Ganymede Script execution

This page details the scripts and how they get executed. In all cases where a script says it can be executed manually, all other scripts/processes must **NOT** be running with the exception of the API (agent_api.py) which should always be running.

agent_api.py

About

This Python script provides the Ganymede API. It is designed to be running all the time in a web server context.

Usage

This script is run as a daemon using init scripts. It is not executed manually.

jupiter.py

About

This Python script coordinates actions based on the transaction log in the Ganymede schema in MySQL.

Usage

This script should be executed from the 'ganymede' user's crontab once an hour. It can be run manually.

```
$ python /usr/local/bin/jupiter.py
```

ganymede_agent.sh

About

This Bash script gets installed on a server in each GEO.

Usage

It should get called from cron of the 'stats' user once an hour. A unique agent ID is required and can be hardcoded in the script or passed via command line. It can be run manually.

```
$ /usr/local/bin/ganymede_agent.sh
```

ganymede_worker.sh

About

This Bash script runs on the Ganymede server. It gets called by *jupiter.py* for each successfully completed Agent transaction. It can be run manually.

Usage

A transaction ID is required to be passed via command line.

```
/usr/local/bin/ganymede_worker.sh 22b5afaf-b950-4322-9215-74a76c06acba
```

ganymede_worker_etl.py

About

This Python script gets called by the *jupiter.py* script to complete a single transaction once an Agent has completed (extract from the local MySQL, transform and load into MongoDB.) It can be called manually.

Usage

A transaction ID is required to be passed via command line.

```
$ python /usr/local/bin/ganymede_worker_etl.py --transactionid=22b5afaf-b950-4322-9215-74a76c06acba
```

ganymede_glassbeam_extract.sh

About

This Bash script gets called by the *jupiter.py* script once all Worker transactions are complete or a configured time period has elapsed. It can be called manually.

Usage

It takes, as optional arguments, a time period for selecting data from MongoDB. NOTE: The time period values **must** be specified using UTC. Use military time for specifying the hour. Do **NOT** use leading zeros (use '3' and not '03'.)

```
# Use current UTC time
$ /usr/local/bin/ganymede_glassbeam_extract.sh

# Specify exact time period of September 13th, 2015, 9 am UTC
$ /usr/local/bin/ganymede_glassbeam_extract.sh -y 2015 -m 9 -d 13 -h 9
```

hourly_backup.sh

About

This Bash script runs once an hour after *jupiter.py* has completed processing. It dumps the Ganymede schema, then queries the Ganymede database to get a list of all archives Agents uploaded during the hour. It then creates a new bucket in Cloud Files (using the S3 API) for that hour and, finally, uploads all files from that hour to Cloud Files (the Ganymede dump and the archives.) Once all files have been transferred, it moves the files involved in the backup out of /var/lib/ganymede/upload to /var/lib/ganymede/archives

Usage

This script takes no arguments. It can be run manually.

Important: Credentials for Cloud Files are **NOT** stored in any of the Ganymede configuration files nor in the script itself. They are stored in the .s3curl file in the ganymede home directory. If you need to change the Cloud Files account used by the script, you must change the credentials in this file. Also know that you **DO NOT** use the username and password used when creating the Cloud Files account. You **MUST** use the Interop credentials generated by Mezeo.

```
$ /usr/local/bin/hourly_backup.sh
```