Securing Your Open Source Geospatial Stack with Single Sign On

Ian Turton

Introduction

I'm Ian Turton, I work for Astun Technology providing coding, support and training for a range of open source programs:

- GeoServer
- QGis
- MapServer
- Python for GIS
- PostGIS

In my spare time I'm a moderator at `gis.stackexchange.com` and a regular contributor to GeoTools and GeoServer.

Get the slides

The Problem

- Provide a single login for QGIS users to WMS layers in GeoServer and PostGIS data tables
- Restrict access by team or user
- GUI for administration team
- Use existing Azure Active Directory

What is Single Sign On (SSO)?

Single sign-on (SSO) is a session and user authentication service that permits a user to use one set of login credentials – for example, a username and password – to access multiple applications. This is managed by providing a ***federated identity system****.*

Implementation

- Both GeoServer and PostGIS can make use of an LDAP service to provide authentication via a federated identity system
- LDAP provides the central store of identities and GeoServer and PostGIS pass username/password pairs to the LDAP server to check if the user is known to the system.

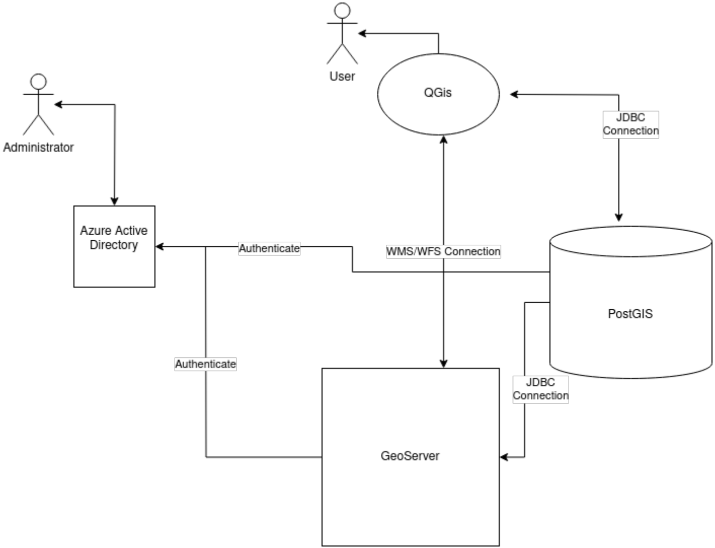
GeoServer

- GeoServer uses the built in LDAP Authentication Provider
- allows the use of users and groups for authorization
- same configuration as any other custom authentication provider (GUI)

PostGIS

> *This authentication method operates similarly to password except that it uses LDAP as the password verification method.* ***LDAP is used only to validate the user name/password pairs.*** *Therefore* ***the user must already exist in the database*** *before LDAP can be used for authentication.*

Add a line to the `pg_hba.conf` file in the same way as any other authentication method (e.g. peer, md5)


User
QGis
Administrator
JDBC Connection
Azure Active Directory
Authenticate
WMS/WFS Connection
PostGIS
Authenticate
JDBC Connection
GeoServer

The Pivot

- Corporate IT team inform client there is no way that the GIS team are altering entries in the Active Directory!
- Back to the drawing board
- Can we replace an expensive piece of Microsoft code with some open source tools.

Providing LDAP

- OpenLDAP (https://www.openldap.org/)
  - Provides a complete LDAP server
  - allows the import of users and groups (from exports from AD)
  - Comes pre-dockerized for easy installation

Add users to LDAP Server

```
# cat adam.ldif
dn: uid=adam,ou=users,dc=tgs,dc=com
objectClass: top
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount
cn: adam
uid: adam
uidNumber: 16859
gidNumber: 100
```

