Installation de wordpress et Load balancer sur deux VM

Installation de wordpress et load balancer sur deux VM's par script.

Driton Gorani

20/06/2022 15:35

Copy commit SHA GITLAB: `e7a9b5524d29867e6fd16679270adc5fd29b49a4`

Recuperation des donnés.

```
echo "Enter your ressource group name?"
read RGname
#echo "Enter your VM username?"
#read Vmusername
#echo "Enter your VM password?"
#read Vmpassword
#echo "Enter your Mariadb name?"
#read  MdbName
#echo "Enter your Mariadb username?"
#read Mariadbuser
#echo "Enter your Mariadb password?"
#read Mariadbpass
Vmusername=tonytony
Vmpassword=Driton123456.
MdbName=mymariadbtony
Mariadbuser=dritontony
Mariadbpass=Tonytony123456.
location=eastus
```

Creation d'une groupe de ressource:

`az group create --name $RGname --location $location`

Creation d'un virtual network

`az network vnet create --resource-group $RGname --location $location --name myVNet --address-prefixes 10.1.0.0/16 --subnet-name myBackendSubnet --subnet-prefixes 10.1.0.0/24`

Creation d'un IP Public

`az network public-ip create --resource-group $RGname --name myPublicIP --sku`

`Standard --location $location`

Creation d'un Load Balancer

`az network lb create --resource-group $RGname --name myLoadBalancer --sku Standard --public-ip-address myPublicIP --frontend-ip-name myFrontEnd --backend-pool-name myBackEndPool`

Creation d'un load balancer health probe

`az network lb probe create --resource-group $RGname --lb-name myLoadBalancer --name myHealthProbe --protocol tcp --port 80`

Creation d'un load balancer HTTP RULE

`az network lb rule create --resource-group $RGname --lb-name myLoadBalancer --name myHTTPRule --protocol tcp --frontend-port 80 --backend-port 80 --frontend-ip-name myFrontEnd --backend-pool-name myBackEndPool --probe-name myHealthProbe --disable-outbound-snat true --idle-timeout 15 --enable-tcp-reset true`

Creation d'une groupe de security NSG

`az network nsg create --resource-group $RGname --name myNSG`

Create network securit group rule

`az network nsg rule create --resource-group $RGname --nsg-name myNSG --name myNSGRuleHTTP --protocol '*' --direction inbound --source-address-prefix '*' --source-port-range '*' --destination-address-prefix '*' --destination-port-range 80 --access allow --priority 200`

NSG Rule pour le port 22 SSH

```
az network nsg rule create --resource-group $RGname --nsg-name myNSG --name
NSGule22 --protocol '*' --direction inbound --source-address-prefix '*' --
source-
port-range '*' --destination-address-prefix '*' --destination-port-range 22
--access-allow --priority 300
```

Créer des interfaces réseau pour les machines virtuelles

```
array=(myNicVM1 myNicVM2)
    for vmnic in "${array[@]}"
    do
        az network nic create --resource-group $RGname --name $vmnic --vnet-
name myVNet --subnet myBackEndSubnet --network-security-group myNSG
  done
```

Premier machine virtuelle

`az vm create --resource-group $RGname --name myVM1 --nics myNicVM1 --image Debian:debian-11-daily:11-gen2:0.20220521.1022 --admin-username $Vmusername --admin-password $Vmpassword --zone 1 --no-wait --location $location`

Deuxieme machine virtuelle

`az vm create --resource-group $RGname --name myVM2 --nics myNicVM2 --image Debian:debian-11-daily:11-gen2:0.20220521.1022 --admin-username $Vmusername --admin-password $Vmpassword --zone 2 --no-wait --location $location`

Ajouter des machines virtuelles au pool de back-ends de l'équilibreur de charge

```
array=(myNicVM1 myNicVM2)
  for vmnic in "${array[@]}"
  do
    az network nic ip-config address-pool add --address-pool myBackendPool --ip-config-name ipconfig1 --nic-name $vmnic --resource-group $RGname --lb-name myLoadBalancer
  done
```

Créer une passerelle NAT

IP Adresse publique.

az network public-ip create --resource-group $RGname --name myNATgatewayIP --sku Standard --zone 1 2 3

Ressource de passrelle NAT.

`az network nat gateway create --resource-group $RGname --name myNATgateway --public-ip-addresses myNATgatewayIP --idle-timeout 10`

Associer une passerelle NAT au sous-réseau.

