

Fase 4 - Activitat 5.3: Ampliació de coneixements sobre Vagrant, Git i Github

0- Identificació del grup i activitat:

Curs: ASIX2

Projecte: GP2 DevOps i Cloud Computing

Fase: 4

Activitat: 5.3

Grup: Individual

Membres: ---

1.- Objectiu de l'activitat

Els objectius objectius d'aquesta activitat són:

- Aprendre noves ordres per treballar amb **Vagrant**
- Aprendre noves ordres per treballar amb **Git i Github**
- Ampliar els nostres coneixements sobre **Vagrant**.
- Ampliar els nostres coneixements sobre **Git i Github**.
- Tenir els equips de tots els membre de tots els grups del projecte preparats per treballar plenament amb **Vagrant, Git i Github**.

2.- Ampliació de Vagrant

2.1- Mostrant les màquines virtuals gestionades amb Vagrant amb: vagrant global-status --prune

a) Abans de començar i per evitar conflictes entre ports oberts per diferents màquines virtuals, farem un canvi en la **redireccionament de ports** (port forwarding) de la màquina **m08uf1pr2**. Accedeix a la carpeta de **projectes** → **m08uf1pr2** i modifica els paràmetres **config.vm.network "forwarded_port"** del fitxer **Vagrantfile** de la següent manera:

```
config.vm.network "forwarded_port", guest: 80, host: 18000
config.vm.network "forwarded_port", guest: 8080, host: 18080
config.vm.network "forwarded_port", guest: 8888, host: 18888
```

c) Posa en marxa les màquines virtuals que vas crear a l'activitat **gp1f4a5.2**. Has de tenir dues màquines: **gp1f4a5.2** i **xxyyzz** (a on **xxyyzz** són les 2 primeres lletres del teu nom, 1r cognom i 2n cognom):

- Accedeix a la carpeta **m08uf1pr2** i executa: **vagrant up**
- Accedeix a la carpeta **gp1f4a5.2** i executa: **vagrant up**

d) Torna al teu directori personal i comprova que les dues màquines estan en marxa executant:

```
vagrant global-status --prune
```

i comprova que el resultat és similar a aquest:

```
daniel@sh3:~$ vagrant global-status --prune
id      name      provider  state  directory
-----
89c24f4 default  virtualbox running /home/daniel/tmp/projectes/m08uf1pr2
a8575ee default  virtualbox running /home/daniel/tmp/projectes/gp1f4a5.2/vm

The above shows information about all known Vagrant environments
on this machine. This data is cached and may not be completely
up-to-date (use "vagrant global-status --prune" to prune invalid
entries). To interact with any of the machines, you can go to that
directory and run Vagrant, or you can use the ID directly with
Vagrant commands from any directory. For example:
"vagrant destroy 1a2b3c4d"
```

Prova que aquesta ordre funciona des de qualsevol altre directori del teu sistema i que per tant **tens una ordre que et mostra totes les màquines virtuals gestionades amb Vagrant**.

e) Ara accedeix a la carpeta **m08uf1pr2** una altra vegada i executa: **vagrant halt**

f) A continuació executa: **vagrant global-status --prune**

g) Comprova que mostra totes les màquines virtuals gestionades amb Vagrant i també el seu estat a la columna "state" i el directori des d'on es gestionen a "directory". Això és molt útil per poder treballar amb totes les màquines gestionades des de **Vagrant**.

2.2.- Comprovant l'estat d'una màquina virtual amb vagrant status. Esborrant una màquina virtual gestionada amb Vagrant utilitzant l'ordre: vagrant destroy

a) Accedeix a la carpeta **m08uf1pr2** i comprova que la màquina està aturada executant:

vagrant status

i comprova que el resultat és similar a aquest:

```
daniel@sh3:~/tmp/projectes/m08uf1pr2$ vagrant status
Current machine states:

default                poweroff (virtualbox)

The VM is powered off. To restart the VM, simply run `vagrant up`
```

La diferència de **vagrant status** amb **vagrant global-status --prune** és que la primera val per comprovar l'estat monés d'una màquina i has d'estar dins del directori a on es troba el fitxer **Vagrantfile** de la màquina. En aquest cas veiem que està aturada.

b) Un cop sabem que està aturada una màquina, la podem **esborrar** executant l'opció **destroy**. Dins de la carpeta **m08uf1pr2** executa:

vagrant destroy

i a la pregunta:

```
default: Are you sure you want to destroy the 'default' VM? [y/N]
```

contesta **y**, o sigui:

```
default: Are you sure you want to destroy the 'default' VM? [y/N] y
```

Comprova que dins de **VirtualBox** que la màquina ha desaparegut, i després, per no deixar cap rastre de la màquina, esborra el directori **.vagrant** que encara hi ha dins del directori **m08uf1pr2**. Dins de **m08uf1pr2** executa:

rm -rf .vagrant

2.3.- Comprovant els ports oberts per una màquina virtual amb vagrant port.

a) Accedeix a la carpeta de **projectes** → **gp1f4a5.2** → **vm** i comprova que està en marxa executant **vagrant status** i que es mostra alguna cosa d'aquest estil:

```
daniel@sh3:~/tmp/projectes/gp1f4a5.2/vm$ vagrant status
Current machine states:

default                running (virtualbox)

The VM is running. To stop this VM, you can run `vagrant halt` to
shut it down forcefully, or you can run `vagrant suspend` to simply
suspend the virtual machine. In either case, to restart it again,
simply run `vagrant up`.
```

b) Ara executa:

vagrant port

i comprova que el resultat és aquest:

```
daniel@sh3:~/tmp/projectes/gp1f4a5.2/vm$ vagrant port
The forwarded ports for the machine are listed below. Please note that
these values may differ from values configured in the Vagrantfile if the
provider supports automatic port collision detection and resolution.

  22 (guest) => 2200 (host)
  80 (guest) => 8000 (host)
 443 (guest) => 8443 (host)
```

Això vol dir que:

- El port **22** de la màquina **virtual** es redirecciona cap el port **2200** de la màquina **física**
- El port **80** de la màquina **virtual** es redirecciona cap el port **8000** de la màquina **física**
- El port **443** de la màquina **virtual** es redirecciona cap el port **8443** de la màquina **física**

Per tant tens una ordre que et **mostra tots els ports oberts dins de la màquina física que es redireccionen cap a ports de la màquina virtual** gestionada amb **Vagrant**.

És important tenir en compte que aquesta ordre només funciona si la **màquina està en funcionament** i l'**ordre s'executa dins del directori** a on es troba el fitxer **Vagrantfile** de la màquina amb la qual volem treballar.

2.4.- Treballant amb boxes de Vagrant utilitzant l'ordre vagrant box.

a) Un **box** de Vagrant és una imatge d'un sistema operatiu bàsic, un conjunt d'aplicacions ja instal·lades i una configuració predefinida que un cop descarregada es pot personalitzar per crear una màquina virtual que s'adapti a les nostres necessitats. Cada cop que es crea un fitxer **Vagrantfile**, s'haurà d'especificar el box base a partir del qual es treballa.

Habitualment els boxes de Vagrant es poden descarregar del núvol [Vagrant Cloud](#) de l'empresa **HashiCorp** **Vagrant** però a també es possible crear un núvol corporatiu privat amb boxes que una empresa necessiti per realitzar les seves activitats, o crear-nos els nostres propis boxes i allotjar-los en un servidor gestionat per nosaltres mateixos.

Existeixen boxes de Vagrant diferent per la mateixa imatge en funció de si utilitzes com a gestor de màquines virtuals **virtualbox**, **vmware**, **aws** (amazon web services), **libvirt**, **parallels**, etc...Als gestors de màquines virtual se'ls anomena **providers**.

b) Si vols veure la llista de boxes descarregats dins del teu sistema executa (des de qualsevol directori) l'ordre:

vagrant box list

i el resultat hauria de ser similar a això:

```
daniel@sh3:~$ vagrant box list
debian/bullseye64 (virtualbox, 11.20230615.1)
generic/debian10 (virtualbox, 4.3.2)
generic/debian11 (virtualbox, 4.3.2)
daniel@sh3:~$
```

