Server Clustering Guide

Introduction

Overview

This document is intended to describe the steps required to deploy **Joget Large Enterprise Edition (LEE)** in a clustered environment for scalability and redundancy.

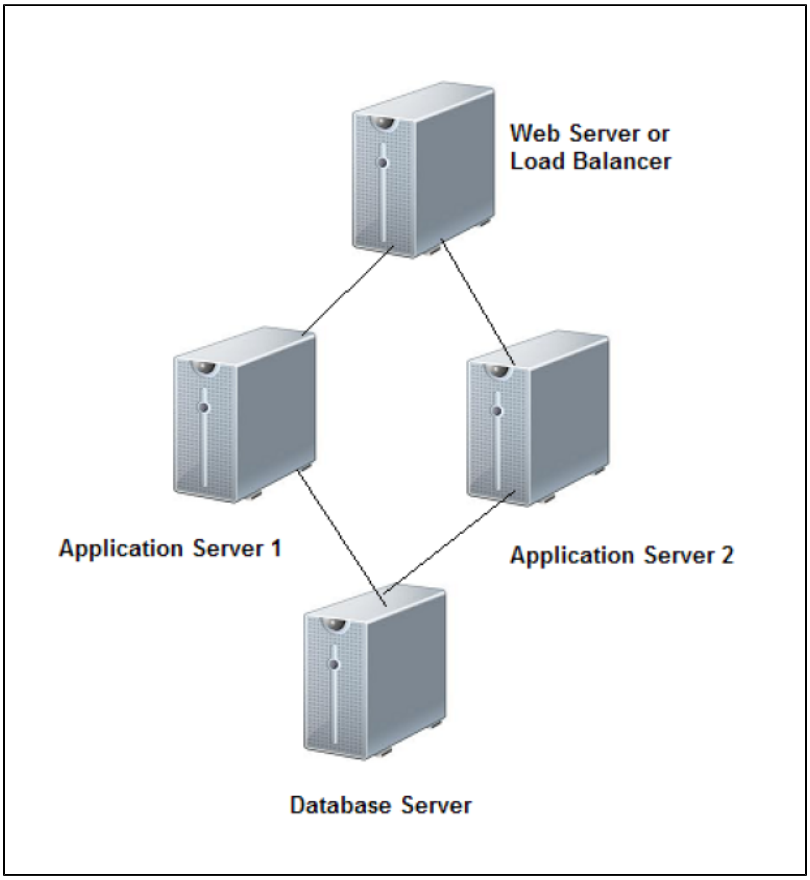
Requirements

In order for clustering to work, the **Large Enterprise Edition** is required. The standard Enterprise Edition will not work due to licensing restrictions. Clustering requires several layers to be prepared and configured:

- Load Balancers
- Application Servers
- Shared File Directory
- Shared Database

Architecture

There are many ways to design the clustering architecture, but the core concepts will be similar. In this document, the architecture used is as follows:



Deployment and Configuration Guide

This guide describes the steps required to setup Joget LEE clustering. The exact steps will depend on the actual products used in each layer.

> IMPORTANT: Please note that there is minimal configuration required in Joget LEE itself, and almost all the work is done on the separate layers so it is vital to ensure that you have sufficient expertise in your chosen products.

Pre-Deployment Requirements

Before the clustering installation can be done, the following prerequisites are needed:

Shared file directory

Common directory to be accessed by the application servers with read/write permissions. This directory is used to store shared configuration files, system generated files, and uploaded files. Verify that the shared directory is mounted on the application servers and that files can be accessed with read and write permissions.

Shared database

Common database to be accessed by the application servers with permission to select, update, delete, create and alter tables. Verify that the application servers can connect and query the shared database.

Application servers

Java web application server to be installed and running on each server in the cluster. Verify that each application server has been installed correctly and can be accessed with a web browser.

Session replication

Session replication to be configured on the application servers and network. Verify that session replication has been configured for each application server and the network.

Load balancer

Load balancer (hardware or software) to be installed and configured to direct traffic for requests beginning with **/jw** to the application servers. Verify that the load balancer has been installed and configured correctly so that web traffic is directed to the individual application servers.

Joget Clustering Configuration

It is important to ensure that the pre-deployment requirements have been verified. Once verified, the Joget specific steps are as follows:

Datasource Configuration

Configure the datasource properties files in the shared directory

1. Copy all the files and directories from the **wflow** directory of a Joget LEE bundle into the shared file directory.
2. Edit **app_datasource-default.properties** and set the database connection settings for the shared database, e.g. for MySQL, change the bold values below:

```
workflowDriver=com.mysql.jdbc.Driver
workflowUrl=jdbc\:mysql\://host\:port/database_name?characterEncoding\=UTF-8
workflowUser=username
profileName=
workflowPassword=password
```

Application Deployment and Configuration

Deploy the Joget WAR file to the application servers and configure the startup properties to point to the shared directory.

1. Deploy the WAR file **jw.war** from the LEE bundle to each of the application servers e.g. for Apache Tomcat, copy the files into the webapps directory
2. Add the following Java options in the application server startup e.g. for Apache Tomcat, modify the JAVA_OPTS line.

   Note: You can download the wflow-cluster.jar from here. The "wflow-cluster.jar" option must be added before any other "-javaagent" option.

```
export JAVA_OPTS="-Xmx1024M -Dwflow.home=/shared_directory_path -javaagent:/shared_directory_path/wflow-cluster.
jar -javaagent:/path_to/lib/aspectjweaver-1.8.5.jar -javaagent:/shared_directory_path/glowroot/glowroot.jar"
```

⚠ Please note that **-Dwflow.name=** is optional, but it is **required** if your nodes are in the **same server**. **-Dwflow.name=** must be placed before **-javaagent:/shared_directory_path/wflow-cluster.jar**.

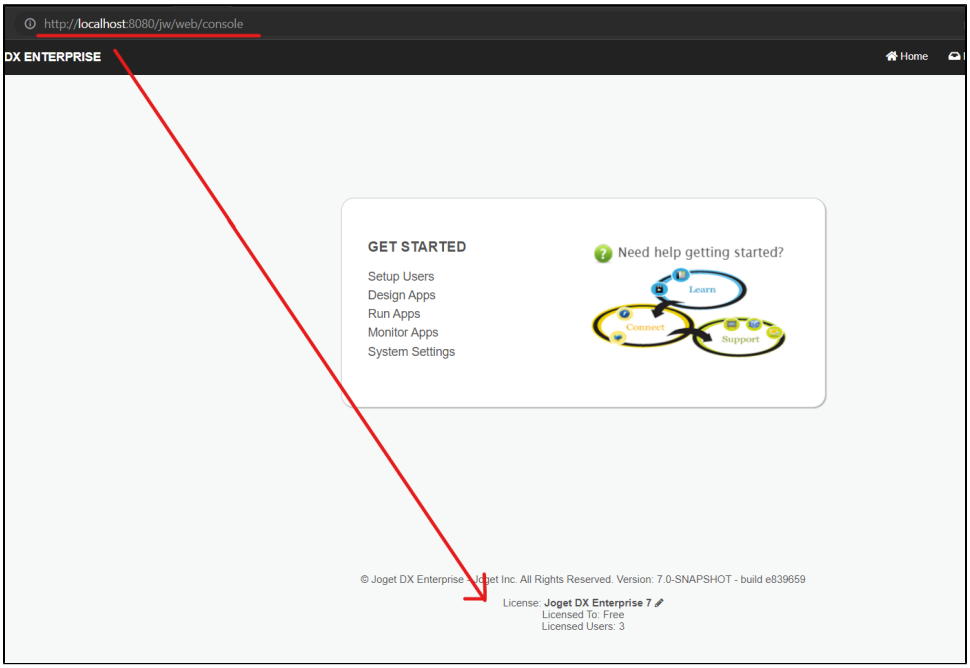
Example: **-Dwflow.name=node1**

License Activation

Activate license for each server. Each server has a unique system key and requires a separate license activation.

