**WEB SPHERE**

**DMGR\_port\_No (8879)**

**DMGR:**./startManager.sh

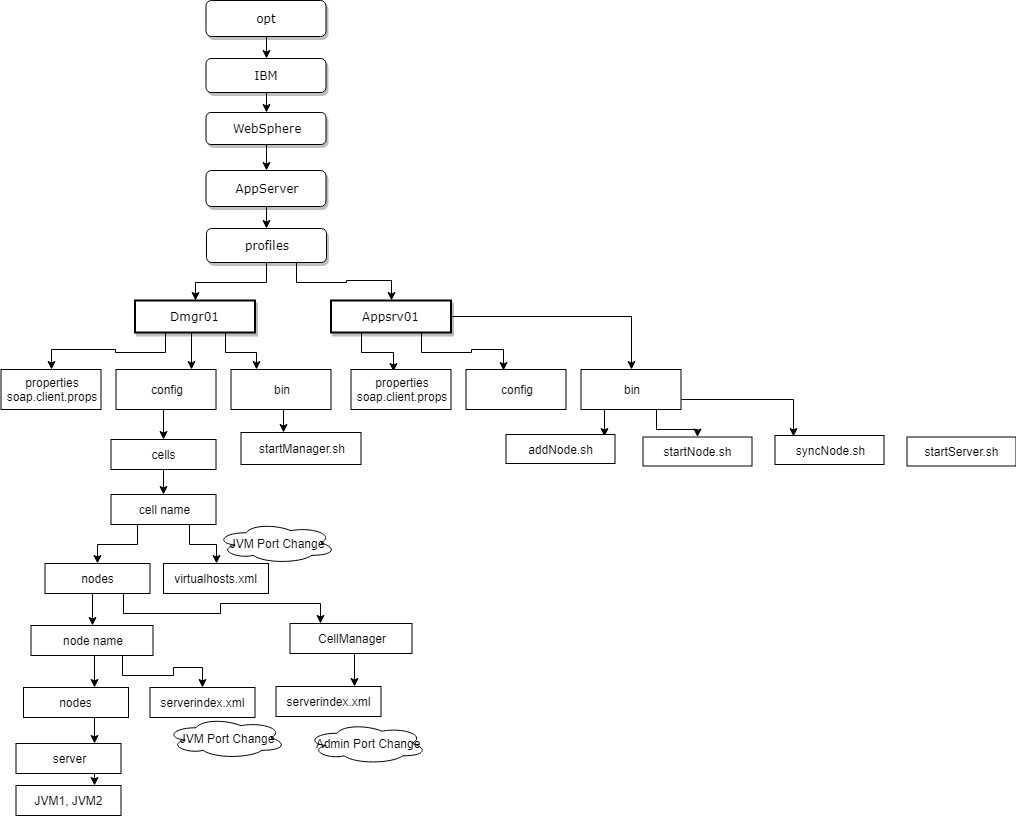
**Federate Node:**./addNode.shDMGR\_HostNameDMGR\_port\_No

**NODE AGENT:** ./startNode.sh

**Start JVM:** ./startServer.sh JVMName

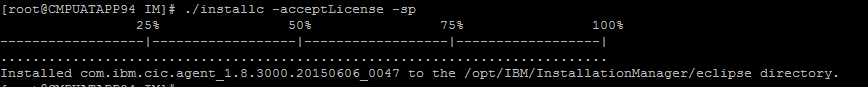
**Sync Node:** ./syncNode.sh DMGR\_HostNameDMGR\_port\_No

**Check Server DMGR Running:** ./serverStatus.sh -all

****

**IM (Installation Manager) installation:**

**[root@CMPUATAPP94 IM]# unzip agent.installer.linux.gtk.x86\_64\_1.8.3000.20150606\_0047.zip& insall**

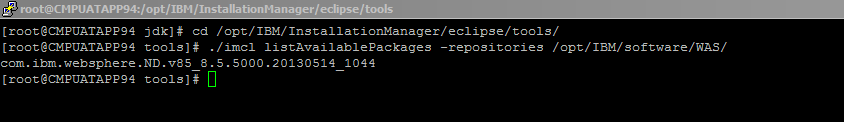
****

**Method 1:**

**Navigate /opt/IBM/InstallationManager/eclipse/tools**

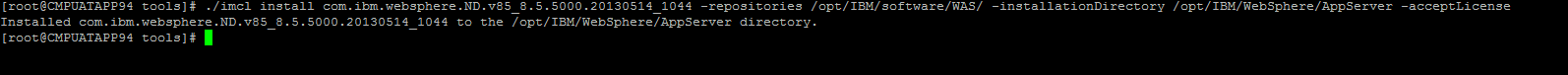
**List WAS ND Packages**

./imcllistAvailablePackages -repositories /opt/IBM/software/WAS/



**INSTALL WAS ND Packages**

./imclinstall com.ibm.websphere.ND.v85\_8.5.5000.20130514\_1044 -repositories /opt/IBM/software/WAS/ -installationDirectory /opt/IBM/WebSphere/AppServer -acceptLicense -sp



**Method 2:**

Note: we can install the WAS ND or FIX packs using repository & secure storage file

1. create a secure storage file

imutilsc saveCredential -secureStorageFile storage\_file -userName sudhakar\_ande@satyam.com -userPassword qweasd123 -url <https://www-147.ibm.com/software/repositorymanager/com.ibm.websphere.ND>

1. ./imcl listAvailablePackages -secureStorageFile storage\_file -repositories <https://www-147.ibm.com/software/repositorymanager/com.ibm.websphere.ND>
2. ./imcl install com.ibm.websphere.liberty.ND.v85\_8.5.190012.20191120\_0300 -secureStorageFile storage\_file -repositories https://www-147.ibm.com/software/repositorymanager/com.ibm.websphere.ND -installationDirectory /opt/IBM/WebSphere/AppServer -acceptLicense -sp

🡪 Checking Version of installed WAS

a) /opt/IBM/InstallationManager/eclipse/tools

./imcl -c

b) /opt/IBM/WebSphere/AppServer/bin

./versionInfo.sh

**FixpackListing & Installing:**

**Note:** Make sure no java process is running before fix pack updation (Dmgr, nodeagent, Appserver is in stop condition)

**Navigate to /opt/IBM/InstallationManager/eclipse/tools**

./imcllistAvailablePackages -repositories /opt/IBM/software/fixpack/

./imcl install com.ibm.websphere.ND.v85\_8.5.5011.20161206\_1434 -repositories /opt/IBM/software/fixpack/ -installationDirectory /opt/IBM/WebSphere/AppServer/ -acceptLicense -sp



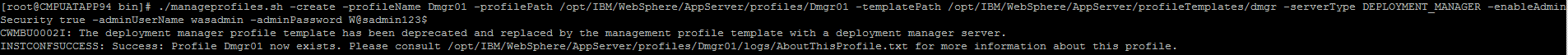
**\*\*\* Change ownership to normal user**

**Create the Profiles**

**Navigate to /opt/IBM/WebSphere/AppServer/bin**

**Create Deployment Manager Profile**

./manageprofiles.sh -create -profileName Dmgr01 -profilePath /opt/IBM/WebSphere/AppServer/profiles/Dmgr01 -templatePath /opt/IBM/WebSphere/AppServer/profileTemplates/dmgr -serverType DEPLOYMENT\_MANAGER -enableAdminSecurity true –adminUserNamewasadmin -adminPassword W@sadmin123$



**Creating Application Server Profile**

./manageprofiles.sh -create -profileName AppSrv01 -profilePath /opt/IBM/WebSphere/AppServer/profiles/AppSrv01 -templatePath /opt/IBM/WebSphere/AppServer/profileTemplates/default



**\*\*\*Changing Default port no if needed at**

**Changing Default port no of Admin Server**

/opt/IBM/WebSphere/AppServer/profiles/Dmgr01/config/cells/IFAMS-APP02Cell01/nodes/IFAMS-APP02CellManager01/serverindex.xml

**Changing Default port no of JVMS**

/opt/IBM/WebSphere/AppServer/profiles/Dmgr01/config/cells/IFAMS-APP02Cell01/nodes/IFAMS-APP02Node01/serverindex.xml

**STARTING SERVICES**

**Starting DMGR& Checking Status**

Start DMGR

Cd /opt/IBM/profiles/DMGR01/bin

./startManger.sh

./serverStatus.sh -all

**Federate Nodes to DMGR:**

Go to: /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/bin

./addNode.shDMGR\_HostNameDMGR\_port\_No (8879)

/opt/IBM/WebSphere/AppServer/profiles/Dmgr01/config/cells/IFAMS-APP02Cell01/virtualhost.xml

**Start JVM**

./startServer.sh JVMName

🡪 Like boot.properties we have,

/opt/IBM/WebSphere/AppServer/profiles/Dmgr01/properties/soap.client.props

/opt/IBM/WebSphere/AppServer/profiles/Apsrv01/properties/soap.client.props

**JDK**

Before Installing Java V1.8 we have to take a backup configuration

🡪DMGR Backup (/opt/IBM/WebSphere/AppServer/profiles/Dmgr01/bin/)

./backupConfig.sh –nostop

🡪Take a backup of /opt/Backup

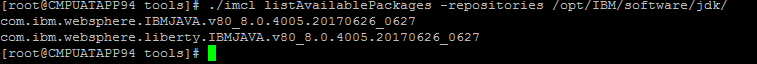
🡪Take a backup of jar files (local\_policy.jar & US\_export\_policy.jar) from /opt/IBM/WebSphere/AppServer/java/jre/lib/security. And move backup jar’s to /tmp

🡪stop the services what are the running JVM’s ( including Node agent & DMGR)

Unzip JDK binaries, List & Install JDK

Navigate to /opt/IBM/InstallationManager/eclipse/tools/

./imcllistAvailablePackages -repositories /opt/IBM/software/jdk/

****

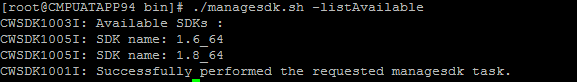
./imcl install com.ibm.websphere.IBMJAVA.v80\_8.0.4005.20170626\_0627 -repositories /opt/IBM/software/jdk/ -installationDirectory /opt/IBM/WebSphere/AppServer/ -acceptLicense



🡪Then we have to check the list of available java SDK

opt/IBM/WebSphere/AppServer/bin

./managesdk.sh -listAvailable



After checking list of available java SDK’s then we have to enable java SDK at DMGR level by using below command in below path

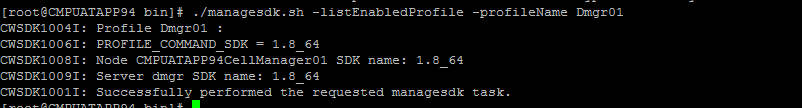
opt/IBM/WebSphere/AppServer/bin

./managesdk.sh -enableProfile -sdkName 1.8\_64 -profileName Dmgr01



We need to cross verify that SDK enabled or not at DMGR level

./managesdk.sh -listEnabledProfile -profileName Dmgr01



After enable SDK we sync the Node with DMGR

./syncNode.sh CMPUATAPP94 8879

Now we are enable SDK at each JVM (under which node) level at what are federated nodes to DMGR

opt/IBM/WebSphere/AppServer/bin

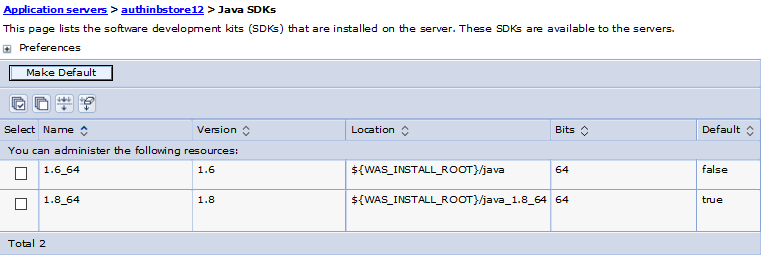
./managesdk.sh -enableProfile -sdkName 1.8\_64 -profileName Appsrv01 -enableServers -user wasadmin -password W@sadmin123$



**Note:-**Otherwise we are enable java v1.8 at each JVM from Console also but take time enable.

click on Application server🡪<JVM Name>🡪 Java SDKs

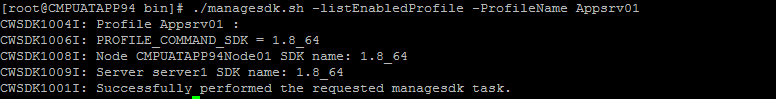
then select latest java V1.8 then click make as default.



We need to cross verify that SDK enabled or not at each JVM’s level.

in opt/IBM/WebSphere/AppServer/bin

./managesdk.sh -listEnabledProfile -ProfileName Appsrv01



Again enable SDK we sync the Node with DMGR

./syncNode.sh CMPUATAPP94 8879

**-------------------------------------------------------------------------------------------------------------------------**

**IHS Installation & Configuration:**

Required binaries for IM :

InstalMgr

IHS Binaries :

WAS\_V8.5.5\_SUPPL\_1\_OF\_3.zip

WAS\_V8.5.5\_SUPPL\_2\_OF\_3.zip

WAS\_V8.5.5\_SUPPL\_3\_OF\_3.zip

IHS Fix pack Binaries:

8.5.5-WS-WASSupplements-FP011-part3.zip

8.5.5-WS-WASSupplements-FP011-part2.zip

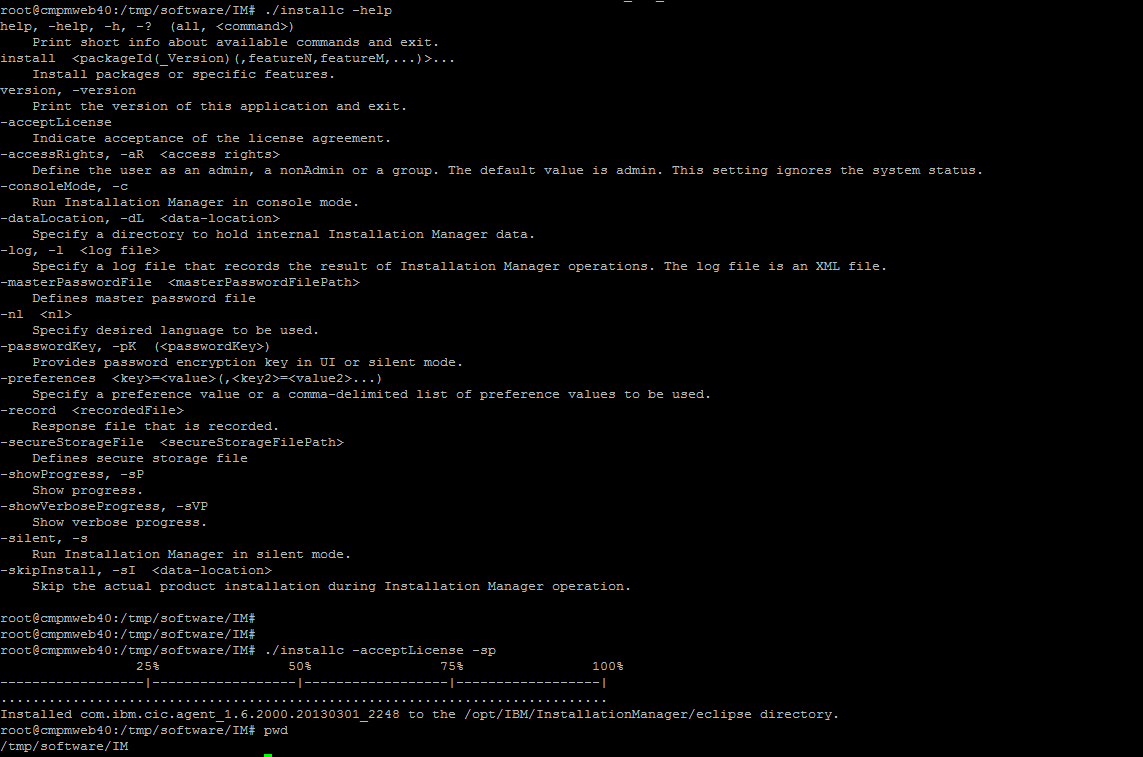
8.5.5-WS-WASSupplements-FP011-part1.zip

**IM Installation:**

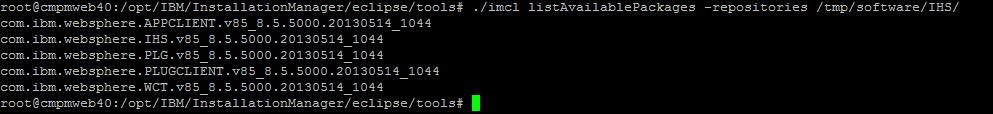
agent.installer.linux.gtk.x86\_64\_1.8.3000.20150606\_0047.zip

Unzip the installtion manager file by using the below command:



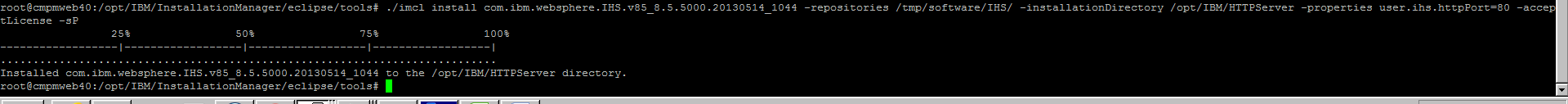


To see list available packages of IHS



for IHS installation use the below command:

./imcl install com.ibm.websphere.IHS.v85\_8.5.5000.20130514\_1044 -repositories /tmp/software/IHS/ -installationDirectory /opt/IBM/HTTPServer -properties user.ihs.httpPort=80 -acceptLicense -sP



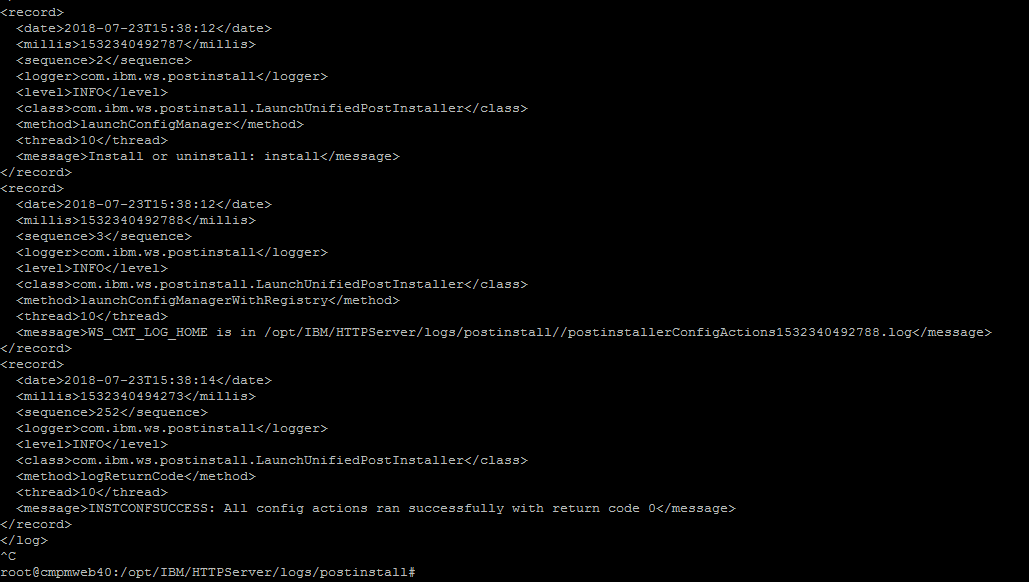
to check whether the installtion is done correctly or not go to below path and check the logs:

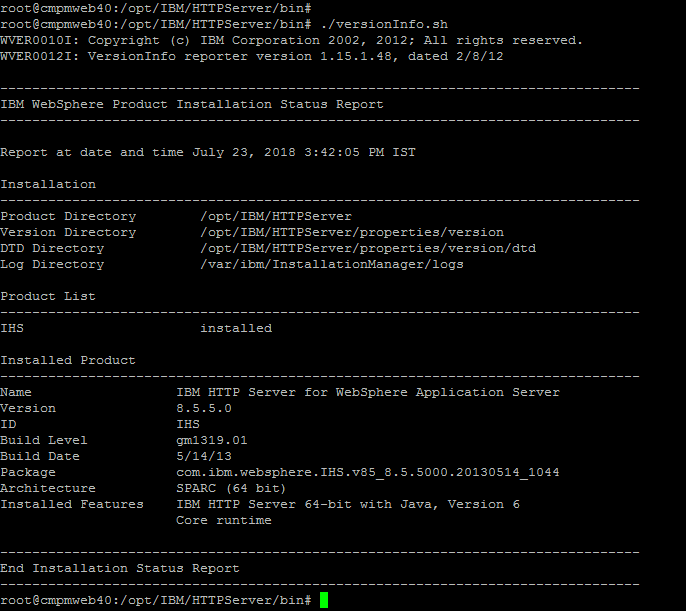
/opt/IBM/HTTPServer/logs/postinstall

cat postinstall.log

if you see the msg at last line mentioned below the installtion is successfully done.

