

Scaling by Clustering

This article describes how to cluster Jedox components on multiple machines, with a minimum of two machines:

- Machine 1 running a full Jedox installation (OLAP host with all Jedox components: Apache, Core, Tomcat, and OLAP)
- Machine 2...Machine N running Jedox Web host (Apache and Core)

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

File name	Linux path	Windows path
palo.ini	<Jedox Path>\ps\data\palo.ini	<Jedox Path>\palo\data\palo.ini
server.xml	<Jedox Path>\ps\tomcat\conf\server.xml	<Jedox Path>\tomcat\conf\server.xml
rpc.properties	<Jedox Path>\tomcat\webapps\rpc\WEB-INF\classes\rpc.properties	<Jedox Path>\tomcat\webapps\rpc\WEB-INF\classes\rpc.properties
httpd.conf	<Jedox Path>\ps\etc\httpd\conf\httpd.conf	<Jedox Path>\httpd\conf\httpd.conf
config.php	<Jedox Path>\ps\httpd\appletc\config.php	<Jedox Path>\httpd\appletc\config.php
palo_config.xml	<Jedox Path>\ps\core-Linux-x86_64\etc\palo_config.xml	<Jedox Path>\core\palo_config.xml

Installing Jedox software

First, install all required components on the machines. As an example, we will set up a Jedox cluster with 2 machines with the following IPs:

- Machine 1 (OLAP host) has a network interface on IP address 192.168.2.128
- Machine 2 (Web host) has a network interface on IP address 192.168.2.127

Only ports 80 and 7777 must be accessible from the outside for users accessing the Jedox system. **Note:** if you are adding SSL configurations, you need to open the port accordingly (standard is 443).

Configuring Machine 1 (OLAP, Tomcat, Web and Core):

1. In **palo.ini**, change the http interface, e.g. 127.0.0.1, to **all** interfaces. This 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
```

2. 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="7775" 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="7775" 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" />
```

3. In **rpc.properties**, set the Tomcat AJP port to point to the one defined in **server.xml** in step 2 above. Using the example above, the port would be set as follows:

```
tomcat.port=8010
```

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
```

to:

```
web.url_from_hdr=true
```

4. In **httpd.conf**, set Apache to communicate with Tomcat on Machine 1 (OLAP host) instead of localhost.

For example, if the OLAP host is at 192.168.2.128, you would change

```
Define JDX_HOST_TC_AJP "127.0.0.1"
```

to:

```
Define JDX_HOST_TC_AJP "192.168.2.128"
```

Then adjust the address in the proxy for AJP calls, e.g.:

```
ProxyPass /tc/ws/
```

```
ws://192.168.2.128:${JDX_PORT_TC}/  
ProxyPassReverse /tc/ws/  
ws://192.168.2.128:${JDX_PORT_TC}/
```

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

Change the following line from

```
#Define JDX_HEADER
```

to:

```
Define JDX_HEADER
```

Adjust the address of this line from:

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

to the IP/DNS of the OLAP host:

```
RequestHeader set X-JDX-Web-URL  
"http://192.168.2.128:80"
```

This information will be transmitted to distinguish between the different web servers.

5. In **config.php**, set Apache to communicate with Machine 1 (OLAP host) instead of localhost.

For example:

```
define('CFG_PALO_HOST', '192.168.2.128');
```

Make a note of the following definitions. You will need them

when you set up Machine 2 (see step 2 below). 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.

Also note that the parameters below are just examples and may differ from your instance.

