# epiKshare Administration Guide

## Release 8.2

## Introduction

Welcome to the epiKshare Server Administration Guide. This guide describes administration tasks for epiKshare, the exible open source le synchronization and sharing solution. epiKshare includes the epiKshare server, which runs on Linux, client applications for Microsoft Windows, Mac OS X and Linux, and mobile clients for the Android and Apple iOS operating systems.

## Target Audience

This guide is for users who want to install, administer, and optimize their epiKshare servers. To learn more about the epiKshare Web user interface, and desktop and mobile clients, please refer to their respective manuals:

• epiKshare User Manual

• epiKshare Desktop Client

• epiKshare Android App

• epiKshare iOS App

## System Requirements

### Memory

Memory requirements for running an epiKshare server are greatly variable, depending on the numbers of users and les, and volume of server activity. epiKshare needs a minimum of 128MB RAM, and we recommend a minimum of 512MB.

## epiKshare Deployment Recommendations

What is the best way to install and maintain epiKshare? The answer to that is “it depends” because every epiKshare customer has their own particular needs and IT infrastructure. epiKshare and the LAMP stack are highly-congurable, so we will present three typical scenarios and make best-practice recommendations for both software and hardware.

### General Recommendations

Note: Whatever the size of your organization, always keep one thing in mind: The amount of data stored in epiKshare will only grow. Plan ahead.

Consider setting up a scale-out deployment, or using Federated Cloud Sharing to keep individual epiKshare instances to a manageable size.

#### Small Workgroups or Departments

• Number of users Up to 150 users.

• Storage size 100 GB to 10TB.

• High availability level

• Zero-downtime backups via Btrfs snapshots, component failure leads to interruption of service. Alternate backup scheme on other lesystems: nightly backups with service interruption.

**GRAFIK: epiKshare-Server->LDAP-Server/WinAD**

### Recommended System Requirements

One machine running the application server, Webserver, database server and local storage. Authentication via an existing LDAP or Active Directory server.

• Components One server with at least 2 CPU cores, 16GB RAM, local storage as needed.

• SSL Conguration The SSL termination is done in Apache. A standard SSL certicate is needed. For installation assistance, please contact the epiKshare support.

• Load Balancer: None.

• Backup: Install epiKshare, epiKshare data directory and database on Btrfs lesystem. Make regular snapshots at desired intervals for zero downtime backups. Mount DB partitions with the “nodatacow” option to prevent fragmentation. Alternatively, make nightly backups.

• Authentication: User authentication via one or several LDAP or Active Directory servers. (See User Authentication with LDAP for information on conguring epiKshare to use LDAP and AD.)

• Session Management: Local session management on the application server. PHP sessions are stored in a tmpfs mounted at the operating system-specic session storage location.

• Memory Caching: A memcache speeds up server performance, and epiKshare supports four memcaches; please contact the epiKshare support in order to get assistance with the activation of memory caching mechanisms.

• Storage: Local storage.

## Mid-sized Enterprises

• Number of users: 150 to 1,000 users.

• Storage size: Up to 200TB.

• High availability level: Every component is fully redundant and can fail without service interruption. Backups without service interruption.

### Recommended System Requirements

• 2 to 4 application servers with 4 sockets and 32GB RAM.

• A cluster of two database servers

• Storage on an NFS server.

• Authentication via an existing LDAP or Active Directory server.

**GRAFIK: Admin S 16**

• 2 DB servers with 4 sockets and 64GB RAM.

• 1 HAproxy load balancer with 2 sockets and 16GB RAM

• NFS storage server as needed.

• SSL Conguration: The SSL termination is done in the HAProxy load balancer. A standard SSL certicate is needed, installed according to the HAProxy documentation.

• LoadBalancer: HAProxy running on a dedicated server in front of the application servers. Sticky session needs to be used because of local session management on the application servers.

• Backup: Minimum daily backup without downtime. Please contact our support for more information.

• Session Management: Session management on the application server. PHP sessions are stored in a tmpfs mounted at the operating system-specic session storage location.

• Memory Caching: A memcache speeds up server performance, and epiKshare supports four memcaches; contact an epiKshare tech representative for guidance on selecting and conguring a memcache.

• Storage: Use an off-the-shelf NFS solution, such as IBM Elastic Storage, RedHat Ceph or SUSE Enterprise Storage.

## Large Enterprises and Service Providers

• Number of users: 5,000 to >100,000 users.

• Storage size: Up to 1 Petabyte

• High availability level: Every component is fully redundant and can fail without service interruption. Backups without service interruption

### Recommended System Requirements

• 4 to 20 application/Webservers.

• A cluster of two or more database servers.

• Storage is an NFS server, or an object store that is S3 compatible.

• Cloud federation for a distributed setup over several data centers.

• Authentication via an existing LDAP or Active Directory server, or SAML.

**GRAFIK: Admin S 17**

### Hardware Considerations

• Solid-state drives (SSDs) for I/O.

• Separate hard disks for storage and database, SSDs for databases.

• Multiple network interfaces to distribute server synchronisation and backend trafc across multiple subnets.

-Single Machine / Scale-Up Deployment

The single-machine deployment is widely used in the community.

Pros:

• Easy setup: no session storage daemon, use tmpfs and memory caching to enhance performance, local storage.

• No network latency to consider.

• To scale buy a bigger CPU, more memory, larger hard drive, or additional hard drives.

Cons:

• Fewer high availability options.

• The amount of data in epiKshare tends to continually grow. Eventually a single machine will not scale; I/O performance decreases and becomes a bottleneck with multiple up- and downloads, even with solid-state drives.

### Scale-Out Deployment

Scale-Out Deployment

Provider setup:

• DNS round robin to HAProxy servers (2-n, SSL ofoading, cache static resources)

• Least load to Apache servers (2-n)

• Memcached/Redis for shared session storage (2-n)

• Database cluster with single Master, multiple slaves and proxy to split requests accordingly (2-n)

• GPFS or Ceph via phprados (2-n, 3 to be safe, Ceph 10+ nodes to see speed benets under load)

Pros:

• Components can be scaled as needed.

• High availability.

• Test migrations easier.

Cons:

• More complicated to setup.

• Network becomes the bottleneck (10GB Ethernet recommended).

• Currently DB le cache table will grow rapidly, making migrations painful in case the table is altered.

#### What about Nginx / PHP-FPM?

Could be used instead of Haproxy as the load balancer. But on uploads stores the whole le on disk before handing it over to PHP-FPM.

#### A Single Master DB is Single Point of Failure, Does Not Scale

When master fails another slave can become master. However, the increased complexity carries some risks: Multi master has the risk of splitbrain, and deadlocks. EpiKshare tries to solve the problem of deadlocks with high-level le locking.

#### File Storage

While many customers are starting with NFS, sooner or later that requires scale-out storage. Currently the options are GPFS or GlusterFS, or an object store protocol like S3 (supported in Enterprise Edition only) or Swift. S3 also allows access to Ceph Storage.

#### Session Storage

• Redis: provides persistence, nice graphical inspection tools available, supports epiKshare high-level lelocking.

• If Shibboleth is a requirement you must use Memcached, and it can also be used to scale-out shibd session storage (see Memcache StorageService).

## Installation notices

### Downgrading not supported

Downgrading is not supported and risks corrupting your data! If you need to revert to an older epiKshare version, install it from scratch and then restore your data from backup. Before doing this, le a support ticket and ask for help to see if your issue can be resolved without downgrading.

### Additional Installation Guides and Notes

For guidance on how to install the epiKshare appliance, please refer to our guide "Deploying an epiKshare server on-premise".

## Trusted Domains

All URLs used to access your epiKshare server must be whitelisted in your config le, under the trusted_domains setting. Users are allowed to log into epiKshare only when they point their browsers to a URL that is listed in the trusted_domains setting. You may use IP addresses and domain names. A typical conguration looks like this:

'trusted_domains' => array ( 0 => 'localhost', 1 => 'server1.example.com', 2 => '192.168.1.50', ),

The loopback address, 127.0.0.1, is automatically whitelisted, so as long as you have access to the physical server you can always login. In the event that a load balancer is in place there will be no issues as long as it sends the correct X-Forwarded-Host header. When a user tries a URL that is not whitelisted, an error appears – telling that the user is accessing epiKshare from an untrusted domain. If clicking the button on the warning message site doesn't work, please contact the epiKshare support to manually edit the configuration file for you.

## Setting Strong Directory Permissions

For hardened security we recommend setting the permissions on your epiKshare directories as strictly as possible, and for proper server operations. This should be done immediately after the initial installation and before running the setup. Your HTTP user must own the config/, data/ and apps/ directories so that you can congure epiKshare, create, modify and delete your data les, and install apps via the epiKshare Web interface. You can nd your HTTP user in your HTTP server conguration les. Or you can use PHP Version and Information (Look for the User/Group line).

* The HTTP user in the epiKshare appliance is *wwwrun*, and the HTTP group is *www*

Note: When using an NFS mount for the data directory, do not change its ownership from the default. The simple act of mounting the drive will set proper permissions for epiKshare to write to the directory. Changing ownership as above could result in some issues if the NFS mount is lost.

The easy way to set the correct permissions is to copy and run the following script. Replace the espath variable with the path to your epiKshare directory, and replace the htuser and htgroup variables with your HTTP user and group:

```
#!/bin/bash

espath='/var/www/epiKshare'

htuser='www-data'

htgroup='www-data'

rootuser='root'

printf "Creating possible missing Directories\n"

mkdir -p $espath/data

mkdir -p $espath/assets

printf "chmod Files and Directories\n"

find ${espath}/ -type f -print0 | xargs -0 chmod 0640

find ${espath}/ -type d -print0 | xargs -0 chmod 0750

printf "chown Directories\n"

chown -R ${rootuser}:${htgroup} ${espath}/

chown -R ${htuser}:${htgroup} ${espath}/apps/

chown -R ${htuser}:${htgroup} ${espath}/config/

chown -R ${htuser}:${htgroup} ${espath}/data/

chown -R ${htuser}:${htgroup} ${espath}/themes/

chown -R ${htuser}:${htgroup} ${espath}/assets/

chmod +x ${espath}/occ

printf "chmod/chown .htaccess\n"

if [ -f ${espath}/.htaccess ] then
```

```
chmod 0644 ${espath}/.htaccess

chown ${rootuser}:${htgroup} ${espath}/.htaccess

fi

if [ -f ${espath}/data/.htaccess ] then

chmod 0644 ${espath}/data/.htaccess

chown ${rootuser}:${htgroup} ${espath}/data/.htaccess

fi
```

If you have customized your epiKshare installation and your lepaths are different than the standard installation, then modify this script accordingly. This lists the recommended modes and ownership for your epiKshare directories and les:

• All les should be read-write for the le owner, read-only for the group owner, and zero for the world

• All directories should be executable (because directories always need the executable bit set), read-write for the directory owner, and read-only for the group owner

• The apps/ directory should be owned by [HTTP user]:[HTTP group]

• The config/ directory should be owned by [HTTP user]:[HTTP group]

• The themes/ directory should be owned by [HTTP user]:[HTTP group]

• The assets/ directory should be owned by [HTTP user]:[HTTP group]

• The data/ directory should be owned by [HTTP user]:[HTTP group]

• The [espath]/.htaccess le should be owned by root:[HTTP group]

• The data/.htaccess le should be owned by root:[HTTP group]

• Both .htaccess les are read-write le owner, read-only group and world

These strong permissions prevent upgrading your epiKshare server. To change the permissions for an upgrade, please contact an epiKshare tech representative.

### Working with Apps and Adding Apps

Workin with unapproved apps and installing apps to your epiKshare appliance is currently not supported.

If you need extended functionality for your epiKshare installation, please contact the epiKshare support.

### Enabling SSL

*****Doku von Bene*****

### PHP-FPM

If you plan to use PHP-FPM, please contact the epiKshare support.

## epiKshare Server Configuration

### Warnings on Admin Page

Your epiKshare server has a built-in conguration checker, and it reports its ndings at the top of your Admin page. These are some of the warnings

you might see, and what to do about them.

### Security & setup warnings

**“No memory cache has been congured. To enhance your performance please congure a memcache if available.”**

You can signicantly improve your epiKshare server performance with memory caching, where frequently-requested objects are stored in memory for faster retrieval. There are two types of caches to use: a PHP opcode cache, which is commonly called opcache, and data caching for your Web server. If you do not install and enable a local memcache you will see a warning on your epiKshare admin page. A memcache is not required and you may safely ignore the warning if you prefer. If you need to enable caching, please get in touch with the epiKshare support.

**“You are accessing this site via HTTP. We strongly suggest you congure your server to require using HTTPS instead.”**

Please take this warning seriously; using HTTPS is a fundamental security measure. You must congure your Web server to support it, and then there are some settings in the Security section of your epiKshare Admin page to enable.

For guidance on how to enable SSL, please refer to "Enabling SSL".

## Configuring the Activity App

You can congure your epiKshare server to automatically send out e-mail notications to your users for various events like:

• A le or folder has been shared

• A new le or folder has been created

• A le or folder has been changed

• A le or folder has been deleted Users can see actions (delete, add, modify) that happen to les they have access to. Sharing actions are only visible to the sharer and sharee.

To congure your epiKshare to send out e-mail notications a working Email Conguration is mandatory. Please set up the email configuration on the Admin page.

## Enabling ClamAV

If you need to enable virus scanning on your epiKshare installation, please contact an epiKshare tech representative from the epiKshare support.

## Advanced configuration of background activities

If you need to change the configuration of background activities to CRON, please contact an epiKshare tech representative from the epiKshare support.

## Email Configuration

epiKshare is capable of sending password reset emails, notifying users of new leshares, changes in les, and activity notications. Your users cong ure which notications they want to receive on their Personal pages. epiKshare does not contain a full email server, but rather connects to your existing mail server. You must have a functioning mail server for epiKshare to be able to send emails. You may have a mail server on the same machine as epiKshare, or it may be a remote server.

To access the email configuration wizard, please go to the "Admin" section.

The epiKshare Email wizard supports three types of mail server connections: SMTP, PHP, and Sendmail. Use the SMTP congurator for a remote server, and PHP or Sendmail when your mail server is on the same machine as epiKshare.

Note: The Sendmail option refers to the Sendmail SMTP server, and any drop-in Sendmail replacement such as Postx, Exim, or Courier. All of these include a sendmail binary, and are freely-interchangeable.

## Configuring an SMTP Server

You need the following information from your mailserver administrator to connect epiKshare to a remote SMTP server:

• Encryption type: None, SSL, or TLS

• The From address you want your outgoing epiKshare mails to use

• Whether authentication is required

• Authentication method: None, Login, Plain, or NT LAN Manager • The server's IP address or fully-qualied domain name

• Login credentials, if required

Your changes are saved immediately, and you can click the Send Email button to test your conguration. This sends a test message to the email address you congured on your Personal page. The test message says:

If you received this email, the settings seem to be correct.

## Configuring PHP and Sendmail

Conguring PHP or Sendmail requires only that you select one of them, and then enter your desired return address.

How do you decide which one to use?

In most cases the smtp option is best, because it removes the extra step of passing through PHP, and you can control all of your mail server options in one place, in your mail server conguration.

## Using Email Templates

Another useful new feature is editable email templates. Now you can edit epiKshare's email templates on your Admin page. These are your available templates: • Sharing email (HTML) – HTML version of emails notifying users of new le shares

• Sharing email (plain text fallback) –

• Plain text email notifying users of new le shares

• Lost password mail

• Password reset email for users who lose their passwords.

• Activity notication mail

• Notication of activities that users have enabled in the Notications section of their Personal pages.

In addition to providing the email templates, this feature enables you to apply any precongured themes to the email.

To modify an email template to users:

1. Access the Admin page.

2. Scroll to the Mail templates section.

3. Select a template from the drop-down menu.

4. Make any desired modications to the template. The templates are written in PHP and HTML, and are already loaded with the relevant variables such as username, share links, and lenames. You can, if you are careful, edit these even without knowing PHP or HTML; don't touch any of the code, but you can edit the text portions of the messages. For example, this the lost password mail template:

```
<?php

echo str_replace('{link}', $_['link'], $l->t('Use the following link to reset your password: {link}'));
```

You could change the text portion of the template, Use the following link to reset your password: to say something else, such as Click the following link to reset your password. If you did not ask for a password reset, ignore this message. Again,be very careful to change nothing but the message text, because the tiniest coding error will break the template

Note: You can edit the templates directly in the template text box, or you can copy and paste them to a text editor for modication and then copy and paste them back to the template text box for use when you are done.

### Linking External Sites

This is useful for quick access to important Web pages such as the epiKshare manuals and informational pages for your company, and for presenting external pages.

The External sites app is included in all versions of epiKshare. Go to Apps > Not Enabled to enable it. Then go to your epiKshare Admin page to create your links, which are saved automatically. There is a dropdown menu to select an icon, but there is only one default icon so you don't have to select one. Hover your cursor to the right of your links to make the trashcan icon appear when you want to remove them.

The links appear in the epiKshare dropdown menu on the top left after refreshing your page, and have globe icons. Your links may or may not work correctly due to the various ways that Web browsers and Web sites handle HTTP and HTTPS URLs, and because the External Sites app embeds external links in IFrames. Modern Web browsers try very hard to protect Web surfers from dangerous links, and safety apps like PrivacyBadger and ad-blockers mayblock embedded pages. It is strongly recommended to enforce HTTPS on your epiKshare server; do not weaken this, or any of your security tools, just to make embedded Web pages work. After all, you can freely access them outside of epiKshare. Most Web sites that offer login functionalities use the X-Frame-Options or Content-Security-Policy HTTP header which instructs browsers to not allow their pages to be embedded for security reasons (e.g. "Clickjacking"). You can usually verify the reason why embedding the website is not possible by using your browser's console tool. For example, this page has an invalid SSL certicate. On this page, X-Frame-Options prevents the embedding.

There isn't much you can do about these issues, but if you're curious you can see what is happening.

### Language Configuration

In normal cases epiKshare will automatically detect the language of the Web-GUI. If this does not work properly or you want to make sure that epiKshare always starts with a given language, please contact the epiKshare support to have your configuration changed accordingly.

### Logging Configuration

Use your epiKshare log to review system status, or to help debug problems. Logging levels range from DEBUG, which logs all activity, to FATAL, which logs only fatal errors.

• 0: DEBUG: All activity; the most detailed logging.

• 1: INFO: Activity such as user logins and le activities, plus warnings, errors, and fatal errors.

• 2: WARN: Operations succeed, but with warnings of potential problems, plus errors and fatal errors.

• 3: ERROR: An operation fails, but other services and operations continue, plus fatal errors.

• 4: FATAL: The server stops.

By default the log level is set to 2 (WARN). Use DEBUG when you have a problem to diagnose, and then reset your log level to a less-verbose level as DEBUG outputs a lot of information, and can affect your server performance.

## Hardening and Security Guidance

epiKshare aims to ship with secure defaults that do not need to get modied by administrators. However, in some cases some additional security hardening can be applied in scenarios where the administrator has complete control over the epiKshare instance.

Note: epiKshare will warn you in the administration interface if some critical security-relevant options are missing. However, it is still up to the server administrator to review and maintain system security.

**Please inform an epiKshare tech representative if you need advice to secure and/or harden your system.**

### Deployment

For guidance on how to deploy epiKshare, please refer to this article in our knowledge base.

### Ensure that your epiKshare instance is installed in a DMZ

As epiKshare supports features such as Federated File Sharing we do not consider Server Side Request Forgery (SSRF) part of our threat model. In fact, given all our external storage adapters this can be considered a feature and not a vulnerability. This means that a user on your epiKshare instance could probe whether other hosts are accessible from the epiKshare network. If you do not want this you need to ensure that your epiKshare is properly installed in a segregated network and proper rewall rules are in place.

### Reverse Proxy Configuration

epiKshare can be run through a reverse proxy, which can cache static assets such as images, CSS or JS les, move the load of handling HTTPS to a different server or load balance between multiple servers. If you want to run epiKshare with behind a reverse proxy, please contact an epiKshare tech representative.

## Server Tuning & Performance Tips

#### Use cron to perform background jobs

If you want to use CRON to perform background tasks, please contact an epiKshare tech representative.

#### Caching

If you need to enable memory caching for your epiKshare installation, please contact an epiKshare tech representative.

#### JavaScript and CSS Asset Management

In production environments, JavaScript and CSS les should be delivered in a concatenated and compressed format. If you need to change settings regarding the delivery of JavaScript and CSS files, please contact the epiKshare support.

## User Management

On the User management page of your epiKshare Web UI you can:

• Create new users

• View all of your users in a single scrolling window

• Filter users by group

• See what groups they belong to

• Edit their full names and passwords

• See their data storage locations

• View and set quotas

• Create and edit their email addresses

• Send an automatic email notication to new users

• Delete them with a single click The default view displays basic information about your users.

**GRAFIK: UserManagement**

The Group lters on the left sidebar lets you quickly lter users by their group memberships, and create new groups. Click the gear icon on the lower

left sidebar to set a default storage quota, and to display additional elds: Show storage location, Show last login, Show user backend, Send email to new users, and Show email address.

User accounts have the following properties: Login Name (Username) The unique ID of an epiKshare user, and it cannot be changed. Full Name The user's display name that appears on leshares, the epiKshare Webinterface, and emails. Admins and users may change the Full Name anytime. If the Full Name is not set it defaults to the login name. Password The admin sets the new user's rst password. Both the user and the admin can change the user's password at anytime. Groups You may create groups, and assign group memberships to users. By default new users are not assigned to any groups. Group Admin Group admins are granted administrative privileges on specic groups, and can add and remove users from their groups. Quota The maximum disk space assigned to each user. Any user that exceeds the quota cannot upload or sync data. You have the the option to include external storage in user quotas.

## Creating a New User

To create a user account:

• Enter the new user's Login Name and their initial Password

• Optionally, assign Groups memberships

• Click the Create button

Login names may contain letters (a-z, A-Z), numbers (0-9), dashes (-), underscores (_), periods (.) and at signs (@). After creating the user, you may ll in their Full Name if it is different than the login name, or leave it for the user to complete. If you have checked Send email to new user in the control panel on the lower left sidebar, you may also enter the new user's email address, and epiKshare will automatically send them a noticat ion with their new login information. You may edit this email using the email template editor on your Admin page.

## Reset a User's Password

You cannot recover a user's password, but you can set a new one:

• Hover your cursor over the user's Password eld

• Click on the pencil icon

• Enter the user's new password in the password eld, and remember to provide the user with their password.

If you have encryption enabled, there are special considerations for user password resets. Please make sure you have created a recovery password for the user files before you reset a user's password.

## Renaming a User

Each epiKshare user has two names: a unique Login Name used for authentication, and a Full Name, which is their display name. You can edit the display name of a user, but you cannot change the login name of any user. To set or change a user's display name:

• Hover your cursor over the user's Full Name eld

• Click on the Pencil icon

• Enter the user's new display name

## Granting Administrator Privileges to a User

epiKshare has two types of administrators: Super Administrators and Group Administrators. Group administrators have the rights to create, edit and delete users in their assigned groups. Group administrators cannot access system settings, or add or modify users in the groups that they are not Group Administrators for. Use the dropdown menus in the Group Admin column to assign group admin privileges.

Super Administrators have full rights on your epiKshare server, and can access and modify all settings. To assign the Super Administrators role to a user, simply add them to the admin group.

## Managing Groups

You can assign new users to groups when you create them, and create new groups when you create new users. You may also use the Add Group button at the top of the left pane to create new groups. New group members will immediately have access to file shares that belong to their new groups.

### Setting Storage Quotas

Click the gear on the lower left pane to set a default storage quota. This is automatically applied to new users. You may assign a different quota to any user by selecting from the Quota dropdown, selecting either a preset value or entering a custom value. When you create custom quotas, use the normal abbreviations for your storage values such as 500 MB, 5 GB, 5 TB, and so on.

Metadata (such as thumbnails, temporary files, and encryption keys) takes up about 10% of disk space, but is not counted against user quotas. Users can check their used and available space on their Personal pages. Only files that originate with users count against their quotas, and not files shared with them that originate from other users. For example, if you upload files to a different user's share, those files count against your quota. If you re-share a file that another user shared with you, that file does not count against your quota, but the originating user's.

Encrypted files are a little larger than unencrypted files; the unencrypted size is calculated against the user's quota.

Deleted files that are still in the trash bin do not count against quotas. The trash bin is set at 50% of quota. Deleted file aging is set at 30 days. When deleted files exceed 50% of quota then the oldest files are removed until the total is below 50%.

When version control is enabled, the older file versions are not counted against quotas.

When a user creates a public share via URL, and allows uploads, any uploaded files count against that user's quota.

### Deleting users

Deleting a user is easy: hover your cursor over their name on the Users page until a trashcan icon appears at the far right. Click the trashcan, and they're gone. You'll see an undo button at the top of the page, which remains until you refresh the page. When the undo button is gone you cannot recover the deleted user.

All of the files owned by the user are deleted as well, including all files they have shared. If you need to preserve the user's files and shares, you must first download them from your epiKshare Files page, which compresses them into a zip file, or use a sync client to copy them to your local computer. See *File Sharing* to learn how to create persistent file shares that survive user deletions.

### Resetting a Lost Admin Password

The normal ways to recover a lost password are:

1. Click the password reset link on the login screen; this appears after a failed login attempt. This works only if you have entered your email address on your Personal page in the epiKshare Web interface, so that the epiKshare server can email a reset link to you.
2. Ask another epiKshare server admin to reset it for you.

If neither of these is an option, then you have a third option, and that is contacting the epiKshare support.

### User Authentication with IMAP, SMB, and FTP

> Note: A non-blocking or correctly configured SELinux setup is needed for these backends to work.

If you need to use one of the following authentication methods, please ask an epiKshare technician for assistance: IMAP, SMB, FTP.

## User Authentication with LDAP

epiKshare ships with an LDAP application to allow LDAP users (including Active Directory) to appear in your epiKshare user listings. These users will authenticate to epiKshare with their LDAP credentials, so you don't have to create separate epiKshare user accounts for them. You will manage their epiKshare group memberships, quotas, and sharing permissions just like any other epiKshare user.

