



Magnolia International Ltd.

## Cloud

### Magnolia CMS Tech Brief

Magnolia International Ltd.

10/13/2010

Copyright Magnolia International Ltd. 2010.

Magnolia is a registered trademark of Magnolia International Ltd.

All rights reserved. No part of this document may be reproduced in any form without prior permission in writing from Magnolia International Ltd.

## Table of contents

<b>Deploying Magnolia CMS in a cloud .....</b>	<b>3</b>
<b>Why cloud?.....</b>	<b>4</b>
<b>Possible solutions, types of cloud deployments.....</b>	<b>5</b>
<b>Technical requirements and limitations .....</b>	<b>8</b>
<b>To do .....</b>	<b>0</b>

Cloud computing happens on shared resources in the Internet. Services are provided to computers on demand, like water, electricity and other utilities. A benefit to CMS administrators is simplicity as they no longer need to know all the details of the underlying technology and infrastructure - the cloud provider takes care of that. The provider offers dynamically scalable, virtualized resources that can be managed with Web-based tools.

## Deploying Magnolia CMS in a cloud

Magnolia CMS is typically deployed as a stand-alone system. This means that each Magnolia instance is running on separate hardware or in a separate virtual machine. In order to scale up, you add more physical servers or deploy more virtual machines. In this setup each instance is completely independent, which means an instance can be shut down and started up independent of the other instances.

Contrast this with a cloud setup. In a cloud, Magnolia is deployed to cloud infrastructure such as Amazon EC2. A master deployment image ("start state") is created in the cloud. In order to deploy new instances, you clone the master. The result is an exact copy of the master. The master is configured as a clustered node. Any clones derived from it are also automatically part of the cluster and work together to spread load and ensure high availability. While you can achieve a similar result by deploying stand-alone instances to a cluster, the advantage provided by the cloud infrastructure is the ease of deployment. Additional instances are replicated from the master by the cloud infrastructure, which makes administration simpler.

## Cloud benefits

Generic cloud benefits:

- Responsiveness to peak load by creating new instances
- Cost is lower as infrastructure is provided by the cloud provider and pricing is usage-based
- Multi-tenancy enables sharing of resources and costs across users
- Reliability is improved with multiple redundant instances
- Scalability via dynamic ("on-demand") provisioning of instances
- Security through centralization of data
- Metering resources usage per client and application on a daily, weekly, monthly, and yearly basis.

Magnolia specific cloud benefits:

- Balance the load with a clustered setup
- On-demand allocation of resources makes it easier to deal with peak loads such as campaign launches
- No hardware maintenance, all costs are in one payment
- Automated allocation of resources to geo area, for example to deal with North American demand

## Possible solutions and types of cloud deployments

**Note!** Cloud deployments are only relevant to public Magnolia instances. Author instance is always a single instance.

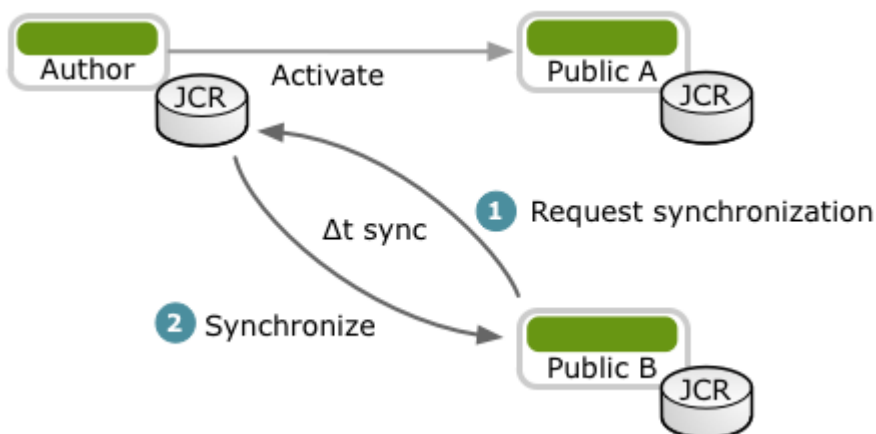
**Key requirement:** Jackrabbit clustering at repository level. Any public instances that are in the cloud have to have a clustered Jackrabbit repository. When you bring up another node you want all the content accessible to it instantly.

### Option 1: Instances have their own JCR (not recommended)

Each public instance has its own JCR content store. When a new public instance (Public B) is created:

