JAVA means DURGA SOFT

JAVA FAQ`S

Most important Questions

IN

Weblogic Application Server

DURGA M.Tech

(Sun certified & Realtime Expert)

Ex. IBM Employee

**Trained Lakhs of Students
for last 14 years across INDIA**

India's No.1 Software Training Institute

DURGASOFT

www.durgasoft.com Ph: 9246212143 ,8096969696

Weblogic Application Server FAQs

1.What is application server?

- An application server is a software framework dedicated to the efficient execution of procedures (scripts, routines, programs ...) for supporting the construction of applications. The term was created in the context of web applications. In these, the application server acts as a set of components accessible to the software developer through an API defined by the platform itself. These components are usually performed in the same machine where the web server is running, and their main job is to support the construction of dynamic pages.
- Other uses of the term can refer to:
    - o the services that a server makes available
    - o the computer hardware on which the services run
- **Java application servers**
- Following the success of the Java platform, the term application server sometimes refers to a J2EE or Java EE 5 application server. Some of the better-known Java Enterprise Edition application servers include:
    - o Apache Tomcat (Apache Software Foundation)
    - o Tcat Server (MuleSoft)
    - o WebSphere Application Server and WebSphere Application Server Community Edition (IBM)
    - o Sybase Enterprise Application Server (Sybase Inc)
    - o WebLogic Server (Oracle)
    - o JBoss (Red Hat)
    - o Apache Geronimo (Apache Software Foundation)
    - o Oracle OC4J (Oracle)
    - o SAP Netweaver AS (ABAP/Java) (SAP)
    - o WebObjects (Apple Inc.)
- The web modules include servlets and JavaServer Pages. Business logic resides in Enterprise JavaBeans (EJB-3 and later). The Hibernate project offers an EJB-3 container implementation for the JBoss application server. Tomcat from Apache and JOnAS from ObjectWeb exemplify typical containers which can store these modules.
- A Java Server Page (JSP) (a servlet from Java) executes in a web container — the Java equivalent of CGI scripts. JSPs provide a way to create HTML pages by embedding references to the server logic within the page. HTML coders and Java programmers can work side by side by referencing each other's code from within their own. JavaBeans are the independent class components of the Java architecture from Sun Microsystems.

2.What is web server?

- The primary function of a web server is to deliver web pages (HTML documents) and associated content (e.g. images, style sheets, JavaScript's) to clients. A client, commonly a web browser or web crawler, makes a request for a specific resource using HTTP and, if all goes well, the server responds with the content of that resource. The resource is typically a real file on the server's secondary memory, but this is not necessarily the case and depends on how the web server is implemented.
- While the primary function is to serve content, a full implementation of HTTP also includes a way of receiving content from clients. This feature is used for submitting web forms, including uploading of files.
- Many generic web servers also support server-side scripting (e.g. Apache HTTP Server and PHP). This means that the behavior of the web server can be scripted in separate files, while the actual server software remains unchanged. Usually, this functionality is used to create HTML documents on-the-fly as opposed to return fixed documents. This is referred to as dynamic and static content respectively.
- **History of web servers**
- In 1989 Tim Berners-Lee proposed to his employer CERN (European Organization for Nuclear Research) a new project, which had the goal of easing the exchange of information between scientists by using a hypertext system. As a result of the implementation of this project, in 1990 Berners-Lee wrote two programs:
  - o a browser called Worldwide Web;
  - o the world's first web server, later known as CERN httpd, which ran on NeXTSTEP.
- Between 1991 and 1994 the simplicity and effectiveness of early technologies used to surf and exchange data through the World Wide Web helped to port them to many different operating systems and spread their use among lots of different social groups of people, first in scientific organizations, then in universities and finally in industry.
- In 1994 Tim Berners-Lee decided to constitute the World Wide Web Consortium to regulate the further development of the many technologies involved (HTTP, HTML, etc.) through a standardization process.
- **Common features**
  - o Virtual hosting to serve many web sites using one IP address.
  - o Large file support to be able to serve files whose size is greater than 2 GB on 32 bit OS.
  - o Bandwidth throttling to limit the speed of responses in order to not saturate the network and to be able to serve more clients.

3.What is the difference between Web server and Application Server?
Application Server

- Webserver serves pages for viewing in web browser, application server provides exposes business logic for client applications through various protocols
- Webserver exclusively handles http requests. Application server serves business logic to application programs through any number of protocols.
- Webserver delegation model is fairly simple, when the request comes into the webserver, it simply passes the request to the program best able to handle it(Server side program). It may not support transactions and database connection pooling.
- Application server is more capable of dynamic behavior than webserver. We can also configure application server to work as a webserver. Simply application server is a superset of webserver.

WEB Server

- Web Server serves static HTML pages or gifs, jpegs, etc., and can also run code written in CGI, JSP etc. A Web server handles the HTTP protocol. Eg of some web server are IIS or apache.
- An Application Server is used to run business logic or dynamically generated presentation code. It can either be .NET based or J2EE based (BEA WebLogic Server, IBM WebSphere, JBoss).
- A J2EE application server runs servlets and JSPs (infact a part of the app server called web container is responsible for running servlets and JSPs) that are used to create HTML pages dynamically. In addition, J2EE application server can run EJBs - which are used to execute business logic.
- An Application server has a 'built-in' web server; in addition to that it supports other modules or features like e-business integration, independent management and security module, portlets etc.

4.What is the Domain in Weblogic serve?

- o Domain is a logically related group of Oracle WebLogic Server resources that are managed as a single unit
- o Domain Provides one point of administration
- o Can logically separate:
  - Development, test, and production applications
  - Organizational divisions

5.What are the Domain Restrictions?

- Each domain requires its own Administration Server.
- A cluster cannot span multiple domains.
- The Managed Servers in a domain must run the same version of Oracle WebLogic Server.
- The Administration Server in a domain must run the same or higher version as Managed Servers in the domain.

6.What is the server?

- A server is an instance of weblogic.Server executing in a Java Virtual Machine (JVM).
- A server:
  - Runs on a designated Oracle WebLogic Server machine
  - Has a dedicated amount of RAM
  - Is multithreaded
- Two types of servers:
  - Administration Server
  - Managed Server

7.What is the Administration server?
Administration server is central point of control for a domain. It stores the configuration information and logs for a domain. And it runs the Weblogic Administration console.

8.What is the Managed Server?
Managed server is a server in a domain that is not the Administration server. It contacts the administration server for configuration information. It runs business application in a production environment. It is independent of all other Managed servers in a domain (unless they are not in a cluster). You can have many managed servers in a domain. Individual managed servers are typically added for capacity and application isolation.

9.How Administration server and Managed servers will interact?

- The Administration Server stores the master copy of the domain configuration, including the configuration for all Managed Servers in the domain.
- Each Managed Server stores a local copy of the domain configuration file.
- When a Managed Server starts, it connects to the Administration Server to synchronize the configuration.
- When the configuration is changed, the Administration Server sends the changed configuration to the Managed Servers.

10.What is a Machine in Oracle Weblogic Server?
A Machine in a Oracle Weblogic server is a computer that hosts the Oracle Weblogic Server instances. Runs a supported operating system platform and it is used by Node Manager to restart a failed Managed servers.

11.What is a Cluster in Oracle Weblogic server?
A cluster is a Logical group of Weblogic servers. Oracle Weblogic server provides HIGH AVAILABILITY & LOAD BALANCING with help of cluster.

12.What is a Node Manager?

A Node Manager is a utility or process running on a physical server that enables starting, stopping, suspending or restarting the Administration and Managed servers remotely. It is not associated with a Domain. [Node manager can start any server instances that are resides on the same physical server].

13.How many ways we can install Oracle Weblogic Server?
You can install Oracle Weblogic server in three different ways.

- GUI mode (extract server103_linux32.bin/double click on server103_win32.exe)
- Console mode (c:\>server103_linux32.bin –mode=console –log=c:\consoleinstal.log)
- Silent mode (c:\> server103_linux32.bin –mode=silent –silent_xml=path_to_silent.xml –log=c:\silentinstal.log)

14.What is the default database for Weblogic server?
Point base is the default database. This database comes along with Weblogic software bundle.

15.How many ways we can configure a domain?
You can configure Oracle Weblogic server domains in two ways.

- Graphical Mode (config.cmd [Scripts are in the <WL_HOME>/common/bin directory])
- Console Mode (config.cmd –mode=console)

16.How many ways you can start Administration server?
You can start the Administration server using 5 ways.

- o weblogic.Server (only in development)
- o Start menu (only Windows)
- o DOMAIN_DIR/bin/startWebLogic.sh
- o WebLogic Scripting Tool (WLST) and Node Manager
- o WLST without Node Manager

17.How many ways you can configure managed servers?
You can configure the Managed server using 3 was.

- o Domain Configuration Wizard
- o Administration Console
- o Command Line (WLST)

18.How many ways you can configure a machines?
You can configure machines by using following:

- o Domain Configuration Wizard
- o Administration Console
- o Command Line (WLST)

19.How many ways you can start managed servers?
Start Managed Servers by using:

- o weblogic.Server
- o DOMAIN_DIR/bin/startManagedWebLogic.sh
- o Administration Console
- o WLST and Node Manager

20.How do I provide user credentials for starting a server?
When you create a domain, the Configuration Wizard prompts you to provide the username and password for an initial administrative user. If you create the domain in development mode, the wizard saves the username and encrypted password in a boot identity file. A WebLogic Server instance can refer to a boot identity file during its startup process. If a server instance does not find such a file, it prompts you to enter credentials.

If you create a domain in production mode, or if you want to change user credentials in an existing boot identity file, you can create a new boot identity file. WebLogic Server does not support copying a boot identity file from one server root directory to another. For information on creating and using boot identity files, see Boot Identity Files in Administration Console Online Help.

21.Can I start a Managed Server if the Administration Server is unavailable?
By default, if a Managed Server is unable to connect to the specified Administration Server during startup, it can retrieve its configuration by reading a configuration file and other files directly. You cannot change the server's configuration until the Administration Server is available. A Managed Server that starts in this way is running in Managed Server Independence mode. For more information, see Starting a Managed Server When the Administration Server Is Not Accessible in Configuring and Managing WebLogic Server.

22.What is the function of T3 in WebLogic Server?

T3 provides a framework for WebLogic Server messages that support for enhancements. These enhancements include abbreviations and features, such as object replacement, that work in the context of WebLogic Server clusters and HTTP and other product tunneling. T3 predates Java Object Serialization and RMI, while closely tracking and leveraging these specifications. T3 is a superset of Java Object. Serialization or RMI; anything you can do in Java Object Serialization and RMI can be done over T3. T3 is mandated between WebLogic Servers and between programmatic clients and a WebLogic Server cluster. HTTP and IIOP are optional protocols that can be used to communicate between other processes and WebLogic Server. It depends on what you want to do. For example: when you want to communicate between a browser and WebLogic Server -- use HTTP, or an ORB and WebLogic Server-IIOP.

23.What is the easiest way to set the classpath?

WebLogic Server installs the following script that you can use to set the classpath that a server requires:

WL_HOME\server\bin\setWLSEnv.cmd (on Windows)
WL_HOME/server/bin/setWLSEnv.sh (on UNIX)

Where WL_HOME is the directory in which you installed WebLogic Server. For more information, see "Setting the Classpath" in the WebLogic Server Command Reference.

24.How do I edit the config.xml file?

The persistent configuration for a domain of WebLogic Servers and clusters is stored in an XML configuration file (config.xml). You can modify this file in the following ways:

- Using the Administration Console.
- If you want to create scripts that automate domain management, use the weblogic.Admin utility. See "weblogic.Admin Command-Line Reference".
- If you want to create Java-based management applications, use the Java Management Extensions (JMX) Application Programming Interface (API). See the Programming WebLogic Management Services with JMX guide.
- If you want to edit the config.xml file directly (not recommended), see the BEA WebLogic Server Configuration Reference.

25.Is there a quick way to create and start a remote Managed Server?

The recommended approach is to use the Domain Configuration Wizard, as described in "Setting Up and Starting Managed Servers on a Remote Machine" in Creating WebLogic Configurations Using the Configuration Wizard at
http://download.oracle.com/docs/cd/E13196_01/platform/docs81/confgwiz/multi.html.
For a streamlined approach, follow the instructions at "Starting Managed Servers From a WebLogic Server Script" in the Administration Console Online Help.

26. The Tree View pane of the WebLogic Console is not visible in my browser. How do I enable it?

- Enable the Sun Java Plug-In from the control panel.

27. What is the importance of the Boot Identity file and how will you create it?

- o If you create boot identity file, it will not ask the user name and password at server startup time.
- o Create a file called boot.properties in the <DOMAIN_HOME>\servers\<server_name>\security directory that contains two lines:
  - ▪ username=username
  - ▪ password=password
- o The first time you start the server; the server reads the Boot Identity file and overwrites it with an encrypted version of the username and password.
- o Thereafter, the server remembers the credentials for subsequent startup cycles.

28. What is the MSI mode in Weblogic? How can you enable and disable this option?

- o MSI is nothing but Managed Server Independence.
- o By default, Managed Servers can function independently of the Administration Server.
- o A Managed Server instance can start in MSI mode if the Administration Server is unavailable.
- o Configure MSI mode from the Administration Console.
- o To start a Managed Server in MSI mode, perform the following:
  - ▪ Ensure that the Managed Server's root directory contains the config subdirectory.
  - ▪ If the config subdirectory does not exist, copy it from the Administration Server's root directory.
  - ▪ Start the Managed Server at the command line or by using a script.
- o Environment > Servers > Server_Name > Tuning > Advanced > Managed Server Independence Enabled check box.

29. If the Administration server not available while starting the Managed server which is already enabled MSI, what are the files it will look for?

- o If the Administration Server is unavailable at boot time, Managed Servers search for:
  - ▪ config.xml
  - ▪ SerializedSystemIni.dat
  - ▪ boot.properties(optional)
- o Each Managed Server looks in its local config directory for config.xml, a replica of the domain's config.xml.
- o You cannot change the configuration of the Managed Server that is running in MSI mode until it restores communication with the Administration Server.

30. What if Administration server goes down? What is the behavior of the managed servers? What are all the things will available or not available?

- o The Administration Server:
  - ▪ Can go down without affecting the operation of the Managed Servers

▪ Can be restarted when the Managed Servers are still running
o When an Administration Server goes down:
  ▪ The domain log entries are unavailable while it is down
  ▪ Managed Servers can start in independent mode
  ▪ The Administration Console and the management tools are unavailable

31.If an administration server running machine got crashed, how will you restart the server with same configuration on new machine?

o Oracle WebLogic Server allows the creation of a backup of the server as follows:
  ▪ Install Oracle WebLogic Server on a backup machine.
  ▪ Copy the application files to a backup machine.
  ▪ Copy the configuration files to a backup machine.
  ▪ Restart the Administration Server on a new machine.
o The new Administration Server contacts the Managed Servers and informs them that it is running on a new IP address.

32.How can you run Multiple Weblogic server instances in a same physical machine?

o You can run multiple instances of WLS using different configurations on the same physical machine at the same time by either:
  ▪ Assigning multiple IP addresses to a machine (multihoming) and defining each server to use a unique IP address
  ▪ Specifying the same IP address but using different listen ports
o A multihomed machine:
  ▪ Is a machine with multiple IP addresses
  ▪ Can run a different WLS instance that is bound to each IP address
  ▪ Can be used to configure a cluster on a single machine

33.How will you create Domain Template? Explain briefly?

o A domain template defines the full set of resources within a domain.
o Oracle provides sample templates for creating any platform domain.
o There are three ways to create domain templates:
  ▪ WLST offline command line tool
  ▪ pack command
  ▪ Domain Template Builder (config_builder.sh under WL_HOME/common/bin)
o Use the Domain Template Builder to create a domain template or an extension template.
o Using the Domain Template Builder:
  ▪ Define a domain and replicate it across multiple projects
  ▪ Distribute a domain packed with an application that has been developed to run in it

34.What are the default Weblogic provided Groups for security realm?

- Administrators
- Deployers
- Operators
- Monitors
- AppTesters
- CrossDomainConnectors
- AdminChannelUsers

35.What are the default Weblogic provided Roles for security realm?

- Admin
- Deployer
- Operator
- Monitor
- AppTester
- CrossDomainConnectors
- AdminChannelUsers
- Anonymous

36.What is the default Weblogic provided domain template file name and location?
Wls.jar is the default domain template and the location is WL_HOME\common\templates\domains

37.What are the elements of the Administration console?

- Change Center
- Domain Structure
- How do I...
- Tool Bar
- Breadcrumb Navigation
- System Status

38.What are the Node elements of the Administration console or Domain Structure?

- Environment (Servers, Clusters, Virtual Hosts, Migratable Targets, Machines ...)
- Deployment
- Services (Messaging, JDBC, Persistent Store, JTA, File T3, jCOM ...)
- Security Realms
- Interoperability
- Diagnostics (Log Files, Diagnostics Modules, Diagnostics Images, Archives, Context)

39.What are the Tool Bar elements in Weblogic?

- Welcome Message
- Connected to
- Home
- Log Out
- Preferences
- Record
- Help
- Search

40.How will you enable the Administration Console?
By default, the Administration Console is enabled. If you disable it, you can re-enable it using the WebLogic Scripting Tool (WLST). Start the Administration Server, then invoke WLST and use the following commands:
Using WLST to Re-enable the Console
connect("username","password")
edit()
startEdit()
cmo.setConsoleEnabled(true)
save()
activate()
The following attribute(s) have been changed on MBeans which require server re-start.MBean Changed : com.bea:Name=mydomain,Type=Domain Attributes changed :
ConsoleEnabled
Activation complete
ddisconnect()
exit()

41.How will you Enable and disable the domain configuration lock?
The Administration Console Change Center provides a way to lock a domain configuration so you can make changes to the configuration while preventing other accounts from making changes during your edit session.

The domain configuration locking feature is always enabled in production domains. It can be enabled or disabled in development domains. It is disabled by default when you create a new development domain.

To enable or disable the domain configuration locking feature in a development domain:

- In the banner toolbar region at the top of the right pane of the Console, click Preferences.
- Click User Preferences.
- Select or clear Automatically Acquire Lock and Activate Changes to enable or disable the feature.
- Click Save.
- After you finish

When you enable domain configuration locking, you must use the Change Center to lock and edit for the domain configuration.

42.What are Dynamic and Non-Dynamic Changes in the Weblogic Console? what is the difference?
Some changes you make in the Administration Console take place immediately when you activate them. Other changes require you to restart the server or module affected by the change. These latter changes are called non-dynamic changes. Non-dynamic changes are indicated in the Administration Console with this warning icon,.

Changes to dynamic configuration attributes become available once they are activated, without restarting the affected server or system restart. These changes are made available to the server and run-time hierarchies once they are activated. Changes to non-dynamic configuration attributes require that the affected servers or system resources be restarted before they become effective.

If a change is made to a non-dynamic configuration setting, no changes to dynamic configuration settings will take effect until after restart. This is to assure that a batch of updates having a combination of dynamic and non-dynamic attribute edits will not be partially activated.

Note that WebLogic Server's change management process applies to changes in domain and server configuration data, not to security or application data.

43.What is the information is going to store in the "security" folder of the Domain directory contents?
This directory holds the security-related files that are the same for every WebLogic Server instance in the domain:

- SerializedSystemIni.dat

This directory also holds security-related files that are only needed by the domain's Administration Server:

- DefaultAuthorizerInit.ldift
- DefaultAuthenticatorInit.ldift
- DefaultRoleMapperInit.ldift

44.What is the use of SerializedSystemIni.dat file in Weblogic?
It is important to protect passwords that are used to access resources in a WebLogic Server domain. In the past, usernames and passwords were stored in clear text in a WebLogic security realm. Now all the passwords in a WebLogic Server domain are hashed. The SerializedSystemIni.dat file contains the hashes for the passwords. It is associated with a specific WebLogic Server domain so it cannot be moved from domain to domain.

If the SerializedSystemIni.dat file is destroyed or corrupted, you must reconfigure the WebLogic Server domain. Therefore, you should take the following precautions:

Make a backup copy of the SerializedSystemIni.dat file and put it in a safe location.
Set permissions on the SerializedSystemIni.dat file such that the system administrator of a WebLogic Server deployment has write and read privileges and no other users have any privileges.

45.Explain about Domain Directory Contents?
By default, WebLogic Server creates domain directories under the
BEA_HOME/user_projects/domains directory. This section describes the contents of the domain
directory and its subfolders. In this section, domain-name, deployment-name, and server-name
represent names that you define when you create a domain.
Individual applications in a domain might create additional files and directories in the domain
directory.
If you have not yet created a domain, you can see an example of an existing domain directory by
looking in WL_HOME/examples/domains/wl_server where WL_HOME is the directory in which you
installed WebLogic Server.
Domain-name
The name of this directory is the name of the domain.
autodeploy
This directory provides a quick way to deploy applications in a development server. When the
WebLogic Server instance is running in development mode, it automatically deploys any applications
or modules that you place in this directory.

The files you place in this directory can be Java EE applications, such as:

- An EAR file
- A WAR, EJB JAR, RAR, or CAR archived module
- An exploded archive directory for either an application or a module

Bin
This directory contains scripts that are used in the process of starting and stopping the
Administration Server and the Managed Servers in the domain. These scripts are generally provided
as .sh files for UNIX and .cmd files for Windows. The bin directory can optionally contain other
scripts of domain-wide interest, such as scripts to start and stop database management systems,
full-text search engine processes, etc.

Config
This directory contains the current configuration and deployment state of the domain. The central
domain configuration file, config.xml, resides in this directory.

Config/configCache
Contains data that is used to optimize performance when validating changes in the domain's
configuration documents. This data is internal to WebLogic Server and does not need to be backed
up.

Config/diagnostics
This directory contains system modules for instrumentation in the WebLogic Diagnostic Framework.

Config/jdbc
This directory contains system modules for JDBC: global JDBC modules that can be configured
directly from JMX (as opposed to JSR-88).

Config/jms
This directory contains system modules for JMS: global JMS modules that can be configured directly
from JMX (as opposed to JSR-88).

Config/lib
This directory is not used in the current release of WebLogic Server.

Config/nodemanager
This directory holds configuration information for connection to the Node Manager.

Config/security
This directory contains system modules for the security framework. It contains one security provider configuration extension for each kind of security provider in the domain's current realm.

Config/startup
This directory contains system modules that contain startup plans. Startup plans are used to generate shell scripts that can be used as part of server startup.

ConfigArchive
This directory contains a set of JAR files that save the domain's configuration state. Just before pending changes to the configuration are activated, the domain's existing configuration state, consisting of the config.xml file and the other related configuration files, is saved in a versioned JAR file with a name like config.jar#1, config.jar#2, etc.

The maximum number of versioned JAR files to be kept is specified by the archiveConfigurationCount attribute of DomainMBean. Once this maximum number is reached, the oldest conversion archive is deleted before a new one is created.

Console-ext
This directory contains extensions to the Administration Console, which enable you to add content to the WebLogic Server Administration Console, replace content, and change the logos, styles and colors without modifying the files that are installed with WebLogic Server. For example, you can add content that provides custom monitoring and management facilities for your applications. See Extending the Administration Console.

Init-info
This directory contains files used for WebLogic domain provisioning. You should not modify any files in this directory.

Lib
Any JAR files you put in this directory are added to the system classpath of each server instance in the domain when the server's Java virtual machine starts.

Pending
This directory contains domain configuration files representing configuration changes that have been requested, but not yet activated. Once the configuration changes have been activated, the configuration files are deleted from this directory.

Security
This directory holds those security-related files that are the same for every WebLogic Server instance in the domain:

- SerializedSystemIni.dat

This directory also holds security-related files that are only needed by the domain's Administration Server:

- DefaultAuthorizerInit.ldift
- DefaultAuthenticatorInit.ldift
- DefaultRoleMapperInit.ldift

Servers
This directory contains one subdirectory for each WebLogic Server instance in the domain. The subdirectories contain data that is specific to each server instance.

Servers/server-name
This directory is the server directory for the WebLogic Server instance with the same name as the directory.

Servers/server-name/bin
This directory holds executable or shell files that can be or must be different for each server. The server environment script (setServerEnv.sh or setServerEnv.cmd) is an example of a file that resides here because it can differ from one WebLogic Server instance to the next, for example, depending on whether the server instance has its own startup plan.

