

Cours : présentation de Kathara

Présentation de Kathar  

Kathar   est un Framework de l'Universit   de Rome qui succ  de    **Netkit** en utilisant **Docker** et **Python**. Cette solution permet de cr  er de concevoir et de tester des architectures de r  seaux locaux en impl  mentant ma-chines virtuelles l  g  res sous la forme de conteneur Docker reli  s ensemble par un r  seau virtuel ind  pendant du r  seau de la machine d'accueil. Il est ainsi possible de tester une configuration r  seau complexe sans la n  cessit   de droits sp  ciaux sur l'**ordinateur h  te**.

La r  alisation d'infrastructure r  seau est de nos jour de plus en plus complexe car cela met en   uvre :

- Des ordinateurs diff  rents de type serveur et client,
- Des services et protocoles r  seaux divers (DHCP, DNS, Web, Supervision, RIP, etc.)
- Des   quipements actifs comme les switches, les routeurs, les pare feux,
- La gestion de plusieurs interfaces r  seaux au niveau d'un m  me   quipement,

Cela se traduit par des topologies r  seaux vari  es en plus d'  tre complexes.

Dans un cadre d'enseignement, o   l'on ne dispose pas des mat  riels n  cessaires pour impl  menter les topolo-gies r  seaux afin de les   tudier et les tester, il est possible d'utiliser des logiciels de simulation comme Cisco Paquet Tracer. Mais ce logiciel qui permet l'  tude d'infrastructure r  seau est moins adapt      la mise en place de services r  seaux. Il permet de reproduire un certain nombre de fonctionnalit  s sans en reproduire le comporte-ment r  el en termes de performance.

Kathar   permet de mettre en   uvre des protocoles r  cents permettant la virtualisation des fonctions de r  seau (NFV) et la mise en r  seau d  finie par logiciel (SDN). Cela va modifier la mani  re de mettre en r  seau des ser-vices, en permettant la programmation des infrastructures r  seaux dans le but de s  parer la logique de l'infrastructure    r  aliser du mat  riel qui va la mettre en   uvre. Ensemble, ils pr  sentent plusieurs avantages, principalement en termes d'  volutivit   et de flexibilit  , pour d  ployer des fonctions de r  seau virtuel (VNF)

Kathar   permet d'  muler des r  seaux d'ordinateurs de type Linux en utilisant Docker. Chaque   quipement r  seau (serveur, client, routeur, switch) est un conteneur Docker. La souplesse de Docker permet d'avoir des conteneurs bas  s sur des images personnalis  e permettant la mise en   uvre de :

- **Quagga**, une suite de logiciels de routage impl  mentant les protocoles OSPF, RIP, BGP et IS-IS pour les routeurs,
- **Open vSwitch**, une solution de gestion de commutateur virtuel,
- **P4** un langage de programmation des   quipements r  seaux.

Dans le cours ne sera pas abord   que ces fonctionnalit  s de l'utilisation de Kathara permettant de maquetter des infrastructures r  seaux offrant des services de base.

Lien vers le site officiel de Kathara de l'Universit   de Rome : <http://www.kathara.org/>

L'  mulation de p  riph  riques r  seau

Avec Kathara, chaque   quipement r  seau qui sera cr    dans un conteneur Docker poss  de :

- Une console en mode texte,
- De la m  moire,
- Un syst  me de fichiers,
- Une ou plusieurs interfaces r  seaux selon vos besoins.

Voici un exemple d'  quipement r  seau qui correspond    un client :

```

sta1
root@sta1:/# route
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Ifac
e
default          ubuntu docker  0.0.0.0         UG    0      0      0 eth0
172.20.0.0        0.0.0.0        255.255.0.0     U      0      0      0 eth0
root@sta1:/# traceroute www.cned.fr
traceroute to www.cned.fr (195.220.181.14), 30 hops max, 60 byte packets
 1  ubuntu docker (172.20.0.1)  0.049 ms  0.008 ms  0.007 ms
 2  bbox.lan (192.168.1.1)  1.792 ms  2.220 ms  2.199 ms
 3  176-145-144-2.abo.bbox.fr (176.145.144.2)  6.436 ms  6.595 ms  6.804 ms
 4  212.194.170.233 (212.194.170.233)  14.803 ms  14.781 ms  14.761 ms
 5  be5.cbr01-ntr.net.bbox.fr (212.194.171.137)  14.396 ms  13.626 ms  13.72
0 ms

```

En utilisant Docker, Kathara permet :

- De créer et de gérer plusieurs **machines virtuelles (VMs)** sous forme de **conteneurs Docker**,
- De relier ces VMs à des **domaines de collision** qui sont des hubs virtuels pour permettre aux VMs de communiquer entre elles,
- De définir le rôle de chaque VM soit comme simple ordinateur **client linux**, **comme serveur en installant si nécessaire des paquets logiciels supplémentaires, comme routeur ou comme switch**. ===== Retour Accueil SIS3 ===== * **SIS3**

From:

/ - Les cours du BTS SIO

Permanent link:

</doku.php/kathara/decouverte?rev=1567952802>

Last update: 2019/09/08 16:26

