IGS ZeRo Services Deployment

Identity Governance Service

Release 1.0.0

GEMEINSAM. DIGITAL. VERNETZT.

IGS ZeRo Services Deployment

1.2 Edition

by Adrien Farkas, Dieter Steding, and Sylvert Bernet

Program	Polizei 20/20
Program Director	Holger Gadorosi
Project Manager	Norbert Linde
Document Titel	IGS.ZeRo Services Deployment
Version	1
Creation Date **Created By**	2023-03-31 Adrien Farkas
Latest Update **Latest Update By**	2023-07-15 Dieter Steding

Revision History

Revision	Date	Author	Reference
1.0	2023-03-31	A. Farkas	First initial release
1.1	2023-05-26	A. Farkas	Added codeBase grant section
1.2	2023-07-15	D. Steding	Provisioning Service added

Table of Contents

Preface

This preface describes the document accessibility features and conventions used in this guide.

Audience

This guide is intended for resource administrators and target system integration teams.

Reference Documents

For information about installing and using Oracle Identity and Access Management, visit the following Oracle Help Center page:

- *https://docs.oracle.com/en/middleware/idm/suite/12.2.1.3/index.html*
- *https://docs.oracle.com/en/middleware/fusion-middleware/12.2.1.3/asadm/index.html*

Confidentiality

The material contained in this documentation represents proprietary, confidential information pertaining to Oracle products and methods.

The audience agrees that the information in this documentation shall not be disclosed outside of the project, and shall not be duplicated, used, or disclosed for any purpose other than to evaluate this procedure.

Typographical Conventions

The following table describes the typographic changes that are used in this document.

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Symbol Conventions

The following table explains symbols that might be used in this document.

Symbol	Meaning
[ ]	Contains optional arguments and command options.

Symbol	Meaning
{ | }	Contains a set of choices for a required command option.
${ }	Indicates a variable reference.
-	Joins simultaneous multiple keystrokes.
+	Joins consecutive multiple keystrokes.
>	Indicates menu item selection in a graphical user interface.

Deployment

To deploy the service the following high-level steps need to be followed:

- Assigning Codebase Grant's
- Deploying WebApplication Archives

Assigning Codebase Grant's

For authentication and authorization purposes, the applications use either technical or personalized user accounts to authenticate and authorize those accounts to the service interfaces that Oracle Identity Governance Suite offers.

In order to be able to carry out this authentication and authorization, the applications must be authorized for it. This is accomplished by granting appropriate permissions (*Codebase Grant*'s) to the binary archive files. *Codebase Grant*'s are are policies governing the execution rights of code running in a Java Virtual Machine (JVM).

The following *Codebase Grant*'s are required to make the service operational:

Element	Value
Resource Name	IdentityAssertion
Permission Actions	execute
Permission Class	oracle.security.jps.JpsPermission

Important

That permission have to be granted to `file:${XL.HomeDir}/client/oimclient.jar`.

To grant the required permissions you can:

- Granting using Enterprise Manager
- Granting using WLST

Granting using Enterprise Manager

The follwing steps needs to be performed to grant the permissions:

- Log in to the Enterprise Manager console using the user name and password you specified when creating the domain.

Granting using WLST

Below are the steps to grant the permissions to codebase using WLST:

1. Starting WLST shell by exceuting:

```
. igd.env
${WLS_DOMAIN}/bin/setDomainEnv.sh
```

```
${FMW_HOME}/oracle_common/common/bin/wlst.sh
```

You should see folowing output:

```
Initializing WebLogic Scripting Tool (WLST) ...

Jython scans all the jar files it can find at first startup. Depending on the system, this
process may take a few minutes to complete, and WLST may not return a prompt right away.

Welcome to WebLogic Server Administration Scripting Shell

Type help() for help on available commands

wls:/offline>
```

2. Connecting to the domain through:

```
import os
adminurl = os.getenv('ADMIN_URL')
username = 'weblogic'
password = '<password>'
connect(username, password, adminurl)
```

> **Note**
>
> Importing the Jython module `os` offers the capabilities to expand environment variables by calling `os.getenv`.

The shell will respond with:

```
Connecting to t3s://<server-host-name>:7002 with userid weblogic ...
Successfully connected to Admin Server "igd" that belongs to domain "identity".
```

3. Grant required permission with:

```
codeBase = 'file:${XL.HomeDir}/client/oimclient.jar'
grantPermission(codeBaseURL=codeBase, permTarget='IdentityAssertion', permActions='execute', permClass=
```

The output is similar to:

```
Location changed to domainRuntime tree. This is a read-only tree
with DomainMBean as the root MBean.
For more help, use help('domainRuntime')
```

4. Disconnect and exit the shell:

```
disconnect()
exit()
```

> **Important**
>
> The Admin Server and all Managed Servers needs to be restarted.

