

Wickr Enterprise

Installation and Maintenance

Version 1.2.0, May 27, 2021

Table of Contents

1. Changelog	1
2. Deploy Overview	2
3. Requirements	2
3.1. Messaging Server	2
3.2. Voice and Video Server (Optional)	3
3.3. Security Recommendations	4
3.4. Replicated Overview	4
3.5. Privacy Information	5
4. Getting Started - Install Enterprise	5
4.1. Online Install - Internet Access	6
4.2. Offline Install - Airgapped	6
4.3. Access the Web Installer	7
4.4. Configure the Web Installer	8
4.5. Add Voice and Video Server (Optional)	11
5. Configure Wickr Enterprise	13
5.1. Log in to the Wickr Admin Console	14
6. Optional Enterprise Components	15
6.1. Messaging Proxies	15
6.2. Calling Proxies	15
6.3. Compliance Service	17
6.4. Advanced Configuration	17
7. Software Updates	17
7.1. Troubleshooting Updates	19
8. Maintenance	20
8.1. Restoring a backup	21
8.2. Removing Replicated	22
9. Troubleshooting	22
9.1. Clients are unable send messages	22
9.2. Clients are unable to make calls	23
Appendix A: Container Descriptions	25
A.1. Base Services	25
A.2. Voice and Video Services	26

This document describes how to install, deploy, and manage Wickr Enterprise 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

1.2.0

- 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

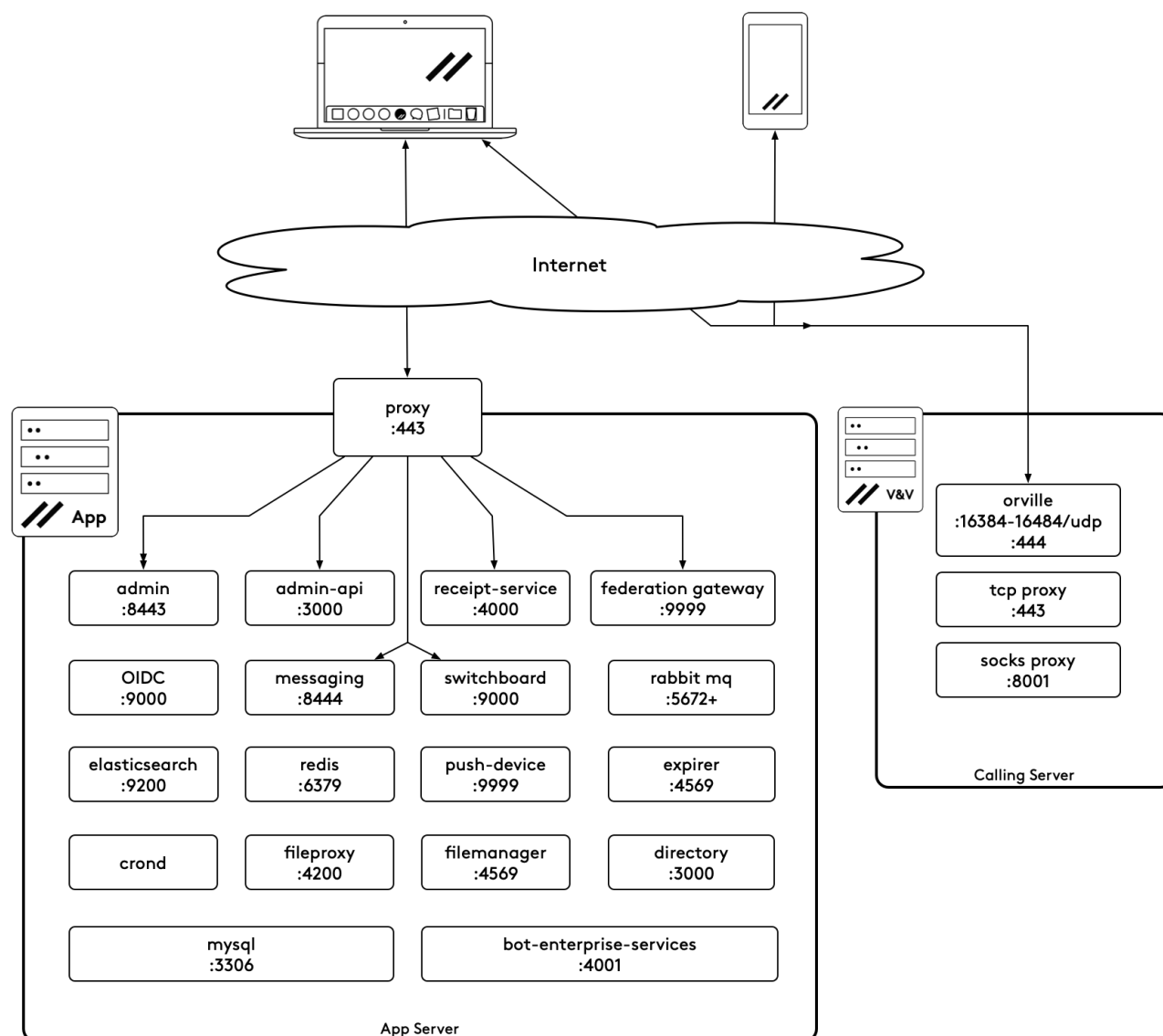
- Calling containers have changed. New diagrams added.
- Calling port range can be adjusted to allow for more calls
- Calling log locations have changed
- Upgrades can fail when using specific replicated versions. Fixes added to the [Troubleshooting Updates](#) section.

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 18.04+ or RHEL/CentOS 7
CPU	2+ Cores/vCPUs
RAM	8GB+
Disk Space	120GB+



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.

3.1.1. Software Requirements

Resource	Recommendation
Docker Version	1.7.1 to 18.09.2-ce
AppArmor or SELinux	Disabled or in permissive mode



If using an XFS filesystem with the `overlay` or `overlay2` Docker storage driver, ensure that the partition was created with `ftype=1`. There are known issues with using overlay filesystems on XFS with `ftype=0`.

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 (Optional)

The Voice and Video server needs the following. It allows for 50 concurrent calls and 50 people per call. Each call uses two UDP ports.



The new calling service supports a larger port range than the defaults shown below. Details on configuring this are listed in [Advanced Configuration](#).

