Jboss Basic Concepts

Version 1.0

DigiTalk Systems

https://digitalksystems.com/

Reach us at: digitalk.fmw@gmail.com

DigiTalk Channel: https://www.youtube.com/channel/UCCGTnI9vvF_ETMhGUXGdFWw

WebLogic Server Architecture: https://youtu.be/gNqeIfLjUqw

Table of Contents

DigiTalk Udemy Courses and Coupon Code

Docker Administration

https://www.udemy.com/course/mastering-docker-administration-from-fundamentals-to-expert/?referralCode=12387A8B195F7B90381D

Coupon Code: 4400C856A93FC2F851F3

Kubernetes Administration

https://www.udemy.com/course/mastering-kubernetes-administration-from-intro-to-advanced/?referralCode=7E0250BEC21F46192834

Coupon Code: 44C67B0AA69362E8BDE1

WebSphere Administration

https://www.udemy.com/course/mastering-ibm-websphere-9x-administration/?referralCode=F372B3EA78D7BEA92416

Coupon Code: 9C42313FDEFC9DCE4A21

Mastering SSL (TLS), Keys and Certificates Management

https://www.udemy.com/course/mastering-ssl-tls-keys-and-certificates-management/?referralCode=DB9F1429CB01921969F6

Coupon Code: 5B702665FBF721B6F315

SOA Suite Administration

https://www.udemy.com/course/mastering-oracle-soa-suite-12c-administration/?referralCode=2BCF3895067AA7BC22E0

Coupon Code: BF1D2646A214C279C7B1

JBoss 8 Administration

https://www.udemy.com/course/mastering-jboss-eap-8-administration-from-intro-to-advanced/?referralCode=D095CE8B1EE9D065C822

Coupon Code: 87936B19622155F1DC68

OHS Administration

https://www.udemy.com/course/mastering-oracle-ohs-http-12c-web-server-administration/?referralCode=165F6A3A8B662CEA1489

Coupon Code: 33542ABA75949ED745F8

WebLogic Server Administration

https://www.udemy.com/course/oracle-weblogic-server-12c-and-14c-administration/?referralCode=340FBB94DBFEE1A8CC09

Coupon Code: BB5CB01A2CA9498DBFED

Mastering Oracle Internet Direct (OID) Administration

https://www.udemy.com/course/mastering-oracle-internet-direct-oid-administration/?referralCode=906E9AC0059A29A3CD0D

Coupon Code: 8D43AFACC93B009932E1

For any query or concerns please write us on digitalk.fmw@gmail.com

Introduction to JBoss and Controllers

JBoss Application Server (now known as WildFly) is a popular open-source Java EE application server used for deploying and managing enterprise applications. In JBoss, the concepts of **Domain Controller**, **Host Controller**, and **Process Controller** are critical components in its domain management architecture, particularly when running JBoss in **domain mode** (as opposed to standalone mode). These components work together to manage multiple server instances, configurations, and deployments across a distributed environment.

- **Domain Mode**: Allows centralized management of multiple JBoss server instances (grouped into server groups) across one or more hosts, enabling scalability and coordinated administration.
- **Standalone Mode**: A single server instance running independently, not relevant to these controllers.

The **Domain Controller**, **Host Controller**, and **Process Controller** are hierarchical components in domain mode that manage the configuration, coordination, and execution of JBoss server instances. Below, we'll explore each component, provide examples, and compare them.

1. Domain Controller

Definition

The **Domain Controller** is the central management point in a JBoss domain. It maintains the domain configuration, including server groups, profiles, and deployments, and propagates this configuration to all Host Controllers in the domain. There is only one Domain Controller per domain, acting as the single source of truth for the domain's configuration.

Responsibilities

- Stores and manages the domain-wide configuration (e.g., domain.xml).
- Coordinates with Host Controllers to distribute configuration and deployment updates.
- Handles management operations (e.g., deploying applications, updating configurations) initiated by administrators via CLI, web console, or management API.
- Ensures consistency across all servers in the domain.

