Table of Contents

Note on site-launched startd's

For a trusted resource, the site should create a pilot certificate from a CA recognized at CERN and send the DN to the Submission Infrastructure group (cms-htcondor-admins@cernNOSPAMPLEASE.ch) who will enter the certificate DN in the condor_mapfile of the central manager, etc. In special circumstances, the Submission Infrastructure group can provide a CMS pilot certificate.

Creating Pilot and Service Certificates

Every grid job, be it a glideinWMS pilot (glidein) or a user job, needs a grid proxy in order to authenticate at sites. A proxy is created on the frontend from a grid certificate and a key. This page describes how to obtain new pilot certificates from the CERN Certificate Authority.

Creating a New Certificate

On the CERN Certificate Authority website, go to "New Grid Host Certificate" and choose "Request certificate using OpenSSL (for Linux machines)" here. Create a certificate request with a subject, for example:

```
cmspilot01/vocms080.cern.ch
```

You need to be an owner of the machine at CERN in the certificate request name. You should use a certificate-friendly browser like Firefox if you want to not make your life difficult.

You will next be asked to generate a certificate-key pair. You can do this on lxplus:

```
openssl req -new -subj "/CN=cmspilot01\/vocms080.cern.ch" -out newcsr.csr -nodes -sha512 -newkey
```

or follow the updated instructions on the CERN CA webpage if they have changed. Once the certificate is generated, download the base64 certificate to a file, which we will call **host.cert**.

Next we will export the certificate from this file to a **p12** file:

```
dir=cmspilot01
openssl pkcs12 -export -inkey privkey.pem -in host.cert -out ${dir}.p12
```

Create cert file and key file from the p12 file and set permissions correctly:

```
openssl pkcs12 -clcerts -nokeys -in ${dir}.p12 -out ${dir}cert.pem
openssl pkcs12 -nocerts -in ${dir}.p12 -out ${dir}key-enc.pem
openssl rsa -in ${dir}key-enc.pem -out ${dir}key.pem
chmod 400 ${dir}key*
chmod 600 ${dir}cert.pem
```

Next verify the cert and key have the same hash:

```
openssl x509 -noout -modulus -in ${dir}cert.pem | openssl md5
openssl rsa -noout -modulus -in ${dir}key.pem |openssl md5 | uniq
```

This should always be the case unless you did something really incorrect like mix the files from two different certificates.

Lastly, get some information about the certificate like its Distinguished Name (DN) subject, and its validation period:

```
openssl x509 -in ${dir}cert.pem -noout -subject -startdate -enddate
```

Register the Certificate with CMS

For service certificates (i.e. for the frontend), you can skip this step.

"Lasciate ogne speranza, voi ch'intrate" - Dante Alighieri

In order to use CMS resources, every pilot certificate must be registered with the CMS VOMS group as a member of CMS, as well as get special roles like the **pilot** role. This distinguishes pilot jobs from other types of jobs like user jobs. Follow the procedure to "Add an additional certificate" on the CMS VOMS administration website (direct link may work, if it does not, use the previous link and scroll down about a page to find the button on the right side) VOMS administrators are Stefano, Andreas Pfeiffer and Tony, if you need speedy approval. Note also that as of Summer 2015 the Global Pool certificate needs also the **production** role since fair-share at the Tier-1 sites depends on having it.

Collector Authorization

As user condor, authorize the new certificate on the collectors:

```
glidecondor_addDN -daemon "cms pilot cert DN" "/DC=ch/DC=cern/OU=computers/CN=cmspilot12/vocms016
condor_reconfig
```

Each glidein needs to communicate with the collector, and cannot do so without authorization in the **condor_mapfile**. Never update the **condor_mapfile** by hand! Use the script **glidecondor_addDN**.

Certificate Repository

There isn't one.

Creating a Proxy

You should store the certificate and key files on the glideinWMS frontend. Next you will have to create proxies. The proxies are short-lived versions of the certificate information that can be used more safely on the grid. In the Global Pool, these are found in **_gfrontend@vocms0167.cern.ch:/home/gfrontend/.globus**, and the certificate and key files in the **certs** directory below that. The fundamental procedure boils down to:

Create the base proxy:

```
certdir=/home/gfrontend/.globus/certs
idstr=01
voms-proxy-init -cert ${certdir}/cmspilot${idstr}cert.pem -key ${certdir}/cmspilot${idstr}key.pem
```

Add the appropriate VOMS roles to the base proxy, for example.

```
export X509_USER_PROXY=${here}/x509_pilot${idstr}_cms.proxy.tmp
voms-proxy-init -cert ${certdir}/cmspilot${idstr}cert.pem -key ${certdir}/cmspilot${idstr}key.pem
```

See the script **renew_proxies.sh** for the full details. This script is run as a cron job. The resulting proxy files can be used in the glideinWMS frontend configuration xml file.

List of Certificates in Production

Next expiration date: April 28, 2020

```
subject= /DC=ch/DC=cern/OU=computers/CN=cmspilot01/vocms080.cern.ch
notAfter=Apr 28 22:53:58 2021 GMT
subject= /DC=ch/DC=cern/OU=computers/CN=cmspilot02/vocms080.cern.ch
notAfter=Apr 28 22:59:02 2021 GMT
subject= /DC=ch/DC=cern/OU=computers/CN=cmspilot03/vocms080.cern.ch
notAfter=Apr 28 23:02:03 2021 GMT
subject= /DC=ch/DC=cern/OU=computers/CN=cmspilot04/vocms080.cern.ch
notAfter=Apr 28 23:06:18 2021 GMT
subject= /DC=ch/DC=cern/OU=computers/CN=cmspilot05/vocms080.cern.ch
notAfter=Apr 28 23:07:59 2021 GMT
subject= /DC=ch/DC=cern/OU=computers/CN=frontend01/vocms080.cern.ch
notAfter=Apr 28 23:10:58 2021 GMT
subject= /DC=ch/DC=cern/OU=computers/CN=frontend02/vocms080.cern.ch
notAfter=Apr 28 23:12:42 2021 GMT
subject= /DC=ch/DC=cern/OU=computers/CN=frontend03/vocms080.cern.ch
notAfter=Apr 28 23:14:31 2021 GMT
subject= /DC=ch/DC=cern/OU=computers/CN=frontend04/vocms080.cern.ch
notAfter=Apr 28 23:17:35 2021 GMT
subject= /DC=ch/DC=cern/OU=computers/CN=frontend05/vocms080.cern.ch
notAfter=Apr 28 23:19:14 2021 GMT
```

List of Active certificates

some of the nodes have been configured to use puppet managed certificate renewal. It use certmgr puppet module provided by CERN IT. Here is the manifest defined in glideinwmsfrontend module Once the certificate is in the proper place it is expected to be renewed automatically. Following is the list of all active pilot and frontend certificates in Submission Infrastructure

Node	Service/use	Partial DN	Certificate management/Next expiration
vocms080	Global pool FE service	/CN=frontend02/vocms080.cern.ch	puppet managed auto renewal,(expiry:May-11,2022)
vocms080	Global pool pilot	/CN=cmspilot02/vocms080.cern.ch	puppet managed auto renewal,(expiry:May-11,2022)
vocms0819	CERN pool FE service	/CN=frontend05/vocms0819.cern.ch	puppet managed auto renewal,(expiry:May-11,2022)
vocms0819	CERN pool pilot	/CN=cmspilot05/vocms0819.cern.ch	puppet managed auto renewal,(expiry:May-11,2022)
UCSD	UCSD Pilot	/CN=cmspilot01/vocms080.cern.ch	Manual renewal: expiry 28-April,2021
vocms0801	ITB-dev pool FE service	/CN=frontend03/vocms0801.cern.ch	puppet managed auto renewal
vocms0801	ITB-dev pool pilot	/CN=cmspilot03/vocms0801.cern.ch	puppet managed auto renewal
vocms0802	ITB pool FE service	/CN=frontend04/vocms0802.cern.ch	puppet managed auto renewal
vocms0802	ITB pool pilot	/CN=cmspilot04/vocms0802.cern.ch	puppet managed auto renewal
vocms0840	Volunteer pool FE service	/CN=frontend04/vocms0840.cern.ch	puppet managed auto renewed
vocms0840	Volunteer pool pilot	/CN=cmspilot06/vocms0840.cern.ch	puppet managed auto renewal

Script to simplify certificate creation (July 2016, last rev. March 2020) for all certificates at once.

```
#!/bin/sh
# 24-Mar-2020 Some urls changed since last year
TOP=`pwd`
HOST=`basename $TOP`

if [ $HOST == "vocms080" ] ; then
  DIRS="cmspilot01 cmspilot02 cmspilot03 cmspilot04 cmspilot05 frontend01 frontend02 frontend03 f
elif [ $HOST == "vocms052" ] ; then
  DIRS="tw"
else
  exit 1
fi

for dir in $DIRS ; do
  if [ ! -d $TOP/$dir ] ; then
    mkdir $TOP/$dir
  fi
  cd $TOP/$dir
  if [ ! -f host.cert ] ; then
    cp /etc/pki/tls/openssl.cnf .
cat >>openssl.cnf <<END
[req]
req_extensions = v3_req

[ v3_req ]
basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment
subjectAltName = @alt_names

[alt_names]
END
    echo "DNS.1 = ${HOST}.cern.ch" >> openssl.cnf

    openssl req -new -subj "/CN=${dir}\/${HOST}.cern.ch" \
      -out newcsr.csr -nodes -sha512 -newkey rsa:2048 \
      -config ${TOP}/${dir}/openssl.cnf
    echo
    echo "Go to the following url and request the certificate with the"
    echo "information in newcsr.csr for certificate $dir/${HOST}.cern.ch:"
    #echo "https://ca.cern.ch/ca/host/HostSelection.aspx?template=EE2Host&instructions=openssl"
    echo "https://ca.cern.ch/ca/host/Submit.aspx?template=ee2host&instructions=openssl&subject=${
    echo "Download the base-64 certificate and then copy host.cert from"
    echo "your desktop to this area:"
    pwd
  else
    echo
    echo "*** Enter when prompted:"
    echo
    openssl pkcs12 -export -inkey privkey.pem -in host.cert -out ${dir}.p12
    openssl pkcs12 -clcerts -nokeys -in ${dir}.p12 -out ${dir}cert.pem
    openssl pkcs12 -nocerts -in ${dir}.p12 -out ${dir}key-enc.pem
    openssl rsa -in ${dir}key-enc.pem -out ${dir}key.pem
    chmod 400 ${dir}key*
    chmod 600 ${dir}cert.pem
    openssl x509 -noout -modulus -in ${dir}cert.pem | openssl md5
    openssl rsa  -noout -modulus -in ${dir}key.pem | openssl md5 | uniq
  fi
done

exit 0
```

JamesLetts - 2020-03-24

This topic: CMSPublic > CompOpsGlideinWMSCerts
Topic revision: r37 - 2021-04-06 - SaqibHaleem