

Table of Contents

| | |
|---|----------|
| CRAB3-Panda Operations..... | 1 |
| CRAB Client..... | 1 |
| CRAB REST APIs..... | 1 |
| Task Worker..... | 1 |
| Service host..... | 1 |
| Cert used by service..... | 1 |
| Configuration files..... | 1 |
| Component operations..... | 1 |
| Logs and tips..... | 2 |
| APF..... | 2 |
| Service host..... | 2 |
| Cert used by service..... | 3 |
| Configuration files..... | 3 |
| Component operations..... | 3 |
| Logs and tips..... | 3 |
| Kill all the jobs of a given queue..... | 3 |
| Monitor job status..... | 3 |
| CMSAdder Component..... | 4 |
| Service host..... | 4 |
| Cert used by service..... | 4 |
| Config file description..... | 4 |
| Component operations..... | 4 |
| Logs and tips..... | 4 |

CRAB3-Panda Operations

This twiki is intended to track all the info and tips to allow debugging the operations of the system. Main components we'd like to document are REST APIs, TaskWorker, CMSAdder Component, ASO, APF

CRAB Client

Ops documentation is here

CRAB REST APIs

Ops documentation is here

Task Worker

Ops documentation here

The full AsyncStageOut wiki management page can be found here

<https://svnweb.cern.ch/trac/CMSDMWM/wiki/AsyncStageOutManagement>

Service host

- ASO service is setup in devaso1. It runs as vosusr01. So to operate the service you need to log into the machine through vosusr01 account (login first on lxplus then ssh on devaso1)

Cert used by service

- To contact FTS/DBS/CS REST interface, a proxy is required. The service proxy is renewed via a crontab. The log of the proxy renewal can be found here `/tmp/vomsrenew.log` while the script to renew it is here `/home/vosusr01/proxy/vomsrenew.sh`. The script generates a valid proxy and write it here `/home/vosusr01/gridcert/proxy.cert`.

Configuration files

- The ASO relies on 1 configuration file for all component. The config file is located here `/data/ASO/async_install_common_pre16/current/config/asyncstageout/config.py`. Each component config section modification requires the restart of the component. There is also a `config_database` in couch located here http://devaso1.cern.ch:5184/_utils/database.html?asynctransfer_config. Each component has an associated document that you can modify. Such modification does not required the restart of the components.

Component operations

All followings operations command must be done under `/data/ASO/async_install_common_pre16/current`

- To start/stop the tool: `./config/asyncstageout/manage stop-asyncstageout`
- To start/stop a single ASO component: `./config/asyncstageout/manage execute-asyncstageout wmcord --shutdown/start --component AsyncTransfer/DBSPublisher/Analytics`
- To start/stop the database: `./config/asyncstageout/manage stop-services`

Logs and tips

- The log of all ASO components can be found here:
/data/ASO/async_install_common_pre16/current/install/asyncstageout/
- The locations of FTS log files are printed in the AsyncTransfer component log file:
/data/ASO/async_install_common_pre16/current/install/asyncstageout/AsyncTransfer/stderr.log.
- The active users are printed in the AsyncTransfer log file:

```
[vosusr01@devasol current]$ tail -f install/asyncstageout/AsyncTransfer/stderr.log
DEBUG:PhEDEX:getData:
  url: tfc
  data: {'node': u'T2_US_Nebraska'}
DEBUG:PhEDEX:Data is from the cache
DEBUG:PhEDEX:getData:
  url: tfc
  data: {'node': u'T3_IT_Perugia'}
DEBUG:PhEDEX:Data is from the cache
DEBUG:root:kicking off pool
DEBUG:root:current_running [[u'santocch', u'', u'']]
```

- To check if system is correctly transferring and not stucking, you need to check the FTS log files:

```
[vosusr01@devasol current]$ tail -1000 install/asyncstageout/AsyncTransfer/stderr.log|grep ftslog
DEBUG:AsyncTransfer-Worker-santocch:log file created: /data/ASO/async_install_common_pre16/v01/in
DEBUG:AsyncTransfer-Worker-santocch:executing command: ftscp -copyjobfile=/tmp/tmpzjxzyw -server=
DEBUG:AsyncTransfer-Worker-santocch:log file created: /data/ASO/async_install_common_pre16/v01/in
DEBUG:AsyncTransfer-Worker-santocch:executing command: ftscp -copyjobfile=/tmp/tmpoKVPWF -server=
```

A correct on-going transfer should print something like the following in the log:

```
[vosusr01@devasol current]$ tail -f /data/ASO/async_install_common_pre16/v01/install/asyncstageou
+ glite-transfer-status -s https://fts.cr.cnaf.infn.it:8443/glite-data-transfer-fts/services/File
213cf55e-f381-11e2-9991-b28fbfac286f status is Active
+ glite-transfer-status -s https://fts.cr.cnaf.infn.it:8443/glite-data-transfer-fts/services/File
213cf55e-f381-11e2-9991-b28fbfac286f status is Active
+ glite-transfer-status -s https://fts.cr.cnaf.infn.it:8443/glite-data-transfer-fts/services/File
213cf55e-f381-11e2-9991-b28fbfac286f status is Active
+ glite-transfer-status -s https://fts.cr.cnaf.infn.it:8443/glite-data-transfer-fts/services/File
213cf55e-f381-11e2-9991-b28fbfac286f status is Active
+ glite-transfer-status -s https://fts.cr.cnaf.infn.it:8443/glite-data-transfer-fts/services/File
213cf55e-f381-11e2-9991-b28fbfac286f status is Active
```

- Make sure to source the UI environment before any restart of the components.
- The ASO monitoring can help on spotting problems:
http://devasol.cern.ch:5184/asynctransfer/_design/monitor/index.html

APF

Auto Pilot Factory. Check status of the Panda Queues and submit pilot wrapper accordingly to the status of the queue and to the APF queues configuration.

