

ceph-volume - Bug #43283

unit test failure: UnicodeDecodeError: 'utf-8' codec can't decode byte 0xd0 in position 0: invalid continuation byte

12/12/2019 12:42 PM - Jan Fajerski

Status: New	% Done: 0%
Priority: Normal	
Assignee: Alfredo Deza	
Category:	
Target version:	
Source:	Affected Versions:
Tags:	ceph-qa-suite:
Backport:	Pull request ID:
Regression: No	Crash signature (v1):
Severity: 3 - minor	Crash signature (v2):
Reviewed:	

Description

```
_____ TestFunctionalCall.test_unicode_encoding _____  
  
self = <ceph_volume.tests.test_process.TestFunctionalCall object at 0x7f14a38912b0>  
  
  def test_unicode_encoding(self):  
>     process.call(['echo', u'\xd0'])  
  
ceph_volume/tests/test_process.py:83:  
-----  
  
command = ['/bin/echo', 'ï¿½'], kw = {}, executable = '/bin/echo'  
terminal_verbose = False, logfile_verbose = True, verbose_on_failure = True  
show_command = False, command_msg = 'Running command: /bin/echo ï¿½', stdin = None  
  
  def call(command, **kw):  
      """  
      Similar to ``subprocess.Popen`` with the following changes:  
  
      * returns stdout, stderr, and exit code (vs. just the exit code)  
      * logs the full contents of stderr and stdout (separately) to the file log  
  
      By default, no terminal output is given, not even the command that is going  
      to run.  
  
      Useful when system calls are needed to act on output, and that same output  
      shouldn't get displayed on the terminal.  
  
      Optionally, the command can be displayed on the terminal and the log file,  
      and log file output can be turned off. This is useful to prevent sensitive  
      output going to stderr/stdout and being captured on a log file.  
  
      :param terminal_verbose: Log command output to terminal, defaults to False, and  
                             it is forcefully set to True if a return code is non-zero  
      :param logfile_verbose: Log stderr/stdout output to log file. Defaults to True  
      :param verbose_on_failure: On a non-zero exit status, it will forcefully set logging ON fo  
r  
                             the terminal. Defaults to True  
  
      """  
      executable = which(command.pop(0))  
      command.insert(0, executable)
```

```

terminal_verbose = kw.pop('terminal_verbose', False)
logfile_verbose = kw.pop('logfile_verbose', True)
verbose_on_failure = kw.pop('verbose_on_failure', True)
show_command = kw.pop('show_command', False)
command_msg = "Running command: %s" % ' '.join(command)
stdin = kw.pop('stdin', None)
logger.info(command_msg)
if show_command:
    terminal.write(command_msg)

process = subprocess.Popen(
    command,
    stdout=subprocess.PIPE,
    stderr=subprocess.PIPE,
    stdin=subprocess.PIPE,
    close_fds=True,
    **kw
)

if stdin:
    stdout_stream, stderr_stream = process.communicate(as_bytes(stdin))
else:
    stdout_stream = process.stdout.read()
    stderr_stream = process.stderr.read()
returncode = process.wait()
if not isinstance(stdout_stream, str):
>     stdout_stream = stdout_stream.decode('utf-8')
E     UnicodeDecodeError: 'utf-8' codec can't decode byte 0xd0 in position 0: invalid continuation byte

ceph_volume/process.py:211: UnicodeDecodeError
----- Captured log call -----
INFO     ceph_volume.process:process.py:191 Running command: /bin/echo i¿½

```

Seen here <https://jenkins.ceph.com/job/ceph-volume-pr/606/console>

@Alfredo mind having a look at this?