Cloud Computing

Grid Computing

* is a network of comp work together
* easy to process data than single machine
* all comp under the network work of same protocol
* Acts as virtual supercomputer
* Used where → analysing huge dataset
  → simultaneous requirement of high computing power
* Computers on network contribute res like → processing power
  → storage capacity

* Grid comp is a subset of distributed comp
* virtual super computer are connected by buses, Ethernet and sometime internet
* lly to parallel computing instead of multi core in single machine it contains multi core spread across

Working:

1) Control node:
   * like server
   * group of comp administere & keeps account
2) Provider: Comp that contribute res like storage, comp power etc.
3) User: Comps that use res on the network.

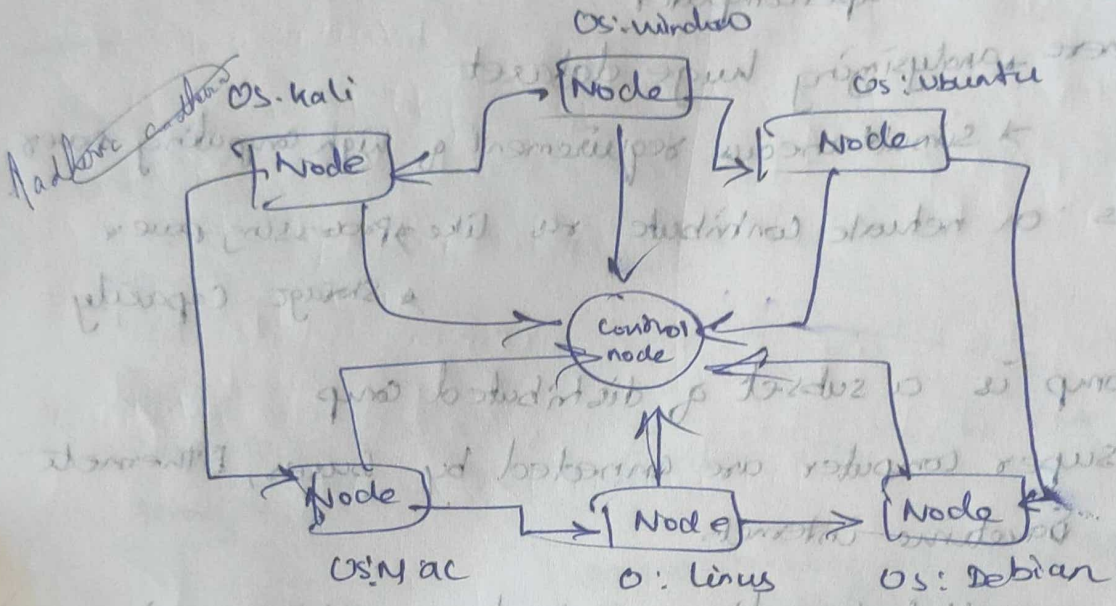
* User make request for res to the control node
* if res available it provides to user
* If the res is not used by user, it should send back res to node
* hence a comp in node can access like provide & user
* Node many contain comp of same platform (os) homogenous network
* " " " diff (os) heterogenous "
* This is diff bet. Grid & distributed comp

* To control network & its resources of software/network protocol generally known as Middleware

↓

responsible to control node & data

* Middleware work is to provide resources that is free & not to overload provider
* other job is authorization.



Topology of grid

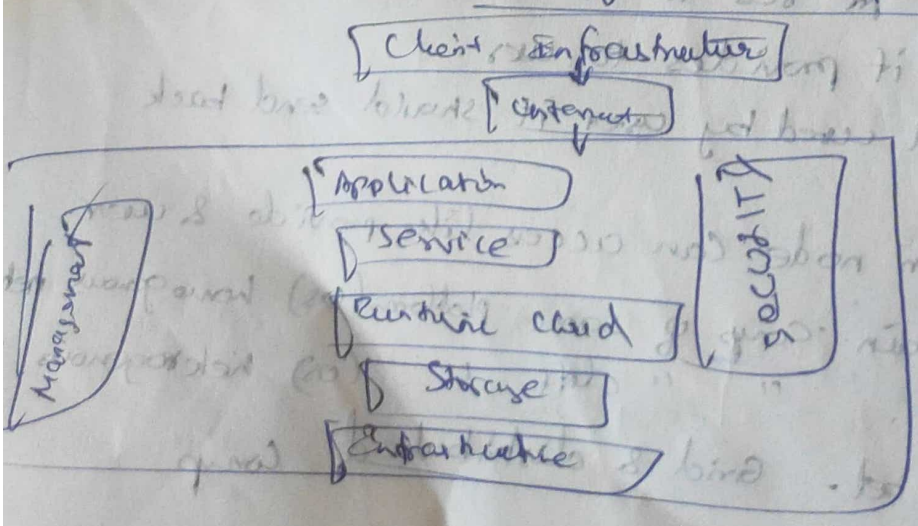
Advantages

* not centralized
* no server only control node - used for controlling process
* Allow heterogenous network.
* pay per use

Disadvantages

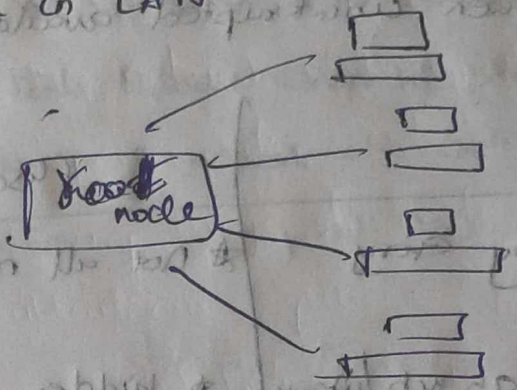
* in involution stage
* many license issue

