SDLC for the project.

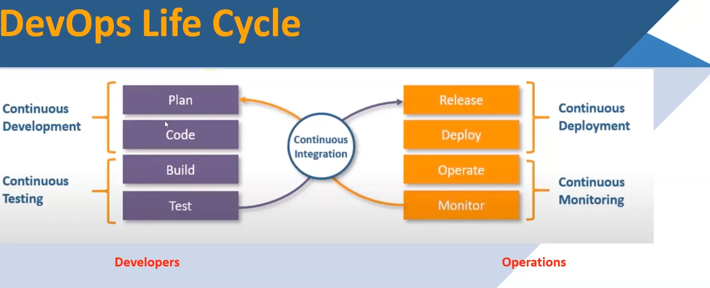
* Planning -> Defines project goals ,scope, resoureces, costs and risks.
* Requirement analysis -> Involves gathering, analyzing ,and documenting functional and non functional requirements.
* Design -> Translates requirements into a blueprint for the application, covering architecture, components and user interface.
* Implement -> Actual coding and development take place based on design specifications.
* Test -> Conducts various test such as unit testing, integration testing, system testing and user accept testing(UAT).
* Deploy -> Releases the software to users after through testing.
* Maintain -> Ensure the software continuous to operate as a required. Address bug fixes etc.

SDLC Models

1. Waterfall Model :-> A linear and sequential approach, where each phase must be complete before the next begins.
2. Agile Models -> an iterative approach that emphasizes flexibility, collaboration and frequent releases.

