Ansible (II)

1.-a) Consulta la documentació oficial del mòdul "**fetch**" (recorda, la d'aquest i la de tota la resta de mòduls oficials es troba a https://docs.ansible.com/ansible/latest/collections/ansible/builtin) i digues què fa aquest "play". Prova'l (contra, ja ho pots veure, la màquina "B" de l'inventari que ja deus tenir creat del document anterior). ¿Què n'obtens, a la màquina "controladora" (dins de la carpeta on has executat la comanda *ansible-playbook*), després d'executar-lo? ¿Per a què serveix el paràmetre "flat" d'aquest mòdul?

```
- name: Un play
  hosts: B
  remote_user: pepito
  become: yes
  tasks:
  - name: Lerele
    fetch:  src=/etc/passwd  dest=/home/usuari
```

NOTA: El "warning" que apareix en executar el "playbook" dient que s'està usant l'intèrpret "/usr/bin/python" a les màquines "víctima" perquè no s'hi ha trobat cap altre intèrpret explícitament es pot silenciar si indiquem el valor "auto_silent" a la línia "interpreter_python" de l'arxiu "/etc/ansible/ansible.cfg" (per defecte val "auto").

aII) ¿Quina diferència fonamental hi ha, doncs, entre el mòdul "fetch" i el mòdul "**copy**" (vist a exercicis anteriors)?

b) Consulta la documentació oficial del mòdul "**unarchive**" i digues què faria aquest "play" (en concret per a què serveixen els paràmetres "src", "dest" i "remote_src"):

```
- name: Un play
  hosts: B
  remote_user: pepito
  become: yes
  tasks:
  - name: Lerele
    unarchive:  src=unfitxer.tgz  dest=/home/usuari
  - name: Larala
    unarchive:  src=altrefitxer.zip  dest=/opt  remote_src=yes
```

NOTA: Si *remote_src=no* (per defecte), la ruta relativa del fitxer indicat a *src=* ho serà respecte la ubicació del "playbook". Si *remote_src=yes*, la ruta relativa del fitxer indicat a *src=* ho serà respecte la carpeta personal de l'usuari amb què s'executi el "playbook" ("pepito", en aquest cas)

bII) ¿Quina diferència fonamental hi ha, doncs, entre el mòdul "unarchive" (amb el paràmetre *remote_src=no*) i el mòdul "copy"?

2.-a) Consulta la documentació oficial del mòdul "**blockinfile**" i digues què fa aquest "play". Prova'l.

```
- name: Un play
  hosts: victimes
  remote_user: pepito
  tasks:
  - name: Pirripipi
    blockinfile:
        path: /home/pepito/.bashrc
        block: |
            export HOLA=1234
            echo "Benvingut"
```

NOTA: El símbol "|" (propi de l'estàndard YAML) indica que els salts de línia existents al bloc de text indentat que va a continuació han de ser interpretats tal qual són: salts de línia d'un bloc únic. El símbol ">" faria, en canvi, que els salts de línia es transformessin en espais en blanc.
NOTA: Per defecte, el bloc s'inserirà al final del fitxer, a no ser que s'usin marques (com per exemple a l'apartat següent)

b) I aquest "play" (suposant que a les víctimes tens instal.lat un servidor Apache2)?

```
- name: Un play
  hosts: victimes
  remote_user: pepito
  become: yes
  tasks:
  - name: Polopolo
    blockinfile:
      path: /var/www/html/index.html
      insertafter: "<body>"
      block: |
        <h1>Hola</h1><p> El cel és blau </p>
```