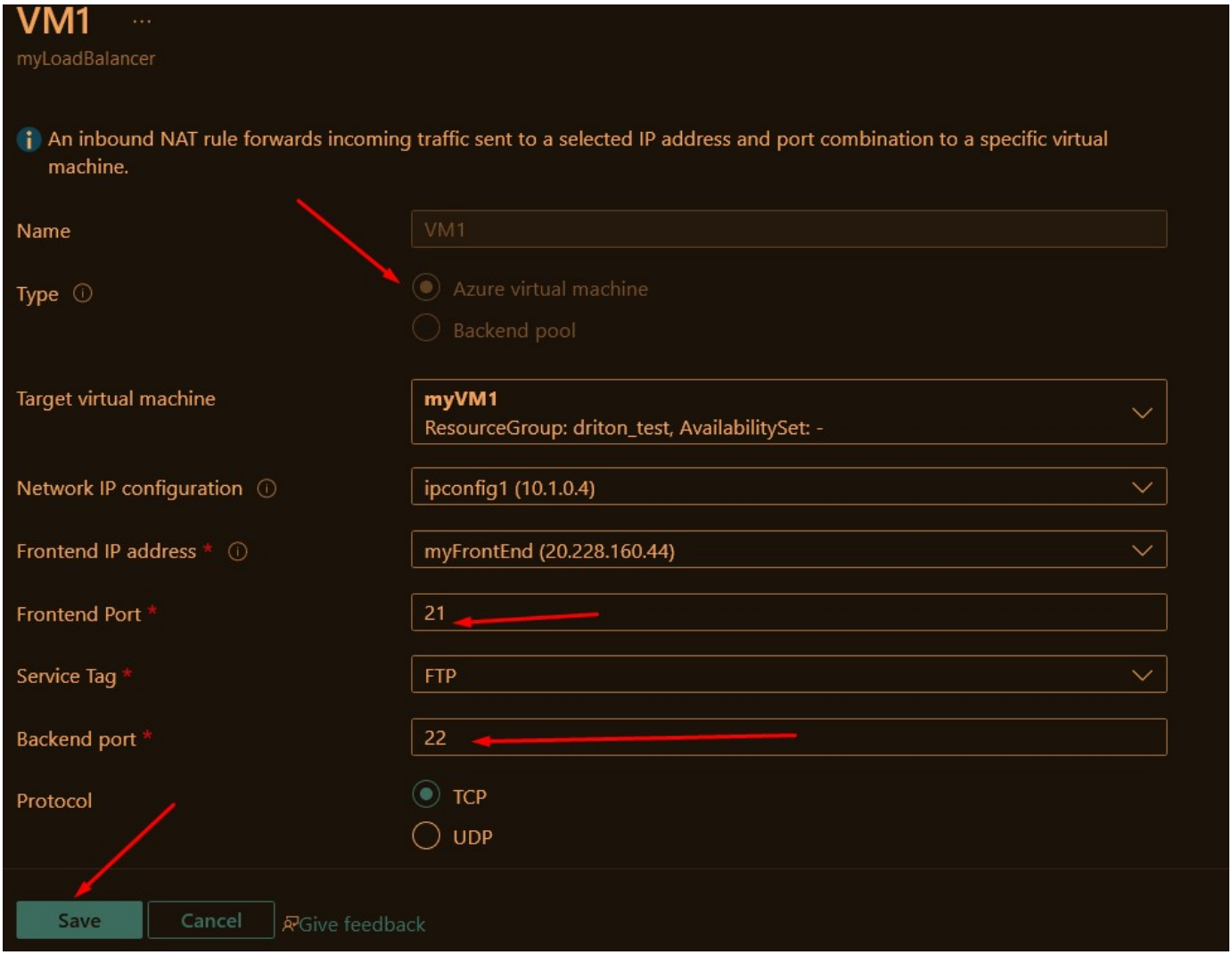
`az network vnet subnet update --resource-group $RGname --vnet-name myVNet --name myBackendSubnet --nat-gateway myNATgateway`

Création d'une base de donnés MARIADB-Server

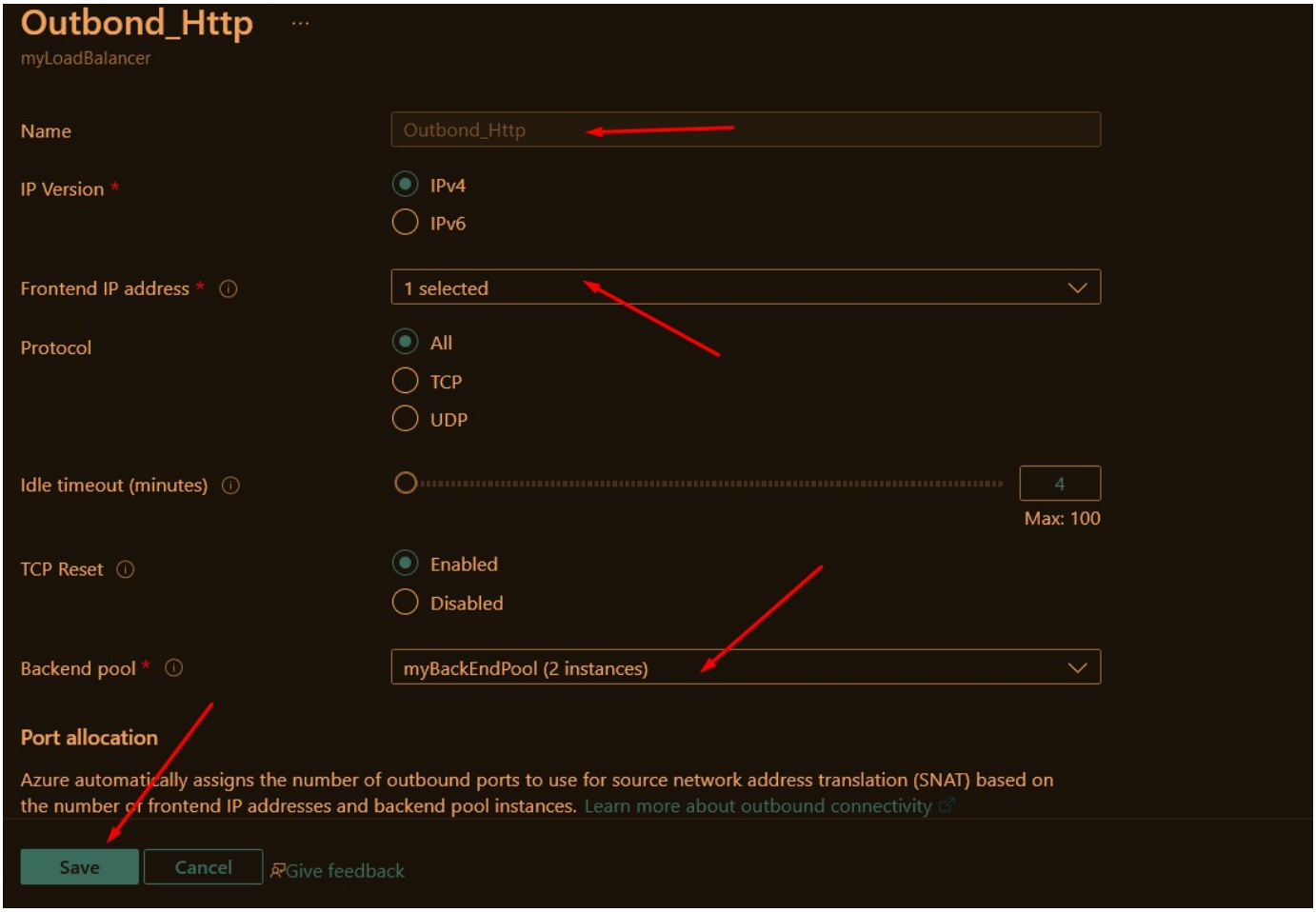
`az mariadb server create -l $location -g $RGname -n $MdbName -u $Mariadbuser -p $Mariadbpass --sku-name B_Gen5_1 --ssl-enforcement Disabled --backup-retention 10 --geo-redundant-backup disabled --storage-size 51200 --tags "key=value" --version 10.2`
``