Resource	Recommendation
OS	Ubuntu 18.04+ or RHEL/CentOS 7
CPU	2+ Cores/vCPUs
RAM	8GB+
Disk Space	60GB+

3.2.1. Software Requirements

Resource	Recommendation
Docker Version	1.7.1 to 18.09.2-ce
AppArmor or SELinux	Disabled or in permissive mode

3.2.2. Networking Requirements

Protocol	Port(s)	Network Range	Purpose
TCP	443	ALL	TCP Proxy
TCP	8001	ALL	SOCKS Proxy
UDP	16384-16484*	ALL	Audio/Video Streams
TCP	444	Messaging Server IP	HTTP API
TCP	22	Admin Networks	SSH



Port 22 should be restricted to administrative network ranges or individual IP addresses for security purposes. We do not recommend allowing port 22 globally.

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 infrastructure 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](#)

This install requires downloading the Wickr Enterprise files ahead of time. It can be automated using instructions here: [Automating Replicated Installs](#). No information is shared with Replicated or Wickr when installing offline.

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

There are two installation methods available, however regardless of which is used it will still be necessary to **disable firewalld** or add the following rules to **firewalld** configuration.

```
firewall-cmd --add-port=8800/tcp
firewall-cmd --add-port=9870-9881/tcp
firewall-cmd --permanent --zone=trusted --add-interface=docker0
firewall-cmd --reload
```

4.1. Online Install - Internet Access

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

```
curl -sSL -o install.sh https://get.replicated.com/docker/wickrenterprise/stable
sudo bash ./install.sh
```

Continue with firewalld active? (Y/n)

After answering 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

Replicated allows for offline installations via an 'airgapped' package.



Please make sure to download the airgap file before uploading to the server. This will ensure you have the latest version available. The airgap file name will have a three digit number that will correspond to a Wickr Enterprise release. Details can be found on the **Releases** page after installation.

Status	Version	Date Released	Date Installed
Current	1.2.2 (293)	Jun 21, 2019 1:53 PM	Jun 25, 2019 12:06 PM
	1.2.1 (286)	Unknown	Jun 20, 2019 3:14 PM

Install Docker on your airgapped server via your package manager or normal package installation process. The required version range for Docker is in the [Software Requirements](#) section of this document.

Download the latest replicated release from the following link and upload it to your airgapped server.


```
https://s3.amazonaws.com/replicated-airgap-work/stable/replicated-
2.44.2%2B2.44.2%2B2.44.2.tar.gz
```

Download your **.airgap** package from the link provided by Wickr and upload it to your airgapped server.

```
URL: https://get.replicated.com/airgap/#/wickrenterprise/*****
Password: *****
```

Make a note of the path on your server to this **.airgap** file. You'll reference this path in the Web Installer when prompted.



The verification process after selecting your **.airgap** file in the Web Installer can take quite a long time, this is normal.

Unpack and install replicated on the **Messaging** server from replicated.tar.gz.

```
tar xzvf replicated-2.44.2+2.44.2+2.44.2.tar.gz
cat ./install.sh | sudo bash -s airgap
```

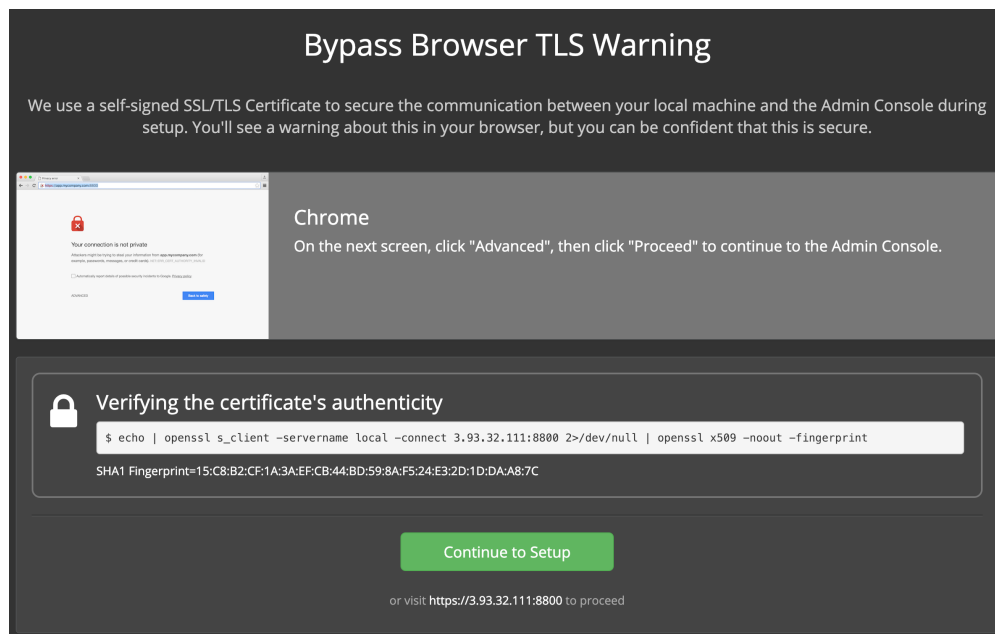
After answering questions about your network configuration the script will install the Replicated Web Interface. When this step is complete, the installer will direct you to a URL to continue the installation:

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

```
https://$YOUR_IP:8800
```

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):



Your connection is not private

Attackers might be trying to steal your information from **3.93.32.111** (for example, passwords, messages, or credit cards). [Learn more](#)

NET::ERR_CERT_AUTHORITY_INVALID

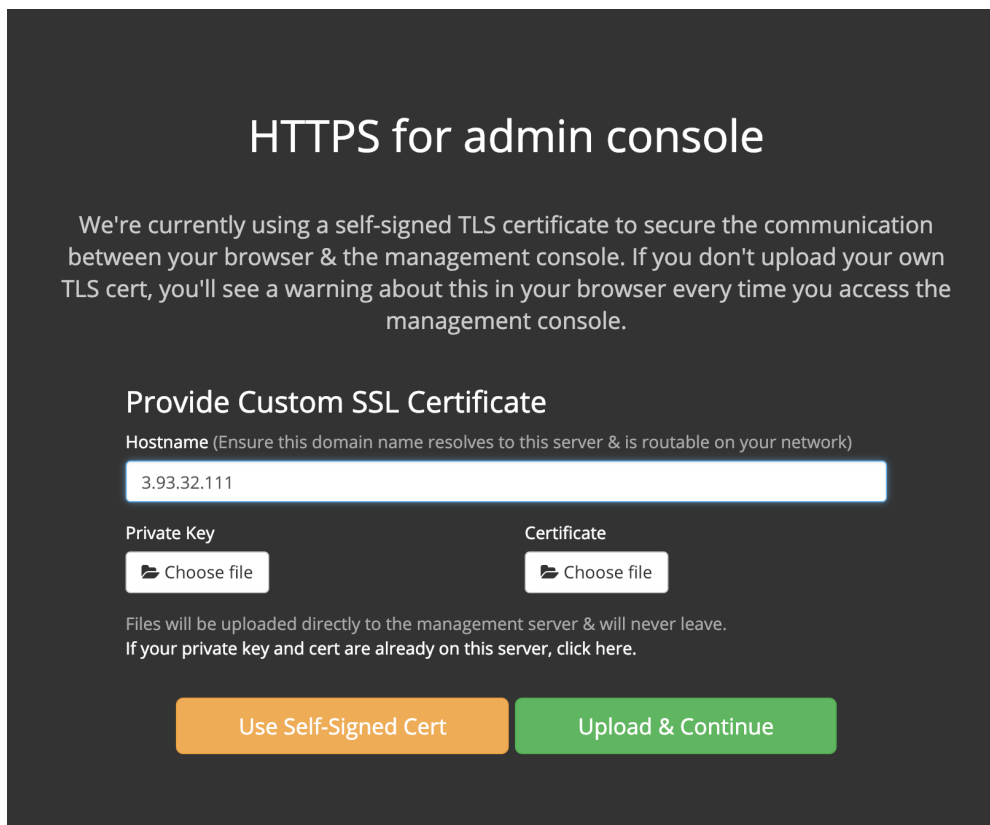
Advanced

Back to safety

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.

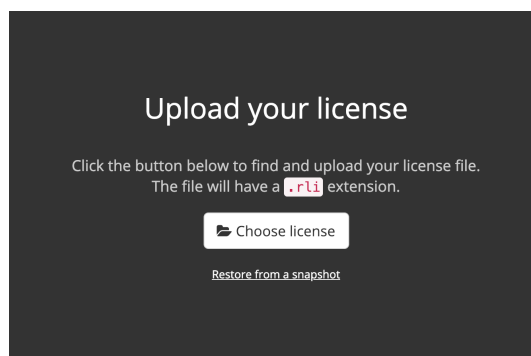
You also have the option of automatically generating a self-signed certificate or uploading a custom certificate and key. For most installations, using a self-signed certificate will suffice.

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

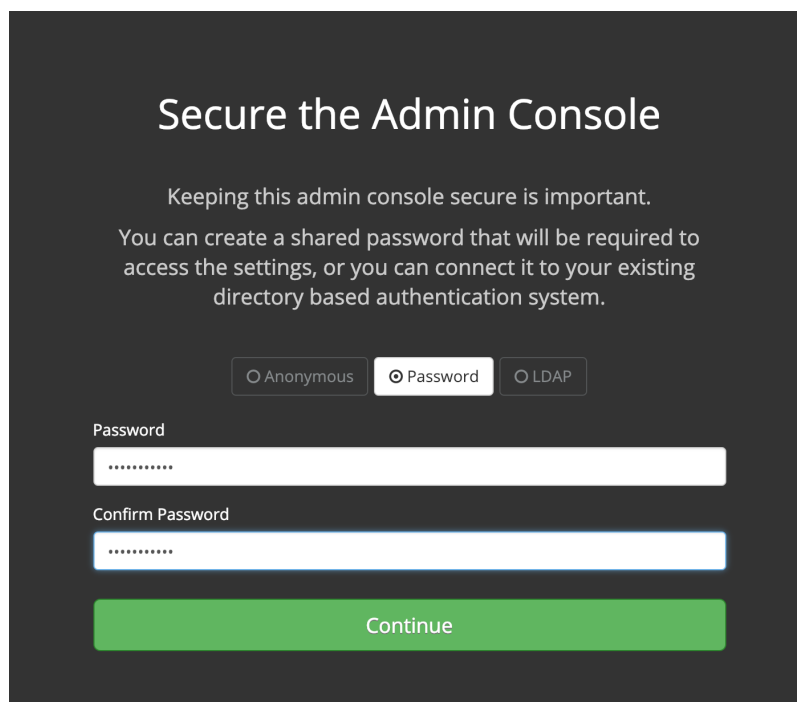


Upload your Wickr Enterprise license supplied by Wickr. If you don't have a license or have lost your license, contact your Wickr representative.

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



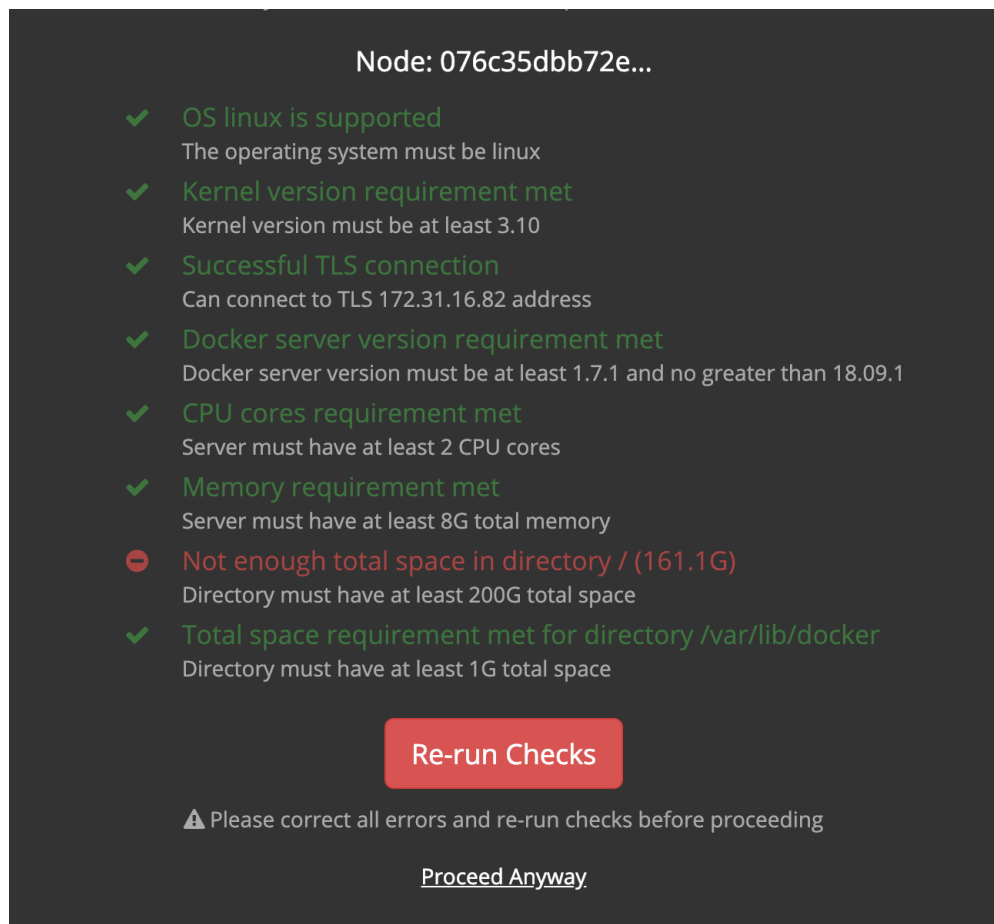
Set a secure password for accessing the Wickr Enterprise web installer. You can also configure authentication via LDAP or allow access without authentication (not recommended).



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.

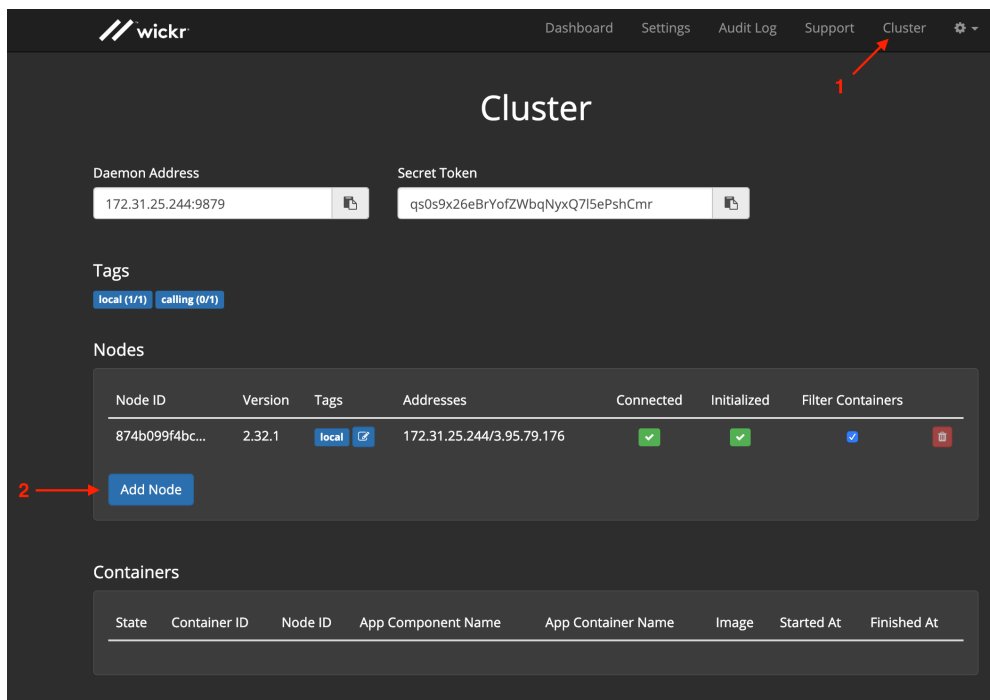


If you have failed to meet any requirements, you can either resolve the issue and click the **Re-run Checks** button or click the **Proceed Anyway** link to ignore the failing checks (not recommended).

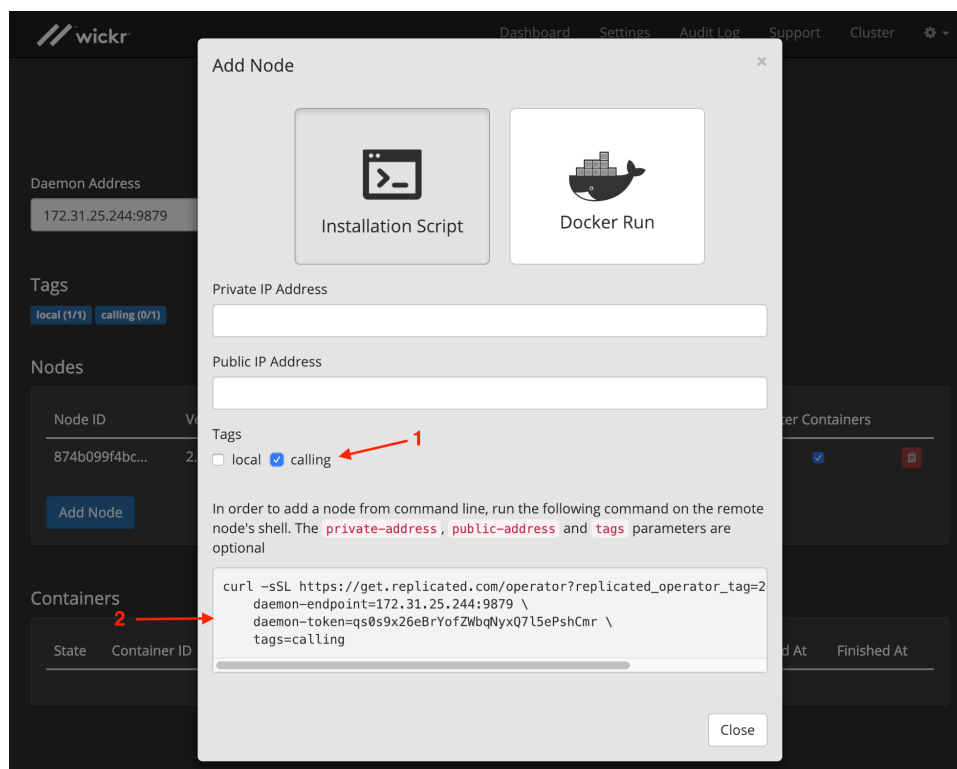
4.5. Add Voice and Video Server (Optional)

If you used the accompanying Wickr Terraform scripts to deploy your infrastructure, you will not need to add the node as described below.

After the preflight checks, the installer will take you to the Settings page, but you must first add your Voice and Video server to the cluster so that the installer can schedule services to it. Click on the **Cluster** menu item (Arrow #1 below) to get started.



1. Click **Add Node** (Arrow #2 above)



2. On the Add Node screen, check the box for the **calling** tag (Arrow #1 above), and optionally supply the public and private IP address of the Voice and Video server.
3. Download the latest replicated release from the following link and upload it to your airgapped server.

```
https://s3.amazonaws.com/replicated-airgap-work/stable/replicated-2.44.2%2B2.44.2%2B2.44.2.tar.gz
```

4. Decompress the Replicated archive.

```
tar xzvf replicated-2.44.2+2.44.2+2.44.2.tar.gz
```

5. Copy the generated curl command (Arrow #2 above) and run it on your Voice and Video server as a user with sudo access.

After answering questions about your network configuration, the script will install the services required to join the cluster. When complete, you should see two nodes and their IDs listed on the Cluster page. Click on the **Settings** menu and proceed to the [Configure Wickr Enterprise](#) section to continue the installation process.

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.



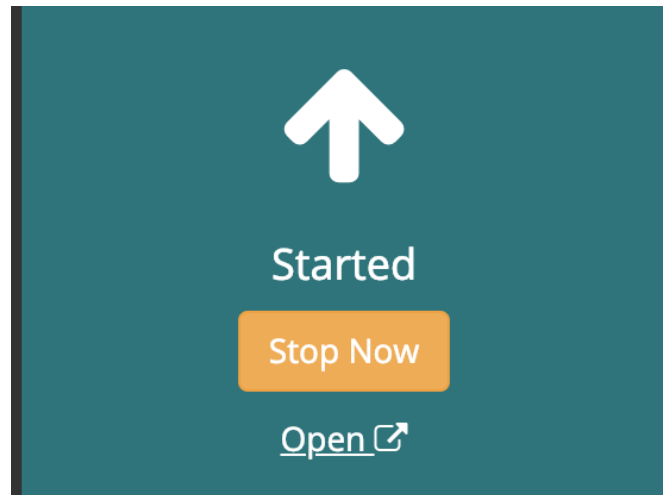
The option to enable or disable calling on the cluster is available on this page, along with the **Calling Hostname** value. Enter the IP or DNS information for the endpoint the clients will use to communicate with the calling server here.

Click the **Save** button and then **Restart Now** to begin downloading and starting the Wickr

Enterprise services. 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.

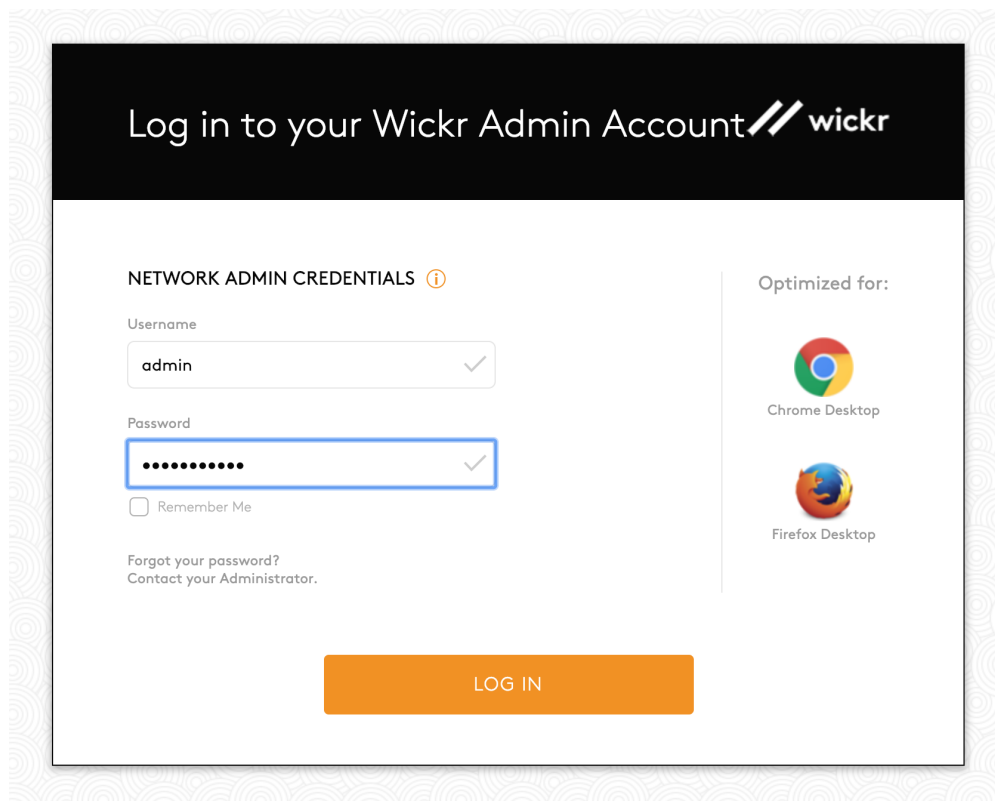


If the installation fails with the error **Error placing components: not enough operators available for component "calling"**, you need to add a server to the cluster with the **calling** tag. See the [\[Add Voice and Video Server\]](#) section for more information.



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](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

the network endpoints of your installation.

There are two basic methods to setup a call proxy:

1. Traffic Forwarding
2. TLS to TLS (SSL Termination)

We recommend forwarding because it's faster, reduces latency on calls, and supports both UDP and TCP calls. If terminating SSL, the certificate needs to be added to the Push Config Certificate list in the Admin Panel. It will also introduce more latency which will degrade calling performance.



A certificate from the app server **must** be added to push config for any Security Group using call proxies. This is a limitation in the current service and will be addressed in a later release.

The default CA can be found here on the **App** server: `/var/lib/replicated/secrets/ca.crt`.

If other certificates and CA are supplied during installation administrators will need to add that CA to any push config security groups that use call proxies.

The examples below are all valid calling proxies:

1. NAT Rule pointing to the calling server
2. Multiple network interfaces on the calling server
3. Any TCP/UDP load balancer



Remember to also route UDP traffic, unless forcing Open Access or your users force TCP calling on their devices. See [Voice and Video Server \(Optional\)](#) for the full port list.

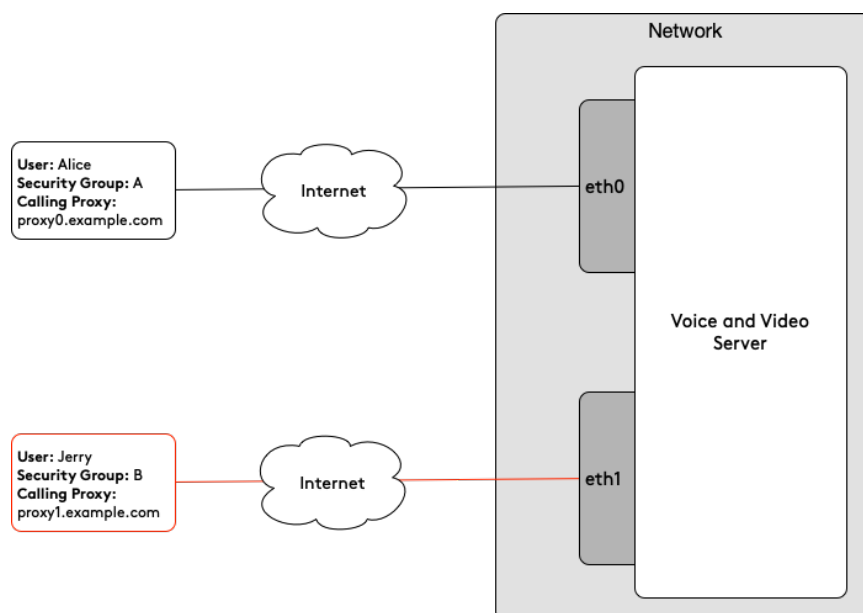


Figure 1. Multiple Network Interfaces

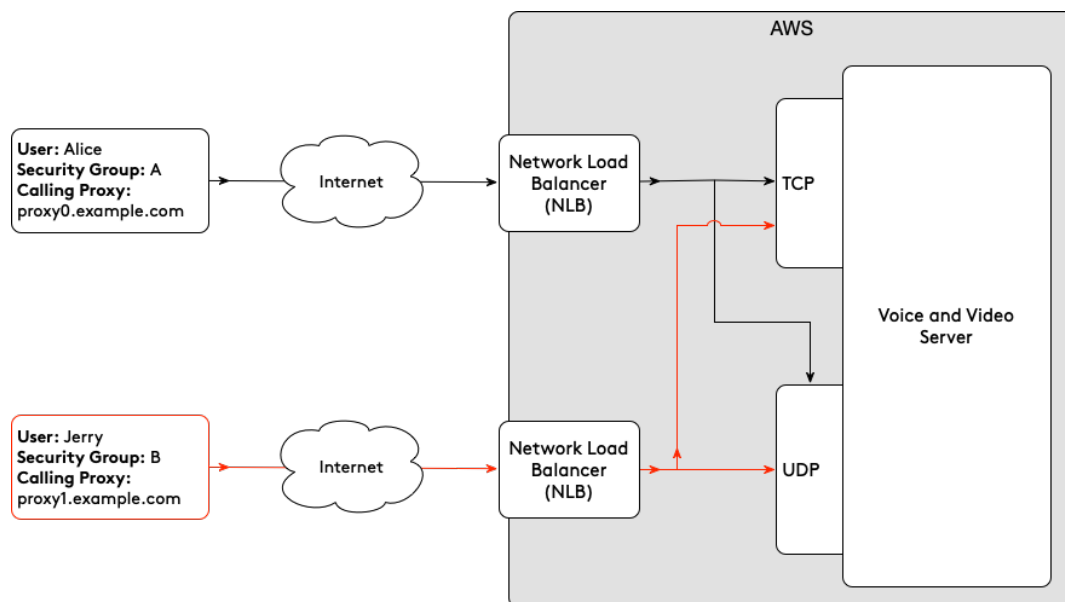


Figure 2. NLB Proxy

6.3. 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.

6.4. Advanced Configuration

As of the July 2020 release, number 611, calling can be modified to allow more than the default 50 concurrent calls. This is done by adjusting the port range inside the container using environment variables:

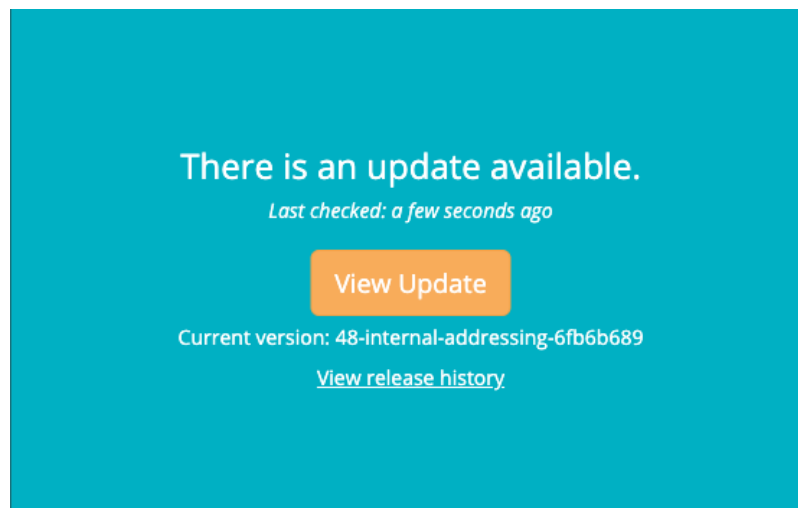
```
ORVILLE_MAX_CALLS=500
```

Each call requires two UDP ports. The default is set to 50 calls, which uses 100 UDP ports (16384-16484). The port range needed to allow for 500 as shown above would need to be 16384-17384 if using the default starting port.

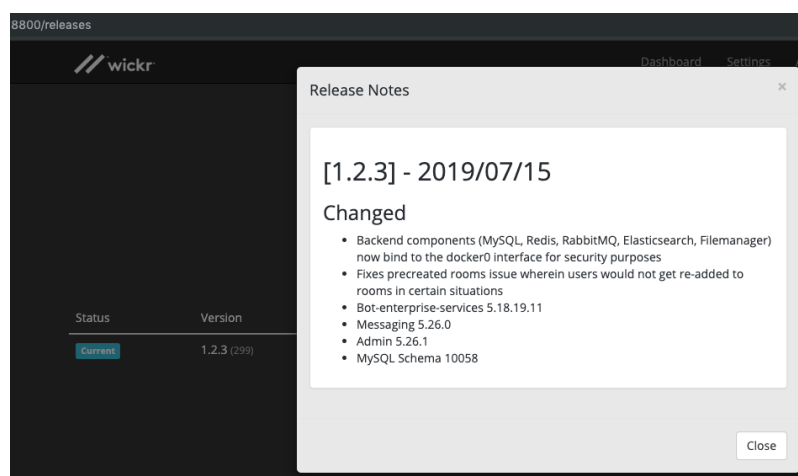
It is also possible to pass this value as a flag to the `orville` service running inside the container with `--max-calls <int>`, but typically environment variables make this easier to manage. The implementation will depend on how Wickr Enterprise was deployed.

