JUDCon

JBoss Users & Developers Conference

London:2011

Configuration and Management

with AS7

Emanuel Muckenhuber

Software Engineer, Red Hat Inc.

Agenda

- Core management concepts
- Multi server management
- OperationHandlers
- Demo

AS7 Key Features

- Fast and lightweight
- Modularity
- Unified, user configuration
- Multi-node (domain) management
- Multiple consistent management interfaces

AS7 Architecture

- Redesigned from scratch with focus on
  - **Performance**
  - **Modularity**
  - **Configuration**
  - **Management**

User oriented configuration

- **Separation** of configuration, binaries and wiring
- **Centralized** in a few files
- **Unification** of common configuration
  - socket-bindings, paths, system-properties

Simplified configuration

```
<bean name="TransactionManager"
class="com.arjuna.ats.jbossatx.jta.TransactionManagerService">
        <annotation>@org.jboss.aop.microcontainer.aspects.jmx.JMX
(name="jboss:service=TransactionManager",
exposedInterface=com.arjuna.ats.jbossatx.jta.TransactionManagerServiceMBean.class,
registerDirectly=true)</annotation>
        <annotation>@org.jboss.managed.api.annotation.ManagementObject
(name="TransactionManager",componentType=@org.jboss.managed.api.annotation.ManagementCompone
nt(type = "MCBean", subtype =
"JTA"),targetInterface=com.arjuna.ats.jbossatx.jta.TransactionManagerServiceMBean.class)
        </annotation>

        <property name="transactionTimeout">300</property>
        <property name="objectStoreDir">${jboss.server.data.dir}/tx-object-store</property>
```

```
<subsystem xmlns="urn:jboss:domain:transactions:1.0">
            <recovery-environment socket-binding="txn-recovery-environment"
                                  status-socket-binding="txn-status-manager"/>
            <core-environment socket-binding="txn-socket-process-id"/>
</subsystem>
```

Standalone.xml (I)

```
<server xmlns="urn:jboss:domain:1.1">
    ....
    <interfaces>
        <interface name="public">
            <inet-address value="192.168.0.1" />
        </interface>
    </interfaces>
    <socket-binding-group default-interface="public">
        <socket-binding name="http" port="8080" />
    </socket-binding-group>
</server>
```

Standalone.xml (II)

```
<server xmlns="urn:jboss:domain:1.1">
   <extensions />
   <profile >
      <subsystem ... />
   </profile>

   <interfaces />
   <paths />
   <socket-binding-group />
   <system-properties />
   <deployments />
</server>
```

Extensions

- **Load additional capabilities** into the AS
  - Separate from deployment
  - Reference a module
  - Declared as part of the configuration

```
<extension module="org.jboss.as.transactions"/>
```

Subsystems

- **THE** extensibility point for AS
  - Registers all required services and deployers
  - Defines management model and operations
  - Provides a specific xml configuration schema

Subsystem (II)

```
<extension module="org.jboss.as.web" />
...
<subsystem xmlns="urn:jboss:domain:web:1.0">
    <connector name="http" protocol="HTTP/1.1"
                           socket-binding="http" />
    <virtual-server name="example" >
        <alias name="example.com" />
    </virtual-server>
</subsystem>
```

Profile

- Collection of functionality / capabilities a server or group of servers runs
  - Single profile for standalone
  - Multiple profiles in domain
    - web, messaging, datagrid, EE

Management API

- Based on a **detyped model**
  - Everything is described as a simple type

```
"connector" => {
   "http" => {
      "enabled" => true,
      "protocol" => "HTTP/1.1",
      "scheme" => "http",
      "secure" => false,
      "socket-binding" => "http" } }
```

Management API (II)

- Stable and complete API
  - Configuration and runtime view
- Interaction with the model only through **management operations**
- Supports multiple clients
  - CLI, HTTP, Java API, Console

Manageable “Resources”

- Model stored as a tree of resources
- Addressable entities
  - Operations
  - Attributes
  - Description

