

IT 3110: Systems Automation

Ansible 2

Assignment

This assignment will hone your system automation skills with ansible. You should also focus on using variables in your playbook.

Tasks

Description

You should utilize your AWS environment and terraform to do this assignment. You should already have a terraform file that will create instances for you. The only modification that you will have to make, is that you will also have to attach an additional 1GB EBS volume to your EC2 instance(s). Any new partitions should be created on this new volume.

An example terraform config that will show you how to attach a volume can be found [here](#)

Recreate the [following](#) with ansible. A sample playbook to create a single partition on a GPT filesystem is given [here](#)

You should not use the `command` module. You should research and utilize native ansible modules where possible.

The hardest part is finding the start and end of your partitions. Well, once you find the starting location, it isn't hard to add 50 each time. An easy way to view the correct start location is to `parted /dev/sda` on the machine you want to partition (or whatever your device it, it won't be /dev/sda). At the prompt, change the units to MB by doing `unit MB`. Then show the free space by doing `print free`. You need to start your first partition at the beginning of the free space, then just add 50 to it for the end space. The start of the next partition will be the same number as the ending of the previous.

- Your assignment, should create the appropriate partitions, format them correctly, and mount them. You should use loops and variables where possible.
- Your assignment should also run the appropriate inodehog programs in each of the 10 partitions.
- After running the hog program, use a raw shell module to get the output of `ls -l | wc -l` to see how many files were created. Store the result in a variable and have ansible print that variable out in a debug message.

Submission

Pass off in class.