

Apache Geronimo: Best Practices

Bruce Snyder
Jailey Solutions, LLC





Apache Geronimo

- Open Source Enterprise Infrastructure
- Highly modular architecture
- Initial manifestation as a J2EE Server
- Other possible configurations
 - Native Spring Integration
 - Bi-directional PicoContainer Integration
 - Native Portlet Integration
 - Native Web Services Integration
 - Deployment Chaining
 - and many more ...





What Is Apache Geronimo?

It is Not...

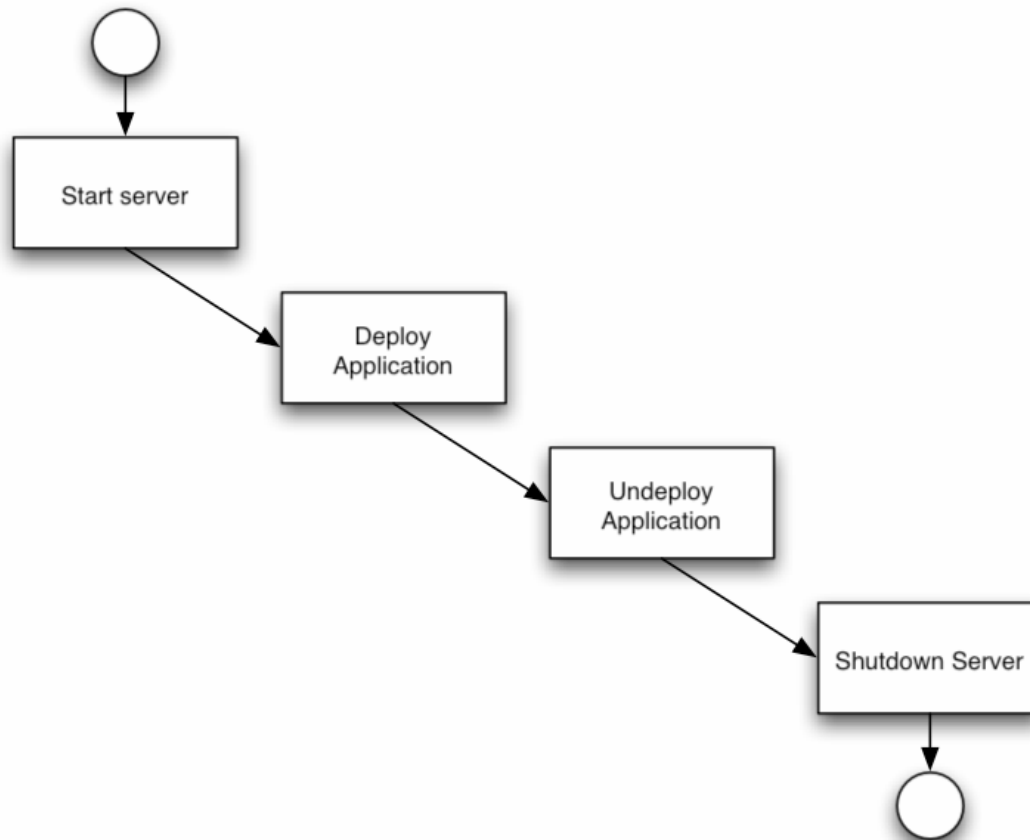
- Yet another lightweight container
- Yet another web framework
- Yet another AOP framework
- An MVC framework

It is...

- Designed for long running servers
- Designed to tolerate partial component failures
- System oriented services



Deployment with Geronimo





Install Geronimo

- With Maven


- Maven Geronimo Deploy plugin

- Install Geronimo:

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<project default="installGeronimo"
  xmlns:ant="jelly:ant"
  xmlns:deploy="geronimo:deploy">

  <goal name="installGeronimo">
    <ant:delete dir="${maven.build.dir}/geronimo"/>
    <deploy:unpackServer geronimoVersion="1.0-
SNAPSHOT"/>
  </goal>
</project>
```





Start Geronimo

- With Maven
 - Maven Geronimo Deploy plugin
 - Start Geronimo:

```
<goal name="startGeronimo">
  <deploy:startRemoteServer
    geronimoTarget="${maven.build.dir}/geronimo"
    configs="org/apache/geronimo/DebugConsole"/>
  <deploy:waitForStarted
    uri="deployer:geronimo:jmx:rmi://localhost/jndi/rmi:/JMXConnector"
    username="system"
    password="manager"
    id="org/apache/geronimo/DebugConsole"/>
  <echo message="Geronimo server has started"/>
</goal>
```





Deploy an Application

- With Maven
 - Maven Geronimo Deploy plugin
 - Deploy and start MyApp:

```
<goal name="startMyApp">
  <deploy:distribute
    uri="deployer:geronimo:jmx:rmi://localhost/jndi/rmi:/JMXConnector"
    username="system"
    password="manager"
    home="${basedir}"
    module="${maven.repo.local}/myproject/jars/myapplication-1.0-SNAPSHOT.jar"
  />
  <deploy:start
    uri="deployer:geronimo:jmx:rmi://localhost/jndi/rmi:/JMXConnector"
    username="system"
    password="manager"
    id="org/myproject/MyConfigurationName"/>
</goal>
```





Undeploy an Application

- With Maven
 - Maven Geronimo Deploy plugin
 - Stop and undeploy MyApp:

