**DevOps**

**Trainer:** Hiran Ram Babu or HRB

**Date:** 23-09-2019

**Synopsis:**

* GitBlade
* Jenkins
* Docker
* Ansible
* Nagios
* Sonar

**Day 1:**

**What is Devops?**

Culture

Together

Informed

Feedback – kick start from stage 1 if any failure

Pipelines – building the code

**Tools used in Devops**

Planning – JIRA

Code – IDE

Dashboard – Kanban

Store – Repository

Distributed – GIT

Centralized – SVN

Pipelines –Jenkins

Build – Maven, Ant, MS Build

Artifacts – Filesystem, Nexus, Azure Artifacts

Testing – Selenium

Releases - Ansible, Pupet, Saltstack, Chef

Deploy

Monitoring –Nagios

Being a devops engineer should know Pipeline and Release Deploy tools

While both **RPA** and **RBA** operate under similar concepts, the difference typically lies in how each is used as well as their complexity. ... Essentially, it can be said that **RPA**is automation for the end-user while **RBA** is more behind-the-scenes automation.

[Robotic Process Automation (RPA) vs. Run Book Automation](https://www.linkedin.com/pulse/robotic-process-automation-rpa-vs-run-book-vinil-g)

Traditional Method:

Hardware

A1, A2, A3

OS

H/W

Capex – Spent all money for building home

Opex – Taxes paying for water bill etc

Method 2: VM Workstation

1.Type 1 Hypervisor

App1 App2 App3

OS OS

Base Station

OS

H/W

Dev/Test

2.Type 2 Hypervisor - Virtualization

App

OS OS

ESXi

H/W

PROD

3. Containerzation

Server less Architecture : e.g Containers

FAAS – SG Lambdas

Architecture:

Vm – docker engine container

Container

Laptop

Hyper V / Oracle Vbox

Windows Pro

Linux:

Single Node

C1,C2,C3

C1,C2,C3

C1,C2,C3

Unix OS

H/W

Multi Node:

C1,C2,C3

C1,C2,C3

C1,C2,C3

VM

VM

VM

Hypervisor

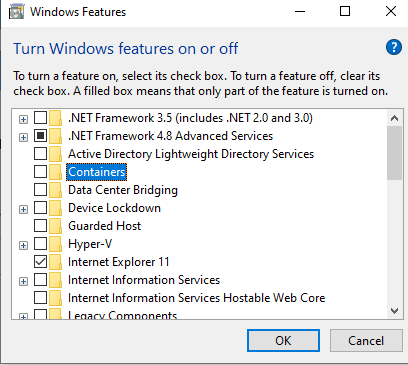
Unix OS

H/W

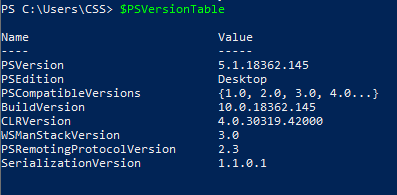
Kubernetes – Manage the container

Swarm –

**Installation Docker and Setup:**



Make sure Containers and Hyper V should be disabled for windows



If it is 3.x PS version then most of the commands will not work. Better to have 5.1

Check whether docker is installed or not



If not installed download it

<https://www.docker.com/products/docker-desktop>

Linux Docker Toolbox:

Docker Engine

GItBash

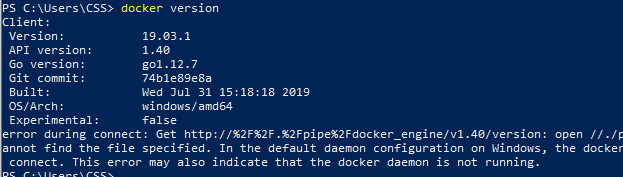
Virtual Box

Windows Docker Toolbox:

Git

Hyper V enabled





Code Repository for Docker:

