Requirements:

1. Docker Desktop should be installed and running on your local computer.
2. Minikube must be installed on your local computer. Minikube is a single-node Kubernetes cluster used for testing purposes.
3. You must have a Kubernetes cluster running on a cloud provider such as Azure. The simplest way to set up a Kubernetes cluster is with Azure Kubernetes Service (AKS) as it only requires you to define the number of nodes and their computational resources, and all services are installed by AKS.

Open the CMD prompt and follow these steps:

1. Change the path to the folder where the YAML files are located: **gps-yaml**.

cd C:\VehicleTrackingKubernetes\gps-yaml

1. Connect to your Azure Kubernetes cluster (following AKS instructions) and deploy the following files:

kubectl apply -f <filename>.yaml

* + **gps-server.yaml**
  + **gps-server-sv.yaml**
  + **tracking-server.yaml**
  + **tracking-server-svlb.yaml**
  + **mongo.yaml**
  + **redis.yaml**
  + **mongo-sv.yaml**
  + **redis-sv.yaml**
  + **redis-pvc.yaml**
  + **mongo-pvc.yaml**
  + **redis-config.yaml**
  + **mongo-config.yaml**

1. Once these services are running in your Azure Kubernetes cluster, start a Minikube instance on your local machine:

minikube start

1. Open the **gps-server.yaml** file and edit the following environment variables accordingly. Refer to the comments in the file for an understanding of their meanings:

* **INIT\_LAT** - Initial decimal degrees latitude where the vehicle starts.
* **INIT\_LONG** - Initial decimal degrees longitude where the vehicle starts.
* **CLIENT\_ID** - Client/car identifier, e.g., (1, 2, etc.).
* **ENDPOINT** - The entry point to the "gps-server" service, which is provided when you run the **gps-server-sv.yaml** file (a Service-load balancer). This remains the same for all clients/cars.

1. Finally, for each local machine running Minikube, execute the node command with the updated environment variables. Your car marker with its ID should appear on the map when accessing the endpoint defined in the source files for tracking and GPS server, e.g., **<http://ip:port>**. Similarly, for accessing the vehicle route: **<http://ip:port/route>**, and for adding stops: **<http://ip:port/stops>**. Note that "ip:port" must be changed to the actual IP and port defined when deploying the load balancer services.

kubectl apply -f gps-server.yaml