

Setting up a Jackrabbit Clustering



Your Rating: Results: 163 rates



Content of this page has been updated to comply with Magnolia CMS 5.6.x and add some more details on how to configure your Jackrabbit Data Store properly which was not mentioned in its previous version. However audiences still able to retrieve previous version using Page History, choose [v. 19](#) in the list.

- [Scenario](#)
- [A Note on Clustering](#)
- [Before setup](#)
- [What shall we do](#)
 - [An overview of steps](#)
- [Magnolia properties](#)
- [repositories.xml](#)
- [Jackrabbit configuration file](#)
- [Set the cluster id](#)
 - [Sync Delay](#)
- [Subscribers](#)
- [Warning: loading of workspace configuration](#)
- [Verify your setup](#)
- [Clean up your Journal](#)

Scenario

-- Updated use-case to demonstrate clustering in Magnolia CMS 5.6.x. Since Forum module was deprecated so we switch to Contacts app and its workspace for easier to follow.

We want the two public instances to share the `comments` contacts which are stored in the `forum` contacts workspace. But otherwise we want to keep the content independent.

Magnolia demo bundle already included demo Contacts module with all of its related sample content, app, and configurations.

See:

- [Apache Jackrabbit : Clustering](#)
- <https://documentation.magnolia-cms.com/display/DOCS/Clustering>

A Note on Clustering

-- Thanks to [Bradley Andersen](#) for your provided info in this section

We can either cluster, or not cluster. Setting up clustering is harder, but, if we do not cluster, we need to deal with:

- Synchronization
- Transactional Activation
- Sticky Sessions (think PUR module) More things to back up
- Etc.

On the other hand, clustering introduces some problems:

- If you use PostgreSQL, the journal can grow to the point it shuts down the DB server
- It introduces a single point of failure
- You can't do a rolling update if you only have one DB
- Does not scale - a good rule of thumb seems to be: one DB connection per JCR workspace is open. In an OOTB configuration, there are about 30 JCR workspaces. If we're above, say, 4 publics, we actually have too many simultaneous DB connections.
- Each cluster node needs its own (private) file system and search index.

Note that certain things should naturally be clustered (unless we want to create a service to reverse-publish from a public to the author, and then the author to the other publics):

- User generated content such as comments written by site visitors
- Public User Accounts
- Forum Posts

A potential solution for all these issues is [Amazon Aurora](#).

A potential solution to the single point of failure problem is: create a redundant, second Jackrabbit cluster to avoid single point of failure in the content store.

Before setup

Please note that customers who want to use Clustering function have to follow Jackrabbit requirements below ([original link here](#)):



Clustering in Jackrabbit works as follows: content is shared between all cluster nodes.

- That means all Jackrabbit cluster nodes need access to the **same** persistent storage (persistence manager, data store, and repository file system).
- The **persistence manager must be clusterable** (eg. central database that allows for concurrent access, see [PersistenceManagerFAQ](#)); **any DataStore (file or DB) is clusterable** by its very nature, as they store content by unique hash ids.
- However, **each cluster node needs its own (private) repository directory**, including repository.xml file, workspace [FileSystem](#) and [Search](#) index.
- Every change made by one cluster node is reported in a journal, which can be either file based or written to some database.

What shall we do

We will use MySQL database which supported concurrent access for our persistence manager. Also we will need a shared folder (either NFS or local file system) for our DataStore location. At the end, all clustered content of Contacts will be stored in MySQL and its related binary objects (contact images in this case) will be stored in this shared folder.

Sample MySQL script to create 'magnolia_cluster' database, create 'admin' user using 'admin' password on 'localhost' and grant him all permissions on created DB:

```
CREATE USER 'admin'@'localhost' IDENTIFIED BY 'admin';
CREATE SCHEMA `magnolia_cluster` DEFAULT CHARACTER SET utf8 ;
GRANT ALL PRIVILEGES ON magnolia_cluster.* TO 'admin'@'localhost' WITH GRANT OPTION;
```

We will configure a clustered repository by changing Magnolia provided WEB-INF/config/default/repositories.xml file into a clustered one and duplicate WEB-INF/config/repo-conf/jackrabbit-bundle-mysql-search.xml to WEB-INF/config/repo-conf/jackrabbit-bundle-mysql-cluster.xml for its configuration. Note that we still keep our previous one for non-cluster content. This means you will have 2 repositories working at the same time when we start our Magnolia instance.



It is possible to use H2 file system persistence storage for non-cluster repository / content while configuring MySQL database persistence storage for clustered content.

An overview of steps

1. Configure Magnolia author and public system wide properties in `WEB-INF/config/default/magnolia.properties`
2. Configure author and public Jackrabbit repositories in `WEB-INF/config/default/repositories_cluster.xml` which is a duplication of Magnolia provided `WEB-INF/config/default/repositories.xml`
3. Configure your cluster details in `WEB-INF/config/repo-conf/jackrabbit-bundle-mysql-cluster.xml`

Magnolia properties

-- Reference here for a complete list of all configuration items [Configuration management](#).

As we mentioned above in the prerequisite, "Each cluster node must have its own repository configuration." → So we will use this property to set its repository location:

```
magnolia.repositories.cluster=${magnolia.home}/repositories_cluster
```

Just like "magnolia.repositories.jackrabbit.config" configuration item, you are also expected to provide cluster configuration file location in

```
magnolia.repositories.jackrabbit.cluster.config=WEB-INF/config/repo-conf/jackrabbit-bundle-mysql-cluster.xml
```

Also this property would help identifying the instance as a cluster master node. During installation and update Magnolia bootstraps content only into master nodes. This ensures that other (replica) nodes installed later don't override already bootstrapped content. *default is false*. Note that I'm setting it to *true* in our author instance for demonstration purpose, however you would have to consider where to put your master cluster due to your practical scenario.

```
magnolia.repositories.jackrabbit.cluster.master=true
```

repositories.xml



Note that the position where you put your Repository definition tag in 'repository.xml' will determine the initiation order of Magnolia CMS repositories. Clustered repository is recommended to be placed after default one so that Magnolia CMS related configurations could be initiated first.

1. add a new repository configuration in `.../WEB-INF/config/default/repositories.xml`

```
<!-- magnolia non-default repository -->
<Repository name="magnoliaccluster" provider="info.magnolia.jackrabbit.ProviderImpl" loadOnStartup="
true">
  <param name="configFile" value="${magnolia.repositories.jackrabbit.cluster.config}" />
  <param name="repositoryHome" value="${magnolia.repositories.cluster}" />
  <!-- the default node types are loaded automatically
  <param name="customNodeTypes" value="WEB-INF/config/repo-conf/nodetypes/magnolia_nodetypes.
xml" />
  -->
  <param name="contextFactoryClass" value="org.apache.jackrabbit.core.jndi.provider.
DummyInitialContextFactory" />
  <param name="providerURL" value="localhost" />
  <param name="bindName" value="cluster-${magnolia.webapp}" />
  <!-- since forum module has been deprecated, we switch to contacts module for demonstration. -->
  <!-- <workspace name="forum" /> -->
  <workspace name="contacts" />
</Repository>
```

2. add a mapping to the clustered repository for the workspace to tell the system that this workspace lives in a different repository (the clustered one)

```

<RepositoryMapping>
  <Map name="website" repositoryName="magnolia" workspaceName="website" />
  ...
  <!-- since forum module has been deprecated, we switch to contacts module for demonstration. -->
  <!-- <Map name="forum" repositoryName="magnoliacluster" workspaceName="forum" /> -->
  <Map name="contacts" repositoryName="magnoliacluster" workspaceName="contacts" />
</RepositoryMapping>

```

3. We already set `magnolia.repositories.jackrabbit.cluster.config` in the `magnolia.properties` to `WEB-INF/config/repo-conf/jackrabbit-bundle-mysql-cluster.xml` however you can use whatever folder you want in file system using absolute path.
 - see: <http://documentation.magnolia-cms.com/technical-guide/configuration-mechanisms.html>