DevOps -> DevOps started in 2007 by Patrics debois

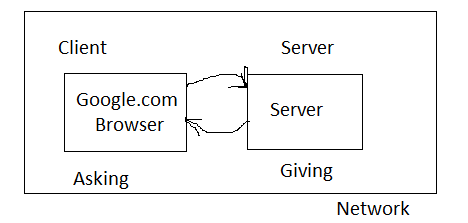
DevOps lifecycle



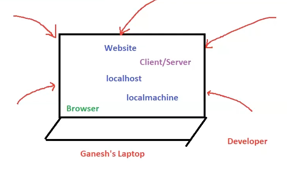
Client Server Architecture

* Client = which required a resources
* Server= which response to the resources

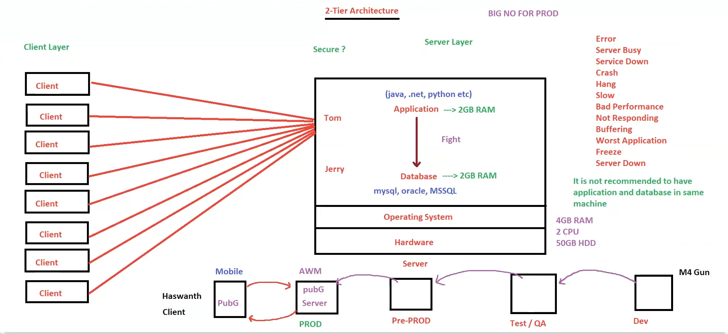
note – Resource = information



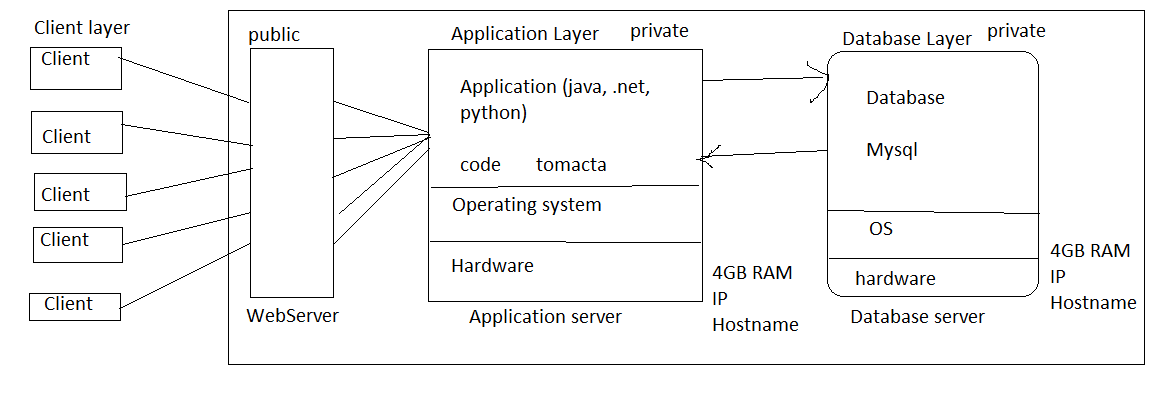
* 1 – Tier Architecture



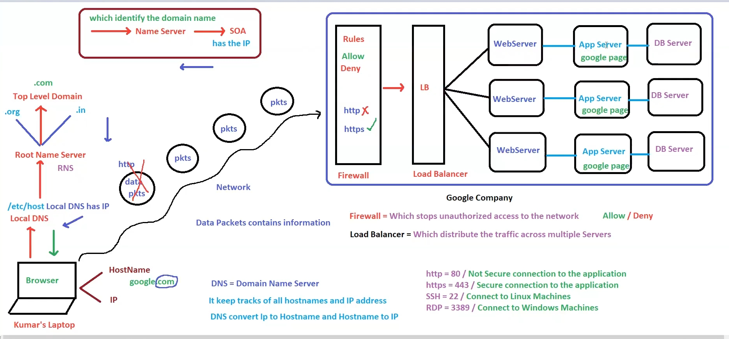
* 2 Tier Architecture

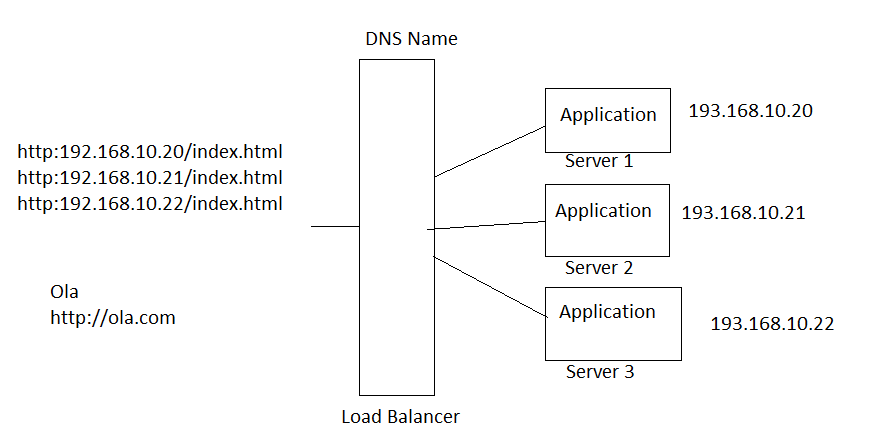


3 Tier Architecture



* Application Server = The server which has application hosted is called application server.
* Database server = The server which has database installed, the server where the database is stored is called a database server.
* Device communicate with each other in the network with IP and Hostname.
* IP (Internet Protocol) = IP is unique identifier for a device in the network
* Host name = name of the server.
* Webserver = Takes the request and redirect to the application server
* Application server should be always in private network, don’t expose to Internet

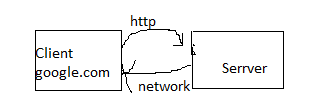




* Round robin method
* Load Balancer = which distribute the traffic across multiple servers

Protocols

* HTTP – Hyper Text Transfer Protocol.
* HTTP default port number 80.
* HTTP transfer the data to and fro from browser to server.



* For customer it should be always default port number (80 or 443).
* htpp://google.com - > protocol://domainname:portNumber
* <http://192.168.10.20:80> –wrong

<http://192.168.10.20:8080> – correct

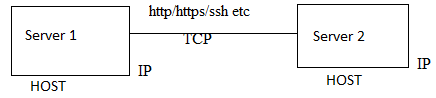
<http://192.168.10.20> – wrong

* You can customize the port number at application level but for customer, it should be always default port number.
* HTTP Status code

