

42-Cloud-1 Project:

1) Diagramme

Cf. piece jointe

2) Security Tables

a) myVpcBastionSecurityGroup:

Inbound				
Type	Protocol	Port Range	Source	Description
All traffic	All	All	0.0.0.0/0	Pour tester
SSH	TCP	22	0.0.0.0/0	J'accepte toutes les addresses IP sources qui se connectent a moi. Je peux aussi faire en sorte que je n'accepte que les addresses IP source provenant de l'ecole 42

Outbound				
Type	Protocol	Port Range	Destination	Description
All	All	All	0.0.0.0/0	Default: c est pareil que ci-dessus

b) myVpcDatabaseSecurityGroup:

Inbound				
Type	Protocol	Port Range	Source	Description
MYSQL/Aurora	TCP	3306	myVpcWebSecurityGroup	Je n'accepte que ce que le trafic qui provient de myVpcWebSecurityGroup, i.e. des instances Wordpress

				Je n ouvre que le port 3306 puisque les instances se connectent a moi en 'mySQL'
--	--	--	--	--

Outbound				
Type	Protocol	Port Range	Destination	Description
All	All	All	0.0.0.0/0	Default: c est pareil que ci-dessus

c) myVpcEfsMountTargetSecurityGroup:

Inbound				
Type	Protocol	Port Range	Source	Description
NFS	TCP	2049	myVpcWebSecurityGroup	Je n'accepte que ce que le trafic qui provient de myVpcWebSecurityGroup, i.e. des instances Wordpress Je n ouvre que le port 2049 puisque les instances se connectent a moi en 'NFS'
NFS	TCP	2049	myVpcEfsMountTargetSecurityGroup	!!!
SSH	TCP	22	myVpcBastionSecurityGroup	!!!

Outbound				
Type	Protocol	Port Range	Destination	Description
All	All	All	0.0.0.0/0	Default: c est pareil que ci-dessus

d) myVpcLoadBalancerSecurityGroup:

Inbound				
Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	J accepte tout le trafic ip provenant de mes clients Je n ouvre que le port 80 puisque les clients se connectent a moi en 'http'
HTTPS	TCP	443	0.0.0.0/0	!!! TBDr

Outbound				
Type	Protocol	Port Range	Destination	Description
All	All	All	0.0.0.0/0	Default: c est pareil que ci-dessus

e) myVpcWebSecurityGroup:

Inbound				
Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	myVpcLoadBalancerSecurityGroup	Je ne suis pas accessible depuis l exterieur et le suis pour le trafic provenant du Load Balancer
HTTPS	TCP	443	myVpcEfsMountTargetSecurityGroup	!!!

SSH	TCP	22	myVpcBastionSecurityGroup	L instance Bastion pourra se connecter a moi
-----	-----	----	---------------------------	--

Outbound				
Type	Protocol	Port Range	Destination	Description
All	All	All	0.0.0.0/0	Default: c est pareil que ci-dessus

3) Route Tables

a) myVpcPublicSubnetRouteTable:

myVpcPublicSubnetRouteTable			
Destination	Target	Propagated	Description
10.0.0.0/16	Local		
0.0.0.0/0	myVpcInternetGateway		Pour se connecter a l exterieur du reseau local

b) myVpcWebAndDataSubnet1aRouteTable:

myVpcWebAndDataSubnet1aRouteTable			
Destination	Target	Propagated	Description
10.0.0.0/16	Local		
0.0.0.0/0	myVpcNatGateway1a		<p>Je peux me connecter a l'exterieur mais personne ne pourra pas se connecter a moi directement</p> <p>Par exple, je veux upgrader Wordpress ou bien ma BD</p>

c) myVpcWebAndDataSubnet1bRouteTable:

myVpcWebAndDataSubnet1aRouteTable			
Destination	Target	Propagated	Description
10.0.0.0/16	Local		
0.0.0.0/0	myVpcNatGateway1b		<p>Je peux me connecter a l'exterieur mais personne ne pourra pas se connecter a moi directement</p> <p>Par exple, je veux upgrader Wordpress ou bien ma BD</p>

4) Load test instance

C'est une AMI classique (Free Tiers) où l'on installe ce script, qui permet de lancer des requêtes HTTP sur le load balancer pour tester le 'scaling'

```
e1r12p13% cat loadTest.sh
#!/bin/bash -ex
yum -y update
yum install -y httpd24-tools
ab -n 500000 -c 15 -t 600 -s 120 -r http://myVpcLoadBalancer-1118744953.eu-central-1.elb.amazonaws.com/
e1r12p13%
```

5) Wordpress First Install

C'est un script à améliorer :

- On update l'instance
- On installe :
 - o Httpd24 : apache
 - o php70 : version de php
 - o php70-mysqlnd : extension requise par php afin de se connecter à la BD
- On attend que le filesystem NFS (EFS) soit monté
- On installe Wordpress sur le montage NFS ainsi il sera 'vu' par toutes les instances de toutes les 'availability zones'
- On configure Wordpress

