JBoss Class Notes

Application:

Application is a collection of resources like servlets, JSP's ,EjB's, html files, xml files,Dot properties build together to fullfill the business requirement.

EjB's : Enterprise Java Beans
Applications we will deploy in Application Server

Types of Applications:
1. Web Application: It’s is a collection of resources like servlets & JSP's as the Major Components.
* If it is a web application, the application team will build it as a **.war file (web archive)**

2. Enterprise Applications: I,ts is a collection of resources like servlets & JSP's, and EJB's(session beans, entity beans, message-driven beans) as the Major Components.
* If it is Enterprise application, the application team will build it as **.ear file (enterprise archive)**

Application server:
* It provides the Run-Time environment for the application.
* It takes the responsibility to receive the end-user request, to identify a resource, to execute a resource and to generate a response to the end user.

Types of Application server: **(There are four types of Application Servers)**

1. WAS(Websphere Application server)	**--- IBM**
2. Weblogic	**-- Oracle**
3. Jboss	**--- Red hat**
4. Apache Tomcat	**--- Apache**

Introduction about Jboss Application Server:

Developed by **Marc Flury** in 1999.
Redhat acquired Jboss in June 2006 for $420 millions
http://jbossas.jboss.org/downloads

* Upto Version 7, it was Open source(free of cost) and were calling as JBOSS Application Server.

* From 7.1.1 and above it is split into ------- wildfly (open source Community Edition) stable Release (10.1.0) final
------- EAP (chargable) 7.0.0

How to install jboss, in how many ways, prerequisites?

Prerequisites: Before installation we need to fulfill the requisites

- Before installing jboss, java must be installed
- we need to set java path
- 256MB Ram , 500MB Free space.

(I) We can install jboss in 5 ways

1. **zip extraction**
2. **GUI mode (by using executable jar file)**
3. **Console mode**
4. **Automatic installation by xml file.**
5. **By using RPM file.**

1.Zip extraction:

- Download zip bundle in redhat site
- extract zipped bundle.
- Give the location for installation
- start your server
- try to access admin console **http://localhost:9990/console**
- Add user **EAP_Home/bin#./add-user.bat**
- give parameters
- (mgmt_user.properties) file updated Location: /EAP_Home/standalone/configuration

2. GUI mode:

- Execute the exactable jar file
- Edit the parameters

3. Console Mode:

- Go to your location of setup # **java -jar jboss.eap.6.3.4-installer.jar -console**
- provide the parameters
- then start your server
- Access the console **http://localhost:9990/console**

4. Automatic installation:

- go to the set up, generate auto.xml file, Take back up and edit the parameters
- execute the command **java -jar jboss.eap.6.3.4-installer.jar <path of the auto.xml>**
- provide the credentials

5. RPM file:

- **rpm –ivh setup.rpm** file

Uninstallation:

- goto EAP_Home and uninstaller directory
- give cmd **java -jar uninstaller.jar -console**

(II) Application Deployment:

Def. of Application: An application is a collection of resources like servlets, jsp's ,ejb's build together to fulfill the business requirement.

Application can be deployed in 5 ways:

1. **By using management/Admin/webConsole**
2. **By using system/Hot Deployment**
3. **By using CLI(command Line Interface)**
4. **By using Maven script.**
5. **By using Eclipse IDE(Internally DevEnv).**

1,By using management/admin/web console:

6x
Start server----- > Login to Console ------ > select deployments ------ > select add & browse---- > Select enable ------- > now access the application. By using url **http://localhost:8080/<context root>**

7x
Start server---->Login to console ----> deployments -----> Add ---- > Browse location of application ----> click on application ------ > view ---- > remove/disable/enable ---- > access the application by using url **http://localhost:8080/<context root>**

2: By using CLI (Command Line Interface):

- Start the standalone server by using the command /opt/EAP/bin # **./standalone.sh**
- Eap/bin> **./ jboss-cli.sh**
- Type : **connect**
- standalone@localhost:9999 # **deploy /path of application**.

Note: standalone@localhost:9999 # **deploy /path of application –disable**(application is deployed, but not active)

Access the application

- Refresh admin console & access the application.

To undeploy application:

- undeploy applicationname

3. By using hot deployment/ file system/ file based deployment:

- Copy the .war or .ear file directly to deployment directory under **EAP/standalone/deployment.**
- Access the application by using **http://localhost :8080/<context root>**
- *By default, every application that is placed into this folder is automatically deployed on to the server.*
- The service that scans for deployed resources is called ***deployment scanner*** and it is configured within the standalone.xml file.
- <subsystem xmlns=‘urn:jboss:domain:deployment-scanner:1.0>

<deployment-scanner name=“default” path=“deployments” scan-interval=“5000” relative-to=“jboss.server.dir”/>
</subsystem>

- Here by default the server scans in the deployment folder for every 5000ms.
- The properties that control the auto-deploy feature are auto-deploy-zipped and auto-deploy-exploded.