For more info about license activation, see Activate your Joget DX Enterprise License.

1. For each of the application servers, use the browser to directly access the Joget web console bypassing the load balancer e.g. http://server1:8080/jw/web/console/home
2. Request for license and activate it using the link in the web console footer.



Post-Deployment Testing

Once the pre-deployment and clustering configuration has been done, the testing is a matter of using a web browser to access the load balancer.

Sample Installation and Configuration

This sample describes an installation using the following products:

Joget	Joget DX LEE
Load Balancer	Apache HTTP Web Server 2.4 with mod_proxy and mod_balancer (proxy and load balancing modules) running on Ubuntu 18.04
Application Servers	Apache Tomcat 8.5 running on Ubuntu 18.04
Shared File Directory	NFS on Ubuntu 18.04
Shared Database	MySQL 5.7 on Ubuntu 18.04

IMPORTANT: Please note that this is not a comprehensive guide and does not cover production-level requirements e.g. user permissions, network and database security, etc. Please ensure that these are covered by your system, network and database administrators.

Create a Shared File Directory

Share a file directory to be accessed by the application servers. This directory is used to store configuration files, system generated files, and uploaded files.
In this sample, the shared file directory will be a directory **/export/wflow** in the file server

In the file server, install the NFS server

```
sudo apt-get install portmap nfs-kernel-server
```

Create shared directory and set permission

```
sudo mkdir -p /export/wflow
sudo chown nobody:nogroup /export/wflow
```

Configure NFS to export the shared directory, edit **/etc/exports** to export the directory to the local 192.168.1.0 subnetwork with your favourite editor

```
sudo vim /etc/exports
```

The **/etc/exports** should contain the following:

```
/export/wflow 192.168.1.0/255.255.255.0(rw,no_subtree_check,async)
```

Export the shares and restart NFS service

```
sudo exportfs -ra
sudo service nfs-kernel-server restart
```

Mount the Shared Directory in the Application Servers

In the application servers, install the NFS client

```
apt-get install nfs-common
```

Create new directory **/opt/joget/shared/wflow** to mount the shared directory and set the directory permissions

```
sudo mkdir -p /opt/joget/shared/wflow
sudo chmod 777 /opt/joget/shared/wflow
```

Mount the shared directory.

```
sudo mount -t nfs wflow:/export/wflow /opt/joget/shared/wflow
```

Test read-write permissions to confirm that the directory sharing works.

```
echo test123 > /opt/joget/shared/wflow/test.txt
```

Create a Shared Database

Install MySQL (https://help.ubuntu.com/18.04/serverguide/mysql.html)

```
sudo apt-get install mysql-server
```

Create a database called **jwedb** accessible to the application servers.

```
mysql -u root
```

Run the following MySQL commands to create a blank database

```
create database jwedb;
quit
```

Populate the newly created database with the Joget database schema

```
mysql -uroot jwedb < /path/to/jwdb-mysql.sql
```

Configure database permissions

```
mysql -u root
```

Run the following MySQL commands to grant permissions to user **joget** and password **joget**

```
grant all privileges on jwedb.* to 'joget'@'%' identified by 'joget';
flush privileges;
quit
```

Configure MySQL to listen to database connections from remote hosts. Edit the my.cnf file with your favourite editor

```
sudo vim /etc/mysql/my.cnf
```

Comment away the bind-address directive by adding a # in front of the line

```
#bind-address = 127.0.0.1
```

Restart MySQL

```
sudo service mysql restart
```

In the application server, test a remote connection to the database server **database_host**

```
mysql -h database_host -u joget -p
```

Deploy Application Servers

Install Apache Tomcat on each of the application servers. In each application server, run the following to extract tomcat into /opt/joget:

```
sudo mkdir -p /opt/joget/
sudo tar xvfz apache-tomcat-8.5.41.tar.gz /opt/joget/
```

Start each application server

```
sudo cd /opt/joget/apache-tomcat-8.5.41
sudo ./bin/catalina.sh start
```

