Clustering Jedox with Five Machines

This article describes how to cluster Jedox components across five machines, whereby each machine runs a different Jedox component:

- Machine 1 running OLAP, the In-Memory DB
- Machine 2 running Apache, the Web UI server
- Machine 3 running Core, the Spreadsheet server
- Machine 4 running Tomcat Integrator, the ETL component
- Machine 5 running Tomcat RPC, the remote procedure call service

Most cluster configurations run Integrator and OLAP on one machine and run Core and Apache on separate machines. These instructions can be modified to accommodate that type of setup.

Before starting, you should prepare a location, such as a network drive, where stateful information can be saved (such as storage, reports, databases, and logs). Keep in mind that using a network drive requires the services to run under a user with network access.

To make a simple, recognizable example, we'll use these IP addresses in the configuration to symbolize the correct server:

- 10.0.0.1 for Machine 1 running OLAP

- 10.0.0.2 for Machine 2 running Apache
- 10.0.0.3 for Machine 3 running Core
- 10.0.0.4 for Machine 4 running Tomcat Integrator
- 10.0.0.5 for Machine 5 running Tomcat RPC

Setting up multiple machines involves making changes to the Jedox configuration files **palo.ini**, **server.xml**, **rpc.properties, etl-mngr.properties**, **httpd.conf**, **config.php**, and **palo_config.xml**. These files can be found in the following locations:

File name	Component	Linux path	Windows path
palo.ini	OLAP	<Jedox Path>/ps/Data/palo.ini	<Jedox Path>\palo\data\palo.ini
server.xml	TOMCAT ETL	<Jedox Path>/ps/tomcat-etl/conf/server.xml	<Jedox Path>\tomcat\conf\server.xml
component.xml	TOMCAT ETL	<Jedox Path>/ps/tomcat-etl/webapps/etlserver/config/standard/component.xml	<Jedox Path>\tomcat\webapps\etlserver\config\standard\component.xml
server.xml	TOMCAT RPC	<Jedox Path>/ps/tomcat-rpc/conf/server.xml	NA
rpc.properties	TOMCAT RPC	<Jedox Path>/tomcat-rpc/webapps/rpc/WEB-INF/classes/rpc.properties	<Jedox Path>\tomcat\webapps\rpc\WEB-INF\classes\rpc.properties
rpc.properties	TOMCAT RPC	<Jedox Path>/tomcat-rpc/webapps/hlbrowser/WEB-INF/classes/rpc.properties	<Jedox Path>\tomcat\webapps\hlbrowser\WEB-INF\classes\rpc.properties
etl-mngr.properties	TOMCAT RPC	<Jedox Path>/tomcat-rpc/webapps/hlbrowser/WEB-INF/classes/etl-mngr.properties	<Jedox Path>\tomcat\webapps\rpc\WEB-INF\classes\etl-mngr.properties
httpd.conf	APACHE	<Jedox Path>/ps/etc/httpd/conf/httpd.conf	<Jedox Path>\httpd\conf\httpd.conf
config.php	APACHE	<Jedox Path>/ps/httpd/app/etc/config.php	<Jedox Path>\httpd\app\etc\config.php
config.xml	CORE	<Jedox Path>/ps/core-Linux-x86_64/etc/config.xml	<Jedox Path>\core\config.xml
palo_config.xml	CORE	<Jedox Path>/ps/core-Linux-x86_64/etc/palo_config.xml	<Jedox Path>\core\palo_config.xml

Installing Jedox software

First, install the complete Jedox Suite on each of the machines. This step is necessary, because several components (including Tomcat) will need files from other Jedox Services.

Ensure that communication is not blocked through a firewall.

Standard ports are 7777, 7778, 80, 443, 8193, 7775, 7776, 8010, 8443

OLAP: 7777, 7778

Apache: 80, 443

Core: 8193

Tomcat: 7775, 7776,8010, 8443

Important!

Make a note of the following definitions from the config.php file on Machine 1. You will need them to set up the other machines, as noted in red in the steps that follow. Note that CFG_SECRET and CFG_PALO_PASS are randomly generated during the setup process. You need to copy the strings generated for Machine 1 to all of the other machines in the cluster.

The parameters below are just examples and will differ from your instance.

```
define('CFG_SECRET', '1234567890123456');
define('CFG_PALO_HOST', '10.0.0.1');
define('CFG_PALO_PORT', '7777');
define('CFG_PALO_PASS', ' a
12345678901234567890123456789012345678901234567890123456789012==');
```

Important: do not make any further changes to the settings on the OLAP machine! Just copy them to the config.php of the other machines.

