Software Engineer (.NET 5, REST, OpenAPIv3)

Challenge

1. The API and Persistence

You have been tasked with creating a C# .NET 5.0 API that is compliant with OpenAPI spec V3.

The API represents a simplified Virtual Machine management service where Users, Machines and Applications are managed with the API. You are required to implement all operations for this API based on the API Requirements section below. The solution should have appropriate testing.

Operations on data models for the API should be persisted to a (local) database of your choosing.

2. Containerisation

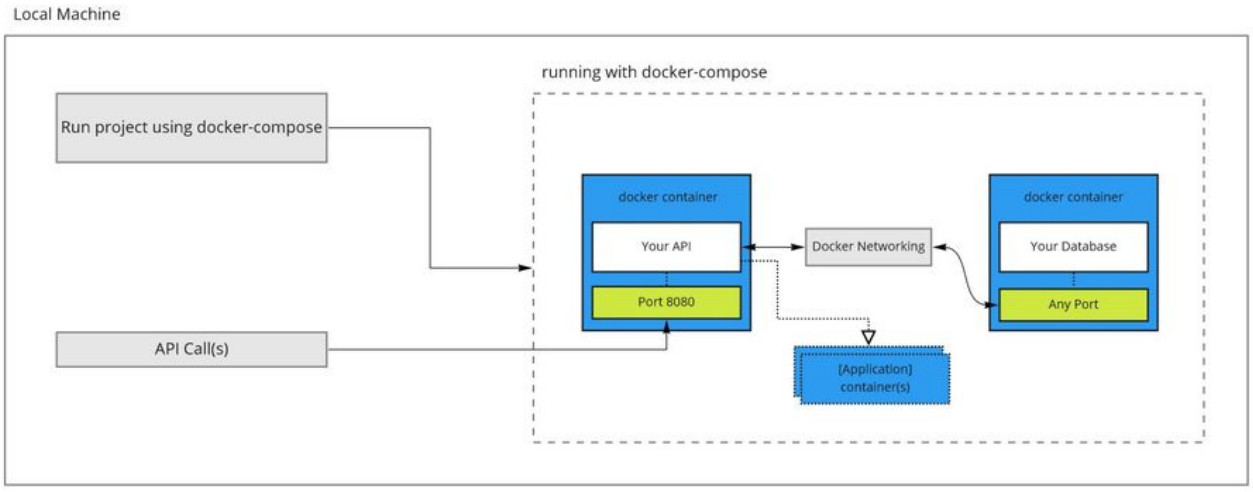
The solution should be containerised, built locally and deployed locally using docker-compose.

3. Running Applications (containers) and creating the K8S deployment

Once the API is deployed locally using docker-compose; an instance of Application can be `started` or `stopped`.

Using tooling of your choice, create Kubernetes deployment files for the service. Think about what is required to successfully run a service in Kubernetes reliably. Place these files in a `/deployment` directory. This deployment will be deployed into a Polystream test cluster after submission.

Architecture



API Requirements

This API should provide the following:

- Operations for a User.

- Operations for a Machine.
- Operations for an Application.
- Operations for Administration.
- **As an API consumer**
  - I should be able to register as a [User]
  - As a [User]; I should be able to create a [Machine] definition which I own
  - As a [User]; I should be able to create an [Application] definition which I own.
  - As a [User]; I should be able to create an [Application] that can be ran on a [Machine]
  - As a [User]; I should be able to query the API for information about my [Application(s)] and [Machine(s)]
- **As an API owner**
  - An API owner should be able to create or remove [User(s)], [Machine(s)] and [Application(s)]

Data Model

The data model for the API should contain at least the following classes. These classes are to be defined by you based on the requirements above.

```
User
Machine
Application
```

Example Application

```
{
  "id": "guid",
  "container" : "busybox",
  "state": "running"
}
```

Notes

Try to complete the task as quickly as possible.

The entire project (including database) should be able to run from docker-compose.

You must use C# .NET 5, docker and docker-compose for the task.

The project must run in docker-compose; both the API and the Database associated with it.

Deliverables

- A service that hosts a RESTful API
- An archive (zip) of your entire project, including all YAML (Dockerfile & docker-compose)
  - with a `/deployment` directory containing kubernetes deployment config
- A brief document detailing your approach and the reason behind
  any design decisions you feel need explaining
- Please supply all source code (with or without binaries)