Service host

- vocms60: serves the setup of poc3test
- vocms73: serves the setup of c3p1

Both machines run the APF under root account. Log into the machine through vosusr01 account (login first on lxplus then ssh on the node you want)

Cert used by service

The APF uses a pool of certificates for the pilot (wrapper) submission. Configuration is on `/etc/apf/proxy.conf`. Certs are in `/data/vosusr01/proxy`

Configuration files

There are two main config files: `/etc/apf/queues.conf` and `etc/apf/factory.conf`. Sites specific information are in `queues.conf`. There are few common parameters and then there is a section for each queue.

There is also the wrapper script which is not really a configuration file but it may requires changes, fixes and/or adjustments. It is in

1. `/data/apf/runpilot3-wrapper-cms-new_1.sh`

If you open it, you can see what it does. Mainly you must know that it is here where we define which is the actual pilot framework. (Currently we are downloading it from <http://cmsdoc.cern.ch/~spiga/pilot.tar.gz>)

Component operations

There are two services which run on the APF. The APF itself and condor. You may need to start and stop both. Log in as root and

1. `service condor start/stop`
2. `service factory start/stop`

condor restart should not be needed. Also the APF doesn't requires any restart so often

Logs and tips

There are two kind of logs. The APF log and the condor job logs. Both are in `/data/apf/log/`

1. `/data/apf/log/apf`
2. `/data/apf/log/`

job logs are kept for three days and this is a configurable parameter.

Kill all the jobs of a given queue

```
for i in `condor_q -wide | grep "Rome" | awk '{print $1}'`; do echo $i; condor_rm $i ; done
```

Monitor job status

List all the wrapper submitted (irrespective of the status / queue)

1. `condor_q -wide`

List all the wrapper submitted to a site (irrespective of the status)

1. `condor_q -wide | grep Pisa`

List all the wrapper in status Idle submitted to a specific queue

1. `condor_q -wide | grep " I " | grep Pisa`

CMSAdder Component

- The CMS Adder code is located in `/data/atlpan/srv/lib/python2.6/site-packages/pandaserver/dataservice/AdderCmsPlugin.py`. To edit it, you need a sudo privilege since this plugin is run as atlpan user.
- The CMS Adder component retrieves the jobs/outputs metadata from panda database and:
 1. upload files metadata into crabserver DB
 2. insert document for each files into ASO database
 3. send back a status code to panda server: 0 (interactions succeeded), 1 (interactions succeeded) and 2 (interactions failed due to a fatal error)
- The crabserver interface and ASO database are specified in the config file.

Service host

- The CMS Panda Component is running in the the same host of panda server, voatl294. To login into this machine, you need to login into one machine of lxvoadm.cern.ch cluster with your lxplus username and password. And then from there login into voatl294 using always your lxplus username and password. For the access to lxvoadm.cern.ch cluster you need to contact `jorge.amando.molina-perez@cern.NOSPAMPLEASE.ch` and for voatl294 you need that someone from Atlas open a ticket in "ATLAS Central Services Operations" project asking the sudo access for you on that machine.

Cert used by service

- To contact the CS REST interface, a proxy is required. The service proxy is renewed via a crontab. The log of the proxy renewal can be found here `/tmp/vomsrenew.log` while the script to renew it is here `/data/atlpan/proxy/vomsrenew.sh`. The script generates a valid proxy and write it here `/data/atlpan/proxy/proxy.cert`

Config file description

- The Adder component has only 1 config file located here `/data/atlpan/srv/etc/panda/auth_aso_plugin.txt`. It is a json file where the `ASO_DB_URL`, `PROXY` and `ASO_CACHE` are specified. To edit it you need sudo privilege.

Component operations

- The component is called by panda server. So there is not a command to start or stop the component. The component is started and stopped with panda server.

Logs and tips

- The log of this component can be found here: `/data/atlpan/srv/var/log/panda/ASOPlugin.log`
- To ensure that the component is uploading correctly files into CS database, you need to see in the log lines like these ones:

```
< CMS-Server-Time: D=461573 t=1371549244778066^M
< ^M
{ [data not shown]
```

CRAB3PandaOps < Sandbox < TWiki

```
^M100 576 100 16 100 560 29 1026 --:--:-- --:--:-- --:--:-- 1029
* Connection #0 to host c3p1.cern.ch left intact
* Closing connection #0
* SSLv3, TLS alert, Client hello (1):
} [data not shown]
, 0
```

while for the ASO interactions, you need to see something like the following in the log:

```
2013-06-18 14:27:11,947 DEBUG Trying to commit {'dn': '/DC=ch/DC=cern/OU=Organic Units/OU=Users/C
2013-06-18 14:27:11,978 INFO ASOPlugin ends.
```

- In general, the failure of the Adder to contact the REST (you do not see the message above) can be caused by:
 - ◆ Expired proxy: in this case ensure that the proxy /data/atlpan/proxy/proxy.cert is valid and check /tmp/vomsrenew.log to make sure that the proxy renewal is done correctly
 - ◆ Troubles with CS REST/database: you can try to contact the CS REST interface manually using curl. If the CS REST interface does not reply correctly to your request you need to go to CS REST machine and see what is happening
- In general, the failure of the Adder to contact the the ASO (you do not see the message above) is caused by troubles of the ASO couchdb service: try to open the ASO database index page http://devaso1.cern.ch:5184/_utils/index.html to make sure the ASO database in couch is up and running correctly. If the ASO database does not respond you need to go to the ASO instance machine and see what is happening.

This topic: Sandbox > CRAB3PandaOps

Topic revision: r4 - 2014-10-07 - AndresTanasijczuk



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback