Deploying the Catalog microservice to AKS

Let's deploy the Catalog microservice to our Kubernetes cluster.

In Play.Catalog

1. Update catalog.yaml adding a Deployment and Service (start from copy of identity.yaml):

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: catalog-deployment
spec:
  selector:
    matchLabels:
      app: catalog
  template:
    metadata:
      labels:
        app: catalog
        azure.workload.identity/use: "true"
    spec:
      serviceAccountName: identity-serviceaccount
      containers:
        - name: catalog
          image: playeconomy.azurecr.io/play.catalog:1.0.4
          imagePullPolicy: Always
          env:
            - name: ServiceSettings__MessageBroker
              value: SERVICEBUS
            - name: ServiceSettings__KeyVaultName
              value: playeconomy
            - name: ServiceSettings__Authority
              value: https://playeconomy.eastus.cloudapp.azure.com/identity-svc
          resources:
            limits:
              memory: "128Mi"
              cpu: "150m"
          ports:
            - containerPort: 5000
          livenessProbe:
            httpGet:
              path: /health/live
              port: 5000
            initialDelaySeconds: 10
          readinessProbe:
```

```
        httpGet:
          path: /health/ready
          port: 5000
        initialDelaySeconds: 10

---
apiVersion: v1
kind: Service
metadata:
  name: catalog-service
spec:
  type: ClusterIP
  selector:
    app: catalog
  ports:
    - port: 80
      targetPort: 5000

---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: catalog-serviceaccount
  annotations:
    azure.workload.identity/client-id: 0f44e49e-7c49-44af-8eaf-b3c4aa9d2793
  labels:
    azure.workload.identity/use: "true"
```

2. Deploy the updated catalog.yaml:

```
kubectl apply -f .\kubernetes\catalog.yaml -n $namespace
```

3. Start waiting for pod to reach Running state

In Play.Infra

"So, here we are back in Infra…"

4. Add the catalog mapping:

```
---
apiVersion: getambassador.io/v3alpha1
kind: Mapping
metadata:
  name: catalog-mapping
```

```
spec:
  hostname: playeconomy.eastus.cloudapp.azure.com
  prefix: /catalog-svc/
  service: catalog-service.catalog
```

5. Apply mapping.yaml
6. Commit and push

In Postman

7. Add catalogBaseUrl to the Local and Cloud Postman environments
   https://localhost:5001
   https://playeconomy.eastus.cloudapp.azure.com/catalog-svc

8. Select the Cloud environment

9. Open the POST /items request in Catalog collection

10. Ensure the auth scopes include catalog.fullaccess

11. Get an access token

12. Update the POST Url to use catalogBaseUrl.

13. Send a POST request to create a Potion:

```
{
   "name": "Potion",
   "description": "Restores a small amount of HP",
   "price": 5
}
```

14. Open the GET /items request

15. Update the GET Url to use catalogBaseUrl and send it

16. Commit and push

In the next assignment you will deploy the Inventory and Trading microservices.