En aquest exemple, tots els boxes descarregats serveixen per ser utilitzats amb **virtualbox** com a **provider**.

c) Tenir un box de Vagrant emmagatzemat al disc dur estalvia molt de temps de creació de la màquina virtual i potser t'interessa tenir al disc una box que faràs servir sovint. Si vols descarregar-lo abans de començar a utilitzar-lo executa l'ordre **vagrant box add --provider virtualbox <nom_box>** . Per exemple, per afegir el box **generic/debian12** per **virtualbox** hauries d'executar:

```
vagrant box add --provider virtualbox generic/debian12
```

i el resultat hauria de ser similar a aquest:

```
daniel@sh3:~$ vagrant box add --provider virtualbox generic/debian12
==> box: Loading metadata for box 'generic/debian12'
box: URL: https://vagrantcloud.com/generic/debian12
==> box: Adding box 'generic/debian12' (v4.3.2) for provider: virtualbox
box: Downloading: https://vagrantcloud.com/generic/boxes/debian12/versions/4.3.2/providers/virtualbox/unknown/vagrant.box
box: Calculating and comparing box checksum...
==> box: Successfully added box 'generic/debian12' (v4.3.2) for 'virtualbox'!
```

d) Un box de Vagrant ocupa espai de disc dur que normalment pot ser des d'uns centenars de MiB a uns pocs GiB. Si vols alliberar espai de disc ocupat per un box que no utilitza cap màquina virtual i que creus que no utilitzaràs en el futur, executa l'ordre **vagrant box remove --provider virtualbox <nom_box>**. Per exemple, per esborrar el box **generic/debian12** per **virtualbox** hauries d'executar:

```
vagrant box remove --provider virtualbox generic/debian12
```

i el resultat hauria de ser aquest:

```
daniel@sh3:~$ vagrant box remove --provider virtualbox generic/debian12
Removing box 'generic/debian12' (v4.3.2) with provider 'virtualbox'...
```

2.5.- Canvis en la configuració d'una màquina gestionada amb Vagrant utilitzant vagrant reload

Modificar la configuració d'una màquina virtual és fàcil. Només cal modificar el paràmetre desitjat dins del fitxer **Vagrantfile** i a continuació executar (dins del directori a on es troba el fitxer) l'ordre:

```
vagrant reload
```

i la nova configuració tindrà efecte immediat. Per exemple, podem fer un canvis simples a la màquina amb nom **xyyzz** de l'activitat **gp1f4a5.2**:

- Farem que el seu nou nom sigui **xyyzz_gp1f4a5.2** perquè sigui més fàcil d'identificar. Canvia el paràmetre **v.name** així: **v.name=xyyzz_gp1f4a5.2** (i recorda el significat de **xyyzz**).
- Farem que sigui membre del grup de màquines **ASIX2**. Canvia el paràmetre **v.customize** així: **v.customize ['modifyvm', :id, '--clipboard', 'bidirectional', '--groups', '/ASIX2']**
- Farem que treballi amb 1024MiB de RAM. Canvia el paràmetre **v.memory** així: **v.memory = 1024**
- Executa: **vagrant reload**
- Comprova que els canvis han tingut efecte des de la pantalla de gestió de la màquina de **virtualbox**.

Aquesta ordre es pot executar tant si la màquina està en marxa com si està aturada.

2.6.- Aprovisionament de programari amb vagrant provision

a) Aprovisionar programari consisteix en afegir o esborrar programari base a una màquina ja existent modificant el fitxer **Vagrantfile** i executant l'ordre **vagrant provision** amb la màquina virtual en execució.