Jackrabbit configuration file

see: <http://wiki.apache.org/jackrabbit/Clustering>

1. make a copy of the non-clustering configuration file (`jackrabbit-bundle-mysql-cluster.xml` in this case)
2. make sure that both the instances use the same underlying database (MySQL `magnolia_cluster` schema in this case)
 - a. Sample MySQL datasource configuration

```

<DataSources>
  <DataSource name="magnolia_cluster">
    <param name="driver" value="com.mysql.jdbc.Driver" />
    <param name="url" value="jdbc:mysql://localhost:3306/magnolia_cluster" />
    <param name="user" value="admin" />
    <param name="password" value="admin" />
    <param name="databaseType" value="mysql" />
    <param name="validationQuery" value="select 1" />
  </DataSource>
</DataSources>

```

3. add the cluster configuration to the configuration file

```

<Cluster syncDelay="2000" id="mclul">
  <Journal class="org.apache.jackrabbit.core.journal.DatabaseJournal">
    <param name="revision" value="${rep.home}/revision"/>
    <param name="driver" value="com.mysql.jdbc.Driver"/>
    <param name="url" value="jdbc:mysql://localhost:3306/magnolia_cluster"/>
    <param name="user" value="admin"/>
    <param name="password" value="admin"/>
    <param name="databaseType" value="mysql" />
    <param name="schemaObjectPrefix" value="JOURNAL_" />
  </Journal>
</Cluster>

```

4. Configure `DataStore` using your shared folder. **This section is important to share binary objects amongst your clustered instances.** Note that you could able to use database datastore by configure `org.apache.jackrabbit.core.data.db.DbDataStore` in below section. Reference to [Jackrabbit Datastore documentation](#) for more details on limitations, garbage collection, and the way it work.

```

<DataStore class="org.apache.jackrabbit.core.data.FileDataStore">
  <param name="path" value="YOUR_SHARED_CLUSTERED_LOCATION"/>
  <param name="minRecordLength" value="1024" />
</DataStore>

```



Note that your `'magnolia.repositories.cluster=${magnolia.home}/repositories_cluster'` must point to different physical locations on all your author and public instances due to Jackrabbit clustering requirement that 'each cluster node needs its own (private) repository directory'. However `'YOUR_SHARED_CLUSTERED_LOCATION'` in `DataStore FileDataStore` location must point to the same location on all your instances to share their binary data objects. Please don't confuse on this point otherwise you will get into trouble when starting the instances.

Set the cluster id

The cluster id identifies the instance and is used to write changes to the journal as well as to load changes from the journal. **Make sure this is a unique value and is not shared with the other nodes in the cluster.**

Cluster id can be defined either in the properties file (most convenient way) or in the persistence manager in the cluster configuration (both ways are used in the attached files):

```
<Cluster id="mclu1" syncDelay="2000">
  ....
</Cluster>
```

Setting the cluster id in the properties file, will save you from having two different persistence manager files with just this little change.

1. set `magnolia.clusterid` property in the `magnolia.properties` file
 - see: <http://documentation.magnolia-cms.com/technical-guide/configuration-mechanisms.html>

Sync Delay

By default, cluster nodes read the journal and update their state every 5 seconds (5000 milliseconds). To use a different value, set the attribute `syncDelay` in the cluster configuration. `syncDelay="2000"` means states are synch every 2000 milliseconds.

Subscribers

Make sure that the content is not activated to both the clustered instances.

- only one subscriber should have a subscription to the clustered workspace(s) in `/server/activation/subscribers/xxx/subscriptions`

Warning: loading of workspace configuration

Once a workspace has been created a copy of jackrabbit configuration is saved to the workspace folder (`workspace.xml`)

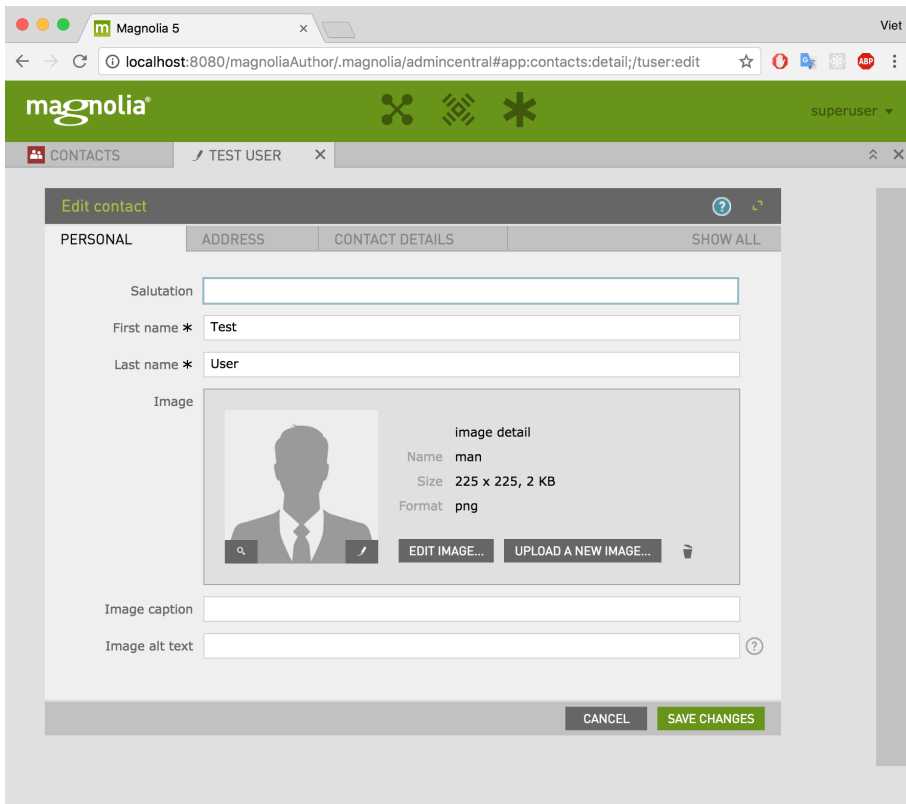
- changing the original jackrabbit configuration file won't have any effect
- changes have to be made in the `workspace.xml`

Verify your setup

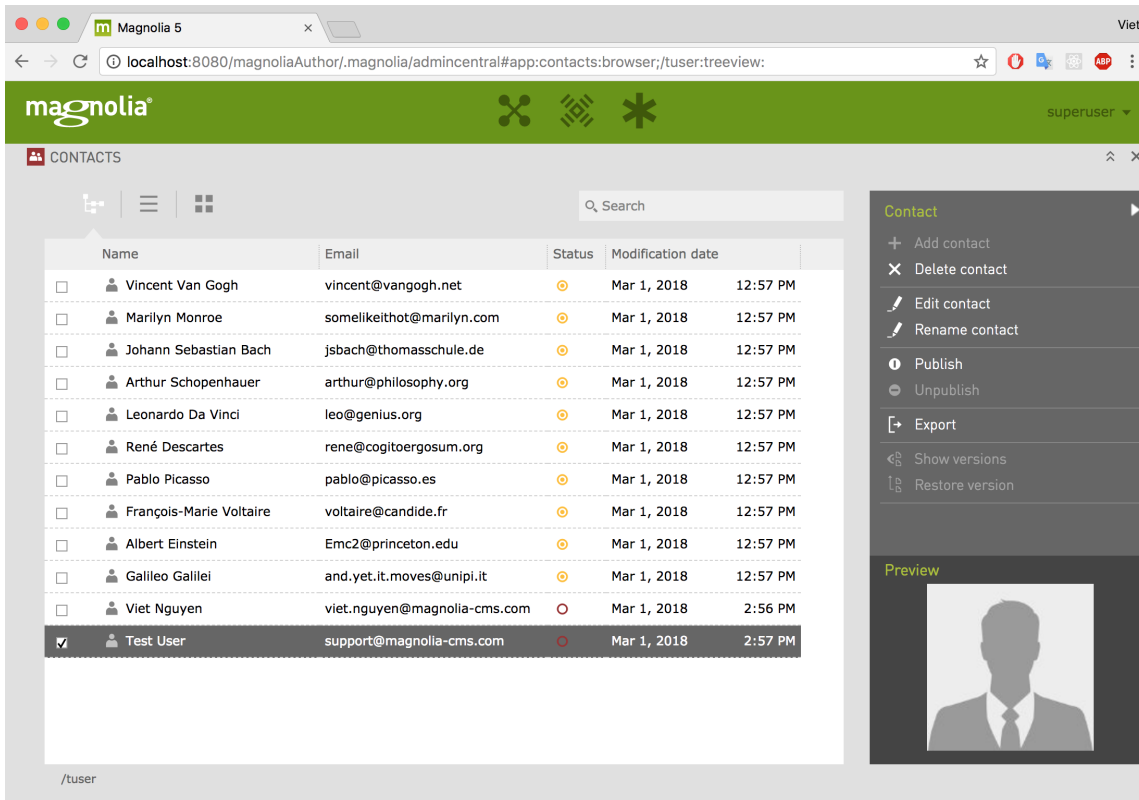
Bring up your instances, note that your author is our master cluster in this case, need to be installed first.