<u>Note:</u>

- We can find deployment / undeloyment's in log messages in EAP/standalone/log/server.log
- At the time of application deployment, standalone.xml file will get updated with a tag,

<Deployments>
<deployment name = “secondapp.war” Runtime-name = “secondapp.war”>
</Deployments>

- When an application is undeployed, an entry of application deployment under <deployments> tag is deleted.

(III) <u>Database Configuration:</u> (It can be done in three ways)

1. **Admin console**
2. **CLI mode**
3. **XML mode**

1. **Steps to configure Oracle with Jboss by using Admin console:**

   1. Login to admin console
   2. Deploy ojdbc6.jar file.
   3. Click on Configuration → subsystems → datasources → Non-XA → add
   Choose data source as **Oracle datasource**
   3. Give datasource Name : gpbabu
   Jndi Name : java:/gpbabu
   4. choose detected driver : ojdbc6.jar
   Connection-url : jdbc:oracle:thin:@localhost:1521:xe
   Username : SYSTEM
   Password: gpbabu
   5 . Click on Data source name and Test the connection.

2. **Configuring Oracle with Jboss by using Jboss CLI**

- Start Jboss server.
- Go to ,EAPhome/bin > ./jboss-cli.sh
- Connect
- **Add Modules:**

module add --name=com.oracle --resources=location of ojdbc6.jar --dependencies= javax.api,javax.transaction.api

[Result: Go to EAP home/ modules , under that com is created ---- > com/oracle/main
Ojdbc6.jar & module.xml is automatically created under main directory.]

- **Add Drivers:**

/subsystem=datasources/jdbc-driver=oracle:add(driver-name=oracle,driver-module-name=com.oracle)

[Result: <driver name="oracle" module="com.oracle"/>]

- **Add DataSource:**

```
data-source add --jndi-name=java:/jboss/ mydatasource --name= mydatasource
--connection-url= jdbc:oracle:thin:@localhost:1521:xe --driver-name=oracle --user-
name=SYSTEM --password=admin
```

- **Enable Datasource:**

  ```
  data-source enable --name=mydatasource
  ```

- **TEST JDBC CONNECTION:**

```
/subsystem=datasources/data-source= mydatasource:test-connection-in-pool
```

3. Configuring Oracle with Jboss by using XML

1. Goto **/EAP/modules/system/layers/base/com/** .. create **/oracle/main** directory
   Paste the two files i) module.jar file and ojdbc6.jar files
2. Open standalone.xml file from
   /EAP/standalone/configuration/standalone.xml
3. Find datasources

```
<datasource jndi-name="java:/jboss/datasources/gpbabu" pool-name=" gpbabu "
enabled="true">
            <connection-url>jdbc:oracle:thin:@192.168.111.1:1521:xe</connection-url>
            <driver>oracle</driver>
            <security>
               <user-name>SYSTEM</user-name>
               <password>gpbabu</password>
            </security>
         </datasource>

<drivers>
            <driver name="oracle" module="com.oracle" />
</driver>
```

4. Restart the server
5. Open the console, choose the datasrc and test the connection.

(IV) REQUEST FLOW:

1. Whenever user makes request from browser, it initially forwards to DNS and then forwards to load balancer……
2. Load balancer Distributes to webserver….we are using Apache as a webserver…when ever webserver get started httpd.conf file contains get updated.
3. Httpd.conf file contain mod-jk.so and mod-jk.conf
4. Mod_jk.so acts as a plugin b/n webserver and appserver…where as mod-jk.conf knows the path of workers.properties and uriworkermap.properties.
5. worker.properties contain node name, ajp port, host name,
6. uriworkermap. Properties contain context root information.
7. If the request contains ajp resources like serlets,jsp,ejp and session beans then request forwards to ejp container through jndi.
8. From webserver the request forwards to appserver which web container and ejb container…if the request contains web resources like serlets,jsp then request forwards to web container.
9. If the request looking for database interaction then it will forward to connection tool based on connection tool properties it will attains an connection object,once the transaction is completed the request will forwards to webcontainer through jndi and then forwards to webserver,it will forwards to user.

Difference between webserver and appserver.

