SAFR® Large Scale Deployment

Documentation Version = 3.017

Publish Date = April 11, 2021

Contents

1 SAFR Server Clusters

At some point, your SAFR system's capacity and/or performance may degrade if the number of face recognition requests sent to your SAFR Server overwhelms your server's capacity. (Performance problems may also arise if the number of people in your Person Directory becomes too large.) Fortunately, you can install additional SAFR Servers on other machines in order to increase your SAFR system's capacity, improve performance, and improve resiliency. The first SAFR Server you install is automatically your primary server, while all additional servers are secondary servers.

In order to install additional servers, you must first install an SSL certificate on your primary server. See SSL Certificate Installation for information about how to do this.

Note: You can change which machine is the primary server by uninstalling the primary server, waiting 24 hours, and then re-installing the SAFR Server on a different machine. The 24 hour wait time can be avoided if you contact your SAFR Account Manager and ask them to manually reset your IP address.

1.1 Understand When to Scale

A single SAFR Server that's also running a Desktop Client can handle up to 16 cameras, (assuming each camera view contains just a single face), as long as the host machine meets the recommended hardware requirements If the machine running the server doesn't have any cameras connected directly to it, then the server's capacity increases to 25 cameras, again assuming that each camera view contains a single face. A higher number of faces per camera or a higher number of cameras requires either vertical scaling of a single server (i.e. more or faster CPUs) or horizontal scaling by installing more SAFR Servers.

Another possible performance bottleneck is the network throughput of the primary server. You may want to monitor its network throughput during maximum concurrency times to make sure the network is not over-saturated.

1.2 Load Balancing Configurations

For prescribed deployments, the system requirements of the Desktop Client need to be combined with those of SAFR Server. A single Desktop Client typically handles up to 16 cameras as long as it is equipped with a GPU card (see SAFR System Requirements). In this way, running SAFR Server and the Desktop Client on the same machine using the recommended configuration can host up to 16 cameras, assuming each camera view contains with a single face.

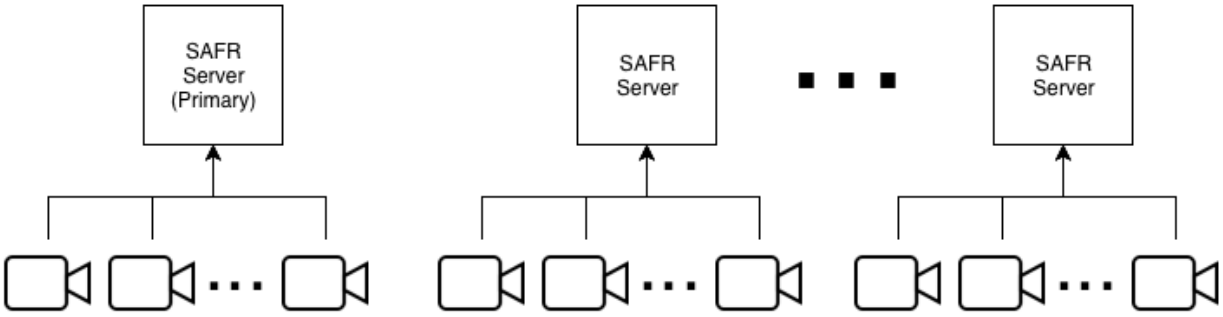
There are three different load balancing configurations you can choose from.

- **Prescribed Load Balancing Configuration**: Cameras are connected to Desktop Clients or Video Recognition Gateway (VIRGO) video feeds running on the same machines that are hosting your SAFR Servers. This gives you tight control over how your face recognition load is distributed, since the video feeds' face recognition requests are processed on the same machine where the video feeds are connected. The system requirements of the Desktop Client need to be combined with those of SAFR Server when calculating the system requirements for a machine hosting the SAFR Server and Desktop Client. A single Desktop Client typically handles up to 16 cameras as long as it is equipped with a GPU card (see SAFR System Requirements). Thus, a machine running SAFR Server and the Desktop Client which meets the recommended system requirements can host up to 16 cameras, assuming each camera view contains just one single face.
- **Software-Based Load Balancing Configuration**: In this configuration the machines hosting SAFR Servers do not also have cameras connected to them. All face recognition requests are initially sent to the primary server, and the primary server acts as the load balancer for the server cluster.
- **External Load Balancing Configuration**: In this configuration all face recognition requests are directed at one or more external load balancer(s), which handle load balancing duties for the SAFR system.

1.2.1 Prescribed Load Balancing Configuration

In the prescribed configuration, you run multiple SAFR Servers by connecting cameras to Desktop Clients or VIRGO video feeds running on the same machines that are hosting SAFR Servers. In this way, you have tight control over which servers take the video feed load. This is also a useful configuration for systems with very low video feed count totals where running a Desktop Client on a separate machine from the SAFR Server would take more resources than are required for the given use case.

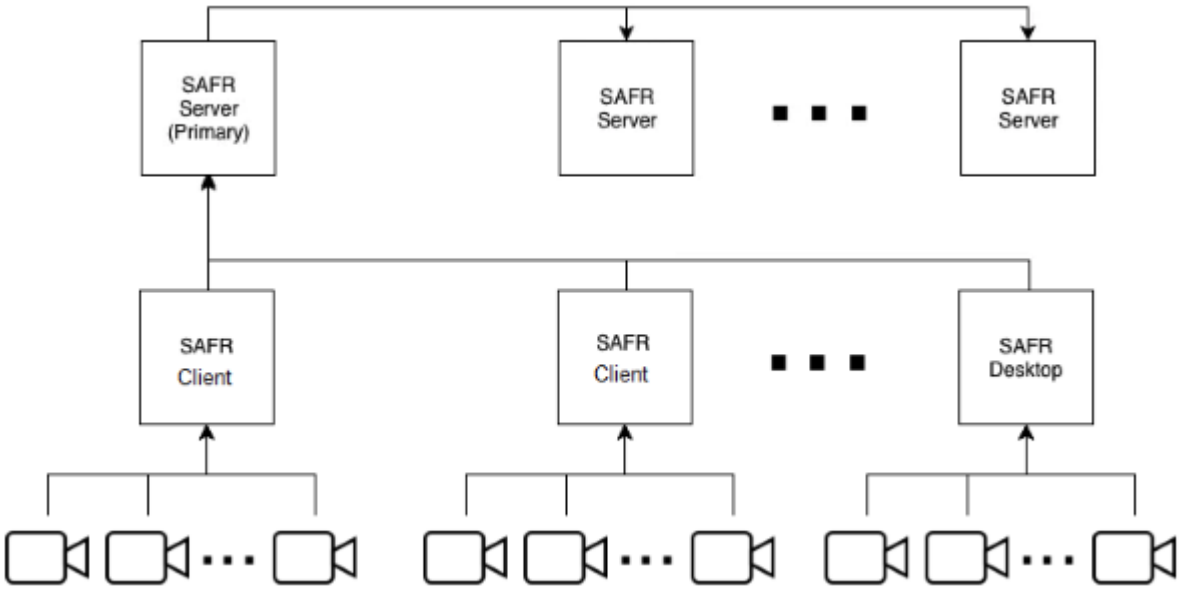
The following diagram illustrates this configuration:



Most services (e.g. face service, events, and reports) are performed on the server where recognition requests are sent.

