Installing DXP in a Clustered Environment



Renáta Kovács
Actualización hace 2 años

This document has been updated and ported to Liferay Learn and is no longer maintained here.

Many enterprise environments utilize clustering for both scalability and availability. This article provides specific instructions for installing a basic configuration of Liferay DXP in a pre-existing clustered environment.

A common misconception is that by merely configuring Liferay DXP, a high-availability / clustered environment is created automatically. However, by definition, a clustered environment includes load balancers, clustered application servers and databases. Once the clustered environment is set up, Liferay DXP can then be installed into that environment. This article expounds upon the Clustering section of our User Guide by giving further instructions.

Users can also determine whether the cluster uses multicast or unicast settings. By default, Liferay DXP uses multicast clustering. In the `portal-ext.properties`, users can change the multicast port numbers so that they do not conflict with other instances running. If the user decides to use unicast cluster, users have several options available that are supported by Liferay DXP, such as TCP, Amazon S3, File Ping and JDBC Ping.

Resolution

To set up a fully clustered environment:

1. Cluster Activation Keys need to be deployed on each node.
2. All nodes are pointing to the same Liferay DXP database or database cluster.
3. Documents and Media repositories is accessible to all nodes of the cluster.
4. Search indexes are configured to use a separate search server (Elasticsearch / Solr).
5. The cache is distributed.
6. Hot deploy folders are configured for each node if you are not using centralized server farm deployment.
7. Patch levels are identical for each node.

CLUSTER ACTIVATION KEYS

DXP Instance.

Additionally, Cluster Link must be enabled for cluster activation keys to work. To do this, set the following in portal-ext.properties:

```
cluster.link.enabled=true
```

DATABASE

Make sure all nodes are pointed to the same Liferay database. Configure the JDBC from portal-ext.properties or directly on the application server.

To Test:

1. Start both Tomcats (Nodes 1 and 2) ***sequentially***. The reason is so that the Quartz Scheduler can elect a master node!
2. Log in and add a portlet (e.g. Hello Velocity) to Node 1.
3. On Node 2, refresh the page.

The addition should show up on Node 2. Repeat with the nodes reversed to test the other node.

DOCUMENT AND MEDIA LIBRARY SHARING

Please note that the following properties are specifically for use with *AdvancedFileSystemStore*.

The following properties in `portal-ext.properties` have been moved to OSGI configuration files:

```
dl.store.file.system.root.dir=

dl.store.impl=com.liferay.portal.store.file.system.AdvancedFileSystemStore
```

To:

```
osgi/configs:
com.liferay.portal.store.file.system.configuration.AdvancedFileSystemStoreConfiguratio
```

Property	Default	Required
rootDir	data/document_library	false

separate Document Library repositories.

To Test:

1. On Node 1, upload a document to the Document Library.
2. On Node 2, download the document.

If successful, the document should download. Repeat with the nodes reversed.

Note 1: Advanced File System Store is an available option for high availability environments. Besides Advanced File System Store, there are other options of sharing the Document and Media Library. Keep in mind that the different types of file stores cannot communicate with each other, so changing from one to the other will cause the portal to be unable to read previously uploaded files. If the user needs to change the type of store and preserve previously uploaded files, execute a File Store migration.

Note 2: If storing your documents in file system is not an option then DBStore storage method is available by using the following portal property:

```
dl.store.impl=com.liferay.portal.store.db.DBStore
```

Note 3: JCRStore on a database is another option. Because Jackrabbit does not create indexes on its own tables, over time this may be a performance penalty. Users must change manually the index for the primary key columns for all the Jackrabbit tables. Other configuration to take note of is the limit of the amount of connections to your database.

Note 4: The number of connections to the database is another factor. Consider increasing the number of database connections to the application server.

Note 5: For an in-depth description of each type of file store, see the official documentation for Liferay for all repository types found here: Document Repository Configuration.

SEARCH AND INDEX SHARING

Starting from Liferay DXP the search engine needs to be separated from the main Liferay server for scalability reasons. For it there are two ways to achieve it: Elasticsearch or Solr.

To Test:

1. On Node 1, go to *Control Panel -> Users* and create a new user
2. On Node 2, go to *Control Panel -> Users*. Verify that the new user has been created

Note: Storing indexes locally is not an option anymore: `lucene.replicate.write=true` is deprecated.

DISTRIBUTED CACHING (MULTICAST OR UNICAST?)

Distributed caching allows a Liferay cluster to share cache content among multiple cluster nodes via Ehcache. Liferay has a specific article on managing a distributed cache.

Note 1: Ehcache has a lot of different modifications that can be done to cache certain objects. Users can tune these settings for their needs. Please see more about the caching settings in the Distributed Caching section of the User Guide. For advanced optimization and configuration, please refer to the Ehcache documentation: http://www.ehcache.org/documentation/configuration

Note 2: To learn more about Ehcache's default cache replication techniques or to learn how to deploy the tuning cache to the portal, please see the Advanced Ehcache Configuration knowledge base article.

HOT DEPLOY FOLDERS

Hot deploy folders are a mechanism provided by Liferay to install new components in the system. Liferay listens to that folder and proceeds to install the files copied there into the local installation. By default the hot deploy folder is created at `${liferay.home}` . You may change the directory location and name of the folder by changing this property in your `portal-ext.properties`

```
# Set the directory to scan for layout templates, portlets, and themes to

# auto deploy.

#

auto.deploy.deploy.dir=${liferay.home}/deploy
```

Keep in mind that by default, all deployed modules must be individually deployed to every node in the cluster. Whether this is done manually, programmatically, or via your application server's centralized server farm deployment, is up to your organization's needs and requirements.

The main point here is that every Liferay node needs to have the same portlets/modules deployed, otherwise you may experience inconsistencies in behavior or even data corruption. At this time, Liferay ***does not*** have any mechanisms to verify configuration, patch level or portlet/module consistency, and it will be the responsibility of the system administrator to maintain consistency across all nodes.

PATCH LEVELS

Similar to hot deploy folders, the Liferay platform needs to have the same patches installed across all nodes. While Liferay does not provide a patch consistency mechanism, our documentation on Using Profiles with the Patching Tool can assist in patching multiple Liferay nodes in a cluster.

Other Issues to Check

- On some operating systems, IPv4 and IPv6 addresses are mixed so clustering will not work. To solve this, add the following JVM startup parameter:

```
–Djava.net.preferIPv4Stack=true.
```

- If you run your cluster node on the same machine you need to configure:
  - Application Server ports
  - OSGI console ports via portal-ext.properties:

```
module.framework.properties.osgi.console=localhost:11311
```

Additional Information

The links contained in this article will be updated as we create new content. Thank you for your understanding and patience.

- Document Repository Configuration (see this article if using a repository other than AdvancedFileStore)
- Whitepaper Documentation: DXP Deployment Checklist and Upgrading to Liferay DXP

¿Fue útil este artículo?

Sí No

Usuarios a los que les pareció útil: 0 de 0

¿Necesitas más ayuda?

Abre un ticket

Contacta con nosotros

Pregunta en la Comunidad

Liferay University
Forma a tu equipo en los puntos esenciales del uso de Liferay con videos detallados y ejercicios prácticos diseñados para simular una formación presencial

Liferay Global Services
Aumenta tu conocimiento sobre el producto y el desarrollo gracias a expertos que te ayudarán a maximizar tu ROI