OpenLDAP database non relazionale con specifica aperta ad oggetti

Indice
Introduzione e Storia
Applicazioni directory LDAP
Concetti ed organizzazione contenuti
Database di Controllo e configurazione
Linguaggio LDIF
Strumenti OpenSource
Esempi/Risorse

Pisa 18.05.2016 relatore: De Massis Fabrizio

Introduzione e Storia

Nasce fine anni '80 come derivato del DAP il servizio di Directory del protocollo ISO/OSI. Questo era implementato solo su MainFrame per via del carico di lavoro e del protocollo di rete.

LDAP acronimo di “Lightweight Directory Access protocol”, quindi un protocollo leggero, standard aperto, basato su TCP/IP per l'accesso ai dati della Directory.

Directory è un insieme di dati complessi organizzati in modo gerarchico, tipo il DNS (Domain Name System), oppure un albero gerarchico in un contesto aziendale, o un inventario, in pratica una strutturazione dei contenuti ad albero.

Rif.
https://it.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol

Applicazioni Directory LDAP

Rubriche aziendali (Gestione grandi Enti/Università)

Gestione degli indirizzi postali (una delle applicazioni iniziali)

Backend di autenticazione (attuale utilizzo più diffuso) [Utenti/Gruppi/Password/Configurazioni]

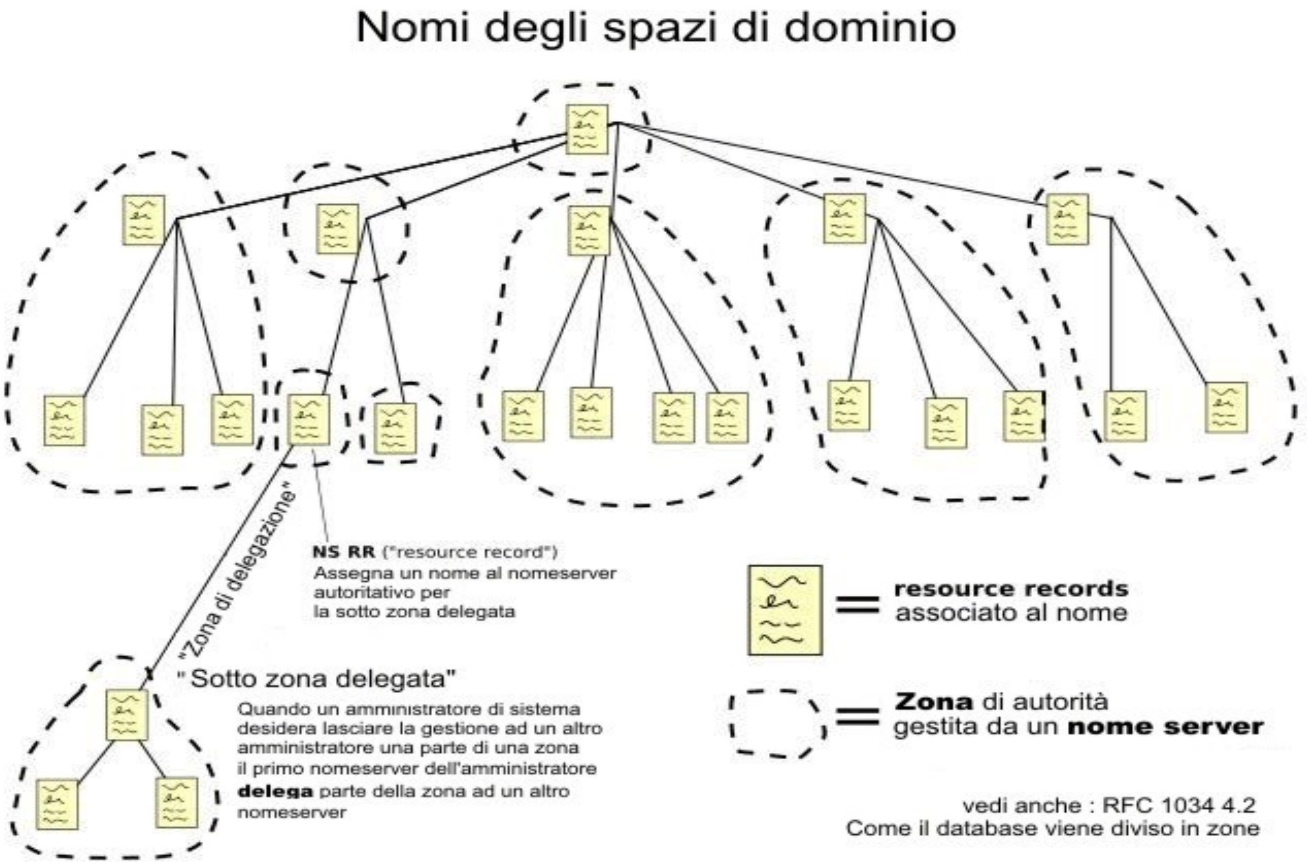
Contenitore per catalogazione di dispositivi di rete e stazioni di lavoro (Microsoft Active Directory, Apple OpenDirectory, quest'ultimo basato su OpenLDAP)

Rappresentazione di organigrammi (Microsoft Active Directory, Apple OpenDirectory, quest'ultimo basato su OpenLDAP)

Introduzione e Storia (DNS)

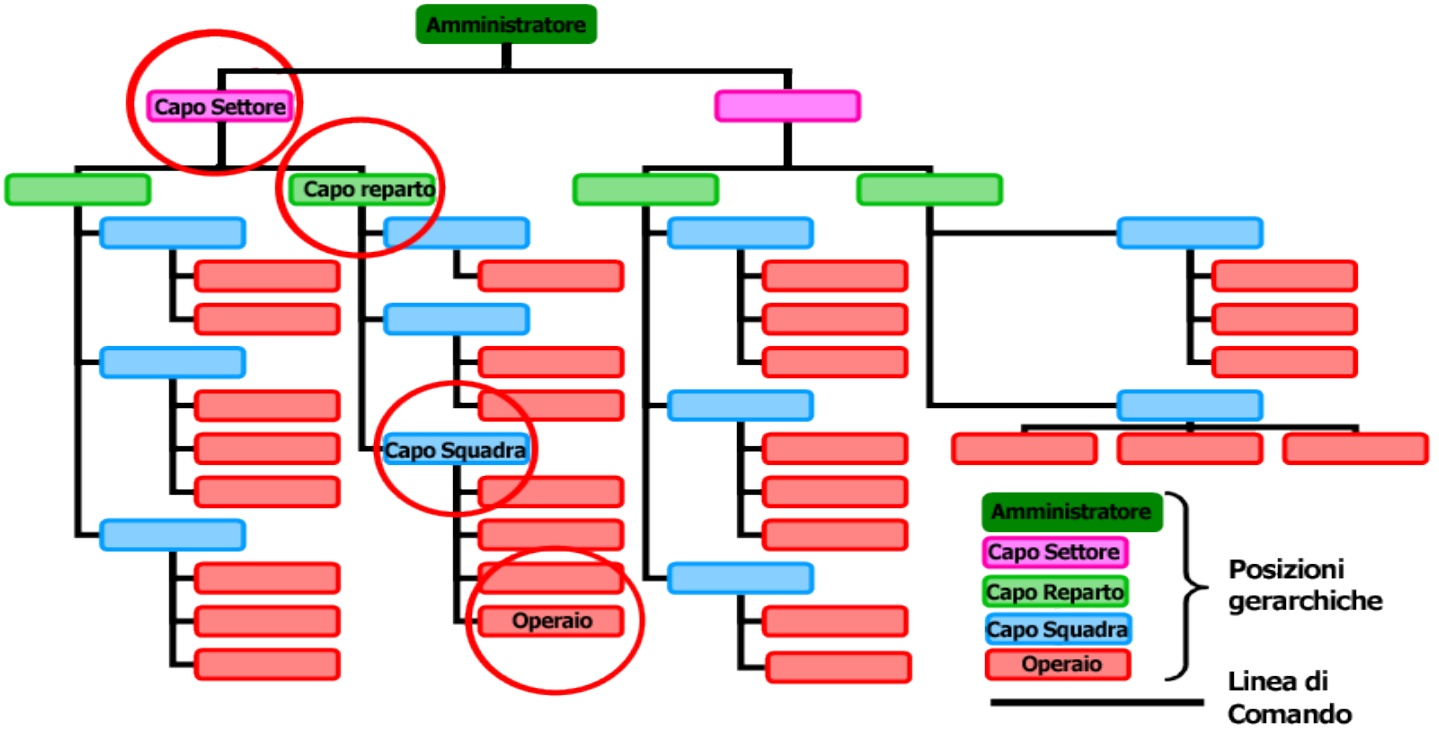
DNS: (ma anche organizzazione territoriale: INDIRIZZI POSTALI)

