GT 6 Quickstart

Introduction

This is a quickstart that shows a full installation of the Toolkit on two Fedora Linux machines, named `elephant` and `donkey`. It shows the installation of prereqs, installation of the toolkit, creation of *certificates*, and configuration of services. It is designed to supplement the main admin guide, Installing GT 6.0.

Scenarios are shown for running *GridFTP* and *GRAM5* services, and using GridFTP and GRAM clients.

Table of Contents

1. Typographical Conventions

Where there is a command to be typed, it will be preceded by one of the following prompts:

`elephant#` , `donkey#`	Run this command as the `root` super-user, on the `elephant` or `donkey` hosts respectively. You might have to use a command like **su**(8) or **sudo**(8) to start a root shell before executing the command.
`myproxy@elephant%`	Run this command as the `myproxy` user, on the `elephant` host. This user is created automatically when the myproxy-server package is installed.
`quser@elephant%` , `quser@donkey%`	Run this command as the normal user account you are intending to interact with your Globus sevices, on the `elephant` or `donkey` hosts. In this document, we use the `quser` accout for this, but if you have another user, you can use it for that purpose.

Commands themselves will be typeset as **run-this-command -with-arguments**, and responses to the commands like this `Some Response Text`. If there is some portion of a command which should be replaced by value, such as a version number, it will be typeset like this: *`REPLACEME`*.

Finally, in some cases you will be prompted for a passphrase. When that occurs, the entry of the passphrase will be indicated by **`******`**, even though nothing will be printed to the screen.

2. Prerequisites

We distribute the Globus Toolkit 6 as a set of RPM and Debian packages for Linux systems, as an installable package for Mac OS X, as a .zip file for Windows and Cygwin, as well as a source installer which can be used on other operating systems. In this quickstart, we will be installing RPM packages. Thus, it is a prerequisite for following this quickstart that you are running a distribution for which we provide RPMs. If you are running a supported Debian or Ubuntu system, the process is very similar, but you'll need to use the **apt-get** or similar tools to install the packages. For the source installer, there is more work involved, and you'll need to consult the full installation guide.

First, we will to set up our system to use the Globus package repository. This repository contains the Globus software packages, signed by our build manager. We provide RPM and Debian packages that contain a source configuration file and the public key which can be used to verify the packages. If your distribution has Globus 6.0 packages within its repository, you can skip to the next section.

The globus toolkit package repo RPM can be downloaded from the repo RPM package on globus.org[1].

To install binary RPMs, download the globus-toolkit-repo package from the link above and install it with the command:

```
elephant# rpm -hUv globus-toolkit-repo-6-1.noarch.rpm
```

The globus toolkit package repo Debian file can be downloaded from the repo Debian package on globus.org[2].

To install Debian or Ubuntu package, download the globus-toolkit-repo package from the link above and install it with the command:

```
elephant# dpkg -i globus-toolkit-repo_6-2_all.deb
```

Once you've installed the Globus repository package, you can use your operating system's packaging tools: **yum** or **apt-get**, to install the Globus components.

Important

For operating systems based on RHEL (such as Red Hat Enterprise Linux, CentOS, and Scientific Linux), the compatible EPEL repository must be enabled before installing myproxy. For OS versions 5.x, install the EPEL 5 package[3], and for OS version 6.x, use 6 package[4].

For information about installing these, see the EPEL FAQ[5].

This step is not needed for Fedora, Debian, or Ubuntu systems.

3. Setting up the first machine (GridFTP, GRAM, and MyProxy services)

3.1. Installing the Toolkit

Install packages:

```
elephant# yum install globus-gridftp globus-gram5 globus-gsi myproxy \
    myproxy-server myproxy-admin
```

This will install the GridFTP, GRAM, and *MyProxy* services, as well as set up a basic *SimpleCA* so that you can issue security credentials for users to run the Globus services.

Note

For Debian and Ubuntu systems, use **apt-get** or **aptitude** or another package manager to install the same packages as in the **yum** command above.

[1] http://www.globus.org/ftppub/gt6/installers/repo/globus-toolkit-repo-6-1.noarch.rpm
[2] http://www.globus.org/ftppub/gt6/installers/repo/globus-toolkit-repo_6-2_all.deb

3.2. Setting up security on your first machine

The Globus Toolkit uses X.509 certificates and *proxy certificates* to authenticate and authorize grid users. For this quickstart, we use the Globus SimpleCA tools to manage our own *Certificate Authority*, so that we don't need to rely on any external entitty to authorize our grid users.

Note

In many deployment scenarios, certificates for both services and users are obtained through one or more third party CAs. In such scenarios, it is unnecessary to use SimpleCA or MyProxy to issue certificates. Since this quickstart is intended to describe a simple, standalone deployment scenario, we describe how to use these tools to issue your own certificates.

When the globus-simple-ca package is installed, it will automatically create a new Certificate Authority and deploy its public certificate into the globus trusted certificate directory. It will also create a host certificate and key, so that the Globus services will be able to run.

We'll also need to copy the host certificate and key into place so that the myproxy service can use it as well.

```
elephant# install -o myproxy -m 644 \
    /etc/grid-security/hostcert.pem \
    /etc/grid-security/myproxy/hostcert.pem
elephant# install -o myproxy -m 600 \
    /etc/grid-security/hostkey.pem \
    /etc/grid-security/myproxy/hostkey.pem
```

3.3. Creating a MyProxy Server

We are going to create a MyProxy server on elephant, following the instructions at http://grid.ncsa.illinois.edu/myproxy/fromscratch.html#server. This will be used to store our user's certificates. In order to enable myproxy to use the SimpleCA, modify the `/etc/myproxy-server.config` file, by uncommenting every line in the section `Complete Sample Policy #1` such that section looks like this myproxy configuration[6]:

```
#
# Complete Sample Policy #1 - Credential Repository
#
# The following lines define a sample policy that enables all
# myproxy-server credential repository features.
# See below for more examples.
accepted_credentials      "*"
authorized_retrievers     "*"
default_retrievers        "*"
authorized_renewers       "*"
default_renewers          "none"
authorized_key_retrievers  "*"
default_key_retrievers    "none"
trusted_retrievers        "*"
default_trusted_retrievers "none"
cert_dir /etc/grid-security/certificates
```

We'll next add the myproxy user to the simpleca group so that the myproxy server can create certificates.

```
elephant# usermod -a -G simpleca myproxy
```

[6] myproxy-server.config

Start the myproxy server:

```
elephant# service myproxy-server start
Starting myproxy-server (via systemctl):                   [  OK  ]
```

Note

For Debian and Ubuntu systems, use the **invoke-rc.d** command in place of **service**.

Check that it is running:

```
elephant# service myproxy-server status
myproxy-server.service - LSB: Startup the MyProxy server daemon
   Loaded: loaded (/etc/rc.d/init.d/myproxy-server)
   Active: active (running) since Fri, 02 Nov 2012 09:07:51 -0400; 1min 20s ago
  Process: 1205 ExecStart=/etc/rc.d/init.d/myproxy-server start (code=exited, status=0/SUC
   CGroup: name=systemd:/system/myproxy-server.service
    # 1214 /usr/sbin/myproxy-server -s /var/lib/myproxy

Nov 02 09:07:51 elephant.globus.org runuser[1210]: pam_unix(runuser:session):...
Nov 02 09:07:51 elephant.globus.org myproxy-server[1212]: myproxy-server v5.9...
Nov 02 09:07:51 elephant.globus.org myproxy-server[1212]: reading configurati...
Nov 02 09:07:51 elephant.globus.org myproxy-server[1212]: usage_stats: initia...
Nov 02 09:07:51 elephant.globus.org myproxy-server[1212]: Socket bound to 0.0...
Nov 02 09:07:51 elephant.globus.org myproxy-server[1212]: Starting myproxy-se...
Nov 02 09:07:51 elephant.globus.org runuser[1210]: pam_unix(runuser:session):...
Nov 02 09:07:51 elephant.globus.org myproxy-server[1205]: Starting myproxy-se...
```

The important thing to see in the above is that the process is in the `active (running)` state.

Note

For other Linux distributions which are not using systemd, the output will be different. You should still see some information indicating the service is running.

As a final sanity check, we'll make sure the myproxy TCP port `7512` is in use via the netstat command:

```
elephant# netstat -an | grep 7512
tcp        0      0 0.0.0.0:7512            0.0.0.0:*               LISTEN
```

3.3.1. User Credentials

We'll need to specify a full name and a login name for the user we'll create credentials for. We'll be using the `QuickStart User` as the user's name and `quser` as user's account name. You can use this as well if you first create a quser unix account. Otherwise, you can use another local user account. Run the **myproxy-admin-adduser** command as the `myproxy` user to create the credentials. You'll be prompted for a passphrase, which must be at least 6 characters long, to encrypt the *private key* for the user. You must communicate this passphrase to the user who will be accessing this credential. He can use the **myproxy-change-passphrase** command to change the passphrase.

The command to create the myproxy credential for the user is

```
elephant# su - -s /bin/sh myproxy
myproxy@elephant% PATH=$PATH:/usr/sbin
myproxy@elephant% myproxy-admin-adduser -c "QuickStart User" -l quser
```

```
Legacy library getopts.pl will be removed from the Perl core distribution in the next majo
Enter PEM pass phrase: ******
Verifying - Enter PEM pass phrase:******

The new signed certificate is at: /var/lib/globus/simple_ca/newcerts/02.pem

using storage directory /var/lib/myproxy
Credential stored successfully
Certificate subject is:
/O=Grid/OU=GlobusTest/OU=simpleCA-elephant.globus.org/OU=local/CN=QuickStart User
```

3.3.2. User Authorization

Finally, we'll create a *grid map file* entry for this credential, so that the holder of that credential can use it to access globus services. We'll use the **grid-mapfile-add-entry** program for this. We need to use the exact string from the output above as the parameter to the `-dn` command-line option, and the local account name of user to authorize as the parameter to the `-ln` command-line option.

```
elephant# grid-mapfile-add-entry -dn \
    "/O=Grid/OU=GlobusTest/OU=simpleCA-elephant.globus.org/OU=local/CN=QuickStart User" \
    -ln quser
Modifying /etc/grid-security/grid-mapfile ...
/etc/grid-security/grid-mapfile does not exist... Attempting to create /etc/grid-security/
New entry:
"/O=Grid/OU=GlobusTest/OU=simpleCA-elephant.globus.org/OU=local/CN=QuickStart User" quser
(1) entry added
```

3.4. Setting up GridFTP

Now that we have our host and user credentials in place, we can start a globus service. This set up comes from the GridFTP Admin Guide.

Start the GridFTP server:

```
elephant# service globus-gridftp-server start
Started GridFTP Server                                     [  OK  ]
```

Check that the GridFTP server is running and listening on the gridftp port:

```
elephant# service globus-gridftp-server status
GridFTP Server Running (pid=20087)
elephant# netstat -an | grep 2811
tcp        0      0 0.0.0.0:2811            0.0.0.0:*               LISTEN
```

Now the GridFTP server is waiting for a request, so we'll generate a proxy from the myproxy service by using **myproxy-logon** and then copy a file from the GridFTP server with the **globus-url-copy** command. We'll use the passphrase used to create the myproxy credential for `quser`.

```
quser@elephant% myproxy-logon -s elephant
Enter MyProxy pass phrase: ******
A credential has been received for user quser in /tmp/x509up_u1001
quser@elephant% globus-url-copy gsiftp://elephant.globus.org/etc/group \
    file:///tmp/quser.test.copy
quser@elephant% diff /tmp/quser.test.copy /etc/group
```

At this point, we've configured the myproxy and GridFTP services and verified that we can create a security credential and transfer a file. If you had trouble, check the security troubleshooting section in the Security Admin Guide. Now we can move on to setting up GRAM5 resource management.

3.5. Setting up GRAM5

Now that we have security and GridFTP set up, we can set up GRAM for resource management. There are several different Local Resource Managers (LRMs) that one could configure GRAM to use, but this guide will explain the simple case of setting up a "fork" jobmanager, without auditing. For details on all other configuration options, and for reference, you can see the GRAM5 Admin Guide. The GRAM service will use the same host credential as the GridFTP service, and is configured by default to use the fork manager, so all we need to do now is start the service.

Start the GRAM gatekeeper:

```
elephant# service globus-gatekeeper start
Started globus-gatekeeper                                  [  OK  ]
```

We can now verify that the service is running and listening on the GRAM5 port:

```
elephant# service globus-gatekeeper status
globus-gatekeeper is running (pid=20199)
elephant# netstat -an | grep 2119
tcp6       0      0 :::2119                 :::*                    LISTEN
```

The gatekeeper is set up to run, and is ready to authorize job submissions and pass them on to the fork job manager. We can now run a couple of test jobs:

```
quser@elephant% myproxy-logon -s elephant
Enter MyProxy pass phrase: ******
A credential has been received for user quser in /tmp/x509up_u1001.
quser@elephant% globus-job-run elephant /bin/hostname
elephant.globus.org
quser@elephant% globus-job-run elephant /usr/bin/whoami
quser
```

If you had trouble, check the security troubleshooting section in the Security Admin Guide. To learn more about using GRAM 5, take a look at the GRAM User's Guide.

4. Setting up your second machine

Alas, it's not much of a grid with just one machine. So let's start up on another machine and add it to this little test grid.

4.1. Setting up your second machine: Prereqs

See Prereqs.

4.2. Setting up your second machine: Installation

Install packages as before:

```
donkey# yum install globus-gridftp myproxy globus-gram5
```

4.3. Setting up your second machine: Security

Now let's get security set up on the second machine. We're going to trust the original simpleCA to this new machine; there's no need to create a new one. First, we'll bootstrap trust of the SimpleCA running on elephant:

```
donkey# myproxy-get-trustroots -b -s elephant
Bootstrapping MyProxy server root of trust.
New trusted MyProxy server: /O=Grid/OU=GlobusTest/OU=simpleCA-elephant.globus.org/CN=host/
New trusted CA (e3d1c34d.0): /O=Grid/OU=GlobusTest/OU=simpleCA-elephant.globus.org/CN=Glob
Trust roots have been installed in /etc/grid-security/certificates/.
```

This allows clients and services on `donkey` to trust certificates which are signed by the CA on `elephant` machine. If we weren't going to run any Globus services on `donkey`, then we could stop here. Users on `donkey` could acquire credentials using the **myproxy-logon** command and perform file transfers and execute jobs using the **globus-url-copy** and **globus-job-run** commands. However, we'll continue to configure the GridFTP and GRAM5 services on `donkey` as well.

We're going to create the host certificate for donkey, but we create it on elephant, so that we don't have to copy the certificate request between machines. The **myproxy-admin-addservice** command will prompt for a passphrase for this credential. We will use this passphrase to retrieve the credential on donkey.

```
myproxy@elephant% myproxy-admin-addservice -c "donkey.globus.org" -l donkey
Legacy library getopts.pl will be removed from the Perl core distribution in the next majo
Enter PEM pass phrase:******
Verifying - Enter PEM pass phrase:******

The new signed certificate is at: /var/lib/globus/simple_ca/newcerts/03.pem

using storage directory /var/lib/myproxy
Credential stored successfully
Certificate subject is:
/O=Grid/OU=GlobusTest/OU=simpleCA-elephant.globus.org/OU=local/CN=donkey.globus.org
```

Next we'll retrieve the credential on donkey as the root user.

```
donkey# myproxy-retrieve -s elephant -k donkey.globus.org -l donkey
Enter MyProxy pass phrase: ******
Credentials for quser have been stored in
/etc/grid-security/hostcert.pem and
/etc/grid-security/hostkey.pem.
```

At this point, we no longer need to have `donkey`'s host certificate on `elephant`'s myproxy server, so we'll delete it.

```
donkey# myproxy-destroy -s elephant -k donkey.globus.org -l donkey
MyProxy credential 'donkey.globus.org' for user donkey was successfully removed.
```

And as a final setup, we'll add quser's credential to the grid-mapfile on `donkey`, so that the `quser` account can access services there as well.

```
donkey# grid-mapfile-add-entry -dn \
    "/O=Grid/OU=GlobusTest/OU=simpleCA-elephant.globus.org/OU=local/CN=QuickStart User" \
    -ln quser
Modifying /etc/grid-security/grid-mapfile ...
/etc/grid-security/grid-mapfile does not exist... Attempting to create /etc/grid-security/
```

```
New entry:
"/O=Grid/OU=GlobusTest/OU=simpleCA-elephant.globus.org/OU=local/CN=QuickStart User" quser
(1) entry added
```

At this point, we have set up security on donkey to trust the CA on elephant. We have created a host certificate for donkey so that we can run Globus services on donkey, and we have enabled the quser account to use services on donkey. The last thing to do is to turn on the Globus services on donkey.

4.4. Setting up your second machine: GridFTP

GridFTP set up on the second machine is identical to the first. I'll just list the commands here; see Section 3.4, "Setting up GridFTP" for additional information.

```
donkey# service globus-gridftp-server start
Started GridFTP Server                                     [  OK  ]
```

Now we can test it.

First, we'll retrive a proxy credential from the myproxy server so that the user on donkey can interact with the Globus services. Here we'll use the same passphrase as we used to create the `quser` credential.

```
quser@donkey% myproxy-logon -s elephant
Enter MyProxy pass phrase: ******
A credential has been received for user quser in /tmp/x509up_u1001.
```

Next we'll transfer a file between the gridftp servers on donkey and elephant:

```
quser@donkey% globus-url-copy gsiftp://elephant.globus.org/etc/group \
   gsiftp://donkey.globus.org/tmp/from-elephant
```

That was a slightly more complicated test than we ran on elephant earlier. In this case, we did a third-party transfer between two GridFTP servers. It worked, so I have the local and remote security configured correctly.

If you run into problems, perhaps you have a firewall between the two machines? GridFTP needs to communicate on data ports, not just port 2811. The error for this condition looks like:

```
error: globus_ftp_client: the server responded with an error
500 500-Command failed. : callback failed.
500-globus_xio: Unable to connect to 140.221.8.19:42777
500-globus_xio: System error in connect: No route to host
500-globus_xio: A system call failed: No route to host
500 End.
```

You can set up a range of ports to be open on the firewall and configure GridFTP to use them. See the GridFTP admin firewall doc.

4.5. Setting up your second machine: GRAM5

Now we can submit a staging job. This job will copy the **/bin/echo** program from donkey to a file called `/tmp/my_echo`. Then it runs it with some arguments, and captures the stderr/stdout. Finally, it will clean up the `my_echo` file when execution is done.

```
quser@donkey% globus-job-run elephant \
    -x '(file_stage_in=(gsiftp://donkey.globus.org/bin/echo /tmp/echo)) \
        (file_clean_up=/tmp/echo)' /bin/ls -l /tmp/echo
```

```
-rw-r--r-- 1 quser quser 27120 Nov  2 09:56 /tmp/echo
```

This example staged in a file, had an executable act on that file, and cleaned up the file afterward.

You can get other examples of GRAM files from GRAM usage scenarios.

5. Conclusion

Hopefully this guide has been helpful in familiarizing you with some of the administration tasks and tools to use the Globus Toolkit. If you've reached this point successfully, you should have enough knowledge to enable additional hosts to use your grid by repeating the tasks in Section 4, “Setting up your second machine”. Also, by repeating the tasks in Section 3.3.1, “User Credentials” and Section 3.3.2, “User Authorization” you can enable additional users to access your compute and data resources.

Glossary

C

Certificate Authority (CA)	An entity that issues certificates.
certificate	A public key plus information about the certificate owner bound together by the digital signature of a CA. In the case of a CA certificate, the certificate is self signed, i.e. it was signed using its own private key.

G

Grid Resource Allocation and Management (GRAM)	This component is used to locate, submit, monitor, and cancel jobs on Grid computing resources.
GridFTP	A file transfer protocol based on FTP with extensions for security and parallel data transfers.
grid map file	A file containing entries mapping certificate subjects to local user names. This file can also serve as a access control list for GSI enabled services and is typically found in `/etc/grid-security/grid-mapfile`. For more information see the Gridmap section here.

M

MyProxy	Myproxy manages X.509 credentials (certificates and private keys). MyProxy combines an online credential repository with an online certificate authority to allow users to securely obtain credentials.

P

private key	The private part of a key pair. Depending on the type of certificate the key corresponds to it may typically be found in `$HOME/.globus/userkey.pem` (for user certificates), `/etc/grid-security/hostkey.pem` (for host certificates) or `/etc/grid-security/<service>/<service>key.pem` (for service certificates).

For more information on possible private key locations see this.

proxy certificate

A short lived certificate issued using a EEC. A proxy certificate typically has the same effective subject as the EEC that issued it and can thus be used in its place. GSI uses proxy certificates for single sign on and delegation of rights to other entities.

For more information about types of proxy certificates and their compatibility in different versions of GT, see http://dev.globus.org/wiki/Security/ProxyCertTypes.

S

SimpleCA

SimpleCA is a tool for creating and managing a CA. It provides a way to implement a X.509 trust root and sign certificates for users and hosts.