Servers/server-name/cache
This directory holds directories and files that contain cached data. By "cached" here we mean that the data is a copy, possibly in a processed form (compiled, translated, or reformatted), of other data.

Servers/server-name/cache/EJBCompilerCache
This directory is a cache for compiled EJBs.

Servers/server-name/data
This directory holds files that maintain persistent per-server state used to run the WebLogic Server instance, other than security state, as opposed to temporary, cached or historical information. Files in this directory are important data that must be retained as the WebLogic Server instance is brought up, is brought down, crashes, restarts, or is upgraded to a new version.

Servers/server-name/data/ldap
This directory holds the embedded LDAP database. The run-time security state for the WebLogic Server instance is persisted in this directory.

Servers/server-name/data/store
This directory holds WebLogic persistent stores. For each persistent store, there is a subdirectory that holds the files that represent the persistent store. The name of the subdirectory is the name of the persistent store. By convention there is one store named default.

Servers/server-name/logs
This directory holds logs and diagnostic information. This information is historical in nature. It is not crucial to the operation of the server, and can be deleted (while the WebLogic Server instance is down, at least) without affecting proper operation. However, the information can be quite useful for debugging or auditing purposes and should not be deleted without good reason.

Servers/server-name/logs/diagnostic_images
This directory holds information created by the Server Image Capture component of the WebLogic Diagnostic Framework.

Servers/server-name/logs/jmsServers
This directory contains one subdirectory for each JMS server in the WebLogic Server instance. Each such subdirectory contains the logs for that JMS server. The name of the subdirectory is the name of the JMS server.

Servers/server-name/logs/connector
This directory is the default base directory for connector module (JCA ResourceAdapter) logs.

Servers/server-name/security
This directory holds security-related files that can be or must be different for each WebLogic Server instance. The file boot.properties is an example of a file that resides here because it can differ from one server to the next. This directory also maintains files related to SSL keys.

Servers/server-name/tmp
This directory holds temporary directories and files that are created while a server instance is running. For example, a JMS paging directory is automatically created here unless another location is specified. Files in this directory must be left alone while the server is running, but may be freely deleted when the server instance is shut down.

Tmp
This directory stores temporary files used in the change management process. You should not modify any files in this directory.

user_staged_config
By default, configuration information is automatically copied from the Administration Server to each Managed Server. If instead you prefer to stage configuration changes manually, you can use this directory as an alternative to the config directory.

46.How many ways you can change the configuration changes?
The change management features of WLS:

- o Enable you to distribute configuration changes throughout a domain securely, consistently, and predictably
- o Are the same, regardless of whether you are using:

- The WLS Administration Console
- The WebLogic Scripting Tool (WLST)
- The Java Management Extension (JMX) APIs

47.What is the user of WLST in Weblogic?

- The WLS command-line tools are useful:
  - For automating common administration activities
  - As an alternative to the Administration Console
  - When graphical tools are not supported
- WLST provides a command-line interface for:
  - Creating new WLS domains
  - Retrieving and updating WLS domain configurations
  - Deploying applications
  - Obtaining run-time server statistics

48.How many WLST modules are there? Explain?

- **Online mode:**
  - Connected to a running server
  - Access to all WLS configuration and run-time attributes
  - Create and activate change sessions similar to the WLS console
- **Offline mode:**
  - Domain not running
  - Access to only persisted domain configuration (config.xml)
  - Create or update domains similar to using the Configuration Wizard

49.What is the Node Manager (NM)? Explain briefly?
Node Manager (NM):

- Starts and stops Managed Servers remotely: server, domain, and cluster
- Available as either a Java-based or (for UNIX or Linux) a script-based process
- Monitors and acts on server health
- Runs on the same computers as the Managed Servers
- Can be run automatically in the background, as a Windows service or a UNIX daemon

50.How many versions of Node Managers are available?

- There are two versions of Node Manager:
  - Java-based Node Manager
  - Script-based Node Manager
- Java-based Node Manager runs within a Java Virtual Machine (JVM) process.
- Script-based Node Manager (used only for UNIX and Linux systems) does not have as much security, but provides the ability to remotely manage servers over a network using Secure Shell (SSH).

51.How Node Manager will work with the Weblogic Server? How will you configure Node Manager in WLS?

- o Node Manager must run on each computer that hosts the WLS instances that you want to control with Node Manager.
- o You should configure each computer as a machine in Oracle WebLogic Server, and assign each server instance, which is to be controlled by Node Manager, to the machine that the server instance runs on.
- o Node Manager should run as an operating system service, so that it automatically restarts upon system failure or reboot.

52.What is the Node Manager Default Behavior?

- o After WebLogic Server is installed, Node Manager is "ready-to-run" if Node Manager and Administration Server are on the same machine.
- o By default, the following behaviors are configured:
    - ▪ The Administration Console can use Node Manager to start the Managed Servers.
    - ▪ Node Manager monitors the Managed Servers that it started.
    - ▪ The automatic restart of Managed Servers is enabled.

53.To start Node Manager at system start up time, what we have to do?
We have to configure Node Manager as a Operating System Service.

- o It is recommended that you run Node Manager (NM) as:
    - ▪ A Windows service on Windows platforms and
    - ▪ A daemon on UNIX platforms
- o Running NM during system startup allows it to restart automatically when the system is rebooted.
- o Node Manager can be configured to start at boot time, as either of these:
    - ▪ A Windows service
    - ▪ A UNIX daemon

54.How will you configure Node Manager as Windows Service?

- o Edit **installNodeMgrSvc.cmd** to specify Node Manager's listen address and listen port.
- o Run **installNodeMgrSvc.cmd** to reinstall Node Manager as a service, listening on the updated address and port.
- o Delete the Node Manager Service using **uninstallNodeMgrSvc.cmd**.

55.Explain about Weblogic server Log Message Format?
When a WebLogic Server instance writes a message to the server log file, the first line of each message begins with #### followed by the message attributes. Each attribute is contained between angle brackets.

Here is an example of a message in the server log file:
####<Jan 05, 2010 10:46:51 AM EST> <Notice> <WebLogicServer> <MyComputer> <examplesServer> <main> <<WLS Kernel>> <> <null> <1080575211904> <BEA-000360> <Server started in RUNNING mode> In this example, the **message attributes** are: **Locale-formatted Timestamp**, **Severity**, **Subsystem**, **Machine Name**, **Server Name**, **Thread ID**, **User ID**,**Transaction ID**, **Diagnostic Context ID**, **Raw Time Value**, **Message ID**, and **Message Text**.
If a message is not logged within the context of a transaction, the angle brackets for Transaction ID are present even though no Transaction ID is present.
If the message includes a stack trace, the stack trace is included in the message text.
WebLogic Server uses the host computer's default character encoding for the messages it writes.
56.What is the Log Message Formant of Output to Standard Out and Standard Error?
When a WebLogic Server instance writes a message to standard out, the output does not include the #### prefix and does not include the Server Name, Machine Name, Thread ID, User ID, Transaction ID, Diagnostic Context ID, and Raw Time Value fields.

Here is an example of how the message from the previous section would be printed to standard out:

<jan 01, 2010 10:51:10 AM EST> <Notice> <WebLogicServer> <BEA-000360> <Server started in RUNNING mode>In this example, the message attributes are: **Locale-formatted Timestamp**, **Severity**, **Subsystem**, **Message ID**, and **Message Text**.
57.How many log Message Severity levels are there in Weblogic? Explain?
The severity attribute of a WebLogic Server log message indicates the potential impact of the event or condition that the message reports.

Severity	Meaning
TRACE	Used for messages from the Diagnostic Action Library. Upon enabling diagnostic instrumentation of server and application classes, TRACE messages follow the request path of a method.
INFO	Used for reporting normal operations; a low-level informational message.
NOTICE	An informational message with a higher level of importance.
WARNING	A suspicious operation or configuration has occurred but it might not affect normal operation.
ERROR	A user error has occurred. The system or application can handle the error with no interruption and limited degradation of service.
CRITICAL	A system or service error has occurred. The system can recover but there might be a momentary loss or permanent degradation of service.

ALERT	A particular service is in an unusable state while other parts of the system continue to function. Automatic recovery is not possible; the immediate attention of the administrator is needed to resolve the problem.
EMERGENCY	The server is in an unusable state. This severity indicates a severe system failure or panic.
DEBUG	A debug message was generated.

58.What is the default log Message Severity levels in Weblogic?
WebLogic Server subsystems generate many messages of lower severity and fewer messages of higher severity. For example, under normal circumstances, they generate many **INFO** messages and no **EMERGENCY** messages.

59.What is the Log Message Severity Level sequence from lowest to highest impact?
A log level object can specify any of the following values, from lowest to highest impact:
TRACE, DEBUG, INFO, NOTICE, WARNING, ERROR, CRITICAL, ALERT, EMERGENCY

60.How will you specify the logging implementation in Weblogic?
We will specify the logging implementation using "**Java Logging API**" or "**Log4j**" in Weblogic.

About Log4j
Log4j has three main components: **loggers, appenders, and layouts**. The following sections provide a brief introduction to Log4j.

Loggers
Log4j defines a Logger class. An application can create multiple loggers, each with a unique name. In a typical usage of Log4j, an application creates a Logger instance for each application class that will emit log messages. Loggers exist in a namespace hierarchy and inherit behavior from their ancestors in the hierarchy.

Appenders
Log4j defines appenders (handlers) to represent destinations for logging output. Multiple appenders can be defined. For example, an application might define an appender that sends log messages to standard out, and another appender that writes log messages to a file. Individual loggers might be configured to write to zero or more appenders. One example usage would be to send all logging messages (all levels) to a log file, but only ERROR level messages to standard out.

Layouts
Log4j defines layouts to control the format of log messages. Each layout specifies a particular message format. A specific layout is associated with each appender. This lets you specify a different log message format for standard out than for file output, for example.

Java Logging API
WebLogic logging services provide the Commons LogFactory and Log interface implementations that direct requests to the underlying logging implementation being used by WebLogic logging services.

To use Commons Logging, put the WebLogic-specific Commons classes,
$BEA_HOME/modules/com.bea.core.weblogic.commons.logging_1.3.0.0.jar, together with the commons-logging.jar file in one of the following locations:

APP-INF/LIB or WEB-INF/LIB directory
DOMAIN_NAME/LIB directory
Server CLASSPATH
Note: WebLogic Server does not provide a Commons logging version in its distribution

61.What is the user of Log Filters in Weblogic?
Log filters:

- o Control the log messages that get published
- o Are based on the values of message attributes
- o Can be applied to different message destinations:
  - Server log file
  - Server memory buffer
  - Server standard out
  - Domain log file

62.What is the user of Network channels in Weblogic?
Adds flexibility to the networking configuration:

- o Multiple NICs for a single WLS server
- o Specific NICs or multiple port numbers on a NIC for specific WLS servers
- o Ability to use multiple IP addresses with each server
- o Ability to use a single IP address with multiple ports
- o Ability to configure the cluster multicast port number independently of the port numbers used by the cluster members
- o Multiple SSL configurations on one server

Network channels:

- o Define the set of basic attributes of a network connection to WLS
- o Can assign multiple channels to a single server (segment network traffic)
- o Can prioritize internal (non-URL) connections
- o Can separate incoming client traffic from internal server to server traffic in a domain
- o A "default" channel gets generated when a server is created.

63.How will you configure a web application in Weblogic?
Web applications are configured using the web.xml and weblogic.xml deployment descriptors, which:

- Define the run-time environment
- Map URLs to servlets and JSPs
- Define application defaults such as welcome and error pages
- Specify J2EE security constraints
- Define work managers for applications
- Set the context root for the application

64.What information will be available in "web.xml" file?

The web.xml file is a deployment descriptor that is used to configure the following:

- Servlets and JSP registration
- Servlet initialization parameters
- JSP tag libraries
- MIME type mappings
- Welcome file list
- Error pages
- Security constraints and roles
- Resources
- EJB references

65.What information will be available in "weblogic.xml" file? Using weblogic.xml, you can configure the following:

- The application's root context path
- Application logging
- Security role mappings
- Advanced session settings
- Session clustering
- References to shared libraries
- References to server resources (data sources, EJBs, and so on)
- Work managers and threading
- Virtual directories
- JSP compiler options

66.To configure a "web service" Applications in Weblogic, what are all the files required as a deployment descriptor?