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.



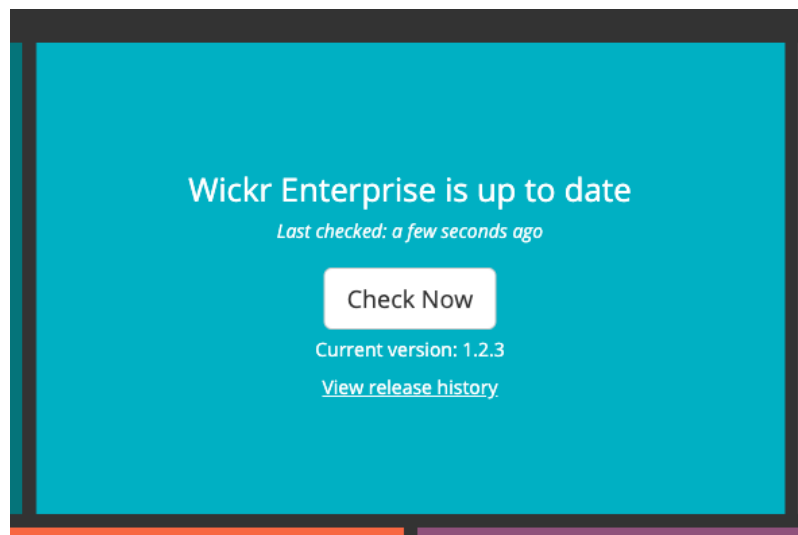
Applying the update will download and stage the new container images. Once they're ready and have been validated, the Replicated services will restart the necessary services with the new images in place. The downtime is minimal, but we recommend doing so during a designated maintenance window.

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.

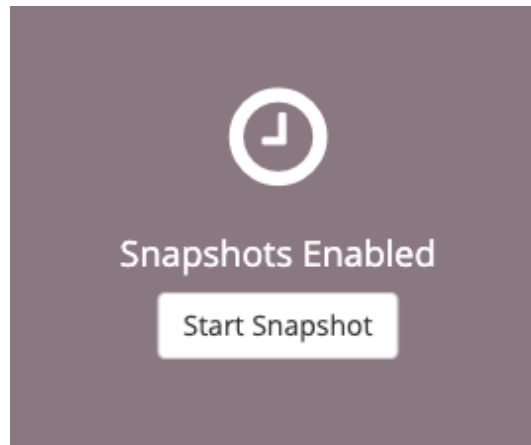
```
cat ./install.sh | sudo bash -s airgap force-replicated-downgrade
```

And run this command on the Calling server.

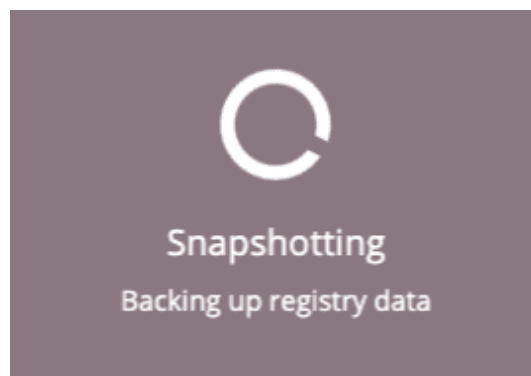
```
cat ./operator_install.sh | sudo bash -s airgap
```

8. Maintenance

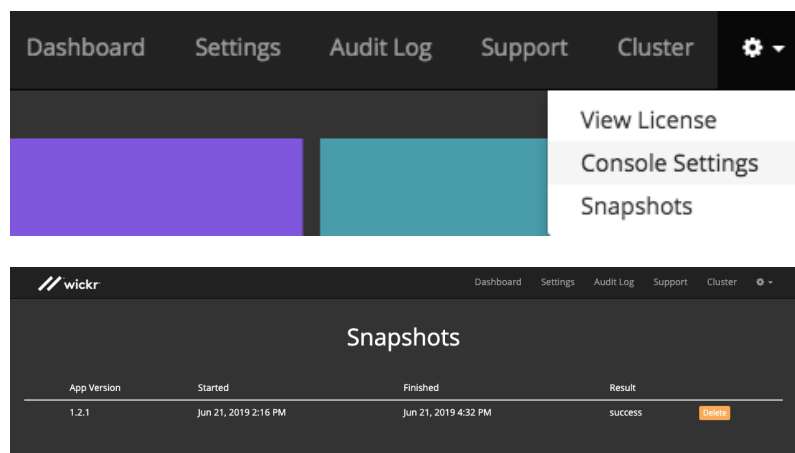
Replicated allows for full snapshots to backup your Wickr Enterprise install. You can take these at any time and restore to a new deploy or to overwrite to an older version of an existing install. The **Start Snapshot** button in the Replicated Dashboard will start this process.



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>

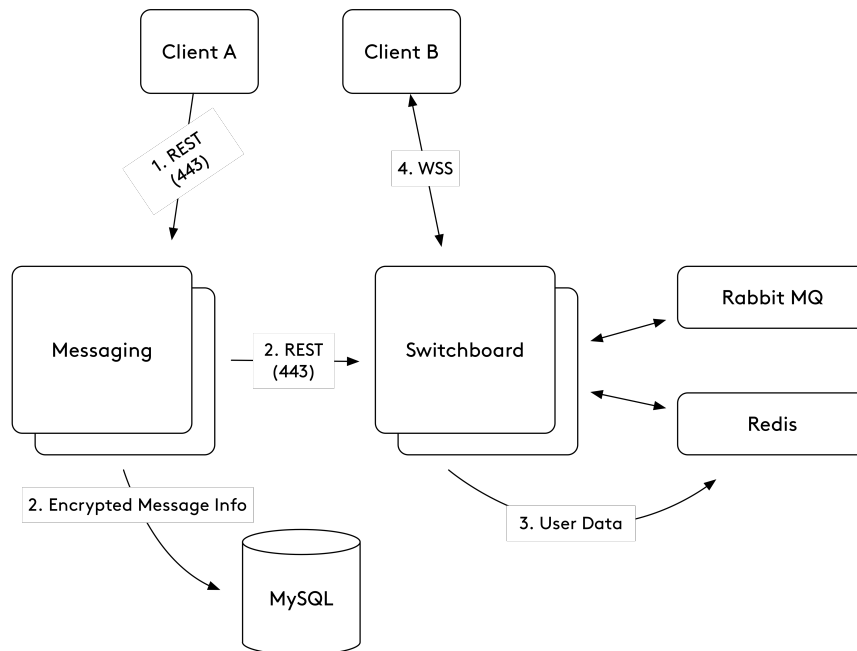
Example Commands to remove Replicated in CentOS

```
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.

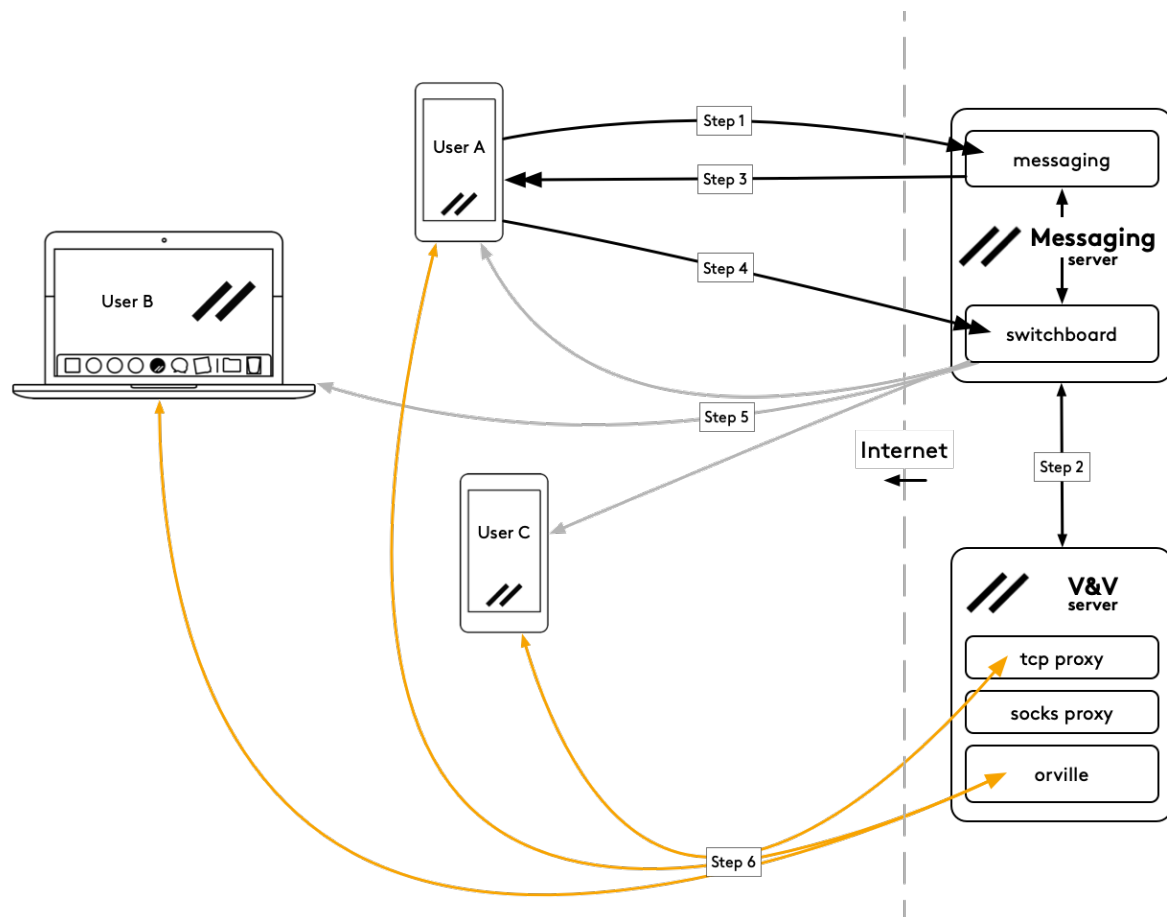
If the end clients can't connect to Wickr services, the first thing to check is if they're able to route to the Wickr address. You can verify this in a browser by appending `/checkConfig.php?api=117` to the **Application** server IP address over HTTPS.

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
```

9.2. Clients are unable to make calls

The following diagram shows how calls are connected to end clients:



1. User A initiates a call on their device. This sends the call start message to the messaging service.
2. The messaging service then requests a call start from the orville service. Orville replies with the call info.
3. Messaging then returns that call info to User A's device.
4. User A's device then sends that info to the switchboard service as a background message.
5. The switchboard service sends the call info to all recipients.
6. All call members then join the call via UDP or TCP.



Calls are attempted using UDP first, then within 3000ms a TCP attempt is made. Whichever connects first is used, unless forced to use TCP ahead of time.

Follow these steps to check calling:

1. Use the command `wickr test-calling` from the App server.

The output should look like this:

```
{"cluster":"enterprise","state":{"eventId":"5bda4906dd4d41ff996ada256cef5308","roomURI_ipv6":"","pid":"f985","roomURI":"12.34.56.78:16384"}}
```

If this command returns a blank line or a 404 error, this usually means the firewall rules on the servers aren't applied correctly. `firewalld` changes require the docker containers to be restarted after applying new rules so keep that in mind.

2. If `test-calling` returns a proper `roomURI` with the correct endpoint listed, then it is likely a routing issue connecting the clients to that. `traceroute`, `tcpdump`, or Wireshark can help identify where the issue is.

9.2.1. Calling Log Locations

The calling service doesn't log to a file on disk by default. Using the `docker logs orville` command on the Voice and Video server will display them. Alternatively, the docker process will save the recent log data to a temporary file on disk. This file changes each time the container is restarted, so to find the file it is necessary to run `docker inspect orville` and look for the `LogPath` value.

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).

- Traefik 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.

A.2. Voice and Video Services

- Zookeeper

The `zookeeper` container is the configuration management software for the Voice and Video services.

- Slave

The `slave` container starts, stops, and manages calls in progress sent from `master`.

- Master

The `master` container manages the `slave` container services, passing requests to start calls.

- Spinning Rooms

The `spinning-rooms` container handles incoming requests from the `nginx` container and forwards them to the `master` service.

- Nginx

The `nginx` container is a reverse proxy which sends requests to the `spinning-rooms` container. It hides the `spinning-rooms` service from the internet and also acts as a layer 7 load balancer.