Configuring Machine 1 (OLAP):

1. In **palo.ini**, change the http interface, e.g. 127.0.0.1, to **all** interfaces. This step allows OLAP to listen on the external

interface and to be accessible to the Web Host.

For our example, you would change

```
http "127.0.0.1" 7777
```

to:

```
http "" 7777
```

Note: the Supervision Server is running as a sub process of OLAP; therefore it's not necessary to change the port address for that.

Configuring Machine 2 (Apache):

1. In **httpd.conf**, set Apache to communicate with the other machines instead of localhost**.**
   For example, if the Tomcat RPC host is at 10.0.0.5, you would change

   ```
   Define JDX_HOST_TC_AJP "127.0.0.1"
   ```

   to:

   ```
   Define JDX_HOST_TC_AJP "10.0.0.5"
   ```

   Activate the header handling so it can send header information to other servers.

   Change the following lines from

   ```
   #Define JDX_HEADER
   ```

   to:

```
Define JDX_HEADER
```

Adjust the address of these lines from:

```
RequestHeader set X-JDX-Web-URL
"http://127.0.0.1:80"
RequestHeader set X-JDX-SSS-URL
"http://127.0.0.1:80/ub"
```

to the IP/DNS of the Apache host:

```
RequestHeader set X-JDX-Web-URL
"http://10.0.0.2:80"
RequestHeader set X-JDX-SSS-URL
"http://10.0.0.2:80/ub"
```

Adjust the address in the proxy for AJP calls, which will forward communication to the RPC servlet.

```
ProxyPass /tc/ws/
ws://10.0.0.5:${JDX_PORT_TC}/
ProxyPassReverse /tc/ws/
ws://10.0.0.5:${JDX_PORT_TC}/
```

This will forward communication to the CORE servlet.

```
BalancerMember
ajp://10.0.0.3:${JDX_PORT_SSS} retry=0
loadfactor=50 route=node1
ProxySet stickysession=JDX_WSS_BSID
```

2. In **config.php**, set Apache to communicate with Machine 1 (OLAP host) instead of localhost.
   For example:
   `define('CFG_PALO_HOST', '10.0.0.1');`
3. Adjust config.php to match settings of Machine 1 (OLAP host), as described in the text box above.

Configuring Machine 3 (Core):

1. In **palo_config.xml**, adjust the host address, port, and secret to match the settings of Machine 1 (OLAP host), as described in the text box above.
   For example, change
   `<host>127.0.0.1</host>`
   to
   `<host>10.0.0.1</host>`
2. Then change the secret you've made from Machine 1 (Olap)
   `<secret>70b4e3dd9f1ffe18</secret>`
   to
   `<secret>123456789012345</secret>`
3. Adjust **config.php** to match settings of Machine 1 (OLAP host), as described in the text box above.
4. Adjust **config.xml** to point to the address where Apache runs.
   For example, change

<web_service url="http://**127.0.0.1:80**" </>

to

<web_service url="http://**10.0.0.1:80**" </>

Note that the port must be adjusted if Apache is not running on 80.

Configuring Machine 4 (Tomcat-ETL):

1. In **server.xml**, change the address for HTTP to 0.0.0.0. This allows Tomcat to be visible from the outside and to listen to all interfaces.
   For our example, you would change
   <Connector port="7775" address="**127.0.0.1**" connectionTimeout="20000" protocol="HTTP/1.1" redirectPort="8443" />
   to:
   <Connector port="7775" address="**0.0.0.0**" connectionTimeout="20000" protocol="HTTP/1.1" redirectPort="8443" />
2. In the **component.xml** on the machine where the Tomcat is running, change
   <parameter name="web.url">http://127.0.0.1:80</parameter>
   to
   <parameter name="web.url">http://**10.0.0.2:80**</parameter

```
>
```

3. Adjust config.php to match settings of Machine 1 (OLAP host), as described in the text box above.

Note: On Windows you don't need machine 5 because there is no separate RPC Server. The steps from Machine 5 must be done on (this)Machine 4.

