**Interview Questions:**

Ansible:

1. What you have done in ansible ?
2. Difference between jenkins and Ansible

|  |  |  |
| --- | --- | --- |
|  | **Ansible/Ansible Tower** | **Jenkins** |
| Purpose | Configuration management, application deployment, and infrastructure automation | Automation of Build and Release process and Orchestration of pipelines |

1. Ansible handler

Handlers are just like normal [tasks](https://docs.ansible.com/ansible/latest/user_guide/playbooks_intro.html#tasks-list) in an Ansible playbook but they run only when if the Task contains a “notify” directive. It also indicates that it changed something.

Regardless of how many tasks notify a handler, it will run only once, after all of the tasks completed in a particular play.

1. Restart the service when it is not active

Can define when condition

1. Download from internet using ansible

get\_url module

* name: Download Tomcat using get\_url  
  become: yes  
  get\_url:  
   **url:** [**https://www-us.apache.org/dist/tomcat/tomcat-8/v8.5.40/bin/apache-tomcat-8.5.40.tar.gz**](https://www-us.apache.org/dist/tomcat/tomcat-8/v8.5.40/bin/apache-tomcat-8.5.40.tar.gz)  
   dest: /opt/tomcat8  
   mode: 0755  
   **checksum: sha512:5bdea5414713c9ba39e226f062701fa14998b1a798c9750f956a0f59b5edabb8d83af9ec9f81cf9f47fa92c21b560c9b2be1b543d0bd8f1b49579b69101d3a8f**  
   group: tomcat  
   owner: tomcat

1. Ansible playbook to start ngnix

/etc/ansible/hosts

[test-server]

172.17.0.2

File.yml

---

hosts: test-server

sudo: yes

vars:

- server\_port: 8080

tasks:

- name: install nginx

yum:

name: nginx

started: installed

- name: serve nginx

template: src=../files/flask.conf dest=/etc/nginx/conf.d

notify:

- restart nginx

handlers:

- name: restart nginx

service: name=nginx state= restarted

Jenkins:

1. Jenkins source code- How to create docker image from Jenkins and where do you store the docker image.

<https://karthi-net.medium.com/docker-tutorial-build-docker-images-using-jenkins-d2880e65b74>

1. Jenkins architecture
2. Make file
3. Jenkins pipeline brief (maven steps)
4. Plugins installed in jenkins
5. Integration of sonarqube with jenkins
6. Integration of Artifactory with jenkins.
7. Jenkins\_home. - /var/lib/jenkins
8. Jenkins\_home what all it contains. - (config.xml, nodes(slaves configuration), workspace[JOBNAME], plugins,jobs[JOBNAME,config.xml,latest,builds(build.xml,log,changelog.xml)]
9. jenkins scripted pipeline
10. Jenkins home directory can be changed in the "/etc/sysconfig/jenkins"

Text

Description automatically generated

1. Install jenkins using war - java -jar jenkins.war. After that /var/lib/jenkins directory will be created

Graphical user interface, text, application

Description automatically generated

1. To check Jenkins Versions:

java -jar jenkins-cli.jar -s <http://localhost:8080/> version

1. Gracefull restart

java -jar jenkins-cli.jar -s <http://localhost:8080/> restart

1. To upgrade jenkins, replace jenkins.war in the "/usr/lib/jenkins"
2. **Post Upgrade Activates.**

We might not need all the following activities need to be done but do check all the listed activities after the upgrade of Jenkins.

**Update all the installed plugins.** To do so, Go to Manage Jenkins, then, click Manage Plugins, then, click “updates” Tab. Then Select plugins that need to be upgraded and then click “Download now and install after restart” button.

**Check the Connection to the Jenkins Slaves.**To check this, Go to *Manage Jenkins*, Then Click *Manage Nodes* then check the slaves. If any of your slaves is offline, Then Make it online.