1. Nous devons ajouter une regle NAT Inboud dans Load Balancer pour pouvoir acceder a nos vm's via SSH porte/22.
2. Ajouter une regle Outbound pour pouvoir donner access a nos VMs sur l'internet porte 80 HTTP.
3. Autoriser porte 22/SSH sur nos Vm's pour pouvoir y'acceder.

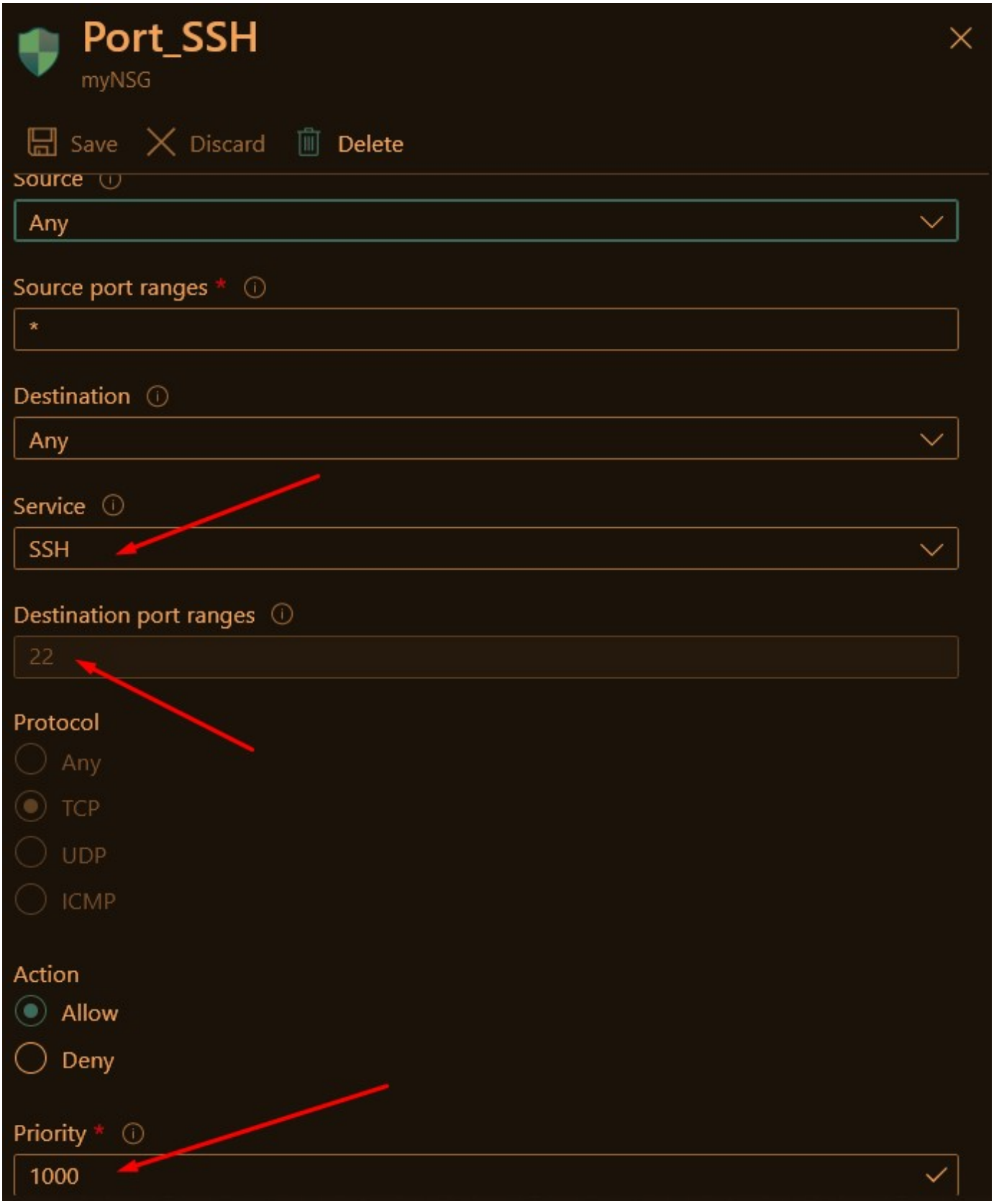
Ouvrir une porte d'éntrer dans load-balancer INBOUND NAT RULE pour nos deux vms, faire la méme chose pour deuxieme vm.



