**Linux Administration**

**Environment:**

* Ubuntu 20 EC2 t2.micro with public access (Name=Linux)
* 1GB ebs storage attached to the EC2
* **Attach security group that allow 0.0.0.0/0 all protocols - Inbound rule**
* **Use pem that you already have or create and download a new one**

**TIP:** When in doubt, use “man”

1. Write a command to print lines 10-20 of /var/log/dmesg

**sed -n '10,20p' /var/log/dmesg**

1. Write a command to print lines that contain the word **systemd** or the word **zone** in /var/log/dmesg

**awk '/systemd|zone/ {print}' /var/log/dmesg**

1. Write a command that prints only the part of the date from /var/log/syslog

**awk '{print $3}' /var/log/syslog**

1. How can you replace or substitute strings in a given file?   
   Replace each appearance of ‘unix’ with ‘linux’ in file test.txt with a bash command (That yolv u choose) or using a command inside preferred editor (vi, vim, nano)

**‘test.txt’ content:**

**sed -i 's/unix/linux/g' test.txt**

-------------------------------------------------------------------------------------------------------------

unix is great os. unix is opensource. unix is free os.

learn operating system.

unix linux which one you choose.

unix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.

-------------------------------------------------------------------------------------------------------------

1. How do you handle installation, removal and updates of software in Ubuntu linux systems?

**sudo apt-get install** **[package name]**

**sudo apt-get remove [package name]**

**sudo apt-get update [package name]**

1. Install ‘postgresql 12’ on the server and make sure it is running and enabled after reboot.

* Attach a screenshot of postgres service current status and an output of a command that prints postgres version

1. What is the difference between two commands:
   1. Apt-get update
   2. Apt-get upgrade

**If we take the python pachage for the example update wil uptdate the current version of the python uf it was 3.8 it will stay 3.8 with all the updates of this version if we take the upgrade command the pachage will aupgrade the python version to higher one (3.9 ) it can cause some difficulties for the developpers**

1. What should you do if you want to update all your installed packages to their latest version?

**Apt-get upgrade**

1. How the server knows where to look for a package, how can you add a new package source to the system?

**There is a source list file wich can be edited if you want ta add specials repository or externals source codes**

**You can locate it at /etc/apt/sources. List**

1. How would you check the file system disk space usage on a Linux system?

Attach the output of the command

1. How do you list all running processes?

**sudo ps -a**

1. How do you list all installed services (available services) on the system?

**systemctl --full --type service --all**

1. You have 1GB unpartitioned empty disk attached to your Linux Server.

Complete the following:

1. Complete all required steps to create 1GB filesystem of ‘ext4’ format using the attached empty disk.
   1. Create Partition of 1 GB
   2. Create physical volume from the partition
   3. Create ‘wave-lv0’ Logical Volume 1Gb of size as part of ‘wave-vg’ volume group. The ‘wave-vg’ volume group should from the Physical Volume you created earlier.
   4. Make sure Logical Volume filesystem is ‘ext4’
2. Ensure created filesystem permanently mounted on “/<your\_name>-mount” and is persistent after reboot.

Please attach:

* Screenshot that proves that the filesystem is mounted
* Screenshot of /etc/fstab

**Configuration Management with Ansible**

**Environment:**

* Ubuntu 20 EC2 t2.micro with public access (Name=Controler)
* Ubuntu 20 EC2 t2.micro private only (Name=node01)
* Ubuntu 20 EC2 t2.micro private only (Name=node02)

**Tip:** When in doubt, use “ansible-doc”

1. Configure ssh-key based authorization with user “wave”wave
   1. Create user wave on **control hosts and on node01-node02**
   2. Generate SSH-key pair for user wave on “control host”
   3. Copy the generated ssh public key to remote ‘authorized\_keys’ of remote hosts
   4. Make sure “wave” is part of sudoers and no password is required to run privileged commands

**Expected Results**: Ansible should be able to connect to all managed hosts using “wave” user.

1. Installing and configuring Ansible
   1. Make sure ansible installed on your ‘Control’ host
   2. Create directory /home/wave/ansible. All Ansible manifests should be placed in this directory, eventually you will push the answers to the Git repository.
   3. Configure inventory file in /home/wave/ansible/inventory with the following:
      1. node01 in ‘dev’ group
      2. node02 in ‘prod’ group
   4. Create ansible configuration file in /home/wave/ansible/ansible.cfg, the configuration should point to ‘inventory’
   5. Make sure you are able to access your remote hosts
   6. Write down an ad hoc command that is using privilege escalation to modify the contents of the “/etc/motd” file on node01 and node02 so that it contains the string “Managed by Ansible\n”. Use “wave” as the remote user.

(Hint, use the relevant module).

**ansible all -m copy --args="content='Managed by Ansible\n' dest=/etc/motd"**

* 1. Write down an ad hoc command that you used to verify that the contents of the “/etc/motd” file on node01 and node02 are identical.

**ansible all -m command -m ="src=/etc/motd dest=/etc/motd"**

3. Implementing Playbooks -

*Configure cronjob using ansible*

Create a playbook ‘backup-cron.yml’ in /home/wave/ansible folder. The playbook will execute a ‘home-backup.sh’ script that will backup all ‘/home’ folders on the remote host.

Instructions:

1. On the controller host under ‘/home/wave/ansible/scripts’ create ‘home-backup.sh’ bash script, attach your script to answers:
   1. The script will archive the ‘/home’ directory and its content.
   2. The archive (zip file) is saved as a file ‘/backups/home-backup.zip’. (***No need to check if the folder exists or create it in this step or in the script***)
2. Create a playbook ‘/home/wave/ansible/backup-cron.yml’ that applies to ‘dev’ and ‘prod’ host groups. Add an entry to enable privilege escalation.
3. Add first task that ensures that ‘/backups’ directory exists on remote hosts
4. Create a second task that will copy the backup script that you created in task ‘a’ to ‘/backups/home-backup.sh’ on remote hosts. Set the owner and group to be ‘root’.
5. Create a third task that will schedule the execution of the backup script on *‘prod’ hosts* every day at 1 AM using crontab, the script should be runned by user root.(Use the appropriate module)
6. Create a fourth task that will schedule the execution of the backups script on *‘dev’ hosts* on days sunday-thursday at 10 PM, the script should be runned by user root.(Use the appropriate module)
7. Write down an ad-hoc command that shows the crontab configuration on remote hosts.

**ansible all -m command -a 'cat /etc/crontab'**

When you finish, archive the /etc/ubuntu/ansible folder to ‘zip’ file and attach it to your answers email.