

LDAP Provider Generic Setup (Authentik)

Create User/Group

1. Create a new user account to bind with under *Directory* -> *Users* -> *Create*, in this example called `ldapservice`.

Note the DN of this user will be `cn=ldapservice,ou=users,dc=ldap,dc=goauthentik,dc=io`

2. Create a new group for LDAP searches. In this example `ldapsearch`. Add the `ldapservice` user to this new group.NFO

Note: The `default-authentication-flow` validates MFA by default, and currently everything but SMS-based devices are supported by LDAP. If you plan to use only dedicated service accounts to bind to LDAP, or don't use SMS-based authenticators, then you can use the default flow and skip the extra steps below and continue at [Create LDAP Provider](#)

LDAP Flow

Create Custom Stages

1. Create a new identification stage. *Flows & Stage* -> *Stages* -> *Create*
`general_setup1-28b081e6970d26d4c2bf089c6db0c572.png`
2. Name it something meaningful like `ldap-identification-stage`. Select User fields Username and Email (and UPN if it is relevant to your setup).
`general_setup2-aedc0968da698af2d004996d6bf32172.png`
3. Create a new password stage. *Flows & Stage* -> *Stages* -> *Create*
`general_setup3-23a6d9635092816513b8bedcf5827ddb.png`
4. Name it something meaningful like `ldap-authentication-password`. Leave the defaults for Backends.
`general_setup4-9f6514bc905e1c7fa252313c83b443e6.png`
5. Create a new user login stage. *Flows & Stage* -> *Stages* -> *Create*
`general_setup5-b9e688db2f3f23b57cdd23ab3ce74182.png`
6. Name it something meaningful like `ldap-authentication-login`.
`general_setup6-a39cb7f17b1e677695ad7ac20659fc36.png`

Create Custom Flow

1. Create a new authentication flow under *Flows & Stage -> Flows -> Create*, and name it something meaningful like `ldap-authentication-flow`
general_setup7-5d28d5df6b1b2e69952919bfe30779b5.png
2. Click the newly created flow and choose *Stage Bindings*.
general_setup8-514c7914c49400ee061d0a0df8e2b9f5.png
3. Click `Bind Stage` choose `ldap-identification-stage` and set the order to `10`.
general_setup9-6a04b5c7c774b548fdd3dccb887a155f.png
4. Click `Bind Stage` choose `ldap-authentication-login` and set the order to `30`.
general_setup11-efd91bc25673f7df384b6d60de33f085.png
5. Edit the `ldap-identification-stage`.
general_setup12-25cc9fcea6654935c3587b21922b77fd.png
6. Change the Password stage to `ldap-authentication-password`.
general_setup13-a5006f341fdab49f32894d50e769a9c7.png

Create LDAP Provider

1. Create the LDAP Provider under *Applications -> Providers -> Create*.
general_setup14-9c2a270903b5bf88b693884c22d32c96.png
2. Name is something meaningful like `LDAP`, bind the custom flow created previously (or the default flow, depending on setup) and specify the search group created earlier.
general_setup15-3935cd6f3c497e436edacb53bec095e1.png

Create LDAP Application

1. Create the LDAP Application under *Applications -> Applications -> Create* and name it something meaningful like `LDAP`. Choose the provider created in the previous step.
general_setup16-1c57654a5ad05b12d3a21fd58e0e8350.png

Create LDAP Outpost

1. Create (or update) the LDAP Outpost under *Applications -> Outposts -> Create*. Set the Type to `LDAP` and choose the `LDAP` application created in the previous step.
general_setup17-1a8524a769f4f4d24e59522be62ea6cb.png

INFO

The LDAP Outpost selects different providers based on their Base DN. Adding multiple providers with the same Base DN will result in inconsistent access

Idpsearch Test

Test connectivity by using ldapsearch.

INFO

ldapsearch can be installed on Linux system with these commands

```
sudo apt-get install ldap-utils -y # Debian-based systems
sudo yum install openldap-clients -y # CentOS-based systems
```

```
ldapsearch \
-x \
-h <LDAP Outpost IP address> \
-p 389 \ # Production should use SSL 636
-D 'cn=ldapservice,ou=users,DC=ldap,DC=goauthentik,DC=io' \
-w '<ldapuserpassword>' \
-b 'DC=ldap,DC=goauthentik,DC=io' \
'(objectClass=user)'
```

INFO This query will log the first successful attempt in an event in the *Events* -> *Logs* area, further successful logins from the same user are not logged as they are cached in the outpost.

Manual Outpost deployment in docker-compose

To deploy an outpost with docker-compose, use this snippet in your docker-compose file.

You can also run the outpost in a separate docker-compose project, you just have to ensure that the outpost container can reach your application container.

Proxy outpost

```
version: "3.5"

services:
  authentik_proxy:
    image: ghcr.io/goauthentik/proxy
    # Optionally specify which networks the container should be
    # might be needed to reach the core authentik server
    # networks:
    #   - foo
    ports:
      - 9000:9000
      - 9443:9443
    environment:
      AUTHENTIK_HOST: https://your-authentik.tld
      AUTHENTIK_INSECURE: "false"
      AUTHENTIK_TOKEN: token-generated-by-authentik
      # Starting with 2021.9, you can optionally set this too
      # when authentik_host for internal communication doesn't match
the public URL
      # AUTHENTIK_HOST_BROWSER: https://external-domain.tld
```

LDAP outpost

```
version: "3.5"

services:
  authentik_ldap:
    image: ghcr.io/goauthentik/ldap
    # Optionally specify which networks the container should be
    # might be needed to reach the core authentik server
    # networks:
    #   - foo
    ports:
      - 389:3389
      - 636:6636
    environment:
      AUTHENTIK_HOST: https://your-authentik.tld
      AUTHENTIK_INSECURE: "false"
      AUTHENTIK_TOKEN: token-generated-by-authentik
```

