

- what is polymorphism
- type of polymorphism
- static vs dynamic poly

- what is overloading
- type of overloading
- why overloading
- where you used overloading.

→ Can we overload static, main, return type, <sup>(Ans)</sup>  
Chores, access modifier changes, exception throw changes.  
final, abstract, any why.

- when changing of return types → block
- can we overload class, variables and constructors.
- @ overloading in type conversion / type casting
- what is type Casting and its types.
- overloading in same or different class.
- write a program for overloading.

- overriding is what.
- overriding with me. Same or different class.
- write a program for overriding.
- why overriding
- Real time example of overriding.
- what is covariant types in overriding.
- can final, static, private, abstract ~~be~~ be overriden.
- can abstract, public, protected, default, ~~be~~ be overriden.
- can & covariant types work on primitive data types
- How covariant type work on nonprimitive data types  
think by examples.
- checked and unchecked example ~~given~~ using three conditions
- can we override a variable (static, non static, local)
- can we override a constructor
- can we override a class
- can we override block

Final To stop overriding.

→ final Can be class or why

→ final variable and why

→ what is final variable and why it created.

→ Three ways to initialize final variable.

→ Can we reassigned final variable

→ Can we create a local variable as final.

→ what is block final final.

→ How to initialize block final variable

→ Real time example of block final variable ex-PAN

→ Can a static variable be final. will find.

→ what is final method and why

→ final Can be private, public, protected, default,  
static, abstract,

Final method always bind at Compile time  
so its more performance like static method

→ final Can be block and constructor.

→ what is final class and why we create it

→ What is Static Variable

→ Static with local and global variable.

→ When Static variable used in JVM.

→ I'd tell, Create of Static variable.

→ What is Static pool.

→ Static variable is class variable or object

→ Can we Create Static variable inside

Static method, Non Static method, Constructor  
block,

→ How to Access Static variable

→ Real time example Ex- College Name.

→ Why used Static variable

→ Can we access Static variable in Static and  
Non Static method,

→ Can we access Non Static in Static  
and Non Static method

→ What is Static method.

→ We need Object to Call Static method?

→ Can Static method access ~~Static~~ Static variable  
and ~~non~~ Non Static method.

→ Can Static method access ~~Static~~ Non-Static  
method / variables. Direct or ~~Object~~

→ Can we Create class object in Static method.

→ Can we used this with Static. X

→ Can we used Super with Static X

→ Class can be Static ↘

→ Block can be Static ↘

→ Constructor can be Static X

→

this and super

DATE \_\_\_\_\_  
PAGE \_\_\_\_\_

- this what is this.
- this can be class, block, method, constructor
- this can be used with global variable / access / local variable access / static variable.
- this can be used constructor call ✓
- this can be first to last or mid line ✗
- this can be used with method call ✓
- this can be block call ✗
- How constructor chaining by this:

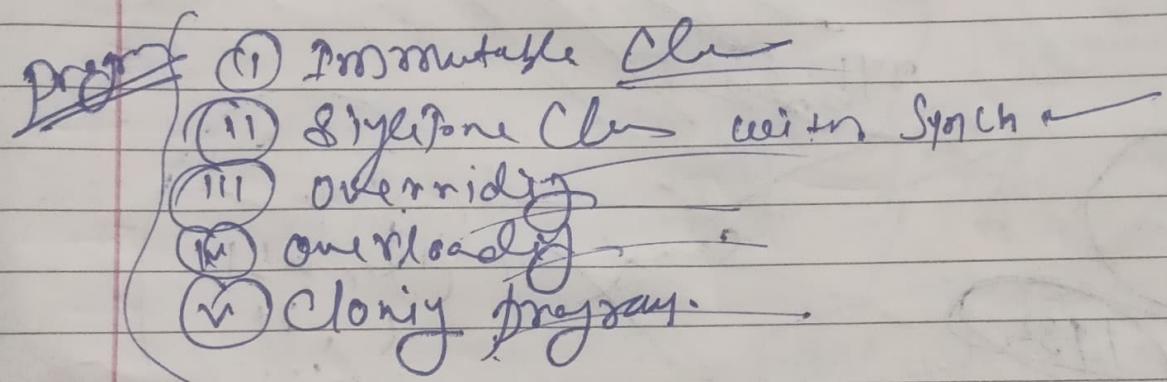
- Constructor overloading by three types
- what is constructor and when it called.
- what is default constructor when it created class.
- Can default constructor call when we write a constructor
- Constructor Chaining by two ways (super/this)
- why constructor chaining.
- Real time example