Configuring Machine 5 (Tomcat-RPC):

1. In **server.xml**, change the address for HTTP and AJP to 0.0.0.0. This allows Tomcat to be visible from the outside and to listen to all interfaces.
   For our example, you would change
   ```
   <Connector port="7776" address="127.0.0.1"
   connectionTimeout="20000"
   protocol="HTTP/1.1" redirectPort="8443" />
   <Connector port="8010" protocol="AJP/1.3"
   address="127.0.0.1" />
   ```
   to:
   ```
   <Connector port="7776" address="0.0.0.0"
   connectionTimeout="20000"
   protocol="HTTP/1.1" redirectPort="8443" />
   <Connector port="8010" protocol="AJP/1.3"
   address="0.0.0.0" />
   ```
2. In **rpc.properties.xml**, change the address for web.url

and sss.url to the Machine 2 (Apache) host.

For our example, you would change

`web.url=http://127.0.0.1`

`sss.url=http://127.0.0.1/ub`

to:

`web.url=http://10.0.0.2`

`sss.url=http://10.0.0.2/ub`

The next step is to activate the header handling for the Web RPC component, so it can determine which web server sends the requests.

Change

`web.url_from_hdr=false`

`sss.url_from_hdr=false`

to:

`web.url_from_hdr=true`

`sss.url_from_hdr=true`

3. In **etl-mngr.properties**, change the URL for etl.server.url to the host of Machine 4 (Tomcat Integrator).

   For our example, you would change

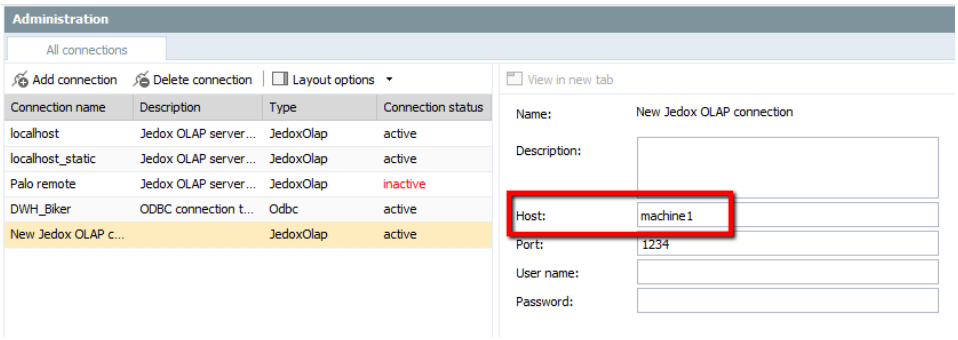
   `etl.server.url=http://127.0.0.1:7775/etlserver/services/ETL-Server?wsdl`

   to:

   `etl.server.url=http://10.0.0.4:7775/etlserver/services/ETL-Server?wsdl`

4. In /tomcat-rpc/webapps/hlbrowser/WEB-INF/classes/**rpc.properties** make the following adjustment:
   For our example, change
   `web.url=http://127.0.0.1`
   `sss.url=http://127.0.0.1/ub`
   to:
   `web.url=http://10.0.0.2`
   `sss.url=http://10.0.0.2/ub`
5. Adjust config.php to match settings of Machine 1 (OLAP host), as described above.

In Jedox Web, open **Administration→Connections** and make sure the connections point to the IP/DNS of Machine 1 (OLAP) host, **not** localhost. In the example below, the host machine is specified as machine1, not 127.0.0.1 (localhost IP).



Preparing the management of stateful information:

Important: You need to synchronize `storage` and `pr` from Machine 1 to Machine 2 (Apache) and Machine 3 (Core), so that all web servers use the same information.

The directory paths for `storage` and `pr` are

- **Linux:** `<Jedox Path>/ps/storage` and `<Jedox Path>/ps/htdocs/app/docroot/pr`
- **Windows**: `<Jedox Path>\storage` and `<Jedox Path>\httpd\app\docroot\pr`

One way of doing this is by mounting these directories. Below is an example for a mount command in Linux, executed on **Machine 2 (Apache)**:

```
sudo mount -t cifs -o user=Administrator%Adminpass //10.0.0.1/<Jedox Path>/ps/storage /<Jedox Path>/ps/storage
```

In Windows, you can mount a network drive or use mklink. (follow the link for more informations)

Starting and stopping the service/processes on

Machines 1 – 5

After you have verified that the system is properly configured, you can start the required processes on the machines.

Linux: enter the command `sudo <Jedox Path>/ps/jedox-suite.sh start service_name` (e.g. apache, olap, core, tomcat-rpc, tomcat-etl).

Windows: stop all services and start only the configured service (e.g. JedoxSuiteCoreService or JedoxSuiteHttpdService).