**Examine Agent to Master Security subsystem.**For this, Go to Manage Jenkins, Then Click *Configure global security* then, go to *Plugin Manager* Section and check in the “*Enable Slave -> Master Access Control*”.

**Manage Old Data.**Sometimes, The Format of data stored inside Jenkins will vary for certain Jenkins versions. This needs to be properly addressed. To do so, click the *Manage Jenkins.*Then, click *Manage Old Data*then, check for the old data. If it is available, then you can always change it back to when you Downgrade your Jenkins with the backup war file, we have taken in Step 5. If you do not have any problem after the Upgrade, no need to change the version here.

These are the major activities that need to be verified right after upgrading Jenkins. Once after verifying all the above activities, you can say that you upgrade process is successful.

1. Jenkins Access control:

Role-based Authorization Strategy

Manage Role

Global Role

project Role

Assign Role

Global Role

Project Role

Patterns are case-sensitive. To perform a case-insensitive match, use (?i) notation: upper, Roger-.\* vs. lower, roger-.\* vs. case-insensitive, (?i)roger-.\*.

1. How to run the build steps parallely in jenkins

1. Diff between WEBHOOK trigger and Poll SCM

Webhook source code wiil send the info but in poll scm jenkins will always check for the changes

Webhook api : It trigger http request (host url/generic-webhook-trigger/invoke)

1. **Execute the same job in parallel ?**

* **"Execute concurrent builds if necesarry"**

1. **Multiple repo addition - pipelinescript from SCM**
2. Scripted and declarative

**Built in environment variables**

Jenkins provides a set of environment variables. You can also define your own. Here is a list of built in environment variables:

* **BUILD\_NUMBER** - The current build number. For example "153"
* **BUILD\_ID** - The current build id. For example "2018-08-22\_23-59-59"
* **BUILD\_DISPLAY\_NAME** - The name of the current build. For example "#153".
* **JOB\_NAME** - Name of the project of this build. For example "foo"
* **BUILD\_TAG** - String of "jenkins-${JOB\_NAME}-${BUILD\_NUMBER}".
* **EXECUTOR\_NUMBER** - The unique number that identifies the current executor.
* **NODE\_NAME** - Name of the "slave" or "master". For example "linux".
* **NODE\_LABELS** - Whitespace-separated list of labels that the node is assigned.
* **WORKSPACE** - Absolute path of the build as a workspace.
* **JENKINS\_HOME** - Absolute path on the master node for Jenkins to store data.
* **JENKINS\_URL** - URL of Jenkins. For example [*http://server:port/jenkins/*](http://server:port/jenkins/)
* **BUILD\_URL** - Full URL of this build. For example [*http://server:port/jenkins/job/foo/15/*](http://server:port/jenkins/job/foo/15/)
* **JOB\_URL** - Full URL of this job. For example [*http://server:port/jenkins/job/foo/*](http://server:port/jenkins/job/foo/)

1. **any**– Which mean the whole pipeline will run on any available agent.

* **none**– Which mean all the stages under the block will have to declared with agent separately.
* **label**–  this is just a label for the Jenkins environment
* **docker**–  this is to run the pipeline in Docker environment.

1. Paramaters in jenkins
2. Rebuild and replay
3. Multiple branch pipeline way of working
4. Scripted pipeline
5. Parallel stage
6. Slave configuration
7. Jenkins saferestart
8. Artifactory retention policy

retentionPolicy:

    enabled: true

    maxAgeDays: 90

    minRuns: 10

1. Flow of pipeline
2. Cron that runs every Monday and Wednesday at 10pm

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

**Docker:**

1. How to version the docker images.

We use tags for versioning docker images

1. Type of networkn in docker (bridge, host, overlay, none)
2. Docker image and container difference

Docker Image is a set of files which has no state, whereas [Docker Container](https://go4hosting.in/knowledgebase/docker/what-is-docker-container) is the instantiation of Docker Image. In other words, Docker Container is the run time instance of images.