Management Operations

- Described using the detyped API
- Interactions through operations only

```
operation {
    "operation" => "operation-name",
    "address" => [],
    "request-parameters" => ...
}
```

Common operations

- `add()`
- `remove()`

- `read-attribute()`
- `read-children-names()`
- `read-operation-names()`
- `read-operation-description()`
- `read-resource()`
- `read-resource-description()`

- `write-attribute()`

Two operational “modes”

- **Standalone**
  - “traditional” single JVM server
  - Management facilities IN-VM
  - Configuration via standalone.xml

Managed Domain Mode

- Manage **heterogeneous** multi-server environments
- from a single point

- Multiple servers **managed as a group**

Managed Domain Terms

- (Master) DomainController
- HostController
- Server
- ServerGroup
- Cluster

Domain Configuration

- **domain.xml**
  - Primary domain configuration
  - Represents the current state
- **host.xml**
  - Environment / Host specific configuration

domain.xml

```
<domain xmlns="urn:jboss:domain:1.1">
   <profiles>
     <profile name=”web” > ... </profile>
     <profile name=”messaging” > ... </profile>
   </profiles>
   <paths />
   <interfaces />
   <socket-binding-groups />
   <server-groups />
   <deployment />
</domain>
```

server-groups

```
<server-groups>
   <server-group name="web-server-group"
                 profile="web">
   <jvm name="default">
       <heap size="64m" max-size="512m" />
   </jvm>
   <socket-binding-group ref="web-sockets" />
   <deployments />
</server-group>
```

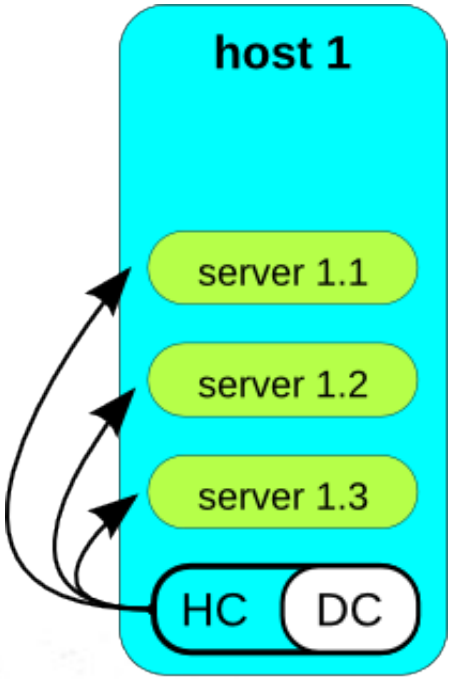
host.xml

```
<host xmlns="urn:jboss:domain:1.1">
   <domain-controller />
   <jvms />
   <interfaces />
   <paths />
   <servers>
      <server name=”web-server” group=”web” />
      <server name=”messaging” group=”messaging” />
   </servers>
</host>
```

