**Jenkins -**

* + **Jenkins installation ways with explanation**

Jenkins as a service –

Download Jenkins for appropriate OS and install as a normal software and run the localhost:8080 on any Browser

Secret admin password copy from secretPassword and pest it at first time while Jenkins is startes

8080 is default port.

Jenkins on top of Tomcat:

Download Jenkins.war file from Jenkins site.

Copy Jenkins.war into apache webapps folder

Go to apache/bin and run startup.sh/bat file

Goto browser localhost:8080/Jenkins

Jenkins is fully up and running.

Admin password is find on apache running screen or in secrete file also u can find path

Temporarily run Jenkins –

Goto Start-cmd-path where Jenkins.war is available and

Run command java -jar Jenkins.war

If port no is need to change then run

java -jar Jenkins.war --httpPort=8085

* + **How to secure Jenkins**
  + **Plugin installation**

Manage Jenkins- manage plugins- Search for required plugin and install it.

* + **Mostly used plugins**
  + **How to restart Jenkins when jenkins as a service and on top of tomcat**
  + **How to create job in Jenkins – all tabs description**
  + **Manage Jenkins**
  + **Matrix based authorization**
  + **Which scripting used for Jenkins deployment automation**
  + **How to enable/disable security**
  + **How to login if password forgot**
  + **How to login without password**
  + **Database used in Jenkins**
  + **Authorization securities**
  + **CI/CD pipeline/ Build pipeline**
  + **Cron tab**
  + **Pipelining**
  + **what is syntax if you want to install jenkins on different port**
    - **java -jar jenkins.war --httpPort=8088**

1. **What is ANT.**

ANT (Another Neat Tool)is a Build Tool based on Java.

Tasks –

Compiling java code into byte code

Place this byte code into pkg

Creation of .jar/war/ear files.

Deployment to production env

1. **ANT Functionality**

Open source under Apache so that we can download and update source code.

Can be integrated with any editor or development env easily.

Ant uses XML build files which make its development easy.

1. **Explain ANT with example from start.**

**Steps**

* 1. Download apache ANT
  2. Set **JAVA\_HOME, ANT\_HOME** and edit **PATH** by appending **%JAVA\_HOME%/bin**; and **%ANT\_HOME%/bin**
  3. Write “Hello world” build.xml

<project name=”HelloWorld” default=”compiler” basedir=”.”>

<target name=”compiler”>

<mkdir dir = “Helloworldclassfiles”>

<javac srcdir=”Dirhelloworld” destdir=”Helloworldclassfiles”>

</target>

</project>

* + **If Build.xml is not available in dir and is saved with different name**
  + **Tags**
  + **Build.xml file structure**

<?xml version="1.0"?>

<project name="Ant-Test" default="main" basedir=".">

*<!-- Sets variables which can later be used. -->*

*<!-- The value of a property is accessed via ${} -->*

<property name="src.dir" location="src" />

<property name="build.dir" location="bin" />

<property name="dist.dir" location="dist" />

<property name="docs.dir" location="docs" />

*<!-- Deletes the existing build, docs and dist directory-->*

<target name="clean">

<delete dir="${build.dir}" />

<delete dir="${docs.dir}" />

<delete dir="${dist.dir}" />

</target>

*<!-- Creates the build, docs and dist directory-->*

<target name="makedir">

<mkdir dir="${build.dir}" />

<mkdir dir="${docs.dir}" />

<mkdir dir="${dist.dir}" />

</target>

*<!-- Compiles the java code (including the usage of library for JUnit -->*

<target name="compile" depends="clean, makedir">

<javac srcdir="${src.dir}" destdir="${build.dir}">

</javac>

</target>

*<!-- Creates Javadoc -->*

<target name="docs" depends="compile">

<javadoc packagenames="src" sourcepath="${src.dir}" destdir="${docs.dir}">

*<!-- Define which files / directory should get included, we include all -->*

<fileset dir="${src.dir}">

<include name="\*\*" />

</fileset>

</javadoc>

</target>

*<!--Creates the deployable jar file -->*

<target name="jar" depends="compile">

<jar destfile="${dist.dir}\de.vogella.build.test.ant.jar" basedir="${build.dir}">

<manifest>

<attribute name="Main-Class" value="test.Main" />

</manifest>

</jar>

</target>

<target name="main" depends="compile, jar, docs">

<description>Main target</description>

</target>

</project>

* + - **Project tag**
    - **Target**
    - **Depends**
    - **Task**
    - **Property file etc**
  + **How to run build.xml**
  + **How to integrate ant with Jenkins**
  + **Troubleshooting of build errors in Jenkins**

1. **Difference between ant and maven**
2. **Maven basics**

**What is POM.xml**

**What can maven does**

**POM.xml file structure**