Architecture of Cloud Comp



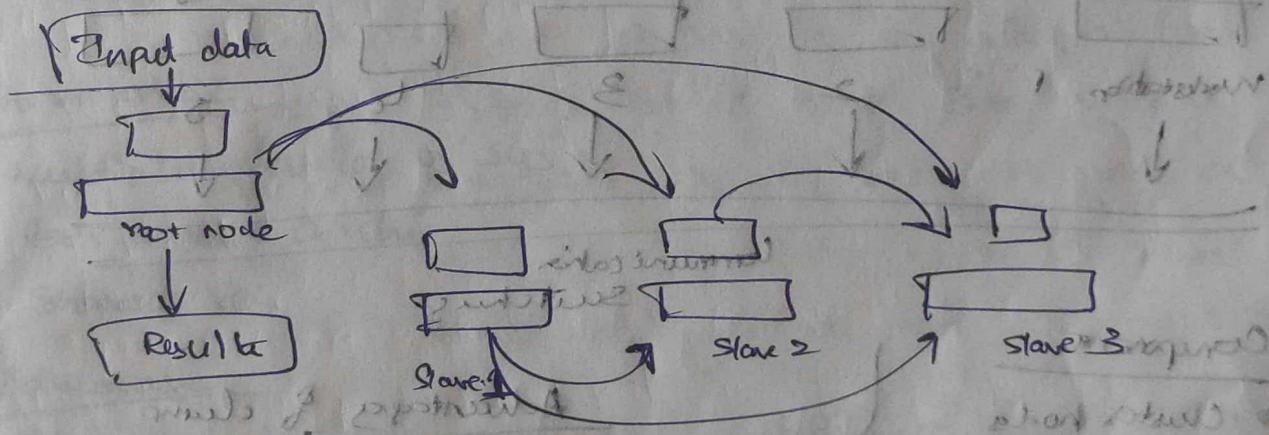
Cluster computing

* " is a collection of tightly or loosely connected comp that perform as single entity
* executes program as single machine
* Clusters connected in LAN



Advantage

* not expensive
* easy to process large data, fast & reliable
* many org & IT use cluster comp to increase availability, fast, scalability.
* Uses only single strategy to implement application.



Types

1) High performance (HP) cluster
   * HP clusters use comp clusters & super comp to solve complex prob
   * where nodes need to communicate to completing job
   * use Parallel comp power of all nodes.
2) Load balancing cluster
   * Distribute the request equal to all the slave comp nodes
   * This reduces time consume & shared acc
   * fast & don't all one node to be overloaded
   * used in web hosting environment

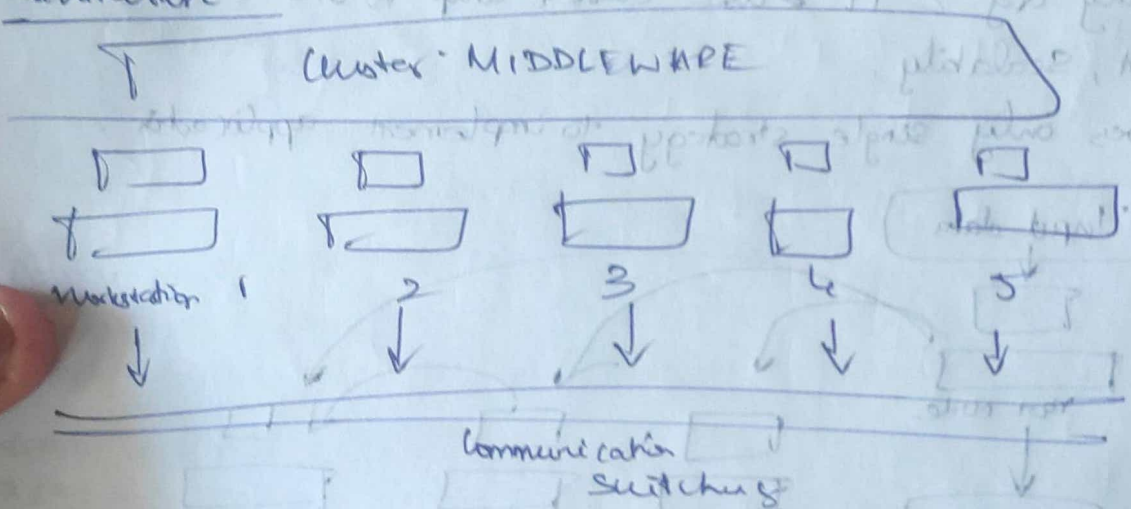
3) High availability cluster (HA)

* It maintains a back up nodes
* if any node fails it acts instead of that thereby providing resources continuity available.
* Designed to given uninterrupted availability of data.

Classification

Open cluster	Close cluster
* IP are needed by every node	* Not all nodes require IPs
* Accessed only through web/internet	* Hidden behind gateway nodes
* enhance protection	* Increased protection

Architecture



Components

* Cluster nodes
* cluster OS
* Switch or node interconnect
* network switching hardware

Advantages of clusters

1) High performance
2) Easy to manage
3) Scalable
4) Expandability
5) Availability
6) Flexible

Application

* used in weather forcasting
* aerodynamics
* ising systems
* earthquake simulation

Distributed computing

* No central servers / hubs
* Distributed comp refers to where processing & data is distributed across multi comp.
* each comp has it own processing power & storage
* ex: this is like cloud comp sys, where res like comp power, storage & network sys are delivered over internet on demand servie

Components

* Devices & sys
* Network
* Resource manage.

* Distribute comp is like peer-to peer sys where can con act as both client & server to directly communicate each other.

Characteristic

* Multiple devices or sys
* Peer to peer archi
* Shared res

Advantage

Scalability: * more scalable than centralized sys
* we can add or remove any no of comp to incre/decre comp power / strge

Reliability: even if 1 device fails it can perform op

Flexibility: easy to config & reconfig than centralized sys

Disadvantage

* complexity
* security
* performance

Application

* cloud comp
* peer to peer
* distributed architecture.

Utility Comp

* provides services & comp rcs to customer.
* pay per use
* provides on demand service
* allow org to allocate & segregate comp rcs to basis requirement
* less code, easy to manage, Reduce IT cost, flexible, reliable etc.
* focuses on acquiring comp rcs.
* 2 types
  internal & external utility
* Used by large org like amazon, google etc.
* thirupi adha eh cloudu

Cloud Comp

* Cloud computing is a use of rcs and storage over internet that is not owned by the user as pay per use & on demand service.
* Lower IT cost
* access rcs from nearby data centers.

NIST Model (National Institute of Standards & Technology)

* 5 essential characteristics
* 3 service models
* 4 deployment models.

Essential Characteristics

1) On Demand self service

User can access cloud rcs by himself without any help centers, IT dept etc. Can use till needed & can cancel if not needed.

2) Broad Network access

The service should be available any where so the user can access it from the desired location at any time.

3) Resource Pooling

The res over cloud is shared by many people so the personal & official data must be secured.

