WebSphere Application Server Network

Table of Contents

Table of Contents

3. ADMINISTRATIVE CONSOLE

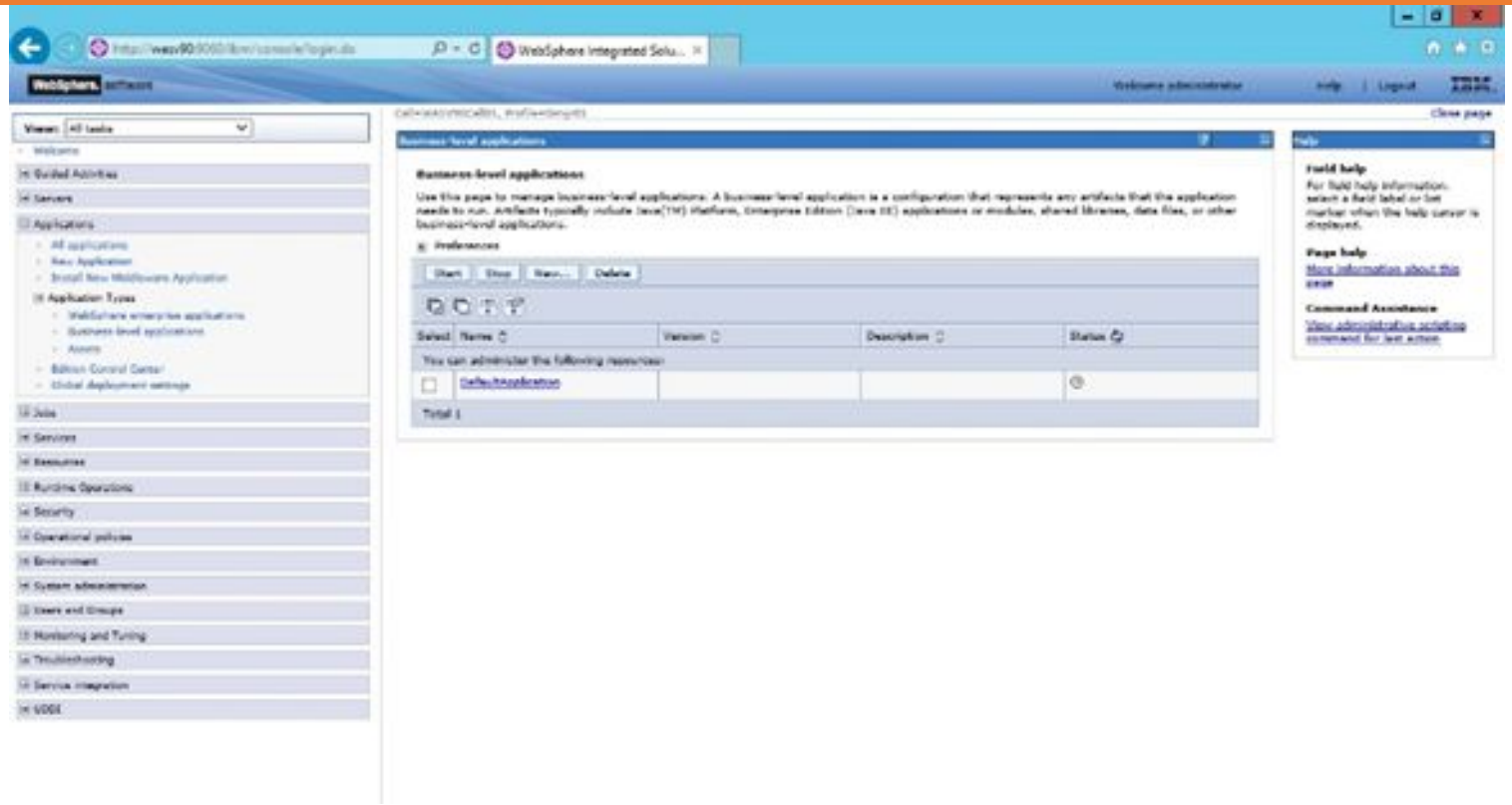
ADMINISTRATIVE CONSOLE

Theory

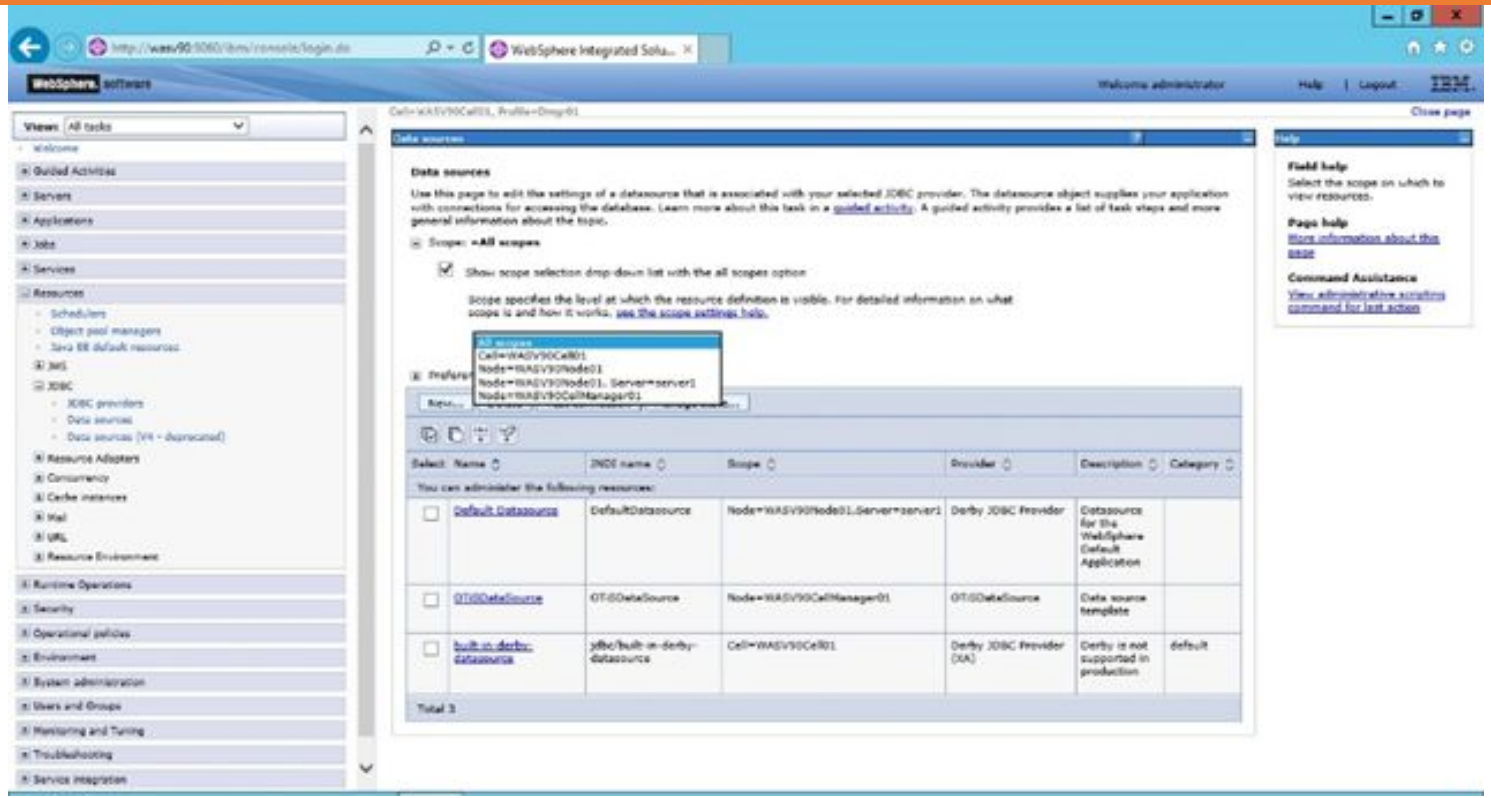
- WebSphere resources can be configured and managed via web based graphical tool.

- Although this tool is called officially as “Websphere Integrated Solutions Console”, you can usually find resources referring this console as “administrative console” or “admin console”.

ADMINISTRATIVE CONSOLE



ADMINISTRATIVE CONSOLE



ADMINISTRATIVE CONSOLE

- In admin console, you can choose the scope level as shown in the image and then click “Apply” to set the level.
- After that, the configuration item you changed will be effective for that level.
- Each scope level configuration is stored in different files (resources.xml) such as <profile_home>/config/cells/cell_name/nodes/<node>/resources.xml for node level or <profile_home>/config/cells/cell_name/resources.xml for cell level.

ADMINISTRATIVE CONSOLE

AIM

The lab exercise contains following tasks:

1. Uninstall & Install Administration Console
2. Secure Administration Console
3. Stop & Start Application
4. Stop & Start Application Server
5. Restart & Stop Node Agent
6. Stop Deployment Manager

ADMINISTRATIVE CONSOLE

Lab Exercise 3: ADMINISTRATIVE CONSOLE

SUMMARY

- IBM WebSphere Application System provides a web based graphical user interface to perform administrative tasks.

- The official name for this tool is “WebSphere Integrated Solutions Console” but in many resources you may see different terms such as “admin console” and “administrative console” for the same interface.

4. ADMINISTRATION VIA SCRIPTS

ADMINISTRATION VIA SCRIPTS

Theory

- WebSphere Application Server provides several command line tools for you to stop, start, and check status of application server processes or nodes.
- Those command line tools can run only local servers and nodes.
- They are located under ‘bin’ directory of application server or deployment manager profiles.

ADMINISTRATION VIA SCRIPTS

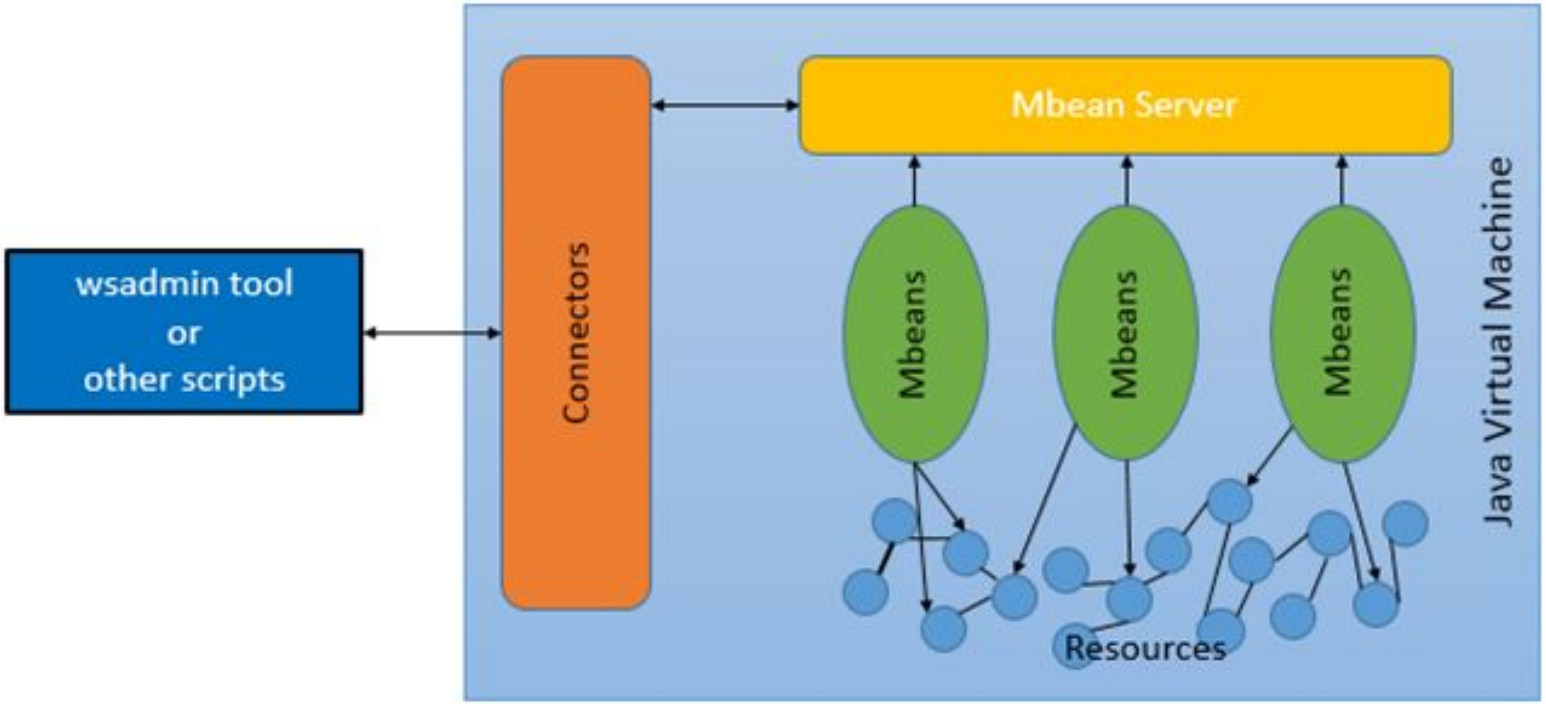
The wsadmin tool has objects available:

- AdminApp contains set of commands to manage installation, removal and editing of the applications.
- AdminConfig contains set of commands to create, remove and modify the elements of WebSphere Application Server

ADMINISTRATION VIA SCRIPTS

- The wsadmin tool can work both on local and remote systems.
- On local mode, you can even work when the application server is down where on remote mode, you have to have the application server up and running.
- On the other hand, local mode can have issues due to multiple access triggered synchronization of changes.

ADMINISTRATION VIA SCRIPTS



ADMINISTRATION VIA SCRIPTS

AIM

In order to achieve this goal, you will need to perform following tasks:

- SOAP Configuration
- Stop & Start Application Server
- Stop & Start Node Agent
- Stop & Start Deployment Manager
- wsadmin scripting with Jacl
- wsadmin scripting with Jython

Lab Exercise 4: ADMINISTRATION VIA SCRIPTS

SUMMARY

- WebSphere Applications Server provides you to use several options to perform administration.
- There are already built in command line tools that gives you the possibility to perform operations such as stop, start, add, remove an application server, node, and cell and so on.
- It is also possible to use scripting for administration purposes.

5. IBM HTTP SERVER AND PLUG-IN

IBM HTTP SERVER AND PLUG-IN

Theory

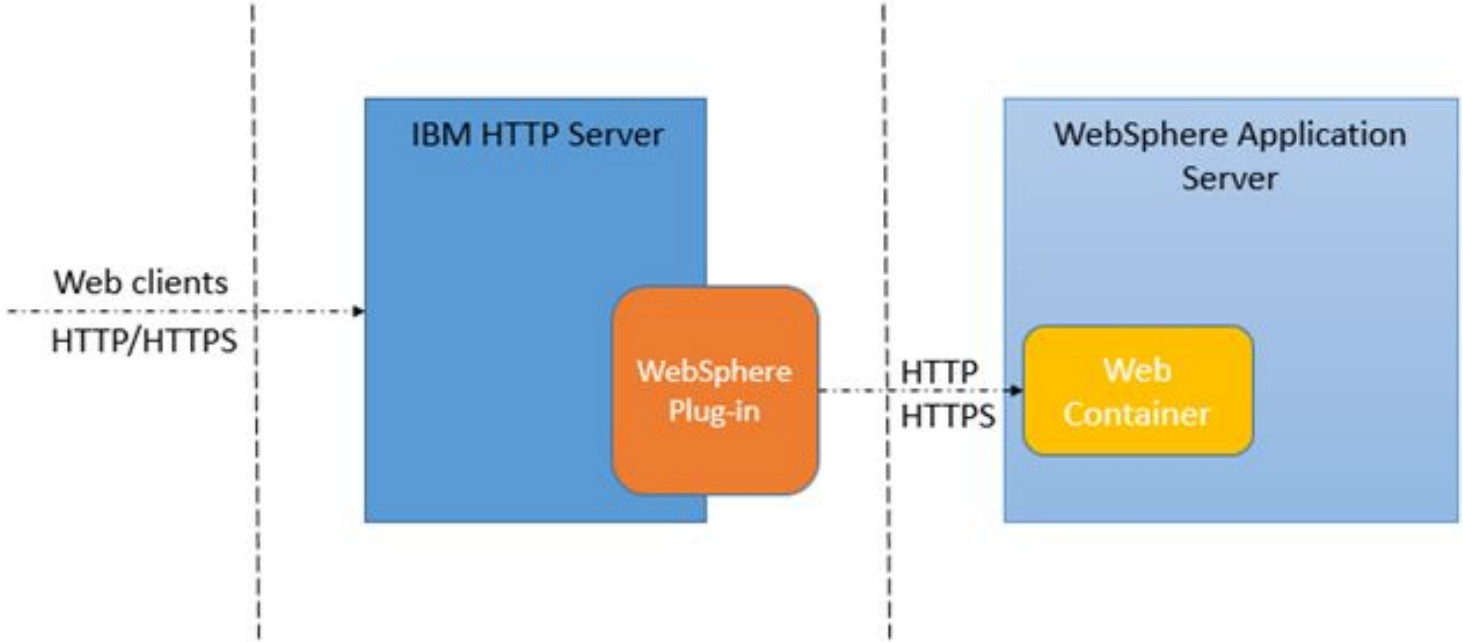
- Web servers are one of the most important part of the multi-tier architecture and they provide scalability, security and performance.
- Web servers are defined to WebSphere Application Server as either managed or unmanaged node and in both configurations, they can be managed through administrative console.

IBM HTTP SERVER AND PLUG-IN

There are couple features added to Apache HTTP Server:

- WebSphere Integrated Solutions Console support
- Consistent installation via IBM Installation Manager
- Fast Response Cache Accelerator (FRCA), that improves serving of static contents, is available for AIX 5 and later and certain Windows operating systems.
- Dynamic content generation with FastCGI
- Multiple language and platform support

IBM HTTP SERVER AND PLUG-IN



IBM HTTP SERVER AND PLUG-IN

WebSphere Customization Toolbox, (WCT), is a set of tools to manage, configure and migrate different parts of WebSphere Application Server. It has two different offerings with different combinations of tools:

- Embedded: It is installed during WebSphere Application Server installation and contains the “Profile Management Tool” and the “Configuration Migration” tool.

IBM HTTP SERVER AND PLUG-IN

AIM

- The aim of the lab exercise is to install IBM HTTP Server, WebSphere Web Server Plug-ins and WebSphere Customization Toolbox using IBM Installation Manager.

- You will need to configure the web server and the Plug-ins to work together.

IBM HTTP SERVER AND PLUG-IN

In order to achieve this, you need following steps:

- Install IBM Installation Manager
- Install IBM HTTP Server
- Install WebSphere Plug-ins
- Install WebSphere Customization Toolbox
- Create web server definition using scripts created by WCT
- Generate and propagate plug-in for the default application

Lab Exercise 5: IBM HTTP SERVER AND PLUG-IN

SUMMARY

- Web servers are one of the most important components of the multi-tier architectures.
- Numerous numbers of web servers such as Apache HTTP Server, Microsoft IIS, and IBM HTTP Server, are supported by IBM WebSphere Application Server.
- WebSphere Plug-ins provides better performance and security by adding a smart layer between the web server and WebSphere Application Server.

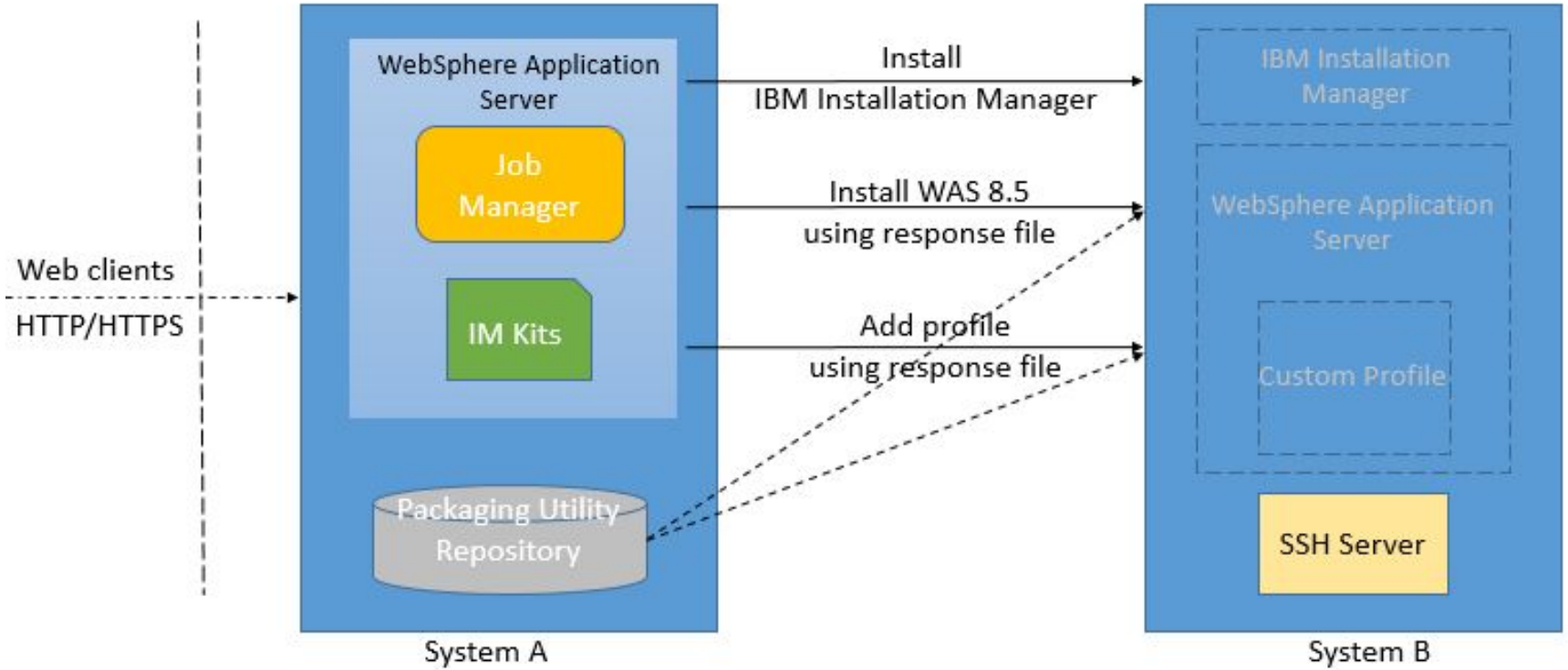
6. CENTRALIZED INSTALLATION MANAGER

CENTRALIZED INSTALLATION MANAGER

Theory

- Centralized Installation Manager (CIM) was first introduced in WebSphere Virtual Enterprise version 6.1 and WebSphere Application Server version 7.
- CIM can be accessed from the deployment manager.
- You can use CIM to install WebSphere Application Server and apply maintenance

CENTRALIZED INSTALLATION MANAGER



CENTRALIZED INSTALLATION MANAGER

You can install following products using CIM:

- IBM Installation Manager
- WebSphere Application Server
- IBM HTTP Server
- Application Clients
- DMZ Security Proxy Server
- WebSphere Web server Plug-ins
- WebSphere Customization Toolbox

CENTRALIZED INSTALLATION MANAGER

You can submit the following CIM jobs from the Jobs manager:

- Inventory, you can use this collect information for installation.
- Install, update and uninstall Installation Manager.
- Manage offerings, you can use this job to install, update or uninstall the products listed above.

