Assignment: Deploying microservices to AKS

In this assignment you will deploy the Inventory and Trading microservices to your AKS cluster.

Deploy the microservices to AKS

In the Inventory and Trading repositories:

1. Create a kubernetes directory

2. In the new directory, create the Kubernetes yaml configuration file for the microservice
   - The easiest way is probably to get a copy of catalog.yaml and modify it accordingly
   - If you start from a catalog copy, make sure all references to catalog properly updated
   - Use the latest container image version you deployed for the microservice to ACR
   - Use the correct port for containerPort and the health probes.
     - Inventory: 5004
     - Trading: 5006

3. Apply your new configuration yaml file to your AKS cluster

4. Make sure you update the microservice README with the command you used to apply the yaml file.

5. Verify the new pod reaches the RUNNING state

6. Commit and push your changes

Add an API Gateway route for the new microservices

In the Play.Infra repository:

7. Update mappings.yaml to include new mappings Inventory and Trading

8. Apply mappings.yaml to your AKS cluster

9. Commit and push your changes

Test the deployed microservices

In Postman:

10. Make a POST or PUT request to the Catalog REST API to create or update an item, so that the new microservices (Inventory and Trading) are notified of the new or updated item(s).

11. Make a PUT request to the Users REST API in the Id

12. Add new variables to the Local and Cloud environments that point to Inventory and Trading.

13. Test any of the REST API operations on Inventory and Trading using the new environment variables

    - ❖ If you would like to test the deployed microservices using any Catalog item created before deploying Inventory and Trading, make sure you first update that item by sending a PUT request to the Catalog /items REST API so that Inventory and Trading get notified of the existence of that item.

    - ❖ If you would like to test Trading operations using an existing player, make sure you first update that player by sending a PUT request to the Identity /users REST API so that Trading gets notified of the amount of gil that player has available.