SNO	WebServer	AppServer
1	webserver serves only static resources like html,image files etc.,	It can serve both static and dynamic resources like html, image, servlet, JSP, EJB's etc.,
2	In webserver we don't have any containers like web container and EJB container	Here we have two containers like web and EJB containers.
3	Here we cannot deploy both web and enterprise applications.	Here we can deploy both web and enterprise applications.
4	Here we can configure forward, reverse proxy and URL redirections.	Here we don't have configuration files like forward, reverse proxy and URL redirections.
5	We can start the webserver by using ./apachectl-k	we can start the application server by using ./standalone.sh

(V) mod-jk connectivity:

----

3 ways, to configure Webserver with Application server in jboss

a) mod.jk b)mod proxy c)mod cluster

a) modjk:

How to configure webserver(apache) with Jbossappplication server:

1. Install Application server
2. Install apache webserver
3. download and copy **mod-jk.so** modules to apache/modules directory.
4.copy three files in apache/conf directory. those are
a)mod-jk.conf b)workers.properties c)uriworkermap.properties
5. goto **apache/conf/httpd.conf file** and add modules
LoadModule jk_module modules/mod_jk.so
Include conf/mod-jk.conf
6. Specify workers.properties location in mod-jk.conf
7. In workers.properties file specify workers list, jboss server information like 'hostname', 'ajp connection port number' , 'lbfactor' , protocal
8. In uriworkermap.properties, we can find context root information.
9. If any connectivity issues, there is no response from webserver, we can find log details in logs/mod_jk.log.

(VI) <u>Profiles in jboss:</u>

1. **default[standalone.xml]:** this profile supports javaeewebprofile plus some extensions like RESTFUL webservices and ejb remote invocations.

2.**full[standalone-full]:** supports jee full profile and all server capabilities without clustering.

3.**ha[standalone-ha]**:supports default profile with clustering capabilities.

4.**full-ha[standalone-full-ha]**:supports full profile with clustering capabilities.

(**VII) Clustering:**

Definition: It is a group of servers used to achieve **single point of management, fail over, workload management and High availability.** There are 2 types of clusters,

a. Vertical Clustering
b. Horizantal clustering.

a) Vertical Clustering :

Here we are creating the cluster members in same box/machine. If anyone of the cluster member is down, requests will be routed to another cluster member.

Suppose, if machine is completely crashed, we won't access the applications from any of the cluster members.

So, it is not advisable for critical applications in production environment.

b) Horizontal clustering:

Here we are creating cluster members in different boxes/machines.

If anyone of the cluster member is down or the machine is completely crashed at that time also we can access the application from another cluster member

So, this type of clustering is advisable for production environment because here we can achieve high availability.

In clustering work load management is done by using Round Robin Algorithm:

cluster members	**Weight**					
M A ----	5	4	3	2	1	0
S B ----	4	3	2	1	0	
R c -----	4	3	2	1	0	

<u>How to create vertical clustering in standalone mode:</u>

1. we can create any number of instances by copying standalone directory & pasting in the same location
2. Rename the copied standalone instances as per our requirements
   a) standalone-node1
   b) standalone-node2
3. for standalone-node1: under application server level, open the standalone-ha.xml file & modify the server name
<server **name="node1"** xmlns="urn:jboss:domain:4.1>
   for standalone-node2: under application server level, open the standalone-ha.xml file & modify the server name
<server **name="node2**" xmlns="urn:jboss:domain:4.1>
4. under webserver level **E\apache\conf\workers.properties**
go to workers.properties file & change few parameters

```
worker.list=failover

    #define node1
    workers.node1.port=8009
    workers.node1.host=localhost
    workers.node1.type=ajp13
    workers.node1.ping_mode=A
    workers.node1.lbfactor=1

    #define node2
    workers.node2.port=8109
    workers.node2.host=localhost
    workers.node2.type=ajp13
    workers.node1.ping_mode=A
    workers.node1.lbfactor=1

 # load-balancing behaviour
worker.loadbalancer.type=lb
worker.loadbalancer.balance-worker=node1,node2
worker.loadbalancer.sticky-session=false
worker.liot=load balancer
```

5. save the changes & restart the server
6. start apache server &jboss instances
7.To start jboss instances with default port-offset=0

```
./standalone.sh -c standalone-ha.xml -Djboss.server.base.dir=\opt\EAP\standalone-node1
```

to start the second instances with port-offset=100

```
./standalone.sh -c standalone-ha.xml -Djboss.server.base.dir=\opt\EAP\standalone-node2
-Djboss.socket.binding.port-offset=100
```

NOTE:

Clustering can be configured only under full-ha & ha-profiles. and under these two profiles by default we have ajp connectors.

But for default & full profile ,we have to provide manually under xml files......

