Road Map for Deveops

Linux OS 🡪 Knowledge about OS,

Git HUB 🡪 Source code management,

Shell/ Python 🡪 Scripting knowledge,

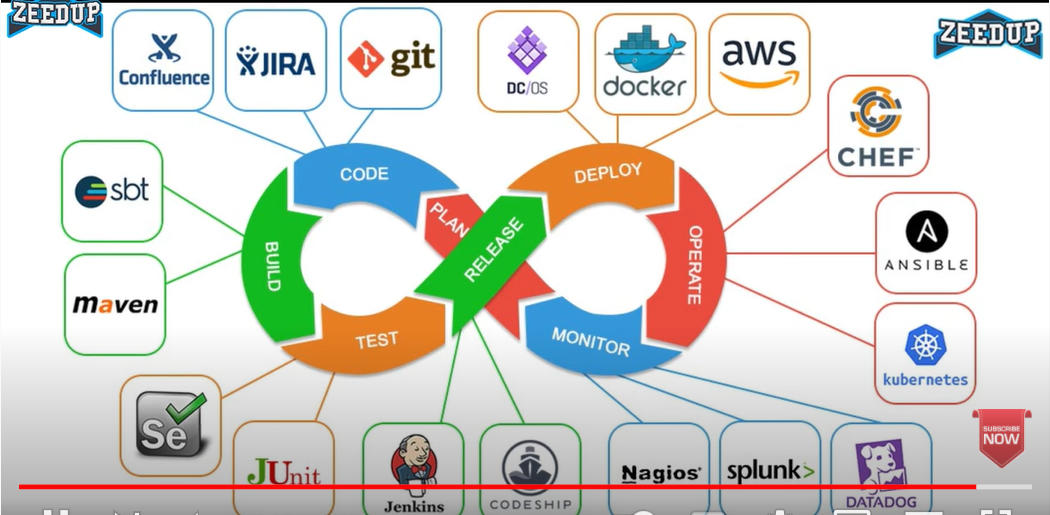
Jenkins / Bamboo 🡪 CI/CD tools,

Ansible/ chef 🡪 Configuration management tool,

AWS/Azure 🡪 Cloud storage tool,

Terraform/ Cloud Formation 🡪 Infra as a code,

Kubenetes 🡪 Container Orchestration.



What is devops and Purpose of devops ?

Devops is a culture, the combination of development & operations. It includes lot tools to deliver an application in a faster way without any pause or any glitches.

It provides communication, integration, automation, and close cooperation among all the people needed to plan, develop, test, deploy, release, and maintain a Solution. DevOps is part of the Agile Product Delivery competency of the Lean Enterprise

Deveops – GIT HUB

Storing a code in git hub and deploying in a server, so that we can access the website in internet – Manual procedure of doing it.

CICD – Continues integration and continues deployment – Automating the code or updating the code online is called CICD. Instead of deploying manually, code can be automatically deployed using CICD functionality.

Jenkins -

Who arranged for interviews call,

https://drive.google.com/drive/mobile/folders/1NutVlzffKdfaNxa07evP86xq16z3cJEp

LINUX:

Linux is the command line operating system.

All the operations are done through command.

Directories – folders in linux

Normal files

Special files

Commands: CAL – Calendar, whoami, who, halt – brings the system down immdtly, shutdown,

reboot.

Pwd is command will tell you on which path or which location we are in.

Ls will show the folder names and file names. Folder name will be in blue color and file name will be in white color.

Ls –l will display the detailed informations of folder and files.

Ls –a will disply the hidden files in the folder.

Vi file name will create a new file

After creating a file we are allowed to enter the txt using the letter “i” and “esc button” will stop or will not allow editing anything in that file.

:wq will help us to save the file and return to the command page.

Cat filename will display the text we have saved in that particular file.

:q! will not save the newly edited txt or file, it will restore the old txt in the file

Using key strokes we can do the following actions in linux OS

Keystroke is nothing but the letter “K”

Keystroke plus yy will copy the current line

Keystroke plus d will delete the entire lines

Keystroke plus d3d will delete from 3rd line

U will undo the previous command

/”Word” will find the particular word in the file

If we r in last line and need to find a word means need to give esc click in “?” and give the word

“:s/good/bad/g” this command will replace the word from good to bad

Cp n.txt to m.txt, this command will copy the entire file from n file to m file

Wc file name will display the word counts in that file

Rm file name will remove the particular file from the folder

Rm file file file name will delete the multiple files from the folder

Mkdir will create a new folder

To create an empty file need to use “touch” plus filename.txt

“rmdir foldername” will delete the folder if it is empty

“rm –rf foldername” will delete the folder even if it has a file in it

“mv foldername1 to foldername2” it will rename the old name to new name

“cd ..” will take u to the previous folder and “cd ../.. will take u back to the two folders like two directories back.

We can give permissions to the owner, group and others

Read has a value of 4, r – read (4)

Write has a value of 2, r – write (2)

Execute has a value of 1, x – execute (1)

Chmod 777 “file name”

777 means give full access to the file name like, read, write and execute.

Grep will search a file or string in a file keyword “grep string filename”

“Sort filename” will sort the file name in ascending order

“sort –n filename” will sort the file in numeric order like 1,2,3

“sort –r filename” will sort the file in the reverse order

“sort-f filename” will sort the file both the upper and lower case together

System related and process related commands

Top command is used to find the number of systems or process in the particular system

If we need to delete a system need to use the command “kill -9 and enter the PID number”

“ps –ef” command will display the process running in the systems

“ps –ef | grep process name” process name in the sense if we need to know python process or java process, we can get from the above command

\*\*\*\*\*\*\*\*\*\*\*\*LINUX BASICS OVER\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Mr devops

Rajesh devops

Mathusuthana reddy devops

Linux, AWS Cloud VPC EC2 S3 Cloud watch lamda , Git & GitHUB, Maven, Jenkins, Anisable, Terraform, Qubernets, Dockers

**GIT and GIT HUB:**

GIT and GIT HUB is code repository which is used to store the codes. From here we can directly upload the code to the server.