4) Rapid Elasticity

* it is flexibility of cloud that provides required res to the user depending on need

5) Measured service

Measure the service provided by cloud to user and charge them accordingly which is all transparent to user

Service Models

1) Software as a service (SaaS)

* provide software on demand service
* it is a software in which application are hosted by cloud service provider.
* user can access there software by internet web browsers.

Characteristics

* manage from central location.
* hosted on remote servers
* accessable over internet
* user not responsible software/hardware updates
* update automatically,
* pay per use
* ex: dropbox, google app, Cisco webex etc.

Iaas

* hardware as a service
* managed over internet
* This avoid the cost of purchasing costly hardware/servers for the user themselves.

Characteristics

* as available as services
* scalable
* Dynamic & flexible
* GUI & API based access
* eg: Aws, microsoft Azure

Paas

* Developed for programmers to develop, test, run & manage applications.

Characteristics

* accessible to various users via same apps.
* integrated with web service & DB
* build on virtualization.
* supp multi langs & frameworks.

eg: google app engine.

Deployment models

Public Cloud

* open to all
* all can access & store data using pay per use
* Comp res are manage & operated by Cloud service provider.
* eg: IBM, microsoft, google app engine
* Low cost
* Low security
* no maintenance
* flexible, location independ.
* scalable, can be accessed by public people.

Private cloud

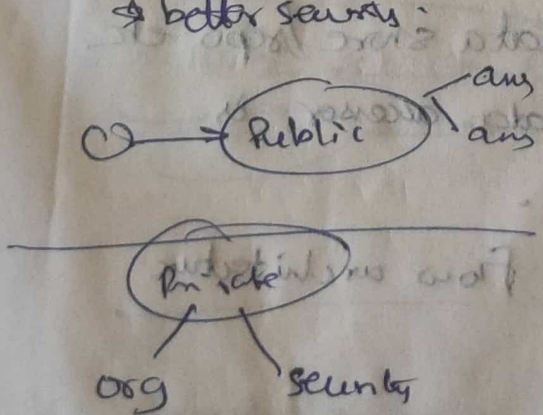
* also called internal cloud or corporate cloud.
* used by org
* 2 types
  - on premise private cloud (hosted within org, security)
  - out source private cloud (managed & hosted by third party)
* highly secure & fast

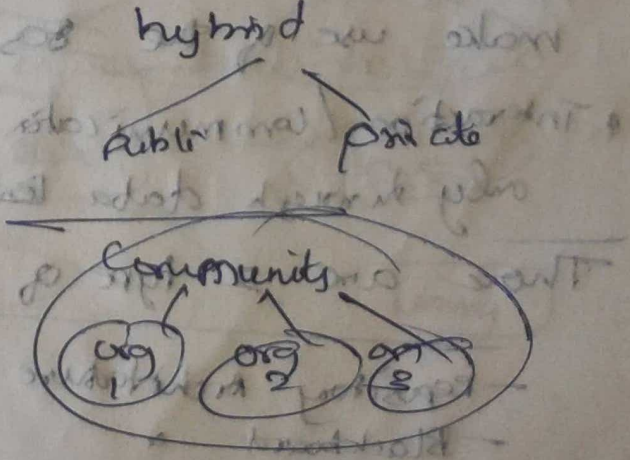
Hybrid cloud

* Public + Private = hybrid
* partially secure
* User can access service running in public cloud freely & can access services running on private cloud only if it is authorized.
* es: office 365, AWS
* Secure than public cloud, less secure than private cloud

Community cloud

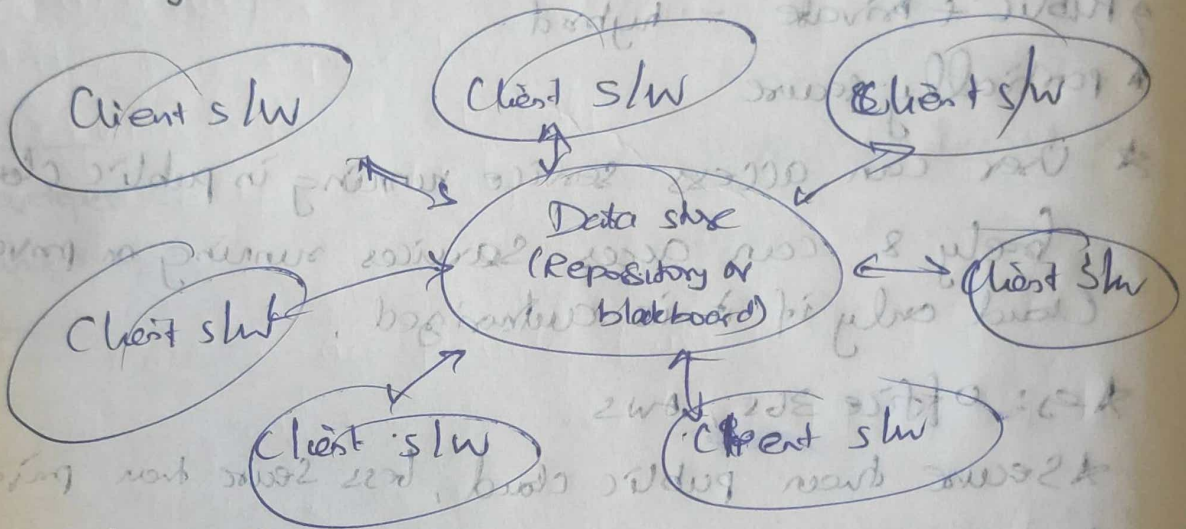
* allows services, data to be access by group of users or org, 3rd party etc
* es: health care community cloud
* cost efficient
* better security:





Data Centered architecture

* The data is centralized and accessed frequently by other component
* This communicate through shared repositories
* Shared data structures
* eg: is DB architecture in which common database schema is created with data definition protocol
* another eg: web architecture that has common data repository