CENTRALIZED INSTALLATION MANAGER

For the following tasks, you can use IBM Packaging Utility:

- Generate a new network repository to be used in your local network.
- You can copy multiple packages to one repository.
- You can copy multiple version of the same product to one repository to be used to update products.
- You can save disk space to have all packages to store in one and only place.
- You can enable your repository to be accessible thru HTTP, HTTPS, and FTP or file sharing

CENTRALIZED INSTALLATION MANAGER

AIM

In order to achieve these, you need to complete following task:

- Install and Configure IBM Packaging Utility
- Add Installation Manager Installation kits
- Register a remote host
- Install IBM Installation Manager on remote host
- Install WebSphere Application Server on remote host
- Define a profile on remote host

Lab Exercise 6: CENTRALIZED INSTALLATION MANAGER

SUMMARY

- Centralized Installation Manager gives you the possibility to install, update, and uninstall WebSphere Application Server and supplementary products to remote hosts without connecting them.

- With the support of Packaging Utility, you even don’t need to copy files over network, which saves time, disk space and CPU utilization.

7. CONNECT TO A DATABASE

CONNECT TO A DATABASE

Theory

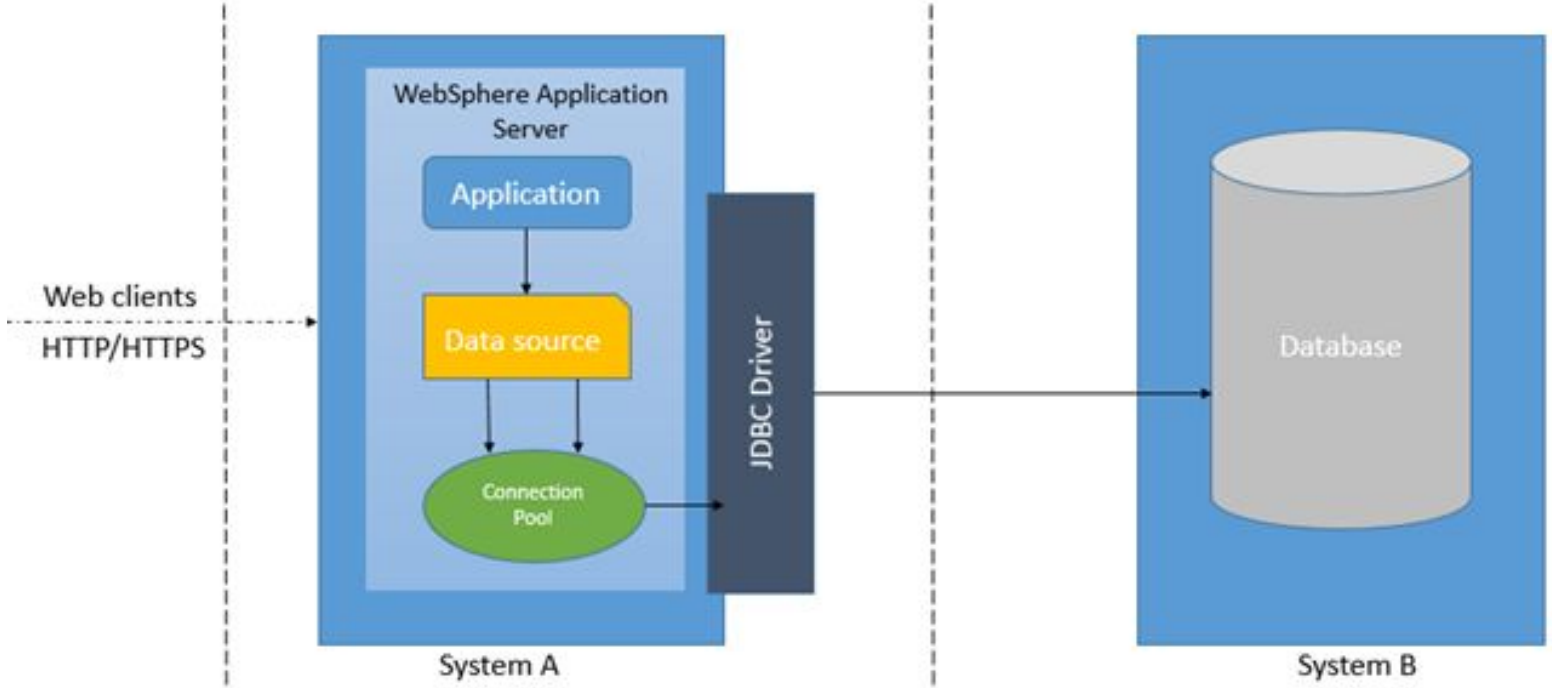
- Business applications running on WebSphere Application Server requires access to database systems.
- In order to access to databases, we need to define a data source for each database.

CONNECT TO A DATABASE

For better understanding of the tasks required to access databases, we need to understand following terms:

- JDBC, Java Database Connectivity, is a Java-based API technology to access databases. You can connect to a database, query and change data in a database.
- Data source, is referred to the name of the configuration properties of the database in order to connect and run queries.

CONNECT TO A DATABASE



CONNECT TO A DATABASE

In order to provide access to a database from an application that runs on WebSphere Application Server, you need to follow 2 basic steps:

1. Create and J2C authentication alias to store and encrypt credentials which will be used to connect to the database.
2. Create a JDBC provider that contains information of database drivers, type of access and location of the files needed for the implementation.
3. Create a data source that defines which JDBC driver to use, database name and location, and other connection properties

CONNECT TO A DATABASE

AIM

- A running database instance
- Hostname or IP address of the server where the database runs
- Port number to connect to the database
- Sample database name
- Username and password to connect to the database server and the database.

CONNECT TO A DATABASE

You need to follow the tasks below:

- Task 1: Create an authentication alias
- Task 2: Create JDBC provider
- Task 3: Create data source
- Task 4: Enable JDBC trace logs

CONNECT TO A DATABASE

Lab Exercise 7: CONNECT TO A DATABASE

SUMMARY

- Business applications running on WebSphere Application Server requires access to database systems.
- In order to access to databases, we need to define a data source for each database.
- You need to create a JDBC provider that contains information of database drivers, type of access and location of the files needed for the implementation and to create a data source that defines which JDBC driver to use, database name and location, and other connection properties

8. CONNECT TO A MESSAGING PROVIDER

LEARNING VOYAGE

CONNECT TO A MESSAGING PROVIDER

Theory

- WebSphere Application Server supports asynchronous messaging that allows to create, send, receive and read requests.

- Asynchronous messaging support is based on Java Message Service (JMS) and Java EE Connector Architecture (JCA).

CONNECT TO A MESSAGING PROVIDER

A JMS message consist of three parts;

- Header, must exist in every JMS message and it is assigned automatically mostly by JMS provider when the message put to a JMS destination.
- Properties, are optional and can be application related, provider related and standard properties. Properties provide an efficient mechanism to filter application defined messages.
- Body, contains the main information which needs to be exchanged

CONNECT TO A MESSAGING PROVIDER

There are 6 different types of JMS messages based on the data it contains:

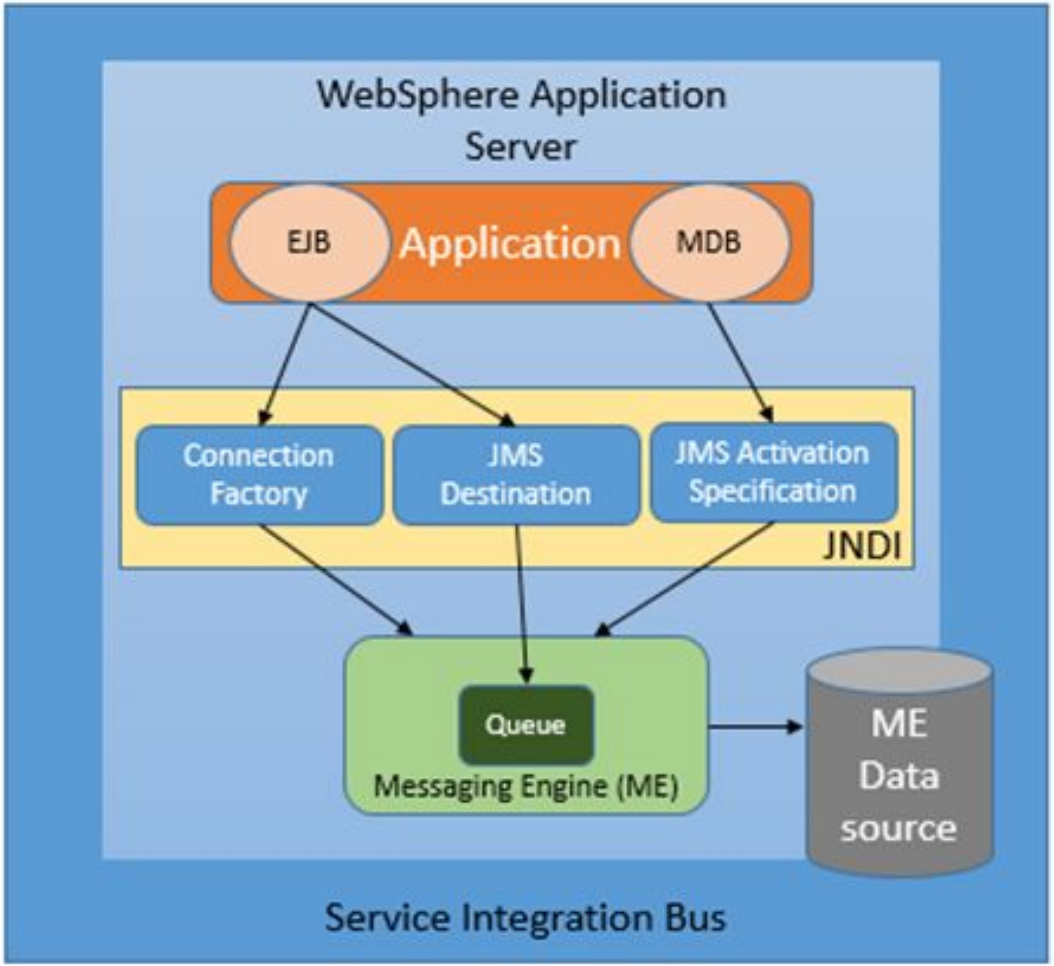
- Message, it is the base class and used for event notification.
- BytesMessage, stores the data as sequence of bytes and used for exchanging data in a format that is native to the application.
- TextMessage, stores the data as a string and used for simple text message exchanging.

CONNECT TO A MESSAGING PROVIDER

- Any application can exchange messages with any application using destination.
- A message producing application can produce messages for a destination regardless of the messaging engine that the producer uses to connect to the bus.
- A message consuming application can consume messages from a destination regardless of which messaging engine the consumer uses to connect to the bus.

CONNECT TO A MESSAGING PROVIDER

- A connection factory is an object that is used by JMS client to establish the connection to a JMS provider.
- It is stored and managed object in a JNDI namespace.
- A connection factory can create connections to one messaging provider only.
- If you need to connect to a different provider, you need to create a new connection factory.