server

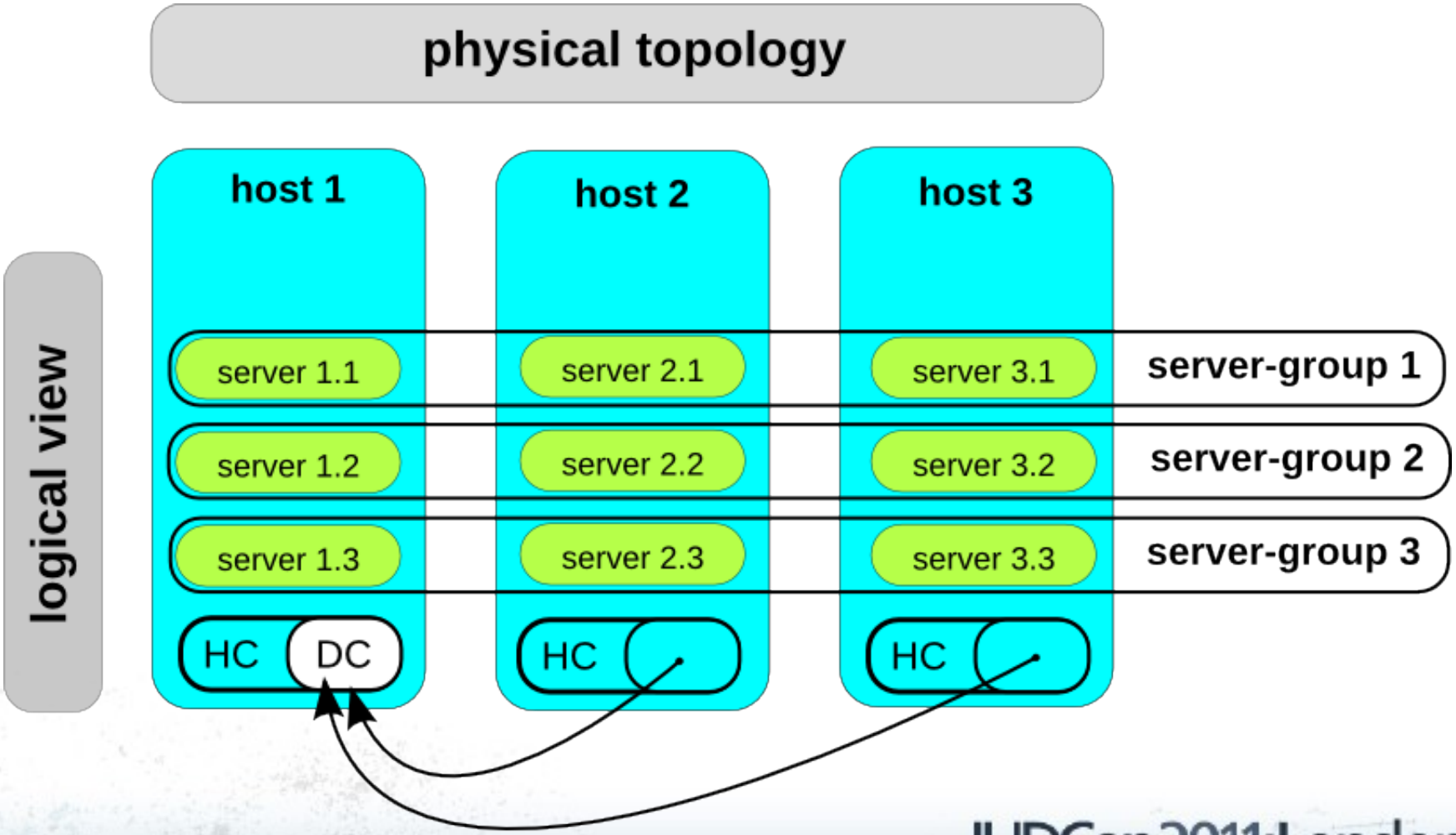
```
<server name="web-one" group="web" auto-start="true">
   <socket-binding-group ref="web-sockets"
           port-offset="150"/>
   <jvm name="default">
      <heap size="64m" max-size="256m"/>
   </jvm>
   <system-properties />
</server>
```

Managed Domain (I)



- Servers defined **per host**
- Managed by the HostController
- **Retrieves configuration** from the domain