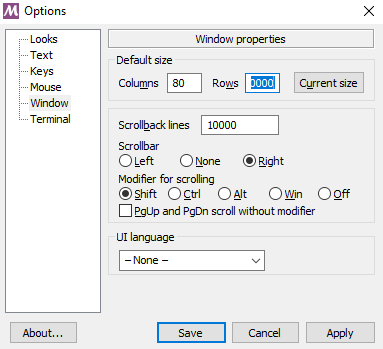
<https://hub.docker.com/signup>

Sign up and Login

Now use two terminal

1. Powershell
2. Git Bash

**Git Bash**



**Steps to follow:**

Login / Signup

Pull

Images / Image

Run – Detached, Interactive, Terminal, Port, Volume, Network

Attach

Commit

Tag / Rename

Push

Image to Container conversion should use Run command

Container to Image – Commit

Repo to Image – Pull

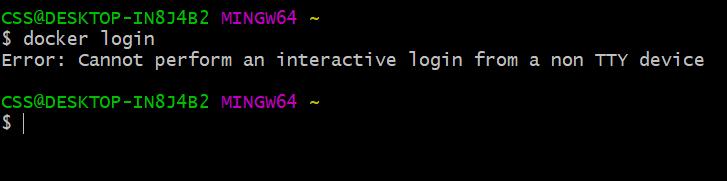
Image to Docker Repo – Push

Save convert to -> .tar

Load - >

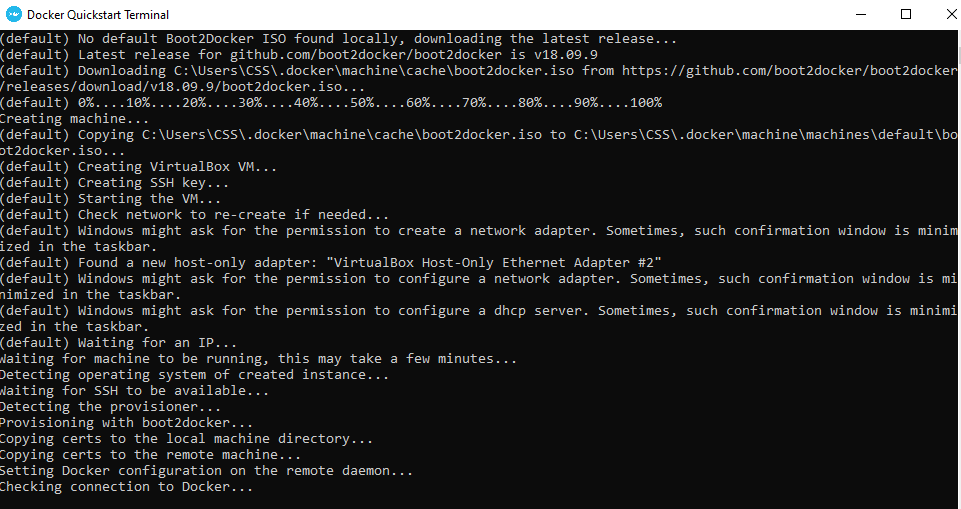
Docker Client – Type - Power Shell or Git Bash

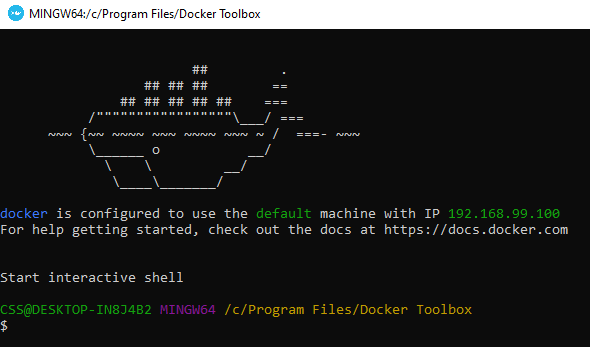
Docker Enine - Execution – Virtual Machine



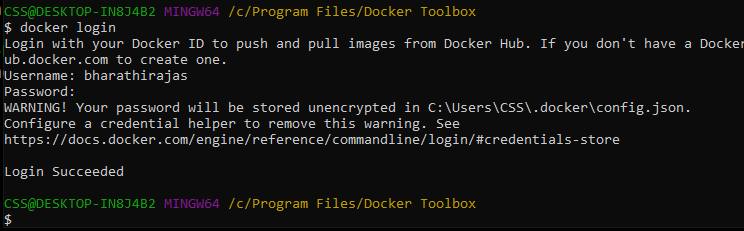
If any error come like above then use docker