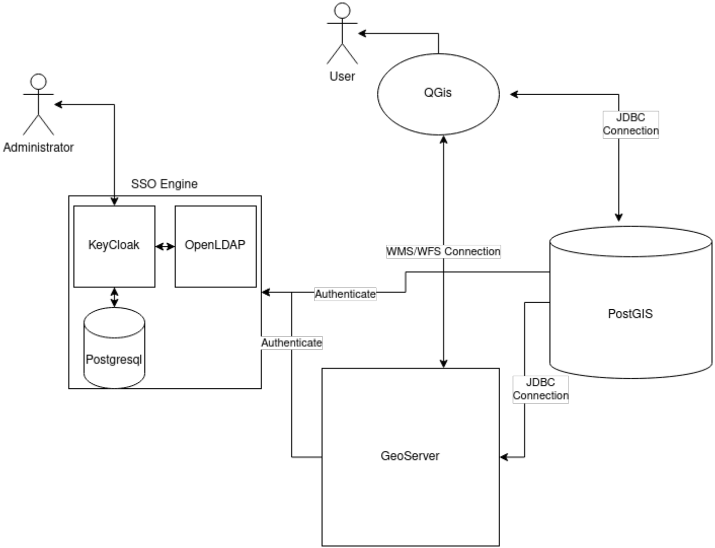
```
ldapadd -x -W -D "cn=ramesh,dc=tgs,dc=com" -f adam.ldif
```

Providing a “nice” GUI for Administrators

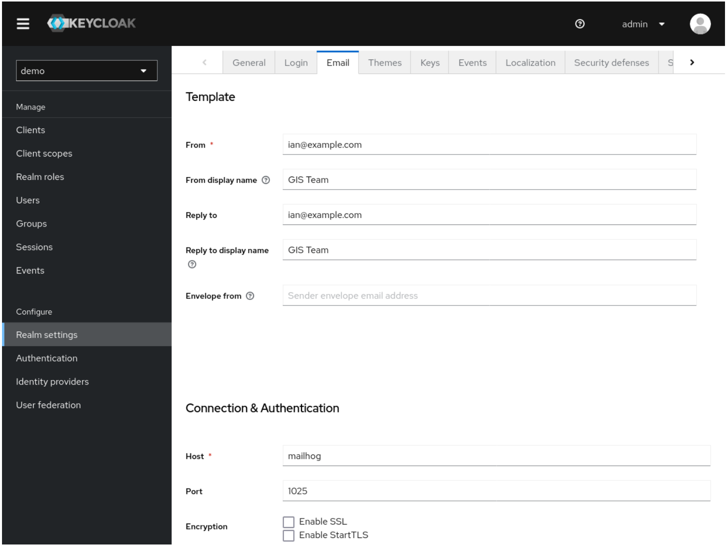
- These guys are GIS experts not IT experts
- So no CLI and python scripts :-(
- Deploy KeyCloak to manage the human to LDAP interactions

KeyCloak

- KeyCloak (https://www.keycloak.org/)
- Can be used to handle all your identity provider needs
- GeoServer support with an extension
- No PostGIS support!
- So use in federation mode, as a front end to OpenLDAP
- Comes pre-dockerized (with PostgreSQL in a container)


User
QGis
Administrator
JDBC Connection
SSO Engine
KeyCloak
OpenLDAP
WMS/WFS Connection
PostGIS
Authenticate
Authenticate
Postgresql
JDBC Connection
GeoServer

Set up KeyCloak Administration



Template

From * ian@example.com

From display name GIS Team

Reply to ian@example.com

Reply to display name GIS Team

Envelope from Sender envelope email address

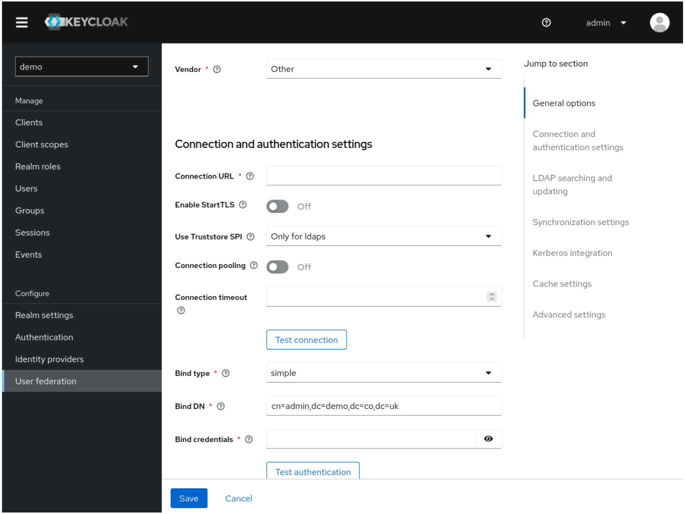
Connection & Authentication

Host * mailhog

Port 1025

Encryption
□ Enable SSL
□ Enable StartTLS

Connect to the LDAP server



Vendor * : Other

Connection and authentication settings

Connection URL * :

