solarwinds

INTERNET INFORMATION SERVICES (IIS) IMPLEMENTATION BEST PRACTICES:

SECURITY, LOAD TESTING, SCALABILITY, LOAD BALANCING

By: Terri Donahue, Microsoft IIS MVP

SOLARWINDS
WHITEPAPER

INTERNET INFORMATION SERVICES (IIS) IMPLEMENTATION BEST PRACTICES

INTRODUCTION

Before implementing your Internet Information Services (IIS) environment, consider the following critical elements: security, load testing, scalability, and load balancing. Each can greatly impact the user experience of your site. This paper outlines specific decisions and tasks you can perform to effectively plan and implement your IIS environment.

TABLE OF CONTENTS

SECURITY CONSIDERATIONS

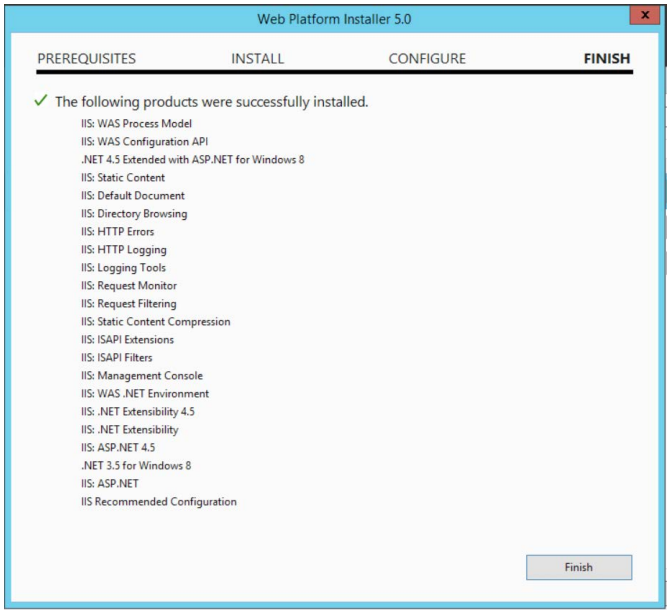
Features and Modules

To maintain optimal security, you need to evaluate your Web applications and decide what modules to install. Perform this evaluation when you deploy major application updates. Instead of enabling every module that IIS supports, enable only the modules required to keep your application functioning. Unused modules increase the footprint of your application to exploitable bugs. The process of removing unused modules from servers should be performed routinely.

You enable the IIS Role and additional features using the Web Platform Installer. After you install the application, complete these steps:

1. Run the application and search for **IIS Recommended Configuration**.
2. Click **Add** to select the option you want to install.
3. Search for **IIS: ASP**. Click **Add** on the **IIS: ASP.NET 4.5** option. Click Install at the bottom of the window, and click I Accept to begin the installation process

This enables the IIS Role, applies the recommended feature configuration, and enables .Net 4.5. Installing these features will configure IIS correctly to serve ASP.Net websites and applications. You will need to install any additional required IIS features beyond the following, which install automatically with this method:



File Locations

A default installation of IIS places the folders on the C drive. It is recommended to move your inetpub directory to a non-system partition. This should be done to ensure that if your IIS implementation is compromised, that access to the system folders is not granted. This blog post on iis.net explains how to move the folders to a different partition. It is written for IIS7, but the

same steps can be used to move the folders in IIS8 as well.

User Identities

As part of Service Pack 2 for Windows Server® 2008, Microsoft® introduced a new security feature for IIS called application pool identities. This new identity allows you to run an application pool under a unique identity without having to create or manage local or domain users. Beginning with IIS 7.5 (Windows Server 2008 R2), this is the default user account for any newly created application pool. For IIS7 (Windows Server 2008 SP2), you need to manually set up this user for each application pool. You can do this via the GUI or by using appcmd. Run the following from a command prompt to update the user of each application pool:

```
%windir%\system32\inetsrv\appcmd.exe set AppPool <your AppPool> -processModel.
identityType:ApplicationPoolIdentity
```

To Encrypt or Not Encrypt

Another part of the process involves deciding whether the information/functionality you are providing should be encrypted. You can perform the encryption on the communication between the client and server, and at the data layer. A recent federal employee breach involved unencrypted data (Social Security numbers).

An SSL certificate is used to encrypt sensitive data transferred over an unsecure network, such as the Internet. Without SSL implementation, data is visible as it is transmitted between your server and the recipient of the requested data. With SSL implementation, the data is encrypted until it reaches the destination computer. This protects the data, such as your credit card number, and ensures that only the requesting entity can decrypt the data for actual viewing. If an account or login is required, an SSL certificate should be acquired and implemented. The login page and all subsequently browsed pages of your application should require a secure browsing session. If you are simply providing information to anonymous or non-logged in clients, then an SSL certificate probably not needed.

Encryption Protocols and Ciphers

Configure cipher suites and protocols to address known vulnerabilities. This is a server setting rather than a setting specifically for IIS. Bettercrypto.org is an excellent source for up-to-date information about new vulnerabilities and recommended settings. Due to constantly changing recommendations, there is not a default configuration for these settings. As more vulnerabilities are discovered, weaker protocols and ciphers are recommended to be disabled to increase the security of your server. As a reference point, SSLv2 and SSLv3 should be disabled. IIS Crypto from Nartac Software is a handy tool for configuring both protocols and ciphers without having to make manual registry entry updates.

LOAD TESTING

Load testing provides a baseline for how your applications function under normal and peak loads. This can point to bottlenecks in the infrastructure that can then be addressed to ensure that your application performs in the best possible way. Visual Studio and other 3rd-party vendors offer ways to load test your application. By performing load testing, you can determine if features such as caching can aid in application performance.

Load testing can also reveal long-running SQL queries. If all Web metrics are within acceptable limits for your application, but pages are still returning data slowly, you might need to optimize SQL queries or stored procedures. You can implement SQL Query Analyzer to help determine whether changes need to be made on the SQL Server® database(s). All of these things can improve performance without having to scale.

When running load tests, you need to consider some key metrics. As new operating systems and IIS versions are released, these key metrics continue to be the best options for evaluating the health of your Web applications:

- System counters
  - Processor\% Processor Time
  - System\Processor Queue Length
  - Memory\Available Mbytes
  - Memory\Pages/sec
  - PhysicalDisk\% Disk Time
  - Network Interface\Bytes Total/sec
- IIS role-specific counters
  - ASP.NET Applications\Requests/Sec
  - ASP.NET\Application Restarts
  - ASP.NET\Request Wait Time
  - ASP.NET\Requests Queued
  - .NET CLR Exceptions\# of Exceptions Thrown / sec
  - .NET CLR Memory\# Total Committed Bytes
  - Web Service\Get Requests/sec
  - Web Service\Post Requests/sec
  - Web Service\Current Connections

- Web Service\URI Cache Flushes
- Web Service\URI Cache Hits
- Web Service\URI Cache Hits%
- Web Service\URI Cache Misses

SCALABILITY

Application

Scalability can relate to the application itself, or to the server infrastructure that hosts the application. Improvements for scalability have been included with each new version of IIS. IIS6 introduced application pools for process isolation. This provided the ability to host multiple applications or websites on the same server and ensure that the failure of any application was isolated from other applications running on the same server. The ability to limit CPU usage by an application pool also addressed runaway processes that could disrupt server activity.

Starting with Windows Server® 2012, several other enhancements were introduced that increased the performance and scalability of IIS websites. The Centralized Certificate Store and Server Name Indication are a few of the new features that have led to increased performance of websites on an IIS implementation.

IIS8.5 brought even more scalability with the introduction of Dynamic Website Activation and Idle Worker Process page-out. IIS8 changed the way SSL certificates are loaded, stored, and configured within IIS. Before IIS8, all SSL certificates were loaded into memory as IIS loaded. Starting with IIS8, SSL certificates are loaded as secured site requests are made, rather than loading into memory and holding that memory space hostage. With the introduction of worker process page-out and dynamic Web activation, you no longer need to load websites on start-up, or shut down worker processes. These two enhancements for more websites to be hosted on a single server, since rarely used applications and websites will not hold resources hostage that could be used for active sites.

Server

Scalability of the servers themselves leads to load balancing and high availability, which will be discussed later. Load balancing requires some configuration of the servers themselves to help make sure that IIS configuration files and Web root files stay in sync.

To prepare your application servers for load balancing, enable the following features:

- Distributed File System (DFS) Replication, if you have Active Directory®.
- Management Service.

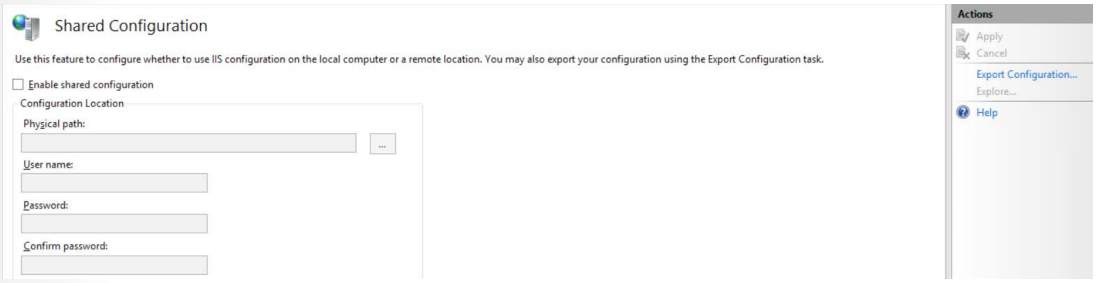
Depending on your network configuration, you can use PowerShell™ to install these features based on the following criteria:

1. If you have Active Directory:
   a. Open Windows PowerShell.
   b. Enter **Add-Windows Feature FS-DFS-Replication**; **Add-WindowsFeature Web-Mgmt-Service**.
   c. Click **Enter**.
   d. When the installation completes, close the PowerShell window.
2. If you have standalone\workgroup servers:
   a. Open Windows PowerShell.
   b. Enter **Add-WindowsFeature Web-Mgmt-Service**.
   c. Click **Enter**.
   d. When the installation completes, close the PowerShell window.

Depending on the environment, DFS/Robocopy is used to keep not only the Web root data synced between the member servers, but also the IIS configuration. Robocopy is a core Windows® executable beginning with Windows Server® 2008. You can find information related to configuring Robocopy here. For configuration of DFS, please see this blog post.

To configure IIS Shared Configuration:

A. Create a folder within inetpub named **Config**. This is the location where the IIS configuration will be saved. A DFS replication group or Robocopy task will be created for this folder.
B. Configure IIS Shared Configuration.
   a. Open IIS Manager.
   b. Select the server name.
   c. In the Features window, double click on the **Shared Configuration** icon to open the configuration screen.



   d. Select **Export Configuration** from the Actions pane.
   e. Enter the folder location you created previously and enter an encryption password. Document this password because you will need it later to enable shared configuration on all nodes. This password will be stored and is the

method that each server will use to decrypt any encrypted sections of the IIS configuration files.

f. Click **OK** on the export and successful boxes. You are now ready to set up Shared Configuration.

g. Click the **Enable Shared Configuration** checkbox.

h. Enter the folder location where you stored the configuration files in the previous step.

i. Leave the username/password boxes empty and click **Apply**.

j. Enter the encryption password and click **OK**.

k. You will be prompted to restart IIS Manager and the Management Service for the changes to take effect. Repeat these steps on all other nodes that will be load balanced.

LOAD BALANCING AND HIGH AVAILABILITY

The phrases load balancing and high availability are sometimes used interchangeably. Just because a system is highly available does not necessarily mean it is load balanced, and vice versa. Load balancing is a way of distributing work across multiple resources. High availability, on the other hand, relates to operational performance during a given period of time. The question is whether you need a load balanced and/or a highly available solution. Some applications, such as a personal blog, may not require either. It is frustrating when your blog experiences downtime, but it does not result in the loss of revenue. However, if you are hosting a sales application, extended downtime could result in lost revenue. For this instance, both a load balancing and highly available solution may be the correct choice. For either a load balanced or highly available solution, you need at least two Web servers. I will explain available options beginning with the least expensive.

Round Robin DNS

Round robin DNS, which requires at least two IIS servers, is the least expensive way to implement load balancing. DNS is used to route requests between the servers. Multiple A records are created for the same DNS name; one for each server to be included. When a DNS lookup is requested, the DNS server responds with the first record on the list. The record is moved to the end of the list and DNS replies to the next request with the next server. This continues until all records have been returned and the list starts over at the beginning. Round robin DNS does not provide a verification method to see if the destination is available before replying with the IP address for the request. Because of this limitation, you can still have end-users routed to a server that is not available to service requests.

Application Request Routing

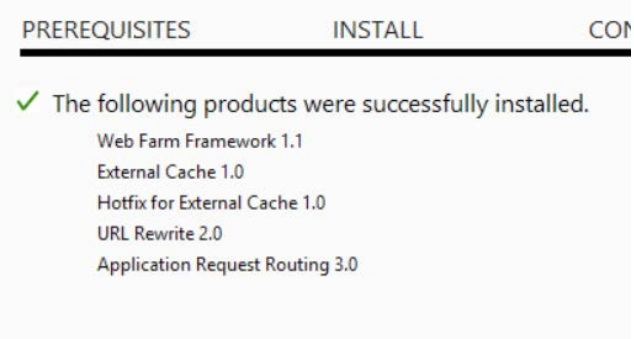
Application Request Routing (ARR) is a universal load balancing solution. It can be configured as a single node or a highly available cluster depending on your needs. Application Request Routing is an IIS module that provides a software-based load balancing solution. ARR is a Level 7 load balancer that monitors and routes traffic at the application layer (HTTP) rather than at the

network layer. ARR requires URL Rewrite, Web Farm Framework, and External Cache modules as well. The combination of these three modules provides the ability to define your server farms (groups of servers to be load balanced), route traffic to the correct nodes using URL Rewrite, and test individual Web nodes to ensure that they are healthy and available to accept Web traffic. If you need to perform traffic analysis that requires a preserved client IP address, you should also install ARR Helper on each of the Web nodes to which ARR will route traffic.

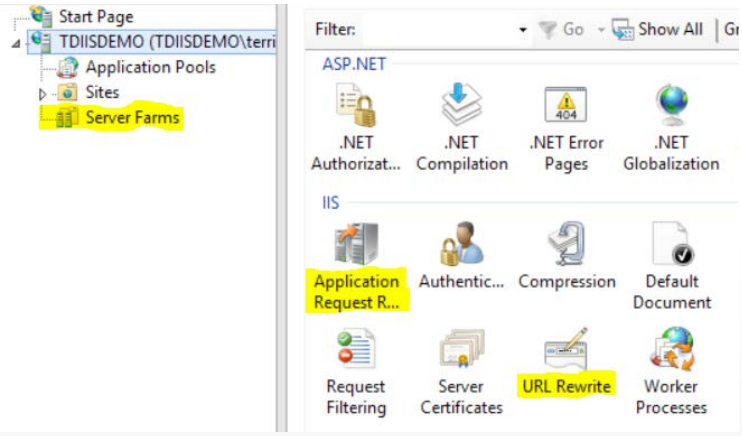
ARR is a unique solution for IIS as a software-based load balancing solution because it is a Microsoft® module that integrates with IIS and is configured using IIS Manager. The application can be configured to load balance port 80 and 443 (standard Web application ports) and any other ports you choose.

Before beginning the install and initial ARR configuration, know that the ARR server should be a separate server from the Web nodes.