1.2.2 Software-Based Load Balancing Configuration

In the software-based load balancing configuration, cameras aren't connected to machines running SAFR Servers. When newly installed secondary servers are configured, they check in with the primary server and announce that they're ready to receive load-balanced traffic. All recognition requests go through the primary server. which balances the load among itself and all other servers in the SAFR system. The following illustration demonstrates this setup:

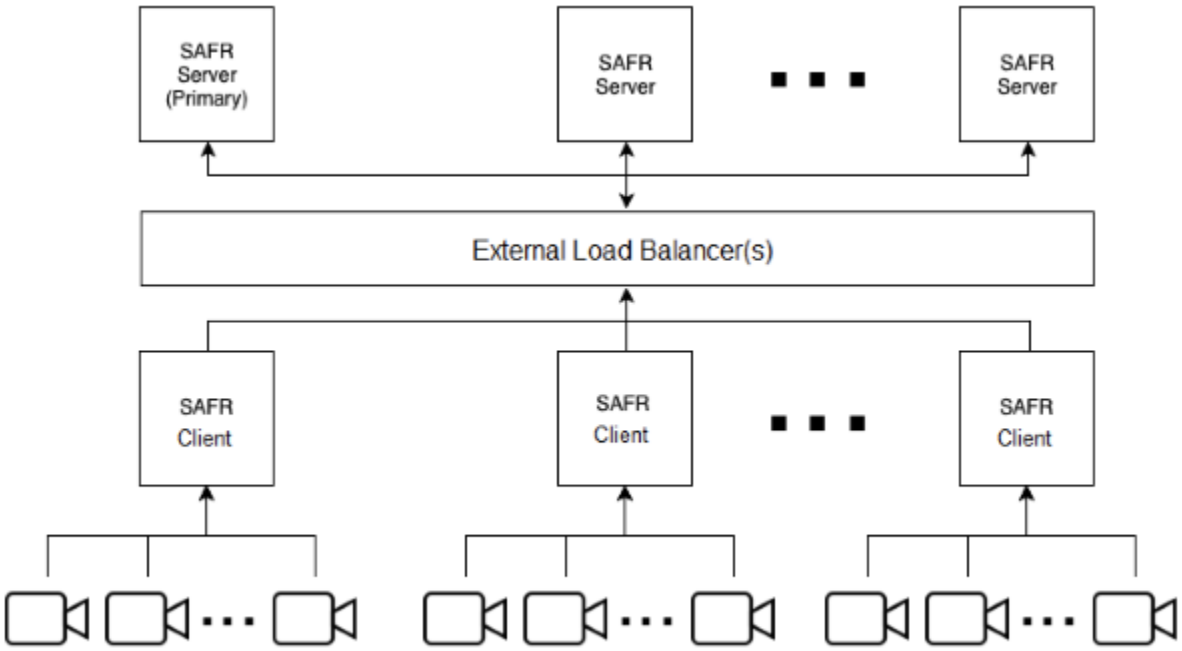


1.2.3 External Load Balancing Configuration

The software-based load balancing configuration has the limitation that the primary server is a single point of failure. All traffic is routed through the primary server before any traffic is redirected to the rest of the servers. If the primary server is down, all traffic will stop. External load balancing is an alternate configuration that can be used to provide a more robust setup that can better deal with server failure.

When using an external load balancing configuration, all network traffic is first routed to one or more load balancer(s), and the load balancer(s) proxy requests to the backend servers over either HTTP or HTTPS. HTTP would be OK in situations where network traffic is isolated to a trusted network, or when network sniffing by non-target hosts is impossible.

If HTTPS is used to proxy traffic to SAFR servers, you should manually disable load balancing on all secondary servers as described below so that the primary server isn't double load balancing traffic to them. A valid (i.e. non self-signed) SSL certificate would still need to be installed and configured on the primary server. Secondary servers should be fine with the default (i.e. self-signed) certificate, if your load balancer allows it.



1.2.4 Manually Configure Load Balancing

SAFR Servers can be manually enabled or disabled to accept load balancing traffic.
Note: If the server you want to disable is the only one configured to take traffic, you receive a warning and prompt to continue. In this case, should you proceed, your system will most likely go offline.

Disable Load Balancer Traffic

To stop receiving traffic on a server, log in to a shell on the server and run the appropriate command for your server's OS:

OS	Command
Windows	`"C:\Program Files\RealNetworks\SAFR\bin\server-status.py" --disable`

OS	Command
macOS	`/Library/RealNetworks/SAFR/bin/server-status.py --disable`
Linux	`sudo /opt/RealNetworks/SAFR/bin/server-status.py --disable`

It may take up to one minute for the desired traffic state to change.

Enable Load Balancer Traffic

To resume receiving traffic on a server, log in to a shell on the server and run the appropriate command for your server's OS:

OS	Command
Windows	`"C:\Program Files\RealNetworks\SAFR\bin\server-status.py" --enable`
macOS	`/Library/RealNetworks/SAFR/bin/server-status.py --enable`
Linux	`sudo /opt/RealNetworks/SAFR/bin/server-status.py --enable`

It may take up to one minute for the desired traffic state to change.

2 Add Secondary Servers

The first SAFR Server you install will automatically become the primary server. All subsequent servers you install will be secondary servers. There are two types of secondary servers:

- **Simple**: Does not replicate the database data.
- **Redundant**: Replicates database data from the primary server. If there are at least two redundant secondary servers (three servers total), fail-over functionality is enabled, which means that if the primary server is offline, the secondary servers will continue to function.
  Note: Only Windows and Linux SAFR Servers can become redundant secondary servers.

2.1 Add a Secondary Server While Connected to the Internet

If your system is connected to the Internet, do the following to add a secondary server:

1. Download and install SAFR Platform on the additional machine.
2. Log in to the he SAFR auto-discovery process:
   - Connect the Desktop Client to the primary server (for macOS and Windows) as described here.
   - Connect your Web Console to the primary server (for Linux) as described here.
3. During auto-discovery, the following automatically happens:
   1. The secondary server contacts a SAFR Licensing Server in the cloud to acquire a license.
   2. The SAFR Licensing Server authenticates the SAFR account credentials.
   3. The SAFR Licensing Server identifies the license and deployment type.
   4. A suitable license is returned to the secondary server and information about the primary server is returned to the secondary server, including the hostname.
4. If your new secondary server is on a Windows or Linux machine, you will be prompted to choose which kind of secondary server you want: simple or redundant. If your new secondary server is on a macOS machine no prompt will occur; macOS secondary servers are always simple.
5. Auto-discovery will now continue, with the following automatically occurring:
   1. The secondary server re-configures itself to reference the primary server.
   2. The secondary server registers itself with the primary server.
   3. The primary server updates its local database.
   4. If you're using a Software-Based Load Balancing Configuration, the primary server now adds the new secondary server to its load balancer configuration and uses it as an additional node in its cluster.

2.2 Add a Secondary Server While Offline

If you are not connected to the Internet, you can still connect your new secondary server to the primary server, but the auto-discovery process is not available. You must instead manually configure the newly installed secondary server to locate the primary server. If your new secondary server is on a Windows or Linux machine, you'll need to choose which kind of secondary server you want: simple or redundant. If your new secondary server is on a macOS machine no such decision is required; macOS secondary servers are always simple.

