Authentication

- LDAP via BindDN
- LDAP using simple auth
- Verify group membership in LDAP
- PAM (Pluggable Authentication Module)
- FreeIPA

Both the LDAP via BindDN and the simple auth LDAP share the following fields:

- Authorization Name **(required)**
  - A name to assign to the new method of authorization.
- Host **(required)**
  - The address where the LDAP server can be reached.
  - Example: `example.com`
- Port **(required)**
  - The port to use when connecting to the server.
  - Example: `389` for LDAP or `636` for LDAP SSL
- Enable TLS Encryption (optional)
  - Whether to use TLS when connecting to the LDAP server.
- Admin Filter (optional)
  - An LDAP filter specifying if a user should be given administrator privileges. If a user account passes the filter, the user will be privileged as an administrator. In order to select a subset of the LDAP users, this filter is applied to the result set of the User Filter.
  - Example: `(objectClass=adminAccount)`
  - Example for Microsoft Active Directory (AD): `(memberOf=CN=admin-group,OU=example,DC=example,DC=org)`
- Username attribute (optional)
  - The attribute of the user's LDAP record containing the user name. Given attribute value will be used for the new Forgejo account username after the first successful sign-in. Leave empty to use the login name given on the sign-in form.
  - This is useful when the supplied login name is matched against multiple attributes, but only a single specific attribute should be used for the Forgejo account name; see "User Filter".
  - Example: `uid`
  - Example for Microsoft Active Directory (AD): `sAMAccountName`
- First name attribute (optional)

    - The attribute of the user's LDAP record containing the user's first name. This will be used to populate their account information.
    - Example: `givenName`
- Surname attribute (optional)
    - The attribute of the user's LDAP record containing the user's surname. This will be used to populate their account information.
    - Example: `sn`
- E-mail attribute **(required)**
    - The attribute of the user's LDAP record containing the user's email address. This will be used to populate their account information.
    - Example: `mail`

LDAP via BindDN

Adds the following fields:

- Bind DN (optional)
    - The DN to bind to the LDAP server with when searching for the user. This may be left blank to perform an anonymous search.
    - Example: `cn=Search,dc=example,dc=com`
- Bind Password (optional)
    - The password for the Bind DN specified above, if any. *Note: The password is stored encrypted with the SECRET_KEY on the server. It is still recommended to ensure that the Bind DN has as few privileges as possible.*
- User Search Base **(required)**
    - The LDAP base at which user accounts will be searched for.
    - Example: `ou=Users,dc=example,dc=com`
- User Filter **(required)**
    - An LDAP filter declaring how to find the user record that is attempting to authenticate. The `%[1]s` matching parameter will be substituted with the login name given on the sign-in form.
    - Example: `(&(objectClass=posixAccount)(|(uid=%[1]s)(mail=%[1]s)))`
    - Example for Microsoft Active Directory (AD): `(&(objectCategory=Person)(memberOf=CN=user-group,OU=example,DC=example,DC=org)(sAMAccountName=%s)(!(UserAccountControl:1.2.840.113556.1.4.803:=2)))`
    - To substitute more than once, `%[1]s` should be used instead, e.g., when matching the supplied login name against multiple attributes such as a user identifier, email, or even phone number.
    - Example: `(&(objectClass=Person)(|(uid=%[1]s)(mail=%[1]s)(mobile=%[1]s)))`
- Enable user synchronization
    - This option enables a periodic task that synchronizes the Forgejo users with the LDAP server. The default period is every 24 hours, but that can be changed in the app.ini file. See the `[cron.sync_external_users]` section in the sample app.ini for

detailed comments about that section. The *User Search Base* and *User Filter* settings described above will limit which users can use Forgejo and which users will be synchronized. When initially run, the task will create all LDAP users that match the given settings, so take care if working with large Enterprise LDAP directories.