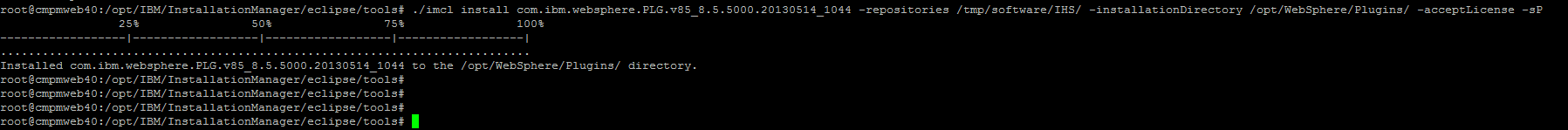
<message>INSTCONFSUCCESS: All config actions ran successfully with return code 0</message>



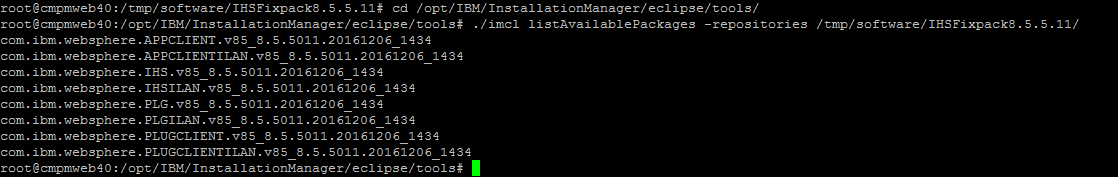
to verify the version of IHS installation

For plugin installtion: run the below command to install the plugins :

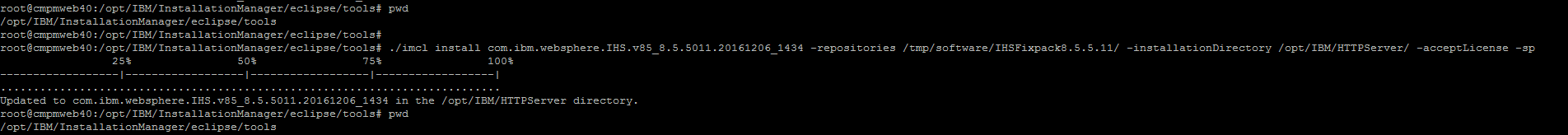
./imcl install com.ibm.websphere.PLG.v85\_8.5.5000.20130514\_1044 -repositories /tmp/software/IHS/ -installationDirectory /opt/WebSphere/Plugins/ -acceptLicense -sP

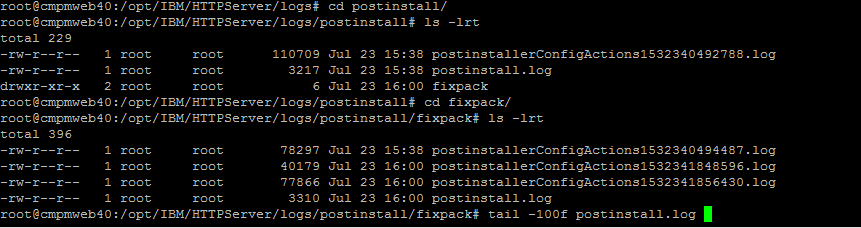


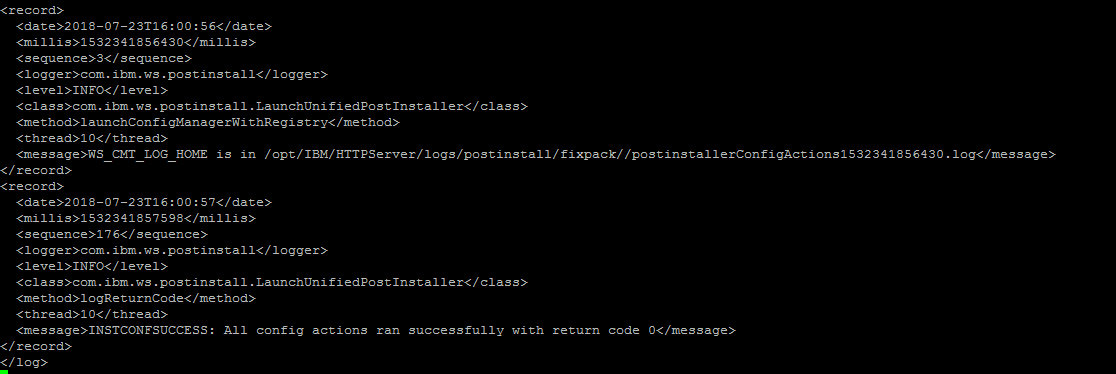
To installed the fixpack for IHS as well as Pluginsby using the below command:

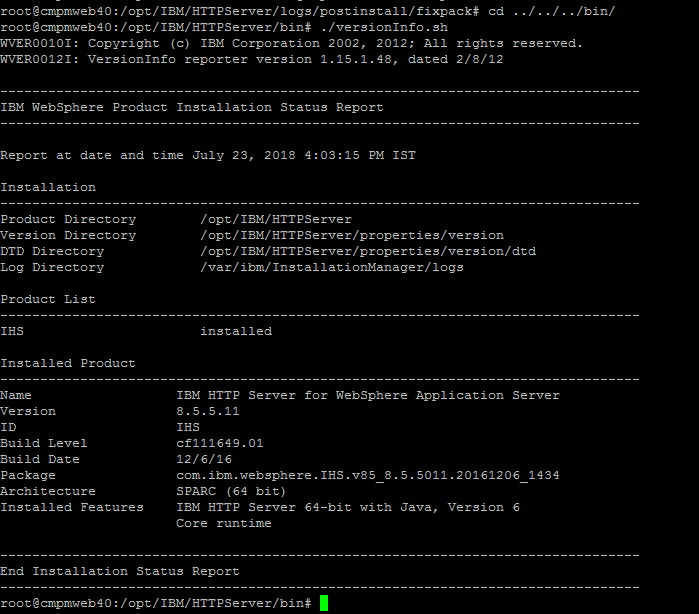


./imcl install com.ibm.websphere.IHS.v85\_8.5.5011.20161206\_1434 -repositories /tmp/software/IHSFixpack8.5.5.11/ -installationDirectory /opt/IBM/HTTPServer/ -acceptLicense -sp



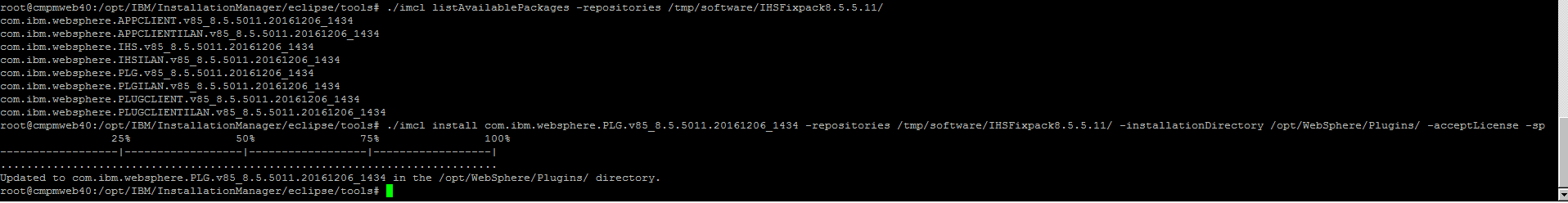






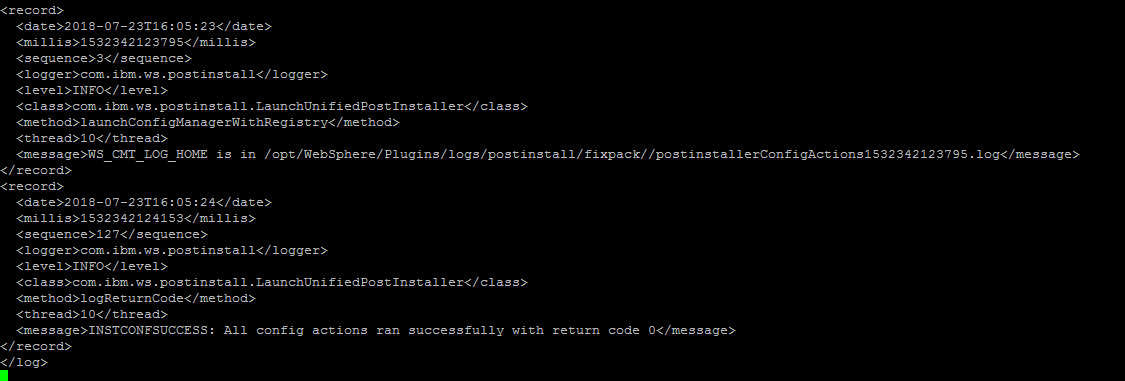
plugin fixpackinstalltion:

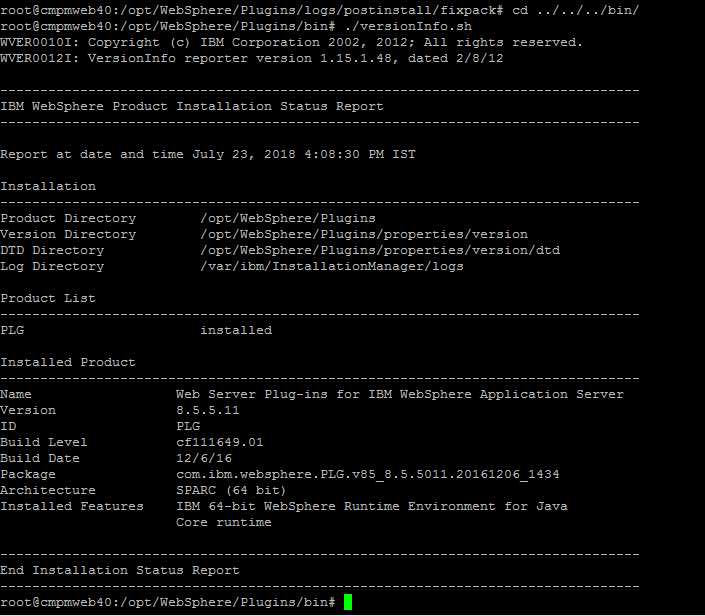
./imcl install com.ibm.websphere.PLG.v85\_8.5.5011.20161206\_1434 -repositories /tmp/software/IHSFixpack8.5.5.11/ -installationDirectory /opt/WebSphere/Plugins/ -acceptLicense -sp



verify the logs under the path and check the version info of plugins:







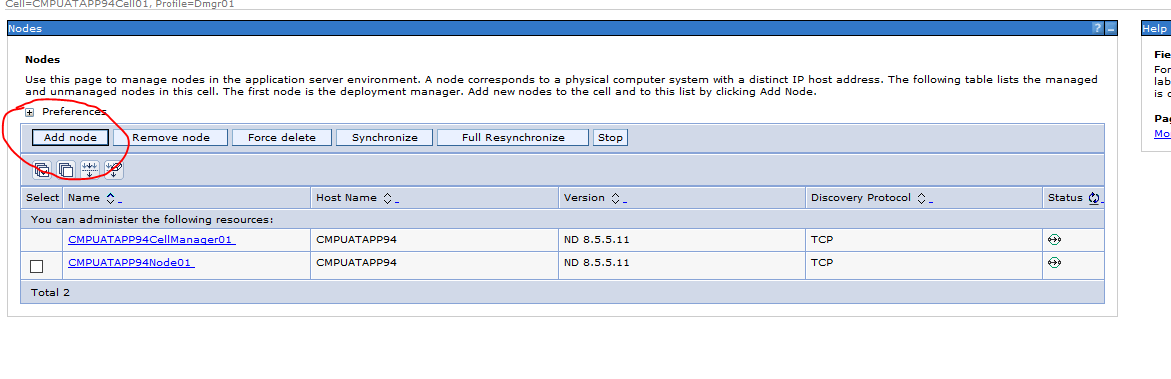
take the back up of original conf and replace the httpd.conf from prod intranet webserver or dr intranet webser 10.248.92.10-> change the ip in conf-> create the folder on the name of hostname under the path /opt/WebSphere/Plugins/config/cmpmweb40

go to /etc/hosts and add the below app server entries in webservers and appservers.

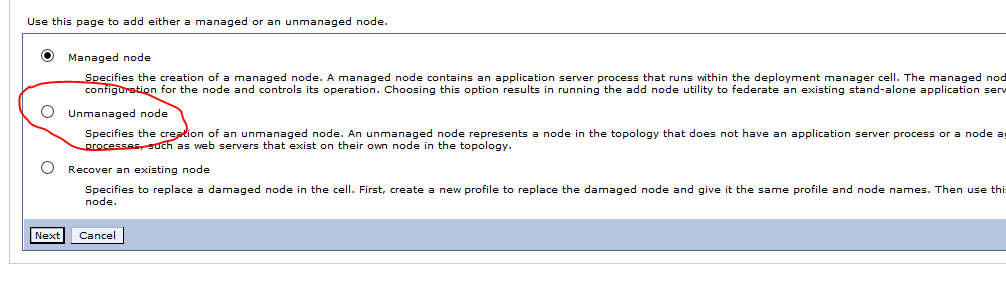
**After Installing IHS, integration of IHS with WAS**

31) Console level node integration

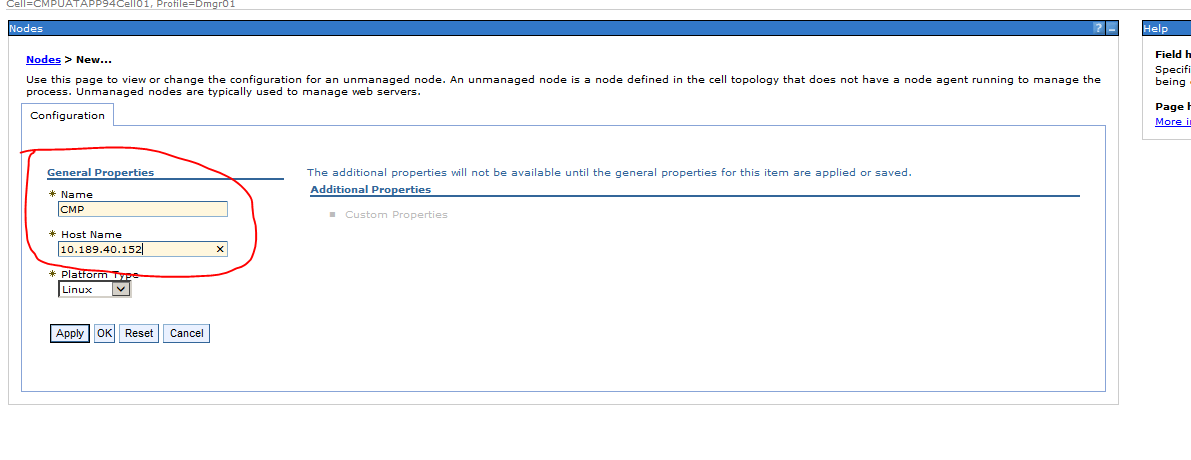
goto System Administration -> select nodes -> click on Add node



32)select unmanaged node and click on next

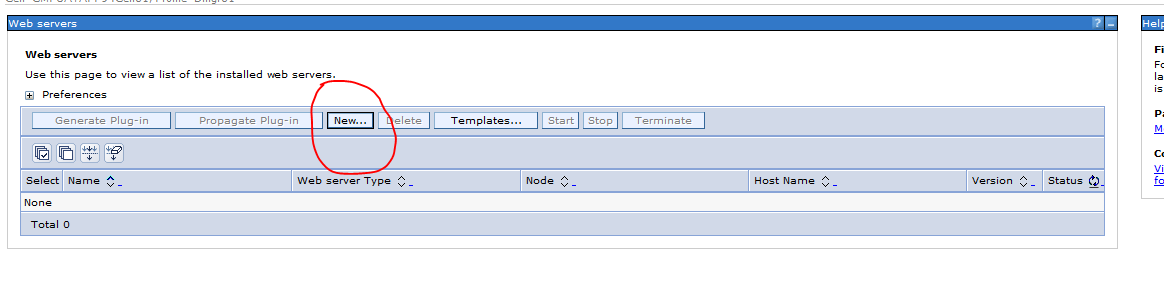


33)Give the webserver node name and host name

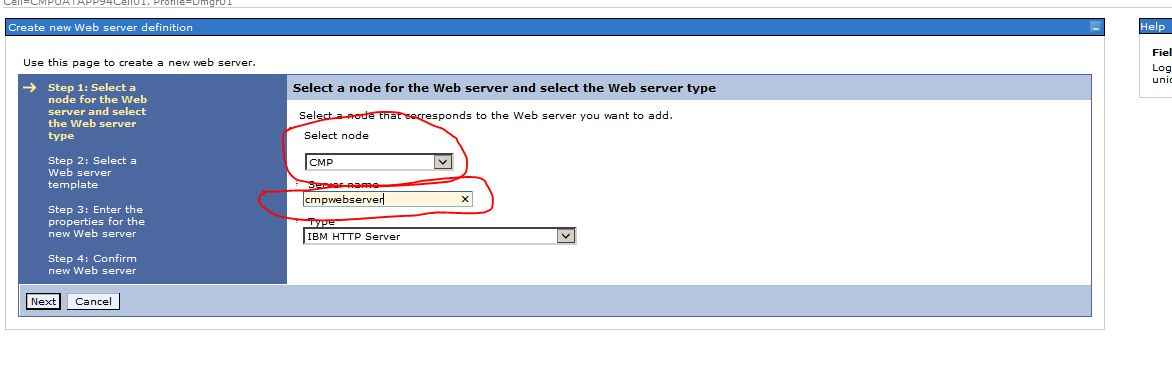


34) save the changes and synchronize the node.

35)goto servers ->webservers->click on new

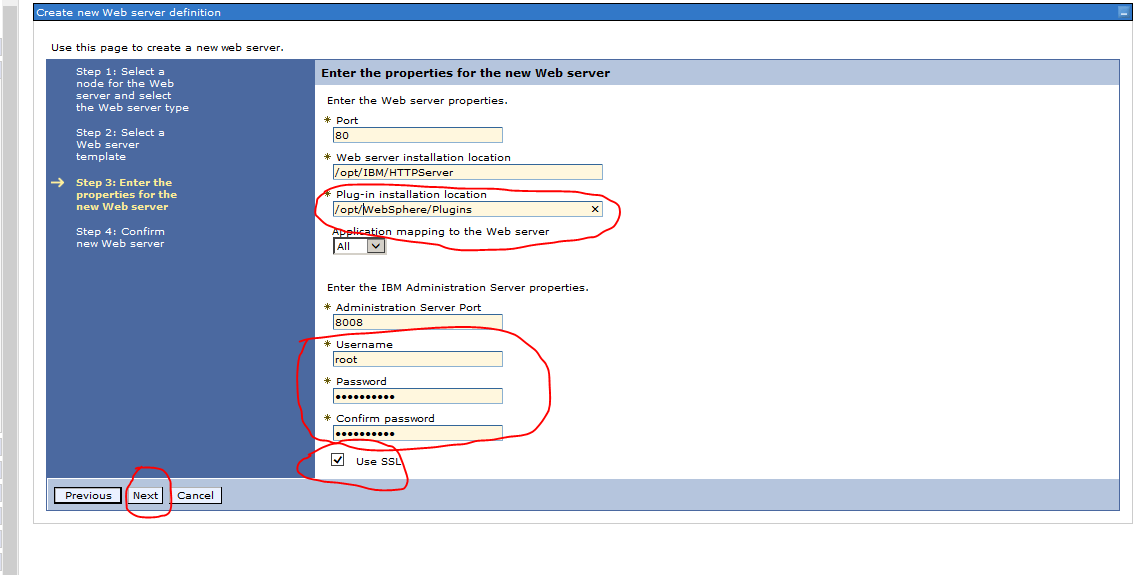


36)select the node and -> give the server name as per requirement and click on next - > next

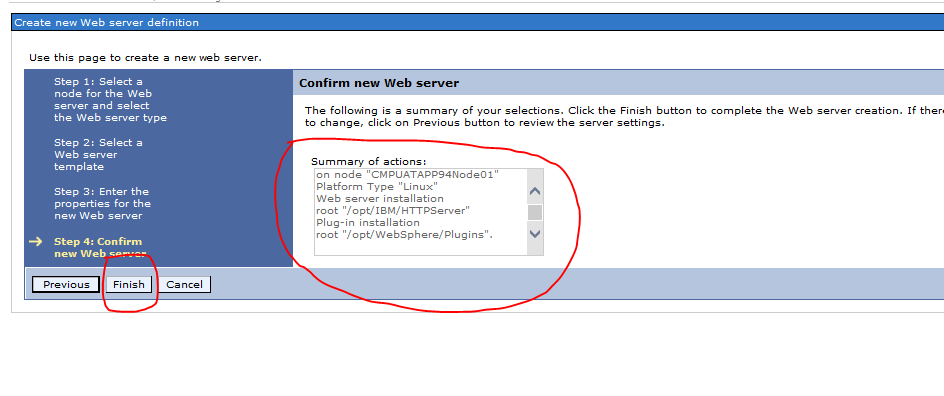


37)give the webserver user name and password select the check box of use ssl and click on next.