Enable StartTLS : Off

Use Truststore SPI : Only for ldaps

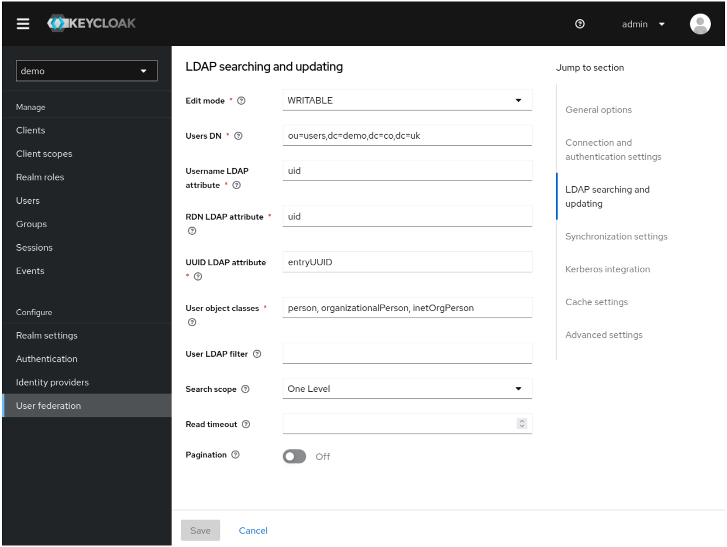
Connection pooling : Off

Connection timeout :

Bind type * : simple

Bind DN * : cn=admin,dc=demo,dc=co,dc=uk

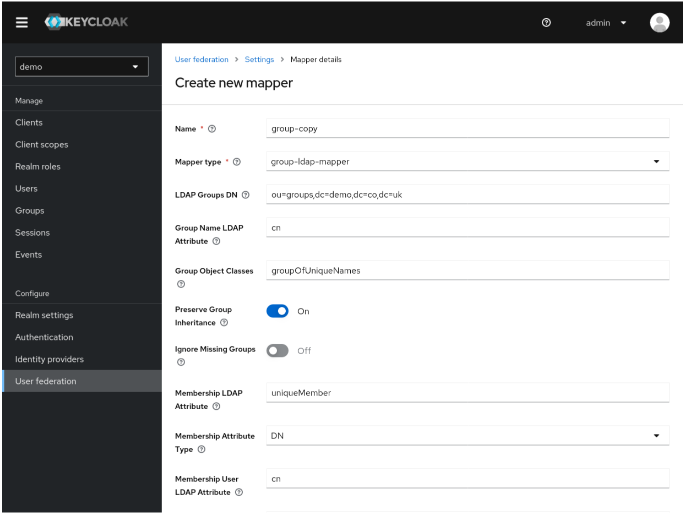
Bind credentials * :



LDAP searching and updating

Edit mode *	WRITABLE
Users DN *	ou=users,dc=demo,dc=co,dc=uk
Username LDAP attribute *	uid
RDN LDAP attribute *	uid
UUID LDAP attribute *	entryUUID
User object classes *	person, organizationalPerson, inetOrgPerson
User LDAP filter	
Search scope	One Level
Read timeout	
Pagination	Off

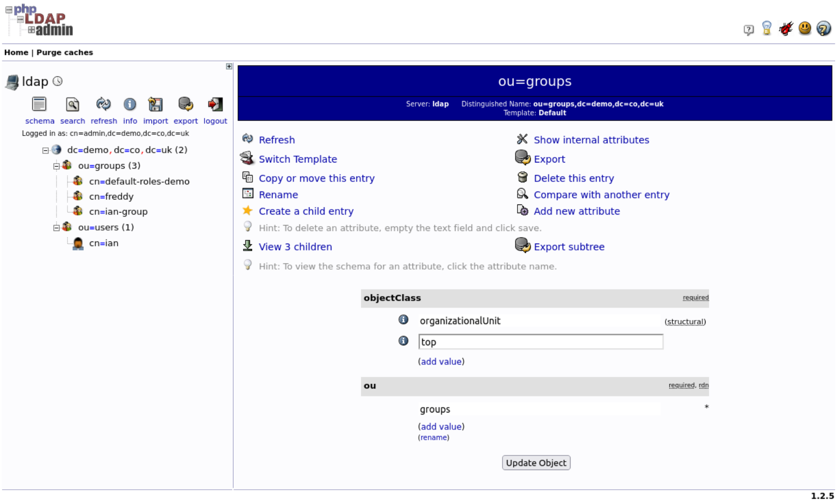
Configure Rules for mapping between KeyCloak and LDAP



Create new mapper

Name *	group-copy
Mapper type *	group-ldap-mapper
LDAP Groups DN	ou=groups,dc=demo,dc=co,dc=uk
Group Name LDAP Attribute	cn
Group Object Classes	groupOfUniqueNames
Preserve Group Inheritance	On
Ignore Missing Groups	Off
Membership LDAP Attribute	uniqueMember
Membership Attribute Type	DN
Membership User LDAP Attribute	cn

Create Group and User “Organizations” in LDAP



Logged in as: cn=admin,dc=demo,dc=co,dc=uk

- dc=demo, dc=co, dc=uk (2)
  - ou=groups (3)
    - cn=default-roles-demo
    - cn=freddy
    - cn=ian-group
  - ou=users (1)
    - cn=ian

ou=groups

Server: **ldap** Distinguished Name: **ou=groups,dc=demo,dc=co,dc=uk**
Template: **Default**

Hint: To delete an attribute, empty the text field and click save.

Hint: To view the schema for an attribute, click the attribute name.

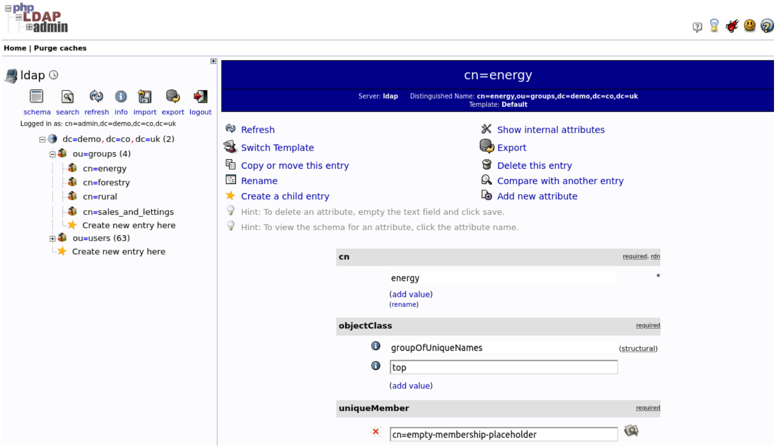
objectClass required

- organizationalUnit (structural)
- top

ou required, rdn

- groups *

Create groups in LDAP or KeyCloak



Home | Purge caches

ldap

Logged in as: cn=admin,dc=demo,dc=co,dc=uk

- dc=demo, dc=co, dc=uk (2)
  - ou=groups (4)
    - cn=energy
    - cn=forestry
    - cn=rural
    - cn=sales_and_lettings
    - Create new entry here
  - ou=users (63)
  - Create new entry here

cn=energy

Server: ldap Distinguished Name: cn=energy,ou=groups,dc=demo,dc=co,dc=uk
Template: Default

Hint: To delete an attribute, empty the text field and click save.

