Recommended Settings and Tips

- Recommended Settings
  - [database].DB_TYPE
  - [cache].ADAPTER
  - [repository.signing].DEFAULT_TRUST_MODEL
  - [security].LOGIN_REMEMBER_DAYS
  - [service].ENABLE_CAPTCHA
  - [ui].ONLY_SHOW_RELEVANT_REPOS
  - [service].USERNAME_COOLDOWN_PERIOD
- OpenSSH recommendations
  - Git's wire protocol v2
  - Per source penalty
  - AuthorizedKeys command
- Tips
  - Database performance

The default settings are applicable to a broad range of Forgejo instances. Nevertheless, not every instance has the same purpose and number of users. This document lists settings suggested for various situations, the results of such configuration changes, and their use cases.

Recommended Settings

In this document, configuration parameters are described as `[section.part].ENABLE_FEATURE_XY`, with `section` and `.part` indicating the section and subsection accordingly, and `ENABLE_FEATURE_XY` reflecting the key. As an example, here's how it would look like in the `app.ini` file:

```
[section.part]
ENABLE_FEATURE_XY = new_value
```

Please bear in mind that you don't have to create new sections for every value; you can have multiple settings in the same section, such as:

```
[section.part]
ENABLE_FEATURE_XY = new_value
DISABLE_FEATURE_Z = other_new_value
```

The recommended settings are sorted alphabetically rather than by importance.

[database].DB_TYPE

For every action, operation or page you visit, Forgejo must query a database for data. The database is an important part of the Forgejo stack and therefore must be configured with care to ensure proper operation. However, it is difficult to give a recommendation due to its general nature and the many factors that affect the use of the database. So keep the following text in mind when configuring and choosing the database and always test carefully if the configuration suits your situation and ask for assistance if needed.

If your instance is going to see low to moderate activity, it is recommended to change this value to **sqlite3**. SQLite3 is a simple database that doesn't require maintenance and exists as just one to three files on disk. It is by far the easiest database to configure and has many other advantages over the other databases, but it becomes a poor choice once you see a lot of concurrent activity in which case performance may decrease, but SQLite can go a long way in the early years of an instance.

If your instance will see high activity, it is recommended to change this value to **mysql** or **postgres**. There is no best managed database server, and it mainly depends on your previous experience and knowledge about the database you want to use, both can handle large amounts of activity, please refer to the database-configuration about what other values should be configured to connect to the database server.

> Keep in mind that transitioning to another database with an existing database is not a trivial task and must be done carefully.

[cache].ADAPTER

Forgejo uses caching to avoid repeating expensive operations. The default cache implementation is basic: items expire after a fixed duration (time-to-live) regardless of usage frequency, and it lacks eviction strategies based on memory pressure or access patterns (like LRU). This simplicity means it may not be optimal for all workloads. To illustrate the potential impact of caching, consider these common use cases:

- Calculating commit counts for repositories or branches (reducing potentially expensive Git operations).
- Retrieving specific file or commit history (reducing Git I/O).
- Fetching user settings (minimizing database lookups).
- Computing avatar hashes (saving CPU time).

If your instance does not see much activity, it is recommended to change this value to **twoqueue**. This will use a size-limited LRU cache, which will keep frequently used items and remove the ones that are not used often. It is also recommended to change `[cache].HOST` to `{"size":100, "recent_ratio":0.25, "ghost_ratio":0.5}` because the default value has a limit of 50,000 items and since by default it will keep items for 16 hours, it is not memory efficient to possibly keep so many items for a small instance.

If your instance sees a lot of activity, it is recommended to change this value to **redis** or **memcache**. In that case, caching is outsourced to third-party software designed to cache items. Keep in mind that you will probably also need to modify `[cache].HOST` to configure the use of this software.

The **redis** adapter should support most software that works with a limited subset of the Redis Go Client APIs. In particular, Forgejo has integration tests against:

- Redis v7.2; higher versions will not be officially supported due to licensing concerns,
- Redict, a fork of Redis v7.2 licensed under the Lesser GNU General Public license (`LGPL-3.0-only`),
- Valkey, a fork of Redis v7.2 stewarded by the Linux Foundation, and
- Garnet, an independently implemented cache-store using Redis's RESP.

`[repository.signing].DEFAULT_TRUST_MODEL`

When Forgejo needs to verify a GPG or SSH signed commit on a repository, it checks who it can trust to have a verified commit on that repository. The default value for this setting is that Forgejo trusts only contributors to that repository to have signed commits, this can cause unexpected behavior for those used to GitHub when hosting repositories that anyone can contribute to, because non-collaborators with signed commits on that repository are shown as unverified.

If your instance expects users to contribute to other repositories within your instance, it is recommended that you change this value to **committer**. In that case, signed commits will appear

as verified even if they are not collaborators.

`[security].LOGIN_REMEMBER_DAYS`

When a user logs in with the remember option enabled, they receive a long-term authentication cookie that is remembered for a number of days, determined by this setting. The default value is one month, which may be less than other services and could be frustrating for the user to log in every month.

If your instance does not need to adhere to a security policy that mandates a different value, it is recommended that you change this value to **365**. In that case, the long-term authentication cookie will be stored on the user's device for one year.

`[service].ENABLE_CAPTCHA`

In recent years, spam bots have gotten smarter and have automated more than ever. A simple register form such as in Forgejo is a simple task for bots to register a new user and comment wherever they can about their favorite scam website. Forgejo provides the option to enable captcha on the register form.

