**Waterfall methodology:**

Waterfall model is a traditional approach of software development.

Requirement analysis --> Design --> Development --> Testing --> Maintenance

There is no/less communication b/w client and developer.

**Agile methodology:**

We are working on sprints, communication between client and developer has been improved.

Standard Sprint cycle - 14 days

**Devops:** Development + Operations - Advanced version of Agile.

In agile methodology Development team & Testing team works as individual team, where as in devops every team will work as same team.

Development Team - Plan, Code, Build & Test

Operations Team - Release, Deploy, Operate & Monitor

For code we use Git & GitLab

==================================================================================

**Types of environments**

1) Development

2) UAT/Staging/Testing (User acceptance Testing)

3) Production

**Client Server Architecture:**

**Client:** Client is the one who sends the request.

**Server:** Server is the one who responds to the request.

1. **1 Tier architecture –** *Client and Server installed on same server(Dev) and it is available over internet.*
2. **2 Tier architecture-** *Client will be on one server and Application & DB on the second server**and it is available over internet(DEV & UAT).*
3. **3 Tier architecture**
4. **N Tier architecture**

Fire wall will manages & controls incoming and outgoing traffic.

Load balancer will distribute the load to App servers

Web servers: Nginix, apache

Web server doesn’t contain any code and it will contains the configurations of App server

Configuration in the sense how the request to be redirected to app server,how our webserver should communicate with app server,

App servers: tomcat, Weblogic & jboss

Application server is used to deploy application.

What happens when we browse any URL in browser?

1)When we type any URL in the browser it will check for Hostname/IPaddress in local DNS(Domain name server) server .

Local DNS path : C:\Windows\System32\drivers\etc

2) If the hostname/IPaddress doesn’t find in local DNS server, It will send the request to RNS(Root name server) server.

3) The RNS server will identify the domain extension(i.e .com/.in/.gov/.tech etc) and send this to “top level domain”.

4) The top level domain this information to Name server(NS). NS will identify the domain.

5) NS will pass this information to SOA(Start of authority). SOA will have the information about all the IP’s.

6) once we get the the IP the request will be sent to   
 firewall🡪loadbalancer🡪Web server🡪App server🡪 DB server

sudo apt install ifupdown net-tools - if you don't see interfaces file cd /etc/network

sudo apt-get install openssh-server openssh-client - if you are unable to connect from putty

sudo systemctl status ssh

sudo systemctl enable ssh --now - to enable SSh

sudo ufw status - to check firwall status

sudo ufw allow ssh - to allow ssh in firewall

sudo lsof -i -P -n | grep -i listen

Linux-yum, Ubuntu-apt & Mac-brew are package mangers to download software's from cloud repository.

rpm packages also used to download software's which are not available in repository, so we will download the software's from official websites by using RPM packages.

apt install docker - will install docker on ubuntu server.

==================================================================================

if we want to login with password less authentication:

1) generate ssh-keygen -t rsa in source system

2) update the ssh key in target system in authorized\_keys file in .ssh/authorized\_keys under home directory of that user.

==================================================================================

systemctl status <process name> - to check the process status

systemctl stop <process name> - to stop the process

systemctl start <process name> - to start the process

==================================================================================

Web server: Most popular web servers are mentioned below.

Web servers will help us to manage/deploy static websites. Static web page will remain same across the globe like fb login page.

1) Apache(httpd) - configuration: yum install httpd/ apt install httpd

default port number for apache is 80

default deployment location: cd /var/www/html

To change port number : cd /etc/httpd/conf/httpd.conf

/etc/apache2/apache2.conf

==============================================================================

How to check process is running or not at OS level

Netstat –na | grep <port number>

Lsof –i tcp:80

Ps –ef | grep –I httpd

==============================================================================

Is it possible to deploy multiple apache web servers on same host?

Yes, by changing the port number we can install multiple apache webservers.

You can see error log messages in cd /var/log/apache2

2) Nginx

apt install nginx -y

yum install nginx -y

if we use -y option then there is no human/manual interaction is not needed

Default Deployment location: cd /usr/share/nginx/html/

Default port number: 80

Default configuration: /etc/nginx/nginx.conf

**In B/w apache & nginx which one is best one?**

1) Both are webservers and used to deploy/server static content.