Docker Quickstart Terminal

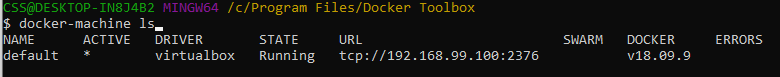




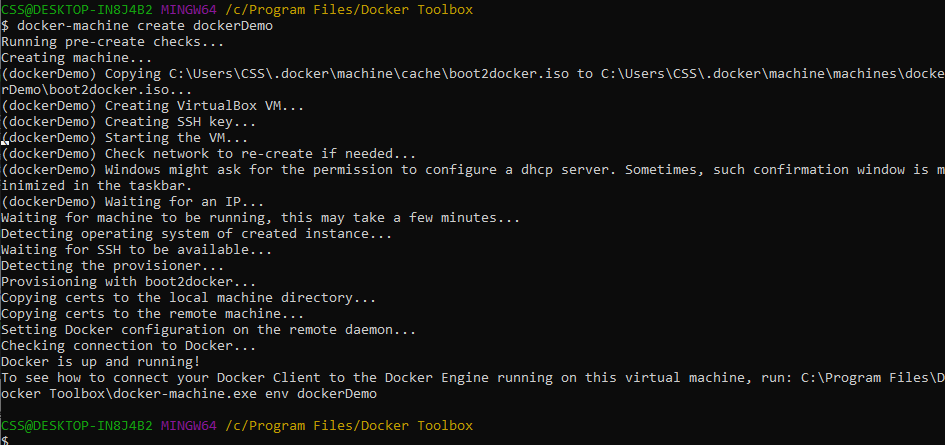
Ship image should display.



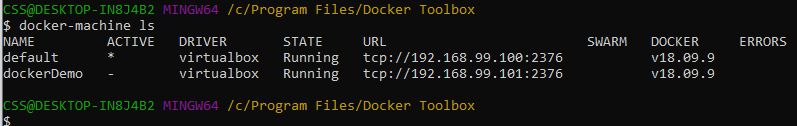
docker-machine ls – manage the VM, list the VM



Docker-machine create <name>



To verify the new docker



**Minimum space leave for Windows 10 machine:**

20GB Harddisk

1 CPU

2GB – RAM

Docker-machine create –help

**Default Installation Path of docker:**

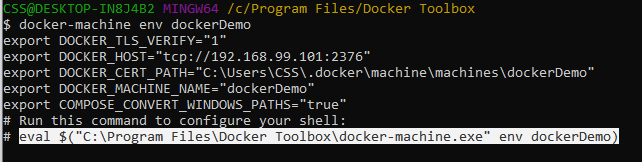
C:\Users\CSS\.docker\machine\machines\default\config.json

To see all configuration stuff either in config.json or below command

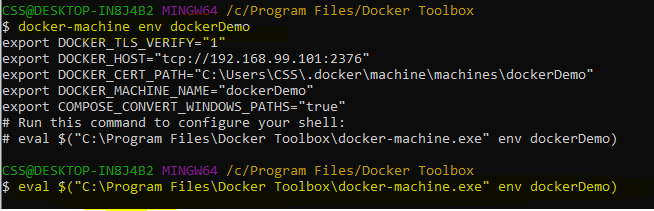
Docker-machine inspect <dockerName>

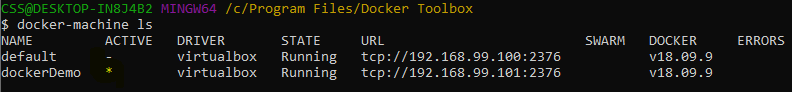
Make newly created docker as **active**:

Docker-machine env <dockerName>



Copy the highlighted one and paste it and enter





Provision is a command will recreate all container. Don’t use this in production.

