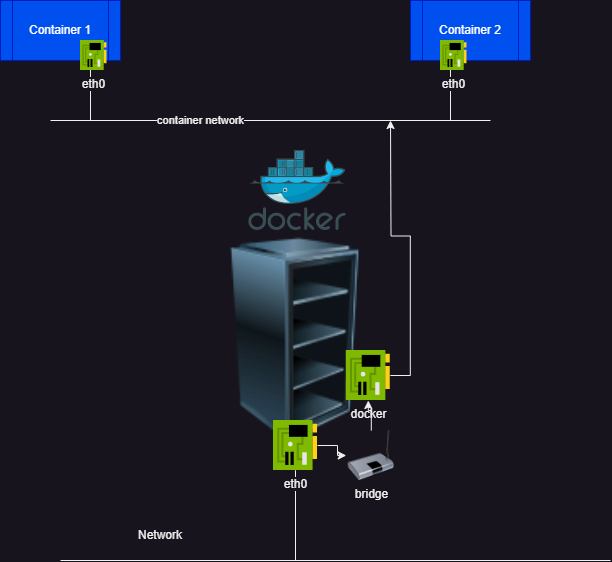
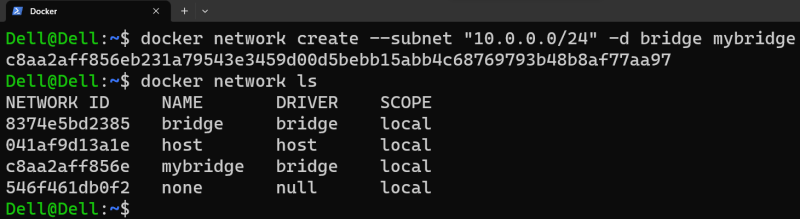
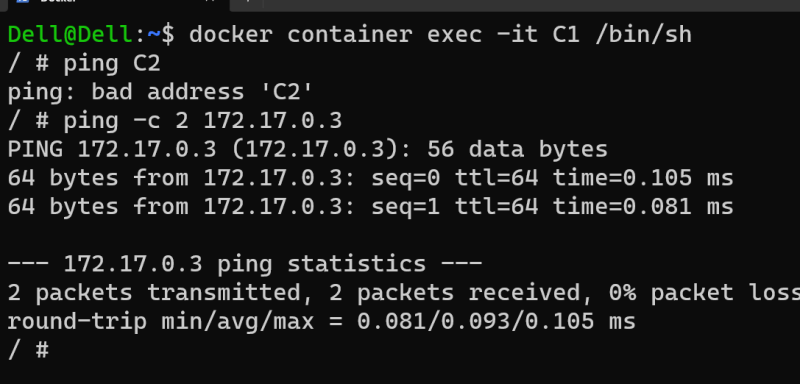
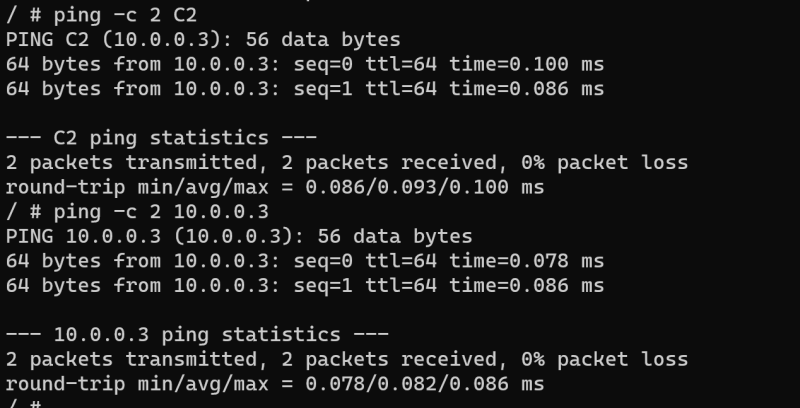
**Docker Networking**

* <https://github.com/moby/libnetwork/blob/master/docs/design.md> for container networking model
* Docker implements network with the libnetwork component which is implementation of CNM.
* Docker daemon interacts with lib network.
* Docker implements networking based on drivers
* <https://www.docker.com/blog/understanding-docker-networking-drivers-use-cases/> for drivers and use cases
* <https://directdevops.blog/2019/10/05/docker-networking-series-i/> this article from direct Devops
* Bridge: Bridge is generally a networking device that creates a single netowrk multiple network segments. Docker implements brigde network in linux as default network  
  
* Let’s create our own bridge network  
  
* let’s create two containers (C1 and C2) in default bridge, exec into C1 and ping C2 by name and ip  
  
* As we see, we are able to reach other container by its ip not by name.
* Lets attach container C1 and C2 to the bridge network which we created. Now ping  
  
* we can ping other container by name and ip address

**Activity1: Build a spring petclinic jar**

* Create a linux vm with atleast 2 vcpus and 4 GB RAM
* install java 17
* install maven
* clone spring petclinic

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

cd spring-petclinic

mvn package

* Building packages can be done as part of building images this lead to something which we call as multi stage building
* <https://docs.docker.com/build/building/multi-stage/> for multistage building
* Lets do that in phases, Create a new empty directory and add a Dockerfile

FROM maven:3-amazoncorretto-17 AS builder

COPY ./spring-petclinic /spring-petclinic

RUN cd /spring-petclinic && \

mvn package

# /spring-petclinic/target/spring-petclinic-3.1.0-SNAPSHOT.jar

FROM alpine:3.18.2

LABEL author=shaikkhajaibrahim

ARG JAVA\_PACKAGE=openjdk17-jdk

ARG USER=spc

ARG HOME\_DIR=/spc

ARG USER\_SHELL=/bin/sh

RUN apk update && \

apk add ${JAVA\_PACKAGE} && \

echo ${JAVA\_TEST}

RUN adduser -h ${HOME\_DIR} -s ${USER\_SHELL} -D ${USER}

USER ${USER}

WORKDIR ${HOME\_DIR}

COPY --from=builder --chown=${USER}:${USER} /spring-petclinic/target/spring-petclinic-3.1.0-SNAPSHOT.jar spring-petclinic-3.1.0-SNAPSHOT.jar

EXPOSE 8080

ENTRYPOINT ["java"]

CMD ["-jar", "spring-petclinic-3.1.0-SNAPSHOT.jar"]