2) Nginx is pretty faster than apache.

3) Nginx can be used as load balancer with no extra configurations are required but where as apache can be used as load balancer and we need to do some extra configurations.

==================================================================================

how to check the particular process is running

1) systemctl status <process name>

2) ps -ef | grep -i <process name>

3) netstat -an | grep -i <port number>

==================================================================================

Application server:

The below three are major application servers available in market.

1)apache tomcat

2)jboss

3) weblogic

Out of theses we will see how to install/deploy tomcat

Default port number for tomcat server is 8080, you can see the port number in server.xml file.

root@ubuntu:/opt/apache-tomcat-10.1.28/conf# vi server.xml

root@ubuntu:/opt/apache-tomcat-10.1.28/conf# pwd

/opt/apache-tomcat-10.1.28/conf

if the package is not available in the repository, we need to download the software from particular website using "wget".

java binaries are required to install tomcat application server.

- prerequisite install java. to know the java version we need to use java -version.

wget https://dlcdn.apache.org/tomcat/tomcat-10/v10.1.28/bin/apache-tomcat-10.1.28.tar.gz

untar the tomcat software, and go to /bin directory and execute startup.sh

/opt/apache-tomcat-10.1.28/bin

./startup.sh

you can test it by using <publicip>:8080, you will successfully installed tomcat on web page.

if you want to deploy any application(dynamic web pages) please webapps folder

root@ubuntu:/opt/apache-tomcat-10.1.28/webapps# pwd

/opt/apache-tomcat-10.1.28/webapps

==================================================================================

**Git & Git hub**

Git is a tool which used for version control tool

It is used to track all the changes of your source code.

Git is a distributed version control, Alternatives to Git is SVN.

SVN is a centralized version control - there should be continuous connection need to have with centralized repo.

We have 2 types of version controls

1) centralized version control & 2) distributed version control

untracking area --git add(command)--> tracking area -- git commit(command)--> commit

**Git hub:** is a central repository

bitbucket and GitLab alternatives to git hub i.e they are also central repositories

**Git flow branching strategy:**

We have overall 5 branching strategies in git flow

We use "release" branch strategy for production.

We use "develop" branch strategy for Dev/staging & UAT environment.

we use "feature" branch strategy for next updates.

we use "hotfix" to fix the bugs of production server.

We use "Master/Main" is used as a copy of release branch.

we have 3 environments, those are Dev, UAT & Prod.

Git config --list

Git status

Git init

Git add

Git commit

git remote set-url origin <https://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@github.com/RAVIKANTH432/github_aws>

git remote set-url origin <https://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@github.com/RAVIKANTH432/github_aws>

Region is nothing but geographic location.

Availability zone nothing but actual datacentre where exists.

EC2 – elastic could compute

AMI – Amazon machine image 🡪 it will be consisting set of software configurations to communicate with our hardware.

What are the different types of AMI’s.

1. Quick start AMI:
2. MY AMI:
3. AWS MARKETPLACE AMI: it’s mostly third party AMI’s
4. Community AMI:

For launching ec2 instances we need to check with our internal team like TL.

WE can loginto our EC2 instance in 2 ways.

1)CLI

2)GUI

a) install AWS client

b)generate accesskey from console

AKIAQWHCPX534KL5ZSP3 - Accesskey  
h8SNyfIpk2KjhDMnafUaOZUUxkP9KLLJfAzMsT1G – secret access key

aws client

aws –version

aws ec2 describe-instances

aws configure to configure your aws account on windows.

Then we need to provide our access key, security key, region & default output format.

You can see your aws configure file on below path.

How to install AWS client

<https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html>

based upon your OS version, we need to select the binaries and copy paste in your cmd prompt to install aws client.

*By mistake you have created your ec2 instance in different region, is it possible to move that ec2 instance to different region?*

*Yes, we need to take snapshot on existing region and we need to resore in the actual region.*

*Which type of instances used in your environment?*

*T2x.large*

*Please focus on Capacity planning*

**User data/bootstraping:**

With the help of user data/boot straping you can install any software without logging into the instance during run time.

