Housekeeping Script and Configuration Management Documentation

Jamie Clarke

February 18, 2019

1 Overview

The script was written in Python 3.7 targeting Ubuntu 16.04. Linux was chosen as the target platform as the application sounded as if it will be running on a server in an enterprise environment.

2 Housekeeping Script

The housekeeping script has three different modes of operation that can be used to reduce the space log files are taking up once the specified threshold has been reached. The application has the following parameters seen as variables at the top of the script:

- Mode - Determines how log files will be treated
- Disk - Specify which disk on the server you want to scan for usage. May be useful if application is on a second drive or non conventional directory
- Threshold - Specify at what percentage of hard disk usage should action be taken against log file. ('95' would mean action was taken against log files once disk was 95% full)
- Logging Directory - Specify where the logs being generating are stored

2.1 Log Rotation

The first method provided in the housekeeping script is log rotation. When used in this mode the script will read each log file in from a specified directory and date order them. Only the newest log file is kept and the rest of the files are deleted. This method could be modified based on requirements to keep log files for N number of days (1 log file per day from email) while deleting the rest, however the version I provided just keeps the latest log file.

2.2 Compression

The second method included is compression. If you run the script in ZIP mode it will read the logging directory and compress all the files into a .zip file. This method reclaims alot of space. The testing I did showed a 168kb log file reduced down to 571 bytes. The disadvantage to this method is that its then difficult to access the compressed log files again programmatically and will require uncompressing the files first.

2.3 Wiping the directory

The easiest solution to the problem is to run the script in 'WIPE' mode. This method will simply iterate through files in the target folder and delete all of them. This method is less effective than the other two as all log files are lost but may be useful when disk space is near a critical level. Both log rotation and/or compression would be preferable.

2.4 Execution

This script should run periodically to check if disk space is violating the threshold. When tested this was done by using a cron job on Ubuntu 16.04. Adding the following line to the unix crontab file will schedule the housekeeping script to run every 30 minutes indefinitely.

*/30 * * * * python setup.py

Alternative: I could have ran the Python script indefinitely on a thread using Pythons time.sleep to wake the script every 30 minutes however this did not feel as lightweight as the cronjob solution.

3 Configuration Management

For this section I opted to use two batch scripts that has been tested on Ubuntu 16.04. The script can be ran anywhere as long as the user has sudo privileges. The script creates a 'bedegaminguser' account and a 'bedegaminggroup' group. The user is assigned to the group and then the logging directory created within the users home directory. This looks something like '/home/bedegaming/loggingdirectory'. Once this is done read and write permissions are granted to the 'bedegaminggroup. Finally the script will enter this newly created directory and using github as the hosting service download a copy of the housekeeping script using git clone. Alternatives for hosting this file might have looked like hosting it within an Amazon S3 bucket or just using local infrastructure the company might have.

I looked into more viable solutions for an enterprise environment to this problem and came across applications such as Puppet, Ansible or SCCM. If I had the licenses or a set-up these products may have been a safe bet to provide

a more scalable solution if the application in the email was rolled out to multiple servers.

The second script aims to maintain write privileges for the logging directory. This can be added as a cron job that runs every minute and should work for 99% of cases.

*/1 * * * * maintainWritePrivs.sh

4 Testing

Within the git repository I included a 'Simulate Logging' file which will write out 5 log files each with a different time stamp to the logging directory, the cleanup script can then be ran to see what the effects are.

As the script only runs when disk space is near full I've included a log that will show what percentage full your hard drive is. You can then set the threshold variable to be below that number to test the script.

5 Other

Git Repository = https://github.com/Jamie-Clarke/HouseKeeping

5.1 External Libraries

pip install psutil

6 Contact

Thank you for taking the time to review the code and this document, if there are any problems or you'd like more information my email is below

Email = jamie.clarke@northumbria.ac.uk