# LDAP Authentication

Redmine natively supports LDAP authentication using one or multiple LDAP directories.

## Declaring the LDAP

Go to Administration and click **LDAP authentication** in the menu.

Enter the following:

- **Name**: an arbitrary name for the directory
- **Host**: the LDAP host name
- **Port**: the LDAP port (default is 389)
- **LDAPS**: check this if you want or need to use LDAPS to access the directory
- **Account**: enter a username that has read access to the LDAP , otherwise leave this field empty if your LDAP can be read anonymously (Active Directory servers generally do not allow anonymous access)
- **Password**: password for the account
- **Base DN**: the top level DN of your LDAP directory tree
- **Login attribute**: enter the name of the LDAP attribute that will be used as the Redmine username

Redmine users should now be able to authenticate using their LDAP username and password if their accounts are set to use the LDAP for authentication.

To test this, create a Redmine user with a login that matches his LDAP account (normally, Redmine will advise you by looking up the LDAP data), select the newly created LDAP in the **Authentication mode** drop-down list (this field is visible on the account screen only if a LDAP is declared) and leave his password empty. Try to log in into Redmine using the LDAP username and password.

## On the fly user creation

By checking **on-the-fly user creation**, any LDAP user will have his Redmine account automatically created the first time he logs into Redmine.
For that, you have to specify the LDAP attributes name (firstname, lastname, email) that will be used to create their Redmine accounts.

Here is an typical example using Active Directory:

```
Name     = My Directory
Host     = host.domain.org
Port     = 389
LDAPS    = no
Account  = MyDomain\UserName (or UserName@MyDomain depending on AD server)
Password = <password>
Base DN  = CN=users,DC=host,DC=domain,DC=org

On-the-fly user creation = yes
Attributes
  Login     = sAMAccountName
  Firstname = givenName
  Lastname  = sN
  Email     = mail
```

Here is another example for Active Directory with a compartmentalized intranet:

```
Name     = Just a description for the auth modes page
Host     = DepartmentName.OrganizationName.local
Port     = 389
LDAPS    = no
Account  = DepartmentName\UserName (or UserName@MyDomain depending on AD server or bind DN uid=Manager,cn=users,dc=MyDomain,dc=com)
Password = <password>
Base DN  = DC=DepartmentName,DC=OrganizationName,DC=local
```

```
On-the-fly user creation = yes
Attributes
  Login     = sAMAccountName
  Firstname = givenName
  Lastname  = sN
  Email     = mail
```

Note that LDAP attribute names are **case sensitive**.

### Dynamic Bind Account

The above setup would need a special account on the directory server which Redmine uses to pre-authenticate. It is possible to use the keyword **$login** in the account field which then would be replaced by the current login. The password can be left empty in this case, for example:

```
Account: $login@COMPANY.DOMAIN.NAME
```

or

```
Account: company\$login
```

### Base DN variants

Although it's quite possible that the Base DN above is standard for Active Directory, the Active Directory at my employer's site does not use the Users container for standard users, so those instructions sent me down a long and painful path.  I recommend also trying just "DC=host,DC=domain,DC=org" if login fail swith the settings there.

## Group based LDAP login

If you want to just allow logins to users that belongs to a particular LDAP group you should follow below instructions. They are based on OpenLDAP LDAP server and redmine 2.3.0.

1. (OpenLDAP server) Enable memberof overlay

1.1. Create a file:

```
vim ~/memberof_add.ldif
```

With below content:

```
dn: cn=module,cn=config
objectClass: olcModuleList
cn: module
olcModulePath: /usr/lib/ldap
olcModuleLoad: memberof
```

1.2. Create a file:

```
vim ~/memberof_config.ldif
```

With below content:

```
dn: olcOverlay=memberof,olcDatabase={1}hdb,cn=config
objectClass: olcMemberOf
objectClass: olcOverlayConfig
objectClass: olcConfig
objectClass: top
olcOverlay: memberof
olcMemberOfDangling: ignore
olcMemberOfRefInt: TRUE
olcMemberOfGroupOC: groupOfNames
olcMemberOfMemberAD: member
olcMemberOfMemberOfAD: memberOf
```

1.3. Load them. It will depend on your OpenLDAP configuration, so we will propose some possibilities:

```
sudo ldapadd -c -Y EXTERNAL -H ldapi:/// -f memberof_add.ldif
sudo ldapadd -c -Y EXTERNAL -H ldapi:/// -f memberof_config.ldif
```

Or:

```
ldapadd -D cn=admin,cn=config -w "password" -H ldapi:/// -f memberof_add.ldif
ldapadd -D cn=admin,cn=config -w "password" -H ldapi:/// -f memberof_config.ldif
```

A restart is NOT needed if you use dynamic runtime configuration engine (slapd-config).

1.4. (Optional) Test it:

```
ldapsearch -D cn=admin,dc=example,dc=com -x -W -b 'dc=example,dc=com' -H 'ldap://127.0.0.1:389/'
'(&(objectClass=posixAccount)(memberOf=cn=ldapredmine,ou=groups,dc=example,dc=com))'
```

2. (OpenLDAP server) Create the group. In this example the user is "ldap_user_1" and the group is "ldapredmine":

```
dn: cn=ldapredmine,ou=groups,dc=example,dc=com
cn: ldapredmine
description: Staff members allowed to login to redmine ticketing system
member: cn=ldap_user_1,ou=people,dc=example,dc=com
objectclass: groupOfNames
objectclass: top
```

Adjust "dn" and "cn"s to fit to your DIT structure

3. (Redmine) Edit the LDAP authentication mode. In my case "ldap_user_1" is a "posixAccount" objectclass:

```
Base DN: dc=example,dc=com
Filter: (&(objectClass=posixAccount)(memberOf=cn=ldapredmine,ou=groups,dc=example,dc=com))
```

## Troubleshooting

If you want to use on-the-fly user creation, make sure that Redmine can fetch from your LDAP all the required information to create a valid user.
For example, on-the-fly user creation won't work if you don't have valid email adresses in your directory (you will get an 'Invalid username/password' error message when trying to log in).
(This is not true with newer Redmine versions; the user creation dialog is populated with everything it can find from the LDAP server, and asks the new user to fill in the rest.)