- What is inheritance
- what is DSA
- Why inheritance.
- Real time example of inheritance
- Type of inheritance.
- Can we extend interface inside interface.
- How to extends multiple interface.
- Why doesn't support true multiple inheritance
- what is (Graph) Composition
- what is Aggregation
- what is the diff. b/w Composition, association, aggregation and inheritance
- what is aggregation.
- what is Diamond problem
- what is ambiguity
- How to resolve multiple inheritance by two ways (Java 8).

### Interface

- What is abstraction with real time.
- ~~What~~ How to achieve abstraction
- what is abstract Class / Interface.
- Can we create object of abstract class / Interface
- Real time example of abstract class
- When to use abstract Class and when to use interface - Real time examples.
- what is difference b/w abstract class and interface.
- Can we create Constructor / method / variable / block ~~field~~ in Interface and abstract class.
- What is Toward Changes in interface.
- Can we create main methods in interface.

- Can we use both abstract class and interface simultaneously together?
- Show its encapsulation and Real time example
  - Why encapsulation
- How to create immutable class with program
- What is wrapper class and why



- 8 methods of object class
- deepClone vs shallowClone
- Equal vs hashCode()
- Serializable vs Deserializable
- final finally and finn

- Types of design patterns -
- Singleton / Factory / Builder /

## Tana-8

Even number

②  
divide  
at 20  
at 20

10 → visit → end = visit

④  
CPY  
interface abc  
DATE  
and PAGE

- Normal, parametric function found a creation utility
- what is functional interface.
- find odd/even number by predicates (true/false)
- multiply even by two and return result (function)
- print your name by consumer.
- generate random number by supplier.

~~Simple API~~

- How to create anonymous function.
- ~~what is~~ lambda as anonymous function.
- What is functional interface and when we use it.
- Create it. How to create function interface.
- Give a situation where we will create function interface.

- What is predicates, Consumer, Function, Supplier
- What is BiPredicate, BiConsumer, BiFunction, BiSupplier
- How to pass three values in predicate
- What is optional and its usages and how to use some functions - and why it useful
- What is method referencing and why it useful
- What is date and time Chaged in Tana-8
- interface Tana-8 Changes (with default, static, private)
- try with yours -
- Stream Joiner
- Tana-8 Comparator and Comparable programs.
- Stream API functionality -

- ① Partial injection.
- ② If container, default setter injection
- ③ Not new & ~~will~~ PAGE 6 every time.

- Spring and Spring Boot version
- What is PoC Container.
- Type of dependency injection (Constructor, Setter)
- Consumer VS Setter injection
- ④ Primary and ⑤ Qualifier
- Why of dependency injection (Mannual and autowired)
- Can Primitive and String be autowired.
- ~~surgeon~~ Mode of Autowiring (by name / ByType / No / -)
- default mode of autowired (ByType and if not found then ByName)
- ⑥ Required (if field mandatory for object creation)
- How to get data from proper files at class level
- How to get data from property files' level.
- Type of Application Context via bean factory
- ⑦ Bean life cycle
  - ⑧ def
- ⑨ PrePostConstruct and ⑩ PostDestroy
  - ⑪ Initialization
- BeanScope ⑫ Singleton ⑬ Prototype
  - ⑭ Request ⑮ Session ⑯ Application
- What is default method of BeanScope.
- ⑰ LookUp annotation
- What is dispenser Jewell (front controller)
  - (pre / Post processing)
- Handler method (~~process~~ <sup>process</sup> PostController method)
- Key Component of Spring boot (Spring boot Starter Class, Configuration, Actuator)

→ difference b/w @Primary Vs @Qualifier

→ what is dependency injectors.