> Note: The PHP LDAP module is required. It is included in the epiKshare appliance.

The LDAP application supports:

- LDAP group support
- File sharing with epiKshare users and groups

- Access via WebDAV and epiKshare Desktop Client
- Versioning, external Storage and all other epiKshare features
- Seamless connectivity to Active Directory, with no extra configuration required
- Support for primary groups in Active Directory
- Auto-detection of LDAP attributes such as base DN, email, and the LDAP server port number
- Only read access to your LDAP (edit or delete of users on your LDAP is not supported)

Warning: The LDAP app is not compatible with the User backend using remote HTTP servers app. You cannot use both of them at the same time.

> Note: A non-blocking or correctly configured SELinux setup is needed for the LDAP backend to work.

## Configuration

> **Warning**
> We strongly encourage you to get help with the configuration from the epiKshare team. Please only proceed if you know exactly what you're doing.

First enable the LDAP user and group backend app on the Apps page in epiKshare. Then go to your Admin page to configure it.

The LDAP configuration panel has four tabs. A correctly completed first tab (“Server”) is mandatory to access the other tabs. A green indicator lights when the configuration is correct. Hover your cursor over the fields to see some pop-up tooltips.

## Server Tab

Start with the Server tab. You may configure multiple servers if you have them. At a minimum you must supply the LDAP server’s hostname. If your server requires authentication, enter your credentials on this tab. epiKshare will then attempt to auto-detect the server’s port and base DN. The base DN and port are mandatory, so if epiKshare cannot detect them you must enter them manually.

Server configuration: Configure one or more LDAP servers. Click the Delete Configuration button to remove the active configuration.

Host: The host name or IP address of the LDAP server. It can also be a ldaps:// URI. If you enter the port number, it speeds up server detection.

**Examples:**

- *directory.my-company.com*
- *ldaps://directory.my-company.com*
- *directory.my-company.com:9876*

Port: The port on which to connect to the LDAP server. The field is disabled in the beginning of a new configuration. If the LDAP server is running on a standard port, the port will be detected automatically. If you are using a non-standard port, epiKshare will attempt to detect it. If this fails you must enter the port number manually.

**Example:**

- *389*

User DN: The name as DN of a user who has permissions to do searches in the LDAP directory. Leave it empty for anonymous access. We recommend that you have a special LDAP system user for this.

**Example:**

- *uid=epiKsharesystemuser,cn=sysusers,dc=my-company,dc=com*

Password: The password for the user given above. Empty for anonymous access.

Base DN: The base DN of LDAP, from where all users and groups can be reached. You may enter multiple base DNs, one per line. (Base DNs for users and groups can be set in the Advanced tab.) This field is mandatory. epiKshare attempts to determine the Base DN

according to the provided User DN or the provided Host, and you must enter it manually if epiKshare does not detect it.

**Example:**

- *dc=my-company,dc=com*

## User Filter

Use this to control which LDAP users are listed as epiKshare users on your epiKshare server. In order to control which LDAP users can login to your epiKshare server use the Login filter. Those LDAP users who have access but are not listed as users (if there are any) will be hidden users. You may bypass the form fields and enter a raw LDAP filter if you prefer.

only those object classes: epiKshare will determine the object classes that are typically available for user objects in your LDAP. epiKshare will automatically select the object class that returns the highest amount of users. You may select multiple object classes.

only from those groups: If your LDAP server supports the member-of-overlay in LDAP filters, you can define that only users from one or more certain groups are allowed to appear in user listings in epiKshare. By default, no value will be selected.

You may select multiple groups.

If your LDAP server does not support the member-of-overlay in LDAP filters, the input field is disabled. Please contact your LDAP administrator.

Edit raw filter instead: Clicking on this text toggles the filter mode and you can enter the raw LDAP filter directly.

**Example:**

**(&(objectClass=inetOrgPerson)(memberOf=cn=epiKshareusers,ou=groups, dc=example,dc=com))**

x users found: This is an indicator that tells you approximately how many users will be listed in epiKshare. The number updates automatically after any changes.

## Login Filter

The settings in the Login Filter tab determine which LDAP users can log in to your epiKshare system and which attribute or attributes the provided login name is matched against (e.g. LDAP/AD username, email address). You may select multiple user details. (You may bypass the form fields and enter a raw LDAP filter if you prefer.)

You may override your User Filter settings on the User Filter tab by using a raw LDAP filter.

LDAP Username: If this value is checked, the login value will be compared to the username in the LDAP directory. The corresponding attribute, usually *uid* or *samaccountname* will be detected automatically by epiKshare.

LDAP Email Address: If this value is checked, the login value will be compared to an email address in the LDAP directory; specifically, the *mailPrimaryAddress* and *mail* attributes.

Other Attributes: This multi-select box allows you to select other attributes for the comparison. The list is generated automatically from the user object attributes in your LDAP server.

Edit raw filter instead: Clicking on this text toggles the filter mode and you can enter the raw LDAP filter directly.

The %uid placeholder is replaced with the login name entered by the user upon login.

**Examples:**

- only username:

```
(&(objectClass=inetOrgPerson)(memberOf=cn=epiKshareusers,ou=groups, dc=example,dc=com)(uid=%uid)
```

- username or email address:

```
((&(objectClass=inetOrgPerson)(memberOf=cn=epiKshareusers,ou=groups,
dc=example,dc=com)(|(uid=%uid)(mail=%uid)))
```

## Group Filter

By default, no LDAP groups will be available in epiKshare. The settings in the group filter tab determine which groups will be available in

epiKshare. You may also elect to enter a raw LDAP filter instead.

only those object classes: epiKshare will determine the object classes that are typically available for group objects in your LDAP server. epiKshare will only list object classes that return at least one group object. You can select multiple object classes. A typical object class is “group”, or “posixGroup”.

only from those groups: epiKshare will generate a list of available groups found in your LDAP server. and then you select the group or groups that get access to your epiKshare server.

Edit raw filter instead: Clicking on this text toggles the filter mode and you can enter the raw LDAP filter directly.

**Example:**

- `objectClass=group`
- `objectClass=posixGroup`

y groups found: This tells you approximately how many groups will be available in epiKshare. The number updates automatically after any change.

## Advanced Settings

The LDAP Advanced Setting section contains options that are not needed for a working connection. This provides controls to disable the current configuration, configure replica hosts, and various performance-enhancing options.

The Advanced Settings are structured into three parts:

- Connection Settings
- Directory Settings
- Special Attributes

### Connection Settings

Configuration Active: Enables or Disables the current configuration. By default, it is turned off. When epiKshare makes a successful test connection it is automatically turned on.

Backup (Replica) Host: If you have a backup LDAP server, enter the connection settings here. epiKshare will then automatically connect to the backup when the main server cannot be reached. The backup server must be a replica of the main server so that the object UUIDs match.

**Example:**

- *directory2.my-company.com*

Backup (Replica) Port: The connection port of the backup LDAP server. If no port is given, but only a host, then the main port (as specified above) will be used.

Disable Main Server: You can manually override the main server and make epiKshare only connect to the backup server. This is useful for planned downtimes.

Turn off SSL certificate validation: Turns off SSL certificate checking. Use it for testing only!

Cache Time-To-Live: A cache is introduced to avoid unnecessary LDAP traffic, for example caching usernames so they don’t have to be looked up for every page, and speeding up loading of the Users page. Saving the configuration empties the cache. The time is given in seconds.

Note that almost every PHP request requires a new connection to the LDAP server. If you require fresh PHP requests we recommend defining a minimum lifetime of 15s or so, rather than completely eliminating the cache.

**Examples:**

- ten minutes: *600*
- one hour: *3600*

See the Caching section below for detailed information on how the cache operates.

## Directory Settings

User Display Name Field: The attribute that should be used as display name in epiKshare.

- **Example:**

```
displayName
```

2nd User Display Name Field: An optional second attribute displayed in brackets after the display name, for example using the mail attribute displays as Molly Foo (molly@example.com).

Base User Tree: The base DN of LDAP, from where all users can be reached. This must be a complete DN, regardless of what you have entered for your Base DN in the Basic setting. You can specify multiple base trees, one on each line.

- **Example:**

```
cn=programmers,dc=my-company,dc=com cn=designers,dc=my-company,dc=com
```

User Search Attributes: These attributes are used when searches for users are performed, for example in the share dialogue. The user display name attribute is the default. You may list multiple attributes, one per line.

If an attribute is not available on a user object, the user will not be listed, and will be unable to login. This also affects the display name attribute. If you override the default you must specify the display name attribute here.

- **Example:**

```
displayName mail
```

Group Display Name Field: The attribute that should be used as epiKshare group name. epiKshare allows a limited set of characters (a-zA-Z0-9.-_@). Once a group name is assigned it cannot be changed.

- **Example:**

```
cn
```

Base Group Tree: The base DN of LDAP, from where all groups can be reached. This must be a complete DN, regardless of what you have entered for your Base DN in the Basic setting. You can specify multiple base trees, one in each line.

- **Example:**

```
cn=barcelona,dc=my-company,dc=com cn=madrid,dc=my-company,dc=com
```

Group Search Attributes: These attributes are used when a search for groups is done, for example in the share dialogue. By default the group display name attribute as specified above is used. Multiple attributes can be given, one in each line.

If you override the default, the group display name attribute will not be taken into account, unless you specify it as well.

- **Example:**

```
cn description
```

Group Member association: The attribute that is used to indicate group memberships, i.e. the attribute used by LDAP groups to refer to their users.

epiKshare detects the value automatically. You should only change it if you have a very valid reason and know what you are doing.

- **Example:**

```
uniquemember
```

## Special Attributes

Quota Field: epiKshare can read an LDAP attribute and set the user quota according to its value. Specify the attribute here, and it will return human-readable values, e.g. “2 GB”. Any quota set in LDAP overrides quotas set on the epiKshare user management page.

- **Example:**

```
epiKshareQuota
```

Quota Default: Override epiKshare default quota for LDAP users who do not have a quota set in the Quota Field.

- **Example:**

```
15 GB
```

Email Field: Set the user's email from their LDAP attribute. Leave it empty for default behavior.

- **Example:**

```
mail
```

User Home Folder Naming Rule: By default, the epiKshare server creates the user directory in your epiKshare data directory and gives it the epiKshare username, .e.g /var/www/epiKshare/data/alice. You may want to override this setting and name it after an LDAP attribute value. The attribute can also return an absolute path, e.g. /mnt/storage43/alice. Leave it empty for default behavior.

- **Example:**

```
cn
```

In all epiKshare installations the home folder rule is enforced. This means that once you set a home folder naming rule (get a home folder from an LDAP attribute), it must be available for all users. If it isn't available for a user, then that user will not be able to login. Also, the filesystem will not be set up for that user, so their file shares will not be available to other users.

## Expert Settings

In the Expert Settings fundamental behavior can be adjusted to your needs. The configuration should be well-tested before starting production use.

Internal Username: The internal username is the identifier in epiKshare for LDAP users. By default it will be created from the UUID attribute. The UUID attribute ensures that the username is unique, and that characters do not need to be converted. Only these characters are allowed: [a-zA-Z0-9_.@-]. Other characters are replaced with their ASCII equivalents, or are simply omitted.

The LDAP backend ensures that there are no duplicate internal usernames in epiKshare, i.e. that it is checking all other activated user backends (including local epiKshare users). On collisions a random number (between 1000 and 9999) will be attached to the retrieved value. For example, if "alice" exists, the next username may be "alice_1337".

The internal username is the default name for the user home folder in epiKshare. It is also a part of remote URLs, for instance for all *DAV services.

You can override all of this with the Internal Username setting. Leave it empty for default behaviour. Changes will affect only newly mapped LDAP users.

- **Example:**

```
uid
```

Override UUID detection: By default, epiKshare auto-detects the UUID attribute. The UUID attribute is used to uniquely identify LDAP users and groups. The internal username will be created based on the UUID, if not specified otherwise.

You can override the setting and pass an attribute of your choice. You must make sure that the attribute of your choice can be fetched for both users and groups and it is unique. Leave it empty for default behaviour. Changes will have effect only on newly mapped LDAP users and groups. It also will have effect when a user's or group's DN changes and an old UUID was cached, which will result in a new user. Because of this, the setting should be applied before putting epiKshare in production use and clearing the bindings (see the User and Group Mapping section below).

- **Example:**

```
cn
```

Username-LDAP User Mapping: epiKshare uses usernames as keys to store and assign data. In order to precisely identify and recognize users, each LDAP user will have a internal username in epiKshare. This requires a mapping from epiKshare username to LDAP user. The created username is mapped to the UUID of the LDAP user. Additionally the DN is cached as well to reduce LDAP interaction, but it is not used for identification. If the DN changes, the change will be detected by epiKshare by checking the UUID value.

The same is valid for groups.

The internal epiKshare name is used all over in epiKshare. Clearing the Mappings will have leftovers everywhere. Never clear the mappings in a production environment, but only in a testing or experimental server.

Clearing the Mappings is not configuration sensitive, it affects all LDAP configurations!

## Testing the configuration

The Test Configuration button checks the values as currently given in the input fields. You do not need to save before testing. By clicking on the button, epiKshare will try to bind to the epiKshare server using the settings currently given in the input fields. If the binding fails you'll see a yellow banner with the error message "The configuration is invalid. Please have a look at the logs for further details."