Hint: To view the schema for an attribute, click the attribute name.

cn required, rdn

energy *

(add value)
(rename)

objectClass required

groupOfUniqueNames (structural)

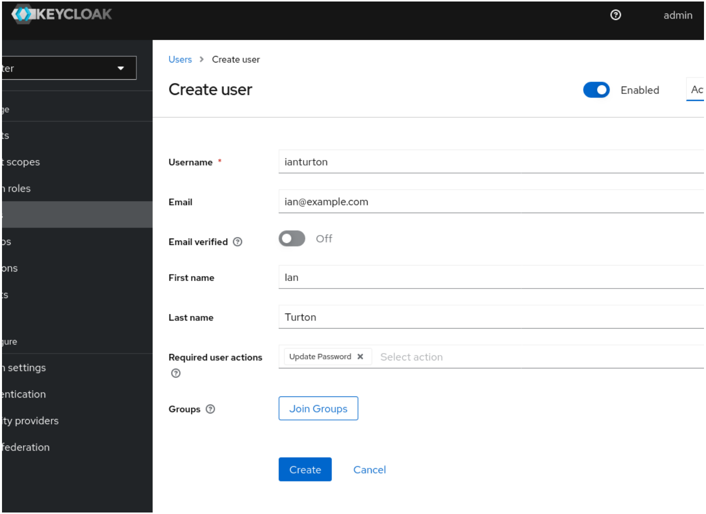
top

(add value)

uniqueMember required

cn=empty-membership-placeholder

Create Users in LDAP via KeyCloak or import



Create user

Enabled

Username * ianturton

Email ian@example.com

Email verified Off

First name Ian

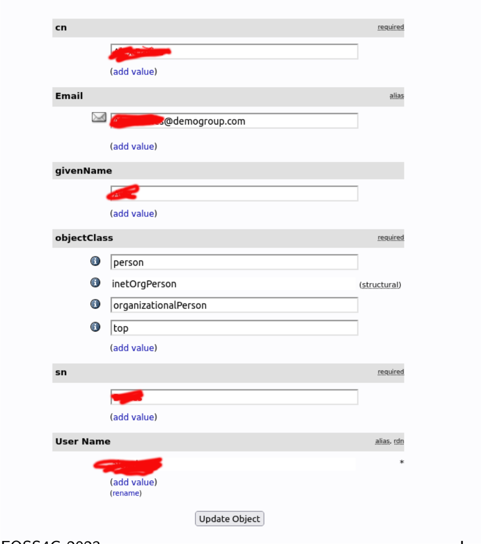
Last name Turton

Required user actions: Update Password ✕ Select action

Groups: Join Groups

Create Cancel

A user in LDAP (GDPR redacted)



cn required

[redacted]

(add value)

Email alias

[redacted]@demogroup.com

(add value)

givenName

[redacted]

(add value)

objectClass required

person

inetOrgPerson (structural)

organizationalPerson

top

(add value)

sn required

[redacted]

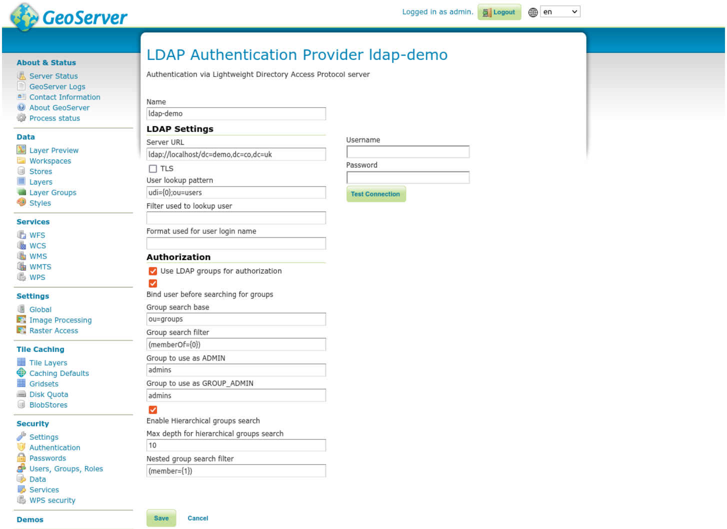
(add value)

User Name alias, rdn

[redacted] *

(add value)
(rename)

Configure GeoServer Authentication Provider



LDAP Authentication Provider ldap-demo

Authentication via Lightweight Directory Access Protocol server

Name: ldap-demo

LDAP Settings

Server URL: ldap://localhost/dc=demo,dc=co,dc=uk

☐ TLS

User lookup pattern: udi={0};ou=users

Filter used to lookup user:

Format used for user login name:

Username:

Password:

Test Connection

Authorization

☑ Use LDAP groups for authorization

☑ Bind user before searching for groups

Group search base: ou=groups

Group search filter: (memberOf={0})

