**HOW TO TAKE JENKINS BACKUP :**

Suppose if i want to take jenkins backup what i will do is first i have to install backup plugin from the plugin section after that i have to go to manage jenkins if i go down there i can see the option like BACKUP MANAGEMENT so i have to select that then i will get three options like save and backup jenkins and restore jenkins now what i will do is i will select jenkins backup and i will choose what and all i need to take backup like workspace and build history and jobs etc.

if i want to restore then inside the backup management i will select the restore jenkins ,

**HOW TO CREATE MULTI BRANCH CONFIGURATION:**

how i will create multi branch configuration is first i will click on new item and i will specify the job name there different types of job configuration will be there so in that i will select multi branch configuration and i will enter into that configuration page if i go down there i can see ADD SOURCE so i will click on that i will choose git and i will specify the git url and credential here and i will save the configuration and i will trigger the build.

before we are configuring Git url inside the multi branch configuration first we need to checkin the code and pom.xml and multiple branches and jenkins file into version control system.

------------------------------------------------------------------------------------------------------------

**CONTINUES DELIVERY AND CONTINUES DEPLOYMENT :**

life cycle is ,

Dev Environment -------------> where develpers developing the code

Build Environment ------------> creating the packages

UAT OR QA Environment ---------> It is like where testers testing the packages

Production Environment --------> It is like Live, where we are releasing the packages into live

**[ DEV , QA , UAT ] this is we called as Continues delivery**

Configuration settings will be different in Dev Environment and QA Environment and UAT Environment and Production Environment.

**WHAT IS RELEASE:**

Release is nothing after creating the packages where we are keeping the package like nexus or local folders on desktop it is called RELEASE , after release we are deploying into QA OR UAT Environments .

**1)IN YOUR ENVIRONMENT WHAT TYPE EC2 INSTANCE YOUR USING ?**

We are using micro and large and xlarge ,

**For what pupose your using micro and large ?**

According to the requirements we have to create ,

this we will give to developement team inside that developement team will take care only creation part we will take care.

**2) WHAT IS THE SPECIFIC REASON TO CREATE EC2 MICRO AND EC2 LARGE ?**

**Specific reason is based on the requirements of ram**

**3)HOW DO YOU DELETE A BRANCH IN YOUR REMOTE REPO ?**

First we have to clone the branch from remote repo after that, **$git delete branch-name** after that **$git commit -m "message"** and **$git push origin/master --delete branch-name**

git push origin --delete branch-name

**4)HOW DO YOU SWITCH BETWEEN BRANCHES IN YOUR LOCAL GIT ?**

We have a command git checkout branch-name

**5)WHAT IS GIT TAG ?**

GIT tag its like if you want to refer the commit directly we can use the git tag whatever the code you commited.

**6)WHEN WILL YOU MERGE YOUR HOTFIX OR YOUR RELEASE BRANCHES TO YOUR MASTER ?**

We will do every monthly release we will merge the code from one branch to another branch.

**7)WHAT IS THE DIFFERENCE BETWEEN JENKINS AND CI/CD PIPELINE ?**

JENKINS , we are creating jobs for the different different requirements like package creation and all but ,

PIPELINE means multiple jobs together we can configure and at a time we can build the all jobs one by one stage by stage and environment by environment we can build the code.

**8)WHAT ARE THE ADVANTAGES OF PIPELINE ?**

-->At a time we can run the build on the different different environments that is one thing and at a time we can pass the multiple requirements into the different different jobs and also we can create the job mutliple jobs for the at a time deployment pupose.

**9) Load Balancing ,**

while we are creating the load balancer we can give the consugative period its like time i will give like 1 hour or 2 hours to check the the load in servers ,

**10)what is the script do ?**

Ususally how my script will work is we are doing the deployment using the rundeck tool so how it will work is in that i will mention the package id and yum install i will mention to install the packages so yum install that package name i should mention in the package name and then location of the script like backend server where i saved that location of that script i will pass from the script and then deploy equal to i will give the that package id and then later if condition i will mention like if deploy = yes then echo it should deploy the packages , sleep mode i will mention 2 minutes to get the packages from the server redhat satelite server so after that install.sh i will run the script so according to this script it will deploy the packages

**11) do you take care of database server ?**

no that and all my developers will take and server setup and all environments team will take care.