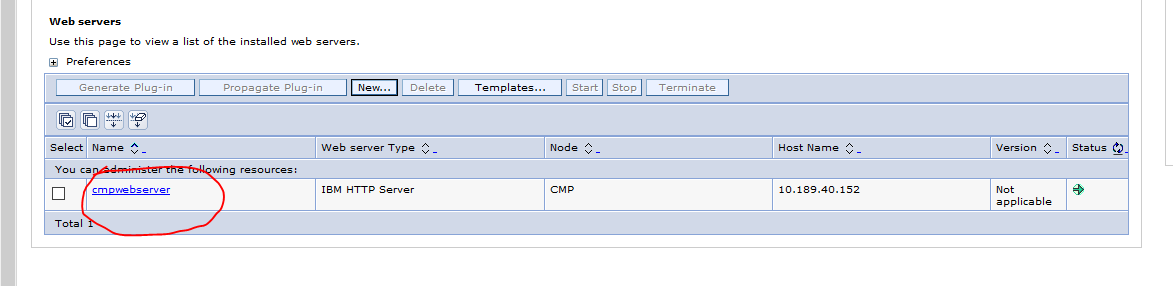
give the plugin installation path and click on next



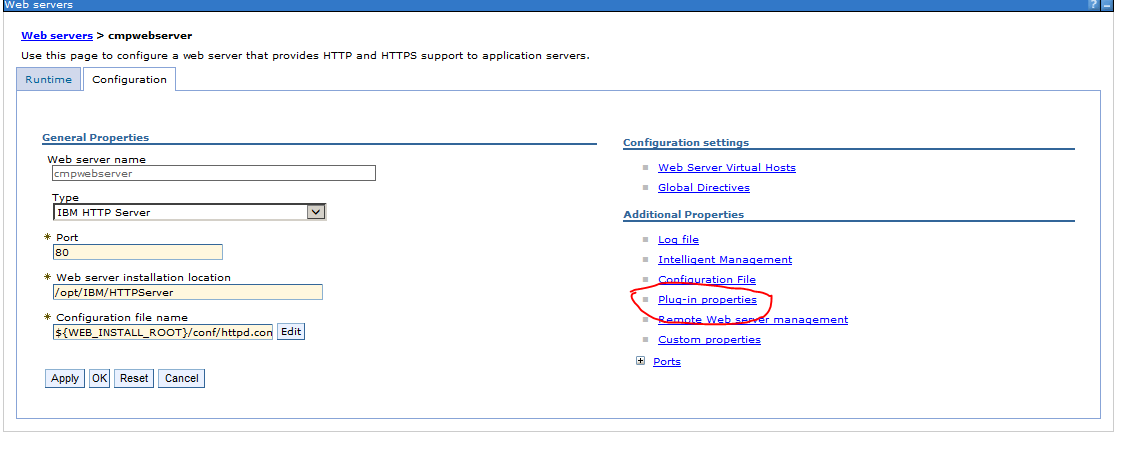
38) now check the summary and click on finish



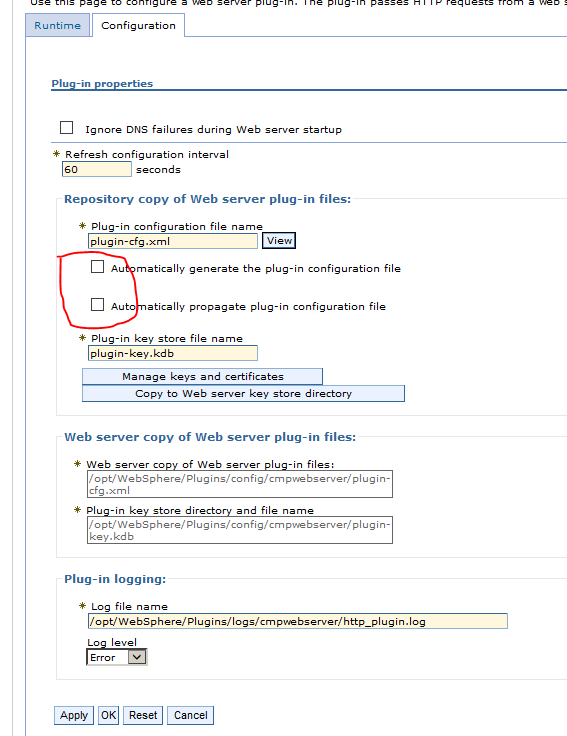
39) click on the webserver



40) select plugin properties



41) uncheck the both checkbox and click on ok and save.



42) Now select the webserver and generate the plugin and placed in webserver

console-> Webserver-> Click on GenPlugin tab->

Files get generated at …Dmgr/config/cells/cellname/nodes/webservernode/servers/jvm

Copy these files to Websphere/Plugins/config/

Place the location of plugin-cfg.xml in httpd.conf at the bottom

WebSpherePluginConfig /u01/WebSphere/Plugins/config/IFAMS-APP01.SBI/plugin-cfg.xml

To start IHS

Navigate to HTTPServer/bin;

./apachectl –k start

**For Address already in use: make\_sock: could not bind to address issues**

Given Non ssl port and virtual host port different

**For Library Issues**

<https://www.ibm.com/support/knowledgecenter/en/SSZH4A_6.2.0/com.ibm.worklight.installconfig.doc/admin/t_troubleshooting_ihs_startup.html>

**SSl using GSKCAPICMD for IBM**

More details:

<https://www.ibm.com/support/knowledgecenter/en/SSEQTJ_9.0.5/com.ibm.websphere.ihs.doc/ihs/rihs_ikeycmdsyn.html>

**How to raise CSR**

=========================

1. Create a blank .kdb file (at your desired location with desired name) using below command from the location /opt/IBM/HTTPServer/bin of any web server as root user

./gskcapicmd -keydb -create -db "/tmp/CERTS\_2019/uatfreedom.onlinesbi\_CSR/uatfreedom\_onlinesbi\_com.kdb" -pw "admin" -type cms -stash

1. Now create a certificate request using below command by providing all necessary details carefully.

./gskcapicmd -certreq -create -db "/tmp/CERTS\_2019/uatfreedom.onlinesbi\_CSR/uatfreedom\_onlinesbi\_com.kdb" -pw "admin" -dn "CN=uatfreedom.onlinesbi.com,O=STATE BANK OF INDIA,OU=MOBILE SERVICES & MOBILE BANKING,L=NaviMumbai,ST=Maharashtra,C=IN" -label "uatfreedom.onlinesbi.com" -size "2048" -sigalg "sha512" -file "/tmp/CERTS\_2019/uatfreedom.onlinesbi\_CSR/uatfreedom\_onlinesbi\_com.csr"

**TO List the SSL Certificates in the .KDB File**

To read & list the certificates in kdb

./gskcapicmd -cert -list -db "/tmp/CERTS\_2019/yono\_280619/yonobusiness.kdb" -type "kdb" -pw "admin"

**TO RECEIVE and ADD the SSL Certificates in the .KDB File:**

To add the root & intermediate certificates provided by CA into KDB & receive the domain certificate into KDB file so that we can place KDB, STSH, RDB& P12 files in webserver location and map the path in httpd.conf file

**Intermedaite certificate:**

./gskcapicmd -cert -add -db "/tmp/14122018/mysbiworld.kdb" -type "kdb" -pw "admin" -file "/tmp/14122018/certs/DigiCert-SHA2-Extended-Validation-Server-CA.cer" -label "DigiCert-SHA2-Extended-Validation-Server-CA.cer" -trust "enable"

**Root certificate:**

./gskcapicmd -cert -add -db "/tmp/14122018/mysbiworld.kdb" -type "kdb" -pw "admin" -file "/tmp/14122018/certs/DigiCert-High-Assurance-EV-Root-CA.cer" -label "DigiCert-High-Assurance-EV-Root-CA.cer" -trust "enable"

**Domain certificate:**

./gskcapicmd -cert -receive -db "/tmp/14122018/mysbiworld.kdb" -type "kdb" -pw "admin" -file "/tmp/14122018/certs/mysbiworld\_onlinesbi\_com.cer"

**TO Convert the Data from .KDB to .PKCS12:**

========================================

./gskcapicmd -cert -export -db "/tmp/CERTS\_2019/yono\_280619/yonobusiness\_bkp\_original\_28062019.kdb" -pw "admin" -label "yonobusiness.sbi" -type "cms" -target "/tmp/CERTS\_2019/yono\_280619/yonobusiness.p12" -target\_pw admin -target\_type "pkcs12"

**TO verify and Display the SSL Certificates from .P12 file**

==================================================================================

./gskcapicmd -cert -details -showOID -db "/tmp/CERTS\_2019/yono\_280619/yonobusiness.p12" -type "pkcs12" -pw "admin" -label "yonobusiness.sbi"

We can also verify the same using openssl command also.

openssl pkcs12 -info -in /tmp/14122018/certs/mysbiworld.p12

**setdefault for domain certs**

===================================

./gskcapicmd -cert -setdefault -label trgmobilitymfconsole.onlinesbi.com -db "/home/wasadmin/Madhu\_konycerts/trgmobilitymfconsole.onlinesbi.com/trgmobilitymfconsole\_onlinesbi\_com.kdb"

* Kindly place below things in httpd.conf

KeyFile "/home/webadmin/mab/mab.kdb"

SSLServerCert MABSBIMU {like Alias}

SSLStashFile "/home/webadmin/mab/mab.sth"

Self Signed Certificate generation:

=======================================

1.Run the following OpenSSL command to generate your private key and public certificate. Answer the questions and enter the Common Name when prompted.

openssl req -newkey rsa:2048 -nodes -keyout key.pem -x509 -days 365 -out certificate.pem

2.Review the created certificate:

openssl x509 -text -noout -in certificate.pem

3.Combine your key and certificate in a PKCS#12 (P12) bundle: Remember the Export PSWD

openssl pkcs12 -inkey key.pem -in certificate.pem -export -out certificate.p12

4. if you have intermediate certificates from your CA, or if u have multiple certificates then concatenate them into a single pem file to build your caChain.

Be sure to enter a new line following each certificate's data.

cat ca1.pem ca2.pem ca3.pem > caChain.pem

openssl pkcs12 -inkey key.pem -in caChain.pem -export -out certificate.p12 #to export multiple certificates to p12

openssl pkcs12 -inkey key.pem -in certificate.pem -export -out certificate.p12 -CAfile caChain.pem -chain #to export CA chain along with domain certificate

4.Validate your P12 file.

openssl pkcs12 -in certificate.p12 -noout -info

1. Check .p12 / .pfx certificate expiration date:=====================

openssl pkcs12 -in testuser1.pfx -nokeys | openssl x509 -noout -enddate

openssl pkcs12 -in testuser1.pfx -nokeys | openssl x509 -text # for complete information

To specify password in plain text, add -passin pass:”${pass}”

2. Export key and cert from .p12 / .pfx:==========

openssl pkcs12 -clcerts -nokeys -in myContainer.p12 -out usercert.pem

openssl pkcs12 -nocerts -in myContainer.p12 -out userkey.pem

3. Connect to HTTPS server with client certificate:=======================

openssl s\_client -connect gmail.com:443 -cert usercert.pem -key userkey.pem

1. Generate a RSA praivate Key. Like KDB we need to generate a key first using below command. Remeber the Pswd

openssl genrsa -des3 -out /home/wasadmin/sslSelf/test.key 2048

2. CSR

===============

openssl req -key /home/wasadmin/sslSelf/test.key -new -out /home/wasadmin/sslSelf/testCsr.csr

Remaining details are need to be provided when requested

3. CA Signing

=========================

After signing we will get domain.cer, root.cer, intermediate.cer

3. Receive certificate into key & export to p12

===========================================================

openssl pkcs12 -export -in domain.cer -inkey test.key -out certificate.pfx # receive only domain certificate

## if you have intermediate certificates from your CA, or if u have multiple CA certificates then concatenate them into a single pem file to build your caChain.

##Be sure to enter a new line following each certificate's data.

===============================================================

cat root.cer intermediate.cer > caChain.cer

openssl pkcs12 -inkey test.key -in caChain.cer -export -out certificate.p12 #to export multiple certificates only to p12

openssl pkcs12 -inkey test.key -in certificate.cer -export -out certificate.p12 -CAfile caChain.cer -chain #to export CA chain along with domain certificate

4.Validate your P12 file.

openssl pkcs12 -in certificate.p12 -noout -info

5. P12 to kdb & stash file using gskcapicmd

=================================================

go to any IHS web server

/opt/IBM/HTTPServer/bin/gskcapicmd -cert -export -db mabprmup12.p12 -pw admin -type pkcs12 -target mabprmup12tokdb.kdb -target\_pw admin -target\_type kdb # p12 to kdb

# Now KDB, CRL & RDB files are generated.

/opt/IBM/HTTPServer/bin/gskcapicmd -keydb -stashpw -db "mabprmup12tokdb.kdb" -pw "admin" # genrate stash file from kdb

**JVM CREATION & CLUSTER**

**To start CLUSTER Members from Console use Ripplestart means To start one by one JVM**

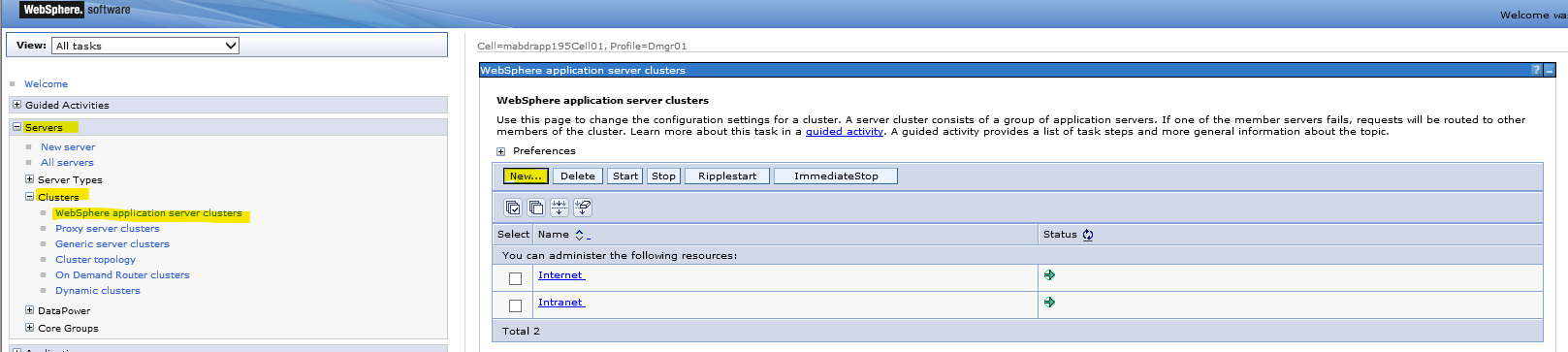
**Or use script** nohup ./startServer.sh [JVM MEMBER NAME]

APP Server Default PORT:9081

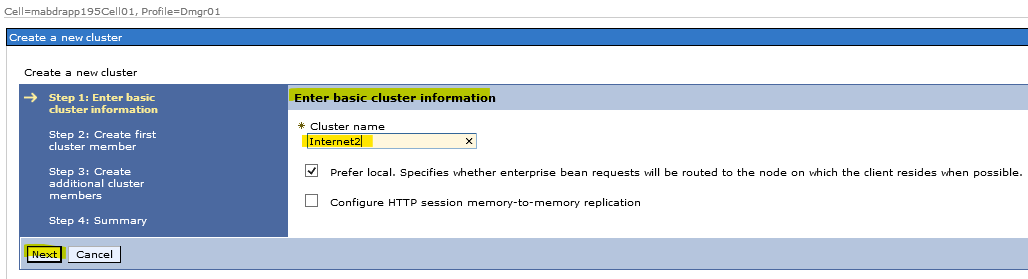
Prerequisites:

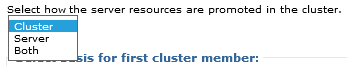
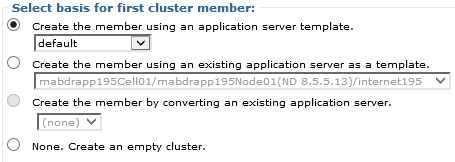
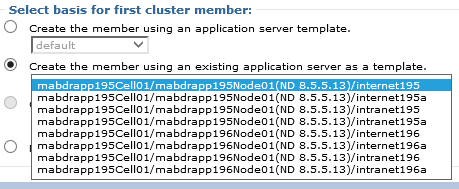
* Make sure all expected nodes are federated with DMGR
* Make sure DMGR and all NODE agents are running
* Identify how many cluster members and how each member configuration required.

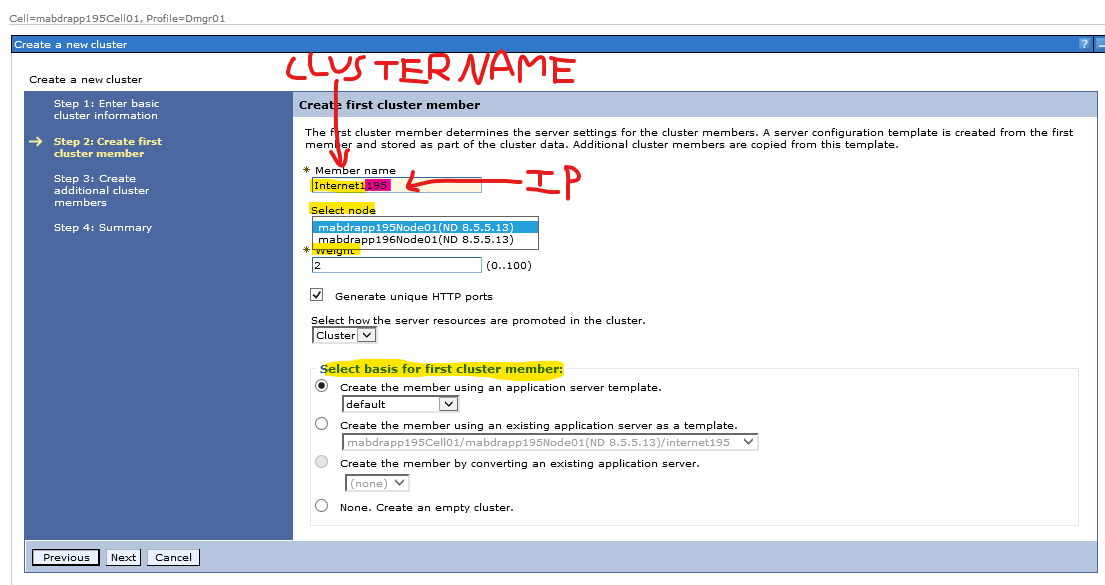
1. Create a cluster if the environment need to be made HIGH VAILABILITY otherwise you can directly configure the already created (during federating the NODE server1 will automatically created) “server1” JVM
   1. Open CONSOLE, click on SERVERS, then CLUSTERS then “WebSphere application server clusters” and then click on NEW



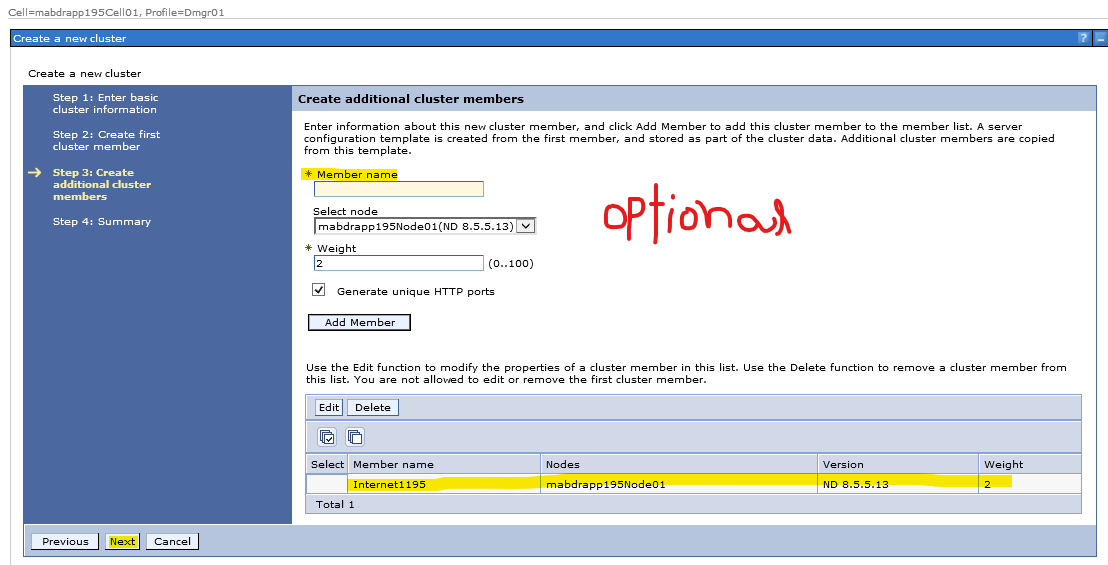
1. Give the cluster name as per the routine and click NEXT
   1. PREFER LOCAL, option is checked by default
   2. CONFIGURE HTTP, option need to be checked only, if you want to maintain each user request and status, in all cluster members. So that if a cluster member fails to complete the user request then other cluster member will serve that request without showing error to user.



