Installation manual for ESGF Middleware

Bugs, fixes and everything in between!

Prashanth Dwarakanath

National Supercomputer Center, Linköping Sweden

Contents

Chapter

1 ESGF Test-bed Setup

In this chapter, we'll briefly get acquainted with the various components in a full-fledged ESGF setup and learn about configuring them

1.1 Components

There are four major components in the full ESGF Node software stack. They are:-

1. idp
2. data
3. index
4. compute

1.1.1 Identity provider: idp

The idp node comprises of :-

- **OpenID identity provider web application**: It allows users to register and authenticate with the system, including single-sign-on functionality for browser-based access throughout the federation.
- **Globus SimpleCA and MyProxy**: MyProxy is used to create short term certificates from certs which are signed with the Globus SimpleCA.

1.1.2 Data node

The data node comprises of :-

- **ESGF Node Manager**: Web app to enable peer-to-peer interaction among nodes across the federation. Creates the ESGF registry which contains service endpoint lists.
- **ESGF Publisher and Postgres database**: Desktop app that allows users to publish data into a node. Metadata from files is extracted and THREDDS XML catalogs created. Postgres database used.
- **Thredds Data Server and ESGF Security filters**: TDS is the standard mechanism for serving data in various forms and protocols. Security built-in.
- **GridFTP server**: High performance gsi-enabled data transfer.

1.1.3 Index node

The index node comprises of :-

- **Apache Solr**: High performance web-app for storing and searching metadata.
- **ESGF Search back-end utils**: For harvesting external metadata repsos such as THREDDS XML catalogs produced by ESGF Publisher), and for searching metadata indexs deployed within the federation.
- **ESGF Web Portal**: A full blown website for users.

1.1.4 Compute Node

The compute node comprises of :-

- **Live Access Server**: Visualization software

1.2 PreSetup

The esgf-node installation script does not come with an uninstall script, to return the node to the state prior to commencement of installation. This can be inconvenient. Tripwire can be used to track all the changes made to the system by the installation and the changes can be reverted manually. For instructions on how to install, refer to Section 4.2

1.3 Running the script

All the commands in this section are to be run as root user. Exceptions will be mentioned clearly. For our installation, we'll try to setup all of the roles on the same machine.

1.4 Installing the middleware

1. Create a directory esgf-stuff and change into it.
2. Fetch the esgf-bootstrap script from http://rainbow.llnl.gov/dist/esgf-installer/esg-bootstrap
3. Run the bootstrap script. It will check for updates if any, and proceed to install a binary called esg-node in /usr/local/bin.
   `bash esg-bootstrap`
4. Choose the role you wish to install (data/index/idp/compute/all) and run the esg-node binary with appropriate arguments.
   `esg-node --type all --install`

1.4.1 Filling in the interactive questionnaire

Prompt Question	Value
Admin password	<the password>
Organization name	orgname
Short name	shortname
Descriptive long name	longname
Namespace	network domain
Node peer group	peer group of your choice
Default peer	hostname of default peer
Index peer	hostname of index peer
Email id	email id of admin
Is database external to this node?	no
Database connection string	postgresql://dbsuper@localhost:5432/esgcet
Low-priv db account	esgcet
db password for publisher user	<some other password>
Password for dbsuper role	<the password>

If you enter the password for the dbsuper role incorrectly, installation will fail.

1.4.2 Bug in sqlalchemy

This bug was fixed in later versions of the esgf installer. For details about the bug, refer to Section 3.1

1.4.3 Filling in the interactive questionnaire (contd)

Prompt Question	Value
Would you like to use the DN: (OU=ESGF.ORG, O=ESGF) ? [Y/n]	Y
Enter key password for my_esgf_node	CA passphrase
Do you wish to generate a Certificate Signing Request at this time?	Y

1.5 Bug in install script for 'data-only installation

A bug was discovered, when a 'data-only' setup of ESGF middleware was attempted. For details, refer to Section 3.2

1.6 Sequence of package installation for datanode

1. openssl
2. curl
3. git
4. java
5. ant
6. posgresql
7. cdat
8. esgcet
9. tomcat
10. thredds
11. node manager
12. dashboard services
13. dashboard ip
14. esgf desktop
15. openid relaying party
16. esgf security
17. idp
18. globus gridftp
19. globus myproxy
20. esg search (solr)
21. esg-web-frontend
22. ferret
23. las

