Farming using Deployment

{scrollbar}

INLINE

A configuration can be deployed to a cluster of Geronimo servers via a single logical deployment step.

Once deployed to a cluster, this configuration can then be transparently started, stopped or undeployed across all the cluster members.
These two features greatly streamline the maintenance of applications running on a cluster as Geronimo takes care of cluster-wide application distribution and management for you.

Configuration of Cluster Members

1. A specific name should be given to your Geronimo instance. This can be achieved by updating the `clusterNodeName` property defined within <g`eronimo_home>/var/config/config-substitutions.properties`: dashed clusterNodeName=*NODE1*
   RemoteDeployHostname=*NODE1_IP*
2. Cluster members are configured on a Geronimo server, which may but is not required to be a member of the cluster. This is achieved by adding a **org.apache.geronimo.farm.config.BasicNodeInfo** GBean for each cluster member to the **farming** configuration of `config.xml`. This GBean looks like this: <gbean name="org.apache.geronimo.configs/farming/2.2/car?ServiceModule=org.apache.geronimo.configs/farming/2.2/car, j2eeType=NodeInfo,name=Node2_description" gbeanInfo="org.apache.geronimo.farm.config.BasicNodeInfo"> <attribute name="name">NODE2< /attribute> <attribute propertyEditor="org.apache.geronimo.farm.config.BasicExtendedJMXConnectorInfoEditor" name=" extendedJMXConnectorInfo"> <ns:javabean class="org.apache.geronimo.farm.config.BasicExtendedJMXConnectorInfo" xmlns:ns4=" http://geronimo.apache.org/xml/ns/attributes-1.2" xmlns:ns="http://geronimo.apache.org/xml/ns/deployment/javabean-1.0" xmlns=""> <ns:property name="username">system</ns:property> <ns:property name="password">manager</ns:property> <ns:property name="protocol">rmi</ns: property> <ns:property name="host">NODE2_IP</ns:property> <ns:property name="port">1100</ns:property> <ns:property name="urlPath" >JMXConnector</ns:property> <ns:property name="local">false</ns:property> </ns:javabean> </attribute> </gbean> It defines network address (host, port, urlPath) and credentials (username and password) to be used to connect to the cluster member via JMX. Such declarations are to be included within the `config.xml` file, **farming** module of your Geronimo server.
   By default, the farming configuration defines the local server as a cluster member. To exclude it, you can prevent the GBean **NodeInfo** to start.

Farm Deployment

To deploy a configuration to configured members, you simply deploy it to the MasterConfigurationStore repository defined by the farming configuration. This configuration being stopped out-of-the-box, you may have to start it the first time via this GShell command:

deploy/start org.apache.geronimo.configs/farming//car

farming adds two new repositories to the server: **MasterConfigurationStore** and **ClusterStore**. **MasterConfigurationStore** is the repository you should use most of the time if not always. **ClusterStore** is a repository you may have to use in specific and infrequent scenarios.
To deploy to MasterConfigurationStore, you pass the --targets flag to the **distribute** or **deploy** commands like this:

deploy/distribute --targets XXX,name=MasterConfigurationStore <your module>

Following this deployment

- *MasterConfigurationStore* contains a kind of virtual configuration under an altered name, a '_G_MASTER' is appended to the configuration name, and
- *ClusterStore* contains the actual configuration which has the same name than your deployment and defines GBeans controlling the remote start and stop of the actual configuration cluster-wide;

Cluster-wide Management of Configurations

To undeploy across the farm, users need to execute the undeploy command on the same node that they used to deploy/distribute their application across the farm. Note that this node acts as a kind of Administration Server, whereby it is the only node whose farming configuration needs to be maintained to reflect farm members and any lifecycle operations against farmed modules (e.g. start/stop) are propagated to the other nodes. To start, stop or undeploy configurations across a cluster, you perform the relevant deployment task against the virtual configuration.

deploy/undeploy <your module>_G_MASTER

All the cluster members configured at the time of the initial deployment must be running otherwise these operations are partially executed. For instance, if three members were configured and only two of them are running when the virtual configuration is undeployed, then the configuration is removed from the ClusterStore of the two running servers. The ClusterStore of the third server still contains the configuration and this latter will have to be manually undeployed on this server.