Wednesday, July 23, 2014

Embedding python in bash scripts

As a software development consultant, I do a lot of bash scripting. I can do a lot of really creative things using nothing but bash and the binutils at my disposal, but sometimes I'll come across something that's just easier to do in a higher level scripting language. Enter python.

Conversely, there are a lot of things that are just easier and more straight forward to do in bash, so writing everything in straight python may be more work than it's worth.

Here's a quick posting to describe how you can embed some python code into your bash scripts and get the best of both worlds. *Note: just as a heads, up, the examples in this posting are quite contrived.*

Calling python from bash is easy. You simply use python's '-' argument and pipe in your python code. I typically wrap my python code in a bash function.

```
#!/bin/bash

function current_datetime {
python - <<END
import datetime
print datetime.datetime.now()
END
}

# Call it
current_datetime

# Call it and capture the output
DT=$(current_datetime)
echo Current date and time: $DT
```

You can also pass data into your embedded python script. I do that using environment variables:

```
#!/bin/bash

function line {
PYTHON_ARG="$1" python - <<END
import os
line_len = int(os.environ['PYTHON_ARG'])
print '-' * line_len
END
}

# Do it one way
line 80

echo 'Handy'

# Do it another way
echo $(line 80)
```

My usual use-case for doing this is if I'm extending someone else's bash scripts and have to 'go off the reservation' a bit. Sometimes I'm updating an existing 'legacy' script and need to look up some data... maybe do a REST call or something. Here's an example bash script that uses curl to call a REST service to get some weather data. Then is passes the raw JSON response to an embedded python script to interpret and format the results:

```
#!/bin/bash

function format_weather_data() {
PYTHON_ARG="$1" python - <<END
import os
import json

json_data = os.environ['PYTHON_ARG']
data =json.loads(json_data)
lookup = {
    '200': 'thunderstorm with light rain',
    '201': 'thunderstorm with rain',
    '202': 'thunderstorm with heavy rain',
    '210': 'light thunderstorm',
    '211': 'thunderstorm',
    '212': 'heavy thunderstorm',
    '221': 'ragged thunderstorm',
    '230': 'thunderstorm with light drizzle',
    '231': 'thunderstorm with drizzle',
    '232': 'thunderstorm with heavy drizzle',
    '300': 'light intensity drizzle',
    '301': 'drizzle',
    '302': 'heavy intensity drizzle',
    '310': 'light intensity drizzle rain',
    '311': 'drizzle rain',
    '312': 'heavy intensity drizzle rain',
    '313': 'shower rain and drizzle',
    '314': 'heavy shower rain and drizzle',
    '321': 'shower drizzle',
    '500': 'light rain',
    '501': 'moderate rain',
    '502': 'heavy intensity rain',
    '503': 'very heavy rain',
    '504': 'extreme rain',
    '511': 'freezing rain',
    '520': 'light intensity shower rain',
    '521': 'shower rain',
    '522': 'heavy intensity shower rain',
    '531': 'ragged shower rain',
    '600': 'light snow',
    '601': 'snow',
    '602': 'heavy snow',
    '611': 'sleet',
    '612': 'shower sleet',
    '615': 'light rain and snow',
    '616': 'rain and snow',
    '620': 'light shower snow',
    '621': 'shower snow',
    '622': 'heavy shower snow',
    '701': 'mist',
    '711': 'smoke',
    '721': 'haze',
    '731': 'sand, dust whirls',
    '741': 'fog',
    '751': 'sand',
    '761': 'dust',
    '762': 'volcanic ash',
    '771': 'squalls',
    '781': 'tornado',
    '800': 'clear sky',
    '801': 'few clouds',
    '802': 'scattered clouds',
    '803': 'broken clouds',
    '804': 'overcast clouds',
    '900': 'tornado',
    '901': 'tropical storm',
    '902': 'hurricane',
    '903': 'cold',
    '904': 'hot',
    '905': 'windy',
    '906': 'hail',
    '950': 'setting',
    '951': 'calm',
    '952': 'light breeze',
    '953': 'gentle breeze',
    '954': 'moderate breeze',
    '955': 'fresh breeze',
    '956': 'strong breeze',
    '957': 'high wind, near gale',
    '958': 'gale',
    '959': 'severe gale',
    '960': 'storm',
    '961': 'violent storm',
    '962': 'hurricane',
}

print "Current temperature: %g F" % data['main']['temp']
print "Today's high: %g F" % data['main']['temp_max']
print "Today's low: %g F" % data['main']['temp_min']
print "Wind speed: %g mi/hr" % data['wind']['speed']
weather_descs = [lookup.get(str(i['id']), '*error*') for i in data['weather']]
print "Weather: %s" % ', '.join(weather_descs)

END
}

WEATHER_URL="http://api.openweathermap.org/data/2.5/weather?
q=Cincinnati,OH&units=imperial"

format_weather_data "$(curl -s $WEATHER_URL)"
```

Hope you find this information useful.

Posted by bhfsteve at 1:17 PM

Labels: bash python

11 comments:

Cj Welborn July 23, 2014 at 10:32 PM

This is awesome. I've needed this so many times and didn't know it was possible. No more little scripts.

Reply

Anonymous December 2, 2014 at 8:30 AM

This is superb. I would possibly keep the python part in a separate script and source it in - I like to be organised! :)

Reply

Pierre Loicq October 7, 2015 at 10:39 AM

Thanks you, it works ! Make sure not to set indentation to the python block, otherwise you could get some errors

Reply

weidenrinde January 7, 2016 at 4:18 AM

Another, possibly easier way of passing variables to and from the python part is as command line arguments:

```
function to_upper() {
output=$(python - "$1" << END
import sys
print sys.argv[1].upper()
END
)
}

to_upper asdfasdf
echo $output
# gives ASDFASDF
```

Reply

Dumidu Handakumbura April 24, 2017 at 12:49 PM

Thanks man. Just what I was looking for.

Reply

go2pacha July 20, 2017 at 2:07 PM

thanks for the writeup, finally got my bash to run embedded py w arguments.

Reply

Anonymous January 15, 2018 at 3:52 PM

The cool thing is that you also could do it not either with python but with c/c++ or many other high language, to make thing much easier than creating sometimes nested bash-scripts

Reply

Anonymous February 5, 2018 at 4:54 PM

Hi and thanks for the info.

Have a case where this doesn't seem to work as expected. I'm sure it's all me.... I'm attempting to use pssh (https://github.com/lilydjwg/pssh) which is a parallel ssh tool (sends same command to many hosts at same time).

if you run the pssh command from cli with arguments, you'll get expected output, for example:

pssh -h /tmp/hostlist -o /tmp/output "uname -ar"

it'll populate /tmp/output with the output of uname -ar from all hosts listed in hostlist. perfect.

I am trying to call this command with the same arguments from within bash but it isn't working for me.Everytime I get the command usage help, as you would if you passed nothing to it.

Would you mind providing an example of how you would call a python script with arguments?

thank you -

Reply

Replies

Anonymous February 5, 2018 at 5:08 PM

As is often the case, I ask then figure things out shortly afterwards...

This is what I did to resolve this, in case it helps others. Hopefully I am not spreading bad coding practices...

I used exec..

in the bash script:

exec /usr/bin/python /usr/bin/pssh -h /tmp/pshhl2 -o /tmp/output "uname -ar"

Reply

infotechbrn1@gmail.com August 14, 2018 at 7:12 AM

Thank you for your post. This is excellent information. It is amazing and wonderful to visit your blog.
iOS App Development Course
iPhone App Training Course

Reply

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)