NOTA: L'opció "insertafter" (així com també una anomenada "insertbefore", de significat obvi) admeten expressions regulars com a valors. Si al fitxer en qüestió hi ha més d'una ocurrència de l'expressió regular indicada, el bloc només s'inserirà després (o abans, segons l'opció emprada) de la darrera ocurrència, no pas en totes. D'altra banda, you can insert at the beginning of the file using "*insertbefore: BOF"*

bII) Un cop executat el "play" anterior, ¿què creus que faria aquest "play"?

```
- name: Un play
  hosts: victimes
  remote_user: pepito
  become: yes
  tasks:
  - name: Polopolo
    blockinfile:
      path: /var/www/html/index.html
      state: absent
```

c) ¿Com hauria de ser un "play" que fes servir el mòdul "blockinfile" per escriure les següents dues línies al final de l'arxiu "/etc/hosts"? ¿Si executes el mateix "play" vàries vegades, les línies s'afegiran repetides o no? Per què creus que passa això (pista: observa la marca que escriu Ansible abans i després de les línies)?

```
10.10.1.10  host1.domini.com
10.10.1.11  host2.domini.com
```

NOTA: Es pot actualitzar un bloc de línies ja inserit canviant el contingut del paràmetre *block*. Ansible primer comprova si hi ha un bloc amb el marcador i, si és així, actualitza l'arxiu remot amb el nou bloc.

3.-a) Consulta la documentació oficial del mòdul "**lineinfile**" (ja vist en algun exercici anterior) i digues concretament què fa aquest "play" (només possible a Fedora), i prova'l:

```
- name: Un play
  hosts: victimes
  remote_user: pepito
  become: yes
  tasks:
  - name: Pipipipi
    lineinfile:
      path: /etc/selinux/config
      regexp: '^SELINUX='
      line: SELINUX=enforcing
```

NOTA: L'opció "regex" només aplica el canvi de línia a la darrera línia trobada que quadri amb l'expressió regular (d'entre les vàries línies que es poden trobar). Si no es trobés cap (o no s'indica l'opció "regex"), la línia s'afegirà al final del fitxer
NOTA: Ansible usa la llibreria d'expressions regulars integrada de Python per modificar els fitxers mitjançant mòduls com "lineinfile" o "blockinfile". Consulta https://docs.python.org/3/library/re.html per saber quines expressions regulars estan disponibles

b) I aquest "play"?

```
- name: Un play
  hosts: victimes
  remote_user: pepito
  become: yes
  tasks:
  - name: Piupiupiu
    lineinfile:
      path: /etc/hosts
      regexp: '^127\.0\.0\.1'
      line: 127.0.0.1 localhost
```

c) I aquest?

```
- name: Un play
  hosts: victimes
  remote_user: pepito
  become: yes
  tasks:
  - name: Errequerre
    lineinfile:
      path: /etc/sudoers
      regexp: '^%wheel'
      state: absent
```

d) ¿Com faries servir el mòdul "lineinfile" per tal de comentar una línia present a l'arxiu "/etc/fstab" que fos aquesta: *192.168.1.12:/share /opt/punto nfs4 defaults 0 0* (l'hauràs d'escriure entre cometes per a què Ansible no es queixi dels dos punts que allà hi apareixen!) ?

NOTA: Es pot afegir l'opció "create: yes" si volem crear l'arxiu en qüestió en el cas de que no existís prèviament
NOTA: En comptes de l'opció "regex" es poden utilitzar les opcions "insertafter" o "insertbefore". La diferència principal amb "regex" és que en comptes de substituir la línia trobada per la nova, aquesta línia nova s'afegeix (ja sigui després de la trobada o abans, respectivament). Aquestes opcions només afegeixen la línia nova al voltant de la darrera **línia** trobada que quadri amb l'expressió regular indicada (d'entre les vàries línies que es poden haver trobat). Si no es trobés cap, la línia s'afegirà al final del fitxer. Si es vol que l'afegit s'apliqui al voltant de la primera línia trobada, cal especificar l'opció "firstmatch: yes". En el cas particular de voler afegir la línia sempre al principi del fitxer es pot escriure "*insertbefore:BOF*"