**Purchasing options for ec2:**

1)on-demand instances:

Flexibility: Pay for **compute capacity by the hour or second** (depending on the instance type) without long-term commitments.

Use Case: Ideal for applications with **unpredictable workloads** or that cannot be interrupted.

2)Reserved instances:

Cost Savings: Commit to using a specific instance type in a specific region for a one or three-year term to get significant discounts (up to 75% compared to on-demand pricing).

Types:

* Standard RIs: More cost-effective for steady-state workloads.
* Convertible RIs: Allow you to change the instance type and still get savings, but with less of a discount compared to standard RIs.

Use Case: Best for predictable workloads.

3)spot instances:

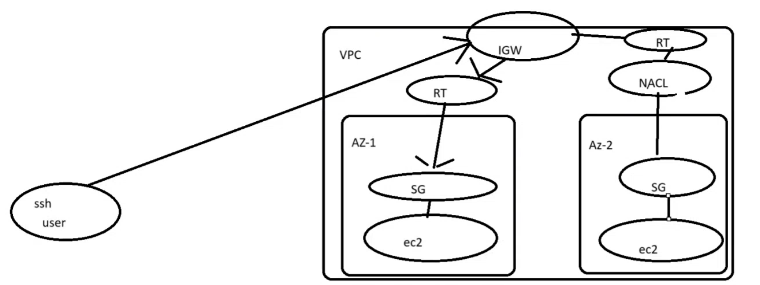
Cost Efficiency: **Purchase spare EC2 capacity at reduced rates, often up to 90% off on-demand** **prices.**

Risk: **Instances can be terminated by AWS when they need the capacity back**, so they’re suitable for fault-tolerant and flexible applications.

Use Case: Good for batch jobs, big data processing, and other flexible workloads.

**Security group:** is nothing but a firewall which is working at **instance level.**

**VPC(Virtual private cloud) in AWS | Vnet(Virtual network) in Azure**

****

IGW: Internet Gate way: To provide inbound and outbound access

RT(Route table) : To define traffic routing table.i.e which ec2 instance we need to connect.

NACL(Network Access Control List): NACL will restrict the access at subnet level.

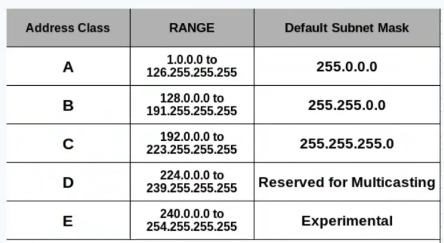
SG(Security Group): SG will restrict the access at instance level.

Both are firewalls, both will help us to restrict unauthorized access.

***Differences b/w NACL and SG:***

1. Both are Firewalls which will help us to restrict unauthorized access.
2. NACL will work at subnet level.
3. SG will work at instance level.
4. By default everything is denied at SG level.
5. In NACL we can allow or deny the request.
6. NACL will work based on rule number.

**CIDR(Classless inter Domain Range):**

****

Class B & Class C IP addresses used for public IP addresses.

Public IP:  
Private IP:

/16 – subnet masking – will help us to classify our IP addresses.

Subnet is smaller portion of VPC.  
How many types of subnets?

1)Public subnet: Having access to internet.  
2)Private subnet: No internet.

DHCP(Dynamic Host Control Protocal) will assign IP address to our instances.

AWS will reserve 5 IP addresses for internal communication.(first 4 + last 1 IP)

192.168.0.0, 192.168.0.1, 192.168.0.2, 192.168.0.3 & 192.168.0.15

**NAT GATEWAY/NAT Instance:**

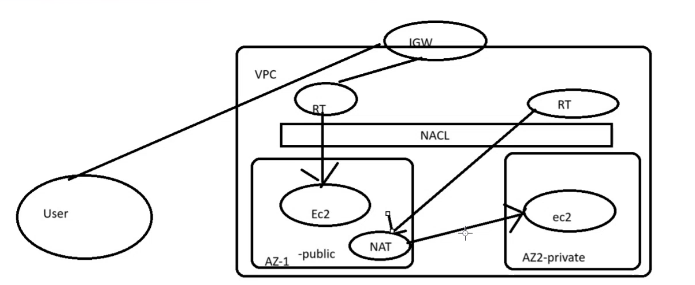
NAT: network address translation

NAT GATEWAY is AWS provided service, maintained and managed by AWS itself.