b) Per exemple, si volem afegir el servidor proxy **nginx** a la màquina **xyyzz_gp1f4a5.2** i també l'eina de depuració de PHP **xdebug** podem afegir al final de la secció **config.vm.provision** (just abans del paràmetre **SHELL**) les ordres per aprovisionar:

```
sudo apt-get install -y nginx  
sudo apt-get install -y php7.4-xdebug
```

i a continuació executar (:

```
vagrant provision
```

c) Si vols ara eliminar per exemple el paquet **python3.9** de la màquina **xyyzz_gp1f4a5.2**, podem afegir al final de la secció **config.vm.provision** l'ordre aprovisionar:

```
sudo apt-get remove -y python3.9
```

i a continuació executar:

```
vagrant provision
```

2.7.- Inicialitzant un directori per treballar amb Vagrant amb l'ordre vagrant init

a) Si has començat un projecte des de zero, pots inicialitzar el directori de treball simplement creant un fitxer **Vagrantfile** inicial utilitzant l'ordre:

```
vagrant init
```

que després podrà ser fàcilment modificat i personalitzat.

b) També pots executar l'ordre **init** indicant el box amb el qual vols treballar. Per exemple:

```
vagrant init generic/debian12
```

No és una ordre obligatòria perquè simplement podem crear el nostre fitxer **Vagrantfile** manualment i també funcionarà perfectament.

3.- Ampliació de Git

3.1.- Afegint un nou commit local i pujant-lo al dipòsit remot amb l'ordre git push

3.1.1- Abans de començar

a) Crea una carpeta de nom **pizza** dins de la teva carpeta **projectes**. Descarrega aquests 2 arxius dins de la carpeta **pizza**:

- El fitxer [pizza.html](#)
- El fitxer [pizza.php](#)

b) Inicialitza un dipòsit dins de la carpeta **pizza**, fes un **add** dels fitxers **pizza.html** i **pizza.php** i finalment fes un **commit** amb el comentari "**Commit 1 del projecte pizza**".

c) Crea un dipòsit remot dins del teu compte de **GitHub** de nom **pizza** i de tipus **Public**. Sincronitza el dipòsit local amb el remot i comprova que els 2 dipòsits tenen el mateix fitxers i les mateixes versions pujades. Comprova que:

- Hi han els 2 fitxers
- Que la branca inicial es diu main en els dos dipòsits
- Que només hi ha un commit en els 2 dipòsits
- El codi Hash del commit en els 2 dipòsits és el mateix.

3.1.2.- Afegint un nou commit local i pujant-lo al dipòsit remot amb l'ordre git push

a) Afegeix una nova línia després de l'etiqueta `</form>` de la línia **16** del fitxer **html** amb aquest contingut:

Nom del desenvolupador: `xyyzz
`

a on **xyyzz** són **xx** → les dues primeres lletres del nom, **yy** → les dues primeres lletres del primer cognom, **zz** --> les dues primeres lletres del segon cognom.

b) Desa el fitxer amb el canvi, i torna a fer una add i un commit amb el comentari "**Commit 2 del projecte pizza**".

c) Puja el nou commit al dipòsit remot. Executa:

`git push -u origin main`

d) Comprova que al dipòsit remote:

- Que la branca main ara hi han 2 commits
- Que els codis Hash dels commit en els 2 dipòsits són el mateixos.
- Que el codi dels fitxer del darrer commit coincideix en el 2 dipòsits

3.2.- Baixant un nou commit del dipòsit remot al dipòsit local amb git pull

a) Dins del directori **projectes** excuta l'ordre de **git** per **clonar** el dipòsit **gp1f4a5.3** que té la següent URL: **<https://github.com/globproj2/gp1f4a5.3.git>**

b) Accedeix a la carpeta **gp1f4a5.3** que s'ha creat i comprova que hi ha un un únic commit executant l'ordre:

`git log`

c) Espera a que el propietari del dipòsit (o sigui, el professor) puji una nova versió del codi. Un cop hagi estat pujada la nova versió executa:

`git pull origin main`

i comprova que s'ha baixat al teu dipòsit local la darrera versió pujada al dipòsit remot executant novament:

`git log`

3.3.- Navegant entre versions amb git checkout

a) Dins del directori **projectes** clona el dipòsit **m08uf2pr1** que es troba a l'adreça <https://github.com/dacomo2021daw2/m08uf2pr1.git>.

