
Karaage Admin Documentation

Release 3.1.32.dev32+g65cd93f

Brian May

Aug 04, 2017

1	Introduction	3
1.1	What is Karaage?	3
1.2	Features	3
1.3	Admin access	3
1.4	User access	4
2	Getting Started	5
2.1	Assumptions	5
2.2	Installation	5
2.3	SSL certificate	6
2.4	Apache Configuration	6
2.5	MySQL configuration	8
2.6	Initial setup	8
2.7	Data stores	9
2.8	Plugins	9
2.9	Cluster tools	9
3	Data Stores	11
3.1	Adding MOAB Account Manager	11
3.2	Adding OpenLDAP	12
3.3	Adding Slurm	17
4	Plugins	19
4.1	Karaage Usage plugin	19
5	Cluster tools	21
5.1	Debian Installation	21
5.2	Configuring	21
6	Common Tasks	23
6.1	Adding machines	23
6.2	Adding people	24
7	Upgrading	25
7.1	Upgrading from 2.7.x to 3.1.x	25
7.2	Upgrading from 3.0.x to 3.1.x	31

8	Reference	33
8.1	Commands	33
8.2	Karaage Settings	37
9	Change log	45
9.1	3.1.31 - 2017-05-03	45
9.2	3.1.30 - 2016-09-11	46
9.3	3.1.29 - 2016-08-11	46
9.4	3.1.28 - 2016-05-11	46
9.5	3.1.27 - 2016-05-10	46
9.6	3.1.26 - 2016-05-06	47
9.7	3.1.25 - 2016-05-03	47
9.8	3.1.24 - 2016-05-03	47
9.9	3.1.23 - 2016-04-29	47
9.10	3.1.22 - 2015-06-19	47
9.11	3.1.21 - 2015-06-17	48
9.12	3.1.20 - 2015-06-05	48
9.13	3.1.19 - 2015-05-29	48
9.14	3.1.18 - 2015-04-13	49
9.15	3.1.17 - 2015-03-30	49
9.16	3.1.16 - 2015-03-17	49
9.17	3.1.15 - 2015-03-10	49
9.18	3.1.14 - 2015-02-19	49
9.19	3.1.13 - 2015-02-17	49
9.20	3.1.12 - 2015-02-16	50
9.21	3.1.11 - 2015-02-12	50
9.22	3.1.10 - 2014-12-01	50
9.23	3.1.9 - 2014-10-30	50
9.24	3.1.8 - 2014-10-13	50
9.25	3.1.7 - 2014-10-10	50
9.26	3.1.6 - 2014-09-30	51
9.27	3.1.5 - 2014-09-18	51
9.28	3.1.4 - 2014-09-15	51
9.29	3.1.3 - 2014-09-09	51
9.30	3.1.2 - 2014-08-27	51
9.31	3.1.1 - 2014-08-19	52
9.32	3.1.0 - 2014-07-30	52
9.33	3.0.15 - 2014-06-17	53
9.34	3.0.14 - 2014-05-27	53
9.35	3.0.13 - 2014-05-05	54
9.36	3.0.12 - 2014-05-01	54
9.37	3.0.11 - 2014-04-10	54
9.38	3.0.10 - 2014-04-02	54
9.39	3.0.9 - 2014-03-25	55
9.40	3.0.8 - 2014-03-14	55
9.41	3.0.7 - 2014-03-13	55
9.42	3.0.6 - 2014-03-11	56
9.43	3.0.5 - 2014-03-03	56
9.44	3.0.4 - 2014-02-27	56
9.45	3.0.3 - 2014-02-24	56
9.46	3.0.2 - 2014-02-05	56
9.47	3.0.1 - 2014-01-30	57
9.48	3.0.0 - 2013-07-18	57
9.49	2.7.6 - 2013-03-27	58

9.50	2.7.5 - 2013-03-25	58
9.51	2.7.4 - 2013-03-22	58
9.52	2.7.3 - 2013-03-15	58
9.53	2.7.3 - 2013-03-15	58
9.54	2.7.2 - 2013-02-19	59
9.55	2.7.1 - 2013-02-11	59
9.56	2.7.0 - 2013-02-11	59
9.57	2.6.8 - 2012-11-19	59
9.58	2.6.7 - 2012-11-14	59
9.59	2.6.6 - 2012-11-13	59
9.60	2.6.5 - 2012-10-16	59
9.61	2.6.4 - 2012-03-22	60
9.62	2.6.3 - 2012-02-07	61
9.63	2.6.2 - 2011-10-19	61
9.64	2.6.1 - 2011-08-30	61
9.65	2.6 - 2011-08-02	61
9.66	2.5.17 - 2011-07-15	62
9.67	2.5.16 - 2011-06-27	62
9.68	2.5.15 - 2011-06-14	62
9.69	2.5.14 - 2011-06-10	62
9.70	2.5.13 - 2011-06-03	63
9.71	2.5.12 - 2011-06-03	63
9.72	2.5.11 - 2011-06-01	63
9.73	2.5.10 - 2011-05-25	63
9.74	2.5.9 - 2011-05-18	63
9.75	2.5.8 - 2011-05-04	64
9.76	2.5.7 - 2011-03-30	64
9.77	2.5.6 - 2011-03-09	64
9.78	2.5.5 - 2011-03-08	64
9.79	2.5.4 - 2011-02-23	64
9.80	2.5.3 - 2011-02-21	64
9.81	2.5.2 - 2010-12-15	65
9.82	2.5.1 - 2010-12-10	65
9.83	2.5 - 2010-11-17	65
9.84	2.4.14 - 2010-11-17	66
9.85	2.4.13 - 2010-10-20	66
9.86	2.4.12 - 2010-10-13	66
9.87	2.4.11 - 2010-10-07	66
9.88	2.4.10 - 2010-10-04	67
9.89	2.4.9 - 2010-09-29	67
9.90	2.4.8 - 2010-09-15	67
9.91	2.4.7 - 2010-08-25	67
9.92	2.4.6 - 2010-08-25	67
9.93	2.4.5 - 2010-08-17	67
9.94	2.4.4 - 2010-08-12	68
9.95	2.4.3 - 2010-07-28	68
9.96	2.4.2 - 2010-07-28	68
9.97	2.4.1 - 2010-07-27	68
9.98	2.4 - 2010-07-27	68
9.99	2.3.11 - 2010-07-21	68
9.100	2.3.10 - 2010-07-20	69
9.101	2.3.9 - 2010-06-08	69
9.102	2.3.8 - 2010-06-08	69
9.103	2.3.7 - 2010-05-31	69

9.104 2.3.6 - 2010-05-28	69
9.105 2.3.5 - 2010-05-28	69
9.106 2.3.4 - 2010-05-26	69
9.107 2.3.3 - 2010-05-19	70
9.108 2.3.2 - 2010-05-19	70
9.109 2.3.1 - 2010-05-19	70
9.110 2.3 - 2010-05-19	70
9.111 2.2.1 - 2010-05-17	70
9.112 2.2 - 2010-05-17	70
9.113 2.1.1 - 2010-05-07	71
9.114 2.1 - 2010-05-06	71
9.115 2.0.16 - 2010-05-05	71
9.116 2.0.15 - 2010-05-05	71
9.117 2.0.14 - 2010-05-03	71
9.118 2.0.13 - 2010-05-03	71
9.119 2.0.12 - 2010-04-29	71
9.120 2.0.11 - 2010-04-28	72
9.121 2.0.10 - 2010-04-28	72
9.122 2.0.9 - 2010-04-28	72
9.123 2.0.8 - 2010-04-01	72
9.124 2.0.7 - 2010-03-31	72
9.125 2.0.6 - 2010-03-24	72
9.126 2.0.5 - 2010-03-22	72
9.127 2.0.4 - 2010-03-22	72
9.128 2.0.3 - 2010-03-22	72
9.129 2.0.2 - 2010-03-19	73
9.130 2.0.1 - 2010-03-19	73
10 Glossary	75
11 Indices and tables	77

This is documentation for Karaage 3.

Date Aug 04, 2017

Version 3.1

Contents:

What is Karaage?

Karaage is a cluster account management tool. It can manage users and projects in a cluster and can store the data in various backends.

Features

- Can store user information and/or posix account information in LDAP/Active Directory/passwd file.
- Email notifications.
- Auto account creation - Allow project leaders to manage their users.
- Applications work flow - Users can apply for accounts and be approved by project leaders.
- Usage reporting. Report on a per institute, per project or per user for CPU usage.
- Track usage of software and versions. Keep track of what software (and version) and type of jobs a user is running.

Admin access

When an administrator logs into karaage, they can do the following:

- See/modify information about all users/projects/institutes/applications/usage etc.
- Many changes are logged.
- Anything a normal user can.

User access

When a normal user logs into Karaage (or an administrator), they can do the following.

Allows users to:

- Modify their own account.
- Change password.
- Manage what projects they are in.

Allow project leaders to:

- Approve/Decline applications to join their project.
- Reset passwords for their members.
- Track their resource utilisation and software utilisation.

Allow institute delegates to:

- Approve/Decline new project applications.
- Manage all projects and users under the institute.

This section is for administrators who don't already have Karaage installed and wish to get started for the first time. If you do have an older version of Karaage already installed, please see the *Upgrading* section.

Assumptions

Assumptions made by this documentation. Other configurations are possible, however you will have to adapt from this documentation.

- You have a cluster running Slurm or PBS that is already up and running.
- You want to OpenLDAP with ppolicy support, on the same server.
- You have a Debian Jessie server already setup for Karaage.
- You will be installing all components on a single system.
- You are upgrading to Karaage 3.1.
- The visible hostname is `www.example.org`. This will have to be changed as required.

Installation

1. If you require a proxy server for out going connections, set it up now.

```
export http_proxy=http://proxy.example.org
```

2. You need to install the VPAC Debian Archive signing key:

```
wget http://linuxpenguins.xyz/debian/vpac-debian-key.gpg -O - | apt-key add -
```

3. Create a `/etc/apt/sources.list.d/karaage.list` containing:

```
deb http://linuxpenguins.xyz/debian jessie main
deb-src http://linuxpenguins.xyz/debian jessie main
```

4. Update your apt database and install the packages:

```
apt-get update
```

SSL certificate

You should create a signed SSL certificate for Apache and LDAP.

1. Generate a SSL private key, a CSR.

```
cd /etc/ssl/private
openssl genrsa -out www_privatekey.pem 2048
chmod 640 www_privatekey.pem
openssl req -new -key www_privatekey.pem -out www_csr.pem -sha256
```

2. Submit `www_csr.pem` to a CA, and get it signed. Copy resultant certificate into `www_cert.pem`. Check this file is sha256:

```
openssl x509 -text -noout -in www_cert.pem
```

You should see the following text:

```
Signature Algorithm: sha256WithRSAEncryption.
```

You may need an intermediate certificate too. Copy this into `www_intermediate.pem`.

3. Join certificate with intermediate (required for some versions of slapd):

```
cd /etc/ssl/private
cat www_cert.pem www_intermediate.pem > www_combined.pem
```

4. Setup the permissions:

```
apt-get install ssl-cert
cd /etc/ssl/private
chown root:ssl-cert www_*.pem
```

Apache Configuration

Karaage, by default, requires a https connection. While this default can be changed, this is not advisable on a production system.

In the following steps, replace `www.example.org` with the visible hostname of your server.

1. Install apache2.

```
apt-get install apache2
```

2. Setup Apache to support secure https connections. Changes should be made to `/etc/apache2/sites-available/default-ssl`:

```
SSLCertificateFile /etc/ssl/private/www_cert.pem
SSLCertificateKeyFile /etc/ssl/private/www_privatekey.pem
SSLCertificateChainFile /etc/ssl/private/www_intermediate.pem
```

For more details on what changes are required, see the [Apache howto](#).

3. Connections to http should be redirected to https. Please replace the `/etc/apache2/sites-available/default` file entirely with the following:

```
<VirtualHost *:80>
    ServerName www.example.org
    Redirect permanent / https://www.example.org/
</VirtualHost>
```

For more information on this step, see the [Apache wiki](#).