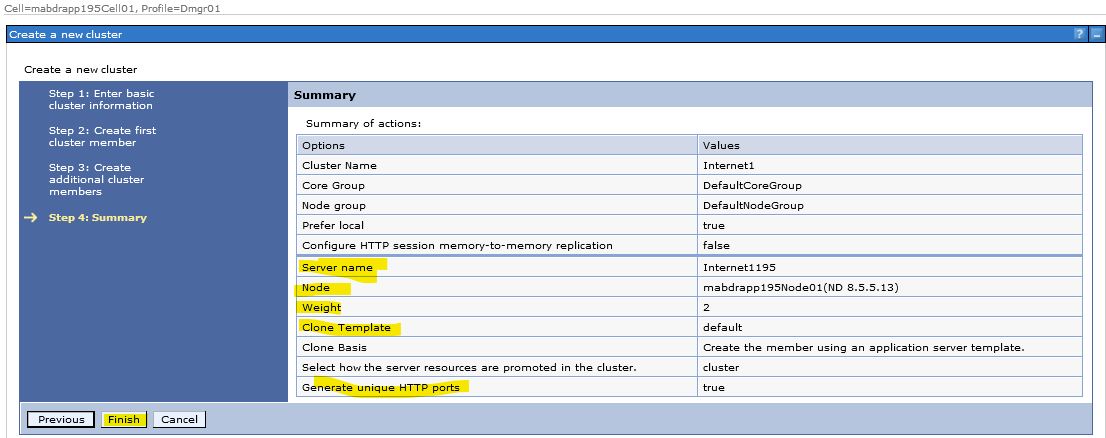
1. Enter the 1st cluster member name & configure the given parameters as given below
   1. Make sure the member name includes the details of CLUSTER & SERVER IP. Ex Internet1195
   2. Select the NODE from drop down list (It will show all nodes federated with DMGR) on which you want to implement the CLUSTER & JVM. You can create different JVMs associated with different NODES in the single cluster.
   3. Enter the WEIGHT details in between 0 to 100. By default it will be 2. Equal weight value to all cluster members represents that all JVMs are serving equal number of requests.
   4. “Generate unique HTTP ports” option is checked by default. Keep it.
   5.  Cluster by default
   6.  This option is selected when you want to define your own JVM parameters. Or  this option is selected when you want to assign this selected pre-configured JVM parameter to this member and all other members which are added (in this cluster) in the next step. Remaining two options are self-explanatory. Note: If you are planning to add multiple members to a cluster with same JVM configurations, then first add one member and configure JVM parameters as per requirement then add members to the cluster using the 2nd option shown above. So that you don’t have to configure same JVM parameters to each cluster member. Then click NEXT



1. As discussed above, you can add more members to this cluster if you want, along with 1st member. Else without giving any details simply click NEXT



1. Cross check the given details and click FINISH



1. Click REVIEW then SAVE after Synchronisation.
2. Now configure the JVMs

---------------------------------------------------------------------------------------------------------------------------

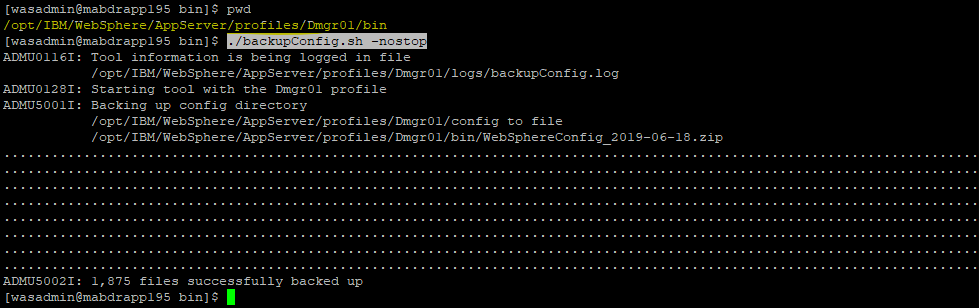
**JVM HEAP CONFIGURATION**

1. Go to DMGR bin and take Config backup as mentioned below

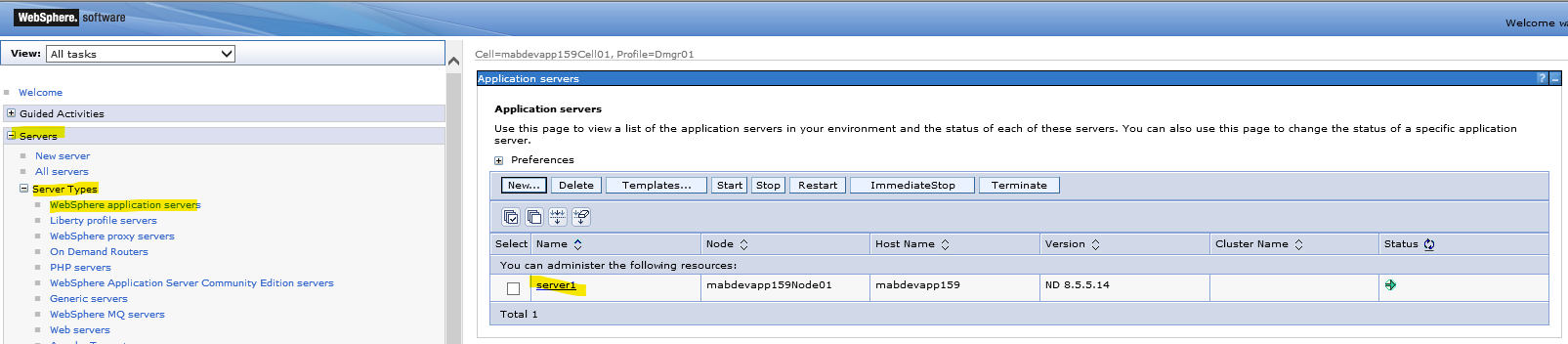
Path: /opt/IBM/WebSphere/AppServer/profiles/Dmgr01/bin

Command: ./backupConfig.sh –nostop

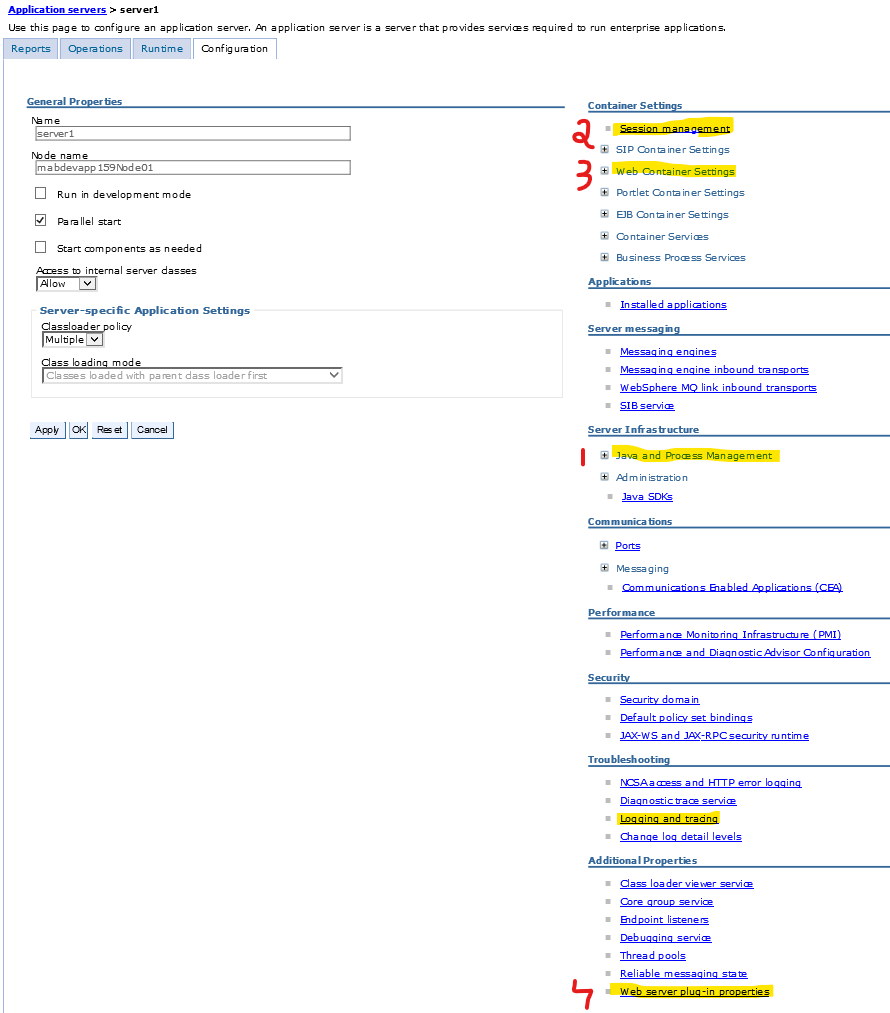
Note: use nostop argument so that running JVM’s will not be impacted, else all the services running will get stopped.



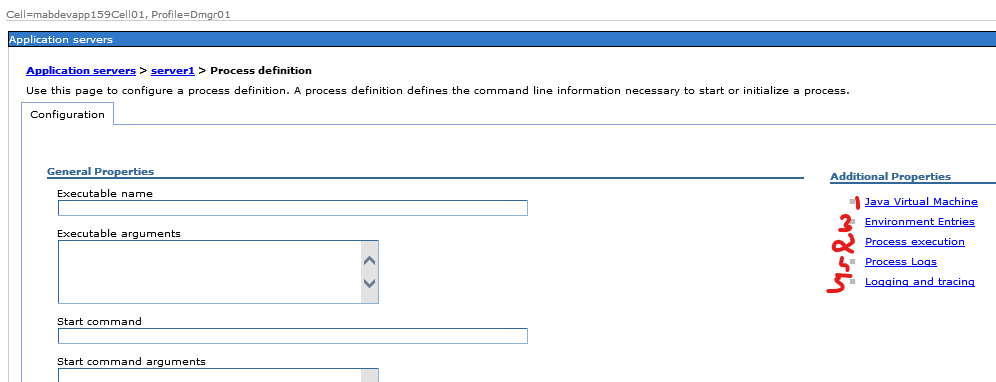
1. Open CONSOLE, click on SERVERS, SERVER TYPES, WEBSPHERE APPLICATION SERVERS and click on JVM/ Application Server Name, which you want to configure



1. Select and configure the options one by one.



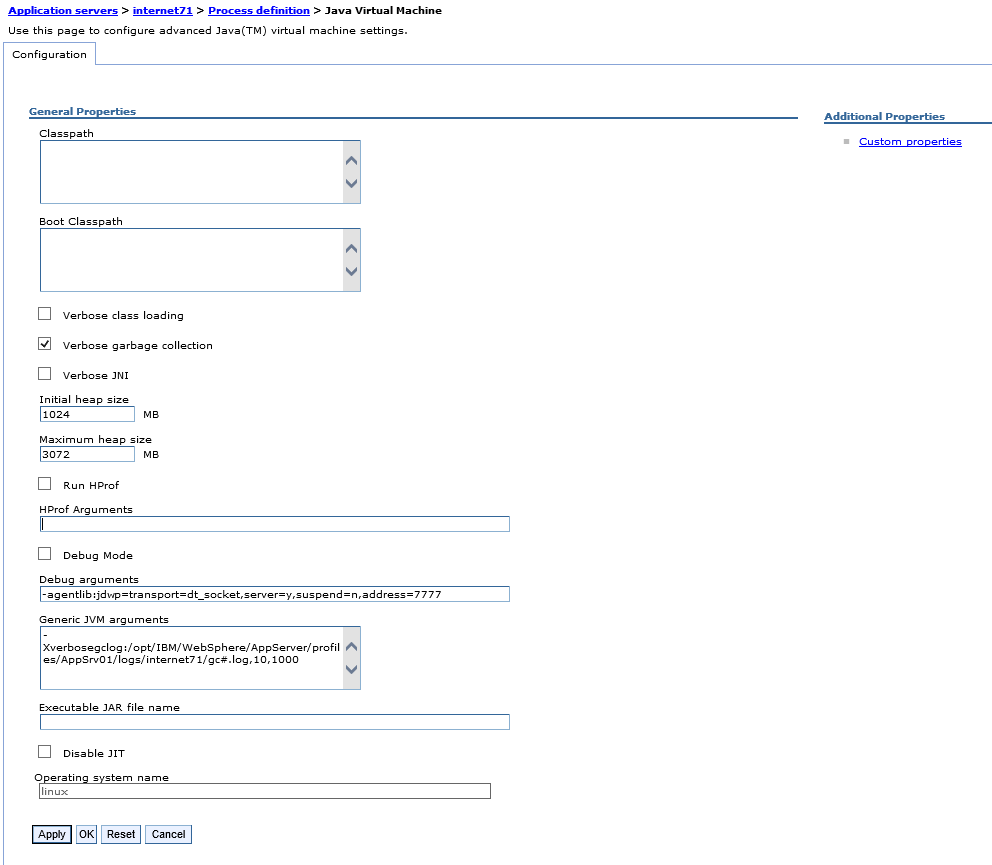
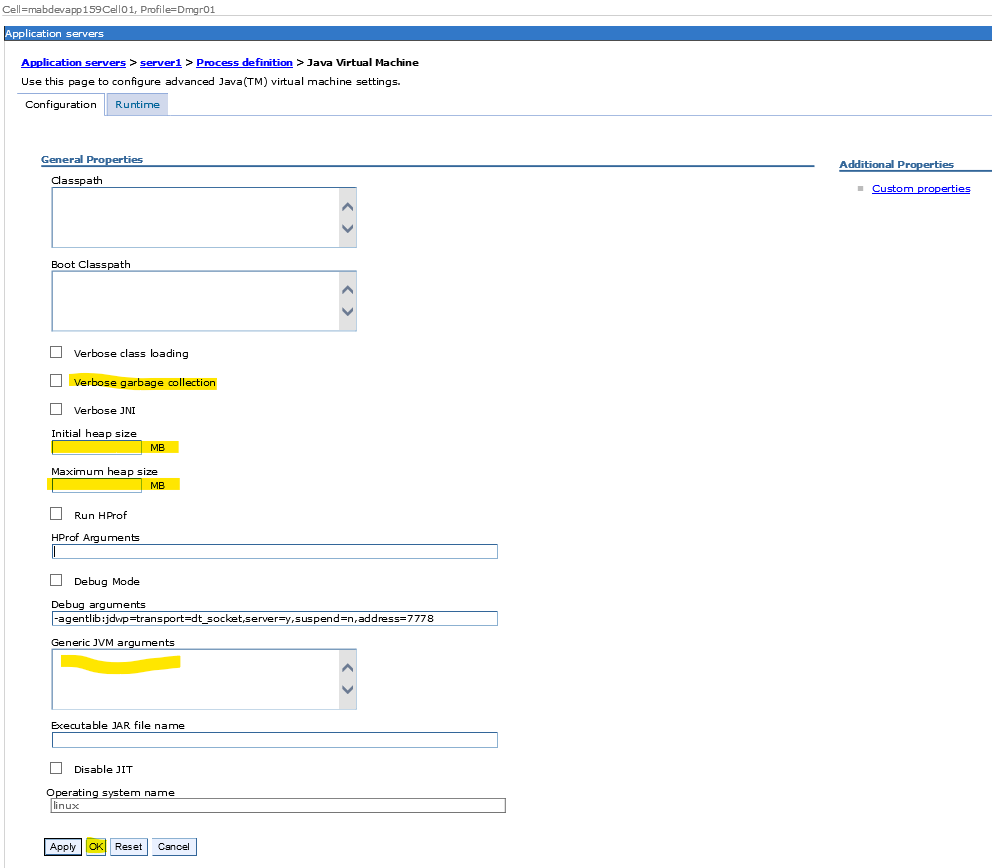
1. Click on JAVA & PROCESS MANAGEMENT, then PROCESS DEFINATION.



1. You have to configure 5 properties carefully. First Select Java Virtual Machine

First image : Empty default JVM

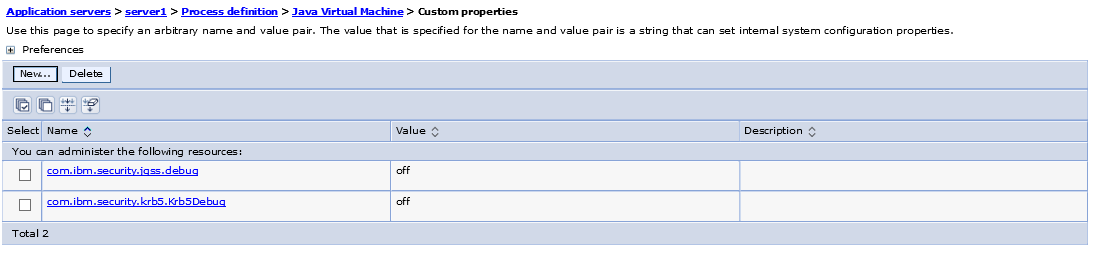
2nd image: configured JVM in production

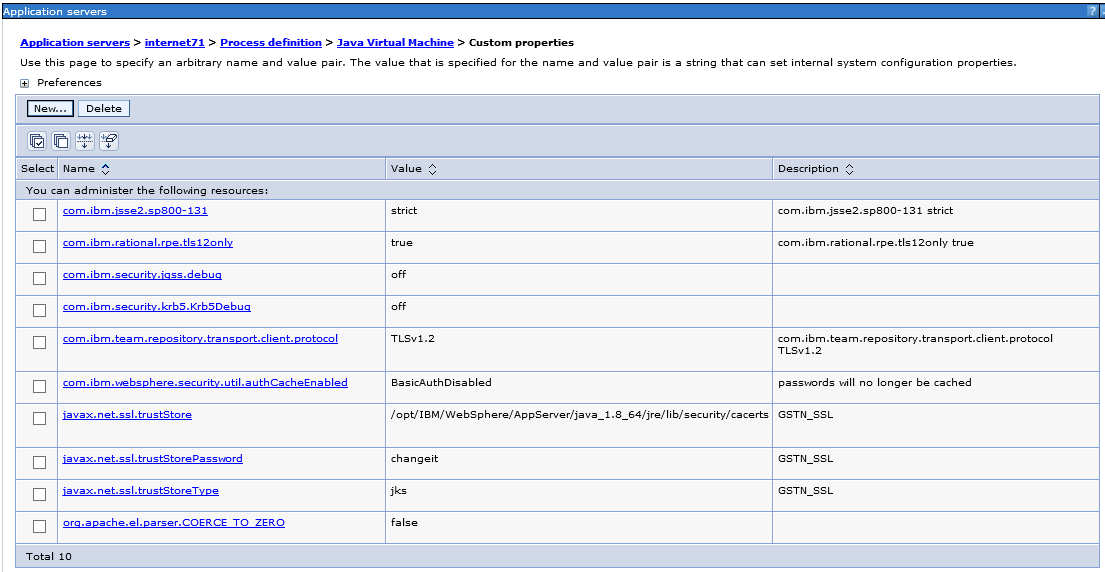


-Xverbosegclog:/opt/IBM/WebSphere/AppServer/profiles/AppSrv01/logs/internet195/gc#.log,10,1000

Check “Verbose garbage collection”, give initial 512 or 1024 & Max heap size as 3072. You may provide JVM argument based on your environment. Then click OK you will be taken back to PROCESS DEFINATION (PD) page. Then again select JVM, then CUSTOM PROPERTIES

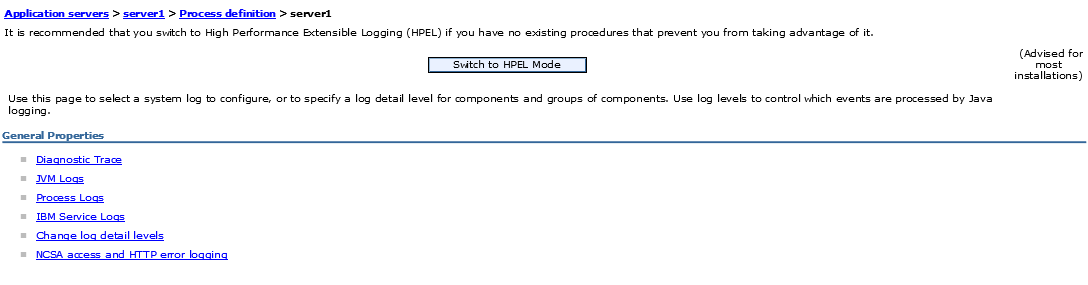
1. Then add properties. Then click on PROCESS DEFINATION on top blue link of same page.