→ What is autoConfiguration

→ What is @SpringBootApplication and all three  
Context

→ What is Spring bean and Component Scanning

→ What will happen if two base package choosed.

→ Application Runner vs Command Runner - interface

→ How to disable autoConfiguration

→ @RestController = @Controller + @ResponseBody

→ @RequestMapping - what is? (Versioning, default)

→ Maven goal ~~and add~~ - Path uniqueness?

→ How to create jar QA / Build in Spring boot

→ Jar vs WAR difference in Spring boot

→ Maven goal ~~test~~, test

→ profiles (property file (UAT, Prod, Preprod), Prod)

→ How to execute specific code on UAT:

→ what is actuator and list three end point of actuator

→ Spring boot VS Spring MVC

→ @Component, @Service, @Repository, @

→ @Bean (manuf object creation)

→ Spring-boot-Starter-parent ~~define~~ and

Spring-boot-Starter-web .n

→ Query VS path VS Body and its annotation.

→ How or what is @Query

→ How to create API Query param non mandatory.

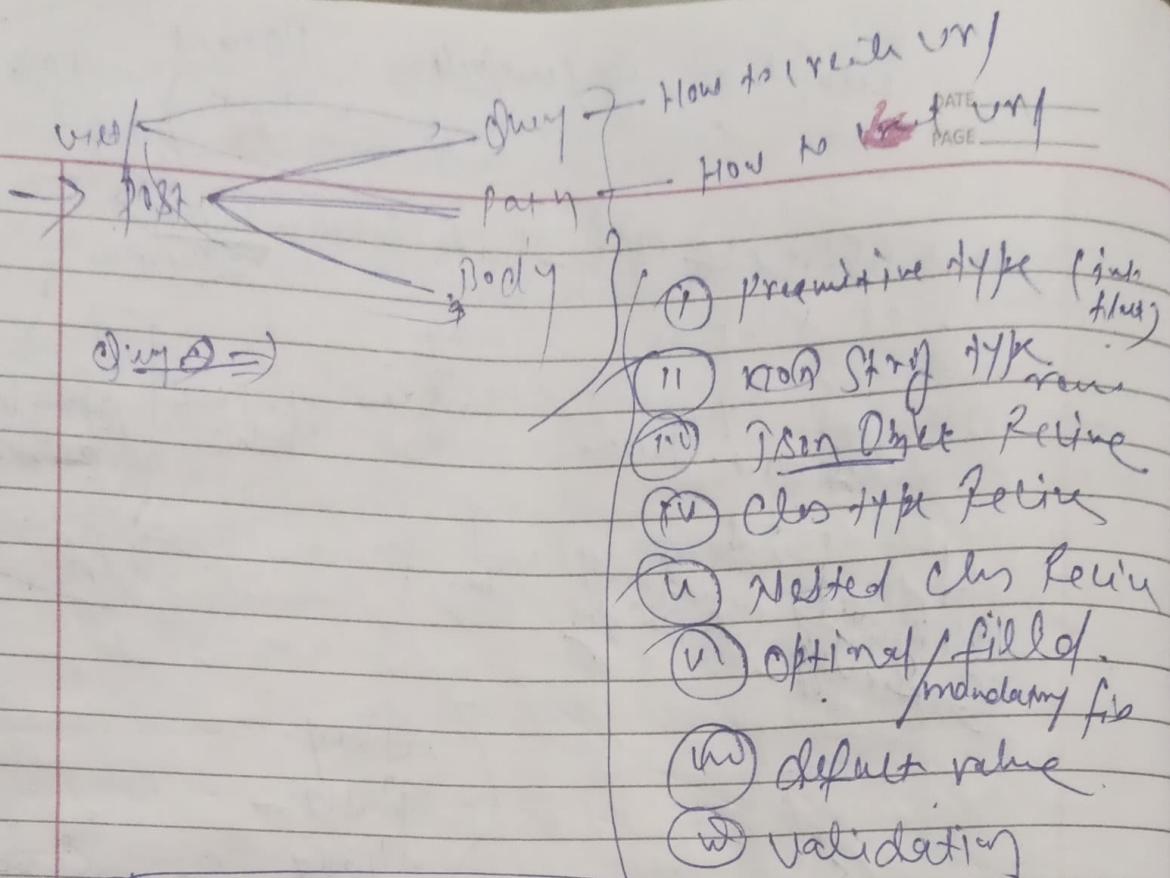
→ How to default value in Query/Path param