NAT instance is our regular EC2 instance with NAT configuration and we are responsible for maintain and managed by us.

NAT Gateway will provide internet to ec2 instances which are in private subnet.

We should deploy our NAT GATEWAY always in public subnet.



*Differences b/w IGW and NAT gateway:*

Internet gate way is used for inbound and outbound access.  
NAT is used for out bound access.

Elastic IP is nothing but static/fixed IP address.

Elastic IP address 44.213.205.7 (eipalloc-0e1ecca1ab85791c7) allocated.

**VPC Flow logs:** is useful in monitoring VPC i.e who are accessing the VPC.

We can send the logs to amazon S3 bucket or cloudwatch.

S3(Simple storage service) nothing but storage.

Cloudwatch: is a service available on AWS to monitor our applications/resources.

IAM(identity access management) role: Role is to communicate b/w different services.

**VPC Peering:** will help us to communicate b/w multiple vpc in same region/different/different AWS account.

Peering will establish bidirectional connection.  
Billing will be charged based upon amount of data transferred.

No two VPC CIDR range should be overlapped.  
VPC1: 192.168.0.0/24  
VPC2:192.168.0.0/24  
  
**VPC transist gateway:**

**IAM(Identity access management)**

IAM identity: identity can be a user, group or role.

Console/GUI access

CLI/programmatic access

The are some services are global and some services are regional based.

Eg for global S3bucket,IAM

**Policy:** defines permissions of the IAM identity. i.e what are the different types of permissions need to define user/group/role.

**Types of policies:**

1. **Identity based policies**

Identity based policy is nothing but it is user/group/role based policy. Identity based policies can control what actions an identity (users, groups of users, and roles) can perform, on which resources, and under what conditions.

Identity based policies are 2 types

1a) Inline policy: Inline policy is a policy which is created for single IAM identity. Inline policy maintain strict one-to-one relationship b/w policy and identity. The inline policy get deleted when we delete the identity.

1b) Managed policy:

1b-1) AWS managed policy: AWS managed policies are created and managed by AWS.

1b-2)Customer managed policy: customer managed policies are created and managed by us in our AWS account. Customer managed policies provide more precise control over your policies than AWS managed policies.

1. **Resource based policies:**

Resource based policies attached directly to resource and specify permission for specific action on the

Resource based policies grant the specified principal permission to perform specific actions on that resource and defines under what conditions this applies.

1. **IAM permissions boundaries**

IAM permission boundaries define maximum permissions for an IAM entity and are used as safeguards.

A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity.

1. **Service control policies (SCPs)**

SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU).

1. **Access control lists (ACLs)**

Access control lists (ACLs) are service policies that allow you to control which principals in another account can access a resource.

Amazon S3, AWS WAF, and Amazon VPC are examples of services that support ACLs.

1. **Session policies:**

Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user.

**Role in IAM:** We can use roles to delegate access to users, applications, or services that don't normally have access to your AWS resources.

Policies will assign to users/groups. Roles will also have specific permissions which will not be assigned to users. We will assign Roles to communicate to AWS resources.

**EC2 snapshot:** Snapshot are nothing but backup of your instance which can be used to build the new instance with old snapshot.

**EBS volumes: (Elastic block storage**): nothing but our storages.

We need at least on1 EBS volume to start our instance.

**S3 Bucket:** Simple storage service and it is a global service.

IAM, S3 , route 53, CloudMap.

S3 is global service but the bucket we create in S3 is regional specific.

S3 bucket name should be unique globally.

**Different types of storages available in AWS:**

1. S3 Bucket
2. EBS volume
3. EFS
4. RDS

S3 is for fixed objects. Object can be file, image, PDF & anything. i.e when your we upload any object to S3, we don’t have option to edit that file.

We have used S3 bucket to

1. To store the images of Application
2. To store the logs of VPC flow logs.
3. To store the logs of load balancer.
4. To store the terraform state file.

EBS volume (Elastic block storage is used to store the Persistent(Permanent) data.