Www(funzione).comune(ente).pisa(città).toscana(regione).it(paese)



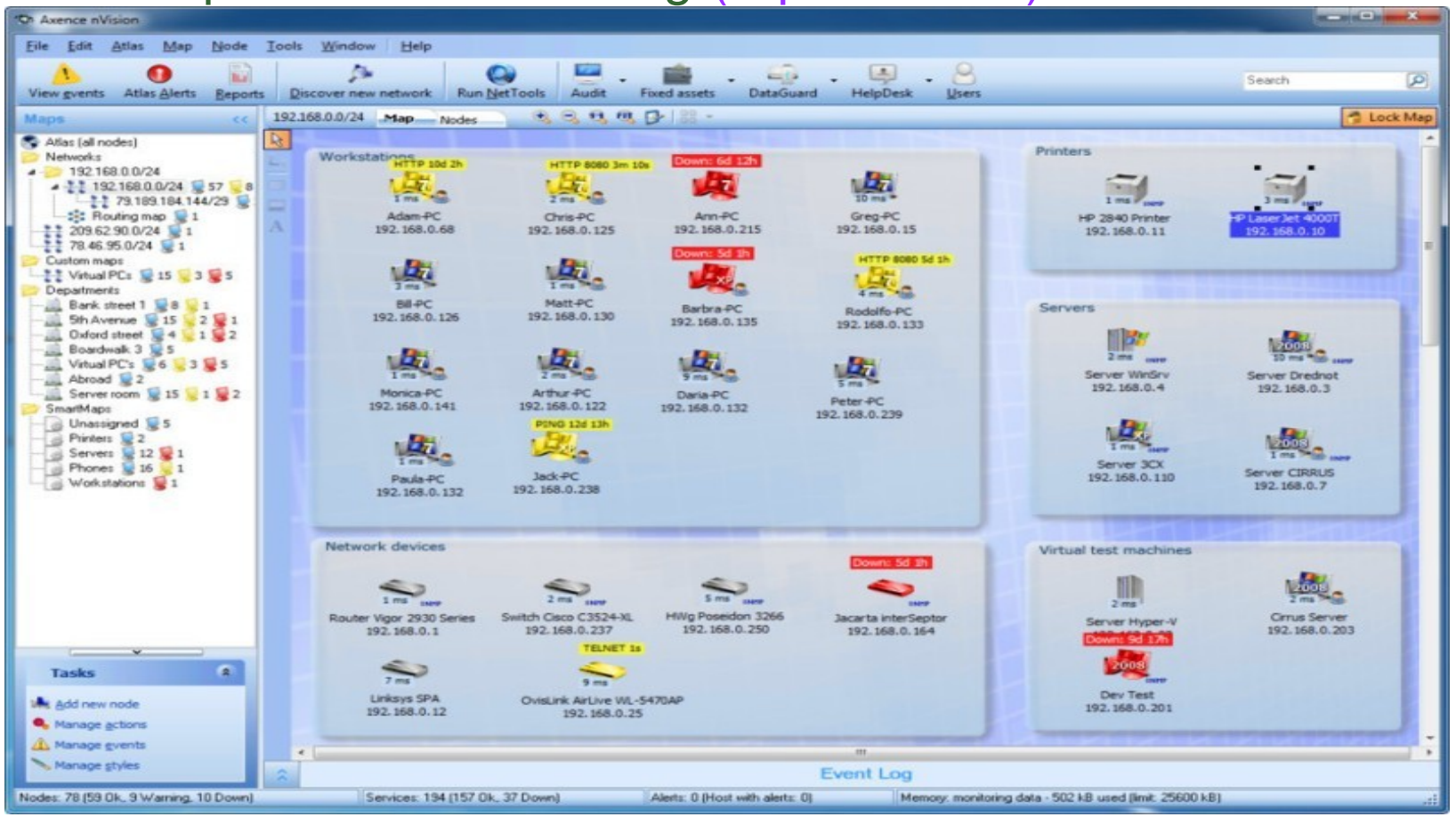
Introduzione e Storia (Organigramma)

Organigramma: “Quello_che_lavora”(RuoloLivello4)->
Capo Squadra(RuoloLivello3)->
Capo Reparto(Ruolo Livello 2)->
Capo Settore(Ruolo Livello 1)->
Amministratore(Ruolo Livello 0)



Introduzione e Storia (Inventario)

Inventario: “Stampante Konica 232”(Dispositivo di rete)->
“Sottorete 10.10.121.0/24”(Sottorete)->
“Dipartimento Marketing”(Dipartimento)



Concetti: DIT e Percorsi

In un server LDAP la struttura ad albero che logicamente rappresenta l'informazione prende il nome di **DIT (Directory Information Tree)**. Specificatamente OpenLDAP utilizza per ogni DIT un distinto database (*non relazionale e non transazionale*), che ha un oggetto radice ed una serie di oggetti *"sottostanti"*, sostanzialmente organizzati come di DNS. Un nodo dell'albero si individua con un percorso, che può essere assoluto o relativo e prende il nome di **DN (Distinguished Name)** se assoluto, **RDN (Relative Distinguished Name)** se anziché partire dalla radice inizia da un nodo specifico dell'albero. I percorsi si costruiscono da sinistra (*parte bassa dell'albero*) a destra (*verso la radice*) citando, separati da virgole, il valore univoco di uno specifico attributo con cui si identifica il nodo foglia e quelli degli oggetti che costituiscono il percorso verso la radice dell'albero.

-->> DN: cn=f.demassis,ou=Users,dc=acme,dc=com (assoluto)
Partendo invece da: **DN: ou=Users,dc=acme,dc=com (sottoalbero)**
-->> RDN: cn=f.demassis (relativo)

Concetti: DIT e Percorsi

dc=com
|
dc=acme
|
ou=Users
|
cn=f.demassis

DN: cn=f.demassis,ou=Users,dc=acme,dc=com

Partendo da: **DN: ou=Users,dc=acme,dc=com**

RDN: cn=f.demassis

Concetti: Attributi e Schema

Si sono involontariamente introdotti i concetti di **attributi** ed **oggetti**.
Ogni **nodo dell'albero** è costituito da **attributi**, ovvero l'unità minimale di informazione paragonabili, per confronto, ad un campo di una tabella di un database relazionale o ad una proprietà di una classe in un linguaggio di programmazione ad oggetti.
Quindi l'attributo costituisce il nome con cui si referenzia un tipo di dato che memorizza l'unità minimale di informazione, che può essere di diversi tipi quali **data**, **testo**, **numero**, etc.
L'oggetto invece rappresenta un insieme di attributi. In generale gli oggetti sono standardizzati e descritti con le loro proprietà semantiche negli **schema**. Questi ultimi in genere sono disponibili con l'installazione del server OpenLDAP o possono essere aggiunti mediante apposite configurazioni ed arricchiscono di oggetti standard con cui costruire alberi DIT il server medesimo.