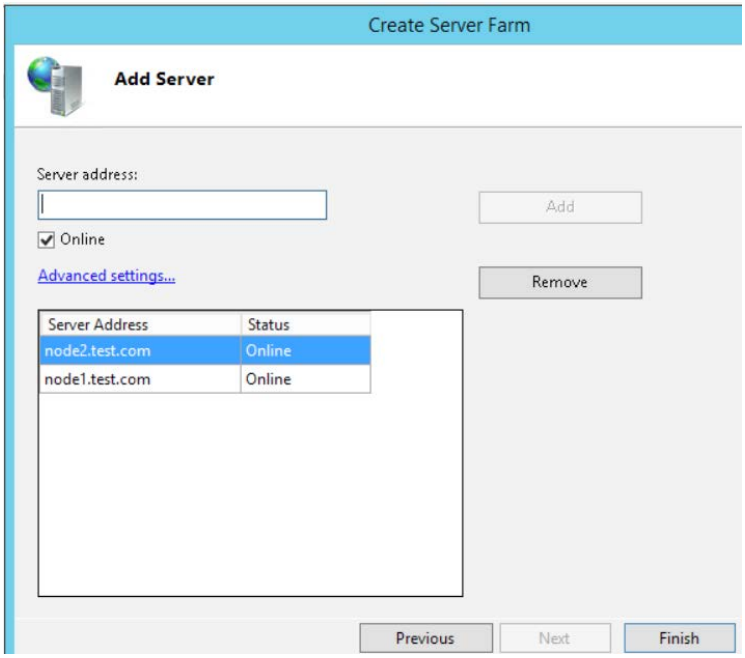
If you do not already have IIS installed, complete the installation steps at the beginning of this document. You can then use the Web Platform Installer to add the additional modules to create the ARR instance. To do this, search for ARR and click **Add** on **Application Request Routing 3.0**. Click Install and I **Accept** to run the installation process. Upon completion of the install, a list of all dependent modules that were also installed displays.



As soon as the installation is complete, you are ready to begin configuring your ARR instance for load balancing. Open IIS Manager. You will notice a few new icons (**Application Request Routing** and **URL Rewrite**) at the server level and the **Server Farms** group.



To configure your first server farm, right-click on **Server Farms** and select **Create Server Farm**. Enter an appropriate name for your server farm and click **Next**. On the following screen, enter the DNS name or IP address of the first server and click **Add**. Repeat these steps for each server you want to add to load balancing.

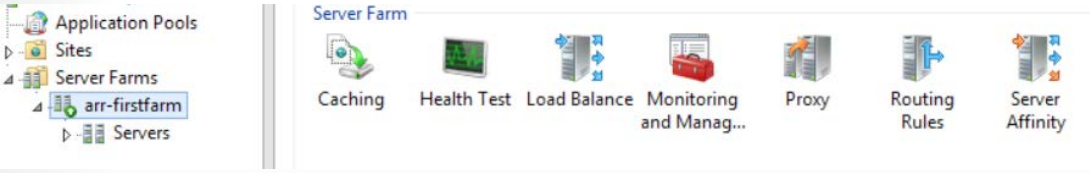


Click **Finish** to complete the setup of the server farm, which is the group of servers to which the ARR server will route requests. Click **OK** to create the URL Rewrite rule for this server farm. Note: choose this option only for the first server farm to ensure that routing is not broken as you create subsequent farms.

By automatically creating the rule for the first farm, you have a template to use for future reference. You will need to edit the rule to match a specific HTTP_HOST if you have more than one server farm defined. To view the rule, click on the **Server Name** in IIS Manager and then the **URL Rewrite** icon. You will see your rule listed in the GUI. This rule examines all traffic to match the wildcard pattern and then routes it to the server farm **arr-firstfarm**. Based on the settings for the server farm, the traffic is then routed to the nodes associated with the server farm based on the defined rules configured.

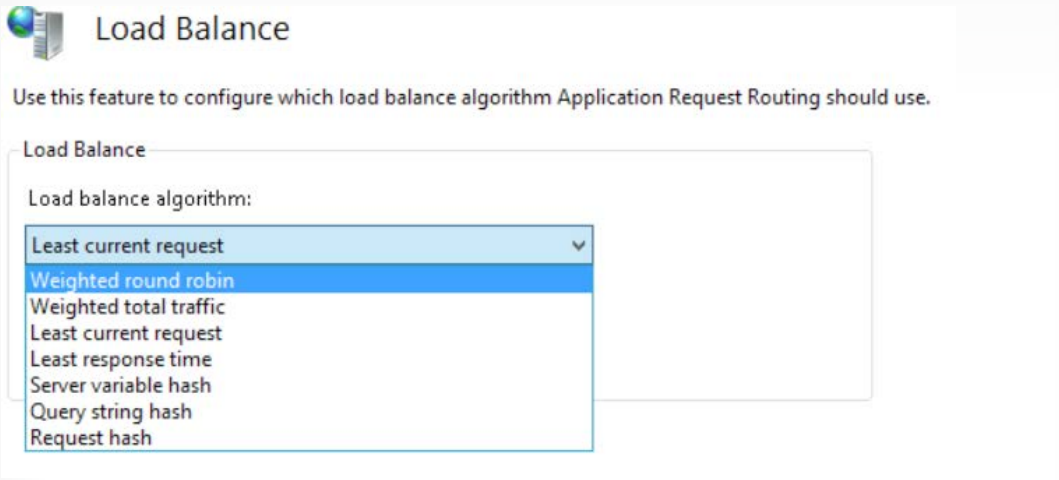
Name	Input	Match	Pattern	Action Type	Action URL	Stop Proce...	Entry Type
ARR_arr-firstfarm_loadb...	URL Path	Matches	*	Rewrite	http://arr-firstfarm/{R:0}	True	Local

The IIS Manager GUI displays multiple options to configure your server farm for load balancing.

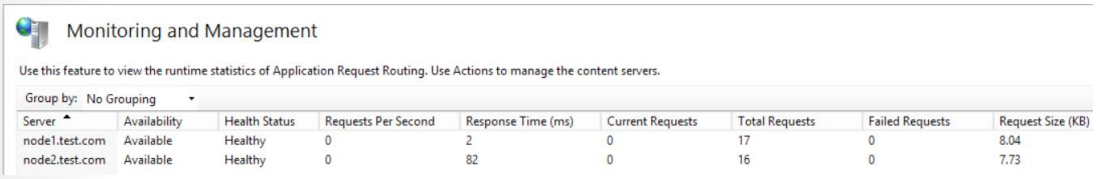


To begin, the **Load Balance** feature has every option you could possibly want to configure. The options include traditional as well as hash options for routing clients to the configured

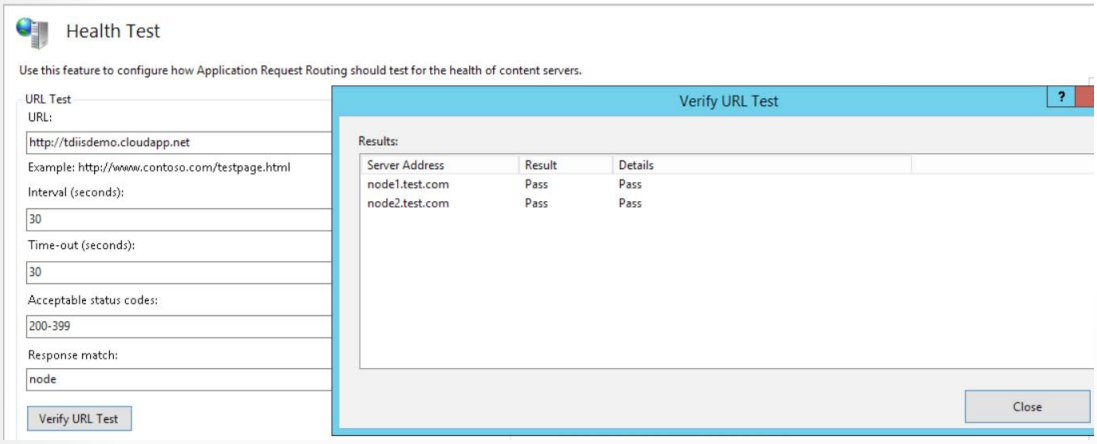
nodes. Weighted round robin ensures that traffic is routed to each node using even or custom distribution (different percentage to each node) setting.



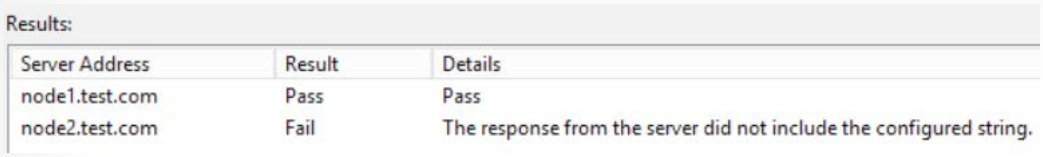
The Monitoring and Management feature allows you to view the traffic that is being routed to the individual nodes. The screenshot below shows traffic patterns from an evenly distributed weighted round robin implementation. Node 1 is in the same physical location and network as the ARR server. Node 2 is an external server, which explains the difference in response time for the requests.



