+++ title = "LDAP Authentication" description = "Grafana LDAP Authentication Guide" keywords = ["grafana", "configuration", "documentation", "ldap", "active directory"] aliases = ["/docs/grafana/latest/installation/ldap/"] weight = 300 +++

LDAP Authentication

The LDAP integration in Grafana allows your Grafana users to login with their LDAP credentials. You can also specify mappings between LDAP group memberships and Grafana Organization user roles.

> [Enhanced LDAP authentication]({{< relref "../enterprise/enhanced_ldap.md" >}}) is available in Grafana Cloud Advanced and in [Grafana Enterprise]({{< relref "../enterprise" >}}).
>
> Refer to [Fine-grained access control]({{< relref "../enterprise/access-control/_index.md" >}}) in Grafana Enterprise to understand how you can control access with fine-grained permissions.

Supported LDAP Servers

Grafana uses a third-party LDAP library under the hood that supports basic LDAP v3 functionality. This means that you should be able to configure LDAP integration using any compliant LDAPv3 server, for example OpenLDAP or Active Directory among others.

Enable LDAP

In order to use LDAP integration you'll first need to enable LDAP in the [main config file]({{< relref "../administration/configuration.md" >}}) as well as specify the path to the LDAP specific configuration file (default: `/etc/grafana/ldap.toml`).

```
[auth.ldap]
# Set to `true` to enable LDAP integration (default: `false`)
enabled = true

# Path to the LDAP specific configuration file (default: `/etc/grafana/ldap.toml`)
config_file = /etc/grafana/ldap.toml

# Allow sign up should almost always be true (default) to allow new Grafana users to be cre
# false only pre-existing Grafana users will be able to login (if LDAP authentication is ok
allow_sign_up = true
```

Grafana LDAP Configuration

Depending on which LDAP server you're using and how that's configured your Grafana LDAP configuration may vary. See configuration examples for more information.

LDAP specific configuration file (ldap.toml) example:

```
[[servers]]
# Ldap server host (specify multiple hosts space separated)
host = "127.0.0.1"
# Default port is 389 or 636 if use_ssl = true
port = 389
# Set to true if LDAP server should use an encrypted TLS connection (either with STARTTLS o
use_ssl = false
# If set to true, use LDAP with STARTTLS instead of LDAPS
start_tls = false
# set to true if you want to skip SSL cert validation
ssl_skip_verify = false
# set to the path to your root CA certificate or leave unset to use system defaults
# root_ca_cert = "/path/to/certificate.crt"
# Authentication against LDAP servers requiring client certificates
# client_cert = "/path/to/client.crt"
# client_key = "/path/to/client.key"

# Search user bind dn
bind_dn = "cn=admin,dc=grafana,dc=org"
# Search user bind password
# If the password contains # or ; you have to wrap it with triple quotes. Ex """#password;"
bind_password = "grafana"

# User search filter, for example "(cn=%s)" or "(sAMAccountName=%s)" or "(uid=%s)"
# Allow login from email or username, example "(|(sAMAccountName=%s)(userPrincipalName=%s))
search_filter = "(cn=%s)"

# An array of base dns to search through
search_base_dns = ["dc=grafana,dc=org"]

# group_search_filter = "(&(objectClass=posixGroup)(memberUid=%s))"
# group_search_filter_user_attribute = "distinguishedName"
# group_search_base_dns = ["ou=groups,dc=grafana,dc=org"]

# Specify names of the LDAP attributes your LDAP uses
[servers.attributes]
member_of = "memberOf"
email =  "email"
```

Using environment variables

You can interpolate variables in the TOML configuration from environment variables. For instance, you could externalize your `bind_password` that way:

```
bind_password = "${LDAP_ADMIN_PASSWORD}"
```

LDAP Debug View

> Only available in Grafana v6.4+

Grafana has an LDAP debug view built-in which allows you to test your LDAP configuration directly within Grafana. At the moment of writing, only Grafana admins can use the LDAP debug view.

Within this view, you'll be able to see which LDAP servers are currently reachable and test your current configuration.