Deploying WebApplication Archives

All services are deployed as web applications in the same Weblogic Server domain as the applications of the Oracle Identity Governance Suite itself. This achieves a high degree of reusability of libraries provided by the Oracle Identity Governance Suite.

Deplyoment staging mode

WebLogic Server supports serving Web Applications from WAR files and from exploded archive directories. You also have a choice of *nostage* or *stage* mode of deployment. If you choose to serve a Web Application Archive file using *nostage* mode, the Web Application will be served directly from the packed or exploded archive.

The deployment staging mode determines how a module's archive files are made available to target servers that must deploy the module. WebLogic Server provides three different options for staging archive files: *stage* mode, *nostage* mode, and *external_stage* mode. You can set the staging mode either at the WebLogic Server level or at the application level, which overrides the server setting.

Stage mode

Stage mode indicates that the Administration Server copies the deployment files from their original location to the staging directories of each targeted server. For example, if you deploy a JEE Application to three servers in a cluster, the Administration Server copies the deployment files to directories on each of the three server machines. Each server then deploys the JEE Application using its local copy of the archive files.

Stage mode is the default (and preferred) mode when deploying to more than one WebLogic Server instance.

Nostage mode

Nostage mode indicates that the Administration Server does not copy the archive files from their source location. Instead, each targeted server must access the archive files from a single source directory for deployment. For example, if you deploy a JEE Application to three servers in a cluster, each server must be able to access the same application archive files (from a shared or network-mounted directory) to deploy the application.

In *nostage* mode, the web application container automatically detects changes to JSPs and servlets.

Nostage mode is the default mode when deploying only to the Administration Server (for example, in a single-server domain). You can also select *nostage* mode if you run a cluster of server instances on the same machine.

External_stage mode

External_stage mode is similar to *stage* mode, in that the deployment files must reside locally to each targeted server. However, the Administration Server does not automatically copy the deployment files to targeted servers in external_stage mode; instead, you must ensure that the files are copied to the staging directory of each targeted server.

External_stage mode is the least common deployment staging mode. It is generally used only in environments that are managed by third-party tools that automate the required copying of files.

Deployment process

To deploy the application you can:

- Deploying using WebLogic Administration Console
- Deploying using WebLogic Deployer
- Deploying using WLST

Deploying using WebLogic Administration Console

To deploy the WebApplication using WebLogic Administration Console perform following steps:

1. Log in to the console using the user name and password you specified when creating the domain.

2. (Optional) Lock console for **Edit** if necessary.

3. In the Domain Structure on the left, click **Deployments**.

4. Click **Install** and then navigate to the location where you downloaded the web application WAR file. Usually the directory of the location is `${ORACLE_HOME}/12.2.1/idm/server/apps`.

5. Select the WAR file `bka-zero-service.war` and click **Next**.

6. Select entire cluster **oic** or the appropriate server under the cluster **oic** and click **Next**.

7. Select **Install this deployment as an application** and click **Next**.

8. Click **Finish**.

9. (Optional) In the Change Center on the left, activate **Activate Change**s if necessary..

Deploying using WebLogic Deployer

To deploy the WebApplication with WebLogic Deployer execute the commandline:

```
. igd.env
${WLS_DOMAIN}/bin/setDomainEnv.sh
${JAVA_HOME}/bin/java -cp ${WLS_HOME}/server/lib/weblogic.jar weblogic.Deployer \
  -adminurl ${ADMIN_URL} \
  -user weblogic \
  -password <password> \
  -deploy \
  -name bka-zero-provisioning \
  -appversion 12.2.1.3 \
  -targets oim \
  -[nostage | stage] \
  -deploymentorder 501 \
  -securityModel DDOnly \
  -source ${OIM_ORACLE_HOME}/apps/bka-zero-provisioning.war
```

> **Note**
>
> Choose the approriate staging mode.

You should see something similar in the output:

```
weblogic.Deployer invoked with options:  -adminurl t3s:<server-host-name>:7002
-user weblogic -deploy -name bka-zero-provisioning -appversion 12.2.1.3 -targets oic
-nostage -verbose
-source /opt/oracle/product/iam/12.2.1/idm/server/apps/bka-zero-provisioning.war
<Jul 16, 2023 8:57:59 PM CEST> <Info> <J2EE Deployment SPI>
<BEA-260121> <Initiating deploy operation for application,
bka-zero-provisioning
[archive: /opt/oracle/product/iam/12.2.1/idm/server/apps/bka-zero-provisioning.war],
to oic.>
Task 0 initiated: [Deployer:149026]deploy application bka-zero-provisioning
[Version=12.2.1.3] on oic.
Task 0 completed: [Deployer:149026]deploy application bka-zero-provisioning
[Version=12.2.1.3] on oic.
Target state: deploy completed on Server oic

Target Assignments:
+ bka-zero-provisioning  oic
```