→ How to Create ~~to~~ Request Class in ~~param~~ ~~object~~

→ Can we receive primitive type in controller ~~and Path param~~

→ How to Create nested Request Class ~~and~~ ~~in~~

Query param and Path param and body



(v) Get date in header

→ Idempotency method (post and patch  
is not idempotent)  
→ `requestParam()` / `pathParam()`

→ Can we use path and request both in style API (yes) -

→ Body size limit vs weight limit

Used query with  $\Rightarrow$  ① Sort by, ② filter, ③

⑧ put @ with  $\Rightarrow$  ⑨ Delete by id Sesung

- Marshaling / Unmarshaling
- Serialization vs Deserialization

→ Content-type ↗ Accept. ↗ Content-length vs  
producer vs Consumers

→ Content negotiation (xml or json)

⇒ How to validate Query / Party from Validator  
S/3212 last

→ Response Entity class way,  
Can we accept and return JSON Object.

→ write a simple APP (what you will do)

(I) Other get way

(II) XML / JSON

(III) input and output (req / res)

(IV) XML / JSON (both)

(V) GET / POST / Delete / PUT

(VI) Request Class define package (Send, Create, Update, Delete)

(VII) Create Controller Class and @PostMapping, by

(VIII) versioning (V1 and user / admin)

by @RequestMapping (V1 or V2)