b) Accedeix a **m08uf2pr1** i executa:

```
git log --pretty=oneline
```

i comprova que la darrera versió té el codi **hash** següent: **7fa86f61da66db0b31ed0cb73c9ef5c34f62680c**.

c) Comprova que la mida del fitxer **Vagrantfile** amb l'ordre: `ls -ls Vagrantfile`

d) Comprova ara la mida del fitxer README en la versió del projecte que s'identifica amb el codi **hash** següent: **da5a0d89833e974b4f73b3d692ffb07617be3506**.

e) Passa a treballar amb la versió del projecte identificada amb el codi **hash** indicat a l'apartat anterior. Executa l'ordre:

```
git checkout da5a0d
```

Comprova ara la mida del fitxer **Vagrantfile** executant: `ls -ls Vagrantfile`. Ha canviat?. Quina és la diferència en bytes entre versions?.

f) Torna a la darrera versió del projecte que s'identifica amb el codi **hash** següent: **7fa86f61da66db0b31ed0cb73c9ef5c34f62680c**. Executa:

```
git checkout 7fa86f
```

g) Comprova les diferències entre les versions del fitxer README que hi ha entre el **commit** identificat amb el **hash** següent: **da5a0d89833e974b4f73b3d692ffb07617be3506** i l'identificat amb el **hash** següent: **7fa86f61da66db0b31ed0cb73c9ef5c34f62680c**. Executa l'ordre:

```
git diff 7fa86f da5a0d Vagrantfile
```

Quin ha estat el canvi entre els fitxers de les dues versions?

3.4- Directori de treball. Àrea de preparació. Dipòsit. Ordre git status

a) **Director de treball (Working directory)**: El directori local a on es troben els fitxers del nostre projecte amb els quals estem treballant (llegint, modificant,...) i el directori **.git**. Per exemple, seria el directori **pizza** de l'apartat 3.1, o el directori **gp1f4a5.3** de l'apartat 3.2 o el directori **m08uf2pr1** de l'apartat 3.3. Al directori de treball van a parar els fitxers de les versions del projecte quan fem un **checkout**.

b) **Àrea de preparació (Staging area / index area)**: És l'espai dins de **.git** a on s'emmagatzemen el conjunt de fitxers que seran enviats al dipòsit en el següent commit.

c) **Dipòsit (repository)**: Lloc a on s'emmagatzemen el conjunt de **commits** realitzats. Cada **commit** és correspon a una **versió** del projecte. Es troba dins del directori **.git**.

d) L'ordre **git status** ens dona informació sobre:

- L'estat dels fitxers del directori de treball
- L'estat dels fitxers de l'àrea de preparació

e) Accedeix al dipòsit **m08uf2pr1**. Modifica el fitxer **Vagrantfile**. Afegeix a la línia **11** el següent paràmetre:

```
config.vm.network "public_network"
```

Comprova l'estat del directori de treball i àrea de preparació. Executa:

```
git status
```

i comprova que el resultat és com aquest:

```
daniel@sh3:~/tmp/GP1/projectes/m08uf2pr1$ git status
HEAD detached at 7fa86f6
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   Vagrantfile

no changes added to commit (use "git add" and/or "git commit -a")
```

Això vol dir que:

- El fitxer **Vagrantfile** ha estat modificat però encara no s'ha enviat a l'àrea de preparació
- En aquests moments a l'àrea de preparació no hi ha cap fitxer.

f) Executa ara:

```
git add Vagrantfile
```

```
git status
```

i comprova que el resultat és com aquest:

```
daniel@sh3:~/tmp/GP1/projectes/m08uf2pr1$ git status
HEAD detached at 7fa86f6
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   Vagrantfile
```

Això vol dir que:

- No hi ha cap fitxer modificat que no hagi estat enviat a l'àrea de preparació (cap fitxer de color vermell).
- En aquests moments a l'àrea de preparació hi ha un fitxer preparat per ser enviat que és el fitxer **Vagrantfile**.

Lliurament de l'activitat

Aquesta activitat s'avalua per mitjà de l'assistència a classe el **dimarts 3-10-23** de les **18.15** a **20.05**