{{< figure src="/static/img/docs/ldap_debug.png" class="docs-image–no-shadow" max-width="600px" >}}

To use the debug view:

1. Type the username of a user that exists within any of your LDAP server(s)
2. Then, press "Run"
3. If the user is found within any of your LDAP instances, the mapping information is displayed

{{< figure src="/static/img/docs/ldap_debug_mapping_testing.png" class="docs-image–no-shadow" max-width="600px" >}}

Bind

Bind and Bind Password By default the configuration expects you to specify a bind DN and bind password. This should be a read only user that can perform LDAP searches. When the user DN is found a second bind is performed with the user provided username and password (in the normal Grafana login form).

```
bind_dn = "cn=admin,dc=grafana,dc=org"
bind_password = "grafana"
```

Single Bind Example If you can provide a single bind expression that matches all possible users, you can skip the second bind and bind against the user DN directly. This allows you to not specify a bind_password in the configuration file.

```
bind_dn = "cn=%s,o=users,dc=grafana,dc=org"
```

In this case you skip providing a `bind_password` and instead provide a `bind_dn` value with a `%s` somewhere. This will be replaced with the username entered in on the Grafana login page. The search filter and search bases settings are still needed to perform the LDAP search to retrieve the other LDAP information (like LDAP groups and email).

POSIX schema

If your LDAP server does not support the memberOf attribute add these options:

```
## Group search filter, to retrieve the groups of which the user is a member (only set if me
group_search_filter = "(&(objectClass=posixGroup)(memberUid=%s))"
## An array of the base DNs to search through for groups. Typically uses ou=groups
group_search_base_dns = ["ou=groups,dc=grafana,dc=org"]
## the %s in the search filter will be replaced with the attribute defined below
group_search_filter_user_attribute = "uid"
```

Group Mappings

In `[[servers.group_mappings]]` you can map an LDAP group to a Grafana organization and role. These will be synced every time the user logs in, with LDAP being the authoritative source. So, if you change a user's role in the Grafana Org. Users page, this change will be reset the next time the user logs in. If you change the LDAP groups of a user, the change will take effect the next time the user logs in.

The first group mapping that an LDAP user is matched to will be used for the sync. If you have LDAP users that fit multiple mappings, the topmost mapping in the TOML configuration will be used.

LDAP specific configuration file (ldap.toml) example:

```
[[servers]]
# other settings omitted for clarity

[[servers.group_mappings]]
group_dn = "cn=superadmins,dc=grafana,dc=org"
org_role = "Admin"
grafana_admin = true # Available in Grafana v5.3 and above

[[servers.group_mappings]]
group_dn = "cn=admins,dc=grafana,dc=org"
org_role = "Admin"

[[servers.group_mappings]]
group_dn = "cn=users,dc=grafana,dc=org"
org_role = "Editor"

[[servers.group_mappings]]
group_dn = "*"
org_role = "Viewer"
```

Setting	Required	Description	Default
`group_dn`	Yes	LDAP distinguished name (DN) of LDAP group. If you want to match all (or no LDAP groups) then you can use wildcard (`"*"`)	
`org_role`	Yes	Assign users of `group_dn` the organization role `"Admin"`, `"Editor"` or `"Viewer"`	
`org_id`	No	The Grafana organization database id. Setting this allows for multiple group_dn's to be assigned to the same `org_role` provided the `org_id` differs	`1` (default org id)
`grafana_admin`	No	When `true` makes user of `group_dn` Grafana server admin. A Grafana server admin has admin access over all organizations and users. Available in Grafana v5.3 and above	`false`

Nested/recursive group membership

Users with nested/recursive group membership must have an LDAP server that supports `LDAP_MATCHING_RULE_IN_CHAIN` and configure `group_search_filter` in a way that it returns the groups the submitted username is a member of.

To configure `group_search_filter`:

- You can set `group_search_base_dns` to specify where the matching groups are defined.
- If you do not use `group_search_base_dns`, then the previously defined `search_base_dns` is used.