(IX) Create Handler method and define  
URL of APP POST / GET /

(X) Create Request Class

(XI) validate Request Class by Spring Validation

(XII) Generate Service Class by @Service

(XIII) Create method for Create for primitive type

(XIV) Attach Service by Controller

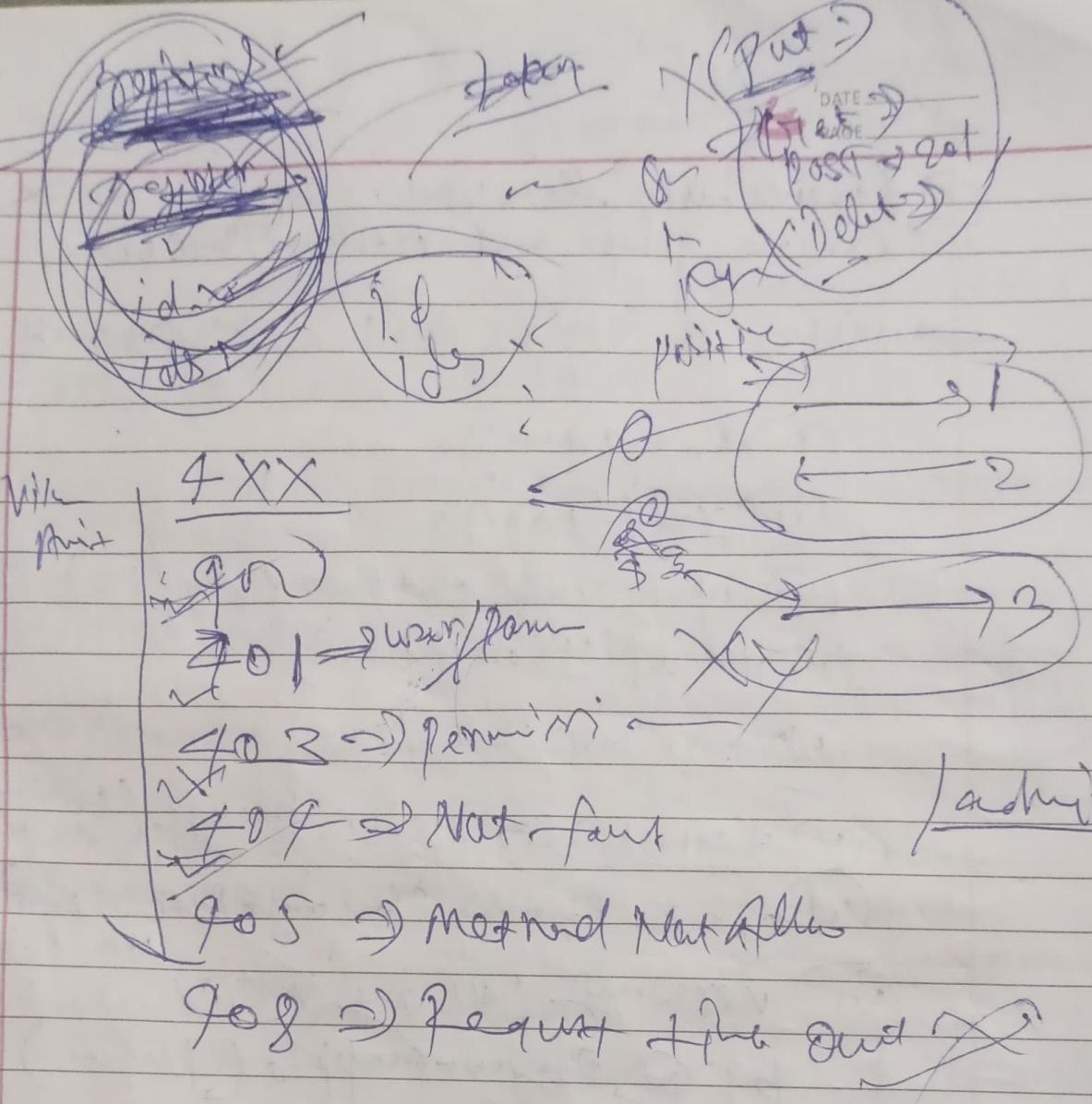
and Call Service by @Autowired

(XV) Create Repository by implicitly extend  
interface IPA or card  
repository

- (W) Create entity class by annotations ~~(D)~~ annotation
- (W) Create interface Repository interface and
- (W) Annotate Repo by @Repository. DATE \_\_\_\_\_  
PAGE \_\_\_\_\_
- (W) Write custom query / findBy SQL / predefine query inside Repo
- (W) Attach mapping by Service with @Autowire
- (W) Create Create Response Class
- (W) Use ResponseEntity to return response
- (W) Use Correct Status Code like ~~200~~ 200, 201, 204
- (W) Actions 200, 201, 204
- (W) Documentation by Swagger
- (H)

~~General  
API practice~~

- (W) Naming Conventions Convention of URL / variable
- (W) Versioning
- (W) Auth / Authorization
- (W) Proper validation in return / response
- (W) Correct Status Code.
- (W) Correct Use of create / post / query / Path / Body
- (W) Best + use # Consumer / Producer Content type
- (W) Del ~~use~~ Swagger for docs
- (W) Actions for health checkups
- (W) Less Code in a Single Class
- use Solid principle. one Class  
one file of work.



