Web Farm Installation

Version 54

Published 5/30/2011 10:00 AM by Vivek Thakur Last updated 7/26/2019 06:53 PM by Grace Kamau

---

Communifire Web Farm Settings and Configuration Guide

This guide is applicable to Communifire version 5.2 and above.

If you would like to deploy Communifire on a web farm (more than one server), you need to take care of the following points:

- Each server in the web farm will have its own instance of Communifire . This means each server will have its own copy of Communifire cached data.
- The database can be on a single server (the database server) and all running instances of Communifire will connect to this single Communifire database.
- You can use MSDeploy to configure Communifire on an IIS 6 based web farm. This link has detailed information on synchronizing IIS6 websites for a web farm scenario.
- For IIS7, refer these links: Shared Configuration on IIS7 and Configure a Web Farm on IIS7
- **All the instances of a Communifire application deployed under different servers on web farm must be having the same version of dlls.**

Machine key & Cookies in a Web Farm Deployment

If you deploy Communifire in a Web farm, you must ensure that the configuration files on each node share the same value for *validationKey* and *decryptionKey*, which are used for hashing and decryption respectively. This is required because you cannot guarantee which server will handle successive requests.

If you want to isolate Communifire from other applications on the same server, place the *<machineKey>* in the Web.config file for the Communifire application on each server in the farm. Do not use the keys for Communifire application in the config file for other web applications on the farm. You can use this link to generate machine keys:

http://www.developerfusion.com/tools/generatemachinekey/

Sharing Authentication Tickets Across Applications

If you need a single login to work across multiple applications (for example, Communifire is hosted in a virtual directory inside a parent web application and you want SSO from that parent application to work with Communifire) located in separate virtual directories, you need to share a common authentication ticket. To configure a common authentication ticket, you must manually generate validationKey and decryptionKey values and ensure that each application shares these values.

If you want to share tickets across all applications on your server you can set these manual values on the <machineKey> element in the machine level Web.config file. To share tickets across specific applications, you can use a <machineKey> element with common validationKey and decryptionKey values in the relevant application's Web.config files.

Cookie names in a web farm

Ensure that cookie names set in the web.config file are the same on each node in the Web farm. Refer the image below where we have set the cookie name to MyApplicationAUTHCookie (just an example name):

Shared Media Servers

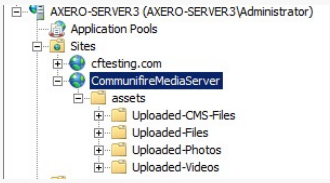
Communifire uses the concept of Media Storage Servers to store physical files (files uploaded by users, photos, videos etc). In a web farm environment, you can either have a single shared file server (UNC path) for all running instances of Communifire or use tools like DFS (Distributed File System) to synchronize files across servers. We recommend using shared UNC path instead of DFS as DFS required domain controller.

To share a single media server location, you need to map the media servers in Communifire to point to a network path on a local file server. All of the load balanced web servers share the same file store location. Network shared content uses a back-end file server to manage website content. All web servers point to a shared folder on the file server over a UNC path. To reduce the risk of failure, the file server is often mirrored to another server with some method

of fail over provided.

You can configure media servers in different ways. Let us take the following example to show the steps to setup the media servers using UNC paths. AXERO-SERVER1 and AXERO-SERVER2 are our two nodes in a web farm and AXERO-SERVER3 is the file server.

1. To configure shared content, you first set up the shared folder on the file server (AXERO-SERVER3 in our example). Create a custom user for each Communifire application pool (on each web server) and assign that user to the shared folder. You can create local users and groups as long as the same username and password is assigned to each web server (Unless these servers are not on a domain they will not recognize users from other devices. You can create identical users on both devices with the same username and password as a potential workaround). Make sure that the application pool identity (for each Communifire application) along with IUSR user has read-write access to the UNC path on AXERO-SERVER3. Refer the image below on how the IIS is setup for the UNC share path on the file server (using the site name as *CommunifireMediaServer*):



2. Next, on each web server in the farm, setup the below virtual directories to point to the shared UNC path. Here is the list of the four media server paths which should be set to a UNC paths:

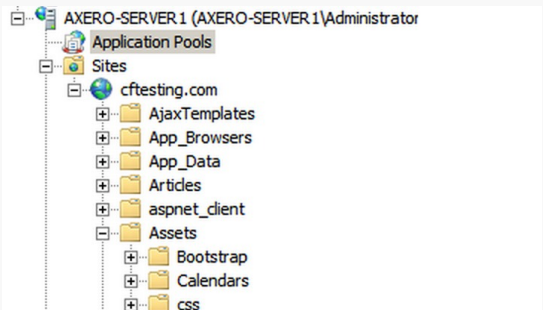
/assets/Uploaded-CMS-Files

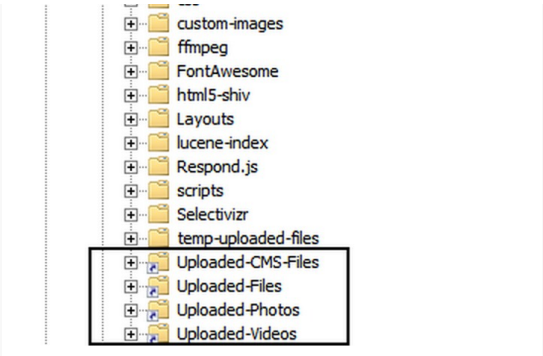
/assets/Uploaded-Files

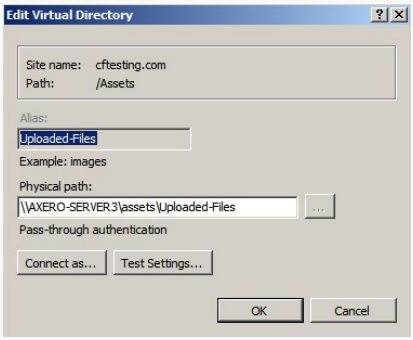
/assets/Uploaded-Photos

/assets/Uploaded-Videos

In this IIS screenshot below, AXERO-SERVER1 is a node in the webfarm, and we have created 4 virtual directories for each of the media servers and mapped them to the UNC path:

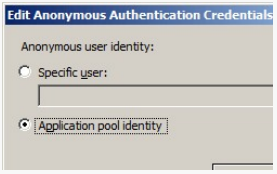






In the above image, AXERO-SERVER3 is the "file server", and we have set Uploaded-Files to point to this UNC path. Similarly the other three virtual directories need to be pointed to their respective UNC paths in AXERO-SERVER3 file server. And this step **has to be done on each of the node in the web farm**.

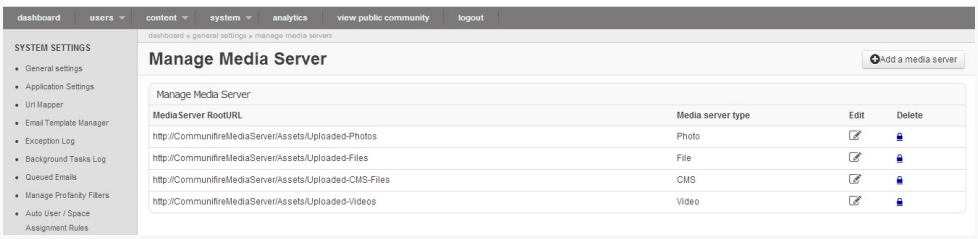
3. In order to allow other "Guest" users of Communifire application to access the /Assets and /Themes folder of other server (Media Server) you need to set "Anonymous Authentication Credentials" to "Application pool identity". The default setting when any site is created is "IUSER" but you need to change that to "Application pool identity" so that the credentials of the custom user account which is set as app pool identity can be used to access that mediaserver folder present at network path.



Further reference can be found here [Anonymous Authentication]

You need to change this setting of Authentication section in Communifire IIS site of all the servers present in web farm.

4. Set these media servers in the Communifire administration section as shown below:



Caching in a load balanced environment

By default, Communifire uses the ASP.NET Cache object for cache management. On a web farm, each server running Communifire will have its own in-memory cache store. If you change something (like create a new space) on a server, it will not get reflected on other servers unless the cache gets refreshed. This happens because Communifire caches all spaces, and also global settings.

To fix this:

- Edit the web.config file on other servers so that their cache gets refreshed.
- Use a distributed caching system like Redis or memcached. (Communifire comes with a default Redis and memcached provider)
- Use cache dependency to synchronize cache across all servers. Refer to this article for details.

We recommend using Redis.

SignalR Backplane in a load balanced environment

To make sure that Communifire notifications and chats (which are dependent on SignalR) work in a web farm, you need to configure the SignalR backplane. Communifire comes with Redis-based backplane for SignalR. Redis is an in-memory key-value store. It also supports a messaging system with a publish/subscribe model. The SignalR Redis backplane uses the pub/sub feature to forward messages to other servers.

To configure Redis backplane in Communifire, follows these instructions:

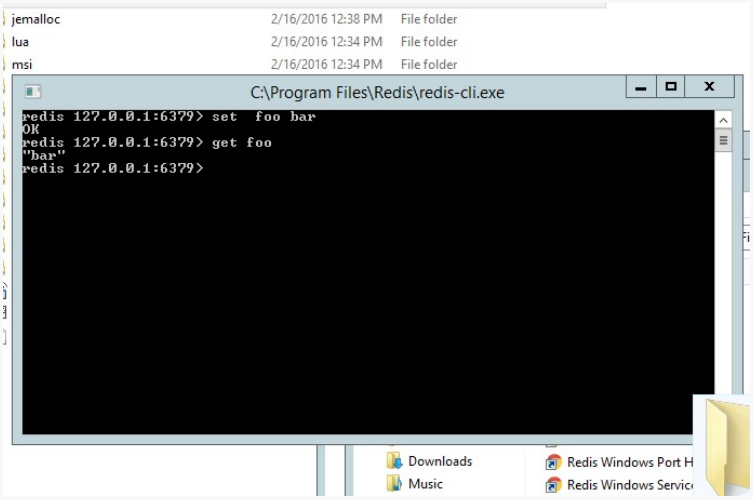
1.Download and install Redis Windows Port (on any server you want to install Redis on) using this link:

2. Go to Communifire admin section, select *System > System Properties,* and then select *Redis Config* category from the left side pane. You will see SignalR-Redis specific settings, you

need to edit and put the right values (the below example uses sample values):

> "UseRedisForSignalR" value="true"
> "RedisServerNameForSignalR" value="10.10.1.11"
> "RedisServerPassword" value=xxxx
> "RedisServerPortForSignalR" value="6379"

3. To confirm Redis is working, run the following command (from the Redis install directly, which is usually C:\Program Files\Redis directory) as shown in this screen shot below:



After these changes, SignalR Redis backplane will be used to forward SignalR messages to other servers in the web farm.

Elasticsearch in a load balanced environment

In a web farm environment, you can have any node as a search server where Elasticsearch will be installed or you can have a separate search server. Please refer this wiki for details on how to setup & install Elasticsearch.

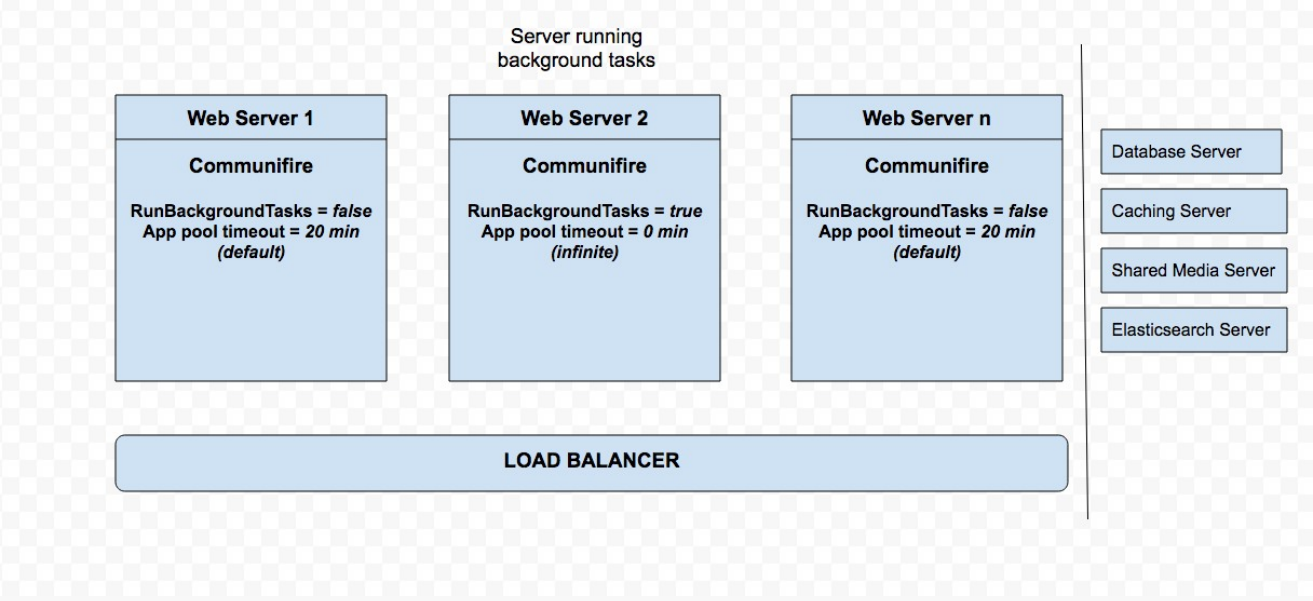
Background tasks synchronization in a web farm

Communifire uses a lot of web-based background tasks, for example, sending queued emails, notifications/reminders and much more. We deliberately did not use a Windows Service solution but created recurring background tasks which run within the ASP.NET application pool. **But on a web farm, there should be only one web server which should be running these tasks**. So when configuring Communifire on a web farm, you need to specify the "RunBackgroundJobs" property to true in CFAppSettings-Override.config file for only one server which will handle these recurring tasks:

> <!-- Communifire uses background tasks/jobs which fire at regular intervals to perform different works (like sending daily digest emails). On a single web server environment, this value should be true. But on a Web Farm, set this value to true ONLY FOR ONE WEB SERVER in that farm which means that only that web server is designated to perform all the recurring background tasks. For other web servers this value should be FALSE so that only one web server can perform these background tasks.
>
> NOTE: For the web server which is designated to run the background tasks, its application pool should never time out (Timeout ==0).-->
>
> <addkey="RunBackgroundJobs" value="true"/>

For all other web servers, this property should be set to "false". Also make sure that the server where this property is true, its application pool timeout should be set to 0 in IIS so that the application pool never expires. For other web servers it can be set to the default 20 minutes, though we recommend that all **Communifire application pools should never time out** (on all nodes in the farm). **Make sure to disable application pool recycle on all nodes on the farm.**

Here is a sample deployment diagram to configure Communifire on a load-balanced web farm:



tags : configuration, site-administrator, web-farm