When the configuration test reports success, save your settings and check if the users and groups are fetched correctly on the Users page.

## epiKshare Avatar integration

epiKshare supports user profile pictures, which are also called avatars. If a user has a photo stored in the *jpegPhoto* or *thumbnailPhoto* attribute on your LDAP server, it will be used as their avatar. In this case the user cannot alter their avatar (on their Personal page) as it must be changed in LDAP. *jpegPhoto* is preferred over *thumbnailPhoto*.

If the *jpegPhoto* or *thumbnailPhoto* attribute is not set or empty, then users can upload and manage their avatars on their epiKshare Personal pages. Avatars managed in epiKshare are not stored in LDAP.

The *jpegPhoto* or *thumbnailPhoto* attribute is fetched once a day to make sure the current photo from LDAP is used in epiKshare. LDAP avatars override epiKshare avatars, and when an LDAP avatar is deleted then the most recent epiKshare avatar replaces it.

Photos served from LDAP are automatically cropped and resized in epiKshare. This affects only the presentation, and the original image is not changed.

## Troubleshooting, Tips and Tricks

### SSL Certificate Verification (LDAPS, TLS)

A common mistake with SSL certificates is that they may not be known to PHP. If you have trouble with certificate validation make sure that

- You have the certificate of the server installed on the epiKshare server
- The certificate is announced in the system's LDAP configuration file (usually */etc/ldap/ldap.conf*
- Using LDAPS, also make sure that the port is correctly configured (by default 636)

### Microsoft Active Directory

Compared to earlier epiKshare versions, no further tweaks need to be done to make epiKshare work with Active Directory. epiKshare will automatically find the correct configuration in the set-up process.

### memberOf / Read MemberOf permissions

If you want to use memberOf within your filter you might need to give your querying user the permissions to use it. For Microsoft Active Directory this is described here.

### Duplicating Server Configurations

In case you have a working configuration and want to create a similar one or "snapshot" configurations before modifying them you can do the following:

1. Go to the Server tab
2. On Server Configuration choose *Add Server Configuration*
3. Answer the question *Take over settings from recent server configuration?* with *yes*.
4. (optional) Switch to Advanced tab and uncheck Configuration Active in the *Connection Settings*, so the new configuration is not used on Save
5. Click on Save

Now you can modify and enable the configuration.

## epiKshare LDAP Internals

### User and Group Mapping

In epiKshare the user or group name is used to have all relevant information in the database assigned. To work reliably a permanent internal user

name and group name is created and mapped to the LDAP DN and UUID. If the DN changes in LDAP it will be detected, and there will be no conflicts.

Those mappings are done in the database table ldap_user_mapping and ldap_group_mapping. The user name is also used for the user's folder (except if something else is specified in *User Home Folder Naming Rule*), which contains files and meta data.

As of epiKshare 5 the internal user name and a visible display name are separated. This is not the case for group names, yet, i.e. a group name cannot be altered.

That means that your LDAP configuration should be good and ready before putting it into production. The mapping tables are filled early, but as long as you are testing, you can empty the tables any time. Do not do this in production.

### Caching

The LDAP cache is only a memory cache, and you must install and configure the memory cache. The epiKshare Cache helps to speed up user interactions and sharing. It is populated on demand, and remains populated until the Cache Time-To-Live for each unique request expires. User logins are not cached, so if you need to improve login times set up a slave LDAP server to share the load.

In order to enable caching, please get in touch with the epiKshare support.

### Handling with Backup Server

When epiKshare is not able to contact the main LDAP server, epiKshare assumes it is offline and will not try to connect again for the time specified in Cache Time-To-Live. If you have a backup server configured epiKshare will connect to it instead. When you have scheduled downtime, check Disable Main Server to avoid unnecessary connection attempts.

### LDAP User Cleanup

LDAP User Cleanup is a new feature in the LDAP user and group backend application. LDAP User Cleanup is a background process that automatically searches the epiKshare LDAP mappings table, and verifies if the LDAP users are still available. Any users that are not available are marked as deleted in the database. Please consider to get technical assistance by the epiKshare support in order to automate this task.

## User Provisioning API

The Provisioning API application enables a set of APIs that external systems can use to create, edit, delete and query user attributes, query, set and remove groups, set quota and query total storage used in epiKshare. Group admin users can also query epiKshare and perform the same functions as an admin for groups they manage. The API also enables an admin to query for active epiKshare applications, application info, and to enable or disable an app remotely. HTTP requests can be used via a Basic Auth header to perform any of the functions listed above. The Provisioning API app is enabled by default.

The base URL for all calls to the share API is epiKshare_base_url/ocs/v1.php/cloud.

### Instruction Set For Users

#### users / adduser

Create a new user on the epiKshare server. Authentication is done by sending a basic HTTP authentication header. Syntax: ocs/v1.php/cloud/users

- HTTP method: POST
- POST argument: userid - string, the required username for the new user
- POST argument: password - string, the required password for the new user

Status codes:

- 100 - successful
- 101 - invalid input data
- 102 - username already exists
- 103 - unknown error occurred whilst adding the user

##### Example

- `POST http://admin:secret@example.com/ocs/v1.php/cloud/users -d userid="Frank" -d password="frankspassword"`

- Creates the user Frank with password frankspassword

#### XML Output

```
<?xml version="1.0"?>
<ocs>
<meta>
<status>ok</status>
<statuscode>100</statuscode>
<message/>
</meta>
<data/>
</ocs>
```

### users / getusers

Retrieves a list of users from the epiKshare server. Authentication is done by sending a Basic HTTP Authorization header.

Syntax: ocs/v1.php/cloud/users

- HTTP method: GET
- url arguments: search - string, optional search string
- url arguments: limit - int, optional limit value
- url arguments: offset - int, optional offset value

Status codes:

- 100 - successful

#### Example

- `GET http://admin:secret@example.com/ocs/v1.php/cloud/users?search=Frank`
- Returns list of users matching the search string.

#### XML Output

```
<?xml version="1.0"?>
<ocs>
<meta>
<statuscode>100</statuscode>
<status>ok</status>
</meta>
<data>
<users>
<element>Frank</element>
</users>
</data> </ocs>
```

### users / getuser

Retrieves information about a single user. Authentication is done by sending a Basic HTTP Authorization header. Syntax:

ocs/v1.php/cloud/users/{userid}

- HTTP method: GET

Status codes:

- 100 - successful

### Example

- `GET http://admin:secret@example.com/ocs/v1.php/cloud/users/Frank`
- Returns information on the user Frank

### XML Output

```
<?xml version="1.0"?>
<ocs>
<meta>
<statuscode>100</statuscode>
<status>ok</status>
</meta>
<data> <email>frank@example.org</email>
<quota>0</quota>
<enabled>true</enabled>
</data> </ocs>
```

## users / edituser

Edits attributes related to a user. Users are able to edit email, displayname and password; admins can also edit the quota value. Authentication is done by sending a Basic HTTP Authorization header.

Syntax: ocs/v1.php/cloud/users/{userid}

- HTTP method: PUT
- PUT argument: key, the field to edit (email, quota, display, password)
- PUT argument: value, the new value for the field

Status codes:

- 100 - successful
- 101 - user not found
- 102 - invalid input data

### Examples

- `PUT PUT http://admin:secret@example.com/ocs/v1.php/cloud/users/Frank -d key="email" -d value="franksnewemail@example.org"`
- Updates the email address for the user Frank
- `PUT PUT http://admin:secret@example.com/ocs/v1.php/cloud/users/Frank -d key="quota" -d value="100MB"`
- Updates the quota for the user Frank

### XML Output

```
<?xml version="1.0"?>
<ocs>
<meta>
```

```
<statuscode>100</statuscode>
<status>ok</status>
</meta>
<data/>
</ocs>
```

## users / deleteuser

Deletes a user from the epiKshare server. Authentication is done by sending a Basic HTTP Authorization header. Syntax: ocs/v1.php/cloud/users/{userid}

- HTTP method: DELETE

Statuscodes:

- 100 - successful
- 101 - failure

### Example

- `DELETE` `http://admin:secret@example.com/ocs/v1.php/cloud/users/Frank`

Deletes the user Frank

### XML Output

```
<?xml version="1.0"?>
<ocs>
<meta>
<statuscode>100</statuscode>
<status>ok</status>
</meta>
<data/> </ocs>
```

## users / getgroups

Retrieves a list of groups the specified user is a member of. Authentication is done by sending a Basic HTTP Authorization header.

Syntax: ocs/v1.php/cloud/users/{userid}/groups

- HTTP method: GET

Status codes:

- 100 - successful

### Example

- `GET` `http://admin:secret@example.com/ocs/v1.php/cloud/users/Frank/groups`
- Retrieves a list of groups of which Frank is a member

#### XML Output

```
<?xml version="1.0"?>
<ocs>
<meta>
<statuscode>100</statuscode>
<status>ok</status>
</meta>
<data>
<groups>
<element>admin</element>
<element>group1</element>
</groups>
</data> </ocs>
```

### users / addtogroup

Adds the specified user to the specified group. Authentication is done by sending a Basic HTTP Authorization header. Syntax: ocs/v1.php/cloud/users/{userid}/groups

- HTTP method: POST
- POST argument: groupid, string - the group to add the user to

Status codes:

- 100 - successful
- 101 - no group specified
- 102 - group does not exist
- 103 - user does not exist
- 104 - insufficient privileges
- 105 - failed to add user to group

#### Example

- `POST http://admin:secret@example.com/ocs/v1.php/cloud/users/Frank/groups -d groupid="newgroup"`
- Adds the user Frank to the group newgroup

#### XML Output

```
<?xml version="1.0"?>
<ocs>
<meta>
<statuscode>100</statuscode>
<status>ok</status>
</meta>
<data/>
</ocs>
```

### users / removefromgroup

Removes the specified user from the specified group. Authentication is done by sending a Basic HTTP Authorization header.

Syntax: ocs/v1.php/cloud/users/{userid}/groups

- HTTP method: DELETE
- POST argument: groupid, string - the group to remove the user from

Status codes:

- 100 - successful
- 101 - no group specified
- 102 - group does not exist
- 103 - user does not exist
- 104 - insufficient privileges
- 105 - failed to remove user from group

#### Example

- `DELETE http://admin:secret@example.com/ocs/v1.php/cloud/users/Frank/groups -d groupid="newgroup"`
- Removes the user Frank from the group newgroup

#### XML Output

```
<?xml version="1.0"?>
<ocs>
<meta>
<statuscode>100</statuscode>
<status>ok</status>
</meta>
<data/> </ocs>
```

### users / createsubadmin

Makes a user the subadmin of a group. Authentication is done by sending a Basic HTTP Authorization header. Syntax: ocs/v1.php/cloud/users/{userid}/subadmins

- HTTP method: POST
- POST argument: groupid, string - the group of which to make the user a subadmin

Status codes:

- 100 - successful
- 101 - user does not exist

1. - group does not exist
2. - unknown failure

#### Example

- `POST https://admin:secret@example.com/ocs/v1.php/cloud/users/Frank/subadmins -d groupid="group"`
- Makes the user Frank a subadmin of the group group

#### XML Output

```
<?xml version="1.0"?>
<ocs>
<meta>
<statuscode>100</statuscode>
<status>ok</status>
```

```
</meta>
<data/> </ocs>
```

## users / removesubadmin

Removes the subadmin rights for the user specified from the group specified. Authentication is done by sending a Basic HTTP Authorization header.

Syntax: ocs/v1.php/cloud/users/{userid}/subadmins

- HTTP method: DELETE
- DELETE argument: groupid, string - the group from which to remove the user's subadmin rights

Status codes:

- 100 - successful
- 101 - user does not exist
- 102 - user is not a subadmin of the group / group does not exist
- 103 - unknown failure

### Example

- `DELETE https://admin:secret@example.com/ocs/v1.php/cloud/users/Frank/subadmins -d groupid="oldgroup"`
- Removes Frank's subadmin rights from the oldgroup group

### XML Output

```
<?xml version="1.0"?>
<ocs>
<meta>
<statuscode>100</statuscode>
<status>ok</status>
</meta>
<data/> </ocs>
```

## users / getsubadmingroups

Returns the groups in which the user is a subadmin. Authentication is done by sending a Basic HTTP Authorization header.

Syntax: ocs/v1.php/cloud/users/{userid}/subadmins

- HTTP method: GET

Status codes:

- 100 - successful
- 101 - user does not exist
- 102 - unknown failure

### Example

- `GET https://admin:secret@example.com/ocs/v1.php/cloud/users/Frank/subadmins`
- Returns the groups of which Frank is a subadmin

### XML Output

```
<?xml version="1.0"?>
```

```
<ocs>
<meta>
<status>ok</status>
<statuscode>100</statuscode>
<message/>
</meta>
<data>
<element>testgroup</element>
</data>
</ocs>
```

## Instruction Set For Groups

### groups / getgroups

Retrieves a list of groups from the epiKshare server. Authentication is done by sending a Basic HTTP Authorization header.

Syntax: ocs/v1.php/cloud/groups

- HTTP method: GET
- url arguments: search - string, optional search string url arguments: limit - int, optional limit value

  url arguments: offset - int, optional offset value

Status codes:

- 100 - successful

#### Example

- `GET http://admin:secret@example.com/ocs/v1.php/cloud/groups?search=adm`
- Returns list of groups matching the search string.

#### XML Output

```
<?xml version="1.0"?>
<ocs>
<meta>
<statuscode>100</statuscode>
<status>ok</status>
</meta>
<data>
<groups>
<element>admin</element>
</groups>
</data> </ocs>
```

### groups / addgroup

Adds a new group. Authentication is done by sending a Basic HTTP Authorization header.

Syntax: ocs/v1.php/cloud/groups

- HTTP method: POST
- POST argument: groupid, string - the new groups name

Status codes:

- 100 - successful
- 101 - invalid input data
- 102 - group already exists
- 103 - failed to add the group

#### Example

- `POST http://admin:secret@example.com/ocs/v1.php/cloud/groups -d groupid="newgroup"`
- Adds a new group called newgroup

#### XML Output

```
<?xml version="1.0"?>
<ocs>
<meta>
<statuscode>100</statuscode>
<status>ok</status>
</meta>
<data/> </ocs>
```

### groups / getgroup

Retrieves a list of group members. Authentication is done by sending a Basic HTTP Authorization header.

Syntax: ocs/v1.php/cloud/groups/{groupid}

- HTTP method: GET

Status codes:

- 100 - successful

#### Example

- `POST http://admin:secret@example.com/ocs/v1.php/cloud/groups/admin`
- Returns a list of users in the admin group

#### XML Output

```
<?xml version="1.0"?>
<ocs>
<meta>
<statuscode>100</statuscode>
<status>ok</status>
</meta>
<data>
<users>
<element>Frank</element>
```

```
</users>
</data> </ocs>
```

## groups / getsubadmins

Returns subadmins of the group. Authentication is done by sending a Basic HTTP Authorization header.

Syntax: ocs/v1.php/cloud/groups/{groupid}/subadmins

- HTTP method: GET

Status codes:

1. - successful
2. - group does not exist

- 102 - unknown failure

### Example

- `GET https://admin:secret@example.com/ocs/v1.php/cloud/groups/mygroup/subadmins`
- Return the subadmins of the group: mygroup

### XML Output

```
<?xml version="1.0"?>
<ocs>
<meta>
<status>ok</status>
<statuscode>100</statuscode>
<message/>
</meta>
<data>
<element>Tom</element>
</data> </ocs>
```

## groups / deletegroup

Removes a group. Authentication is done by sending a Basic HTTP Authorization header.

Syntax: ocs/v1.php/cloud/groups/{groupid}

- HTTP method: DELETE

Status codes:

- 100 - successful
- 101 - group does not exist
- 102 - failed to delete group

### Example

- `DELETE http://admin:secret@example.com/ocs/v1.php/cloud/groups/mygroup`
- Delete the group mygroup

### XML Output

```
<?xml version="1.0"?>
<ocs>
<data>
<meta>
<statuscode>100</statuscode>
<status>ok</status>
</meta>
<data/>
</ocs>
```

## Instruction Set For Apps

### apps / getapps

Returns a list of apps installed on the epiKshare server. Authentication is done by sending a Basic HTTP Authorization header.

Syntax: ocs/v1.php/cloud/apps/

- HTTP method: GET
- url argument: filter, string - optional (enabled or disabled)

Status codes:

- 100 - successful
- 101 - invalid input data

#### Example

- `GET http://admin:secret@example.com/ocs/v1.php/cloud/apps?filter=enabled`
- Gets enabled apps

#### XML Output

```
<?xml version="1.0"?>
<ocs>
<meta>
<statuscode>100</statuscode>
<status>ok</status>
</meta>
<data>
<apps>
<element>files</element>
<element>provisioning_api</element>
</apps>
</data> </ocs>
```

### apps / getappinfo

Provides information on a specific application. Authentication is done by sending a Basic HTTP Authorization header. Syntax: ocs/v1.php/cloud/apps/{appid}

- HTTP method: GET

Status codes:

- 100 - successful

### Example

- `GET http://admin:secret@example.com/ocs/v1.php/cloud/apps/files`
- Get app info for the files app

### XML Output

```
<?xml version="1.0"?>
<ocs>
<meta>
<statuscode>100</statuscode>
<status>ok</status>
</meta>
<data>
<info/>
<remote>
<files>appinfo/remote.php</files>
<webdav>appinfo/remote.php</webdav>
<filesync>appinfo/filesync.php</filesync>
</remote>
<public/>
<id>files</id>
<name>Files</name>
<description>File Management</description>
<licence>AGPL</licence>
<author>Robin Appelman</author>
<require>4.9</require>
<shipped>true</shipped>
<standalone></standalone>
<default_enable></default_enable>
<types>
<element>filesystem</element>
</types>
</data> </ocs>
```

## apps / enable

Enable an app. Authentication is done by sending a Basic HTTP Authorization header.

Syntax: ocs/v1.php/cloud/apps/{appid}

- HTTP method: POST

Status codes:

- 100 - successful

#### Example

- `POST http://admin:secret@example.com/ocs/v1.php/cloud/apps/files_texteditor`
- Enable the files_texteditor app

#### XML Output

```
<?xml version="1.0"?>
<ocs>
<meta>
<statuscode>100</statuscode>
<status>ok</status>
</meta> </ocs>
```

### apps / disable

Disables the specified app. Authentication is done by sending a Basic HTTP Authorization header.

Syntax: ocs/v1.php/cloud/apps/{appid}

- HTTP method: DELETE

Status codes:

- 100 - successful

#### Example

- `DELETE http://admin:secret@example.com/ocs/v1.php/cloud/apps/files_texteditor`
- Disable the files_texteditor app

#### XML Output

```
<?xml version="1.0"?>
<ocs>
<meta>
<statuscode>100</statuscode>
<status>ok</status>
</meta>
</ocs>
```

## File Sharing and Management

epiKshare users can share files with their epiKshare groups and other users on the same epiKshare server, with epiKshare users on other epiKshare servers, and create public shares for people who are not epiKshare users. You have control of a number of user permissions on file shares:

- Allow users to share files
- Allow users to create public shares
- Require a password on public shares

- Allow public uploads to public shares
- Require an expiration date on public share links
- Allow resharing
- Restrict sharing to group members only
- Allow email notifications of new public shares
- Exclude groups from creating shares

Configure your sharing policy on your Admin page in the Sharing section.

- Check Allow apps to use the Share API to enable users to share files. If this is not checked, no users can create file shares.
- Check Allow users to share via link to enable creating public shares for people who are not epiKshare users via hyperlink.
- Check Enforce password protection to force users to set a password on all public share links. This does not apply to local user and group shares.
- Check Allow public uploads to allow anyone to upload files to public shares.
- Check Allow users to send mail notification for shared files to enable sending notifications from epiKshare. (Your epiKshare server must be configured to send mail) • Check Set default expiration date to set a default expiration date on public shares.
- Check Allow resharing to enable users to re-share files shared with them.
- Check Restrict users to only share with users in their groups to confine sharing within group memberships.

> This setting does not apply to the Federated Cloud sharing feature. If Federated Cloud Sharing is enabled, users can still share items with any users on any instances (including the one they are on) via a remote share.

- Check Allow users to send mail notification for shared files enables users to send an email notification to every epiKshare user that the file is shared with.
- Check Exclude groups from sharing to prevent members of specific groups from creating any file shares in those groups. When you check this, you'll get a dropdown list of all your groups to choose from. Members of excluded groups can still receive shares, but not create any
- Check Allow username autocompletion in share dialog to enable auto-completion of epiKshare usernames.

> epiKshare does not preserve the mtime (modification time) of directories, though it does update the mtimes on files.

## File Sharing

### Creating Persistent File Shares

When a user is deleted, their files are also deleted. As you can imagine, this is a problem if they created file shares that need to be preserved, because these disappear as well. In epiKshare files are tied to their owners, so whatever happens to the file owner also happens to the files.

One solution is to create persistent shares for your users. You can retain ownership of them, or you could create a special user for the purpose of establishing permanent file shares. Simply create a shared folder in the usual way, and share it with the users or groups who need to use it. Set the appropriate permissions on it, and then no matter which users come and go, the file shares will remain. Because all files added to the share, or edited in it, automatically become owned by the owner of the share regardless of who adds or edits them.

## Configuring Federated Cloud Sharing

With just a few clicks you can easily and securely link file shares between epiKshare servers, in effect creating a cloud of epiKshares. You can automatically send an email notification when you create the share, share directly with users on other epiKshare servers, add password protection, allow users to upload files, and set an expiration date.

> Currently, Federated shares cannot be re-shared, and the only permissions option when you create the share is “Can edit”.

### Creating a Direct Share Link

Follow these steps to create a new Federated Cloud share:

1. Go to your epiKshare Admin page and scroll to the Federated Cloud Sharing section of the Sharing section.
2. Check Allow other users on this server to send shares to other servers and Allow users on this server to receive shares from other servers. Leaving the checkboxes blank disables Federated Cloud sharing.

3. In the Sharing section, check Allow users to share via link and Allow users to send mail notification for shared files.
4. Now you and your users can go to your Files pages to create a new federated cloud share. Click the Share icon onthe file or directory you want to share to expose your first sharing option.

This dialog allows you to create local shares with users and groups on your local epiKshare server, and also to create federated cloud shares with users on remote epiKshare servers by typing a link to the remote server in the form of <user>@<link-to-epiKshare>. In this screenshot the remote epiKshare server is on the local network, so the URL form is user@hostname/epiKshare, or layla@remote-server/epiKshare in the example. The URL you type is echoed by the form, and labeled as (remote).

Press the return key, and then wait for the link to be established. You'll see a status message while it is working.

When the remote server has been successully contacted you'll see a confirmation.

The link is created when your remote user confirms the share by clicking the Add remote share button.

> You can return to the share dialog any time to see a list of everyone you have shared with, and federated cloud shares are labeled as (remote).

Click the trash can icon to disconnect the share.

### Creating Federated Cloud Shares via Public Link Share

Check the Share Link checkbox to expose more sharing options (which are described more fully in File Sharing). You may create a federated cloud share by allowing epiKshare to create a public link for you, and then email it to the person you want to create the share with.

You may optionally set a password and expiration date on it. When your recipient receives your email they must click the link, or copy it to a Web browser. They will see a page displaying a thumbnail of the file, with a button to Add to your epiKshare.

Your recipient should click the Add to your epiKshare button. On the next screen your recipient needs to enter the URL to their epiKshare server, and then press the return key.

Your recipient has to take one more step, and that is to confirm creating the federated cloud share link by clicking the Add remote share button.

Un-check the Share Link checkbox to disable any federated cloud share created this way.

### Configuration Tips

The Sharing section on your Admin page allows you to control how your users manage federated cloud shares:

- Check Enforce password protection to require passwords on link shares.
- Check Set default expiration date to require an expiration date on link shares.
- Check Allow public uploads to allow two-way file sharing.

Your Apache Web server must have mod_rewrite enabled, and you must have trusted_domains correctly configured to allow external connections. Consider also enabling SSL to encrypt all traffic between your servers .

Your epiKshare server creates the share link from the URL that you used to log into the server, so make sure that you log into your server using a URL that is accessible to your users. For example, if you log in via its LAN IP address, such as http://192.168.10.50, then your share URL will be something like http://192.168.10.50/epiKshare/index.php/s/jWfCfTVztGlWTJe, which is not accessible outside of your LAN. This also applies to using the server name; for access outside of your LAN you need to use a fully qualified domain name such as http://myserver.example.com, rather than http://myserver.

## Uploading big files > 512MB

The default maximum file size for uploads is 512MB. You can increase this limit up to what your filesystem and operating system allows. There are certain hard limits that cannot be exceeded:

- < 2GB on 32Bit OS-architecture
- < 2GB on Windows (32Bit and 64Bit)
- < 2GB with Server Version 4.5 or older
- < 2GB with IE6 - IE8
- < 4GB with IE9 - IE11

64-bit filesystems have much higher limits; consult the documentation for your filesystem.

> The epiKshare sync client is not affected by these upload limits as it is uploading files in smaller chunks.

### System Configuration

- Disable user quotas, which makes them unlimited
- Your temp file or partition has to be big enough to hold multiple parallel uploads from multiple users; e.g. if the max upload size is 10GB and the average number of users uploading at the same time is 100: temp space has to hold at least 10x100 GB

### Configuring Your Webserver

> epiKshare comes with its own epiKshare/.htaccess file. Because php-fpm can't read PHP settings in .htaccess these settings must be set in the epiKshare/.user.ini file. Please contact an epiKshare tech representative for assistance.

### Configuring PHP

If you don't want to use the epiKshare .htaccess or .user.ini file, you may configure PHP instead. Make sure to comment out any lines .htaccess pertaining to upload size, if you entered any.

If you are running epiKshare on a 32-bit system, any open_basedir directive in your php.ini file needs to be commented out.

Please contact the epiKshare support to get assistance with this advanced setting.

### Configuring upload limits within the GUI

If all prerequisites described in this documentation are in place an admin can change the upload limits on demand by using the File handling input box within the administrative backend of epiKshare.

Depending on your environment you might get an insufficient permissions message shown for this input box.

> Setting Strong Directory Permissions might prevent write access to these files. As an admin you need to decide between the ability to use the input box and a more secure epiKshare installation where you need to manually modify the upload limits in the .htaccess and .user.ini files described above.

## Configuring the Collaborative Documents App

The Documents application supports editing documents within epiKshare, without the need to launch an external application. The Documents app supports these features:

- Cooperative edit, with multiple users editing files simultaneously.
- Document creation within epiKshare.
- Document upload.
- Share and edit files in the browser, and then share them inside epiKshare or through a public link.

Supported file formats are *.odt*, *.doc*, and *.docx*. *.odt* is supported natively in epiKshare, and you must have LibreOffice or OpenOffice installed on the epiKshare server to convert *.doc*, and *.docx* documents.

## Providing Default Files

You may distribute a set of default files and folders to all users by placing them in the epiKshare/core/skeleton directory on your epiKshare server. These files appear only to new users after their initial login, and existing users will not see files that are added to this directory after their first login. The files in the skeleton directory are copied into the users' data directories, so they may change and delete the files without affecting the originals.They appear on the user's epiKshare Files page just like any other files.

## Configuring External Storage (GUI)

The External Storage Support application enables you to mount external storage services and devices as secondary epiKshare storage devices. You may also allow users to mount their own external storage services.

## Enabling External Storage Support

The External storage support application is enabled on your Apps page.

## Storage Configuration

To create a new external storage mount, select an available backend from the dropdown Add storage. Each backend has different required options, which are configured in the configuration fields.

Each backend may also accept multiple authentication methods. These are selected with the dropdown under Authentication. Different backends support different authentication mechanisms; some specific to the backend, others are more generic.

When you select an authentication mechanism, the configuration fields change as appropriate for the mechanism. Some backends are not yet migrated to the new authentication mechanism system, and are displayed with a mechanism of Built-in. The SFTP backend, to give an example, supports both password-based authentication and public key authentication.

Required fields are marked with a red border. When all required fields are filled, the storage is automatically saved. A green dot next to the storage row indicates the storage is ready for use. A red or yellow icon indicates that epiKshare could not connect to the external storage, so you need to re-check your configuration and network availability.

## User and Group Permissions

A storage configured in a user's Personal settings is available only to the user that created it. A storage configured in the Admin settings is available to all users by default, and it can be restricted to specific users and groups in the Available for field.

## Mount Options

Hover your cursor to the right of any storage configuration to expose the settings button and trashcan. Click the trashcan to delete the mountpoint. The settings button allows you to configure each storage mount individually with the following options:

- Encryption
- Previews
- Filesystem check frequency (Never, Once per direct access, every time the filesystem is used)

## Using Self-Signed Certificates

When using self-signed certificates for external storage mounts the certificate must be imported into the personal settings of the user. Please contact the epiKshare support for more information.

## Available storage backends

The following backends are provided by the external storages app. Other apps may provide their own backends, which are not listed here.

### Amazon S3

To connect your Amazon S3 buckets to epiKshare, you will need:

- S3 access key
- S3 secret key
- Bucket name

In the Folder name field enter a local folder name for your S3 mountpoint. If this does not exist it will be created.

In the Available for field enter the users or groups who have permission to access your S3 mount.

The Enable SSL checkbox enables HTTPS connections; using HTTPS is always highly-recommended.

Optionally, you can override the hostname, port and region of your S3 server, which is required for non-Amazon servers such as Ceph Object Gateway.

Enable path style is usually not required (and is, in fact, incompatible with newer Amazon datacenters), but can be used with non-Amazon servers where the DNS infrastructure cannot be controlled. Ordinarily, requests will be made with http://bucket.hostname.domain/, but with path style enabled, requests are made with http://hostname.domain/bucket instead.

## Dropbox

While Dropbox supports the newer OAuth 2.0, epiKshare uses OAuth 1.0, so you can safely ignore any references to OAuth 2.0 in the Dropbox configuration.

Connecting Dropbox is a little more work because you have to create a Dropbox app. Log into the Dropbox Developers page and click App Console:

If you have not already created any Dropbox apps it will ask you to accept their terms and conditions. Then you are presented with the choice to create either a Drop-ins App or a Dropbox API App. Click Dropbox API App, and then check:

- Files and datastores.
- No – My app needs access to files already on Dropbox.
- All file types – My app needs access to a user's full Dropbox. Only supported via the CoreAPI.

Then enter whatever name you want for your app.

Now click the Create App button. Under Status, do not click Development (Apply for production status) because that is for apps that you want to release publicly.

Click Enable additional users to allow multiple epiKshare users to use your new Dropbox share.

Note your App key and App secret, which you will enter in the External Storage form on your epiKshare Admin page.

Your epiKshare configuration requires only the local mount name, the App Key and the App Secret, and which users or groups have access to the share.

You must be logged into Dropbox, and when epiKshare successfully verifies your connection Dropbox will ask for verification to connect to your Dropbox account. Click Allow, and you're done.

## FTP/FTPS

To connect to an FTP server, you will need:

- A folder name for your local mountpoint; the folder will be created if it does not exist
- The URL of the FTP server
- Port number (default: 21)
- FTP server username and password
- Remote Subfolder, the FTP directory to mount in epiKshare. epiKshare defaults to the root directory. If you specify a subfolder you must leave off the leading slash. For example, public_html/images

Your new mountpoint is available to all users by default, and you may restrict access by entering specific users or groups in the Available for field.

Optionally, epiKshare can use FTPS (FTP over SSL) by checking Secure ftps://. This requires additional configuration with your root certificate if the FTP server uses a self-signed certificate.

> Note: The external storage FTP/FTPS needs the allow_url_fopen PHP setting to be set to 1. When having connection problems be sure that it has not been set to 0 in your php.ini.

FTP uses the password authentication scheme.

## Google Drive

epiKshare uses OAuth 2.0 to connect to Google Drive. This requires configuration through Google to get an app ID and app secret, as epiKshare registers itself as an app.

All applications that access a Google API must be registered through the Google Cloud Console. Follow along carefully because the Google interface is a bit of a maze and it's easy to get lost.

If you already have a Google account, such as Groups, Drive, or Mail, you can use your existing login to log into the Google Cloud Console. After logging in click the Create Project button.

Give your project a name, and either accept the default Project ID or create your own, then click the Create button.

You'll be returned to your dashboard.

Google helpfully highlights your next step in blue, the Use Google APIs box. Make sure that your new project is selected, click on Use Google APIs , and it takes you to Google's APIs screen. There are many Google APIs; look for the Google Apps APIs and click Drive API.

- Drive API takes you to the API Manager overview. Click the blue Enable API button.
- Now you must create your credentials, so click on Go to credentials.
- For some reason Google warns us again that we need to create credentials. We will use 0Auth 2.0.
- Now we have to create a consent screen. This is the information in the screen Google shows you when you connect your new Google app to epiKshare the first time. Click Configure consent screen. Then fill in the required form fields. Your logo must be hosted, as you cannot upload it, so enter its URL. When you're finished click Save.
- The next screen that opens is Create Client ID. Check Web Application, then enter your app name. Authorized JavaScript Origins is your root domain, for example https://www.example.com, without a trailing slash. You need two Authorized Redirect URIs, and they must be in this form:

https://example.com/epiKshare/index.php/settings/personal https://example.com/epiKshare/index.php/settings/admin

- Replace https://example.com/epiKshare/ with your own epiKshare server URL, then click Create.
- Now Google reveals to you your Client ID and Client Secret. Click OK.

You can see these anytime in your Google console; just click on your app name to see complete information.

Now you have everything you need to mount your Google Drive in epiKshare.

Go to the External Storage section of your Admin page, create your new folder name, enter the Client ID and Client Secret, and click Grant Access. Your consent page appears when epiKshare makes a successful connection. Click Allow.

When you see the green light confirming a successful connection you're finished.

### Local

Local storages provide access to any directory on the epiKshare server. Since this is a significant security risk, Local storage can only be configured in the epiKshare admin settings. Non-admin users cannot create Local storage mounts.

Use this to mount any directory on your epiKshare server that is outside of your epiKshare data/ directory. This directory must be readable and writable by your HTTP server user.

In the Folder name field enter the folder name that you want to appear on your epiKshare Files page.

In the Configuration field enter the full filepath of the directory you want to mount.

In the Available for field enter the users or groups who have permission to access the mount. By default all users have access.

### OpenStack Object Storage

OpenStack Object Storage is used to connect to an OpenStack Swift server. Two authentication mechanisms are available: one is the generic OpenStack mechanism, and the other is used exclusively for Rackspace, a provider of object storage that uses the OpenStack Swift protocol.

The bucket will be created if it does not exist.

The OpenStack authentication mechanism uses the OpenStack Keystone v2 protocol, connecting to the server specified in the URL of Identity Endpoint field. You need your Username, Tenant name and Password.

The Rackspace authentication mechanism requires a Rackspace Username and API key.

It may be necessary to specify a Service name or Region. The timeout of HTTP requests is set in the Request timeout field, in seconds.

### epiKshare

An epiKshare storage is a specialized WebDAV storage, with optimizations for epiKshare-epiKshare communication. See the WebDAV document ation to learn how to configure an epiKshare external storage.

When filling in the URL field, use the path to the root of the epiKshare installation, rather than the path to the WebDAV endpoint. So, for a server at http://example.com/epiKshare, use http://example.com/epiKshare and not http://example.com/epiKshare/remote.php/webdav.

### SFTP

epiKshare's SFTP (FTP over an SSH tunnel) backend supports both password and public key authentication.

The Host field is required; a port can be specified as part of the Host field in the following format: hostname.domain:port. The default port is 22 (SSH).

For public key authentication, you can generate a public/private key pair from your SFTP with secret key login configuration.

After generating your keys, you need to copy your new public key to the destination server to .ssh/authorized_keys. epiKshare will then use its private key to authenticate to the SFTP server.

The default Remote Subfolder is the root directory (/) of the remote SFTP server, and you may enter any directory you wish.

### SMB/CIFS

epiKshare can connect to Windows file servers or other SMB-compatible servers with the SMB/CIFS backend.

> The SMB/CIFS backend requires smbclient to be installed on the epiKshare server. If you're not sure about what to do, please contact the epiKshare suppport.

You need the following information:

- Folder name for your local mountpoint.
- Host: The URL of the Samba server.
- Username: The username or domain/username used to login to the Samba server.
- Password: the password to login to the Samba server.
- Share: The share on the Samba server to mount.
- Remote Subfolder: The remote subfolder inside the Samba share to mount (optional, defaults to /). To assign the epiKshare logon username automatically to the subfolder, use $user instead of a particular subfolder name.
- And finally, the epiKshare users and groups who get access to the share.

Optionally, you can specify a Domain. This is useful in cases where the SMB server requires a domain and a username, and an advanced authentication mechanism like session credentials is used so that the username cannot be modified. This is concatenated with the username, so the backend gets domain\username

### WebDAV

Use this backend to mount a directory from any WebDAV server, or another epiKshare server.

You need the following information:

- Folder name: The name of your local mountpoint.
- The URL of the WebDAV or epiKshare server.
- Username and password for the remote server
- Secure https://: We always recommend https:// for security, though you can leave this unchecked for http://.

Optionally, a Remote Subfolder can be specified to change the destination directory. The default is to use the whole root.

> CPanel users should install Web Disk to enable WebDAV functionality.

> **Please note: A non-blocking or correctly configured SELinux setup is needed for these backends to work. Please refer to the *SELinux Configuration*.**

## Allow Users to Mount External Storage

Check Enable User External Storage to allow your users to mount their own external storage services, and check the backends you want to allow. Beware, as this allows a user to make potentially arbitrary connections to other services on your network!

### Adding Files to External Storages

We recommend configuring the background job Webcron or Cron to enable epiKshare to automatically detect files added to your external storages.

epiKshare may not always be able to find out what has been changed remotely (files changed without going through epiKshare), especially when

it’s very deep in the folder hierarchy of the external storage.

Please contact an epiKshare tech representative to schedule a trigger to rescan the user’s files periodically (for example every 15 minutes), which includes the mounted external storage.

## Configuration File

Storage mount configurations are stored in a JSON formatted file. Admin storages are stored in data/mount.json, while personal storages are stored in data/$user/mount.json. For more advanced use cases, including provisioning external storages from outside epiKshare, please contact the epiKshare support.

## Using self-signed certificates

When using self-signed certificates for external storage mounts the certificate needs to be imported in the personal settings of the user. Please contact us to get more information.

## Configuring Temporary Disk Space Needs

Not all external storage types are currently enabled for, or support streaming. Therefore epiKshare needs temporary space to buffer data for transfers. This can occur when there are many concurrent users transferring data with a higher volume over small bandwidth. epiKshare may need, in these cases, additional temporary space.

Example: 100 concurrent users uploading each a 300MB file with a total transfer time of 6000s (1h 40min). The temporary space needed by epiKshare for this period of time is 30GB. Even though it is not mandatory, the location of the temp directory used by epiKshare can be configured manually. If you need to do so, please contact the epiKshare support.

As of writing, following external storage list uses temp files for up/download:

- FTP
- SMB / SMB_OC
- WebDAV
- Amazon S3
- Dropbox
- Google Drive
- OpenStack SWIFT

External storage list that uses direct file streaming:

- Local
- SFTP

## External Storage Password Management

epiKshare handles passwords for external mounts differently than regular epiKshare user passwords.

The regular user and file share passwords (when you use the default epiKshare user backend) are stored using a strong cryptographically secure hashing mechanism in the database. On a new user account with a new password, the password is hashed and stored in the epiKshare database. The plain-text password is never stored. When the user logs in, the hash of the password they enter is compared with the hash in the database. When the hashes match the user is allowed access. These are not recoverable, so when a user loses a password the only option is to create a new password.

Passwords which are used to connect against external storage (e.g. SMB or FTP), there we have to differentiate again between different implementations:

- Login with epiKshare credentials

When a mountpoint has this option, for example SMB / CIFS using OC login, the password will be intercepted when a user logs in and written to the PHP session (which is a file on the filesystem), and written encrypted into the session with a key from the configuration file. Every time that password is required epiKshare reads it from the PHP session file.

When you use this option, features such as sharing will not work properly from that mountpoint when the user is not logged-in.

- Stored credentials

When you enter credentials into the files_external dialog those are stored on the filesystem and encrypted with a key stored in config.php. This is required since epiKshare needs access to those files and shares even when the user is not logged-in to have sharing and other key features properly working.

To sum up:

You can protect your PHP session files using protections available in your filesystem. Stored credentials are always accessible to the epiKshare instance.

### External Storage Authentication mechanisms

epiKshare storage backends accept one or more authentication schemes such as passwords, OAuth, or token-based, to name a few examples. Each authentication scheme may be implemented by multiple authentication mechanisms. Different mechanisms require different configuration parameters, depending on their behaviour.

#### Special Mechanisms

The None authentication mechanism requires no configuration parameters, and is used when a backend requires no authentication.

The Built-in authentication mechanism itself requires no configuration parameters, but is used as a placeholder for legacy storages that have not been migrated to the new system and do not take advantage of generic authentication mechanisms. The authentication parameters are provided directly by the backend.

#### Password-based Mechanisms

The Username and password mechanism requires a manually-defined username and password. These get passed directly to the backend.

The Session credentials mechanism uses the epiKshare login credentials of the user to connect to the storage. These are not stored anywhere on the server, but rather in the user session, giving increased security. The drawback is that sharing is disabled when this mechanism is in use, as epiKshare has no access to the storage credentials and so other users cannot use it.

#### Public-key Mechanisms

Currently only the RSA mechanism is implemented, where a public/private keypair is generated by epiKshare and the public half shown in the GUI. The keys are generated in the SSH format, and are currently 1024 bits in length. Keys can be regenerated with a button in the GUI.

#### OAuth

OAuth 1.0 and OAuth 2.0 are both implemented, but currently limited to the Dropbox and Google Drive backends respectively. These mechanisms require additional configuration at the service provider, where an app ID and app secret are provided and then entered into epiKshare. Then epiKshare can perform an authentication request, establishing the storage connection.

## Encryption Configuration

The primary purpose of the epiKshare server-side encryption is to protect users' files on remote storage, such as Dropbox and Google Drive, and to do it easily and seamlessly from within epiKshare.

In epiKshare 8.2 the server-side encryption has a number of changes and improvements, including:

- An option to create a master encryption key, which replaces all individual user keys. This is especially useful for single-sign on.
- Encrypt all data files at once when enabling encryption.
- Decrypt all data files, or per user.
- Users may decrypt their own files.
- Move your keys to a different folder.

epiKshare server-side encryption encrypts files stored on the epiKshare server, and files on remote storage that is connected to your epiKshare server. Encryption and decryption are performed on the epiKshare server. All files sent to remote storage will be encrypted by the epiKshare server, and upon retrieval, decrypted before serving them to you and anyone you have shared them with.

> Encrypting files increases their size by roughly 35%, so you must take this into account when you are provisioning storage and setting storage quotas. User's quotas are based on the unencrypted file size, and not the encrypted file size.

When files on external storage are encrypted in epiKshare, you cannot share them directly from the external storage services, but only through epiKshare sharing because the key to decrypt the data never leaves the epiKshare server.

epiKshare's server-side encryption generates a strong encryption key, which is unlocked by user's passwords. Your users don't need to track an extra password, but simply log in as they normally do. It encrypts only the contents of files, and not filenames and directory structures.

You should regularly backup all encryption keys to prevent permanent data loss. Please contact epiKshare for more information.

When encryption is enabled, all files are encrypted and decrypted by the epiKshare application, and stored encrypted on your remote storage. This protects your data on externally hosted storage. The epiKshare admin and the storage admin will see only encrypted files when browsing backend storage.

Warning: Encryption keys are stored only on the epiKshare server, eliminating exposure of your data to thirdparty storage providers. The encryption app does not protect your data if your epiKshare server is compromised, and it does not prevent epiKshare administrators from reading user's files. This would require client-side encryption, which this app does not provide. If your epiKshare server is not connected to any external storage services then it is better to use other encryption tools, such as file-level or whole-disk encryption.

Note also that SSL terminates at or before Apache on the epiKshare server, and all files will exist in an unencrypted state between the SSL connection termination and the epiKshare code that encrypts and decrypts files. This is also potentially exploitable by anyone with administrator access to your server.

## Before Enabling Encryption

> **Warning**
> Plan very carefully before enabling encryption because it is not reversible via the epiKshare Web interface. If you lose your encryption keys your files are not recoverable. Always have backups of your encryption keys stored in a safe location, and consider enabling all recovery options.

## Enabling Encryption

epiKshare encryption consists of two parts. The base encryption system is enabled and disabled on your Admin page. First you must enable this, and then select an encryption module to load. Currently the only available encryption module is the epiKshare Default Encryption Module.

First go to the Server-side encryption section of your Admin page and check Enable server-side encryption. You have one last chance to change your mind.

After clicking the Enable Encryption button you see the message "No encryption module loaded, please load a encryption module in the app menu", so go to your Apps page to enable the epiKshare Default Encryption Module.

Return to your Admin page to see the epiKshare Default Encryption Module added to the module selector, and automatically selected. Now you must log out and then log back in to initialize your encryption keys.

## Sharing Encrypted Files

After encryption is enabled your users must also log out and log back in to generate their personal encryption keys. They will see a yellow warning banner that says "Encryption App is enabled but your keys are not initialized, please log-out and log-in again."

Share owners may need to re-share files after encryption is enabled; users trying to access the share will see a message advising them to ask the share owner to re-share the file with them. For individual shares, un-share and re-share the file. For group shares, share with any individuals who can't access the share. This updates the encryption, and then the share owner can remove the individual shares.

## Encrypting External Mountpoints

You and your users can encrypt individual external mountpoints. You must have external storage enabled on your Admin page, and enabled for your users.

## Enabling Users File Recovery Keys

If you lose your epiKshare password, then you lose access to your encrypted files. If one of your users loses their epiKshare password their files are unrecoverable. You cannot reset their password in the normal way; you'll see a yellow banner warning "Please provide an admin recovery password, otherwise all user data will be lost".

To avoid all this, create a Recovery Key. Go to the Encryption section of your Admin page and set a recovery key password.

Then your users have the option of enabling password recovery on their Personal pages. If they do not do this, then the Recovery Key won't work for them.

For users who have enabled password recovery, give them a new password and recover access to their encrypted files by supplying the

Recovery Key on the Users page.

You may change your Recovery Key password.

### Files Not Encrypted

Only the data in the files in data/user/files are encrypted, and not the filenames or folder structures. These files are never encrypted:

- Existing files in the trash bin & Versions. Only new and changed files after encryption is enabled are encrypted.
- Existing files in Versions
- Image thumbnails from the Gallery app
- Previews from the Files app
- The search index from the full text search app
- Third-party app data

There may be other files that are not encrypted; only files that are exposed to third-party storage providers are guaranteed to be encrypted.

### Encryption with LDAP and Other External User Back-ends

If you use an external user back-end, such as an LDAP or Samba server, and you change a user's password on the back-end, the user will be prompted to change their epiKshare login to match on their next epiKshare login. The user will need both their old and new passwords to do this. If you have enabled the Recovery Key then you can change a user's password in the epiKshare Users panel to match their back-end password, and then, of course, notify the user and give them their new password.

## Transactional File Locking

epiKshare's Transactional File Locking mechanism locks files to avoid file corruption during normal operation. It performs these functions:

- Operates at a higher level than the filesystem, so you don't need to use a filesystem that supports locking
- Locks parent directories so they cannot be renamed during any activity on files inside the directories
- Releases locks after file transactions are interrupted, for example when a sync client loses the connection during an upload
- Manages locking and releasing locks correctly on shared files during changes from multiple users
- Manages locks correctly on external storage mounts
- Manages encrypted files correctly

What Transactional File locking is not for: it is not for preventing collisions in collaborative document editing, nor will it prevent multiple users from editing the same document, or give notice that other users are working on the same document. Multiple users can open and edit a file at the same time and Transactional File locking does not prevent this. Rather, it prevents simultaneous file saving.

When you see the warning on your epiKshare admin page "Transactional file locking is using the database as locking backend, for best performance it's advised to configure a memcache for locking", you are not required to use a memcache. File locking is enabled by default, using the database locking backend. This places a significant load on your database. Using memcache.locking relieves the database load and improves performance. Admins of epiKshare servers with heavy workloads should install a memcache.

To use a memcache with Transactional File Locking, you must install the Redis server and corresponding PHP module. Please contact the epiKshare support for assistance.

## Mimetype aliases

epiKshare allows administrators to specify aliases for mimetypes. This makes it possible to show more specific icons for certain mimetypes. Take as an example audio files. It is nicer if they show a nice audio icon instead of the default file icon.

By default epiKshare is distributed with config/mimetypealiases.dist.json. Administrators should not modify this file, as it will be replaced when epiKshare is updated. Please contact our tech team to learn more about introducing more mimetypes.

## Troubleshooting

### How can I find out if my MySQL/PostgreSQL server is reachable?

To check the server's network availability, use the ping command on the server's host name (db.server.com in this example):

ping db.server.dom

PING db.server.dom (ip-address) 56(84) bytes of data.

64 bytes from your-server.local.lan (192.168.1.10): icmp_req=1 ttl=64 time=3.64 ms

64 bytes from your-server.local.lan (192.168.1.10): icmp_req=2 ttl=64 time=0.055 ms

64 bytes from your-server.local.lan (192.168.1.10): icmp_req=3 ttl=64 time=0.062 ms

For a more detailed check whether the access to the database server software itself works correctly, see the next question.

# Operations

Advanced operation including monitoring and scaling across multiple machines.

## Considerations on Monitoring

Large scale epiKshare deployments are typically installed as load balanced n-tier web applications. Successfully managing such an installation requires active monitoring of the application and supporting infrastructure components. The purpose of this section is to outline the components of epiKshare that need to be monitored, and provide guidance on what to look for in epiKshare in an enterprise installation.

## epiKshare Deployment Architecture

Before discussing how to monitor epiKshare, it is important to understand the architecture of a typical epiKshare deployment. These monitoring best practices are developed based on the use of load balanced web servers, a clustered database running a distributed database storage engine, such as MySQL NDB, and a clustered filesystem, such as Red Hat Storage.

It is assumed that specific enterprise tools (monitoring, log management, etc) to monitor operations are available, and that epiKshare is simply a new target for these tools.

## The Important Components of epiKshare

epiKshare is a PHP application that depends on a filesystem for file storage, and a database for storing user and file meta data, as well as some application specific information. While the loss of an app server or a node in the database or storage clusters should not bring the system down, knowing that this happened and resolving it is essential to keeping the service running efficiently. Therefore it is important to monitor the epiKshare servers, the Load Balancer, the Storage Cluster and the Database. This documentation starts with the epiKshare application and works out from there through the layers of infrastructure.

## Status.php

epiKshare provides a very simple mechanism for determining if an application server is up and functioning – call the status.php file on each epiKshare server. This file can be found in the root epiKshare directory on the server, which by default is /epiKshare/status.php. If the server is functioning normally, the response looks something like this:

{"installed":"true","version":"6.0.0.16","versionstring":"6.0.1","edition":""}

We recommend monitoring this file on each epiKshare application server to provide a basic check that the server is operating properly.

## epiKshare.log

epiKshare also provides a built in logging function. If the epiKshare Enterprise Edition logging applications are enabled, this file will track user logins and shared file activity. If these logging applications are not enabled, this log file still tracks basic epiKshare health. Given the potential for this file to get quite large, the log file should be rotated on a daily basis, and given the importance of the error information in the log file, this should be integrated with an enterprise log manager.

Logfile entries that start with the keyword “Error” should be logged and reported to epiKshare support.

## Apache

The apache error and access log should also be monitored. Significant spontaneous changes of the number of requests per second should also

be monitored and looked into.

### Database server

The load and general health of the database server or cluster has to be monitored also. All mysql vendors provide tools to monitor this.

### Clustered Filesystem

The available space of the filesystem should be monitored to prevent a full epiKshare. This functionality is provided by the operating-system and/or the cluster filesystem vendor.

