WAS Topics :

1. Installation
2. Profile Creation
3. Federation
4. Application Deployment
5. Apache Configuration
6. Global security
7. Class Loaders
8. Fix Packs
9. Dumps
10. DB Configuration

**WAS Installation:**

Prerequisites:

* Before installing WAS check Prerequisites
* 2GB Disk Space
* 1GB RAM
* 1GB free space in temp directory
* File permissions
* Compatibility
* Go to the setup
* Take the back up of response file and edit parameters

License Acceptance =true

Installation Type=” Install New”

Installation location=/opt/IBM/Websphere/Appserver

Check the Pre-requisites=true

Execute command: ./install –options <path of modified response file> -silent

**PROFILE: --**

A profile is an environment, which contains server, admin console, node and some supporting configuration files, which helps to do admin activities on servers and applications.

TYPES OF PROFILES: --

1). Default Profile (Application Server Profile)

2). Deployment manager profile (DMGR profile)

3). Custom or managed profile

4). Cell profile

5). Admin Agent

6). Secure Proxy

7). Job Manager

We can create profile in 3 ways.

1). GUI Mode

2). Silent Mode

3). Command line mode

**COMMAND LINE MODE**:Go to WAS-Root/bin directory and execute the below command

WAS-Root/bin> ./manageprofile.sh -create –profileName<name of the profile> -profilePath<absolute path of the profile> -templatePath<absolute template path> -nodeName<name of the node> -cellName<name of the cell> -hostName<name of the host>

**Federation:**

Federation is the process adding a node either from Appserver profile (or) cusstom profile to the Dmgr cell.

we can add a node by using the command >appserv01/bin>addnode.sh <hosname of DMGR><SOAP ]connector port number of DMGR> -includeapps -userName \*\*\* -password \*\*\*

\*At the time of fedaration,a process called "Node Agent" will be created to communicate between Dmgr and Federated profile sever and application.

\*Once the node of Appserver profile and custom profile is available with the Dmgr, We can be

Managed through a single Dmgr admin console.

Note 1:

\*After federation and configuration, if a node agent is down at that time there won't

be any impact to the appsever and application.

\*End user can access applications normally but we can't do administration activities through the Dmgr admin console.

Note 2:

After federation and configuration, if the Dmgr is down at that time also, there won't be any impact to the appservers and applications.

\*But we can't access admin console of Dmgr .

Note 3:

A node which is having a NodeAgent as "Managed Node".

A node which is don't have a NodeAgent as "Un-Managed Node".

**Application Deployment:**

We can deploy on application in 4 ways

1. By using admin console
2. By using scripting-jYTHON/JACL
3. By using rapid deployment/hot deployment
4. By using application server tool kit.

Steps to deploy an application by using Admin Console:

1. Login to the admin console and expand application.
2. Select enterprise application and select install.
3. Browse the location of war file (or) ear file (if it is a war file provide a context root).
4. Provide the application parameters like application name, installation location, it you don’t specify any location, by default application will be deployed under

<profile-home>/installed apps directory, whether it contains web services and EJB’s.

1. Specify the target under which server(Q) which cluster that application had to be deployed.
2. Map the data sources and EJB’s with JNDI
3. Select the virtual host name. By default, it will take default-host, save the changes and start the application.

Steps to Access an application:

1. Find the target of the application.
2. Find the virtual host name.
3. Find the port no for that virtual host,
4. Make sure that port no is registered under virtual host, host aliases.
5. Find the context root of the application and access the application.

Apache configuration:

1. Install app server
2. Install web server
3. Install plug-in while installing the plug-in we have to specify to which web server we are configuring plugins, httpd.conf file location, web server port number, web server definition and app server location.
4. Generate plugin-cfg.xml file by using this command--------genplugincfg.sh
5. Copy that generate plugin-cfg.xml file to the plugin where httpd.conf file, web sphere plug-in config path is pointing.