Active Directory example:

Active Directory groups store the Distinguished Names (DNs) of members, so your filter will need to know the DN for the user based only on the submitted username. Multiple DN templates can be searched by combining filters with the LDAP OR-operator. Two examples:

```
group_search_filter = "(member:1.2.840.113556.1.4.1941:=%s)"
group_search_base_dns = ["DC=mycorp,DC=mytld"]
group_search_filter_user_attribute = "dn"
```

```
group_search_filter = "(member:1.2.840.113556.1.4.1941:=CN=%s,[user container/OU])"
group_search_filter = "(|(member:1.2.840.113556.1.4.1941:=CN=%s,[user container/OU])(member
group_search_filter_user_attribute = "cn"
```

For more information on AD searches see Microsoft's Search Filter Syntax documentation.

For troubleshooting, by changing `member_of` in `[servers.attributes]` to "dn" it will show you more accurate group memberships when debug is enabled.

Configuration examples

OpenLDAP

OpenLDAP is an open source directory service.

LDAP specific configuration file (ldap.toml):

```
[[servers]]
host = "127.0.0.1"
port = 389
use_ssl = false
start_tls = false
ssl_skip_verify = false
bind_dn = "cn=admin,dc=grafana,dc=org"
bind_password = "grafana"
search_filter = "(cn=%s)"
search_base_dns = ["dc=grafana,dc=org"]

[servers.attributes]
member_of = "memberOf"
email =  "email"

# [[servers.group_mappings]] omitted for clarity
```

Multiple LDAP servers

Grafana does support receiving information from multiple LDAP servers.

LDAP specific configuration file (ldap.toml):

```
# --- First LDAP Server ---

[[servers]]
host = "10.0.0.1"
port = 389
use_ssl = false
start_tls = false
ssl_skip_verify = false
bind_dn = "cn=admin,dc=grafana,dc=org"
bind_password = "grafana"
search_filter = "(cn=%s)"
search_base_dns = ["ou=users,dc=grafana,dc=org"]

[servers.attributes]
member_of = "memberOf"
email =  "email"

[[servers.group_mappings]]
```

```
group_dn = "cn=admins,ou=groups,dc=grafana,dc=org"
org_role = "Admin"
grafana_admin = true

# --- Second LDAP Server ---

[[servers]]
host = "10.0.0.2"
port = 389
use_ssl = false
start_tls = false
ssl_skip_verify = false

bind_dn = "cn=admin,dc=grafana,dc=org"
bind_password = "grafana"
search_filter = "(cn=%s)"
search_base_dns = ["ou=users,dc=grafana,dc=org"]

[servers.attributes]
member_of = "memberOf"
email =  "email"

[[servers.group_mappings]]
group_dn = "cn=editors,ou=groups,dc=grafana,dc=org"
org_role = "Editor"

[[servers.group_mappings]]
group_dn = "*"
org_role = "Viewer"
```

Active Directory

Active Directory is a directory service which is commonly used in Windows environments.

Assuming the following Active Directory server setup:

- IP address: `10.0.0.1`
- Domain: `CORP`
- DNS name: `corp.local`

LDAP specific configuration file (ldap.toml):

```
[[servers]]
host = "10.0.0.1"
port = 3269
use_ssl = true
start_tls = false
```

```
ssl_skip_verify = true
bind_dn = "CORP\\%s"
search_filter = "(sAMAccountName=%s)"
search_base_dns = ["dc=corp,dc=local"]

[servers.attributes]
member_of = "memberOf"
email =  "mail"

# [[servers.group_mappings]] omitted for clarity
```

Port requirements In above example SSL is enabled and an encrypted port have been configured. If your Active Directory don’t support SSL please change `enable_ssl = false` and `port = 389`. Please inspect your Active Directory configuration and documentation to find the correct settings. For more information about Active Directory and port requirements see link.

Troubleshooting

To troubleshoot and get more log info enable LDAP debug logging in the [main config file]({{< relref “../administration/configuration.md” >}}).

```
[log]
filters = ldap:debug
```