2023

Web farm Configuration Guide

DNN EVOQ 9.1.1

DNN SUPPORT SERVICES

DNN SOFTWARE | http://www.dnnsoftware.com

ABSTRACT

DNN Evoq Content is used for a broad range of websites; from single server installations to large enterprise class installs running in web farms. Because of the issues that are unique to web farms, the standard installation guidance is not sufficient. This document will outline the recommended and supported configuration for setting up your DNN Evoq Content installation in a web farm environment.

NOTICE

Information in this document, including URL and other Internet Web site references, is subject to change without notice. The entire risk of the use or the results of the use of this document remains with the user.

The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, places, or events is intended or should be inferred.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of DNN Corp. DNN Corp. may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from DNN Corporation, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

TABLE OF CONTENTS

1. INTRODUCTION

The purpose of this document is to articulate the requirements for running Evoq Content 7 and higher in a specific web farm configuration. Conventions used in this document will assume some familiarity with IIS and Windows administrative functions. External resources will be noted throughout this document to provide addition information.

In this document we will assume usage of the following versions of relevant required technologies:

- Microsoft Windows Server 2012r2
- .NET 4.0
- Microsoft SQL Server 2012

This does not mean that these versions are currently the minimum required versions, however, we do recommend when starting with a new project to utilize current versions of software. Please see Appendix 1 for an overview of minimum required platform components.

- The configuration will utilize the following components:
- All machines are members of an Active Directory Domain.
- Load Balancer managing traffic between 2 or more Web Servers.
- Web Servers running Windows 2012 Server, .NET 4.0 and IIS 7.5
- DB Server running SQL Server 2012 (web farm should have no impact on this configuration). For fail safe setups, use a SQL Server Failover Cluster
- File Server running Windows 2012 R2 Server to serve application files and content. For fail safe setups, use a Windows Failover Cluster

ASSUMPTIONS

An instance of DNN has already been configured and installed in a NON Load balanced configuration and is ready to be moved into the farm. The point of this assumption is to assure proper initial configuration of SQL and DNN and to avoid the possibility of corrupting the install process while the web- farm environment is being properly configured.

We also assume that your load balancer is correctly configured to direct traffic via an IP shared by all Web Servers. Further, that the individual loopback IP's of the Web Servers are also accessible, in order to enable reaching the Web Servers individually. SSL Offloading to the Load Balancer is supported from DNN 6.2.2

Also, we assume that all the servers are part of the same domain. Although this greatly simplifies configuration, there are use cases where this is not possible. In that case, local accounts that are used on each server, need to have the same username/password combination, and clear text authentication needs to be used to authenticate one server to the other.

Throughout this document we assume that the reader is familiar with the standard DNN configuration requirements. We will focus on those settings and configuration items where the web farm installation requires an alternative configuration.

MORE INFORMATION

1. SQL Server Failover Cluster Installation

   http://msdn.microsoft.com/en-us/library/hh231721

2. How to Configure a Clustered Storage Space in Windows Server 2012

   http://blogs.msdn.com/b/clustering/archive/2012/06/02/10314262.aspx

2. SCALABILITY

Before setting up a web farm, it is important to think about what type of scaling is needed for your installation. DNN supports different types of scaling, depending on your needs.

STANDARD

This web farm setup is best suited for an application with 1 or few distinct sites, which receive a high volume of traffic. Each web server serves all configured web sites. The number of web heads depicted is just an example.

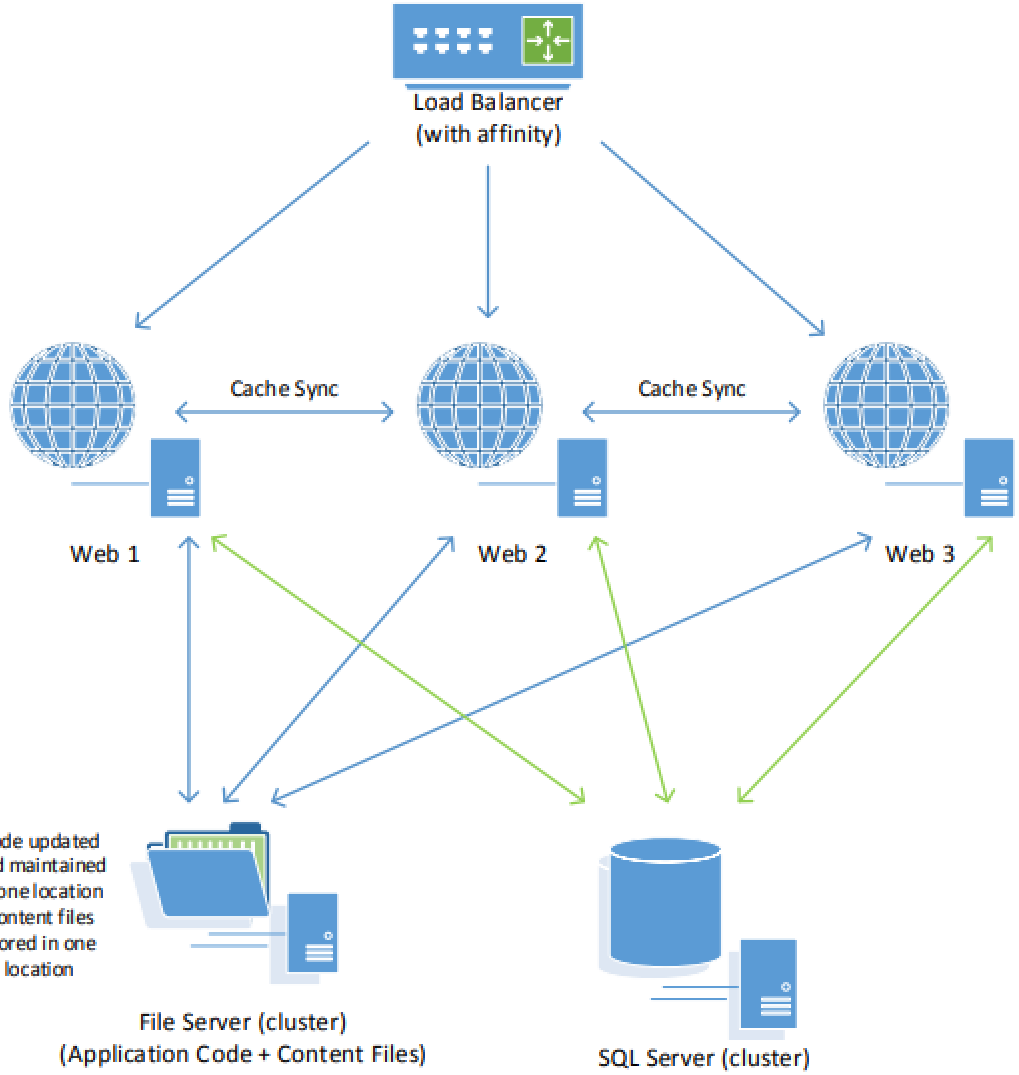

Load Balancer
(with affinity)
Cache Sync
Cache Sync
Web 1
Web 2
Web 3
Code updated and maintained in one location Content files stored in one location
File Server (cluster)
(Application Code + Content Files)
SQL Server (cluster)

Figure 1 – Standard Web Farm Setup

SEGMENTED – MANY SITES

This setup, which technically is NOT a web farm, is best suited for a situation with many configured web sites, where each web site receives a limited amount of traffic. Each server is set up to serve X number of sites in a domain block, there is no need for synchronization between servers.