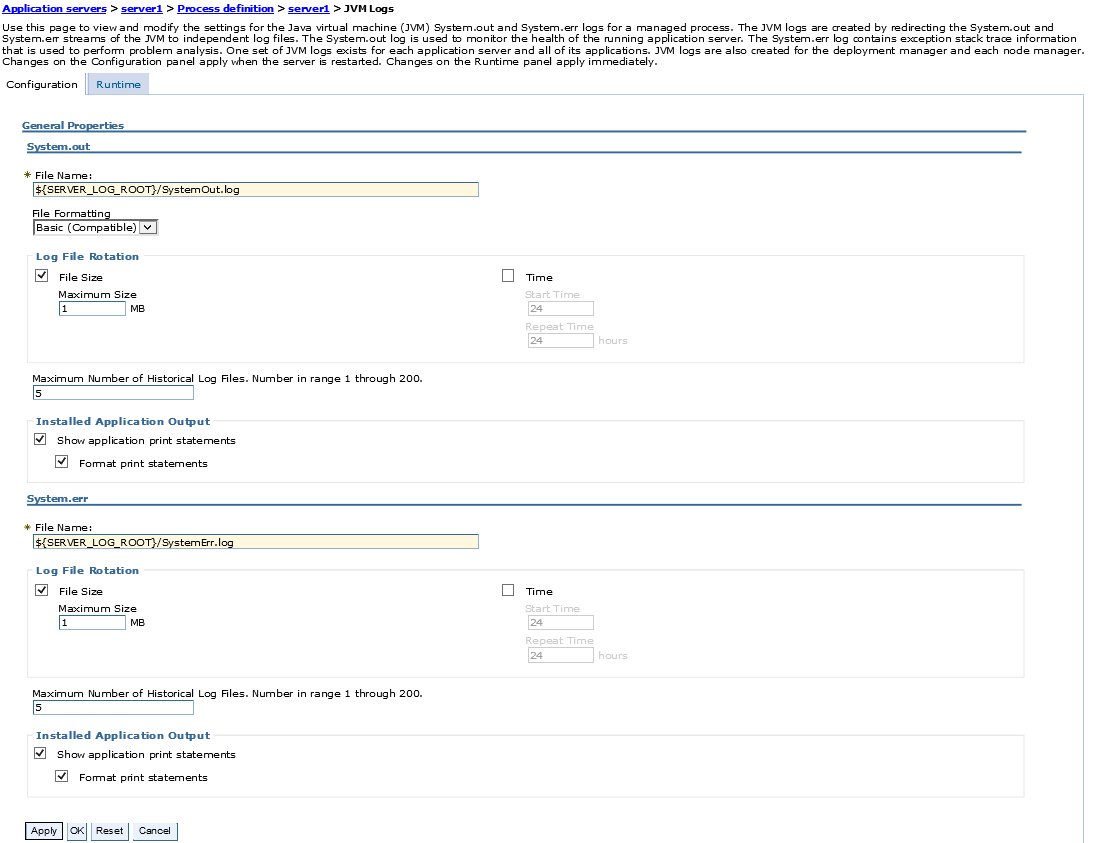


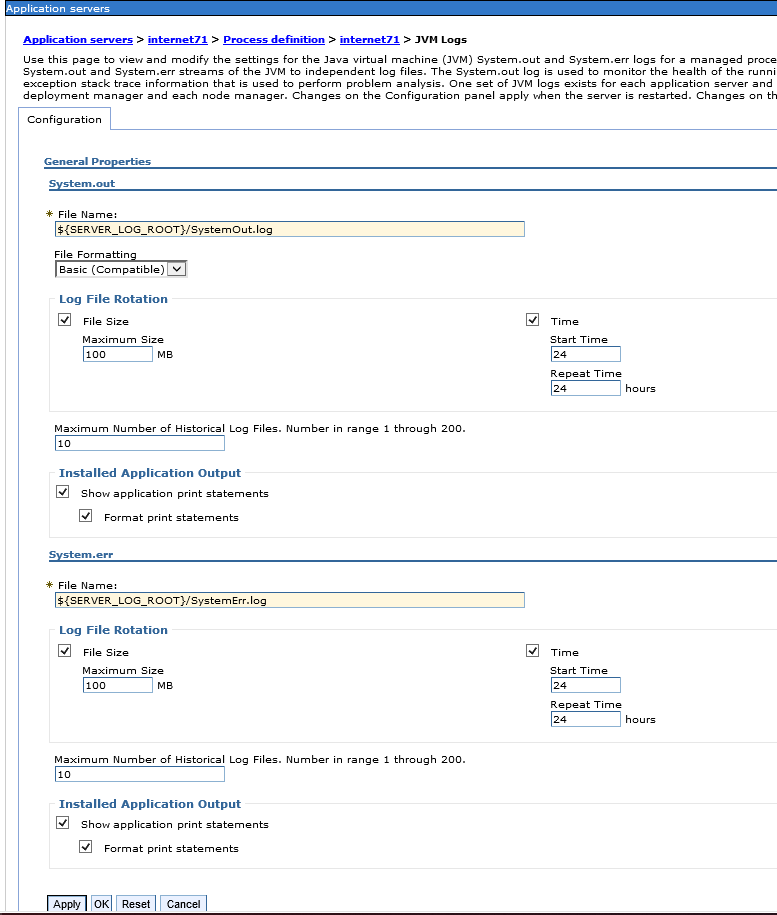
These extra properties are added based on VA & SCD

1. Environment Entries, Process execution & Process Logsdetails are not modified in this project. Then click on Logging & Tracing



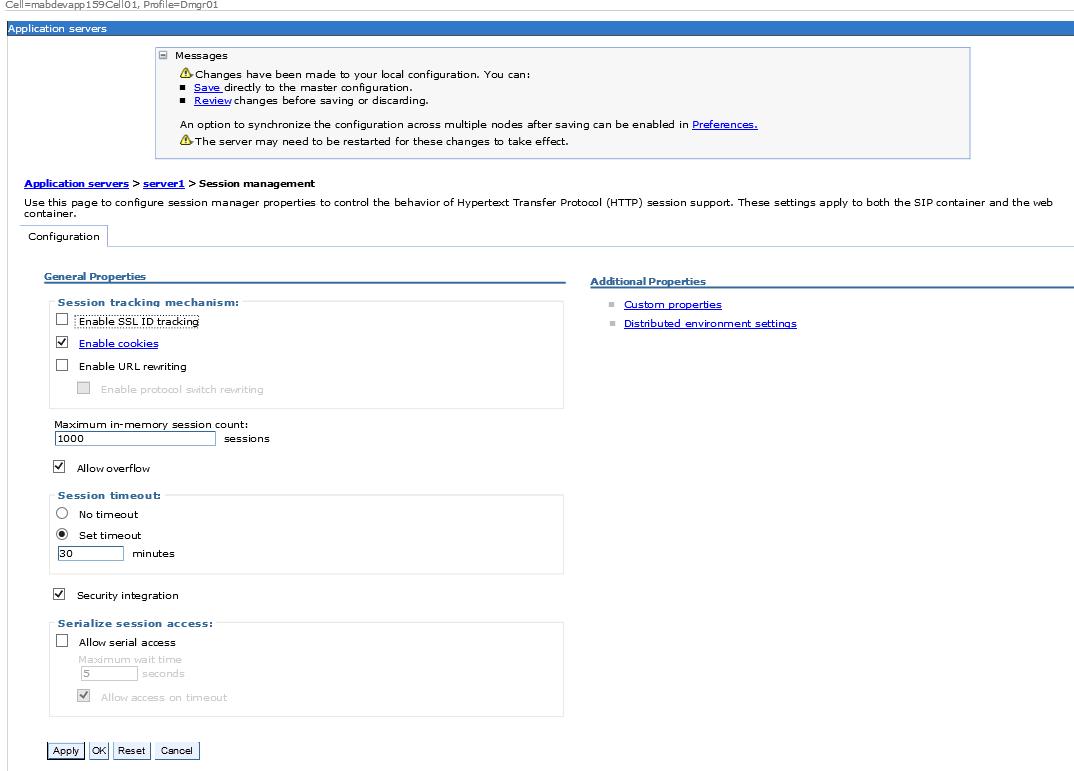
In above, expect PROCESS LOGS, remaining all are left default without changes.

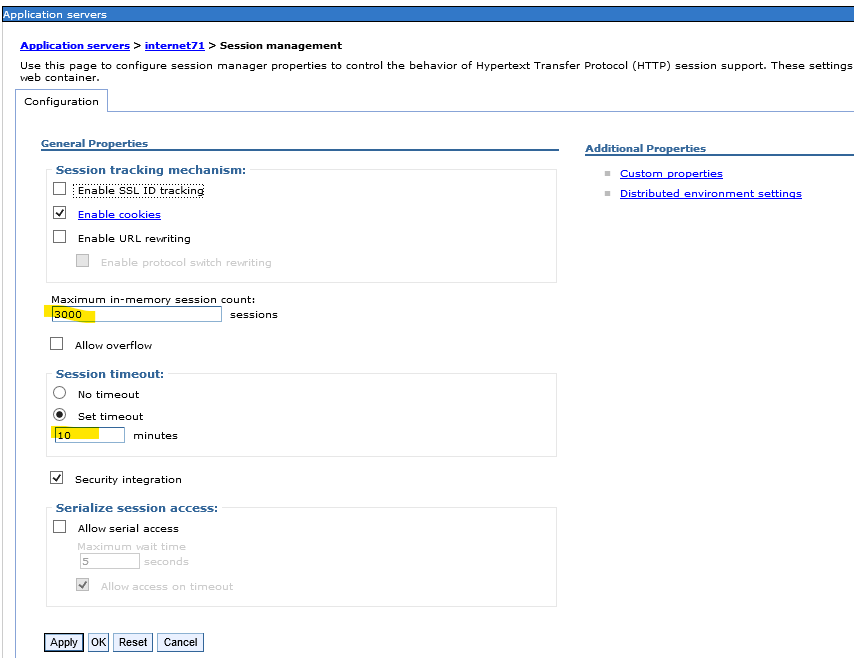




With these modifications go back to Server1 page. Java and Process Management is complete now

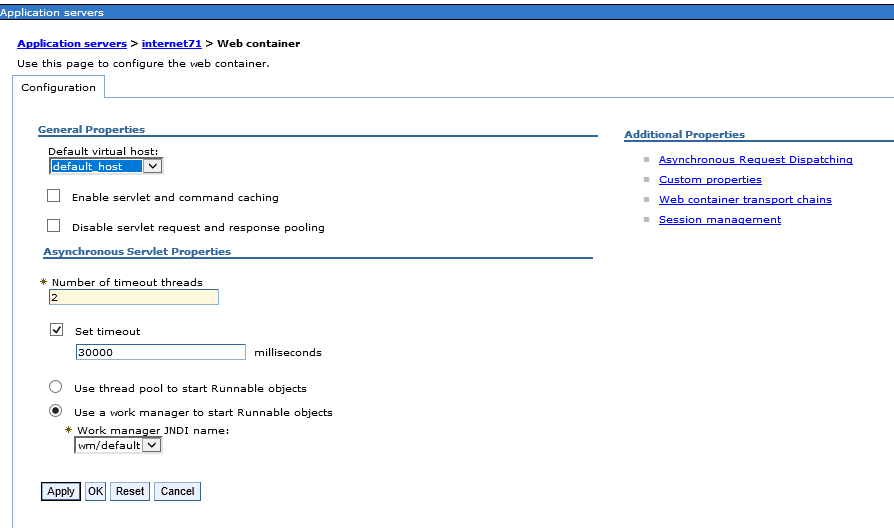
1. Click on SESSION MANAGEMENT





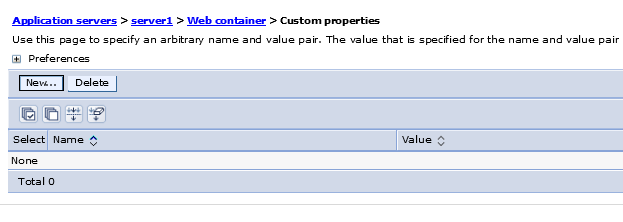
Modify the both. Remaining IN SESSION MANAGEMENT are not required.

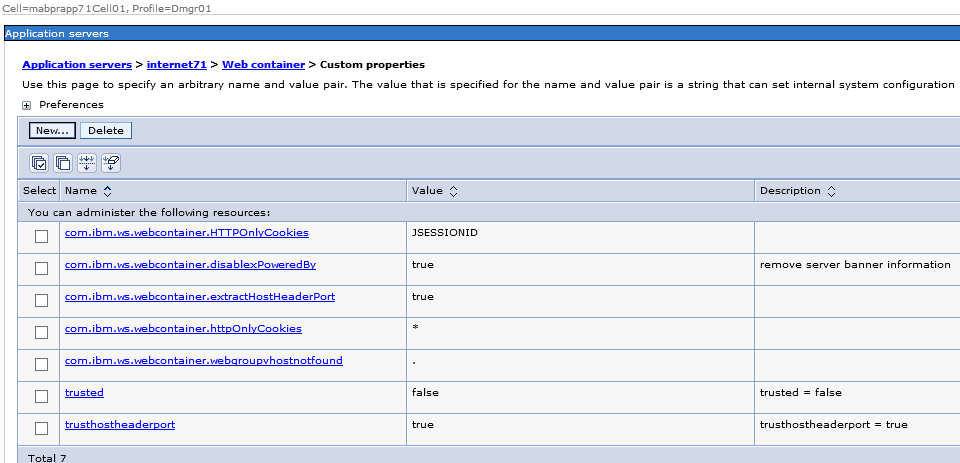
1. Then go back to Server1 page and click on click on WEB CONTAINERS



This page is default, Asynchronous Request Dispatching no changes, [Session management](https://10.209.72.159:9043/ibm/console/com.ibm.ws.console.servermanagement.forwardCmd.do?csrfid=923842969&forwardName=SessionManager.config.view&sfname=services&lastPage=webcontainer.config.view&resourceUri=server.xml&parentRefId=WebContainer_1183122130078&contextId=cells%3Amabdevapp159Cell01%3Anodes%3Amabdevapp159Node01%3Aservers%3Aserver1&perspective=tab.configuration) (Already modified). Changes required in Custom properties &Web container transport chains.

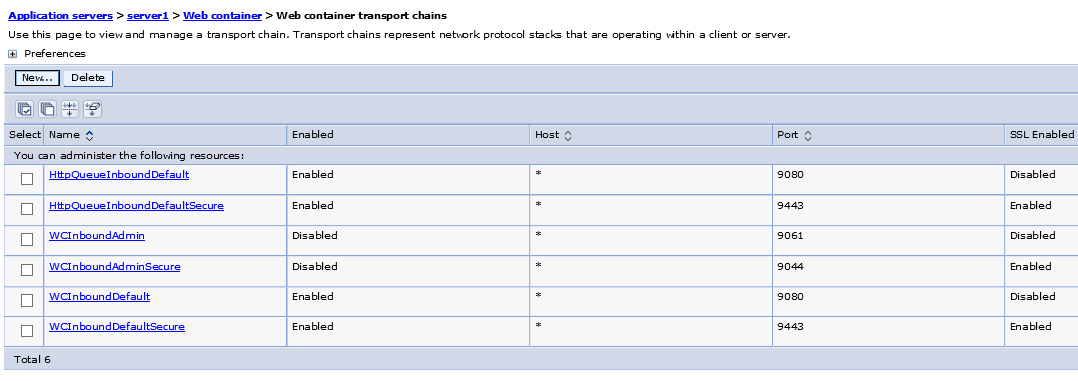
1. Select Custom properties

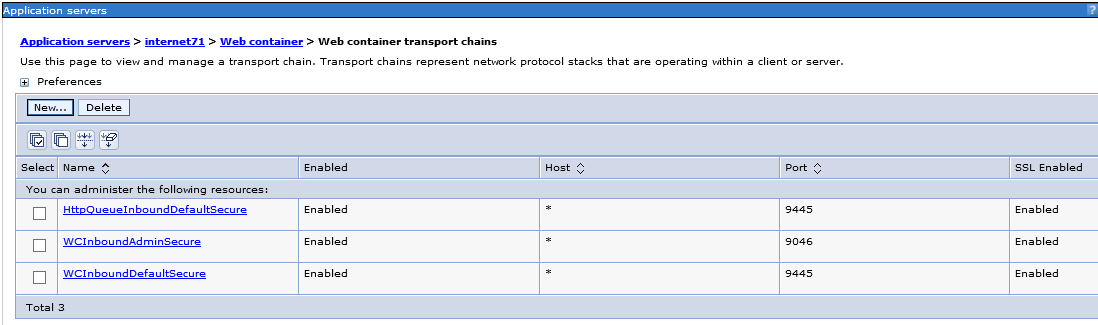




These are based on VA.

1. Go back and select Web container transport chains

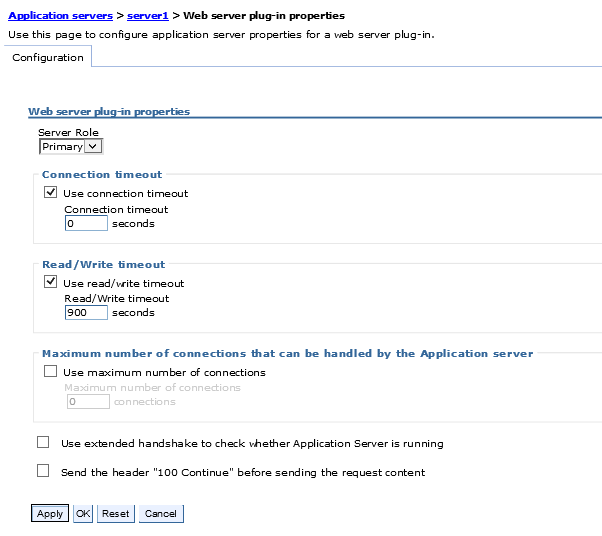


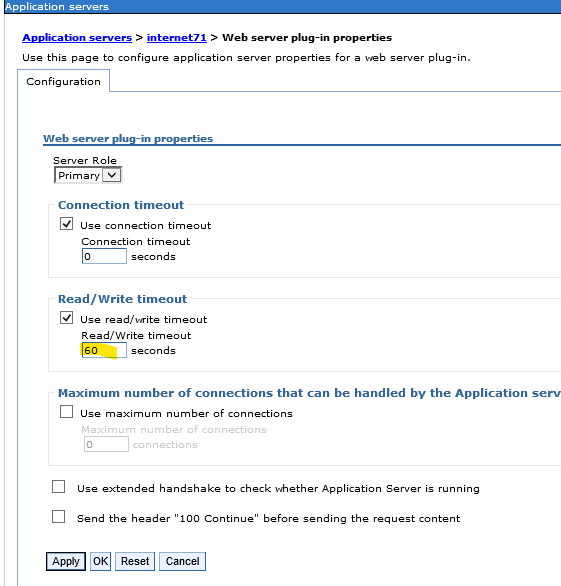


Here you need to disable all non-secure ports. This too based on your environments but mostly we disable non-secure.

Then Web containers is done.

