

AWS - Entorn de treball per proves i desenvolupament

- 1 Matriu versions software verificades
- 2 Instal·lar VirtualBox
- 3 Instal·lar Vagrant
 - 3.1 Instal·lar plugins de vagrant
- 4 Configurar la MV
 - 4.1 Fitxer Vagrantfile
 - 4.2 Carpetes compartides
 - 4.3 Arrencar la MV
- 5 Treballar amb la MV
 - 5.1 Altres comandes Vagrant
 - 5.2 git
 - 5.3 saml2aws
 - 5.4 packer
 - 5.5 ansible
 - 5.6 terraform
 - 5.7.1 Algunes comandes i tips & tricks

Matriu versions software verificades

En cas de problemes, assegura't de tenir les versions verificades en una data concreta:

Versió de... App / Plugin / VM / Mòdul ↓		Versions verificades en DATA ↓	
		06/08/2020	
Windows		7	
PowerShell		Major Minor Build Revision ----- 5 1 14409 1018	
VirtualBox		6.1.12r139181 (Qt5.6.2)	
Vagrant		2.2.9	
	vagrant-proxyconf	2.0.10	
	vagrant-vbguest	0.24.0	
	vagrant-winfsd	1.4.0	
VM		secavm.0.2032.1.box	
	Terraform	v0.12.19	
	provider.awsan	v.2.70.0	
	Git	1.8.3.1	
	Ansible	2.9.10	
Git		2.28.0.windows.1	
Visual Studio Code		1.47.3 (23/07/2020)	
	Remote - SSH	v0.51.0	
	Remote - SSH: Editing Configuration Files	v0.51.0	
	Remote - Containers	v0.128.0	
	Remote - WSL	v0.44.4	
	Remote Development	v0.20.0	
	Vagrantfile Support	v.0.0.7	
	HashiCorp Terraform	v.2.1.1	
	JenkinsFile Support	v.1.1.0	
SourceTree		3.3.9	

Instal·lar VirtualBox

Podem descarregar-nos l'instal·lable de:

<https://www.virtualbox.org/wiki/Downloads>

L'instal·lable de VirtualBox es fa amb un assistent. De tota manera en el següent link teniu la documentació de instal·lació a Windows:

https://www.virtualbox.org/manual/ch02.html#installation_windows

Instal·lar Vagrant

Cal instal·lar vagrant i pujar de versió el powershell de la màquina (això últim només per entorns amb Windows 7). En tots dos casos demana reiniciar la màquina, s'ha de fer:

- Vagrant: <https://www.vagrantup.com/downloads.html>
- Powershell: <https://www.microsoft.com/en-us/download/details.aspx?id=54616> (és un paquet inclòs al 'Windows Management Framework')

Instal·lar plugins de vagrant

Un cop instal·lat, s'ha d'anar al CMD de Windows i executar:

```
vagrant plugin install vagrant-vbguest
```

```
vagrant plugin install vagrant-proxyconf
```

```
vagrant plugin install vagrant-winrm
```

Configurar la MV

Repository imatges: <http://vm-dev.pub.dtvclocal/>

Fitxer Vagrantfile

En el directori en el que volem treballar hi posem el fitxer **Vagrantfile**. Aquest fitxer té la següent configuració:

```
Vagrant.configure("2") do |config|
```

```
  config.vm.box = "http://vm-dev.pub.dtvclocal/secavm.0.2032.1.box"
```

```
  config.vm.box_check_update = true
```

```
  config.vm.synced_folder '.', '/vagrant', type: 'nfs'
```

```
  config.vm.synced_folder 'D:/vm/shared', '/home/dev/shared', type: 'nfs'
```

```
  config.vm.synced_folder 'D:/vm/repository', '/home/dev/.m2/repository', type: 'nfs'
```

```
  config.vm.hostname = "secavm"
```

```
  config.vm.network :private_network, ip: "192.168.56.104"
```

```
  config.vm.network "forwarded_port", guest: 3128, host: 3128
```

```
  config.vm.network "forwarded_port", guest: 22, host: 2222
```

```
  if Vagrant.has_plugin?("vagrant-proxyconf")
```

```
    config.proxy.http = "http://proxy-test.pub.dtvclocal:3128/"
```

```
    config.proxy.https = "http://proxy-test.pub.dtvclocal:3128/"
```

```
    config.proxy.no_proxy = "localhost,127.0.0.1"
```

end

```
config.vm.provider :virtualbox do |v|  
  v.memory = 1024  
  v.cpus = 1  
end
```

end

Aquest fitxer tan sols és una plantilla d'inici, si cal i/o volem podem modificar les configuracions del fitxer:

- podem donar-li més recursos i incrementar els paràmetres memory i cpus
- podem modificar la IP i els ports d'accés, o afegir-ne algun més si fos necessari
- les carpetes compartides les podem treure si no ens cal o podem modificar el path en cas necessari
- ... tot el que s'us acudeixi que hagueu de canviar.

Carpetes compartides

si manteniu les carpetes compartides D:/vm/shared i D:/vm/repository abans d'arrencar la màquina cal assegurar que existeixen

Arrencar la MV

Per arrencar la màquina des de un CMD de Windows ens situem al directori de treball i executem la següent comanda:

```
vagrant up
```

Això ens arrencarà la màquina i hi podrem accedir per ssh des de localhost pel port 2222.

Per entrar teniu l'usuari:

- dev (password: 123456)

Treballar amb la MV

Altres comandes Vagrant

Si executeu vagrant a seques us sortiran totes les opcions possibles.