GIT or GITHUB usually work under VERSION CONTROL SYSTEM. It is a repository which is used for multiple people to work on the same file at the same time. Here we can find who made the changes and also can switch back to the older version.

Each machine act as a repository, if there is any glitch in server. Each machine has an original backup which will push the code back to the server. So that there is no lag in the project or process.

Once installed we need to config the git with following commands: **GIT Commands**

* Git config –global user.name “manu”
* Git config –global user.gmail [manu@gmail.com](mailto:manu@gmail.com)
* Git config –list
* Git –help
* Git init – will create an empty repository
* Ls –la – will display complete details of the folder
* After config very first command in git is “**gitinit**”
* Next one is “**mkdir gitclass**” and “**git status**”
* “**Touch filename”** will create a new file with dummy
* Once the file is added, when we are cking the status that particular folder will be untracked. So to add the file to tracked on use the command as “**git add filename**”
* “**git add .”**  this command will add the edited file to the repository
* “**git commit –m “add some words here”**” this command will commit the added files to the repository
* “**git log”** it will display what are the insertion happened in this commit
* “**git log –author=”username”**” this command will display the commits done by particular user like author in the sense the person who logged in
* “**ls –lrt”** will display only the files in the folder
* **“ls-la”** will display both the files and hidden files also.
* Whenever we are entering new text to the existing file, without giving the command “git add .” if we give “git status” it will show error. We can get the diff using the command “**git diff**”.

**Git – It is a version control management tool**

**Github – It is a Centralized repository**

**GITHUB Repository connection of GIT and GITHUB:**

* There is a green color “code” button will be available, copy the URL from there
* And use the command **git remote add origin “url”**
* **Git pull** command will link from main repository to the local
* **Git pull origin master** will fetch the branch master
* If we get error like “**fatal: refusing to merge unrelated histories**” in the above command we need to use this command to resolve the error **“git pull origin master --allow-unrelated-histories**”
* First we need to connect local repository and server repository using the command “**git remote add origin + url of the server repository**”
* Then we need to git commit command i.e., “**git commit –m “command of commiting”**” It will commit in the local repository
* “**git add .**” will help u to add the new deletion or addition of the folder or file
* “**git status**”this will show file which is recently added or deleted
* “**git pull origin master**” will load the previous folder from server repository to local one or if we directly create a file in server repository will also send to local one by using pull command
* “**git push origin master**” will push the edited file or folder to the server repository
* So far we have worked in the “**master**” branch, now we are going to add a new branch
* When we are pushing our code to master existing code or entire project will collapse
* To avoid it each developer need to create their own branch with their name, once the push was done to their branch or repository. After verifying the code it will be merged or pushed to the master branch
* By creating the new branch it will show the new files and existing files also
* But when we are creating a new file in new branch will not reflect in the older or previous branch. Branch is also known as pointers.
* “**git branch branchname**” will create a new branch
* After creating a new file in the branch have to be committed as usual
* “**git branch**” will tell u which branch we are actually in
* “**git checkout branchname**” will switch from one branch to another branch
* “**git merge branch1 master”** in this command we are merging the files from “**branch1**” to the “**master**” i.e., folders from branch1 will be copied to the master branch
* **“git revert branch1 master”** this straight opposite to the merge option it will remove or delete the merged file and return back to the same status of the folder or branch or pointer
* After doing “**merging**” & “**reverting**” actions need to give push command i.e., “**git push origin master**” after giving push command we can see the changes in the centralized server or repository
* “**git reset head <filename>**” this command will reset the command “**git add .**” after editing any file we will give git add . Command to add that particular file to the local machine or local repository. So to remove from local repository we need to give the “**git** **reset head <file name>**” command
* “**git branch –m oldfilename newfilename**” it will rename the files. Eg name of the branch1 will be rename to branch2
* “**git branch –d branchname**” this will delete file from the repository
* “**git stash”** this command will commit the existing code, which will not save the newly added or edited txt to the particular file. **FYI, newly added text will not exist in that file** without giving git add. Command, we can commit using “**git stash** is ntg but **q!**”
* “**git tag tagname**” it is something called creating a clone of the repository
* “**git tag –d tagname**” will delete the particular tag from the git
* “**clear**” will clear all the command we entered and take u to top of the page

Jenkins will be connected to the GITHUB, which will convert the code to JAR or WAR file by writing some codes.

It will also deploy in the tomcat server and do the testing, like unit testing and other things, if all the test cases are passed, the code is ready to deploy in the real server.

Service-oriented architecture (SOA)- This is the old version, like in java all the page for eg login page after that home page all the pages are, so if there is a big billion day. There will be a multiple logins like lakhs and crs, at that time there will be glitches or chaos will happen so to avoid this MICRO SERVIES introduced.

MICRO SERVICES: In this every page will be act or perform as a diff application, it will deploy in the docker, which will have containers (coz it is lit weight), these containers will next moved to kubernetes orchestration. In this kubernetes orchestration we have an auto-scaling concept that will increase the size based on the volume of the logins. It will increase the size if there is a huge login and reduce the size when there is less logins.

CLUSTER: The size increases and decreases based on the concept called cluster of node or cluster of server.