4. (recommended) It is recommended that you change the following settings in `/etc/apache2/mods-available/ssl.conf` to make SSL more secure by disabling insecure protocols and ciphers:

```
SSLProtocol all -SSLv2 -SSLv3
SSLCipherSuite ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-
↪RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-SHA384:DHE-RSA-AES128-GCM-
↪SHA256:DHE-DSS-AES128-GCM-SHA256:kEDH+AESGCM:ECDHE-RSA-AES128-SHA256:ECDHE-
↪ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-
↪AES256-SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA:ECDHE-ECDSA-AES256-
↪SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES128-SHA:DHE-DSS-AES128-SHA256:DHE-RSA-
↪AES256-SHA256:DHE-DSS-AES256-SHA:DHE-RSA-AES256-SHA:AES128-GCM-SHA256:AES256-
↪GCM-SHA384:AES128-SHA256:AES256-SHA256:AES128-SHA:AES256-SHA:AES:CAMELLIA:DES-
↪CBC3-SHA:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5:!PSK:!aECDH:!EDH-DSS-DES-CBC3-
↪SHA:!EDH-RSA-DES-CBC3-SHA:!KRB5-DES-CBC3-SHA
```

Note however that the `SSLProtocol` will break IE6, and the `SSLCipherSuite` setting will break IE on XP. For more information on securing Apache, see the [Mozilla website](#).

5. Enable `default-ssl` with the following commands:

```
a2enmod ssl
a2ensite default-ssl.
service apache2 restart
```

6. Test by loading both `http://www.example.org/` and `https://www.example.org/` in your browser.
7. (recommended) Enable **HSTS** support with the following commands:

```
echo 'Header always set Strict-Transport-Security "max-age=31536000;␣
↪includeSubDomains"' > /etc/apache2/conf-available/hsts.conf
a2enmod headers
a2enconf hsts
service apache2 restart
```

8. Test by loading both `http://www.example.org/` and `https://www.example.org/` in your browser.
9. Test website with **SSL Test**.

MySQL configuration

1. Run the following commands:

```
apt-get install mysql-server
```

This should ask for a password for the root mysql user. Make sure this is a secure password. You can use `makepasswd` if you want. For the purpose of this documentation, we will assume you used `XXXXXXXX`. Do not use `XXXXXXXX` for your password on a production system.

2. (optional) Create a `/root/.my.cnf` file containing:

```
[client]
user      = root
password  = XXXXXXXX
```

3. Create a `/etc/mysql/conf.d/karaage.cnf` file containing:

```
[mysqld]
character_set_server=utf8
default-storage-engine = innodb
sql_mode = STRICT_ALL_TABLES

[client]
default-character-set = utf8
```

Note: these settings may affect other applications that use this database.

4. Restart mysql server to load config, and connect to it:

```
service mysql restart
mysql
```

5. Create a user and database for karaage:

```
mysql> create database karaage;
mysql> CREATE USER 'karaage'@'localhost' IDENTIFIED BY 'YYYYYYYYY';
mysql> GRANT ALL PRIVILEGES ON karaage.* TO 'karaage'@'localhost';
```

You will use the values you set, later, in the karaage settings. Do not use `YYYYYYYYY` on a production system.

Initial setup

1. Install the packages:

```
apt-get install karaage3
apt-get install python-mysqldb
```

2. Edit the `DATABASES` setting in `/etc/karaage3/settings.py`:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'karaage',
        'USER': 'karaage',
        'PASSWORD': 'YYYYYYYYY',
```

```
'HOST': 'localhost',
'PORT': '',
'ATOMIC_REQUESTS': True,
'OPTIONS': {
    'sql_mode': 'STRICT_ALL_TABLES'
},
}
```

3. Add the `HTTP_HOST` setting in `/etc/karaage3/settings.py`:

```
HTTP_HOST = "www.example.org"
```

Replace `www.example.org` with the visible hostname of your server.

4. Update other settings in `/etc/karaage3/settings.py` as required. See comments in this file and *Karaage Settings*.
5. Reload Apache after changing `/etc/karaage3/settings.py`.

```
service apache2 reload
```

6. Create DB tables:

```
kg-manage migrate
```

7. Create a karaage superuser using `kgcreatesuperuser`:

```
kg-manage kgcreatesuperuser
```

8. Setup cron job to automatically start `daily_cleanup`. Edit the `/etc/cron.d/karaage3-database` file:

```
10 1 * * * www-data /usr/bin/kg-manage daily_cleanup
```

9. Test. You should now be able to go to `http://www.example.org/karaage/`.

Data stores

So far you have not configured any external datastores. Karaage will work, however probably won't do anything useful. See the next section to configure datastores (*Data Stores*).

Plugins

For information on configuring additional plugins, see *Plugins*.

Cluster tools

If installing Karaage on a cluster, you may want to install the cluster tools, see *Cluster tools*.

A *data store* is how Karaage updates external databases, such a LDAP, PBS, and Slurm.

Contents:

Adding MOAB Account Manager

1. First configure the LDAP datastore. See *Adding OpenLDAP*.
2. Install mam command line.
3. Test using `www-data` user.
4. Add project to MAM that has no access. Call it `default` (or whatever else you want).
5. Edit the `MACHINE_CATEGORY_DATASTORES` setting in `/etc/karaage3/settings.py`:

```
MACHINE_CATEGORY_DATASTORES = {
    'ldap': [
        {
            'DESCRIPTION': 'LDAP datastore',
            ...
        },
        {
            'DESCRIPTION': 'MAM datastore',
            'ENGINE': 'karaage.datastores.mam.MamDataStore',
            'PREFIX': [],
            'PATH': '/usr/local/mam/bin:/usr/local/mam/sbin',
            'NULL_PROJECT': 'default',
        },
    ],
    'dummy': [
    ],
}
```

Values `PREFIX`, `PATH`, and `NULL_PROJECT` are defaults and can be omitted.

6. In `/etc/karaage3/settings.py` uncomment the `LOGGING` assignment lines related to mam.
7. Reload apache.

```
service apache2 reload
service python-karaage-celery restart
```

Adding OpenLDAP

Assumptions

You will need to substitute correct values for the following when applicable:

- Base DN: `dc=example,dc=org`
- Administrator DN: `cn=admin,dc=example,dc=org`
- Administrator password: `XXXXXXXXXX` (do not use `XXXXXXXXXX`).

RHEL 6 installation

1. Run the following commands:

```
yum install openldap-servers
yum install openldap-clients
cp -rv /usr/share/openldap-servers/DB_CONFIG.example /var/lib/ldap/DB_CONFIG
chown -R ldap:ldap /var/lib/ldap
cd /etc/openldap/slapd.d/cn=config
```

Do not start the server yet.

2. Encrypt the admin password:

```
slappasswd
```

Enter `XXXXXXXXXX` twice. This should output an encrypted password starting with `XXXXXXXXXX`. Copy that into the clipboard.

The result for `XXXXXXXXXX` is `{SSHA}4bxi0+aXeYvv2TGT10VWUIwcaynqBbxH` (do not use this value).

3. Edit `olcDatabase={2}bdb.ldif`, and update/add the following values. Do not change anything else:

```
olcSuffix: dc=example,dc=org
olcRootDN: cn=admin,dc=example,dc=org
olcRootPW: {SSHA}4bxi0+aXeYvv2TGT10VWUIwcaynqBbxH
```

4. Edit `olcDatabase={1}monitor.ldif`, and update update the admin DN. Do not change anything else:

```
olcAccess: {0}to * by dn.base="gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth" read by dn.base="cn=admin,dc=example,dc=org" read by * none
```

5. Run the following commands:

```
service slapd start
chkconfig slapd on
```

6. Create the file with the following contents in /tmp/ldapssl.ldif:

```
dn: cn=config
changetype: modify
replace: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ssl/private/www_cert.pem
-
replace: olcTLSCertificatekeyFile
olcTLSCertificatekeyFile: /etc/ssl/private/www_privatekey.pem
-
replace: olcTLSCACertificateFile
olcTLSCACertificateFile: /etc/ssl/private/www_intermediate.pem

dn: olcDatabase={2}bdb,cn=config
changetype: modify
delete: olcTLSCertificateFile
-
delete: olcTLSCertificateKeyFile
```

7. Import with the following command:

```
ldapmodify -Y EXTERNAL -H ldapi:/// < /tmp/ldapssl.ldif
```

8. Edit /etc/sysconfig/ldap:

```
SLAPD_LDAPS=yes
```

9. Restart LDAP server.

```
service slapd restart
```

10. Create the file with the following contents in /tmp/ppolicy1.ldif:

```
dn: cn=module,cn=config
objectClass: olcModuleList
cn: module
olcModulepath: /usr/lib/ldap
olcModuleload: ppolicy.so

dn: olcOverlay=ppolicy,olcDatabase={1}hdb,cn=config
objectClass: olcPPolicyConfig
olcPPolicyDefault: cn=default,ou=policies,dc=example,dc=org

dn: olcDatabase={2}bdb,cn=config
changetype: modify
add: olcAccess
olcAccess: to attrs=userPassword,shadowLastChange by anonymous auth by dn=
↔"cn=admin,dc=example,dc=org" write by * none
olcAccess: to * by dn="cn=admin,dc=example,dc=org" write by * read
```

11. Import with the following command:

```
ldapadd -x -H ldapi:/// -D cn=admin,dc=example,dc=org -W < /tmp/ppolicy1.ldif
```

12. Create the file with the following contents in /tmp/ppolicy2.ldif:

```
dn: dc=example,dc=org
objectClass: top
```

```
objectClass: domain

dn: ou=Accounts,dc=example,dc=org
objectClass: organizationalUnit

dn: ou=Groups,dc=example,dc=org
objectClass: organizationalUnit

dn: ou=policies,dc=example,dc=org
objectClass: organizationalUnit

dn: cn=default,ou=policies,dc=example,dc=org
objectClass: top
objectClass: device
objectClass: pwdPolicy
pwdAttribute: userPassword
```

13. Import with the following command:

```
ldapadd -Y EXTERNAL -H ldapi:/// < /tmp/ppolicy2.ldif
```

14. Test ldap connections.

```
ldapsearch -x -b'dc=example,dc=org' -D cn=admin,dc=example,dc=org -W -ZZ
```

Fix any errors.

15. Force the use of SSL for accessing the main database without disabling access to cn=config. Create the file with the following contents in /tmp/security.ldif:

```
dn: olcDatabase={2}bdb,cn=config
changetype: modify
replace: olcSecurity
olcSecurity: tls=1
```

16. Import with the following command:

```
ldapmodify -Y EXTERNAL -H ldapi:/// < /tmp/security.ldif
```

Note: This won't guarantee that LDAP passwords are never sent in the clear, however such attempts should fail.

Debian installation

1. Run the following commands:

```
apt-get install slapd
apt-get install ldap-utils
addgroup openldap ssl-cert
```

Enter XXXXXXXXX when prompted for administrator's password.

2. Create the file with the following contents in /tmp/ppolicy1.ldif:

```
dn: cn=module,cn=config
objectClass: olcModuleList
cn: module
olcModulepath: /usr/lib/ldap/
olcModuleload: ppolicy.la

dn: olcOverlay=ppolicy,olcDatabase={1}hdb,cn=config
objectClass: olcPPolicyConfig
olcPPolicyDefault: cn=default,ou=policies,dc=example,dc=org
```

3. Create the file with the following contents in /tmp/ldapssl.ldif:

```
dn: cn=config
changetype: modify
replace: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ssl/private/www_cert.pem
-
replace: olcTLSCertificatekeyFile
olcTLSCertificatekeyFile: /etc/ssl/private/www_privatekey.pem
-
replace: olcTLSCACertificateFile
olcTLSCACertificateFile: /etc/ssl/private/www_intermediate.pem
```

4. Import with the following command:

```
ldapadd -Y EXTERNAL -H ldapi:/// < /etc/ldap/schema/ppolicy.ldif
ldapadd -Y EXTERNAL -H ldapi:/// < /tmp/ppolicy1.ldif
ldapmodify -Y EXTERNAL -H ldapi:/// < /tmp/ldapssl.ldif
```

5. Create the file with the following contents in /tmp/ppolicy2.ldif:

```
dn: ou=policies,dc=example,dc=org
objectClass: organizationalUnit

dn: ou=Accounts,dc=example,dc=org
objectClass: organizationalUnit

dn: ou=Groups,dc=example,dc=org
objectClass: organizationalUnit

dn: cn=default,ou=policies,dc=example,dc=org
objectClass: top
objectClass: device
objectClass: pwdPolicy
pwdAttribute: userPassword
```

6. Import with the following command:

```
ldapadd -x -H ldapi:/// -D cn=admin,dc=example,dc=org -W < /tmp/ppolicy2.ldif
```

7. Test ldap connections.