### Load Balancer

The load balancer is monitoring the health of the application servers and is distributing the traffic in the optimal way. The application-servers should also be monitored to detect long lasting OS or hardware problems. Monitoring solutions like Nagios provide built in functionality to do this.

## Bugs

If you think you have found a bug in epiKshare, please:

- Search for a solution (see the options above)
- Double-check your configuration

> If you can’t find a solution, please contact our epiKshare support team.

## General Troubleshooting

### epiKshare Logfiles

In a standard epiKshare installation the log level is set to Normal. To find any issues you need to raise the log level to Everything on your epiKshare Admin page. Please see *Logging Configuration* for more information on these log levels.

For JavaScript issues you will also need to view the javascript console. All major browsers have developer tools for viewing the console, and you usually access them by pressing F12. For Firefox we recommend to installing the Firebug extension.

### Debugging Sync Issues

> **Warning**
> The data directory on the server is exclusive to epiKshare and must not be modified manually.

Disregarding this can lead to unwanted behaviours like:

- Problems with sync clients
- Undetected changes due to caching in the database

If you need to directly upload files from the same server please use a WebDAV command line client like cadaver to upload files to the WebDAV interface at:

> https://example.org/epiKshare/remote.php/webdav

### Common problems / error messages

Some common problems / error messages found in your logfiles as described above:

- SQLSTATE[HY000] [1040] Too many connections -> You need to increase the connection limit of your database, please refer to the manual of your database for more information.
- SQLSTATE[HY000]: General error: 5 database is locked -> You’re using SQLite which can’t handle a lot of parallel requests. Please

consider converting to another database like described in *Converting Database Type*.

- SQLSTATE[HY000]: General error: 2006 MySQL server has gone away -> The database request takes too long and therefore the MySQL server times out. Its also possible that the server is dropping a packet that is too large. Please refer to the manual of your database for how to raise the config options wait_timeout and/or max_allowed_packet.
- SQLSTATE[HY000] [2002] No such file or directory -> There is a problem accessing your SQLite database file in your data directory (data/epiKshare.db). Please check the permissions of this folder/file or if it exists at all. If you're using MySQL please start your database.
- Connection closed / Operation cancelled -> This could be caused by wrong KeepAlive settings within your Apache config. Make sure that KeepAlive is set to On and also try to raise the limits of KeepAliveTimeout and MaxKeepAliveRequests.
- No basic authentication headers were found -> There is (most probably) a mistake in the configuration. Please contact the epiKshare support for assistance.

### Other issues

Some services like *Cloudflare* can cause issues by minimizing JavaScript and loading it only when needed. When having issues like a not working login button or creating new users make sure to disable such services first.

## License Keys

### Introduction

You'll need to install a license key to use epiKshare 8. There are two types of license keys: one is a free 30-day trial key. The other is a full license key for customers.

### Configuration

Once you get your license key, it needs to be installed via the Admin section of your epiKshare installation. Please login as an administrator and install the license in the License seciton of the Admin panel.

## Configuring SharePoint Integration

Native SharePoint support has been added to the epiKshare as a secondary storage location for SharePoint 2007, 2010 and 2013. When this is enabled, users can access and sync all of their SharePoint content via epiKshare, whether in the desktop sync, mobile or Web interfaces. Updated files are bi-directionally synced automatically. SharePoint shares are created by the epiKshare admin, and optionally by any users who have SharePoint credentials.

The epiKshare SharePoint plugin uses SharePoint document lists as remote storage folders. epiKshare respects SharePoint access control lists (ACLs), so epiKshare sharing is intentionally disabled for SharePoint mountpoints. This is to preserve SharePoint ACLs and ensure content is properly accessed as per SharePoint rules.

The plugin uses the Simple Object Access Protocol (SOAP) and WebDAV for the uploads and downloads to talk to SharePoint servers.

The supported authentication methods are:

- Basic Auth
- NTLM (Recomended)

### Enabling the SharePoint Plugin

The SharePoint plugin is a native plugin, so the first step is to enter the Apps administration page and enable it.

Next, enter the Admin panel to set up SharePoint connections in the SharePoint Drive Configuration section.

- First, enter your SharePoint Listing credentials. These credentials are not stored in the database, but are used only during plugin setup to list the Document Libraries available per SharePoint site.
- Global credentials is an optional field. If you fill in this field, these credentials will be used on on all

  SharePoint mounts where you select: "Use global credentials" as the Authentication credentials

- Enter your epiKshare mount point in the Local Folder Name column. This is the name of the folder that each user will see on the epiKshare filesystem. You may use an existing folder, or enter a name to create a new mount point
- Select who will have access to this mountpoint, by default "All users", or a user or a group
- Enter your SharePoint server URL
- Then click the little refresh icon to the left of the Document Library field. If your credentials and URL are correct you'll get a dropdown list of available SharePoint libraries
- Select the document library you want to mount

- Select which kind of Authentication credentials you want to use for this mountpoint. If you select use custom credentials, you will have to enter the the credentials on this line. Otherwise, the global credentials or the user's own credentials will be used
- Click Save, and you're done

Please see Connecting to SharePoint in the User Manual to learn how to use your new SharePoint connections.

### Note

> Speed up load times by disabling file previews in config.php, because the previews are generated by downloading the remote files to a temp file. This means epiKshare will spend a lot of time creating previews for all of your SharePoint content. To disable file previews, please inform the epiKshare support to have the setting changed accordingly.

### Troubleshooting

SharePoint unsharing is handled in the background via Cron. If you remove the sharing option from a Sharepoint mount, it will take a little time for the share to be removed, until the Cron job runs.

Global mount points can't be accessed: You have to fill out your SharePoint credentials as User on the personal settings page, or in the popup menu. These credentials are used to mount all global mount points.

Personal mount points can't be accessed: You have to fill your SharePoint credentials as User on the personal settings page in case your personal mount point doesn't have its own credentials.

A user can't update the credentials: Verify that the correct credentials are configured, and the correct type, either global or custom.

## Installing and Configuring the Windows Network Drive App

The Windows Network Drive app creates a control panel on your Admin page for seamless mounting of SMB/CIFS file shares on epiKshare servers.

Any Windows file share, and Samba servers on Linux and other Unix-type operating systems use the SMB/CIFS file-sharing protocol. The files and directories on the SMB/CIFS server will be visible on your Files page just like your other epiKshare files and folders. They are labeled with a little four-pane Windows-style icon, and the left pane of your Files page includes a Windows Network Drive filter. Figure 1 shows a new Windows Network Drive share marked with red warnings. This indicates that epiKshare cannot connect to the share because it is not available, or there is an error in the configuration.

Files are synchronized bi-directionally, and you can create, upload, and delete files and folders. epiKshare server admins can create Windows Network Drive mounts, and optionally allow users to create their own personal Windows Network Drive mounts. The password for each mount is encrypted and stored in the epiKshare database, using a long random secret key stored in the epiKshare config. This allows epiKshare to access the shares when the users who own the mounts are not logged in.

### Installation

Enable the Windows Network Drive app on your epiKshare Apps page.

### Creating a New Share

When you create a new SMB share you need the login credentials for the share, the server address, the share name, and the folder you want to connect to.

1. First enter the epiKshare mountpoint for your new SMB share. This must not be an existing folder.
2. Then enter which epiKshare users or groups get access to the share. The default is all users.
3. Next, enter the address of the server that contains the SMB share.
4. Then the Windows share name.
5. Then the root folder of the share. This is the folder name, or the $user variable for user's home directories.

   Note that the LDAP Internal Username Attribute must be set to the samaccountname for either the share or the root to work, and the user's home directory needs to match the samaccountname.

1. Then your login credentials.

You have four options for login credentials:

- User credentials.
- Global credentials, which uses the credentials set in the Global credentials fields
- Login credentials is for users to connect to the mountpoint using their DOMAIN/logincredentials; enter the domain in the Domain field.
- Custom Credentials

When you're finished click the Save button.

> When you create a new mountpoint using Login credentials you must log out of epiKshare, and then log back in so you can access the share. You only have to do this the first time.

### Personal SMB Mounts

Users create their own personal SMB mounts on their Personal pages. These are created the same way as Admincreated shares. Users have only two options for login credentials:

- Personal Credentials.
- Custom Credentials

## User Management

### Shibboleth Integration

#### Introduction

The epiKshare Shibboleth user backend application integrates epiKshare with a Shibboleth Service Provider (SP) and allows operations in federated and single-sign-on (SSO) infrastructures. Setting up Shibboleth has two big steps:

1. Enable and configure the Apache Shibboleth module.
2. Enable and configure the epiKshare Shibboleth app.

#### The Apache Shibboleth module

Please contact the epiKshare support to get assistance with activating the Apache Shibboleth module and integrating the SSO authentification.

#### WebDAV Support

Users of standard WebDAV clients can use an alternative WebDAV Url, for example https://cloud.example.com/remote.php/nonshib-webdav/ to log in with their username and password. The password is generated on the Personal settings page.

#### Known Limitations

##### Encryption

File encryption can only be used together with Shibboleth when the *master key-based encryption* is used because the per- user encryption requires the user's password to unlock the private encryption key. Due to the nature of Shibboleth the user's password is not known to the service provider.

##### Other Login Mechanisms

You can allow other login mechanisms (e.g. LDAP or epiKshare native) by creating a second Apache virtual host configuration. This second location is not protected by Shibboleth, and you can use your other epiKshare login mechanisms.

### Session Timeout

Session timeout on Shibboleth is controlled by the IdP. It is not possible to have a session length longer than the length controlled by the IdP. In extreme cases this could result in re-login on mobile clients and desktop clients every hour.

The session timeout can be overridden in the service provider, but this requires a source code change of the Apache Shibboleth module. A patch can be provided by the epiKshare support team.

### UID Considerations and Windows Network Drive compatability

When using user_shibboleth in Single sign-on only mode, together with user_ldap, both apps need to resolve to the same uid. user_shibboleth will do the authentication, and user_ldap will provide user details such as email and displayname. In the case of Active Directory, multiple attributes can be used as the uid. But they all have different implications to take into account:

sAMAccountName

- *Example:* jfd
- *Uniqueness:* Domain local, might change e.g. marriage
- *Other implications:* Works with windows_network_drive app

userPrincipalName

- *Example:* jfd@epiKshare.com
- *Uniqueness:* Forest local, might change on eg. marriage
- *Other implications:* TODO check WND compatability

objectSid

- *Example:* S-1-5-21-2611707862-2219215769-354220275-1137
- *Uniqueness:* Domain local, changes when the user is moved to a new domain
- *Other implications:* Incompatible with windows_network_drive app

sIDHistory

- *Example:* Multi-value
- *Uniqueness:* Contains previous objectSIDs
- *Other implications:* Incompatible with windows_network_drive app

objectGUID

- *Example:* 47AB881D-0655-414D-982F-02998C905A28
- *Uniqueness:* Globally unique
- *Other implications:* Incompatible with windows_network_drive app

Keep in mind that epiKshare will derive the home folder from the uid, unless a home folder naming rule is in place. The only truly stable attribute is the objectGUID, so that should be used. If not for the uid then at least as the home folder naming rule. The tradeoff here is that if you want to use windows_network_drive you are bound to the sAMAccountName, as that is used as the login.

Also be aware that using user_shibboleth in Autoprovision Users mode will not allow you to use SSO for additional user_ldap users, because uid collisions will be detected by user_ldap.

## Enabling Anonymous Uploads with Files Drop

The Files Drop application allows anyone to upload files with the click of a button to the directory of your choosing, without needing a login, and they cannot see or change the contents of the directory. It is the perfect replacement for attaching large files to email, maintaining an FTP server, and commercial file-sharing services.

When files are uploaded to your Files Drop directory, you can manage them just like any other epiKshare share: you may share them, restrict access, edit, and delete them.

### Setting Up the Files Drop App

Setting up Files Drop is a matter of a few clicks. First go to your Apps page and enable it.

Now your users will see a configuration section on their Personal pages.

Click the Choose button to open a dialog to select your upload directory. You may wish to first create a special upload directory (on your Files page), which in the following example is name upload.

On your Personal page you should now see a URL for your upload directory. Share this URL with anyone you want to allow uploads to your File Drop folder. Note that the maximum upload size in this example is 512MB. (The default epiKshare upload file size limit is 512MB, but it can be changed. Please refer to the upload file size limit configuration section.)

### Using the Files Drop App

Uploading files via the Files Drop app is simple. Open your Web browser to the share URL created by epiKshare. Click the Click to upload file button. This opens a file picker, and you select the file or directory you want to upload.

When your upload is completed, you'll see a confirmation message with the filenames.

## Enterprise Logging

There are two enterprise logging apps available to epiKshare customers: Log file sharing and Log user actions. The Log file sharing app records the file sharing activity of your users, and Log user actions records user logins and logouts.

These two apps work together, and should be enabled together. Your logging level must be set to at least Info, warnings, errors, and fatal issues on your epiKshare admin page.

View your logfiles on your admin page. This shows which logging app recorded the entries, timestamps, usernames, and their activities:

Click the Download logfile button to dump the plain text log, or open the logfile directly in a text editor.

## Related articles

- epiKshare User
- Deploying an epiKshare server on-premise
- epiKshare Administrator
- epiKshare Administration Guide
- First Steps