(VIII) Difference between domain and standalone:

SNo.	Domain mode	Standalone Mode
1	In domain mode we can achieve single point management.	we can't achieve single point management.
2	There are 2 XML files for configuration, domain.xml & host.xml	We have one xml file here, standalone.xml
3	We don't have deployments directory for hot deployment of application	We have deployments directory for hot deployment of application
4	For all profiles configuration, we will configure under domain.xml & host.xml	Here each profile has its own xml file.
5	Here we have process controller, host controller, & domain controller for communicating and controlling of servers	Here we don't have any controllers.
6	Here all servers are considered as a part of one domain. There is only one central configured repository for the whole domain	Each profile has different config files. Here we don't have central config repository for all the clusters.
7	JMS can be configured under full & full-ha profiles	JMS can't be configured.

DOMAIN MODE

When we start server in domain mode, 3 java processors started

a) Process controller ---- maintain the configuration file of instances that are created.
b) Host controller---manages the process control and starts the services of servers
c) Domain Controller

Application Deployment in Domain mode:

1) By using command line Interface
2) Using the Admin console

1)By using command line interface:- **(done two ways)**

a) Deploy to all server groups
b) Deploy to a single server group.

a) Deploy to all server groups:
goto EAP/bin/ **./ jboss-cli.sh**
connect
deploy location of war/ear file **--all-server-groups**

b) Deploy to a single server group:
goto /EAP/bin? **./ jboss-cli.sh**
connect
deploy location of war/ear file **--server-groups=group name**

Undeployment:

undeploy name of the war/ear file **--server-groups= group name**

From the process point of view, a domain is made up of four elements:

1. **Domain controller**: The domain controller is the management control point of your domain. An AS instance running in domain mode will have at most one process instance acting as a domain controller. The domain controller holds a centralized configuration, which is shared by the node instances belonging to the domain.

2. **Host controller**: It's a process that is responsible for coordinating with a domain controller the lifecycle of server processes and the distribution of deployments, from the domain controller to the server instances.

3. **Process controller**: It's a very lightweight process whose primary function is to spawn server processes and host controller processes, and manage their input/output streams. This also allows the host controller to be patched and restarted without impacting the associated servers.

4. **Application server nodes**: These are regular Java processes that map to instances of the application server. Each server node, in turn, belongs to a domain group. Domain groups are explained in detail when we discuss the domain configuration file.

The Java Message Service(**JMS**) API is a Java Message Oriented Middleware API for sending messages between two or more clients. It is an implementation to handle the Producer-consumer problem

host (in computing)

A host (also known as "network host") is a computer or other device that communicates with other hosts on a network. Hosts on a network include clients and servers -- that send or receive data, services or applications.

(IX) TYPES OF LOGS

1 . Server.log
2 . Host-controller.log
3 . Process-controller.log
4 . Installation.log
5 . GC.log

1 .**Server.log** : The booting information of a server regarding the modules, classes, we can find in this log file. If there is any issue while starting a server, deploying, an application, configuring db, & port conflicts, based on the log messages we can understand what is the problem & we can start debugging the errors as per the log messages. This log is located under **EAP_HOME/Standalone/log-directory** in case of standalone mode. **EAP_HOME/Domain/servers/server-name/log**-directory in case of domain-mode.

2 . **Host-controller.log** : It will update at the time of starting host controller & also we can find what are the servers that are starting under server groups. eg :- starting server server-one, starting server server-two. Regratering server server-one.

3 . **Process-controller.log** : At the time of starting process controller, it will get updated & it is located under **EAP_HOME/domain/log**.

4 .**Installation.log** : At the time of jboss installation, it will be updated with the log messages like http port, ajp port, jboss installation location, multicast IP & is located under **EAP/installation/installation.log.** At the time of jboss installation if there are any errors we will look into it.

5 **.GC.log** : It contains garbage collector information, edenspace, tenurespace, initial heap size, max heap size, located under **EAP_HOME/standalone logs** and **EAP_HOME/domain/servers/server-name/log**

(X) Thread Dumps and HeapDumps :

Thread Dumps:

It contains the information about the threads, we can find the status of threads like running, deadlock, wait-on condition, blocked and hang suspect.
Thread dumps are most useful in **debugging Hung threads.**
We can generate thread dumps as follows,

i) find pid of the server (to know pid : **pgrep -f server name**)
ii) Generate thread dump :
goto **java-home/bin : jstack -i pid >> give location where to generate thread dump**
We can analyze these thread dumps by using thread dump analyzer (gca455.jar).

Heap Dumps:
It contains information about heap object or java object.
We can find the status of objects like size of the object, references for the object and relationship between the object. Heap dumps are most useful in **debugging in memory leaks**.

