**Devops Tools**

Git

Maven

Jenkins

Nexus

Sonarqube

Ansible

Docker

Kubernetes

Linux

Jira

CI-CD pipe line using Jenkins file

**AWS Services**

IAM

EC2

S3

VPC

ECS  
EKS  
ECR  
Cloud Watch

Terraform

**Azure Services**

Azure Storage account

Azure data bricks

Keyvault

Aks

Acr

ARM templates

Log Analytics Workspace

Azure Devops Pipeline

AAD🡪similar aws iam(Azure active directory)

VMS

Vnet

Teams

Dev team

Devops team(Build,Deployment,Automation)

Testing Team(Stagging,QA…ENV)

Production Team

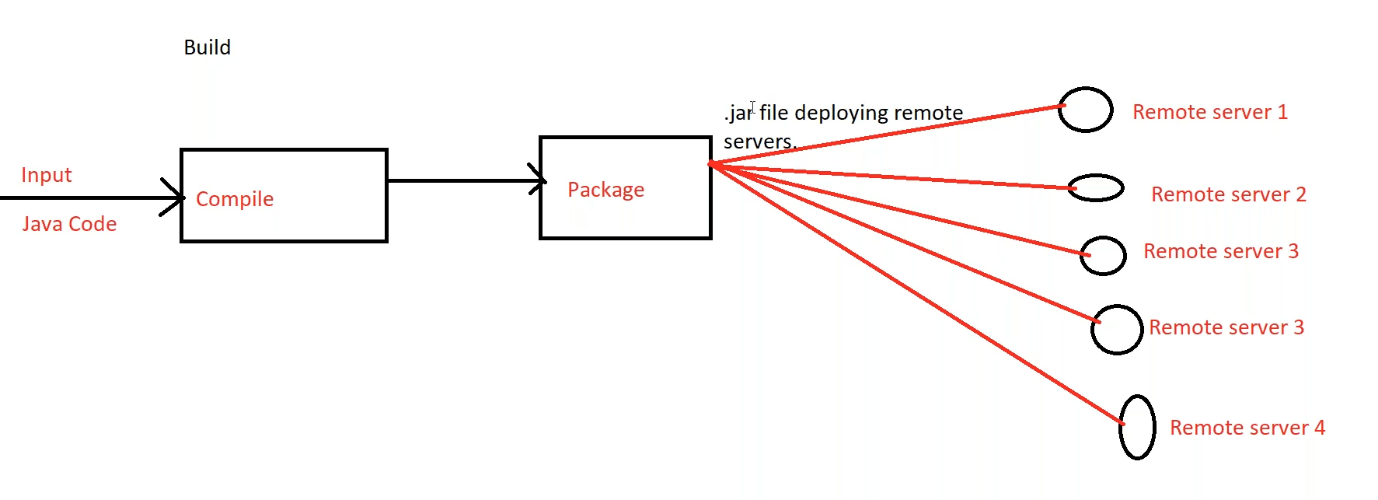
Build🡪creating the package

EX:Jar ,war,tar,rpm

Deployment🡪Installing the packages in remote servers

Ex:Ansible playbooks using for deployment

Automation: Writing the scriptes in terraform ,Ansible playbooks,ARM templates,Kubernates Deployment files,Jenkins files



Git is a version control system tools used to store the source code

Maven🡪Build tool convert to jar (for packaging the source code)

------------------------ **Day 1 Completed-----------------------------------**

Deployment

CI/CD

CI🡪Continuous Integration

CD🡪Continuous Deployment

Sonar qube🡪code quality analysis

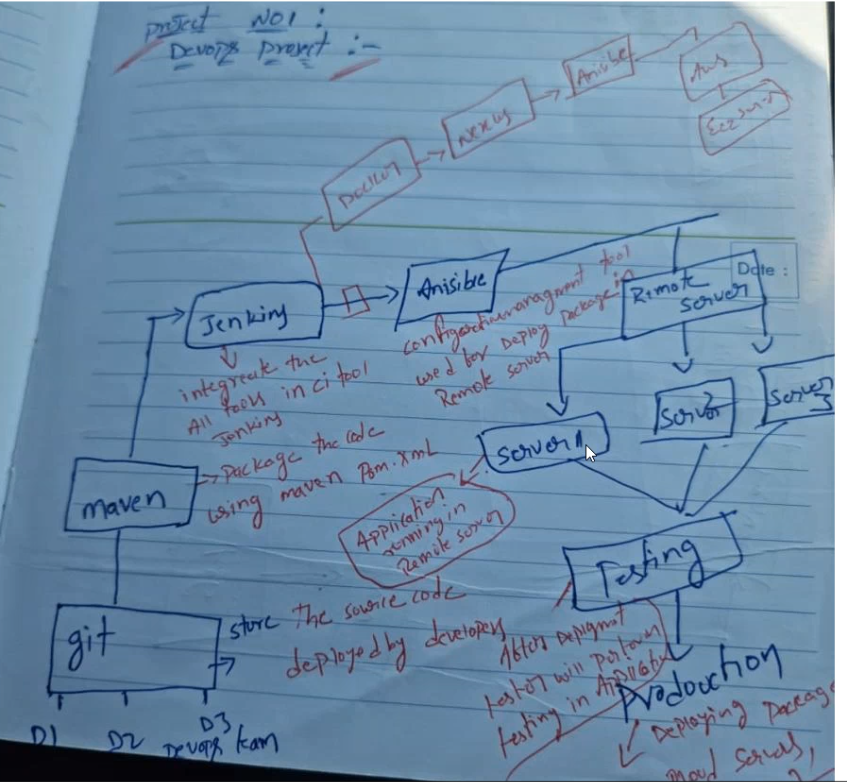
Sonar qube should be integrated to Jenkins only