1. EntryPoint and CMD

This is because you cannot override ENTRYPOINT instructions, whereas with CMD you can easily do so.

1. COPY and ADD

**COPY** copies a file/directory from your host to your image.

**ADD** copies a file/directory from your host to your image, but can also fetch remote URLs, extract TAR files, etc...

Use **COPY** for simply copying files and/or directories into the build context.

Use **ADD** for downloading remote resources, extracting TAR files, etc..

1. Different between docker and ansible

Docker container is implemented with host OS software including process, chroot, cgroup, network and so on to utilize independent environment directly on host OS.

On the other hand, [Ansible](https://intellipaat.com/blog/what-is-ansible/) is a configuration management tool.It provides an application for deployment, installation, and configuration of a server. You write a configuration file for those and run it to deploy, install and configure things on servers. Using Ansible tool, you do not have to run the installation and configuration programs to the servers one-by-one. This tool just manages to automate installation and configuration to all the servers.

[**Ansible**](https://www.ansible.com/): It is an open-source, IT automation engine system. This server and configuration management tool, which is supported by Red Hat, makes IT automation simple as it ends repetitive tasks and enables faster application deployments, thus allows DevOps teams to perform more strategic work. It automates configuration management, orchestration, application deployment, cloud provisioning, and a number of other IT requirements. It further allows users to control multi-tier complex deployment and security management. Companies like Tokopedia, Revolut, Trivago use Ansible.

[**Docker**](https://www.docker.com/): It is a software container technology platform that enables its users to create, deploy, run, and manage applications within the containers. Its modular design enables users to build applications securely, both on-premises and in the cloud. Additionally, it uses a number of the Linux kernel’s features such as namespaces, cgroups, AppArmor profiles, and more, to sandbox processes into existing configurable virtual environments. Also, due to the least compatibility issues, applications can run wherever one wants without causing compatibility hurdles. Companies like Twitter, Spotify, Pinterest, PayPal, Vox media use Docker.

[**Kubernetes**](https://kubernetes.io/): It is an open-source system that provides mechanisms to deploy, maintain, and scale containerized applications with automation. Designed by Google and is currently supported by the Cloud Native Computing Foundation (CNCF). This multi-layered tool implements the Infrastructure as Code principle of DevOps, which allows independent management of each infrastructure layer, i.e., from a single container to pods, nodes, namespaces, and clusters, along with networking and physical hosts. It fulfills customers’ demands by deploying applications predictably and quickly, scaling them, launching new features, and limiting hardware usage to only the needed resources. Companies like Google, StackShare, Slack use Kubernetes.

**Kubernetes:**

1. What you have done in Kubernetes from the start ?
2. Services in kubernetes
3. Encrypt traffic between container
4. Kubernetes architec
5. Namespace
6. Kubernetes architecture
7. Kubernetes servcies
8. Ingress
9. Kubelet
10. Statefull and stateless
11. Demon set
12. Manifest file stored: /etc/kubernetes/manifest
13. Kubectl log -c <containername> - to check log of particular contianer log
14. Kubectl create deployment -n name --image:nginx --replicas=2
15. Network kubernetes
16. Health check of a pod - liveliness and Readiness
17. Deployment Strategy - Rolling upate and Recreat
18. Kube proxy -
19. Demon set
20. Config map and secret
21. Namespace , POD, container
22. RBAC
23. How to restrict communication between container in same pod

1. How to do trouble shoot the load of an application in the container

Text

Description automatically generated

1. To check the node with which os

Kubectl get nodes -o wide

or

Kubectl get nodes -o wide --no-headers |wc -

Kubectl get nodes --show labels |grep -iwindows

1. NodeSelector only label:

Pod creation manifest file,

apiversion: v1

Kind: Pod

metadata:

name: nginx

namespace: default

spec:

containers:

- name: nginx

image: artifactory.wdc.com/library/nginx:latest

ports:

- containerPort: 80

nodeSelectors: kubernetes.io/os=windows

1. To assign label

Kubectl label nodes uls-od-sdfk13.wd.com foo-bar

To see more about node

Kubectl describe node uls-od-sdfk13.wd.com



**Shell/Bash:**

2. Script to transfer file from source machine to multiple serves

pscp command to transfer

for HOST in server1 server2 server3; do  
 scp somefile $HOST:~/somedir/  
 Done

1. Bash script/python to write for loop

**for** VARIABLE **in** 1 2 3 4 5 .. N  
**do**  
        command1  
        command2  
        commandN  
**Done**

**Python:**

fruits = ["apple", "banana", "cherry"]

for x in fruits:

  print(x)

1. **What is $ and # in linux -** $ at the end of the terminal prompt indicates that you are now as regular user. It has other meaning in shell scripting, it indicates a variable. And for regular expressions, we use $ as end-of-string anchor.
2. **user@machine$ ~/data**
3. Compare 2 number

a = 10  
b = 20

print(a == b)

a = input("Enter the first number: ")  
b = input("Enter the second number: ")  
# if a is b: - Compares id's, and can cause inconsistencies. Use == instead.  
if a == b:  
 print "Both inputs are equal"  
else:  
 print "Your input is not equal."

1. Compare 2 string

**#!/bin/bash**

**str1="Hello Bash"**

str2=**"Hello Bash"**

if [ "$str1" == "$str2" ]; then

**echo "Strings are equal"**

else

**echo "Strings are not equal"**

Fi

str1 **=** "Geek"

str2 **=** "Geek"

if string1 == string2 :  
 print("Strings are equal with text : ", string1," & " ,string2)  
else :  
 print ("Strings are not equal")

Print 80th line from text file

Print interger in reverse order

Shell :

n=123465

sd=0

rev=0

while [ $n -gt 0 ]

do

sd=$(( $n % 10 ))

rev=$(( $rev \* 10 + $sd ))

n=$(( $n / 10 ))

done

echo "Reverse number of entered digit is $rev"

Python:

n **= 4567**

rev **=** 0

**while**(n > 0):

    a **=** n **%** 10

    rev **=** rev **\*** 10 **+** a

    n **=** n **//** 10

print(rev)

**Python :**

1. How can looping be done over a list of hosts in a group, inside of a template?

{% for host in groups['db\_servers'] %} {{ hostvars[host]['ansible\_eth0']['ipv4']['address'] }} {% endfor %}.

1. **Print last word in a sentence. Str1 = "Ram ate apple" str2=str1.split() print(str2[-1])**
2. Python

\_\_name\_\_ = \_\_main\_\_

If this is called the python file and it we run this script all functions defined in this script will execyte

Suppose if we use this script as a module in somther script at that time in the import if we mention only the function which is requried it will be called

1. Read log file and print only the error lines

Def sum(a, b):

c= a + b

return c

Print(sum(1,2))

**Git:**

1. Git merge and git rebase diff
2. Git cherrypick
3. Merge conflict
4. Git fetch and pull
5. **Branching Strategies (advantages/Disadvantages)**

**SonarQube:**

1. Sonarqube. (quality gate and quality profile diff ). Code coverag
2. **How to create reports in SonarQube?**

**A:** To create reports using SonarQube

mvn clean install

mvn sonar:sonar -Dsonar.issuesreport.html.enable=true

1. **Why use SonarQube ?**

**A:**Sonar covers the 7 sections of code quality

* Architecture and Design
* Unit tests
* Duplicated code
* Potential bugs
* Complex code
* Coding standards
* Comments

Maven :

**How Settings.xml different from pom.xml?**

Ans: settings.xml is your user preferences. It lives in your main Maven directory (usually $HOME/.m2) and holds your own settings, like listings for non-public repositories, usernames, and other personalized configuration.

pom.xml is the control file for each Maven project or module. It tells Maven which dependencies the project needs, what processing to apply to build it, and how to package it when it's ready. The POM is part of the project itself, and so information that's necessary to build the project (such as listing which plug-ins to use when building) should go there.