- How to Call two API Simultaneously
- How to encrypt password in property file.

300

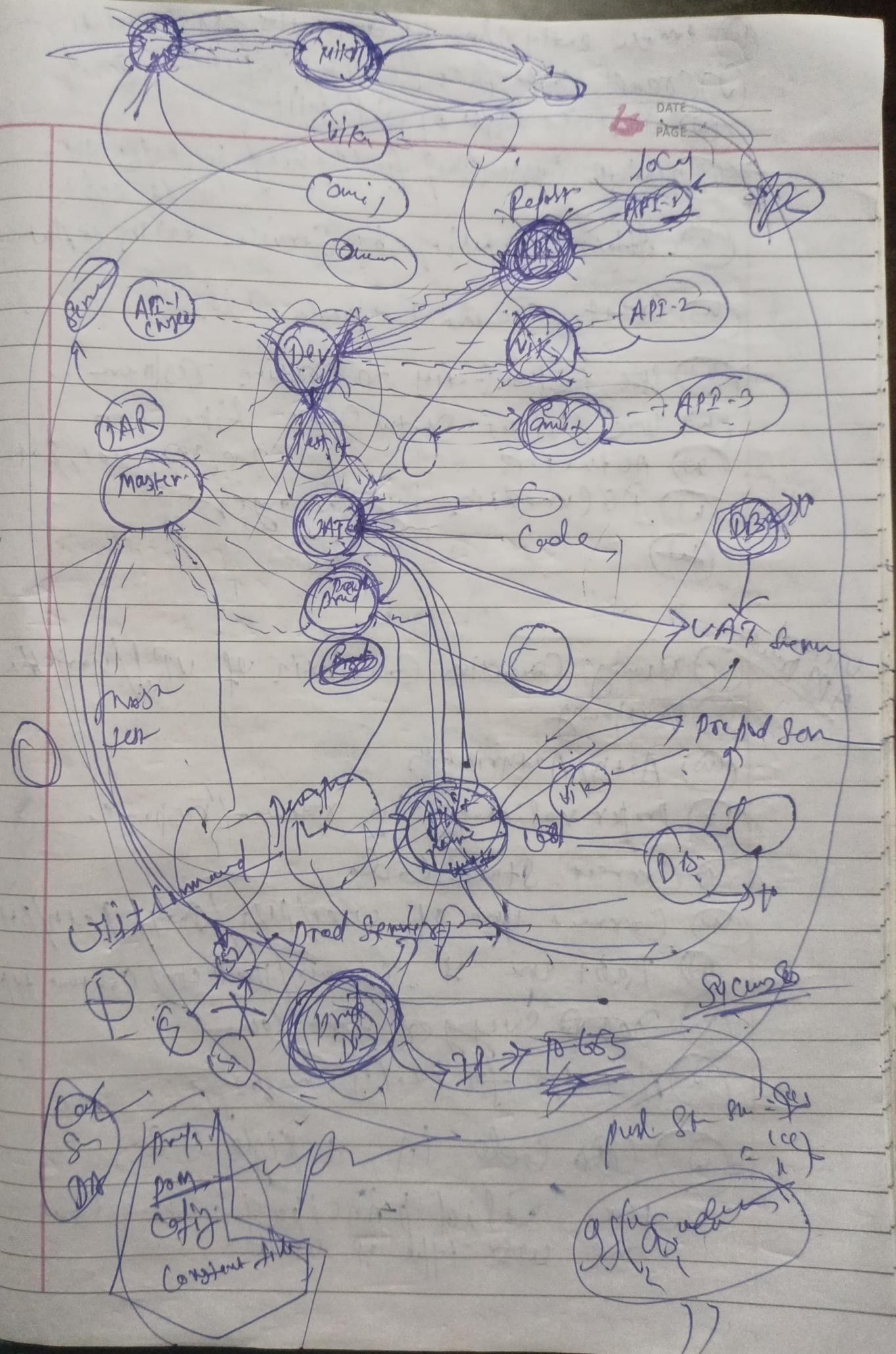
- How to add external Jar in your project:
- Maven
- Central vs remote vs local repository Repos.
- Aim to create Build Jar in Spring Boot.
- Git vs GitHub vs GitLab vs SVN vs Code Commit
- Git add, git add;

- Git init, git, r
- Repository vs branch.
- Master (prod. branch) / QA Preprod / UAT env.

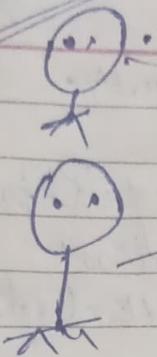
→ Git clone → <sup>Nikhilman</sup> URL of Repo.

→ Git add, git Commit, git pull, git push,  
 Git pull, git stash, Create new branch, Clone  
 particular branch, git merge, git  
 (Switch,

- Check current branch (git branch)
- Revert Commit from branch.
- Cherrypick in git (Move prod to UAT  
 copy)
- Revert Commit.
- Revert from add.
- Revert pull.
- Check logins

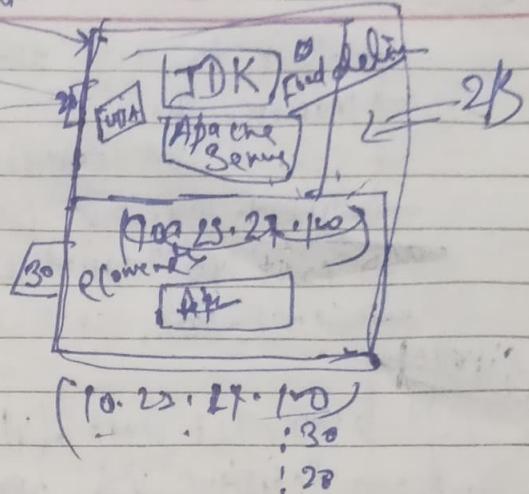


(All in home  
on-premises) Infrastructure



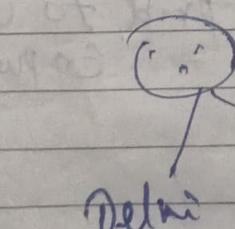
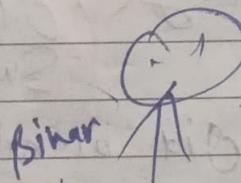
aws

Internet connecting (partie 2)  
PAGE



⇒ Remove problem by Cloud (AWS, Google Cloud, Azure)

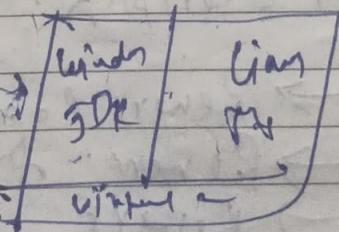
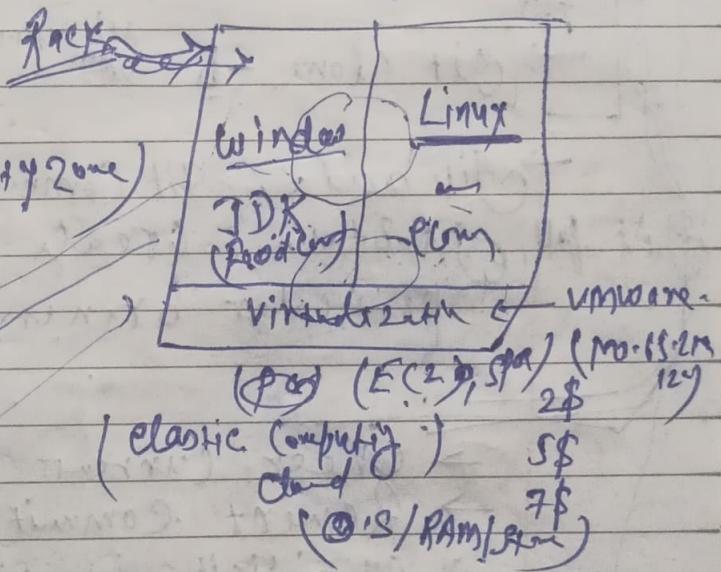
- Virtualization
- Mumbai (Region)
- Sva, Spb, SPC (Availability Zone)
- Mumbai



