

Architettura Sicurezza - HOWTO aggiunta di un nuovo utente

L'architettura.

L'infrastruttura di griglia EDG (*European Data Grid*) si basa sul *GLOBUS Toolkit 2.1*, il cui meccanismo di sicurezza è basato su **certificati** e si chiama GSI (*Grid Security Infrastructure*).

Ogni utente di griglia deve avere un certificato, il cui *certificate subject* lo identifichi.

Ciascuna macchina che partecipa alla griglia mettendo a disposizione risorse di calcolo e spazio disco, mantiene un file con una lista di corrispondenze tra *certificate subject* e nomi *d'utente locale*. Affinché un utente possa utilizzare le risorse di una macchina, l'amministratore di quella macchina dovrà richiedergli il *certificate subject* e includerlo nella lista facendolo corrispondere ad un utente locale esistente.

Quando un utente esegue un comando di griglia, tale comando invierà prima una copia del certificato, quindi la macchina di destinazione verificherà se il *certificate subject* è in corrispondenza con qualche utente locale. In caso non lo sia, l'azione richiesta non verrà eseguita; altrimenti si procede.

Tutti i programmi e servizi di griglia *GLOBUS* sono compilati con le librerie C di GSI: è il meccanismo base di sicurezza intrinseco in tutti i programmi *GLOBUS*.

Nella pratica:

I programmi che l'utente impiega per utilizzare la griglia sono raccolti e configurati in una macchina che prende il nome di *UI User Interface*.

Ciascun utente ha un account nella UI; nella propria *home* esiste la directory *.globus* che contiene il certificato (file *usercert.pem*) e la chiave privata che viene generata assieme al certificato (file *userkey.pem*). Il certificato è di tipo X.509.

In ciascuna macchina della griglia esiste la directory */etc/grid-security* che contiene il file *grid-mapfile*: è il file con le corrispondenze tra *certificate subject* e nome *utente locale*. Tutti i servizi e programmi di griglia leggono le informazioni di autenticazione da questo file.

Il sistemista della macchina che eroga servizi di griglia deve provvedere alla creazione dei rispettivi utenti locali: possono essere utenti qualsiasi con nomi qualsiasi! **Non** contengono i certificati! Sono **completamente separati, distaccati e trasparenti** alla griglia!

Esempio di *grid-mapfile*:

```
"/C=IT/L=Trieste - Italy/O=EGRID/OU=ICTP/CN=Adam Ponzi" aponzi
```

```
"/C=IT/L=Trieste - Italy/O=EGRID/OU=ICTP/CN=Alessandra Tedeschi" atedeschi
```

```
"/C=IT/L=Trieste - Italy/O=EGRID/OU=ICTP/CN=Alessio Terpin" aterpin
```

```
"/C=IT/L=Trieste - Italy/O=EGRID/OU=ICTP/CN=Alvise Nobile" anobile
```

"/C=IT/L=Trieste - Italy/O=EGRID/OU=ICTP/CN=Angelo Leto" aleto
"/C=IT/L=Trieste - Italy/O=EGRID/OU=ICTP/CN=Clement Onime" conime
"/C=IT/L=Trieste - Italy/O=EGRID/OU=ICTP/CN=Cristian T. Brownlees" ctbrownlees
"/C=IT/L=Trieste - Italy/O=EGRID/OU=ICTP/CN=Cristian Zoicas" zoicas

Documento di riferimento per chiavi pubbliche, certificati e GSI: *IBM Red Book
Introduction to Grid Computing with Globus* presso
<http://www.redbooks.ibm.com/redbooks/pdfs/sg246895.pdf>

EDG - Primo salto d'architettura

Dal punto di vista dei sistemisti è molto pesante gestire la corrispondenza uno-a-uno tra *certificate subject* e nome *utente locale*, perché ad ogni nuovo utente di griglia a cui si vuole dare accesso alle risorse della propria macchina bisogna creare un nuovo utente locale. Si può utilizzare un unico *nome utente* per tutti i *certificate subject*, ma vi sono problemi di accesso concorrenziale da parte di più utenti.

L'idea è di utilizzare un meccanismo di *pool account*: avere un insieme di utenti locali che a rotazione vengono associati al *certificate subject* presentato al momento della richiesta di calcolo o spazio disco.