**Maven**: It is a kind of application that provides a lot of functionalities for effective project management. It performs the functions of documentation, building, and reporting.

**MAVEN** is a built automation tool. It describes how the software is developed and its dependencies. It helps in executing unit tests as a part of the normal established cycle and supports projects written in Ruby, C#

**The Maven’s order of inheritance has four different things or components such as**

* Parent Pom
* Project Pom
* Settings
* CLI parameters

**Jar** requires Java installation. It contains class files, resources like .java, and property files. They are to be appended to CLASSPATH- an environment variable to any java application to access from the remote package.

**Ear** requires a full Java platform. The applications of the enterprise that are to be deployed in EJB containers are placed within the .ear file.

**War** requires a web.xml file stored within a WEB-INF file. These files have a .war extension. The web application to be deployed on a [JSP container](https://docs.oracle.com/cd/A87860_01/doc/java.817/a83726/genlovw2.htm#:~:text=A%20JSP%20container%20is%20an,executed%20by%20a%20servlet%20container.)or the servlet is too converted into .war files and is developed using the TOMCAT browser. It contains many important files required for web applications such as HTML, .js, .jsp.

38) Explain the procedure of deployment?

Ans:

**PLAN**- Assemble a team and study the environment. Design the architecture and zone structure.

**PREPARE**- Configure an active directory and install the Centrify Suite.

**DEPLOY**- Download the software and install agents. Join a certify zone.

**VALIDATE**- Test the windows login role and verify the application rights and desktop. Check all the audited sessions.

**MANAGE**- Add custom roles and delegate administrative tasks. Add group policies and agents.

**How can I change the default location of the generated jar when I command "mvn package"?**

By default, the location of the generated jar is in ${project.build.directory} or in your target directory. We can change this by configuring the outputDirectory of maven-jar-plugin.

**How do I determine which POM contains missing transitive dependency?**

run mvn -X

**What is the difference between compile and install?**

Compile compiles the source code of the project

whereas

Install installs the package into the local repository, for use as a dependency in other projects locally

Bamboo:

What are the benefits of running automated tests in Bamboo?

Ans: It will make finding and correctcting the errors fast and help users saving their valuable time.

A Plan in Bamboo defines a sequence of tasks for Bamboo to perform. When a plan is triggered, Bamboo executes the defined tasks sequentially. It also provides options to define final tasks. The steps below show an example of configuring a plan. Your build plan may look different from this example.

Graphical user interface, application, Teams

Description automatically generated

Task runs sequentially inside the job

Graphical user interface, text, application

Description automatically generated

Jobs runs parellely

Jobs within stages runs parallel if enough build agents are available

Bamboo will wait till all jobs are compeleted with one stage before moving to the next stage

Graphical user interface, text, application

Description automatically generated

**Database :**

**What are Tables and Fields?**

A table is an organized collection of data stored in the form of rows and columns. Columns can be categorized as vertical and rows as horizontal. The columns in a table are called fields while the rows can be referred to as records.

**What are Constraints in SQL?**

Constraints are used to specify the rules concerning data in the table. It can be applied for single or multiple fields in an SQL table during creation of table or after creationg using the ALTER TABLE command. The constraints are:

NOT NULL - Restricts NULL value from being inserted into a column.

CHECK - Verifies that all values in a field satisfy a condition.

DEFAULT - Automatically assigns a default value if no value has been specified for the field.

UNIQUE - Ensures unique values to be inserted into the field.

INDEX - Indexes a field providing faster retrieval of records.

PRIMARY KEY - Uniquely identifies each record in a table.

FOREIGN KEY - Ensures referential integrity for a record in another table.