



# Management and Monitoring - labs

Rostislav Svoboda

Senior Quality Assurance Engineer, JBoss by Red Hat

Advanced Java EE Lab @ FI MUNI

May 17th 2012

# Agenda

- Monitoring
  - JDK tools
  - System tools
- AS7 Domain
- CLI
- Java API
- HTTP API
- WebUI



# JDK tools - JAR level investigation

`jar tf $file` or `unzip -l $file`

```
jar tf jboss-modules.jar
```

```
for i in `find . -name '*.jar'`; do echo "-- $i"; jar tf "$i"; done
```

`javap -classpath $file FQCN`

```
javap -classpath jboss-modules.jar org.jboss.modules.JarModuleLoader
```

```
javap -private -classpath jboss-modules.jar org.jboss.modules.JarModuleLoader
```

`javap -c -classpath $file FQCN`

```
javap -c -classpath jboss-modules.jar org.jboss.modules.JarModuleLoader
```



# JDK tools – info about process

```
bin/standalone.sh &
```

```
bin/jboss-cli.sh -c command=":shutdown"
```

```
bin/jboss-cli.sh --connect controller=127.0.0.1:9999  
command=":shutdown"
```

```
jps -l [-m -v]
```

```
export PID=`jps -l | grep jboss-modules.jar | cut -d" " -f1`
```

```
jinfo $PID
```



# JDK tools – info about memory

`jmap $PID`

`jmap -heap $PID`

`jmap -dump:file=heap-dump $PID`

`jhat heap-dump`

Check <http://127.0.0.1:7000/>



# JDK tools – stack trace and JVM stats

```
jstack -l $PID
```

```
kill -QUIT $PID or Ctrl + \
```

```
jstat -gcutil -t $PID 1s 30
```

man jstat or <http://docs.oracle.com/javase/1.5.0/docs/tooldocs/share/jstat.html>

```
jstat -class $PID
```



# JDK tools – GUI

jconsole \$PID

bin/jconsole.sh

jvisualvm



# System tools

uname -a, cat /etc/redhat-release

top, cat /proc/cpuinfo

free, vmstat -a

df -h, du -h, mount

ps aux, top, kill -9

netstat -natup





# Domain

- bin/domain.sh
- docs/schema/jboss-as-config\_1\_2.xsd
  - which interfaces can be used to bind AS7
- bind AS7 to 127.0.0.2 (all services)
- check docs/schema/\*.xsd
- domain.xml, host.xml
  - reconfigure JVM parameters for main-server-group
  - server-one have http connector on port 8180



# CLI

- `bin/jboss-cli.sh -c controller=127.0.0.1:9999`
- Interactive mode
  - Commands
    - `cd, ls, deploy --help, undeploy --help`
    - <http://dl.dropbox.com/u/6677495/testapp.war>
  - Operations (`:whoami, :read-*`)
    - `:read-operation-description(name="read-attribute")`
    - `:read-resource(recursive=true,include-runtime=true)`
  - Tab completion



# CLI

- Non-interactive mode
  - Command and file arguments
    - `bin/jboss-cli.sh -c command="ls -l"`
    - `bin/jboss-cli.sh -c file=commands.cli`
      - `commands.cli` contains 2 lines:  
ls -l  
:whoami
- GUI mode
  - `bin/jboss-cli.sh -c --gui`



# Java API

- Maven artifact `org.jboss.as:jboss-as-controller-client`
- Check Git repo
- Read release version
- Read recursively resources include runtime
- Read resource descriptions about web subsystem
- Connect to remote AS7 instance



# HTTP API

## Simple get operations

- <http://localhost:9990/management?recursive&include-runtime&json.pretty>
  - `management?operation=resource-description&recursive&operations`
  - `management/subsystem/web/connector/http?include-runtime&json.pretty`
  - `management/subsystem/web?operation=operation-names&json.pretty`
- read just release version (attribute)
- profile ha, remoting subsystem, worker-task-max-threads value



# HTTP API

## Create HTTP connector on port 8181 for domain

```
curl --digest -L -D - http://localhost:9990/management --header "Content-Type: application/json" -d '{"address" : [{ "socket-binding-group" : "full-sockets" },{ "socket-binding" : "test" }], "operation" : "add", "port" : 8181, "json.pretty":1}' -u ferda:mravenec
```

```
curl --digest -L -D - http://localhost:9990/management --header "Content-Type: application/json" -d '{"address" : [{ "profile" : "full" },{ "subsystem" : "web" }, { "connector" : "test-connector" }], "operation" : "add", "socket-binding" : "test", "scheme" : "http", "protocol" : "HTTP/1.1", "enabled" : true, "json.pretty":1}' -u ferda:mravenec
```

```
curl --digest -L -D - http://localhost:9990/management --header "Content-Type: application/json" -d '{"address" : [{ "profile" : "full" }, { "subsystem" : "web" }, { "connector" : "test-connector" }], "operation" : "remove", "json.pretty":1}' -u ferda:mravenec
```

```
curl --digest -L -D - http://localhost:9990/management --header "Content-Type: application/json" -d '{"address" : [{ "socket-binding-group" : "full-sockets" },{ "socket-binding" : "test" }], "operation" : "remove", "json.pretty":1}' -u ferda:mravenec
```



# WebUI

<http://127.0.0.1:9990/console/>

- Check environment properties
- Start/stop server-one
- Create and run server instance (Heap 128m, PermGen 64m)
  - JVM configuration defined for particular server
  - JVM configuration defined for new server group
- Check ExampleDS datasource configuration, create TestDS
- Create JMS queue in Messaging subsystem



That's all ...

