



Axcient

Architecting a High Availability Private
Cloud Environment

axcient.com

Contents

Contents	2
Introduction	4
Architectural Overview	4
Scaling Parameters Summary and Examples	6
Basic Server Requirements.....	8
Advanced Server Requirements	8
Anchor Server Requirements.....	8
Apache Server Requirements	9
PostgreSQL Server Requirements	9
Bandwidth Requirements.....	10
Storage Requirements	10
Load Balancing and Networking Recommendations.....	12
Setting Up a High Availability Configuration	13
Configuration Examples.....	13
Requirements	14
Getting Started Checklist.....	14
Configuration Instructions.....	14
Step 1: Install Anchor and PostgreSQL on Machine 1.....	14
Step 2: Install Anchor and Apache on Machine 2	20
Step 3: Install Anchor on Machine 3	21
Option 1: Configuring a Locally Mounted Storage Drive	24
Configuration Instructions.....	24
Step 1: Map Drives	24
Step 2: Register Anchor	27
Step 3: Modify Anchor Configuration Files.....	29
Option 2: Configuring a NAS, SAN, or Other Storage Device	32

Configuration Instructions.....	32
Step 1: Create Local Admin Accounts on Each Machine.....	32
Step 2: Register Anchor	32
Step 3: Copy Metadata	34
Step 4: Change Services to Run as Local Admin	34
Step 5: Modify Anchor Configuration Files.....	34
Step 6: Update PostgreSQL with the UNC Path	36
Configuring Backups	40
Reviewing Mappings for the Load Balancer or DNS Server	41
Configuring Dual Hostname Settings	42
Configuration Instructions.....	43
Step 1: Configure a Second Domain or Subdomain	43
(Optional) Step 1a: Configure an Additional IP Address.....	43
(Optional) Step 1b: Stop the Apache and Anchor Services	46
Step 2: Point the New Domain or Subdomain to the new IP Address.....	47
Step 3: Configure the Port Address Translation (PAT) Settings	47
Step 4: Configure the SSL Certificates.....	47
Step 5: Update the Apache Configuration File.....	48
Step 6: Specify the App Server Hostname within the Web Portal.....	50
Finding Additional Support	52

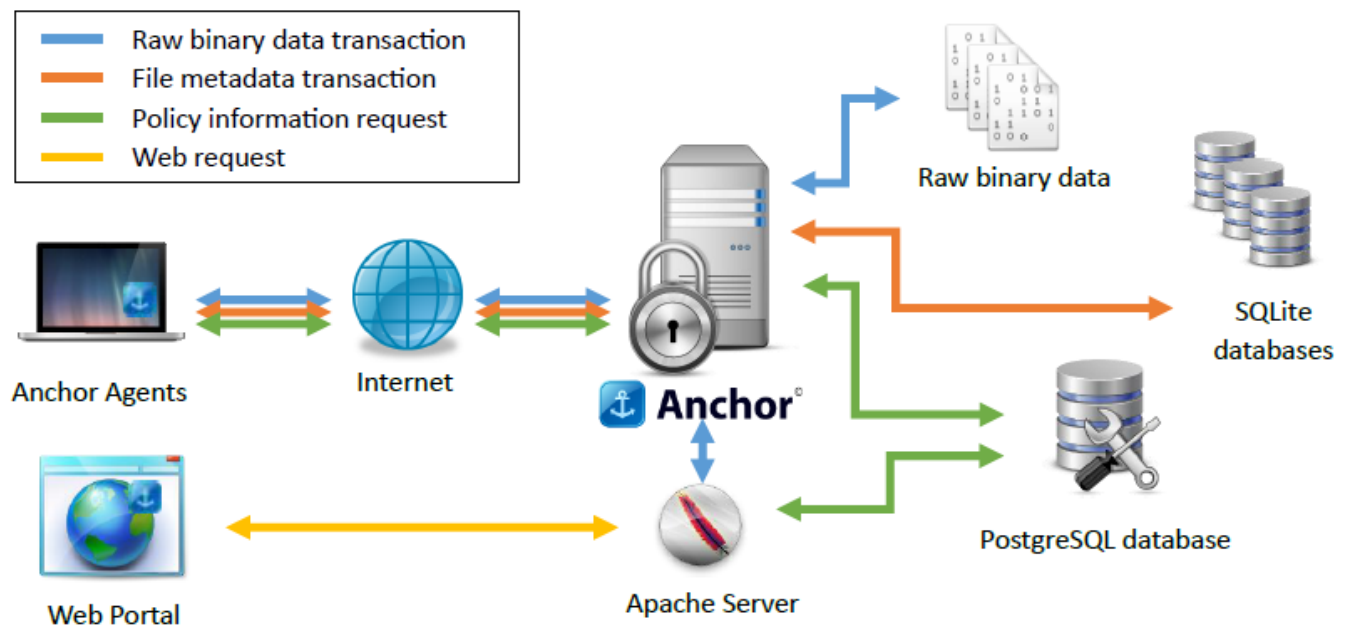
Introduction

Anchor is a software platform that enables real-time file synchronization, sharing, collaboration, and backup for businesses. With a focus on security, reliability, control, and integration, Anchor provides IT professionals with a solution to the universal access, sharing, and file-recovery problems that plague organizations.

The platform is comprised of two services: server and agent. The server service can be hosted within a company's internal infrastructure (private cloud) or can be provided by Anchor's hosted cloud infrastructure (SaaS). This guide will help you plan and implement an Anchor private cloud environment according to established guidelines and best practices.

Architectural Overview

The Anchor software architecture is comprised of various components:



- **Anchor Agents and Apps**—software applications for Windows and OSX, and mobile apps for iOS, Android, and Windows Mobile. These agents and apps communicate with both the Apache Server and Anchor Server.
- **Web Portal**—an HTML5/Javascript frontend that communicates with the API on the Apache Server.
- **Anchor Servers**—one or more virtual or physical servers running the proprietary Anchor backend server software on Windows or Linux (recommended for high scale)

deployments). Each agent will keep one persistent connection open to one Anchor Server. If multiple servers are configured, a cluster is formed, allowing agents to connect to any cluster member to service any request, and allowing instant yet efficient notification of file changes to any connected agents. The Anchor server is dependent on the PostgreSQL database, SQLite databases, and the raw binary data stores.

- **Apache Server**—a virtual or physical server running Apache and the proprietary Anchor web backend application on Windows or Linux (recommended for high scale deployments), and is configured to use your custom SSL certificate. The Apache server depends upon the Anchor servers, PostgreSQL, and the SQLite databases.
- **PostgreSQL DB**—stores policy and account information. It does not store information about individual files, and therefore remains relatively small, but critical. We recommend a high availability PostgreSQL setup with master/slave replication, as the Anchor service will not operate if PostgreSQL is down. *You must back up the PostgreSQL database.*
- **SQLite databases** (root stores)—stores information about each revision of a synchronized file, with one SQLite database per top-level folder (root) synchronized. One or multiple mount points are supported. For Anchor private cloud deployments across more than one server, the SQLite databases must be accessible on a network addressable filesystem that can be accessed concurrently by multiple systems; this filesystem must also support proper file locking, such as CIFS or NFSv4. *You must back up the SQLite databases.*
- **Raw binary data** (file stores)—one or more filesystem mount points that will store the actual encrypted binary data for each file revision. For Anchor private cloud deployments across more than one server, the file stores must be accessible on a network filesystem, such as CIFS or NFSv4. A subdirectory structure will automatically be created so that any individual directory will only have a maximum of a few thousand files in it. *You must back up the file stores.*

Only the Anchor servers and Apache server needs to be exposed to the Internet through your firewall/router or load balancer. All external communication is over well-defined TCP ports. Public DNS entries are needed for the Anchor and Apache servers. In many cases, round robin DNS with a short TTL is sufficient for load balancing, or any load balancer that supports TCP (for example, HAProxy, F5).

This document gives general guidelines for infrastructure, based on observed average resource requirements for at-scale systems. Your actual resource requirements might vary based on usage patterns. It is typically observed that for every 1000 users provisioned in the system, there are 700 active connections.

Scaling Parameters Summary and Examples

For every 1400 provisioned users, you will need the following resources (it is usually observed that for every 1400 provisioned users, there are 1000 active connections).

Resource	1400 Users
Anchor Server vCPU	2.5
Anchor Server vRAM (GB)	5
Apache Server vCPU	0.25
Apache Server vRAM (GB)	1
PostgreSQL vCPU	0.25
PostgreSQL vRAM (GB)	0.5
Bandwidth (Mbit/sec)	35

For every 1000 provisioned users, you will need the following storage resources:

Resource	1,000 Users
PostgreSQL storage (GB) SSD recommended	5
Root stores storage (GB) SSD storage <i>required</i>	4

Root stores avg read IOPS	150
Root stores max read IOPS (burst)	300
Root stores avg write IOPS	80
Root stores max write IOPS (burst)	200
File stores storage (GB) Storage averages will vary greatly per use case; 7200 rpm storage is sufficient	15,000 (15 TB)

The following table estimates the resources required to support a given number of *provisioned* users:

Resource	1,000 Users	5,000 Users	10,000 Users	25,000 Users	50,000 Users	100,000 Users
Anchor Server vCPU	1.8	8.8	17.5	43.8	87.5	175.0
Anchor Server vRAM (GB)	3.5	17.5	35.0	87.5	175.0	350.0
Apache Server vCPU	0.2	0.9	1.8	4.4	8.8	17.5
Apache Server vRAM (GB)	0.7	3.5	7.0	17.5	35.0	70.0
PostgreSQL vCPU	0.2	0.9	1.8	4.4	8.8	17.5
PostgreSQL vRAM (GB)	0.4	1.8	3.5	8.8	17.5	35.0
PostgreSQL storage (GB)	5	25	50	125	250	500
Root stores storage (GB)	4	20	40	100	200	400

Root stores avg read IOPS	150	750	1,500	3,750	7,500	15,000
Root stores max read IOPS	300	1,500	3,000	7,500	15,000	30,000
Root stores avg write IOPS	80	400	800	2,000	4,000	8,000
Root stores max write IOPS	200	1,000	2,000	5,000	10,000	20,000
File stores storage (GB)	15,000	75,000	150,000	375,000	750,000	1,500,000
WAN bandwidth (Mbit/sec)	25	123	245	613	1,225	2,450
WAN outbound GB/month	3,500	17,500	35,000	87,500	175,000	350,000
WAN inbound GB/month	1,750	8,750	17,500	43,750	87,500	175,000

In addition to the estimated resources above, it is important to consider high availability requirements—as well as the scalability of the network file storage—for your root stores and file stores.

Basic Server Requirements

By default, the Anchor Server Installer installs a single Anchor, Apache, and database server on one machine (physical or virtual). The basic recommended server requirements for this type of environment can be found in the [Anchor Architecture Overview](#) Knowledgebase article. The recommended system requirements in this article are known to support 5,000 simultaneous connections. A single server can scale up to much higher simultaneous connection counts, as per guidelines given in this document. Additionally, the Anchor servers can scale out to as many servers as needed to meet the load.

Advanced Server Requirements

For an advanced rollout—intended to support a high number of users—it is recommended that you utilize a high availability configuration and scale out as needed.

Anchor Server Requirements

- Any physical or virtual x86 server that is compatible with Debian amd64 Linux

- Requires at least 4 CPU cores and 6 GB of RAM
- Gigabit NICs are required; 10Gbit is recommended
- 2.5 vCPUs (CPU cores) and 5 GB vRAM per 1000 active connections (for example, a dual-socket Xeon E5-2670v3 system with 64GB RAM would support 10,000 active connections)
- For high availability, start with two servers and scale out as needed (for example, an environment that intends to support 300,000 active connections might use 30 Anchor servers)

Apache Server Requirements

- Any physical or virtual x86 server that is compatible with Debian amd64 Linux
- Requires at least 4 CPU cores and 6 GB of RAM
- 0.25 vCPUs and 1 GB vRAM per 1000 active connections

PostgreSQL Server Requirements

- Any physical or virtual x86 server that is compatible with Debian amd64 Linux
- Requires at least 4 CPU cores and 6 GB of RAM
- 0.25 vCPUs and 0.5 GB vRAM per 1000 active connections
- 5 GB storage per 1000 users (SSD or 15K RPM or 10K RPM storage is recommended)
- Two PostgreSQL servers (master and slave) running in a cluster, each having an independent IP address; an additional cluster IP address will only be used by the master node. If the cluster fails over, then the slave becomes the master and takes over the cluster IP address. The Anchor app and Web app utilize the cluster IP when talking to PostgreSQL.



Note: You must *frequently* back up your PostgreSQL database.

For more information on PostgreSQL high availability, please reference the following articles:

- [PostgreSQL High Availability](#)
- [Determining IO Capacity using SQLIO Disk Subsystem Benchmark Tool](#)
- [Failover Clustering Hardware Requirements](#)
- [Deploying a Hyper-V Cluster](#)

Bandwidth Requirements

Bandwidth usage will vary based on synchronization and collaboration patterns, but typically follows the scaling parameters as described below.



Note: The numbers given here assume that LAN Sync is turned off, and that each agent endpoint is in a different LAN network (worst case scenario). Please note that Anchor's LAN Sync feature should be turned on to allow multiple agents in the same LAN to pull changes from each other, rather than across the Internet from the Anchor servers. Turning on LAN Sync will significantly reduce bandwidth requirements.

Each month, every active connection will use about 5 GB of outbound bandwidth and 2.5 GB of inbound bandwidth. We estimate that for every 1000 active connections, you will need approximately 35 Mbit/sec of available Internet bandwidth across your edge router/firewall.

Internet network traffic between cluster members is relatively minimal, except for any network traffic relating to the network-accessible root stores and file stores, which can be significant (see storage requirements section for details).

Storage Requirements

In an Anchor private cloud system with multiple Anchor servers, you must use network addressable storage mount points for both root stores and file stores that support file locking and concurrent access (e.g., CIFS or NFSv4).



Note: The Apache server only needs *read* access to root stores and does not need access to file stores. This helps minimize file locking traffic caused by Apache servers accessing the root stores.

Root Store Recommendations

- SSD storage is *required* for proper performance for systems with 1000 users or more
- The system can distribute root store DBs across multiple root store mount points
- Characteristics per 1000 provisioned users:
 - Average sqlite DB file is 1.5 MB
 - 4 GB data

- 150 read IOPS (average), 300 read IOPS (burst)
- 80 write IOPS (average), 200 write IOPS (burst)
- Any POSIX network filesystem with file locking is supported (for example, CIFS, NFSv4)
- Possible systems for small or non-HA setups:
 - Solaris/Linux/BSD/FreeNAS ZFS over SSDs, exported via NFSv4
 - Windows hardware RAID6 over SSDs, exported via CIFS
- Possible systems for larger or HA setups:
 - Scale-out NAS/SAN that expose filesystem storage over CIFS or NFSv4
 - Software-defined storage, such as Scality



Note: SQLite test scripts can be requested to ensure your desired storage is compatible.



Note: You must *frequently* back up your root store data.

File Store Recommendations

- 7200 rpm storage is sufficient
- The system can distribute file data across multiple file store mount points
- Characteristics per 1000 provisioned users:
 - Average file is 0.8 MB
 - 15,000 GB (15 TB) data and 18,750 files per 1000 users
- RAID6 (or equivalent high levels of redundancy) and a storage subsystem with proper on-demand and periodic integrity checking and self-healing for the data
- Any POSIX network filesystem is supported (file locking is not required)
 - Typically, network storage systems are used that can provide CIFS or NFSv4 storage at scale



Note: You must *frequently* back up your file store data.

Load Balancing and Networking Recommendations

- Both the Anchor and Apache servers use the TLS (SSL) protocol to encrypt transmissions
- An SSL/TLS offloader is *not* required
- Edge routing and firewall equipment must support the number of concurrent TCP connections matching the number of users; it must also support at least 40 Mbit/sec of throughput per 1000 users



Note: Edge routing and firewall equipment should be configured in high availability clusters to ensure proper uptime of the service.

Setting Up a High Availability Configuration

For advanced private cloud environments—intended to support a high number of users—it is recommended that you configure a high availability (HA) environment, where the Apache server and PostgreSQL database server are installed on separate machines, and Anchor is replicated on multiple machines. Ultimately, this eliminates a single-point of failure at the Anchor level, and allows for distribution of load across multiple Anchor servers.



Note: These instructions will help you set up a high availability configuration for the Anchor server only. While this process allows you to place Apache or PostgreSQL on the server of your choice within the high availability cluster, it does not replicate the Apache or PostgreSQL servers. The system will only recognize one instance of the Apache server and one instance of the PostgreSQL server.

Instead, both Apache and PostgreSQL can be set up on a host, either with VM or HyperV.

For more information on setting up a windows cluster in VMware, please reference:

- [VMware vSphere High Availability](#)
- [Creating a vSphere HA Cluster](#)

For more information on setting up fault tolerance with Native VMware, please reference:

- [Setup for Failover Clustering and Microsoft Cluster Service](#)
- [Microsoft Clustering on VMWare vSphere](#)

Configuration Examples

High availability can be configured in various ways.

For example, when using two machines, you can configure the high availability cluster as follows:

- Machine 1: Anchor server, Apache server, and PostgreSQL server
- Machine 2: Anchor server

When using three or more machines, you can configure the high availability cluster as follows:

- Machine 1: Anchor server and PostgreSQL

- Machine 2: Anchor server and Apache server
- Machine 3 (or more): Anchor server

Requirements

The following items are required when configuring an environment for high availability:

- Two or more servers that meet system requirements as described above
- Load balancer or Round-robin DNS (with short 5-minute TTL)
- Root store and file store network storage mount points that can be accessed by all machines



Note: Storage can be permanently mounted on one server, and mapped to the other servers. NAS and SAN devices are also supported. This guide provides instructions for configuring both storage options.

Getting Started Checklist

It is assumed that the following tasks have been completed:

- Servers are not members of a domain
- IIS or additional web server software has been removed from each machine
- Ports 443, 80, and 510 are all available (dual hostnames will later be configured to remove the need for port 510)
- Data center infrastructure and networking components are installed
- The following access rights have been acquired:
 - Access to the load balancer/DNS server
 - Administrative privileges on all machines
- SSL certificates are installed and configured
- IP addresses of each server have been recorded for easy reference

Configuration Instructions

Step 1: Install Anchor and PostgreSQL on Machine 1

After you prepare for the high availability configuration, you can install the Anchor and PostgreSQL servers on machine 1.



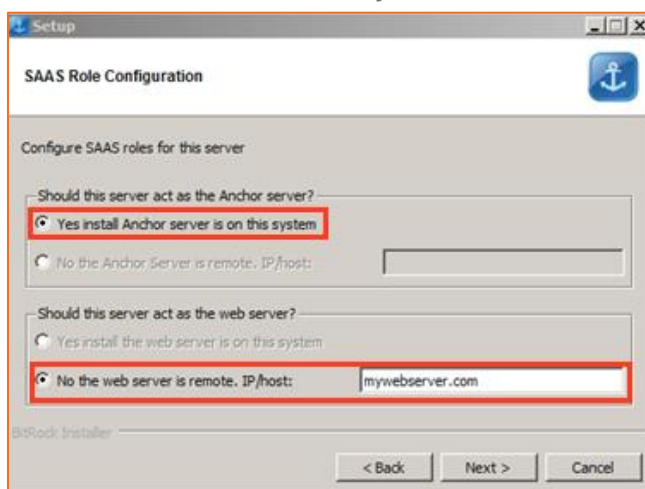
Note: Anchor must be installed on the PostgreSQL server first.

1. Log into machine 1 (the Anchor and PostgreSQL server) as a local admin.
2. On the server, download the latest Anchor installation file.
3. Open an elevated Command Prompt window. The Command Prompt window displays.
4. In the Command Prompt window, run the Anchor Server executable with the `--SaaS 1` argument, which will allow you to select specific components to install during the installation process.

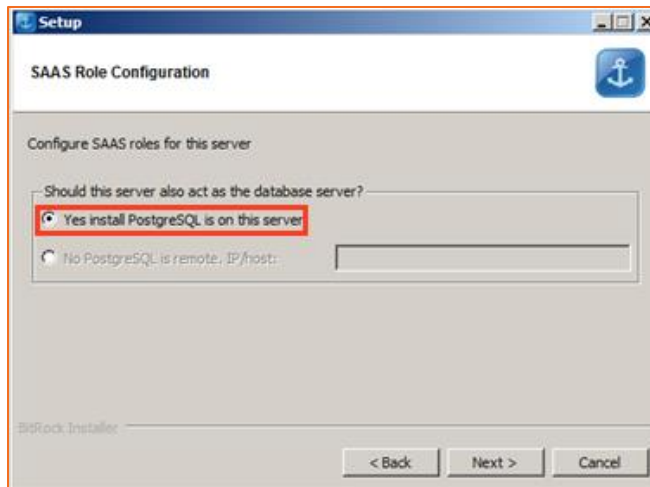
```
Administrator: C:\Windows\system32\cmd.exe
C:\Users\Administrator\Desktop>AnchorServer.exe --saas 1
```

The *Anchor Setup Wizard* will launch.

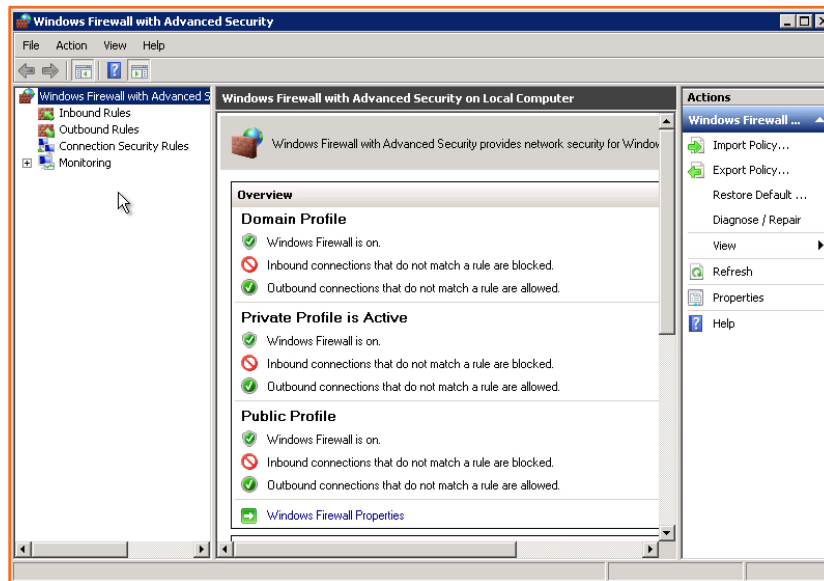
5. In the *Anchor Setup Wizard*, click the series of **Next** buttons to proceed through the setup process, making sure to pause at the *SAAS Role Configuration* screen.
6. In the *SAAS Role Configuration* screen, select the following options:
 - a. Select the Yes install Anchor server on this system radio button.
 - b. Select the No the web server is remote radio button.
 - c. Enter the web server **IP address** or **hostname**.
 - d. Click the **Next** button when you are finished.



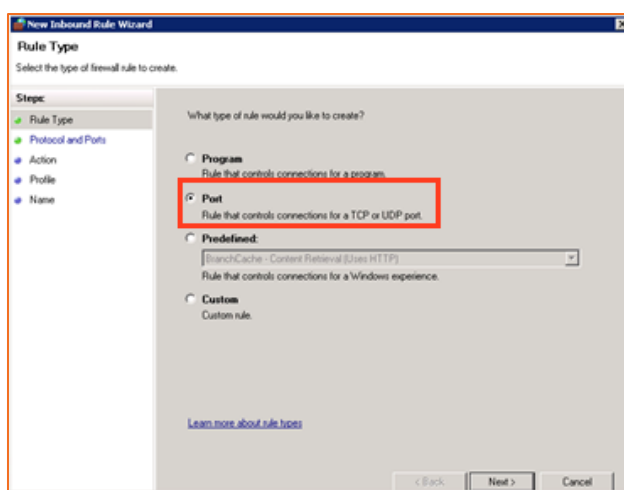
7. In the next *SaaS Role Configuration* screen, click the **Yes install PostgreSQL is on this server** radio button, and then click the **Next** button.



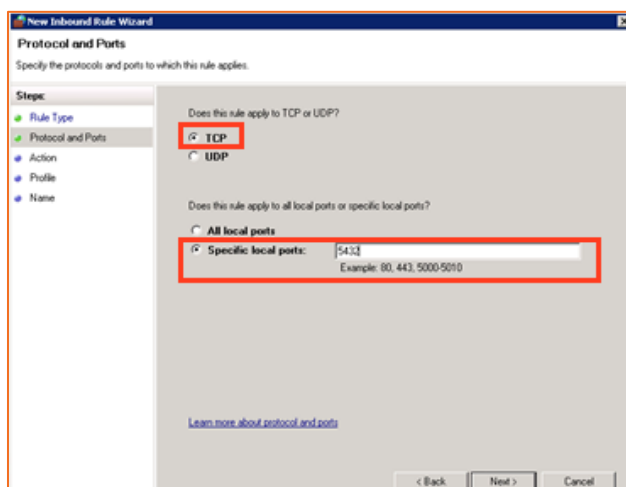
8. When the *Anchor Setup Wizard* completes, click the **Finish** button.
9. While still on the PostgreSQL server, configure the firewall for port 5432.
 - a. In the Windows Server *Start* menu, enter **Windows Firewall with Advanced Security** in the *search* box. The *Windows Firewall with Advanced Security* window displays.



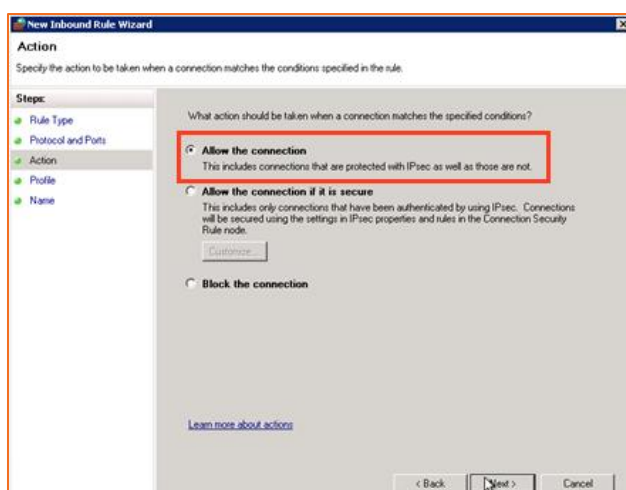
- b. Right-click the *Inbound Rules* navigation item and select **New Rule**. The *New Inbound Rule Wizard* displays.
- c. In the *New Inbound Rule Wizard*, select the **Port** radio button, and then click the **Next** button. The wizard advances to the *Protocol and Ports* screen.



- d. In the *Protocol and Ports* screen, configure the following options:
- Select the **TCP** radio button.
 - Select the Specific local ports radio button.
 - Enter port **5432**.
 - Click the **Next** button to continue to the *Action* screen.

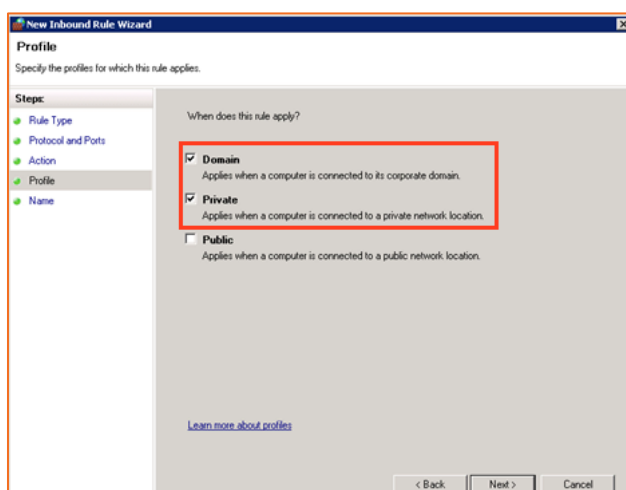


- e. In the *Action* screen, click the **Allow the connection** radio button, and click the **Next** button.



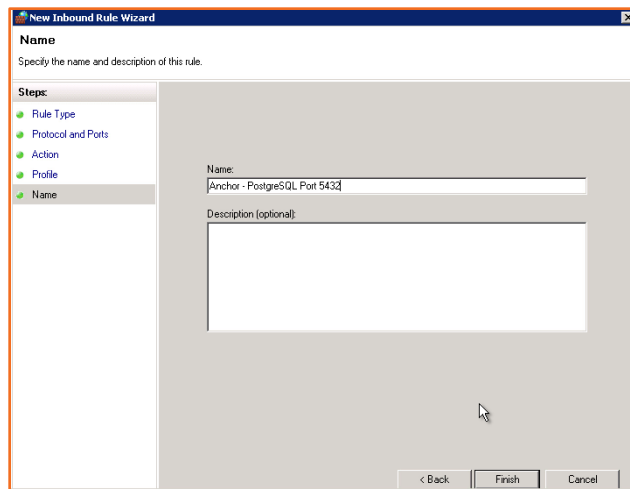
The *Profile* screen displays.

- f. In the *Profile* screen, ensure that *Domain* and *Private* are selected, and *Public* is unselected. Then, click the **Next** button to continue.



The *Name* screen displays.

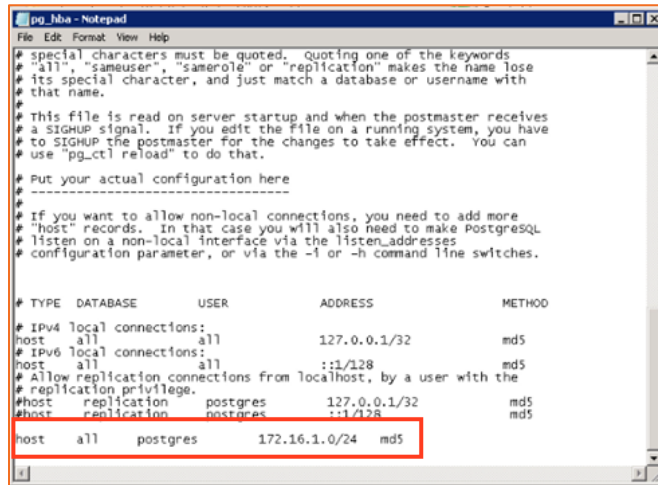
- g. In the *Name* screen, enter an appropriate **name**, and optionally, a **description** of the new rule.



- h. Click the **Finish** button.
10. While still on the PostgreSQL server, configure a PostgreSQL configuration file, so that the PostgreSQL server knows to accept connections from another machine.
 - a. Navigate to the *PostgreSQL* installation directory, which will be located at the default installation location (for example, *C:\PostgreSQL9.1*).
 - b. In the *PostgreSQL* directory, click the **data** folder, and then open the **pg_hba.conf** file.
 - c. In the *PostgreSQL* configuration file, scroll down to the bottom of the file and add the following configuration parameters that specify the machines, databases, and users that can connect to PostgreSQL, including:
 - i. Type of connection (for example, a *host* connection);
 - ii. Type of database (for example, *all* databases);
 - iii. The user;
 - iv. The IP address of the current server—in standard dotted decimal notation—and an CIDR Mask Link, which indicates the number of significant bits that must make up the client IP address (for example, *172.16.1.0/24*); and
 - v. The hash algorithm (for example, *md5*).

For more information on PostgreSQL client authentication and the

pg_hba.conf file, please review the [PostgreSQL manual](#).



```

pg_hba - Notepad
File Edit Format View Help
# special characters must be quoted. Quoting one of the keywords
# "all", "sameuser", "samerole" or "replication" makes the name lose
# its special character, and just match a database or username with
# that name.
#
# This file is read on server startup and when the postmaster receives
# a SIGHUP signal. If you edit the file on a running system, you have
# to SIGHUP the postmaster for the changes to take effect. You can
# use "pg_ctl reload" to do that.
#
# Put your actual configuration here
#
#
# If you want to allow non-local connections, you need to add more
# "host" records. In that case you will also need to make PostgreSQL
# listen on a non-local interface via the listen_addresses
# configuration parameter, or via the -l or -h command line switches.
#
# TYPE DATABASE USER ADDRESS METHOD
# IPv4 local connections:
host all all 127.0.0.1/32 md5
# IPv6 local connections:
host all all ::1/128 md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
#host replication postgres 127.0.0.1/32 md5
#host replication postgres ::1/128 md5
host all postgres 172.16.1.0/24 md5

```

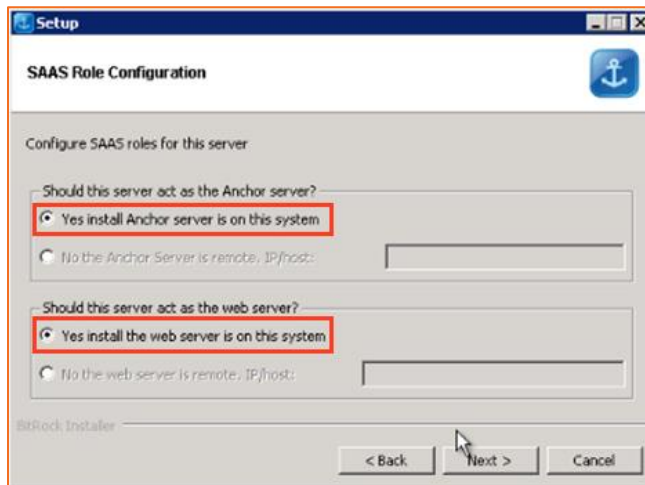
- d. Save and close the *PostgreSQL* configuration file.
- e. Restart the *PostgreSQL* service.

Step 2: Install Anchor and Apache on Machine 2

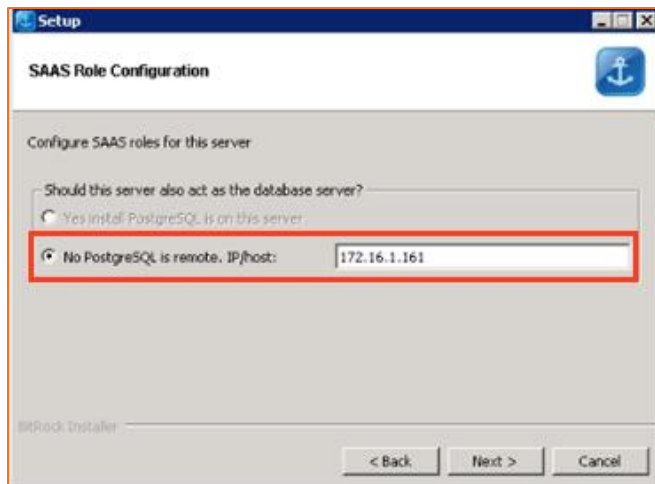
After Anchor and PostgreSQL are installed on machine 1, you can configure machine 2. For this example, we will install Anchor and Apache servers on machine 2.

1. Log into machine 2 (the Anchor and Apache server) as a local admin.
2. In the server, download the latest Anchor Server installation file.
3. Open an elevated Command Prompt window. The Command Prompt window displays.
4. In the Command Prompt window, run the Anchor Server executable with the *--SaaS 1* argument, which will allow you to select specific components to install during the installation process. The *Anchor Setup Wizard* will launch.
5. In the *Anchor Setup Wizard*, click the series of **Next** buttons to proceed through the setup process, making sure to pause at the *SAAS Role Configuration* screen.
6. In the *SAAS Role Configuration* screen, select the **Yes install Anchor server is on this system** radio button, the **Yes install the web server is on this system** radio button, and

then click the **Next** button to continue.



7. In the next *SaaS Role Configuration* screen, select the **No PostgreSQL is remote** radio button, enter the **IP address** or **hostname** of the Anchor and PostgreSQL server (machine 1), and then click the **Next** button to continue.



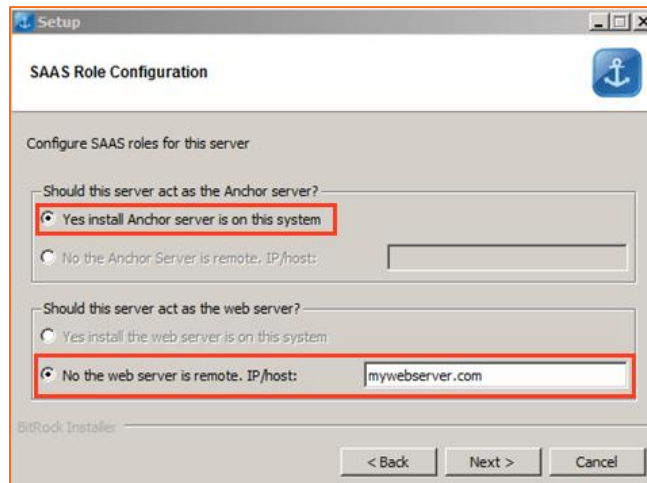
8. During the installation process, a *Question* dialog box will display, asking you if you would like to use the existing database the installer found on the system. Select the **Yes** button to confirm and continue.
9. Click the **Finish** button when the installation process is complete.

Step 3: Install Anchor on Machine 3

After you install Anchor and Apache on machine 2, you can install the third instance of Anchor on machine 3.

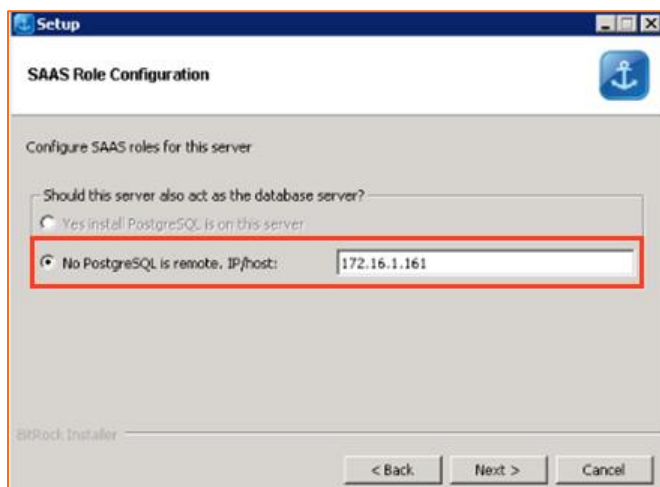
1. Log into machine 3 (the Anchor server) as a local admin.

2. In the server, download the latest Anchor installation file.
3. Open an elevated Command Prompt window. The Command Prompt window displays.
4. In the Command Prompt window, run the Anchor Server executable with the `--SaaS 1` argument, which will allow you to select specific components to install during the installation process. The *Anchor Setup Wizard* will launch.
5. In the *Anchor Setup Wizard*, click the series of **Next** buttons to proceed through the setup process, making sure to pause at the *SAAS Role Configuration* screen.
6. In the *SAAS Role Configuration* screen, select the following options:
 - a. Select the Yes install Anchor server on this system radio button.
 - b. Select the No the web server is remote radio button.
 - c. Enter the web server **IP address** or **hostname**.
 - d. Click the **Next** button when you are finished.



7. In the next *SaaS Role Configuration* screen, select the **No PostgreSQL is remote** radio button, enter the **IP address** or **hostname** of the Anchor and PostgreSQL server (machine

1), and then click the **Next** button to continue.



8. During the installation process, a *Question* dialog box will display, asking you if you would like to use the existing database the installer found on the system. Select the **Yes** button to confirm and continue.
9. Click the **Finish** button when the installation process is complete.

Option 1: Configuring a Locally Mounted Storage Drive



Note: These steps are only applicable if you plan to use a locally mounted drive. In most instances—especially for advanced environments—it is recommended that you plan to use a NAS or SAN storage device. For more information, please reference the [Option 2: Configuring a NAS or SAN Storage Device](#) section of this guide.

After the server services are installed, you need to create a location for the root store and the file store. You must then map these stores to each of the machines under the system context. For the purposes of this document, we will set up file stores on machine 1, and map these stores to machine 2 and machine 3.

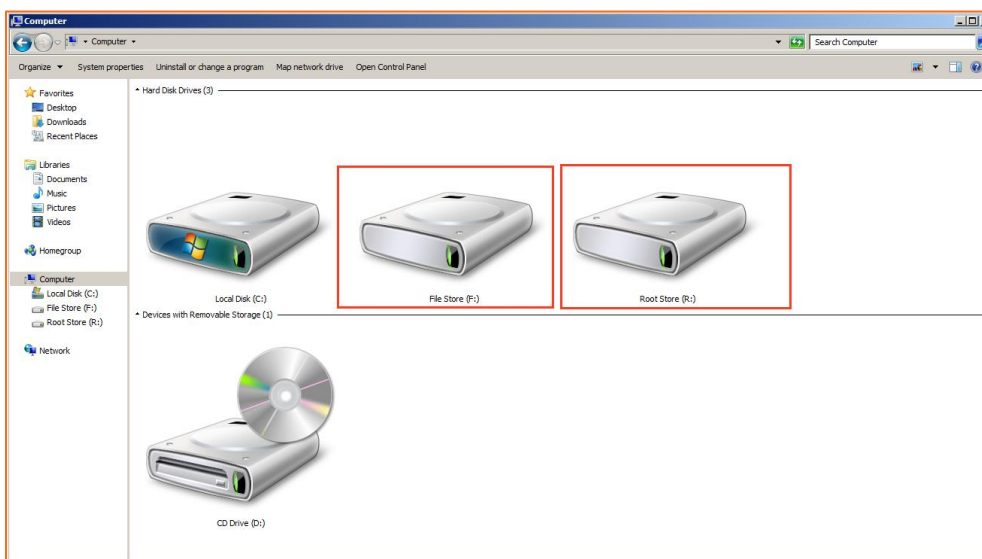
Configuration Instructions

Step 1: Map Drives



Note: When mapping drives, the paths must be the same (same drive letter) on each of the servers. Additionally, these drives need to be mapped under the system context.

1. On machine 1 (the Anchor and PostgreSQL server), create a root store location and a file store location.



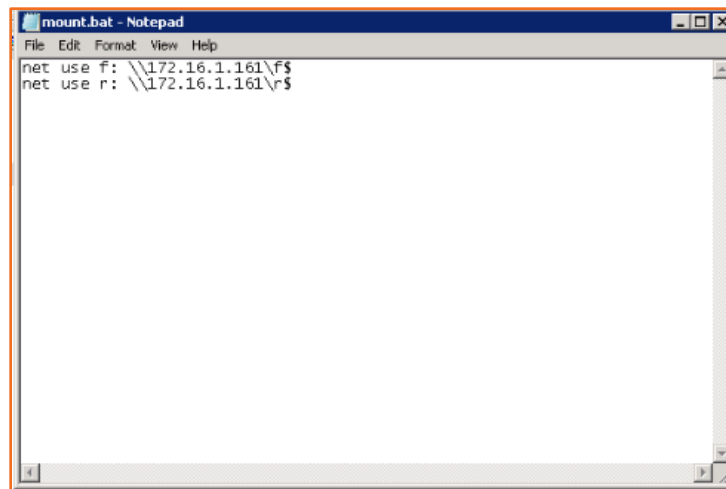
2. On machine 2 (the Anchor and Apache server), map a drive under the system context to the root and file stores on machine 1.



Note: The following script will *not* function when the server restarts or is unavailable.

- a. Navigate to the Anchor server installation drive (for example, *C:|Anchor Server*), and create a new batch file titled, *mount.bat*.
- b. Open *mount.bat* and use the **net use** command to map the drives locally to machine 1 as follows:

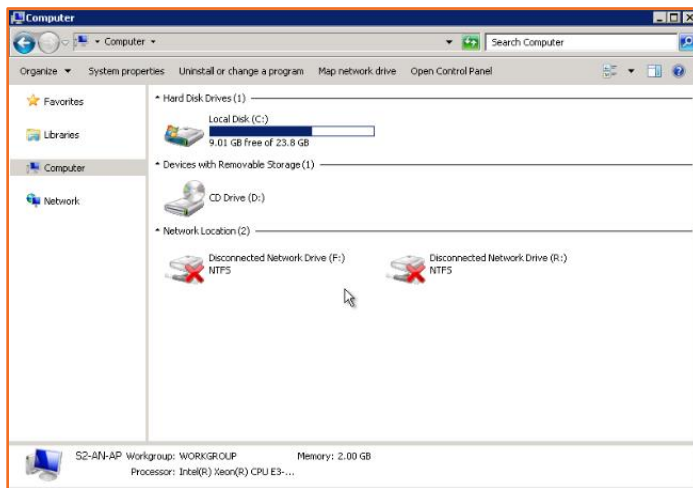
```
net use <drive letter>: \\<ip_of_storage_server>\<roots_mount_name>
net use <drive letter>: \\<ip_of_storage_server>\<files_mount_name>
```



- c. Save and close the file.
- d. While still on machine 2, in the Windows Server *Start* menu, enter **gpedit.msc** in the *search* box. The *Local Group Policy Editor* window displays.
- e. In the *Local Group Policy Editor* window, click **Computer Configuration**, then click **Windows Settings**, then click **Scripts**, and then select **Startup**. The *Startup Properties* dialog box displays.
- f. In the *Startup Properties* dialog box, click the **Add** button. The *Add a New Script* dialog box displays.
- g. In the *Add a New Script* dialog box, select the *mount.bat* file that was just created.
- h. Click the **OK** button and close all of the dialog boxes and windows.
3. On machine 3 (the Anchor server), map a drive under the system context to the root and file stores on machine 1.
 - a. Navigate to the Anchor server installation drive (for example, *C:|Anchor Server*), and create a new batch file titled, *mount.bat*.
 - b. Open *mount.bat* and use the **net use** command to map the drives locally to machine 1 as follows:


```
net use <drive letter>: \\<ip_of_storage_server>\<roots_mount_name>
net use <drive letter>: \\<ip_of_storage_server>\<files_mount_name>
```
 - c. Save and close the file.
 - d. While still on machine 3, in the Windows Server *Start* menu, enter **gpedit.msc** in the *search* box. The *Local Group Policy Editor* window displays.

- e. In the *Local Group Policy Editor* window, click **Computer Configuration**, then click **Windows Settings**, then click **Scripts**, and then select **Startup**. The *Startup Properties* dialog box displays.
 - f. In the *Startup Properties* dialog box, click the **Add** button. The *Add a New Script* dialog box displays.
 - g. In the *Add a New Script* dialog box, select the *mount.bat* file that you just created.
 - h. Click the **OK** button and close all of the dialog boxes and windows.
4. Restart machine 2 and machine 3.
 5. When you log in to machine 2 and machine 3, you will see the mapped drives displayed as follows:








Step 2: Register Anchor


Now that your server services are installed, and storage locations are configured, you can set up the Anchor system. For full instructions on how to set up the Anchor system, please reference the [Anchor System Setup](#) Knowledgebase article.



1. On the Anchor and Apache server machine (for our example, machine 2), click the **Anchor Server Management** desktop shortcut. The *administrative web portal* launches in your browser.
2. In the *License* tab and *System Admin* tab, enter the required information.
3. In the *System Settings* tab, you must configure hostname settings for the load balancer, as well as the file and DB store locations.
 - a. In the *Hostname* field, enter the **hostname** of the load balancer.

- b. Use the *Dual Hostname* checkbox and *App Server Hostname* field to configure a different hostname for the app server and web server. You can configure these settings after you complete the initial setup process. For more information, please reference [Configuring Dual Hostname Settings](#) section of this guide.

 License	 System Admin	 System Settings	 Encryption	 Email Server
---	--	---	--	--

Hostname or IP

**Note:** The IP address or hostname (recommended) you set below will be embedded in the agent installation. Agents will not be able to connect to the server if the hostname does not properly resolve. If you plan on using unique hostnames for each customer, you will be able to specify those after system set-up.

Hostname		<input type="text"/>	*
Dual Hostnames		<input type="checkbox"/> Use different hostnames for the app server and web server.	
App Server Hostname	<input type="text"/>		

- c. In the *File and DB Store Locations* section, navigate to the ***file store location*** created in the steps above.

- d. In the *Database Store* field, select **Custom Location**, and then navigate to the **root store location** created in the steps above.

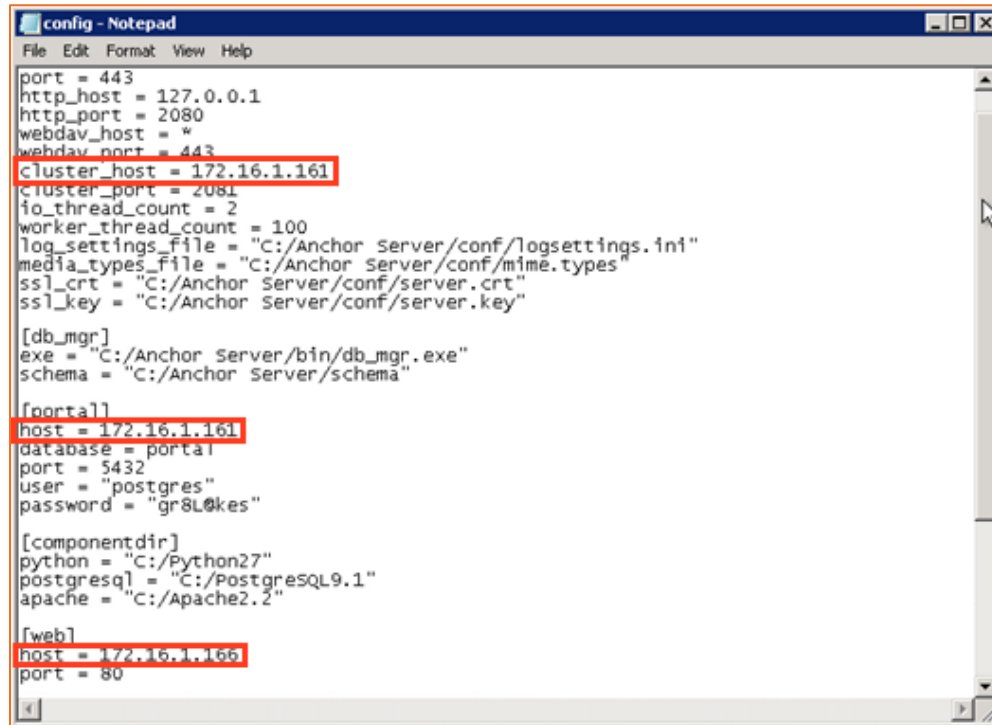
The screenshot shows a configuration window titled "File and DB Store Locations". It contains a section "Choose One or More Folders" with a help icon. To the right of this section is a file explorer view showing the "C:/" directory. Below the file explorer are two buttons: "REFRESH SELECTED FOLDERS" and "NEW FOLDER". At the bottom of the window, there is a "Database Store" section with a help icon. It contains two radio buttons: "Default Location C:\Anchor Server" (which is selected) and "Custom Location" (which is unselected).

Step 3: Modify Anchor Configuration Files

After the Anchor system has been set up, you must modify the Anchor configuration file on each machine, and then restart each Anchor Server service.

1. On machine 1 (the Anchor and PostgreSQL server), modify the Anchor configuration file.
 - a. Open the *Anchor Server* installation directory (for example, *C:\Anchor Server*).
 - b. In the *Anchor Server* directory, click the **conf** folder, and then open the **config.ini** file. The *config.ini* file displays.
 - c. Change the *cluster host*, the *portal host*, and the *web host* parameters in the *config.ini* file.
 - i. Cluster host—set this parameter to the IP address of the current machine (in this case, machine 1, or the Anchor and PostgreSQL server).
 - ii. Portal host—set this parameter to the explicit IP address of the Anchor and PostgreSQL server (machine 1).
 - iii. Web host—this parameter will be set to the explicit IP address of the Anchor and Apache server (machine 2).

- d. Save and close the *config.ini* file.



```

config - Notepad
File Edit Format View Help
port = 443
http_host = 127.0.0.1
http_port = 2080
webdav_host = *
webdav_port = 443
cluster_host = 172.16.1.161
cluster_port = 2081
io_thread_count = 2
worker_thread_count = 100
log_settings_file = "C:/Anchor Server/conf/logsettings.ini"
media_types_file = "C:/Anchor Server/conf/mime.types"
ssl_cert = "C:/Anchor Server/conf/server.crt"
ssl_key = "C:/Anchor Server/conf/server.key"

[db_mgr]
exe = "C:/Anchor Server/bin/db_mgr.exe"
schema = "C:/Anchor Server/schema"

[portal]
host = 172.16.1.161
database = portal
port = 5432
user = "postgres"
password = "gr8L0kes"

[componentdir]
python = "C:/Python27"
postgresql = "C:/PostgreSQL9.1"
apache = "C:/Apache2.2"

[web]
host = 172.16.1.166
port = 80
  
```

2. On machine 2 (the Anchor and Apache server), repeat the same process.
 - a. Open the *Anchor Server* installation directory (for example, *C:\Anchor Server*).
 - b. In the *Anchor Server* directory, click the **conf** folder, and then open the **config.ini** file. The *config.ini* file displays.
 - c. Change the *cluster host*, the *portal host*, and the *web host* parameters in the *config.ini* file.
 - i. Cluster host – this parameter will be set as the IP address of the current machine (in this case, machine 2, or the Anchor and Apache server).
 - ii. Portal host – this parameter will be set to the explicit IP address of the Anchor and PostgreSQL server (machine 1).
 - iii. Web host – this parameter will be set to the explicit IP address of the Anchor and Apache server (machine 2).
 - d. Save and close the *config.ini* file.
3. On machine 3 (the Anchor server), repeat the same process.
 - a. Open the *Anchor Server* installation directory (for example, *C:\Anchor Server*).
 - b. In the *Anchor Server* directory, click the **conf** folder, and then open the **config.ini** file. The *config.ini* file displays.

- c. Change the *cluster host*, the *portal host*, and the *web host* parameters in the *config.ini* file.
 - i. Cluster host – this parameter will be set as the IP address of the current machine (in this case, machine 3, or the Anchor server).
 - ii. Portal host – this parameter will be set to the explicit IP address of the Anchor and PostgreSQL server (machine 1).
 - iii. Web host – this parameter will be set to the explicit IP address of the Anchor and Apache server (machine 2).
 - d. Save and close the *config.ini* file.
4. Restart the Anchor server service on each machine.

Option 2: Configuring a NAS, SAN, or Other Storage Device

In instances where storage is set up on a NAS, SAN, or other storage device, you will need to enable the use of a UNC path so that each server can access the root store and the file store.

For the purposes of this document, we will describe the process of enabling and using a UNC path within a three server, high-availability environment (using Windows Server 2008 or 2012), as outlined below:

- Machine 1: Anchor server and PostgreSQL
- Machine 2: Anchor server and Apache server
- Machine 3 (or more): Anchor server



Note: This process requires the creation and use of a local admin account on each server; therefore, each server must be a non-domain controller (DC) server.

Configuration Instructions

Step 1: Create Local Admin Accounts on Each Machine

As a first step, create a local administrator account on each machine.



Note: Each local admin account *must* have the same username and password. For instructions on how to create a local admin account, refer to the Microsoft Technet articles, [Create a Local User Account](#) and [Add a Member to a Local Group](#).

Step 2: Register Anchor

1. On the Anchor and Apache server machine (for our example, machine 2), click the **Anchor Server Management** desktop shortcut. The *administrative web portal* launches in your browser.
2. In the *License* tab and *System Admin* tab, enter the required information.
3. In the *System Settings* tab, you must configure hostname settings for the load balancer, as well as the file and DB store locations.
 - a. In the *Hostname* field, enter the **hostname** of the load balancer.

- b. Use the *Dual Hostname* checkbox and *App Server Hostname* field to configure a different hostname for the app server and web server. You can configure these settings after you complete the initial setup process. For more information, please reference [Configuring Dual Hostname Settings](#) section of this guide.

License	System Admin	System Settings	Encryption	Email Server
---------	--------------	-----------------	------------	--------------

Hostname or IP

Note: The IP address or hostname (recommended) you set below will be embedded in the agent installation. Agents will not be able to connect to the server if the hostname does not properly resolve. If you plan on using unique hostnames for each customer, you will be able to specify those after system set-up.

Hostname *

Dual Hostnames ☐ Use different hostnames for the app server and web server.

App Server Hostname

- c. In the *File and DB Store Locations* section, create a temporary folder for configuration purposes (for example, c:\Anchor Server\file_store_1). You will later move the location of this folder.
- d. In the *Database Store* field, select the **Default Location** radio button to create a temporary folder for configuration purposes (for example, c:\Anchor

Server\store_1). You will later move the location of this folder.

File and DB Store Locations

Choose One or More Folders ?

C:/

REFRESH SELECTED FOLDERS NEW FOLDER

Database Store ?

☒ Default Location C:\Anchor Server

☐ Custom Location

Step 3: Copy Metadata

When Anchor installation and registration is complete, and you are able to access the web portal, copy metadata from the temporary folders created above to the new location of your choice (for example, a location within a NAS or SAN device).

1. Copy all contents in the temporary file store (for example c:\Anchor Server\file_store_1) to the permanent file store location.
2. Copy all contents in the temporary root store (for example, c:\Anchor Server\store_1) to the permanent root store location.

Step 4: Change Services to Run as Local Admin

1. For each Anchor server service, change the user under which the service is running to the local admin created above.
2. Restart all services.

Step 5: Modify Anchor Configuration Files

After the Anchor system has been set up, you must modify the Anchor configuration file on each machine, and then restart each Anchor Server service.

1. On machine 1 (the Anchor and PostgreSQL server), modify the Anchor configuration file.
 - a. Open the *Anchor Server* installation directory (for example, *C:\Anchor Server*).
 - b. In the *Anchor Server* directory, click the **conf** folder, and then open the **config.ini** file. The *config.ini* file displays.
 - c. Change the *cluster host*, the *portal host*, and the *web host* parameters in the *config.ini* file.
 - i. Cluster host—set this parameter to the IP address of the current machine (in this case, machine 1, or the Anchor and PostgreSQL server).
 - ii. Portal host—set this parameter to the explicit IP address of the Anchor and PostgreSQL server (machine 1).
 - iii. Web host—this parameter will be set to the explicit IP address of the Anchor and Apache server (machine 2).
 - d. Save and close the *config.ini* file.

```

config - Notepad
File Edit Format View Help
port = 443
http_host = 127.0.0.1
http_port = 2080
webdav_host = *
webdav_port = 443
cluster_host = 172.16.1.161
cluster_port = 2081
io_thread_count = 2
worker_thread_count = 100
log_settings_file = "C:/Anchor Server/conf/logsettings.ini"
media_types_file = "C:/Anchor Server/conf/mime.types"
ssl_cert = "C:/Anchor Server/conf/server.crt"
ssl_key = "C:/Anchor Server/conf/server.key"

[db_mgr]
exe = "C:/Anchor Server/bin/db_mgr.exe"
schema = "C:/Anchor Server/schema"

[portal]
host = 172.16.1.161
database = portal
port = 5432
user = "postgres"
password = "gr8L0kes"

[componentdir]
python = "C:/Python27"
postgresql = "C:/PostgreSQL9.1"
apache = "C:/Apache2.2"

[web]
host = 172.16.1.166
port = 80
  
```

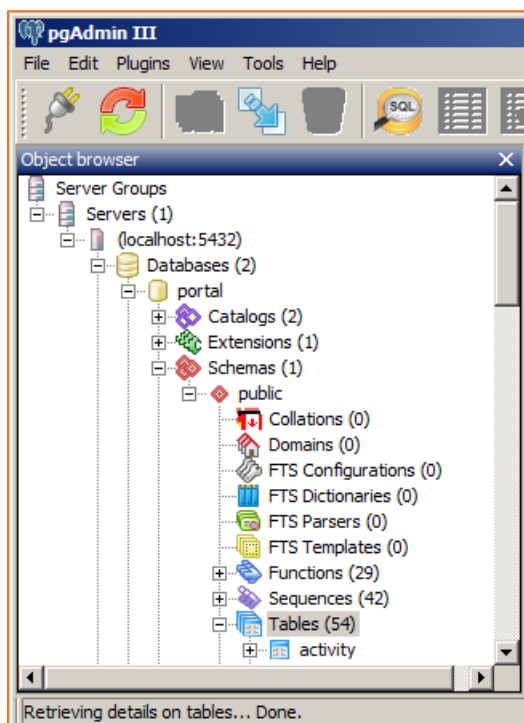
2. On machine 2 (the Anchor and Apache server), repeat the same process.
 - a. Open the *Anchor Server* installation directory (for example, *C:\Anchor Server*).
 - b. In the *Anchor Server* directory, click the **conf** folder, and then open the **config.ini** file. The *config.ini* file displays.

- c. Change the *cluster host*, the *portal host*, and the *web host* parameters in the *config.ini* file.
 - i. Cluster host – this parameter will be set as the IP address of the current machine (in this case, machine 2, or the Anchor and Apache server).
 - ii. Portal host – this parameter will be set to the explicit IP address of the Anchor and PostgreSQL server (machine 1).
 - iii. Web host – this parameter will be set to the explicit IP address of the Anchor and Apache server (machine 2).
 - d. Save and close the *config.ini* file.
 3. On machine 3 (the Anchor server), repeat the same process.
 - a. Open the *Anchor Server* installation directory (for example, *C:\Anchor Server*).
 - b. In the *Anchor Server* directory, click the **conf** folder, and then open the **config.ini** file. The *config.ini* file displays.
 - c. Change the *cluster host*, the *portal host*, and the *web host* parameters in the *config.ini* file.
 - i. Cluster host – this parameter will be set as the IP address of the current machine (in this case, machine 3, or the Anchor server).
 - ii. Portal host – this parameter will be set to the explicit IP address of the Anchor and PostgreSQL server (machine 1).
 - iii. Web host – this parameter will be set to the explicit IP address of the Anchor and Apache server (machine 2).
 - d. Save and close the *config.ini* file.
 4. Restart the Anchor server service on each machine.

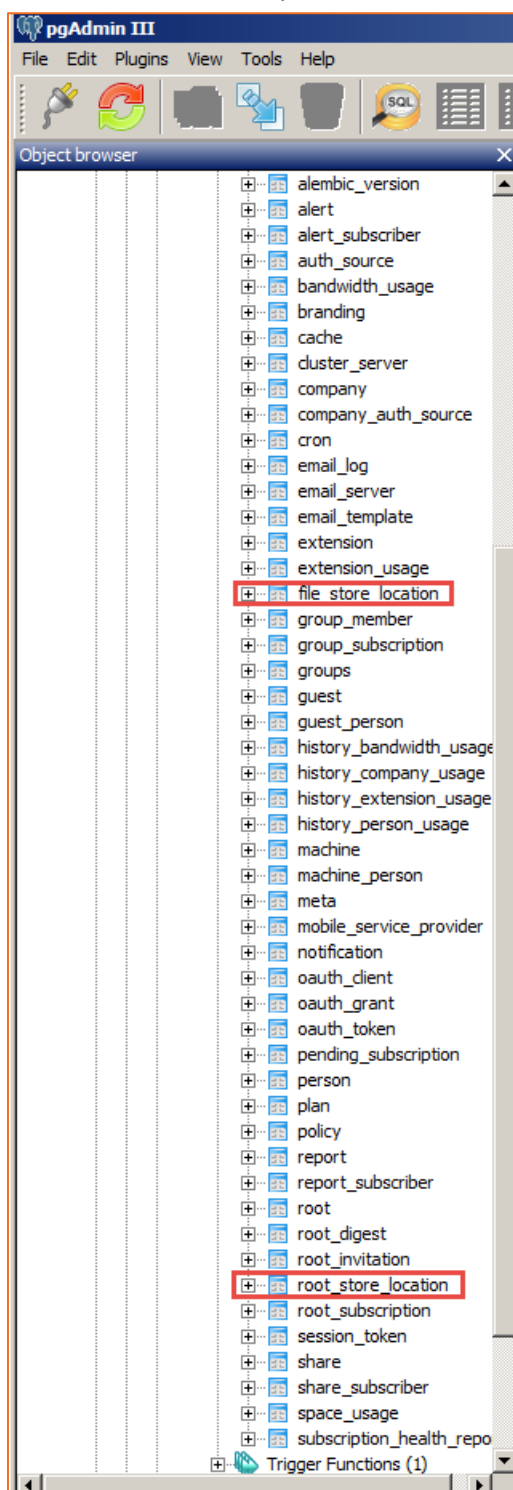
Step 6: Update PostgreSQL with the UNC Path

1. On machine 1 (Anchor server and PostgreSQL server) navigate to *C:\PostgreSQL9.1\bin*, and launch **pgAdmin3**. The application will open, prompting you to log in (password: *gr8L@kes*).
2. In the Object Browser, click to expand **Databases**, then click to expand **Portal**, then click to expand **Schemas**, then click to expand **Public**, and finally, click to expand

Tables.

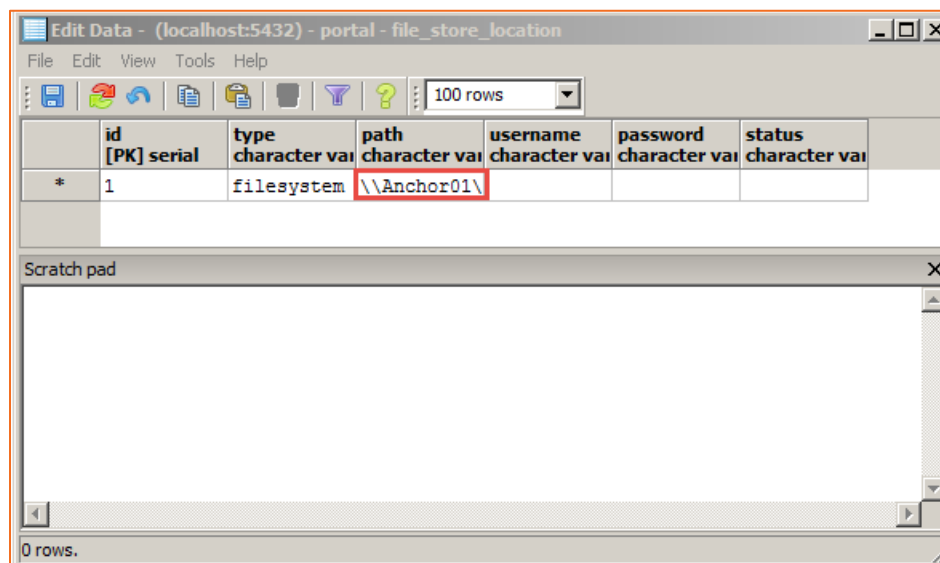


3. In the *Tables* section, locate the File and Root store tables.



4. In both the File Store and Root Store tables, update the path location.

- a. Right click the table, point to *View Data*, and select **View Top 100 rows**.
- b. In the *path* field, enter the **UNC path**.
- c. Save your changes.



5. Restart all services.

Configuring Backups

It is recommended that you back up the root store, the file store, and PostgreSQL. For information about configuring backups, please reference the [Backing Up Anchor Server and Data](#) Knowledgebase article.

Reviewing Mappings for the Load Balancer or DNS Server

After storage is set up in your environment, you can review IP mappings for the load balancer or DNS server. Under the default settings, port 443 is reserved for the Anchor service, and ports 80 and 510 are reserved for the Apache service.

Default Port Settings for an Anchor Server:

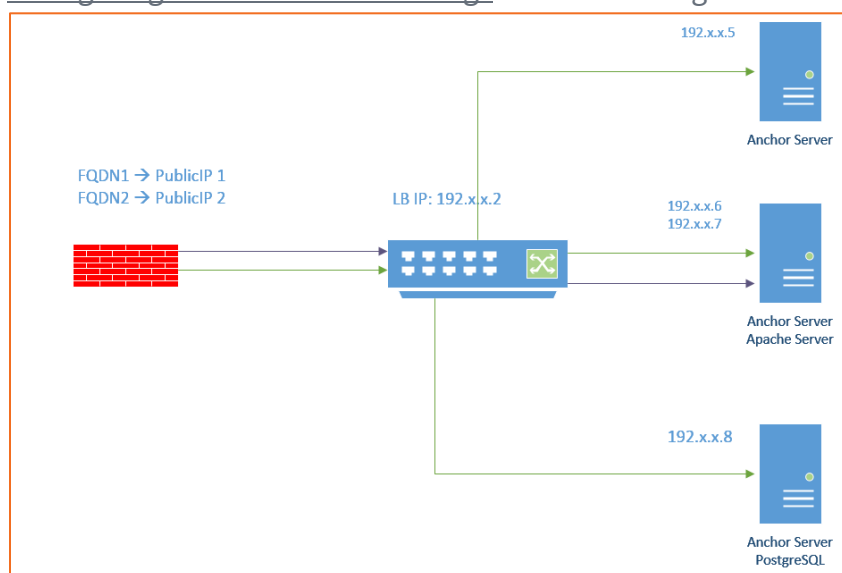
External_IP_1 FQDN1:443 | Load_Balancer_IP → Internal_IP_1:443

Default Port Settings for an Apache Server:

External_IP_1 FQDN1:510 | Internal_IP_1:510

External_IP_1 FQDN1:80 | Internal_IP_1:80

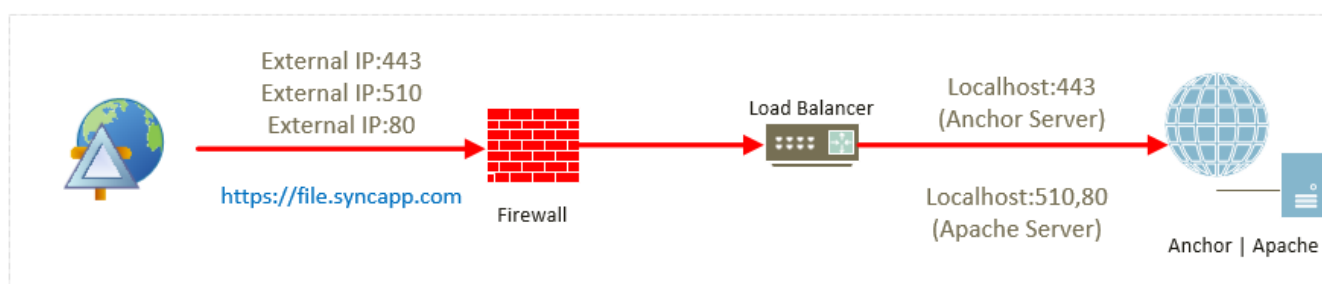
For security purposes, it is recommended that you remove port 510 and instead utilize port 410 through the use of dual hostname settings. For more information, please reference the [Configuring Dual Hostname Settings](#) section of this guide.



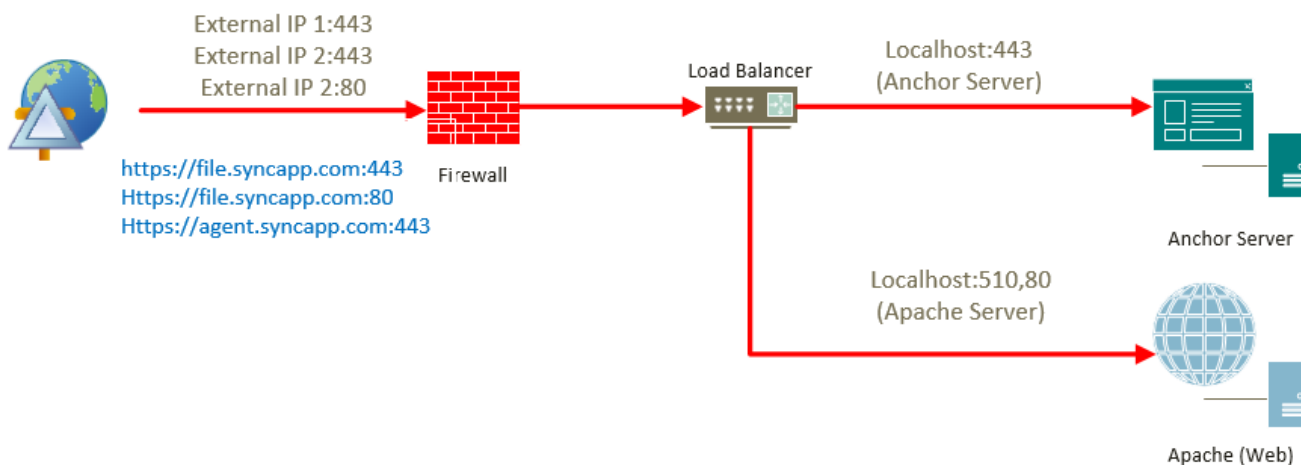
Configuring Dual Hostname Settings

By default, the Anchor and Apache services are installed on one server with one shared domain (for example, <https://file.syncapp.com>), and run on the localhost. Port 443 can only be used by one service. Under the default settings, port 443 is reserved for the Anchor service, and ports 80 and 510 are reserved for the Apache service.

Default Port Settings for an Anchor Server:



As an alternative to this configuration, Anchor and Apache can be configured to use two separate domains (or use a domain and a subdomain, such as <http://SyncApp.com> and <http://web.SyncApp.com>). A dual Host Cert and two Public IP's is required. Under this configuration, both the Anchor service and the Apache service can allow external connections on 443, which improves access.



Running two (2) apache server services is not support. In this setup Port forwarding will be used to NAT traffic from <https://file.syncapp.com:443> to Apache Service Server IP 10.X.X.12:510. Apache will continue listening on port 510 internally.

If you would like to completely remove all traces of port 510 then either an additional internal IP will be need to allow for the anchor and apache services to both listen on port 443 or Apache can be run on a VM that is not running anchor.

Configuration Instructions

Step 1: Configure a Second Domain or Subdomain



Note: If you have already deployed agents, you will need to reserve your original domain for the Anchor service, and assign the new domain or subdomain to Apache; this means that you and your users will no longer be able to access the web portal through the existing domain. Agents are hard-coded, and will not recognize a new domain without a full uninstall and reinstall of the agents.

Please contact Support if you are deploying this into a production environment.

As a first step, you will need to configure a second domain, or create a new subdomain, through your DNS hosting provider's web site. For specific instructions, please contact your DNS hosting provider.

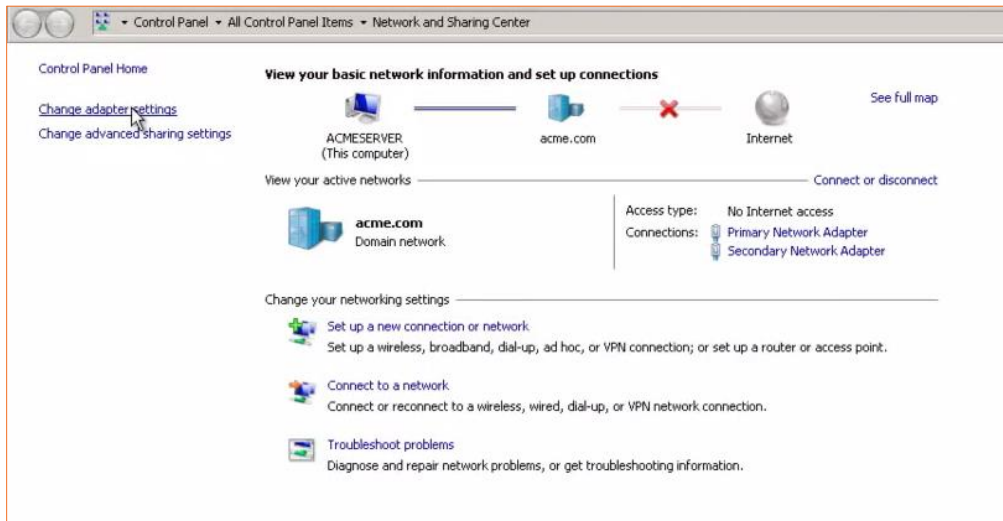
(Optional) Step 1a: Configure an Additional IP Address

This step is only necessary if you would like to completely remove all traces of port 510. After you configure a second domain or a subdomain, you can configure a new IP address. You do not need a new NIC.

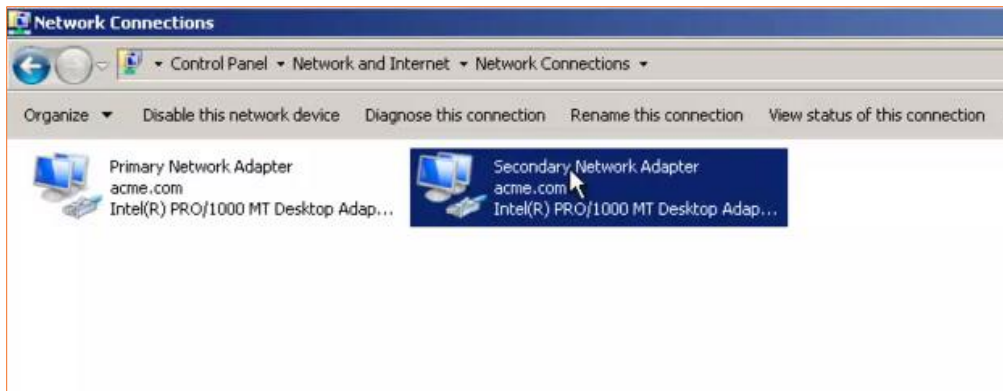


Note: These steps will help you configure an IP address at the OS-level. Alternatively, you can utilize a different configuration method (for example, [network level configuration](#)).

1. In the *Start* menu, point to *Control Panel*, and select **Network and Sharing Center**. The *Network and Sharing Center* window displays.

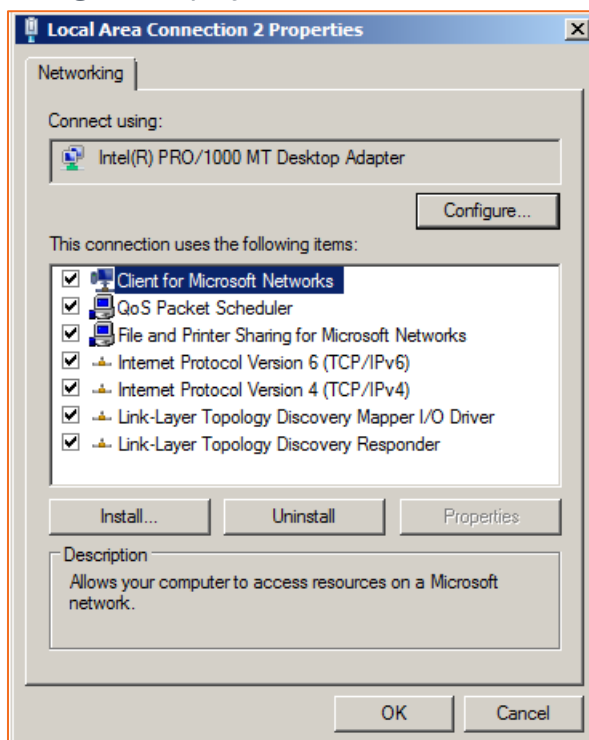


2. In the *Network and Sharing* window, click **Change Adapter Settings**. The *Network Connections* window displays.



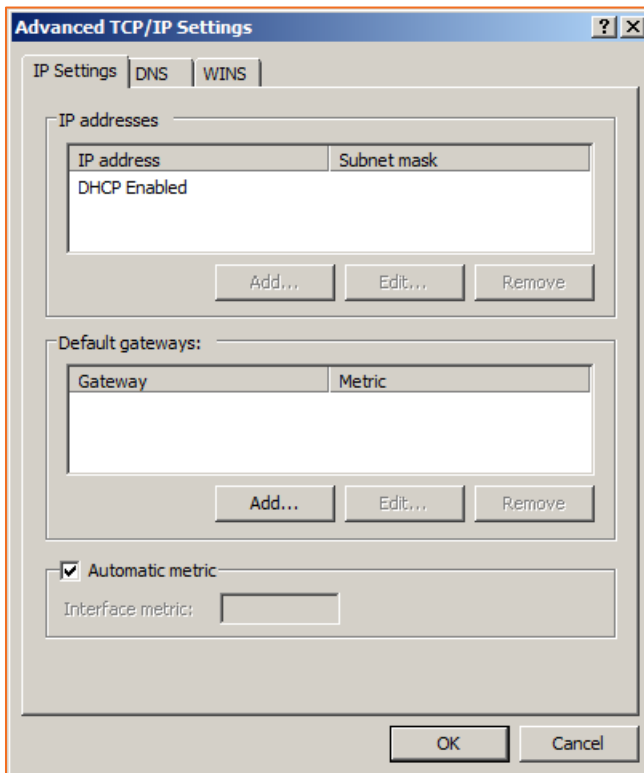
3. In the *Network Connections* window, right-click the **network adapter** to which the IP address will be added, and select **Properties**. The *Local Area Connection Properties*

dialog box displays.