Regenerate –search

**Stop Docker Container:**

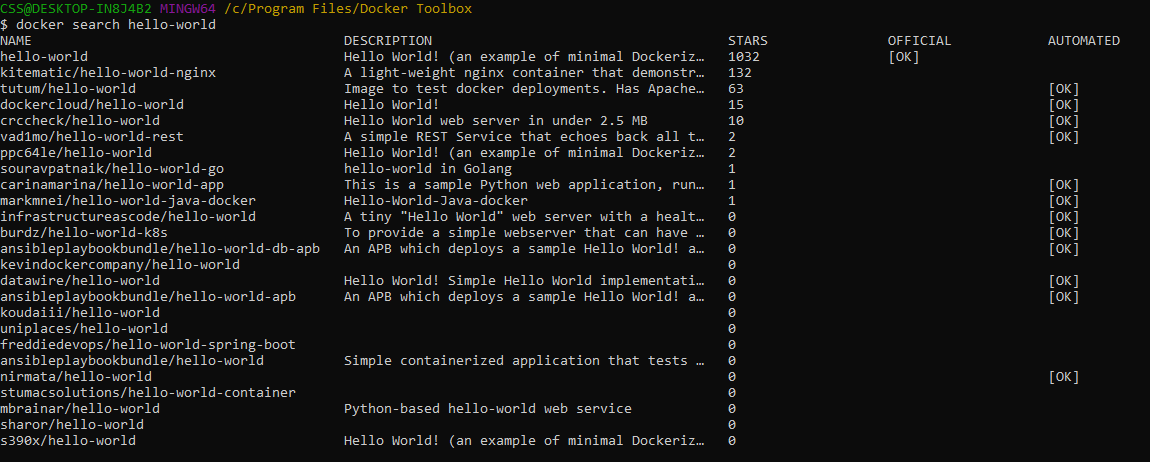
docker-machine stop <name1, name2>

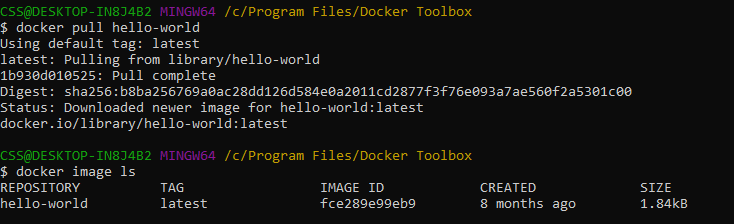
**Start Docker Container:**

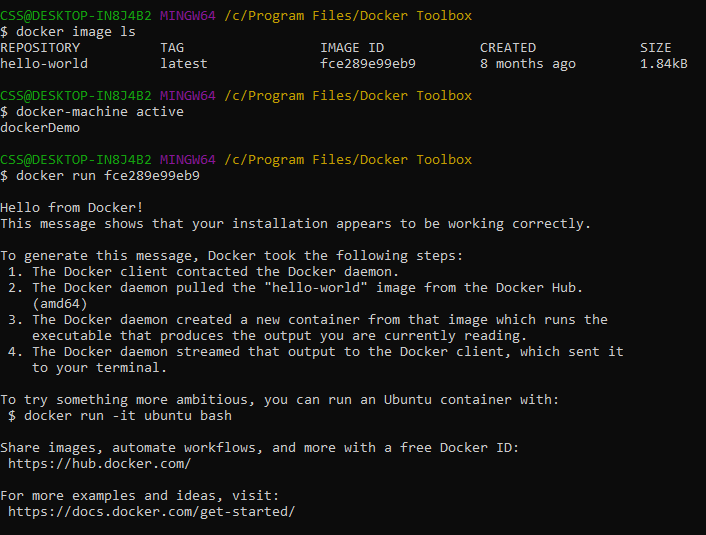
docker-machine start <name1, name2>

See list of docker image:









Docker container ls

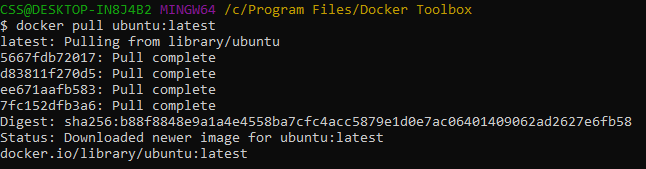
Docker ps

Will show only running container

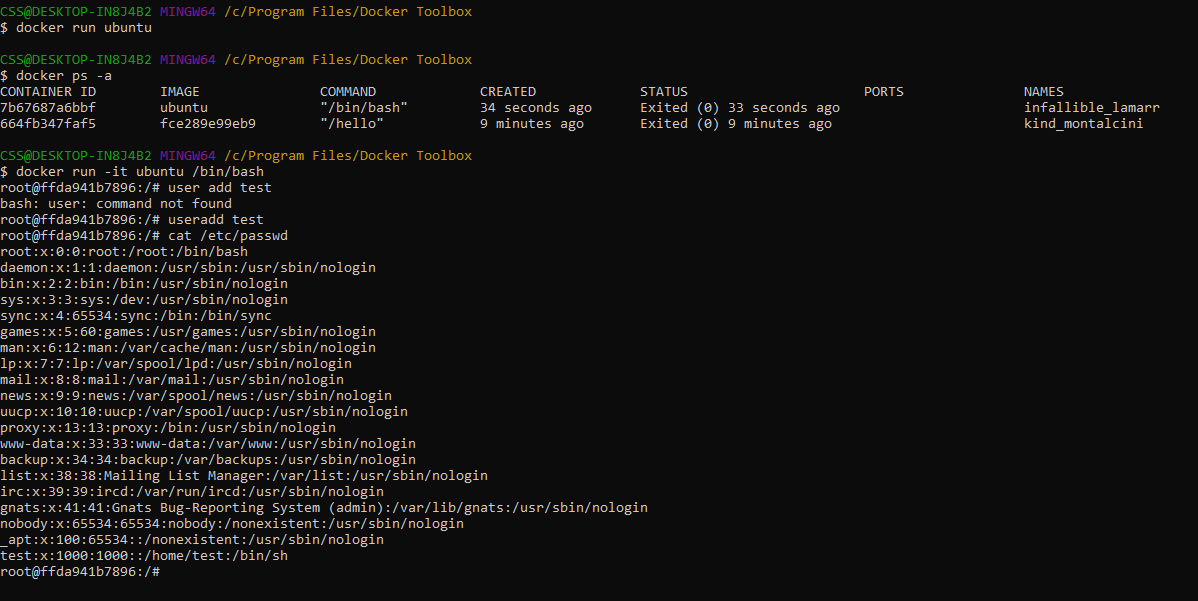
Docker ps –a

Will show all available container either active or inactive

Installing Ubuntu on docker







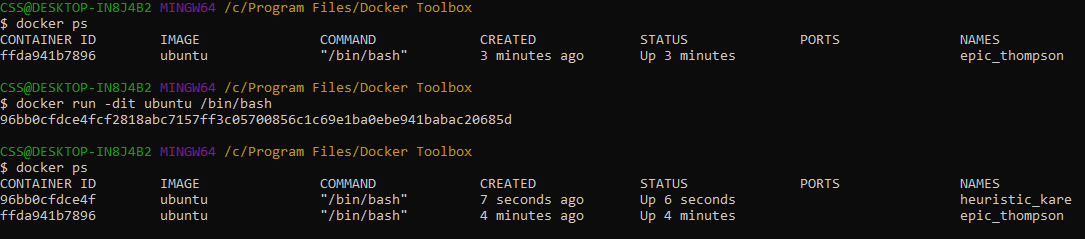
ctrl + p + q - control termination command