Then open your Contacts app such as (<http://localhost:8080/magnoliaAuthor/.magnolia/admincentral#app:contacts:browser;/:treeview:>)

Create a testing contact and upload an image for him



Remember to save your info



Switch to another instance, also open Contacts app (such as <http://localhost:8180/magnoliaPublic/.magnolia/admincentral#app:contacts:browser;/:treeview:>) and make sure that your created one was there (after `synchDelay=2000` milliseconds)

The screenshot shows the Magnolia CMS admin interface. The main content area displays a table of contacts with columns for Name, Email, and Modification date. The 'Test User' contact is selected, and its details are shown in a sidebar on the right. The sidebar includes options like 'Add contact', 'Delete contact', 'Edit contact', 'Rename contact', 'Publish', 'Unpublish', 'Export', 'Show versions', and 'Restore version'. A preview of the contact's profile picture is also visible.

Name	Email	Modification date
Vincent Van Gogh	vincent@vangogh.net	Mar 1, 2018 12:57 PM
Marilyn Monroe	somelikeithot@marilyn.com	Mar 1, 2018 12:57 PM
Johann Sebastian Bach	jsbach@thomasschule.de	Mar 1, 2018 12:57 PM
Arthur Schopenhauer	arthur@philosophy.org	Mar 1, 2018 12:57 PM
Leonardo Da Vinci	leo@genius.org	Mar 1, 2018 12:57 PM
René Descartes	rene@cogitoergosum.org	Mar 1, 2018 12:57 PM
Pablo Picasso	pablo@picasso.es	Mar 1, 2018 12:57 PM
François-Marie Voltaire	voltaire@candide.fr	Mar 1, 2018 12:57 PM
Albert Einstein	Emc2@princeton.edu	Mar 1, 2018 12:57 PM
Galileo Galilei	and.yet.it.moves@unipi.it	Mar 1, 2018 12:57 PM
Viet Nguyen	viet.nguyen@magnolia-cms.com	Mar 1, 2018 2:56 PM
Test User	support@magnolia-cms.com	Mar 1, 2018 2:57 PM

Clean up your Journal

This is important to prevent your database to be overloaded or hang. Thank you [Jordie Diepeveen](#) for reminding us about this.

! The journal can potentially become very large. By default, old revisions are not removed.

We recommend turning on the janitor functionality for clusters:

https://wiki.apache.org/jackrabbit/Clustering#Removing_Old_Revisions

```
<Cluster ....>
  <Journal ...>
    .....
    <param name="janitorEnabled" value="true" />
  </Journal>
</Cluster>
```

Reference to [Magnolia Clustering - Cleaning the Jackrabbit journal](#) and [Apache Jackrabbit Clustering - Removing Old Revisions](#) recommendations for more details.

Have a good day!