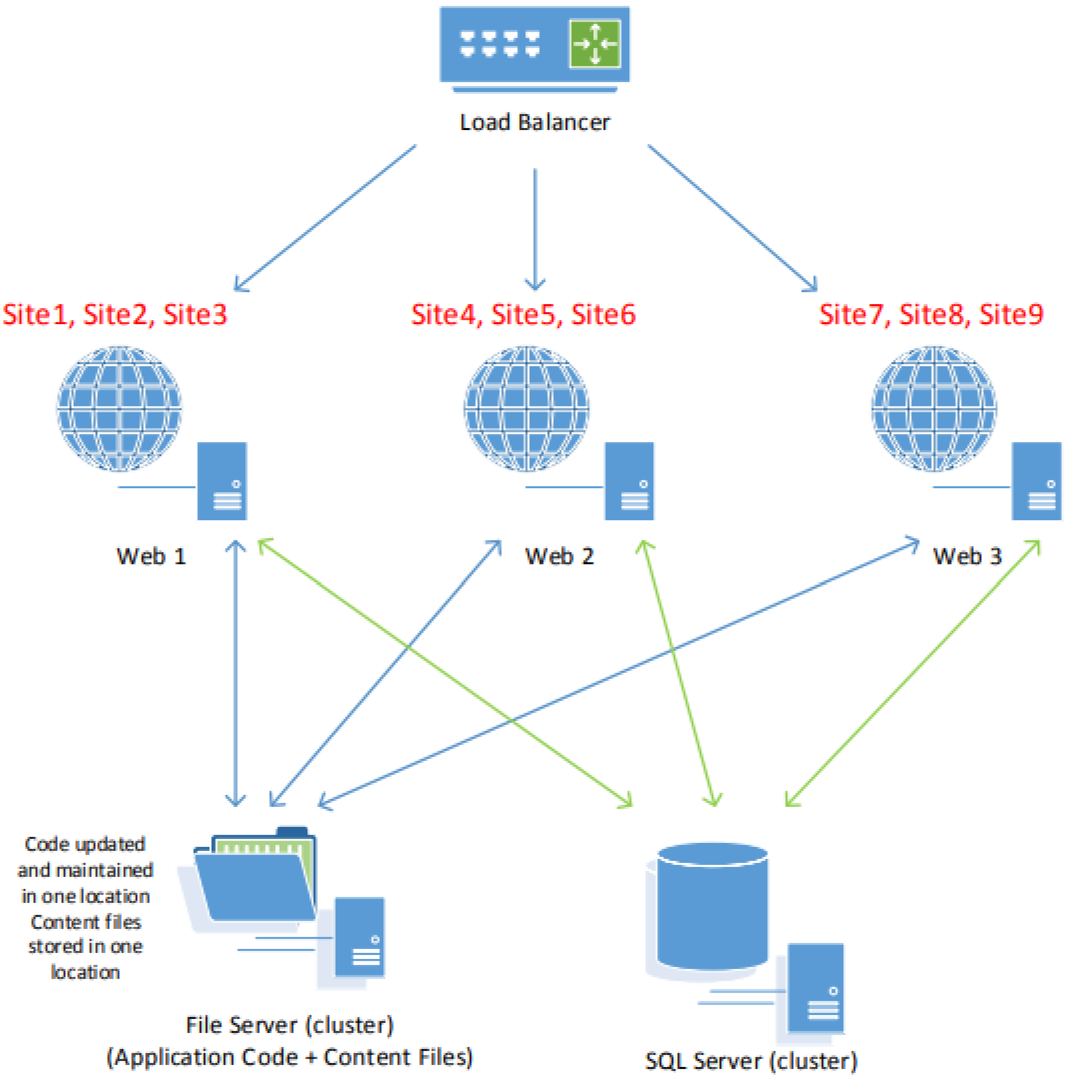


Figure 2 – Segmented Web Farm Setup

3. FILE SYSTEM

The file server is a very important part of the web farm. In this chapter the different configurations for fileserver will be discussed.

There are two options to host the application files and content in a web farm environment, hosting the files locally or in a UNC Share. Both options have pros and so please use your discretion to make a choice that is more fitting to your requirements.

UNC SHARE

The easiest file system setup uses a file share (UNC share). Using a file share ensures that files used by the DNN application are always the same (and in sync). The file server requires two sets of permissions to be configured:

1. Network Share Permissions (SMB Permission)
2. File/Folder permissions (NTFS Permission)

Both sets of permissions will be applied to the account used by the Application Pool on Web Servers.

For more information about how to set up a shared folder, please see this third party resource:

- *Setup Shared Folder in Windows Server 2012*:

  http://www.mustbegeek.com/setupshared-folder-in-windows-server-2012

Network Share Permissions (SMB Permission)

Full access permissions should be granted to the user that will be used by the web servers to access the share (this will be the identity that runs application pool on Web Server).

Folder/File Permissions (NTFS Permission)

The next step is to set the NTFS file permissions for the accessing identity. This step is required as DNN needs to be able to create directories and files.

To set the correct File permissions use Windows File Explorer to locate the physical folder where the DNN application files exist. Right-click on the folder, select Properties and choose the Security Tab. Select the user account (identity that runs application pool – in case of UNC share this is normally a domain account) and assign the necessary permissions

– *Full Control or Modify, make sure permissions have been applied to all files and folders in the website root folder.*

This third party resource shows more information about NTFS permissions:

Setting basic NTFS permissions in Windows Server 2012:

http://www.techrepublic.com/blog/data-center/setting-basic-ntfs-permissions-inwindows-server-2012

SMB 3.0

Windows Server 2012 introduces new server message block (SMB) file server features. The main benefit of this newer version (as opposed to a much older version that was default on Windows 2003) is that it can handle many more file handles, in a much more efficient way. To take advantage of these new features, the SMB client (Web Server) and SMB server (UNC Share Server) must support SMB 3.0

For more information about SMB, please see these Microsoft resources:

1. Server Message Block overview:

   http://technet.microsoft.com/enus/library/hh831795.aspx

2. New SMB 3.0 features in the Windows Server 2012 file server:

   http://support.microsoft.com/kb/2709568/en-us

HIGH AVAILABILITY (FAILOVER CLUSTERING)

The perception is that when a UNC share is used, there will be a single point of failure in the web farm setup. This is true if there is just one server used to serve the file share. However, there options supported by the Windows Server platform to ensure High Availability. It is beyond the scope of this document to go very deep into how this work, however, these are some excellent resources to gain more background on this topic:

- Failover Clustering Overview (Windows Server 2012):

  http://technet.microsoft.com/en-us/library/hh831349.aspx

- Scale-Out File Server for Application Data Overview (Windows Server 2012):

  http://technet.microsoft.com/en-us/library/hh831349.aspx

- SMB Transparent Failover – making file shares continuously available:

  http://blogs.technet.com/b/clausjor/archive/2012/06/07/smb-transparentfailover-making-file-shares-continuously-available.aspx

- Windows Server “8” – Taking Server Application Storage to Windows File Shares:

  http://blogs.technet.com/b/windowsserver/archive/2012/03/15/windowsserver-8-taking-server-application-storage-to-windows-file-shares.aspx

- Designing systems for continuous availability - multi-node with remote file storage:

  http://channel9.msdn.com/Events/BUILD/BUILD2011/SAC-444T

- Clustering and High-Availability: Installing the Failover Cluster Feature and Tools in Windows Server 2012:

  http://blogs.msdn.com/b/clustering/archive/2012/04/06/10291601.aspx

- Clustering and High-Availability: Creating a Windows Server 2012 Failover Cluster:

  http://blogs.msdn.com/b/clustering/archive/2012/05/01/10299698.aspx

FILE REPLICATION

As an alternative, you can choose to setup the file system for the web farm in separate locations (usually local to the web server), and replicate files in some way or form across the different locations. File synchronization can be set up separately using one of the options below:

- DFS Replication: http://technet.microsoft.com/en-us/library/dn270370.aspx
- Robocopy: http://technet.microsoft.com/en-us/library/cc733145.aspx
- Microsoft RichCopy: http://www.symantec.com/connect/articles/readyhowreplicate-file-shares-using-microsoft-richcopy

Please note that Search may be impacted by replication due to file locking during replication process. File ***\App_Data\Search\write.lock*** should always be excluded from replication.

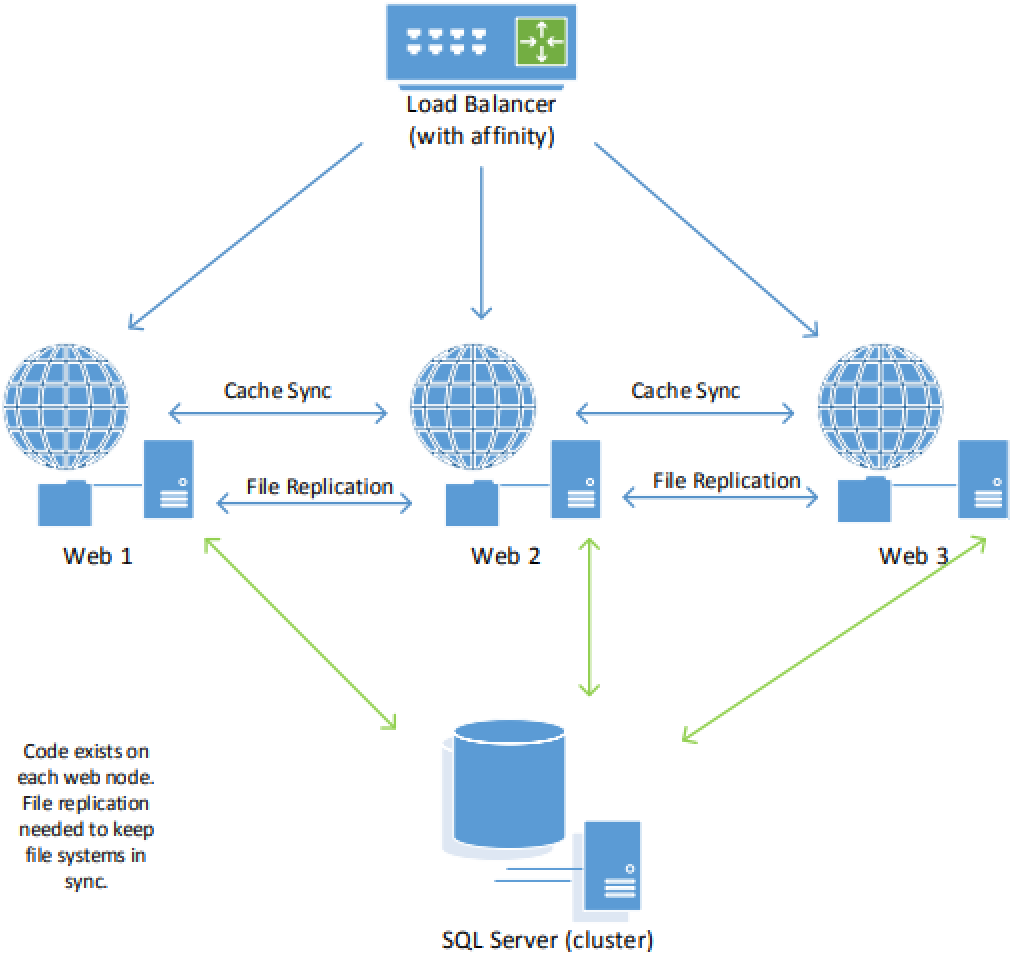


Figure 3 – File Replication

4. WEB SERVER

Every web farm consists of multiple Web Servers which are responsible for serving the content. At this point we should have our File server ready and can move to Web Server configuration.

PERMISSIONS AND CODE ACCESS SECURITY (UNC ONLY)

When using a UNC file share, some special permissions and security settings need to be applied.

Permissions (Domain Account)

We must give “domain\username” sufficient access to be used as an application pool security principal. There are a few places where this must be addressed. Proper access to the web root is required but we will defer this topic until we address the file server and remote file share where the web root will reside.

