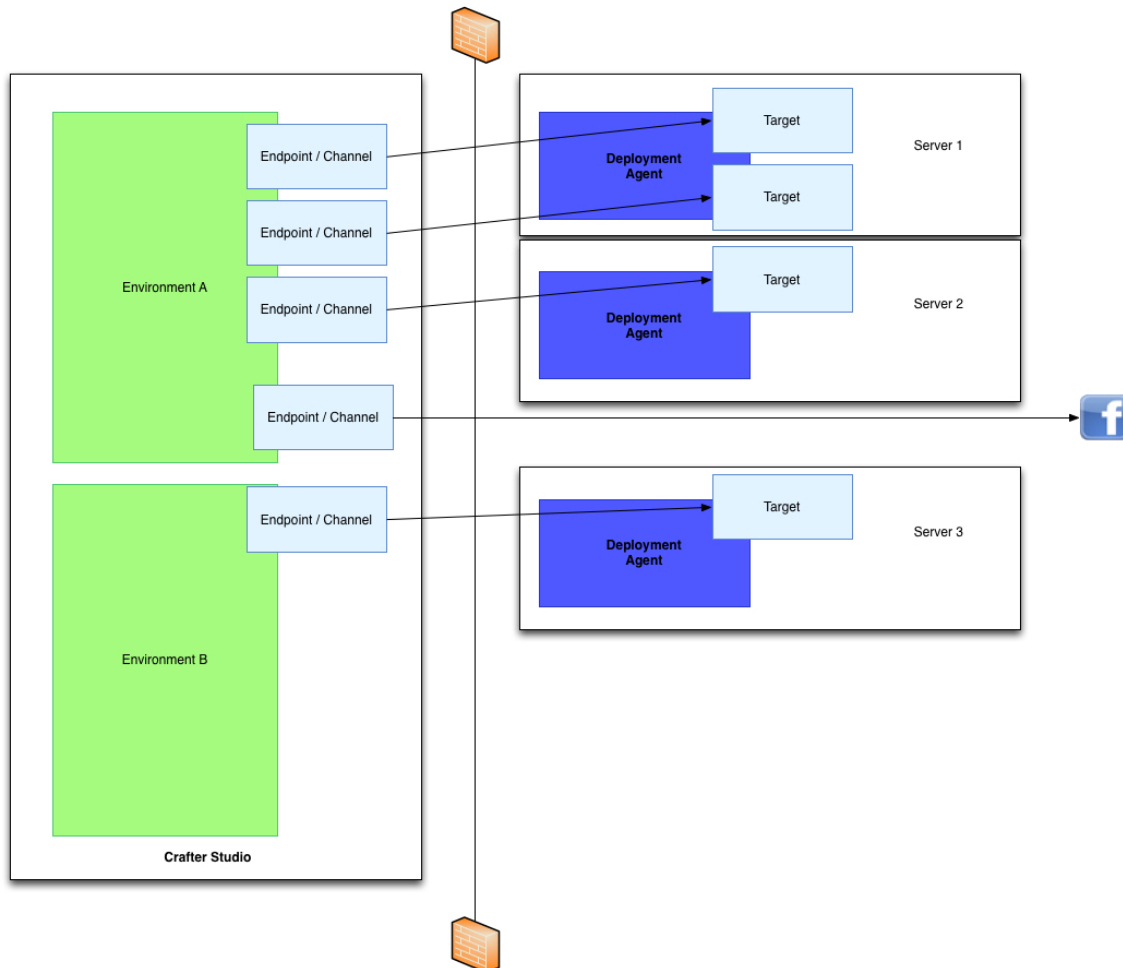


# Crafter Deployment Agent

- Deployment Model
- Configuring Endpoints in Crafter Studio
  - Defining Endpoints
  - Using Alfresco's Social Channels
  - Assigning Endpoints to Environments
- Crafter CMS Deployment Agent
- Configuring Crafter CMS Deployment Agent Targets
- Coding Callbacks
- Built In Callbacks

