JBoss **Application Server 7**

Agenda

- Operations, Administration & Management (OA&M):
  - Key Goals for JBoss Application Server 7 / JBoss Enterprise Application Platform 6
- Domain Mode and Standalone Mode
- Key Management Model Concepts
- Demo
- Q&A

Key OA&M Goal – Simplified Configuration

- End user configuration centralized in a few files
  - No longer scattered all over the distribution
- Configuration files focused on end user configuration
  - No internal service wiring details
- Config changes made via management tools persisted back to the config file

Key OA&M Goal – Robust Management API

- Complete: expose everything in the config schema
  - Plus metrics, runtime operations
- Stable: no incompatible changes across EAP 6.x series
- Secure remote access via:
  - Native Java interface
  - HTTP + JSON
  - CLI

Key OA&M Goal – Multi-Server Management

- Multi-server management as a core part of AS 7 / EAP 6 itself
- Manage multiple servers from a single control point
  - Start/quiesce/stop servers
  - Rolling deployment to a set of servers
  - Roll a config change out to a set of servers
  - Roll back changes

Choices for How to Manage Your AS Instances

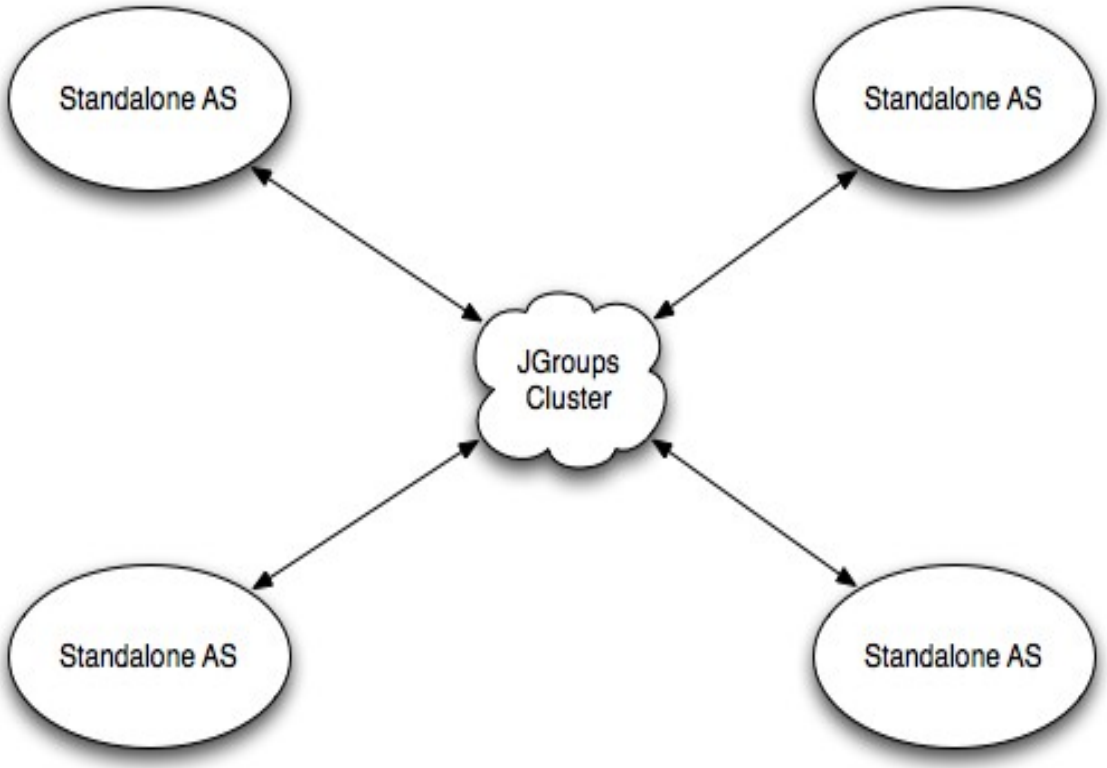
- Two different operational modes
- Basically, do you want to take advantage of our multi-server management features?
  - Yes: run in *Domain Mode*
    - `[brian bin]$ ./domain.sh`
  - No: run in *Standalone Mode*
    - `[brian bin]$ ./standalone.sh`
- Either way, you still get simplified configuration and a robust management API

Standalone Mode

- Each server is independently managed, a la AS 3/4/5/6 and EAP 4/5
- User is responsible for coordinating changes across servers
- Good for many development use cases
- An option for enterprises with their own preferred tooling for multi-server management
- Single configuration file:
  - `standalone/configuration/standalone.xml`

