Lecture 29  Application Server

1. Application servers in general

2. JBoss
What is application server

• An application server is a software engine that delivers applications to client computers or devices

• Provide efficient execution of procedures (programs, routines, scripts) for supporting its applied applications

• Contain a comprehensive service layer model

• Act as a set of components accessible to the software developer through an API defined by the platform itself

• For Web applications, these components are usually performed in the same running environment as its web server(s), and one of the main job is to support the construction of dynamic pages

• Application servers target much more than just Web page generation: they implement services like clustering, fail-over, and load-balancing, so developers can focus on implementing the business logic. Ease the application development, since applications need not be built from scratch, but are assembled from building blocks provided by the application server
Examples of application server

1. Oracle: GlassFish

2. Oracle: WebLogic Server

3. IBM: WebSphere Application Server

4. Red Had: JBoss (renamed to WildFly), Enterprise Application Platform (EAP)
Java EE Servers

• A Java EE server is a server application that the implements the Java EE platform APIs and provides the standard Java EE services. Java EE servers are sometimes called application servers, because they allow you to serve application data to clients, much like web servers serve web pages to web browsers.

• Java EE servers host several application component types that correspond to the tiers in a multi-tiered application. The Java EE server provides services to these components in the form of a container.
Java EE Containers

• Java EE containers are the interface between the component and the lower-level functionality provided by the platform to support that component.

• The functionality of the container is defined by the platform, and is different for each component type.

• The server allows the different component types to work together to provide functionality in an enterprise application.
The Web Container

• The web container is the interface between web components and the web server

• A web component can be a servlet, a JSF page, or a JSP page.

• The container manages the component's lifecycle, dispatches requests to application components, and provides interfaces to context data, such as information about the current request.
The Application Client Container

• The application client container is the interface between Java EE application clients, which are special Java SE applications that use Java EE server components, and the Java EE server.

• The application client container runs on the client machine, and is the gateway between the client application and the Java EE server components that the client uses.
The EJB Container

• The EJB container is the interface between enterprise beans

• provide the business logic in a Java EE application, and the Java EE server.

• The EJB container runs on the Java EE server and manages the execution of an application's enterprise beans.
JavaBeans

- **JavaBeans** are reusable software components for Java.

- They are classes written in the Java conforming to a particular convention
  1. The class must have a public default constructor (with no-arguments)
  2. The class properties must be accessible using `get`, `set`
  3. The class should be serializable

- They are used to encapsulate many objects into a single object (the bean), so that they can be passed around as a single bean object instead of as multiple individual objects
JBoss

- JavaBeans Open Source Software Application Server (JBoss AS, or simply JBoss) is an application server that implements the Java Platform, Enterprise Edition (Java EE).

- It’s a Java-based *middleware application server*

- **Middleware** is a piece of software that connects two or more software applications so that they can exchange data.
Why JBoss

• Pain in application development, deployment, execution management in a complex enterprise computing environment.
  – Application server is the solution

• JBoss is the foremost and popular open source solution for creating, distributing, and hosting enterprise Java applications and services

• Alternatives
  – GlassFish, open source, from Sun/Oracle
  – WebLogic, Oracle
  – WebSphere, IBM
Example of application security
Little history of JBOSS

• In 1999, Marc Fleury started a free open source software project named EJB-OSS (stands for Enterprise Java Bean Open Source Software) implementing the EJB API from J2EE
  – Sun Microsystems asked the project to stop using the EJB trademark within its name. EJB-OSS was then renamed to JBOSS, then JBoss later

• JBoss Group, LLC was incorporated in 2001.

• JBoss became a corporation under the name JBoss, Inc. in 2004.

• JBoss Inc was acquired by Red Hat in 2006 at $420 million.
Version history

- JBoss AS 4.0, a Java EE 1.4 application server, features an embedded Apache Tomcat 5.5 servlet container, 2005
- JBoss AS 5.1, released in 2009, operates as a Java EE 5 application server. It is a minor update of the major release JBoss AS 5.0, which was in development for at least three years and was built on top of a new JBoss microcontainer.
- JBoss AS 6.0, an unofficial implementation of Java EE 6, was released on December 28, 2010.
- JBoss AS 7 was released on July 12, 2011.
- JBoss AS 7.1, the current stable version, was released in February 2012. The remaining parts of the EE spec were implemented, and this version was certified for the EE full profile.
- Renamed to WildFly, 2013, WildFly 8.0

- Based on Jboss, Enterprise Application Platform (EPA) was developed, and distributed, enhanced security, performance, certified, for product deployment
Features of JBOSS

1. Aspect-Oriented Programming (AOP) support
2. Clustering
3. Deployment API
4. Distributed caching (using JBoss Cache a standalone product)
5. Distributed deployment (farming)
6. Enterprise Java Beans versions 3 and 2.1
7. Failover (including sessions)
8. Hibernate-integration for persistence programming; JPA (Java Persistence Architecture)
9. JAAS (Java Authentication and Authorization Service) integration
10. JCA (Java Connector Architecture) integration
11. JME (Java Management Extensions) integration
12. JMS (Java Message Service) integration
Features of JBOSS continue

13. JNDI (Java Naming and Directory Interface)
14. JTA (Java Transaction API)
15. JACC (Java Authorization Contract for Containers)-integration
16. JavaMail
17. JavaServer Faces 1.2 (Mojarra)
18. JSP/Servlet (Tomcat)
19. JDBC (Java database connectivity)
20. Load balancing
21. Management API
22. RMI-IIOP (JacORB, contraction of Java and CORBA)
23. SAAJ (SOAP with Attachments API for Java)
24. Support for Java EE-Web Services like JAX-WS
25. Teiid data virtualization system
26. OSGI support
JBoss in Action

• Installation
• Configuration
• Deployment

• Reference:
JBoss in Action
Author: Javid Jamae, Peter Johnson

Demo