Ansible🡪Cofiguration management tools we use for CD

Continuous Deployment🡪

In Jenkins we are going to create a package

Using CI-CD pipeline we deploy packages into remote server



Ansible is located between Jenkins and remote servers

Ansible role is take the package from remote server install into remote servers

We need to configure the server ip’s in ansible

Ansible is a mediator between Jenkins and remote server

Ansible purpose instalations(configuration)

Git

Maven

Jenkins

Nexus

Sonar

Ansible

With in Jenkins server we need to install multiple services

(How to integrate Jenkins with other tools that is the task)🡪 this is the responsibilty

Nexus Repository is used for taking backups

In your project which artifact or registory you are using🡪nexus repository or docker hub

Docker hub is for only docker images

But in nexus any file like .jar.war.rpm docker images

Git –code

Maven-build (packages)

Jenkins for continuous integration

Sonar qube—code quality

Nexus –back up

Ansible—Deploy package in remote server

After server deployment testing and then production

**------------------ Day2 Completed-------**

IAM🡪Identity acess management

We need to focus on providing the acess

Creating the groups

Creating the policies

Under IAM we have user data

Acess key and secret key🡪to aces aws account from out side

EC2🡪Elastic compute cloud (use for creating the number of servers)

Each server configuration from front end and backend

Launch instance is nothing but creating the servers

EC2 will provide load balancing and autoscaling

Load balancers are used for docker and kubernates also

Distribute the load according to the configuration

Three types of load balancers

Elastic load balancers

Appn load balancers

Network load balancers

Auto scaling 🡪declare min and max conditions

AMI 🡪entire backup of the instance (servers)

Azure🡪using azure vms for creating the number of servers

S3🡪Using for storage (S3 bucket)

Storage accounts in azure

The naming convention of bucket is unique.

In buckets we store the data

Iam custom policy for bucket 🡪scure(other user acess denied)

Encryption to bucket🡪(if you want to acess the bucket a key is needed)

For encryptions 🡪KMS key management services

S3 lifecycle rules to auto delete

S3 will provide versioning

Azure🡪container same as bucket

Kms🡪key vault(in azure ) same as kms in aws

VPC 🡪virtual private cloud

CIDR🡪limitations

Azuree🡪vnet same as cidr vpc

Azure data brics (it is not available in aws)

Aws🡪amazon

Azure🡪microsoft

Azure data bricks is to run the spark code or spark data

We need to maintain the clustures

We need to automate for the creation and run of clustures from the azure

ECS🡪elastic container service

ECR🡪elastic container Registry

EKS🡪elastic Kubernates service

AKS

Azure Devops Pipeline

Terram form and arm templates🡪backend

Terraform is comapatable for any azure aws and other also

**----------------DAY-3 Ended--------**

**//creation of aws account creating instance and installing git bash**

**AWS EC2 Server Creation**

**Centos 7:**

Right side region untadi (ne name tarvata)

Oka region lo chasina servers inkoka resignlo reflect chayav so rember which region you have created server

Click on the launch instance

Choose an Amazon Machine Image 🡪centos (if it is not there in quick start it will be there in market place)

Centos🡪continue🡪instance type🡪t2.micro(instance type)🡪launch instance🡪Create new key pair(type in key pair name)🡪createkey🡪lauchinstance🡪view all instances🡪give the name to server

//centos server created

Install git bash for windows

Open git bash in downloads ssh -i pemfilename.pem centos@public ip address(right click and paste)

**---------Day 4 ended----**

**Git**

Git is a version control tool

Git clients :