```
define('CFG_SECRET', '1234567890123456');  
define('CFG_PALO_HOST', '192.168.2.128');  
define('CFG_PALO_PORT', '7777');  
define('CFG_PALO_PASS', ' a  
1234567890123456789012345678901234567890123456789012==  
' );
```

Important: do not make any further changes to the settings on the OLAP machine! Just copy them to config.php of Machine 2 (Web host).

6. In **palo_config.xml**, adjust the host address, port, and secret to match the ones listed in step 5 above.

For example, change

```
<host>127.0.0.1</host>
```

to

```
<host>192.168.2.128</host>
```

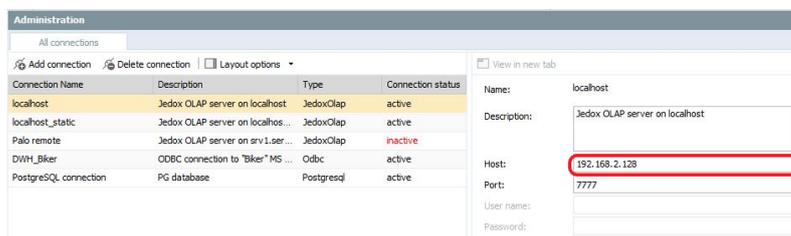
Then change

```
<secret>70b4e3dd9f1ffe18</secret>
```

to

```
<secret>123456789012345</secret>
```

- In Jedox Web, open **Administration→Connections** and make sure the connections point to the IP/DNS of the OLAP host (e.g. 192.168.2.128), **not** localhost (127.0.0.1), as indicated in red below.



The screenshot shows the 'Administration' section of the Jedox web interface. On the left, there is a table of connections with columns for 'Connection Name', 'Description', 'Type', and 'Connection status'. The 'localhost' connection is highlighted in yellow and has an 'active' status. On the right, there is a configuration form for the selected connection. The 'Host' field is highlighted with a red circle and contains the IP address '192.168.2.128'. Other fields include 'Name' (localhost), 'Description' (Jedox OLAP server on localhost), 'Port' (7777), 'User name', and 'Password'.

Connection Name	Description	Type	Connection status
localhost	Jedox OLAP server on localhost	JedoxOlap	active
localhost_static	Jedox OLAP server on localhost...	JedoxOlap	active
Palo remote	Jedox OLAP server on srv1.ser...	JedoxOlap	inactive
DWH_Biker	ODBC connection to "Biker" MS ...	Odbc	active
PostgreSQL connection	PG database	Postgresql	active

Configuring Machine 2...Machine N (Web and Core):

- In **config.php**, enter the parameters that you noted in step 5 above. These settings enable Apache to communicate with OLAP on Machine 1 (OLAP host) instead of localhost. Again, note that the parameters below are just examples and may differ from your instance.


```
"http://127.0.0.1:80"
```

to the IP/DNS of **Machine 2 (not OLAP host)**:

```
RequestHeader set X-JDX-Web-URL
```

```
"http://192.168.2.127:80"
```

3. In **palo_config.xml**, adjust the address, port, and secret to match those of the OLAP host. This allows Core Spreadsheet Service to communicate with Machine 1 (OLAP host).

For example, change

```
<host>127.0.0.1</host>
```

to

```
<host>192.168.2.128</host>
```

Then change

```
<secret>70b4e3dd9f1ffe18</secret>
```

to

```
<secret>123456789012345</secret>
```

Important:

You need to synchronize `storage` and `pr` from the OLAP host on the Web host, so that all Web servers use the same ones. 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 to mount these directories. Below is an example for a mount command in Linux, executed on **Machine 2 (Web host)**:

```
sudo mount -t cifs -o user=Administrator%Adminpass //192.168.2.128/<Jedox Path>/ps/storage /<Jedox Path>/ps/storage
```

In Windows, you can mount a network drive or [mklink](#).

Starting the process on Machines 1 and 2...N

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

Starting Machine 1 (OLAP host)

Linux: enter the command `sudo <Jedox Path>/ps/jedox-suite.sh start`

Windows: restart all services (JedoxSuiteCoreService, JedoxSuiteHttpdService, JedoxSuiteMolapService, JedoxSuiteTomcatService)

Starting Machine 2...Machine N (Web hosts):

Linux: enter the following commands:

- `sudo <Jedox Path>/ps/jedox-suite.sh start httpd`
- `sudo <Jedox Path>/ps/jedox-suite.sh start Core`

Windows: stop all services and start only these services:

JedoxSuiteCoreService and JedoxSuiteHttpdService