If your instance has an open registration, it is recommended to change this value to **true**. You also need to choose which captcha you want to use, you can find more about that in the service configuration. Keep in mind that certain types of captcha limit the accessibility of registering new users because they often use images that are difficult to see.

`[ui].ONLY_SHOW_RELEVANT_REPOS`

The explore page is a good way to quickly check what is happening with an instance because it shows the most recent repositories on which action has been taken. If your instance sees a lot of activity, such as with a large user base, this explore page can quickly become a cluttered place where most of the repositories may not be relevant, such as a test repository or someone's personal dotconfig repository, which does not necessarily add to the effect of seeing what is happening on that instance.

If you have an instance with a lot of activity, it is recommended to change this value to **true**. In that situation, the explore page will filter out repositories that are likely to be less relevant in order to see what interesting projects are available on that instance.

`[service].USERNAME_COOLDOWN_PERIOD`

If a user or an organization is renamed, each old username, until it is claimed, will redirect to the latest username. In the context of public forges, to prevent namesquatting and allow users to

reverse their username change within a certain period of time, a cooldown period can be configured to ensure that no user can claim the username within the cooldown period (with the exception of the original user and administrators).

If your instance has an open registration, it is recommended to change this value to **7**. This means that there is a seven-day cooldown period for old usernames before they can be claimed by anyone, and that users can revert to their old username within seven days.

There is a closely related configuration option `[service].MAX_USER_REDIRECTS` for which the default value is **5** when a cooldown period is specified. This controls the number of user redirects a user can have (and therefore how many usernames are subject to a cooldown period). The intention of this option is to prevent abuse of the cooldown feature by limiting the amount of usernames that one user can keep on cooldown.

OpenSSH recommendations

This section only applies if you've enabled SSH (enabled by default), and are using sshd provided by OpenSSH on the host. Using OpenSSH is Forgejo's recommended practice. Forgejo also provides a built-in SSH server (`[ssh].START_SSH_SERVER` set to true) where these OpenSSH recommendations would not be applicable.

We recommend that you scope these settings in the sshd_config for the `git` user (or another user if it's set differently in `[ssh].SSH_USER`). All the recommendations below match on the `git` user.

Git's wire protocol v2

Git will tell the server over SSH that it is able to use a newer version of the Git Wire Protocol, but by default, SSH servers like OpenSSH will not pass the environment to Forgejo.

You have to tell the SSH server to pass the `GIT_PROTOCOL` environment, which can be configured as follows:

```
Match User git
  AcceptEnv GIT_PROTOCOL
```

Per source penalty

This only applies if you reverse proxy SSH connections, and thus sshd do not get the original client IP.

Since OpenSSH 9.8, sshd will penalize sources (IP addresses) if they do something other than authenticating correctly. Thus, in the case that sshd only ever sees the IP address of your proxy software, then after enough penalties it will block all incoming SSH connections. This is undesirable as legitimate users are now penalized by the bad behavior of malicious users, as such we recommend to disable the feature altogether as follows:

```
Match User git
  PerSourcePenalties no
```

If you still want to have this feature while using a reverse proxy, you have two options. sshd doesn't support the PROXY protocol, so you either you have to use mmproxy[^1] or transparent proxying if your proxy software supports that. Both require changes in the firewall setup and transparent proxy additionally requires kernel parameters changes, which might be undesirable for your use case.

[^1]: Cloudflare introduced the concept of mmproxy, but we recommend go-mmproxy and mmproxy-rs as implementation over the implementation by Cloudflare.

AuthorizedKeys command

This is mostly applicable if you administer a Forgejo instance for many users, as this solves a scaling problem.

The default configuration is to create a `authorized_keys` file in the `git` user's `.ssh` directory, which contains all public SSH keys that users have uploaded to their account. sshd reads the entire file[^2] when verifying if the public key is accepted, this in turn can lead to many `read` syscalls if the `authorized_keys` file is of a large size.

[^2]: Yes, it really consumes the whole file. https://github.com/openssh/openssh-portable/blob/f44f124a234785b9d2cd59a6da28737fb3c4ad9a/auth2-pubkeyfile.c#L433-L435

Forgejo can provide the authorized keys on-demand to sshd, by configuring `AuthorizedKeysCommand`. If you compile `forgejo` from source, it should be noted that sshd requires the binary to be owned by root and not not writable by groups and others. Furthermore you must specify the full path to the forgejo binary.

> [!info] Info The path might differ for your install of Forgejo, ensure you've got a matching path.

```
Match User git
  AuthorizedKeysCommandUser git
```

```
AuthorizedKeysCommand /usr/local/bin/forgejo --config /etc/forgejo/app.ini keys -u %u -t %t -k
%k
```

As of writing, you will run into another scaling problem. The whole config is parsed on each execution, depending on your config this might take a few seconds.

Tips

Database performance

When trying to understand poor database performance in conjunction with Forgejo, it is useful to set `[database].SLOW_QUERY_THRESHOLD` to a value lower than the default (5 seconds). This ensures that SQL queries that are slow, but not too slow to meet the default value, are logged and can provide insight into what kind of SQL queries are slow, moreover, this is useful information for Forgejo contributors to find the code where this SQL query is executed and understand the context.

□□□ #2
□ □□□ □□□ 2026-10-31 13:34:09 UTC
□ □□□ □□□ 2026-10-31 13:34:10 UTC