Types of Components

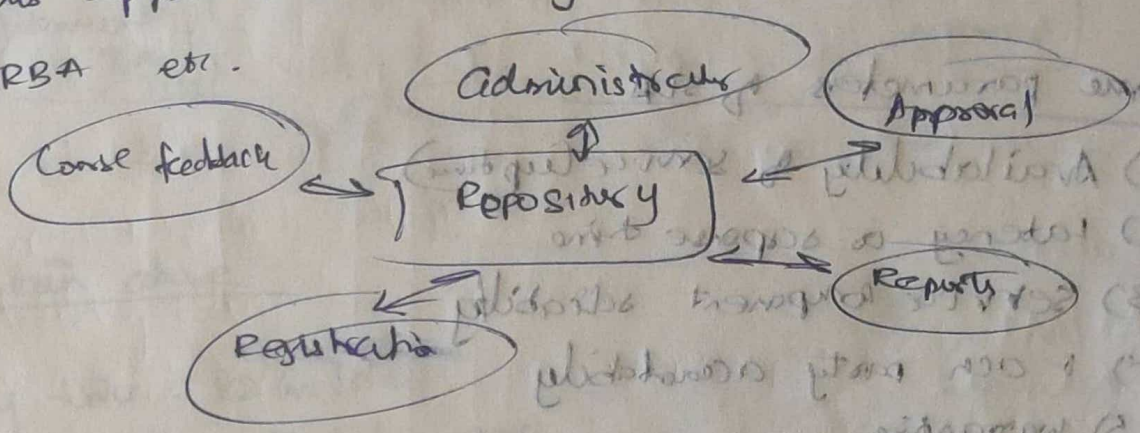
* **Central data**
  - This is a structure / data store / data repository permanently responsible to store data
* **Data accessor**
  - a collection on independent users that make use of the ds is data store / repo etc.
  - Interaction / communication bet data accessor is only through data store.

There are 2 type of Data flow architecture

- Repository Architecture style
- Blackboard "

Repository Architecture style

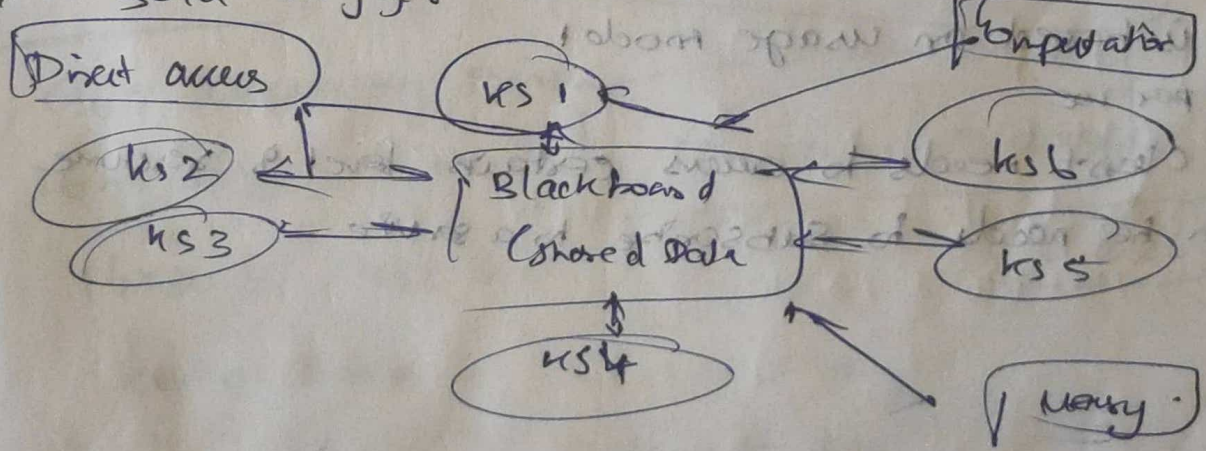
* Data store is passive, and client of datastore is active
* Client sends req that trigers the acts.
* This type of transaction is an input stream used in traditional DB
* this approach is widely used in DBMS, informatics, CORBA etc.



* Advantage
* provide data integrity, backup & restore
* provide scalability & integrity.

Blackboard Architecture style

* The datastore is active & Client is passive
* no. of components that acts independly is stored on blackboard.
* Components only interact through blackboard & and alert client whenever there is a datastore change.
* it send trigger when changes occur in data



KS - Knowledge source

SLA (Service level agreement)

* is a bond bet cloud service provider & client

Diff levels

* Customer based SLA
* Service " "
* Multi level "

Some parameters of SLA

1) Availability of service (up time)
2) Latency or response time
3) Service component reliability
4) Each party accountability
5) warranties

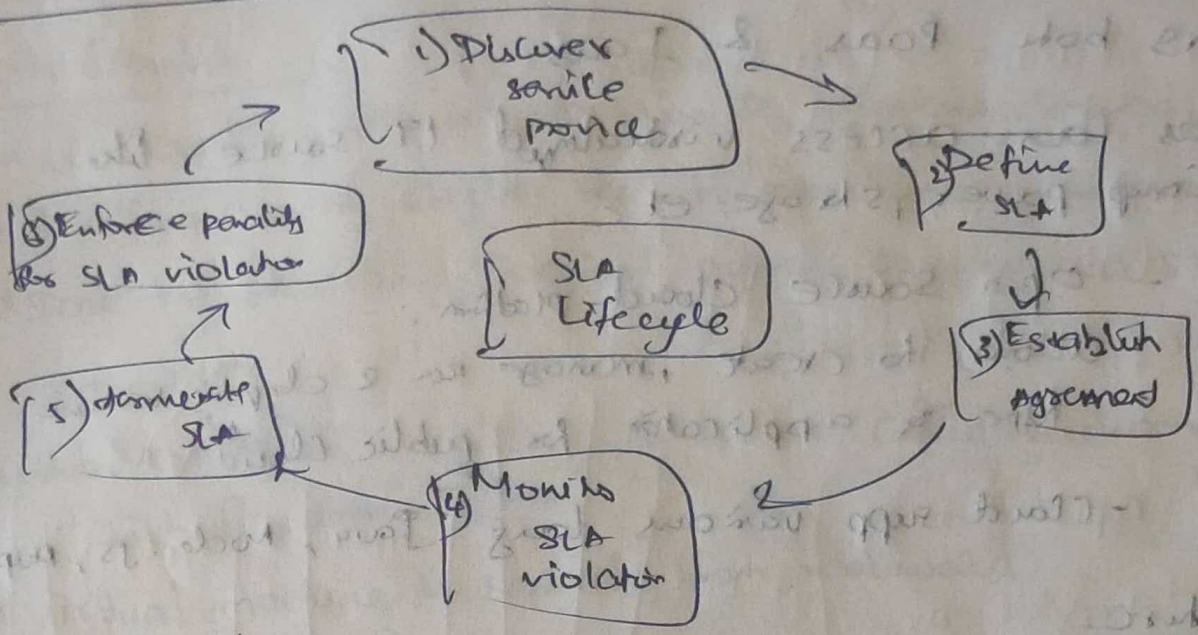
If the cloud service provide fails in any of the criterial of said res he needs to pay penalty to the consumer as per agreement.