1.7 Installing the middleware: Brower Park Release

The much awaited v1.5 is here! The full version identifier is v1.5.0-brower_park-release-3-g9a0c6de-master. Despite the discussions on the mailing list talking about blown bash stacks, I decided to do an all-role install and brave it out. The installation proceeded without any hitches, till it came to globus myproxy install/config part where it came to the simple ca setup part and just stopped. `top` revealed a simpleca setup attempt which was going nowhere. I ctrl-ced it and did the install again, this time with `--force` turned on, nixed everything including the simpleca setup and then the installer completed the installation. The installation questionnaire hasn't changed much, so Section 1.4 is still relevant. The sequence of package installation is the same as described in Section 1.6. I manually setup simpleca as described in Section 1.8.1. There is a bug in this version that causes thredds data server to not start. For the fix, refer to Section 3.7.

1.8 Setting up self-signed certificates

This section is only for those intending to deploy a stand-alone test deployment. For a production deployment, simply generate a CSR and submit it to Gavin for signing.

To setup a completely self-contained esgf installation, we first need to setup local trust, i.e. self-signed certificates. Let's explore how certificates are managed in the esgf middleware. The certificate directory is /usr/local/tomcat/conf. The private key is hostkey.pem. The

setup creates <fqdn.csr> certificate signing request file which is to be signed by the CA. In this case, we intend to self-sign it. After signing the certificate, the script recommends using the esg-node --installed-keypair <private key> <signed cert> command, which is what we hope to use, when ready.

1.8.1 Getting our hands dirty with the certs

We start by customizing globus simple ca. This is done by:-

1. Export correct value for GLOBUS_LOCATION
2. If your Simple CA is already installed, you may skip to point 5.
3. Execute /usr/local/globus/setup/globus/setup-simple-ca
4. Set unique subject name for CA like **cn=Globus Simple CA, ou=ESGF.ORG, o=ESGF**. Use a cn which is appropriate for your setup, but keep the ou and o parts as shown in the example.
5. Run the following command:

```
/usr/local/globus/setup/globus_simple_ca_<certhash>_setup/setup-gsi
```

6. Follow up with this command:

```
/usr/local/globus/bin/grid-ca-sign -in \\
<csrfile from tomcat directory> -out ./hostcert.pem
```

1.8.2 Installing your newly signed certificate for use with Tomcat

The instructions in this section are the same, whether you are trying to install your own temporary self-signed certificates, or if you have obtained signed certificates which are valid across the ESGF trust fabric.

1. Copy the newly generated certificate into the tomcat config directory[1] and then execute the following:

```
esg-node --install-keypair ./hostcert.pem ./hostkey.pem
```

2. You'll be asked to set the keystore password. Set it.
3. You'll be asked for the Certificate Authority's chain file. Here's what you need to know:
   - This is the CA's own certificate. This file is already present on your system and will be present in the `/etc/grid-security/certificates` directory, as a <certhash>.0 file.

[1] It's in `/esg/config/tomcat` since Brower Park. It used to be in `/usr/local/tomcat/config` earlier.

- On your new host certificate, execute `openssl x509 -noout -issuer_hash -in ./hostcert.pem`. This will produce a hash value as the output. That is the hash of the CA certificate you need.
- Provide the full path of that file, which is `/etc/grid-security/certificates/<certhash>.0` and hit enter.
- Instructions in this section assume you have a host certificate called 'hostcert.pem'. If that is not the case, use the appropriate file name!

4. It prompts you to replace current tomcat keystore with /usr/local/tomcat/conf/keystore-tomcat, if necessary. You don't need to do anything.

5. Hit enter and after a while, tomcat will be setup with the new certificates and it'll prompt for a node restart. Execute 'esg-node restart' at this point.

1.9 esgf-sh commands

Commandline	Action Description
su	switch to admin mode
show -ag	show all groups
show -au	show all users
show -d -u <username>	show all info about user
show -ar	show all roles
show -all	show all information
associate -add -r <rolename> -u <username> -g <groupname>	user/group/role mapping
groupadd -auto -d <description> -vis <groupname>	create a visible group to which users can add themselves

1.10 Useful openssl commands

Commandline	Action Description
openssl ca -config /.globus/simpleCA/grid-ca-ssl.conf -revoke /.globus/simpleCA/newcerts/01.pem	cert revocation
openssl pkcs12 -in <p12 file> -clcerts -nokeys -out usercert.pem	p12 to pem
openssl pkcs12 -in <p12 file> -nocerts -out userkey.pem	p12 to pem
openssl x509 -noout -modulus -in certicate.crt | openssl md5	verification
openssl rsa -noout -modulus -in privateKey.key | openssl md5	verification
openssl req -noout -modulus -in CSR.csr | openssl md5	verification
openssl x509 -noout -issuer_hash -in ./hostcert.pem	output ca hash

Chapter

2 Startup Notes

2.1 MyProxy fails to start

Refer to Section 3.3 for details on how to solve this issue.

2.2 Apache Solr fails to start

The startup script jumps the gun in not waiting for the service to actually start before reporting a failure. The service is starting up normally. Refer to Section 3.4 for details.

2.3 Fetching federation trust store fails

Would probably also lead to Apache not listening on port 443. May happen if chosen trust store password is different from the master password. Reinstall tomcat and configure with master password.

2.4 Apache not listening on port 443

Refer to Section 2.3

2.5 esgf-dashboard, esg-orp produces 404 error

Refer to Section 3.6

2.6 esg-node –test-pub fails

esg-node --test fails with '`rootAdmin is not authorized to publish/unpublish`...' error. Refer to Section 3.6

2.7 Thredds server fails to start up

This is a problem seen in version 1.5 aka Brower Park release of the esgf middleware. Refer to Section 3.7 for the solution.

Chapter

3 Caveats

This chapter lists out the bugs/caveats encountered during various stages of setting up the ESGF data node services.

3.1 Bug in sqlalchemy

Status: Fixed in Brower Park release.
It was observed that the installation was failing after a python exception. This bug was still present, as of 16th January. This document will be updated once the bug is addressed. Error extract given below:

```
Installed /usr/local/cdat/lib/python2.6/site-packages/esgf_dashboard-0.0.1-py2.
6.egg
Processing dependencies for esgf-dashboard==0.0.1
Finished processing dependencies for esgf-dashboard==0.0.1
Traceback (most recent call last):
  File "/usr/local/cdat/bin/esgf_dashboard_initialize", line 5, in <module>
   pkg_resources.run_script('esgf-dashboard==0.0.1', 'esgf_dashboard_initialize')
  File "build/bdist.linux-x86_64/egg/pkg_resources.py", line 489, in run_script
  File "build/bdist.linux-x86_64/egg/pkg_resources.py", line 1207, in run_script
  File "/usr/local/cdat/lib/python2.6/site-packages/esgf_dashboard-0.0.1-py2.6
.egg/EGG-INFO/scripts/esgf_dashboard_initialize", line 8, in <module>
    from migrate.versioning.shell import main as versioning_main
  File "/usr/local/cdat/lib/python2.6/site-packages/sqlalchemy_migrate-0.6-py2.6
.egg/migrate/versioning/shell.py", line 11, in <module>
    from migrate.versioning import api, exceptions
  File "/usr/local/cdat/lib/python2.6/site-packages/sqlalchemy_migrate-0.6-py2.6.egg/
migrate/versioning/api.py", line 32, in <module>
    from migrate.versioning import (exceptions, repository, schema, version,
```

```
  File "/usr/local/cdat/lib/python2.6/site-packages/sqlalchemy_migrate-0.6-py2.6.egg/
migrate/versioning/schema.py", line 10, in <module>
    from sqlalchemy import exceptions as sa_exceptions
ImportError: cannot import name exceptions
ERROR: Could not create esgf dashboard database tables in esgcet
```

A google search on the import error revealed a possible bug. Upon modifying the offending import statement to 'from sqlalchemy import exc as sa_exceptions', the import error disappeared.

3.2 Bug in install script for 'data-only' installation

Status: Fixed in Brower Park release.
The installer logged the following:

```
Oops, Don't see ROOT web application

*******************************
Setting up Apache Tomcat...(v6.0.36) ROOT webapp
*******************************

Downloading ROOT application from http://198.128.245.140/dist/ROOT.tgz
 Hmmm... Could not find local file /usr/local/src/esgf/workbench/esg/ROOT.tgz
Fetching file from http://198.128.245.140/dist/ROOT.tgz -to-> ROOT.tgz
--2013-01-16 14:56:09--  http://198.128.245.140/dist/ROOT.tgz
Connecting to 198.128.245.140:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1213 (1.2K) [text/plain]
Saving to: 'ROOT.tgz'

100%[======================================>] 1,213       --.-K/s   in 0s

2013-01-16 14:56:09 (55.3 MB/s) - 'ROOT.tgz' saved [1213/1213]

[VERIFIED]
unpacking ROOT.tgz...
tar: /usr/local/tomcat/webapps: Cannot chdir: No such file or directory
tar: Error is not recoverable: exiting now
 ERROR: Could not extract /usr/local/src/esgf/workbench/esg/ROOT.tgz
```

A work-around was found, by manually calling the setup_tomcat and setup_java routines. The patch file which I submitted to the esgf-user list was as follows:

```
diff --git a/esg-node b/esg-node
index ecbf2ce..3865197 100755
--- a/esg-node
```

```
+++ b/esg-node
@@ -2153,6 +2153,9 @@ setup_root_app() {
     pushd ${workdir} >& /dev/null

     echo "Downloading ROOT application from ${root_app_dist_url}"
+    echo "Trying to circumvent installer bug by setting up tomcat and java myself"
+    setup_tomcat;
+    setup_java;
     checked_get ${root_app_dist_url}
     (( $? > 1 )) && echo " ERROR: Could not download ROOT app archive" && popd && check
     echo "unpacking ${root_app_dist_url##*/}..."
```

3.3 Borked MyProxy config file

Status: Fixed in Brower Park release.
A bug in the installer script resulted in junk characters being written into /etc/myproxy-server.config file, instead of the actual passphrase, where spaces were part of the passphrase. As of Version 1.4.2-brooklyn_heights-release-4, the problem is still not fixed. The solution is to remove junk characters if any and put double quotes around the passphrase.

3.4 Issue with Apache Solr startup

Status: Fixed in devel branch, with longer time delay
During the startup, the Apache solr service is reported to have failed to start. This is generally due to the script not waiting long enough for the service to start. To confirm, wait another minute or so and do
`netstat -an|grep 898` The solr services listen on 8983 and 8984 ports. If they are listening, everything is just fine.

3.5 Operation Napalm: Solr Reinstall

If the solr install process ever should fail, it's often observed to become impossible to reinstall, even with a force install. The workaround seems to be a little extreme, but it works. I call it 'Operation Napalm'

- Remove all shard config files from /esg/config
- Remove the /usr/local/esgf-solr-3.6.0 directory
- Commence a force install. Given below is the responses to the interactive questionnaire. Say no to:

1. openssl
2. curl
3. git
4. java
5. ant
6. postgresql
7. cdat
8. esgcet
9. tomcat
10. thredds
11. node manager
12. dashboard services
13. dashboard ip
14. esgf desktop
15. openid relaying party
16. idp
17. globus gridftp
18. globus myproxy

Say yes to the following, while saying no (optionally) to backups:

1. search service
2. web front-end
3. compute role
4. ferret installation

3.6 rootAdmin is not authorized...

Status: Fixed in Brower Park release.

When executing the esg-node --test-pub command (or any other kind of publish command), the following error was observed:

```
esgcet.publish.hessianlib.RemoteCallException: Java ServiceException: User:
https://esg-test.nsc.liu.se/esgf-idp/openid/rootAdmin is not authorized to
publish/unpublish resource: http://esg-test.nsc.liu.se/thredds/esgcet/1
/pcmdi.liu.dn2.test.mytest.v1.xml
```

This problem has been fixed, thanks to Carlsten Ehbrecht of DKRZ. It involved copying /usr/local/tomcat/webapps/esg-search/WEB-INI/lib/log4j-x.x.xx.jar to /usr/local/tomcat/webapps/esg-orp. This problem also fixes, in most cases, esg-org and esgf-dashboard services which fail, producing a 404 error page. It has been communicated to the esgf-devel and esgf-user lists.

3.7 Thredds server doesn't start

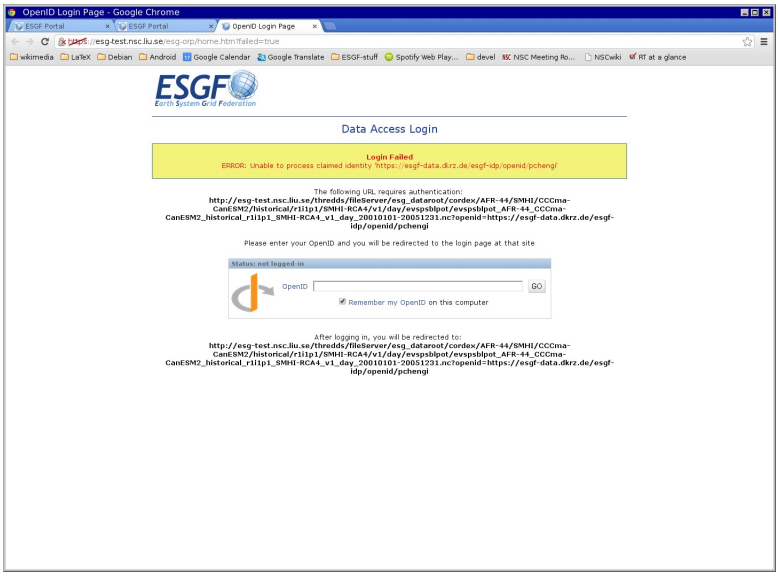
Status: Fixed in devel branch

In release v1.5 aka Brower Park, thredds server refuses to start due to a missing file which is not even logged. The problem can be fixed by `touch`-ing two files.

```
touch /esg/content/las/conf/server/las_servers_static.xml
touch /esg/content/las/conf/server/las_servers.xml
```

3.8 Unable to process claimed identity error

You have successfully logged in to a web-fe using your openid but when trying to download data, you get an error that says '`Unable to process claimed identity..`'. It looks like this:



This error means that the site you are trying to download the data from does not have the openid provider listed in their `/esg/config/esgf_idp_static.xml` file. Drop a mail to the esgf-user mailing list explaining this finding. It'll be tended to by the respective site admin.

3.9 Removal of unwanted tomcat applications

Status: Addressed in commit 01881a2d8b1ba6ce9e7836a724a4506b96647df1

A discussion on the mailing list revealed a couple of apps which are part of the standard installation of Tomcat which may represent security risks. If you are running an older install, you may have to manually remove the apps. They are:
`<tomcatdir>/webapps/manager`
`<tomcatdir>/webapps/host-manager`

Chapter

4 Recommended Optional Components

4.1 Setting up Systemtap

- Fetch the kernel-debuginfo and kernel-debuginfo-common packages for the running kernel.
- yum localinstall kernel *
- yum install systemtap-devel systemtap-testsuite glibc-debuginfo glibc-debuginfo-common
- Add users to stapdev/stapuser groups.
- Run stap scripts with the stap command.
- Here's an example of a simple stap script

```
#!/usr/bin/stap
probe begin {
        printf("Starting up!\n")

}
probe syscall.open {
        printf("%d %s %d running open args:<%s>\n",gettimeofday_us(),execname(),
        pid(),argstr)
}
```

Here's an stap script to check on dns lookups

```
#!/usr/bin/stap
probe begin {
```

```
    printf("Starting up!\n")

}
probe process("/lib64/libc.so.6").function("getaddrinfo").return {
    printf("%d returning from getaddrinfo %s(%d): %d\n", gettimeofday_us(),
    execname(), pid(), $return)
}
```

4.2 Tripwire Setup

1. Fetch the tripwire tarball (tripwire-2.4.2.2-src.tar.bz2) from http://sourceforge.net/projects/tripwire/files/latest/download

2. I have chosen to install it in `/usr/local/tripwire`. Modify the instructions as per your choice of install location.

3. Put together a script called twpoltest.sh which looks like this:

```
#!/bin/bash

TWINSTDIR=/usr/local/tripwire

for i in ‘cat $TWINSTDIR/etc/twpol.txt |tr -s ’ ’|grep ^’ ’|grep -v ’#’ \\
|grep ’/’|awk ’{print $1}’|grep -v ’!’|grep -v ’(’‘; do
    if [ ! -e $i ]; then
        echo "$i not found";
    fi
done
```

4. Install tripwire:-

   a) `uz tripwire-2.4.2.2-src.tar.bz2`

   b) `cd tripwire-2.4.22-src`

   c) `./configure --prefix=/usr/local/tripwire`

   d) `make`

   e) In the install/install.cfg file, set TWLATEPROMPTING=true and TWLOOSEDIRCHK=true

   f) `make install`

   g) Check for non-existent directories/files in policy file with twpoltest.sh (see point 2) and remove them from /usr/local/tripwire/etc/twpol.txt file.

   h) `/usr/local/tripwire/sbin/twadmin --create-polfile -S \`
   `/usr/local/tripwire/etc/site.key /usr/local/tripwire/etc/twpol.txt`
   //to generate your new policy file

i) `/usr/local/tripwire/sbin/tripwire -m i` //to initialize triwire db

j) `/usr/local/tripwire/sbin/tripwire -m c -r /tmp/report` //to generate report

k) To check the report, `/usr/local/tripwire/sbin/twprint -m r -r /tmp/report` //to print generated report

l) If you ever need to modify your policy:
Generate the policy file in text by `/usr/local/tripwire/sbin/twadmin --print-polfile /usr/local/tripwire/etc/twpol.txt`
Make changes in the twpol.txt file and update the policy by running
`/usr/local/tripwire/sbin/tripwire -m p -Z low /usr/local/tripwire/etc/twpol.txt` //to update the policy file

m) If you need to make changes to the config, you can print the cfgfile with this command:
`/usr/local/tripwire/sbin/twadmin -m f`
You can then make changes there. For instance, to add default email, enter the following line:
`GLOBALEMAIL = <user@example.org>`
You can read in the new cfg with the command:
`/usr/local/tripwire/sbin/twadmin -m F -S site.key twcfg.txt`

Chapter

5 Full Uninstall

Since the uninstall feature of the esgf installer script doesn't really give you a 'clean' machine, I used Tripwire to find out where the installer puts stuff, to come up with a full uninstaller. This list has been generated for the Brooklyn Heights release, but most of the paths are pretty much unchanged, even in Brower Park. The best way to be sure is to setup tripwire and then do a clean install yourself. You can refer to Section 4.2 for help with setting up Tripwire. I haven't done any installation with default paths ever since we upgraded to Brower Park, so this may not give you a 100% clean machine. I'll update this when I get the time to do so. **Please note that this report lists `/etc/grid-security`. If you already had a globus install on the machine prior to installing esgf middleware, you'll probably not want to lose this. Proceed with caution.**

```
/esg
/etc/bash_completion.d/esg-node
/etc/esg.env
/etc/grid-security
/etc/logrotate.d/esgf_tomcat
/root/.esgcet
/root/.globus
/usr/local/ant
/usr/local/apache-ant-*
/usr/local/apache-tomcat*
/usr/local/bin/esg-access-logging-filter
/usr/local/bin/esg-dashboard
/usr/local/bin/esg-desktop
/usr/local/bin/esgf-crawl
/usr/local/bin/esgf-optimize-index
/usr/local/bin/esgf-policy-check
/usr/local/bin/esgf-sh
/usr/local/bin/esgf-spotcheck
```

```
/usr/local/bin/esg-functions
/usr/local/bin/esgf-user-migrate
/usr/local/bin/esg-globus
/usr/local/bin/esg-idp
/usr/local/bin/esg-node
/usr/local/bin/esg-node-manager
/usr/local/bin/esg-orp
/usr/local/bin/esg-search
/usr/local/bin/esg-security
/usr/local/bin/esg-security-las-ip-filter
/usr/local/bin/esg-security-token-filters
/usr/local/bin/esg-security-tokenless-filters
/usr/local/bin/esg-web-fe
/usr/local/cdat
/usr/local/curl
/usr/local/esgf-dashboard-ip
/usr/local/esgf-solr-*
/usr/local/ferret
/usr/local/ferret_data
/usr/local/geoip
/usr/local/git
/usr/local/globus
/usr/local/gsoap
/usr/local/jdk*
/usr/local/las-esg*
/usr/local/openssl
/usr/local/pgsql
/usr/local/src/esgf
/usr/local/tomcat
/var/log/esgf-dashboard-ip.*
/var/run/tomcat-jsvc.pid
```

Chapter 6

Useful links

URL	Description
http://esg-pcmdi.llnl.gov/internal/esg-data-node-documentation/introduction-to-esg-data-node-configuration	Overview of esg.ini
http://www2-pcmdi.llnl.gov/Members/bdrach/.personal/esg-publisher-configuration/	Detailed explanation of configuration options
http://www2-pcmdi.llnl.gov/Members/bdrach/.personal/esg-publication-scripts/	Overview of data publication
http://esgf.org/wiki/ESGF_Data_Publishing	Data publish and unpublish
http://esg-pcmdi.llnl.gov/internal/esg-data-node-documentation/customizing-the-esg-publisher-with-handlers	Customizing the publisher

Table 6.2: List of useful links