404 – page not found , 500 – Internal server error , 503 – service unavailable

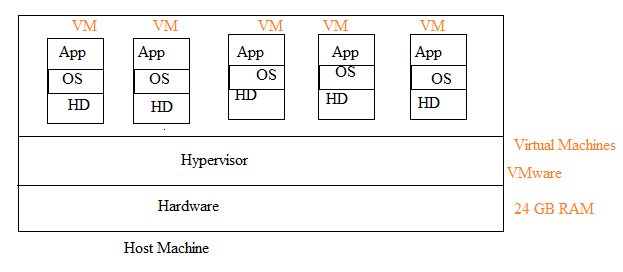
200 – page found, success

* HTTP – is not secure
* HTTPS is secure https = 443 certificates SSL/TLS/HTTPS – Encrypted.
* HTTP Over TCP/IP (Transmission control protocol)
* TCP establish the connection between 2 hosts.
* TCP is like a bridge.



DataCenters

Data Centers which are handled by us is called on-premises .



* P2V Migration (Physical to virtual)
* V2C Migration(Virtual to Cloud)

AWS

* AWS has global infrastructure AWS is providing infrastructure as a service , cloud is present in the remote location remote location contains datacenters datacenter contains infrastructure
* Infrastructure contains storage, database, servers, network, VM etc .
* AWS has approximate 36 place his infrastructure
* we need Internet to connect to the cloud
* Amazon Web Services access through Amazon management console
* AWS is a group of services AWS is a cloud provider who provide infrastructure as a service.

Cloud computing

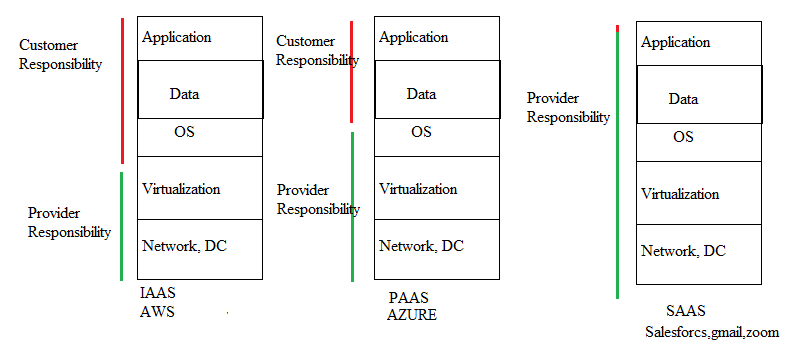
* Instead of doing computing on on-premises/ local machine, now you are doing computing in the remote location(cloud) that is called cloud computing.

Deployment model in cloud (Type of clouds)

* public cloud : - the providers services which are accessed by everyone like AWS, Azure, GCP etc.
* private cloud : - the providers services which are access within the organization like Oracle, IBM
* hybrid cloud : - the combination of public and private cloud.

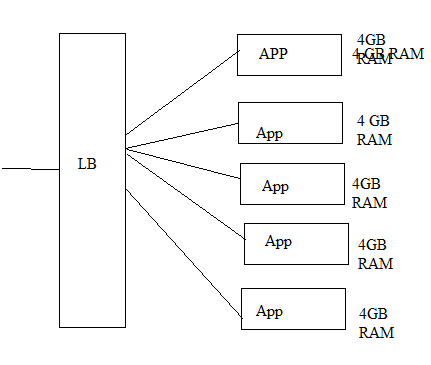
Service model

* Infrastructure as a service (IAAS)
* Platform as a service [PAAS]
* Software as a service [SAAS]
* AWS does not have any access inside your VM .
* ElasticBeanStalk = Easy and quick deployment of application in AWS.
* AWS work on share responsibility model.
* AWS is a group of service we can access AWS service through Amazon management console
* VM = instances
* EC2 = Elastic compute cloud.
* EC2 is AWS service where we can create virtual machines
* EC2 is aws service where we can launch AC2 instances.



