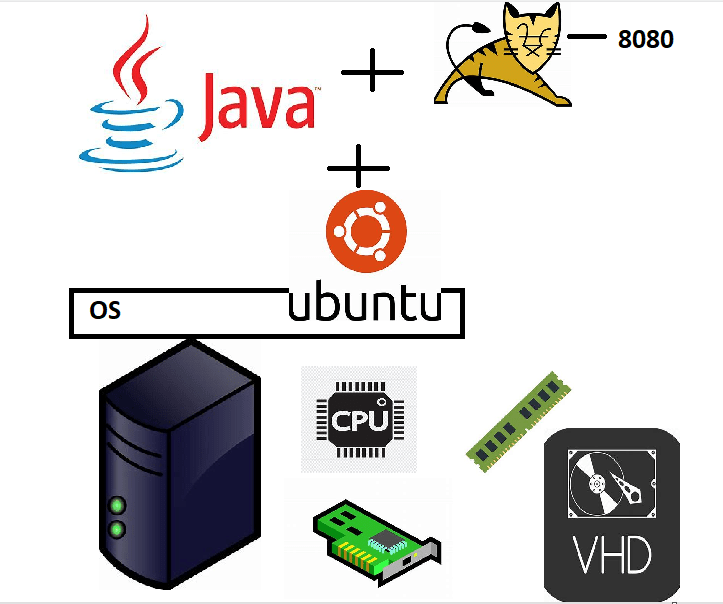
**Application**: Applications bring revenue to enterprises and to run applications we need servers, os etc

**Let’s create a Linux server and install tomcat in it**

* Create a linux server (AWS/Azure/GCP)  
  
* We have experimented in the linux vm
  + network interface gives network connectivity
  + cpu, ram and disk are available
  + to install softwares we have used package manager apt
* Commands

sudo apt update

sudo apt install net-tools openjdk-11-jdk tomcat9 -y

# to check network

ifconfig

# to check java

java -version

# to check tomcat

sudo systemctl status tomcat9

* We were able to exactly the similar operations inside container as well.

**Let me take another application**

* This is spring pet clinic application
* Lets try to run this application on linux
* <https://github.com/spring-projects/spring-petclinic>
* To build the code

sudo apt update

sudo apt install openjdk-17-jdk maven -y

git clone https://github.com/spring-projects/spring-petclinic.git

cd spring-petclinic

# java package

mvn package

java -jar target/spring-petclinic-3.0.0-SNAPSHOT.jar

* Docker way of working
  + We create a docker image (docker packaging format)
    - We need to create Dockerfile
    - push the image to registry (docker hub)
    - create the container using the image anywhere
* Dockerfile

FROM amazoncorretto:17-alpine-jdk

LABEL author=khaja

ADD target/spring-petclinic-3.0.0-SNAPSHOT.jar /springpetclinic.jar

EXPOSE 8080

CMD ["java", "-jar", "/springpetclinic.jar"]

* Create docker image

docker image build -t spc:1.0 .

* to create container

docker container run -d -P spc:1.0

docker container run -d -P spc:1.0

docker container run -d -P spc:1.0

**Next Steps**

* How is container able to give ip address/storage/process/cpu/ram etc
* Container architectures (3 versions)
* understanding image and container
* docker container life cycle
* containerization
* networking, storage aspects