

# Kubernetes

Recitation 7, CMU 17-313, Fall 2020

**Goal:** During this recitation, we will explore deploying Mayan EDMS with Kubernetes and deploy an application using a typical microservice architecture.

## Task:

1. Ensure you have Docker installed.
2. In the Docker configuration, enable Kubernetes.
3. Using the [Mayan Docker Compose file](#), create a Kubernetes configuration for deploying Mayan-EDMS.
  - a. For this task, you should be creating 6 files (or, 6 configurations in a single file):
    - i. *mayan-edms-service.yaml*:  
This should be a LoadBalancer service that maps the correct ports for Mayan EDMS.
    - ii. *mayan-edms-deployment.yaml*:  
This file should provide the deployment information for mayan-edms, including how to setup the required environment variables, ports, and required image.
    - iii. *postgresql-service.yaml*:  
This should be a LoadBalancer service that maps the correct ports for PostgreSQL.
    - iv. *postgresql-deployment.yaml*:  
This file should provide the deployment information for postgres, including how to setup the required environment variables, ports, and required image.
    - v. *redis-service.yaml*:  
This should be a LoadBalancer service that maps the correct ports for Redis.
    - vi. *redis-deployment.yaml*:  
This file should provide the deployment information for Redis, including how to setup the required environment variables, ports, and required image.
4. *Hints:*
  - a. *(It might make sense to try starting with Redis first, should be the easiest, then do Postgres, then finally do Mayan.)*
  - b. *Skip anything about volumes: for an actual deployment we would need this to ensure our state persists across reboots, but we will skip this for now.*
  - c. *Feel free to replace any environment variables like `POSTGRES_USER`, etc. with hardcoded values to make this exercise easier.*
  - d. *Feel free to run Redis without a password (i.e., you can ignore everything in the command section of the Docker Compose file.)*
  - e. *You'll probably need to do the LoadBalancer service configurations after their related Deployment configurations as one references the other.*
  - f. *Ask the instructor if get the stuck!*
5. Try running your configuration with `kubectl`.

## Examples:

Here's what a LoadBalancer service could look like:

```
apiVersion: v1
kind: Service
metadata:
  name: ?
  labels:
    app: ?
spec:
  type: LoadBalancer
  ports:
  - port: ?
    name: ?
    targetPort: ?
  selector:
    app: ?
```

Here's what a deployment could look like:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: ?
spec:
  replicas: 1
  selector:
    matchLabels:
      app: ?
  template:
    metadata:
      labels:
        app: ?
    spec:
      containers:
      - name: ?
        image: ?
        imagePullPolicy: Always
        ports:
        - containerPort: ?
        env:
        - name: ?
          value: ?
```