4.-a) Si en comptes de voler modificar només una línia d'un fitxer volem modificar totes les que concorden amb una determinada expressió regular, en comptes del mòdul "lineinfile" hem de fer servir el mòdul **"replace".** A partir de consultar la seva documentació oficial, digues què fa el següent "play" i prova'l

```
- name: Un play
  hosts: victimes
  remote_user: pepito
  become: yes
  tasks:
  - name: yyy
    replace:
      path: /etc/hosts
      regexp: '^127\.0\.0\.1.+$'
      replace: "127.0.0.1\tAulaAda-{{ ansible_default_ipv4.address.split('.')[3] }}"
```

NOTA: Es pot afegir l'opció "*after: cadena"* o "*before: cadena*" si volem aplicar els canvis només a les línies que apareguin després o abans de la cadena indicada

PISTA: Els valors escrits entre "{{" i "}}" indiquen valors de variables, els quals s'han hagut d'obtenir d'alguna forma externa al propi "playbook", com per exemple a través del paràmetre *-e* de la comanda *ansible-playbook* (o d'altres maneres). Més concretament, en aquest cas la variable *{{ ansible_default_ipv4.address }}* és un "fact" predefinit. Fixa't, d'altra banda, en l'ús del mètode *split()* de Python i de la conseqüent obtenció d'un determinat element de l'array generat.

b) Observa el següent "play" (que fa ús també del mòdul "replace", tot i que en aquest cas també es podria haver usat el mòdul "lineinfile") i dedueix per a què serviria (pots consultar https://wiki.debian.org/UnattendedUpgrades#Automatic_call_via_.2Fetc.2Fapt.2Fapt.conf.d.2F20auto-upgrades per saber per a què serveix l'arxiu en qüestió, només vàlid a Ubuntu):

```
- name: Un play
  hosts: victimes
  remote_user: pepito
  become: yes
  tasks:
  - name: xxx
      replace:
        path: /etc/apt/apt.conf.d/20auto-upgrades
        regexp: '^APT::Periodic::Update-Package-Lists "1";'
        replace: 'APT::Periodic::Update-Package-Lists "0";'
 - name: yyy
      replace:
        path: /etc/apt/apt.conf.d/20auto-upgrades
        regexp: '^APT::Periodic::Unattended-Upgrade "1";'
        replace: 'APT::Periodic::Unattended-Upgrade "0";'
```

5.- En aquest exercici veurem altres mòduls oficials interessants variats. Concretament:

a) Consulta la documentació oficial del mòdul "**get_url**" i digues què fa aquest "play" (prova-ho si vols):

```
- name: Un play
  hosts: victimes
  remote_user: pepito
  become: yes
  tasks:
  - name: Kikiriki
    get_url: url=http://mirror.slitaz.org/iso/rolling/slitaz-rolling.iso  dest=/opt/slitaz5.iso  mode=0640
```

NOTA: L'opció "dest" sempre fa referència a una ruta (de carpeta o de fitxer si es vol indicar un altre nom de l'original) ubicada a la màquina víctima, no pas la controladora. D'altra banda, si no s'indiqués, s'assumeix que és la ruta de la carpeta personal de l'usuari amb qui s'està entrant a la màquina víctima (en aquest cas, "pepito").

b) Consulta la documentació oficial del mòdul "**uri**" i digues què fa/mostra aquest "playbook". Prova'l:

```
- name: Un play
  hosts: victimes
  remote_user: pepito
  gather_facts: no
  tasks:
  - name: Chimpun
    uri: url=http://www.microsoft.com
    register: resposta
  - name: Veure resposta
    debug: msg= {{ resposta }}
```

PISTA: La línia "gather_facts:no", tal com s'explicà a la teoria, indica que no es vol fer el pas preliminar d'obtenir els "facts" de la màquina víctima en qüestió (això serveix per accelerar l'execució del "playbook" si no necessitem realment cap informació dels "facts" per poder realitzar les tasques del "playbook").