Les més habituals seran:

- Per arrencar la màquina:
vagrant up
- Per destruir la màquina i l'entorn:
vagrant destroy
- Per aturar i tancar la màquina:
vagrant halt
- Per deixar la màquina en suspensió:
vagrant suspend
- Per reanudar una màquina que està en suspensió:
vagrant resume
- Per accedir a la MV per ssh des del cmd:
vagrant ssh

- Per mostrar tots els ports mapejats:
vagrant port

git

Tant si us descarregueu el projecte des de Windows per després treballar des de les carpetes compartides de la MV com si ho feu directament des del prompt de la MV hauríeu de configurar git tal que:

```
git config --global core.autocrlf input
git config --global i18n.commitEncoding utf-8
git config --global i18n.logOutputEncoding utf-8
```

saml2aws

La màquina te instal·lada l'aplicació saml2aws que permet fer servir el login d'ADFS per connectar a AWS. El projecte el podeu trobar a: <https://github.com/Versent/saml2aws/blob/master/README.md>

El primer cop cal configurar l'eina:

```
/opt/saml2aws/saml2aws configure
```

Li passarem una configuració com aquesta però proveint el nostre login per facilitar la feina:

```
? Please choose a provider: ADFS
? Please choose an MFA Auto
? AWS Profile saml
? URL https://adfs.ccma.cat
? Username ad-ccrtv\your_login
```

```
account {
  URL: https://adfs.ccma.cat
  Username: ad-ccrtv\your_login
  Provider: ADFS
  MFA: Auto
  SkipVerify: false
  AmazonWebservicesURN: urn:amazon:webservices
  SessionDuration: 3600
  Profile: saml
  RoleARN:
}
```

Configuration saved for IDP account: default

Per logar-nos podem fer servir la comanda:

```
/opt/saml2aws/saml2aws login
```

Això ens demanarà les nostres credencials li hem de passar el nostre usuari : ad-ccrtv\meu_login, la nostra contrasenya i el rol que volem assumir de tots els que ens mostri.

Per a que a la nostra sessió es mantinguin les credencials podem executar la comanda:

```
eval $(/opt/saml2aws/saml2aws script)
```

Per executar comandes amb un rol sobre un compte diferent al de sso (on hem fet login) cal fer la següent operativa

- Configurar el fitxer ~/.aws/config tal i com segueix

```
[profile portal-ccmaAdmin]
```

```
source_profile=saml
```

```
role_arn=arn:aws:iam::569477207866:role/ccmaAdmin
```

```
[profile shared-ccmaAdmin]
```

```
source_profile=saml
```

```
role_arn=arn:aws:iam::317014775945:role/ccmaAdmin
```

...

```
[profile seca-labs-ccmaSECAlabsRol]
```

```
source_profile=saml
```

```
role_arn=arn:aws:iam::310106061799:role/ccmaSECAlabsRol
```

...

- Crear entrades de profile per cada rol que es vulgui utilitzar, configurant per cada un d'ells el role_arn amb el compte i el nom de role a assumir.
- Per executar una comanda amb un dels profile anterior, utilitzar
/opt/saml2aws/saml2aws exec --exec-profile shared-ccmaAdmin 'aws s3 ls'

packer

Aquesta eina permet crear imatges per planxar a màquines. En el nostre cas ho fem servir per crear AMIs a AWS.

Alguns projectes que tenim a Bitbucket amb Packer (sempre segueixen la nomenclatura: ccma-seca.ami.xxx):

<https://bitbucket.ccma.cat/projects/SAWS/repos/ccma-seca.ami.apache-frontend/browse>

<https://bitbucket.ccma.cat/projects/SAWS/repos/ccma-seca.ami.cache/browse>

<https://bitbucket.ccma.cat/projects/SAWS/repos/ccma-seca.ami.centos7-base/browse>

<https://bitbucket.ccma.cat/projects/SAWS/repos/ccma-seca.ami.seleccio/browse>

ansible

Ansible és una eina per automatitzar el desplegament de configuracions a màquines. En el cas del cloud l'estem fent servir per posar les configuracions a les AMIs que creem.

terraform

Eina per al desplegament d'infraestructura. Actualment ho fem servir per generar infraestructura a AWS. Alguns projectes que tenim a Bitbucket amb Terraform (sempre segueixen la nomenclatura: ccma-seca.infraestructura.xxx):

<https://bitbucket.ccma.cat/projects/SAWS/repos/ccma-seca.infraestructura.hello-world/browse>

<https://bitbucket.ccma.cat/projects/SAWS/repos/ccma-seca.infraestructura.shared/browse>

<https://bitbucket.ccma.cat/projects/SAWS/repos/ccma-seca.infraestructura.transitgw/browse>

Algunes comandes i tips & tricks

Inicialització projecte

terraform init

Es descarrega totes les dependències del projecte

Planificació

terraform plan

Mostra el resultat de tot el que farà abans d'executar el projecte

Aplicació

terraform apply

Aplica tots els canvis que encara no s'hagin fet sobre a infraestructura. Si la infraestructura no existeix la crearà. Si ja existeix només aplicarà els canvis.

Destrucció

terraform destroy

Destruïx tota la infraestructura creada des del projecte que es trobi a l'estat de terraform.

Llistat

terraform show

Retorna informació de tota la infraestructura que ha estat desplegada amb el projecte

Format

terraform fmt --diff

Corregeix la visualització dels fitxers aplicant un correcte identat del codi

Debugar una comanda

Debugging

TF_LOG=DEBUG terraform init