1. Then select WEB SERVER PLUGIN PROPERTIES





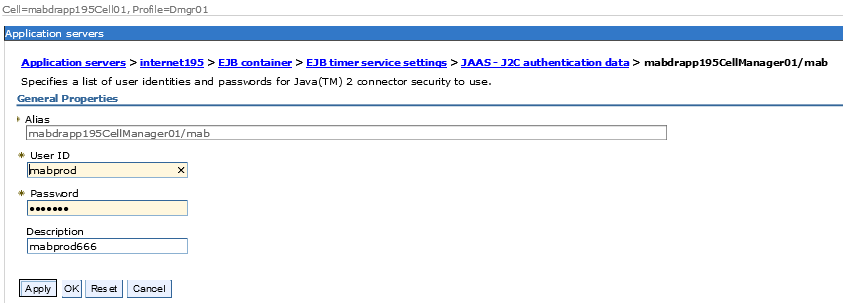
Modify Read/Write time out.

1. Logging & Tracing is already done in Java & Process Management stage.

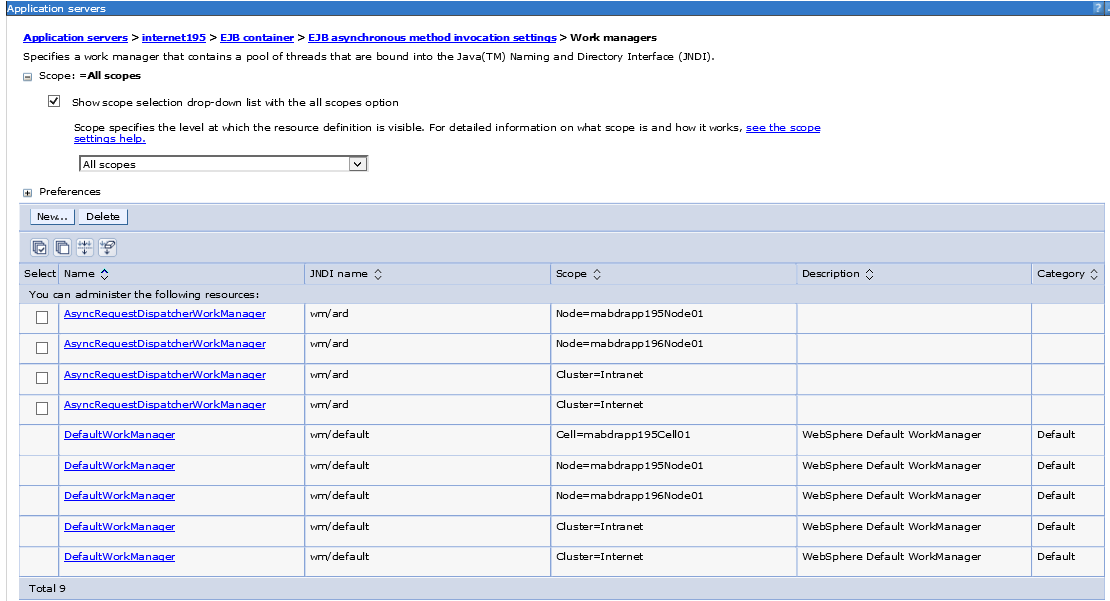
JVM Configuration is done.

In cluster, each JVM has to be configured individually. Even if you add extra member to cluster after configuring one JVM, then also newly added JVM is empty, so you need to configure it. While creating a new cluster, then only you can use pre-configured JVM parameters of another cluster, to assign them to the members of this new cluster.

* Other than above, location need to be configured “JAAS - J2C authentication data”. U can also configure it during data source configuration.



DB mabprod password, ask DB or DEV team



Before deploying App you need to create JDBC Connector, Data source &configure data sources and DB url.

------------------------------------------------------------------------------------------------------------------------

**WAS JDBC**

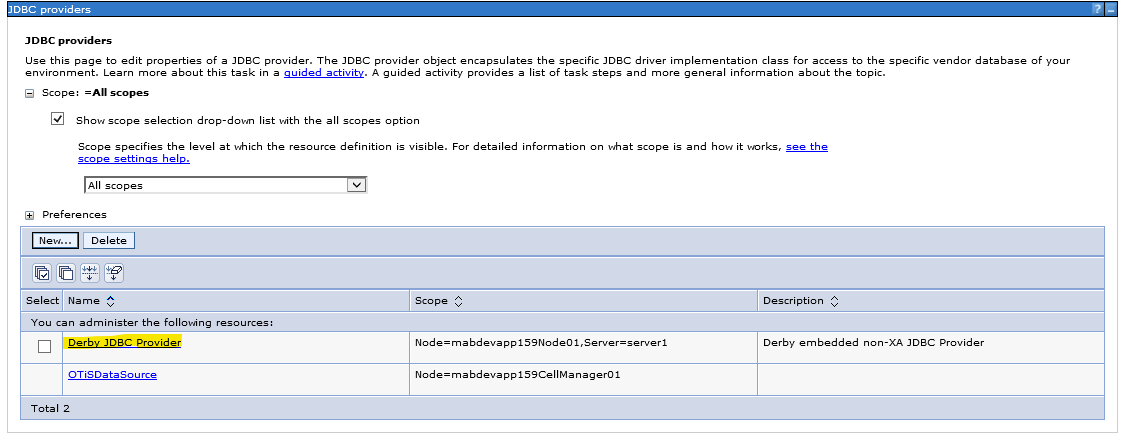
* After configuring JVM parameters we need to create JDBC driver, create data source for JDBC drive & finally configure it (Data source). These properties define how we connect the App to DB server & how to process.

Below images shows:

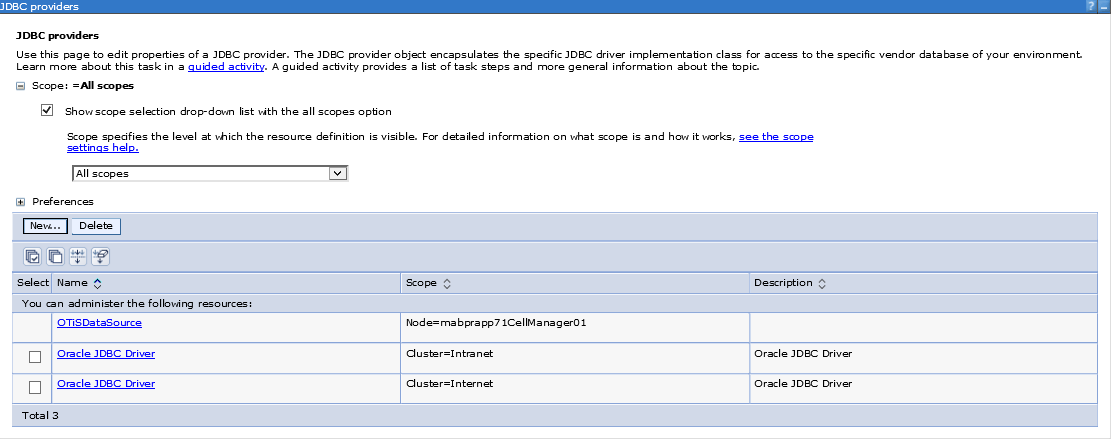
1st image: new

2nd image : Pre- configured

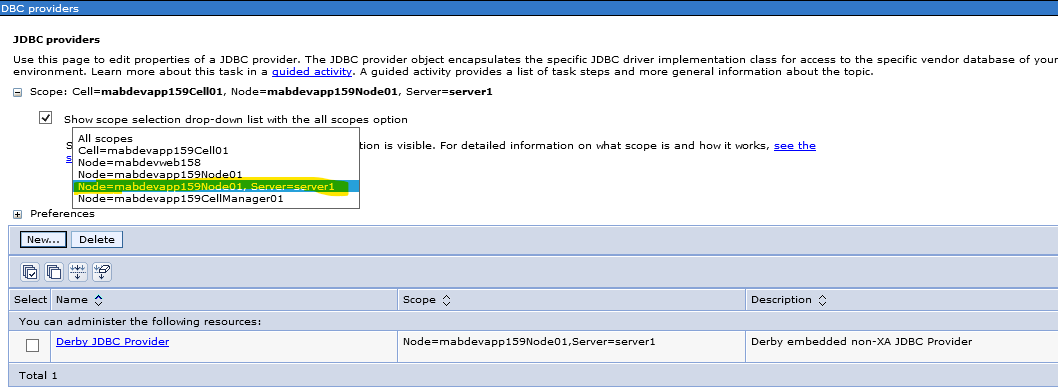
1. JDBC Driver creation
2. Go to Console, Resources, JDBC & JDBC Provider



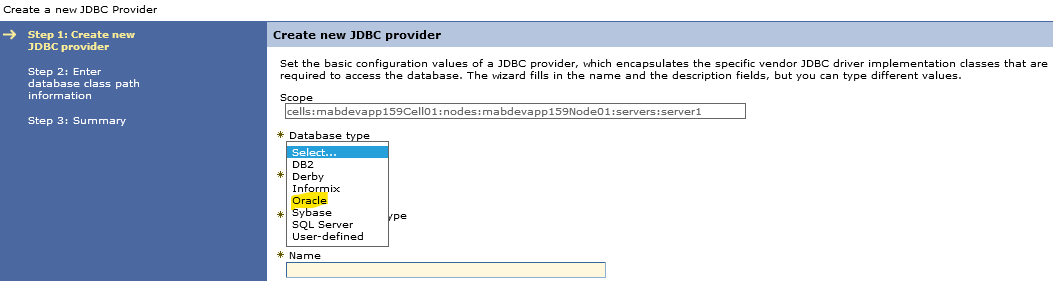
Highlighted one may not exists in newly created server. Doubt.

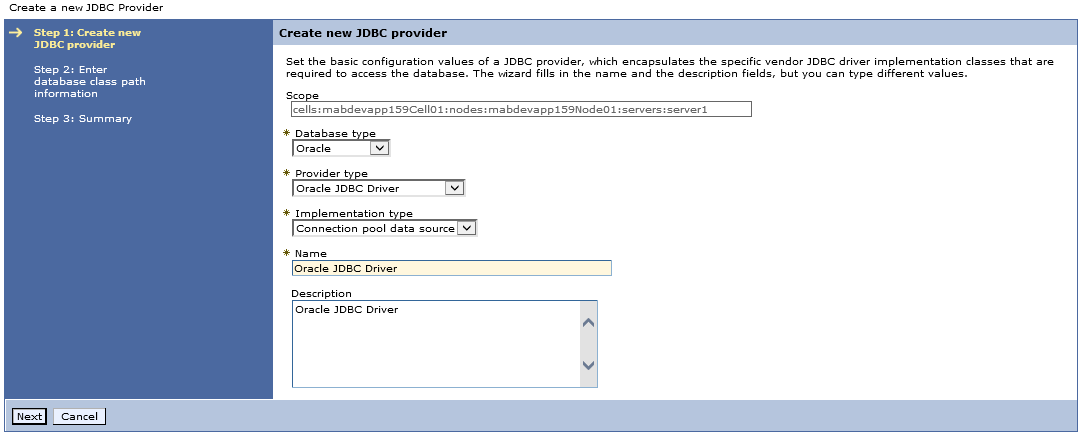


1. Select the Scope and Click on new. If there are clusters you need to configure JDBC for each cluster. If it is a standalone system then you select the Server.

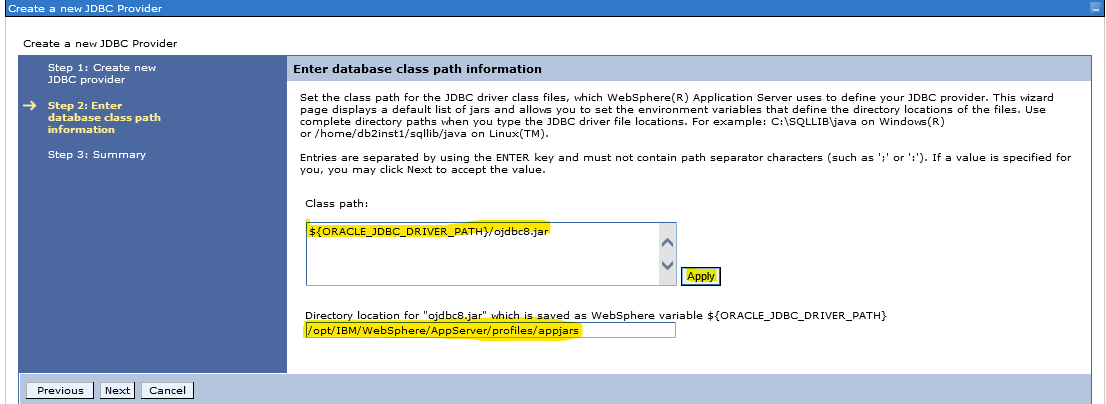


1. Select the database type based on your requirement. Then select the remaining properties based on this DB type and click NEXT

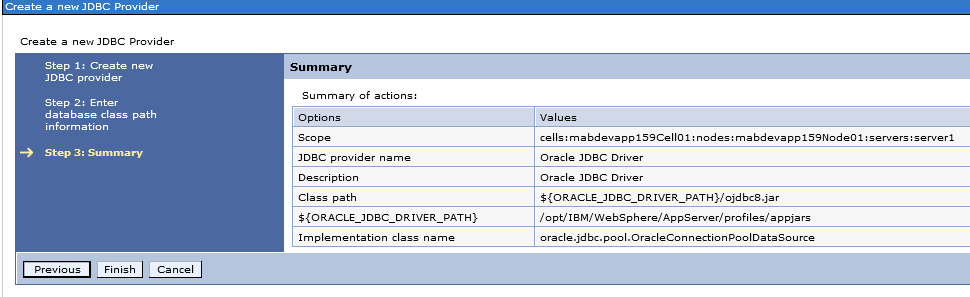




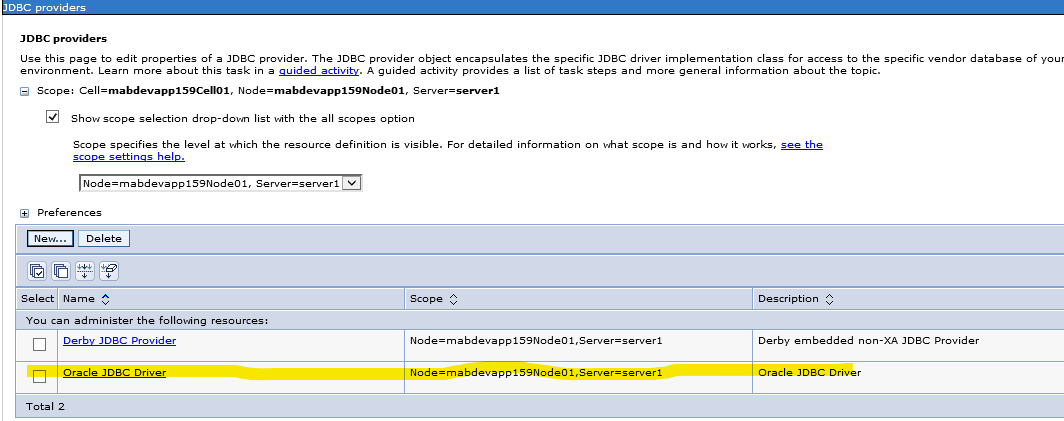
1. In “Enter database class path information ”, here ask developers to provide OJDBC file and its version. Create a folder named “appjars” @ /opt/IBM/WebSphere/AppServer/profiles/appjars& give 755 permission &wasadmin ownership to entire folder (This folder is created in DMGR existed server only in case of non-standalone). Copy the developer provided jar file to that location. Insert the file name including extension in 1st column and its complete path in 2nd column & click apply. Then click NEXT

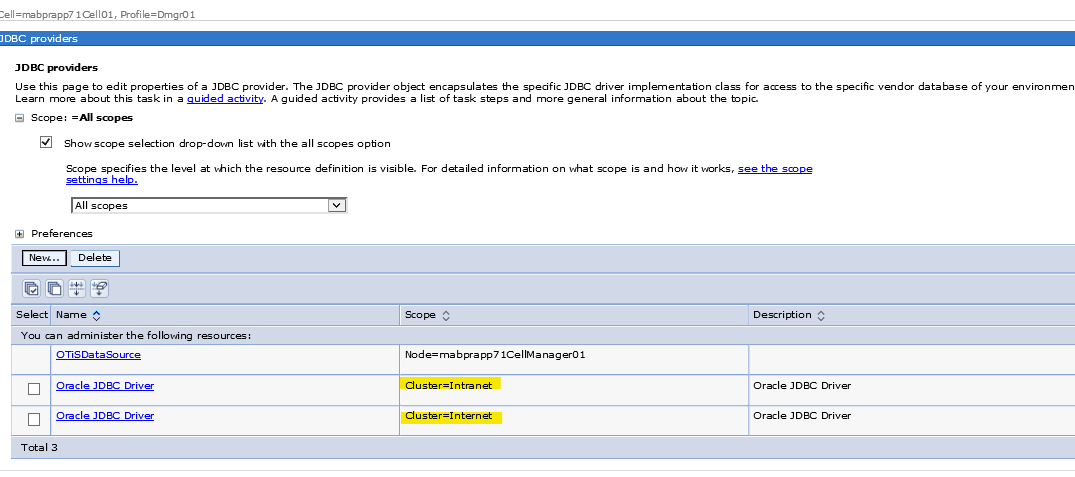


1. Click FINISH, review and save

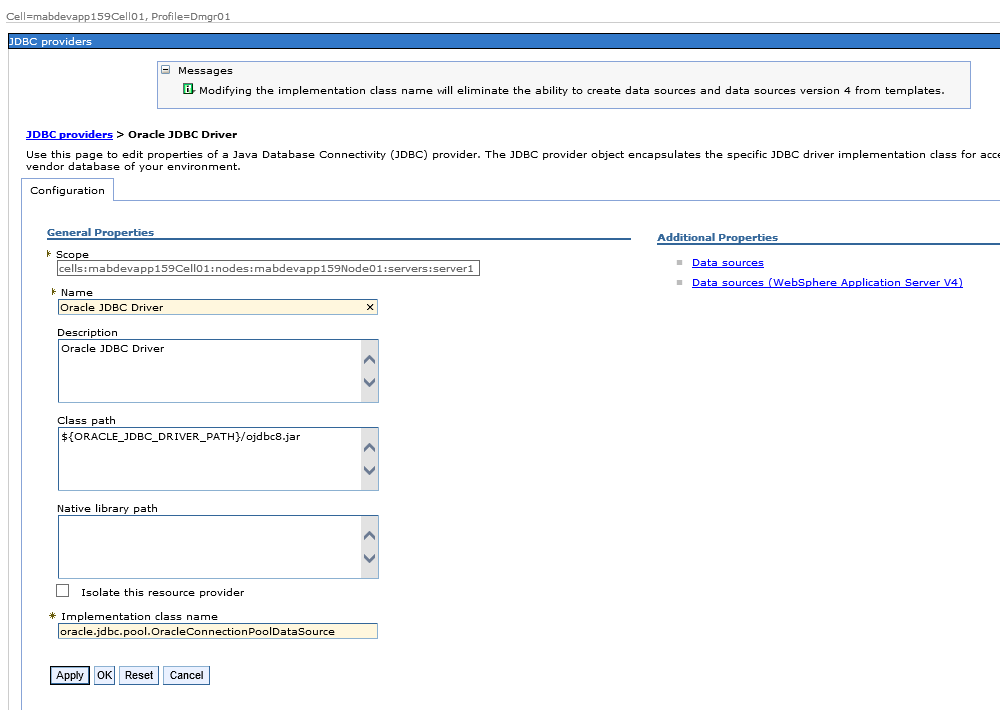


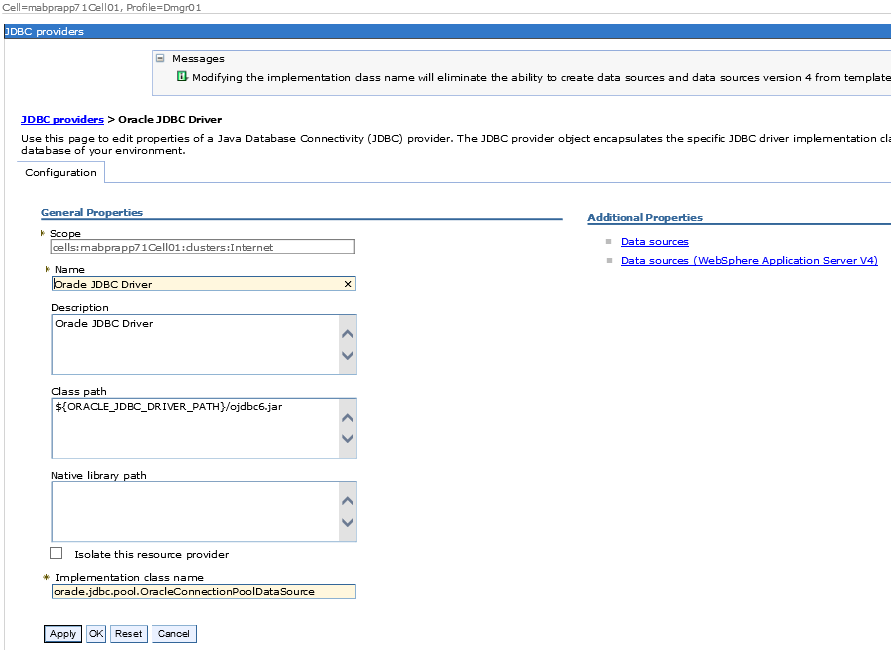
1. Then a DB provider is created by the server name or by cluster name