1. Instance requests content synchronization from author.
2. Author activates all content to new instance.

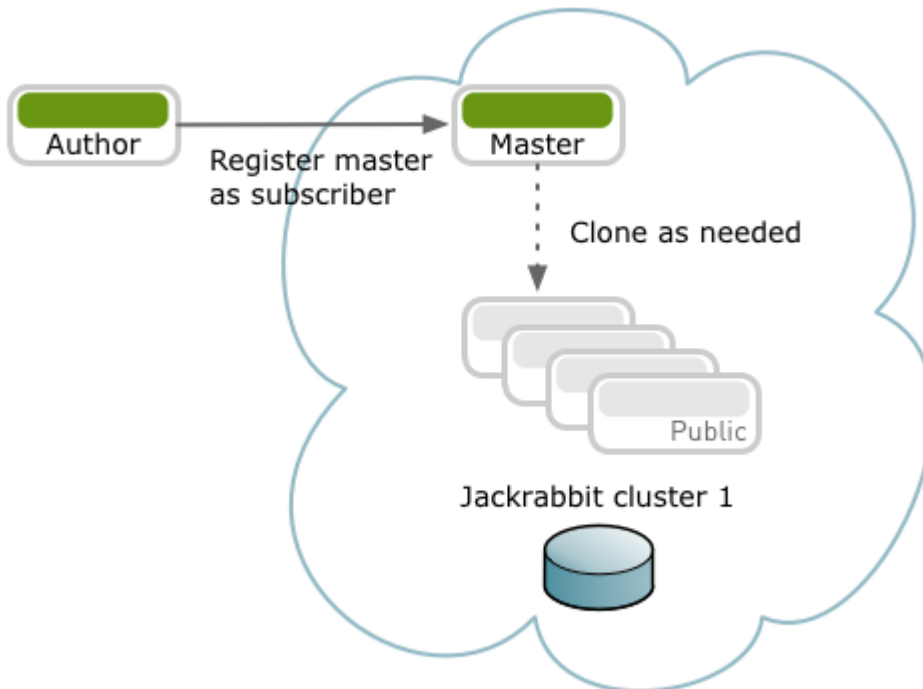
Time delay ( $\Delta t$  sync) before new public is in sync.



Downside: Author needs to activate to each instance. Bringing up a new instance requires sync of data, which means time delay. This setup is not ideal for responding to increased load rapidly.

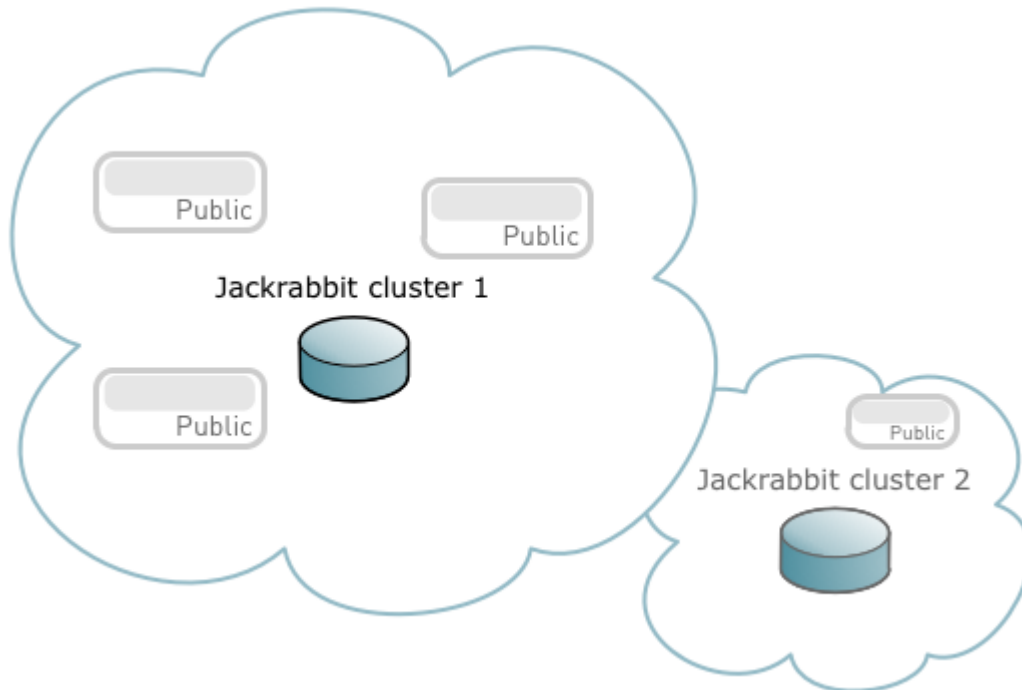
## Option 2: Content stored in a cluster (recommended)

Author activates only to master start state. New public instances are cloned from master as needed. Instance clones tap into a shared store. Instances don't need to register themselves as subscribers because all data is already in a clustered repository.



Downside: cluster (data store) is a single point of failure. This is a good high availability setup but doesn't guarantee zero downtime.

To mitigate the downside you can deploy two independent JCR clusters in the cloud, each connected to a separate Magnolia public master instance. Removes single point of failure.



## Technical requirements and limitations

### Requirements:

- Magnolia CMS 4.3.x or later
- Jackrabbit as repository. Fully clustered setup (all repositories and workspaces)

### Gains:

- High availability
- Scalability
- Security and other traditional cloud benefits
- Administration simplicity

### Losses:

- Complex environment setup initially. You need to set up clustering, make sure that new clones join the cluster, make sure that one public node is always running and that activation is configured on author to publish to it)
- Complex upgrade: shutting down all nodes means that no instance is serving public content during upgrade. Perform upgrade on a single node and then clone that node.
- Content storage is a single point of failure unless deployed in multiple clouds as illustrated above.