PISTA: La directiva "register" (de la qual en parlarem amb més profunditat més endavant) és una directiva general que serveix per guardar en una variable (anomenada com volguem) el resultat obtingut de la tasca en qüestió (en aquest cas, la resposta obtinguda de fer la petició a la URL indicada) per tal de poder-lo aprofitar en tasques següents
PISTA: El mòdul "debug" serveix per mostrar a pantalla el missatge indicat com a valor de la seva opció "msg". Aquest mòdul és necessari quan s'executen "playbooks" per "trencar" la sortida estàndard d'estadístiques què per defecte *ansible-playbook* ofereix. En aquest cas, el que mostra és el valor de la variable definida a la directiva "register" anterior. A Ansible, per indicar el valor d'una variable, s'ha d'escriure el seu nom entre "{{" i "}}"

bII) ¿Què veus a la sortida de l'execució del "playbook" anterior si ara fas que la seva darrera línia sigui *debug: msg= {{ resposta.status }}* (en lloc de la línia que hi havia)?

bIII) Afegeix la directiva *return_content=yes* al mòdul "uri" del "playbook" anterior. ¿Quin resultat diferent observes a pantalla ara en executar aquest nou "playbook"? En aquest sentit, ¿què veus si ara fas que la seva darrera línia sigui *debug: msg= {{ resposta.content }}* (en lloc de la línia que hi havia)?

c) Consulta la documentació oficial del mòdul "**hostname**" i digues què fa aquest "playbook". Prova'l (hauràs d'utilitzar el paràmetre *-e nomVariable=valor* de la comanda *ansible-playbook*...llegeix la "pista" inferior) i comprova que, efectivament, a les màquines víctima se'ls hi ha canviat el nom adientment:

```
- name: Un play
  hosts: victimes
  remote_user: pepito
  become: yes
  tasks:
  - name: xxx
    hostname: name={{ aula }}-{{ ansible_default_ipv4.address.split('.')[3] }}
```

PISTA: Els valors escrits entre "{{" i "}}" indiquen valors de variables, els quals s'han hagut d'obtenir d'alguna forma (externa al propi "playbook"). Concretament, el valor de la variable *{{ aula }}* s'haurà d'indicar en el paràmetre *-e* de la comanda *ansible-playbook* (aquesta variable representa el nom d'una aula de l'institut: "stallman", "torvalds", etc) però la variable *{{ ansible_default_ipv4.address }}* és, en canvi, un "fact" predefinit. Fixeu-vos, en aquest sentit, en l'ús del mètode *split()* de Python i de la conseqüent obtenció d'un determinat element de l'array generat.

A més dels mòduls oficials, existeixen molts altres mantinguts per la "comunitat", molts dels quals estan disponibles dins del paquet "**ansible-collection-community-general**"

6.-Instal·la el paquet mencionat a la màquina controladora (o bé, el paquet anomenat "ansible", el qual conté aquesta "collection" i moltes altres més), i a partir d'aquí, consulta (a https://docs.ansible.com/ansible/latest/collections/community/general/index.html) la documentació oficial dels mòduls mencionats a continuació per tal de realitzar els següents apartats:

NOTA: La llista de tots els mòduls, ja siguin oficials o de la comunitat (o d'altres categories) es pot trobar aquí: https://docs.ansible.com/ansible/latest/collections/index_module.html Més en general, la llista de no només els mòduls sinó d'altres elements que formen Ansible (com ara els "plugins", els "callback", els "lookup" i d'altres que estudiarem més endavant) es pot trobar aquí: https://docs.ansible.com/ansible/latest/collections/all_plugins.html

a) Crea un fitxer anomenat "/etc/somefile.conf" a la màquina "B" amb el següent contingut...:

```
[food]
quantity=4
vegan=true
[drinks]
temperature=cold
alcohol=false
```