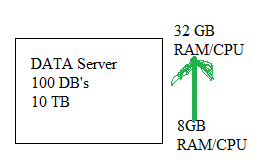
Elasticity

* Increasing and decreasing the number of servers/instances based on the load is called electricity
* Elasticity is a short term
* Elasticity can be achieved in AWS using auto scaling
* Auto scaling = scale out[increasing ,adding] and scale in[decreasing ,removing]
* Elasticity is also called as horizontal scaling
* Use the same capacity of the server is auto scaling



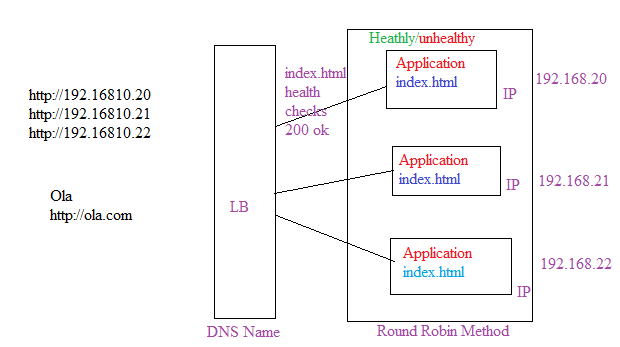
Scalability

* Increase the capacity of servers is called scalability
* Scalability = scale up and scale down
* Scalability is also called as vertical scaling.
* Scalability is long term.
* Scalability can be achieved in AWS by changing the instance type.
* Instance type = memory +CPU



High Availability

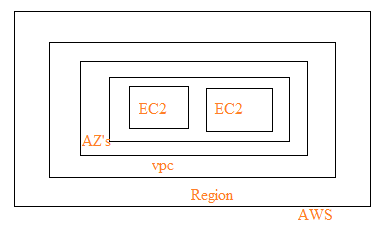
* The period of time the service is available to the customer is called high availability
* The period of time the service is not available to the customer is called downtime.
* Load balancer will do the health checks for application not a server
* Load balancer is doing the failover.
* Redundancy = duplicate/ having the same application on different servers
* Monitoring = load balancer will check if application is reachable or not using health checks
* Failover = if one server goes down, other server will take the request send by load balancer
* 0 downtime = auto scaling = fault tolerance



Regions and Availability Zones.

* Region = Its a geographical area, example AWS- Region = Mumbai
* Availability zone = Simply a Datacenter[AZ]
* AWS has global infrastructure
* Region is a place where AWS has its infrastructure
* A region has multiple data centers, A region has multiple DataCenters
* A Region has multiple AZ’s.
* Server = Instances Server/Instance are placed in AZ’s
* AZ’s are sync with each other[network], not a data
* Best practice is to distribute the instances across multiple AZ’s
* 1a or 1b or 1c = group of DataCentres
* 1 AZ’s is a group of DataCentres
* Instances across AZ’s can share the data if required as AZ’s are inter-connected with each other.
* Load balancer can distribute the traffic to multiple EC2 instance across AZ’s
* LB is a specific to region not a AZ’s
* Mumbai = ap-south-1 |||| AZ’s ap-south-1a, ap-south-1b, ap-south-1c
* Region and AZ’s are managed by AWS
* AZ’s can communicate with each other by default
* AZ’s network are inter-connected
* Region don’t communicate with each other by default, if required yes.
* EC2 instance is a specific to Region and AZ

VPC (Virtual private cloud)



* Every Region contains a default VPC
* 2 VPC’s will not communicate with each other by default , if required yes.

EC2 (Elastic compute cloud)

* In EC2 service, we can launch EC2 instances
* Servers = Instance / EC2 instance (VM’s)
* AWS service can be either regional or global
* EC 2 is a regional service.
* Load balancer = which distribute the traffic to multiple servers.
* Elastic Load Balancer(ELB) = ELB distribute the traffic to multiple EC2 instance across AZ’s.
* ELB is completely managed by AWS (HA,AS, scalability, performance etc
* ELB is a service from AWS not a server.
* You can’t login into ELB, you can access ELB with DNS name .
* ELB doesn’t have any AZ’s it is created at regional level.
* EC2 to launch easy , configure ,deployment, maintain.
* BeanStack - just upload the application and give some configuration.
* Elastic BeanStack = easy and quick deployment of application in AWS ,in general ,PAAS -> you don’t have any control on the servers
* Backbone of BeanStack EC2 instance.
* AWS BeanStack you have full control on EC2 instance launch by BeanStack
* BeanStack handle EC2 instance behalf of us.
* LightSail = if you want to setup and create a virtual lightSaill instance which already have everything installed and ready( WordPress, GitLab, Node jS, Joomla ) NO HA, NO AS, NO scalability.

Lambda