Example

Suppose you have a JBoss domain with two hosts, each running multiple server instances. The Domain Controller runs on one of these hosts (e.g., Host A) and is configured in domain.xml:

```
<domain-controller>
<local/>
</domain-controller>
```

- The Domain Controller on Host A defines two server groups: main-server-group (using profile full) and web-server-group (using profile web).
- An administrator deploys a web application (app.war) to main-server-group via the Domain Controller using the JBoss CLI:

```
deploy app.war --server-groups=main-server-group
```

- The Domain Controller pushes this deployment to all Host Controllers, which then ensure the application is deployed on the relevant server instances.

Key Points

- Runs on one host, designated as the primary management point.
- Uses domain.xml for domain-wide configuration and host.xml for its own host-specific settings.
- Can be configured as a local Domain Controller (<local/>) or remote (<remote host="..."/>).

2. Host Controller

Definition

The **Host Controller** manages one or more JBoss server instances on a single host (physical or virtual machine). It communicates with the Domain Controller to receive domain-wide configurations and deployments, then applies them to its local server instances. Each host in a domain has its own Host Controller.

Responsibilities

- Manages server instances (processes) on its host.
- Applies domain configurations (from the Domain Controller) to its servers.
- Reports server status and metrics back to the Domain Controller.
- Starts, stops, and monitors server instances based on domain instructions.

Example

Consider a host (Host B) with two server instances, server-one and server-two, part of the main-server-group. The Host Controller on Host B is configured in host.xml:

```
<servers>
    <server name="server-one" group="main-server-group"/>
    <server name="server-two" group="main-server-group"/>
</servers>
```

- The Host Controller on Host B connects to the Domain Controller on Host A.
- When the Domain Controller deploys app.war to main-server-group, the Host Controller on Host B receives the deployment and starts it on server-one and server-two.

Key Points

- Each host has one Host Controller.
- Uses host.xml to define host-specific settings (e.g., server instances, network interfaces).
- Acts as an intermediary between the Domain Controller and local server instances.

3. Process Controller

Definition

The **Process Controller** is a lightweight process responsible for managing the lifecycle of JBoss server instances (Java Virtual Machine processes) on a specific host. It is started when the Host Controller is launched and ensures that server instances are started, stopped, or restarted as needed.

Responsibilities

- Launches and terminates server instance processes.
- Monitors server instances for crashes or failures and can restart them if configured to do so.
- Acts as a low-level process manager for the Host Controller.

Example

When the Host Controller on Host B starts server-one and server-two, the Process Controller spawns the actual Java processes for these servers. If server-one crashes due to an out-of-memory error, the Process Controller detects the failure and may restart it based on the configuration in host.xml:

```
<server name="server-one" group="main-server-group" auto-start="true">
    <jvm name="default"/>
</server>
```

- The Process Controller ensures that the JVM for server-one is running and restarts it if it fails.

Key Points

- Runs as part of the Host Controller process but is responsible for managing server instance JVMs.
- Does not handle configuration or deployment; it focuses on process management.
- Invisible to administrators, as it operates behind the scenes.

Comparison of Domain Controller, Host Controller, and Process Controller

Aspect	Domain Controller	Host Controller	Process Controller
Role	Central management point for the entire domain.	Manages server instances on a single host.	Manages the lifecycle of server instance processes.
Scope	Domain-wide (all hosts and servers).	Host-specific (servers on one host).	Server instance-specific (JVM processes).
Configuration File	domain.xml, host.xml (for its host).	host.xml (for its host).	None (controlled by Host Controller).
Responsibilities	Manages domain configuration, deployments, and coordination with Host Controllers.	Applies domain config to local servers, starts/stops servers, reports status.	Spawns, monitors, and restarts server JVMs.
Number per Domain	One (single point of control).	One per host in the domain.	One per host (tied to Host Controller).
Communication	Communicates with Host Controllers.	Communicates with Domain Controller and Process Controller.	Communicates with Host Controller.
Example Action	Deploys app.war to a server group.	Ensures app.war is deployed on its servers.	Starts the JVM for a server instance.
Visibility	Visible to administrators (CLI, web console).	Visible to administrators (host-specific config).	Invisible to administrators (internal process).
Dependency	Depends on its Host Controller (if local).	Depends on Domain Controller for config.	Depends on Host Controller for instructions.