2 Major SLA

* Windows Azure SLA → has separate SLA for Comp & storage
* SQL Azure SLA → has connectivity bet SQL Azure & client gateway

* SLA is based on usage model
* pay per use
* if client needs to access certain level of resource then he needs to subscribe to a service.

SLA Lifecycle



Explain above

1 day time = 12 hr 1 day

initial availability = 99%

End of month 10.75 hrs outage

Did service provider violate SLA

Sol

total app running in month = 12 * 30 = 360

Outage time = 10.75 hr

Service duration = 360 - 10.75 = 349.25 hr

% availability = $\left(1 - \frac{10.75}{349.25}\right) \times 100 = 96.92$

96.92 < 99 . CSP violated SLA

96% accuracy given

16 hr/day, 1 month, 12.30 hr - Outage

16 x 30 = 480 hr

Service duration = 480 - 12.30

= 467 - 30 hrs

$\% \text{ ava} = 1 - \frac{12.30}{467.30} = 1 - 0.026$

$= 0.97 \times 100 = 97\%$

0.026

It gives accuracy more than said

IBM Cloud

* offers both Paas & Iaas
* Iaas can access virtualized IT source like Comp power, storage etc
* Paas - open source cloud platform.
  allows to create, manage, run & deploy various type of applications for public cloud.
  - Cloud supp various langs Java, Node.JS, PHP,

Feature

1) AI/ML
2) Automation
3) Containers
4) IBM Cloud packs
5) Networking
6) Storage
7) security
8) database
9) Analytics
10) IoT
11) Blockchain

IBM Deployment models

1) Public
2) Dedicated - private, VPN
3) IBM Cloud private

Module - 2

virtualization

* it is creating virtual version of desktop, servers, storage.
* it allow sharing single res to multi org / customers.
* Done by providing logical name to physical storage & provider.
* when created on existing hardware called hardware virtualization
* virtual machine is separate from real world
* Main machine - host machine
* virtual - guest

Level of virtualization -

* Before virtualization there will be only one host OS sharing hardware res.
* but after virtualization there can be many guest OS that access same shared res.
* This is often done by adding additional software called virtualization layer.
* Virtualization layer also called hypervisor / VMM (virtual machine monitor)

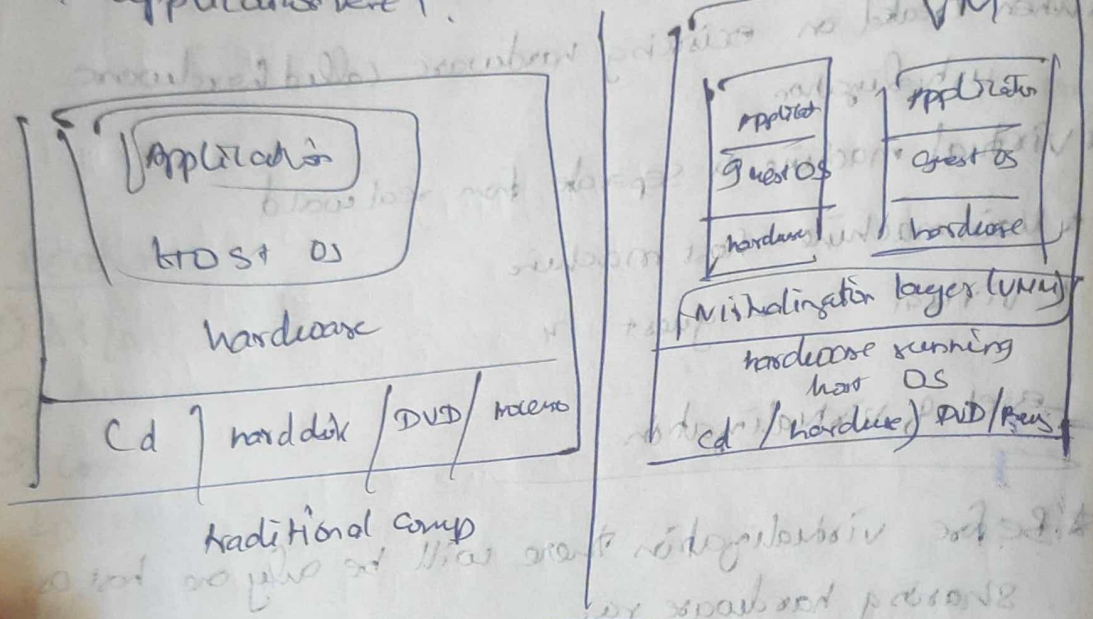
Software layer

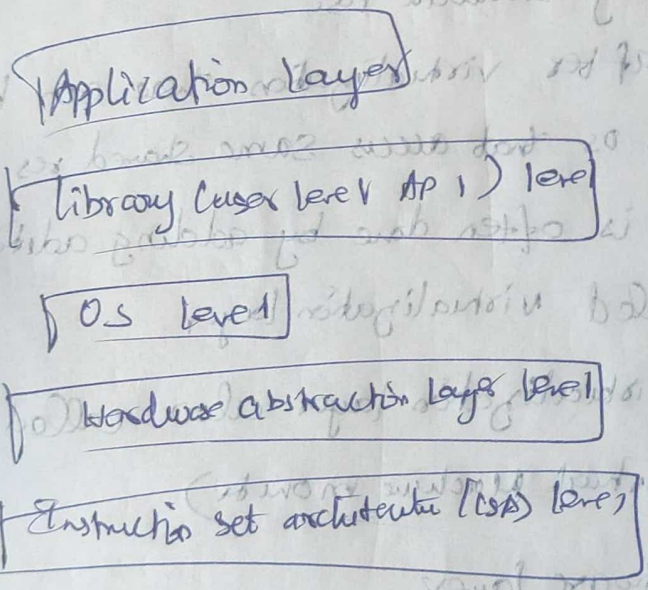
The main fn is to virtualize the physical hardware of host machine into virtual res to be used by VMs.

* virtualization software create many abstraction of VM by imposing virtualization at various levels of comp sys.