LDAP using simple auth

Adds the following fields:

- User DN **(required)**
  - A template to use as the user's DN. The `%s` matching parameter will be substituted with the login name given on the sign-in form.
  - Example: `cn=%s,ou=Users,dc=example,dc=com`
  - Example: `uid=%s,ou=Users,dc=example,dc=com`
- User Search Base (optional)
  - The LDAP base at which user accounts will be searched for.
  - Example: `ou=Users,dc=example,dc=com`
- User Filter **(required)**
  - An LDAP filter declaring when a user should be allowed to log in. The `%s` matching parameter will be substituted with the login name given on the sign-in form.
  - Example: `(&(objectClass=posixAccount)(|(cn=%[1]s)(mail=%[1]s)))`
  - Example: `(&(objectClass=posixAccount)(|(uid=%[1]s)(mail=%[1]s)))`

Verify group membership in LDAP

Uses the following fields:

- Group Search Base (optional)
  - The LDAP DN used for groups.
  - Example: `ou=group,dc=example,dc=com`
- Group Name Filter (optional)
  - An LDAP filter declaring how to find valid groups in the above DN.
  - Example: `(|(cn=forgejo_users)(cn=admins))`
- User Attribute in Group (optional)
  - The user attribute that is used to reference a user in the group object.
  - Example: `uid` if the group object contains a `member: bender` and the user object contains a `uid: bender`.
  - Example: `dn` if the group object contains a `member: uid=bender,ou=users,dc=planetexpress,dc=com`.
- Group Attribute for User (optional)
  - The attribute of the group object that lists/contains the group members.
  - Example: `memberUid` or `member`

PAM (Pluggable Authentication Module)

This procedure enables PAM authentication. Users may still be added to the system manually using the user administration. PAM provides a mechanism to automatically add users to the current database by testing them against PAM authentication. To work with normal Linux passwords, the user running Forgejo must also have read access to `/etc/shadow` in order to check the validity of the account when logging in using a public key.

> [!note] Note If a user has added SSH public keys into Forgejo, the use of these keys *may* bypass the login check system. Therefore, if you wish to disable a user who authenticates with PAM, you *should* also manually disable the account in Forgejo using the built-in user manager.

1. Configure and prepare the installation.
   - It is recommended that you create an administrative user.
   - Deselecting automatic sign-up may also be desired.
2. Once the database has been initialized, log in as the newly created administrative user.
3. Navigate to the user settings (icon in the top-right corner), and select `Site Administration` -> `Authentication Sources`, and select `Add Authentication Source`.
4. Fill out the fields as follows:
   - `Authentication Type`: `PAM`
   - `Name`: Any value should be valid here; use "System Authentication" if you'd like.
   - `PAM Service Name`: Select the appropriate file listed under `/etc/pam.d/` that performs the authentication desired.[^1]
   - `PAM Email Domain`: The e-mail suffix to append to user authentication. For example, if the login system expects a user called `gituser`, and this field is set to `mail.com`, then Forgejo will expect the `user email` field for an authenticated GIT instance to be `gituser@mail.com`.[^2]

> [!note] Note PAM support is added via build-time flags (`TAGS="pam" make build`), and the official binaries provided do not have this enabled. PAM requires that the necessary `libpam` dynamic library be available and the necessary PAM development headers be accessible to the compiler.

[^1]: For example, using standard Linux log-in on Debian "Bullseye" use `common-session-noninteractive` - this value may be valid for other flavors of Debian including Ubuntu and Mint; consult your distribution's documentation.

[^2]: **This is a required field for PAM**. Be aware: In the above example, the user will log into the Forgejo web interface as `gituser` and not `gituser@mail.com`.

FreeIPA

- In order to log in to Forgejo using FreeIPA credentials, a bind account needs to be created for Forgejo:
- On the FreeIPA server, create a `forgejo.ldif` file, replacing `dc=example,dc=com` with your DN, and provide an appropriately secure password:

```
dn: uid=forgejo,cn=sysaccounts,cn=etc,dc=example,dc=com
changetype: add
objectclass: account
objectclass: simplesecurityobject
uid: forgejo
userPassword: secure password
passwordExpirationTime: 20380119031407Z
nsIdleTimeout: 0
```

- Import the LDIF (change localhost to an IPA server if needed). A prompt for the Directory Manager password will be presented:

```
ldapmodify -h localhost -p 389 -x -D \
"cn=Directory Manager" -W -f forgejo.ldif
```

- Add an IPA group for forgejo_users:

```
ipa group-add --desc="Forgejo Users" forgejo_users
```

- Note: For errors about IPA credentials, run `kinit admin` and provide the domain admin account password.
- Log in to Forgejo as an Administrator and click on "Authentication" under the Admin Panel. Then click `Add New Source` and fill in the details, changing all where appropriate.

□□□ #2
□ □□□ □□□ 2026-10-31 13:34:03 UTC
□ □□□ □□□ 2026-10-31 13:34:04 UTC