Concetti: Attributi e Schema

La classe **InetOrgPers**, definita dalla RFC 2798, rappresenta una persona nell'ambito di una azienda/istituzione e contiene le informazioni di contatto quali telefono, e-mail, cellulare, foto, etc. in attributi i cui nomi e tipologia dati sono sempre i medesimi. Può essere utilizzata per definire una rubrica per esempio in un albero gerarchico suddiviso in dipartimenti. Per rappresentare questi ultimi potremmo utilizzare la classe **ou Organizational Unit** definita, insieme ad altri oggetti, nella RFC 4519. Possiamo asserire quindi che **non creiamo oggetti a caso**, ma per lo più utilizziamo oggetti predefiniti, che sono stati standardizzati dai comitati dell' IETF.Il motivo di tale scelta è che le applicazioni che utilizzano il server Ldap sono molteplici, quali server di files, di posta elettronica, di autenticazione dispositivi di rete [radius], server dei Nomi [DNS], di indirizzi [DHCP], di messaggistica testuale [Jabber], etc e tutti si sono fondati su classi comuni eventualmente estese quando necessario specializzando classi più generiche.

Rif. RFC 2798
(https://tools.ietf.org/html/rfc2798)
Rif. RFC 4519
(https://tools.ietf.org/html/rfc4519)

Concetti: Attributi e Schema

Le proprietà di un oggetto o di un insieme di oggetti e a loro semantica sono racchiuse in uno **schema**, ovvero un file di configurazione per il server OpenLDAP che permette di estendere numero e tipologia di oggetti da questo supportati.
Nel file di schema gli oggetti e le proprietà sono classificati con un indicatore univoco numerico organizzato ad albero simile a quello delle MIB del protocollo SNMP.
Il server OpenLDAP di serie viene installato con alcuni schema base. Per aggiungere funzionalità gestibili tramite LDAP occorre installare schema aggiuntivi, in genere mediante un pacchetto della distribuzione, associato al prodotto che si vuole gestire (esempio: *postfix-ldap*).

Rif. samba3.schema
http://www.zytrax.com/books/ldap/ape/samba.html
Rif. Installare samba.schema in Ubuntu 14.04 LTS
https://help.ubuntu.com/lts/serverguide/samba-ldap.html

Database di Controllo e Configurazione

Una volta installato ed inizialmente configurato (*creazione della DIT*) il server OpenLDAP ascolta sulla porta **389/tcp**. Se si usano certificati SSL il server ascolta sulla porta **636/tcp**, anche se ormai tale configurazione è desueta utilizzando TSL sempre sulla porta 389.
Per accedervi con un qualsiasi client LDAP si possono utilizzare i seguenti URL:

ldap://<Indirizzo IP> (implica porta 389)
ldaps://<Indirizzo IP> (implica SSL, porta 636)

Se utilizzate l'accesso locale al database tramite IPC, per esempio se utilizzate una console testuale sul medesimo host quel il comando **ldapmodify**, l'URL di cui servrsi è il seguente:

ldapi:///

Rif. http://www.openldap.org/doc/admin24/runningslapd.html

Database di Controllo e Configurazione

Una volta installato ed inizialmente configurato (*creazione della DIT*) il server OpenLDAP ascolta sulla porta **389/tcp**. Se si usano certificati SSL il server ascolta sulla porta **636/tcp**, anche se ormai tale configurazione è desueta utilizzando TSL sempre sulla porta 389.
Per accedervi con un qualsiasi client LDAP si possono utilizzare i seguenti URL:

ldap://<Indirizzo IP> (implica porta 389)
ldaps://<Indirizzo IP> (implica SSL, porta 636)

Se utilizzate l'accesso locale al database tramite **IPC/Unix Sockets**, per esempio se utilizzate una console testuale sul medesimo host quel il comando **ldapmodify**, l'URL di cui servirsi è il seguente:

ldapi:///

Rif. http://www.openldap.org/doc/admin24/runningslapd.html

Database di Controllo e Configurazione

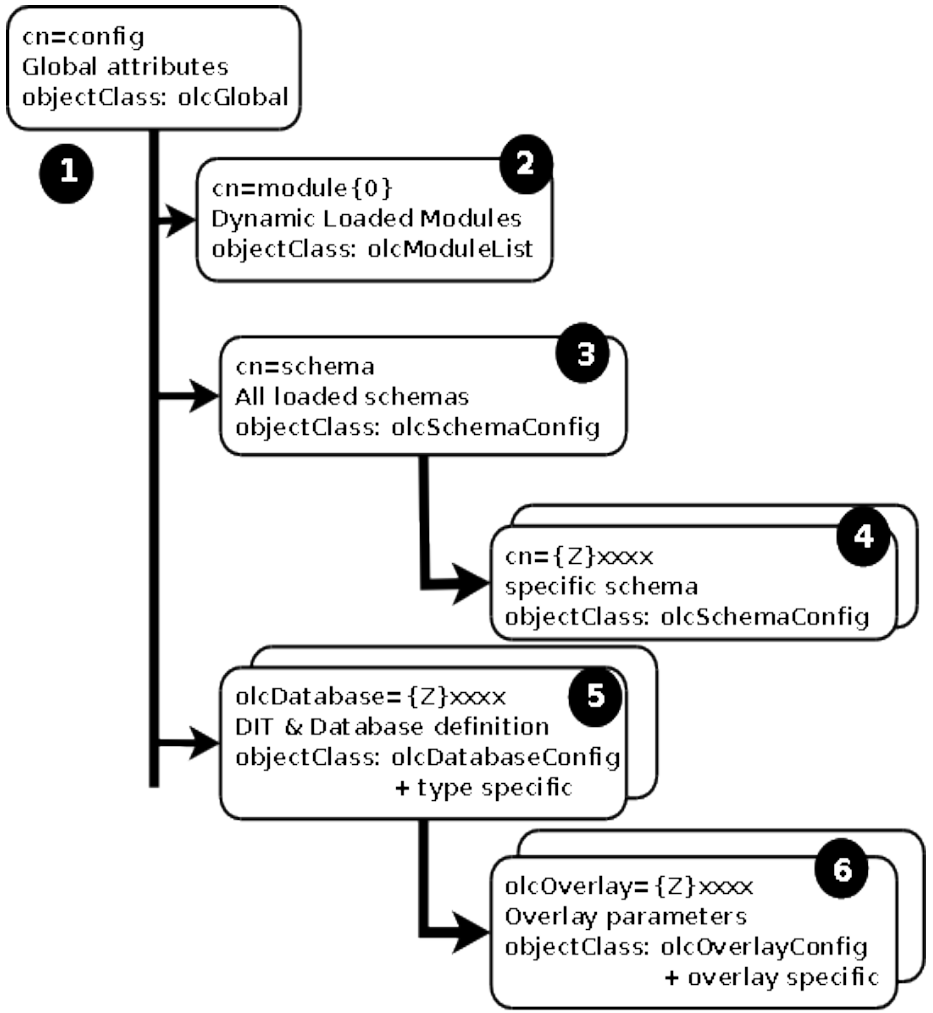
Dalla **Release 2.3 il server LDAP** ha introdotto una **DIT** denominata **cn=config**, che serve a configurare il server stesso.
Il concetto è molto simile al **metadatabase** o **dictionary** presente nei dababase relazionali.
I vantaggi di avere una soluzione in cui il server si configura con gli stessi comandi con cui si modificano i dati è duplice:

>> **uniformità**, il server si configura con gli stessi strumenti con cui si gestiscono i dati (formato **ldif**);

>> **modifiche in tempo reale**, precedentemente alla release indicata il server si configurava con un file **/etc/slapd.conf**, per cui quasi ogni modifica richiedeva il riavvio del server.

