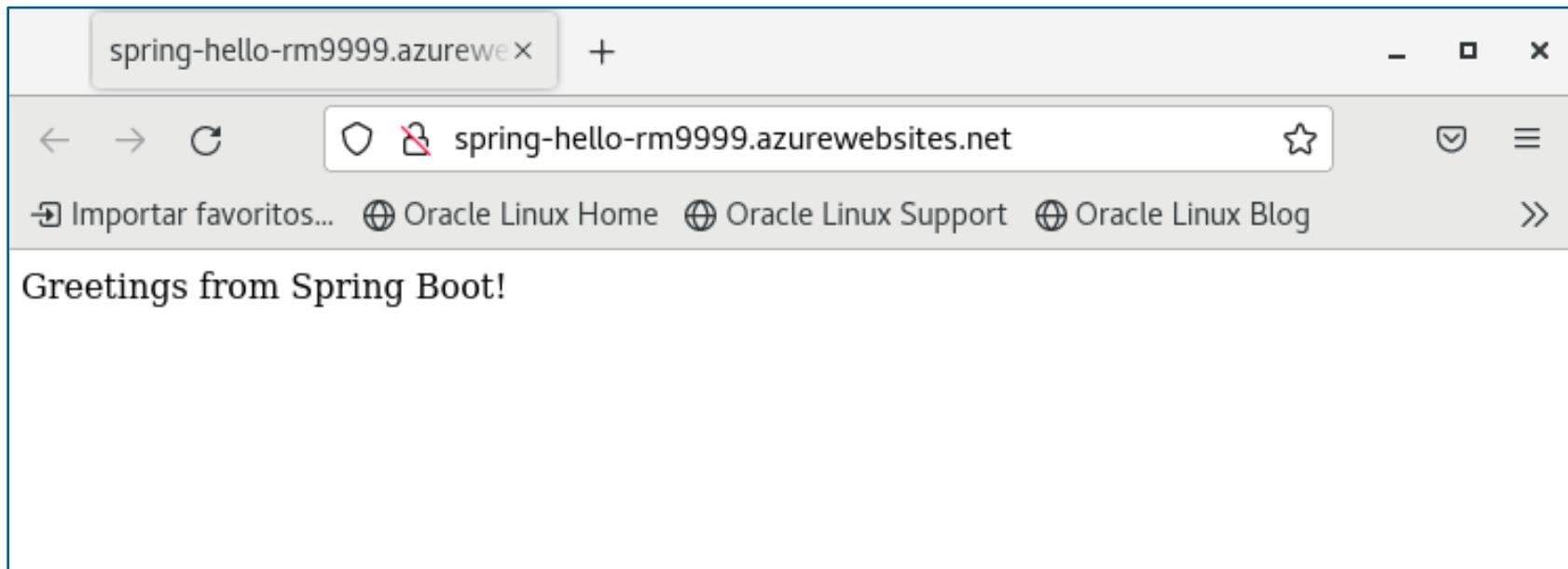
Aguarde o Deploy e realize os testes em Nuvem





Pode ser que demora alguns instantes para a mudança ser efetivada

Tente um Reload caso a alteração efetuada não apareça no primeiro acesso (é o tempo de reiniciar e ativar o App no recurso)