```
ldapsearch -x -b'dc=example,dc=org' -ZZ
```

Fix any errors.

8. Force the use of SSL for accessing the main database without disabling access to cn=config. Create the file with the following contents in /tmp/security.ldif:

```
dn: olcDatabase={1}hdb,cn=config
changetype: modify
replace: olcSecurity
olcSecurity: tls=1
```

9. Import with the following command:

```
ldapmodify -Y EXTERNAL -H ldapi:/// < /tmp/security.ldif
```

Note: This won't guarantee that LDAP passwords are never sent in the clear, however such attempts should fail.

Configuring Karaage to use LDAP

1. Add the *LDAP* and *MACHINE_CATEGORY_DATASTORES* settings to `/etc/karaage3/settings.py`:

```
LDAP = {
    'default': {
        'ENGINE': 'tldap.backend.fake_transactions',
        'URI': 'ldap://www.example.org',
        'USER': 'cn=admin,dc=example,dc=org',
        'PASSWORD': 'XXXXXXXXX',
        'REQUIRE_TLS': True,
        'START_TLS': True,
        'TLS_CA': None,
    }
}

MACHINE_CATEGORY_DATASTORES = {
    'ldap': [
        {
            'DESCRIPTION': 'LDAP datastore',
            'ENGINE': 'karaage.datastores.ldap.MachineCategoryDataStore',
            'LDAP': 'default',
            'ACCOUNT': 'karaage.datastores.ldap_schemas.openldap_account',
            'GROUP': 'karaage.datastores.ldap_schemas.openldap_account_group',
            'PRIMARY_GROUP': "institute",
            'DEFAULT_PRIMARY_GROUP': "dummy",
            'HOME_DIRECTORY': "/home/%(uid)s",
            'LOCKED_SHELL': "/usr/local/sbin/locked",
            'NUMBER_SCHEME': 'default',
            'LDAP_ACCOUNT_BASE': 'ou=Accounts,dc=example,dc=org',
            'LDAP_GROUP_BASE': 'ou=Groups,dc=example,dc=org',
        },
    ],
    'dummy': [
    ],
}
}
```

2. (optional) If you require people to be recorded in LDAP, add the *GLOBAL_DATASTORES* setting to `/etc/karaage3/settings.py`:

```
GLOBAL_DATASTORES = [
    {
```

```

        'DESCRIPTION': 'LDAP datastore',
        'ENGINE': 'karaage.datastores.ldap.GlobalDataStore',
        'LDAP': 'default',
        'PERSON': 'karaage.datastores.ldap_schemas.openldap_person',
        'GROUP': 'karaage.datastores.ldap_schemas.openldap_person_group',
        'NUMBER_SCHEME': 'global',
        'LDAP_PERSON_BASE': 'ou=People,dc=example,dc=org',
        'LDAP_GROUP_BASE': 'ou=People_Groups,dc=example,dc=org',
    },
]

```

For best results the base settings should be different for the `GLOBAL_DATASTORES` and the `MACHINE_CATEGORY_DATASTORES` settings.

3. Reload apache.

```

service apache2 reload
service python-karaage-celery restart

```

4. Log into web interface and add a machine category that references the ldap datastore. This should automatically populate LDAP with any entries you have created.

5. Add missing LDAP entries:

```

kg-manage migrate_ldap

```

Adding Slurm

1. First configure the LDAP datastore. See *Adding OpenLDAP*.
2. Install slurm command line.
3. Add to `/etc/sudoers`:

```

www-data ALL=(slurm) NOPASSWD: /usr/local/slurm/latest/bin/sacctmgr

```

4. Test using `www-data` user:

```

sudo -uslurm /usr/local/slurm/latest/bin/sacctmgr -ip

```

5. Add project to slurm that has no access. Call it default (or whatever else you want).
6. Edit the `MACHINE_CATEGORY_DATASTORES` setting in `/etc/karaage3/settings.py`:

```

MACHINE_CATEGORY_DATASTORES = {
    'ldap': [
        {
            'DESCRIPTION': 'LDAP datastore',
            ...
        },
        {
            'DESCRIPTION': 'Slurm datastore',
            'ENGINE': 'karaage.datastores.slurm.SlurmDataStore',
            'PREFIX': [ "sudo", "-uslurm" ],
            'PATH': "/usr/local/slurm/latest/bin/sacctmgr",
            'NULL_PROJECT': 'default',
        },
    ],
}

```

```
    ],  
    'dummy': [  
    ],  
}
```

Values PREFIX, PATH, and NULL_PROJECT are defaults and can be omitted.

7. In `/etc/karaage3/settings.py` uncomment the `LOGGING` assignment lines related to slurm.
8. Reload apache.

```
service apache2 reload  
service python-karaage-celery restart
```


There are a number of plugins for Karaage that are not enabled by default.

- Karaage Applications: application process for karaage.
- Karaage Software: keep track of software.
- Karaage Usage: keep track of usage.

Karaage Usage plugin

Karaage usage requires karaage applications and karaage software, and will automatically pull in the required dependencies. It is an external plugin that provides usage graphs and statistics for cluster systems. To enable it:

1. Install the extra required package:

```
apt-get install python-kgusage
```

2. Add the following to `/etc/karaage3/settings.py`:

```
PLUGINS = [  
    'karaage.plugins.kgusage.plugin',  
]
```

3. Run the database migrations, restart apache, and install the celery daemon.

```
kg-manage migrate --all  
service apache2 reload  
apt-get install karaage3-celery
```

4. Check to ensure the celery daemon is running.

You may need to update PBS/slurm logging to talk correctly to Karaage. Only do this if `kg-pbs-logger` was previously configured.

Karaage no longer requires a dedicated account for `kg-pbs-logger`. Rather it uses the machine entry.

For every machine:

- Navigate to machine entry in admin website.
- Click password button to reset the password.
- Update `/etc/karaage/pbs-logger.cfg` and for `ws_username` use the machine name in karaage, and for `ws_password` use the password obtained in the previous step.
- Test.

Note: You should have the karaage-usage plugin installed before continuing.

Debian Installation

1. If you require a proxy server for out going connections, set it up now.

```
export http_proxy=http://proxy.example.org
```

2. You need to install the VPAC Debian Archive signing key:

```
wget http://linuxpenguins.xyz/debian/vpac-debian-key.gpg -O - | apt-key add -
```

3. Create a /etc/apt/sources.list.d/karaage.list containing:

```
deb      http://linuxpenguins.xyz/debian  jessie main
deb-src  http://linuxpenguins.xyz/debian  jessie main
```

4. Update your apt database and install the packages:

```
apt-get update
```

5. Install the the packages:

```
apt-get install karaage-cluster-tools
```

Configuring

1. Ensure Karaage is working, with karaage-usage plugin configured.

2. Create a machine category if not already defined.
3. Create a machine, and get its password with the reset password operation.
4. Edit `/etc/kaaraage/kaaraage-cluster-tools.cfg` with appropriate values.
5. Test.

Contents:

Adding machines

To add a machine, you need a machine category first.

Creating Machine Category

A *machine category* uses a *data store*. This should already be defined, as per instructions in the *Data Stores* section.

1. Login to Karaage as administrator.
2. Select `machines` in the menu on the left.
3. Select `Add machine category` in the action bar.
4. Type in the name and select a datastore.
5. Select save.

Adding a Machine

You can add a *machine* to a *machine category*.

1. Login to Karaage as administrator.
2. Select `machines` in the menu on the left.
3. Select `Add machine` in the action bar.
4. After creating the machine, you may need to assign it a password. Click the password button to generate a new password automatically.

5. If/when setting up django-pbs-logger, make sure it connects to Karaage using the machine name and the password you were given above.

Adding people

There are two ways of adding a new person to Karaage. One is that the person accesses Karaage, and places a request to join a project or create a new project. This request must then be approved by the project leader (existing project), or institute delegate (new project), and a system administrator.

The other way is for a system administrator to add the person in Karaage.

If you are an administrator and want to add a person:

1. Check there is an institute appropriate for the person, under Institutes. If not, add an institute.
2. If person is to join a project, check the project already exists, under Projects. If not, add the project.
3. Check if the person already exists, under People. If not, add the person.

Contents:

Upgrading from 2.7.x to 3.1.x

This section is for administrators who have an older 2.7.x version of Karaage already installed, and wish to upgrade to 3.1.x.

These instructions are also usable for upgrades from 2.6.x, however Debian/jessie is a prerequisite.

If you don't already have Karaage installed and wish to get started for the first time, please see the *Getting Started* section.

<p>Warning: This is a non-trivial upgrade with non-trivial database migrations. The database migrations will take some time to complete and could fail or cause data loss. It is recommended that you copy your production data and run the database migrations on a test system, as shown below.</p>
--

Assumptions

Assumptions made by this documentation. Other configurations are possible, however you will have to adapt from this documentation.

- You have a cluster running Slurm or PBS that is already up and running.
- You want to OpenLDAP with ppolicy support, on the same server.
- You have a Debian Wheezy server already setup for Karaage.
- You will be upgrading all components on a single system, called production.

Backup production box

Warning: Any changes to database while after the backup, but before the upgrade is finished, may be lost. It is recommended to shutdown Karaage before running the final run to ensure important changes are not made.

Backup various things on production box. Some of the following steps aren't strictly required, however recommended to guarantee no data lose will occur during the upgrade.

1. Backup `/etc/karaage` and everything under this directory.
2. Backup `/etc/ldap` or `/etc/openldap` and everything under this directory.
3. Dump mysql database on production box.

```
mysqldump karaage > karaage.sql
```

You will use this file in the next step.

4. Dump LDAP database on production box.

```
slapcat > karaage.ldif
```

5. Copy these files off the production box to somewhere secure.

Operating System

Ensure production box is running Debian jessie with all applicable updates. You will not be able to install Karaage3 if you are running anything older than Wheezy. It is also suggested (but not required) that you upgrade to the latest 2.7 Karaage here, as this will make downgrading easier in case of problems upgrading to 3.0.

Check MySQL server config

1. (optional) Create a `/root/.my.cnf` file containing:

```
[client]
user      = root
password  = XXXXXXXX
```

2. Create a `/etc/mysql/conf.d/karaage.cnf` file containing:

```
[mysqld]
character_set_server=utf8
default-storage-engine = innodb
sql_mode = STRICT_ALL_TABLES

[client]
default-character-set = utf8
```

Note: these settings may affect other applications that use this database.

3. Restart mysql server to load config:

```
service mysql restart
```


Installing Karaage3

Warning: Double check what you are doing to ensure you don't accidentally delete production data.

Warning: You must have sufficient disk space on the production box to create a copy of the Karaage database.

Note: The migrations do not make any invasive changes to LDAP in any way until you run `migrate_ldap`, manually, by hand. This can be done after testing everything to make sure it is working fine. This means you don't need to copy LDAP data, however making a backup is still recommended.

If any institute references a group by `gidNumber` that does not exist, and cannot be found by name either, it will automatically be created. Not having a group is an error, even for Karaage 2.7.x.

Note: If you do encounter any problems performing database migrations, please file a bug report, so the problem can be fixed. Bug reports can be submitted at [github](#).

1. Create a copy of the karaage database on your production mysql server:

- (a) Create a new user and database for karaage3:

```
mysql> create database karaage3;
mysql> CREATE USER 'karaage3'@'localhost' IDENTIFIED BY 'YYYYYYYYY';
mysql> GRANT ALL PRIVILEGES ON karaage3.* TO 'karaage3'@'localhost';
```

You will use the values you set, later, in the karaage settings. Do not use YYYYYYYYY on a production system.

It is recommended that the user `karaage3` should not have any access to the karaage database, to avoid the potential for mistakes being made.

- (b) Copy mysql to new database on production box:

```
mysql karaage3 < karaage.sql
```

- (c) Check data in karaage3 looks correct, before proceeding.

2. Check mysql data is correct.

- (a) Connect to mysql database:

```
mysql karaage3
```

- (b) Ensure all tables in database are using innodb and utf8 encoding:

```
mysql> use karaage3
mysql> show table status
```

- (c) If not all tables are using innodb and utf8, this can be corrected with the following script:

```
#!/bin/sh
set -x
```

```

DBNAME="karaage3"
USERNAME="karaage3"
PASSWORD="YYYYYYYY"

for i in $(mysql -B -N -e "SHOW TABLES" -u$USERNAME -p$PASSWORD $DBNAME); do
    mysql -B -N -e "SET foreign_key_checks = 0; ALTER TABLE $i ENGINE=InnoDB;
↪" -u$USERNAME -p$PASSWORD $DBNAME
    mysql -B -N -e "SET foreign_key_checks = 0; ALTER TABLE $i CONVERT TO_
↪CHARACTER SET utf8 COLLATE utf8_general_ci" -u$USERNAME -p$PASSWORD $DBNAME
done

```

(d) The above can also be done manually. For every table that is not innodb, convert it with:

```
mysql> ALTER TABLE table_name ENGINE=InnoDB;
```

For every table that is not utf8, convert it with:

```
mysql> ALTER TABLE table_name CONVERT TO CHARACTER SET utf8 COLLATE utf8_
↪general_ci
```

Some of these conversions may take some time to complete, especially for the cpufreq table.

(e) Ensure mysql data is correct without any obvious signs of problems.

3. (optional) If uncertain of upgrade, setup a test system. Read the steps in *Getting Started*. Do not set up any data stores yet. These steps will instruct you to install a mysql database, which isn't essential, however does help ensure karaage is working correctly at this point.
4. Otherwise, install the latest version of karaage3:

```
apt-get install karaage3
apt-get install python-mysqldb
```

5. Check the *DATABASES*, *LDAP*, *MACHINE_CATEGORY_DATASTORES* and *KG27_DATASTORE* settings in `/etc/karaage3/settings.py`. The mysql server should refer to the *new* database you created above, the LDAP server should refer to your production server.

```

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'karaage3',
        'USER': 'karaage3',
        'PASSWORD': 'YYYYYYYY',
        'HOST': '',
        'PORT': '',
        'ATOMIC_REQUESTS': True,
        'OPTIONS': {
            'sql_mode': 'STRICT_ALL_TABLES'
        },
    },
}

LDAP = {
    'default': {
        'ENGINE': 'tldap.backend.fake_transactions',
        'URI': 'ldap://production',
        'USER': 'cn=admin,dc=example,dc=org',
        'PASSWORD': 'XXXXXXXX',
        'REQUIRE_TLS': False,
    },
}

```

```

        'START_TLS' : False,
        'TLS_CA' : None,
    }
}

MACHINE_CATEGORY_DATASTORES = {
    'ldap' : [
        {
            'DESCRIPTION': 'LDAP datastore',
            'ENGINE': 'karaage.datastores.ldap.MachineCategoryDataStore',
            'LDAP': 'default',
            'ACCOUNT': 'karaage.datastores.ldap_schemas.openldap_account',
            'GROUP': 'karaage.datastores.ldap_schemas.openldap_account_group',
            'PRIMARY_GROUP': "institute",
            'DEFAULT_PRIMARY_GROUP': "dummy",
            'HOME_DIRECTORY': "/home/%(uid)s",
            'LOCKED_SHELL': "/usr/local/sbin/locked",
            'NUMBER_SCHEME': 'default',
            'LDAP_ACCOUNT_BASE': 'ou=Accounts,dc=example,dc=org',
            'LDAP_GROUP_BASE': 'ou=Groups,dc=example,dc=org',
        },
    ],
    'dummy' : [
    ],
}

KG27_DATASTORE = {
    'DESCRIPTION': 'LDAP datastore',
    'ENGINE': 'karaage.datastores.ldap.MachineCategoryDataStore',
    'LDAP': 'default',
    'ACCOUNT': 'karaage.datastores.ldap_schemas.openldap_kg27',
    'GROUP': 'karaage.datastores.ldap_schemas.openldap_kg27_group',
    'PRIMARY_GROUP': "institute",
    'DEFAULT_PRIMARY_GROUP': "dummy",
    'HOME_DIRECTORY': "/home/%(uid)s",
    'LOCKED_SHELL': "/usr/local/sbin/locked",
    'NUMBER_SCHEME': 'default',
    'LDAP_ACCOUNT_BASE': 'ou=People,dc=example,dc=org',
    'LDAP_GROUP_BASE': 'ou=Groups,dc=example,dc=org',
}

```

The values given for `LDAP_ACCOUNT_BASE` and `LDAP_GROUP_BASE` in `KG27_DATASTORE` should correspond with your existing LDAP data.

Warning: You must have LDAP configured correctly before you proceed to the migration step. The migrations reference LDAP information if it is available. If it is not available, the database may end up with incorrect information.

6. (optional) If you require people to be recorded in LDAP:

```

GLOBAL_DATASTORES = [
    {
        'DESCRIPTION': 'LDAP datastore',
        'ENGINE': 'karaage.datastores.ldap.GlobalDataStore',
        'LDAP': 'default',
        'PERSON': 'karaage.datastores.ldap_schemas.openldap_person',
    }
]

```

```
'GROUP': 'karaage.datastores.ldap_schemas.openldap_person_group',
'NUMBER_SCHEME': 'global',
'LDAP_PERSON_BASE': 'ou=People,dc=example,dc=org',
'LDAP_GROUP_BASE': 'ou=People_Groups,dc=example,dc=org',
},
]
```

7. Migrate DB tables.

If you have upgraded Django to 1.7 you will need to run south migrations first. Running this command is perfectly safe from Django 1.6, however not required. This must be run before running Django 1.7 migrations however.

```
apt-get install virtualenv          # Debian Jessie only
apt-get install python-virtualenv  # Debian Wheezy only
kg-migrate-south
```

Regardless of what version of Django you have installed, continue and run the native migrations.

```
kg-manage migrate
```

8. Check mysql data is correct.

(a) Connect to mysql database:

```
mysql karaage3
```

(b) Ensure all tables in database are using innodb and utf8 encoding:

```
mysql> use karaage3
mysql> show table status
```

(c) Ensure mysql data is correct without any obvious signs of problems.

9. If you have any other datastores, configure them now. See *Data Stores*.

10. Restart karaage processes.

```
service apache2 start
service karaage3-celery start
```

11. You can run the `migrate_ldap` command to check what LDAP changes are required:

```
kg-manage migrate_ldap --delete --dry-run
```

If you are happy, repeat without the `--dry-run` option.

You may want to consider not using the `--delete` option if you are still using the old data.

This need to be run before the upgrade can be considered complete, however most things should still work without doing this change.

12. Test. You should now be able to go to `http://hostname/kgadmin/`. Apply any local customizations you need to have a fully operational system now. Do not continue if you are not completely happy.

Final server configuration

Up to this point you have not made any changes to your production server, apart from installing the newer packages. If you are not happy with the upgrade, you can revert to your production system. After you pass this point, it will still

be possible to revert, only it will be slightly harder, as you have to undo the following steps. Once you proceed past this point, reverting may require restoring LDAP from the backup.

1. Run the following commands to check what LDAP changes are required:

```
kg-manage migrate_ldap --dry-run
```

Make the LDAP changes if happy:

```
kg-manage migrate_ldap
```

2. If required, install Karaage plugins. See [Plugins](#) for more information.
3. Test production box and make sure everything is working.

Other changes

Check that remote services that access Karaage work. For example, with `torque_submitfilter`, you need to update the Karaage URL to `/karaage/`.

You may need to update PBS/slurm logging to talk correctly to Karaage. Only do this if `kg-pbs-logger` was previously configured. Karaage no longer requires a dedicated account for `kg-pbs-logger`. Rather it uses the machine entry. For every machine:

1. (optional) Delete the dedicated account that was previously used for authenticating this system.
2. If using MAM, edit the `MACHINE_CATEGORY_DATASTORES` setting in `/etc/karaage3/settings.py`:

```
{
    'DESCRIPTION': 'MAM datastore',
    'ENGINE': 'karaage.datastores.mam.MamDataStore',
    ...
}
```

3. Navigate to machine entry in admin website.
4. Click password button to reset the password.
5. Update `/etc/karaage/pbs-logger.cfg` and for `ws_username` use the machine name in karaage, and for `ws_password` use the password obtained in the previous step.
6. You may need to reconfigure LDAP clients.
 - If *people* require access to service even if they don't have an account (rare):
`ou=People,dc=example,dc=org` and `ou=People_Groups,dc=example,dc=org`
 - If people only should get access with an *account* (most common):
`ou=Accounts,dc=example,dc=org` and `ou=People_Groups,dc=example,dc=org`

Upgrading from 3.0.x to 3.1.x

1. Upgrade packages. Run the following commands:

```
apt-get update
apt-get install karaage3
```

If asked if you want to run to run migrations, so No. This will happen later.

2. Merge in changes from `/etc/karaage/global_settings.py` into `/etc/karaage3/settings.py`. Or just copy the file should be sufficient.
3. If required, install Karaage plugins. See *Plugins* for more information.
4. Otherwise purge `karaage3-celery` if `kgusage` not required.
5. Any old Apache configuration might conflict with the new Apache configuration. Apache 2.2 in Wheezy uses `/etc/apache2/conf.d/karaage3-wsgi.conf`. Any other configuration files may be obsolete and should be deleted (or moved elsewhere).
6. Run the database migrations and restart apache.

If you have upgraded Django to 1.7 you will need to run south migrations first. Running this command is perfectly safe from Django 1.6, however not required. This must be run before running Django 1.7 migrations however.

```
apt-get install virtualenv          # Debian Jessie only
apt-get install python-virtualenv  # Debian Wheezy only
kg-migrate-south --noinput
```

Regardless of what version of Django you have installed, continue and run the native migrations.

```
kg-manage migrate
service apache2 reload
```

7. Remove old package that might cause problems.

```
dpkg --purge python-karaage-celery
```

8. Old packages and files can be removed. This is optional.

```
dpkg --purge karaage3-admin
dpkg --purge karaage3-registration
dpkg --purge python-karaage-doc
rm -rf /etc/karaage
rm -rf /var/log/karaage
rm -rf /var/log/karaage3-celery
rm -rf /var/cache/karaage
```

Other changes

- Check that remote services that access Karaage work. For example, with `torque_submitfilter`, you need to update the Karaage URL to `/karaage/`.
1. If using MAM, edit the `MACHINE_CATEGORY_DATASTORES` setting in `/etc/karaage3/settings.py`:

```
{
    'DESCRIPTION': 'MAM datastore',
    'ENGINE': 'karaage.datastores.mam.MamDataStore',
    ...
}
```

Contents:

Commands

Contents:

kg-manage

This command is used for managing karaage.

Usage

Basic usage:

```
kg-manage <command> [options]
```

The following is not a complete list of all commands available. To get a full list of commands, run the following:

```
kg-manage --help
```

To get help on a particular command, run the following:

```
kg-manage <command> --help
```

Django

dbshell

kg-manage dbshell

Enter db shell for administration. See [dbshell](#).

shell

kg-manage shell

Enter python shell for administration. See [shell](#).

migrate

kg-manage migrate

Run migrations on database. See [migrate](#).

Karaage Core

migrate_ldap

kg-manage migrate_ldap

Run migrations on LDAP servers.

--dry-run

Don't make any of the changes, display what would be done instead. Note the base dn objects will always be created.

--delete

Delete old records that are no longer used.

change_username

kg-manage change_username

Change the username for a person and related accounts.

changepassword

kg-manage changepassword

Change the password for a person and related accounts.

import_csv_users

kg-manage import_csv_users

Import people from a csv file.

kgcreatesuperuser

kg-manage kgcreatesuperuser

Create a superuser without an account.

lock_expired

kg-manage lock_expired

Automatically lock expired accounts.

Called automatically by *daily_cleanup*.

lock_training_accounts

kg-manage lock_training_accounts

Automatically lock training accounts.

unlock_training_accounts

kg-manage unlock_training_accounts

Automatically lock training accounts.

change_pid

kg-manage change_pid

Change a PID for a project.

daily_cleanup

kg-manage daily_cleanup

Daily cleanup for Karaage, should be called by cron job. This will automatically call all other applicable cleanup commands.

The exact commands executed depends on which plugins are configured. By default, will call *lock_expired*.

Karaage Applications Plugin

application_cleanup

kg-manage application_cleanup

Cleanup complete/old applications.

Called automatically by *daily_cleanup*.

Karaage Usage Plugin

clear_usage_cache

kg-manage clear_usage_cache

Delete the usage cache.

Called automatically by *daily_cleanup*.

clear_usage_graphs

kg-manage clear_usage_graphs

Delete the usagee graphs.

Called automatically by *daily_cleanup*.

link_software

kg-manage link_software

Automatically link software in usage table.

Called automatically by *daily_cleanup*.

kg-migrate-south

This command will install the latest South migrations.

Description

This command will install Django 1.6 and south in a virtualenv and then run kg-manage migrate to run all South migrations.

After running this command, it should then be safe to run *migrate* to run the Django 1.7 migrations.

This command should not be run except from upgrades from Django 1.6 or earlier to Django 1.7.

All options supported by the django south migrate command can be used here.

Requirements

This command requires virtualenv to be installed. On Debian Jessie install it with:

```
apt-get install virtualenv
```

On Debian jessie install it with:

```
apt-get install python-virtualenv
```

kg_set_secret_key

This command is use for resetting the secret key in `/etc/karaage3/settings.py`.

Description

To reset the secret key, run this command without any parameters:

```
kg_set_secret_key
```

Karaage Settings

There are many settings in `/etc/karaage3/settings.py` that can be customized for local requirements.

Django settings

Any Django settings can be used, although it is recommended only to modify the settings described in this section.

HTTP_HOST

Default: FQDN hostname

FQDN host, used in default settings for `ALLOWED_HOSTS`, `REGISTRATION_BASE_URL`, and `ADMIN_BASE_URL`.

Not a Django setting, but listed here regardless.

DEBUG

Default: `False`

Never deploy a site into production with `DEBUG` turned on.

Did you catch that? NEVER deploy a site into production with `DEBUG` turned on.

One of the main features of debug mode is the display of detailed error pages. If your app raises an exception when `DEBUG` is `True`, Django will display a detailed traceback, including a lot of metadata about your environment, such as all the currently defined Django settings (from `settings.py`).

See `DEBUG`.

ALLOWED_HOSTS

Default: `['%(HOST)']`

A list of strings representing the host/domain names that this Django site can serve. This is a security measure to prevent an attacker from poisoning caches and password reset emails with links to malicious hosts by submitting requests with a fake `HTTP Host` header, which is possible even under many seemingly-safe web server configurations.

`%(HOST)` will be substituted with the `HTTP_HOST` setting.

See `ALLOWED_HOSTS`.

SESSION_COOKIE_SECURE

Default: `True`

Whether to use a secure cookie for the CSRF cookie. If this is set to `True`, the cookie will be marked as “secure,” which means browsers may ensure that the cookie is only sent under an `HTTPS` connection.

See `SESSION_COOKIE_SECURE`.

ADMINS

Default: `()` (Empty tuple)

A tuple that lists people who get code error notifications. When `DEBUG=False` and a view raises an exception, Django will email these people with the full exception information. Each member of the tuple should be a tuple of (Full name, email address).

See `ADMINS`.

MANAGERS

Default: `()` (Empty tuple)

A tuple in the same format as `ADMINS` that specifies who should get broken link notifications when `BrokenLinkEmailsMiddleware` is enabled.

See `MANAGERS`.

DATABASES

Default: `{}` (Empty dictionary)

A tuple in the same format as `ADMINS` that specifies who should get broken link notifications when `BrokenLinkEmailsMiddleware` is enabled.

See `DATABASES`.

SERVER_EMAIL

Default: `'root@localhost'`

The email address that error messages come from, such as those sent to `ADMINS` and `MANAGERS`.

See `SERVER_EMAIL`.

EMAIL_HOST

Default: `'localhost'`

The host to use for sending email.

See `EMAIL_HOST`.

EMAIL_SUBJECT_PREFIX

Default: `'[Django] '`

Subject-line prefix for email messages sent with `django.core.mail.mail_admins` or `django.core.mail.mail_managers`. You'll probably want to include the trailing space.

See `EMAIL_SUBJECT_PREFIX`.

TIME_ZONE

Default: 'America/Chicago'

A string representing the time zone for this installation, or None. See the [list of time zones](#).

See [TIME_ZONE](#).

LANGUAGE_CODE

Default: 'en-us'

A string representing the language code for this installation. This should be in standard [language ID format](#). For example, U.S. English is "en-us". See also the [list of language identifiers](#).

See [LANGUAGE_CODE](#).

SECRET_KEY

Default: '' (Empty string)

A secret key for a particular Django installation. This is used to provide cryptographic signing, and should be set to a unique, unpredictable value.

See [SECRET_KEY](#).

LOGGING

Default: A logging configuration dictionary.

A data structure containing configuration information. The contents of this data structure will be passed as the argument to the configuration method described in [LOGGING_CONFIG](#).

See [LOGGING](#).

Django Pipeline settings

Pipeline is an asset packaging library for Django, providing both CSS and JavaScript concatenation and compression, built-in JavaScript template support, and optional data-URI image and font embedding.

PIPELINE_CSS_COMPRESSOR

Default: 'pipeline.compressors.yui.YUICompressor'

Django pipeline setting.

Compressor class to be applied to CSS files.

If empty or None, CSS files won't be compressed.

PIPELINE_JS_COMPRESSOR

Default: `'pipeline.compressors.yui.YUICompressor'`

Django pipeline setting.

Compressor class to be applied to JS files.

If empty or None, JavaScript files won't be compressed.

Karaage core settings

These are settings defined and used by Karaage core.

ACCOUNTS_EMAIL

Default: No default; must be set

Users are advised to contact this address if having problems. This is also used as the from address in outgoing emails.

ACCOUNTS_ORG_NAME

Default: No default; must be set

This organisation name, used in outgoing emails.

REGISTRATION_BASE_URL

Default: `'https://%(HOST)s/users'`

Registration base URL - Used in email templates.

`%(HOST)` will be substituted with the `HTTP_HOST` setting.

ADMIN_BASE_URL

Default: `'https://%(HOST)s/kgadmin'`

Admin base URL - Used in email templates.

`%(HOST)` will be substituted with the `HTTP_HOST` setting.

SHIB_SUPPORTED

Default: False

Is Shibboleth supported?

AUP_URL

Default: Django template `karaage/common/aup-detail.html`

Path to AUP policy. Note that setting this will not disable the Karaage default page, it might be better to replace the AUP with a file in the templates directory `karaage/common/aup-detail.html` if required.

ALLOW_REGISTRATIONS

Default: False

Plugin: kgapplications

Do we allow anonymous users to request accounts?

ALLOW_NEW_PROJECTS

Default: True

Plugin: kgapplications

Do we allow applications for new projects?

USAGE_IS_PUBLIC

Do we allow any logged in user to access all usage information?

PLUGINS

Default: [] (Empty list)

A list of classes that define Karaage plugins. For more information on creating plugins from scratch, please see the Karaage programmers documentation.

GLOBAL_DATASTORES

Default: [] (Empty list)

This is a list of dictionaries, that define the *global data stores*.

An example:

```
GLOBAL_DATASTORES = [
    {
        'DESCRIPTION': 'LDAP datastore',
        'ENGINE': 'karaage.datastores.ldap.GlobalDataStore',
        ...
    }
]
```

The settings for each datastore will vary depending on the value of ENGINE supplied. For more information, see *Data Stores*.

MACHINE_CATEGORY_DATASTORES

Default: {} (Empty dictionary)

This is a dictionary containing a list of dictionaries, that define the *machine category data stores*.

An example:

```
MACHINE_CATEGORY_DATASTORES = {
  'ldap': [
    {
      'DESCRIPTION': 'LDAP datastore',
      'ENGINE': 'karaage.datastores.ldap.MachineCategoryDataStore',
      ...
    },
  ],
  'dummy': [
  ],
}
```

The settings for each datastore will vary depending on the value of `ENGINE` supplied. For more information, see *Data Stores*.

KG27_DATASTORE

Default: None

Datastore used for upgrades from Karaage 2.7. For more information, see *Upgrading*.

This datastore is never written to, unless you have used the same settings for `GLOBAL_DATASTORES` or `MACHINE_CATEGORY_DATASTORES`.

LDAP

Default: {} (Empty dictionary)

This setting defines LDAP settings for a connection to a LDAP server. It is only used if you have configured `GLOBAL_DATASTORES` or `MACHINE_CATEGORY_DATASTORES` to use ldap.

An example:

```
LDAP = {
  'default': {
    'ENGINE': 'tldap.backend.fake_transactions',
    'URI': 'ldap://localhost',
    'USER': 'cn=admin,dc=example,dc=org',
    'PASSWORD': 'topsecret',
    'USE_TLS': False,
    'TLS_CA': None,
  }
}
```

USERNAME_VALIDATION_RE

Default: '[-\w]+'

Regular expression that defines a valid username for a *person* or an *account*.

Warning: Do not change unless you are sure you understand the potential security ramifications in doing so.

USERNAME_VALIDATION_ERROR_MSG

Default: 'Usernames can only contain letters, numbers and underscores'

Error message that is displayed to user if the username for a *person* or *account* doesn't pass the `USERNAME_VALIDATION_RE` check.

PROJECT_VALIDATION_RE

Default: '[-\w]+'

Regular expression that defines a valid username for *projects*.

Warning: Do not change unless you are sure you understand the potential security ramifications in doing so.

PROJECT_VALIDATION_ERROR_MSG

Default: 'Project names can only contain letters, numbers and underscores'

Error message that is displayed to user if a name for a *project* doesn't pass the `PROJECT_VALIDATION_RE` check.

GROUP_VALIDATION_RE

Default: '[-\w]+'

Regular expression that defines a valid name for a *group*.

Warning: Do not change unless you are sure you understand the potential security ramifications in doing so.

GROUP_VALIDATION_ERROR_MSG

Default: 'Group names can only contain letters, numbers and underscores'

Error message that is displayed to user if a name for a *group* doesn't pass the `GROUP_VALIDATION_RE` check.

Karaage applications settings

Settings specific to the Karaage applications plugin.

EMAIL_MATCH_TYPE

default: 'exclude'

Settings to restrict the valid list of email addresses we allow in applications. `EMAIL_MATCH_TYPE` can be 'include' or 'exclude'. If 'include' then the email address must match one of the RE entries in `EMAIL_MATCH_LIST`. If 'exclude' then then email address must not match of the the RE entries in `:setting:EMAIL_MATCH_LIST`.

EMAIL_MATCH_LIST

Default: [] (Empty list)

Settings to restrict the valid list of email addresses we allow in applications. `EMAIL_MATCH_TYPE` can be 'include' or 'exclude'. If 'include' then the email address must match one of the RE entries in `EMAIL_MATCH_LIST`. If 'exclude' then then email address must not match of the the RE entries in `:setting:EMAIL_MATCH_LIST`.

Appendices:

All notable changes to this project will be documented in this file. The format is based on [Keep a Changelog](#) and this project adheres to [Semantic Versioning](#).

3.1.31 - 2017-05-03

Changed

- Improve password fussiness.
- Find training accounts that are system users.
- TRAINING_ACCOUNT_PREFIX is now a regexp.

Fixed

- Applicants can have duplicate email and usernames.
- Various Django 1.10 fixes. Django >= 1.10 not yet supported.
- Improved error handling on approving applications.
- Update various dependencies, and fix related issues.
- Fix typo in emails. Administrator not Administrator.
- All tests pass under Django 1.10.
- Support tldap 1.4.1 and ldap3 2.2.3.
- Explicitly use bcrypt/pbkdf2_sha256 by default for passwords instead of SHA1. Update tests to use pbkdf2_sha256. Outside tests this was already the default with recent versions of Django.

3.1.30 - 2016-09-11

- Fix various errors.
- Fix broken calls to render.
- Fix validation logic, licence details, new software.
- Save institute even if commit==False.

3.1.29 - 2016-08-11

- Fix broken tests.
- Require at least one institute delegate.
- Make project leaders a required value.
- Don't crash if institute form invalid.
- Add mark_safe to required template tags.
- Strip leading and trailing space from input fields.
- Clarify purpose of Karaage password.
- Try to eliminate confusion in entering names.
- Make department field mandatory on applicant form.
- Update empty_text for similar_people_table.
- Don't output empty fields from Slurm.
- Requires Django >= 1.8
- Fix Django 1.10 compatibility warnings.

3.1.28 - 2016-05-11

- Ensure version.py gets installed.
- Ensure logout works with shibboleth.

3.1.27 - 2016-05-10

- Update programmer's documentation.
- Remove references to legacy documentation.
- Automatically fill username from shibboleth if we can.
- Various fixes for shibboleth work flow.
- Tests for login/logout.
- Display Karaage version in footnote.

3.1.26 - 2016-05-06

- Add shibboleth Apache configuration.
- Fix various shibboleth problems.
- Update shibboleth to use /Shibboleth.sso/Login instead of /Shibboleth.sso/DS.
- Display request.META values in SAML profile page.
- Fix render_link with latest Django Tables 1.2.0.
- Make first name and last name optional in shibboleth.
- Auto populate project application with email from shibboleth if possible.
- Redirect unknown shibboleth user to project application.
- Remove confusing shibboleth register button in application process.
- Make shibboleth login default in shibboleth enabled.
- Updates to programmer's documentation.

3.1.25 - 2016-05-03

- Move karaage.common.logging to common.logging, as we cannot initialize karaage.common at time logging is loaded with Django 1.9. Will require config change.

3.1.24 - 2016-05-03

- Updates to packaging.
- Updates to documentation.
- Fix tests for django_tables 1.2.0.
- Enable travis tests.

3.1.23 - 2016-04-29

- Fix tests and ensure everything still works.

3.1.22 - 2015-06-19

- Documentation updates.

3.1.21 - 2015-06-17

- Fix broken people list links.
- Fix Jessie references in documentation.
- Enhance `unlock_training_account` function.
- Add documentation on making new Karaage releases.

3.1.20 - 2015-06-05

- Override admin email addresses using `APPROVE_ACCOUNTS_EMAIL` setting.
- Fix flake8 tests in migrations.
- Change order of `deactivate()` function to avoid multiple updates to accounts.
- Fix issues with MAM datastore.
- Sort applications in admin list by reverse expiry date by default.
- Display `machine_category` in account lists.
- Don't allow editing project leaders through edit view.
- Don't allow revoking last project leader.
- Add new `ALLOW_NEW_PROJECTS` setting, if set to False user's will not be able to apply for new projects, only existing projects.

3.1.19 - 2015-05-29

- Numerous bug fixes.
- Display software stats correctly.
- Update documentation for Jessie.
- Update MAM and slurm documentation.
- Fix problems with latest slurm.
- Fix institute form.
- Support undelete project button.
- Fix display of leaders in bounce list.
- Set `date_approved` in approved applicants.
- Add more tests.
- Add HSTS to instruction.
- Change name of "Is existing person" button to "Mark duplicate user".
- Attempt to clarify emails.
- Add work around for ds389 bug. Note this won't work when adding a person and setting their password at the same time; in this case please manually reset the password to get it to work. https://bugzilla.redhat.com/show_bug.cgi?id=1171308

3.1.18 - 2015-04-13

- Django 1.8 and 1.9 fixes. * Minor Schema change to last_login field of Person and Machine. * Email length in Person increased. * Fix RelatedObject related issues in Applications. * Plus others.
- Fix bug in software application listing.
- Fix incorrect name of query and jquery-ui files.

3.1.17 - 2015-03-30

- Cleanup code.
- Cleanup css files and remove unused selectors.
- Support latest factory-boy.

3.1.16 - 2015-03-17

- Generate error if alogger does not supply project in usage.
- Rebuild static files when upgrading package.
- Extend application expiry after it is approved.
- Allow resetting password even if no password set.
- Django 1.6 support was broken in 3.1.15, now fixed.
- Fix default URLs.
- Simplify autoconfiguration of plugins.

3.1.15 - 2015-03-10

- Various bug fixes.
- Simplification of code, mainly alogger and tests.

3.1.14 - 2015-02-19

- Add missing depends.
- Fix errors in installation documentation.
- Add untested Active Directory schema support.

3.1.13 - 2015-02-17

- Fix package cleanup.
- Ensure config file not world readable.

3.1.12 - 2015-02-16

- New upstream release.
- Move plugins to karaage.plugins.
- Various minor bug fixes.

3.1.11 - 2015-02-12

- Merge plugins into one source.
- Merge kgapplications and kgsoftware into karaage package.

3.1.10 - 2014-12-01

- Bug fixes.
- Fix problems with django-pipeline 1.4.0.
- Updates to documentation.

3.1.9 - 2014-10-30

- Documentation: update apache configuration.
- Python3 fixes.
- UTF8 related fixes.
- Updates to upgrade documentation.

3.1.8 - 2014-10-13

- Fix daily cleanup. Work properly with plugins.
- Test daily cleanup.

3.1.7 - 2014-10-10

- Fix various MAM issues.
- Support MAM 2.7.

3.1.6 - 2014-09-30

- More Django 1.7 updates.
- Django 1.6 should continue to work. For now.
- migrate_ldap always creates global DN in ldap if required.
- Fix problems with logentry migrations.

3.1.5 - 2014-09-18

- Fix karaage3-database upgrade.
- Make work with Django 1.7
- Fix crash if no defined HTTP session with Django 1.6.
- We should fully support Django 1.7 now.

3.1.4 - 2014-09-15

- Updates to fix Django 1.7 issues.
- Django 1.7 should really work now, however upgrade from earlier versions not yet documented.

3.1.3 - 2014-09-09

- Rewrite migrate_ldap.
- Add Django 1.7 migration.
- Documentation updates.
- New kg-migrate-south command.
- Django 1.7 should work, however not yet recommended for production use.

3.1.2 - 2014-08-27

- Remove odd,even row classes.
- Fix broken templates.
- Move emails template directories.
- Move people template directories.
- Move machines template directories
- Move project template directories
- Move institutes template directories.
- Move common template directories

- Ensure migrate_ldap works properly with groups.
- Fix display of institute in migration.

3.1.1 - 2014-08-19

- Update documentation.
- Fix formatting.
- djcelery kludge.
- Split software out into plugin in karaagee-usage.
- Fix copyright.
- Use roles in applications.
- Fix project application specific wording.
- Make sure we include *.json files.
- Fix faulty role checks.
- Remove Django South hack.
- Make sure we kill the LDAP server after test fails.
- Fix migration errors.
- Turn karaage into one Django app.
- Fix management commands.
- Split applications into kgapplications.
- Update documentation.
- Fix migration issues.
- libapache2-mod-wsgi-py3 should be sufficient.
- Remove python2 specific use of iteritems.
- Remove software specific datastores.
- Combine templates.
- Cleanup links.
- Fix release tag.

3.1.0 - 2014-07-30

[Brian May] * Update software usage statistics. * Per institute software usage statistics. * Verbose logging when creating application accounts * Change link expiry text in emails.

[Andrew Spiers] * Fix typo in kg-daily-cleanup.rst

[Brian May] * userPassword should be text, not binary. * Fix strings for Python 3.2. * Make all strings in migrations “normal” strings. * Fix migrate_ldap operation. * Fix PEP8 issues. * Fix Python 3 compatibility issues. * Fix __unicode__ methods for Python 3. * Python 3 tests. * Python3 tracing change. * Disable usage / south stuff if not available. * Fix *all* PEP8 issues. * More Python3 syntax errors fixed. * Fix double quoted strings in migrations. *

Remove deprecated warnings. * Fix Python3 PEP8 errors. * Recommend mysql.connector.django over mysqldb. * Redo Debian packaging. * Support TLDAP 0.3.3 * Rename global_settings.py to settings.py * Copy and adapt file from django-xmlrpc. * Fix copyright declaration. * Declare Python 3 compatible. * Split usage stuff into kgusage. * Fix Debian packaging issues. * Update documentation. * Update plugin API. * Fix directory name in comment. * Change permissions for all of /var/cache/karaage3. * Don't run migrations unless Karaage is configured. * More changes to plugin API. * Add missing dpkg triggers. * Conceal stderr output from init.d script. * Apache2.2 and 2.4 autoconfiguration. * Depend on apache2. * Don't import debconf everywhere. * Simplify apache2.2 config. * Rename check() to check_valid(). * karaage3-apache supercedes old packages. * Disable django-south if not available. * Silence Django 1.7 upgrade warnings. * Add south to build depends. * Fix XMLRPC and add tests. * Remove legacy project_under_quota function. * Update changelog. * Add build depends on flake8. * Fix lintian issues and other problems. * Make tests optional. * Combine apache config files into one. * Rename karaage3-apache to karaage3-wsgi. * Move non-py files to common package. * Add lintian override for karaage3-wsgi. * Add lintian overrides for karaage3-database. * Modify Apache2.2 test.

3.0.15 - 2014-06-17

- Fix account detail page for admin.
- Change get_absolute_url for accounts.
- Paranoid security checks.
- Allow users to change default project.
- Remove depends on python.
- Ensure admin request emails have correct link.
- PEP8 improvement.
- Fix PEP8 issue in comment.
- Remove non-PEP8 compliant white space.
- Support searching multiple directories for gold.
- Display more project application details.

3.0.14 - 2014-05-27

- Put all tables inside table-container.
- Remove calc from css.
- Make headings more consistent.
- Update depends.
- Remove legacy stuff.
- Update LDAP documentation.
- Fix uninitialized is_admin value.
- Remove unneeded import.
- Ensure username is not included in the password.
- Revert "Use named URLs in get_email_link"
- Fix account permissions.

3.0.13 - 2014-05-05

- Specify python/debian mappings.
- Update migrations threshold.
- Remove duplicate active row.
- Fix incorrect link.
- Don't migrate if configure not called.
- Triggers for static files.

3.0.12 - 2014-05-01

[Brian May] * Remove unused file. * Fix PEP8 issues in initial config. * Update jquery. * Remove make_leader option from applicant form. * Don't set make_leader to False for new projects. * Display if this application has make_leader set. * Use python-pipeline to compress css and js files. * Fix display of icons. * Remove Javascript global variables. * New setting for debugging django-pipeline. * Change commented out value of ALLOW_REGISTRATIONS. * Create log files owned by www-data user. * Fix: Include header message in invitation. * Don't reset created_by on reopening application. * Simplify invite process. * Grant leader/revoke leader operations.

[Kieran Spear] * Honour 'make_leader' for application approval

[Brian May] * Use css style, instead of direct icon reference. * Use django-filter and django-tables2 for people. * Use django-filter and django-tables2 for institutes. * Use django-filter and django-tables2 for projects. * Use django-filter and django-tables2 for machines. * Use django-filter/django-tables2 for applications. * Use django-filter/django-tables2 for software. * Use django-filter/django-tables2 for logs. * Use django-filter/django-tables2 for usage. * Remove obsolete cruft. * Sort order of INSTALLED_APPS. * Replace gen_table with django_tables. * Use th instead of td for table headings. * Show extra buttons for inactive people. * datastore get_*_details don't error if not found. * Remove legacy code; self._person is always defined. * Remove legacy db table. * Fix migration error. * Active column for people. * More work on active/status indication. * Remove debugging. * Simplify account display. * Tidy code. * Tweak filters. * Improvements to pagination. * Use correct format specifier for minutes. * Show if person is admin or not in details page.

3.0.11 - 2014-04-10

[Brian May] * Fix replaces/breaks headers. * Test password reset procedure. * Add documentation for CLI commands. * Correct copyright statement.

[Kieran Spear] * Use named URLs in get_email_link * Don't hardcode login_url in login_required decorator

[Brian May] * Fix migration errors during upgrades from 2.7. * Update kgcreatesuperuser command: * Use new TLDP check_password method. * Fix application errors selecting projects. * Fix error saving group. * Fix errors changing passwords. * Test password change forms. * Ensure errors are emailed. * If applicant is admin let them edit application. * Use autocomplete to select leader/project. * Add "make leader" field to project select form.

3.0.10 - 2014-04-02

- Add migration to resize applicant.username.
- Fix typo in in 389 support.

- Update LDAP settings for latest TLDAP.
- Move kg-manage and kg-daily-cleanup from karaage-admin.
- Fix issue with datastore methods being called incorrectly.
- Validate group name for new institutes.
- Validate group name for new software.
- Update logging calls.

3.0.9 - 2014-03-25

[Russell Sim] * Increase max length of institute identifier to 255 * Increase max length of account username to 255 * Increase the max length of group name to 255 * Increase the max username length to 255 * Increase application username length to 255

[Brian May] * Don't use shell=True * Allow displaying of all errors.

[Russell Sim] * Fixed bug with incorrect mixin declaration

[Brian May] * Fix pep8 issues. * Institute graphs report unused space * Fix undefined variables. * Add test to change group in related objects.

[Russell Sim] * Fix failure when using cracklib * Moved test packages out of the install section * Added unit test base class * Better testing of institutional group changes * New logging API

[Brian May] * Use python logging. * Move project_trend_graph to projects directory. * Fix PEP8 issues. * Check if userapplication content type exists. * Fix PEP8 issues. * Add missing import. * Fix PEP8 issues. * Fix error referencing DoesNotExist. * Add missing import. * Fix PEP8 issues. * Add missing import. * Remove change_default_project xmlrpc function. * Fix more pep8 issues. * PEP8 fixes. * More PEP8 fixes. * PEP8 fixes. * Fix PEP8 issues in migrations. * PEP8 issue solved. * PEP8 issue solved. * Fix breakage introduced in PEP8 cleanup * Use django's validate_email function. * Update authors.

[Russell Sim] * Fixed flake8 check * Increase project pid to 255 * Better testing of project group changes

3.0.8 - 2014-03-14

- Remove REMOTE_USER middleware from karaage.middleware.auth Django now has django.contrib.auth.middleware.RemoteUserMiddleware and django.contrib.auth.backends.RemoteUserBackend.
- Fix error in calling log function in Applications.
- Test changes in Karaage source code with flake8.

3.0.7 - 2014-03-13

- Numerous fixes to logging.
- Fix password reset URL.
- Numerous errors fixed.
- Updates to documentation.
- Fix to SAML middleware.

- Fix account username validation.
- Fixes to renaming people and projects.
- Hide project edit button if not leader.

3.0.6 - 2014-03-11

- Various bugs fixed.
- Update python packaging.
- Rename Debian packages to Debian python compliant names.
- Add legacy packages for backward compatibility.

3.0.5 - 2014-03-03

- Start arranging code into correct modules.
- Display profile menu in top level profile page.
- Cosmetic changes.

3.0.4 - 2014-02-27

- Redesign datastores.
- Some small config changes required. See `/usr/share/doc/karaage3/NEWS`.
- Bugs fixed.
- New theme.

3.0.3 - 2014-02-24

- New release of Karaage.
- Updates to theme.
- Lots of bug fixes.
- Updates to documentation.
- Restructure the views.

3.0.2 - 2014-02-05

- Bugs fixed.
- Update documentation.
- Updates to installation procedures.

3.0.1 - 2014-01-30

- Various bugs fixed.
- Add unique constraints to usage caches.
- Usage uses django-celery.

3.0.0 - 2013-07-18

- MAJOR CHANGES. BACKUP EVERYTHING ***BEFORE*** INSTALLING. BACKUP MYSQL. BACKUP OPENLDAP. TEST YOU CAN USE RESTORE MYSQL AND OPENLDAP. TEST MIGRATIONS WORK ON TEST SYSTEM WITH REAL DATA BEFORE INSTALLING ON PRODUCTION BOX. MIGRATIONS MAY TAKE SOME TIME TO COMPLETE ON REAL DATA (ESPECIALLY IF CPUJob CONTAINS MANY ITEMS).
- Improved support for transactions.
- Various bugs fixed.
- Make mysql database authoritative over LDAP.
- Add is_locked field to Person and UserAccount.
- Add shell attribute to UserAccount.
- Add group model.
- Clean up data stores.
- Validate telephone numbers.
- Use dpkg triggers to migrate db changes.
- Update packaging.
- People don't have a LDAP entry unless they have an account.
- User's set password after account is created via password reset email.
- Use new methods stuff in tldap 0.2.7.
- We no longer require placard, change depends to depends on django-tldap.
- Remove project machine_category and machine_categories fields.
- Rename user fields to person.
- Rename ProjectCache.pid to ProjectCache.project
- Rename UserAccount to Account.
- Rename UserCache to PersonCache.
- Merge User db model/table into Person.
- For Project table, pid is no longer PK.
- Migrations for all of the above.
- Rewrite graphs.
- Existing LDAP entries for non-accounts will get deleted in db migration.
- URLs changed.

- Cleaned templates.
- Intergrate slurm/gold functionality as datastores.
- Simplify dependencies.
- Rewrite applications app.
- Anything not mentioned above was also changed.
- World peace is still to come.

2.7.6 - 2013-03-27

- Fix authentication for user's without a cluster account.
- Fix account expiry process.

2.7.5 - 2013-03-25

- Fix error creating new accounts.
- Reverse lock/unlock links when editing person.

2.7.4 - 2013-03-22

- Fix software data stores.
- Fix various errors initializing data for new users.
- Fix error in pbsmoab if user could not be found.

2.7.3 - 2013-03-15

- Don't support Python 2.5

2.7.3 - 2013-03-15

- Simplify default arguments.
- Remove duplicate initialization of machinecategory.
- Fix broken link in institute_form.html
- Update wiki link.
- Use GET for search, not POST.
- Fix confusion between person and accounts.
- debian

2.7.2 - 2013-02-19

- Tests all work now.

2.7.1 - 2013-02-11

- Increase the version number in `__init__.py`.

2.7.0 - 2013-02-11

- New version.
- Based on latest django-placard.
- Lots of changes to templates. Existing templates might not display correctly.

2.6.8 - 2012-11-19

- Fix error in template. Requires permissions to see `` end tag.

2.6.7 - 2012-11-14

- Fix placard templates, accidentally broken in last release.

2.6.6 - 2012-11-13

- Fix broken software email templates.
- Update loginShell form processing.
- Updates to django ajax selects stuff.

2.6.5 - 2012-10-16

- Update for latest django-ajax-selects.
- Remove obsolete code.
- Convert everything to use Django staticfiles.
- Make telephone number required in applicant form.
- Additional email address checks.
- Support Django 1.4.
- See <https://github.com/Karaage-Cluster/karaage/issues?milestone=2&state=closed>
- django-ajax-selects update

- project description
- Non-privileged admins can edit machine category
- latest django-ajax-selects support
- link_software error when unicode
- Error when no shell on unlocking
- Convert media files to staticfiles

2.6.4 - 2012-03-22

- See <https://github.com/Karaage-Cluster/karaage/issues?milestone=5&state=closed>
- Method to get a users projects via XML RPC
- Comments for Applications
- Don't allow people to join a project they are already a member of
- Project management as a project leader
- View pending project details before accepting
- Users stay in LDAP group when deleting project
- Set default project by webpage
- logging in takes you to home page
- application list doesn't display the application title
- Unlocking an account that is already unlocked
- Make default shell configurable
- Make bounced shell configurable
- Display application type in application table
- Multiple invitations to same email for same project
- Page 2 of applications on User site is Empty
- Project start date in form
- Deleted and Rejected applications
- Usage divide by zero issue
- Approve software request link doesn't show up
- SAML duplicate email error
- Spelling mistake.
- update project fails
- Machine Category usage cache errors

2.6.3 - 2012-02-07

- Jobname for a CPU Job increased to 256 characters
- Fixed bug for trend graphs when institute name had a / in it
- Ensure locked users can't change login shell
- Add users title to ldap
- Make names of software packages unique
- Log when user details are changed
- Added debconf question for DB migrations
- Added password reset function
- Allow project leaders to invite users to their projects
- Allow users to change their default project
- Show change password view on profile page
- Added management commands to lock/unlock training accounts

2.6.2 - 2011-10-19

- Handle module strings with // as a separator
- More filtering on software list
- Ensure usage index page is only accessible if allowed
- Other minor bug fixes

2.6.1 - 2011-08-30

- Fixed out by 1 error when calculating available cpus
- Added memory and core usage reports
- Fixed institute usage permission view
- More sensible redirect after accepting a license
- Added DB index to date field on CPUJob
- Fixed longstanding matplotlib project graph error

2.6 - 2011-08-02

- Institutes now have 0 or many delegates, got rid of active/sub delegates
- Removed deprecated requests app
- Refactor Account datastores. Setting now stored in DB
- Archive applications

- Ability to add/edit machine categories
- Reverse order of applications in admin site
- Set DEFAULT_FROM_EMAIL to be equal to ACCOUNTS_EMAIL
- Added software field to CPUJob
- Added CPU Job detail and list pages
- Send admin notification for pending project applications too
- Ability for an admin to modify an applicant
- Only create a group for a software package if it's restricted or has a license
- New management command to change a users username
- Added software usage statistics views
- Removed is_expertise field from projects
- Made the Send Email function more generic

2.5.17 - 2011-07-15

- Workaround for long standing matplotlib bug. Don't error if can't display graph.
- Fixed another SAML_ID unique bug

2.5.16 - 2011-06-27

- Fixed institute usage bug

2.5.15 - 2011-06-14

- Fixed bug in user invite email sending
- Fixed broken decline link in project applications
- Fixed bug in software detail template

2.5.14 - 2011-06-10

- Ability to view accepted licenses
- Fixed bug where utilisation only showed up after 2nd request
- Prevent saml_id and passwords from being edited in any forms
- Other minor bug fixes

2.5.13 - 2011-06-03

- Ensure SAML ID doesn't get set on new applications
- This fixes a serious bug

2.5.12 - 2011-06-03

- Project approved emails were going to the wrong place
- Log view for applications. Log against the parent Application model
- Add example setting for REGISTRATION_BASE_URL
- Minor bugs fixed

2.5.11 - 2011-06-01

- Ensure project PIDs and institute names don't clash
- Fixed bug in application invites
- Added Project decline functions
- Ensure institute name is unique. Ensure saml attributes are unique
- Have a variable for user site for url links in emails
- Refactored email templates. Use .example as suffixes

2.5.10 - 2011-05-25

- Fixed SAML entity ID bug when editing institutes
- Password encoding bug for AD fixed
- Project application workflows - Admin approval
- Admin context processor for pending app count
- Improvements in the institute form
- Ability to override UserApplicationForm

2.5.9 - 2011-05-18

- Fixed bug in graph generation when usage is unknown.
- Fixed bug in application saml institute logic
- Show unknow usage if user or project is NULL

2.5.8 - 2011-05-04

- Show all unknow usage function
- Set defaults for PERSONAL_DATASTORE and ACCOUNT_DATASTORES
- Use one template file for account approvals.
- Minor bug fixes

2.5.7 - 2011-03-30

- Project Caps, multiple caps allowed
- Got rid of need for unknown user and project for missing usage
- Added software datastore
- Fixed some LDAP caching issues
- Various bug fixes and RPM packaging improvements

2.5.6 - 2011-03-09

- Bug fixes
- Show saml ids in admin detail pages
- Changed create_password_hash to handle different formats

2.5.5 - 2011-03-08

- Added initial code for SAML support
- Don't assume LDAP in kgcreateuser command
- Add CAPTCHA to application forms if in use

2.5.4 - 2011-02-23

- Change default url for graphs to /karaage_graphs/
- Minor bug fixes

2.5.3 - 2011-02-21

- New application state ARCHIVE, handle multiple applications per applicant
- APPROVE_ACCOUNTS_EMAIL added
- Active Directory datastore

- Project applications
- Management command now deletes all applications that have been complete for 30 days.
- Ability to allow public access to usage information. Default is to keep restricted.
- Add CAPTCHA fields to application forms if no token and open registrations allowed.

2.5.2 - 2010-12-15

- Add transaction middleware
- Force close LDAP connection to avoid stale data
- Update person when changing default project
- Update homeDir on account update
- friendlier message when application not in correct state
- Delete the applicant associated with application on deletion
- Added logging for application state changes

2.5.1 - 2010-12-10

- Return distinct results in global search form
- Raise 403 error instead of 404 when application exists but is in wrong state.
- Force user sync for LDAP on changing default project
- Show secret token in admin view
- Use model auth backend too to support alogger and the likes
- Use andsome's is_password_strong method instead of own

2.5 - 2010-11-17

- Project Datastores
- Support for system users
- Machine scaling factor
- Handle Applications more generically and allow easier subclassing
- Institute datastores
- ProjectApplications
- Create default machine category when machines app is created
- Generate SECRET_KEY in new installations
- Many bug fixes throughout code

2.4.14 - 2010-11-17

- Added CSV user import command
- Ensure applicant with same email doesn't exist when inviting
- Minor bug fixes

2.4.13 - 2010-10-20

- Make sure invitation isn't expired
- Send different email if existing user on account creation
- Usage bug fixes
- allow admin to change application request options
- optional redirect after changing default project
- Ability to delete applications in admin view
- Spelling mistakes
- Other various bug fixes

2.4.12 - 2010-10-13

- Make header_message required field in application invite form
- Only show software that has a license for users to accept
- Bug fixes

2.4.11 - 2010-10-07

- Select related to lessen SQL queries
- Fixed bug in log parser if user has two accounts
- Allow existing users to apply for new projects
- Added project application form
- Ensure Applicant email is unique
- Changes to ProjectApplication model
- Display pending applications to project leaders in profile
- Allow project leader to select 'needs account'
- Pending applications for user plus decline applications
- Name of NEW state is Invitation sent
- More explicit confirm when inviting users that already exist in system

2.4.10 - 2010-10-04

- Fixed serious circular import bug

2.4.9 - 2010-09-29

- New Application app
- Fixed bug in password done template
- Other minor fixes

2.4.8 - 2010-09-15

- Added memory and core usage reports
- Use django-ajax-selects
- Use new messaging framework
- Ability to change is_staff and is_superuser
- Bug fixes and code cleanup

2.4.7 - 2010-08-25

- Use django-andsome baseurl context
- Bug fixes

2.4.6 - 2010-08-25

- Added ability to request software.
- Cleaned up permission system on who can view what
- Moved project usage URL
- Bug fixes

2.4.5 - 2010-08-17

- Use BigInteger field in usage fields
- Fix import error in request forms

2.4.4 - 2010-08-12

- Set django password to unusable once user has password in ldap
- Removed required fields on most user form fields.
- Only able to change password if user is unlocked. Fixes #63
- Remove hardcoded link to VPAC usage graph.
- Other small bug fixes

2.4.3 - 2010-07-28

- Make kgcreatesuperuser script smarter
- Don't error if graphs not implemented in specific library

2.4.2 - 2010-07-28

- Dropped support for Django 1.1.1
- LOGIN_URL settings move to karaage-admin

2.4.1 - 2010-07-27

- Added command to create a karaage superuser
- Make LDAP Auth backend the default
- If no logged in user log events under the new user
- Make country field optional on Person model

2.4 - 2010-07-27

- Minor config changes
- Changes to default settings for new installs
- Bug fixes to project form

2.3.11 - 2010-07-21

- Compatible with Django 1.1
- Other tweaks to default configuration.
- Minor updates to configuration.
- Add script to set default secret.

2.3.10 - 2010-07-20

- Change to non-native format.
- Use new configuration system.
- Other improvements to packaging.

2.3.9 - 2010-06-08

- Fixed syntax error

2.3.8 - 2010-06-08

- Fixed Django 1.2 incompatibility

2.3.7 - 2010-05-31

- Remove username from account creation form, fixes #43.
- Allow searching for project ID's in choose project that are longer than 8 characters

2.3.6 - 2010-05-28

- Removed comment field from request detail
- Only activate a user if not already active

2.3.5 - 2010-05-28

- Fix issue of not being able to search from page 2+ in userlist, Fixes #40
- Fixed #44 </tr> tag now in correct place for valid html
- Fixed bug in get_available_time and created a test to make sure it's correct
- Allow PID to be specified in admin project form
- More testing

2.3.4 - 2010-05-26

- Decreased verbosity in management scripts
- Split user forms up one with username/password, one without
- Don't update datastore when saving a user in create script
- Only require required attributes in create_new_user method

- Moved to using django-simple-captcha instead of custom one
- Changed ordering when updating users in ldap datastore. Fixes #41
- More unit tests

2.3.3 - 2010-05-19

- Gecos and gidNumber are now also configurable via ldap_attrs

2.3.2 - 2010-05-19

- Pull in django-south dependency

2.3.1 - 2010-05-19

- Use active institutes in forms

2.3 - 2010-05-19

- Use Django-south for DB migrations
- Added is_active field to Institute

2.2.1 - 2010-05-17

- Fixed create_account bug with ldap_attrs

2.2 - 2010-05-17

- Use dynamic values when creating an LDAP account. Also supply default_project when creating accounts
- Code clean up
- Added unittests for people and set up testing framework and project
- Bugfix for graphs when no machines
- Quota equals zero bug and signals to add IntituteChuck automatically
- Show jobID in default usage list.
- Added pylint file

2.1.1 - 2010-05-07

- Ability to set LOCKED_SHELL. Fixes #34
- objectClass now configurable

2.1 - 2010-05-06

- Changed size of cpu_job.jobname from 20 -> 100. REQUIRES DB change
- removed is_expertise from user project form
- By default expect a non expertise project when creating a project ID
- Removed VPAC in text on admin person form

2.0.16 - 2010-05-05

- Better way of checking to see if user is locked or not

2.0.15 - 2010-05-05

- Removed VPAC specific lock DN

2.0.14 - 2010-05-03

- Fixed usage bug when no projectchunk
- Fixed JS broken link on project form

2.0.13 - 2010-05-03

- Added initial data for default MachineCategory. Fixes #31
- Added initial api docs
- Added some management commands for clearing and populating usage cache and locking expired users
- Don't display title if it doesn't exist. Fixes #30

2.0.12 - 2010-04-29

- Attempts to fix usage error. addresses #25

2.0.11 - 2010-04-28

- Fixed broken graph urls

2.0.10 - 2010-04-28

- Provide GRAPH_URL in template context

2.0.9 - 2010-04-28

- GRAPH_URL and GRAPH_ROOT settings if graph dir separate to MEDIA_*

2.0.8 - 2010-04-01

- Fixed software_detail bug

2.0.7 - 2010-03-31

- Fixed totals displaying in usage_institute_detail page
- Use new django aggregation support instead of raw sql
- Show project usage based on machine_category

2.0.6 - 2010-03-24

- Fixed bug in project reports url redirection

2.0.5 - 2010-03-22

- Depend on django-xmlrpc package

2.0.4 - 2010-03-22

- Actually use new alogger library

2.0.3 - 2010-03-22

- Depend on python-logger

2.0.2 - 2010-03-19

- Fixed gdchart2 requirment

2.0.1 - 2010-03-19

- Initial release.

account A person may have one or more accounts. An account allows a person to access *machines* on a given *machine category*.

administrator A person who has unlimited access to Karaage.

data store A list of external databases that we should link to and update automatically. Supported databases include LDAP, MAM, and Slurm.

global data store A *data store* for storing global data. The global datastores are responsible for writing global data, such as *people* (not *accounts*) to external databases such as LDAP.

group A list of *people*. Usually maps directly to an LDAP Group, but this depends on the data stores used.

institute An entity that represents the organisation or group that every *person* and *project* belongs to.

institute delegate A person who manages an *term:institute*, and can allow new *project's* for the institute.

machine A single cluster or computer which is managed as a distinct unit.

machine category A group of *machines* that share the same authentication systems.

machine category data store A *data store* for storing *machine category* specific data. The machine category datastores are specific to a given machine, and are responsible for writing machine category specific data, such as *accounts* (not *people*) to external databases such as LDAP.

person A person who has access to the Karaage system. A person could have one/more accounts, be an administrator, be a project leader, be an *institute delegate*. These are optional.

project A list of *people* who share a common goal.

project leader A person who manages a *project*, and can allow new user's to use the project.

CHAPTER 11

Indices and tables

- `genindex`
- `search`

Symbols

- delete
 - django-admin command-line option, 34
- dry-run
 - django-admin command-line option, 34

A

- account, 75
- ACCOUNTS_EMAIL
 - setting, 40
- ACCOUNTS_ORG_NAME
 - setting, 40
- adding
 - machine, 23
 - machine category, 23
 - person, 24
- ADMIN_BASE_URL
 - setting, 40
- administrator, 75
- ADMINS
 - setting, 37
- ALLOW_NEW_PROJECTS
 - setting, 41
- ALLOW_REGISTRATIONS
 - setting, 40
- ALLOWED_HOSTS
 - setting, 37
- application_cleanup
 - django-admin command, 35
- AUP_URL
 - setting, 40

C

- change_pid
 - django-admin command, 35
- change_username
 - django-admin command, 34
- changepassword
 - django-admin command, 34

- clear_usage_cache
 - django-admin command, 35
- clear_usage_graphs
 - django-admin command, 36
- commands
 - main, 33

D

- daily_cleanup
 - django-admin command, 35
- data store, 75
 - gold, 11
 - main, 9
 - openldap, 12
 - slurm, 17
- DATABASES
 - setting, 38
- dbshell
 - django-admin command, 34
- DEBUG
 - setting, 37
- django-admin command
 - application_cleanup, 35
 - change_pid, 35
 - change_username, 34
 - changepassword, 34
 - clear_usage_cache, 35
 - clear_usage_graphs, 36
 - daily_cleanup, 35
 - dbshell, 34
 - import_csv_users, 34
 - kgcreatesuperuser, 34
 - link_software, 36
 - lock_expired, 35
 - lock_training_accounts, 35
 - migrate, 34
 - migrate_ldap, 34
 - shell, 34
 - unlock_training_accounts, 35
- django-admin command-line option

- delete, 34
- dry-run, 34

E

- EMAIL_HOST
 - setting, 38
- EMAIL_MATCH_LIST
 - setting, 43
- EMAIL_MATCH_TYPE
 - setting, 43
- EMAIL_SUBJECT_PREFIX
 - setting, 38

G

- global data store, 75
- GLOBAL_DATASTORES
 - setting, 41
- gold
 - data store, 11
- group, 75
- GROUP_VALIDATION_ERROR_MSG
 - setting, 43
- GROUP_VALIDATION_RE
 - setting, 43

H

- HTTP_HOST
 - setting, 37

I

- import_csv_users
 - django-admin command, 34
- institute, 75
- institute delegate, 75

K

- Karaage
 - features, 3
- KG27_DATASTORE
 - setting, 42
- kgcreatesuperuser
 - django-admin command, 34

L

- LANGUAGE_CODE
 - setting, 39
- LDAP
 - setting, 42
- link_software
 - django-admin command, 36
- lock_expired
 - django-admin command, 35
- lock_training_accounts

- django-admin command, 35

- LOGGING
 - setting, 39

M

- machine, 75
 - adding, 23
- machine category, 75
 - adding, 23
- machine category data store, 75
- MACHINE_CATEGORY_DATASTORES
 - setting, 41
- mam, *see* gold
- MANAGERS
 - setting, 38
- migrate
 - django-admin command, 34
- migrate_ldap
 - django-admin command, 34

O

- openldap
 - data store, 12

P

- person, 75
 - adding, 24
- PIPELINE_CSS_COMPRESSOR
 - setting, 39
- PIPELINE_JS_COMPRESSOR
 - setting, 39
- PLUGINS
 - setting, 41
- project, 75
- project leader, 75
- PROJECT_VALIDATION_ERROR_MSG
 - setting, 43
- PROJECT_VALIDATION_RE
 - setting, 43

R

- reference
 - main, 32
- REGISTRATION_BASE_URL
 - setting, 40

S

- SECRET_KEY
 - setting, 39
- SERVER_EMAIL
 - setting, 38
- SESSION_COOKIE_SECURE
 - setting, 37

- setting
 - ACCOUNTS_EMAIL, 40
 - ACCOUNTS_ORG_NAME, 40
 - ADMIN_BASE_URL, 40
 - ADMINS, 37
 - ALLOW_NEW_PROJECTS, 41
 - ALLOW_REGISTRATIONS, 40
 - ALLOWED_HOSTS, 37
 - AUP_URL, 40
 - DATABASES, 38
 - DEBUG, 37
 - EMAIL_HOST, 38
 - EMAIL_MATCH_LIST, 43
 - EMAIL_MATCH_TYPE, 43
 - EMAIL_SUBJECT_PREFIX, 38
 - GLOBAL_DATASTORES, 41
 - GROUP_VALIDATION_ERROR_MSG, 43
 - GROUP_VALIDATION_RE, 43
 - HTTP_HOST, 37
 - KG27_DATASTORE, 42
 - LANGUAGE_CODE, 39
 - LDAP, 42
 - LOGGING, 39
 - MACHINE_CATEGORY_DATASTORES, 41
 - MANAGERS, 38
 - PIPELINE_CSS_COMPRESSOR, 39
 - PIPELINE_JS_COMPRESSOR, 39
 - PLUGINS, 41
 - PROJECT_VALIDATION_ERROR_MSG, 43
 - PROJECT_VALIDATION_RE, 43
 - REGISTRATION_BASE_URL, 40
 - SECRET_KEY, 39
 - SERVER_EMAIL, 38
 - SESSION_COOKIE_SECURE, 37
 - SHIB_SUPPORTED, 40
 - TIME_ZONE, 38
 - USAGE_IS_PUBLIC, 41
 - USERNAME_VALIDATION_ERROR_MSG, 42
 - USERNAME_VALIDATION_RE, 42
 - from 2.7.x, 25
 - from 3.0.x, 31
 - main, 24
 - USAGE_IS_PUBLIC
 - setting, 41
 - USERNAME_VALIDATION_ERROR_MSG
 - setting, 42
 - USERNAME_VALIDATION_RE
 - setting, 42
- shell
 - django-admin command, 34
- SHIB_SUPPORTED
 - setting, 40
- slurm
 - data store, 17
- T**
- TIME_ZONE
 - setting, 38
- U**
- unlock_training_accounts
 - django-admin command, 35
- upgrading