K8S Deployment helper

Pre-requisites:

1. Make sure got the access to all the argoid-related docker images.

Guidelines:

1. Maintain k8s manifest files for each microservice file separately.

Create a Kubernetes namespace :

wiser-search-ns.yaml

```
apiVersion: v1
kind: Namespace
metadata:
  name: wiser-search-ns
```

Create a Kubernetes secret file to store your ACR registry credentials:

```
kubectl create secret docker-registry <acr-registry-secrets> \
    --namespace wiser-search-ns \
    --docker-server=<ACR-URL> \
    --docker-username=<ACR-USER> \
    --docker-password=<ACR-Pwd>
```

Create a Kubernetes configuration file to define the environment variables for the application:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: search-engine-config
data:
  MIN_HEAP_SIZE: "-Xms2g"
  MAX_HEAP_SIZE: "-Xmx8g"
  KAFKA_SERVER_URL: "<Kafka-Host>:<Port>"
  SQL_SERVER_URL: "<SqlServer-Endpoint>"
  BLOB_URL: "<BLOB Endpoint>"
```

Create a Kubernetes deployment file for the search-engine application:

search-engine-deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: search-engine
  namespace: wiser-search-ns
spec:
  selector:
    matchLabels:
      app: search-engine
  replicas: 1
  template:
    metadata:
      labels:
        app: search-engine
    spec:
      containers:
        - name: search-engine
          image: argoidpepsico.azurecr.io/search-engine:0.0.7-snacks_com
          imagePullPolicy: Always
          ports:
            - containerPort: 8092
            - containerPort: 40024
          envFrom:
            - configMapRef:
                name: search-engine-config
      imagePullSecrets:
        - name: <acr-registry-secrets>
```

Create a Kubernetes service file for the application:

search-engine-service.yaml

```
---
apiVersion: v1
kind: Service
metadata:
  name: search-engine
  namespace: wiser-search-ns
  labels:
    app-svc: search-engine
spec:
  type: ClusterIP
  selector:
    app: search-engine
  ports:
    - name: http
      port: 80
      targetPort: 8092
    - name: jmx
      port: 40024
      targetPort: 40024
```

Create a Kubernetes ingress file for the application and point to the Azure Application Gateway with the domain name :

search-engine-ingress.yaml

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: search-engine-ingress
  namespace: wiser-search-ns
  annotations:
    appgw.ingress.kubernetes.io/appgw-ssl-certificate: <SSL Certificate
Name>
    kubernetes.io/ingress.class: azure/application-gateway
    appgw.ingress.kubernetes.io/health-probe-status-codes: "200-599"
spec:
  rules:
  - host: <domain-name>
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: search-engine
            port:
              number: 8092
```