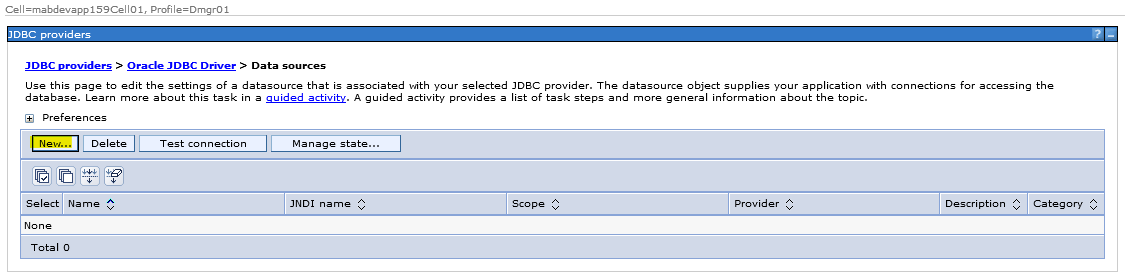


1. Now create the Data source for the Created JDBC driver then later configure it.
2. Now click on the created OJDBC driver. You can check the different versions of OJDBC in two images. It is very important

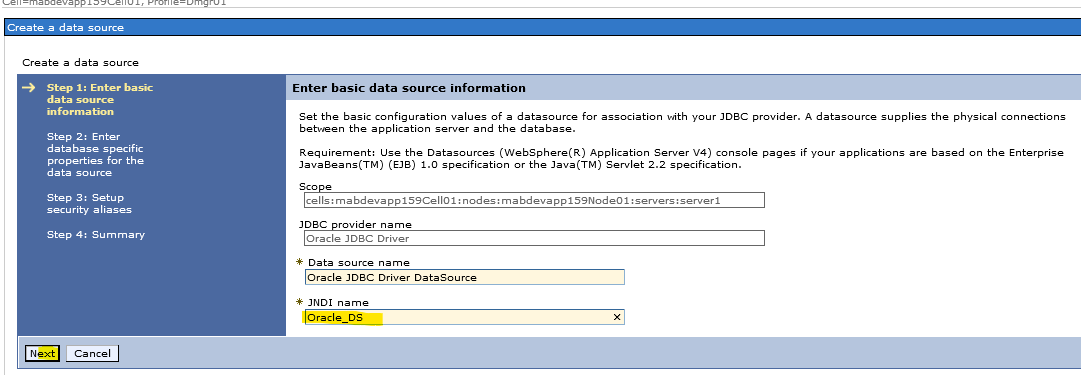




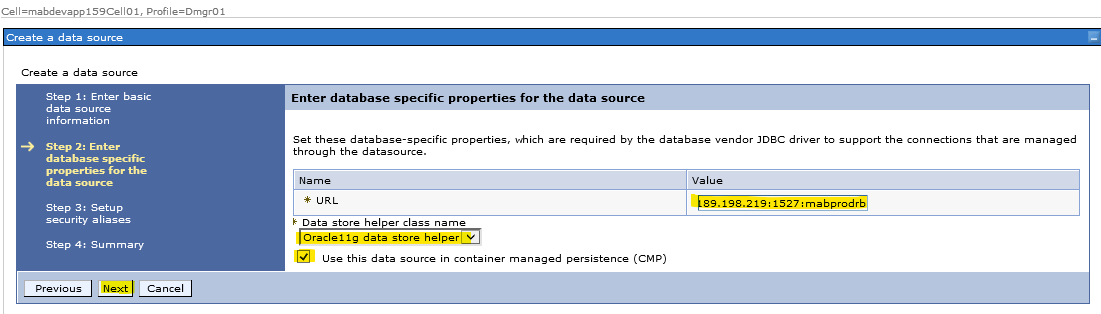
1. Click on DATA SOURCES and create new data source by clicking on NEW



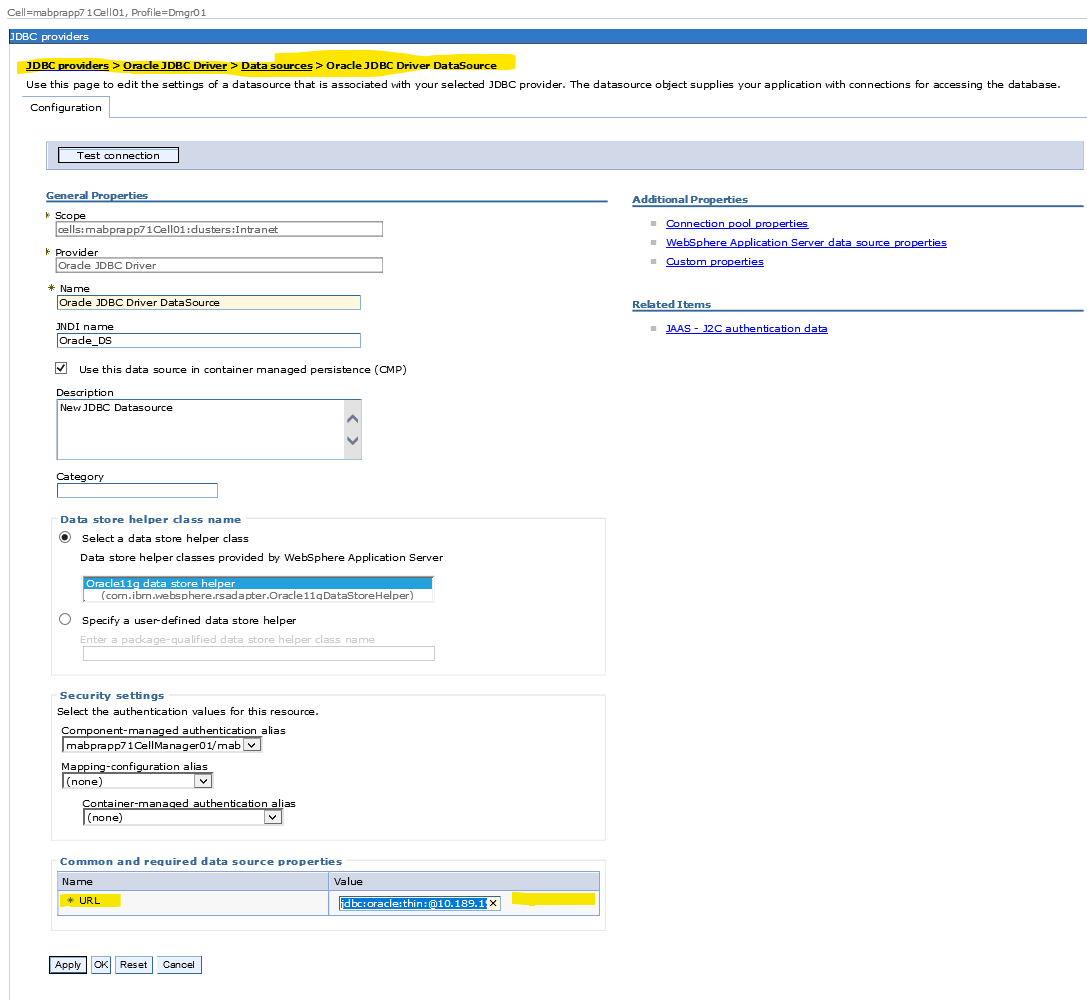
1. Provide the JNDI name and click NEXT. Ask developers for JNDI name. It is case sensitive & Important



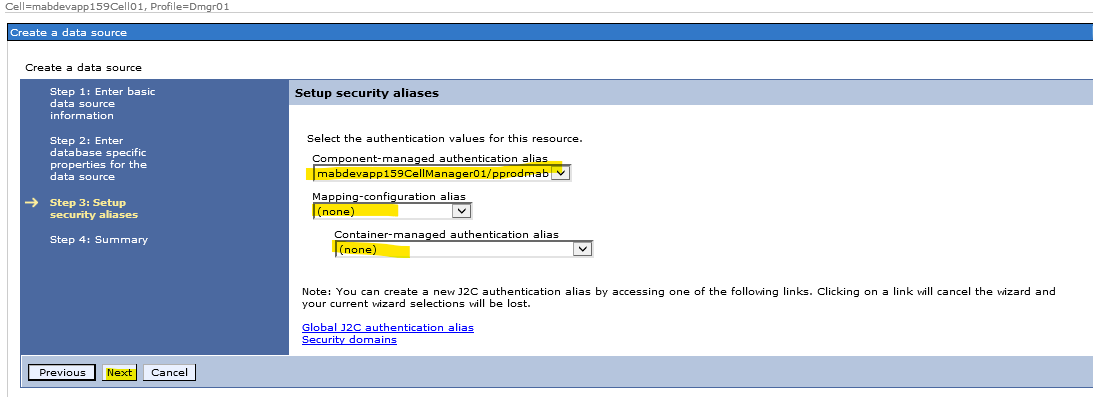
1. Now provide DB URL and click NEXT. Be aware of URL

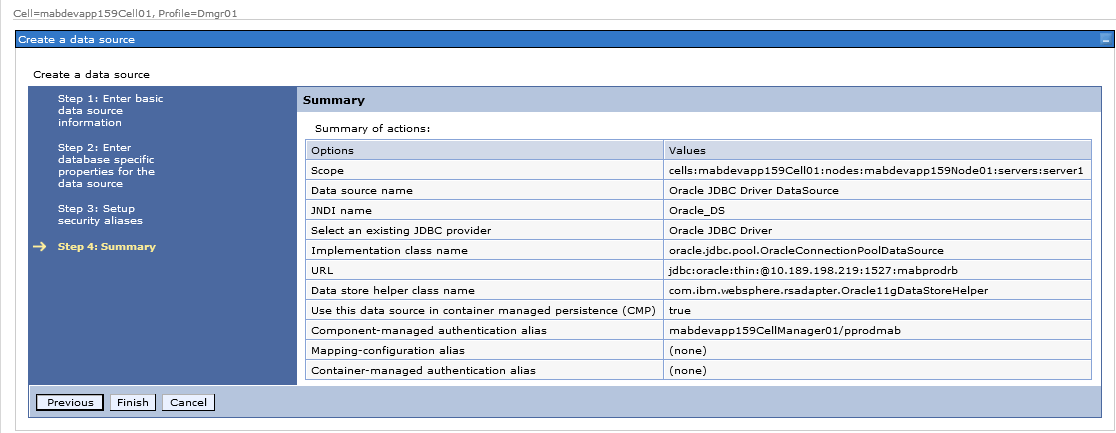


Ask DB team or Developers for this URL or you can also get it from already created datasource as mentioned in below image.

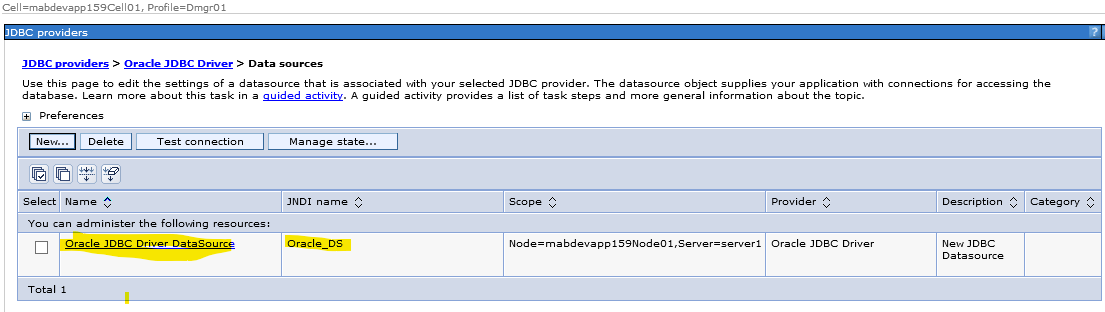


1. Noe select the Security Aliases & click NEXT Then FINISH. Review then save after sync

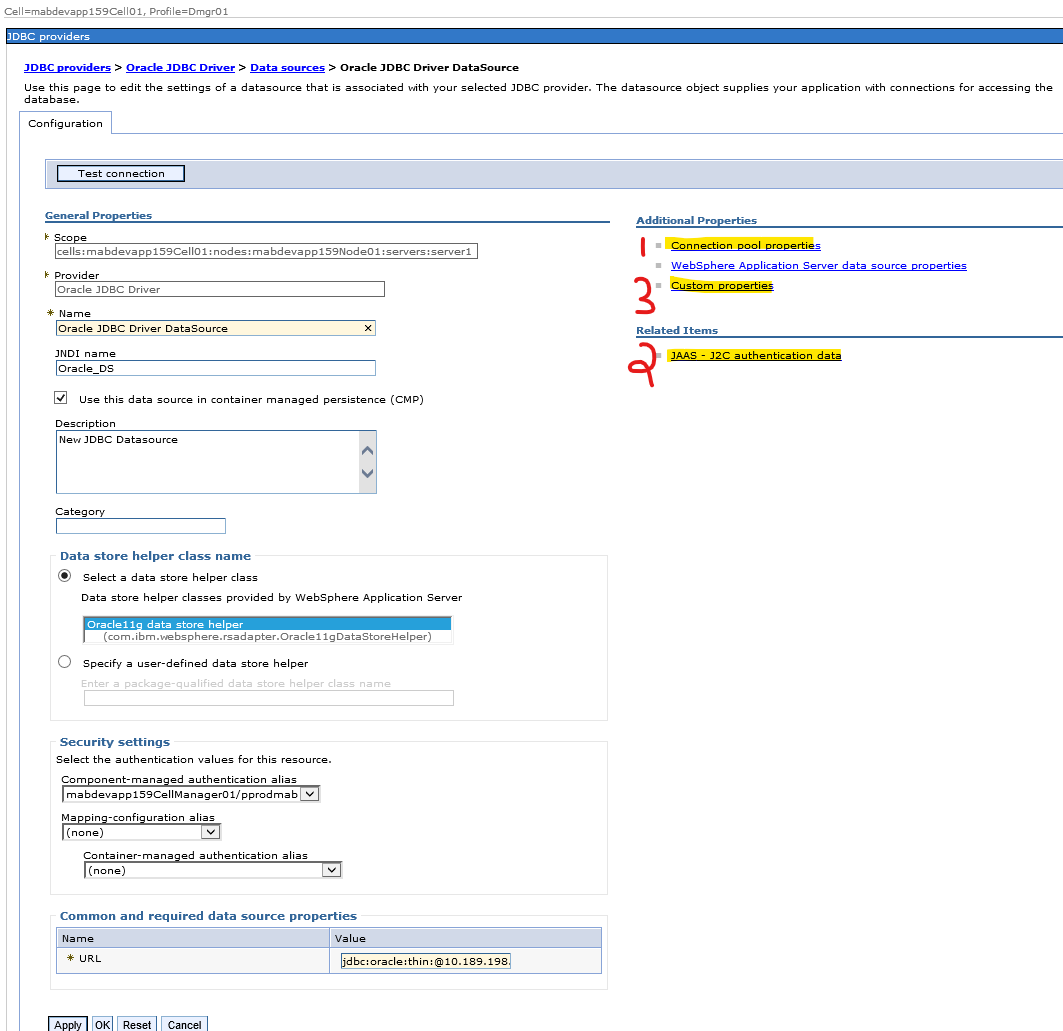




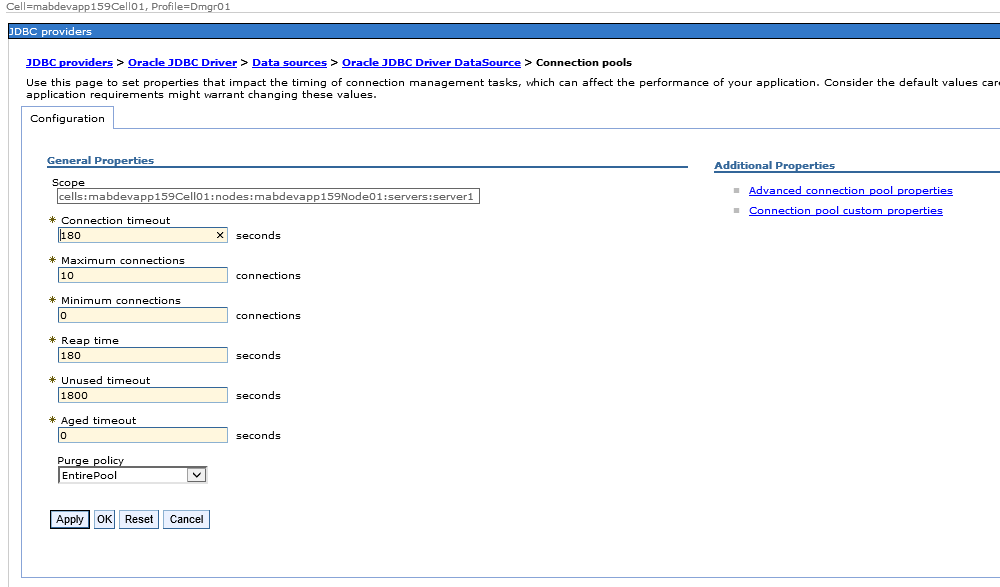
1. Now you got the created data source

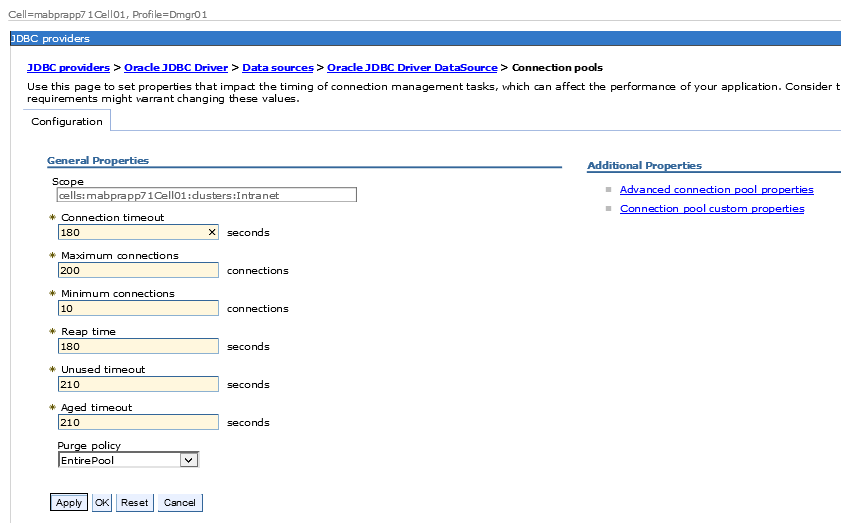


1. Now configure the created Data source.
2. Click on the created data source. Then select Connection pool properties



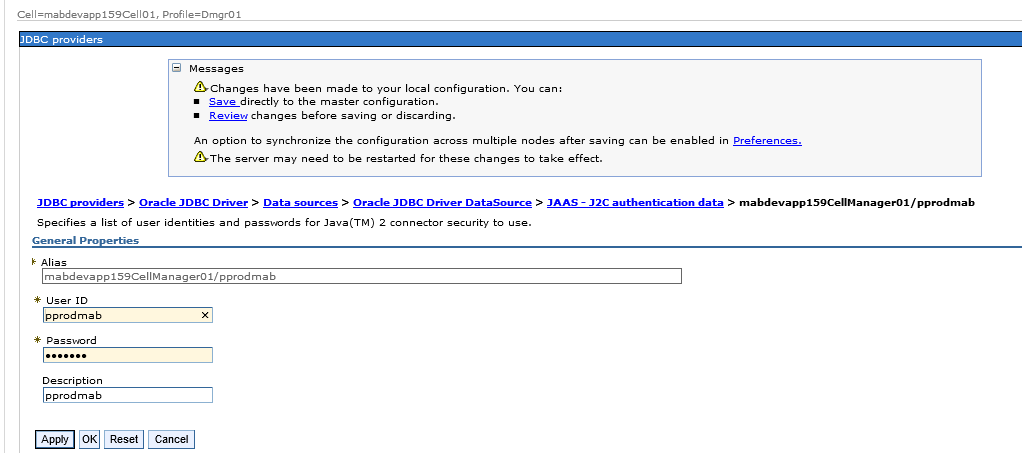
1. Change the parameters as per the project requirement and click OK





Remaining additional properties are not required.