In questo modo si elimina la necessità al sistemista della macchina in griglia di aggiungere fisicamente un utente nuovo. Basta aggiungere una riga al file delle corrispondenze, con il nuovo *certificate subject*.

Nella pratica

EDG ha apportato un patch alle librerie C del GSI per avere i pool-account.

Il patch è tale per cui nel *grid-mapfile* basta far corrispondere **ciascun** *certificate subject* ad un **medesimo** nome *utente locale* con la particolarità che questo inizi con un “.”; per esempio si ha il seguente file:

```
"/C=IT/L=Trieste - Italy/O=EGRID/OU=ICTP/CN=Adam Ponzi" .gridit  
"/C=IT/L=Trieste - Italy/O=EGRID/OU=ICTP/CN=Alessandra Tedeschi" .gridit  
"/C=IT/L=Trieste - Italy/O=EGRID/OU=ICTP/CN=Alessio Terpin" .gridit  
"/C=IT/L=Trieste - Italy/O=EGRID/OU=ICTP/CN=Alvise Nobile" .gridit  
"/C=IT/L=Trieste - Italy/O=EGRID/OU=ICTP/CN=Angelo Leto" .gridit  
"/C=IT/L=Trieste - Italy/O=EGRID/OU=ICTP/CN=Clement Onime" .gridit  
"/C=IT/L=Trieste - Italy/O=EGRID/OU=ICTP/CN=Cristian T. Brownlees" .gridit  
"/C=IT/L=Trieste - Italy/O=EGRID/OU=ICTP/CN=Cristian Zoicas" .gridit
```

Nella macchina di griglia l'amministratore dovrà semplicemente creare più utenti fisici il cui nome inizia con quello utilizzato nel *grid-mapfile*, omettendo il “.” iniziale, e accodando dei numeri. Complessivamente gli utenti fisici hanno nomi con numeri in sequenza:

```
gridit001  
gridit002  
gridit003
```

La forma del suffisso numerico è quella illustrata. Non ci sono alternative.

In */etc/grid-security* deve essere creata una directory *grid-mapdir* che contiene per ciascun utente fisico del *pool account*, un file **vuoto** con lo stesso nome dell'utente fisico.

Il numero di utenti fisici può essere limitato a solo una manciata: fin tanto che ci sono utenti liberi verranno accettate nuove richieste di servizi di griglia. Esauriti gli utenti liberi ogni richiesta ulteriore viene respinta.

Il meccanismo d'associazione

Quando una richiesta di calcolo o di spazio disco arriva, se il *certificate subject* è presente nel *grid-mapfile* e vi sono utenti liberi nel *pool account*, allora ne viene scelto uno. Successivamente nel *grid-mapdir* viene **creato un hardlink** con nome quello del *certificate subject* richiedente il servizio, e puntante verso uno dei file vuoti: il nome del file vuoto coincide con quello dell'utente prescelto dal *pool account*.

Per vedere i hardlink e i file a cui sono associati basta eseguire il seguente comando in */etc/grid-security/grid-mapdir/*:

```
ls -l | sort
```

Per liberare utenti del *pool account*, basta cancellare il hardlink. Un processo esterno quale un **cron job** periodicamente libera gli account: in questo modo l'utente fisico torna a far parte delle risorse da utilizzare.

Nota

In EDG esiste il concetto di VO *Virtual Organization*: l'insieme delle persone che appartengono al medesimo gruppo di lavoro. Una VO è identificata semplicemente da un nome: non è definita esplicitamente da nessuna parte per cui non c'è una lista di utenti che vi appartengono. Quando viene erogato un certificato per un utente nuovo, nel certificato si specifica la VO di appartenenza.

Per semplificare la gestione, il nome scelto per il pool account è proprio quello della VO.