La struttura e gli oggetti presenti in tale database sono memorizzati nella cartella **/etc/ldap/slap.d** che contiene un file in formato **ldif** per ogni oggetto.

Database di Controllo e Configurazione



DIT **cn=config**

---------- [1]
cn=config.ldif << file .ldif
cn=config
./cn=config: << cartella
---------- [2]
cn=module{0}.ldif
---------- [3]
cn=schema.ldif
cn=schema
./cn=config/cn=schema ---------- [4]
cn={0}core.ldif
cn={1}cosine.ldif
cn={2}nis.ldif
cn={3}inetorgperson.ldif
cn={4}samba.ldif
---------- [5]
olcDatabase={-1}frontend.ldif
olcDatabase={0}config.ldif
./cn=config/olcDatabase={0}config
olcOverlay={0}syncprov.ldif [6]
---
olcDatabase={1}hdb.ldif
./cn=config/olcDatabase={1}hdb
olcOverlay={0}syncprov.ldif
---
olcDatabase={2}monitor.ldif
---

http://www.zytrax.com/books/ldap/ch6/slapd-config.html

Linguaggio LDIF

Il linguaggio **LDIF** è il linguaggio che descrive (*una delle*) la DIT di un server LDAP. Utilizzando uno degli strumenti in seguito illustrati si può modificare la struttura della DIT aggiungendo o togliendo nodi, modificandone i valori di uno o più nodi. Si consideri che le modifiche non sono transazionali, ovvero non possono essere raggruppate più operazioni in una singola attività atomica, che ha successo o fallisce lasciando tutto inalterato. In genere un server LDAP subisce **poche modifiche/scritture** e **molte letture**, per cui è ottimizzato per tale situazione. Inoltre la struttura dell'albero in genere è **statica**, nel senso che è connessa con la specifica applicazione per cui è utilizzato il server e gli oggetti vengono aggiunti o modificati soltanto nei singoli rami. Infine sempre con il medesimo linguaggio è possibile esprimere delle ricerche su tutta o porzioni della DIT.
Il linguaggio è piuttosto semplice ed ha la seguente struttura:
dn: <percorso completo del nodo>
attr.name: attr.value
Per gli attributi con valori multipli questi vengono specificati di seguito.

Linguaggio LDIF

Esempi di linguaggio **LDIF**:

---

file: cn=config.ldif

dn: cn=config <<< Distinguished Name (1a riga del file)

objectClass: olcGlobal <<< Coppia Nome Attributo/Valore Attributo
cn: config
olcArgsFile: /var/run/slapd/slapd.args
olcPidFile: /var/run/slapd/slapd.pid
olcToolThreads: 1
structuralObjectClass: olcGlobal
entryUUID: 3f9ab81e-53b7-1035-8976-1d54a7d41b7f
creatorsName: cn=config
createTimestamp: 20160120114655Z
olcServerID: 1 ldap://Ldap-1/
olcServerID: 2 ldap://Ldap-2/
olcTLSCACertificateFile: /etc/ssl/certs/cacert.pem
olcTLSCertificateFile: /etc/ssl/certs/ldap-1_slapd_cert.pem
olcTLSCertificateKeyFile: /etc/ssl/private/ldap-1_slapd_key.pem
olcLogLevel: 0

Linguaggio LDIF

Esempi di linguaggio **LDIF**:

---

file: olcDatabase={1}hdb.ldif

dn: **olcDatabase={1}hdb** <<< Distinguished Name (1a riga del file)
objectClass: olcDatabaseConfig <<< Classe da cui deriva l'Oggetto
objectClass: olcHdbConfig <<< Classe da cui deriva l'Oggetto
olcDatabase: {1}hdb <<< Nome del database
olcDbDirectory: /var/lib/ldap <<< Cartella dove sono i file del database

olcDbIndex: objectClass eq <<< Indici sulle proprietà delle classi nella DIT
olcDbIndex: uidNumber eq
olcDbIndex: gidNumber eq
olcDbIndex: cn eq

structuralObjectClass: olcHdbConfig <<< Oggetto *"base"* su cui è costituito l'oggetto attuale
olcSuffix: dc=adf,dc=local <<< Dominio della DIT
olcRootDN: cn=manager,dc=adf,dc=local <<< Utente titolare della DIT
olcAccess: {0}to attrs=userPassword,shadowLastChange by self write by anonymou
s auth by dn="uid=admin,ou=users,dc=adf,dc=local" write by * none
olcAccess: {1}to dn.base="" by * read <<< Access List (Attributo Multivalore)
olcAccess: {2}to * by self write by dn="uid=admin,ou=users,dc=adf,dc=local" wr
ite by * read

Linguaggio LDIF

Esempi di linguaggio **LDIF**:

–----------

Negli esempi precedenti ci sono alcune caratteristiche del server e del linguaggio da approfondire:

objectClass: olcDatabaseConfig **<<< Classe da cui deriva l'Oggetto**
objectClass: olcHdbConfig **<<< Classe da cui deriva l'Oggetto**
structuralObjectClass: olcHdbConfig **<<< Classe che descrive la struttura del nodo**

Negli schema base del **Server LDAP** la classe **olcHdbConfig** che descrive la configurazione del **database ldap basata su backend AltibaseDB**, deriva dalla classe generica **olcDatabaseConfig**, che contiene le proprietà comuni alle configurazioni basate su tutti i possibili backend. Quando si definisce la struttura di un oggetto vanno *"citate"* tutte le classi coinvolte. Effettuando un paragone con un linguaggio ad oggetti è come se **olcHdbConfig** sia una specializzazione di **olcDatabaseConfig.**

Rif. https://en.wikipedia.org/wiki/ALTIBASE_HDB

Linguaggio LDIF

Esempi di linguaggio **LDIF**:

La sezione relativa agli indici invece definisce quali attributi ed oggetti vengono *“indicizzati”* per consentirne un accesso più rapido.

olcDbIndex: objectClass eq **<<< Indici sulle proprietà delle classi nella DIT**
olcDbIndex: uidNumber eq

Questo attributo della configurazione permette di definire quali attributi sono dotati di indice. Una volta definiti gli indici vengono mantenuti automaticamente dal server, a meno che non vengano ridefiniti (*ovvero cambiata la lista di attributi*). In questo ultimo caso va utilizzato il comando **slapindex**, con il servizio spento, per ricostruirli.
La tipologia di confronto è indicata dall'ultimo parametro e può essere: **eq** (equaglianza esatta), **pres** (se si effettuano confronti sul nome degli attributi es: 'objectclass=person' o 'attribute=mail'), **approx** (se si effettuano ricerche 'like'), **sub** (se si cerca una sottostringa nel valore tipo: *search, *search*, search*).
Rif.http://www.zytrax.com/books/ldap/apa/indeces.html

Linguaggio LDIF

Esempi di linguaggio **LDIF**:

Un altro esempio che prendiamo in considerazione è quello relativo all'oggetto di configurazione **olcAccess**:

```
olcAccess: {0}to attrs=userPassword,shadowLastChange by self write by anonymous auth by
dn="uid=admin,ou=users,dc=adf,dc=local" write by * none
olcAccess: {1}to dn.base="" by * read
olcAccess: {2}to * by self write by dn="uid=admin,ou=users,dc=adf,dc=local" write by * read
```

questo rappresenta **le regole per l'accesso alla DIT ed agli attributi** dei singoli oggetti. In generale le modifiche alla DIT sono abilitate per l'utente **olcRootDN**, ma la regola può essere modificata a piacimento. I numeri racchiusi tra parentesi graffe **{n}**, rappresentano il **nr. di riga** in un valore multiriga di un attributo, in questo caso **olcAccess**, in cui il numero di riga implica la sequenza di applicazione della regola.