1. Click on “JAAS - J2C authentication data”. You can also configure it during JVM configuration. If you did it there, then you don’t have to do it. It is DB mabprod password, ask DB or DEV teamThen click OK



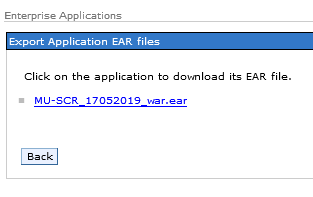
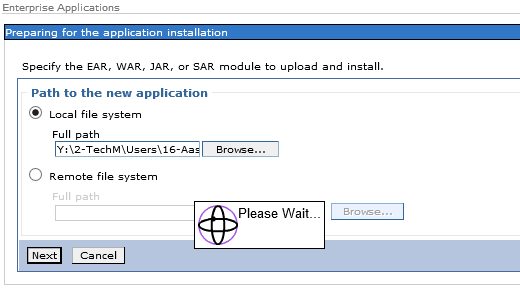
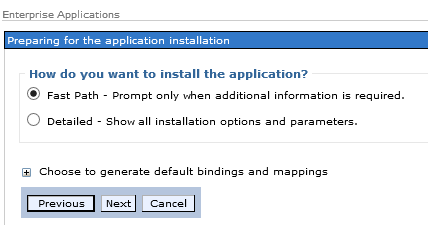
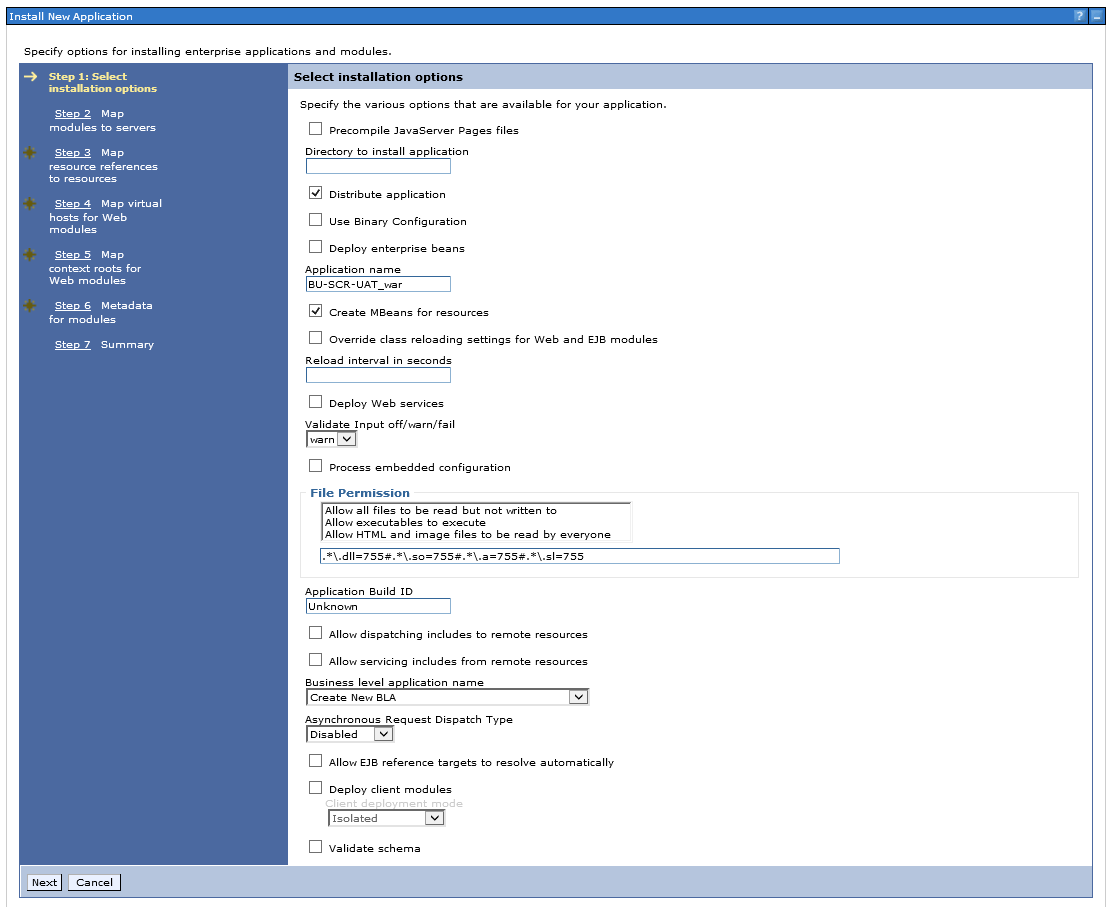
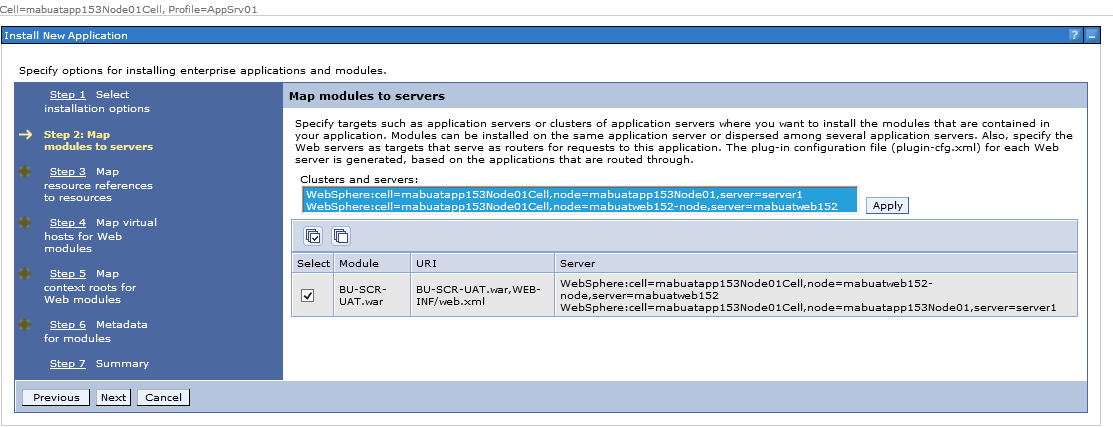
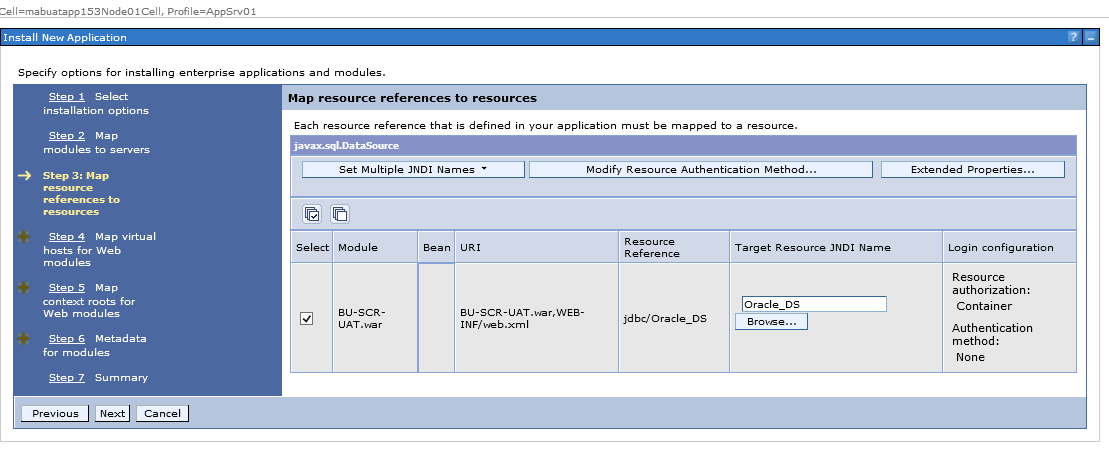
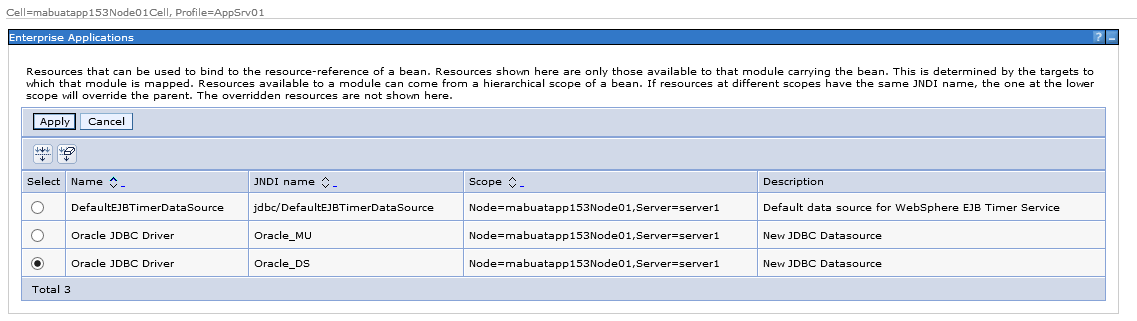
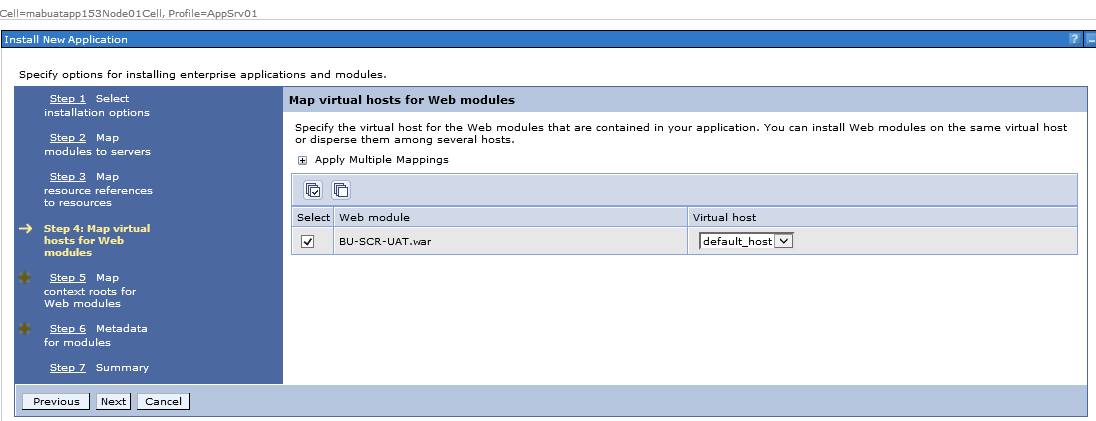
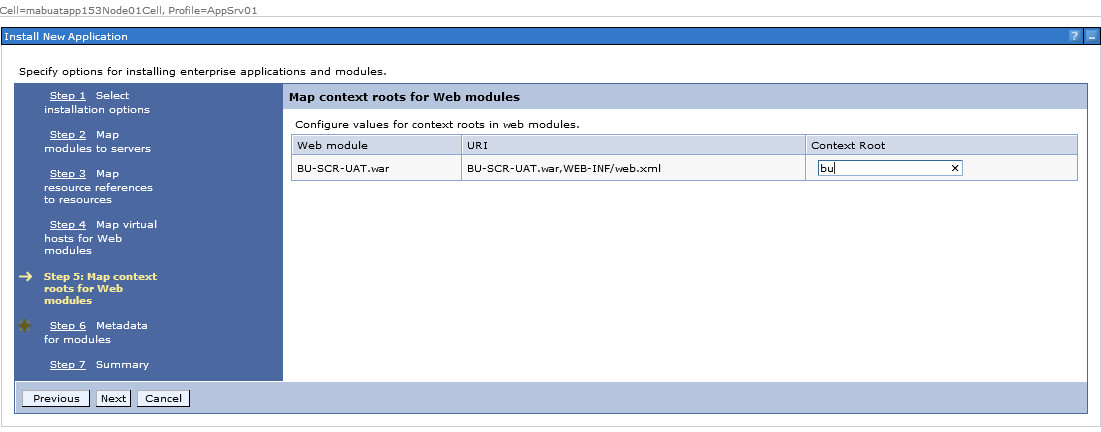
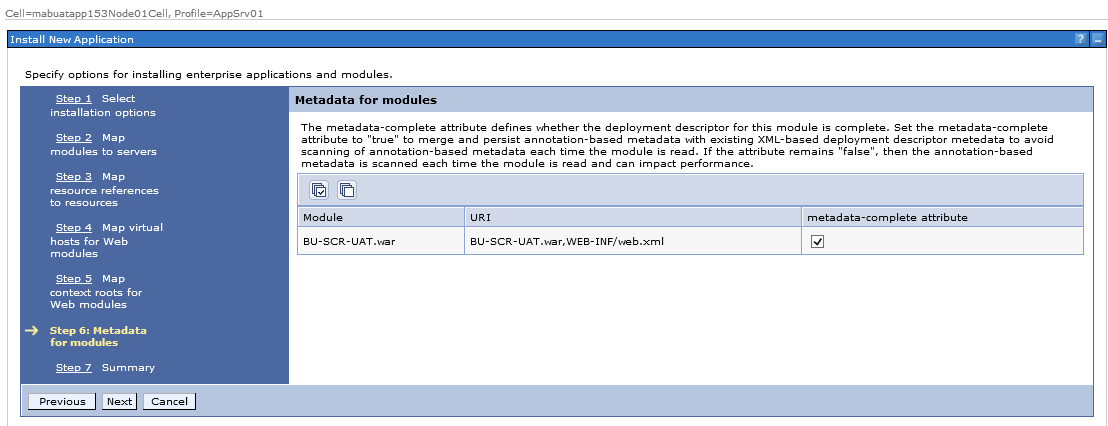
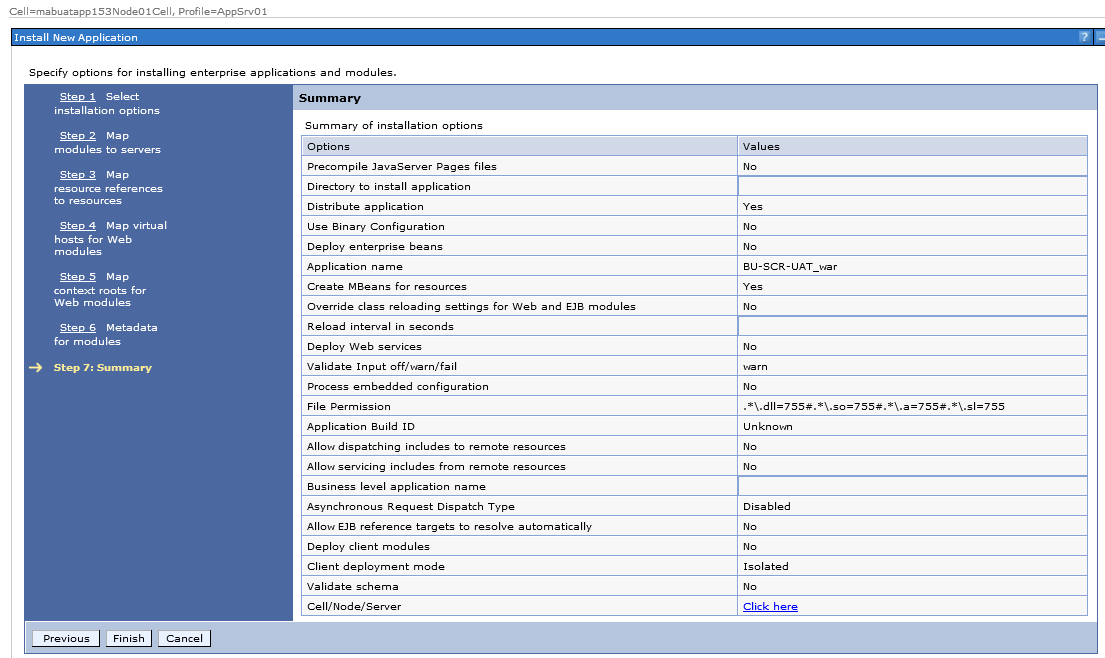
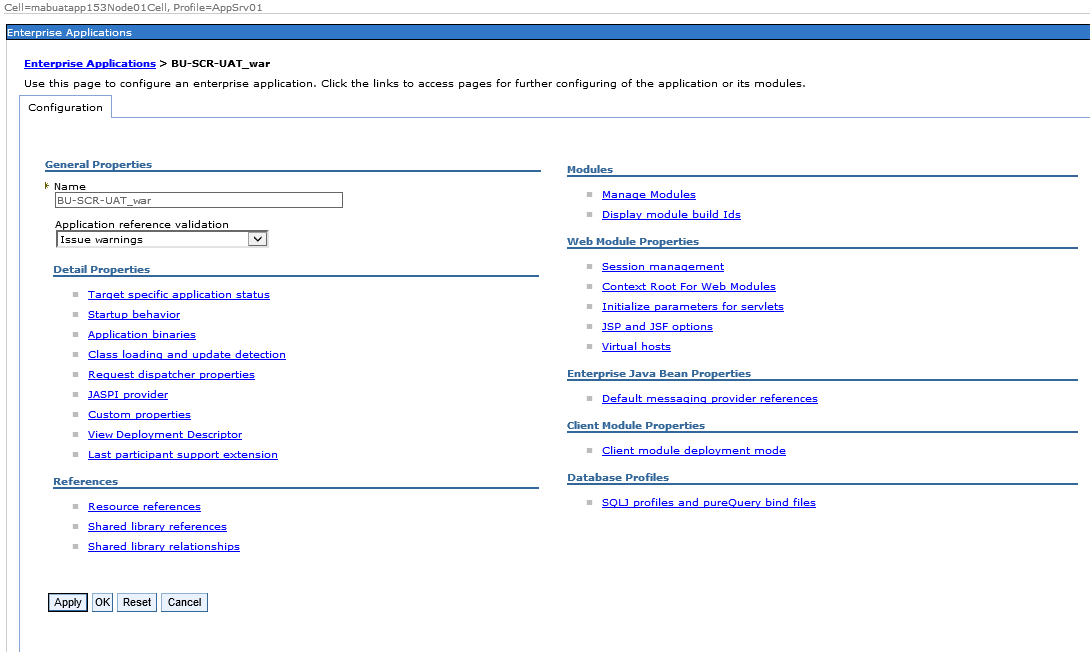
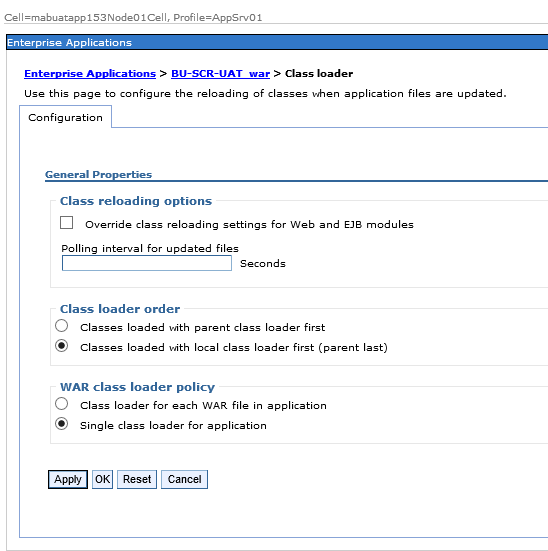
1. Now Custom Properties and these have to be added based on VA.

Now completed.

-------------------------------------------------------------------------------------------------------------

**WAS APP Deployment**

App Deployment (Installation):

* Select the Application and STOP it
* Select the App and take backup by clicking EXPORT button
* 
* Click on that particular .war file to get it downloaded
* Select the App& click UNINSTALL
* Click on INSTALL button to install the App and select the path where the .war files is stored in local system or remote system
* 
* Select the FAST PATH option to move quick and click next
* 
* 
* Click next
* 
* Select the “Clusters and Servers” and check the “Module” click APPLY then click NEXT
* 
* Browse for JNDI name
* 
* Select the proper JNDI name and click APPLY.
* You will be back to Step 3: check it and click NEXT
* 
* Select default\_host and check it and click NEXT
* 
* Give the context root and click NEXT
* 
* Check the option and click NEXT
* Check the summary
* 
* Click FINISH
* Click SAVE
* Click on Installed Application
* 
* Click on “Class loading and update detection ”
* 
* Select the options shown above based on Class loader policy
* Click OK, Review then SAVE.