

## Test system Build Diary

### **2022 R2 - Create MongoDB Replica Set and Syracuse Cluster**

## Disclaimer

This document is provided "as is" and is for your guidance and educational purposes only. It does not replace the Online documentation, nor is any warranty expressed nor implied for the steps described herein.

## Document Information

Author: Mike Shaw, Sage UK X3 Support Team

## Contents

Test system Build Diary.....	1
Introduction .....	3
2022 R2 - MongoDB Replica Set and Syracuse Cluster build diary.....	4
Objective .....	4
Starting architecture and notes.....	4
Target architecture (Sage X3 2022 R2 aka Version 12 Patch 30).....	5
Documentation to use for planning and execution of this task.....	6
Initial steps .....	7
MongoDB Replica Set .....	8
Syracuse Cluster.....	11
Conclusion.....	20
Appendix A: mzSage-ssl-LBR.conf.....	21

## Introduction

### What is a "Build Diary"

A Build Diary simply describes the steps taken by Sage Support to perform a task or tasks on our internal test systems. Build diaries could be created for major multi-node installations, but may also just be describing the steps taken when installing a small hotfix, or anything in-between.

### Why is this being shared

It may be useful for you to see the steps we have taken to create or implement some feature or installation, as this may highlight "gotcha's", issues encountered or just give you some guidance if you are planning something similar yourself.

You could potentially use these documents as the base for your own "Workplan document" (Described in "Overview of patching X3 and supporting technologies"

<https://www.sagecity.com/gb/sage-x3-uk/b/sage-x3-uk-support-insights/posts/sage-x3-technical-support-tips-and-tricks---march-2021-index> ) when you are planning your own activities

### Target Audience

This document is aimed at Sage X3 Certified Technical consultants. Sage prescribe that X3 system installation, maintenance, migrations, etc. should be performed by suitably qualified Sage X3 consultants. The prerequisite consideration would be for them to have the latest "Sage X3 Certified Technical Consultant" certification. You can read more about the Sage X3 qualifications and requirements in Sage University ( <https://sageu.csod.com/catalog/CustomPage.aspx?id=20000242#tc> )

### Additional things to note

- This document does NOT purport to illustrate "best practice" for the task being described
- The steps described will not necessarily be for a "perfect" task, as there may have been issues that needed to be overcome, worked around, or ignored
- The Sage internal test system has network and hardware configuration specific to Sage
- The Sage internal test system does not necessarily include a Windows Domain and has Sage sandbox specific Windows security setup, so operating system permissions are generally not discussed
- If you intend to use these notes as a guide for your own activities, use with caution and perform your own testing to ensure the described steps are suitable and identify any additional considerations that apply to your own situation
- Ensure you only install and use software you are licensed for

### What does this Build Diary describe?

This build diary primarily describes creating a MongoDB Replica Set and then implementing a Syracuse Cluster.

## 2022 R2 – MongoDB Replica Set and Syracuse Cluster build diary

### Objective

I want to install a MongoDB Replica Set on three separate servers, then install Syracuse on two of these same servers as a Syracuse Cluster.

### Starting architecture and notes

One Windows Server 2019 (Server name: X3ERP12SQLVM)

General software already loaded:

- Windows Server 2019
- SQL Server 2019
- OpenJDK 1.8.0\_332
- Apache 2.4
- Elastic Search 7.16.3

Sage X3 software already installed and configured:

- Application 12.0.30
- Runtime 94.1.17
- AdxAdmin 94.1.17
- Print Server 2.26
- X3 Console 2.54.0.5
- SEED folder

Two Windows Server 2022 (Server names: WEB01 and WEB02)

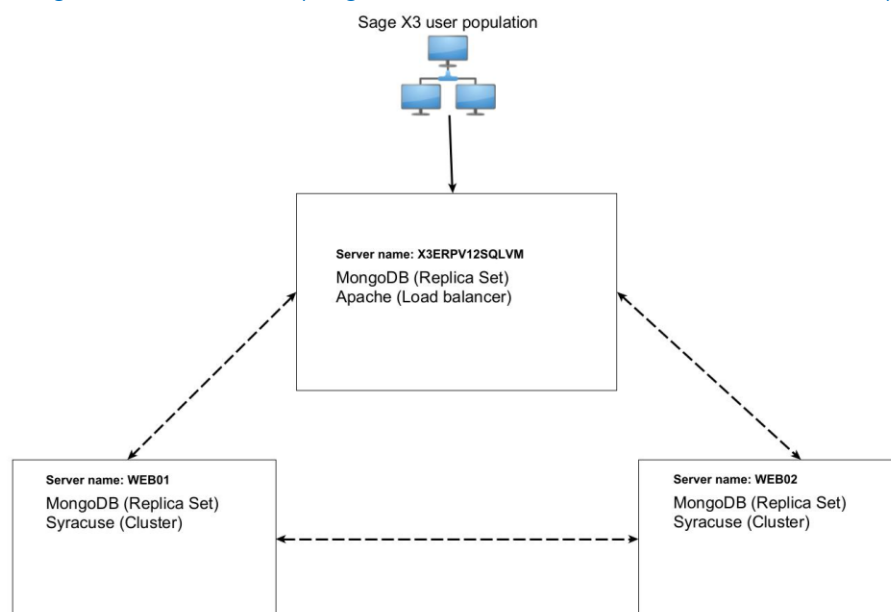
Software already loaded:

- Windows Server 2022
- OpenJDK 1.8.0\_342

All servers have windows users setup as local users:

- “x3admin” for installation and management
- “X3run” for service runtime

## Target architecture (Sage X3 2022 R2 aka Version 12 Patch 30)



## Summary of steps to create MongoDB replica set and Syracuse cluster

- Install MongoDB on X3ERP12SQLVM
- Install MongoDB on WEB01
- Install MongoDB on WEB02
- Create MongoDB replica set
- Install Syracuse on WEB01
- Install Syracuse on WEB02
- Configure Syracuse cluster load balancing
- Configure Apache as an external load balancer (Optional)

Documentation to use for planning and execution of this task

Sage Online documentation

Overall V12 documentation

<http://online-help.sageerpx3.com/erp/12/public/index.html>

Pre-requisites

[http://online-help.sageerpx3.com/erp/12/public/Prerequisites-\(Last-version\).html](http://online-help.sageerpx3.com/erp/12/public/Prerequisites-(Last-version).html)

[http://online-help.sageerpx3.com/erp/12/public/prerequisites\\_overview.html](http://online-help.sageerpx3.com/erp/12/public/prerequisites_overview.html)

Installation documentation

[http://online-help.sageerpx3.com/erp/12/public/getting-started\\_sage-erp-x3-installation-procedure.html](http://online-help.sageerpx3.com/erp/12/public/getting-started_sage-erp-x3-installation-procedure.html)

Sage Knowledgebase articles or Blogs

Which firewall ports need to be open in a multi-node environment

<https://support.na.sage.com/selfservice/viewdocument.do?externalId=102936>

“SSL certificates on Syracuse & MongoDB” presentation available from “Index page: Sage X3 Technical Support Tips and Tricks (September 2021)”

<https://www.sagecity.com/gb/sage-x3-uk/b/sage-x3-uk-support-insights/posts/index-page-sage-x3-technical-support-tips-and-tricks-september-2021>

Test system Build Diary: 2021 R3 (V12 patch 27) MongoDB cluster implementation

<https://www.sagecity.com/gb/sage-x3-uk/b/sage-x3-uk-support-insights/posts/test-system-build-diary-2021-r3-v12-patch-27-mongodb-cluster-implementation>

How do I disable the Internal Load Balancer for Syracuse?

<https://support.na.sage.com/selfservice/viewdocument.do?externalId=108166>

External sites (e.g. Microsoft, etc.)

IANA-managed Reserved Domains <https://www.iana.org/domains/reserved>

*MongoDB*

<https://docs.mongodb.com/v4.4/replication/>

<https://docs.mongodb.com/v4.4/administration/replica-set-deployment/>

<https://docs.mongodb.com/v4.4/tutorial/convert-standalone-to-replica-set/index.html>

## Initial steps

As I am using a non-routable domain (example.com) and don't have a DNS server, so will setup the hosts file on each server

Make note of the IP addresses for the three servers. This hosts information will be merged into each servers existing "hosts" file (located in "C:\Windows\System32\drivers\etc"). I also modify the IP Address to "127.0.0.1" instead of the IP address for the server I'm on.

For example:

```
### Hosts file
10.10.10.59 X3ERP12SQLVM X3ERP12SQLVM.example.com
10.10.10.38 WEB01 WEB01.example.com
10.10.10.43 WEB02 WEB02.example.com
```

Update Firefox, Chrome and Edge browsers to make sure I'm on the latest versions

Copy over the latest Sage Support "Investigation Scripts"

## MongoDB Replica Set

My MongoDB Replica Set installation will be very similar to the steps in “[Test system Build Diary: 2021 R3 \(V12 patch 27\) MongoDB cluster implementation](#)” so will be fairly brief in these notes regarding the MongoDB steps. Refer to the above document to get the details if you need them.

The main differences are that I will use MongoDB 4.4.12 rather than 4.2.2 and will be doing a fresh install of MongoDB on all three servers

1. Create SSL certificates required for the installation and copy to all servers

You can either use the “mzcertificates” or your own scripts to create these required SSL certificates. I am copying these certificates to the “Documents” folder on all three servers.

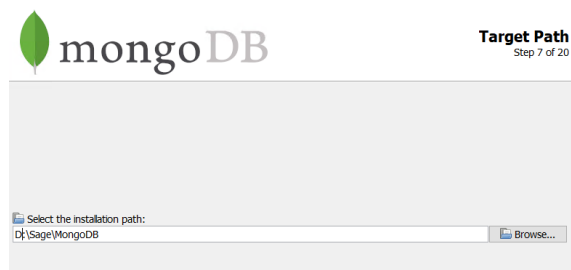
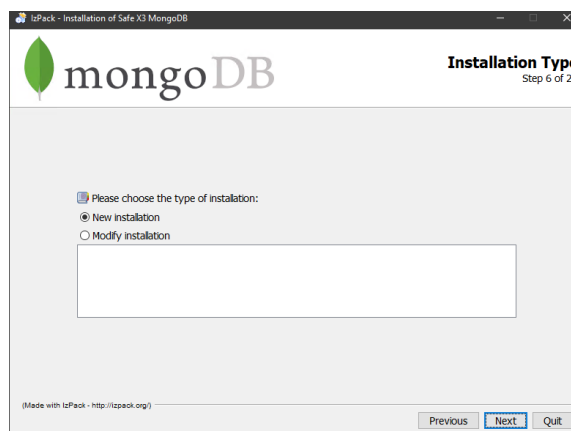
2. Update firewall rules as per “Which firewall ports need to be open in a multi-node environment”

<https://support.na.sage.com/selfservice/viewdocument.do?externalId=102936>

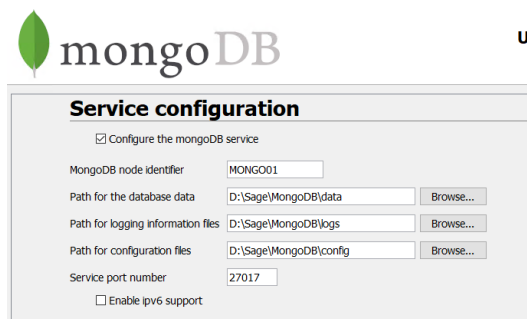
Update each server to allow TCP 27017 (MongoDB) from the other two servers

3. Install MongoDB

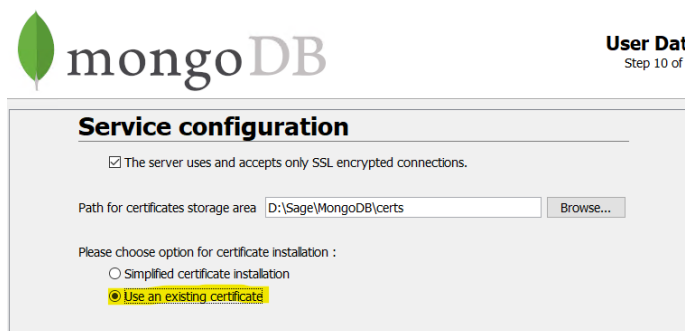
I will first do a standard MongoDB install onto server name X3ERP12SQLVM.







Make sure you use the generated certificates, NOT the simplified certificate installation option.



Pick the certificates matching the specific server name in question.



Once completed take the following steps:

- Shutdown MongoDB service
- Change mongod.conf.  
Add the following lines (note than Mongo is very fussy about spacing and line feeds)  

```
replication:
  replSetName: X3
```

NOTE: the replSetName can be anything you like, but probably best to keep it simple.

```

mongodb.conf x
1 # for documentation of all options, see:
2 # http://docs.mongodb.org/manual/reference/configuration-options/
3
4 systemLog:
5   ... verbosity: 0
6   ... quiet: false
7   ... traceAllExceptions: false
8   ... path: "D:\\Sage\\MongoDB\\logs\\mongodb.log"
9   ... logAppend: true
10  ... logRotate: rename
11  ... destination: file
12  ... timeStampFormat: iso8601-local
13 net:
14  ... bindIpAll: true
15  ... port: 27017
16  ... wireObjectCheck: true
17  ... ipv6: false
18  ... tls:
19    ... mode: requireTLS
20    ... CAFile: "D:\\Sage\\MongoDB\\certs\\ca.cacrt"
21    ... certificateSelector: thumbprint=d5721c5f020fa5a84916170a541eac4b75bc6479
22 security:
23  ... authorization: disabled
24  ... javascriptEnabled: true
25
26 operationProfiling:
27  ... slowOpThresholdMs: 2500
28
29 storage:
30  ... dbPath: "D:\\Sage\\MongoDB\\data"
31  ... journal:
32    ... enabled: true
33    ... directoryPerDB: false
34
35 replication:
36  ... replSetName: X3
37

```

- Restart the MongoDB service

Repeat the above steps for WEB01 and WEB02 servers as well.

#### 4. Configure the MongoDB Replica Set

On server name X3ERP12SQLVM:

- Implement “Investigation Scripts” to use “mzMongoShell.cmd” or use your own favorite Mongo Shell
- Setup the Relica Set
 

```
rs.initiate()
rs.add( { host: "WEB01:27017" } )
rs.add( { host: "WEB02:27017" } )
```
- Check configuration
 

```
rs.conf()
```
- Check status
 

```
rs.status()
```

The MongoDB Replica Set is now operational.

## Syracuse Cluster

I will be installing Syracuse Cluster on two of the Servers, WEB01 and WEB02

1. Update firewall rules as per “Which firewall ports need to be open in a multi-node environment”  
<https://support.na.sage.com/selfservice/viewdocument.do?externalId=102936>

Update both WEB01 and WEB02 servers to allow TCP 8124 (the first Syracuse port) from the other servers.

I also need to update firewall rules on server X3ERP12SQLVM to allow access from both WEB01 and WEB02 for the following ports:

TCP 9200 (Elastic Search)

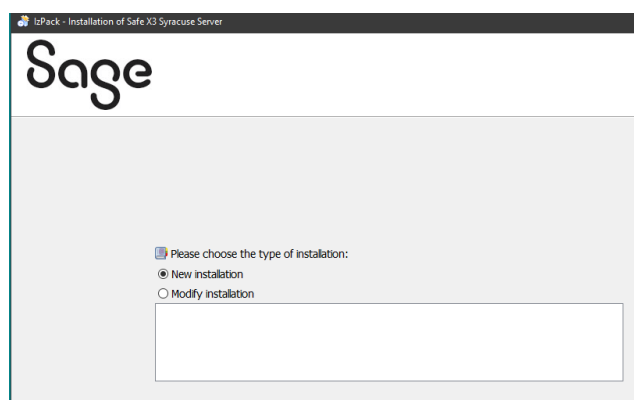
TCP Port 8080 (Apache)

TCP Port 50012 (Runtime)

TCP Port 50001 (Print Server)

2. Install Syracuse on WEB01

I will first do a standard Syracuse install onto server name WEB01. I will be using Syracuse 12.16.0.37 as a new installation.





## Sage

### Syracuse service configuration

(\*) All fields are mandatory.

Syracuse node identifier

Path for logging information files

Http Service port number

Number of Syracuse processes

Number of web services processes

License file

Warning : You must provide an account and a password in order to create a service. This account must be valid and must have the 'log on as a service' right.

Service user

Password for this account

As this is the first server installation for my Syracuse cluster, I will choose the “Simplified certificate installation”

## Sage

### Certificate management

Path for certificates storage area

Please choose option for certificate

☒ Simplified certificate installation for a single server

☐ Use an existing certificate

For the” Mongoddb connection” certificates, I need to use the pre-created certificates I dropped into the “Documents” directory that I used for the MongoDB install.

### Mongodb connection

Please fill in the parameters of the url to connect to MongoDB

Server name (FQDN) or a member of replica set

Service port number

☒ The server uses and accepts only SSL encrypted connections.

(\*) The setup may copy all required files into the directory "D:\Sage\Syracuse\syracuse\certs"

Client certificate file (\*.crt)

Client private key file (\*.key)

CA certificate file

Once completed, I need to copy the “web01.pem” file from WEB01 server directory “D:\Sage\Syracuse\syracuse\certs\_tools\output” to the “D:\Sage\X3ERP12\Runtime\keys” directory on the runtime server (X3ERP12SQLVM)

### 3. Complete the basic X3 installation steps

Although I only have one Syracuse node running, I can login from WEB01 using URL <http://localhost:8124/> and complete the normal installation steps:

- Configure solution
- Check/update X3 folder endpoint
- Perform personalization and menu initialization
- Install supervisor component
- Create SEED folder endpoint

### 4. Install Syracuse on WEB02

The definition of a Syracuse cluster is simply where you have two or more Syracuse nodes installed and using the same MongoDB database. In this case, the Syracuse cluster is automatically configured and is in operation without any additional steps being required.

There are two main problem areas with Syracuse Clusters:

- In order for the multiple Syracuse nodes to trust each other they MUST use certificates generated from the SAME certificate authority.
- The Syracuse nodes communicate with each other via the first defined port number (8124 by default) so this port must be open in the firewall between the Syracuse nodes.

The steps to install the second Syracuse node will therefore be:

- Create certificates on the WEB01 server and copy over to WEB02

Launch the script “certgen.bat” from directory from “D:\Sage\Syracuse\syracuse\certs\_tools” directory. Select option 1 to create a new set of certificate files for WEB02

```

C:\Windows\system32\cmd.exe

D:\Sage\Syracuse\syracuse\certs_tools>java -jar certgen.jar
Read certificate output/ca.cacrt

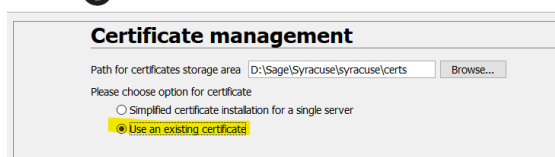
Which task do you want to perform?
(1) Create new certificate and private key
(2) Renew the validity of the certificate
(3) Renew the validity of all certificates
(4) Generate new private key and create certificate with same subject as before
(5) Change the subject of the certificate
(6) Show certificate data
(7) Show certificate data of all certificates
(8) Delete certificate and private key for the named server
(9) Transfer certificate and private key to the named server
(10) End
Please enter the number of the option: 1

Task: Create new certificate and private key
Name of server: WEB02
Server name for TCP connections [web02]:
Enter days of validity (when input is empty, validity will be until 25 July 2032 14:44:39 BST):
Enter passphrase for new private key:
Confirm passphrase of private key:
Enter passphrase of private key of CA certificate:
Read private key private/ca.cakey ...
Port of Syracuse server 'web02' if data should be transferred (no transfer for value 0) [0]:
Generate key pair ...
Generate certificate ...
Write private key output/web02.key ...
Write certificate output/web02.crt ...
Write public key output/web02.pem ...
Finished

Which task do you want to perform?
(1) Create new certificate and private key
(2) Renew the validity of the certificate
(3) Renew the validity of all certificates
(4) Generate new private key and create certificate with same subject as before
(5) Change the subject of the certificate
(6) Show certificate data
(7) Show certificate data of all certificates
(8) Delete certificate and private key for the named server
(9) Transfer certificate and private key to the named server
(10) End
Please enter the number of the option: 10_

```

- On WEB01 from the “D:\Sage\Syracuse\syracuse\certs\_tools\output” directory:
  - o copy the ca.cacrt, web02.crt and webv02.key files to WEB02 into a temporary directory
  - o copy web02.pem to “D:\Sage\X3ERP12\Runtime\keys” directory on the runtime server (X3ERP12SQLVM)
- On WEB02
  - o Run a normal install of Syracuse on WEB02 with the only difference being to use EXISTING certificates when prompted



### Certificate setup

(\*) The setup may copy all required files into the directory "D:\Sage\Syracuse\syracuse\certs"

Certificate file:

Private key file:

Passphrase of the private key:

CA certificate file:

☐ Specify an X3 Runtime path and install certificates

☐ Specify an X3 webservices server path and install certificates

Just to be clear, when selecting the “Mongodb connection” certificates, I still need to pick the pre-created certificates I dropped into the “Documents” directory that I used for the MongoDB install.

### Mongodb connection

Please fill in the parameters of the url to connect to MongoDB

Server name (FQDN) or a member of replica set:

Service port number:

☒ The server uses and accepts only SSL encrypted connections.

(\*) The setup may copy all required files into the directory "D:\Sage\Syracuse\syracuse\certs"

Client certificate file (\*.crt):

Client private key file (\*.key):

CA certificate file:

## 5. Check the results

You should see both Syracuse nodes as “OK” when connected to each node. So here I connect to WEB02 using <http://localhost:8124/> then navigate to Administration, Administration, Servers, Hosts and can see both nodes as expected. I should see the same screen when connected to WEB01 as well.

Sage   X3								
All > Administration > Administration > Servers								
Hosts								
Host name	Number of child processes	Deactivated	Started	Status	Security	Code version	TCP host name	
WEB01	2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	OK	<input checked="" type="checkbox"/>	12.16.0.37-0	web01	
WEB02	2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	OK	<input checked="" type="checkbox"/>	12.16.0.37-0	web02	

Drill into each node in turn and check there are no “Missing certificates”, “Missing CA certificates” or “Untrusted hosts”

## Host WEB01

### Information Configuration

#### Information

Host name	Deactivated	Started	Status
WEB01	✗	✓	OK

#### Configuration

##### Connections

Port	Active	SSL	Client authentication	Server certificate	Client certificate
8124	✓	✗	✗		
<div> <div>Number of child processes</div> <div>2</div> <div>Number of web service child processes</div> <div>1</div> <div>Code version</div> <div>12.16.0.37-0</div> <div>TCP host name</div> <div>web01</div> </div> <div> <div>PID</div> <div>3024</div> <div>Respawn limit</div> <div>10</div> <div>Return request timeout</div> <div>30</div> <div>Missing CA certificates</div> <div>Untrusted hosts</div> <div>Child process information</div> <div>Security</div> <div>Respawn time</div> <div>120</div> </div>					

## 6. Next steps

The first port used by Syracuse should be reserved for only the Syracuse management purposes. (This is true even for a single node) I am going to setup a second port for both Syracuse nodes using HTTPS (aka SSL). For this initial testing I will just use the internal certificate for my SSL port. Whilst this is quick and convenient, this certificate should not be used for either LIVE nor TEST servers when you let real users on, as it introduces a security risk.

In hosts, add port 443 as an SSL port for both WEB01 and WEB02, selecting the appropriate server certificate for each one. NOTE: this will automatically restart Syracuse when you save this change.

#### Configuration

##### Connections

Port	Active	SSL	Client authentication	Server certificate	Client certificate
8124	✓	✗	✗		
443	✓	✓	✗	web01	?

Test you can now connect to both WEB01 and WEB02 using the URL <https://localhost/>

## 7. Load balancing

- Load balancing by Syracuse (default setting)

By default, when you login to either Syracuse node, they will load balance users to either of the nodes. i.e. if a user connects to WEB01 their session could be redirected automatically to WEB02 and vice-versa.

To test the current behavior, login several users from different browsers or different PCs to check the load balancing.

To see which session is where, navigate to Administration, Usage, Sessions management, Session information.

With three users connected from WEB02, I can see connections directed to WEB01 and WEB02. Also note both N0 and N1 are being used on WEB02. This is the expected result.



## Web client sessions

▼	Type	Session ID	User login	Status	Web client	Host	Port	Process name	Process ID	Endpoint	X3 login	X3 sessions	Badge	Last user access	Expired at
▼	Standard	64ea9c1d8b0	admin	Authenticated	127.0.0.1	WEB02	8124	N1	5048	SEED	admin	1	ERPDEV	2022-07-28T15:42:28.607Z	2022-07-28T16:02:28.607Z
▼	Standard	0796e6a36a48	admin	Authenticated	□1	WEB02	8124	N0	3828	SEED	admin	1	ERPDEV	2022-07-28T15:46:55.524Z	2022-07-28T16:06:55.524Z
▼	Standard	015293cae41a	admin	Authenticated	□1	WEB01	8124	N1	3564	SEED	admin	1	ERPDEV	2022-07-28T15:47:53.434Z	2022-07-28T16:07:53.434Z

If I then launch a fourth user session, this gets directed to WEB01 N0 process, which is again as I would expect.

## Web client sessions

▼	Type	Session ID	User login	Status	Web client	Host	Port	Process name	Process ID	Endpoint	X3 login	X3 sessions	Badge	Last user access	Expired at
▼	Standard	64ea9c1d8b0	admin	Authenticated	127.0.0.1	WEB02	8124	N1	5048	SEED	admin	1	ERPDEV	2022-07-28T15:49:41.272Z	2022-07-28T16:09:41.272Z
▼	Standard	0796e6a36a48	admin	Authenticated	□1	WEB02	8124	N0	3828	SEED	admin	1	ERPDEV	2022-07-28T15:46:55.524Z	2022-07-28T16:06:55.524Z
▼	Standard	015293cae41a	admin	Authenticated	□1	WEB01	8124	N1	3564	SEED	admin	1	ERPDEV	2022-07-28T15:51:24.913Z	2022-07-28T16:11:24.913Z
▼	Standard	0fb643e0c221	admin	Authenticated	□1	WEB01	8124	N0	5940	SEED	admin	1	ERPDEV	2022-07-28T15:50:23.988Z	2022-07-28T16:10:23.988Z

- Load balancing with external load balancer

You may wish to have an external load balancer do the load balancing between the Syracuse nodes. In this case you can disable the Syracuse layer load balancing, otherwise you will be load balancing twice! (It's your choice...)

To disable Syracuse load balancing, you need to manually update the "nodelocal.js" file in the directory "D:\Sage\Syracuse\syracuse\bin" This is discussed in KB article "How do I disable the Internal Load Balancer for Syracuse?" (

<https://support.na.sage.com/selfservice/viewdocument.do?externalId=108166>)

Edit nodelocal.js to add the following section:

```
hosting: {
  localBalancer: true
}
```

```
1 exports.config = {
2   port: 8124,
3   streamlineFromCI: true,
4   x3key: true,
5   collaboration: {
6     driver: "mongodb",
7     dataset: "syracuse",
8     hostname: "web02",
9     port: 27017,
10    logpath: "D:\\Sage\\Syracuse\\syracuse\\logs",
11    certDir: "D:\\Sage\\Syracuse\\syracuse\\certs",
12    cacheDir: "D:\\Sage\\Syracuse\\syracuse\\cache"
13  },
14  mongodb: {
15    options: {
16      writeConcern: { w: 1 },
17      ssl: true,
18      sslCA: require('fs').readFileSync("D:\\Sage\\Syracuse\\syracuse\\certs\\mongodb\\ca.crt"),
19      sslValidate: true,
20      sslKey: require('fs').readFileSync("D:\\Sage\\Syracuse\\syracuse\\certs\\mongodb\\client.pem"),
21      sslCert: require('fs').readFileSync("D:\\Sage\\Syracuse\\syracuse\\certs\\mongodb\\client.pem")
22    }
23  },
24  hosting: {
25    localBalancer: true
26  },
27  session: {
28    timeout: 20, // minutes
29    checkInterval: 60, // seconds
30    auth: "basic"
31  },
32  searchEngine: {
33    hostname: "X3ERP\\129Q\\N1",
34    port: 9200,
35  },
36  x3fusion: {
37    records: {
38      dumpPath: "D:\\Sage\\Syracuse\\syracuse\\cache\\_crg_\\USER\\NAME_"
39    }
40  }
41 }
```

You need to manually restart Syracuse service for this change to be picked up.

If I now do the same test with four users connecting to WEB02 I see that all users now connect to WEB02. They are still load balanced between the N0 and N1 processes but only on the local server.

All > Administration > Usage > Sessions management

**Web client sessions** Action

Type	Session ID	User login	Status	Web client	Host	Port	Process name	Process ID	Endpoint	X3 login	X3 sessions	Badge	Last user access	Expired at
Standard	20805e12722c	admin	Authenticated	127.0.0.1	WEB02	443	N1	4444	SEED	admin	1	ERPDEV	2022-08-01T12:24:15.324Z	2022-08-01T12:44:15.324Z
Standard	76845775a35	admin	Authenticated	□1	WEB02	443	N1	4444	SEED	admin	1	ERPDEV	2022-08-01T12:23:52.752Z	2022-08-01T12:43:52.752Z
Standard	ac18520f7ca9	admin	Authenticated	□1	WEB02	8124	N1	4444	SEED	admin	1	ERPDEV	2022-08-01T12:24:02.701Z	2022-08-01T12:44:02.701Z
Standard	a628ae3a83a9	admin	Authenticated	□1	WEB02	443	N0	4984	SEED	admin	0	ERPDEV	2022-08-01T12:24:22.224Z	2022-08-01T12:44:22.224Z

## - Setup External load balancer

For “proper” testing you should of course configure your TEST environment with the same load balancing solution as you will use for LIVE, but if this is not an option for you immediately there are a few alternatives you could consider.

In my situation, I don’t have an external load balancer available at this stage. Given my current setup, the easiest option is for me to setup the Apache listener already installed for X3 and add configuration to act as an “external” software load balancer which I will describe below. Even if you are not using Apache with your own X3 instance, it is simple to setup and configure Apache for load balancing so could still be used if you have no other options immediately available.

**BEWARE:** doing this is not a recommended nor the most secure architecture, so should only be used for internal testing only.

### a. Create certificates for Apache

I wish to use HTTPS to connect to my Apache Load Balancer, so need some SSL certificates. As it happens, I already have some SSL certificated created for my MongoDB install, so can re-use these.

- Launch CMD prompt and navigate to directory “D:\SageSupport\mzcertificates\Mongo\certs”
- Run the following command:  
D:\OpenSSL\bin\openssl rsa -in X3ERPv12SQLVM.key -out apache\_X3ERPv12SQLVM.key  
Enter the appropriate passphrase when prompted

b. Create directory “D:\Sage\FreeComponents\Apache24\certs” and copy the X3ERPv12SQLVM.crt, apache\_X3ERPv12SQLVM.key and ca.cacrt certificates to this directory

c. Copy “mzSage-ssl-LBR.conf” to “D:\Sage\FreeComponents\Apache24\conf” and edit this file as needed. NOTE: this file is included in Appendix A of this document

d. Edit file “D:\Sage\FreeComponents\Apache24\conf\httpd.conf” as below:

Add the line below to the end of the file

```
include conf/mzSage-ssl-LBR.conf
```

Uncomment the following lines:

```
LoadModule access_compat_module modules/mod_access_compat.so
LoadModule headers_module modules/mod_headers.so
LoadModule lbmethod_byrequests_module modules/mod_lbmethod_byrequests.so
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule slotmem_shm_module modules/mod_slotmem_shm.so
LoadModule socache_shmcb_module modules/mod_socache_shmcb.so
LoadModule ssl_module modules/mod_ssl.so
```

Save these changes then re-start the Apache service

- Update the Firewall rules
  - o Update both WEB01 and WEB02 servers to allow TCP 443 from X3ERP12SQLVM
  - o Update server X3ERP12SQLVM to allow TCP 443 for your users
- Test you can login and use X3 when using the Apache load balancer URL  
<https://X3ERP12SQLVM/>

All > Administration > Usage > Sessions management

**Web client sessions** Acti

12 Results Display: 20

Type	Session ID	User login	Status	Host	Port	Process name	Process ID	Endpoint	X3 login	X3 sessions	Badge	Last user access	Expired at
Standard	05322f7ba6a3	admin	Authenticated	WEB01	443	N0	4948	SEED	admin	1	ERPDEV	2022-08-01T15:40:00.431Z	2022-08-01T16
Standard	044d4ed14072	admin	Authenticated	WEB01	443	N1	5112	SEED	admin	1	ERPDEV	2022-08-01T15:33:01.625Z	2022-08-01T15
Standard	86babeabbc09	admin	Authenticated	WEB01	443	N1	5112	SEED	admin	1	ERPDEV	2022-08-01T15:33:42.299Z	2022-08-01T15
Standard	ead3ab0b7430	admin	Authenticated	WEB01	443	N0	4948	SEED	admin	1	ERPDEV	2022-08-01T15:35:28.767Z	2022-08-01T15
Standard	44329ee1f894	admin	Authenticated	WEB01	443	N1	5112	SEED	admin	0		2022-08-01T15:36:46.943Z	2022-08-01T15
Standard	d1c11dd686c5	admin	Authenticated	WEB01	443	N1	5112	SEED	admin	1	ERPDEV	2022-08-01T15:37:30.394Z	2022-08-01T15
Standard	cc16bd4ad51	admin	Authenticated	WEB02	443	N1	5104	SEED	admin	1	ERPDEV	2022-08-01T15:32:11.873Z	2022-08-01T15
Standard	f2dc6c75be94	admin	Authenticated	WEB02	443	N1	5104	SEED	admin	1	ERPDEV	2022-08-01T15:40:10.400Z	2022-08-01T16
Standard	a951899728ea	admin	Authenticated	WEB02	443	N0	4448	SEED	admin	1	ERPDEV	2022-08-01T15:33:37.073Z	2022-08-01T15
Standard	cef063cfe492	admin	Authenticated	WEB02	443	N0	4448	SEED	admin	1	ERPDEV	2022-08-01T15:35:16.174Z	2022-08-01T15
Standard	a5a5b7ed828c	admin	Authenticated	WEB02	443	N1	5104	SEED	admin	0		2022-08-01T15:36:23.808Z	2022-08-01T15
Standard	d7db02ed43ca	admin	Authenticated	WEB02	443	N0	4448	SEED	admin	0		2022-08-01T15:37:02.612Z	2022-08-01T15

In this case we can see the 12 connections balanced equally across WEB01 and WEB02 as expected.

## Conclusion

This build diary has described creating a MongoDB Replica Set and then implementing a Syracuse Cluster, followed by setting up an external load balancer for testing purposes.

## Appendix A: mzSage-ssl-LBR.conf

```
#####
####          Configure Apache as a SSL Load Balancer          #####
####          Beginning of block that may already be defined elsewhere      #####
####          (in httpd.conf for example)                                #####
SSLCipherSuite HIGH:MEDIUM:!MD5:!RC4
SSLProxyCipherSuite HIGH:MEDIUM:!MD5:!RC4
SSLHonorCipherOrder on
SSLProtocol all -SSLv3
SSLPassPhraseDialog builtin
SSLSessionCache "shmcb:D:/Sage/FreeComponents/Apache24/logs/ssl_scache(512000)"
SSLSessionCacheTimeout 300
####          End of block that may already be defined elsewhere      #####
#####
Listen 443
<VirtualHost *:443>
    DocumentRoot "D:/Sage/FreeComponents/Apache24"
    ServerName X3ERPv12SQLVM
    ServerAdmin admin@example.com
    ErrorLog "logs\LBR_ssl_error.log"
    TransferLog "logs\LBR_ssl_access.log"
    SSLEngine on
    SSLCertificateFile "certs/X3ERPv12SQLVM.crt"
    SSLCertificateKeyFile "certs/apache_X3ERPv12SQLVM.key"
    SSLCACertificateFile "certs/ca.cacrt"
    <FilesMatch "\.(cgi|shtml|phtml|php)$">
        SSLOptions +StdEnvVars
    </FilesMatch>
    <Directory "D:/Sage/FreeComponents/Apache24/cgi-bin">
        SSLOptions +StdEnvVars
    </Directory>
    BrowserMatch "MSIE [2-5]" \
        nokeepalive ssl-unclean-shutdown \
        downgrade-1.0 force-response-1.0
    CustomLog "logs\LBR_ssl_request.log" "%h %l %u %t %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\"
%>s %b %T %D"
    ### add in the proxy
    SSLProxyProtocol all -SSLv3
    ProxyRequests on
    ProxyErrorOverride On
    ProxyPreserveHost On
    SSLProxyEngine On
    # Need the CA certificate for the backend server(s)
    ##SSLProxyCACertificateFile "certs/remoteca.cacrt"
    ## In this case, the CA is actually the same
    SSLProxyCACertificateFile "certs/ca.cacrt"
    ##### Quick hack (if needed) for dealing with invalid SSL Certs on the backend
server(s)
    ##### ONLY EVER DO THIS FOR A TEST INSTANCE!!
    SSLProxyCheckPeerName off
    SSLProxyCheckPeerExpire off
    SSLProxyVerify none
    ##### End of quick hack
    Header add Set-Cookie "ROUTEID=session.%(BALANCER_WORKER_ROUTE)e; path=/"
env=BALANCER_ROUTE_CHANGED
    <Proxy "balancer://myLbr">
        Require all granted
        BalancerMember "https://web01" route=WEB01
        BalancerMember "https://web02" route=WEB02
        ProxySet lbmethod=byrequests stickysession=ROUTEID
    </Proxy>
    <Location "/">
        ## Retain SSL traffic through to backend server
        ProxyPass "balancer://myLbr/"
        ProxyPassReverse "balancer://myLbr/"
    </Location>
</VirtualHost>
#####
```