Group to use as ADMIN: admins

Group to use as GROUP_ADMIN: admins

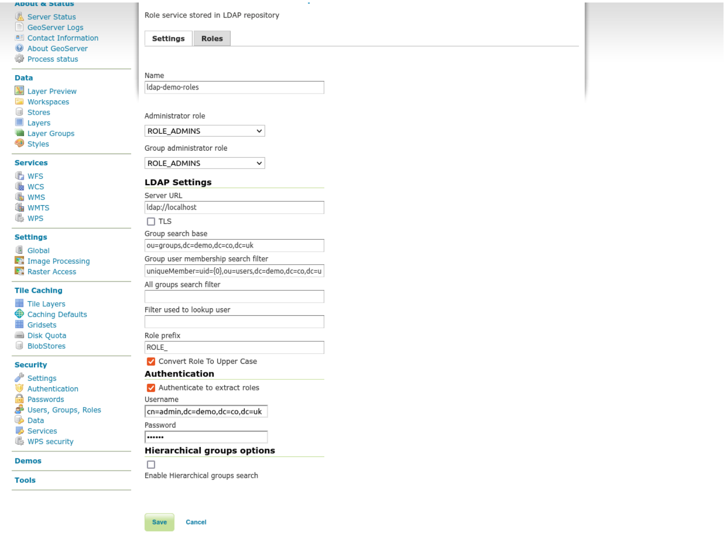
☑ Enable Hierarchical groups search

Max depth for hierarchical groups search: 10

Nested group search filter: (member={1})

Save Cancel

Configure GeoServer Role Provider



Role service stored in LDAP repository

Name: ldap-demo-roles

Administrator role: ROLE_ADMINS

Group administrator role: ROLE_ADMINS

LDAP Settings

Server URL: ldap://localhost

□ TLS

Group search base: ou=groups,dc=demo,dc=co,dc=uk

Group user membership search filter: uniqueMember=uid={0},ou=users,dc=demo,dc=co,dc=u

All groups search filter:

Filter used to lookup user:

Role prefix: ROLE_

☑ Convert Role To Upper Case

Authentication

☑ Authenticate to extract roles

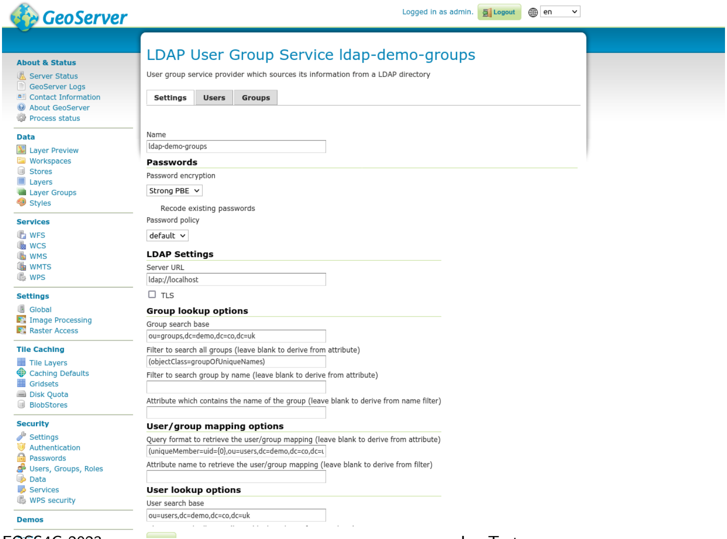
Username: cn=admin,dc=demo,dc=co,dc=uk

Password: ••••••

Hierarchical groups options

□ Enable Hierarchical groups search

Configure GeoServer Groups



LDAP User Group Service ldap-demo-groups

User group service provider which sources its information from a LDAP directory

Name: ldap-demo-groups

Passwords

Password encryption: Strong PBE

Recode existing passwords

Password policy: default

LDAP Settings

Server URL: ldap://localhost

□ TLS

Group lookup options

Group search base: ou=groups,dc=demo,dc=co,dc=uk

Filter to search all groups (leave blank to derive from attribute): (objectClass=groupOfUniqueNames)