Virtualization layers

* Instruction set architecture (ISA)
* hardware level
* OS level
* library / support level
* application level.





Instruction set archi level

* It uses emulator
* emulating a given ISA by the ISA of host machine.
* It is possible to run various binary code on host machine
* emulator works basically through interpretation.

* This interrupts programe line by line to target instruction.
* One source instruction can require tens/hundred of native target machine.
* For better performance use dynamic binary translation that translate basic block of source code to target code

2) Hardware abstraction level

* This layer is above hardware
* This helps to operate virtual hardware layers for VM
* This virtualize components like storage, processor, I/O devices
* This helps upgrade hardware utilization to multi users

3) operating sys level

* Refers to obstraction layer bet traditional os & user applications
* This creates containers & we can make use of instance to access the hardware res
* Mostly used in virtual hosting environment

4) Library level (Support)

* Most user use API instead of using system calls by os
* Software tool WINE has implemented this approach to support windows Application on top of UNIX hosts.

5) User Application level

* This virtualize application as VM
* in traditional os application run as process
* also known as process level virtual machine
* popular approach is deploy High level language (HLL)
* JVM is good eg: of VM. any program written in HLL and compiled for VM run on it.

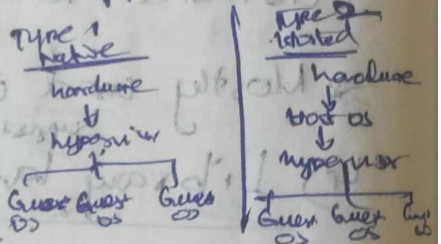
Virtualization structure tools & mechanisms

* It is a key technology used to improve reliability flexibility, scalability
* It allows multiple VMs to act on single physical machine thereby reducing hardware cost & improve its utilization.

Several tools

1) Hypervisors:

* Also known as VMM (Virtual Machine Manager)
* is a software layer that allows many VMs to act on a single machine.
* Hypervisor create virtual envi where each VM operates as if it runs on its own phys machine.
* It sits betn physical hardware & its OS
* provides hypercalls for its guest OS & applications.
* Depending on need it acts as



microkernel archi

- has only basic & unchangeable fns
- the device drivers are outside of it
- smaller than monolithic

monolithic hypervisor archi

- has all fns & device drivers within it
- Bigger than micro kernel

2) Virtual Machine Image

* It is a predefined template that contains an OS, software applications & other necessary components to run specific workload.

* Cloud providers typically offer some templates to create VMs easily.

* VMI is a virtual envir associated with VMs, harddisk. Ex.

* is a Comp Engine resource that stores all configuration, metadata, permissions, data from multi device etc...

* Diff type
  - Virtual Disk Image
  - Virtual Hard Disk
  - Virtual Machine Disk

3) Containers

* are light weight virtualization tech that allows multi isolated instance of application to run on a single host machine

* It does not need full OS it requires only a part of OS to perform certain application.

** Eg: youtube, Gmail. everything run as container in google

4) virtual network

* provides logical abstraction of physical network infrastructure allows multi VMs to communicate with each other & with outside world.

* Virtual network can be configured & managed using software defined network (SDN) tools.

5) virtual storage

* " allows multi VMs to share single storage or solid state drives.

* is typically implemented using SDN tool

6) orchestration tools : automate deployment, configuration & management of virtualized infra structure. administer diff templates or blue print to diff VMs & DCs.

Cloud computing with AWS

* AWS is a cloud service from Amazon
  * It provides service over internet
  * used to create & deploy any type of application on cloud.
  * pay per use

AWS EC2

* provides security & scalability capacity in cloud.
* used to launch many virtual servers as u need.

AWS S3

* provides storage service
  * u can store & retrieve any amt of data on internet

AWS is a secure cloud service platform, providing
Comp power, DB storage, content delivery

* Security
* Compute
* Storage
* DB
* Migrate
* network & content delivery
* Management tools
* Messaging

Virtualization of CPU, Memory, I/O devices

* Modern OS & processor allow multi process to run simultaneously
* if there is no protection mechanism in processor, OS then all process will access devices at same time leading to sys crash.
* So every processor should have 2 modes
  * user mode (unprivileged instr)
  * supervisor mode (privileged instruc)

CPU virtualization

* Enable efficient sharing of comp res.
* CPU virtualization allows many VMs to run on single phys machine by creating envi where each VM behave as if they are running on their own CPU.
* 2 approach of CPU virtualization

1) Full virtualization:-

* hypervisor create a VMs of holding seperate CPU, memory, storage from the given hardware.
* so each VM is allocated res as they are running on their own CPU
* less efficient than paravirtualization.
* Fully secure & flexible.
* no need to modify OS but bys trans can slow down performance

2) Paravirtualization

* In this the hypervisor provide virtual interface to underlying hardware by providing each VMs the same CPU.
* Require modification in guest OS
* Guest OS are modified to communicate directly to hypervisor
* better performance & efficiency.

Memory virtualization

* Enable efficient res utilization & sharing.
* memory virtualization allows many VMs to share same phy memory as if they use their own memory.

Reveal tech

1) Memory ballooning:

This is used to make release of the memory holded by idle VMs & provide it to the other VMs.

2) Memory overcommitment:

* It allocate more virtual memory to VM than available in phy sys.
* Done by use of page sharing & memory compression
* as not all VMs use memory at same time.

3) Memory paging:

* This allows Guest OS to move memory pages bet physical memory & disk storage
* Why phy memory full the unused memory of VM is moved to disk storage so VM can make use of memory more than available in phy machine.

4) Memory swapping

* Same as memory paging moves memory pages bet phys. memory & disk storage.

I/O virtualization

* Enable sharing of I/O devices like adaptors, ports, other peripherals.
* allows multi VMs to make use of same phy devices as an illusion they are its own.

several techs

1) virtual I/O devices :

* hypervisor present virtual I/O devices to each VMs, which operates as if it is phy device
* hypervisor makes phy device to be multiplexes as VM device

2) Direct device assignment

* in this hypervisor assign phy device directly to VMs by passing virtualization layer.
* require additional hardware support

3) I/O virtualization over ethernet :

* Enable multi VMs to share single network adaptor

4) Storage virtualization

* provide virtual storage to each VMs
* can take snapshots, data migration

Virtual clusters

* enable efficient OS management & scheduling workloads.
* virtual clusters are abstraction of phy clusters that makes use of shared res.