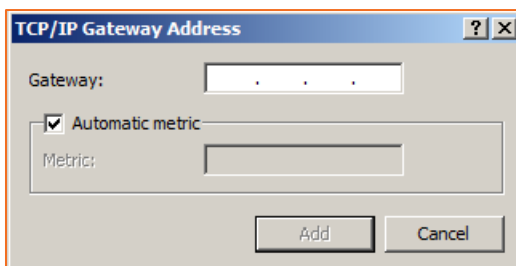


4. In the *Local Area Connection Properties* dialog box, click to highlight **Internet Protocol Version 4 (TCP/IPv4)**. With *Internet Protocol Version 4 (TCP/IPv4)* highlighted, click the **Properties** button. The *Internet Protocol Version 4 (TCP/IPv4) Properties* dialog box displays.

- In the Internet Protocol Version 4 (TCP/IPv4) Properties dialog box, click the **Advanced** button. The Advanced TCP/IP Settings dialog box displays.



- In the *Advanced TCP/IP Settings* dialog box, click the **Add...** button. The *TCP/IP Address* dialog box displays.



- In the *TCP/IP Address* dialog box, enter the **IP address** and **Subnet mask**, and click the **Add** button. The *Advanced TCP/IP Settings* dialog box now shows the new *IP address* and *Subnet mask*.
- Click the **OK** or **Close** buttons to return to the *Network and Sharing Center* window.

(Optional) Step 1b: Stop the Apache and Anchor Services

After you configure a new domain and a new IP address, you will need to stop the Apache and Anchor services.



Note: Ensure that any affected customer knows of this temporary downtime before stopping the services.



Note: The Apache service should be stopped before the Anchor service.

1. On the Apache server, launch *services.msc*.
2. In the *Services* window, right-click the **Apache** service and select **Stop**. The Apache service is now stopped.
3. While still in the *Services* window, right-click the **Anchor** service and select **Stop**. The Anchor service is now stopped.

Step 2: Point the New Domain or Subdomain to the new IP Address

When the services have been stopped, you can then point your new domain to the new IP address configured in the steps above.

For specific instructions, please reference guidelines provided by your web hosting provider. For example, [GoDaddy Support for Updating your Domain Name IP Address](#).

Step 3: Configure the Port Address Translation (PAT) Settings

The following PAT settings need to be configured:

file.syncapp.com:443 → External_IP1:443 → localhost:510 → Apache

file.syncapp.com:80 → External_IP1:80 → localhost:80 → Apache

agent.syncpp.com:443 → External_IP2:443 → localhost:443 → Anchor service

You can remove Port 510 completely, assuming Anchor Server and Apache are running on the same VM:

file.syncapp.com:443 → External_IP1:443 → Internal IP 1:443 → Apache

file.syncapp.com:80 → External_IP1:80 → localhost:80 → Apache (Apache does the redirect (httpd) and should always be set to listen on port 80 on the localhost)

Step 4: Configure the SSL Certificates

For information on configuring SSL certificates, please reference the [How Do I Configure a Single Domain or Wildcard SSL Certificate](#) Knowledgebase article.

Step 5: Update the Apache Configuration File

Next, the Apache configuration file will need to be updated with the new domain or subdomain.

1. In your Apache server, open the httpd.conf file, which is located at *[target drive]:\Apache24\conf*.
2. Copy and paste the file in the current location. This will create a backup of the current running Config file.
3. Compare the httpd.conf file to the http.conf file listed below. Make the necessary replacements, making sure to retain appropriate SSL certificate file paths.

```
Listen 80

LoadModule actions_module modules/mod_actions.so
LoadModule alias_module modules/mod_alias.so
LoadModule asis_module modules/mod_asis.so
LoadModule auth_basic_module modules/mod_auth_basic.so
LoadModule authn_core_module modules/mod_authn_core.so
LoadModule authn_file_module modules/mod_authn_file.so
LoadModule authz_core_module modules/mod_authz_core.so
LoadModule authz_groupfile_module modules/mod_authz_groupfile.so
LoadModule authz_host_module modules/mod_authz_host.so
LoadModule authz_user_module modules/mod_authz_user.so
LoadModule autoindex_module modules/mod_autoindex.so
LoadModule cgi_module modules/mod_cgi.so
LoadModule env_module modules/mod_env.so
LoadModule include_module modules/mod_include.so
LoadModule isapi_module modules/mod_isapi.so
LoadModule log_config_module modules/mod_log_config.so
LoadModule mime_module modules/mod_mime.so
LoadModule negotiation_module modules/mod_negotiation.so
LoadModule setenvif_module modules/mod_setenvif.so
LoadModule ssl_module modules/mod_ssl.so
LoadModule socache_shmcb_module modules/mod_socache_shmcb.so
LoadModule wsgi_module modules/mod_wsgi.so
LoadModule rewrite_module modules/mod_rewrite.so

ErrorLog "logs/error.log"
```



```

LogLevel warn

<IfModule log_config_module>
    LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\""
combined
    LogFormat "%h %l %u %t \"%r\" %>s %b" common
    CustomLog "logs/access.log" common
</IfModule>

<IfModule mime_module>
    TypesConfig conf/mime.types
    AddType application/x-compress .Z
    AddType application/x-gzip .gz .tgz
</IfModule>

AllowEncodedSlashes On
TraceEnable Off
AcceptFilter http none
AcceptFilter https none
EnableSendfile Off
EnableMMAP Off

RewriteEngine On
RewriteCond %{HTTPS} off
RewriteCond %{REQUEST_URI} !^/updater
RewriteCond %{REQUEST_URI} !^/static/assets/
RewriteCond %{REQUEST_URI} !^/server/hostname
RewriteRule (.*?) https://%{HTTP_HOST}443%{REQUEST_URI}

### SSL ###
Listen 510
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]

<VirtualHost _default_:510>
    [REDACTED]
    [REDACTED]

```

```

SSLCertificateFile "conf/ssl/server.crt"
SSLCertificateKeyFile "conf/ssl/server.key"
SSLCertificateChainFile "conf/ssl/bundlecrt.crt"
AllowEncodedSlashes On
</VirtualHost>

<IfModule ssl_module>
    SSLRandomSeed startup builtin
    SSLRandomSeed connect builtin
</IfModule>
### END-SSL ###

WSGIPythonHome "C:/Anchor Server/penv"
WSGIScriptAlias / "C:/Anchor Server/web/anchor.wsgi" application-
group=%{GLOBAL}
WSGIImportScript "C:/Anchor Server/web/anchor.wsgi" application-group=%{GLOBAL}
WSGIPassAuthorization On

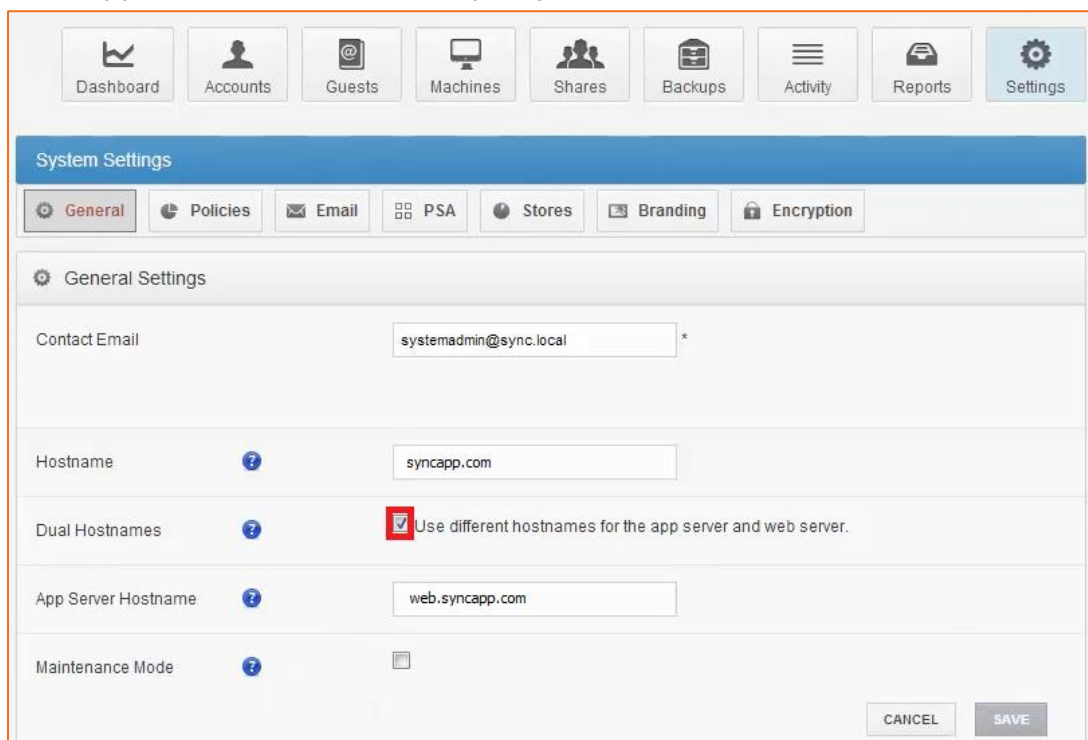
```

Save the file when you are finished and restart Apache server service. Apache should start with no exception. If there is an error, consult the event logs.

Step 6: Specify the App Server Hostname within the Web Portal

1. With both the Apache and Anchor services running, open the web portal in your browser.
2. From the web portal dashboard, click the **Settings** tab. The *Settings* page displays.
3. In the *Settings* page, click the **General** tab. The *General Settings* page displays.
4. In the *Dual Hostnames* field, click the **Dual Hostnames** checkbox.

5. In the *App Server Hostname* field, specify the new **app server hostname address**.



The screenshot shows the Axcient System Settings interface. At the top, there is a navigation bar with icons for Dashboard, Accounts, Guests, Machines, Shares, Backups, Activity, Reports, and Settings. Below this is a 'System Settings' header with tabs for General, Policies, Email, PSA, Stores, Branding, and Encryption. The 'General' tab is selected. Under 'General Settings', there are several fields: 'Contact Email' (systemadmin@sync.local), 'Hostname' (syncapp.com), 'Dual Hostnames' (checked), 'App Server Hostname' (web.syncapp.com), and 'Maintenance Mode' (unchecked). The 'App Server Hostname' field is highlighted with a red box. At the bottom right, there are 'CANCEL' and 'SAVE' buttons.

6. Click the **Save** button when you are finished.
7. Restart the *Anchor* and *Apache* services.

After these steps are complete, email links will use the web domain name, and agents will connect to the app domain name.

Finding Additional Support

Please contact us if you need additional support.

Call: 800-352-0248

Email: billing@axcient.com