We can generate heapdumps as follows,

i) find the Pid of server (to know pid : **pgrep -f servername**)
ii) goto **java-home/bin : jmap –dump:live,format=b,file=loc. of dumpfile.txt pid**

(XI) <u>HEAP SIZE:</u>

a) How to Increase the heap size in standalone mode:

Goto **/EAP-home/bin/** edit standalone.conf file (vi standalone.conf)

We can change the heap size in standalone.conf

In Java VM section

```
[#
# Specify options to pass to the Java VM.
#
if [ "x$JAVA_OPTS" = "x" ]; then
   JAVA_OPTS="-Xms1303m -Xmx1303m -XX:MaxPermSize=256m -Djava.net.preferIPv4Stack=true"]
```

Min: 128MB max: 512MB

```
set JAVA_OPTS=%JAVA_OPTS% -Xms128m -Xmx512m
```

{ http://middlewaretechadmin.blogspot.in/ 2015/11/jvm-set-tings-in-jboss-managed-domain.html } in domain mode for change the heap size information avilble in this link

b) How to increase the heap size in domain mode:

In domain mode we can change jvm heap size in 3 ways
1.Host Level: In this case, The configuration will apply to all servers that are defined in host.xml
2.Server Group Level: In this case, The configuration will apply to all servers that are part of the group
3.Server level: In this case, The Configuration is used just for the single host.

<u>Cluster Building Blocks:</u>

1. **Jgroups**: It is a subsystem which is used for the underlying communication between nodes.
2. **Infinispan** : Subsystem which handles the cluster consistency using its advanced data grid platform.
3. **Messaging Subsystem:** uses the **HornetQclusterable** implementation.

<u>Multicasting:</u>

It is a mechanism, where the servers will inform, the other process that they are alive, and will communicate with each other by using heart beat messages.

If one jboss instance didn’t communicate with other jboss instance, it assumes that the instance is down, that instance will send some heartbeat message to apache.

(XII) jboss Patching:

Patching can be done in 2 modes.

a) CLI Mode
b) Console mode

a) CLI Mode:

1. Download the patch file.
2. Check the version by using,
standalone.bat -V
3. Start the Server
4. Connect to CLI by using, jboss-cli.bat
5.Give the command,
patch history ---- to check whether the patch is applied earlier or not. If not applied then,
6.Give the command,
patch apply "path of patch"
if Outcome=success then
7. **shutdown --restart=true**
8.Now check **patch history**
note down the patch ID
9.If we want to rollback, give command

patch rollback --patch-id= patch ID --reset-configuration=TRUE
then outcome =success
10. **Shutdown --restart=true**
11. After that check the version again to know whether the patch is roll back or not

b) Console Mode:

1. Login to console ---> goto patching ---->apply a new patch----> choose a file--->open ---->next
2.Goto patching again, we will get latest patch applied
3.goto rollback--->select patch---> rollback
4. select reset configuration and override all----> next--->dismiss--->finish
5. In console at bottom we can see the version.

Linux Basic Commands:

To know username : type **whoami**
To root user : type **su root**
To know present working Directory location: type **pwd**
To know list of files :**ls - lrt** (**ls**- listing ,**lrt** - long list format)
To come out from directory : type **cd .**
To come out from home : type **cd ..**
To go to destination folder ex home : type **cd home**
To create directory: type **mkdir <name of the directory>**
To remove Directory : type **rm –rf <name of directory>**
to create empty file : **touch <name of the file>**
To copy file : type **cp <name of file to be copied> <path of the destination>**

Tools :

Ticketing Tool : BMC remedy / Service now
Incident tickets:
Service request(RITM)
Problem ticket : RCA
change request : ANY CHANGES IN ENV ----- INC HEAP SIZE

==

SLA : (Service Level Agrrement)

P0 - 2 HRS
P1 - 4 HRS
P2 - 1DAY
P3 - 3 DAYS
P4 - 5 DAYS

ENVIRONMENTS --- DEV QA Stage Prod

Team Size: 7 24/5
non business hours project -
bangalore - 5 resources
mumbai - 2 resources
us- 6 resources

rotational shift
11.00 -5.00am = in us there is another team
that team joins
shift = 6 - 3.30pm
1.00 - 11.00pm
mon - fri
rest of two day - on call - priority tickets p0,p1,p2

SLA (Service Level Agreement)

In my environment we have separate monitoring team, they will raise the ticket by using BMC remedy tool

1. Whenever they will raise the ticket, they will specify the security of the ticket and i will get a test to my mail & mobile.
2. Based upon the severity, I have to respond and resolve it.
3. We have SLA that, for serv1 ticket I have to respond immediately and resolve it within one hour
4. For serv2 ticket I have to resolve it within 8 hours
5. For serv3 ticket I have to resolve it within 8 hours
6. Once after resolving ticket, I will send a mail that everything has been resolved to ticket raised person and entire team.
7. After getting signoff mail from client, I will close the ticket
8. Suppose if i am unable to resolve the ticket with in the SLA period, we have an escalation procedure that immediately we have to interact with onsite supporting team and i have to explain what is the issue, what are the errors still we are getting, why it has not been resolved. I will ask them to join in call, sometimes client also will join in the call, I will explain the problem, i will take help from onsite supporting team. Issue will be resolved.
9. Suppose there are some issues both onsite & offsite support team cannot resolve the issue then we will raise PMR (Problem Management Record) with IBM. This is our process (or) SLA
10. In general, we won't escalate the ticket.