Documenti di riferimento

Patch per *GLOBUS*: <http://www.gridpp.ac.uk/gridmapdir>

EDG - Secondo salto d'architettura

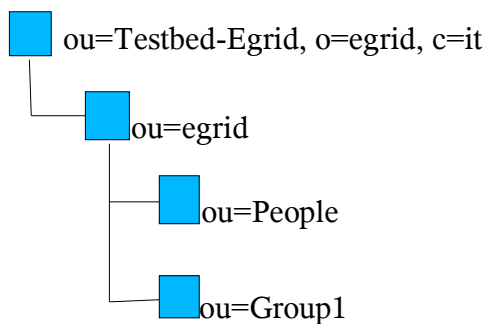
Il secondo problema amministrativo pesante riguarda l'aggiunta di *Certificate Subject* di utenti nuovi in ciascun *grid-mapfile* delle macchine facenti parte della VO.

L'idea è di avere un server LDAP con le liste di *Certificate Subject*, e tramite un processo esterno quale un cron job eseguito su ciascuna macchina della VO, aggiornare periodicamente i *grid-mapfile*.

Nella pratica

In EDG il meccanismo appena descritto prende il nome di *VOServer*.

Ramo LDAP contenete i *Certificate Subject*:



Il *VOServer* ha due rami che finiscono con:

ou=People
ou=Group1

Script che aggiorna il grid-mapfile:

Lo script che legge il *VOServer* e aggiorna il *grid-mapfile* è `/opt/edg/sbin/edg-mkgridmap`.

Lo script viene eseguito da un cron job ogni 6 ore; il job è dell'utente *root* per cui si può visualizzare da utente *root* con:

```
crontab -l
```

Ramo che finisce con ou=Group1:

È il ramo utilizzato dallo script `/opt/edg/sbin/edg-mkgridmap` per riempire il *grid-mapfile*.

Esempio di file LDIF che descrive il contenuto:

```
dn: ou=Group1, ou=egrid, ou=Testbed-Egrid,o=egrid,c=it
owner: cn=Manager,ou=Group1,ou=egrid,ou=Testbed-Egrid,o=egrid,c=it
ou: Group1
objectClass: groupofnames
cn: Group1
member: cn=Alessandra Tedeschi,ou=People,ou=egrid,ou=Testbed-Egrid,o=egrid,c=it
member: cn=Alvise Nobile,ou=People,ou=egrid,ou=Testbed-Egrid,o=egrid,c=it
member: cn=Alessio Terpin,ou=People,ou=egrid,ou=Testbed-Egrid,o=egrid,c=it
```

...

In particolare è importante la riga *member* che contiene il *Certificate Subject*.

Ramo che finisce con ou=People:

Ramo che mantiene informazioni sulle persone, e che viene utilizzato per operazioni di filtraggio: cioè c'è la possibilità di impostare una lista di allow/deny *Certificate Subject*.

Le informazioni in questo ramo vengono utilizzate dallo script `/opt/edg/sbin/edg-mkgridmap` durante l'interrogazione del *VO Server*, per filtrare i *Certificate Subject*. L'opzione di filtraggio è impostata in `/opt/edg/etc/edg-mkgridmap.conf`.

Esempio di file LDIF che descrive il contenuto:

```
dn: cn=Ezio Corso,ou=People,ou=egrid,ou=Testbed-Egrid,o=egrid,c=it
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: pkiUser
cn: Ezio Corso
sn: Corso
description: subject= /C=IT/L=Trieste - Italy/O=EGRID/OU=ICTP/CN=Ezio Corso
```

(Su `egrid-1.egrid.it:/root/EGRID/Egrid_CA/Certificate/Users/ldif_files/` esistono copie di sicurezza dei file LDIF)

Come funziona il meccanismo

`edg-mkgridmap` prende i *Certificate Subject* dal *VO Server*; esegue l'operazione di filtraggio se impostata; infine prende il file `/opt/edg/etc/grid-mapfile-local`, e accoda il contenuto sul *grid-mapfile* creato, sopra scrivendo le entrate già presenti.

Attenzione! Group presente nel LDAP nostro contiene commenti che sono fuorvianti!!! Non e' come nel commento!

Documenti di riferimento

VO Server e `edg-mkgridmap`:

<http://grid-deployment.web.cern.ch/grid-deployment/gis/release-docs/GMF-index.html>;

LDAP: *IBM Red Book Understanding LDAP - Design and Implementation*

<http://www.redbooks.ibm.com/redbooks/pdfs/sg244986.pdf>; primi 3 capitoli

Ulteriori note su LDAP:

1. Ogni entry LDAP deve avere un *DN Distinguished Name* che l'identifica univocamente, e almeno un *objectClass* che stabilisce quali sono gli attributi per quel *DN*. Gli attributi sono di due tipi: obbligatori e facoltativi.
2. *objectClass person* richiede il *sn* (surname) ed il *cn* (common name) come attributi obbligatori.

