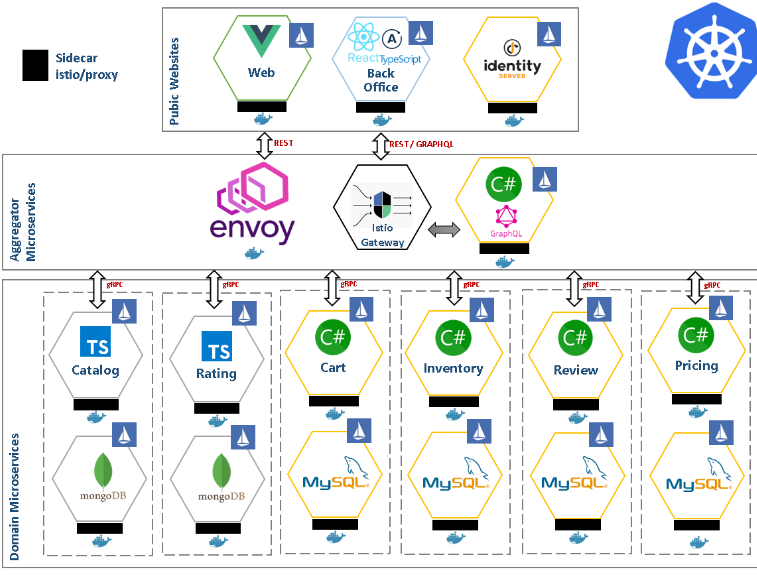
# High Level Software Architecture



Prerequisite

1. Window 10 – the OS for developing and building this demo application

2. Docker for desktop (Kubernetes enabled) – the easiest tool to run Docker, Docker Swarm and Kubernetes on Linux

3. Kubernetes /AKS – the app is designed to run on Kubernetes

4. Istio – application works on Istio services mesh

5. helm – the best package manager to find, share, and use software built for Kubernetes

6. .NET Core SDK 2.x - .NET Framework and .NET core, including ASP.NET and ASP.NET Core

7. nodejs 10.x JavaScript – Javascript superset of JavaScript that compiles to plain Javascript

8.identityserver – The Identity and Access Control solution for .NET core

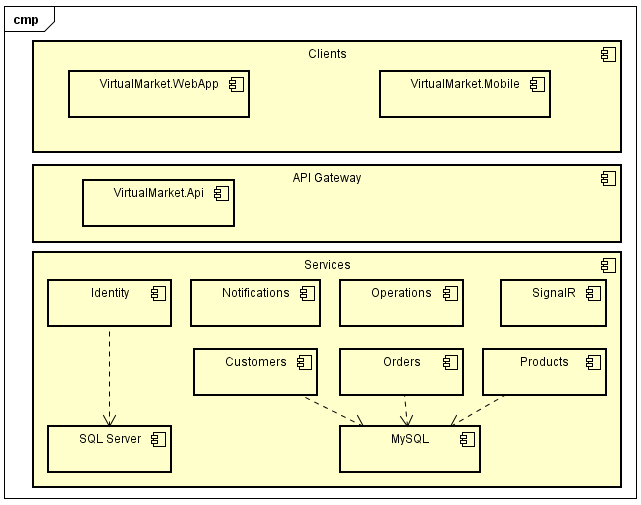
9.gRPC – a high-performance, open-source universal RPC framework

10. create – react - app: a modern web app by running one command

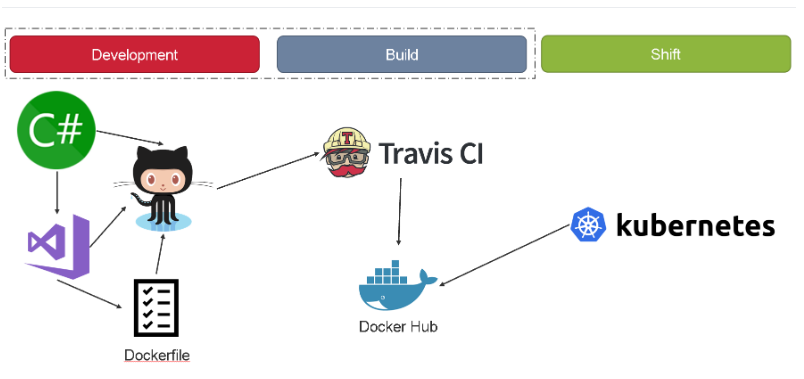
11. vue-cli – standard tooling for Vue.js development

12. Apollo-client – the best way to use GraphQL to build client applications

# Architecture



# CI/CD



**Consul** which have role a service discovery

is a distributed service mesh to connect secure, and configure service across any runtime platform and public or private cloud.

**Fabio** is an HTTP and TCP reserve proxy that configures itself with data from Consul

Traditional load balancers and reserve proxies need to be configure with a config file. The configuration contains the hostnames and paths the proxy is forwarding to upstream services. This process can be automated with tool like consul-template that generate config file and trigger a reload.

* Fabio work differently since it updates it routing table directly from the data stored in Consul as soon as there is a change and without restart and reloading.
* When you register a service in Consul all you need to add in is a tag that announces the paths the upstream service accepts, e.g: urlprefix-/user or urlprefix-/order and Fabio will do the test.

Jaeger Tracing

Jaeger is a CNCF distributed tracing system released by Uber that enables debugging, monitoring, and analysis of your services, and is based on OpenTracing API. It uses distributed context propagation, which essentially is the basis of distributed tracing, to assign metadata to request as they propagate through your system. List some advantages of using Jaeger:

* Performance and latency analysis.
* Service dependency analysis; you can view a DAG of your system in the UI
* Logs associated with each span
* Organization of logs into calling hierarchy
* Cost attribution
* Useful UI and libraries in Go, Node, C#, and others.
* Different levels of sampling can be configured to reduce load: constant , probabilistic, rate limiting, and remote.