Rif. http://www.openldap.org/devel/admin/slapdconf2.html

Linguaggio LDIF

Esempi di linguaggio **LDIF**:

La regola costruttiva delle Access List è la seguente:

olcAccess: <access directive>
<access directive> ::= to <what>
[by <who> [<access>] [<control>]]+
<what> ::= * |
[dn[.<basic-style>]=<regex> | dn.<scope-style>=<DN>]
[filter=<ldapfilter>] [attrs=<attrlist>]

Nella **riga 0** del nostro esempio gli attributi: **userPassword,shadowLastChange** sono modificabili dall'utente che ha effettuato il bind (*self*), mentre qualsiasi utente non autenticato può effettuare l'autenticazione (*anonymous*) ed un **dn** (*che rappresenta un utente*) specifico la modifica su tutto l'albero.

Rif. http://www.openldap.org/devel/admin/slapdconf2.html

Linguaggio LDIF

Esempi di linguaggio **LDIF**:

L'ultima componente dell'oggetto **olcHdbConfig** che esaminiamo è quella relativa all'operazione di **replica Multimaster**. La replica multimaster permette di propagare le modifiche alla DIT tra istanze OpenLDAP distinte presenti su server interconnessi. La scrittura può essere effettuata su uno qualsiasi dei nodi e questa viene propagata ai nodi configurati. La configurazione è per database, per cui nel nostro caso stiamo trattando il database nr. 1, ma anche il database nr. 0 (*quello della configurazione cn=config*), può essere configurato allo stesso modo, per cui da quel momento in poi le modifiche alla configurazione vengono automaticamente condivise tra tutti i nodi.
La replica multimaster è un meccanismo applicativo interno al server OpenLDAP, che ha più di una modalità di replica.
Tale meccanismo può essere utilizzato per avere istanze ridondate equivalenti, in modo da garantire la continuità del servizio.

Rif. http://www.openldap.org/devel/admin/slapdconf2.html

Linguaggio LDIF

Esempi di linguaggio **LDIF**:

```
olcMirrorMode: TRUE

olcSyncrepl: {0}rid=003 provider=ldap://Ldap-1/ binddn="cn=manager,dc=adf,dc=local"
  bindmethod=simple credentials=****** searchbase="dc=adf,dc=local"
  type=refreshAndPersist retry="5 5 300 5" timeout=1

olcSyncrepl: {1}rid=004 provider=ldap://Ldap-2/ binddn="cn=manager,dc=adf,dc=local"
   bindmethod=simple credentials=****** searchbase="dc=adf,dc=local"
   type=refreshAndPersist retry="5 5 300 5" timeout=1
```

Anche l'oggetto **olcSyncrepl** è multivalore. Ogni riga rappresenta una connessione di replica. In ogni riga sono specificati:
il nr. della connessione: **rid**, l'URL **del servizio**: ldap://Ldap-1 (*Ldap-1 = nome host*), il **dn dell'utente** possessore della DIT e la relativa password, il punto iniziale da cui effettuare le ricerche degli oggetti modificati sotto forma di dn, altri parametri quali il tipo di allineamento, i time-out, etc.

Rif. http://www.openldap.org/devel/admin/slapdconf2.html

Strumenti

Distinguiamo **strumenti generici** da **strumenti custom**. Quelli generici costituiscono un mezzo di interazione con il server LDAP per fornire comandi **in formato LDIF** con i quali configurare, caricare e modificare i **contenuti della DIT**. Potrebbero essere equiparati ai tools più diffusi dei database relazionali, quali console testuali o grafiche/WEB.
Definiremo **strumenti Custom** quelli relativi ad una specifica applicazione, quale la gestione degli utenti. In questo caso OpenLDAP rappresenta semplicemente il **“database non relazionale”** dove salvare i dati necessari a diversi sistemi per recuperare gli utenti. Le interfacce in questo caso sono mirate a fornire all'utilizzatore i controlli e le validazioni necessarie allo scopo. E' evidente che un tool generico può essere utilizzato anche in contesti relativi ad una specifica applicazione, ma ciò è sconsigliato in quanto l'operatore dovrebbe avere un minimo di cognizioni relative all funzionamento del server.

Strumenti - Testuali

Gli strumenti testuali più importanti sono parte dei pacchetti **slapd, ldaputils** sulle distribuzioni Debian Based. Tali utilità sono una serie di script mediante i quali è possibile modificare la **DIT** del server.

Il più semplice è **slapcat**, che a servizio spento permette di estrarre il contenuto della **DIT** in formato LDIF.

Quindi i comandi seguenti effettuano un backup **LDIF** della **DIT**.
(*si presuppone di essere utente root. In alternativa si può utilizzare sudo*)

#service slapd stop
#slapcat > /var/tmp/DIT.ldif
#service slapd start

ldapmodify/ldapadd permettono di intervenire sulla DIT.

Strumenti - Testuali

Esempio di modifica in linea del valore **olcServerID** utilizzando l'accesso IPC sulla medesima macchina dove risiede il server **OpenLDAP**:

```
cat <<EOF | ldapmodify -Y EXTERNAL -H ldapi:///
dn: cn=config
changetype: modify
add: olcServerID
olcServerID: 1
EOF
```

-Y = tipologia di autenticazione SASL, mediante l'utente del sistema (EXTERNAL)

Utilizzando il file **olcServerID.ldif** con il seguente contenuto:

```
dn: cn=config
changetype: modify
add: olcServerID
olcServerID: 1
-
```

```
ldapmodify -Y EXTERNAL -H ldapi:/// -f olcServerID.ldif
```

Se le entry fossero più di una nel file sarebbero separate dal simbolo -.

Strumenti - Grafici

Se si vuole un accesso client-server con una console grafica il miglior tool OpenSource è **Apache Directory Studio**

http://directory.apache.org/studio/

L'installazione richiede il pacchetto **Java Runtime**, preferibilmente la versione **Oracle**, installabile nelle varie distribuzioni mediante alcuni artifici. (*dato che Oracle ha rimosso la distribuzione mediante i Repository Linux per problemi di licenze*)

Avendo Java funzionante (*test: java -version in un terminale*)

