JobSub - Bug #23603

jobsub_server can't decode base64 strings with "-" in them as produced by the client

11/14/2019 04:34 PM - Shreyas Bhat

Status:	Resolved	**Start date:**	12/03/2019
Priority:	Normal	**Due date:**	
Assignee:	Shreyas Bhat	**% Done:**	100%
Category:	JobSub Server RPM	**Estimated time:**	0.00 hour
Target version:	v1.3.2	**Spent time:**	0.00 hour
First Occurred:		**Stakeholders:**	
Occurs In:	v1.3.03		

Description

In INC000001090158, Pengfei describes submitting a job using the following:

```
export RELEASE_NAME=v9_33_00&&export OUT_DIR=/pnfs/GM2/scratch/users/lwelty/hadd-tests/dingpf_endG
ame_20191114_103550&&export LIST_FILE_URL=http://home.fnal.gov/~dingpf/20191114_103550.list&&expor
t NJOBS=1&&export HADD_METHOD=hadd&&export GRIDUSER=dingpf&&jobsub_submit -N 1 --resource-provides
=usage_model=DEDICATED,OPPORTUNISTIC -e RELEASE_NAME -e OUT_DIR -e LIST_FILE_URL -e NJOBS -e HADD_
METHOD -e GRIDUSER -G gm2 file:///gm2/app/users/dingpf/hadd_grid_script.sh
```

The jobsub server returned an error saying that $RELEASE_NAME wasn't set. This isn't actually the issue.

What's happening is that jobsub_client is passing to the server an argument "--export-env", which is a base64-encoded string (using python's base64.urlsafe_b64encode method). This inserts "-" in certain spots. The options Pengfei passed in happened to yield an export string of:

```
ZXhwb3J0IEdSSURVU0VSPWRpbmdwZjtleHBvcnQgSEFERF9NRVRIT0Q9aGFkZDtleHBvcnQgTkpPQlM9MTtleHBvcnQgTElTVF
9GSUxFX1VSTD1odHRwOi8vaG9tZS5mbmFsLmdvdi9-ZGluZ3BmLzIwMTkxMTE0XzEwMzU1MC5saXN0O2V4cG9ydCBPVVRfRElS
PS9wbmZzL0dNMi9zY3JhdGNoL3VzZXJzL2x3ZWx0eS9oYWRkLXRlc3RzL2RpbmdwZl9lbmRHYW1lXzIwMTkxMTE0XzEwMzU1MD
tleHBvcnQgUkVMRUFTRV9OQU1FPXY5XzMzXzAwOw==
```

On the jobsub server side, in jobsub_env_runner.py we do something like "echo <b64_string> | base64 ~~d" to decode the string. However, base64 on SL7 doesn't know how to handle the "~~" in the middle of the string above. Thus, the submission fails.

We need to change this decoding line so that it recognizes urlsafe base64 (https://en.wikipedia.org/wiki/Base64#Variants_summary_table).

One possible workaround that seems to work on Pengfei's encoded string is:

```
echo <b64_string> | python -c "import base64, sys; [sys.stdout.write(base64.urlsafe_b64decode(line
.strip())) for line in sys.stdin]"
```

We'll need to think about this and see if there's a better way to go about this or if this will work. Hopefully, this will go out in the release due in mid-December.

Subtasks:

Bug # 23692: Review request [commit:5ab2faad865abaefa7e948b83797492ecfc93675: Changed b...	**Closed**

History

#1 - 11/19/2019 03:06 PM - Shreyas Bhat

- Assignee set to Shreyas Bhat

#2 - 12/03/2019 02:52 PM - Shreyas Bhat

- % Done changed from 0 to 50

When we do the python3 porting, we'll need to use the line

```
echo $b64 | python3 -c "import base64, sys; [sys.stdout.write(str(base64.urlsafe_b64decode(line.strip()))) for
 line in sys.stdin]"
```

Because python3's str.write() method expects a string.

#3 - 12/03/2019 04:24 PM - Shreyas Bhat

Done and tested on fermicloud074.

I managed to test that the original line in this ticket worked, but *not* that the original case didn't.

Opening a ticket to Dennis to review.

#4 - 12/03/2019 04:24 PM - Shreyas Bhat

- % Done changed from 50 to 90

- Status changed from New to Feedback

#5 - 12/03/2019 04:28 PM - Shreyas Bhat

- Start date changed from 11/14/2019 to 12/03/2019

- Due date set to 12/03/2019

due to changes in a related task: #23692

#6 - 12/06/2019 03:16 PM - Shreyas Bhat

Note: for python3, we'd have to use something like this (since sys.stdout.write expects a string and base64.urlsafe_b64decode gives a byte string:

```
echo <base64string> | python3 -c "import base64, sys; [sys.stdout.write(str(base64.urlsafe_b64decode(line.stri
p()))) for line in sys.stdin]"
```

#7 - 04/07/2020 12:06 PM - Dennis Box

Looks OK to merge

#8 - 04/07/2020 12:21 PM - Shreyas Bhat

- Status changed from Feedback to Resolved

Merged.