dont use exit - it will quit all containers

-I interactive

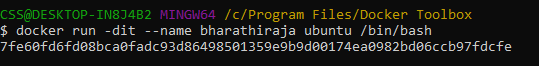
-d detached

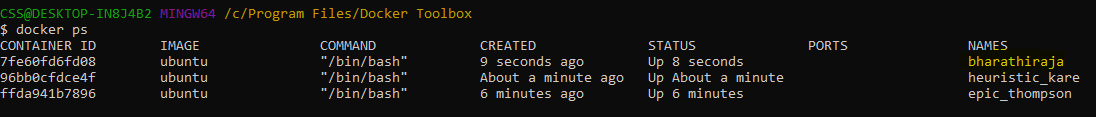
-t terminal

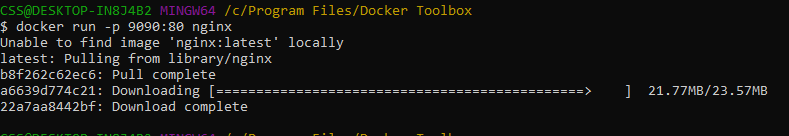


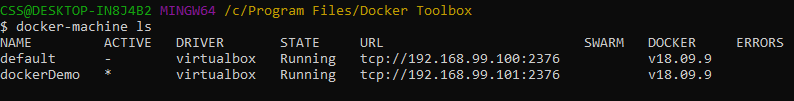
Docker <options> <commands>

docker run -dit --name bharathiraja ubuntu /bin/bash



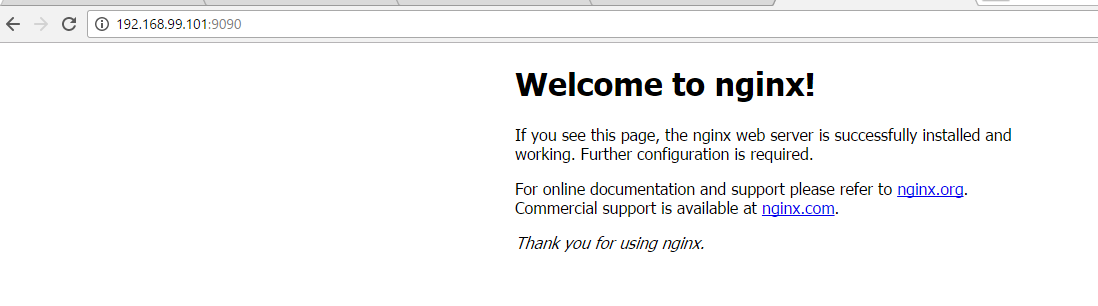






9090 - docker machine

80 – nginx



Refer - <https://hub.docker.com/_/nginx>

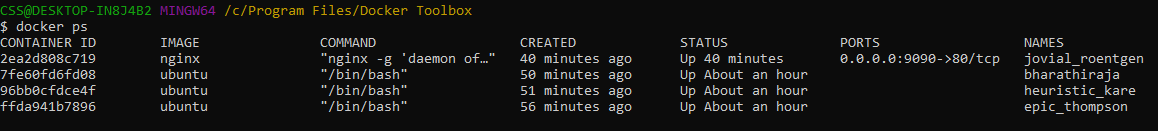
**To create new image:**

Docker compose using yaml file

Yaml – Python (indentation) + JSON

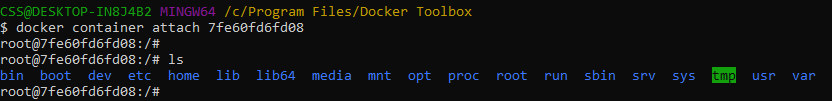
**Docker-compose.yml**

Save and Load is for loading tar file



Docker exec <id> hostname - Without connecting to docker need to know details

Want to get inside in to container use attach command

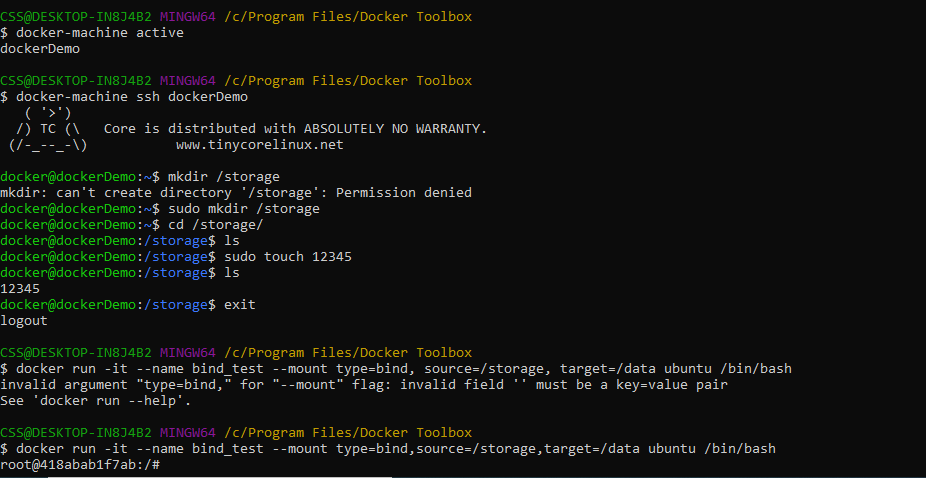


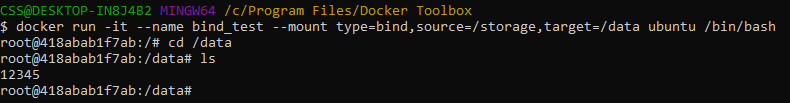