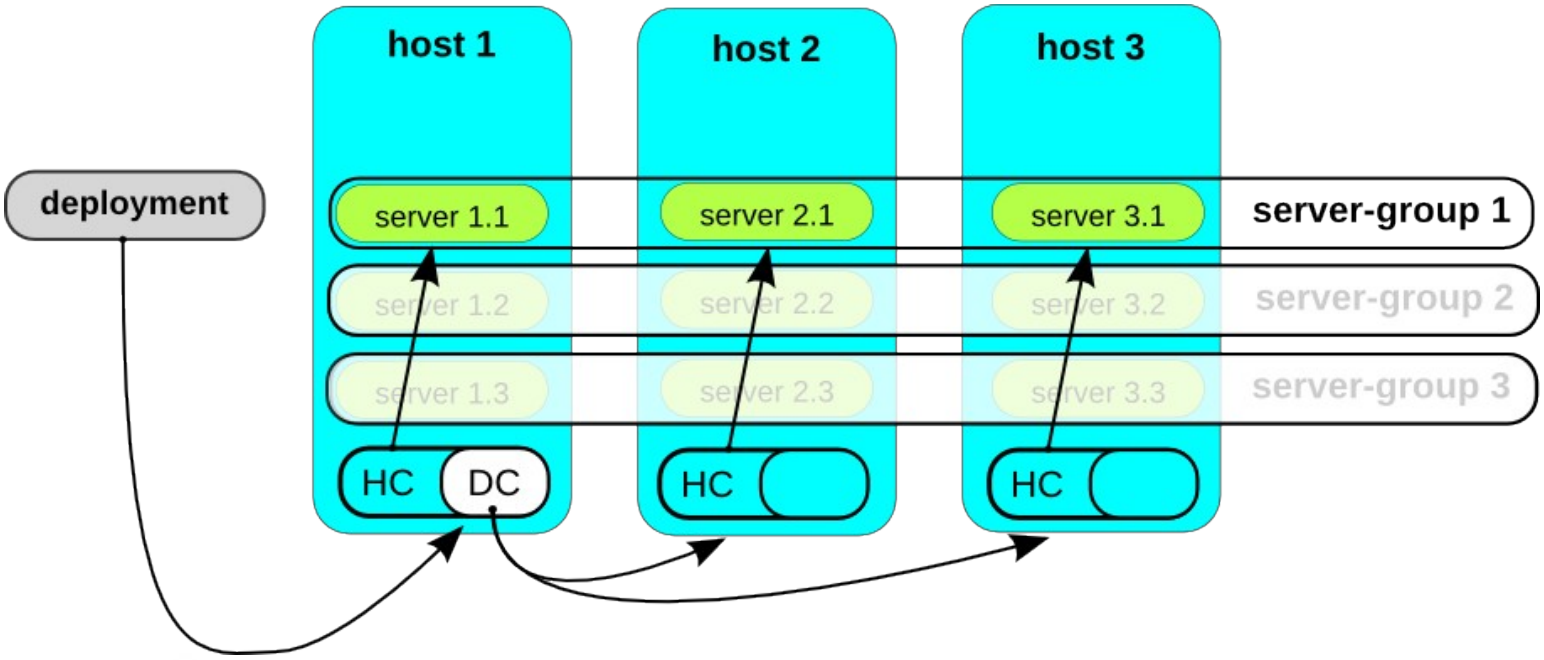
Managed Domain (II)



Deployment plans

- Defines the execution of deployment actions
- Provides different strategies
  - Rolling deployment
    - Within a server-group
    - Across multiple server-groups
  - Rollback on failure

Deployment plans



Operation Handlers

- Only way to interact with the model
- Handle model and runtime updates
- Executed in multiple “steps”
  - model, runtime, verify

OperationStepHandler

```
public class ReadAttributeHandler implements OperationStepHandler {

    @Override
    public void execute(final OperationContext context, final ModelNode operation)
            throws OperationFailedException {

        final Resource resource = context.readResource(PathAddress.EMPTY_ADDRESS);
        final ModelNode subModel = resource.getModel();

        final String attributeName = operation.get("name").asString();
        final ModelNode attribute = subModel.get(attributeName);

        context.getResult().set(attribute);
        context.completeStep();
    }
}
```

Demos

Resources

- Documentation
  - http://www.jboss.org/as7
  - https://docs.jboss.org/author/display/AS7/Documentation
- Download
  - http://www.jboss.org/jbossas/downloads
- Issue tracker
  - https://issues.jboss.org/browse/AS7
- User forums
  - http://community.jboss.org/community/jbossas/as7_users

Q&A