Introduction

Scenario

Imagine that you recently heard about Azure Spring Cloud, a new managed service for running Spring Cloud apps on Azure. You would like to try it out with a sample Spring application, which already exists, called PiggyMetrics. You are interested in setting up a new service, configuring and deploying PiggyMetrics, and trying out a few other features.

Task 1: Provision the Service

Navigate to the Azure portal:
https://ms.portal.azure.com/?microsoft_azure_marketplace_ItemHideKey=AppPlatformExtension#blade/Microsoft_Azure_Marketplace/MarketplaceOffersBlade/selectedMenuItemId/home/searchQuery/spring

Note: If you have an Azure account and are logged into the portal, please use a private browsing window.

Log into the Portal with the following credentials:

- Username: mstest_chpay@outlook.com
- Password: 1234567890MS

Commented [IS1]: It looks like this account is used by lots of people, so we won't be able to change the password every time, right? If so, we need to remember to ask them to explicitly log out of the portal. And also somehow ensure they don't have this information available anymore…

Provision a new Azure Spring Cloud (ASC) service with the following details:

- **Name:** *Choose your own*
- **Subscription:** Java Tooling Tests with TTL = 7 Days
- **Resource Group:** rg-usabilitytesting
- **Location:** *Choose your own*

Task 2: Configure your Application

To save time, we have pre-created an Azure Spring Cloud (ASC) service for you called **asc-usabilitytesting.**

In the next few tasks, you will be asked to configure and deploy an application called PiggyMetrics. Application configuration for PiggyMetrics is stored in the following Git repository: https://github.com/xscript/piggymetrics-config.

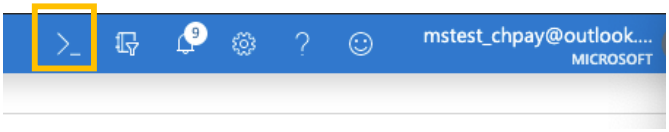
Task: Set up the config server in the **asc-usabilitytesting** service so that the applications read configuration data from the Git repository above.

Task 3 Build and Deploy three Applications

Build and deploy three Azure Spring Cloud applications (gateway, account-service, and auth-service) using the Azure CLI.

3.1

Open the Azure Cloud Shell



Clone git repository by running the following command:

```
git clone https://github.com/xscript/PiggyMetrics
```

Change directory and build the project:

```
cd PiggyMetrics
```

```
mvn clean package -DskipTests (06:37 min)
```

Log in to Azure CLI and set your active subscription. Choose the subscription which has been whitelisted & substitute your subscription ID in the following command:

```
# Login to Azure CLI
az login

# List all subscriptions
az account list -o table

# Set active subscription
az account set --subscription <target subscription ID>
```

Configure the default resource group name and service instance name. Substituting your source group name and service instance name in the following commands:

```
az configure --defaults group= rg-usabilitytesting
```

```
az configure --defaults spring-cloud=<service instance name>
```

3.2

Create a deploy the gateway application via the Azure Portal UI

3.3

Create and deploy the account-service and auth-service applications using Azure cloud shell

```
az spring-cloud app create -n account-service
```

```
az spring-cloud app deploy -n account-service --jar-path ./account-service/target/account-service.jar
```

```
az spring-cloud app create -n auth-service
```

```
az spring-cloud app deploy -n auth-service --jar-path ./auth-service/target/auth-service.jar
```

Task 4: Restart

Restart the gateway application.

Task 5: Assign a public endpoint

Assign a public endpoint to the gateway application and access that endpoint.

Task 6: Adjust your application's resources

Make it so that the gateway app is running on 3 instances with 3 CPU and 2GB of memory each.