Ajouter une regle Outbound pour pouvoir donner access a nos VMs sur l'internet porte 80 HTTP.



Autoriser porte 22/SSH sur nos Vm's pour pouvoir y'acceder dans: (Networking)



Installation de wordpress par script

Data collection

```
data_base_name=wordpressttony
Mariadbpass=Tonytony123456.
Mariadbuser=dritontony
```

```
Hostnamewp=mymariadbtony.mariadb.database.azure.com
username_wp=wp-admin
password_wp=Driton123456.
```

```
sudo apt update -y
sudo apt upgrade -y
sudo apt install -y apache2 mariadb-client php-mysql php wget
wget -O /tmp/wordpress.tar.gz https://wordpress.org/latest.tar.gz
sudo tar -xzvf /tmp/wordpress.tar.gz -C /var/www/html
sudo chown -R www-data.www-data /var/www/html/wordpress
sudo chmod -R 755 /var/www/html/wordpress
sudo cp var/www/html/wordpress/wp-config-sample.php
var/www/html/wordpress/wp-config.php
sudo chown -R www-data:www-data /var/www/html/
sudo service apache2 reload
```

```
#echo "Enter your Mariadb host name?"
#read MariadbHost
#echo "Enter your mariadb username?"
#read Mariadbuser
#echo "Enter your mariadb password?"
#read Mariadbpass
```

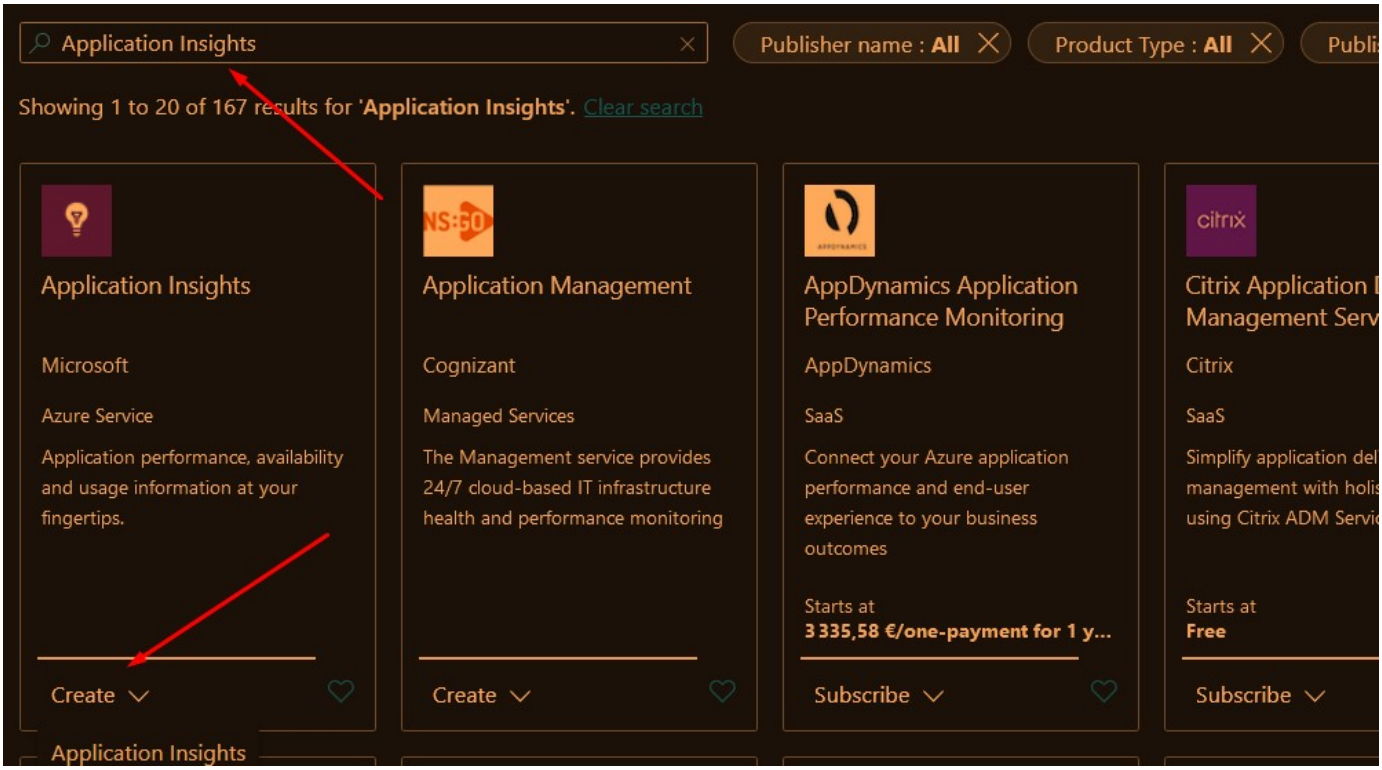
```
cd /var/www/html/wordpress
sudo sed -i "s/database_name_here/$data_base_name/" wp-config.php
sudo sed -i "s/username_here/$Mariadbuser/" wp-config.php
sudo sed -i "s/password_here/$Mariadbpass/" wp-config.php
sudo sed -i "s/localhost/$Hostnamewp/" wp-config.php
```

> *Nous devons nous connecter sur mariadb-client avec notre adresse de basse nom d'utilisateur et mot de passe,pour créer une basse de donné.*
>
> ```
> mysql -h (hostname) -u (username) -p (pass)
> CREATE DATABASE "NOM3";
> mysql> GRANT ALL PRIVILEGES ON database_name.* TO
> 'username'@'localhost';
> FLUSH PRIVILEGES;
> ```

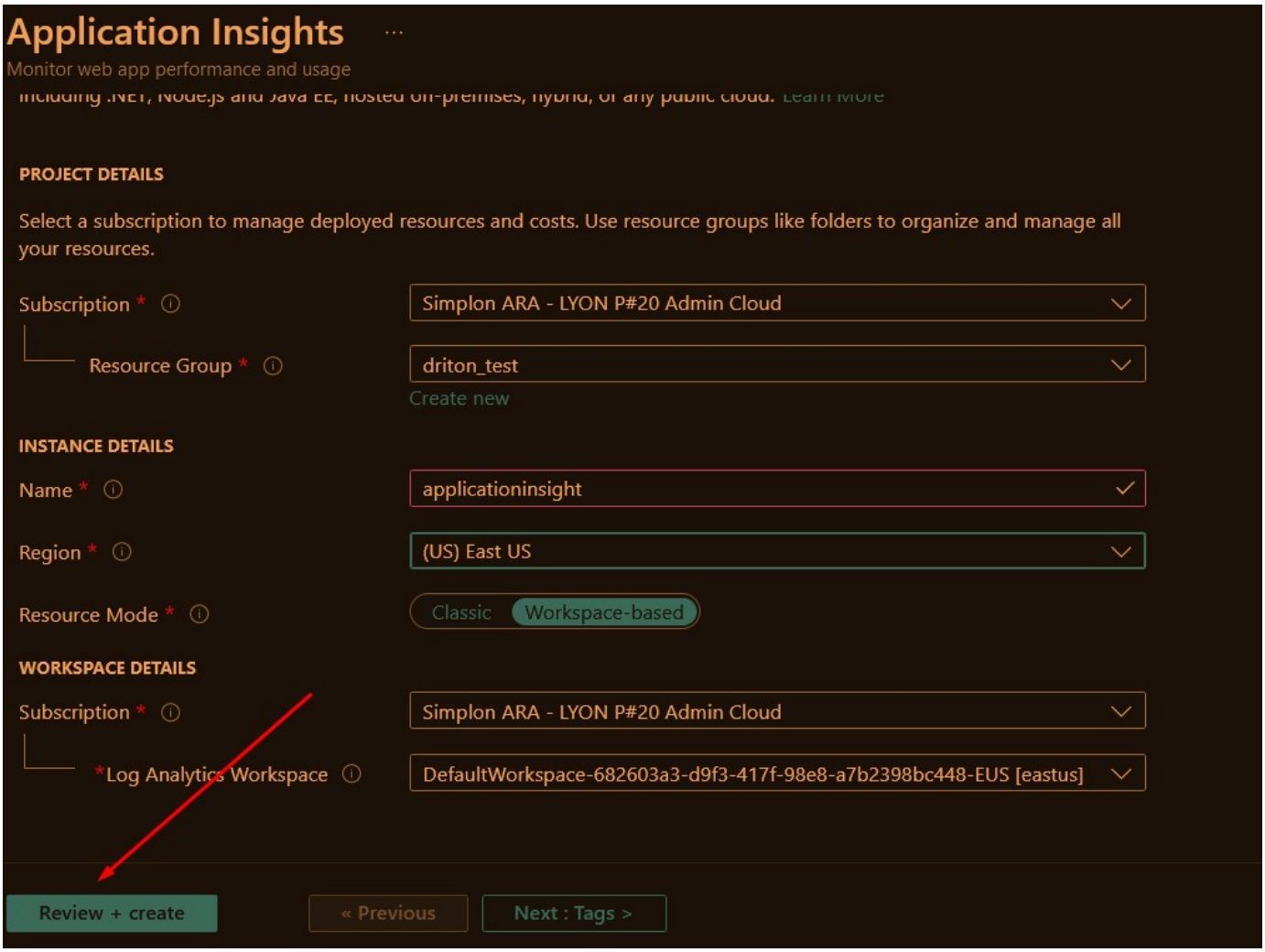
Installation d'Azure Monitor

---

*Nous chercons "Application Insight" dans la barre de rechereche sur le Portail Azure.

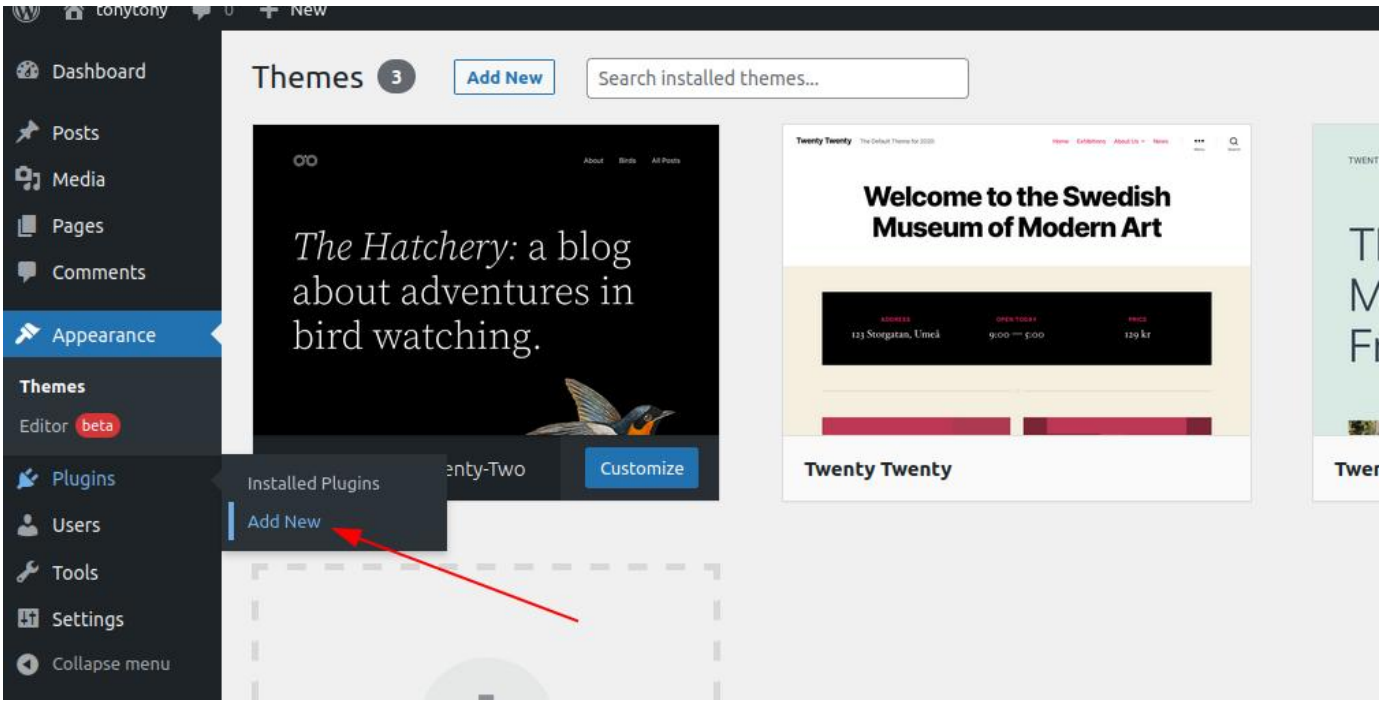


Nous remplissons le formulaire

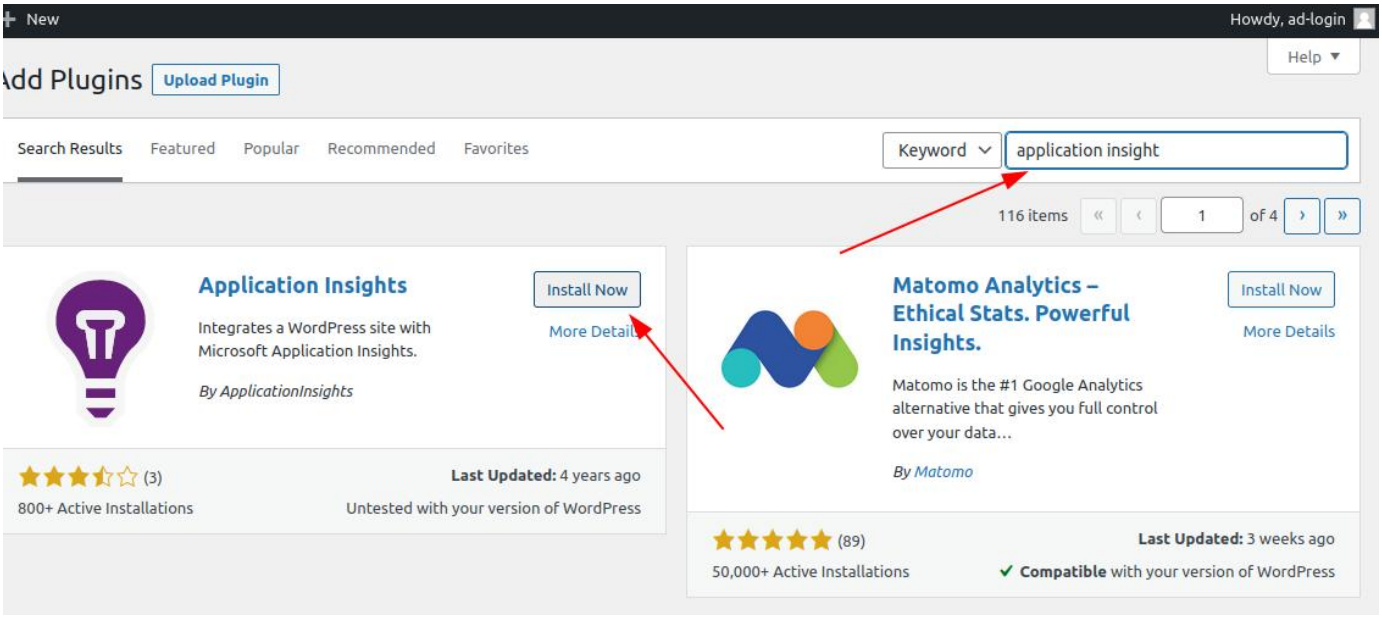


Nous installons le pluggin dans notre site internet wordpress

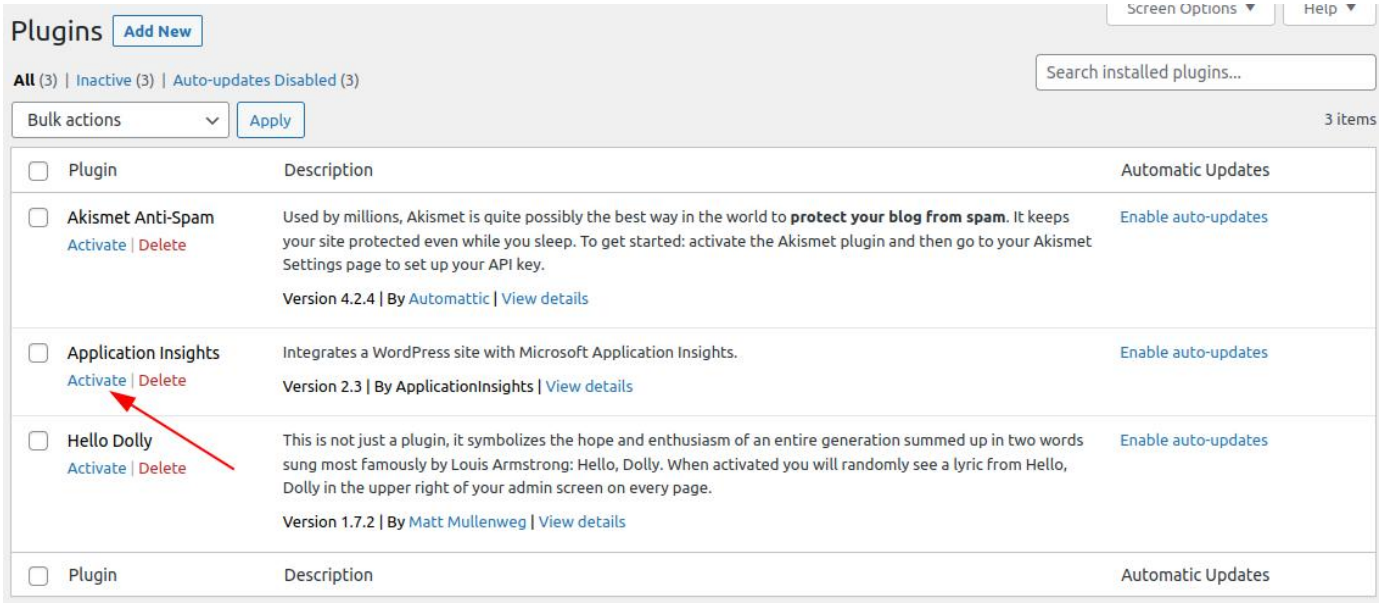
- Pluggin
- Add new



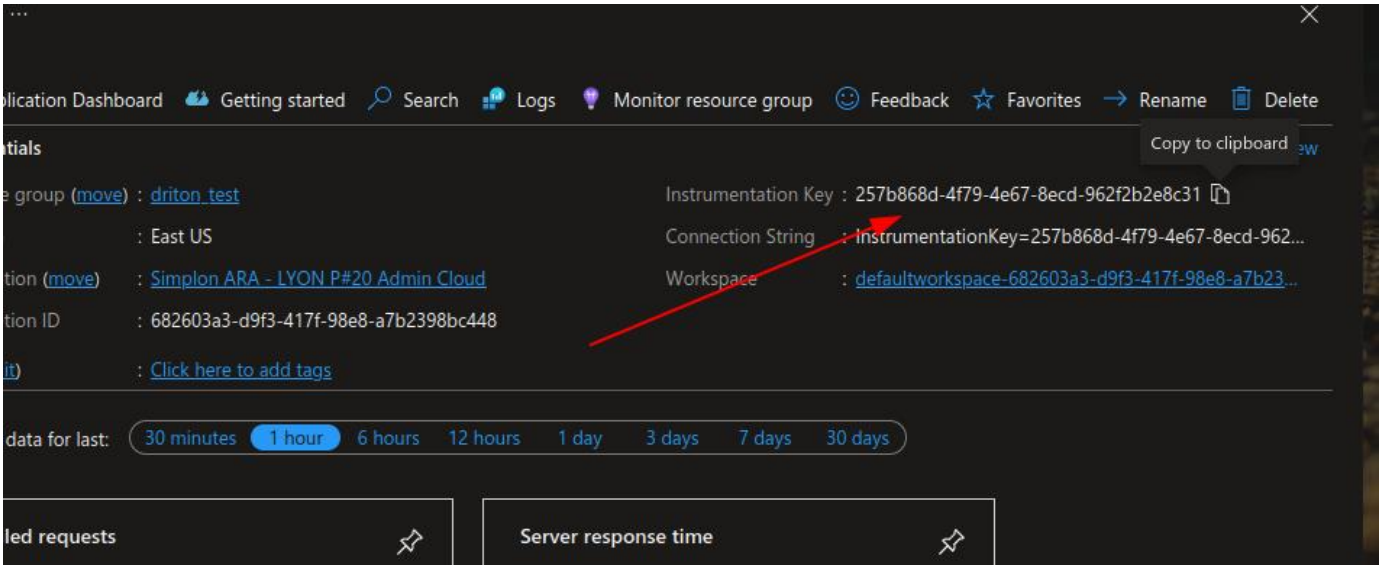
Nous cherchons application insight dans la barre de recherche



Activer l'application

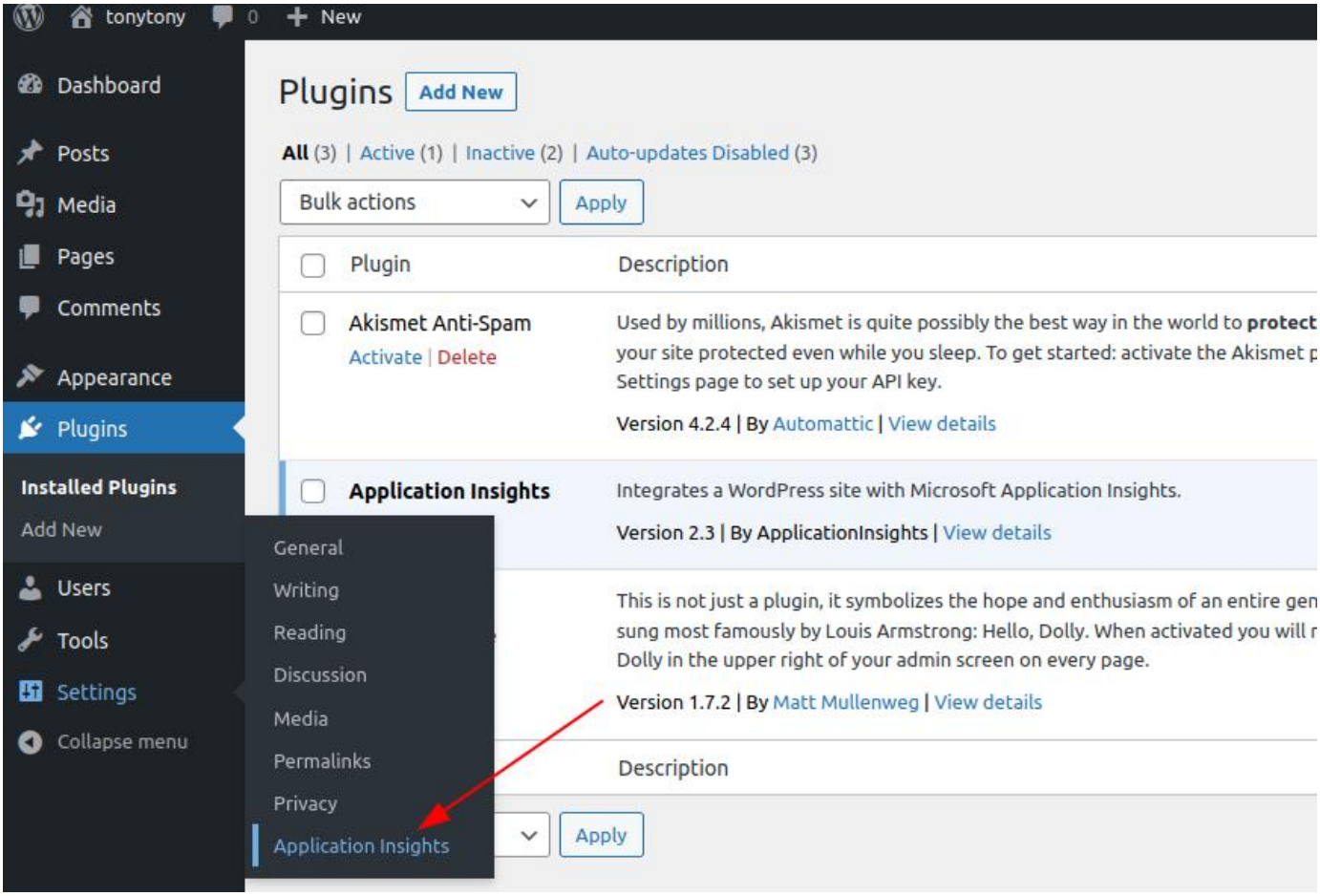


Copier la clef de notre application insight depuis le portail azure



Coller la clef directement dans notre site wordpress

- Settings
- Application insight



Nous arretons une vm pour tester le load balancer, et puis on renverse.Grace a load balancer on doit toujours avoir accés a notre site web méme si une serveur est étient.