WebSphere Application Server
EJB
Application
MDB
Connection Factory
JMS Destination
JMS Activation Specification
JNDI
Queue
Messaging Engine (ME)
ME Data source
Service Integration Bus

CONNECT TO A MESSAGING PROVIDER

AIM

In order to achieve this, you need to complete following steps:

1. Create a Service Integration Bus
2. Configure JMS connection factory
3. Add a Bus member
4. Configure JMS destination
5. Configure JMS queue
6. Configure JMS Activation Specifications

CONNECT TO A MESSAGING PROVIDER

Lab Exercise 8: CONNECT TO A MESSAGING PROVIDER

SUMMARY

- WebSphere Application Server supports asynchronous messaging that allows to create, send, receive and read requests between applications.
- For this communication, you need to configure different components of JMS depending on your application.
- WebSphere can work with default messaging provider, or with WebSphere MQ or with many other messaging software

9. STATIC CLUSTER

STATIC CLUSTER

Theory

- A cluster, generally, referred as a logical grouping of servers or nodes to improve performance and availability.

- IBM WebSphere Application Server Network Deployment has a built-in clustering function that refers to grouping of application servers that host same enterprise application(s).

STATIC CLUSTER

Clusters can be configured in different ways to serve your business needs:

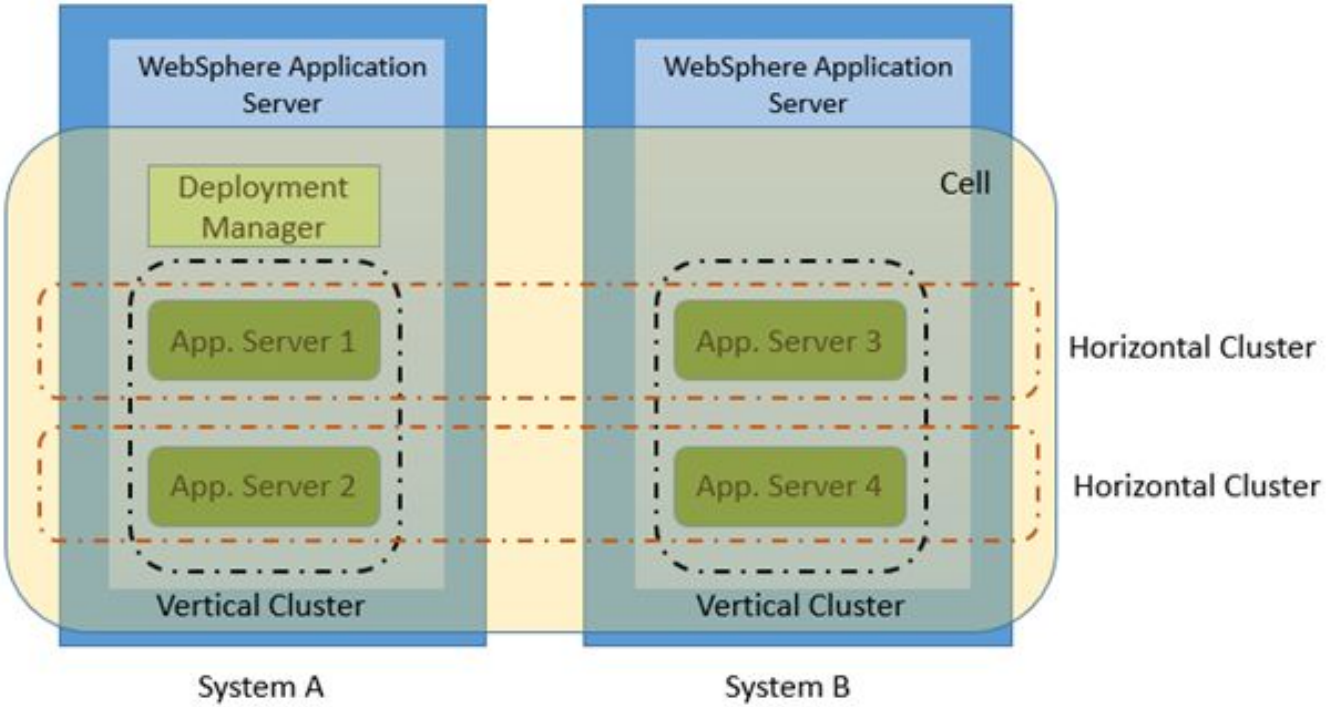
- Vertical clustering
- Horizontal clustering:

STATIC CLUSTER

- **Vertical clustering**: In this type of cluster, all members of an application server cluster reside on the same physical machine or node.

- You can prefer vertical clustering to use machine's power more efficiently.

STATIC CLUSTER



STATIC CLUSTER

- **Horizontal clustering:** In this type of clustering, cluster members are created on multiple physical machines or nodes.

- Horizontal clustering allows you to run an enterprise application on several machines to have usage of resources on distributed systems.

STATIC CLUSTER

Clustering application servers provides workload management (WLM) and failover ability. In a typical cluster, following components can be workload managed:

- Http requests and web servers
- Https requests and plug-in
- EJB requests

STATIC CLUSTER

AIM

- Federate a node to a cell
- Create a static cluster
- •Change weight of a cluster member

STATIC CLUSTER

Lab Exercise 9: STATIC CLUSTER

SUMMARY

- IBM WebSphere Application Server Network Deployment has a built-in clustering function that refers to grouping of application servers that host same enterprise application(s).

- There are two types of clustering that are vertical clustering and horizontal clustering.

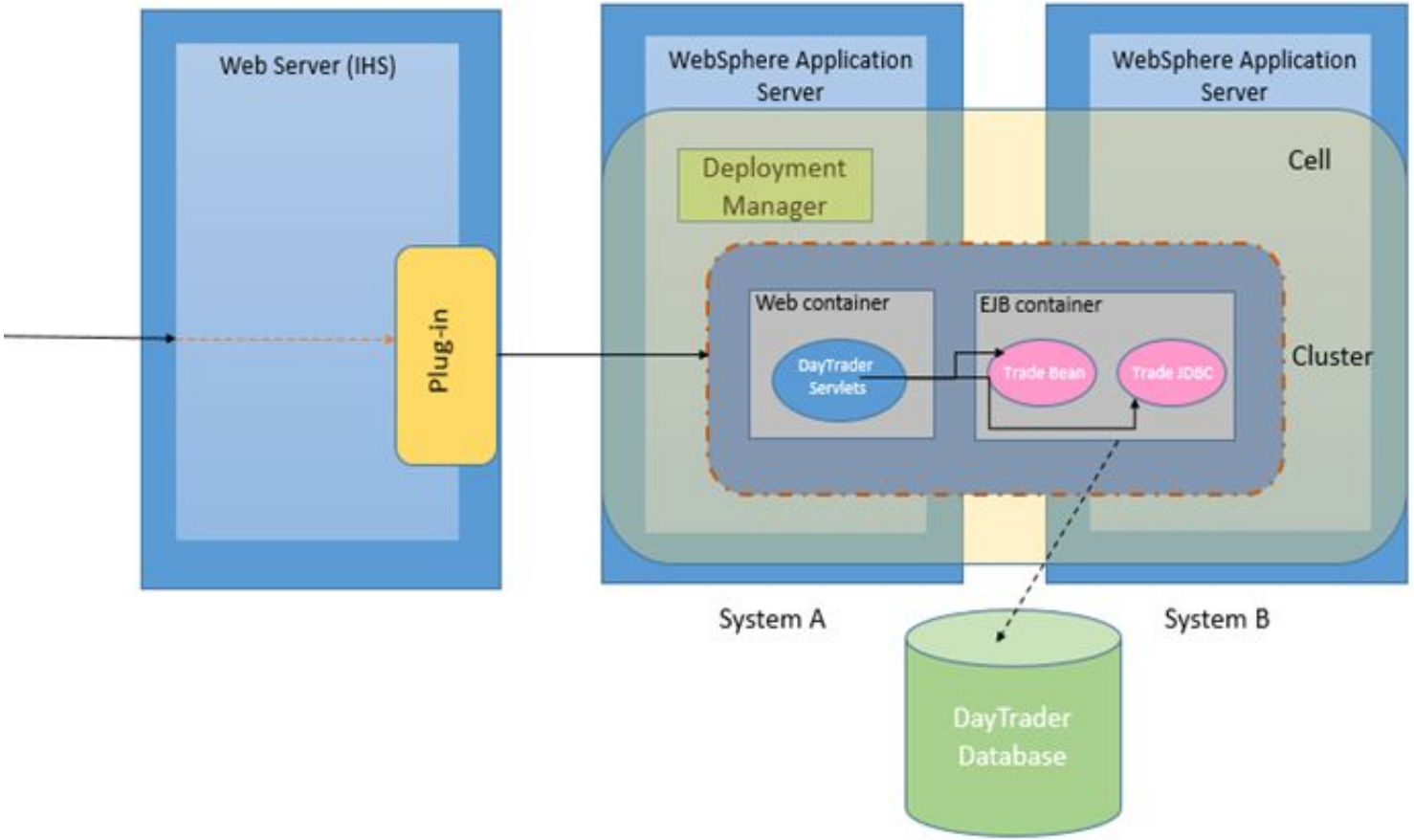
10. APPLICATION DEPLOYMENT

APPLICATION DEPLOYMENT

Theory

WebSphere Application Server supports three different types of applications to be deployed that are:

- Enterprise Applications, that can be an EAR (Enterprise Archive), a WAR (Web Application) file or a JAR (Java Archive) file.
- When an EJB module or Web module is installed, WebSphere Application Server will automatically pack it as an EAR file.
- It is similar case for WAR files, meaning, deployment process will convert WAR file in to EAR automatically.


Web Server (IHS)
Plug-in
WebSphere Application Server
WebSphere Application Server
Deployment Manager
Cell
Web container
EJB container
DayTrader Servlets
Trade Bean
Trade JDBC
Cluster
System A
System B
DayTrader Database

APPLICATION DEPLOYMENT

- Before deploying an application, it is better to setup JDBC and JMS settings that will be used by the application.
- During installation of an application, you can set many options specific to your deployment
- One of the most important steps during the installation is mapping the application modules.

APPLICATION DEPLOYMENT

AIM

- Day Trader is a benchmark application that simulates an online stock trading system.
- It was originally developed by IBM and later donated to the Apache Geronimo community.
- In this lab exercise, we will install Day Trader application and to achieve this goal, we need to follow 2 steps:

Task 1: Setup JMS and JDBC settings

Task 2: Deploy a sample application

Lab Exercise 10: APPLICATION DEPLOYMENT

SUMMARY

- WebSphere Application Server supports three different types of applications to be deployed that are enterprise applications, business level applications and assets.
- You can deploy those applications from your system or you can send the files to the system that hosts deployment manager.
- Before installing the application, you need to setup the environment according to the needs of application like JMS and JDBC settings.

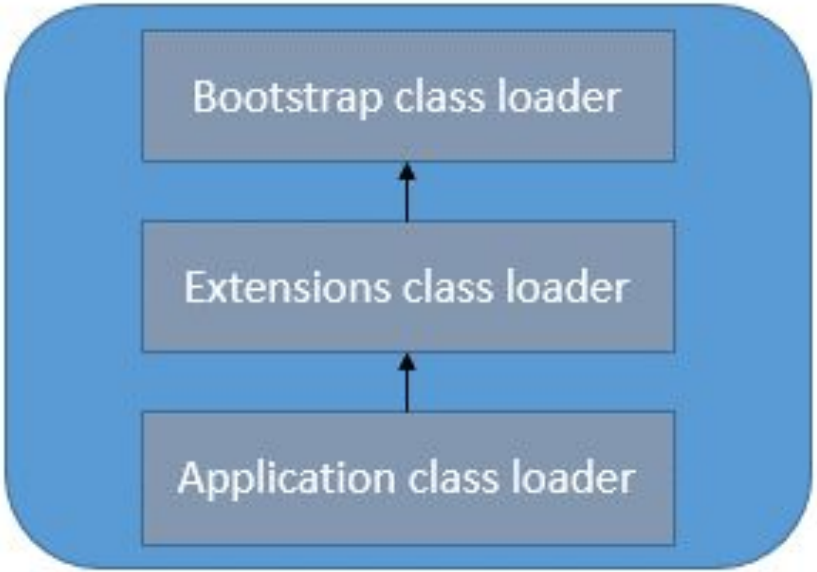
11. CLASS LOADERS

CLASS LOADERS

Theory

- With the help of the class loaders, WebSphere Application Server enables applications to access available class and resources.
- Having known the structure and the hierarchy of class loaders will help you to understand and troubleshoot advances environments against errors such as “ClassNotFoundException”.

