**YAML**

* YAML is key value or name value pair collection

<name>: <value>

* Lets try to understand with an example of sports club
  + name: ltsportsclub
  + branches:
    - Ameerpet
      * indoor
        + carrom
        + bowling
        + chess
        + ludo
      * outdoor
        + badminton
        + volley ball
        + swimming
    - KPHB
      * indoor:
        + carrom
        + bowling
        + chess
        + ludo
      * outdoor
        + badminton
        + volley ball
        + swimming
        + cricket
    - Gachibowli
      * indoor:
        + bowling
        + playstation
        + snooker
      * outdoor
        + badminton
        + volley ball
        + swimming
        + cricket
        + golf
* Sample yaml

---

name: ltsportsclub

headoffice:

address:

flatno: 407

building: mytrivanam

street: ameerpet main road

area: Ameerpet

city: Hyderabad

pincode: 500082

mobile: 9999999999

email: headoffice@ltsportsclub.com

branches:

- ameerpet:

sports:

indoor:

- carrom

- chess

- bowling

- ludo

outdoor:

- badminton

- volley ball

- swimming

address:

flatno: 408

building: ameerpet main

street: ameerpet main

city: Hyderabad

mobile: 88888888888

email: ameerpet@ltsportsclub.com

- kphb:

sports:

indoor:

- carrom

- chess

- bowling

- ludo

outdoor:

- badminton

- volley ball

- swimming

- cricket

address:

flatno: 408

building: kphb main

street: kphb main

city: Hyderabad

mobile: 88888888888

email: kphb@ltsportsclub.com

- gachibowli:

sports:

indoor:

- carrom

- bowling

- snooker

outdoor:

- badminton

- volley ball

- swimming

- cricket

- golf

- tennis

address:

flatno: 408

building: gachibowli main

street: gachibowli main

city: Hyderabad

mobile: 88888888888

email: gachibowli@ltsportsclub.com

* Note: YAML Tutorial <https://www.youtube.com/watch?v=ggOmHlnhPaM>
* YAML Types:
  + Simple:
    - Text
    - Number
    - boolean: true/false yes/no
  + complex
    - list/array (-)
    - dictionary/map/object
* Schema: yaml schema represents structure of yaml which we would be authoring

**Docker Compose**

* <https://docs.docker.com/compose/compose-file/compose-file-v3/> for docker compose reference
* Basic structure

---

version: string

services:

networks:

volumes:

configs:

secrets:

references

* + version: <https://docs.docker.com/compose/compose-file/compose-file-v3/#compose-and-docker-compatibility-matrix>
  + services: <https://github.com/compose-spec/compose-spec/blob/master/05-services.md>
  + networks: <https://github.com/compose-spec/compose-spec/blob/master/06-networks.md>
  + volumes: <https://github.com/compose-spec/compose-spec/blob/master/07-volumes.md>

**Activity: execute these by using docker cli**

* create a nginx container expose 80 port to any port
* create an apache container expose 80 to any port
* create an jenkins/jenkins container: expose 8080 to any free port

docker container run -d -P nginx

docker container run -d -P httpd

docker container run -d -P jenkins/jenkins

* docker-compose

---

version: "3.8"

services:

nginx:

image: nginx

ports:

- "80"

apache:

image: httpd

ports:

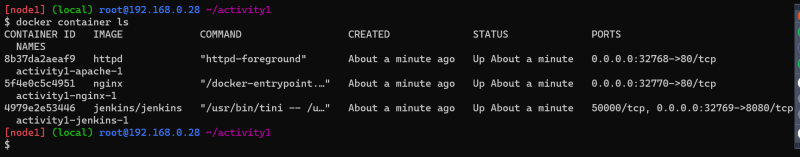
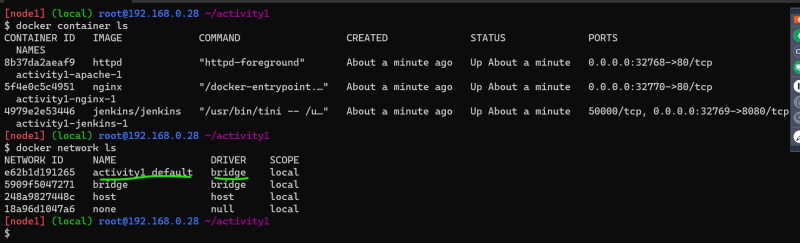
- "80"

jenkins:

image: jenkins/jenkins

ports:

- "8080"

* Now create and start containers  
    
  
* let’s view the containers and network  
    
  
* To remove whatever was created  
  