**Types of Mounts:**

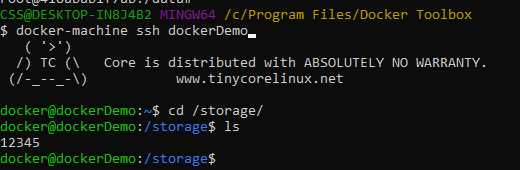
* Volume -
* Bind – connecting mobile with laptop and can see files in both devices
  + Source – VM / DM
  + Target – Container
* Temporary file system(tmpfs) – Only destination

By default docker swarm uses Bind

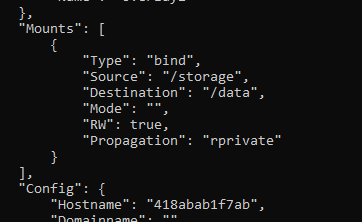
**Create space for the container**



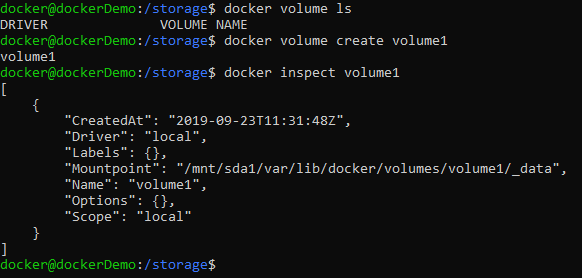
Ctrl P Q

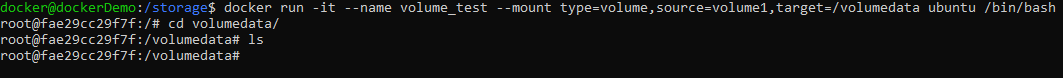




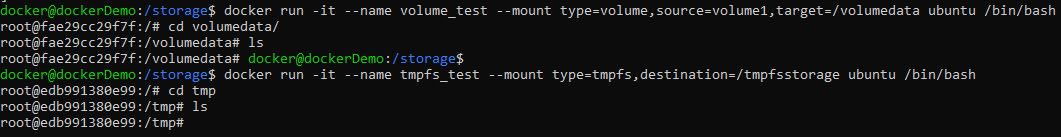


**Volume:**

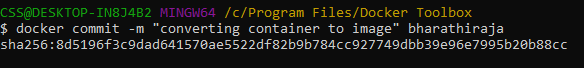


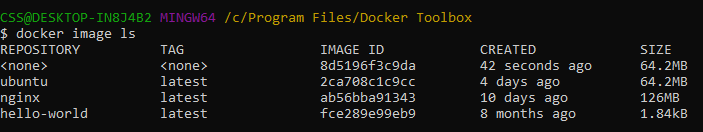


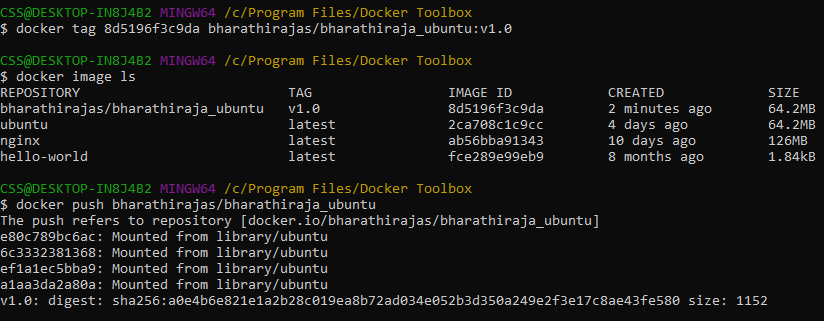
**Tmpfs:**











**Network:**

Every network will have DHCP

Types:

Bridge – Same n/w like the

Host only – only within the host

NAT (Network Address Trans)- IP to IP config

Physical NIC

Virtual NIC

Docker network connect <network name> <container name>

Day 2: Agenda

Network

Gitblade

Jenkins

Ansible

info@mjit.in