CLASS LOADERS

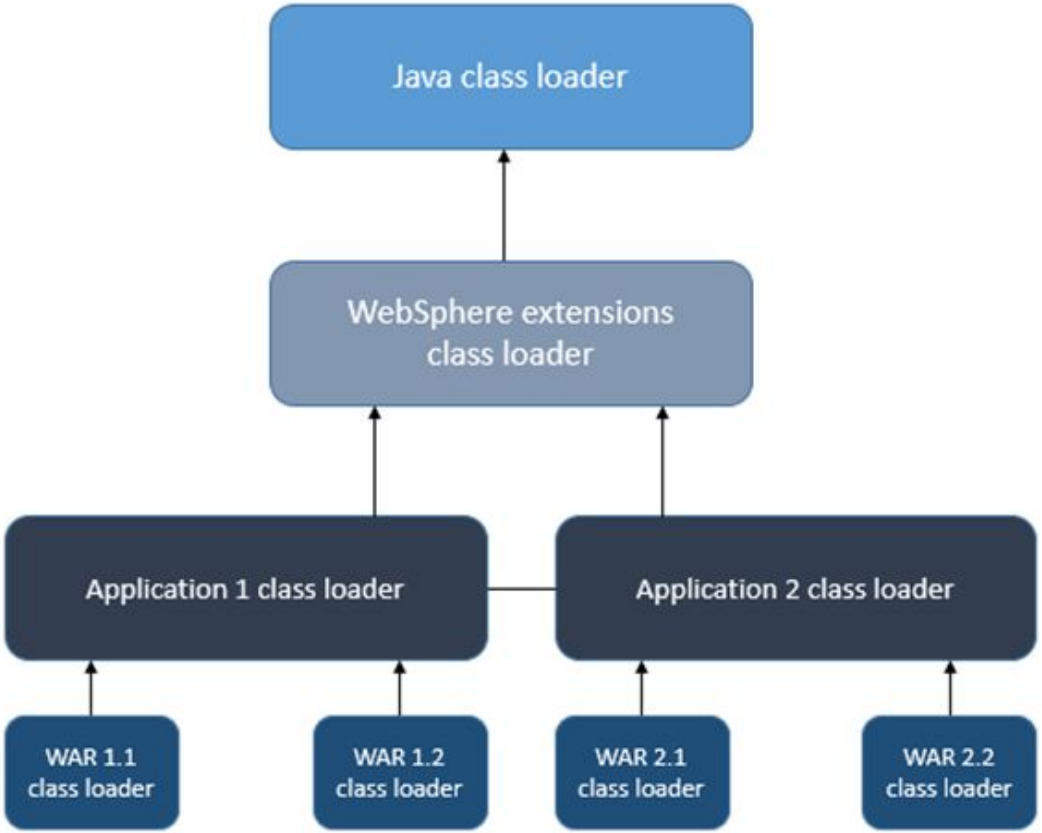


Java class loaders hierarchy

CLASS LOADERS

Application or web module class loaders load elements of enterprise applications running on the server. These application elements can be:

- Web modules
- EJB modules
- Application client modules
- Resource adapter archives
- Utility JAR files



WebSphere class loaders hierarchy

CLASS LOADERS

Application Server class loaders: You can configure 2 types of class loader policies,

- Single: In this case, applications are not isolated and uses a single application loader to load all of the modules, libraries and JAR files that are required by all the applications installed into the JVM.
- Multiple: All applications are isolated and each of them has its own class loader to load EJB modules, libraries and etc.

CLASS LOADERS

Application class loaders: You can set the class loader order and WAR class loader policy within WebSphere Application Server. For class loader order you have 2 options:

- Classes loaded with parent class loader first, sets the loading of classes to its parent class loader before attempting to load the class from its local class path.
- Classes loaded with application class loader first, tells the class loader to start with loading classes from its local class path before asking its parent.

CLASS LOADERS

Web module class loaders: You can set the class loader order that specifies the search order of the class whether to look first in the parent class loader or in the application class loader. You can set the order as following:

- Classes loaded with parent class loader first, the class loader searches first the application class loader
- Classes loaded with application class loader first, the class loader searches first in the WAR class loaders to load a class.

CLASS LOADERS

AIM

- The aim of the lab exercise is to get familiar with class loader settings in different levels. In order to achieve this goal, you need to perform following tasks:

Task 1: Configure class loaders on server level
Task 2: Configure class loaders on application level
Task 3: Configure web module class loader

Lab Exercise 11: CLASS LOADERS

SUMMARY

- WebSphere Application Server enables applications to access available class and resources using class loaders.
- Understand the class loaders will help you to identify and solve the issues in an advanced WebSphere environment.
- WebSphere Application Server enables you to configure class loaders on server level, application level and web module level

12. INTELLIGENT MANAGEMENT

Lab Exercise 12: INTELLIGENT MANAGEMENT

SUMMARY

- Intelligent management is the integration of WebSphere Virtual Enterprise into WebSphere Application Server Network Deployment version 8.5 that provides capabilities autonomic components responding dynamically to the real time conditions of WebSphere environment.

- ODR is an intelligent proxy that performs request prioritization, flow control, and dynamic workload management.

13. JOB MANAGEMENT

JOB MANAGEMENT

Theory

- Batch applications are used to run complex and long tasks that contain typically transactional and multi-step processes.

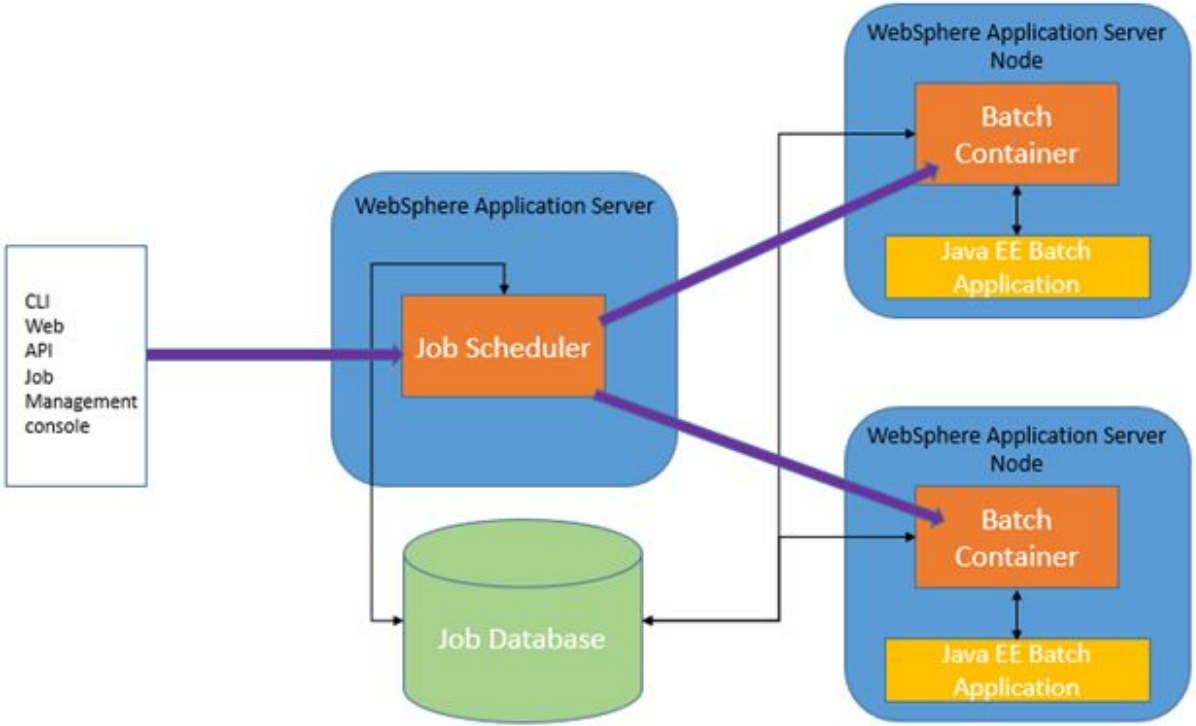
- This type of processes are more demanding for resources thank OLTP systems and they can run for hours.

JOB MANAGEMENT

A batch job contains directives to run one or more batch applications. It performs a specific task in a specific and predefined order. Batch jobs are executed in a batch container. Batch jobs are packaged as EAR and they are deployed to the batch containers to run. A batch job can be

- Transactional batch, contains large number of repeating jobs
- Compute intensive batch, requires high system resources in terms of CPU and memory.

JOB MANAGEMENT



JOB MANAGEMENT

Job scheduler provides all job management functions. Besides basic operations like submitting or restarting a job, they also keep the history of all jobs. In WebSphere Application Server, you can reach the job scheduler in 3 ways:

- Job Management Console, provides a web interface to perform all job management activities.
- Command Line Interface, allows you to submit and control the batch jobs with the help of “lrcmd.(bat|sh)” command.
- APIs, that are available as either web services or EJBs

JOB MANAGEMENT

Job database contains:

- Scheduler tables store job information in a relational database that is supported by WebSphere Application Server.
- In a clustered environment, you have to use a network database that is available to all members of the cluster.
- Container tables store check point information of transactional batch applications in a relational database.

Lab Exercise 13: JOB MANAGEMENT

SUMMARY

- Batch applications are used to run complex and long tasks that contain typically transactional and multi-step processes.
- WebSphere Application Server uses an XML based language, xJCL, to provide consistent architecture that is optimized for Java and long running batch applications.