1. Download and install SAFR Platform on the second machine.
2. Run the *safr-worker* script on your secondary server by doing the following:

   For macOS:

   1. Open Terminal.
   2. Run the following command, substituting the primary SAFR hostname for HOSTNAME:

      ```
      sudo /Library/RealNetworks/SAFR/bin/safr-worker HOSTNAME
      ```

   For Windows

1. On the primary server record the contents of `C:\ProgramData\RealNetworks\SAFR\mongo\.adminpass` and `C:\ProgramData\RealNetworks\SAFR\mongo\mongod.keyfile`
2. On the new secondary server, open a command prompt by right-clicking on the **Start** menu, selecting **Run**, and entering `cmd`.
3. If you want it to be a simple secondary server, in the new command prompt run the following command, substituting the password from Step 1 for PASSWORD and the primary server hostname for HOSTNAME:

```
python "C:\Program Files\RealNetworks\SAFR\bin\safr-worker.py" -p
   PASSWORD HOSTNAME
```

   OR

4. If you want it to be a redundant secondary server, in the new command prompt run the following command, substituting the `mongod.keyfile` contents from Step 1 for KEYFILE, the password from Step 1 for PASSWORD, and the primary server hostname for HOSTNAME:

```
python "C:\Program Files\RealNetworks\SAFR\bin\safr-worker.py" -s KEYFILE
   -p PASSWORD HOSTNAME
```

For Linux

1. On the primary server, record the contents of `/opt/RealNetworks/SAFR/mongo/.adminpass` and `/opt/RealNetworks/SAFR/mongo/mongod.keyfile`
2. If you want it to be a simple secondary server, on the new secondary server run the following command, substituting the password from Step 1 for PASSWORD and the primary server hostname for HOSTNAME:

```
sudo python /opt/RealNetworks/SAFR/bin/safr-worker.py -p PASSWORD HOSTNAME
```

   OR

3. If you want it to be a redundant secondary server, on the new secondary server run the following command, substituting the `mongod.keyfile` contents from Step 1 for KEYFILE, the password from Step 1 for PASSWORD, and the primary server hostname for HOSTNAME:

```
sudo python /opt/RealNetworks/SAFR/bin/safr-worker.py -s KEYFILE -p
   PASSWORD HOSTNAME
```

2.3 Error Messages

When attempting to join a new secondary server, you might encounter the following error messages:

Error Message	Description
System is offline	Network or system connectivity issue. Attempt to access the system at a later time.
SAFR master host is not reachable	Ensure all servers are connected to the same network and try again.
Improperly configured SSL certificate	SSL certificates are required to set up multiple servers. See the SSL Certificate Installation page for information about how to install an SSL certificate.
Secure connection error. Check server for valid SSL certificate	SSL certificates are required to set up multiple servers. See the SSL Certificate Installation page for information about how to install an SSL certificate.

Error Message	Description
Incomplete server connection	Attempt to join again; a persistent issue may require either uninstalling and reinstalling SAFR Platform on your servers or contacting your SAFR support representative.

2.4 Secondary Server Health Checks

- At startup each server, both primary and secondary, registers itself by posting its status to the database on the primary server.
- The primary server directs requests to all secondary servers in a *least connection method* that keeps the load evenly balanced among all secondary servers.
- As long as a given secondary server remains healthy, the primary server keeps that secondary server in its load balance rotation.
- Status information about all secondary servers is stored in the database on the primary server.
- Every minute all servers (the primary server as well as the secondary servers) send a status update to the database on the primary server.
- Every five seconds, the primary server attempts to ping all servers (the primary server as well as all secondary servers) via the SAFR heath check API.
- If the health check fails for a given secondary server for 15 seconds (i.e. for 3 health check API calls in a row), that secondary server is removed from load balancing rotation and face recognition requests are no longer routed to it. If the health check succeeds for the removed secondary server for ten seconds (i.e. for 2 health check API calls in a row), the secondary server is returned to the load balancing rotation and resumes accepting face recognition requests.
- If a secondary server's status has not been reported for over five minutes, it is removed from the load balancer configuration. In this case, it is no longer sent face recognition requests or health check API calls.
- If a secondary server has been pulled out of rotation for not responding to health checks, or is removed from the load balancer configuration for not reporting status for more than five minutes, it can still be put back in rotation through any of the following:
  - If a network interruption prevents the secondary server from sending a request, the secondary server continues to send a status update at its regularly scheduled interval after it goes back online and its status is updated in the primary server.
  - If the secondary server is restarted, it sends a status update after all services are started and ready.
  - If the secondary server IP address is changed, the secondary server must be manually restarted to force it to send a status update to the primary server with the new IP address.

3 Database and Object Storage Redundancy

3.1 Database Redundancy

The first SAFR Server you install will automatically become the primary server. All subsequent servers you install will be secondary servers. There are two types of secondary servers:

- **Simple**: Does not replicate database data.
- **Redundant**: Replicates CoVi and Event database data from the primary server. If there are at least two redundant secondary servers, fail-over functionality is enabled, which means that if the primary server is offline, the secondary servers will continue to function.
  Note: Only Windows and Linux SAFR Servers can become redundant secondary servers.

With both types of secondary servers services such as feed management, reports, and the Web Console are not load-balanced and are always served from the primary server.

3.2 Object Storage Redundancy

Note: Object Storage Redundancy is only available on Windows and Linux.

The Object Storage Service is used for storing objects, such as profile and event images, as well as ephemeral data, such as event reply messages.

The service can operate in a redundant configuration when you have multiple SAFR servers running. All redundant secondary servers are load-balanced by the primary server for all Object Storage Service requests it receives.

3.2.1 Shared Object Storage (Network Storage)

Using shared object storage provides a shared location for each server to save and retrieve objects from. This provides each Object Storage Server with access to all of the objects, rather than just objects saved to their local storage.

Shared storage also provides an easier backup process, as you only have to run it from the primary server.

3.2.2 Local Object Storage (Not Recommended)

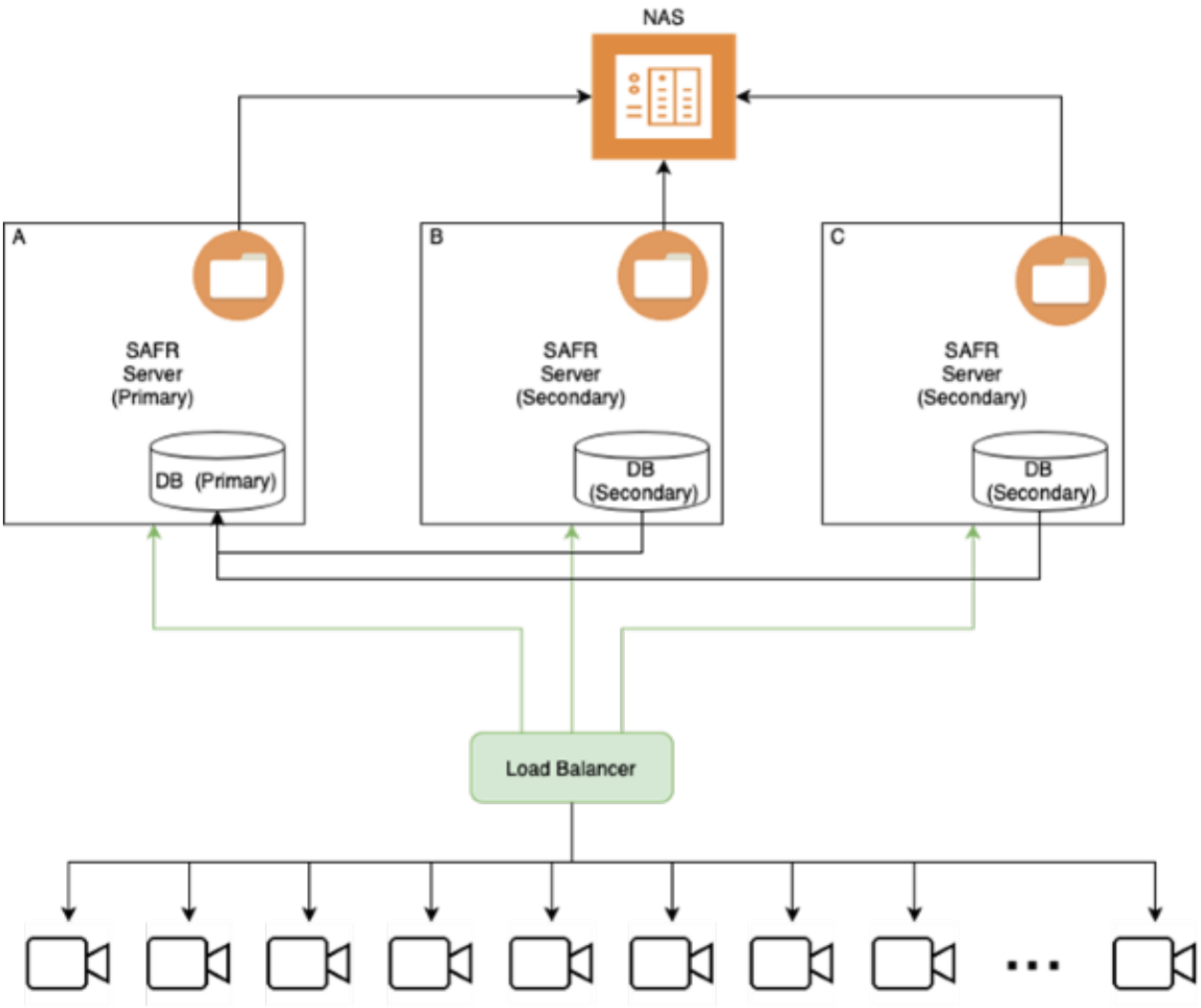
By default all redundant servers will save objects locally, and ask other Object Storage Servers for objects it does not have locally.

When you're using local object storage, you will lose access to all objects that are only stored by an offline Object Storage Server until the server becomes healthy again. If that server's objects are lost, and you do not have backups, they will be unrecoverable.

Backups must be run on every redundant server that has Object Storage enabled.

3.3 External Load Balancing (ELB) Walkthrough

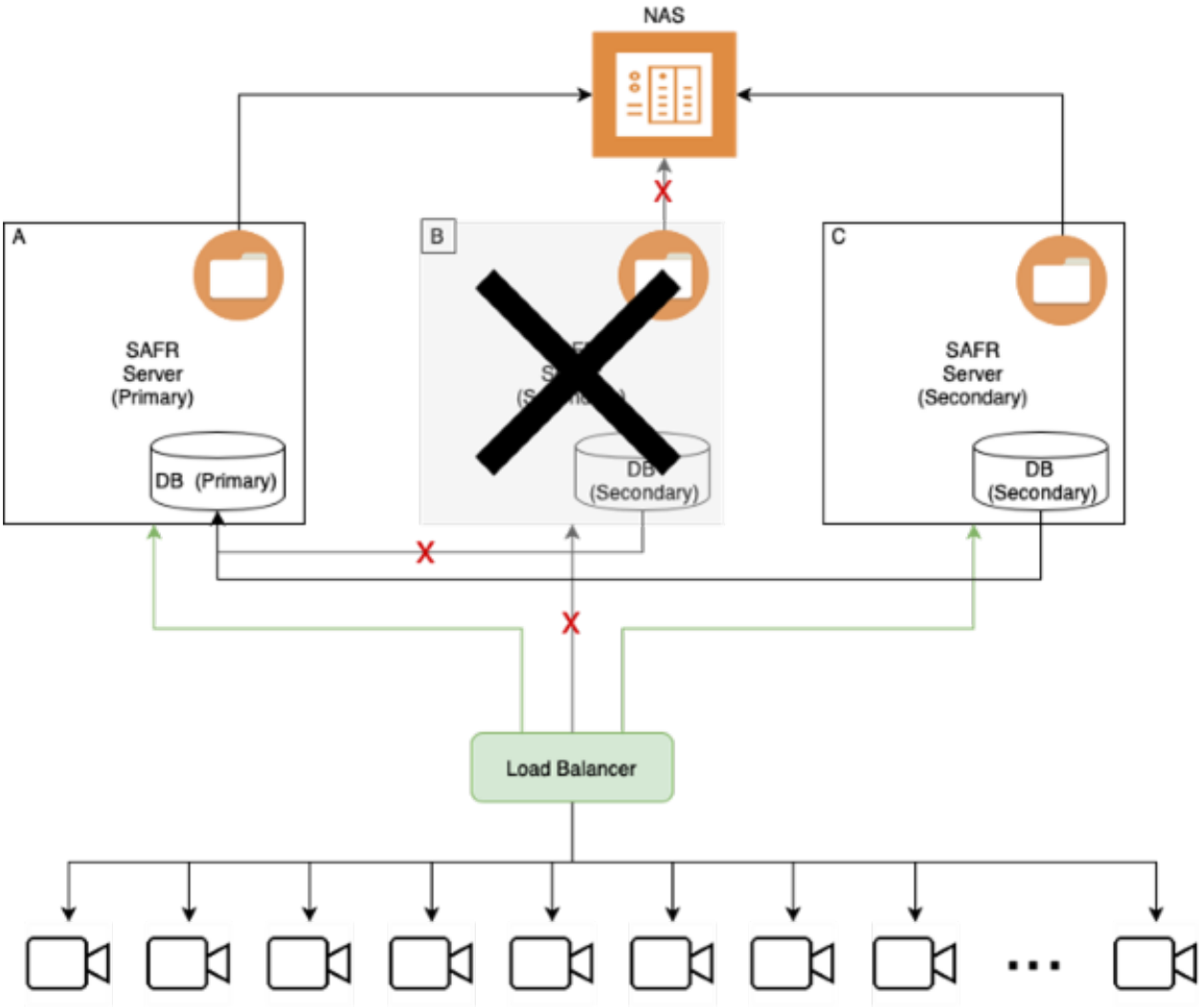
This section describes the functionality of a common large scale deployment configuration: an external load balanced system with 2 redundant secondary servers and a network-attached storage (NAS), a type of shared object storage.



When all the servers are working, this configuration has the following properties:

- Face recognition requests are distributed across all three servers.
- Server A's database is the primary database (i.e. performs database writes).
- All three server share in the database read load.
- All three servers read and write directly to the NAS.
- There isn't a single point of failure.
- Data is redundantly protected across all three servers.

3.3.1 Secondary Server Failure in an ELB Configuration



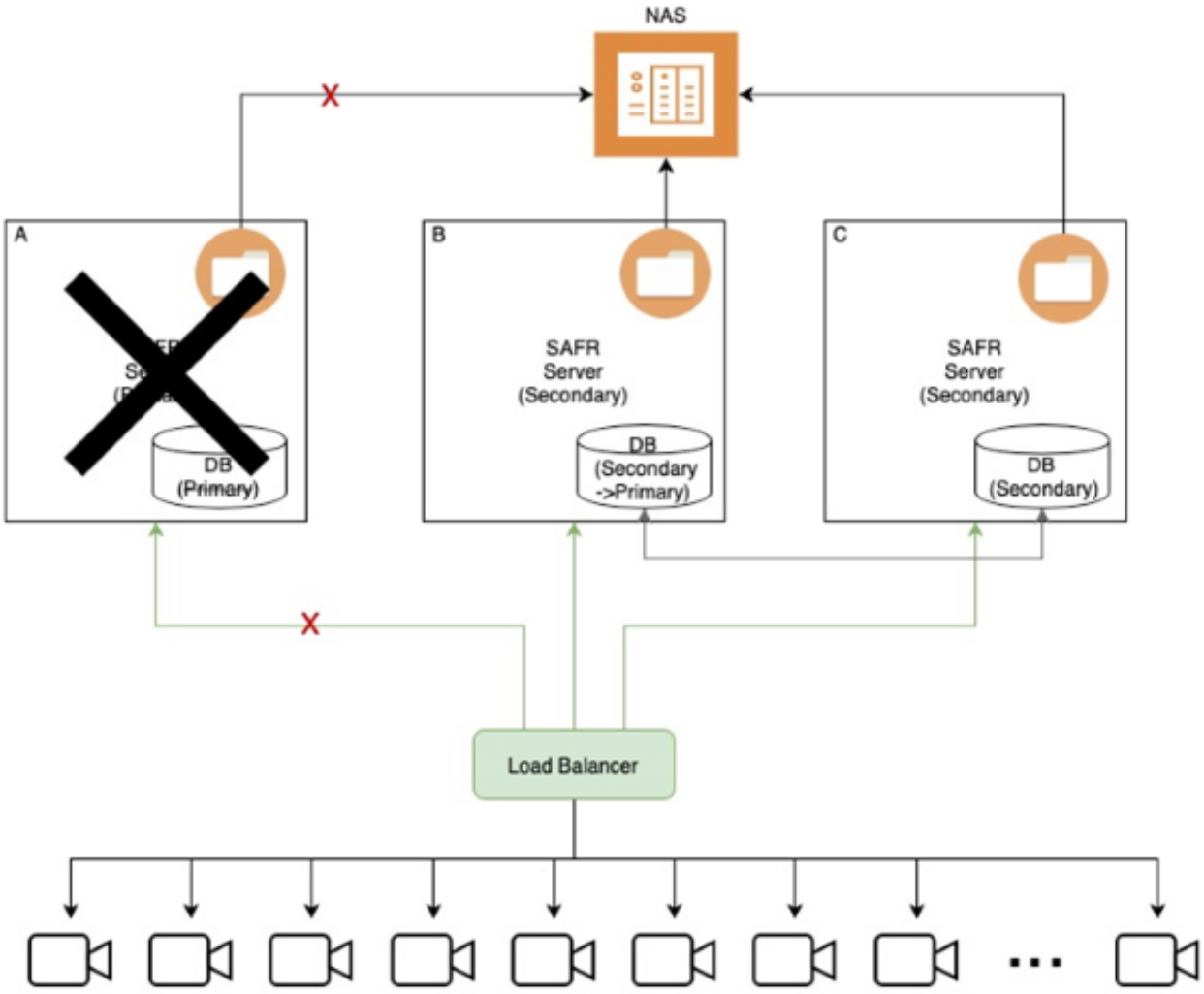
When one of the redundant secondary servers fail, the following occurs:

- Face recognition requests are handled by servers A (primary) and C (secondary).
- Server B is removed from rotation by the load balancer.
- Server B is removed from the database replica set.
- Server A's database continues operating as the primary database (i.e. it performs database writes).
- Servers A and C share in the database read load.
- Both remaining servers read and write directly to the NAS.

Impact:

- No service outage occurs, but longer latency may result.
- Data remains redundantly protected.
- There isn't any impact to object storage.

3.3.2 Primary Server Failure in an ELB Configuration



When the primary server fails, the following occurs:

- Face recognition requests are handled by servers B (secondary) and C (secondary).
- Server A is removed from rotation by the load balancer.
- Server A is removed from the database replica set.
- The server B database takes over as the primary database (i.e. it performs database writes).
- Servers B and C share in the database read load.
- Both remaining servers read/write directly to the NAS.

Impact:

- No service outage occurs, but longer latency may result.
- Data remains redundantly protected.
- There isn't any impact to object storage.

3.4 Migrate from Local to Shared Storage

If you start with local storage but later decide to move to shared storage, you will need to consolidate all of your objects to the new shared storage location, delete the local copies, and then mount the shared storage to the correct location. To do this, do the following:

1. Back up both the primary and redundant secondary servers to ensure you have a full backup of all SAFR content.
   - **On Linux**:
     - **Primary**: `python /opt/RealNetworks/SAFR/bin/backup.py`
     - **Redundant Secondaries**: `python /opt/RealNetworks/SAFR/bin/backup.py -o`
   - **On Windows**:
     - **Primary**: `python "C:\Program Files\RealNetworks\SAFR\bin\backup.py"`
     - **Redundant Secondaries**: `python "C:\Program Files\RealNetworks\SAFR\bin\backup.py" -o`
2. Stop all primary and redundant secondary servers by using the **stop** command. This can be done by doing the following on each server:
   - **On Linux**: `/opt/RealNetworks/SAFR/bin/stop`
   - **On Windows**: `"C:\Program Files\RealNetworks\SAFR\bin\stop.bat"`
3. Mount the new shared storage to a temporary location on primary and redundant secondary servers.
4. Copy all files from the primary server and every redundant secondary server(s) to the temporary location of the shared storage. from within the following paths:
   - **On Linux**: `/opt/RealNetworks/SAFR/cv-storage`
   - **On Windows**: `C:\ProgramData\RealNetworks\SAFR\cv-storage`
5. Delete or move the contents of the CV Storage folder on each primary and redundant secondary server as specified below.
   - **On Linux**: `/opt/RealNetworks/SAFR/cv-storage`
   - **On Windows**: `C:\ProgramData\RealNetworks\SAFR\cv-storage`
6. Unmount the temporary location of the new shared storage.
7. Mount the shared storage to the correct CV Storage location, or create a symlink to the shared storage location.
8. Start the primary and redundant secondary servers by using the **start** command. On each server, do the following:
   - **On Linux** `/opt/RealNetworks/SAFR/bin/start`
   - **On Windows** `"C:\Program Files\RealNetworks\SAFR\bin\start.bat"`
9. Disable any automatic backups on redundant secondary servers.
   - Now that you're using shared storage, only the primary server needs to be backed up. Disable any automatic backups you may have configured on your secondary servers.

3.5 Simple Secondary Server Behavior with Local Object Storage

On simple secondary servers, the Object Storage Service will operate in proxy mode.

Object Storage Servers operating in proxy mode will not attempt to use their own storage for objects, but will instead proxy the request to Object Storage Services that are running on either the primary server or on a redundant secondary server. If the redundant server it contacts doesn't have the object, the contacted redundant server will ask all other redundant servers for the object.

The list of servers that run the Object Storage Service is stored in the database and updated every minute. If a host does not respond within a timeout, it is de-prioritized.

3.6 Redundant Secondary Server Behavior with Local Object Storage

On both the primary server and on redundant secondary servers the Object Storage Service stores new objects in storage.

When a server receives a request for a file it does not find in its storage, it will request the object from other Object Storage Servers via HTTPS, and return the object if found. (The same applies for DELETEs.) This allows multiple Object Storage Servers to operate without using shared network storage, with each server saving a subset of the total objects, and relaying requests for other objects to its neighbors.

Even when using shared network storage, sometimes a request will come in for a new object before it is

visible to all systems on the shared storage. The Object Storage Service will ask all the other Object Storage Servers for the object until it finds one that has the object.

3.7 Backup and Restore

The SAFR backup and restore process when using shared network storage is straightforward - you just need to back up the primary server. This will back up all configs, database content, and Object Service Storage objects.

When using local storage, however, the objects are distributed to multiple servers, so the backup must be run on the primary server as well as all redundant secondary servers.

The primary server should run a regular backup, while the redundant secondary servers run an '*objects only*' backup. The difference is just the addition of the "**-o**" flag to the backup script.

When restoring multiple backups, you can restore them all to the primary server, or you can restore the '*object only*' backups back to the same servers that they were backed up from.

3.7.1 Backup for Local Storage

- **On Linux**
  - **Primary**: `python /opt/RealNetworks/SAFR/bin/backup.py`
  - **Redundant Secondaries**: `python /opt/RealNetworks/SAFR/bin/backup.py -o`
- **On Windows**
  - **Primary**: `python "C:\Program Files\RealNetworks\SAFR\bin\backup.py"`
  - **Redundant Secondaries**: `python "C:\Program Files\RealNetworks\SAFR\bin\backup.py" -o`

3.7.2 Restore for Local Storage

- **On Linux**
  - **Primary**: `python /opt/RealNetworks/SAFR/bin/restore.py BACKUPFILENAME`
  - **Redundant Secondaries**: `python /opt/RealNetworks/SAFR/bin/restore.py -o BACKUPFILENAME`
- **On Windows**
  - **Primary**: `python "C:\Program Files\RealNetworks\SAFR\bin\restore.py" BACKUPFILENAME`
  - **Redundant Secondaries**: `python "C:\Program Files\RealNetworks\SAFR\bin\restore.py" -o BACKUPFILENAME`

3.8 Example Shared Storage Configurations

Below are two example shared storage configurations.

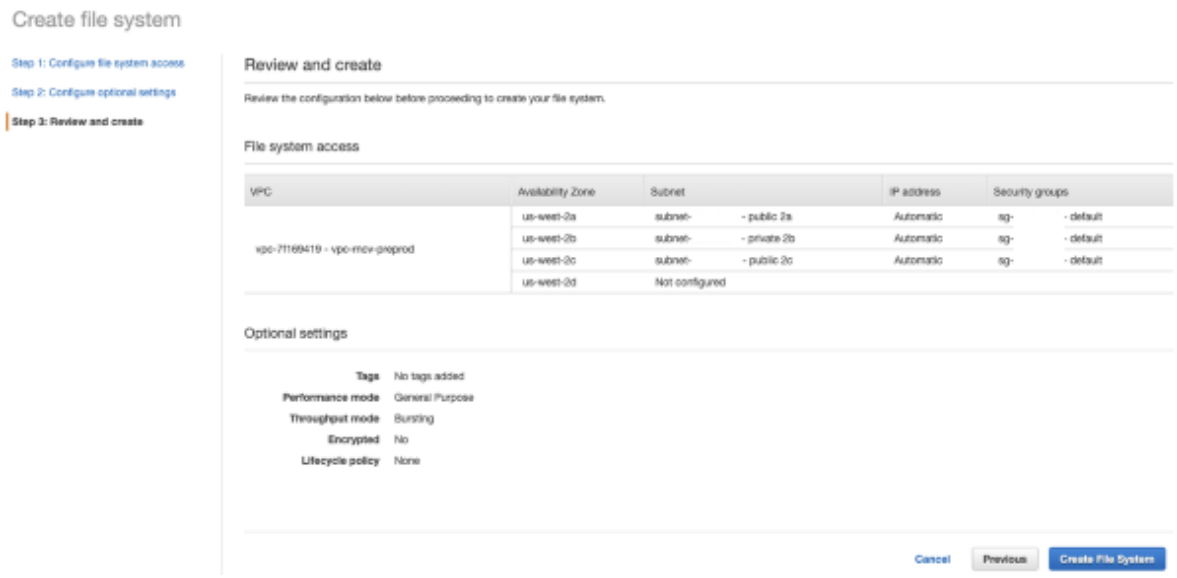
3.8.1 Linux

Shared storage on Linux is very straightforward. Simply mount your shared storage to the `/opt/RealNetworks/SAFR/cv-storage` location.

1. Stop SAFR.

   ```
   /opt/RealNetworks/SAFR/bin/stop
   ```

2. Create a shared storage location. The example below uses Amazon's Elastic File System (EFS).



3. Edit `/etc/fstab` to create a mount point of `/opt/RealNetworks/SAFR/cv-storage` for your shared storage. The specific mount options should be provided by your specific storage service or device.

```
fs-12345678.efs.us-west-2.amazonaws.com:/
   /opt/RealNetworks/SAFR/cv-storage nfs4
   nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,_netdev
   0 0
```

4. Mount the remote share.

```
sudo mount -a
```

5. Start SAFR.

```
/opt/RealNetworks/SAFR/bin/start
```

3.8.2 Windows

Windows cannot mount a shared storage location directly to `C:\ProgramData\RealNetworks\SAFR\cv-storage`. It must instead create a symbolic link by doing the following:

1. Stop SAFR.

```
"C:\Program Files\RealNetworks\SAFR\bin\stop.bat"
```

2. Create a shared storage location.
3. Delete the existing `C:\ProgramData\RealNetworks\SAFR\cv-storage` by running `rmdir /q /s C:\ProgramData\RealNetworks\SAFR\cv-storage` in an administrative command prompt. Deleting the existing cv-storage allows you to create a symbolic link from the *cv-storage* location to your shared storage location.
   Note: Be sure you either followed the migration steps above to consolidate your objects onto the new shared storage location, or that you're doing this on a new system without any data.
4. Create the symbolic link from `C:\ProgramData\RealNetworks\SAFR\cv-storage` to your shared storage location. To do this, run the appropriate command in an administrative command prompt:
   - If you're using a mapped network drive, run

     ```
     mklink /d C:\ProgramData\RealNetworks\SAFR\cv-storage Z:\
     ```

- If you're using an SMB share, run

  ```
  mklink /d C:\ProgramData\RealNetworks\SAFR\cv-storage \\servername\share
  ```

5. Start SAFR.

   ```
   "C:\Program Files\RealNetworks\SAFR\bin\start.bat"
   ```

4 SSL Certificate Installation

A properly installed secure sockets layer (SSL) certificate is critical to the secure operation of your SAFR Server. SAFR uses SSL certificates to establish secure network connections and data transfers. (i.e. https connections) SAFR requires https connections between SAFR Servers and between SAFR Servers and iOS Mobile Clients. None of the other SAFR components require https connections.

Before you can install an SSL certificate on your SAFR Server, you must first configure a Domain Name System (DNS) hostname for your server within your network domain, as described below.

4.1 DNS Hostnames

If you do not currently have a domain, you need to first obtain a domain name registered and configured with an accredited domain registrar.

4.1.1 How to Obtain a Domain Name

In order to set up a DNS, you need a domain within which you can register hostnames. ICANN maintains a list of accredited registrars from which to choose.

The following is a list of common registrars:

- GoDaddy
- Google Domains
- AWS
- HostGator

Follow the processes on these websites to find, purchase, and configure your domain name. Most registrars offer the ability to host your DNS for you and most also give you a web interface for managing it.

The following links lead to instructions on how to modify DNS entries:

- GoDaddy
- Google Domains
- AWS
- HostGator

After you have your domain, you can create a DNS hostname entry for your SAFR Server.

4.1.2 What a DNS Hostname Entry Does

DNS is a system that translates a hostname to a network IP address. For example, when a user types `www.example.com` into their browser, DNS servers resolve it to the IP address where the website is hosted.

To provide this translation, DNS requires an entry for each hostname. This entry typically takes the form of an *A record* (the A stands for "Address") which defines the hostname to IP address translation in DNS. An *A record* is the most basic type of syntax used in DNS records.

The following is an example of an *A record*:

```
safr.example.com     A    12.34.56.78
```

4.1.3 Set Up a DNS Hostname Entry for your Primary Server

DNS can be managed in numerous ways. This might be a text file or a web interface for configuring the DNS entries. If you are not sure, contact the person managing the domain name for your network.

4.1.4 What Type of IP Address Should I Use?

You should use a static IP address. If you instead choose to use DHCP to get a dynamically assigned IP address, and your IP address happens to change, your DNS hostname entry will stop working until you update the entry.

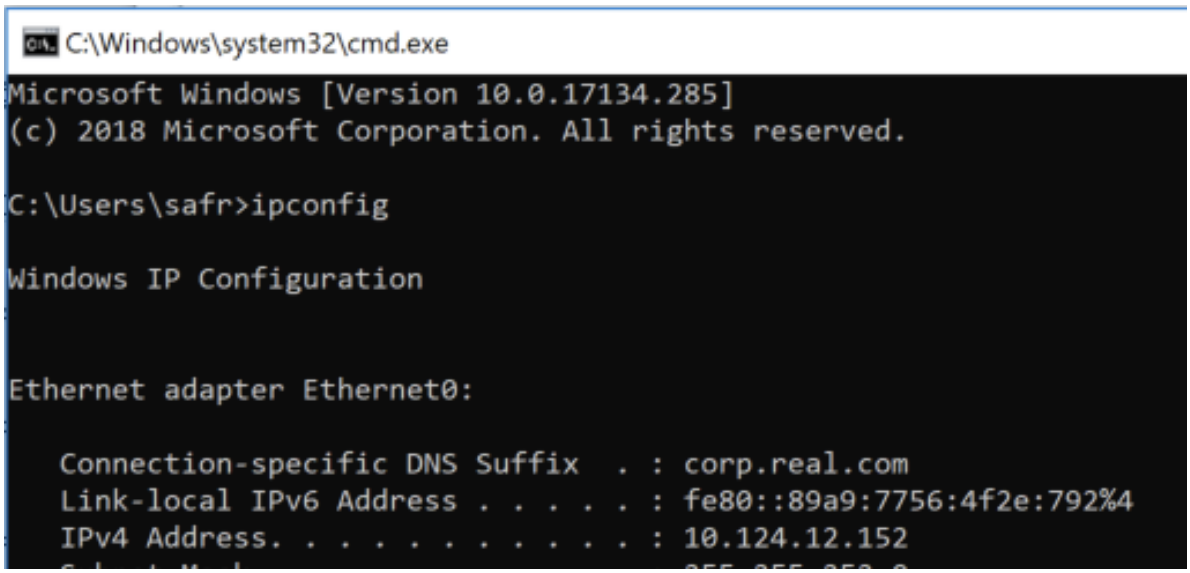
4.1.4.1 Configure a Static IP

1. Obtain a static IP from your network administrator. The information should include the following:
   - Static IP address
   - Subnet mask
   - Default gateway
2. Configure your system as described below:
   - For Windows, see https://www.howtogeek.com/howto/19249/how-to-assign-a-static-ip-address-in-xp-vista-or-windows-7/.
   - For macOS, see https://www.howtogeek.com/howto/22161/how-to-set-up-a-static-ip-in-mac-os-x/.

The IP address should be the internal IP address of the computer running the SAFR Server. This should not be your public IP address because the public IP address usually points at your router, modem, or similar device. The internal IP address is the IP used locally by the computer. It can be determined by doing the following:

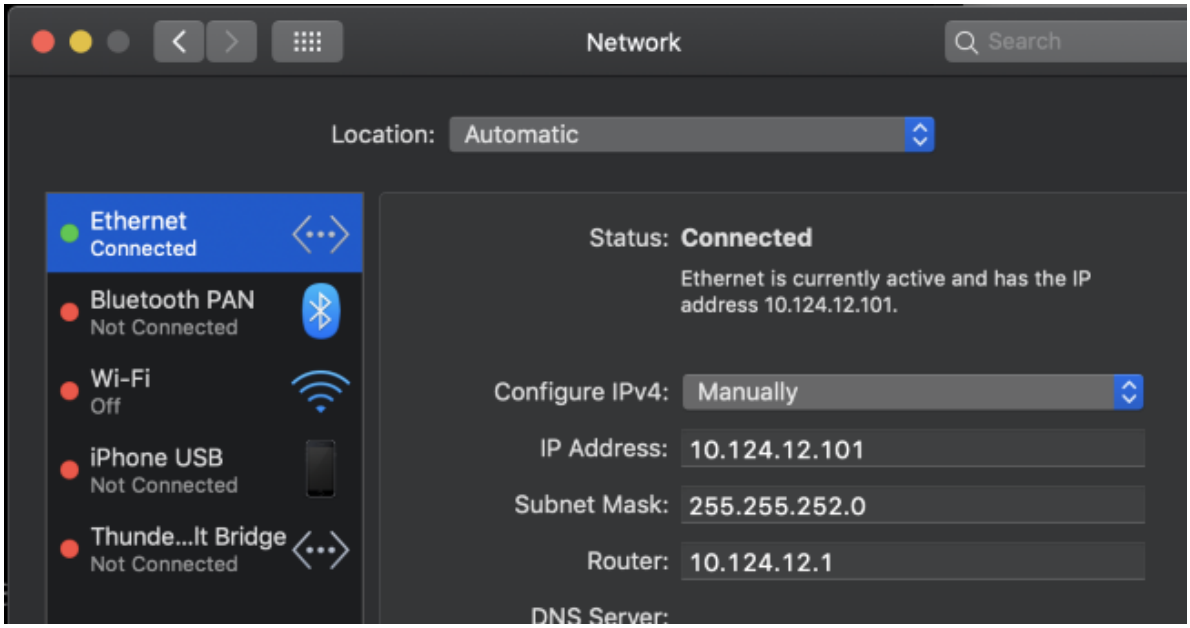
For Windows 10

1. Open a command prompt (cmd.exe).
2. Run ipconfig.
3. The IP address is listed as the IPv4 Address.



For macOS

1. Open **System Preferences**.
2. Open **Network**.
3. Click the active network connection (usually WiFi or Ethernet).
4. The IP address is displayed in the dialog.



