

**add 25 26**

xavki authored 1 year ago

deploy.sh 3.52 KiB

```
1  #!/bin/bash
2
3  #####
4  #
5  #  Description : déploiement à la volée de conteneur docker
6  #
7  #  Auteur : Xavier
8  #
9  #  Date : 28/12/2018 - V2.0
10 #
11 #####
12
13
14 # Functions #####
15
16 help(){
17 echo "
18
19 Options :
20
21     - --create : lancer des conteneurs
22
23     - --drop : supprimer les conteneurs créer par le deploy.sh
24
25     - --infos : caractéristiques des conteneurs (ip, nom, user...)
26
27     - --start : redémarrage des conteneurs
28
29     - --ansible : déploiement arborescence ansible
30
31 "
32 }
33
34 createNodes() {
35     # définition du nombre de conteneur
36     nb_machine=1
37     [ "$1" != "" ] && nb_machine=$1
38     # setting min/max
39     min=1
40     max=0
41
42     # récupération de idmax
43     idmax=`docker ps -a --format '{{.Names}}' | awk -F "-" -v user="$USER" '$0 ~ user"-debian" {print $3}' |`
44     # redéfinition de min et max
45     min=$((idmax + 1))
46     max=$((idmax + $nb_machine))
47
48     # lancement des conteneurs
49     for i in $(seq $min $max);do
50         docker run -tid --privileged --publish-all=true -v /srv/data:/srv/html -v /sys/fs/cgroup:/sys/fs/
51         docker exec -ti $USER-debian-$i /bin/sh -c "useradd -m -p sa3tHJ3/KuYvI $USER"
52         docker exec -ti $USER-debian-$i /bin/sh -c "mkdir ${HOME}/.ssh && chmod 700 ${HOME}/.ssh && chow
53     docker cp $HOME/.ssh/id_rsa.pub $USER-debian-$i:$HOME/.ssh/authorized_keys
54     docker exec -ti $USER-debian-$i /bin/sh -c "chmod 600 ${HOME}/.ssh/authorized_keys && chown $USER:$USER $
55     docker exec -ti $USER-debian-$i /bin/sh -c "echo '$USER  ALL=(ALL) NOPASSWD: ALL'>>/etc/sudoers"
56     docker exec -ti $USER-debian-$i /bin/sh -c "service ssh start"
57     echo "Conteneur $USER-debian-$i créé"
58     done
59     infosNodes
60
61 }
62
63 dropNodes(){
64     echo "Suppression des conteneurs..."
65     docker rm -f $(docker ps -a | grep $USER-debian | awk '{print $1}')
66     echo "Fin de la suppression"
67 }
68
69 startNodes(){
70     echo ""
```

```
70     docker start $(docker ps -a | grep $USER-debian | awk '{print $1}')
71 for conteneur in $(docker ps -a | grep $USER-debian | awk '{print $1}');do
72     docker exec -ti $conteneur /bin/sh -c "service ssh start"
73 done
74     echo ""
75 }
76
77
78 createAnsible(){
79     echo ""
80     ANSIBLE_DIR="ansible_dir"
81     mkdir -p $ANSIBLE_DIR
82     echo "all:" > $ANSIBLE_DIR/00_inventory.yml
83     echo "  vars:" >> $ANSIBLE_DIR/00_inventory.yml
84     echo "    ansible_python_interpreter: /usr/bin/python3" >> $ANSIBLE_DIR/00_inventory.yml
85     echo "  hosts:" >> $ANSIBLE_DIR/00_inventory.yml
86     for conteneur in $(docker ps -a | grep $USER-debian | awk '{print $1}');do
87         docker inspect -f '    {{.NetworkSettings.IPAddress }}:' $conteneur >> $ANSIBLE_DIR/00_inventory.yml
88     done
89     mkdir -p $ANSIBLE_DIR/host_vars
90     mkdir -p $ANSIBLE_DIR/group_vars
91     echo ""
92 }
93
94 infosNodes(){
95     echo ""
96     echo "Informations des conteneurs : "
97     echo ""
98     for conteneur in $(docker ps -a | grep $USER-debian | awk '{print $1}');do
99         docker inspect -f '    => {{.Name}} - {{.NetworkSettings.IPAddress }}' $conteneur
100     done
101     echo ""
102 }
103
104
105
106 # Let's Go !!! ##### "
107
108 #si option --create
109 if [ "$1" == "--create" ];then
110     createNodes $2
111
112 # si option --drop
113 elif [ "$1" == "--drop" ];then
114     dropNodes
115
116 # si option --start
117 elif [ "$1" == "--start" ];then
118     startNodes
119
120 # si option --ansible
121 elif [ "$1" == "--ansible" ];then
122     createAnsible
123
124 # si option --infos
125 elif [ "$1" == "--infos" ];then
126     infosNodes
127
128 # si aucune option affichage de l'aide
129 else
130     help
131
132 fi
133
134
135
136
137
```