EGRID – Salto d'architettura

L'ultimo grosso problema amministrativo che rimane riguarda la sicurezza dei file.

L'idea è di modificare il *grid-mapfile* solo del SE in modo che il mapping non avvenga più su *pool account*, ma su utenti specifici. A questi utenti specifici locali vengono impostati i diritti sui file in modo accurato e dettagliato, con la classica terna di permessi Unix. Questo risolve il problema della sicurezza dei dati.

Per ottenere sul SE il mapping giusto, sfruttiamo il *grid-mapfile-local* del *VOServer*: inserendovi le corrispondenze tra utente e *Certificate Subject* che vogliamo – ogni volta che viene eseguito *edg-mkgridmap* verranno accodati/sovrascritti i nostri utenti.

Per automatizzare la distribuzione del *grid-mapfile-local*, utilizziamo un server LDAP con un nostro schema che raccoglie le corrispondenze *Certificate Subject* e nome *utente locale*, che poi un nostro script andrà a leggere e quindi aggiornare il *grid-mapfile-local*.

Per snellire ulteriormente l'amministrazione del SE, invece di dover creare utenti fisici nuovi ogni volta che un nuovo utente si aggiunge alla VO (problema originariamente risolto con i *pool account* di cui però qui dobbiamo farne a meno), impostiamo l'autenticazione utenti Unix del SE tramite LDAP. Per cui la macchina del SE chiederà */etc/passwd* e */etc/group* al server LDAP: basterà aggiungere gli utenti in LDAP per avere aggiornata la lista di utenti con accesso al SE. Le schema per l'autenticazione tramite LDAP prende il nome di *objectClass POSIX*.

Aggiungeremo alle *objectClass POSIX* **anche** il riferimento al *Certificate Subject*, onde evitare di dover mantenere due Server LDAP. Il nostro script interrogherà questo stesso Server LDAP e costruirà l'opportuno *grid-mapfile-local*.

Complessivamente quando un nuovo utente della VO arriva, basta aggiungere una nuova entry in questo LDAP Server: d'un colpo solo verranno gestiti sia l'utente fisico che il *grid-mapfile-local*.

Nella pratica

Script che aggiorna il *grid-mapfile-local*:

Si trova in */usr/local/sbin/ldap2gridmap.pl* di ciascun SE dove l'accesso ai dati è delicato.

Un cron job lo esegue regolarmente; il cron job si trova in */etc/cron.d/ldap2gridmap.cron*:

```
*/* * * * root /usr/local/sbin/ldap2gridmap.pl egrid-1 10389 ou=people,ou=egrid,ou=testbed-egrid,o=egrid,c=it > /opt/edg/etc/grid-mapfile-local
```

Ramo LDAP:

Siccome le *objectClass POSIX* per l'autenticazione LDAP vanno messe su un ramo *Group* per quanto riguarda i dati del file */etc/group* e in un ramo *People* per quanto riguarda i dati del file */etc/passwd*, sfruttiamo il ramo *People* del *VOServer* stesso e aggiungiamo gli attributi necessari.

Un esempio di file LDIF del *VOServer* **modificato** in questo modo è, per *People*:

```
dn: cn=Ezio Corso, ou=People, ou=egrid, ou=Testbed-Egrid,o=egrid,c=it
shadowMin: -1
sn: Corso
userPassword:: e1NIQX0rZmt2Q3hXUytGb2J3YW00Ri9SRXJqbm84UVk9
loginShell: /bin/bash
uidNumber: 2008
gidNumber: 2008
shadowFlag: 0
shadowExpire: -1
shadowMax: 99999
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: pkiUser
objectClass: posixAccount
objectClass: shadowAccount
objectClass: top
uid: ecorso
shadowLastChange: 11192
cn: Ezio Corso
homeDirectory: /home/egrid/ecorso
shadowInactive: -1
description: subject= /C=IT/L=Trieste - Italy/O=EGRID/OU=ICTP/CN=Ezio Corso
shadowWarning: 7
```

Il nostro script *ldap2gridmap.pl* legge il *Certificate Subject* dall'attributo *description*.