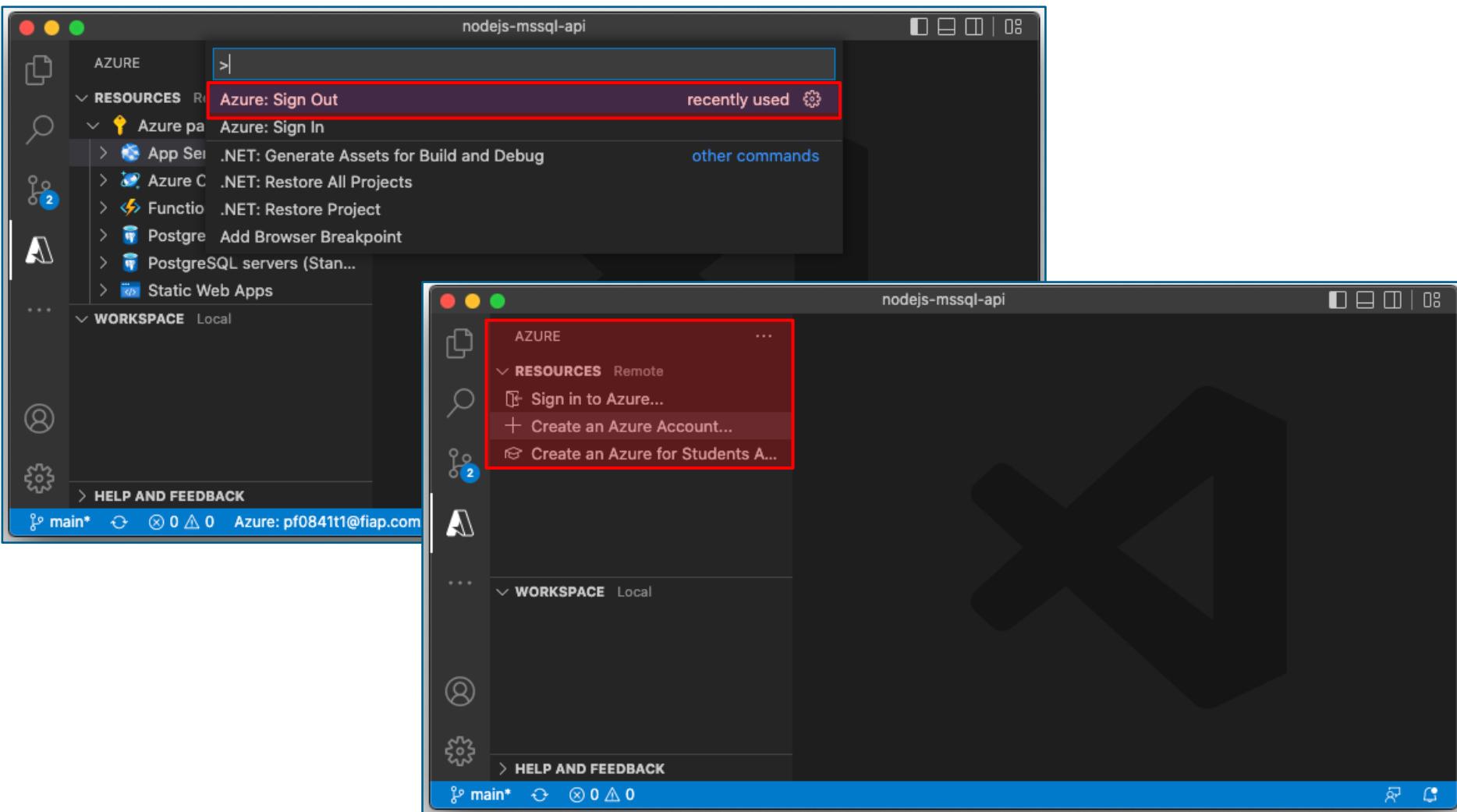


Veja os logs de Deploys que realizamos no **Serviços de Aplicativos** no Portal, em **Centro de Implantação → Logs**

The screenshot shows the Microsoft Azure portal interface for the 'spring-hello-rm9999' App Service. The 'Logs' tab is selected in the navigation bar. The deployment history table lists four successful deployments on Tuesday, June 27, 2023. The commit IDs for these deployments are highlighted with red boxes: d4072b5, 69b5dfe, e028e57, and a1edb84.

Time	Commit ID	Commit Author	Status	Message
Tuesday, June 27, 2023 (4)	d4072b5	ms-azuretools-vscode	Success (Active)	Created via a push deployment
	69b5dfe	ms-azuretools-vscode	Success	Created via a push deployment
	e028e57	ms-azuretools-vscode	Success	Created via a push deployment
	a1edb84	ms-azuretools-vscode	Success	Created via a push deployment

No menu do VSC vá em: **View -> Command Palette** e clique em **Azure: Sign Out (opcional)**



Após as verificações, exclua o Grupo de Recursos criado em nosso exercício no Portal. **Não deixe nenhum recurso**

The screenshot shows the Microsoft Azure portal interface. On the left, the sidebar lists various resource group management options like Visão geral, Log de atividade, IAM, Marcações, Visualizador de recursos, and Eventos. Under Configurações, it includes Implantações, Segurança, Políticas, Propriedades, and Bloqueios. The main content area displays the 'rg-spring-hello' resource group details. It shows two resources: 'PlanoServiçoSpring' and 'spring-hello-rm9999'. A modal window titled 'Excluir um grupo de recursos' is open, containing a warning message about permanent deletion and a confirmation section. The 'Excluir' button is highlighted with a red box. At the bottom of the modal, there's a text input field with 'rg-spring-hello' typed into it, followed by 'Excluir' and 'Cancelar' buttons.

Copyright © 2023 Prof. João Menk

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor)