

Automated installation with autoconfig.php



- [Parameters](#)
 - [Encrypted database connection](#)
 - [Examples](#)
 - [Data Directory](#)
 - [SQLite database](#)
 - [MySQL / MariaDB database](#)
 - [PostgreSQL database](#)
 - [Encrypted MySQL / MariaDB connection](#)
 - [Encrypted PostgreSQL connection](#)
 - [Complete non-interactive setup](#)
-

Use `config/autoconfig.php` to prefill, or fully automate, the web-based initial setup of Nextcloud. This is useful when installing multiple identical or similar instances.

Create `config/autoconfig.php` and include the installation parameters that you want to prefill. Any values not provided in the file can be completed in the setup form when you first launch Nextcloud.

To complete the installation without interaction, provide a valid database configuration, data directory, and both `adminlogin` and `adminpass`. Otherwise, Nextcloud displays the setup form so that missing or invalid values can be corrected.

Nextcloud removes `config/autoconfig.php` after a successful installation. The file remains in place if installation fails.



The supplied database account must be able to connect to the database server and have sufficient privileges for the selected configuration. Depending on the database backend and the account's privileges, Nextcloud may create the

database and a dedicated database user. Alternatively, create the database and database user in advance, and grant the user the required privileges, as described in [Database configuration](#).

“

`autoconfig.php` commonly contains plaintext database and administrator passwords. Restrict access to the file before writing secrets to it. If installation fails, remove the file or rotate any exposed secrets.

Parameters

The following parameters are commonly used:

- `directory` (written to `config.php` as `datadirectory`)
- `dbtype`, `dbname`, `dbuser`, `dbpass`, `dbhost`, and `dbtableprefix`
- `adminlogin` and `adminpass`
- optionally, `trusted_domains` and `adminemail`
- optionally, the parameters of an [encrypted database connection](#)

Two parameters have different names from their corresponding `config.php` settings:

<code>autoconfig.php</code>	<code>config.php</code>
<code>directory</code>	<code>datadirectory</code>
<code>dbpass</code>	<code>dbpassword</code>

Encrypted database connection

“ *New in version 35*

An SSL/TLS encrypted connection to the database server can be configured during the installation. Use this when the database does not run on the same host as Nextcloud, so that the credentials and all queries are not sent in plaintext.

The following parameters are independent of the database backend. Nextcloud translates them into the corresponding `config.php` settings before the first connection is opened, so the installation itself already uses an encrypted connection.

Parameter	Supported by	Description
<code>dbsslmode</code>	PostgreSQL	Encryption mode of the connection, for example <code>require</code> or <code>verify-full</code> . See the PostgreSQL documentation for the available modes.
<code>dbsslca</code>	MySQL/MariaDB, PostgreSQL	Path to the CA certificate the database server is verified against.
<code>dbsslcert</code>	MySQL/MariaDB, PostgreSQL	Path to the client certificate used to authenticate against the database server.
<code>dbsslkey</code>	MySQL/MariaDB, PostgreSQL	Path to the private key belonging to the client certificate.
<code>dbsslcrll</code>	PostgreSQL	Path to the certificate revocation list.
<code>dbsslnoverify</code>	MySQL/MariaDB	Set to <code>true</code> to not verify that the certificate of the database server matches the hostname used to connect. MySQL and MariaDB verify this by default, PostgreSQL only in the <code>verify-full</code> mode.

Connection encryption parameters

Note the following restrictions:

- `dbsslcert` and `dbsslkey` have to be provided together.
- A parameter that the selected database does not support is rejected with an error instead of being ignored, and the installation does not proceed. SQLite and Oracle support none of them — an Oracle connection is encrypted through the connect string and `sqlnet.ora` instead.
- The certificates and keys have to be readable by the PHP process.

Alternatively, the backend-specific `dbdriveroptions` (MySQL/MariaDB) and `pgsql_ssl` (PostgreSQL) settings, as documented in [Configuration Parameters](#), can be written to `autoconfig.php` verbatim. They are passed to `config.php` as provided and are not validated. Values set through the parameters above take precedence over individual entries of these arrays.

Examples

The following examples show partial and complete automatic configurations.

Data Directory

This configuration prefills the data directory. Complete the database and administrator-account settings in the setup form.

```
<?php
$AUTOCONFIG = [
    "directory"    => "/www/htdocs/nextcloud/data",
];
```

SQLite database

This configuration prefills the SQLite database settings. Complete the data directory and administrator-account settings in the setup form.

```
<?php
$AUTOCONFIG = [
    "dbtype"       => "sqlite",
    "dbname"       => "nextcloud",
    "dbtableprefix" => "",
];
```

MySQL / MariaDB database

This configuration prefills the MySQL or MariaDB settings. Complete the data directory and administrator-account settings in the setup form.

```
<?php
$AUTOCONFIG = [
    "dbtype"       => "mysql",
    "dbname"       => "nextcloud",
    "dbuser"       => "username",
    "dbpass"       => "password",
    "dbhost"       => "localhost",
    "dbtableprefix" => "",
];
```

PostgreSQL database

This configuration prefills the PostgreSQL settings. Complete the data directory and administrator-account settings in the setup form.

```
<?php
$AUTOCONFIG = [
    "dbtype"      => "pgsql",
    "dbname"      => "nextcloud",
    "dbuser"      => "username",
    "dbpass"      => "password",
    "dbhost"      => "localhost",
    "dbtableprefix" => "",
];
```

Encrypted MySQL / MariaDB connection

This configuration connects to a remote MySQL or MariaDB server over an encrypted connection, verifying the server against a CA certificate and authenticating with a client certificate.

```
<?php
$AUTOCONFIG = [
    "dbtype"      => "mysql",
    "dbname"      => "nextcloud",
    "dbuser"      => "username",
    "dbpass"      => "password",
    "dbhost"      => "db.example.com",
    "dbtableprefix" => "",
    "dbsslca"     => "/etc/ssl/nextcloud/ca-cert.pem",
    "dbsslcert"   => "/etc/ssl/nextcloud/client-cert.pem",
    "dbsslkey"    => "/etc/ssl/nextcloud/client-key.pem",
];
```

Nextcloud stores these paths as `dbdriveroptions` in `config.php`.

Encrypted PostgreSQL connection

This configuration requires an encrypted connection to a remote PostgreSQL server and verifies both the certificate of the server and the hostname it was issued for.

```
<?php
$AUTOCONFIG = [
    "dbtype"      => "pgsql",
    "dbname"      => "nextcloud",
    "dbuser"      => "username",
    "dbpass"      => "password",
    "dbhost"      => "db.example.com",
    "dbtableprefix" => "",
    "dbsslmode"   => "verify-full",
    "dbsslca"     => "/etc/ssl/nextcloud/ca-cert.pem",
];
```

Nextcloud stores these values as `pgsql_ssl` in `config.php`.

Complete non-interactive setup

When all required values are present and valid, the installation proceeds without requiring user interaction.

The following configuration bypasses the setup form:

```
<?php
$AUTOCONFIG = [
    "dbtype"      => "mysql",
    "dbname"      => "nextcloud",
    "dbuser"      => "username",
    "dbpass"      => "password",
    "dbhost"      => "localhost",
    "dbtableprefix" => "",
    "adminlogin"   => "root",
    "adminpass"    => "root-password",
    "directory"    => "/www/htdocs/nextcloud/data",
];
```

📄 #3

📄 cocobook-sync 📄 2026-10-28 16:26:17 UTC

📄 cocobook-sync 📄 2026-10-29 01:09:13 UTC