Execute the built in aspnet_regiis command with the –ga option by opening a command window to the .NET Framework 4.0 folder and running the command with the following syntax:

```
aspnet_regiis –ga “domain\username”
```

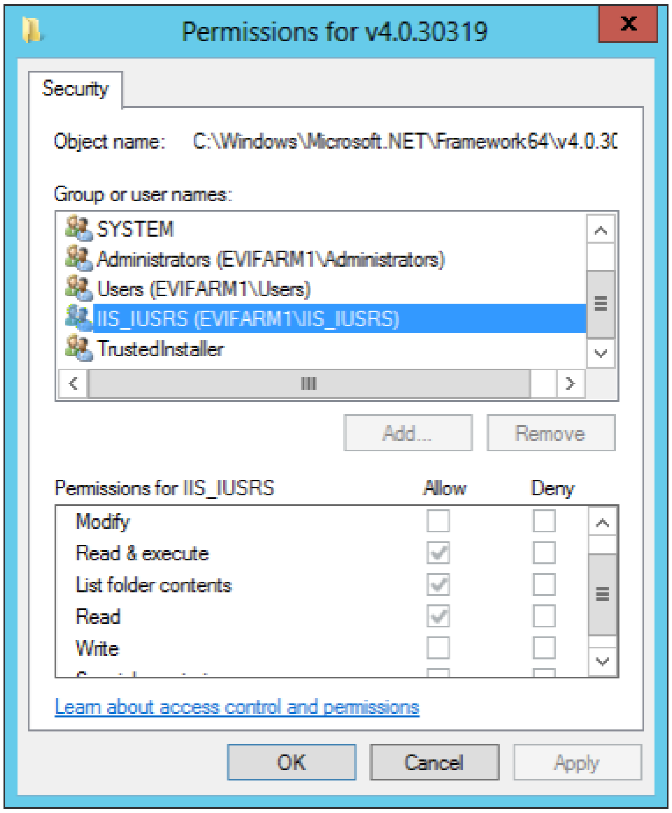
Expected output:

```
Start granting domain\username access to the IIS metabase and other directories used by ASP.NET.

Finished granting domain\username access to the IIS metabase and other directories used by ASP.NET.
```

The key permissions to verify are that access is granted to the following system directories:

- %windir%\ Microsoft.NET\Framework64\v4.0.30319
- %windir%\Temp



Permissions for v4.0.30319

Security

Object name: C:\Windows\Microsoft.NET\Framework64\v4.0.3(

Group or user names:

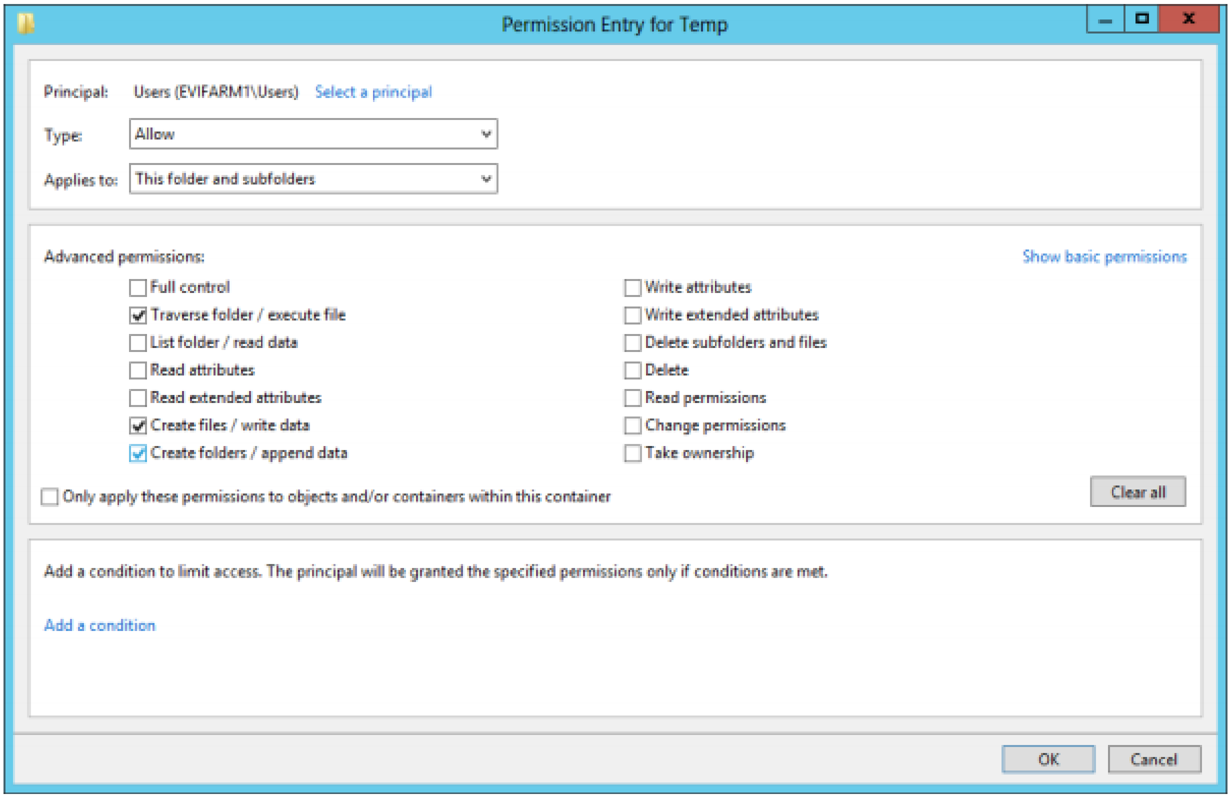
- SYSTEM
- Administrators (EVIFARM1\Administrators)
- Users (EVIFARM1\Users)
- IIS_IUSRS (EVIFARM1\IIS_IUSRS)
- TrustedInstaller

Permissions for IIS_IUSRS	Allow	Deny
Modify	□	□
Read & execute	☑	□
List folder contents	☑	□
Read	☑	□
Write	□	□

Figure 4 - Framework folder permissions

This illustrates permissions applied to the ASP.NET Framework folder. The following KB article summarizes key permissions used by ASP.NET.

ASP.NET Required Access Control Lists (ACLs): http://msdn.microsoft.com/enus/library/kwzs111e.aspx



Permission Entry for Temp

Principal: Users (EVIFARM1\Users) Select a principal

Type: Allow

Applies to: This folder and subfolders

Advanced permissions:

Show basic permissions

□ Full control
☑ Traverse folder / execute file
□ List folder / read data
□ Read attributes
□ Read extended attributes
☑ Create files / write data
☑ Create folders / append data

□ Write attributes
□ Write extended attributes
□ Delete subfolders and files
□ Delete
□ Read permissions
□ Change permissions
□ Take ownership

□ Only apply these permissions to objects and/or containers within this container

Add a condition to limit access. The principal will be granted the specified permissions only if conditions are met.

Add a condition

Figure 5 - Windows Temp folder permissions

Code Access Security

ASP.NET, by default, does not "trust" content hosted remotely. So depending upon where you are in setup, you may be able to "see" the content on your remote share, but ASP.NET (at runtime) will terminate execution because it content location is not "trustworthy". To rectify this condition, we need to set appropriate security policy at the machine level.

This can be accomplished using either the. The appropriate command using caspol.exe would be:

```
caspol.exe -m -ag 1 -url "\\<ServerName>\<FolderName>\*" FullTrust -exclusive on
```

Make sure you are running the 64bit version of caspol

64Bit: %windir%\Microsoft.NET\Framework64\v4.0.30319

Once your Runtime Security Policies have been established restart your IIS services for them to take effect.

More information about the CasPol tool can be found here:

- Caspol.exe (Code Access Security Policy Tool):

  http://msdn.microsoft.com/enus/library/cb6t8dtz(v=vs.110).aspx

WEBSITE CONFIGURATION

As File server and Web Server have been configured already last step is to configure the web site.

Let's look at configuration of the web site first. Bear in mind that the web site and application pool will be configured identically on each Web Server in the farm with the exception of any scheduled app pool restarts. It is recommended that scheduled app pool restarts (if applicable) be staggered within the farm.

IIS Shared Configuration allows you to use a shared location (UNC path) to store the IIS Host configuration file. This makes it extremely easy to have the same configuration on every IIS server. See here for more info on how to set up Shared Configuration:

http://www.iis.net/learn/manage/managing-your-configuration-settings/sharedconfiguration_264

Although it is possible to use IIS Shared Configuration, one downside to using that is that all application pools in the farm will be configured identically, which would mean that they would all recycle at the same time, which may cause unexpected performance issues with your site.

Application Pool

To have higher isolation level we recommend to run the DNN website in a separate application pool. By default IIS creates new application pool for each website, but there is a possibility to change that setting, therefore insure DNN website is running in dedicated application pool (other sites or applications shouldn't use it).

Physical Path

We must configure the website itself to reference the UNC Share (in case if files are hosted locally specify local path on the server). Please note that if the username you use to administer IIS on the webserver does not have permission to access the share, you will not see the files in the web root (although the IIS application will).

Trust Level

Trust level should be set to "Full" as some of the functionality, like search, requires Full Trust level.

Authentication (Anonymous Users)

This step is required only for UNC Share. We need to ensure that site visitors will have access to the needed files. Enter the user name and password for the domain account that has read access over the UNC folder. Alternatively, you can create a local user with the same user name and password as a user account on the file server which will be used for file/share permissions.

The following article explains more about the Anonymous Authentication Identity

- Configure the Anonymous Authentication Identity (IIS 7.0):

  http://technet.microsoft.com/en-us/library/cc770966%28WS.10%29.aspx

Host Headers (Bindings)

Each web server must be identified using *unique URL that can be accessed directly without going through the load balancer*. For this reason there must be host headers (bindings) assigned to each site (should be also added as site aliases in Admin->Site Settings). Usually internal an IP address is used for this purpose. This ensures that the website is only accessible from the internal network, and not from the internet.

APPLICATION POOL CONFIGURATION

.NET Version

Evoq Content 9.x only runs when .NET 4 is selected. If .NET 4.5 is installed, that will be used instead.

Managed Pipeline Mode

Please use "Integrated". DNN does not support Classic Mode

Application Pool Identity (UNC Share Only)

In IIS, every application pool has its own identity. By default, this is the automatic "Application Pool Identity", which is an automatic local server account. In order to allow the application to access network resources, the Identity of the Application Pool should be configured with a Domain Account that can access each of the machines in your web farm.

This name replaces the common usage of the "Application Pool Identity" or "Network Service" user and should be granted a set of specific permissions in a variety of locations in the configuration.

Load User Profile

Should be set to *True*

Idle Time-out Action (Available in IIS 8.5)

Set to Suspend. Below article explains more about this feature:

Welcome OS Family 4: Windows Server 2012 R2, IIS 8.5 and .NET Framework 4.5.1:

http://geeks.ms/blogs/davidjrh/archive/2013/10/20/welcome-os-family-4-windows-server-2012-r2-iis-8-5-and-net-framework-4-5-1.aspx

Maximum Worker Process

Please ensure it is set to 1, as we do not support Web gardens.

Shutdown Limit Time

Depending on the web site functionality, value can be increased from default 90 seconds. This value should be enough for all functionality to shut down gracefully.

Startup Limit Time

Depending on the web site functionality, value can be increased from default 90 seconds. This value should be enough for all functionality to shut down gracefully.

Recycling

Configuring scheduled recycling depends on your requirements. It is recommended that scheduled app pool restarts (if applicable) be staggered within the farm.

.NET FILE CHANGE NOTIFICATION

By default, ASP.NET sets up a file change notification for each directory in the application. You can force ASP.NET to setup a single file change notification object on just the web application root folder.

In .NET 4, this needs to be set in the registry of all web servers in the farm. Add the following registry key (you will need to create the key if it doesn't exist):

```
HKLM\Software\Microsoft\ASP.NET\FCNMode
```

FCNMode should be set to a DWORD with the value of 2.

In .NET 4.5, this can be set in web.config, which allows for more granular control, and also ensures that this setting will be carried over if you move the installation other servers. The **httpRuntime** element supports a new attribute **fcnMode**, which should be set to "Single":

```
<httpRuntime fcnMode="Single" />
```

The **httpruntime** element for a default web farm installation would then look like this:

```
<httpRuntime shutdownTimeout="120" executionTimeout="900"
useFullyQualifiedRedirectUrl="true" maxRequestLength="12288"
requestLengthDiskThreshold="12288" requestPathInvalidCharacters="&lt;,&gt;,*,%,:,\,?"
requestValidationMode="2.0" fcnMode="Single" />
```

See here for more information about fcnMode:

http://support.microsoft.com/kb/911272

http://msdn.microsoft.com/en-us/library/system.web.configuration.fcnmode.aspx

http://msdn.microsoft.com/enus/library/system.web.configuration.httpruntimesection.fcnmode.aspx

At this point your Web Server is completely configured. You will need to perform the same steps for each server in the web farm.

5. DATABASE

INTEGRATED SECURITY

Integrated Security allows the DNN application to connect to the database using the application pool identity credentials (domain account in most cases). This requires the application pool identity to be added as a login with DBO permissions for the application database.

Example connection string:

```
"Data Source=DatabaseServer; Initial Catalog=DNN; Integrated Security=True;"
```

SQL AUTHENTICATION

When using SQL Authentication application connects to the DB using existing SQL account. This account should have DBO permissions over the application database.

Example connection string:

```
"Data Source=DatabaseServer; Initial Catalog=DNN; User ID=dnnuser; Password=1234567;"
```

HIGH AVAILABILITY (FAILOVER CLUSTER / MIRRORING)

Microsoft's recommendation for providing data protection for your SQL Server environment are as follows:

1. For data protection through a third-party shared disk solution (a SAN), we recommend that you use AlwaysOn Failover Cluster Instances.
2. For data protection through SQL Server, we recommend that you use AlwaysOn Availability Groups.

Below are some extra resources with extra information on this matter:

High Availability Solutions (SQL Server):

http://technet.microsoft.com/enus/library/ms190202.aspx

AlwaysOn Architecture Guide:

http://download.microsoft.com/download/D/2/0/D20E1C5F-72EA-4505-9F26-FEF9550EFD44/Building_a_HA_and_DR_Solution_using_AlwaysON_SQL_FCIs_and_AGs v1.docx

6. APPLICATION FEATURES

By now you should have more than one DNN Web Server setup and pointed to the same database and possibly using the same physical path under IIS (UNC Share). The only functionality missing from this setup to be a complete Web farm is cache synchronization.

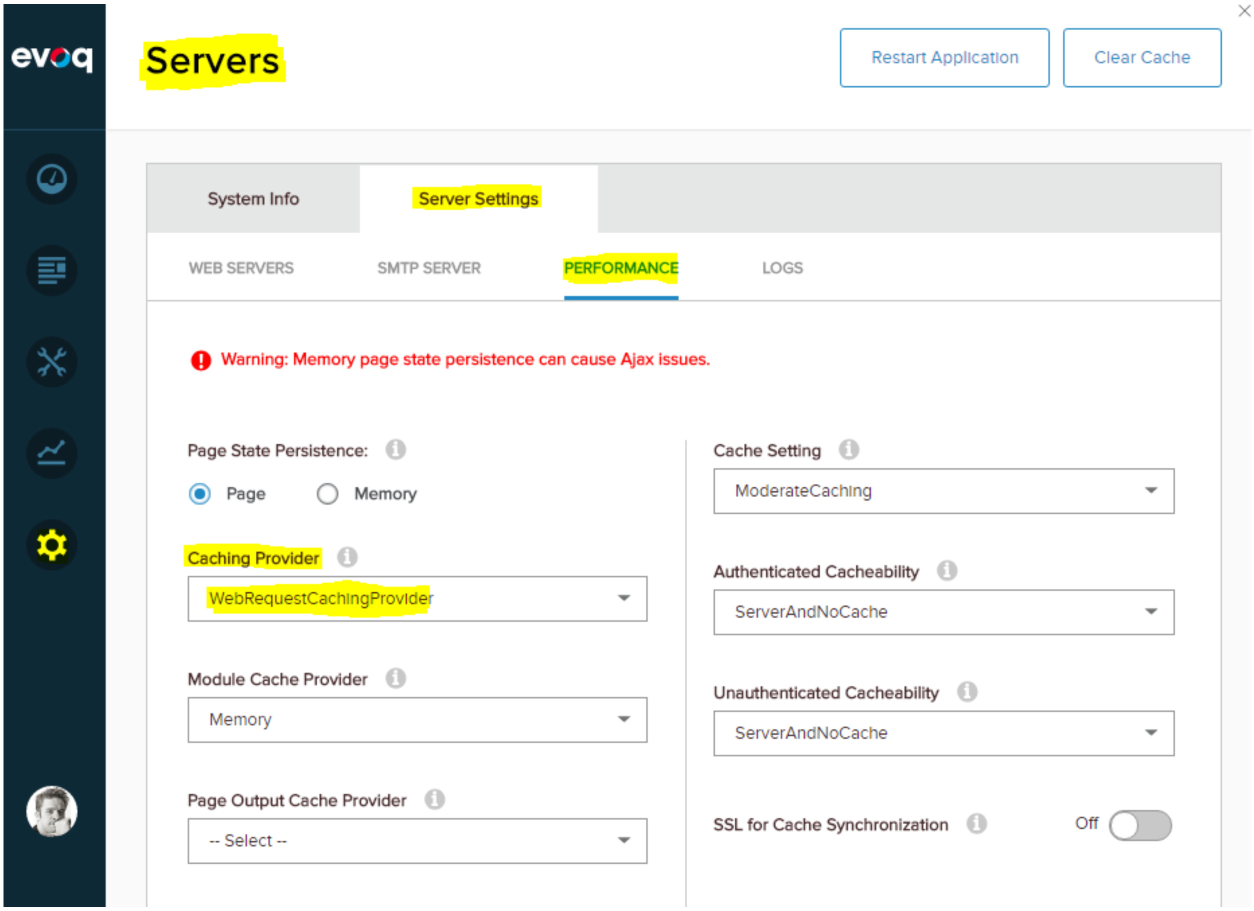
CACHING PROVIDER OVERVIEW

Currently the recommended caching provider for web farm configuration is **WebRequestCachingProvider** that uses web requests to keep the cached items synchronized across Web Servers. This provider delivers higher levels of efficiency since it doesn't rely on database or the file system.

Make sure the Caching Provider is set to **WebRequestCachingProvider**.

Navigate to:

Personabar > Settings > Servers > Server Settings > Performance > Caching Provider



Warning: Memory page state persistence can cause Ajax issues.