14. HIGH AVAILABILITY

HIGH AVAILABILITY

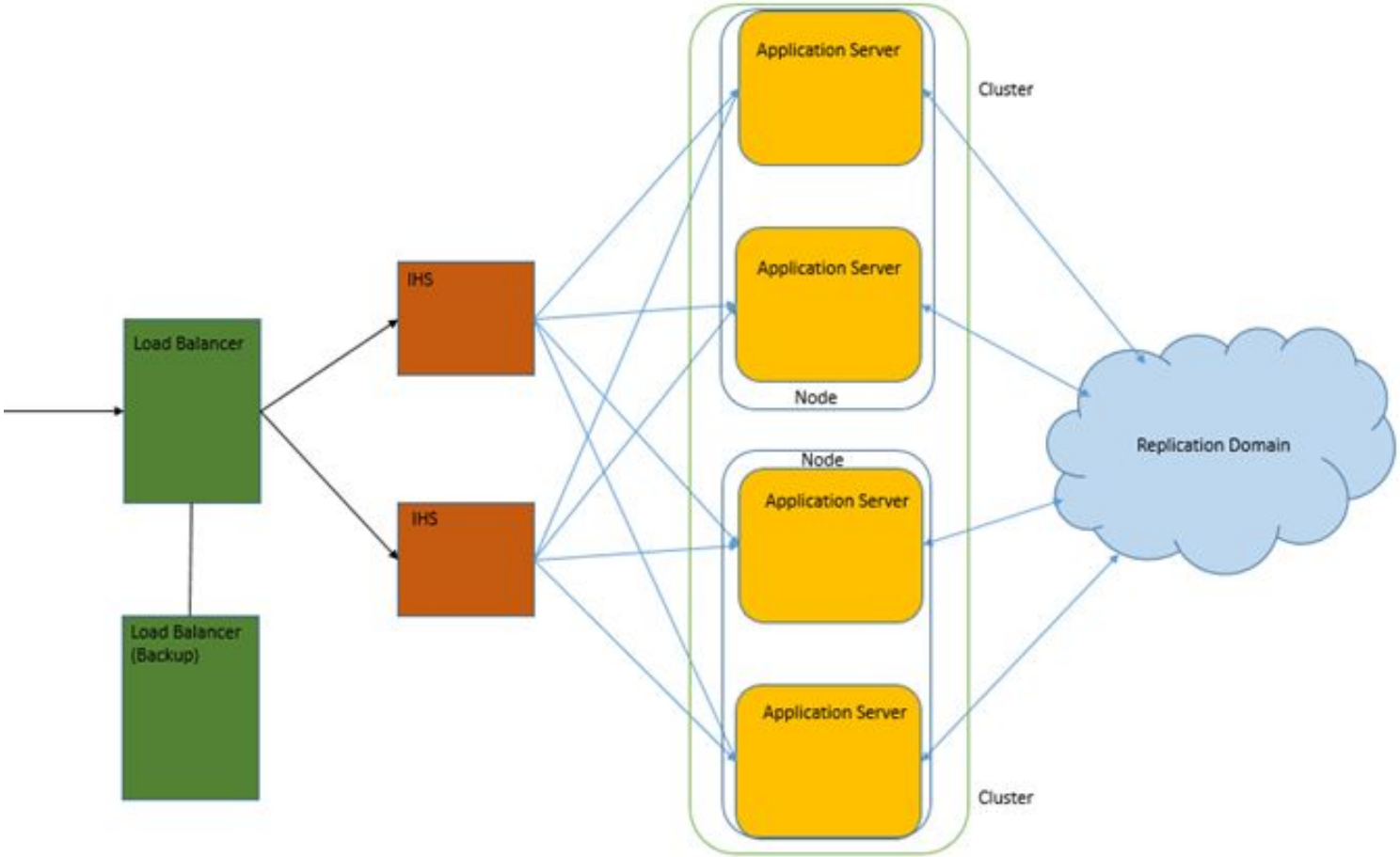
Theory

- High Availability is the ability to tolerate a certain amount of failures without any interruption to the service.
- It is not a consideration point only for WebSphere Application Server and it can be achieved by only having more redundancy in your architecture.
- WebSphere Application Server Network Deployment gives you different options to have a highly available environment for your business critical applications

HIGH AVAILABILITY

In a WebSphere Application Server environment, there are multiple areas that are critical to data availability:

- Databases
- HTTP session state
- EJB session state
- EJB persistence


Application Server
Cluster
Application Server
Node
IHS
Load Balancer
Node
Application Server
Replication Domain
IHS
Application Server
Load Balancer
(Backup)
Cluster

HIGH AVAILABILITY

- A core groups is the high available part of a WebSphere Application Server cell.
- Each HA Manager creates connectivity with all the other HA Manager instances in the same core group.
- HA Manager periodically runs number of tasks in background to provide following services:
- Memory-to-memory replication is provided by Data Replication Service (DRS) which is part of WebSphere Application Server

HIGH AVAILABILITY

Workload management routing has following types:

- Routing of the default messaging bus. (SIB)
- Routing of HTTP requests through WebSphere Application Server proxy server
- Routing of Web Services Addressing requests through WebSphere Application Server proxy server
- Routing of Session Initiation Protocol requests

HIGH AVAILABILITY

AIM

- Create a new core group
- Configure session replication
- Configure cache replication
- Configure high availability application update

Lab Exercise 14: HIGH AVAILABILITY

SUMMARY

- High Availability is the ability to tolerate a certain amount of failures without any interruption to the service.

- WebSphere Application Server Network Deployment gives you different options to have a highly available environment for your business critical applications. manager (HA Manager) provides singleton processes to have process high availability

15. PERFORMANCE TUNING

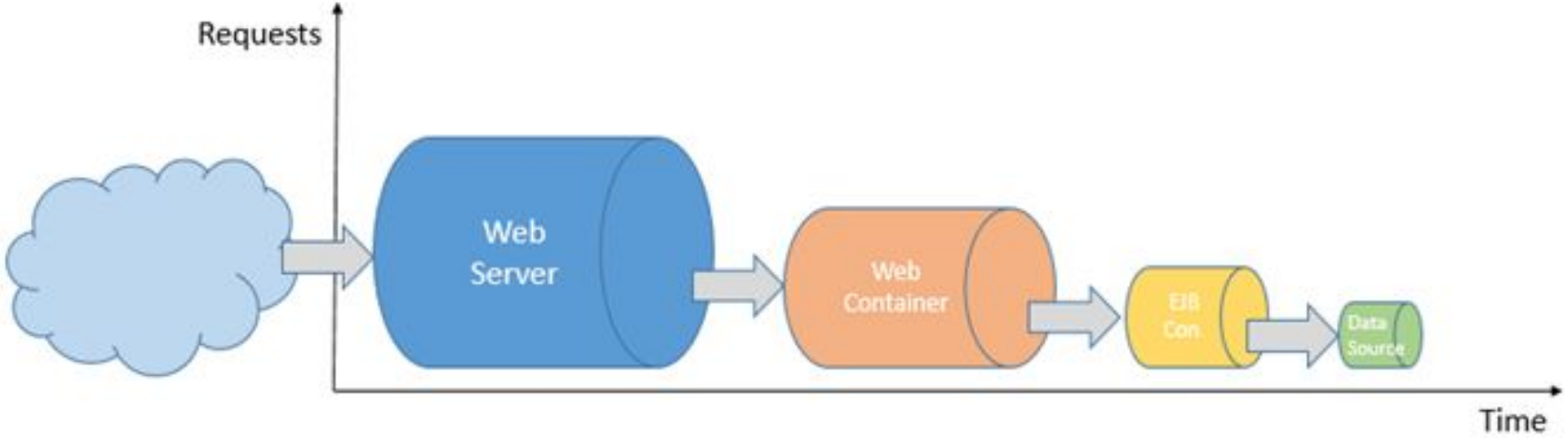
LEARNING VOYAGE

PERFORMANCE TUNING

Theory

- There are multiple areas in WebSphere Application Server to tune for better performance and there is no set of parameters that can fit into all systems.
- You need to consider your application, infrastructure and your workload to plan your environment.
- Furthermore, you need to test and analyze test results to find areas to tune

PERFORMANCE TUNING



PERFORMANCE TUNING

- WebSphere Application Server uses connection pools for data source connections, because the initial connection to a database consumes higher resources.
- It is usually a better idea to set connection pool size lower than the “Max Connections” parameter of the “Web Container”.
- A deadlock occurs when application requires multiple connections per thread and the connection pool size is not large enough.

PERFORMANCE TUNING

- There are also tweaks to tune EJB containers for better performance.
- Inactive pool cleanup interval is one of the options for tuning.
- This setting decides the frequency of cleaning of the EJB beans from the memory.
- You can set this interval to smaller values if it takes longer time to create new EJB bean.
- EJB cache size is another parameter to improve EJB container performance

PERFORMANCE TUNING

IBM Tivoli Performance Viewer is very helpful tool to find out optimum values for performance tuning. With this tool:

- You can view PMI data of local and remote application servers
- You can get configuration advice for better performance
- You can log the performance data for later use such as comparison of the effects of the changed parameters.
- You can view and record server performance logs

PERFORMANCE TUNING

AIM

In this lab exercise, you will be familiar with the common performance tuning parameters. In order to achieve this goal, you need to dive into following tasks:

- Change JDBC connection pool size
- Change JVM heap size
- Change web container thread pool size.

Lab Exercise 15: PERFORMANCE TUNING

SUMMARY

- WebSphere Application Server provides multiple areas for tuning for better performance. Better tuning comes with better knowledge of your application, infrastructure and your workload.

- It is a continuous cycle of testing and analyzing the test results and tuning different parts of the application server environment.

16. MONITORING

Lab Exercise 16: MONITORING

SUMMARY

- IBM WebSphere Application Server provides strong infrastructures to gather monitoring data to be used in application environment management activities including performance tuning.

- PMI collects runtime application server and system data and provides interfaces to allow internal and external applications to reach the collected data.

17. SECURITY

LEARNING VOYAGE

SECURITY

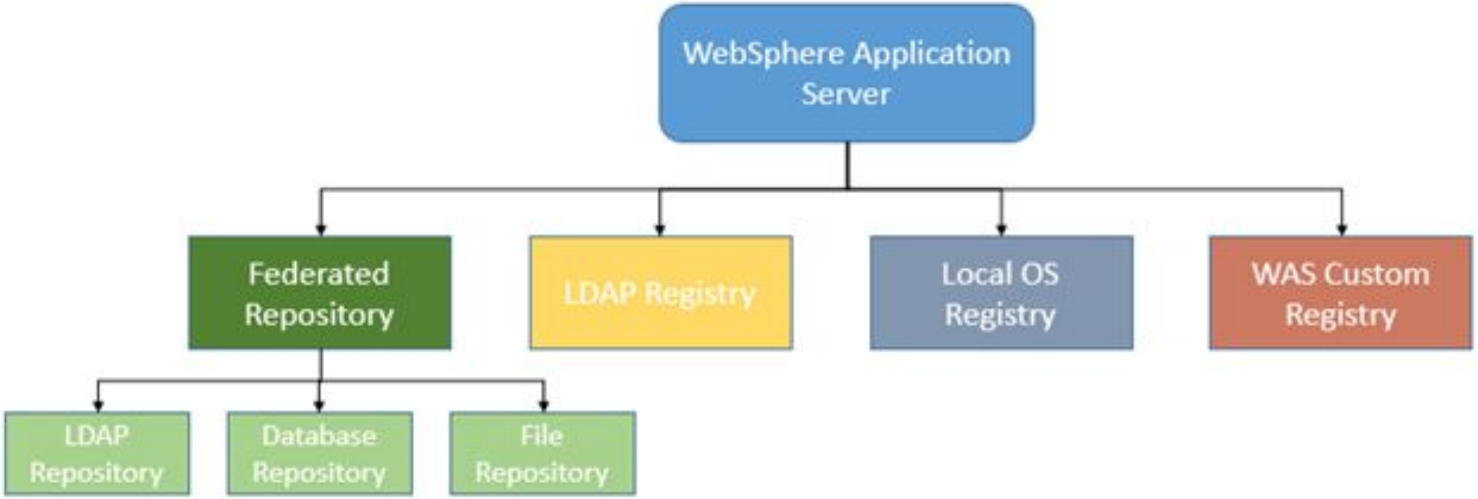
Theory

WebSphere Application Server provides a strong security infrastructure that is usually complex and requires a high level understanding of different areas such as application, network infrastructure and user management. Therefore, WebSphere Application Server has different layers for security management:

- Physical security points out the areas where the environment is physically located.
- Network security contains areas to be protected against network based attacks.

SECURITY

- Authentication confirms a user's identity using user registries. WebSphere Application Server supports different types of user registries:



SECURITY

WebSphere Application Server uses different roles each of which has a set of typical user tasks. This helps to control the WebSphere Application Server environment in terms of security and management. In WebSphere, following administrative roles are defined:

- Monitor gives you the least permissions that allow you to view configuration and current state.
- Configurator, in addition to monitor allows you to change configuration.
- Operator has same permissions as monitor and also able to change the runtime state.

AIM

In this lab exercise, you will be able to configure the most common and important security settings of WebSphere Application Server in an enterprise environment. In order to achieve this goal, you need to complete following tasks:

1. Configure federated user repository
2. Add new users and groups
3. Assign users and groups to roles

Lab Exercise 17: SECURITY

SUMMARY

- WebSphere Application Server provides a strong security infrastructure in different layers that are physical, network, operating system, JVM and so on.
- As part of authentication, WebSphere Application Server supports different types of user registries.
- It is also possible to use more than one user registry by using federated repositories.

