

Quickstart for HTCondor

Honey Gupta

This doc contains instructions on how to run a sample python code on HTCondor. Refer to the links mentioned in the end for detailed documentation and troubleshooting.

For queries or more details, contact: Honey Gupta (hn.gpt1@gmail.com)

Quick-check before starting

Login to lxplus terminal and type ``klist``

Sample output:

```
[hgupta@lxplus724 ~]$ klist
Ticket cache: FILE:/tmp/krb5cc_134713_fpa2kB
Default principal: hgupta@CERN.CH

Valid starting    Expires          Service principal
2020-06-18T17:08:04 2020-06-19T18:08:04  krbtgt/CERN.CH@CERN.CH
    renew until 2020-06-23T17:08:04
2020-06-18T17:08:05 2020-06-19T18:08:04  afs/cern.ch@CERN.CH
    renew until 2020-06-23T17:08:04
```

Valid kerberos credentials should be present before invoking the commands. If the credentials are not valid, authentication won't pass and operations will fail.

This can be solved by running ``kinit``.

[\[Need more details\]](#)

A *Hello world!* case for running a job on HTCondor

To start, you need **three files** to run the batch job on HTCondor:

- The code to run - most probably a **python script**
- A bash/executable file - a **bash script** that contains any pre-installation instructions
- A **submit file** - That contains the details of the job you want to submit

1. Create a hello.py file with the following script:

```
print("Hello world!")
```

2. Create a hello.sh file with:

```
#!/bin/bash
```

```
python hello.py
```

3. Create a hello.sub file with:

```
executable          = hello.sh
arguments           = $(ClusterId)$(ProcId)
output              = hello.$(ClusterId).$(ProcId).out
error               = hello.$(ClusterId).$(ProcId).err
log                 = hello.$(ClusterId).log
transfer_input_files = hello.py
Queue               =
```

4. Make the .sh file executable

```
$ chmod +x hello.sh
```

5. Submit the job

```
$ condor_submit hello.sub
```

Sample output:

```
[hgupta@lxplus747 ex_hello]$ condor_submit hello.sub
Submitting job(s).
1 job(s) submitted to cluster 1013175.
[hgupta@lxplus747 ex_hello]$ _
```

6. Check the status of the job

```
$ condor_q
```

Sample output:

```
[hgupta@lxplus747 ex_hello]$ condor_q

-- Schedd: bigbird10.cern.ch : <188.185.70.244:9618?... @ 06/18/20 17:36:37
OWNER  BATCH_NAME      SUBMITTED   DONE    RUN    IDLE  TOTAL JOB_IDS
hgupta ID: 1013175    6/18 17:35      _      _      1      1 1013175.0

Total for query: 1 jobs; 0 completed, 0 removed, 1 idle, 0 running, 0 held, 0 suspended
Total for hgupta: 1 jobs; 0 completed, 0 removed, 1 idle, 0 running, 0 held, 0 suspended
Total for all users: 20667 jobs; 402 completed, 2 removed, 17829 idle, 2373 running, 61 held, 0 suspended
```

Pointers:

- Idle = in the queue

- Run = running
- Done = done (obvious :p)
- Hold = (according to my observations but not sure about it) the code was running but some error occurred and unless the error is solved, the code cannot run further. Details of the error can be found in the specified error file. The job will be eventually killed after timeout.

7. Check the log file

```
$ cat hello.*.log
```

Sample output:

```
[hgupta@lxplus747 ex_hello]$ cat hello.*.log
000 (1013176.000.000) 06/18 17:39:12 Job submitted from host: <188.185.70.244:9618?addrs=188.185.70.244-9618+[2001-1458-d00-7--100-2ee]-9618&alias=bigbird10.cern.ch&noUDP&sock=schedd_1831871_b74a_252>
...
040 (1013176.000.000) 06/18 17:39:25 Started transferring input files
    Transferring to host: <128.142.70.253:9618?addrs=128.142.70.253-9618+[2001-1458-301-52--100-622]-9618&noUDP&sock=2637_c144_6654>
...
040 (1013176.000.000) 06/18 17:39:25 Finished transferring input files
...
001 (1013176.000.000) 06/18 17:39:26 Job executing on host: <128.142.70.253:9618?addrs=128.142.70.253-9618+[2001-1458-301-52--100-622]-9618&noUDP&sock=12651_1cb9_3>
...
006 (1013176.000.000) 06/18 17:39:26 Image size of job updated: 1
    0 - MemoryUsage of job (MB)
    0 - ResidentSetSize of job (KB)
...
040 (1013176.000.000) 06/18 17:39:26 Started transferring output files
...
040 (1013176.000.000) 06/18 17:39:27 Finished transferring output files
...
005 (1013176.000.000) 06/18 17:39:27 Job terminated.
    (1) Normal termination (return value 2)
        Usr 0 00:00:00, Sys 0 00:00:00 - Run Remote Usage
        Usr 0 00:00:00, Sys 0 00:00:00 - Run Local Usage
        Usr 0 00:00:00, Sys 0 00:00:00 - Total Remote Usage
        Usr 0 00:00:00, Sys 0 00:00:00 - Total Local Usage
    72 - Run Bytes Sent By Job
    28 - Run Bytes Received By Job
    72 - Total Bytes Sent By Job
    28 - Total Bytes Received By Job
Partitionable Resources :   Usage   Request Allocated
Cpus                    :         0         1         1
Disk (KB)               :        22         1       270178
Memory (MB)             :         0       2000       2000
...
[hgupta@lxplus747 ex_hello]$ _
```

8. Check the output

```
$ cat hello.*.out
```

Sample output:

```
[hgupta@lxplus747 ex_hello]$ cat hello.*.out
Hello world!
```

9. Empty .out file? Check the error file:

```
$ cat hello.*.err
```

Need more details?

- Batch Service Concepts: <https://batchdocs.web.cern.ch/concepts/index.html>
- Quickstart doc: <https://batchdocs.web.cern.ch/local/quick.html>
- Tutorials to learn more: <https://batchdocs.web.cern.ch/tutorial/introduction.html>
- Use GPUs: <https://batchdocs.web.cern.ch/tutorial/exercise10.html>