**Creating Docker Image using Dockerfile**

* In this approach we create a file with instructions on what should docker image have.
* Using docker image build instruction we create a docker image for which the instructions in a file will be the input
* Any changes in the container are not supposed done directly, the new version of the image which is called as tag needs to built and distributed. This represents the immutable infrastructure
* docker images have the following naming convention

<registry-account>/<application-name>:<tag>

jenkins/jenkins:lts-jdk11

* Docker hub will not have registry account for official images

<application-name>:<tag>

nginx:1.12

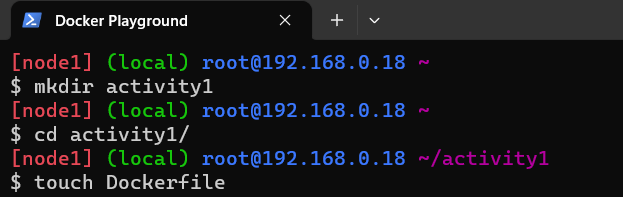
alpine:edge

* If the tag is not given docker daemon assumes it to be latest tag

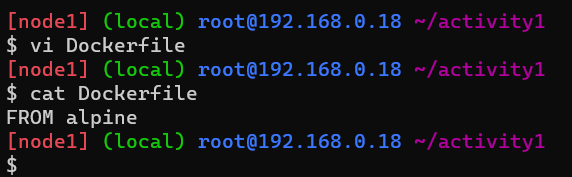
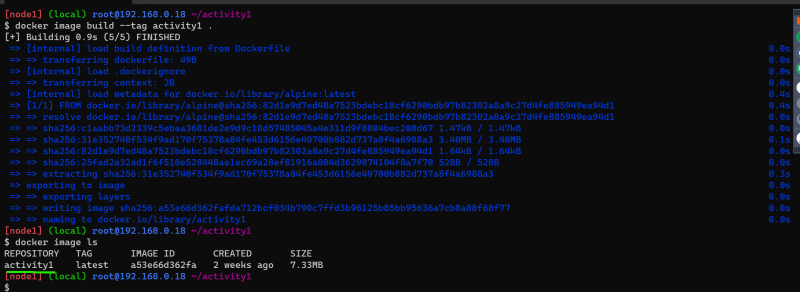
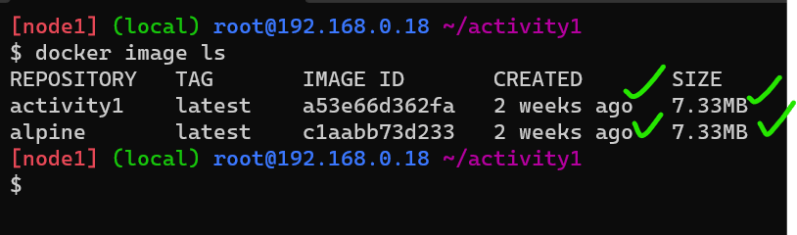
docker container run -d -P nginx ==> docker container run -d -P nginx:latest

* Dockerfile has set of instructions expressed in the form of <INSTRUCTION> <VALUE>
* Instructions of Dockerfile
  + FROM: This instruction sets the base image.
  + ARG
  + LABEL <https://docs.docker.com/engine/reference/builder/#label>
  + ENV
  + WORKDIR
  + RUN: This is most frequently used command, which does the activity of executing a command for some installation/configuration of your application.
  + COPY
  + ADD
  + USER
  + VOLUME
  + EXPOSE
  + CMD
  + ENTRYPOINT
  + HEALTHCHECK
* For all instructions <https://docs.docker.com/engine/reference/builder/>
* Docker image is collection of readonly image layers

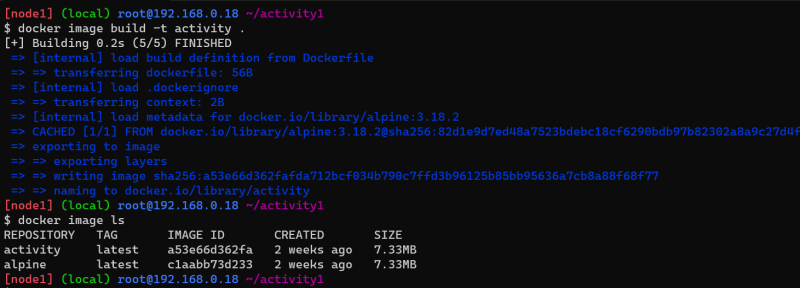
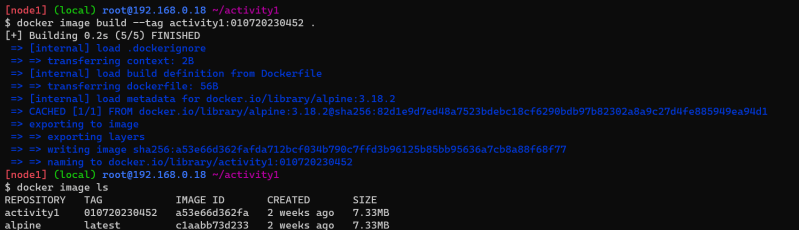
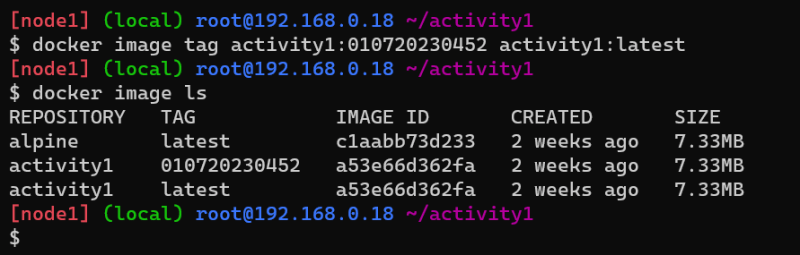
**Activity 1: Create a simple docker image with alpine as a base image**

Create a file Dockerfile in some directory  


The Dockerfile is as follows >FROM alpine

  
\* Now lets build a image called as activity1  
  
\* Now lets look at alpine image  
  
\* In this case we are seeing both images are of same size and showing as if they were create two weeks ago.  
\* Best Practice:  
\* Never use the Base image with latest or no tag  
\* Whenever we create image try to create it with some tag value and if require re-tag the image with latest for convinience  
\* Lets change the docker file to use the tag

FROM alpine:3.18.2

**Activity2: let’s install java 11**

* Manual steps based on alpine

apk update

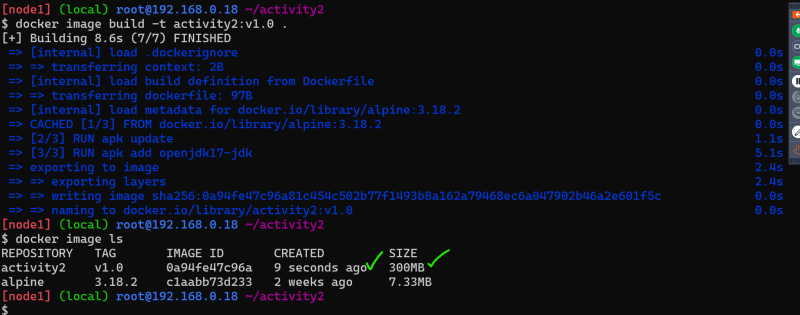
apk add openjdk-17-jdk

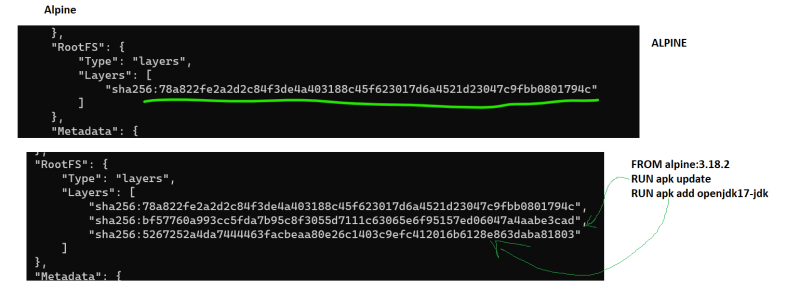
* Create a folder called as activity2
* Create a Dockerfile with the following content

FROM alpine:3.18.2

RUN apk update

RUN apk add openjdk17-jdk

Create a docker image with name activity2:v1.0  


This lead to the creation of 2 more layers to alpine image  


In the above case creating two layers is un-necessary

FROM alpine:3.18.2

RUN apk update && \

apk add openjdk17-jdk

* This leads to one extra layer
* Image layer id is sha256 hash of contents
* Docker image id is sha256 hash of all the layers and metadata
* Lets add some metadata to the docker image by using LABEL <https://docs.docker.com/engine/reference/builder/#label>

FROM alpine:3.18.2

LABEL author=shaikkhajaibrahim

RUN apk update && \

apk add openjdk17-jdk