...i tot seguit executa el següent "play" (el qual empra el mòdul "**ini_file**"). ¿Què li ha passat a aquest fitxer?

```
- name: Un play
  hosts: B
```

```
  remote_user: pepito
  become: yes
  tasks:
  - name: Chiquichiqui
    ini_file: path=/etc/somefile.conf  section=drinks  option=temperature  value=hot
```

b) Digues què fa aquest "play" (el qual empra el mòdul "**nmcli**") i prova'l (canviant adientment els valors de les opcions pels que necessitis):

```
- name: Un play
  hosts: C
  remote_user: pepito
  become: yes
  tasks:
  - name: Lilili
    nmcli:
        conn_name: miCon
        ifname: enp0s8
        type: ethernet
        ip4: 192.168.222.222/24
        gw4: 192.168.222.1
        dns4:
            - 1.1.1.1
            - 8.8.8.8
```

bII) I aquest?

```
- name: Un play
  hosts: C
  remote_user: pepito
  become: yes
  tasks:
  - name: Lilili
    nmcli:
        conn_name: miCon
        state: absent
```

c) Digues què faria aquest "play" (el qual empra el mòdul "**wakeonlan**"):

```
- name: Un play
  hosts: victimes
  remote_user: pepito
  tasks:
  - name: Kikiriki
    wakeonlan: mac="00:17:ef:ab:46:6a"
```

d) Digues què faria aquest "play" (el qual empra el mòdul "**parted**"):

```
- name: Un play
  hosts: victimes
  remote_user: pepito
  become: yes
  tasks:
  - name: Purrupupu
    parted:
      device: /dev/sdb
      number: 1
      state: present
      part_end: 1GiB
```

e) Digues què faria aquest "play" (el qual empra el mòdul "**filesystem**"):

```
- name: Un play
  hosts: victimes
  remote_user: pepito
  become: yes
  tasks:
  - name: Perrepepe
    filesystem:
      device: /dev/sdb1
      fstype: ext4
```

f) A continuació es presenta un "playbook" que configura certs aspectes de l'escriptori Gnome mitjançant un conjunt de mòduls ja coneguts a més del mòdul "**dconf**". Digues per a què servirien cadascuna de les tasques (anomenades "MisteriX") aquí escrites:

```
- name: Un play
  hosts: unaMaquinaAmbGnome
  remote_user: pepito
  become: yes
  tasks:
  - name: Misteri1
    copy:   src="nextcloud.desktop"    dest="/usr/share/applications"
  - name: Misteri2 (1ª part)
     file:    path=~/.config/autostart   state=directory
  - name: Misteri2 (2ª part)
     copy:  src="/usr/share/applications/nextcloud.desktop"   dest="~/.config/autostart"   remote_src=yes
  - name: Misteri3
    dconf: key="/org/gnome/desktop/interface/clock-show-seconds"  state=read
  - name: Misteri4
    dconf: key="/org/gnome/desktop/interface/clock-show-seconds"  value="true"  state=present
  - name: Misteri5
    dconf:
        key: "/org/gnome/desktop/wm/preferences/button-layout"
        value: "'minimize,maximize,close'"
        state: present
```

PISTA: A la carpeta "/usr/share/applications" es troben tots els arxius "*.desktop" junts corresponents als diferents programes instal.lats al sistema. Recordeu que els arxius "*.desktop" són accesos directes gràfics i els podem trobar, si així els hem escollit,a la barra d'icones del Gnome o sobre l'escriptori mateix, per exemple.
PISTA: A la carpeta "~/.config/autostart" (que pot no existir inicialment) es poden incloure els arxius "*.desktop" corresponents als programes que volguem posar en marxa automàticament en iniciar sessió al Gnome.
PISTA: El mòdul "dconf" serveix per gestionar la base de dades de configuració del Gnome, la qual està en un format propi anomenat DConf, precisament. En aquesta base de dades les diferents opcions (que no són més que parelles clau<->valor, on el valor pot ser cadena, booleà o numèric) es troben penjant dins d'un arbre des d'una determinada categoria inicial (per exemple, "org") i a partir d'aquí, sota alguna de les subcategories escalonades que hi hagi. Es pot manipular interactivament aquesta base de dades amb la comanda gràfica "dconf-editor".