Also, make sure you don't have any custom field marked as **required** for user accounts. These custom fields would prevent user accounts from being created on the fly.

Errors in the login system are not reported with any real information in the Redmine logs, which makes troubleshooting difficult. However, you can found most of the information you need using Wireshark between your Redmine host and the LDAP server. Note that this only works if you have permissions to read network traffic between those two hosts.

You can also use the tool 'ldapsearch' to test if your settings are correct. Log into the Linux machine hosting your redmine (and possibly install ldaputils) and run this:

```
ldapsearch -x -b  "dc=example,dc=com" -H ldap://hostname/ -D "DOMAIN\USER" -w mypassword [searchterm]
```

If succesful, you will get a listing of the contents of the AD, matching your search query. Then, you will know what how to fill out the fields in the LDAP config in Redmine.

### Account value format

The username for the bind credentials might need to be specified as a DN rather than as a UPN(user@domain.com) or as domain\user, as pointed out by this comment in source:trunk/vendor/plugins/ruby-net-ldap-0.0.4/lib/net/ldap.rb:

```
  # As described under #bind, most LDAP servers require that you supply a complete DN
  # as a binding-credential, along with an authenticator such as a password.
```

Therefore user with MyDomain\MyUserName or MyUserName@MyDomain.com username might enter only MyUserName as a Redmine login name.

### Slow LDAP authentification

If LDAP authentification is slow and you have an AD cluster, try to specify in Host field one of the AD physical servers ( http://www.redmine.org/boards/2/topics/3056). It may help.

### OpenDS

If you are using the OpenDS server, you might have issues with the request control "Paged results" sent with the initial query searching for the user by the specified login attribute. This request control 1.2.840.113556.1.4.319 is not allowed for anonymous users by default, thus preventing redmine from finding the user in the directory even before the binding takes place.

Add a global ACI like this

```
./dsconfig -h SERVER_IP -p 4444 -D cn="Directory Manager" -w PASSWORD -n set-access-control-handle
r-prop --trustAll
--add global-aci:\(targetcontrol=\"1.2.840.113556.1.4.319\"\)\ \(version\ 3.0\;\ acl\
\"Anonymous\ control\ access\ to\ 1.2.840.113556.1.4.319\"\;\ allow\ \(read\)\ userdn=\"ldap:///an
yone\"\;\)
```

Note: Enter the command on one line, use the escaping exactly as indicated (the \ after "acl" is meant to be "\ " for a space).

## Solutions:

### Zentyal 3.2, Redmine 2.3.x

I successfully updated and (re)setup my Zentyal 3.2 on an Ubuntu 12.04 LTS server.

Because this really drove me nuts after an update to Zentyal 3.2 and Redmine 2.3, I like to make the story short and share this simple solution with you:

- https://wiki.blue-it.org/Zentyal#LDAP

Using zentyals readonly credentials:

```
> Basedomain (Base DN):  dc=your_domain,dc=your_tld
  Rootdomain (Root DN):  cn=zentyal,dc=your_domain,dc=your_tld
  Password:              <admin_secret_pass>
> Read-only root DN:     cn=zentyalro,dc=your_domain,dc=your_tld
> Read-only password:    <ro_secret_pass>
  Default Users DN:      ou=Users,dc=your_domain,dc=your_tld
  Default Groups DN:     ou=Groups,dc=your_domain,dc=your_tld
```

And in Redmine (use the credentials above, without <>) and be aware of the changed LDAP port 390 (read the article above):

```
Name      = Just a description for the auth modes page
> Host      = <IP of the host>
> Port      = <390>
  LDAPS     = no
> Account   = <Read-only root DN>
> Password = <ro_secret_pass>
> Base DN   = <Basedomain (Base DN)>

On-the-fly user creation = yes
```

```
Attributes
>  Login     = uid
   Firstname = givenName
>  Lastname  = sN
  Email     = mail
```

### Zentyal 4.x, Redmine 3.3.x

Use port 389 and sAMAccountName instead of uid

### Zentyal 5.0, Redmine 3.2

```
  Name     = Just a description for the auth modes page
  Host     = <IP of the host>
  Port     = <389>
  LDAPS    = no (yes/checked is ok too with Port set to 636)
  Account  = username@domain.tld
  Password = <username_pass>
  Base DN  = < Default Users DN > (cn=Users,dc=domain,dc=tld)

On-the-fly user creation = yes
Attributes
   Login     = sAMAccountName
   Firstname = givenName
   Lastname  = sN
   Email     = mail
```

### OpenLDAP, Redmine 3.4.2.stable

```
  Name     = Some random description
  Host     = <IP of the host>
  Port     = <389>
  LDAPS    = no
  Account  = < Admins DN > (cn=admin,dc=domain,dc=tld)
  Password = < Admins Pass >
  Base DN  = < Default Users DN > (cn=People,dc=domain,dc=tld)

On-the-fly user creation = yes
Attributes
   Login     = uid
   Firstname = givenName
   Lastname  = sn
   Email     = mail
```

The admin account may be any other LDAP account with global read permission. The "domain" and "tld" part has to fit the LDAP setup, as everything else. Login attribute is used for login. The rest has to be according to LDAP setup.