```
<goal name="stopMyApp">
  <deploy:stop
    uri="deployer:geronimo:jmx:rmi://localhost/jndi/rmi:/JMXConnector"
    username="system"
    password="manager"
    id="org/myproject/MyConfigurationName"/>
  <ant:echo>undeploy</ant:echo>
  <deploy:undeploy
    uri="deployer:geronimo:jmx:rmi://localhost/jndi/rmi:/JMXConnector"
    username="system"
    password="manager"
    id="org/myproject/MyConfigurationName"/>
</goal>
```





Stop Geronimo

- With Maven
 - Maven Geronimo Deploy plugin
 - Stop Geronimo:

```
<goal name="stopGeronimo">  
  <deploy:stopRemoteServer  
    uri="deployer:geronimo:jmx:rmi://localhost/jndi/rmi:/JMXConnector"  
    username="system"  
    password="manager"/>  
</goal>
```





Geronimo Deployment Tool

- JSR-88 deployment tool
- Command line tool for deploying modules (builders)
 - J2EE modules (initial Geronimo manifestation)
 - WAR modules
 - EJB modules
 - Spring modules
 - *etc.*
- Driven by a Deployment Plan



J2EE Deployment

- JSR-88 Deployment Steps
 - 1) Deployment
 - Read the ear file and the server config(s)
 - 2) Distribution
 - Send the ear file and server config(s) to the server
- Geronimo performs step one the first time only
 - Huge speed improvement on server startup





Geronimo Deployment Plan

- XML describing how to load a module
- File resides:
 - Inside jar next to spec-required deployment descriptor, or
 - Outside jar and located using additional deployment parameter
- `<gbean>` elements can also reside in:
 - Any descriptor
- Governed by an XSD



Deployment Plan Example (a)

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration
  xmlns="http://geronimo.apache.org/xml/ns/deployment"
  configId="org/apache/geronimo/Server"
  parentId="org/apache/geronimo/System">

  <dependency>
    <uri>
      geronimo-spec/jars/geronimo-spec-j2ee-${geronimo_spec_j2ee_version}.jar
    </uri>
  </dependency>
  <dependency>
    <uri>geronimo/jars/geronimo-j2ee-${geronimo_version}.jar</uri>
  </dependency>
```





Deployment Plan Example (b)

```
<gbean name="geronimo.security:type=SecurityRealm,realm=geronimo-properties-realm"
  class="org.apache.geronimo.security.realm.providers.PropertiesFileSecurityRealm">
  <attribute name="realmName" type="java.lang.String">
    geronimo-properties-realm
  </attribute>
  <attribute name="maxLoginModuleAge" type="long">10000</attribute>
  <attribute name="usersURI" type="java.net.URI">
    var/security/users.properties
  </attribute>
  <attribute name="groupsURI" type="java.net.URI">
    var/security/groups.properties
  </attribute>
  <reference name="ServerInfo">
    geronimo.system:role=ServerInfo
  </reference>
</gbean>
```





Install Geronimo

- Without Maven
 - Download binary:
<http://geronimo.apache.org/download.html>
 - Expand archive





Start Geronimo

- Without Maven
 - Start Geronimo:

```
$ java -jar bin/server.jar
```





Deploy an Application

- Without Maven
 - Deploy and start MyApp:

```
$ java -jar bin/deployer.jar  
    --install  
    --module YourWebApplication.war
```

```
$ java -jar bin/server.jar your/domain/name/Example
```





Undeploy an Application

- Without Maven
 - Stop and undeploy MyApp:

`Unavailable at the time of this writing`





Stop Geronimo

- Without Maven
 - Stop Geronimo:

CTRL-C





Packaging Applications

- Almost all descriptors created on the fly by the builders
 - Exception: `geronimo-ra.xml`





Logging

- Controlled by a GBean

```
<gbean name="geronimo.system:role=LogAppender,type=DailyRollingFileAppender"
  class="org.apache.geronimo.system.logging.log4j.appender.RollingFileAppenderService">
  <attribute name="threshold" type="java.lang.String">DEBUG</attribute>
  <attribute name="layoutPattern" type="java.lang.String">
    %d{ABSOLUTE} %-5p [%c{1}] %m%n
  </attribute>
  <attribute name="append" type="boolean">>true</attribute>
  <attribute name="file" type="java.lang.String">var/log/geronimo.log</attribute>
  <attribute name="bufferedIO" type="boolean">>false</attribute>
  <attribute name="bufferedSize" type="int">8192</attribute>
  <attribute name="maxBackupIndex" type="int">1</attribute>
  <attribute name="maxFileSize" type="java.lang.String">10MB</attribute>
  <reference name="ServerInfo">geronimo.system:role=ServerInfo</reference>
</gbean>
```





Spring Applications

- No deployer yet that recognizes spring.xml
- Start up an arbitrary collection of Spring enabled services
- Register Spring enabled services
- No GBean wrapper yet





Who Is Bruce Snyder?

- Independent software consultant
- Speaker at software conferences
 - JavaOne
 - ApacheCon
 - Colorado Software Summit
- The Castor Project (<http://www.castor.org/>)
- Apache Geronimo (<http://geronimo.apache.org/>)
- TranQL Project (<http://tranql.codehaus.org/>)
- Author

 ➤ *Castor Live* (SourceBeat Publishing)



Project Contributors

- Aaron Mulder
- Bruce Snyder
- Davanum Srinivas
- David Jencks
- Gianni D'Amour
- Jacek Laskowski
- Jan Bartel
- Jeremy Boynes
- Richard Monson-Haefel
- Srinath Perera
- Alan Cabrera
- Dain Sundstrom
- David Blevins
- Geir Magnusson Jr
- Greg Wilkins
- James Strachan
- Jason Dillon
- Jules Gosnell
- Simone Bordot





-
- Open up for Questions and Answers