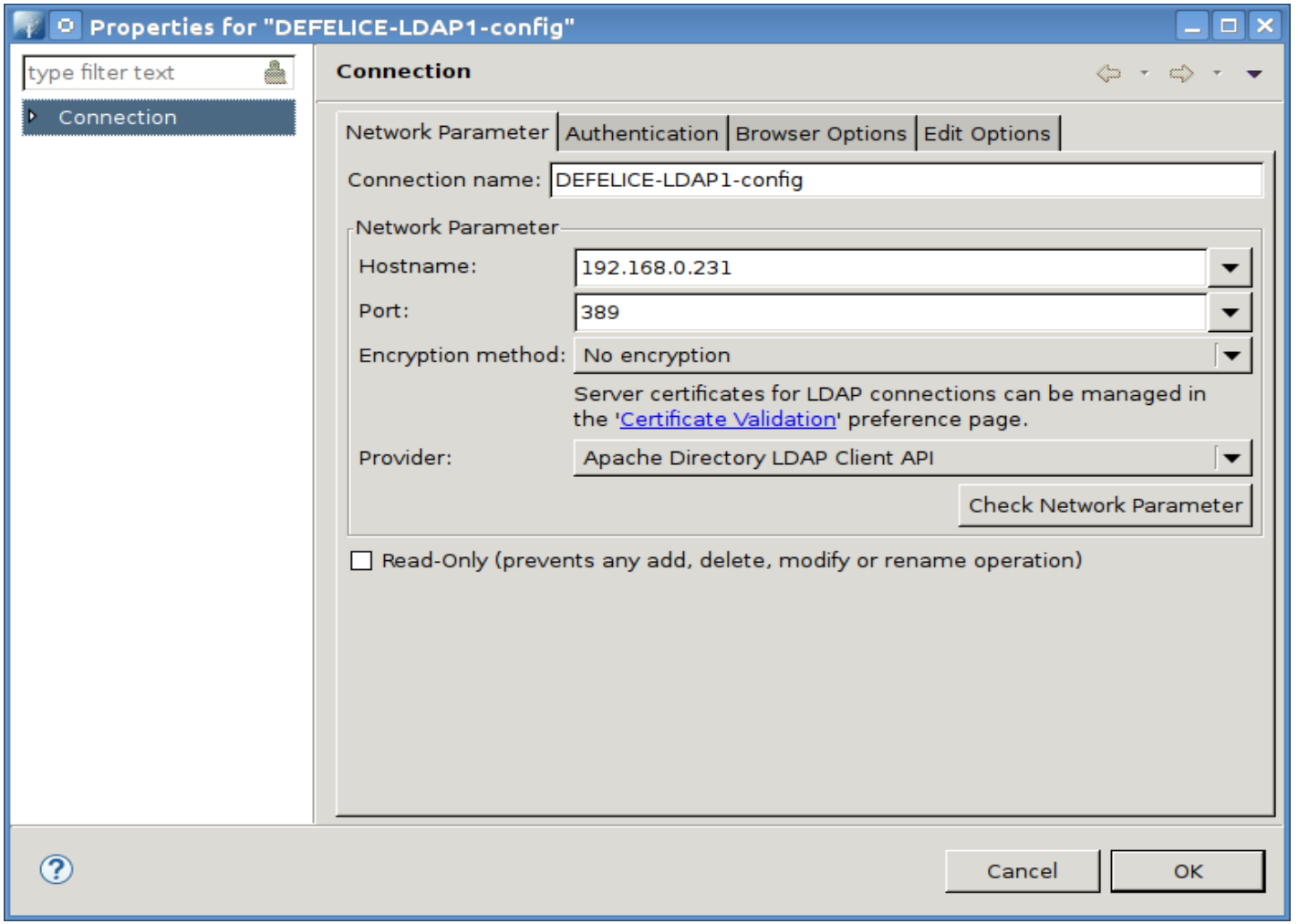
```
demassis@Micro-1:~$ java -version
java version "1.8.0_91"
Java(TM) SE Runtime Environment (build 1.8.0_91-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.91-b14, mixed mode)
```

Strumenti - Grafici

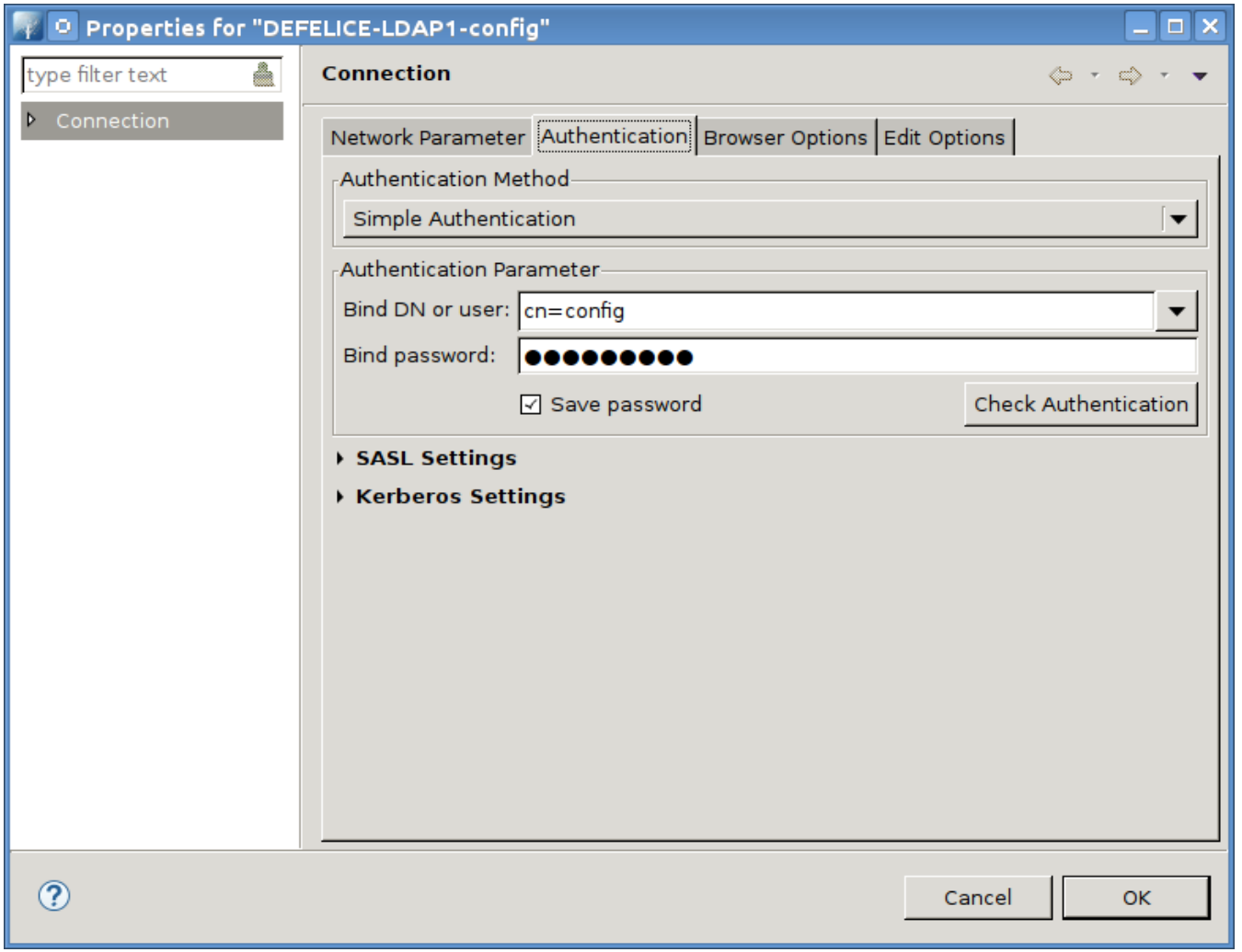
Per installare lo strumento basta scompattare il file in una cartella ed eseguire il file: **ApacheDirectoryStudio**, eventualmente configurando un link o una entry del menu nell'ambiente grafico Linux preferito (Gnome o KDE). Seguono nelle Slides seguenti le istantanee della configurazione.