A Web service application:

- Responds to HTTP client requests using the Simple Object Access Protocol (SOAP)
- Uses the same structure as a Java EE Web application
- Supports two additional deployment descriptors:
  - webservices.xml
  - weblogic-webservices.xml

67. What is the Virtual directory Mappings? Which file you are going to provide these virtual directory mappings?

Virtual directories:

- Can be used to refer to physical directories
- Enable you to avoid the need to hard code paths to physical directories
- Allow multiple Web applications to share common physical directories for specific requests such as images
- Decrease duplication of files across applications
- Are configured in weblogic.xml

Example:
```
<virtual-directory-mapping>
  <local-path>c:/usr/gifs</local-path>
    <url-pattern>/images/*</url-pattern>
    <url-pattern>*.jpg</url-pattern>
</virtual-directory-mapping>
<virtual-directory-mapping>
    <local-path>c:/usr/common_jsps.jar</local-path>
    <url-pattern>*.jsp</url-pattern>
</virtual-directory-mapping>
```

68.What is the deployment descriptor file for ejb applications? What are all the information is going to provide in that file?
Ejb application deployment descriptor file in Weblogic is "weblogic-ejb-jar.xml".
Using weblogic-ejb-jar.xml, you can configure the following:

- Security role mappings
- Advanced security settings
- EJB clustering
- EJB pooling and caching
- Work managers and threading
-

69. What is an Enterprise Application?

- An enterprise application is a grouping of several resources into one deployable unit that is packaged in an .ear file.
- These resources include:
    - Web applications (.war)
    - EJB applications (.jar)
    - Java applications (.jar)
    - Resource adapters (.rar)

70. What is the user of Enterprise Applications? Use enterprise applications to:

- Avoid namespace clashes
- Declare application wide security roles
- Deploy an application as one unit
- Share application wide EJB resources
- Configure local JDBC data sources
- Configure local JMS resources
- Configure local XML resources

71. What is the user of "weblogic-application.xml" deployment descriptor file?
Using weblogic-application.xml, you can configure:

- o References to shared libraries
- o Work managers and threading
- o Default EJB and Web application parameter values

We can configure enterprise wide WLS-specific features with the weblogic-application.xml deployment descriptor:

- o XML parsers
- o XML entity mappings
- o JDBC data sources
- o JMS connection factories and destinations
- o Security realms

72.What is the user of Weblogic shared java EE Libraries?
A Shared Java EE library:

- o Is a reusable portion of a Web or enterprise application
- o Is referenced by other deployed applications
- o Avoids duplicating source files among Java EE projects
- o Can contain deployment descriptors that are merged with the application's descriptors

73.Explain about deployment methods in Weblogic?

- o WLS supports three deployment methods:
  - Auto-deployment
  - Console deployment
  - Command-line deployment
- o You can deploy:
  - Enterprise, Web, and EJB applications
  - Web services
  - J2EE libraries
  - JDBC, JMS, and Diagnostic Framework modules
  - Resource adapters
  - Optional packages
  - Client application archives
- o Applications and EJBs can be deployed:
  - In an archived file (.ear, .war, .jar)
  - In an exploded (open) directory format

74.How many ways we can deploy an application to Weblogic servers?
Several methods are available to deploy the Oracle WebLogic Server applications and shared libraries, including:

- o Administration Console
- o WebLogic Scripting Tool (WLST)
- o weblogic.Deployer Java class

- o wldeploy Ant task
- o Auto-deployment folder

75.Explain about auto deployment in Weblogic?
If Production Mode is OFF:

- o You can install an application simply by copying it (manually or using the console) to the "autodeploy"
  directory of the domain
- o The Administration Server monitors this directory for new, changed, or removed applications
- o This configures, targets, and deploys the application only to the Administration Server
- Location of Applications Directory:
- $BEA_HOME/user_projects/domains/*domain_name*/autodeploy

76.Explain about FastSwap and On-Demand Deployment in Weblogic?

- o WebLogic's FastSwap feature is:
    - Enabled using the WebLogic deployment descriptors
    - Available only if the domain is *not* running in production mode
    - Applicable only to Web applications that are *not* archived
- o When enabled:
    - WebLogic automatically reloads the modified Java class files within applications
    - Developers can perform iterative development without an explicit redeployment
- On-demand deployment:

Excerpt from weblogic.xml:
<fast-swap>true</fast-swap>

77.While deploying an application to Weblogic, what is the difference between Development and Production Mode?

- o An Administration Server starts using either:
    - ▪ The development mode, which turns auto-deployment on
    - ▪ The production mode, which turns auto-deployment off
- o The Administration Server starts in the mode selected at domain creation time.
- o The mode is set for all Oracle WebLogic Servers in a given domain.

78.Explain about console deployment method?
Deploying with the console allows full administrator control:

- o Installation from a location of your choice
- o Manual configuration of application name
- o Targeting of application to individual servers and/or clusters
- o Configuring the application without targeting it
- o Activating deployment when desired

79.Explain about command line deployment?

- o The weblogic.Deployer utility allows you to perform deployment operations similar to those available in the console.
- o weblogic.Deployer actions can also be scripted with the Ant task wldeploy.

weblogic.Deployer Syntax:
% java weblogic.Deployer [options]
[-deploy|-undeploy|-redeploy|-start|-stop|-listapps] [file(s)]
Prepare and deploy a new application:
java weblogic.Deployer -adminurl t3://adminserver:7001
   -username myuser –password mypass –name HRServices
   -source /usr/HRServices.ear -targets serverA **–deploy**
Redeploy an application:
java weblogic.Deployer -adminurl t3://adminserver:7001
   -username myuser –password mypass –name HRServices
   –redeploy
Undeploy an application:
java weblogic.Deployer -adminurl t3://adminserver:7001
   -username myuser –password mypass –name HRServices
   –undeploy
List all applications:
java weblogic.Deployer -adminurl t3://adminserver:7001
   -username myuser –password mypass **-listapps**
80.What is JNDI?

- o The Java Naming and Directory Interface is an API for uniformly accessing the different naming and directory services.
- o This is a major step forward because:
    - ▪ Different services use vastly different naming schemes
    - ▪ Java applications can now navigate seamlessly across databases, files, directories, objects, and networks

81.Why the JNDI required in Weblogic?
In Oracle WebLogic Server, JNDI serves as a repository and lookup service for J2EE objects, including:

- o   Enterprise JavaBeans (EJB) home stubs
- o   JDBC DataSources
- o   JMS connection factories, queues, and topics
- o   Remote Method Invocation (RMI) stubs

82.What is the use of Naming Services?
A naming service provides a method for mapping identifiers to entities or objects.

Term	Definition	Example
Binding	The association of an atomic name and an object	www.example.com is bound to 209.10.217.38.
Namespace	A set of unique names in a naming system	www.example.com/ products

83.Explain about Contexts and Sub-contexts in JNDI?

- o   Subcontexts are referenced through the dot delimiters (.).
- o   The subcontexts must be created before objects are placed into them.
- o   Typically when objects are bound to a JNDI tree, subcontexts are automatically created based on the JNDI name.

 If the following context exists: com.oracle.examples
You cannot bind: com.oracle.examples.ejb.SomeObject
Without first creating: com.oracle.examples.ejb

84.How can I set deployment order for applications?
 WebLogic Server 8.1 allows you to select the load order for applications. See the ApplicationMBean LoadOrder attribute in Application. WebLogic Server deploys server-level resources (first JDBC and then JMS) before deploying applications. Applications are deployed in this order: connectors, then EJBs, then web Applications. If the application is an EAR, the individual components are loaded in the order in which they are declared in the application.xml deployment descriptor.

85.Can I refresh static components of a deployed application without having to redeploy the entire application?
Yes. You can use weblogic.Deployer to specify a component and target a server, using the following syntax:
java weblogic.Deployer -adminurl http://admin:7001 -name appname -targets server1,server2 -deploy jsps/*.jsp

86.When should I use the -nostage option?

Set the staging mode to -nostage (using weblogic.Deployer or the Administration Console) if you don't want to copy deployment files but want to deploy an application from its present location. All target servers must be able to access the same set of deployment files.

87.When should I use the external_stage option?

Set -external_stage using weblogic.Deployer if you want to stage the application yourself, and prefer to copy it to its target by your own means.

88.What are the Deployment Tools for Developers?

WebLogic Server provides several tools for deploying applications and stand-alone modules:

- wldeploy is an Ant task version of the weblogic.Deployer utility. You can automate deployment tasks by placing wldeploy commands in an Ant build.xml file and running Ant to execute the commands.
- weblogic.PlanGenerator is a command-line tools that enables developers to export an application's configuration for deployment to multiple WebLogic Server environments.
- The deployment API allows you to perform deployment tasks programmatically using Java classes.
- The autodeploy domain directory allows you to deploy an application quickly for evaluation or testing in a development environment.

89.What is the Deployment order of Weblogic Server at Server Startup time?

By default, WebLogic Server deploys applications and resources in the following order:

- JDBC system modules
- JMS system modules
- J2EE Libraries and optional packages
- Applications and stand-alone modules
- Startup classes

Note: WebLogic Server security services are always initialized before server resources, applications, and startup classes are deployed. For this reason, you cannot configure custom security providers using startup classes, nor can custom security provider implementations rely on deployed server resources such as JDBC.

90.How will you "Kill the JVM" or running Weblogic server in different operating systems?

Each WebLogic Server instance runs in its own JVM. If you are unable to shut down a server instance using the scripts which are provided by the Weblogic (stopWeblogic.cmd/ stopManagedWeblogic.cmd), you can use an operating system command to kill the JVM.

Caution: If you kill the JVM, the server immediately stops all processing. Any session data is lost. If you kill the JVM for an Administration Server while the server is writing to the config.xml file, you can corrupt the config.xml file.

Some common ways to kill the JVM are as follows:

- If the shell (command prompt) in which you start the server is still open, you can type Ctrl-C.
- On a Windows computer, you can use the Task Manager to kill a JVM.

On a UNIX computer, you can use the "ps" command to list all running processes. Then you can use the kill command to kill the JVM .

91. Can I generate deployment descriptor files automatically?
Yes, WebLogic Builder automatically generates deployment descriptor files for your J2EE applications. See WebLogic Builder Online Help.

92.Can I set the deployment order for application modules? For standalone modules?
The Load Order attribute controls the deployment order of standalone modules and applications relative to other modules and applications of the same type.

For example, standalone EJBs with smaller Load Order values are deployed before those with higher values.

Modules that are deployed as part of an Enterprise Application (EAR file or directory) are deployed in the order in which they are specified in the application.xml deployment descriptor.

93.What is the difference between the WL_HOME/config/examples/applications folder and the WL_HOME/config/examples/stage folder?
The applications folder is intended for applications that are not yet ready for a production environment. WebLogic Server dynamically deploys the contents of the applications folder. The stage folder (or a folder that you create for the same purpose) is for storing copies of deployment files that are ready for deployment in a production environment (deployments that use the stage or external_stage deployment modes).

94.How do I turn the auto-deployment feature off?
The auto-deployment feature checks the applications folder every three seconds to determine whether there are any new applications or any changes to existing applications and then dynamically deploys these changes.

The auto-deployment feature is enabled for servers that run in development mode. To disable auto-deployment feature, use one of the following methods to place servers in production mode:

In the Administration Console, click the name of the domain in the left pane, then select the Production Mode checkbox in the right pane.
At the command line, include the following argument when starting the domain's Administration Server:
-Dweblogic.ProductionModeEnabled=true Production mode is set for all WebLogic Server instances in a given domain.

95.I downloaded the WebLogic Server installation file, but the installation program will not run. What should I do?

The installation file may have been corrupted during the download. Run a checksum on the installation file and check with technical support for the proper values.

96.Do I need to install WebLogic Server as root on Solaris?

No you don't need to be root, depending on directory permissions.

97.Can I run the Configuration Wizard outside the installer?

Yes. You can start the Configuration Wizard from the Start menu or using a script in the utils directory. See Creating Domains and Servers in Configuring and Managing WebLogic Server.

98.How do I edit the config.xml file?