Deploying using WLST

Below are the steps to deploy the WebApplication using WLST:

1. Starting WLST shell by exceuting:

   ```
   . igd.env
   ${WLS_DOMAIN}/bin/setDomainEnv.sh
   ${FMW_HOME}/oracle_common/common/bin/wlst.sh
   ```

   You should see folowing output:

   ```
   Initializing WebLogic Scripting Tool (WLST) ...

   Jython scans all the jar files it can find at first startup. Depending on the system, this
   process may take a few minutes to complete, and WLST may not return a prompt right away.

   Welcome to WebLogic Server Administration Scripting Shell

   Type help() for help on available commands

   wls:/offline>
   ```

2. Connecting to the domain through:

   ```
   import os
   adminurl = os.getenv('ADMIN_URL')
   username = 'weblogic'
   password = '<password>'
   connect(username, password, adminurl)
   ```

   Note

   Importing the Jython module `os` offers the capabilities to expand environment variables by calling `os.getenv`.

   The shell will respond with:

```
Connecting to t3s://<server-host-name>:7002 with userid weblogic ...
Successfully connected to Admin Server "igd" that belongs to domain "identity".
```

3. Deploy the application with:

```
target   = 'oic'
module   = 'bka-zero-provisioning'
version  = '12.2.1.3'
artifact = os.getenv('OIM_ORACLE_HOME') + '/apps/' + module +'.war'
edit()
startEdit()
deploy(appName=module, appVersion=version, archiveVersion=version, targets=target, stageMode='nostage',
save()
activate(20000,block="true")
```

The output will look like:

```
Warning: Unrecognized option appVersion is being ignored
Deploying application from /opt/oracle/product/iam/12.2.1/idm/server/apps/bka-zero-provisioning.war to
<Jul 16, 2023 9:18:47 PM CEST> <Info> <J2EE Deployment SPI> <BEA-260121> <Initiating deploy operation f
..Completed the deployment of Application with status completed
Current Status of your Deployment:
Deployment command type: deploy
Deployment State : completed
Deployment Message : no message
```

4. Disconnect and exit the shell:

```
disconnect()
exit()
```

Configure Logging

Identity Governance Services using Oracle Diagnostic Logging (ODL) as its principal logging service. For ODL logging to work, both loggers and log handlers need to be configured. Loggers send messages to handlers, and handlers accept messages and output them to log files.

Logging configuration is controlled by the `logging.xml` file described in "Log Handler and Logger Configuration". This file can either be edited directly or edited through the Enterprise Manager. On the Enterprise Manager, the logging configuration can be accessed by clicking the *Identity Manager* server link and by selecting the WebLogic Server drop down from the top, and then clicking on **Logs - Log Configuration**.

To be continued ...

Apply Security

To access the services accounts are required to be authenticated and authorized by the services. For that purpose in the connected directory service you have to create:

- Enterprise Roles
- Technical Accounts

Enterprise Roles

The application assumes group principal names to be:

- `zero_admin`
- `zero_viewer`

If the names differ from the naming convention used throughout OIM deployment the **WEB-INF/weblogic.xml** file needs to be modified accordingly and the application redeployed.

The actual directory location for the groups should follow all IAM conventions in place, for the application only the **group name** (`cn` attribute) is important.

Technical Accounts

After the roles (groups) are created one or more account(s) needs to be created and made member of these groups. For the time being, only `zero_viewer` group member will be used.

The user should have password set according to the IAM policies and the user name (`uid` attribute) and password must be made known.

The actual directory location for the user(s) should follow all IAM conventions in place, for the application only the **user name** (`uid` attribute) is important.

HTTP-Proxy Configuration

For security reasons access to the services is proxied by a HTTP-WebServer deployed as a Reverse Proxy in the DMZ. The use of *reverse* originates in its counterpart *forward* proxy since the *reverse* proxy sits closer to the application server and serves only a restricted set of sites.

Therefore this HTTP-WebServer needs to be configured to reroute the traffic through its server and deliver it to the application server.

- TDB

Issues

Codebase Grant

Why the hell stametemant below can run without codebase grant?

```
final String home = System.getProperty("XL.HomeDir");
```

The expected behavior is the the server shuold unwilling to perform, but it doesn't throw any exception. As I understood Java Security at least either the deployment descriptor `weblogic.xml` or a security policy has to define:

```
<security-permission>

<description>
    Allow reading the XL.HomeDir system property.
  </description>


<security-permission-spec>
    grant {
      permission java.util.PropertyPermission "XL.HomeDir","read";
    };
  </security-permission-spec>

</security-permission>
```