The goal of these projects is to help you familiarize yourself with the core technologies underlying microservices platforms. In particular, you will deploy a microservice and enhance the platform with additional functionality to improve diagnosis and efficiency. These projects will slowly introduce you to concepts in a manner that parallels and mirrors our discussions in class. For many of these technologies, there are online resources and documentation (e.g., tutorials and walkthroughs) for setting up and integrating them. In these assignments, you should consult them and use them as aids in your assignments.

While there are a number of sample microservices available online, we will be using Google’s microservices-demo which naturally interoperates with a plethora of existing technologies. Note: promising alternatives, e.g., SockShop do not support sufficient tracing points, and BookInfo is limited to four nodes. Google’s microservice-demo is sufficiently complex, and it interoperates with several of the existing technologies.

**Hand-Ins:** You will either handin screenshots of commands or a PDF project report. For the project reports, you will use the microservices’ load generator to generate traffic and analyze the impact of your changes on the performance of the microservices. Do the new additions impact latency or throughput? Does it change the utilization and load behavior of the microservice deployment?

1 **HW 1: Microservices Hello World!**

- Assigned: Feb 4
- Due: Feb 11
- No Groups: Solo project.
- Hand-In: Screenshots.

In the first project, you will follow instructions to deploy the microservices-demo on Istio. The [github repository](https://github.com/) includes detailed documentation for setting up an initial microservice. You should follow option 1 or option 3. Either will help you setup your service.

1.1 **What to HandIn:**

Screenshots of your services. Email them to me.
2 HW 2: Enhanced Microservices!

- Assigned: Feb 13
- Due: March 3
- Groups of two.
- Hand-In: Codebase + project report.

In the second assignment, we will break out from using the traditional microservice deployment. In particular, we will make two key changes:

2.1 API Gateway:
In particular, you will add an API-gateway (i.e., Ambassador) to expose several of your internal services to the outside world. We will expose the following services: FrontEnd. There are several API-gateway platforms in addition to Ambassador, and you are free to choose one.

**Note:** Initially, the goal was to instrument both HTTP (FrontEnd Service) and gRPC (RecommendationService, AdService, and CartService); however, the newest version of Ambassador has issues with gRPC.

2.2 Authentication:
Next, you will setup an authentication service for incoming requests into your infrastructure. There are several options for doing this (ranked from 'simple' to 'hard'):

- OPA Finally, can setup OPA (Open Policy Agent) as an external entity to perform authorization. The most prominent link for OPA is [here](https://www.getambassador.io/user-guide/auth-tutorial/)

- **Authentication Services:** Ambassador comes with built in constructs for an external authentication services called 'AuthService'. You can find a tutorial on how to set it up [here](https://www.getambassador.io/user-guide/auth-tutorial/)

- **Ambassador Filters:** Alternatively, you can use the filter constructs to redirection request to an external Authentication service. You can find instructions on how to deploy and enhance your services to include Authentication [here](https://www.getambassador.io/user-guide/auth-tutorial/)

You will benchmark the FrontEnd Services using the load generator included in the github. Here is a direct link to the load **LoadGenerator**

2.3 What to HandIn:
(1) The Yaml files. (2) If you use the Ambassador filters, then the filter service. (3) A README files which: (a) includes the load generator results before you add the Authentication+API-Gateway after after you add them. (b) specifies the username/password supported. Note the different tutorials will lead to different username/passwords.