EBS is used to attach Ec2 instance. We can’t attach multiple Ec2 instance to same EBS volume.

EFS : Elastic File system is one type storage will get extended automatically based upon usage. We can attach multiple Ec2 instances to same EFS volume.

RDS: Relational Database

We have 2 types of policies in S3.

1) Bucket level policy

2) Object level policy

ACL(Access control list):

Versioning: will help in storing the multiple versions(modifications) of same file.

Storage class/Object Life Cycle in S3:

1. Standard
2. Standard infrequent access
3. Glacier
4. Deep archive Glacier

**Different types of monitoring services available in AWS:**

**1)Cloud Trial  
2)Cloudwatch**

[AWS CloudTrail](https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-user-guide.html) monitoring service helps you audit the governance, compliance, and operational risk of your AWS account by recording the actions taken by a user, role, or an AWS service.

Cloudtrial is a monitoring service that will help us to track activities on our AWS account.

These recorded sessions are called events.

Cloudtrail by default enabled on AWS account when you create it.

We can see the events in Event History, where you can view, search and download the past 90days activities in our AWS account.

We have 3 types of events

1. **Management events:** Capture management operations performed on AWS resources.

We can create one management event by free of cost.

Management events will be created by default.

Eg: who signed in at what time and other API calls executed on resources.

1. **Data events:** Logs the resource operations performed on or with in a resource.

Eg: whenever user uploaded, deleted and downloaded files in S3.

**3) Insight events:** Identify unusual activity, errors or user behaviour in your account.

Eg: Unauthorized API calls indicate someone tried to perform an action in your AWS account that they don’t have permission to carry out.

**Jenkins**

Jenkins is automation server. Jenkins is open source and having multiple plugins which help in integrating with most of the tools

Jenkin is used for CI/CD.

CI: Continuous Integration

CD: Continuous development

Jenkin is Java based Web Application.

Jenkin works on port number: 8080

Java-11 is pre requisite for installing Jenkins.

Jenkin workspace: Jenkin workspace contains job related information.

Maven is build tool, to compile the source code and creates the package(Eg war) file.

Sonarcube is responsible for static code analysis i.e it will check the quality of code.

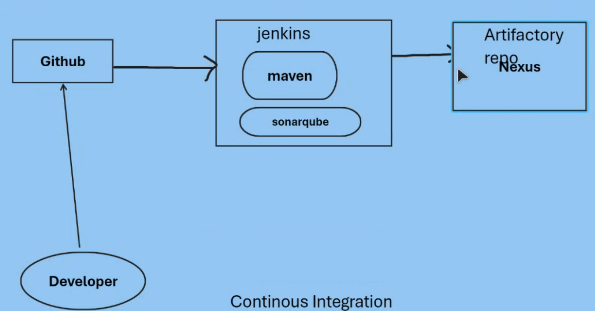
Alternatives to Jenkins: Github, Github actions, AWS Codebuild, code deploy, azure deops

Jenkins Default location: cd /var/lib/jenkins

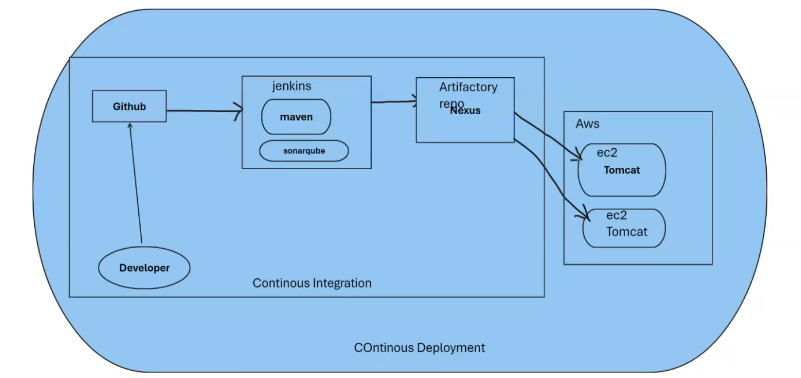
Jenkins supports 2 types of jobs:

1. Plugin based jobs
2. Pipeline based jobs

**Continuous Integration:**



**Continuous Development**



**Upstream & Downstream:**