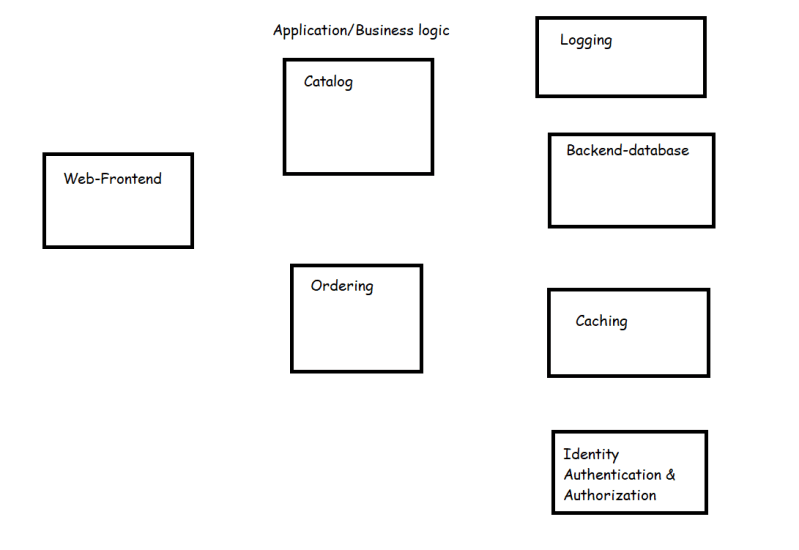
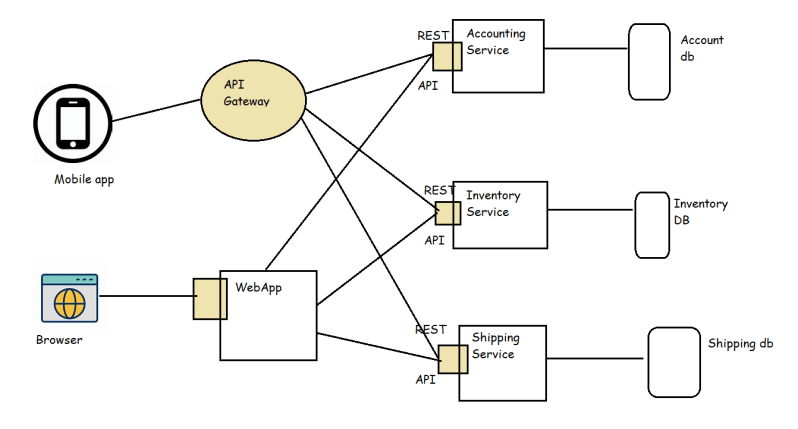
**Docker compose**

* This is a tool for defining an running multi container Docker applications
* All the modern applications will have the some/all of the following components 
* When all of these components work together we have a successful application
* Lets understand microservices
* To bringup our application we need to run multiple containers in a specific order.
* Docker compose is one approach to bring up multiple container of your application on a single docker host platform.
* Docker compose is widely used in Developer environments
* Docker stack is one more approach which helps to run multiple containers of your application on multi nodes.

Microservices

* This is an architectural pattern that structures and application as a collection of services that are
  + Highly maintable and testable
  + Loosely coupled
  + Independently deployable
  + Organized around business capabilities
* Simple microservice architecture 
* Now we can deploy each service in a container/vm Now this gives us the following benifits
  + We can have parallel developments for each service
  + Reusability of services can be done across applications
  + We can scale the individual services when there is load.
* The best way to run individual services is in containers and to manage the whole application deploymets we use orchestration platforms such as
  + Kubernetes
  + Docker Swarm
  + OpenShift

How to work with Docker compose

* In Docker compose we create a file called docker-compose.yaml and we specify the containers to be created, their network, their volumes as specified in docker compose specifications [Refer Here](https://docs.docker.com/compose/compose-file/)
* In with the code repository docker-compose.yaml is commited to make it easier to bring up multiple containers to run an application
* Docker-Compose is pre-installed when you install docker on mac and windows 10
* On Linux machines we need to install docker-compose.

1. **sudo curl -L https://github.com/docker/compose/releases/download/1.21.2/docker-compose-`uname -s`-`uname -m` -o /usr/local/bin/docker-compose**
2. **sudo chmod +x /usr/local/bin/docker-compose**
3. **docker-compose –version**

* In this lets try to create a httpd container and nginx container
* version: '3'
* services:
* httpd:
* image: httpd:latest
* ports:
* - 80:80
* nginx:
* image: nginx:latest
* ports:
* - 8080:80

docker-compose build

docker-compose up -d

docker container ls

docker image ls

FROM maven:3-jdk-8 AS mvn

RUN git clone https://github.com/wakaleo/game-of-life.git && cd game-of-life && mvn package

FROM tomcat:9-jdk8

COPY --from=mvn /game-of-life/gameoflife-web/target/gameoflife.war /usr/local/tomcat/webapps/gameoflife.war

EXPOSE 8080

version: "3.9"

services:

  gol:

    build: ./gol

    ports:

      - 8080:8080

    volumes:

      - type: volume

        source: loggol

        target: /var/log

    networks:

      - mynetwork

  spc:

    build: ./spc

    ports:

      - 8081:8080

    volumes:

      - type: volume

        source: logspc

        target: /var/log

    networks:

      - mynetwork

volumes:

  logspc:

  loggol:

networks:

  mynetwork:

    driver: bridge

docker-compose build

docker-compose up -d

docker container ls

docker image ls

Create a Dockerfile for a dotnet application

* Try to create a Dockerfile to containerize nop commerce application [Refer Here](https://docs.nopcommerce.com/en/installation-and-upgrading/installing-nopcommerce/installing-on-linux.html)
* [Refer Here](https://github.com/asquarezone/DockerZone/commit/a439cbf702e8e9004f9a9bb1c0c9b86027f0d3bc) for the dockerfile and compose file created.

Exercise:

* Try to run java based shopizer in containers [Refer Here](https://github.com/shopizer-ecommerce/shopizer)
* Try to create docker containers to run this sample services [Refer Here](https://github.com/umermansoor/microservices)