Per *Group* abbiamo:

```
dn: cn=ecorso, ou=Group, ou=egrid, ou=Testbed-Egrid,o=egrid,c=it
gidNumber: 2008
objectClass: posixGroup
objectClass: top
cn: ecorso
```

Documenti di riferimento

RFC Parte 1: <http://www.egrid.it>

RFC Parte 2: <http://www.egrid.it>

HOWTO aggiungere un utente

Creazione del certificato utente

Noi abbiamo messo sù una CA in *egrid-1.egrid.it*. Gli script per la creazione dei certificati si trovano presso */root/EGRID/Egrid_CA*.

1. Eseguire lo script *./User.sh* e compilare le informazioni richieste a schermo. Al termine viene creata la *private key* nel file *userkey.pem* ed il *certificato* nel file *usercert.pem*, ambedue in *./tmp*. Il file *usercert.pem* contiene il *Certificate Subject* con le informazioni dell'utente quali nome e cognome, organizzazione, Virtual Organization, ecc.
2. Eseguire lo script *./User_sign.sh* per firmare il certificato; viene creato il file *userreq.pem* in *./tmp*.
3. Creare una directory in *./Certificate/Users/* col nome dell'utente, e copiarvi i tre file *.pem*. Questo viene fatto solo per organizzazione nostra: in caso di perdita di una macchina abbiamo un deposito dove andare a recuperare i certificati.

Per esempio creare un certificato per l'utente Ezio Corso Prova e firmarlo; file *usercert.pem*:

Certificate:

Data:

Version: 3 (0x2)

Serial Number: 55 (0x37)

Signature Algorithm: md5WithRSAEncryption

Issuer: C=IT, O=EGRID, CN=Egrid CA Test

Validity

Not Before: Jul 15 14:21:10 2004 GMT

Not After : Jul 15 14:21:10 2005 GMT

Subject: C=IT, L=Trieste - Italy, O=EGRID, OU=ICTP, CN=Ezio Corso Prova

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public Key: (2048 bit)