18. SSL

SSL

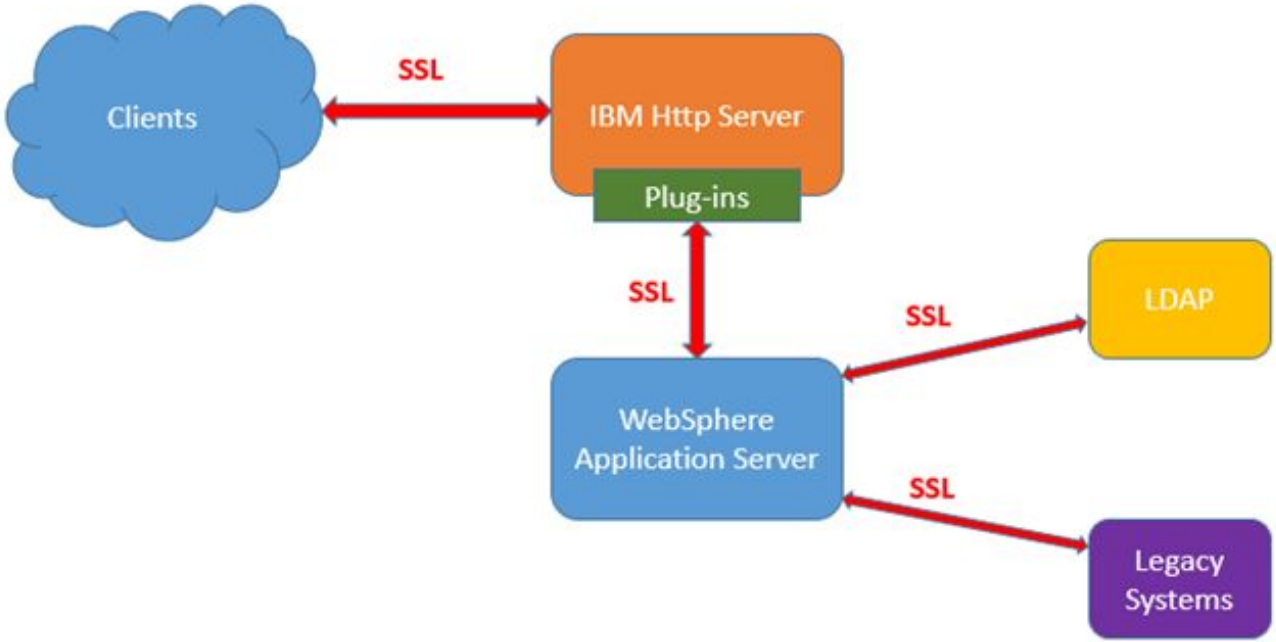
Theory

- WebSphere Application Server communicates with other components of a middleware environment and in order to prevent intrusions, WebSphere uses Java Secure Socket Extension (JSSE) which is an SSL implementation.
- SSL capabilities such as handshaking are delivered by JSSE using X.509 standard public key infrastructure (PKI).

SSL

- SSL is a protocol which is designed to provide secure communication over networks.
- It uses certificates to authenticate the service provider by exchanging keys between server and client.
- WebSphere Application Server store these certificates in password protected files, in keystores.

SSL


Clients
SSL
IBM Http Server
Plug-ins
SSL
SSL
LDAP
WebSphere
Application Server
SSL
Legacy
Systems

SSL

It is also possible to secure the communication between external systems that WebSphere Application Server transmits sensitive information. Following systems worth to be considered to use SSL secured communication:

- Database connection
- LDAP connection
- Message channels (i.e. WebSphere MQ connection)
- Web services connections

SSL

AIM

In this lab exercise, you will perform common SSL operations. When you finish this lab, you will be able to create a keystore and a personal certificate and configure it to use in communication between WebSphere Application Server and the web server. In order to achieve this goal, you need to complete following tasks:

- Create a keystore
- Create a self-signed certificate
- Configure plug-in to use new certificate

Lab Exercise 18: SSL

SUMMARY

- WebSphere uses Java Secure Socket Extension (JSSE) which is an SSL implementation.
- SSL capabilities such as handshaking are delivered by JSSE using X.509 standard public key infrastructure (PKI).
- SSL uses certificates to authenticate the service provider by exchanging keys between server and client.

19. TROUBLESHOOTING

TROUBLESHOOTING

Theory

- Troubleshooting of WebSphere Application Server may be required due to several reasons like performance issues or application unavailability.

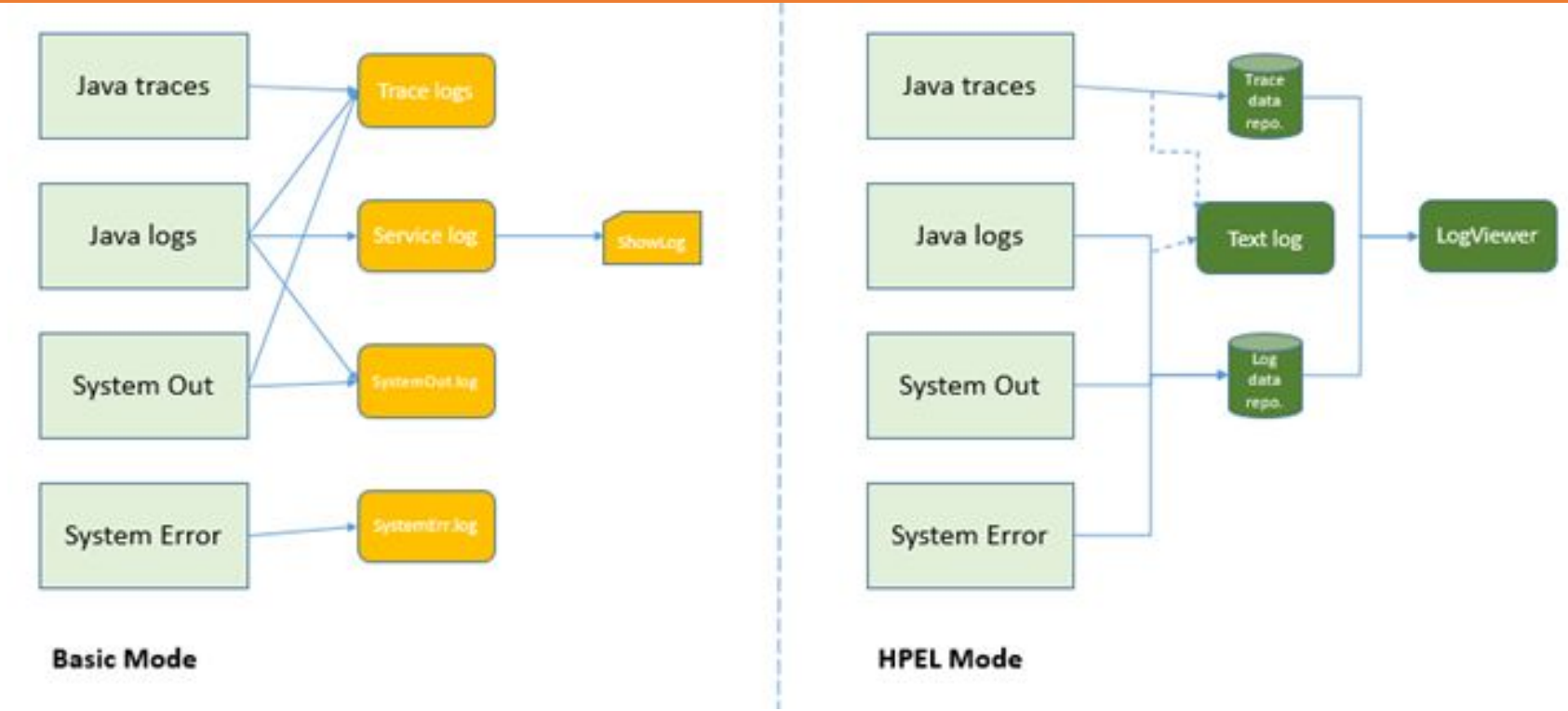
- In order to resolve the problem, we need to collect as much data as possible to identify it.

TROUBLESHOOTING

Here is the list of log files that are used in problem determination:

- **SystemOut.log and SystemErr.log:** In these files standard JVM output and errors are stored.
- **stopServer.log and startServer.log:** These logs store information during stop and start of application servers.
- **trace.log:** This file contains diagnostic trace information if the tracing is enabled.
- **native_stdout.log and native_stderr.log:** These files are used by operating system to log memory exceptions and garbage collection data.

TROUBLESHOOTING



TROUBLESHOOTING

WebSphere Application Server provides 2 different modes of logging and tracing:

- **Basic Mode:** This is the standard and default mode for logging and tracing. This mode is available from older versions of WebSphere Application Server.
- **HPEL Mode:** High Performance Extensible Logging (HPEL) mode is a new framework for logging and tracing which provides a log data repository that contains SystemOut and SystemErr logs, and a trace data repository for trace content

TROUBLESHOOTING

WebSphere Application Server provides features to help you to capture more data for problem determination such as:

- Hang detection policy is enabled by default to report potential hangs. A hung thread can be a result of many different causes and without a proper reporting system will run in a degraded mode.
- Memory leak detection policy can be configured to detect, prevent and take action against potential application memory leaks.

AIM

In this lab exercise, you will configure most used configuration items of WebSphere Application Server for problem determination. To achieve this goal, you will need to complete following tasks:

- Change JVM log settings
- Enable HPEL
- Enable trace
- Generate heap dump

Lab Exercise 19: TROUBLESHOOTING

SUMMARY

- In order to resolve issues like response time, application unavailability and so on, WebSphere Application Server provides several different ways for problem determination.

- There are multiple ways of data collection in WebSphere Application Server including java log files, java traces, thread dumps, heap dumps and system dumps.

20. MIGRATION

MIGRATION

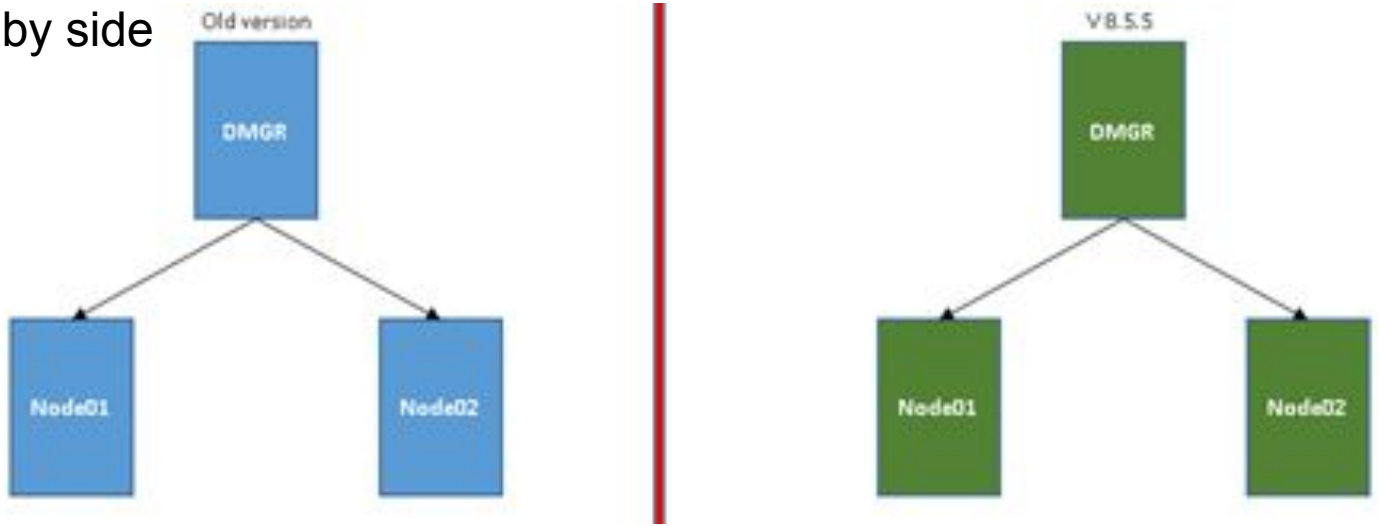
Theory

Migration is moving from one platform to another platform. It can be from another application server such as Apache Tomcat to WebSphere Application server or it can be from an older version of WebSphere to a newer version. For both of the migration types, there are 2 important parts:

1. Application migration
2. Configuration migration

MIGRATION

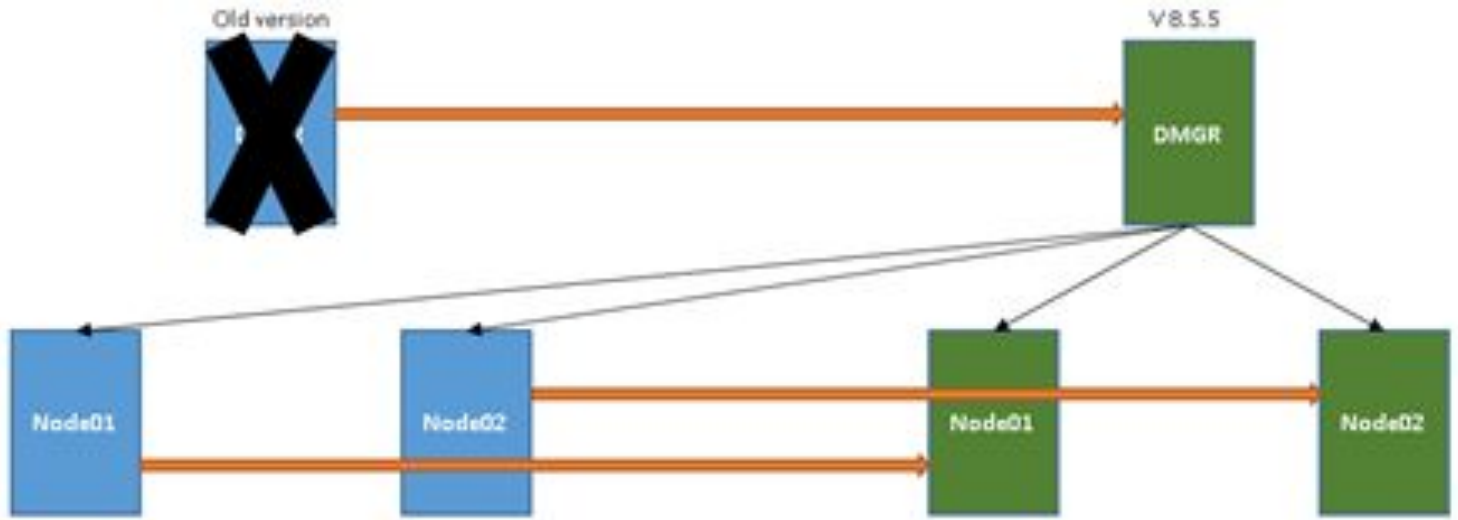
- Side by side



  - You need to create a new cell and populate everything with tools or manually.
  - With this strategy, you don't need any runtime migration tools.

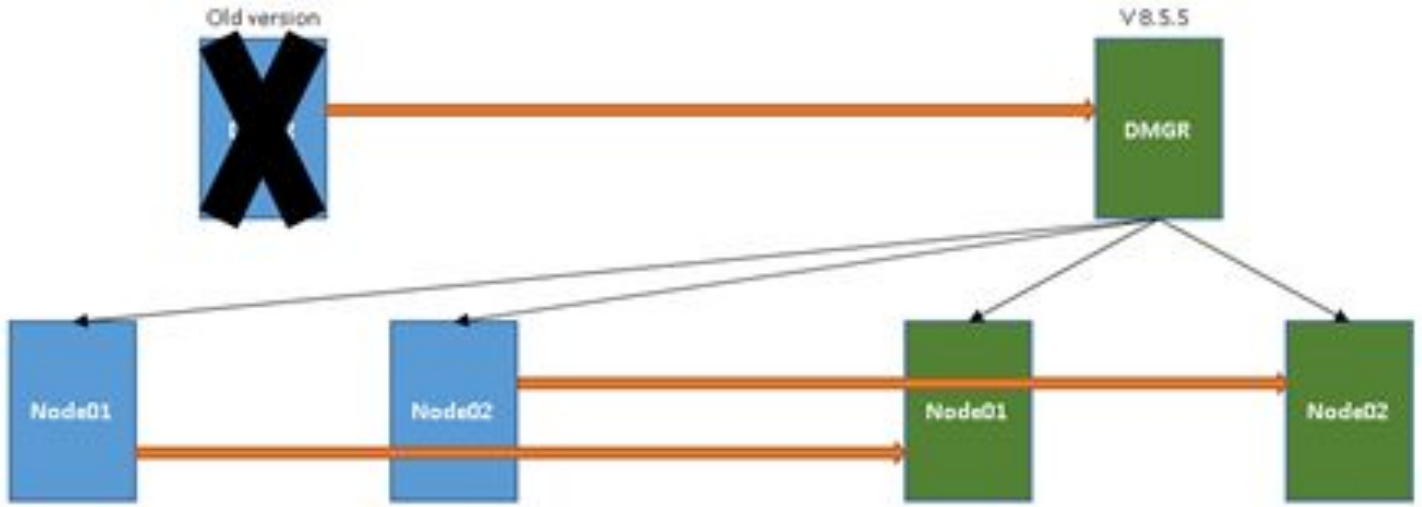
MIGRATION

- In place (replace cell)



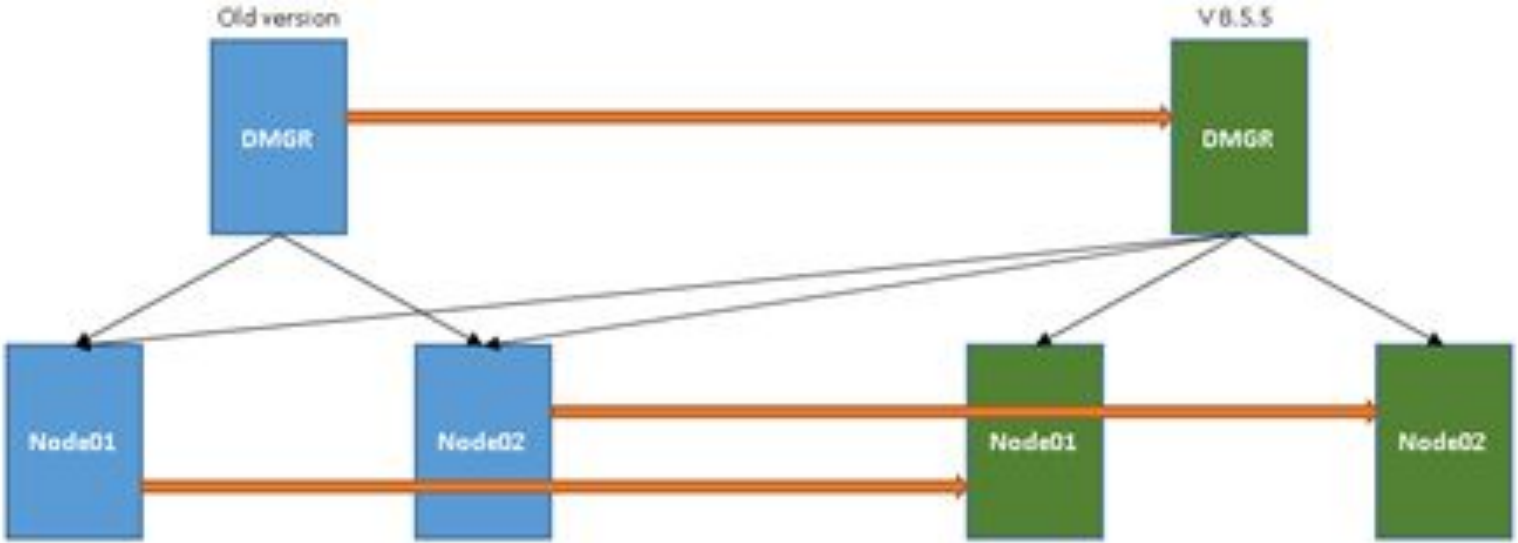
MIGRATION

- In place (replace DMGR)



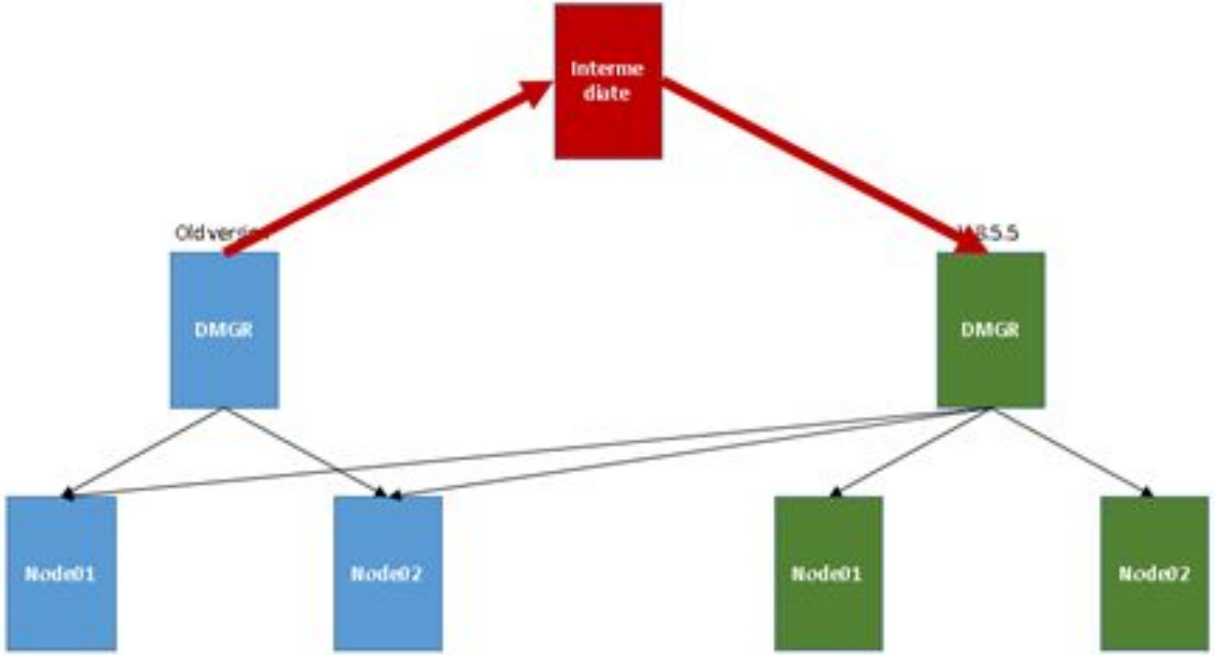
MIGRATION

- In place (co-exist)



MIGRATION

- Side by side (with intermediate profile)



AIM

- In this lesson, you'll be able to install and work with Eclipse IDE for Java EE Developers tool to work on the applications that will be migrated.

- Using this tool, you will import an application, check and mitigate potential problems and then export them.

Lab Exercise 20: MIGRATION

SUMMARY

- Migration, whether from different product or different version of WebSphere Application Server, needs to planned and performed carefully.
- There are different tools provided for migration activities such as Application Migration Toolkit that helps complex migrations.
- There are 2 main points to consider during migration planning: application migration and configuration migration.

THANK YOU ☺