

Week 4 Assignment

Description

For this assignment you should create a script that receives three command-line arguments. The first argument is an IP address the second is a number (selection). The last one is the username that the script should ssh as. Your script *MUST* expect passwordless ssh to work. If you have to manually do anything during script execution, you need to fix it. Remember that to do passwordless ssh, you can use the `ssh-copy-id` command to copy your keys to a remote machine. For example, if you are on the ssh server and want to login to scratch without a password, you can do `ssh-copy-id d0000xxyy@scratch`

The script does the following based on the number that they pass in:

- 1 - Test if host is awake -> this will send a single ping to the remote host. If the host is awake, your script should print out the string "ip is awake." (replace ip with the actual ip address), otherwise it will print "ip is not awake.". This should be done in a function named `one`.
- 2 - Get the apache process status on remote host . (Remember you can manually see the service status by doing `service ssh status` (replace ssh with whatever service that you are checking for)). You will have to do some grepping and awking to just get the actual status. I.e. I can grep this line `Active: active (running) since Fri 2022-01-07 09:56:13 MST; 1 months 4 day>` to get the second word. So, the script will just print out that second word. This should be done in a function named `two`.
- 3 - Get disk percentage used on root filesystem -> will return just the percentage field for the root `/` filesystem on the remote machine. Hint: You can get the whole line for the `/` by doing `df | egrep '/$'`. You would then need to awk it to just give the percent field. This should be done in a function named `three`.
- 4 - Get hostname of remote machine -> returns the hostname command results. This should be done in a function named `four`.
- 5 - display a count of how many users are logged in -> Just a single integer will print to the screen. This should be done in a function named `five`.

You should have a `main` function that checks to see if any of the required input parameters are blank. If so, it should exit with a status of `1`(one). Otherwise, it determines what the `selection` variable is set to and calls the appropriate function.

Notes

- You don't want anything printing out except as required from above. (I.e. you could redirect ping response to `/dev/null`).
- You can do a command over ssh like: `ssh user@host "hostname"`
- Your ssh command MUST have this option `ssh -o 'StrictHostKeyChecking no'`
- When doing an awk within an ssh command, it tends to lose sight of the awk variable, so you have to escape like this: `ssh user@host "some command | awk '{ print \$2 }'`. See the backslash in the awk statement.
- When calling main, you should do similar to your first assignment: `[[ "$0" == "${BASH_SOURCE[0]}" ]] && main "$@" || true`
- Your script *must* be marked executable on github. The easiest way to make sure that this is the case is to clone your repo, set the execute permission on the shell script, and commit and push the changes back. There is NOT a way to do this via the web interface.
- A good way to capture the output of the ssh command is as follows:

```
status=$(ssh -o 'StrictHostKeyChecking no' $user@$ip ${cmd})
echo $status
```

Submission

- Upload to your `week4/assignment.sh` in your github repo.
- You may also be required to demo this in class.