Modulus (2048 bit):

```
00:ca:df:c1:1f:9a:c3:ac:f6:a2:9e:88:34:6e:b3:
2e:0a:e9:b8:9f:60:9e:93:00:c3:db:82:9c:40:b9:
a6:1e:30:29:10:42:e1:1b:fd:3c:dc:ea:d5:25:fd:
d0:1d:81:83:43:e3:19:34:56:c7:0a:9d:10:00:48:
6d:58:85:5c:23:66:bd:2b:6b:58:c6:2c:1b:06:7e:
b9:e7:5b:0b:72:71:74:bf:28:26:f7:02:b8:3d:62:
9a:e6:f4:c4:89:c0:24:22:d8:e2:7e:45:68:db:c2:
b2:eb:b1:be:32:cb:71:ad:73:26:99:4a:71:3b:68:
d9:81:7d:f7:26:cb:50:18:17:d1:02:a7:0f:48:7e:
af:e2:34:99:ac:c9:8e:b4:ff:54:1c:09:bd:b0:54:
3a:a0:ee:8d:e6:5d:f5:cd:b9:63:bd:29:f6:5b:6d:
58:d0:55:bc:fe:45:da:78:c7:a4:d3:ba:45:11:12:
03:fd:17:57:14:24:f2:de:28:60:1a:8e:15:af:5a:
4a:ad:cf:52:bb:af:ca:8e:e7:1d:57:fa:59:9b:5b:
e9:9b:86:97:f9:b2:25:fe:35:c8:c9:d8:83:61:3e:
e2:5c:9b:86:b3:b9:ea:9a:5c:a5:cb:a4:92:cb:47:
70:ba:69:49:80:fe:49:a6:6a:f7:54:78:20:fa:c7:
5d:83
```

```

Exponent: 65537 (0x10001)
X509v3 extensions:
  Netscape Cert Type:
    SSL Client, SSL Server, S/MIME, Object Signing
Signature Algorithm: md5WithRSAEncryption
b8:2d:2b:c8:19:d0:cc:19:ef:d9:9b:b8:7e:2e:ca:a5:ed:ab:
d8:14:60:61:53:21:b9:66:72:7e:91:59:4e:2e:9d:b4:a6:6a:
a9:c4:32:f9:6c:35:a5:b6:2f:36:ad:3c:61:19:9c:48:4a:7a:
3b:47:8e:39:27:fe:37:a4:1d:a4:1c:42:e8:df:74:e5:fb:da:
e4:4c:9d:9c:1f:aa:be:b3:ff:94:25:f8:04:a3:17:9c:dc:39:
b9:85:e6:41:ac:28:59:fd:33:16:cd:51:f4:59:b1:fc:c3:bb:
1b:56:a9:67:99:3a:1f:f2:e6:a7:7a:01:ed:4f:d1:11:e2:d6:
db:23
-----BEGIN CERTIFICATE-----
MIICpTCCAg6gAwIBAgIBNzANBgkqhkiG9w0BAQQFADA1MQswCQYDVQQGEwJJVDEO
MAwGA1UEChMFRUdSSUQxRjAUBGNVBAMTDUUncmkiENBIFRlc3QwHhcNMDQwNzE1
MTQyMTEwWWhcNMDUwNzE1MTQyMTEwWjBhMQswCQYDVQQGEwJJVDEYMBYGA1UEBxMP
VHJpZXR0ZSAtIEI0YWx5MQ4wDAYDVQQKEwVFR1JJRDENMAAsGA1UECzMESUNUUEZJ
MBcGA1UEAxMQQRXppbyBDb3JzbyBQcm92YTCASlwdQYJKoZIhvcNAQEBBQADggEP
ADCCAQoCggEBAMrfwR+aw6z2op6ING6zLgrpuJ9gnpMAw9uCnEC5ph4wKRBC4Rv9
PNzq1SX90B2Bg0PjGTRWxwqdEABlVfIXCNmvStrWMYsGwZ+uedbC3JxdL8oJvcC
uD1imub0xlnAJCLY4n5FaNvCsuuXvjLLca1zJpIKcTto2YF99ybLUBgX0QKnD0h+
r+l0mazJrT/VBwJvbBUOqDujeZd9c25Y70p9lWNBVvP5F2njHpNO6RRESA/0X
VxQk8t4oYBqOfa9aSq3PUruvyo7nHVf6WZtb6ZuGI/myJf41yMnYg2E+4lybhrO5
6ppcpcukkstHclppSYD+SaZq91R4IPrHXymCAwEAAAMVMBMwEQYJYIZIAYb4QgEB
BAQDAgTwMA0GCSqGSIb3DQEBAUAA4GBALgtK8gZ0MwZ79mbuH4uyqXtq9gUYGFT
lblmnc6RWU4unbSmaqNEMvisNaW2LzatPGEZnEhKejtHjkn/jekHaQcQujfdOX7
2uRMnZwfqr6z/5Ql+ASjF5zcObmF5kGsKFn9MxbNUfRZsfzDuxtWqWeZOh/y5qd6
Ae1P0RHi1tsj
-----END CERTIFICATE-----

```

Creazione di un utente nella UI

Creare per prima cosa una home sulla macchina; nell'esempio di prima: accedere come root alla macchina UI e creare la home `/home/egrid/ecp` per l'utente Ezio Corso Prova.

Ricordarsi alla fine di aggiustare ownership della home e dei file creati, assegnandoli all'utente aggiunto e non lasciarli a root!

Nella home dell'utente creare la directory `.globus` dove copiare i tre file `.pem` del punto precedente.

La macchina UI è impostata per autenticazione LDAP: basta aggiungervi due opportune entry, una che aggiusterà `/etc/passwd` e l'altra per `/etc/group`. Dati del nostro LDAP Server:

```

Host:          egrid-1.egrid.it
Port:          10389 (non SSL - oppure 10636 per quella SSL)
Base DN:       ou=Testbed-Egrid,o=egrid,c=it
User DN:       cn=Manager
Password:      (quella solita!)
Append Base DN -> true!

