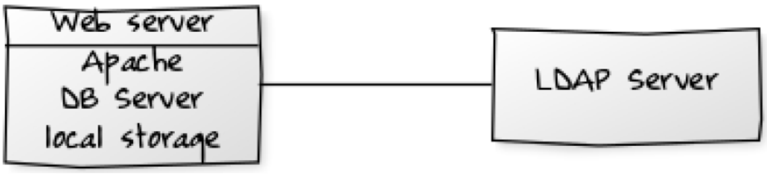
# ownCloud Deployment Recommendations

What is the best way to install and maintain ownCloud? The answer to that is *“it depends”* because every ownCloud customer has their own particular needs and IT infrastructure. ownCloud and the LAMP stack are highly-configurable, 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 ownCloud will only grow. Plan ahead.*

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

- **Operating system:** Linux.
- **Webserver:** Apache 2.4.
- **Database:** MySQL/MariaDB.
- **PHP 5.5+:** PHP 5.4 is the minimum supported version; note that it reached end-of-life in September 2015 and is no longer supported by the PHP team. Some Linux vendors, such as Red Hat, still support PHP 5.4. 5.6+ is recommended. mod_php is the recommended Apache module because it provides the best performance.

## 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 filesystems: nightly backups with service interruption.

**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.
- **Operating system:** Enterprise-grade Linux distribution with full support from OS vendor. We recommend Red Hat Enterprise Linux or SUSE Linux Enterprise Server 12.
- **SSL Configuration:** The SSL termination is done in Apache. A standard SSL certificate is needed, installed according to the Apache documentation.
- **Load Balancer:** None.
- **Database:** MySQL, MariaDB or PostgreSQL. We currently recommend MySQL / MariaDB, as our customers have had good experiences when moving to a Galera cluster to scale the DB.
- **Backup:** Install owncloud, ownCloud data directory and database on Btrfs filesystem. Make regular snapshots at desired intervals for zero downtime backups. Mount DB partitions with the “nodatacow” option to prevent fragmentation.

Alternatively, make nightly backups with service interruption:

- Shut down Apache.
- Create database dump.
- Push data directory to backup.
- Push database dump to backup.
- Start Apache.

Then optionally rsync to a backup storage or tape backup. (See the Maintenance section of the Administration manual for tips on backups and restores.)

- **Authentication:** User authentication via one or several LDAP or Active Directory servers. (See User Authentication with LDAP for information on configuring ownCloud 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-specific session storage location. You can find out where that is by running

  ```
  grep -R ’session.save_path‘ /
  etc/php5
  ```

  and then add it to the

  `/etc/fstab` file, for example:

  ```
  echo "tmpfs /var/lib/php5/
  pool-www tmpfs
  defaults,noatime,mode=1777 0
  0" >> /etc/fstab.
  ```

- **Memory Caching:** A memcache speeds up server performance, and ownCloud supports four memcaches; refer to Configuring Memory Caching for information on selecting and configuring a memcache.
- **Storage:** Local storage.
- **ownCloud Edition:** Standard Edition. (See ownCloud Server or Enterprise Edition for comparisons of the ownCloud editions.)

## Small Workgroups or Departments

- **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. A cluster of two database servers. Storage on an NFS server. Authentication via an existing LDAP or Active Directory server.

- **Components:**
  - 2 to 4 application servers with 4 sockets and 32GB RAM.
  - 2 DB servers with 4 sockets and 64GB RAM.
  - 1 HAproxy load balancer with 2 sockets and 16GB RAM.
  - NFS storage server as needed.
- **Operating system:** Enterprise grade Linux distribution with full support from OS vendor. Red Hat Enterprise Linux or SUSE Linux Enterprise Server 12 are recommended.
- **SSL Configuration:** The SSL termination is done in the HAProxy load balancer. A standard SSL certificate is needed, installed according to the HAProxy documentation.
- **Load Balancer:** 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.
- **Database:** MySQL/MariaDB Galera cluster with master-master replication.
- **Backup:** Minimum daily backup without downtime. All MySQL/MariaDB statements should be replicated to a backup MySQL/MariaDB slave instance.
  - Create a snapshot on the NFS storage server.
  - At the same time stop the MySQL replication.
  - Create a MySQL dump of the backup slave.
  - Push the NFS snapshot to the backup.
  - Push the MySQL dump to the backup.
  - Delete the NFS snapshot.
  - Restart MySQL replication.
- **Authentication:** User authentication via one or several LDAP or Active Directory servers. (See User Authentication with LDAP for information on configuring ownCloud to use LDAP and AD.)
- **LDAP:** Read-only slaves should be deployed on every application server for optimal scalability
- **Session Management:** Session management on the application server. PHP sessions are stored in a tmpfs mounted at the operating system-specific session storage location. You can find out where that is by running

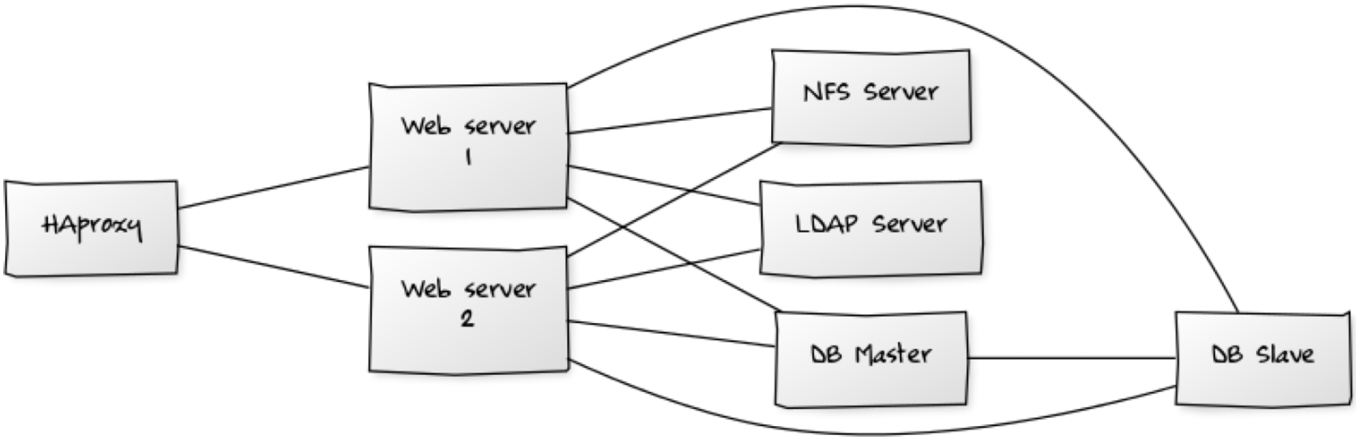
  ```
  grep -R ’session.save_path‘ /
  etc/php5
  ```

  and then add it to the

  `/etc/fstab` file, for example:

  ```
  echo "tmpfs /var/lib/php5/
  pool-www tmpfs
  defaults,noatime,mode=1777 0
  0" >> /etc/fstab.
  ```

- **Memory Caching:** A memcache speeds up server performance, and ownCloud supports four memcaches; refer to Configuring Memory Caching for information on selecting and configuring a memcache.
- **Storage:** Use an off-the-shelf NFS solution, such as IBM Elastic Storage or RedHat Ceph.
- **ownCloud Edition:** Enterprise Edition. (See ownCloud Server or Enterprise Edition for comparisons of the ownCloud editions.)



## 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.

- **Components:**
  - 4 to 20 application servers with 4 sockets and 64GB RAM.
  - 4 DB servers with 4 sockets and 128GB RAM
  - 2 Hardware load balancer, for example BIG IP from F5
  - NFS storage server as needed.

- **Operating system:** RHEL 7 with latest service packs.

- **SSL Configuration:** The SSL termination is done in the load balancer. A standard SSL certificate is needed, installed according to the load balancer documentation.

- **Load Balancer:** A redundant hardware load-balancer with heartbeat, for example F5 Big-IP. This runs two load balancers in front of the application servers.

- **Database:** MySQL/MariaDB Galera Cluster with 4x master – master replication.

- **Backup:** Minimum daily backup without downtime. All MySQL/MariaDB statements should be replicated to a backup MySQL/MariaDB slave instance.

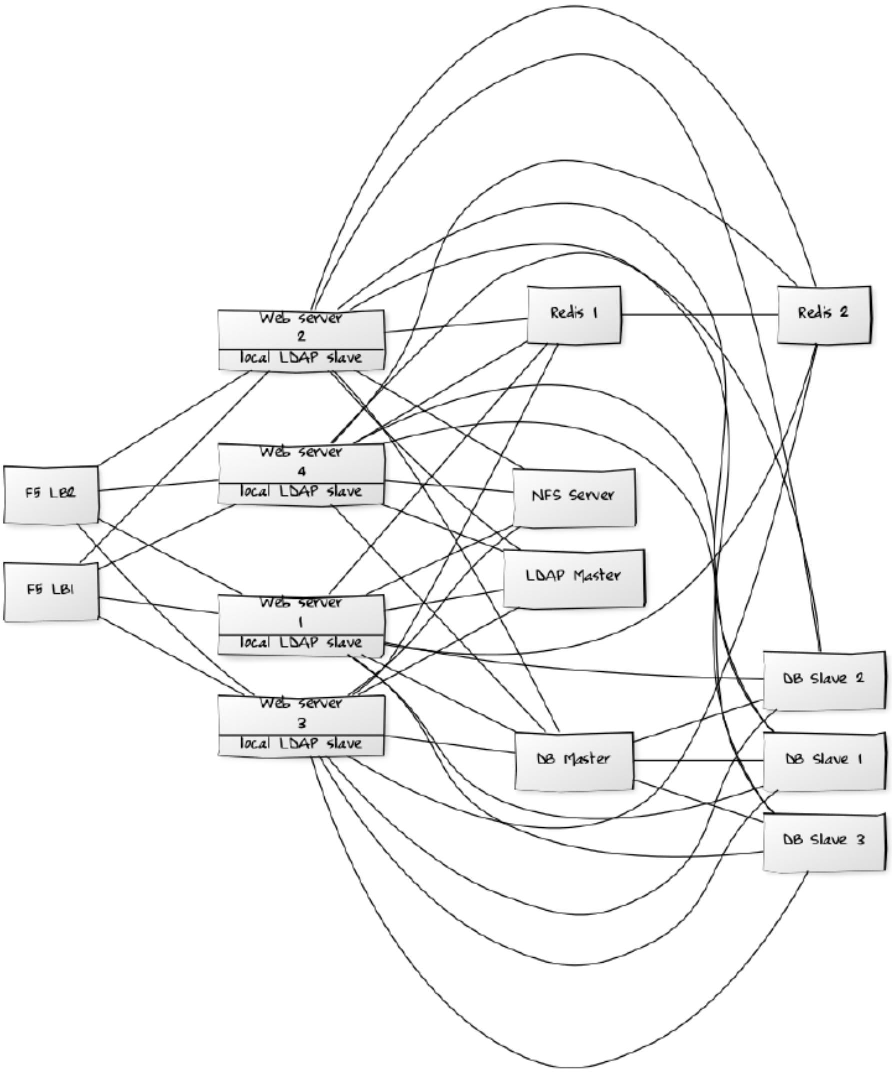
  - Create a snapshot on the NFS storage server.
  - At the same time stop the MySQL replication.
  - Create a MySQL dump of the backup slave.
  - Push the NFS snapshot to the backup.
  - Push the MySQL dump to the backup.
  - Delete the NFS snapshot.
  - Restart MySQL replication.

- **Authentication:** User authentication via one or several LDAP or Active Directory servers, or SAML/Shibboleth. (See User Authentication with LDAP and Shibboleth Integration.)

- **LDAP:** Read-only slaves should be deployed on every application server for optimal scalability.

- **Session Management:** Redis should be used for the session management storage.
- **Caching:** Redis for distributed in-memory caching (see Configuring Memory Caching).

- **Storage:** An off-the-shelf NFS solution should be used. Examples are IBM Elastic Storage or RedHAT Ceph. Optionally, an S3 compatible object store can also be used.
- **ownCloud Edition:** Enterprise Edition. (See ownCloud Server or Enterprise Edition for comparisons of the ownCloud editions.)


Web server 2 local LDAP slave
Redis 1
Redis 2
Web server 4 local LDAP slave
F5 LB2
NFS Server
F5 LB1
LDAP Master
Web server 1 local LDAP slave
DB Slave 2
Web server 3 local LDAP slave
DB Master
DB Slave 1
DB Slave 3

## 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 traffic 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 ownCloud 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

Provider setup:

- DNS round robin to HAProxy servers (2-n, SSL offloading, 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 benefits 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 filecache 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 file 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: Multimaster has the risk of split brain, and deadlocks. ownCloud tries to solve the problem of deadlocks with high-level file locking.

## Software Considerations

### Operating System

We are dependent on distributions that offer an easy way to install the various components in up-to-date versions. ownCloud has a partnership with RedHat and SUSE for customers who need commercial support. Canonical, the parent company of Ubuntu Linux, also offers enterprise service and support. Debian and Ubuntu are free of cost, and include newer software packages. CentOS is the community-supported free-of-cost Red Hat Enterprise Linux clone. openSUSE is community-supported, and includes many of the same system administration tools as SUSE Linux Enterprise Server.

### Relational Database

More often than not the customer already has an opinion on what database to use. In general, the recommendation is to use what their database administrator is most familiar with. Taking into account what we are seeing at customer deployments, we recommend MySQL/MariaDB in a master-slave deployment with a MySQL proxy in front of them to send updates to master, and selects to the slave(s). The second best option is PostgreSQL (alter table does not lock table, which makes migration less painful) although we have yet to find a customer who uses a master-slave setup.

What about the other DBMS?

- Sqlite is adequate for simple testing, and for low-load single-user deployments. It is not adequate for production systems.
- Microsoft SQL Server is not a supported option.
- Oracle DB is the de facto standard at large enterprises and is fully supported with ownCloud Enterprise Edition only.

## File Storage

Our main use case is up- and download of files. 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 ownCloud high-level file locking.
- If Shibboleth is a requirement you must use Memcached, and it can also be used to scale-out shibd session storage (see Memcache StorageService).

## Storage

- Database High Availability
- Performance enhancements for Apache and PHP
- How to Set Up a Redis Server as a Session Handler for PHP on Ubuntu 14.04

**ownCloud GmbH**
Leipziger Platz 21
90491 Nürnberg
Germany

www.owncloud.com/contact
phone: +49 911 14888690

**www.owncloud.com**

@ownCloud
facebook.com/owncloud
gplus.is/owncloud
linkedin.com/company/owncloud