Delhi

Router

Load Balancer



(E2 SVA)  
(100.65.27.123)

→ Routing path base Router + Load Balance = Load Balancers

www. vica. com / Bus / book ⇒ Send on Bus server

www. vica. com / flight / book ⇒ Send in flight ser

Send Path base route on target Path.

- Docker have Load balancer (LB) have DNS name or ip
- LB IP set in Route 53 (DNS)
- LB on multiple AZ (Availability Zone)
- LB use actuator for health check of Application domain of (UTC) given time period.
- AutoScaling is Server instead of diverse automatic on load.
  - Horizontal Scaling on Cloud
  - Vertical Scaling based on CPU memory at EC2 Select
- Shell script is a command of Linux

- Docker is a Containerization methods and we not required to install any SW on PC to run application.
- We used Docker files to give instruction to Docker
- ECR is just like Docker in AWS (Elastic Cloud Registry)
- Similar creating method in AWS.
- ① non-Containerization (EC2, LB, AutoScaling)
- ② Containerization (ECS, Kubernetes)

- ECS used Docker Swarm just like kubernetes
- ECS given by AWS for Container
- Kubernetes given by Google for best orchestration of cluster POD and used for Containerization
- AWS have also Kubernetes Service Called (K8S) (KES)

- Kubernetes is used .yaml file extensions
- AWS Lambda Service is serverless service and ~~not use Docker~~ <sup>not use Docker</sup> and ~~image~~
- ~~AWS~~ Target is just like Lambda and Docker images.
- Pay as use Service is Lambda and Target.
- Pay as use Service is Lambda and Target

~~AWS Services~~

SaaS → IAAS :- Infra as a Service (EC2, autoscale)

PAAS → Platform as a Service.

CA2

- Elastic block storage like pendrive / harddisk.  
EBS Stores operating system and other local files. It gives 8GB of space by default. More space will cost. It stores OS and some files. It is fast due to local storage.
- EFS (elastic file system) for distributed storage. It also gives 3TB of free and some will charge. It is just like Google Drive, Drop box. It is same store. It can attached to EC2 instances.
- SB Storage (Simple Service Storage) for big data like files, image/video/files. It provides URLs of image, video/audio. It can not be attached in EC2 machine.

Storage S3 Storage Class :-

- ① frequently access
- ② ~~mid frequency~~ - frequent access
- ③ ~~infrequent~~ (after 1 year)

→ CloudWatch used to check server like  
RAM / CPU utilization

→ SNS (Simple Notification Service)

① Message / email / Alert / Call

② SMS Service for OTP (Simple SMS service)  
SES (Simple Email Service)

both services need Cost for sending.

→ AWS Shield and firewall and WAF Web application firewall, and antivirus to load or balance traffic.

Monitor attack like DDoS and other.

→ DDoS attack is more request from single IP. It's stop by AWS Shield.

→ SQL injection is when SQL query send by user in edittext. It's stop by WAF web application firewall.

→ AM service is paid.

→ CDN (Cloud Delivery Network) is also called Edge location in AWS.

→ Latency is how much time take to access website.

→ latency should be minimum.

→ CDN have also Cost.

→ RDS (Relational Database Service)

MySQL | Oracle | MongoDB | DynamoDB.

It has some and it is costly.

→ virtual private Network (VPC) used for

Step E2 access outside  
files.

→ Security groups used to Spec IP who can access me on E2 VPC.

Terraform

Nsible