```

Ramo che gestisce `/etc/passwd`:

Esportare un'entry esistente dal ramo People del LDAP Server, cambiare i parametri relativi, ricaricare la entry modificata. Per continuare con il nostro esempio:

dn: cn=Ezio Corso Prova, ou=People, ou=egrid, ou=Testbed-Egrid,o=egrid,c=it *NOTA (1)*
shadowMin: -1
sn: Corso Prova *NOTA (2)*
userPassword:: e1NIQX0veVhvUW9WMW9jaW9BOGhQTGhFL0JoVHB2bUE9 *NOTA (3)*
loginShell: /bin/bash
uidNumber: 2020 *NOTA (4)*
gidNumber: 2020 *NOTA (5)*
shadowFlag: 0
shadowExpire: -1
shadowMax: 99999
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: pkiUser
objectClass: posixAccount
objectClass: shadowAccount
objectClass: top
uid: ecp *NOTA(6)*
shadowLastChange: 11192
cn: Ezio Corso Prova *NOTA(7)*
homeDirectory: /home/egrid/ecp *NOTA(8)*
shadowInactive: -1
description: subject= /C=IT/L=Trieste - Italy/O=EGRID/OU=ICTP/CN=Ezio Corso Prova *NOTA(9)*
shadowWarning: 7

Note:

1. Modificare il cn
2. Modificare il sn
3. Modificare la Password: deve essere il hash della password voluta; se si usa LDAPBrowser esiste una funzionalità che automaticamente converte la password digitata in chiaro.
4. Per assegnare UID, vedere le altre entry LDAP e trovare la prima vuota.
5. Analogamente a 4
6. Impostare il Nome utente locale
7. Impostare il cn
8. Impostare la Home
9. Riportare il *Certificate Subject* dell'utente, creato al passo precedente.

Ramo che gestisce /etc/group:

Esportare un'entry esistente dal ramo Group del LDAP Server, cambiare i parametri relativi, ricaricare la entry modificata. Per continuare con il nostro esempio:

dn: cn=ecp, ou=Group, ou=egrid, ou=Testbed-Egrid,o=egrid,c=it *NOTA(1)*
gidNumber: 2020 *NOTA(2)*
objectClass: posixGroup
objectClass: top
cn: ecp *NOTA(3)*

Note:

1. cn deve essere cambiato a *ecp*, il nostro utente! È il DN del ramo: non si applica ad

altri contesti di autenticazione LDAP ma solo al nostro per via do come abbiamo fuso il *VOServer* con l'autenticazione LDAP.

2. Il GUID del gruppo che vogliamo aggiungere: in particolare vogliamo far coincidere il gruppo con l'utente stesso.
3. Il nome da associare al gruppo: *ecp* (vedi nota precedente)

Aggiunta utente alla VO

L'autenticazione LDAP ricicla parte dei rami del *VOServer*, per cui al punto precedente si sono già parzialmente compilate le informazioni richieste: in particolare la *Nota (9)* soddisfa la questione dei filtraggi per lo script *edg-mkgridmap*, e la questione della sicurezza del SE per lo script *ldap2gridmap*.

Manca da aggiustare il ramo Group1, utilizzato da *edg-mkgridmap*: anche per questo ramo esportare una entry, modificarla e ricaricarla.

```
dn: ou=Group1, ou=eGRID, ou=Testbed-Egrid, o=eGRID, c=it
owner: cn=Manager, ou=Group1, ou=eGRID, ou=Testbed-Egrid, o=eGRID, c=it
ou: Group1
objectClass: groupofnames
cn: Group1
member: cn=Alessandra Tedeschi, ou=People, ou=eGRID, ou=Testbed-Egrid, o=eGRID, c=it
member: cn=Alvise Nobile, ou=People, ou=eGRID, ou=Testbed-Egrid, o=eGRID, c=it
member: cn=Alessio Terpin, ou=People, ou=eGRID, ou=Testbed-Egrid, o=eGRID, c=it
member: cn=Ezio Corso Prova, ou=People, ou=eGRID, ou=Testbed-Egrid, o=eGRID, c=it
```

Nota (1)

Note:

1. Basta aggiungere l'attributo *member* come illustrato, cioè con i dati del *Certificate Subject*.

Attivare gli utenti sulle macchine della griglia

Aspettare le ore di esecuzione degli script oppure eseguirli su tutte le macchine, sia *edg-mkgridmap* che *ldap2gridmap*.