Page State Persistence: ● Page ○ Memory

Caching Provider: WebRequestCachingProvider

Module Cache Provider: Memory

Page Output Cache Provider: -- Select --

Cache Setting: ModerateCaching

Authenticated Cacheability: ServerAndNoCache

Unauthenticated Cacheability: ServerAndNoCache

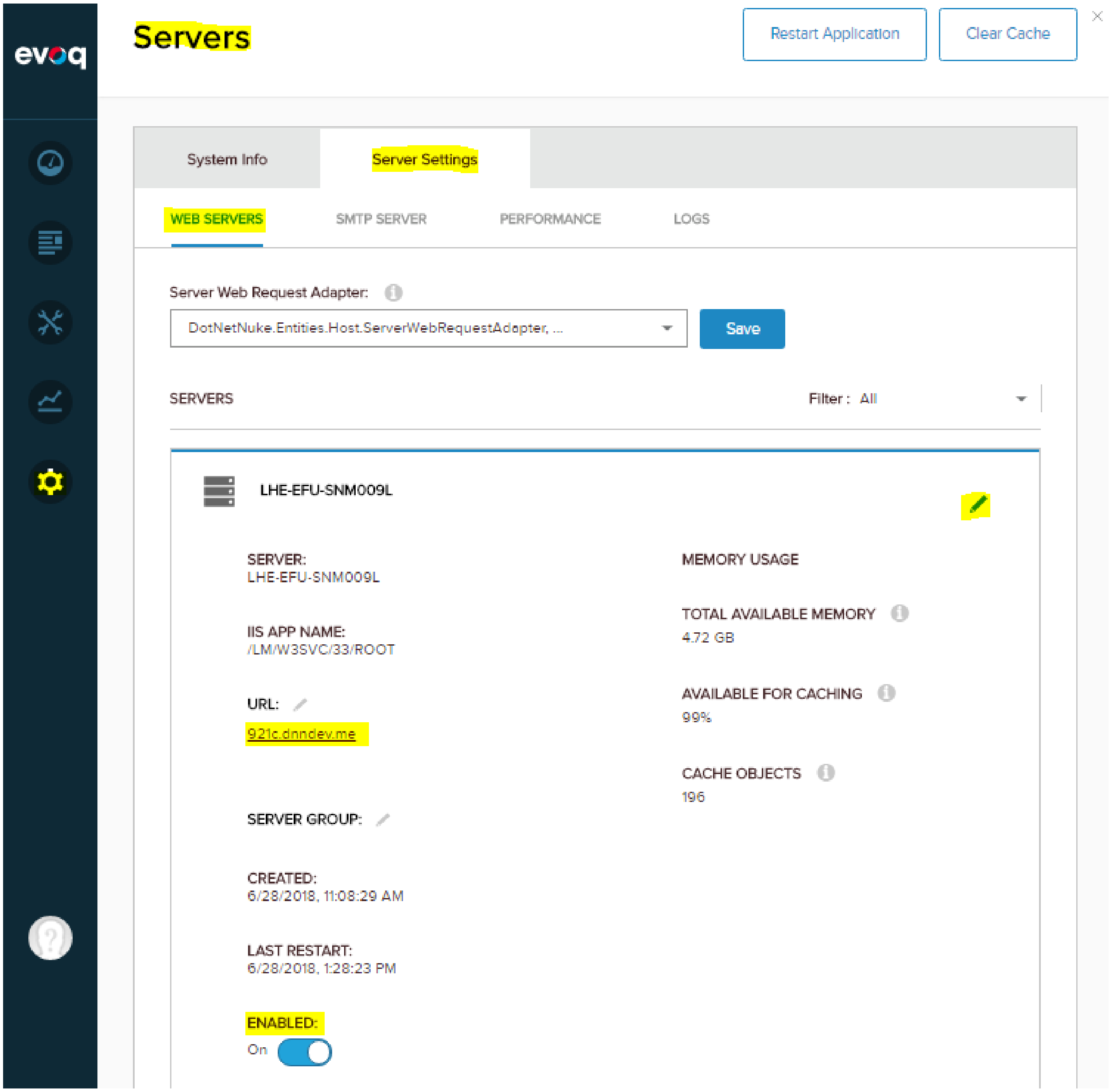
SSL for Cache Synchronization: Off

WEB SERVER CONFIGURATION

To Configure Web Servers that will participate in the web farm login as Host and navigate to

Personabar > Settings > Servers > Server Settings > Web Severs

You will also notice that none of the web servers are enabled by default. Click the edit icon and check the enabled check box and save the changes. That will cause the servers to join the web farm.



Server Web Request Adapter: DotNetNuke.Entities.Host.ServerWebRequestAdapter, ...

Filter : All

LHE-EFU-SNM009L

SERVER:
LHE-EFU-SNM009L

IIS APP NAME:
/LM/W3SVC/33/ROOT

URL:
921c.dnndev.me

SERVER GROUP:

CREATED:
6/28/2018, 11:08:29 AM

LAST RESTART:
6/28/2018, 1:28:23 PM

ENABLED:
On

MEMORY USAGE

TOTAL AVAILABLE MEMORY
4.72 GB

AVAILABLE FOR CACHING
99%

CACHE OBJECTS
196

WebRequestCachingProvider will work with the most common web farm setup, which utilizes a load balancer that sends requests to machines via a unique address.

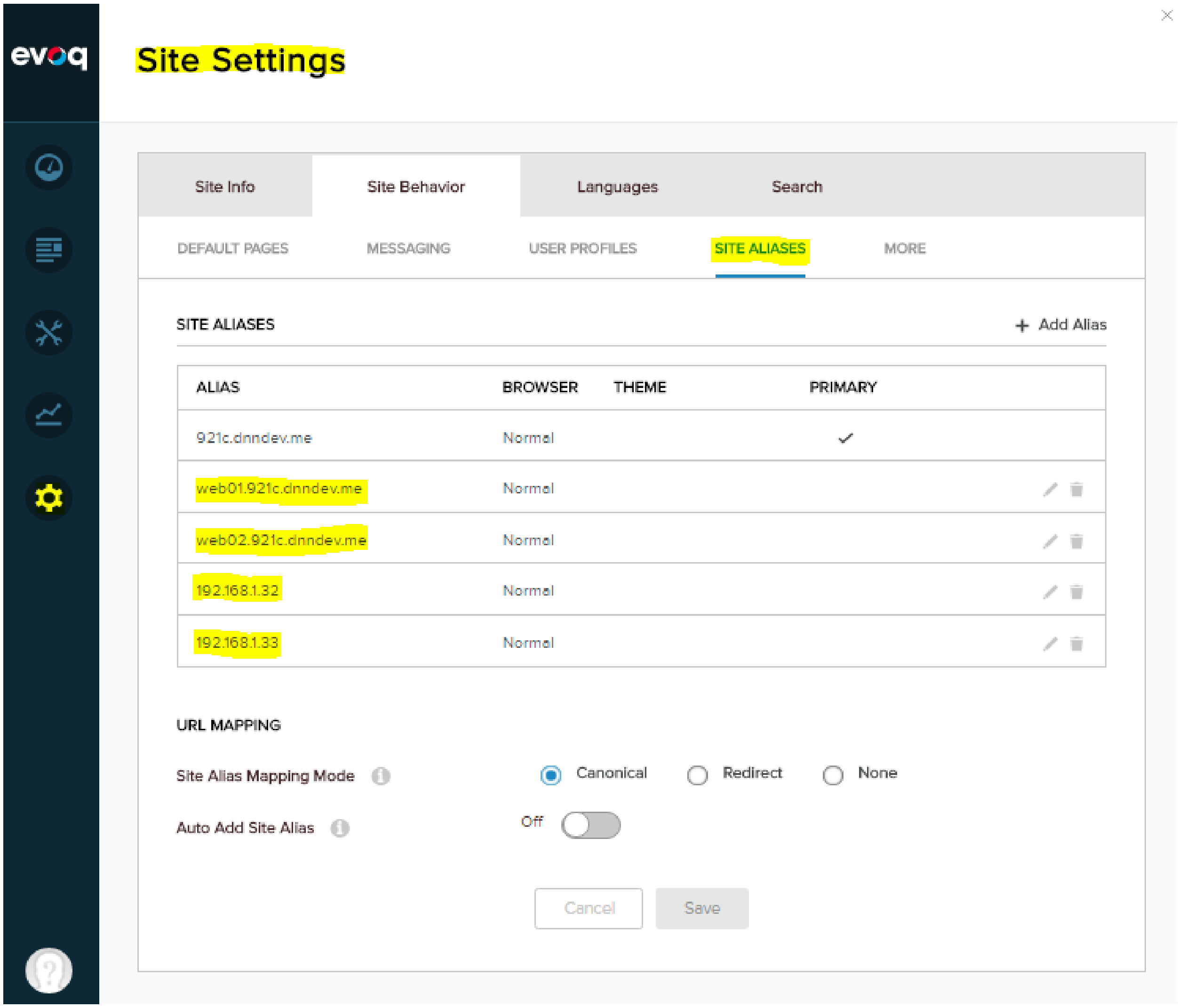
The WebRequestCachingProvider is then able to use these unique addresses to maintain cache integrity. However some sites choose to use virtualized addresses - either the fully qualified domain name or a fixed IP, and the **WebRequestCachingProvider** can't do its job as it has no unique addresses. To resolve this issues there are a few additional steps to perform.

The first step is to assign a unique URL to each of the servers that are participating in the web farm. As you can see in the previous figure both servers share the same URL which could cause problems with cache sync.

To resolve this we need to assign the unique URLs of each machine to the list of Portal Aliases.

Personabar > Settings > Site Settings > Site Behavior > Site Aliases

You'll need to add unique portal alias's for both machines - typically the machines internal IP or internal name are the best options. In this example we use the internal machine IP.



SITE ALIASES

ALIAS	BROWSER	THEME	PRIMARY
921c.dnndev.me	Normal		✓
web01.921c.dnndev.me	Normal		
web02.921c.dnndev.me	Normal		
192.168.1.32	Normal		
192.168.1.33	Normal		

URL MAPPING

Site Alias Mapping Mode ● Canonical ○ Redirect ○ None

Auto Add Site Alias Off

Requests come to **921c.dnndev.me** but are sent to unique machine URL's such as If we use IP to identify each machine e.g. **192.168.1.32** or **192.168.1.33** etc. Another approach would be to use unique FQDN, e.g.**web01.921c.dnndev.me**, **web02.921c.dnndev.me** etc. domain name should resolve to correct IP/Server

Repeat this for every server that needs to be in the web farm.

Now you should have a fully functional Web farm setup with synchronized caching.

SSL OFFLOADING

When using SSL, it is sometimes preferable terminate the SSL tunnel at the load balancer (offload SSL to load balancer). DNN Evoq supports this, with the option to add a specific HTTP header, depending on the load balancer.

Some examples of the headers include:

Citrix supports custom headers and recommends using SSL_REQUEST.

https://www.cisco.com/c/dam/en/us/td/docs/switches/datacenter/nexus1000/solutions/citrix-ns/citrix_release_notes/netscaler_10_5_63_8_release_notes.pdf

Weblogic uses a fixed header of WL-Proxy-SSL.

http://fusionsecurity.blogspot.nl/2011/04/ssl-offloading-and-weblogic-server.html

BigIP/F5 supports custom headers via their iRule rewrite function. See

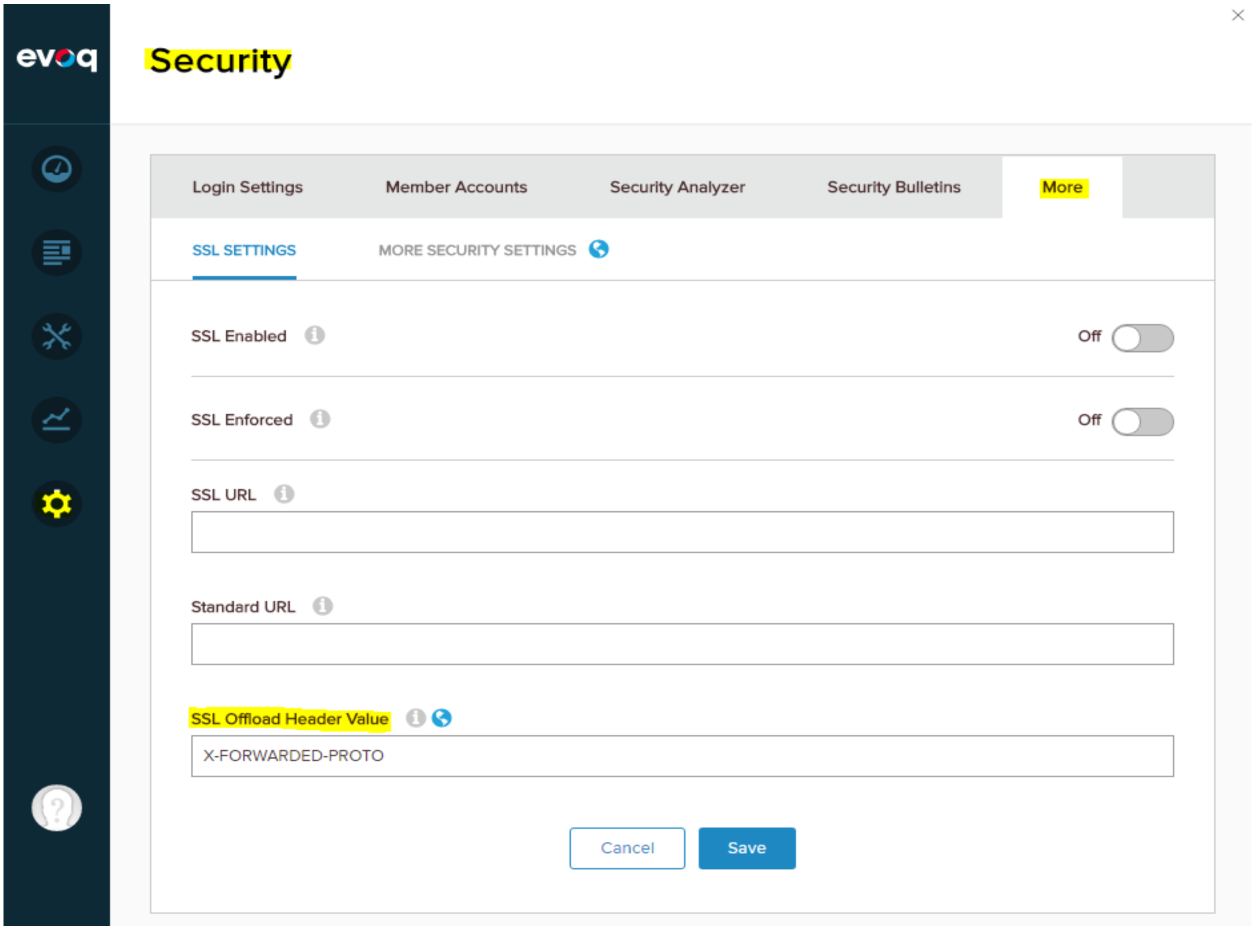
http://www.lullabot.com/blog/article/setting-ssl-offloading-termination-f5-big-ipload-balancer

Prerequisites

SSL must be enabled for the required sites. See "Setting SSL Settings for a Single Site"

Navigate to **Personabar > Settings > Security > More > SSL Settings > SSL Offload Header Value**

In the Value text box, enter the value. E.g. **X-FORWARDED-PROTO** and **Save**



SSL Enabled: Off

SSL Enforced: Off

SSL URL

Standard URL

SSL Offload Header Value: X-FORWARDED-PROTO

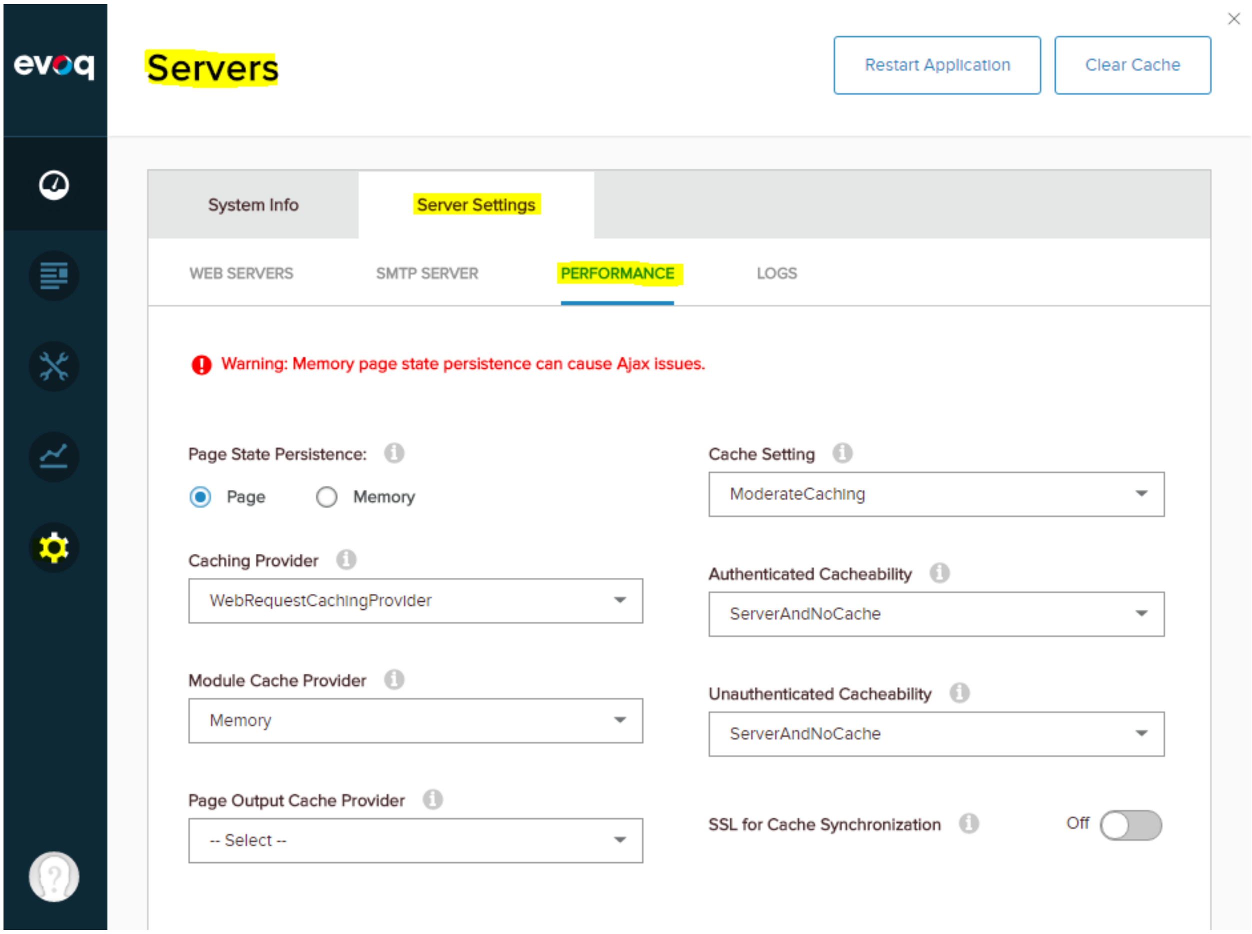
Now when a request arrives at the load balancer, if it has SSL offloading enabled it will pass the request onto the web server with the request rewritten from a secure to insecure request (e.g. https://mysite.com/default.aspx to http://mysite.com/default.aspx). This will be the request that DNN processes. Normally DNN would then determine that the request is for a "secure" page and rewrite the path back to https://mysite.com/default.aspx, but the existence of the header ensures that DNN knows it should instead serve the page up via HTTP.

The results will then be passed back to the SSL-Offloading load balancer which will return the page to the user as those an SSL request was made (as is the case as the SSL certificate was verified by the load balancer which processes SSL requests more efficiently than the individual web server(s) would - as well as simplifying management by ensuring only the load balancer needs the SSL certificate installed rather than each web server)

PAGE AND MODULE CACHING

When server resources permit (enough ram available in the system), it is recommended to switch to the Memory Caching Provider for both Page and Module caching. Caching providers are set per page and per module, and use a default setting as set in

Personabar > Settings > Servers > Server Settings > Performance



Warning: Memory page state persistence can cause Ajax issues.

Page State Persistence: ● Page ○ Memory

Caching Provider: WebRequestCachingProvider

Module Cache Provider: Memory

Page Output Cache Provider: -- Select --

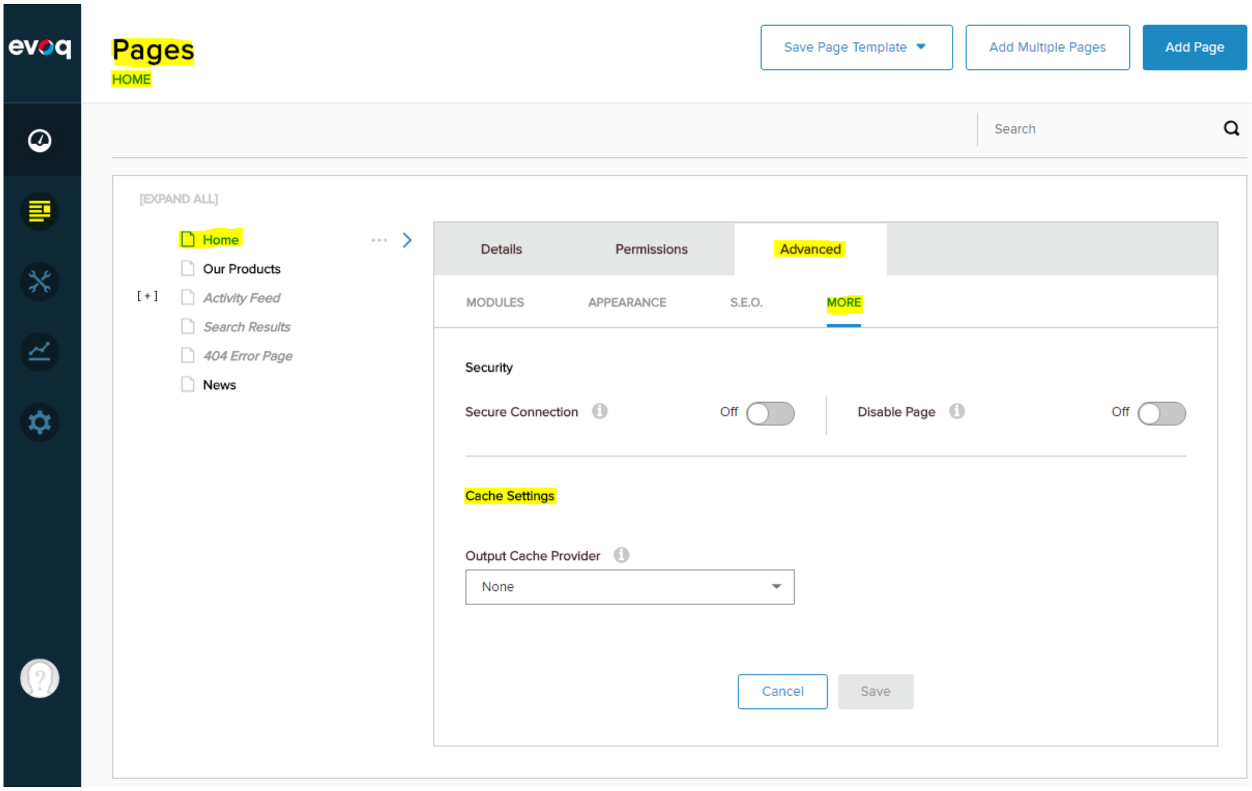
Cache Setting: ModerateCaching

Authenticated Cacheability: ServerAndNoCache

Unauthenticated Cacheability: ServerAndNoCache

SSL for Cache Synchronization: Off

Universal Cache settings



Secure Connection Off

Disable Page Off

Output Cache Provider: None

Page Cache Settings

The benefit of using memory caching everywhere is that 2 scheduled tasks can be disabled as well:

1. Purge Module Cache
2. Purge Output Cache

In total this will save a tremendous amount of disk traffic, which is especially interesting in a web farm situation

DNN SCHEDULER

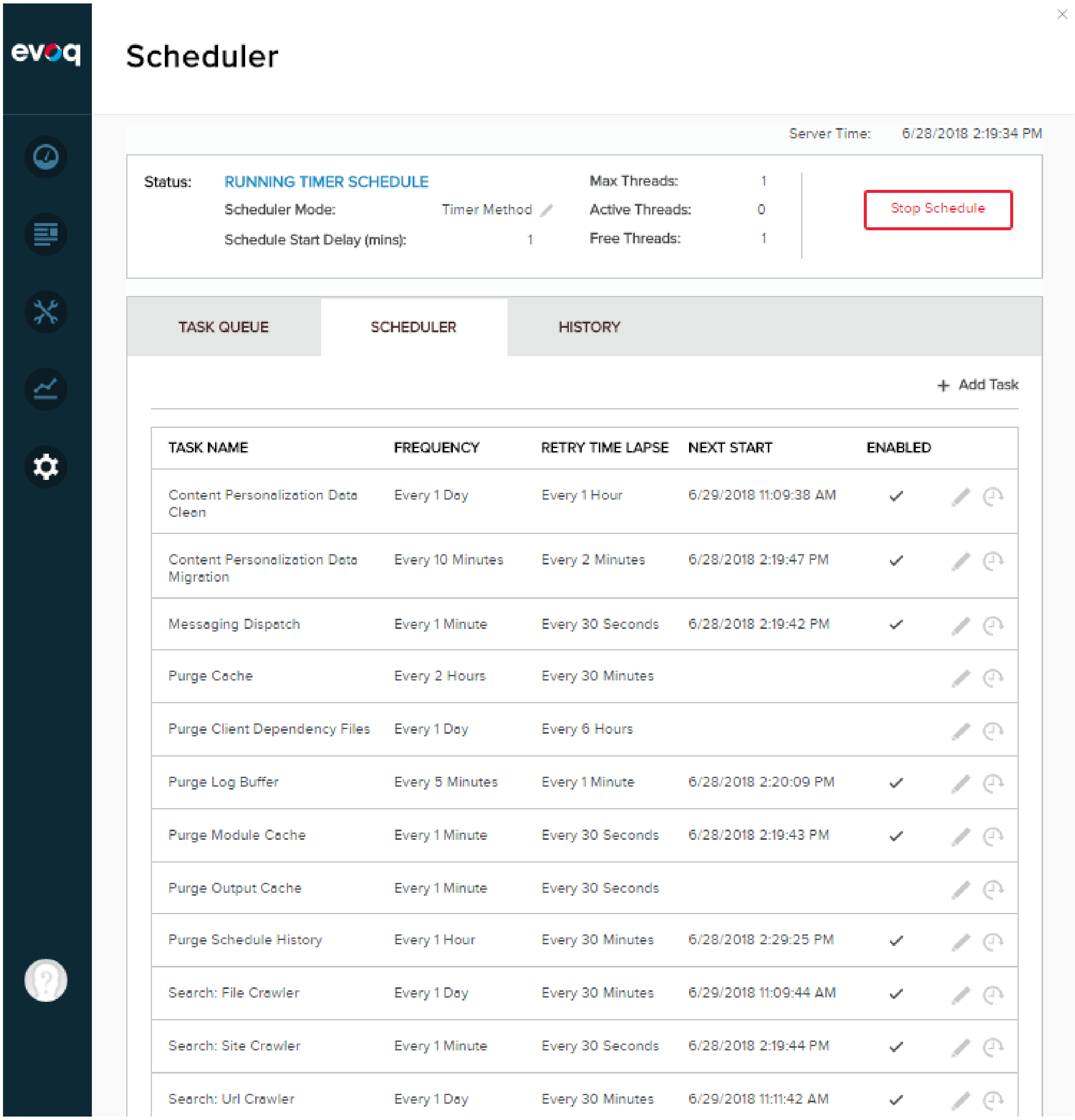
The DNN Scheduler is able to run background tasks within your DNN installation. Scheduled tasks are recurring events that typically perform housecleaning tasks in the database or file system. Almost all scheduled tasks that are delivered with DNN are designed to run on a single Web Server.

Based on Core Tasks, the following should run on ALL web servers in web farm:

1. Purge Log Buffer
2. Purge Users Online

To configure scheduled tasks, login as a Host user and go to the

Personabar > Settings > Scheduler > Scheduler



Scheduler

Server Time: 6/28/2018 2:19:34 PM

Status: RUNNING TIMER SCHEDULE

Scheduler Mode: Timer Method

Schedule Start Delay (mins): 1

Max Threads: 1

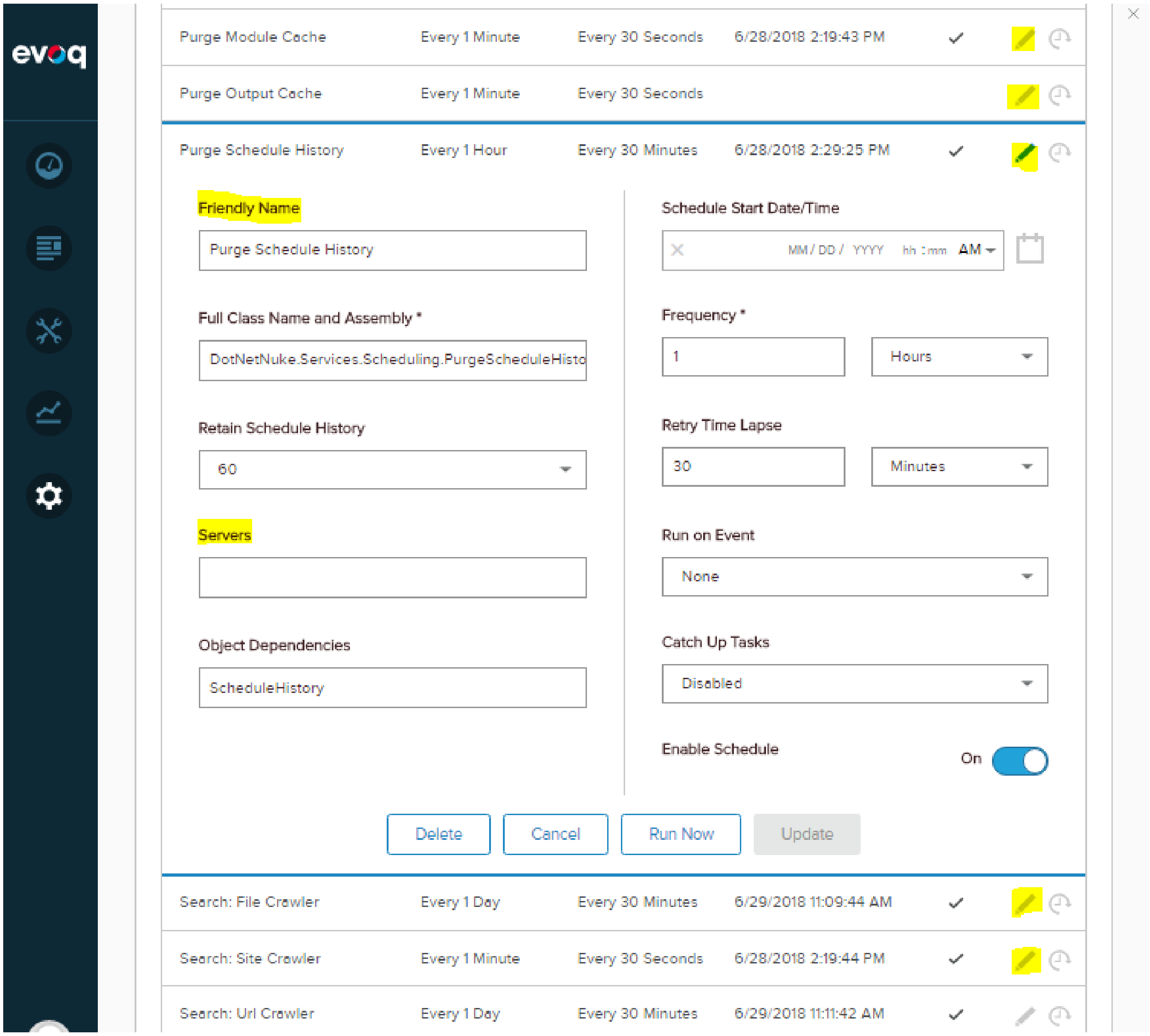
Active Threads: 0

Free Threads: 1

TASK NAME	FREQUENCY	RETRY TIME LAPSE	NEXT START	ENABLED
Content Personalization Data Clean	Every 1 Day	Every 1 Hour	6/29/2018 11:09:38 AM	✓
Content Personalization Data Migration	Every 10 Minutes	Every 2 Minutes	6/28/2018 2:19:47 PM	✓
Messaging Dispatch	Every 1 Minute	Every 30 Seconds	6/28/2018 2:19:42 PM	✓
Purge Cache	Every 2 Hours	Every 30 Minutes		
Purge Client Dependency Files	Every 1 Day	Every 6 Hours		
Purge Log Buffer	Every 5 Minutes	Every 1 Minute	6/28/2018 2:20:09 PM	✓
Purge Module Cache	Every 1 Minute	Every 30 Seconds	6/28/2018 2:19:43 PM	✓
Purge Output Cache	Every 1 Minute	Every 30 Seconds		
Purge Schedule History	Every 1 Hour	Every 30 Minutes	6/28/2018 2:29:25 PM	✓
Search: File Crawler	Every 1 Day	Every 30 Minutes	6/29/2018 11:09:44 AM	✓
Search: Site Crawler	Every 1 Minute	Every 30 Seconds	6/28/2018 2:19:44 PM	✓
Search: Url Crawler	Every 1 Day	Every 30 Minutes	6/29/2018 11:11:42 AM	✓

Click on the edit button next to each scheduled task to change the settings. Edit the "Run on Servers" setting with the machine name of the server or servers that should run this task.

Use a comma delimited list if you are running the task on multiple servers.



Purge Module Cache	Every 1 Minute	Every 30 Seconds	6/28/2018 2:19:43 PM	✓
Purge Output Cache	Every 1 Minute	Every 30 Seconds		
Purge Schedule History	Every 1 Hour	Every 30 Minutes	6/28/2018 2:29:25 PM	✓

Friendly Name: Purge Schedule History

Full Class Name and Assembly *: DotNetNuke.Services.Scheduling.PurgeScheduleHisto

Retain Schedule History: 60

Servers:

Object Dependencies: ScheduleHistory

Schedule Start Date/Time: MM / DD / YYYY hh : mm AM

Frequency *: 1 Hours

Retry Time Lapse: 30 Minutes

Run on Event: None

Catch Up Tasks: Disabled

Enable Schedule: On

Search: File Crawler	Every 1 Day	Every 30 Minutes	6/29/2018 11:09:44 AM	✓
Search: Site Crawler	Every 1 Minute	Every 30 Seconds	6/28/2018 2:19:44 PM	✓
Search: Url Crawler	Every 1 Day	Every 30 Minutes	6/29/2018 11:11:42 AM	✓

At this point your DNN installation is fully configured and ready for operation in the web farm.

INSTALLATION AND UPGRADE

Installing and upgrading DNN Evoq Content in a web farm environment is not much different from doing that on a single instance. However, it is a good practice to make sure that only one web head is actually being used to perform the installation or upgrade. This is to ensure that only one process is being used to run install procedures.

Installation

For the initial installation, this is fairly straightforward: make sure that the other web heads in the farm are not active yet.

Upgrade

During an upgrade, as a standard practice, we need to ensure that the application pools on other web heads are shut down. Post upgrade, ensure that a few page requests are made prior to starting the application pools on other web heads. This will allow running of IUpgradable on just one web head -- or else it may run on all the heads and result into collision. IUpgradable is an interface that can be implemented by modules, and this usually contains code that will upgrade module data. Usually this code is not re-entrant, which means it should only run once.

Another reason to shut down all but one web heads in the farm is that there may be some interference from the web request caching provider, which may result in errors during the upgrade.

7. TROUBLESHOOTING

Here are some ways you can check if web farm is configured properly:

1. ALL Webserver should use unique URLs.
2. The caching provider has to be able to talk to all servers outside of the load balancing mechanism.
3. In DNN HOST->Professional Features->Manage Web Servers are configured, are you able to browse to each server from the other servers in the farm? There should be direct connection (without any redirects) between all servers in the web farm. Make sure **WebRequestCachingProvider** is used. Try access remotely to each web server (use unique URL), and then open the web browser, and make a request to other servers (use unique URL, not URL of load balancer).
4. We do not support Web Gardens, so it shouldn't be used in IIS. See here for more background on DNN and web gardens: http://www.dnnsoftware.com/wiki/page/webgardens
5. In cloud environments, such as Windows Azure, machine names will be automatically assigned, and will change over time (after a redeployment). This impacts the way the Task Scheduler works, as in a web farm, tasks can be assigned to run on a specific server.
6. If the name of a server that is used to run a specific task on is no longer part of the web farm, the task will no longer be executed. In order to resolve this, edit the task and select the correct server(s) to run the task on.
7. In IIS logs do you see if the servers are receiving the request (/SynchronizeCache.aspx) from each other, and responding with a 200. If not, there could be something wrong with the underlying network or it can be result of URL Rewriting or handler problem.

SynchronizeCache.aspx is generated by DNN:

```
<add name="CacheSynchonizationHandler" verb="*" path="SynchronizeCache.aspx"
type="DotNetNuke.Professional.Providers.CachingProviders.WebRequestCachingPro
vid er.CacheSynchonizationHandler, DotNetNuke.Professional"
preCondition="integratedMode" />
```

Make sure .net version for this handler, if it is stated as pre-condition, corresponds with .net version for application pool.

APPENDIX 1: SYSTEM REQUIREMENTS

The following are required for DNN Platform 9.1.1 and DNN Evoq 9.1.1.

SUPPORTED OPERATING SYSTEMS

Windows 10

Windows 11

Windows Server 2016

.NET FRAMEWORK

4.7.2+

WEB SERVER

Microsoft IIS 7.5, 8.0, 8.5, 10

DATABASE SERVER

Microsoft SQL Server 2014

Microsoft SQL Server 2014 Express

Microsoft SQL Server 2016 (for DNN 8.0.4+)

Microsoft Azure SQL Database

Microsoft SQL Server 2017

Microsoft SQL Server 2019

Microsoft SQL Server 2022

BROWSERS

Chrome

Firefox

IE 11

Microsoft Edge

Safari for Mac

ADDITIONAL INFORMATION

ERRORS AND OMISSIONS

If you discover any errors or omissions in this document, please email marketing@ignitetech.com. Please provide the title of the document, the page number of the error and the corrected content along with any additional information that will help us in correcting the error.

DOCUMENT HISTORY

Version	Last Update	Author(s)	Changes
1.0	Nov 9, 2008	Scott Willhite	• Web Server Configuration
1.1	Mar 31, 2009	Joe Brinkman	• File Server Configuration • DB Configuration • DNN Configuration
1.2	Jan 15, 2010	Keivan Beigi	• Added Support IIS 7.0/Windows Server 2008 • Replaced FileBasedCachingProvider with WebRequestCachingProvider • Fixed naming inconsistencies throughout the document
2.0	Mar 26, 2010	Unknown	
3.0	July 1, 2012	Erik van Ballegoij	• Major Overhaul • Updated for newer software versions • Added SSL offload information • Added FAQ
4.0	Feb 1, 2014	Eleonora Ikponmwosa Erik van Ballegoij	• Complete Overhaul
4.1	Mar 7, 2014	Erik van Ballegoij	• Added Installation & Upgrade • Added task scheduler troubleshooting
9.0	June 30, 2018	Hamid Waqas	Updated Configuration according to 9.x.x Updated UI Screens to 9.x.x Updated System Requirements
9.1.1	Dec 14, 2023	Rodolpho Malvestiti	Updated System Requirements