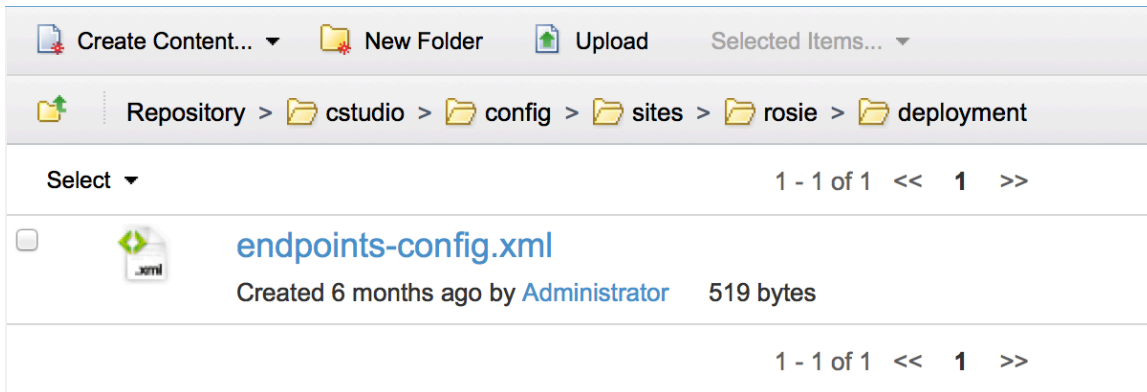
# Deployment Model

Below you see a diagram that illustrates the basic deployment model for Crafter CMS. For each web experience you are managing in Crafter Studio you can configure one or more environments (Live, Stage, etc.) Each environment is configured with one or more endpoints. An Endpoint represents a remote location where content will be deployed. The most common endpoint is a Crafter CMS deployment agent which is a simple process that runs at a remote location and can receive content for one or more targets (websites.)



## Configuring Endpoints in Crafter Studio

### Defining Endpoints



```
<config>
  <endpoint>
    <name>rosie</name>
    <type>crafterrivet-deployer</type>
    <server-url>http://localhost:9191/publish</server-url>
    <status-url>http://localhost:9191/api/1/monitoring/status</status-url>
    <version-url>http://localhost:9191/api/1/version</version-url>
    <bucket-size>10</bucket-size>
    <password>admin</password>
    <target>rosie</target>
    <send-metadata>true</send-metadata>
  </endpoint>
</config>
```

## Using Alfresco's Social Channels

Alfresco™

My Dashboard Sites People Repository More...

## Admin Console

**Tools**

- Application
- Category Manager
- Node Browser
- Tag Manager

**File Management**

- Trashcan


**Content Publishing**

- Channel Manager**

**Repository**


- Replication Jobs
- Repository Tools

### Channel Manager



acme.com Facebook Page

[Permissions](#) | [Reauthorize](#) | [Delete](#)




acme.com Twitter feed

[Permissions](#) | [Reauthorize](#) | [Delete](#)

## Assigning Endpoints to Environments

Repository > cstudio > config > sites > rosie > environment-overrides > local

Select 1 - 1 of 1 << 1 >>

<input type="checkbox"/>		<a href="#">environment-config.xml</a>	Created 6 months ago by Administrator 618 bytes
--------------------------	---	--	---

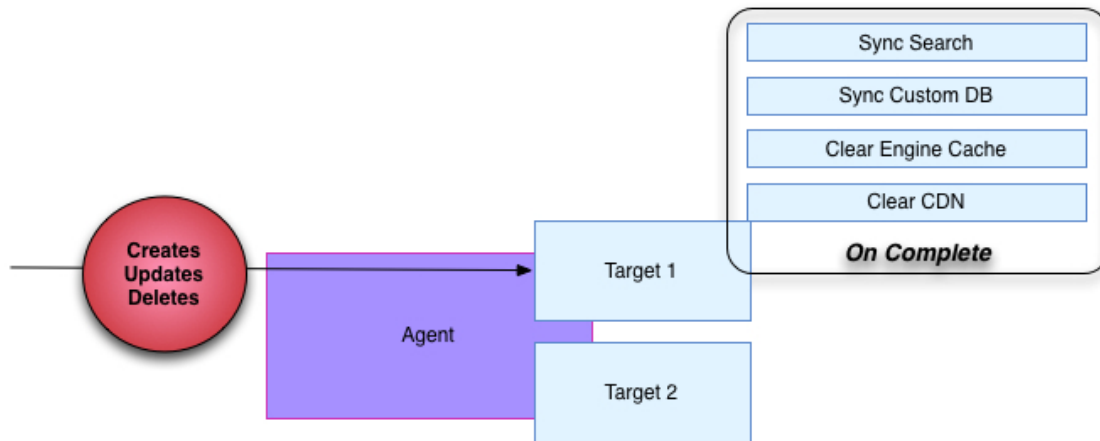
1 - 1 of 1 << 1 >>

```

<environment-config>
  <preview-server-url>http://127.0.0.1:8080</preview-server-url>
  <authoring-server-url>http://127.0.0.1:8080/share</authoring-server-url>
  <form-server-url>http://127.0.0.1:8080/form-server</form-server-url>
  <live-server-url>http://rosie</live-server-url>
  <publishing-channels>
    <channel-group>
      <label>Production</label>
      <channels>
        <channel>rosie</channel>
      </channels>
    </channel-group>
  </publishing-channels>
  <cookie-domain>127.0.0.1</cookie-domain>
  <open-site-dropdown>>false</open-site-dropdown>
</environment-config>

```

## Crafter CMS Deployment Agent



## Configuring Crafter CMS Deployment Agent Targets

```
<bean id="SampleTarget"
  class="org.craftercms.cstudio.publishing.target.PublishingTarget"
  init-method="register">
  <property name="name">
    <value>sample</value>
  </property>
  <property name="manager" ref="TargetManager" />
  <property name="postProcessors">
    <list>
      <ref bean="sampleCacheInvalidate" />
    </list>
  </property>
  <property name="params">
    <map>
      <entry key="root">
        <value>target/sample</value>
      </entry>
      <entry key="contentFolder">
        <value>content</value>
      </entry>
      <entry key="metadataFolder">
        <value>meta-data</value>
      </entry>
    </map>
  </property>
</bean>
```

## Coding Callbacks

```

package org.craftercms.cstudio.publishing.processor;

import java.util.Map;

import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import org.craftercms.cstudio.publishing.PublishedChangeSet;
import org.craftercms.cstudio.publishing.exception.PublishingException;
import org.craftercms.cstudio.publishing.target.PublishingTarget;

public class SamplePostProcessor implements PublishingProcessor {

    private static Log LOGGER = LogFactory.getLog(SamplePostProcessor.class);

    public void doProcess(PublishedChangeSet changeSet, Map<String, String> parameters, PublishingTarget target) throws PublishingException {
        if (LOGGER.isInfoEnabled()) {
            if (changeSet != null) {
                LOGGER.info("Printing the list of files deployed.");
                if (changeSet.getCreatedFiles() != null) {
                    for (String file : changeSet.getCreatedFiles()) { LOGGER.info("[CREATED] " + file); }
                }

                if (changeSet.getUpdatedFiles() != null) {
                    for (String file : changeSet.getUpdatedFiles()) { LOGGER.info("[UPATED] " + file); }
                }
                if (changeSet.getDeletedFiles() != null) {
                    for (String file : changeSet.getDeletedFiles()) { LOGGER.info("[DELETED] " + file); }
                }
                LOGGER.info("end of the list of files deployed.");
            }
        }
    }

    public String getName() {
        return "SamplePostProcessor";
    }
}

```

## Built In Callbacks

Built In Deployer Callbacks