Management and Monitoring - labs

Zbyněk Roubalík

Senior Quality Engineer, JBoss by Red Hat

Advanced Java EE Lab @ ČVUT

Nov 04 2016

Agenda

- Monitoring
  - JDK tools
  - System tools
- 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

JDK tools – process

bin/standalone.sh

jps -l [-m -v]

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

JDK tools – memory

jmap $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

jstat -gcutil -t $PID 1s 30

man jstat or http://docs.oracle.com/javase/7/docs/technotes/tools/share/jstat.html

JDK tools – GUI

jconsole $PID

bin/jconsole.sh

VisualVM

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

CLI

- bin/jboss-cli.sh -c
- Interactive mode
  - Tab completion
  - Commands
    - ls, cd, deploy, undeploy
  - Operations (:whoami, :read-*)
    - :read-operation-description(name="read-attribute")
    - :read-resource(recursive=true,include-runtime=true)

CLI

- Non-interactive mode
  - Commands 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.wildfly.core:wildfly-controller-client
- *management-00*: initial commit, start here
- *management-01*: Read product version
- *management-02*: Read recursively resources include runtime
- *management-03*: Read recursively resource description of remoting, namely http-remoting-connector
- *management-04*: connect remotely to running WF8 instance
- https://docs.jboss.org/author/display/WFLY10/The+native+management+API
-

Java API – Product Version App

- ProductVersionApp.java
- Get product version (WF version)
- Use Java API ***operation***

- Hint:
  - :read-attribute(name=product-version)

Java API – Resources Recursively App

- ResourcesRecursivelyApp.java
- Get all resources recursively
- Use Java API ***operation*** *and set* ***parameters***
  - Include runtime information
  - Set recursive depth

- Hint:
  - :read-resource(include-runtime=true,recursive-depth=5)

Java API – Resources Description for Remoting subsystem App

- ResourcesDescriptionRemotingSubsystemApp.java
- Read resources description for http-remoting-connector in remoting subsystem
- Use Java API ***operation*** and set ***address*** (subsystem)
  - remoting / http-connector / http-remoting-connector
  - Read recursively

- Hint:
  - /subsystem=remoting/http-connector=http-remoting-connector:read-resource-description(recursive=true)
  -

Java API – Remote connection App

- RemoteConnectionApp.java
- In WF:
  - Add user with username ane password
  - Disable default-user
- In Java API:
  - Use specified credentials to connect via remote api
    - ModelControllerClient with client authentication
  - Run ***whoami operation*** to verify credentials

HTTP API

Simple get operations

- http://localhost:9990/management?recursive&include-runtime&json.pretty
  - management/subsystem/infinispan/cache-container/ejb
  - management/subsystem/remoting?operation=attribute&name=worker-task-max-threads

Web Console

http://127.0.0.1:9990/console/

- Check environment properties
- Reload server

- Check ExampleDS datasource configuration, create TestDS
- Create JMS queue in Messaging subsystem

Thank you for your attention.