

# Enabling HTTP/2 on an IBM® Lotus Domino® Server

## Setup Guide



## CONTENTS

Introduction .....	3
Requirements .....	3
About HTTP/2 .....	3
About NGINX .....	3
How this works .....	4
Step 1 – Install NGINX .....	5
Step 2 – Setting up NGINX to run as a Windows Service.....	6
Step 3 – Update Windows Hosts File.....	8
Step 4 – Add another local IP Address .....	8
Step 5 - Creating SSL Certificate Files .....	9
Extracting certificates from an existing Domino Keyring File.....	9
Create a new SSL Certificate .....	11
Step 6 – The NGINX Configuration File .....	13
Step 7 - Key Exchange Security .....	14
Step 8 – Enable Connector Headers .....	14
Step 9 – Testing .....	15
Step 10 – Update Forwarding on your Router .....	16
Step 11 – Firewall Settings .....	16
Step 12 - Final Setup .....	16
Tips .....	17
Check your website's performance .....	17
Multiple Websites .....	17
Forcing all connections to be secure .....	17
Bringing your Domino server down for maintenance.....	18
Suggestions or Comments.....	19
About AGEKOM.....	19

## INTRODUCTION

This guide describes how you can provide HTTP/2 connectivity to your Lotus Domino server using NGINX - a free open-source high-performance HTTP server and reverse proxy. With HTTP/2 connectivity available to your Domino server web pages will load much faster.

The steps covered in this guide describe how to install and configure NGINX and set it up as a HTTP/2 proxy for incoming connections.

The AGEKOM website provides HTTP/2 connectivity using the configuration described in this guide. You can check the AGEKOM website at: <https://www.agecom.com.au>

## REQUIREMENTS

The information provided in this guide is based on the following system settings:

- Microsoft Windows server 2008 or later.
- Lotus Domino 9.0.1 or later

You may be able to use the information for guidance with other platforms and versions.

It is assumed you already have an active website running on your Domino server.

## ABOUT HTTP/2

HTTP/2 improves speed mainly by creating one constant connection between the browser and the server, as opposed to a connection every time a piece of information is needed. This significantly reduces the amount of data being transferred. Plus, it transfers data in binary, a computer's native language, rather than in text.

Other features of HTTP/2 include "multiplexing" (sending and receiving multiple messages at the same time), the use of prioritization (more important data is transferred first), compression (squeezing information into smaller chunks) and "server push," where a server makes an educated guess about what your next request will be and sends that data ahead of time.

## ABOUT NGINX

NGINX is a free open-source high-performance HTTP server and reverse proxy, a mail proxy server, and a generic TCP/UDP proxy server. It's known for its high performance, stability, rich feature set, simple configuration, and low resource consumption.

Many web servers and application servers use a simple threaded or process-based architecture however NGINX stands out with a sophisticated event-driven architecture that enables it to scale to hundreds of thousands of concurrent connections on modern hardware.

Once you've got NGINX setup it can receive all incoming connections and redirect them to the appropriate Domino server / website.

You can install NGINX on your Domino server or any other server on your network.

More information regarding NGINX can be found on the NGINX website at: <http://nginx.org>



## HOW THIS WORKS

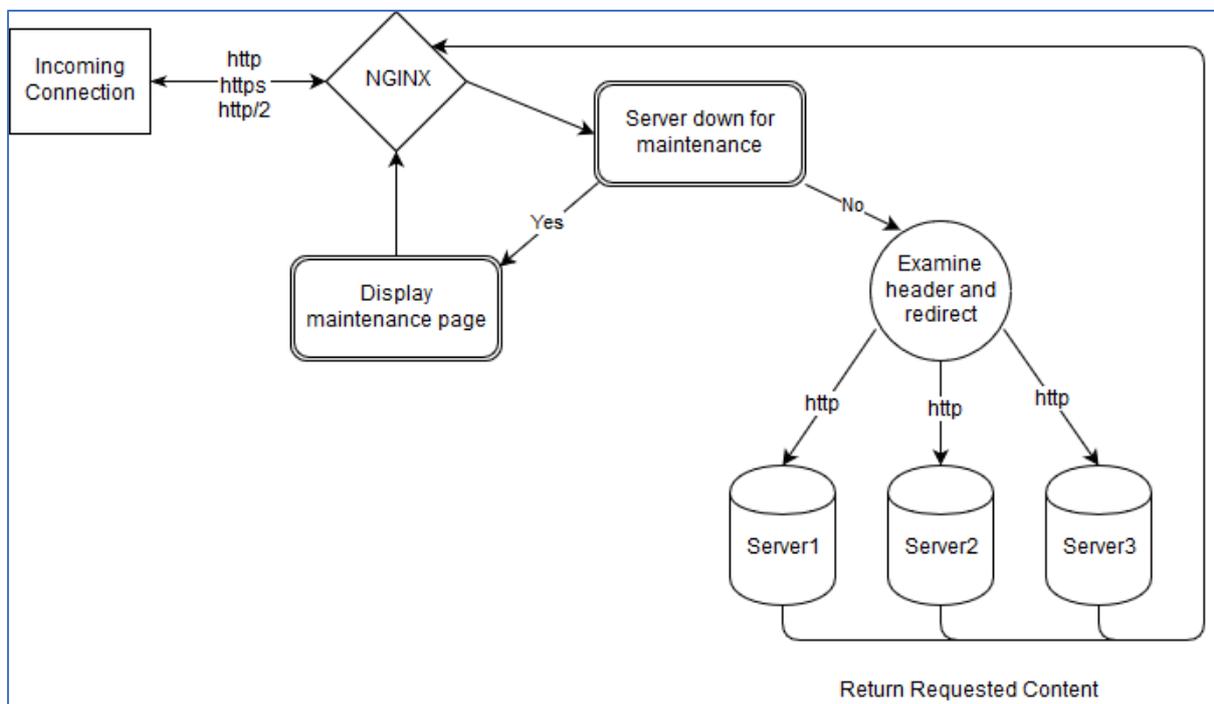
After following the steps outlined in this guide you'll have NGINX handling incoming connections from the Internet and directing them to the appropriate server based on the target hostname. If you've specified to display a maintenance page (in the event a server is down for maintenance) the maintenance page will be returned.

Connections from the Internet may be http, https, or http/2. NGINX is responsible for establishing and maintaining secure connections (SSL). You can also configure NGINX to force all incoming connections to establish a secure connection.

Connections from NGINX to your internal servers are established over http so once you have this setup in place you won't need to maintain a Domino keyring file.

Requested content is returned from your servers back to NGINX and it in turn returns that content to the requestor. Secure connectivity is always maintained if that's how the original connection was established.

The following diagram shows the connectivity flow in basic terms.



Features of NGINX:

- Application Acceleration
- Content Cacheing
- SSL and SPDY Termination
- Bandwidth Management
- Content-Based Routing
- Request Manipulation
- Response Rewriting
- Authentication
- Streaming Media
- Monitoring
- Configuration

## STEP 1 – INSTALL NGINX

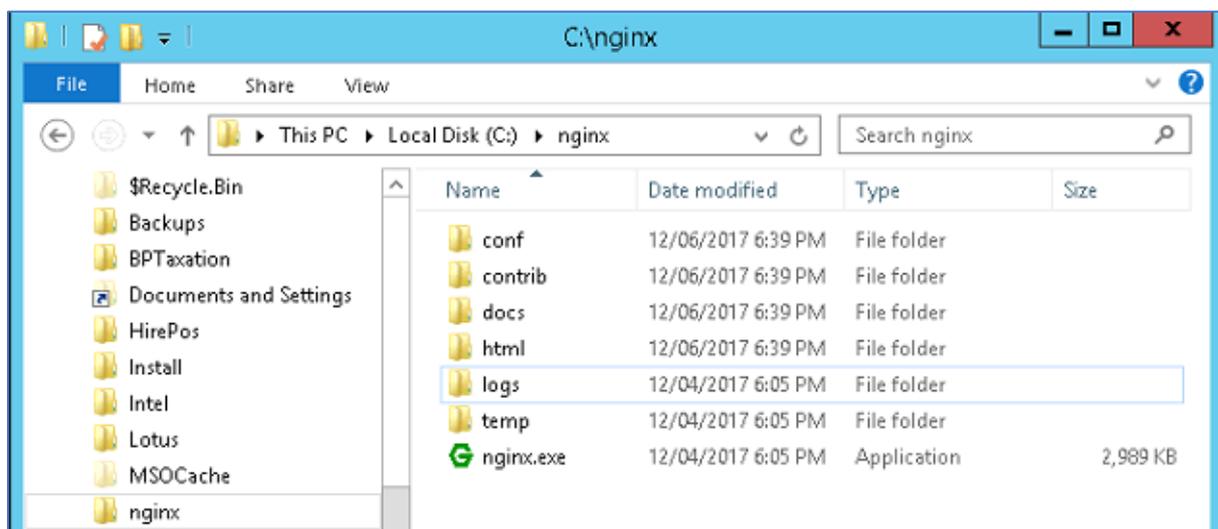
The NGINX Windows binary file should be downloaded from the NGINX website at:

<http://nginx.org/en/download.html>

The most recent stable version should be downloaded. It's recommended you make a note of the version downloaded so you can reference it later when you need to determine what version is currently installed.

NGINX may be installed on your Domino server or any other Windows server.

Create a folder in the root directory of your server called 'nginx' and extract all files from the downloaded zip file in the folder. There should now be an executable file called nginx.exe in the nginx folder and several other subfolders.



That's pretty much it to installing nginx. We still, however, need to do the following which is covered in this guide:

- Setup nginx to run as a Windows service
- Setup the NGINX configuration file
- Setup SSL

## STEP 2 – SETTING UP NGINX TO RUN AS A WINDOWS SERVICE

NGINX is not an executable that can be installed as a Windows service. To work around this a Windows Service Wrapper is available which can be installed as a Windows service and in turn will control the loading and unloading of NGINX. The wrapper can actually be used to host any executable as a Windows service.

The Windows Service Wrapper is available on Github at: <https://github.com/kohsuke/winsw>

You can go straight to the downloads page at: <https://github.com/kohsuke/winsw/releases>

Download the latest Windows service executable and save it to the NGINX folder. At the time this guide was written the recommended version to download is the one written for .NET4 (ie. WinSW.NET4.exe).

After the executable has been downloaded it is recommended that you rename it to 'nginx-winsw.exe' to make it clear which particular application the wrapper service will be responsible for loading and unloading.

The wrapper service requires configuration settings to be placed in an xml file matching the name of the executable. For this an xml file called 'nginx-winsw.xml' should be created in the same folder as the nginx-winsw.exe file.

The following items demonstrate the minimum settings required to be added to the xml file:

```
<service>
  <id>nginx</id>
  <name>nginx</name>
  <description>nginx</description>
  <executable>c:\nginx\nginx.exe</executable>
  <logpath>c:\nginx\</logpath>
  <logmode>roll</logmode>
  <depend></depend>
  <startargument>-pc:\nginx</startargument>
  <stopexecutable>c:\nginx\nginx.exe</stopexecutable>
  <stopargument>-s</stopargument>
  <stopargument>stop</stopargument>
</service>
```

A sample xml file for the Windows service wrapper can be downloaded from the [AGECOM Support Note](#).

After the Windows service wrapper has been downloaded and the xml configuration file created it must now be installed as a Windows service.

To install nginx-winsw.exe as a Windows service open a command prompt as Administrator and go to the nginx folder. Type in the following command: **nginx-winsw.exe install**



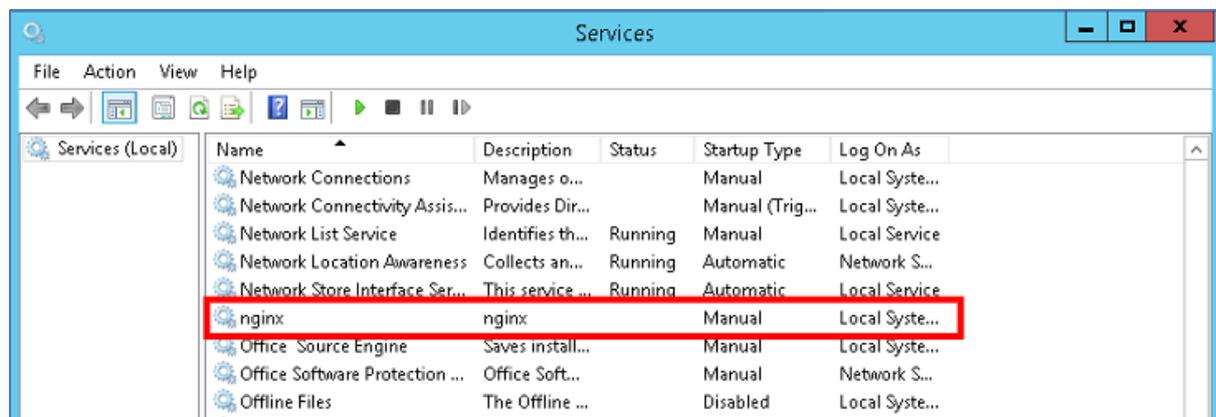
```
Administrator: Command Prompt
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>cd\nginx
C:\nginx>nginx-winsw.exe install
2017-06-12 18:59:44,547 INFO - Starting ServiceWrapper in the CLI mode
2017-06-12 18:59:45,923 INFO - Completed. Exit code is 0

C:\nginx>_
```

A new Windows service called 'nginx' will now have been created and set to start automatically on boot.

It is recommended you go into Windows services and change this to start Manually for now. Eventually this will be set to start Automatically (Delayed) once everything has been confirmed as working correctly.



### STEP 3 – UPDATE WINDOWS HOSTS FILE

The hostname of your web server and the local IP address it is bound to need to be added to the Windows hosts file that NGINX is being installed on - normally found in the C:\Windows\System32\Drivers\etc folder.

For example if your hostname is www.acme.com and it is bound to the local ip address of 10.0.0.25 (the ip address your Domino server http task is listening on for incoming connections) then you would add the following entries to the hosts file:

- 10.0.0.25 acme.com
- 10.0.0.25 www.acme.com

After updating the hosts file you should ping both forms of the above hostnames on your server to ensure they resolve to the expected local ip address and you get a successful reply back.

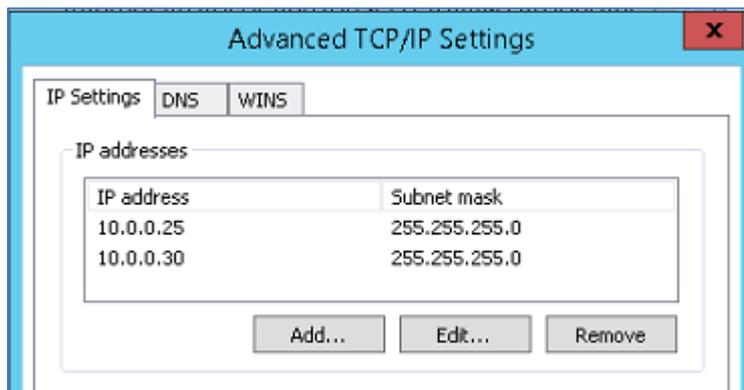
### STEP 4 – ADD ANOTHER LOCAL IP ADDRESS

If NGINX has been installed on your Domino server the server will require a minimum of 2 local static ip addresses. One ip address will be used by NGINX to receive incoming connections, and another ip address will be used by the Domino server.

NGINX receives incoming HTTP connections on one of the ip addresses and will redirect traffic to the ip address used by Domino (either on the current machine or another machine where Domino has been installed).

You should now go into the Windows network settings and add another static ip address that is not being used elsewhere on the network.

The following image shows a secondary ip address of 10.0.0.30 that has been added:



Note: If NGINX has been installed on another server where there are no other applications that are using any of the http ports (80, 443) then you should be able to use a single ip address and won't need to add a second ip address.

## STEP 5 - CREATING SSL CERTIFICATE FILES

As most browsers only support HTTP/2 over SSL you will need setup SSL for your website domain. The following sections describe how to extract your keys and certificates from an existing Domino keyring file or create a new SSL certificate from scratch.

### EXTRACTING CERTIFICATES FROM AN EXISTING DOMINO KEYRING FILE

If you have an existing Domino Keyring file which contains certificates that have not expired you can extract them using the Domino Keyring Tool (version 1.1 or later). For more information about this tool and to download it please refer to the following link on IBM's website:

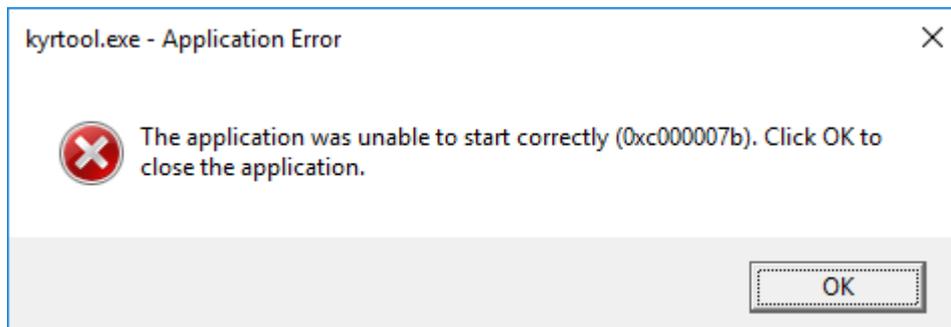
<http://www-01.ibm.com/support/docview.wss?uid=swg21966137>

After downloading the tool copy the 32-bit version to your Lotus Notes programs folder. You will also need to copy the Domino keyring file (kyr) and the password stash file (sth) to your local drive. Open a command prompt and switch to the Notes program folder.

Enter the following command to view the public and private keys in the file:

**kyrtool show keys -k [path to keyring file]**

Note: If the following error message is displayed after typing in the above command then you have most likely copied the 64-bit version of KyrTool. Ensure you copy the 32-bit version of the tool then try again.



Upon successful execution of the above command you will see output similar to the following:

```
Using keyring path '[path of your keyring file]'
```

```
Key length: 2048 bits
```

```
-----BEGIN PUBLIC KEY-----
```

```
[Your public key]
```

```
-----END PUBLIC KEY-----
```

```
Key length: 2048 bits
```

```
-----BEGIN RSA PRIVATE KEY-----
```

```
[Your private key]
```

```
-----END RSA PRIVATE KEY-----
```

The private key needs to be saved to a file. The easiest way to do this is to repeat the last command but direct output to a file. Enter the following command:

```
kyrtool show keys -k [path to keyring file] > private.key
```

The website domain certificate, and the certificate issuer's intermediate and root certificates also need to be extracted and saved to another file. Enter the following command to extract them to a file:

```
kyrtool show certs -k [path to keyring file] > bundle.crt
```

Since the file will contain multiple certificates we recommend naming it to something like 'bundle.crt'. You might also want to prefix with the file with something indicating the certificate issuers name (eg. Godaddy-bundle.crt, entrust-bundle.crt, etc).

The private.key file should now be edited to remove content that is not required – leaving only the private key.

Open the private.key file in a text editor. All content before the line '-----BEGIN RSA PRIVATE KEY-----' should be removed and any content after the line '-----END RSA PRIVATE KEY-----' should be removed.

The contents of the private key should now be only:

```
-----BEGIN RSA PRIVATE KEY-----  
  
[Your private key]  
  
-----END RSA PRIVATE KEY-----
```

Save and close the private key file.

The bundle.crt file should now be edited. Everything outside of the blocks starting with '-----BEGIN CERTIFICATE-----' and ending with '-----END CERTIFICATE-----' should be removed. The contents of the bundle file should now contain only the domain certificate, and issuer's intermediate & root certificates in the following format:

```
-----BEGIN CERTIFICATE-----  
  
[Your certificate]  
  
-----END CERTIFICATE-----  
  
-----BEGIN CERTIFICATE-----  
  
[Intermediate Certificate]  
  
-----END CERTIFICATE-----  
  
-----BEGIN CERTIFICATE-----  
  
[Root Certificate]  
  
-----END CERTIFICATE-----
```

Save and close the bundle file.

When you've finished extracting the keys and editing the files they should now be copied to the 'conf' folder under the main NGINX folder.

## CREATE A NEW SSL CERTIFICATE

If you don't have an existing Domino Keyring file or need to create a new SSL certificate you will need to generate a certificate signing request (CSR) and obtain one from a certificate authority (CA). A CSR consists mainly of the public key of a key pair (private and public), and some additional information (such as domain name, location, etc). Both of these components are inserted into the certificate when it is signed.

The easiest way to generate these is using OpenSSL which can be downloaded at;

<https://www.openssl.org/>

You can download an OpenSSL binary distribution with Installer from the following website:

<https://slproweb.com/products/Win32OpenSSL.html>

You should download the latest 32-bit or 64-bit version and run the installer.

Using OpenSSL the following example command would be entered at the command prompt (in the OpenSSL bin folder) for generating the private key and CSR files for the Acme domain:

```
openssl req -new -newkey rsa:2048 -nodes -keyout acme.key -out acme.csr
```

Note: you should replace 'acme' in the above command with your own domain name.

Additional information may be requested after entering the above command which you should supply such as:

- Country Code (eg. AU)
- State, Territory, or Province in full (eg. New South Wales)
- City or Locality (eg. Sydney)
- Organization Name (eg. Acme Pty Ltd)
- Organization Unit Name / Department (eg. Marketing)
- Common Name / Full Qualified Domain Name (eg. www.acme.com)
- Email Address
- Challenge Password
- Optional company name

Note: If you get the following error: *WARNING: can't open config file: /usr/local/ssl/openssl.cnf* then type in the following command: **set OPENSSL\_CONF=[path-to-OpenSSL-install-dir]\bin\openssl.cfg** (example: *set OPENSSL\_CONF=c:\OpenSSL-Win64\bin\openssl.cfg*) then try the original openssl command again.

The following two files should now have been generated:

- acme.key – this contains your private key and should be kept secure.
- acme.csr – this contains the certificate signing request and what you will submit to the certificate issuer.

The file containing the private key (ie. acme.key) should be copied to the 'conf' folder under the main NGINX folder.

The CSR file should now be used to submit your certificate request to your chosen certificate issuer. Information on submitting your request, obtaining your certificate and intermediate / root certificates should be available on the issuer's website.

After your certificate has been issued it should be downloaded and put into a file. The issuer's root certificates (and any intermediate certificates) also need to be put into the same file so it's recommended it's named something like domain-bundle.crt (eg. acme-bundle.crt).

The order of the certificates in the file is important for SSL to work. The first certificate in the bundle file should be the certificate for your domain, then the issuers intermediate certificates, and the issuer's root certificate appended to it.

The bundle file should be formatted like the following:

```
-----BEGIN CERTIFICATE-----  
[Your certificate]  
-----END CERTIFICATE-----  
-----BEGIN CERTIFICATE-----  
[Intermediate Certificate]  
-----END CERTIFICATE-----  
-----BEGIN CERTIFICATE-----  
[Root Certificate]  
-----END CERTIFICATE-----
```

The certificate bundle file should be copied to the 'conf' folder under the main NGINX folder when you've finished adding all certificates to it.

## STEP 6 – THE NGINX CONFIGURATION FILE

The NGINX configuration file contains all the settings used by NGINX. The file is called 'nginx.conf' and resides in the 'conf' folder under the main NGINX folder.

It is beyond the scope of this article to describe the settings in configuration file. There are plenty of resources on the internet which cover the settings such as:

[NGINX - Beginners Guide](#)

[NGINX - Configuring HTTPS Servers](#)

[Martin Fjordvald - NGINX Configuration Primer](#)

[DigitalOcen - Understanding the NGINX Configuration File Structure and Configuration Contexts](#)

The sample configuration file attached to the [AGECOM Support Note](#) is ready for use and can be copied to the 'conf' folder under the main NGINX folder. It should be named 'nginx.conf'.

If you choose to use the sample file the following customizations will be required:

- The IP address assigned for NGINX to listen on for incoming connections (Lines 59, 79)
- The domain names for your site(s) (Lines 60, 67, 68, 80, 92)
- The path / filename of the private key file (Line 126)
- The path / filename of the bundled certs file (Line 123)

You will also notice the sample configuration file refers to a file in the html folder called 'maintenance.html' (also attached to the support note). If NGINX finds this file it will display the contents of it instead of redirecting to your website.

This is useful if you want to bring your Domino server down for maintenance and provide a message back to incoming connections to indicate maintenance is being performed. It should be copied to the 'html' folder which is a subfolder of the main NGINX folder.

Whenever you want to display the maintenance message simply rename the file to 'maintenance.html' and when it's not required rename it to something else such as 'zz-maintenance.html'.

Note: The maintenance.html file should be customized with your website and name and email address.

## STEP 7 - KEY EXCHANGE SECURITY

The first step in the establishment of a secure connection is the exchange of the private keys between server and client. By default, NGINX uses a 1028-bit DHE (Ephemeral Diffie-Hellman) key, which is relatively easy to decrypt. To provide maximum security it is recommended you create your own more secure DHE key.

Again using OpenSSL open a command prompt and go to the OpenSSL bin folder. Enter the following command:

**set RANDFILE=[path-to-OpenSSL-install-dir]\bin\rnd**

Example: set RANDFILE=c:\OpenSSL-Win64\bin\rnd

Now enter the following command to create a parameters file with a 4096-bit key length:

**openssl dhparam -out dhparam.pem 4096**

The generation process will take several minutes.

Note: if you get an error message '*Unable to write random state*' while the file is being generated make sure you entered the 'set RANDFILE=...' command correctly first then try running the openssl command again. If the problem persists open a new command prompt with Administrator privileges and try running the commands again.

Once the process has completed copy the dhparam.pem file to the 'conf' folder under the main NGINX folder.

## STEP 8 – ENABLE CONNECTOR HEADERS

The following settings should be added to the notes.ini file used by the Domino server if it doesn't already have it:

**HTTPEnableConnectorHeaders=1**

This setting enables the Domino HTTP task to process special headers that are added to requests by NGINX.

When NGINX passes the HTTP requests to Domino it adds headers that include information about the incoming connection such as the remote IP address. Without this any IP address recorded for incoming HTTP connections will be that used by NGINX rather than the actual remote user's address.

Information added to the headers can be seen in the nginx.conf file as 'proxy\_set\_header' items.

## STEP 9 – TESTING

Now that all setup and configuration steps have been performed you should be able to test the connection.

Go to the Windows Services on the Domino server and ensure the 'nginx' service has been started. If not then start it. If the service fails to start successfully check the 'error.log' file in the 'logs' folder under the main NGINX folder.

On the machine you are testing from (which should not be the machine NGINX or your Domino server is running on) the hostname of your website and the ip address which NGINX is listening on should be added to the Windows hosts file - normally found in the C:\Windows\System32\Drivers\etc folder. This will ensure when you type in the address of your website in a browser window the connection will be directed to NGINX.

For example if your hostname is [www.acme.com](http://www.acme.com) and the local ip address NGINX is listening on is 10.0.0.20 then you would add the following entries to the hosts file:

- 10.0.0.20 acme.com
- 10.0.0.20 www.acme.com

Launch a web browser and type in your website address (eg. [www.acme.com](http://www.acme.com)). If everything has been configured correctly the connection will be received by NGINX and then forwarded to your Domino server. The web page should be successfully displayed.

Now test the secure SLL connection by preceding the website name in your web browser with 'https://' (eg. <https://www.acme.com>). If SSL and NGINX have been configured correctly an SSL connection will be successfully established by NGINX, the connection forwarded to the Domino server and the website displayed securely.

To check if content has been return using the http/2 protocol you can check the networking responses using the browser's debugger. In most browsers pressing the F12 key will open the debugging window. Open the networking tab in the debugging window then go to the website address line in the browser and type in the hostname of your site preceded with 'https'.

The following image shows a request for a resource on the AGECOM website over https and it being returned back using http/2:



Request URL: [https://www.agecom.com.au/agecom.nsf/\(Images\)/Logo/\\$File/agecom-6 ...](https://www.agecom.com.au/agecom.nsf/(Images)/Logo/$File/agecom-6 ...)  
Request method: GET  
Status code: ○ 200 OK Edit and Resend Raw headers  
Version: HTTP/2.0

## STEP 10 – UPDATE FORWARDING ON YOUR ROUTER

If your router has been configured to forward incoming traffic on port 80 or 443 (or other custom ports you have specified for incoming http connections) you should update the target ip address to that assigned to NGINX.

The following image shows typical forwarding settings in a router:

ID	Service Port	Internal Port	IP Address	Protocol	Status	Modify
1	1352	1352	192.168.2.30	TCP	Enabled	<a href="#">Modify</a> <a href="#">Delete</a>
2	25	25	192.168.2.30	TCP	Enabled	<a href="#">Modify</a> <a href="#">Delete</a>
3	80	80	192.168.2.20	TCP	Enabled	<a href="#">Modify</a> <a href="#">Delete</a>
4	443	443	192.168.2.20	TCP	Enabled	<a href="#">Modify</a> <a href="#">Delete</a>

We can see that ports 1352 (Notes / Domino) and 25 (SMTP) are forwarded directly to the Domino server's IP address and ports 80 (http) and 443 (https) are forwarded to the ip address assigned to NGINX.

## STEP 11 – FIREWALL SETTINGS

If Windows Firewall, or any other firewall, has been enabled on the server you may need to add an entry for nginx. You should check the firewall settings and if necessary add an entry for nginx.exe to allow incoming connections for the application.

## STEP 12 - FINAL SETUP

Now that you've confirmed your site is correctly working with NGINX you should go to the Windows services on the machine that NGINX was installed on and set the NGINX service to automatically start. We recommend setting it to start Automatically with a Delayed Start:

Network List Service	Identifies th...	Running	Manual	Local Service
Network Location Awareness	Collects an...	Running	Automatic	Network Service
Network Store Interface Service	This service ...	Running	Automatic	Local Service
nginx	nginx	Running	Automatic (Delayed Start)	Local System
Offline Files	The Offline ...	Disabled	Disabled	Local System
Optimize drives	Helps the c...		Manual	Local System

## TIPS

Here's a list of tips to get the most out of using NGINX in your environment.

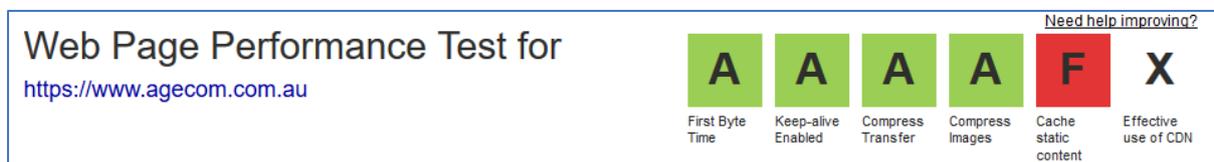
### CHECK YOUR WEBSITE'S PERFORMANCE

You can check the performance of your website and receive tips on how to improve it by going to the following site:

<http://www.webpagetest.org>

You will be asked to type in the address for your website. We recommend performing tests on your website using both the http and https connections.

It will give rankings for various aspects of your websites and tips on how to improve it. The following image shows an example of the various tests and the rankings that were given:



### MULTIPLE WEBSITES

If you are hosting multiple sites you can easily add additional sites to the NGINX configuration file. NGINX can handle all the incoming connections and redirect them to the appropriate server. You'll be able to provide HTTP/2 connectivity for all your domains.

### FORCING ALL CONNECTIONS TO BE SECURE

To force all connections to your website to be secure change the following lines in the sample nginx configuration file (Lines 74 – 76).

From:

```
# Forward traffic to port 80 on the target server or redirect traffic to https
proxy_pass http://www.acme.com:80;
#return 301 https://www.acme.com$request_uri;
```

To:

```
# Forward traffic to port 80 on the target server or redirect traffic to https
#proxy_pass http://www.acme.com:80;
return 301 https://www.acme.com$request_uri;
```

This will redirect incoming connections over port 80 to port 443.

## BRINGING YOUR DOMINO SERVER DOWN FOR MAINTENANCE

If you wish to bring your Domino server down for maintenance NGINX can return a page back to incoming connections indicating the site is down for maintenance.

You should have previously downloaded the 'sample-maintenance.html' file from the [AGECOM Support Note](#) and saved it to the 'html' folder under the main nginx folder.

Rename the sample maintenance file to 'maintenance.html'. NGINX will detect the existence of this file and return the contents of it. When your Domino server is back online simply rename the maintenance file to something like 'zz-maintenance.html' and NGINX will automatically resume directing traffic to your Domino server.

Note: The maintenance.html file should be customized with your website and name and email address before using it.

## SUGGESTIONS OR COMMENTS

Thank you for reading this guide.

If you have some suggestions for improving the content of this guide, find any errors, or just want to drop us a comment please [Contact Us](#) and let us know.

## ABOUT AGECOM

AGECOM have built a reputation for supplying quality products, support and specialist consulting services to the IBM Lotus community globally since 1995. Our applications include:

### [Import for Lotus Notes](#)

Import data directly from the following formats in your Lotus Notes databases: Comma Delimited (CSV), MS-Access, MS-Excel, MS-Outlook, XML, File attachments, Notes Structured Text, and other character delimited files. Existing documents can be updated with imported content. Imports can be run on demand or scheduled.

### [Export for Lotus Notes](#)

Export your Lotus Notes documents and view data to many different file formats including: Comma Delimited (CSV), Fixed Width, HTML, IBM Connections, Microsoft Excel, Microsoft Access, Microsoft Word, PDF, Tab Delimited, XML and more. Exports can be run on demand or scheduled.

### [Agent Monitor for Lotus Domino](#)

Centrally manage, monitor & report on all agents in all databases on your Domino servers (shared, private, scheduled). See when agents are scheduled & by who, view agent logs, missed schedules & agents with errors, change agent settings, enable / disable agents and more. A must for all Domino administrators.

We also have [other applications](#), utilities and software which are freely available to the Lotus community.