Filter to search group by name (leave blank to derive from attribute):

Attribute which contains the name of the group (leave blank to derive from name filter):

User/group mapping options

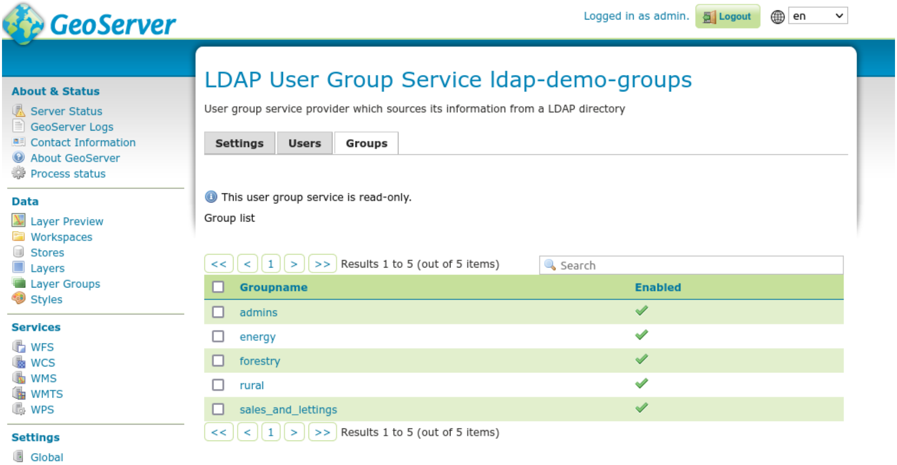
Query format to retrieve the user/group mapping (leave blank to derive from attribute): (uniqueMember=uid={0},ou=users,dc=demo,dc=co,dc=u

Attribute name to retrieve the user/group mapping (leave blank to derive from filter):

User lookup options

User search base: ou=users,dc=demo,dc=co,dc=uk

All the LDAP groups are imported automagically



LDAP User Group Service ldap-demo-groups

User group service provider which sources its information from a LDAP directory

This user group service is read-only.

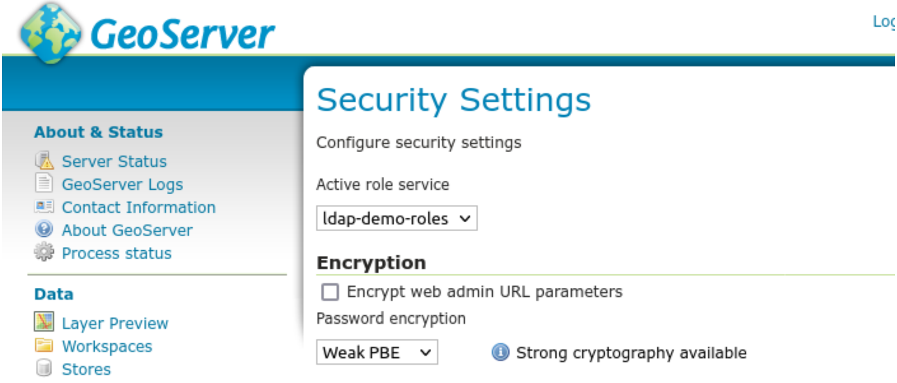
Group list

Results 1 to 5 (out of 5 items)

□	Groupname	Enabled
□	admins	✓
□	energy	✓
□	forestry	✓
□	rural	✓
□	sales_and_lettings	✓

Results 1 to 5 (out of 5 items)

