**Bash**

**(1) Move log file**

1. Copy the file to a linux server
2. Run “./copy\_file.sh X”
   1. X should be the number of days
   2. If no value was added , the default will be 7

**(2) Process**

There are 2 options,

1. Define a cron job that will run the script every X time
2. Register the script as a service (system). This option will require to define will true loop

**(3) Copy files**

There are few options to preform the task, I describe a solution based on Ansible and python

1. Using ansible
   1. Define the server Ips as inventory file (Username and password for each server if needed)
   2. Create the test.sh on ansible server using ansible file module; This in order to avoid file creation on every server
   3. Copy the file to all Linux server using ansible copy module

\*\*It is preferred that username and password will be the same in all servers, If not   
1. In inventory file username and password can be defined per server

2.You can create a yaml file that includes the username and password per server Ip and create the inventory in memory and part of the task you can loop the yaml file and find the username and password per each server based on IP match

1. Using python, I would run the following steps
   1. As a prerequisite I would create a file that includes the servers ip (Username and password for each server if needed)
   2. Create the test.sh on my local computer; This in order to avoid file creation on every server
   3. Read the server Ips from the file and define them inside array variable
   4. Connecting the each server via scp and copy the file ; preforming loop based on the server IP

**Python**

1. Copy the url.py and config.json to the same directory
2. Run **python3 url.py**
   1. The execution will a log file “output.log”
   2. The execution will update the config.json

**Devops tool**

(1)Terrafom is infrastructure as code tool that can build or change an environment based on virtualization or public cloud. The tool saves a state per environment this can help find changes that crated manually and revert the environment to the required configuration.

The test.tf creates and environment on Amazon cloud with the following options

1. The state file on the environment is sabe on amazon S3 bucket.
2. The objects are based on module configuration
3. 4 instances based on M5.4xlarge
4. 10 Amazon RDS instances based on mysql
5. Amazon queue service