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. DB configuration
10. Dumps
11. Clustering
12. Logs

1. WAS Installation:

Prerequisites:

- ➢ Before installing WAS check Prerequisites
- ➢ 2GB Disk Space
- ➢ 1GB RAM
- ➢ 1GB free space in temp directory
- ➢ File permissions
- ➢ Compatibility
- ➢ Installation Manager

  - ✓ 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

2. PROFILE CREATIONS: --

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
8). Liberty Profile

Note:

Was-root-----→profiles----→Appsrv01----→cell---→node---→server---→applications

Appserver Profile: server1, admin console, node
Deployment manger- DMGR- cell, node, admin console
Appsrv01
Dmgr01

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>./manageprofiles.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>

3. Federation:

Federation is the process adding a node either from Appserver profile (or) custom 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 federation, 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".

4. 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, if you don't specify any location, by default application will be deployed under **<profile-home>**/installedapps directory, whether it contains web services and EJB's.
5. Specify the target under which server(Q) which cluster that application had to be deployed.
6. Map the data sources and EJB's with JNDI
7. 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.

5. Apache configuration:

1. Install appserver
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 appserver 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.

WAS 8.5 IHS Configurations

1. UNZIP IHS.zip

unzip IHS.zip

2. After unzipping **IHS_Extracted** directory is generated.

i) cd Disk-1

ii) open diskTag.inf ,   #vi diskTag.inf

then copy diskSetID

iii) note the path of repository.config [/IHS_Extracted/repository.config]

3. Goto /opt/IBM/IM/eclipse/tools and give the command to install IHS

./imcl install <diskSetID> **-repositories** ../.. /IHS_Extracted/repository.config **-installationDirectory** /opt/IBM/IHS/ **-acceptLicense –properties** user.ihs.httpPort=80 **–showProgress**

4. go to /opt/IBM/IHS/bin

Now start IHS server by giving cmd ./apachectl –k start

5. open the console type url http://localhost:80/

    home of IHS is displayed.

Installation of Plugin (creates bridge between IHS and AppServer)

1. Unzip Plugins.zip

```
unzip Plugins.zip
```

2. After unzipping **Plugins_Extracted** directory is generated.

    i) cd Disk-1

    ii) open diskTag.inf ,   #vi diskTag.inf

    then copy diskSetID

    iii) note the path of repository.config [/Plugins_Extracted/repository.config]

3. Goto /opt/IBM/IM/eclipse/tools and give the command to install Plugins

```
./imcl install <diskSetID> -repositories ../../Plugins_Extracted/repository.config –installationDirectory /opt/IBM/IHS/  -acceptLicense   –showProgress
```

Generate plugin-cfg.xml

1. Goto **Dmgr/bin** directory, give CMD

```
./GenPluginCfg.sh
```

Now **plugin-cfg.xml** file is generated in /opt/IBM/WAS/profiles/Dmgr01/config/cells/

2. Copy **plugin-cfg.xml** from the above location to /opt/IBM/Plugin/config/

```
cp /opt/IBM/WAS/profiles/Dmgr01/config/cells/plugin-cfg.xml  /opt/IBM/Plugin/config/
```

Load Module in **/opt/IBM/IHS/conf/httpd.conf** file.

Open the file httpd.conf, go to end of the file

```
LoadModule  was_ap22_module  /opt/IBM/Plugin/bin/mod_ap22_http.so
WebspherePluginConfig   /opt/IBM/Plugin/config/plugin-cfg.xml
```

Save and Quit

Check for syntax and modules loading

```
./apachectl –t
./apachectl -M
```

3. go to **/opt/IBM/Plugin/config/**

Edit plugin-cfg.xml file

Change the default path to /opt/IBM/Plugin in all the lines of code

Finally stop and start the IHS server (Webserver) and AppServers.

Check the application is started. If not started, then start the application.

Open the admin console access the application http://localhost:9060/ibm/console

Open the browser and type url to access the application from AppServer http://localhost:9080/snoop

Open the browser and type url to access from the HIS webserver http://localhost:80/snoop.

6.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 loads 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 ext directories

3. Web sphere server class loaders:

It loads the jar files which are under shared libraries.

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 module 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.

