ICT & Infra S3 Automation week 2

Class:	*CB01*
Student number:	4961854
Student name:	*Heiko Morales Aloria*

Introduction

This week you will learn how to install Ansible, create/modify inventory file, execute simple Ad hoc commands and Playbooks. Before executing the assignments, start with installation of Ansible for your chosen OS. Beware, Windows is not supported for the control node.

Installation Guide: *https://docs.ansible.com/ansible/latest/installation_guide/index.html*

Additionally, you will need to create/modify inventory list to successfully execute Ansible commands in one or more clients. Therefore, after installation of Ansible, investigate the article "How to build your inventory" (*https://docs.ansible.com/ansible/latest/user_guide/intro_inventory.html*). Later, create a new group with several hosts (for example: another VM, the same control node, another computer/device in your house that supports SSH connectivity).

Assignment 1. Create an Ad hoc command to execute in a selected group of hosts

Difficulty: ★★☆☆☆.

Let's execute few commands in Ansible using the simplest possible way. Think about few commands you can execute in host machines. Provide successful results after running these commands. **Tip**: use "command" or "shell" module to run shell commands in remote hosts.

More information about Ad hoc commands: *https://www.middlewareinventory.com/blog/ansible-ad-hoc-commands*

More information about Modules: *https://docs.ansible.com/ansible/2.9/modules/modules_by_category.html*

Provide screenshots (evidence) for your solution. Always explain your evidence!

Solution:

Ansible has been installed on the VMController server and the "/etc/ansible/hosts" file has been configured to control via ssh the 10.0.2.6 server known as VMDataBase.

```
heiko@controller:~$ ansible all -m command -a "uname -a"
mysqlserver | CHANGED | rc=0 >>
Linux database 5.4.0-125-generic #141-Ubuntu SMP Wed Aug 10 13:42:03 UTC 2022 x86_64 x86_64 x86_64 G
NU/Linux
heiko@controller:~$ ansible all -a "df -h"
mysqlserver | CHANGED | rc=0 >>
Filesystem                         Size  Used Avail Use% Mounted on
udev                               445M     0  445M   0% /dev
tmpfs                               98M  1.1M   97M   2% /run
/dev/mapper/ubuntu--vg-ubuntu--lv   24G  4.8G   18G  21% /
tmpfs                              489M     0  489M   0% /dev/shm
tmpfs                              5.0M     0  5.0M   0% /run/lock
tmpfs                              489M     0  489M   0% /sys/fs/cgroup
/dev/loop0                          62M   62M     0 100% /snap/core20/1328
/dev/loop1                          68M   68M     0 100% /snap/lxd/21835
/dev/loop2                          44M   44M     0 100% /snap/snapd/14978
/dev/sda2                          1.5G  106M  1.3G   8% /boot
tmpfs                               98M     0   98M   0% /run/user/1000
heiko@controller:~$ _
```

As can be seen in the image, 2 commands have been executed on the remote server which allowed to see the name, the version and the storage of the machine.

Assignment 2. Create your first Playbook

Difficulty: ★★★☆☆.

Now, let's create Playbooks for the same Ad hoc commands you created before. Create one Playbook per command. Provide screenshots of the configuration files and their results.

More information about Playbooks and their execution:

- *https://docs.ansible.com/ansible/latest/user_guide/playbooks_intro.html*
- *https://www.middlewareinventory.com/blog/ansible-playbook-example/*

Provide screenshots (evidence) for your solution. Always explain your evidence!

Solution: *In this case, the solution consists of creating an yml file containing the instructions to be executed by ansible. That is to say, first we create a new directory and inside we generate a file with the configuration shown in the following image.*

```
---
  - name: Check the remote host uptime
    hosts: dbservers
    tasks:
      - name: Execute the Uptime command over Command module
        register: uptimeoutput
        command: "uptime"

      - debug:
          var: uptimeoutput.stdout_lines
```

This task consists of monitoring the power-on time of the machine and if we run the command: "ansible-playbook playbook1.yml" we will see the following output.

```
heiko@controller:~/ansible-Playbooks$ ansible-playbook playbook1.yml

PLAY [Check the remote host uptime] ********************************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************************************
ok: [mysqlserver]

TASK [Execute the Uptime command over Command module] **************************************************************************************************************
changed: [mysqlserver]

TASK [debug] *******************************************************************************************************************************************************
ok: [mysqlserver] => {
    "uptimeoutput.stdout_lines": [
        " 20:55:04 up 52 min,  1 user,  load average: 0.00, 0.01, 0.00"
    ]
}

PLAY RECAP *********************************************************************************************************************************************************
mysqlserver                : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

heiko@controller:~/ansible-Playbooks$
```

Assignment 3. Create a Playbook with multiple commands

Difficulty: ★★★★☆.

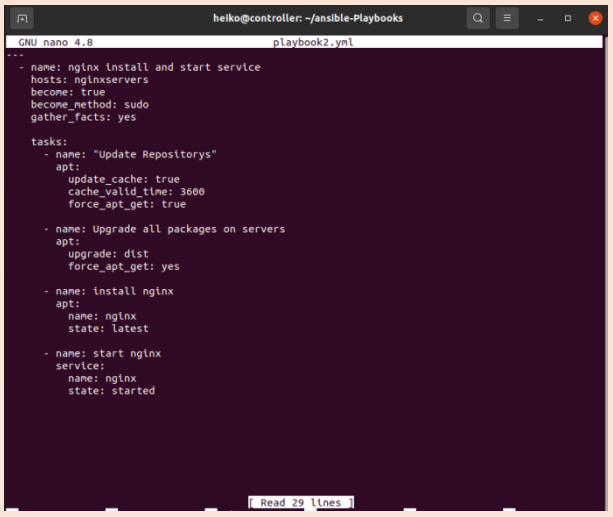
Now it's time to make a list of commands to be executed in a Playbook. Create a Playbook of your choice, that uses at least 4 different Modules. As evidence, explain your scenario (why did you execute the selected commands in given order) and provide the Playbook file and its result screenshots.

Provide screenshots (evidence) for your solution. Always explain your evidence!

Solution:

For this solution we have created a playbook that updates the servers and installs the latest version of nginx. This solution is designed for the large management that a company can do by having several servers with web pages.

First we generate a new file with the following configuration:



As can be seen there are 4 different tasks to complete the task.

Finally, an example of the execution of this playbook is shown.

```
heiko@controller:~/ansible-Playbooks$ ansible-playbook playbook2.yml

PLAY [nginx install and start service] *****************************************

TASK [Gathering Facts] *********************************************************
ok: [nginxserver]

TASK [Update Repositorys] ******************************************************
ok: [nginxserver]

TASK [Upgrade all packages on servers] *****************************************
ok: [nginxserver]

TASK [install nginx] ***********************************************************
ok: [nginxserver]

TASK [start nginx] *************************************************************
ok: [nginxserver]

PLAY RECAP *********************************************************************
nginxserver                : ok=5    changed=0    unreachable=0    failed=0    skipped=0    resc
ued=0    ignored=0
```