4.2 SSL Certificates

After you have configured a DNS hostname for your primary server, you can now install an SSL certificate.

4.2.1 What an SSL Certificate Does

SSL certificates are small data files that digitally bind a cryptographic key to an organization's information. When installed on a server, an SSL certificate allows secure connections from the server to a browser or other program and protects sensitive information.

A common use for SSL certificates is to enable a web server to provide a secure connection with a web browser (i.e. an https:// connection instead of an http:// connection).

4.2.2 Obtain an SSL Certificate

SSL certificates need to be issued from either a trusted certificate authority or from an accredited domain registrar.

Browsers, operating systems, and mobile devices maintain lists of trusted certificate authority root certificates, which must be present on a computer for it to trust the certificate.

The following is a list of popular certificate authorities from which you can obtain an SSL certificate:

- Comodo
- IdenTrust
- GoDaddy
- GlobalSign
- Digicert
- Certum
- Entrust

Go to ICANN for a complete list of accredited domain registrars.

Because SAFR uses Apache as its web server, request SSL certificate files for Apache web server. You will receive the following three files SAFR uses to configure the Apache web server:

- **Key**: This is your key file and should not be shared publicly.
- **Certificate**: The SSL certificate for your domain.

- **Ca_bundle**: Signer root/intermediate certificate. This file is optional; it's not always provided by the SSL certificate provider.

Note: Self-signed certificates do not work.

4.2.3 Provision SSL Certificates for your Primary Server

Do the following to configure Apache to serve the request over HTTPS:

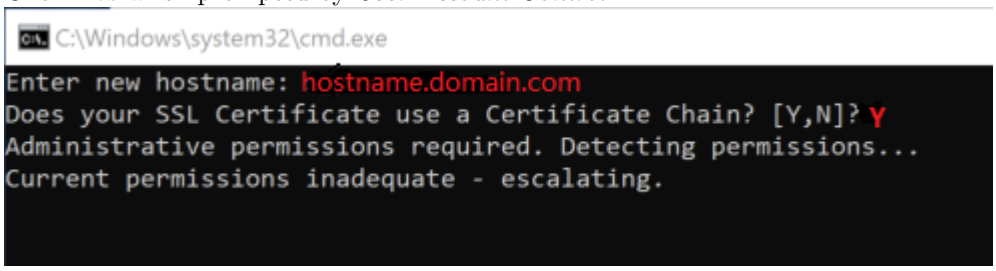
1. Log in to your primary server.
2. It is recommended that you make a backup of the default SSL files and save them in case you need to perform a rollback to the earlier version.
   - On macOS, back up the following files:
     - `/etc/apache2/ssl/SAFR.key`
     - `/etc/apache2/ssl/SAFR.crt`
   - On Windows, back up the following files:
     - `C:\Program Files\RealNetworks\SAFR\httpd\conf\ssl\SAFR.key`
     - `C:\Program Files\RealNetworks\SAFR\httpd\conf\ssl\SAFR.crt`
   - On Linux, back up the following files:
     - `/opt/RealNetworks/SAFR/httpd/ssl/SAFR.key`
     - `/opt/RealNetworks/SAFR/httpd/ssl/SAFR.crt`
3. Upload the certificate-related files to the SSL certificate folder:
   - SSLCertificateFile – Certificate CRT
   - SSLCertificateKeyFile – `Private.a` key file
4. Change the names of the following files:
   - Rename *_certificate.crt to SAFR.crt
   - Rename *_private.key to SAFR.key
5. If your certificate authority provided an intermediate certificate chain, do the following:
   1. Save your SSL intermediate certificate chain file to the following location:
      - **On macOS**:
        - `/etc/apache2/ssl/SAFR-ca.crt`
      - **On Windows**:
        - `C:\Program Files\RealNetworks\SAFR\httpd\conf\ssl\SAFR-ca.crt`
      - **On Linux**:
        - `/opt/RealNetworks/SAFR/httpd/ssl/SAFR-ca.crt`
   2. Check the `SAFR-ssl-cert.inc` file to connect your SSL certificate to the certificate chain.
      - **On macOS**:
        - `/etc/apache2/other/SAFR-ssl-cert.inc`
        - `#Define ssl_certificate_chain_file "/private/etc/apache2/ssl/SAFR-ca.crt"`
      - **On Windows**:
        - `C:\Program Files\RealNetworks\SAFR\httpd\conf\ssl\SAFR-ca.crt`
        - `#Define ssl_certificate_chain_file "conf/ssl/SAFR-ca.crt"`
      - **On Linux**:
        - `/opt/RealNetworks/SAFR/httpd/ssl/SAFR-ca.crt`
        - `#Define ssl_certificate_chain_file "/opt/RealNetworks/SAFR/httpd/ssl/SAFR-ca.crt"`
      - Certificate file mappings

Certificate file	Certificate file in SAFR
*.domainname.key	SAFR.key
*.domainname_chain.crt	SAFR-ca.crt

Certificate file	Certificate file in SAFR
*.domainname_public.crt	SAFR.crt

6. Run the SAFR reconfigure script, as described below.

   - **On macOS**:
     - Open **Applications > Utilities > Terminal** to open a Terminal window.
     - Run the following command after replacing hostname.domain.com with your hostname and domain:
       - `/Library/RealNetworks/SAFR/bin/reconfigure hostname.domain.com`
   - **On Windows**:
     - Enter this command: `"C:\Program Files\RealNetworks\SAFR\bin\reconfigure.bat"`
     - Enter the hostname and click **Yes** when prompted if your SSL certificate uses a certificate chain.
     - Click **Yes** when prompted by *User Account Control*.
   - **On Linux**:
     - Open a Terminal window. Run the following command after replacing hostname.domain.com with your hostname and domain:
       - `/opt/RealNetworks/SAFR/bin/reconfigure hostname.domain.com`
     - Click **Yes** when prompted by *User Account Control*.



7. Verify that your services are running and your SSL certificate is properly installed by opening a browser and opening `https://hostname.domain.com:8085/health`. (Replace hostname.domain.com with your hostname and domain.)

   You should receive the following message:

```
{ "status" : "up" }
```

4.3 Troubleshoot

Database Service Down

Problem: You receive an error report saying Database (MongoDB) Service Down when you run the **check** command after you install SSL.

Solution: The cause may be that the DNS hostname IP is different from the IP when you installed SAFR without SSL installed.

Use the following workaround:

Add the following line to your primary server /etc/hosts file:

```
127.0.0.1 hostname.domain.com
```