- On configure opcache afin que le site web wordpress soit plus responsif

```
#!/bin/bash -x
bash -ex << "TRY"
  EFS_ENDPOINT="fs-27f9217e.efs.eu-central-1.amazonaws.com"
  # update
  yum -y update

  # install httpd v2.4, php7. php70-mysqlnd (extension required by php to connect to mysql db),
  # php70-opcache (optimise)
  yum -y install httpd24 php70 php70-mysqlnd php70-opcache

  # wait until EFS file system is available
  while ! nc -z ${EFS_ENDPOINT} 2049; do sleep 10; done
  sleep 10

  # mount EFS file system
  echo "${EFS_ENDPOINT}:/ /var/www/ nfs4
nfsvers=4.1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2,_netdev 0 0" >> /etc/fstab
  mount -a

  # install wordpress
  cd /var/www/html
  wget https://wordpress.org/latest.zip
  chown -R apache:apache /var/www/html/
  chmod 2775 /var/www
  find /var/www -type d -exec chmod 2755 {} \;

  # config opcache
  sed -i 's/;opcache.revalidate_freq=2/opcache.revalidate_freq=300/g' /etc/php-7.0.d/10-
opcache.ini
```



```
# config httpd
perl -i -0pe 's/<Directory "\/var\/www">\n\s*AllowOverride None\n/<Directory "\/var\/www">\nAllowOverride All\n/' /etc/httpd/conf/httpd.conf
chkconfig httpd on

# config opcache
sed -i 's;/opcache.revalidate_freq=2/opcache.revalidate_freq=300/g' /etc/php-7.0.d/10-opcache.ini

# config httpd
perl -i -0pe 's/<Directory "\/var\/www">\n\s*AllowOverride None\n/<Directory "\/var\/www">\nAllowOverride All\n/' /etc/httpd/conf/httpd.conf
chkconfig httpd on

# start httpd
service httpd start

TRY
```

6) AutoScaling : userdata

Lorsque l'AutoScaling va créer des nouvelles instances, il faut lui donner de la 'matière' afin qu'il puisse configurer ces nouvelles instances au 'bootstrap'

Pas grande chose :

- On installe ce dont a besoin

- On configure
- Puis on demarre les services

```
# userdata

#!/bin/bash -x
bash -ex << "TRY"
    EFS_ENDPOINT="fs-27f9217e.efs.eu-central-1.amazonaws.com"
    #-----
    # Logs are stored in /var/log/cloud-init-output.log
    #-----
    # update
    yum -y update

    # install httpd v2.4, php7. php70-mysqldb (extension required by php to connect to mysql db),
    # php70-opcache (optimise)
    yum -y install httpd24 php70 php70-mysqldb php70-opcache

    # wait until EFS file system is available
    while ! nc -z ${EFS_ENDPOINT} 2049; do sleep 10; done
    sleep 10

    # mount EFS file system
    echo "${EFS_ENDPOINT}:/ /var/www/ nfs4
nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,_netdev 0 0" >> /etc/fstab
    mount -a

    # config opcache
    sed -i 's/;opcache.revalidate_freq=2/opcache.revalidate_freq=300/g' /etc/php-7.0.d/10-
    opcache.ini

    # config httpd
```

```
perl -i -0pe 's/<Directory "\/var\/www">\n\s*AllowOverride None\n/<Directory "\/var\/www">\nAllowOverride All\n/' /etc/httpd/conf/httpd.conf
chkconfig httpd on

# start httpd
service httpd start
```

TRY

7) Divers

a) Se connecter au Bastion

```
# Utiliser 1 agent ssh

ssh-add -K myFranckfurtKey.pem
ssh-add -L

# Se connecter a 1 instance Bastion :
# Copier 1 adresse IP public de cette instance

ssh -A ec2-user@<adresse IP public de 1 instance Bastion>

<Bastion prompt $> ssh ec2-user@<address ip privee de 1 instance de votre choix>
```

b) Afficher l'adresse ip de l'instance Web sur le site Web

J'ai ajouté le plugin 'What the file'

- Cela me permet de repérer directement le fichier php que je dois modifier
- J'ai choisi de modifier le fichier 'partials/header/logo.php' et d'ajouter ces instructions php à l'endroit où le site affiche le nom du site

PS. J'ai aussi utilisé un plugin Elementor afin de configurer WordPress. Donc il se peut que le fichier à modifier soit différent

```
$server_ip = getHostByName(getHostName());  
echo esc_html( get_bloginfo( 'name' ) . ": " . $server_ip );
```

c) Sauvegarder en format fichier yaml l'infrastructure configurée sur le cloud AWS

J'ai utilisé CloudFormer de AWS

PS : Je n'ai pas testé le re-déploiement à l'identique à partir de ce fichier. Car la correction de ce projet n'est pas encore faite

Cf. fichier : *42-cloud-1.project.by.cloudformer.yaml*