**DB Configuration:**

* Create JDBC providers
* Create a Data Source
* Test the connection

**JDBC Provider**:

* It specifies the type of the database and implementation type either connection pool data source or XA data source

**Data Source**:

* It is a unique name which holds information about Database server like hostname of Database, type of the server, port number of the DB server, DB name, user name and password

**Create a JDBC provider:**

* Login to admin console and expand Resources
* Expand JDBC
* Expand JDBC provider
* Select Scope : cell: profilename cell
* Select new
* Select Data Base type: Oracle
* Select provider type : Oracle JDBC Driver
* Select implementation type : connection pool data source
* Specify the jar file location

**Create a Data source:**

* Expand resources
* Expand Data Source
* Select new
* Provide Data source name (oracle\_DS)
* Provide JNDI Name(jdbc/oracle\_DS)
* Provide URL : jdbc:oracle:thin:@localhost:1521:xe
* Select J2C authentication alias
* Select New
* Provide alias
* Provide User ID and Password of Data Base
* Select Data Source
* Select New
* Provide Data Source name and JNDI name
* Provide Data Base type, Provider Type and Implementation type
* Select component managed authentication alias
* Select OK
* Save changes

**Test the Connection:**

* Select the Data Source and Test the connection
* If the test connection is successful it will show a message Test connection successful.
* If the test connection is failure, synchronize the node

**For synchronization:**

* First stop all the node agent
* syncnode.bat hostname of dmgr soapconnector port num of dmgr
* startnode.sh
* Test for the connection

**Class loaders:**

Class loader is a component under JVM which loads jar files into the JVM. The different types of class loaders are:

1. JVM class loaders
2. Web Sphere Extension class loaders
3. Web sphere server class loaders
4. Application module class loaders
5. Web module class loaders

1.JVM class loaders:

It lodes the jar files which are under JVM class path.

2. Web Sphere Extension class loaders:

It loads the jar files which are under <was-root>/lib, class and extension directories

3. Web sphere server class loaders:

It loads the jar files which are under shared libraries. (shared libraries :which is common to all)

4. Application module class loaders:

It loads application related jar file.

5. Web module class loaders

It loads web module jar files into the JVM.

Class loader policies:

Under server level

* Single
* Multiple

Under Application level

* Application
* Module
* If a class loader policy under server level is single a single application class loader will be created for all the applications of the server.
* If a class loader policy under server level is multiple a repeated application class loader will be created each and every application.
* If a class loader policy under application level is “application” web modules class loader will not be created, all the web modules jar files will be loaded by application class loader.
* If a class loader policy under application level is “module” web modules class loader will be created, all the web module s jar files will be loaded by web module class loader.

**Global Security:**

It’s to provide security to our environment, only authenticated and authorized users login the servers. We can enable global security in 4 ways

1. Local OS user Registry
2. Custom user Registry
3. LDAP user Registry(light weight directory access protocol)
4. Federated Repository

Steps to configure Global security by using local O.S user Registry:

1. Create user Accounts in your Operating System.
2. Assign password for that accounts.
3. Login to the admin console and expand security.
4. Select: Global Security.
5. Select security configuration wizard.
6. Select local O.S option to configure with local O.S user Registry.
7. Provide user ID and password.
8. Under LTPA authentication mechanism, confirm the password once again.
9. Enable administrative security check box.
10. Select local operating system under available realm definitions.
11. Save the changes and restart the server.(Dmgr)
12. Now Access the admin console by using http:<host-name>:9043/ibm/console.
13. Provide username and password to login to the Admin console.

Steps to configure Global security by using custom user Registry:

1. Create two files a)Users.Registry. b)Groups.Registry
2. Add user accounts information under User.Registry file.
3. Add groups information under Groups.Registry.
4. Login to the admin console and expand security.
5. Select Global security
6. Select security configuration wizard and select custom user registry option.
7. Create two variables users file and groups file.
8. Provide the absolute path of users.registry and groups.registry as a value for that variable.
9. Enable administrative security check box and select custom registry under available reatm definition.
10. Save the changes and restart the server.
11. Login to the admin console by using http://<host-name>:9043/ibm/console url.

Steps to configure Global security by using LDAP user Registry:

1. Login to the admin console.
2. Expand security and select Global security.
3. Select security configuration Wizard and select standalone LDAP registry.
4. Provide LDAP server details like username ,type of directory server, host name of directory server, port no of directory server(by default LDAP server port no is 389,if SSL is enabled the port no is 636), based distinguished name, bind distinguished name, and bind password.
5. Confirm the password under LTPA authentication mechanism.
6. Save the changes and restart the Dmgr.

**Connection pool:**

It contains pre-defined connection objects, server we need to create a new connection object for each and every request.

Server will use existing connection objects from connection pool, once a transaction is completed that connection object will be back to the pool and the same connection object can be assigned for further incoming request. So that we can reduce time delay in the response.

**Connection pool properties:**

**connection Timeout:**

It is the time interval which indicates how long a request can wait to get a connection object.

* It a request don’t get a connection object with-in this time interval, it will throw an exception called ‘connection wait time-out exception’.
* By default, connection timeout interval is 180sec.

**Maximum connection:**

It indicates up to what extent connection pool can grow.by default ‘10’ is the maximum connection.

**Minimum connection:**

It specifies the minimum number of connections to be maintained even though it is not in use .by default ‘1’ is the minimum connection.

**Reap time:**

It is the time interval which indicates where the pool maintain thread has to run across the connection pool by default 180secs is the reap time that less of the reap time more the accuracy.

**Unused timeout:**

It is the time interval which indicates, when the connection objects can be discarded from the pool. if it is not in use.by default unused timeout interval is 1800secs.

**Aged timeout:**

It is the time interval which indicates how long a request can hold a connection object(Q) the life time of a connection object with a particular request.by default aged timeout interval is ‘0’secs, which indicates these is no limits.

**Fix packs:**

Release:

It is the term which specifies the major versions of web sphere.

eg:6.0 6.1 7.0 7.1 8.0

Refresh pack:

The third digit of the version number indicate refresh pack.it contains

future additions and changes.

eg: 6.0.**1** 6.1.**2** 7.0.**1** 7.0.**2**  8.0.**1**

Fix pack:

It contains mostly mainly defect fixes, the fourth digit of the

version number identifies a fix pack.

eg: 6.0.1.**43** 6.1.2.**47** 7.0.1.**41** 7.0.2.**12** 8.0.1.**42**

Interim fix/e-fix:

It’s an emergency fix, which will be applied for particular environment for specific defects. this type of fix we called as emergency fix/e-fix.

cmd to find the version: was-root/bin>versionInfo.sh

**how to apply a fix pack/refresh pack:**

1)download and install update installer

2)download a proper fix pack which you want to install.

3)before installing fix pack, stop all the servers and backup your configuration

by using backconfig.sh cmd.

4)take a backup of response file, which is available under updateinstaller home/response files

directory(install.txt).

5)specify the values, WAS product location and fix pack location.

6)execute that modified response file by using cmd

./update -options<path of the modified response file> -silent

7)if it is success we will get a message INSTCONFSUCCESS.

**Thread Dump:**

* It contains information about Threads.
* By using thread dumps we can find the state of threads like wait on condition, blocked, hungs, suspend, dead lock and runnable threads.
* The thread dumps are more useful in debugging hung threads.
* We can Generate a Thread dumps by using this command

Kill -3 pid

**Another way:**

* Go to WAS-Root/profiles/Dmgr/bin directory
* Dmgr/bin> ./wsadmin.sh
* Set the environment by using below command.
* wsadmin>set jvm [$AdminControlCompleteObjectName type=JVM,process=server1,\*]
* wsadmin>$AdminControl invoke $jvmdumpThreads
* by default The thread dump file format is **javacore.timestamp.pid.dumpnumber.txt** files

**Heap Dump:**

* It contains information about java object/heap objects.
* We can find the size of the object, references of an object, current heap size and available heap size.
* The heap dumps are more useful in debugging memory leaks.
* we can generate a heap dump by using this command

wsadmin>$AdminControl invoke $jvm Heapdump

* after setting the parameter,

set jvm [$AdminControlCompleteObjectName type=JVM,process=server1,\*]

* by default heap dump will be generate under,

<profile-home> with a file name

* By default The Heap dump file format is **heapdump.timestamp.pid.dumpnumber.phd** file