PTO

Tools

1) Container Orchestration clusters

* Containers like kubernetes, Docker swarm etc makes virtual clusters by abstraction phy res & create common platform to access containerized applica

2) VNM:

allocate of virtual cluster by abstraction phy res & can be create or deploy VM.

3) Resource Management framework

* allow creation of virtual clusters by providing distributed comp platform

---

Resource Management

* Virtual clusters are typically managed by cluster schedulers.
* which is responsible for assigning res to particular workload depending on res availability, request & policies. optimization res
* Cluster schedulers has many algos & heuristics etc.

Virtualization of data centre automation

* enables efficient automation & management of shared resource.
* Data center virtualization involves abstraction of res like servers, storage, processing power so it can be accessed by diff VMs
* automation enables org to provision res at regular intervals :

tools

* orchestration & automation tools
* Infrastructure as code (IaC)
* Service Management framework

Module - 3

Cloud security Risks

0) Insider threat

1) Data loss

Cloud security risk

* are potential vulnerabilities that can affect integrity, confidentiality and availability of data stored in cloud.
* these may occur due to service providers, user & attacker.

→ also know data leakage
→ happen when user data can't be accessed, read, stored, or deleted by someone.
→ happen when corrupted, harddisk not working etc.

2) Hacked interfaces & API

* API is a way we communicate to cloud through internet.
* need protection from third party API

3) Data Breach

* is a process in which the user/org data is been accessed (altered & deleted / steal) by someone who is without authorization.
* full data will be deleted.

4) Vendor lock in

As the org provide service from one platform to other there may occur prob.

During transmission Vulnerabilities

5) DOS attack

Flood the server with more request making it buffer overflow with traffic makes sys collapse & not to respond to anything - deadlock.

6) Account hijackers

7) Mis configuration

8) Malware attack

* regular update
* access control
* regular testing
* monitor anomaly activity

Trust

Prevention
* Encrypt data
* strong password

* help user to select provide / cloud service dependes on policies / trust it offers.
* important factor in cloud security.
* as user must trust their data is safe in cloud.
* trust can be build by transparent policies, loss etc.
* policy should provide functionality of user & provider

OS Security

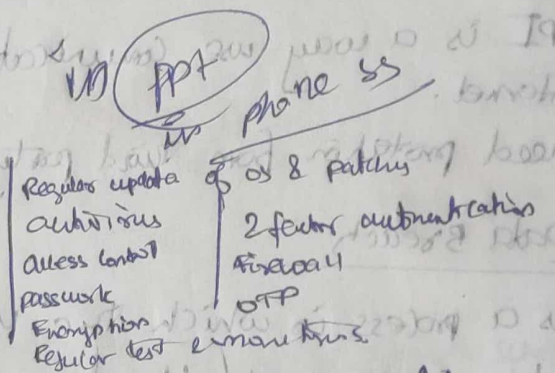
* ensure availability, reliability, integrity etc
* protect OS from trojan, malware, attacks, hackers etc
* Done by antivirus, firewall etc

2 vidation

1) Threats → pros has potential to harm sys
2) Attacks - breach of security that allows unauthorized access to res.

1) Data breach
2) unauthorized access
3) DOS

Goal of security



1) Integrity :
   user without access should not access file
   user with acces should not change/alter import files

2) Security
   only accessed by authorized users, not by all

3) Availability
   Accessable to all authorized user with any lack in res

Types of threats (Program threat)

1) Virus
   * Can replicate in sys
   * small piece of code
   * modify sys files

2) Trojan horse
   * Capture login credentials & send it to hackers / misfe

3) Logic Bomb

* misbehave only when cert cond met otherwise behave normal

4) trap door / back door

* a vulnerable pt in prog/sys that allow hackers to walk through.

Sys threats

1) Port scanners
2) Worm
3) DOS

Os threats

1) Malware
2) network prob
3) Buffer overflow

Security Prevention

1) authentication
2) One time password
3) 2 factor authentication
4) Fire wall

VM security

* Virtual machine can used to run sensitive data
* is isolated from underlying hardware
* vulnerable to threats, malware, attack etc.

Enhance security of VM

1) Secure Configurations

During configuration itself security should be checked → Strong password
→ authorization
→ removing unneeded services
→ enable encryption etc.

2) Patch Management

* It should update to the latest version
→ ensure they are protected against vulnerabilities

3) Access Control :

* only authorized users should user certain VMs
* decrease security & data loss

4) ~~Virtual~~ Machine Monitoring :

5) Virtual network security

* include firewall, VPN's, encryption

6) Data encryption

* encrypt stored data in VM
* protect it from unauthorized access / theft

7) VM Backup & Recovery

regular backup should be performed if there is need we can back up.

4 steps to ensure VM security

1) Protect hosted Elements by segregation
2) All components are tested and reviewed.
3) Seperate Management APIs to protect network
4) Keep connections secure & separate.

Virtualization security

* same as VM security

* benefits
  - Cost efficient
  - flexible
  - operational efficient
  - Regulars compliance

Types of security

* network security
* application "
* cloud "

Features

* segmentation
* micro segmentation
* isolation

Security risk of shared image

* Shared img refers to prebuild template of VMs
* They share OS, storage, comp power etc.
* security risks are

1) Vulnerabilities

→ shared img may contain vulnerabilities & security flaws
→ the os may contain vulnerabilities that affect VMs

2) Malware

* Malware includes trojans, virus, worms
* lead to data breach, account hijack etc.

3) Configuration error

* we need to config img properly or it lead to vulnerabilities
* Eg: not changing default password

allow unwanted access to VMs lead to vulnerabilities

4) Data leakage

5) Insider threat

Prevention

1) Verify img source:

org should use img only from trusted source & verify integrity before deployment.

2) Apply security Patches :

should perform regular updates to org core vuln

3) Use Access control

only authorized user should access specific VMs

4) VM monitor : Should monitor VMs for suspicious / unauthorized activities

5) Encrypt sensitive data

security risk posed by management OS

* Management operating sys (MOS) refers to software that runs servers that host VMs
* responsible for managing & orchestrating VMs
* provide interact bet user & Cloud infrastructure.

risks

1) Misconfiguration
2) Malware
3) Insider threat
4) Vulnerabilities
5) DDOS

Prevention

1) Regular update
2) Access control
3) Encryption
4) Network segmentation

Data privacy & security issues

Challenges

1) Data Replication
2) Data loss / leakage
3) Data breach
4) Insecure APIs
5) Internal threat
6) lack of control
6) read user agreement
7) Access limitation
8) confi sys update
9) Platform & control server moving
10) Encrypt data
11) audit & monitor data transfer

Prevention

1) keep local backup
2) Don't share sensitive data
3) Data encryption

Identity & access Management (IAM)

* ensures that only authorized users can make use of cloud res, process power, storage etc.
* IAM contains policies, processes & technology to manage user identities & control access to cloud res

1) Authentication

* is a process of verifying user identity or application
* may include passbord, finger print, biometrics, authentication factors to verify user's identity.

2) Authorization:

* is a process of granting access to cloud res based on user identity.
* Cloud providers use access control to manage authorization like role-based access control (RBAC)
  - Attribute "" (ABAC)

3) Identity Provisioning

* involves creating, deleting, modifying user accounts & access privileges in cloud sy.
* Provisioning can be automated through APIs or other provision tools

4) Federation:

* Allows users to access Cloud res by using credentials from external service providers.
* Can simply authenticate process for users & improve security.

5) Monitor & auditing.

* essential for detecting & preventing security breaches

Operation

1) Strong Password policies
2) RBAC or ABAC
3) Multi factor authentication
4) Regular audit
5) Encryption

Access control & authentication

↓ what ses & data should user access & given access to

↘ verifying the users credentials

1) Multi factor authentication
2) role based Access control (RBAC)
* assign roles to use depadonding to jobs
* allow access to cloud depending on role
3) Attribute based Access control (ABAC)
* Checks whether user have access to that attributes
* Dependent on attributes like time, place, location, etc.
user can access cloud data
4) Identity federation
5) Least privilege
Allow access to res that are only required by user remove all other unwanted access to res

Microsoft Azure

* cloud comp platform
* used to build, deploy & manage applications in cloud.

Components

1) Azure Compute:
   Contains all severs, serverless, containers, VM etc.
2) Azure storage:
   * used to store both structured & unstructured data
   table / not table
   * DB, sql, mongoDB etc.
3) Azure networks:
   * provide virtual network
   * load balancers.
   * reduce traffic
4) Azure DB
   Store both structured & unstructure DB
5) Azure security
   * encryption
   * protocol
   * access control

GCP (Google cloud platform)

* same
* extra it allows analyse data & perform ml models

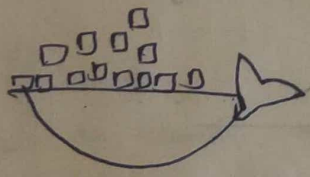
Same components

1) Compute
2) Storage
3) Networking
4) DB
5) ML
6) security

Mod- 4

Docker & container essentials

* Docker is a open source platform
* allows developers to package, deploy & run application with containers
* Containers are lightweight packages that contains everything to run a prog including code, library, sys tools etc.
* Docker provides a way to make use of PDK
  helps developer to build, test & deploy applications
* Docker typically creates a Dockerfile which is a text file that has instructions to create a docker img
* Once the docker img created in can run on any machine in which docker is installed.
* Docker is a tool used to build & deploy applications mainly used in Cloud Computing & DevOps.



Docker

Docker Containers

1) Images:
   * This is read only templates
   * That contains info to build Image
   * Then Images can be stored in registers like Dockerhub
   * Can be daonloaded by user as containers.

Containers:

* are instance of img running on host sys
* Containers are seperated from each other & from host os
* It contains all info to run application including code, library

Dockerfile

* is a txt file that contains info how to build a Docker img.
* specifies base img, basic dependancies & fn.

Docker Compose

* Is a tool that allows to run multi docker ~~applica~~ container application.
* It uses YAML file to define services & dependencies of your application.

Docker register

* is a storage and distribution sys for docker img
* Dockes hub is default public register
* U can also set private registers to share img.

Working with Docker:

1) volumes:

* is a way to persit data
* it helps to transfer data bet host machine & container
* or bet many containe
* volumes can be create use 'docker volume' command or by specifying in dockerfile.

2) Bind Mounts

* way to map host machine to container
* share data betwe host, machine & container

3) Docker file

4) Docker Compose

5) Data container

6) Backup & restore

-> Create backup of volumes & data containers using "docker export"

Docker network

* allows container to communicate with each other & with outside world

1) Bridge network

* default
* allows to communicate ~~container~~ containers with each other & host machi

2) Host network

* allow containers to make use of host network instead of its own

3) overlay network

* allows containers to comm multi docker host
* allow multi host service like Docker swarm, kubernetes

4) Network Driver

* allows diff type of network to be used
* eg: allows bridge, host network, overlay, network

5) Port Mapping

* Expose network port of container to host machine

6) DNS resolution

* allows DNS resolution for containers, which allows Containers to refer to each other by Container names.

Kubernetes

* is an open source orchestration platform used to automate containerized applications
* Originally developed by google now maintained by Cloud Native Computing Foundation (CNCF)
* use declarative approach - that means developer can define desired state of application in configuration file,
* kubernetes automatically manage & scale them the containers are in desired state.

Key components

1) Nodes:

* Each nodes are separate host where containers are deployed.
* Each nodes has its own cpu, memory, processor

2) Pods:

* are smallest unit in kubernetes
* can have one or more container in nodes that share same res.

3) Services

* It can provide way to expose pods
* they provide load balancing, service discovery & other network related services

4) Deployments

* way to manage rollouts & scaling applications, updates
* we can find no. of applications running
* and can rollback to prev version if there is issue in newer

5) ConfigMap & secrets

* provides a way to manage passwords, API key and other needed for containers.

6) Ingress:

* provides ext access to services running in clusters
* " load balancing on hostname & URL path.

* kubernetes provide powerfull API & CLI (command line interface) for managing clusters & applications.
* can run on diff platform data centered, public, private etc
* Scalable & flexible
* focus on building & deploying application than managing.

KinD (Kubernetes in Docker)

* is a lightweight tool for running

Steps

1) Install Docker
2) Install KinD and KinD Github repository
3) create new clusters in KinD by "kind create cluster"
4) verify cluster by "kubectl cluster-info"
5) Deploy application - once kubernetes cluster is up & run u can deploy it in cluster using kubernetes manifests
6) Manage the clusters - using kubectl command
7) Clean up - once u are done with kubernetes clusters u can del by "kind delete cluster".