g) Consulta els següents enllaços i digues per a què serveixen els mòduls corresponents:

* https://docs.ansible.com/ansible/latest/collections/community/general/ipify_facts_module.html
* https://docs.ansible.com/ansible/latest/collections/community/general/mail_module.html
* https://docs.ansible.com/ansible/latest/collections/community/general/telegram_module.html
* https://docs.ansible.com/ansible/latest/collections/community/general/jabber_module.html
* https://docs.ansible.com/ansible/latest/collections/community/general/modprobe_module.html
* https://docs.ansible.com/ansible/latest/collections/community/general/archive_module.html

Encara hi ha més mòduls disponibles en altres "collections". Per exemple, el paquet "**ansible-collection-ansible-posix**" inclou diversos mòduls relacionats específicament amb sistemes que segueixen l'estàndard POSIX (és a dir, bàsicament els que són de la família UNIX: Linux, BSD, etc)

7.-Instal·la el paquet mencionat a la màquina controladora (o bé el paquet "ansible", que inclou moltes "collections"), i a partir d'aquí, consulta (a https://docs.ansible.com/ansible/latest/collections/ansible/posix/index.html) la documentació oficial dels mòduls mencionats a continuació per tal de realitzar els següents apartats:

> NOTA: La llista de totes les col·leccions disponibles, ja siguin oficials o de la comunitat es pot trobar aquí: https://docs.ansible.com/ansible/latest/collections/index.html

a) Digues què faria aquest "play" (el qual empra el mòdul "**mount**"):

```
- name: Un play
  hosts: victimes
  remote_user: pepito
  become: yes
  tasks:
  - name: Pirripipi
    mount:
      path: /mnt/dvd
      src: /dev/sr0
      fstype: iso9660
      opts: ro,auto
      state: present
```

aII) I aquest "play"?

```
- name: Un play
  hosts: victimes
  remote_user: pepito
  become: yes
  tasks:
  - name: Pirripipi
    mount:
      path: /mnt/dvd
      state: unmounted
```

b) Digues què faria aquest "play" (el qual empra el mòdul "**synchronize**"); pots provar-lo si vols:

```
- name: Un play
  hosts: B
  remote_user: pepito
  tasks:
  - name: Lerele
    synchronize:
        src: ..
        dest: /opt
```

> NOTA: El mòdul "synchronize" es basa en l'eina "rsync", la qual haurà d'estar instal.lada tant a la màquina controladora Ansible com a les màquines "víctima". Rsync és una comanda que permet fer còpies recursives de fitxers i carpetes entre sistemes remots fent ús del protocol "rsync://" (o també en un mateix sistema).

bII) I aquest "play", què faria?

```
- name: Un play
  hosts: B
  remote_user: pepito
```

```
tasks:
- name: Lerele
  synchronize:
      src: /etc
      dest: rsync://192.168.12.123/carpeta
  delegate_to: B
```

NOTA: La directiva *delegate_to* serveix per a què la màquina de referència des d'on es llença la tasca (és a dir, la que normalment és la controladora Ansible) sigui una altra màquina diferent. En aquest cas concret, per tant, la màquina on hi ha la carpeta indicada a *src*= serà B.

c) Consulta els següents enllaços i digues per a què serveixen els mòduls corresponents:

* https://docs.ansible.com/ansible/latest/collections/ansible/posix/acl_module.html
* https://docs.ansible.com/ansible/latest/collections/ansible/posix/sysctl_module.html
* https://docs.ansible.com/ansible/latest/collections/ansible/posix/firewalld_module.html
* https://docs.ansible.com/ansible/latest/collections/ansible/posix/authorized_key_module.html