The persistent configuration for a domain of WebLogic Servers and clusters is stored in an XML configuration file (config.xml). You can modify this file in the following ways:

- Use the Administration Console. See "Using the Administration Console" in the Administration Console Online Help.
- If you want to create scripts that automate domain management, use the weblogic.Admin utility. See "weblogic.Admin Command-Line Reference" in the .
- If you want to create Java-based management applications, use the Java Management Extensions (JMX) Application Programming Interface (API). See the Programming WebLogic Management Services with JMX guide.
- If you want to edit the config.xml file directly (not recommended), see the BEA WebLogic Server Configuration Reference.

99.What is the free pool?

The free pool is a data structure the EJB container uses to cache anonymous instances of a given bean type. The free pool improves performance by reusing objects and skipping container callbacks when it can.

100.Can I use the PointBase DBMS included with WebLogic Server for development or production?

PointBase Server is an all-Java DBMS product included in the WebLogic Server distribution solely in support of WebLogic Server evaluation, either in the form of custom trial applications or through packaged sample applications provided with WebLogic Server. Non-evaluation development and/or production use of the PointBase Server requires a separate license be obtained by the end user directly from PointBase.

101.How can I enable Oracle Advanced Security encryption on the JDBC Oracle Thin driver with a WebLogic JDBC Connection Pool?

Oracle Advanced Security encryption relies on features available through connection properties in the JDBC driver from Oracle. You can specify connection properties in a WebLogic JDBC connection pool in the Properties attribute. This attribute is available on the JDBC Connection Pool —> Configuration —> General tab in the Administration Console. When WebLogic Server creates database connections for the connection pool, it passes the properties to the JDBC driver so that connections are created with the specified properties.

For example, to enable Oracle Advanced Security encryption, you may want to specify the following options:

Properties: user=SCOTT

oracle.net.encryption_client=ACCEPTED

oracle.net.encryption_types_client=RC4_256   oracle.net.crypto_checksum_client=ACCEPTED

protocol=thin

Note: See the Oracle documentation for details about required properties for Oracle Advanced Security encryption. Properties listed above are for illustration only.

The resulting entry in the config.xml file would look like:
```
<JDBCConnectionPool
DriverName="oracle.jdbc.driver.OracleDriver"
Name="oraclePool"
Password="{3DES}1eNn7kYGZVw="
Properties="user=SCOTT;
oracle.net.encryption_client=ACCEPTED;
oracle.net.encryption_types_client=RC4_256;    oracle.net.crypto_checksum_client=ACCEPTED;
 protocol=thin"
URL="jdbc:oracle:thin:@server:port:sid"/>
```
Note: Line breaks added for readability.

102.When should I use a TxDataSource instead of a DataSource?
When you select Honor Global Transactions in the Administration Console, you create a JDBCTxDataSource in the config.xml file (the default). If you clear the Honor Global Transactions check box, you create a JDBCDataSource in the config.xml file. See "When to Enable Global Transactions in a Data Source" in the Administration Console Online Help.

103.Can I enable requests to a JDBC connection pool for a database connection to wait until a connection is available?
Yes. You can set two JDBC connection pool properties to enable connection requests to wait for a connection:

- ConnectionReserveTimeoutSeconds
- HighestNumWaiters

104.What happens when my database is restarted or becomes unreachable? Does my connection pool stick around?
Yes. The pool is independent of its ability to communicate with to the DBMS. All connections in the connection pool may become defunct, but the connection pool still exists. You can configure the connection pool so that WebLogic Server tests the connections in the pool and replaces bad connections when it can.

To manually restart the connection pool using the Administration Console after a database failure, you can undeploy the connection pool by removing all of its deployment targets, and then redeploy the connection pool by adding deployment targets.

To do this from the command line using "weblogic.Admin", set the "Targets" attribute of the pool to an empty string ("") and then set it to the desired set of targets.

105.When should I use MultiPools?
You can use MultiPools in one of two ways
1) For high availability in the event a database connection fails, or
2) For load balancing between JDBC connection pools. Because you can choose only one option, you need to determine the primary purpose of your MultiPool.

Note: If you implement Multipools for a JDBC application, do not configure driver-level load balancing or failover for the connection pools used by the MultiPool—the MultiPool provides the

same functionality as configuring driver-level load balancing or failover.

106.What is JDBC?

JDBC is an API for accessing databases in a uniform way.
JDBC provides:

- o   Platform-independent access to databases
- o   Location transparency
- o   Transparency to proprietary database issues
- o   Support for both two-tier and multitier models for database access

The Java Database Connectivity (JDBC) specification:

- o   Is a platform- and vendor-independent mechanism for accessing and updating a database
- o   Provides transparency from proprietary vendor issues
- o   Requires the use of a *driver*

JDBC drivers are supplied by WLS or your database vendor.

107.What is the use of Data Source in Weblogic?

Data sources:

- o   Allow database connectivity to be managed by the application server
- o   Use a dynamic pool of reusable database connections
- o   Are obtained by applications from the server's JNDI tree

108.What is the scope of the Data Source in Weblogic server?

- o   Each data source configuration or "module" is persisted as a separate XML document.
- o   The system modules that are created with the console or WLST are:
  - ▪   Stored in the domain's config/Jdbc directory
  - ▪   Available to all applications in the domain
- o   Application-specific modules are:
  - ▪   Deployed as part of Java EE enterprise applications
  - ▪   Accessible only by the containing application

109.List default Weblogic provided JDBC Drivers?

- o Oracle and third-party drivers are included in WLS installation for many popular database products:
  - Oracle 9*i*, 10*g*, and 11*g*
  - Sybase Adaptive Server
  - Microsoft SQL Server
  - IBM DB2
  - Informix
  - MySQL
  - PointBase
- o By default, these drivers are added to server's classpath.

110.What is a Data Source?

- o A data source object provides a way for a JDBC client to obtain a database connection from a connection pool.
- o A data source:
  - Is stored in the Oracle WebLogic Server JNDI tree
  - Can support transactions
  - Is associated with a connection pool

111.What is a Connection Pool?

- o A connection pool is a group of ready-to-use database connections associated with a data source.
- o Connection pools:
  - Are created at Oracle WebLogic Server startup
  - Can be administered using the Administration Console
  - Can be dynamically resized to accommodate increasing load

112.What are the benefits of having Data Sources and Connection Pools in Weblogic?

- o The following are some advantages of this approach:
  - Time and overhead are saved by using an existing database connection.
  - Connection information is managed in one location in the Administration Console.
  - The number of connections to a database can be controlled.
  - The DBMS can be changed without the application developer having to modify the underlying code.
- A connection pool allows an application to "borrow" a DBMS connection

113.How Data Sources are used in Weblogic?

A client retrieves a data source through a JNDI lookup and uses it to obtain a database connection.

114.What is Multi Data Sources?

- o Multi data source:
    - Is an abstraction around a group of data sources
    - Determines which data source to use to satisfy the request depending on the algorithm selected in the multi data source configuration:
        - Load balancing or failover
    - Is bound to the JNDI tree
- o XA support for multi data sources:
    - The WLS JDBC supports using multi data sources in XA transactions.
    - You can configure the data sources contained within the multi data source to use XA JDBC drivers.

115.What is the functionality of "Message-Oriented Middleware"?

- Message-oriented middleware refers to an infrastructure that supports messaging.
- Typical message-oriented middleware architectures define the following elements:

- Message structure
- The way to send and receive messages
- Scaling guidelines

116.Explain about Point-to-Point (Queue)?

Many producers can serialize messages to multiple receivers in a queue.

117.Explain about Publish-Subscribe (Topics)?

Publishing and subscribing to a topic decouples producers from consumers.

118.Explain about Oracle WebLogic Server JMS Features?

Oracle WebLogic Server JMS supports:

- o PTP and pub/sub domains
- o Guaranteed and transactional message delivery
- o Durable subscribers
- o Distributed destinations
- o Recovery from failed servers

119.How will you implement "Messaging Service" in Weblogic?

- o In Oracle Weblogic Server, the messaging service is implemented through a JMS server.
- o A JMS server receives and distributes messages.

120.What is the user of "Connection Factories" in JMS?

- o JMS connection factories are used to set default client connection parameters, including:
    - Message priority
    - Message time-to-live (TTL)
    - Message persistence
    - Transactional behavior

- Acknowledgement policy
- Flow control
- WLS provides a default client connection factory that:
    - Uses WebLogic's default connection settings
    - Is located on the server JNDI tree at weblogic.jms.ConnectoryFactory

121.What is the use of "Destination" in JMS? How many types of "Destinations" are available in JMS?

- A destination is a lightweight object that is stored in JNDI.
- It is the target on a JMS server for sending or receiving messages.
- The JMS destination types are:
    - Queue
    - Topic

122.Explain about "Queue Destinations"?
In JMS point-to-point messaging, note the following:

- Clients communicate with a *queue* destination.
- Messages are distributed to consumers in a serial fashion (first in, first out).
- Each message is delivered only to a single consumer.

123.Explain about "Topic Destinations"?
In JMS publish/subscribe messaging, the following is true:

- Clients communicate with a *topic* destination.
- Messages are broadcast to all subscribers.
- A message can be saved until at least one subscriber has consumed it ("durable").

124.Explain about Threshold and Quotas in JMS?

- A threshold and a quota can be set for the server and destination objects.
- A quota is a limit defined for the JMS-administered objects; it includes the following values:
    - The maximum number of bytes that can be stored
    - The maximum number of messages that can be stored

o A threshold is a limit that triggers message paging, flow control, and logged warnings, using:
  ▪ Upper and lower values for the number of bytes
  ▪ Upper and lower values for the number of messages

125.Difference between "Durable Subscribers and Subscriptions"?

o Durable subscribers register durable subscriptions for guaranteed message delivery even if the subscribers are inactive.
o A subscriber is considered active if the Java object that represents it exists.
o By default, subscribers are nondurable.
o Administrators configure:
  ▪ Where the messages are persisted
  ▪ Persistent connection factories and destinations

126.What is "Persistent Messaging"? When to Use it?

o Persistent messaging permits messages in memory to be written out to a persistent store.
o Configure persistent messaging if:
  ▪ Development requires durable subscriptions (use durable subscribers in the application)
  ▪ You require that in-progress messages persist across server restarts

127.How will you Configuring a Durable Subscription in Weblogic?

o To configure durable subscriptions, an administrator must:
  ▪ Create and configure a JMS store
  ▪ Configure connection factories or destinations as persistent
  ▪ Associate the JMS store with the JMS server
o The JMS store can be configured to use either:
  ▪ A file store
  ▪ A JDBC store (a connection pool)

128.How a Durable Subscription Works?

o If a subscriber client is active, messages are delivered normally.

   o When the client becomes active again, its ID is used to retrieve and redeliver messages.

129.What Node manager can do in Weblogic server?

You can use Node Manager to:

- Start, shut down, and restart an Administration Server
- Start, shut down, suspend, and restart Managed Servers
- Automatically restart the Administration and Managed Servers on failure
- Monitor servers and collect log data

30.Does the WebLogic JMS server find out about closed or lost connections, crashes, and other problems and does it recover from them?

Yes, but how it does this depends on whether a Java client crashes or WebLogic Server crashes, as follows:

- If a Java client crashes then the JMS server will clean up all the outstanding server-side resource from the crashed client JVM, such as:
- JMS connection(s) from the crashed client JVM
- JMS temporary destination(s) created under the above JMS connection(s)
- JMS session(s) created under the above JMS connection(s)
- JMS client(s) created under the above JMS session(s) (connection consumer and regular consumer)
- JMS browser(s) created under the above session(s)
- JMS producer(s) created under the above session(s)
- If WebLogic Server crashes and it is the front-end to the JMS server, then:
  - A JMS client will lose all the server-side resources listed above.
  - The client's javax.jms.ExceptionListener.onException(...) will be called (if javax.jms.JMSConnection.setExceptionListener is set) with a LostServerException, which extends JMSException.
- If WebLogic server crashes and it is a back-end to the JMS server, then:
- A JMS client may partially lose some of the server-side resources listed above (only the resource on the crashed server, such as JMS temporary destination(s), JMS client(s) and JMS browser(s).
- The client's javax.jms.ExceptionListener.onException(...) will be called (if weblogic.jms.extensions.WLSession.setExceptionListener is set) with a ConsumerClosedException, which extends JMSException.

131.What Is the Java Message Service?

An enterprise messaging system enables applications to communicate with one another through the exchange of messages. A message is a request, report, and/or event that contains information needed to coordinate communication between different applications. A message provides a level of abstraction, allowing you to separate the details about the destination system from the application code.

The Java Message Service (JMS) is a standard API for accessing enterprise messaging systems. Specifically, JMS:

- Enables Java applications sharing a messaging system to exchange messages
- Simplifies application development by providing a standard interface for creating, sending, and receiving messages

132.How many Messaging Modules are available in Weblogic?

JMS supports two messaging models: point-to-point (PTP) and publish/subscribe (pub/sub). The messaging models are very similar, except for the following differences:

- PTP messaging model enables the delivery of a message to exactly one recipient.
- Pub/sub messaging model enables the delivery of a message to multiple recipients.

133.Explain about Point-to-Point Messaging?

The point-to-point (PTP) messaging model enables one application to send a message to another. PTP messaging applications send and receive messages using named queues. A queue sender (producer) sends a message to a specific queue. A queue receiver (consumer) receives messages from a specific queue.

134.Explain about Publish/Subscribe Messaging?

The publish/subscribe (pub/sub) messaging model enables an application to send a message to multiple applications. Pub/sub messaging applications send and receive messages by subscribing to a topic. A topic publisher (producer) sends messages to a specific topic. A topic subscriber (consumer) retrieves messages from a specific topic.

135.Explain about Message Persistence?

As per the "Message Delivery Mode" section of the JMS Specification, messages can be specified as persistent or non-persistent:

- A **persistent message** is guaranteed to be delivered once-and-only-once. The message cannot be lost due to a JMS provider failure and it must not be delivered twice. It is not considered sent until it has been safely written to a file or database. WebLogic JMS writes persistent messages to a WebLogic persistent store (disk-base file or JDBC-accessible database) that is optionally targeted by each JMS server during configuration.
- **Non-persistent messages** are not stored. They are guaranteed to be delivered at-most-once, unless there is a JMS provider failure, in which case messages may be lost, and must not be delivered twice. If a connection is closed or recovered, all non-persistent messages that have not yet been acknowledged will be redelivered. Once a non-persistent message is acknowledged, it will not be redelivered.

136.Topics vs. Queues?

Surprisingly, when you are starting to design your application, it is not always immediately obvious whether it would be better to use a Topic or Queue. In general, you should choose a Topic only if one of the following conditions applies:

- The same message must be replicated to multiple consumers.
- A message should be dropped if there are no active consumers that would select it.
- There are many subscribers, each with a unique selector.

It is interesting to note that a topic with a single durable subscriber is semantically similar to a queue. The differences are as follows:

- If you change a topic selector for a durable subscriber, all previous messages in the subscription are deleted, while if you change a queue selector for consumer, no messages in the queue are deleted.
- A queue may have multiple consumers, and will distribute its messages in a round-robin fashion, whereas a topic subscriber is limited to only one consumer.

137.Asynchronous vs. Synchronous Consumers?

In general, asynchronous (onMessage) consumers perform and scale better than synchronous consumers:

- Asynchronous consumers create less network traffic. Messages are pushed unidirectionally, and are pipelined to the message listener. Pipelining supports the aggregation of multiple messages into a single network call.

Note: In WebLogic Server, your synchronous consumers can also use the same efficient behavior as asynchronous consumers by enabling the Prefetch Mode for Synchronous Consumers option on JMS connection factories

- Asynchronous consumers use fewer threads. An asynchronous consumer does not use a thread while it is inactive. A synchronous consumer consumes a thread for the duration of its receive call. As a result, a thread can remain idle for long periods, especially if the call specifies a blocking timeout.

- For application code that runs on a server, it is almost always best to use asynchronous consumers, typically via MDBs. The use of asynchronous consumers prevents the application code from doing a blocking operation on the server. A blocking operation, in turn, idles a server-side thread; it can even cause deadlocks. Deadlocks occur when blocking operations consume all threads. When no threads remain to handle the operations required unblocking the blocking operation itself, that operation never stops blocking.

138.What is a Distributed Destination?
A distributed destination is a set of destinations (queues or topics) that are accessible as a single, logical destination to a client. A distributed destination has the following characteristics:

- It is referenced by its own JNDI name.
- Members of the set are usually distributed across multiple servers within a cluster, with each destination member belonging to a separate JMS server.

139.Why Use a Distributed Destination?
Applications that use distributed destinations are more highly available than applications that use simple destinations because WebLogic JMS provides load balancing and failover for member destinations of a distributed destination within a cluster. Once properly configured, your producers and consumers are able to send and receive messages through the distributed destination. WebLogic JMS then balances the messaging load across all available members of the distributed destination. When one member becomes unavailable due a server failure, traffic is then redirected toward other available destination members in the set.
140.How many Types of Distributed Destinations are available?

- Uniform Distributed Destinations
- Weighted Distributed Destinations

Uniform Distributed Destinations
In a uniform distributed destination (UDD), each of the member destinations has a consistent configuration of all distributed destination parameters, particularly in regards to weighting, security, persistence, paging, and quotas.
Oracle recommends using UDDs because you no longer need to create or designate destination members, but instead rely on WebLogic Server to uniformly create the necessary members on the JMS servers to which a UDD is targeted. This feature of UDDs provides dynamic updating of a UDD when a new member is added or a member is removed.
Weighted Distributed Destinations

In a weighted distributed destination, the member destinations do not have a consistent configuration of all distributed destination parameters, particularly in regards to weighting, security, persistence, paging, and quotas.

Oracle recommends converting weighted distributed destinations to UDDs because of the administrative inflexibility when creating members that are intended to carry extra message load or have extra capacity (more weight). Lack of a consistent member configuration can lead to unforeseen administrative and application problems because the weighted distributed destination cannot be deployed consistently across a cluster.

141.What is JNDI?

JNDI (Java Naming and Directory Interface) is a Java EE lookup service that maps names to services and resources. JNDI provides a directory of advertised resources that exist on a particular stand-alone (unclustered) WebLogic server, or within a WebLogic server cluster. Examples of such resources include JMS connection factories, JMS destinations, JDBC (database) data sources, and application EJBs.

A client connecting to any WebLogic server in a WebLogic cluster can transparently reference any JNDI advertised service or resource hosted on any WebLogic server within the cluster. The client doesn't require explicit knowledge of which particular WebLogic server in the cluster hosts a desired resource.

142.What is a JMS connection factory?

A JMS connection factory is a named entity resource stored in JNDI. Applications, message driven beans (MDBs), and messaging bridges lookup a JMS connection factory in JNDI and use it to create JMS connections. JMS connections are used in turn to create JMS sessions, producers, and consumers that can send or receive messages.

143.What is a JMS connection-id?

JMS connection-ids are used to name JMS client connections. Durable subscribers require named connections, otherwise connections are typically unnamed. Note that within a clustered set of servers or stand-alone server, only one JMS client connection may use a particular named connection at a time. An attempt to create new connection with the same name as an existing connection will fail.

144.What is the difference between a JMS topic and a JMS queue?

JMS queues deliver a message to one consumer, while JMS topics deliver a copy of each message to each consumer.

145.What is a non-durable topic subscriber?

A non-durable subscriber creates unnamed subscriptions that exist only for the life of the JMS client. Messages in a non-durable subscription are never persisted—even when the message's publisher specifies a persistent quality of service (QOS). Shutting down a JMS server terminates all non-durable subscriptions.

146.What is a durable subscriber?

A durable subscriber creates named subscriptions that continue to exist even after the durable subscriber exits or the server reboots. A durable subscriber connects to its subscription by specifying topic-name, connection-id, and subscriber-id. Together, the connection-id and subscriber-id uniquely name the subscriber's subscription within a cluster. A copy of each persistent message published to a topic is persisted to each of the topic's durable subscriptions. In the event of a server crash and restart, durable subscriptions and their unconsumed persistent messages are recovered.

147.What is the WebLogic Store-and-Forward Service?

The WebLogic Store-and-Forward (SAF) Service enables WebLogic Server to deliver messages reliably between applications that are distributed across WebLogic Server instances. For example, with the SAF service, an application that runs on or connects to a local WebLogic Server instance can reliably send messages to a destination that resides on a remote server. If the destination is not available at the moment the messages are sent, either because of network problems or system failures, then the

messages are saved on a local server instance, and are forwarded to the remote destination once it becomes available.

148.When should I use the WebLogic Store-and-Forward Service?

The WebLogic Store-and-Forward (SAF) Service should be used when forwarding JMS messages between WebLogic Server 9.0 or later domains. The SAF service can deliver messages:

- Between two stand-alone server instances.
- Between server instances in a cluster.
- Across two clusters in a domain.
- Across separate domains.

149.What is a messaging bridge?

Messaging bridges are administratively configured services that run on a WebLogic server. They automatically forward messages from a configured source JMS destination to a configured target JMS destination. These destinations can be on different servers than the bridge and can even be foreign (non-WebLogic) destinations. Each bridge destination is configured using the four common properties of a remote provider:

- The initial context factory.
- The connection URL.
- The connection factory JNDI name.
- The destination JNDI name.

Messaging bridges can be configured to use transactions to ensure exactly-once message forwarding from any XA capable (global transaction capable) JMS provider to another.

150.When should I use a messaging bridge?

Typically, messaging bridges are used to provide store-and-forward high availability design requirements. A messaging bridge is configured to consume from a sender's local destination and forward it to the sender's actual target remote destination. This provides high availability because the sender is still able to send messages to its local destination even when the target remote destination is unreachable. When a remote destination is not reachable, the local destination automatically begins to store messages until the bridge is able to forward them to the target destination when the target becomes available again.

151.When should I avoid using a messaging bridge?

Other methods are preferred in the following situations:

- Receiving from a remote destination—use a message driven EJB or implement a client consumer directly.
- Sending messages to a local destination—send directly to the local destination.
- Environment with low tolerance for message latency. Messaging Bridges increase latency and may lower throughput. Messaging bridges increase latency for messages as they introduce an extra destination in the message path and may lower throughput because they forward messages using a single thread.
- Forward messages between WebLogic 9.0 domains—Use WebLogic Store-and-Forward.

152.How many types of JMS stores are available?

- The JMS store can be configured to use either:
  - A file store
  - A JDBC store (a connection pool)

153.How will you configure a JMS JDBC Store?

- To configure JMS JDBC persistence, perform the following:
  - Create a JDBC DataSource.
  - Create a JMS store and refer to the JDBC DataSource.
  - Refer to the JMS store from the JMS server configuration.
- The required infrastructure (tables and so on) is created automatically.

154.What Is a Transaction?

- A transaction is a mechanism to handle groups of operations as though they were one.
- Either all operations in a transaction occur or none occur at all.
- The operations involved in a transaction might rely on multiple servers and databases.

155.How many Types of Transactions are there? Explain?

- A local transaction deals with a single resource manager. Local transactions use the non-Extended Architecture (non-XA) interface between Oracle WebLogic Server and the resource manager.
- A distributed transaction coordinates or spans multiple resource managers.
- Global transactions can deal with multiple resource managers. Global transactions use the Extended Architecture (XA) interface between Oracle WebLogic Server and the resource managers.

- You need to create non-XA or XA resources for local transactions. However, for global transactions, you need to create only XA resources.

156.Explain about Two-Phase Commit Protocol?

- o The Two-Phase Commit (2PC) protocol uses two steps to commit changes within a distributed transaction.
  - Phase 1 asks the RMs to prepare to make the changes.
  - Phase 2 asks the RMs to commit and make the changes permanent or to roll back the entire transaction.
- o A global transaction ID (XID) is used to track all the changes associated with a distributed transaction.

157.Explain about Extended Architecture Protocol (XA)?
The Extended Architecture (XA) protocol:

- o Is the interface that is used between WLS and the RMs
- o Implements the 2PC protocol
- o Allows programs to control the RMs that are involved in distributed transactions

158.What is the user of "Transaction Log"?

- o Each server has a transaction log that stores information about committed transactions coordinated by the server that may not have been completed.
  - Oracle WebLogic Server uses the transaction log when recovering from system crashes or network failures.
- You cannot directly view the transaction log because the records are in a binary format and are stored in the default persistent store for the server.

159.Explain about Logging Last Resource?