When you have tested and verified that traffic is routing to the nodes as expected, you can configure a health test page. Configure the URL, acceptable status codes, and response match. For this demo, the default page returns a text value of node1 or node2. Clicking **Verify URL** Test sends a request to each node and verifies that the returned status code and response match are correct and will return a result of **Pass**.



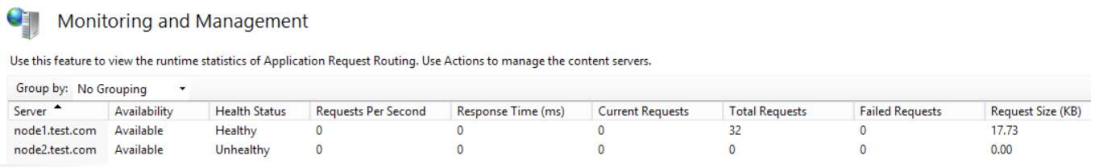
If either of the returned values do not match the defined settings, you will receive a failure message in the Verify URL Test Results box.



Results:

Server Address	Result	Details
node1.test.com	Pass	Pass
node2.test.com	Fail	The response from the server did not include the configured string.

ARR then marks the node as unhealthy and will not route additional traffic to that node until the issue is resolved.



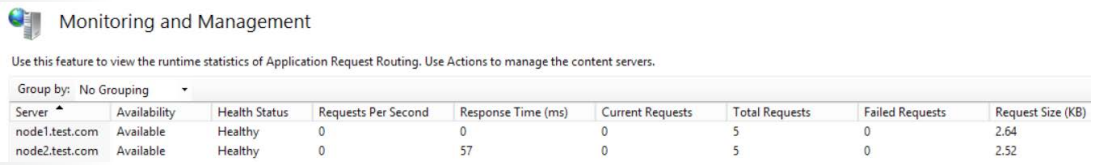
Monitoring and Management

Use this feature to view the runtime statistics of Application Request Routing. Use Actions to manage the content servers.

Group by: No Grouping

Server	Availability	Health Status	Requests Per Second	Response Time (ms)	Current Requests	Total Requests	Failed Requests	Request Size (KB)
node1.test.com	Available	Healthy	0	0	0	32	0	17.73
node2.test.com	Available	Unhealthy	0	0	0	0	0	0.00

After the issue is resolved on the unhealthy node, ARR returns the node to a healthy status and begins routing traffic to it again.



Monitoring and Management

Use this feature to view the runtime statistics of Application Request Routing. Use Actions to manage the content servers.

Group by: No Grouping

Server	Availability	Health Status	Requests Per Second	Response Time (ms)	Current Requests	Total Requests	Failed Requests	Request Size (KB)
node1.test.com	Available	Healthy	0	0	0	5	0	2.64
node2.test.com	Available	Healthy	0	57	0	5	0	2.52

As you can see, ARR is a robust software-based load balancer that offers health check capabilities for real-time monitoring of the nodes, and automatically directs traffic as needed if a node becomes unhealthy.

Once testing and verification are complete, the DNS entry for websites that are not load balanced should be updated to point to the IP(s) of the ARR server.

Hardware Load Balancers

Hardware load balancers are physical devices that intercept and route traffic according to a set of rules. This is the most expensive, but most robust solution for load balancing. Hardware devices are designed for a specific purpose. Because of this, the attack footprint is smaller and updates are not released as often as for software-based implementations. Hardware load balancers can also provide higher scalability than software load balancers due to the reduced overhead required by the software application. These types of load balancers are generally Level 4 (network) devices. There are many providers of hardware load balancers, and pricing varies. These types of devices are also able to provide the health check capabilities of ARR to ensure that traffic is only routed to healthy nodes. The configuration of these devices can be performed via a command line or Web-based utility depending on the manufacturer.

ABOUT THE AUTHOR

Terri is an infrastructure administrator who has been working with IIS since version 4.0. She has been a Microsoft® MVP for IIS since 2013. She is one of the authors of the Official Microsoft Learning Product 10972B: Administering the Web Server (IIS) Role of Windows Server. She has extensive hands-on experience with many Web servers, including Lotus® Domino, Apache®, and, of course, IIS. She is passionate about helping people solve technology related problems.