Don’t forget to make the role service active



Security Settings

Configure security settings

Active role service

ldap-demo-roles

Encryption

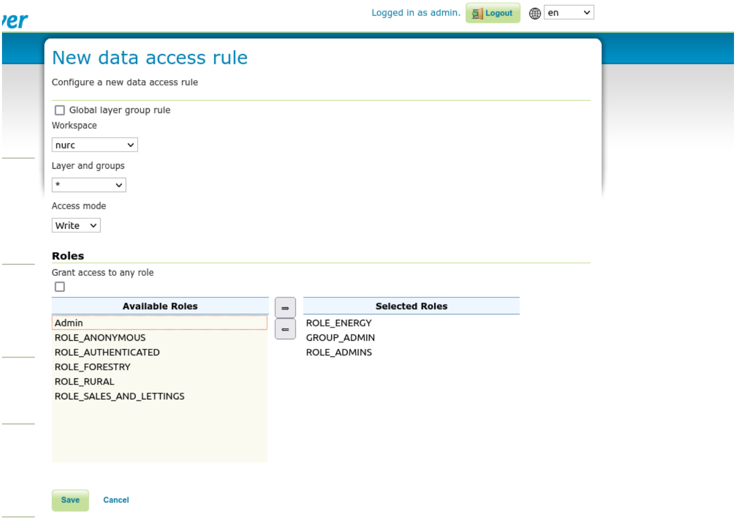
□ Encrypt web admin URL parameters

Password encryption

Weak PBE

Strong cryptography available

Limit access to a workspace (as per usual)



New data access rule

Configure a new data access rule

☐ Global layer group rule

Workspace: nurc

Layer and groups: *

Access mode: Write

Roles

Grant access to any role
☐

Available Roles	Selected Roles
Admin	ROLE_ENERGY
ROLE_ANONYMOUS	GROUP_ADMIN
ROLE_AUTHENTICATED	ROLE_ADMINS
ROLE_FORESTRY	
ROLE_RURAL	
ROLE_SALES_AND_LETTINGS	

Configure ldap-sync to keep PostGIS up to date

```
ldap_connection:
  host: localhost
  port: 389
  auth:
    method: :simple
    username: CN=admin,DC=demo,DC=co,DC=uk
    password: admin
```

```
# Search parameters for LDAP users which should be synchronized
ldap_users:
  base: ou=users,DC=demo,DC=co,DC=uk
  # LDAP filter (according to RFC 2254)
  # defines to users in LDAP to be synchronized
  filter: (&(objectClass=person)(cn=*))
  # this attribute is used as PG role name
  name_attribute: uid
  # lowercase name for use as PG role name
  lowercase_name: true
  # Add lowercase name *and* original name for use as PG role names (usefu
  bothcase_name: false

# Search parameters for LDAP groups which should be synchronized
ldap_groups:
  base: ou=groups,dc=demo,dc=co,dc=uk
  filter: objectClass=groupOfUniqueNames
  # this attribute is used as PG role name
  name_attribute: cn
```

Configure a cron job

```
*/1 * * * * pg_ldap_sync -c ldap-sync-config.yaml  >>
/var/log/sync.log 2>&1
```

- run the sync job every minute to copy any new LDAP users or groups to PostgreSQL

```
I, [2022-11-24T16:53:01 #24436]  INFO -- : found user-dn: uid=iturton,cn=u
I, [2022-11-24T16:53:01 #24436]  INFO -- : found user-dn: uid=test,cn=user
I, [2022-11-24T16:53:01 #24436]  INFO -- : found group-dn: cn=users,dc=gal
I, [2022-11-24T16:53:01 #24436]  INFO -- : found pg-user: "iturton"
I, [2022-11-24T16:53:01 #24436]  INFO -- : found pg-user: "test"
I, [2022-11-24T16:53:01 #24436]  INFO -- : found pg-group: "users" with me
I, [2022-11-24T16:53:01 #24436]  INFO -- : user stat: create: 0 drop: 0 ke
I, [2022-11-24T16:53:01 #24436]  INFO -- : group stat: create: 0 drop: 0 k
```

Modify the pg_hba.conf file to let PostgreSQL know who to trust

```
hostssl    all    all    127.0.0.1/32  ldap ldapserver=ldap-machine
     ldapbinddn="cn=admin,dc=demo,dc=co,dc=uk" ldapbindpasswd=secret!
     ldapbasedn="ou=users,dc=demo,dc=co,dc=uk"
```

```
psql -U astun -h demo-db -d "dbname=postgres
sslmode=require"
```

Problems

- managing email authentication
- not being flagged as phishing attempts by Microsoft Outlook
- custom KeyCloak event triggers so we can send custom emails
- Automating initial import of users, write a python script

Conclusions

- It is possible to provide a single sign on service to QGIS users to GeoServer and PostGIS
- KeyCloak is a very simple and intuitive interface to LDAP

Get the slides