- o You can configure a JDBC data source to enable the Logging Last Resource (LLR) transaction optimization, which:
  - Enables one non-XA resource to participate in a global transaction
  - Has improved performance and the same ACID guarantee as XA
- o The LLR optimization improves performance by:
  - Removing the need for an XA JDBC driver to connect to the database. XA JDBC drivers are typically inefficient compared to non-XA JDBC drivers.
  - Reducing the number of processing steps to complete the transaction, which also reduces network traffic and I/O
  - Removing the need for XA processing at the database level (if the database is the one non-XA resource)

160.What Is LDAP?
The Lightweight Directory Access Protocol:

- o Is derived from X.500
- o Provides a hierarchical lookup service
- o Supports sophisticated searching
- o Can be secured via SSL

161.Explain Security Realms?

- o A security realm is a collection of system resources and security service providers.
- o Only one security realm can be active at a given time.
- o A single security policy is used in any realm.
- o Users must be recognized by an authentication provider of the security realm.
- o Administration tasks include creating security realms.

162.What Is SSL?

Secure Sockets Layer (SSL) is a protocol that enables:

- o Connection security through encryption
- o A server to authenticate to a client
- o A client to authenticate to a server (optional)
- o Data integrity such that the data that flows between a client and server is protected from tampering by a third party

163.What Is a Deployment Plan?

- ▪ It is an optional XML document that resides outside an application archive.
- ▪ It configures an application for deployment to a specific WLS environment.
- ▪ It is created and owned by administrators or developers for a particular environment.

A JavaEE deployment plan:

- ▪ Is an XML file that is associated with an application
- ▪ Resides outside an application archive
- ▪ Sets or overrides the values in the JavaEE deployment descriptors

Allows a single application to be easily customized to multiple deployment environments

164.What are the Advantages of Production Redeployment? Saves the trouble of:

- ▪ Scheduling application downtime

- Setting up redundant servers to host new application versions
- Managing client access to multiple application versions manually
- Retiring older versions of an application manually
- **Explain about Packaging Applications?**
  - When you deploy applications to a single Managed Server, you can deploy the applications in an exploded format.
  - Oracle recommends deploying packaged applications to a cluster of Managed Servers as .war, .ear, or .jar file

165.What are the Oracle WebLogic Server SSL Requirements?

- To enable Oracle WebLogic Server SSL, you must:
  - Obtain an appropriate digital certificate
  - Install the certificate
  - Configure SSL properties
  - Configure two-way authentication (if desired)
    - SSL impacts performance.

166.What is the user of keytool Utility in WLS?

- keytool is a standard J2SE SDK utility for managing:
  - o The generation of private keys and the corresponding digital certificates
  - o Keystores (databases) of private keys and the associated certificates
- The keytool utility can the display certificate and keystore contents.

167.How WLS Protecting Against Attacks? What kinds of Attacks are there?

- WLS can help protect applications against several attacks:
  - Man-in-the-middle attacks
  - Denial of service (DoS) attacks
  - Large buffer attacks
  - Connection starvation attacks
- The slides that follow detail the countermeasures that WLS provides to address these attacks.

168.Explain about Man-in-the-Middle Attacks?

- In the "man-in-the-middle" attack, a third party poses as a destination host intercepting messages between the client and the real host.
- Instead of issuing the real destination host's SSL certificate, the attacker issues his or her own hoping that the client would accept it as being from the real destination host.
- The "man-in-the-middle" attacks can be resisted by using a Host Name Verifier.
- A Host Name Verifier validates that the host to which an SSL connection is made is the intended or authorized party.
- WLS provides a Host Name Verifier by default.

- A custom Host Name Verifier can be created by implementing the interface weblogic.security.SSL.HostnameVerifier

169.Explain about Denial of Service Attacks (DOS)?

- DoS attacks are attempts by attackers to prevent legitimate users of a service from using that service.
- There are three basic types of attack:
  - Consumption of scarce, limited, or nonrenewable resources
  - Destruction or alteration of configuration information
  - Physical destruction or alteration of network components

Harden WLS against "denial of service" attacks by:

- Filtering incoming network connections
- Configuring consumable WLS resources with the appropriate threshold and quotas
- Limiting access to configuration information and backing up configuration files
- Preventing unauthorized access by protecting passwords against password-guessing attacks

170.Explain about Filtering Network Connections?

- WLS can be configured to accept or deny network connections based on the origin of the client.
- This feature can be used to:
  - Restrict the location from which connections to WLS are made
  - Restrict the type of connection made, that is, allow only SSL connections and reject all others
- To filter network connections, create a class that implements the ConnectionFilter interface and install it using the Administration Console.

171.Explain about Large Buffer Attacks?

- o Individuals can try to bring down a Web site by sending a large buffer of data, which starves the system of memory.
- o Administrators can combat this attack by setting a threshold for incoming data.

172.Explain about Connection Starvation?

- o Individuals can try to take down a Web site by sending small, incomplete messages that cause the server to wait.
- o Administrators can combat this attack by setting a threshold.
- o Connections time out while waiting for the remainder of the data if they have reached the threshold set by the administrator.

173.Explain about User Lockout?

- o Individuals attempt to hack into a computer using various combinations of usernames and passwords.
- o Administrators can protect against this security attack by setting the lockout attributes.
- o The administrator can unlock a locked user using the console.

174.How will you Protecting the Administration Console?

- o You can configure a separate administration port for all administration traffic.
- o You can change the Context path of the console.
- o You can disable the console (application).

175.What are the advantages of Deployment Plan?

- o Works by setting or overriding the deployment property values that are defined in an application's WLS deployment descriptor
- o Helps to easily modify an application's WLS configuration for deployment into different multiple WLS environments without modifying the deployment descriptor files that are included in the application archive

Enables an application to be deployed to multiple domains or to multiple target servers and clusters that have a different configuration within the same

1. domain

176.What is the user of weblogic.PlanGenerator?

1. Is a Java-based deployment configuration tool
2. Is primarily intended for developers who want to export portions of an Oracle WebLogic Server deployment configuration into an XML deployment plan
3. Enables you to generate a basic Oracle WebLogic Server configuration for applications that have only J2EE deployment descriptors

177.How will you generate a Deployment Plan Using the Administration Console?

1. The Administration Console automatically generates or updates the deployment plan.
2. You can generate a deployment plan using the Administration Console using the following steps:
    - Prepare the deployment files.
    - Install the application archive.
    - Save the configuration changes to a deployment plan.

178.How many types of Staged Deployment are available in WLS?
You can configure deployment per server or for each application as:

1. **staged** (**default**): Files are copied to the preconfigured staging directory for preparation and activation.
2. **nostage**: Files are deployed from a static location.
3. **external_stage**: Files are copied by a user or a third-party tool before deployment.

179.Explain about Production Redeployment?

1. You can redeploy a revised version of a production application alongside the older version:
    - Without affecting the existing clients to the application
    - Without interrupting the availability of the application to the new client request

2. Oracle WebLogic Server automatically manages client connections so that:
   - Existing clients continue to use the older application
   - New client requests are directed to the newer application

The older version is undeployed after all current clients complete their work.

180.Explain about Application Availability, after deploying/install to WLS?

1. By default, when an application is redeployed:
   - It is unavailable to clients for a brief time
   - Existing clients lose any conversational state
2. Some types of applications require availability 24 hours a day, seven days a week.
3. Third-party proxy solutions are possible, but they require multiple servers.

181.Explain about WebLogic Production Redeployment?

1. Production redeployment:
   1. Allows two versions of a single Web application or module to run simultaneously
   2. Requires you to include unique version information either:
      - Within the application's META-INF/MANIFEST.MF file
      - As part of the deployment process
2. When a new version is redeployed, WLS automatically:
   - Routes existing clients to the prior (retired) version
   - Routes new clients to the new version
   - Undeploys the prior version when all existing clients finish their work or their conversations time out

182.What are the Requirements and Restrictions for Production Redeployment in WLS?

1. Production redeployment is supported for:
   - Stand-alone Web Application (WAR) modules and Enterprise Applications (EARs) whose client accesses the application via a Web application (HTTP)
   - Enterprise Applications that are accessed by inbound JMS messages from a global JMS destination, or from inbound JCA requests
   - All types of Web Services, including conversational and reliable Web Services
2. Production redeployment is not supported for:
   - Stand-alone EJB or Resource Archive (RAR) modules
   - Applications that use JTS drivers
   - Applications that obtain JDBC data sources via the DriverManager API instead of using the JNDI lookup

- Applications that include the EJB 1.1 container-managed persistence (CMP) EJBs
3. A deployed application must specify a version number.
4. WLS can host a maximum of two different versions of an application at one time.
5. When you are redeploying a new version of an application, the following features cannot change:
    - Deployment targets
    - Security model
    - Persistent store settings

183.Redeploying Versus Distributing an applications in WLS?

1. Distributing is an alternative to deploying an application.
    - Distributing an application prepares it for deployment by copying its files to all target servers and validating the files.
    - You can start a distributed application in Administration mode. Access to the application is then restricted to a configured Administration channel.
2. Distributing a new version of the application makes it available for testing before being released for general consumption.
3. Redeploying a new version of an application places the application immediately into use and makes it available to new client requests.

184.How will you Distributing a New Application Version in WLS?

1. Use the weblogic.Deployer –distribute command.
2. After the application is distributed, start the application in Administration mode.
3. Test the application.
4. When ready, start the application (without using –adminmode).
5. Optionally set a retirement timeout for the older version of the application.

185.How will you Create MANIFEST.mf file for Production Redeployment?

1. Create MANIFEST.mf with the following contents:
    Name: java/util/
    Specification-Title: Java Utility Classes
    Specification-Version: 1.2
    Specification-Vendor: Sun Microsystems, Inc.
    Implementation-Title: java.util
    Implementation-Version: build57
    Implementation-Vendor: Sun Microsystems, Inc.
2. Then add an entry for WLS versioning:

Weblogic-Application-Version: Version1.0Beta

186.What is a cluster Definition?

1. A cluster is a group of Oracle WebLogic Server instances that work in coordination.
2. Clustering provides:
    - High availability
    - Load balancing
    - Scalability

187.What Is a Cluster?
A cluster:

1. Is a logical group of Oracle WebLogic Servers within a domain
2. Supports features to
   provide high
   availability for:
   - Whole servers
   - Web applications/services
   - EJB applications
   - JMS
3. Is transparent to a client

188.What are the Benefits of Clustering?

1. There are two main benefits of clustering together Oracle WebLogic Servers:
   1. Scalability
   2. High availability
2. Scalability is the ability to provide more capacity for an application, in this case, by adding additional servers without having to make major architectural changes.
3. High availability ensures that when a server (in a cluster) fails, there are other servers to take over the work, so that the client is not affected.

189.What are the Key Capabilities of clustering?
The key capabilities of a WebLogic cluster are:

1. Application failover
   - When an object in an application that is performing a task becomes unavailable, another object takes over and finishes the job.
2. Site failover
   - When all the services and applications in a single site fail, they can switch to a separate site and continue processing.
3. Server migration
   - When a server fails, pinned services can be migrated to another server in a cluster.
4. Load balancing

- Tasks and communications are evenly distributed across multiple servers.

190.Explain about Cluster Architecture?

1. Applications are generally broken into multiple tiers, each representing its distinct functionality:
2. Web tier
3. Presentation tier
4. Business or object tier
5. WebLogic provides clustering support for all three tiers.
6. Other services, such as JMS and JDBC, can take advantage of clusters but load-balancing and failover is a little different.

191.How the Server will communicate in a Cluster?

- The Oracle WebLogic Server instances in a cluster communicate with one another using two different techniques:
2. Unicast/multicast (UDP)
3. Sockets (peer-to-peer TCP)
- The server instances use IP unicast or multicast to broadcast the availability of services and heartbeats that indicate continued availability.
- IP multicast broadcasts one-to-many communications among clustered instances.
- IP unicast is an alternative to multicast to handle cluster messaging and communications. The unicast configuration is much easier because it does not require cross-network configuration that multicast requires.
- IP sockets are used for peer-to-peer communications between server instances.

192.Explain about Cluster Communication in WLS?
- Members of a cluster cooperate to achieve high availability using the following:
- Broadcast messages such as "heartbeats"
- Peer-to-peer IP sockets
- You can configure broadcast communication to use either:
- IP unicast
- A dedicated IP multicast address (224.0.0.0 through 239.255.255.255) and port
- If heartbeats are not received from a cluster member, the server is marked as "failed" and its services are not used.

193.How the Weblogic Server detect a server Failure?
WebLogic clusters detect the failure of a server instance in the following ways:

Through the use of IP sockets

Through the Oracle WebLogic Server heartbeat

If a server in the cluster unexpectedly closes its socket, it is marked as "failed" and its services are not used.