7. 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.

8. Fix packs:

Release:

It is the term which specifies the major versions of websphere.
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**

CMD to take back for all configuration files: was-root/profiles/AppSer/config>./**backconfig.sh -nostop**

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 backupconfig.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.

Difference between Connection Pool Data Source and XA Source

Connection Pool Data Source	XA Data Source
Supports Single phase commit	Supports Double phase commit
Local Transactions	Global Transactions
In case of connection pool DS, once the transaction is completed, immediately that DB will be committed. It won't wait for its dependencies.	In case of XA DS, once the transaction is completed with particular DB, it won't commit immediately until and unless all the dependence DB transactions are successful.

9. DB Configuration:

1. Create JDBC providers
2. Create a Data Source
3. Test the connection

1. JDBC Provider:

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

2.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)
Select the Jdbc Provider
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
```

8. Connection pool:

It contains pre-defined connection objects, server need not 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. If a request doesn'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 maintains 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.

10. 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 [$AdminControl completeObjectName type=JVM,process=server1,*]
- wsadmin>$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 $jvmHeapDump
```

- after setting the parameter,

```
set jvm [$AdminControl completeObjectName 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

11. Clustering

Group/collection of servers to achieve single point of Management; fail over, workload management and high availability.

Two types: horizontal and vertical clustering

Procedure for creating vertical clustering

1. login in to admin console
2. expand servers, expand clusters, expand Websphere Enterprise server clusters
3. click New,
   i)Enter cluster name : cluster1
   ii)Enter cluster member name: server11
   iii)Enter the weight : 6 (0 to 100)
   Click next.
   Continue, how many cluster members you want to create by adding Members.
4. Configuration for cluster members
   i) click on cluster name
   ii) click on cluster members (We can find all the cluster members) and check/note the default portnos are registered in virtual Host and hot aliases.
   iii) install the application, select the cluster1 for deploying.
5. Access the application by giving the port nos of cluster members.
6. To access application from the webserver(IHS), we need to generate plugin from <Dmgr home>/bin directory.

i). copy the newly generated plugin-cfg.xml from Dmgr/config/cells/ to /opt/IBM/Plugin/config/plugin-cfg.xml

ii) edit the plugin-cfg.xml from /opt/IBM/IHS/config/
change the default path to /opt/IBM/Plugin in all the lines of code.

[**Note**: you see the next port nos. are reflected in the new plugin-cfg.xml]

iii) Now open the browser type url : **http://localhost:80/snoop,** now you can access the application.

12. Logs

1. JVM Logs (There are two types in JVM logs)

i) Systemout.log: This file contains information about server services.
At the time of starting server, we can check
--what are the services that are started (like work manager service, SIB-Service Integration Bus service, SOAP container service, scheduler service),
--what are the applications that are started
-- what are the jar files that are uploaded
-- what are the modules that are uploaded
At the time of Application start up, if the configuration and any server issues, I will check this Systemout.log file to troubleshoot.

ii) SystemErr.log:

If the server is not started , application is not started then the error messages will be displayed in SystemErr.log file. It contains only error messages but not server service messages and application startup messages.

<u>**Note:**</u> ** These log files are located under **/<profile-home>/logs/<servername>**

2. Process.log or Native logs (There are two types)

i) Native_stdout.log : This file contains information about .dll, .exe, .ios files information. We can also find any of the above files are missing.

ii) Native_stderr.log: This file contains garbage collector information. This log file will get updated when we enable verbose garbage collector from the console.
Path : console →servers→Was Enterprise server→ servername→ Java Process Management→Process Definition→ JVM→ enable (tickmark) verbose Garbage Collector

3. Trace.log

This file contains detailed information about each and every activity of particular server.
This log file will be helpful to pinpoint a failure
This log is available under **/<profile-hme>/logs/<servername>**

4. Activity.log / Service.log

There is only one Activity.log file for entire profile. It is available under **/<profile-home>/logs/**

- It contains all server information under a particular profile.
- It can be analyzed by using Analyzing tool called as log analyzer.
- Log analyzer compares the servers log error messages with a symptom database for the known problems.

5. **Command Line Logs:** There are five types of cmd logs