**12)daily works when you reach office ?**

Usually my environment is like we are using the service now as ticketing system tool once i reach the office i will the service now like usually we have different different release structures daily releases we have and monthly releases we have according to that what i will do is i will check the ticketing system tool tickets like what and all newly came today so in that if anything related to SVN and Jenkins CI job creation request may be i will get ,

explain about daily activities

**13) What is forking in GIT ?**

Forking means scenario will explain ,

suppose your my developer you checked in some source code but when i try to build the packages some issue in the code so lets say you checked in the source code like pipeline i want to check the groovy file so that time if i do the changes in the groovy file in your code the commit id will be generate new id but dont want to change the commit id i just want to test my groovy stage by stage where exactly the issue there i want to found so that scenario what i will do is just fork the repository into my location machine so that i can configure the jenkins dummy job for the forking repository even if i do the modification in the code also it wont reflect to original repository that scenario we are using git forking

A **fork** is a copy of a repository that allows you to freely experiment with changes without affecting the original project.

**14) Incase If any conflicts happen in your code how do you reslove it ?**

Basically while doing the merging concept in git we are getting the some kind of conflicts like tree conflicts and parallel conflicts also basically what type of conflicts we are getting is if i do the merging suppose if i already committed the same version if i do the existing release branch we have if i do the merge into the existing branch with same code in that scenario we are getting the issues like not able to merge and duplicates will be create so that kind of scenario we need to check the conflicts exacly cd repository name we have go and git status we have to run and there you can get the complite information about the conflict exacly like unmerged path and declared path you will get so according to issue we can remove the existing things like $git rm file-name and then we can do the merge and then i can commit and push this is the steps which we are performing in git resolvation.

for this and all we will go for developers help like commit id's and all

**15) What are the branching strategy in git ?**

we are maintaining the separate branches like **dev branch** and **release branches** and **bug fix branches** so according to the developement we are maintaining the **developer branch** there complete developement will do and every month we have to the releses so once we are doing the build and packaging releases then it will keep into the

-----------------------------------------------------------------------------------------------------------------

git remote add (git-url) -------------> to connect loca repo to remote repo

git push origin --delete branch-name ------> to delete remote repo

netstat -lntp -----------------> to check running ports

ps -au ------------------------> it will display all process

du --------------------> to check disk usage

cat /etc/os-release -----------------> to check os in any machine

git commit --amend ----------> to rename the commit-id

ping -------------------> to check the communication of other servers

git blame file-name --------------> it will display who did modification in same file

**git branch --merged master(branch name)** ----------> how to know what are the branches merged in master

git push origin master1 master2 master3 master4 ------------> to push code into multiple branches at a time

**Suppose if you want to copy file or directory from one location to another location in the same machine ,**

the command is ,

#cp srcpath(/home/ubuntu/trunk) destpath(/home/ubuntu/branch)

#cp -r trunk branch

Examples :

cp venkat venkatesh

cp ven.txt venkatesh

cp manju venkatesh

cp -r venkat venkatesh

cd venkatesh/

**>>Copy the files or directories from one server to another server ,**

the command is ,

scp -r /var/www/file.sql user@456.456:/var/www/

example :

scp -r trunk vagrant@192.168.33.10:/var/lib/

**16)WHAT IS NETSTAT COMMAND IN LINUX ?**

netstat (network statistics) is a command line tool for monitoring network connections both incoming and outgoing as well as viewing routing tables, interface statistics etc. netstat is available on all Unix-like Operating Systems and also available on Windows OS as well.

Listing all ports (both TCP and UDP) using netstat -a option.

**netstate -a | more**

Listing TCP Ports connections ---------> **# netstat -at**

Listing only UDP (User Datagram Protocol ) port connections using --------> **netstat -au**

Listing all LISTENING Connections ---------------------> **netstat -l**

Listing all TCP Listening Ports ----------------> **netstat -lt**

Listing all UDP Listening Ports ---------------> **netstat -lu**

**17)How can you connect to other devices within Ansible?**

#ansible all -m ping or #ansible -m ping all

**18) Multi branch pipeline ?**

If you want to trigger the build with multiple branches then you have to create multiple job configuration,

inside that we will get ADD SOURCE so click on that and select VCS like svn or git and pass the ip address and save the configuration trigger the build.

**19)HOW TO PUSH PARTICULAR BRANCH INTO REMOTE REPOSITORY :**

#git push origin master1 -------------> from master1 branch only we have to push master1 into remote repo

**20) Difference between authentication and authorization?**

**Authentication:**

To give credentials to jenkins

**Authorization:**

To give permissions to the services

**21) ADVANTAGES OF ANSIBLE ?**

--> Ansible is a agent less configuration management tool and it is a Orchestration and provisioning ,

--> Ansible is a open source configuration tool

--> We can do the deployment on multiple servers using ansible

--> Playbooks are required for ansible