Standalone Mode Allows HA Clusters

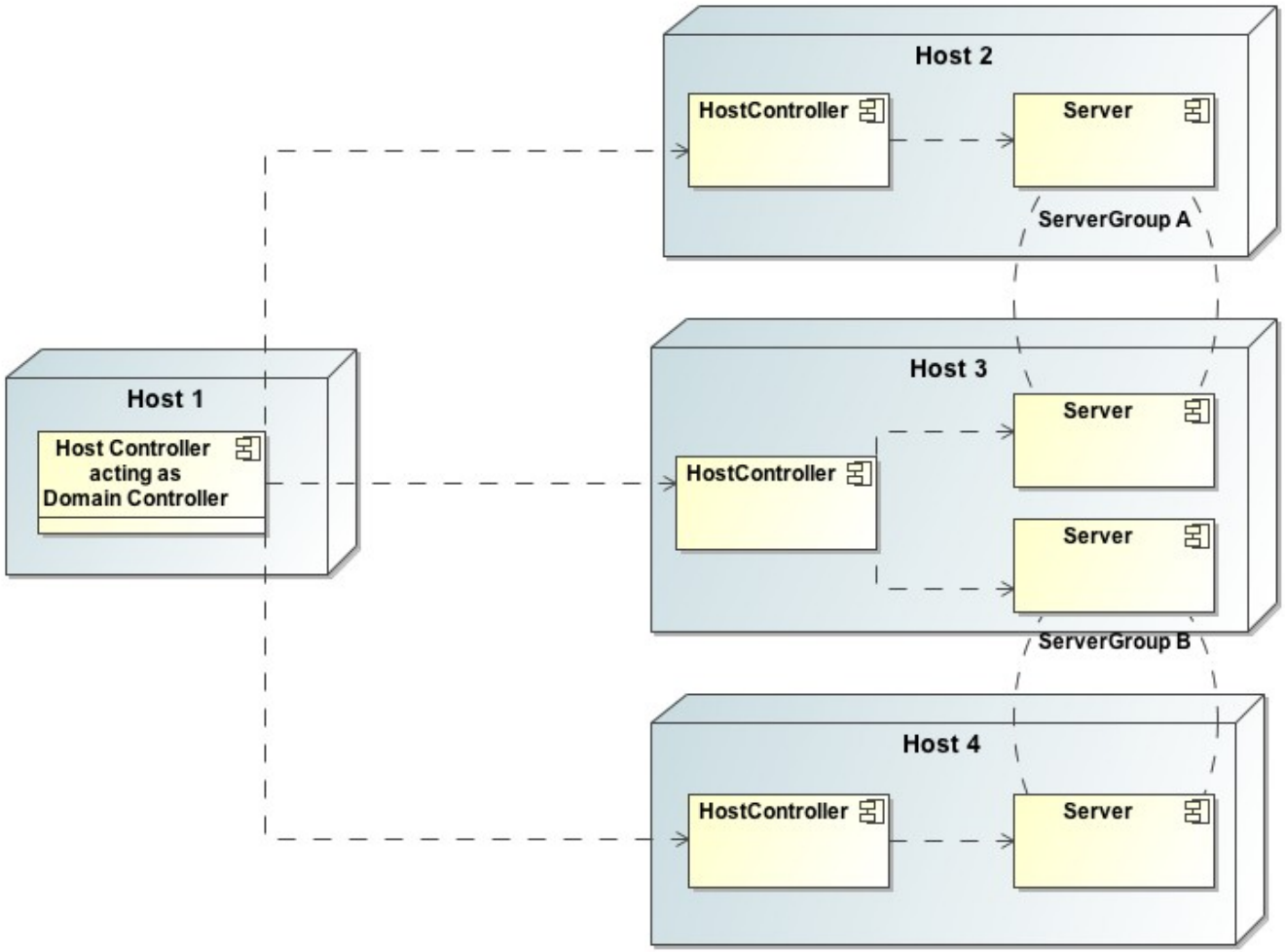
- Standalone mode is about *management*, not how managed services operate



Domain Mode

- Domain: a set of servers with a common management policy
  - Policy is defined in `domain.xml` config file
  - Servers can be heterogeneous in a domain
- We provide processes that coordinate management across the domain
  - Domain Controller
  - Host Controller

Domain Mode Architectural Elements



Key Management Model Concepts

- Subsystem: a particular set of capabilities that extend the application server core
  - Webserver, Transaction Manager, EJB3 etc
- Profile: the set of subsystems a server or group of servers runs
  - Change your profile to expand or narrow the capabilities of your servers