Server instances use multicast to broadcast heartbeats every 10 seconds to the other server instances in the cluster.

If three heartbeats are missed from a peer server, the server is marked as "failed" and its services are not used.

194.Explain about One-to-Many Communications in WLS?
* Oracle WebLogic Server uses one-to-many communication for:
* Clusterwide JNDI updates
* Cluster "heartbeats"
* Because all one-to-many communications occur over IP multicast, when you design a cluster, consider the following factors:
* If your cluster spans multiple subnets, your network must be configured to reliably transmit messages.
* A firewall can break IP multicast transmissions.
* The multicast address should not be shared with other applications.
* Multicast storms may occur.

195.Explain about Peer-to-Peer Communications in WLS?
Oracle WebLogic Server uses peer-to-peer communications for:

Accessing the nonclustered objects that reside on a remote server instance in the cluster
Replicating HTTP session states and stateful session EJB states between a primary and a secondary server
Accessing the clustered objects that reside on a remote server instance (typically, in a multitier cluster architecture)

196.Explain about Multitier Communications in WLS?
* Multitier clusters require more IP sockets than a combined-tier cluster:
* One socket for replicating session states
* One socket for each Oracle WebLogic Server in the EJB cluster, for accessing remote objects

As an example, using a three-node cluster, the worst-case scenario would be five open sockets per server:

- One primary and secondary replicated session
- Each server simultaneously invokes a remote EJB method on each node in the cluster.

197.How many cluster Configuration Options are available in WLS?
There are multiple ways to create and configure an Oracle WebLogic Server cluster:

    Administration Console
    Configuration Wizard
    WebLogic Scripting Tool (WLST)
    Using the Cluster MBean

198.Explain about Two-Phase Deployment?

    Applications are deployed using two-phase deployment (TPD).
    Applications are copied to the cluster and activated in two phases:
    - Phase 1: Application components and modules are distributed to the server.
    - Phase 2: The application is deployed if phase 1 is successful and client access is permitted.
    This ensures that an application is available and active on each node before clients can access it.

199.How will you Deploy Applications to a Cluster?

    All nodes must be running before an application is deployed to a cluster.
    If phase 2 of the two-phase deployment fails, the application is still deployed to other nodes in the cluster.
    WebLogic allows partial deployment of applications to a partitioned server.
    Session replication for deployed applications may fail if a node is partitioned at the time of deployment.
    - Avoid this by using the enforceClusterConstraints tag with weblogic.Deployer.
    - Or select the Enable Cluster Constraints check box in the console.
    Do not change cluster membership while deploying applications to the cluster.

200.Explain about HTTP Session Failover?

Web applications use HTTP sessions to track information in server memory for each client.
By default, when a client fails over to another server in the cluster, its session information is lost.
Oracle WebLogic Server supports several *Session Replication* strategies to recover sessions from failed servers:

- In-memory replication
- JDBC replication
- File replication

Replication is configured for each Web application within its weblogic.xml file.
Similar options are available for stateful EJB applications.

201.Explain about HTTP Session State Replication in WLS?

1. Oracle WebLogic Server provides clustering support for JSPs and servlets by replicating the HTTP session state.
2. To benefit from HTTP session state clustering, you must ensure that the session state is persistent, by configuring:
   - In-memory replication
   - JDBC replication
   - File system replication
3. You must also access the cluster via a collection of Web servers with identically configured proxy plug-ins or load-balancing hardware.
4. Session persistence is configured using the <session-descriptor> element in the weblogic.xml deployment descriptor file.
   - Each persistence method has its own set of configurable parameters.

202.Explain about Replication Groups in WLS?

1. A replication group is a logical grouping of related servers in a cluster.
2. WLS enables you to determine where to put backup objects using replication groups.
3. WLS attempts to:
   - Send backup objects to a preferred secondary replication group, if it is configured
   - Send backup objects to a different machine
   - Avoid sending backup objects to servers in the same replication group

203.What is the user of Replication Groups in WLS?

1. Replication groups:
   - Represent a subset of servers within a cluster
   - Help to determine the placement of secondary sessions (avoid replicating within the same room, for example)
   - Are not explicitly defined in the console-like machines and clusters
2. WLS attempts to:
   1. Send secondary sessions to servers that are assigned to the *preferred secondary replication group* of the primary server
   2. Avoid sending secondary sessions to servers that are assigned to the same replication group as the primary server

204.Explain about In-Memory Replication in WLS?

1. Each user's session always exists on two servers:
   - Primary
   - Secondary
2. Every update to the primary session is automatically replicated on the secondary server, either:
   - Synchronously (default)
   - Asynchronously (batch)
3. WLS can replicate:
   - HttpSession objects
   - Stateful session EJBs
4. Session objects exist on only two servers.
5. Secondary:
   - The server is determined by the replication group and machine definition.
   - The object is created immediately after the primary object is created.
6. Primary failure makes the backup object the primary object.

205.What are all the Requirements for In-Memory Replication in WLS?

1. Subsequent requests from the same client must have access to the same primary object.
2. To use in-memory replication for the HTTP session state, clients must access the cluster using either:
   1. The load-balancing hardware (WLS aware)
   2. A collection of Web servers, or a single Web server, with WebLogic proxy plug-ins (configured identically)
   3. Oracle WebLogic Server configured with HTTPClusterServlet

206.How will you Configuring In-Memory Replication in WLS?

1. Configure the proxy server (if applicable).
2. Optionally define replication groups and/or machines.
3. Specify the persistence type in the weblogic.xml deployment descriptor; the options include:
   1. replicated
   2. replicated_if_clustered
   3. async-replication-across-cluster

207.Explain about JDBC Replication in WLS?

- HTTP sessions are persisted to a database using a common JDBC data source.
- The required data definition language (DDL) file is available in the documentation.
- *All* members of the cluster have access to any client's session for failover purposes (no primary or secondary).
- All server instances have access to all sessions.
- Subsequent requests from the same client can be handled by any server.
6. Great failover capability
7. Significant performance reduction
- Changing session objects causes (slow) database synchronization.

203.What is the user of Replication Groups in WLS?

- Replication groups:
    - Represent a subset of servers within a cluster
    - Help to determine the placement of secondary sessions (avoid replicating within the same room, for example)
    - Are not explicitly defined in the console-like machines and clusters
- WLS attempts to:
    - Send secondary sessions to servers that are assigned to the *preferred secondary replication group* of the primary server
    - Avoid sending secondary sessions to servers that are assigned to the same replication group as the primary server

204.Explain about In-Memory Replication in WLS?

- Each user's session always exists on two servers:
    - Primary
    - Secondary
- Every update to the primary session is automatically replicated on the secondary server, either:
    - Synchronously (default)
    - Asynchronously (batch)
- WLS can replicate:
    - HttpSession objects
    - Stateful session EJBs
- Session objects exist on only two servers.
- Secondary:
    - The server is determined by the replication group and machine definition.
    - The object is created immediately after the primary object is created.
- Primary failure makes the backup object the primary object.

205.What are all the Requirements for In-Memory Replication in WLS?

- Subsequent requests from the same client must have access to the same primary object.
- To use in-memory replication for the HTTP session state, clients must access the cluster using either:
  - The load-balancing hardware (WLS aware)
  - A collection of Web servers, or a single Web server, with WebLogic proxy plug-ins (configured identically)
  - Oracle WebLogic Server configured with HTTPClusterServlet

206.How will you Configuring In-Memory Replication in WLS?

- Configure the proxy server (if applicable).
- Optionally define replication groups and/or machines.
- Specify the persistence type in the weblogic.xml deployment descriptor; the options include:
  - replicated
  - replicated_if_clustered
  - async-replication-across-cluster

207.Explain about JDBC Replication in WLS?

- HTTP sessions are persisted to a database using a common JDBC data source.
- The required data definition language (DDL) file is available in the documentation.
- *All* members of the cluster have access to any client's session for failover purposes (no primary or secondary).
- All server instances have access to all sessions.
- Subsequent requests from the same client can be handled by any server.
  - Great failover capability
  - Significant performance reduction
- Changing session objects causes (slow) database synchronization.

208.How will you Configuring JDBC Replication in WLS?

- Create the required table in the database.
- Create a JDBC data source that has read/write privileges for your database.
- Configure JDBC session persistence in the weblogic.xml deployment descriptor.

Ex:
```
<session-descriptor>
  <persistent-store-type>jdbc</persistent-store-type>
  <persistent-store-pool>MyDataSource</persistent-store-pool>
</session-descriptor>
```

209.JDBC Persistent Table Configuration

A database table named WL_SERVLET_SESSIONS must exist with read/write access:

210.Explain about File Replication?

File replication is similar to JDBC replication, but it persists sessions to a highly available file system**.**

- Session state may also be stored in a file.
- For file-based persistence:
  - You must create the directory in which to store the file
  - The file must have the appropriate access privileges

211.How will you Configuring File Replication in WLS?

- Create a folder shared by all servers on the cluster on a highly available file system.
- Assign read/write privileges to the folder.
- Configure file session persistence in the weblogic.xml deployment descriptor.

Ex:
<session-descriptor>
  <persistent-store-type>*file*</persistent-store-type>
  <persistent-store-dir>*/mnt/wls_share*</persistent-store-dir>
</session-descriptor>

212.Explain about Cross-Cluster Replication in WLS?

WebLogic provides the ability to replicate HTTP sessions across two clusters in separate domains:

- o This is most applicable to clusters that are distributed geographically.
- o Configure a global proxy to direct clients back to the same cluster ("cluster affinity").
- o Configure a specific network channel for cross-cluster communication.

213.When can't I use WebLogic Store-and-Forward?

You can't use the WebLogic Store-and-Forward service in the following situations:

- Receiving from a remote destination—use a message driven EJB or implement a client consumer directly.
- Sending messages to a local destination—send directly to the local destination.
- Forwarding messages to prior releases of WebLogic Server. See Q. What is a messaging bridge?.

- Interoperating with third-party JMS products (for example, MQSeries). See Q. What is a messaging bridge?.
- When using temporary destinations with the JMSReplyTo field to return a response to a request.
- Environment with low tolerance for message latency. SAF increases latency and may lower throughput.

214..Explain about Embedded LDAP Server?

- In WLS, users, groups, and authorization information is stored in an embedded LDAP server.
- Several properties can be set to manage the LDAP server, including:
  - Credentials
  - Backup settings
  - Cache settings
  - Replication settings

215.How will you create a Deployment Plan?

- Tools for creating a deployment plan:
  - weblogic.PlanGenerator
  - Administration Console
- Goals for creating a deployment plan:
  - To expose the external resource requirements of the application as variables in the deployment plan
  - To expose additional configurable properties, such as tuning parameters as variables in the deployment plan
- WLS includes tools to accelerate deployment plan creation.
- The Administration Console:
  - Generates a skeleton plan.xml if a plan folder is detected with a newly deployed application
  - Updates plan.xml when you use the console to modify the deployment descriptor settings
- The weblogic.PlanGenerator Java class can also generate a skeleton plan.xml for an existing application.

216..Explain the relationship between Transaction and Resource Managers?

- o A transaction manager coordinates multiple resource managers.
- o The 2PC protocol is used to coordinate the transaction.
- o The XA protocol implements 2PC.

217..What is a topic subscription?

A topic subscription can be thought of as an internal queue of messages waiting to be delivered to a particular subscriber. This internal queue accumulates copies of each message published to the topic after the subscription was created. Conversely, it does not accumulate messages that were sent before the subscription was created. Subscriptions are not sharable, only one subscriber may subscribe to a particular subscription at a time.

218.Explain about Trust and Identity in SSL?

- o SSL and keystore are configured independently.
- o For the purpose of backward compatibility, this release of Oracle WebLogic Server supports private keys and a trusted WebLogic Keystore provider.
- o Identity:
    - Private key and digital certificate (can now be looked up directly from the keystore, not necessarily as a stand-alone file outside the keystore)
- o Trust:
    - Certificates of trusted Certificate authorities

219..How will you access SSL enabled applications?

- o WLS uses SSL to secure HTTP and t3 communication.
- o To use SSL, clients access WLS via the https or t3s protocols.
    - https://localhost:7002/orderStock
    - t3s://localhost:7002/useCreditCard