# Wickr Enterprise NIAP Version

## *Installation and Maintenance*

Version 1.30.0

# Table of Contents

This document describes how to install, deploy, and manage Wickr Enterprise NIAP Version on self-hosted infrastructure. It can be installed with or without internet access. It covers the Base services, the optional Voice and Video service, and proxies for either.

# 1. Changelog

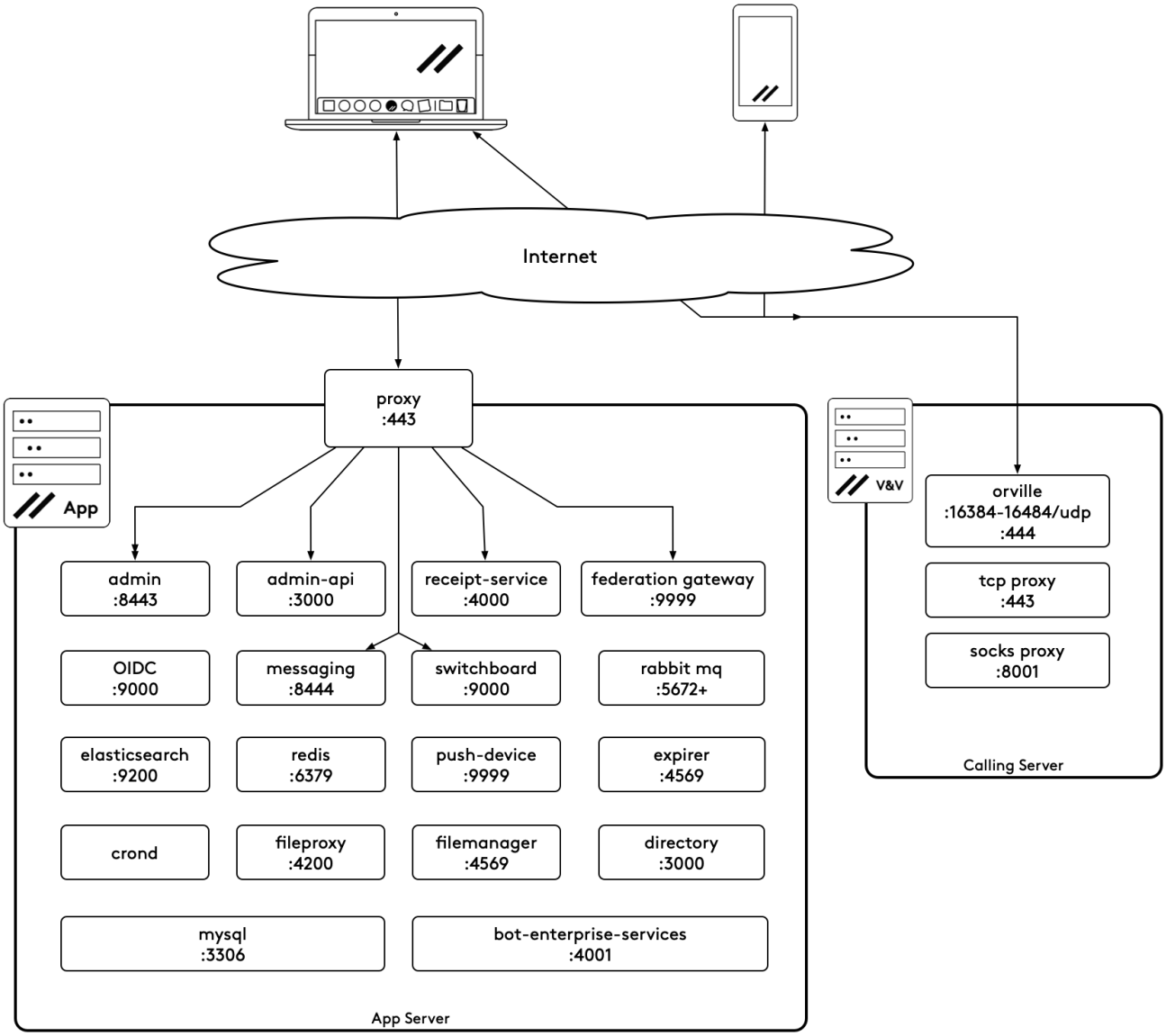
1.30.0

Adds NIAP specific instructions

> 1.2.0
>
> - Adds security and device management recommendations
>
> 1.1.1
>
> - Updates infrastructure diagrams with fileproxy container
> - MySQL schema updates are now handled with a schema container. It is only used during upgrades and is stopped otherwise.
>
> 1.1.0
>
> - Updates infrastructure diagrams
>
> 1.0.9
>
> - Calling containers have changed. New diagrams added.
> - Calling port range can be adjusted to allow for more calls
> - Calling log locations have changed
> - Upgrades can fail when using specific replicated versions. Fixes added to the Troubleshooting Updates section.
>
> 1.0.8
>
> - Added Troubleshooting Updates for potential issues upgrading to release 501
> - Adds instructions for Restoring a backup

# 2. Deploy Overview

The diagram below shows an installation composed of both Base and Voice and Video.




# 3. Requirements

Wickr Enterprise requires a single server for the Base services. The optional services, Voice and Video and/or Compliance, require their own servers. At most Wickr Enterprise would require three servers if using all services.

Enterprise also requires a license file (.rli) to complete the installation. Specific requirements are below:

## 3.1. Messaging Server

The Messaging server will need a Linux server with the following specifications. These numbers assume 200 to 300 users with moderate usage. Increase the storage space if transferring large files with long expiration times.

| Resource | Recommendation |
|---|---|
| **OS** | Ubuntu 20.04 |
| **CPU** | 2+ Cores/vCPUs |
| **RAM** | 8GB+ |
| **Disk Space** | OS Drive Block device A 60GB+<br>Encrypted Drive Block device B 120GB + UNFORMATTED |

Recommended disk space requirements will vary based on the amount of space you wish to have available for file uploads. We recommend 1-5GB per user if using the standard 30 day retention. If you are unsure what to set Wickr Support can help you calculate this number.

The Encrypted Drive Block device will be reformatted! It is recommended that this drive be unformatted.

### 3.1.1. Software Requirements

| Resource | Recommendation |
|---|---|
| **Docker Version** | 20.10.17 |
| **AppArmor** or **SELinux** | Disabled or in permissive mode |

### 3.1.2. Networking Requirements

| Protocol | Port(s) | Network Range | Purpose |
|---|---|---|---|
| TCP | 443 | ALL | Wickr Services |
| TCP | 22 | Admin Networks | SSH |
| TCP | 8800 | Admin Networks | Wickr Installer UI |
| TCP | 9870-9881 | Internal Network | Replicated Services |

You will need access to ports 22 and 8800 to complete the installation. Both ports should be restricted to administrative network ranges or individual IP addresses for security purposes.

## 3.2. Voice and Video Server

The Voice and Video server is disabled in the NIAP version.

## 3.3. Security Recommendations

In addition to restricting SSH access to the Wickr infrastructure, we recommend following your organizations security policies and best practices to secure and further lock down access to your Enterprise deployment. This can include, but isn't restricted to, firewall rules, host server access auditing, regular host server OS updates, and monitoring. Wickr can help make specific

recommendations during the deployment process, but it will be up to your teams to ensure your infrastrucure is adequately protected.

Beyond the server at the user level, if there is a requirement to restrict the types of devices your users can use with Wickr Enterprise we recommend using a Mobile Device Management (MDM) solution.

## 3.4. Replicated Overview

Wickr Enterprise has moved to using a third party service called Replicated to install and manage the software setup and deployment. There are a number of improvements this offers, most notably container monitoring and automatic service restarts. It also allows for full snapshots of your install to quickly redeploy elsewhere or for backup purposes.

Wickr will provide a license file to use during your install. This license file contains the following information:

- Services allowed in your deploy (Calling and/or Compliance)
- Software Updates
- Online and/or Offline installs
- Installable versions (Stable, Beta, Unstable)

There are two installation options:

- Online Install - Internet Access

  This install will reach out to the Replicated repositories and pull down the latest Wickr Enterprise containers available. It will automatically check for updates, but will not install them until an administrator does so manually.

- Offline Install - Airgapped

  The NIAP version has only been tested with Online Installs. Airgapped installs will be available in a future version.

## 3.5. Privacy Information

Wickr can see the following in an ***online*** install:

- When the Replicated license was first activated
- What release version was installed
- If the license is activated or inactive

The Replicated services will reach out and poll the following URLs periodically to check for updates to Wickr Enterprise:

- get.replicated.com
- api.replicated.com
- registry.replicated.com
- registry-data.replicated.com
- quay.io

Wickr can't see any information about ***offline*** installs, but the license file restrictions will still apply.

# 4. Getting Started - Install Enterprise

## NIAP Install script

```
#!/bin/bash
set -e
if [ $EUID -ne 0 ]; then
   printf "This script needs to be run as root or with sudo.\n" > /dev/stderr
   exit 1
fi
printf "\nWelcome to the Wickr Enterprise NIAP install script!\n\nThis script will ask for a password and a salt word and will then begin the installation
process.\nThis installation requires Ubuntu 20.04 and an empty block device with at least 100 GB of free space.\n\n"
RELEASE=$(lsb_release -r | awk '{print $2}')
if [ $RELEASE != 20.04 ]; then
   printf "This installer requires Ubuntu 20.04!\n"
   exit 1
fi
printf "Ubuntu 20.04 check passed!\n\n"
read -p "Please enter a password to be used to create the pbkdf2 password hash."$'\n' PASSWORD
if [ -z "$PASSWORD" ]; then
   printf "Please enter a password.\n"
   exit 1
fi
read -p "Please enter a word to be converted to hexadecimal to be used as the salt for the pbkdf2 password hash."$'\n' SALT
if [ -z "$SALT" ]; then
   printf "Please enter a word to be used as the password hash salt.\n"
   exit 1
fi
read -p "Please enter block device path to be used for encrypted storage. IE /dev/sdb1"$'\n' BLOCK_DEVICE
if [ -z "$BLOCK_DEVICE" ]; then
   printf "Please enter a block device path to be used as encrypted storage.\n"
   exit 1
fi
read -p "Please save your password and salt word in a safe place. Type YES in all caps to continue."$'\n' REPLY
if [[ ! $REPLY = "YES" ]]
then
   printf "\n\nQuitting!\n\n"
   exit 1
fi
printf "\n\nCreating encrypted volume for docker images...\n\n"
apt-get update && apt-get install -y nettle-bin cryptsetup
echo -n $PASSWORD | nettle-pbkdf2 --iterations=1000000 --length=16 --hex-salt $(echo -n $SALT | xxd -ps) | tr -d " " > /etc/crypttab.keyfile
CRYPTFS_ROOT=/cryptfs
mkdir -p $CRYPTFS_ROOT
cryptsetup -y -q luksFormat $BLOCK_DEVICE --key-file=/etc/crypttab.keyfile
cryptsetup luksOpen $BLOCK_DEVICE cryptfs --key-file=/etc/crypttab.keyfile
mkfs.ext4 /dev/mapper/cryptfs
mount /dev/mapper/cryptfs /cryptfs
echo "/dev/mapper/cryptfs /cryptfs ext4" >> /etc/fstab
UUID=$(blkid /dev/mapper/cryptfs | awk -F "" 'FNR==1 {print $2}')
echo "cryptfs "$BLOCK_DEVICE" /etc/crypttab.keyfile luks" >> /etc/crypttab
mkdir -p /cryptfs/docker-data
mkdir -p /etc/docker
echo '{"data-root": "/cryptfs/docker-data"}' > /etc/docker/daemon.json
printf "\n\nStarting Replicated installer...\n\n"
curl -sSL -o install.sh https://get.replicated.com/docker/wickrenterprise/niap
chmod +x install.sh
./install.sh
```

## 4.1. Online Install - Internet Access

Run the NIAP install script above on the **Messaging** server as a user with sudo access to install the components necessary to begin the installation.

The install script will check for root access and Ubuntu 20.04. The script will then ask for a Password, Salt word, and block device path. Save your Password and Salt word in a secure location. The script will then use the provided Password and Salt word to generate a PBKDF2 generated passphrase which is then used to secure and automatically unlock the encrypted drive where the Docker volumes are stored. Once the script finishes creating the encrypted drive it will start the Replicated Wickr Enterprise installer set to the NIAP branch.

Password

This password will be used as the password in the PBKDF2 generated passphrase used to secure the enrypted drive

Salt word

This salt word will be used as the salt in the PBKDF2 generated passphrase used to secure the encrypted drive

Block device path

This is the path to the block device that will be encrypted and used to store the docker containers Ex. /dev/sdb1

After the encrypted drive is created the Replicated installer is automatically started. The Replicated install script will ask a few questions about your network configuration, the script will install Docker and the Replicated web interface.

When the script is complete, the installer will direct you to a URL to continue the installation:

```
To continue the installation, visit the following URL in your browser:

https://$YOUR_IP:8800
```

Continue to Access the Web Installer.

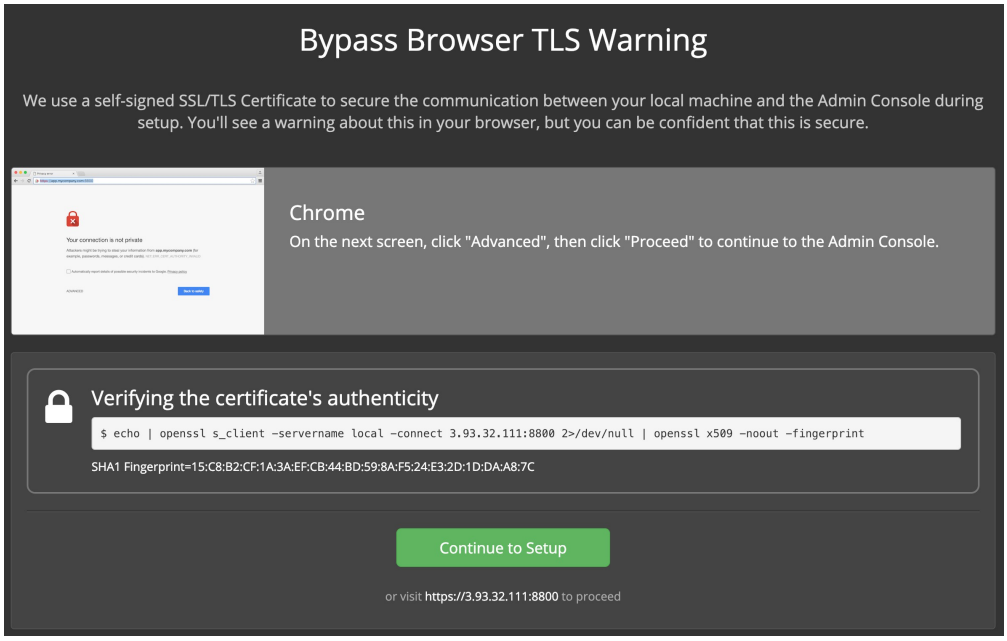
## 4.2. Offline Install - Airgapped

This will be available in a future version.

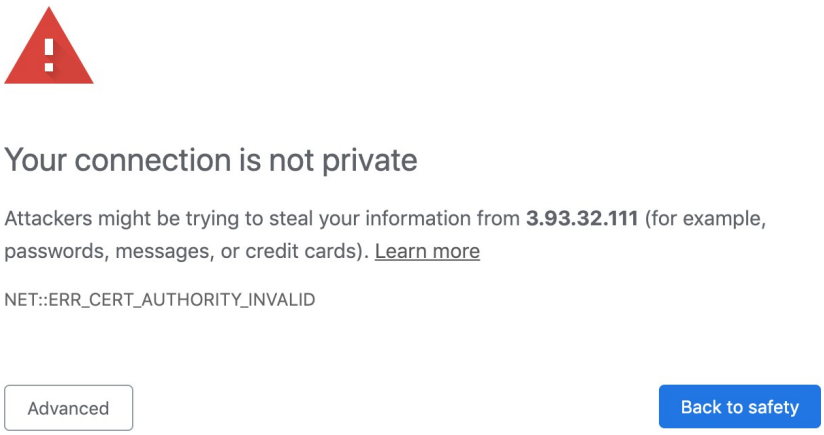
## 4.3. Access the Web Installer

When the server has started, visit the supplied URL to complete the Wickr Enterprise installation via

the Web UI. The first time you visit the page, you will see an SSL security warning, but you can accept the self-signed certificate and proceed for the initial setup.



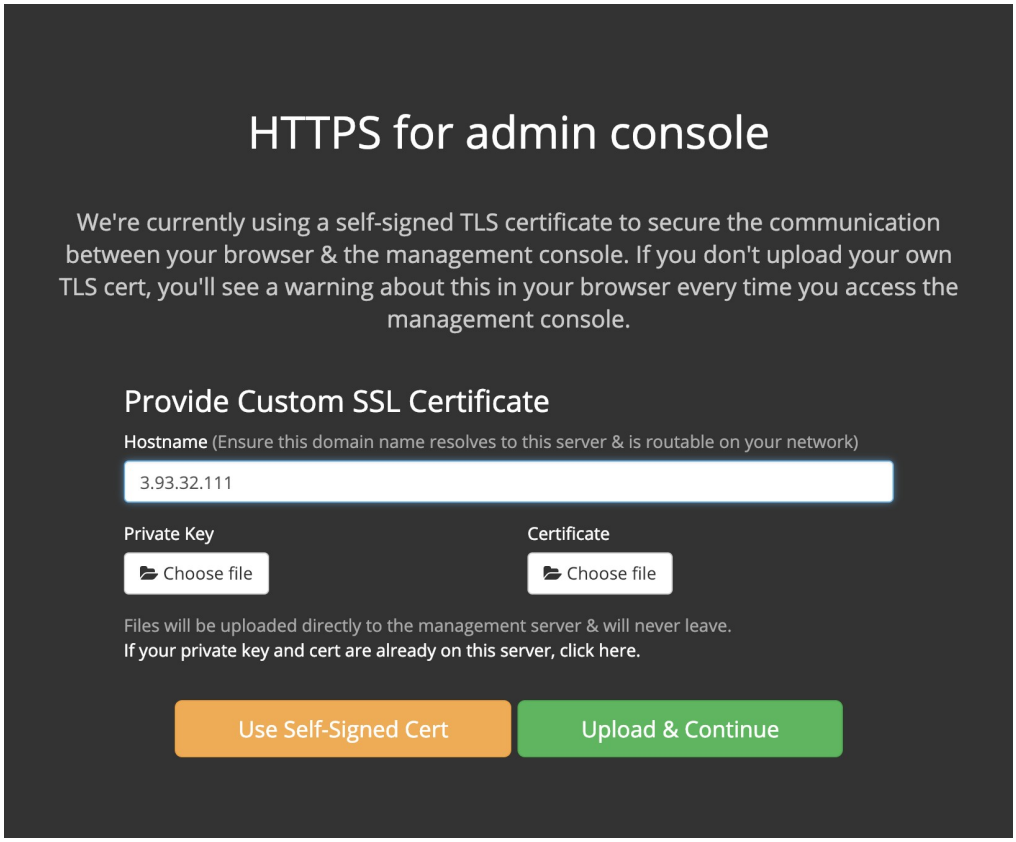

Clicking **Continue to Setup** will lead you to the following page (or similar, depending on your browser):




Click **Advanced** and then the **Proceed to...** link to continue to the installation page.

# 4.4. Configure the Web Installer

## 4.4.1. Hostname and TLS




On this page, you should set the hostname to either the IP address of your Messaging server (without the port) or to a DNS name which resolves to the same address.

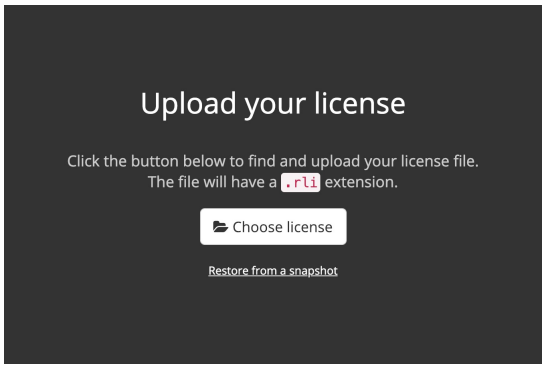
You also have the option of automatically generating a self-signed certificate or uploading a custom certificate and key. For most installations, using a self-signed certificate will suffice.

### 4.4.2. Custom SSL Certificates

If you opted to supply a custom SSL certificate during setup, this SSL certificate will also be used by the calling server for any connections.

It is not possible to have different certificates for the messaging service and the calling services at this time.

### 4.4.3. Upload License

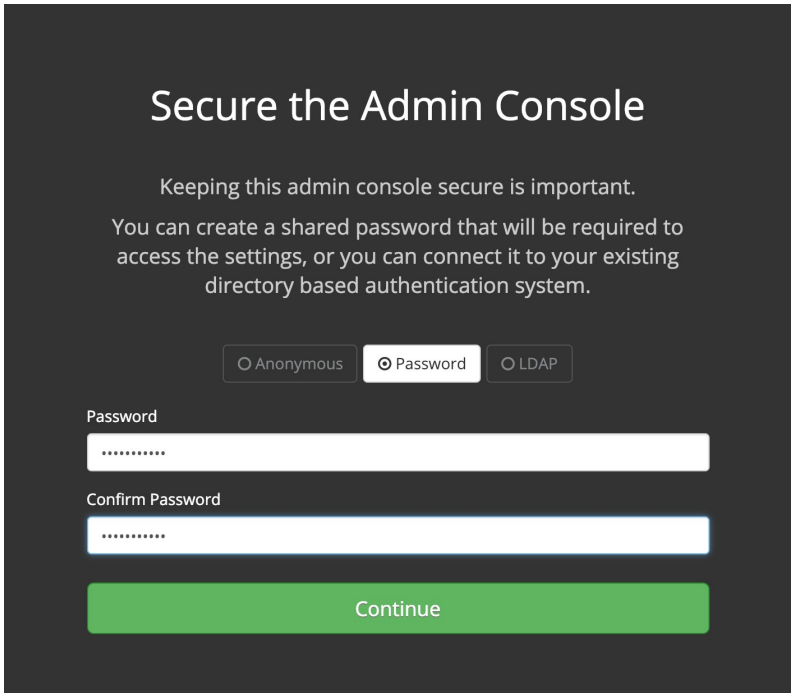



Upload your Wickr Enterprise license supplied by Wickr. If you don't have a license or have lost your license, contact your Wickr representative.

After uploading your license, you may have the option of choosing between an **Online** or **Airgapped**

install. Most installs are Online, but if you are deploying Wickr Enterprise into a network with limited or no internet connectivity, an Airgapped installation may be more appropriate.

### 4.4.4. Authentication



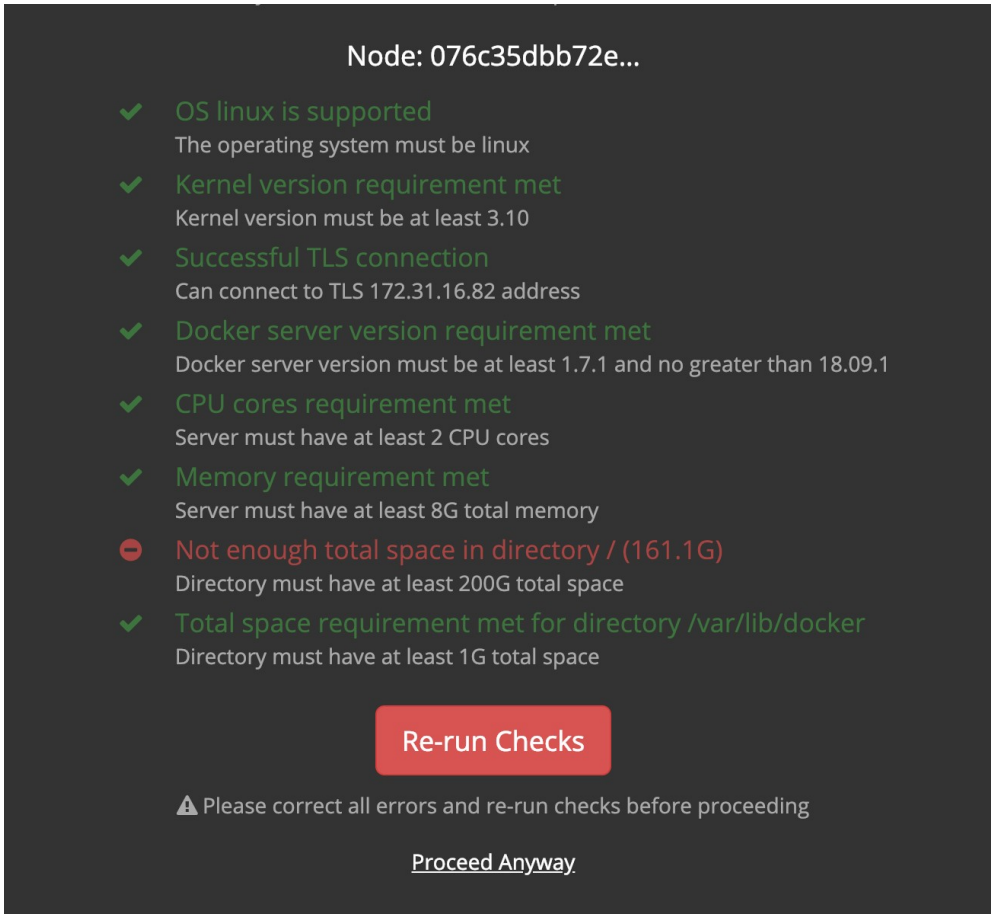

Set a secure password for accessing the Wickr Enterprise web installer. You can also configure authentication via LDAP or allow access without authentication (not recommended).

This password is for the installation interface only, not the Wickr Enterprise Admin Console.

### 4.4.5. Preflight Checks

The web installer will now run preflight checks to ensure that your system meets the requirements for running Wickr Enterprise.

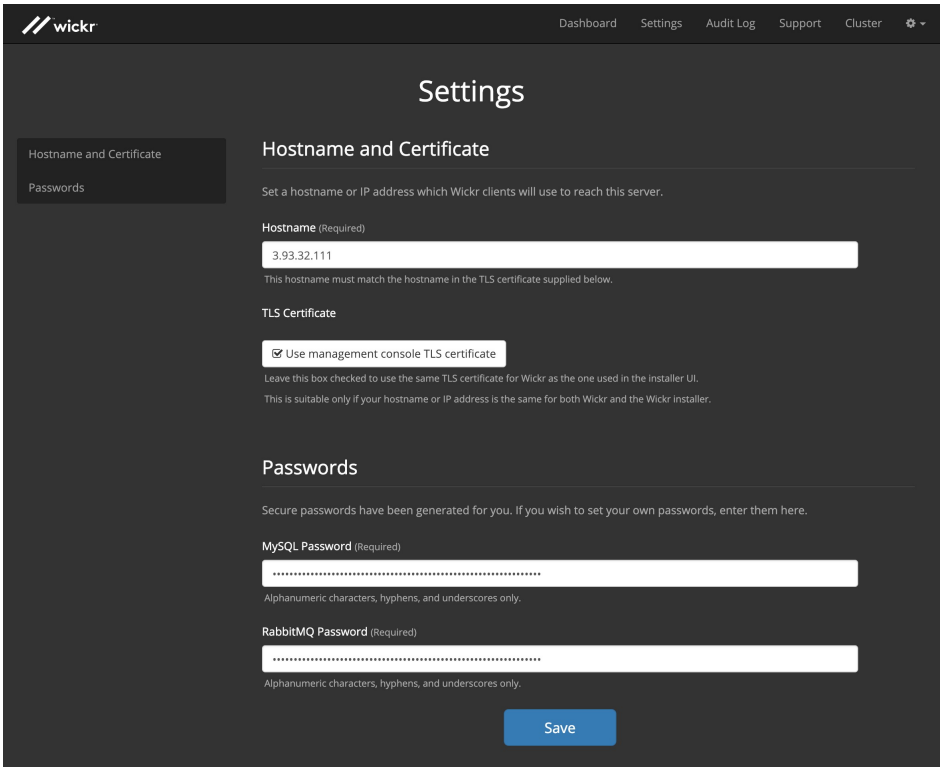



If you have failed to meet any requirements, you can either resolve the issue and click the **Re-run Checks** button or click the **Proceed Anyway** link to ignore the failing checks (not recommended).

## 4.5. Add Voice and Video Server (Optional)

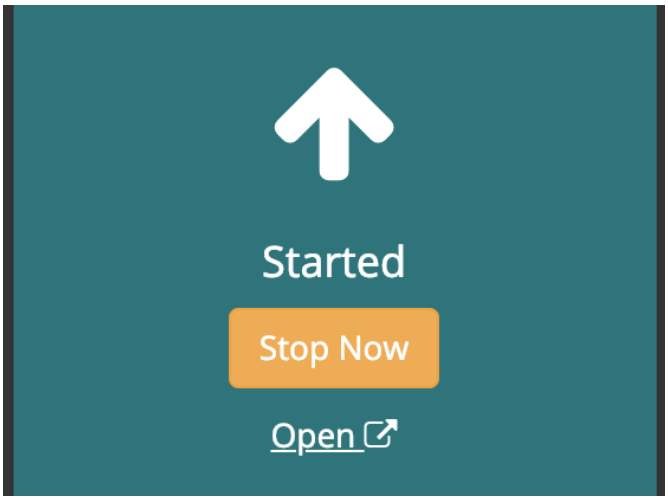
This feature will be available in a future release.

# 5. Configure Wickr Enterprise



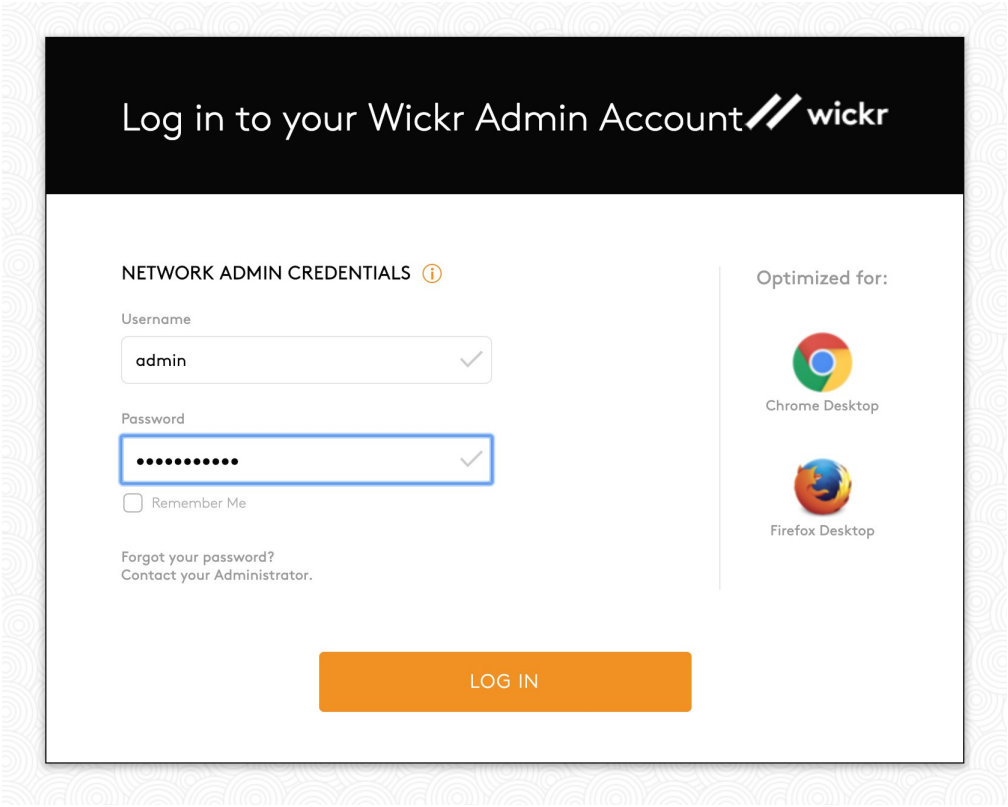
This page allows you to change settings for your Wickr Enterprise server. The default settings will suffice for most cases, but you can supply your own TLS certificates or service passwords if you prefer.

Wickr Enterprise services are starting. When the process has completed, the installer will show a state of "Started" (see the example below) and you can click the **Open** link to access the Wickr Enterprise Admin Console.




# 5.1. Log in to the Wickr Admin Console

After the installation has completed, you can click the **Open** link to access your Wickr Enterprise Admin Console. It's also available at https://$YOUR_IP/admin, where $YOUR_IP is the IP address or hostname of your Wickr server.




The default password for the admin user is Password123. You must update this password upon first

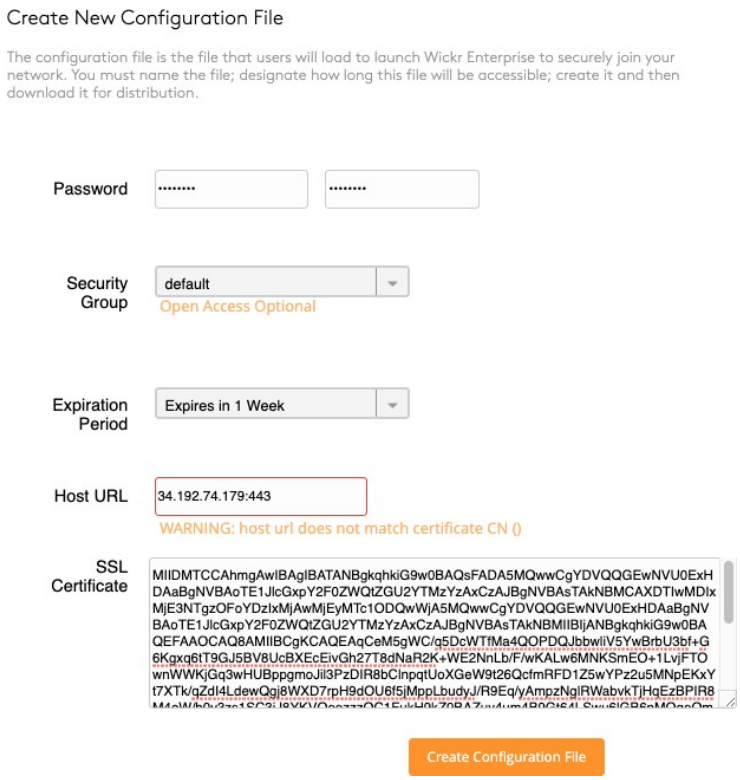
 

login. See the Wickr Enterprise Administrator's guide for more information on navigating the Admin Console.

# 6. Optional Enterprise Components

## 6.1. Messaging Proxies

In some cases it may be necessary to route traffic exclusively through a proxy, not allowing traffic to reach the Enterprise messaging endpoint directly. This is done by manually editing a config file to point only to the proxy and also by adding the SSL certificate the proxy uses.




Each certificate must be a single line, stripped of beginning and ending tags, and also stripped of any whitespace. An example command to make a certificate into one line is here:

+

```
cat cert.pem | sed s/.*BEGIN.*$// | sed s/.*END\ CERTIFICATE.*$// | xargs echo -n | sed s/\ //g
```

The certificate you add to the config file should be as close to your CA as possible. This ensures any certificate changes later will still allow a connection.

## 6.2. Compliance Service

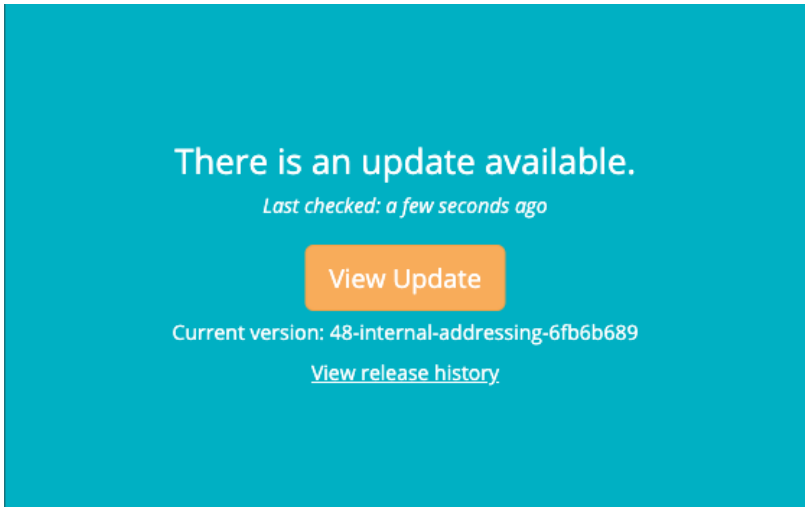
The Wickr Compliance Service allows for Administrators to save all messages and files sent from and within a network. It does not capture information coming into the Compliance network from a non-Compliance network, but it will capture information sent *from* a Compliance network.

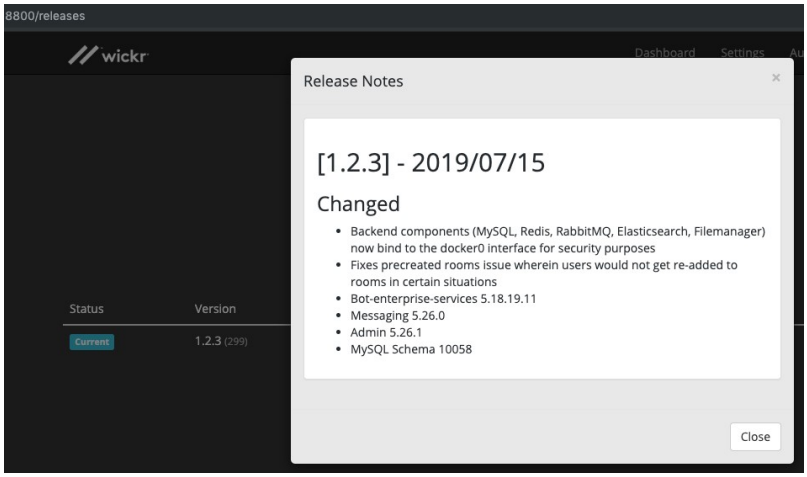
If using the Compliance Service please refer to the [Wickr Compliance Deploy] document for install instructions and requirements.

# 7. Software Updates

If using an online install you will see an **Update Available** message and a **View Update** button in the Replicated dashboard, shown here.




Clicking the **View Update** button will take you to the Releases page, which shows information about the release history and any updates available.



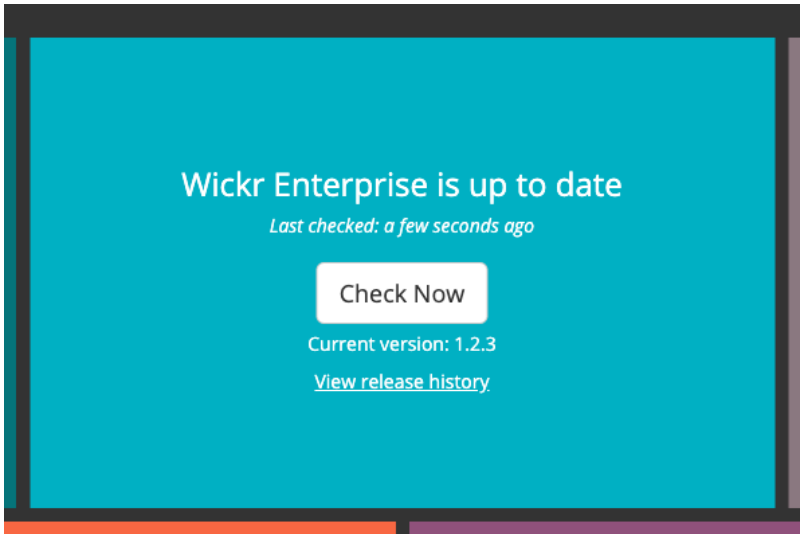

Applying the update will download and stage the new container images. Once they're ready and have been validated, the Replicated services will restart the necessary services with the new images in place. The downtime is minimal, but we recommend doing so during a designated maintenance window.

For offline installs the process is more involved, but still relatively simple. The first step is to download a new .airgap file using your unique URL and password.

Once the .airgap file is available, place it on the App server and replace the old .airgap file with the new one. Once it is in place you can click the **Check Now** button. Replicated will detect the new file, scan the contents, compare them to the current versions, and finally make the update available on the **Releases** page.

If the .airgap file location has changed you can update this information in the **Console Settings** panel in the Replicated Dashboard.




## 7.1. Troubleshooting Updates

If your update fails please run this command to check logs for errors related to Replicated.

```
grep -rni 'replicated' /var/log/* | grep 'ERRO'
```

### 7.1.1. Known issues

If the command above returns an error similar to the following you will need to downgrade replicated.

```
May 12 21:33:11 ip-172-22-33-10 docker[5432]: ERRO 2020-05-12T21:33:11+00:00
tasks/app_tasksteps.go:536 Failed auto-update Replicated: current Replicated version is not
compatible with app release 501.0, which requires "< 2.43.0"
```

To downgrade Replicated run the following commands on both the Messaging and the Calling servers.

```
mkdir replicated-2.44.2 cd replicated-2.44.2/
wget --trust-server-names --content-disposition 'https://s3.amazonaws.com/replicated- airgap-
work/stable/replicated-2.44.2%2B2.44.2%2B2.44.2.tar.gz'
tar -zxvf replicated-2.44.2+2.44.2+2.44.2.tar.gz
```

To complete the downgrade run this command on the Messaging server.

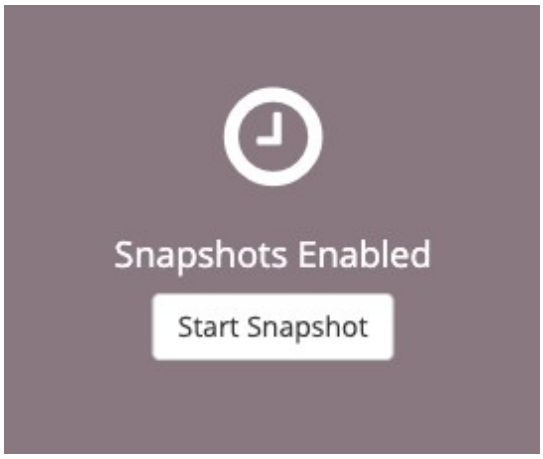
```
cat ./install.sh | sudo bash -s airgap force-replicated-downgrade
```

# 8. Maintenance

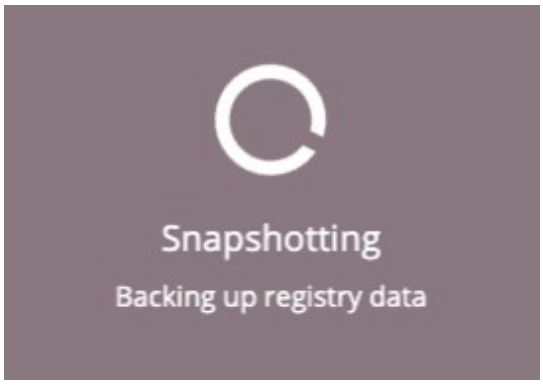
Replicated allows for full snapshots to backup your Wickr Enterprise install. You can take these at any time and restore to a new deploy or to overwrite to an older version of an existing install. The

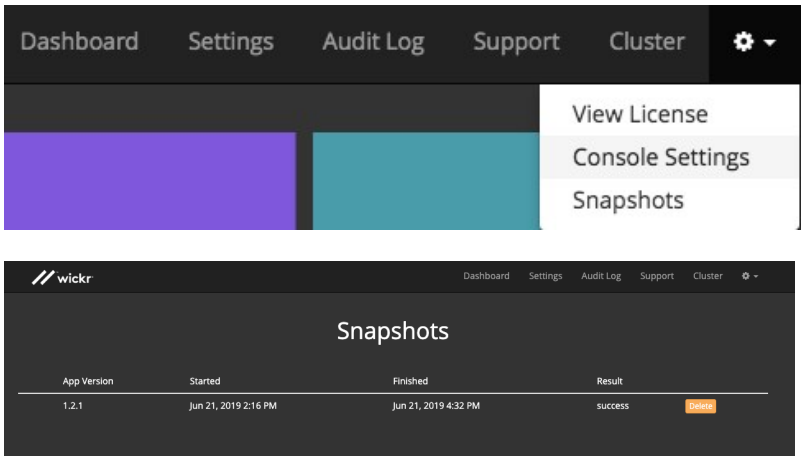
**Start Snapshot** button in the Replicated Dashboard will start this process.




The snapshot process can take more than 15 minutes to complete. It will take longer the more data the installation has.




Snapshots can be managed from the **Snapshot** menu option pictured below.



# 8.1. Restoring a backup

When a Wickr deployment is restored from a backup, the database may need to be modified after the restore in order for messages to be delivered successfully to your Wickr clients. The clients use a sequence number (known as the MSN) to determine whether or not they should download new messages, and after restoring from a backup it's possible that the client and server MSN have become out of sync. As a result, clients will not receive messages until the server MSN is higher than the client MSN.

For example, if the current MSN is 1000, clients will download messages 1001 and beyond. If a backup is restored with an MSN of 900, clients will not download messages until they see a count of at least 1001.

To get the current MSN from the database, you can use this query:

```
SELECT `AUTO_INCREMENT`
FROM INFORMATION_SCHEMA.TABLES
WHERE  TABLE_SCHEMA  =  'wickrdb'
AND  TABLE_NAME  =  'message';
```

It will return a number value:

```
++
| AUTO_INCREMENT |
++
|1000 |
++
```

If it is not possible to retrieve the current MSN count prior to restoring the backup, increase by thousands, hundred thousands, or millions until messages begin to be received by clients successfully. This increase value will depend on how many messages have been sent since this backup was taken.

If the database is accessible prior to the restore you can pull the current MSN value and change the value to that once the restore is complete.

Using the example 1000 above, the MSN value could be changed to 50000 like so to ensure it is high enough:

*Overkill Increment Example*

```
ALTER TABLE message AUTO_INCREMENT = 50000;
```

In addition to increasing the MSN value, the encryption keys used will be different and need to be purged. Truncating a table in the database will force the clients to generate and upload new keys to use going forward.

```
TRUNCATE  TABLE  ecckey;
```

As always, if you have any questions or encounter an issue not covered here, please reach out to Wickr Support.

## 8.2. Removing Replicated

If you need to remove the Wickr Enterprise installer and Replicated, you can use the following commands. Instructions for Ubuntu and CentOS are available here: https://help.replicated.com/docs/native/customer-installations/installing-via-script/#removing-replicated

*Example Commands to remove Replicated in CentOS*

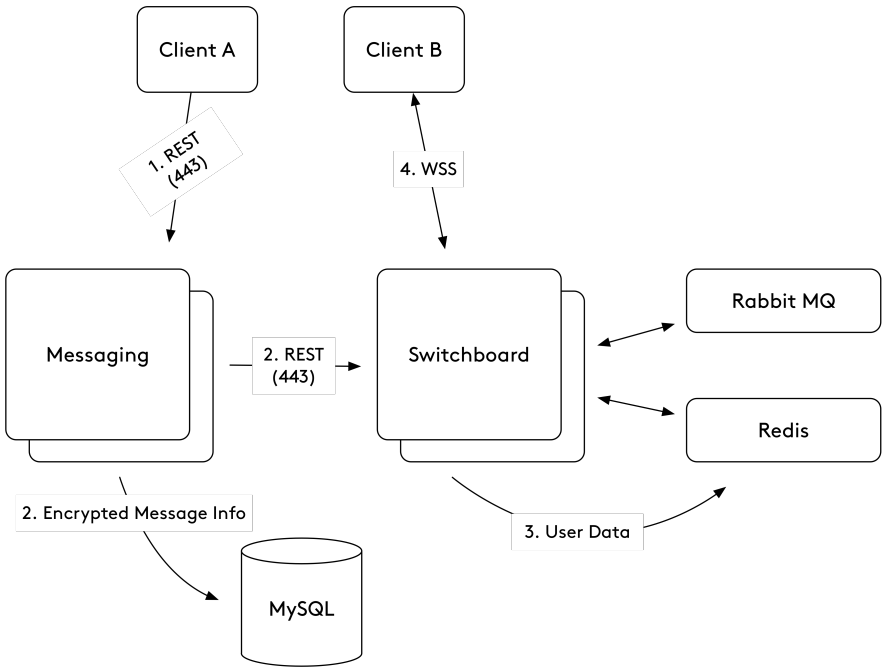
```
systemctl stop replicated replicated-ui replicated-operator service replicated stop
service replicated-ui stop service replicated-operator stop docker stop replicated-premkit docker
stop replicated-statsd
docker rm -f replicated replicated-ui replicated-operator replicated-premkit replicated-statsd retraced-
api retraced-processor retraced-cron retraced-nsqd retraced-postgres
docker images | grep "quay\.io/replicated" | awk '{print $3}' | xargs sudo docker rmi
-f
docker images | grep "registry\.replicated\.com/library/retraced" | awk '{print $3}' | xargs sudo docker
rmi -f
yum remove -y replicated replicated-ui replicated-operator
rm -rf /var/lib/replicated* /etc/replicated* /etc/init/replicated*
/etc/default/replicated* /etc/systemd/system/replicated* /etc/sysconfig/replicated*
/etc/systemd/system/multi-user.target.wants/replicated*  /run/replicated*
```

# 9. Troubleshooting

## 9.1. Clients are unable send messages

The images below describe how messages are sent and received between end clients.




1. When a message is sent from a client, it first hits the Messaging server using a REST call on port 443.

2. Messaging then sends the encrypted message content to MySQL, informs Switchboard a new message is available, and Redis caches the client information for faster lookups.

3. Switchboard pulls the web socket info from RabbitMQ to prepare to send the message.

4. Switchboard then sends the message to the client using Web Socket Secure, and the client sends an acknowledgement that it has been received.

If the end clients can't connect to Wickr services, the first thing to check is if they're able to route to the Wickr address. You can verify this in a browser by appending /checkConfig.php?api=117 to the **Application** server IP address over HTTPS.

*Example URLs, be sure to replace with your IP or DNS address.*

```
https://10.0.0.10/checkConfig.php?api=117 https://wickr.example.com/checkConfig.php?api=117
```

# Appendix A: Container Descriptions

## A.1. Base Services

- Admin

  The admin container runs the Administrative Panel used to manage Administrators, Users, and Security Group Settings. It connects to the Switchboard container, the Pre-Created Rooms bot, RabbitMQ, and the database.

- Messenger

  The messaging container handles incoming messages, client registration, login, and call routing. It connects to Switchboard, RabbitMQ, the Voice & Video containers, and the Database.

- Cron

  The crond container handles regular maintenance of the other services internally. It connects to the database and RabbitMQ.

- Switchboard

  The switchboard container handles message transfer to Wickr clients. It communicates with the clients via a Websocket, and will establish connections to the database, Redis, RabbitMQ, and mobile notification services (GCM and APN).

- Nginx Proxy

  The proxy container receives all incoming HTTPS traffic and routes it to the appropriate backend container.

- Elasticsearch

  The elasticsearch container manages the search and listing of the user directory service.

- Directory

  The directory container manages the user directory that clients will use to find other users in their network. It connects to the database, RabbitMQ, and Elasticsearch.

- OIDC

  The oidc container manages SSO setup and connections. It connects to the database, Switchboard, RabbitMQ, and any identity providers configured by the Wickr Admin.

- Push Device

  The push_device container manages adding devices using a QR code and is only used in conjunction with SSO.

- Bot Enterprise Services

  The bot-enterprise-services container runs the Pre-Created Rooms bot. It connects to RabbitMQ and the Admin container.

- Enterprise Init

  The enterprise-init container bootstraps your deployment by populating required fields in the database. It will exit with return code 0 after it has successfully completed the initialization process.

- MySQL

  The mysql container is the database which houses all stateful data for the deployment. This includes users, messages, administrators, security group settings, and client information for linked devices on user accounts.

- RabbitMQ

  The rabbitmq container serves as a message bus for communication between the various Wickr services.

- Redis

  The redis container caches client connection data. It receives connections from Switchboard, Messaging, and Push Device.

- File Manager

  The filemanager container houses attachments and files in an encrypted datastore.