Practical Example of Interaction

Imagine a JBoss domain with two hosts:

- **Host A**: Runs the Domain Controller and a Host Controller managing two servers (server-a1, server-a2).
- **Host B**: Runs a Host Controller managing two servers (server-b1, server-b2).

1. **Domain Controller (Host A)**:
    - Configured in domain.xml with a server group main-server-group.
    - An administrator deploys app.war to main-server-group using the CLI.
    - The Domain Controller sends the deployment to Host Controllers on Host A and Host B.
2. **Host Controller (Host A and Host B)**:
    - Host A's Host Controller deploys app.war to server-a1 and server-a2.
    - Host B's Host Controller deploys app.war to server-b1 and server-b2.
    - Each Host Controller instructs its Process Controller to start the server processes.
3. **Process Controller (Host A and Host B)**:
    - On Host A, the Process Controller starts JVMs for server-a1 and server-a2.
    - On Host B, the Process Controller starts JVMs for server-b1 and server-b2.
    - If server-b1 crashes, Host B's Process Controller restarts it.

Summary

- The **Domain Controller** is the brain of the domain, managing configurations and deployments across all hosts.
- The **Host Controller** is the local manager, applying domain instructions to servers on its host.
- The **Process Controller** is the executor, handling the low-level process management of server instances.

Together, these components enable JBoss to manage complex, distributed environments efficiently, ensuring centralized control, scalability, and fault tolerance.

Disclaimer and Consent

This document is being provided by DigiTalk as part of its effort to assist users in understanding and working with IBM WebSphere. The Company wishes to emphasize that this document is not affiliated with IBM in any way, and the content contained herein is based solely on publicly available product documentation provided by IBM.

While every effort has been made to ensure the accuracy and reliability of the information presented in this document, there is a possibility of typographical errors or inaccuracies. DigiTalk does not guarantee the correctness or completeness of the content provided in this document.

Users of this document are encouraged to cross-reference the information presented here with IBM's official documentation available on their website or other authoritative sources. Any discrepancies or inaccuracies found in this document should be reported to us at digitalk.fmw@gmail.com.

We would like to make it clear that the code snippets and examples used in this document are sourced from publicly available IBM demo applications, provided by IBM for educational and learning purposes. These materials are subject to IBM's copyright and licensing terms.

We would like to emphasize that our charges are solely for the efforts and resources invested in preparing the lab documents and explanations to enhance the learning experience. DigiTalk is not charging for the usage of IBM's copyrighted material.

By using or accessing these documents, you acknowledge and agree that the IBM code snippets and examples are the intellectual property of IBM. We do not claim ownership of the code, and any trademarks or copyrights associated with IBM's remain the property of IBM.

By using this document, you acknowledge and consent to the following:

This document is not officially endorsed or verified by IBM.

The Company makes no claims or guarantees about the accuracy or suitability of the information contained in this document.

Users are responsible for verifying and validating any information presented here for their specific use case.

DigiTalk disclaims any liability for any errors, omissions, or damages that may result from the use of this document.

If you discover any inaccuracies or errors in this document, please report them to digitalk.fmw@gmail.com, and the Company will endeavor to correct them as necessary.

This consent statement is provided to ensure transparency and understanding of the limitations of the information contained in this document. By using this document, you agree to abide by the terms and conditions outlined herein.