1) Connessione (indirizzo IP del server/Porta)
2) Autenticazione (utente possessore della DIT e relativa password)
3) vari dettagli dell'Editor LDAP
4) uso di una connessione sul database nr. 0 di configurazione
5) esempio di un database per l'autenticazione

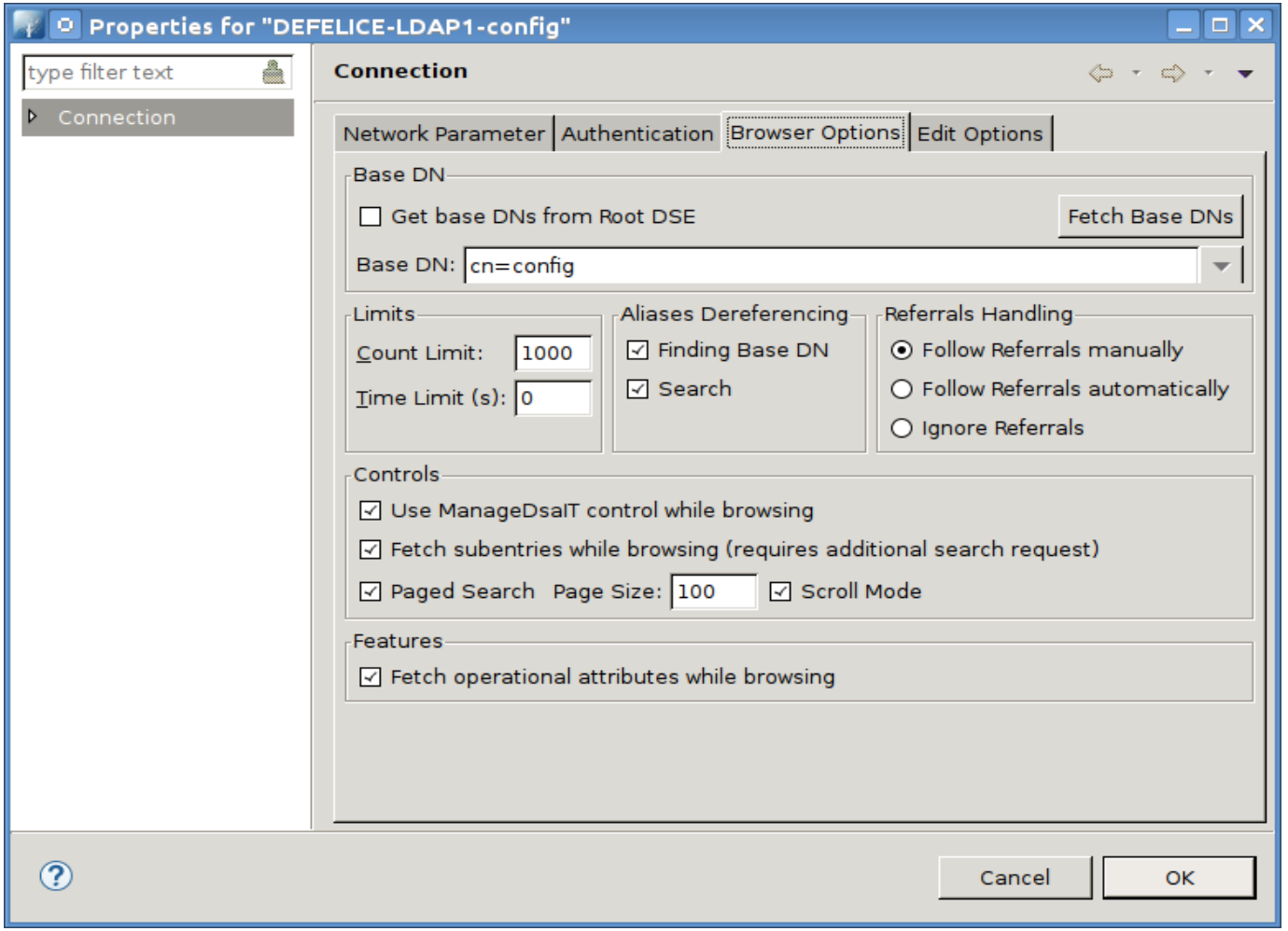
Strumenti - Grafici



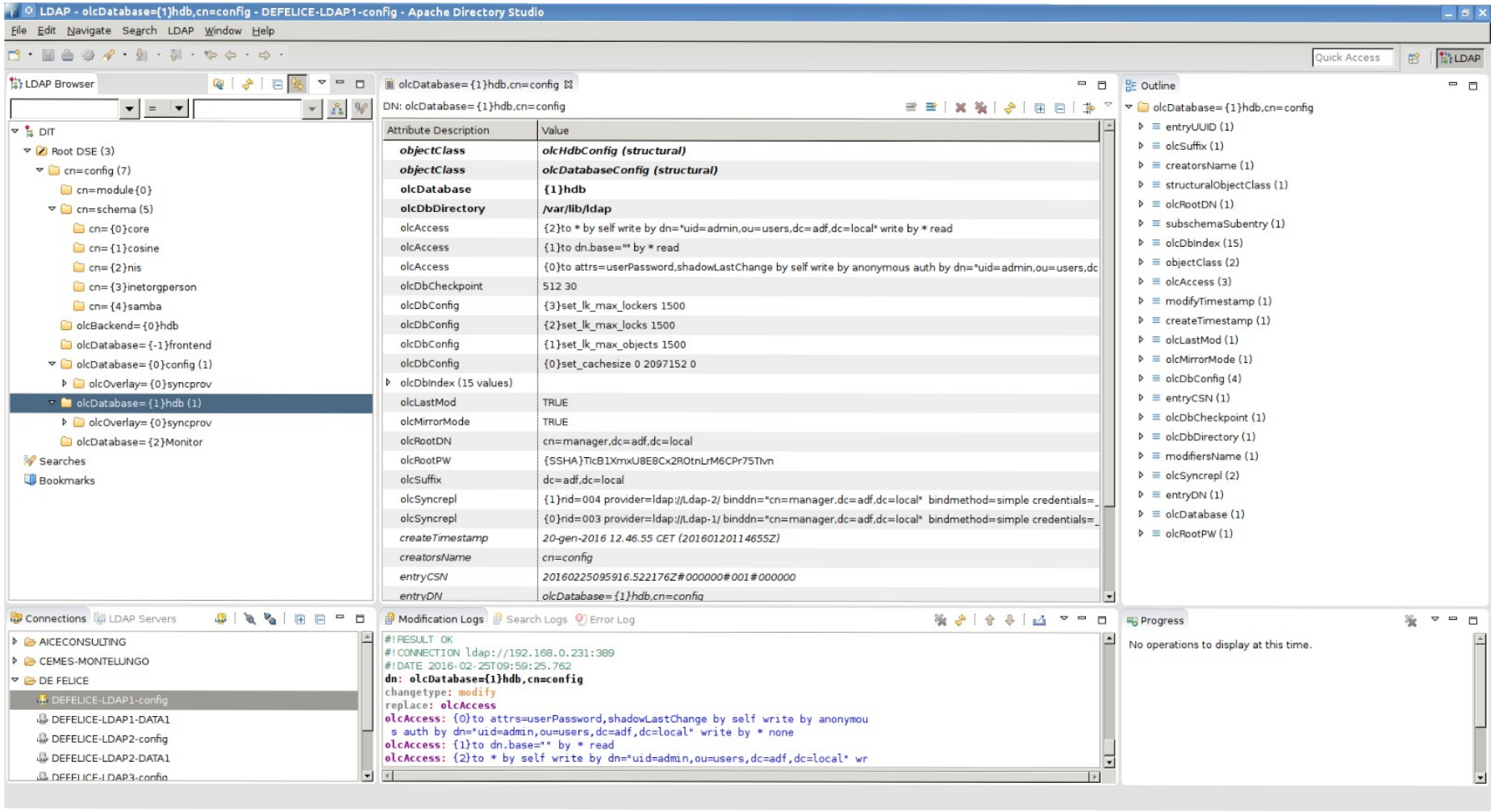
Strumenti - Grafici



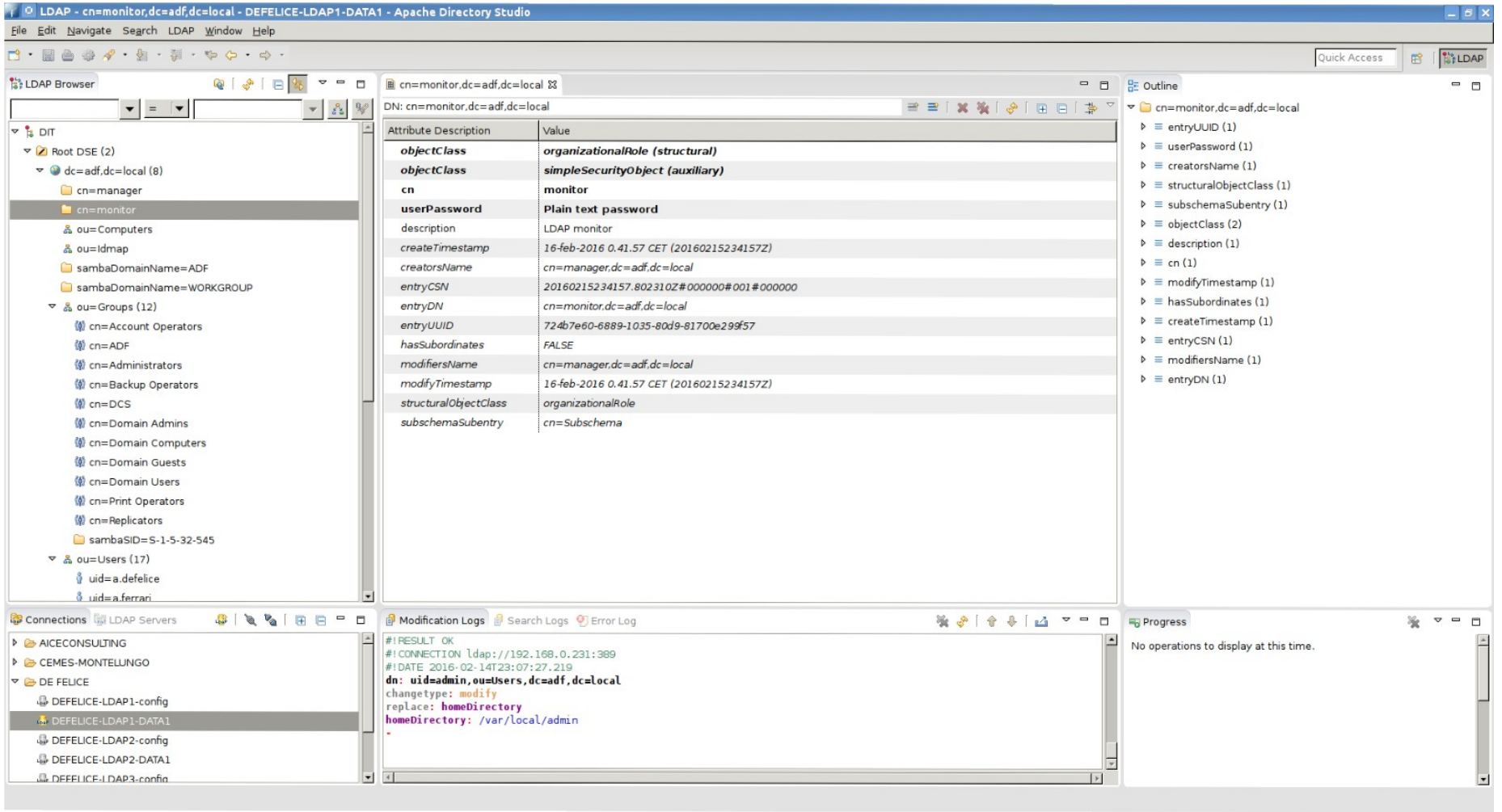
Strumenti - Grafici



Strumenti - Grafici

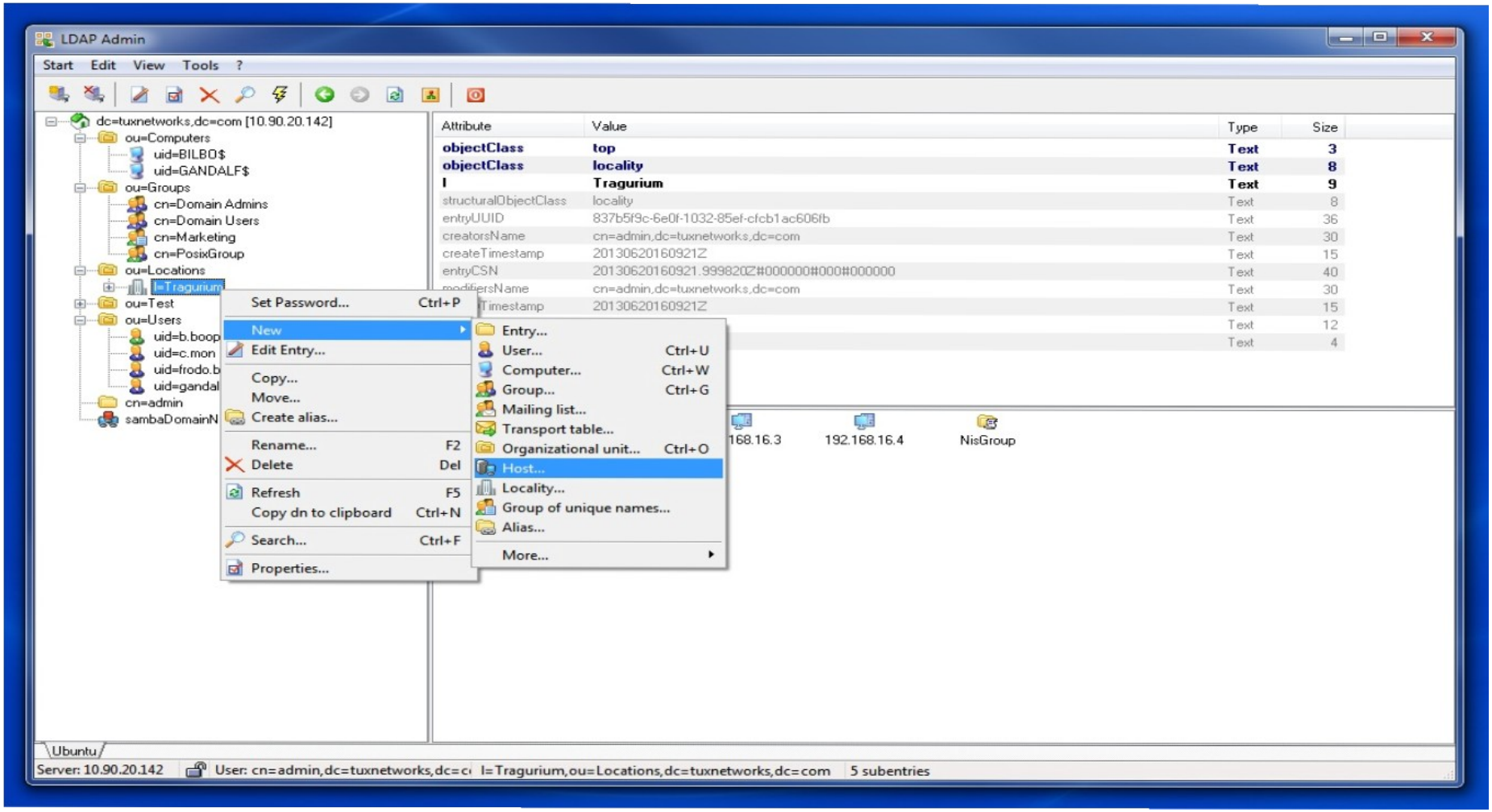


Strumenti - Grafici



Strumenti – WEB phpLdapAdmin

http://phpldapadmin.sourceforge.net/wiki/index.php/Main_Page



Strumenti – WEB Ldap Account Manager

https://www.ldap-account-manager.org/lamcms/