Open a web browser and access each server to confirm that http://server:8080/jw

Configure Application Server Session Replication

Configure Tomcat for clustering by editing apache-tomcat-8.5.41/conf/server.xml. Add **jvmRoute="node01"** to the **Engine** tag and uncomment the **Cluster** tag.

```
<Engine name="Catalina" defaultHost="localhost" jvmRoute="node01">
<Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster"/>
```

Configure local domain IP. Verify that the local server name resolves to the IP and not 127.0.1.1. Assuming the server name is server1 and the IP is 192.168.1.10, edit /etc/hosts and set:

```
192.168.1.10 server1
```

Verify multicast is enabled between the application servers by running **ifconfig** and look for MULTICAST. Try http://blogs.agilefaqs.com/2009/11/08/enabling-multicast-on-your-macos-unix/ if there are issues.

Restart the Tomcat servers.

```
sudo cd /opt/joget/apache-tomcat-8.5.41
sudo ./bin/catalina.sh stop
sudo ./bin/catalina.sh start
```

Verify session replication working between the application servers. The **catalina.out** log file in apache-tomcat-8.5.41/logs should show something similar to:

```
INFO: Starting clustering manager at localhost#/jw
Jan 17, 2016 11:21:32 AM org.apache.catalina.ha.session.DeltaManager getAllClusterSessions
INFO: Manager [localhost#/jw], requesting session state from org.apache.catalina.tribes.membership.MemberImpl
[tcp://{127, 0, 0, 1}:4001,{127, 0, 0, 1},4001, alive=55733886, securePort=-1, UDP Port=-1, id={-57 118 -98 -98
110 -38 64 -68 -74 -25 -29 101 46 103 5 -48 }, payload={}, command={}, domain={}, ]. This operation will
timeout if no session state has been received within 60 seconds.
Jan 17, 2016 11:21:32 AM org.apache.catalina.ha.session.DeltaManager waitForSendAllSessions
INFO: Manager [localhost#/jw]; session state send at 1/17/16 11:21 AM received in 104 ms.
```

More information on Tomcat clustering is at http://tomcat.apache.org/tomcat-8.5-doc/cluster-howto.html

Configure Load Balancer

In the load balancer server, install Apache HTTP Server

```
sudo apt-get install apache2
```

Install proxy and balancer modules

```
sudo a2enmod headers proxy proxy_balancer proxy_http
```

If you are running Apache 2.4, you will need to also enable the following module.

```
sudo a2enmod lbmethod_byrequests
```

Configure a new site with the proxy and balancer modules. Create a new file in /etc/apache2/sites-available, named jwsite

```
sudo vim /etc/apache2/sites-available/jwsite.conf
```

Add the contents

```
NameVirtualHost *
<VirtualHost *>
        DocumentRoot "/var/www/jwsite"
        ServerName localhost
        ServerAdmin support@example.com
```

```
         ErrorLog /var/log/apache2/jwsite-error.log
    CustomLog /var/log/apache2/jwsite-access.log combined
        DirectoryIndex index.html index.htm
        <Proxy balancer://wscluster>
                BalancerMember ws://server1:8080 route=node01
                BalancerMember ws://server2:8080 route=node02
                Order deny,allow
                Allow from all
        </Proxy>
        ProxyPass /jw/web/applog balancer://wscluster/jw/web/applog stickysession=JSESSIONID|jsessionid
        ProxyPassReverse /jw/web/applog balancer://wscluster/jw/web/applog
        <Proxy balancer://cluster>
                BalancerMember http://server1:8080 route=node01
                BalancerMember http://server2:8080 route=node02
                Order deny,allow
                Allow from all
        </Proxy>
        ProxyPass /jw balancer://cluster/jw stickysession=JSESSIONID|jsessionid
        ProxyPassReverse /jw balancer://cluster/jw
        ProxyPreserveHost On
</VirtualHost>
```

Enable the new site and restart Apache

```
sudo a2ensite jwsite
sudo service apache2 reload
```

Deploy and Configure Joget LEE

Deploy and configure Joget LEE as described earlier in **2.2 Joget Clustering Configuration**