1)Bit bucket

2)Git hub

3)Git labs

4)Girrit

**Git Commands**

Git clone repository name🡪to clone the remote repository

Git status 🡪to check untracked files status

Git add🡪to add the untracked files

Git log🡪to check history or logs

Git diff 🡪to see difference between two changes

Git checkout -b new branch name 🡪to create the new branch in git

Git branch🡪to check the list of branches

Git delete original branch name🡪to delete the remote branch

Git commit-m “message”🡪to comit the changes

Git push origin branch name🡪to push the changes from local to remote repository

Git merge

Git pull

Git fetch

Git pull request

Git reset

Git revert

Git stash

Creation of git hub account

Repository🡪 to store the source code

For each commit or change git will generate the commit ID

Based on commit id will track the source changes

Also git provides you branches based on developer requirements we used to create the number of branches

If file is adding or deleting git will ask the user details like username and password

Based on user details we can track the changes in source code

For each merging git have pull request once your pull request is approved by reviewer you can merge the changes

When you create repository defaultly master branch will create we can call it parent branch

Branch is nothing but sub folder

Merging🡪combin ing the data

Git bash is communicator bwtween local repo and remote repo

While you are first commiting (git commit) it asks for configuration email and username(only for first time)

Step 1 Create repo

Step2 Create folder in desktop and click git bash here And clone by using

git clone repo url

Step 3 cd repo name

Step 4 paste the code in the folder(local repo)

Step 5 git status

Step 6 git add

Step 7 git commit

Step 8 git push

Git add 🡪sends the modified to the stage

Git commit 🡪save the changes

Git push local to remote

**-----Day 5 ends----**

**//practically pushed the code into git hub**

**Can you explain how you people are maintaining git branches**

Branch is a sub folder

When you create repo by default you will be in main branch

Based on developer needs we create branches

We should always maintain the proper branching strategy

Dev branch Relase branch Bugfix branch

When we are going to add new functionality in my application then we will create development branch

Ex:

Dev 1.0

Dev 1.1

Dev 1.2

Release branch

Once development is completed the code will go to testing phase after testing completed then we will create release branch the branch naming conventions followed by below

Rel -dev 1.0-15/05/21

Rel -dev 1.1-17/07/22

After relase is completed In case any issue happened on exsisting releases then we are creating bug fix branches branch naming convention

Bugfix-dev1.0.1

Bugfix-dev 1.3.1

**--------------------Day 6 endss-----------------**

What is merging?

What is pull request?

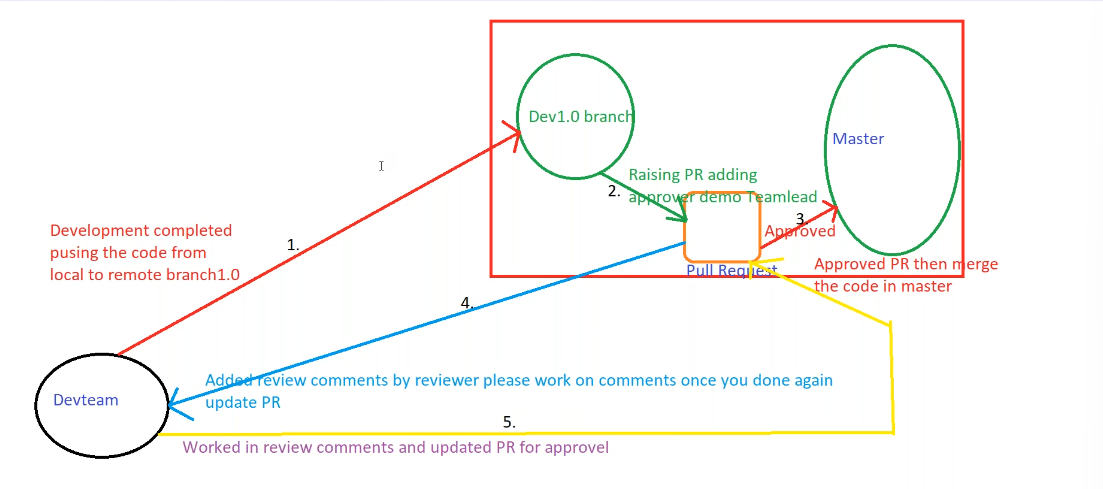
1)Merging is nothing but combining the changes from specific branch to master branch

2)in my project when we are going to merging the code we are following pull request process

3)Under the pull request we are adding reviewer name once I developed the code I will raise the pr(pull request)once I raised pr the request will go to reviewer

4) once my reviewer reviewed the changes then reviewer going to approve the pull request then I will merge the changes from specific branch to master branch

5)incase my reviewer will give any review comments on my code then I need to work on review comments once I completed working on review comments then I will update the pr



Git command to switch from one branch to another **git checkout branchname**

When you create a branch what ever is present in the master branch the same content will be in new branch also (only for the intial time)

After creating the branches if you do changes to the master it will not automatically come to the new branch you need to merge or rebase