i) **addNode.log** : This log get updated when we give CMD at console as <profile-home>/IBM/WAS/profiles/<serverName>/bin/ **./addNode.sh**
i.e., this log is updated when we do federation process.

-- This is file available under **/<profilehome>/logs/**

ii) **syncNode.log:** It will get updated at the time of synchronization of node whenever we execute command **./syncnode.sh** .

-- it contains synchronization related messages, whether it is properly synchronized or not or any error messages while synchronization.

-- This is file available under **/<profilehome>/logs/**

iii) **serverStatus.log:** It will get updated when we give command **./serverStatus.sh**

-- it contains server status, what are servers/ clusters members are started or stopped.

-- This log file is available under **/<profilehome>/logs/**

-- There will be only one serverStatus.log file for entire profile.

iv) **startServer.log:** It will get updated when we give command **./serverStart.sh**

- It will be updated at the time of server starting.
- This log file is available under **/<profilehome>/logs/**
- This file contains the PID of the server.

v) **stopServer.log** : It will get updated when we give command **./stopServer.sh**

-- It contains server stopping information

-- This log file is available under **/<profilehome>/logs/**

6. PostInsall.log :

-- This log will get updated under **<WAS-home>/logs/Post Install/**postinstall.log

-- This log file contains information related server starting, server installation success.

message as "INSTCONFSUCCUSS"

7. **Profile Creation logs:**

--This log is updated when profiles are created succuessfully.

-- This log file is available under **<WAS-home>/logs/manage Profiles/profileName_Create**.log

-- here it also contain /**profileName_delete.log (**get updated when we delete the profile**)**

8. **Fix Pack Logs:**

-- At the time of applying Fix and refreshing the Packs this log (**updatelog.txt**) will get updated and it is available under **/WAS-home/update/fixpack directory/updatelog.txt**

-- If there are any errors, while applying the packs, refresh packs, I will check this log file and start troubleshoot.

9. **FFDC logs:** (First Failure Data Capture)

-- This log file contains runtime events of a server.

-- This log file is available under <profile-home>/logs/FFDC/

-- This log file also contains information about failures and data capture.

10. **IHS Logs** : There are four types of IHS logs.

i) **Access.log**:

-- It contains request information like IP address, time stamp, status code, get/post message, no. of request bytes.

-- It will get updated whenever a request is reached to webserver.

-- It is available under **/IHS-home/logs/**

ii) **Err.log:**

-- It contains error messages while starting/stopping the webserver.

-- It is available under **/IHS-home/logs/**

iii) **Admin-Err.log:**

--This log file will get update whenever they are errors with IHS admin service.

---- It is available under **/IHS-home/logs/**

iv) **log.txt**:

-- At the time of IHS installation, log.txt file will get updated. If it is success then this file contains “INSTCONFSUCCESS” message at the end of the file.

11. **Plugin.log (**There are two types of logs**)**

i) HTTP_plugin.log:

-- It available under **/Pluginhome/logs/**

-- It enable tracing for Plugins, we can trace how the request is mapping to a server, where the request is getting failed and additional routing information.

ii) **logs.txt:**

-- It available under **/pluginhome/logs/log.txt**

-- At the time of Plugin installation, this log.txt file will get updated. We can confirm the Plugin is successfully installed by seeing this file. Inside this file , at last we can find “INSTCONFSUCCESS” message.

IMCL COMMANDS

<u>IMCL Command the Installation Manager:-</u>

./imcl install <offering id> -repositories <path of repository.config> –installationDirectory<Installation Path> -acceptLicense   -showProgress.

<u>Offeringid</u>:- Inside extracted IM folders inside repository.xml

Repository.config:- Inside extracted IM folders inside repository.config.

<u>HOW TO UNINSTALL THE IM:-</u>

Go to the var/IBM/installationmanager/uninstall /./uninstallC

<u>IMCL Command the installation of WAS:-</u>

./imcl install <diskset id> -repositories <path of repository.config> -installationDirectory<installationPath> -acceptLicense -showProgress.

<u>Disksetid</u>: Disk1

<u>Repository.config</u>:-comes with unzip of 3 files.

<u>IMCL Command the installation of IHS:-</u>

./imcl install <disksetid> -repositories <path of repository.config> -installationDirectory<installationPath> -acceptLicense -properties user.ihs.httpPort=80 -showProgress.

<u>Disksetid</u>:-Inside extracted IHS folders Disk1/diskstag.id.

<u>IMCL Command the installation of plugin:-</u>

./imcl install <disksetid> -repositories <path of repository.config> -installationDirectory<installationPath> -acceptLicense -showProgress.

<u>Disksetid</u>:- Inside extracted plugin folder Disk1/disktag.id

<u>IMCL COMMAND   UNINSTALL THE WAS:-</u>

Go to the <IM HOME >/eclipse/tools

First to check the list of packages   ./imcl listInstalledPackages

Imcl command uninstall the plugin:-   ./imcl uninstall <pluginid> -sP

Imcl command uninstall the IHS:-   ./imcl uninstall <IHS ID> -sP

Imcl command uninstall the WAS:-   ./imcl uninstall <WAS ID> -sP

Imcl command uninstall the IM:-   ./imcl uninstall <IM ID> -sP