Example standalone.xml

```
<server name="example" xmlns="urn:jboss:domain:1.0">
    ....
    <profile>
        ....
        <subsystem xmlns="urn:jboss:domain:web:1.0">
            <connector name="http" protocol="HTTP/1.1"
                 socket-binding="http" scheme="http"/>
            <virtual-server name="localhost">
                <alias name="example.com"/>
            </virtual-server>
        </subsystem>
        <subsystem xmlns="urn:jboss:domain:weld:1.0"/>
    </profile>
    ....
```

Key Management Model Concepts

- Other configuration elements refer to sockets and interfaces by *logical names*, not specifics
  - `<connector socket-binding=”http” .../>`
  - Not `<connector address=”10.0.0.5” port=”8080”.../>`
- Allows centralized socket configs
- In Domain Mode, each host can control how a logical interface name resolves to an actual IP address

Example standalone.xml

```
...
  <interfaces>
    <interface name="local">
      <inet-address value="127.0.0.1"/>
    </interface>
    <interface name="wildcard">
      <any-ipv4-address/>
    </interface>
    <interface name="internal">
      <nic name="eth1"/>
    </interface>
  </interfaces>
  <socket-binding-group name="standard" default-interface="local">
    <socket-binding name="jndi" interface="internal" port="1099"/>
    <socket-binding name="jmx-connector-registry" port="1090"/>
    <socket-binding name="jmx-connector-server" port="1091"/>
    <socket-binding name="http" interface="wildcard" port="8080"/>
    ....
  </socket-binding-group>
```

Key Management Concepts – Deployments

- The configuration file includes a listing of available deployments

```
...
  <deployments>
    <deployment name="foo.war.v2" runtime-name="foo.war">
      <content hash="6fcd9eae343ed6d5aa9fffa83012d155b1ef911c"/>
    </deployment>
    <deployment name="foo.war.v1" runtime-name="foo.war"
                enabled=”false”>
      <content hash="dda9881fa7811b22f1424b4c5acccb13c71202bd"/>
    </deployment>
  </deployments>
</server>
```

CLI

- Launch from `bin` dir via `jboss-admin.sh` or `jboss-admin.bat`
- Connect to any DC, HC or standalone server
- Commands:
  - Low-level: provide resource address, operation name and params and you can invoke any operation exposed by any resource
  - High-level: simple convenience commands
- Can read commands from command line, file or an interactive shell

Demo

- Manage a standalone server using the CLI

Configuration of an AS Instance in Domain Mode

- An individual server's config comes from 2 sources
  - `domain/configuration/domain.xml` on host with DC
    - Elements that are consistent across the domain
  - `domain/configuration/host.xml` on each host
    - Elements specific to the host the server runs on
- Host Controller process combines domain.xml data + host.xml data to derive server config(s)

Domain-wide Configuration – domain.xml

- “Palettes” of config to apply to servers
  - One or more profiles (sets of subsystem configs)
  - One or more sets of socket configs
  - Available deployments
- Server Groups: sets of homogeneous servers managed as a unit
  - All servers belong to a group
  - `<server-group>` element specifies which items from the “palettes” – the profile, sockets, deployments – to use on servers in the group

Example domain.xml

```
<domain xmlns="urn:jboss:domain:1.0">
 ...
 <profiles>
   <profile name="web">... details of the web profile</profile>
   <profile name="messaging">... details of the messaging profile</profile>
 </profiles>
 <socket-binding-groups>
   <socket-binding-group name="web-sockets" default-interface="local">
     ... details of sockets in the 'web-sockets' group
   </socket-binding-group>
   <socket-binding-group name="msg-sockets" default-interface="local">
     ... details of sockets in the 'msg-sockets' group
   </socket-binding-group>
 </socket-binding-groups>
 ...
 <server-groups>
   <server-group name="web-group" profile="web">
     <socket-binding-group ref="web-sockets"/>
   </server-group>
   <server-group name="messaging-group" profile="messaging">
     <socket-binding-group ref="msg-sockets"/>
   </server-group>
 </server-groups>
</domain>
```

Example host.xml

```
<host name="host-1" xmlns="urn:jboss:domain:1.0">
  <management>
    <management-interfaces>
      <native-interface interface="internal" port="9999"/>
      <http-interface interface="internal" port="9990"/>
    </management-interfaces>
  </management>
  <domain-controller>
    <local/>  <!-- We are the Domain Controller -->
    <!-- if not: <remote address=”192.168.204.1” port=”9999”/> -->
  </domain-controller>
  <interfaces>
    <interface name="internal">
      <inet-address value="192.168.204.13"/>
    </interface>
  </interfaces>
  <servers>
    <server name="web-one" group="web-group"/>
    <server name="messaging-one" group="messaging-group"/>
  </servers>
</host>
```

Demo

- Domain management with the web console

CR1 “White Rabbit” Released!

Get it:

http://jboss.org/jbossas/downloads

Code it:

http://github.com/jbossas