

MyGestIONR

1) Présentation générale

MyGestIONR est un outil de gestion de parc informatique orienté réseau, il a pour but de faciliter le quotidien de technicien en réseau en incluant une variété d'outils pour effectuer certaines tâches comme le regroupement d'équipement réseau dans une base de données ou encore des pings et des vérifications sur l'état de certains équipements.

2) Détail des fonctionnalités

MyGestIONR possède une variété d'outils incluant une base de données mais pas seulement, MyGestIONR utilise MariaDB afin de pouvoir avoir une base de données permettant de regrouper les différents équipements. Pour modifier cette base de données, l'outil dispose de scripts d'ajouts, de modifications, de suppressions et d'affichage de données pour avoir un maximum d'options de modifications. En plus de ces fonctionnalités de modifications de base de données, MyGestIONR possède un système de backup permettant à l'utilisateur de pouvoir créer des sauvegardes de la base de données et de pouvoir restaurer les précédentes. **En ce qui concerne les outils réseaux, MyGestIONR est équipé d'un script de ping d'IP pour pouvoir vérifier si une IP est connectée ou non, il est également équipé d'un outil permettant de vérifier si un port TCP est ouvert ou non sur une adresse IP.** Il est également possible de réaliser un fichier .txt contenant l'ensemble de la base de données afin de pouvoir exporter les données ailleurs. Le logiciel est capable de consulter les IP actives ou non. **Tout ceci n'est pas réalisable sans l'utilisation du script d'installation au préalable qui s'occupe d'installer les paquets nécessaires ainsi que de créer la base de données**

3) Mes réalisations

Pour le projet MyGestIONR, j'ai pu réaliser la partie réseau (TestTCP et TestIP) ainsi que le fichier d'installation permettant au logiciel de pouvoir fonctionner convenablement, en voici quelques captures d'écran :

```

install > $ install.sh
1  #!/bin/bash
2
3  currentUser=$(whoami)
4  if [[ $currentUser != 'root' ]]; then
5      echo "Connectez vous en root pour continuer l'installation"
6      exit 2
7  fi
8
9  echo "-----Etape 1, installation client/serveur-----"
10
11 read -p "L'outil est-il sur une machine serveur ou cliente ? (0 pour client, 1 pour serveur)" Serv
12
13 while [ $Serv -ne 0 -a $Serv -ne 1 ]; do
14     echo "Veuillez saisir 0 pour une installation serveur, 1 pour installation client"
15     read -p "L'outil est-il sur une machine serveur ou cliente ? (0 pour client, 1 pour serveur)" Serv
16 done
17
18 if [[ $Serv -eq 0 ]]; then
19     echo serveur=0 > ../config.sh
20     install=0
21 fi
22
23 if [[ $Serv -eq 1 ]]; then
24     echo serveur=1 > ../config.sh
25     install=1
26 fi

```

```

27
28 echo "-----Etape 2, installation des paquets nécessaires-----"
29
30 apt update && apt install -y mariadb-server figlet netcat-openbsd sudo
31
32 if [[ $install -eq 0 ]]; then
33     apt install -y lolcat
34 fi
35
36 echo "-----Etape 3, installation de la BDD-----"
37
38 mysql -e "
39 CREATE DATABASE MyGest;
40 CREATE USER 'GestAdmin'@'localhost';
41 GRANT ALL PRIVILEGES ON MyGest.* TO 'GestAdmin'@'localhost';
42 FLUSH PRIVILEGES;"
43
44 mysql -u GestAdmin MyGest < myGestIOOnR.sql

```

```
1  #!/bin/bash
2  source fonction.sh
3
4  # Fonction pour valider l'adresse IP
5  valider_ip() {
6      local ip=$1
7      local regex='^([0-9]{1,3}\.){3}[0-9]{1,3}$'
8
9      if [[ $ip =~ $regex ]]; then
10         IFS='.' read -r octet1 octet2 octet3 octet4 <<< "$ip"
11         if (( octet1 >= 0 && octet1 <= 255 )) && \
12             (( octet2 >= 0 && octet2 <= 255 )) && \
13             (( octet3 >= 0 && octet3 <= 255 )) && \
14             (( octet4 >= 0 && octet4 <= 255 )); then
15             return 0
16         fi
17     fi
18     return 1
19 }
20
21 # Fonction pour vérifier si le port est valide
22 valider_port() {
23     local port=$1
24     # Vérifier si le port est un entier dans la plage de 0 à 65535
25     if [[ "$port" =~ ^[0-9]+$ ]] && (( port >= 0 && port <= 65535 )); then
26         return 0
27     else
28         return 1
29     fi
30 }
```

```
31
32 # Demander à l'utilisateur une adresse IP
33 read -p "Entrez une adresse IP : " ip
34
35 # Vérifier si l'adresse IP est valide
36 ✓ if ! valider_ip "$ip"; then
37 |     echo "L'adresse IP '$ip' n'est pas valide."
38 |     exit 1
39 fi
40
41 # Demander à l'utilisateur un numéro de port
42 read -p "Entrez un numéro de port : " port
43
44 # Vérifier si le port est valide
45 ✓ if ! valider_port "$port"; then
46 |     echo "Le port '$port' n'est pas valide."
47 |     exit 1
48 fi
49
50 # Test du port avec netcat
51 nc -zv "$ip" "$port" &>/dev/null
52
53 # Vérification du statut de la commande précédente
54 ✓ if [ $? -eq 0 ]; then
55 |     echo "Le port $port est ouvert sur l'adresse IP $ip."
56 ✓ else
57 |     echo "Le port $port est fermé sur l'adresse IP $ip."
58 fi
59
60 afficheTitre 'Outils reseau'
```

```
1  #!/bin/bash
2  source fonction.sh
3
4  valideIP=0
5
6  # Fonction pour valider l'adresse IP
7  valider_ip() {
8      local ip=$1
9      local regex='^([0-9]{1,3}\.){3}[0-9]{1,3}$'
10
11      if [[ $ip =~ $regex ]]; then
12          IFS='.' read -r octet1 octet2 octet3 octet4 <<< "$ip"
13          if (( octet1 >= 0 && octet1 <= 255 )) && \
14              (( octet2 >= 0 && octet2 <= 255 )) && \
15              (( octet3 >= 0 && octet3 <= 255 )) && \
16              (( octet4 >= 0 && octet4 <= 255 )); then
17              return 0
18          fi
19      fi
20      return 1
21  }
22
```

```

22
23 while [ $validiteIP -eq 0 ]; do
24     read -p "Veuillez saisir une adresse IP : " ipAddress
25
26     if valider_ip "$ipAddress"; then
27         echo "Envoie de la requête ICMP ..."
28         ping -c 1 -W 1 "$ipAddress" &> /dev/null
29
30         if [ $? -eq 0 ]; then
31             echo "L'adresse IP répond aux pings."
32             validiteIP=1
33         else
34             echo "L'adresse IP ne répond pas aux pings."
35         fi
36     else
37         echo "L'adresse IP n'est pas valide."
38     fi
39 done
40
41 echo "[Appuyez sur entrée pour continuer]"
42 read
43
44 afficheTitre 'Outils reseau'
45

```

4) Démonstration technique

En ce qui concerne mes réalisations ainsi que une démonstration technique permettant de comprendre les fonctionnalités, je vous invite à regarder les capsules vidéos suivante :

<https://vimeo.com/1067430325/732017173a>

<https://vimeo.com/1067430361/674680388e>

<https://vimeo.com/1067430953/321f826f55>

5) Sources

Base du projet:

<https://pedagogeek.fr/public/>

Aide extérieur:

<https://chatgpt.com>